



**PREDIKSI DATA CUACA BERDASARKAN ALGORITMA XGBOOST  
PADA SISTEM STASIUN CUACA BERBASIS LORA**

**PROPOSAL TUGAS AKHIR**

Diajukan sebagai salah satu syarat menyelesaikan pendidikan pada  
Program Studi Sarjana Terapan Teknologi Rekayasa Otomasi  
Departemen Teknologi Industri Sekolah Vokasi  
Universitas Diponegoro

Oleh:

Jidan Humaidi

40040321650029

**PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI REKAYASA OTOMASI  
DEPARTEMEN TEKNOLOGI INDUSTRI SEKOLAH VOKASI  
UNIVERSITAS DIPONEGORO  
SEMARANG  
2025**

## PROPOSAL TUGAS AKHIR

### PREDIKSI DATA CUACA BERDASARKAN ALGORITMA XGBOOST PADA SISTEM STASIUN CUACA BERBASIS LORA

Diajukan oleh:

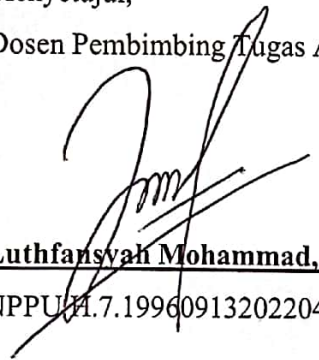
Jidan Humaidi

40040321650029

TELAH DISETUJUI DAN DITERIMA DENGAN BAIK OLEH

Menyetujui,

Dosen Pembimbing Tugas Akhir

  
Luthfansyah Mohammad, S.Tr.T, M.T

NPPU 11.7.199609132022041001

Tanggal 21 MARET 2025.

Mengetahui,

Ketua

Program Studi S.Tr Teknologi Rekayasa Otomasi

Departemen Teknologi Industri Sekolah Vokasi

Universitas Diponegoro

Priyo Sasmoko, S.T., M.Eng

NIP. 197009161998021001

Tanggal

## DAFTAR ISI

<b>PROPOSAL TUGAS AKHIR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>v</b>
<b>DAFTAR TABEL .....</b>	<b>vii</b>
<b>DAFTAR SINGKATAN DAN SIMBOL .....</b>	<b>viii</b>
<b>ABSTRAK .....</b>	<b>xii</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Tujuan .....	3
1.4    Manfaat .....	3
1.5    Pembatasan Masalah .....	4
1.6    Sistematika Proposal Tugas Akhir .....	4
<b>BAB II .....</b>	<b>6</b>
<b>DASAR TEORI .....</b>	<b>6</b>
2.1    Tinjauan Pustaka .....	6
2.2    Stasiun Cuaca .....	8
2.3    LoRa.....	9
2.4    LoRa32 V2.1_1.6.....	13
2.5    Sensor Cuaca HONDE HD-WSM-U-07.....	15
2.6    Protokol RS-485.....	19
2.7    Modbus.....	20
2.8    Modbus RTU.....	22
2.9    UART TTL <i>to</i> RS-485 <i>Converter</i> .....	23
2.10   Modul Mini W5500 SPI <i>to</i> LAN .....	25
2.11   SMPS 24V-5A .....	27
2.12   Rangkaian Buck <i>Converter</i> 5V LM2596 .....	29
2.13   DIP <i>Switch</i> 2 pin .....	30
2.14 <i>Push button</i> 4 Kaki.....	31
2.15   MQTT .....	31
2.16   Server .....	32

2.17	Linux .....	33
2.18	MySQL.....	34
2.19	Flask .....	35
2.20	Algoritma <i>Machine learning</i> XGBoost.....	36
2.21	Evaluasi Metrik .....	40
2.22	Google <i>Colab</i> .....	41
2.23	Python .....	41
2.24	Scikit-Learn.....	42
2.25	Visual Studio Code .....	42
2.26	Arduino IDE.....	43
<b>BAB III.....</b>		<b>45</b>
<b>METODE PENELITIAN .....</b>		<b>45</b>
3.1	Diagram Blok dan <i>Flowchart</i> .....	45
3.1.1	Blok diagram.....	45
3.1.2	Rangkaian elektrikal keseluruhan .....	50
3.1.3	Rangkaian elektrikal modul komunikasi .....	51
3.1.4	<i>Flowchart</i> sistem utama.....	55
3.1.5	<i>Flowchart</i> pengelolaan data.....	58
3.1.6	<i>Flowchart</i> sistem prediksi.....	60
3.2	Proyeksi 3 Dimensi Perangkat Keras Sistem .....	61
3.3	Teknik Fabrikasi.....	63
3.3.1	Alat dan bahan .....	63
3.3.2	Perancangan perangkat keras .....	64
3.3.3	Perancangan perangkat lunak .....	65
3.4	Jadwal Pembuatan dan Penyusunan Tugas Akhir.....	67
<b>DAFTAR PUSTAKA .....</b>		<b>68</b>

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Stasiun Cuaca [14].....	8
<b>Gambar 2. 2</b> Perbandingan Protokol Komunikasi [16] .....	9
<b>Gambar 2. 3</b> Paket Data LoRa .....	10
<b>Gambar 2. 4</b> Arsitektur LoRa Point-to-Point [18].....	12
<b>Gambar 2. 5</b> Pinout LoRa32 V2.1_1.6 [19] .....	13
<b>Gambar 2. 6</b> Bagian-Bagian Sensor Cuaca HONDE HD-WSM-U-07 [8].....	16
<b>Gambar 2. 7</b> Konfigurasi Kabel Komunikasi HONDE HD-WSM-U-07 [8] .....	17
<b>Gambar 2. 8</b> Perbandingan RS-485 dengan Komunikasi Lain [23] .....	19
<b>Gambar 2. 9</b> Alur Data Modbus [26].....	21
<b>Gambar 2. 10</b> Format Frame Modbus RTU [26].....	22
<b>Gambar 2. 11</b> UART TTL to RS-485 Converter [27] .....	23
<b>Gambar 2. 12</b> Wiring UART TTL to RS-485 Converter .....	25
<b>Gambar 2. 13</b> Modul Mini W5500 SPI to LAN [30] .....	25
<b>Gambar 2. 14</b> Wiring Modul Mini W5500 SPI to LAN.....	27
<b>Gambar 2. 15</b> SMPS 24V-5A [32] .....	27
<b>Gambar 2. 16</b> Pinout SMPS 24V-5A [34].....	28
<b>Gambar 2. 17</b> Rangkaian Buck Converter 5V LM2596.....	29
<b>Gambar 2. 18</b> Dip Switch 2 pin [37] .....	30
<b>Gambar 2. 19</b> Push button 4 Kaki.....	31
<b>Gambar 2. 20</b> Alur Kerja MQTT [39] .....	31
<b>Gambar 2. 21</b> Alur Kerja Client-Server.....	32
<b>Gambar 2. 22</b> Arsitektur MySQL[43] .....	35
<b>Gambar 2. 23</b> Lingkup Machine Learning [45].....	36
<b>Gambar 2. 24</b> Flowchart XGBoost [47] .....	37
<b>Gambar 2. 25</b> Arsitektur Development View Arduino IDE [55] .....	43
<b>Gambar 3. 1</b> Diagram Blok Sistem Modul Transmitter .....	45
<b>Gambar 3. 2</b> Diagram Blok Sistem Modul Receiver.....	49
<b>Gambar 3. 3</b> Rangkaian Elektrikal .....	50
<b>Gambar 3. 4</b> Skematik Rangkaian Blok Power .....	51

<b>Gambar 3. 5</b> Skematik Rangkaian Blok Input.....	51
<b>Gambar 3. 6</b> Skematik Rangkaian Blok Kontrol.....	52
<b>Gambar 3. 7</b> Skematik Rangkaian Blok Output .....	53
<b>Gambar 3. 8</b> Flowchart Sistem Utama .....	55
<b>Gambar 3. 9</b> Flowchart sistem.....	58
<b>Gambar 3. 10</b> Flowchart sistem prediksi .....	60
<b>Gambar 3. 11</b> Proyeksi 3 Dimensi Perangkat Keras Sistem Outdoor .....	62
<b>Gambar 3. 12</b> Proyeksi 3 Dimensi Perangkat Keras Sistem Indoor .....	62
<b>Gambar 3. 13</b> Proyeksi 3 Dimensi Perangkat Keras .....	63

## DAFTAR TABEL

<b>Tabel 2. 1</b> Tinjauan Pustaka .....	6
<b>Tabel 2. 1</b> Lanjutan.....	7
<b>Tabel 2. 2</b> Spreading factor [17].....	11
<b>Tabel 2. 3</b> Coding Rate [17] .....	11
<b>Tabel 2. 4</b> Spesifikasi LoRa32 V2.1_1.6 [19] .....	14
<b>Tabel 2. 5</b> Spesifikasi Parameter Sensor Cuaca HONDE HD-WSM-U-07 [8] ...	17
<b>Tabel 2. 6</b> Alamat Komunikasi RS-485 [8].....	18
<b>Tabel 2. 7</b> Spesifikasi RS-485 [25] .....	20
<b>Tabel 2. 8</b> Function Code Modbus [26] .....	22
<b>Tabel 2. 9</b> Spesifikasi UART TTL to RS-485 Converter [29] .....	24
<b>Tabel 2. 10</b> Spesifikasi Modul Mini W5500 SPI to LAN [31] .....	26
<b>Tabel 2. 11</b> Spesifikasi SMPS 24V-5A [33] .....	28
<b>Tabel 2. 12</b> Spesifikasi Rangkaian Buck Converter 5V LM2596 [36] .....	29
<b>Tabel 2. 12</b> Lanjutan.....	30
<b>Tabel 2. 13</b> Spesifikasi PC Server .....	33
<b>Tabel 2. 14</b> Spesifikasi Google Colab [51] .....	41
<b>Tabel 3. 1</b> Konfigurasi DIP Switch .....	48
<b>Tabel 3. 2</b> Konfigurasi I/O Modul Komunikasi .....	53
<b>Tabel 3. 3</b> Lanjutan.....	54
<b>Tabel 3. 4</b> Alat dan Bahan .....	63
<b>Tabel 3. 5</b> Lanjutan.....	63
<b>Tabel 3. 6</b> Konsumsi Daya Setiap Komponen .....	64
<b>Tabel 3. 6</b> Lanjutan.....	65
<b>Tabel 3. 7</b> Jadwal Pelaksanaan Tugas Akhir .....	67

## DAFTAR SINGKATAN DAN SIMBOL

Singkatan	Definisi
AC	<i>Alternating Current</i>
AI	<i>Artificial Intelligence</i>
API	<i>Application Programming Interface</i>
AWS	<i>Automatic Weather Station</i>
BMKG	Badan Meteorologi, Klimatologi, dan Geofisika
CPU	<i>Central Processing Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
DC	<i>Direct Current</i>
DCL	<i>Data Control Language</i>
DDL	<i>Data Definition Language</i>
DML	<i>Data Manipulation Language</i>
ECMWF	<i>European Centre for Medium-Range Weather Forecasts</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
GPIO	<i>General Purpose Input/Output</i>
GPU	<i>Graphics Processing Unit</i>
GSM	<i>Global System for Mobile Communications</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
LAN	<i>Local Area Network</i>
LoRa	<i>Long Range</i>
MCU	<i>Microcontroller Unit</i>
ML	<i>Machine Learning</i>
MODBUS	<i>Modular Data Bus</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
MSE	<i>Mean Squared Error</i>



NASA	<i>National Aeronautics and Space Administration</i>
PC	<i>Personal Computer</i>
PLN	<i>Perusahaan Listrik Negara</i>
R <sup>2</sup>	<i>Coefficient of Determination</i>
RAM	<i>Random Access Memory</i>
RDBMS	<i>Relational Database Management System</i>
RMSE	<i>Root Mean Squared Error</i>
RS-485	<i>Recommended Standard 485</i>
RTU	<i>Remote Terminal Unit</i>
SD	<i>Secure Digital</i>
SNR	<i>Signal-to-Noise Ratio</i>
SoC	<i>System on a Chip</i>
SPI	<i>Serial Peripheral Interface</i>
SPI	<i>Serial Peripheral Interface</i>
SQL	<i>Structured Query Language</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TPU	<i>Tensor Processing Unit</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
VS Code	<i>Visual Studio Code</i>
Wi-Fi	<i>Wireless Fidelity</i>
XGBoost	<i>Extreme Gradient Boosting</i>

Simbol	Definisi	Satuan
A	Arus	Ampere
$f_t(x_i)$	Prediksi dari model yang baru	-
$f_k(x_i)$	Prediksi dari pohon ke- $k$ untuk input $x_i$	-
$K$	Jumlah pohon	-

$L(y_i, \hat{y}_i)$	Fungsi <i>loss</i> yang mengukur kesalahan antara nilai sebenarnya ( $y_i$ ) dan nilai prediksi ( $\hat{y}_i$ )	-
$L_j$	Fungsi <i>loss</i> untuk target ke- $j$	-
$\lambda$	Koefisien regulasi L2 dalam XGBoost	-
$MSE$	<i>Mean Squared Error</i> , ukuran kesalahan prediksi	-
$n$	Jumlah total data	-
$Obj^{(t)}$	Fungsi objektif dalam XGBoost pada iterasi $t$	-
$\Omega(f_k)$	Reguralisasi	-
$Obj^{(t)}$	Fungsi objektif dalam XGBoost pada iterasi $t$	-
$P$	Daya	Watt
$RMSE$	<i>Root Mean Squared Error</i> , akar kuadrat dari MSE	-
$R^2$	Koefisien determinasi, menunjukkan seberapa baik model menjelaskan variabilitas data	-
$T$	Jumlah daun di pohon	-
$V$	Tegangan	Volt
$x_i$	Fitur input dari data ke- $i$	-
$y_{ij}$	Nilai sebenarnya dari target ke- $j$ untuk sampel ke- $i$	-
$\hat{y}_{ij}$	Prediksi untuk target ke- $j$ untuk sampel ke- $i$	-
$\gamma$	<i>Hyperparameter</i> yang mengontrol kekuatan regularisasi	-
$\hat{y}_i^{(t-1)}$	Prediksi dari model sebelumnya	-

---

$\hat{y}_i$	Nilai prediksi dari XGBoost untuk data ke-i	-
$y_i$	Nilai aktual dari data ke-i	-

---

## ABSTRAK

### **PREDIKSI DATA CUACA BERDASARKAN ALGORITMA XGBOOST PADA SISTEM STASIUN CUACA BERBASIS LORA**

Jidan Humaidi

Program Studi Teknologi Rekayasa Otomasi, Sekolah Vokasi, Universitas  
Diponegoro

Kondisi cuaca sangat berpengaruh pada peradaban umat manusia. Di masa modern ini prediksi cuaca sangat dibutuhkan guna mengarahkan peradaban ke arah yang lebih baik. Misalnya, di sektor energi, transportasi, dan konstruksi, prediksi cuaca meningkatkan potensi manfaat sebesar Rp2.500 triliun per tahun. Stasiun cuaca adalah salah satu sistem yang dirancang untuk mengambil data cuaca di permukaan bumi. Kemudian, data cuaca ini menjadi komponen utama dalam prediksi cuaca. Akan tetapi, instalasi stasiun cuaca di Indonesia mengalami beberapa masalah. Misalnya, kondisi topografi Indonesia yang beragam dan jaringan jaringan 4G/LTE di Indonesia yang masih mengalami variasi kecepatan hingga  $3.2 \times (5.7-18.5 \text{ Mbps})$  dan belum merata di seluruh wilayah Indonesia. Ditambah, data cuaca dari satelit tidak bisa merepresentasikan kondisi cuaca di permukaan bumi. Oleh sebab itu, di buatlah stasiun cuaca berbasis LoRa dengan prediksi cuaca menggunakan algoritma XGBoost. Teknologi LoRa mengakomodasi kebutuhan transmisi data jarak jauh (90% lebih jauh daripada komunikasi via Wi-Fi, Zigbee dan Bluetooth) dengan konsumsi daya yang rendah. Ditambah, sistem ini dilengkapi dengan sensor cuaca HONDE HD-WSM-U-07 yang bisa mengambil data suhu, kelembapan, tekanan atmosfer, kecepatan angin, arah angin, curah hujan, dan luminansi. Selain itu, model prediksi cuaca untuk regresi multivariat data titik tunggal harus memiliki matrik evaluasi yang baik, seperti algoritma XGBoost yang mendapat nilai  $R^2$  sebesar 0,9 untuk memprediksi data iklim Kota Semarang tahun 2020-2023. Dengan adanya sistem ini, diharapkan stasiun cuaca bisa dipasang di daerah terpencil dan tidak mengandalkan sinyal GSM sebagai transmisi data.

**Kata kunci:** Stasiun Cuaca, LoRa, Prediksi Cuaca, XGBoost.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Hajat hidup manusia sangat bergantung pada kondisi cuaca. Prediksi cuaca yang akurat memberikan peran yang besar di beberapa sektor. Misalnya, di sektor energi, transportasi, dan konstruksi, prediksi cuaca meningkatkan potensi manfaat sebesar Rp2,500 triliun per tahun [1]. Sedangkan, pemanfaatan prediksi angin di sektor energi milik Amerika Serikat dapat menghemat Rp2,4 triliun per tahun [1]. Di lain sisi, investasi dalam prediksi cuaca di Amerika Serikat berhasil memberi keuntungan ekonomi sebesar Rp525 triliun per tahun [2].

Data untuk pelatihan model prediksi cuaca diambil dari beberapa sumber. Misalnya, satelit geostasioner dan *polar-orbiting* berperan dalam observasi data atmosfer [3]. Sedangkan, radar cuaca seperti *Doppler on Wheels*, memberikan informasi mengenai curah hujan dan badai. Selain itu, data atmosfer dari bawah dikumpulkan oleh *drone* cuaca. Di sisi lain, stasiun cuaca digunakan untuk memberikan data cuaca di permukaan bumi [4].

Stasiun cuaca sangat penting adanya di dalam prediksi cuaca. Salah satu alasannya, data yang diambil satelit, tidak sepenuhnya merepresentasikan data cuaca di permukaan bumi. Misalnya, satelit Terra milik NASA dengan sensor MODIS yang digunakan untuk mengukur suhu permukaan laut, memiliki bias sebesar 0,23°C dan RMSE sebesar 0,83°C dibandingkan dengan data suhu lokal dari pelampung [5].

Instalasi stasiun cuaca di Indonesia memiliki tantangan tersendiri. Salah satunya adalah kecepatan jaringan 4G/LTE di Indonesia yang masih mengalami variasi kecepatan hingga  $3,2 \times (5,7-18,5 \text{ Mbps})$  dan belum merata di seluruh wilayah Indonesia [6]. Hal ini menyebabkan, implementasi IoT, khususnya di bidang *remote sensing* seperti stasiun cuaca, tidak dapat berjalan dengan maksimal. Maka dari itu, dibutuhkan stasiun cuaca yang bisa mentransmisi data walaupun tanpa jaringan internet sekalipun dan memiliki beberapa fitur tambahan seperti *monitoring* dan prediksi lokal untuk kebutuhan prediksi lokal.

Stasiun cuaca berbasis LoRa dapat menjadi solusi. Teknologi LoRa mengakomodasi kebutuhan transmisi data jarak jauh (90% lebih jauh daripada komunikasi via Wi-Fi, Zigbee dan Bluetooth) dengan konsumsi daya yang rendah [7]. Selain itu, stasiun cuaca harus dilengkapi oleh sensor cuaca yang bisa mengambil beberapa variabel cuaca. Sebagai contoh, sensor cuaca HONDE HD-WSM-U-07 yang bisa mengambil data suhu, kelembapan, tekanan atmosfer, kecepatan angin, arah angin, curah hujan, dan iradiasi matahari [8]. Di samping itu, model prediksi cuaca untuk regresi multivariat data titik tunggal harus memiliki matriks evaluasi yang baik, seperti algoritma XGBoost yang mendapat nilai  $R^2$  sebesar 0,9 untuk memprediksi data iklim Kota Semarang tahun 2020-2023 [9].

Beberapa peneliti sudah mengembangkan beberapa variasi stasiun cuaca. Misalnya, Soy dan Delay (2021) yang membuat prototipe stasiun cuaca dengan menggunakan LoRa untuk lahan pertanian [10]. Demikian pula, Divya *et al.* (2022) membuat sistem pemantauan data cuaca menggunakan ESP32 [11]. Sapkota *et al.* (2023) mengembangkan sistem prediksi cuaca dengan metode regresi linear [12]. Sementara itu, Rivera *et al.* (2023) merancang stasiun cuaca berbiaya rendah dengan menggunakan sensor BME280 [13]. Tapi, penelitian di atas belum mengintegrasikan semua fitur, seperti *monitoring* data cuaca, transmisi data menggunakan komunikasi LoRa, sensor yang akurat, dan model yang tepat untuk memprediksi data cuaca titik tunggal.

Oleh karena itu, penulis mengangkat judul penelitian “Prediksi Data Cuaca Berdasarkan Algoritma XGBoost Pada Sistem Stasiun Cuaca Berbasis LoRa”. LoRa32 V2.1\_1.6 digunakan sebagai modul MCU dan komunikasi LoRa Versi 2.1 yang memungkinkan pengguna mentransmisi data cuaca dari modul *transmitter* ke modul *receiver* tanpa jaringan GSM. Selain hal tersebut, sensor cuaca dari Honde Technology Co. Ltd. dengan seri produk HONDE HD-WSM-U-07 digunakan sebagai pengambil data dan MQTT sebagai protokol komunikasi transmisi data ke server. Demikian pula, algoritma XGBoost digunakan dalam model prediksi cuaca lokal.

Diharapkan, dengan menggunakan komunikasi LoRa maka komunikasi yang bergantung pada GSM tidak diperlukan lagi, sehingga stasiun cuaca dapat

ditempatkan pada area terpencil. Selain itu, diharapkan semua variabel dapat dicatat dengan sistem *logger* yang baik, dan didapatkan hasil prediksi data cuaca yang baik.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah diuraikan, terdapat beberapa rumusan masalah yaitu:

1. Bagaimana penerapan komunikasi LoRa dalam sistem stasiun cuaca guna meningkatkan jangkauan transmisi data?
2. Bagaimana konfigurasi sistem sensor *transmitter receiver* pada stasiun cuaca berbasis LoRa guna meningkatkan presisi penerimaan data yang baik?
3. Bagaimana tingkat keakurasian model dengan algoritma XGBoost berdasarkan evaluasi metrik pada regresi multivariat dengan data cuaca titik tunggal?

