

LAPORAN TUGAS BESAR IF4070
PENGIMPLEMENTASIAN PENGEMBANGAN KECERDASAN
BUATAN SEDERHANA



DISUSUN OLEH

13519018 Mohammad Sheva Almeyda Sofjan

13519069 Muhammad Fikri. N

13519091 Mohammad Yahya Ibrahim

INSTITUT TEKNOLOGI BANDUNG
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
TEKNIK INFORMATIKA

2022/2023

Daftar Isi

Daftar Isi	1
Latar Belakang	2
Business Understanding	2
Data Understanding	2
Langkah Pengembangan dan Implementasi	4
Data Preparation	4
Modelling	5
Model Evaluation	5
Hasil	7
Deployment	7
Kesimpulan	10
Referensi	11

Latar Belakang

Business Understanding

Industri infrastruktur terus berkembang dalam mengembangkan sistem yang dapat memonitor dan mengontrol penggunaan energi supaya energi yang digunakan seminimal mungkin. Akan tetapi, masih terdapat kemungkinan terjadinya kesalahan pada sistem bangunan yang tidak dapat terdeteksi. Oleh karena itu, dikembangkan *Fault Detection and Diagnostics* (FDD) yang merupakan sebuah sistem yang dapat mengidentifikasi ketidaknormalan atau ketidakberesan pada peralatan vital seperti boiler, pendingin, pompa, *exhaust fan*, dan lain-lain. FDD dapat melakukan translasi dari data yang abnormal atau tidak beres ke sebuah kategori permasalahan yang dapat dimengerti oleh operator atau teknisi bangunan yang dilengkapi deskripsi detail permasalahan sampai akar permasalahan beserta solusinya.

Perusahaan XYZ bergerak dalam industri infrastruktur. Perusahaan ingin meningkatkan layanan agar dapat mengontrol penggunaan energi supaya energi yang digunakan seminimal mungkin. Oleh karena itu, perusahaan membuat sebuah proyek *data science* untuk menganalisis terkait *Fault Detection and Diagnostics* (FDD).

Data Understanding

Kebutuhan perusahaan untuk meningkatkan kualitas pembangunan dan layanan yang diberikan maka data terkait *Fault Detection and Diagnostics* (FDD) dikumpulkan untuk mendukung proyek *data science* ini. Data yang relevan terkait proyek ini diperoleh dari data.gov USA [1]. Data tersebut menelaah terkait *Fault Detection Ground Truth* yang penting untuk perusahaan agar dapat memberikan layanan yang semakin baik lagi dalam bidang pembangunan. Data yang ada merupakan hasil pengamatan seiring waktu terhadap *Air Handling Unit (AHU)* dari suatu bangunan. Kolom-kolom yang terdapat pada data tersebut di antaranya sebagai berikut.

1. Datetime, merupakan waktu pengamatan atau pengambilan entri data
2. Supply Air Temperature, menggambarkan hasil pengukuran *supply air temperature*
3. Supply Air Temperature Set Point, menggambarkan *set point* dari data kolom sebelumnya
4. Outdoor Air Temperature, menggambarkan hasil pengukuran *outdoor air temperature*
5. Mixed Air Temperature, menggambarkan hasil pengukuran *mixed air temperature*
6. Return Air Temperature, menggambarkan hasil pengukuran *return air temperature*
7. Supply Air Fan Status, menggambarkan status dari *supply air fan*
8. Return Air Fan Status, menggambarkan status dari *return air fan*
9. Supply Air Fan Speed Control Signal
10. Return Air Fan Speed Control Signal
11. Outdoor Air Damper Control Signal
12. Return Air Damper Control Signal
13. Cooling Coil Valve Control Signal
14. Heating Coil Valve Control Signal
15. Supply Air Duct Static Pressure
16. Supply Air Duct Static Pressure Set Point

17. Occupancy Mode Indicator, yang merupakan indikator apakah sistem beroperasi pada mode *occupied*/sibuk (1) atau tidak (0).
18. Fault Detection Ground Truth, merupakan indikator apakah ada *fault* pada AHU di hari dilakukannya pengamatan, di mana 0 menggambarkan *unfaulted* dan 1 menggambarkan *faulted*.

Data pada poin 1-5 merupakan data temperatur *Air Supply Unit (AHU)* yang memiliki nilai dalam satuan derajat Fahrenheit. Data pada poin 6-7 merupakan data status kipas milik AHU yang memiliki nilai kategorikal bernilai 0 (mati) atau 1 (menyala). Data pada poin 8-9 merupakan data kecepatan kipas milik AHU yang memiliki rentang nilai antara 0 hingga 1, di mana 0 merupakan kecepatan kipas 0% (mati), dan 1 merupakan kecepatan kipas 100% (kecepatan maksimum). Data pada poin 10-13 menggambarkan Control Signal untuk *dampers* dan *valves* milik AHU yang memiliki rentang nilai antara 0 hingga 1, di mana 0 merupakan keadaan tertutup secara penuh, dan 1 merupakan keadaan terbuka secara penuh. Data pada poin 14 dan 15 masing-masing merupakan hasil pengukuran tekanan dan *set point* (dalam satuan psi) untuk *supply air duct* milik AHU.

