

**PENGEMBANGAN SISTEM PERINGATAN DINI
KEBOCORAN GAS MENGGUNAKAN *DEEP
NEURAL NETWORK* (DNN) BERBASIS *INTERNET of
THINGS* (IoT)**

TUGAS AKHIR

Diajukan Oleh:

**IHSAN PERDANA PUTRA
NIM. 200705028**

**Mahasiswa Falkutas Sains dan Teknologi
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2025 / 1446H**

**PENGEMBANGAN SISTEM PERINGATAN DINI KEBOCORAN GAS
MENGUNAKAN DEEP NEURAL NETWORK (DNN) BERBASIS
INTERNET of THINGS (IoT)**

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi UIN Ar-raniry Banda Aceh sebagai salah satu persyaratan penulisan Tugas akhir dalam Prodi Teknologi Informasi

Oleh:

IHSAN PERDANA PUTRA

200705028

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**

Disetujui Untuk di Munaqasyahkan Oleh:

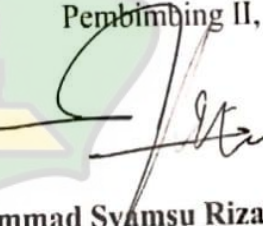
Pembimbing I,



Khairan AR, M.Kom

NIP: 198607042014031001

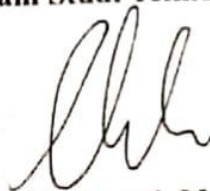
Pembimbing II,



Muhammad Syamsu Rizal, S.T., M.T

Mengetahui,

Ketua Program Studi Teknologi Informasi



Malahayati, M.T

NIP. 198301272015032003

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM PERINGATAN DINI KEBOCORAN GAS MENGUNAKAN DEEP NEURAL NETWORK (DNN) BERBASIS INTERNET of THINGS (IoT)

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munqasah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh Dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Pada Program Studi Teknologi Informasi

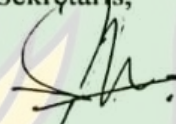
Pada Hari/Tanggal : Rabu, 15 Januari 2025
15 Rajab 1446H

Panitia Ujian Munqasah Tugas Akhir :

Ketua,


Khairan AR, M.Kom
NIP: 198607042014031001

Sekretaris,


Muhammad Syamsu Rizal, S.T., M.T

Penguji I,

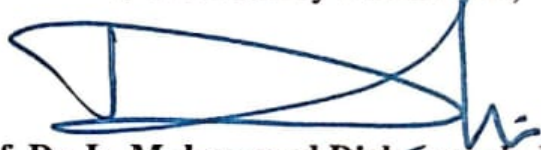

Hendri Ahmadian, S.Si, M.I.M
NIP. 198301042014031002

Penguji II,


Ridha Ilahi, M.T
NIP. 197905302014031001

Mengetahui :

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,


Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini :

Nama : Ihsan Perdana Putra
NIM : 200705028
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul Tugas Akhir : Sistem Peringatan Dini Kebocoran Gas Menggunakan Deep Neural Network (DNN) Berbasis Internet of Things (IoT)

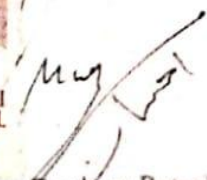
Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah tugas akhir orang lain;
3. Tidak menggunakan tugas akhir orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik tugas akhir;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri tugas akhir ini dan mampu bertanggung jawab atas tugas akhir ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas tugas akhir saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 13 Januari 2025


(Ihsan Perdana Putra)

10000
METERAI TEMPEL
DE 394AMX129826791

ABSTRAK

Nama : Ihsan Perdana Putra
NIM : 200705028
Program Studi : Teknologi Informasi
Tanggal Sidang : 15 Januari 2025 / 15 Rajab 1446 H
Jumlah Halaman : 73 Halaman
Pembimbing I : Khairan AR, M.Kom
Pembimbing II : Muhammad Syamsu Rizal, S.T., M.T
Kata Kunci : IoT, Deep Neural Network, Sistem Peringatan Dini, Kebocoran Gas, Ubidots

Penelitian ini mengembangkan sistem peringatan dini kebocoran gas berbasis Internet of Things (IoT) dengan algoritma Deep Neural Network (DNN). Sistem ini dirancang menggunakan mikrokontroler ESP32, sensor MQ-2 untuk deteksi gas, dan DHT22 untuk pemantauan suhu serta kelembapan. Data yang dikumpulkan oleh sensor diproses melalui DNN untuk mendeteksi kebocoran gas secara akurat. Model DNN dilatih menggunakan dataset yang mencakup parameter seperti suhu, kelembapan, konsentrasi gas hidrogen, dan etanol. Teknik normalisasi data Min-Max Scaler serta Synthetic Minority Over-sampling (SMOTE) digunakan untuk meningkatkan performa model.

Hasil penelitian menunjukkan akurasi model sebesar 98% dengan f1-score 0.98, mengindikasikan kemampuan sistem dalam mendeteksi kebocoran gas secara efisien. Data prediksi kemudian dikirim ke platform Ubidots secara real-time, memungkinkan monitoring jarak jauh. Sistem ini berpotensi untuk diimplementasikan dalam lingkungan rumah tangga maupun industri guna meningkatkan keselamatan terhadap bahaya kebakaran dan keracunan gas.

Kata Kunci: IoT, Deep Neural Network, Sistem Peringatan Dini, Kebocoran Gas, Ubidots.

ABSTRACT

Name : Ihsan Perdana Putra
Student ID : 200705028
Study Program : Information Technology
Thesis Defense Date : January 15, 2025 / Rajab 15, 1446 H
Total Pages : 73 Pages
Supervisor I : Khairan AR, M.Kom
Supervisor II : Muhammad Syamsu Rizal, S.T., M.T
Keywords : IoT, Deep Neural Network, Early Warning System, Gas Leakage, Ubidots

This study develops an early warning system for gas leakage detection based on the Internet of Things (IoT) using Deep Neural Network (DNN) algorithms. The system is designed using an ESP32 microcontroller, an MQ-2 sensor for gas detection, and a DHT22 sensor for monitoring temperature and humidity. Data collected by the sensors is processed through the DNN to accurately detect gas leaks. The DNN model was trained using a dataset that includes parameters such as temperature, humidity, hydrogen gas concentration, and ethanol concentration. Min-Max Scaler data normalization and Synthetic Minority Over-sampling Technique (SMOTE) were employed to enhance model performance.

The results of the study demonstrated a model accuracy of 98% with an F1-score of 0.98, indicating the system's efficiency in detecting gas leaks. Prediction data is transmitted to the Ubidots platform in real-time, enabling remote monitoring. This system has the potential to be implemented in both household and industrial environments to enhance safety against fire hazards and gas poisoning.

Keywords: IoT, Deep Neural Network, Early Warning System, Gas Leakage, Ubidots

KATA PENGANTAR

Bissmillahirrahmanirahim

Segala puji bagi Allah, Tuhan dari semesta alam. Shalawat dan salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad Shallallahu ‘Alaihi Wa sallaam, keluarga dan sahabatnya. Alhamdulillah dengan rahmat Allah yang Maha Rahman dan yang Maha Rahim, sehingga penulis dapat menyelesaikan tugas akhir dengan judul **“Pengembangan Sistem Peringatan Dini Kebocoran Menggunakan Deep Neural Network (DNN) Berbasis Internet of Things (IoT)”** ini. Tugas akhir ini merupakan salah satu syarat untuk menyelesaikan program studi Strata satu Teknologi Informasi pada Fakultas Sains dan Teknologi di Universitas Islam Negeri Ar-Raniry Banda Aceh.

Ucapan terima kasih penulis sampaikan kepada berbagai pihak yang menjadi sebab dari mereka penulis belajar, mendapatkan ilmu, mendapatkan dukungan, serta mendapatkan hal yang bermanfaat lainnya sehingga penulis sampai pada titik menyelesaikan tugas akhir ini. Terutama dalam konteks ini penulis sampaikan kepada:

1. Kedua orang tua dan keluarga besar penulis yang selalu memberikan dukungan, dan doa yang tak ternilai harganya selama ini. Hanya Allah yang mampu membalas kasih sayang mereka yang tak terhingga, semoga selalu Allah limpahkan rahmat kepada mereka dan mendapatkan ridha serta cinta dari-Nya.
2. Bapak Khairan Ar, M.Kom dan Bapak Muhammad Syamsu Rizal, S.T., M.T. selaku pembimbing yang selalu bersedia meluangkan waktu dan pikirannya untuk membimbing penulis demi kesempurnaan tugas akhir ini. Ketua dan Sekretaris Program Studi Teknologi Informasi, Ibu Malahayati, M.T. dan Bapak Khairan Ar, M.Kom, serta Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan dalam bidang

3. Teknologi Informasi kepada penulis sehingga penulis mampu menyelesaikan tugas akhir ini.
4. Staf Prodi Ibu Cut Ida Rahmadiana S,Si. yang telah membantu membantu penulis dalam hal pengurusan administrasi dan surat-surat untuk keperluan penyelesaian tugas akhir.
5. Teman-teman mahasiswa Prodi Teknologi Informasi dan teman teman penulis di luar perkuliahan yang selalu memberikan dukungan dan perspektif pemikiran yang membantu peneliti dalam membuat tugas akhir ini.

Penulis berharap hasil tugas akhir ini dapat berguna dalam meningkatkan kesiapan siagaan terhadap bencana kebakaran dan dapat berguna untuk bangsa dan negara.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi mendapatkan hasil yang lebih baik. Semoga perjalanan mempelajari dan berkarya (penyusunan tugas akhir) pada salah satu ilmu milik-Nya ini dapat menghantarkan penulis agar dapat mengenal-Nya dan kekasih-Nya lebih banyak serta mendapatkan ridho dan cinta-Nya yang Maha Rahman dan Rahim.

Banda Aceh, 13 Januari 2025

Penulis,

Ihsan Perdana Putra

DAFTAR ISI

LEMBAR PENGAJUAN TUGAS AKHIR.....	Error! Bookmark not defined.
LEMBAR PENGESAHAN	Error! Bookmark not defined.
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	Error! Bookmark not defined.
ABSTRAK.....	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	x
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
BAB II LANDASAN TEORI.....	5
2.1 Relevansi Penelitian	5
2.2 Landasan Teori	7
2.2.1 Internet of Things (IoT)	7
2.2.2 Modul ESP32 Dev Kit V1	7
2.2.3 Sensor MQ-2	8
2.2.4 Sensor DHT22	9

2.2.5	Deep Neural Network (DNN)	10
2.2.6	Sistem Peringatan Dini Berbasis IoT.....	12
2.2.7	TensorFlow.....	12
2.2.8	Python.....	13
2.2.9	Ubidots.....	13
BAB III METODE PENELITIAN		15
3.1	Waktu dan Tempat Penelitian.....	15
3.2	Alat dan Bahan.....	15
3.3	Alur Penelitian.....	15
3.4	Tahap Penelitian	16
3.5	Observasi Awal.....	16
3.6	Analisa Kebutuhan Alat dan Bahan	16
3.7	Perancangan sistem	17
3.7.1	Perangkat Keras (<i>Hardware</i>).....	17
3.7.2	Perangkat Lunak (<i>Software</i>).....	19
3.7.3	<i>Device Ubidots</i>	19
3.7.4	Integritas Ubidots dengan Arduino IDE	20
3.8	Kalibrasi Sensor	21
3.9	Arsitektur DNN	22
3.10	Pengumpulan data	24
3.11	Rincian dataset	25
3.12	Normalisasi Data	25
3.12.1	Pemisahan kolom	26
3.12.2	Metode <i>Synthetic Minority Over-sampling</i> (SMOTE)	27
3.12.3	Data Training.....	29

3.12.4 Data Testing.....	29
3.12.5 Data Validasi.....	30
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	31
4.1 Pengujian Sensor MQ2 dan DHT22.....	31
4.1.1. Hasil Output Pendeteksian Sensor MQ2 dan DHT22	31
4.1.2. Hasil Pengiriman Data ke Python	32
4.2 Hasil Normalisasi Data.....	33
4.3 Model DNN.....	35
4.4 Hasil <i>Model Accuracy</i> dan <i>Model Loss</i>	35
4.5 Pengiriman Data ke <i>Device Ubidots</i>	37
BAB IV KESIMPULAN DAN SARAN.....	42
5.1 Kesimpulan.....	42
5.2 Saran.....	42
DAFTAR PUSTAKA.....	43
LAMPIRAN	45

DAFTAR GAMBAR

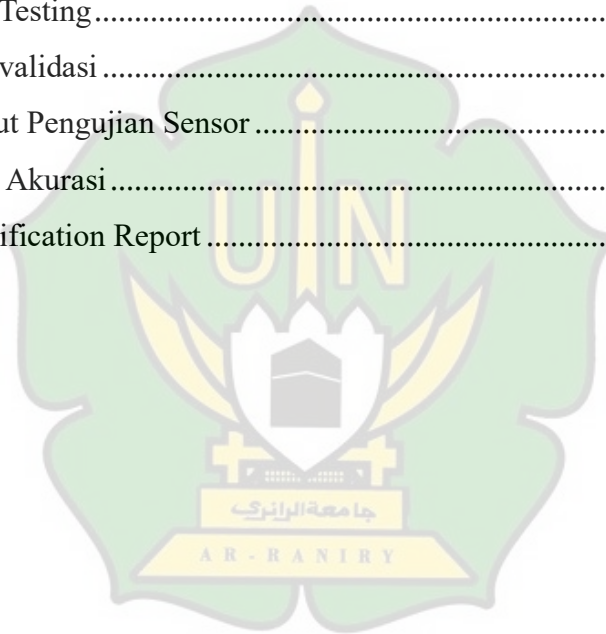
Gambar 2. 1 ESP32 DEV Kit.....	8
Gambar 2. 2 Sensor MQ2.....	9
Gambar 2. 3 Sensor DHT22.....	9
Gambar 2. 4 Deep Neural Network (DNN)	10
Gambar 3. 1 Alur Penelitian.....	16
Gambar 3. 2 Perancangan Perangkat Keras	17
Gambar 3. 3 Prototype Hardware.....	18
Gambar 3. 4 Perancangan Perangkat Lunak	19
Gambar 3. 5 Devices ubidost	20
Gambar 3. 6 Arduino IDE	21
Gambar 3. 7 kodingan kalibrasi sensor	22
Gambar 3. 8 Arsitektur Deep Neural Network layer.....	23
Gambar 3. 9 Tampilan Dataset.....	24
Gambar 3. 10 kodingan pemisah kolom.....	26
Gambar 3. 11 kodingan metode smote	28
Gambar 4. 1 Hasil Pengiriman Data Sensor.....	32
Gambar 4. 2 Dataset Sebelum Normalisasi.....	33
Gambar 4. 3 kodingan min-max scaler	34
Gambar 4. 4 Data yang telah di Normalisasi.....	34
Gambar 4. 5 kodingan pembuatan model dnn.....	35
Gambar 4. 6 Model Accuracy dan model loss	36
Gambar 4. 7 proses pengiriman data ke ubidots	37
Gambar 4. 8 Tanda terkoneksi ke ubidots	38
Gambar 4. 9 Tampilan Device Ubidots	38
Gambar 4. 10 Tanda Gas Terdeteksi.....	39
Gambar 4. 11 Tampilan ubidots tidak terdeteksinya gas.....	39

Gambar 4. 12 Tampilan Ubidots Terdeteksinya gas..... 40



DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	5
Tabel 3. 1 Alat dan Bahan	15
Tabel 3. 2 Rincian PIN	18
Tabel 3. 3 Parameter Model	24
Tabel 3. 4 Rincian data	25
Tabel 3. 5 Parameter Kolom.....	26
Tabel 3. 6 Data Training	29
Tabel 3. 7 Data Testing	29
Tabel 3. 8 Data validasi	30
Tabel 3. 9 Output Pengujian Sensor	31
Tabel 4. 1 Hasil Akurasi	36
Tabel 4. 2 Classification Report	37



LAMPIRAN

Lampiran 1 Dokumentasi Hardware	45
Lampiran 2 Kode Pada <i>Arduino IDE</i>	45
Lampiran 3 Kode Normalisasi Data	47
Lampiran 4 Kode Pengiriman Data ke Ubidots	52
Lampiran 5 Data Normalisasi	56



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi saat ini terus maju dan melibatkan banyak kalangan. Peralatan modern dirancang untuk efisiensi dan kemudahan penggunaan. Banyak dari peralatan ini dapat bekerja tanpa instruksi manual yang berkelanjutan. Ada beberapa permasalahan pada masyarakat dalam menghadapi bencana kebakaran seperti kesiapsiagaan masyarakat dalam menghadapi bencana tersebut, oleh karena itu kurangnya infrastruktur pencegahan kebakaran seperti alarm kebakaran, sistem pemadam otomatis, dan akses mudah ke layanan pemadam kebakaran dapat meningkatkan risiko kerugian saat terjadi kebakaran.

Dalam konteks ilmiah, kebakaran biasanya teridentifikasi ketika api telah berkembang atau ketika emisi asap hitam teramati keluar dari struktur bangunan. Konsekuensi kerugian yang ditimbulkan oleh insiden kebakaran ini dapat sangat substansial. Penanganan kebakaran yang efektif dan upaya pengurangan kerugian dapat dicapai melalui deteksi kebakaran yang canggih dan responsif diperlukan untuk memberikan awal tentang keberadaan kebakaran di dalam ruangan atau di lokasi publik, memungkinkan tindakan preventif yang cepat untuk menghindari kerugian lebih lanjut akibat kebakaran.

Kebocoran gas bukan hanya menimbulkan kebakaran saja melainkan ada dampak lain yang berefek dari kebocoran gas tersebut yaitu keracunan gas, seperti karbon monoksida (CO) dan gas alam, hal ini merupakan ancaman serius yang sering kali tidak disadari hingga terlambat. Dampak dari keracunan gas ini sangat berbahaya, termasuk gangguan Kesehatan.

