

**PEMBELAJARAN MESIN PADA DATA *PREPROCESSING* DENGAN  
METODE *PRINCIPAL COMPONENT ANALYSIS* DAN SMOTE  
(*MACHINE LEARNING ON DATA PREPROCESSING WITH PRINCIPAL  
COMPONENT ANALYSIS AND SMOTE METHOD*)**

**TUGAS AKHIR**

Disusun sebagai syarat untuk memperoleh gelar Sarjana Teknik

Di Program Studi S1 Teknik Elektro

Disusun Oleh :

**GIOVANO TRIHADE PUTRA**

**1102184045**



**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2023**

**LEMBAR PENGESAHAN**  
**TUGAS AKHIR**  
**PEMBELAJARAN MESIN DENGAN MENGOLAH DATA**  
***PREPROCESSING DENGAN METODE PRINCIPAL COMPONENT***  
***ANALYSIS DAN SMOTE***  
***(MACHINE LEARNING ON DATA PREPROCESSING WITH PRINCIPAL***  
***COMPONENT ANALYSIS AND SMOTE METHOD)***

**Telah disetujui dan disahkan sebagai Tugas Akhir**

**Program Studi Teknik Elektro**

**Fakultas Teknik Elektro**

**Universitas Telkom**

**Disusun oleh :**

**GIOVANO TRIHADE PUTRA**

**1102184045**

**Bandung, Januari 2023**

Pembimbing I



**IG. Prasetya Dwi Wibawa, S.T., M.T.**

NIP. 14870043

Pembimbing II



**Dr. Meta Kallista, S.Si., M.Si**

NIP. 18890137

## LEMBAR PERNYATAAN ORISINALITAS

Nama : Giovano Trihade Putra  
NIM : 1102184045  
Alamat : Dago Asri Blok A/14, Bandung  
No. HP : +62 81270567982  
E-mail : [giovanoputra@student.telkomuniversity.ac.id](mailto:giovanoputra@student.telkomuniversity.ac.id)

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri, dengan judul :

**PEMBELAJARAN MESIN DENGAN MENGOLAH DATA  
*PREPROCESSING* DENGAN METODE *PRINCIPAL COMPONENT  
ANALYSIS* DAN SMOTE**

***(MACHINE LEARNING ON DATA PREPROCESSING WITH PRINCIPAL  
COMPONENT ANALYSIS AND SMOTE METHOD)***

Atas pernyataan ini, saya siap menanggung resiko / sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menjatuhkan ketidak aslian karya ini.



**Bandung, 23 Januari 2023**



**Giovano Trihade Putra**  
**1102184045**

## ABSTRAK

Pencemaran udara merupakan dampak negatif dari aktivitas manusia terhadap lingkungan. Udara yang tercemar oleh partikel yang berbahaya dapat membahayakan lingkungan, menyebabkan masalah pernafasan pada manusia, pemanasan global dan mempengaruhi metabolisme lingkungan. Indonesia menduduki peringkat ke-17 pada tahun 2022 dalam indeks kualitas udara (AQI) Internasional. Untuk menurunkan tingkat pencemaran udara, diperlukan adanya kesadaran masyarakat tentang pencemaran udara.

Kota Bandung adalah kota besar di Indonesia yang berlokasi dekat ibu kota negara. Lokasi geografis yang strategis membuat kota ini cocok untuk ditempati, namun juga memiliki tingkat populasi dan aktivitas industri yang tinggi, yang menyebabkan tingginya tingkat pencemaran udara. Walaupun Kota Bandung sudah memiliki sistem pemantauan kualitas udara, sistem ini hanya memantau parameter umum seperti ozon, monoksida karbon, dan partikulat. Oleh karena itu, perlu adanya kesadaran masyarakat tentang tingkat pencemaran udara dengan menggunakan Indeks Standar Pencemaran Udara (ISPU) sebagai acuan.

Dalam penelitian ini, penulis mengumpulkan data terkini sesuai dengan kebutuhan ISPU. Preprocessing digunakan untuk menentukan akurasi pembacaan kualitas udara dan memberikan prediksi kualitas udara kepada masyarakat. Dengan informasi ini, masyarakat dapat mengambil tindakan pencegahan untuk mengurangi produksi pencemaran.

**Kata Kunci:** *Polusi Udara, ISPU, Preprocessing, Akurasi, Prediksi.*

## **ABSTRACT**

*Air pollution is a negative impact of human activities on the environment. Contaminated air can be harmful to the environment, causing respiratory problems for humans, global warming and affecting the environment's metabolism. Indonesia ranks 17th in the world in the 2022 AQI (Air Quality Index) International rankings. To reduce air pollution, it is necessary to raise public awareness about air pollution.*

*The city of Bandung, Indonesia is a large city located near the capital. Its strategic geographical location makes it a suitable place to live, but also has a high population density and industrial activity, leading to high levels of air pollution. Although Bandung already has an air quality monitoring system, it only monitors general parameters such as ozone, carbon monoxide, and particulate matter. To increase public awareness, the city needs to use the Air Pollution Standard Index (ISPU) as a reference.*

*In this study, the author collected the most recent data according to the ISPU requirements. Preprocessing was used to determine the accuracy of air quality readings and provide air quality predictions to the public. With this information, the public can take precautions to reduce the production of pollution.*

**Keywords:** *Air Pollution, ISPU, Preprocessing, Accuracy, Prediction.*

## **KATA PENGANTAR**

Assalamualaikum Wr. Wb.

Puji serta syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat serta karunianya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Sistem Monitoring Polusi Udara Multi Titik Dengan Data Preprocessing”. Penyusunan Tugas Akhir ini sebagai salah satu syarat untuk menyelesaikan Pendidikan Strata 1 Program Studi Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan arahan serta masukan dalam penyusunan Tugas Akhir ini sehingga dapat berjalan dengan lancar. Perjalanan Panjang dalam rangka penyusunan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis memohon maaf atas segala kesalahan selama pengerjaan Tugas Akhir ini.

Penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi pembaca dan dapat menambah wawasan serta dapat dijadikan referensi bagi para pembaca dan juga penulis demi pengembangan kedepannya. Penulis mengharapkan kritik maupun saran yang membangun agar lebih baik dan menjadi bahan evaluasi penulis di kemudian hari.

Wassalamualaikum Wr. Wb.

## UCAPAN TERIMA KASIH

Puji syukur kepada Allah SWT penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Sistem Monitoring Polusi Udara Multi Titik Dengan Data Preprocessing”. Tugas Akhir ini dapat selesai karena dukungan doa, dan bimbingan yang diberikan kepada penulis. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua, Bapak Drs. Masri Hanus Damefriza , MA dan Ibu Indra Dewi, S.Pd serta Gladys selaku kakak, Gefry selaku abang kandung, membimbing penulis sehingga dapat menyelesaikan masa perkuliahan dan Tugas Akhir ini.
2. Bapak IG. Prasetya Dwi Wibawa, S.T., M.T. dan Ibu Dr. Meta Kallista, S.Si., M.Si. selaku dosen pembimbing 1 dan pembimbing 2 yang telah banyak membimbing dan membantu penulis dengan meluangkan waktu dukungan dan masukan yang telah diberikan kepada penulis dalam menyelesaikan Tugas Akhir ini.
3. Bapak Ir. Porman Pangaribuan selaku dosen wali, yang telah banyak membantu dan memberikan bimbingan serta arahan kepada penulis dari awal perkuliahan sampai saat ini.
4. Bapak dan Ibu dosen yang tidak dapat disebutkan namanya satu per satu, Terima kasih telah memberikan ilmu selama masa perkuliahan.
5. Partner Tugas Akhir saya, Steven Mulya Manik, Rafli Ananda, dan sepupu penulis Adhit, terima kasih atas segala bantuan dan tenaga yang diberikan pada penulis dari penelitian ini.
6. Teman-teman Keluarga TUNFC yang sudah memberikan saran ataupun masukan pada penelitian Tugas Akhir ini.
7. Teman-teman seperjuangan yang berada di Telkom University yang selalu memberikan dukungan motivasi dan moral selama di perkuliahan.
8. Teman-teman seperjuangan Almizan, Zein, Ricky, Guntur, Rafli Ananda dan warga bem kema griya bandung asri yang selalu menerima penulis untuk melakukan penelitian dan penulisan Tugas Akhir ini.
9. Keluarga besar yang tidak dapat saya sebut satu-persatu, terima kasih telah memberikan bantuan material maupun arahan selama perkuliahan.

10. Terakhir saya ingin berterima kasih kepada Audrey Anastasya Bella yang senantiasa memberikan dukungan motivasi dan semua usaha yang telah diberikan sehingga saya mendapatkan motivasi lebih besar.

Terima kasih kepada seluruh pihak lain yang tidak dapat disebutkan satu-satu, semoga Allah SWT dapat terbalaskan semua kebaikan yang telah diberikan kepada penulis.



## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN ORISIONALITAS .....	ii
ABSTRAK .....	iii
<i>ABSTRACT</i> .....	iv
KATA PENGANTAR .....	v
UCAPAN TERIMA KASIH .....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	vi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	1
1.3 Tujuan Masalah .....	2
1.4 Batasan Masalah .....	2
1.5 Metode Penelitian .....	2
1.6 Sistematika Penulisan .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1 Prinsip Kerja Solusi .....	5
2.2 Penelitian Sebelumnya .....	5
2.3 Kualitas Udara .....	9
2.3.1 <i>Partikulat (PM<sub>10</sub>)</i> .....	10
2.3.2 <i>Gas Karbon Monoksida (CO)</i> .....	11
2.3.3 <i>Gas Ozon (O<sub>3</sub>)</i> .....	11
2.3.4 <i>Gas Sulfur Dioksida (SO<sub>2</sub>)</i> .....	12
2.3.5 <i>Gas Natriun Diksida (NO<sub>2</sub>)</i> .....	13
2.4 Indeks Standar Pencemar Udara .....	14
2.5 Data Preprocessing .....	17
2.6.1 <i>Principal Component Analysis</i> .....	17
2.6.2 <i>Standard Scaller</i> .....	18
2.6.3 <i>Oversampling</i> .....	18
2.6.4 <i>SMOTE</i> .....	19
2.6.5 <i>Supervised Vector Machine (SVM)</i> .....	20

2.6.6	<i>Gradient Descent</i> .....	22
2.6.7	<i>Root Mean Squares Error</i> .....	22
BAB III PERANCANGAN SISTEM .....		24
3.1	Desain Sistem .....	24
3.2	Diagram Blok .....	24
3.2.1	<i>Spesifikasi Komponen</i> .....	25
3.3	Desain Perangkat Lunak.....	31
3.3.1	<i>Diagram Alir</i> .....	32
BAB IV HASIL DAN ANALISIS .....		35
4.1	Subscribe Software IoT .....	35
4.2	Hasil Analisis Metode Support Vector Machine.....	36
4.3	Hasil Analisis menggunakan metode Support Vector Machine dengan Principal Component Analysis .....	38
4.4	Hasil Analisis Menggunakan SMOTE .....	41
4.5	Perbandingan Pengujian Metode SVM, SVM dengan PCA dan SVM dengan SMOTE.....	44
4.6	Hasil Pengujian Prediksi Sensor.....	46
5.1	<b>Kesimpulan</b> .....	52
5.2	<b>Saran</b> .....	52
DAFTAR PUSTAKA .....		53
LAMPIRAN.....		56

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Desain Sistem Secara Keseluruhan .....	5
<b>Gambar 2. 3</b> Pembuatan Data Sintesis SMOTE .....	19
<b>Gambar 3. 1</b> Diagram Blok Monitoring Polusi Udara .....	24
<b>Gambar 3. 6</b> Diagram Pinout ESP32 DOIT DevKit v1 .....	25
<b>Gambar 3. 7</b> Modul Sensor MQ-7 .....	26
<b>Gambar 3. 8</b> Modul Sensor MQ-131 .....	27
<b>Gambar 3. 9</b> Modul Sensor Sharp GP2Y1010AU0F .....	28
<b>Gambar 3. 10</b> Modul Sensor MQ-136 .....	29
<b>Gambar 3. 11</b> Modul Sensor MiCS 6814 .....	30
<b>Gambar 3. 12</b> Diagram Alir Keseluruhan Sistem .....	32
<b>Gambar 3. 13</b> Diagram Alir Preprocessing .....	33
<b>Gambar 3. 14</b> Diagram Alir Prediksi Kualitas Udara .....	34
<b>Gambar 4. 1</b> Hasil Subscribe data IoT .....	35
<b>Gambar 4. 2</b> Penggunaan Parameter pada SVM .....	36
<b>Gambar 4. 3</b> Hasil Akurasi SVM .....	36
<b>Gambar 4. 4</b> Rata-rata Persentase Kesalahan Absolut .....	37
<b>Gambar 4. 5</b> Color Map Data ISPU .....	39
<b>Gambar 4. 6</b> Cumulative Variance .....	39
<b>Gambar 4. 7</b> Hasil Presentase SVM dan PCA .....	39
<b>Gambar 4. 8</b> Data Imbalance Sistem Polusi Udara .....	41
<b>Gambar 4. 9</b> Hasil Persentase SVM dan SMOTE .....	42
<b>Gambar 4. 10</b> Data Test Pembacaan Sensor .....	46
<b>Gambar 4. 11</b> Data Pengujian Sensor dengan AQI .....	47
<b>Gambar 4. 12</b> Line Graph Data Percobaan Sensor .....	48
<b>Gambar 4. 13</b> Grafik Prediksi Sensor .....	49
<b>Gambar 4. 14</b> Nilai Hasil Prediksi pada Pembelajaran Mesin .....	50
<b>Gambar 4. 15</b> Hasil Prediksi pada IoT .....	51
<b>Gambar 4. 16</b> Hasil Prediksi pada IoT (2) .....	51

## DAFTAR TABEL

<b>Tabel 2. 1</b> Perbandingan Dengan Penelitian Terdahulu .....	7
<b>Tabel 2. 2</b> Rentang Indeks Standar Pencemaran Udara [20].....	14
<b>Tabel 2. 3</b> Rentang Indeks Standar Pencemar Udara dengan Partikel Parameter Terkait [20].....	15
<b>Tabel 2. 4</b> Batas Indeks Standar Pencemar Udara Dalam SI [20].....	16
<b>Tabel 4. 3</b> Tabel Perbandingan Akurasi Setiap Metode .....	44

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Pencemaran merupakan dampak negatif dari suatu aktivitas terhadap lingkungan. Udara yang terkontaminasi oleh partikel tersebut dapat memberikan buruk bagi lingkungan, contohnya masalah pernafasan manusia, pemanasan global dan efek pada metabolisme lingkungan. Di Indonesia sendiri mendapatkan peringkat no.17 pada 2022 pada AQI (Air Quality Index) Internasional. Untuk menurunkan tingkat polusi udara ini diperlukan adanya kesadaran masyarakat tentang polusi udara.

Kota Bandung merupakan kota besar Indonesia yang posisinya berdekatan dengan ibu kota negara. Letak geografis yang strategis menjadikan kota ini cocok ditempati oleh masyarakat. Namun, hal ini juga menyebabkan banyak industri dan penduduk yang padat, yang berdampak pada tingginya tingkat polusi udara. Di kota Bandung sudah memiliki indikator pemantau kualitas udara, namun untuk parameter yang ditinjau hanya sebatas parameter umum seperti Ozon, Karbon Monoksida, dan Partikulat. Dengan hal ini dibutuhkan adanya kesadaran masyarakat dalam tingkat polusi udara dengan harapan masyarakat dapat mengurangi sedikit demi sedikit produksi polusi yang dihasilkan. Untuk parameter indikator yang digunakan berdasarkan Indeks Standar Polusi Udara (ISPU).

Pada penelitian ini penulis mengambil sampel dengan waktu terkini sesuai dengan kebutuhan pada ISPU. Untuk menemukan tingkat akurasi pembacaan ini dibutuhkan adanya pengolahan *preprocessing*. Dengan *preprocessing* dapat meningkatkan kinerja dari pembelajaran mesin, sehingga dapat menentukan kadar kualitas udara dan dapat memberikan prediksi kualitas udara kepada masyarakat untuk dapat berwaspada.

### **1.2 Rumusan Masalah**

Adapun rumusan masalah tugas akhir ini sebagai berikut:

1. Bagaimana metode pembelajaran mesin yang tepat pada tahap preprocessing untuk meningkatkan nilai akurasi dari PCA dan SMOTE?

### **1.3 Tujuan Masalah**

Adapun tujuan dan manfaat dari penelitian ini sebagai berikut:

1. Menganalisis metode pembelajaran mesin yang terbaik pada tahap preprocessing untuk meningkatkan nilai akurasi dari PCA dan SMOTE.

### **1.4 Batasan Masalah**

Adapun batasan masalah yang penulis rencanakan sebagai berikut:

1. Analisis kinerja sistem pemantauan polusi udara hanya difokuskan pada tingkat akurasi dan kehandalan sistem.
2. Evaluasi dilakukan hanya pada sistem pemantauan polusi udara yang telah selesai dibuat dan tidak termasuk proses pembuatannya.
3. Analisis kinerja dan evaluasi sistem pemantauan polusi udara hanya dilakukan pada kondisi operasi normal dan tidak termasuk kondisi darurat atau kondisi cuaca ekstrem.
4. Sistem pemantauan polusi udara yang dibahas hanya digunakan untuk lingkungan pemukiman saja, tidak termasuk lingkungan industri atau lingkungan lain.
5. Analisis kinerja sistem pemantauan polusi udara hanya dilakukan pada satu lokasi saja, tidak termasuk analisis kinerja sistem pemantauan polusi udara di lokasi lain.
6. Evaluasi yang dilakukan hanya berdasarkan data yang dihasilkan oleh sistem pemantauan polusi udara saja, tidak termasuk data dari sumber lain.
7. Analisis kinerja dan evaluasi sistem pemantauan polusi udara hanya dilakukan dalam jangka waktu yang singkat, tidak termasuk analisis kinerja sistem dalam jangka waktu yang panjang.