## 1.3 Tujuan

1. Mengetahui penerapan komunikasi LoRa dalam sistem stasiun cuaca.
2. Merancang konfigurasi sistem sensor *transmitter receiver* pada stasiun cuaca berbasis LoRa.
3. Mengetahui tingkat akurasi model dengan algoritma XGBoost berdasarkan evaluasi metrik pada regresi multivariat dengan data cuaca titik tunggal.

## 1.4 Manfaat

1. Bagi Penulis

Dengan menyusun proposal ini, penulis dapat mendokumentasikan pembuatan stasiun cuaca berbasis LoRa. Di lain sisi, penulis dapat mengejawantahkan ide penerapan algoritma XGBoost pada *dataset* titik tunggal untuk prediksi regresi multivariat yang dapat di aplikasikan pada stasiun cuaca lokal berbasis LoRa.

2. Bagi Masyarakat

Data cuaca lokal yang akurat sesuai dengan yang terjadi di permukaan bumi dapat diterima oleh masyarakat. Hal ini berbeda dengan kondisi sebelumnya, yakni data cuaca untuk prediksi yang diterima masyarakat

adalah hasil prediksi berdasar pada data satelit atau data dari stasiun cuaca milik BMKG yang terletak jauh dari pemukiman masyarakat.

### 3. Bagi Pembaca

Sebagaimana slogan dalam dunia sains, “*Standing on the shoulders of giants*”, penelitian ini dapat menjadi landasan bagi pembaca untuk mengembangkan atau menemukan inovasi baru di bidang peramalan cuaca, khususnya dalam penerapan algoritma XGBoost dengan data titik tunggal untuk prediksi data cuaca menggunakan regresi multivariat pada stasiun cuaca berbasis LoRa.

## 1.5 Pembatasan Masalah

1. Model algoritma yang digunakan untuk prediksi regresi multivariat terbatas pada XGBoost dengan output numerik.
2. Variabel data cuaca yang diprediksi adalah temperatur, kelembapan, curah hujan, dan iradiasi per hari untuk 7 hari ke depan.
3. Jarak radio pengiriman komunikasi LoRa berkisar pada 0-500 meter dengan halangan beberapa bangunan.
4. Tempat pengujian hanya dilakukan di sekitar lingkungan Program Studi Teknologi Rekayasa Otomasi, Sekolah Vokasi, Universitas Diponegoro.
5. *Frame* waktu pengambilan data cuaca selama 3 bulan.
6. Dataset yang digunakan untuk membuat model adalah data cuaca dari AWS Universitas Diponegoro (ID 160030), ECMWF, NASA *Power* dan data input dari sensor cuaca berbasis LoRa.
7. Lingkup penelitian tidak membahas klasifikasi peramalan cuaca.

## 1.6 Sistematika Proposal Tugas Akhir

Proposal Tugas Akhir ini dibagi menjadi tiga bab yang saling berhubungan satu sama lain. Sistematika penulisan dan pembahasan dalam penulisan ini yaitu:

### **BAB 1 PENDAHULUAN**

Pada BAB ini akan dibahas hal yang melatarbelakangi pembuatan suatu alat atau sistem pada Tugas Akhir, rumusan masalah alat atau sistem, tujuan alat atau



sistem, manfaat alat atau sistem, batasan masalah alat atau sistem, dan juga sistematika penulisan.

## **BAB 2 DASAR TEORI**

Pada BAB ini akan dibahas tinjauan pustaka yang berisikan mengenai penelitian terdahulu yang memiliki relevansi dan juga teori-teori tentang komponen maupun *tools* yang dipakai dalam pembuatan Tugas Akhir.

## **BAB 3 METODE PENELITIAN**

Pada BAB ini akan dibahas mengenai tentang komponen-komponen yang dibutuhkan untuk pembuatan alat, BAB ini juga membahas tentang bagaimana sistem berjalan, spesifikasi sistem dan teknik fabrikasinya.

## **BAB 4 PENGUJIAN DAN ANALISA**

Pada BAB ini membahas terkait pengujian alat dan sistem serta analisa alat yang sudah dirancang sebelumnya.

## **BAB 5 PENUTUP**

Pada BAB ini membahas terkait kesimpulan yang didapatkan dari hasil yang sudah diuji dan di analisa serta saran untuk penelitian selanjutnya.

## **DAFTAR PUSTAKA**

Pada bagian ini memuat referensi penulisan serta sumber yang dikutip oleh penulis.

## **LAMPIRAN**

Pada bagian ini berisi hal-hal sisipan dan keterangan oleh penulis yang berisikan tentang apa yang penulis tulis.

## BAB II

### DASAR TEORI

#### 2.1 Tinjauan Pustaka

Setelah dilakukan studi literatur, baik dari konferensi internasional maupun beberapa jurnal dan *paper*, didapatkan beberapa penelitian yang menjadi inspirasi. Beberapa penelitian tersebut yaitu sebagai berikut.

**Tabel 2. 1** Tinjauan Pustaka

No.	Judul Penelitian	Keterangan	Perbaikan
1	<i>A Conceptual Design of LoRa based Weather Moitoring System for Smart Farming</i> (Soy dan Dilay, 2021)	LoRa digunakan untuk memberikan peringatan kepada petani terkait kerusakan tanaman akibat cuaca ekstrem di Turki, contohnya seperti embun beku. <i>Monitoring</i> kondisi dilakukan dengan menggunakan jaringan berdaya rendah berbasis LPWAN yang dikirim dari stasiun pemantauan ke stasiun pengolahan data [10].	Penelitian ini belum memiliki model prediksi cuaca yang dapat bermanfaat sebagai bentuk pencegahan atas cuaca ekstrem di lahan pertanian, sehingga para petani dapat melakukan tindakan <i>preventif</i> guna memaksimalkan hasil panen mereka.
2	<i>IoT Weather Monitoring System Using ESP32</i> (Divya, dkk, 2022)	Sistem IoT untuk pemantauan cuaca dengan menggunakan modul ESP32 yang menampilkan beberapa variabel seperti suhu, kelembapan, tekanan, intensitas Cahaya, dan level CO di suatu area. Data dikirim secara <i>real-time</i> melalui Wi-Fi ke platform ThingSpeak untuk ditampilkan di <i>dashboard</i> . Sistem memiliki ukuran yang kompak, ringan, dan hemat energi. Sistem ini dapat ditenagai oleh panel surya [11].	Penelitian ini masih menggunakan komunikasi Wi-Fi, yang hanya memiliki jangkauan 100-200 meter. Sehingga, modul ini kurang cocok jika diletakan di daerah terpencil yang membutuhkan jangkauan transmisi data lebih dari 1 kilometer. Variabel cuaca penting lainnya juga belum ada, seperti curah hujan, iradiasi, dan kecepatan angin.

Tabel 2. 2 Lanjutan

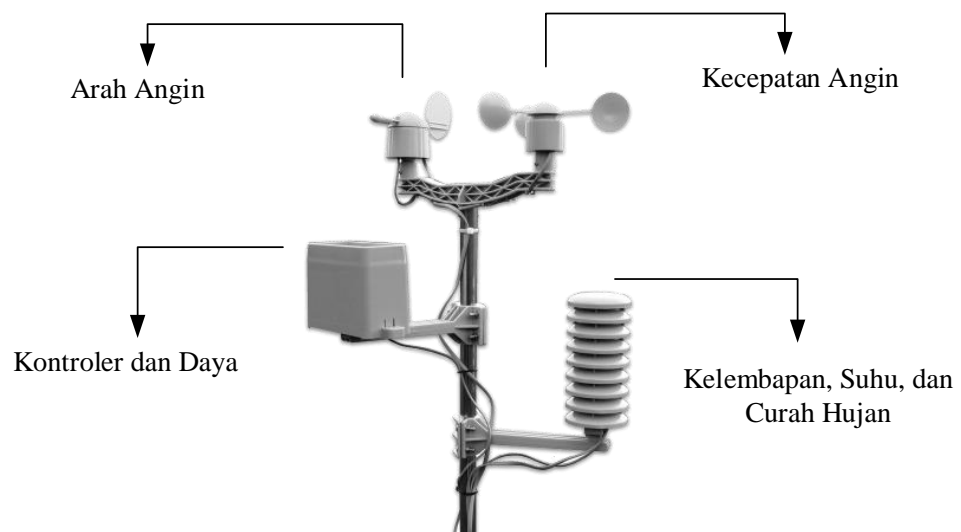
No.	Judul Penelitian	Keterangan	Perbaikan
3	<i>Weather Prediction System Using Linear Regression</i> (Sapkota, dkk, 2023)	Algoritma linear regresi digunakan untuk memprediksi parameter cuaca seperti suhu, curah hujan, kecepatan angin, dan kelembapan. Dengan model yang dibuat dengan dataset yang diambil dari satelit, radar dan stasiun cuaca, model ini mendapatkan akurasi sebesar 86% untuk prediksi suhu, dan 89% untuk prediksi curah hujan [12].	Penelitian ini masih menggunakan algoritma linear regresi yang belum bisa mendeteksi hubungan kompleks dalam data cuaca. Sedangkan, dalam data cuaca, satu fitur dengan fitur lainnya sangat berhubungan.
4	<i>Local Weather Station Design and Development for Cost-Effective Environmental Monitoring and Real-Time Data Sharing</i> (Rivera, dkk, 2023)	Stasiun cuaca dengan menggunakan data historis untuk membuat model Arima yang digunakan untuk memprediksi cuaca jangka pendek. ESP8266 digunakan sebagai MCU utama dengan sensor BME280 sebagai salah satu pengambil data. Data secara <i>real-time</i> ditransmisikan ke platform ThingSpeak melalui Wi-Fi [13].	Penelitian ini memakai sensor BME280 yang memiliki <i>error</i> yang cukup besar. Misalnya, untuk parameter suhu, sensor ini memiliki <i>error</i> sebesar $\pm 1^{\circ}\text{C}$ , dan tekanan atmosfer sebesar $\pm 1\text{hPa}$ . Hal tersebut menyebabkan model yang dibuat dengan data yang memiliki <i>error</i> yang tinggi akan menghasilkan prediksi yang kurang akurat.

Dataset yang digunakan dalam penelitian ini untuk pelatihan model berasal dari data internal input sensor HONDE HD-WSM-U-07 pada stasiun cuaca berbasis LoRa. Selain itu, guna mendapatkan dataset yang lebih komprehensif maka digunakan juga data eksternal yang berasal dari AWS Universitas Diponegoro (ID 160030) serta data cuaca yang berasal dari ECMWF dan NASA *Power*.

Pada penelitian ini, terdapat 2 modul inti, yakni modul *transmitter* sebagai penyalur data dari sensor, dan modul *receiver* sebagai penerima data dari modul *transmitter* melalui komunikasi LoRa versi 2.1 dan dikirim ke server melalui Wi-Fi. Sedangkan, MCU yang digunakan adalah LoRa32 V2.1\_1.6. Di lain sisi, sensor

cuaca HONDE HD-WSM-U-07 digunakan untuk mengambil parameter data cuaca seperti data suhu, kelembapan, tekanan atmosfer, kecepatan angin, arah angin, curah hujan, dan iradiasi matahari. Selanjutnya, MQTT digunakan sebagai protokol komunikasi transmisi data ke server. Demikian pula, XGBoost yang digunakan sebagai algoritma *machine learning* dalam model prediksi cuaca lokal.

## 2.2 Stasiun Cuaca



**Gambar 2. 1** Stasiun Cuaca [14]

Stasiun Cuaca adalah infrastruktur pengamatan dan pengambilan data cuaca yang biasanya di peruntukan untuk pemantauan dan prediksi cuaca. Variabel yang diakuisisi oleh stasiun cuaca adalah temperatur, kelembapan, curah hujan, arah angin, kecepatan angin, tekanan atmosfer, iradiasi matahari, dan sebagainya [15].

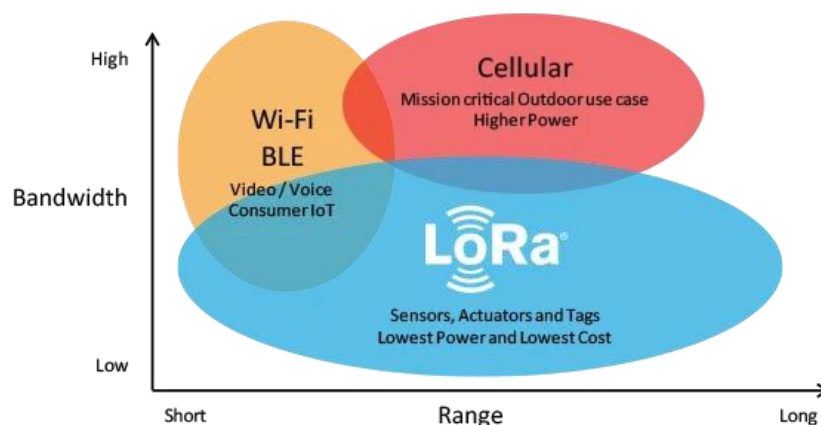
Seperti yang ditunjukkan pada gambar 2.1, biasanya stasiun cuaca dilengkapi dengan beberapa sensor cuaca seperti HONDE HD-WSM-U-07, modul kontrol yang berisi MCU seperti LoRa32 V2.1\_1.6, dan beberapa modul komunikasi tambahan seperti RS-485 untuk memproses dan mentransmisikan data cuaca. Selain itu, terdapat modul daya yang biasanya berisi catu daya dan *converter* tegangan seperti Buck *Converter* guna menjaga agar stasiun cuaca tetap hidup.

Pada penelitian ini, stasiun cuaca digunakan sebagai perangkat utama dalam pengambilan data cuaca yang akan di dukung dengan komunikasi LoRa sebagai

media transmisi data, sehingga stasiun cuaca dapat dipasang di lingkungan terpencil.

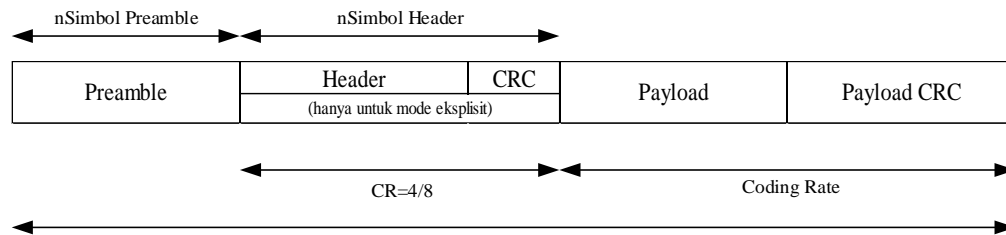
### 2.3 LoRa

LoRa (*Long Range*) adalah teknologi komunikasi nirkabel yang dibuat oleh Semtech untuk kebutuhan transmisi data jarak jauh dengan konsumsi daya yang rendah yang sangat cocok untuk kebutuhan IoT. Selain itu, LoRa menggunakan modulasi *Chirp Spread Spectrum* (CSS) untuk transmisi data dengan frekuensi rendah melalui pita ISM (*Industrial, Scientific, and Medical*) tanpa lisensi, yang memungkinkan transmisi data jarak jauh hingga 10km di area terbuka dan 2-3km di area urban [7].



**Gambar 2. 2** Perbandingan Protokol Komunikasi [16]

Gambar 2.2 menunjukkan bahwa LoRa memiliki jangkauan yang lebih jauh dengan *bandwidth* yang lebih rendah jika dibandingkan dengan teknologi lain seperti Wi-Fi, BLE, dan Seluler. Selain itu, LoRa memiliki efisiensi daya yang sangat baik. LoRa hanya memiliki penerimaan arus sebesar 10mA dan *sleep current* kurang dari 200nA yang mendukung masa hidup baterai. LoRa juga dapat bertahan di kondisi yang memiliki banyak *noise* dengan menggunakan teknologi *spread spectrum* di mana LoRa menyebarkan sinyal yang lebih luas daripada luasan yang diperlukan untuk transmisi data normal.



**Gambar 2. 3** Paket Data LoRa

Berdasarkan gambar 2.3, LoRa memiliki beberapa parameter kunci di dalam pengiriman paket datanya, yaitu:

a. *Preamble*

*Preamble* adalah pembukaan yang digunakan agar *receiver* siap dalam menerima data. Secara *default*, *preamble* terdiri dari 12 simbol. Akan tetapi, jumlah simbol *preamble* dapat di program dan ditentukan oleh *user* (mulai dari 6-65,536) akan tetapi nilai sebenarnya yang terjadi di lapangan adalah ditambahkan dengan 4, sehingga *range* simbol *preamble* yang dapat digunakan adalah 10-65,540 [17].

b. *Header*

Terdapat dua mode utama dalam proses transmisi LoRa, yaitu mode eksplisit dan mode implisit. Mode eksplisit adalah ketika LoRa mengirimkan tiga pesan utama, yakni:

- I. *Payload Length* (berapa banyak byte data yang dikirim)
- II. *Forward Error Correction* (FEC) (sebagai perlindungan terhadap data agar data dapat diterima *receiver* tanpa terjadi kehilangan data atau *error*)
- III. *CRC* (16-bit opsional untuk memeriksa *error*)

Sedangkan mode implisit membuat konfigurasi di awal sehingga tidak perlu mengirimkan *payload length*, FEC, dan CRC setiap pengiriman data dilakukan. Dengan demikian, mode implisit dapat mengirimkan data lebih cepat dibanding mode eksplisit yang mengharuskan mengirimkan *header* setiap pengiriman data dilakukan [17].

c. *Spreading factor* (SF)

Ketika *spread spectrum* dilakukan, setiap bit data akan dipecah menjadi 64-4,096 bagian atau simbol agar data terhindar dari *noise*. SF adalah seberapa banyak simbol yang mewakili satu bit data.

**Tabel 2. 3** *Spreading factor* [17]

<i>Spreading factor</i>	<b>Simbol</b>	<b>SNR</b>
6	64	-5dB
7	128	-7,5dB
8	256	-10dB
9	512	-12,5dB
10	1,024	-15dB
11	2,048	-17,5dB
12	4,096	-20dB

Berdasarkan tabel 2.2, ditunjukkan ketika SF12 paket data dapat diterima di -20dB, itu menunjukan bahwa semakin besar *spreading factor* maka jumlah simbol yang mewakili satu bit data akan semakin besar dan ini membuat *receiver* menjadi lebih sensitif dan lebih jelas dalam menerima sinyal [17].

d. *Coding rate*

Saat transmisi data dilakukan, LoRa dapat mengirimkan data lain yang berfungsi sebagai penjagaan data agar data yang dikirim dengan baik, sehingga saat terjadi kesalahan atau ada data yang hilang saat proses transmisi terjadi, maka data tersebut dapat diperbaiki sebelum sampai ke *receiver*.

**Tabel 2. 4** *Coding Rate* [17]

<i>Encoding rate</i>	<i>Cyclic coding rate</i>	<i>Cost ratio</i>
1	4/5	1,25
2	4/6	1,5
3	4/7	1,75
4	4/8	2

Berdasarkan tabel 2.3, ditunjukkan bahwa saat *encoding rate* bernilai tinggi, maka *cyclic coding rate* juga akan tinggi, itu berarti penjagaan atas data yang dikirim lebih tinggi. Sebagai contoh, *encoding rate* bernilai 4 memiliki *cyclic coding rate* 4/8 yang berarti 4 bit data untuk 8 bit total, akan tetapi *cost ratio* juga akan mengalami peningkatan sehingga waktu transmisi akan lebih lama [17].

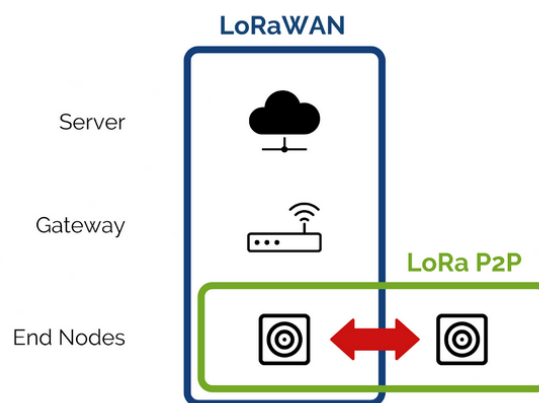
e. *Bandwidth*

*Bandwidth* adalah jarak antara frekuensi tertinggi dan frekuensi terendah. Maka, jika jarak frekuensi lebih besar maka *bandwidth* akan semakin lebar. Selain itu, *bandwidth* berbanding lurus dengan nominal bit rate (bps) yang dapat dikirimkan dan konsumsi daya yang diperlukan [17].

f. LoRa Modem

Modulasi dan demodulasi LoRa bergantung pada tiga parameter, yaitu SF, *bandwidth*, dan *coding rate*, yang menentukan kecepatan transmisi data di dalam LoRa [17].

Pita frekuensi yang digunakan dalam komunikasi LoRa pada penelitian ini adalah dengan *range* 433-915MHz sesuai dengan standar pita frekuensi di Asia [17].



**Gambar 2. 4** Arsitektur LoRa *Point-to-Point* [18]

Secara umum, LoRa memiliki tiga komponen utama, yaitu *End Devices*, *Gateway*, dan *Network Server*. Namun, dalam penelitian ini, arsitektur LoRa yang dipakai adalah *point-to-point*, yang memiliki arsitektur sebagai berikut:



### 1. Modul *Transmitter* (LoRa)

Modul ini digunakan untuk mengambil data dari sensor. Setelah itu, modul ini akan mentransmisikan data ke modul *receiver*.

### 2. Modul *Receiver* (LoRa)

Modul ini digunakan untuk menerima data dari modul *transmitter* dan meneruskannya ke server melalui jaringan internet atau lokal.

