

**IMPLEMENTASI MODIFIKASI BOOST KONVERTER DC-
DC TOPOLOGI SEPIC PADA PROTOTIPE PEMBANGKIT
LISTRIK TENAGA SURYA DILENGKAPI SISTEM
MONITORING ARUS DAN TEGANGAN BERBASIS
INTERNET OF THINGS DI BANDAR UDARA TANJUNG
REDEP KALIMARAU BERAU**

PROYEK AKHIR



Oleh :

ULFIANA DYAH PRAMESTI
NIT. 30121024

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2024

**IMPLEMENTASI MODIFIKASI BOOST KONVERTER DC-
DC TOPOLOGI SEPIC PADA PROTOTIPE PEMBANGKIT
LISTRIK TENAGA SURYA DILENGKAPI SISTEM
MONITORING ARUS DAN TEGANGAN BERBASIS
INTERNET OF THINGS DI BANDAR UDARA TANJUNG
REDEP KALIMARAU BERAU**

PROYEK AKHIR

Diajukan sebagai Syarat Menempuh Mata Kuliah Tugas Akhir pada
Program Studi Diploma 3 Teknik Listrik Bandara.



Oleh :

ULFIANA DYAH PRAMESTI
NIT. 30121024

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2024

LEMBAR PERSETUJUAN

IMPLEMENTASI MODIFIKASI BOOST KONVENTER DC-DC TOPOLOGI
SEPIC PADA PROTOTIPE PEMBANGKIT LISTRIK TENAGA SURYA
DILENGKAPI SISTEM MONITORING ARUS DAN TEGANGAN BERBASIS
INTERNET OF THINGS DI BANDAR UDARA TANJUNG REDEP
KALIMARAU BERAU

Oleh :

ULFIANA DYAH PRAMESTI
NIT. 30121024

Disetujui untuk diujikan pada :
Surabaya, 9 Agustus 2024

Dosen Pembimbing 1 : YUDHIS THIRO KABUL YUNIOR, S.T., M.Kom.
NIP. 19870224 202203 1 003



Dosen Pembimbing 2 : Dr. Gunawan Sakti, S.T, M.T.
NIP. 19881001 200912 1 003



LEMBAR PENGESAHAN

IMPLEMENTASI MODIFIKASI BOOST KONVENTER DC-DC TOPOLOGI
SEPIC PADA PROTOTIPE PEMBANGKIT LISTRIK TENAGA SURYA
DILENGKAPI SISTEM MONITORING ARUS DAN TEGANGAN BERBASIS
INTERNET OF THINGS DI BANDAR UDARA TANJUNG REDEP
KALIMARAU BERAU

Oleh :

ULFIANA DYAH PRAMESTI
NIT. 30121024

Telah dipertahankan dan dinyatakan lulus Ujian Proyek Akhir
Program Pendidikan Diploma 3 Teknik Listrik Bandar Udara
Politeknik Penerbangan Surabaya
Pada tanggal : 9 Agustus 2024

Panitia Penguji :

Ketua : Drs. Hartono, S.T, M.Pd., M.M.
NIP. 19630408 198902 1 001

Sekretaris : Dr. Gunawan Sakti, S.T., M.T.
NIP. 19881001 200912 1 003

Anggota : YUDHIS THIRO KABUL YUNIOR, S.T., M.Kom.
NIP. 19870224 202203 1 003

Ketua Program Studi
D3 Teknik Listrik Bandara

Dr. Gunawan Sakti, S.T., M.T.
NIP. 19881001 200912 1 003

ABSTRAK

IMPLEMENTASI MODIFIKASI BOOST KONVENTER DC-DC TOPOLOGI
SEPIC PADA PROTOTIPE PEMBANGKIT LISTRIK TENAGA SURYA
DILENGKAPI SISTEM MONITORING ARUS DAN TEGANGAN BERBASIS
INTERNET OF THINGS DI BANDAR UDARA TANJUNG REDEP
KALIMARAU BERAU

Oleh:
ULFIANA DYAH PRAMESTI
NIT. 30121024

Perancangan sistem monitoring PLTS yang terhubung pada beban, diperlukan beberapa komponen salah satunya konverter DC-DC. Konverter ini digunakan untuk meningkatkan tegangan keluaran panel surya. Dalam perancangan konverter DC-DC, biasanya digunakan konverter dengan *gain* yang tinggi untuk meningkatkan tegangan keluaran *photovoltaic* dan memperoleh efisiensi konversi yang tinggi. Pada penelitian ini dilakukan modifikasi pada boost konverter konvensional. Modifikasi dilakukan dengan penambahan diode dan kapasitor sebagai rangkaian pengganda tegangan. Hasil dari modifikasi dapat menaikkan tegangan 8 kali tegangan masukan.

Modifikasi *boost* konverter DC-DC Topologi SEPIC diimplementasikan di Bandar Udara Tanjung Redep Kalimantan Berau. Pada bandara tersebut memiliki PLTS berkapasitas 540 KWP. Monitoring yang dilakukan pada PLTS bandara ini masih dilakukan secara manual yaitu dengan melihat *display* monitor yang terletak pada inverternya. Hal ini membutuhkan beberapa waktu untuk melihatnya. Oleh karena itu penerapan monitoring dan modifikasi boost konverter DC-DC topologi SEPIC mempermudah dalam pemantauan PLTS dan dapat memaksimalkan *gain* yang tinggi sebelum masuk pada konverter.

