LAPORAN TUGAS IF4043

Pengembangan SI (KBS) dengan Data Science

DISUSUN OLEH

Kadek Dwi Bagus Ananta Udayana / 13519057

Muhammad Fahmi Alamsyah / 13519077

Mohammad Yahya Ibrahim / 13519091

INSTITUT TEKNOLOGI BANDUNG

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

TEKNIK INFORMATIKA

2022/2023

Daftar Isi

Deskripsi Permasalahan Bisnis

1. **Deskripsi Masalah**

   Dalam upaya pengendalian pencemaran udara, Kementerian Lingkungan Hidup dan Kehutanan (KLHK) berkomitmen untuk memberikan informasi mutu udara yang tepat dan akurat kepada masyarakat. Menurut peraturan KEPMEN LK 45/1997 penentuan kualitas udara dipengaruhi oleh gas-gas seperti PM_{10}, CO, O_3, SO, dan NO2. Pemerintah DKI Jakarta mengukur kandungan gas-gas tersebut dengan mengoperasikan Stasiun Pemantau Kualitas Udara (SPKU) yang disebar di beberapa daerah strategis di Jakarta. Data informasi kualitas udara yang dihasilkan oleh SPKU tersebut sangatlah penting bagi masyarakat untuk mengetahui udara yang mereka hirup dalam suatu daerah sudah tercemar atau tidak. Oleh karena itu, pengolahan data informasi SPKU ini merupakan permasalahan yang sangat krusial bagi pemerintah agar nantinya upaya pengendalian pencemaran udara dapat menjadi terhambat. Selain itu juga, pengolahan data informasi yang baik akan memberikan kemudahan pemerintah dalam menentukan strategi-strategi selanjutnya untuk mengatasi permasalahan pencemaran udara di Jakarta.

2. **Metrik Pengukuran Keberhasilan**

   Metrik pengukuran keberhasilan SPKU didasarkan pada Indeks Standar Pencemaran Udara yang mengacu pada Permen LHK No.14 Tahun 2020. Kategori penjelasan ISPU adalah sebagai berikut.

Kategori	Keterangan
Baik	Tingkat kualitas udara yang sangat baik dan tidak memberikan efek negatif terhadap manusia, hewan, dan tumbuhan
Sedang	Tingkat kualitas udara masih dapat diterima pada kesehatan manusia, hewan, dan tumbuhan
Tidak Sehat	Tingkat kualitas udara yang bersifat merugikan pada manusia, hewa, dan tumbuhan
Sangat Tidak Sehat	Tingkat kualitas udara yang dapat meningkatkan risiko kesehatan pada sejumlah segmen populasi yang terpapar
Berbahaya	Tingkat kualitas udara yang dapat merugikan kesehatan serius pada populasi dan perlu penanganan cepat

Deskripsi Analitik

1. Deskripsi Analitik

Kebutuhan akan suatu sistem yang mampu menganalisis kualitas tingkat udara suatu daerah sangatlah penting. Analisis yang kuat dan tepat akan mampu membantu pemerintah dalam menentukan kebijakan-kebijakan yang strategis untuk menghindari pencemaran udara terutama di kota Jakarta. Oleh karena itu, perlu dikembangkan sistem model *machine learning* berbasis *Decision Tree Algorithm* untuk memudahkan mengolah data mentah menjadi suatu informasi. Model *machine learning* ini digunakan nantinya bertujuan untuk mencari keterkaitan antara data kandungan udara dengan klasifikasi ISPU yang telah dibuat oleh *expert system*. Dengan demikian, harapannya adalah dengan adanya model *machine learning* ini sistem mampu menganalisis data-data baru dan mengelompokkan data tersebut secara akurat dan cepat sesuai dengan klasifikasi ISPU.

2. Metrik Pengukuran Keberhasilan

Untuk menguji keberhasilan dan performa model *machine learning* berbasis *Decision Tree Learning* yang telah dibangun maka digunakan metrik pengukuran *accuracy* dan *f1-score*. Metrik pengukuran accuracy berfungsi untuk mengukur performa berdasarkan rasio dari prediksi benar dengan keseluruhan data, sedangkan metrik pengukuran *f1-score* berfungsi untuk mengukur performa berdasarkan perbandingan rata-rata (*harmonic mean*) dari metrik *precision* dan *recall*. *Precision* sendiri merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif, sedangkan *recall* adalah rasio prediksi benar positif dibandingkan dengan keseluruhan data yang faktanya benar positif.