### **1.5 Metode Penelitian**

Metode penelitian pada tugas akhir ini sebagai berikut:

1. Studi Literatur

Studi literatur digunakan untuk mencari untuk mencari ilmu pengetahuan yang terkait penelitian sehingga dapat menunjang

penelitian. Referensi yang digunakan adalah jurnal, buku, diskusi daring, lembar data, dan sumber lainnya di internet.

## 2. Perancangan

Perancangan dengan membuat diagram blok, *flowchart* sistem perangkat lunak dan perangkat keras untuk menunjang pengerjaan tugas akhir.

## 3. Implementasi

Menguji alat yang telah dibuat untuk mengetahui apakah sistem bekerja sesuai yang diharapkan dan data yang didapat digunakan untuk di analisis.

### 1.6 Sistematika Penulisan

Tugas Akhir ini dibagi dalam lima bab pembahasan yang disusun sebagai berikut:

#### **BAB I PENDAHULUAN**

Berisi mengenai latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan metode penelitian.

#### **BAB II TINJAUAN PUSTAKA**

Berisi mengenai desain konsep solusi dan kajian permasalahan pada Tugas Akhir.

#### **BAB III PERANCANGAN SISTEM**

Berisi mengenai deskripsi sistem yang dibuat digambarkan melalui diagram blok dan diagram alir. Bab ini juga berisi mengenai desain sistem, desain perangkat keras, dan desain perangkat lunak.

#### **BAB IV ANALISIS DAN PENGUJIAN**

Berisi mengenai hasil dan analisa dari pengujian yang dilakukan pada Tugas Akhir.

#### **BAB V KESIMPULAN DAN SARAN**

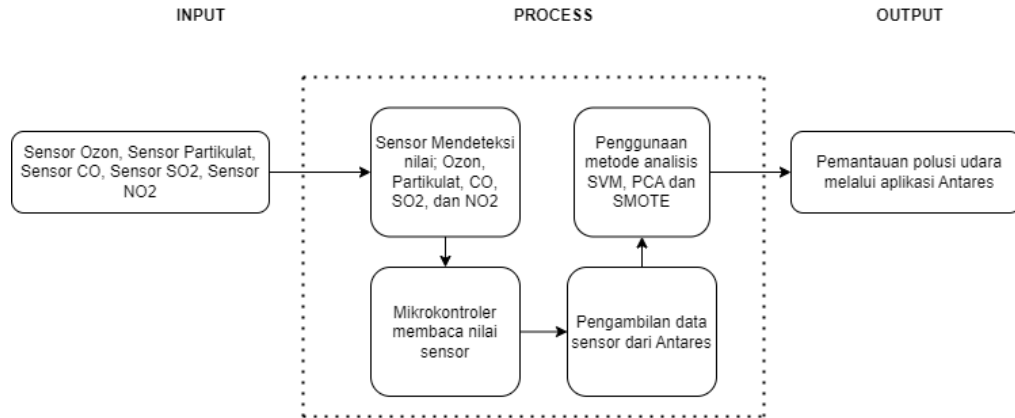
Berisi mengenai kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian berikutnya.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Prinsip Kerja Solusi



**Gambar 2. 1** Desain Sistem Secara Keseluruhan

Pada pengerjaan Tugas Akhir ini akan dirancang sebuah sistem monitoring polusi udara dengan menggunakan beberapa sensor, yaitu; MQ7, MQ131, MICS6814, MQ136 dan Sharp GP2Y1010AU0F. Tugas Akhir ini akan fokus membahas rancang bangun sistem monitoring polusi udara dengan sistem dengan konsentrasi pada pre-processing yang seperti pada gambar 2.1. dengan hasil preprocessing akan memberikan dampak yang lebih baik pada hasil pengolahan pembelajaran mesin. Penggunaan pengolahan preprocessing juga akan memberikan data yang lebih teratur, serta dengan penggunaan preprocessing memberikan hasil pengolahan metode yang akan memberikan hasil prediksi yang jelas dengan label pada setiap nilai yang di dapatkan.

#### 2.2 Penelitian Sebelumnya

Sistem monitoring polusi udara bukanlah hal yang baru. Pasalnya sudah banyak penelitian sebelumnya yang telah melakukan penelitian di bidang ini. Sensor yang beragam serta pemilihan gas yang diukur menjadikan penelitian ini berbeda dengan penelitian sebelumnya. Banyak penelitian monitoring yang menggunakan standar berbeda, hal ini memberikan pengaruh tidak adanya standar polusi udara internasional. Namun, akibat dari berbagai macam penelitian ini menjadikan setiap negara memiliki standar pengukuran yang berbeda. Tabel 2.1

menjelaskan perbandingan metode dan kekurangan pada setiap topik penelitian terdahulu.

**Tabel 2. 1** Perbandingan Dengan Penelitian Terdahulu

Referensi	Judul	Penulis	Metode	Kekurangan
[1]	<i>Monitoring Polusi Udara Menggunakan Mikrokontroler Hemat Daya Multi Titik Dengan Jaringan Mesh Nirkabel.</i>	Nadhif Muhammad Fauzan (2022)	Jaringan komunikasi Mesh Polutan : CO, O <sub>3</sub> dan PM <sub>10</sub> . Jaringan sensor nirkabel dengan jangkauan luas.	Penelitian hanya mencakup CO, PM <sub>10</sub> , dan O <sub>3</sub> hanya terfokus ke ketiga partikel.
[2]	<i>Monitoring dan Klasifikasi Polusi Udara Menggunakan Random Forest dengan Imbalanced Dataset</i>	Andri Aryarasyid Dharmasaputro (2022)	Monitoring kualitas udara di kota besar, penggunaan Random Forest, data preprocessing menggunakan MICE dan SMOTE dengan performasi G-Mean	
[3]	<i>Pengukuran Parameter Kualitas Udara (CO, NO<sub>2</sub>, SO<sub>2</sub>, O<sub>3</sub>, dan PM<sub>10</sub>) di</i>	Agusta Kurniawan (2017)	Polutan: CO, NO <sub>2</sub> , SO <sub>2</sub> , O <sub>3</sub> , dan PM <sub>10</sub> Database: Raspberry Pi3 Tampilan: Web	Tidak menjelaskan desain rangkaian dan pengaruh dari ruang lingkup penelitian sensor.

	<i>Bukit Kotatabang Berbasis ISPU</i>		Server	
[4]	<i>Real-time air pollution monitoring with sensors on city bus</i>	Sami Kaivonen, Edith C.-H. Ngai (2019)	Sensor yang diukur meliputi methane (CH <sub>4</sub> ), ethylene (C <sub>2</sub> H <sub>4</sub> ), ammonia (NH <sub>3</sub> ), benzene (C <sub>7</sub> H <sub>8</sub> ), LPG (C <sub>4</sub> H <sub>10</sub> ), CO <sub>2</sub> , CO and nitrogen oxides (NO <sub>x</sub> )	Penelitian menggunakan boardband mobile network, dimana adanya packet loss yang cukup tinggi.
[5]	<i>Health Effects of Ambient Air Pollution in Developing Countries</i>	Pier Mannuccio Mannucci, and Massimo Franchini (2017)	Survei tentang efek kesehatan polutan partikulat PM <sub>10</sub> , PM <sub>2.5</sub> , gas NO <sub>2</sub> , CO, dan SO <sub>2</sub> di area rural di negara-negara berkembang.	Tidak spesifik menjelaskan lokasi ruang survei. Penelitian membutuhkan daya yang terhubung langsung.
[6]	<i>Data Preprocessing and Intelligent Data Analysis</i>	A. Famili, Wei-Min Shen, Richard Weber, Evangelos Simoudis (1996)	Penelitian meliputi gambaran umum preproses data, fokus pada masalah yang muncul pada dunia nyata.	Contoh makalah berasal dari domain tertentu (manufaktur semikonduktor dan aerospace) sehingga hasil mungkin tidak dapat diterapkan

			Memberikan penjelasan 14 teknik preprocessing secara terperinci.	pada industri lainnya. Dan membahas tantangan dalam bidang, yang berarti penelitian mungkin tidak memiliki hasil eksperimental atau empiris untuk mendukung klaimnya.
--	--	--	--	---

Monitoring kualitas udara sudah banyak dilakukan penelitian sebelumnya dengan berbagai macam standar AQI (Air Quality Index) yang berbeda. Namun sudah ada penelitian sebelumnya yang sudah menggunakan ISPU sebagai AQI sehingga sudah ada kemajuan dari penelitian ini. Untuk penelitian kali ini menggunakan topologi jaringan star nirkabel menjadi pengembangan dari penelitian yang menggunakan jaringan sensor nirkabel yang telah ada sebelumnya.

### 2.3 Kualitas Udara

Kualitas udara merupakan suatu takaran keadaan udara pada atmosfer di mana manusia hidup dan beraktivitas. Contoh dari gas oksigen (O<sub>2</sub>) dan karbon dioksida (CO<sub>2</sub>) dengan konsentrasi tertentu akan sangat penting untuk perkembangan kehidupan. Rata-rata penduduk Indonesia dapat diperkirakan akan kehilangan 1,2 tahun ekspektasi harapan kehidupan dengan tingkat polusi saat ini, menurut Air Quality Life Index (AQLI) karena kualitas udara gagal memenuhi pedoman Organisasi Kesehatan Dunia (WHO) [7][8]. Ketika pemerintah menyadari masalah kualitas udara, AQLI menunjukkan bahwa Indonesia memiliki peluang untuk memperoleh keuntungan yang sangat besar untuk memperhatikan kualitas udara di Indonesia [8].

Pemerintah menerapkan suatu cara untuk membuat regulasi untuk mengatur keadaan kualitas udara yang terbaik untuk kehidupan manusia melalui Keputusan

Menteri Negara Lingkungan Hidup Nomor : KEP 45/MENLH/1997 Tentang Indeks Standar Pencemaran Udara telah memiliki standar yang disebut ISPU (Indeks Standar Pencemaran Udara) [9] . Situasi yang tidak memenuhi standar yang ditetapkan dianggap sebagai kondisi yang dapat memiliki berbagai efek yang tidak diinginkan pada banyak organisme di planet ini, termasuk manusia dan tumbuhan.

### 2.3.1 Partikulat (PM<sub>10</sub>)

Partikulat Material 10 biasa disebut dengan PM<sub>10</sub> merupakan partikel pada udara yang memiliki ukuran 10 $\mu$ m. Partikel tersebut berada di udara dalam waktu yang relative lama dalam keadaan melayang-layang dan mengakibatkan masuknya ke tubuh manusia [7]. Organisasi Kesehatan Dunia (WHO, 2000) menetapkan batasan yang ketat, untuk di Indonesia di angka 20 g/m<sup>3</sup> untuk rata-rata tahunan dan 150g/m<sup>3</sup> untuk batasan rata-rata harian. (DOE Indonesia, 1999) [10].

PM<sub>10</sub> dipengaruhi oleh sumber yang bergantung pada musim emisi dan factor meteorologi seperti angin dan curah hujan. Fak Saat ini, lebih dari 600 juta orang yang tinggal di daerah perkotaan di seluruh dunia terpapar pada tingkat berbahaya dari polusi udara yang dihasilkan oleh produksi pabrik tambang, pabrik produksi kebutuhan, kendaraan bermotor, dan lainnya.

#### United State Air Quality Index Range

US AQI	US-EPA range (24 jam)	
	PM <sub>10</sub> (ug/m <sup>3</sup> )	PM <sub>2.5</sub> (ug/m <sup>3</sup> )
0-50	0-54	0-12.0
51-100	55-154	12.1-35.4
101-150	155-254	35.5-55.4
151-200	255-354	55.5-150.4
201-300	355-424	150.5-250.4
301-500	425-604	250.5-500.4

### 2.3.2 Gas Karbon Monoksida (CO)

Selama beberapa decade terakhir, Karbon Monoksida telah menerima banyak perhatian dari ilmuwan karena memiliki pengaruh buruk terhadap kehidupan. Di Pakistan, volume terbesar yang diproduksi oleh CO merupakan dari pabrik industri dan kendaraan. CO dapat menyubstitusi oksigen di hemoglobin yang mengakibatkan masalah kesehatan seperti memburuknya tingkat kardiovaskular dan sistem saraf pusat [11]. Secara umum, sekitar 20% dari penilaian observasi tentang CO sudah melampaui batas aman dari NEQS (National Environmental Quality Standards). Itu menjadi catatan untuk semua nilai perhitungan atas semua studi yang diadakan [12].

Karbon Monoksida (CO) adalah hasil dari pembakaran tidak sempurna. Sumber dalam ruangan termasuk pemanas dan kompor, asap tembakau, dan lilin. CO bersifat racun bagi manusia. Udara normalnya mengandung sekitar 1 ppm karbon monoksida. Di jalan raya yang cukup sibuk, kandungan mol CO bisa berada di kisaran 10-20 ppm. Tingkat dalam ruangan umumnya mengikuti tingkat luar ruangan kecuali di ruangan berventilasi buruk dengan sumber CO dalam ruangan. Sekitar 600 kematian akibat kecelakaan akibat keracunan karbon monoksida dilaporkan setiap tahun di Amerika Serikat dan jumlah kematian terkait karbon monoksida yang disengaja adalah 5 sampai 10 kali lebih tinggi dari kematian oleh kecelakaan kendaraan bermotor.

### 2.3.3 Gas Ozon (O<sub>3</sub>)

Ozon (O<sub>3</sub>) merupakan suatu molekul triatomik yang terdiri dari tiga molekul oksigen yang memiliki sifat yang sangat reaktif dibandingkan oksigen. Berat dari molekul ini yaitu 47,98 g/mol dan secara termodinamika senyawa yang tidak stabil, yang bergantung pada kondisi sistem seperti suhu dan tekanan [13]. Molekul ini sangat reaktif dengan bahan kimia dan sistem biologis lainnya. Gas ini dikenal sebagai polutan udara biasa. Ozon merupakan polutan udara yang tidak langsung dipancarkan dari industri kecuali dalam keadaan tertentu [11].

Banyak pengaruh buruk dari penipisan kadar ozon di udara, seperti pada bidang kesehatan meningkatnya korban penyakit kanker kulit, jaringan organ

penglihatan, tingkat kekebalan dan peluang terjadinya perubahan genetik yang terjadi pada manusia [14].

US AQI	US-EPA Range CO (ppm)
0-50	0-4.4
51-100	4.5-9.4
101-150	9.5-12.4
151-200	12.5-15.4
201-300	15.5-30.4
301-500	30.5-50.4

#### 2.3.4 Gas Sulfur Dioksida (SO<sub>2</sub>)

Sulfur Dioksida (SO<sub>2</sub>) adalah bahan kimia komoditas yang umum ditemukan pada produksi pembakaran. Sulfur Dioksida dihasilkan dari pembakaran yang mengandung unsur belerang, seperti batu bara dan minyak, mengalami pembakaran dan selama penyulingan minyak bumi. SO<sub>2</sub> adalah gas beracun, berbau busuk menyengat, mudah larut dalam air dan memiliki titik didih sekitar -10°C [15]. Paparan sulfur dioksida sudah banyak dikaitkan dalam bidang kesehatan dengan berbagai gejala termasuk dispnea, batuk dan juga telah terlibat dalam memperparah penyakit kardiovaskular dan pernapasan yang sudah ada sebelumnya [16].

Banyak studi mendukung penurunan SO<sub>2</sub> tentang bidang pernapasan. Sheppard dkk<sup>59</sup> ditemukan bahwa konsentrasi SO<sub>2</sub> 0,05 ppm dan 0,25 ppm menyebabkan peningkatan resistansi pada penderita asma.[16] Studi lain penyelidik yang sama menemukan bahwa peningkatan hambatan jalan napas ini dapat dilemahkan dengan pretreatment cromolyn [16].

US AQI	US-EPA Range for SO <sub>2</sub> (ppb) (1 Hour)
0-50	0-35
51-100	36-75
101-150	76-185
151-200	186-304



201-300	305-604 (24 hours)
301-500	605-1004 (24 hours)

### 2.3.5 Gas Natrium Dioksida (NO<sub>2</sub>)

Nitrogen Dioksida (NO<sub>2</sub>) adalah polutan atmosfer dimana-mana karena tingkat sebaran luas dari sumber alami dan antropogenik yang dapat terjadi dengan sengaja ataupun tidak sengaja, dan dapat menjadi iritasi pernapasan ketika dihirup pada konsentrasi tinggi. Nitrogen dioksida merupakan masalah lingkungan yang serius di daerah yang sangat terkontaminasi. NO<sub>2</sub> diproduksi selama proses pembakaran bahan bakar, misalnya di mobil, menyebabkan berbagai macam kesehatan dan dampak lingkungan, seperti pembentukan ozon troposfer dan kabut asap perkotaan melalui reaksi fotokimia dengan hidrokarbon. Selanjutnya, Natrium dioksida (NO<sub>2</sub>) bersama dengan Sulfur Dioksida (SO<sub>2</sub>) adalah kontribusi utama untuk “hujan asam” [17].

Saat ini, ada beberapa standar nasional dan internasional yang tersedia untuk mengukur efisiensi oksidasi fotokatalitik bahan semikonduktor. Beberapa dari penelitian tersebut menggunakan Natrium Oksida /NO<sub>x</sub> (NO ataupun NO<sub>2</sub>) untuk melakukan pengujian, tetapi pengujian dilakukan dengan mengintegrasikan cahaya yang mengakibatkan terjadinya reaksi homogen dan fotolisis NO<sub>x</sub> terjadi [18][16].