Kata kunci : Pembangkit Listrik tenaga surya, monitoring arus dan tegangan, konverter DC-DC topologi SEPIC

ABSTRACT

IMPLEMENTATION OF SEPIC TOPOLOGY DC-DC BOOST CONVERTER MODIFICATION ON A SOLAR POWER PLANT PROTOTYPE EQUIPPED WITH AN INTERNET OF THINGS BASED CURRENT AND VOLTAGE MONITORING SISTEM AT TANJUNG REDEP AIRPORT, KALIMARAU BERAU

By:
ULFIANA DYAH PRAMESTI
NIT. 30121024

To design a PLTS monitoring system that is connected to a load, several components are required, one of which is a DC-DC converter. This converter is used to increase the output voltage of solar panels. In designing DC-DC converters, converters with high gain are usually used to increase the photovoltaic output voltage and obtain high conversion efficiency. In this research, modifications were made to the conventional boost converter. Modifications are made by adding diodes and capacitors as a voltage doubler circuit. The results of the modification can increase the voltage 8 times the input voltage. In designing a grid-connected PLTS monitoring sistem, a DC-DC converter is needed to increase the output voltage of the solar panels. When designing a DC-DC converter, a high gain converter is usually used to increase the photovoltaic output voltage and obtain high conversion efficiency. Therefore, in this research, modifications were made to improve the conventional converter. Modifications are made by adding diodes and capacitors. The results of the modification can increase the voltage up to 7 times the input voltage.

Modification of the DC-DC boost converter SEPIC topology was implemented at Tanjung Redep Airport, Kalimantan Berau. The airport has a PLTS with a capacity of 540 KWP. Monitoring carried out at this airport PLTS is still done manually, namely by looking at the monitor display located on the inverter. This takes some time to see. Therefore, implementing monitoring and modifying the SEPIC topology DC-DC boost converter makes it easier to monitor PLTS and can maximize high gain before entering the converter.

Keywords : Solar power generation, current and voltage monitoring, SEPIC topology DC-DC converter

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama	: Ulfiana Dyah Pramesti
NIT	: 30121024
Program Studi	: D3 Teknik Listrik Bandara
Judul Proyek Akhir	Implementasi Boost Konverter DC-DC Topologi Sepic Pada Prototipe Pembangkit Listrik Tenaga Surya Dilengkapi Sistem Monitoring Arus dan Tegangan Berbasis Internet of Things di Bandar Udara Tanjung Redep Kalimantan Berau

dengan ini menyatakan bahwa :

1. Proyek akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun di Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti *Non Eksklusif* (*Non-Exclusive Royalty-Free Right*) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*Database*), merawat, dan mempublikasikan Proyek akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya, 20 Agustus 2024
Yang membuat pernyataan



ULFIANA DYAH PRAMESTI
NIT. 30121024

MOTTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya dan hanya Allah lah yang maha tau diletakkan semuanya kepada pundak-pundak yang Ia mau dan mampu”

-Percaya dan ingat Allah selalu membersamai-

- Q.S Al Baqarah : 286 -



KATA PENGANTAR

Puji syukur kami panjatkan kepada kehadiran Allah SWT, berkat limpahan rahmat, kesehatan dan hidayahNya, sehingga penulis dapat menyusun Proposal Tugas Akhir yang berjudul “IMPLEMENTASI MODIFIKASI BOOST KONVERTER DC-DC TOPOLOGI SEPIC PADA PROTOTIPE PEMBANGKIT LISTRIK TENAGA SURYA DILENGKAPI SISTEM MONITORING ARUS DAN TEGANGAN BERBASIS *INTERNET OF THINGS* DI BANDAR UDARA TANJUNG REDEP KALIMARAU BERAU” dapat diselesaikan dengan sesuai waktu yang telah ditentukan dan diharapkan proyek akhir ini dapat bermanfaat untuk kemajuan teknologi yang ada sekarang.

Pada Penyusunan Proposal Tugas Akhir ini dimaksudkan sebagai syarat untuk menyelesaikan pendidikan bagi taruna program Diploma 3 di Politeknik Penerbangan Surabaya sehingga dapat memperoleh gelar Ahli Madya (A.Md).

Pada penyusunan tugas kali ini penulis mendapat bantuan doa, support, dan dukungan. Maka dari hal itu ucapan terima kasih penulis sampaikan kepada segenap pihak yang telah membantu selama proses penyusunan proyek akhir ini, terutama kepada:

1. Tuhan Yang Maha Esa yang telah memberi karunianya sehingga penulis bisa menyelesaikan penyusunan Proyek Akhir.
2. Kedua orang tua dan adik, yang senantiasa memberikan doa, kasih sayang, serta dukungan penuh baik berupa moril maupun materi
3. Bapak Ahmad Bahrawi selaku Direktur Politeknik Penerbangan Surabaya.
4. Bapak Dr. Gunawan Sakti, S.T., M.T. selaku Ketua Program Studi D3 Teknik Listrik Bandara.
5. Bapak Yudhis Thiro Kabul Yunior selaku dosen pembimbing 1 yang senantiasa membimbing dalam penyusunan Tugas Akhir.
6. Bapak Dr. Gunawan Sakti, ST., M.T. selaku pembimbing 2 yang senantiasa membimbing dalam penyusunan Tugas Akhir.
7. Seluruh dosen dan civitas akademika Prodi D3 Teknik Listrik Bandara Politeknik Penerbangan Surabaya
8. Teman-teman seangkatan TLB XVI dan teman-teman atas dukungan yang telah diberikan.

