BAB II
TINJAUAN PUSTAKA

2.1 Penelitian yang Relevan

Dari beberapa literatur yang telah ditinjau melalui *Systematic Literature Review* (SLR) dari beberapa artikel ilmiah yang bereputasi, dapat ditemukan beberapa penelitian terdahulu yang terkait, seperti penelitian mengenai kuantifikasi efek pendinginan perairan (*cooling effect*) terhadap produktivitas energi pada sistem PLTS Apung (Dörenkämper et al., 2021). Penelitian tersebut membahas secara mendalam bagaimana keberadaan air di bawah modul fotovoltaik memengaruhi suhu operasional panel serta perolehan energinya, dengan membandingkan instalasi terapung dan instalasi darat pada dua zona iklim yang berbeda. Pendekatan yang digunakan bersifat eksperimental-termal berbasis pengukuran fisik lapangan, tanpa melibatkan pemodelan prediktif berbasis *machine learning*. Kesamaan dengan penelitian ini terletak pada objek kajian yang sama, yaitu sistem PLTS Apung yang berada di bawah pengaruh iklim mikro tropis. Perbedaannya adalah penelitian tersebut berhenti pada analisis termal fisik, sedangkan penelitian ini melangkah lebih jauh dengan menerapkan pemodelan komparatif *XGBoost* dan *Random Forest*. Meskipun demikian, penelitian tersebut memiliki relevansi yang tinggi sebagai landasan ilmiah yang memperkuat pentingnya faktor suhu dalam dinamika daya keluaran PLTS Apung, sehingga menjadi rujukan kuat dalam pemilihan variabel penelitian ini.

Penelitian kedua yang relevan dengan penelitian ini adalah penelitian yang membahas peramalan produksi daya listrik fotovoltaik terapung pada kondisi cuaca berawan ekstrem menggunakan pendekatan komputasi cerdas (Khortsriwong et al., 2023). Tujuannya adalah memprediksi penurunan *output* daya PLTS Apung akibat disrupsi awan mendadak dengan memanfaatkan data meteorologi. Hasilnya, arsitektur hibrida CNN-BiGRU yang digunakan mampu menangani anomali cuaca berawan pekat dengan tingkat akurasi yang tinggi, meskipun disertai beban komputasi yang berat. Persamaannya dengan penelitian ini adalah sama-sama menjadikan estimasi nilai daya keluaran aktual pada sistem PLTS Apung sebagai target utama. Perbedaannya adalah penelitian tersebut menggunakan arsitektur *deep learning* tingkat lanjut, sedangkan penelitian ini memilih model *ensemble learning* berupa *XGBoost* dan *Random Forest* yang lebih ringkas dan efisien secara komputasi.

Pada penelitian lainnya yang relevan dengan penelitian ini membahas mengenai perbandingan keandalan algoritma *Random Forest* dan *XGBoost* dalam menghasilkan prediksi radiasi matahari dari data meteorologi harian (Mishra et al., 2023). Penelitian tersebut secara khusus menempatkan kedua algoritma dalam sebuah komparasi *head-to-head* yang paling langsung, sehingga hasilnya dapat menunjukkan algoritma mana yang lebih unggul pada kondisi data yang sama. Hasil pengujian membuktikan keunggulan *Random Forest* dengan nilai R^2 sebesar 0.809 dan RMSE 108.55, dibandingkan *XGBoost* yang memperoleh R^2 sebesar 0.645 dan RMSE 122.12. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pengujian komparasi langsung antara *XGBoost* dan *Random Forest* pada *dataset* indikator cuaca. Namun, terdapat celah penelitian yang cukup mendasar, yaitu evaluasi tersebut dilakukan sepenuhnya pada stasiun darat dengan karakteristik termal yang stabil. Oleh karena itu, penelitian ini hadir untuk menguji apakah keunggulan *Random Forest* tersebut tetap konsisten pada ekosistem PLTS Apung, atau justru *XGBoost* yang lebih superior dalam menangkap pola non-linear dari anomali iklim mikro perairan.

Kemudian, penelitian lain yang relevan membahas peramalan daya keluaran fotovoltaik menggunakan model susun (*stacking ensemble*) yang menggabungkan beberapa algoritma sekaligus (Abdellatif et al., 2022). Penelitian tersebut tidak memilih satu algoritma tunggal, melainkan menggabungkan *Random Forest*, *XGBoost*, dan AdaBoost sebagai *base learner* yang dipadukan melalui *meta-learner* Extra Trees. Hasilnya menunjukkan bahwa pendekatan penggabungan tersebut mampu menurunkan tingkat kesalahan prediksi secara signifikan dibandingkan kinerja masing-masing algoritma tunggal penyusunnya. Kesamaan dengan penelitian ini terletak pada pemanfaatan *Random Forest* dan *XGBoost* sebagai komponen inti pemodelan. Perbedaannya adalah penelitian tersebut menggabungkan kedua algoritma ke dalam satu arsitektur susun pada instalasi PLTS darat, sedangkan penelitian ini justru mengomparasikan keduanya secara terpisah untuk menemukan algoritma yang paling unggul secara mandiri pada ekosistem PLTS Apung.

Penelitian selanjutnya yang relevan dengan penelitian yang dilakukan membahas prediksi daya *output* sistem panel surya menggunakan metode Regresi Linear Berganda (Bramasto & Khairiani, 2022). Penelitian tersebut bertujuan mengestimasi nilai daya aktual panel surya dengan kerangka kerja CRISP-DM berdasarkan data pengukuran

historis di lapangan. Hasil pemodelannya menunjukkan keterbatasan, di mana tingkat korelasi yang diperoleh hanya mencapai 52% (koefisien korelasi 0.52) karena metode linear tidak mampu menangkap sifat parameter cuaca yang non-linear. Kesamaan penelitian tersebut dengan penelitian ini adalah sama-sama menargetkan nilai daya aktual sistem PLTS berdasarkan data historis parameter cuaca. Perbedaannya terletak pada metode, yaitu penelitian tersebut menggunakan pendekatan statistik linear pada instalasi darat, sedangkan penelitian ini membangun arsitektur non-linear menggunakan *XGBoost* dan *Random Forest* pada PLTS Apung. Keterbatasan pendekatan linear inilah yang justru menjadi salah satu dasar argumentasi penulis untuk beralih pada pendekatan berbasis *ensemble*.

Penelitian lain yang menggunakan pendekatan serupa dengan penelitian ini membahas prakiraan *output* daya energi surya di kawasan Timur Tengah menggunakan beragam model *machine learning* (Lari et al., 2025). Penelitian tersebut menempatkan *Random Forest* dan *XGBoost* dalam satu tolok ukur (*benchmark*) bersama model berbasis pohon lainnya, seperti *Linear Regression* dan *Decision tree*, untuk melihat posisi relatif performa setiap algoritma. Hasil evaluasinya menyimpulkan bahwa model berbasis pohon keputusan, khususnya *Random Forest*, memiliki keandalan tertinggi di antara model yang diuji dalam memetakan interaksi variabel lingkungan terhadap fluktuasi daya listrik. Kesamaan dengan penelitian ini adalah sama-sama mengomparasikan performa *XGBoost* dan *Random Forest* untuk memprediksi keluaran daya panel surya. Perbedaannya adalah penelitian tersebut memakai data sekunder dari lingkungan gurun gersang di Doha, Qatar, sedangkan penelitian ini memanfaatkan data primer dari iklim mikro perairan tropis di Danau IT-PLN.

Selanjutnya, penelitian yang relevan dengan penelitian yang dilakukan membahas perancangan sistem *Internet of Things* (IoT) berbiaya rendah untuk pemantauan variabel iklim dan generasi fotovoltaik secara *real-time* (Gomes De Melo et al., 2021). Penelitian tersebut membangun perangkat akuisisi data murah yang mampu merekam tegangan, arus, dan daya bersama sejumlah variabel meteorologi langsung dari sisi panel surya. Hasilnya, sistem berhasil merekam data secara berkelanjutan langsung dari panel, bukan dari inverter, serta menyimpannya dalam bentuk *time-series* yang siap dimanfaatkan untuk pemodelan lanjutan. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pembangunan sistem akuisisi data mandiri berbasis sensor cuaca sekaligus

pengukuran daya keluaran secara langsung di lokasi pembangkit. Perbedaannya adalah penelitian tersebut berhenti pada tahap pemantauan dan visualisasi data, sedangkan penelitian ini melanjutkan data hasil akuisisi tersebut ke tahap komparasi *XGBoost* dan *Random Forest* untuk estimasi daya.

Penelitian lainnya yang relevan membahas peningkatan akurasi prediksi daya fotovoltaik melalui pemodelan komputasi dan seleksi fitur lingkungan (Amiri et al., 2024). Berbeda dengan penelitian yang berfokus pada perbandingan atau penggabungan algoritma, penelitian ini menyoroti pengaruh tahapan seleksi fitur (*feature selection*) terhadap performa *Random Forest* dan *XGBoost*. Tujuannya adalah mereduksi dimensi data yang rumit untuk mengidentifikasi kombinasi variabel cuaca yang paling optimal bagi model regresi. Hasilnya membuktikan bahwa tahapan seleksi fitur secara signifikan mampu meminimalkan kesalahan prediksi sekaligus mempercepat waktu pelatihan algoritma. Kesamaan dengan penelitian ini terletak pada perhatian terhadap pentingnya seleksi fitur iklim mikro dalam menentukan variabel penentu daya listrik surya. Perbedaannya adalah penelitian tersebut mengkaji sistem PLTS darat dengan cakupan parameter makro yang luas, sedangkan penelitian ini membatasi fitur masukan secara ketat pada suhu udara, kelembapan udara, dan intensitas cahaya dari sensor lokal di atas permukaan air.

Kemudian, penelitian yang menggunakan pendekatan mutakhir membahas pengembangan model prakiraan daya surya berbasis *hybrid* (Nguyen et al., 2025). Penelitian tersebut memadukan tiga algoritma *ensemble* sekaligus, yaitu *XGBoost*, LightGBM, dan *Random Forest*, sebagai *base learner* dalam kerangka *stacking ensemble* yang bobot kontribusinya dioptimalkan menggunakan algoritma L-BFGS-B. Hasilnya menunjukkan bahwa model gabungan tersebut mampu menghasilkan prediksi yang jauh lebih akurat dan efisien secara komputasi dibandingkan model *deep learning* seperti LSTM dan GRU. Kesamaan dengan penelitian ini terletak pada penempatan *XGBoost* dan *Random Forest* sebagai algoritma inti untuk mengestimasi daya surya berbasis parameter cuaca. Perbedaannya adalah penelitian tersebut menggabungkan ketiga algoritma menjadi satu model *hybrid* yang kompleks pada PLTS darat berskala besar, sedangkan penelitian ini mempertahankan efisiensi dengan mengomparasikan *XGBoost* dan *Random Forest* secara langsung pada PLTS Apung berskala kecil.

