

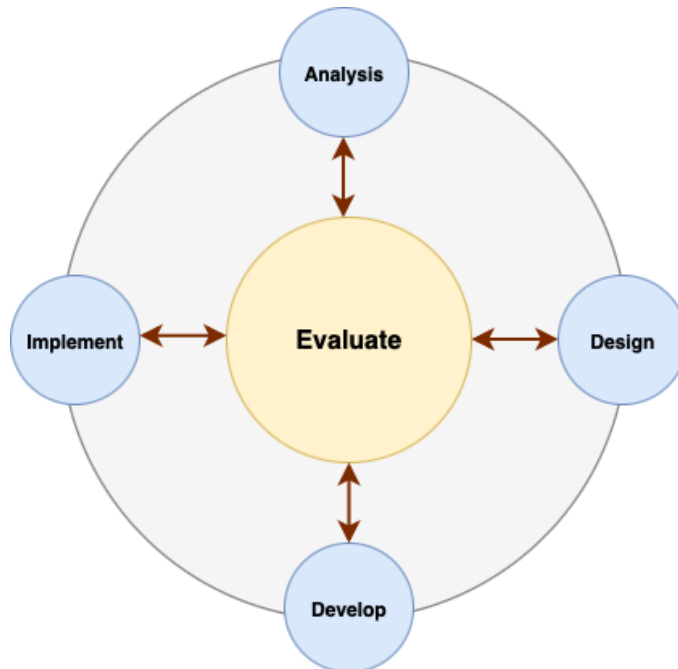
BAB III

METODE PENELITIAN

Pada bab ini akan menjelaskan secara terperinci mengenai metodologi yang digunakan dalam penelitian. Pembahasan akan dimulai dari jenis penelitian, alur, dan deskripsi umum, lalu dilanjutkan dengan perancangan sistem yang mencakup perangkat keras dan perangkat lunak. Terakhir, bab ini akan menguraikan secara lengkap setiap tahapan dalam perancangan sistem prediksi *downtime*, mulai dari persiapan data hingga evaluasi dan implementasi model.

3.1 Jenis Penelitian

Penelitian ini merupakan jenis penelitian *Research and Development* (R&D) yang bertujuan untuk merancang dan membangun sistem monitoring perangkat jaringan berbasis SNMP, dengan integrasi analisis prediskis *downtime* sebagai analisis tambahan menggunakan pendekatan ML. Dalam penelitian ini, algoritma *Logistic Regression* dan *Random Forest* digunakan untuk melakukan prediksi *downtime* berdasarkan data historis yang dikumpulkan oleh sistem.

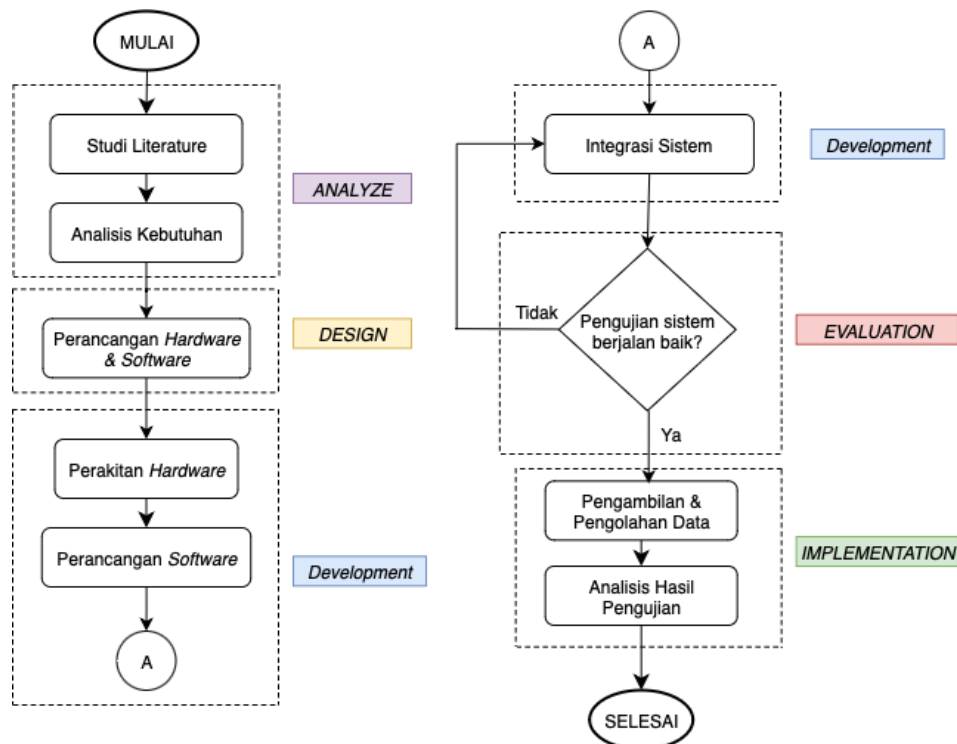


Gambar 3. 1 Desain Penelitian Menggunakan ADDIE

Penelitian ini menggunakan pendekatan *Analysis, Design, Development, Implementation, Evaluation* (ADDIE). Pendekatan ini dipilih karena menyediakan alur kerja yang sistematis dan terstruktur, mulai dari identifikasi masalah, perancangan sistem, pengembangan solusi, hingga pengujian produk yang akan dikembangkan. ADDIE tidak hanya berfokus pada penciptaan produk, tetapi juga memastikan bahwa produk yang dihasilkan efektif, efisien dan sesuai dengan kebutuhan pengguna. Model ini umum digunakan dalam pengembangan teknologi, khususnya untuk menghasilkan produk inovatif dan fungsional (Waruwu, 2024). Pendekatan ADDIE digunakan dalam penelitian ini, sebagaimana ditampilkan pada Gambar 3.1.

3.2 Alur Penelitian

Alur penelitian ini disusun mengikuti pengembangan sistem berbasis model ADDIE. Berikut merupakan diagram alur tahapan penelitian yang mengikuti model ADDIE.



Gambar 3. 2 Alur Penelitian Dengan ADDIE

1. *Analysis*

Pada tahap *analysis* bertujuan untuk mengidentifikasi permasalahan utama, yaitu belum adanya sistem monitoring perangkat jaringan yang efisien dan mampu melakukan prediksi *downtime* secara akurat, khususnya pada perangkat jaringan yang menggunakan sumber daya energi alternatif seperti PV. Analisis mencakup identifikasi kebutuhan perangkat keras (*hardware*), seperti jenis mikrokontroler dan sensor yang sesuai, kebutuhan perangkat lunak (*software*), seperti protokol komunikasi (SNMP) dan platform visualisasi (PRTG), serta kebutuhan model ML untuk memprediksi *downtime* berdasarkan pola historis data sensor lingkungan dan kelistrikan.

2. *Design*

Pada tahap *design* dilakukan perancangan sistem secara menyeluruh, baik dari sisi *hardware* dan *software*. Perancangan *hardware* meliputi pemilihan komponen utama seperti DHT22 (suhu dan kelembaban), HSTS016L (arus), sensor tegangan, BH1750 (intensitas cahaya), mikrokontroler Arduino Nano V3, Mini-PC (Orange Pi Zero 3), serta jalur komunikasi RS485. Dari sisi *software*, dirancang komunikasi antar perangkat melalui RS485, integrasi dengan protokol SNMP, dan visualisasi data menggunakan PRTG *Network Monitor*. Selain itu, pada tahap ini juga dirancang alur data dan skema pemodelan ML untuk memprediksi *downtime* berdasarkan parameter lingkungan dan kelistrikan yang terkumpul.

