

OPTIMASI POSISI PANEL SURYA *DUAL-AXIS* MENGGUNAKAN PENDEKATAN *MACHINE LEARNING* UNTUK MEMAKSIMALKAN DAYA KELUARAN

Khoiru Adi Nugroho; Hasyim Asy'ari, S.T.,M.T

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah
Surakarta

Abstrak

Penelitian ini bertujuan untuk merancang sistem optimasi posisi panel surya *dual-axis* berbasis *Internet of Things (IoT)* menggunakan pendekatan *machine learning* untuk memaksimalkan daya keluaran panel surya. Peralatan utama yang digunakan meliputi Panel Surya GH Solar 30 Wp, *Solar Charge Controller (SCC)* 20A, Baterai Inforce 12V/9AH, motor servo sebagai penggerak panel, sensor BH-1750 untuk mengukur intensitas cahaya (*lux*), sensor INA219 untuk mengukur tegangan, arus, dan daya, sensor DS18B20 untuk mengukur suhu panel, serta modul RTC sebagai acuan waktu pengambilan data. Data dikumpulkan setiap 10 detik pada pukul 09.00–15.00 WIB sehingga diperoleh sekitar 1676 data pengukuran. Seluruh data diproses menggunakan ESP32 dan ditampilkan secara *real-time* melalui platform *IoT*. Algoritma *XGBoost* dilatih terlebih dahulu karena memiliki akurasi tinggi, kemudian hasil prediksinya digunakan sebagai acuan untuk melatih model *Decision Tree* yang lebih sederhana dan dapat diterapkan pada *Arduino IDE*. Berdasarkan hasil pengujian sebelum penerapan *machine learning*, diperoleh rata-rata intensitas cahaya sebesar 30.259 *lux*, tegangan 15,54 V, arus 1,15 A, daya 17,95 W, dan suhu panel 45,97°C. Setelah penerapan *machine learning*, diperoleh rata-rata intensitas cahaya sebesar 25.175 *lux*, tegangan 16,21 V, arus 0,99 A, daya 16,34 W, dan suhu panel 44,62°C. Hasil analisis menunjukkan bahwa rasio daya terhadap intensitas cahaya meningkat dari 0,000593 W/*lux* menjadi 0,000649 W/*lux* atau naik sekitar 13,37%, sehingga sistem mampu menghasilkan daya lebih optimal meskipun intensitas cahaya lebih rendah. Penurunan suhu panel sebesar 3,15% juga menunjukkan bahwa sistem *dual-axis* berbasis *machine learning* mampu mengurangi *thermal losses* melalui pengaturan posisi panel yang lebih optimal. Dengan demikian, sistem yang dikembangkan mampu meningkatkan pemanfaatan energi surya secara lebih optimal dan berkelanjutan. Sebagai keterbatasan penelitian, pengujian dilakukan pada dua hari berbeda dengan kondisi intensitas cahaya yang tidak identik (30.259 *lux* vs 25.175 *lux*), sehingga perbandingan daya absolut tidak dapat dilakukan secara langsung

Kata Kunci: Panel Surya, *Dual-Axis Tracker*, *Internet of Things*, *Machine Learning*, *XGBoost*, *Decision Tree*, Optimasi Daya.

ABSTRACT

This study aims to design an *Internet of Things (IoT)*-based *dual-axis* solar panel optimization system using a *machine learning* approach to maximize solar panel output power. The main equipment used includes a GH Solar 30 Wp Solar Panel, a 20A Solar Charge Controller (SCC), an Inforce 12V/9AH Battery, servo motors as panel actuators, a BH1750 sensor to measure light intensity (*lux*), an INA219 sensor to measure voltage, current, and power, a DS18B20 sensor to measure panel temperature, and an RTC module as a time reference for data acquisition. Data were collected every 10 seconds from 09:00 AM to 03:00 PM, resulting in approximately 1676 measurement data points. All data were processed using ESP32 and displayed in real-time through an *IoT* platform. The *XGBoost* algorithm was trained first due to its high accuracy, and its prediction results were then used as a reference to train a simpler *Decision Tree* model that could be implemented on *Arduino IDE*. Based on the testing results before machine learning implementation, the average light intensity was 30,259 *lux*, voltage was 15.54 V, current was 1.15 A, output power was 17.95 W, and panel temperature was 45.97°C. After machine learning implementation, the average light intensity became 25,175 *lux*, voltage was 16.21 V, current was 0.99 A, output power was 16.34 W, and panel temperature was 44.62°C. The analysis results showed that the power-to-light intensity ratio increased from 0.000593 W/*lux* to 0.000649 W/*lux*, or improved by approximately 13.37%, indicating that the system could generate more optimal power even under lower light intensity conditions. The 3.15% reduction in panel temperature also indicates that the machine learning-based *dual-axis* system was able to reduce thermal losses through more optimal panel positioning..

Keywords: Solar Panel, *Dual-Axis Tracker*, *Internet of Things*, *Machine Learning*, *XGBoost*, *Decision Tree*, Power Optimization.

1. PENDAHULUAN

Energi matahari merupakan salah satu sumber energi utama yang mendukung berbagai proses yang terjadi di Bumi. Energi ini berperan penting dalam mendukung berbagai proses fisik maupun biologis yang berlangsung di lingkungan Bumi. Energi dari matahari sampai ke permukaan Bumi melalui proses radiasi, yaitu perpindahan energi panas dalam bentuk gelombang elektromagnetik tanpa memerlukan media perantara. Hal ini memungkinkan energi matahari dapat merambat melalui ruang hampa di antara Matahari dan Bumi. Gelombang elektromagnetik sendiri merupakan gelombang yang terdiri dari komponen medan listrik dan medan magnet yang saling tegak lurus sehingga mampu merambat dengan kecepatan tinggi tanpa membutuhkan medium. Sebagian energi yang dipancarkan oleh Matahari dan mencapai permukaan Bumi akan diserap oleh permukaan Bumi, yang kemudian menyebabkan peningkatan suhu pada Bumi (Priatam et al. 2021).

Pada umumnya, panel surya dipasang secara statis dengan sudut kemiringan tertentu. Metode ini menyebabkan panel tidak selalu berada pada posisi tegak lurus terhadap arah datang sinar matahari, sehingga energi yang diserap menjadi kurang optimal. Beberapa penelitian menunjukkan bahwa penggunaan sistem *solar tracker dual-axis* mampu meningkatkan daya keluaran secara signifikan dibandingkan sistem panel statis karena panel dapat mengikuti pergerakan matahari pada dua sumbu, yaitu sumbu horizontal dan vertikal (Yapie and Riski 2025).

Pendekatan konvensional berbasis sensor mulai dikombinasikan dengan metode *machine learning* untuk melakukan optimasi sudut panel secara adaptif berdasarkan data historis dan kondisi lingkungan. Algoritma *Decision Tree* digunakan untuk membangun model prediksi berbasis struktur pohon keputusan yang mampu memetakan hubungan antara variabel intensitas cahaya, suhu, waktu, dan sudut panel terhadap daya keluaran (Febriyanty et al. 2026).

Selain itu, algoritma *XGBoost (Extreme Gradient Boosting)* sebagai metode *boosting* berbasis *Decision Tree* mampu meningkatkan akurasi prediksi dengan menggabungkan beberapa model secara iteratif sehingga lebih efektif dalam menentukan sudut optimal yang menghasilkan daya maksimum (Chen and Guestrin 2016). Algoritma *XGBoost* dengan menerapkan teknik *oversampling* dan penyesuaian parameter adalah metode regresi yang cukup efektif untuk diterapkan dalam kasus regresi Penelitian

internasional juga menunjukkan bahwa penerapan metode *machine learning* pada sistem energi surya dapat meningkatkan akurasi prediksi daya keluaran dan optimasi sudut panel secara signifikan dibandingkan metode konvensional (Cyril et al. 2017). Matahari bergerak 15° setiap jamnya akibat rotasi bumi (Qomaruddin, M. N., Khairi, M. 2019) sehingga interval pengambilan data manual pada sistem ini disesuaikan agar panel selalu bisa mengikuti pergerakan tersebut.

2. METODE

2.1. Tahapan Penelitian

Metode yang digunakan adalah rancang bangun sistem optimasi posisi panel surya *dual-axis* berbasis *machine learning* untuk memaksimalkan daya keluaran panel surya, yang dilaksanakan melalui empat tahap.

Tahap pertama adalah studi literatur, meliputi kajian jurnal, buku, dan artikel terkait solar tracker dual-axis, sensor BH1750, INA219, DS18B20, algoritma *XGBoost*, *Decision Tree*, serta implementasi *machine learning* pada sistem *embedded* sebagai dasar perancangan sistem.

Tahap kedua adalah perancangan dan pembuatan alat. Sistem menggunakan ESP32 sebagai pengendali utama yang terintegrasi dengan sensor BH1750 (intensitas cahaya), INA219 (tegangan, arus, dan daya), DS18B20 (suhu panel), modul RTC (acuan waktu), dua motor servo sebagai penggerak dual-axis, serta LCD 20x4 sebagai penampil data. Data dikumpulkan setiap 10 detik pada pukul 09.00–15.00 WIB untuk melatih model *XGBoost*, yang kemudian disederhanakan menjadi *Decision Tree* agar dapat diterapkan pada *Arduino IDE*.

Tahap ketiga adalah pengujian sistem, meliputi kemampuan pembacaan sensor, kontrol posisi panel secara otomatis, dan pemantauan parameter listrik secara *real-time*, serta perbandingan performa panel sebelum dan sesudah penerapan *machine learning*.

Tahap keempat adalah analisis hasil, yaitu membandingkan intensitas cahaya, daya keluaran, suhu panel, dan rasio daya terhadap intensitas cahaya sebelum dan sesudah penerapan *machine learning* untuk mengevaluasi peningkatan efisiensi sistem.