IoT adalah salah satu pilar utama dalam perkembangan revolusi industri 4.0. Dasar dari IoT dapat dianggap sebagai jaringan global yang terdiri dari berbagai perangkat yang saling terhubung, mengandalkan teknologi sensorik, komunikasi, jaringan, dan pemrosesan informasi. IoT secara luas dapat didefinisikan sebagai jaringan global yang menghubungkan objek, perangkat, dan entitas baik secara fisik

maupun virtual melalui objek cerdas, komunikasi, dan kemampuan aktuasi (Sinta dkk., 2023).

Dengan merancang dan membangun sistem pendeteksi kebakaran dan monitoring berbasis IoT menggunakan mikrokontroler Esp32 Dev kit V1, sensor yang digunakan termasuk MQ-2 untuk mendeteksi asap dan gas, dan DHT22. (Subhan & Virgian Shaka Yudha Sakti, 2023).

Pada penelitian ini, focus pada pemanfaatan teknologi *Internet of Things* (IoT) dan *Deep Neural Network* (DNN) sebagai solusi untuk sistem peringatan dini kebocoran gas menjadi relevan. Dengan memanfaatkan konektivitas internet yang semakin meluas, Iot dan Dnn memungkinkan pengembangan sistem yang lebih responsif dan terintegrasi dengan perangkat yang sudah ada di Masyarakat.

Beberapa implementasi kebocoran gas yang relevan adalah menggunakan sensor gas seperti MQ-2 yang terhubung ke jaringan *wifi* atau seluler. Sensor ini dapat mendeteksi kebocoran gas secara *real-time* dan mengirimkan peringatan kepada pengguna melalui berbagai *device*. Keuntungan lain dari teknologi IoT adalah kemampuannya untuk mencakup wilayah lain yang luas. Termasuk daerah terpencil yang sulit dijangkau oleh sistem peringatan konvensional (Wikantama dkk., 2024).

Pengembangan sistem peringatan dini untuk kebocoran gas menggunakan DNN melibatkan penggunaan teknologi untuk meningkatkan deteksi dan respons terhadap kebocoran gas yang berpotensi berbahaya. Sistem ini didasarkan pada pengumpulan data sensor gas yang mencatat konsentrasi gas tertentu variabel lingkungan lainnya. Data tersebut kemudian di proses untuk membersihkan noise dan dinormalisasikan agar konsisten dalam skala. Dengan memanfaatkan model DNN, sistem dapat diprediksi kemungkinan kebocoran gas dengan akurat berdasarkan pola dan tren teridentifikasi dalam data historis. Tujuan utama sistem ini adalah untuk memberikan notifikasi dini kepada pengguna atau operator ketika terdeteksi potensi kebocoran, memungkinkan respons cepat dan mitigasi risiko yang lebih efektif dalam lingkungan industri atau rumah tangga

Dengan demikian, penelitian ini diharapkan dapat memberikan solusi yang lebih efektif dalam meningkatkan keselamatan masyarakat terhadap ancaman kebocoran gas dan potensi kebakaran di masa depan.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana kinerja *Deep Neural Network (DNN)* untuk sistem peringatan dini terhadap kebocoran gas berbasis *Internet of Things (IoT)*?

1.3 Tujuan Penelitian

Berdasarkan uraian rumusan masalah tersebut, tujuan yang ingin dicapai dari penelitian ini adalah mengembangkan *prototype* sistem peringatan dini berbasis IoT yang menggunakan DNN untuk mendeteksi kebocoran gas dan memberikan hasil prediksi yang akurat.

1.4 Manfaat Penelitian

Penelitian ini akan bermanfaat dalam hal:

1. Secara praktis, rancangan alat prediksi kebakaran dapat membantu masyarakat untuk mengurangi terjadinya kerugian saat terjadi kebakaran.
2. Secara teoritis, penelitian ini akan menambah referensi dan pengetahuan tentang bagaimana merancang alat pendeteksi dengan *output* prediksi.
3. Secara kebijakan, sistem ini dapat memberikan rekomendasi taktis kepada petugas pemadam kebakaran untuk penanganan yang lebih efisien.

1.5 Batasan Masalah

Batasan dalam penelitian ini adalah sebagai berikut :

1. Sistem hanya akan mendeteksi jenis gas tertentu, seperti *gas ethanol* (C_2H_5OH), *gas hidrogen* (H_2), *Temperature* (C), dan *Humidity* sesuai dengan dataset yang diteliti dan aplikasi spesifik dari sistem.
2. Jenis sensor gas (MQ2) dan sensor suhu (DHT22) yang digunakan dalam sistem dibatasi pada sensor yang kompatibel dengan teknologi IoT dan mampu mengirimkan data prediksi.

3. Model *deep neural network* (DNN) dibatasi pada dataset yang tersedia dan mencerminkan kondisi operasional, dengan mempertimbangkan sumber data yang realistis dan terukur.
4. Waktu respon sistem dibatasi pada interval hitungan detik hingga menit, dengan memastikan sistem dapat memberikan peringatan dini secara efektif.
5. Dikarnakan ketidakseimbangannya dataset ketika jumlah sampel dalam satu kelas jauh lebih besar dari kelas lain dan menyebabkan model belajar dengan performa yang buruk maka di perlukan teknik *Sysnthetic Minority Over-sampling* (SMOTE)



BAB II LANDASAN TEORI

2.1 Relevansi Penelitian

Dalam konteks penelitian yang akan dilakukan, penulis memerlukan referensi atau studi terkait untuk menghindari duplikasi dan plagiarisme. Penelitian terdahulu bertujuan sebagai bahan perbandingan dan pedoman. Selain itu, untuk melihat perbedaan dengan penelitian ini, maka dalam landasan teori ini penulis mencantumkan hasil-hasil penelitian terdahulu seperti terlihat pada tabel II :

Tabel 2. 1 Penelitian Terdahulu

No	Referensi	Judul Penelitian	Hasil Penelitian	Persamaan
				Perbedaan
1	(Suryana, 2021)	Implementasi Modul Sensor MQ-2 untuk Mendeteksi Adanya Polutan Gas di Udara.	Penelitian ini membantu mengetahui gas apa saja yang dapat dideteksi oleh MQ-2 : LPG, <i>Hidrogen</i> (H ₂), <i>Metana</i> (CH ₄), <i>Karbon Monoksida</i> (CO), Alkohol, Asap Rokok dan <i>Propana</i>	Hampir secara keseluruhan, penelitian ini sama dengan Penelitian oleh penulis, dikarenakan sama-sama mendeteksi gas dan asap. Perbedaan nya yaitu penulis hanya mendeteksi gas atau asap yang disebabkan oleh adanya api atau kebocoran gas LPG.
2	(Nur Anggraeni dkk., 2023)	Prediksi penyakit jantung menggunakan metode deep neural network dengan	Aplikasi ini dirancang untuk menerima input data pasien dan melakukan prediksi apakah pasien tersebut positif terkena	Persamaan dari penelitian ini yaitu sama-sama menggunakan arsitektur DNN dengan hasil prediksi

No	Referensi	Judul Penelitian	Hasil Penelitian	Persamaan
				Perbedaan
		memanfaatkan <i>internet of things</i>	penyakit jantung atau negatif.	Perbedaan nya yaitu penelitian ini menggunakan dataset dan inputan yang berbeda.
3	(Istiyanto dkk., 2022)	Alat pendeteksi dini kebocoran gas LPG dengan sensor MQ2 dan sensor api berbasis Iot menggunakan <i>NodeMCU</i> .	Berdasarkan penelitian dan pembahasan yang telah diuraikan dapat disimpulkan bahwa alat ini dapat bertugas sebagai <i>early warning</i> melalui aplikasi telegram.	Hampir secara keseluruhan, penelitian ini sama dengan penelitian penulis, dikarenakan sama dalam pengimplementasian sebagai pendeteksi terjadinya kebakaran dan dapat mengirimkan notifikasi peringatan dini. Perbedaannya yaitu notifikasi penelitian ini menggunakan aplikasi telegram, sedangkan penulis menggunakan <i>iot cloud</i> dan <i>platform</i> lain.
4	(Riyadi, 2024)	Prediksi daerah potensi bencana kebakaran hutan akibat perubahan penggunaan lahan di kabupaten paser, kalimantan timur.	Penelitian ini menghasilkan peta daerah potensi kebakaran hutan di kabupaten paser, kalimantan timur, yang menunjukkan wilayah dengan resiko tinggi akibat perubahn iklim.	Penelitian ini sama-sama menggunakan sistem prediksi peringatan dini dan dataset Pada perbedaan penelitian ini penulis menggunakan metode DNN dan memiliki beberapa sensor sebagai inputan nya.

2.2 Landasan Teori

Adapun teori-teori yang mendukung penelitian ini antara lain sebagai berikut:

2.2.1 Internet of Things (IoT)

Internet of Things (IoT) adalah konsep dimana objek fisik, seperti perangkat, kendaraan, dan peralatan, yang terhubung ke internet dan saling berkomunikasi tanpa interaksi manusia. IoT memungkinkan pengumpulan data secara *real-time* dan respons cepat terhadap situasi darurat. IoT memainkan peranan penting dalam berbagai sektor, termasuk pertahanan, perindustrian, dan pemerintahan (Andika, 2024).

IoT memiliki ciri utama berupa dapat terkoneksi ke internet dan menggunakan sensor untuk mengumpulkan serta dapat menganalisis secara *real-time*. Data yang dikumpulkan dapat digunakan untuk monitoring jarak jauh dengan otomatis dengan mengirimkan data kepada pengguna.

Pemanfaatan IoT pada penelitian ini ialah dengan menggunakan sensor dengan akurat di harapkan mendapatkan data tekanan gas yang bocor yang mana nantinya akan memberikan peringatan dini sebagai mitigasi dini kebocoran gas, untuk menghindari terjadinya keracunan ketika menghirup gas berlebihan dan juga terjadinya kebakaran di sebabkan adanya gas bocor.

2.2.2 Modul ESP32 Dev Kit V1

ESP32 Dev Kit V1 adalah Mikrokontroler yang sudah menyediakan *Wifi* dan *Bluetooth* yang terintegrasi secara *built-in* di dalam *Chipnya*. Modul ini dapat mendukung kinerja alat dan sensor yang akan digunakan terus-menerus. *ESP32 Dev Kit V1* memiliki prosesor *Dual-core 32bit* yang lebih cepat dari pada *ESP8266* (Athallah & Agung, 2022). *Modul ESP32 Dev Kit V1* dapat dilihat pada gambar 2.1.

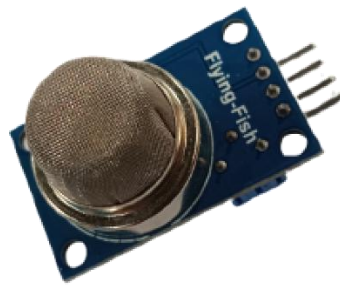


Gambar 2. 1 ESP32 DEV Kit

ESP32 memiliki prosesor Dual-Core yang di lengkapi dengan Xtensa LX6 yang memberikan frekuensi hingga 240 MHz. keunggulan utama dari mikrokontroler ini mampu beroperasi dalam mode hemat daya atau mode *deep sleep*, modul ini dapat mengurangi konsumsi daya yang sangat signifikan dan dapat menghemat daya ketika memakai sumber daya baterai.

2.2.3 Sensor MQ-2

Sensor MQ-2 adalah sebuah sensor digital yang dapat mendeteksi berbagai jenis gas di udara, termasuk *Liquefied Petroleum Gas* (LPG), Alkohol, Asap, *propana*, *Hidrogen*, *Metana*, dan karbon monoksida. Sensor MQ-2 sensitif terhadap berbagai zat berbahaya, prinsip deteksinya bergantung pada perubahan resistansi bahan sensitif saat terkena gas. Ketika gas terpapar gas tertentu, resistensi internalnya berubah, dan perubahan ini diubah menjadi sinyal listrik yang dapat diukur. Meskipun sensor MQ-2 dapat mendeteksi banyak gas, sensor ini tidak dapat mengidentifikasi secara spesifik (Suryana, 2021). MQ-2 dapat di lihat di gambar 2.2.



Gambar 2. 2 Sensor MQ2

2.2.4 Sensor DHT22

Sensor DHT22 adalah alat yang digunakan untuk mengukur suhu dan kelembaban udara relatif. Sensor ini memiliki tingkat daya tahan yang tinggi, sehingga sangat cocok untuk digunakan dalam jangka waktu yang lama. Berbeda dengan DHT11, DHT22 menawarkan akurasi yang lebih tinggi, dengan tingkat ketelitian relative untuk suhu sekitar 4% dan kelembaban sekitar 18%. Sensor DHT22 mampu mengukur suhu udara dalam rentang -40 hingga 80 derajat Celsius dengan akurasi kurang dari 0,5 derajat Celsius (Andika,2024).



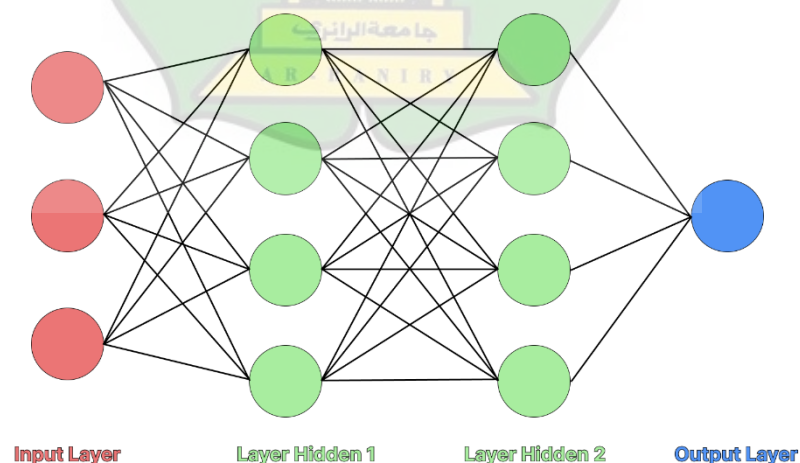
Gambar 2. 3 Sensor DHT22

Data dari sensor ini dikirim melalui protokol komunikasi satu kawat (single-wire), yang memudahkan integrasinya dengan mikrokontroler seperti ESP32. Dalam aplikasi IoT, DHT22 sering digunakan untuk memantau kondisi lingkungan.

secara real-time, seperti di ruangan server, rumah kaca, atau kandang ternak. Kemampuan sensor ini untuk memberikan pembacaan stabil dan konsisten membuatnya menjadi pilihan utama dalam berbagai sistem otomatisasi yang memerlukan pengendalian suhu dan kelembapan. Dengan kalibrasi pabrik yang presisi, DHT22 siap digunakan tanpa memerlukan penyesuaian tambahan, menjadikannya solusi yang efisien untuk berbagai kebutuhan teknologi dan lingkungan.

2.2.5 Deep Neural Network (DNN)

Algoritma DNN telah banyak diterapkan dalam berbagai aplikasi, seperti klasifikasi, pemrosesan bahasa, dan pemahaman citra. DNN berupaya meniru cara kerja jaringan saraf di otak manusia, sehingga sistem tersebut dapat memahami dan menginterpretasikan berbagai hal serta membuat keputusan yang mirip dengan manusia. Algoritma ini berfungsi untuk memetakan input ke output melalui serangkaian lapisan (Fajar & Nugraha, 2023). Dnn dapat di lihat pada gambar 2.4.



Gambar 2. 4 Deep Neural Network (DNN)

Pada arsitektur DNN terdapat tiga jenis lapisan utama: lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Setiap lapisan terdiri dari satu atau lebih unit pemrosesan yang dikenal sebagai *neuron* atau *node*. *Neuron* dalam lapisan masukan menerima data *input* dan mengirimkan ke lapisan tersembunyi. Lapisan tersembunyi memproses *input* tersebut lebih lanjut dan meneruskan ke lapisan berikut. Akhirnya, lapisan keluaran menghasilkan output akhir dari jaringan (Nur Anggraeni dkk., 2023). Jumlah *input* yang telah dibobotkan kemudian ditambahkan dengan bias. Perhitungan ini dapat diilustrasikan dalam bentuk fungsi seperti terlihat pada persamaan 1 dan 2:

$$dot_j = \sum_i^3 w_{ji}x_i + b_j \quad (1)$$

$$h_j = \sigma(dot_j) = \max(0, dot_j) \quad (2)$$

Dalam neural network, fungsi aktivasi (activation function) adalah fungsi non-linear yang digunakan dalam proses pelatihan neural network, dengan tujuan menentukan output dari neuron berdasarkan serangkaian input yang diberikan. Cara kerja fungsi aktivasi dalam neural network adalah sebagai berikut : input diterima oleh network melalui input layer. Di dalam neuron pada layer berikutnya, jumlah bobot dari input dihitung, biasanya ditambahkan dengan bias. Hasil penjumlahan ini kemudian dilewatkan melalui fungsi aktivasi (Akil, 2023).

Dalam pengembangan sistem peringatan dini kebocoran gas menggunakan DNN berbasis IoT, pemilihan jenis fungsi aktivasi yang tepat sangat penting untuk memastikan performa optimal dari model. Jenis fungsi aktivasi yang cocok untuk penelitian ini adalah jenis *Rectified Linear Unit* (ReLU).

ReLU adalah fungsi aktivasi yang populer digunakan dalam neural network, terutama pada jaringan yang dalam (deep learning). Fungsi ini didefinisikan secara matematis. ReLU sering digunakan di hidden layers dari jaringan neural karena sifatnya yang efisien dan kemampuannya dalam membantu jaringan untuk belajar representasi yang lebih dalam dan kompleks dari data, $ReLU(x) = \max(0, x)$.