3. *Development*

Pada tahap *development* mencakup proses realisasi rancangan menjadi sistem yang dapat diuji. Tahap ini terbagi menjadi tiga bagian, yaitu perakitan *hardware*, perancangan *software*, dan integrasi sistem. Pada tahap perakitan *hardware*, dilakukan proses penyusunan dan pengkabelan komponen seperti sensor, mikrokontroler, dan jalur komunikasi hingga seluruh perangkat dapat berfungsi sesuai rancangan. Pada tahap perancangan *software*, dilakukan pengembangan kode pada mikrokontroler untuk pembacaan data sensor, pengiriman data melalui RS285, serta implementasi SNMP di Mini-PC yang sudah diatur sedemikianrupa untuk diintegrasikan dengan PRTG *Network Monitor*. Pada tahap integrasi sistem, dilakukan penggabungan perangkat

Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

hardware dan *software* menjadi satu kesatuan yang saling terhubung. Pada tahap terakhir ini dilakukan pengujian awal untuk memastikan data sensor dapat ditransmisikan dan divisualisasikan dengan benar. Jika pengujian integrasi sistem berjalan dengan baik, proses dilanjutkan ke tahap *Implementation*. Namun, jika ditemukan kendala, perbaikan dilakukan kembali pada tahap *Development*, khususnya pada proses integrasi sistem.

4. *Implementation*

Pada tahap *implementation*, sistem yang telah dikembangkan perlu dilakukan pengujian langsung dalam lingkungan operasional yang sesungguhnya. Data sensor yang dikirimkan oleh mikrokontroler melalui komunikasi RS485 diteruskan ke Mini-PC untuk dikirimkan oleh SNMP, kemudian diolah dan ditampilkan pada PRTG *Network Monitor*. Model ML kemudian dijalankan menggunakan data historis untuk memprediksi kemungkinan *downtime* dan sistem diamati kestabilannya dalam merespon perubahan merespons perubahan kondisi lingkungan secara *real-time*.

5. *Evaluation*

Pada tahap *evaluation* digunakan untuk mengukur kinerja sistem monitoring dan Tingkat akurasi model prediksi *downtime*. Evaluasi meliputi keakuratan data sensor, kestabilan komunikasi SNMP, ketahanan sistem terhadap gangguan, serta akurasi prediksi model ML. Hasil evaluasi ini menjadi dasar untuk melakukan pengembangan lebih lanjut agar sistem menjadi lebih andal, presisi, dan berkelanjutan.

3.3 Deskripsi Umum Penelitian

Penelitian ini bertujuan untuk merancang dan membangun sistem monitoring perangkat jaringan berbasis protokol SNMP yang dilengkapi dengan fitur prediksi *downtime* menggunakan algoritma ML. Sistem ini memanfaatkan sumber energi dari PV dan ditujukan untuk menjaga kestabilan konektivitas perangkat jaringan seperti modem, khususnya di wilayah yang tidak terjangkau oleh listrik PLN secara merata seperti wilayah 3T.

Perangkat monitoring yang dikembangkan terdiri dari sensor-sensor lingkungan dan kelistrikan meliputi suhu, kelembaban, intensitas cahaya, tegangan, dan arus. Data dari sensor dikirimkan ke mikrokontroler Arduino Nano V3, lalu diteruskan ke Orange Pi Zero 3 (Mini-PC) melalui jalur komunikasi RS485. Selanjutnya data diproses dan dikirimkan ke server pemantauan menggunakan protokol SNMP untuk divisualisasikan melalui PRTG Network Monitor.

Selain fungsi monitoring, sistem ini dilengkapi dengan kemampuan prediktif menggunakan dua algoritma ML, yaitu *Logistic Regression* sebagai baseline model yang mudah diinterpretasi, dan *Random Forest* untuk menangani hubungan non-linier antar variabel serta meningkatkan akurasi prediksi. Hasil prediksi ini diharapkan dapat membantu pengambilan keputusan preventif sebelum terjadi gangguan, sehingga *downtime* dapat diminimalkan dan kontinuitas layanan jaringan berbasis energi terbarukan dapat terjaga.

3.4 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di PT Pasifik Satelit Nusantara, dengan ruang lingkup yang telah dikondisikan dalam lingkungan laboratorium atau ruangan internal perusahaan. Penelitian tidak dilakukan di lapangan secara langsung, namun dilaksanakan dalam bentuk simulasi dan integrasi sistem monitoring berbasis SNMP yang telah dirancang.

Adapun waktu pelaksanaan penelitian ini berlangsung dalam dua periode. Tahap perancangan dan pengembangan sistem dilakukan pada 1 Januari hingga 14 Februari 2025, sedangkan pengumpulan data aktual untuk proses monitoring dan prediksi *downtime* dilakukan pada 1 April hingga 21 Mei 2025, sesuai dengan ketersediaan dan akses data yang relevan.

3.5 Perancangan Sistem

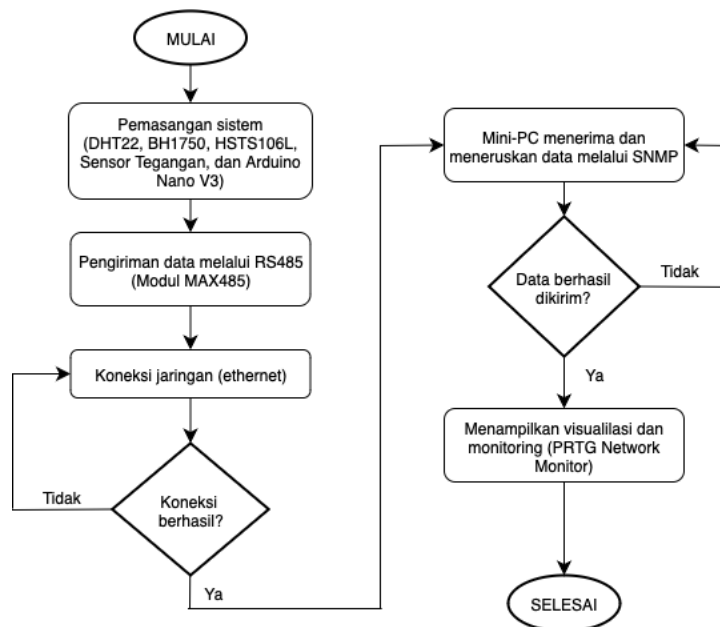
Perancangan sistem dilakukan untuk membangun perangkat monitoring lingkungan dan kelistrikan menggunakan mikrokontroler Arduino Nano V3 serta Mini-

PC Orange Pi Zero 3 yang berperan sebagai *SNMP Agent*. Sistem ini terintegrasi dengan PRTG Network Monitor dan dilengkapi dengan algoritma ML untuk prediksi downtime. Perancangan dibagi menjadi tiga bagian utama, yaitu perancangan perangkat keras (*hardware*), perancangan perangkat lunak (*software*), dan perancangan sistem prediksi *downtime*.

3.5.1 Perancangan Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) pada sistem ini terdiri dari beberapa sensor utama yang terhubung ke mikrokontroler Arduino Nano V3 sebagai pusat pengolahan data sebelum dikirimkan ke Mini-PC. Sensor yang digunakan meliputi sensor suhu dan kelembaban DHT22, sensor intensitas cahaya BH1750, sensor arus HSTS106L, dan sensor tegangan dari MH Electronic.

Data yang diperoleh dari sensor diproses oleh Arduino Nano V3 dan dikirimkan ke Mini-PC menggunakan protokol RS485 melalui modul converter MAX485. RS485 dipilih karena mampu mengirimkan data pada jarak jauh dengan kecepatan stabil serta tahan terhadap gangguan elektromagnetik.

