

**RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGA
DENGAN METODE *BACKPROPAGATION* BERBASIS
*INTERNET OF THINGS (IOT)***

PROYEK AKHIR



Oleh:

NASYAFIA AHSANUL AMALA
NIT. 30121019

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

2024

**RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGA
DENGAN METODE *BACKPROPAGATION* BERBASIS
*INTERNET OF THINGS (IOT)***

PROYEK AKHIR

Diajukan Sebagai Salah Satu Syarat untuk Mendapatkan Gelar Ahli Madya
(A.Md.) pada Program Studi Diploma 3 Teknik Listrik Bandara



Oleh:

NASYAFIA AHSANUL AMALA
NIT. 30121019

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

2024

LEMBAR PERSETUJUAN

RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGAH DENGAN
METODE *BACKPROPAGATION* BERBASIS *INTERNET OF THINGS* (IOT)

Oleh :

NASYAFIA AHSANUL AMALA

NIT. 30121019

Disetujui untuk Diujikan pada :

Surabaya, 08 Agustus 2024

Pembimbing I : Dr. KUSTORI, S.T, M.M.
NIP. 19590305 198503 1 002



Pembimbing II : Dr. SUDRAJAT, S.E,M.M.
NIP. 19600514 197912 1 001



LEMBAR PENGESAHAN

RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGAH DENGAN
METODE *BACKPROPAGATION* BERBASIS *INTERNET OF THINGS* (IOT)

Oleh :

NASYAFIA AHSANUL AMALA
NIT. 30121019

Telah Dipertahankan dan Dinyatakan Lulus pada Ujian Proyek Akhir
Program Studi Diploma 3 Teknik Listrik Bandar Udara
Politeknik Penerbangan Surabaya

Pada Tanggal : Surabaya, 08 Agustus 2024

Panitia Penguji

Ketua : Drs. HARTONO, S.T, M.Pd, M.M.
NIP. 19610727 198303 1 002

Sekretaris : Dr. SUDRAJAT, S.E,M.M.
NIP. 19600514 197912 1 001

Anggota : Dr. KUSTORI, S.T, M.M.
NIP. 19590305 198503 1 002

Mengetahui,
Ketua Program Studi
D3 Teknik Listrik Bandar Udara

Dr. GUNAWAN SAKTI, ST, MT.
NIP. 19881001 200912 1 003

LEMBAR PERSEMBAHAN

MOTTO

“ Allah tidak akan membebani seseorang, melainkan sesuai dengan kesanggupannya”.

(Q.S Al-Baqarah : 286)

“Nak, buktikan bahwa lahir dari keluarga yang sederhana, bisa mengangkat derajat kedua orangtua ya”.

-Ibu

PERSEMBAHAN :

Tuhan Yang Maha Esa dengan segala rahmat dan karunia-Nya. Kepada dua orang paling istimewa, Bapak Moh Firdaus dan Ibu Tatik Maulida atas didikannya, kerja keras serta kesabarannya yang selalu mengusahakan segalanya untuk putri satu satunya, terimakasih selalu memberikan cinta, motivasi, nasihat serta doa yang tak pernah putus. Terimakasih juga diucapkan kepada satu satunya saudara kandung saya, Mas Yusril Zulfan Firdaus yang sudah memberi banyak pengalaman hidup dan motivasi serta dukungannya.

Sebagai tanda bakti, hormat, dan rasa terima kasih yang tidak dapat diutarakan.

Saya persembahkan karya kecil ini kepada kalian orang- orang istimewa atas segala cinta dan kasih sayang yang telah diberikan, yang tidak mungkin dapat saya balas hanya dengan selembar kertas yang bertuliskan kata cinta dalam kata persembahan. Semoga ini menjadi langkah awal untuk membuat kalian bahagia.

Terimakasih

ABSTRAK

RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGA DENGAN METODE *BACKPROPAGATION* BERBASIS *INTERNET OF THINGS* (IOT)

Oleh :

NASYAFIA AHSANUL AMALA

NIT. 30121019

Meningkatnya kebutuhan listrik di Indonesia menyebabkan penurunan pasokan energi. Salah satu solusi adalah pemanfaatan energi baru terbarukan yang memiliki potensi cukup besar di Indonesia karena kondisi geografisnya yaitu Pembangkit Listrik Tenaga Surya (PLTS). Namun, efektivitas panel surya seringkali tidak maksimal karena pengaruh intensitas cahaya dan sudut kemiringan matahari, terutama pada panel yang ditempatkan statis. Untuk meningkatkan efisiensi penyerapan energi, diperlukan sistem yang dapat mengikuti pergerakan matahari yakni *solar tracker*.

Penelitian ini membahas pengaplikasian *Solar Tracker* sumbu tunggal menggunakan Metode *Backpropagation* berbasis IoT dengan panel surya 50WP sebagai sumber energi. Dalam perancangan sistem ini menggunakan sensor cahaya BH1750, mikrokontroler Arduino Mega, modul Wemos D1 Mini Pro untuk konektivitas IoT, dan motor *linear actuator* sebagai penggerak mekanisme *tracking*. Metode *backpropagation* digunakan untuk menyesuaikan posisi panel berdasarkan data intensitas cahaya, meningkatkan efisiensi penyerapan energi surya.

Hasil pengujian menunjukkan bahwa sistem *solar tracker* ini memiliki akurasi tinggi dalam mengikuti pergerakan matahari, yang secara signifikan meningkatkan efisiensi penyerapan energi dibandingkan dengan panel surya statis. Sistem ini berhasil menunjukkan peningkatan *output* energi yang dihasilkan. Integrasi dengan teknologi IoT memungkinkan pemantauan kondisi sistem secara *real-time*, mempermudah operasional dan pemeliharaan. *Solar tracker* ini terbukti sebagai solusi efektif untuk meningkatkan kinerja dan produktivitas sistem energi surya, serta memberikan kontribusi positif terhadap pengelolaan energi terbarukan.

Kata kunci: *Solar Tracker*, Pembangkit Listrik Tenaga Surya, Metode *Backpropagation*, IoT, Sensor BH1750.

ABSTRACT

DESIGN AND DEVELOPMENT OF A SINGLE-AXIS SOLAR TRACKER USING BACKPROPAGATION METHOD BASED ON INTERNET OF THINGS (IOT)

By :

NASYAFIA AHSANUL AMALA

NIT. 30121019

The increasing electricity demand in Indonesia has led to a decline in energy supply. One solution is to utilize renewable energy, which has significant potential in Indonesia due to its geographic conditions, such as Solar Power Plants (PLTS). However, the effectiveness of solar panels is often not optimal due to the influence of light intensity and the angle of the sun, especially for static panels. To enhance energy absorption efficiency, a system that can track the movement of the sun, known as a solar tracker, is needed.