2.2.6 Sistem Peringatan Dini Berbasis IoT

Sistem peringatan dini adalah serangkaian sistem komunikasi yang menghubungkan informasi dan sensor untuk mendeteksi kejadian dan mengambil keputusan secara otomatis. Semua tahapan ini bekerja bersama-sama untuk memprediksi gangguan yang dapat berdampak negatif terhadap stabilitas fisik dunia. Sistem ini memberikan waktu bagi entitas terkait untuk merespons dan mempersiapkan diri menghadapi kejadian buruk yang akan segera terjadi, dengan tujuan meminimalkan dampaknya. Ketika situasi kritis terjadi, dengan tujuan meminimalkan dampak. Ketika situasi kritis terjadi, peringatan dini biasanya disampaikan melalui berbagai metode seperti sirine, kentongan, prediksi, dan lain sebagainya (Satria, 2023).

2.2.7 TensorFlow

Tensorflow adalah sebuah *library* yang dibuat untuk membangun model *machine learning*. *Library* ini menyediakan *tools* yang mendukung pembuatan model pada berbagai tingkat kesulitan yang di gunakan untuk melihat atau memprogram suatu model. *TensorFlow* memungkinkan penggunaan API (Application Programming Interface) tingkat rendah untuk membangun model dengan mendefinisikan operasi matematis secara langsung, atau API tingkat tinggi untuk menerapkan arsitektur yang sudah ada, seperti regresi *linier* dan *neural network*.

Pendekatan ini dapat dianalogikan dengan cara kerja kode Python. Secara sistematis Python yang berjalan di berbagai hardware, *TensorFlow* juga dapat menjalankan grafik komputasinya di platform seperti CPU, GPU, maupun TPU.

API tingkat tinggi menawarkan kemudahan penggunaan namun kurang fleksibel dibanding API tingkat rendah. Disarankan untuk memulai dengan API tingkat tinggi untuk memastikan fungsionalitas dasar berjalan dengan baik. Setiap lapisan API dalam *TensorFlow* dibangun di atas API yang lebih mendasar, sehingga transisi antar level tetap intuitif dan mudah diimplementasikan. (Yusuf, 2020)

2.2.8 Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer dan serbaguna. Dikembangkan pertama kali oleh Guido van Rossum pada awal 1990-an, Python kini menjadi salah satu pilihan utama di kalangan pengembang perangkat lunak di seluruh dunia.

Keunggulan utama Python terletak pada kemudahan penggunaannya serta sintak yang sederhana dan mudah dipahami, membuatnya ideal bagi pemula maupun pengembang berpengalaman. Python juga bersifat lintas *platform*, sehingga dapat dijalankan di berbagai sistem operasi seperti *Windows*, *macOS*, dan *Linux*. Bahasa ini digunakan dalam berbagai bidang, mulai dari pengembangan web dan aplikasi hingga kecerdasan buatan.

Ciri khusus Python adalah filosofinya yang menekankan keterbacaan kode, memudahkan kolaborasi dan pemeliharaan proyek perangkat lunak. Python memiliki pustaka standar yang sangat lengkap dan kaya, menyediakan berbagai modul dan fungsi bawaan yang mempermudah proses pengembangan aplikasi. Dengan pustaka standar ini, pengembang dapat menghemat waktu dan usaha dalam mengimplementasikan fungsionalitas umum yang sering digunakan dalam pengembangan perangkat lunak.

Selain itu, Python dikenal sebagai bahasa pemrograman yang mendukung berbagai paradigma pemrograman. Selain pemrograman berorientasi objek, Python juga mendukung pemrograman imperatif dan fungsional, menjadikannya bahasa yang sangat fleksibel dan cocok digunakan untuk beragam jenis proyek pengembangan perangkat lunak. (Santos dkk., 2024)

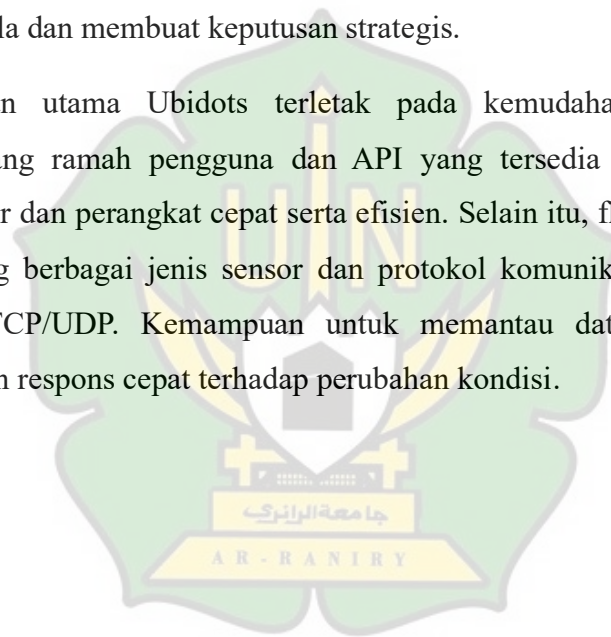
2.2.9 Ubidots

Ubidots adalah platform Internet of Things (IoT) berbasis cloud yang berasal dari Boston, Amerika Serikat. IoT sendiri mengacu pada jaringan perangkat elektronik yang saling terhubung melalui internet untuk bertukar data dan informasi secara otomatis. Ubidots dirancang untuk mempermudah pengguna dalam

mengumpulkan data dari sensor, mengolahnya menjadi informasi yang bermakna, dan mendukung pengambilan keputusan berbasis data.(Teguh dkk., 2018)

Salah satu fungsi utama Ubidots adalah pengumpulan data dari sensor. Platform ini memudahkan integrasi perangkat sehingga data dapat dikirimkan secara real-time ke cloud. Data yang terkumpul kemudian dapat divisualisasikan dalam bentuk grafik, tabel, atau indikator interaktif melalui dashboard yang intuitif. Selain itu, Ubidots memungkinkan pengguna mengatur aksi otomatis, seperti menghidupkan perangkat atau mengirimkan peringatan berdasarkan data yang diterima. Fitur analisis dan prediksi juga menjadi keunggulan, membantu pengguna mendeteksi pola dan membuat keputusan strategis.

Kelebihan utama Ubidots terletak pada kemudahan penggunaannya. Antarmuka yang ramah pengguna dan API yang tersedia menjadikan proses integrasi sensor dan perangkat cepat serta efisien. Selain itu, fleksibilitas platform ini mendukung berbagai jenis sensor dan protokol komunikasi, seperti MQTT, HTTP, dan TCP/UDP. Kemampuan untuk memantau data secara real-time memungkinkan respons cepat terhadap perubahan kondisi.



BAB III METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan dari bulan Agustus 2024 hingga Desember 2024 di Gampong Jawa, Kecamatan Kuta Raja, Kota Banda Aceh, Aceh.

3.2 Alat dan Bahan

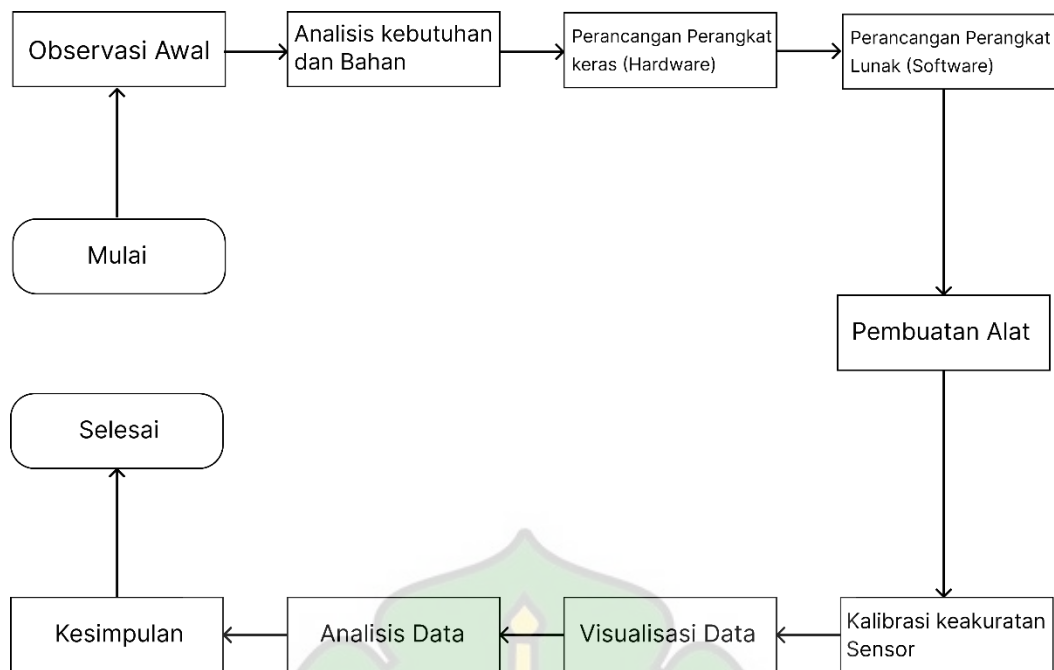
Alat dan bahan yang digunakan pada penelitian ini adalah dapat dilihat pada Tabel 3.1:

Tabel 3. 1 Alat dan Bahan

NO	KEBUTUHAN
1	Hardware: 1. <i>ESP32 Dev Kit V1</i> 2. Sensor MQ2 3. Sensor DHT22 4. Kabel Jumper <i>Male to Female</i> 5. Kable Jumper <i>Male to Male</i> 6. <i>Breadboard Project</i> 7. <i>Box Project</i>
2	Software: 1. <i>Ubidots</i> 2. <i>Arduino IDE</i> 3. <i>TensorFlow</i>

3.3 Alur Penelitian

Alur pada penelitian ini dituangkan pada gambar 3.1



Gambar 3. 1 Alur Penelitian

3.4 Tahap Penelitian

Pada tahapan penelitian ini penulis menggunakan *metode R&D (Research and Development)*. *Metode R&D* adalah metode penelitian dan pengembangan yang bertujuan untuk menciptakan produk baru, memecahkan masalah, atau memperbaiki proses. Rancangan penelitian ini meliputi beberapa tahap, dimulai dari observasi awal, analisis kebutuhan peralatan dan bahan, perancangan *hardware* dan *software*, pembuatan prototipe, analisis data, dan penarikan kesimpulan. Proses penelitian ini dijelaskan sebagai berikut:

3.5 Observasi Awal

Pada tahap observasi awal penulis melakukan penelitian literatur dengan mengumpulkan dataset yang mencakup indikator yang di proses oleh sistem.

3.6 Analisa Kebutuhan Alat dan Bahan

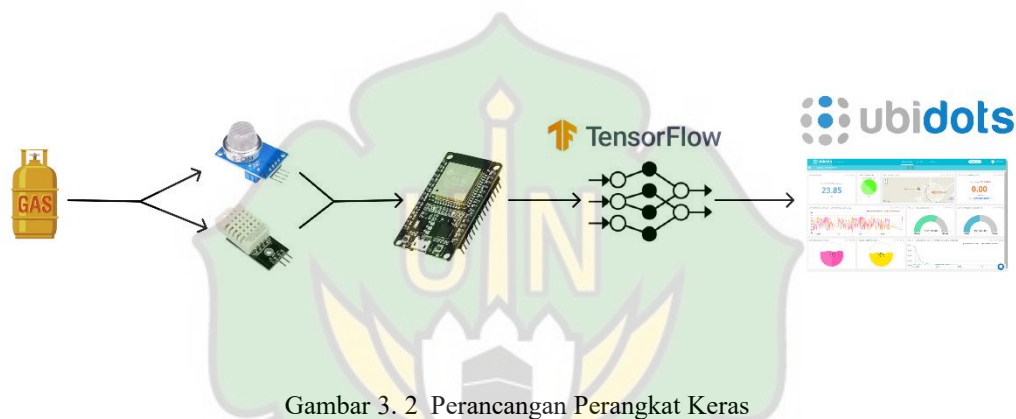
Alat dan bahan dibutuhkan dalam penelitian ini oleh karena itu menganalisa alat dan bahan sangat penting mengingat beberapa alat yang harus disesuaikan dalam fungsinya, hal ini dapat mempermudah penulis dalam penelitian.

3.7 Perancangan sistem

Adapun sistem yang mendukung dalam penelitian ini adalah sebagai berikut:

3.7.1 Perangkat Keras (*Hardware*)

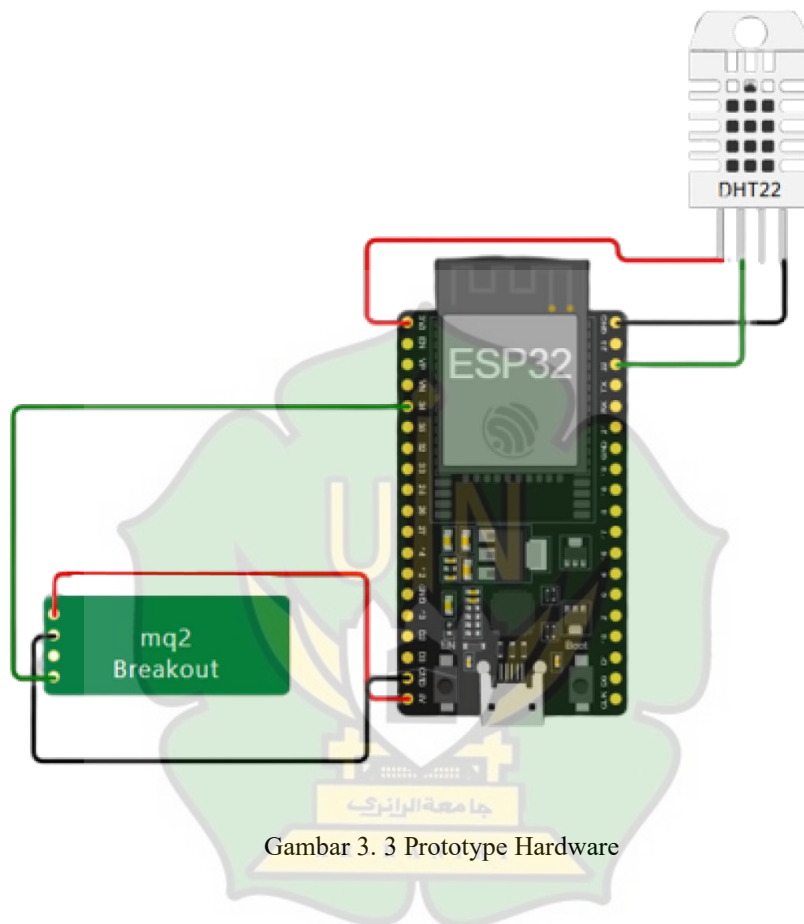
Perancangan alat ini mencakup keseluruhan aspek perangkat keras (*Hardware*) dan perangkat lunak (*software*). Pada bagian perangkat keras, dijelaskan komponen-komponen yang digunakan dalam sistem. Sementara itu, bagian perangkat lunak berfokus pada perancangan program yang digunakan dalam sistem pendeteksi kebocoran gas, perancangan perangkat keras dapat dilihat pada gambar 3.2.



Gambar 3. 2 Perancangan Perangkat Keras

Dalam perancangan perangkat keras (*Hardware*), terdapat beberapa yang harus dilakukan. Pertama, sensor MQ2 dan sensor DHT22 yang terhubung melalui kabel jumper akan mengirimkan data deteksi ke mikrokontroler Esp 32 Dev kit V1. Pada mikrokontroler, data yang diperoleh dari sensor MQ2 dan DHT22 akan diproses untuk normalisasi data.. Model DNN yang sudah dilatih ini perlu dikoversi menggunakan *TensorFlow* agar kompatibel dengan mikrokontroler. Setelah itu, *TesnsorFlow* menggunakan model DNN untuk melakukan prediksi berdasarkan data input dari sensor. Python menjalankan model DNN dengan data input tersebut untuk menghasilkan output prediksi. Output yang dihasilkan akan mengirim hasil prediksi ke *device ubidots*.

Dalam penelitian pembuatan *hardware* mencakup mikrokontroler ESP32, sensor gas MQ2, dan sensor suhu DHT22 dengan dihubungkan oleh kabel jumper. Pada gambar 3.3, berikut prototype dari hardware tersebut.