Penelitian terakhir yang relevan dengan penelitian yang dilakukan membahas prediksi *output* daya sistem fotovoltaik terapung menggunakan pendekatan hierarki (Sulaiman et al., 2025). Penelitian tersebut membandingkan lima model *machine learning*, yaitu *FFNN*, *Random Forest*, *ELM*, *SVM*, dan *XGBoost*, untuk memprediksi daya PLTS Apung berkapasitas 157.2 kWp di Malaysia. Hasilnya menyimpulkan bahwa *Feed-Forward Neural Network* memberikan akurasi tertinggi dengan R^2 mendekati 1, diikuti *Random Forest* dengan R^2 0.99997 dan *XGBoost* dengan R^2 0.996. Kesamaan penelitian tersebut dengan penelitian ini adalah sama-sama meneliti PLTS Apung di kawasan beriklim tropis serta membandingkan *XGBoost* dan *Random Forest* dengan metrik RMSE, MAE, dan R^2. Perbedaannya adalah penelitian tersebut memakai parameter kelistrikan berupa arus dan tegangan *MPPT* dalam kerangka prediksi bertingkat pada sistem berskala besar, sedangkan penelitian ini menggunakan parameter cuaca berupa suhu, kelembapan, dan intensitas cahaya pada satu unit PLTS Apung berskala kecil.

Berdasarkan tinjauan terhadap berbagai literatur dan penelitian terdahulu di atas, dapat ditarik sebuah sintesis bahwa algoritma berbasis *ensemble*, khususnya *Random Forest* dan *XGBoost*, terbukti secara empiris sebagai metode pemodelan regresi yang paling tangguh dalam menangani fluktuasi data meteorologi. Ketangguhan tersebut telah dikaji dari beragam sudut pendekatan, mulai dari komparasi langsung, analisis seleksi fitur, penggabungan melalui *stacking*, hingga pengembangan model *hybrid* mutakhir. Meskipun demikian, mayoritas penelitian tersebut masih berfokus pada instalasi fotovoltaik terestrial. Oleh karena itu, penelitian ini dirancang untuk mengisi celah literatur (*research gap*) dengan menerapkan pengujian komparatif performa *XGBoost* dan *Random Forest* secara spesifik pada ekosistem PLTS Apung, yang dinamika parameter iklim mikronya memiliki karakteristik non-linear jauh lebih kompleks dan terekam langsung melalui instrumen *Automatic Weather Station* (AWS).

2.2 Landasan Teori

Landasan teori membahas teori-teori yang berkaitan atau berhubungan dengan penelitian yang dilakukan. Berikut dijelaskan landasan teori pada penulisan ini :

2.2.1 Pembangkit Listrik Tenaga Surya (PLTS) Apung

Sistem Pembangkit Listrik Tenaga Surya (PLTS) Apung atau *Floating Photovoltaic* (FPV) merupakan teknologi penempatan panel surya di atas permukaan badan air seperti waduk, danau, maupun kolam retensi, yang hadir sebagai solusi menjanjikan di wilayah dengan keterbatasan ketersediaan lahan. Komponen utama penyusun sistem FPV terdiri atas struktur penopang terapung (*pontoon*), sistem penambatan (*mooring system*), modul fotovoltaik, serta kabel dan koneksi kelistrikan (Koondhar et al., 2024). Sama seperti sistem fotovoltaik pada umumnya, konversi energi pada FPV bertumpu pada efek fotovoltaik, yaitu ketika foton cahaya matahari menabrak material semikonduktor berbasis silikon dan memicu eksitasi elektron pada sel surya sehingga menghasilkan energi listrik searah (*Direct Current*/DC).

Keunggulan fisis mendasar dari infrastruktur PLTS Apung dibandingkan sistem konvensional di daratan (*ground-mounted PV*) terletak pada eksistensi efek pendinginan alami (*cooling effect*) yang disumbangkan oleh badan air di bawahnya. Suhu operasional sel fotovoltaik secara termodinamika berbanding terbalik dengan efisiensi kinerjanya. Akumulasi panas berlebih pada modul akan meningkatkan resistansi internal material semikonduktor yang berujung pada penurunan tegangan (*voltage drop*) dan reduksi total daya keluaran. Mekanisme evaporasi dan konveksi termal dari permukaan air secara konstan menyerap panas dari panel sehingga menjaga suhu modul tetap berada pada koridor kerja optimal dan menghasilkan perolehan energi (*energy yield*) yang lebih tinggi, sehingga sistem FPV terbukti mampu menghasilkan daya keluaran yang lebih besar dibandingkan instalasi daratan pada kondisi lingkungan yang setara (Koondhar et al., 2024). Kendati demikian, penempatan di atas permukaan air membentuk sebuah ekosistem iklim mikro (*microclimate*) yang unik, di mana interaksi fluktuatif antara parameter atmosfer lokal dengan permukaan air menghasilkan pola persebaran data yang bersifat kompleks dan sangat non-linear.

Oleh karena itu, pemahaman dan pengukuran yang spesifik mengenai variabel-variabel pembentuk iklim mikro tersebut menjadi fondasi krusial. Identifikasi terhadap parameter lingkungan ini mutlak diperlukan sebagai atribut masukan (*input*) untuk memprediksi anomali fluktuasi daya keluaran PLTS secara akurat.

2.2.2 Parameter Iklim Mikro dan Intensitas Cahaya

Karakteristik keluaran daya dari sistem fotovoltaik di atas permukaan air memiliki ketergantungan yang bersifat linier maupun non-linier dan sangat sensitif terhadap dinamika iklim mikro di sekitarnya. Variabilitas cuaca lokal bertindak sebagai faktor eksternal utama yang menentukan fluktuasi pasokan energi harian. Oleh karena itu, pemahaman terhadap perilaku setiap parameter iklim mikro menjadi landasan penting dalam memodelkan estimasi daya keluaran panel surya. Analisis teoritis pada bagian ini difokuskan pada tiga variabel penentu berikut:

1. Suhu (*Temperature)*:

   Suhu memiliki pengaruh yang berlawanan terhadap kinerja sel surya. Meskipun cahaya diperlukan untuk membangkitkan energi, kenaikan suhu sel yang berlebihan justru menurunkan efisiensi konversi. Secara fisika padat, peningkatan suhu mempercepat laju rekombinasi pembawa muatan dan menurunkan tegangan rangkaian terbuka (*open-circuit voltage*), sehingga daya keluaran berkurang. Setiap kenaikan suhu sel sebesar 1 °C dilaporkan menurunkan keluaran daya sekitar 0,5% dari kapasitas nominalnya, dengan efisiensi konversi yang cenderung menurun secara linier terhadap kenaikan suhu (Li et al., 2021). Fenomena inilah yang menyebabkan panel surya kerap bekerja di bawah kapasitas idealnya saat suhu operasi meningkat, sehingga pengelolaan suhu operasi menjadi aspek penting dalam menjaga performa panel.

2. Kelembapan Udara (*Relative Humidity*):

   Kelembapan relatif memengaruhi kinerja panel surya secara tidak langsung melalui interaksinya dengan cahaya dan permukaan modul. Kandungan uap air dan butiran air di atmosfer dapat membiaskan, memantulkan, dan menghamburkan cahaya matahari, sehingga mengurangi komponen radiasi langsung yang sampai ke permukaan sel. Semakin tinggi kelembapan, semakin besar pula efek hamburan yang terjadi. Selain itu, kelembapan yang tinggi mendorong terbentuknya lapisan pengotor (*soiling*) yang lengket pada permukaan modul dan berpotensi menyebabkan korosi dalam jangka panjang. Sebagai gambaran, efisiensi sel surya dilaporkan meningkat dari 9,7% menjadi 12,04% ketika kelembapan relatif diturunkan dari 60% menjadi 48% (Hasan et al., 2022). Dengan demikian, kelembapan yang tinggi cenderung diikuti oleh penurunan daya keluaran panel.

3. Intensitas Cahaya (*Illuminance*):

   Intensitas cahaya merupakan faktor penggerak utama dalam proses konversi energi pada sel surya. Ketika foton dengan energi yang memadai mengenai permukaan sel, elektron akan tereksitasi dan membentuk pasangan elektron-hole yang menghasilkan arus listrik. Semakin besar intensitas cahaya yang diterima, semakin banyak pula pasangan pembawa muatan yang terbentuk sehingga arus dan daya keluaran meningkat. Hasil penelitian menunjukkan bahwa arus hubung singkat (*short-circuit current*) meningkat secara linier seiring bertambahnya intensitas cahaya, sementara tegangan rangkaian terbuka (*open-circuit voltage*) meningkat secara logaritmik, sehingga daya keluaran maksimum turut meningkat (Li et al., 2021).

Secara teoretis, ketiga parameter iklim mikro tersebut tidak bekerja secara terpisah, melainkan saling berinteraksi dalam menentukan besarnya daya keluaran panel surya. Kondisi cuaca di sekitar pembangkit, seperti intensitas penyinaran matahari maupun keberadaan awan dan hujan, secara langsung memengaruhi jumlah radiasi matahari yang mampu diserap oleh permukaan modul sehingga turut menentukan energi listrik yang dihasilkan. Oleh karena itu, pemantauan parameter lingkungan di lokasi pembangkit menjadi aspek penting dalam menjaga kinerja optimal pembangkit listrik tenaga surya (Haris et al., 2020).

2.2.3 Sistem Pengukuran Daya Arus Searah (DC)

Panel surya pada PLTS Apung menghasilkan energi listrik dalam bentuk arus searah (*Direct Current*/DC), sehingga diperlukan sistem pengukuran DC yang presisi untuk memantau daya keluarannya secara *real-time* (Zurifqyaldi et al., 2025). Secara teoretis, daya listrik pada sirkuit DC dihitung berdasarkan Hukum Watt, yaitu hasil kali antara tegangan (V) dan arus (I):

$$P = V \times I \tag{2.1}$$

Keterangan:

P : Daya listrik (Watt)

V : Tegangan Listrik (Volt)

I : Arus Listrik (Ampere)

Pengukuran daya DC memerlukan dua besaran utama, yaitu tegangan dan arus. Arus DC umumnya diukur menggunakan *shunt resistor*, yaitu resistor bernilai sangat kecil yang dipasang seri pada jalur arus dan menghasilkan jatuh tegangan proporsional terhadap besar arus yang mengalir (Gunoto et al., 2022).

