

**SISTEM PENGINDERAAN TERINTEGRASI UNTUK PEMANTAUAN  
KONDISI LAHAN TANAM DAN OTOMATISASI PENJADWALAN  
PENYIRAMAN TANAMAN MENGGUNAKAN ALGORITMA  
*LONG SHORT-TERM MEMORY***

**SKRIPSI**

**ALI HIDAYAT**

**171402019**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2024**

SISTEM PENGINDERAAN TERINTEGRASI UNTUK PEMANTAUAN  
KONDISI LAHAN TANAM DAN OTOMATISASI PENJADWALAN  
PENYIRAMAN TANAMAN MENGGUNAKAN ALGORITMA  
LONG SHORT-TERM MEMORY

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Teknologi Informasi

ALI HIDAYAT

171402019



PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024

**PERSETUJUAN**

Judul : SISTEM PENGINDERAAN TERINTEGRASI  
UNTUK PEMANTAUAN KONDISI LAHAN  
TANAM DAN OTOMATISASI PENJADWALAN  
PENYIRAMAN TANAMAN MENGGUNAKAN  
ALGORITMA LONG SHORT-TERM MEMORY

Kategori : SKRIPSI

Nama : ALI HIDAYAT

Nomor Induk Mahasiswa : 171402019

Program Studi : SARJANA (S1) TEKNOLOGI INFORMASI

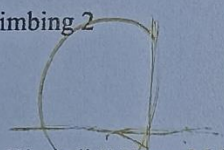
Departemen : TEKNOLOGI INFORMASI

Fakultas : FAKULTAS ILMU KOMPUTER DAN  
TEKNOLOGI INFORMASI

Medan, 11 Juli 2024

Komisi Pembimbing:

Pembimbing 2



Ainul Hizriadi S.Kom, M.Sc  
NIP. 198510272017061001

Pembimbing 1



Baihaqi Siregar S.Si, MT.  
NIP. 197901082012121002

Diketahui/Disetujui oleh

Program Studi S-1 Teknologi Informasi

Dedy Arisandi S.T, M.Kom.  
NIP. 197908312009121002



**PERNYATAAN**

**SISTEM PENGINDERAAN TERINTEGRASI UNTUK PEMANTAUAN KONDISI  
LAHAN TANAM DAN OTOMATISASI PENJADWALAN PENYIRAMAN  
TANAMAN MENGGUNAKAN ALGORITMA  
LONG SHORT-TERM MEMORY**

**SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Juli 2024



Ali Hidayat

171402067

## UCAPAN TERIMA KASIH

Alhamdulillahirabbil'alamin, puji syukur saya ucapkan atas kehardirat Allah SWT. Atas berkah dan rahmat-Nya, skripsi dengan judul “Sistem Penginderaan Terintegrasi Untuk Pemantauan Pertumbuhan Pada Budidaya Tanaman” telah selesai dikerjakan. Skripsi ini merupakan salah satu syarat memperoleh gelar sarjana Komputer di Program Studi S-1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Dalam proses pengerjaan skripsi ini banyak sekali bimbingan, arahan, serta dukungan maupun doa dari berbagai pihak disekitar penulis. Untuk itu penulis ingin mengucapkan terima kasih kepada seluruh pihak yang turut memberikan dukungan kepada penulis baik secara langsung maupun tidak langsung, antara lain :

1. Kedua orangtua penulis yaitu Suparno dan Nia br. Sitorus yang tak henti-hentinya memberikan dukungan, semangat dan doa, abang dan kakak yang selalu mendorong, memberikan semangat dan perhatian kepada penulis.
2. Bapak Baihaqi Siregar S.Si,MT. selaku Dosen Pembimbing I dan Bapak Ainul Hizriadi S.Kom, M.Sc. selaku Dosen Pembimbing II yang selalu memberikan bimbingan, arahan, dan saran yang banyak sekali selama proses pengerjaan skripsi.
3. Ibu Dr. Maya Silvi Lydia B.Sc. M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Dedy Arisandi S.T., M.Kom. selaku Ketua Program Studi S-1 Teknologi Informasi Univerista Sumatera Utara.
5. Seluruh dosen dan staff pegawai Program Studi S-1 Ilmu Komputer Fakultas IlmuKomputer dan Teknologi Informasi Universitas Sumatera Utara.
6. Teman-teman seperjuangan yaitu M. Bayhaqi Daulay, Jackie Chandra, Nurul Andini, Lisa Ayuning Tias, Ihsan Maulana, dan Fahmi Rizal yang telah membantupenulis untuk tak putus semangat dan terus mendorong penulis untuk

menyelesaikan skripsi ini.

7. Serta semua pihak yang terkait dan terlibat yang telah membantu penulis yang tidakbisa disebutkan semua namanya.

Penulis sadar bahwa masih banyak kesalahan dan kekurangan pada penulisan skripsi ini, sehingga penulis mengharapkan kritik dan saran yang membangun demi mengkoreksi dan menyempurnakan penulisan skripsi. Sekali lagi penulis ucapkan terima kasih atas dukungannya, semoga hal yang baik dibalas dengan kebaikan pula yang berlipat-lipat ganda.

Medan, 11 Juli 2024

A handwritten signature in blue ink, featuring a large, stylized initial 'A' followed by a series of loops and a final horizontal stroke.

Ali Hidayat

171402019

## ABSTRAK

Dengan perkembangan teknologi saat ini manusia dapat memanfaatkan teknologi sebagai asisten dalam setiap hal, termasuk dalam pertanian. Pelaku pertanian yakni petani dapat memanfaatkan teknologi sebagai asisten dalam tahap perawatan tanaman seperti menggunakan robot pertanian. Robot pertanian bertujuan untuk membantu petani merawat tanaman seperti memantau keadaan lahan tanam dan tentunya dapat bekerja secara otomatis. Namun otomatisasi ini memerlukan optimasi seperti optimasi daya dengan membatasi ataupun menjadwalkan kapan robot melakukan tugas rutin, salah satunya adalah tugas penyiraman tanaman. Demi tujuan robot pertanian ini tercapai, penulis merancang sistem penginderaan terintegrasi yang bertujuan untuk memantau kondisi lahan tanam dan otomatisasi penjadwalan penyiraman tanaman dengan memanfaatkan teknologi Pembelajaran Mesin menggunakan algoritma *Long Short-Term Memory (LSTM)* dengan memprediksi tingkat kelembaban tanah. Model prediksi ini juga memanfaatkan prakiraan nilai kondisi lingkungan dari sebuah penyedia data cuaca dan geolokasi Antarmuka Pemrograman Aplikasi (API) sebagai input prediksi. Model data yang digunakan adalah hasil penginderaan tanah berupa nilai kelembaban dan temperatur tanah, suhu udara, dan waktu yang dikumpulkan selama tiga bulan yang kemudian dilatih menggunakan algoritma LSTM dengan *hyperparameter* layer units 416, learning rate 0,001, dan rasio pemisahan data *training* dan testing 70:30. Hasil dari prediksi berupa nilai kelembaban tanah akan dijadikan acuan dalam menentukan penjadwalan penyiraman. Prediksi nilai kelembaban tanah memiliki rata-rata nilai prediksi yang menyimpang (deviasi) berkisar 5,13% ~ 5,60%. Hasil akhir dari penelitian ini adalah terwujudnya robot pertanian yang memiliki kemampuan penginderaan lahan tanam yang terintegrasi dengan sistem penjadwalan penyiraman otomatis dan dapat dipantau melalui aplikasi dasbor berbasis web.

Kata kunci: robot pertanian, sistem penginderaan terintegrasi, pemantauan, penjadwalan penyiraman, aplikasi dasbor, *long short-term memory*

**INTEGRATED SENSING SYSTEM FOR MONITORING THE CONDITION  
OF PLANTING LAND AND AUTOMATING PLANT WATERING  
SCHEDULES USING LONG SHORT-TERM MEMORY**

**ALGORITHM**

**ABSTRACT**

With current technological advancements, humans can utilize technology as an assistant in various fields, including agriculture. Agricultural practitioners, specifically farmers, can leverage technology as an assistant in the crop care stages, such as using agricultural robots. Agricultural robots aim to assist farmers in caring for crops by monitoring the condition of the planting area and working automatically. However, this automation requires optimization, such as power optimization by limiting or scheduling when the robot performs routine tasks, one of which is watering crops. To achieve this goal, the author designed an integrated sensing system to monitor the condition of the planting area and automate the watering schedule by utilizing Machine Learning technology using the Long Short-Term Memory (LSTM) algorithm to predict soil moisture levels. This prediction model also leverages forecasted environmental condition values from a weather data provider and geolocation Application Programming Interface (API) as input for predictions. The data model used consists of soil sensing results, including soil moisture and temperature values, air temperature, and time, collected over three months and then trained using the LSTM algorithm with hyperparameters of 416 layer units, learning rate of 0.001, and training and testing data split ratio of 70:30. The results of the prediction, in the form of soil moisture values, will serve as a reference in determining the watering schedule. The predicted soil moisture values have an average prediction deviation ranging from 5.13% to 5.60%. The final result of this research is the realization of an agricultural robot capable of integrated sensing of the planting area with an automated watering scheduling system that can be monitored through a web-based dashboard application.

**Keywords:** agricultural robot, integrated sensing system, monitoring, watering scheduling, dashboard application, long short-term memory



## DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	4
BAB 2 LANDASAN TEORI	
2.1 Robot Pertanian	6
2.2 Arduino Mega 2560	6
2.3 NodeMCU ESP8266 ESP-12E	7
2.4 RS485-LoRa Wireless Station	8
2.5 Dragino LoRa Shield	9
2.6 RS485 Soil NPK PH EC Sensor	9
2.7 Sensor BMP280	10
2.8 Framework Yii2	11

2.9	Python Flask	11
2.10	WeatherAPI	11
2.11	LoRa	11
2.12	MQTT (Message Queuing Telemetry Transport)	12
2.13	Long Short-Term Memory (LSTM)	13
2.14	Penelitian Terdahulu	13
<b>BAB 3 ANALISIS DAN PERANCANGAN SISTEM</b>		
3.1	Analisis Sistem	16
3.1.1	Arsitektur Umum	16
3.1.1.1	<i>Input</i>	16
3.1.1.2	<i>Proses</i>	16
3.1.1.3	<i>Output</i>	17
3.1.2	Penjadwalan Penyiraman dengan Memprediksi Kelembaban Tanah	19
3.2	Perancangan Antarmuka Sistem	20
3.2.1	<i>Use Case Diagram</i>	20
3.2.2	Data yang Digunakan	21
3.2.3	Flowchart Sistem	22
3.2.3.1	<i>Flowchart Penjadwalan Penyiraman</i>	22
3.2.3.2	<i>Flowchart Pelatihan Model LSTM</i>	23
3.2.3.3	<i>Flowchart Aplikasi Prediksi Flask</i>	32
3.2.4	Komponen Alat	34
3.2.5	Perancangan Sistem Penginderaan	35
3.2.6	Perancangan Robot Pertanian	35
3.2.7	Ilustrasi Bentuk Antarmuka Aplikasi	36
3.2.7.1	<i>Ilustrasi Tampilan Home</i>	37
3.2.7.2	<i>Ilustrasi Tampilan Halaman Dashboard</i>	37
3.2.7.3	<i>Ilustrasi Tampilan Halaman Prediction</i>	38
<b>BAB 4 IMPLEMENTASI DAN PENGUJIAN</b>		
4.1	Spesifikasi Perangkat Keras dan Perangkat Lunak	40
4.1.1	Perangkat Keras Pada Sistem	40

	<i>4.4.1.1 Perangkat Lunak Pada Sistem</i>	40
4.2	Implementasi Ilustrasi Tampilan Aplikasi Web	40
	4.2.1 Implementasi Ilustrasi Tampilan Home	40
	4.2.2 Implementasi Ilustrasi Tampilan Dashboard	41
	4.2.3 Implementasi Ilustrasi Tampilan Prediction	42
4.3	Implementasi Perancangan Robot Pertanian	42
4.4	Implementasi Perancangan Sistem Penginderaan	43
	4.4.1 Implementasi Rangkaian Sensor Tanah	43
	4.4.2 Implementasi Rangkaian Sensor Udara	45
	4.4.3 Implementasi LoRa Gateway	46
4.5	Pengujian Sistem	48
	4.5.1 Pengujian Sistem Penginderaan	48
	4.5.1.1 <i>Pengujian Rangkaian Sensor Tanah</i>	48
	4.5.1.2 <i>Pengujian Skema Penyimpanan Data</i>	56
	4.5.1.3 <i>Pengujian Pengambilan Data selama 24 Jam</i>	62
	4.5.2 Pengujian Sistem Penjadwalan Penyiraman	64
	4.5.2.1 <i>Pengujian Model Prediksi Long Short-Term Memory</i>	65
	4.5.2.2 <i>Pengujian Aplikasi Prediksi Flask</i>	70
	4.5.3 Pengujian Akurasi Model LSTM	74
	4.5.3.1 <i>Pengujian Akurasi pada LSTM</i>	75
	4.5.4 Pengujian Sistem Penjadwalan Penyiraman	77
	4.5.4.1 <i>Pengecekan Ketersediaan Data Jadwal Penyiraman</i>	78
BAB 5 KESIMPULAN		
5.1	Kesimpulan	84
5.2	Saran	84
DAFTAR PUSTAKA		86

## DAFTAR TABEL

Tabel 2. 1	Spesifikasi Arduino Mega 2560	7
Tabel 2. 2	Spesifikasi Raspberry Pi 4	8
Tabel 2. 3	Spesifikasi RS485-LoRa Wireless station	9
Tabel 2. 4	Spesifikasi Sensor RS485 Soil NPK pH EC	10
Tabel 2. 5	Penelitian Terdahulu	14
Tabel 3. 1	Contoh Penentuan Penjadwalan Penyiraman	19
Tabel 3. 2	Pemetaan Tingkat Kelembaban Tanah	20
Tabel 3. 3	Penjelasan <i>Use Case Diagram</i>	21
Tabel 3. 4	Contoh Data Sebelum Interpolasi	25
Tabel 3. 5	Contoh Data Sesuah Interpolasi	25
Tabel 3. 6	Contoh Data Sebelum Forward Fill	26
Tabel 3. 7	Contoh Data Sesudah Forward Fill	26
Tabel 3. 8	Komponen Alat	34
Tabel 3. 9	Rincian Alat dan Bahan Penyusun Robot Pertanian	35
Tabel 4. 1	Perangkat Lunak	40
Tabel 4. 2	Penjelasan Antarmuka RS485 LoRa Wireless Station	44
Tabel 4. 3	Komunikasi Antarmuka 12C ESP-12E dengan BMP280	46
Tabel 4. 4	Parameter Model Builder Function	66
Tabel 4. 5	Hasil Pengujian Hyperparameter Keras Tuner pada Model	66
Tabel 4. 6	Hasil Evaluasi Model dengan Skenario 1	68
Tabel 4. 7	Hasil Evaluasi Model dengan Skenario 2	68
Tabel 4. 8	Hasil Evaluasi Model dengan Skenario 3	69
Tabel 4. 9	Kumpulan Data Hasil Percobaan Dari Ketiga Skenario	70
Tabel 4. 10	Pengambilan Data Kelembaban dan Suhu Terbaru melalui API	71
Tabel 4. 11	Data Sampel Hari 1	74
Tabel 4. 12	Data Sampel Hari 2	75
Tabel 4. 13	Data Sampel Hari 3	75
Tabel 4. 14	Akurasi Model LSTM dengan Data Sampel Hari 1	75

Tabel 4. 15	Akurasi Model LSTM dengan Data Sampel Hari 2	76
Tabel 4. 16	Akurasi Model LSTM dengan Data Sampel Hari 3	76

## DAFTAR GAMBAR

Gambar 2. 1	Robot Pertanian	6
Gambar 2. 2	Arduino Uno R3	6
Gambar 2. 3	NodeMCU ESP8266 ESP-12E	7
Gambar 2. 4	RS485-LoRa Wireless station	8
Gambar 2. 5	Dragino LoRa Shield v1.4	9
Gambar 2. 6	RS485 Soil NPK PH EC Sensor	10
Gambar 2. 7	Sensor BMP280	10
Gambar 2. 8	Skema LoRa Singla Channel	12
Gambar 2. 9	Skema MQTT	12
Gambar 2. 10	Jaringan LSTM	13
Gambar 2. 11	Struktur LSTM	13
Gambar 3. 1	Arsitektur Umum	18
Gambar 3. 2	Use Case Diagram	20
Gambar 3. 3	Data yang Digunakan	22
Gambar 3. 4	Flowchart Penjadwalan Penyiraman	23
Gambar 3. 5	Flowchart Pre-processing Data	27
Gambar 3. 6	The Vanishing Gradient Problem pada RNN	29
Gambar 3. 7	Flowchart Pelatihan Model Long Short-Term Memory	32
Gambar 3. 8	Flowchart Aplikasi Prediksi Flask	34
Gambar 3. 9	Rancangan Desain Robot Pertanian	36
Gambar 3. 10	Ilustrasi Tampilan Halaman Home	37
Gambar 3. 11	Ilustrasi Tampilan Beranda	38
Gambar 3. 12	Ilustrasi Tampilan Prediksi	39
Gambar 3. 13	Implementasi Tampilan Home	41
Gambar 4. 1	Implementasi Tampilan Dashboard	41
Gambar 4. 2	Implementasi Halaman Prediksi	42
Gambar 4. 3	Implementasi Robot Pertanian	42
Gambar 4. 4	Skema Rangkaian Sensor	43

Gambar 4. 5	Konfigurasi Umum Jaringan RS485	44
Gambar 4. 6	Antarmuka RS485 LoRa Wireless Station	44
Gambar 4. 7	Komponen - komponen Penyusun RS485-LoRa Wireless Station	45
Gambar 4. 8	Implementasi Rangkaian Sensor Tanah	45
Gambar 4. 9	Hubungan Komunikasi Antarmuka 12C ESP-12E dengan BMP280	46
Gambar 4. 10	Sensor Udara	46
Gambar 4. 11	Rancangan LoRa Gateway	47
Gambar 4. 12	Implementasi LoRa Gateway	47
Gambar 4. 13	Penginderaan Tanah pada Pot	48
Gambar 4. 14	Aktivasi dan Proses pada Board RS485-LoRa Wireless Station	52
Gambar 4. 15	Aktivasi dan Proses pada Board ESP-12E	55
Gambar 4. 16	Aktivasi dan Proses pada LoRa Gateway	56
Gambar 4. 17	Arsitektur Umum Bagian Proses	57
Gambar 4. 18	Flowchart Skema Penyimpanan Data	58
Gambar 4. 19	Pengujian Penginderaan Sensor Tanah	59
Gambar 4. 20	Pengujian Penginderaan Sensor Udara	59
Gambar 4. 21	Pengujian LoRa Gateway	60
Gambar 4. 22	Contoh Hasil Identifikasi dan Proses Data	62
Gambar 4. 23	Data yang Berhasil Disimpan pada Basis Data	62
Gambar 4. 24	Pengujian Pengambilan Data Tanah	63
Gambar 4. 25	Pengujian Pengambilan Data Udara	63
Gambar 4. 26	Flowchart Sistem Penjadwalan Penyiraman	64
Gambar 4. 27	Perubahan Training dan Validation Loss Trial ke 7	67
Gambar 4. 28	Visualisasi Prediksi pada Skenario 1	68
Gambar 4. 29	Visualisasi Prediksi pada Skenario 2	69
Gambar 4. 30	Visualisasi Prediksi pada Skenario 3	69
Gambar 4. 31	Hasil Prediksi dalam bentuk JSON	74
Gambar 4. 32	Flowchart Pengujian Penjadwalan Penyiraman	78
Gambar 4. 33	Pesan dari Endpoint gateway	79
Gambar 4. 34	Pesan Pada Aplikasi Flask	80
Gambar 4. 35	Tampilan Tidak Ada Data pada Basis Data Lokal	80
Gambar 4. 36	Tampilan Tidak Ada Data pada basis	80

Gambar 4. 37	Tampilan Hasil Prediksi	81
Gambar 4. 38	Log Penerimaan Data	82
Gambar 4. 39	Data Jadwal Penyiraman Berhasil Disimpan	83



## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Proses budidaya dan produksi pertanian memiliki faktor yang kompleks, beragam, dan unik untuk setiap jenis tanaman. Faktor-faktor penting yang mempengaruhi proses budidaya dan produksi pertanian yang menjadi tantangan seperti karakteristik dan kebutuhan tanaman, lingkungan geografis dan iklim, maupun permintaan pasar dan kesanggupan pelaku pertanian menjadi poin penting dalam industri pertanian (Bechar & Vigneault 2016). Sumber informasi tentang kebutuhan tanaman, lahan tanam, maupun lingkungan menjadi salah satu fokus dalam hal ini. Pengumpulan data terhadap lingkungan lahan pertanian adalah hal yang paling memakan waktu dan membutuhkan biaya yang mahal dan menjadi hambatan besar untuk penerapannya bagi pelaku produksi pertanian (Conesa-Muñoz *et al.*, 2015).

Industri pertanian pada umumnya dilakukan oleh sekelompok orang dan pada lahan yang luas. Namun dengan perkembangan industri pertanian saat ini, pertanian dapat dilakukan dengan skala yang lebih kecil dan dapat dilakukan oleh perorangan, contohnya dengan metode pertanian *Green house* atau rumah kaca. Saputra *et al.* (2023) merancang bangun sebuah sistem monitoring *smart green house* yang mampu memantau kondisi lingkungan dan tumbuhan pada rumah kaca. Rumah kaca merupakan wujud teknologi pertanian berupa bangunan tembus pandang yang memiliki fungsi menjaga tanaman dari faktor-faktor yang mempengaruhi pertumbuhan tanaman seperti kondisi lingkungan tertentu, hama, maupun faktor eksternal lainnya agar terjaga kualitas pertumbuhan dan hasil tanaman. Teknologi pertanian ini memanfaatkan teknologi *Internet of Things (IoT)* sebagai instrument sistem pemantauan pertanian. Namun pada penelitian mereka, sistem penginderaan lingkungan rumah kaca tidak terintegrasi oleh aplikasi dan data hanya dapat ditampilkan dalam bentuk tabel pada basis data dan data yang dikumpulkan hanya disimpan dan tidak diolah untuk mendapatkan informasi yang dapat digunakan untuk mendukung penelitian.

Penelitian lain yaitu oleh Rahman *et al.* (2023) merancang sebuah sistem monitoring kondisi lingkungan berbasis *Wireless Sensor Network* (WSN). Penelitian ini berfokus pada pengaplikasian WSN pada sistem pertanian hidroponik sebagai instrument penginderaan yang memantau kondisi media tanam dan perancangan pengisian nutrisi otomatis.

Kemudian Griva *et al.* (2023) melakukan penelitian tentang kemampuan LoRa (*Long Range*) dalam mentransmisikan data jarak jauh dan hemat daya. Dengan pemanfaatan teknologi LoRa ini diharapkan sistem bisa bekerja jika sistem dibangun pada jarak yang jauh dari sumber internet maupun jika sistem dibangun pada daerah terpencil yang minim sinyal internet dan daya.

Lalu Iman & Wulandari (2023) melakukan penelitian prediksi harga saham menggunakan Metode Long Short-Term Memory. Dalam penelitian mereka, mereka memanfaatkan data perkembangan saham pada PT Telkom Indonesia untuk kemudian diprediksi agar hasilnya dapat dimanfaatkan oleh investor untuk mengetahui prediksi saham di masa depan.

Dari penelitian-penelitian yang telah dilakukan sebelumnya, penulis mengembangkan ide dan hasil dari penelitian yang sudah dilakukan yaitu merancang sebuah sistem pertanian terkendali berupa budidaya pertanian dalam ruangan (*indoor*) seperti rumah kaca yang memiliki sistem penginderaan terintegrasi yang dapat memantau kondisi lingkungan secara *real-time* dan dapat dipantau menggunakan aplikasi berbasis web. Sistem penginderaan ini selain memanfaatkan teknologi *IoT* juga memanfaatkan teknologi *LoRa* agar sistem dapat beradaptasi jika harus diterapkan pada daerah terpencil. Data hasil pemantauan yang dikumpulkan akan dikelola dengan memanfaatkan teknologi pembelajaran mesin (*Machine Learning*) dengan melatih data menggunakan metode *Long Short-Term Memory (LSTM)* untuk menemukan informasi pola dari tingkat kelembaban tanah dan menggunakan informasi tersebut untuk menentukan penjadwalan penyiraman tanaman. Dari ide-ide tersebut penulis mengajukan sebuah penelitian dengan judul *SISTEM PENGINDERAAN TERINTEGRASI UNTUK PEMANTAUAN KONDISI LAHAN TANAM DAN OTOMATISASI PENJADWALAN PENYIRAMAN TANAMAN MENGGUNAKAN ALGORITMA LONG SHORT-TERM MEMORY*.

## 1.2 Rumusan Masalah

Praktik pertanian tidak terbatas pada pengolahan lahan dan tanaman di area yang luas, tetapi dapat juga dilakukan pada lahan yang terbatas. Potensi ini mendorong setiap orang untuk melakukan praktik pertanian perorangan di lahan terbatas yang tentu dapat menjadi sumber ekonomi tambahan. Meskipun demikian, praktik pertanian tidaklah mudah. Terdapat beberapa tahap tanam seperti penumbuhan bibit dan pemindahan bibit ke lahan tanam, dan pemeliharaan setelah tanam sampai masa panen tiba. Tahap pemeliharaan tanaman membutuhkan waktu dan perhatian terhadap pemberian nutrisi agar tanaman dapat tumbuh dengan baik. Namun hal ini sulit dilakukan karena kesibukan masyarakat yang beragam sehingga tanaman tidak mendapatkan nutrisi yang diperlukan tepat waktu. Sementara tanaman harus dipantau kebutuhannya seperti air dan unsur hara lainnya secara optimal sehingga tanaman dapat tumbuh dengan maksimal.

## 1.3 Tujuan Penelitian

Tujuan penelitian ini adalah membangun sistem penginderaan terintegrasi untuk memonitor kondisi lahan tanam secara *real-time* dan mengatur tingkat kelembaban tanah dengan memprediksi penjadwalan penyiraman dalam kurun waktu 24 jam menggunakan algoritma *Long Short-Term Memory*.

## 1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Penelitian ini dilakukan pada lahan tanam seluas 1,8 x 4 m<sup>2</sup> tertutup plastic UV.
2. Pemantauan lahan tanam dilakukan menggunakan rangkaian sensor yang hanya melakukan penginderaan tingkat kelembaban, pH, suhu, dan unsur hara (NPK) tanah.
3. Prediksi penjadwalan penyiraman tanaman hanya memprediksi tingkat kelembaban tanah.
4. Aplikasi pemantauan kondisi lahan tanam berbasis web.
5. Skenario pengumpulan data hanya ada 2, yaitu dalam kurun waktu 3 bulan dengan pengambilan sampel data perjam, dan dalam kurun waktu 24 jam dengan pengambilan sampel data permenit.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan memberi manfaat kepada pelaku pertanian dalam menciptakan *smart farming precision agriculture* atau pertanian pintar yang memanfaatkan teknologi dan bermanfaat dalam mempermudah kegiatan pertanian.

## 1.6 Metodologi Penelitian

Metodologi penelitian memiliki beberapa tahapan yaitu :

### 1. Studi Literatur

Tahap pertama adalah studi literatur, di mana informasi dikumpulkan dari berbagai sumber, seperti buku, skripsi, dan jurnal ilmiah. Data dan informasi memiliki keterkaitan dengan penelitian.

### 2. Analisis Permasalahan

Pada tahap ini menganalisa permasalahan yang ada pada tahap pertama. Tahap ini bertujuan untuk mendapatkan sebuah metode yang akan menjadi acuan dalam menyelesaikan masalah pada penelitian.

### 3. Perencanaan Sistem

Tahap ini dilakukan berdasarkan kesimpulan dari analisis permasalahan yang kemudian dijadikan perencanaan sistem.

### 4. Hasil dan Kesimpulan serta Dokumentasi

Pada tahap terakhir ini melakukan dokumentasi dari perencanaan sistem, hasil penelitian, dan kesimpulan dari penelitian.

## 1.7 Sistematika Penulisan

Sistematika penulisan memiliki 5 bagian yaitu :

### Bab 1: Pendahuluan

Bab 1 yaitu pendahuluan adalah bab yang berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

### Bab 2: Landasan Teori

Bab 2 yaitu landasan teori adalah bab yang membahas tentang teori pendukung penelitian, seperti metode *long short-term memory*, perangkat lunak dan perangkat keras terkait dengan rangkaian sensor, protokol transmisi data *LoRa*, *API*, dan aplikasi berbasis web.