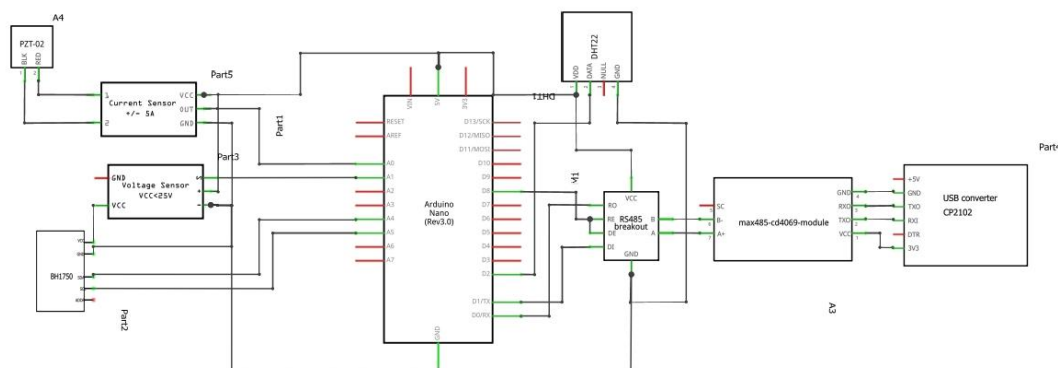


Gambar 3. 3 Alur kerja sistem *hardware*

Sebelum data diterima oleh Mini-PC, sistem akan memastikan koneksi jaringan ethernet. Jika koneksi belum tersedia, sistem akan melakukan percobaan ulang hingga koneksi berhasil. Setelah koneksi internet terjamin, Mini-PC menerima data dari Arduino Nano V3, menjalankan fungsi sebagai *SNMP Agent*, dan meneruskan data tersebut ke server pemantauan (*dashboard* monitoring PRTG). Pada tahap ini, sistem melakukan pengecekan apakah data berhasil dikirim ke server. Jika pengiriman data berhasil, informasi dari sensor akan divisualisasikan secara *real-time* melalui PRTG *Network Monitor* sehingga pengguna dapat memantau kondisi suhu, kelembaban, intensitas cahaya, arus, dan tegangan secara langsung. Namun, jika data tidak berhasil terkirim, proses akan kembali ke tahap penerimaan data di Mini-PC untuk dilakukan pengiriman ulang melalui SNMP.

3.5.1.1 Skema Rangkaian

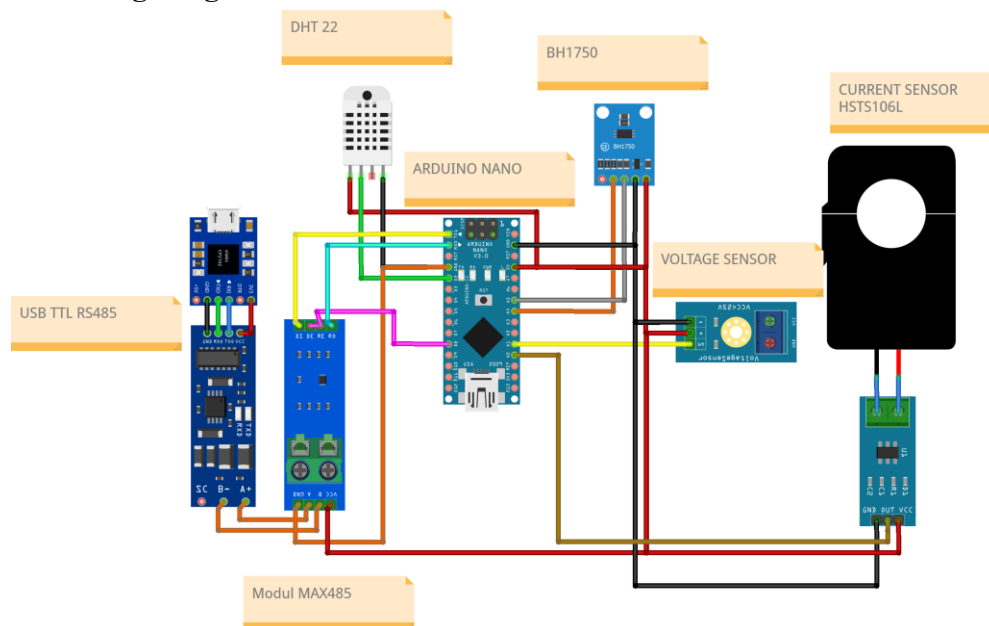
Skema keseluruhan sistem monitoring yang dibangun seperti pada Gambar 3.3 digunakan untuk memvisualisasikan bagaimana setiap sensor terhubung dan terkoneksi antar komponen lainnya. Seluruh sensor, yaitu DHT22, BH1750, HSTS106L, dan sensor tegangan terhubung ke mikrokontroler Arduino Nano V3 sebagai pusat akuisisi data. Sensor DHT22 berfungsi untuk mengukur suhu dan kelembaban, BH1750 digunakan untuk mengukur intensitas cahaya, HSTS106L untuk mengukur arus listrik, dan sensor tegangan untuk mengukur tegangan output dari PV.



Gambar 3. 4 Perancangan Schematic diagram

Data dari setiap sensor diteruskan ke Arduino Nano V3, yang kemudian mengirimkan data tersebut ke Mini-PC melalui jalur komunikasi RS485 menggunakan modul konverter MAX485. Pada *schematic*, koneksi RS485 digambarkan sebagai jalur komunikasi dua kabel (*differential pair*) yang terhubung ke terminal konverter, memberikan ketahanan terhadap gangguan elektromagnetik dan memungkinkan transmisi jarak jauh. Modul RS485 dihubungkan ke modul konverter USB dengan konfigurasi silang, kemudian modul USB tersebut dikoneksikan ke Mini-PC. Mini-PC berperan sebagai *SNMP Agent* yang meneruskan data sensor ke server pemantauan dan divisualisasikan menggunakan PRTG Network Monitor.

3.5.1.2 Wiring Diagram dan Interkoneksi Pin



Gambar 3. 5 Perancangan *Wiring* diagram

Wiring diagram pada Gambar 3.5 digunakan sebagai panduan utama dalam proses perakitan perangkat keras (*hardware*) sistem monitoring. Perancangan koneksi ini dilakukan dengan mengacu pada datasheet setiap sensor dan modul yang digunakan, sehingga memastikan bahwa hubungan antar komponen telah sesuai dengan spesifikasi teknis dan mampu berfungsi secara optimal. Jalur komunikasi dan suplai daya dirancang berdasarkan karakteristik masing-masing perangkat, termasuk kebutuhan

tegangan, jenis sinyal, dan protokol komunikasi yang digunakan. Untuk memberikan gambaran yang lebih jelas mengenai konfigurasi sistem, detail interkoneksi antara Arduino Nano V3, sensor-sensor lingkungan dan kelistrikan, modul RS485, serta Mini-PC disajikan pada Tabel 3.1.