Pada hal ini penulis tentunya menyadari bahwa karya tulis ini masih jauh dari kata sempurna, maka dari itu saran, kritik, dan masukan yang membangun penting bagi penulis demi karya yang lebih baik di masa mendatang. Atas segala kesalahan dan kata – kata yang kurang berkenan, penulis memohon maaf dari lubuk hati yang paling dalam.

Surabaya, 1 Agustus 2024
Penyusun

Ulfiana Dyah Pramesti

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	iv
ABSTRACT	v
PERNYATAAN KEASLIAN DAN HAK CIPTA.....	vi
MOTTO	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Sistematika Penulisan	5
BAB 2 TINJAUAN TEORI.....	6
2.1. Pembangkit Listrik Tenaga Surya (PLTS).....	6
2.1.1. Panel Surya	7
2.2. Konverter DC-DC Topologi SEPIC Konvensional	7
2.3. Modifikasi Konverter DC-DC Topologi SEPIC.....	11
2.3.1. Mode Pengoperasian ketika Saklar Q Tertutup (t_{on})	12
2.3.2. Mode Pengoperasian Ketika Saklar Q Terbuka (t_{off})	13
2.4. Penurunan Parameter Komponen	13
2.5. Pembangkit Listrik Tenaga Surya.....	14
2.6. Mikrokontroler/Arduino UNO.....	15
2.7. Driver Mosfet.....	16

2.8.	ESP8266.....	17
2.9.	Comporator	18
2.10.	Ramp Generator	18
2.11.	Driver L298N.....	18
2.12.	<i>Pulse Width Modulation (PWM)</i>	19
2.13.	Sensor INA219.....	20
2.14.	Beban	21
2.15.	Kajian Penelitian Terdahulu yang Relevan.....	21
BAB 3	METODE PENELITIAN.....	25
3.1.	Desain Penelitian	25
3.2.	Perencanaan Alat	27
3.2.1.	Desain Alat.....	27
3.2.2.	Cara Kerja	29
3.2.3.	Spesifikasi Panel surya.....	30
3.2.4.	Spesifikasi Alat Ukur	30
3.2.5.	Sensor Arus INA219	31
3.2.6.	Perancangan Sensor Tegangan.....	32
3.2.7.	Perancangan Modifikasi Konverter Topologi Sepic	33
3.2.8.	<i>Firebase</i>	37
3.2.9.	Internet of Things (IoT)	38
3.3.	Teknik Pengujian	39
3.4.	Teknik Analisis Data.....	40
3.5.	Waktu Dan Pelaksanaan Penelitian	41
BAB 4		42
HASIL DAN PEMBAHASAN		42
4.1.	Uraian Rencana Penelitian	42
4.2.	Pengujian Validasi Alat Ukur pada Panel Surya	42
4.3.	Pengujian Mikrokontroler	46
4.4.	Pengujian Wemos D1 mini	47
4.5.	Pengujian Buck Konverter	49
4.6.	Pengujian Sensor Arus dan Tegangan	51
4.7.	Pengujian Boost Konverter	53

4.8. Pengujian Daya Input dan Output.....	59
4.9. Pengujian Intensitas pada <i>Duty cycle</i>	63
4.10. Pengujian Perbedaan Arus dan Tegangan Pada LCD, Avometer, dan Monitoring.....	66
BAB 5	70
KESIMPULAN.....	70
DAFTAR PUSTAKA.....	71
LAMPIRAN.....	A



DAFTAR GAMBAR

Gambar 2. 1 Rangkaian Konverter DC-DC topologi SEPIC konvensional.....	8
Gambar 2. 2 Mode Saklar Tertutup pada SEPIC Konverter	8
Gambar 2. 3 Mode Saklar Terbuka pada SEPIC Konverter	9
Gambar 2. 4 Duty cycle	10
Gambar 2. 5 Rangkaian Modifikasi Konverter DC-DC Topologi SEPIC	11
Gambar 2. 6 Mode Operasi saat Saklar Q Tertutup	12
Gambar 2. 7 Mode Operasi saat Saklar Q Terbuka	13
Gambar 2. 8 fotovoltaik	15
Gambar 2. 9 Atmega 16	16
Gambar 2. 10 Driver Mosfet	17
Gambar 2. 11 ESP8266	18
Gambar 2. 12 L298N Driver	19
Gambar 2. 13 Pulse Width Modulation (PWM)	20
Gambar 2. 14 Sensor INA219	20
Gambar 3. 1 Flowchart Desain Penelitian	25
Gambar 3. 2 Blok Diagram	27
Gambar 3. 3 Flowchart Cara kerja	29
Gambar 3. 4 Sensor INA219	31
Gambar 3. 5 Rangkaian Sensor Tegangan	32
Gambar 3. 6 Modifikasi Boost Konverter	33
Gambar 3. 7 Dioda MUR 1660 CT	36
Gambar 3. 8 MOSFET IRFP 4332	36
Gambar 3. 9 <i>Firestore</i>	38
Gambar 3. 10 Arsitektur Internet of Things	39
Gambar 4. 1 Diagram pengujian panel surya	43
Gambar 4. 2 Pengujian panel surya	43
Gambar 4. 3 Skematik Rangkaian Arduino Nano	46
Gambar 4. 4 Diagram pengujian mikrokontroler	46
Gambar 4. 5 Mikrokontroler	47
Gambar 4. 6 Skematik Wemos D1 Mini	48
Gambar 4. 7 Gambar 4. 11 Diagram Pengujian Wemos D1 mini	48
Gambar 4. 8 Monitoring	48
Gambar 4. 9 Skematik Buck Konverter	49
Gambar 4. 10 Diagram pengujian buck konverter	49
Gambar 4. 11 Buck konverter	50
Gambar 4. 12 Skematik Sensor Arus ACS	51
Gambar 4. 13 Diagram Pengujian Sensor	51
Gambar 4. 14 Sensor Arus dan Tegangan	52
Gambar 4. 15 Rangkaian skematik boost konverter Topologi SEPIC	53