US AQI	US-EPA Range for NO <sub>2</sub> (ppb) (1 Hour)
0-50	0-53
51-100	54-100
101-150	101-360
151-200	361-649
201-300	650-1249
301-500	1250-2049

## 2.4 Indeks Standar Pencemar Udara

Indeks Standar Pencemaran Udara merupakan suatu angka tersusun yang tidak mempunyai satuan yang menggambarkan kondisi udara ambien di lokasi dan waktu dan waktu tertentu yang didasarkan kepada dampak terhadap kesehatan manusia, nilai estetika dan makhluk hidup lainnya.[19] Secara resmi Indeks tersebut telah diatur dalam Keputusan Menteri Negara Lingkungan Hidup Nomor KEP-45/MENLH/10/1997 tentang Indeks Standar Pencemar Udara.[19] Pada Pasal 2 ayat 2 dalam peraturan tersebut menjelaskan bahwa Indeks Standar Pencemaran Udara (ISPU) ditetapkan dengan cara mengubah kadar pencemar udara yang terukur menjadi suatu angka yang tidak berdimensi. [19] Rentang Indeks Standar Pencemar Udara, dapat dilihat pada tabel 2.2.

**Tabel 2. 2** Rentang Indeks Standar Pencemaran Udara [19]

KATEGORI	RENTANG	PENJELASAN
Baik	0-50	Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan dan tidak berpengaruh pada tumbuhan, bangunan atau nilai estetika.
Sedang	51-100	Tingkat kualitas udara yang tidak berpengaruh pada kesehatan manusia ataupun hewan tetapi berpengaruh pada tumbuhan yang sensitif, dan nilai estetika.
Tidak sehat	101-199	Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika.
Sangat tidak sehat	200-299	Tingkat kualitas udara yang dapat merugikan kesehatan pada sejumlah segmen populasi yang terpapar.

Berbahaya	300-Lebih	Tingkat kualitas udara berbahaya yang secara umum dapat merugikan kesehatan yang serius.
-----------	-----------	--

Parameter yang digunakan sesuai ISPU untuk mengukur tingkat pencemaran udara adalah:

- Partikulat (PM<sub>10</sub>)
- Karbondioksida (CO)
- Sulfur dioksida (SO<sub>2</sub>).
- Nitrogen dioksida (NO<sub>2</sub>).
- Ozon (O<sub>3</sub>)

**Tabel 2.3** Rentang Indeks Standar Pencemar Udara dengan Partikel Parameter Terkait [19]

Rentang	Carbon Monoksida (CO)	Nitrogen (NO <sub>2</sub> )	Ozon (O <sub>3</sub> )	Sulfur Dioksida (SO <sub>2</sub> )	Partikulat (PM <sub>10</sub> )
Baik	Tidak ada efek	Sedikit berbau	Luka pada spesies tumbuhan akibat kombinasi dengan SO <sub>2</sub> (selama 4 jam)	Luka pada beberapa spesies tumbuhan akibat kombinasi dengan O <sub>3</sub> (selama 4 jam)	Tidak ada efek
Sedang	Perubahan kimia darah tapi tidak terdeteksi	Berbau	Luka pada beberapa spesies tumbuhan	Luka pada beberapa spesies tumbuhan	Terjadi penurunan pada jarak pandang
Tidak Sehat	Peningkatan pada kardiovaskular	Bau dan kehilangan warna.	Penurunan kemampuan	Bau, meningkatnya	Jarak pandang turun dan

	pada perokok yang sakit jantung	Peningkatan reaktivitas pembuluh tenggorokan pada penderita asma.	pada atlet yang berla	kerusakan tanaman	terjadi pengotoran debu di mana-mana
--	---------------------------------	---	-----------------------	-------------------	--------------------------------------

Angka indeks pencemar udara pada tiap sampel gas dan partikelnya dapat dilihat pada tabel 2.3.

**Tabel 2. 4** Batas Indeks Standar Pencemar Udara Dalam SI [19]

Indeks Standar Pencemar Udara	24 jam PM10 $\mu\text{g}/\text{m}^3$	24 jam SO <sub>2</sub> $\mu\text{g}/\text{m}^3$	8 jam CO $\mu\text{g}/\text{m}^3$	1 jam O <sub>3</sub> $\mu\text{g}/\text{m}^3$	1 jam NO <sub>2</sub> $\mu\text{g}/\text{m}^3$
50	50	80	5	120	80
100	150	365	10	235	365
200	350	800	17	400	1130
300	420	1600	34	800	2260
400	500	2100	46	1000	3000
500	600	2620	57.5	1200	3750

Keterangan :

(1) Pada 25 C dan 760 mmHg.

(2) Tidak ada isndeks yang dapat dilaporkan pada konsentrasi rendah dengan jangka pemaparan yang pendek.

## 2.5 Data Preprocessing

Data merupakan suatu nilai, hasil, angka yang dapat diolah menjadi sesuatu. Data yang dimaksud merupakan suatu nilai hasil dari pembacaan sensor yang telah dilakukan pengujian pembacaan sensor. Data preprocessing merupakan hal yang penting karena dapat mengurangi data spike karena nilai yang diberikan sensor kadangkala dapat berubah terlalu rendah sepersekian detik akibat dari pengaruh internal maupun eksternal dari komponen. Akibat dari efek spike ini flow data yang dibaca oleh skematik sensor tidak sempurna. Dari preprocessing ini menggunakan dua metode yaitu dengan cara mereduksi data dan upscaling data. Reduksi merupakan suatu cara yang digunakan untuk mengurangi nilai yang tidak diinginkan sehingga hasil yang diberikan akan terlihat lebih sempurna. Upscaling merupakan suatu cara memberikan data yang kosong atau tidak terbaca dengan memasukkan nilai random dari jangkauan pembacaan sensor.

### 2.6.1 Principal Component Analysis

Analisis komponen utama (PCA) merupakan suatu teknik analisis data multivariable yang digunakan untuk mengurangi dimensi data set. Hal ini dilakukan dengan menemukan suatu set variabel baru yang tidak berkorelasi, yang disebut dengan komponen utama, yaitu komponen yang menjelaskan mayoritas variasi dalam data asli. Komponen utama pertama (first principal component) merupakan arah dimana data paling bervariasi. Komponen utama kedua (second principal component) merupakan arah yang menjelaskan varian paling banyak yang orthogonal (tegak lurus) dengan komponen utama pertama, dan seterusnya. Komponen utama ini dapat dianggap sebagai perangkat sumbu baru yang sejajar dengan arah varian maksimum dalam data.

PCA suatu metode pengembangan dari teori Karhunen-Loève Transform (KLT) yang merupakan transformasi linear pada pengolahan citra. Pada praktiknya, metode PCA dan KLT dapat dikatakan sebagai metode yang sama[20]. Perhitungan proses PCA dalam prosesnya menggunakan persamaan berikut:

$$Y = \phi^T X \quad (1)$$

Dimana:

$X$  = matriks dimensi  $N \times M$  dengan  $X = [x_1, \dots, x_m]^T$

$\Theta = N \times M$  matriks orthogonal berbasis vector

$Y = N \times M$  koefisien weighting dari matriks

$N$  = jumlah sampel data

$M$  = jumlah fitur data

### 2.6.2 Standard Scaller

Standard scaller merupakan suatu library pada sklearn preprocessing yang bertujuan sebagai normalisasi data untuk menghasilkan akurasi yang akurat. Metode ini akan mengubah setiap fitur pada data menjadi skala yang sama dengan membuat rata-rata setiap fitur menjadi 0 dan standar deviasi menjadi 1. Hal ini dilakukan dengan mengurangi setiap nilai fitur dengan rata-rata fitur dan kemudian membagi dengan standar deviasi fitur. Metode ini berguna untuk mengatasi perbedaan skala pada fitur-fitur yang berbeda sehingga algoritma pembelajaran mesin dapat bekerja lebih baik. Variabel standard scaler akan memanggil kelas `StandardScaler()`, fungsi `scaler.fit(X_train)` akan digunakan untuk melatih data pada parameter `X_train`.

### 2.6.3 Oversampling

Oversampling adalah teknik yang digunakan dalam pembelajaran mesin untuk menangani dataset yang tidak seimbang. Dalam dataset yang tidak seimbang, kelas minoritas (kelas dengan contoh yang lebih sedikit) sering kali lebih banyak daripada kelas mayoritas. Hal ini dapat menyebabkan bias pada model, karena model akan dilatih untuk memprediksi kelas mayoritas dengan lebih akurat. Oversampling mengatasi masalah ini dengan meningkatkan jumlah contoh kelas minoritas dalam dataset. Hal ini dapat dilakukan dengan menduplikasi contoh kelas minoritas yang sudah ada, atau dengan membuat contoh sintetis dari kelas minoritas menggunakan teknik seperti SMOTE (Synthetic Minority Over-sampling Technique).

Tujuan dari oversampling adalah untuk menyeimbangkan dataset, sehingga model dilatih pada distribusi kelas yang lebih merata. Hal ini dapat

meningkatkan performa pada kelas minoritas, karena model akan terpapar pada lebih banyak contoh kelas tersebut selama pelatihan. Namun, oversampling juga dapat menyebabkan overfitting, karena model dapat menjadi terlalu terbiasa dengan contoh kelas minoritas yang terlalu banyak diambil, dan berkinerja buruk pada contoh-contoh baru yang belum pernah dilihat.

#### 2.6.4 SMOTE

Synthetic Minority Oversampling Technieque (SMOTE) merupakan suatu algoritma pada preprocessing yang biasa dilakukan pada suatu data yang tidak seimbang. Hal ini disebabkan oleh kesederhanaan dalam desain prosedur dan ketangguhan ketika diterapkan pada berbagai jenis masalah. Smote telah terbukti berhasil dalam berbagai aplikasi dari beberapa domain yang berbeda. SMOTE juga telah menginspirasi beberapa pendekatan untuk mengatasi masalah ketidakseimbangan kelas, dan juga secara signifikan berkontribusi pada paradigma pembelajaran baru yang diawasi, termasuk klasifikasi multilabel, pembelajaran in-cremental, pembelajaran semi-supervised, pembelajaran multi-instance, dan lain-lain.

Perhitungan dilakukan dengan rumus berikut[21]:

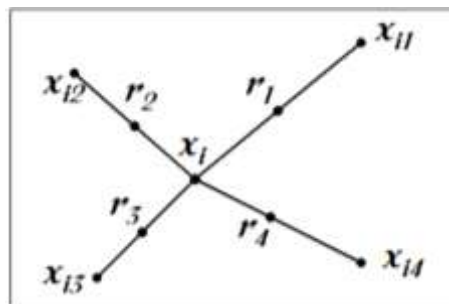
$$x_{gen} = x + \alpha \cdot (x' - x) \quad (1)$$

$x_{gen}$  = data yang dibuat

$x$  = data pada kelas minoritas

$x_1$  = data nearest neighbour dari  $x$

$\alpha$  = nilai random antara 0 hingga 1.



**Gambar 2. 2** Pembuatan Data Sintesis SMOTE

Nilai  $k$  untuk menentukan nearest neighbour harus ditentukan karena memiliki pengaruh yang besar terhadap data sintesis. Semakin besar nilai  $k$ , maka kemungkinan membuat data sintesis yang noisy akan semakin tinggi, namun di sisi lain, nilai  $k$  yang semakin kecil akan menyebabkan data sintesis hanya berkumpul pada beberapa titik tertentu.

### 2.6.5 Supervised Vector Machine (SVM)

Support Vector Machine (SVM) adalah algoritma pembelajaran terawasi yang dapat digunakan untuk tugas klasifikasi dan regresi. Ide dasar di balik SVM adalah menemukan batas keputusan, atau hyperplane, yang memisahkan kelas-kelas yang berbeda dalam kumpulan data. Batas keputusan dipilih sedemikian rupa sehingga memaksimalkan margin, atau jarak, antara contoh terdekat dari kelas-kelas yang berbeda. Batas keputusan direpresentasikan sebagai persamaan linear dalam ruang multi-dimensi. Sebagai contoh, dalam ruang dua dimensi, batas keputusan adalah sebuah garis, sedangkan dalam ruang tiga dimensi adalah sebuah bidang. Namun, dalam praktiknya, data sering kali tidak dapat dipisahkan secara linier, yang berarti bahwa batas keputusan linier tidak dapat ditemukan untuk memisahkan kelas-kelas secara sempurna. Untuk mengatasi hal ini, SVM memperkenalkan konsep "kernel trick" yang memungkinkan untuk mengubah data menjadi ruang dimensi yang lebih tinggi dimana kelas-kelasnya dapat dipisahkan secara linear.

Konsep SVM dapat disederhanakan dengan tujuan utama yaitu mencari hyperlane terbaik yang berfungsi sebagai pemisah dua buah class. Hyperlane pemisah terbaik dapat ditemukan dengan mengukur margin hyperlane dan mencari titik maksimalnya. Margin ini merupakan jarak antar hyperlane dengan pattern terdekat dari masing-masing class. Pattern yang paling dekat ini disebut sebagai support vector.

SVM memiliki perhitungan sebagai berikut:

$$\vec{x}_i \in \mathbb{R}^d$$

Data yang tersedia dinotasikan sebagai  $\vec{x}_i$  sedangkan label masing-masing label masing-masing dinotasikan  $y_i \in \{-1, +1\}$  untuk  $i = 1, 2, \dots, n$



$1, 2, \dots, l$ , yang mana  $l$  adalah banyak data. Diasumsikan kedua class -1 dan +1 dapat terpisah secara sempurna oleh hyperlane berdimensi  $d$ , yang didefinisikan

$$\vec{w} \cdot \vec{x} + b = 0 \quad (2)$$

$\vec{x}_i$  Pattern yang termasuk class -1 (sampel negatif) dapat dirumuskan sebagai pattern yang memenuhi pertidaksamaan

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \quad (3)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperlane dan titik terdekat  $\|\vec{w}\|$ , yaitu . Hal ini dapat dirumuskan sebagai Quadratic Programming (QP) problem, yaitu mencari titik minimal persamaan (4), dengan memperhatikan constraint persamaan (5).

$$\min_{\vec{w}} \tau(\vec{w}) = \frac{1}{2} \|\vec{w}\|^2 \quad (4)$$

$$y_i (\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0, \quad \forall i \quad (5)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya Lagrange Multiplier.

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\vec{x}_i \cdot \vec{w} + b) - 1) \quad (6)$$

$\alpha_i$  adalah Lagrange multipliers, yang bernilai nol atau positif ( $\alpha_i \geq 0$ ). Nilai optimal dari persamaan (6) dapat dihitung dengan meminimalkan  $L$  terhadap  $w$  dan  $b$ , dan memaksimalkan  $L$  terhadap  $\alpha_i$ . Dengan memperhatikan sifat bahwa pada titik optimal gradient  $L = 0$ , persamaan (6) dapat dimodifikasi sebagai maksimalisasi problem yang hanya mengandung saja  $\alpha_i$ , sebagaimana persamaan (7) di bawah.

Maximize:

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i x_j \quad (7)$$

Subject to:

$$\alpha_i \geq 0 \quad (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i y_i = 0$$

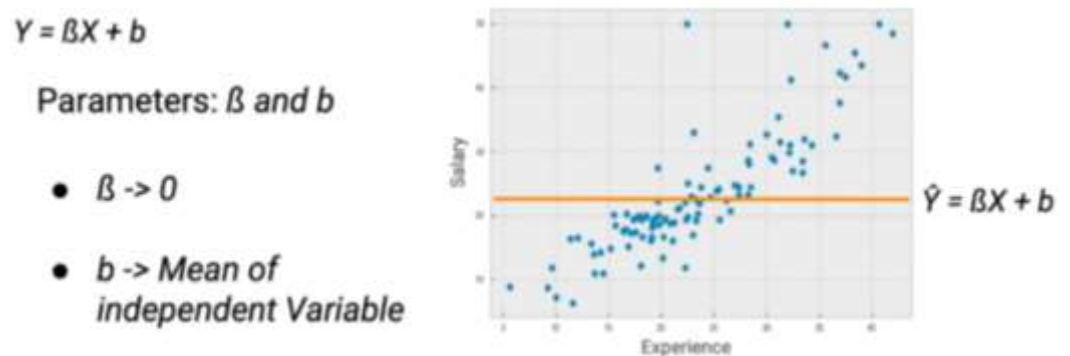
(8)

Dari hasil dari perhitungan ini diperoleh  $\alpha_i$  kebanyakan bernilai positif. Data yang berkorelasi dengan  $\alpha_i$  yang positif inilah yang disebut dengan support vector.[22]

### 2.6.6 Gradient Descent

Gradient descent merupakan suatu metode algoritma optimasi yang digunakan untuk meminimalkan fungsi kerugian dengan memperbarui parameter model secara berulang-ulang ke arah penurunan paling curam dari fungsi kerugian. Algoritma ini dimulai dengan satu set parameter awal dan menghitung gradien fungsi kerugian sehubungan dengan parameter tersebut. Gradien tersebut kemudian digunakan untuk memperbarui parameter dengan cara mengurangi nilai fungsi kerugian. Proses ini diulangi sampai nilai minimum dari fungsi kerugian tercapai atau kriteria penghentian terpenuhi.