Tabel 3. 1 Konfigurasi Pin

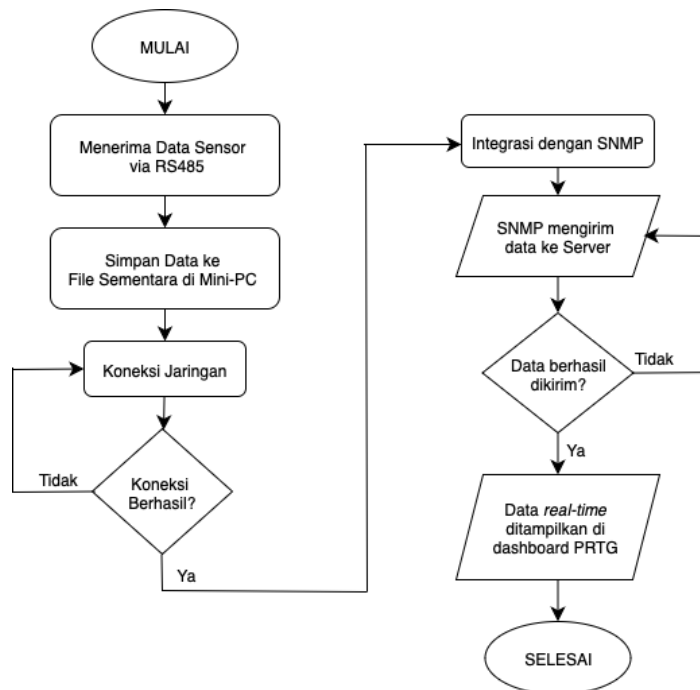
Komponen	Pin Modul	Pin Arduino Nano V3	Keterangan
DHT22 (Suhu & Kelembaban)	VCC	5V	Catu daya sensor DHT22
	OUT	D2	Pin digital untuk membaca data suhu dan kelembaban
	GND	GND	Ground sensor
BH1750 (Intensitas Cahaya)	VCC	5V	Catu daya sensor BH1750
	GND	GND	Ground sensor
	SCL	A5	Serial Clock Line (I2C)
	SDA	A4	Serial Data Line (I2C)
HSTS106L (Sensor Arus)	VCC	5V	Catu daya sensor HSTS106L
	GND	GND	<i>Ground</i> sensor
	OUT	A0	Output analog tegangan yang mewakili arus
MH Electronic (Sensor Tegangan)	S	A1	Output analog tegangan terukur
	VCC (+)	5V	Catu daya sensor
	GND (-)	GND	<i>Ground</i> sensor
	VCC <i>Output Object</i>	5V	Tegangan yang ingin diukur (input dari sistem)
	GND <i>Output Object</i>	GND	<i>Ground</i> sistem
RS485 (Modul MAX485)	VCC	5V	Catu daya modul komunikasi RS485
	GND	GND	<i>Ground</i> modul
	A, B	B, A (USB to TTL RS485)	Komunikasi <i>differential</i> RS485
	RO	RX (D0)	<i>Receive Output</i> ke Arduino
	DI	TX (D1)	Data Input dari Arduino
	DE/RE	D8	Kontrol arah komunikasi (Enable Transmit)

USB to TTL RS485	A, B	B, A (dari MAX485)	Disambungkan ke Mini-PC (Orange Pi) melalui koneksi USB
	USB	Port USB Mini-PC	Menghubungkan Arduino dengan Mini-PC via serial RS485

3.5.2 Perancangan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) dalam penelitian ini dirancang untuk mengatur alur komunikasi data antara mikrokontroler dan sistem pemantauan jaringan menggunakan protokol SNMP. Proses diawali dengan pembacaan data dari berbagai sensor, seperti suhu dan kelembaban (DHT22), intensitas cahaya (BH1750), tegangan (MH Electronic), dan arus (HSTS016L). Data tersebut dikirim dari mikrokontroler Arduino Nano V3 menuju Mini-PC melalui jalur komunikasi RS485.

Pada Mini-PC, skrip berbasis Python dijalankan untuk membaca data yang masuk melalui port USB RS485, memisahkan nilai setiap sensor, lalu menyimpannya ke file sementara (*temporary file*). Data yang telah tersimpan kemudian dipetakan ke OID tertentu melalui konfigurasi SNMP Daemon, sehingga dapat diakses oleh sistem pemantauan dan dihubungkan ke server.



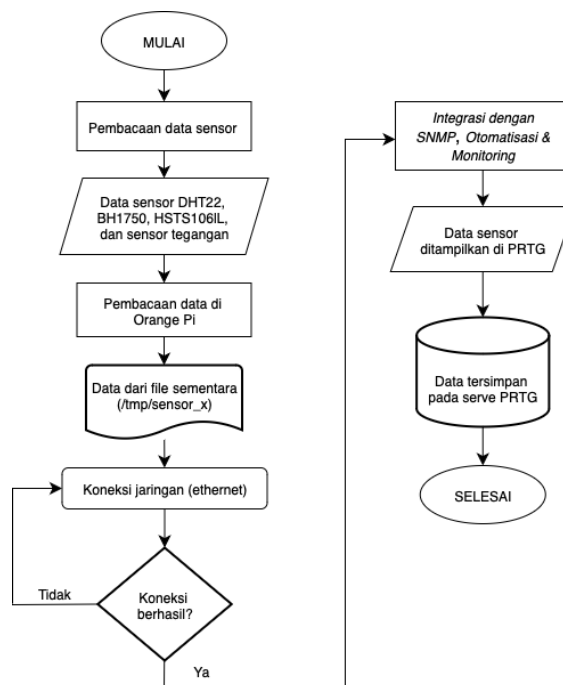
Gambar 3. 6 Alur kerja sistem *software*

Sebelum proses pengiriman data, perangkat Mini-PC melakukan pengecekan konektivitas jaringan ethernet. Jika jaringan terhubung, data sensor dikirimkan ke server PRTG *Network Monitor* melalui SNMP. Apabila data gagal terkirim, proses akan mengulang ke pengecekan konektivitas jaringan. Setelah itu dilakukan pengecekan keberhasilan pengiriman kembali. Jika data berhasil terkirim, data sensor ditampilkan secara *real-time* pada dashboard PRTG. Jika data gagal terkirim, proses akan kembali ke tahap pengiriman data oleh SNMP ke server hingga berhasil.

Perancangan perangkat lunak ini memastikan seluruh data sensor dapat dipantau secara *real-time*, tersimpan sementara untuk mencegah kehilangan data, serta dikirim secara konsisten melalui protokol SNMP untuk mendukung analisis dan prediksi *downtime* pada tahap selanjutnya.

3.5.2.1 Konfigurasi SNMP

Konfigurasi SNMP dilakukan menjadi empat tahapan, yaitu pembacaan data sensor, pembacaan data di orange pi, integrasi dengan SNMP, serta otomatisasi dan monitoring.



Gambar 3. 7 Alur diagram perancangan konfigurasi SNMP

Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

1. Pembacaan Data Sensor

Data sensor dari mikrokontroler Arduino Nano V3 dihubungkan ke Orange Pi Zero 3 (Mini-PC) melalui jalur komunikasi fisik RS485. Pada penelitian ini, digunakan dua modul RS485, yaitu modul MAX485 dan konverter USB to RS485. Modul MAX485 terhubung langsung dengan pin pada Arduino Nano V3, sedangkan modul konverter USB to RS485 dihubungkan dengan port USB pada Mini-PC. Kedua modul ini saling dihubungkan secara silang, seperti pin A pada modul MAX485 dihubungkan dengan pin B pada modul konverter USB to RS485, dan sebaliknya.

Beberapa data sensor yang dikirim ke orangepi berasal dari sensor DHT22 (suhu dan kelembapan), BH1750 (intensitas cahaya), MH Electronic (tegangan), dan HSTS016L (arus).

Tabel 3. 2 Format Data Serial yang Dikirim

26,81 67,44 4198,25 1,48 0,13

2. Pembacaan Data di Orange Pi Zero 3 (Mini-PC)

Mini-PC menjalankan skrip Python untuk membaca data yang masuk port USB RS485 setiap waktu. Melalui SNMP data yang dikirim dari Mini-PC diterima dan ditampilkan dalam bentuk grafik di dashboard PRTG. Data ini dipisah dan disimpan ke file sementara (*temporary file*).