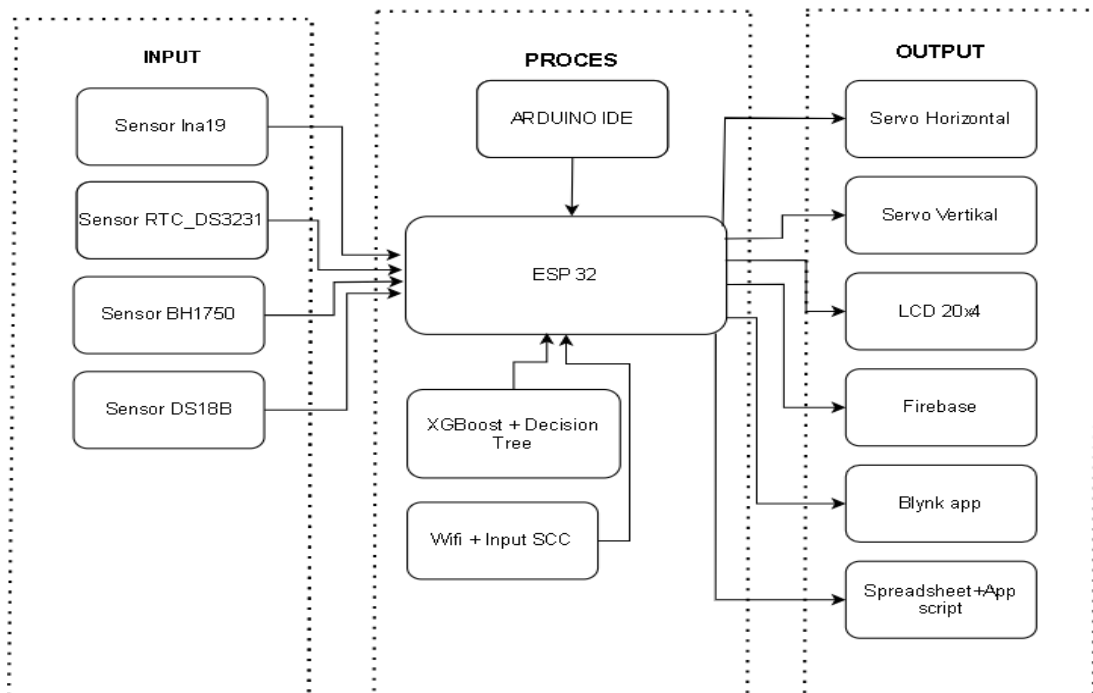
2.2 Alat dan Bahan

Dalam penelitian ini terdiri dari beberapa komponen yang mendukung keberhasilan kinerja sistem. Pemilihan komponen dilakukan menyesuaikan kebutuhan penelitian. Daftar alat dan bahan yang digunakan tertera pada tabel 1 berikut.

Tabel 1. Alat dan Bahan

No.	Alat dan Bahan	Jumlah
1	Panel Surya GH Solar 30wp	1
2	<i>Solar Charge Controller</i> (SCC) 20A	1
3	Baterai Inforce 12V/9AH	1
4	Multimeter/Clampmeter	1
5	MCB DC	2
6	Kabel NYAF 1.5 mm	1
7	Skun	12
8	Panel Box	1
9	ESP32	1
10	Motor Servo MG995	2
11	<i>Real Time Clock</i> (RTC)	1
12	Kabel Jumper	3
13	PCB	1
14	Bearing Shaft 10mm	2
15	Sensor BH1750	1
16	LCD I2C 20x4	1
17	Mounting Bracket (Baja ringan)	1
18	Sensor DS18B20	1
19	Sensor INA219	1

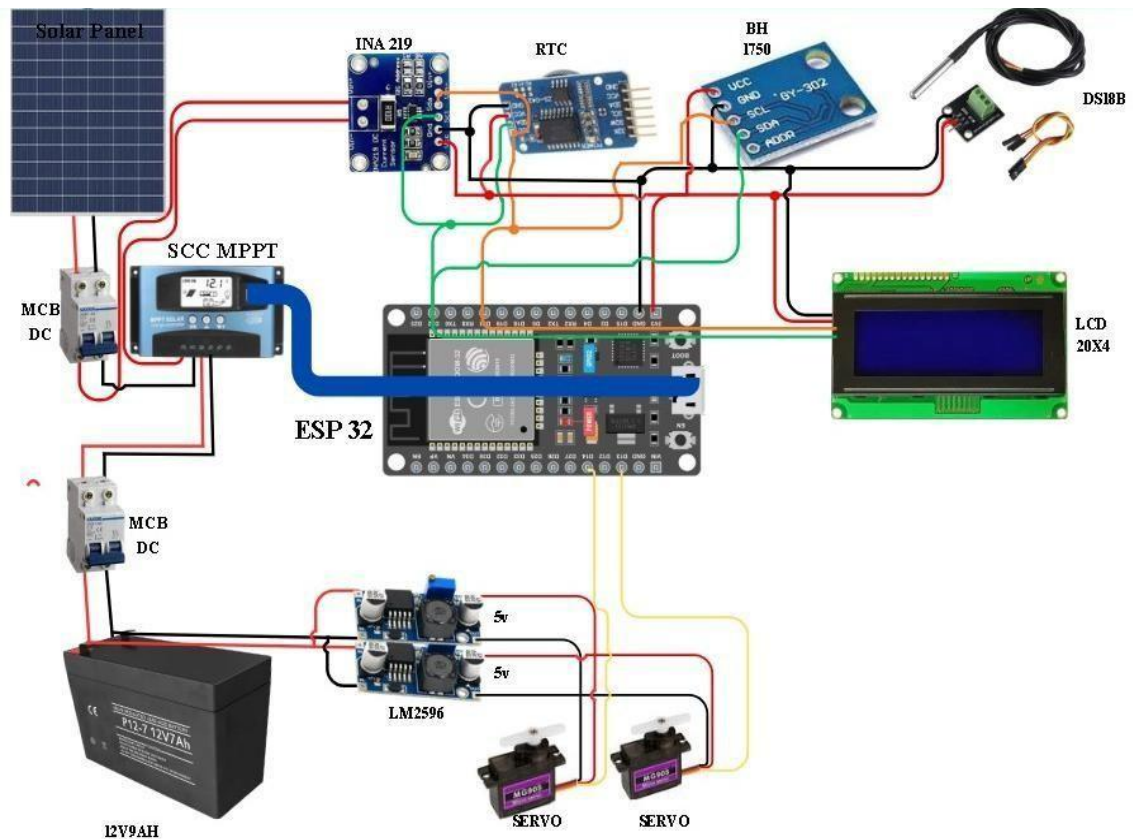
2.3 Blok Diagram Sistem



Gambar 1. Blok Diagram Sistem

Gambar 1 menunjukkan bahwa sistem *solar tracking* PLTS berbasis *Internet of Things (IoT)* dan *Machine Learning* dibagi menjadi tiga bagian utama, yaitu *input*, proses, dan *output*. Pada bagian *input* digunakan sensor INA219 untuk membaca tegangan, arus, dan daya panel surya, sensor RTC DS3231 untuk membaca waktu secara *real-time*, sensor BH1750 untuk mengukur intensitas cahaya matahari dalam satuan *lux*, serta sensor DS18B20 untuk memantau suhu panel surya. Seluruh data sensor kemudian dikirim ke ESP32 sebagai pusat kontrol untuk diproses menggunakan model *machine learning* XGBoost dan *Decision Tree* yang telah dilatih sebelumnya guna menentukan sudut optimal panel surya berdasarkan waktu, intensitas cahaya, dan suhu. Sistem diprogram menggunakan *Arduino IDE* serta dilengkapi koneksi *WiFi* dan *SCC* sebagai sumber daya utama sistem. Hasil prediksi sudut digunakan untuk menggerakkan servo horizontal dan servo vertikal agar panel surya selalu menghadap ke arah matahari secara optimal. Seluruh data *monitoring* ditampilkan secara *real-time* pada LCD 20x4, dikirim ke *Firestore RTDB* sebagai penyimpanan data *cloud*, dipantau melalui aplikasi *Blynk* pada *smartphone*, serta dicatat otomatis ke *Google Spreadsheet* melalui *App Script* untuk keperluan analisis data lebih lanjut.