Gambar 4. 16 Diagram pengujian boost konverter Topologi SEPIC.....	53
Gambar 4. 17 Modifikasi Boost Konverter Topologi SEPIC	54
Gambar 4. 18 Rangkaian Boost Konverter Konvensional	56
Gambar 4. 19 Diagram pengujian boost konverter konvensional.....	56
Gambar 4. 20 Boost konverter konvensional	56
Gambar 4. 21 Pengujian boost konverter konvensional.....	57
Gambar 4. 22 Grafik perbandingan tegangan	58
Gambar 4. 23 Grafik perbandingan arus	58
Gambar 4. 24 Pengujian daya input dan output	59
Gambar 4. 25 Grafik input dan output daya.....	62
Gambar 4. 26 Duty Cycle 10%	66
Gambar 4. 27 Diagram Pengujian Perbedaan Arus dan Tegangan.....	67
Gambar 4. 28 Grafik Tegangan.....	68



DAFTAR TABEL

Tabel 3. 1 Parameter awal Desain Konverter.....	33
Tabel 3. 2 Komponen.....	37
Tabel 3. 3 Jadwal Kegiatan	41
Tabel 4. 1 Hasil Pengujian Panel Surya	44
Tabel 4. 2 Pengujian buck konverter.....	50
Tabel 4. 3 Pengujian Sensor Arus dan Tegangan	52
Tabel 4. 4 Pengujian boost konverter.....	54
Tabel 4. 5 Boost konverter konvensional.....	57
Tabel 4. 6 Pengujian duty cycle	64
Tabel 4. 7 Pengujian perbedaan arus dan tegangan lcd dan avometer.....	67



DAFTAR LAMPIRAN

LAMPIRAN A Standart Operational Procedure (SOP).....	A-1
LAMPIRAN B Koding Alat	B-1
LAMPIRAN C Daftar Riwayat Hidup.....	C-1



DAFTAR PUSTAKA

- Anisya Sonita, R. F. (2018). Aplikasi E-Order Menggunakan Fire Base dan Algoritme Nuthmorris Pratt berbasis Android. *Pseudocode*.
- Arianto, S. D. (2021). Rangkaian Resistor dengan Induktor. *Jurnal Teknik Elektro Repoteknologi.id*.
- Aswandi Alfaris, M. Y. (2020). Sitem Kendali Dan Monitoring Boost Converter Berbasis GUI (Graphical User Interface) Matlab Menggunakan Arduino. *Jurnal Teknik Elektro Indonesia*.
- Candra Cahaya Utama, T. S. (2021). Implementasi Teknik Counter Pada Air Mancur untuk Membuat Animasih Air Berbasis Mikrokontroler ATMEGA 16. *Jurnal Teknologi Komputer dan Sistem Informasih*, 13-18.
- Dani Sasmoko, Y. A. (2017). Implementasi Penerapan Internet of Things (IoT) pada monitoring Infus Menggunakan ESP8266 dan Web untuk Berbagi data. *Manajemen Informatika dan Sistem Komputer, Sekolah Tinggi Elektronika dan Komputer Semarang*, 2(1).
- Deni Almanda, N. M. (2019). Studi Analisa Penyebab Kerusakan Kapasitor Bank Sub Station Welding di PT. Astra Daihatsu Motor. *Jurnal Teknik Elektro Universitas Muhammadiyah Jakarta*, 2(1).
- Devha P Sinaga, e. S. (2016). rancang Bangun Kestabilan Posisi Sistem Kendali Manual Robot Kapal Selam Menggunakan Metode Fuzzy Logic. *Jurnal Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom*, 3(1).
- Dody Susilo, B. F. (2023). Alat Penghitung Bibit Ikan Lele Berbasis Mikrokontroler ATmega 8535. *Jurnal Elektro Universitas PGRI Madiun*, 3(2).
- DT Valent, B. H. (2016). *Essential Matlab For Engineers And Scientists*. Academic Press.
- Faisal Irsan Pasaribu, M. R. (2021). Rancang Bangun Charging Station Berbasis Arduino. *Jurnal Teknik elektro, Rekayasa Elektrikal dan Energi Universitas Muhammadiyah Sumatera Utara*.
- Gu, B. (2013). High Boost Ratio Hybrid Transformer DC–DC Converter for Photovoltaic Module Applications. *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 28(4), 2048-2058.
- Gules, R. (2014). A Modified SEPIC Converter With High Static Gain. *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 29(11), 5860-5871.