This study discusses the application of a single-axis Solar Tracker using the Backpropagation Method based on IoT with a 50WP solar panel as the energy source. The system design employs a BH1750 light sensor, an Arduino Mega microcontroller, a Wemos D1 Mini Pro module for IoT connectivity, and a linear actuator motor as the tracking mechanism drive. The backpropagation method is used to adjust the panel position based on light intensity data, improving solar energy absorption efficiency.

The testing results show that this solar tracker system has high accuracy in following the sun's movement, which significantly increases energy absorption efficiency compared to static solar panels. The system successfully demonstrated an increase in the output energy produced. Integration with IoT technology allows for real-time monitoring of the system's condition, facilitating operation and maintenance. This solar tracker has proven to be an effective solution for enhancing the performance and productivity of solar energy systems and contributes positively to renewable energy management.

Keywords: *Solar Tracker, Solar Power Plants, Backpropagation Method, IoT, BH1750 Sensor.*

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Nasyafia Ahsanul Amala
NIT : 30121019
Program Studi : D3 Teknik Listrik Bandar Udara
Judul Tugas Akhir : Rancang Bangun *Solar Tracker* Sumbu Tunggal
Dengan Metode *Backpropagation* Berbasis *Internet Of Things* (IOT)

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, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan Proyek Akhir dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya dengan sebenarnya. Apabila di kemudian hari terdapat 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, 2 Agustus 2024
Yang membuat pernyataan



Nasyafia Ahsanul Amala
NIT.30121019

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas berkat rahmat, hidayah dan karunia-Nya penulis berhasil menyelesaikan Proyek Akhir yang berjudul “RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGAH DENGAN METODE *BACKPROPAGATION* BERBASIS *INTERNET OF THINGS (IOT)*” dengan baik dan tepat waktu. Penyusunan Proyek Akhir ini dimaksudkan sebagai salah satu syarat menyelesaikan program pendidikan Diploma 3 Teknik Listrik Bandara di Politeknik Penerbangan Surabaya untuk mendapatkan gelar Ahli Madya (A.Md.)

Dalam proses penyusunan Proyek Akhir ini tidak terlepas dari bantuan berbagai pihak baik secara spiritual, materi serta nasihat yang mendukung penulis untuk menyelesaikan penyusunan Proyek Akhir. Untuk itu penulis mengucapkan terima kasih kepada :

1. Kedua Orang Tua dan Keluarga yang selalu memberikan restu, doa, bantuan serta dukungan kepada penulis sehingga dapat menyelesaikan penyusunan Proyek Akhir dengan baik dan lancar.
2. Bapak Ahmad Bahrawi, S.E, M.T. Selaku Direktur Politeknik Penerbangan Surabaya.
3. Bapak Dr. Gunawan Sakti, ST, MT. Selaku Ketua Program Studi Teknik Listrik Bandara Politeknik Penerbangan Surabaya.
4. Bapak Dr. Kustori, S.T, M.M. Selaku Dosen Pembimbing I yang senantiasa membimbing dan membantu dalam perancangan alat.
5. Bapak Dr. Sudrajat, S.E, M.M. Selaku Dosen Pembimbing II yang senantiasa membimbing dan membantu dalam penyusunan Proyek Akhir.
6. Para Dosen dan Staff Program Studi Teknik Listrik Bandara Politeknik Penerbangan Surabaya yang tak henti-hentinya memberikan ilmu dan bimbingannya kepada penulis.
7. Rekan-rekan TLB XVI *Alpha* dan *Bravo* atas kebersamaan, kekompakkan dan kerjasamanya di segala kondisi. Senior dan Adik-adik TLB XVII atas dukungan serta doa yang diberikan kepada penulis.

Penulis menyadari bahwa penulisan ini masih jauh dari kata sempurna, maka dari itu penulis mengharapkan saran, kritik dan masukan yang membangun dari semua pihak agar dapat membantu penyusunan proyek akhir selanjutnya lebih baik. Akhir kata, penulis berharap semoga penulisan ini dapat bermanfaat dan selanjutnya dapat dikembangkan serta berguna bagi semua pihak.

Surabaya, 08 Agustus 2024



Nasyafia Ahsanul Amala
NIT.30121019

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERSEMBAHAN	iv
ABSTRAK	v
ABSTRACT	vi
PERNYATAAN KEASLIAN DAN HAK CIPTA	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	8
2.1 Teori Penunjang	8
2.1.1 Panel Surya	8
2.1.2 <i>Solar tracker</i>	10
2.1.3 Arduino Mega 2560	11
2.1.4 Sensor BH1750	12
2.1.5 Sensor INA219	12
2.1.6 Sensor IMU	13
2.1.7 Buck Converter MP1587	14
2.1.8 MPPT (Maximum Power Point Tracking)	14
2.1.9 RTC (<i>Real Time Clock</i>)	15

2.1.10 Modul Wemos D1 Pro	16
2.1.11 Motor Linear Actuator.....	17
2.1.12 BTS 7960 Driver Motor	17
2.1.13 Battery / Accu.....	18
2.1.14 Arduino IDE	19
2.1.15 Internet Of Things (IoT).....	20
2.1.16 Server Hosting.....	20
2.1.17 Metode <i>Backpropagation</i>	21
2.2 Kajian penelitian terdahulu yang relevan.....	22
BAB 3 METODE PENELITIAN	24
3.1 Desain Penelitian.....	24
3.2 Perancangan Alat.....	25
3.2.1 Desain Alat	25
3.2.2 Cara Kerja Alat.....	29
3.2.3 Komponen Alat	31
3.2.3.2 Perangkat Keras (Hardware).....	31
3.2.3.3 Perangkat Lunak (Software)	33
3.3 Teknik Pengujian.....	34
3.4 Teknik Analisa Data	35
3.5 Tempat dan Waktu Penelitian.....	36
BAB 4 HASIL DAN PEMBAHASAN	38
4.1 Hasil Penelitian	38
4.1.1 Perangkat Keras.....	39
4.1.1.1 Panel Surya	39
4.1.1.2 Arduino Mega 2560	40
4.1.1.3 Sensor BH1750	41
4.1.1.4 Sensor INA219	41
4.1.1.5 Sensor IMU.....	42
4.1.1.6 Buck Converter MP1587	43
4.1.1.7 MPPT (Maximum Power Point Tracking).....	43
4.1.1.8 RTC (Real Time Clock)	44
4.1.1.9 Modul Wemos D1 Mini Pro.....	44