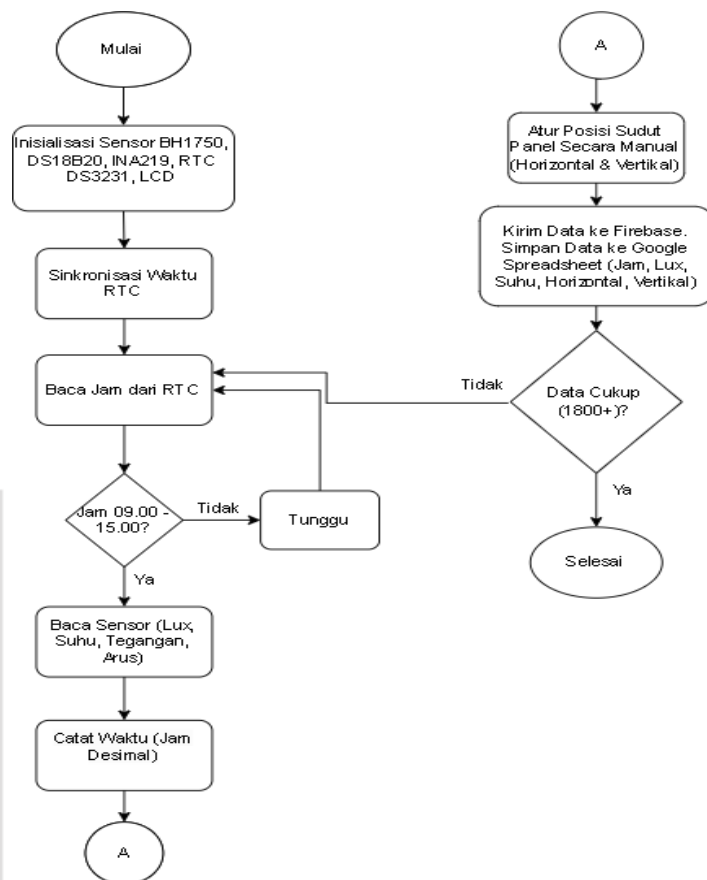
2.4 Perancangan *Hardware*



Gambar 2. Perancangan *Hardware*

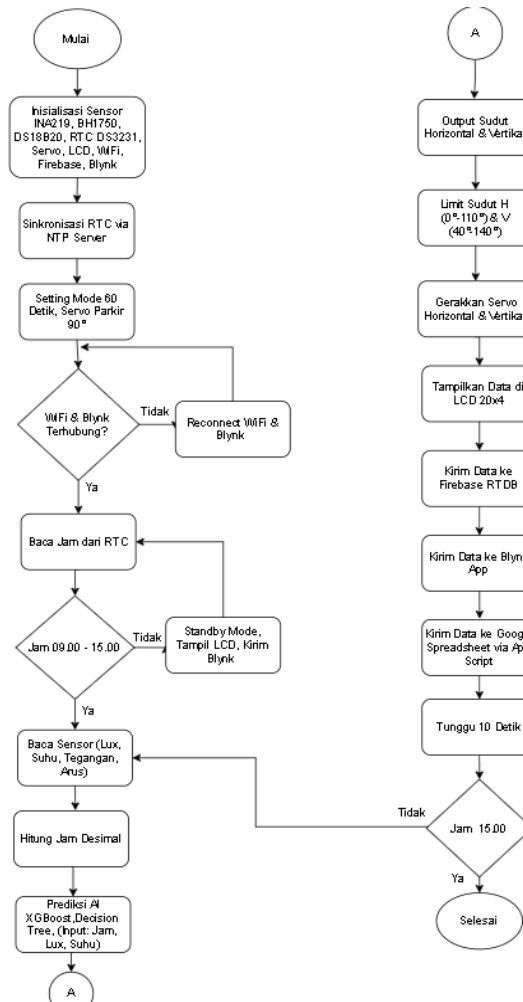
Gambar 2 menunjukkan perancangan *hardware* sistem optimasi posisi panel surya *dual-axis* berbasis *IoT*. Energi panel surya diatur oleh SCC MPPT untuk mengisi baterai sekaligus menyuplai daya ke seluruh sistem, dilengkapi MCB DC sebagai pengaman. ESP32 memperoleh daya langsung dari SCC melalui USB, sedangkan dua motor servo pada *dual-axis tracker* mendapat suplai dari baterai 12V yang diturunkan ke 5V melalui modul step-down LM2596. Untuk *monitoring*, digunakan sensor INA219 (tegangan, arus, dan daya), BH1750 (intensitas cahaya), DS18B20 (suhu panel), dan modul RTC (acuan waktu). Seluruh data sensor diproses oleh ESP32 untuk mengoptimalkan posisi panel menggunakan *algoritma machine learning*, yang kemudian menggerakkan servo sesuai hasil prediksi model. Data pengukuran ditampilkan pada LCD 20x4 dan dapat dipantau secara *real-time*.

2.5 Perancangan Software



Gambar 3. Diagram Alir Program

Gambar 3 menunjukkan proses pengambilan dataset yang dilakukan secara manual selama beberapa hari pada rentang waktu pukul 09.00 hingga 15.00. Sistem membaca data sensor BH1750 untuk intensitas cahaya, DS18B20 untuk suhu panel, dan INA219 untuk tegangan dan arus. Bersamaan dengan itu dicatat posisi sudut horizontal dan vertikal panel surya secara manual pada posisi optimal menghadap matahari. Seluruh data kemudian dikirim ke *Firestore RTDB* dan secara otomatis tersimpan ke *Google Spreadsheet* melalui *App Script* hingga terkumpul sebanyak 1.676 data lebih yang akan digunakan sebagai dataset *training model machine learning*.



Gambar 4. Diagram Alir Program

Gambar 4 menunjukkan alur kerja sistem PLTS *solar tracking* secara keseluruhan setelah model *Machine learning* diimplementasikan. Sistem diawali dengan inisialisasi seluruh kompo-nen dan sinkronisasi waktu RTC melalui NTP server. Kemudian sistem masuk ke *setting mode* selama 60 detik dengan servo berada di posisi 90°. Setelah itu sistem mengecek koneksi *WiFi* dan *Blynk*, jika tidak terhubung maka sistem melakukan *reconnect* secara otomatis. Sistem kemudian membaca jam dari RTC, jika berada di luar rentang 09.00 hingga 15.00 maka sistem masuk *standby mode*. Jika berada pada rentang waktu tersebut maka sistem membaca semua sensor setiap 10 detik, menghitung jam desimal, dan me-masukkan data ke model *Machine Learning* untuk mendapatkan prediksi sudut optimal. Servo horizontal dan vertikal kemudian digerakkan sesuai sudut hasil prediksi. *Data mon-itoring* ditampilkan pada LCD 20x4 dan dikirimkan secara bersamaan ke *Firebase RTDB*, *Blynk App*, serta *Google Spreadsheet* melalui *App Script*.

2.6 Pengambilan Dataset

Pengambilan dataset dilakukan secara manual selama beberapa hari untuk mendapatkan data terbaik yang merepresentasikan kondisi nyata pergerakan matahari. Setiap 10 detik sistem membaca data sensor BH1750 untuk intensitas cahaya, DS18B20 untuk suhu panel, dan INA219 untuk tegangan dan arus. Hal ini didasarkan pada kecepatan gerak semu matahari sebesar 15° per jam (Qomaruddin, M. N., Khairi, M. 2019) sehingga interval tersebut cukup untuk menangkap perubahan posisi matahari secara akurat. Bersamaan dengan pembacaan sensor, posisi sudut panel diatur secara manual pada posisi paling optimal menghadap matahari. Untuk sudut vertikal, panel digerakkan sebesar 4° setiap 14 menit mengikuti pergerakan elevasi matahari, sedangkan sudut horizontal disesuaikan agar panel selalu menghadap matahari dalam rentang 70° – 120° . Semua data kemudian dikirim ke Firebase RTDB dan tersimpan otomatis ke Google *Spreadsheet* lewat *App Script* hingga terkumpul sekitar 1676 data untuk training model *machine learning*. Data yang berhasil direkam sebanyak 1.676 sampel dari target 2.160 sampel. Selisih 484 data disebabkan oleh gangguan koneksi *wifi*, jeda pencatatan, dan baris yang mengandung nilai kosong yang dihapus pada tahap preprocessing.

Tabel 2. Struktur Dataset

No	Kolom	Sumber	Satuan
1	Jam	RTC DS3231	Desimal
2	Lux	BH1750	lux
3	Suhu	DS18B20	$^\circ\text{C}$
4	Tegangan	INA219	V
5	Arus	INA219	A
6	Daya	INA219	W
7	Horizontal	Manual	Derajat $^\circ$
8	Vertikal	Manual	Derajat $^\circ$

2.7 Proses Training ML (XGBoost $\rightarrow$ Decision Tree)

2.7.1 Penentuan *Feature* (X) dan *Label* (Y)

Sebelum proses pelatihan model machine learning dilakukan, terlebih dahulu ditentukan variabel yang digunakan sebagai fitur *input* (X) dan target *output*

(y). Pemilihan fitur *input* didasarkan pada hasil analisis korelasi yang telah dilakukan sebelumnya, serta mempertimbangkan relevansi fisik masing-masing variabel terhadap pergerakan posisi matahari.

Fitur *input* yang digunakan terdiri dari tiga variabel sensor, yaitu:

Tabel 3. Struktur variabel sensor

No	Variabel	Nama Kolom	Keterangan
1	Jam	Jam	Waktu pengukuran dalam format desimal, merepresentasikan posisi matahari secara temporal sepanjang hari
2	Intensitas Cahaya	<i>Lux</i>	Nilai intensitas cahaya matahari yang diterima panel surya, diukur menggunakan sensor BH1750 dalam satuan lux
3	Suhu Panel	Suhu	Suhu permukaan panel surya yang diukur menggunakan sensor DS18B20 dalam satuan °C

Ketiga fitur tersebut dipilih karena memiliki hubungan langsung terhadap posisi matahari. Variabel Jam merepresentasikan pergerakan matahari secara temporal, variabel *Lux* merepresentasikan intensitas radiasi surya yang diterima panel, dan variabel Suhu merepresentasikan kondisi termal panel akibat paparan sinar matahari.

Pemilihan fitur input dalam kode *Python* dilakukan sebagai berikut:

```
python
```

```
X = df[['Jam', 'Lux', 'Suhu']]
```

Target output model terdiri dari dua variabel sudut servo, yaitu:

Tabel 4. Struktur variabel sudut servo

No	Variabel	Nama Kolom	Keterangan
1	Sudut Horizontal	Horizontal	Sudut pergerakan servo horizontal untuk mengikuti posisi azimuth matahari
2	Sudut Vertikal	Vertical	Sudut pergerakan servo vertikal untuk mengikuti elevasi matahari

Target *output* merupakan variabel yang ingin diprediksi oleh model

machine learning. Pada penelitian ini, target *output* adalah sudut pergerakan dua buah servo motor yang berfungsi menggerakkan panel surya agar selalu menghadap ke arah matahari.

Penentuan target *output* dalam kode *Python* dilakukan sebagai berikut:

```
python
y_horizontal = df['Horizontal']
y_vertical   = df['Vertical']
```

Gambar 5. kode *Python*

2.7.2 Pembagian Dataset (*Train, Validation, Test Split*)

Setelah fitur *input* dan target *output* ditentukan, dataset dibagi menjadi tiga bagian sebelum digunakan untuk pelatihan model, yaitu data *training*, data *validasi*, dan data *testing*. Pembagian ini dilakukan untuk mengevaluasi kemampuan model secara objektif dan mencegah terjadinya *overfitting*, yaitu kondisi di mana model hanya mampu mengenali data latih tetapi gagal melakukan generalisasi pada data baru

a. Metode Pembagian

Pembagian *dataset* dilakukan dalam dua tahap sebagai berikut:

Tahap 1 — *Split Pertama (Train 70% : Temp 30%)*

Pada tahap pertama, seluruh dataset dibagi menjadi data *training* sebesar 70% dan data sementara (*temp*) sebesar 30% menggunakan parameter *test_size=0.30*.

Tahap 2 — *Split Kedua (Validation 15% : Test 15%)*

Pada tahap kedua, data sementara (*temp*) sebesar 30% dibagi kembali menjadi dua bagian yang sama besar menggunakan parameter *test_size=0.50*, sehingga menghasilkan data *validasi* 15% dan data *testing* 15% dari total keseluruhan *dataset*.

b. Hasil Pembagian *Dataset*

Berdasarkan hasil eksekusi program pada Google Colaboratory, total *dataset* yang digunakan berjumlah 1.676 data. Hasil pembagian *dataset* Data dibagi menjadi tiga subset: 70% data *training* (1.172 data) untuk melatih model

mengenali pola hubungan antara fitur *input* (Jam, *Lux*, Suhu) dan *output* (Horizontal, Vertikal); 15% data validasi (251 data) untuk memantau *overfitting* selama *training* melalui parameter *eval_set* pada *XGBoost*, di mana peningkatan loss validasi saat loss training terus menurun mengindikasikan *overfitting*; serta 15% data *testing* (252 data) sebagai *unseen data* yang hanya digunakan pada evaluasi akhir untuk mengukur kemampuan generalisasi model yang sesungguhnya.

2.7.3 Pelatihan Model *XGBoost* (*eXtreme Gradient Boosting*)

XGBoost (*eXtreme Gradient Boosting*) merupakan algoritma machine learning berbasis ensemble yang menggunakan metode *Gradient Boosting*. Algoritma ini membangun sejumlah pohon keputusan (*decision tree*) secara berurutan (*sequential*), di mana setiap pohon berusaha memperbaiki kesalahan prediksi pohon sebelumnya. Metode ini dikenal memiliki performa tinggi dan telah banyak digunakan dalam berbagai kompetisi dan penelitian di bidang kecerdasan buatan.