### Bab 3: Analisis dan Perancangan Sistem

Bab 3 yaitu analisis dan perancangan sistem adalah bab yang membahas analisis tentang penelitian yang dikerjakan dan perancangan sistem yang membahas rancangan sistem seperti arsitektur umum, antarmuka sistem maupun data pendukung.

#### **Bab 4: Implementasi dan Pengujian Sistem**

Bab 4 yaitu implementasi dan pengujian sistem adalah bab yang membahas tentang implementasi berupa ilustrasi tampilan sistem, pembahasan program, dan hasil pengujian sistem dari penelitian ini.

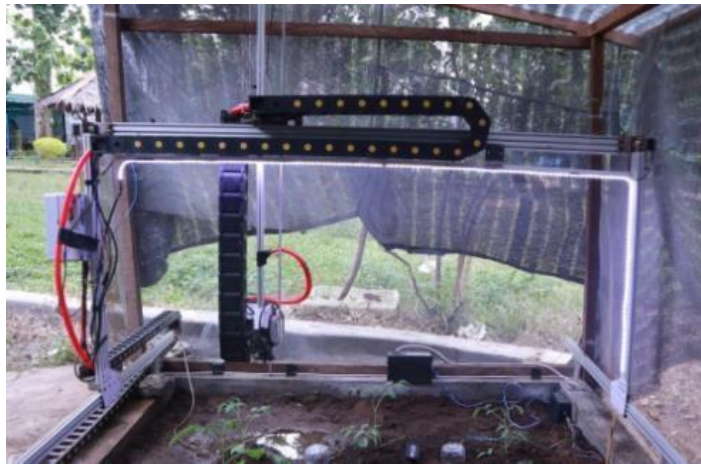
#### **Bab 5: Kesimpulan dan Saran**

Bab 5 yaitu Kesimpulan dan saran adalah bab yang membahas tentang kesimpulan dan intisari dari penelitian dan juga berisi beberapa saran untuk dibahas kembali di penelitian selanjutnya.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Robot Pertanian**



**Gambar 2. 1** Robot Pertanian

Pertanian modern dapat dipastikan dilengkapi dengan peralatan canggih yang bertujuan agar dapat mewujudkan segala tujuan yang ada pada saat menjalankan aktivitasnya. Peralatan yang menjadi pendukung aktivitas tersebut berupa sensor tanah dan tanaman, drone, penyiraman otomatis, dan lain sebagainya. Setiap perangkat memiliki tujuan dan keunggulannya tersendiri, contohnya termasuk robot pertanian atau agribot. Robot pertanian memiliki fungsi yang dapat bekerja secara otomatis maupun semi otomatis yang memudahkan manusia dalam aktivitas pertanian (Alkausar *et al.* 2022).

#### **2.2 Arduino Mega 2560**



**Gambar 2. 2** Arduino Uno R3

Arduino UNO R3 adalah sebuah produk unggulan dari Arduino yang memberi kemudahan untuk pemula yang baru memasuki dunia elektronik dan sebagai sarana Pendidikan (Arduino.cc). UNO board ini mudah digunakan dan cukup untuk kebutuhan proyek IoT yang ringkas.

Spesifikasi singkat dari Arduino Mega 2560 bisa dilihat pada Tabel 2.1.

**Tabel 2. 1** Spesifikasi Arduino Mega 2560

Spesifikasi	Keterangan
Mikrokontroller	ATMega328P
Tegangan Operasional	5V
Tegangan Input (direkomendasikan)	2.7-5.5 V
Tegangan Input (limit)	6-20V
Pin Digital I/O	10
Pin Analog Input	4
Arus DC per I/O	20 mA

### 2.3 NodeMCU ESP8266 ESP-12E



**Gambar 2. 3** NodeMCU ESP8266 ESP-12E

NodeMCU ESP8266 ESP-12E adalah modul terkini yang populer karena modul ini menggunakan modul WiFi yang dapat membantu board Arduino yang tidak memiliki modul WiFi dan dapat diaplikasikan dalam berbagai hal (protosupplies.com).

Spesifikasi NodeMCU ESP8266 ESP-12E bisa dilihat di Tabel 2.2.

**Tabel 2. 2** Spesifikasi Raspberry Pi 4

Spesifikasi	Keterangan
Mikrokontroler	ESP-8266 32-bit
Clock Speed	80 MHz
Konversi USB	CP2102
Sambungan USB	Micro USB
Tegangan Operasional	3.3V
Flash Memory	4 MB
Digital I/O	11
Input Analog	1
Komunikasi	Serial, SPI, I2C and 1-Wire menggunakan library perangkat
WiFi	Built-in 802.11 b/g/n

## 2.4 RS485-LoRa Wireless Station

**Gambar 2. 4** RS485-LoRa Wireless station

RS485-LoRa Wireless station adalah sebuah papan pengembangan (*development board*) yang sudah terpasang WiFi, Bluetooth, dan LoRa modul di dalamnya. Papan pengembangan ini dikembangkan oleh Makerfabs dengan tujuan untuk diaplikasikan pada proyek agrikultur. Papan pengembangan ini dapat disambungkan dengan sensor tanah yang berguna untuk penginderaan kondisi tanah.

Berikut spesifikasi dari RS485-LoRa Wireless station :



**Tabel 2. 3** Spesifikasi RS485-LoRa Wireless station

Spesifikasi	Keterangan
Mikrokontroler	ESP32 WROOM
Kompatibel Arduino	Ya
Modul lain yang terpasang	WiFi, Bluetooth, LoRa 915MHz
Rentang pengukuran	-40°C~80°C , 0~100% kelembaban
Akurasi pengukuran	±0.5°C, ±3% kelembaban
Jarak komunikasi	2 KM atau lebih
Daya yang diperlukan	5V

## 2.5 Dragino LoRa Shield

**Gambar 2. 5** Dragino LoRa Shield v1.4

Dragino LoRa Shield adalah sebuah transceiver (*transfer – receiver*) jarak jauh berbentuk perisai yang dipasang pada mikrokontroler Arduino UNO/Mega dengan modul LoRa terpasang padanya. Modul LoRa ini dapat digunakan untuk komunikasi data jarak jauh sebagai pengirim maupun penerima data dengan penggunaan daya yang rendah.

## 2.6 RS485 Soil NPK PH EC Sensor



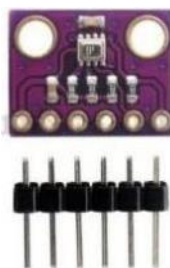
**Gambar 2. 6** RS485 Soil NPK PH EC Sensor

RS485 Soil NPK PH EC sensor adalah sensor penginderaan tanah yang menghasilkan data penginderaan suhu, kelembaban, PH, Nitrogen, Fosfor, dan Potasium pada tanah. Sensor ini dapat melakukan penginderaan tanah dengan cepat dan akurat. Berikut spesifikasi dari RS485 Soil NPK PH EC :

**Tabel 2. 4** Spesifikasi Sensor RS485 Soil NPK pH EC

Spesifikasi	Keterangan
Daya	5-30 VDC
Penggunaan daya maksimum	$\leq 0.15W$
Suhu operasional	-40~80°C
Jangkauan penginderaan	1-1999 mg/kg(mg/L)
NPK	Sesuai parameter yang ditetapkan
Sinyal output	RS485
Waktu responsif	$\leq 1$ detik
Presisi	$\pm 2\%FS$

## 2.7 Sensor BMP280

**Gambar 2. 7** Sensor BMP280

Sensor BMP280 adalah sensor yang berfungsi untuk mengukur suhu dan tekanan atmosferik. Modul sensor ini mendukung antarmuka SPI dan I2C. Dengan kedua antarmuka tersebut, modul sensor ini dapat dikalibrasi secara manual yang memungkinkan pengguna merubah nilai dari parameter yang ada.

Framework Yii2 adalah salah satu framework berbahasa pemrograman PHP populer yang memenuhi semua kunci fitur situs web modern dan pengembangan aplikasi web (Zulkifli, *et al.* 2020). Framework Yii2 juga mengklaim bahwa framework ini menjadi framework berbasis PHP tercepat dalam hal masalah kinerja. Framework ini dapat digunakan untuk membuat berbagai macam aplikasi berbasis web dengan bahasa

pemrograman PHP sebagai basisnya. Framework Yii2 adalah versi ke 2.0 dari Yii dan juga versi terbaru untuk saat ini.

## 2.8 Framework Yii2

Framework Yii2 adalah salah satu framework berbahasa pemrograman PHP populer yang memenuhi semua kunci fitur situs web modern dan pengembangan aplikasi web (Zulkifli, *et al.* 2020). Framework Yii2 juga mengklaim bahwa framework ini menjadi framework berbasis PHP tercepat dalam hal masalah kinerja. Framework ini dapat digunakan untuk membuat berbagai macam aplikasi berbasis web dengan bahasa pemrograman PHP sebagai basisnya. Framework Yii2 adalah versi ke 2.0 dari Yii dan juga versi terbaru untuk saat ini.

## 2.9 Python Flask

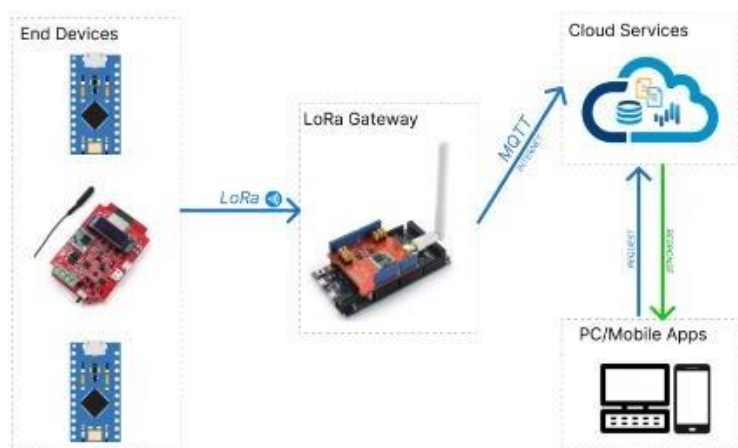
Python Flask adalah sebuah kerangka mikro WSGI dengan basis bahasa pemrograman Python yang menyediakan fungsional dasar dengan kemampuan menambahkan modul yang diperlukan sesuai kebutuhan pengguna. Karena sifatnya yang modular, pengguna dapat menambahkan modul/alat yang dapat digunakan dalam pengembangan proyek (Grinberg, 2018). Fleksibilitas arsitektur dan ketersediaan ekstensi yang besar dan siap pakai memungkinkan Flask digunakan untuk membangun proyek sederhana dan kompleks. Flask biasanya digunakan untuk membangun layanan RESTful API (Antarmuka Pemrograman Aplikasi) untuk pengembangan prototipe sebuah aplikasi (Kornienko, *et al.* 2021).

## 2.10 WeatherAPI

WeatherAPI adalah sebuah penyedia layanan data cuaca dan geografis dalam bentuk JSON/XML melalui RESTful API. Hal ini memungkinkan pengembang aplikasi untuk menggunakan layanan data cuaca dan geo untuk membuat aplikasi yang memanfaatkan data cuaca dan geo dengan pemantauan waktu nyata dengan mudah.

## 2.11 LoRa

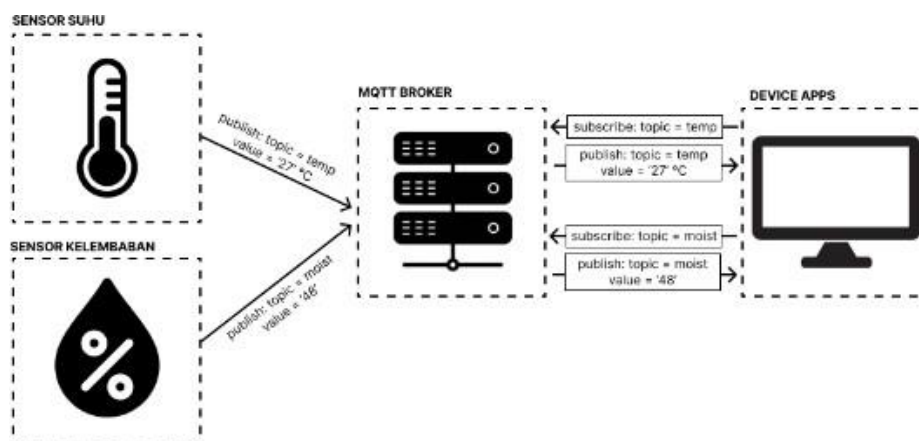
LoRa (Long Range) adalah teknik modulasi penyebaran spektrum dari Semtech (Bor & Roedig 2017). Teknik modulasi spektrum ini bernama *Chirp Spread Spectrum* (CSS). Modulasi ini memungkinkan LoRa mentransfer sinyal dengan jarak yang jauh. LoRa bekerja dengan memanfaatkan frekuensi radio tertentu yang memungkinkan komunikasi antar perangkat secara nirkabel. Kelebihan dari LoRa adalah memiliki jangkauan komunikasi yang luas namun dengan konsumsi daya yang rendah (Susanto, *et al.* 2019).



**Gambar 2. 8** Skema LoRa Singla Channel

## 2.12 MQTT (Message Queuing Telemetry Transport)

Protokol MQTT (Message Queue Telemetry Transport) adalah sebuah protokol pesan yang ringan (*lightweight*) berbasis *publish-subscribe* yang digunakan di atas protokol TCP/IP. Ciri dari protokol ini memiliki ukuran paket data yang kecil, *bandwidth* kecil, dan sifatnya terbuka (Wibowo & Zain 2021). Protokol ini menggunakan komunikasi *machine-to-machine* (M2M) dalam pengaplikasiannya pada *Internet of Things* (IoT). Pada praktiknya, MQTT memiliki 3 poin komunikasi yakni *publisher* (penerbit), *Broker* (penyebarkan), dan *subscriber* (pelanggan). Berikut skema MQTT ditampilkan pada Gambar 2 9 :



**Gambar 2. 9** Skema MQTT

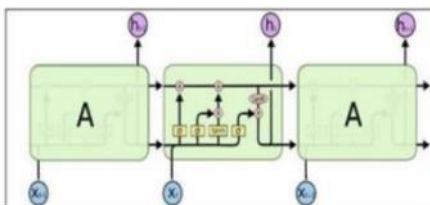
Dari Gambar 2 9, dapat dilihat Sensor Suhu dan Sensor Kelembaban adalah *publisher* yang menerbitkan nilai suhu (temp) dan kelembaban (moist) kepada *Broker* MQTT Broker. Kemudian MQTT Broker akan menyebarkan nilai suhu dan kelembaban kepada setiap *subscriber* yang berlangganan pada *topic* 'temp' dan 'moist'. Sedangkan *Topic*

atau topik adalah sebuah label atau identifikasi yang mengelompokkan pesan berdasarkan nama topik yang ditetapkan. Topik memungkinkan pesan yang diterbitkan dapat dikenali dan diterima secara spesifik.

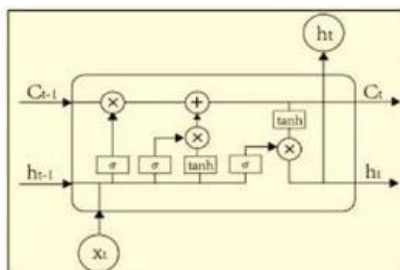
### 2.13 Long Short-Term Memory (LSTM)

*Long Short-Term Memory* (LSTM) adalah sebuah tipe atau jenis *Recurrent Neural Network* (RNN). LSTM menyimpan informasi tentang pola data dan memilih data mana yang akan digunakan pada proses selanjutnya dan data mana yang akan dibuang agar tidak ikut serta dalam tahap jaringan berikutnya. Hal ini dikarenakan neuron LSTM memiliki tiga jenis gerbang (gerbang masukan, gerbang lupa, dan gerbang keluaran) yang dapat diukur dan disusun ulang (Aldi *et al.* 2018).

LSTM merupakan pengembangan jaringan syaraf tiruan (*neural network*) yang dapat digunakan untuk memodelkan data deret waktu (*time-series*). LSTM dapat mengatasi ketergantungan input jangka panjang. Sel dalam LSTM menyimpan nilai atau status (status sel) untuk jangka waktu yang lama atau singkat. LSTM memiliki blok memori yang menentukan nilai mana yang dipilih sebagai keluaran relatif terhadap masukan tertentu. Hal ini merupakan keunggulan LSTM (Iman *et al.* 2023)



**Gambar 2. 10** Jaringan LSTM  
(Colah, 2015)



**Gambar 2. 11** Struktur LSTM  
(Iman *et al.* 2023)

### 2.14 Penelitian Terdahulu

Beberapa penelitian terkait sistem monitoring lahan pertanian dan prediksi menggunakan algoritma LSTM yang telah dilakukan sebelumnya dan memiliki ide dan hasil penelitian yang bersinggungan. Penelitian-penelitian tersebut dirangkum pada Tabel 2.5.

**Tabel 2. 5** Penelitian Terdahulu

No	Peneliti	Judul	Tahun	Keterangan
1	Roni Saputra, Muhammad Fadlan Siregar, Jhoni Hidayat	Rancang Bangun Sistem Monitoring Smart Green House Di Laboratorium Pertanian Universitas Tjut Nyak Dhien	2023	Merancang bangun sebuah sistem monitoring <i>Smart Green House</i> dengan tujuan memantau kondisi suhu dan kelembaban <i>Green House</i> .
2	Aswadul Fitri Saiful Rahman, Muhammad Satria Nugraha, Agung Nugraha	Perancangan dan Monitoring Sistem Pertanian Hidroponik Berbasis Wireless Sensor Network (WSN)	2023	Penelitian ini berfokus pada pengaplikasian <i>Wireless Sensor Network (WSN)</i> pada sistem pertanian hidroponik sebagai instrument penginderaan yang memantau kondisi media tanam dan perancangan pengisian nutrisi otomatis.
3	Wahyu Adi Prayitno, Adharul Muttaqin, Dahnial Syauqy	Sistem Monitoring Suhu, Kelembaban, dan Pengendali Penyiraman Tanaman Hidroponik menggunakan Blynk Android	2017	Mengembangkan sistem monitoring kondisi media tanam hidroponik dan pengendali penyiraman menggunakan aplikasi Blynk android.

**Tabel 2. 6** Penelitian Terdahulu ( Lanjutan )

No	Peneliti	Judul	Tahun	Keterangan
4	Aikaterini I. Griva, Achilles D. Boursianis, Shaohua Wan 2, Panagiotis Sarigiannidis, Konstantinos E. Psannis, George Karagiannidis, Sotirios K. Goudos	<i>LoRa-Based IoT Network Assessment in Rural and Urban Scenarios</i>	2023	Artikel ini membahas tentang dampak <i>Internet of Things (IoT)</i> pada jaringan pintar, menjadikan kemampuan jarak jauh dan hemat daya LoRa ( <i>Long Range</i> ) sebagai sorotannya.
5	Fiqih Nur Iman, Dena Wulandari	Prediksi Harga Saham Menggunakan Metode Long Short-Term Memory	2023	Memanfaatkan algoritma Long Short-Term Memory untuk memprediksi saham PT Telkom Indonesia agar memudahkan para investor dalam membuat keputusan untuk membeli atau menjual saham.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis Sistem**

##### **3.1.1 Arsitektur Umum**

Pada bagian ini penulis akan menjelaskan 3 tahap analisis sistem yaitu input (masukan), proses, dan output (keluaran) yang akan menyusun sistem yang dibuat.

##### *3.1.1.1 Input*

Input atau masukan pada penelitian ini adalah data kondisi tanah pada lahan tanam yang didapat melalui proses penginderaan lahan tanam.

Terdapat 2 macam penginderaan, yaitu penginderaan tanah dan penginderaan suhu ruangan :

1. Penginderaan tanah dilakukan menggunakan sensor RS485 Soil NPK PH EC yang terhubung dengan RS485-LoRa Wireless station. Hasil penginderaan akan diteruskan ke *Gateway* melalui protokol komunikasi LoRa. Setelah data sampai ke *Gateway*, data akan di-*publish* kepada MQTT *broker* dengan topik *farmbot/sensors*.
2. Penginderaan kondisi ruangan dilakukan oleh sensor BMP280 yang terhubung pada mikrokontroler ESP8266 ESP-12E. Mikrokontroler pertama-tama akan terhubung pada *access point* berupa WiFi router untuk mendapatkan akses ke internet, kemudian melakukan penginderaan terhadap udara untuk mendapatkan data, lalu data akan di-*publish* ke topik *farmbot/sensor* untuk kemudian disimpan ke basis data.

##### *3.1.1.2 Proses*

Bagian proses dimulai saat MQTT Broker menyebarkan data dari topik *farmbot/sensor* ke *subscriber* yang berlangganan ke topik tersebut. MQTT *client* yang terpasang pada *cloud server* sebagai *subscriber* akan menerima pesan dari topik dan kemudian memproses pesan tersebut. Di sini pesan berupa data dari hasil penginderaan akan ditangkap kemudian disimpan ke dalam basis data pada *cloud*



*server*. Data dari kedua penginderaan yang sama-sama di-*publish* ke topik *farmbot/sensors* akan diidentifikasi agar dapat dibedakan. Pada *cloud server*, akan mengidentifikasi pesan yang di-*subscribe* melalui panjang pesan. Data yang disimpan pada basis data kemudian dapat ditampilkan pada aplikasi berbasis web sebagai data *real-time* dan atau diambil untuk keperluan pelatihan data model prediksi. Pada bagian proses ini *cloud server* juga melakukan *request* data prakiraan cuaca kepada WeatherAPI. Data prakiraan cuaca ini akan dijadikan fitur masukan untuk prediksi tingkat kelembaban tanah nantinya.

#### 3.1.1.3 Output

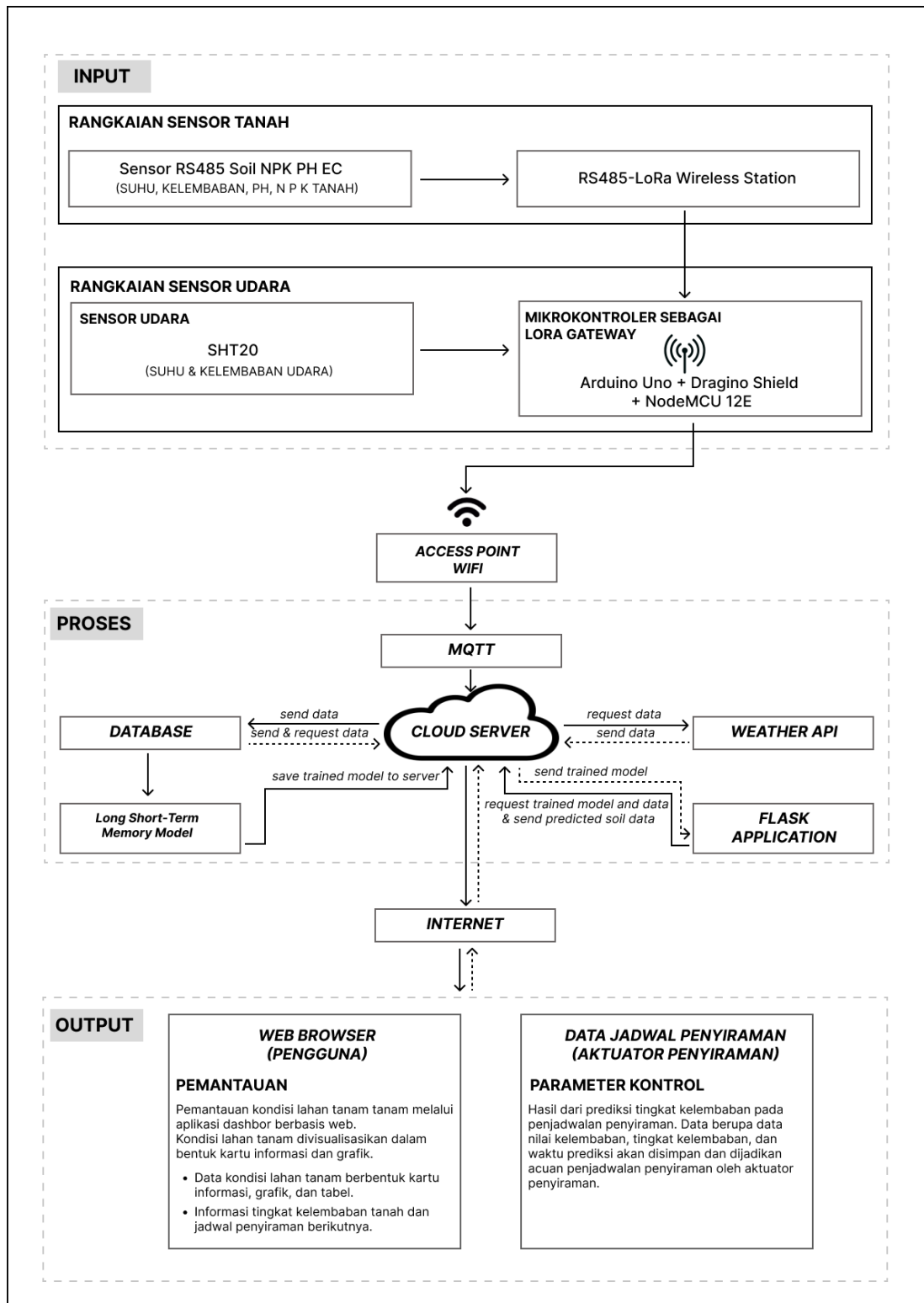
Pada tahapan ini ada 2 output fitur yang dapat bermanfaat untuk pengguna, yaitu pemantauan dan parameter kontrol :

##### 1. Pemantauan

Fitur ini memungkinkan pengguna untuk memantau data kondisi lingkungan lahan tanam secara *real-time* dan jadwal penyiraman berikutnya melalui aplikasi berbasis web.

##### 2. Parameter Kontrol

Hasil dari prediksi tingkat kelembaban pada penjadwalan penyiraman memberikan parameter kontrol untuk mengatur kapan robot pertanian melakukan aksi penyiraman.



Gambar 3. 1 Arsitektur Umum

### 3.1.2 Penjadwalan Penyiraman dengan Memprediksi Kelembaban Tanah

Pada tahap ini, data yang telah dikumpulkan digunakan untuk penentuan penjadwalan penyiraman. Penentu penjadwalan penyiraman adalah hasil prediksi tingkat kelembaban tanah dalam kurun waktu 24 jam. Sistem akan melakukan pengecekan nilai kelembaban tanah mulai pukul 00.00 hingga 23.00 dalam interval jam dengan memanfaatkan nilai kelembaban tanah terbaru hasil penginderaan dan data prakiraan cuaca suhu dan kelembaban lingkungan. Jika nilai prediksi kelembaban tanah menyentuh tingkat kelembaban ‘Basah’ dan ‘Tepat’ maka tidak dilakukan penjadwalan pada waktu tersebut. Jika nilai prediksi kelembaban tanah menyentuh tingkat kelembaban ‘Kering’ maka sistem akan menetapkan waktu tersebut sebagai jadwal penyiraman.