4.1.1.10 Motor Linear Actuator	45
4.1.1.11 BTS7960 Driver Motor.....	46
4.1.1.12 Baterai / Aki.....	46
4.1.2 Perangkat Lunak.....	47
4.1.2.1 Aplikasi Arduino IDE (Integrated Development Environment)	47
4.1.2.2 Aplikasi Server Hosting.....	47
4.1.2.3 Algoritma Backpropagation.....	48
4.1.3 Sinkronisasi Perangkat Keras dan Aplikasi.....	50
4.2 Pembahasan Hasil Penelitian	50
4.2.1 Hasil Pengujian.....	51
4.2.1.1 Pengujian Rangkaian Buck Converter	51
4.2.1.2 Pengujian Sensor IMU	53
4.2.1.3 Pengujian Sensor INA219	53
4.2.1.4 Pengujian Wemos D1 Mini Pro	54
4.2.1.5 Pengujian Arduino Mega	55
4.2.1.6 Pengujian Real Time Clock (RTC)	55
4.2.1.7 Pengujian MPPT (Main Power Point Tracking)	56
4.2.1.8 Pengujian Baterai.....	56
4.2.1.9 Pengujian BTS7960 Driver Motor.....	57
4.2.1.10 Pengujian Panel Surya Keseluruhan.....	58
4.2.1.11 Pengujian Aplikasi Arduino IDE (Integrated Development Environment).....	62
4.2.1.12 Pengujian Internet of Things (IoT)	64
4.2.2 Kekurangan dan Kelebihan Alat	65
4.2.2.1 Kelebihan.....	65
4.2.2.2 Kekurangan.....	66
BAB 5 PENUTUP.....	67
5.1 Simpulan.....	67
5.2 Saran.....	68
DAFTAR PUSTAKA.....	69
LAMPIRAN.....	

DAFTAR GAMBAR

Gambar 2. 1 Jenis Panel Surya.....	9
Gambar 2. 2 Ecoflow <i>Solar tracker</i>	10
Gambar 2. 3 <i>Single Axis Solar tracker</i>	11
Gambar 2. 4 <i>Dual Axis Solar tracker</i>	11
Gambar 2. 5 Arduino Mega 2560.....	12
Gambar 2. 6 Sensor BH1750	12
Gambar 2. 7 Sensor INA219	13
Gambar 2. 8 Sensor IMU	14
Gambar 2. 9 Buck Converter MP1587.....	14
Gambar 2. 10 MPPT (Maximum Power Point Tracking)	14
Gambar 2. 11 RTC	16
Gambar 2. 12 Modul Wemos D1 Mini	16
Gambar 2. 13 Motor Linear Actuator.....	17
Gambar 2. 14 Driver Motor BTS 7960.....	18
Gambar 2. 15 Battery/Accu 12V.....	18
Gambar 2. 16 Arduino IDE	19
Gambar 2. 17 Internet Of Things (IoT).....	20
Gambar 2. 18 Web Hosting.....	20
Gambar 2. 19 Algoritma <i>Backpropagation</i>	21
Gambar 3. 1 Flowchart Desain Penelitian.....	24
Gambar 3. 2 Desain Block Diagram System	26
Gambar 3. 3 Design Main Controller System.....	27
Gambar 3. 4 Block Diagram Desain <i>Hosting Database</i>	28
Gambar 3. 5 Flowchart cara kerja alat	29
Gambar 4. 1 Tampilan Alat Secara Keseluruhan	38
Gambar 4. 2 Panel Surya 50WP.....	39
Gambar 4. 3 Rangkaian Arduino Mega2560.....	40
Gambar 4. 4 Rangkaian Sensor BH1750	41
Gambar 4. 5 Sensor BH1750	41
Gambar 4. 6 Rangkaian Sensor INA219.....	41
Gambar 4. 7 Modul Sensor INA219	42
Gambar 4. 8 Rangkaian Sensor IMU	42
Gambar 4. 9 Sensor IMU	42
Gambar 4. 10 Rangkaian Buck Converter MP1587.....	43
Gambar 4. 11 Buck Converter MP1587.....	43
Gambar 4. 12 MPPT (Maximum Power Point Tracking)	43
Gambar 4. 13 Rangkaian RTC (Real Time Clock)	44
Gambar 4. 14 Modul RTC (Real Time Clock).....	44
Gambar 4. 15 Rangkaian Wemos D1 Mini Pro.....	44
Gambar 4. 16 Modul Wemos D1 Mini Pro	45

Gambar 4. 17 Motor Linear Actuator.....	45
Gambar 4. 18 Rangkaian BTS7960 Driver Motor	46
Gambar 4. 19 BTS7960 Driver Motor	46
Gambar 4. 20 Baterai/Aki	46
Gambar 4. 21 Aplikasi Arduino IDE.....	47
Gambar 4. 22 Tampilan Aplikasi Solar Tracker	48
Gambar 4. 23 Data <i>Training Backpropagation</i>	48
Gambar 4. 24 Pengujian Tegangan Input <i>Buck Converter</i>	51
Gambar 4. 25 Pengujian Tegangan <i>output Buck Converter</i>	52
Gambar 4. 26 Pengujian Sensor IMU	53
Gambar 4. 27 Pengujian Wemos D1 Mini	54
Gambar 4. 28 Pengujian RTC	55
Gambar 4. 29 Pengujian MPPT	56
Gambar 4. 30 Pengujian Baterai	57
Gambar 4. 31 Pengujian Driver Motor	58
Gambar 4. 32 Pengujian Intensitas Cahaya.....	58
Gambar 4. 33 Pengujian Tegangan dan Arus Panel Surya	59
Gambar 4. 34 Pengujian Panel Surya Static	60
Gambar 4. 35 Pengujian Panel Surya Tracker	61
Gambar 4. 36 Pemilihan Board.....	63
Gambar 4. 37 Proses Compiling	63
Gambar 4. 38 Done Compiling.....	64
Gambar 4. 39 Grafik Input Data Solar Tracker.....	64
Gambar 4. 40 Grafik Output Data Solar Tracker	65

DAFTAR TABEL

Tabel 1. 1 Jumlah pelanggan daya listrik 2015-2023.....	1
Tabel 1. 2 Data nilai Rs dan n setelah di konversi	3
Tabel 3. 1 <i>Timeline</i> Waktu Penelitian.....	36
Tabel 4. 1 Pengujian panel surya <i>static</i>	59
Tabel 4. 2 Pengujian Solar Tracker	60
Tabel 4. 3 Data Hasil Pengujian Buck Converter	52



DAFTAR LAMPIRAN

LAMPIRAN A Standard Operational Procedure (SOP).....	A-1
LAMPIRAN B Coding Solar Tracker Master	B-1
LAMPIRAN C Rancangan Anggaran Biaya	C-1
LAMPIRAN D Daftar Riwayat Hidup	D-1