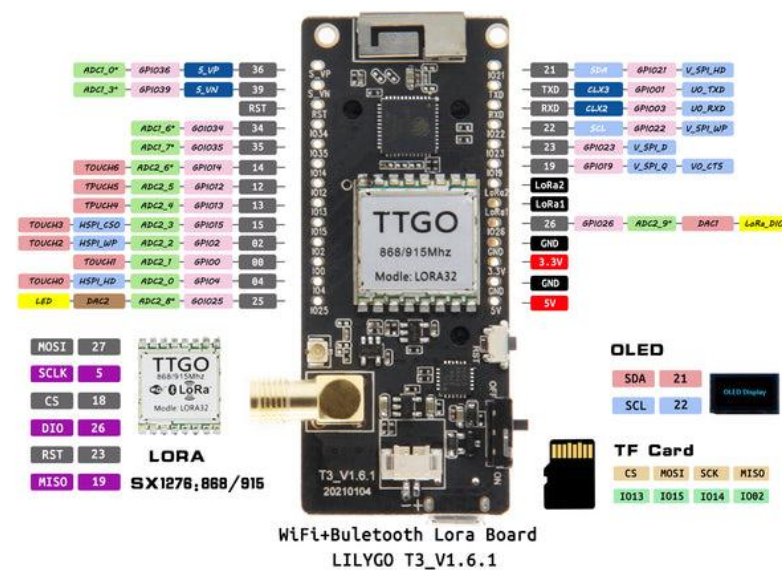
### 3. Server

Server digunakan untuk menerima data dari modul *receiver*. Kemudian, server menyimpan dan memproses data tersebut.

Dalam penelitian ini, jenis komunikasi LoRa yang digunakan adalah *point-to-point* yang berguna dalam proses transmisi data cuaca dari modul *transmitter* ke modul *receiver* dengan jarak yang jauh.

## 2.4 LoRa32 V2.1\_1.6

LoRa32 V2.1\_1.6 adalah modul MCU berbasis ESP32 dan *chip* LoRa SX1276 yang memungkinkan berkomunikasi pada frekuensi 915MHz. Dengan adanya komunikasi Wi-Fi, Bluetooth, dan LoRa, maka modul ini cocok digunakan untuk kebutuhan IoT yang membutuhkan komunikasi jarak jauh dengan konsumsi daya yang rendah, kebutuhan pengiriman data secara cepat, atau kebutuhan untuk terkoneksi dengan internet [19].



Gambar 2. 5 Pinout LoRa32 V2.1\_1.6 [19]

*Chip* LoRa SX1276 bekerja pada tegangan 1,8V hingga 3,7V. Di samping itu, modul ini memiliki layar OLED 0.96 inci, *slot Micro* SD, konektor SMA untuk antena eksternal dan mendukung mode modulasi LoRa, dengan sensitivitas penerimaan hingga -148dBm dan kecepatan *transceive* 300kbps. Selain itu, konsumsi arusnya bervariasi bergantung pada daya transmisi, contohnya 120mA pada +20dBm.

LoRa32 V2.1\_1.6 memiliki spesifikasi yang sangat cukup dalam pembuatan stasiun cuaca. Adapun spesifikasi lengkapnya adalah sebagai berikut:

**Tabel 2. 5** Spesifikasi LoRa32 V2.1\_1.6 [19]

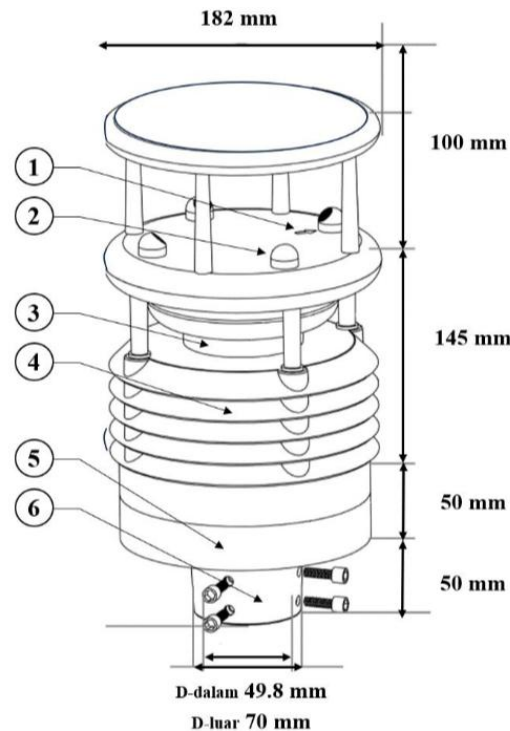
Spesifikasi	LoRa32 V2.1_1.6
SoC	ESP32 <i>Dual-Core</i> (Xtensa LX6)
Memori	520KB SRAM, 4MB <i>Flash</i>
CPU	2 <i>Core</i> , 32-bit
<i>Chip</i> LoRa	SX1276
<i>Clock Speed</i>	240MHz
Frekuensi	915MHz
<i>Chip</i> Serial	CH9102
GPIO	19 pin GPIO
Protokol Nirkabel	Wi-Fi + Bluetooth 4.2
Kartu SD	<i>Micro</i> SD
Antarmuka Digital	SPI, I2C, UART, PWM
Sensitivitas <i>Receiver</i>	-148dBm
<i>Rate</i> Data	0.3kbps hingga 50kbps
Jangkauan	1km tanpa halangan
Fungsi <i>Onboard</i>	<i>Reset \ Power switch</i>
Antena	3D Wi-Fi Antena (mendukung antena eksternal)
<i>Power Supply</i>	3.3V - 5V ( <i>Micro</i> -USB atau pin VIN) JST GH 2 pin 1.25mm
Dimensi	58mm x 25mm

Modul ini memakai SPI untuk komunikasi dengan *chip* LoRa SX1276. SPI digunakan karena memiliki latensi yang lebih rendah daripada UART atau I2C dan memiliki efisiensi dalam komunikasi perangkat, serta mendukung *multiple slave* dengan konfigurasi bus sederhana [20]. Selain itu, untuk berkomunikasi dengan sensor cuaca HONDE HD-WSM-U-07, LoRa32 V2.1\_1.6 menggunakan *port* UART melalui konverter UART TTL ke RS-485 yang menggunakan protokol RS-485 untuk membaca data cuaca. UART adalah protokol komunikasi serial yang bekerja secara asinkron dan tanpa memerlukan sinyal *clock* eksternal dengan menggunakan dua jalur utama yaitu TX untuk transmisi dan RX untuk menerima data [21]. Selain itu, modul juga telah dilengkapi dengan *slot micro-SD* dengan kapasitas hingga 32GB agar seluruh proses komunikasi data dan aktivitas *logging* sistem tetap berjalan tanpa adanya interupsi atau gangguan. Untuk tampilan, OLED 0,96 inci beroperasi dengan I2C. I2C adalah protokol komunikasi serial sinkron yang menggunakan dua jalur utama yaitu, SDA dan SCL. I2C digunakan karena OLED hanya membutuhkan kecepatan data menengah yaitu 3,4Mbps dengan *wiring* sederhana [22]. Selain hal tersebut, *port* GPIO digital digunakan untuk menghubungkan MCU dengan *push button* dan DIP Switch.

Pada penelitian ini, LoRa32V2.1\_1.6 digunakan sebagai MCU utama yang bertugas dalam mengelola komunikasi LoRa antara modul *transmitter* dan *receiver* di dalam sistem stasiun cuaca berbasis LoRa.

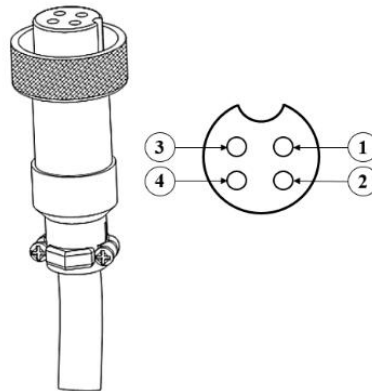
## 2.5 Sensor Cuaca HONDE HD-WSM-U-07

HONDE HD-WSM-U-07 adalah sensor cuaca yang dikembangkan oleh Honde Technology Co. Ltd untuk pengambilan data cuaca secara *real-time* dengan daya operasi 12-24V. Data cuaca yang di ambil oleh sensor dengan ditransmisikan melalui protokol komunikasi Modbus melalui RS-485 dengan *baud rate* 9600, 8 bits, 1 stop bit, dan tanpa *parity check*. Selain itu, protokol RS-485 mencakup spesifikasi komunikasi, deskripsi CRC (*Cyclic Redundancy Check*) untuk verifikasi data, dan deskripsi register Modbus [8].



**Gambar 2. 6** Bagian-Bagian Sensor Cuaca HONDE HD-WSM-U-07 [8]

Secara fisik, sensor ini memiliki konstruksi yang dapat dilihat pada gambar 2.6. Pada gambar 2.6, dapat dilihat jika bagian ke-1 adalah gambar panah yang dicetak timbul pada badang sensor. Fungsinya adalah sebagai petunjuk sisi utara dari sensor. Ketika instalasi, sensor dapat dihadapkan ke arah utara sesuai dengan arah panah. Bagian ke-2 adalah 4 sensor ultrasonik yang berfungsi melakukan *sensing* sebagian variabel pengukuran. Bagian ke-3 adalah *control unit* atau pusat proses yang berupa modul elektronik dengan fungsi pusat pemrosesan sinyal dan data digital. Bagian ke-4 adalah *shutter box* atau wadah dengan material plastik ABS yang digunakan untuk membantu sensor dalam melakukan pengukuran beberapa variabel seperti temperatur, kelembaban, dan tekanan udara. Bagian ke-5 adalah bagian sensor yang digunakan untuk mengurangi *noise* dalam proses pembacaan variabel ukur akibat faktor eksternal. Bagian ke-6 adalah *mechanic support* atau *plat* besi berbentuk lingkaran yang difungsikan untuk membantu proses instalasi mekanik dari sensor dengan tiang (opsional).



**Gambar 2. 7** Konfigurasi Kabel Komunikasi HONDE HD-WSM-U-07 [8]

Sensor Honde -HD-WSM-U-07 memiliki 4 pin *male* seperti yang ditunjukkan pada gambar 2.7. Pin ke-1 merupakan jalur suplai daya positif, pin ke-2 merupakan jalur suplai daya negatif, pin ke-3 memiliki fungsi penyalur data menggunakan protokol komunikasi RS-485 A, dan pin ke-4 sebagai penyalur RS-485 B. Dalam penelitian ini, suplai daya ke sensor berasal dari SMPS 24V-5A, dan pin RS-485 A dan B terhubung dengan UART TTL *to* RS-485 *converter* agar dapat terhubung dengan MCU.

**Tabel 2. 6** Spesifikasi Parameter Sensor Cuaca HONDE HD-WSM-U-07 [8]

Parameter	Range	Resolusi	Akurasi
Suhu	-40-60°C	0,01°C	±0,3°C (25°C)
Kelembapan Udara	0-100%RH	0,01%RH	±3%RH
Tekanan Atmosfer	300-1100hpa	0,1hpa	±0,5hpa (0-30°C)
Kecepatan Angin	0-60m/s	0,01m/s	(0-30m/s) ±0,3m/s atau ±3% (30-60m/s) ±5%
Arah Angin	0-360°	0,1°	±2°
Curah Hujan	0-200mm/h	0,1mm	Error ±10%
Iradiasi Matahari	0-2000W/m <sup>2</sup>	1W	5%

Berdasarkan tabel 2.5, sensor cuaca Honde HONDE HD-WSM-U-07 memiliki kualitas pengambilan parameter cuaca yang cukup baik. Selain itu, komunikasi yang digunakan untuk mengakses data sensor adalah RS-485 dengan Modbus RTU. Adapun alamat dari masing-masing parameter bisa dilihat pada tabel 2.6.

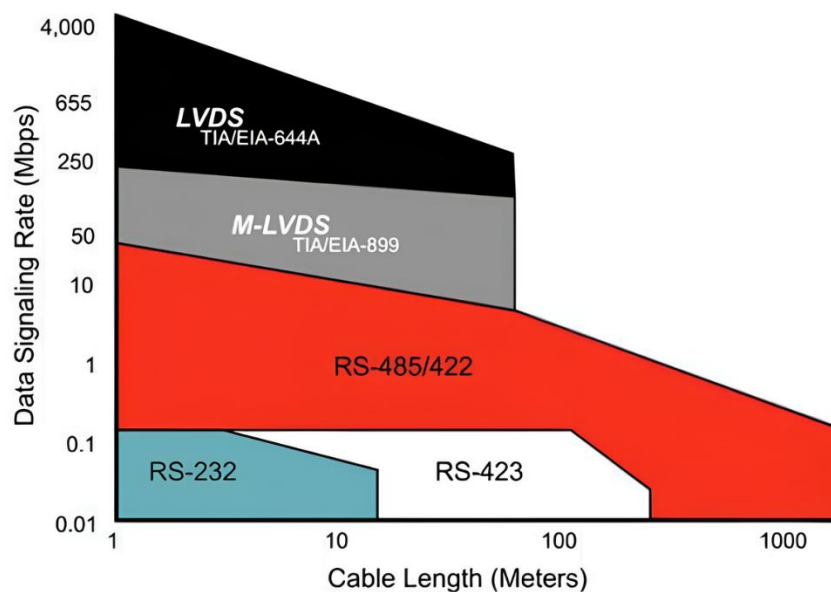
**Tabel 2. 7** Alamat Komunikasi RS-485 [8]

<i>Address</i>	<i>Operasi</i>	<i>Isi</i>
0x0009	<i>Read only</i>	Suhu. Dihitung dengan mengonversi bilangan heksadesimal ke desimal, membaginya dengan 100, lalu mengurangnya dengan 40. Contohnya, 0x1B00 adalah $(6,912/100) - 40 = 29,12^{\circ}\text{C}$ .
0x000A	<i>Read Only</i>	Kelembapan udara. Kelembapan dikali 100 lalu dikonversi ke bilangan heksadesimal. Contohnya, 56,35%RH berarti $56,35 \times 100 = 5635$ (1603 heksadesimal) maka ditulis 0x1605.
0x000B	<i>Read Only</i>	Tekanan atmosfer. Bilangan dikali 10 dan diubah ke dalam heksadesimal. Contohnya, 0x2784 berarti $10116/10 = 1011\text{hpa}$
0x000C	<i>Read Only</i>	Kecepatan angin. Kecepatan angin diperoleh dengan mengonversi bilangan heksadesimal ke desimal, membaginya dengan 100. Contohnya, 0x0125 adalah $(293/100) - 40 = 2,93\text{m/s}$ .
0x000D	<i>Read Only</i>	Arah angin. Arah angin diperoleh dengan mengonversi bilangan heksadesimal ke desimal, membaginya dengan 100. Contohnya, 0x0C14 adalah $(3092/100) - 40 = 309,2^{\circ}$ .
0x000E	<i>Read Only</i>	Curah hujan 10 menit. Curah hujan diperoleh dengan memperbesar bilangan heksadesimal sepuluh kali. Contohnya, 0x0016 berarti 2,2mm.
0x000F	<i>Read Only</i>	Iradiasi. Kecepatan angin diperoleh dengan mengonversi bilangan heksadesimal ke desimal. Contohnya, 0x0172 adalah $370\text{W/m}^2$ .
0x0000	<i>Read Write</i>	Address sensor, jangkauan (0x01 sampai 0xFF) dengan setelan pabrik 0xFF.

Berdasarkan tabel 2.6, bisa dilihat bahwa parameter data sensor diakses melalui *function code 03 (read holding register)*, sedangkan untuk alamat sensor bisa diakses dengan *function code 06 (write single register)*.

Dalam penelitian ini, data sensor dikirim ke MCU melalui komunikasi Modbus RTU melalui RS-485 dan dihubungkan ke UART TTL to RS-485 *converter* agar bisa diakses oleh MCU.

## 2.6 Protokol RS-485



**Gambar 2. 8** Perbandingan RS-485 dengan Komunikasi Lain [23]

Protokol RS-485 adalah komunikasi yang mengirim data berdasarkan perbedaan tegangan antara dua kabel. Selain itu, protokol ini mendukung banyak perangkat pada satu bus atau hanya dua kabel *twisted pair*, menjadikannya ideal untuk aplikasi di lingkungan dengan banyak noise. RS-485 ini beroperasi dengan sinyal *differential* untuk meningkatkan ketahanan *noise* dan laju transmisi data yang lebih tinggi dibanding alternatif lain seperti RS-232. Berdasarkan gambar 2.8, RS-485 memiliki jarak panjang kabel yang lebih unggul dari pada komunikasi lain dengan pengiriman sinyal antara 0,1-10Mbps hingga 50Mbps dengan jarak yang pendek. Umumnya, RS-485 menggunakan protokol Modbus untuk komunikasi data [24]. Spesifikasi lengkapnya adalah sebagai berikut:

**Tabel 2. 8** Spesifikasi RS-485 [25]

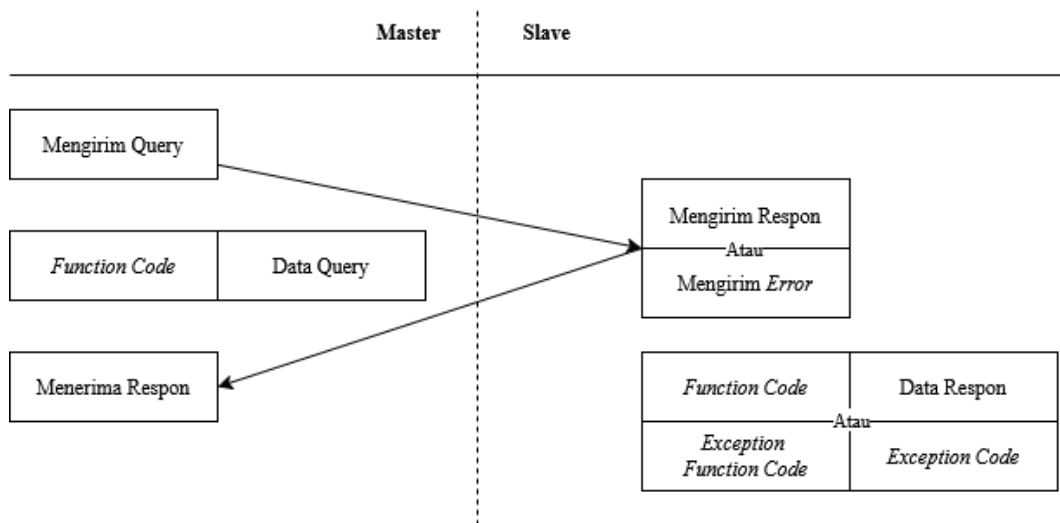
<b>Spesifikasi</b>	<b>RS-485</b>
Mode Operasi	<i>Half-duplex, multidrop</i>
Panjang Kabel Maksimal	1,200 meter (4,000 kaki)
Jumlah <i>Node</i> Maksimal	32 (dapat ditingkatkan dengan <i>repeater</i> )
Tegangan Operasi	-7V hingga +12V
Impedansi Karakteristik	120 ohm
<i>Baud Rate</i>	Hingga 10 Mbps (bervariasi tergantung jarak)
Topologi Jaringan	Bus
<i>Parity Check</i>	<i>None, Even, Odd</i>
<i>Stop Bits</i>	1 atau 2
<i>Data Bits</i>	7 atau 8

Dalam penelitian ini, RS-485 digunakan sebagai media transmisi data sensor Honde HONDE HD-WSM-U-07 dengan protokol komunikasi Modbus RTU. Jalur RS-485 A dan B dihubungkan dengan dihubungkan ke UART TTL to RS-485 *converter* agar bisa diakses oleh MCU.

## 2.7 Modbus

Modbus adalah protokol komunikasi yang dapat beroperasi dengan koneksi serial (seperti RS-232 atau RS-485) dan koneksi ethernet (TCP/IP) yang memungkinkan komunikasi jarak jauh dan integrasi mudah dengan MCU. Modbus bekerja dengan prinsip *master-slave*, dimana perangkat *master* akan mengirimkan permintaan (*query*) kepada perangkat *slave*, kemudian perangkat *slave* akan merespons dengan data yang diminta oleh perangkat *master* [26].





**Gambar 2. 9** Alur Data Modbus [26]

Gambar 2.9 menjabarkan alur data di dalam modbus. Pertama, *master* mengirim *query* yang berisi *function code* dan data ke *slave*. Kemudian, jika data *query* yang dikirimkan *master* adalah benar, maka *slave* akan mengirimkan pesan respon ke *master* berupa *function code* dan data respons. Namun, apabila *query* yang dikirimkan *master* terdapat kesalahan, maka *slave* akan memberikan pesan *error* yang terdiri dari *exception function code* dan *exception code* ke *master* [24].

Terdapat 3 kemungkinan kondisi komunikasi yang dapat terjadi antara *master* dan *slave*, yaitu:

1. Jika *slave* menerima pesan *query* dari *master* tanpa adanya kesalahan, maka *slave* akan mengirimkan pesan respons dengan normal.
2. Jika *slave* tidak menerima pesan *query* apapun dari *master* karena kesalahan komunikasi atau pesan *query* dapat diterima oleh *slave*, akan tetapi kesalahan komunikasi (*parity*, LRC, CRC) terdeteksi, maka *slave* tidak akan memberikan respons apapun dan *master* akan memberikan kondisi *time-out* untuk pesan *query* tersebut.
3. Jika *slave* menerima pesan *query* tanpa adanya kesalahan komunikasi, akan tetapi *slave* tidak bisa menangani perintah pesan *query* tersebut, maka *slave* akan mengirimkan *exception response* untuk menginformasikan letak kesalahan yang terjadi.

*Function Code* adalah kode yang mengisyaratkan perintah *master* yang harus dilakukan oleh *slave*. Adapun kode fungsi modbus adalah sebagai berikut:

**Tabel 2. 9** *Function Code* Modbus [26]

Kode Fungsi	Fungsi
1 = 01H	<i>Read Coil Status</i>
2 = 02H	<i>Read Discrete Input</i>
3 = 03H	<i>Read Holding Registers</i>
4 = 04H	<i>Read Input Registers</i>
5 = 05H	<i>Force Single Coil</i>
6 = 06H	<i>Preset Single Register</i>
15 = 0FH	<i>Force Multiple Coils</i>
16 = 10H	<i>Preset Multiple Registers</i>

*Response exception* merupakan respons dari *slave* ketika *slave* tidak bisa menjalankan operasi yang diminta oleh *master*. Hal tersebut bisa terjadi karena *function code* tidak sesuai, nilai data tidak valid, atau alamat yang diminta berada di luar rentang yang disediakan *slave*.