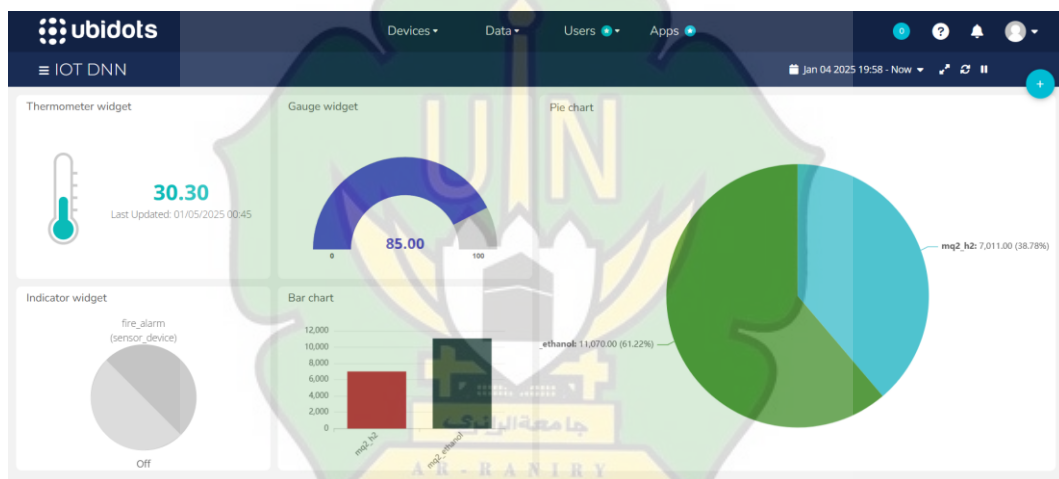
Gambar 3. 3 Prototype Hardware

Tabel 3. 2 Rincian PIN

NO	ESP32	SENSOR MQ2	SENSOR DHT22
1	VIN/3V3	VCC	VCC
2	GND	GND	GND (-)
3	PIN 34	AO	
4	PIN 22		OUT (+)

3.7.2 Perangkat Lunak (*Software*)

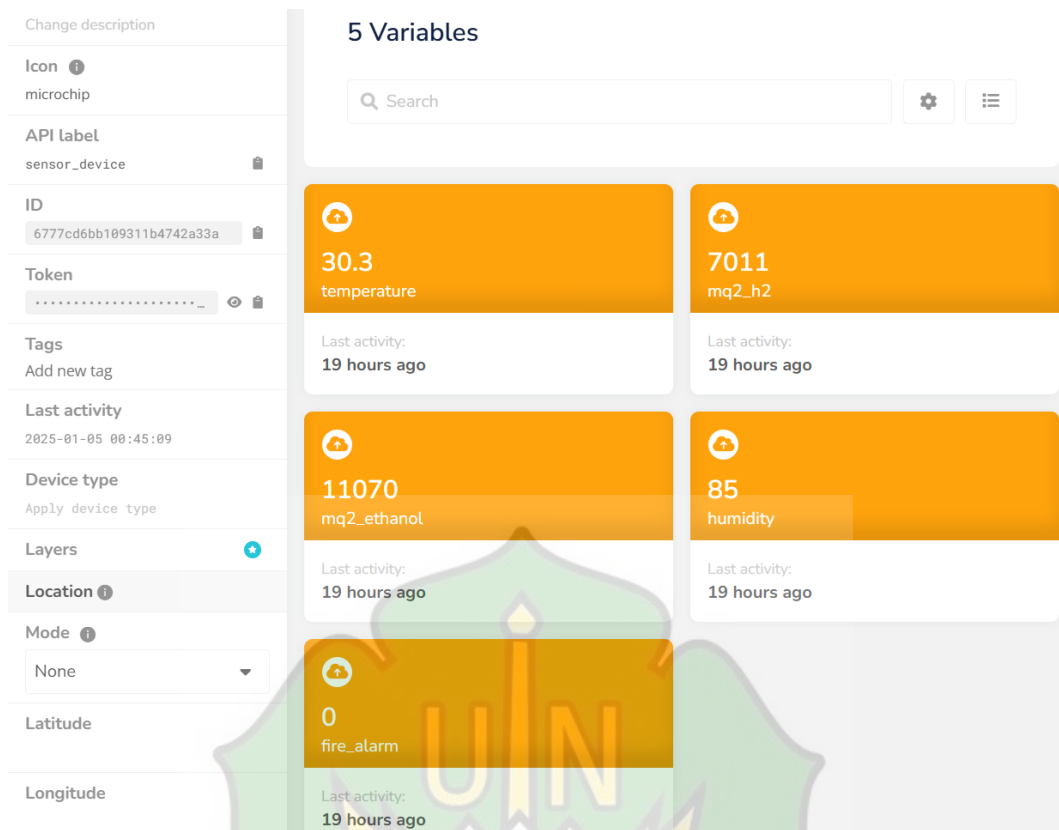
Dalam pengembangan perangkat lunak sistem prediksi sebagai output setelah melalui proses normalisasi data pada *tensorflow*. Sistem prediksi ini akan menggunakan data yang dikumpulkan oleh *mikrokontroler ESP32 Dev Kit V1* dan di simpan di *platform ubidots*. Data tersebut akan dianalisis menggunakan *TensorFlow* atau modul prediksi tertentu, seperti DNN. Data yang didapatkan dari dataset dan sensor akan menjadi input untuk sistem prediksi yang akan diimplementasikan. Dataset ini berisi berbagai atribut atau fitur yang relevan dengan masalah yang ingin diprediksi, perancangan perangkat lunak dapat dilihat pada gambar 3.3.



Gambar 3. 4 Perancangan Perangkat Lunak

3.7.3 Device Ubidots

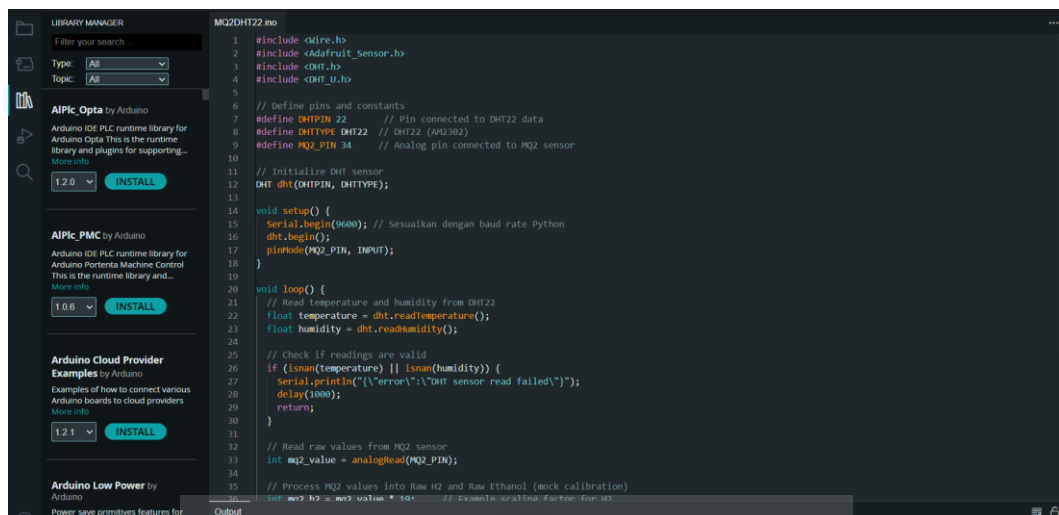
Pembuatan *dashboard* di *ubidots* melibatkan langkah-langkah mulai dari pendaftaran dan pembuatan akun di *platform ubidots*, pengaturan perangkat dan variabel untuk menyimpan data sensor, hingga pengiriman dengan data secara *real-time* menggunakan API yang disediakan. Setelah data terkirim, langkah selanjutnya adalah membuat *dashboard* baru di *ubidots* dan menyesuaikan tampilan dengan menambahkan widget grafik garis, batang, angka, dan indikator lainnya sesuai dengan jenis data yang diukur. *Devices ubidot* dapat dilihat pada gambar 3.4.



Gambar 3. 5 Devices ubidost

3.7.4 Integritas Ubidots dengan Arduino IDE

Menghubungkan Arduino IDE dengan Ubidots dan mengirimkan data sensor, langkah pertama adalah membuat akun ubidost dan membuat perangkat serta variable di *dashboard platform* tersebut. Selanjutnya dalam Arduino IDE, instal library Ubidots yang tersedia melalui menu *manager libraries*, dan tambahkan kode Arduino yang sesuai. Arduino IDE dapat dilihat pada gambar 3.5.



Gambar 3. 6 Arduino IDE

3.8 Kalibrasi Sensor

Kalibrasi sensor adalah proses penyesuaian sensor agar hasil pengukurannya akurat dan sesuai dengan nilai standar atau referensi yang telah diketahui. Proses ini penting untuk memastikan bahwa sensor memberikan output yang tepat ketika digunakan dalam penelitian ini.

Tujuan utama kalibrasi adalah meningkatkan akurasi pengukuran sensor. Dengan kalibrasi yang baik, sensor mampu mencatat nilai yang sesuai dengan dataset. Selain itu, kalibrasi membantu memperbaiki drift, yaitu perubahan kinerja sensor yang terjadi seiring waktu atau akibat paparan lingkungan ekstrem. Proses ini juga memastikan bahwa sensor memberikan hasil yang konsisten dan memenuhi spesifikasi standar yang dibutuhkan dalam pengolahan dataset.

Kalibrasi biasanya dimulai dengan persiapan sensor, di mana perangkat dibersihkan dan disiapkan agar bebas dari faktor eksternal yang dapat mempengaruhi pengukuran. Selanjutnya, sensor dioperasikan dalam kondisi dengan nilai acuan yang diketahui, seperti konsentrasi gas atau suhu standar. Jika output sensor menyimpang dari nilai acuan, parameter sensor dikoreksi atau

perangkat lunak pemrosesan sinyalnya diperbarui. Setelah itu, dilakukan verifikasi untuk memastikan sensor telah berfungsi dengan akurasi yang sesuai.

```

//( calibration)
int mq2_h2 = mq2_value * 50;
int mq2_ethanol = mq2_value * 80;

```

Gambar 3. 7 kodingan kalibrasi sensor

Frekuensi kalibrasi bergantung pada jenis sensor, aplikasi, sensor yang digunakan untuk mendeteksi gas beracun di lingkungan berbahaya memerlukan kalibrasi lebih sering dibandingkan sensor suhu yang digunakan di lingkungan stabil. Pemeliharaan berkala dapat membantu menjaga kinerja sensor tetap optimal.

Untuk sensor MQ2, kalibrasi dilakukan dengan memaparkan sensor ke konsentrasi gas yang diketahui dan menyesuaikan output sensor agar sesuai dengan konsentrasi tersebut. Sedangkan sensor DHT22, yang digunakan untuk mengukur suhu dan kelembapan, biasanya telah dikalibrasi dari pabrik. Namun, dalam beberapa kondisi khusus, kalibrasi tambahan dapat dilakukan di laboratorium menggunakan perangkat kalibrasi standar.

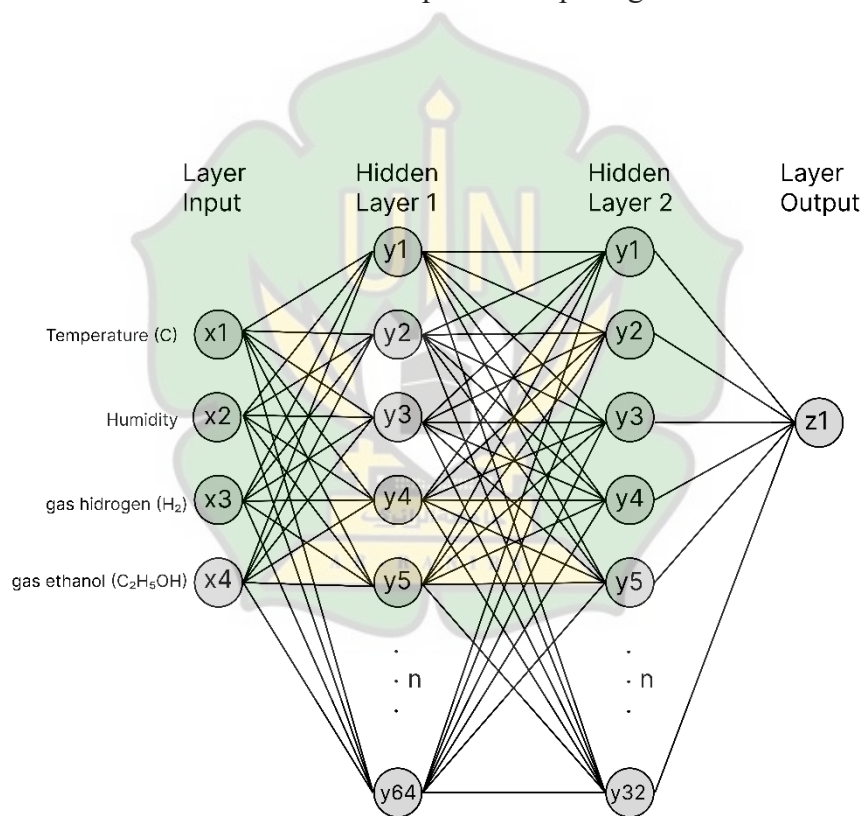
Kalibrasi yang dilakukan secara rutin dan tepat dapat meningkatkan akurasi dan keandalan sensor, sehingga data yang dihasilkan dapat mendukung pengambilan keputusan yang lebih baik dalam sistem pemantauan.

3.9 Arsitektur DNN

Arsitektur *Deep Neural Networks* (DNN) adalah tipe jaringan saraf tiruan yang terjadi dari beberapa *Hidden layer* antara *input* dan *output*, yang digunakan dalam aplikasi kecerdasan buatan modern seperti pengenalan gambar dan pemrosesan Bahasa alami. Arsitektur DNN mencakup lapisan *input* yang menerima data

mentah, beberapa lapisan tersembunyi yang memproses data melalui fungsi aktivitas seperti ReLU, dan lapisan *output* yang menghasilkan prediksi dengan fungsi aktifitas seperti *softmax*.

Pada penelitian ini penulis menggunakan arsitektur DNN dengan fungsi aktivasi ReLU yang terjadi dari 4 neuron pada *input layer*, 64 neuron pada *hidden layer* pertama, 32 neuron pada *hidden layer* kedua, dan 1 neuron pada output layer. Fungsi aktivasi ReLU, yang didefinisikan sebagai $f(x) = \max(0, x)$, membantu mengatasi masalah vanishing gradient dan meningkatkan efisiensi komputasi. Dalam arsitektur ini. Arsitektur dnn dapat dilihat pada gambar 3.6.



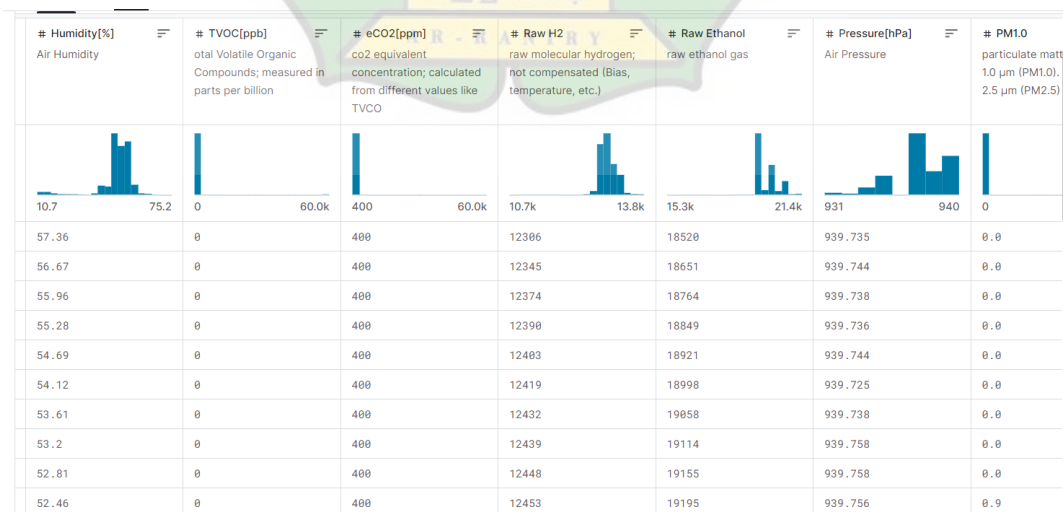
Gambar 3. 8 Arsitektur Deep Neural Network layer

Tabel 3. 3 Parameter Model

Arsitektur	Kompilasi	Pelatihan
3 layer dense, dengan 64, 32, dan 1 neuron.	Optimizer: Adam.	Epoch: 10
Fungsi aktivasi ReLU sebagai hidden layer dan sigmoid sebagai output layer.	Loss: Binary crossentropy (untuk klasifikasi biner).	Batch size:4
	Metrik: Akurasi.	Data validasi: 20% dari data pelatihan.

3.10 Pengumpulan data

Dataset yang digunakan di ambil dari situs *Kaggle*. Terdiri dari data, *Temperature (C)*, *Humidity*, *Raw H2*, *Raw Ethanol*. Jumlah dataset yang digunakan sebanyak 62.630 data. <https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset>.



Gambar 3. 9 Tampilan Dataset

3.11 Rincian dataset

Adapun dataset yang digunakan memiliki parameter, penjelasan, dan kegunaan yang mengukur potensi adanya gas. Berikut rincian dari dataset dapat dilihat pada tabel III. 5.

Tabel 3. 4 Rincian data

Parameter	Penjelasan	Kegunaan
<i>Temperature (C)</i>	Suhu diukur dalam derajat <i>Celcius</i> ($^{\circ}\text{C}$)	Menunjukkan tingkat panas atau dingin dari suatu lingkungan atau objek.
<i>Humidity</i>	Kelembaban adalah jumlah Uap air yang ada di udara. Dinyatakan dalam persentase (%)	Mengukur kelembaban dapat mendeteksi kebocoran gas di ruangan
<i>RAW Hidrogen (H2)</i>	Konsentrasi Hidrogen Mentah di udara.	Mengidentifikasi kebocoran Etanol serta untuk memantau kualitas udara dan deteksi kebocoran gas
<i>RAW Ethanol</i>	Senyawa berbentuk cair, bening, mudah menguap, dan mudah terbakar	Mengukur konsentrasi ethanol dengan akurasi tinggi menggunakan gas sebagai media pemisah

3.12 Normalisasi Data

Normalisasi data sangat berpengaruh dalam pembuatan model, normalisasi dalam penelitian ini menggunakan metode Min-Max Scaler. Ada beberapa tahapan normalisasi data, yaitu sebagai berikut:

3.12.1 Pemisahan kolom

pemisahan kolom ini sangat berpengaruh dalam menghasilkan data yang akurat dikarenakan pada proses training data akan terjadi error jika tidak dipisahkan dengan format yang sesuai.

```
columns = ["Temperature", "Humidity", "Raw H2", "Raw Ethanol", "Fire Alarm"]
try:
    data_split = data.iloc[:, 0].str.split(';', expand=True)
    if data_split.shape[1] != len(columns):
        raise ValueError(f"Expected {len(columns)} columns, but got {data_split.shape[1]} columns.")
    data_split.columns = columns # Assign proper column names
    data = data_split
except Exception as e:
    raise ValueError(f"Error splitting columns: {e}")

# Convert data types to numeric and handle errors
data = data.apply(pd.to_numeric, errors='coerce')

# Drop rows with NaN values
data = data.dropna()

# Separate features and target variable (using all 4 features)
X = data[["Temperature", "Humidity", "Raw H2", "Raw Ethanol"]]
y = data["Fire Alarm"]
```

Gambar 3. 10 kodingan pemisah kolom

Tabel 3. 5 Parameter Kolom

NO	PARAMETER
1	<i>Temperature</i>
2	<i>Humidity</i>
3	<i>Raw H2</i>
4	<i>Raw Ethanol</i>
5	<i>dan Fire Alarm</i>

Pada tabel tersebut merepresentasikan data yang diambil dari sensor, di mana masing-masing kolom menyimpan informasi terkait suhu, kelembapan, kadar gas hidrogen (H2) dan etanol, serta status alarm kebakaran.