DAFTAR PUSTAKA

- Andesta Putri, W., & Krismadinata, K. (2022). *Monitoring Hasil Pelacakan Cahaya Matahari Sumbu Tunggal*. *MSI Transaction on Education*, 3(1), 15–28.
- Jember, U. (2021). *Repository Digital Repository Universitas Jember Jember Digital Digital Repository Repository Universitas Universitas Jember Jember Proceedings of the International Conference on Climate Change and Sustainability Engineering in ASEAN*.
- Erman Al Hakim, Fahrudin, B., Asni Tafrikhatin, & Jati Sumarah. (2023). *Rancang Bangun Solar tracker Berbasis Arduino*. In *JASATEC : Journal of Students of Automotive, Electronic and Computer* (Vol. 2, Issue 1).
- Prasetyo, M. A., & Wardana, H. K. (2021). *Rancang Bangun Monitoring Solar Tracking System Menggunakan Arduino dan Nodemcu Esp 8266 Berbasis IoT*. *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*, 4(2), 163.
- Gutierrez, S., Rodrigo, P. M., Alvarez, J., Acero, A., & Montoya, A. (2020). *Development and testing of a single-axis photovoltaic sun tracker through the internet of things*. *Energies*, 13(10).
- Syafrialdi, R., & Wildian. (2017). *Rancang Bangun Solar tracker Berbasis Mikrokontroler Atmega8535 Dengan Sensor Ldr Dan Penampil Lcd*. *Jurnal Fisika Unand*, 4(2), 113–122.
- Hariz Elvia Santoso. (2014). *Rancang Bangun Solar Tracking System Menggunakan Kontrol Pid Pada Sumbu Azimuth*. *Teknik Fisika FTI-ITS*, 1–81.
- Santika, D. D., Susanti, B., Anderson, W., & Wongso, K. (2007). *Implementasi Metode Backpropagation Dalam Klasterisasi Objek*. *CommIT (Communication and Information Technology) Journal*, 1(1), 30.
- Putra, A. M., & Aslimeri, A. (2020). *Sistem Kendali Solar tracker Satu Sumbu berbasis Arduino dengan sensor LDR*. *JTEV (Jurnal Teknik Elektro Dan Vokasional)*, 6(1), 322.
- Hakim, T. D., & Sukma, M. (2022). *Rancang Bangun Dual-Axis Solar tracker Menggunakan Mikrokontroler Arduino Mega 2560*. *Jurnal Elektro*, 10(ISSN), 2302–4712.

- Swamardika, I. B. A., Wijaya, I. W. A., & W, I. M. B. P. (2015). *Rancang Bangun Sistem Tracking Panel Surya Berbasis Mikrokontroler Arduino*. *E-Journal SPEKTRUM*, 2(2), 115–120.
- Wijaya, I, K. (2020). *Penggunaan Dan Pemilihan Pengaman Mini Circuit Breaker (Mcb)*. *Teknologi Elektro*, 6(2), 1–4.
- Suriansyah, B. (2019). *RANCANG BANGUN CATU DAYA CADANGAN BERKAPASITAS 100 Ah / 12 V UNTUK LABORATORIUM OTOMASI INDUSTRI*. *Jurnal INTEKNA : Informasi Teknik Dan Niaga*, 19(2), 73–77.
- Athar, M., Sidjoko, R. I., & Suhanto. (2020). *Rancang Bangun Kontrol dan Monitoring Pembangkit Energy Hybrid Solar Cell dan Hydropower Berbasis Internet of Things*. *Snitp*, 5, 1–8.
- Azis, A., Rijaluddin Tahfiz, M., & Nurdiana, N. (2023). *Perancangan Sistem Penggerak Panel Surya Pada Pembangkit Listrik Tenaga Surya Mobile Berbasis Arduino*. *Electrician : Jurnal Rekayasa Dan Teknologi Elektro*, 17(2), 161–168.
- Statistik, S. T. (n.d.). *Statistik 2 pln 3. 3*.
- Wahab, F. (2021). *Rancang bangun penjejak posisi matahari menggunakan kamera dan single board computer*. *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, Dan Listrik Tenaga)*, 1(1), 43–50.
- Setyono, A. E., & Kiono, B. F. T. (2021). *Dari Energi Fosil Menuju Energi Terbarukan: Potret Kondisi Minyak dan Gas Bumi Indonesia Tahun 2020 – 2050*. *Jurnal Energi Baru Dan Terbarukan*, 2(3), 154–162.
- Almas, M. F., Setiawan, B. D., & Sutrisno. (2018). *Implementasi Metode Backpropagation untuk Prediksi Harga Batu Bara*. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 6502–6511.
- Saputra, E. A., Fauzi, R., Solli, T. S., & Demmassewa, J. (2023). *Rancang Bangun Solar tracker Dengan Pengukuran Arus Dan Tegangan Secara Digital Berbasis Mikrokontroler*. *Foristek*, 14(1).
- Ardina, G. B. (2019). *Rancang Bangun Dual Axis Solar tracker Pembangkit Listrik Tenaga Surya Berbasis Mikrokontroler Arduino Uno*. *Seminar Hasil Elektro S1 ITN Malang*, 1(1), 1–11.
- Pangestuningtyas, D. ., Hermawan, H., & Karnoto, K. (2020). *Analisis sudut panel solar cell terhadap daya output dan efisiensi yang dihasilkan*. *Turbo : Jurnal Program Studi Teknik Mesin*, 8(2), 0–7.

- Kurniawan, A. H., & Rivai, M. (2018). *Sistem Stabilisasi Nampun Menggunakan IMU Sensor Dan Arduino Nano*. *Jurnal Teknik ITS*, 7(2).
- Suryadi. (2017). *Sistem Kendali dan Monitoring Listrik Rumahan Menggunakan Ethernet Sheeld dan RTC (Real Time Clock) Arduino*. *Jurnal Teknologi Dan Rekayasa*, 2(1), 14.
- Fachruddin, Imam. (2009). *Desain Penelitian*. Malang: Universitas Islam Negeri
- Suandri, S. E., & Hais, Y. R. (2023). *RANCANG BANGUN SOLAR TRACKER SATU SUMBU PADA PANEL SURYA 50 WP BERBASIS INTERNET OF THING (IOT)*. x, 374–386.
- Utama, A. W. (2019). *Rancang Bangun Solar tracker dengan Sensor Light Dependent Resistor Berbasis Arduino*. *Jurnal Citra Widya Edukasi*, XI(1), 101–118.
- Felycia, F. (2020). *Solar Cell Tracking System Dengan Lux Meter Berbasis Arduino Uno R3*. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 7(2), 132–140.
- Rahmatullah, al, Uji Krismanto, A., & Pudji Muljanto, W. (2023). *Rancang Bangun dan Analisa Unjuk Kerja Single Axis Solar Tracker Berbasis Logika Fuzzy*. *Jurnal Magnetika*, 7(2), 365–373.
- Zakariah, A., Famarzi, M., Jamian, J. J., & Yunus, M. A. M. (2015). *Medium size dual-axis solar tracking system with sunlight intensity comparison method and fuzzy logic implementation*. *Jurnal Teknologi*, 77(17), 145–157. <https://doi.org/10.11113/jt.v77.6468>
- Syhab, A. S., Romadhon, H. C., & Hakim, M. L. (2019). *Rancang Bangun Solar Tracker Otomatis Pada Pengisian Energi Panel Surya Bebasis Internet of Things*. *Jurnal Meteorologi Klimatologi Dan Geofisika*, 6(2), 21–29. <https://doi.org/10.36754/jmkg.v6i2.120>

