BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori merupakan kerangka konseptual yang menopang seluruh rancangan dan implementasi penelitian ini. Bagian ini menyajikan tinjauan pustaka dan konsep-konsep dasar yang relevan dengan topik penelitian, mulai dari pengenalan objek studi hingga teknologi yang digunakan, guna memastikan dasar ilmiah yang kuat untuk pengembangan system.

2.1.1 Kualitas Udara

Kualitas udara merupakan tingkat baik atau buruknya campuran berbagai gas yang tidak berwarna dan tidak berbau, yang mana memenuhi seluruh ruang di atas bumi, dan digunakan untuk makhluk hidup bernapas. Dengan kata lain, kualitas udara adalah tingkatan baik atau buruknya udara.

Pada tahun 2020, KLHK telah mengeluarkan Peraturan Menteri Lingkungan Hidup dan Kehutanan nomor 14 tahun 2020 tentang Indeks Standar Pencemar Udara. Pada peraturan ini, tercantum bahwa perhitungan ISPU dilakukan pada 7 (tujuh) parameter yakni PM10, PM2.5, NO2, SO2, CO, O3, dan HC. Terdapat penambahan 2 (dua) parameter yakni HC dan PM2.5 dari peraturan sebelumnya. Penambahan parameter tersebut didasari pada besarnya resiko HC dan PM2.5 terhadap kesehatan manusia. Berikut adalah kategori dari Indeks Standar Pencemar Udara (ISPU):

Tabel 2. 1 Kategori Indeks Standar Pencemar Udara (ISPU)

Rentang	Kategori	Penjelasan
0-50	Baik	Tingkat mutu udara yang sangat baik, tidak memberikan efek negatif terhadap manusia, hewan dan tumbuhan.
51-100	Sedang	Tingkat mutu udara masih dapat diterima pada kesehatan manusia, hewan dan tumbuhan.
101-200	Tidak Sehat	Tingkat mutu udara yang bersifat merugikan pada manusia, hewan dan tumbuhan.

201-300	Sangat Tidak Sehat	Tingkat mutu udara yang dapat meningkatkan resiko kesehatan pada sejumlah segmen populasi yang terpapar.
301+	Berbahaya	Tingkat mutu udara yang dapat merugikan kesehatan serius pada populasi dan perlu penanganan cepat.

Tabel konversi nilai konsentrasi parameter ISPU dan cara perhitungan sebagai berikut:

Tabel 2. 2 Konversi Nilai Konsentrasi

ISPU	24 Jam PM10 (µg/m³)	24 Jam PM2.5 (µg/m³)	24 Jam SO2 (µg/m³)	24 Jam CO (µg/m³)	24 Jam O3 (µg/m³)	24 Jam NO2 (µg/m³)	24 Jam HC (µg/m³)
0-50	50	15,5	52	4000	120	80	45
51-100	150	55,4	180	8000	235	200	100
101-200	350	150,4	400	15000	400	1130	215
201-300	420	250,4	800	30000	800	2260	432
>300	500	500	1200	45000	1000	3000	648

Perhitungan ISPU dilakukan berdasarkan nilai ISPU batas atas, ISPU batas bawah, ambien batas atas, ambien batas bawah, dan konsentrasi ambien hasil pengukuran. Persamaan matematika perhitungan ISPU sebagai berikut:

$$I = \frac{Ia-Ib}{Xa-Xb}\ (Xx - Xb) + Ib \qquad (1)$$

Dimana,

I = ISPU terhitung

Ia = ISPU batas atas

Ib = ISPU batas bawah

Xa = Konsentrasi ambien batas atas (µg/m3)

Xb = Konsentrasi ambien batas bawah (µg/m3)

Xx = Konsentrasi ambien nyata hasil pengukuran (µg/m3)

2.1.2 Particulate Matter (PM)

Particulate Matter (PM) adalah partikel yang biasanya terbentuk di atmosfer akibat dari reaksi kimia beberapa polutan yang berbeda. Partikel dengan ukuran 10 micrometer (µm) atau lebih kecil disebut dengan PM10 dan partikel dengan diameter 2.5 micrometer (µm) atau lebih kecil disebut PM2.5 (Manisalidis dkk., 2020). Sumber dari partikel ini umumnya dari emisi kendaraan dan industri, asap rokok, dan asap dari kebakaran hutan (Aryanta dan Maharani, 2023). Polutan ini dapat menyebabkan gangguan pernapasan, yaitu Infeksi Saluran Pernapasan Akut (ISPA), kanker paru-paru, penyakit kardiovaskular, dan penyakit paru obstruktif kronis (Kamil, 2024).