2.2.4 Perangkat Akuisisi Data

Perangkat akuisisi data merupakan kesatuan perangkat keras yang berfungsi untuk mengukur, mencatat, dan menyimpan parameter penelitian secara otomatis dan berkelanjutan. Pada sistem pemantauan berbasis Internet of Things (IoT), perangkat ini umumnya tersusun atas sebuah mikrokontroler sebagai unit pemrosesan utama yang terhubung dengan sejumlah sensor pengukur dan modul pendukung (Munawar, 2025). Dalam penelitian ini, perangkat akuisisi data dirancang untuk merekam parameter cuaca sekaligus daya keluaran panel surya secara bersamaan. Komponen penyusun perangkat tersebut diuraikan pada subbab berikut.

2.2.4.1 Mikrokontroler ESP32

ESP32 merupakan mikrokontroler berdaya rendah yang telah dilengkapi modul Wi-Fi dan Bluetooth terintegrasi di dalam chip, sehingga banyak digunakan sebagai unit pemrosesan utama pada sistem pemantauan berbasis IoT (Munawar, 2025). Pada penelitian ini, ESP32 berperan sebagai pengendali utama yang bertugas membaca data dari seluruh sensor, memberikan penanda waktu pada setiap pembacaan, serta mengatur proses penyimpanan hasil perekaman ke media penyimpanan.

2.2.4.2 Sensor Suhu dan Kelembapan DHT22

Sensor DHT22 merupakan sensor digital yang mampu mengukur suhu udara dalam satuan derajat Celsius dan kelembapan relatif udara dalam satuan persen secara bersamaan dengan tingkat akurasi yang memadai untuk keperluan pemantauan lingkungan (Ramawardana, 2025). Dalam penelitian ini, sensor DHT22 digunakan untuk merekam dua parameter iklim mikro, yaitu suhu dan kelembapan udara di sekitar panel surya.

2.2.4.3 Sensor Intensitas Cahaya BH1750

Sensor BH1750 merupakan sensor cahaya digital dengan antarmuka komunikasi Inter-Integrated Circuit (I2C) yang mampu mengukur intensitas cahaya atau iluminansi dalam satuan lux pada rentang 1 sampai 65535 lux (Khuriati, 2022). Sensor ini digunakan

untuk merekam intensitas cahaya matahari yang diterima oleh permukaan panel surya sebagai salah satu variabel prediktor.

2.2.4.4 Modul Pengukuran Daya PZEM-017 DC dan Resistor Shunt

Modul PZEM-017 DC merupakan modul pengukur parameter listrik arus searah yang dapat membaca tegangan, arus, daya, dan energi hingga 300 VDC. Pengukuran arus pada modul ini memanfaatkan resistor shunt eksternal sebagai sensor arus, sedangkan komunikasi datanya menggunakan protokol Modbus RTU melalui antarmuka RS485 (Munawar, 2025). Dalam penelitian ini, PZEM-017 DC beserta resistor shunt digunakan untuk mengukur daya keluaran panel surya pada sisi arus searah sebagai variabel target.

2.2.4.5 Modul Pendukung Sistem

Selain komponen utama tersebut, sistem dilengkapi sejumlah modul pendukung. Karena PZEM-017 DC berkomunikasi menggunakan protokol Modbus RTU, diperlukan modul konverter MAX485 (TTL ke RS485) sebagai penghubung antara PZEM-017 DC dengan ESP32. Modul Real-Time Clock (RTC) digunakan untuk memberikan penanda waktu (*timestamp*) yang akurat pada setiap perekaman, sedangkan modul microSD berfungsi sebagai media penyimpanan data. Adapun modul step-down digunakan untuk menstabilkan tegangan catu daya ESP32 yang disuplai melalui adaptor daya.

Rincian seluruh komponen perangkat akuisisi data beserta fungsinya disajikan pada Tabel 2.1.

Tabel 2.1 Komponen Perangkat akuisisi data

No	Komponen	Fungsi
1	ESP32	Mikrokontroler atau unit pemrosesan utama
2	Sensor DHT22	Membaca suhu dan kelembapan udara
3	Sensor BH1750	Membaca intensitas cahaya
4	PZEM-017 DC	Mengukur tegangan, arus, dan daya keluaran DC
5	Resistor Shunt	Resistor pengukur arus DC untuk PZEM-017 DC

6	Modul MAX485	Konverter komunikasi TTL ke RS485 (Modbus)
7	Modul RTC	Memberikan penanda waktu (*timestamp*)
8	Modul microSD	Media penyimpanan data
9	Modul step-down	Penstabil tegangan catu daya ESP32
10	Adaptor Daya	Catu daya utama perangkat akuisisi
11	Kabel jumper dan kabel surya (solar PV)	Jalur data antar komponen serta penghantar arus DC dari panel surya

2.2.5 *Machine Learning*

Machine Learning (Pembelajaran Mesin) merupakan salah satu cabang fundamental dari disiplin ilmu Kecerdasan Buatan (AI) yang memberikan kapabilitas pada sistem komputasi untuk belajar secara otomatis dari sekumpulan data historis dan membuat prediksi tanpa harus diprogram secara eksplisit. Secara konseptual, teknologi ini bekerja dengan mengidentifikasi pola-pola yang tersembunyi di dalam sekumpulan data untuk memecahkan berbagai tugas analitik, mulai dari klastering, klasifikasi, hingga estimasi numerik (Jatnika et al., 2021). Di dalam arsitektur pembelajarannya, pemodelan *Machine Learning* yang paling umum dan banyak diteliti adalah *Supervised Learning* (Pembelajaran Terbimbing) (Najjar & Majeed Breesam, 2023).

Supervised Learning bekerja dengan melatih algoritma menggunakan *dataset* yang telah diberi label target atau kunci jawaban (*labeled output*), sehingga model dapat mempelajari hubungan antara variabel masukan (*input*) dan variabel keluaran (*output*). Melalui proses pelatihan tersebut, algoritma mengekstraksi pola dari data historis agar mampu melakukan generalisasi, yaitu menghasilkan prediksi yang akurat ketika dihadapkan pada data baru yang belum pernah dilihat sebelumnya. Pendekatan ini umumnya digunakan pada permasalahan yang memiliki pasangan data masukan dan target yang jelas, seperti pada kasus estimasi maupun prediksi nilai. Dalam penelitian ini, pendekatan *Supervised Learning* diterapkan karena *dataset* yang digunakan telah memiliki label target yang jelas.

2.2.5.1 Regresi

Di dalam koridor *Supervised Learning*, tugas komputasi diklasifikasikan secara ketat menjadi dua domain utama berdasarkan karakteristik tipe data pada variabel targetnya, yaitu Klasifikasi (*Classification*) dan Regresi (*Regression*). Masalah klasifikasi digunakan apabila target keluaran merepresentasikan label atau kelas kategorikal. Sebaliknya, metode regresi diimplementasikan secara mutlak apabila model diharuskan untuk memprediksi target keluaran yang berupa nilai numerik kontinu (Najjar & Majeed Breesam, 2023).

Metode regresi bertujuan membangun sebuah fungsi pendekatan (*approximation function*) yang mampu memetakan hubungan antara variabel prediktor dengan variabel target yang bersifat numerik kontinu. Fungsi tersebut digunakan untuk memperkirakan nilai keluaran berdasarkan pola hubungan yang dipelajari dari data, termasuk pola yang bersifat non-linear. Berbeda dengan klasifikasi yang keluarannya berupa kategori, keluaran pada regresi berupa nilai numerik yang bergerak secara dinamis dalam suatu rentang tertentu. Dalam penelitian ini, pendekatan regresi digunakan karena variabel targetnya berupa nilai numerik kontinu, sehingga seluruh model yang dievaluasi difungsikan sebagai *regressor*.

2.2.5.2 *Ensemble Learning*

Ensemble Learning adalah salah satu arsitektur pemodelan tingkat lanjut dalam *Machine Learning* yang mengintegrasikan beberapa model pembelajaran dasar (*base learners*) yang relatif lemah untuk membentuk satu kesatuan model prediktif yang jauh lebih tangguh, akurat, dan stabil (Abdelsattar et al., 2024). Pendekatan komputasi ini terbukti secara empiris sangat efektif dalam meminimalkan varians dan bias, serta mencegah model dari kondisi terlalu menghafal detail spesifik dari data latih yang berujung pada *overfitting*.

Secara matematis, prediksi akhir dari sebuah model *ensemble* merupakan hasil agregasi dari prediksi beberapa model dasar penyusunnya. Formulasinya dapat direpresentasikan sebagai berikut:

$$F(x) = \sum_{m=1}^{M} \gamma_m h_{m(x)} \tag{2.2}$$

Keterangan:

$F(x)$: Hasil prediksi akhir dari model *ensemble*

M : Jumlah total model dasar (seperti pohon keputusan) yang digunakan dalam pemodelan

γ_{m} : Bobot kontribusi yang diberikan pada setiap model penyusunnya

$h_{\mathrm{m}}(x)$: Hasil prediksi individual dari model dasar ke-m

x : Vektor fitur masukan (variabel independen)

Secara garis besar, terdapat dua teknik utama dalam arsitektur *Ensemble Learning* yang berbasis pada pohon keputusan (*decision tree*) untuk pemodelan data energi dan meteorologi, yaitu:

1. Bagging (*Bootstrap Aggregating*):

   Metode ini beroperasi dengan cara membangun sejumlah besar model pohon keputusan secara paralel dan sepenuhnya independen. Setiap pohon dilatih menggunakan *subset* data yang diambil secara acak dengan pengembalian (*random sampling with replacement*) dari himpunan data pelatihan. Untuk kasus pemodelan regresi, estimasi akhir diperoleh dengan cara mengambil nilai rata-rata (*averaging*) dari seluruh prediksi pohon tersebut untuk menghasilkan model yang stabil dan minim varians.

2. *Boosting*:

   Berbeda secara fundamental dengan bagging, teknik boosting membangun arsitektur model pohon keputusan secara berurutan (sequential). Setiap model pohon baru yang ditambahkan memiliki fungsi khusus untuk mengevaluasi, mempelajari, dan mengoreksi residu kesalahan (*error*) yang dihasilkan oleh susunan pohon-pohon sebelumnya. Evaluasi berkelanjutan melalui teknik koreksi gradien ini membuat algoritma boosting memiliki ketajaman akurasi yang sangat tinggi dan ketahanan yang superior dalam memetakan kompleksitas data yang non-linear.