Dalam penelitian ini, Modbus digunakan sebagai protokol komunikasi dengan media transmisi RS-485 untuk sensor Honde HONDE HD-WSM-U-07. Data yang dikirimkan oleh Modbus akan diproses oleh MCU dan dikirimkan ke modul *receiver* melalui komunikasi LoRa.

## 2.8 Modbus RTU

Protocol Data Unit					
Start	Address (Slave ID)	Function Code	Data	CRC Error Check	End
3,5 bytes	1 byte	1 byte	n bytes	2 bytes	3,5 bytes
Application Data Unit					

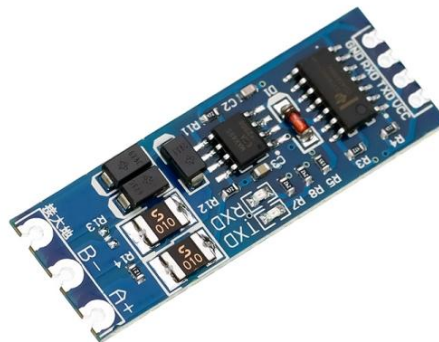
**Gambar 2. 10** Format *Frame* Modbus RTU [26]

Gambar 2.10 menunjukkan format *frame* Modbus RTU, dengan keterangan adalah sebagai berikut:

- a. *Start* menandakan awal dari *frame* baru.
- b. *Address* adalah alamat dari *slave* yang dituju. Memiliki rentang dari 1-247. Sedangkan, alamat 0 digunakan untuk fungsi *broadcast*.
- c. *Function Code* adalah perintah dari *master* yang harus dilaksanakan oleh *slave*.
- d. Data berisi alamat register, data yang akan ditulis, dan jumlah data.
- e. *CRC Error Check* berguna untuk mendeteksi kerusakan pada data yang dikirim.
- f. *End* menandakan akhir dari *frame* data.

Pada penelitian ini, Modbus RTU digunakan sebagai protokol komunikasi untuk transmisi data dari sensor Honde HONDE HD-WSM-U-07 melalui RS-485 ke MCU dengan UART TTL to RS-485 converter. Setelah itu, data akan diproses oleh MCU untuk kemudian diubah formatnya menjadi JSON dan dikirimkan ke modul *receiver* melalui komunikasi LoRa.

## 2.9 UART TTL to RS-485 Converter



**Gambar 2. 11** UART TTL to RS-485 Converter [27]

UART TTL to RS-485 Converter adalah perangkat yang digunakan untuk mengonversi sinyal UART TTL (*Transistor-Transistor Logic*) yang beroperasi pada tegangan rendah yaitu 3.3V atau 5V menjadi sinyal RS-485 yang menggunakan sinyal diferensial sehingga tahan terhadap noise dan memungkinkan

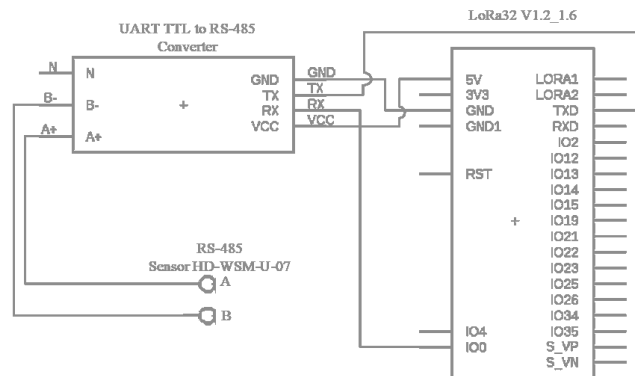
transmisi data hingga 1200 meter. Dengan demikian, perangkat yang menggunakan sinyal TTL seperti MCU dapat berkomunikasi dengan perangkat lain yang menggunakan komunikasi RS-485 yang memungkinkan integrasi dengan jangkauan yang lebih besar dan kompleks [28].

Cara kerja UART TTL to RS-485 Converter ini adalah dengan mengubah level tegangan TTL menjadi perbedaan tegangan antara dua jalur (A dan B) yang digunakan dalam RS-485. Berikut adalah spesifikasi dasar dari UART TTL to RS-485 Converter:

**Tabel 2. 10** Spesifikasi UART TTL to RS-485 Converter [29]

Spesifikasi	UART TTL to RS-485 Converter
Tipe Konversi	UART TTL ke RS-485
Level Tegangan TTL	3,3V atau 5V
Jarak Transmisi RS-485	800 meter
Mode Operasi	<i>Half-duplex</i>

Dalam penelitian ini, UART TTL to RS-485 Converter digunakan sebagai penghubung antara sensor HONDE HD-WSM-U-07 dengan komunikasi RS-485 agar bisa berkomunikasi dengan MCU yang menggunakan komunikasi UART sehingga data dapat dibaca dan diproses oleh MCU. Pin A+ UART TTL to RS-485 Converter dihubungkan dengan pin A RS-485 dari sensor HONDE HD-WSM-U-07 dan pin B- UART TTL to RS-485 Converter dihubungkan dengan pin A RS-485 dari sensor HONDE HD-WSM-U-07. Sedangkan pin TX UART TTL to RS-485 Converter dihubungkan dengan pin TX LoRa32 V2.1\_1.6 dan pin RX UART TTL to RS-485 Converter dihubungkan dengan pin GPIO0 LoRa32 V2.1\_1.6 yang digunakan sebagai RX. Wiring lengkap UART TTL to RS-485 Converter bisa dilihat pada gambar 2.12.



**Gambar 2. 12** Wiring UART TTL to RS-485 Converter

## 2.10 Modul Mini W5500 SPI to LAN



**Gambar 2. 13** Modul Mini W5500 SPI to LAN [30]

Modul Mini W5500 SPI to LAN adalah modul ethernet yang memungkinkan konketivitas dengan MCU melalui antarmuka SPI (*Serial Peripheral Interface*). Selain itu, modul ini menggunakan *chip* W5500 yang memiliki tumpukan protokol TCP/IP terintegrasi yang berguna untuk implementasi komunikasi jaringan tanpa kode tingkat rendah. Di lain sisi, modul ini dapat diterapkan pada aplikasi jaringan yang kompleks dengan mudah karena didukung oleh beberapa protokol, seperti TCP, UDP, IGMP, ARP, IPv4, PPPoE, dan ICMP [31].

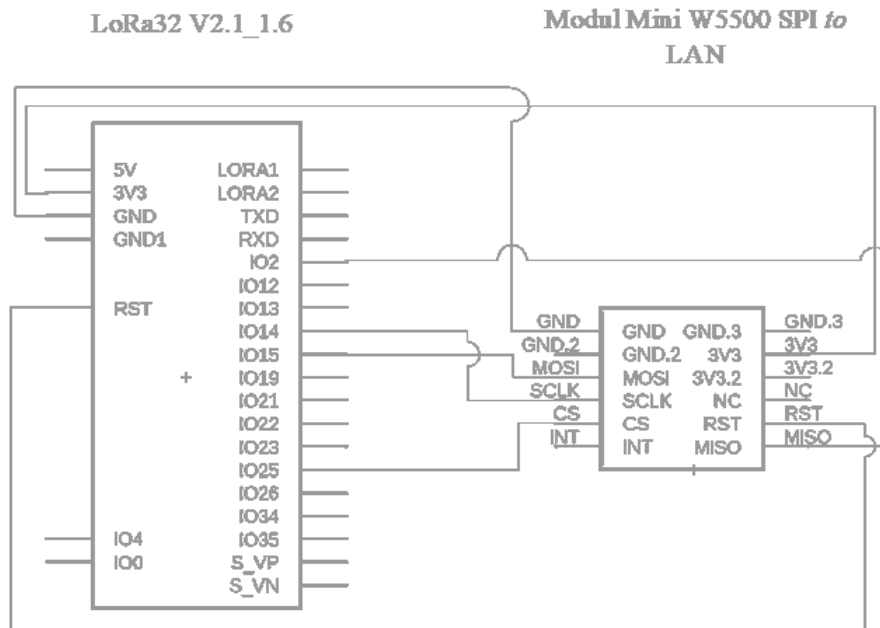
Modul W5500 beroperasi dengan antarmuka SPI yang memiliki kecepatan komunikasi sebesar 80MHz. Selain hal tersebut, modul ini memungkinkan pengguna untuk memakai 8 koneksi soket secara bersamaan dan *cache* internal

sebesar 32KB untuk pengiriman serta penerimaan data. Berikut adalah spesifikasi Modul Mini W5500 SPI to LAN:

**Tabel 2. 11** Spesifikasi Modul Mini W5500 SPI to LAN [31]

Spesifikasi	Modul Mini W5500 SPI to LAN
<i>Chip</i>	W5500
Antarmuka	SPI ( <i>Serial Peripheral Interface</i> )
Protokol	TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
Kecepatan SPI	Hingga 80MHz
Jumlah Koneksi <i>Socket</i>	Hingga 8 koneksi
<i>Cache</i> Internal	32KB
Tegangan Operasi	3,3V atau 5V
Ukuran Modul	28,5mm x 23mm x 24mm

Modul Mini W5500 SPI to LAN memungkinkan LoRa32 V2.1\_1.6 melakukan koneksi jaringan tanpa perlu mengimplementasikan TCP/IP secara manual ke dalam MCU. W5500 akan menerima perintah dari MCU melalui SPI yang kemudian meneruskan data ke jaringan LAN melalui konektor RJ45 Ethernet. Saat dalam mode DHCP, W5500 akan meminta alamat IP dari *router*, sedangkan dalam mode statis, IP ditentukan oleh pengguna secara manual. Komunikasi SPI menggunakan konsep *master-slave*, di mana MCU sebagai master. Berdasarkan gambar 2.14 , pin SCLK Modul Mini W5500 SPI to LAN terhubung dengan GPIO14 LoRa32 V2.1\_1.6 sebagai pengatur kecepatan transfer data, pin MISO Modul Mini W5500 SPI to LAN terhubung dengan GPIO2 LoRa32 V2.1\_1.6 sebagai jalur *slave* mengirim data ke *master*, pin MOSI Modul Mini W5500 SPI to LAN terhubung dengan GPIO15 LoRa32 V2.1\_1.6 sebagai jalur *master* mengirimkan data ke *slave*, pin CS Modul Mini W5500 SPI to LAN terhubung dengan GPIO25 LoRa32 V2.1\_1.6 sebagai pengatur yang memastikan hanya satu perangkat yang aktif dalam satu waktu guna menghindari konflik dalam bus SPI jika digunakan oleh beberapa perangkat.



**Gambar 2. 14** Wiring Modul Mini W5500 SPI to LAN

Dalam penelitian ini, Modul Mini W5500 SPI to LAN digunakan sebagai komponen yang mendukung agar MCU dapat mengirimkan data sensor ke *database*. Perangkat ini dipakai dengan tujuan menjadi opsi apabila di tempat instalasi stasiun cuaca tidak terdapat fasilitas Wi-Fi, sehingga stasiun cuaca tetap dapat menyimpan data ke server atau *database* melalui ethernet/LAN.

## 2.11 SMPS 24V-5A



**Gambar 2. 15** SMPS 24V-5A [32]

SMPS atau *Switch Mode Power Supply* adalah perangkat yang berfungsi sebagai pengubah input AC menjadi output DC. Prinsip kerja SMPS adalah dengan menyearahkan tegangan AC yang masuk dengan *bridge diode*, lalu *noise* dihilangkan menggunakan filter dan diubah menjadi sinyal DC yang stabil. Selain itu, konversi energi yang efisien dapat dihasilkan karena proses *switching* yang dilakukan dengan frekuensi tinggi [33].

**Tabel 2. 12** Spesifikasi SMPS 24V-5A [33]

Spesifikasi	SMPS 24V-1A
Tegangan Masukan	110V - 220V AC 50/60Hz
Tegangan Keluaran	DC 24V
Arus Keluaran	5A
Efisiensi	85%
Daya Maksimum	150W
Dimensi	20 x 11 x 5 cm

Berdasarkan gambar 2.16, tegangan 220VAC yang terdiri dari 220VAC (L), nertal (N), dan *ground* merupakan input SMPS 24V-5A, lalu SMPS 24V-5A memiliki 4 pin output yang berupa 2 pin 24VDC+ dan 2 pin 24VDC-.

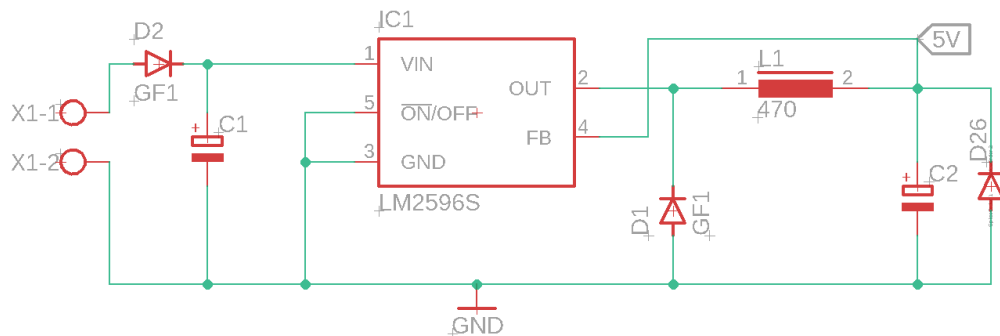


**Gambar 2. 16** Pinout SMPS 24V-5A [34]



Dalam penelitian ini, SMPS 24V-5A digunakan sebagai penyalur daya antara catu daya dari PLN sebesar 220VAC ke 24VDC sebagai suplai daya bagi modul komunikasi dan sensor HONDE HD-WSM-U-07.

### 2.12 Rangkaian Buck Converter 5V LM2596



**Gambar 2. 17** Rangkaian Buck Converter 5V LM2596

Rangkaian Buck Converter adalah rangkaian penurun tegangan DC ke DC yang mengonversi tegangan input yang lebih tinggi ke tegangan yang lebih rendah. Berdasarkan gambar 2.17, prinsip kerja modul ini adalah dengan memanfaatkan *switching* cepat menggunakan transistor yang diintegrasikan di dalam IC LM2596, dioda, induktor, dan kapasitor. Di saat transistor aktif, energi tersimpan ke dalam induktor. Sedangkan ketika transistor dalam keadaan non-aktif, maka induktor melepaskan energi melalui dioda ke beban. Selain itu, rangkaian induktor-kapasitor berfungsi sebagai filter riak tegangan, sehingga tegangan yang dihasilkan menjadi stabil. Sedangkan, dioda zener digunakan untuk menjaga nilai tegangan output tidak lebih dari 5V sehingga beban dapat terlindungi [35].

**Tabel 2. 13** Spesifikasi Rangkaian Buck Converter 5V LM2596 [36]

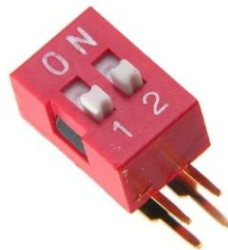
Spesifikasi	Rangkaian Buck Converter 5V LM2596
Tegangan Input	7V-24V
Ripple Tegangan Output	<50mV (pada beban penuh 3A)
Suhu Operasi	-40°C hingga +125°C
Efisiensi Maksimal	92% (pada $V_{in}=12V$ , $I_{out}=2A$ )
Dioda	Diode SS510
Dioda Zener	Diode Zener 5.1V

Tabel 2. 14 Lanjutan

Spesifikasi	Rangkaian Buck Converter 5V LM2596
Induktor	Induktor 33uH SMD
Kapasitor	ElcoSMD 330uF 25V

Dalam penelitian ini, Buck Converter digunakan untuk menurunkan tegangan 24VDC dari SMPS 24V-5A menjadi 5VDC sehingga dapat sesuai dengan tegangan yang dibutuhkan untuk mengoperasikan LoRa32 V2.1\_1.6.

### 2.13 DIP Switch 2 pin



Gambar 2. 18 Dip Switch 2 pin [37]

DIP Switch (*Dual In-Line Package Switch*) adalah rangkaian sakelar yang berada dalam bentuk *chip IC (Integrated Circuit)* yang terdiri dari beberapa sakelar yang bisa di ubah kondisinya dengan cara manual. Biasanya, Dip Switch digunakan dalam beberapa perangkat keras untuk kebutuhan konfigurasi mode [37].

Dalam penelitian ini, DIP Switch digunakan sebagai antarmuka pengaturan mode modul komunikasi, yaitu *transmitter*, *receiver* menggunakan komunikasi Wi-Fi, dan *receiver* menggunakan komunikasi LAN. Konfigurasi dilakukan sebelum sistem mendapat suplai daya. Jika tidak memungkinkan untuk melakukan konfigurasi mode pada saat sistem dalam keadaan mati, maka mode tetap dapat diubah dengan cara mengubah konfigurasi DIP Switch lalu menekan tombol *reset*. Untuk melakukan konfigurasi mode, dapat dengan mengatur posisi DIP Switch Mode dengan fungsi dan kombinasi sebagai berikut:

### 2.14 Push button 4 Kaki

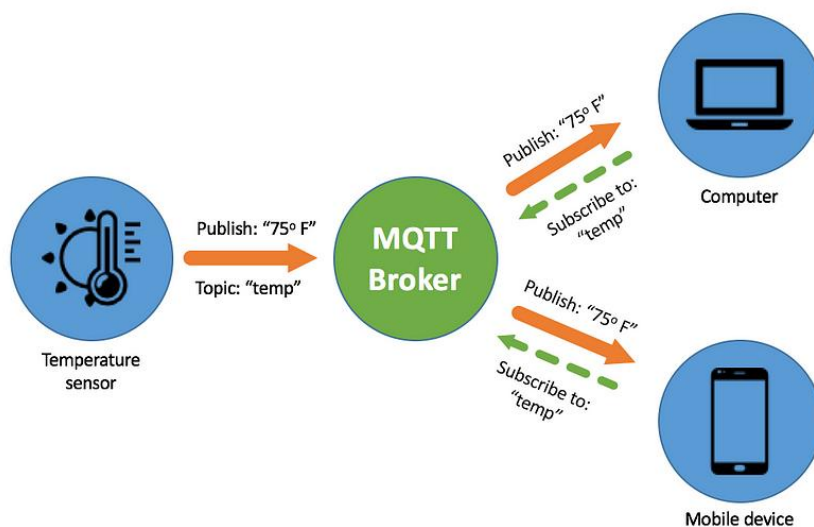


**Gambar 2. 19** Push button 4 Kaki

*Push button* merupakan perangkat yang digunakan untuk menghubungkan dan memutuskan arus listrik. Umumnya, *push button* memiliki logika *Normally Open* (NO) dan *Normally Close* (NC). Namun, *push button* yang digunakan dalam penelitian ini adalah *push button Normally Open*. Dengan demikian, ketika *push button* tidak ditekan maka arus listrik akan terputus, sedangkan ketika *push button* ditekan maka arus listrik akan terhubung [38].

Dalam penelitian ini, *push button* digunakan sebagai tombol untuk *select* dan *reset*. *Push button Select* berfungsi untuk mengoperasikan modul ketika tidak tersedia *Micro SD*, sedangkan tombol *reset* digunakan untuk konfigurasi mode saat sistem dalam keadaan menyala dengan mengatur posisi *DIP Switch* lalu menekan tombol *reset*.

### 2.15 MQTT



**Gambar 2. 20** Alur Kerja MQTT [39]

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi yang di rancang untuk aplikasi *machine-to-machine* dan *Internet of Things* (IoT) karena menggunakan model *publish/subscribe* yang memungkinkan perangkat dengan sumber daya terbatas dapat menggunakannya dengan ringan dan efisien.

Berdasarkan gambar 2.20, MQTT memiliki tiga komponen utama yang terdiri dari:

a. *Publisher*

*Publisher* adalah data dengan topik yang sudah ditentukan ke broker.

b. *Subscriber*

*Subscriber* adalah penerima data dengan topik tertentu.

c. *Broker*

*Broker* adalah penghubung dan pengelola proses *publish/subscribe*.

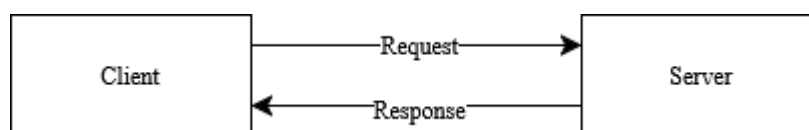
d. Topik

Topik adalah pengelompokan pesan yang digunakan untuk mengarahkan pesan antara *publisher* dan *subscriber* melalui *broker*.

Cara kerja MQTT adalah *publisher* mengirimkan data ke topik tertentu ke *broker*, lalu *subscriber* memilih topik mana yang akan di *subscribe*, setelah itu data akan diterima oleh *subscriber*. Dengan demikian, antara *publisher* dan *subscriber* tidak perlu mengetahui data pribadi masing-masing terlebih dahulu, akan tetapi hanya memilih topik yang sama. Sehingga, penggunaan MQTT dalam skala besar sangat di mungkinkan [39].

Dalam penelitian ini, MQTT digunakan sebagai protokol komunikasi transmisi data antara modul *receiver* dan server. Modul *receiver* berperan sebagai *publisher* dan server sebagai *subscriber*. Modul *receiver* akan terhubung dengan internet lalu *publish* data sensor dengan topik tertentu, lalu server melakukan *subscribe* topik yang sama sehingga data dapat diterima oleh server.

## 2.16 Server



**Gambar 2. 21** Alur Kerja *Client-Server*

Server adalah sistem komputer terspesialisasi yang di desain untuk mengatur, menyimpan, dan memproses data. Selain itu server juga dapat digunakan untuk mengatur komunikasi dan berbagi sumber daya melalui sebuah jaringan. Berdasarkan gambar 2.21, prinsip kerja server berbasis pada *client-server*. *Client* mengirimkan permintaan data atau layanan pada server, setelah itu server akan memproses permintaan dan meresponsnya dengan mencari data yang dibutuhkan. Selain itu, *client* dapat menyimpan data di server atau meminta izin akses data tertentu yang ada di server [40]. Spesifikasi PC Server yang digunakan dalam penelitian ini adalah sebagai berikut:

**Tabel 2. 15** Spesifikasi PC Server

Server	Spesifikasi
Nama Komputer	DESKTOP-DO3DUJ5
Sistem Manufaktur	ASRock
Model Sistem	B660M Pro RS
BIOS	12,03
<i>Processor</i>	12th Gen Intel(R) Core(TM) i5-12400F (12 CPUs), ~2,5GHz
Memori	16,384MB RAM

Dalam penelitian ini, PC digunakan sebagai server lokal. PC Server digunakan sebagai pengelola utama data yang dikirim oleh stasiun cuaca sampai data dapat ditampilkan di dalam *dashboard*, termasuk mengatur *end-point*, *database*, broker MQTT, *back-end*, dan *front-end*.