Rumus Gradient Descent pada Regresi Linear[23] :



### 2.6.7 Root Mean Squares Error

Kesalahan kuadrat rata-rata adalah perbedaan kuadrat rata-rata antara nilai estimasi dan nilai aktual. Suatu model prakiraan memiliki kualitas yang baik jika memiliki nilai RMSE yang rendah, yang menandakan bahwa perbedaan antara nilai yang diprediksi dan nilai yang sebenarnya sangat dekat. RMSE mengukur tingkat ketidakcocokan antara dua set nilai, dan semakin kecil nilai RMSE, semakin baik kualitas model prakiraan tersebut.

$$\text{RMSE} = \left( \frac{\sum (y_i - \hat{y}_i)^2}{n} \right)^{1/2}$$

Keterangan:

RMSE = nilai root mean square error

$y$  = nilai hasil observasi

$\hat{y}$  = nilai hasil prediksi

$i$  = urutan data pada database

$n$  = jumlah data

## BAB III

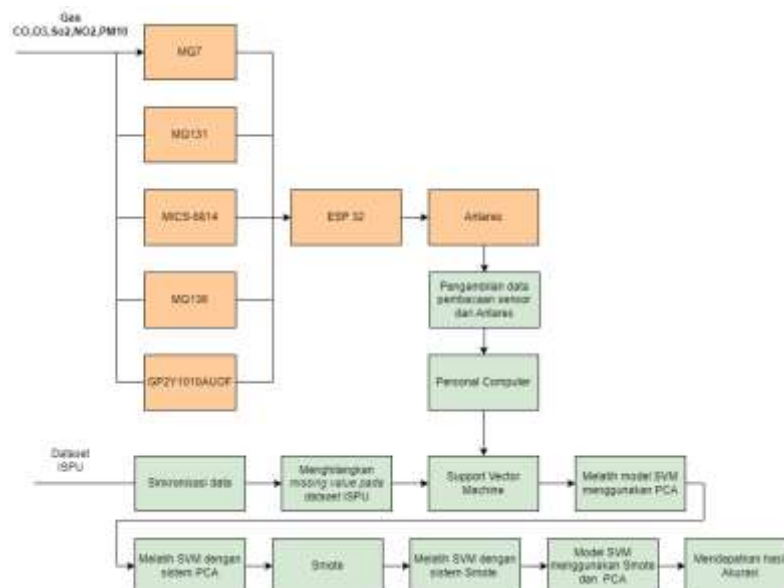
### PERANCANGAN SISTEM

#### 3.1 Desain Sistem

Penelitian tugas akhir akan dirancang untuk memonitoring polusi udara secara multi titik. Monitoring dilakukan dengan menggunakan sistem mikrokontroler berbasis ESP32 yang di program menggunakan perangkat lunak Arduino IDE. Sensor nirkabel akan terhubung dimana sensor mengumpulkan data dan data di akusisi menjadi data yang siap dikirim kepada jaringan. sensor memiliki sensor sebagai berikut; sensor GP2Y1010AU0F, MQ-136, MQ-131, MiCS 6814 dan MQ-7.

#### 3.2 Diagram Blok

Sistematika kajian tugas akhir ini merupakan gabungan dari tiga tugas akhir. Tiga di antaranya adalah rancang bangun monitoring polusi udara dengan sistem IOT, klasifikasi data pengolahan polusi udara dan data preprocessing berdasarkan data yang diperoleh dari sensor. Tugas akhir ini akan fokus terhadap proses pra-data sensor untuk menemukan akurasi yang tepat. Hal ini dapat dilihat pada bagian hijau dari diagram blok pada Gambar 3.1.

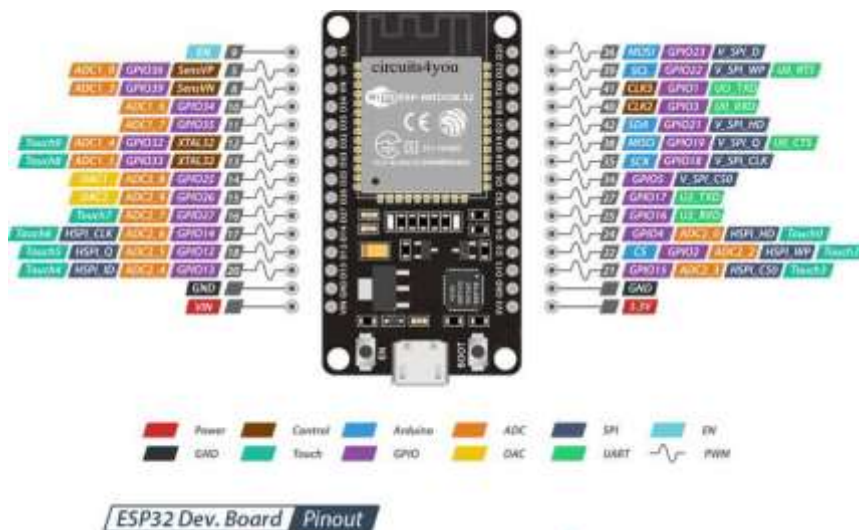


**Gambar 3. 1** Diagram Blok Monitoring Polusi Udara

### 3.2.1 Spesifikasi Komponen

#### 1. Mikrokontroler ESP 32

ESP32 adalah sistem tertanam dengan arsitektur Harvard dual-core Xtensa LX6 CPU. Semua memori tertanam, memori eksternal, dan periferal terletak di bus data dan/atau bus instruksi CPU. Mikrokontroler memiliki inti ganda - PRO\_CPU untuk protokol dan APP\_CPU untuk aplikasi, tetapi tujuannya tidak tetap. Ruang alamat untuk bus data dan instruksi adalah 2MB, dan ruang alamat periferal adalah 512KB. Selain itu, memori yang disematkan adalah ROM 448KB, SRAM 520KB, dan dua memori RTC 8KB. Memori eksternal mendukung hingga empat kali 16MB Flash.



**Gambar 3. 2** Diagram *Pinout* ESP32 DOIT DevKit v1

Jenis ESP yang digunakan pada penelitian ini merupakan ESP32 DOIT DevKit v1. Mikrokontroler ESP32 jenis ini memiliki 25 GPIO yang digunakan. Berikut adalah fitur utama dari ESP32 DOIT DevKit v1 adalah:

- 18 Analog-to-Digital Converter (ADC) channels
- 10 Capacitive sensing GPIOs
- 3 UART interfaces
- 3 SPI interface

- 2 I2C interface
- 16 PWM output channels
- 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces

## 2. Sensor MQ – 7



**Gambar 3. 3** Modul Sensor MQ-7

MQ-7 merupakan sensor gas yang digunakan dalam peralatan untuk mendeteksi gas karbon monoksida (CO) dalam dunia sehari-hari, industri, dan mobil. Fitur dari sensor gas MQ7 yaitu mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil dan berumur panjang. Pada penggunaan umumnya sensor ini dapat digunakan di mobil maupun pada industri. Sensor ini memiliki bahan Tin Dioksida  $\text{SnO}_2$  yang sensitif pada deteksi karbon monoksida pada suhu yang telah dipanaskan.

**Tabel 3. 1** Spesifikasi Teknis Sensor MQ-7

Simbol	Nama Parameter	Kondisi Teknis
$V_C$	Tegangan Rangkaian	5Volt $\pm$ 0.1
-	Jarak Pengukuran	20-2000ppm
$V_H$ (H)	Tegangan Pemanas (tinggi)	5V $\pm$ 0.1
$V_H$ (L)	Tegangan Pemanas (rendah)	1.4 $\pm$ 0.1
$R_L$	Resistansi Beban	Dapat disesuaikan
$R_H$	Resistansi Pemanas	33 $\Omega$ $\pm$ 5%

$T_H$ (H)	Waktu Pemanasan (tinggi)	$60 \pm 1$ seconds
$T_H$ (L)	Waktu Pemanasan	$90 \pm 1$ seconds
$P_H$	Konsumsi Pemanasan	350 mW
$R_s$	Tahanan Permukaan Terhadap Karbon Monoksida	2-20k pada 100ppm (CO)
Waktu pemanasan	Tidak kurang dari 48 jam	Jangkauan deteksi: 20ppm – 2000ppm karbon monoksida

### 3. Sensor MQ – 131



**Gambar 3. 4** Modul Sensor MQ-131

Merupakan suatu sensor gas yang sensitif terhadap ozon. Sensor ini biasanya digunakan untuk mengetahui indikator pencemaran ozon ( $O_3$ ). Ketika sensor ini bertemu gas ozon maka konduktivitas dari sensor berkurang. Maka gas ozon dapat dihitung dengan menghitung perubahan konduktivitas dari sensor. Sensor ini memiliki kelebihan yaitu masa hidup yang tinggi, sirkuit yang cukup simpel. Sensor ini juga banyak digunakan di alat peringatan konsentrasi ozon, peralatan ozon untuk industri, dan portable ozone detector.

**Tabel 3. 2** Spesifikasi Teknis MQ-131

Simbol	Nama Parameter	Kondisi Teknis
$V_C$	Tegangan Loop	$\leq 24V$ DC

$V_H$	Tegangan Pemanas	$5V \pm 0.1$ DC atau AC
$R_L$	Tahanan Load	Dapat diatur
$R_H$	Tahanan Pemanas	$34\Omega \pm 3\Omega$ (suhu kamar)
$P_H$	Konsumsi Daya Pemanas	$\leq 900mW$
$\Delta V_S$	Tegangan Output	$\geq 1.0V$ (dalam 200ppb $O_3$ )
	Jangkauan Deteksi	10 - 1000ppb Ozon
	Waktu Pemanasan Awal	Lebih dari 48 jam

#### 4. Sensor Sharp GP2Y1010AU0F



**Gambar 3. 5** Modul Sensor Sharp GP2Y1010AU0F

GP2Y1010AU0F adalah sensor partikel debu untuk digunakan dengan sistem sensor optik. Sensor ini cocok untuk mengukur partikel PM10 di udara. Sensor bekerja dengan dioda pemancar inframerah (IRED) dan fototransistor diagonal, sehingga partikel yang melewati ruang sensor menyebabkan pantulan optik. Sensor memiliki keuntungan dari konsumsi arus rendah (maks 20mA), sehingga cocok untuk sistem daya rendah. Sensornya sangat efisien bahkan bisa membaca partikel dari asap rokok. Spesifikasi teknis dari sensor Sharp GP2Y1010AU0F diperlihatkan pada table



**Tabel 3. 3** Spesifikasi teknis sensor Sharp GP2Y10100F[24]

Simbol	Parameter	Min	Rerata	Maks
K	Sensitivitas	0.35 V/(0.1mg/m <sup>3</sup> )	0.5 V/(0.1mg/m <sup>3</sup> )	0.65 V/(0.1mg/m <sup>3</sup> )
V <sub>OC</sub>	Output tanpa debu	0V	0.9V	1.5V
V <sub>OH</sub>	Jangkauan output tegangan	3.4V	-	-
I <sub>LED</sub>	Arus LED terminal	-	10mA	20mA
I <sub>CC</sub>	Konsumsi arus	-	11mA	20mA
RH	Heater Resistance	31Ω ± 5Ω (room temp.)		
PH	Heater Comsumption	≤ 900mW		
α	Concentration Slope	≤ 0.6 (R <sub>10ppm</sub> / R <sub>100ppm</sub> O <sub>3</sub> )		

### 5. Sensor MQ-136

**Gambar 3. 6** Modul Sensor MQ-136

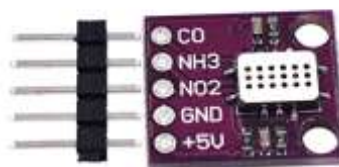
Sensor gas MQ-136 merupakan suatu sensor gas yang dapat digunakan pada SnO<sub>2</sub>, yang memiliki konduktifitas rendah di udara bersih. Namun sensor gas ini dapat membaca adanya gas SO<sub>2</sub>, sehingga ketika konsentrasi gas SO<sub>2</sub> meningkat maka konduktivitas sensor akan

meningkat juga. Sensor gas MQ-136 memiliki kepekaan yang tinggi terhadap SO<sub>2</sub>, dan juga sensor ini dapat membaca adanya jenis sulfur lainnya. Karakter utama pada MQ-136 ; memiliki sensitivitas baik pada gas SO<sub>2</sub>, ketahanan yang baik sehingga memiliki umur yang panjang dengan biaya yang rendah, dan memiliki jalur driver yang simple. Sensor ini dapat membaca SO<sub>2</sub> yang masih di dalam rentang konsentrasi 1-200ppm.

**Tabel 3. 4** Spesifikasi Teknis MQ-136[25]

Simbol	Parameter	Kondisi Teknis
-	Concentration	1-200ppm (SO <sub>2</sub> )
VC	Loop Voltage	≤ 24V DC
VH	Heater Voltage	5.0V±0.2V AC/DC
RH	Heater Resistance	31Ω± 3Ω (Room Temp.)
PH	Heater Consumption	≤ 900mW
RS	Sensing Resistance	2Ω - 20KΩ (in 50ppm SO <sub>2</sub> )
-	Standard Test Circuit	V <sub>C</sub> : 5.0V±0.1V; V <sub>H</sub> : 5.0V±0.1V
-	Preheat time	Over 48 hours

## 6. Sensor MiCS 6814



**Gambar 3. 7** Modul Sensor MiCS 6814

Sensor MiCS-6814 merupakan sensor MEMS (Micro Electro Mechanical Systems) yang kuat untuk mendeteksi polusi dari knalpot mobil dan untuk bau pertanian atau industri. Sensor ini merupakan sensor yang kompleks yang dapat mendeteksi beberapa gas, seperti

karbon monoksida, etanol, amonia, metana, nitrogen dioksida, dan propane. Sensor MiCS jenis ini memiliki desain yang kompak yang memiliki kemampuan membaca NO<sub>2</sub> yang cukup sensitive, sehingga pada penelitian ini penulis mempertimbangkan untuk menggunakan sensor MiCS-6814.

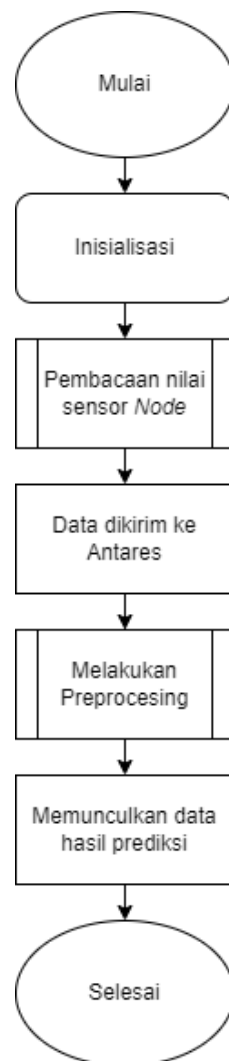
**Tabel 3. 5** Tabel Spesifikasi Teknis MiCS 6814

Simbol	Parameter	Kondisi Teknis
P <sub>H</sub>	Disipasi daya pemanas maksimum	88 mW
P <sub>S</sub>	Disipasi daya lapisan sensitif maksimum	8 mW
V <sub>supply</sub>	Pasokan tegangan arus pemanas	4.9 – 5.1 V
RH	Kisaran kelembapan relative	5 – 95 %RH
P <sub>H</sub>	Daya Pemanas	60-73 mW
V <sub>H</sub>	Tegangan Pemanas	2.2 V
I <sub>H</sub>	Arus Pemanas	30 mA
R <sub>H</sub>	Tahanan panas pada daya nominal	64-80 $\Omega$ (optimal di 72 $\Omega$ )

### 3.3 Desain Perangkat Lunak

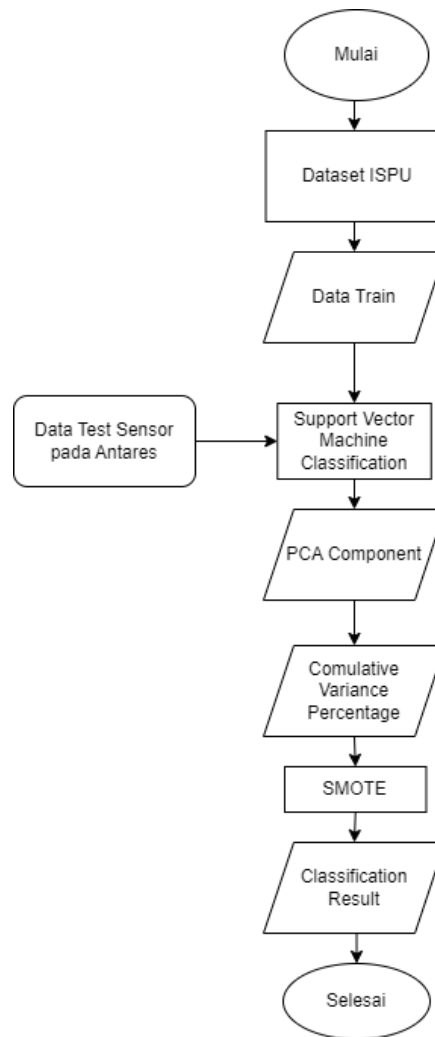
Perangkat lunak pada penelitian ini menggunakan beberapa aplikasi seperti Jupiter Notebook, Arduino IDE. Data sensor yang sudah di terbaca akan dikirimkan ke perangkat lunak Arduino IDE. Dalam perangkat lunak tersebut akan dimunculkan data hasil pembacaan sensor yang akan di ekspor ke iot. Hasil pada pembacaan sensor akan dibutuhkan kembali pada proses preprocessing dimana data tersebut akan di unduh. Setelah data hasil pembacaan sensor tersebut di dapatkan, maka selanjutnya data tersebut akan dilatih pada perangkat lunak Jupiter Notebook.