2.2.5.3 *Random Forest Regressor*

Istilah Random Forest terbentuk dari dua kata, yaitu Random dan Forest. Kata Forest yang berarti hutan merujuk pada kumpulan pohon keputusan (*decision tree*) dalam jumlah banyak yang menyusun model, sebagaimana sebuah hutan tersusun atas banyak

pohon. Kata Random yang berarti acak merujuk pada dua bentuk pengacakan yang diterapkan ketika membangun setiap pohon, yaitu pengambilan sampel data secara acak dengan pengembalian (*bootstrap*) dan pemilihan sebagian fitur secara acak pada setiap pemecahan cabang. Dengan demikian, Random Forest dapat diartikan sebagai algoritma pembelajaran mesin yang membangun banyak pohon keputusan dari sampel data dan fitur yang diacak, kemudian menggabungkan seluruh prediksi pohon tersebut menjadi satu prediksi akhir. Pada persoalan regresi seperti estimasi daya, prediksi akhir diperoleh dengan merata-ratakan hasil seluruh pohon.

Secara sederhana, Random Forest dapat diibaratkan sebagai sebuah panel yang terdiri atas banyak ahli, di mana setiap ahli adalah satu pohon keputusan. Setiap pohon dilatih menggunakan potongan data dan sudut pandang yang sedikit berbeda, lalu memberikan prediksinya masing-masing, dan prediksi akhir diperoleh dengan merata-ratakan seluruh prediksi tersebut. Dengan cara ini, kesalahan yang mungkin dibuat oleh satu pohon dapat dikoreksi oleh pohon lainnya, sehingga hasil akhirnya cenderung lebih stabil dan tidak mudah terpengaruh oleh data yang menyimpang. Prinsip inilah yang membuat Random Forest dikenal andal dan tahan terhadap overfitting.

Random Forest Regressor (RFR) adalah varian dari algoritma *ensemble learning* yang dirancang secara spesifik untuk menyelesaikan masalah regresi, yakni memprediksi target variabel yang bernilai numerik dan kontinu, seperti nilai daya listrik atau temperature. Algoritma ini dibangun menggunakan metode *Bagging* (*Bootstrap Aggregating*), yang bekerja dengan cara membangkitkan dan melatih ratusan pohon keputusan (*decision trees*) secara paralel dan independen. Kemampuan *Random Forest Regressor* dalam memetakan interaksi non-linear yang kompleks sangat ideal untuk pemodelan iklim mikro. Hal ini divalidasi pada pengujian prediksi parameter meteorologi, di mana metode *Bagging* pada algoritma ini terbukti sangat tangguh dalam mereduksi *noise* saat dihadapkan pada anomali cuaca yang fluktuatif (Segovia et al., 2023).

Secara konseptual, *Random Forest* merupakan metode pembelajaran terbimbing (*supervised learning*) yang dibangun dari sekumpulan pohon keputusan (*decision tree*) sehingga membentuk sebuah 'hutan' (*forest*). Setiap pohon dibangun dari titik akar (*root*) hingga daun (*leaf*) untuk menentukan keluaran, dan secara umum semakin banyak jumlah

pohon yang digunakan akan meningkatkan akurasi model yang dihasilkan (Jatnika et al., 2025).

Dalam proses pembentukan setiap pohon regresi (*Classification and Regression Trees* / CART), algoritma ini tidak menggunakan *Gini Impurity* atau *Entropy* seperti pada kasus klasifikasi. Sebagai gantinya, pencarian titik potong percabangan (*node split*) terbaik dieksekusi dengan mencari fitur yang menghasilkan penurunan varians terbesar, atau dengan meminimalkan nilai *Sum of Squared Errors* (SSE). Secara matematis, evaluasi pemisahan *node* induk menjadi dua *node* anak (*child nodes*) dirumuskan sebagai berikut:

$$SSE = \sum_{i \in R_1} (y_i - \overline{y_1})^2 + \sum_{i \in R_2} (y_i - \overline{y_2})^2 \tag{2.3}$$

Keterangan:

SSE : Total kuadrat galat dari pemisahan *node*

R_1 : Himpunan data pada cabang anak pertama (kiri)

R_2 : Himpunan data pada cabang anak kedua (kanan)

y_i : Nilai daya aktual pada data ke-i

$\overline{y_1}$: Nilai rata-rata daya aktual pada cabang anak pertama

$\overline{y_2}$: Nilai rata-rata daya aktual pada cabang anak kedua

Nilai SSE pada Persamaan (2.3) dievaluasi untuk setiap kandidat nilai ambang (*threshold*) yang tersedia pada node yang sedang dipecah. Kandidat nilai ambang tersebut tidak ditentukan secara sembarang, melainkan diturunkan langsung dari data pada node tersebut. Prosesnya dilakukan dengan mengurutkan nilai fitur secara menaik, mengambil nilai-nilai yang berbeda (unik), kemudian menghitung titik tengah dari setiap pasangan nilai yang berurutan. Titik tengah digunakan sebagai perwakilan karena setiap nilai pemisah yang terletak di antara dua nilai fitur yang berurutan akan menghasilkan pengelompokan data yang identik, sehingga seluruh kemungkinan pemisahan pada rentang tersebut cukup diwakili oleh satu nilai yang berjarak sama terhadap kedua nilai fitur tersebut. Secara matematis, kandidat nilai ambang dirumuskan sebagai berikut:

$$t_k = \frac{x_{(k)} + x_{(k+1)}}{2}, \quad k = 1,2, \dots, m - \tag{2.4}$$

Keterangan:

t_k : kandidat nilai ambang (*threshold*) ke-k

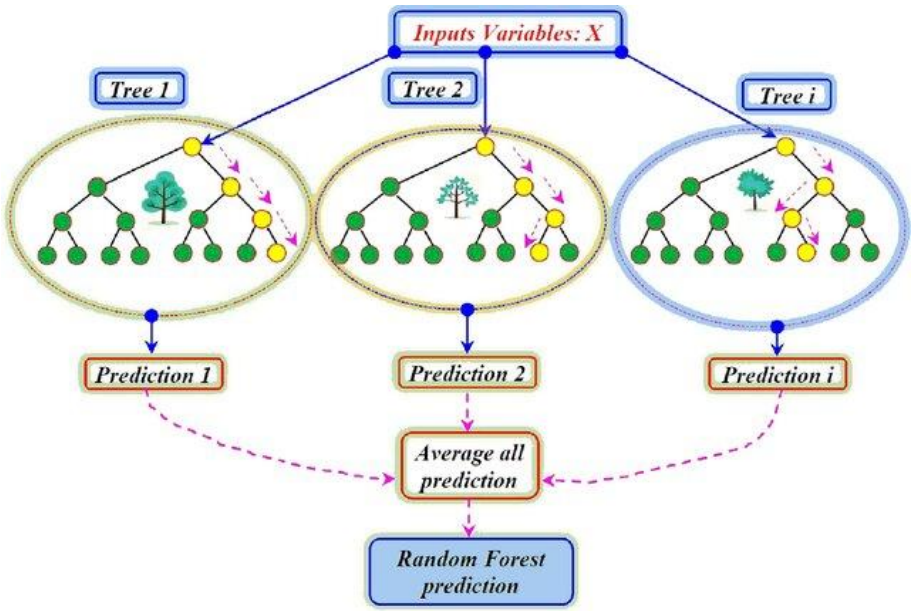
$x_{(k)}$: nilai fitur unik ke-k setelah data diurutkan secara menaik

$x_{(k+1)}$: nilai fitur unik pada urutan setelahnya

m : jumlah nilai fitur unik pada node yang sedang dipecah

Berdasarkan Persamaan (2.4), sebuah node yang memiliki m nilai fitur unik akan menghasilkan sebanyak m-1 kandidat nilai ambang. Seluruh kandidat tersebut dievaluasi menggunakan Persamaan (2.3), dan kandidat yang menghasilkan nilai SSE terkecil dipilih sebagai titik pemecahan node. Mekanisme penentuan kandidat nilai ambang ini berlaku umum pada algoritma berbasis pohon keputusan, termasuk *XGBoost*, dengan perbedaan terletak pada kriteria yang digunakan untuk memilih kandidat terbaik.

Setelah seluruh pohon regresi berhasil dibangun secara utuh hingga batas *max_depth* tertentu, tahap selanjutnya adalah melakukan agregasi. Berbeda dengan varian *classifier* yang menggunakan sistem pemungutan suara terbanyak (*majority voting*), agregasi hasil akhir pada *Random Forest Regressor* dilakukan dengan menghitung nilai rata-rata (*averaging*) dari seluruh hasil prediksi pohon keputusan individu. Secara visual, struktur pemrosesan paralel dan metode agregasi pada algoritma ini dapat dilihat pada Gambar 2.1.



Gambar 2.11 Arsitektur dari *Random Forest* (Hatamiafkoueieh et al., 2023)

Gambar 2.1 memperlihatkan arsitektur algoritma *Random Forest* yang tersusun atas sejumlah pohon keputusan (decision tree) yang dilatih secara paralel dan saling independen. Setiap pohon dibangun dari subset data hasil pengambilan sampel acak dengan pengembalian (bootstrap) serta pemilihan sebagian fitur secara acak pada tiap percabangan, sehingga terbentuk pohon-pohon yang beragam. Pada persoalan regresi, keluaran akhir *Random Forest* diperoleh dengan merata-ratakan hasil prediksi seluruh pohon, sehingga model menjadi lebih stabil dan lebih tahan terhadap overfitting dibandingkan penggunaan satu pohon tunggal.

Secara matematis, prediksi akhir dari sekumpulan pohon dirumuskan sebagai berikut:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \tag{2.5}$$

Keterangan:

$\hat{y}$: Nilai prediksi akhir daya keluaran (Watt)
B : Jumlah total pohon keputusan (*decision trees*) dalam model
$f_b(x)$: Nilai prediksi yang dihasilkan oleh pohon ke $-b$
x : Variabel *input* atau parameter iklim mikro (suhu, kelembapan, cahaya)

Melalui kombinasi peminimalan *Sum of Squared Errors* (SSE) pada setiap cabang dan sistem agregasi rata-rata di akhir proses, algoritma *Random Forest Regressor* terbukti mampu membaca korelasi antara perubahan cuaca terhadap total keluaran daya listrik dengan tingkat galat yang sangat rendah, sehingga sangat andal digunakan untuk mengestimasi fenomena *power drop* di PLTS (Shang, 2025).