Tabel 3. 3 Skrip Pembacaan Data di Mini-PC

```
def write_sensor_data(index, value):
    with open(f'/tmp/sensor_{index}', 'w') as f:
        f.write(str(value))

while True:
    if ser.in_waiting > 0:
        line = ser.readline().decode('utf-8').strip()
        data = line.split(",")
        for i, value in enumerate(data, 1):
            write_sensor_data(i, value)
```

3. Integrasi dengan SNMP

Tahap awal integrasi dengan SNMP adalah menghubungkan Mini-PC dengan jaringan melalui Ethernet. Mini-PC harus terkoneksi dengan jaringan agar dapat diketahui alamat IPnya. Selain itu, perlu dilakukan instalasi beberapa paket seperti `snmpd`, `snmp`, dan `libsnmp-dev`. Seluruh proses pengaturan pada Mini-PC dilakukan melalui terminal dengan perintah standar pada sistem operasi Linux.

Setelah seluruh paket yang diperlukan terinstall, dilakukan konfigurasi SNMP agar dapat membaca file sensor melalui OID tertentu. File konfigurasi SNMP (`snmpd.conf`) perlu disesuaikan dengan perintah *pass*, yang menunjuk ke skrip *bash* untuk setiap sensor.

Tabel 3. 4 Skrip bash untuk salah satu sensor

```
#!/bin/bash
if [ "$1" = "-g" ]; then
    echo .1.3.6.1.3.2010.5.0.5.5.1
    echo gauge
    cat /tmp/sensor_1 2>/dev/null || echo 0
fi
exit 0
```

4. Otomatisasi dan Monitoring

Setiap skrip memiliki OID yang berbeda sesuai dengan jenis sensor. Untuk menjaga agar skrip pembacaan sensor tetap berjalan, digunakan *cronjob* yang menjalankan pengecekan setiap menit. Hal ini bertujuan agar data sensor terus tersedia dan dapat dimonitor secara konsisten oleh SNMP agent.

Tabel 3. 5 Penggunaan Cronjob pada Mini-PC

```
* * * * * /var/scripts/run_sensor.sh
```

Monitoring dapat dilakukan langsung melalui terminal Mini-PC. Dengan mengetahui versi SNMP yang digunakan, *community string* SNMP, alamat IP yang dituju serta OID untuk mengetahui data spesifik yang ingin diambil.

Tabel 3. 6 Monitoring SNMP secara langsung pada terminal Mini-PC

```
# Get Data from SNMP
snmpget -v 2c -c public localhost .1.3.6.1.3.2010.5.0.5.5.1
```

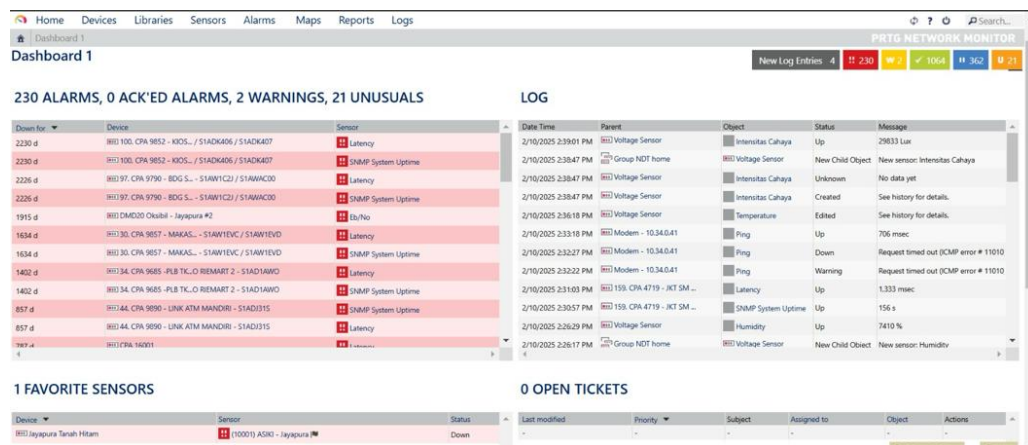
```
snmpget -v 2c -c public 192.168.1.159
.1.3.6.1.3.2010.5.0.5.5.1
```

3.5.2.2 Dashboard PRTG Network Monitor

PRTG *Network Monitor* berfungsi sebagai antarmuka visual yang menampilkan data sensor yang telah berhasil dikirimkan oleh SNMP pada Mini-PC. Setelah data dari sensor melewati proses pembacaan, penyimpanan sementara, pemetaan OID, dan pengiriman melalui SNMP, PRTG membaca OID yang telah dikonfigurasi dan menampilkannya dalam bentuk grafik secara *real-time*.

Setiap parameter sensor, yaitu suhu, kelembaban, intensitas cahaya, tegangan, dan arus, diatur sebagai *channel* terpisah di PRTG. Karena SNMP menyediakan data dalam bentuk satuan per OID, visualisasi dilakukan secara individual untuk tiap sensor. Jika diperlukan analisis kolektif, penggabungan data dilakukan di sisi server sebelum diolah lebih lanjut.

PRTG juga menyediakan fitur notifikasi dan alarm otomatis ketika terjadi terjadi nilai ekstrem atau gangguan lainnya. Dalam penelitian ini, dashboard digunakan hanya sebagai pengguna (*user*) tanpa modifikasi lanjutan, cukup dengan menambahkan sensor berbasis OID melalui konfigurasi yang telah disiapkan. Dengan adanya dashboard ini, kondisi lingkungan dan kelistrikan perangkat jaringan dapat dipantau secara visual, membantu mendeteksi potensi masalah lebih dini, dan mendukung sistem prediksi *downtime* yang dirancang.



Gambar 3. 8 Notifikasi pada dashboard PRTG *Network Monitor*

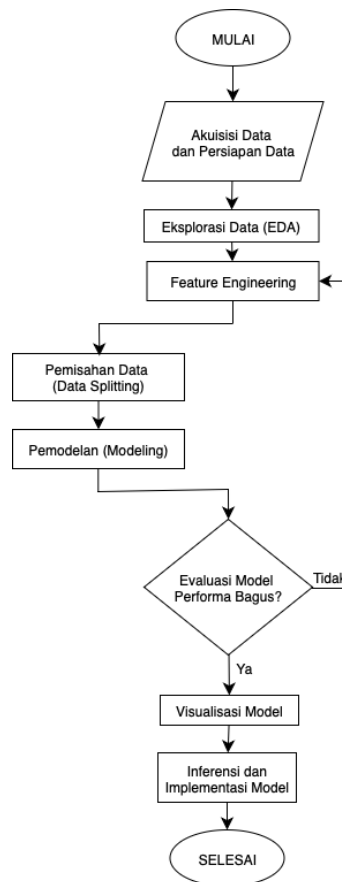
Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

