

LAPORAN UJIAN AKHIR SEMESTER
MATA KULIAH SAINS DATA

Dosen : Isa Albanna, S.Si., M.Si.



Oleh:

Adam Rahmat Ilahi	[13.2021.1.01030]
Abdullah Atiq	[13.2021.1.01046]
Muhammad Zidane Ramadhan	[13.2021.1.01054]

PROGRAM STRATA-1

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI ADHI TAMA SURABAYA

2024

DAFTAR ISI

DAFTAR ISI	2
BAB I	3
PENDAHULUAN	3
1. Latar Belakang.....	3
2. Rumusan Masalah	3
3. Tujuan Penelitian.....	4
4. Batasan Penelitian	4
BAB II.....	5
PENJELASAN.....	5
1. Rumus Dasar	5
2. Implementasi Iterasi	6
3. Konteks Algoritma ID3	6
BAB III	7
KESIMPULAN	7
REFERENSI	8
LAMPIRAN	9

BAB I

PENDAHULUAN

1. Latar Belakang

Pohon keputusan (decision tree) adalah salah satu metode klasifikasi yang populer dalam pembelajaran mesin (machine learning) karena kemampuannya dalam menghasilkan model yang sederhana, mudah diinterpretasi, dan efisien. Dalam fenomena idealis, model klasifikasi yang ideal harus mampu mengolah data secara efisien, memberikan prediksi yang akurat, serta mempertimbangkan faktor kompleksitas dan kemudahan implementasi. Namun, dalam fenomena realistis, banyak algoritma pembelajaran mesin yang menghasilkan model terlalu kompleks, sulit diinterpretasi, atau bahkan overfitting pada data pelatihan.

Kasus diabetes merupakan salah satu studi kasus yang relevan karena melibatkan banyak faktor, seperti usia, kadar glukosa, tekanan darah, dan indeks massa tubuh (BMI), yang semuanya berkontribusi terhadap risiko diabetes. Identifikasi pasien yang berisiko diabetes sangat penting untuk mendukung tindakan pencegahan. Dalam penelitian ini, algoritma **ID3 (Iterative Dichotomiser 3)** digunakan untuk membangun model pohon keputusan berdasarkan kriteria entropi. ID3 adalah algoritma dasar untuk menghasilkan pohon keputusan yang berfokus pada pemilihan fitur dengan Information Gain tertinggi pada setiap tingkat pemisahan.

Dengan menggunakan dataset diabetes yang mengandung beberapa atribut penting seperti kadar glukosa dan hasil diagnosis (Outcome), penelitian ini bertujuan untuk mengeksplorasi efisiensi algoritma ID3 dalam menghasilkan model klasifikasi yang efektif, mudah dipahami, dan cukup akurat.

2. Rumusan Masalah

1. Bagaimana algoritma ID3 dapat digunakan untuk membangun model klasifikasi berbasis pohon keputusan pada dataset diabetes?

2. Seberapa baik akurasi dan efektivitas model pohon keputusan ID3 dibandingkan dengan tingkat kompleksitas yang dihasilkan?
3. Bagaimana iterasi pada pohon keputusan mempengaruhi hasil klasifikasi?

3. Tujuan Penelitian

1. Mengimplementasikan algoritma ID3 untuk membangun model klasifikasi diabetes berbasis pohon keputusan.
2. Mengevaluasi performa model berdasarkan tingkat akurasi, entropi, dan Information Gain.
3. Memahami proses iterasi dalam membangun pohon keputusan untuk kasus klasifikasi.

4. Batasan Penelitian

Penelitian ini menggunakan dataset diabetes dengan atribut utama seperti **Glucose, Blood Pressure, BMI**, dan lainnya. Metode yang digunakan adalah ID3 berbasis entropi tanpa membandingkan dengan algoritma lain. Analisis dibatasi hingga tiga iterasi dengan visualisasi pohon keputusan.

BAB II

PENJELASAN

1. Rumus Dasar

Pohon keputusan ID3 memanfaatkan kriteria entropi untuk mengukur homogenitas suatu grup data. Rumus untuk menghitung entropi adalah:

$$Entropy(S) = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

dengan:

- S : dataset yang sedang dievaluasi,
- p_i : proporsi elemen kelas i dalam dataset S .