Langkah Pengembangan dan Implementasi

Implementasi pemodelan menggunakan bahasa pemrograman python. Terdapat beberapa tahapan dalam pengembangan model yaitu :

1. Pembacaan Data

Data yang akan digunakan dalam pembangunan model machine learning adalah data yang diambil pada situs Open Data mengenai Indeks Standar Pencemaran Udara di SPKU Bulan Januari Tahun 2020.
https://data.jakarta.go.id/dataset/indeks-standar-pencemaran-udara-ispu-tahun-2020/resource/0f168955-5771-43a2-9fed-9c74ac3c268e

```
# Read the dataset
df = pd.read_csv('indeks-standar-pencemar-udara-di-spku-bulan-januari-tahun-2020.csv')
print(df.head())
```

2. Data Preparation

Berdasarkan data yang diperoleh dilakukan pemilahan dan pembersihan data. Data yang digunakan untuk *modelling* akan berfokus pada data seperti jumlah PM_{10}, SO_2, CO, O3, dan NO_2. Namun, sebelum memproses data menjadi model maka data harus dilakukan cleansing terlebih dahulu untuk menghindari data yang null atau pun tidak valid. Data yang null atau pun tidak valid nantinya akan diganti dengan *average* dari atribut tersebut

```
# Cleanse the data : Eliminate null values and replace them with AVG value

# Replace invalid value "---" with NaN
df = df.replace('---',np.nan)
df['pm10'] = df['pm10'].astype(float)
df['so2'] = df['so2'].astype(float)
df['co'] = df['co'].astype(float)
df['o3'] = df['o3'].astype(float)
df['no2'] = df['no2'].astype(float)

# Find AVG for feature data
avg_pm10 = df['pm10'].mean(skipna=True)
avg_so2 = df['so2'].mean(skipna=True)
avg_co = df['co'].mean(skipna=True)
avg_o3 = df['o3'].mean(skipna=True)
avg_no2 = df['no2'].mean(skipna=True)

print(avg_pm10)
```

```
print(avg_so2)
print(avg_co)
print(avg_o3)
print(avg_no2)

# Replace NaN value with corresponding feature AVG
df['pm10'] = df['pm10'].replace(np.nan, avg_pm10)
df['so2'] = df['so2'].replace(np.nan, avg_so2)
df['co'] = df['co'].replace(np.nan, avg_co)
df['o3'] = df['o3'].replace(np.nan, avg_o3)
df['no2'] = df['no2'].replace(np.nan, avg_no2)

print(df.head())
```

Target yang digunakan dalam *modelling* ini adalah kolom *Fault Detection Ground Truth* yang merepresentasikan fakta sebenarnya apakah terdapat kondisi abnormal atau tidak normal dalam ruangan dengan parameter terkontrol. Kemudian, Data dipisah menjadi *training data* dan *testing data* dengan proporsi 80% dan 20%.

```
# Splitting the Data for TRAINING and TESTING purpose

# Test size ratio
# 80% data training dan 20% data testing
TEST_SIZE_RATIO = 0.2

# Feature names
df_featureNames = df.columns[2:7]

# Feature data
df_x = df.iloc[:, 2:7]

# Target data
df_y = df.iloc[:, 9]

# Split the data using Scikit-Learn's train_test_split
df_x_train, df_x_test, df_y_train, df_y_test = train_test_split(df_x, df_y,
test_size=TEST_SIZE_RATIO, random_state=12)
```