3.5.3 Perancangan Sistem Prediksi *Downtime*

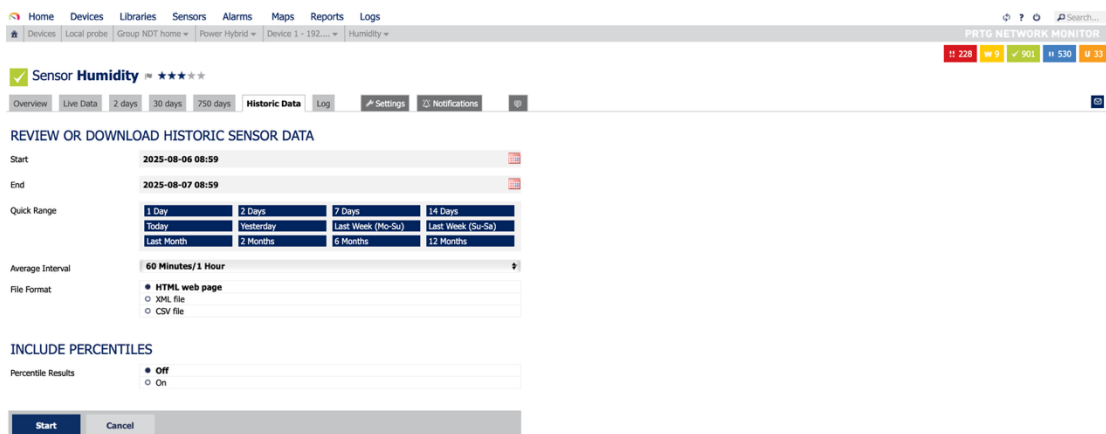
Sistem prediksi *downtime* pada penelitian ini dirancang untuk menganalisis pola dari data sensor lingkungan dan kelistrikan guna memperkirakan potensi gangguan yang dapat menyebabkan perangkat tidak berfungsi (*downtime*). Tujuannya adalah memberikan peringatan dini sehingga tindakan preventif dapat dilakukan sebelum gangguan terjadi. Proses prediksi dilakukan menggunakan algoritma ML dengan tahapan yang sistematis, dimulai dari akuisisi data, persiapan data, pelatihan model, hingga evaluasi performa model. Model ML yang digunakan dalam penelitian ini adalah *Logistic Regression* dan *Random Forest*, karena keduanya mampu memproses data multivariant dan memiliki kelebihan masing-masing dalam akurasi dan interpretabilitas.



Gambar 3. 9 Alur metodologi sistem prediksi *downtime*

3.5.3.1 Akuisisi Data dan Persiapan Data

Data yang digunakan pada penelitian ini diperoleh dari sensor-sensor lingkungan dan kelistrikan yang terdiri atas suhu ($^{\circ}\text{C}$), kelembaban (%), intensitas cahaya (lux), tegangan (Volt), dan arus (Ampere). Seluruh sensor telah diintegrasikan dengan SNMP dan dimonitor melalui PRTG *Network Monitor*. Periode pengambilan data dilakukan pada 1 April hingga 21 Mei 2025. Proses pengambilan data dari PRTG dilakukan dengan memilih sensor yang akan diakses dengan mengunduh data, menentukan rentang waktu pengambilan data (baik secara manual memilih rentang tanggal atau menggunakan preset bawaan PRTG), menetapkan interval waktu pengambilan data, serta memilih format keluaran dalam format file seperti apa. Pada proses ini, format keluaran yang dipilih dalam bentuk CSV. Dataset hasil unduhan kemudian diproses lebih lanjut.



Gambar 3. 10 Pengumpulan Data pada PRTG

Tahap persiapan data meliputi beberapa langkah penting. Pertama, seluruh nilai dari kolom numerik yang masih menggunakan tanda koma (,) sebagai pemisah desimal diubah menjadi titik (.) agar dapat diproses sebagai tipe data float. Kolom yang mengalami konversi ini meliputi suhu, kelembaban, cahaya, tegangan, dan arus. Kedua, kolom Timestamp dibersihkan dari format teks tambahan dengan hanya mengambil bagian tanggal dan waktu utama, kemudian dikonversi menjadi tipe data datetime dengan format standar YYYY-MM-DD HH:MM:SS sehingga dapat digunakan dalam analisis berbasis waktu. Setelah itu, dilakukan pengecekan ulang tipe

Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

data setiap kolom untuk memastikan kesesuaian dengan kebutuhan analisis. Dataset yang telah bersih dan terkonversi ini kemudian disimpan sebagai data siap pakai untuk tahap analisis selanjutnya.

3.5.3.2 Eksplorasi Data (EDA)

Tahap EDA dilakukan untuk memahami karakteristik dan pola dari dataset sebelum dilakukan pemodelan. Tahapan ini mencakup analisis statistik awal untuk memahami sebaran data, hubungan antar variabel, dan deteksi anomali. Langkah pertama adalah melakukan pemeriksaan dimensi dataset, jumlah kolom, dan jumlah baris yang tersedia, untuk memastikan kelengkapan data hasil akuisisi. Selanjutnya dilakukan pengecekan tipe data setiap kolom agar sesuai dengan karakteristik data yang diharapkan, serta adanya *missing values* atau anomali. Pada tahap ini menunjukkan bahwa dataset memiliki 1225 baris dengan 8 kolom, tidak terdapat data hilang (*missing values*) dan data duplikat (*Duplicated data*).

Analisis statistik deskriptif diterapkan pada kolom numerik untuk mendapatkan gambaran umum seperti nilai minimum, maksimum, rata-rata, dan standar deviasi dari setiap variabel sensor, meliputi suhu, kelembaban, tegangan, arus, dan intensitas cahaya. Visualisasi data kemudian digunakan untuk memahami distribusi nilai dan hubungan antar variabel. Histogram digunakan untuk melihat distribusi data tiap sensor, sedangkan line plot membantu mengidentifikasi tren atau pola fluktuasi nilai sensor terhadap waktu.

Selain itu, dilakukan analisis korelasi antar variabel menggunakan *heatmap* untuk mengetahui hubungan linier antara parameter-parameter sensor. Hasil dari analisis ini membantu dalam menentukan fitur mana saja yang relevan dan memiliki kontribusi potensial terhadap terjadinya *downtime*, sekaligus menjadi dasar bagi pembuatan fitur tambahan pada tahap *feature engineering*.

3.5.3.3 Pemisahan Data (*Data Splitting*)

Tahap *data splitting* dilakukan sebelum proses *feature engineering* untuk memisahkan dataset mentah menjadi dua subset secara temporal untuk menjaga urutan

kronologis data. Metode ini dipilih karena data yang digunakan merupakan data deret waktu (*time-series*) dari pembacaan sensor, sehingga penting untuk memastikan bahwa data uji merepresentasikan kondisi di masa mendatang yang belum pernah dilihat oleh model saat pelatihan.

Proporsi pembagian data adalah 80% untuk pelatihan (*training set*) dan 20% untuk pengujian (*testing set*). Proses pembagian dilakukan dengan menentukan titik pemisah (*split point*) berdasarkan indeks baris data, di mana bagian awal dataset digunakan sebagai *training set* dan bagian akhir digunakan sebagai *testing set*. Hasil dari proses ini adalah *train_data* yang digunakan untuk melatih model, dan *test_data* yang digunakan untuk mengukur performa model pada data baru. Metode ini memastikan bahwa model hanya dilatih menggunakan data dari periode awal dan dievaluasi pada data dari periode setelahnya, sehingga dapat menguji kemampuan model dalam memprediksi kondisi yang benar-benar baru. Setelah pemisahan ini, masing-masing subset data akan diproses pada tahap *feature engineering* secara terpisah untuk menambahkan fitur turunan sebelum memasuki proses pelatihan dan evaluasi model.

3.5.3.4 Feature Engineering

Tahap *feature engineering* dilakukan untuk memperluas dan memperkaya informasi pada dataset sehingga model dapat mengenali pola yang lebih kompleks terkait potensi terjadinya *downtime*. Proses ini dilakukan dengan menambahkan beberapa jenis fitur turunan yang berasal dari data asli pembacaan sensor.

Pertama, dibuat fitur berbasis waktu (*time features*) yang diekstrak dari kolom *timestamp*, meliputi *hour* (jam pengambilan data), *day_of_week* (hari dalam minggu), *is_weekend* (indikator akhir pekan), dan *is_daytime* (indikator waktu siang). Fitur ini membantu model dalam mengenali perbedaan pola sensor pada waktu-waktu tertentu, misalnya antara siang dan malam atau hari kerja dan akhir pekan.