Langkah Pengembangan dan Implementasi

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

Data Preparation

Berdasarkan data yang diperoleh dilakukan pemilahan dan pembersihan data. Data yang digunakan untuk *modelling* hanya untuk fitur atau kolom yang relevan saja. Untuk data kolom seperti *datetime* (tanggal) dibuang dan tidak dimasukkan ke dalam fitur untuk proses *modelling*. Setelah pemilahan dan pembersihan data diperoleh fitur - fitur yang digunakan adalah sebagai berikut :

1. AHU: Supply Air Temperature (SAT)
2. AHU: Supply Air Temperature Set Point (SATSP)
3. AHU: Outdoor Air Temperature (OAT)
4. AHU: Mixed Air Temperature (MAT)
5. AHU: Return Air Temperature (RAT)
6. AHU: Supply Air Fan Status (SAFS)
7. AHU: Return Air Fan Status (RAFS)
8. AHU: Supply Air Fan Speed Control Signal (SAFSCS)
9. AHU: Return Air Fan Speed Control Signal (RAFSCS)
10. AHU: Outdoor Air Damper Control Signal (OADCS)
11. AHU: Return Air Damper Control Signal (RADCS)
12. AHU: Cooling Coil Valve Control Signal (CCVCS)
13. AHU: Heating Coil Valve Control Signal (HCVCS)
14. AHU: Supply Air Duct Static Pressure Set Point (SADSPSP)
15. AHU: Supply Air Duct Static Pressure (SADSP)
16. Occupancy Mode Indicator (OMI)

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

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

# Feature names
data_feature_names = data.columns[1:17]
data_feature_names
```

```
# Store the feature datas
data_x = data.iloc[:, 1:17]

# Store the target data
data_y = data.iloc[:, 17]

# Split the data using Scikit-Learn's train_test_split
data_x_train, data_x_test, data_y_train, data_y_test =
train_test_split(data_x, data_y, test_size=TEST_SIZE_RATIO, random_state=12)
```

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.

```
dtl_model = DecisionTreeClassifier()

dtl_model.fit(data_x_train, data_y_train)
```

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.

```
# Generating prediction models
dtl_y_test_prediction = dtl_model.predict(data_x_test)

dtl_test_accuracy = accuracy_score(data_y_test, dtl_y_test_prediction)
dtl_test_f1 = f1_score(data_y_test, dtl_y_test_prediction)

# Test Dataset
print("Accuracy Score Data Test : ", dtl_test_accuracy)
print("F1 Score Data Test : ", dtl_test_f1)
```

```

# Generating prediction models
dtl_y_test_prediction = clf.predict(data_x_test)

dtl_test_accuracy = accuracy_score(data_y_test, dtl_y_test_prediction)
dtl_test_f1 = f1_score(data_y_test, dtl_y_test_prediction)

# Test Dataset
print("Accuracy Score Data Test : ", dtl_test_accuracy)
print("F1 Score Data Test : ", dtl_test_f1)

Accuracy Score Data Test :  0.9892342739564962
F1 Score Data Test :  0.9937318158480232

```

Hasil pengujian model terhadap data uji (*data_test*) yang dibangun menunjukkan nilai yang sangat baik yaitu dengan akurasi sekitar 0.989 dan F1 Score sekitar 0.9937.

```

inputDataFrame =
pd.DataFrame([[68,55.04,32,68,75.2,0,0,0,0,0,0,0,0,0.04,0,0]],
columns=data_feature_names.tolist())
result = dtl_model.predict(inputDataFrame)
print(result)

```

```

[ ] # Coba masukan data
inputDataFrame = pd.DataFrame([[68,55.04,32,68,75.2,0,0,0,0,0,0,0,0,0.04,0,0]], columns=data_feature_names.tolist())
result = clf.predict(inputDataFrame)
print(result)

[0]

```

Selain itu dilakukan pengujian menggunakan salah satu entri data dalam dataset, yaitu data entri pertama dengan ekspektasi target nilai 0. 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.

Hasil

Deployment

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://moshval.github.io/simple_kbs

Backend (POST Request Only) : <https://simple-kbs.herokuapp.com/predict>

Repository Kode : https://github.com/moshval/simple_kbs

Pada bagian *backend*, program menerima format berupa *request* masukan (*input*) dari 16 kolom fitur