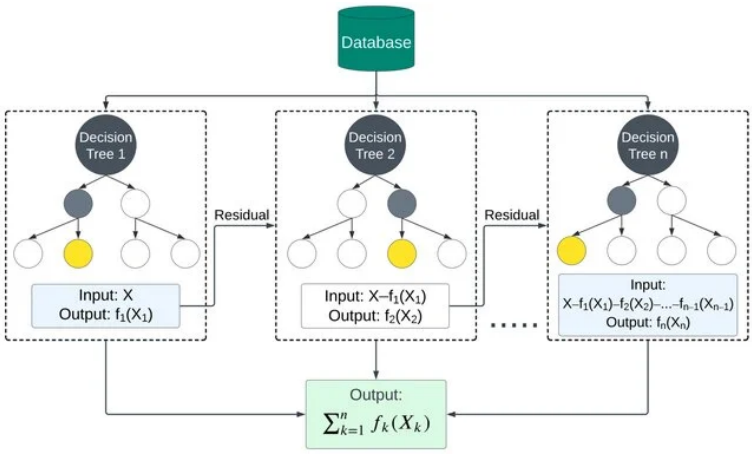
2.2.5.4 *Extreme Gradient Boosting (XGBoost) Regressor*

Adapun XGBoost merupakan kependekan dari Extreme Gradient Boosting, sebuah nama yang mencerminkan tiga komponen utama pembentuknya, yaitu Boosting, Gradient, dan Extreme. Komponen Boosting yang berarti penguatan merujuk pada teknik membangun banyak pohon keputusan secara berurutan, di mana setiap pohon baru bertugas memperbaiki kesalahan yang masih tersisa dari pohon sebelumnya. Komponen Gradient yang berarti gradien merujuk pada cara model menentukan arah perbaikan kesalahan tersebut, yaitu dengan memanfaatkan turunan (gradien) dari fungsi kerugian.

Sementara itu, komponen Extreme yang berarti ekstrem merujuk pada berbagai optimasi tambahan yang membuat algoritma ini bekerja lebih cepat dan lebih efisien, seperti penerapan regularisasi untuk mencegah model menjadi terlalu kompleks. Dengan demikian, XGBoost dapat diartikan sebagai algoritma boosting berbasis gradien yang membangun pohon keputusan secara bertahap dan berurutan sambil menerapkan regularisasi, sehingga menghasilkan prediksi yang teliti sekaligus tetap terkendali dari risiko overfitting.

Secara sederhana, XGBoost dapat diibaratkan sebagai seorang siswa yang belajar dari kesalahannya secara bertahap. Model dibangun dari banyak pohon keputusan yang dibuat satu per satu secara berurutan, di mana setiap pohon baru berfokus memperbaiki kesalahan yang masih tersisa dari pohon sebelumnya. Proses perbaikan bertahap ini dikendalikan oleh laju pembelajaran agar setiap pohon hanya memberikan koreksi dalam porsi kecil, sehingga model belajar secara hati-hati dan terarah. Berkat mekanisme ini, XGBoost mampu menangkap pola yang kompleks di dalam data dengan tingkat ketelitian yang tinggi.

Extreme Gradient Boosting (XGBoost) merupakan implementasi mutakhir dari algoritma *Gradient Boosting Decision tree* (GBDT) yang membangun model prediktif secara aditif dan sekuensial. Dalam arsitektur ini, setiap pohon keputusan baru secara spesifik dirancang untuk mengoreksi kesalahan (*residual error*) dari akumulasi pohon-pohon sebelumnya (Chen & Guestrin, 2016). Pembaruan berurutan inilah yang membedakannya secara fundamental dari metode *bagging*. Secara visual, alur pemrosesan sekuensial pada arsitektur *XGBoost* ini direpresentasikan pada Gambar 2.2.



Gambar 2.22 Arsitektur dari *XGBoost* (Taheri & Sobanjo, 2024)

Gambar 2.2 memperlihatkan arsitektur algoritma Extreme Gradient Boosting (*XGBoost*) yang membangun pohon keputusan secara bertahap atau sekuensial melalui mekanisme boosting. Berbeda dengan *Random Forest* yang melatih pohon secara paralel, setiap pohon baru pada *XGBoost* dilatih untuk memperbaiki galat (residual) dari gabungan pohon-pohon sebelumnya. Prediksi akhir merupakan penjumlahan terboboti dari seluruh pohon, sehingga model secara bertahap menekan kesalahan prediksi hingga diperoleh hasil yang semakin akurat.

Secara matematis, fungsi prediksi akhir dari model ansambel *XGBoost* untuk sebuah *dataset* dengan n observasi dan K jumlah pohon diformulasikan sebagai berikut:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i) \tag{2.6}$$

Keterangan:

$\widehat{y_i}$: Nilai prediksi akhir untuk observasi ke-i

K : Jumlah total pohon keputusan dalam model

$f_k(x_i)$: Fungsi pohon keputusan independen ke-k

Proses boosting tidak dimulai dari keadaan tanpa prediksi, melainkan dari sebuah prediksi awal (base score) yang bernilai sama untuk seluruh observasi. Pada kasus regresi dengan fungsi kerugian kuadratik, prediksi awal tersebut ditetapkan sebagai nilai rata-rata target pada data latih, sebagaimana dinyatakan sebagai berikut:

$$\hat{y}^{(0)} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.7}$$

Keterangan:

$\hat{y}^{(0)}$: Nilai prediksi awal (*base score*) sebelum pohon pertama dibangun

n : Jumlah observasi pada data pemodelan

y_i : Nilai daya aktual pada observasi ke-i

Berbeda secara fundamental dengan algoritma ansambel tradisional, *XGBoost* mengintegrasikan mekanisme regularisasi eksplisit ke dalam fungsi objektifnya untuk mengendalikan kompleksitas model secara ketat dan mencegah terjadinya fenomena

overfitting selama proses pelatihan komputasi. Fungsi objektif yang harus diminimalkan pada algoritma ini didefinisikan secara ekuivalen sebagai:

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.8}$$

Dengan parameter regularisasi struktural yang dihitung melalui persamaan:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} w_j^2 \tag{2.9}$$

Keterangan:

$l(y_i, \hat{y}_i)$: Fungsi kerugian (*loss function*) antara nilai aktual dan prediksi
$\Omega(f_k)$: Fungsi regularisasi untuk pohon ke-k
T_k : Jumlah total daun pada struktur pohon
w_j : Bobot skor pada setiap daun ke-j
γ : Penalti terhadap penambahan jumlah daun baru
λ : Parameter regularisasi L2 (*Ridge*)

Untuk mencapai tingkat konvergensi optimasi yang sangat cepat dan efisien secara komputasi, *XGBoost* mengeksekusi ekspansi Taylor orde kedua secara langsung pada fungsi kerugiannya. Melalui pendekatan ini, pembaruan fungsi objektif pada iterasi ke-t dapat diaproksimasi menjadi:

$$Obj^{(t)} \approx \sum_{i=1}^{n} \left[l\left(y_i, \widehat{y_i^{(t-1)}}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2.10}$$

Keterangan:

$Obj^{(t)}$: Nilai fungsi objektif pada iterasi pohon ke-t
$l\left(y_i, \widehat{y_i^{(t-1)}}\right)$: Fungsi kerugian (*loss function*) dari prediksi iterasi sebelumnya
g_i : Turunan parsial orde pertama (*Gradien*) dari fungsi kerugian
h_i : Turunan parsial orde kedua (*Hessian*) dari fungsi kerugian
$\Omega(f_t)$: Parameter regularisasi (penalti kompleksitas model) pada pohon ke-t

Pada kasus regresi dengan fungsi kerugian kuadratik, gradien orde pertama dan gradien orde kedua (Hessian) untuk setiap observasi diperoleh dari penurunan fungsi kerugian terhadap prediksi pada iterasi sebelumnya, sehingga dirumuskan sebagai berikut:

$$g_i = \hat{y}_i^{(t-1)} - y_i, \quad h_i = 1 \tag{2.11}$$

Keterangan:

g_i : Nilai fungsi objektif pada iterasi pohon ke-t

h_i : Fungsi kerugian (*loss function*) dari prediksi iterasi sebelumnya

$\hat{y}_i^{(t-1)}$: Nilai prediksi pada iterasi sebelumnya untuk observasi ke-i

y_i : Nilai daya aktual pada observasi ke-i

Dengan mendefinisikan himpunan indeks observasi pada daun ke-j dan menyederhanakan total gradien sebagai $G_j = \sum_{i \in I_j} g_i$ serta total *Hessian* sebagai $H_j = \sum_{i \in I_j} h_i$, struktur fungsi objektif tersebut dapat direduksi menjadi bentuk kuadratik independen yang pada akhirnya menghasilkan formulasi bobot optimal daun sebesar:

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{2.12}$$

Selanjutnya, untuk menentukan topologi pemecahan pohon (*splitting*) yang paling ideal, *XGBoost* menerapkan algoritma pencarian rakus (*greedy algorithm*) yang mengevaluasi kelayakan setiap skenario pemisahan cabang dengan menghitung nilai peningkatan metrik (*Gain*) melalui persamaan:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma \tag{2.13}$$

Keterangan:

$Gain$: Nilai peningkatan metrik untuk mengevaluasi kelayakan pemisahan cabang

G_L, H_L : Total gradien dan *Hessian* pada cabang anak pertama (kiri)

G_R, H_R : Total gradien dan *Hessian* pada cabang anak kedua (kanan)

γ : Total gradien dan *Hessian* pada cabang kiri setelah pemisahan

λ : Parameter regularisasi L2 (*Ridge*)

Apabila nilai *Gain* yang dihasilkan dari suatu percabangan bernilai negatif atau tidak mampu melampaui ambang batas penalti kompleksitas γ, maka algoritma secara otomatis akan menghentikan pemisahan cabang tersebut.

Setelah struktur pohon terbentuk dan bobot optimal setiap daun diperoleh, prediksi model diperbarui secara bertahap dengan menambahkan kontribusi pohon baru yang telah dikalikan dengan laju pembelajaran (*learning_rate*), sebagaimana dinyatakan pada Persamaan (2.14).

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \times w_j \tag{2.14}$$

Dengan y topi pangkat t adalah prediksi pada iterasi ke-t, y topi pangkat t dikurangi satu adalah prediksi pada iterasi sebelumnya, η adalah laju pembelajaran (*learning_rate*), dan w indeks j adalah bobot optimal pada daun ke-j.