### 3.3.1 Diagram Alir



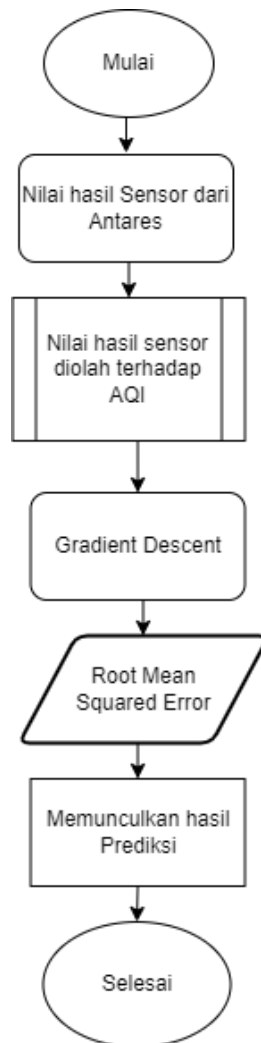
**Gambar 3. 8** Diagram Alir Keseluruhan Sistem

Pada diagram alir gambar 3.12 merupakan gambaran sistem keseluruhan sistem akan dimulai dengan inisiasi seluruh port serial dari sensor yang digunakan. Sensor membaca nilai kualitas udara dan mengambil data yang dikirimkan ke IoT. Setelah data didapatkan dari IoT, maka dilakukan *preprocessing* untuk mendapatkan nilai prediksi.



**Gambar 3. 9** Diagram Alir Preprocessing

Pada diagram alir gambar 3.13 merupakan sistem pada tahap preprocessing sistem akan dimulai dengan melatih dataset ISPU. Data latih ISPU tersebut digunakan untuk mengolah data hasil pembacaan sensor yang menjadi data test pada proses ini. Hasil pembacaan data sensor akan diolah dengan metode Support Vector Machine bertujuan untuk klasifikasi yang akan mengeluarkan hasil akurasi dari metode ini. Dan selanjutnya data test pembacaan sensor akan diproses dengan metode Principal Component Analysis dan mengeluarkan Cumulative Variance Percentage. Selanjutnya dilakukan metode SVM yang dipadukan dengan SMOTE. Metode ini digunakan untuk melihat apakah hasil proses PCA tersebut akan naik atau tidaknya.



**Gambar 3. 10** Diagram Alir Prediksi Kualitas Udara

Pada diagram alir gambar 3.14 merupakan sistem pada tahap prediksi kualitas udara dari pembacaan sensor yang dirancang. Data hasil pembacaan sensor didapatkan langsung dari perangkat lunak IoT. Sistem ini terhubung langsung dengan menggunakan subscribe data langsung dari IoT. Data test tersebut diolah dengan standard AQI. Setelah mendapatkan hasil AQI, selanjutnya akan diproses dengan metode Gradient Descent untuk mendapatkan nilai optimasi dari pembacaan sensor. Setelah mendapatkan nilai optimalnya, sistem akan masuk ke proses RMSE (Root Mean Squared Error), yaitu suatu cara untuk mengevaluasi model regresi linear dengan mengukur tingkat akurasi hasil perkiraan nilai sensor.

## BAB IV

### HASIL DAN ANALISIS

#### 4.1 Subscribe Software IoT

Data yang telah di proses pada pengujian analisis komponen sensor akan dikirimkan melalui IoT. Pada IoT akan terlihat semua hasil pembacaan sensor dengan beberapa parameter diantaranya; mq131(O3), mq136(SO2), mq7(CO), mics6814(NO2), dan sharp gp2y(PM10).

	O3_SubIndex	NO2_SubIndex	PM10_SubIndex	CO_SubIndex	SO2_SubIndex
0	336.915993	4.3375	144.589267	40.38585	1.623190
1	338.223785	4.3250	85.119350	38.87555	1.516079
2	338.413006	4.2750	10.045160	38.33695	1.222757
3	341.725993	4.2000	182.976333	39.69400	1.251506
4	339.754991	4.2250	123.458067	37.27645	1.148999
5	338.793340	4.2625	104.081733	34.71980	1.310850
6	339.754991	4.0875	0.000000	38.33695	1.122395
7	340.535622	4.0625	110.662400	35.59955	0.981860
8	341.926679	4.1875	8.454838	35.85385	1.013417
9	339.949184	4.1750	182.976333	35.59955	1.021449
10	338.793340	4.0125	182.976333	38.47105	1.104969
11	334.209573	4.0250	163.600000	37.27645	0.997528
12	338.602857	4.0750	166.634400	37.01475	1.021449
13	339.176122	3.9625	172.301067	37.40780	0.899576
14	339.176122	4.1375	182.976333	35.72650	0.864262
15	338.035176	4.0625	182.976333	36.75440	1.176173
16	338.413006	4.0375	110.443000	35.98150	0.906793
17	338.223785	3.9125	111.722600	33.97840	0.885299
18	340.929814	4.0625	149.159133	36.75440	0.850489
19	343.144917	4.0250	112.636533	32.76825	0.850489
20	341.127885	3.9875	111.795733	33.49055	0.857350
21	333.858256	4.0250	148.281733	34.84450	0.892413
22	332.304063	4.0000	118.083867	34.22425	0.850489
23	333.683451	4.1000	179.393533	34.59545	0.899576
24	334.033655	4.0125	120.167733	36.10945	0.857350
25	333.683451	4.0750	125.322600	34.10115	0.850489
26	332.645714	4.0250	102.802133	33.61205	0.836910
27	332.134007	4.0125	127.516133	34.10115	0.885299
28	333.162152	4.1375	172.301067	32.64895	0.885299
29	328.046623	4.0625	182.976333	34.84450	1.005445

**Gambar 4. 1** Hasil Subscribe data IoT

Pada tahap ini data hasil pengujian sensor akan dimasukkan untuk dapat diproses pada perbandingan dari hasil pengujian sensor dan juga hasil data train. Nilai pembacaan sensor ini didapatkan secara realtime dengan perangkat keras sistem monitoring. Dengan interval waktu pengujian selama 8 jam dengan interval pengambilan sampel setiap 15 menit.

#### 4.2 Hasil Analisis Metode Support Vector Machine

Tahap ini dilakukan pengujian hasil nilai sensor dengan metode Support Vector Machine. Tujuan dari penggunaan SVM ini yaitu sebagai cara untuk melihat apakah sistem Principal Componen Analysis ini dapat digunakan untuk menemukan akurasi yang lebih baik. SVM ini menggunakan parameter untuk menguji model pembelajaran mesin dengan parameter Gamma, Cost, dan Kernel RBF. Penggunaan data train yang digunakan sebanyak 200 data, data test yang digunakan sebanyak 30 data dengan atribut yang digunakan ada 5, yaitu PM10, Karbon Monoksida, Sulfur Dioksida, Ozon, Nitrogen Dioksida. Penggunaan 5 atribut tersebut menggunakan 3 kategori yaitu baik, tidak baik, dan sangat tidak baik.

```
from sklearn.svm import SVC
clf = SVC(kernel = 'rbf', gamma=1, C=1)
clf.fit(X_train, y_train)

SVC(C=1, gamma=1)
```

**Gambar 4. 2** Penggunaan Parameter pada SVM

Pada data hasil akurasi pembacaan sensor didapatkan nilai akurasi sebesar 0,7 atau 70% dari keseluruhan data. Untuk hasil akurasi tersebut dikategorikan baik untuk penggunaan metode pembelajaran mesin.

```
clf.score(X_test, y_test)

0.7
```

**Gambar 4. 3** Hasil Akurasi SVM



```
# Using MAPE error metrics to check for the error rate and accuracy level
SVR_MAPE= MAPE(y_test,y_pred)
print("MAPE: ",SVR_MAPE)

MAPE: 30.0
```

**Gambar 4. 4** Rata-rata Persentase Kesalahan Absolut

Gambar 4.4 merupakan MAPE (*Mean Absolut Percentage Error*) yaitu rata-rata persentase kesalahan absolut yang digunakan untuk mengukur galat ramalan dibandingkan dengan nilai actual. Penggunaan ini untuk melihat prediksi akurasi semakin tinggi nilai MAPE maka akan semakin kecil nilai akurasinya. Karena MAPE yang di hasilkan memiliki nilai 30% maka prediksi dikategorikan layak.

	Real Values	Predicted Values
0	Sangat Tidak Baik	Sangat Tidak Baik
1	Sangat Tidak Baik	Sangat Tidak Baik
2	Sangat Tidak Baik	Sangat Tidak Baik
3	Sangat Tidak Baik	Sangat Tidak Baik
4	Sangat Tidak Baik	Sangat Tidak Baik
5	Sangat Tidak Baik	Sangat Tidak Baik
6	Sangat Tidak Baik	Sangat Tidak Baik
7	Sangat Tidak Baik	Sangat Tidak Baik
8	Sangat Tidak Baik	Sangat Tidak Baik
9	Sangat Tidak Baik	Sangat Tidak Baik
10	Sangat Tidak Baik	Sangat Tidak Baik
11	Sangat Tidak Baik	Sangat Tidak Baik
12	Sangat Tidak Baik	Sangat Tidak Baik
13	Sangat Tidak Baik	Sangat Tidak Baik
14	Sangat Tidak Baik	Sangat Tidak Baik
15	Sangat Tidak Baik	Sangat Tidak Baik
16	Sangat Tidak Baik	Sangat Tidak Baik
17	Sangat Tidak Baik	Tidak Baik
18	Sangat Tidak Baik	Sangat Tidak Baik
19	Sangat Tidak Baik	Sangat Tidak Baik
20	Sangat Tidak Baik	Sangat Tidak Baik
21	Sangat Tidak Baik	Tidak Baik
22	Sangat Tidak Baik	Tidak Baik
23	Sangat Tidak Baik	Tidak Baik
24	Sangat Tidak Baik	Tidak Baik
25	Sangat Tidak Baik	Tidak Baik
26	Sangat Tidak Baik	Tidak Baik
27	Sangat Tidak Baik	Tidak Baik
28	Sangat Tidak Baik	Tidak Baik
29	Sangat Tidak Baik	Sangat Tidak Baik

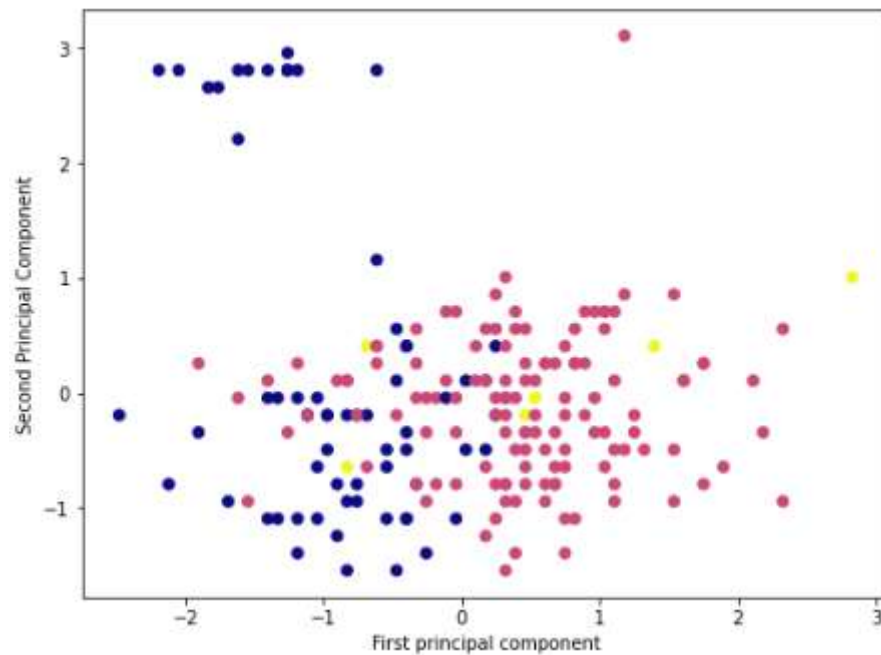
**Gambar 4. 5** Hasil Peramalan Metode Pembelajaran Mesin SVM

Hasil dari penggunaan metode SVM dapat dilakukan pemberian label pada setiap hasil pengolahannya, dengan fitur ini dapat memberikan hasil pembacaan yang terbaik yang digunakan pada tahap *preprocessing*.

#### **4.3 Hasil Analisis menggunakan metode Support Vector Machine dengan Principal Component Analysis**

Sistem menggunakan metode Principal Component Analysis untuk melihat apakah ada pengaruh hasil akurasi data dengan hanya menggunakan metode SVM

saja. Proses ini dilakukan dengan menggunakan beberapa Principal Component digabungkan dengan model SVM.



**Gambar 4. 6** Color Map Data ISPU

```
Cumulative Variances (Percentage):
[ 51.62838829  70.63835891  86.19103274  94.42078254 100.      ]
Number of components: 5
```

**Gambar 4. 7** Cumulative Variance

Pada gambar 4.6 merupakan *cumulative variance* yaitu merupakan persentase varian data yang digunakan dalam PCA. Pada penelitian ini menggunakan persentase varian 100%. Komponen PCA yang digunakan berjumlah 5. Hal ini digunakan untuk memperoleh nilai akurasi yang tinggi.

```
SVC_model.score(X_test_pca,y_test)
0.6666666666666666
```

**Gambar 4. 8** Hasil Presentase SVM dan PCA

Terlihat pada gambar 4.8 pada penelitian ini menghasilkan akurasi yaitu 0.6 atau 66% dari keseluruhan sistem dengan persentase di 5 komponen. Untuk hasil ini dikategorikan cukup baik untuk penggunaan metode pembelajaran mesin.

	Real Values	Predicted Values
0	Sangat Tidak Baik	Sangat Tidak Baik
1	Sangat Tidak Baik	Sangat Tidak Baik
2	Sangat Tidak Baik	Sangat Tidak Baik
3	Sangat Tidak Baik	Sangat Tidak Baik
4	Sangat Tidak Baik	Sangat Tidak Baik
5	Sangat Tidak Baik	Sangat Tidak Baik
6	Sangat Tidak Baik	Sangat Tidak Baik
7	Sangat Tidak Baik	Sangat Tidak Baik
8	Sangat Tidak Baik	Sangat Tidak Baik
9	Sangat Tidak Baik	Sangat Tidak Baik
10	Sangat Tidak Baik	Sangat Tidak Baik
11	Sangat Tidak Baik	Sangat Tidak Baik
12	Sangat Tidak Baik	Sangat Tidak Baik
13	Sangat Tidak Baik	Sangat Tidak Baik
14	Sangat Tidak Baik	Sangat Tidak Baik
15	Sangat Tidak Baik	Sangat Tidak Baik
16	Sangat Tidak Baik	Sangat Tidak Baik
17	Sangat Tidak Baik	Tidak Baik
18	Sangat Tidak Baik	Sangat Tidak Baik
19	Sangat Tidak Baik	Sangat Tidak Baik
20	Sangat Tidak Baik	Sangat Tidak Baik
21	Sangat Tidak Baik	Tidak Baik
22	Sangat Tidak Baik	Tidak Baik
23	Sangat Tidak Baik	Tidak Baik
24	Sangat Tidak Baik	Tidak Baik
25	Sangat Tidak Baik	Tidak Baik
26	Sangat Tidak Baik	Tidak Baik
27	Sangat Tidak Baik	Tidak Baik
28	Sangat Tidak Baik	Tidak Baik
29	Sangat Tidak Baik	Sangat Tidak Baik

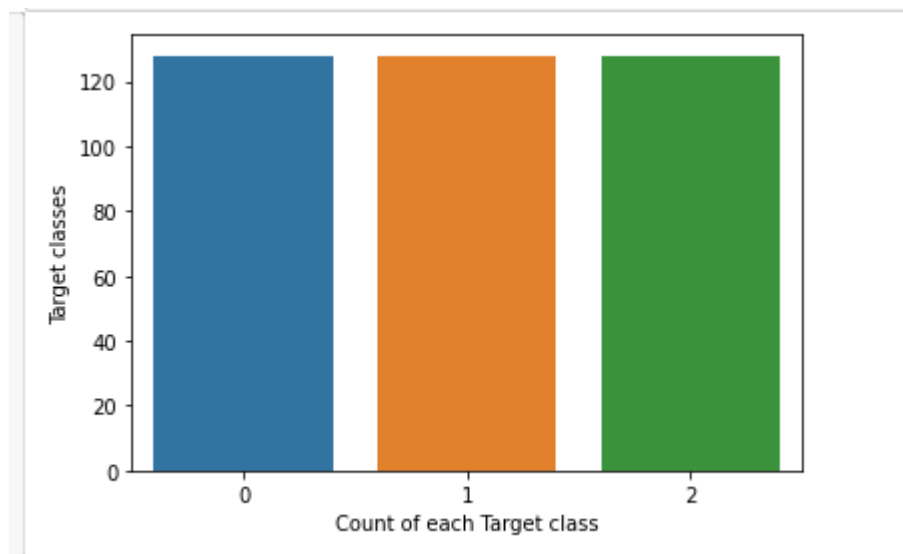
**Gambar 4. 9** Hasil Peramalan Metode Pembelajaran Mesin SVM dengan PCA

Pembacaan dari prediksi hasil metode SVM dan PCA dapat dilihat pada gambar 4.9, untuk hasil prediksi cukup tidak berubah dari metode SVM

dikarenakan metode hasil akurasi yang berbeda sedikit. Sehingga untuk keluaran prediksi dari metode SVM dan SVM dengan PCA tidak adanya perubahan.

#### 4.4 Hasil Analisis Menggunakan SMOTE

Data yang diterima saat proses sebelumnya masih berupa data imbalance. Sehingga dibutuhkan adanya proses SMOTE. Data imbalance ini memiliki nilai target kelas tertinggi di nilai 120 pada kelas “1” atau sedang, dapat dilihat pada gambar 4.10 berikut.