LAMPIRAN

LAMPIRAN A Standard Operational Procedure (SOP)

RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGAH DENGAN
METODE BACKPROPAGATION BERBASIS *INTERNET OF THINGS* (IOT)

Oleh :

Nasyafia Ahsanul Amala
NIT. 30121019

Standard Operational Procedure (SOP) dalam pengoperasian alat adalah panduan yang merinci langkah-langkah yang harus diikuti untuk menggunakan alat dengan benar dan aman. SOP ini mencakup instruksi rinci tentang cara menghidupkan dan mematikan alat. Dimana dengan menggunakan Standar Operational Procedure (SOP) yang telah dibuat bertujuan untuk menjaga alat Proyek Akhir terhindar dari kerusakan atau salah prosedur, setiap pengguna alat akan mengikuti prosedur yang sama, sehingga dapat menjamin konsistensi dan akurasi hasil. Para pengguna harus memahami dan mengikuti SOP ini dengan cermat untuk memastikan operasi yang aman, efisien, dan tepat sesuai dengan tujuan penggunaan alat tersebut.

Berikut merupakan Standar Operational Procedure (SOP) untuk menghidupkan dan mematikan alat Tugas Akhir dengan judul “RANCANG BANGUN *SOLAR TRACKER* SUMBU TUNGGAH DENGAN METODE BACKPROPAGATION BERBASIS *INTERNET OF THINGS* (IOT)” sebagai berikut :

1. Persiapan Awal

1. Periksa Peralatan:

- a. Langkah: Periksa semua komponen sistem, termasuk panel surya, baterai, MPPT controller, SCC, driver motor, dan sensor.

- b. Tujuan: Memastikan semua peralatan tersedia dan dalam kondisi baik sebelum pemasangan.

2. Hubungan Kabel

- a. Hubungan Kabel Controller dari Panel Surya:

Langkah: Sambungkan kabel output dari panel surya ke input pada Solar Charge Controller (SCC).

Tujuan: Mengalirkan energi dari panel surya ke SCC untuk pengaturan lebih lanjut.

- b. Hubungan Kabel dari SCC ke Battery:

Langkah: Hubungkan kabel output dari SCC ke terminal baterai, memastikan polaritas kabel sesuai.

Tujuan: Mengisi baterai dengan energi yang dihasilkan dari panel surya.

- c. Hubungan Kabel Output SCC ke Driver Motor:

Langkah: Sambungkan kabel output dari SCC ke input driver motor (BTS7960).

Tujuan: Mengontrol motor linear actuator yang menggerakkan panel surya.

- d. Hubungkan Kabel Output ke Beban:

Langkah: Hubungkan kabel output dari driver motor ke panel surya.

Tujuan: Memastikan panel surya dapat diposisikan sesuai dengan sinyal dari driver motor.

3. Pengaktifan Sistem

- 1. Tekan Tombol ON pada Output MPPT:

Langkah: Aktifkan sistem dengan menekan tombol ON pada MPPT controller.

Tujuan: Memulai proses tracking dan pengaturan untuk efisiensi maksimum.

4. Pengaturan dan Monitoring

Buka Aplikasi Solar Tracker:

Langkah: Buka aplikasi solar tracker yang dikonfigurasi pada perangkat IoT.

Tujuan: Memantau dan mengontrol sistem solar tracker secara jarak jauh.

5. Pantau Hasil Tegangan dari Tracking Solar Panel:

Langkah: Gunakan aplikasi untuk memantau hasil tegangan dan posisi panel surya.

Tujuan: Memastikan panel surya mengikuti pergerakan matahari dengan benar dan memaksimalkan penyerapan energi.

6. Pemeriksaan dan Penyesuaian

a. Periksa Kinerja Sistem:

Langkah: Tinjau data dari aplikasi termasuk tegangan output, posisi panel, dan respons motor.

Tujuan: Memastikan sistem berfungsi sesuai harapan.

b. Lakukan Penyesuaian:

Langkah: Sesuaikan pengaturan solar tracker melalui aplikasi jika diperlukan.

Tujuan: Mengoptimalkan kinerja sistem untuk efisiensi terbaik.

6. Dokumentasi dan Pelaporan

7. Catat Hasil:

Langkah: Dokumentasikan hasil pengujian seperti tegangan terukur, posisi panel, dan respons motor.

Tujuan: Menyimpan data untuk analisis dan referensi di masa depan.

8. Laporan:

Langkah: Buat laporan mengenai kinerja sistem termasuk masalah dan tindakan yang diambil.

Tujuan: Memberikan ringkasan hasil dan rekomendasi untuk perbaikan.

LAMPIRAN B Coding Solar Tracker Master

```
#include <Arduino.h>

#define NumberOf(arg) ((unsigned int) (sizeof (arg) / sizeof (arg [0]))) //
calculates the number of layers (in this case 4)
#define _1_OPTIMIZE 0B11010010 // 0B1000000 For PROGMEM see also
https://github.com/GiorgosXou/NeuralNetworks#define-macro-properties

#include <NeuralNetwork.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_INA219.h>
#include <uRTCLib.h>
#include <FastIMU.h>
#include <BH1750.h>
#include <SoftwareSerial.h>

#define INA219_INPUT_ADDRESS 0x40
#define INA219_OUTPUT_ADDRESS 0x41
#define RTC_DS3231_ADDRESS 0x68
#define LCD_I2C_ADDRESS 0x27
#define MPU_ADDRESS 0x69
#define LUX1_BH1750_ADDRESS 0x23
#define LUX2_BH1750_ADDRESS 0x5C

#define PWM1_Pin 4
#define PWM2_Pin 5
#define LDR1_Pin 2
#define LDR2_Pin 3
#define SER2_RX_Pin 17
#define SER2_TX_Pin 16

#define LCD_WIDHT 16
#define LCD_HEIGHT 2

LiquidCrystal_I2C lcd(LCD_I2C_ADDRESS, LCD_WIDHT,
LCD_HEIGHT); // set the LCD address to 0x27 for a 16 chars and 2 line
display
Adafruit_INA219 ina219_IN(INA219_INPUT_ADDRESS);
```

```

Adafruit_INA219 ina219_OUT(INA219_OUTPUT_ADDRESS);
uRTCLib rtc(RTC_DS3231_ADDRESS);
MPU6500 IMU;
BH1750 lightMeter1(LUX1_BH1750_ADDRESS);
BH1750 lightMeter2(LUX2_BH1750_ADDRESS);
SoftwareSerial SoftSerial(SER2_RX_Pin, SER2_TX_Pin);
NeuralNetwork *NN;

calData calib = { 0 }; //Calibration data
AccelData accelData; //Sensor data
GyroData gyroData;
MagData magData;

byte rtcModel = URTCLIB_MODEL_DS3232;

int period = 1000;
unsigned long time_now = 0;

float sourceVoltage = 0;
int sourceCurrent_mA = 0;
float loadVoltage = 0;
int loadCurrent_mA = 0;
int ldr1Val = 0;
int ldr2Val = 0;
unsigned int lux1 = 0;
unsigned int lux2 = 0;
int pwm1Value = 0;
int pwm2Value = 0;
int xDegree = 0;

uint8_t motorState = 0;
String motorState_message[3]={"OFF", "CW", "CCW"};

//=====
//=====
//=====
//=====//

const unsigned int layers[] = {2, 6, 6, 2};
float *output; // 4th layer's output(s)