2.1.3 Nitrogen Dioksida (NO2)

Nitrogen dioksida adalah sumber utama dari aerosol nitrat yang membentuk PM2.5 dan ozon (O3) jika ada sinar ultraviolet dari matahari (Aryanta dan Maharani, 2023). Polutan udara ini terbentuk dari pembakaran bahan bakar fosil pada suhu tinggi, yaitu batu bara, minyak mentah, bensin, atau solar (Kamil, 2024). Dampak dari polutan ini adalah penyakit pernapasan, batuk-batuk, sesak nafas, bronkospasme bahkan edema paru. Paparan jangka panjang dapat menyebabkan kanker paru-paru kronis dan merusak indra penciuman (Manisalidis dkk., 2020).

2.1.4 Sensor Nova PM SDS011

Sensor Nova PM SDS011 adalah perangkat yang digunakan untuk mengukur konsentrasi PM2.5 dan PM10 di udara. Sensor ini bekerja dengan menggunakan prinsip pemancaran cahaya dan deteksi pantulan cahaya untuk menghitung jumlah partikel di udara. Kelebihan dari sensor ini adalah hasil yang akurat, konsisten, dan memiliki respon waktu yang cepat dalam mendeteksi perubahan konsentrasi partikel (Kusumaningrum dkk., 2023).

Gambar 2. 1 Sensor Nova PM SDS011

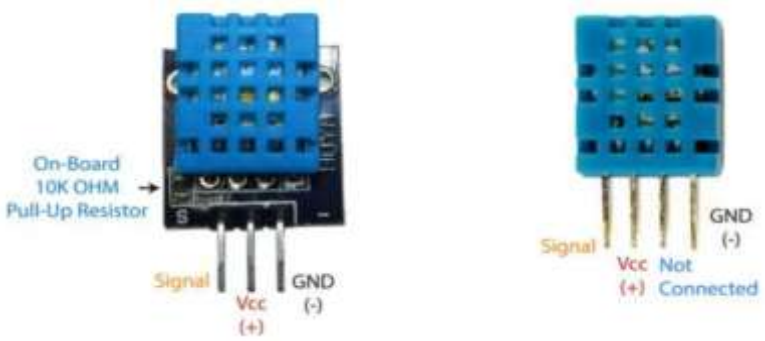
2.1.5 Sensor MQ-135

Sensor MQ-135 adalah sensor yang dapat mendeteksi gas amonia (NH3), benzena (C6H6), karbon dioksida (CO2), nitrogen dioksida (NO2), gas berbahaya lainnya dan asap. Sensor ini memiliki pin output digital dan analog. Ketika tingkat gas melampaui batas ambang di udara, pin digital menjadi *HIGH*, untuk pin keluaran analog mengeluarkan tegangan analog yang dapat digunakan untuk memperkirakan tingkat gas di udara. Sensor ini memiliki cakupan deteksi yang luas, respon cepat, sensitifitas tinggi, stabil, dan tahan lama (Rombang dkk., 2022).

Gambar 2. 2 Sensor MQ-135

2.1.6 Sensor DHT22

DHT22 adalah sensor suhu dan kelembapan dengan *Output* sudah berupa sinyal digital dengan konversi dan perhitungan dilakukan oleh MCU 8-bit. DHT22 lebih akurat dan presisi dalam hasil pengukuran, *range* pengukuran yang lebih suhu dan kelembapan yang lebih lebar dan mampu mentransmisikan sinyal *output* melewati kabel yang panjang (hingga 20m) sehingga cocok ditempatkan dimana saja (Siswanto dkk., 2019)



Gambar 2. 3 Sensor DHT22

2.1.7 Message Queue Telemetry Transport (MQTT)

Menurut Dinculeană & Cheng (2019), *Message Queue Telemetry Transport* (MQTT) adalah protokol yang digunakan untuk komunikasi dalam lingkungan IoT yang beroperasi di atas *Transport Control Protocol* (TCP). Protokol ini dikembangkan oleh IBM sebagai metode komunikasi machine-to-machine yang ringan.

MQTT telah distandarisasi oleh ISO/IEC 20922 dan kemudian diterima sebagai bagian dari OASIS. Pada dasarnya, MQTT adalah protokol pesan yang menggunakan model komunikasi *publish-subscribe*, di mana klien tidak memerlukan pembaruan secara langsung. Hal ini menyebabkan pengurangan penggunaan sumber daya, sehingga model ini optimal digunakan dalam lingkungan dengan bandwidth rendah.