Selanjutnya, untuk menentukan fitur terbaik sebagai akar pohon, digunakan **Information Gain**:

$$\begin{aligned} Inform. Gain(S, A) \\ = Entropy(S) - \sum_{i=1}^v \frac{|S_i|}{|S|} \cdot Entropy(S) \end{aligned}$$

dengan:

- A : Atribut
- v : Jumlah nilai unik dari atribut A ,
- S_i : subset data yang dihasilkan setelah pemisahan berdasarkan nilai i .

2. Implementasi Iterasi

1. **Iterasi Pertama:** Memilih fitur dengan Information Gain tertinggi sebagai akar pohon.
2. **Iterasi Kedua:** Mengembangkan cabang dari akar dengan memilih fitur terbaik berikutnya pada subset data.
3. **Iterasi Ketiga:** Melengkapi pohon hingga seluruh data selesai dipisahkan, menghasilkan pohon penuh.

Kode program dari setiap iterasi dapat disisipkan dalam lampiran atau bagian terpisah, seperti pada "Lampiran Kode". Hasil klasifikasi dan visualisasi pohon dapat dimasukkan di bagian ini, bersama grafik akurasi dan evaluasi model.

3. Konteks Algoritma ID3

Dengan menggunakan algoritma ID3 berbasis Entropi maka dapat diraih hasil yang tepat untuk memutuskan suatu akar dari pokok permasalahan dari penyakit diabetes, karena sudah melakukan tiga iterasi untuk mendapatkan hasil terbaik. Maka dari itu setiap hasil dari fitting yang menghasilkan penambahan cabang dari pohon Keputusan merupakan akurasi yang semakin akurat.

Jika kita lihat pada hasil iterasi pertama bahwa terdapat dua entropi berpengaruh dengan hasil yang lebih rendah yang mendekati dengan kriteria atribut penyakit diabetes, sedangkan yang memiliki angka lebih tinggi maka dapat dinyatakan false/kesalahan yang tidak sesuai dengan kriteria atribut. Dengan hasil tersebut maka hasil percabangan dan pengujian pertama dari iterasi pertama mendapatkan hasil dua node true dan false seperti pada lampiran.

Pada iterasi yang kedua memunculkan node dari hasil entropi yang di dapatkan pada iterasi yang kedua, node dari yang sebelumnya telah di dapatkan akan di uji dan mendapatkna nilai fitting yang lebih akurat. Selanjutnya, iterasi ketiga menghasilkan daripada bentuk Algoritma ID3 sebagai decision tree yang sebenarnya menggunakan python, pada lampiran hasil visualiasasi terdapat hingga 17 node yang ada dalam dataset diabetes tersebut dengan berawal dari iterasi pertama yang telah di tentukan.

BAB III

KESIMPULAN

Penelitian ini menunjukkan bahwa algoritma ID3 berbasis entropi merupakan teknik yang efektif untuk membangun pohon keputusan. Dengan tiga iterasi, model berhasil menghasilkan akurasi yang meningkat secara bertahap seiring dengan penambahan kedalaman pohon. Namun, risiko overfitting perlu diperhatikan ketika pohon menjadi terlalu kompleks.

Studi ini juga menegaskan pentingnya pemilihan fitur awal dengan Information Gain tertinggi, karena fitur ini menentukan struktur dasar pohon. Algoritma ID3 dapat diterapkan pada dataset lain dengan karakteristik serupa untuk kasus klasifikasi yang membutuhkan interpretasi mudah.