Langkah selanjutnya adalah memproses data yang terdapat pada kolom pertama, yang mana berisi data dalam format string yang dipisahkan dengan tanda titik koma (;). Fungsi *split* digunakan untuk membagi data dalam kolom tersebut menjadi beberapa kolom terpisah berdasarkan pemisah tersebut. Jika jumlah kolom yang dihasilkan tidak sesuai dengan jumlah kolom yang didefinisikan sebelumnya (5 kolom), maka akan muncul error yang memberitahukan adanya ketidaksesuaian. Setelah pemisahan ini, nama kolom yang telah didefinisikan sebelumnya diberikan pada *DataFrame* yang telah dipisah tersebut.

Setelah data terpisah dengan benar, langkah berikutnya adalah memastikan bahwa semua nilai dalam *Data Frame* dapat diproses sebagai angka. Fungsi *apply* diterapkan pada seluruh *DataFrame* untuk mengkonversi nilai-nilai dalam setiap kolom menjadi tipe data numerik. Jika ada nilai yang tidak bisa dikonversi menjadi angka, nilai tersebut akan digantikan. Ini penting untuk memastikan bahwa data yang digunakan dalam analisis atau pelatihan model tidak mengandung tipe data yang tidak sesuai.

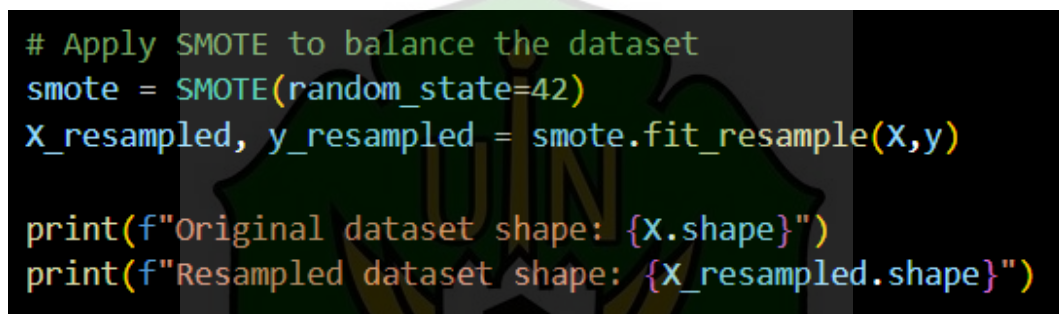
Data dipisahkan menjadi dua bagian: fitur (X) dan variabel target (y). Fitur terdiri dari kolom-kolom yang berisi informasi sensor, yaitu *Temperature*, *Humidity*, *Raw H2*, dan *Raw Ethanol*, yang digunakan untuk memprediksi variabel target, yaitu *Fire Alarm*. *Fire Alarm* adalah variabel yang menunjukkan apakah alarm kebakaran terpicu berdasarkan input sensor yang diberikan. Pembagian ini penting karena fitur akan digunakan untuk mempelajari pola dalam data, sedangkan target digunakan sebagai output yang ingin diprediksi dalam model pembelajaran.

3.12.2 Metode *Synthetic Minority Over-sampling* (SMOTE)

SMOTE (*Synthetic Minority Oversampling Technique*) adalah metode yang digunakan untuk mengatasi ketidakseimbangan data dalam pembelajaran mesin. Ketidakseimbangan data terjadi ketika jumlah sampel di satu kelas jauh lebih banyak dibandingkan kelas lainnya. Hal ini dapat menyebabkan bias pada model yang cenderung memprediksi kelas dengan sampel terbanyak. SMOTE membantu

dengan membuat sampel sintetis untuk kelas yang lebih sedikit (kelas minoritas) sehingga distribusi data menjadi lebih seimbang.

SMOTE bekerja dengan memilih sampel dari kelas minoritas, kemudian menentukan tetangga terdekatnya berdasarkan jarak, biasanya menggunakan metrik jarak Euclidean. Setelah tetangga terdekat ditemukan, SMOTE menciptakan data baru dengan interpolasi antara sampel asli dan tetangganya. Data sintetis yang dihasilkan berada di sepanjang garis antara dua titik yang dipilih, yang membantu memperluas representasi kelas minoritas di ruang data.



```
# Apply SMOTE to balance the dataset
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X,y)

print(f"Original dataset shape: {X.shape}")
print(f"Resampled dataset shape: {X_resampled.shape}")
```

Gambar 3. 11 kodingan metode smote

Salah satu keuntungan utama SMOTE adalah kemampuannya untuk mengurangi masalah overfitting yang sering terjadi saat menggunakan metode oversampling tradisional seperti duplikasi data minoritas. Dengan menyeimbangkan distribusi kelas, SMOTE dapat meningkatkan kinerja model dalam memprediksi sampel dari kelas minoritas. Namun, metode ini juga memiliki tantangan, termasuk risiko menciptakan data yang tidak sepenuhnya realistis dan kurang efektif dalam kasus di mana kelas minoritas memiliki distribusi yang kompleks atau tumpang tindih dengan kelas mayoritas.

Dalam proyek pembelajaran mesin seperti prediksi penyakit, deteksi penipuan, atau klasifikasi anomali, SMOTE sering digunakan untuk memastikan

bahwa model dapat mengenali pola dari semua kelas dengan baik. Penggunaannya dapat diterapkan dengan mudah menggunakan pustaka seperti imblearn di Python.

3.12.3 Data Training

Data training adalah kumpulan data yang digunakan untuk melatih model agar dapat mengenali pola atau hubungan fitur dan label. Model belajar dari dataset dengan menyesuaikan parameter internalnya untuk meminimalkan kesalahan prediksi.

Tabel 3. 6 Data Training

Temperatur	Humidity	Raw H2	Raw Ethanol	Fire Alaram
20.0	57.36	12306	18520	0
20.015	55.67	12345	18651	0
1.251	53.19	12944	19385	1
1.235	53.32	12946	19376	1

3.12.4 Data Testing

Data testing adalah kumpulan data yang di gunakan untuk mengevaluasi performa model setelah pelatihan selesai. Dataset yang tidak pernah digunakan selama proses pelatihan, sehingga memberikan objektif terhadap model dalam memprediksi data baru.

Tabel 3. 7 Data Testing

Temperatur	Humidity	Raw H2	Raw Ethanol	Fire Alaram
20.029	55.96	12374	18764	0
20.044	55.28	12390	18849	0
1.201	53.41	12947	19387	1

Temperatur	Humidity	Raw H2	Raw Ethanol	Fire Alaram
1.185	53.48	12956	19393	1

3.12.5 Data Validasi

Data validasi adalah kumpulan data yang digunakan selama proses pelatihan untuk mengatur parameter model dan mencegah *overfitting*. Data validasi ini memantau performa model di setiap iterasi pelatihan sehingga dapat digunakan untuk menentukan kapan pelatihan harus dihentikan atau untuk memilih parameter terbaik.

Tabel 3. 8 Data validasi

Temperatur	Humidity	Raw H2	Raw Ethanol	Fire Alaram
20.059	54.69	12403	18921	0
20.073	54.12	12419	18998	0
1.152	53.60	12951	19394	1
1.135	5368	12951	19388	1

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengujian Sensor MQ2 dan DHT22

Pada penelitian ini pengujian alat sangat di butuhkan mengingat tergantungannya data yang di kirim oleh sensor ke model DNN sangat berpengaruh dalam menentukan hasil prediksi nantinya. Berikut langkah – langkah pengujianya:

4.1.1. Hasil Output Pendeteksian Sensor MQ2 dan DHT22

Sensor MQ2 digunakan untuk mendeteksi keberadaan gas seperti h₂ dan ethanol yang sesuai dengan dataset, dengan mengukur perubahan resistansi dengan terdeteksinya gas. Sensor DHT22 berfungsi untuk mengukur parameter lingkungan seperti suhu dan kelembapan udara dengan presisi tinggi.

Hasil output dari kedua sensor tersebut umumnya berupa nilai digital yang mencerminkan kondisi lingkungan dalam bentuk konsentrasi gas dalam *Parts per Million* (PPM) untuk sensor MQ2 dan data suhu serta kelembapan dalam derajat celsius dan persentase untuk sensor DHT22.

Pada pengujian kali ini peneliti menggunakan gas pemantik api sebagai media penelitian dengan mengarahkan gas tersebut ke sensor, pada lampu yang berwarna hijau adalah sebagai patokan jika ada gas yang terdeteksi oleh sensor.

Tabel 3. 9 Output Pengujian Sensor

04:38:36.206	Temperature":30.30	Humidity":83.50	Raw H2":35350	Raw Ethanol":56560
04:38:38.170	Temperature":30.20	Humidity":83.30	Raw H2":28050	Raw Ethanol":44880
04:38:40.186	Temperature":30.30	Humidity":83.30	Raw H2":24100	Raw Ethanol":38560
04:38:42.184	Temperature":30.30	Humidity":83.30	Raw H2":21450	Raw Ethanol":34320
04:38:44.222	Temperature":30.30	Humidity":83.20	Raw H2":19350	Raw Ethanol":30960
04:38:46.211	Temperature":30.30	Humidity":83.30	Raw H2":18100	Raw Ethanol":28960
04:38:48.204	Temperature":30.30	Humidity":83.30	Raw H2":16750	Raw Ethanol":26800
04:38:50.205	Temperature":30.30	Humidity":83.40	Raw H2":15750	Raw Ethanol":25200
04:38:52.200	Temperature":30.30	Humidity":83.40	Raw H2":16050	Raw Ethanol":25680
04:38:54.216	Temperature":30.30	Humidity":83.30	Raw H2":14700	Raw Ethanol":23520

04:38:56.243	Temperature":30.30	Humidity":83.30	Raw H2":14300	Raw Ethanol":22880
04:38:58.248	Temperature":30.30	Humidity":83.30	Raw H2":14900	Raw Ethanol":23840
04:39:00.238	Temperature":30.30	Humidity":83.30	Raw H2":13750	Raw Ethanol":22000
04:39:02.230	Temperature":30.30	Humidity":83.30	Raw H2":13800	Raw Ethanol":22080
04:39:04.235	Temperature":30.30	Humidity":83.20	Raw H2":14650	Raw Ethanol":23440
04:39:06.243	Temperature":30.30	Humidity":83.20	Raw H2":13750	Raw Ethanol":22000
04:39:08.253	Temperature":30.30	Humidity":83.10	Raw H2":12800	Raw Ethanol":20480
04:39:10.259	Temperature":30.30	Humidity":83.00	Raw H2":12300	Raw Ethanol":19680
04:39:12.264	Temperature":30.30	Humidity":82.90	Raw H2":13250	Raw Ethanol":21200
04:39:14.293	Temperature":30.30	Humidity":82.90	Raw H2":13800	Raw Ethanol":20000
04:39:16.282	Temperature":30.30	Humidity":82.90	Raw H2":12600	Raw Ethanol":20160

Pada output pengujian sensor dapat dilihat pada “Raw H2”: 13750 dan “Raw Ethanol”: 22000, yang berarti adanya gas terdeteksi, dengan landasan logika informatika, jika lebih besar dari nilai raw h2 (12419ppm), dan raw ethanol (18998ppm), maka nilai yang di keluarkan 1 atau sama dengan gas terdeteksi.

4.1.2. Hasil Pengiriman Data ke Python

Pada tahap ini sensor akan mengirimkan data ke python yang mana nantinya, python yang akan menjalankan tensorflow dan memproses data mentah menjadi data proses untuk dijadikan hasil prediksi.

```
Data berhasil dikirim ke Ubidots pada 2025-01-12 03:30:21
Data mentah: {"Temperature":30.20,"Humidity":81.40,"Raw H2":12500,"Raw Ethanol":20000}
Data diterima: {'Temperature': 30.2, 'Humidity': 81.4, 'Raw H2': 12500, 'Raw Ethanol': 20000}
esp: [[ 30.2 81.4 12500. 20000. ]]
```

Gambar 4. 1 Hasil Pengiriman Data Sensor

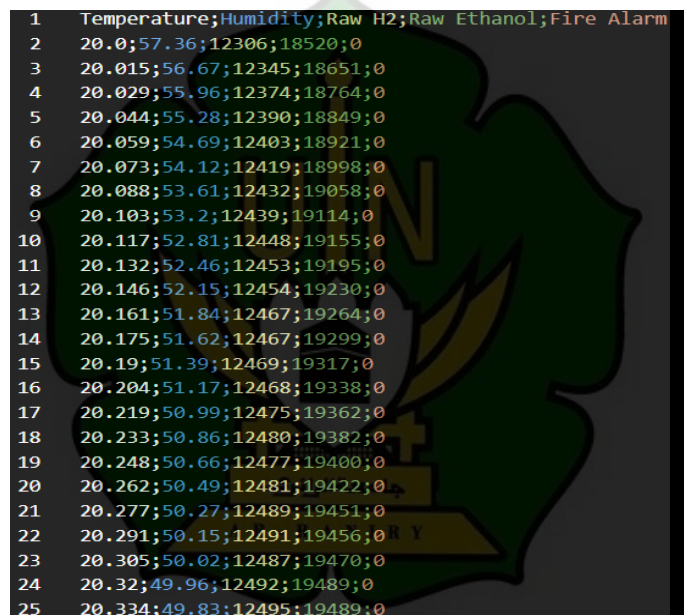
Data mentah yang dikirim ke Ubidots berupa format JSON dengan nilai: suhu 30,20°C, kelembaban 81,40%, konsentrasi gas Hidrogen sebesar 12.500 satuan mentah, dan konsentrasi gas Ethanol sebesar 20.000 satuan mentah. Setelah data diterima oleh python, tensorflow tersebut memformat ulang data ke dalam struktur dictionary Python yang tetap mempertahankan nilai dan akurasi informasi yang dikirim.

Di sisi perangkat ESP32, data tersebut disusun dalam bentuk array matriks dengan format satu baris empat kolom, yaitu {30.2, 81.4, 12500, 20000}. Format

ini memudahkan perangkat dalam proses analisis atau mengolah data lebih lanjut, misalnya untuk penerapan pembelajaran mesin atau pengambilan keputusan otomatis berbasis data sensor.

4.2 Hasil Normalisasi Data

Normalisasi data adalah proses mengubah nilai dari dataset ke skala yang sama untuk memastikan setiap parameter memiliki data yang seimbang agar tidak mendominasi pelatihan model. Dalam penelitian ini normalisasi data yang di pakai dengan metode *min-max scaling* , metode ini dipakai dikarenakan untuk mencari nilai data 1/0.



	Temperature	Humidity	Raw H2	Raw Ethanol	Fire Alarm
1	20.0	57.36	12306	18520	0
2	20.015	56.67	12345	18651	0
3	20.029	55.96	12374	18764	0
4	20.044	55.28	12390	18849	0
5	20.059	54.69	12403	18921	0
6	20.073	54.12	12419	18998	0
7	20.088	53.61	12432	19058	0
8	20.103	53.2	12439	19114	0
9	20.117	52.81	12448	19155	0
10	20.132	52.46	12453	19195	0
11	20.146	52.15	12454	19230	0
12	20.161	51.84	12467	19264	0
13	20.175	51.62	12467	19299	0
14	20.19	51.39	12469	19317	0
15	20.204	51.17	12468	19338	0
16	20.219	50.99	12475	19362	0
17	20.233	50.86	12480	19382	0
18	20.248	50.66	12477	19400	0
19	20.262	50.49	12481	19422	0
20	20.277	50.27	12489	19451	0
21	20.291	50.15	12491	19456	0
22	20.305	50.02	12487	19470	0
23	20.32	49.96	12492	19489	0
24	20.334	49.83	12495	19489	0
25					

Gambar 4. 2 Dataset Sebelum Normalisasi

Normalisasi data ini menggunakan metode min-max scaler yang dimana persamaan nilai-nilai data ke dalam rentang tertentu dengan nilai 1/0, dapat di lihat pada persamaan 3:

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \quad (3)$$

Penjelasan :

- X adalah nilai data asli.
- Xmin adalah nilai minimum dalam data
- Xmax adalah nilai maksimum dalam data.
- Hasilnya adalah data yang telah diubah ke dalam skala yang sama.