Selain itu, protokol ini berfungsi dalam sistem server-klien, di mana server yang disebut *broker* mengirimkan pembaruan ke klien MQTT. Klien tidak akan mengirim pesan secara langsung satu sama lain, melainkan bergantung pada *broker* untuk melakukan hal ini. Setiap pesan MQTT memiliki topik, yang diorganisir dalam struktur pohon, di mana klien dapat berlangganan (*subscribe*) atau menerbitkan (*publish*) pesan. *Broker* menerima pesan yang diterbitkan oleh klien yang berisi nilai atau perintah tertentu, kemudian meneruskan informasi tersebut ke setiap klien yang telah berlangganan ke topik yang sama.

Protokol MQTT dirancang untuk komunikasi asinkron, di mana proses *subscribe* atau *publish* dapat terjadi secara paralel. Protokol ini juga mampu menyediakan transfer data yang andal dengan memilih antara tiga jenis mekanisme keandalan, yang disebut *Quality of Service* (QoS).

Jika dibandingkan dengan protokol lain yaitu HTTP, protokol MQTT memiliki jejak data yang jauh lebih kecil. Hal ini menjadikan MQTT lebih cocok untuk lingkungan dengan keterbatasan sumber daya.

2.1.8 MongoDB

MongoDB adalah *database* NoSQL yang berorientasi dokumen yang digunakan untuk penyimpanan sebuah data dalam volume tinggi. MongoDB termasuk ke dalam DBMS (*Database Management System*) yang menggunakan skema dinamis, yaitu dapat membuat record tanpa harus menentukan strukturnya. MongoDB juga dapat mengubah struktur catatan atau dokumen dengan menambahkan *field* baru atau menghapus *field* yang sudah ada (Naufal dkk., 2022).

2.1.9 Machine Learning

Menurut Sidik dan Ansawarman (2022), *machine learning* merupakan salah satu bidang dari ilmu komputer yang merancang sebuah algoritma agar komputer dapat belajar melalui data. *Machine learning* menggunakan data masa lalu yang digunakan untuk pembelajaran model sehingga mendapatkan kinerja yang optimal dalam menggali informasi dari suatu kumpulan data.

Secara garis besar, *machine learning* dibagi menjadi 3, yaitu:

1) *Supervised Learning*

   Supervised Learning adalah algoritma *machine learning* yang proses pembelajarannya di bawah pengawasan. Algoritma ini dilatih untuk mengenali pola antara input data dan label output. Klasifikasi dan regresi termasuk dalam *supervised learning*.

2) *Unsupervised Learning*

   Unsupervised Learning adalah algoritma *machine learning* yang proses pembelajarannya tanpa pengawasan. Algoritma ini dirancang untuk dapat "belajar mandiri" dalam mengumpulkan informasi, termasuk mengenali data yang tidak berlabel. *Clustering* dan *dimensionality reduction* termasuk dalam *unsupervised learning*.

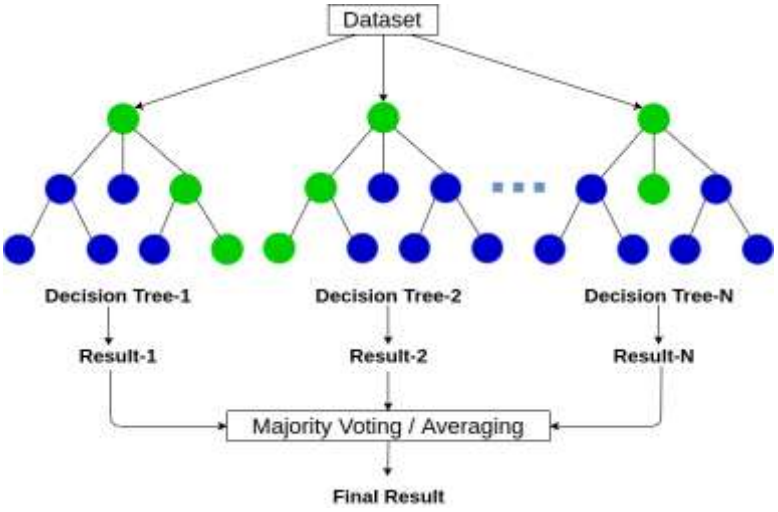
3) *Reinforcement Learning*

   Reinforcement Learning adalah algoritma *machine learning* yang dapat membuat *agent software* mesin bekerja secara otomatis untuk menentukan

perilaku ideal sehingga dapat memaksimalkan kinerja algoritma.