Keandalan arsitektur matematis dan mekanisme pemangkasan (*pruning*) ini tidak hanya kokoh secara komputasi teoretis, tetapi juga telah divalidasi keberhasilannya pada berbagai implementasi pemodelan regresi dan deret waktu di dunia nyata. Algoritma *XGBoost* yang berbasis sistem *ensemble* ini terbukti sangat tangguh dalam memproses kumpulan data numerik yang kompleks, terutama pada himpunan data yang memiliki banyak dimensi fitur. Lebih lanjut, kemampuan inheren dari algoritma ini dalam menahan laju *overfitting* menjadikannya salah satu metode regresi dengan tingkat akurasi yang sangat superior, sehingga sangat kompatibel untuk memodelkan variabel-variabel pengganggu eksternal yang bersifat non-linear dan fluktuatif (Idham & Yosrita, 2025).

2.2.6 Pra-pemrosesan Data (Data *Preprocessing*)

Sebelum data digunakan untuk melatih model, diperlukan serangkaian tahapan pra-pemrosesan data (data *preprocessing*) guna menjaga kualitas dan konsistensi *dataset*. Beberapa teknik pra-pemrosesan data yang diterapkan dalam penelitian ini diuraikan pada subbab berikut.

2.2.6.1 *Interpolasi Linear*

Interpolasi linear merupakan salah satu metode estimasi nilai data yang hilang atau tidak tercatat dengan mengasumsikan bahwa perubahan nilai di antara dua titik data yang diketahui mengikuti pola garis lurus. Metode ini bekerja dengan menarik garis lurus di

antara dua titik data yang saling berdekatan, kemudian nilai pada titik yang kosong diperkirakan berdasarkan posisinya di sepanjang garis tersebut. Karena kesederhanaan dan efektivitasnya, interpolasi linear banyak digunakan untuk menangani data hilang pada data deret waktu (*time series*), khususnya ketika data yang hilang berada dalam rentang waktu yang pendek sehingga pola antar titik cenderung stabil (Mohamad et al., 2022).

Secara matematis, nilai hasil interpolasi linear dihitung menggunakan persamaan berikut:

$$y = y_1 + \frac{x - x_1}{x_2 - x_1}(y_2 - y_1) \tag{2.15}$$

Keterangan:

x_1, y_1 : Titik data sebelum *missing value*

x_2, y_2 : Titik data sesudah *missing value*

x : Titik (waktu) dari nilai yang dicari

y : Nilai data yang dicari (hasil interpolasi)

Nilai yang hilang diperoleh dengan menghitung proporsi jarak titik yang dicari terhadap dua titik data yang mengapitnya, lalu proporsi tersebut dikalikan dengan selisih nilai kedua titik dan ditambahkan pada nilai titik awal.

2.2.6.2 Imputasi Data Daya Berdasarkan Kemiripan Kondisi Cahaya

Imputasi data merupakan proses pengisian nilai yang hilang atau tidak valid dengan nilai estimasi yang dianggap paling mewakili kondisi sebenarnya. Pada pemodelan daya surya, pembacaan sensor daya sering mengalami gangguan sehingga nilai yang tercatat menjadi tidak wajar atau bernilai nol. Apabila dibiarkan, data tidak valid tersebut dapat membuat model mempelajari pola yang keliru sehingga menurunkan akurasi estimasi.

Salah satu metode imputasi yang efektif untuk data daya pembangkit surya adalah pengisian nilai berdasarkan kemiripan kondisi cahaya (intensitas cahaya dalam satuan Lux). Prinsipnya, daya keluaran panel surya sangat dipengaruhi oleh intensitas cahaya, sehingga baris data dengan intensitas cahaya yang serupa cenderung menghasilkan nilai daya yang serupa pula. Berdasarkan prinsip tersebut, setiap nilai daya yang tidak valid diisi dengan rata-rata daya dari baris lain yang memiliki intensitas cahaya serupa (dalam

rentang toleransi tertentu) dan bernilai daya lebih besar dari nol, sebagaimana dirumuskan berikut:

$$\hat{P}_i = \frac{1}{|N_i|} \sum_{j \in N_i} P_j \tag{2.16}$$

Keterangan:

$\hat{P}_i$: Nilai daya hasil imputasi pada baris ke-i (Watt)

N_i : Himpunan baris pembanding yang intensitas cahayanya serupa dengan baris ke-i

$| N_i |$: Jumlah baris pembanding di dalam himpunan N_i

j : Indeks baris pembanding yang memenuhi syarat

P_j : Nilai daya pada baris pembanding ke-j (Watt)

Adapun himpunan baris pembanding N_i yang digunakan pada rumus (2.16) didefinisikan sebagai:

$$N_i = \{ j : 0.85\, L_i \leq L_j \leq 1.15\, L_i,\ P_j > 0 \} \tag{2.17}$$

Keterangan:

N_i : Himpunan baris pembanding yang intensitas cahayanya serupa dengan baris ke-i

j : Indeks baris pembanding yang memenuhi syarat ke-i

L_i : Intensitas cahaya pada baris ke-i (Lux)

L_j : Intensitas cahaya pada baris pembanding ke-j (Lux)

$P_j > 0$: Syarat bahwa daya pembanding harus bernilai lebih besar dari nol

Pemilihan intensitas cahaya sebagai dasar imputasi didasarkan pada kenyataan bahwa cahaya merupakan faktor paling dominan terhadap daya keluaran. Imputasi daya pembangkit surya terbukti paling efektif apabila dilakukan berbasis iradiasi matahari, karena iradiasi merupakan prediktor dominan dengan tingkat kepentingan gabungan mencapai sekitar 94 persen (Costa et al., 2024).

2.2.6.3 Normalisasi Data

Data pada pemodelan regresi umumnya terdiri atas beberapa variabel dengan satuan dan rentang nilai yang berbeda-beda. Perbedaan skala yang cukup besar antar variabel tersebut dapat menyebabkan variabel dengan nilai yang lebih besar mendominasi proses perhitungan jarak atau bobot pada algoritma pembelajaran mesin. Untuk mengatasi hal ini, dilakukan proses normalisasi data, yaitu tahap pra-pemrosesan yang bertujuan menyeragamkan rentang nilai seluruh variabel agar berada pada skala yang sebanding (Ambarwari et al., 2020).

Teknik normalisasi yang digunakan dalam penelitian ini adalah normalisasi Min-Max (*Min-Max Normalization*). Teknik ini dipilih karena memiliki beberapa keunggulan yang sesuai dengan karakteristik data penelitian. Pertama, normalisasi Min-Max mempertahankan hubungan proporsional antar nilai pada data asli karena transformasinya bersifat linier, sehingga tidak mengubah distribusi data (Nasution et al., 2019). Kedua, rentang keluaran yang tetap berada pada interval [0, 1] memudahkan interpretasi hasil normalisasi dan konsisten untuk seluruh variabel. Ketiga, teknik ini sangat efektif apabila data tidak mengandung pencilan (*outlier*) ekstrem dalam jumlah besar (Ambarwari et al., 2020). Secara matematis, normalisasi Min-Max dirumuskan sebagai berikut:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.18}$$

Keterangan:

X' : Nilai hasil normalisasi

X : Nilai asli data yang akan dinormalisasi

X_{min} : Nilai minimum pada variabel tersebut

X_{max} : Nilai maksimum pada variabel tersebut

Melalui persamaan tersebut, nilai terkecil pada suatu variabel akan bernilai 0 dan nilai terbesar akan bernilai 1, sedangkan nilai lainnya akan berada di antara rentang tersebut secara proporsional.

Setelah model menghasilkan prediksi, nilai keluaran yang masih berada dalam skala ternormalisasi perlu dikembalikan ke satuan aslinya agar dapat diinterpretasikan secara

fisis. Proses ini disebut transformasi balik (*inverse transform*), yang dirumuskan sebagai berikut:

$$X = X' \times (Xmax - Xmin) + Xmin \quad (2.19)$$

Keterangan:

X : Nilai dalam skala asli (hasil transformasi balik)

X' : Nilai ternormalisasi (keluaran model)

X_{min} : Nilai minimum yang digunakan saat normalisasi

X_{max} : Nilai maksimum yang digunakan saat normalisasi

Dengan diterapkannya normalisasi Min-Max, seluruh variabel masukan memiliki rentang nilai yang seragam sehingga proses pelatihan model dapat berjalan lebih stabil dan tidak didominasi oleh variabel berskala besar.

2.2.6.4 Data Time Series

Data deret waktu (*time series*) adalah serangkaian data yang dikumpulkan atau dicatat secara berurutan berdasarkan interval waktu tertentu. Berbeda dengan data biasa yang setiap barisnya berdiri sendiri, data deret waktu memiliki ketergantungan temporal, yaitu nilai pada suatu waktu cenderung berkorelasi dengan nilai pada waktu-waktu yang berdekatan (autokorelasi). Karakteristik ini banyak dimanfaatkan dalam berbagai penelitian prediksi berbasis *machine learning* (Rianto & Yunis, 2021).

Dalam penelitian ini, meskipun data direkam secara berurutan menurut waktu, estimasi daya dilakukan berdasarkan parameter cuaca pada waktu yang sama (bersifat sesaat), bukan berdasarkan urutan nilai pada waktu-waktu sebelumnya. Oleh karena itu, penelitian ini tidak menggunakan transformasi jendela geser (sliding window) maupun fitur autoregresif, sehingga hubungan antara parameter cuaca dan daya diperlakukan sebagai persoalan regresi antar variabel pada tiap titik waktu.

Karena estimasi daya tidak bergantung pada nilai daya di waktu sebelumnya, keterikatan antar waktu (autokorelasi) tidak dimanfaatkan sebagai dasar prediksi. Dengan demikian, seluruh data digunakan bersama dalam proses pemodelan sebagai persoalan regresi antara parameter cuaca dan daya keluaran pada tiap titik waktu.