**Tabel 3. 1** Contoh Penentuan Penjadwalan Penyiraman

Waktu	Prediksi Nilai Kelembaban	Status Tingkat kelembaban
00.00	62.12	Basah
01.00	61.44	Basah
02.00	60.82	Basah
...	...	...
13.00	54.24	Tepat
14.00	53.85	Tepat
15.00	52.20	Tepat
...	...	...
17.00	46.74	Tepat
18.00	44.80	Kering
19.00	43.20	Kering
...	...	...
23.00	37.44	Kering

Dari Tabel 3.1 dapat dilihat contoh penjadwalan penyiraman berdasarkan prediksi tingkat kelembaban tanah dalam kurun waktu 24 jam. Tingkat kelembaban tanah didapat dengan memetakan nilai kelembaban tanah menjadi tiga tingkat. Berikut pemetaan tingkat kelembaban tanah disajikan pada Tabel 3.2

**Tabel 3. 2** Pemetaan Tingkat Kelembaban Tanah

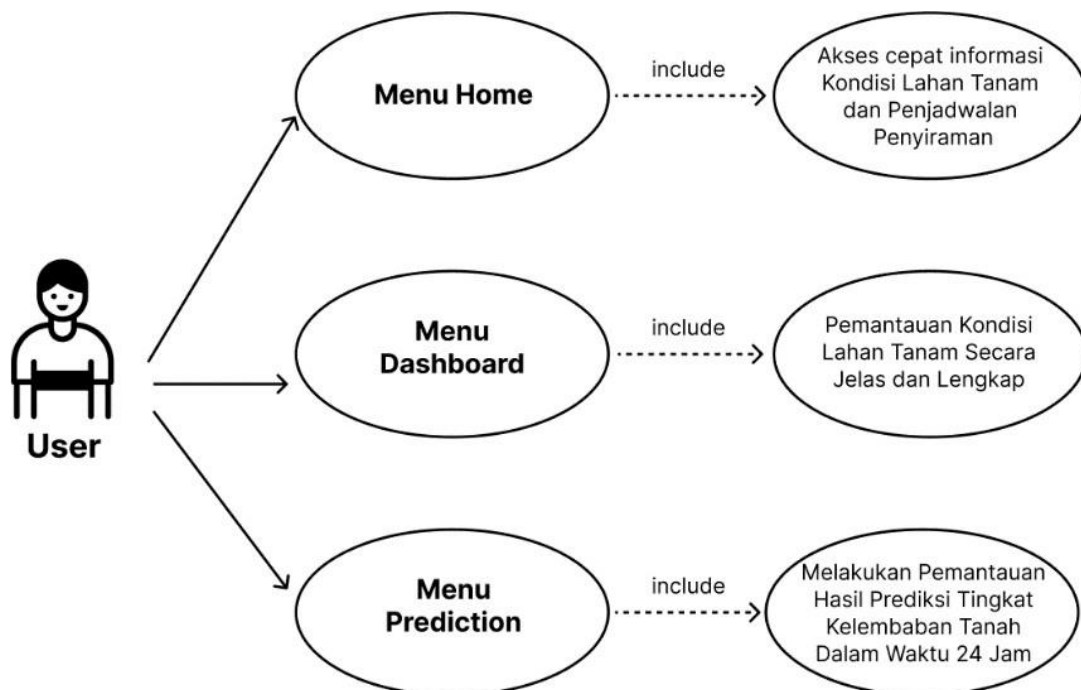
Nilai Kelembaban Tanah	Tingkat Kelembaban Tanah
$\leq 45,0$	Kering
$>45,0$ hingga $55,0$	Tepat
$>55,0$	Basah

Setelah mendapatkan prediksi tingkat kelembaban, sistem akan mencatat waktu berupa jam pertama dengan status tingkat kelembaban ‘Kering’. Pada contoh pada Tabel 3.2 tingkat ‘Kering’ pertama kali muncul pada waktu 18.00 yang berarti jadwal penyiraman pada hari itu dijadwalkan pada pukul 18.00.

### 3.2 Perancangan Antarmuka Sistem

#### 3.2.1 Use Case Diagram

Use case diagram adalah diagram yang menjelaskan secara umum interaksi apa yang dapat dilakukan pengguna dengan sistem yang memiliki manfaat untuk membantu pengguna berinteraksi dengan sistem. Use case diagram yang dirancang dapat dilihat pada Gambar 3.2.

**Gambar 3. 2** Use Case Diagram

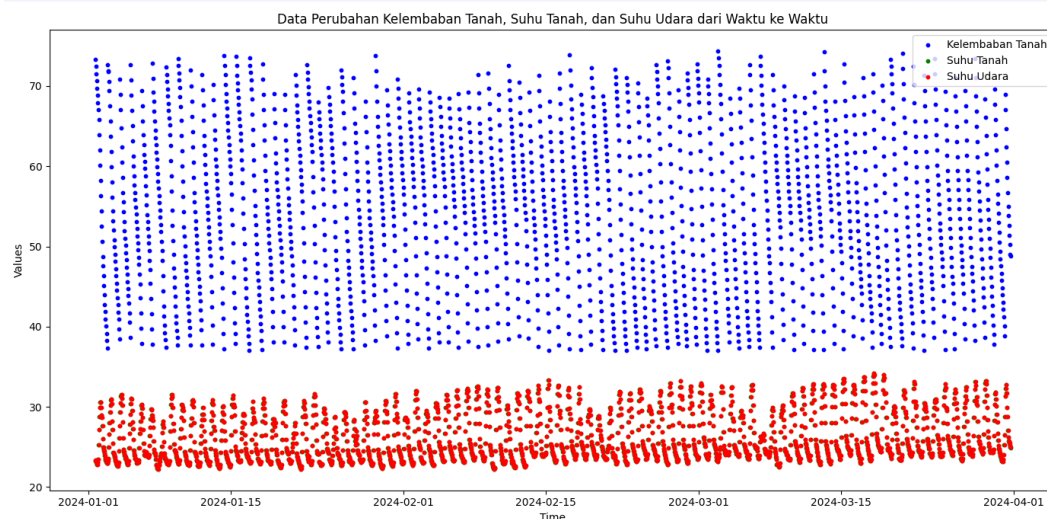
Adapun penjelasan Gambar 3.2 dapat dilihat pada Tabel 3.3.

**Tabel 3. 3** Penjelasan *Use Case Diagram*

<b>No.</b>	<b>Use Case</b>	<b>Penjelasan</b>
<b>1</b>	Menu Home	Saat pertama kali masuk ke aplikasi, user akan dialihkan ke halaman menu Home. Di sini user dapat melihat informasi kondisi lahan tanam dan penjadwalan penyiraman tanaman sebagai akses cepat.
<b>2</b>	Menu Dashboard	Menu dashboard mengalihkan pengguna ke halaman dengan informasi pemantauan kondisi lahan tanam yang lebih jelas dan lengkap dengan visualisasi dalam bentuk kartu info, grafik, dan tabel.
<b>3</b>	Menu Prediction	Menu prediction mengalihkan pengguna ke halaman dengan informasi hasil prediksi selama 24 jam mulai dari jam 00.00 hingga 23.00 dalam bentuk tabel.

### 3.2.2 Data yang Digunakan

Penelitian ini menggunakan data input berasal dari sistem penginderaan lahan tanam berupa rangkaian sensor sebagai sumber utama data. Data tersusun dari data sensor tanah dan udara yang disimpan dan dikumpulkan dengan interval pengambilan data per satu jam selama tiga bulan.



**Gambar 3. 3** Data yang Digunakan

Data lain yang digunakan adalah data prakiraan cuaca yang didapat dari penyedia layanan data cuaca dan geologi melalui Antarmuka Pemrograman Aplikasi (API) secara *online*. Data ini digunakan hanya sekali pakai guna menjadi data input saat melakukan prediksi. Data yang digunakan adalah prakiraan suhu lingkungan lahan tanam berdasarkan garis lokasi bujur dan garis lintang.

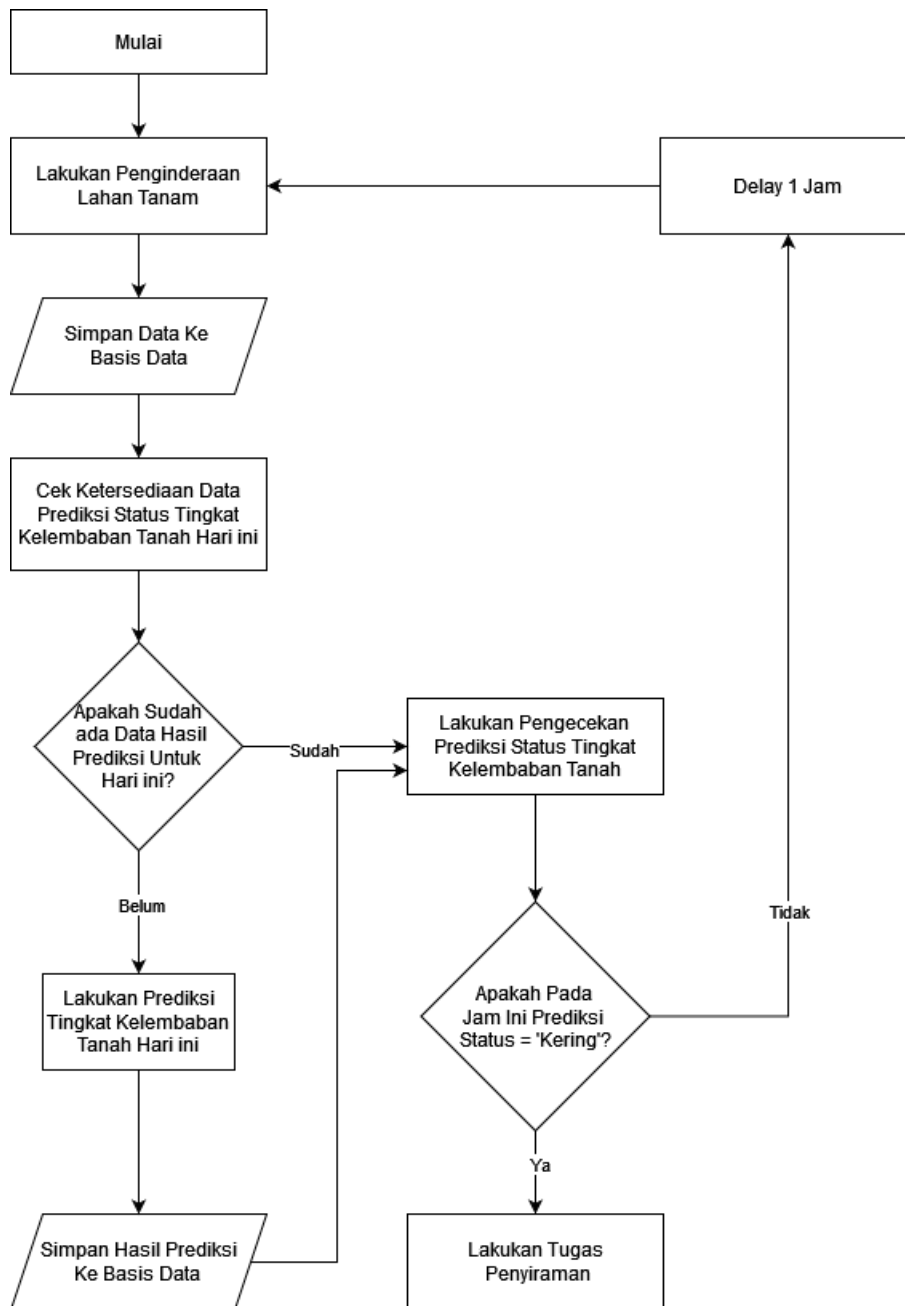
### 3.2.3 Flowchart Sistem

Dalam perancangan sistem, flowchart menjadi acuan bagaimana sistem bekerja jika dihadapkan dengan kondisi tertentu sesuai alur yang dirancang. Ada tiga flowchart yang akan dijelaskan pada flowchart sistem ini, yaitu flowchart penjadwalan penyiraman, flowchart pelatihan model LSTM, dan flowchart aplikasi prediksi Flask menggunakan model LSTM.

#### 3.2.3.1 Flowchart Penjadwalan Penyiraman

Sistem memulai tugas rutin dengan melakukan penginderaan lahan tanam yang menghasilkan data kondisi tanah dan udara, lalu data yang didapat dikirimkan ke server untuk kemudian disimpan ke dalam basis data. Kemudian sistem melakukan pengecekan ketersediaan data prediksi status tingkat kelembaban tanah untuk hari ini. Jika data belum ada, sistem akan melakukan prediksi tingkat kelembaban, kemudian menyimpan hasil prediksi ke dalam basis data. Jika data sudah ada, sistem melanjutkan pengecekan status tingkat kelembaban tanah yang disimpan. Jika status tingkat kelembaban adalah 'Kering', maka tugas penyiraman tanaman dilakukan. Jika status tingkat kelembaban

bukan ‘Kering’ maka sistem akan melakukan penundaan (*delay*) selama satu jam hingga pengulangan tugas rutin kembali dilakukan.



**Gambar 3. 4** Flowchart Penjadwalan Penyiraman

### 3.2.3.2 Flowchart Pelatihan Model LSTM

Pelatihan model dimulai dengan memuat himpunan data (*dataset*). Dataset kemudian harus di-*preprocess* terlebih dahulu sebelum melakukan proses pelatihan. Menurut Misra & Yadav (2019), keakuratan prediksi bergantung pada kualitas data. Namun data

yang dihasilkan dari berbagai sumber biasanya memiliki ketidaksempurnaan data seperti adanya data yang hilang, banyaknya *noise*, tidak konsisten, jumlah data yang sangat banyak, dan kelasnya yang tidak seimbang. Sehingga data yang tidak sempurna ini perlu tahap persiapan data untuk membersihkan dan menyiapkan data untuk dapat dianalisis kedepannya. Dalam studinya, Misra & Yadav mempelajari tentang algoritma pemrosesan data, yakni Pembersihan Data (*Data Cleaning*), Penskalaan Fitur (*Feature Scaling*), dan Pengurangan Data (*Data Reduction*).

Pada penelitian ini, data yang dikumpulkan dari sistem penginderaan adalah data kondisi lahan tanam selama kurun waktu 3 bulan dengan interval per jam yang berarti dataset yang didapat berupa data deretan waktu (*time-series*). Data yang berasal dari rangkaian sensor ini biasanya terdapat ketidaksempurnaan seperti data yang hilang karena faktor kesalahan sensor maupun kegagalan transmisi data. Data juga seringkali disertai dengan *noise* dan *outlier* yang mempunyai dampak signifikan terhadap hasil dari proses. Untuk itu penulis akan melakukan *preprocessing* data agar memastikan hasil prediksi yang didapatkan maksimal dari kondisi data yang maksimal pula.

#### 3.2.3.2.1 *Data Preprocessing*

Terdapat dua tahap *preprocessing* data yang akan dilakukan, yaitu Pembersihan Data dan Penskalaan Fitur.

#### 3.2.3.2.2 *Pembersihan Data*

Misra & Yadav (2019) pada studinya menyatakan bahwa penanganan data yang tidak sempurna yang benar adalah dengan melibatkan langkah-langkah seperti Penanganan Nilai yang Hilang (*Missing Value Treatment*), Membuang Nilai yang Hilang (*Discarding the Missing Values*), dan Metode Estimasi Parameter (*Parameter Estimation Method*), dan Imputasi K-Nearest Neighbor (*K-Nearest Neighbor Imputation*). Dalam hal ini, dataset akan melewati langkah penanganan nilai yang hilang dengan cara mengganti nilai *null* (hilang) dan nol menjadi Nilai Bukan Angka (NaN), metode estimasi parameter dengan interpolasi nilai yang hilang, dan imputasi data menggunakan metode Forward Fill.

#### 3.2.3.2.3 *Mengganti nilai null dan nol dengan Nilai Bukan Angka (NaN)*

Data dengan nilai *null* (hilang) dan nol pada dataset adalah sebuah *outlier* yang dapat mengakibatkan lonjakan data. Nilai *null* dan nol juga menjadi indikasi kesalahan sensor



saat melakukan pengideran ataupun kesalahan saat transmisi data. Oleh karena itu, nilai nol harus diubah dan diisi dengan nilai baru.

#### 3.2.3.2.4 *Interpolasi nilai yang hilang*

Interpolasi merupakan sebuah metode yang digunakan untuk mendapatkan nilai diantara dua data atau lebih yang telah diketahui nilainya. Metode ini digunakan untuk menyisipkan nilai baru ke dalam data dengan nilai NaN dari nilai data yang mengapit nilai NaN tersebut. Cara kerja interpolasi data ini bergantung pada sifat data dan kebutuhan analisis, contohnya interpolasi linear. Pada interpolasi linear, cara kerjanya adalah menghitung nilai yang hilang dengan menarik garis lurus antar dua titik data yang diketahui nilainya dan memperkirakan pada titik yang hilang. Sebagai contoh dapat dilihat pada Tabel 3.4.

**Tabel 3. 4** Contoh Data Sebelum Interpolasi

id	PH	time
1	7.1	2024-01-01
2	7.15	2024-01-02
3	NaN	2024-01-03
4	7.31	2024-01-04
5	7.41	2024-01-05
6	NaN	2024-01-06
7	7.35	2024-01-07

**Tabel 3. 5** Contoh Data Sesudah Interpolasi

id	PH	time
1	7.1	2024-01-01
2	7.15	2024-01-02
3	7.23	2024-01-03
4	7.31	2024-01-04
5	7.41	2024-01-05
6	7.31	2024-01-06
7	7.35	2024-01-07

### 3.2.3.2.5 Mengisi nilai NaN dengan Forward Fill

Forward Fill merupakan sebuah metode untuk mengisi nilai yang hilang dalam sebuah dataset. Jika setelah proses interpolasi ternyata masih ada nilai yang hilang atau NaN, proses Forward Fill ini akan mengisi data dengan cara menggantikan nilai yang hilang dengan nilai terakhir yang valid sebelum nilai yang hilang tersebut dengan asumsi bahwa data tidak berubah atau perubahan data yang tidak terlalu signifikan. Sebagai contoh Forward Fill dapat dilihat pada Tabel 3.6 dan Tabel 3.7.

**Tabel 3. 6** Contoh Data Sebelum Forward Fill

id	PH	time
1	7.1	2024-01-01
2	7.15	2024-01-02
3	NaN	2024-01-03
4	7.31	2024-01-04
5	7.41	2024-01-05
6	NaN	2024-01-06
7	7.35	2024-01-07

**Tabel 3. 7** Contoh Data Sesudah Forward Fill

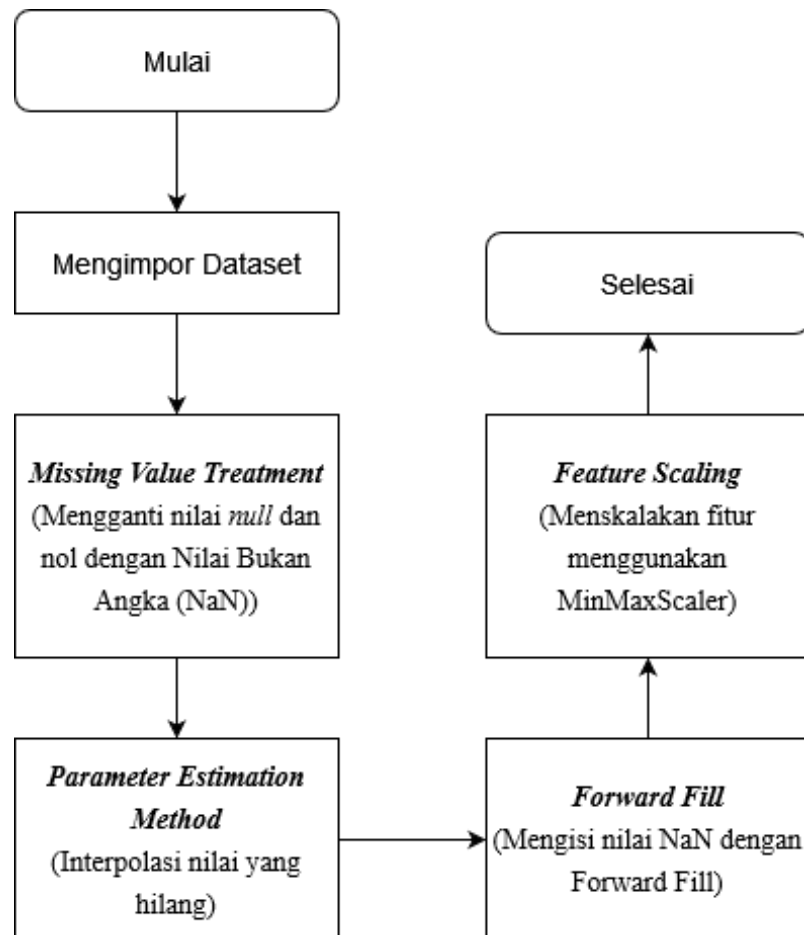
id	PH	time
1	7.1	2024-01-01
2	7.15	2024-01-02
3	7.15	2024-01-03
4	7.31	2024-01-04
5	7.41	2024-01-05
6	7.41	2024-01-06
7	7.35	2024-01-07

### 3.2.3.2.6 Penskalaan Fitur

Sebelum data diproses, kolom data yang dipilih sebagai fitur output akan diskalakan menggunakan MinMaxScaler. Menurut Misra & Yadav (2019) MinMaxScaler adalah sebuah algoritma Min-Max yang menskalakan fitur ke rentang tetap  $[-1,1]$  atau  $[0,1]$ . Penskalaan ini membutuhkan sampel minimum dan maksimum sebagai  $x_{min}$  dan  $x_{max}$

yang kemudian menggantikan fitur orisinal menjadi fitur yang diskalakan sebagai

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}}$$



**Gambar 3. 5** Flowchart Pre-processing Data

#### 3.2.3.2.7 Input Reshaping

Setelah dataset berhasil di-*preprocessing*, selanjutnya melakukan pembentukan ulang (*reshaping*) data input. Dalam analisa data deretan waktu (*time-series*), khususnya menggunakan model LSTM, data disusun menjadi rangkaian observasi sepanjang waktu. Setiap observasi (atau titik data) pada rangkaian mewakili cuplikan urutan waktu pada langkah waktu tertentu. Model LSTM belajar dari urutan ini untuk menangkap dan membentuk relasi dan pola temporal dalam data. Model LSTM dalam melakukan *reshaping* data input melakukan tiga tahap, yakni menentukan *sequence length* (panjang urutan), membuat urutan masukan, dan bentuk masukan (*input shape*) untuk LSTM.

1. Menentukan *sequence length* (*seq\_length*)

Parameter ini menentukan berapa banyaknya langkah waktu (atau observasi) sebelumnya yang harus dipertimbangkan oleh LSTM dalam melakukan prediksi. Sebagai contoh, jika  $seq\_length = 12$ , maka model akan melihat 12 langkah waktu terakhir untuk memprediksi langkah waktu berikutnya.

## 2. Membuat urutan masukan (*input sequence*)

Pada tahap ini *input sequence* menjadi sampel masukan untuk model LSTM. Sebagai contoh, jika dataset yang tersedia memiliki 1000 observasi, *reshaping* dengan  $seq\_length = 12$  akan menghasilkan sebanyak 989 urutan ( $1000 - 12 + 1$ ).

## 3. *Input Shape* untuk LSTM

Model LSTM membutuhkan input data dalam format 3 dimensi: (sample, langkah waktu, dan fitur). Sampel adalah titik data yang tersedia pada dataset, langkah waktu adalah jumlah observasi dalam setiap *sequence length*, dan fitur adalah variabel (berupakolum pada dataset) yang digunakan untuk mengamati dalam setiap langkah waktu.

Dalam penelitian ini, data kelembaban tanah yang harus diprediksi adalah sebanyak 24 titik waktu (selama 24 jam) yang berarti model LSTM akan mempertimbangkan langkah waktu sebanyak 24 langkah dengan total 977 urutan ( $1000 - 24 + 1$ ) dalam melakukan prediksi. Hasil dari prediksi adalah sebanyak 24 data prediksi dalam interval waktu satu jam.

### 3.2.3.2.8 *Inisialisasi, Model Training, dan Model Compiling*

Selanjutnya membagi dataset menjadi dua set; set pelatihan (*train*) dan set pengujian (*test*). Dataset dibagi dengan perbandingan 20:80. 20% dari dataset digunakan sebagai pengujian dan 80% digunakan untuk pelatihan. Perbandingan dapat diubah-ubah sesuai pengaturan pengujian.

Langkah selanjutnya yaitu menginisialisasikan algoritma Long Short-Term Memory (LSTM) dan melatih model. LSTM membutuhkan *hyperparameter*, yaitu :

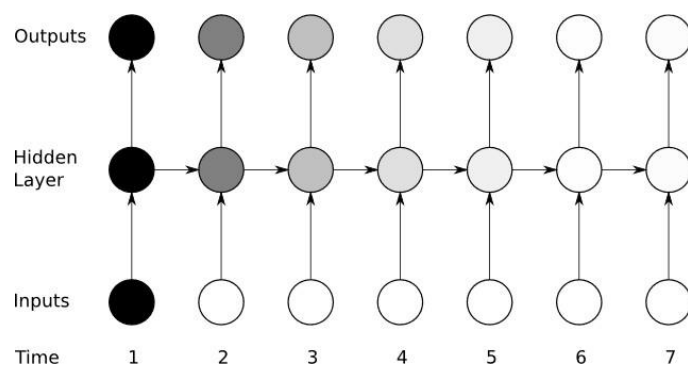
#### 1. *Learning rate value*

*Learning rate* adalah parameter yang berperan penting yang mengontrol seberapa banyak model harus diubah sebagai respons terhadap estimasi error setiap kali bobot model diperbarui. *Learning rate* biasanya berkisar antara 0.0 hingga 1.0, atau sering kali di menggunakan nilai yang lebih kecil seperti 0.001 atau 0.0001.

#### 2. Jumlah *Neuron* pada *Hidden Layer*

Arsitektur LSTM terdiri dari sekumpulan subnet yang saling terhubung dan berulang, yang dikenal sebagai blok memori (Graves 2012). Blok-blok ini dapat dianggap sebagai memori chip pada komputer digital. Setiap blok berisi satu atau lebih sel memori yang terhubung sendiri dan tiga unit perkalian — gerbang input, output dan lupa (*forget gates*) — yang memberikan analogi berkelanjutan dari operasi tulis, baca, dan reset pada sel.

LSTM Layer adalah blok penyusun inti model yang dirancang untuk menangkap relasi jangka panjang dalam data sekuensial yang terdiri dari sel-sel memori dan gerbang (gerbang input, output, lupa) yang memungkinkan LSTM menyimpan atau melupakan informasi secara selektif seiring waktu. Dense Layer atau biasa dikenal lapisan yang terhubung sepenuhnya, ditambahkan untuk memetakan output LSTM ke output prediksi yang diinginkan.



**Gambar 3. 6** *The Vanishing Gradient Problem pada RNN*

#### 1. Error Target berupa MSE (Mean Squared Error)

Target eror adalah treshold yang ditentukan untuk metrik kesalahan yang digunakan untuk menentukan kriteria penghentian proses pelatihan. Target eror merupakan nilai minimum yang diinginkan untuk loss function, yaitu MSE.

#### 2. Maksimum Epoch

Istilah epoch pada Neural Network mengacu pada satu lintasan lengkap dari suatu data pelatihan melalui algoritma pembelajaran (deepai.org). Jumlah maksimum epoch memiliki pengaruh terkait underfitting dan overfitting, yang berarti maksimum epoch berpengaruh langsung terhadap akurasi model yang dapat menjadi rendah ataupun menjadi tinggi pada data training ataupun data testing. Tingkat akurasi berdampak pada hasil prediksi.