3. Modelling

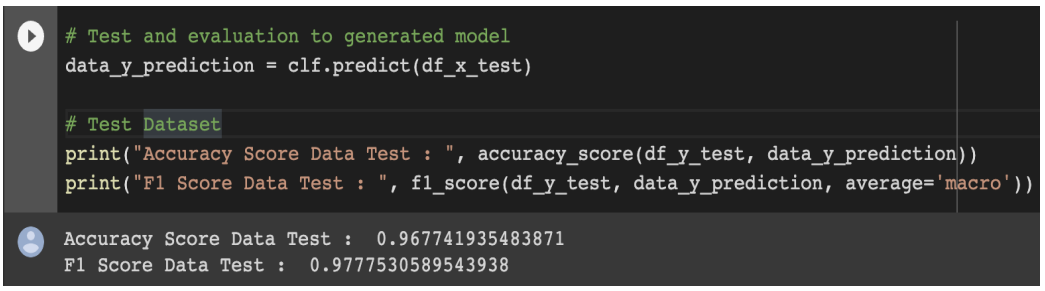
Tahapan setelah melakukan data preparation adalah *modelling*. Model yang dibangun menggunakan *Decision Tree Algorithm*. Model tersebut melakukan pembelajaran menggunakan *data train* yang sudah dipisah sebelumnya.

```
# Generating a Decision Tree Classifier model

clf = DecisionTreeClassifier().fit(df_x_train, df_y_train)

print("Decision Tree Classifier")
print()

dtl_export = export_text(clf, feature_names=df_featureNames.tolist())
print(dtl_export)
```

4. Model Evaluation

Evaluasi model diperlukan untuk mengevaluasi model yang telah dibangun pada tahap Modelling. Proses evaluasi model menggunakan *data test* yang sudah dipisah pada tahap Data Preparation. Metrik yang digunakan untuk mengevaluasi model adalah akurasi dan F1-score.

```
# Test and evaluation to generated model
data_y_prediction = clf.predict(df_x_test)

# Test Dataset
print("Accuracy Score Data Test : ", accuracy_score(df_y_test, data_y_prediction))
print("F1 Score Data Test : ", f1_score(df_y_test, data_y_prediction, average='macro'))
```



Hasil pengujian model terhadap data uji (*data_test*) yang dibangun menunjukkan nilai yang sangat baik yaitu dengan akurasi sekitar 0.967 dan F1 Score sekitar 0.977

Selain itu dilakukan pengujian menggunakan salah satu entri data dalam dataset, yaitu data entri pertama dengan ekspektasi target nilai 'BAIK'. Hasil prediksi menggunakan model yang dibuat menghasilkan target nilai yang sesuai dengan ekspektasi sehingga dapat dikatakan model yang berhasil dibangun memiliki ketepatan yang cukup bagus.

```
# Coba masukin data
inputDataFrame = pd.DataFrame([[30,20,10,32,9]], columns=df_featureNames.tolist())
result = clf.predict(inputDataFrame)
print(result)
```

```
['BAIK']
```

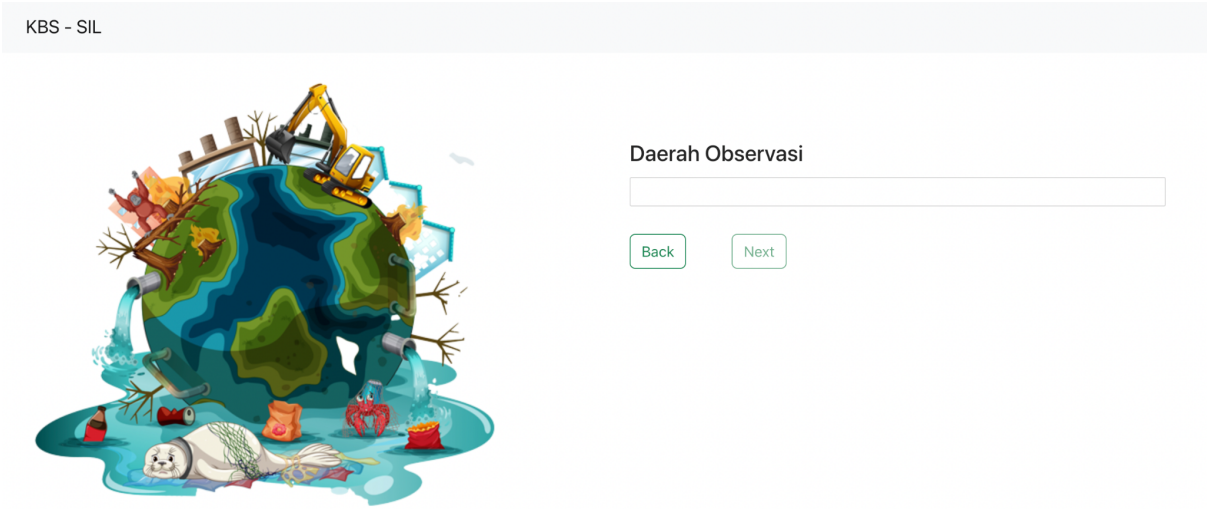
Analisa Hasil

1. **Hasil Deployment Program**

   Hasil prototipe Knowledge-based System yang dibuat terbagi ke dalam komponen Frontend yang diakses *end user* dan Backend yang mengandung *classifier* yang sudah dibuat dan melakukan prediksi data pada *request* dengan *classifier*, menghasilkan hasil prediksi berupa nilai *Fault Detection Ground Truth*. Sistem dapat diakses pada *url* berikut.