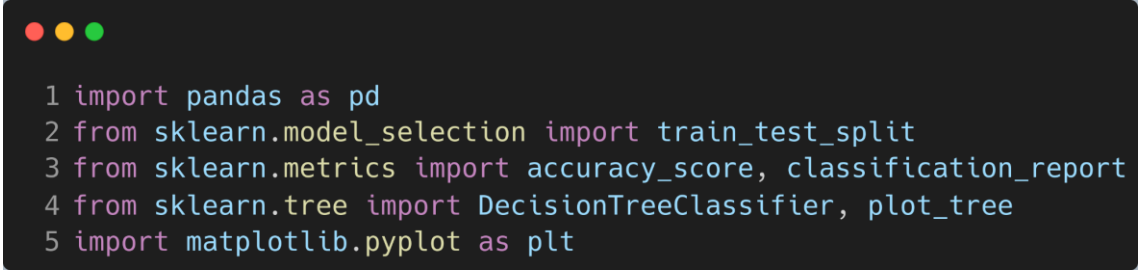
REFERENSI

- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106.
- Han, J., Kamber, M., & Pei, J. (2020). Data mining: Concepts and techniques (3rd ed.). Morgan Kaufmann.
- Safavian, S. R., & Landgrebe, D. (2021). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 660–674.
- Rokach, L., & Maimon, O. (2021). Decision trees. *Data Mining and Knowledge Discovery Handbook*, 165–192.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (2017). Classification and regression trees. CRC press.
- Mitchell, T. M. (2020). Machine learning. McGraw-Hill Education.
- Murthy, S. K. (2021). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4), 345–389.
- Song, Y. Y., & Ying, L. U. (2021). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2), 130.
- Kotsiantis, S. B. (2021). Decision trees: A recent overview. *Artificial Intelligence Review*, 39(4), 261–283.
- Tomar, D., & Agarwal, S. (2021). A survey on decision tree-based approaches in data mining. *Journal of Information Processing Systems*, 13(4), 867–880.

LAMPIRAN

1. kode Program Iterasi:

1.1 import library & dependency



```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.metrics import accuracy_score, classification_report
4 from sklearn.tree import DecisionTreeClassifier, plot_tree
5 import matplotlib.pyplot as plt
```

Gambar : Proses impor library Python yang diperlukan untuk mengimplementasikan algoritma Decision Tree. Library yang diimpor mencakup library untuk manipulasi data (Pandas), pelatihan model machine learning (scikit-learn), dan pembuatan visualisasi (Matplotlib).

2.1 inialisasi data dari dataset



```
6 df = pd.read_csv('diabetes.csv')
```

Gambar : Cara memuat dataset diabetes ke dalam sebuah DataFrame menggunakan Pandas. Dataset ini memuat data yang diperlukan untuk melatih model machine learning yang akan digunakan untuk klasifikasi diabetes.

1. Proses untuk pengecekan nilai kosong pada data dataset

```
7 print(df.isnull().sum())
```

Gambar : Proses pengecekan untuk mengetahui apakah terdapat nilai yang hilang pada dataset ditampilkan menggunakan fungsi `isnull().sum()`. Langkah ini penting untuk memastikan bahwa dataset siap untuk analisis lebih lanjut tanpa adanya data yang hilang.

2. inisialisasi sumbu X dan sumbu Y

```
8 X = df.drop('Outcome', axis=1)
9 y = df['Outcome']
```

Gambar : Pembagian dataset menjadi fitur (X) dan target (y). Fitur adalah semua kolom selain "Outcome", sedangkan target adalah kolom "Outcome" yang menunjukkan apakah seseorang mengidap diabetes atau tidak.

3. Membagi dataset menjadi data pelatihan dan pengujian

```
10 X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=42)
```

Gambar : Kode dibagian `train_test_split()` digunakan untuk membagi dataset menjadi dua bagian: satu untuk pelatihan model (training set) dan satu lagi untuk pengujian (testing set). Pembagian ini penting agar model dapat dievaluasi dengan data yang belum pernah dilihat sebelumnya.

4. Membangun model untuk Decision Tree dengan Iterasi 1

```
11 model_iter1 = DecisionTreeClassifier(criterion='entropy', max_depth=1,
    random_state=42)
12 model_iter1.fit(X_train, y_train)
13
14 y_pred_iter1 = model_iter1.predict(X_test)
15 print(f"Iterasi 1 - Akurasi: {accuracy_score(y_test, y_pred_iter1)}")
16 print("Iterasi 1 - Laporan Klasifikasi:")
17 print(classification_report(y_test, y_pred_iter1))
18
19 plt.figure(figsize=(10, 10))
20 plot_tree(model_iter1, feature_names=X.columns, class_names=['No', 'Yes'],
    filled=True)
21 plt.title("Pohon Keputusan - Iterasi 1", fontsize = 40)
22 plt.show()
```

Gambar : Proses pembangunan model Decision Tree pada iterasi pertama menggunakan parameter max_depth=1. Model ini akan menghasilkan pohon keputusan pertama berdasarkan data tersebut.