### 3.2.3.2.9 Melakukan Prediksi

Langkah selanjutnya yaitu melakukan prediksi pada *test* data ( $X_{test}$ ). Kemudian menyimpan model dan penskala (*scaler*) untuk digunakan oleh aplikasi prediksi nantinya. Setelah  $X_{test}$  diprediksi, evaluasi performa model menggunakan Root MeanSquare Error (RMSE) dan R-Squared Score (R2 Score). Perhitungan error pada model dilakukan dengan cara menghitung perbedaan antara nilai yang diprediksi ( $\hat{y}$ ) dengan nilai sesungguhnya ( $y_i$ ). Perhitungan error dapat dihitung menggunakan persamaan 3.1 (Chai & Draxler 2014) sebagai berikut.

$$\text{Error}_i = \hat{y}_i - y_i \quad (3.1)$$

Keterangan :

$\hat{y}_i$  = Nilai prediksi

$y_i$  = Nilai aktual

Kemudian hasil error dikuadratkan (squared) untuk mendapatkan hasil nilai error yang positif dan untuk memberikan bobot lebih pada error yang lebih besar. Berikut persamaan 3.2 (Chai & Draxler 2014) untuk menghitung nilai squared error.

$$\text{Squared Error}_i = (\text{Error}_i)^2 = (\hat{y}_i - y_i)^2 \quad (3.2)$$

Keterangan :

$\text{Error}_i$  = Persamaan error

$\hat{y}_i$  = Nilai prediksi

$y_i$  = Nilai aktual

Untuk mendapatkan Mean Square Error (MSE), Squared Errori kemudian dirata-ratakan untuk mendapatkan gambaran keseluruhan tentang performa model. MSE dapat dihitung menggunakan persamaan 3.3 (Chai & Draxler 2014).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (3.3)$$

Keterangan :

$n$  = Jumlah observasi pada dataset

$\hat{y}$  = Nilai prediksi

$y_i$  = Nilai aktual

Root Mean Square Error (RMSE) merupakan sebuah metrik statistik standar untuk mengevaluasi performa model (Chai & Draxler 2014). RMSE merupakan akar pangkat dua (square root) dari MSE. RMSE memberikan ukuran rata-rata besarnya error dalam satuan yang sama dengan variabel yang diprediksi. Berikut persamaan menghitung RMSE menggunakan persamaan 3.4 (Chai & Draxler 2014).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (3.4)$$

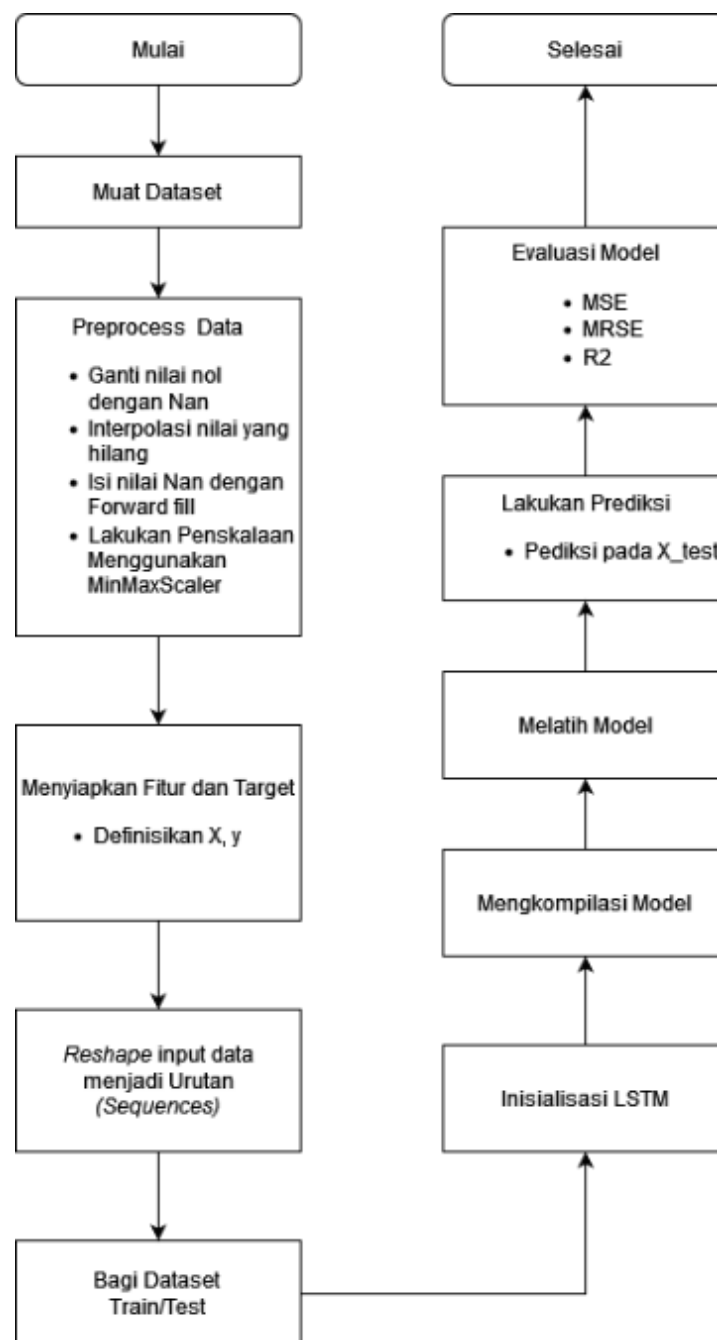
Keterangan :

$n$  = Jumlah observasi pada dataset

$\hat{y}_i$  = Nilai prediksi

$y_i$  = Nilai aktual

Chai & Draxler juga menyimpulkan bahwa RMSE lebih cocok untuk digunakan dibandingkan dengan Mean Absolute Error (MAE) ketika error pada model mengikuti distribusi normal. R-Squared Score (R2 Score) atau dikenal sebagai Coefficient of determination dapat diinterpretasikan sebagai proporsi dari varian variabel dependen yang dapat diprediksi dari variabel independen (Chicco et al. 2021). R2 Score digunakan untuk mengevaluasi seberapa baik model regresi sesuai dengan data yang diamati. R2 Score memiliki nilai antara 0 hingga 1, dimana 1 berarti model mampu menjelaskan seluruh variasi data, dan 0 menunjukkan model tidak menjelaskan variasi sama sekali dan mungkin tidak berguna dalam memprediksi. Setelah proses evaluasi model selesai dilakukan maka proses pelatihan model selesai dilakukan. Alur proses pelatihan dapat disimak pada flowchart pada Gambar 3.7



**Gambar 3. 7** Flowchart Pelatihan Model Long Short-Term Memory

### 3.2.3.3 Flowchart Aplikasi Prediksi Flask

Aplikasi prediksi Flask adalah sebuah aplikasi yang dibangun menggunakan Python Flask yang berfungsi sebagai layanan dengan fungsi prediksi kelembaban tanah. Aplikasi ini berjalan pada *cloud server* yang aktif menerima perintah prediksi. Layanan prediksi akan dilakukan jika aplikasi diakses melalui API pada *endpoint* /prediction pada web browser. Pemaparan layanan aplikasi prediksi Flask akan dijelaskan sebagai berikut :



#### 3.2.3.3.1 *Memuat Model LSTM dan MinMaxScaler*

Pertama-tama output atau keluaran dari model LSTM dan MinMaxScaler berupa berkasobjek serial (.pkl) dimuat. Model dan *scaler* ini menjadi acuan aplikasi melakukan prediksi.

#### 3.2.3.3.2 *Menentukan Sequence Length*

Merujuk pada pelatihan model, *sequence length* merupakan berapa banyaknya langkahwaktu atau observasi sebelumnya yang harus dipertimbangkan LSTM dalam melakukan prediksi. *Sequence length* harus ditentukan dan harus sama dengan *sequencelength* pada model, yaitu 12.

#### 3.2.3.3.3 *Menentukan Fitur Input*

Fitur input merupakan variabel yang akan dipertimbangkan dalam memprediksi variabel target. Fitur input di sini adalah nilai ‘kelembaban tanah’ terbaru, dan ‘suhu tanah’, ‘suhu udara, dan ‘waktu’ yang didapat dari WeatherAPI. Sedangkan variabel target adalah ‘kelembaban tanah’.

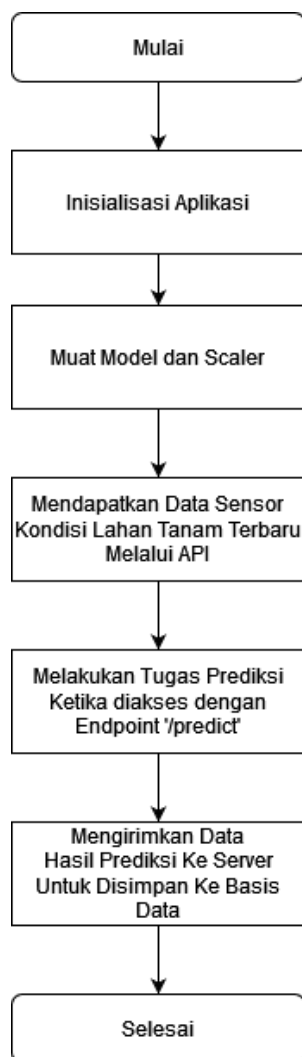
#### 3.2.3.3.4 *Mendapatkan Nilai Fitur Input*

Nilai fitur input didapatkan dari dua sumber, yakni dari sistem (*cloud server*) dan WeatherAPI. Nilai ‘kelembaban tanah’ terbaru didapatkan dari sistem via API dengan *endpoint* `/api/get-current-sensor-data`. Sedangkan ‘suhu tanah’, ‘kelembaban tanah’, ‘suhu udara, dan ‘waktu’ didapat dari WeatherAPI.

#### 3.2.3.3.5 *Melakukan Prediksi*

Setelah model dan *scaler* dimuat, parameter telah ditentukan, dan fitur input didapat,selanjutnya melakukan prediksi variabel target ‘kelembaban’ tanah.

Flowchart aplikasi prediksi Flask dapat disimak pada Gambar 3.8.



**Gambar 3. 8** *Flowchart* Aplikasi Prediksi Flask

#### 3.2.4 Komponen Alat

Beberapa komponen alat yang digunakan untuk membangun rangkaian sistem. Alat dan komponen pendukung lainnya dapat dilihat pada Tabel 3.8.

**Tabel 3. 8** Komponen Alat

No	Alat	Unit	Fungsi
1	Arduino UNO R3	1 unit	Mikrokontroler pengatur Dragino LoRa Shield
2	NodeMCU ESP8266 ESP -12E	1 unit	Mikrokontroler dengan WiFi modul
3	RS485-LoRa Wireless Station	1 unit	Sebagai mikrokontroler RS485 Soil NPK PH EC Sensor

**Tabel 3. 9** Komponen Alat ( Lanjutan )

4	<b>Dragino LoRa Shield</b>	<b>1 unit</b>	<b>Mikrokontroler dengan modul LoRa</b>
5	RS485 Soil NPK PH EC Sensor	1 unit	Sebagai sensor NPK tanah
6	Sensor BMP280	1 unit	Sebagai sensor udara

### 3.2.5 Perancangan Sistem Penginderaan

Dalam perancangan sistem penginderaan berupa rangkaian sensor, dibutuhkan bahan-bahan berupa barang elektronik dan kabel. Informasi bahan-bahan tersebut dipaparkan pada Tabel 3.2.

### 3.2.6 Perancangan Robot Pertanian

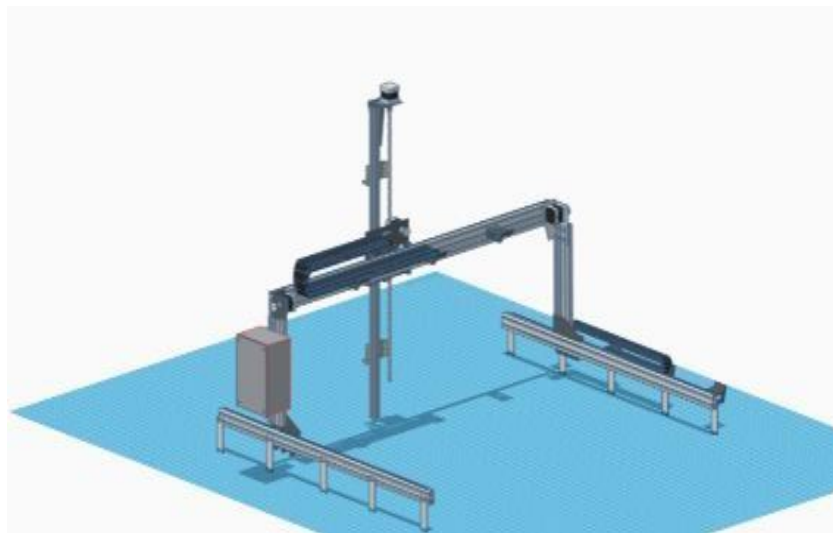
Dalam melakukan perancangan, diperlukan alat maupun bahan-bahan penyusun robot dan desain bangun robot. Alat dan bahan dipaparkan pada Tabel 3.10 dan desain bangun robot ditampilkan pada Gambar 3.9.

**Tabel 3. 10** Rincian Alat dan Bahan Penyusun Robot Pertanian

No	Alat	Ukuran	Jumlah
1	V slot 2040	400cm	2
2	V slot 2020	120 cm	1
3	V slot 2060	80cm	2
4	V slot 2060	200cm	1
5	Besi Ulir	120 cm	1
6	V wheels	1 inch	23
7	Cable Carrier X-axis	25x38 mm/2.5m	3
8	Cable Carrier Y-axis	35x60 mm/2.5m	3
9	Cable Carrier Z-axis	35x60 mm/1.5m	1
10	Kabel Jumper	10cm	16
11	Kabel 0.75	2m	2
12	Baut 8	1 inch 5 mm	25
13	Kabel nema 17	1000 mm	2
14	Kabel nema 17	2000mm	3

**Tabel 3. 11** Rincian Alat dan Bahan Penyusun Robot Pertanian ( Lanjutan )

15	Kabel nema 17	40mm	1
16	Timing belt	16 m	1
17	Katup solenoid 12V	12V & 220V	1
18	Junction Box	115x115x110mm	1
19	Selang Air	1m	4
20	Kabel 0.75 blitz	1m	24
21	Dudukan V slot	6mmx110mm	1
22	Plat besi	150 mm x 125.5 mm	2
23	Plat besi	60 mm x 110 mm	2
24	Plat besi	50 mm x 80 mm	1
25	Plat besi	185mm x 254 mm	1
26	Water solenoid	0.5 inch 220v	1
27	Gt 2 20T	5 mm	3
28	Shaft Coupler	5mm to 8mm	1
29	T nut	5mm	50

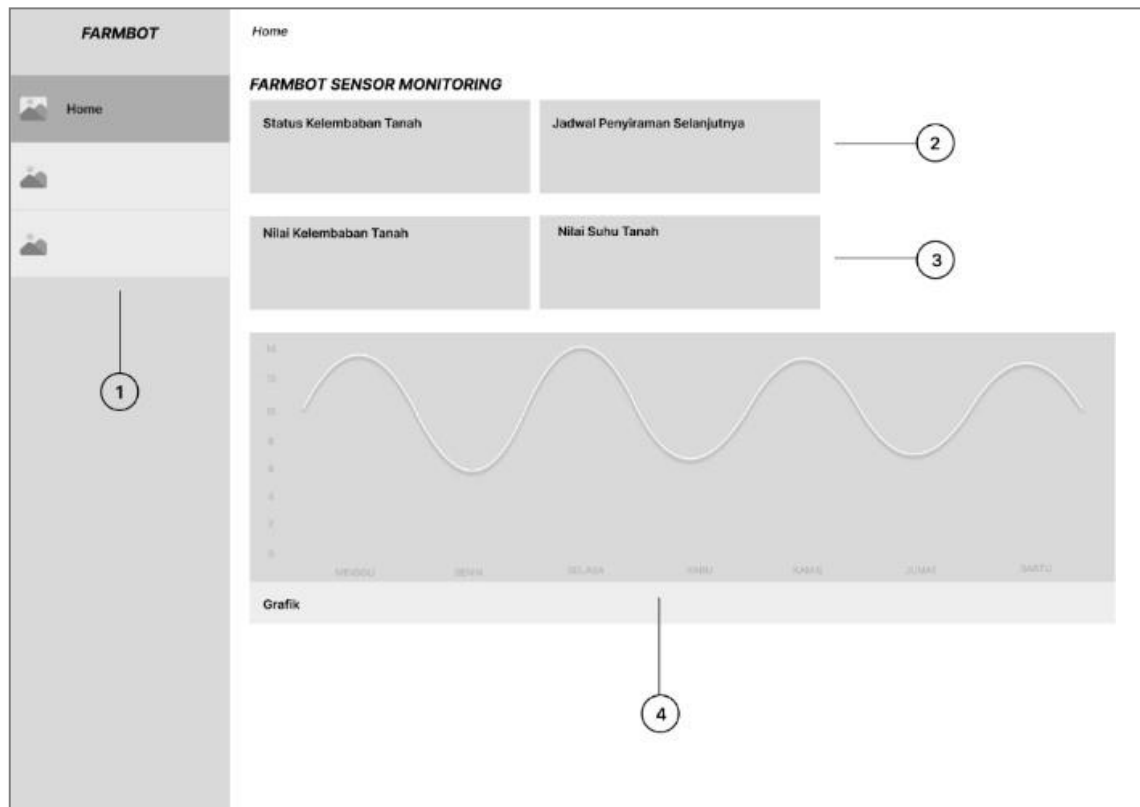
**Gambar 3. 9** Rancangan Desain Robot Pertanian

### 3.2.7 Ilustrasi Bentuk Antarmuka Aplikasi

Aplikasi berbasis web yang dibuat memiliki rancangan antarmuka. Rancangan antarmuka tersebut meliputi halaman home, halaman dashboard, dan halaman prediction.

### 3.2.7.1 Ilustrasi Tampilan Home

Tampilan *Home* adalah halaman yang akan langsung ditampilkan kepada pengguna. Halaman ini menjadi sebuah tampilan akses cepat bagi pengguna yang ingin melihat informasi kondisi lahan tanam dan jadwal penyiraman. Desain tampilan *Home* diilustrasikan pada Gambar 3.10.



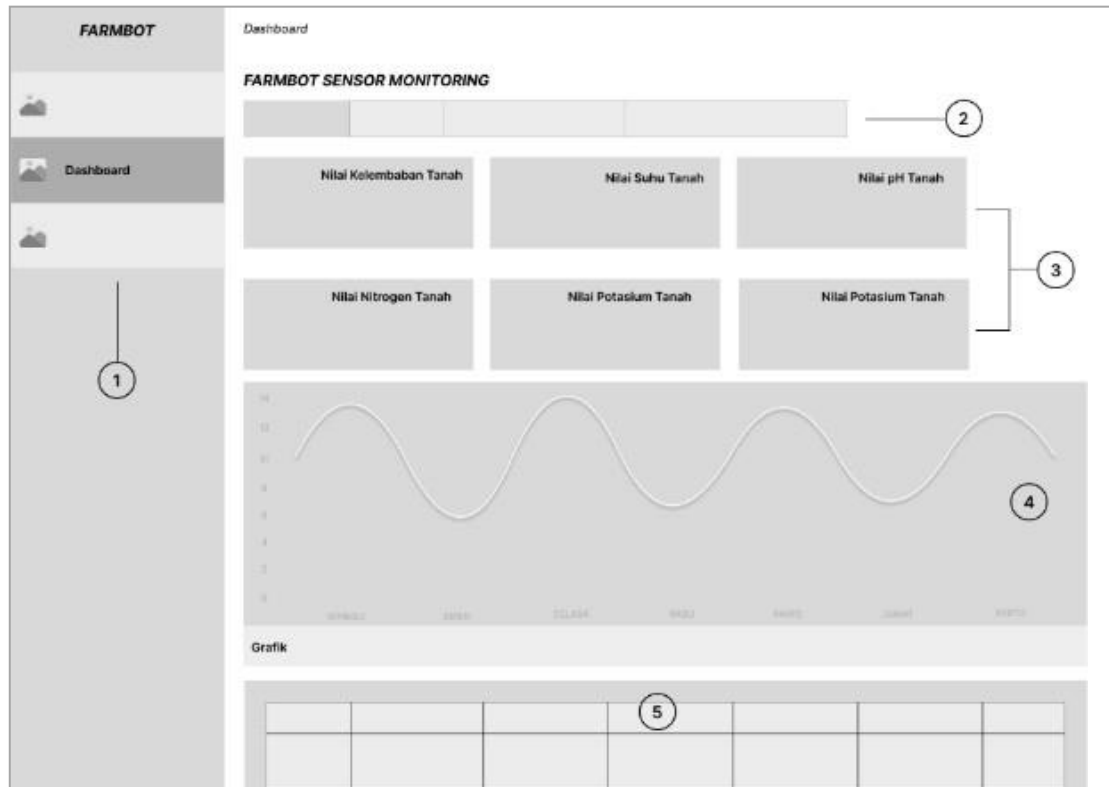
**Gambar 3. 10** Ilustrasi Tampilan Halaman *Home*

Keterangan :

1. Merupakan navigasi *Home*.
2. Merupakan Status Kelembaban Tanah dan Jadwal Penyiraman.
3. Merupakan kartu informasi nilai sensor tanah.
4. Merupakan grafik informasi nilai sensor mingguan.

### 3.2.7.2 Ilustrasi Tampilan Halaman Dashboard

Tampilan halaman *Dashboard* adalah halaman yang memuat informasi penuh tentang kondisi lingkungan. Informasi berupa data ditampilkan dalam bentuk kartu informasi, grafik, dan tabel. Desain ini diilustrasikan pada Gambar 3.11.



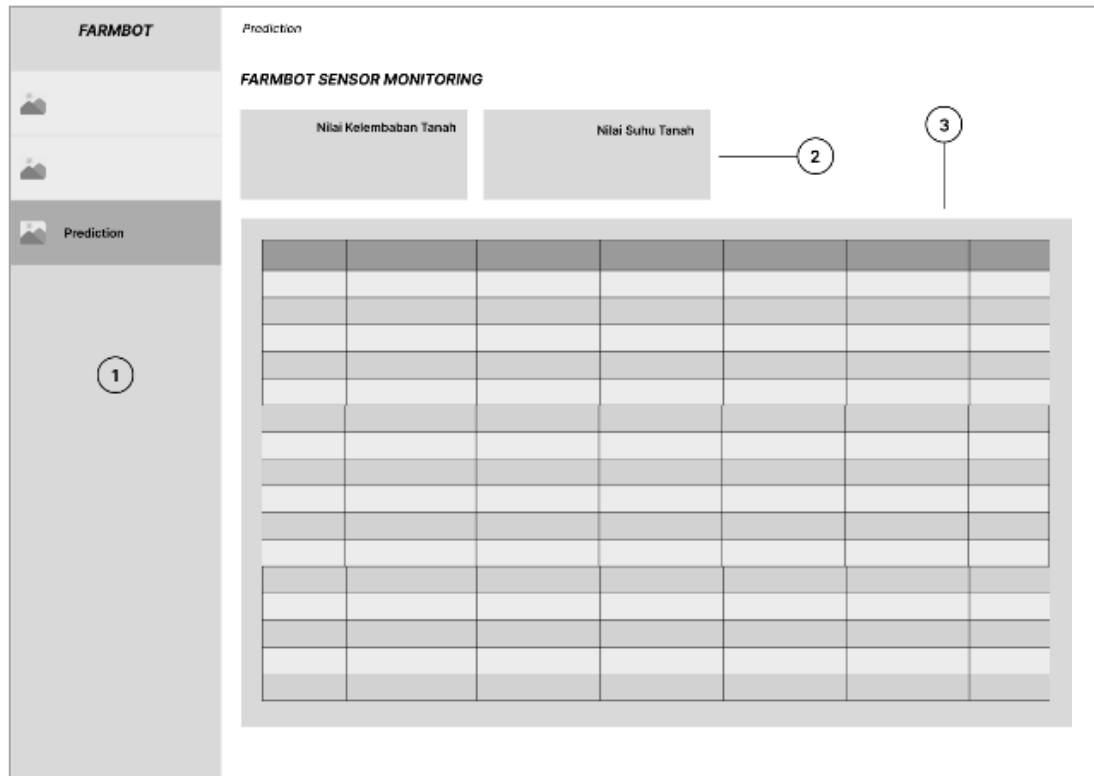
**Gambar 3. 11** Ilustrasi Tampilan Beranda

Keterangan :

1. Merupakan navigasi *Dashboard*.
2. Merupakan menu navigasi untuk menampilkan data yang ingin dilihat.
3. Merupakan kartu informasi nilai sensor.
4. Merupakan Grafik informasi nilai sensor mingguan.
5. Merupakan Tabel riwayat data sensor.

### 3.2.7.3 Ilustrasi Tampilan Halaman Prediction

Tampilan halaman *Prediction* merupakan halaman yang menampilkan hasil prediksi status kelembaban tanah dalam kurun waktu 24 jam dalam bentuk tabel. Berikut desain halaman *Prediction* ditampilkan pada Gambar 3.12.



**Gambar 3. 12** Ilustrasi Tampilan Prediksi

Keterangan :

1. Merupakan navigasi *Prediction*.
2. Merupakan kartu informasi nilai sensor.
3. Merupakan tabel yang memuat informasi hasil prediksi status kelembaban.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

##### 4.1.1 Perangkat Keras Pada Sistem

Komponen perangkat keras yang digunakan pada penelitian ini adalah sebuah laptop Acer Aspire Z3-451 dengan spesifikasi prosesor AMD A10-5757M, AMD Radeon HD 8650G, RAM 8 GB DDR3 Harddisk 1TB + SSD 120 GB.

##### 4.1.1.1 Perangkat Lunak Pada Sistem

Berikut adalah perangkat lunak yang digunakan pada perangkat keras sistem dan *Cloud Server* yang digunakan untuk menangani proses penyimpanan data pada Tabel 4.1.

**Tabel 4. 1** Perangkat Lunak

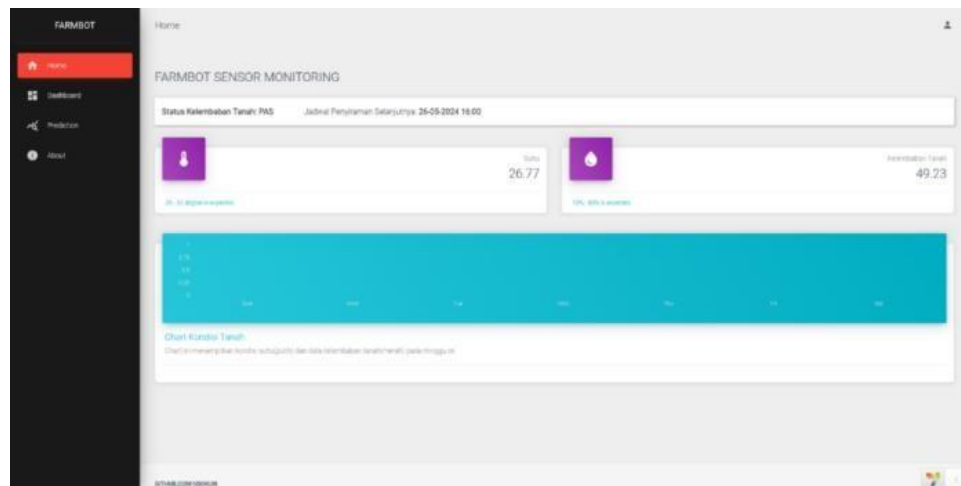
No	Komponen	Perangkat Lunak
1	Acer Aspire Z3-451	Windows 11
		Visual Studio Code
		Figma Dekstop
		Yii2-Framework & NodeJs + ExpressJs
		Arduino IDE
2	Cloud Server	Mosquitto MQTT
		XAMPP
		Yii2-Framework & NodeJs + ExpressJs

#### 4.2 Implementasi Ilustrasi Tampilan Aplikasi Web

##### 4.2.1 Implementasi Ilustrasi Tampilan Home

Tampilan *Home* adalah tampilan awal sistem sebagai akses cepat bagi pengguna. Halaman *Home* menyajikan tampilan ringkas kondisi tanah lahan tanam dan jadwal penyiraman selanjutnya. Implementasi ilustrasi ditampilkan pada Gambar 4.1.

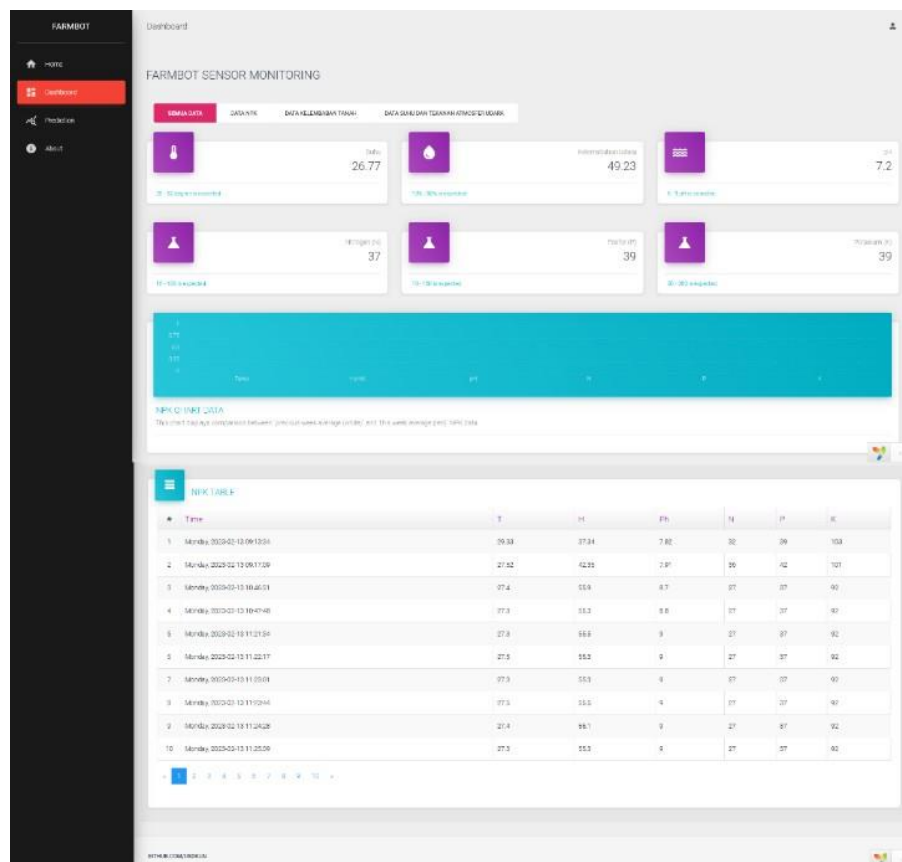




**Gambar 4. 1** Implementasi Tampilan *Home*

#### 4.2.2 Implementasi Ilustrasi Tampilan Dashboard

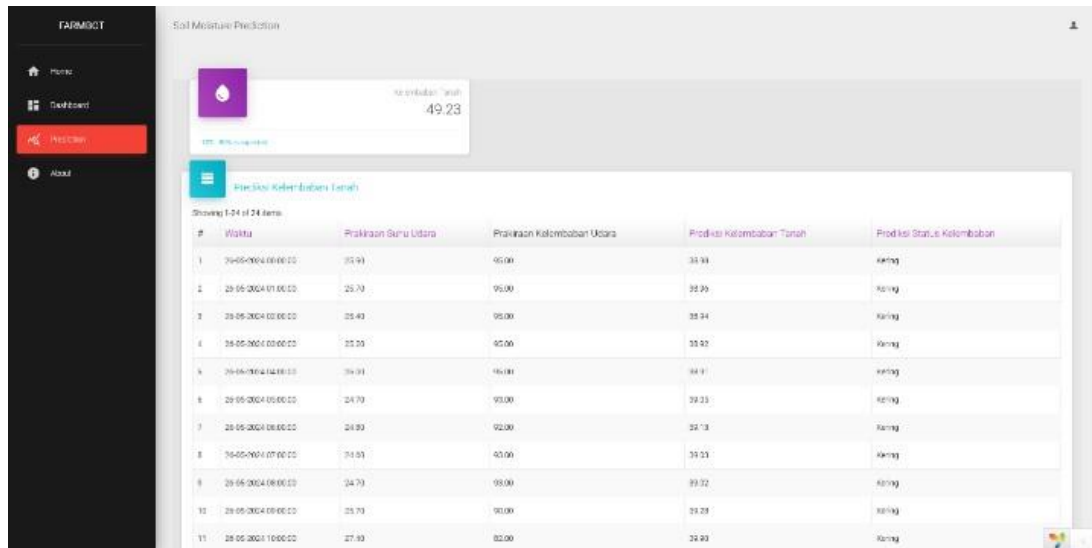
Tampilan ini menyajikan data lengkap kondisi lahan tanam yang tersaji dalam bentuk kartu informasi, grafik, dan tabel. Ilustrasi tampilan *Dashboard* ditampilkan pada Gambar 4.2.