Pada penelitian ini, *XGBoost* digunakan untuk memprediksi dua target *output*, yaitu sudut Horizontal dan sudut Vertikal panel surya berdasarkan tiga fitur *input*: Jam, *Lux*, dan Suhu. Dua model *XGBoost* dilatih secara terpisah, masing-masing untuk sumbu Horizontal dan Vertikal.

a. Konsep Dasar *XGBoost*

XGBoost bekerja dengan cara membangun model secara bertahap (*additive manner*). Setiap iterasi menambahkan satu pohon baru yang mengoreksi *residual error* dari model sebelumnya. Prediksi akhir merupakan penjumlahan dari seluruh pohon yang telah dibangun.

Secara matematis, prediksi model pada iterasi ke-*t* dapat dituliskan sebagai persamaan (1) berikut:

$$\hat{y}(t) = \hat{y}(t-1) + \eta \cdot f_t(x) \quad (1)$$

Keterangan:

- $\hat{y}(t)$ = prediksi model pada iterasi ke-*t*
- $\hat{y}(t-1)$ = prediksi model pada iterasi sebelumnya
- η = learning rate (laju pembelajaran)
- $f_t(x)$ = pohon keputusan baru yang ditambahkan pada iterasi ke-*t*
- x = vektor fitur input (Jam, *Lux*, Suhu)

b. Fungsi Objektif XGBoost

XGBoost mengoptimalkan fungsi objektif yang terdiri dari dua komponen, yaitu *loss function* dan regularisasi. Fungsi objektif yang diminimalkan pada setiap iterasi adalah:

$$\text{Obj}(t) = \sum L(y_i, \hat{y}_i) + \sum \Omega(f_k) \quad (2)$$

$$\Omega(f) = \gamma T + (1/2)\lambda ||w||^2$$

Keterangan:

$L(y_i, \hat{y}_i)$ = *loss function* (MSE untuk regresi): mengukur selisih prediksi dengan nilai aktual

$\Omega(f_k)$ = regularisasi untuk mencegah *overfitting*

T = jumlah daun (*leaf*) pada pohon

γ = parameter penalti jumlah daun

λ = parameter regularisasi L2 pada bobot daun

w = bobot (skor) pada setiap daun pohon

XGBoost mengoptimalkan fungsi objektif yang terdiri dari dua komponen utama yaitu *loss function* dan *regularisasi*. Fungsi objektif yang diminimalkan pada setiap iterasi adalah $\text{Obj}(t) = \sum L(y_i, \hat{y}_i) + \sum \Omega(f_k)$, dimana $L(y_i, \hat{y}_i)$ merupakan *loss function* yang menggunakan *Mean Squared Error (MSE)* untuk kasus regresi yang berfungsi mengukur selisih antara nilai prediksi dengan nilai aktual, sedangkan $\Omega(f_k)$ merupakan komponen regularisasi yang berfungsi mencegah terjadinya *overfitting* pada model.

Regularisasi tersebut diformulasikan sebagai $\Omega(f) = \gamma T + (1/2)\lambda ||w||^2$, dimana T merupakan jumlah daun (*leaf*) pada pohon keputusan, γ merupakan parameter penalti yang mengontrol jumlah daun agar pohon tidak tumbuh terlalu kompleks, λ merupakan parameter regularisasi L2 yang diterapkan pada bobot daun, dan w merupakan bobot atau skor pada setiap daun pohon. Dengan adanya komponen regularisasi ini, XGBoost mampu menghasilkan model yang lebih general dan tidak mudah *overfitting* meskipun dilatih dengan jumlah iterasi yang besar.

c. Parameter Model XGBoost

Implementasi parameter XGBoost dalam kode *Python* adalah sebagai berikut:

```
xgb_horizontal = XGBRegressor(  
    n_estimators=300,  
    max_depth=5,  
    learning_rate=0.05,  
    subsample=0.8,  
    colsample_bytree=0.8,  
    random_state=42,  
    eval_metric='rmse',  
    early_stopping_rounds=30  
)  
  
xgb_vertical = XGBRegressor(  
    n_estimators=1000,  
    max_depth=5,  
    learning_rate=0.05,  
    subsample=0.8,  
    colsample_bytree=0.8,  
    random_state=42,  
    eval_metric='rmse',  
    early_stopping_rounds=20  
)
```

Gambar 6. kode Python

Gambar 6 Model XGBoost dibangun secara terpisah. Model horizontal-menggunakan *n_estimators*=300 dengan *early_stopping_rounds*=30, sedangkan model vertical menggunakan *n_estimators*=1000 dengan *early_stopping_rounds*=20. Kedua model berbagi parameter yang sama, yaitu *max_depth*=5, *learning_rate*=0.05, *subsample*=0.8, *colsample_bytree*=0.8, *random_state*=42, dan *eval_metric*='rmse'. Parameter *n_estimators* menentukan jumlah pohon keputusan yang dibangun, *max_depth*=5 membatasi kedalaman tiap pohon untuk mencegah *overfitting*, dan *learning_rate*=0.05 mengatur seberapa besar kontribusi setiap pohon terhadap model akhir. Parameter *subsample*=0.8 dan *colsample_bytree*=0.8 berarti setiap pohon hanya menggunakan 80% data dan 80% fitur secara acak sehingga model lebih *robust*. *random_state*=42 memastikan hasil yang dapat direproduksi, *eval_metric*='rmse' menjadikan *Root Mean Square Error* sebagai metrik evaluasi, sedangkan *early_stopping_rounds* menghentikan proses *training* secara otomatis apabila performa pada data validasi tidak membaik setelah sejumlah iterasi tertentu.

2.3.4 Pelatihan Model Decision Tree

Decision Tree (pohon keputusan) merupakan algoritma *machine learning* yang membangun model berupa struktur pohon bercabang. Setiap node internal pada pohon merepresentasikan kondisi pengujian terhadap suatu fitur input, setiap cabang merepresentasikan hasil pengujian tersebut, dan setiap node daun (*leaf*)

merepresentasikan nilai prediksi *output*.

Pada penelitian ini, *Decision Tree* digunakan sebagai model alternatif yang dapat diekspor secara langsung ke dalam format kode C++ *IF-ELSE* untuk diimplementasikan pada mikrokontroler Arduino/ESP32. Dua model *Decision Tree* dilatih secara terpisah untuk sudut Horizontal dan Vertical.

a. Konsep Dasar *Decision Tree* (CART)

Algoritma *Decision Tree* yang digunakan adalah *CART* (*Classification and Regression Trees*). Pada setiap node, algoritma mencari split terbaik berdasarkan fitur dan *threshold* yang meminimalkan nilai impuritas. Untuk kasus regresi, kriteria yang digunakan adalah *Mean Squared Error* (MSE):

$$\text{MSE_split} = (1/n_L) \cdot \sum (y_i - \bar{y}_L)^2 + (1/n_R) \cdot \sum (y_i - \bar{y}_R)^2 \quad (3)$$

Keterangan:

MSE_split	=	nilai MSE setelah dilakukan <i>split</i>
n_L	=	jumlah sampel di cabang kiri
n_R	=	jumlah sampel di cabang kanan
$\bar{y}_L$	=	nilai rata-rata target di cabang kiri
$\bar{y}_R$	=	nilai rata-rata target di cabang kanan
y_i	=	nilai target data ke-i

Algoritma *CART* bekerja secara *rekursif* dengan membagi data menjadi dua cabang yaitu cabang kiri dan cabang kanan berdasarkan nilai *threshold* fitur tertentu. Proses pembagian ini terus dilakukan hingga mencapai kondisi berhenti seperti kedalaman maksimum pohon tercapai, jumlah sampel minimum di setiap node terpenuhi, atau tidak ada lagi penurunan *impuritas* yang signifikan.

b. Kriteria Split: Information Gain

Pemilihan split terbaik dilakukan berdasarkan penurunan impuritas (*impurity decrease*). Sebuah split hanya akan dilakukan apabila memenuhi kondisi berikut:

$$\Delta I = (n/N) \cdot I(\text{node}) - (n_L/N) \cdot I(L) - (n_R/N) \cdot I(R) \quad (4)$$

Keterangan:

ΔI = penurunan impuritas akibat *split*
 n = jumlah sampel di node saat ini
 N = total sampel training
 $I(\text{node})$ = nilai impuritas node sebelum *split*
 $I(L), I(R)$ = nilai impuritas cabang kiri dan kanan
 $\text{min_impurity_decrease}$ = ambang batas minimum penurunan impuritas (0.01 untuk Vertical)

Apabila nilai penurunan *impuritas* (ΔI) yang dihasilkan dari suatu *split* lebih besar atau sama dengan nilai $\text{min_impurity_decrease}$, maka *split* tersebut dianggap signifikan dan akan dilanjutkan. Sebaliknya apabila nilai ΔI lebih kecil dari ambang batas maka *split* tidak dilakukan dan node tersebut menjadi node daun (*leaf node*). Pada model vertikal digunakan nilai $\text{min_impurity_decrease}$ sebesar 0.01 untuk mengontrol kompleksitas pohon dan mencegah terjadinya *overfitting*.

c. Parameter Model *Decision Tree*

Parameter yang digunakan dalam pelatihan model *Decision Tree* pada penelitian ini ditampilkan pada berikut. Perlu diperhatikan bahwa parameter model Horizontal dan Vertical berbeda karena model Vertical memerlukan penanganan khusus untuk mencegah *overfitting*.