- Gules, R. (2014). A Modified Sepic Converter with Hight Static Gain. *IEE Transactionson Power Electronics*, 29(11), 5860-5871.
- Gules, Roger. (2014). A Modified SEPIC Converter With High Static Gain. *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 29(11), 5860-5871.
- Harahap, P. (2020). Pengaruh Temperatur Permukaan Panel Surya Terhadap Daya Yang Dihasilkan Dari Berbagai Jenis Sel Surya. *Rekayasa elektrikal dan energi, Jurnal Teknik Elektro, Universitas Muhammadiyah Sumatera Utara (UMSU)*.
- Haryoga Nur Hermala, S. R. (2021). Desain dan Implementasi Kendali Digital Histeresis pada Topologi SEPIC Buck-Boost Konverter. *Riset dan Teknologi Terapan (RITEKTRA), Teknik Elektro, Fakultas Teknik, Universitas Katolik Soegijaptanata*.
- Humaidillah Kurniadi Wardana, M. I. (2022). Sistem Monitoring Arus dan Tegangan pada Pengisian Baterai untuk Pompa Air Tenaga Surya Berbasis IoT menggunakan Tinger.io. *LPPM UNHAS YEBUIRENG JOMBANG*.
- Imran Oktariawan, M. d. (2013). Pembuatan Sistem Otomasi Dispenser Menggunakan Mikrokontroler Arduiono Mega 2560. *Jurnal FEMA, Fakultas Teknik Universitas Lampung*, 1(2).
- Indra Dilianto, H. S. (2018). Desain dan Implementasi Konverter DC-DC Topologi SEPIC Termodifikasi dengan Gain yang Tinggi untuk Aplikasi pada Photovoltaic. *Jurnal Teknologi Penerbangan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS)*.
- J. Jiang, T. H. (2005). Maximum Power Tracking for Photovoltaic Power Sistems. *Tamkang J. Sci.* , 8(2), 147-153.
- Jonathan Lambert, R. M. (2021). Power consumption profiling of a lightweight development board: Sensing with the INA219 and Teensy 4.0 microcontroller. *Electronics*, 10(7).
- Jung, D.-Y. (2010). Interleaved Soft-Switching Boost Converter for Photovoltaic Power-Generation Sistem. *IEEE Transactions on Power Electronics*, 26(4), 1137-1145.
- Mehta, M. (2015). A Breakthrough In Wireless Sensor Networks And Internet Of Things. *nternational Journal of Electronics and Communication Engineering & Technology* .
- Milfiga Septa Yoski, R. M. (2020). Prototipe ROBOT Pembersih Lantai BERbasis Mikrokontroler dengan Sensor Ultrasonik. *Jurnal Teknik Elektro Indonesia*, 1(2).

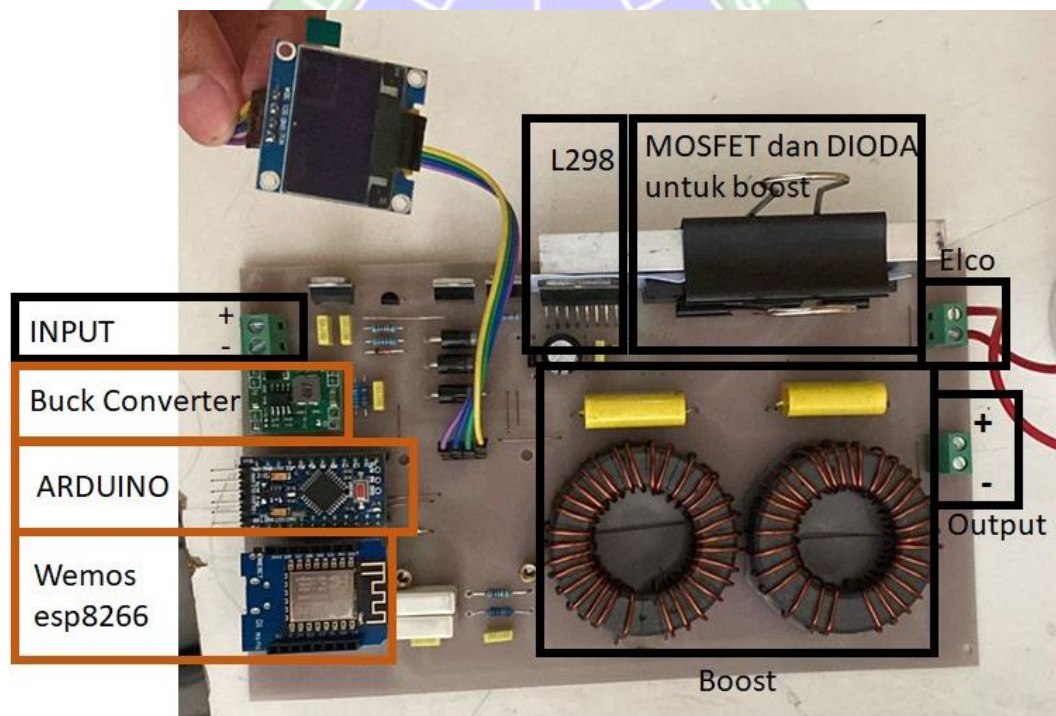
- Muhammad Irfan Dzakwan, I. S. (2018). PERANCANGAN KONVERTER ARUS SEARAH TIPE PENURUN TEGANGAN DENGAN MOSFET SINKRON DAN TANPA MOSFET SINKRON. *Jurnal Ilmiah Teknik elektro, Universitas Diponegoro*.
- Novan Enardo Wijanarko, S. P. (2021). Rancang Bangun Sistem Alat Praktikum MOSFET di Laboratorium ELEktronika Daya. *Jurnal Teknik Elektro Politeknik Negeri Samarinda*, 02(02).
- Steven J. Sokop, D. J. (2016). Trainer Antarmuka Periferal Berbasis Mikrokontroler Arduino Uno. *Jurnal Teknik Elektro dan Komputer UNSRAT*.
- Sukandar Sawidin, Y. R. (2021). Kontrol dan Monitoring Sistem Smart Home Menggunakan Web Thinger.io Berbasis IoT. *Jurnal Teknik Elektro Politeknik Negeri Manado*.
- Zulkarnain, G. R. (2016). *Desain dan Implementasi Konverter DC-DC Rasio Tinggi Berbasis PenSaklaran Kapasitor dan Induktor Terkopel untuk Aplikasi pada Photovoltaic*. Surabaya: Teknik Elektro ITS.