**Gambar 4. 2** Implementasi Tampilan *Dashboard*

#### 4.2.3 Implementasi Ilustrasi Tampilan Prediction

Tampilan ini menyajikan informasi data hasil prediksi kelembaban tanah hari ini dalam kurun waktu 24 jam dalam bentuk tabel. Berikut implementasi ilustrasi *Prediction* ditampilkan pada Gambar 4.3.



#	Waktu	Prediksi Suhu Udara	Prediksi Kelembaban Udara	Prediksi Kelembaban Tanah	Prediksi Status Kelembaban
1	20-05-2024 08:00:00	35.94	95.00	33.34	Kering
2	20-05-2024 09:00:00	25.73	95.00	32.39	Kering
3	20-05-2024 10:00:00	25.43	95.00	32.34	Kering
4	20-05-2024 11:00:00	25.55	95.00	33.42	Kering
5	20-05-2024 12:00:00	34.94	94.00	34.91	Kering
6	20-05-2024 13:00:00	24.73	93.00	32.23	Kering
7	20-05-2024 14:00:00	24.93	92.00	32.18	Kering
8	20-05-2024 15:00:00	31.53	93.00	33.23	Kering
9	20-05-2024 16:00:00	34.73	93.00	33.22	Kering
10	20-05-2024 17:00:00	25.73	93.00	33.28	Kering
11	20-05-2024 18:00:00	27.63	92.00	34.33	Kering

**Gambar 4. 3** Implementasi Halaman Prediksi

#### 4.3 Implementasi Perancangan Robot Pertanian

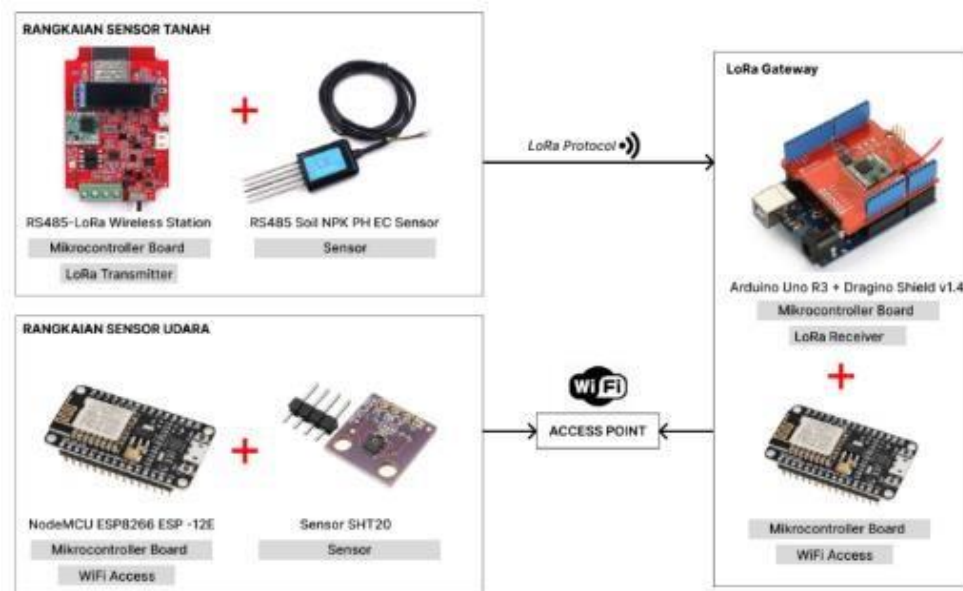
Setelah perancangan robot pertanian yang dilakukan dan alat maupun bahan-bahan penyusun robot telah pertanian terkumpul, maka implementasi perancangan robot pertanian dilakukan. Berikut implementasi perancangan robot pertanian pada Gambar 4.4.



**Gambar 4. 4** Implementasi Robot Pertanian

#### 4.4 Implementasi Perancangan Sistem Penginderaan

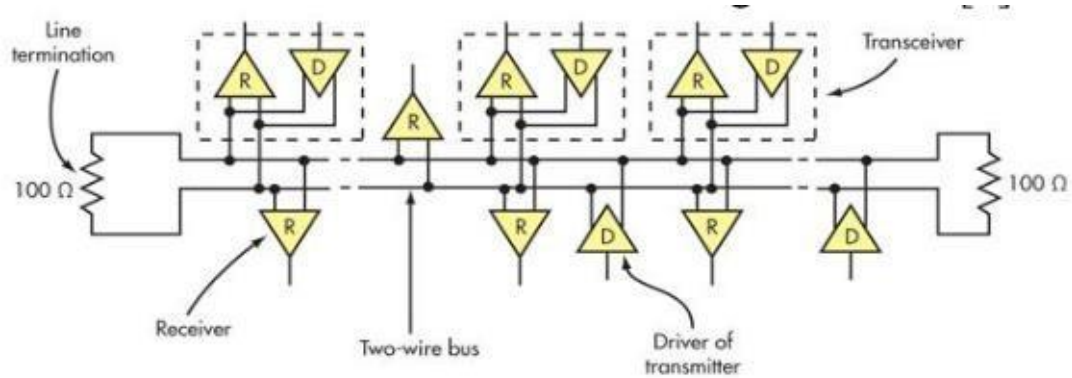
Pada implementasi perancangan sistem penginderaan, sistem penginderaan tersusun oleh dua jenis rangkaian sensor, yaitu rangkaian sensor tanah dan udara. Kemudian terdapat sebuah LoRa Gateway sebagai penerima transmisi (*receiver*) yang berfungsi meneruskan data ke *cloud server*.



**Gambar 4.5** Skema Rangkaian Sensor

##### 4.4.1 Implementasi Rangkaian Sensor Tanah

Rangkaian sensor tanah dibangun menggunakan 2 komponen, yakni RS485-LoRa Wireless Station dan RS485 Soil NPK PH EC Sensor. Kedua komponen ini terhubung melalui antarmuka RS485. Menurut Hung *et al.* (2020) Antarmuka RS485 menentukan pensinyalan diferensial pada dua saluran, bukan saluran tunggal dengan tegangan yang direferensikan ke *ground* antarmuka RS232. Format diferensial menghasilkan *noise cancellation* mode umum yang efektif dengan panjang kabel maksimum 1200 meter dengan kecepatan data maksimum 100 kilobit per detik. Konfigurasi umum antarmuka ini adalah jaringan bus dengan beberapa *drops* atau koneksi. Standar ini menetapkan maksimal 32 driver (pemancar) dan 32 penerima (ditampilkan pada Gambar 4.6). Semua *receiver* terhubung seluruhnya, namun *line drivers* terputus dari garis ketika tidak sedang melakukan transmisi dan saluran ini diakhiri dengan *load matching resistance*.



**Gambar 4. 6** Konfigurasi Umum Jaringan RS485

Pada komponen RS485-LoRa Wireless Station, antarmuka RS485 dipaparkan pada Gambar 4.7 dan dijelaskan pada Tabel 4.2.

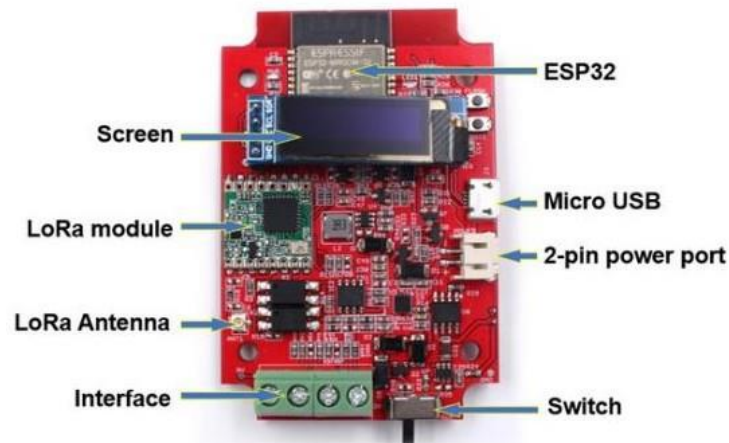


**Gambar 4. 7** Antarmuka RS485 LoRa Wireless Station

**Tabel 4. 2** Penjelasan Antarmuka RS485 LoRa Wireless Station

Pin Sensor	RS485
Cokelat	VCC
Hitam	GND
Kuning	RS485_A
Biru	RS485_B

RS485-LoRa Wireless Station memiliki komponen-komponen penyusun, diantaranya adalah ESP32, Screen, LoRa module, Micro USB module, 2-pin power port, LoRa Antena, Antarmuka RS485, dan tombol Switch.



**Gambar 4. 8** Komponen - komponen Penyusun RS485-LoRa Wireless Station

Rangkaian sensor tanah ini menghasilkan data kelembaban, suhu, pH, dan NPK (Nitrogen, Fosfor, dan Potasium) tanah. Data yang didapatkan kemudian ditransmisikan melalui protokol LoRa ke LoRa *Gateway*.

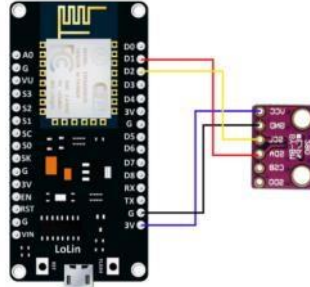


**Gambar 4. 9** Implementasi Rangkaian Sensor Tanah

#### 4.4.2 Implementasi Rangkaian Sensor Udara

Rangkaian sensor udara terdiri dari dua komponen, yaitu NodeMCU ESP2866 ESP-12 E (selanjutnya ditulis ESP-12E) dan sensor BMP280. ESP-12E berperan sebagai mikrokontroler yang mengontrol sensor BMP280. Kedua komponen ini terhubung melalui antarmuka I2C. I2C (Inter-Integrated Circuit) adalah sebuah koneksi protokol antarmuka serial bus. Antarmuka ini berkomunikasi hanya menggunakan dua kabel, yaitu SDA (*serial data*) dan SCL (*serial clock*). Antarmuka I2C bekerja dalam dua mode, yaitu *master mode* dan *slave mode*.

Pada rangkaian ini ESP-12E terhubung ke sensor BMP280 melalui pin SDA dan SCL. Berikut Gambar 4.10 menampilkan hubungan komunikasi melalui kabel dan Tabel 4.3 memaparkan hubungan komunikasi.



**Gambar 4. 10** Hubungan Komunikasi Antarmuka I2C ESP-12E dengan BMP280

**Tabel 4. 3** Komunikasi Antarmuka I2C ESP-12E dengan BMP280

ESP-12E Pin	BMP280 Pin	Warna Kabel	Keterangan
D1	DA	Merah	Kabel Serial Data (SDA)
D2	CL	Kuning	Kabel Serial Clock (SCL)
G	-	Hitam	<i>Ground</i>
3V	+	Biru	Power (VCC)

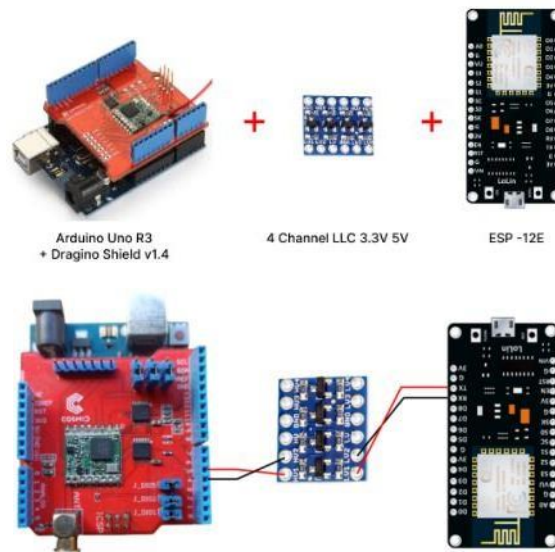


**Gambar 4. 11** Sensor Udara

#### 4.4.3 Implementasi LoRa Gateway

LoRa *Gateway* adalah sebuah perangkat LoRa *receiver* yang menangkap pesan melalui protokol LoRa. Perangkat *transmitter* maupun *receiver* harus memiliki parameter yang diatur dan modul LoRa dengan frekuensi yang sama agar perangkat dapat saling berkomunikasi.





**Gambar 4. 12** Rancangan LoRa Gateway

LoRa Gateway yang dirancang pada sistem ini adalah sebuah gabungan antara mikrokontroler Arduino Uno R3 dengan Dragino LoRa Shield, dan memanfaatkan ESP-12E yang mempunyai modul WiFi agar dapat mengakses internet. Pada rangkaian ini komponen melakukan komunikasi data melalui antarmuka Rx Tx. Rx adalah *receiver* atau penerima dan Tx adalah *Transmitter* atau pengirim. Antarmuka ini menganut sistem pengirim-penerima, yang berarti Rx harus terhubung dengan Tx dan sebaliknya. Pada Gambar 4.13, Arduino Uno R3 + Dragino LoRa Shield pin D5 berperan sebagai Rx dan pin D4 sebagai Tx. Antara rangkaian LoRa Shield dengan ESP-12E, kedua perangkat dihubungkan oleh modul Level Converter Channel (LLC) 3.3V 5V. Modul ini berfungsi untuk menyesuaikan tegangan daya dari Arduino Uno R3 yang bekerja pada tegangan 5V ke ESP-12E yang bekerja pada tegangan 3.3V.



**Gambar 4. 13** Implementasi LoRa Gateway

## 4.5 Pengujian Sistem

### 4.5.1 Pengujian Sistem Penginderaan

Pada pengujian ini, sistem penginderaan yaitu rangkaian sensor tanah dan udara melakukan pengambilan data selama 24 jam dimulai dari pukul 00.00 hingga 23.00. Jumlah data yang diharapkan adalah sebanyak 24 data per jam. Hasil dari pengujian diharapkan sistem penginderaan dapat melakukan penginderaan dengan baik dan mampu mentransmisikan data dan menyimpan data ke basis data. Pengujian ini dilakukan dengan melakukan penginderaan pada tanah di dalam pot sebagai sampel dan pada ruangan di dalam rumah penulis.



**Gambar 4. 14** Penginderaan Tanah pada Pot

#### 4.5.1.1 Pengujian Rangkaian Sensor Tanah

##### 4.5.1.1.1 RS485-LoRa Wireless Station

Kode pemrograman untuk memprogram RS485-LoRa Wireless Station adalah kode pemrograman dengan bahasa pemrograman C++. Berikut *pseudo code* pemrograman yang akan di-*upload* ke RS485-LoRa Wireless Station.

LORA\_TRANSMITTE

RMULAI

SERTAKAN library: RadioLib, SPI, Wire, Adafruit\_GFX, Adafruit\_SSD1306, HardwareSerial

DEFINISIKAN konstanta:

SCREEN\_WIDTH =

128

SCREEN\_HEIGHT = 32

OLED\_RESET = -1

Display\_power = 19

RS485\_power = 18

RX\_RS485 = 23

TX\_RS485 = 22



```

DIO0 = 25
DIO1 = 26
LORA_RST = 33
LORA_CS = 32
SPI_MOSI = 13
SPI_MISO = 12
SPI_SCK = 14
ESP32_SDA = 4
ESP32_SCL = 5
FREQUENCY = 915.0
BANDWIDTH = 125.0
SPREADING_FACTOR = 9
CODING_RATE = 7
OUTPUT_POWER = 10
PREAMBLE_LEN = 8
GAIN = 0

```

DEFINE variabel global:

```

test_command = [0x01, 0x03, 0x00, 0x00, 0x00, 0x07, 0x04, 0x08]
test_response = []
moisture, tem, ph, P_value, N_value, K_value
moisture_value, tem_value, ph_value
P_float_value, N_float_value, K_float_value
groupID = "1"

```

INISIALISASI display

INISIALISASI radio LoRa dengan

parameter INISIALISASI MySerial dengan

parameter FUNCTION setup():

```

MULAI komunikasi serial dengan baud rate
115200 MULAI komunikasi MySerial dengan baud
rate 4800
TAMPILKAN "Menginisialisasi DISPLAY POWER & RS485
POWER..." SETEL pin Display_power dan RS485_power sebagai
OUTPUT TUNGGU 100 milidetik
SETEL pin Display_power dan RS485_power menjadi
HIGH TUNGGU 1 detik
TAMPILKAN "Menginisialisasi SSD1306 (Wire)..."
MULAI komunikasi Wire dengan ESP32_SDA dan
ESP32_SCL INISIALISASI display SSD1306
JIKA inisialisasi display gagal:
    TAMPILKAN "Gagal mengalokasikan SSD1306"

```

```

        AKHIR program
    TAMPILKAN "SSD1306 ditemukan"
    TAMPILKAN "Memulai SPI..."
    MULAI komunikasi SPI
    TAMPILKAN "[SX1276] Menginisialisasi
    ..."INISIALISASI radio LoRa dengan
    parameterJIKA inisialisasi berhasil:
        TAMPILKAN "Memulai radio LoRa: SUKSES!"
    ELSE:
        TAMPILKAN "Memulai radio LoRa: GAGAL!"
        AKHIR program
    TAMPILKAN "Menulis tes awal untuk sensor..."
    PANGGIL writeInitTestCommand()
FUNCTION loop():
    TUNGGU 5 detik
    TAMPILKAN "Mendapatkan nilai sensor berdasarkan test_command:"
    PANGGIL writeInitTestCommand()
    PANGGIL getSensorValue()
    TAMPILKAN "Mentransmisikan Data Sensor..."
    PANGGIL transmitData()
FUNCTION transmitData():
    BUAT paket dengan data
    sensor KIRIM paket
    menggunakan radio JIKA
    pengiriman berhasil:
        TAMPILKAN "Data sensor berhasil DIKIRIM!"
        TAMPILKAN paket
        TAMPILKAN "[SX1278] Datarate:", radio.getDataRate(), "bps"
    ELSE IF paket terlalu panjang:
        TAMPILKAN "terlalu panjang!"
    ELSE IF timeout pengiriman
    terjadi:
        TAMPILKAN
        "timeout!"ELSE:
        TAMPILKAN "gagal, kode", state
FUNCTION writeInitTestCommand():
    KIRIM test_command melalui MySerial
    INISIALISASI indeks i menjadi 0
    SELAMA MySerial memiliki data tersedia:
        BACA data dari MySerial ke dalam test_response[i]
        INKREMENTASI i

```

```

        YIELD
    TAMPILKAN test_response untuk debug
    TUNGGU 1 detik
    PANGGIL
callbackWriteStatus()FUNCTION
callbackWriteStatus():
    TAMPILKAN "Penulisan perintah tes awal selesai."
FUNCTION getSensorValue():
    TAMPILKAN "Mendapatkan Nilai Sensor berdasarkan test_command:"
    TAMPILKAN "Kelembapan, Suhu, pH:"
    PANGGIL writeInitTestCommand()
    HITUNG nilai kelembapan dari
test_responseHITUNG nilai suhu dari
test_response HITUNG nilai pH dari
test_response TAMPILKAN nilai yang
dihitung
    TAMPILKAN nilai pada SSD1306
    TUNGGU 1 detik
    TAMPILKAN "N, P, K:"
    HITUNG nilai N, P, K dari test_response
    TAMPILKAN nilai N, P, K
    TAMPILKAN nilai N, P, K pada SSD1306
END

```

Kode kemudian dikompilasi dan di-*upload* menggunakan Arduino IDE dengan *board* ESP32 Dev Module dan port sesuai dengan port yang terdeteksi. Setelah proses *upload* selesai, aktivasi dan proses pada *board* dapat di lihat pada tampilan Serial Monitor. Berikut pada Gambar 4.15 ditampilkan aktivasi dan proses pada *board* RS485-LoRa Wireless Station.

```
[SX1278] Datarate: 922.13 bps
Getting Sensor Value based on test_command:
1 3 14 1 67 1 55 0 0 0 45 0 0 0 0 0 0 241 98
Writing Init test_command done.
Getting Sensor Value based on test_command:
Moist, Temp, pH:
1 3 14 1 70 1 54 0 0 0 84 0 17 0 23 0 58 221 72
Writing Init test_command done.
326
32.60
31.00
8.40
N, P, K:
N= 17.00 mg/kg
P= 23.00 mg/kg
K= 58.00 mg/kg
Transmitting Sensor Data...
Sensor Data was SUCCESSFULLY TRANSMITTED!
T:31.00;H:32.60;PH:8.40;N:0;P:0;K:0/1
[SX1278] Datarate: 947.79 bps
```

**Gambar 4. 15** Aktivasi dan Proses pada *Board RS485-LoRa Wireless Station*

Pada Gambar 4.15 pertama-tama akan melakukan proses mengambil nilai sensor berdasarkan array `test_command[8] = {0X01, 0X03, 0X00, 0X00, 0X00, 0X07, 0X04, 0X08}`. Array `test_command` merupakan *bytes* perintah awal yang ditulis ke dalam sensor untuk memulai komunikasi maupun pengambilan data. Array `test_command` digunakan oleh fungsi `writeInitTestCommand()` yang berfungsi untuk menuliskan `test_command` kepada sensor dan kemudian membaca respons yang dikembalikan.

```
writeInitTestCommand() :
void writeInitTestCommand()
{
  MySerial.write(test_command, 8);
  int i = 0;
  while (MySerial.available() > 0)
  {
    test_response[i] = MySerial.read();
    i++;
    yield();
  }
  for (int j = 0; j < 19; j++)
  {
    Serial.print((int)test_response[j]);
    Serial.print(" ");
  }
}
```

```

}
Serial.print("\n");
delay(1000);
callbackWriteStatus();
}

```

Berdasarkan response yang dikembalikan, kemudian menghitung nilai sensor dengan fungsi *CalculateValue()* dengan test\_response sebagai parameter.

Setelah nilai sensor didapat, akhirnya data akan ditransmisikan ke LoRa Gateway dengan data yang di dapat. Data yang ditransmisikan pada Gambar 4.12 adalah T:31.00;H:32.60;PH:8.4:N:0;P:0;K:0/1.

#### 4.5.1.1.2 NodeMCU ESP2866 ESP-12 E

NodeMCU ESP2866 ESP-12 E (selanjutnya ditulis ESP-12E) menggunakan tipe *board* ESP2866 yang dapat diprogram menggunakan Arduino IDE dengan bahasa pemrograman sama dengan yang digunakan pada RS485-LoRa Wireless Station. Berikut adalah *pseudo code* pemrograman yang akan dikompilasi dan di-upload pada ESP-12E dengan nama dokumen *Sensor\_udara.ino*.

SENSOR\_UDARA

MULAI

SERTAKAN pustaka: Adafruit\_BMP280, ESP8266WiFi, PubSubClient

DEFINISIKAN MQTT Broker:

```

mqtt_broker = "103.163.139.XXX"
topik = "farmbot/sensors"
mqtt_username = "farmbot-mqtt"
mqtt_password = "farmbotXXX"
mqtt_port = 1883

```

DEFINISIKAN variabel global:

```

groupID = "5"
espClient
client (PubSubClient dengan espClient)
bmp (Adafruit_BMP280 untuk I2C)

```

FUNGSI connectToWifi():

MULAI koneksi WiFi dengan SSID dan password

CETAK "Connecting to wifi"

SELAMA status WiFi tidak terhubung:

```

    TUNGGU 500 milidetik
    CETAK status WiFi
    TUNGGU 500 milidetik

```

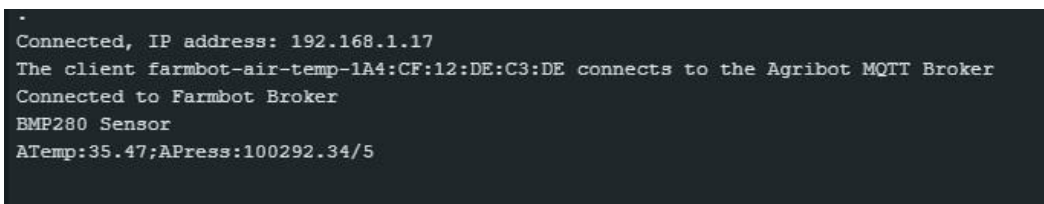
```

    CETAK "."
    CETAK "Connected, IP address: ", WiFi.localIP()
    FUNGSI connectToMQTTBroker():
        SELAMA client tidak terhubung:
            BUAT client_id dari "farmbot-air-temp-1" dan alamat MAC WiFi
            CETAK "The client ", client_id, " connects to the Agribot MQTT Broker"
            JIKA client berhasil terhubung:
                CETAK "Connected to Farmbot Broker"
            JIKA TIDAK:
                CETAK "failed with state ", client.state()
                TUNGGU 1000 milidetik
    FUNGSI callback(topik, payload, panjang):
        CETAK "Message arrived in topic: ", topik
        CETAK "Message:"
        UNTUK setiap byte dalam payload:
            CETAK byte sebagai karakter
        CETAK "-----"
    FUNGSI setup():
        MULAI komunikasi serial pada baud rate 9600
        PANGGIL connectToWifi()
        SET server untuk client MQTT dengan broker dan port
        SET callback untuk client
        PANGGIL connectToMQTTBroker()
        CETAK "BMP280 Sensor"
        JIKA inisialisasi BMP280 gagal:
            CETAK "Could not find a valid BMP280 sensor, check wiring or try a
different address!"
            TUNGGU tanpa batas
        SET konfigurasi untuk BMP280 dengan parameter default dari datasheet
    FUNGSI loop():
        JIKA status WiFi tidak terhubung:
            PANGGIL connectToWifi()
        JIKA client tidak terhubung:
            PANGGIL connectToMQTTBroker()
        BACA suhu dari BMP280
        BACA tekanan dari BMP280
        KONVERSI suhu dan tekanan ke string
        BUAT value_to_publish dari data suhu dan tekanan
        CETAK value_to_publish

```

```
PUBLISH value_to_publish ke topik MQTT
JALANKAN client.loop()
TUNGGU 10000 milidetik
SELESAI
```

Program menggunakan *library* Adafruit\_BMP280 untuk mengontrol sensor BMP280 dan *library* PubSubClient sebagai MQTT *publisher*. Setelah program dikompilasi dan di-*upload*, aktivasi dan proses pada *board* ESP-12E dapat dilihat pada tampilan Serial Monitor. Tampilan Serial Monitor dapat di lihat pada Gambar 4.16.



```

Connected, IP address: 192.168.1.17
The client farmbot-air-temp-1A4:CF:12:DE:C3:DE connects to the Agribot MQTT Broker
Connected to Farmbot Broker
BMP280 Sensor
ATemp:35.47;APress:100292.34/5

```

**Gambar 4. 16** Aktivasi dan Proses pada *Board ESP-12E*

Pertama-tama *board* mencoba terkoneksi ke WiFi. Setelah terkoneksi ke internet melalui WiFi, *board* mencoba terkoneksi ke MQTT broker dengan topik farmbot/sensor. Kemudian *board* akan melakukan pengambilan data dengan bantuan *library* Adafruit\_BMP280, hasilnya kemudian ditampilkan dan di-*publish* ke MQTT broker topik farmbot/sensor.