**Gambar 4. 10** Data Imbalance Sistem Polusi Udara

Metode SMOTE menggunakan data maksimum sebagai target yang akan diterapkan pada setiap variabel minoritas. Hasil yang sudah di proses pada SMOTE ini menghasilkan data yang balance, sehingga variabel satu sama lain berjumlah sama. Pengaruh imbalance data akurasi sebelum dan setelah SMOTE menyebabkan kenaikan pada persentase. Hal ini diakibatkan oleh semakin banyak jumlah data yang diolah semakin akurat nilai akurasi. Naik turunnya presentase tergantung dari sistem random data yang di proses.

Hasil yang sudah di proses pada SMOTE ini menghasilkan data yang balance, sehingga variabel satu sama lain berjumlah sama.

```
clf.score(X_test,y_test)
```

```
0.5666666666666667
```

**Gambar 4. 11** Hasil Persentase SVM dan SMOTE

Terlihat pada gambar 4.11 merupakan hasil proses model SVM digabungkan dengan SMOTE pada sensor. Hasil akurasi yang di dapatkan yaitu 0.56 atau 56%. Nilai yang didapatkan ketika semua data yang tidak balance tersebut menurun dari metode pembelajaran mesin sebelumnya. Hasil ini dapat dikategorikan buruk untuk penggunaan pada metode model pembelajaran mesin.

	Real Values	Predicted Values
0	Sangat Tidak Baik	Sangat Tidak Baik
1	Sangat Tidak Baik	Sangat Tidak Baik
2	Sangat Tidak Baik	Sangat Tidak Baik
3	Sangat Tidak Baik	2
4	Sangat Tidak Baik	Sangat Tidak Baik
5	Sangat Tidak Baik	Sangat Tidak Baik
6	Sangat Tidak Baik	Sangat Tidak Baik
7	Sangat Tidak Baik	Sangat Tidak Baik
8	Sangat Tidak Baik	Sangat Tidak Baik
9	Sangat Tidak Baik	Sangat Tidak Baik
10	Sangat Tidak Baik	2
11	Sangat Tidak Baik	2
12	Sangat Tidak Baik	2
13	Sangat Tidak Baik	Sangat Tidak Baik
14	Sangat Tidak Baik	Sangat Tidak Baik
15	Sangat Tidak Baik	Sangat Tidak Baik
16	Sangat Tidak Baik	Sangat Tidak Baik
17	Sangat Tidak Baik	Tidak Baik
18	Sangat Tidak Baik	Sangat Tidak Baik
19	Sangat Tidak Baik	Sangat Tidak Baik
20	Sangat Tidak Baik	Sangat Tidak Baik
21	Sangat Tidak Baik	Tidak Baik
22	Sangat Tidak Baik	Tidak Baik
23	Sangat Tidak Baik	Tidak Baik
24	Sangat Tidak Baik	Tidak Baik
25	Sangat Tidak Baik	Tidak Baik
26	Sangat Tidak Baik	Tidak Baik
27	Sangat Tidak Baik	Tidak Baik
28	Sangat Tidak Baik	Tidak Baik
29	Sangat Tidak Baik	Tidak Baik

**Gambar 4. 12** Hasil Peramalan Metode Pembelajaran Mesin SVM dengan SMOTE

Pada hasil pembacaan metode SMOTE, hasil prediksi yang dikeluarkan berbeda dengan kedua metode sebelumnya, hal ini dapat dipastikan bahwa hasil dari akurasi yang cukup jauh dari metode SVM dan SVM dengan PCA. Sehingga hasil dari prediksi yang dihasilkan cukup jauh dari metode lainnya.

#### 4.5 Perbandingan Pengujian Metode SVM, SVM dengan PCA dan SVM dengan SMOTE

Bagian ini akan melakukan perbandingan performa metode Support Vector Machine, Support Vector Machine dengan Principal Component Analysis, Support Vector Machine dengan SMOTE.

**Tabel 4. 1** Tabel Perbandingan Akurasi Setiap Metode

Metode <i>Machine Learning</i>	Hasil Akurasi Penggunaan Metode <i>Machine Learning</i>
Support Vector Machine	70%
Support Vector Machine dengan Principal Componen Analysis	66,6%
Support Vector Machine dengan SMOTE	56%

Pada metode Support Vector Machine memiliki tingkat akurasi senilai 70%, metode Support Vector Machine dan Principal Component Analysis memiliki tingkat akurasi senilai 66,6%, dan metode Support Vector Machine dengan SMOTE memiliki tingkat akurasi senilai 56%. Dari hasil penelitian ini dapat disimpulkan bahwa dengan metode Support Vector Machine merupakan hasil model pembelajaran mesin yang sangat baik.

Jika penggunaan Support Vector Machine tersebut digunakan dengan metode Principal Component Analyst maka hasil akurasi yang diperoleh lebih rendah karena parameter yang digunakan akan semakin banyak. Sehingga hasil dari pengolahan akurasi akan menurun. Untuk penggunaan Support Vector Machine dengan SMOTE bisa menyebabkan overfitting dan hasil yang tidak optimal jika tidak digunakan dengan benar.

SMOTE (Synthetic Minority Over-sampling Technique) digunakan untuk membuat sintesis dari data minoritas (kelas minoritas) dengan mengambil sample dari data yang ada dan membuat sample baru dengan interpolasi antara sample.



Sedangkan PCA (Principal Component Analysis) digunakan untuk melakukan dimensi reduksi pada data. Namun, jika data sudah tereduksi dengan baik, pemakaian PCA tidak akan memberikan perubahan signifikan pada hasil SVM.

Namun, jika menggunakan metode ketiga metode tersebut secara bersamaan tingkat akurasi yang didapatkan akan menurun, ini dipengaruhi oleh beberapa kasus:

1. Principal Component Analysis dapat menghilangkan beberapa fitur yang penting dari data yang digunakan oleh SVM untuk membuat keputusan klasifikasi.
2. SMOTE dapat menambahkan data yang tidak sebenarnya dari kelas minoritas, yang dapat menyebabkan overfitting dan menurunkan tingkat akurasi.

#### 4.6 Hasil Pengujian Prediksi Sensor

	mq131	mq136	mq7	no2	gp2y	waktu
0	406.9772	1.298552	0.807717	3.47	166.883900	2023-02-02 09:31:13.270399
1	414.0262	1.212863	0.777511	3.46	85.119350	2023-02-02 09:46:13.351917
2	415.0461	0.978206	0.766739	3.42	10.045160	2023-02-02 10:01:13.556085
3	432.9031	1.001205	0.793880	3.36	224.464500	2023-02-02 10:16:13.644700
4	422.2794	0.919199	0.745529	3.38	135.187100	2023-02-02 10:31:13.746660
5	417.0961	1.048680	0.694396	3.41	106.122600	2023-02-02 10:46:13.819588
6	422.2794	0.897916	0.766739	3.27	0.000000	2023-02-02 11:01:13.900175
7	426.4870	0.785488	0.711991	3.25	115.993600	2023-02-02 11:16:13.983368
8	433.9848	0.810734	0.717077	3.35	8.454838	2023-02-02 11:31:14.047450
9	423.3261	0.817159	0.711991	3.34	224.464500	2023-02-02 11:46:14.117974
10	417.0961	0.883975	0.769421	3.21	224.464500	2023-02-02 12:01:14.187461
11	392.3896	0.798022	0.745529	3.22	195.400000	2023-02-02 12:16:14.282473
12	416.0694	0.817159	0.740295	3.26	199.951600	2023-02-02 12:31:14.359501
13	419.1593	0.719661	0.748156	3.17	208.451600	2023-02-02 12:46:14.433239
14	419.1593	0.691410	0.714530	3.31	224.464500	2023-02-02 13:01:14.505150
15	413.0096	0.940938	0.735088	3.25	224.464500	2023-02-02 13:16:14.566401
16	415.0461	0.725434	0.719630	3.23	115.664500	2023-02-02 13:31:14.638906
17	414.0262	0.708239	0.679568	3.13	117.583900	2023-02-02 13:46:14.796869
18	428.6117	0.680391	0.735088	3.25	173.738700	2023-02-02 14:01:14.882948
19	440.5511	0.680391	0.655365	3.22	118.954800	2023-02-02 14:16:14.947656
20	429.6793	0.685880	0.669811	3.19	117.693600	2023-02-02 14:31:15.151332
21	390.4960	0.713930	0.696890	3.22	172.422600	2023-02-02 14:46:15.226446
22	382.1189	0.680391	0.684485	3.20	127.125800	2023-02-02 15:01:15.300932
23	389.5538	0.719661	0.691909	3.28	219.090300	2023-02-02 15:16:15.371558
24	391.4414	0.685880	0.722189	3.21	130.251600	2023-02-02 15:31:15.426017
25	389.5538	0.680391	0.682023	3.26	137.983900	2023-02-02 15:46:15.489849
26	383.9604	0.669528	0.672241	3.22	104.203200	2023-02-02 16:01:15.581666
27	381.2023	0.708239	0.682023	3.21	141.274200	2023-02-02 16:16:15.661181
28	386.7440	0.708239	0.652979	3.31	208.451600	2023-02-02 16:31:15.720711
29	359.1713	0.804356	0.696890	3.25	224.464500	2023-02-02 16:46:15.798581

**Gambar 4. 13** Data Test Pembacaan Sensor

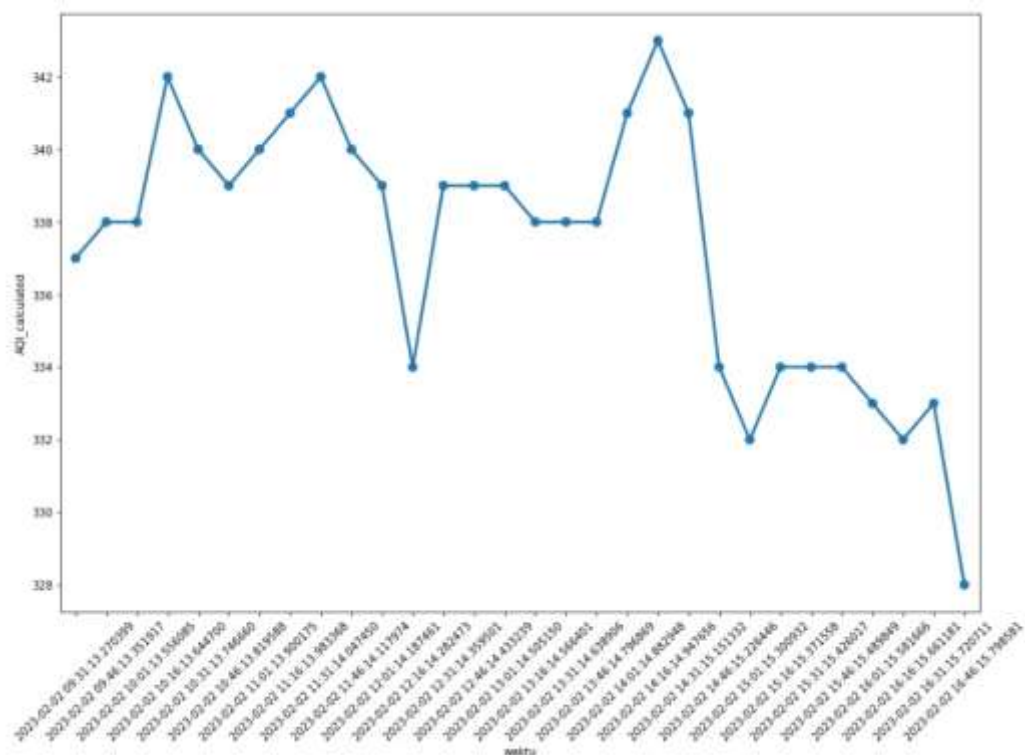
Pengujian sensor pada sensor dilakukan selama 8 jam dengan interval waktu setiap pengambilan sensor yaitu 15 menit. Dan target pengambilan data yaitu 30 data pada setiap sensor.

waktu	O3_Subindex	NO2_Subindex	PM10_Subindex	CO_Subindex	SO2_Subindex	AQI_calculated	AQI_bucket_calculated
2023-02-02 09:31:13.270399	336.915993	4.3375	144.589267	40.38585	1.623190	337.0	1
2023-02-02 09:46:13.351917	338.223785	4.3250	85.119350	38.87555	1.516079	338.0	1
2023-02-02 10:01:13.558085	338.413008	4.2750	10.045160	38.33695	1.222757	338.0	1
2023-02-02 10:16:13.644700	341.725993	4.2000	182.976333	39.69400	1.251506	342.0	1
2023-02-02 10:31:13.746660	339.754991	4.2250	123.458067	37.27645	1.148999	340.0	1
2023-02-02 10:46:13.819588	338.793340	4.2625	104.081733	34.71980	1.310850	339.0	1
2023-02-02 11:01:13.900175	339.754991	4.0875	0.000000	38.33695	1.122395	340.0	1
2023-02-02 11:16:13.983368	340.535622	4.0625	110.862400	35.59955	0.981860	341.0	1
2023-02-02 11:31:14.047450	341.926679	4.1875	8.454838	35.85385	1.013417	342.0	1
2023-02-02 11:46:14.117974	339.949184	4.1750	182.976333	35.59955	1.021449	340.0	1
2023-02-02 12:01:14.187461	338.793340	4.0125	182.976333	38.47105	1.104969	339.0	1
2023-02-02 12:16:14.262473	334.209573	4.0250	163.800000	37.27645	0.997528	334.0	1
2023-02-02 12:31:14.359501	338.802957	4.0750	188.634400	37.01475	1.021449	339.0	1
2023-02-02 12:46:14.433239	339.176122	3.8625	172.301067	37.40780	0.899576	339.0	1
2023-02-02 13:01:14.505150	339.176122	4.1375	182.976333	35.72650	0.864262	339.0	1
2023-02-02 13:16:14.566401	338.035176	4.0625	182.976333	36.75440	1.176173	338.0	1
2023-02-02 13:31:14.638906	338.413008	4.0375	110.443000	35.98150	0.906793	338.0	1
2023-02-02 13:46:14.796889	338.223785	3.9125	111.722800	33.97840	0.885299	338.0	1
2023-02-02 14:01:14.882948	340.929814	4.0625	149.159133	36.75440	0.850489	341.0	1
2023-02-02 14:16:14.947856	343.144917	4.0250	112.636533	32.76825	0.850489	343.0	1
2023-02-02 14:31:15.151332	341.127885	3.9875	111.795733	33.49055	0.857350	341.0	1
2023-02-02 14:46:15.226446	333.858256	4.0250	148.281733	34.84450	0.892413	334.0	1
2023-02-02 15:01:15.300932	332.304063	4.0000	118.083867	34.22425	0.850489	332.0	1
2023-02-02 15:16:15.371558	333.683451	4.1000	179.393533	34.59545	0.899578	334.0	1
2023-02-02 15:31:15.426017	334.033655	4.0125	120.187733	36.10945	0.857350	334.0	1
2023-02-02 15:46:15.489849	333.683451	4.0750	125.322800	34.10115	0.850489	334.0	1
2023-02-02 16:01:15.581866	332.645714	4.0250	102.802133	33.61205	0.836910	333.0	1
2023-02-02 16:16:15.681181	332.134007	4.0125	127.516133	34.10115	0.885299	332.0	1
2023-02-02 16:31:15.720711	333.162152	4.1375	172.301067	32.64895	0.885299	333.0	1
2023-02-02 16:46:15.798581	328.046623	4.0625	182.976333	34.84450	1.005445	328.0	1

**Gambar 4. 14** Data Pengujian Sensor dengan AQI

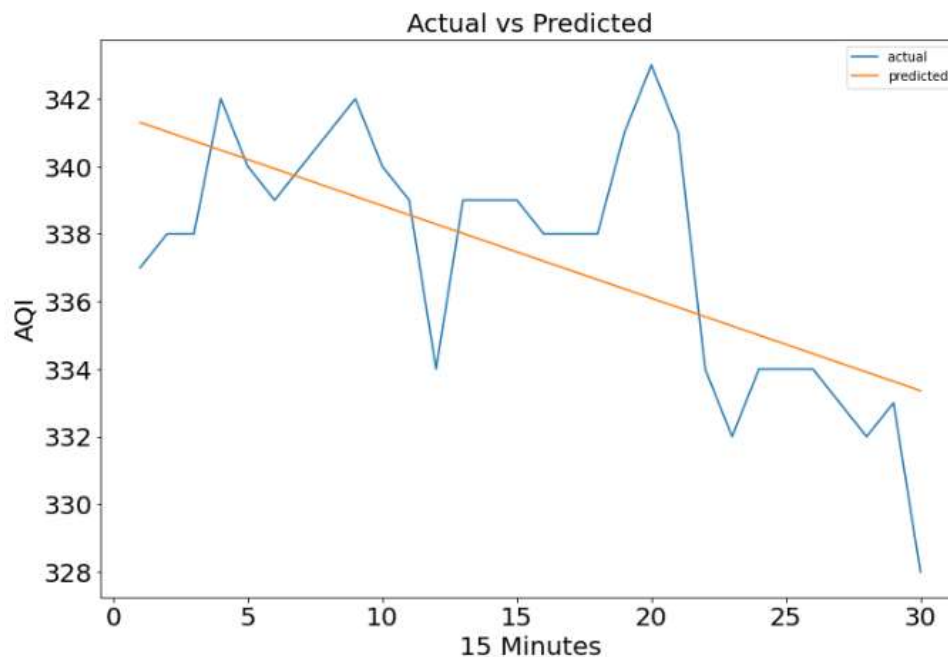
Pada pengujian prediksi dibutuhkan parameter yang telah diolah sebelumnya, yaitu O3 Subindeks, NO2 Subindeks, PM10 Subindeks, CO Subindeks, SO2 Subindeks, dan AQI calculation. Untuk mendapatkan parameter yang memenuhi kriteria penulis, maka penulis memberikan label subindeks pada setiap masing

masing parameter. Kondisi pada AQI Bucketing digunakan untuk memberikan kondisi pada setiap parameter sensor. Hal ini digunakan untuk mencapai hasil klasifikasi yang dibutuhkan pada proses pre-processing. Dengan hasil kondisi pada AQI Bucketing ini memberikan peluang penulis pada proses selanjutnya, yaitu melakukan klasifikasi kepada setiap variabel. Pada AQI Bucket Calculate diambil berdasarkan sensor yang mempunyai nilai maksimal. Pada data ini variabel yang bernilai maksimal adalah Ozon.