Normalisasi ini memastikan bahwa semua nilai berada dalam rentang yang konsisten sehingga lebih efektif untuk analisis dan pemrosesan seperti pembelajaran mesin atau sistem IoT berbasis data.

```
# Initialize MinMaxScaler
scaler = MinMaxScaler()

# Normalize the features
X_normalized = scaler.fit_transform(X_resampled)

# Save the scaler
scaler_path = 'scaler_final.joblib'
joblib.dump(scaler, scaler_path)
print(f"Scaler saved at {scaler_path}")

# Save the normalized data to CSV
normalized_data = pd.DataFrame(X_normalized, columns=["Temperature", "Humidity", "Raw H2", "Raw Ethanol"])
normalized_data['Fire Alarm'] = y_resampled
normalized_data.to_csv('normalized_data.csv', index=False)
print("Normalized data saved to 'normalized_data.csv'")
```

Gambar 4. 3 kodingan min-max scaler

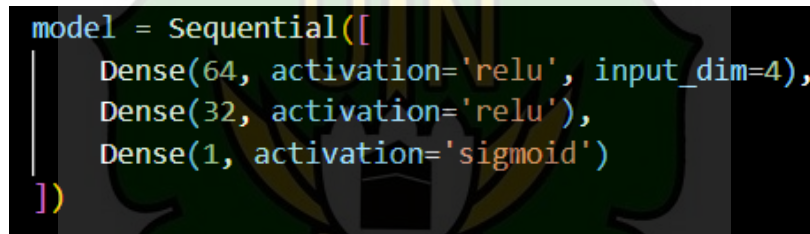
	Temperature	Humidity	Raw H2	Raw Ethanol	Fire Alarm
1	0.5126922138149866	0.7232392181197641	0.5224880382775119	0.525685212538979	0
2	0.5128752745911643	0.7125349053676698	0.5349282296650717	0.5471852946003608	0
3	0.5130461313155968	0.7015203226807322	0.5441786283891545	0.5657311669128506	0
4	0.5132291920917745	0.6909711448960595	0.5492822966507176	0.5796816018381747	0
5	0.5134122528679522	0.6818181818181818	0.5534290271132374	0.5914984408337434	0
6	0.5135831095923847	0.6729754886751473	0.5585326953748004	0.6041358936484489	0
7	0.5137661703685624	0.6650636053366428	0.5626794258373202	0.6139832594780894	0
8	0.5139492311447401	0.658703071672355	0.5649122807017544	0.6231741342524209	0
9	0.5141200878691726	0.6526528079429104	0.5677830940988833	0.6299031675693416	0
10	0.5143031486453503	0.6472230840831523	0.5693779904306218	0.6364680781224354	0
11	0.5144740053697828	0.6424139000930809	0.5696969696969694	0.6422123748563924	0
12	0.5146570661459605	0.6376047161030096	0.5738437001594896	0.6477925488265224	0
13	0.5148272922870393	0.6341917468197331	0.5738437001594896	0.6535368455604793	0
14	0.5150109836465707	0.630623642569035	0.5744816586921848	0.6564910553093712	0
15	0.5151818403710032	0.6272106732857586	0.5741626794258372	0.6599376333497453	0
16	0.5153649011471809	0.6244182438721688	0.5763955342902709	0.663876579681602	0
17	0.5155357578716134	0.6224014892956872	0.5779904306220094	0.6671590349581487	0
18	0.5157188186477911	0.619298789947254	0.5770334928229661	0.6701132447070406	0
19	0.5158896753722236	0.616661495501086	0.578309409888357	0.6737239455112425	0
20	0.5160727361484013	0.6132485262178095	0.5808612440191387	0.6784835056622351	0
21	0.5162435928728338	0.6113869066087495	0.581499202551834	0.679304119481372	0
22	0.5164144495972663	0.6093701520322681	0.5802232854864431	0.6816018381749549	0
23	0.516597510373444	0.6084393422277381	0.5818181818181816	0.6847201706876742	0
24	0.5167683670978765	0.6064225876512566	0.5827751196172248	0.6847201706876742	0
25					

Gambar 4. 4 Data yang telah di Normalisasi

4.3 Model DNN

Model dalam penelitian ini menggunakan *sequential* dengan tiga *layer*: *layer input*, *hidden layer*, dan *output layer*. *Input layer* mencakup 4 parameter seperti *temperature*, *humidity*, *raw h2*, dan *raw ethanol*, pada *hidden layer* pertama terdiri dari 64 *neuron* dengan fungsi aktivasi *ReLU* untuk menangkap pola *non-linear* dari data yang dikirim oleh 4 parameter, sedangkan *hidden layer* kedua terdiri dari 32 *neuron*, pada *layer output* memiliki 1 *neuron* dengan fungsi kativasi *sigmoid*, yang mengubah output menjadi 1/0 sesuai dengan klasifikasi biner pada persamaan 4.

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (4)$$

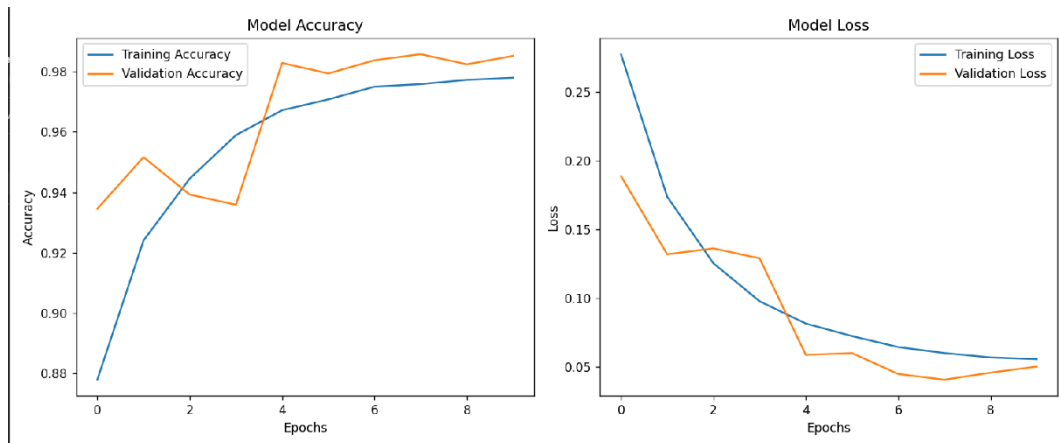


```
model = Sequential([
    Dense(64, activation='relu', input_dim=4),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

Gambar 4. 5 kodingan pembuatan model dnn

4.4 Hasil *Model Accuracy* dan *Model Loss*

Model accuracy dan *model loss* adalah dua metrik utama dalam evaluasi model DNN ketika belajar. Akurasi mengukur persentase prediksi yang benar di bandingkan total prediksi dengan memberikan beberapa baik model mengklasifikasikan data secara keseluruhan agar dapat seimbang. Pada *model loss* menghitung beberapa jauh prediksi model dari nilai target sebenarnya, dengan menggunakan fungsi loss seperti *binery crossentropy* untuk klasifikasi biner. Selama proses pelatihan, akurasi yang meningkat dan loss yang menurun menunjukkan bahwa model belajar dengan baik, sedangkan perbedaan signifikan antara keduanya pada data validasi dapat mengindikasikan *overfitting*, pada penelitian ini akurasi yang di dapat sebesar 98% dan hasil *F1-score* 0.98.



Gambar 4. 6 Model Accuracy dan model loss

Tabel 4. 1 Hasil Akurasi

Epoch	Training Accuracy	Validation Accuracy
1	0.859936	0.928018
2	0.922759	0.940864
3	0.949239	0.959506
4	0.963971	0.965650
5	0.969662	0.979543
6	0.972647	0.973330
7	0.973956	0.984780
8	0.975213	0.984500
9	0.976993	0.979683
10	0.977849	0.966697

Tabel 4. 2 *Classification Report*

	precision	recall	f1-score	support
Class 0	0.991465	0.978437	0.984908	8904.000000
Class 1	0.978938	0.991666	0.985261	8999.000000
accuracy	0.985086	0.985086	0.985086	0.985086
macro avg	0.985201	0.985051	0.985084	17903.000000
weighted avg	0.985168	0.985086	0.985085	17903.000000

4.5 Pengiriman Data ke *Device Ubidots*

Pengiriman data ke ubidots memiliki beberapa tahapan: pertama, data mentah dari sensor yang mendeteksi adanya gas akan mengirimkan data tersebut melewati *serial baud* 9600 dan *serial port COM3* untuk di proses pada *tensorflow*. Tahap kedua data yang di terima oleh *tensorflow* lalu di proses dengan model DNN dan hasil prediksi dari model dikirim ke *ubidots* menggunakan TOKEN dan API.

```

Data mentah: {"temperature":28.80,"humidity":86.30,"mq2_h2":13550,"mq2_ethanol":21680}
Data diterima: {'temperature': 28.8, 'humidity': 86.3, 'mq2_h2': 13550, 'mq2_ethanol': 21680}
esp: [[ 28.8  86.3 13550. 21680. ]]
C:\Users\ihsan\Documents\exampl\.venv\lib\site-packages\sklearn\utils\validation.py:2739: Use
Warning: X does not have valid feature names, but MinMaxScaler was fitted with feature names
warnings.warn(
Data yang dikirim ke model: [[0.62008787 1.17219981 0.91929825 1.04431315]]
1/1 0s 53ms/step
c:\Users\ihsan\Documents\exampl\ubidot2.py:61: DeprecationWarning: Conversion of an array wit
ndim > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single el
ment from your array before performing this operation. (Deprecated NumPy 1.25.)
fire_alarm_pred = int(prediction > 0.5)
Prediksi Fire Alarm: 0
Data berhasil dikirim ke Ubidots

```

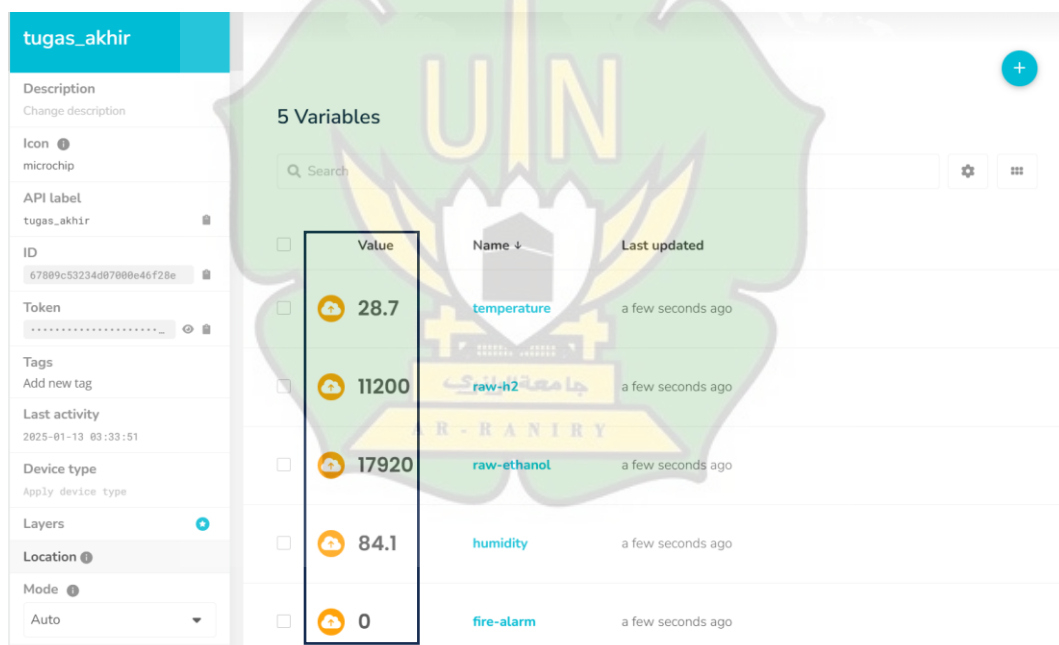
Gambar 4. 7 proses pengiriman data ke ubidots

Pada tahap ini dapat dilihat tanda pada titik biru pada *Name Label* yang menandakan koneksi antara python dengan ubidots berhasil terhubung.



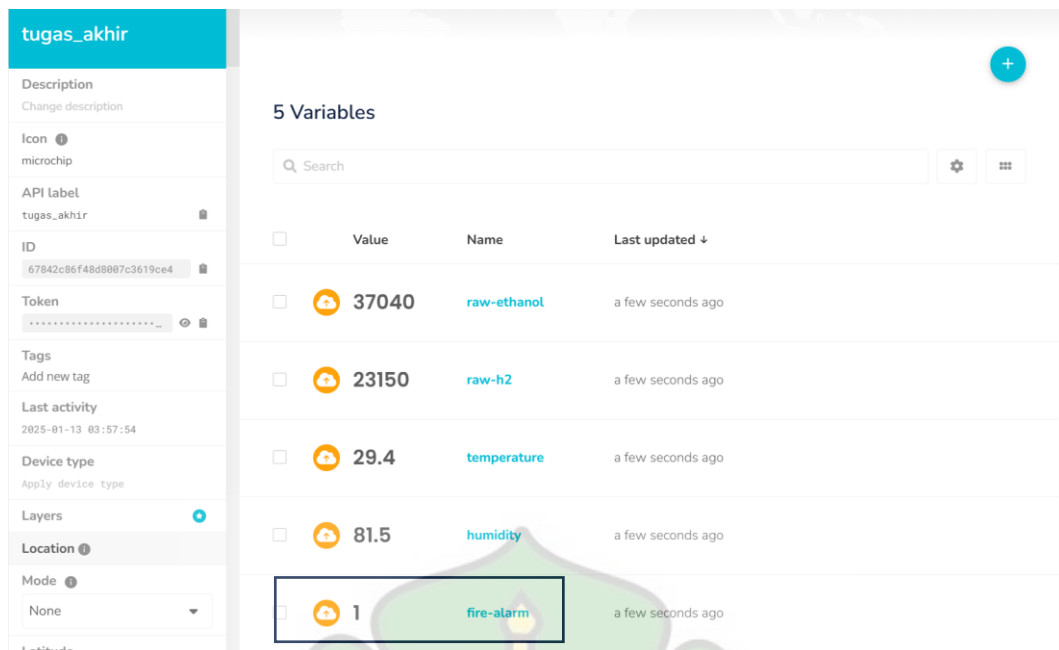
Gambar 4. 8 Tanda terkoneksi ke ubidots

Selanjutnya masuk pada *Name Label tugas_akhir*, pada tampilan ini terdapat 5 *variables* yaitu, temperatur, raw-h2, raw-ethanol, humidity, dan fire-alarm. Pada kolom *value* adalah data yang dikirim dari python sesuai dengan pendeteksian dari sensor.



Gambar 4. 9 Tampilan Device Ubidots

Khususnya pada *variable fire-alarm* nilai value yang di keluarkan 1 atau 0, yang dimana *fire-alarm* tersebut adalah hasil dari nilai prediksi jika nilai 1 maka gas terdeteksi, yang mana proses ini telah ditentukan oleh model DNN yang telah dibangun.



Gambar 4. 10 Tanda Gas Terdeteksi

Pada halaman *dashboard ubidots* menunjukkan diagram dalam 5 *variables*, pada gambar 4.17 dan 4.18, menunjukkan widget yang berwarna hijau dan merah dan bergerakanya *line chart* sebagai pertanda tidak adanya gas dan terdeteksinya gas dari sensor.

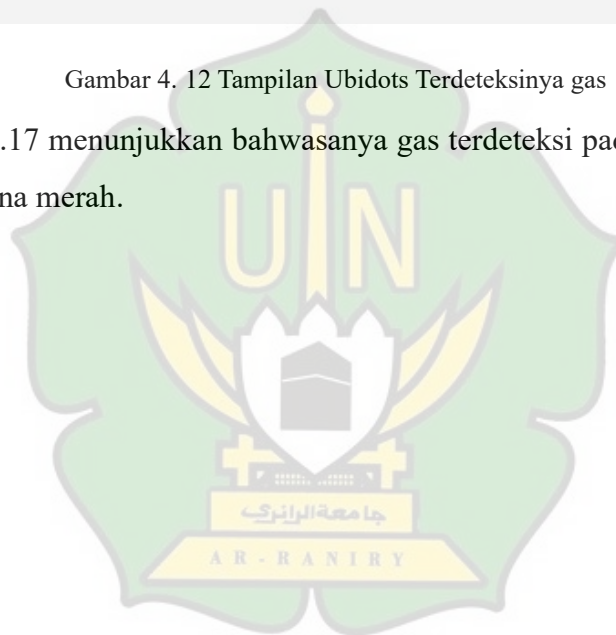


Gambar 4. 11 Tampilan ubidots tidak terdeteksinya gas.



Gambar 4. 12 Tampilan Ubidots Terdeteksinya gas

Pada gambar 4.17 menunjukkan bahwasanya gas terdeteksi pada *indicator widget* dengan berwarna merah.



BAB IV

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penjelasan pada bab sebelumnya, maka dapat di simpulkan beberapa hal sebagai berikut:

1. Pengembangan sistem menggunakan metode deep learning sangat efektif dalam memprediksi adanya kebocoran gas.
2. Sistem yang dikembangkan berdasarkan IoT dapat mengirimkan data secara real-time melalui platform Ubidots dengan akurat.
3. Penggunaan *Deep Neural Network* (DNN) dapat memproses dan mengenali pola data dari sensor dengan lebih baik dan dapat meningkatkan sensitivitas sistem terhadap varian gas.
4. Sistem ini dapat memberikan Solusi inovasi dalam mengembangkan sistem lain yang berbasis IoT
5. Dalam perihal peringatan dini dapat sistem ini sangat berguna untuk di implementasikan dalam media edukasi kesiap siagaan bencana kebakaran.

5.2 Saran

Setelah penelitian ini, peneliti memiliki beberapa saran yang dapat di terapkan pada penelitian berikutnya. Adapun saran tersebut adalah sebagai berikut:

1. Penambahan fitur notifikasi pesan singkat yang dapat di kirim ke pengguna.
2. Penambahan sensor MQ yang bervariasi guna meningkatkan akurasi prediksi yang di berikan oleh sensor.
3. Penambahan parameter pada dataset, berupa mendeteksi CO, CO₂, dan gas beracun.
4. Mengubah sensor gas menjadi kamera thermal dengan metode Convolutional Neural Network (CNN) guna untuk mendeteksi suhu gas.

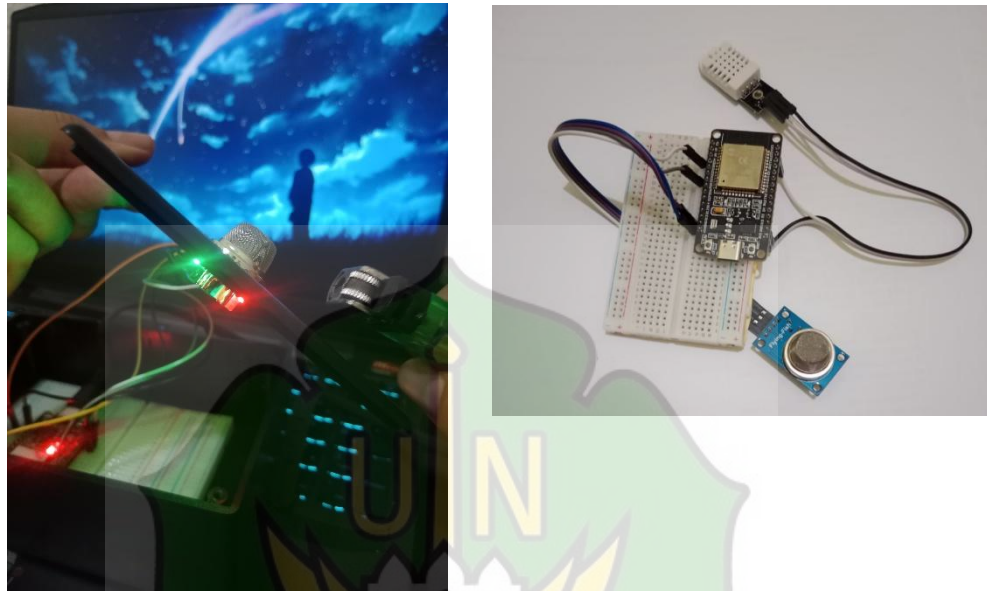
DAFTAR PUSTAKA

- Akil, I. (2023). KOMPARASI FUNGSI AKTIVASI NEURAL NETWORK PADA DATA TIME SERIES. *INTI Nusa Mandiri*, 18(1), 78–83. <https://doi.org/10.33480/inti.v18i1.4288>
- Athallah, Y., & Agung, R. (2022). *JURNAL TEKNIK INFORMATIKA STMIK ANTAR BANGSA Rancang Bangun Prototype Monitoring Lampu Jalan Secara Otomatis Menggunakan Mikrokontroller ESP32 Dan Api Bot Telegram*. <http://awesomerockguy.blogspot.com/2015/10/tutorial->
- Fajar, M., & Nugraha, Z. (2023). *Deteksi Aritmia Menggunakan Algoritma Deep Neural Network (Dnn) Pada Sinyal Elektrokardiogram* (Vol. 10, Nomor 5).
- Istiyanto, I., Solehudin, R., Nofarenzi, Y., Setiyorini, T., & Mandiri, U. N. (2022). Alat Pendeteksi Dini Kebocoran Gas LPG Dengan Sensor MQ2 Dan Sensor Api Berbasis IoT Menggunakan NodeMCU. Dalam *Jurnal* (Vol. 4, Nomor 1). <http://ejournal.bsi.ac.id/ejurnal/index.php/infortech>
- Nur Anggraeni, R., Pratama, P., Sujjada, A., & Fergina, A. (2023). *Prediksi Penyakit Jantung Menggunakan Metode Deep Neural Network dengan Memanfaatkan Internet of Things*. <https://doi.org/10.37034/jidt.v5i1.330>
- Riyadi, M. (2024). *Prediksi Daerah Potensi Bencana Kebakaran Hutan Akibat Perubahan Penggunaan Lahan di Kabupaten Paser, Kalimantan Timur*.
- Santos, A. Dos, Pereira, G. S., Syuhada, R. A., Malays, E., & Sakti, S. (2024). *Uji Coba Keamanan Database Website Menggunakan Python Dan Sqlmap Melalui Command Prompt Pada Sistem Operasi Windows*. <https://doi.org/10.37817/tekinfo.v25i1>
- Satria, D. (2023). Sistem Peringatan Dini Kebakaran Dan Kebocoran Gas LPG Berbasis Notifikasi SMS Gateway. *JURNAL INFORMATIKA*, 2(2). <https://jurnal.uniraya.ac.id/index.php/JI>

- Sinta, T., Teknik Elektro, J., Fauzil Akbar, A., Cahyadi, W., & Muldayani, W. (2023). Prototype Sistem Peringatan Dini Berbasis Iot (Internet of Things) untuk Mitigasi Bencana Tsunami di Pantai Pancer, Puger, Jember. *Jurnal Arus Elektro Indonesia (JAEI)*, 9(1).
- Subhan, S., & Virgian Shaka Yudha Sakti, D. (2023). Alat Pendeteksi Kebakaran Menggunakan NodeMCU ESP8266 Waterpump Flame Sensor dan Sensor MQ-2 Berbasis Web. Dalam *SKANIKA: Sistem Komputer dan Teknik Informatika* (Vol. 6, Nomor 2).
- Suryana, T. (2021). *Implementasi Modul Sensor MQ2 Untuk Mendeteksi Adanya Polutan Gas di Udara*. <http://iot.ciwaruga.com>
- Teguh, R., Abdiel Mempun, K., Palangka Raya, U., Timang Universitas Palangka Raya, J. H., & Timang, J. H. (2018). RANCANG BANGUN DESAIN INTERNET OF THINGS UNTUK PEMANTAUAN KUALITAS UDARA PADA STUDI KASUS POLUSI UDARA. *Jurnal Teknologi Informasi*, 12(2).
- Wikantama, P. T., Bahalwan, M., & Akmal, M. A. G. (2024). SIGEMPA : Sistem Peringatan Dini Gempa Bumi berbasis IoT dengan ESP32. *Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 4(1), 63–70. <https://doi.org/10.55606/teknik.v4i1.2937>
- Yusuf, R. M. (2020). *Rancang Bangun Alat Deteksi Pelanggaran Kendaraan Pada Trotoar Di Wilayah Dinas Perhubungan Provinsi Dki Jakarta (Doctoral dissertation, Universitas Komputer Indonesia)*.

LAMPIRAN

Lampiran 1 Dokumentasi Hardware



Lampiran 2 Kode Pada *Arduino IDE*