## 2.17 Linux

Linux adalah kernel sistem operasi *open-source* yang mampu mengelola sumber daya dan menjadi antarmuka antara perangkat keras dan perangkat lunak. Selain itu, Linux dapat mengerjakan penjadwalan proses, manajemen memori, dan *driver* perangkat, serta di dukung dengan desain modular yang memungkinkan kustomisasi dan fleksibilitas yang terdiri dari komponen seperti *bootloader*, sistem *init*, dan server grafis. Disisi lain, Linux dilengkapi kemampuan *multitasking* dan *multi-user*, memiliki struktur sistem *file* hierarkis, serta menggabungkan beberapa

fitur keamanan. Dengan demikian, Linux sangat mendukung aplikasi seperti sistem tertanam dan superkomputer, serta memungkinkan pengguna untuk memperbesar skala karena Linux memiliki kestabilan dan bisa diadopsi di beberapa platform dan aplikasi komputasi [41]. Linux juga memberikan dukungan untuk *networking*, seperti SSH, VPN, SCP, dan FTP.

Linux digunakan sebagai sistem operasi yang diterapkan di dalam PC Server karena Linux memiliki kestabilan di saat server harus selalu berjalan. Selain itu, Linux kompatibel dengan MySQL dan server MQTT karena memiliki dukungan bawaan.

## 2.18 MySQL

MySQL adalah salah satu sistem *Relational Database Management System* (RDBMS) yang mempunyai karakteristik ringan, mudah, dan andal. Selain itu, MySQL adalah sistem yang *open-source*, sehingga memungkinkan pengguna untuk membuat, mengelola, dan mengoperasikan basis data dengan menggunakan *Structured Query Language* (SQL). Selain hal tersebut, MySQL dapat melakukan *multi-reading*, *multi-user*, dan dapat di terapkan di berbagai sistem operasi seperti Windows, Linux, dan MacOS [42]. Secara umum, MySQL memiliki tiga komponen utama, yaitu:

a. *Data Definition Language* (DDL)

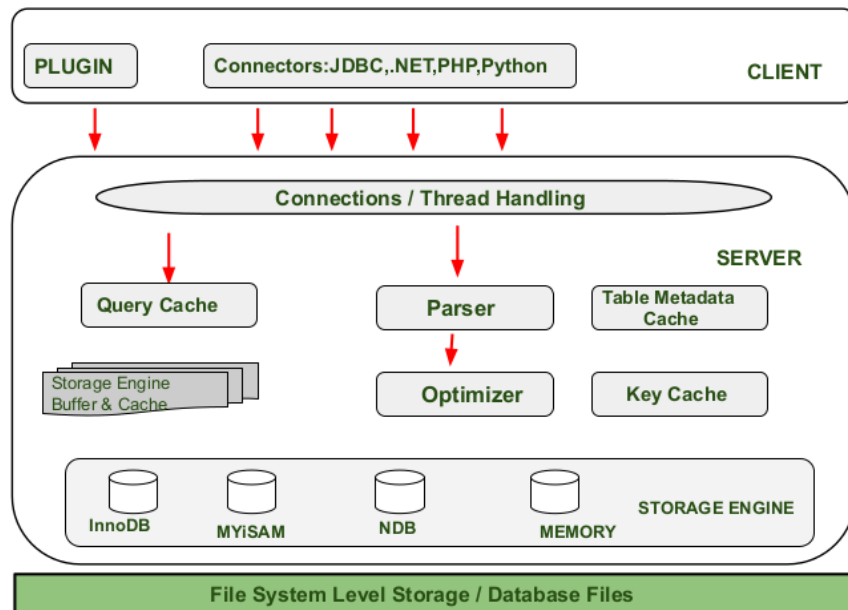
DDL digunakan untuk membuat dan menghapus tabel, termasuk mendefinisikan struktur basis data.

b. *Data Manipulation Language* (DML)

DML digunakan untuk mengelola data seperti memasukkan, mengganti, dan menghapus data.

c. *Data Control Language* (DCL)

DCL berguna untuk mengatur kontrol dan hak akses pada data di dalam basis data.



**Gambar 2. 22** Arsitektur MySQL[43]

Berdasarkan gambar 2.22, MySQL menggunakan arsitektur *client-server* yang terdiri dari tiga lapisan yaitu layer client, layer server, dan layer penyimpanan. Layer *client* memungkinkan pengguna untuk memberi perintah melalui *command prompt* atau GUI sehingga pengguna dapat mengatur koneksi, autentikasi, dan keamanan. Layer server mengelola *thread handling*, *parsing*, optimasi *query*, *caching*, dan metadata untuk memastikan perintah dijalankan dengan efisien. Sedangkan layer penyimpanan menyediakan berbagai *storage engine* yang memungkinkan fleksibilitas dalam penyimpanan data [43].

Dalam penelitian ini, MySQL digunakan sebagai sistem dalam pembuatan, pengelolaan, dan pengoperasian penyimpanan data yang dikirim oleh *receiver* ke server melalui MQTT. MySQL dipilih karena bisa menangani data besar dan terstruktur sehingga memungkinkan analisis historis oleh XGBoost. Selain itu, MySQL memiliki dukungan ACID (*Atomicity, Consistency, Isolation, Durability*) yang membuatnya dapat menyimpan data cuaca dengan aman tanpa korupsi.

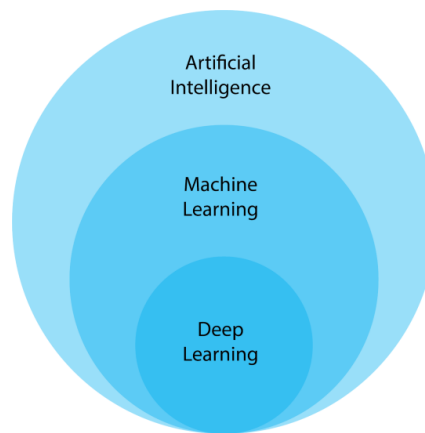
## 2.19 Flask

Flask adalah *framework* web mikro berbasis Python untuk membuat web dan API. Flask menggunakan *routing* sederhana yaitu setiap permintaan HTTP dapat

diarahkan menggunakan dekorator `@app.route()` ke fungsi tertentu. Selain itu, Flask didukung oleh beberapa ekstensi seperti Flask-SQLAlchemy untuk *database*, Flask-RESTful untuk API, dan Flask-CORS untuk komunikasi lintas domain. Flask juga mudah diintegrasikan dengan MQTT, MySQL, dan model *machine learning*. Selain hal tersebut, Flask melindungi informasi dengan menyediakan fitur *request validation*, *session management*, dan *authentication middleware* [44].

Dalam penelitian ini, Flask digunakan sebagai *back-end* utama dalam stasiun cuaca berbasis LoRa sebagai pengelola komunikasi data antara modul *receiver* dan server. Flask berperan sebagai *subscriber* MQTT guna menerima data sensor dari modul *receiver*, lalu menyimpannya di *database* dan menyediakan *endpoint* REST API untuk visualisasi data di web berbasis HTML dan Chart.js. Flask dipilih karena dapat bekerja secara *real-time*, ringan, dan beberapa ekstensi yang mendukung sistem IoT.

## 2.20 Algoritma *Machine learning* XGBoost



**Gambar 2. 23** Lingkup *Machine Learning* [45]

Berdasarkan gambar 2.23, *machine learning* adalah salah satu dari turunan kecerdasan buatan (AI). *Machine learning* memungkinkan komputer untuk belajar dari data yang diberikan tanpa kode yang ditulis secara eksplisit [46]. Secara umum, *machine learning* terbagi menjadi 3 jenis, yaitu:

- a. *Supervised Learning*



*Supervised Learning* yaitu pelatihan model *machine learning* yang menggunakan data yang sudah diberi label, sehingga prediksi yang dihasilkan akan sangat bergantung pada label yang diberikan [46].

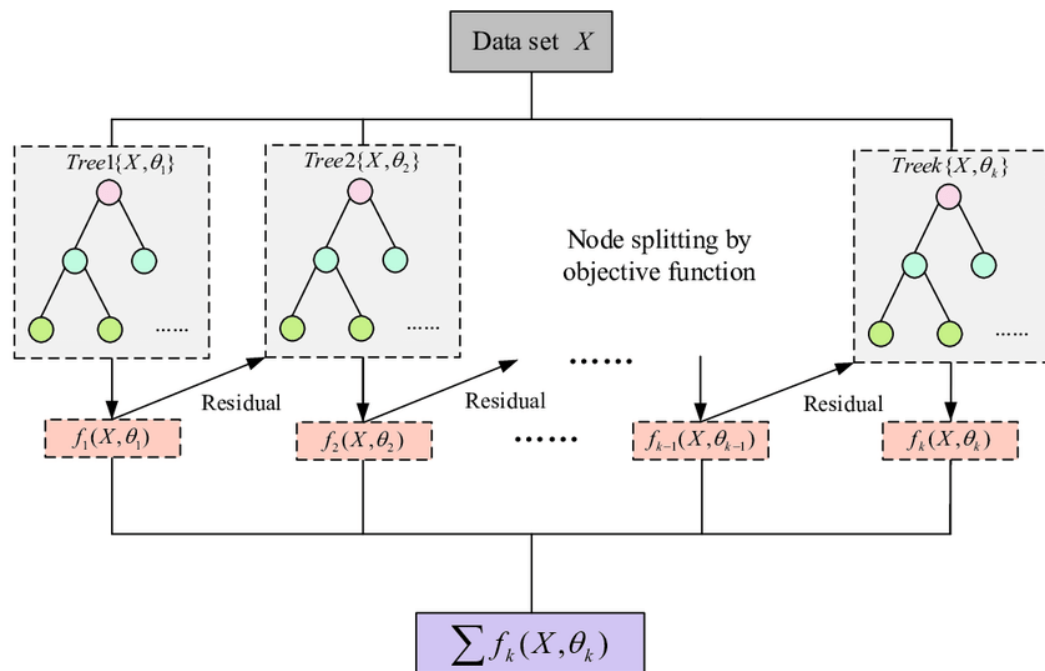
b. *Unsupervised Learning*

*Unsupervised Learning* adalah pelatihan model *machine learning* dengan tanpa memberikan label pada data. Dengan demikian, model *machine learning* akan mencari pola unik yang ada didalam data tak berlabel tersebut [46].

c. *Reinforcement Learning*

*Reinforcement Learning* memanfaatkan agen yang belajar dengan metode *trial-error* untuk mencapai tujuan tertentu [46].

Dalam penelitian ini metode *machine learning* digunakan untuk alat memahami data, analisis data, *preprocessing* data, dan pemodelan data untuk kebutuhan prediksi. Metode *machine learning* yang dipakai adalah *supervised learning*, yaitu data sensor digunakan sebagai data latih sehingga bisa menghasilkan model prediksi.



**Gambar 2. 24** Flowchart XGBoost [47]

Berdasarkan gambar 2.24, XGBoost (*Extreme Gradient Boosting*) adalah implementasi yang dioptimalkan dari algoritma *Gradient Boosting*, yang menggunakan pendekatan *ensemble learning* dengan menggabungkan banyak model sederhana untuk membuat model yang paling baik dan lebih kuat [48]. Selain itu, XGBoost memiliki beberapa konsep dasar, di antaranya adalah sebagai berikut:

a. *Boosting*

*Boosting* adalah teknik *ensemble* yang menggunakan beberapa model lemah yang saling memperbaiki kesalahan model sebelumnya dan model baru meminimalkan fungsi *loss* menggunakan gradien *boosting*.

b. Pohon Keputusan (*Decision Trees*).

Pohon keputusan digunakan sebagai *weak learners*, di mana setiap pohon mempelajari pola dari data dan membuat prediksi berdasarkan aturan percabangan.

c. Regulasi

Regularisasi membantu model untuk tidak terlalu kompleks dan menghindari *overfitting*. Regularisasi di XGboost dapat didefinisikan sebagai:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.1)$$

d. Optimasi

XGBoost menggunakan *gradient descent* untuk meminimalkan fungsi objektif. Pada setiap iterasi, model baru  $f_t$  ditambahkan untuk meminimalkan:

$$Obj^{(t)} = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (2.2)$$

Selain itu, XGBoost memiliki beberapa elemen lain seperti:

a. Fungsi Objektif

Fungsi objektif XGBoost terdiri dari dua bagian, yaitu:

- I. *Loss Function*: Mengukur seberapa baik model dalam memprediksi data.

II. *Regularization Term*: Menghindari *overfitting* dengan mengontrol kompleksitas model.

Fungsi objektif dapat ditulis sebagai berikut:

$$Obj^{(t)} = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (2.3)$$

b. Prediksi

Prediksi XGBoost adalah jumlah prediksi dari semua pohon:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (2.4)$$

Untuk menghasilkan model prediksi yang sesuai dengan kebutuhan, maka hyperparameter dapat diubah sesuai dengan kebutuhan. Beberapa hyperparameter dalam XGBoost adalah *max\_depth* (kedalaman maksimum pohon keputusan dalam boosting), *learning\_rate* (seberapa cepat model belajar dari kesalahan), *n\_estimators* (jumlah pohon keputusan dalam model), *gamma* (regulasi agar *overfitting* tidak terjadi), dan *reg\_lambda* (regulasi L2 untuk mengontrol kompleksitas model).

Dalam penelitian ini, algoritma XGBoost digunakan dalam pembuatan model prediksi regresi multivariat dengan data titik variabel suhu, kelembapan, curah hujan, dan iradiasi matahari.

Regresi multivariat bertujuan untuk memprediksi nilai-nilai variabel dependen secara bersamaan berdasarkan hubungan antara variabel-variabel tersebut, dengan melibatkan pemodelan hubungan antara beberapa variabel dependen ( $Y_1, Y_2, \dots, Y_m$ ) dan variabel independen ( $X_1, X_2, \dots, X_p$ ) [49]. Sedangkan, di dalam algoritma XGBoost, regresi multivariat dilakukan dengan dua pendekatan utama, yaitu:

a. *Multi-Output Wrapper*

*Multi-Output Wrapper* digunakan untuk melatih model XGBoost setiap target secara terpisah.

b. *Custom Objective Function*

*Custom Objective Function* adalah fungsi objektif kustom yang mempertimbangkan semua target secara bersamaan.

Untuk regresi multivariat, fungsi *loss* dimodifikasi supaya semua target dapat menjadi pertimbangan. Misalnya, jika didefinisikan bahwa  $m$  adalah target, maka fungsi *loss* dapat didefinisikan sebagai berikut:

$$L(y_i, \hat{y}_i) = \sum_{j=1}^m L_j(y_{ij}, \hat{y}_{ij}) \quad (2.5)$$

Dalam penelitian ini, metode prediksi yang digunakan adalah regresi multivariat dengan output numerik variabel suhu, kelembapan, curah hujan, dan iradiasi matahari. Selain itu, variabel dependen dalam penelitian ini adalah hasil prediksi parameter cuaca, sedangkan yang dimaksud variabel independen dalam penelitian ini adalah data cuaca historis.

## 2.21 Evaluasi Metrik

Evaluasi metrik adalah alat analisis untuk menilai kinerja suatu model. Beberapa contoh evaluasi metrik yang umum digunakan adalah RMSE (*Root Mean Squared Error*), MSE (*Mean Squared Error*), dan  $R^2$  (*R-Squared*).

- a. MSE memberikan gambaran tentang seberapa besar nilai *error* di dalam model secara keseluruhan dengan cara menghitung rata-rata antara nilai prediksi dan nilai aktual.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.6)$$

- b. RMSE memberikan gambaran ukuran *error* dalam satuan yang sama dengan variabel target. RMSE adalah akar kuadrat dari MSE.

$$RMSE = \sqrt{MSE} \quad (2.7)$$

- c.  $R^2$  berkisar antara 0 hingga 1, di mana jika nilai mendekati 1 maka itu menunjukkan bahwa model lebih baik.  $R^2$  dihitung dengan mengukur proporsi variabilitas dalam variabel dependen yang dapat dijelaskan model.

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (2.8)$$

Dalam penelitian ini, evaluasi metrik yang digunakan adalah RMSE, MSE, dan  $R^2$  untuk mengetahui performa model XGBoost pada regresi multivariat data cuaca satu titik.

## 2.22 Google Colab

Google *Collaboratory* atau Google *Colab* adalah platform berbasis *cloud* yang memungkinkan pengguna untuk melakukan kegiatan menulis kode, dan menjalankan kode melalui web browser. selain itu, Google *Colab* memberikan pengguna akses terhadap GPU dan TPU secara gratis, sehingga pengguna tidak perlu menggunakan perangkat lokal pribadi mereka. Selain hal tersebut, Google *Colab* sudah didukung oleh banyak *library* populer seperti *Scikit-learn*, *keras* dan sebagainya, serta sudah terintegrasi dengan *google drive* yang menambah kemudahan pengelolaan data bagi pengguna [50]. Spesifikasi komputer virtual dari *google colab* adalah sebagai berikut:

**Tabel 2. 16** Spesifikasi Google *Colab* [51]

Google <i>Colab</i>	Spesifikasi
CPU	Intel® Xeon® (2 vCPUs, ~2.2 GHz)
RAM	12,6GB
GPU	NVIDIA Tesla T4 / K80 / P100
VRAM GPU	±16 GB (tergantung tipe GPU)
Penyimpanan <i>Disk</i>	±78 GB ( <i>temporary</i> )
<i>Runtime</i> Limit	Maksimum 12 jam per sesi

Dalam penelitian ini, Google *Colab* digunakan dalam *preprocessing* dataset seperti *cleaning* data, hingga pembuatan model *machine learning* dengan algoritma XGBoost.

## 2.23 Python

Python adalah bahasa pemrograman *interpretative* yang dibuat menjadi bahasa tingkat tinggi yang mudah dipahami manusia alih-alih menggunakan bahasa tingkat

rendah yang lebih mendekati pada mesin. selain itu, Python memungkinkan untuk diterapkan di berbagai sistem operasi, dan berbagai peruntukan seperti proses otomatisasi, pengembangan web, analisis data, dan kecerdasan buatan atau *machine learning*. Dengan demikian, Python menjadi salah satu bahasa pemrograman paling populer di dunia [52].

Dalam penelitian ini, Python digunakan sebagai bahasa utama dalam pengembangan sistem stasiun cuaca berbasis LoRa, khususnya dalam pembangunan model *machine learning*, mengembangkan *back-end* API menggunakan Flask, dan pengolahan data sensor. Selain itu, Python juga mendukung berbagai *library* untuk pemrosesan data, *machine learning*, dan kemudahan integrasi dengan MySQL sebagai penyimpanan data.

## 2.24 Scikit-Learn

Scikit-learn adalah *library machine learning* berbasis Python dan bersifat *open-source* yang digunakan untuk keperluan analisis data, pemrosesan data, dan pembangunan model machine learning. Selain itu, scikit-learn dibangun atas beberapa *library* pengolahan data populer seperti NumPy, SciPy, dan Matplotlib. *Library* ini juga sudah terintegrasi dengan Google Colab atau Jupyter Notebook untuk mengembangkan model *machine learning*. Selain hal tersebut, Scikit-learn mendukung berbagai algoritma *supervised learning* dan *unsupervised learning* termasuk regresi, klasifikasi, dan *clustering*. *Library* ini pula telah dilengkapi dengan beberapa evaluasi metrik seperti MSE dan *precision-recall* [53].

Scikit-learn digunakan sebagai *library* pembantu untuk pengolahan data cuaca mulai dari *preprocessing* hingga pembangunan model dan evaluasi model *machine learning*.

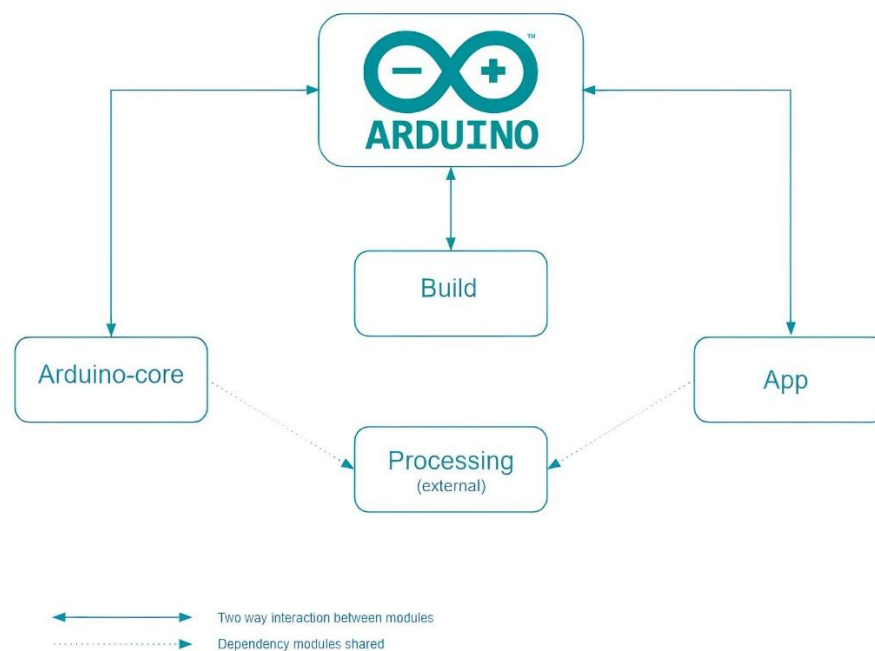
## 2.25 Visual Studio Code

VS Code adalah editor kode gratis dan *open-source* yang dikembangkan oleh Microsoft yang memiliki keandalan, ringan, dan dilengkapi banyak fitur yang bisa divariasikan lebih banyak lagi menggunakan ekstensi. Selain itu, VS Code memiliki banyak dukungan seperti Git, *debugging*, dan terminal terintegrasi. Editor ini berbasis Elektron yang memungkinkan pengguna dapat menjalankan aplikasi di

berbagai sistem operasi seperti Windows, Linux, dan macOS. Selain itu, VS Code memiliki ekosistem ekstensi yang luas, mulai dari penambahan bahasa pemrograman, *tools debugging*, integrasi dengan *cloud*, dan *container* seperti Kubernetes dan Docker [54].