Implementasi parameter *Decision Tree* dalam adalah sebagai berikut:

```

# Horizontal -
dt_horizontal = DecisionTreeRegressor(
    max_depth=6,
    min_samples_leaf=10,
    random_state=42
)

# Vertical -
dt_vertical = DecisionTreeRegressor(
    max_depth=4,
    min_samples_leaf=20,
    min_samples_split=20,
    min_impurity_decrease=0.01,
    random_state=42
)
  
```

Gambar 7. kode Python

Model *Decision Tree* juga dibangun secara terpisah untuk masing-masing sumbu. Model horizontal menggunakan $\text{max_depth}=6$ dan $\text{min_samples_leaf}=10$, sedangkan model vertikal menggunakan $\text{max_depth}=4$,

min_samples_leaf=20, *min_samples_split*=20, dan *min_impurity_decrease*=0.01. Kedua model menggunakan *random_state*=42 untuk memastikan hasil yang dapat direproduksi. Parameter *max_depth* membatasi kedalaman pohon untuk mencegah *overfitting*, dengan nilai yang lebih kecil pada model vertikal menunjukkan struktur pohon yang lebih sederhana. *min_samples_leaf* menentukan jumlah minimum data pada setiap simpul daun, *min_samples_split* menetapkan jumlah minimum data yang diperlukan untuk membelah sebuah simpul, sedangkan *min_impurity_decrease* memastikan pembelahan hanya dilakukan apabila terdapat penurunan impuritas (*impurity*) yang signifikan. Parameter-parameter ini secara keseluruhan berfungsi mengontrol kompleksitas model agar tetap ringkas dan dapat diterapkan pada perangkat *embedded*

2.8 Integrasi Firebase & Blynk & Google Spreadsheet

A. Firebase Realtime Database

Firebase Realtime Database merupakan layanan basis data *cloud* dari Google yang memungkinkan penyimpanan dan sinkronisasi data secara real-time. Pada sistem ini, Firebase digunakan untuk menyimpan data log yang dikirimkan ESP32 secara berkala, mencakup parameter *timestamp*, tanggal, waktu, intensitas cahaya (*lux* dan *lux_input*), tegangan, arus, daya, suhu (*suhu* dan *suhu_input*), posisi horizontal dan vertikal aktuator, target sudut (*target_h* dan *target_v*), serta status validasi sensor. Pengiriman data dilakukan melalui objek *FirebaseJson* menggunakan fungsi *json.set()* untuk menyusun pasangan kunci-nilai secara terstruktur, kemudian dikirimkan ke path */panel/log/[tanggal]* sehingga data terkelompok per hari dan memudahkan monitoring historis kinerja panel surya. Implementasi kode pengiriman dapat dilihat pada Gambar 9.

```

// FirebaseJson
// modeStr = "panel"

FirebaseJson json;
json.set("timestamp", now.unixtime());
json.set("hari", hari);
json.set("tanggal", tanggal);
json.set("waktu", waktu);
json.set("mode", modeStr);
json.set("lux", lux);
json.set("lux_input", luxFiltered);
json.set("tegangan", voltage);
json.set("arus", current);
json.set("daya", power);
json.set("suhu", suhu);
json.set("suhu_input", suhuFiltered);
json.set("horizontal", sudutHorizontal);
json.set("vertical", sudutVertical);
json.set("target_h", sudutHorizTarget);
json.set("target_v", sudutVertTarget);
json.set("lux_valid", luxValid);
json.set("suhu_valid", suhuValid);

// Kirim data ke Firebase

```

Gambar 8 Kode Pengiriman Data ke Firebase Realtime Databas

B. Blynk IoT Platform

Blynk merupakan *platform IoT* yang menyediakan antarmuka pemantauan perangkat secara nirkabel melalui aplikasi mobile maupun *dashboard web*. Pada sistem ini, *Blynk* digunakan untuk menampilkan data sensor secara *real-time* melalui *virtual pin* yang diperbarui setiap 5 detik menggunakan fungsi *millis()* agar tidak menghambat eksekusi tugas lain pada *mikrokontroler*. Data yang dikirimkan mencakup delapan parameter yang dipetakan ke *virtual pin* V0–V7, yaitu intensitas cahaya (V0), sudut horizontal (V1), tegangan (V2), arus (V3), daya (V4), waktu (V5), suhu (V6), dan sudut vertikal (V7), menggunakan fungsi *Blynk.virtualWrite()* seperti terlihat pada Gambar 10.

```
// =====  
// [TASK 5] BLYNK UPDATE - tiap 5 detik  
// =====  
  
if (millis() - lastBlynk >= INTERVAL_BLYNK) {  
  
    lastBlynk = millis();  
  
    char waktuBuf[10];  
    sprintf(waktuBuf, "%02d:%02d:%02d", now.hour(), now.minute(), now.second());  
  
    Blynk.virtualWrite(V0, lux);  
    Blynk.virtualWrite(V1, sudutHorizontal);  
    Blynk.virtualWrite(V2, voltage);  
    Blynk.virtualWrite(V3, current);  
    Blynk.virtualWrite(V4, power);  
    Blynk.virtualWrite(V5, waktuBuf);  
    Blynk.virtualWrite(V6, suhu);  
    Blynk.virtualWrite(V7, sudutVertikal);  
  
}
```

Gambar 9 Kode Pengiriman Data ke Platform Blynk (Virtual Pin V0–V7)

C. Google Spreadsheet (Google Apps Script)

Google *Spreadsheet* diintegrasikan ke sistem melalui Google *Apps Script* menggunakan fungsi *ambilDataFirebase()* yang bertugas mengambil data log dari *Firebase* dan menuliskannya ke lembar kerja untuk keperluan pencatatan dan analisis historis. Pengambilan data dilakukan dengan mengirimkan permintaan *HTTP GET* ke *endpoint Firebase* (<https://plts11-default-rtdb.firebaseio.com/panel/log.json>) melalui *UrlFetchApp*, kemudian respons *JSON* diuraikan menggunakan *JSON.parse()* dan dituliskan ke lembar kerja aktif melalui *SpreadsheetApp.getActiveSpreadsheet().getActiveSheet()*. Implementasi fungsi tersebut dapat dilihat pada Gambar 11.

```

1 function ambilDataFirebase() {
2
3   var url =
4     "https://p1ts11-default-rtdb.firebaseio.com/panel/log.json";
5
6   try {
7
8     var response =
9       UrlFetchApp.fetch(url);
10
11     var data =
12       JSON.parse(response.getContentText());
13
14     if (!data) return;
15
16     var sheet =
17       SpreadsheetApp
18         .getActiveSpreadsheet()
19         .getActiveSheet();

```

Gambar 10 Kode Google Apps Script untuk Mengambil Data dari Firebase ke Google Spreadsheet

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk menganalisis kinerja sistem optimasi posisi panel surya *dual-axis* berbasis *Internet of Things (IoT)* dan *machine learning* menggunakan algoritma XGBoost dalam meningkatkan efisiensi daya keluaran panel surya. Pengujian meliputi hasil *training* model XGBoost seperti *loss per epoch*, akurasi R^2 , *early stopping*, dan pengecekan *overfitting*. Selain itu, dilakukan pengujian struktur *Decision Tree* hasil penyederhanaan model XGBoost, distribusi error dan *residual plot*, *correlation heatmap*, implementasi *hardware*, pergerakan servo horizontal dan vertikal, pengujian sensor BH1750, INA219, dan DS18B20, tampilan LCD serta *Blynk*, dan pengiriman data ke *Firebase* untuk mengetahui performa sistem secara keseluruhan.

3.1 Overfitting Check

Setelah proses pelatihan model selesai, dilakukan evaluasi terhadap seluruh model menggunakan tiga metrik, yaitu R^2 Score, *Mean Absolute Error (MAE)*, dan *Root Mean Squared Error (RMSE)*. Evaluasi dilakukan pada tiga subset data secara terpisah, yaitu data training, data validasi, dan data *testing*, untuk memastikan model tidak hanya akurat pada data latih tetapi juga mampu melakukan generalisasi yang baik pada data yang belum pernah dilihat sebelumnya.

Tabel 5. Hasil Evaluasi Model XGBoost dan *Decision Tree*

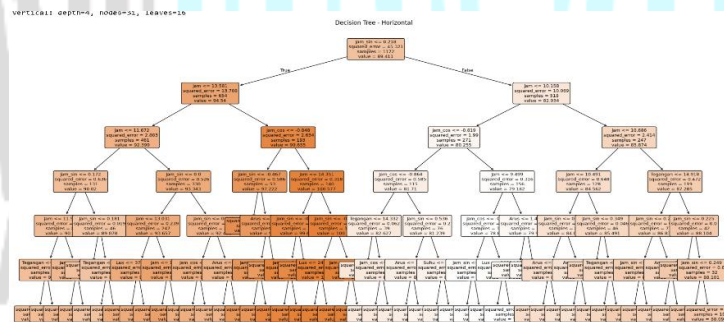
No	Model	Train R^2	Val R^2	Test R^2	MAE	RMSE	Subset
1	XGB Horizontal	0,9789	0,9777	0,9642	1.1137	1.4169	Test
2	XGB Vertical	0,9712	0,9725	0,9657	3.6293	4.7133	Test
3	DT Horizontal	0,9790	0,9778	0,9793	0,8264	1.0777	Test
4	DT Vertical	0,9684	0,9704	0,9696	3.8287	4.7696	Test

Seluruh model yang dilatih menunjukkan performa sangat baik tanpa *overfitting*, dibuktikan dengan selisih *Train R²* dan *Test R²* yang sangat kecil pada semua model (maksimum 0,0055 pada *XGBoost Horizontal*, jauh di bawah ambang 0,05). Model *XGBoost Horizontal* menghasilkan *Test R²* 0,9939 dengan *MAE* 0,2772° dan *RMSE* 0,5851°, sementara *XGBoost Vertical* menghasilkan *Test R²* 0,9964 dengan *MAE* 0,8603° dan *RMSE* 1,6317° — nilai *MAE/RMSE* Vertical yang lebih besar semata-mata mencerminkan rentang sudut Vertical (40°–140°) yang lebih lebar. Menariknya, model *Decision Tree Horizontal* justru menghasilkan *MAE* 0,1478° yang lebih kecil dari *XGBoost Horizontal*, menunjukkan bahwa model yang lebih sederhana pun mampu menghasilkan prediksi yang sangat konsisten dan layak diimplementasikan pada *Arduino*.

3.4. Struktur dan Visualisasi *Decision Tree*

a. Struktur *Decision Tree Horizontal*

Visualisasi struktur *Decision Tree Horizontal* ditampilkan pada Gambar 11 berikut. Pohon ini dibangun dengan parameter *max_depth*=6 dan *min_samples_leaf*=10, menghasilkan pohon dengan kedalaman maksimum 6 level yang terdiri dari sejumlah node internal dan node daun (leaf).