LAMPIRAN

LAMPIRAN A : *Standart Operational Procedure (SOP)*

Standart Operational Procedure (SOP) pada pengoperasian alat merupakan panduan mengenai Langkah-langkah dalam penggunaan alat yang sesuai dan aman. SOP ini bertujuan untuk memberikan panduan yang jelas dan terstruktur tentang tata cara penggunaan “Implementasi Modifikasi Boost Konverter DC-DC Topologi Sepic pada Prototipe Pembangkit Listrik Tenaga Surya Dilengkapi Sistem Monitoring Arus dan Tegangan Berbasis *Internet of Things* di Bandar Udara Tanjung Redep Kalimantan Berau”. Dengan adanya SOP ini diharapkan lebih aman dalam pengoperasian alat ini sehingga dapat alat dapat berfungsi secara optimal, tidak ada kesalahan dan kerusakan pada alat.



A. Langkah-langkah menyalakan alat

1. Meletakkan alat pada bawah sinar matahari sehingga penyerapan sinar matahari oleh panel surya akan optimal.

2. Menyambungkan output rangkaian dengan beban lampu DC yang tersedia. Penyambungan dapat dilihat pada gambar diatas dan memperhatikan positif dan negatif jangan sampai terbalik.
3. Rangkaian tidak boleh dihidupkan tanpa beban oleh karena itu memastikan beban tersambung secara sempurna. Apabila terpasang tanpa beban maka akan merusak komponen bahkan alat bisa mati.
4. Setelah beban terpasang, menyambungkan output panel surya ke input rangkaian. (keterangan pada gambar diatas)
5. Alat akan menyala, lcd menampilkan tegangan arus input maupun output.
6. Selanjutnya menyalakan *hotspot* yang sudah disetting untuk memonitoring arus dan tegangan pada web server maupun aplikasi android yang telah dibuat.
7. Tampilan nilai tegangan dan arus tertampil pada monitoring web dan aplikasi android, hal ini menandakan alat sudah menyala dan dapat digunakan.

B. Langkah-langkah mematikan alat

1. Mematikan hotspot yang terhubung untuk memutuskan koneksi dengan web dan aplikasi android.
2. Melepas sambungan output solar cell pada input rangkaian.
3. Alat sudah mati

LAMPIRAN B: Koding Alat

1. Modifikasi Boost konverter_Streamer

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>
#include <WiFiClient.h>
#include <SoftwareSerial.h>

#define SSER_TX_Pin D1
#define SSER_RX_Pin D2

// software serial #1: RX = digital pin 10, TX = digital pin 11
SoftwareSerial SoftSerial(SSER_RX_Pin, SSER_TX_Pin);

const char* ssid = "createme.id";
const char* password = "ccaaddok";
//=> '/api/data/{vin}/{vout}/{iin}/{iout}/{freq}/{duty}'
//https://solarbooster.cloud/api/data/1.2/1.3/1.4/1.5/100/200
const String urlBase = "https://solarbooster.cloud/api/data";
String url = "";

unsigned long time_now = 0;
const long streamPeriod = 2000;

String dataSerial;

void constructURL(void);
void sendData(void);
void streamData(void);
void checkSoftSerial(void);
```



```

WiFiClient client;

// constants won't change. Used here to set a pin number:
const int ledPin = LED_BUILTIN; // the number of the LED pin

// Variables will change:
int ledState = LOW; // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0; // will store last time LED was updated

// constants won't change:
const long interval = 500; // interval at which to blink (milliseconds)

//
bool uploadState = false;

void setup() {
  // put your setup code here, to run once:
  // put your setup code here, to run once:
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  Serial.println("Hello Streamer");
  delay(100);

  // set the data rate for the SoftwareSerial port
  SoftSerial.begin(115200);
  // SoftSerial.println("Hello Master");
  Serial.println("Init SoftSerial One Success...");
  delay(100);
}

```

```

WiFi.begin(ssid, password, 6);
Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
}
Serial.println("");
Serial.print("Connected! IP address: ");
Serial.println(WiFi.localIP());

pinMode(ledPin, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    streamData();
    checkSoftSerial();
    blink();
    delay(100);
}

void sendData() {
    if (client.connect("solartracker.cloud", 80)) {
        client.print("GET ");
        client.print(url);
        client.println(" HTTP/1.1");
        client.print("Host: ");
        client.println("solartracker.cloud");
        client.println("Connection: close");
        client.println();
        client.stop();
    }
}

```

```

        Serial.println("Send.");
    } else {
        Serial.println("Connection failed.");
    }
}

void constructURL() {
    // Construct the URL based on the variables
    url = urlBase + dataSerial;
    Serial.print("url: ");
    Serial.print(url);
}

void checkSoftSerial(){
    String data="";
    bool serialState = false;
    if (SoftSerial.available() > 0) {

        data = SoftSerial.readString();
        // dataSerial = data;

        serialState = true;

        // Serial.print("serial: ");
        // Serial.println(dataSerial);
    }
    if (Serial.available() > 0) {

        data = Serial.readString();
        // dataSerial = data;
    }
}

```

```

    serialState = true;