#### 4.5.1.1.3 LoRa Gateway

LoRa Gateway berfungsi untuk menerima transmisi data yang disebarkan oleh LoRa *transmitter* dan memastikan data harus dapat diterima. Oleh karena itu, LoRa Gateway dalam proses berkerjanya akan melakukan perulangan program dalam interval waktu 0,1 detik. Berikut adalah *pseudo code* untuk LoRa Gateway dengan nama dokumen *LoRa\_Gateway.ino*.

```
LORA_GATEWAY
MULAI
SERTAKAN library yang diperlukan
DEFINISIKAN pin untuk SX1276 dan SoftwareSerial
INISIALISASI modul radio SX1276
INISIALISASI SoftwareSerial
```

```
FUNGSI setup:
```

```

MULAI komunikasi serial
INISIALISASI modul radio SX1276 dengan pengaturan tertentu
CEK jika inisialisasi berhasil
    CETAK pesan sukses
    JIKA TIDAK
        CETAK pesan kesalahan dan BERHENTI
FUNGSI loop:
    TUNGGU sebentar
    CETAK pesan menunggu transmisi masuk
    TERIMA data sebagai string
    JIKA data berhasil diterima:
        CETAK pesan sukses dan data yang diterima
        CETAK informasi kekuatan sinyal
        KIRIM data yang diterima ke UART serial
    JIKA TIDAK
        TANGANI skenario kesalahan dan CETAK pesan yang sesuai
SELESAI

```

Dari kode pemrograman yang telah dikompilasi dan di-*upload* dapat dilihat aktivasi dan proses yang berjalan pada *board* menggunakan Serial Monitor pada IDE. Berikut tampilan aktivasi dan proses yang berjalan pada board yang dapat dilihat pada Gambar 4.17.

```

[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... success!
[SX1276] Data:      T:31.00;H:32.60;PH:8.40;N:0;P:0;K:0/1
[SX1276] RSSI:      -58.00 dBm
[SX1276] SNR:       13.25 dB
T:31.00;H:32.60;PH:8.40;N:0;P:0;K:0/1printing written data to uartSerial...
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!

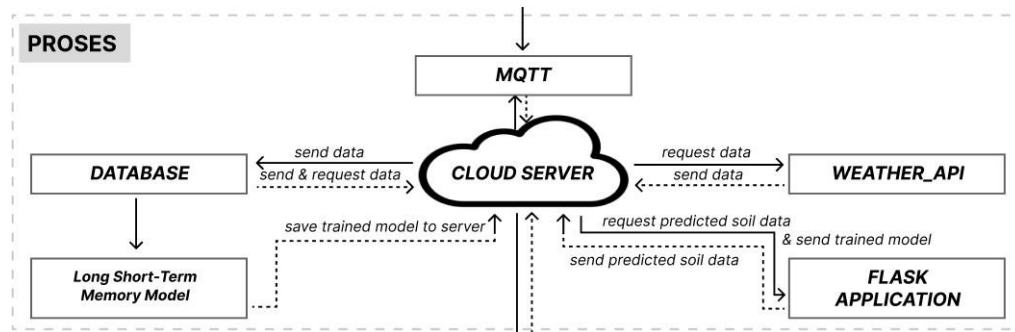
```

**Gambar 4. 17** Aktivasi dan Proses pada LoRa Gateway

#### 4.5.1.2 Pengujian Skema Penyimpanan Data

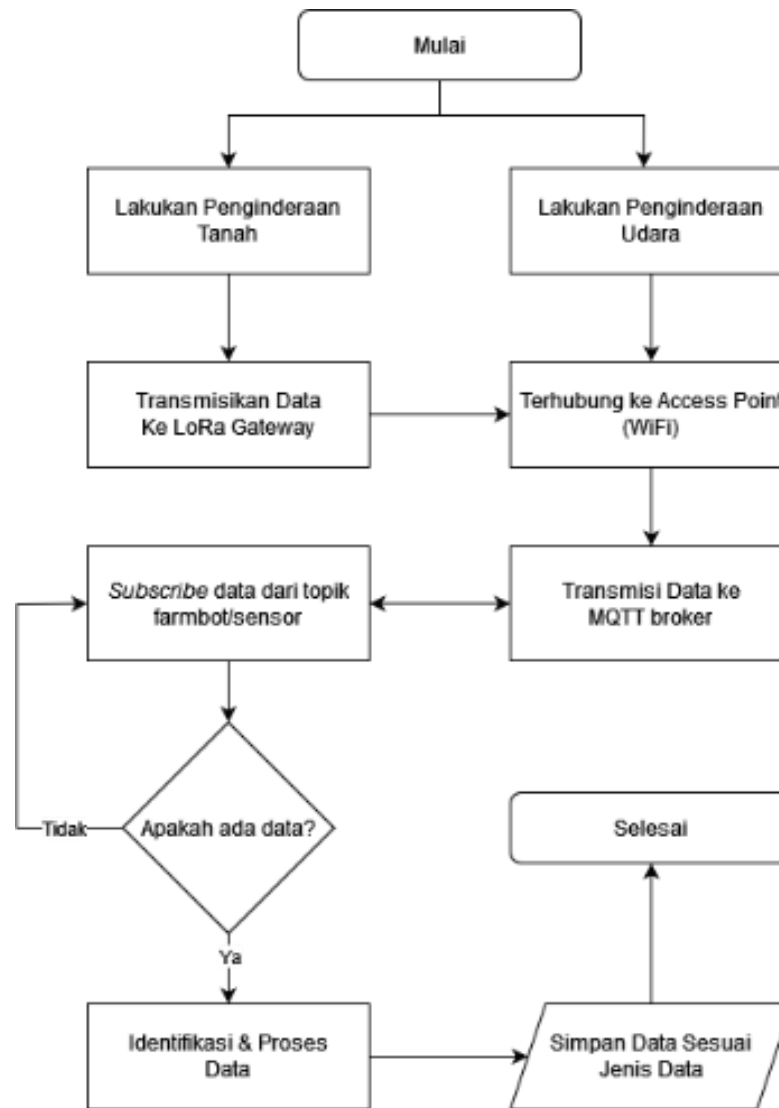


Proses penginderaan dianggap berhasil apabila data yang ditangkap oleh rangkaian sensor berhasil ditransmisikan dan pada akhirnya berhasil disimpan ke dalam basis data pada *cloud server*. Skema penyimpanan terdapat pada arsitektur umum bagian Proses yang telah di bahas pada Bab 3 pada Gambar 3.1. Arsitektur Umum bagian Proses dapat dilihat pada Gambar 4.18.



**Gambar 4. 18** Arsitektur Umum Bagian Proses

*Cloud server* menggunakan MQTT *client* akan melakukan *subscribe* ke MQTT broker dengan topik farmbot/sensor untuk mengambil data. Data berupa *payload* yang diterima akan ditangani oleh sebuah program Javascript. Program ini melakukan *subscribe* ke MQTT broker, kemudian mengidentifikasi data berupa *payload* yang masuk, memprosesnya, lalu menyimpannya ke dalam basis data. Skema penyimpanan data dapat dilihat pada Gambar 4.19 yang disajikan dalam bentuk flowchart.



**Gambar 4. 19** Flowchart Skema Penyimpanan Data

Pengujian skema penyimpanan data dapat dilakukan dimulai dari proses penginderaan data, terhubung ke internet melalui *access point (WiFi)*, sampai proses di sisi *cloud server* yaitu *subscribe* data ke topik farmbo/sensor, kemudian mengidentifikasi data dan memprosesnya, hingga menyimpan data ke basis data.

#### 4.5.1.2.1 Penginderaan Data Sensor Tanah dan Udara

Pengujian penginderaan data sensor tanah dapat dilihat pada Gambar 4.20 berikut:

```

[ SX1278 ] Datarate:          922.13 bps
Getting Sensor Value based on test_command:
1 3 14 1 67 1 55 0 0 0 45 0 0 0 0 0 241 98
Writing Init test_command done.
Getting Sensor Value based on test_command:
Moist, Temp, pH:
1 3 14 1 70 1 54 0 0 0 84 0 17 0 23 0 58 221 72
Writing Init test_command done.
326
32.60
31.00
8.40
N, P, K:
N= 17.00 mg/kg
P= 23.00 mg/kg
K= 58.00 mg/kg
Transmitting Sensor Data...
Sensor Data was SUCCESSFULLY TRANSMITTED!
T:31.00;H:32.60;PH:8.40;N:0;P:0;K:0/1
[ SX1278 ] Datarate:          947.79 bps

```

**Gambar 4. 20** Pengujian Penginderaan Sensor Tanah

Hasil dari pengujian sensor tanah mendapatkan nilai sensor dengan suhu tanah (T) 31.0, kelembaban (H) 32.6, pH (PH) 8.40, dan N P K dengan nilai 0. Nilai 0 pada N P K di sini kemungkinan besar disebabkan oleh kesalahan rangkaian sensor saat membaca nilai kondisi tanah.

Kemudian menguji penginderaan data sensor udara, dapat dilihat pada Gambar 4.21 berikut :

```

.
Connected, IP address: 192.168.1.17
The client farmbot-air-temp-1A4:CF:12:DE:C3:DE connects to the Agribot MQTT Broker
Connected to Farmbot Broker
BMP280 Sensor
ATemp:35.47;APress:100292.34/5

```

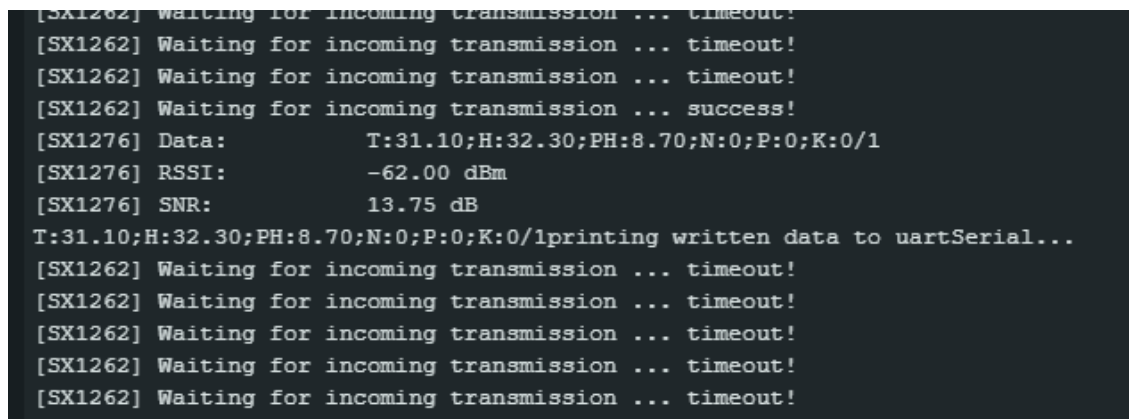
**Gambar 4. 21** Pengujian Penginderaan Sensor Udara

Rangkaian sensor pertama-tama mencoba melakukan koneksi ke WiFi. Setelah berhasil terkoneksi, mencoba melakukan koneksi ke MQTT broker. Jika berhasil, rangkaian akan melakukan proses penginderaan dan mem-*publish* data yang didapat ke MQTT broker dengan topik farmbot/sensor.

Hasil dari pengujian sensor udara ini adalah mendapatkan nilai sensor dengan nilai suhu udara (ATemp) 35.47 dan nilai tekanan udara (APress) 100292.34 dan berhasil mentransmisikan data ke MQTT broker.

#### 4.5.1.2.2 Transmisi Data Sensor Tanah ke LoRa Gateway

LoRa Gateway akan mencoba menangkap data berupa pesan yang di-broadcast pada frekuensi yang sama dengan yang digunakan LoRa Gateway. Frekuensi yang digunakan oleh LoRa Gateway adalah 915.0 MHz. Berikut Gambar 4.22 menampilkan pengujian LoRa Gateway mencoba menangkap pesan dari rangkaian sensor tanah :



```
[SX1262] waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... success!
[SX1276] Data:      T:31.10;H:32.30;PH:8.70;N:0;P:0;K:0/1
[SX1276] RSSI:      -62.00 dBm
[SX1276] SNR:       13.75 dB
T:31.10;H:32.30;PH:8.70;N:0;P:0;K:0/1printing written data to uartSerial...
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
[SX1262] Waiting for incoming transmission ... timeout!
```

**Gambar 4. 22** Pengujian LoRa Gateway

LoRa Gateway melakukan percobaan menangkap pesan setiap 0,5 detik sekali dengan harapan supaya tidak ada data yang ditransmisikan oleh rangkaian sensor tidak diterima oleh LoRa Gateway.

Hasil percobaan ini mendapatkan respons berupa pesan dengan data T:31.10;H:32.30;PH:8.70;N:0;P:0;K:0/1 dengan RSSI (Received Signal Strength Indication) -62.00 dBm dan SNR (Signal-to-Noise Ratio) 13,75dB. Nilai RSSI dapat digunakan untuk mengindikasikan seberapa baik *receiver* (LoRa Gateway) dapat “mendengar” sebuah sinyal dari *transmitter* (rangkaian sensor tanah). RSSI -62.00 mengindikasikan *receiver* dapat menerima sinyal dari *transmitter* dengan baik. Nilai SNR adalah rasio antara sinyal daya yang diterima dan tingkat daya dasar kebisingan (*noise*). Semakin kecil nilai SNR semakin baik sinyal yang dapat diterima oleh *receiver*. Data yang diterima akan langsung ditransmisikan ke MQTT Broker.

#### 4.5.1.2.3 Subscribe Data pada Topik farmbot/sensor

Cloud server melakukan *subscribe* ke MQTT broker topik farmbot/sensor, mendapatkan data berupa *payload*, mengidentifikasi data dan memprosesnya, lalu menyimpannya pada basis data. Tahap proses ini dilakukan oleh program Javascript yang berjalan pada *cloud server*.

#### 4.5.1.2.4 Identifikasi dan Pemrosesan Data

Pada *cloud server*, akan mengidentifikasi pesan yang didapat melalui panjang pesan. Contoh format pesan berdasarkan jenis penginderaan adalah sebagai berikut :

- Penginderaan tanah: T:28.44;H:56.91;PH:7.95;N:31:P:42:K:104/1
- Penginderaan udara: ATemp:30.12;APress: 100292.34/5

Pesan tersusun diawali dengan atribut data diikuti dengan titik dua (:) kemudian diikuti dengan nilai data. Setiap nilai atribut data dipisahkan dengan tanda titik koma (;), dan tanda garis miring (/) di akhir pesan diikuti dengan nilai grup sensor sebagai identifikasi lokasi sensor. Sehingga jika pesan dipecah akan didapatkan data T dengan nilai 28.44, H dengan nilai 56.91, PH dengan nilai 7.59 dan seterusnya. Panjang pesan dapat dihitung dengan menjadikan tanda titik dua (:) sebagai pemisah. Misalnya pada pesan dari penginderaan udara, jika pesan dipisah-pisah menggunakan tanda titik dua (:) akan didapatkan ATemp, 30.12, APress, 100292.34, dan 5 yang berarti memiliki panjang data 5. Sedangkan data tanah memiliki panjang data 13. Dengan panjang data inilah yang akan dijadikan pembeda dan acuan penyimpanan data agar sesuai. Contoh cuplikan kode pemrograman dari program Javascript yang mengidentifikasi pesan ini adalah sebagai berikut :

```
// Air Temperature and Pressure Sensor
if (data.length == 5) {
  console.log("This is from air temp pressure sensor");
  database.table('air_temp_press_t').insert({
    s_group_id: data.slice(-1)[0],
    ATemp: data[1],
    APress: data[3],
    time: Math.floor(Date.now() / 1000)
  }).then(response => {
    console.log(response);
  }).catch(err => { console.log(err) });
}
```

Baris kode `if (data.length == 5)` merupakan pernyataan kondisional yang dapat dipaparkan sebagai “jika panjang data sama dengan 5 dengan tipe data yang sama dengan 5”. Baris kode setelahnya akan dieksekusi jika pernyataan kondisional terpenuhi. Jika terpenuhi, pada baris kode `database.table('air_temp_press_t').insert()` berarti akan melakukan perintah basis data yaitu “INSERT” yang berarti memasukkan data ke dalam tabel. Tabel yang akan digunakan untuk menyimpan data pada baris kode

ini adalah “air\_temp\_press\_t”. Dari pernyataan tersebut, dapat ditarik kesimpulan bahwa, jika panjang data adalah sepanjang 5, maka data akan disimpan ke dalam tabel “air\_temp\_press\_t” menyesuaikan dengan kolom atribut yang ada.

Berikut contoh hasil identifikasi dan proses data yang ditampilkan pada Gambar 4.23.

```
This is from air temp pressure sensor
OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 4762,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
Received payload: ATemp:35.47;APress:100292.34/5
```

**Gambar 4. 23** Contoh Hasil Identifikasi dan Proses Data

#### 4.5.1.2.5 Penyimpanan Data ke Basis Data

Setelah data berhasil diidentifikasi dan diproses, kemudian data disimpan ke dalam basis data. Basis data yang digunakan disini bernama ‘farmbot’ dengan berbagai tabel data, diataranya ada tabel lora\_npk\_t dan air\_temp\_press\_t. Data dari sensor tanah akandisimpan pada tabel lora\_npk\_t dan data dari sensor udara akan disimpan pada tabel air\_temp\_press\_t. Sebagai contoh percobaan, sebelumnya data yang berhasil di- *subscribe* adalah data dari sensor udara dengan data suhu udara 35.47 dan tekanan udara100292.34. Data ini disimpan dengan indikasi berupa notifikasi OkPacket yang dapat dilihat pada Gambar 4.20. OkPacket memberi informasi ‘inserId: 4762’ yang berarti data disimpan dengan id 4762. Data dengan id 4762 dapat dilihat pada tabel air\_temp\_press\_t yang dapat disimak pada Gambar 4.24.

```
MariaDB [farmbot]> select val_id as id, ATemp, APress, from_unixtime(time) as time from air_t
emp_press_t order by time desc limit 5;
+-----+-----+-----+-----+
| id | ATemp | APress | time |
+-----+-----+-----+-----+
| 4764 | 35.44 | 100278.96 | 2024-06-05 13:59:24 |
| 4763 | 35.47 | 100292.34 | 2024-06-05 13:54:23 |
| 4762 | 35.37 | 100290.37 | 2024-06-05 13:53:50 |
| 4761 | 26.22 | 100312.21 | 2024-06-04 00:15:00 |
| 4760 | 26.22 | 100312.21 | 2024-06-04 00:14:01 |
+-----+-----+-----+-----+
5 rows in set (0.009 sec)
```

**Gambar 4. 24** Data yang Berhasil Disimpan pada Basis Data

#### 4.5.1.3 Pengujian Pengambilan Data selama 24 Jam

Pengujian pengambilan data dilakukan pada 5 Juni 2024 mulai pukul 00.00 hingga 23.00 WIB. Rangkaian sensor dihidupkan tepat pada pukul 00.00. Proses pengambilan data biasanya diproses selama 6 hingga 9 detik *runtime*. Berikut pengujian pengambilan data pada Gambar 4.25 oleh rangkaian sensor tanah dan Gambar 4.26 oleh rangkaian sensor udara.

```
MariaDB [farmbot]> select val_id, T, H, PH, N, P, K, from_unixtime(time) as time from lora_npk_t order by time desc limit 24;
```

val_id	T	H	PH	N	P	K	time
9589	26.2	39.8	8.5	32	38	98	2024-06-06 23:00:04
9588	26.6	40.7	8.5	0	0	0	2024-06-06 22:00:04
9587	26.3	41.1	8.5	32	38	98	2024-06-06 21:00:04
9586	28.2	41.7	8.5	28	38	98	2024-06-06 20:00:04
9583	26.3	41.1	8.5	32	38	98	2024-06-06 19:00:07
9580	27.8	44.4	8.5	29	38	98	2024-06-06 18:00:09
9579	28.2	45.1	8.5	0	0	0	2024-06-06 17:00:09
9578	28.6	46.5	8.5	30	38	98	2024-06-06 16:00:08
9577	28.8	47.7	8.5	29	38	98	2024-06-06 15:00:08
9576	29.1	49.6	8.5	0	0	0	2024-06-06 14:00:08
9575	28.8	52.3	8.5	28	38	98	2024-06-06 13:00:08
9574	29.2	54.8	8.5	0	0	0	2024-06-06 12:00:08
9573	28.4	56.8	8.5	28	38	98	2024-06-06 11:00:08
9572	27.6	58.7	8.5	29	38	98	2024-06-06 10:00:07
9571	27.7	59.5	8.5	33	38	98	2024-06-06 09:00:07
9570	26.6	60.4	8.5	31	38	98	2024-06-06 08:00:07
9569	26.6	61.1	8.5	0	0	0	2024-06-06 07:00:07
9568	26.2	63.2	8.5	0	0	0	2024-06-06 06:00:09
9567	0	64.6	8.5	27	38	98	2024-06-06 05:00:08
9566	25.4	65.6	8.5	28	38	98	2024-06-06 04:00:08
9565	25.7	65.9	8.5	26	38	98	2024-06-06 03:00:09
9564	0	66.4	8.5	0	0	0	2024-06-06 02:00:07
9563	24.9	66.8	8.5	0	0	0	2024-06-06 01:00:08
9562	25.6	67.4	8.5	27	38	98	2024-06-06 00:00:07

24 rows in set (0.015 sec)

**Gambar 4. 25** Pengujian Pengambilan Data Tanah

```
MariaDB [farmbot]> select val_id as id, ATemp, APress, from_unixtime(time) from air_temp_press_t order by time desc limit 24;
```

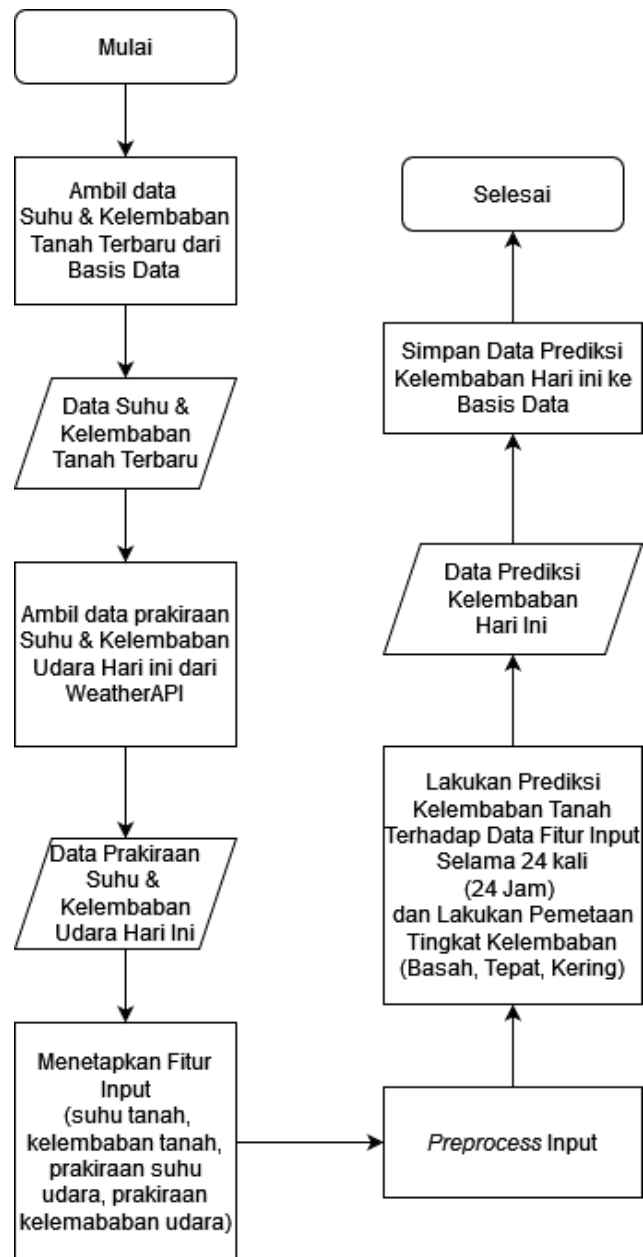
id	ATemp	APress	from_unixtime(time)
4813	25.4	100291.53	2024-06-06 23:00:05
4812	25.9	100292.49	2024-06-06 22:00:04
4811	26.5	100292.53	2024-06-06 21:00:04
4810	27.2	100293.25	2024-06-06 20:00:04
4809	27.7	100293.24	2024-06-06 19:00:04
4808	27.2	100293.12	2024-06-06 18:00:04
4806	27.4	100292.97	2024-06-06 17:00:04
4805	28.8	100294.25	2024-06-06 16:00:04
4804	29.7	100294.27	2024-06-06 15:00:04
4803	30	100294.22	2024-06-06 14:00:04
4802	28.3	100293.34	2024-06-06 13:00:04
4801	28.8	100293.63	2024-06-06 12:00:04
4800	28.5	100293.63	2024-06-06 11:00:04
4799	28.1	100293.24	2024-06-06 10:00:04
4798	27.5	100292.85	2024-06-06 09:00:04
4797	27.2	100292.86	2024-06-06 08:00:04
4796	26.1	100292.73	2024-06-06 07:00:04
4795	25.9	100292.72	2024-06-06 06:00:05
4794	25.6	100292.76	2024-06-06 05:00:04
4793	25.5	100293.52	2024-06-06 04:00:04
4792	25.9	100293.46	2024-06-06 03:00:04
4791	26.4	100293.52	2024-06-06 02:00:03
4790	26.5	100293.23	2024-06-06 01:00:03
4789	26.4	100292.34	2024-06-06 00:00:09

24 rows in set (0.004 sec)

**Gambar 4. 26** Pengujian Pengambilan Data Udara

#### 4.5.2 Pengujian Sistem Penjadwalan Penyiraman

Pengujian sistem penjadwalan dilakukan pada aplikasi prediksi Flask. Langkah proses prediksi kelembaban tanah yang akan dijadikan acuan penjadwalan penyiraman dapat dilihat pada flowchart pada Gambar 4.27.



**Gambar 4. 27** Flowchart Sistem Penjadwalan Penyiraman

Setelah data prediksi berhasil disimpan, kemudian data ini akan digunakan oleh sistem robot pertanian untuk menjadwalkan penyiraman. Sistem robot pertanian akan melakukan pengecekan pada tabel hasil prediksi kelembaban pada basis data.



#### 4.5.2.1 *Pengujian Model Prediksi Long Short-Term Memory*

Model prediksi LSTM dimulai dengan mengimpor *library* dan *dependency* yang dibutuhkan untuk menjalankan program. Kemudian melakukan impor *dataset*. *Dataset* yang diimpor kemudian diambil kolom-kolom yang akan digunakan sebagai variabel yang akan dipertimbangkan (variabel input) untuk memprediksi variabel target (kelembaban tanah). Prediksi variabel target pada model prediksi ini bersifat *autoregressive*. *Autoregressive* (AR) adalah kelas model pembelajaran mesin (ML) yang secara otomatis memprediksi elemen rangkaian berikutnya dengan melakukan pengukuran dari masukan sebelumnya dalam rangkaian tersebut (Binkowski *et al.*, 2018). Kemudian melakukan *preporcessing* data. Setelah itu, menentukan variabel input (X) dan variabel target (y). Kemudian melakukan *reshape* pada data input menjadi berbentuk deretan (*sequence*) yang cocok untuk LSTM. Setelah selesai, data kemudian dipisah (*splitting*) menjadi data *training* dan data *testing* dengan rasio data yang akan diuji. Kemudian menginisialisasi LSTM dengan *hyperparameter* sebagai model. Lalu meng-*compile* model dengan *optimizer* = *adam* (*Adaptive Moment Estimation*) dengan *loss function* = *MSE*. Kemudian lakukan *Training* (pelatihan) data. Model yang telah selesai diproses kemudian di-*dump* menggunakan Joblib, berikut juga *Scaler*. Kemudian evaluasi model yang telah dibuat dengan MSE, RMSE, dan R2 Score. Sebelum pengujian model prediksi dilakukan, penulis akan mencoba melakukan pencarian *hyperparameter* model LSTM terbaik. Hal ini bertujuan agar prediksi dapat bekerja maksimal karena model dan dataset yang cocok.