```
#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>

#include <DHT_U.h>


// Define pins and constants

#define DHTPIN 22    // Pin connected to DHT22 data

#define DHTTYPE DHT22 // DHT22 (AM2302)

#define MQ2_PIN 34  // Analog pin connected to MQ2 sensor
```

```
// Initialize DHT sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {

  Serial.begin(9600); // Sesuaikan dengan baud rate Python

  dht.begin();

  pinMode(MQ2_PIN, INPUT);
}

void loop() {

  // Read temperature and humidity from DHT22

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // Check if readings are valid

  if (isnan(temperature) || isnan(humidity)) {

    Serial.println("{\"error\":\"DHT sensor read failed\"}");

    delay(1000);

    return;

  }

  // Read raw values from MQ2 sensor
```

```

int mq2_value = analogRead(MQ2_PIN);

//( calibration)

int mq2_h2 = mq2_value * 50;

int mq2_ethanol = mq2_value * 80;

// Format JSON payload

String json_payload = "{";

json_payload += "\"Temperature\":" + String(temperature, 2) + ",";

json_payload += "\"Humidity\":" + String(humidity, 2) + ",";

json_payload += "\"Raw H2\":" + String(mq2_h2) + ",";

json_payload += "\"Raw Ethanol\":" + String(mq2_ethanol);

json_payload += "}";

// Send JSON payload via Serial

Serial.println(json_payload);

// Wait before the next reading

delay(2000);

}

```

Lampiran 3 Kode Normalisasi Data

```

import pandas as pd

```



```

import joblib

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import accuracy_score

from imblearn.over_sampling import SMOTE

import tensorflow as tf

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Dense

# Load the dataset

file_path = 'datasensor2.csv'

try:

    data = pd.read_csv(file_path)

except FileNotFoundError:

    raise FileNotFoundError(f'File not found at {file_path}. Please check
the file path.')

# Split the single column into multiple columns based on the ";" separator

columns = ["Temperature", "Humidity", "Raw H2", "Raw Ethanol", "Fire
Alarm"]

try:

    data_split = data.iloc[:, 0].str.split(';', expand=True)

    if data_split.shape[1] != len(columns):

```

```
        raise ValueError(f'Expected {len(columns)} columns, but got
{data_split.shape[1]} columns.")
```

```
    data_split.columns = columns # Assign proper column names
```

```
    data = data_split
```

```
except Exception as e:
```

```
    raise ValueError(f'Error splitting columns: {e}')
```

```
# Convert data types to numeric and handle errors
```

```
data = data.apply(pd.to_numeric, errors='coerce')
```

```
# Drop rows with NaN values
```

```
data = data.dropna()
```

```
# Separate features and target variable (using all 4 features)
```

```
X = data[["Temperature", "Humidity", "Raw H2", "Raw Ethanol"]]
```

```
y = data['Fire Alarm']
```

```
# Apply SMOTE to balance the dataset
```

```
smote = SMOTE(random_state=42)
```

```
X_resampled, y_resampled = smote.fit_resample(X,y)
```

```
print(f'Original dataset shape: {X.shape}')
```

```
print(f'Resampled dataset shape: {X_resampled.shape}')
```

```

# Initialize MinMaxScaler

scaler = MinMaxScaler()

# Normalize the features

X_normalized = scaler.fit_transform(X_resampled)

# Save the scaler

scaler_path = 'scaler_final.joblib'
joblib.dump(scaler, scaler_path)
print(f"Scaler saved at {scaler_path}")

# Save the normalized data to CSV

normalized_data = pd.DataFrame(X_normalized,
                                columns=["Temperature", "Humidity", "Raw H2", "Raw Ethanol"])
normalized_data['Fire Alarm'] = y_resampled

normalized_data.to_csv('normalized_data.csv', index=False)

print("Normalized data saved to 'normalized_data.csv'")

# Split resampled data into training (80%), validation (10%), and testing
(10%) sets

X_train, X_test, y_train, y_test = train_test_split(X_normalized,
                                                    y_resampled, test_size=0.2, random_state=42)

```

```

model = Sequential([
    Dense(64, activation='relu', input_dim=4),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid') # Output layer for binary classification
])

```

```

# Compile the model

```

```

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

```

```

# Train the model

```

```

history = model.fit(X_train, y_train, epochs=10, batch_size=4,
validation_split=0.2, verbose=1)

```

```

# Save the trained model in TensorFlow format

```

```

model_path = 'model_final.keras'

```

```

model.save(model_path)

```

```

print(f"Model saved at {model_path}")

```

```

# Predict and evaluate

```

```

y_pred_proba = model.predict(X_test)

```

```

y_pred = (y_pred_proba > 0.5).astype("int32")

```

```
# Print accuracy

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy * 100:.2f}%')
```

Lampiran 4 Kode Pengiriman Data ke Ubidots

```
import serial

import json

import numpy as np

import joblib

from tensorflow.keras.models import load_model

import requests

import time

from urllib.parse import quote # For URL encoding

# Pengaturan serial

SERIAL_PORT = 'COM3' # Ganti dengan port serial Arduino Anda

BAUD_RATE = 9600

# Pengaturan Ubidots

UBIDOTS_TOKEN = "BBUS-xda9oNBqxEx6cx45OC5VxgCt4YGCnZm"
```

```
UBIDOTS_DEVICE = "tugas_akhir" # Ensure the device name matches
exactly
```

```
UBIDOTS_URL =
f"https://industrial.api.ubidots.com/api/v1.6/devices/{quote(UBIDOTS_
DEVICE)}" # URL-encode the device name
```

```
# Load the scaler and DNN model
```

```
SCALER_FILE = 'scaler_final.joblib'
```

```
MODEL_FILE = 'model_final.keras'
```

```
try:
```

```
    scaler = joblib.load(SCALER_FILE)
```

```
except FileNotFoundError:
```

```
    print(f"Scaler file {SCALER_FILE} tidak ditemukan.")
```

```
    exit(1)
```

```
try:
```

```
    model = load_model(MODEL_FILE)
```

```
except Exception as e:
```

```
    print(f"Gagal memuat model: {e}")
```

```
    exit(1)
```

```
def send_to_ubidots(payload):
```

```

"""Mengirim data ke Ubidots dengan timestamp"""

headers = {

    "X-Auth-Token": UBIDOTS_TOKEN,

    "Content-Type": "application/json"

}

response = requests.post(UBIDOTS_URL, headers=headers,
json=payload)

if response.status_code == 200:

    print(f'Data berhasil dikirim ke Ubidots pada {time.strftime("%Y-
%m-%d %H:%M:%S")}')

    print("-----")

else:

    print(f'Error mengirim data ke Ubidots: {response.status_code},
{response.text}')

def main():

    try:

        ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)

        print(f'Terhubung ke {SERIAL_PORT}')

    while True:

        if ser.in_waiting > 0:

```



```
raw_line = ser.readline().decode('utf-8', errors='ignore').strip()
```

```
print(f'Data mentah: {raw_line}')
```

```
try:
```

```
    data = json.loads(raw_line)
```

```
    print(f'Data diterima: {data}')
```

```
features = ['Temperature', 'Humidity', 'Raw H2', 'Raw Ethanol']
```

```
input_data = np.array([[data[feature] for feature in features]])
```

```
print(f'esp: {input_data}')
```

```
normalized_data = scaler.transform(input_data)
```

```
print(f'Data yang dikirim ke model: {normalized_data}')
```

```
prediction = model.predict(normalized_data)
```

```
fire_alarm_pred = int(prediction > 0.5)
```

```
print(f'Prediksi Fire Alarm: {fire_alarm_pred}')
```

```
payload = {
```

```
    "Temperature": data["Temperature"],
```

```
    "Humidity": data["Humidity"],
```

```
    "Raw H2": data["Raw H2"],
```

```
    "Raw Ethanol": data["Raw Ethanol"],
```

```
    "Fire Alarm": fire_alarm_pred
```

```

    }

    send_to_ubidots(payload)

except json.JSONDecodeError:

    print(f"Gagal memproses data JSON: {raw_line}")

except serial.SerialException as e:

    print(f"Error membuka port serial: {e}")

except KeyboardInterrupt:

    print("Program dihentikan")

finally:

    if 'ser' in locals() and ser.is_open:

        ser.close()

if __name__ == "__main__":