    // Serial.print("serial: ");
    // Serial.println(dataSerial);
}

if(serialState == true){
    serialState = false;

    dataSerial = data;
    Serial.print(dataSerial);

    uploadState = true;
}
}

void streamData(){
    if (((millis() - time_now) >= streamPeriod) && uploadState) {
        time_now = millis();

        // Reconstruct URL with updated values
        constructURL();
        // Send data
        sendData();

        uploadState = false;
    }
}

void blink(){
    // check to see if it's time to blink the LED; that is, if the difference

```

```
// between the current time and last time you blinked the LED is bigger  
than
```

```
// the interval at which you want to blink the LED.
```

```
unsigned long currentMillis = millis();
```

```
if (currentMillis - previousMillis >= interval) {
```

```
    // save the last time you blinked the LED
```

```
    previousMillis = currentMillis;
```

```
    // if the LED is off turn it on and vice-versa:
```

```
    if (ledState == LOW) {
```

```
        ledState = HIGH;
```

```
    } else {
```

```
        ledState = LOW;
```

```
    }
```

```
    // set the LED with the ledState of the variable:
```

```
    digitalWrite(ledPin, ledState);
```

```
}
```

```
}
```

2. Master

```
#include <Arduino.h>
#include <Wire.h>
#include <U8x8lib.h>
#include <Adafruit_INA219.h>
#include <SoftwareSerial.h>
#include <ACS712.h>

#define _PWM_LOGLEVEL_ 4

#include <AVR_PWM.h>

#define SSER_PONE_TX_Pin 5
#define SSER_PONE_RX_Pin 4
#define SSER_PTWO_TX_Pin 11
#define SSER_PTWO_RX_Pin 10
// #define MOSFET_Pin 9 // Timer1A on UNO, Nano, etc
#define I2C_SCL_Pin A5
#define I2C_SDA_Pin A4
#define CURENT_BOOST_Pin A1
#define VOLTAGE_BOOST_Pin A0

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
// Configure orientation of the display.
// 0 = none, 1 = 90 degrees clockwise, 2 = 180 degrees, 3 = 270 degrees
CW
#define ROTATION 0
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset
pin)
```

```
#define SCREEN_ADDRESS 0x3c ///< See datasheet for Address; 0x3D
for 128x64, 0x3C for 128x32
```

```
SoftwareSerial SSerial_portOne(SSER_PONE_RX_Pin,
SSER_PONE_TX_Pin);
// software serial #1: RX = digital pin 10, TX = digital pin 11
SoftwareSerial SSerial_portTwo(SSER_PTWO_RX_Pin,
SSER_PTWO_TX_Pin);
```

```
Adafruit_INA219 ina219;
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/
U8X8_PIN_NONE);
// Arduino UNO has 5.0 volt with a max ADC value of 1023 steps
// ACS712 5A uses 185 mV per A
// ACS712 20A uses 100 mV per A
// ACS712 30A uses 66 mV per A
ACS712 ACS(CURRENT_BOOST_Pin, 5.0, 1023, 185);
```

```
//creates pwm instance
AVR_PWM* PWM_Instance;
```

```
int setPoint_boostVoltage = 110;
int max_dutyCycle = 85;
int min_dutyCycle = 20;
```

```
float frequency = 33000;
int dutyCycle = min_dutyCycle;
float last_dutyCycle = dutyCycle;
```

```
int period = 3000;
unsigned long time_now = 0;
```



```
unsigned long time_dutyCycle = 0;
int dutyCycle_updatePeriod = 1000;
```

```
int sourceVoltage_mV = 0;
int sourceCurrent_mA = 0;
int sourcePower_mW = 0;
int boostVoltage_V = 0;
int boostCurrent_mA = 0;
```

```
char dashLine[]
="=====
=====";
```

```
char startChar = "#";
char stopChar = ";";
```

```
void setup() {
    // put your setup code here, to run once:
    // Open serial communications and wait for port to open:
    Serial.begin(115200);
    Serial.println("Hello Master");

    // set the data rate for the SoftwareSerial port
    SSerial_portOne.begin(115200);
    SSerial_portOne.println("Hello Streamer");
    Serial.println("Init SoftSerial One Success...");
    delay(100);
    SSerial_portTwo.begin(115200);
    SSerial_portTwo.println("Hello Driver");
    Serial.println("Init SoftSerial Two Success...");
    delay(100);
}
```

```

// Setup the OLED display.
delay(250); // wait for the OLED to power up
u8x8.begin();
u8x8.setPowerSave(0);
u8x8.setFont(u8x8_font_chroma48medium8_r);
Serial.println("Init OLED Display Success...");

// Initialize the INA219.
// By default the initialization will use the largest range (32V,
2A). However
// you can call a setCalibration function to change this range (see
comments).
if (! ina219.begin()) {
    Serial.println("Failed to find INA219 chip");
    while (1) { delay(10); }
}
// To use a slightly lower 32V, 1A range (higher precision on amps):
//ina219.setCalibration_32V_1A();
// Or to use a lower 16V, 400mA range (higher precision on volts and
amps):
//ina219.setCalibration_16V_400mA();
Serial.println("Init INA219 Success...");

ACS.autoMidPoint();
// Serial.println(ACS.getMidPoint());
Serial.println("Init ACS712 Success...");
}

void loop() {
    // put your main code here, to run repeatedly:
    sourceVoltage_mV = ina219.getBusVoltage_V()*1000;

```