Kedua, ditambahkan fitur *lag* dan *change* untuk masing-masing parameter sensor suhu, kelembaban, cahaya, arus, dan tegangan. Fitur *lag* merepresentasikan nilai sensor pada beberapa waktu sebelumnya (misalnya 1 langkah dan 3 langkah ke

Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

belakang), sedangkan fitur *change* merepresentasikan selisih nilai sensor terhadap pembacaan sebelumnya. Fitur-fitur ini berguna untuk menangkap dinamika perubahan nilai sensor yang dapat menjadi indikasi awal terjadinya downtime.

Nilai yang hilang (*missing*) yang muncul akibat proses *shift* pada pembuatan *lag features* kemudian diisi menggunakan metode *forward fill* (*ffill*) untuk mempertahankan kontinuitas data, dan apabila masih terdapat nilai kosong, akan diganti dengan nol. Hasil akhirnya adalah kumpulan fitur yang terdiri dari *base features* (data sensor asli), *time features*, dan *lag/change features*, yang seluruhnya bersifat numerik dan siap digunakan pada tahap data splitting untuk pelatihan model.

3.5.3.5 Data Scaling

Tahap *data scaling* dilakukan untuk menyeragamkan skala nilai antar fitur sehingga model ML dapat memproses setiap fitur dengan bobot yang seimbang. Tahap ini dilakukan karena perbedaan rentang nilai antar fitur dapat menyebabkan model lebih condong pada fitur dengan skala lebih besar.

Proses penskalaan dilakukan setelah *data splitting* dan *feature engineering*. Pertama, data dipisahkan menjadi variabel masukan (X) yang berisi seluruh kolom fitur terpilih, dan variabel keluaran (y) yang berisi label target *downtime* (0 atau 1). Selanjutnya, digunakan metode *StandardScaler* dari pustaka *scikit-learn* yang melakukan transformasi dengan mengurangkan rata-rata (*mean*) dan membaginya dengan standar deviasi (*standard deviation*) untuk setiap fitur.

Penerapan *scaling* hanya dilakukan dengan fit pada data latih (*training set*) untuk menghitung nilai rata-rata dan standar deviasi, kemudian nilai tersebut digunakan pada proses transform untuk data latih dan data uji (*testing set*). Pendekatan ini memastikan bahwa informasi dari data uji tidak bocor ke proses pelatihan (*data leakage*). Hasil dari tahap ini adalah *X_train_scaled* dan *X_test_scaled*, yaitu data latih dan uji yang telah berada pada skala yang seragam, siap digunakan untuk pelatihan dan evaluasi model.

3.5.3.6 Pemodelan (*Modeling*)

Tahap *modeling* dilakukan untuk membangun model ML yang dapat memprediksi potensi *downtime* berdasarkan data sensor yang telah melalui tahap pemrosesan. Pada penelitian ini digunakan dua algoritma, yaitu *Logistic Regression* dan *Random Forest*. *Logistic Regression* berfungsi sebagai model linier yang sederhana (*baseline model*), sedangkan *Random Forest* digunakan sebagai model *ensemble* yang menggabungkan beberapa pohon keputusan yang unggul dalam menangkap pola non-linier dan interaksi antar fitur.

Proses pelatihan dilakukan menggunakan metode *Time Series Split Cross Validation* dengan lima lipatan untuk mengevaluasi performa model secara konsisten. Metode ini dipilih karena data yang digunakan merupakan data deret waktu (*time-series*) dari pembacaan sensor, sehingga penting untuk menjaga urutan kronologis data dan menghindari kebocoran informasi (*data leakage*) dari masa depan ke masa lalu. Pada setiap lipatan, model dilatih menggunakan data dari periode sebelumnya dan divalidasi pada data dari periode setelahnya, dengan ukuran *window training* yang terus bertambah (*expanding window approach*).

Pemilihan *Time Series Split* dibandingkan dengan *traditional cross-validation* didasarkan pada karakteristik data sensor yang memiliki ketergantungan temporal. Metode *Stratified K-Fold* dapat menyebabkan *data leakage* karena menggunakan informasi masa depan untuk memprediksi masa lalu, sedangkan *Time Series Split* memastikan setiap *fold* pelatihan hanya menggunakan data historis dan divalidasi pada periode berikutnya (Fahad, 2024). Pendekatan ini lebih cocok untuk simulasi kondisi produksi dimana model hanya memiliki akses pada data masa lalu.

Evaluasi model menggunakan *F1-score* sebagai metrik utama karena penelitian ini merupakan *binary classification problem* dengan distribusi kelas tidak seimbang. Berbeda dengan *traditional time series forecasting* yang fokus pada prediksi nilai kontinu menggunakan RMSE atau MAE, deteksi *downtime* memerlukan metrik yang dapat menangani target diskrit (0 = Normal, 1 = *Downtime*) (Hyndman & Athanasopoulos, 2021). F1-Score memberikan keseimbangan optimal antara *precision*

untuk mengurangi *false alarm* dan *recall* untuk memastikan deteksi maksimal kasus *downtime* kritis. Setelah proses *cross validation* (CV), masing-masing model dilatih kembali menggunakan seluruh data latih yang telah melalui proses penskalaan, sehingga menghasilkan dua model terlatih yang siap diuji pada tahap evaluasi.

Tahap pemodelan dilakukan dengan membangun dua model pembanding, yaitu *Logistic Regression* dan *Random Forest*. Konfigurasi parameter untuk masing-masing model dijelaskan dalam tabel 4.5 dan 4.6.

Tabel 4. 1 Parameter *Logistic Regression*

Parameter	<i>Logistic Regression</i>	Penjelasan
<i>random_state</i>	42	Memastikan konsistensi hasil
<i>max_iter</i>	1000	Jumlah maksimum iterasi untuk konvergensi
<i>C</i>	0.1	Parameter regularisasi untuk mencegah <i>overfitting</i>

Tabel 4. 2 Parameter *Random Forest*

Parameter	<i>Random Forest</i>	Penjelasan
<i>random_state</i>	42	Memastikan konsistensi hasil
<i>n_estimators</i>	100	Jumlah pohon keputusan yang dibentuk
<i>max_depth</i>	4	Kedalaman maksimum setiap pohon
<i>min_samples_split</i>	30	Jumlah minimum sampel sebelum pemisahan cabang
<i>min_samples_leaf</i>	15	Jumlah minimum sampel pada setiap daun
<i>max_features</i>	0.5	Proporsi fitur yang dipertimbangkan setiap pemisahan
<i>bootstrap</i>	<i>True</i>	Pengambilan sampel acak dengan pengembalian
<i>oob_score</i>	<i>True</i>	Menghitung akurasi menggunakan data <i>out-of-bag</i>

3.5.3.7 Evaluasi Model

Evaluasi model *time-series* mempertimbangkan aspek temporal data. Pada *Time Series Split*, setiap *fold* menggunakan *window training* yang semakin besar dan *window validasi* yang bergerak maju dalam waktu. Hal ini memungkinkan evaluasi

Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

yang lebih robust terhadap kemampuan model dalam menggeneralisasi pola temporal dan menghindari *overfitting* pada pola-pola yang hanya berlaku pada periode tertentu.

Proses evaluasi dilakukan dengan menghitung *F1-score* pada setiap *fold validation*, kemudian mengambil rata-rata dan standar deviasi untuk mengukur konsistensi performa model sepanjang periode waktu. Pendekatan ini memberikan gambaran yang lebih realistis tentang performa model dalam kondisi operasional dimana model akan terus menerima data baru secara berurutan.

