

**SISTEM PAKAR MENILAI KUALITAS UDARA KOTA SORONG
MENGGUNAKAN NAIVE BAYES DAN NATURAL LANGUAGE
PROCESSING BERBASIS WEB**



Disusun Oleh:

Kelompok 8

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH
SORONG
2025**

LEMBAR PENGESAHAN
SISTEM PAKAR MENILAI KUALITAS UDARA KOTA SORONG
MENGGUNAKAN NAIVE BAYES DAN NLP BERBASIS WEB

Disusun Oleh :

Desyia Tatuhe (202255202019)

Nurosida Sebualamo (202255202103)

Yulio Delvin Kambu (202355202164)



Telah Disetujui,

Sorong, ... November 2025

Mengetahui,

Dosen Pengampu Mata Kuliah Sistem Pakar

Muhammad Yusuf, S.Kom., M.Kom.

NIDN 1402098101

DAFTAR ISI

LEMBAR PENGESAHAN	ii
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah	5
1.5 Manfaat Penelitian	5
BAB II LANDASAN TEORI	7
2.1 Literatur Terkait	7
2.2 Landasan Teori	14
2.2.1 Kualitas Udara	14
2.2.2 Sistem Pakar	14
2.2.3 <i>Machine Learning</i>	15
2.2.4 <i>Naive Bayes</i>	16
2.2.5 <i>Natural Language Processing</i>	17
2.2.6 <i>Website</i>	18
2.2.7 <i>Transformers</i>	19

2.2.8 <i>Python</i>	21
2.2.9 <i>Visual Studio Code (VSC)</i>	22
2.2.10 <i>Unified Modeling Language</i>	23
2.2.11 <i>Flowchart</i>	25
2.2.12 <i>Extreme Programming</i>	27
2.2.13 <i>Blackbox Testing</i>	29
2.2.14 <i>Usability Testing</i>	29
2.3 Keaslian Penelitian	31

BAB III METODOLOGI PENELITIAN 37

3.1 Waktu & Lokasi Penelitian	37
3.1.1 Lokasi penelitian	37
3.1.2 Waktu Penelitian	37
3.2 Metode Penelitian	38
3.3 Kerangka Berpikir	39
3.4 Alur Penelitian	41
3.5 Tahapan Penelitian	42
3.5.1 Identifikasi masalah	42
3.5.2 Studi Literatur	42
3.5.3 Pengumpulan Data	42
3.5.4 Pengembangan <i>Naive Bayes</i>	43
3.5.5 Pengembangan Sistem	45

DAFTAR TABEL

Tabel 2. 1 Usecase Diagram	24
Tabel 2. 2 Activity Diagram	25
Tabel 2. 3 Flowchart	26
Tabel 2. 4 Keaslian Penelitian	32
Tabel 3. 1 Jadwal Penelitian	38

DAFTAR GAMBAR

Gambar 2. 1 Struktur Transformers	24
Gambar 2. 2 Extreme Programming	32
Gambar 2. 3 Blackbox Testing	33
Gambar 3. 1 Lokasi Penelitian	37
Gambar 3. 2 Kerangka Berpikir	40
Gambar 3. 3 Alur Penelitian	41
Gambar 3. 4 Tahapan Pengembangan <i>Naive Bayes</i>	43
Gambar 3. 5 Tahapan Pengembangan Sistem	45
Gambar 3. 6 Flowchart Sistem	47
Gambar 3. 7 Use Case Diagram	48
Gambar 3. 8 Activity Beranda	48
Gambar 3. 9 Activity Analisis Udara	49

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kualitas udara merupakan komponen penting dalam menjaga kesehatan masyarakat dan keberlanjutan lingkungan. Menurut standar kesehatan lingkungan, pencemaran udara yang tidak terkendali dapat meningkatkan risiko penyakit pernapasan, menurunkan kualitas hidup, dan memicu gangguan ekologis. Seiring pertumbuhan populasi dan aktivitas pembangunan, kota-kota di Indonesia menghadapi tantangan yang semakin besar dalam memantau serta menjaga kualitas udara agar tetap berada pada level aman (Khumaidi et al., 2020).

Kota Sorong sebagai salah satu pusat kegiatan ekonomi di Papua Barat Daya mengalami peningkatan signifikan pada aktivitas transportasi, industri, dan pembangunan infrastruktur. Kondisi ini berpotensi memperburuk kualitas udara, terutama pada wilayah dengan mobilitas tinggi. Di sisi lain, masyarakat dan pemangku kebijakan membutuhkan informasi kualitas udara yang akurat, cepat, dan mudah dipahami untuk mendukung pengambilan keputusan yang tepat, baik dalam penanggulangan pencemaran maupun mitigasi risiko kesehatan. Meskipun

demikian, sistem pemantauan kualitas udara di Kota Sorong masih terbatas, dan informasi yang tersedia sering kali belum dapat dipahami dengan baik oleh masyarakat umum. Minimnya sumber daya manusia yang memiliki keahlian dalam interpretasi data lingkungan juga menyulitkan proses penilaian kualitas udara secara cepat dan konsisten. Oleh karena itu, dibutuhkan sebuah solusi teknologi

yang mampu mengotomatisasi proses analisis sekaligus memberikan penilaian kualitas udara secara mudah, akurat, dan dapat diakses oleh masyarakat.

Salah satu pendekatan yang relevan untuk mengatasi tantangan tersebut adalah pengembangan sistem pakar, yaitu sistem yang dirancang untuk meniru cara kerja seorang pakar dalam menganalisis data dan memberikan kesimpulan berdasarkan aturan atau model tertentu (Marfalino et al., 2022). Sistem pakar memungkinkan proses diagnosa atau penilaian dilakukan secara otomatis, mengurangi ketergantungan pada manusia, serta memberikan hasil yang konsisten. Dalam pengembangannya, sistem pakar sering dikombinasikan dengan metode kecerdasan buatan (Artificial Intelligence/AI) untuk meningkatkan kemampuan analisis data. Salah satu algoritma sistem pakar yang populer untuk klasifikasi data adalah *Naive Bayes*, yang bekerja berdasarkan prinsip probabilistik dan teorema Bayes. *Naive Bayes* dikenal efektif dalam menangani data dengan jumlah fitur besar serta cocok untuk analisis berbasis teks atau data numerik yang memiliki pola klasifikasi tertentu (Suharyanto, 2022). Algoritma ini memiliki kelebihan berupa performa yang baik, waktu komputasi yang cepat, dan tingkat akurasi yang

stabil meskipun menggunakan asumsi independensi antar fitur. Di sisi lain, perkembangan teknologi Natural Language Processing (NLP) turut membuka peluang baru dalam pemrosesan data lingkungan. NLP memungkinkan sistem untuk memahami, mengekstraksi, dan mengolah informasi dari data teks seperti laporan sensor, catatan pengamatan, atau deskripsi parameter kualitas udara yang tersedia dalam format tidak terstruktur (Amien, 2022). Dengan menggabungkan NLP dan *Naive Bayes*, sistem pakar dapat mengolah data yang kompleks baik

berupa angka teknis maupun teks deskriptif, lalu memberikan penilaian kualitas udara secara otomatis. Kombinasi kedua teknologi ini sangat relevan untuk menghasilkan sistem yang tidak hanya cerdas tetapi juga adaptif terhadap variasi data nyata di lapangan.

Pada beberapa penelitian sebelumnya seperti yang telah dilakukan oleh Sugeng Dwi Budi Piantoro Dkk, yang berfokus pada menilai kualitas udara menggunakan algoritma *Naive Bayes* di daerah provinsi DKI Jakarta yang akan diaplikasikan pada sebuah sistem berbentuk website. Pada penelitian ini juga ditambahkan algoritma pembanding seperti K-Nearest Neighbor (KNN). Hasil penelitian menunjukkan bahwa Gaussian *Naive Bayes* memiliki performa yang lebih unggul dibandingkan KNN. GNB menghasilkan akurasi sebesar 90,57%, dengan nilai F1-score yang tinggi dan stabil di seluruh kategori, terutama kategori “Tidak Sehat” yang biasanya sulit diklasifikasikan. Sebaliknya, KNN hanya mencapai akurasi 86% dan menunjukkan performa yang sangat rendah pada kelas minoritas, terbukti dengan F1-score hanya 10% pada kategori “Tidak Sehat”. Hasil K-Fold Cross Validation memperkuat temuan ini, di mana GNB memperoleh rata-rata akurasi 88,86%, jauh lebih tinggi dibandingkan KNN yang hanya mencapai 76,87%. Uji McNemar juga menunjukkan bahwa GNB

memberikan lebih banyak prediksi benar dibandingkan KNN (Dwi et al., 2025). Selanjutnya pada penelitian yang dilakukan oleh Fitri Widiawati Dkk, yang berfokus pada klasifikasi tingkat kualitas udara di daerah Tangerang Selatan menggunakan algoritma *Naive Bayes*. Hasil penelitian menunjukkan bahwa *Naive Bayes* mampu mengklasifikasikan

kualitas udara dengan akurasi sebesar 79,38%, yang menandakan performa cukup baik meskipun belum optimal. Model memperlihatkan presisi sempurna (100%) pada kategori *Baik* dan *Sedang*, tetapi memiliki presisi rendah pada kategori *Tidak Sehat* dengan nilai 22,07%. Namun demikian, recall untuk kategori *Tidak Sehat* mencapai 100%, yang berarti model mampu mengenali seluruh kasus *Tidak Sehat* meskipun banyak prediksi salah terhadap kelas tersebut (Widiawati et al., 2023).

Berdasarkan uraian dari permasalahan diatas, penulis dapat menarik sebuah judul yaitu "**SISTEM PAKAR MENILAI KUALITAS UDARA KOTA SORONG MENGGUNAKAN NAIVE BAYES DAN NLP BERBASIS WEB**". Sistem ini diharapkan dapat menjadi alat bantu dalam meningkatkan kesadaran masyarakat terhadap kualitas udara, membantu pemerintah dalam analisis lingkungan, serta mendorong terciptanya wilayah perkotaan yang lebih sehat dan berkelanjutan.

1.2 Rumusan Masalah

Adapun rumusan masalah yang dapat diangkat berdasarkan permasalahan yang telah dijelaskan, sebagai berikut :

1. Bagaimana merancang dan membangun sebuah sistem pakar berbasis web yang mampu memproses data kualitas udara

menggunakan algoritma *Naive Bayes* dan menghasilkan hasil klasifikasi yang interaktif dan mudah dipahami oleh pengguna?

2. Berapa hasil akurasi algoritma *Naive Bayes* terhadap kualitas udara di Kota Sorong ketika dievaluasi menggunakan *Confusion Matriks*?

1.3 Tujuan Penelitian

Berikut sebagai jawaban rumusan masalah diatas dapat disimpulkan sebagai berikut :

1. Merancang dan membangun sebuah sistem pakar berbasis web yang mampu menghasilkan hasil klasifikasi yang interaktif dan ramah dengan pengguna.
2. Mengetahui hasil akurasi algoritma *Naive Bayes* dalam mengklasifikasikan kualitas udara di Kota Sorong melalui evaluasi menggunakan Confusion Matrix sehingga diperoleh gambaran tingkat ketepatan prediksi model

1.4 Batasan Masalah

Adapun beberapa batasan masalah dalam penelitian ini sebagai berikut :

1. Sistem hanya digunakan untuk menilai kualitas udara di kawasan Kota Sorong
2. Data yang digunakan berasal dari observasi di BMKG Kota Sorong
3. Bahasa pemrograman yang digunakan yaitu python
4. Indikator pengujian *Naive Bayes* menggunakan Confusion Matrix dan pengujian sistem web menggunakan blackbox testing

1.5 Manfaat Penelitian

Berikut beberapa manfaat penelitian yang dapat dikemukakan pada penelitian ini sebagai berikut :

1. Memberikan informasi kualitas udara yang lebih mudah dipahami oleh masyarakat, sehingga dapat meningkatkan kesadaran dan tindakan preventif terhadap risiko kesehatan akibat pencemaran udara.
2. Mendukung pemerintah daerah dalam pemantauan dan pengambilan keputusan terkait pengelolaan kualitas udara, melalui sistem pakar yang mampu memberikan hasil klasifikasi secara cepat dan akurat.

BAB II

LANDASAN TEORI

2.1 Literatur Terkait

Pada penelitian ini, penulis mengambil beberapa penelitian terdahulu yang digunakan sebagai refensi yaitu 5 jurnal bereputasi Sinta dan 2 jurnal bereputasi Scopus. berikut penjabaran sebagai berikut :

- 1. Penelitian dengan judul " Prediksi Kualitas Udara Menggunakan Algoritma K-Nearest Neighbor" oleh Amalia Dkk (2022) berfokus pada pengembangan model prediksi kualitas udara di DKI Jakarta berbasis data Indeks Standar Pencemar Udara (ISPU) serta penentuan nilai parameter K paling optimal pada algoritma K-Nearest Neighbor (KNN) untuk menghasilkan performa klasifikasi terbaik. Penelitian ini menggunakan metode data mining dengan teknik klasifikasi KNN, diawali dengan pengumpulan 450 data ISPU, dilanjutkan proses pre-processing berupa pembersihan data, penghapusan atribut yang tidak relevan, dan pengisian missing value, kemudian data dibagi menjadi 80% data latih dan 20% data uji menggunakan tujuh fitur utama (pm10, pm25, so2, co, o3, no2, max). Evaluasi model dilakukan dengan**

menggunakan confusion matrix untuk nilai $K = 3$ hingga $K = 9$, dengan pengukuran akurasi, presisi, recall, dan f-measure. Hasil penelitian menunjukkan bahwa nilai $K = 7$ merupakan parameter paling optimal, menghasilkan akurasi 96%, presisi 92%, recall 95%, dan f-measure 93%, dan model terbaik tersebut kemudian diimplementasikan dalam bentuk sistem prediksi kualitas udara berbasis web

menggunakan framework Flask, yang mampu memberikan hasil prediksi kualitas udara beserta rekomendasi sesuai input pengguna.

Pada penelitian ini peneliti menggunakan metode machine learning tradisional yaitu KNN sedangkan pada penelitian penulis, mencoba menggabungkan metode machine learning tradisional yaitu *Naive Bayes* sebagai algoritma prediksi probabilitas dengan NLP untuk memproses text (Amalia et al., 2023).

2. Penelitian dengan judul “**Penambahan Parameter PM2.5 dalam Prediksi Kualitas Udara: Long Short Term Memory**” oleh Lisa Kristiana dan Dian Miyanto. penelitian ini berfokus pada pengembangan model prediksi kualitas udara di Kota Bandung dengan menambahkan parameter PM2.5 yang sebelumnya belum digunakan dalam penelitian sejenis. Penelitian ini menggunakan metode Long Short-Term Memory (LSTM), yaitu jenis jaringan Recurrent Neural Network yang mampu menangkap ketergantungan jangka panjang pada data deret waktu, dengan tahapan pre-processing, analisis data eksploratif, visualisasi data, pembersihan data, serta serangkaian eksperimen hyperparameter (epoch dan batch size). Dataset kualitas udara yang digunakan berjumlah 2.831 data dengan fitur AQI, PM10, PM2.5, SO2, CO, O3, dan

NO2, yang dibagi menjadi 70% data latih dan 30% data uji. Hasil penelitian menunjukkan bahwa konfigurasi LSTM dengan batch size 16 dan epoch 500 menghasilkan performa paling optimal, ditunjukkan oleh nilai Root Mean Squared Error (RMSE) sebesar 4.0279

Pada penelitian ini, peneliti berfokus pada penambahan parameter PM 2,5 sebagai indikator pemicu untuk memprediksi kualitas udara menggunakan LSTM sedangkan penulis mencoba menggunakan dua metode yaitu Naïve Bayes dan NLP untuk melakukan prediksi kualitas udara (Kristiana & Miyanto, 2023).