    main()

```

Lampiran 5 Data Normalisasi

Temperature	Humidity	Raw H2	Raw Ethanol	Fire Alarm
0.5126922138 149866	0.723239218 1197641	0.522488038 2775119	0.525685212 538979	0

0.5128752745 911643	0.712534905 3676698	0.534928229 6650717	0.547185294 6003608	0
0.5130461313 155968	0.701520322 6807322	0.544178628 3891545	0.565731166 9128506	0
0.5132291920 917745	0.690971144 8960595	0.549282296 6507176	0.579681601 8381747	0
0.5134122528 679522	0.681818181 8181818	0.553429027 1132374	0.591498440 8337434	0
0.5135831095 923847	0.672975488 6751473	0.558532695 3748004	0.604135893 6484489	0
0.5137661703 685624	0.665063605 3366428	0.562679425 8373202	0.613983259 4780894	0
0.5139492311 447401	0.658703071 672355	0.564912280 7017544	0.623174134 2524209	0
0.5141200878 691726	0.652652807 9429104	0.567783094 0988833	0.629903167 5693416	0
0.5143031486 453503	0.647223084 0831523	0.569377990 4306218	0.636468078 1224354	0
0.5144740053 697828	0.642413900 0930809	0.569696969 6969694	0.642212374 8563924	0
0.5146570661 459605	0.637604716 1030096	0.573843700 1594896	0.647792548 8265224	0
0.5148279228 70393	0.634191746 8197331	0.573843700 1594896	0.653536845 5604793	0

0.5150109836 465707	0.630623642 569035	0.574481658 6921848	0.656491055 3093712	0
0.5151818403 710032	0.627210673 2857586	0.574162679 4258372	0.659937633 3497453	0
0.5153649011 471809	0.624418243 8721688	0.576395534 2902709	0.663876579 681602	0
0.5155357578 716134	0.622401489 2956872	0.577990430 6220094	0.667159034 9581487	0
0.5157188186 477911	0.619298789 947254	0.577033492 8229661	0.670113244 7070406	0
0.5158896753 722236	0.616661495 501086	0.578309409 888357	0.673723945 5112425	0
0.5160727361 484013	0.613248526 2178095	0.580861244 0191387	0.678483505 6622351	0
0.5162435928 728338	0.611386906 6087495	0.581499202 551834	0.679304119 481372	0
0.5164144495 972663	0.609370152 0322681	0.580223285 4864431	0.681601838 1749549	0
0.5165975103 73444	0.608439342 2277381	0.581818181 8181816	0.684720170 6876742	0
0.5167683670 978765	0.606422587 6512566	0.582775119 6172248	0.684720170 6876742	0
0.5169392238 223091	0.605336642 8793049	0.583732057 4162677	0.687017889 3812571	0

0.5171222845 984867	0.603630158 2376667	0.584051036 6826153	0.689151485 3110126	0
0.5172931413 229193	0.602544213 4657151	0.585326953 7480061	0.690628590 1854585	0
0.5174639980 473518	0.599286379 1498603	0.577352472 0893142	0.686525521 0897749	0
0.5176470588 235295	0.602544213 4657151	0.575438596 4912281	0.684391925 1600195	0
0.5178179155 47962	0.610766366 7390629	0.575119617 22488	0.685376661 7429837	0
0.5179887722 723945	0.619609059 8820974	0.578309409 888357	0.688002625 9642213	0
0.5181596289 96827	0.628141483 0902885	0.578628389 1547046	0.689807976 3663221	0
0.5183304857 212595	0.636518771 331058	0.577671451 3556618	0.688987362 5471852	0
0.5185135464 974372	0.644430654 6695625	0.579904306 2200955	0.691777449 5322502	0
0.5186844032 218697	0.651566863 1709587	0.580861244 0191387	0.692269817 8237319	0
0.5188552599 463022	0.657461991 9329816	0.581180223 2854864	0.693254554 4066961	0
0.5190261166 707347	0.663357120 6950046	0.581499202 551834	0.693254554 4066961	0

0.5191969733 951672	0.668631709 587341	0.583732057 4162677	0.695059904 8087969	0
0.5193678301 195997	0.673130623 642569	0.583732057 4162677	0.697849991 7938619	0
0.5195386868 440323	0.677474402 7303753	0.581499202 551834	0.696208764 1555881	0
0.5197095435 684648	0.681507911 8833385	0.583732057 4162677	0.695224027 5726242	0
0.5198804002 928973	0.685231151 1014582	0.585326953 7480061	0.699327096 6683078	0
0.5200512570 173298	0.687868445 5476264	0.583413078 14992	0.697193500 7385524	0
0.5202221137 417623	0.690971144 8960595	0.585964912 2807014	0.697193500 7385524	0
0.5203929704 661948	0.694228979 2119144	0.586921850 0797446	0.699819464 95979	0
0.5205638271 906273	0.699968973 0065156	0.584688995 2153109	0.698670605 6129983	0
0.5207346839 150598	0.701985727 5829972	0.585964912 2807014	0.699983587 7236174	0
0.5209055406 394923	0.703847347 192057	0.586602870 813397	0.700475956 015099	0
0.5210763973 639249	0.704933291 9640087	0.588835725 6778307	0.701953060 8895455	0

0.5212472540 883574	0.707105181 5079118	0.587559808 6124398	0.700804201 5427538	0
0.5214181108 127899	0.707725721 3775985	0.585326953 7480061	0.699655342 1959626	0
0.5215889675 372224	0.709432206 0192366	0.586921850 0797446	0.700804201 5427538	0
0.5217598242 616549	0.710673285 7586099	0.588197767 1451355	0.702281306 4171998	0
0.5219184769 343422	0.711448960 5957183	0.588516746 4114831	0.701460692 5980633	0
0.5220893336 587747	0.712379770 4002481	0.588835725 6778307	0.702117183 6533728	0
0.5222601903 832073	0.716258144 5857895	0.595534290 2711322	0.707533234 859675	0
0.5224310471 076398	0.712069500 4654048	0.594577352 472089	0.707204989 3320203	0
0.5226019038 320724	0.703071672 3549488	0.596491228 0701751	0.706876743 8043655	0
0.5227605565 047596	0.692522494 5702761	0.596810207 3365231	0.707697357 6235024	0
0.5229314132 291921	0.682748991 6227117	0.596810207 3365231	0.708846216 9702936	0
0.5231022699 536247	0.673285758 6099906	0.595853269 5374799	0.707369112 0958476	0

0.5232731266 780571	0.664908470 3692212	0.593620414 6730462	0.707040866 5681929	0
0.5234317793 507445	0.657306856 96556	0.594258373 2057414	0.708682094 2064662	0
0.5236026360 75177	0.650325783 4315854	0.595853269 5374799	0.710651567 3723943	0
0.5237734927 996096	0.644275519 7021408	0.595215311 0047846	0.708682094 2064662	0
0.5239321454 722968	0.639156065 7772261	0.592982456 1403505	0.706712621 0405381	0
0.5241030021 967293	0.634812286 6894197	0.593939393 9393938	0.708682094 2064662	0
0.5242616548 694167	0.630468507 6016134	0.594577352 472089	0.708025603 1511571	0
0.5244325115 938492	0.627055538 3183368	0.594896331 7384366	0.707369112 0958476	0
0.5246033683 182817	0.623332299 1002171	0.595853269 5374799	0.707861480 3873298	0
0.5247620209 909691	0.620229599 7517839	0.593620414 6730462	0.706548498 2767107	0
0.5249328777 154015	0.617592305 3056158	0.595215311 0047846	0.710487444 6085669	0
0.5250915303 880889	0.614955010 8594477	0.593620414 6730462	0.706712621 0405381	0

0.5252623871 125215	0.612783121 3155445	0.594577352 472089	0.707204989 3320203	0
0.5254210397 852087	0.610611231 7716413	0.594896331 7384366	0.708353848 6788115	0
0.5255918965 096412	0.607973937 3254731	0.592663476 8740029	0.706548498 2767107	0
0.5257505491 823287	0.607198262 4883648	0.593301435 4066986	0.706220252 7490564	0
0.5259214059 067611	0.605026372 9444617	0.595853269 5374799	0.708025603 1511571	0
0.5260800585 794485	0.603940428 17251	0.595853269 5374799	0.706056129 985229	0
0.5262387112 521357	0.602078808 5634501	0.596172248 8038275	0.708025603 1511571	0
0.5264095679 765682	0.600837728 8240769	0.595215311 0047846	0.708025603 1511571	0
0.5265682206 492557	0.600372323 921812	0.597129186 6028707	0.709174462 4979484	0
0.5267268733 219429	0.599596649 0847037	0.597129186 6028707	0.709174462 4979484	0
0.5268977300 463755	0.598355569 3453304	0.594896331 7384366	0.707861480 3873298	0
0.5270563827 190627	0.596804219 6711139	0.595853269 5374799	0.706548498 2767107	0

0.5272150353 917502	0.594632330 1272106	0.589154704 9441783	0.702281306 4171998	0
0.5273858921 161827	0.598045299 4104871	0.589473684 2105259	0.704414902 3469553	0
0.5275445447 888699	0.606112317 7164133	0.587559808 6124398	0.702773674 7086819	0
0.5277031974 615574	0.616040955 6313993	0.589792663 476874	0.704579025 1107826	0
0.5278618501 342446	0.625038783 7418554	0.590430622 0095692	0.704250779 5831279	0
0.5280327068 586772	0.633260937 0152033	0.588197767 1451355	0.703922534 0554736	0
0.5281913595 313645	0.640707415 4514427	0.592982456 1403505	0.705727884 4575743	0
0.5283500122 040519	0.648464163 8225256	0.592344497 6076553	0.706056129 985229	0
0.5285086648 767392	0.654359292 5845485	0.592663476 8740029	0.705399638 9299195	0
0.5286673175 494265	0.660564691 2814148	0.591387559 8086125	0.705892007 2214017	0
0.5288381742 73859	0.665684145 2063295	0.593301435 4066986	0.706056129 985229	0
0.5289968269 465463	0.670493329 1964008	0.592025518 3413077	0.705235516 1660921	0

0.5291554796 192337	0.674992243 2516289	0.593939393 9393938	0.704579025 1107826	0
0.5293141322 919209	0.679180887 3720137	0.593620414 6730462	0.707204989 3320203	0
0.5294727849 646084	0.682593856 65529	0.594896331 7384366	0.706876743 8043655	0
0.3830973883 329266	0.715482469 7486812	0.854545454 5454545	0.794846545 2158215	1
0.3829387356 602393	0.742475954 0800496	0.866666666 6666663	0.801739701 2965697	1
0.3827922870 392971	0.770089978 2811046	0.867623604 4657095	0.809289348 4326273	1
0.3826336343 6660974	0.798634812 2866893	0.885167464 1148319	0.820285573 6090595	1
0.3824871857 456676	0.812441824 3872168	0.889633173 8437002	0.825701624 8153617	1
0.3823285330 7298026	0.821129382 5628296	0.891866028 7081339	0.829148202 8557362	1
0.3821820844 520381	0.827955321 1293825	0.895055821 3716109	0.835220745 1173479	1
0.3820234317 793508	0.823146137 1393111	0.896012759 1706541	0.839159691 4492041	1
0.3818769831 584087	0.813527769 1591684	0.892503987 2408291	0.838667323 157722	1

0.3817183304 8572135	0.803443996 2767608	0.889314194 5773522	0.839816182 5045132	1
0.3815718818 6477914	0.793360223 3943531	0.887081339 7129185	0.840472673 5598227	1
0.3814132291 920918	0.784207260 3164753	0.886443381 1802233	0.839980305 2683406	1
0.3812667805 7114965	0.776605646 9128141	0.883572567 7830943	0.838995568 6853767	1
0.3811081278 984623	0.771175923 0530562	0.883572567 7830943	0.839980305 2683406	1
0.3809616792 7752017	0.766211604 0955631	0.880063795 8532693	0.840144428 0321679	1
0.3808030266 0483284	0.761867825 0077567	0.880063795 8532693	0.839323814 213031	1
0.3806565779 838907	0.758299720 7570585	0.878787878 7878789	0.838995568 6853767	1
0.3804979253 1120336	0.755197021 4086254	0.877192982 4561404	0.837518463 8109303	1
0.3803514766 902612	0.753025131 8647223	0.875598086 1244019	0.837682586 5747577	1
0.3801928240 1757387	0.750853242 3208191	0.876555023 9234452	0.837190218 283276	1
0.3800463753 966317	0.748681352 7769158	0.873046251 9936202	0.835384867 8811748	1

0.3798877227 239444	0.747595408 0049643	0.871770334 9282298	0.835056622 3535205	1
0.3797412741 0300224	0.746354328 265591	0.868261562 9984047	0.832758903 6599377	1
0.3795826214 303149	0.744492708 656531	0.865390749 6012758	0.832102412 6046281	1
0.3794361728 0937275	0.743717033 8194229	0.865390749 6012758	0.830625307 7301822	1
0.3792775201 366854	0.742631089 0474712	0.863157894 7368421	0.830297062 2025274	1
0.3791310715 1574327	0.742475954 0800496	0.860925039 8724084	0.828819957 3280814	1
0.3789724188 4305594	0.741079739 3732547	0.858373205 7416267	0.826850484 1621533	1
0.3788259702 221138	0.740614334 4709897	0.859330143 5406699	0.827506975 2174624	1
0.3786673175 4942645	0.739993794 601303	0.859330143 5406699	0.825865747 5791891	1
0.3785208689 284843	0.739373254 7316164	0.855502392 3444978	0.823896274 413261	1
0.3783622162 5579697	0.739218119 7641948	0.855502392 3444978	0.824224519 9409157	1
0.3782157676 348548	0.738752714 8619298	0.852312599 6810208	0.822583292 3026424	1

0.3780571149 621675	0.737977040 0248216	0.851355661 8819775	0.819464959 7899226	1
0.3779106663 4122533	0.737046230 2202916	0.851674641 1483256	0.819300837 0260952	1
0.3777520136 68538	0.736891095 25287	0.848165869 2185006	0.817987854 9154767	1
0.3776055650 4759585	0.735494880 546075	0.844019138 7559803	0.813556540 2921383	1
0.3774469123 749085	0.744182438 7216879	0.839234449 7607658	0.811587067 1262102	1
0.3773004637 5396636	0.758299720 7570585	0.840510366 8261562	0.811422944 3623828	1
0.3771418110 8127903	0.771331058 0204777	0.839234449 7607658	0.810602330 5432464	1
0.3769953624 603369	0.782966180 577102	0.839553429 0271129	0.811094698 8347281	1
0.3768367097 8764955	0.792739683 5246665	0.839553429 0271129	0.809453471 1964547	1
0.3766902611 667074	0.801582376 6677008	0.839234449 7607658	0.808632857 3773183	1
0.3765316084 9402006	0.809494260 0062054	0.835725677 8309408	0.807319875 2666992	1
0.3763851598 730779	0.816320198 5727583	0.835087719 2982455	0.805186279 3369438	1

0.3762265072 003906	0.821439652 4976729	0.832535885 1674638	0.803709174 4624978	1
0.3760800585 7944843	0.826093701 5203227	0.831578947 3684205	0.803380928 934843	1
0.3759214059 067611	0.830747750 5429725	0.831897926 6347686	0.802396192 3518792	1
0.3757749572 8581894	0.834315854 7936703	0.832854864 4338118	0.802888560 6433613	1
0.3756163046 131316	0.837883959 0443686	0.830622009 5693782	0.800754964 7136059	1
0.3754698559 9218946	0.840676388 4579586	0.830940988 8357253	0.801575578 5327423	1
0.3753112033 1950213	0.843468817 8715483	0.830303030 3030301	0.799441982 6029868	1
0.3751647546 9856	0.846106112 3177164	0.827751196 1722483	0.797636632 2008861	1
0.3750061020 2587265	0.847967731 9267763	0.825837320 5741627	0.797636632 2008861	1
0.3748596534 049305	0.849829351 535836	0.826475279 1068579	0.797144263 909404	1
0.3747010007 3224316	0.851225566 242631	0.825837320 5741627	0.795995404 5626128	1
0.3745545521 1130095	0.853242320 8191126	0.827432216 9059012	0.795667159 034958	1

0.3743958994 386136	0.854328265 5910641	0.825518341 3078147	0.794354176 9243394	1
0.3742494508 176715	0.855879615 2652807	0.825837320 5741627	0.795338913 5073032	1
0.3740907981 449842	0.856500155 1349675	0.822966507 1770329	0.791399967 175447	1
0.3739443495 2404204	0.857896369 8417624	0.822647527 9106857	0.791892335 4669292	1
0.3737856968 513547	0.858672044 6788706	0.822328548 6443377	0.792877072 0498934	1
0.3736392482 3041256	0.859602854 4834006	0.821371610 8452944	0.790907598 8839653	1
0.3734805955 577252	0.860533664 2879303	0.821690590 1116425	0.790415230 5924832	1
0.3733341469 36783	0.861154204 1576171	0.821052631 5789473	0.789922862 3010011	1
0.3731754942 640957	0.862395283 8969903	0.818819776 7145136	0.789758739 5371737	1
0.3730290456 4315353	0.864101768 5386284	0.823923444 9760762	0.791399967 175447	1
0.3728703929 704662	0.854328265 5910641	0.825199362 0414675	0.792712949 2860661	1
0.3727239443 495241	0.839900713 62085	0.821371610 8452944	0.788281634 6627277	1

0.3725652916 768368	0.825628296 6180577	0.821371610 8452944	0.789430494 0095189	1
0.3724188430 558946	0.812752094 3220602	0.819776714 5135568	0.788445757 4265551	1
0.3722601903 832073	0.802668321 4396524	0.820414673 046252	0.787296898 0797634	1
0.3721137417 622651	0.793205088 4269314	0.820414673 046252	0.786640407 0244544	1
0.3719550890 8957775	0.785138070 1210052	0.819457735 2472088	0.786804529 7882817	1
0.3718086404 686356	0.778312131 5544522	0.818181818 1818183	0.785491547 6776627	1
0.3716499877 9594827	0.772572137 759851	0.817862838 9154703	0.786476284 260627	1
0.3715035391 750061	0.767607818 802358	0.815629984 0510366	0.784342688 3308715	1
0.3713448865 023188	0.763108904 7471299	0.815311004 7846885	0.782373215 1649434	1
0.3711984378 813767	0.759695935 4638535	0.814992025 5183414	0.782537337 9287708	1
0.3710397852 0868935	0.756748371 0828421	0.814035087 7192981	0.782865583 4564251	1
0.3708933365 8774715	0.753645671 7344089	0.814035087 7192981	0.780731987 5266701	1

0.3707346839 150598	0.750387837 4185541	0.813397129 1866029	0.782373215 1649434	1
0.3705882352 9411766	0.747750542 9723859	0.813397129 1866029	0.781880846 8734613	1
0.3704295826 2143033	0.746044058 3307477	0.814354066 9856462	0.780239619 2351879	1
0.3702831340 004882	0.744337573 6891095	0.812440191 3875596	0.781224355 8181517	1
0.3701244813 2780085	0.742165684 1452062	0.810526315 7894731	0.778434268 8330872	1
0.3699780327 068587	0.740148929 5687248	0.811164274 3221692	0.779911373 7075332	1
0.3698193800 3417136	0.738442444 9270865	0.809569377 9904307	0.776300672 9033317	1
0.3696729314 132292	0.736580825 3180266	0.811483253 5885164	0.777941900 5416051	1
0.3695142787 405419	0.735339745 5786533	0.810526315 7894731	0.777777777 7777777	1
0.3693678301 195997	0.733788395 9044368	0.809888357 2567779	0.776957163 9586408	1
0.3692091774 469124	0.732702451 1324852	0.810845295 0558211	0.778598391 5969146	1
0.3690627288 2597024	0.731616506 3605335	0.809888357 2567779	0.776793041 1948138	1

0.3689040761 532829	0.729910021 7188954	0.807974481 6586922	0.775808304 6118496	1
0.3687576275 3234076	0.728979211 9143654	0.809888357 2567779	0.776957163 9586408	1
0.3685989748 596534	0.727893267 1424138	0.807336523 125997	0.773674708 6820941	1
0.3684525262 387113	0.725100837 7288239	0.801275917 0653903	0.770720498 9332022	1
0.3682938735 6602394	0.734253800 8067018	0.803827751 196172	0.770884621 6970291	1
0.3681474249 450818	0.747440273 0375427	0.800318979 2663479	0.771541112 7523386	1
0.3679887722 7239446	0.759851070 4312751	0.800956937 7990431	0.770392253 4055474	1
0.3678423236 514523	0.770089978 2811046	0.800956937 7990431	0.770556376 1693748	1
0.3676836709 78765	0.779708346 2612472	0.802551834 1307816	0.771212867 2246839	1
0.3675372223 578228	0.787775364 5671735	0.802551834 1307816	0.771705235 516166	1
0.3673785696 851355	0.794446168 1663046	0.801275917 0653903	0.770720498 9332022	1