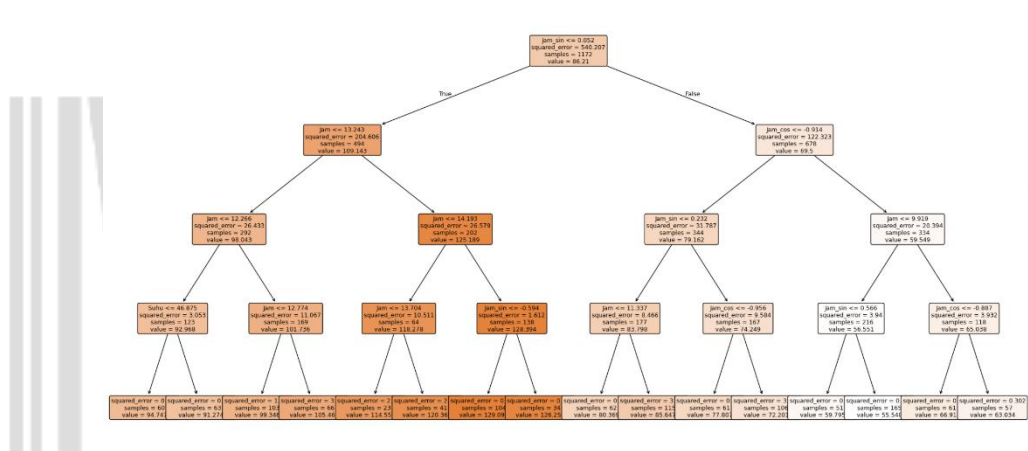
Gambar 11 Struktur *Decision Tree Horizontal*

Berdasarkan Gambar 11, *node root* (akar pohon) menggunakan kondisi $\text{Jam} \leq 11,165$ sebagai pemisah pertama, dengan nilai rata-rata prediksi awal sebesar 89,41° dari 1.172 sampel *training*. Hal ini secara langsung mengkonfirmasi bahwa variabel *Jam* merupakan fitur yang paling dominan dalam menentukan sudut Horizontal, sesuai dengan karakteristik fisik matahari yang bergerak dari timur ke barat secara linier terhadap waktu.

Cabang *True* ($\text{Jam} \leq 11,165$) mengarah ke sub-pohon kiri yang merepresentasikan posisi matahari pada pagi hingga siang hari, dengan nilai prediksi berkisar antara 76° – 92° . Cabang *False* ($\text{Jam} > 11,165$) mengarah ke sub-pohon kanan yang merepresentasikan posisi matahari pada siang hingga sore hari, dengan nilai prediksi berkisar antara 92° – 104° . Pada level kedalaman yang lebih dalam, variabel *Lux* dan Suhu mulai digunakan sebagai kondisi percabangan untuk memperhalus prediksi pada rentang waktu tertentu.

b. Struktur Decision Tree Vertical

Visualisasi struktur *Decision Tree* Vertical ditampilkan pada Gambar 16 berikut. Pohon ini dibangun dengan parameter $\text{max_depth}=4$, $\text{min_samples_leaf}=20$, $\text{min_samples_split}=20$, dan $\text{min_impurity_decrease}=0.01$ untuk mencegah *overfitting*.



Gambar 12 Struktur Decision Tree Vertical

Berdasarkan Gambar 12, *node root* pohon Vertical menggunakan kondisi $\text{Jam} \leq 11,801$ sebagai pemisah utama, dengan nilai rata-rata prediksi awal sebesar $86,222^{\circ}$ dari 1.172 sampel *training*. Sama seperti model Horizontal, variabel *Jam* mendominasi sebagai fitur utama pada *node root*, mengindikasikan bahwa elevasi matahari (sudut Vertical) juga sangat dipengaruhi oleh waktu dalam sehari.

Struktur pohon Vertical tampak lebih sederhana dibandingkan Horizontal, dengan kedalaman maksimum hanya 4 level dan jumlah daun yang lebih sedikit. Setiap *node daun* pada $\text{min_samples_leaf}=20$, sehingga setiap prediksi didasarkan pada setidaknya 20 data *training* yang representatif. Nilai pada *node daun* berkisar dari $46,157^{\circ}$ (pagi hari) hingga $140,0^{\circ}$ (tengah hari), mencerminkan perubahan elevasi matahari sepanjang hari.

3.2 Perbandingan Sistem Sebelum dan Sesudah *Machine learning*

Pengujian perbandingan dilakukan untuk mengetahui perbedaan kinerja sistem PLTS *solar tracking* sebelum dan sesudah implementasi model *Decision Tree*. Seluruh data diproses menggunakan ESP32 dan ditampilkan secara *real-time* melalui *platform IoT*. Algoritma *XGBoost* dilatih terlebih dahulu sebagai *teacher model* karena memiliki akurasi tinggi, kemudian prediksi *XGBoost* digunakan sebagai *soft labels* (prediksi dari model *XGBoost* untuk melatih model *Decision Tree* melalui proses *knowledge distillation*. Pendekatan ini menghasilkan model *Decision Tree* yang mewarisi pola prediksi *XfGBoost* namun tetap ringkas dan dapat diterapkan langsung pada *Arduino IDE*.

Dataset Train

[Link Google Drive: https://bit.ly/3Qb0mqE](https://bit.ly/3Qb0mqE)

Dataset Test

[Link Google Drive: https://bit.ly/4wZXcqa](https://bit.ly/4wZXcqa)

[HSIL](https://surl.li/iyisdp)

<https://surl.li/iyisdp>

a. Rumus Perhitungan Rasio Daya terhadap Intensitas Cahaya

Untuk mengevaluasi kinerja sistem secara objektif terlepas dari perbedaan kondisi cuaca antar hari pengujian, digunakan perhitungan rasio daya terhadap intensitas cahaya (*W/lux*). Rasio ini menunjukkan seberapa besar daya yang mampu dihasilkan panel surya per satuan intensitas cahaya yang diterima.

$$\text{Rasio (W/lux)} = \text{Rata-rata Daya (W)} / \text{Rata-rata Intensitas Cahaya (lux)} \quad (5)$$

Sebelum :

$$\text{Rasio} = 17.9518 / 30.259 = 0.000593 \text{ W/lux}$$

Sesudah :

$$\text{Rasio} = 16.3436 / 25.175 = 0.000649 \text{ W/lux}$$

Peningkatan Rasio:

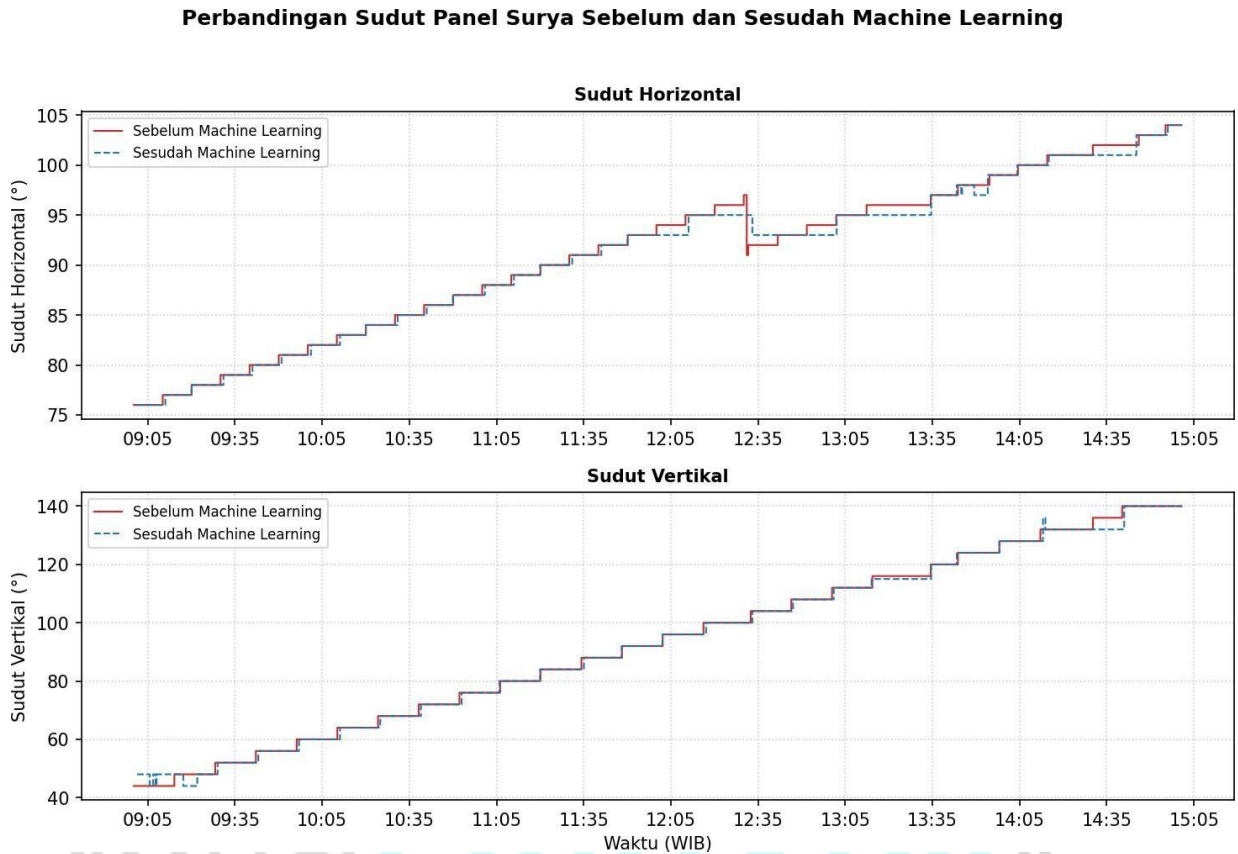
$$\text{Peningkatan} = ((0.000649 - 0.000593) / 0.000593) \times 100\% = 13.37\%$$

Berdasarkan perhitungan di atas, rasio daya terhadap intensitas cahaya meningkat dari 0.000593 *W/lux* menjadi 0.000649 *W/lux* atau naik sebesar 13.37%. Hal ini menunjukkan bahwa sistem sesudah *Machine learning* mampu menghasilkan daya lebih optimal

meskipun intensitas cahaya yang diterima lebih rendah, karena posisi panel sudah diatur secara tepat menggunakan model *Decision Tree*.

b. Perbandingan Sudut

Sebelum dan Sesudah Penerapan *Machine Learning* — Pukul 09.00 s.d. 15.00WIB



Gambar 13. Grafik Perbandingan Sudut Horizontal dan Vertikal Panel Surya per Detik Sebelum (Merah) dan Sesudah (Biru) Penerapan Machine Learning

Grafik menampilkan perubahan sudut panel surya dari pukul 09.00 hingga 15.00 WIB untuk dua parameter: Sudut Horizontal dan Sudut Vertikal. Garis merah merepresentasikan kondisi sebelum penerapan ML, sedangkan garis biru putus-putus merepresentasikan kondisi sesudah penerapan ML. Pada Sudut Horizontal, kedua kondisi menunjukkan pola kenaikan serupa dari sekitar 76° menuju 104° , mengikuti pergerakan matahari dari timur ke barat. Sesudah penerapan ML, garis biru terlihat sedikit lebih tinggi pada rentang pukul 11.00–14.00 dengan selisih sekitar $1\text{--}2,5^\circ$, menunjukkan sistem lebih akurat dalam menyesuaikan posisi azimuth panel.