3. Penelitian dengan judul "**Klasifikasi Kualitas Udara dengan Metode Naïve Bayes Berbasis Web**" oleh Dwi Sugeng Dkk, berfokus pada pembangunan sistem berbasis web untuk mengklasifikasikan kualitas udara menggunakan algoritma Gaussian Naive Bayes (GNB), serta membandingkan performanya dengan KNN. Dataset ISPU DKI Jakarta 2024 digunakan, melalui proses preprocessing seperti *mean imputation* dan pembersihan data. Model dievaluasi menggunakan akurasi, precision, recall, F1-score, McNemar test, dan 10-fold cross validation. Hasil penelitian menunjukkan bahwa Gaussian Naive Bayes lebih unggul, dengan akurasi 91%, F1-score kelas " Tidak Sehat" mencapai 88%, dan rata-rata cross validation 88,86%, sedangkan KNN hanya mencapai akurasi 86%. Sistem web yang dibangun berhasil diujikan secara blackbox dan mampu menampilkan klasifikasi kualitas udara secara akurat dan mudah diakses

Pada penelitian ini, peneliti mencoba memprediksi kualitas udara menggunakan *Naive Bayes* sedangkan penulis mencoba menggunakan dua metode yaitu *Naive Bayes* sebagai algoritma probabilitas dan NLP sebagai metode pemrosesan teks(Dwi et al., 2025).

4. Penelitian dengan judul “ Sistem Pakar Diagnosa Penyakit Pada Tanaman Semangka Merah Menggunakan Metode CNN Berbasis Android” oleh Muhammad Yusuf Dkk, berfokus penelitiannya adalah mengembangkan aplikasi Android yang mampu mendiagnosis penyakit tanaman semangka berdasarkan citra daun. Penelitian menyoroti masalah kurangnya pengetahuan petani tentang hama dan penyakit semangka serta keterbatasan akses terhadap pakar. Untuk menjawab masalah tersebut, peneliti membangun sistem deteksi otomatis menggunakan Convolutional Neural Network (CNN) dengan arsitektur VGG16, yang dipilih karena strukturnya yang sederhana, konsisten, dan terbukti efektif pada klasifikasi gambar. Dataset yang digunakan terdiri dari 300 gambar yang mencakup tiga jenis penyakit: Anthracnose, Downy Mildew, dan Mosaic Virus. Data dibagi dengan rasio 70% train, 20% test, dan 10% validasi, kemudian melalui proses preprocessing seperti resize dan augmentasi. Model CNN dilatih selama 30 epoch, dan hasil pelatihan menunjukkan akurasi train hingga 100% dan akurasi validasi 93,33%. Pengujian menggunakan Confusion Matrix dan F1-Score menunjukkan performa yang tinggi: pada Anthracnose diperoleh precision 100%, recall 100%, f1-score 100%; pada Downy

Mildew precision 100%, recall 94%, f1-score 97%; dan pada Mosaic Virus precision 96%, recall 100%, f1-score 98%. Secara keseluruhan, model mencapai akurasi 98%, menandakan sistem sangat efektif dalam mengenali penyakit. Model kemudian dikonversi ke format tflite untuk diintegrasikan ke aplikasi Android. Sistem diuji dengan blackbox testing dan mendapatkan skor usability 86%, menunjukkan aplikasi berfungsi baik dan mudah digunakan

Pada penelitian ini, peneliti menggunakan CNN sebagai metode utama dalam sistem pakar diagnosa penyakit sebuah tanaman sedangkan penulis menggunakan dua metode yaitu *Naive Bayes* sebagai metode probabilitas dan NLP sebagai metode pemrosesan teks untuk sistem pakar menilai kualitas udara (Yusuf et al., 2024).

5. Penelitian berjudul “ **Comparison of Accuracy Level of Certainty Factor Method and Bayes Theorem on Cattle Disease**” oleh Muhammad Riski Dkk, ini berfokus membandingkan akurasi dua metode diagnosis Certainty Factor (CF) dan Bayes Theorem (BT) untuk mendeteksi penyakit sapi di Distrik Okaba, Merauke. Latar belakang penelitian muncul dari keterbatasan tenaga medis hewan dan jauhnya akses layanan, sehingga diperlukan sistem pakar berbasis web yang memanfaatkan data sembilan penyakit dan 24 gejala dari hasil wawancara dengan pakar. Hasil pengujian menunjukkan bahwa CF jauh lebih akurat dan stabil, dengan akurasi penyakit cacing mencapai 98,79%, sedangkan BT hanya 73%. Pada penyakit lain, CF tetap berada pada kisaran 72– 95%, sementara BT lebih rendah pada 44– 81%. Kesimpulannya, CF lebih efektif untuk lingkungan dengan ketidakpastian tinggi dan data probabilistik terbatas. Peneliti

merekomendasikan pengembangan metode lanjutan seperti fuzzy logic, deep neural network, serta algoritma pencarian untuk peningkatan akurasi sistem pakar di masa depan

Pada penelitian ini, peneliti mencoba membandingkan dua metode yaitu certainty factor dan naïve bayes pada penyakit sapi sedangkan

penulis

disini

mencoba menggabungkan dua metode yaitu *Naive Bayes* dan NLP (Rizki et al., 2025).

6. Penelitian berjudul “ **Enhancing Semantic Inference in Healthcare Systems With Ontology-Enriched NLP**” oleh Wassim Jaziri dan Najla

Sassi berfokus pada membangun kerangka *semantic inference* yang mampu mengekstraksi, memahami, dan menalar informasi klinis dari EHR menggunakan gabungan NLP berbasis transformer (BioBERT, ClinicalBERT), ontologi medis (SNOMED CT, UMLS, TNM staging), serta rule-based reasoning seperti SWRL. Metodologi mencakup tiga komponen utama: (1) *clinical NLP adaptation* untuk NER, relasi entitas, deteksi negasi, temporal extraction, dan disambiguasi singkatan; (2) integrasi ontologi untuk pemetaan konsep standar; dan (3) *semantic reasoning* berbasis hybrid symbolic– statistical inference. Pada evaluasi terhadap 50.000 rekam medis onkologi, model menghasilkan kinerja tinggi: NER precision 94.2%, recall 92.8%, F1 93.5%, *relation extraction* F1 90.6%, *inference accuracy* 88.3%, *clinically relevant inference* 92.1%, serta kinerja pencarian MAP 0.873 dan NDCG 0.912. Uji pengguna menunjukkan 92% peningkatan akses informasi, 88% kepuasan pada

saran inferensi, dan pengurangan waktu review rekam medis sebesar 35%

Pada penelitian ini, peneliti menggunakan NLP untuk meningkatkan kinerja dari sebuah sistem pada bidang ontology, sedangkan penulis mencoba menggabungkan dua metode yaitu NLP dan *Naive Bayes* untuk menilai kualitas udara. (Jaziri & Sassi, 2025).

7. Penelitian dengan judul “ Detection of Anorexic Girls in Blog Posts

Written in Hebrew Using a Combined Heuristic AI and NLP Method”

oleh , Yaakov Hacohen-kerner, berfokus pada mendekripsi ciri-ciri tulisan dari remaja perempuan yang diduga anoreksia melalui blog berbahasa Ibrani, menggunakan kombinasi heuristic AI, text classification (TC), dan NLP. Peneliti membangun dataset 100 postingan positif (diduga anoreksia) dan 100 negatif, diverifikasi oleh pakar anoreksia. Mereka menguji berbagai metode: preprocessing dasar, lima algoritma ML klasik (SVM, Random Forest, MLP, Logistic Regression, *Naive Bayes*), tiga model RNN (SimpleRNN, LSTM, GRU), serta empat model BERT (AlephBERT, HeBERT, WikiBERT, Multilingual BERT). Selain itu, mereka mendefinisikan 28 feature sets (content-based, style-based, sentiment-based), lalu menerapkan dua pendekatan utama: hill-climbing dan heuristic k-out-of-n untuk memilih kombinasi fitur terbaik. Hasilnya, model terbaik menggunakan Random Forest dengan kombinasi 10 feature sets yang kemudian dioptimalkan melalui Mutual Information feature filtering dan parameter tuning, menghasilkan akurasi 90.63%, jauh lebih tinggi daripada baseline 79.75%. RNN dan BERT justru berkinerja lebih rendah karena dataset kecil

Pada penelitian ini peneliti menggunakan dua metode yaitu machine learning dan NLP, tetapi disini pengembangannya hanya berupa sebuah analisis. Sedangkan penulis mencoba menggabungkan dua metode yaitu *Naive Bayes* dan NLP untuk dijadikan lebih lanjut kedalam sebuah sistem berbasis web (Hacohen-kerner et al., 2022).

2.2 Landasan Teori

2.2.1 Kualitas Udara

Kualitas udara merupakan suatu ukuran yang menggambarkan kondisi atmosfer berdasarkan konsentrasi berbagai polutan yang ada di dalamnya, serta sejauh mana polutan tersebut mempengaruhi kesehatan manusia, ekosistem, dan lingkungan fisik. Secara akademik, kualitas udara didefinisikan sebagai tingkat kemurnian udara yang ditentukan melalui pengukuran parameter pencemar—baik pencemar primer maupun sekunder—seperti partikulat ($PM_{2.5}$, PM_{10}), karbon monoksida (CO), sulfur dioksida (SO_2), nitrogen dioksida (NO_2), ozon (O_3), dan senyawa organik volatil (VOC). Penilaian kualitas udara umumnya dinyatakan melalui indeks tertentu, seperti Air Quality Index (AQI), yang mengintegrasikan berbagai parameter polutan ke dalam satu nilai komprehensif untuk menunjukkan tingkat risiko terhadap kesehatan manusia dan kelestarian lingkungan (Khumaidi et al., 2020).

2.2.2 Sistem Pakar

Sistem pakar (*expert system*) merupakan salah satu cabang utama dari kecerdasan buatan yang bertujuan mereplikasi proses berpikir, penalaran, dan pengambilan keputusan seorang ahli pada bidang tertentu melalui suatu sistem komputer. Secara konseptual, sistem pakar adalah sistem yang dibangun dengan memanfaatkan *knowledge base* (basis pengetahuan) yang berisi kumpulan fakta, aturan, prosedur, dan

pengalaman pakar, serta *inference engine* (mesin inferensi) yang berfungsi melakukan penalaran logis layaknya seorang pakar untuk menghasilkan solusi, rekomendasi, atau diagnosis atas suatu masalah. Sistem pakar dirancang untuk bekerja pada domain yang sempit tetapi mendalam, sehingga mampu memberikan keputusan yang konsisten dan dapat dijelaskan (explainable) meskipun tanpa kehadiran pakar manusia. Dalam konteks praktis, sistem pakar memanfaatkan teknik representasi pengetahuan seperti aturan produksi (*if- then rules*), pohon keputusan, jaringan semantik, atau kerangka ketidakpastian seperti *Certainty Factor* dan *Bayesian reasoning*. Ketika diberikan fakta atau gejala sebagai input, sistem ini melakukan proses inferensi—baik *forward chaining* maupun *backward chaining*—untuk memperoleh kesimpulan terbaik berdasarkan pengetahuan yang ada (Marfalino et al., 2022).

2.2.3 Machine Learning

Machine Learning (pembelajaran mesin) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pengembangan algoritma dan model komputasional yang memungkinkan sistem untuk belajar secara otomatis dari data tanpa

harus diprogram secara eksplisit. Secara akademik, machine learning didefinisikan sebagai proses di mana komputer membangun pola, hubungan, atau representasi dari sekumpulan data melalui proses pelatihan (*training*), sehingga mampu melakukan prediksi, klasifikasi, atau pengambilan keputusan berdasarkan

data baru. Dalam machine learning, sistem mempelajari struktur data dengan meminimalkan kesalahan prediksi melalui metode statistik dan matematis. Pembelajaran ini umumnya dibagi menjadi tiga kategori utama: supervised learning (menggunakan data berlabel), unsupervised learning (mencari pola pada data tanpa label), dan reinforcement learning (belajar melalui mekanisme hadiah dan hukuman). Pendekatan-pendekatan tersebut memungkinkan machine learning bekerja secara adaptif, memperbaiki performanya seiring bertambahnya data, dan diterapkan dalam beragam domain seperti pengenalan citra, pemrosesan bahasa alami, deteksi anomalai, sistem rekomendasi, hingga prediksi berbasis data. Dengan demikian, machine learning menjadi fondasi penting dalam pengembangan sistem cerdas modern karena kemampuannya menangani data berskala besar, menemukan pola kompleks, dan menghasilkan keputusan yang mendekati atau bahkan melampaui performa manusia pada tugas-tugas spesifik (Nugroho et al., 2025).

2.2.4 Naive Bayes

Naive Bayes adalah algoritma klasifikasi berbasis probabilitas yang didasarkan pada Teorema Bayes dengan asumsi bahwa setiap

fitur atau atribut dalam data saling independen secara kondisional terhadap kelasnya. Disebut *naive* karena asumsi independensi ini pada umumnya sepenuhnya benar di dunia nyata, namun tetap menghasilkan performa klasifikasi yang efektif dan efisien pada berbagai jenis data. Secara matematis, *Naive Bayes*

menghitung probabilitas suatu kelas berdasarkan kemunculan fitur-fitur tertentu dalam data, kemudian memilih kelas dengan probabilitas paling tinggi sebagai hasil prediksi. Dalam implementasinya, *Naive Bayes* banyak digunakan untuk tugas *text classification*, seperti deteksi spam, analisis sentimen, dan klasifikasi dokumen, karena mampu menangani data berdimensi tinggi dan bekerja sangat cepat. Keunggulannya terletak pada kebutuhan data pelatihan yang relatif kecil, kesederhanaan model, serta kemampuan memberikan hasil yang stabil meskipun asumsi independensi tidak terpenuhi sepenuhnya. Dengan demikian, *Naive Bayes* menjadi salah satu metode probabilistik yang paling populer dalam pembelajaran mesin, terutama pada kasus klasifikasi dengan banyak fitur dan keterbatasan sumber daya komputasi (Suharyanto, 2022).

2.2.5 Natural Language Processing

Natural Language Processing (NLP) adalah bidang interdisipliner dalam kecerdasan buatan dan linguistik komputer yang berfokus pada pengembangan metode dan algoritma untuk memungkinkan komputer memahami, memproses, menganalisis, dan menghasilkan bahasa manusia secara alami. Secara akademik,

NLP bertujuan menjembatani komunikasi antara manusia dan mesin melalui representasi bahasa dalam bentuk yang dapat diolah komputer, sehingga sistem mampu mengekstraksi makna, menemukan pola linguistik, serta melakukan tugas-tugas bahasa seperti klasifikasi teks, penerjemahan, ekstraksi informasi, analisis sentimen, dan

pemahaman konteks. NLP bekerja melalui kombinasi teori linguistik, statistik, dan pembelajaran mesin, termasuk teknik pra-pemrosesan seperti tokenisasi, stemming, dan parsing, serta model modern seperti *transformer-based models* (misalnya BERT, GPT). Dengan pendekatan ini, NLP dapat mengubah data bahasa alami yang kompleks menjadi struktur yang terstandardisasi untuk dianalisis lebih lanjut. Bidang ini menjadi semakin penting dalam berbagai aplikasi, mulai dari chatbot, sistem rekomendasi, pencarian informasi, deteksi emosi, hingga analisis big data berbasis teks (Rivaldi & Wismarini, 2024).

2.2.6 Website

Website merupakan suatu kumpulan halaman web yang saling terhubung melalui tautan (hyperlink) dan dapat diakses secara online melalui jaringan internet menggunakan protokol HTTP atau HTTPS. Halaman-halaman tersebut disimpan pada sebuah web server dan ditampilkan kepada pengguna melalui web browser seperti Chrome, Firefox, atau Safari. Secara akademik, website didefinisikan sebagai platform informasi berbasis *hypertext* dan *hypermedia* yang memungkinkan penyajian konten digital – baik

berupa teks, gambar, audio, video, maupun elemen interaktif— secara terstruktur dan terorganisasi dalam sebuah sistem yang dapat diakses kapan saja dan dari mana saja. Website tidak hanya berfungsi sebagai media penyampaian informasi, tetapi juga sebagai sarana komunikasi, interaksi, dan transaksi dalam berbagai aktivitas, seperti

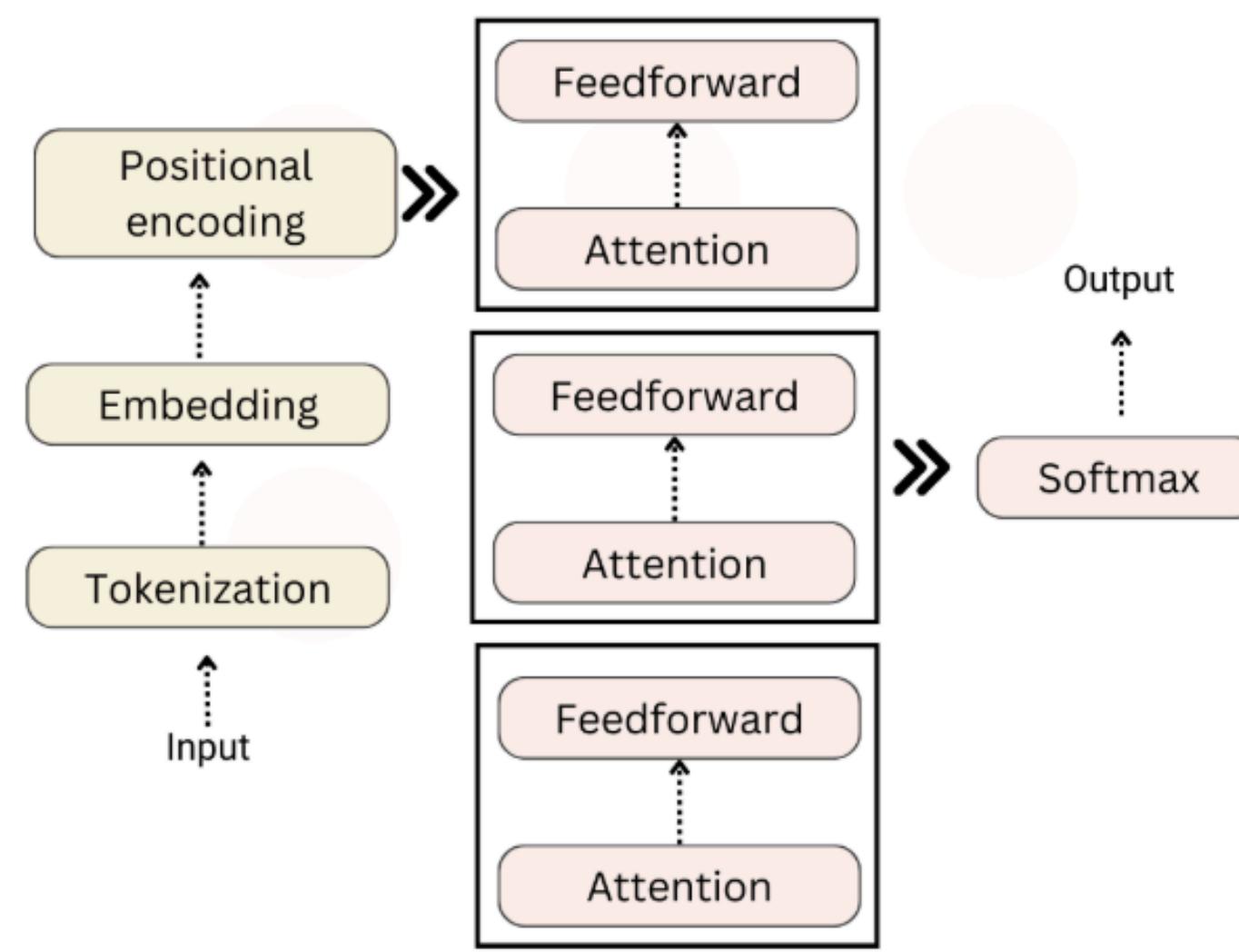
pendidikan, bisnis, layanan publik, hiburan, hingga manajemen data. Secara teknis, website dibangun menggunakan bahasa markup seperti HTML, gaya visual menggunakan CSS, serta logika dan interaktivitas melalui JavaScript atau bahasa server-side seperti PHP, Python, dan Node.js. Struktur website dapat bersifat statis maupun dinamis, bergantung pada kebutuhan pengguna dan kompleksitas sistem yang dikembangkan. Dengan perkembangan teknologi internet dan kebutuhan digital masyarakat, website telah berkembang menjadi komponen penting dalam sistem informasi modern, karena mampu menyediakan akses data secara cepat, efisien, dan terdistribusi. Website juga mendukung integrasi dengan basis data dan API, sehingga memungkinkan pengolahan informasi secara real-time dan mendukung berbagai layanan berbasis web dalam skala kecil maupun besar (Suri & Puspaningrum, 2020).

2.2.7 *Transformers*

Transformers adalah arsitektur model dalam kecerdasan buatan, khususnya di bidang *Natural Language Processing (NLP)* dan *Deep Learning*, yang diperkenalkan oleh Vaswani et al. (2017). Arsitektur ini dirancang untuk memproses data berurutan, seperti

teks, namun berbeda dari model sebelumnya (RNN, LSTM, dan GRU), Transformers tidak lagi bergantung pada urutan (sequential dependency). Sebaliknya, model ini memanfaatkan mekanisme self-attention yang memungkinkan setiap token dalam sebuah kalimat memberi perhatian (attention) kepada token lain

secara paralel, sehingga dapat memahami konteks global secara lebih efisien. Secara umum, Transformers terdiri dari dua komponen utama: encoder dan decoder, yang masing-masing tersusun dari beberapa *layer* yang berisi *multi-head self-attention* dan *feed-forward neural network*. Mekanisme *self-attention* ini memungkinkan model menilai tingkat relevansi antar kata sehingga mampu menangkap hubungan jangka pendek maupun jangka panjang tanpa mengalami masalah *vanishing gradient* seperti pada model sekuens tradisional. Selain itu, fleksibilitas dan skalabilitasnya membuat Transformers menjadi dasar dari berbagai model bahasa modern seperti BERT, GPT, dan T5.. Hal ini dapat dilihat pada struktur transformers di gambar 2.1



Gambar 2. 1 Struktur Transformers

Karena kemampuan pemrosesan parallelnya, Transformers memiliki keunggulan dalam hal skalabilitas, kecepatan pelatihan, dan efektivitas

pada dataset yang sangat besar. Arsitektur ini juga menjadi fondasi bagi banyak model bahasa modern seperti BERT, GPT, T5, RoBERTa, dan model generatif besar lainnya. Dengan demikian, Transformers telah menjadi standar utama dalam pengembangan model NLP modern, menghasilkan performa unggul pada tugas-tugas seperti penerjemahan mesin, klasifikasi teks, ekstraksi informasi, pemahaman konteks, hingga generasi bahasa alami (Arnanda Feza Raffa, 2024).

2.2.8 Python

Python adalah bahasa pemrograman tingkat tinggi (*high-level programming language*) yang bersifat interpreted, general-purpose, dan mendukung berbagai paradigma pemrograman seperti *object-oriented*, *procedural*, dan *functional programming*. Python diperkenalkan oleh Guido van Rossum pada tahun 1991 dan dirancang dengan filosofi yang menekankan *readability*, kesederhanaan sintaks, serta kemudahan pengembangan, sehingga memungkinkan penulisan kode yang bersih, terstruktur, dan mudah dipahami bahkan oleh pemula. Secara umum, Python banyak digunakan dalam penelitian dan pengembangan sistem karena

memiliki ekosistem pustaka yang sangat luas, meliputi *scientific computing* (NumPy, SciPy), *data analysis* (Pandas), *machine learning* (Scikit-learn), *deep learning* (TensorFlow, PyTorch), *web development* (Django, Flask), serta *automation* dan *scripting*. Sistem ekosistem yang kaya ini menjadikan Python salah satu bahasa yang paling dominan di bidang kecerdasan buatan,

data science, pemrosesan citra, dan NLP. Python bersifat *platform-independent*, memiliki komunitas global yang besar, serta mendukung integrasi dengan bahasa lain seperti C, C++, dan Java, sehingga menjadikannya bahasa yang fleksibel untuk proyek skala kecil hingga sistem kompleks. Dalam konteks akademik dan penelitian, Python dipilih karena kemudahan penggunaan, konsistensi performa, serta ketersediaan pustaka inovatif yang mempercepat proses eksperimen dan prototyping (Harahap, 2024).

2.2.9 Visual Studio Code (VSC)

Visual Studio Code (VSC) adalah sebuah *source-code editor* lintas platform yang dikembangkan oleh Microsoft dan dirancang untuk mendukung pengembangan perangkat lunak secara efisien melalui antarmuka yang ringan namun kaya fitur. Secara akademik, VSC dapat didefinisikan sebagai lingkungan pengembangan terpadu (*lightweight Integrated Development Environment/IDE*) yang menggabungkan fungsi-fungsi dasar editor teks dengan fitur-fitur pemrograman tingkat lanjut, seperti *syntax highlighting, intellisense* (auto-completion cerdas), *debugging, version control integration* melalui Git, serta dukungan ekstensi yang luas untuk berbagai

bahasa pemrograman. VSC dibangun menggunakan kerangka *Electron*, sehingga dapat berjalan pada berbagai sistem operasi seperti Windows, macOS, dan Linux. Fleksibilitasnya didukung oleh *marketplace* ekstensi yang memungkinkan pengguna

menambahkan dukungan bahasa, alat analisis kode, framework, hingga integrasi dengan platform cloud. Dalam konteks penelitian dan pengembangan perangkat lunak, VSC menjadi pilihan populer karena sifatnya yang ringan, mudah dikonfigurasi, dan mampu beradaptasi dengan beragam workflow pengembangan, mulai dari pemrograman web, data science, AI/ML, hingga pengembangan aplikasi lintas platform. Dengan fitur *terminal* terintegrasi, *debugger* bawaan, serta kemampuan untuk bekerja secara efisien dengan repositori Git, VSC tidak hanya berfungsi sebagai editor teks, tetapi juga sebagai alat produktivitas penting yang mempercepat proses penulisan, pengujian, dan pemeliharaan kode. Kombinasi kemudahan penggunaan, kemampuan kustomisasi, dan dukungan komunitas besar menjadikan Visual Studio Code salah satu alat utama dalam lingkungan pengembangan modern (Ardianto & Widiyatmoko, 2024).

2.2.10 Unified Modeling Language

Unified Modeling Language (UML), sering disingkat sebagai *UML*, adalah sebuah bahasa standar yang digunakan untuk

menjabarkan, menggambarkan, dan membangun perangkat lunak.

UML menjadi metode yang digunakan dalam pengembangan sistem berbasis objek, berperan sebagai alat bantu penting dalam proses pengembangan sistem (Tella et al., 2024).

a. *Use case Diagram*

Usecase diagram adalah representasi grafis yang menggambarkan beberapa atau semua use case, aktor, dan interaksi di antara mereka untuk menggambarkan sistem secara keseluruhan. Diagram ini memberikan gambaran singkat mengenai fungsi-fungsi yang ada pada sistem yang sudah dirancang, meskipun tidak memberikan detail rinci tentang penggunaan setiap use case (Ramdany, 2024). simbol use case dapat dilihat pada tabel 2.1

Tabel 2. 1 *Usecase Diagram*

Simbol	Keterangan
	Aktor: Mewakili peran orang, sistem yang lain atau alat Ketika berkomunikasi dengan use case
	Use case: Abstraksi dan interaksi antara sistem dengan aktor
	Association: Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi: Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya

	<p><i>Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya jika suatu kondisi terpenuhi</i></p>
---	---

b. Activity Diagram

Diagram aktivitas atau *activity diagram* sebagai penggambaran *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Serarawani et al., 2023). Dapat dilihat detailnya pada tabel 2.2

Tabel 2. 2 *Activity Diagram*

Simbol	Keterangan
	Status Awal: Sebuah <i>diagram</i> aktivitas memiliki sebuah status awal.
	Aktivitas: Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan/ <i>Decision</i> : Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan/ <i>Join</i> : Penggabungan dimana mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir: Status akhir yang dilakukan sistem, sebuah <i>diagram</i> aktivitas memiliki sebuah status

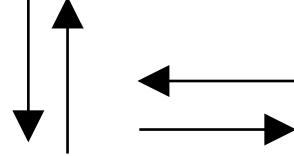
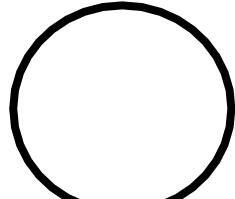
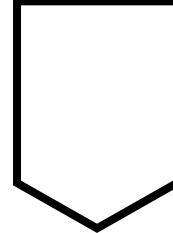
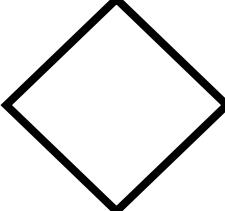
	akhir.
--	--------

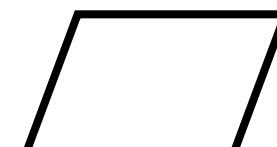
2.2.11 *Flowchart*

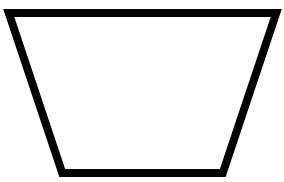
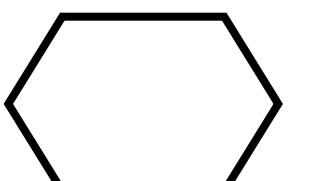
Flowchart adalah sebuah representasi visual berupa gambar atau diagram yang menunjukkan urutan atau langkah-langkah dalam sebuah

program serta hubungan antara proses-proses tersebut, yang disertai dengan pernyataan-pernyataan. *Flowchart* juga memiliki manfaat sebagai sarana komunikasi antara para pemrogram yang bekerja dalam tim pada suatu proyek (Tella et al., 2024). Simbol *flowchart* dapat dilihat pada tabel 2.3

Tabel 2. 3 *Flowchart*

Simbol	Keterangan
	<p>Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan <i>connecting line</i>.</p>
	<p>On-Page Reference Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama.</p>
	<p>Off-Page Reference Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda.</p>
	<p>Terminator Simbol yang menyatakan awal atau akhir suatu program.</p>
	<p>Process Simbol yang menyatakan suatu proses yang dilakukan komputer.</p>
	<p>Decision Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu ya dan tidak.</p>

	<p>Input/Output</p> <p>Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.</p>
---	---

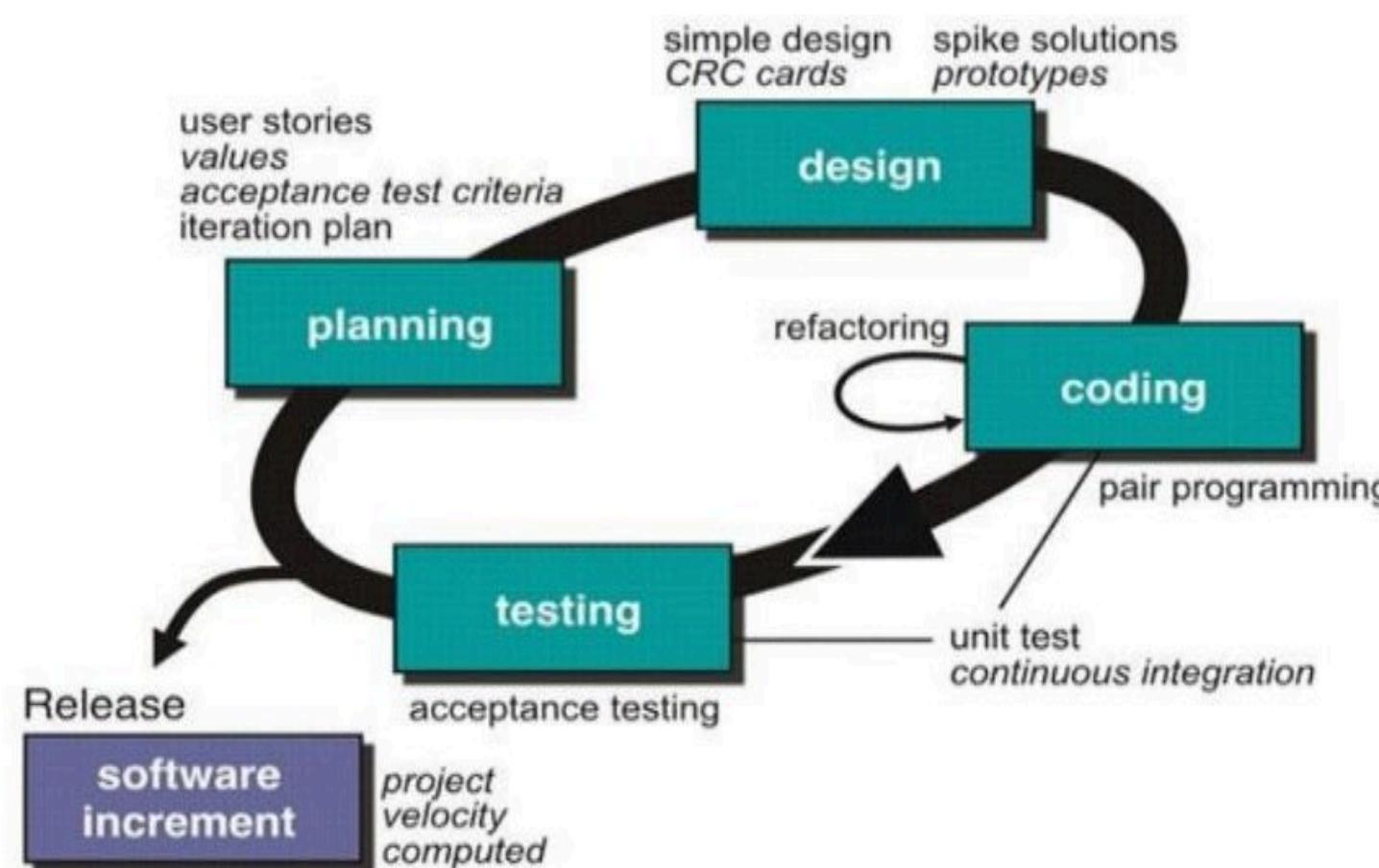
	<p>Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.</p>
	<p>Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.</p>
	<p>Predefine Proses Simbol untuk pelaksanaan suatu bagian (<i>sub-program</i>) atau <i>procedure</i>.</p>
	<p>Display Simbol yang menyatakan peralatan <i>output</i> yang digunakan.</p>
	<p>Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.</p>

2.2.12 Extreme Programming

Extreme Programming (XP) ialah metode pengembangan aplikasi yang ringan serta termasuk salah satu agile methods yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP merupakan pendekatan *agile* yang banyak digunakan karena mampu meningkatkan kualitas perangkat lunak sekaligus memenuhi kebutuhan pengguna secara cepat melalui iterasi yang pendek dan fleksibel. Metode ini menekankan kolaborasi tim, komunikasi yang

intens, serta kemampuan beradaptasi terhadap perubahan kebutuhan selama proses pengembangan. Tahapan-tahapan utama dalam *Extreme Programming* meliputi *Planning*, *Design*,

Coding, dan *Testing*, yang dilakukan secara berulang agar sistem yang dihasilkan lebih stabil, responsif, dan sesuai dengan kebutuhan pengguna (Soekarta et al., 2024). Tahapan *Extreme Programming* dapat dilihat pada gambar 2.2



Gambar 2. 2 *Extreme Programming*

Berikut ini penjelasan pada tahap-tahap yang dilakukan:

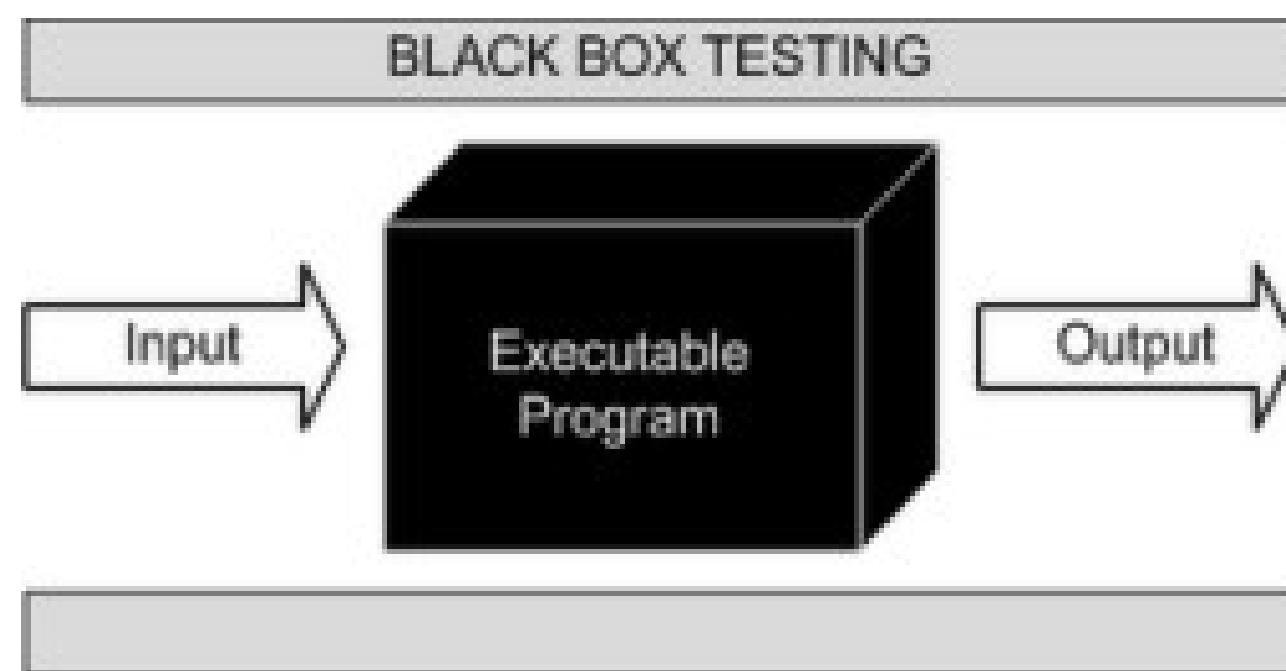
1. *Planning* atau perencanaan merupakan sebuah proses metodis yang dirancang untuk menggapai tujuan tertentu dan pengambilan keputusan untuk mendapatkan hasil yang diinginkan.
2. Desain tahap ini melibatkan tahapan perancangan yang terdapat beberapa proses yaitu *Activity diagram*, *Use case diagram*, *Sequence Diagram*, *Deployment diagram*.

3. *Coding* tahap ini melibatkan proses membuat program aplikasi pembelajaran yang menggunakan bahasa dari pemrograman *python*.

4. *Testing* Pengujian perangkat lunak ditujukan untuk pengujian semua elemen-elemen perangkat lunak yang dibuat.

2.2.13 *Blackbox Testing*

Black Box Testing merupakan pendekatan pengujian yang melibatkan observasi hasil eksekusi perangkat lunak dengan menggunakan berbagai data uji, tanpa memperhatikan struktur *internal* atau logika program. *Equivalence Class Partitioning*, salah satu bagian dari *black box testing*, melibatkan pembagian *domain input* program menjadi beberapa kelas data uji. Metode ini bertujuan untuk menguji penanganan kasus uji yang mewakili setiap kelas data secara *ideal*, dengan harapan dapat mengungkap beberapa kesalahan atau kegagalan dalam program tersebut (Ichsanudin & Yusuf, 2022). tahapan *blackbox testing* dapat dilihat pada gambar 2.3



Gambar 2. 3 *Blackbox Testing*

2.2.14 *Usability Testing*

Usability testing adalah ukuran dari sebuah karakteristik yang mengacu kepada bagaimana seorang pengguna dapat mempelajari dan

menggunakan sebuah sistem atau produk untuk memperoleh tujuan dan kepuasan terhadap penggunaannya. Berdasarkan ISO usability di dalam karya ilmiah didefinisikan sebagai “the effectiveness, efficiency, and satisfaction with which specified users can achieve specified goals in particular environments”, dengan kata lain usability ialah keefektifan, efisiensi, dan kepuasan yang dengannya pengguna tertentu dapat mencapai tujuan yang ditentukan dalam lingkungan tertentu (Valentino Adhy Nuantra et al., 2022).

$$Y = \frac{X}{\text{Skor ideal}} \quad (1)$$

Keterangan :

Y = Nilai Presentase yang dicari (... ... %)

X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi

$(\Sigma = N \times R)$.

N = Nilai setiap jawaban

R = Frekuensi

Skor ideal = Nilai tertinggi dilakukan jumlah sampel.

2.3 Keaslian Penelitian

Sistem Pakar Menilai Kualitas Udara Kota Sorong Menggunakan *Naive Bayes* dan NLP Berbasis Web

Tabel 2. 4 Keaslian Penelitian

No	Judul	Penelitian, Media Publikasi Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1.	Prediksi Kualitas Udara Menggunakan Algoritma K-Nearest Neighbor	Adinda Amalia , Ati Zaidiah, Ika Nurlaili Isnainiyah JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)	Penelitian ini bertujuan mengembangkan model prediksi kualitas udara di DKI Jakarta berbasis data Indeks Standar Pencemar Udara (ISPU).	Hasil penelitian menunjukkan bahwa nilai $K = 7$ merupakan parameter paling optimal,menghasil kan akurasi 96%, presisi 92%, recall 95%, dan f-measure 93%,	Berdasarkan penelitian ini, saran yang dapat diberikan yaitu berupa mengganti algoritma pengolahan data ke stack tech atau meningkatkan basis aplikasi dari web ke android	Pada penelitian penulis akan membuat sistem pakar tetapi menggabungkan NLP dan <i>Naive Bayes</i> sebagai algoritma pengolahan data.

2.	Penambahan Parameter PM2.5 dalam Prediksi Kualitas Udara: Long Short Term Memory	Lisa Kristiana Mind, Dian Miyanto (Multimedia Artificial Intelligent Networking Database) Journal MIND	Penelitian ini bertujuan pada pengembangan model prediksi kualitas udara di Kota Bandung dengan menambahkan parameter PM2.5 yang sebelumnya belum digunakan dalam penelitian sejenis	Hasil penelitian menunjukkan bahwa konfigurasi LSTM dengan batch size 16 dan epoch 500 menghasilkan performa paling optimal, ditunjukkan oleh nilai Root Mean Squared Error (RMSE) sebesar 4.0279.	Berdasarkan penelitian ini, saran yang dapat diberikan yaitu berupa mengganti algoritma pengolahan data ke stack tech atau mengembangkan ke arah sebuah sistem berbasis web.	Pada penelitian penulis, akan membuat sistem pakar tetapi menggabungkan NLP dan Naive Bayes sebagai algoritma pengolahan data.
----	--	--	--	--	--	--

3.	<p>Klasifikasi Kualitas Udara dengan Metode <i>Naive Bayes</i> Berbasis Web</p>	<p>Sugeng Dwi Budi Priantoro, M. Ghofar Rohman M. Ghofar Rohman, Moh Rosidi Zamroni RABIT: Jurnal Teknologi dan Sistem Informasi Univrab, Volume 10, Nomor 2, Tahun 2025</p>	<p>Membangun sistem berbasis web untuk mengklasifikasikan kualitas udara menggunakan algoritma Gaussian <i>Naive Bayes</i> (GNB)</p>	<p>Hasil penelitian menunjukkan bahwa Gaussian <i>Naive Bayes</i> lebih unggul, dengan akurasi 91%, F1-score kelas “ Tidak Sehat” mencapai 88%, dan rata-rata cross validation 88,86 %</p>	<p>Berdasarkan penelitian ini, saran yang dapat diberikan yaitu berupa mengganti algoritma pengolahan data ke stack tech atau meningkatkan basis aplikasi dari web ke android</p>	<p>Pada penelitian penulis, akan membuat sistem pakar tetapi menggabungkan NLP dan <i>Naive Bayes</i> sebagai algoritma pengolahan data.</p>
4.	<p>Sistem Pakar Diagnosa Penyakit Pada Tanaman Semangka Merah Menggunakan</p>	<p>Muhammad Yusuf, Dewi Astria Faroek, Moh Saddam S Pattanang, Anggun M, Ristanti Salam JTS: Jurnal Teknik, Vol.</p>	<p>Penelitian ini mengembangkan aplikasi Android yang mampu mendiagnosa penyakit tanaman semangka</p>	<p>Hasil dari penelitian ini berupa sebuah aplikasi android yang dapat mendeteksi penyakit pada</p>	<p>Berdasarkan penelitian ini, saran yang dapat diberikan yaitu berupa mengganti algoritma pengolahan data ke stack tech yang lebih</p>	<p>Pada penelitian penulis, akan membuat sistem pakar tetapi menggabungkan NLP dan <i>Naive Bayes</i> sebagai algoritma</p>

	Metode CNN Berbasis Android	13, No. 1, Tahun 2024	berdasarkan citra daun	tanaman semangka merah dengan akurasi validasi di angka 93%.	terbaru.	pengolahan data.
5.	Comparison of Accuracy Level of Certainty Factor Method and Bayes Theorem on Cattle Disease	Muhammad Rizki Setyawan, Fajar Rahardika Bahari Putra La Jupriadi Fakhri ILKOM Jurnal Ilmiah Vol. 16, No. 3, Tahun 2024	Membandingkan akurasi dua metode diagnosis Certainty Factor (CF) dan Bayes Theorem (BT) untuk mendekripsi penyakit sapi di Distrik Okaba, Merauke	Hasil pengujian menunjukkan bahwa CF jauh lebih akurat dan stabil, dengan akurasi penyakit cacing mencapai 98,79%, sedangkan BT hanya 73%. Pada penyakit lain, CF tetap berada pada kisaran 72– 95%,	Berdasarkan penelitian ini, saran yang dapat berikan berupa implementasi ke dalam suatu basis sistem agar dampaknya dapat digunakan oleh orang banyak.	Pada penelitian penulis, akan mengimplementasikan NLP dan Naive Bayes untuk menilai kualitas udara ke dalam suatu sistem berbasis web

				sementara BT lebih rendah pada 44– 81%		
6.	Enhancing Semantic Inference in Healthcare Systems With Ontology-Enriched NLP	Wassim Jaziri, And Najla Sassi IEEE Access, Volume 13, Tahun 2025	Tujuan dari penelitian ini yaitu membangun kerangka semantic inference yang mampu mengekstraksi, memahami, dan menalar informasi klinis dari EHR	Hasil penelitian menunjukkan model menghasilkan kinerja tinggi: NER precision 94.2%, recall 92.8%, F1 93.5%, relation extraction F1 90.6%, inference accuracy 88.3%, clinically relevant inference 92.1%, serta kinerja	Berdasarkan penelitian ini, saran yang dapat diberikan berupa mungkin menggabungkan beberapa model NLP agar data yang dihasilkan dapat lebih akurat lagi	Pada penelitian penulis, akan mengimplementasikan NLP dan Naive Bayes untuk menilai kualitas udara ke dalam suatu sistem berbasis web

				pencarian MAP 0.873 dan NDCG 0.912. Uji pengguna menunjukkan 92% peningkatan akses informasi, 88% kepuasan pada saran inferensi, dan pengurangan waktu review rekam medis sebesar 35%.		
7.	Detection of Anorexic Girls in Blog Posts Written in Hebrew Using a	Yaakov Hacohen-Kerner, Natan Manor, Michael Goldmeier, Eytan Bachar	Mendeteksi ciri-ciri tulisan dari remaja perempuan yang diduga anoreksia melalui blog	Hasilnya, model terbaik menggunakan Random Forest dengan	Berdasarkan penelitian ini, saran yang dapat diberikan yaitu berupa implementasi model	Pada penelitian penulis, akan mengimplementasikan NLP dan Naive Bayes untuk menilai

	Combined Heuristic AI and NLP Method	IEEE Access, Volume 10, Tahun 2022	berbahasa Ibrani, menggunakan kombinasi heuristic AI, text classification (TC), dan NLP	kombinasi 10 feature sets yang kemudian dioptimalkan melalui Mutual Information feature filtering dan parameter tuning, menghasilkan akurasi 90.63%, jauh lebih tinggi daripada baseline 79.75%. RNN dan BERT justru berkinerja lebih rendah karena dataset kecil	yang sudah dilatih ke dalam suatu basis sistem	kualitas udara ke dalam suatu sistem berbasis web
--	--------------------------------------	------------------------------------	---	---	--	---

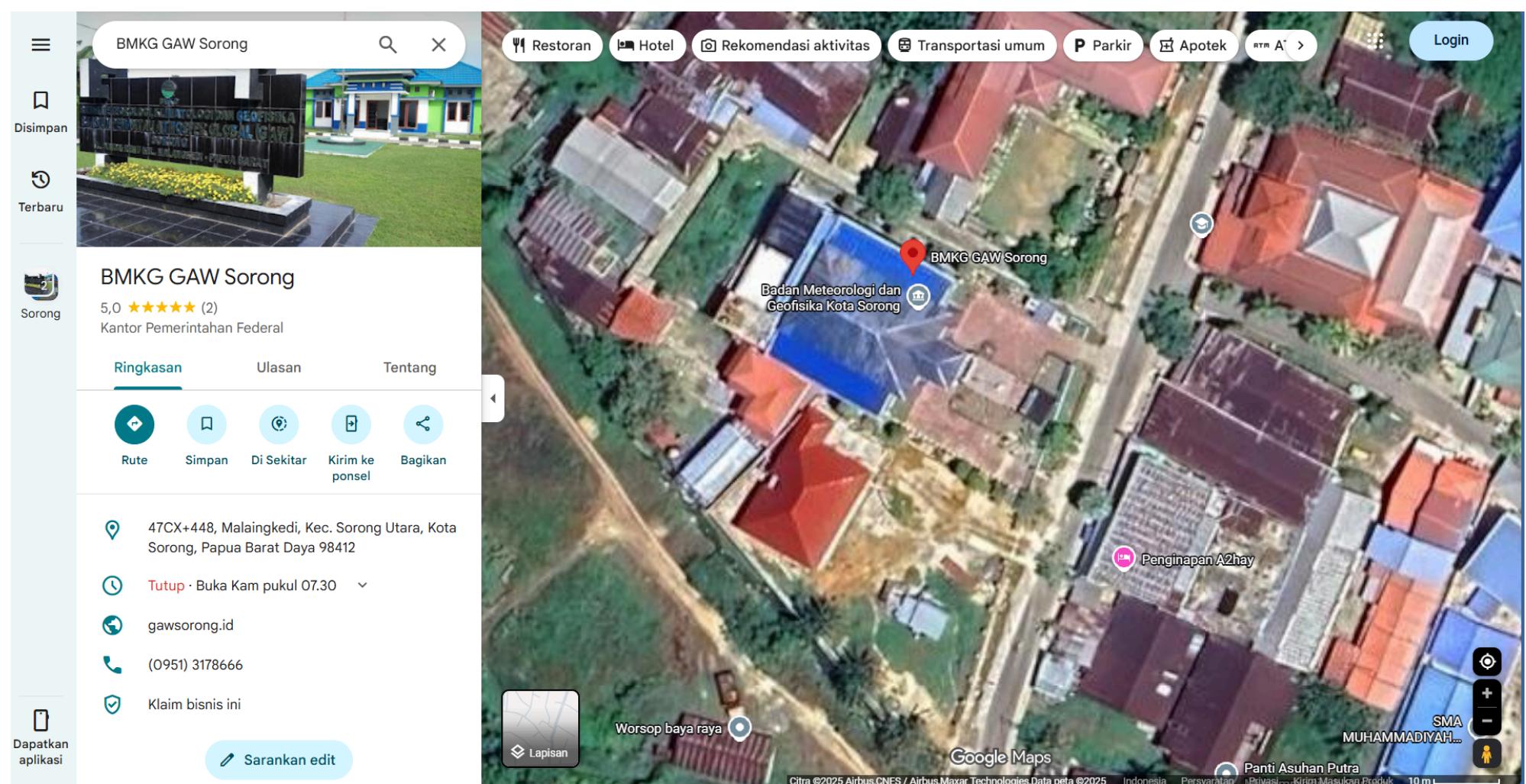
BAB III

METODOLOGI PENELITIAN

3.1 Waktu & Lokasi Penelitian

3.1.1 Lokasi penelitian

Lokasi yang digunakan pada penelitian ini berada di Kantor BMKG yang berada di kilo 8, berdekatan dengan Sekolah AL-Izzah, Kota Sorong, Papua Barat Daya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.1



Gambar 3. 1 Lokasi Penelitian

3.1.2 Waktu Penelitian

Waktu penggeraan penelitian ini kurang lebih dilakukan sekitar 3 bulan. Periode tersebut mencakup pengajuan judul sampai dengan

tahap implementasi dan pengujian sistem Untuk lebih jelasnya dapat

dilihat pada tabel 3.1

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Tahun 2025											
		November				Desember				Januari			
		Minggu		Minggu		Minggu		Minggu		Minggu		Minggu	
	Tanggal	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengajuan Judul												
2	Penyusunan Proposal												
3	Pengajuan proposal penelitian												
4	Perizinan penelitian												
5	Studi Literatur												
6	Wawancara												
7	Observasi												
8	Mendesain Kerangka sistem												
9	Pengkodean (Coding)												
10	Implementasi dan Pengujian												

3.2 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode pengolahan data yang menggabungkan *Naive Bayes* sebagai

metode untuk menentukan probabilistik tingkat kualitas udara dan NLP

sebagai

metode

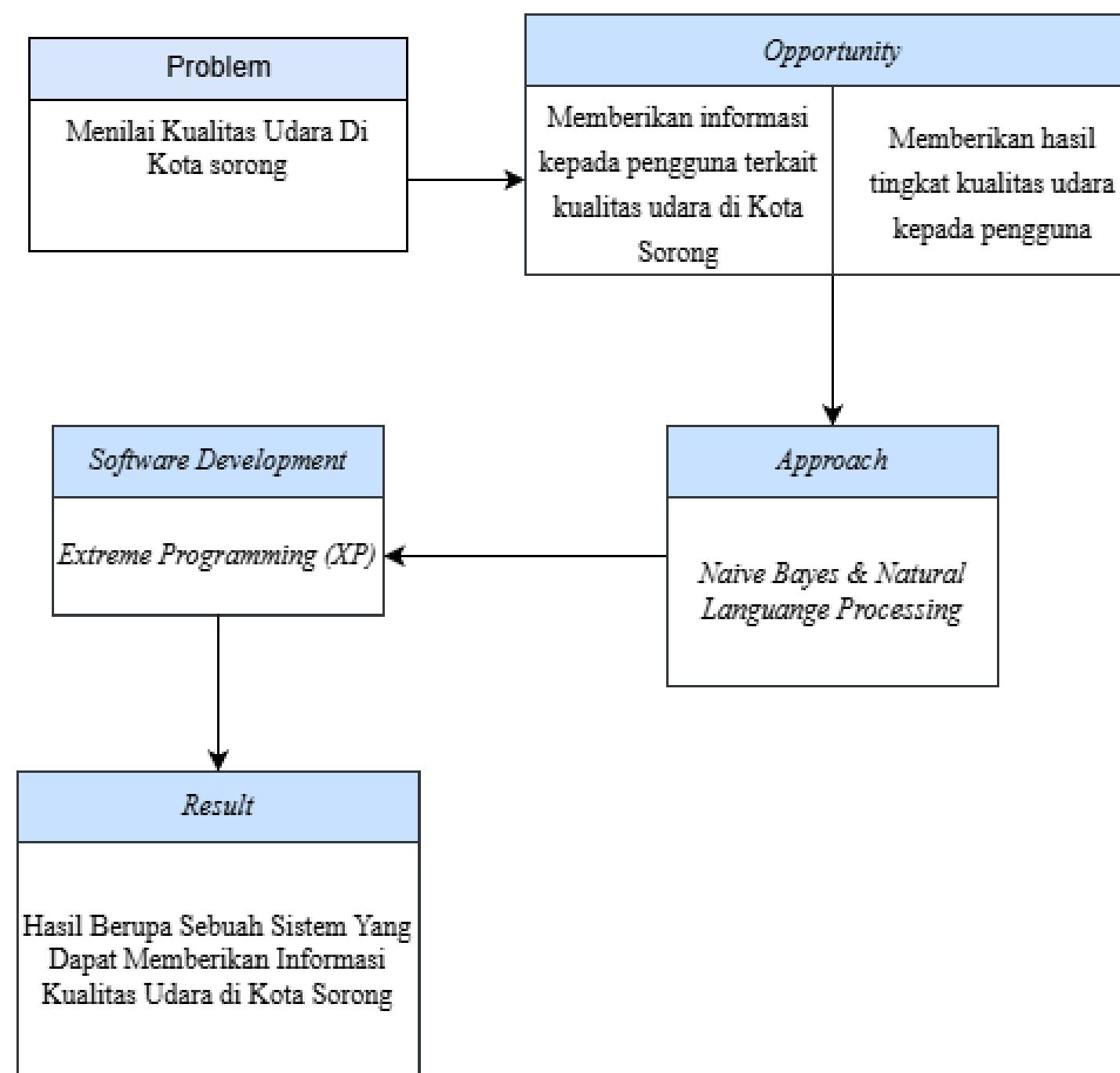
pengolahan teks, kemudian akan diimplementasikan pada suatu sistem dengan basis web yang dikembangkan menggunakan metode SDLC extreme programming. *Naive Bayes* digunakan untuk menghitung probabilitas tingkat kualitas udara berdasarkan parameter pencemar seperti CO₂, SO₂, NO₂, CO, dan O₃. Algoritma ini menentukan kategori kualitas udara (baik, sedang, tidak sehat, dan seterusnya) dengan memilih kelas yang memiliki probabilitas paling tinggi. Sementara itu, metode Natural Langunage Processing digunakan untuk memproses jawaban dari pengguna, sehingga nantinya jawaban dari pengguna dapat dianalisa oleh *Naive Bayes* sebagai model prediktor utama dalam menentukan tingkat kualitas udara. Hasil pemrosesan dari *Naive Bayes* dan *Natural Languange Processing* kemudian akan menghasilkan suatu output berupa hasil tingkat kualitas udara dan saran mengenai kualitas udara tersebut.

3.3 Kerangka Berpikir

Subjek penelitian ini bertujuan untuk merancang sebuah sistem untuk menilai kualitas udara di kota sorong dengan menggabungkan *Naive Bayes* dan NLP. Berikut penjabaran kerangka berpikir yang digunakan pada penilitian ini :

1. Kerangka berpikir dimulai dari permasalahan utama, yaitu belum tersedianya sistem yang dapat menilai dan memberikan informasi mengenai kualitas udara di Kota Sorong. Kondisi ini membuka peluang untuk menghadirkan sebuah solusi yang mampu menyajikan informasi kualitas udara secara cepat, akurat, dan mudah dipahami oleh masyarakat. Peluang tersebut mencakup dua aspek: menyediakan informasi kualitas udara yang relevan serta memberikan hasil penilaian tingkat polusi berdasarkan kondisi yang dirasakan pengguna.
2. Untuk menangani peluang tersebut, digunakan pendekatan *machine learning* melalui algoritma Naive Bayes sebagai prediktor untuk menentukan probabilitas dari kualitas udara dan teknik *Natural Language Processing* (NLP) sebagai metode utama dalam mengolah input pengguna. Proses pengembangan sistem dilakukan dengan metodologi *Extreme Programming* (XP) yang bersifat iteratif, fleksibel, dan memungkinkan penyesuaian cepat selama pembangunan sistem. Melalui pendekatan teknis dan metode pengembangan ini, diperoleh hasil berupa sistem berbasis web yang

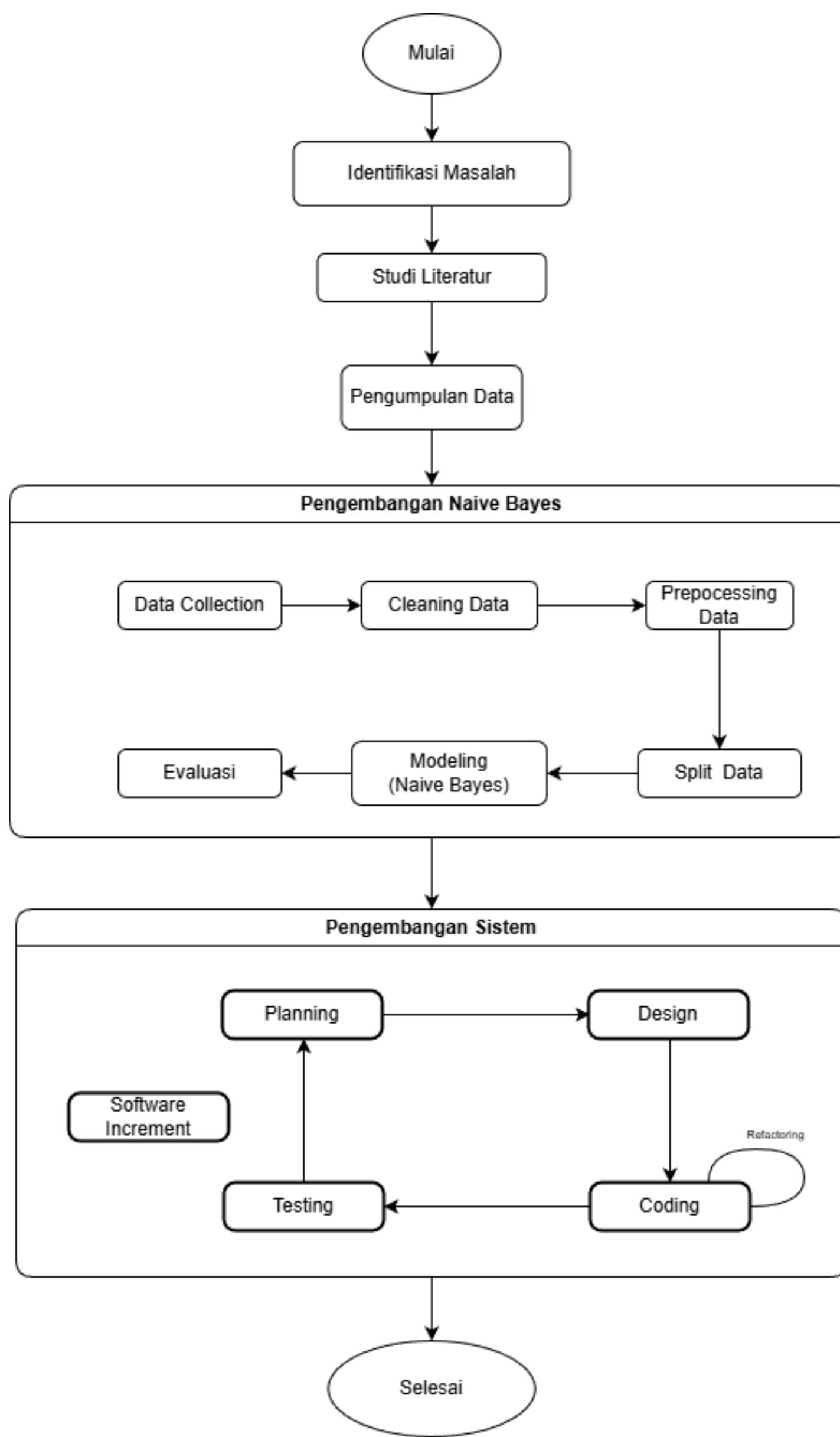
mampu memberikan informasi kualitas udara di Kota Sorong secara efektif.



Gambar 3. 2 Kerangka Berpikir

3.4 Alur Penelitian

Proses penelitian akan digambarkan melalui sebuah diagram alur agar dapat mempermudah menyampaikan informasi terhadap langkah-langkah yang akan dilakukan pada Gambar 3.3



Gambar 3. 3 Alur Penelitian

3.5 Tahapan Penelitian

3.5.1 Identifikasi masalah

Tahap identifikasi masalah pada penelitian ini diangkat dari banyaknya polusi udara yang terjadi di Kota Sorong yang

dikarenakan Kota Sorong sendiri merupakan kota ekonomi di provinsi papua barat daya. Masalah polusi ini juga menjadi semakin parah jika berada di wilayah dengan mobilitas yang tinggi. Oleh sebab itu penulis berupaya untuk membuatkan suatu website yang dapat memberikan pengguna informasi terkait kualitas udara sehingga pengguna dapat terbantu dalam menjalankan aktivitas sehari-hari.

3.5.2 Studi Literatur

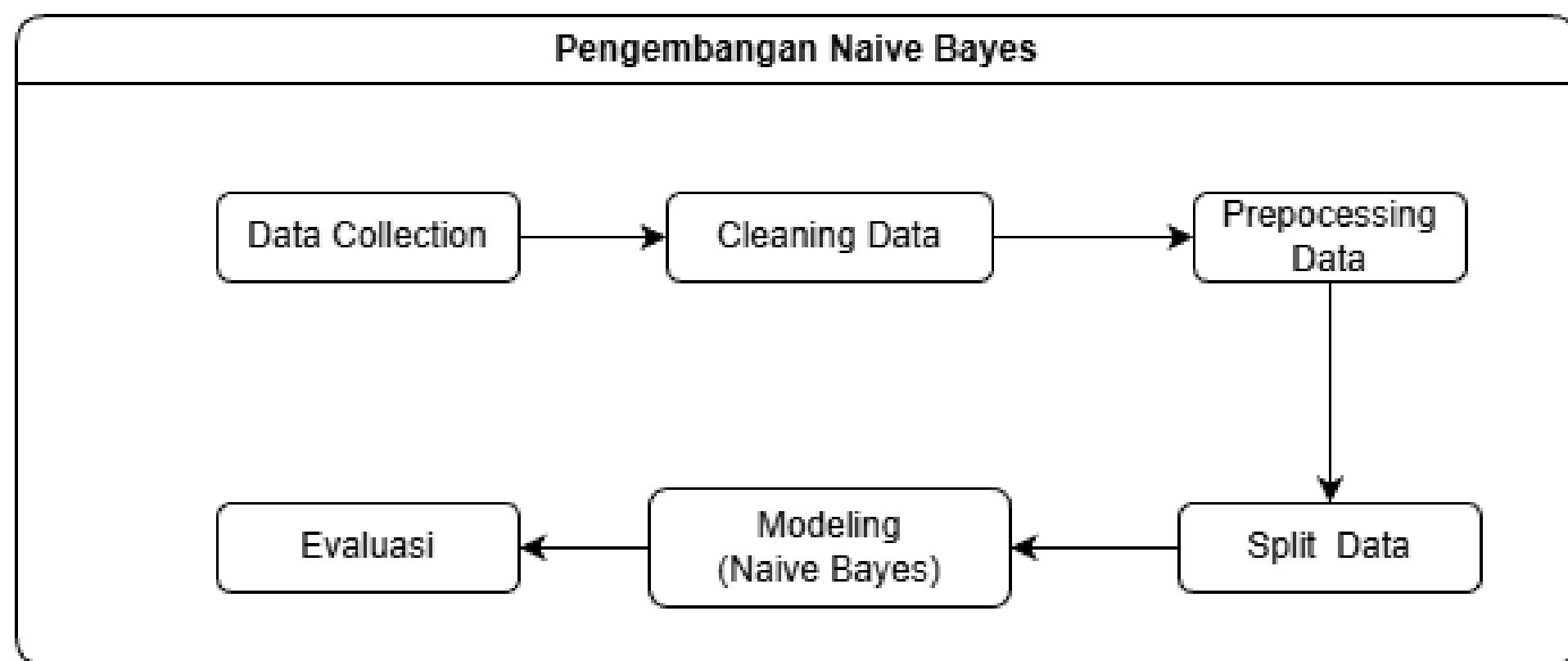
Studi literatur dilakukan dengan mengumpulkan sumber-sumber keilmuan yang mendukung penelitian ini. Pada Tahapan studi literatur hal yang dipelajari yaitu yang berkaitan dengan sistem rekomendasi yang digunakan dalam pengembangan sistem pada buku, jurnal, skripsi maupun sumber lain yang terdahulu guna mengetahui kelebihan serta kekurangannya, yang dapat menjadi acuan untuk pengembangan sistem ini.

3.5.3 Pengumpulan Data

Teknik pengumpulan data pada penelitian ini dilakukan dengan dua cara yaitu observasi dan wawancara di kantor BMKG yang

berada di Kilo 8 Kota Sorong. Penulis juga akan meminta data berupa tingkat polusi yang terdiri dari CO_2 , SO_2 , NO_2 , CO , dan O_3 . Data ini nantinya akan diproses menggunakan algoritma *Naive Bayes* untuk nantinya akan di klasifikasikan ke dalam 3 kelas yaitu kualitas udara baik, sedang, dan tidak sehat. Setelah itu akan disimpan ke dalam bentuk model yang siap digunakan sebagai prediktor utama.

3.5.4 Pengembangan *Naive Bayes*



Gambar 3. 4 Tahapan Pengembangan *Naive Bayes*

Berikut penjelasan dari Gambar 3.4

1) Data Collection

Tahap pertama adalah mengumpulkan data-data yang telah kita kumpulkan pada tahap pengumpulan lewat wawancara dan observasi dalam bentuk csv. Data tersebut

kemudian akan dikategorikan ke dalam 3 kelas yaitu baik, sedang dan buruk.

2) Cleaning Data

Data yang telah kita kumpulkan kemudian akan dibersihkan untuk menghilangkan tanda baca dan spasi yang berlebihan. Proses ini berguna untuk menghilangkan noise dalam data agar nantinya saat dilatih tidak mengalami ambigu dalam mengklasifikasi data.

3) Preprocessing Data

Tahap ketiga adalah tahap preprocessing, yaitu proses menyiapkan data agar layak digunakan dalam pelatihan model. Pada tahap ini dilakukan beberapa langkah penting seperti menangani missing values (nilai kosong), memastikan seluruh data berada dalam format numerik yang konsisten, serta menerapkan standarisasi atau normalisasi. Tujuan utama preprocessing adalah menghasilkan data yang bersih, seragam, dan optimal sehingga algoritma dapat mempelajari pola dengan lebih akurat dan stabil. Dengan data yang telah diproses dengan

baik, proses training model menjadi lebih efektif dan risiko kesalahan prediksi dapat diminimalkan

4) Splitting Data

Pada tahapan ini data akan displit menjadi data train dan data evaluasi yang masing-masing dengan ratio 8:2. Tujuan dari tahapan ini agar mesin dapat belajar pada data train dan akan dievaluasi seberapa mesin sudah paham pada data evaluasi.

5) Modeling

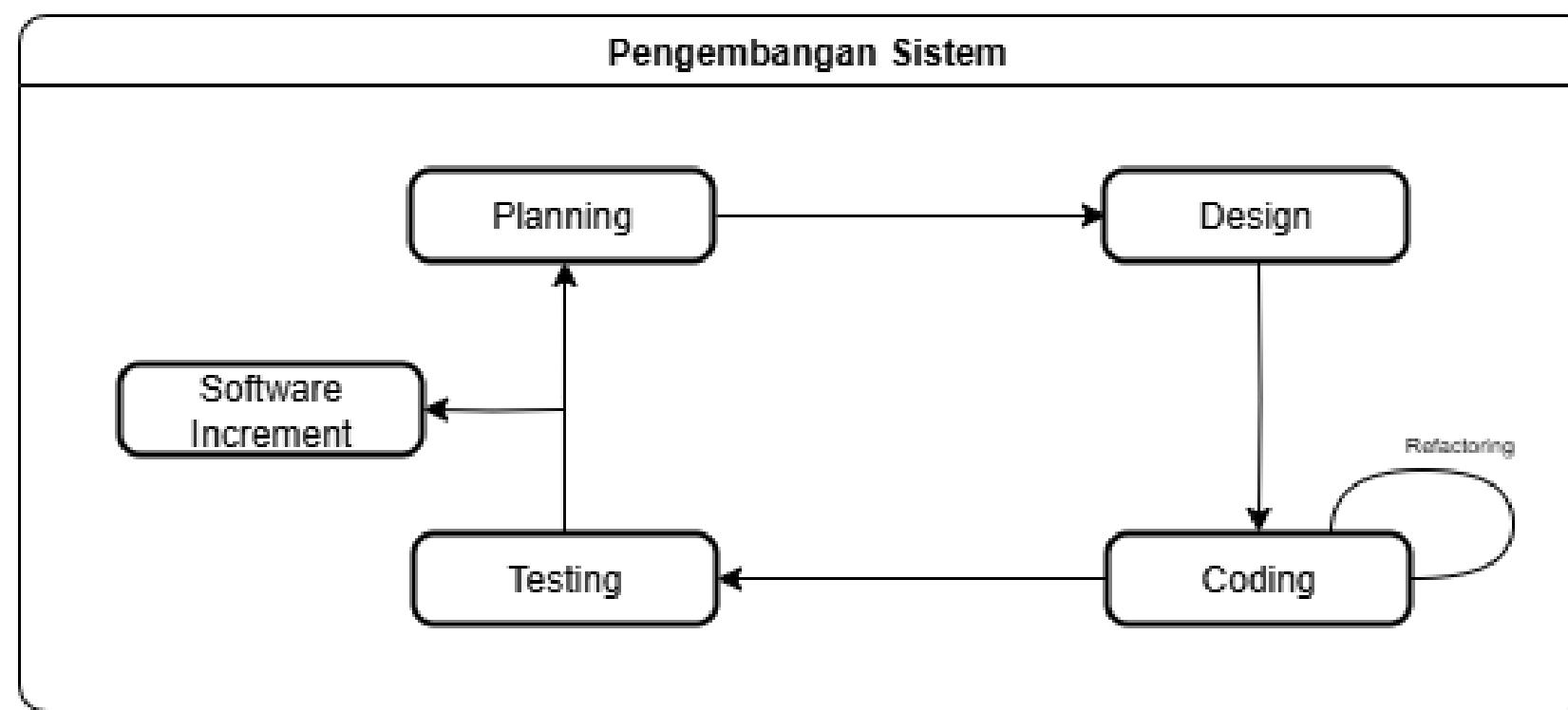
Tahapan kelima yaitu modeling. Pada tahapan ini data train yang sudah dibagi akan di latih oleh model *Naive Bayes*, khususnya *Gaussian Naïve Bayes*.

6) Evaluasi

Data yang telah ditrain akan dievaluasi pada tahap ini. Tools yang digunakan pada tahap ini adalah *confusion matrix*. Tools ini akan mengevaluasi model yang telah dilatih dengan data evaluasi yang telah dibagi sebelumnya pada tahapan split data. Tahapan ini bertujuan agar bisa mengetahui seberapa baik model belajar pada data train. Indikator yang akan evaluasi

pada *confusion matrix* berupa akurasi, *precision*, *recall*, dan *F-1 Score*.

3.5.5 Pengembangan Sistem



Gambar 3. 5 Tahapan Pengembangan Sistem

Berikut penjelasan detail dari Gambar 3.5

1) Planning

Dalam tahap penelitian ini, dilakukan analisis terhadap kebutuhan fungsional dan non-fungsional dari sistem yang akan dikembangkan

a. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional untuk memahami berbagai proses yang dapat dijalankan oleh sistem dan untuk menentukan siapa saja yang dapat memanfaatkan sistem uang sedang dibangun sebagai berikut :

- 1) Sistem memiliki 1 jenis pengguna
- 2) Pengguna dapat memberikan hasil kualitas udara setelah menjawab beberapa pertanyaan dari sistem

b. Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras dan perangkat lunak.

1) Perangkat Keras (*Hardware*)

Perangkat keras yang diperlukan dalam pengembangan sistem rekomendasi berbasis website yaitu :

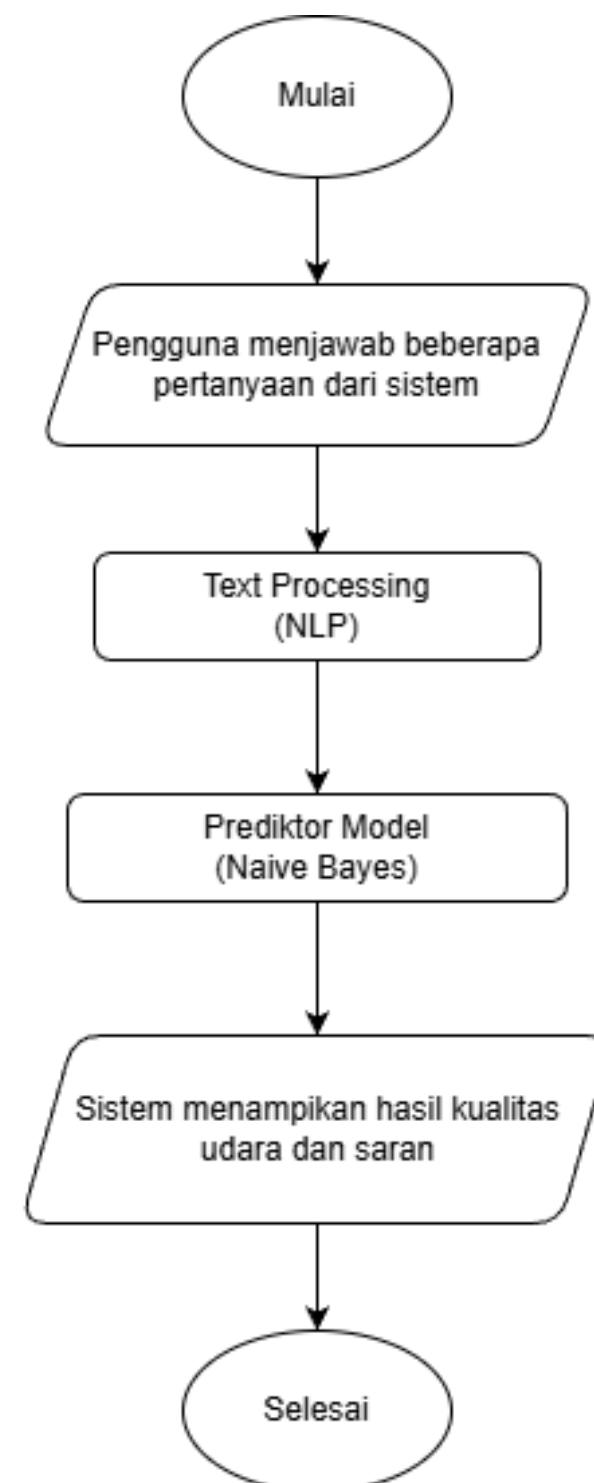
- a) *Laptop Lenovo IdeaPad 3 14ALC6*
- b) *AMD Ryzen 7 5700U With Radeon Graphics*
- c) *RAM 8 GB*
- d) *SSD Micron MTFDHBA512QFD*

2) Perangkat Lunak (*Software*)

- a) *Microsoft Windows 11 Home Single Language*
- b) *Visual Studio Code*
- c) *Chrome*

- d) *Draw.io*
- e) *Microsoft Office*
- 2) Design
- Pada fase Design ini dilakukan visualisasi *flowchart*, *usecase diagram*, *activity diagram*, dan *user interface*. Tujuannya untuk memperjelas gambaran alur pengembangan sistem yang dilakukan sebagai berikut :

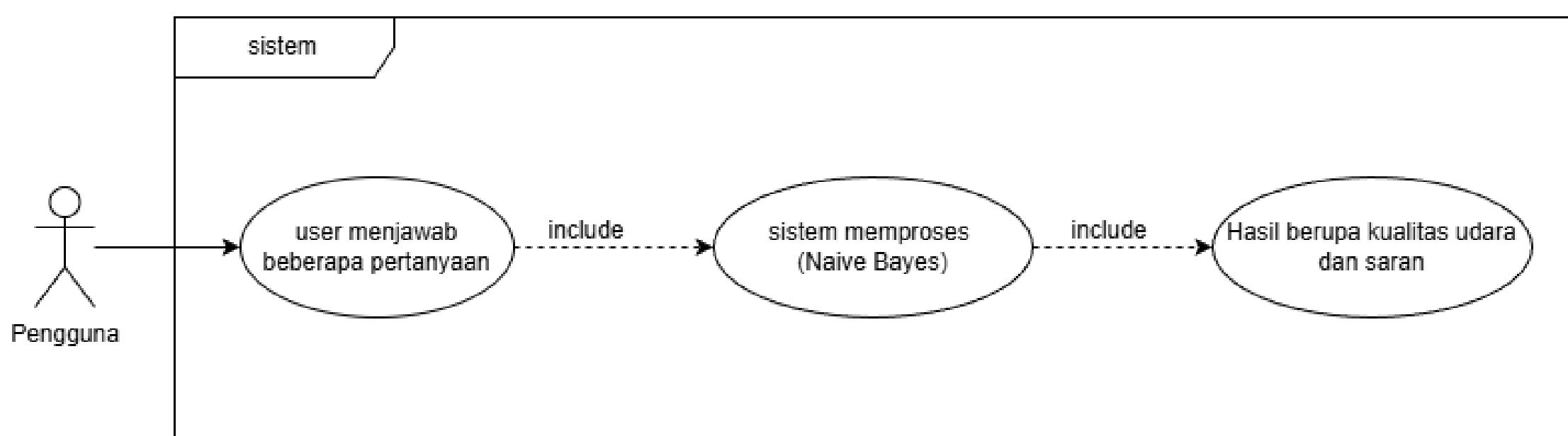
a. *Flowchart Sistem*



Gambar 3. 6 *Flowchart Sistem*

Pada Gambar 3.6 dapat dilihat bahwa sistem dimulai dengan pengguna menjawab beberapa pertanyaan dari sistem kemudian jawaban-jawaban dari pengguna ini akan diproses didalam sistem menggunakan NLP agar nantinya bisa dimengerti oleh *Naive Bayes* sebagai model prediksi. Setelah itu sistem akan menampilkan hasil kualitas udara beserta detail dan saran-saran terhadap hasil kualitas udara

b. *Use Case Diagram*

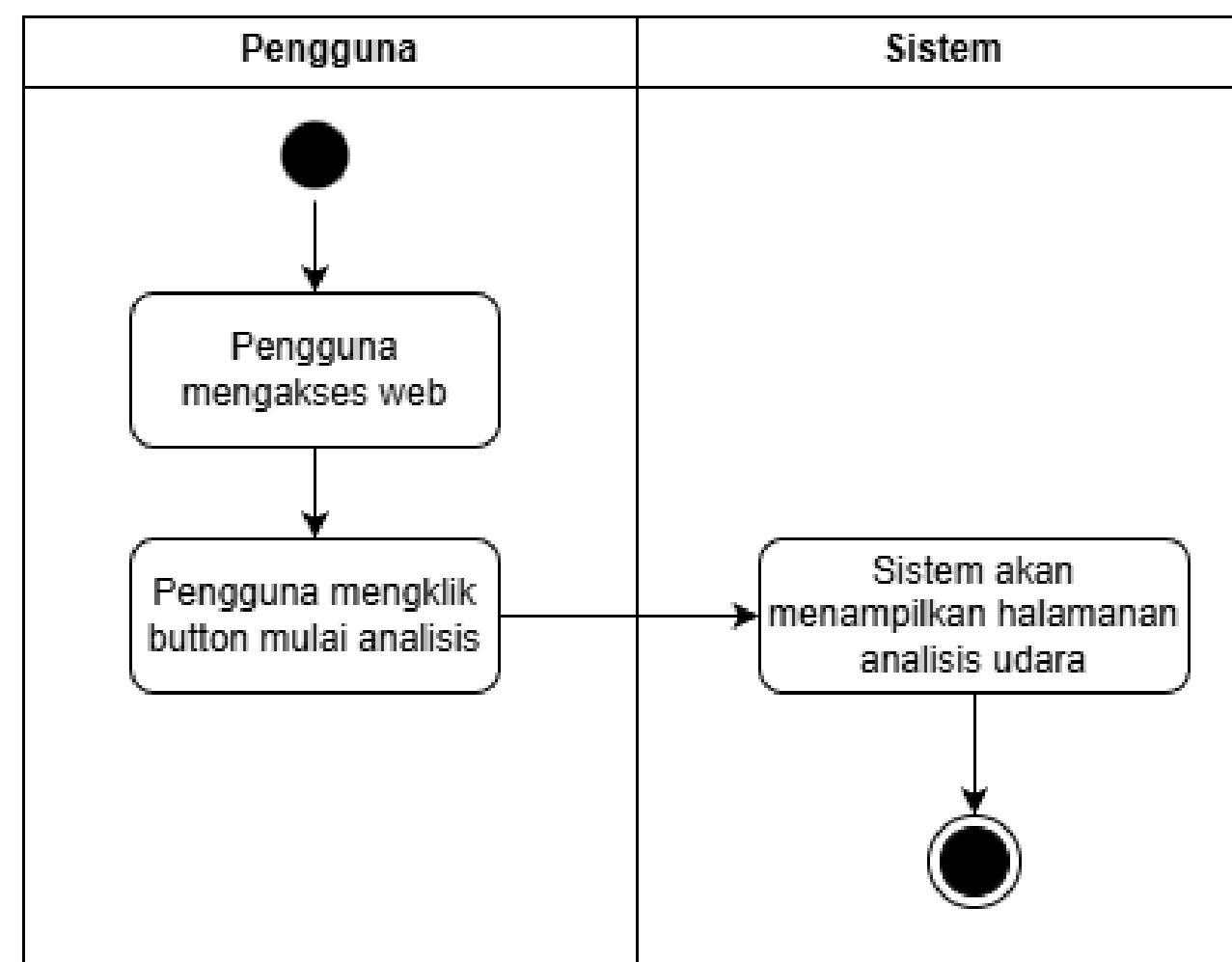


Gambar 3.7 Use Case Diagram

Pada Gambar 3.7 pengguna ketika menggunakan sistem akan diarahkan untuk menjawab beberapa pertanyaan yang diberikan oleh sistem. Kemudian sistem akan memproses dan akan menghasilkan jawaban berupa hasil kualitas udara dan saran.

c. *Activity Diagram*

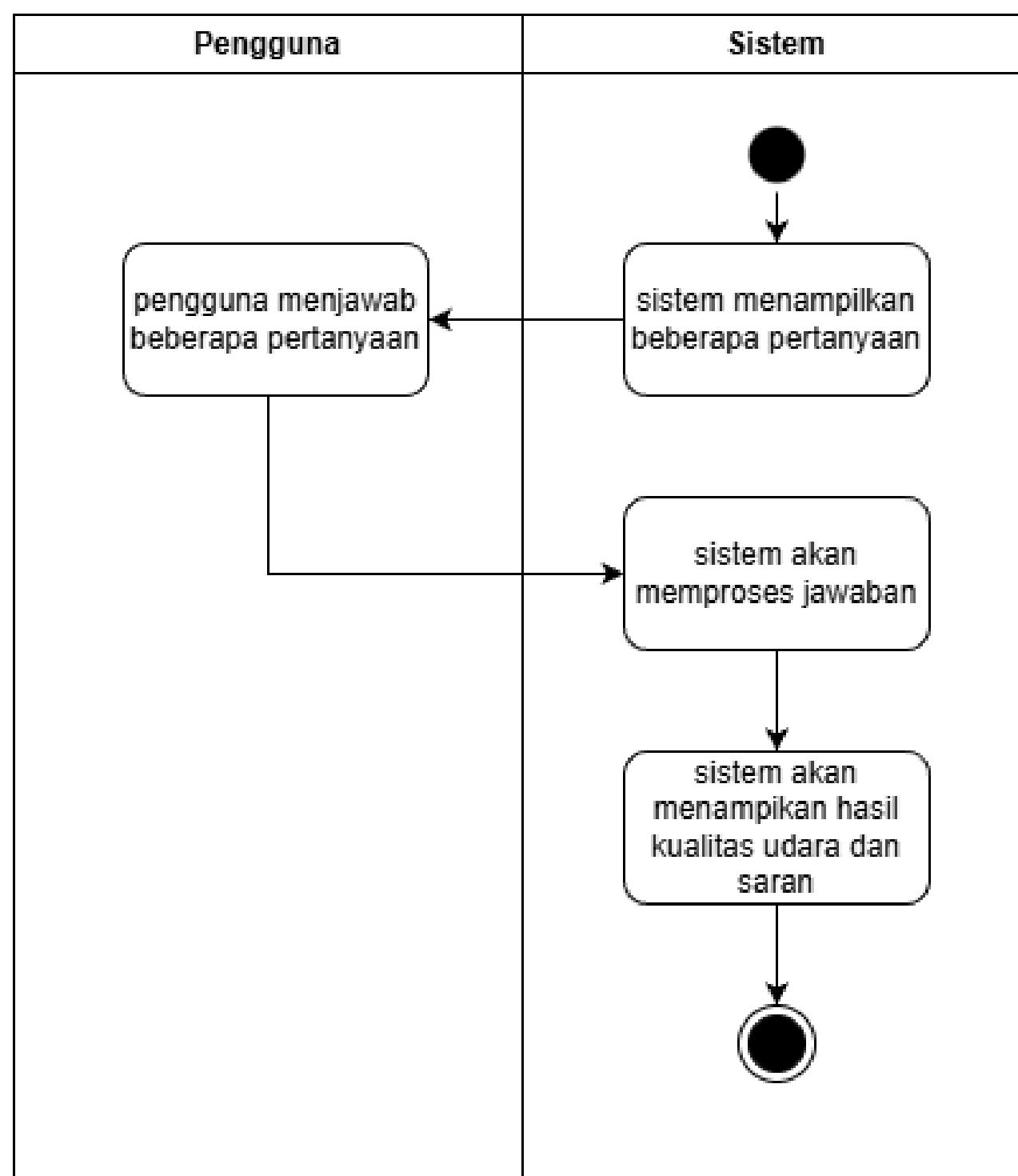
1. Activity Halaman Beranda



Gambar 3. 8 Activity Beranda

Activity ini dimulai Ketika pengguna mengakses website kemudian pengguna mengklik button mulai analisis dan sistem akan langsung berpindah dan menampilkan halaman analisis udara.