Dalam penelitian ini, VS Code digunakan sebagai editor kode utama dalam pengelolaan sumber daya sistem stasiun cuaca yang menggunakan Flask. Selain itu VS Code dipilih karena memiliki dukungan integrasi langsung dengan MySQL, Python, dan *library* Scikit-learn dan XGBoost. Editor ini juga memudahkan *debugging* dan integrasi dengan Git sehingga manajemen kode lebih efisien. VS Code juga memiliki ekstensi untuk Flask dan MQTT memungkinkan pengembangan yang lebih cepat, rapi, dan terstruktur.

## 2.26 Arduino IDE



**Gambar 2. 25** Arsitektur *Development View* Arduino IDE [55]

Arduino IDE adalah lingkungan pengembangan terintegrasi (IDE) berbahasa C/C++ yang digunakan untuk menulis, merekayasa, mengompilasi, dan mengunggah kode ke dalam MCU seperti Arduino, ESP32, dan sebagainya. Berdasarkan gambar 2.25, arsitektur *development view* dari Arduino IDE memiliki beberapa komponen penting yaitu Arduino-core, App (GUI dan Editor), *build system*, dan *processing* (eksternal). Arduino-core yang berguna sebagai penyedia

*library* ini yang bertugas sebagai pengontrol dengan bagian *hardware*. *Build system* berperan sebagai pengelola proses kompilasi, *linking*, dan pengunggahan kode menggunakan *toolchain* seperti *avr-gcc* atau *xtensa-gcc*. *Processing* digunakan sebagai basis pemrosesan kode, sedangkan App menyediakan GUI untuk menulis, merekayasa, dan mengunggah kode ke perangkat *emdedded* [55].

Dalam penelitian ini, Arduino IDE digunakan sebagai IDE utama untuk memprogram LoRa32 V2.1\_1.6 dengan mode *transmitter* dan *receiver*. IDE ini digunakan karena mendukung beberapa *library* tambahan untuk kompatibilitas LoRa, RS-485, Wi-Fi, SPI, I2C, UART, MQTT, dan *library* lain yang dibutuhkan dalam sistem stasiun cuaca berbasis LoRa. Selain itu, IDE ini memiliki serial monitor yang berguna sebagai *debugging*, pengujian komunikasi, dan pemantauan data secara *real-time*.



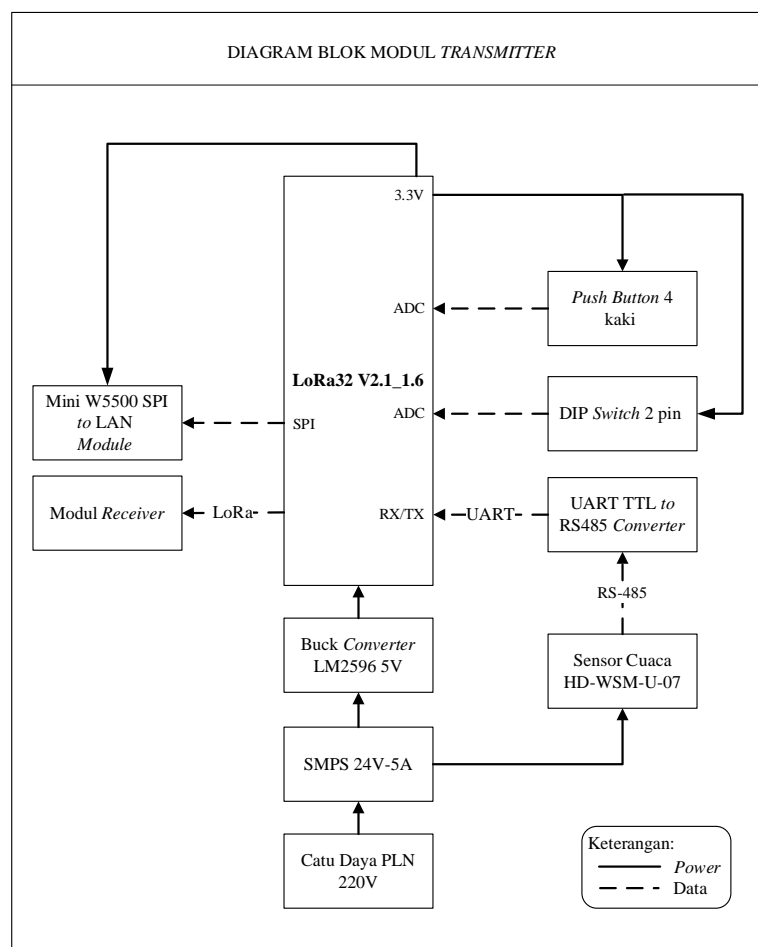
## BAB III

### METODE PENELITIAN

#### 3.1 Diagram Blok dan *Flowchart*

##### 3.1.1 Blok diagram

Berikut adalah skema perangkat stasiun cuaca berbasis LoRa yang akan digunakan di dalam penelitian *Data Forecasting* menggunakan algoritma XGBoost.



**Gambar 3. 1** Diagram Blok Sistem Modul *Transmitter*

Dalam sistem stasiun cuaca berbasis LoRa, diagram blok modul komunikasi terbagi menjadi dua bagian, yaitu modul *transmitter* dan modul *receiver*. Modul *transmitter* bertugas mengirimkan data sensor ke

modul *receiver*, lalu modul *receiver* akan mengirimkan data sensor ke server melalui internet.

Berdasarkan gambar 3.1, bagian-bagian penyusun modul *transmitter* sistem stasiun cuaca berbasis LoRa meliputi:

a. Modul *Receiver*

Modul *receiver* bertugas sebagai penerima data dari modul *transmitter* dan mentransmisikan data ke internet sehingga bisa tersimpan di *database*. Modul ini disuplai oleh tegangan 24VDC yang diberikan oleh SMPS 24V-5A.

b. LoRa32V2.1\_1.6

LoRa32V2.1\_1.6 digunakan sebagai MCU utama yang bertugas dalam mengelola komunikasi LoRa antara modul *transmitter* dan *receiver* di dalam sistem stasiun cuaca berbasis LoRa. Input yang masuk ke dalam modul ini adalah data dari sensor HONDE HD-WSM-U-07 melalui UART TTL to RS-485 Converter, setelah itu data akan dikirimkan melalui LoRa ke modul *receiver* dan diteruskan ke PC Server melalui Wi-Fi.

c. Sensor Cuaca HONDE HD-WSM-U-07

Sensor Cuaca HONDE HD-WSM-U-07 digunakan sebagai perangkat pengambil data cuaca dengan protokol Modbus melalui RS-485. Sensor ini disuplai oleh tegangan 24VDC yang berasal dari SMPS 24V-5A. Data sensor dikirim melalui protokol Modbus RS-485 ke MCU melalui UART TTL to RS-485 Converter. Kabel RS-485 A dan B dihubungkan dengan pin A dan B UART TTL to RS-485 Converter.

d. UART TTL to RS-485 Converter

UART TTL to RS-485 Converter berfungsi sebagai penghubung antara sensor cuaca HONDE HD-WSM-U-07 yang menggunakan protokol RS-485 dengan LoRa32V2.1\_1.6 yang menggunakan TTL (*Transistor-Transistor Logic*) sehingga dapat terjadi komunikasi yang kompatibel. Pin A+ pada UART TTL to RS-485 Converter dihubungkan ke pin A RS-485 dari sensor HONDE HD-WSM-U-07, sedangkan pin

B- dihubungkan ke pin B RS-485 sensor untuk menyalurkan data sensor untuk dikonversi ke UART TTL. Pin TX dari *converter* dihubungkan ke TX LoRa32 V2.1\_1.6, sementara pin RX *converter* dihubungkan ke GPIO0 LoRa32 V2.1\_1.6 sebagai RX untuk menyalurkan data yang telah dikonversi ke MCU.

e. Modul Mini W5500 SPI to LAN

Modul Mini W5500 SPI to LAN digunakan sebagai penghubung antara MCU LoRa32V2.1\_1.6 dengan jaringan internet melalui kabel LAN/ethernet. Modul ini digunakan sebagai opsi apabila di tempat instalasi stasiun cuaca tidak terdapat fasilitas Wi-Fi.

f. *Push button* 4 kaki

*Push button* 4 kaki digunakan sebagai antarmuka fisik. *Push button Select* berfungsi untuk mengoperasikan modul ketika tidak tersedia *Micro SD*. Sistem ON, OLED menampilkan ‘tidak ada *Micro SD*’, sistem beroperasi tanpa *Micro SD*. Tombol ini hanya ditekan ketika modul tidak berhasil mengakses *Micro SD* baik karena memori yang penuh, kerusakan fisik pada *Micro SD*, maupun kekosongan *Micro SD*. Tombol *reset* digunakan untuk konfigurasi mode saat sistem dalam keadaan menyala dengan mengatur posisi DIP *Switch* lalu menekan tombol *reset*.

g. DIP *Switch* 2 pin

DIP *Switch* digunakan sebagai antarmuka pengaturan mode modul komunikasi, yaitu *transmitter*, *receiver* menggunakan komunikasi Wi-Fi, dan *receiver* menggunakan komunikasi LAN. Konfigurasi dilakukan sebelum sistem mendapat suplai daya. Jika tidak memungkinkan untuk melakukan konfigurasi mode pada saat sistem dalam keadaan mati, maka mode tetap dapat diubah dengan cara mengubah konfigurasi DIP *Switch* lalu menekan tombol *reset*. Untuk melakukan konfigurasi mode, dapat dengan mengatur posisi DIP *Switch* Mode dengan fungsi dan kombinasi sebagai berikut:

**Tabel 3. 1** Konfigurasi DIP Switch

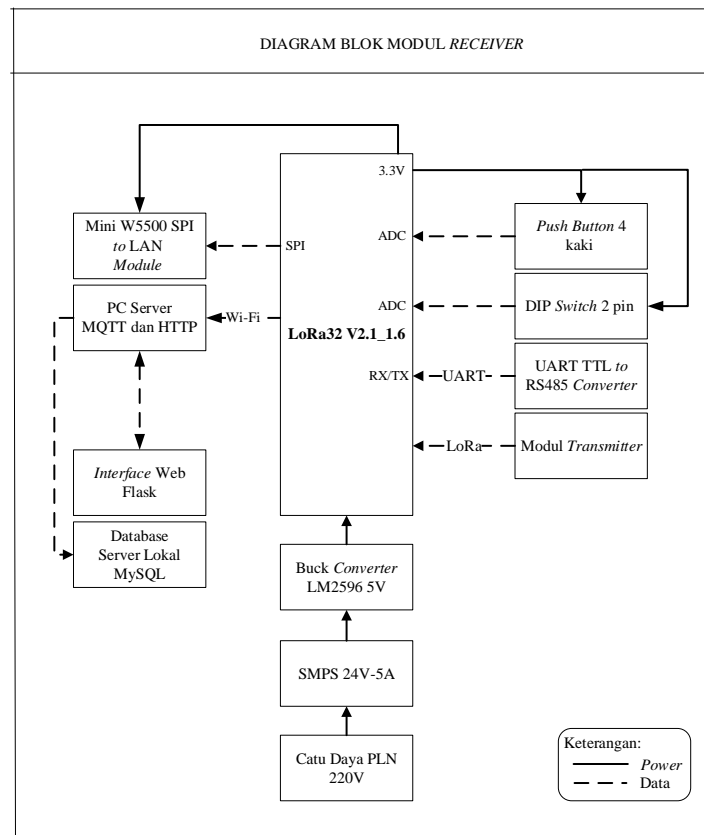
<b>Konfigurasi</b>	<b>Fungsi</b>
0 0	<i>Transmitter</i>
0 1	<i>Transmitter</i>
1 0	<i>Receiver Wi-Fi</i>
1 1	<i>Receiver LAN</i>

h. SMPS 24V-5A

SMPS 24V-5A digunakan sebagai penurun tegangan catu daya PLN yang memiliki nilai tegangan sebesar 220VAC ke tegangan 24VDC agar sesuai dengan kebutuhan modul *transmitter*, modul *receiver*, dan sensor HONDE HD-WSM-U-07.

i. Buck Converter LM2596 5V

Buck Converter LM2596 5V digunakan sebagai penurun tegangan dari output SMPS 24V-5A yang mengeluarkan tegangan sebesar 24VDC ke tegangan sebesar 5VDC agar kompatibel dengan tegangan input yang dibutuhkan oleh LoRa32V2.1\_1.6, yaitu sebesar 5VDC.



**Gambar 3. 2** Diagram Blok Sistem Modul *Receiver*

Berdasarkan gambar 3.2, modul *receiver* memiliki penyusun yang sama dengan modul *transmitter* dengan beberapa tambahan yaitu:

a. PC Server

PC Server digunakan sebagai pusat pengelola data, penyimpanan data, pengelola *back-end*, dan *front-end* di dalam sistem stasiun cuaca berbasis LoRa. PC Server menjalankan Flask sebagai *back-end* API dan menyimpan data cuaca dalam *database* MySQL. PC server menerima data sensor dari modul *receiver* melalui MQTT, lalu menyimpannya ke *database*, dan menampilkan data ke *interface* web.

b. Database

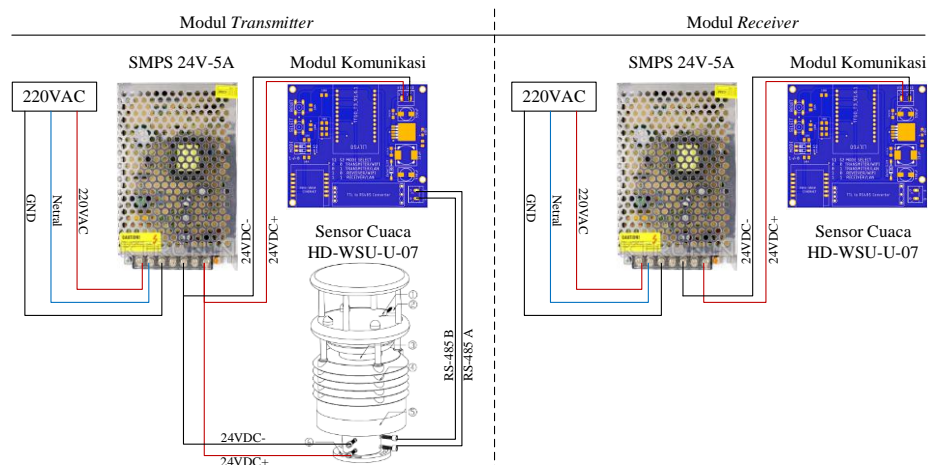
*Database* digunakan sebagai tempat penyimpanan data historis sensor HONDE HD-WSM-U-07 dan hasil prediksi regresi multivariat dengan algoritma XGBoost di dalam sistem stasiun cuaca berbasis LoRa.

Input data masuk melalui Falsk API. Data prediksi dan data sensor dapat di akses oleh *interface* web untuk ditampilkan.

c. *Interface* Web

*Interface* web digunakan sebagai media pemantauan variabel cuaca dari sensor HONDE HD-WSM-U-07 dan hasil prediksi cuaca dengan algoritma XGBoost di dalam sistem stasiun cuaca berbasis LoRa melalui *dashboard* berbasis HTML dan Chart.js. Input *interface* web adalah data yang diambil dari *database* melalui Flask API. Adapun outpunya adalah tampilan *dashboard* mengenai pembacaan parameter cuaca dan hasil prediksi.

### 3.1.2 Rangkaian elektrikal keseluruhan

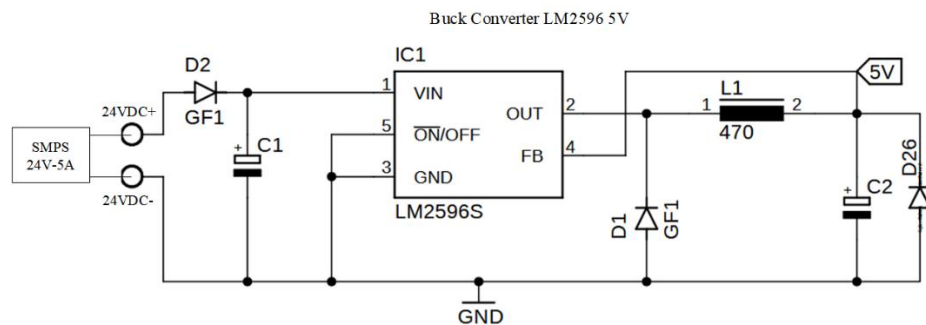


**Gambar 3. 3** Rangkaian Elektrikal

Berdasarkan gambar 3.3, ditunjukkan bahwa sistem stasiun cuaca berbasis LoRa ditenagai oleh catu daya PLN 220VAC. Setelah itu, tegangan masuk ke SMPS 24V-5A untuk diturunkan menjadi 24VDC, dan disalurkan ke sensor cuaca HONDE HD-WSM-U-07 dan ke Buck Converter LM2596 5V yang sudah ada di dalam modul komunikasi. Selain itu, kabel RS-485 (A dan B) dari sensor cuaca HONDE HD-WSM-U-07 di hubungkan ke modul UART TTL to RS-485 yang sudah ada di dalam modul komunikasi.

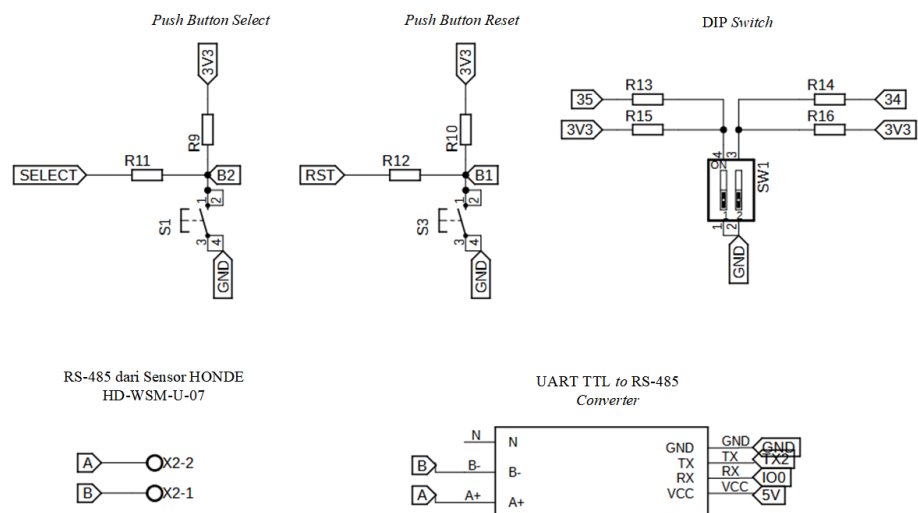
### 3.1.3 Rangkaian elektrik modul komunikasi

Modul komunikasi terdiri atas modul *transmitter* dan *receiver*. Keduanya memiliki skematik elektrik yang sama. Skematik elektrik modul komunikasi dapat dilihat di gambar 3.4.



**Gambar 3. 4** Skematik Rangkaian Blok *Power*

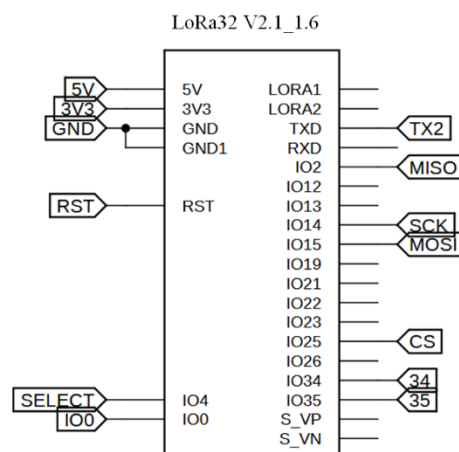
Berdasarkan gambar 3.4, di dalam blok *power* terdapat rangkaian *Buck Converter* LM2596 5V yang berguna untuk menurunkan tegangan keluaran dari SMPS 24V-5A menjadi 5VDC untuk memberikan tegangan pada LoRa32 V2.1\_1.6.



**Gambar 3. 5** Skematik Rangkaian Blok Input

Berdasarkan gambar 3.5, blok input terdiri dari 2 *push button* 4 kaki, *DIP Switch* 2 pin, dan *UART TTL to RS-485 Converter*. *Push button select* terhubung dengan *ground* dan 3,3V sebagai *power* yang berasal dari MCU. Selanjutnya, *push button select* terhubung dengan GPIO4

MCU sebagai jalur data digital dari *push button* ke MCU. Sedangkan *push button reset* terhubung dengan pin RST di MCU. DIP Switch terhubung dengan *ground* dan pin 3,3V MCU sebagai *power* dan pin GPIO34 dan GPIO35 sebagai jalur transmisi data digital yang dikirim DIP Switch ke MCU. Pin A+ UART TTL to RS-485 Converter dihubungkan dengan pin A RS-485 dari sensor HONDE HD-WSM-U-07 dan pin B- UART TTL to RS-485 Converter dihubungkan dengan pin A RS-485 dari sensor HONDE HD-WSM-U-07. Sedangkan pin TX UART TTL to RS-485 Converter dihubungkan dengan pin TX LoRa32 V2.1\_1.6 dan pin RX UART TTL to RS-485 Converter dihubungkan dengan pin GPIO0 LoRa32 V2.1\_1.6 yang digunakan sebagai RX.