```
{
  "SAT" : 68,
  "SATSP" : 55.04,
  "OAT" : 32,
  "MAT" : 68,
  "RAT" : 75.2,
  "SAFS" : 0,
  "RAFS" : 0,
  "SAFSCS" : 0,
  "RAFSCS" : 0,
  "OADCS" : 0,
  "RADCS" : 0,
  "CCVCS" : 0,
  "HCVCS" : 0,
  "SADSPSP" : 0.04,
  "SADSP" : 0,
  "OMI" : 0
}
```

Berikut tampilan antarmuka Frontend yang berhasil dibangun dengan mengintegrasikan program backend yang mengimplementasikan model *decision tree learning*. Pada halaman utama, terdapat 16 *form* yang merepresentasikan fitur-fitur yang digunakan untuk melakukan prediksi nilai *Fault Detection Ground Truth* yang pengguna perlu masukkan

Supply Air Temperature :

68

Supply Air Temperature Set Point :

55.04

Outdoor Air Temperature :

32

Mixed Air Temperature :

68

Return Air Temperature :

75.2

Supply Air Fan Status :

☐ On☒ Off

Return Air Fan Status :

☐ On☒ Off

Supply Air Fan Speed Control Signal :

0

0 - fan speed is 0%, 1 - fan speed is 100%

Return Air Fan Speed Control Signal :

0

0 - fan speed is 0%, 1 - fan speed is 100%

Outdoor Air Damper Control Signal :

0

0 - damper should be fully closed, 1 - damper should be fully open

Return Air Damper Control Signal :

0

0 - damper should be fully closed, 1 - damper should be fully open

Cooling Coil Valve Control Signal :

0

0 - valve should be fully closed, 1 - valve should be fully open

Heating Coil Valve Control Signal :

0

0 - valve should be fully closed, 1 - valve should be fully open

Supply Air Duct Static Pressure Set Point :

0.04

Supply Air Duct Static Pressure :

0

Occupancy Mode Indicator :

☐ Occupied mode☒ Unoccupied mode

Submit

Setelah memasukkan semua 16 input form yang diminta, pengguna dapat menekan tombol *submit* untuk mendapatkan prediksi nilai *Fault Detection Ground Truth* yang ditampilkan pada halaman atas website. Untuk melakukan prediksi ulang, pengguna dapat mengisi ulang 16 *form* dan selanjutnya menekan tombol *submit*.

✕

Result | Fault Detection Ground Truth : Unfaulted (0)

Outdoor Air Temperature :

32

Mixed Air Temperature :

68

Return Air Temperature :

75.2

Supply Air Fan Status :

☐ On☒ Off

Return Air Fan Status :

☐ On☒ Off

Supply Air Fan Speed Control Signal :

0

0 - fan speed is 0%, 1 - fan speed is 100%

Return Air Fan Speed Control Signal :

0

0 - fan speed is 0%, 1 - fan speed is 100%

Outdoor Air Damper Control Signal :

0

0 - damper should be fully closed, 1 - damper should be fully open

Return Air Damper Control Signal :

0

0 - damper should be fully closed, 1 - damper should be fully open

Cooling Coil Valve Control Signal :

0

0 - valve should be fully closed, 1 - valve should be fully open

Heating Coil Valve Control Signal :

0

0 - valve should be fully closed, 1 - valve should be fully open

Supply Air Duct Static Pressure Set Point :

0.04

Supply Air Duct Static Pressure :

0

Occupancy Mode Indicator :

☐ Occupied mode☒ Unoccupied mode

Submit

Kesimpulan

Kelompok kami telah berhasil menyelesaikan implementasi pembelajaran mesin berbasis algoritma *decision tree learning* pada kasus *Fault Detection and Diagnostics* (FDD) untuk melakukan prediksi berjenis klasifikasi apakah terdapat kesalahan pada alat vital dalam bangunan seperti boiler, pendingin, pompa, *exhaust fan*, dan lain-lain. Untuk membangun model *decision tree learning*, digunakan dataset yang telah dipisah menjadi *training data* dan *testing data* dengan proporsi 80% dan 20%. Pengujian dan evaluasi dilakukan pada model yang berhasil dibangun terhadap *testing data* dengan menggunakan metrik *accuracy* dan *F1-score*. Didapat hasil pengujian menggunakan metrik *accuracy* dan *F1-score* masing-masing sebesar 0.989 dan 0.9937, menunjukkan tingkat akurasi ketepatan model sudah cukup baik.

Model *decision tree learning* yang berhasil dibangun diintegrasikan dalam sebuah website yang dapat digunakan oleh pengguna. Pengguna hanya perlu memasukan beberapa data parameter pada *form* yang tersedia seperti temperatur udara, status kipas angin (*Air Fan*), cooling, heating valve control, dan lain-lain. Selanjutnya kesimpulan berupa kesimpulan prediksi berupa klasifikasi (kelas) akan ditampilkan pada halaman website.

Referensi

- [1] <https://catalog.data.gov/dataset/data-sets-for-evaluation-of-building-fault-detection-and-diagnostics-algorithms-2de50>
- [2] <https://www.zenatix.com/what-actually-is-fault-detection-and-diagnosis-in-buildings/>