2.1.10 Random Forest Regression (RFR)

Random Forest Regression adalah algoritma *machine learning ensemble learning* yang digunakan untuk memprediksi nilai berdasarkan satu atau lebih variabel independen. Algoritma ini bekerja dengan membangun kumpulan pohon keputusan (*decision trees)* yang acak, kemudian menggabungkan prediksi dari pohon-pohon tersebut untuk menghasilkan prediksi akhir yang lebih akurat dan stabil. Kelebihan algoritma ini adalah kemampuannya dalam menangani hubungan non-linear dan kompleks, serta memberikan prediksi yang lebih akurat (Farhanuddin dkk., 2024).



Gambar 2. 4 Cara Kerja *Random Forest*

2.1.11 Pengujian IoT

Menurut Dias dkk. (2018), pendekatan pengujian dapat dilakukan dalam beberapa tingkatan, bergantung pada cakupan dan tujuan pengujian. Beberapa tingkatan pengujian yang umum digunakan meliputi *Unit Testing*, *Integration Testing*, *System Testing*, dan *Acceptance Testing*.

1) *Unit Testing*

   Unit Testing adalah proses pengujian terhadap unit atau komponen perangkat keras dan perangkat lunak secara individual. Tujuannya adalah untuk mengisolasi setiap bagian dari sistem, memastikan bahwa bagian tersebut memenuhi persyaratan dan berfungsi sesuai dengan yang diharapkan.

2) *Integration Testing*

Integration Testing dilakukan setelah *Unit Testing* untuk menguji bagaimana berbagai komponen perangkat keras dan perangkat lunak berinteraksi satu sama lain. Pada tahap ini, modul-modul yang telah diuji secara individual digabungkan dan diuji untuk memastikan bahwa mereka dapat beroperasi secara bersama-sama tanpa mengalami kesalahan.

3) *System Testing*

System Testing adalah pengujian terhadap sistem secara keseluruhan dalam kondisi yang menyerupai lingkungan operasional sebenarnya. Pengujian ini bertujuan untuk memastikan bahwa sistem memenuhi spesifikasi yang telah ditentukan dan bekerja sesuai dengan yang diharapkan dalam berbagai skenario penggunaan.

4) *Acceptance Testing*

Acceptance Testing merupakan tahap akhir pengujian yang dilakukan untuk menentukan apakah sistem telah memenuhi kriteria penerimaan yang ditetapkan oleh pengguna atau pemangku kepentingan. Pengujian ini dapat dilakukan dalam lingkungan nyata untuk mengevaluasi kinerja sistem dalam kondisi operasional yang sesungguhnya. Jika sistem memenuhi standar yang ditetapkan, maka sistem dianggap siap untuk digunakan atau diimplementasikan secara luas.

2.1.12 Metrik Evaluasi Random Forest Regression

Menurut Tatachar (2021), metrik yang digunakan untuk mengevaluasi model regresi adalah *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *R-Squared* (R^2).

1) *Mean Squared Error* (MSE)

Mean Squared Error disebut juga *Mean Squared Deviation* adalah selisih kuadrat antara nilai yang sebenarnya dan nilai prediksi. MSE menunjukkan seberapa dekat garis regresi terbaik dengan kumpulan data. Semakin kecil nilai MSE, semakin akurat prediksinya. Rumus dari MSE sebagai berikut:

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

Dimana,

n = Jumlah prediksi

Yi = Nilai aktual

$\hat{Y}i$ = Nilai prediksi

2) *Root Mean Squared Error* (RMSE)

Root Mean Squared Error disebut juga *Root Mean Squared Deviation* adalah akar kuadrat dari rata-rata kuadrat semua *error*. Dengan kata lain, RMSE adalah standar deviasi dari *error*. Sama seperti MSE, RMSE juga menunjukkan seberapa dekat garis regresi terbaik dengan kumpulan data. Rumus dari RMSE sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Yi - \hat{y}i)^2} \quad (3)$$

Dimana,

n = Jumlah prediksi

Yi = Nilai aktual

$\hat{Y}i$ = Nilai prediksi

3) *Mean Absolute Error* (MAE)