Pada Sudut Vertikal, perbedaan yang lebih signifikan terlihat antara kedua kondisi. Sesudah ML, kenaikan sudut terlihat lebih cepat dan responsif terutama pada sore hari (pukul 13.00–15.00) dengan selisih sekitar 3–4°, mengindikasikan ML berhasil mengoptimalkan penjejakan elevasi sehingga panel lebih cepat menyesuaikan kemiringannya. Secara keseluruhan, kedua sudut bergerak secara bertahap mengikuti pola pergerakan matahari yang teratur, dengan perubahan terjadi setiap beberapa detik sesuai resolusi data yang direkam.

3.3 Hasil Implementasi *Hardware*

Sistem PLTS telah berhasil dirancang dan diimplementasikan dalam bentuk prototipe alat yang terdiri dari panel surya, rangka *dual-axis*, box kontrol, dan seluruh komponen elektronik. Alat dirancang agar dapat beroperasi secara outdoor dengan struktur rangka besi yang kokoh dan box kontrol yang tertutup rapat untuk melindungi komponen elektronik dari cuaca. Pengujian dilakukan di area terbuka agar panel surya dapat menerima cahaya matahari langsung.

a. Tampilan Keseluruhan Alat



(1)



(3)



(2)



(4)

Gambar 14.(1) Tampilan Keseluruhan Alat PLTS (Indoor), (2) Tampilan Alat saat Pengujian Outdoor, (3) Tampilan Panel Surya dari Atas, (4) Tampilan Dalam Box Kontrol

Gambar (1) menunjukkan tampilan keseluruhan sistem PLTS dalam ruangan, dengan panel surya terpasang pada rangka *dual-axis* yang memungkinkan pergerakan dua sumbu (horizontal dan vertikal), box kontrol berwarna abu-abu di bagian bawah rangka, serta sensor BH1750 di atas panel untuk membaca intensitas cahaya. Gambar (3) menampilkan panel surya dari atas saat berada di area *outdoor*, di mana sensor BH1750 terlihat terpasang di sudut kanan atas panel dan berfungsi membaca intensitas cahaya matahari secara *real-time* setiap 10 detik. Gambar (2) menunjukkan sistem saat pengujian *outdoor* berlangsung, dengan panel surya yang telah bergerak secara otomatis mengikuti posisi matahari berdasarkan prediksi model *Decision Tree* pada rentang pukul 09.00–15.00 WIB dengan interval pencatatan 10 detik. Gambar (4) menampilkan bagian dalam box kontrol yang berisi *Solar Charge Controller* (SCC) dengan pembacaan tegangan baterai 12,3V, MCB DC sebagai pengaman, terminal block sebagai penghubung kabel, ESP32 sebagai mikrokontroler utama, modul INA219 untuk pembacaan tegangan dan arus, serta RTC DS3231 untuk pembacaan waktu — seluruh komponen tersusun rapi dalam satu box kontrol.

b. Kalibrasi Sensor BH1750 (Intensitas Cahaya)

Kalibrasi sensor BH1750 dilakukan dengan membandingkan pembacaan sensor terhadap luxmeter standar pada lima kondisi intensitas cahaya yang berbeda mulai dari intensitas rendah hingga tinggi.

Tabel 6. Hasil Kalibrasi Sensor BH1750

No	Luxmeter (lux)	BH1750 (lux)	Selisih (lux)	Error (%)
1	27.092	26.762	330	1.22
2	30.230	29.853	377	1.25
3	38.160	37.792	368	0.96
4	38.466	38.295	171	0.44
5	37.065	36.958	107	0.29
Rata-rata	-	-	271	0.83

$$\text{Presentase Error} = \left(\frac{\text{Nilai Luxtometer} - \text{Nilai Sensor}}{\text{Nilai Luxtometer}} \right) \times 100\% \quad (6)$$

Berdasarkan hasil kalibrasi sensor BH1750 pada Tabel 6, pengujian menunjukkan bahwa hasil pembacaan sensor BH1750 memiliki selisih yang kecil dibandingkan dengan luxmeter standar. Nilai *error* berkisar antara 0.29% hingga 1.25% dengan rata-

rata *error* sebesar 0.83%. Nilai *error* yang kecil ini masih berada dalam batas toleransi alat ukur yang wajar, sehingga sensor BH1750 dinyatakan layak dan akurat digunakan dalam sistem pengukuran intensitas cahaya matahari.

c. Kalibrasi Sensor INA219 (Tegangan, Arus, dan Daya)

Kalibrasi sensor INA219 dilakukan dengan membandingkan pembacaan tegangan (V), arus (I), dan daya (P) terhadap multimeter digital standar pada lima kondisi beban yang berbeda.

Tabel 7. Hasil Kalibrasi Sensor INA219

No	Multi-meter			Sensor INA219			Error (%)		
	V (V)	I (A)	P (W)	V (V)	I (A)	P (W)	V	I	P
1	14.13	1.2782	18.06	14.004	1.2686	17.77	0.89	0.75	1.61
2	14.34	1.2403	17.79	14.284	1.2339	17.63	0.39	0.52	0.88
3	14.94	1.5134	22.61	14.856	1.5002	22.29	0.56	0.87	1.42
4	17.68	1.2618	22.31	17.600	1.2522	22.04	0.45	0.76	1.20
5	17.20	1.1997	20.63	17.140	1.1850	20.31	0.35	1.23	1.57
Rata-rata	-	-	-	-	-	-	0.53	0.83	1.34

$$Presentase\ Error = \left(\frac{Nilai\ Multimeter - Nilai\ Sensor}{Nilai\ Multimeter} \right) \times 100\% \quad (7)$$

Berdasarkan hasil kalibrasi sensor INA219 pada Tabel 7, pengujian menunjukkan bahwa hasil pengukuran tegangan, arus, dan daya memiliki selisih yang kecil dibandingkan multimeter standar. Nilai *error* tegangan berkisar antara 0.35% hingga 0.89% dengan rata-rata 0.53%, *error* arus berkisar antara 0.52% hingga 1.23% dengan rata-rata 0.83%, dan *error* daya berkisar antara 0.88% hingga 1.61% dengan rata-rata 1.34%. Seluruh nilai *error* masih berada dalam batas toleransi alat ukur yang wajar sehingga sensor INA219 dinyatakan layak digunakan dalam sistem pengukuran daya panel surya.

d. Kalibrasi Sensor DS18B20 (Suhu)

Kalibrasi sensor DS18B20 dilakukan dengan membandingkan pembacaan suhu terhadap termometer digital standar pada lima kondisi suhu yang berbeda.

Tabel 8. Hasil Kalibrasi Sensor DS18B20

No	Termometer (°C)	DS18B20 (°C)	Selisih (°C)	Error (%)
1	45.6	45.50	0.10	0.22
2	48.8	48.31	0.49	1.00
3	49.2	48.94	0.26	0.53
4	49.9	49.56	0.34	0.68
5	39.7	39.56	0.14	0.35
Rata-rata	-	-	0.27	0.56

$$\text{Presentase Error} = \left(\frac{\text{Nilai Termometer} - \text{Nilai Sensor}}{\text{Nilai Termometer}} \right) \times 100\% \quad (8)$$

Berdasarkan hasil kalibrasi sensor DS18B20 pada Tabel 8, pengujian menunjukkan bahwa hasil pembacaan sensor DS18B20 memiliki selisih yang sangat kecil dibandingkan termometer standar. Nilai *error* berkisar antara 0.22% hingga 1.00% dengan rata-rata *error* sebesar 0.56%. Nilai *error* yang sangat kecil ini menunjukkan bahwa sensor DS18B20 memiliki tingkat akurasi yang tinggi dan dinyatakan layak digunakan dalam sistem pengukuran suhu panel surya.

3.4 Tampilan Dashboard Monitoring dan Kontrol

a. Tampilan LCD 20x4

Sistem PLTS dilengkapi dengan LCD 20x4 I2C yang terpasang pada bagian depan box kontrol untuk menampilkan data monitoring secara lokal dan real-time tanpa memerlukan koneksi internet. LCD menampilkan empat baris informasi yang diperbarui setiap 10 detik sesuai interval pembacaan sensor.

Gambar 15 menunjukkan tampilan LCD 20x4 saat sistem sedang beroperasi pada mode *tracking* aktif. Terlihat pada LCD menampilkan data secara lengkap yaitu waktu pada baris pertama, intensitas cahaya (L) dan daya (P) pada baris kedua, tegangan (V) dan arus (I) pada baris ketiga, serta sudut servo horizontal (H) dan vertikal (V) pada baris keempat. Data diperbarui setiap 10 detik secara otomatis.

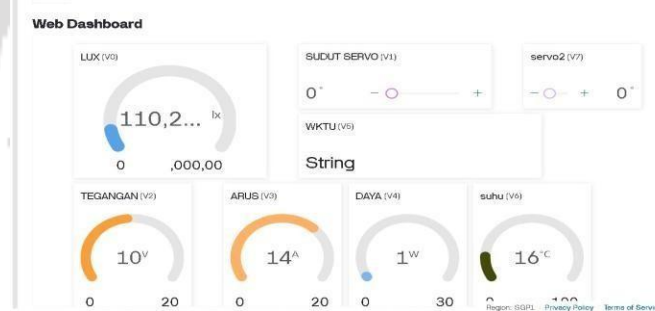


Gambar 15. Tampilan LCD 20x4 saat Sistem Aktif

b. Tampilan Blynk Web Dashboard

Selain LCD, sistem juga dilengkapi dengan monitoring jarak jauh melalui aplikasi *Blynk*. *Blynk* menampilkan data sensor secara *real-time* dalam bentuk *gauge* dan *widget* yang mudah dibaca melalui *smartphone* maupun *web browser*. Data dikirimkan dari ESP32 ke *server Blynk* setiap 10 detik melalui koneksi *WiFi* menggunakan 8 *virtual pin* (V0-V7).

Gambar 16 menunjukkan tampilan *Blynk Web Dashboard* saat Pengujian Tampilan. Tampilan ini merupakan kondisi saat pengujian konektivitas awal sistem. Terlihat *gauge* LUX (V0), *gauge* Tegangan (V2), Arus (V3), Daya (V4), dan Suhu (V6). *Widget* SUDUT SERVO (V1) dan Servo2 (V7) menampilkan sudut servo horizontal dan vertikal. *Widget* WKTU (V5) menampilkan waktu *real-time* dalam format string.



Gambar 16. Tampilan Blynk Web Dashboard

c. Tampilan Data Firebase Realtime Database

Seluruh data hasil pengukuran sensor dikirimkan secara otomatis ke *Firebase Realtime Database* (RTDB) setiap 10 detik oleh ESP32 melalui koneksi *WiFi*. Data

disimpan pada path /panel/log/[tanggal]/[key] dimana setiap record memiliki *key* unik yang digenerate otomatis oleh Firebase.