**Gambar 4. 15** Line Graph Data Percobaan Sensor

Dari hasil percobaan pada sensor, semakin lama udara pada pengujian sensor semakin baik karena kandungan kadar gas Ozon di udara semakin sedikit. Hal ini disebabkan oleh gas Ozon di udara pada pengujian sampel tergolong tidak sehat karena memiliki kandungan Ozon yang banyak. Lalu, setelah dilakukan percobaan selama 8 jam, ternyata kandungan gas ozon di udara semakin sedikit. Oleh karena itu, kualitas udara pada saat percobaan lebih baik dari data sebelumnya, tetapi kualitas udara masih tergolong sangat tidak sehat.



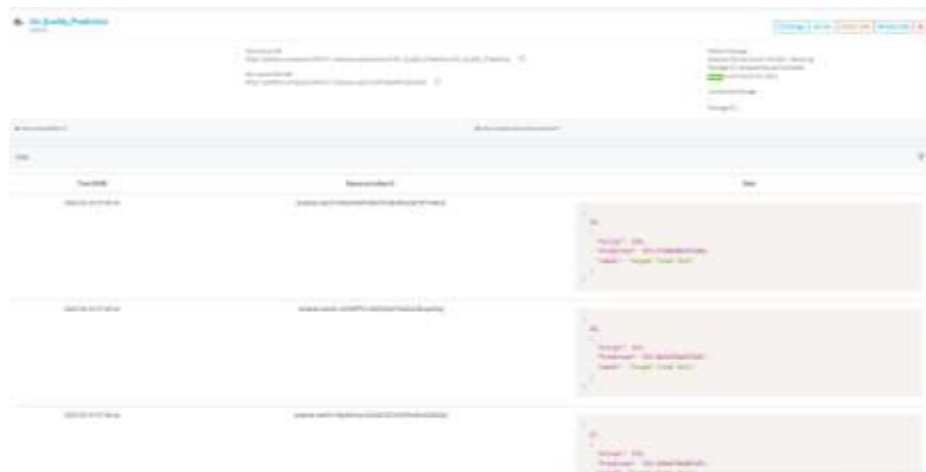
**Gambar 4. 16** Grafik Prediksi Sensor

Setelah dilakukan percobaan sensor menggunakan pembelajaran mesin. didapatkan hasil prediksi seperti gambar 4.23. Dapat dilihat terdapat garis biru pada grafik menandakan data pengujian langsung dan garis kuning merupakan data hasil prediksi. Pengambilan sampel setiap 15 menit di ruang terbuka, dengan rentang waktu 9 pagi hingga 5 sore. Didapatkan data pengujian sekitar 30 hasil pengujian. Hasil pembacaan sensor memiliki data yang cenderung menurun, ini dipengaruhi oleh semakin sedikitnya kadar gas yang menjadi pengujian sensor. Hasil prediksi ini menunjukkan kualitas udara yang semakin baik, begitu juga dengan data pengujian aslinya.

	Actual	Predicted	Label
0	337.0	341.294624	Sangat Tidak Baik
1	338.0	341.021431	Sangat Tidak Baik
2	338.0	340.748239	Sangat Tidak Baik
3	342.0	340.475046	Sangat Tidak Baik
4	340.0	340.201854	Sangat Tidak Baik
5	339.0	339.928661	Sangat Tidak Baik
6	340.0	339.655469	Sangat Tidak Baik
7	341.0	339.382277	Sangat Tidak Baik
8	342.0	339.109084	Sangat Tidak Baik
9	340.0	338.835892	Sangat Tidak Baik
10	339.0	338.562699	Sangat Tidak Baik
11	334.0	338.289507	Sangat Tidak Baik
12	339.0	338.016314	Sangat Tidak Baik
13	339.0	337.743122	Sangat Tidak Baik
14	339.0	337.469930	Sangat Tidak Baik
15	338.0	337.196737	Sangat Tidak Baik
16	338.0	336.923545	Sangat Tidak Baik
17	338.0	336.650352	Sangat Tidak Baik
18	341.0	336.377160	Sangat Tidak Baik
19	343.0	336.103967	Sangat Tidak Baik
20	341.0	335.830775	Sangat Tidak Baik
21	334.0	335.557582	Sangat Tidak Baik
22	332.0	335.284390	Sangat Tidak Baik
23	334.0	335.011198	Sangat Tidak Baik
24	334.0	334.738005	Sangat Tidak Baik
25	334.0	334.464813	Sangat Tidak Baik
26	333.0	334.191620	Sangat Tidak Baik
27	332.0	333.918428	Sangat Tidak Baik
28	333.0	333.645235	Sangat Tidak Baik
29	328.0	333.372043	Sangat Tidak Baik

**Gambar 4. 17** Hasil Prediksi pada Pembelajaran Mesin

Dari pengolahan data pembacaan sensor pada dapat memberikan hasil prediksi yang sangat mendekati nilai aktual. Hal ini memberikan proses hasil prediksi yang sangat sempurna pada sistem monitoring.



**Gambar 4. 18** Hasil Prediksi pada IoT



**Gambar 4. 19** Hasil Prediksi pada IoT (2)

Hasil pengolahan pada pembelajaran mesin berhasil di unggah ke IoT dengan data yang dapat dilihat seperti pada gambar 4.27 dan gambar 4.28. Hasil Prediksi ditampilkan dengan komponen variabel ‘aktual’ dan ‘predicted’ sehingga dapat dilihat bahwa perbandingan nilai dari kedua komponen tersebut tidak terlalu jauh. Dan fitur selanjutnya terdapat hasil prediksi dari pembacaan sensor tersebut yaitu pada variabel “label” yang dapat memberikan informasi kepada masyarakat bahwa prediksi polusi udara sekitar pengujian agar dapat mengantisipasi polutan tersebut.

## **BAB V**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

1. Penelitian ini menggunakan 5 parameter yang memenuhi kebutuhan data ISPU yaitu MQ131(O<sub>3</sub>), MQ136(SO<sub>2</sub>), MQ7(CO), MICS 6814(NO<sub>2</sub>), dan sharp GP2Y(PM<sub>10</sub>).
2. Dari hasil penelitian ini dapat disimpulkan bahwa dengan metode Support Vector Machine merupakan hasil model pembelajaran mesin yang sangat baik pada proses pembelajaran mesin dengan hasil akurasi 70% lebih baik dibandingkan dengan menggunakan metode Principal Component Analysis dan SMOTE yang dapat mereduksi dan memberikan variabel baru segingga.
3. Jika penggunaan Support Vector Machine digunakan dengan metode Principal Component Analyst maka hasil akurasi yang diperoleh lebih rendah karena parameter yang digunakan akan semakin banyak. Sehingga hasil dari pengolahan akurasi akan menurun. Untuk penggunaan Support Vector Machine dengan SMOTE bisa menyebabkan overfitting dan hasil yang tidak optimal jika tidak digunakan dengan benar.

#### **5.2 Saran**

1. Desain dibuat lebih kompak agar dapat lebih mudah dibawa kemana saja.
2. Menggunakan sensor yang dapat mengukur kualitas udara yang lebih akurat.
3. Penelitian untuk selanjutnya lebih terarah dan terfokus, tidak ber ubah-ubah.
4. Melakukan pengujian lebih lanjut mengenai penggunaan metode pembelajaran mesin pada tahap preprocessing



## DAFTAR PUSTAKA

- [1] “Repository - Monitoring Polusi Udara Menggunakan Mikrokontroler Hemat Daya Multi Titik Dengan Jaringan Mesh Nirkabel.”  
<https://repository.telkomuniversity.ac.id/pustaka/179304/monitoring-polusi-udara-menggunakan-mikrokontroler-hemat-daya-multi-titik-dengan-jaringan-mesh-nirkabel.html> (accessed Jun. 26, 2022).
- [2] A. A. Dharmasaputro, I. Prasetya, D. Wibawa, and M. Kallista, “Lembar Pengesahan Proposal Tugas Akhir Monitoring Dan Klasifikasi Polusi Udara Menggunakan Random Forest Dengan Imbalanced Dataset (Air Pollution Monitoring and Classification Using Random Forest With Imbalanced Dataset),” pp. 2–9.
- [3] A. Kurniawan, “Pengukuran Parameter Kualitas Udara (Co, No<sub>2</sub>, So<sub>2</sub>, O<sub>3</sub> Dan Pm<sub>10</sub>) Di Bukit Kototabang Berbasis Ispu,” *J. Teknosains*, vol. 7, no. 1, p. 1, 2018, doi: 10.22146/teknosains.34658.
- [4] S. Kaivonen and E. C. H. Ngai, “Real-time air pollution monitoring with sensors on city bus,” *Digit. Commun. Networks*, vol. 6, no. 1, pp. 23–30, Feb. 2020, doi: 10.1016/J.DCAN.2019.03.003.
- [5] P. M. Mannucci and M. Franchini, “Health effects of ambient air pollution in developing countries,” *Int. J. Environ. Res. Public Health*, vol. 14, no. 9, pp. 1–8, 2017, doi: 10.3390/ijerph14091048.
- [6] A. Famili, W. M. Shen, R. Weber, and E. Simoudis, “Data preprocessing and intelligent data analysis,” *Intell. Data Anal.*, vol. 1, no. 1, pp. 3–23, 1997, doi: 10.3233/IDA-1997-1102.
- [7] V. Roza, M. Ilza, and S. Anita, “Korelasi Konsentrasi Particulate Matter (PM<sub>10</sub>) di Udara dan Kandungan Timbal (Pb) dalam Rambut Petugas SPBU di Kota Pekanbaru,” *Din. Lingkung. Indones.*, vol. 2, no. 1, p. 52, 2015, doi: 10.31258/dli.2.1.p.52-60.
- [8] B. Michael Greenstone and Q. Fan, “Indonesia’s Worsening Air Quality and its Impact on Life Expectancy”.

- [9] M. Negara Lingkungan Hidup, “Keputusan Menteri Negara Lingkungan Hidup No. 45 Tahun 1997 Tentang : Indeks Standar Pencemar Udara”.
- [10] M. Rashid, S. Yunus, R. Mat, S. Baharun, and P. Lestari, “PM10 black carbon and ionic species concentration of urban atmosphere in Makassar of South Sulawesi province, Indonesia,” *Atmos. Pollut. Res.*, vol. 5, no. 4, pp. 610–615, Oct. 2014, doi: 10.5094/APR.2014.070.
- [11] A. Aslam *et al.*, “Mitigation of particulate matters and integrated approach for carbon monoxide remediation in an urban environment,” *J. Environ. Chem. Eng.*, vol. 9, no. 4, p. 105546, 2021, doi: 10.1016/j.jece.2021.105546.
- [12] Y. Niaz, J. Zhou, A. Nasir, M. Iqbal, and B. Dong, “Comparitive study of particulate matter (PM10 and PM2.5) in Dalian-China and Faisalabad-Pakistan,” *Pakistan J. Agric. Sci.*, vol. 53, no. 1, pp. 97–106, 2016, doi: 10.21162/PAKJAS/16.3623.
- [13] P. Graphics Inc, “Ozone Therapy in Medicine and Dentistry,” 2008.
- [14] W. E. Cahyono, “Pengaruh Penipisan Ozon Terhadap Kesehatan Manusia,” *Semnas Penelitian, Pendidik. dan Penerapan MIPA*, pp. 208–214, 2005.
- [15] E. J. Emmett and M. C. Willis, “The Development and Application of Sulfur Dioxide Surrogates in Synthetic Organic Chemistry,” *Asian J. Org. Chem.*, vol. 4, no. 7, pp. 602–611, 2015, doi: 10.1002/ajoc.201500103.
- [16] T. M. Chen, J. Gokhale, S. Shofer, and W. G. Kuschner, “Outdoor air pollution: Nitrogen dioxide, sulfur dioxide, and carbon monoxide health effects,” *Am. J. Med. Sci.*, vol. 333, no. 4, pp. 249–256, 2007, doi: 10.1097/MAJ.0b013e31803b900f.
- [17] M. M. Ballari, Q. L. Yu, and H. J. H. Brouwers, “Experimental study of the NO and NO<sub>2</sub> degradation by photocatalytically active concrete,” vol. 161, no. 2, pp. 175–180, 2011, doi: 10.1016/j.cattod.2010.09.028.
- [18] X. Congresso, N. Igiic, L. Stato, P. Garzolini, and T. Wassermann, “XVIII Congresso Nazionale IGIIC – Lo Stato dell’Arte 18 –Palazzo Garzolini di

Toppo Wassermann, Udine, 29-31 ottobre 2020,” pp. 29–31, 2020.

- [19] “\_PT. ERM INDONESIA”.
- [20] D. A. Ihsani, A. Arifin, and M. H. Fatoni, “Klasifikasi DNA Microarray Menggunakan Principal Component Analysis (PCA) dan Artificial Neural Network (ANN),” *J. Tek. ITS*, vol. 9, no. 1, 2020, doi: 10.12962/j23373539.v9i1.51637.
- [21] A. Walford, “Raw Data:,” *Mult. Nature-Cultures, Divers. Anthropol.*, pp. 65–80, 2020, doi: 10.2307/j.ctv1850gsb.8.
- [22] D. A. Pisner and D. M. Schnyer, “Support vector machine,” *Mach. Learn. Methods Appl. to Brain Disord.*, pp. 101–121, 2019, doi: 10.1016/B978-0-12-815739-8.00006-7.
- [23] E. Bendersky, “Understanding gradient descent,” 2016, [Online]. Available: <https://eli.thegreenplace.net/2016/understanding-gradient-descent/>
- [24] C. Optical and D. Sensor, “GP2Y1010AU0F,” pp. 1–9, 2006.
- [25] ETC2, “MQ136 Semiconductor Sensor for Sulfur Dioxide,” pp. 2–4.

## **LAMPIRAN**

### **A. Kode program Proses Input Data ISPU**

```
import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import seaborn as sns

from datetime import datetime

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn import svm

from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, precision_score, recall_score, f1_score

import warnings

warnings.filterwarnings('ignore')


df1 = pd.read_excel('DATA ISPU BERDASARKAN STASIUN.xlsx',
sheet_name='DKI1')

df2 = pd.read_excel('DATA ISPU BERDASARKAN STASIUN.xlsx',
sheet_name='DKI2')

df3 = pd.read_excel('DATA ISPU BERDASARKAN STASIUN.xlsx',
sheet_name='DKI3')

df4 = pd.read_excel('DATA ISPU BERDASARKAN STASIUN.xlsx',
sheet_name='DKI4')
```

```
df5 = pd.read_excel('DATA ISPU BERDASARKAN STASIUN.xlsx',  
sheet_name='DKI5')
```

```
df_concat = pd.concat([df1,df2,df3,df4,df5])
```

```
df_concat
```

```
kategori = df_concat['Kategori'].value_counts().index
```

```
kategori_dict = { }
```

```
for kat in kategori:
```

```
    kat_t = str(kat)
```

```
    if kat_t not in kategori_dict.keys():
```

```
        kategori_dict[kat_t] = kat_t.lower()
```

```
kategori_dict
```

```
df_concat['Kategori'].replace(kategori_dict, inplace=True)
```

```
df_concat.Kategori.value_counts()
```

```
df_concat.Kategori.unique()
```

```
df_concat = df_concat.fillna(round(df_concat.mean()))
```

```
df_concat.info()
```

```
df_concat = df_concat.drop(['Max','Critical'],axis=1)
```

```
df_concat
```

```
df_concat["Kategori"].unique()
```

```
df_concat.Kategori.value_counts()
```

```
df_concat = df_concat[df_concat.Kategori != 'tidak ada data']
```

```

df_concat

df_concat = df_concat.fillna(round(df_concat.mean()))

df_concat.info()

df_concat = df_concat.dropna()

df_concat

df_concat = df_concat[:200]

df_concat

sns.countplot(df_concat['Kategori'])

plt.xlabel("Count of each Target class")

plt.ylabel("Target classes")

plt.show()

le = LabelEncoder()

df_concat['Kategori'] = le.fit_transform(df_concat['Kategori'])

df_concat

X_train = df_concat.drop(['Tanggal','Wilayah','Kategori'], axis=1)

y_train = df_concat['Kategori']

```

## **B. Kode program Proses Subsscribe Antares**

```

# from antares_http import antares

# antares.setDebug(False)

# antares.setAccessKey('16b15659024fd846:fb77a780be6d9042') // Access Key
Antares

# latestData = antares.get('star', 'Data')

# print(latestData['content'])

```

```
# import time

# waktu = []

# json = []

# runtime = 0

# try:

#     while True:

#         now = datetime.now()

#         latestData = antares.get('star', 'Data')

#         print(latestData['content'])

#         json.append(latestData['content'])

#         waktu.append(now)

#         time.sleep(900.0 - ((0) % 900.0))

#         if len(json) == 30:

#             break

#             #do_something()

#             #print("Hallo")

#         #animals.append(wild_animals)

# except KeyboardInterrupt:

#     pass


# import pandas as pd

# import csv
```

```

# json

# waktu = pd.to_datetime(waktu)

# df = pd.DataFrame(json)

# df['waktu'] = waktu

# df

# df.to_csv("Node1 D2.csv", index=False)