```

```

// Default Test/Input data
float inputs[2];

// 1 for each layer-layer [Pretrained Biases ]
const PROGMEM float biases[] = {1.0000000, 1.0000000, 2.4066517};

// It is 3*9 + 9*9 + 9*1 [Pretrained weights]
const PROGMEM float weights[] = {
-0.0365911,-1.2009180,
0.0090851, 0.9528944,
0.6479358,-1.9393458,
4.8927989,-6.4303450,
-4.4479522, 2.9731180,
-6.0946950, 4.6438279,

0.5616179, 0.0473046, 0.6400791, 0.4222403,-0.8266851, 0.4178025,
-0.6402900,-0.1223926, 0.1093126,-1.0732821, 0.7995800, 0.7497338,
-0.2196514, 0.0410195, 0.1238548, 2.6663417,-2.0681519,-1.9802053,
-0.8379331,-0.3401756,-2.0538005,-5.4181876, 2.8564683, 4.6882381,
0.9094682,-0.5923957, 1.3063753, 5.5710191,-3.2854499,-4.9869074,
-0.4299846, 0.0751875,-0.1499110,-2.4158465, 0.5663628, 1.4677188,

-1.4608854,-2.2856774, 1.8997262,-7.1330175, 5.1937079,-3.1147053,
-2.0096275,-0.1919908,-3.5846047, 4.9163284,-7.6655526, 0.5968750,
};

int period_NeuralNetwork = 500;
unsigned long time_NeuralNetwork = 0;
float backPropagation = 0;
//=====
//=====
//=====
//=====//

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");

```

```

SoftSerial.begin(115200);
SoftSerial.println("Hello Streamer");

// Creating a NeuralNetwork with pretrained Weights and Biases
NN = new NeuralNetwork(layers, weights, biases, NumberOf(layers));
Serial.println("Init Neural Network Success ...");

if (! ina219_IN.begin()) {
  Serial.println("Failed to find INA219 Input chip");
  while (1) { delay(10); }
}
Serial.println("Init INA219 Input Success ...");

if (! ina219_OUT.begin()) {
  Serial.println("Failed to find INA219 Output chip");
  while (1) { delay(10); }
}
Serial.println("Init INA219 Output Success ...");

URTCLIB_WIRE.begin();
// set RTC Model
rtc.set_model(rtcModel);
// refresh data from RTC HW in RTC class object so flags like rtc.lostPower(),
rtc.getEOSCFlag(), etc, can get populated
rtc.refresh();
// RTCLib::set(byte second, byte minute, byte hour (0-23:24-hr mode only),
byte dayOfWeek (Sun = 1, Sat = 7), byte dayOfMonth (1-12), byte month, byte
year)
// rtc.set(0, 56, 11, 1, 2, 6, 24);
if (rtc.enableBattery()) {
  Serial.println("Battery activated correctly.");
} else {
  Serial.println("ERROR activating battery.");
}
Serial.println("Init RTC DS3231 Success ...");

int err = IMU.init(calib, MPU_ADDRESS);
if (err != 0) {
  Serial.print("Error initializing IMU.");
}

```



```

Serial.println(err);
while (true) {
    ;
}
}
err = IMU.setGyroRange(500);    //USE THESE TO SET THE RANGE, IF
AN INVALID RANGE IS SET IT WILL RETURN -1
err = IMU.setAccelRange(2);    //THESE TWO SET THE GYRO RANGE
TO ±500 DPS AND THE ACCELEROMETER RANGE TO ±2g
if (err != 0) {
    Serial.print("Error IMU Setting range.");
    Serial.println(err);
    while (true) {
        ;
    }
}
Serial.println("Init IMU Success ...");

lightMeter1.begin(0x13, LUX1_BH1750_ADDRESS, &Wire);
lightMeter2.begin(0x13, LUX2_BH1750_ADDRESS, &Wire);

Serial.println("Init LUX Sensor Success ...");

lcd.init();           // initialize the lcd
lcd.clear();
lcd.backlight();
Serial.println("Init i2C LCD Success ...");

pinMode(PWM1_Pin, OUTPUT);
pinMode(PWM2_Pin, OUTPUT);
Serial.println("Init GPIO Success ...");

// Print a message to the LCD.
lcd.setCursor(0,0);
lcd.print(" Solar Tracker ");
delay(1000);
lcd.clear();
}

```

```

void loop() {
    // put your main code here, to run repeatedly:
    check_Serial1();
    sensor_getData();
    calculate_NeuralNetwork();

    if(millis() - time_now >= period){
        time_now = millis();
        lcd.clear();

        rtc.refresh();

        Serial.print("Time:" + String(rtc.hour()) + ":" + String(rtc.minute()) + "
Date:" + String(rtc.day()) + "/" + String(rtc.month()) + "/" + String(rtc.year()) +
"\t");
        Serial.print("ldr1:" + String(ldr1Val) + "\t" + "ldr2:" + String(ldr2Val) + "\t");
        Serial.print("lux1:" + String(lux1) + "\t" + "lux2:" + String(lux2) + "\t");
        Serial.print("Deg:" + String(xDegree) + "\t");
        Serial.print("Vin:" + String(sourceVoltage) + "\t" + "Vout:" +
String(loadVoltage) + "\t");
        Serial.print("Iin:" + String(sourceCurrent_mA/1000) + "\t" + "Iout:" +
String(sourceCurrent_mA/1000) + "\t");
        Serial.print("NNin1:" + String(inputs[0]) + "\t" + "NNin2:" +
String(inputs[1]) + "\n");
        Serial.print("NNout1:" + String(output[0]) + "\t" + "NNout2:" +
String(output[1]) + "\n");

        stream_data();
    }

    moving_Motor();
    delay(5);

}

void moving_Motor(){
    if((rtc.hour())>=18 && rtc.hour()<=23)){ //|| (rtc.hour())>=0 &&
rtc.hour()<=4)){
        if(xDegree<134){