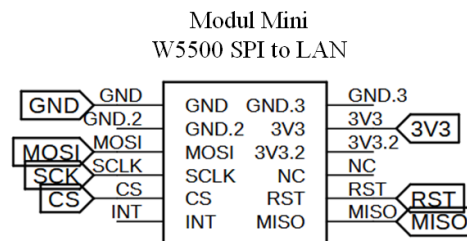


**Gambar 3. 6** Skematik Rangkaian Blok Kontrol

Berdasarkan gambar 3.6, blok *control* terdiri dari LoRa32 V2.1\_1.6 yang menjadi pusat pemrosesan data dalam sistem stasiun cuaca. Modul ini memakai SPI untuk komunikasi dengan *chip* LoRa SX1276. Selain itu, untuk berkomunikasi dengan sensor cuaca HONDE HD-WSM-U-07, LoRa32 V2.1\_1.6 menggunakan *port* UART melalui konverter UART TTL ke RS-485 yang menggunakan protokol RS-485 untuk membaca data cuaca. Selain itu, modul juga telah dilengkapi dengan *slot micro-SD* dengan kapasitas hingga 32GB agar seluruh proses komunikasi data dan aktivitas *logging* sistem tetap berjalan tanpa adanya interupsi atau gangguan.. Untuk tampilan, OLED 0,96 inci beroperasi



dengan I2C. Selain hal tersebut, *port* GPIO digital digunakan untuk menghubungkan MCU dengan *push button* dan *DIP Switch*.



**Gambar 3. 7** Skematik Rangkaian Blok Output

Berdasarkan gambar 3.7, blok output terdiri dari modul Mini W5500 SPI to LAN yang digunakan untuk menghubungkan perangkat ke internet melalui kabel LAN/ethernet. Pin SCLK Modul Mini W5500 SPI to LAN terhubung dengan GPIO14 LoRa32 V2.1\_1.6 sebagai pengatur kecepatan transfer data, pin MISO Modul Mini W5500 SPI to LAN terhubung dengan GPIO2 LoRa32 V2.1\_1.6 sebagai jalur *slave* mengirim data ke *master*, pin MOSI Modul Mini W5500 SPI to LAN terhubung dengan GPIO15 LoRa32 V2.1\_1.6 sebagai jalur *master* mengirimkan data ke *slave*, pin CS Modul Mini W5500 SPI to LAN terhubung dengan GPIO25 LoRa32 V2.1\_1.6 sebagai pengatur yang memastikan hanya satu perangkat yang aktif dalam satu waktu guna menghindari konflik dalam bus SPI jika digunakan oleh beberapa perangkat.

Adapun konfigurasi pin di dalam modul komunikasi diringkas pada tabel sebagai berikut:

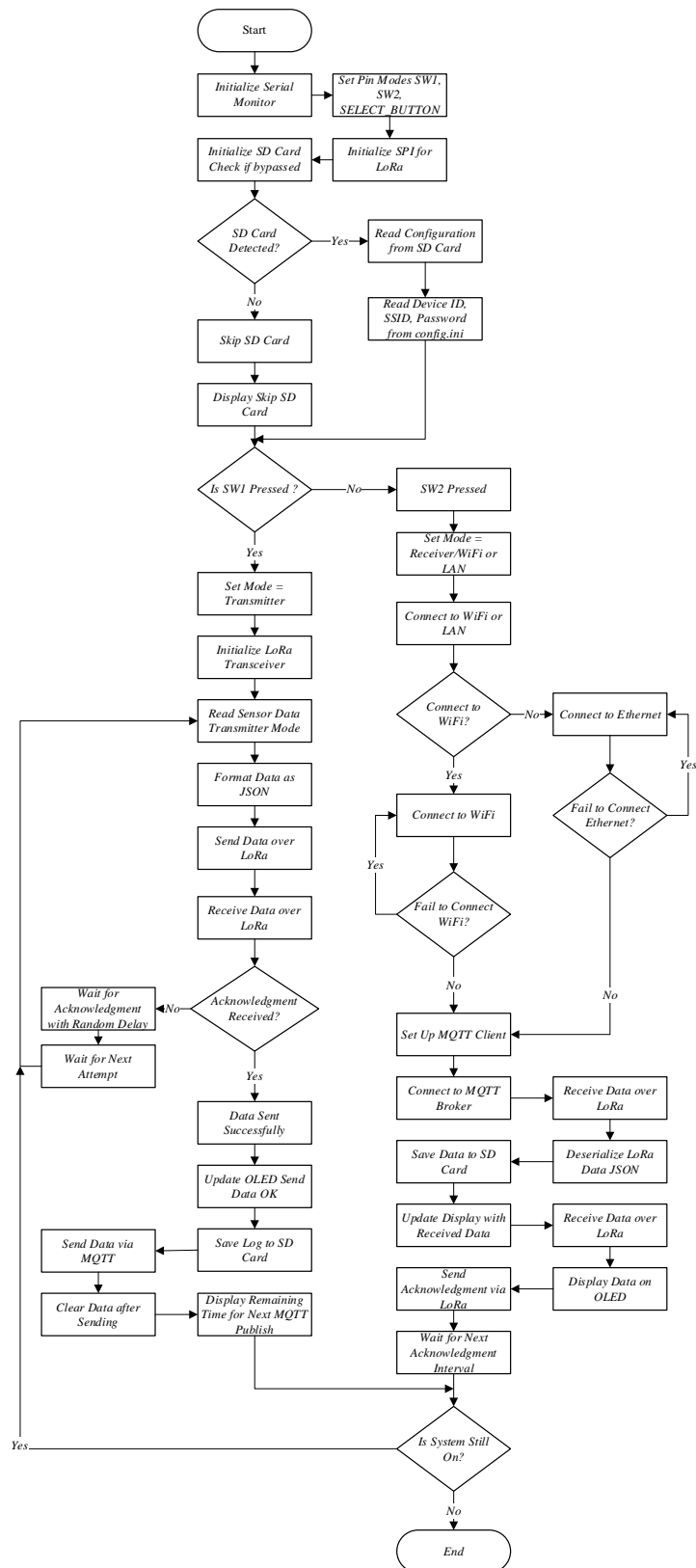
**Tabel 3. 2** Konfigurasi I/O Modul Komunikasi

Komponen	I/O LoRa32 V2.1_1.6
<i>Push button Select</i>	3,3V, GND, GPIO4
<i>Push button Reset</i>	3,3V, GND, RST
<i>DIP Switch</i>	3,3V, IO35, IO34
UART TTL to RS-485	5V, GND, IO0, TXD

**Tabel 3. 3** Lanjutan

<b>Komponen</b>	<b>I/O LoRa32 V2.1_1.6</b>
Buck <i>Converter</i> 5V	5V
Modul Mini W5500 SPI <i>to</i> LAN	3,3V, GND, IO15, IO14, IO2, IO25, RST

### 3.1.4 Flowchart sistem utama



Gambar 3. 8 Flowchart Sistem Utama

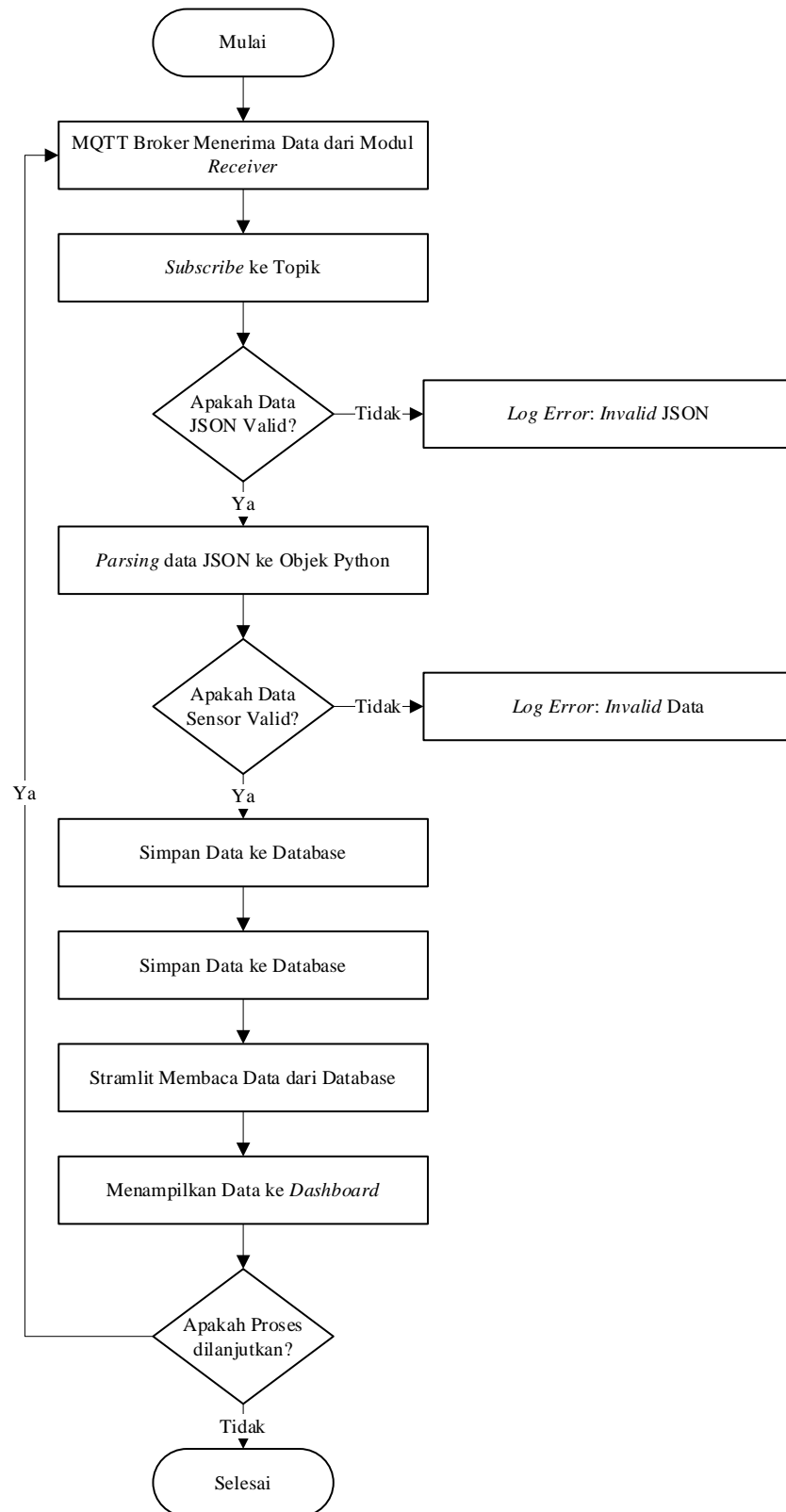
*Flowchart* proses kerja dari sistem utama dari sistem stasiun cuaca ditunjukkan pada gambar 3.8. Proses diawali dengan inisialisasi sistem seperti setup LoRa, dan serial monitor. Setelah itu, mode sistem *transmitter* atau *receiver* dipilih. Apabila kartu SD terdeteksi maka sistem akan membaca konfigurasi dari kartu SD yang berisikan *device ID*, *SSID*, dan *password* dari *file config*. Namun, apabila kartu SD tidak terdeteksi, maka sistem akan melakukan *bypass* kartu SD dan menampilkan bahwa kartu SD tidak terdeteksi.

Proses berlanjut pada pemilihan mode modul komunikasi. Apabila SW1 ditekan, maka mode modul komunikasi menjadi mode *transmitter*. Setelah itu, modul *transmitter* akan membaca data dari sensor dan membuat format data JSON untuk data cuaca, lalu data dikirim melalui LoRa oleh modul *transmitter* dan menunggu modul *receiver* mengirimkan pesan *acknowledgment* yang berarti data sensor dapat diterima dengan baik. Apabila modul *transmitter* tidak menerima pesan *acknowledgment*, maka modul *transmitter* akan menunggu beberapa saat. Apabila modul *transmitter* tetap tidak menerima pesan *acknowledgment* setelah waktu tunggu habis, maka modul *transmitter* akan mengirim data kembali ke modul *receiver* tanpa pesan *acknowledgment* dari modul *receiver*. Sedangkan, apabila pesan *acknowledgment* diterima, maka transmisi data antara modul *transmitter* dan modul *receiver* dapat dilakukan dengan baik, dan pesan sukses akan ditampilkan di OLED, serta data disimpan di kartu SD lalu dikirimkan melalui MQTT ke server. Setelah itu, sistem menampilkan pesan menunggu MQTT *publish*.

Sedangkan, apabila SW 2 ditekan, maka mode modul komunikasi akan menjadi mode modul *receiver* yang memiliki dua mode, yaitu *receiver* Wi-Fi dan *receiver* LAN. Modul akan terhubung ke internet melalui Wi-Fi atau LAN sesuai dengan mode yang dipilih. Setelah itu, modul melakukan *setup* MQTT *client* dan menyambungkan modul ke MQTT broker. Ketika modul *transmitter* mengirimkan data melalui

LoRa, maka data akan diterima oleh modul *receiver* dan disimpan dalam kartu SD, lalu modul akan menampilkan *display* bahwa modul *receiver* menerima data dengan baik serta modul *receiver* akan mengirimkan pesan *acknowledgment* melalui LoRa dan menunggu untuk transmisi pesan berikutnya.

### 3.1.5 Flowchart pengelolaan data



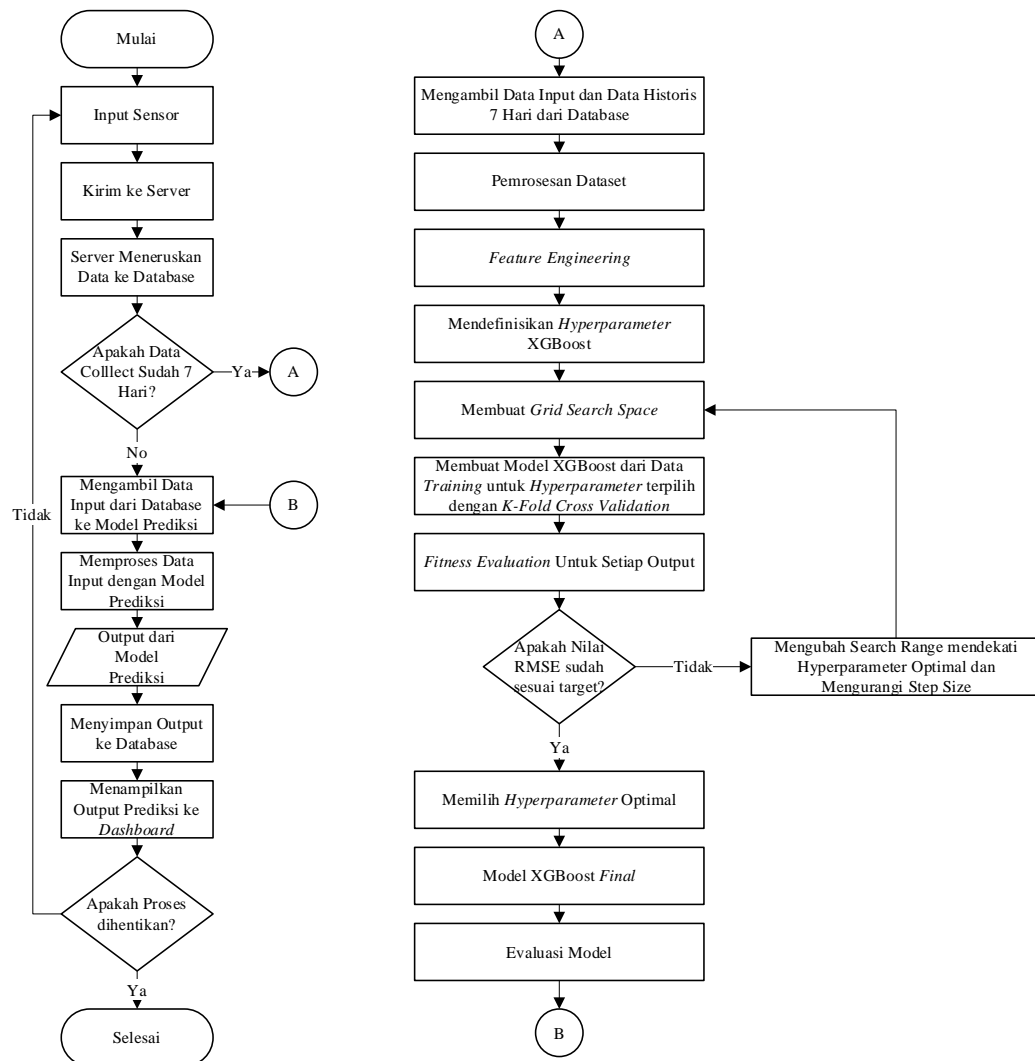
**Gambar 3. 9** Flowchart sistem

Sistem dimulai saat MQTT broker yang berjalan di server lokal menerima data sensor yang dikirim dengan format JSON dari modul *receiver* melalui protokol MQTT. Setelah itu, MQTT *Client Subscriber* men-*subscribe* topik MQTT yang berisi data sensor. Selanjutnya, i JSON dilakukan pada data sensor yang diterima dan diubah ke struktur objek Python. Apabila format JSON tidak valid, seperti terdapat sintaks yang tidak lengkap atau terdapat kesalahan dalam struktur, maka sistem akan mencatat *error* ke *log file* dan data tersebut akan diabaikan.

Data yang berhasil di-*parsing* kemudian akan diperiksa ditahap validasi data sensor. Data dipastikan lengkap, diperiksa tipe datanya, dan memeriksa apakah data yang masuk itu masuk akal. Setelah itu, data akan disimpan ke *database*. Apabila validasi data sensor gagal, maka data tersebut akan dibuang, dan sistem akan mencatat ketidakvalidan data ke *log file*.

Setelah data tersimpan di *database*, maka Flask yang berada di server lokal akan membaca data yang baru masuk dari *database* dan menampilkannya ke dalam *dashboard*.

### 3.1.6 Flowchart sistem prediksi



**Gambar 3. 10** Flowchart sistem prediksi

Proses dimulai dengan server yang menerima data sensor dari modul *receiver* dan menyimpannya di *database*. Setelah itu, apabila data sudah mencapai 7 hari, maka pembaruan model akan dilakukan. Data historis selama 7 hari dan data input memasuki tahap *preprocessing* seperti pembersihan data (*missing value*, *outliers*) dan normalisasi data. Setelah itu, *feature engineering* dilakukan untuk mengekstrak pola temporal dari data *time series*. Selanjutnya, data dibagi menjadi *training set* dan *test test*. Tahap selanjutnya adalah optimasi *hyperparameter* dengan *Grid Search*, dimana *Search Space Hyperparameter* didefinisikan yang

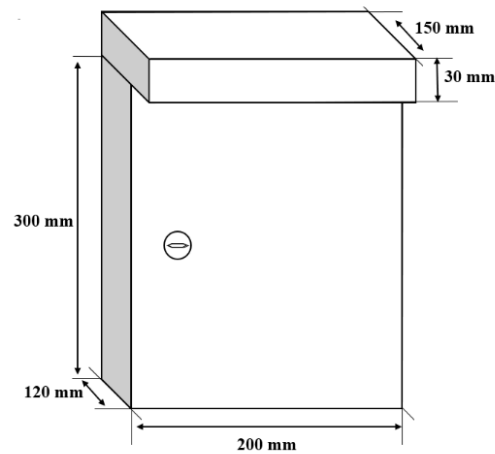


meliputi *max\_depth*, *learning\_rate*, *n\_estimators*, *gamma*, dan *reg\_lambda*. *Grid Search* mencoba semua kombinasi *hyperparameter* terbaik di *search space*. Untuk setiap kombinasi, model XGBoost dilatih dan dievaluasi menggunakan *K-Fold Cross Validation*. Proses ini membagi data *training* menjadi *K subset*, dimana model dilatih pada *K-1 subset* lalu divalidasi pada *subset* yang tersisa. Proses ini diulang sebanyak *K* kali dan performa model dihitung sebagai rata-rata dari semua iterasi. Apabila nilai evaluasi (contohnya adalah RMSE) sudah mencapai akurasi yang diinginkan, maka kombinasi *hyperparameter* tersebut akan dipilih. Namun, jika nilai evaluasi belum mencapai akurasi yang diinginkan, maka proses *Grid Search* akan diulang dengan *search space* yang disempitkan mendekati nilai optimal sebelumnya.

Model yang sudah ada memprediksi nilai data sensor input dan menyimpan hasil prediksi ke *database*. Setelah itu, data prediksi diambil dari *database* dan ditampilkan ke *dashboard*.

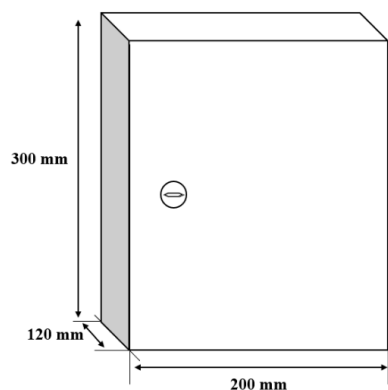
### 3.2 Proyeksi 3 Dimensi Perangkat Keras Sistem

Seperti yang ditampilkan pada gambar 3.11, *housing outdoor* memiliki bentuk persegi panjang dengan material *plat* besi yang memiliki lapisan tahan air dengan penggunaan spesifik pada *outdoor* atau luar ruangan. Panel *box* ini memiliki ukuran tinggi 30 cm, lebar 20 cm, dan tebal 12 cm yang berfungsi sebagai rumah dari modul komunikasi dan elektrik pendukung. Panel *box* dapat dipasang pada dinding dengan cara melepas papan dan modul yang terdapat di dalam panel terlebih dahulu, dan selanjutnya dapat menggunakan mur baut beton/tembok dengan rekomendasi ukuran minimal M3 sebanyak 4 buah. Selain itu, panel *box* juga dapat ditambahkan *bracket* atau penopang bantu untuk pemasangan non tembok/dinding.