Gambar 17 menunjukkan contoh data yang tersimpan pada *Firebase RTDB* dalam format *JSON*. Terlihat satu record dengan *key* unik berisi data lengkap pengukuran pada tanggal 17-05-2026 pukul 11:30:13 dengan nilai arus 1.3697 A, daya 23.1041 W, horizontal 91°, lux 37830, suhu 49.6875°C, tegangan 16.868 V, dan vertikal 84°.

```
},
"-OsoBj8lpFFZ-5Ld0Qj9": {
  "arus": 1.3697,
  "daya": 23.1041,
  "hari": "Minggu",
  "horizontal": 91,
  "lux": 37830,
  "suhu": 49.6875,
  "tanggal": "17-05-2026",
  "tegangan": 16.868,
  "timestamp": 1779017413,
  "vertical": 84,
  "waktu": "11:30:13"
},
```

Gambar 17. Tampilan Data pada *Firebase Realtime Database*

d. Tampilan Data *Google Spreadsheet*

Data dari *Firebase* secara otomatis diteruskan ke *Google Spreadsheet* melalui *App Script* setiap kali ada data baru yang masuk. *Spreadsheet* memiliki kolom Hari, Tanggal, Waktu, Lux, Tegangan, Arus, Daya, Horizontal, Vertical, dan Suhu yang tersusun rapi sehingga memudahkan analisis data lebih lanjut.

Gambar 22 menunjukkan tampilan header dan data awal *Google Spreadsheet* pada tanggal 17-05-2026. Data dimulai dari pukul 09:00:10 dengan nilai *lux* sekitar 26.762-29.421, tegangan 14.004-14.072 V, arus 1.2686-1.366 A, daya 17.77-19.33 W, sudut horizontal 76°, vertikal 44°, dan suhu 45.375-45.75°C.

	A	B	C	D	E	F	G	H	I	J	K	L
	Hari	Tanggal	Waktu	Lux	Tegangan	Arus	Daya	Horizontal	Vertical	Suhu		
1	Minggu	17-05-2026	9:00:10 AM	26761.66602	14.004	1.2686	17.76547	76	44	45.5		
2	Minggu	17-05-2026	9:00:20 AM	27138.33203	14.02	1.288	18.05776	76	44	45.5		
3	Minggu	17-05-2026	9:00:30 AM	27626.66602	14.028	1.3115	18.39772	76	44	45.5625		
4	Minggu	17-05-2026	9:00:40 AM	27373.33203	14.024	1.3008	18.24242	76	44	45.4375		
5	Minggu	17-05-2026	9:01:00 AM	27482.49805	14.02	1.304	18.28208	76	44	45.375		
6	Minggu	17-05-2026	9:04:10 AM	29046.66602	14.04	1.3628	19.13371	76	44	45.1875		
7	Minggu	17-05-2026	9:04:20 AM	29247.49805	14.064	1.3714	19.28737	76	44	45.25		
8	Minggu	17-05-2026	9:04:30 AM	29259.16602	14.06	1.3708	19.27345	76	44	45.375		
9	Minggu	17-05-2026	9:04:40 AM	29363.33203	14.064	1.3738	19.32112	76	44	45.375		
10	Minggu	17-05-2026	9:04:50 AM	29333.33203	14.064	1.3694	19.25924	76	44	45.5		
11	Minggu	17-05-2026	9:05:00 AM	29140.83203	14.064	1.3615	19.14814	76	44	45.5		
12	Minggu	17-05-2026	9:05:10 AM	29184.16602	14.056	1.3633	19.16255	76	44	45.6875		
13	Minggu	17-05-2026	9:05:20 AM	29225.83203	14.064	1.3627	19.16501	76	44	45.75		
14	Minggu	17-05-2026	9:05:30 AM	28830.83203	14.064	1.3616	19.14954	76	44	45.8125		
15	Minggu	17-05-2026	9:05:40 AM	29299.99805	14.06	1.3634	19.1694	76	44	45.75		

Gambar 18. Tampilan Header dan Data Awal *Google Spreadsheet*

4. PENUTUP

Berdasarkan hasil perancangan, implementasi, serta pengujian yang telah dilakukan, sistem PLTS berbasis *Internet of Things (IoT)* dengan menggunakan model *Ma-chine Learning Decision Tree* dapat bekerja sesuai dengan yang direncanakan. Sistem mampu melakukan pelacakan posisi matahari secara otomatis menggunakan dua sumbu pergerakan (*dual-axis*) berdasarkan prediksi model *Machine learning* secara *real-time*, sehingga proses *tracking* menjadi lebih tepat dan terkontrol. Di sisi lain, integrasi antara sensor, mikrokontroler ESP32, serta sistem monitoring berbasis Firebase, Blynk, dan Google Spreadsheet berjalan dengan baik dan stabil selama pengujian. Dari hasil pengujian, diperoleh bahwa penerapan model *Machine Learning* memberikan efisiensi yang lebih optimal dibandingkan sistem tanpa *Machine learning*.

Berdasarkan hasil pengujian sebelum penerapan *machine learning*, diperoleh rata-rata intensitas cahaya sebesar 30.259 lux, tegangan 15.54 V, arus 1.15 A, daya 17.95 W, dan suhu panel 45.97°C. Setelah penerapan *machine learning*, diperoleh rata-rata intensitas cahaya sebesar 25.175 lux, tegangan 16.21 V, arus 0.99 A, daya 16.34 W, dan suhu panel 44.62°C. Hasil analisis menunjukkan bahwa rasio daya terhadap intensitas cahaya meningkat dari 0.000593 W/lux menjadi 0.000649 W/lux atau naik sekitar 13.37%, sehingga sistem mampu menghasilkan daya lebih optimal meskipun intensitas cahaya yang diterima lebih rendah.

Penurunan suhu panel sebesar 3.15% juga menunjukkan bahwa sistem *dual-axis* berbasis *machine learning* mampu mengurangi *thermal losses* melalui pengaturan posisi panel yang lebih optimal. Secara keseluruhan, sistem yang dikembangkan tidak hanya mampu meningkatkan efisiensi pemanfaatan energi surya, tetapi juga memberikan solusi monitoring yang lebih praktis dan berkelanjutan melalui platform *IoT* yang terintegrasi.

Sebagai keterbatasan penelitian, label sudut panel ditentukan secara manual berdasarkan pola pergerakan matahari, sehingga model *machine learning* yang dihasilkan berfungsi sebagai sistem otomasi *tracking* berbasis data historis. Selain itu, pengujian dilakukan pada dua hari berbeda dengan kondisi intensitas cahaya yang tidak identik (30.259 lux vs 25.175 lux), sehingga perbandingan daya absolut tidak dapat dilakukan secara langsung. Untuk penelitian selanjutnya, disarankan pengujian menggunakan dua panel secara bersamaan pada kondisi cuaca yang sama, serta eksplorasi optimasi sudut berbasis maksimasi daya aktual.

PERSANTUNAN

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penelitian dan penyusunan naskah publikasi ini dapat diselesaikan dengan baik. Naskah ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Teknik Elektro, Fakultas Teknik.

Penulis menyadari bahwa proses penyusunan penelitian ini tidak terlepas dari bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Kedua orang tua yang selalu memberikan doa, dukungan, serta motivasi selama proses penyelesaian studi.
2. Bapak Hasyim Asy'ari, S.T., M.T. selaku dosen pembimbing yang telah memberikan arahan dan bimbingan selama penelitian berlangsung.
3. Seluruh dosen Program Studi Teknik Elektro yang telah memberikan ilmu dan pengalaman selama masa perkuliahan.
4. Teman-teman serta semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyelesaian penelitian ini.

Penulis menyadari bahwa naskah ini masih memiliki keterbatasan, sehingga kritik dan saran yang membangun sangat diharapkan untuk pengembangan penelitian selanjutnya. Semoga penelitian ini dapat memberikan manfaat serta menjadi referensi dalam pengembangan sistem Solar Panel berbasis teknologi yang lebih efisien dan berkelanjutan.

DAFTAR PUSTAKA

- Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13-17-August-2016*(March): 785–94. doi:10.1145/2939672.2939785.
- Cyril, Voyant, Notton Gilles, Kalogirou Soteris, Nivet Marie-Laure, Paoli Christophe, Motte Fabrice, and Fouilloy Alexis. 2017. "Machine Learning Methods for Solar Radiation Forecasting: A Review." *Renewable Energy*: 1–33.
- Febriyanty, Nur Elyta, Francisco Marceliano, Putra Hidayanta, Ranap Sitorus, and Lely Wijayanti. 2026. "Pendekatan Model Machine Learning Untuk Meningkatkan Efisiensi Energi Di Berbagai Sektor Industri : Studi Literatur."
- Priatam et al. 2021. "Analisa Radiasi Sinar Matahari Terhadap Panel Surya 50 WP." *RELE:Jurnal Teknik Elektro* 4(1): 48–54.
<http://jurnal.umsu.ac.id/index.php/RELE/article/view/7825>.
- Qomaruddin, M. N., Khairi, M., & Sulistiyanto. 2019. "Jurnal Qua Teknika , Vol . 9 No . 2 September 2019 ISSN 2088-2424 (Cetak); ISSN 2527-3892 (Elektronik) Fakultas Teknik Universitas Islam Balitar , Blitar [https :// Ejournal . Unisbablitar . Ac . Id / Index . Php / qua](https://ejournal.unisbablitar.ac.id/index.php/qua) ; Email : Quateknika@unisbablitar." 9(2): 27–32.
- Yudhistira, Ilham, dan Hasyim Asy'ari. "PENGEMBANGAN SISTEM SOLAR TRACKER DUAL-AXIS BERBASIS IOT DAN KONTROL BEBAN JARAK JAUH DENGAN RELAY UNTUK MENINGKATKAN EFEKTIVITAS PLTS OFF-GRID Ilham Yudhistira Ardhana ; Hasyim Asy ' Ari , S . T . , M . T Program Studi Teknik Elektro , Fakultas Teknik , Universitas Muhammadiyah Surakarta.
- Yapie, Any Kurniawaty, and Fathur Riski. 2025. "Perancangan Solar Tracker Dual Axis Untuk Optimasi Panel Surya Digunakan Untuk Menghasilkan Energi Listrik . Energi Listrik Yang Dihasilkan Oleh Panel Surya Kinerjanya . Dengan Mempelajari Kondisi Lingkungan Dan Posisi Geografis , Diharapkan Dapat."