```

```

    pwm1Value = 200;
    pwm2Value = 0;
}
if(xDegree>=134){
    pwm1Value = 0;
    pwm2Value = 0;
}
}
if(xDegree<=46){
    pwm1Value = 0;
    pwm2Value = 0;
}

if(pwm1Value == pwm2Value){
    motorState = 0;
}
if(pwm1Value > pwm2Value){
    motorState = 1;
}
if(pwm1Value < pwm2Value){
    motorState = 2;
}
analogWrite(PWM1_Pin, pwm1Value);
analogWrite(PWM2_Pin, pwm2Value);
}

void sensor_getData(){
    ldr1Val = map(analogRead(LDR1_Pin), 0, 1023, 0, 100);
    ldr2Val = map(analogRead(LDR2_Pin)-55, 0, 1023, 0, 100);

    lux1 = map(lightMeter1.readLightLevel(), 0, 60000, 0, 1000);
    lux2 = map(lightMeter2.readLightLevel(), 0, 60000, 0, 1000);

    sourceVoltage = ina219_IN.getBusVoltage_V();
    sourceCurrent_mA = ina219_IN.getCurrent_mA();
    loadVoltage = ina219_OUT.getBusVoltage_V();
    loadCurrent_mA = ina219_OUT.getCurrent_mA();

    IMU.update();
    IMU.getAccel(&accelData);

```

```

    xDegree = (accelData.accelX+1)*90-5;
}
void check_Serial1(){
    if (Serial.available() > 0) {
        String str = Serial.readString();
        String startStr = str.substring(str.indexOf('#'), str.indexOf('#') + 1);
        String stopStr = str.substring(str.indexOf(';'), str.indexOf(';') + 1);
        String dataStr;
        // Serial.println(str);
        // Serial.println(stopStr);
        if(startStr.equals("#") && stopStr.equals(";")){
            dataStr = str.substring(str.indexOf('#') + 1, str.indexOf(';'));
            int pwmVal = dataStr.toInt();
            Serial.println(pwmVal);

            if(pwmVal>0){
                analogWrite(PWM1_Pin, pwmVal);
                analogWrite(PWM2_Pin, 0);
            }
            else if (pwmVal<0){
                analogWrite(PWM1_Pin, 0);
                analogWrite(PWM2_Pin, pwmVal*-1);
            }
            else{
                analogWrite(PWM1_Pin, 0);
                analogWrite(PWM2_Pin, 0);
            }
        }

        delay(2000);
        analogWrite(PWM1_Pin, 0);
        analogWrite(PWM2_Pin, 0);
    }
}

void calculate_NeuralNetwork(){
    if(millis() - time_NeuralNetwork >= period_NeuralNetwork){
        time_NeuralNetwork = millis();

        inputs[0] = lux1/1000.0;
    }
}

```

```

    inputs[1] = lux2/1000.0;
    output = NN->FeedForward(inputs);
    backPropagation = output[0];
}

if(output[0]>=0.5){pwm2Value = 200;}
else if(output[0]<0.5){pwm2Value = 0;}
if(output[1]>=0.5){pwm1Value = 200;}
else if(output[1]<0.5){pwm1Value = 0;}

}

void stream_data(){
    // =>
    '/api/data/{vin}/{vout}/{iin}/{iout}/{ldr1}/{ldr2}/{back}/{motor}/{lumens}/
    {slope}/{elongasi}'
    String data = "data/" + String(sourceVoltage) + "/" + String(loadVoltage) + "/"
    + String(sourceCurrent_mA/1000.0) + "/" + String(loadCurrent_mA/1000.0) +
    "/" + String(lux1) + "/" + String(lux2) + "/" + String(backPropagation) + "/" +
    motorState_message[motorState] + "/" + String(lux1) + "/" + String(xDegree)
    + "/" + String(xDegree) ;
    Serial.print("Data to Streamer: ");
    Serial.println(data);
    SoftSerial.println(data);
}

```

1. Coding streamer

```

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>

#define SSER_TX_Pin D1
#define SSER_RX_Pin D2

// software serial #1: RX = digital pin D7, TX = digital pin D8/D4
SoftwareSerial SoftSerial(SSER_RX_Pin, SSER_TX_Pin);

const char* ssid = "createme.id";
const char* password = "ccaaddok";

```

```

// SINGLE AXIS SOLAR TRACKER
// https://solartracker.cloud/
//                                     Api                                     :
https://solartracker.cloud/api/data/14.5/119.20/1.9/2.6/100/200/0.9999/off/100/90/45
//                                     =>
'/api/data/{vin}/{vout}/{iin}/{iout}/{ldr1}/{ldr2}/{back}/{motor}/{lumens}/{slope
}/{elongasi}'
String urlBase = "https://solartracker.cloud/api/";
String url = "";

unsigned long time_now = 0;
const long streamPeriod = 1000;

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)

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 Streamer");

  // 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.println(url);
}

```

```

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

    data = SoftSerial.readString();
    dataSerial = data;
    Serial.print("serial: ");
    Serial.println(dataSerial);
  }
  if (Serial.available() > 0) {

    data = Serial.readString();
    dataSerial = data;
    Serial.print("serial: ");
    Serial.println(dataSerial);
  }
}

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

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

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. Coding Training Backpropagation

```

#define NumberOf(arg) ((unsigned int) (sizeof (arg) / sizeof (arg [0]))) //
calculates the number of layers (in this case 3)
#define _2_OPTIMIZE 0B00000000 // Enable 0B01.. for NO_BIAS or
0B001.. for MULTIPLE_BIASES_PER_LAYER
#define _1_OPTIMIZE 0B00011000 //
https://github.com/GiorgosXou/NeuralNetworks#define-macro-properties
#define Sigmoid // Comment this line to use Sigmoid (the default)
activation function

#include <NeuralNetwork.h>

const unsigned int layers[] = {2, 6, 6, 2}; // 3 layers (1st)layer with 3 input
neurons (2nd)layer 5 hidden neurons each and (3rd)layer with 1 output neuron
float *output; // 3rd layer's output(s)