2.2.7 Metrik Evaluasi Model

Untuk mengukur dan membandingkan secara objektif tingkat keandalan prediksi antar model regresi, diperlukan instrumen pengukuran matematis standar yang dikhususkan untuk pemodelan regresi. Kinerja sebuah model regresi dievaluasi berdasarkan seberapa jauh antara nilai yang diprediksi oleh model terhadap nilai aktualnya, serta seberapa baik model tersebut mampu merepresentasikan variabilitas data secara keseluruhan (Chicco et al., 2021). Tiga parameter metrik evaluasi utama yang digunakan dalam penelitian ini adalah:

2.2.7.1 Mean Absolute Error (MAE)

MAE merupakan metrik fundamental yang mengukur nilai rata-rata dari selisih absolut antara nilai daya prediksi dengan nilai aktual. Metrik ini memberikan gambaran yang linier dan intuitif mengenai seberapa besar rata-rata pergeseran (*error*) prediksi model tanpa mempertimbangkan arah penyimpangan (positif atau negatif). Formulasi matematis MAE didefinisikan sebagai:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.20}$$

Keterangan :

n : Mewakili jumlah total observasi data

y_i : Nilai aktual

$\hat{y}_i$: Nilai hasil prediksi model

2.2.7.2 Root Mean Square Error (RMSE)

RMSE mengukur akar dari rata-rata kuadrat kesalahan antara nilai prediksi dan nilai aktual. Berbeda dengan MAE, proses pengkuadratan pada perhitungan RMSE memberikan penalti kuadratik (eksponensial) yang jauh lebih besar terhadap penyimpangan atau *error* yang ekstrem (Chicco et al., 2021). Karakteristik ini menjadikan RMSE sebagai metrik yang sangat sensitif terhadap data pencilan (*outlier*), yang dapat

muncul akibat adanya anomali atau pencilan pada data. Formulasi matematis RMSE adalah:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.21}$$

Keterangan :

$\boldsymbol{n}$: Mewakili jumlah total observasi data

y_i : Nilai aktual

$\hat{y}_i$: Nilai hasil prediksi model

2.2.7.3 Koefisien Determinasi ($\boldsymbol{R^2}$)

Berbeda dengan MAE dan RMSE yang berfokus pada kuantifikasi *error*, R^2 atau *R-Squared* merupakan metrik statistik yang mengukur tingkat kecocokan data (*goodness-of-fit*). Metrik ini mengevaluasi seberapa besar persentase variabilitas daya keluaran aktual yang dapat dijelaskan secara proporsional oleh model fitur cuaca independen. Rentang nilai R^2 berada di antara 0 hingga 1, di mana nilai yang semakin mendekati 1 merepresentasikan bahwa model prediktif memiliki kinerja generalisasi yang sempurna (Chicco et al., 2021). Formulasi matematis dari Koefisien Determinasi Adalah:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.22}$$

Keterangan :

R^2 : Koefisien determinasi

n : Jumlah total data observasi

y_i : Nilai keluaran aktual pada data ke-i

$\hat{y}_i$: Nilai hasil prediksi pada data ke-i

$\bar{y}$: Nilai rata-rata dari keseluruhan data aktual

Secara teoretis, kinerja suatu model prediksi diukur berdasarkan tingkat kesalahan (error) antara nilai prediksi dengan nilai aktual, di mana semakin kecil nilai error yang dihasilkan menunjukkan tingkat akurasi model yang semakin baik. Mean Absolute Error (MAE) merepresentasikan rata-rata selisih absolut antara nilai prediksi dan nilai aktual,

sedangkan Root Mean Square Error (RMSE) merepresentasikan akar dari rata-rata kuadrat kesalahan yang memberikan bobot penalti lebih besar terhadap kesalahan berukuran besar (Kusuma et al., 2023).

2.2.8 *Hyperparameter*

Hyperparameter merupakan nilai parameter model yang tidak dipelajari secara otomatis selama proses pelatihan, melainkan ditetapkan terlebih dahulu sebelum pelatihan berlangsung. Nilai *hyperparameter* menentukan tingkat kompleksitas serta perilaku model selama proses pembelajaran, misalnya jumlah pohon keputusan yang dibangun, kedalaman maksimum tiap pohon, jumlah minimum sampel pada sebuah simpul, maupun laju pembelajaran pada algoritma berbasis *boosting*. Penetapan nilai *hyperparameter* berpengaruh terhadap keseimbangan antara kemampuan model dalam mengenali pola pada data dan risiko terjadinya *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri pada data pelatihan sehingga kehilangan kemampuan dalam mengenali pola pada data baru.(Renaldi & Maharani, 2022)

2.2.9 *Feature importance*

Secara istilah, *Feature Importance* dapat dipahami dari dua kata penyusunnya, yaitu *Feature* yang berarti fitur dan *Importance* yang berarti kepentingan. Kata *Feature* mengacu pada variabel masukan yang digunakan model, dalam penelitian ini berupa suhu udara, kelembapan udara, dan intensitas cahaya. Sementara kata *Importance* mengacu pada seberapa besar pengaruh atau kontribusi sebuah fitur terhadap hasil prediksi. Dengan demikian, *Feature Importance* dapat diartikan sebagai teknik untuk mengukur dan memberi peringkat tingkat kontribusi setiap fitur masukan terhadap keluaran sebuah model. Pada algoritma berbasis pohon keputusan seperti *Random Forest* dan *XGBoost*, nilai ini dihitung dari seberapa besar sebuah fitur membantu menurunkan kesalahan prediksi setiap kali fitur tersebut digunakan untuk memecah cabang pohon. Semakin besar nilai Feature Importance suatu fitur, semakin dominan pengaruh fitur tersebut terhadap keluaran model. Dalam penelitian ini, analisis *Feature Importance* digunakan untuk mengidentifikasi parameter iklim mikro yang paling berperan terhadap fluktuasi daya keluaran PLTS Apung.

Feature importance merupakan teknik untuk mengukur tingkat kontribusi atau pengaruh setiap fitur masukan terhadap hasil prediksi sebuah model (Zhang et al., 2026).

Pada algoritma berbasis pohon keputusan seperti *Random Forest* dan *XGBoost*, nilai *Feature importance* umumnya dihitung berdasarkan pendekatan *Mean Decrease in Impurity* (MDI), yaitu total penurunan ketidakmurnian (*impurity*) yang disumbangkan oleh suatu fitur setiap kali fitur tersebut digunakan untuk memecah simpul (*node*) di seluruh pohon. Untuk kasus regresi, ketidakmurnian sebuah simpul diukur melalui variansi (galat kuadrat) nilai target pada simpul tersebut. Kontribusi sebuah fitur f pada satu pohon keputusan dirumuskan sebagai berikut:

$$Imp(f) = \sum_{t \in \varphi(f)} \frac{N_t}{N} \Delta i(t) \tag{2.23}$$

Dengan penurunan *impurity* pada simpul t dinyatakan melalui persamaan berikut:

$$\Delta i(t) = i(t) - \frac{N_{t_L}}{N_t} i(t_L) - \frac{N_{t_R}}{N_t} i(t_R) \tag{2.24}$$

Pada model *ensemble* seperti *Random Forest*, nilai akhir kepentingan fitur diperoleh dengan merata-ratakan kontribusi fitur tersebut di seluruh T pohon, sebagaimana persamaan berikut:

$$Imp_{RF}(f) = \frac{1}{T} \sum_{j=1}^{T} I\,mp_j(f) \tag{2.25}$$

Selanjutnya, nilai kepentingan seluruh fitur dinormalisasi agar totalnya bernilai satu, sehingga kontribusi tiap fitur dapat dibandingkan secara relatif, sebagaimana persamaan berikut:

$$FI(f) = \frac{Imp(f)}{\sum_{k=1}^{p} I\,mp(k)} \tag{2.26}$$

Keterangan:

$Imp(f)$: nilai kepentingan (*importance*) fitur f

$\varphi(f)$: Himpunan simpul yang memecah data menggunakan fitur f

N : Jumlah total data

N_t : Jumlah data pada simpul t

$\Delta i(t)$: Penurunan *impurity* pada simpul t

$i(t)$: Nilai *impurity* (variansi) pada simpul t

N_{t_L}, N_{t_R} : Jumlah data pada simpul anak kiri dan kanan

$i(t_L), i(t_R)$: Nilai impurity (variansi) pada simpul anak kiri dan kanan

T : Jumlah pohon dalam *ensemble*

p : Jumlah keseluruhan fitur

$FI(f)$: Nilai *Feature importance* ternormalisasi

Semakin tinggi nilai *Feature importance* suatu fitur, semakin dominan pengaruh fitur tersebut terhadap keluaran model. Dalam penelitian ini, analisis *Feature importance* digunakan untuk mengidentifikasi parameter iklim mikro yang paling berperan terhadap fluktuasi daya keluaran PLTS Apung.

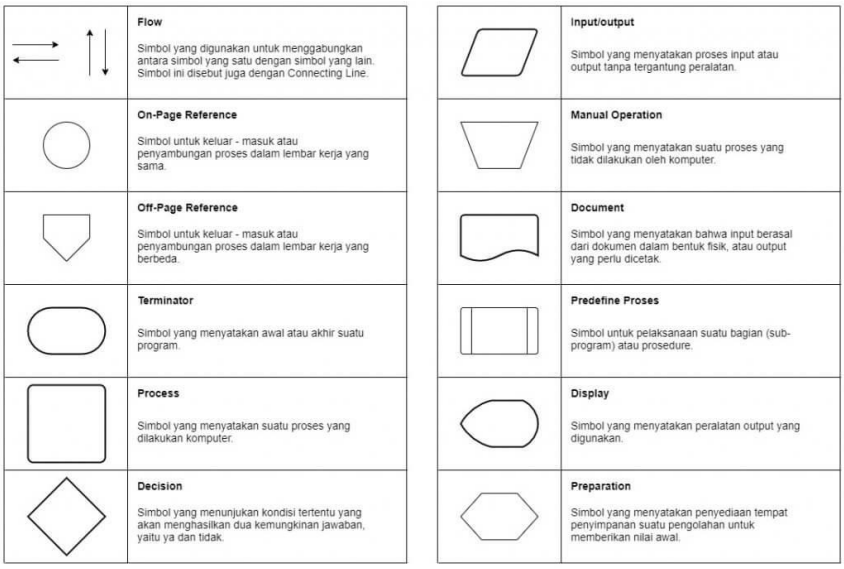
2.2.10 Python

Python merupakan bahasa pemrograman tingkat tinggi berkonsep *open-source* yang dieksekusi melalui metode interpretasi (*interpreted language*) dan pertama kali dirilis oleh Guido van Rossum pada tahun 1991. Bahasa pemrograman ini sangat populer karena memiliki sintaksis yang sederhana, intuitif, dan berorientasi objek, sehingga memudahkan pengembang dalam mengekspresikan algoritma logika yang kompleks menggunakan baris kode yang jauh lebih efisien dibandingkan bahasa pemrograman lainnya. Berkat tingkat fleksibilitas yang tinggi serta dukungan ekosistem pustaka (*library*) yang komprehensif, Python saat ini telah menjadi bahasa standar yang paling banyak diadopsi oleh industri dan akademisi untuk berbagai ranah pengembangan perangkat lunak, khususnya dalam bidang komputasi data ilmiah dan *Machine Learning* (Suryati et al., 2025).