##### 4.5.2.1.1 *Optimisasi hyperparameter model LSTM*

Optimisasi *hyperparameter* model dilakukan dengan proses *hyperparameter tuning* menggunakan Keras Tuner. Keras Tuner adalah sebuah kerangka kerja pengoptimalan *hyperparameter* yang mudah digunakan untuk memecah masalah dalam melakukan pencarian *hyperparameter* yang memanfaatkan algoritma untuk menemukan nilai *hyperparameter* terbaik ([blog.tensorflow.org](http://blog.tensorflow.org)). *Hyperparameter* yang didapatkan dari Keras Tuner ini adalah LSTM Units dan *Learning Rate*.

Pengujian menggunakan Keras Tuner membutuhkan *model builder function* dengan parameter-parameter pengujian, seperti minimum dan maksimum unit layer, fungsi aktivasi, *input shape*, dense layer, *optimizer* (*learning rate*), dan fungsi *loss*. Berikut pada Tabel 4.4 dipaparkan parameter-parameter *model builder function*.

**Tabel 4. 4** Parameter *Model Builder Function*.

Nama Parameter	Nilai Parameter
Unit Layer	32 hingga 512 units
Fungsi Aktivasi	ReLU
Input Shape	Seq_length = 24, shape = 3D array
Dense Layer	1
Optimizer (Learning Rate)	Adam, dengan nilai 0.0001, 0.001, dan 0.01.
Fungsi Loss	MSE

Pengujian ini akan dilakukan sebanyak 10 kali uji coba (*trials*). Hasil dari setiap uji coba akan dipaparkan pada Tabel 4.5.

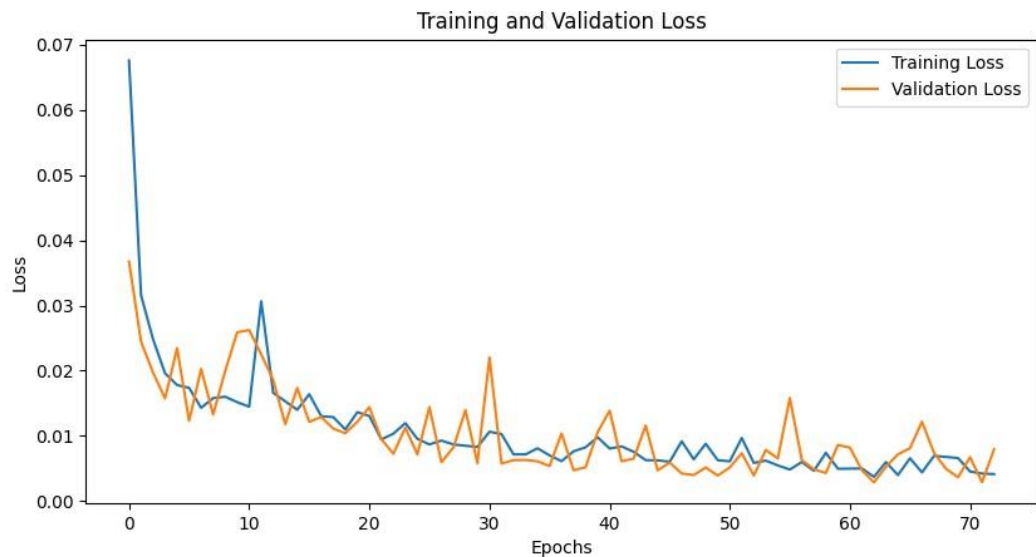
**Tabel 4. 5** Hasil Pengujian *Hyperparameter* Keras Tuner pada Model.

Trial	Layer Units	Learning Rate	Validation Loss	Training Loss	Best Step
1.	288	0.0001	0.0086614	0.0119673	43
2.	128	0.0001	0.0121574	0.0157683	43
3.	224	0.0001	0.0092700	0.0131215	49
4.	288	0.01	0.0065719	0.0106462	25
5.	320	0.0001	0.0069251	0.0101497	47
6.	480	0.0001	0.0077142	0.0119704	35
<b>7.</b>	<b>416</b>	<b>0.001</b>	<b>0.0048852</b>	<b>0.0071017</b>	<b>35</b>
8.	32	0.001	0.0093497	0.0114945	41
9.	96	0.0001	0.0106806	0.0134354	49
10.	320	0.01	0.0190295	0.0232861	14

Keras Tuner melakukan pengujian mencari *hyperparameter* terbaik dengan melakukan pengujian dengan parameter-parameter *model builder function* secaraacak. Hasil pengujiannya pada Tabel 4.5 menunjukkan hasil Validation Loss dan Training Loss terhadap *hyperparameter* yang diuji seperti Layer Units dan Learning Rate. Validation Loss dan Training Loss adalah metrix utama yang menunjukkan seberapa baik kinerja model terhadap data yang tak terlihat dan data *training*. Dari

hasil 10 kali percobaan pada Tabel 4.5 menunjukkan bahwa Trial ke 7 memiliki nilai Validation Loss dan Training Loss terendah, yaitu 0.0048852 dan 0.0071017 berurutan.

Dari Trial ke 7, perubahan Training dan Validation Loss dari epoch ke epoch dapat dilihat pada Gambar 4.28.



**Gambar 4. 28** Perubahan Training dan Validation Loss Trial ke 7

#### 4.5.2.1.2 Optimisasi Rasio Pemisahan Data Training dan Testing

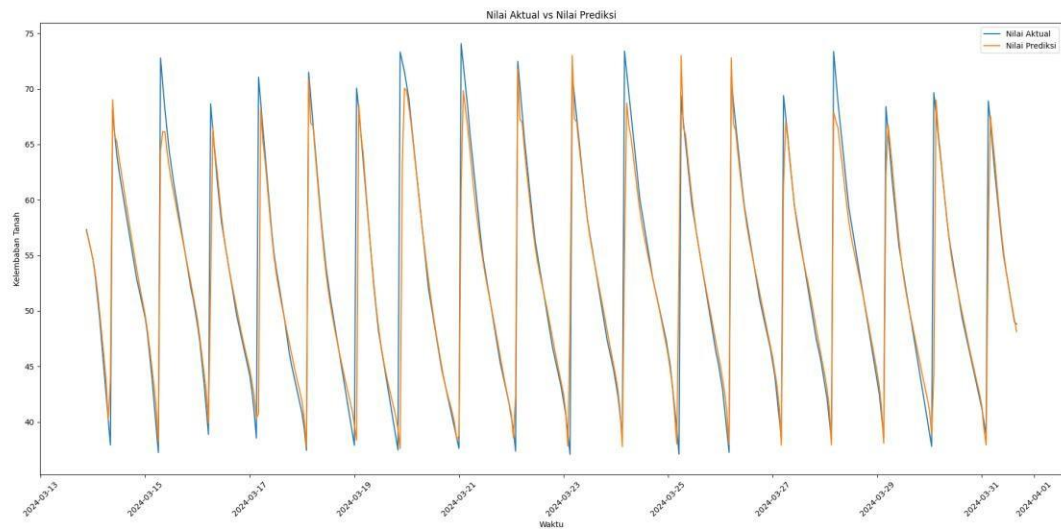
Dalam pengujian optimisasi rasio pemisahan data *training* dan *testing*, terdapat tiga skenario proses pengujian. Masing-masing pengujian akan menguji pengaruh perbandingan rasio pemisahan terhadap performa model dengan menggunakan *hyperparameter* Layer Units 416 dan Learning Rate 0.001. Skenario 1 menggunakan rasio pemisahan *training* dan *testing* 80: 20, skenario 2 dengan rasio pemisahan 70:30, dan skenario 3 dengan rasio pemisahan 60:40.

##### 4.5.2.1.2.1 Skenario 1

Skenario 1 menggunakan rasio pemisahan *training* dan *testing* dengan rasio 80 berbanding 20, yang berarti 80% dari dataset akan dijadikan untuk *training* dan 20% sisanya akan dijadikan *testing*. Berikut Tabel 4.6 menunjukkan hasil evaluasi dari skenario 1.

**Tabel 4. 6** Hasil Evaluasi Model dengan Skenario 1

Layer Units	Learning Rate	RMSE	R2 Score	First Prediction
416	0.001	3.7058	0.84	57.21

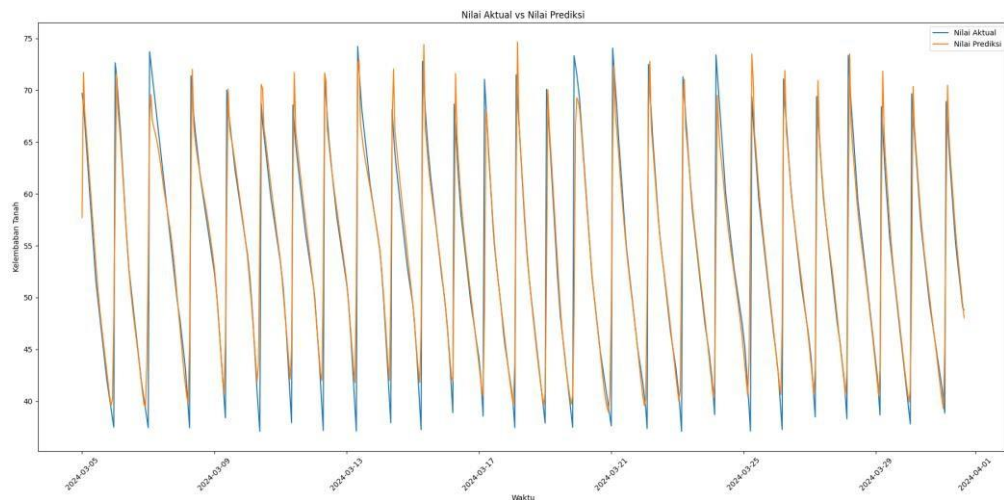
**Gambar 4. 29** Visualisasi Prediksi pada Skenario 1

#### 4.5.2.1.2.2 Skenario 2

Skenario 2 menggunakan rasio pemisahan *training* dan *testing* dengan rasio 70 berbanding 30, yang berarti 70% dari dataset akan dijadikan untuk *training* dan 30% sisanya akan dijadikan *testing*. Berikut Tabel 4.7 menunjukkan hasil evaluasi dari skenario 1.

**Tabel 4. 7** Hasil Evaluasi Model dengan Skenario 2

Layer Units	Learning Rate	RMSE	R2 Score	First Prediction
416	0.001	3.3939	0.87	57.71



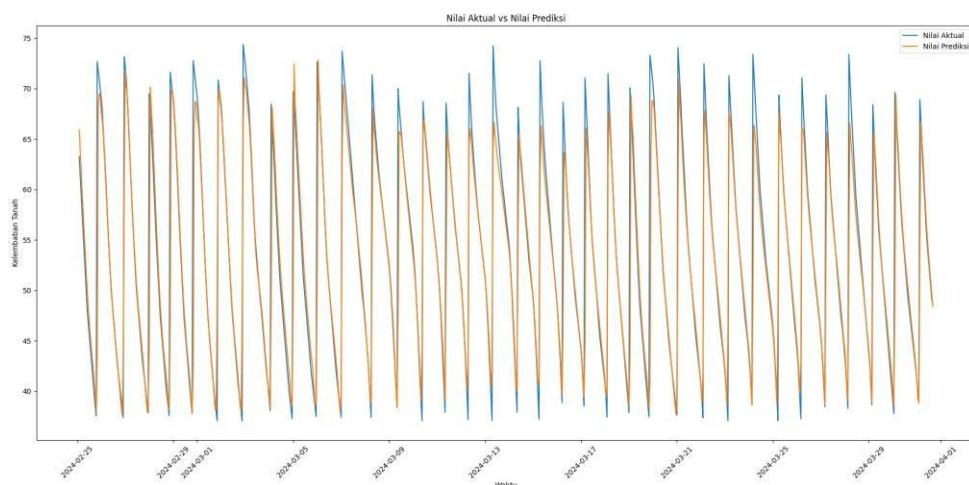
**Gambar 4. 30** Visualisasi Prediksi pada Skenario 2

#### 4.5.2.1.2.1 Skenario 3

Skenario 3 menggunakan rasio pemisahan *training* dan *testing* dengan rasio 60 berbanding 40, yang berarti 60% dari dataset akan dijadikan untuk *training* dan 40% sisanya akan dijadikan *testing*. Berikut Tabel 4.8 menunjukkan hasil evaluasi dari skenario 1.

**Tabel 4. 8** Hasil Evaluasi Model dengan Skenario 3

Layer Units	Learning Rate	RMSE	R2 Score	First Prediction
416	0.001	3.9698	0.84	65.93



**Gambar 4. 31** Visualisasi Prediksi pada Skenario 2

Dari pengujian performa model terhadap penentuan perbandingan nilai rasio perbandingan data *training* dan *testing*, hasil dari Skenario 1, 2, dan 3 dapat dikumpulkan agar dapat diamati. Kumpulan data hasil dari ketiga skenario dapat dilihat pada Tabel 4.9.

**Tabel 4. 9** Kumpulan Data Hasil Percobaan Dari Ketiga Skenario

Skenario	Layer Units	Learning Rate	RMSE	R2 Score	First Prediction
1	416	0.001	3.7058	0.84	57.21
<b>2</b>	<b>416</b>	<b>0.001</b>	<b>3.3939</b>	<b>0.87</b>	<b>57.71</b>
3	416	0.001	3.9698	0.84	65.93

Dari Tabel 4.9, dapat disimak bahwa RMSE dan R2 Score menjadi acuan dalam menentukan hasil dari skenario mana yang terbaik. Nilai RMSE dengan nilai terendah mengindikasikan performa yang terbaik, sementara nilai R2 score tertinggi mengindikasikan performa yang terbaik. Pada Tabel 4.9 dapat dilihat nilai RMSE terendah ada pada Skenario 2 dengan nilai 3.3939. Sementara nilai R2 score tertinggi berasal dari Skenario 2 dengan nilai 0.87. Dari data ini, dapat ditarik kesimpulan bahwa hasil dari percobaan pada Skenario 2 memiliki perbandingan rasio pemisahan data *training* dan *testing* terbaik, yaitu 70:30.

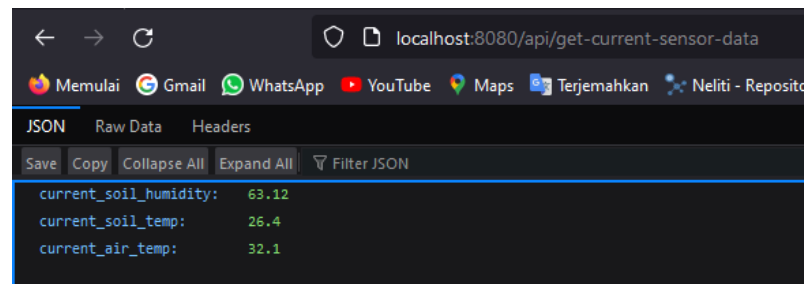
Dari proses optimisasi *hyperparameter* dan optimisasi rasio pemisahan data *training* dan *testing*, didapatkan parameter terbaik untuk model prediksi LSTM yaitu LSTM Layer units 416 dan Learning Rate 0.001 dengan rasio pemisahan data *training* dan *testing* 70 berbanding 30.

#### 4.5.2.2 Pengujian Aplikasi Prediksi Flask

Untuk menguji aplikasi prediksi Flask, dibutuhkan proses pengambilan data dari basis data sistem dan data prakiraan cuaca dari WeatherAPI. Kemudian dilanjutkan dengan menetapkan fitur input, preprocessing data, dan prediksi nilai kelembaban lalu memetakan tingkat kelembaban, lalu menyimpan data hasil prediksi. Pengujian dianggap berhasil apabila proses pengujian hingga penyimpanan data hasil prediksi berhasil.

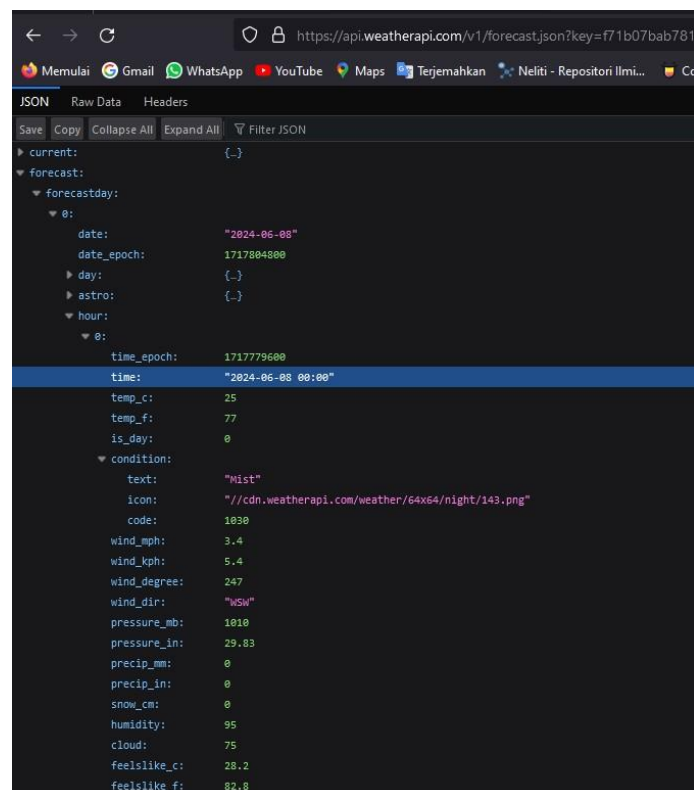
#### 4.5.2.2.1 Pengambilan Data

Pengambilan data kelembaban dan suhu terbaru dari basis data sistem dilakukan melalui API yang menjembatani aplikasi prediksi Flask dengan aplikasi pemantauan berbasis web. Contoh pengambilan data melalui API dapat dilakukan pada browser dengan mengakses URL *http://localhost:8080/api/get-current-sensor-data*.



**Gambar 4. 32** Pengambilan Data Kelembaban dan Suhu Terbaru melalui API

Data yang didapat berformat JSON. Data ini kemudian diekstrak untuk diambil nilai-nilainya. Pengambilan data juga dilakukan pada WeatherAPI. Data didapatkan dengan mengakses API dengan parameter-parameter yang didukung dan key pengguna WeatherAPI. Berikut pada Gambar 4.33 contoh respons data yang didapat ketika API diakses melalui browser.



**Gambar 4.33** Contoh Respons Data WeatherAPI

Dari respons data yang didapat, kemudian mengekstrak data *'temp\_c'* sebagai prakiraan suhu udara dan *'humidity'* sebagai prakiraan kelembaban udara setiap jamnya.

#### 4.5.2.2.2 Menetapkan Fitur Input

Fitur input pada aplikasi prediksi ini serupa mengikuti model LSTM, yakni:

```
input_features = ['SH', 'ST', 'AT', 'hour']
```

Keterangan :

SH = soil humidity

(kelembaban tanah) ST = soil

temperature (suhu tanah)

AT = air temperature (suhu

udara)

hour = waktu

#### 4.5.2.2.3 Preprocessing Data

Data input akan dilakukan penskalaan menggunakan MinMaxScaler yang di-dump ke LSTM model agar format input dan dataset sama.

#### 4.5.2.2.4 Prediksi Kelembaban Tanah

Proses prediksi kelembaban tanah dapat disimak pada penggalan *source code* berupa fungsi *forecastHumidity()* di bawah ini :

```
def forecastHumidity(parse_json, model, scaler, seq_length, num_features,
c_s_humidity, c_s_temp):
    save = []
    forecast_weather = parse_json['forecast']['forecastday'][0]['hour']
    for elemen in forecast_weather:
        timestamp = int(datetime.strptime(elemen['time'], '%Y-%m-%d
%H:%M').timestamp()) * 1000
        datetime_obj = datetime.fromtimestamp(timestamp / 1000)
        hour = datetime_obj.hour
        minute = datetime_obj.minute
        current_soil_humidity = c_s_humidity
        a_temp = elemen['temp_c']
        a_humidity = elemen['humidity']
        input_data = [[current_soil_humidity, current_soil_temp, a_temp, hour]]
        input_data_scaled = preprocess_input(input_data)
        if len(save) >= seq_length:
            save.append(None)
            save.append({"timestamp": timestamp, "air_temp": a_temp,
"air_humidity": a_humidity, "soil_humidity": None, "humidity_state":
None})
        else:
            input_seq = np.zeros((1, seq_length, num_features))
            input_seq[0, :, :] = input_data_scaled

    # Buat prediksi untuk semua Fitur
    predictions_scaled = model.predict(input_seq)
    predictions_5f = np.repeat(predictions_scaled, 5, axis=1)
    # Inverse transform hasil prediksi
    predictions_inverse = scaler.inverse_transform(predictions_5f)
    # Ekstrak hasil prediksi 'SH' (Kelembaban tanah)
    humidity_state = mapHumidity(predicted_sh_inverse)
```

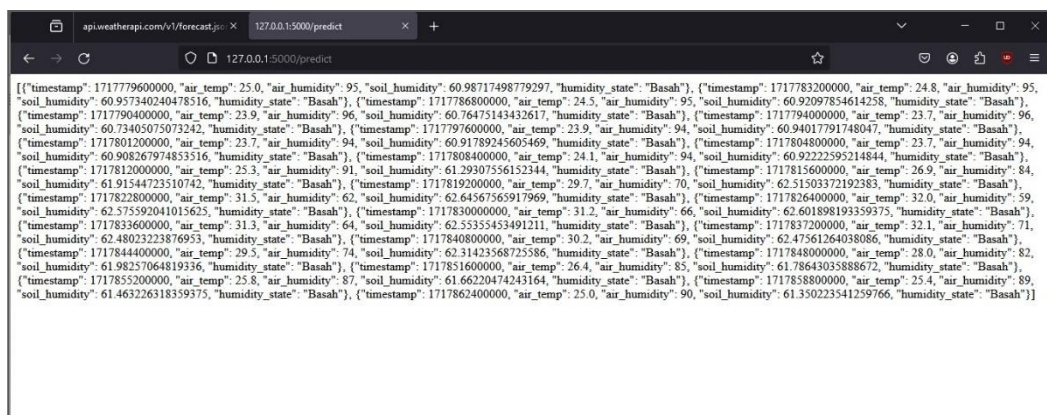


```

predicted_sh_inverse = predictions_inverse[0][0]
save.append({"timestamp": timestamp, "air_temp": a_temp, "air_humidity":
a_humidity, "soil_humidity": float(predicted_sh_inverse),
"humidity_state": humidity_state})
# Convert save menjadi format JSON-serializable (list)
save_serializable = [item if not isinstance(item, np.float32) else
float(item) for item in save]
return json.dumps(save_serializable)

```

Dari penggalan *source code* fungsi *forecastHumidity()* di atas dengan tulisan bercetak merah menyatakan perintah lakukan prediksi model dengan metode *predict()* yang menggunakan *input\_seq*. Fungsi *forecastHumidity()* ini mengembalikan nilai hasil prediksi dalam format JSON-serializable berbentuk list. Berikut pada Gambar 4.34 hasil prediksi dalam bentuk JSON-serializable.



**Gambar 4. 34** Hasil Prediksi dalam bentuk JSON

#### 4.5.3 Pengujian Akurasi Model LSTM

Pengujian akurasi model LSTM dilakukan dengan membandingkan data nilai kelembaban tanah hasil prediksi dengan data nilai kelembaban aktual. Pengujian akurasi dapat dilakukan dengan menghitung nilai rata-rata perbedaan absolut antara nilai hasil prediksi dengan nilai aktual. Metode ini dikenal sebagai Mean Absolute Percentage Error (MAPE). Menurut Nabillah & Ranggadara (2020), MAPPE adalah selisih absolut rata-rata antara nilai prediksi dan nilai aktual, yang dinyatakan dalam persentase dari nilai realisasi. Saat mengevaluasi hasil prakiraan dapat menggunakan kesalahan persentase absolut rata-rata (MAPE) untuk melihat seberapa akurat angka prediksi dan angka sebenarnya.

Nilai MAPE dapat dihitung menggunakan persamaan 4.1 (Nabillah & Ranggadara 2020) berikut :

$$\frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - B_i}{B_i} \right| 100 \quad (4.1)$$

Keterangan :

$A_i$  = Nilai prediksi

$B_i$  = Nilai aktual

$n$  = jumlah observasi

Pengujian ini dilakukan pada model LSTM dengan parameter terbaik yaitu model dengan rasio pemisahan 70:30 dengan LSTM layer units 416 dan Learning Rate 0.001. Data nilai aktual kelembaban tanah yang digunakan adalah data pada pukul 16.00 hingga pukul 23.00. Terdapat tiga sampel data nilai aktual, yaitu Sampel Hari 1, Sampel Hari 2, dan Sampel Hari 3. Data-data sampel ditampilkan pada Tabel 4.10, 4.11 dan 4.12.

**Tabel 4. 10** Data Sampel Hari 1

No.	Waktu	Nilai Kelembaban Tanah
1.	16.00 06-05-2024	36.11
2.	17.00 06-05-2024	35.84
3.	18.00 06-05-2024	35.28
4.	19.00 06-05-2024	34.89
5.	20.00 06-05-2024	34.56
6.	21.00 06-05-2024	34.17

7.	22.00 06-05-2024	33.90
8.	23.00 06-05-2024	33.76

**Tabel 4. 11** Data Sampel Hari 2

No.	Waktu	Nilai Kelembaban Tanah
1.	16.00 06-06-2024	52.24
2.	17.00 06-06-2024	51.69
3.	18.00 06-06-2024	51.31
4.	19.00 06-06-2024	51.02
5.	20.00 06-06-2024	50.81
6.	21.00 06-06-2024	50.49
7.	22.00 06-06-2024	50.26
8.	23.00 06-06-2024	50.02

**Tabel 4. 12** Data Sampel Hari 3

No.	Waktu	Nilai Kelembaban Tanah
1.	16.00 06-07-2024	48.22
2.	17.00 06-07-2024	47.88
3.	18.00 06-07-2024	47.46
4.	19.00 06-07-2024	47.12
5.	20.00 06-07-2024	46.93
6.	21.00 06-07-2024	46.70
7.	22.00 06-07-2024	46.54
8	23.00 06-07-2024	46.34

#### 4.5.3.1 Pengujian Akurasi pada LSTM

Pengujian Akurasi dilakukan dengan membandingkan hasil prediksi dengan nilai aktual Data Sampel Hari 1. Berikut hasil pengujian akurasi pada Tabel 4.13.

**Tabel 4. 13** Akurasi Model LSTM dengan Data Sampel Hari 1

No.	Waktu	Prakiraan Suhu (°C)	Nilai Kelembaban Tanah (Prediksi)	Nilai Kelembaban Tanah (Aktual)	Akurasi (MAPPE)
-----	-------	------------------------	---	---------------------------------------	--------------------

1.	16.00	29.5	37.37	36.11	3.49%
2.	17.00	29.1	37.22	35.84	3.85%
3.	18.00	28.5	36.87	35.28	4.51%
4.	19.00	26.6	36.92	34.89	5.82%
5.	20.00	25.4	36.56	34.56	5.79%
6.	21.00	25.1	36.28	34.17	6.18%
7.	22.00	24.7	35.87	33.90	5.81%
8.	23.00	24.5	35.65	33.76	5.60%
Rata-rata MAPE					<b>5.13%</b>

Berikutnya adalah perbandingan hasil prediksi dengan Data Sampel Hari 2 dipaparkan pada Tabel 4.14.

**Tabel 4. 14** Akurasi Model LSTM dengan Data Sampel Hari 2

No.	Waktu	Prakiraan Suhu (°C)	Nilai Kelembaban Tanah (Prediksi)	Nilai Kelembaban Tanah (Aktual)	Akurasi (MAPPE)
1.	16.00	27.42	54.52	52.24	4.36%
2.	17.00	26.23	54.45	51.69	5.34%
3.	18.00	26.26	53.86	51.31	4.97%
4.	19.00	25.63	53.92	51.02	5.68%
5.	20.00	24.32	53.83	50.81	5.94%
6.	21.00	23.65	53.56	50.49	6.08%
7.	22.00	23.23	53.24	50.26	5.93%
8.	23.00	23.20	53.36	50.02	6.68%
Rata-rata MAPE					<b>5.62%</b>

Berikutnya perbandingan hasil prediksi dengan Data Sampel Hari 3 dipaparkan pada Tabel 4.15.