5. Membangun model untuk Decision Tree dengan Iterasi 2

```
23 model_iter2 = DecisionTreeClassifier(criterion='entropy', max_depth=2,
    random_state=42)
24 model_iter2.fit(X_train, y_train)
25
26 y_pred_iter2 = model_iter2.predict(X_test)
27 print(f"Iterasi 2 - Akurasi: {accuracy_score(y_test, y_pred_iter2)}")
28 print("Iterasi 2 - Laporan Klasifikasi:")
29 print(classification_report(y_test, y_pred_iter2))
30
31 plt.figure(figsize=(10, 10))
32 plot_tree(model_iter2, feature_names=X.columns, class_names=['No', 'Yes'],
    filled=True)
33 plt.title("Pohon Keputusan - Iterasi 2", fontsize = 40)
34 plt.show()
```

Gambar : Iterasi kedua dari pohon keputusan dibuat dengan max_depth=2. Dengan meningkatkan kedalaman pohon, model akan menangkap lebih banyak detail dan meningkatkan presisi dalam membuat keputusan.

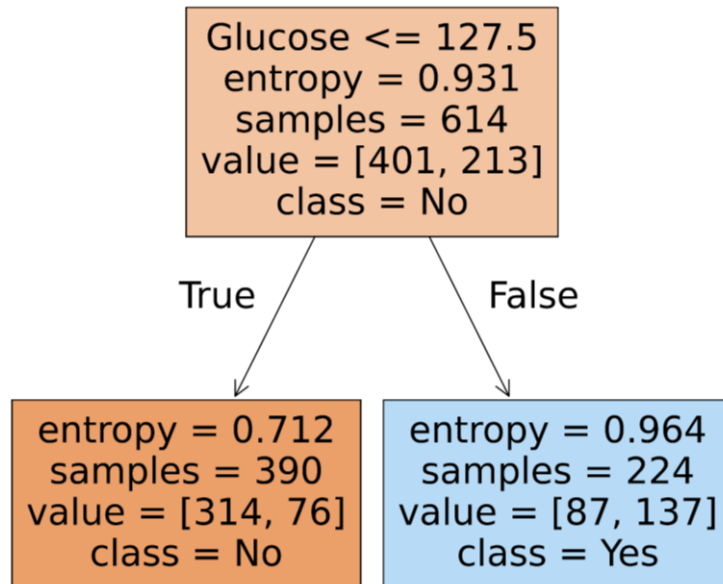
6. Membangun model untuk Decision Tree dengan Iterasi 3

```
35 model_iter3 = DecisionTreeClassifier(criterion='entropy', random_state=42)
36 model_iter3.fit(X_train, y_train)
37
38 y_pred_iter3 = model_iter3.predict(X_test)
39 print(f"Iterasi 3 - Akurasi: {accuracy_score(y_test, y_pred_iter3)}")
40 print("Iterasi 3 - Laporan Klasifikasi:")
41 print(classification_report(y_test, y_pred_iter3))
42
43 plt.figure(figsize=(40, 10))
44 plot_tree(model_iter3, feature_names=X.columns, class_names=['No', 'Yes'],
            filled=True)
45 plt.title("Pohon Keputusan - Iterasi 3", fontsize = 40)
46 plt.savefig("decision_tree_plot_iterasi_3.svg", format='svg',
            bbox_inches='tight', dpi=1200)
47 plt.savefig("decision_tree_plot_iterasi_3.pdf", format='pdf',
            bbox_inches='tight', dpi=1200)
48 plt.savefig("decision_tree_plot_iterasi_3.png", format='png',
            bbox_inches='tight', dpi=1200)
49 plt.show()
```

Gambar : Iterasi ketiga di mana pohon keputusan dibangun dengan kedalaman yang lebih tinggi. Dengan max_depth yang lebih besar, model ini semakin kompleks dan diharapkan dapat membuat keputusan yang lebih akurat. Ini juga menunjukkan bahwa model lebih "fit" dengan data setelah iterasi yang lebih dalam.