Mean Absolute Error disebut juga *Mean Absolute Deviation* menunjukkan rata-rata dari selisih absolut antara nilai yang sebenarnya dan nilai prediksi. Perbedaan antara MAE dan MSE adalah MAE mengambil selisih absolut antara nilai yang sebenarnya dan nilai prediksi, sedangkan MSE mengambil selisih kuadrat. Rumus dari MAE sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| Yi - \hat{Y}i \right| \quad (4)$$

Dimana,

n = Jumlah prediksi

Yi = Nilai aktual

$\hat{Y}i$ = Nilai prediksi

4) *R-Squared* (R^2)

R^2 disebut sebagai koefisien determinasi. *R-Squared* menentukan proporsi variansi dalam variabel dependen yang dapat dijelaskan oleh variabel independen. R^2 memberikan informasi tentang sejauh mana *goodness of fit*

(sejauh mana nilai yang diamati cocok dengan nilai yang di prediksi) untuk model regresi. Nilai R^2 berkisar antara 0 dan 1, dimana nilai semakin mendekati 1, semakin baik. Rumus dari R^2 sebagai berikut:

$$R^2 = 1 - \frac{SSR}{TSS} \quad (5)$$

Dimana,

SSR = *Sum of Squares of Residual* / Jumlah Kuadrat Residual

TSS = *Total Sum of Squares* / Total Jumlah Kuadrat

2.1.13 Flask

Flask adalah sebuah web framework Python yang berfungsi sebagai kerangka kerja aplikasi dan tampilan dari sebuah web. Flask termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya (Santoso dan Saian, 2023). Flask sangat cocok untuk membangun API yang ringan dan efisien.

Gambar 2. 5 Logo Flask

```python
app.py > ...
from flask import Flask, jsonify, request, make_response

app = Flask(__name__)
app.config["DEBUG"] = True

@app.route('/', methods=['GET'])
def hello():
    data = [{
        'nama': 'Galih',
        'pekerjaan': 'Web Engineer',
        'pesan': 'Hello Wolrd'
    }]
    return make_response(jsonify({'data': data}), 200)

app.run()
```

Gambar 2. 6 Tampilan Kode Flask API

2.1.14 React JS

React JS adalah library JavaScript yang bersifat *open-source* yang digunakan untuk membangun *User Interface* (UI) secara spesifik untuk satu halaman dalam aplikasi. React JS digunakan sebagai salah satu *framework* untuk membuat bagian *front-end* dari sebuah aplikasi. Dengan React, user dapat membuat *reusable component* atau jenis komponen yang dapat digunakan kembali tanpa harus membangunnya dari awal lagi. React JS dapat mendesain tampilan sederhana untuk setiap level dalam aplikasi sehingga dapat digunakan untuk membuat dan mengembangkan pembuatan aplikasi berbasis web (Sulistyorini, Sova dan Ramadhan, 2022).

Gambar 2. 7 Logo React JS

2.2 Penelitian Terdahulu

Dalam membangun proyek ini, terdapat beberapa penelitian terdahulu yang digunakan sebagai referensi dalam penulisan proyek ini. Penelitian yang dilakukan oleh Tran dkk. (2022) mengembangkan sistem pemantauan menggunakan sensor dengan biaya yang rendah untuk mengumpulkan data polutan, yaitu PM2.5, CO2, CO, suhu, dan kelembapan. Data tersebut diproses menggunakan beberapa algoritma machine learning, yaitu *K-Nearest Neighbour* (KNN), *Expectation-Maximization* (EM), *Multiple Imputation by Chained Equations* (MICE), *Autoregressive-Moving-Average* (ARMA), *Autoregressive Integrated Moving Average* (ARIMA), dan *Long Short-Term Memory* (LSTM) untuk memprediksi kualitas udara di masa depan. Hasilnya ditampilkan pada halaman web untuk memantau dan memprediksi kualitas udara secara *real-time*.

Kamil (2024) melakukan penelitian yang berfokus pada prediksi *Air Quality Index* (AQI) menggunakan algoritma *Linear Regression* di Jakarta. Data yang

digunakan adalah data dari ISPU DKI Jakarta tanggal 1 Maret 2021 sampai 31 Desember 2021. Parameter yang digunakan untuk prediksi *Linear Regression* adalah PM2.5, PM10, SO2, CO, O3, dan NO2. Visualisasi dari hasil prediksi ditampilkan dalam website beserta nilai parameter, nilai dan kategori dari AQI.