**Tabel 4. 15** Akurasi Model LSTM dengan Data Sampel Hari 3

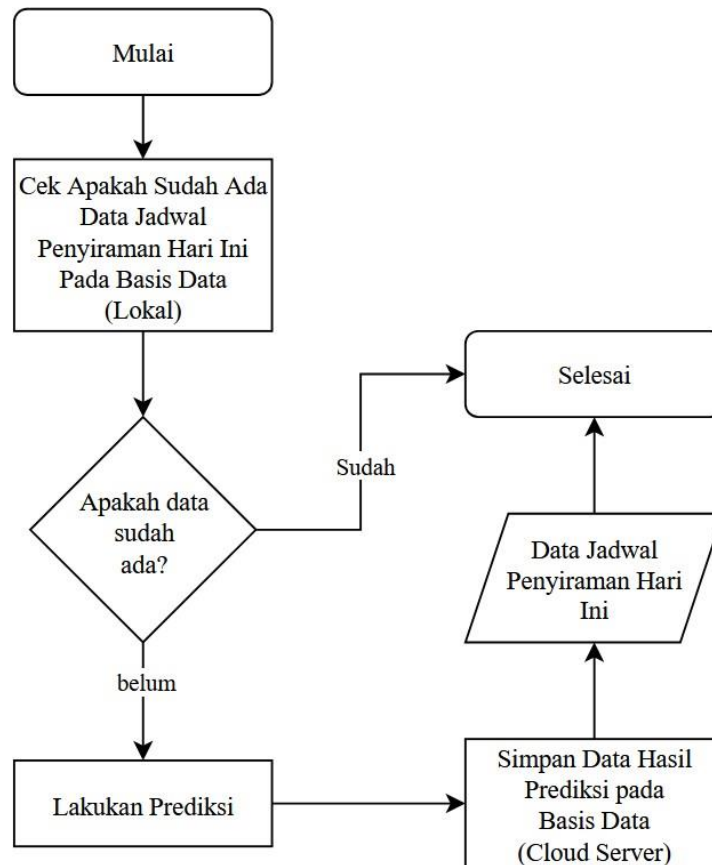
No.	Waktu	Prakiraan Suhu (°C)	Nilai Kelembaban Tanah (Prediksi)	Nilai Kelembaban Tanah (Aktual)	Akurasi (MAPPE)
-----	-------	------------------------	---	---------------------------------------	--------------------

1.	16.00	30.23	50.42	48.22	4.56%
2.	17.00	28.77	50.23	47.88	4.91%
3.	18.00	28.56	50.11	47.46	5.58%
4.	19.00	27.32	49.76	47.12	5.60%
5.	20.00	26.74	49.89	46.93	6.31%
6.	21.00	26.66	49.66	46.70	6.34%
7.	22.00	26.12	49.23	46.54	5.78%
8.	23.00	25.44	48.98	46.34	5.70%
Rata-rata MAPE					<b>5.60%</b>

Dari percobaan yang dilakukan, rata-rata MAPE yang di dapat dengan membandingkan nilai hasil prediksi dengan nilai sampel adalah 4.90%, 24.69%, 5.19%. Ini berarti model LSTM dengan rasio pemisahan data *training* dan *testing* 70:30 dengan LSTM *unit* 416 pada percobaan ini memiliki rata-rata nilai prediksi yang menyimpang atau eror berkisar 5,13% ~ 5,60%.

#### 4.5.4 Pengujian Sistem Penjadwalan Penyiraman

Pengujian sistem penjadwalan penyiraman ini dilakukan pada perangkat keras berupa Laptop penulis. Namun implementasi sistem penjadwalan penyiraman harusnya dilakukan pada *cloud server*. Alur Implementasi sistem penjadwalan penyiraman pada perangkat keras Laptop ini dapat dilihat pada Gambar 4.35.



**Gambar 4. 35** Flowchart Pengujian Penjadwalan Penyiraman

#### 4.5.4.1 Pengecekan Ketersediaan Data Jadwal Penyiraman

Langkah pertama yaitu melakukan pengecekan ketersediaan data jadwal penyiraman pada basis data. Pada percobaan ini pengecekan ketersediaan data dilakukan pada basis data lokal. Pengecekan dilakukan dengan memanfaatkan API endpoint `/api/get-jadwal-nyiram`. API endpoint ini mengirim *request* ke pada aplikasi berbasis web yang berjalan pada lokal yang terhubung dengan basis data lokal untuk mengecek apakah data jadwal penyiraman hari ini tersedia atau tidak. Jika tersedia, maka proses pengecekan data selesai dilakukan. Jika tidak tersedia, maka aplikasi prediksi Flask akan melakukan prediksi. Hasil prediksi kemudian dikirimkan melalui MQTT ke basis data pada *cloud server* untuk disimpan.

*// Fungsi untuk mendapatkan status apakah data jadwal penyiraman hari ini sudah tersedia atau belum.*

```
def get_adakah_jadwal_nyiram():
    try:
        response = requests.post('http://localhost:8080/api/get-jadwal-nyiram')
```

```

    if response.status_code == 200:
        adakah_jadwal = response.json().get('status')
        return adakah_jadwal
    else:
print("Gagal mendapatkan status ketersediaan data. Status code:",
response.status_code)
        return None
except requests.RequestException as e:
    print("Error saat melakukan request status ketersediaan data:", e)
    return None

```

Dari penggalan source code di atas dapat disimak bahwa fungsi melakukan *request* post ke endpoint *api/get-jadwal-nyirami* untuk mendapatkan status tersedia atau tidaknya data pada tabel *jadwal\_nyiram* pada basis data.

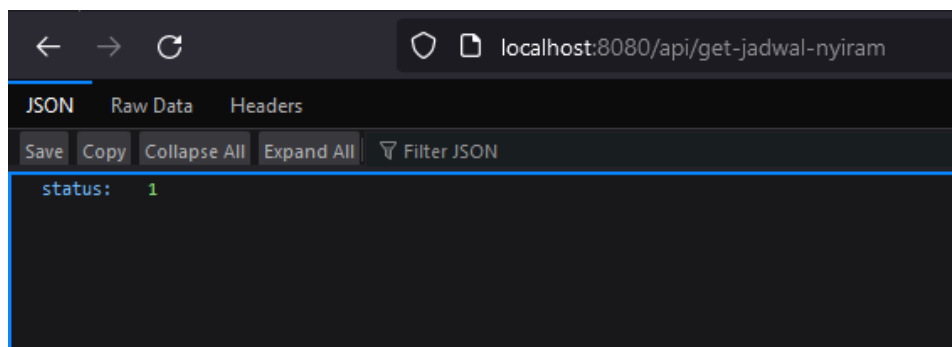
// Cek ketersediaan data.

```

if ($is_today_data_available) {
    $data['status'] = 1; // Today's data is available
} else {
    $data['status'] = 0; // Today's data is not available
}

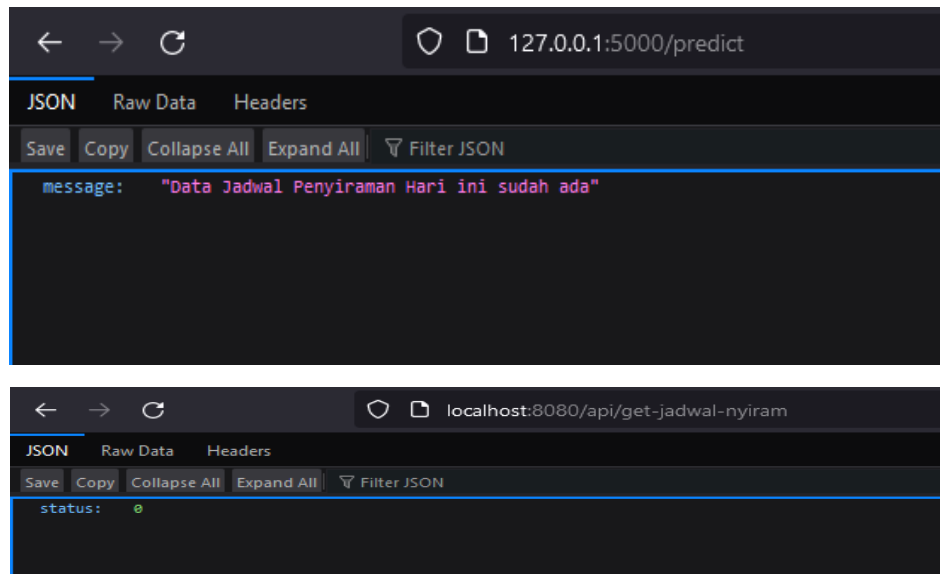
```

Dari penggalan source code, dapat disimak jika data tersedia, akan mengembalikan status = 1. Jika data tidak tersedia akan mengembalikan status = 0. Contoh percobaan dengan data telah tersedia (status = 1) dapat dilihat pada Gambar 4.36.



**Gambar 4. 36** Pesan dari Endpoint *gateway*

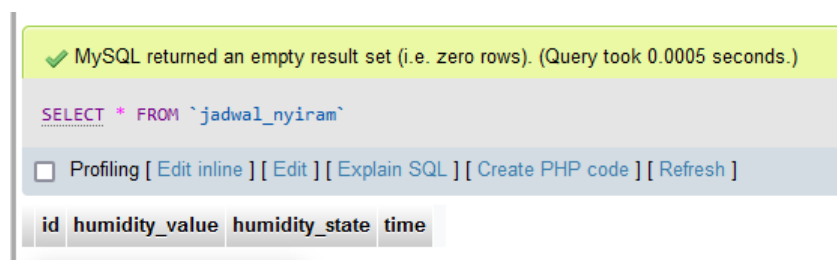
Jika aplikasi prediksi Flask dijalankan, maka aplikasi akan mengembalikan pesan "Data Jadwal Penyiraman Hari ini sudah ada".



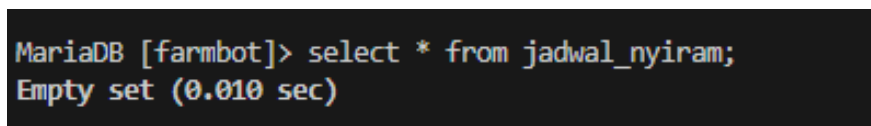
**Gambar 4. 37** Pesan Pada Aplikasi Flask

Adapun yang dimaksud pada Gambar 4.37 yaitu, pesan pada aplikasi prediksi flask jika data jadwal penyiraman hari ini sudah ada dan tidak ada data jadwal penyiraman. Percobaan selanjutnya adalah jika data jadwal penyiraman hari ini belum ada. Jika data belum ada, maka akan dilakukan prediksi nilai tingkat kelembaban tanah hari ini, lalu hasilnya disimpan ke basis data *cloud server*.

Pada percobaan ini, akan ditampilkan bahwa tidak ada data jadwal penyiraman pada hari ini pada basis data lokal dan *cloud server*. Tampilan tidak ada data pada basis data lokal akan ditampilkan pada Gambar 4.38 dan pada *cloud server* ditampilkan pada 4.39.



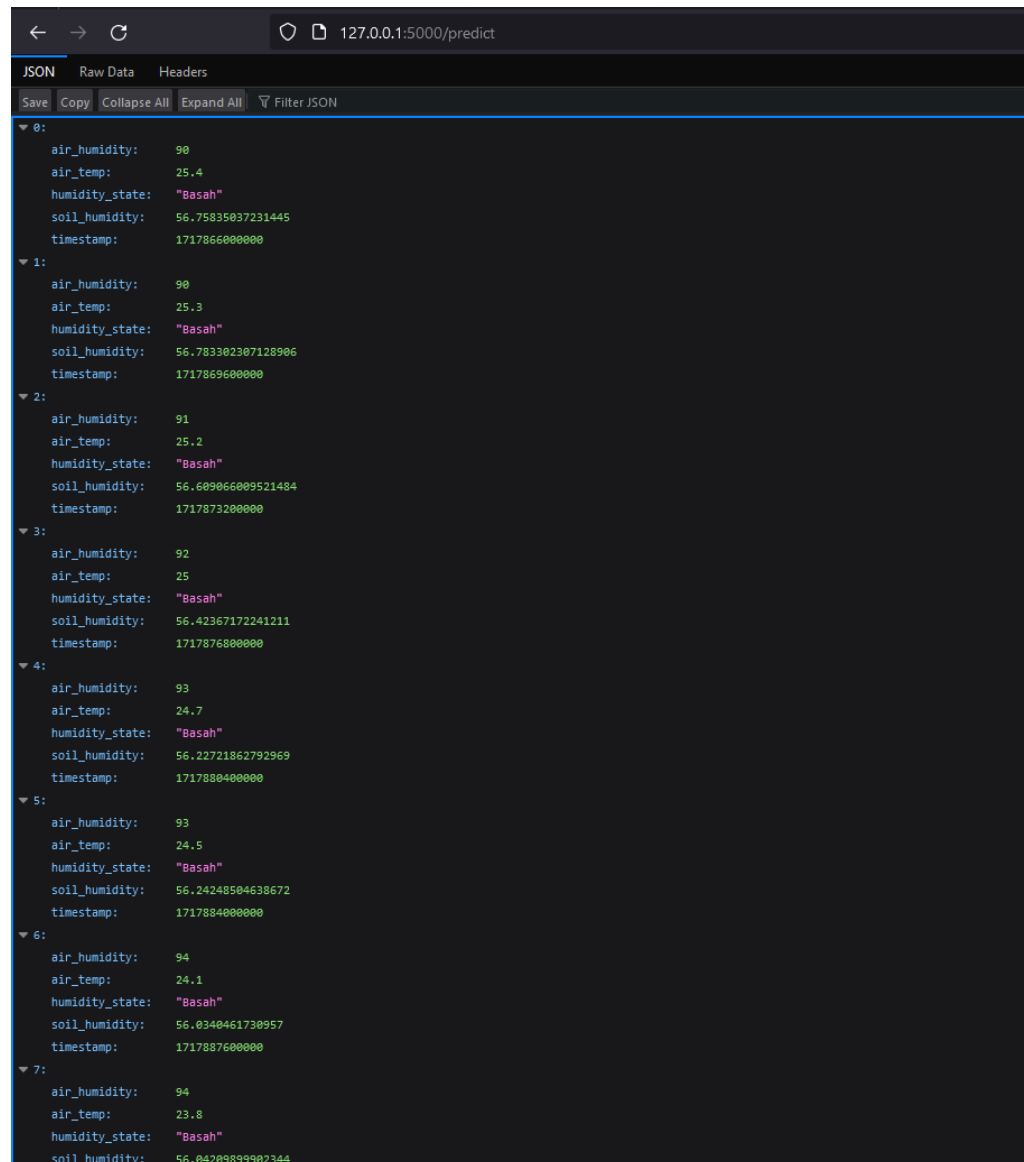
**Gambar 4. 38** Tampilan Tidak Ada Data pada Basis Data Lokal



**Gambar 4. 39** Tampilan Tidak Ada Data pada basis

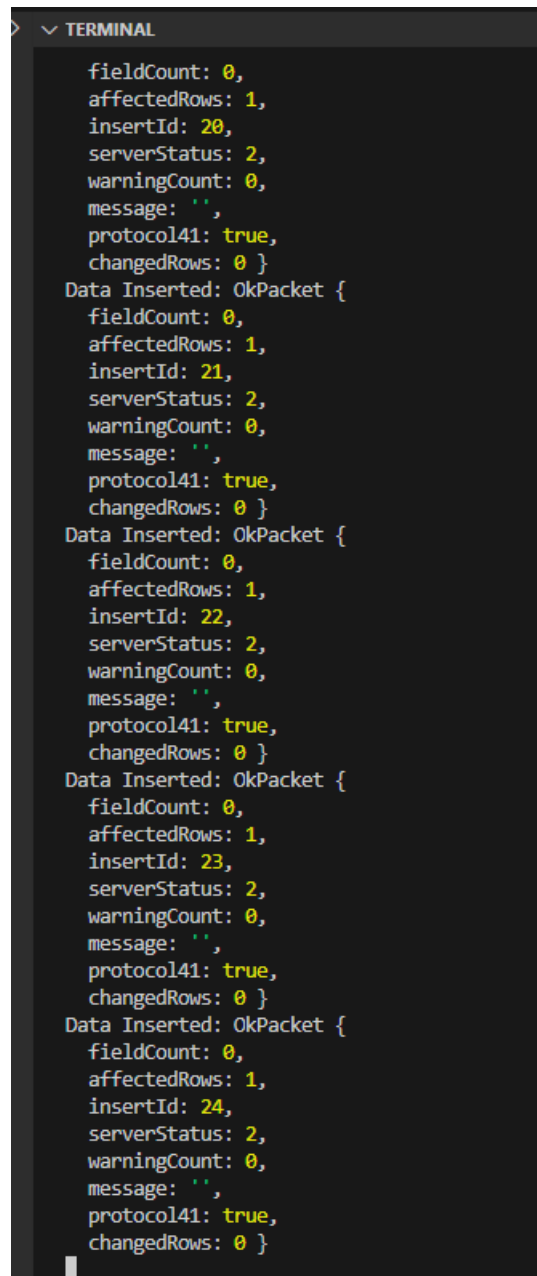
Kemudian akan dilakukan proses prediksi nilai tingkat kelembaban tanah dengan mengakses endpoint */predict*. Hasil prediksi dapat disimak pada Gambar 4.38.





**Gambar 4. 40** Tampilan Hasil Prediksi

Data kemudian dikirim melalui MQTT ke *cloud server*. Log penerimaan data pada aplikasi Javascript pada server dapat dilihat pada 4.40.

A screenshot of a terminal window with a dark background. The title bar at the top says 'TERMINAL'. The terminal displays a series of JSON-like log entries. Each entry starts with a set of key-value pairs, followed by 'Data Inserted: OkPacket {', another set of key-value pairs, and ends with '}'. The 'insertId' values are 20, 21, 22, 23, and 24, appearing in each successive entry. The 'affectedRows' value is consistently 1. The 'serverStatus' is 2, 'warningCount' is 0, 'message' is an empty string, and 'protocol41' is true. The 'changedRows' value is 0 in all entries.

```
> v TERMINAL

  fieldCount: 0,
  affectedRows: 1,
  insertId: 20,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
Data Inserted: OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 21,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
Data Inserted: OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 22,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
Data Inserted: OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 23,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
Data Inserted: OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 24,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
```

**Gambar 4. 41** Log Penerimaan Data

Terdapat insertId: 24 pada Gambar 4.41 yang berarti sebanyak 24 data hasil prediksi berhasil disimpan. Data yang berhasil disimpan ditampilkan pada Gambar 4.42.

```
MariaDB [farmbot]> select * from jadwal_nyiram;
```

id	humidity_value	humidity_state	time
1	56.7584	Basah	1717866000000
2	56.6091	Basah	1717873200000
3	56.7833	Basah	1717869600000
4	56.4237	Basah	1717876800000
5	56.2272	Basah	1717880400000
6	56.2425	Basah	1717884000000
7	56.034	Basah	1717887600000
8	56.0421	Basah	1717891200000
9	57.4528	Basah	1717898400000
10	56.3045	Basah	1717894800000
11	58.9145	Basah	1717902000000
12	61.0512	Basah	1717909200000
13	61.2084	Basah	1717912800000
14	61.3457	Basah	1717920000000
15	61.3107	Basah	1717916400000
16	60.2154	Basah	1717905600000
17	59.8457	Basah	1717923600000
18	58.2677	Basah	1717927200000
19	57.6453	Basah	1717930800000
20	57.1383	Basah	1717934400000
21	58.6709	Basah	1717938000000
22	56.6401	Basah	1717941600000
23	56.4062	Basah	1717945200000
24	56.4075	Basah	1717948800000

24 rows in set (0.002 sec)

**Gambar 4. 42** Data Jadwal Penyiraman Berhasil Disimpan

Akhirnya, sistem robot pertanian akan melakukan pengecekan apakah pada hari ini terdapat status kelembaban (*humidity\_state*) pada tabel *jadwal\_nyiram*. Jika ada, maka pada akan diambil waktu (*time*) dari data tersebut sebagai acuan jadwal penyiraman.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Kesimpulan dari hasil penelitian dapat diuraikan sebagai berikut :

1. Penelitian berhasil dilakukan dengan mengimplementasikan penjadwalan otomatis pada sistem robot pertanian dengan menyediakan data jadwal penyiraman sebagai acuan kapan robot akan melakukan penyiraman.
2. Sistem penginderaan terintegrasi berhasil diimplementasikan dengan membuat rangkaian sensor tanah dan udara dan menyimpan data hasil penginderaan ke dalam basis data pada *cloud server*. Data hasil penginderaan kemudian berhasil ditampilkan pada aplikasi berbasis web sehingga dapat dipantau dari mana saja kapan saja.
3. Model prediksi dengan memanfaatkan algoritma *Long Short-Term Memory* (LSTM) berhasil dibuat dan diimplementasikan sebagai model prediksi nilai tingkat kelembaban tanah dengan rata-rata nilai prediksi yang menyimpang (deviasi) berkisar 5,13% ~ 5,60%. Model LSTM ini menggunakan optimisasi *hyperparameter* yaitu Layer Units 416, Learning Rate 0,001 dengan *validation loss* sebesar 0,0048 dan *training loss* sebesar 0,0071. *Hyperparameter* ini didapat dari percobaan pencarian *hyperparameter* terbaik menggunakan Keras Tuner yang membutuhkan parameter *Model Builder Function* sebagai masukannya. Rasio pemisahan data *training* dan *testing* yang digunakan pada model ini adalah 70:30, yang berarti 70% data *training* dan 30% data *testing*.

#### **5.2 Saran**

Saran untuk penelitian berikutnya adalah sebagai berikut :

1. Pengumpulan data hendaklah dilakukan dalam waktu yang lama, seperti satu tahun penuh atau lebih sehingga data yang didapat dan dikumpulkan mencakup perbedaan kondisi lingkungan dari berbagai musim. Alasannya

karena ada kalanya kondisi lingkungan sangat panas dan kering, dan adakalanya kondisi lingkungan dingin dan lembab yang dipengaruhi oleh musim.

2. Pemilihan algoritma selain *Long Short-Term Memory* dapat dijadikan pertimbangan. Terdapat berbagai macam algoritma prediksi regresi seperti *Random Forest*, *ARIMA*, *Prophet*, dan lain sebagainya yang mungkin dapat memprediksi nilai tingkat kelembaban tanah dengan lebih tepat dan konsisten.
3. Implementasi sistem aplikasi pemantauan berbasis web maupun aplikasi prediksi Flask baiknya diimplementasikan sepenuhnya pada *cloud*. Sehingga sistem dapat bekerja tanpa hambatan teknologi.
4. Pemilihan model teknologi pertanian selain tipe *Green House* (rumah kaca) seperti hidroponik dapat dipertimbangkan agar penelitian lebih bervariasi dan dapat diimplementasikan dalam kehidupan pertanian sehingga menambah opsi masyarakat khususnya petani dalam menentukan model pertanian.
5. Pemilihan jenis mikrokontroler dan sensor lainnya dapat dipertimbangkan. Terdapat mikrokontroler yang mempunyai fungsi dan modul lebih dalam satu perangkat dan sensor yang mungkin lebih akurat dan dapat berfungsi lebih konsisten dibandingkan dengan yang digunakan pada penelitian ini.

## DAFTAR PUSTAKA

- Aldi, M. W. P., Jondri & Aditsania, A. 2018. Analisis dan Implementasi Long Short-Term Memory Neural Network untuk Prediksi Harga Bitcoin. *e-Proceeding of Engineering*, pp. 3548-3555.
- Alkausar, M. F., Dewi, T. & Oktarina, Y. 2022. Impelementasi Image Processing pada Robot Pertanian. *Journal of Applied Smart Electrical Network and Systems (JASENS)* 3(2): 37-42.
- Bechar, A. & Vigneault, C. 2016. Agricultural robots for field operations: Concepts and components. *Biosystems Engineering* 149: 94-111.
- Binkowski, M., Marti, G. & Donnat, P. 2018. Autoregressive Convolutional Neural Networks for Asynchronous Time Series. *Proceeding of the 35<sup>th</sup> International Conference on Machine*, pp.580-589.
- Bor, M. & Roedig, U. 2017. LoRa Transmission Parameter Selection. *2017 13th International Conference on Distributed Computing in Sensor Systems* 17: 2325-2944.
- Chai, T. & Draxler, R. R. 2014. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?. *Journal Geoscientific Model Development* 7: 1525-1534.
- Chicco, D., Warrens, M.J. & Jurman, G. 2021. The Coefficient of Determination R-squared is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Computer Science* 7: 1–24.
- Conesa-Muñoz, J., Gonzalez-de-Soto, M., Gonzalez-de-Santos, P., & Ribeiro, A. 2015. Distributed Multi-Level Supervision to Effectively Monitor the Operations of a Fleet of Autonomous Vehicles in Agricultural Tasks. *Centre for Automation and Robotics* 15(3): 5402-5428.
- Graves, A. 2012. Long Short-Term Memory. *Proceedings of Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 37-45.
- Grinberg, M. 2018. *Flask Web Development*. Allyson MacDonald: United State of America.
- Griva, A. I., Boirsianis, A. D., Wan, S., Sarigiannidis, P., Psannis, K. E., Karagiannidis, G. & Goudos, S. K. 2023. LoRa-Based IoT Network Assessment in Rural and Urban Scenarios. *MDPI Sensors* 23(3): 1-14.

- Iman, F. N. & Wulandari, D. 2023. Prediksi Saham Menggunakan Metode Long Short-Term Memory. *LOGIC: Jurnal Ilmu Komputer dan Pendidikan* 1(3): 601-616.
- Kornienko, D. V., Mishina, S. V., Shcherbatykh, S. V. & Melnikov, M. O. 2021. Principles of Securing RESTful API Web Services Developed with Python Framework. *Journal of Physics: Conference Series* 2094.
- Misra, P. & Yadav, A. S. 2019. Impact of Preprocessing Methods on Healthcare Prediction. *Proceedings of 2<sup>nd</sup> International Conference on Advanced Computing and Software Engineering (ICACSE-2019)*, pp. 144-150.
- Nabillah, I. & Ranggadara, I. 2020. Mean Absolute Error untuk Evaluasi Hasil Prediksi Komuditas Laut. *Journal of Information System* 5(2): 250-255.
- Rahman, A. F. S., Nugraha, M. S., Nugraha, A. 2023. Perancangan dan Monitoring Sistem Pertanian Hidroponik Berbasis Wireless Sensor Network (WSN). *Jurnal Teknik Elektro UNIBA* 7(2): 345-349.
- Saputra, R., Siregar, M. F. & Hidayat, J. 2023. Rancang Bangun Sistem Monitoring Smart Green House Di Laboratorium Pertanian Universitas Tjut Nyak Dhien. *Jurnal Persegi Bulat* 2(1): 16-22.
- Susanto, A. R., Bhawiyuga, A. & Amron, K. 2019. Implementasi Sistem Gateway Discovery pada Wireless Sensor Network (WSN) Berbasis Modul Komunikasi LoRa. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 3(2): 2138-2145.
- Wibowo, A. & Zain, M. M. 2021. Pemanfaatan ReactJS dan Protokol MQTT untuk Visualisasi Sinyal Lampu dan Notifikasi secara Waktu Nyata pada Sistem Pemonitor APILL di Kota Pekanbaru. *Jurnal Politeknik Caltex Riau* 7(2): 314-328.
- Zulkifli, N., Amir, A. L., Shukri, A. H. S. A., Salleh, N. B. M., Suhaimi, E. N. E. M. F. & Osman, M. A. 2020. Examination Management System Using Yii2 Framework. *Melaka International College of Science and Technology* 52: 322-328.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS SUMATERA UTARA  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007  
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI  
NOMOR : 2712/UN5.2.14.D/SK/SPB/2024  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 10 Juli 2024 perihal permohonan ujian skripsi:  
Nama : Ali Hidayat  
NIM : 171402019  
Program Studi : Sarjana (S-1) Teknologi Informasi  
Judul Skripsi : Sistem Penginderaan Terintegrasi Untuk Pemantauan Kondisi Lahan Tanam dan Otomatisasi Penjadwalan Penyiraman Tanaman Menggunakan Algoritma Long Short-Term Memory
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.  
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.  
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.  
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026
- MEMUTUSKAN
- Menetapkan :  
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:  
Ketua : Ivan Jaya S.Si., M.Kom.  
NIP: 198407072015041001  
Sekretaris : Fahrurrozi Lubis B.IT., M.Sc.IT  
NIP: 198610122018052001  
Anggota Penguji : Baihaqi Siregar S.Si., MT.  
NIP: 197901082012121002  
Anggota Penguji : Ainul Hizriadi S.Kom, M.Sc  
NIP: 198510272017061001  
Moderator : -  
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

- Tembusan :
- 1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
  - 2. Yang bersangkutan
  - 3. Arsip

Medan  
Ditandatangani secara elektronik oleh:  
Dekan



Maya Silvi Lydia  
NIP 197401272002122001