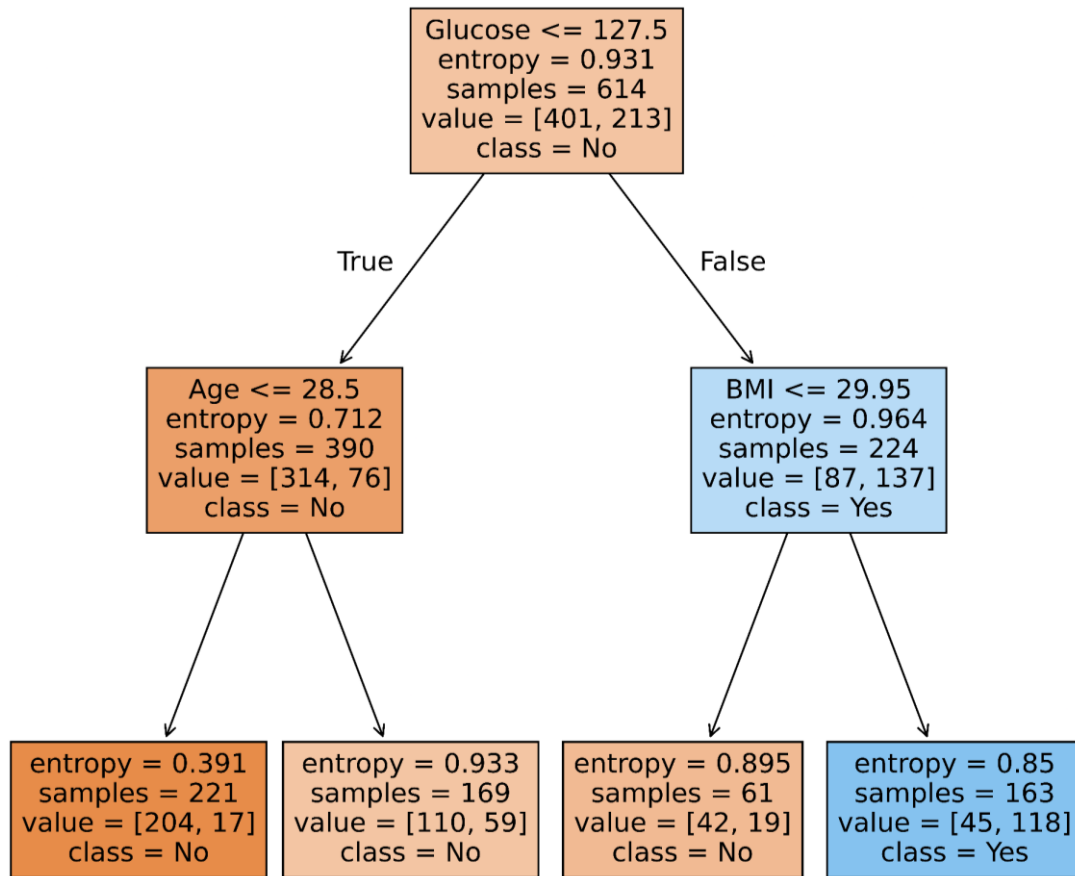
Hasil Visualisasi:

1. Hasil iterasi 1 Decision Tree



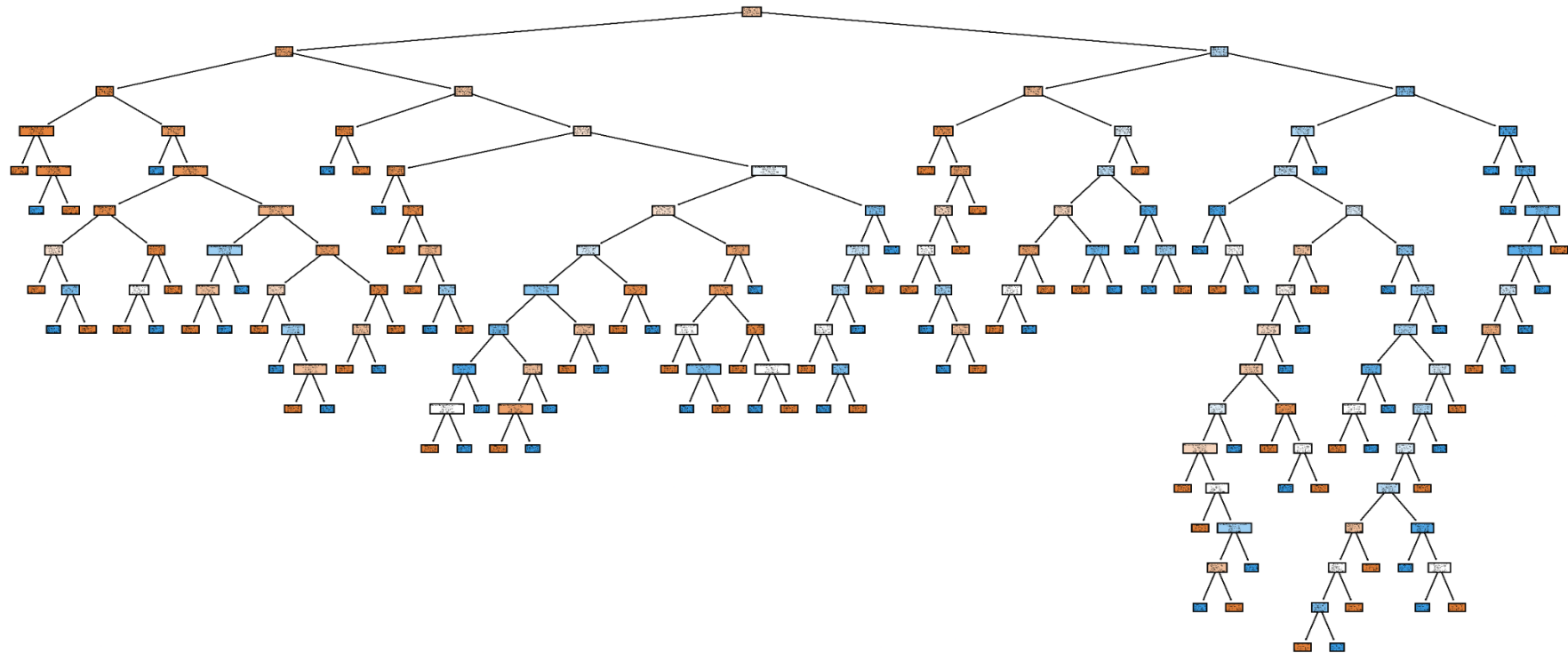
Gambar : Gambar diatas menunjukkan pohon keputusan pertama yang dibangun setelah iterasi pertama. Pada pohon ini, fitur yang digunakan untuk pemisahan adalah "Glucose <= 127.5". Pohon ini kemudian dibagi menjadi dua cabang, yaitu "True" dan "False". Setiap cabang menampilkan entropi, jumlah sampel, nilai-nilai kelas (No/Yes), dan informasi lain seperti jumlah elemen dalam setiap kelas.

2. Hasil iterasi 2 Decision Tree



Gambar : Gambar diatas menunjukkan pohon keputusan yang lebih dalam, setelah iterasi kedua. Setelah pemisahan pertama berdasarkan "Glucose <= 127.5", pohon ini membagi lebih lanjut cabang "True" berdasarkan fitur "Age <= 28.5", dan cabang "False" berdasarkan fitur "BMI <= 29.95". Masing-masing cabang menampilkan informasi entropi, jumlah sampel, dan kelas yang paling dominan pada setiap simpul.

3. Hasil iterasi 3 Decision Tree



Gambar : Dari gambar ini menunjukkan pohon keputusan lengkap setelah iterasi ketiga. Pohon ini lebih kompleks, dengan banyak simpul yang mewakili berbagai pembagian berdasarkan fitur-fitur seperti Glucose, Age, BMI, dan lainnya. Setiap simpul menunjukkan informasi terkait entropi, jumlah sampel, dan nilai kelas yang diprediksi.

Link Github : https://github.com/waroeng-kopi/k7_data_sains