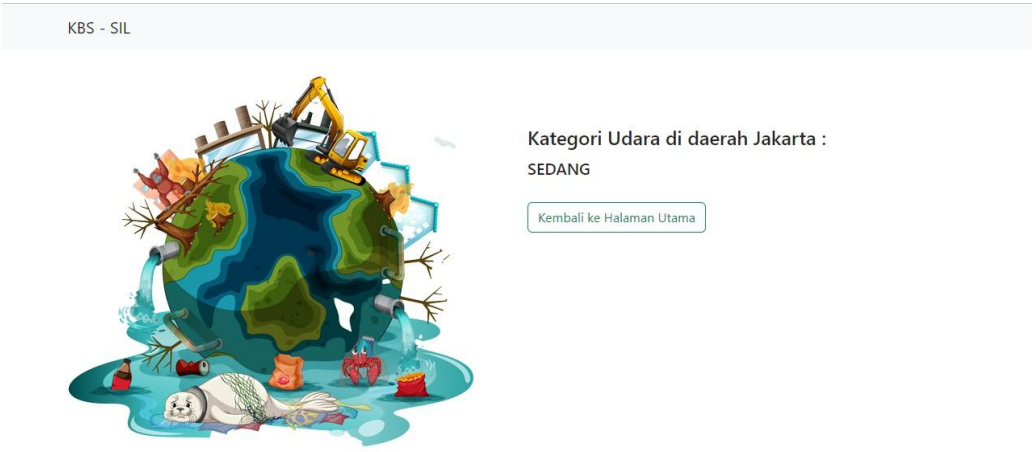
   Frontend : https://kbs-sil-frontend-ed4kbl429-dwibagus154.vercel.app/
   Backend (POST Request Only) : myahyaibrahim.pythonanywhere.com
   Repository Kode : https://github.com/myahyaibrahim/SIL-KBS-Data-Science

   Berikut tampilan antarmuka Frontend yang berhasil dibangun dengan mengintegrasikan program backend yang mengimplementasikan model *decision tree learning*. Nantinya pengguna akan diperintahkan untuk memasukkan input berupa daerah observasi, PM_{10}, CO, O_3, SO, dan NO.



2. **Hasil Analisis Sistem**

   Dari hasil model pembelajaran mesin yang telah dibuat, selanjutnya model akan dites dengan data baru yang tidak ada pada dataset sebelumnya. Setelah dimasukkan untuk data PM_{10}, CO, O_3 , SO, dan NO2 yang masing-masing bernilai 52, 35, 27, 22, 15 didapatkan hasil berupa kategori “SEDANG”.



Hasil kategori ini bisa didapatkan karena dataset yang baru dimasukkan tersebut akan dicari kategorinya pada model *decision tree learning* sebagai berikut.

```
Decision Tree Classifier

|--- o3 <= 48.50
|    |--- pm10 <= 50.50
|    |    |--- class: BAIK
|    |--- pm10 >  50.50
|    |    |--- class: SEDANG
|--- o3 >  48.50
|    |--- o3 <= 98.00
|    |    |--- so2 <= 38.50
|    |    |    |--- class: SEDANG
|    |    |--- so2 >  38.50
|    |    |    |--- class: BAIK
|    |--- o3 >  98.00
|    |    |--- class: TIDAK SEHAT
```

Kesimpulan

Kelompok kami telah berhasil menyelesaikan implementasi pengembangan SI dengan data science berbasis algoritma *decision tree learning* pada kasus *Fault Detection and Diagnostics* (FDD) untuk melakukan prediksi berjenis klasifikasi untuk ISPU pada udara di Jakarta pada bulan Januari 2020. Pengembangan model *machine learning* ini nantinya diharapkan mampu membantu pemerintah-pemerintah di Jakarta untuk membuat keputusan yang strategis guna mengurangi tingkat kualitas udara pada suatu daerah.