```

sourceCurrent_mA = ina219.getCurrent_mA();

boostVoltage_V = map(analogRead(VOLTAGE_BOOST_Pin), 0, 1023.0,
0, 135.0);
boostCurrent_mA = ACS.mA_DC();

if(millis() - time_dutyCycle >= dutyCycle_updatePeriod){
    time_dutyCycle = millis();

    if((setPoint_boostVoltage < boostVoltage_V) && ((boostVoltage_V -
setPoint_boostVoltage)>=2.0f)){
        dutyCycle--;
        if(dutyCycle <= min_dutyCycle)
            dutyCycle = min_dutyCycle;
    }

    if((setPoint_boostVoltage > boostVoltage_V) &&
((setPoint_boostVoltage - boostVoltage_V)>=2.0f)){
        dutyCycle++;
        if(dutyCycle >= max_dutyCycle)
            dutyCycle = max_dutyCycle;
    }

    if(last_dutyCycle != dutyCycle){
        last_dutyCycle = dutyCycle;

        char bufPrint[50];
        memset(bufPrint, 0, sizeof(bufPrint)); sprintf(bufPrint, "%d",
dutyCycle);

        String message = '#' + String(dutyCycle) + ';';
        SSerial_portTwo.listen();
        SSerial_portTwo.print(message);
    }
}

```

```

        Serial.print("Data to Driver: ");
        Serial.println(message);
    }
}

if(millis() - time_now >= period){
    time_now = millis();

    // Update display.
    char bufPrint[50];
    u8x8.clearDisplay();

    u8x8.drawString(3, 0, "MONITORING");

    memset(bufPrint, 0, sizeof(bufPrint)); sprintf(bufPrint,"SV %3d.%02d
V", sourceVoltage_mV/1000, (sourceVoltage_mV%100)/10);
    u8x8.drawString(2, 2, bufPrint);
    memset(bufPrint, 0, sizeof(bufPrint)); sprintf(bufPrint,"SC %3d.%02d
A", sourceCurrent_mA/1000, (sourceCurrent_mA%1000)/10);
    u8x8.drawString(2, 3, bufPrint);
    memset(bufPrint, 0, sizeof(bufPrint)); sprintf(bufPrint,"DC %3d %%",
dutyCycle);
    u8x8.drawString(2, 4, bufPrint);

    memset(bufPrint, 0, sizeof(bufPrint)); sprintf(bufPrint,"BV %3d V",
boostVoltage_V);
    u8x8.drawString(2, 6, bufPrint);
    memset(bufPrint, 0, sizeof(bufPrint)); sprintf(bufPrint,"BC %3d.%02d
A", boostCurrent_mA/1000, (boostCurrent_mA%1000)/10);
    u8x8.drawString(2, 7, bufPrint);

```

```

        Serial.print("Source Voltage: "); Serial.print(sourceVoltage_mV);
        Serial.println(" mV");

        Serial.print("Source Current: "); Serial.print(sourceCurrent_mA);
        Serial.println(" mA");

        Serial.print("Source Power: ");
        Serial.print(sourceVoltage_mV*sourceCurrent_mA/1000); Serial.println("
mW");

        Serial.print("Boosted Voltage: "); Serial.print(boostVoltage_V);
        Serial.println(" V");

        Serial.print("Boosted Current: "); Serial.print(boostCurrent_mA);
        Serial.println(" mA");

        memset(bufPrint, 0, sizeof(bufPrint));
        sprintf(bufPrint, "##d,%d,%d,%d,%d;", sourceVoltage_mV/10,
sourceCurrent_mA/10, dutyCycle, boostVoltage_V,
boostCurrent_mA/10);

        Serial.print("Data to Streamer: ");
        Serial.println(bufPrint);

        SSerial_portOne.listen();
        SSerial_portOne.println(bufPrint);
    }
}

void printPWMInfo(AVR_PWM* PWM_Instance)
{
    Serial.println(dashLine);
    Serial.print("Actual data: pin = ");
    Serial.print(PWM_Instance->getPin());
    Serial.print(", PWM DutyCycle = ");
    Serial.print(PWM_Instance->getActualDutyCycle());

```

```
Serial.print(", PWMPeriod = ");  
Serial.print(PWM_Instance->getPWMPeriod());  
Serial.print(", PWM Freq (Hz) = ");  
Serial.println(PWM_Instance->getActualFreq(), 4);  
Serial.println(dashLine);  
}
```



LAMPIRAN C Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



ULFIANA DYAH PRAMESTI, lahir di Kabupaten Magetan, pada tanggal 30 Oktober 2002. Anak pertama dari Bapak Hariyanto dan Ibu Sumiati. Memiliki 1 saudara kandung yaitu Alvin Dwi Bhektianto. Beragama Islam. Bertempat tinggal di Desa Karangmojo, Rt/Rw 08/03, Kecamatan Kartoharjo, Kabupaten Magetan. Pendidikan formal yang pernah diikuti sebagai berikut :

1. SDN Temenggungan 2 Magetan lulus tahun 2014
2. SMPN 1 Barat Magetan lulus tahun 2017
3. SMAN 1 Maospati Magetan lulus tahun 2020

Pada bulan September 2021 diterima sebagai Taruna di Politeknik Penerbangan Surabaya pada Program Studi Diploma III Teknik Listrik Bandara Angkatan XVI. Melaksanakan On The Job Training di Bandar Udara Kalimantan Berau pada bulan 6 Mei-22 September 2023 dan Bandar Udara Soekarno-Hatta Jakarta pada bulan 2 Oktober 2023-16 Februari 2024. Telah melaksanakan Proyek Akhir sebagai syarat kelulusan dalam Pendidikan di Politeknik Penerbangan Surabaya.