2.2.11 Flowchart

Flowchart atau bagan alur merupakan diagram visual yang merepresentasikan langkah-langkah logis dan pengambilan keputusan dalam suatu program menggunakan simbol geometris yang saling terhubung oleh garis atau anak panah. Fungsi utama dari *flowchart* adalah menyederhanakan rangkaian prosedur yang kompleks menjadi informasi visual yang terstruktur guna menunjukkan jalannya sistem dari satu proses ke proses lainnya. Penggunaan bagan alur ini sangat krusial dalam pengembangan program karena dapat memperjelas fungsionalitas algoritma, meminimalisir risiko salah

penafsiran, serta menjembatani komunikasi antara pemahaman teknis dan non-teknis secara efektif.

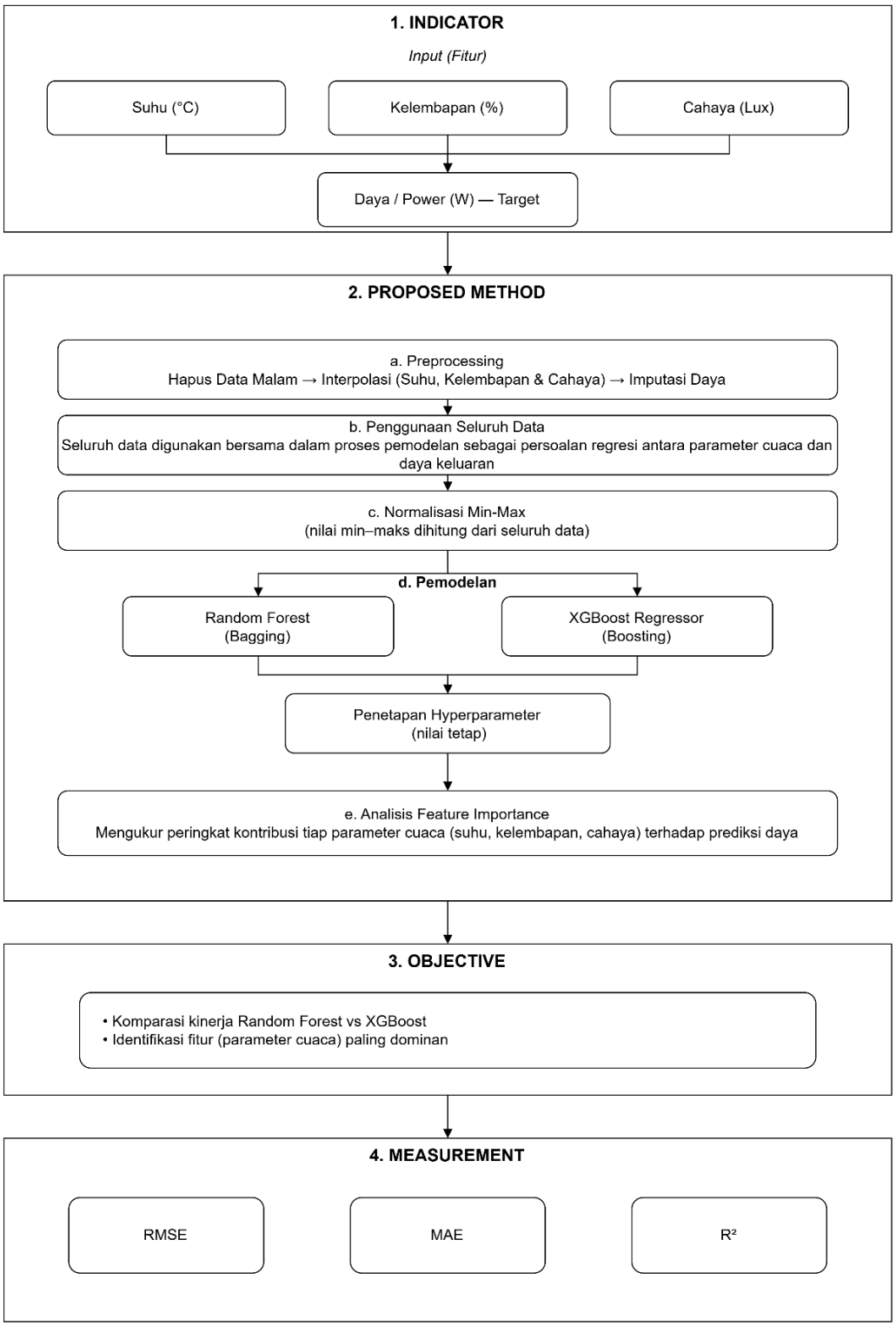


Gambar 2.3 Flowchart (Setiawan, 2021)

Gambar 2.3 memperlihatkan contoh diagram alir (flowchart) beserta simbol-simbol baku yang digunakan untuk menggambarkan urutan suatu proses secara visual. Setiap simbol memiliki makna tersendiri, misalnya simbol terminal untuk menandai awal dan akhir proses, simbol proses untuk menyatakan langkah pengolahan, serta simbol keputusan untuk menggambarkan percabangan kondisi. Notasi baku inilah yang menjadi acuan dalam penyusunan diagram alir tahapan penelitian pada bab selanjutnya.

2.3 Kerangka Pemikiran

Kerangka pemikiran merupakan alur berpikir yang menggambarkan tahapan penyelesaian permasalahan dalam penelitian secara sistematis, mulai dari penetapan indikator hingga pencapaian tujuan penelitian. Pada penelitian ini, kerangka pemikiran disusun untuk memberikan gambaran menyeluruh mengenai keterkaitan antara variabel masukan yang digunakan, metode yang diusulkan, tujuan yang hendak dicapai, serta parameter yang menjadi ukuran keberhasilannya. Adapun kerangka pemikiran penelitian ini secara ringkas ditunjukkan pada Gambar 2.4.



Gambar 2.4 Kerangka Pemikiran

Pada Gambar 2.4 merupakan kerangka pemikiran dari penelitian ini, untuk mengetahui penjelasan dari masing-masing point berikut penjelasannya sebagai berikut :

1. Indicator

   Indikator dalam penelitian ini terdiri atas variabel masukan (*input*/fitur) dan variabel keluaran (*target*). Variabel masukan dibatasi pada tiga parameter iklim mikro, yaitu Suhu Udara (°C), Kelembapan Udara (%), dan Intensitas Cahaya (Lux), sedangkan variabel target adalah Daya keluaran aktual (Watt). Ketiga parameter cuaca tersebut dipilih karena secara langsung memengaruhi besarnya daya yang dihasilkan panel surya. Seluruh data bersumber dari sistem akuisisi berbasis *Internet of Things* (IoT) yang terpasang pada PLTS Apung Danau IT-PLN.

2. Proposed Method

   Pada tahap ini, data parameter cuaca dan kelistrikan yang diperoleh dari sistem akuisisi diproses melalui serangkaian tahapan pemodelan *Machine Learning* sebagai berikut:

a. Pra-pemrosesan Data (*Preprocessing*), yaitu tahapan awal untuk mempersiapkan data mentah agar layak digunakan dalam proses pelatihan model. Tahapan ini meliputi penghapusan data periode malam hari yang tidak menghasilkan daya, interpolasi linear untuk mengisi nilai sensor cuaca yang hilang atau tidak wajar pada variabel suhu, kelembapan, dan intensitas cahaya, serta imputasi nilai daya berdasarkan kemiripan kondisi intensitas cahaya untuk menggantikan pembacaan sensor daya yang tidak valid.
b. Penggunaan Seluruh Data, yaitu seluruh data hasil pra-pemrosesan digunakan bersama dalam proses pemodelan sebagai persoalan regresi antara parameter cuaca dan daya keluaran pada tiap titik waktu. Pendekatan ini dipilih karena estimasi daya bersifat sesaat, yaitu memprediksi daya berdasarkan parameter cuaca pada waktu yang sama.
c. Normalisasi Min-Max, yaitu proses penyeragaman rentang nilai seluruh variabel ke dalam skala [0, 1]. Parameter normalisasi (nilai minimum dan maksimum) dihitung dari seluruh data yang digunakan dalam pemodelan, kemudian diterapkan pada variabel yang bersesuaian agar seluruh variabel berada pada skala yang sebanding.
d. Pembangunan Model Regresi, yaitu mengimplementasikan dan melatih dua algoritma *Ensemble Learning* berbasis pohon keputusan secara paralel. *Random Forest* membangun sejumlah pohon keputusan secara independen melalui

mekanisme *Bagging*, sedangkan *XGBoost* membangun pohon secara berurutan dengan setiap pohon baru mengoreksi kesalahan pohon sebelumnya melalui mekanisme *Boosting*. Nilai *hyperparameter* pada masing-masing model ditetapkan sebagai konfigurasi tetap.

e. Analisis *Feature Importance*, yaitu menghitung peringkat kontribusi masing-masing parameter cuaca dalam menurunkan nilai kesalahan (*error*) pada titik percabangan pohon keputusan, sehingga dapat diidentifikasi parameter cuaca yang paling signifikan mempengaruhi estimasi daya keluaran PLTS Apung.

3. Objective

   Tujuan utama dari kerangka pemikiran ini mencakup dua hal fundamental. Pertama, mengomparasikan kinerja komputasi antara model *Random Forest* dan *XGBoost* secara *head-to-head* guna menentukan arsitektur mana yang paling presisi dalam mengestimasi daya keluaran PLTS Apung berdasarkan parameter cuaca. Kedua, melakukan analisis *Feature importance* untuk memberikan justifikasi objektif mengenai parameter cuaca mana di antara suhu, kelembapan, dan intensitas cahaya yang paling dominan mempengaruhi fluktuasi daya keluaran.

4. Measurement

   Untuk mengukur performa dan memvalidasi tujuan penelitian, kinerja model dievaluasi melalui tiga metrik regresi sebagai berikut:

a. *Root Mean Square Error* (RMSE), yaitu mengukur akar dari rata-rata kuadrat kesalahan prediksi dengan memberikan penalti kuadratik terhadap penyimpangan yang besar, sehingga sangat sensitif terhadap *outlier* dan anomali cuaca.
b. *Mean Absolute Error* (MAE), yaitu mengukur rata-rata penyimpangan absolut antara nilai estimasi model dengan nilai daya aktual, memberikan gambaran yang intuitif mengenai besaran rata-rata kesalahan prediksi.
c. Koefisien Determinasi (R^2), yaitu mengukur persentase variabilitas daya aktual yang dapat dijelaskan oleh model fitur cuaca. Nilai R^2 yang mendekati 1 mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik.