Air Quality Prediction
Laporan Project Machine Learning

Kelompok 3
Kenneth Angelo Sulaiman (2702373715)
David Arifin (2702331151)
Colin Wilson(2702257406)

1. **Latar Belakang dan Masalah**

   Kualitas udara telah menjadi salah satu isu lingkungan dan kesehatan masyarakat yang paling mendesak di era modern, dengan dampak yang terasa luas di berbagai belahan dunia. Laporan dari berbagai institusi, termasuk Organisasi Kesehatan Dunia (WHO), secara konsisten menyoroti bagaimana paparan polusi udara berkontribusi pada peningkatan angka penyakit pernapasan, kardiovaskular, dan berbagai masalah kesehatan kronis lainnya, yang pada gilirannya membebani sistem kesehatan dan mengurangi kualitas hidup masyarakat.

   Di Indonesia, khususnya di kota-kota besar seperti Jakarta, permasalahan kualitas udara seringkali mencapai tingkat yang mengkhawatirkan. Fenomena ini bukan lagi sekadar data statistik, melainkan menjadi realitas harian yang mudah diamati. Sebagai contoh, pengalaman pribadi di area padat penduduk seperti di sekitar tempat studi, seringkali terasa adanya udara yang pengap atau bahkan kabut asap, terutama ketika melihat kendaraan bermotor yang terus-menerus memadat di jalanan. Emisi gas buang dari jutaan kendaraan ini, ditambah dengan kontribusi dari aktivitas industri, pembangkit listrik, serta faktor-faktor lain seperti kepadatan populasi, merupakan sumber utama polutan seperti Particulate Matter (PM10 dan PM2.5), Nitrogen Dioksida (NO_2), Sulfur Dioksida (SO_2), dan Karbon Monoksida (CO).

   Dampak dari polusi udara yang buruk ini sangat dirasakan oleh masyarakat umum. Kesulitan bernapas, iritasi mata, dan batuk-batuk menjadi keluhan umum. Lebih jauh lagi, paparan jangka panjang dapat mengakibatkan masalah kesehatan yang lebih serius, bahkan mengurangi harapan hidup. Dalam kehidupan sehari-hari, kualitas udara yang tidak menentu menyulitkan masyarakat umum untuk membuat keputusan proaktif terkait aktivitas di luar ruangan. Apakah aman untuk berolahraga? Haruskah anak-anak bermain di taman? Perlukah saya memakai masker hari ini? Pertanyaan-pertanyaan ini seringkali sulit dijawab dengan informasi yang tersedia.

   Meskipun saat ini sudah ada aplikasi atau fitur di ponsel pintar yang dapat memberikan informasi Indeks Kualitas Udara (IKU) secara real-time, program-program tersebut umumnya hanya menyajikan angka atau kategori kualitas udara (misalnya, "Baik," "Sedang," "Tidak Sehat"). Kekurangan krusial dari solusi yang ada adalah tidak adanya rekomendasi spesifik dan personal mengenai tindakan yang harus diambil oleh pengguna berdasarkan kondisi udara yang diprediksi. Masyarakat tidak hanya butuh tahu seberapa buruk udaranya, tetapi juga apa yang harus mereka lakukan untuk melindungi diri.

Dalam konteks inilah proyek ini dikembangkan sebagai bagian dari tugas perkuliahan, dengan tujuan untuk mengisi kesenjangan informasi tersebut. Proyek ini tidak hanya akan memprediksi Indeks Kualitas Udara berdasarkan berbagai parameter penting seperti lokasi (kedekatan dengan area industri), suhu, kelembaban, kepadatan populasi, serta tingkat polutan PM10, PM2.5, NO_2, SO_2, dan CO (data diperoleh dari platform Kaggle), namun yang terpenting, ia akan menyediakan rekomendasi tindakan yang jelas dan actionable.

Masalah utama yang ingin diselesaikan oleh program ini adalah memberikan informasi kualitas udara yang komprehensif dan panduan tindakan yang relevan secara langsung kepada pengguna biasa. Program ini bertujuan untuk meningkatkan kesadaran publik dan memberdayakan individu agar dapat membuat keputusan yang lebih cerdas dan bertanggung jawab terkait kesehatan mereka di tengah tantangan polusi udara perkotaan. Dengan demikian, program ini diharapkan dapat menjadi alat praktis yang membantu masyarakat luas dalam menjaga kesehatan mereka dengan lebih baik.

2. Pendekatan dan Metodologi

2.1. Dataset

Dalam proyek ini, kami menggunakan dataset yang diperoleh dari platform Kaggle dengan judul "*Air Quality and Pollution Assessment*". Dataset ini berjumlah 5000 data yang mencakup faktor demografi dan lingkungan yang krusial dalam mempengaruhi tingkat polusi. Author dataset ini tidak memberi tahu lokasi di mana dataset ini dikumpulkan.

Fitur-fitur utama yang tersedia dalam dataset ini antara lain::

- Temperature (°C): Suhu udara dalam derajat Celcius.
- Humidity (%): Kelembaban relatif udara.
- PM2.5 (μg/m³): Konsentrasi partikulat berukuran ≤ 2.5 mikron.
- PM10 (μg/m³): Konsentrasi partikulat berukuran ≤ 10 mikron.
- NO_2 (ppb): Salah satu gas polutan dari kendaraan bermotor.
- SO_2 (ppb): Gas beracun yang dihasilkan dari pembakaran bahan bakar fosil.
- CO (ppm): Gas beracun yang dihasilkan dari pembakaran tidak sempurna bahan yang mengandung karbon dan berbahaya bila terhirup dalam jumlah besar.
- Proximity to Industrial Areas (km): Jarak lokasi terhadap zona industri terdekat
- Population Density (people/km^2): Jumlah kepadatan orang berdasarkan kilometer persegi di lokasi

Target variable dataset ini adalah untuk menentukan level kualitas udara yang dibagi menjadi 4 tipe, yaitu:

1. Good: Udara bersih dengan level polusi rendah
2. Moderate: Kualitas udara bisa diterima (sedang) namun ada polutan
3. Poor: Polusi udara cukup terasa dan mampu mengakibatkan gangguan kesehatan untuk individu yang sensitif
4. Hazardous: Udara sangat tercemar dan menimbulkan risiko kesehatan yang serius terhadap populasi

Kualitas udara sangat bergantung pada kombinasi parameter-parameter ini, dan tujuan proyek ini adalah memprediksi indeks kualitas udara (AQI) berdasarkan nilai-nilai parameter di atas.

2.2. Set Up Dataset

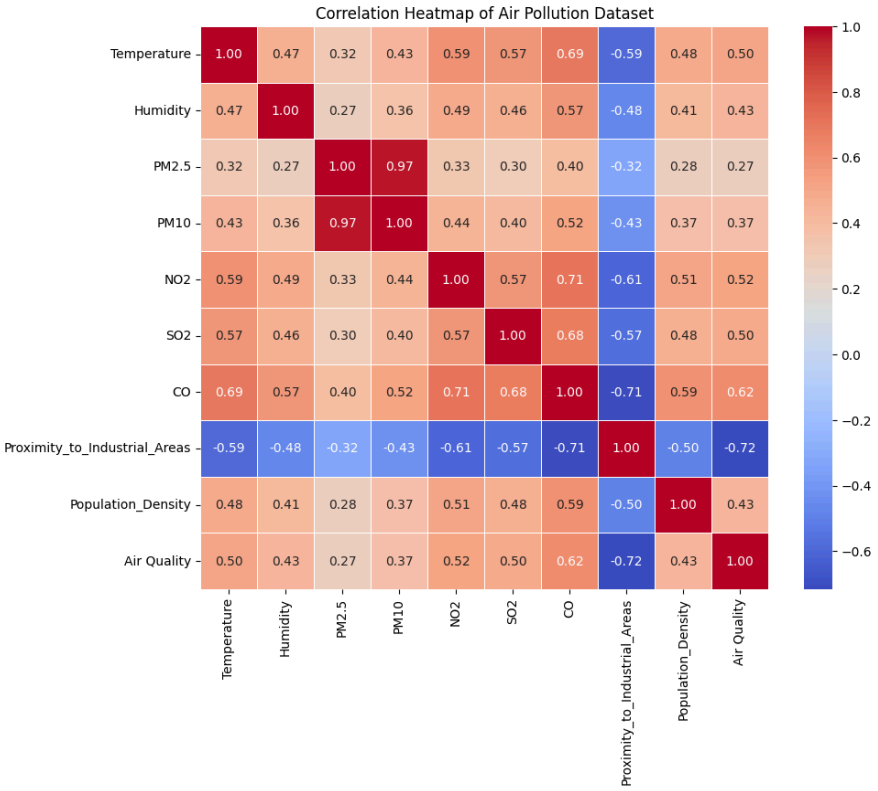
Pertama-tama, kami melakukan read terhadap data terlebih dahulu dan datanya berjumlah 5000 baris x 10 kolom. Kami melihat info data tersebut terlebih dahulu untuk mengetahui apakah data mengandung NaN maupun NULL yang perlu ditangani, tapi ternyata dataset tersebut tidak memiliki NaN and NULL untuk setiap baris yang ada. Selain itu, kami menemukan ada kolom “Air Quality” dengan tipe data “object” sehingga perlu diubah menjadi tipe data “integer”. Kami mengubah tipe data dari “object” ke “integer” dengan menggunakan label encoder dengan mapping

- Good sebagai 0
- Hazardous sebagai 1
- Moderate sebagai 2
- Poor sebagai 3

Selain itu, kami melakukan pengecekan terhadap keseimbangan data dan dataset tersebut sebenarnya imbalance, tapi mempertimbangkan data yang cukup banyak, kami tidak melakukan perubahan terhadap dataset tersebut. Distribusi setiap kelasnya adalah sebagai berikut:

- Good: 2000 rows
- Moderate: 1500 rows
- Poor: 1000 rows
- Hazardous: 500 rows

Berikut merupakan matriks korelasi antar setiap fitur:



Dapat dilihat, fitur PM2.5 dan PM10 cukup tinggi sehingga dapat pilih salah satu saja, tapi kami tidak melakukan hal tersebut karena mempertimbangkan keduanya merupakan indikator yang penting.

Kami menggunakan pembagian train/test dengan ratio 80/20. Dengan semua indikator seperti Temperature, Humidity, PM2.5, PM10, NO2, SO2, CO, Proximity_to_Industrial_Areas, Population_Density sebagai fiturnya, dan Air Quality sebagai target variabelnya.

2.3. Grid Search

Setelah data siap, proses selanjutnya adalah melatih model *machine learning* dan mengoptimalkan performanya melalui penyetelan *hyperparameter*. *Hyperparameter* adalah parameter konfigurasi eksternal model yang tidak dipelajari dari data, melainkan ditetapkan sebelum proses pelatihan. Penyetelan *hyperparameter* yang tepat sangat krusial untuk memastikan model tidak *underfit* atau *overfit*.

Kami menggunakan Grid Search (GridSearchCV), sebuah metode *exhaustive search* yang mencoba semua kombinasi *hyperparameter* yang ditentukan. Untuk setiap kombinasi, model dilatih dan dievaluasi menggunakan 5-fold *cross-validation*. Tujuannya adalah menemukan kombinasi *hyperparameter* yang memberikan skor performa (akurasi) terbaik pada data validasi. Grid Search diterapkan secara terpisah untuk Random Forest Classifier dan XGBoost Classifier.

- Grid Search untuk Random Forest Classifier

Untuk model Random Forest, hyperparameter yang disetel meliputi:

- n_estimators: Jumlah pohon dalam forest. Rentang nilai yang diuji: [100, 200, 300, 400, 500].

- max_depth: Kedalaman maksimum setiap pohon. Rentang nilai yang diuji: [1, 2, ..., 10].
- min_samples_split: Jumlah sampel minimum yang diperlukan untuk membagi sebuah node. Rentang nilai yang diuji: [2, ..., 10]. (Perlu dicatat bahwa nilai 1 untuk min_samples_split tidak valid dan telah diabaikan secara otomatis oleh GridSearchCV).

Metrik evaluasi yang digunakan adalah akurasi (scoring='accuracy'), dan seluruh *core* CPU digunakan (n_jobs=-1) untuk mempercepat proses.

- Grid Search untuk XGBoost Classifier
Untuk model XGBoost, *hyperparameter* yang disetel meliputi:
- n_estimators: Jumlah *boosting rounds* atau pohon yang dibangun. Rentang nilai yang diuji: [100, 200, ..., 1000].
- max_depth: Kedalaman maksimum setiap pohon. Rentang nilai yang diuji: [1, 2, ..., 10].
- learning_rate: Ukuran langkah *shrinkage* yang digunakan dalam pembaruan untuk mencegah *overfitting*. Rentang nilai yang diuji: [0.01, 0.02, ..., 0.1].

Parameter use_label_encoder=False dan eval_metric='mlogloss' disetel untuk kompatibilitas dengan versi XGBoost terbaru dan penggunaan metrik yang tepat untuk klasifikasi multiclass.

2.4. Random Forest
Setelah menentukan *hyperparameter* terbaik melalui Grid Search, model Random Forest Classifier kemudian dilatih menggunakan parameter optimal tersebut**.** Random Forest adalah algoritma *ensemble* yang membangun banyak *decision tree* saat pelatihan dan mengeluarkan kelas yang merupakan modus dari kelas-kelas yang dihasilkan oleh decision tree individu (untuk klasifikasi). Kekuatan Random Forest terletak pada kemampuannya mengurangi *overfitting* sambil mempertahankan akurasi yang tinggi. Model dilatih pada data pelatihan (X_train, y_train) dan kemudian dievaluasi pada data pengujian (X_test, y_test).

2.5. XG Boost
Model XGBoost Classifier dilatih menggunakan *hyperparameter* terbaik yang ditemukan dari Grid Search. XGBoost adalah implementasi gradient boosting yang sangat efisien dan populer, dikenal karena kecepatan dan performanya yang tinggi dalam berbagai Machine Learning. Model dilatih pada data pelatihan (X_train, y_train) dengan evaluasi pada data pengujian (X_test, y_test) untuk memantau *loss* selama pelatihan.

2.6. Ensemble Model (Voting Classifier)
Untuk lebih meningkatkan robusta dan performa prediksi, kami menggabungkan kekuatan model Random Forest dan XGBoost ke dalam sebuah Model Ensemble menggunakan Voting Classifier. Pendekatan *ensemble* bertujuan untuk mendapatkan performa yang lebih baik daripada model individual dengan menggabungkan prediksi dari beberapa model. Dalam kasus ini, kami menggunakan 'soft' voting, di mana probabilitas kelas dari setiap model dijumlahkan (atau dirata-rata), dan kelas dengan

probabilitas gabungan tertinggi akan dipilih sebagai prediksi akhir. Ini memungkinkan model untuk mempertimbangkan tingkat kepercayaan masing-masing model terhadap prediksinya.

- Voting Classifier Tanpa Pemberian Bobot
Voting Classifier dilatih tanpa bobot eksplisit, yang berarti setiap model (Random Forest dan XGBoost) memiliki bobot yang sama dalam proses voting
.
- Voting Classifier dengan Pemberian Bobot (Weighted Voting)
Voting Classifier di sini menggunakan bobot (weights) yang berbeda untuk Random Forest dan XGBoost dalam Voting Classifier. Pemberian bobot ini memungkinkan alokasi prioritas yang lebih tinggi kepada model yang dianggap lebih akurat atau lebih relevan.

3. Implementasi

3.1. Teknologi yang digunakan

Pengembangan proyek ini memanfaatkan Python sebagai bahasa pemrograman utama, didukung oleh pustaka inti data science dan machine learning. Teknologi kunci yang digunakan meliputi:

- Pandas & Numpy: Digunakan untuk pra-pemrosesan, manipulasi, dan komputasi data.
- Scikit-learn: Menyediakan algoritma model (*RandomForestClassifier, VotingClassifier*), fungsionalitas *GridSearchCV* untuk penyetelan hyperparameter, serta metrik evaluasi model.
- XGBoost: Diimplementasikan sebagai salah satu model *ensemble* untuk performa prediksi yang tinggi.
- Matplotlib & Seaborn: Digunakan untuk visualisasi data, seperti heatmap korelasi, yang mendukung analisis awal dataset.
- joblib: Berfungsi untuk serialisasi dan penyimpanan model (*air_quality_model.pkl*) dan *LabelEncoder* (label_encoder.pkl), memfasilitasi integrasi model yang sudah dilatih ke dalam aplikasi.
- Streamlit: Library Python yang dipilih untuk membangun interface aplikasi web yang interaktif secara cepat, memungkinkan interaksi pengguna langsung dengan model.

3.2. Integrasi Model dan Aplikasi Web (Streamlit)

Model *Voting Classifier* dengan bobot (2,1), yang diidentifikasi sebagai model terbaik, beserta *LabelEncoder* yang sesuai, disimpan dalam format .pkl menggunakan *joblib*. Proses ini memungkinkan model dimuat langsung ke dalam aplikasi web tanpa memerlukan pelatihan ulang.

Aplikasi web dikembangkan menggunakan *Streamlit* (berkas app.py), yang dirancang untuk menyediakan user interface yang intuitif dan fungsional:

- Input Data Interaktif: Pengguna memasukkan parameter lingkungan dan polutan (suhu, kelembaban, PM2.5, PM10, NO_2, SO_2, CO, kedekatan industri, kepadatan populasi) melalui komponen *st.number_inpu*t dan *st.slider*.

- Kategorisasi Instan Input: Aplikasi menyertakan fungsi yang memberikan umpan balik visual untuk setiap nilai parameter yang dimasukkan, membantu pengguna memahami level saat itu.
- Proses Prediksi dan Rekomendasi: Setelah semua input dimasukkan, tombol "Prediksi" akan memicu model untuk menghasilkan kategori kualitas udara. Hasil prediksi (misalnya, "Good", "Moderate", "Poor", "Hazardous") ditampilkan bersama rekomendasi tindakan spesifik dan *actionable* yang relevan dengan level kualitas udara tersebut.
- Alur Kerja Aplikasi: Data input dari pengguna akan diproses oleh model *air_quality_model.pkl*. Prediksi numerik dari model kemudian dikonversi kembali ke label kategorikal yang mudah dipahami menggunakan *label_encoder.pkl*. Hasil prediksi beserta rekomendasi yang relevan kemudian disajikan secara visual kepada pengguna.

4. Evaluasi & Hasil

Berikut hasil evaluasi model *machine learning* yang telah dikembangkan, mencakup metrik performa utama dan pembahasan kendala yang dihadapi selama proses pengembangan

4.1. Grid Search

Hasil Grid Search untuk Random Forest:
Best Parameters: {'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 400}
Best Score (‘accuracy’): 0.9550000000000001

Berdasarkan hasil ini, kombinasi *hyperparameter* terbaik untuk Random Forest yang memberikan akurasi tertinggi pada data pelatihan selama *cross-validation* adalah n_estimators=400, max_depth=10, dan min_samples_split=5. Akurasi terbaik yang dicapai adalah sekitar **95.5%**.

Hasil Grid Search untuk XGBoost:
Best parameters: {'learning_rate': 0.08, 'max_depth': 1, 'n_estimators': 700}
Best Score (‘accuracy’): 0.9564999999999999

Dari hasil ini, kombinasi *hyperparameter* terbaik untuk XGBoost adalah n_estimators=700, max_depth=1, dan learning_rate=0.08. Akurasi terbaik yang dicapai selama *cross-validation* adalah sekitar **95.65%**, sedikit lebih tinggi dari Random Forest. Perlu dicatat bahwa meskipun Grid Search merekomendasikan max_depth=1, penggunaan max_depth=5 pada model final mengindikasikan penyesuaian manual yang mungkin bertujuan untuk kompleksitas model yang lebih optimal.

4.2. Random Forest Classifier

Berikut Klasifikasi Random Forest pada Data Pengujian:

Random Forest Classification Report:

	precision	recall	f1-score	support
Good	1.00	1.00	1.00	409
Hazardous	0.92	0.86	0.89	111
Moderate	0.97	0.97	0.97	294
Poor	0.88	0.90	0.89	186
accuracy			0.96	1000
macro avg	0.94	0.93	0.94	1000
weighted avg	0.96	0.96	0.96	1000

Analisis Hasil Random Forest:
Model Random Forest menunjukkan akurasi keseluruhan sebesar **0.96 (96%)** pada data pengujian. Performa model sangat baik untuk kelas 'Good' dan 'Moderate', dengan *precision*, *recall*, dan *f1-score* masing-masing 1.00 dan 0.97. Untuk kelas 'Hazardous' dan 'Poor', performa sedikit lebih rendah tetapi masih kuat, dengan *f1-score* sekitar 0.89. Ini mengindikasikan kemampuan model yang solid dalam memprediksi mayoritas kelas, meskipun terdapat sedikit tantangan pada kelas dengan jumlah *support* yang lebih kecil.

4.3. XGBoost
Berikut Klasifikasi XGBoost pada Data Pengujian:

XGBoost Classification Report:

	precision	recall	f1-score	support
Good	1.00	1.00	1.00	409
Hazardous	0.94	0.91	0.92	111
Moderate	0.97	0.96	0.97	294
Poor	0.90	0.92	0.91	186
accuracy			0.96	1000
macro avg	0.95	0.95	0.95	1000
weighted avg	0.96	0.96	0.96	1000

Analisis Hasil XGBoost:
Model XGBoost mencapai akurasi keseluruhan sebesar **0.96 (96%)** pada data pengujian. Akurasi ini setara dengan performa model Random Forest. Performa model sangat baik untuk kelas 'Good' dan 'Moderate', dengan *f1-score* masing-masing 1.00 dan 0.97. Untuk kelas 'Hazardous' dan 'Poor', performa sedikit

lebih rendah tetapi lebih bagus dibanding Random Forest, dengan *f1-score* sekitar 0.92 dan 0.91 yang menandakan bahwa model lebih bagus untuk menangani data yang imbalance dibanding Random Forest.

4.4. Ensemble Model

Berikut Klasifikasi Voting Classifier (tanpa bobot) pada Data Pengujian:

Voting Classifier Classification Report:

	precision	recall	f1-score	support
Good	1.00	1.00	1.00	409
Hazardous	0.94	0.91	0.93	111
Moderate	0.97	0.97	0.97	294
Poor	0.90	0.92	0.91	186
accuracy			0.96	1000
macro avg	0.95	0.95	0.95	1000
weighted avg	0.97	0.96	0.97	1000

Analisis:

Akurasi keseluruhan dari Voting Classifier tanpa bobot adalah **0.96 (96%)**. Meskipun akurasi keseluruhan sama dengan model individual, terlihat adanya peningkatan pada metrik f1-score untuk kelas 'Hazardous' (0.93) dibandingkan dengan XGBoost (0.92) serta ada peningkatan metrik weighted average untuk *precision* dan *f1-score* hingga **0.97 (97%)**. Ini menunjukkan bahwa menggabungkan kedua model tersebut memberikan performa yang lebih baik dalam melakukan klasifikasi.

Untuk weighted voting classifier, kami melakukan percobaan dengan jumlah bobot (2,1) (1,3) (2.5, 1.5). Hasil menunjukkan bahwa performa pada bobot (2,1) yakni Random Forest lebih dominan mengalami peningkatan untuk weighted average recall hingga 0.97 (97%) sehingga kami memutuskan untuk menggunakan model tersebut.

- Bobot (2,1) (Random Forest lebih dominan)
  Dengan bobot (2,1) (Random Forest diberi bobot 2, XGBoost diberi bobot 1), model Voting Classifier memberikan hasil sebagai berikut:

```
Weights: (2, 1)
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       409
           1       0.94      0.91      0.92       111
           2       0.97      0.97      0.97       294
           3       0.90      0.92      0.91       186

    accuracy                           0.97      1000
   macro avg       0.95      0.95      0.95      1000
weighted avg       0.97      0.97      0.97      1000
```

Analisis:
Akurasi keseluruhan model ensemble meningkat menjadi 0.97 (97%) dengan bobot ini. Peningkatan ini signifikan dibandingkan model individual dan ensemble tanpa bobot. Metrik F1-score untuk kelas '*Hazardous*' (0.92) dan '*Poor*' (0.91) juga menunjukkan robusta prediksi yang lebih baik.

4.5. Uji Coba Model dan Metrik Performa
Setelah melatih dan menyetel *hyperparameter* masing-masing model, performanya dievaluasi pada data pengujian (X_test, y_test) yang belum pernah dilihat oleh model. Evaluasi ini bertujuan untuk mengukur kemampuan generalisasi model terhadap data baru. Metrik utama yang digunakan adalah akurasi, presisi, recall, dan F1-score, yang disajikan dalam bentuk classification report.

Berikut adalah rangkuman performa dari model-model yang diuji: Random Forest Classifier, XGBoost Classifier, serta Voting Classifier:

Model	**Akurasi (Test Set)**	**Presisi (Weighted Avg)**	**Recall (Weighted Avg)**	**F1-Score (Weighted Avg)**
Random Forest Classifier	0.96	0.96	0.96	0.96
XGBoost Classifier	0.96	0.96	0.96	0.96
Voting Classifier (tanpa bobot)	0.96	**0.97**	0.96	**0.97**
Voting Classifier (bobot 2:1)	**0.97**	**0.97**	**0.97**	**0.97**

Analisis Hasil:

- Model Individual (Random Forest dan XGBoost):
  Kedua model individu menunjukkan performa yang sangat kuat dengan akurasi 96% pada data pengujian. Mereka mampu mengklasifikasikan sebagian besar kategori kualitas udara dengan baik. Namun, berdasarkan *classification report* yang lebih rinci untuk Random Forest (yang metrik per kelasnya terlihat), terdapat sedikit penurunan performa pada kelas minoritas seperti 'Hazardous' dan 'Poor'.
- Voting Classifier (Tanpa Bobot):
  Penggabungan kedua model tanpa bobot eksplisit masih mempertahankan akurasi 96%. Namun, terlihat ada sedikit peningkatan pada *weighted average* F1-score menjadi 0.97, menunjukkan bahwa *ensemble* membantu menyeimbangkan bias dari masing-masing model.
- Voting Classifier (Bobot 2:1):
  Penerapan bobot (2,1) pada Voting Classifier (memberikan bobot lebih pada Random Forest) berhasil meningkatkan akurasi keseluruhan menjadi 97%. Ini merupakan peningkatan yang signifikan dan menunjukkan bahwa *ensemble* dengan bobot yang tepat dapat menghasilkan performa superior. Peningkatan ini juga tercermin pada *weighted average* presisi, recall, dan F1-score yang juga mencapai 0.97. Model ini mampu menangkap nuansa data dengan lebih baik, menghasilkan klasifikasi yang lebih akurat dan *robust*.

Kendala yang dihadapi:

- Keterbatasan Dataset: Meskipun dataset yang digunakan cukup besar, keragaman data (misalnya, data dari berbagai musim, kota, atau skenario polusi ekstrem yang berbeda) dapat memengaruhi generalisasi model di dunia nyata. Data yang tersedia mungkin tidak mencakup semua variasi kondisi kualitas udara yang mungkin terjadi.
- Ketidakseimbangan Kelas (Class Imbalance): Variabel target Air Quality memiliki distribusi kelas yang tidak seimbang (misalnya, kelas 'Good' jauh lebih banyak daripada 'Hazardous'). Meskipun model masih berkinerja baik, ketidakseimbangan ini berpotensi mempengaruhi kemampuan model untuk memprediksi kelas minoritas secara akurat jika tidak ditangani dengan teknik khusus (seperti oversampling atau undersampling), yang tidak sepenuhnya dieksplorasi dalam proyek ini.
- Penyetelan Hyperparameter yang Intensif Komputasi: Proses Grid Search untuk mencari hyperparameter optimal cukup memakan waktu dan sumber daya komputasi, terutama dengan rentang nilai yang luas dan cross-validation 5-fold. Meskipun *n_jobs=-1* digunakan, waktu yang dibutuhkan untuk mencari kombinasi terbaik tetap signifikan.
- Interpretasi Model (Kelas Minoritas): Meskipun akurasi keseluruhan tinggi, memahami mengapa model membuat kesalahan prediksi pada kelas-kelas tertentu (terutama yang minoritas seperti 'Hazardous') bisa menjadi tantangan tanpa alat interpretasi model yang lebih mendalam (seperti SHAP atau LIME). Output classification report saja tidak selalu memberikan gambaran lengkap tentang alasan di balik setiap prediksi.

4.6. Kesimpulan

Pengembangan model machine learning untuk prediksi kualitas udara menunjukkan hasil yang akurat. Melalui pra-pemrosesan data yang cermat dan penyetelan hyperparameter optimal menggunakan Grid Search, model Voting Classifier dengan bobot (2,1) teridentifikasi sebagai solusi paling efektif. Model ensemble ini mencapai akurasi 97% pada data pengujian, menunjukkan kemampuan robust dalam mengklasifikasikan berbagai kategori kualitas udara, termasuk kelas minoritas. Model ini kemudian diintegrasikan ke dalam aplikasi web interaktif berbasis Streamlit. Integrasi ini tidak hanya memungkinkan prediksi kualitas udara, tetapi juga menyediakan rekomendasi tindakan yang jelas dan actionable kepada pengguna. Dengan demikian, sistem ini berfungsi sebagai alat praktis yang dapat meningkatkan kesadaran publik dan memberdayakan individu untuk menjaga kesehatan di tengah tantangan polusi udara.

5. **Pembagian Tugas Kelompok**

   Kenneth Angelo Sulaiman: Integrasi model dan streamlit, Website deploy, Laporan
   David Arifin: Model, Laporan
   Colin Wilson: Model, Laporan