Tabel 3. 7 Metrik evaluasi yang dipakai pada penelitian ini

Metrik	Definisi	Alasan Penggunaan
<i>CV F1-Score</i>	Rata-rata <i>F1-score</i> dari validasi silang selama pelatihan	Mengukur konsistensi performa model
<i>Test Accuracy</i>	Persentase prediksi benar dari total prediksi pada data uji	Performa umum, tapi bisa menyesatkan pada data <i>imbalanced</i>
<i>Overall F1</i>	<i>Harmonic mean</i> dari <i>precision</i> dan <i>recall</i> keseluruhan	Keseimbangan performa untuk semua kelas
<i>Downtime Precision</i>	Dari semua prediksi <i>downtime</i> , berapa persen yang benar	Mengurangi false alarm, tapi bukan prioritas utama
<i>Downtime Recall</i>	Dari semua <i>downtime</i> aktual, berapa persen yang terdeteksi	Prioritas #1: deteksi semua kasus <i>downtime</i> kritis
<i>Downtime F1</i>	<i>Harmonic mean precision</i> dan <i>recall</i> untuk kelas <i>downtime</i>	Prioritas #2: keseimbangan optimal untuk target utama
AUC	Kemampuan model membedakan kelas positif dan negatif pada berbagai <i>threshold</i> .	Prioritas #3: kemampuan diskriminasi keseluruhan
<i>Overfitting</i>	Selisih <i>CV F1-Score</i> dengan <i>Test F1</i> (negatif = tidak overfit)	Prioritas #4: memastikan generalisasi yang baik

Evaluasi model dilakukan menggunakan berbagai metrik untuk memberikan gambaran komprehensif tentang performa prediksi. Metrik evaluasi mencakup CV F1-

Score untuk mengukur konsistensi model selama pelatihan menggunakan *Time Series Split validation*, *Test Accuracy* untuk performa umum, *Overall F1* sebagai keseimbangan keseluruhan, serta AUC untuk kemampuan diskriminasi model di berbagai *threshold*. Selain itu, dilakukan perhitungan metrik secara khusus pada kelas *downtime* (label 1) untuk mengetahui kemampuan model dalam mendeteksi kondisi yang berpotensi menyebabkan perangkat berhenti beroperasi.

Metrik khusus ini meliputi *Downtime Precision* yang menunjukkan akurasi prediksi *downtime* (dari semua prediksi *downtime*, berapa persen yang benar), *Downtime Recall* yang mengukur kemampuan deteksi semua kasus *downtime* aktual (dari semua *downtime* yang sebenarnya terjadi, berapa persen yang berhasil dideteksi), dan *Downtime F1-score* sebagai *harmonic mean* antara keduanya untuk memberikan keseimbangan optimal pada kelas target utama. Tingkat *overfitting* dievaluasi melalui selisih antara *CV F1-Score* dengan *Test F1-Score*, di mana nilai negatif menunjukkan tidak terjadi *overfitting* bahkan model berkinerja lebih baik pada data uji.

Untuk aplikasi prediksi *downtime*, empat metrik terpenting yang menjadi fokus evaluasi adalah *Downtime Recall* (prioritas tertinggi untuk deteksi semua kasus *downtime* kritis), *Downtime F1* (keseimbangan optimal untuk target utama), AUC (kemampuan diskriminasi keseluruhan), dan *Overfitting* (memastikan generalisasi yang baik). Prioritas pada *Downtime Recall* didasarkan pada prinsip bahwa dalam maintenance prediktif, lebih baik menghasilkan *false alarm* daripada melewatkan deteksi *downtime* yang dapat berakibat fatal pada peralatan. Model terbaik ditentukan berdasarkan kombinasi keempat metrik tersebut, dengan *Random Forest* menunjukkan superioritas pada seluruh metrik kunci.

3.5.3.8 Visualisasi Model

Tahap visualisasi model dilakukan untuk memberikan gambaran yang lebih jelas terkait performa model dan kontribusi setiap fitur dalam proses prediksi. Visualisasi yang digunakan meliputi *confusion matrix* dan *feature importance*. *Confusion matrix* digunakan untuk menampilkan distribusi hasil prediksi dibandingkan dengan kondisi aktual. Matriks ini membantu mengidentifikasi jumlah *true positive*,

Geralda Livia Nugraha, 2025

SISTEM MONITORING DAN PREDIKSI DOWNTIME MODEM BERBASIS ENERGI SURYA MENGGUNAKAN SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

true negative, *false positive*, dan *false negative* pada data uji. Dengan demikian, dapat dilihat secara langsung bagaimana kemampuan model dalam mengklasifikasikan kondisi *downtime* maupun kondisi normal. Pada penelitian ini, confusion matrix dibuat untuk masing-masing model, yaitu *Logistic Regression* dan *Random Forest*, sehingga perbandingan kinerjanya dapat dilakukan secara visual. Selain itu, untuk model *Random Forest*, dilakukan analisis *feature importance* untuk mengetahui fitur mana yang paling berpengaruh terhadap hasil prediksi. Nilai *importance* yang tinggi menunjukkan bahwa fitur tersebut memiliki kontribusi besar dalam pengambilan keputusan model. Sepuluh fitur teratas dengan nilai *importance* tertinggi divisualisasikan dalam bentuk diagram batang horizontal, sehingga memudahkan interpretasi dan memberikan wawasan fitur-fitur kunci yang berperan dalam prediksi *downtime*.

3.5.3.9 Inferensi dan Implementasi Model

Tahap inferensi dan implementasi model dilakukan untuk menguji kinerja model terbaik pada data yang belum pernah dilihat sebelumnya, serta memastikan model dapat digunakan untuk prediksi secara *real-time* atau pada data sensor baru. Pertama, model terbaik ditentukan berdasarkan nilai *Downtime F1-score* tertinggi dari hasil evaluasi. Model ini kemudian digunakan untuk melakukan prediksi pada sampel data dari test set yang dipilih secara manual maupun acak. Untuk setiap sampel, ditampilkan label prediksi, label aktual, status prediksi (benar atau salah), serta tingkat kepercayaan (*confidence score*) untuk masing-masing kelas (normal dan *downtime*) jika model mendukung probabilitas prediksi. Selanjutnya, dilakukan pengujian menggunakan data sensor baru (*custom input*) yang disusun sesuai format fitur pada data latih. Data ini melalui proses *preprocessing* dan penskalaan menggunakan parameter yang sama dengan data latih. Model kemudian menghasilkan prediksi beserta tingkat kepercayaan untuk masing-masing kelas. Pendekatan ini memungkinkan sistem untuk diintegrasikan pada perangkat pemantauan jaringan secara langsung, sehingga dapat memberikan peringatan dini terhadap potensi

downtime berdasarkan kondisi sensor saat ini, baik dari data historis maupun data sensor yang diperbarui secara *real-time*.

Sebagai tambahan, penelitian ini juga mengimplementasikan prediksi pada data terbaru yang berasal dari periode pemantauan setelah tahap pelatihan selesai. Hal ini bertujuan untuk menguji robustnes model ketika menghadapi data unseen yang benar-benar baru. Proses prediksi dilakukan dengan alur yang sama, yakni data cleaning, feature engineering, penskalaan, hingga inferensi menggunakan model terbaik. Hasil prediksi tidak hanya ditampilkan dalam bentuk nilai klasifikasi (*downtime* atau normal), tetapi juga divisualisasikan dalam grafik yang membandingkan label aktual dengan prediksi model serta probabilitas *downtime*. Dengan cara ini, kinerja model pada data terbaru dapat dievaluasi secara lebih transparan, sekaligus menjadi bukti implementasi model dalam konteks operasional yang mendekati kondisi nyata.