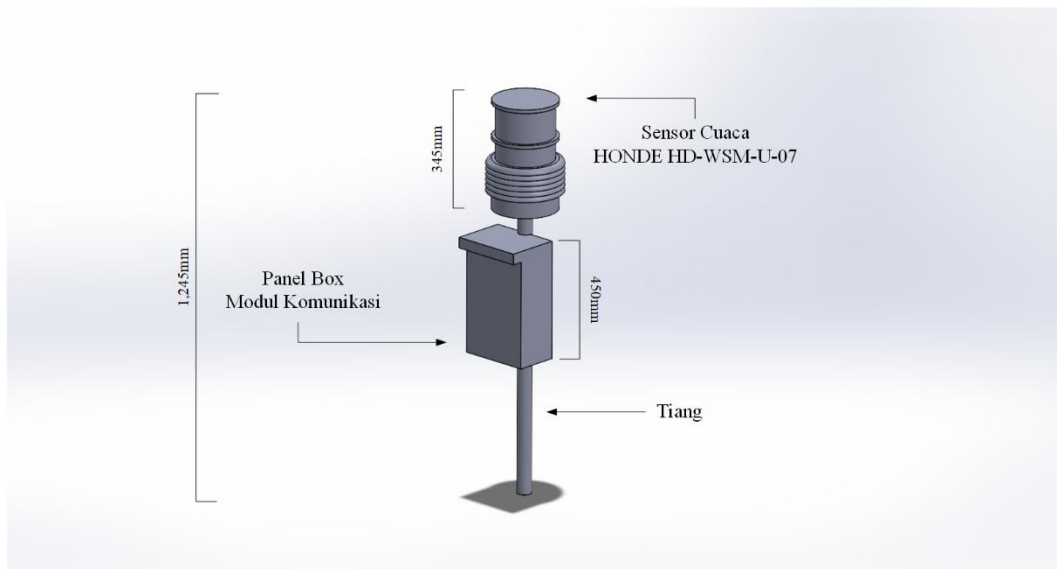
**Gambar 3. 11** Proyeksi 3 Dimensi Perangkat Keras Sistem *Outdoor*

Seperti yang ditampilkan pada gambar 3.12, *housing indoor* memiliki bentuk persegi panjang dengan material plat besi yang memiliki lapisan semi-tahan air dengan penggunaan spesifik pada *indoor* atau dalam ruangan. Panel *box* ini memiliki ukuran tinggi 30 cm, lebar 20 cm, dan tebal 12 cm yang berfungsi sebagai rumah dari modul komunikasi dan elektrik pendukung. Panel *box* dapat dipasang pada dinding dengan cara melepas papan dan modul yang terdapat di dalam panel terlebih dahulu, dan selanjutnya dapat menggunakan mur baut beton/tembok dengan rekomendasi ukuran minimal M3 sebanyak 4 buah. Selain itu, panel *box* juga dapat ditambahkan *bracket* atau penopang bantu untuk pemasangan non tembok/dinding.



**Gambar 3. 12** Proyeksi 3 Dimensi Perangkat Keras Sistem *Indoor*

Berdasarkan gambar 3.13, instalasi sensor cuaca HONDE HD-WSM-U-07 dilakukan di atas tiang dengan panjang 900mm. Sedangkan panel *box* komunikasi dipasang di bawah sensor dengan menempelkan panel *box* ke tiang, ini lakukan agar tidak memerlukan kabel RS-485 yang terlalu panjang.



**Gambar 3. 13** Proyeksi 3 Dimensi Perangkat Keras

### 3.3 Teknik Fabrikasi

#### 3.3.1 Alat dan bahan

Sistem stasiun cuaca berbasis LoRa membutuhkan beberapa alat dan bahan yang digunakan untuk merancang sistem yang baik. Berikut adalah alat dan bahan yang digunakan:

**Tabel 3. 4** Alat dan Bahan

Alat dan Bahan	Fungsi
LoRa32V2.1_1.6	MCU utama
Sensor Cuaca HONDE HD-WSM-U-07	Perangkat pengambil data
UART TTL to RS-485 Converter	Penghubung antara sensor dan MCU
Modul Mini W5500 SPI to LAN	Penghubung antara MCU dan internet melalui LAN

**Tabel 3. 5** Lanjutan

Alat dan Bahan	Fungsi
Buck Converter 5V LM2596	Penurun tegangan ke 5V
SMPS 24V-5A	Penurun tegangan ke 24V
DIP Switch 2 pin	Input nilai antarmuka
Push button 4 kaki	Input nilai antarmuka

### 3.3.2 Perancangan perangkat keras

Pada tahapan perancangan perangkat keras untuk membuat stasiun cuaca berbasis LoRa, dilakukan 3 tahapan proses yaitu:

a. Perakitan perangkat keras

Pada tahapan ini, komponen penyusun stasiun cuaca berbasis LoRa di rakit agar bisa di gunakan dengan baik.

b. Pengujian

Pada tahapan ini, setiap komponen akan di uji, sehingga setiap komponen dapat diketahui bekerja dengan semestinya

c. Kalibrasi

Pada tahapan kalibrasi, sensor dan semua komponen yang memiliki nilai diperiksa keakuratannya dengan menggunakan alat ukur pembanding.

### 3.3.3 Perancangan perangkat keras

Konsumsi daya per jam dari masing-masing komponen yang digunakan dalam sistem stasiun cuaca berbasis LoRa adalah sebagai berikut:

**Tabel 3. 6** Konsumsi Daya Setiap Komponen

Komponen	Tegangan (V)	Arus (A)	Daya (W)
Sensor Cuaca HONDE HD-WSM-U-07	24	0.2	4.8
LoRa32 V2.1_1.6	5	0.3	1.5
UART TTL to RS-485 Converter	5	0.1	0.5

Tabel 3. 7 Lanjutan

Komponen	Tegangan (V)	Arus (A)	Daya (W)
Modul Mini W5500 SPI to LAN	5	0.2	1.0
Push Button & DIP Switch	5	0.05	0.25
OLED	3.3	0.08	0.26
Buck Converter (Efisiensi 90%)	24	1	6

Berdasarkan tabel 3.6, maka total daya yang digunakan dalam sistem stasiun cuaca berbasis LoRa adalah 14,06W (396,99Wh per hari). Adapun arus yang dibutuhkan oleh SMPS dengan input 24V dapat ditentukan dengan perhitungan sebagai berikut:

$$I_{SMPS} = \frac{P_{Total}}{V_{Input}} = \frac{14,06}{24} \approx 0,59 \text{ Ampere} \quad (3.1)$$

Perhitungan 3.1 menunjukkan bahwa arus yang dibutuhkan untuk menjalankan sistem stasiun cuaca berbasis LoRa adalah 0,59A. Agar sistem dapat berjalan dengan stabil, terhindar dari over heating, dan drop saat lonjakan arus, maka total arus yang harus di akomodasi oleh SMPS adalah dua kali arus total sistem, yaitu 1,18A. Dengan demikian SMPS 24V-5A sangat cocok untuk digunakan dalam sistem ini, karena dapat mendukung arus dengan total 4,25A (efisiensi 85% dari 5V).

### 3.3.4 Perancangan perangkat lunak

Pada tahapan perancangan perangkat lunak, terdapat 6 tahapan yang dilakukan, yaitu:

#### a. Pengumpulan dataset

Pengumpulan dataset dilakukan dengan menghimpun data historis yang diambil oleh sensor sekitar 1 bulan dan mencari dataset lain seperti data iklim dari BMKG atau data dari NASA sebanyak lebih dari 2,000 poin data untuk menunjang kurangnya data yang dibutuhkan untuk membuat model.

b. Pengolahan data

Tahapan ini memiliki 3 bagian, yaitu:

I. Pembersihan data

Pada tahapan ini, *missing value* dan *outlier* di hilangkan.

II. Pembagian data

Dataset dibagi menjadi 3 bagian, yaitu 70% menjadi *training set*, 20% menjadi *validation set*, dan 10% menjadi *test set*.

III. *Preprocessing*

Pada tahapan ini, normalisasi data dilakukan agar seluruh fitur mempunyai skala yang sama.

c. Pelatihan model XGBoost

Pada tahapan ini, data akan dilatih agar dapat memprediksi nilai masing-masing data cuaca dengan algoritma XGBoost.

d. Integrasi sistem LoRa dan prediksi

Pada tahapan ini, pengkodean transmisi data dilakukan yaitu data sensor dikemas dalam format JSON dan dikirim via LoRa ke modul *receiver*. Selain itu pengkodean juga dilakukan untuk meneruskan data ke server melalui MQTT dan disimpan ke *database*. Model XGBoost yang sudah dilatih di-*deploy* di server lokal untuk menghasilkan prediksi lokal dan ditampilkan ke *dashboard*.

e. Pengujian dan evaluasi

Pengujian dilakukan pada uji jangkauan transmisi data melalui LoRa. Selain itu, dalam tahap ini dilakukan evaluasi terhadap model prediksi XGBoost dengan beberapa evaluasi metrik seperti RMSE,  $R^2$ , dan MSE.



## DAFTAR PUSTAKA

- [1] “New study shows socio-economic benefits of weather observations,” World Meteorological Organization. Diakses: 28 Februari 2025. [Daring]. Tersedia pada: <https://wmo.int/media/news/new-study-shows-socio-economic-benefits-of-weather-observations-0>
- [2] “Climate change is shaping retail sales this winter. Why weather data holds the key | Reuters.” Diakses: 28 Februari 2025. [Daring]. Tersedia pada: [https://www.reuters.com/business/retail-consumer/changing-climate-retailers-turning-weather-strategies-2024-11-04/?utm\\_source=chatgpt.com](https://www.reuters.com/business/retail-consumer/changing-climate-retailers-turning-weather-strategies-2024-11-04/?utm_source=chatgpt.com)
- [3] “6 tools our meteorologists use to forecast the weather | National Oceanic and Atmospheric Administration.” Diakses: 28 Februari 2025. [Daring]. Tersedia pada: <https://www.noaa.gov/stories/6-tools-our-meteorologists-use-to-forecast-weather>
- [4] R. B. Alley, K. A. Emanuel, dan F. Zhang, “Advances in weather prediction,” *Science*, vol. 363, no. 6425, hlm. 342–344, Jan 2019, doi: 10.1126/science.aav7274.
- [5] “Validation of MODIS Sea Surface Temperature Product in the Coastal Waters of the Yellow Sea | Request PDF,” *ResearchGate*, Okt 2024, doi: 10.1109/JSTARS.2017.2651951.
- [6] S. Millah, “Internet Indonesia Lambat, Operator Bakal Perkuat Infrastruktur,” *Bisnis.com*. Diakses: 27 Februari 2025. [Daring]. Tersedia pada: <https://infografik.bisnis.com/read/20190225/547/893076/internet-indonesia-lambat-operator-bakal-perkuat-infrastruktur>
- [7] “(PDF) A study of LoRa low power and wide area network technology,” dalam *ResearchGate*, doi: 10.1109/ATSIP.2017.8075570.
- [8] “HONDE TECHNOLOGY CO LTD DATASHEET.”
- [9] “Google Colab.” Diakses: 28 Februari 2025. [Daring]. Tersedia pada: <https://colab.research.google.com/drive/1dTArQhfVVXRWgLYaQdJsXiUshq2EZINw>
- [10] H. Soy dan Y. Dilay, “A Conceptual Design of LoRa based Weather Monitoring System for Smart Farming,” *Eur. J. Sci. Technol.*, Okt 2021, doi: 10.31590/ejosat.1011947.
- [11] M. Divya, P. Rakshitha, S. Begum, dan M. D. R. Krishna, “IOT WEATHER MONITORING SYSTEM USING ESP32”.
- [12] K. L. Sairam, “Weather Prediction System Using Linear Regression,” *Int. J. Res. Publ. Rev.*, vol. 4, hlm. 1-31219–1222, Apr 2023.
- [13] A. Rivera, P. Ponce, O. Mata, A. Molina, dan A. Meier, “Local Weather Station Design and Development for Cost-Effective Environmental Monitoring and Real-Time Data Sharing,” *Sensors*, vol. 23, no. 22, Art. no. 22, Jan 2023, doi: 10.3390/s23229060.
- [14] ClimeMET, “Wireless Weather Station CM2016,” ClimeMET. Diakses: 13 Maret 2025. [Daring]. Tersedia pada: <https://www.climemet.com/products/cm2016-wireless-weather-station>
- [15] “(PDF) A smart weather station based on sensor technology.” Diakses: 27 Februari 2025. [Daring]. Tersedia pada:



- [https://www.researchgate.net/publication/336052602\\_A\\_smart\\_weather\\_station\\_based\\_on\\_sensor\\_technology](https://www.researchgate.net/publication/336052602_A_smart_weather_station_based_on_sensor_technology)
- [16] “What are LoRa and LoRaWAN?,” The Things Network. Diakses: 13 Maret 2025. [Daring]. Tersedia pada: <https://www.thethingsnetwork.org/docs/LoRawan/what-is-LoRawan/>
  - [17] “What is LoRa.” Diakses: 13 Maret 2025. [Daring]. Tersedia pada: [https://wiki.dfrobot.com/What\\_is\\_LoRa](https://wiki.dfrobot.com/What_is_LoRa)
  - [18] J. Tan, “What is Peer-To-Peer (P2P) LoRa Communication?,” Latest Open Tech From Seeed. Diakses: 1 Maret 2025. [Daring]. Tersedia pada: <https://www.seeedstudio.com/blog/2021/04/26/what-is-peer-to-peer-p2p-LoRa-communication/>
  - [19] “LoRa32 V2.1\_1.6,” LILYGO®. Diakses: 27 Februari 2025. [Daring]. Tersedia pada: <https://lilygo.cc/products/LoRa3>
  - [20] “(PDF) Serial Peripheral Interface (SPI) Communication Application as Output Pin Expansion in Arduino Uno.” Diakses: 14 Maret 2025. [Daring]. Tersedia pada: [https://www.researchgate.net/publication/353019173\\_SERIAL\\_PERIPHERAL\\_INTERFACE\\_SPI\\_COMMUNICATION\\_APPLICATION\\_AS\\_OUTPUT\\_PIN\\_EXPANSION\\_IN\\_ARDUINO\\_UNO](https://www.researchgate.net/publication/353019173_SERIAL_PERIPHERAL_INTERFACE_SPI_COMMUNICATION_APPLICATION_AS_OUTPUT_PIN_EXPANSION_IN_ARDUINO_UNO)
  - [21] “UART TTL to RS485 Two-way Converter Module + Nodemcu - Projects / Networking, Protocols, and Devices,” Arduino Forum. Diakses: 14 Maret 2025. [Daring]. Tersedia pada: <https://forum.arduino.cc/t/uart-ttl-to-rs485-two-way-converter-module-nodemcu/1079783>
  - [22] J. Valdez dan J. Becker, “Understanding the I2C Bus,” 2015.
  - [23] “RS-485 Basics Series.”
  - [24] H. M. K. K. M. B. Herath, S. V. A. S. H. Ariyathunge, dan H. D. N. S. Priyankara, “Development of a Data Acquisition and Monitoring System Based on MODBUS RTU Communication Protocol,” *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 6, hlm. 433–440, Jun 2020, doi: 10.38124/IJISRT20JUN479.
  - [25] “Wiring of RS485 Communications Networks | Schneider Electric Indonesia,” [www.se.com](http://www.se.com). Diakses: 14 Maret 2025. [Daring]. Tersedia pada: <https://www.se.com/id/en/faqs/FA221785/>
  - [26] T. Tosin, “Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP Pada Sistem Pick-By-Light,” *Komputika J. Sist. Komput.*, vol. 10, no. 1, hlm. 85–91, Mar 2021, doi: 10.34010/komputika.v10i1.3557.
  - [27] “UART TTL to RS485 Two-way Converter | Elecrow.” Diakses: 3 Maret 2025. [Daring]. Tersedia pada: <https://www.elecrow.com/uart-ttl-to-rs485-twoway-converter-p-1545.html>
  - [28] A. Salam dan T. Sucita, “RANCANG BANGUN SISTEM JARINGAN MULTIDROP MENGGUNAKAN RS485 PADA APLIKASI PENGONTROLAN ALAT PENERANGAN KAMAR HOTEL,” 2012.
  - [29] “UART TTL to RS485 Module Two Way Converter,” Digiware Store. Diakses: 19 Maret 2025. [Daring]. Tersedia pada:

- <https://digiwarestore.com/en/pc-linked-converter/uart-ttl-to-rs485-module-two-way-converter-919478.html>
- [30] “W5500 SPI to LAN Ethernet Network Module, Network Status Indicator LED, TCP IP STM32 SPI Interface 3.3V 5V for Arduino WIZ820io RC5: Buy Online at Best Price in UAE - Amazon.ae.” Diakses: 3 Maret 2025. [Daring]. Tersedia pada: <https://www.amazon.ae/W5500-Ethernet-Indicator-Interface-WIZ820io/dp/B0BKZ5DX2K>
- [31] “W5500\_datasheet\_v1.0.2\_1.pdf.” Diakses: 3 Maret 2025. [Daring]. Tersedia pada: [https://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/W5500\\_datasheet\\_v1.0.2\\_1.pdf](https://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/W5500_datasheet_v1.0.2_1.pdf)
- [32] “Smgs 24v 10a 240w Dc Switch Mode Power Supply,” indiamart.com. Diakses: 3 Maret 2025. [Daring]. Tersedia pada: <https://www.indiamart.com/proddetail/smgs-24v-10a-240w-dc-switch-mode-power-supply-2852794726397.html>
- [33] Universitas Lampung, K. Effendi, J. Junaidi, Universitas Lampung, S. W. Suciati, dan Universitas Lampung, “Rancang Bangun Sistem Catu Daya dengan Metode Switching Mode Power Supply (SMPS) Berbasis Arduino untuk Aplikasi Electrosprinter,” *J. Teori Dan Apl. Fis.*, vol. 8, no. 1, hlm. 25–34, Jan 2020, doi: 10.23960/jtaf.v8i1.2315.
- [34] “MEAN WELL Switching Power Supply Manufacturer.” Diakses: 19 Maret 2025. [Daring]. Tersedia pada: <https://www.meanwell.com/productseries.aspx>
- [35] “<https://eprints.uai.ac.id/1461/1/ILS0072-20.pdf>.” Diakses: 3 Maret 2025. [Daring]. Tersedia pada: <https://eprints.uai.ac.id/1461/1/ILS0072-20.pdf>
- [36] “lm2596.pdf.” Diakses: 15 Maret 2025. [Daring]. Tersedia pada: <https://www.ti.com/lit/ds/symlink/lm2596.pdf>
- [37] “PRZ\_Dip02vr\_KAIFENG\_0001 Datasheet.”
- [38] “Push Button Arduino: Pengertian, Fungsi, dan Prinsip Kerja.” Diakses: 4 Maret 2025. [Daring]. Tersedia pada: <https://www.aldyrazor.com/2020/05/push-button-arduino.html>
- [39] E. Purnama, “Berkenalan Dengan Teknologi MQTT,” Pujangga Teknologi. Diakses: 4 Maret 2025. [Daring]. Tersedia pada: <https://medium.com/pujanggateknologi/berkenalan-dengan-teknologi-mqtt-7e63cab9d00d>
- [40] F. Ferdous, “What is a Server?,” TechMormo. Diakses: 4 Maret 2025. [Daring]. Tersedia pada: <https://medium.com/techmormo/what-is-a-server-1f8591ffd1a1>
- [41] “Linux—the Operating System,” *J. Adv. Shell Program.*, vol. 7, no. 3, hlm. 1–5, Mar 2022.
- [42] J. Sandell, E. Asplund, W. Y. Ayele, dan M. Duneld, “Performance Comparison Analysis of ArangoDB, MySQL, and Neo4j: An Experimental Study of Querying Connected Data”.
- [43] “Architecture of MySQL,” GeeksforGeeks. Diakses: 16 Maret 2025. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/architecture-of-mysql/>

- [44] H. L. Walingkas dan P. O. N. Saian, “Penerapan Framework Flask pada Pembangunan Sistem Informasi Pemasok Barang,” *J. JTIK J. Teknol. Inf. Dan Komun.*, vol. 7, no. 2, hlm. 227–234, Apr 2023, doi: 10.35870/jtik.v7i2.729.
- [45] M. Siwach dan S. Mann, “A Compendium of Various Applications of Machine Learning,” vol. 09, no. 07, 2022.
- [46] L. Younes, “Introduction to Machine Learning,” 4 September 2024, *arXiv: arXiv:2409.02668*. doi: 10.48550/arXiv.2409.02668.
- [47] Q. A. Hidayaturrohman dan E. Hanada, “Impact of Data Pre-Processing Techniques on XGBoost Model Performance for Predicting All-Cause Readmission and Mortality Among Patients with Heart Failure,” *BioMedInformatics*, vol. 4, no. 4, hlm. 2201–2212, Nov 2024, doi: 10.3390/biomedinformatics4040118.
- [48] T. Chen dan C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” dalam *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Agu 2016, hlm. 785–794. doi: 10.1145/2939672.2939785.
- [49] A. März, “Multi-Target XGBoostLSS Regression,” 13 Oktober 2022, *arXiv: arXiv:2210.06831*. doi: 10.48550/arXiv.2210.06831.
- [50] B.-E. A. Abdelghany, A. Fernández-Vilas, M. Fernández-Veiga, N. El-Bendary, A. M. Hassan, dan W. M. Abdelmoez, “Scheduling and Communication Schemes for Decentralized Federated Learning,” dalam *2022 32nd International Conference on Computer Theory and Applications (ICCTA)*, Des 2022, hlm. 122–128. doi: 10.1109/ICCTA58027.2022.10206255.
- [51] “What’s the Hardware Spec for Google Colaboratory | Saturn Cloud Blog.” Diakses: 16 Maret 2025. [Daring]. Tersedia pada: <https://saturncloud.io/blog/whats-the-hardware-spec-for-google-colaboratory/>
- [52] A. J. Dhruv, R. Patel, dan N. Doshi, “Python: The Most Advanced Programming Language for Computer Science Applications;,” dalam *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies*, Medan, Indonesia: SCITEPRESS - Science and Technology Publications, 2020, hlm. 292–299. doi: 10.5220/0010307902920299.
- [53] F. Pedregosa *dkk.*, “Scikit-learn: Machine Learning in Python,” 5 Juni 2018, *arXiv: arXiv:1201.0490*. doi: 10.48550/arXiv.1201.0490.
- [54] “Documentation for Visual Studio Code.” Diakses: 4 Maret 2025. [Daring]. Tersedia pada: <https://code.visualstudio.com/docs>
- [55] “The Arduino IDE.” Diakses: 16 Maret 2025. [Daring]. Tersedia pada: <https://se.ewi.tudelft.nl/desosa2019/chapters/arduino-ide/>