2. Activity Halaman Analisis Udara

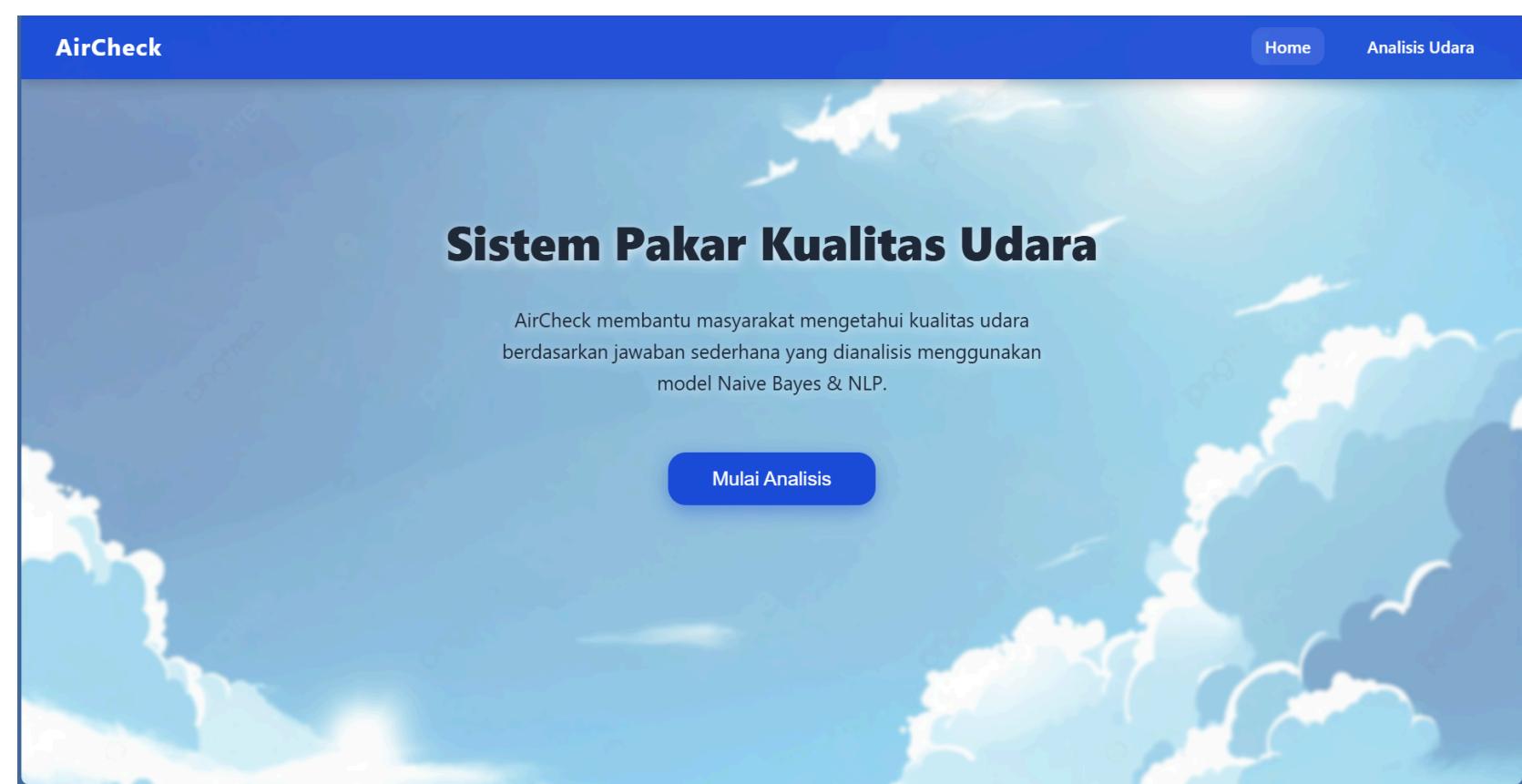


Gambar 3. 9 Activity Analisis Udara

Activity ini dimulai ketika sistem mulai menampilkan beberapa pertanyaan terkait kualitas udara yang berada disekitar kalian, kemudian pengguna akan menjawab pertanyaan-pertanyaan tersebut. Setelah itu, sistem akan memproses jawaban yang kemudian akan menampilkan hasil kualitas udara beserta saran yang ada.

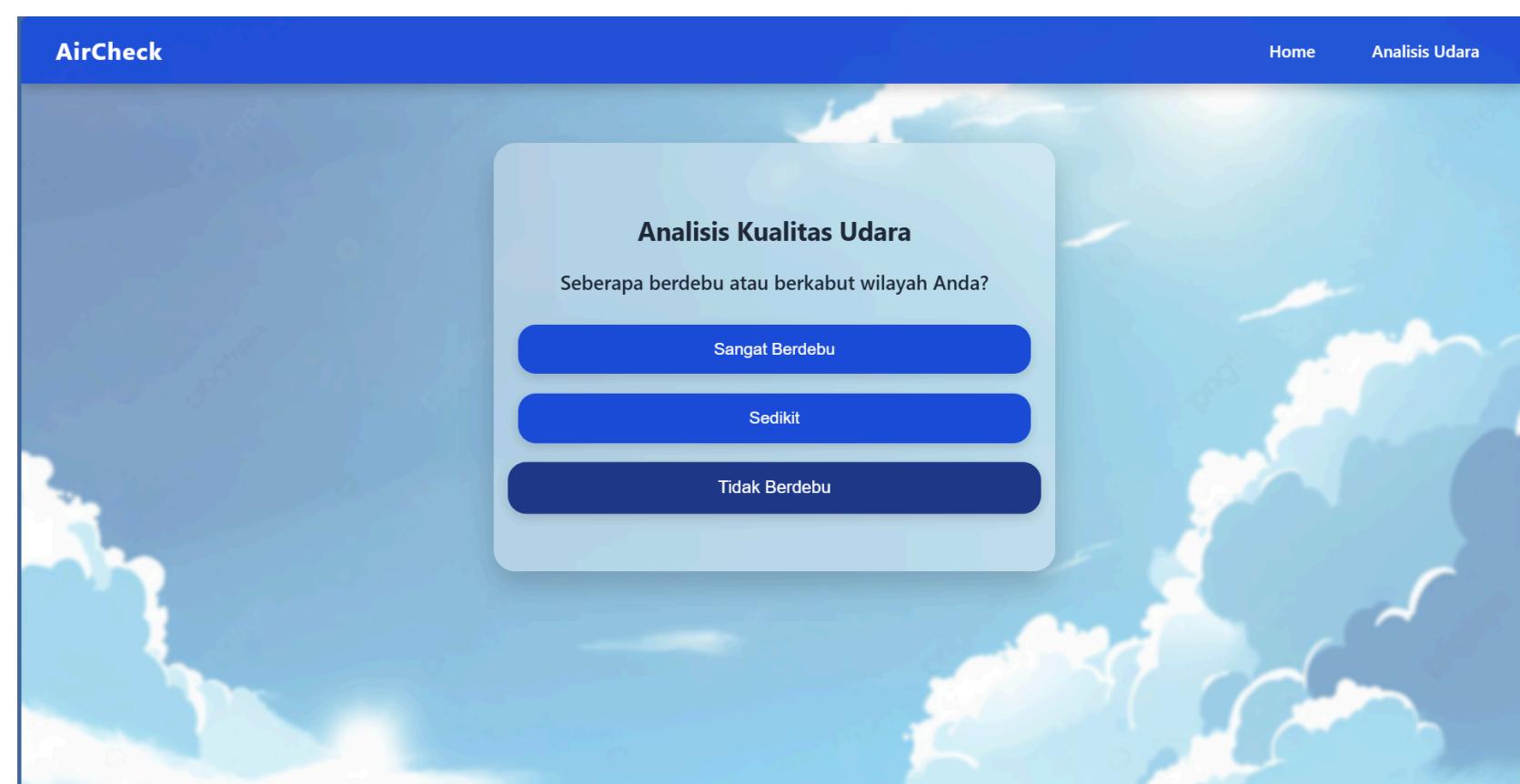
d. User Interface

1. Tampilan Halaman Beranda



Halaman ini merupakan halaman utama ketika pengguna mengakses, yang dimana terdapat dua menu utama yaitu halaman home dan halaman analisis udara.

2. Tampilan Halaman Analisis Udara



Halaman ini merupakan halaman analisis udara yang dimana sistem akan menampilkan beberapa pertanyaan untuk menilai kualitas udara di sekitar anda dan menampilkan hasil dan saran.

3) Coding

Pada tahap ini dilakukan proses implementasi dari sistem yang telah dirancang pada fase sebelumnya, di mana penulis menggunakan bahasa pemrograman *Python* dengan *framework Flask* sebagai *backend* untuk mengintegrasikan model *Naive Bayes* yang telah dilatih sebagai prediktor utama dalam menentukan probabilitas kualitas udara. Sistem juga dilengkapi dengan *Natural Language Processing* (NLP) untuk mengolah jawaban teks dari pengguna melalui tahapan seperti pembersihan data, tokenisasi, dan transformasi teks menjadi representasi numerik yang dapat diproses oleh model. Setelah melalui pipeline NLP, data yang telah terstruktur kemudian dimasukkan ke dalam model *Naive Bayes* untuk menghasilkan prediksi kualitas udara beserta rincian penilaianya. Hasil prediksi tersebut selanjutnya ditampilkan kepada pengguna dalam bentuk informasi yang mudah dipahami, termasuk rekomendasi atau saran sesuai kategori kualitas udara yang diperoleh, sehingga keseluruhan proses berjalan otomatis, responsif, dan user-friendly.

4) Testing

Pada tahap *testing*, pengujian difokuskan pada evaluasi sistem web menggunakan metode *black-box testing*, yaitu pengujian yang menilai fungsi dan keluaran sistem tanpa melihat ke dalam struktur kode program. Pada tahap ini, seluruh fitur aplikasi yang dibangun dengan Flask diuji berdasarkan input dan output untuk memastikan bahwa sistem berfungsi sesuai dengan kebutuhan pengguna. Pengujian dilakukan dengan cara memberikan berbagai macam jawaban dan skenario pada form pertanyaan, mulai dari jawaban normal hingga input yang tidak terduga, untuk melihat apakah sistem dapat menghasilkan prediksi kualitas udara yang benar, menampilkan informasi dengan baik, serta menangani kesalahan input secara tepat. Setiap komponen mulai dari alur pertanyaan, proses pengiriman jawaban, pemanggilan model Naive Bayes, hingga tampilan hasil prediksi diperiksa apakah berjalan konsisten dan tidak menghasilkan error. Melalui pendekatan *black-box* ini, penulis dapat memastikan bahwa sistem web bekerja stabil, responsif, dan

memenuhi standar fungsionalitas tanpa adanya gangguan pada pengalaman pengguna.

DAFTAR PUSTAKA

- Amalia, A., Zaidiah, A., & Isnainiyah, I. N. (2023). *Prediksi kualitas udara menggunakan algoritma k-nearest neighbor*. 07, 496– 507. <https://doi.org/10.29100/jipi.v7i2.2843>
- Amien, M. (2022). *Sejarah Dan Perkembangan Teknik Natural Language Processing (Nlp) Bahasa Indonesia: Tinjauan Tentang Sejarah, Perkembangan Teknologi, Dan Aplikasi Nlp Dalam Bahasa Indonesia*. 2007, 1– 7.
- Ardianto, D., & Widiyatmoko, A. T. (2024). Color Detector in an Image using Python and Computer Vision Library. *Journal of Intelligent Systems and Information Technology*, 1(1), 25– 30. <https://doi.org/10.61971/jisit.v1i1.27>
- Arnanda Feza Raffa, A. (2024). *Deteksi Pergerakan IHSG Berdasarkan Berita Daring Menggunakan Model Deep Learning Berbasis Transformer*. 2022, 439– 448. <https://doi.org/http://dx.doi.org/10.34123/semnasoffstat.v2024i1.1951>
- Dwi, S., Priantoro, B., Rohman, M. G., & Zamroni, M. R. (2025). Klasifikasi kualitas udara dengan metode Naive Bayes berbasis web. *RABIT : Jurnal Teknologi Dan Sistem Informasi Univrab*, 10(2), 1024– 1035.
- Hacohen-kerner, Y., Manor, N., Goldmeier, M., & Bachar, E. (2022). Detection of Anorexic Girls-In Blog Posts Written in Hebrew Using a Combined Heuristic AI and NLP Method. *IEEE Access*, 10, 34800– 34814.

<https://doi.org/10.1109/ ACCESS.2022.3162685>

Harahap, H. syarkowi. (2024). Implementasi Phyton Dalam Matematika. *Mathematical and Data Analytics*, 1(1), 31– 37. <https://doi.org/10.47709/mda.v1i1.3887>

Ichsanudin, M. N., & Yusuf, M. (2022). *Perpustakaan Dengan Metode Black Box Testing Bagi Pemula*. 1(2), 1– 8.

Jaziri, W., & Sassi, N. (2025). Enhancing Semantic Inference in Healthcare Systems

- With Ontology-Enriched NLP. *IEEE Access*, 13(September), 194614–194625. <https://doi.org/10.1109/ACCESS.2025.3632350>
- Khumaidi, A., Raafi, R., & Solihin, I. P. (2020). *Pengujian Algoritma Long Short Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung*. 15(1), 13– 18.
- Kristiana, L., & Miyanto, D. (2023). *Penambahan Parameter PM_{2.5} dalam Prediksi Kualitas Udara : Long Short Term Memory*. 8(2), 188– 202. <https://doi.org/10.26760/mindjournal.v8i2.188-202>
- Marfalino, H., Novita, T., & Djesmedi, D. (2022). *Sistem Pakar Diagnosa Penyakit Saluran Pencernaan Pada Manusia Dengan Metode Cased Based Reasoning*. 1(2), 83– 86.
- Nugroho, G. D., Santoso, I. B., Aji, I. P., & Franskusuma, S. (2025). *SOCIETY : Jurnal Pengabdian Masyarakat , Educating on the Application of Tensorflow in Artificial Intelligence , Machine Learning and Deep Learning* *SOCIETY : Jurnal Pengabdian Masyarakat* , 4(2), 301– 317.
- Ramdany, S. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, 5(1). <https://doi.org/10.31599/2e9afp31>
- Rivaldi, R. C., & Wismarini, T. D. (2024). *Analisis Sentimen Pada Ulasan Produk Dengan Metode Natural Language Processing (NLP) (Studi Kasus Zalika Store 88 Shopee)*. 17(1), 120– 128.
- Rizki, M., Rahardika, F., Putra, B., & Jupriadi, L. (2025). Comparison of Accuracy Level of Certainty Factor Method and Bayes Theorem on Cattle Disease.

ILKOM Jurnal Ilmiah Vol., 16(3), 343– 355.

Serarawani, I., Soekarta, R., & Amri, I. (2023). Rancang Bangun Sistem Informasi
Administrasi Kependudukan Di Kelurahan Klawasi Berbasis Android.
FRAMEWORK, 02(01), 70– 78.
<https://doi.org/doi.org/10.33506/jiki.v2i01.3077>

- Soekarta, R., Ermin, E., Faroek, D. A., Tella, F., & Firmansyah, A. (2024). Penerapan Metode Extreme Programming Pengaduan Penyelundupan Satwa yang Dilindungi. *Techno.Com*, 23(2), 480– 489. <https://doi.org/10.62411/tc.v23i2.10063>
- Suharyanto, E. (2022). *Analisis Data Minat Calon Mahasiswa Universitas Pamulang Dengan Menggunakan Algoritma Naive*. 70– 76.
- Suri, M. I., & Puspaningrum, A. S. (2020). Sistem Informasi Manajemen Berita Berbasis Web. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 8– 14. <https://doi.org/10.33365/jtsi.v1i1.128>
- Tella, F., Setyawan, M. R., Soekarta, R., Sari, N., Teknik, F., & Sorong, U. M. (2024). *Rancang bangun sistem informasi data laporan kepolisian unit pencurian motor berbasis android*. 7, 44– 52. <https://doi.org/10.37600/tekinkom.v7i1.1000>
- Valentino Adhy Nuantra, Rifki Sacky, M., Wahyu Kristianto, M Reza Fadillah, Mila Jamilatul Mahmudah, Sultan Alvian Chrisanda Hanif, & Ferida Yuamita. (2022). Faktor Usability Testing Terhadap Penggunaan Presensi Di Web SIA UTY. *Jurnal Teknologi Dan Manajemen Industri Terapan*, 1(3), 173– 182. <https://doi.org/10.55826/tmit.v1iiii.36>
- Widiawati, F., Kurniawan, R., & Suprapti, T. (2023). *Klasifikasi Data Tingkat Kualitas Udara Di Tangerang Selatan*. 7(6), 3739– 3745.
- Yusuf, M., Faroek, D. A., Pattanang, M. S. S., Salam, R., Informatika, P. T., Teknik, F., Sorong, U. M., Kota, M., Papua, S., & Daya, B. (2024). Sistem Pakar Diagnosa Penyakit Pada Tanaman Semangka Merah Menggunakan

Metode CNN Berbasis Android. *JTS : Jurnal Teknik*, 13(1), 99 – 109.