//Default Inputs/Training-Data
const float inputs[121][2] = {
    0.0, 0.0,
    0.0, 0.1,
    0.0, 0.2,
    0.0, 0.3,
    0.0, 0.4,
    0.0, 0.5,
    0.0, 0.6,
    0.0, 0.7,
    0.0, 0.8,
    0.0, 0.9,
    0.0, 1.0,
    0.1, 0.0,
    0.1, 0.1,
    0.1, 0.2,
    0.1, 0.3,
    0.1, 0.4,
    0.1, 0.5,
    0.1, 0.6,
    0.1, 0.7,
    0.1, 0.8,
    0.1, 0.9,
    0.1, 1.0,

```

0.2, 0.0,
 0.2, 0.1,
 0.2, 0.2,
 0.2, 0.3,
 0.2, 0.4,
 0.2, 0.5,
 0.2, 0.6,
 0.2, 0.7,
 0.2, 0.8,
 0.2, 0.9,
 0.2, 1.0,
 0.3, 0.0,
 0.3, 0.1,
 0.3, 0.2,
 0.3, 0.3,
 0.3, 0.4,
 0.3, 0.5,
 0.3, 0.6,
 0.3, 0.7,
 0.3, 0.8,
 0.3, 0.9,
 0.3, 1.0,
 0.4, 0.0,
 0.4, 0.1,
 0.4, 0.2,
 0.4, 0.3,
 0.4, 0.4,
 0.4, 0.5,
 0.4, 0.6,
 0.4, 0.7,
 0.4, 0.8,
 0.4, 0.9,
 0.4, 1.0,
 0.5, 0.0,
 0.5, 0.1,
 0.5, 0.2,
 0.5, 0.3,
 0.5, 0.4,
 0.5, 0.5,
 0.5, 0.6,
 0.5, 0.7,
 0.5, 0.8,
 0.5, 0.9,
 0.5, 1.0,
 0.6, 0.0,
 0.6, 0.1,



0.6, 0.2,
 0.6, 0.3,
 0.6, 0.4,
 0.6, 0.5,
 0.6, 0.6,
 0.6, 0.7,
 0.6, 0.8,
 0.6, 0.9,
 0.6, 1.0,
 0.7, 0.0,
 0.7, 0.1,
 0.7, 0.2,
 0.7, 0.3,
 0.7, 0.4,
 0.7, 0.5,
 0.7, 0.6,
 0.7, 0.7,
 0.7, 0.8,
 0.7, 0.9,
 0.7, 1.0,
 0.8, 0.0,
 0.8, 0.1,
 0.8, 0.2,
 0.8, 0.3,
 0.8, 0.4,
 0.8, 0.5,
 0.8, 0.6,
 0.8, 0.7,
 0.8, 0.8,
 0.8, 0.9,
 0.8, 1.0,
 0.9, 0.0,
 0.9, 0.1,
 0.9, 0.2,
 0.9, 0.3,
 0.9, 0.4,
 0.9, 0.5,
 0.9, 0.6,
 0.9, 0.7,
 0.9, 0.8,
 0.9, 0.9,
 0.9, 1.0,
 1.0, 0.0,
 1.0, 0.1,
 1.0, 0.2,
 1.0, 0.3,



0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
1, 0,
0, 0,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
1, 0.

The logo of Politeknik Penerbangan Surabaya is a circular emblem. The outer ring is green with the text 'POLITEKNIK PENERBANGAN' at the top and 'SURABAYA' at the bottom in white. The center features a blue globe with white grid lines. Overlaid on the globe is a yellow winged emblem with a green circular center. Below the globe is a white banner with the text 'PILAR HATI BERSAMA' in blue.[illegible]

0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
0, 0,
0, 1,
0, 1,
0, 1,
0, 1,
0, 1,
1, 0,
1, 0,
1, 0,
1, 0.

The logo of Politeknik Penerbangan Surabaya is a circular emblem. The outer ring is green with the text 'POLITEKNIK PENERBANGAN' at the top and 'SURABAYA' at the bottom in white. The center features a purple globe with white grid lines. Overlaid on the globe is a yellow winged figure with its wings spread wide. Below the globe is a white banner with the text 'POLITEKNIK PENERBANGAN SURABAYA' in green.

```

1, 0,
0, 0,
0, 1,
0, 1,
0, 1,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
0, 0,
0, 1,
0, 1,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
0, 0,
0, 1,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
1, 0,
0, 0,
};

```

```

void setup()
{
  Serial.begin(9600);
  NeuralNetwork NN(layers, NumberOf(layers)); // Creating a Neural-
Network with default learning-rates

```



```

do{
    for (int j = 0; j < NumberOf(inputs); j++) // Epoch
    {
        NN.FeedForward(inputs[j]);    // FeedForwards the input arrays through
the NN | stores the output array internally
        NN.BackProp(expectedOutput[j]); // "Tells" to the NN if the output was
the-expected-correct one | then, "teaches" it
    }

    // Prints the Error.
    Serial.print("MSE: ");
    Serial.println(NN.MeanSqrdError,6);

    // Loops through each epoch Until MSE goes < 0.003
}while(NN.getMeanSqrdError(NumberOf(inputs)) > 0.001);

Serial.println("\n =-[OUTPUTS]-=");

NN.print();

//Goes through all the input arrays
    output = NN.FeedForward(inputs[1]); // FeedForwards the input[i]-array
through the NN | returns the predicted output(s)
    Serial.println(output[0], 7);    // Prints the first 7 digits after the comma.
    Serial.println(output[1], 7);    // Prints the first 7 digits after the
comma.    // Prints the weights and biases of each layer
}
void loop() {}

```

LAMPIRAN C Rancangan Anggaran Biaya

NO	NAMA	JUMLAH	HARGA	
			SATUAN	TOTAL
1.	Panel Surya 50wp	1	Rp 500,000	Rp 500,000
2.	Arduino Mega 2560	1	Rp 350,000	Rp 350,000
3.	Sensor BH1750	2	Rp 120,000	Rp 240,000
4.	Sensor INA219	2	Rp 45,000	Rp 90,000
5.	Sensor IMU	1	Rp 100,000	Rp 100,000
6.	MPPT	1	Rp 500,000	Rp 500,000
7.	RTC	1	Rp 45,000	Rp 45,000
8.	Motor Linear Actuator	1	Rp 650,000	Rp 650,000
9.	Driver Motor BTS7960	1	Rp 65,000	Rp 65,000
10.	Battery/Aki	1	Rp 500,000	Rp 500,000
11.	Kabel	1	Rp 30,000	Rp 30,000
12.	Besi Siku Lubang	3	Rp 70,000	Rp 210,000
TOTAL HARGA				Rp 3.290,000

LAMPIRAN D Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



Nasyafia Ahsanul Amala, lahir di Malang Jawa Timur pada tanggal 4 Juni 2002, anak kedua dari dua bersaudara, dari pasangan Bapak Moh.Firdaus dan Ibu Tatik Maulida. Mempunyai satu saudara kandung Yusril Zulfan Firdaus. Beragama Islam. Bertempat tinggal di Jalan Jeruk Dusun Mergosingo No.82 RT 02/RW 03, Desa Jatirejoyoso, Kecamatan Kepanjen, Kabupaten Malang, Provinsi Jawa Timur. Memulai pendidikan sekolah dasar di MINU Jatirejoyoso, dan lulus pada tahun 2014, melanjutkan sekolah di SMPN 1 Kepanjen dan lulus pada tahun 2017, melanjutkan ke MAN 1 MALANG dan lulus tahun 2020. Pada Tahun 2021 diterima sebagai Taruna di Politeknik Penerbangan Surabaya, pada Program Studi Diploma 3 Teknik Listrik Bandara Angkatan XVI. Melaksanakan Kegiatan On The Job Training I di Bandar Udara Juwata Tarakan Kalimantan Utara yang terhitung mulai tanggal 8 Mei 2023 – 22 September 2023 dan On The Job Training II di Bandar Udara Internasional Sultan Hasanuddin Makassar yang terhitung mulai tanggal 2 Oktober 2024 – 29 Februari 2024.