```

```
df = pd.read_csv('Node1 D2 Asli')
```

```
df
```

### **C. Kode program kondisi AQI setiap gas sensor**

```
// Gas Ozon
```

```
def get_O3_subindex(x):
```

```
    if x <= 50:
```

```
        return x * 50 / 50
```

```
    elif x <= 100:
```

```
        return 50 + (x - 50) * 50 / 50
```

```
    elif x <= 168:
```

```
        return 100 + (x - 100) * 100 / 68
```

```
    elif x <= 208:
```

```
        return 200 + (x - 168) * 100 / 40
```

```
    elif x <= 748:
```

```
        return 300 + (x - 208) * 100 / 539
```

```
    elif x > 748:
```



```

        return 400 + (x - 400) * 100 / 539

    else:

        return 0

df["O3_SubIndex"] = df["mq131"].apply(lambda x: get_O3_subindex(x))

df

// Gas Nitrogen Dioksida

def get_NO2_subindex(x):

    if x <= 40:

        return x * 50 / 40

    elif x <= 80:

        return 50 + (x - 40) * 50 / 40

    elif x <= 180:

        return 100 + (x - 80) * 100 / 100

    elif x <= 280:

        return 200 + (x - 180) * 100 / 100

    elif x <= 400:

        return 300 + (x - 280) * 100 / 120

    elif x > 400:

        return 400 + (x - 400) * 100 / 120

    else:

        return 0

df["NO2_SubIndex"] = df["no2"].apply(lambda x: get_NO2_subindex(x))

```

```
df
```

```
// Gas Partikulat Debu
```

```
def get_PM10_subindex(x):
```

```
    if x <= 50:
```

```
        return x
```

```
    elif x <= 100:
```

```
        return x
```

```
    elif x <= 250:
```

```
        return 100 + (x - 100) * 100 / 150
```

```
    elif x <= 350:
```

```
        return 200 + (x - 250)
```

```
    elif x <= 430:
```

```
        return 300 + (x - 350) * 100 / 80
```

```
    elif x > 430:
```

```
        return 400 + (x - 430) * 100 / 80
```

```
    else:
```

```
        return 0
```

```
df["PM10_SubIndex"] = df["gp2y"].apply(lambda x: get_PM10_subindex(x))
```

```
df
```

```
// Gas Karbon Monoksida
```

```
def get_CO_subindex(x):
```

```

if x <= 1:

    return x * 50 / 1

elif x <= 2:

    return 50 + (x - 1) * 50 / 1

elif x <= 10:

    return 100 + (x - 2) * 100 / 8

elif x <= 17:

    return 200 + (x - 10) * 100 / 7

elif x <= 34:

    return 300 + (x - 17) * 100 / 17

elif x > 34:

    return 400 + (x - 34) * 100 / 17

else:

    return 0

```

```
df["CO_SubIndex"] = df["mq7"].apply(lambda x: get_CO_subindex(x))
```

```
df
```

```
// Gas Sulfur Dioksida
```

```

def get_SO2_subindex(x):

    if x <= 40:

        return x * 50 / 40

    elif x <= 80:

        return 50 + (x - 40) * 50 / 40

```

```

elif x <= 380:

    return 100 + (x - 80) * 100 / 300

elif x <= 800:

    return 200 + (x - 380) * 100 / 420

elif x <= 1600:

    return 300 + (x - 800) * 100 / 800

elif x > 1600:

    return 400 + (x - 1600) * 100 / 800

else:

    return 0

```

```
df["SO2_SubIndex"] = df["mq136"].apply(lambda x: get_SO2_subindex(x))
```

```
df
```

#### **D. Kode program labeling AQI gas**

```
## AQI bucketing
```

```
def get_AQI_bucket(x):
```

```

    if x <= 50:

        return "Baik"

```

```

    elif x <= 100:

        return "Sedang"

```

```

    elif x <= 200:

        return "Tidak Baik"

```

```

    elif x <= 300:

        return "Sangat Tidak Baik"

```

```

elif x <= 400:

    return "Sangat Tidak Baik"

elif x > 400:

    return "Sangat Tidak Baik"

else:

    return np.NaN

```

```

df["AQI_calculated"] =
round(df[["SO2_SubIndex","CO_SubIndex","NO2_SubIndex",
"PM10_SubIndex","O3_SubIndex"]].max(axis = 1))

# df["AQI_calculated"] =
round(df[["SO2_SubIndex","CO_SubIndex","NO2_SubIndex",
"PM10_SubIndex","O3_SubIndex"]].mean(axis = 1))

df["AQI_bucket_calculated"] = df["AQI_calculated"].apply(lambda x:
get_AQI_bucket(x))

df

df_klasifikasi = df.drop(['mq131','mq136','mq7','no2','gp2y','waktu'],axis=1)

df_klasifikasi

le = LabelEncoder()

df['AQI_bucket_calculated'] = le.fit_transform(df['AQI_bucket_calculated'])

df['AQI_bucket_calculated'] = df['AQI_bucket_calculated'].replace(0,1)

df

df_klasifikasi =
df.drop(['mq131','mq136','mq7','no2','gp2y','waktu','AQI_calculated'],axis=1)

df_klasifikasi

```

```
X_test = df_klasifikasi.drop(['AQI_bucket_calculated'], axis=1)

y_test = df_klasifikasi['AQI_bucket_calculated']
```

#### **E. Kode program proses Standard Scaller**

```
scaler = StandardScaler()

X_test = scaler.fit_transform(X_test)

X_train = scaler.fit_transform(X_train)
```

#### **F. Kode program metode Support Vector Machine dan Parameter SVM**

```
from sklearn.svm import SVC

clf = SVC(kernel = 'rbf',gamma=1, C=1)

clf.fit(X_train, y_train)
```

```
from sklearn.svm import SVC

clf = SVC(kernel = 'rbf',gamma=1, C=1)

clf.fit(X_train, y_train)
```

```
_pred = clf.predict(X_test)

print('Accuracy Radial Basis Kernel:', y_pred)

def MAPE(Y_actual,Y_Predicted):

    mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100

    return mape

# Using MAPE error metrics to check for the error rate and accuracy level

SVR_MAPE= MAPE(y_test,y_pred)

print("MAPE: ",SVR_MAPE)
```

```

df_label = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})

mapping = {0: 'Tidak Baik', 1:'Sangat Tidak Baik'}

df_label = df_label.replace({'Real Values': mapping, 'Predicted Values':
mapping})

df_label

clf.score(X_test,y_test)

# melihat hasil tabel kebenaran confusion matrix

ax= plt.subplot()

y_pred = clf.predict(X_test)

# y_pred = y_pred.argmax(axis = 1)

# y_predht = (y_predht > 0.5)

cm = confusion_matrix(y_test, y_pred)

sns.heatmap(cm, annot=True, fmt='.2f', ax = ax); #annot=True to annotate cells

# labels, title and ticks

ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');

ax.set_title('Confusion Matrix');

```

### **G. Kode program Support Vector Machine dengan Principal Componen Analysis**

```

pca = PCA(n_components=5)

scaler = StandardScaler()

X_train_pca = pca.fit_transform(X_train)

X_test_pca = pca.transform(X_test)

X_train_pca = scaler.fit_transform(X_train)

```

```

X_test_pca = scaler.transform(X_test)

plt.figure(figsize=(8,6))

plt.scatter(X_train_pca[:,0],X_train_pca[:,1],c=y_train,cmap='plasma')

plt.xlabel('First principal component')

plt.ylabel('Second Principal Component')


print("Cumulative Variances (Percentage):")

print(np.cumsum(pca.explained_variance_ratio_ * 100))

components = len(pca.explained_variance_ratio_)

print(f'Number of components: {components}')

#

plt.plot(range(1, components + 1), np.cumsum(pca.explained_variance_ratio_ *
100))

plt.xlabel("Number of components")

plt.ylabel("Explained variance (%)")


from sklearn.model_selection import cross_val_score, train_test_split

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import StandardScaler, MinMaxScaler

# print(f'"X' shape: {X.shape}")

# print(f'"y' shape: {y.shape}")

pipeline = Pipeline([

    ('std_scaler', StandardScaler()),

```



)

```
_sc = StandardScaler()
```

```
_pca = PCA(n_components = components)
```

```
_model = svm.SVC()
```

```
SVM_model = Pipeline([
```

```
('std_scaler', _sc),
```

```
('pca', _pca),
```

```
('model', _model)
```

```
)
```

```
# train the model using the PCA components
```

```
SVM_model.fit(X_train_pca,y_train)
```

```
SVM_model.score(X_test_pca,y_test)
```

```
y_pred = clf.predict(X_test_pca)
```

```
print('Accuracy Radial Basis Kernel:', y_pred)
```

```
df_label = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
```

```
mapping = {0: 'Tidak Baik', 1:'Sangat Tidak Baik'}
```

```
df_label = df_label.replace({'Real Values': mapping, 'Predicted Values':  
mapping})
```

```
df_label
```

## **H. Kode program metode Support Vector Machine dengan SMOTE**

```
X_train = df_concat.drop(['Tanggal','Wilayah','Kategori'], axis=1)
```

```
y_train = df_concat['Kategori']
```

```
from imblearn.over_sampling import SMOTE

from collections import Counter

oversample = SMOTE()

X_train, y_train = oversample.fit_resample(X_train, y_train)


# summarize the new class distribution

counter = Counter(y_train)

print(counter)


sns.countplot(y_train)

plt.xlabel("Count of each Target class")

plt.ylabel("Target classes")

plt.show()


scaled = StandardScaler()

X_train = scaled.fit_transform(X_train)

clf = svm.SVC()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

y_pred

clf.score(X_test, y_test)

y_pred = clf.predict(X_test)

print('Accuracy Radial Basis Kernel:', y_pred)
```

```

df_label = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})

mapping = {0: 'Tidak Baik', 1:'Sangat Tidak Baik'}

df_label = df_label.replace({'Real Values': mapping, 'Predicted Values':
mapping})

df_label

```

## I. Kode program proses fitur Prediksi

```

df

# data['date'] = pd.to_datetime(data['date'],format='%Y-%m-%d') # date parse

# data['year'] = data['date'].dt.year # year

# data['year'] = data['year'].fillna(0.0).astype(int)

# data = data[(data['year']>0)]


df_predict_test =
df[['AQI_calculated','waktu']].groupby(["waktu"]).median().reset_index().sort
_values(by='waktu',ascending=False)

f,ax=plt.subplots(figsize=(15,10))

sns.pointplot(x='waktu', y='AQI_calculated', data=df)


df_predict_test = df[['AQI_calculated','waktu','AQI_bucket_calculated']]

# df["date"] = pd.to_datetime(df['waktu'])

df_predict_test.info()


# df_predict['waktu'].resample('15Min').sum()

```

```
# df_predict_test['waktu'] = pd.to_datetime(df_predict_test['waktu'])

# df_test = df_predict_test.set_index('waktu')

# df_test = df_test.resample('15Min').sum()

# df_test

n = df_predict_test.shape[0]

train_size = 0.65

features_dataframe = df_predict_test.sort_values('waktu')

train = df_predict_test.iloc[:int(n * train_size)]

test = df_predict_test .iloc[int(n * train_size):]

train.AQI_calculated.plot(figsize=(15,8), title= 'Minutes VARIATIONS',
fontsize=14)

test.AQI_calculated.plot(figsize=(15,8), title= 'Minutes VARIATIONS',
fontsize=14)

plt.show()
```

```

dd= np.asarray(train.AQI_calculated)

y_hat = test.copy()

y_hat['naive'] = dd[len(dd)-1]

plt.figure(figsize=(12,8))

plt.plot(train.index, train['AQI_calculated'], label='Train')

plt.plot(test.index,test['AQI_calculated'], label='Test')

plt.plot(y_hat.index,y_hat['naive'], label='Naive Forecast')

plt.legend(loc='best')

plt.title("Naive Forecast",fontsize=20)

```

```

plt.legend(["actual ", "predicted"])

plt.xlabel("Minutes",fontsize=20)

plt.ylabel("AQI",fontsize=20)

plt.tick_params(labelsize=20)

plt.show()

```

```

df_predict_test['waktu'] = pd.to_datetime(df_predict_test['waktu'])

df_predict_test

```

```

df_predict_test['hour'] = df_predict_test['waktu'].dt.hour

df_predict_test['minute'] = df_predict_test['waktu'].dt.minute

df_predict_test["period"] = df_predict_test["hour"].astype(str) +
df_predict_test["minute"].astype(str)

# df_predict_test = df_predict_test.set_index(['hour','minute'])

```

```
df_predict_test
```

```
# df_predict_test =  
df_predict_test[['AQI_calculated']].groupby(["AQI_calculated"]).mean().reset_index()
```

```
# df_test = df_test.dropna()
```

```
# dd=df_predict_test
```

```
# df_predict_test.describe()
```

```
temp = []
```

```
a = 0
```

```
j = df_predict_test['hour'][1]
```

```
for i in range(len(df_predict_test)):
```

```
    if j > df_predict_test.hour[i]:
```

```
        temp.append(a)
```

```
        j = j+1
```

```
        a = a+1
```

```
    else:
```

```
        a = a+1
```

```
        temp.append(a)
```

```
temp
```

```
df_predict_test['temp'] = temp
```

```
df_predict_test
```

```

df_predict_test =
df_predict_test.drop(['waktu','AQI_bucket_calculated','hour','minute','period'],
axis=1)

df_predict_test

# period = df_predict_test['period'].values

cols = ['temp']

y = df_predict_test['AQI_calculated']

x = df_predict_test[cols]

plt.scatter(x,y)

plt.show()


x = (x - x.mean()) / x.std()

x = np.c_[np.ones(x.shape[0]), x]

x

```

## **J. Kode program metode Gradient Descent**

```

# y = df_test['AQI_calculated']

alpha = 0.1 #Step size

iterations = 3000 #No. of iterations

m = y.size #No. of data points

np.random.seed(4) #Setting the seed

theta = np.random.rand(2) #Picking random values to start with

```

```

def gradient_descent(x, y, theta, iterations, alpha):

    past_costs = []

    past_thetas = [theta]

    for i in range(iterations):

        prediction = np.dot(x, theta)

        error = prediction - y

        cost = 1/(2*m) * np.dot(error.T, error)

        past_costs.append(cost)

        theta = theta - (alpha * (1/m) * np.dot(x.T, error))

        past_thetas.append(theta)

    return past_thetas, past_costs

past_thetas, past_costs = gradient_descent(x, y, theta, iterations, alpha)

theta = past_thetas[-1]

#Printing the results...

print("Gradient Descent: {:.2f}, {:.2f}".format(theta[0], theta[1]))

plt.title('Cost Function J')

plt.xlabel('No. of iterations')

plt.ylabel('Cost')

plt.plot(past_costs)

plt.show()

```



```

newB=[theta[0], theta[1]]

def rmse(y,y_pred):

    rmse=np.sqrt(sum(y-y_pred))

    return rmse

y_pred=x.dot(newB)

dt = pd.DataFrame({'Actual': y, 'Predicted': y_pred})

x=pd.concat([df_predict_test, dt], axis=1)

x

dt

## AQI bucketing

def get_AQI_bucket(x):

    if x <= 50:

        return "Baik"

    elif x <= 100:

        return "Sedang"

    elif x <= 200:

        return "Tidak Baik"

    elif x <= 300:

        return "Sangat Tidak Baik"

    elif x <= 400:

        return "Sangat Tidak Baik"

    elif x > 400:

```

```

        return "Sangat Tidak Baik"

    else:

        return np.NaN

dt["Label"] = round(dt[["Predicted"]].max(axis = 1))

# df["AQI_calculated"] =
round(df[["SO2_SubIndex", "CO_SubIndex", "NO2_SubIndex",
"PM10_SubIndex", "O3_SubIndex"]].mean(axis = 1))

dt["Label"] = dt["Predicted"].apply(lambda x: get_AQI_bucket(x))

dt

from sklearn import metrics

print(np.sqrt(metrics.mean_squared_error(y,y_pred)))

x_axis=x.temp
y_axis=x.Actual
y1_axis=x.Predicted

plt.figure(figsize=(12,8))

plt.plot(x_axis,y_axis)

plt.plot(x_axis,y1_axis)

plt.title("Actual vs Predicted",fontsize=20)

plt.legend(["actual ", "predicted"])

plt.xlabel("15 Minutes",fontsize=20)

```

```

plt.ylabel("AQI",fontsize=20)

plt.tick_params(labelsize=20)

plt.show()

df_predict_test['AQI_calculated'][[1]]

# x['AQI_calculated'] = x['AQI_calculated'].astype(int)

# x['Predicted'] = x['Predicted'].astype(int)

x.info()

testing = dt.to_dict(orient="records")

testing

```

#### **K. Kode program deploy hasil Antares HTTP**

```

from antares_http import antares

import time

antares.setDebug(True)

antares.setAccessKey('5d3da30498954179:9a122b7951f21cf5')

for key in enumerate(testing):

print(key)

antares.send(key, 'Air_Quality_Prediction', 'Air_Quality_Prediction')

time.sleep(60)

```

❖ **Link Source Code Lengkap Proses Preprocessing dan Predict Test**

Berikut tautan untuk source code data pengujian keseluruhan yang terdiri dari :

[https://github.com/rogmeongmeong/Preprocessing\\_with\\_Predict-Test.git](https://github.com/rogmeongmeong/Preprocessing_with_Predict-Test.git)

- Lampiran Kode Program Sensor
- Data CSV Hasil Pengambilan Sample Sensor
- Dataset ISPU