Penelitian yang dilakukan oleh Banciu, Florea dan Bogdan (2024) menganalisis temperatur, kelembapan, PM10, dan PM2.5 menggunakan IoT. Data dari sensor dilatih menggunakan kedua algoritma regresi, yaitu *Linear Regression* dan *Random Forest*. Model *Random Forest* menunjukkan hasil yang paling bagus secara keseluruhan. Data disimpan pada ThingSpeak dan ditampilkan dalam bentuk grafik yang dihasilkan pada ThingSpeak.

Penelitian yang dilakukan oleh Khumaidi, Raafi'udin dan Solihin (2020) mengembangkan model dengan algoritma *Long Short Term Memory* (LSTM) untuk memprediksi data *time series*, yaitu kualitas udara di kota Bandung. Parameter yang digunakan adalah PM10, ISPU, suhu, dan kelembapan. Pemodelan menghasilkan keakuratan prediksi yang cukup baik untuk 3 parameter (suhu, kelembapan, ISPU). Hasil prediksi yang dihasilkan dari keempat parameter pengujian berdasarkan urutan dari yang paling baik, yaitu prediksi kelembapan, suhu, ISPU, dan PM10 dengan hasil prediksi yang kurang baik.

Ameer dkk. (2019) menganalisis perbandingan teknik *machine learning* yang digunakan untuk memprediksi kualitas udara. Algoritma yang diuji adalah *Decision Tree Regression* (DTR), *Random Forest Regression* (RFR), *Multi-layer Perceptron Regression* (MLP), dan *Gradient Boosting Regression* (GBR). Hasil penelitian menunjukkan bahwa RFR memiliki kinerja yang paling baik dalam hal waktu pemrosesan dan tingkat kesalahan (*error rate*) sehingga menjadi pilihan ideal untuk prediksi polusi udara.

Pasedja (2021) membuat sistem *monitoring* kualitas udara yang ditempatkan di lingkungan Politeknik Caltex Riau untuk membantu dalam pengambilan keputusan darurat asap. Sistem dirancang menggunakan NodeMCU ESP8266 sebagai *Master Control Unit* (MCU) dan menggunakan beberapa sensor, yaitu sensor GP2Y1010AU0F sebagai pendeteksi kadar PM10 di udara, sensor DHT11 sebagai pendeteksi suhu dan kelembapan, dan sensor MQ-7 sebagai pendeteksi kadar karbon monoksida di udara. Pengujian sensor PM10 memiliki tingkat *error*

rata-rata dengan alat pembanding sebesar 9,56%, sensor suhu sebesar 8,42%, dan sensor kelembapan sebesar 2,89%. Data sensor tersebut dikirimkan pada database dan ditampilkan pada halaman *website* local (localhost) secara *real-time*.

Penelitian yang akan dilakukan menggunakan data dari 3 sensor, yaitu SDS011 untuk mengukur PM2.5 dan PM10, MQ-135 untuk mengukur NO2, dan DHT22 untuk mengukur suhu dan kelembapan. Algoritma *Random Forest Regression* digunakan untuk memprediksi kualitas udara di masa depan berdasarkan data historis. Hasilnya akan ditampilkan pada website untuk visualisasi dari data dan memberikan notifikasi jika kualitas udara menjadi buruk.

Tabel 2. 3 Penelitian terdahulu dan yang akan dilakukan

Peneliti	Parameter	Algoritma	Hasil
Tran dkk. (2022)	PM2.5, CO2, CO, suhu, dan kelembapan	KNN, EM, MICE, ARMA, ARIMA, LSTM	Website
Kamil (2024)	PM2.5, PM10, SO2, CO, O3, dan NO2	*Linear Regression*	Website
Banciu, Florea dan Bogdan (2024)	Temperatur, kelembapan, PM10, dan PM2.5	*Linear Regression* dan *Random Forest*	Visualisasi menggunakan ThingSpeak
Khumaidi, Raafi'udin dan Solihin (2020)	PM10, ISPU, suhu, dan kelembapan	LSTM	Evaluasi dari model LSTM
Ameer dkk. (2019)	PM2.5, temperatur, kelembapan, tekanan udara, kecepatan angin, arah angin, titik embun, dan curah hujan	DTR, RFR, MLP, GBR	Evaluasi dari masing-masing model yang digunakan
Pasedja (2021)	PM10, suhu, kelembapan, dan karbon monoksida	-	Ditampilkan pada *website*
Penelitian yang akan dilakukan	PM2.5, PM10, NO2, suhu, dan kelembapan	*Random Forest Regression*	*Website*