LAPORAN UJIAN AKHIR SEMESTER
KECERDASAN KOMPUTASI

"Implementasi Algoritma K-Means Clustering dan Perhitungan Entropy untuk Klasifikasi Data"

Dosen Pengampu :

Saiful Nur Budiman, S.Kom., M.Kom.

Disusun Oleh :

Isra Naswa Reyka Swahili	(23104410006)
Agistha Ardha Sulistyo P.	(23104410009)
Zaki Zakaria Zakse	(23104410010)
Jusafa Ido Ad'hareza	(23104410022)
Jovanda Kelvin Wibawa P.	(23104410035)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS ISLAM BALITAR

Januari 2026

KATA PENGANTAR

Alhamdulillah, dengan rasa syukur kehadirat Allah SWT. atas rahmat dan hidayah-Nya, serta Rasulullah saw. atas risalah yang dibawanya, sehingga penyusun dapat menyelesaikan Laporan UAS mata kuliah Kecerdasan Komputasi di Universitas Islam Balitar, Blitar. Terima kasih penyusun sampaikan kepada Bapak Saiful Nur Budiman, S.Kom., M.Kom., selaku Dosen Pengampu.

Segala usaha dan upaya telah penyusun lakukan untuk menyempurnakan penulisan laporan ini, namun penyusun menyadari bahwa laporan ini masih jauh dari kata sempurna. Semoga laporan ini bermanfaat bagi semua pihak yang membacanya.

Blitar, 13 Januari 2026

P e n y u s u n

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan digitalisasi dalam berbagai sektor kehidupan telah menghasilkan volume data yang sangat besar dan terus bertambah setiap harinya. Data-data tersebut menyimpan informasi berharga yang dapat dimanfaatkan untuk pengambilan keputusan strategis, namun seringkali informasi tersebut tersembunyi di balik kompleksitas dan volume data yang besar. Oleh karena itu, diperlukan suatu metode untuk mengekstraksi pengetahuan dan pola-pola tersembunyi dari kumpulan data yang ada.

Data mining merupakan salah satu cabang ilmu komputer yang berfokus pada penemuan pola, informasi, dan pengetahuan yang berguna dari kumpulan data yang besar. Dalam praktiknya, data mining menggunakan berbagai teknik dan algoritma untuk menganalisis data dari berbagai perspektif dan mengubahnya menjadi informasi yang dapat digunakan untuk meningkatkan efisiensi, mengurangi biaya, atau meningkatkan pendapatan dalam berbagai bidang seperti bisnis, kesehatan, pendidikan, dan lain sebagainya.

Dua pendekatan utama dalam data mining adalah unsupervised learning dan supervised learning. Unsupervised learning adalah pendekatan yang tidak memerlukan label atau kategori yang sudah ditentukan sebelumnya, dan salah satu metode yang paling populer dalam kategori ini adalah clustering atau pengelompokan data. Clustering bertujuan untuk mengelompokkan data berdasarkan kesamaan karakteristik atau fitur yang dimiliki, sehingga data dalam satu kelompok memiliki kemiripan yang tinggi, sedangkan data antar kelompok memiliki perbedaan yang signifikan.

K-Means Clustering merupakan salah satu algoritma clustering yang paling banyak digunakan karena kesederhanaannya dan efektivitasnya dalam menangani dataset berukuran besar. Algoritma ini bekerja dengan cara membagi data ke dalam K kelompok berdasarkan jarak Euclidean terhadap centroid atau pusat kelompok. Proses iteratif dilakukan hingga posisi centroid menjadi stabil dan tidak mengalami perubahan signifikan, yang menandakan bahwa pengelompokan telah optimal.

Di sisi lain, supervised learning memerlukan data yang sudah memiliki label atau kategori untuk melakukan proses pembelajaran dan klasifikasi. Dalam konteks ini, Decision Tree adalah salah satu metode klasifikasi yang populer karena mudah dipahami dan diinterpretasikan. Untuk membangun decision tree yang efektif, diperlukan metode pemilihan atribut yang tepat agar pohon keputusan yang dihasilkan memiliki tingkat akurasi yang tinggi dan tidak terlalu kompleks.

Entropy dan Information Gain merupakan konsep fundamental dalam teori informasi yang digunakan dalam algoritma decision tree, khususnya pada algoritma ID3 (Iterative Dichotomiser 3) dan C4.5. Entropy mengukur tingkat ketidakpastian atau kekacauan dalam suatu kumpulan data. Semakin tinggi nilai entropy, semakin heterogen atau beragam data tersebut. Sebaliknya, entropy yang rendah menunjukkan data yang homogen atau memiliki kecenderungan yang jelas. Information Gain digunakan untuk mengukur seberapa baik suatu atribut dapat memisahkan data berdasarkan kelas targetnya. Atribut dengan information gain tertinggi akan dipilih sebagai node pemisah dalam decision tree.

Dalam laporan ini, akan dilakukan analisis mendalam terhadap implementasi kedua pendekatan data mining tersebut. Pertama, akan dibahas implementasi algoritma K-Means Clustering untuk mengelompokkan data berdasarkan fitur numerik yang dimiliki. Proses iterasi akan dijelaskan secara detail mulai dari inisialisasi centroid awal, perhitungan jarak Euclidean, penugasan data ke cluster terdekat, hingga pembaruan posisi centroid pada setiap iterasi. Kriteria konvergensi berdasarkan perhitungan threshold juga akan dianalisis untuk menentukan kapan proses clustering dianggap telah konvergen.

Kedua, akan dilakukan analisis entropy dan information gain pada dataset yang memiliki atribut kategorikal. Dataset yang dianalisis berisi informasi mengenai kondisi cuaca dengan tiga atribut utama yaitu Wind (kekuatan angin), Temperature (tingkat suhu), dan Weather (kondisi cuaca), dengan target klasifikasi berupa keputusan biner (Yes/No). Melalui perhitungan entropy untuk setiap atribut dan information gain yang dihasilkan, akan ditentukan atribut mana yang paling efektif dalam memisahkan data dan memberikan informasi terbanyak untuk proses klasifikasi.

Kombinasi dari kedua metode ini memberikan pemahaman yang komprehensif tentang bagaimana data dapat dianalisis baik dari perspektif pengelompokan tanpa label (unsupervised) maupun klasifikasi dengan label (supervised). Hasil analisis ini diharapkan dapat memberikan wawasan yang berguna dalam pengambilan keputusan berbasis data dan menjadi referensi bagi penerapan data mining dalam berbagai domain aplikasi.

Melalui laporan ini, akan disajikan analisis yang sistematis dan terstruktur mengenai implementasi kedua metode tersebut, lengkap dengan perhitungan matematis, interpretasi hasil, dan kesimpulan yang dapat diambil dari setiap tahapan analisis yang dilakukan.

BAB II
LANDASAN TEORI

2.1 Data Mining

Data mining adalah proses ekstraksi pola dan pengetahuan yang tersembunyi dari kumpulan data yang besar menggunakan metode komputasi. Proses ini melibatkan berbagai teknik dari statistika, machine learning, dan database systems untuk menemukan informasi yang berguna dan dapat ditindaklanjuti.

2.2 K-Means Clustering

2.2.1 Definisi K-Means

K-Means adalah algoritma clustering yang membagi dataset menjadi K kelompok (cluster) berdasarkan kesamaan karakteristik. Setiap data point akan diassign ke cluster dengan centroid terdekat berdasarkan jarak Euclidean.

2.2.2 Langkah-langkah Algoritma K-Means

1. Inisialisasi: Tentukan jumlah cluster K dan pilih K data point secara acak sebagai centroid awal
2. Assignment: Hitung jarak setiap data point ke semua centroid, lalu assign data ke cluster dengan centroid terdekat
3. Update: Hitung centroid baru dengan merata-ratakan semua data point dalam setiap cluster
4. Konvergensi: Ulangi langkah 2-3 hingga centroid tidak berubah signifikan atau kriteria konvergensi terpenuhi

2.2.3 Perhitungan Jarak Euclidean

Jarak Euclidean antara dua titik (x_1, y_1) dan (x_2, y_2) dihitung dengan rumus:

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

2.2.4 Kriteria Konvergensi

Algoritma berhenti ketika perubahan posisi centroid (Delta) antara iterasi saat ini dan sebelumnya lebih kecil dari threshold yang ditentukan:

$$Delta = |F\_baru - F\_lama| < Threshold$$

2.3 Entropy dan Information Gain

2.3.1 Definisi Entropy

Entropy adalah ukuran ketidakpastian atau heterogenitas dalam suatu kumpulan data. Dalam konteks klasifikasi biner, entropy dihitung dengan rumus:

$$\text{Entropy}(S) = -p_1 \log_2(p_1) - p_2 \log_2(p_2)$$

Di mana:

- S adalah kumpulan data
- p_1 adalah proporsi kelas pertama
- p_2 adalah proporsi kelas kedua

Nilai entropy berkisar antara 0 (data homogen/tidak ada ketidakpastian) hingga 1 (data heterogen maksimal).

2.3.2 Information Gain

Information Gain mengukur pengurangan entropy setelah dataset dibagi berdasarkan suatu atribut. Atribut dengan information gain tertinggi memberikan informasi terbanyak untuk klasifikasi.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \Sigma[(|S_v|/|S|) \times \text{Entropy}(S_v)]$$

Di mana:

- S adalah dataset awal
- A adalah atribut yang digunakan untuk membagi data
- S_v adalah subset data untuk setiap nilai v dari atribut A

BAB III
ANALISIS DAN PEMBAHASAN

3.1 Implementasi Algoritma K-Means Clustering

3.1.1 Dataset dan Inisialisasi

Berdasarkan data yang diberikan, proses clustering dilakukan pada dataset dengan 10 data point yang memiliki dua fitur numerik (Fx dan Fy). Tujuan dari clustering ini adalah mengelompokkan data menjadi 3 cluster (K=3) berdasarkan kesamaan karakteristik kedua fitur tersebut.

Data Point yang Dianalisis:

Data	Fx	Fy
1	1	1
2	4	1
3	6	1
4	1	2
5	2	3
6	5	3
7	2	5
8	3	5
9	2	6
10	3	8

Data Awal:

Cluster 1 (K1): Data 1
Cluster 2 (K2): Data 3, 4, 6, 7, 10
Cluster 3 (K3): Data 2, 5, 8, 9

3.1.2 Iterasi 1

Iterasi 1

Menghitung Centroid Setiap Cluster (Iterasi 1)

Data	Fitur x	Fitur y	K1	K2	K3	K1 Fx	K1 Fy	K2 Fx	K2 Fy	K3 Fx	K3 Fy
1	1	1	*			1	1				
2	4	1			*					4	1
3	6	1		*				6	1		
4	1	2		*				1	2		
5	2	3			*					2	3
6	5	3		*				5	3		
7	2	5		*				2	5		
8	3	5			*					3	5
9	2	6			*					2	6
10	3	8		*				3	8		
Total			**1**	**5**	**4**	**1**	**1**	**17**	**19**	**11**	**15**

Kolom penting:

- Fitur x & Fitur y → koordinat data
- K1, K2, K3 → tanda * menunjukkan data masuk ke cluster tersebut
- K1 Fx, K1 Fy, K2 Fx, K2 Fy, K3 Fx, K3 Fy
  → nilai fitur x dan y yang dijumlahkan untuk tiap cluster

Dari tanda * pada tabel:
Cluster K1
Data yang masuk:

- Data 1 → (1, 1)

Jumlah anggota:

- 1 data

Total:

- ΣFx = 1
- ΣFy = 1

Centroid K1:

$$C1 = \left(\frac{1}{1}, \frac{1}{1}\right) = (1,1)$$

Cluster K2
Data yang masuk:

- Data 3 → (6, 1)
- Data 4 → (1, 2)
- Data 6 → (5, 3)
- Data 7 → (2, 5)
- Data 10 → (3, 8)

Jumlah anggota:

- 5 data

Total:

- ΣFx = **17**
- ΣFy = **19**

Centroid K2:

$$C2 = \left(\frac{17}{5}, \frac{19}{5}\right) = (3.4,\ 3.8)$$

Cluster K3

Data yang masuk:

- Data 2 → (4, 1)
- Data 5 → (2, 3)
- Data 8 → (3, 5)
- Data 9 → (2, 6)

Jumlah anggota:

- 4 data

Total:

- ΣFx = 11
- ΣFy = 15

Centroid K3:

$$C3 = \left(\frac{11}{4}, \frac{15}{4}\right) = (2.75,\ 3.75)$$

Kesimpulan Iterasi 1

Data telah terbagi ke dalam 3 cluster

Centroid baru yang dihasilkan:

- C1 = (1, 1)
- C2 = (3.4, 3.8)
- C3 = (2.75, 3.75)

Centroid ini akan digunakan pada iterasi berikutnya

Proses akan berhenti jika pembagian cluster tidak berubah lagi

Hasil Centroid Setiap Cluster:

Kelompok	Centroid Fitur x (Rata-rata)	Centroid Fitur y (Rata-rata)	Koordinat Centroid
1	1	1	**(1, 1)**
2	3.4	3.8	**(3.4, 3.8)**
3	2.75	3.75	**(2.75, 3.75)**

Cluster 1

- Jumlah data: **1**
- Rata-rata fitur x = **1**
- Rata-rata fitur y = **1**

Koordinat centroid Cluster 1:

$$\boldsymbol{C_1 = (1, 1)}$$

Artinya, pusat cluster 1 berada tepat pada titik data tersebut karena hanya memiliki satu anggota.

Cluster 2

- Jumlah data: 5
- Rata-rata fitur x = 3.4
- Rata-rata fitur y = 3.8

Koordinat centroid Cluster 2:

$$\boldsymbol{C_2 = (3.4, 3.8)}$$

Centroid ini menunjukkan bahwa data pada cluster 2 cenderung terkonsentrasi di area tengah dengan nilai fitur x dan y yang relatif lebih besar dibanding cluster 1.

Cluster 3

- Jumlah data: 4
- Rata-rata fitur x = 2.75
- Rata-rata fitur y = 3.75

Koordinat centroid Cluster 3:

$$C_3 = (2.75, 3.75)$$

Cluster ini memiliki karakteristik nilai fitur y yang cukup tinggi dengan nilai fitur x sedang.

Menghitung Jarak Data Ke Centroid (Euclidean Distance) - Iterasi 1

Perhitungan Jarak (Iterasi 1):

Data	Fx	Fy	Jarak ke C1 (1, 1)	Jarak ke C2 (3.4, 3.8)	Jarak ke C3 (2.75,	Jarak Min	K Baru	K Lama
1	1	1	**0.0000**	3.6878	3.2596	0.0000	**1**	1
2	4	1	**3.0000**	2.8636	3.0208	2.8636	**2**	3
3	6	1	**5.0000**	3.8210	4.2573	3.8210	**2**	2
4	1	2	**1.0000**	3.0000	2.4749	1.0000	**1**	2
5	2	3	**2.2361**	1.6125	1.0607	1.0607	**3**	3
6	5	3	**4.4721**	1.7889	2.3717	1.7889	**2**	2
7	2	5	**4.1231**	1.8439	1.4577	1.4577	**3**	2
8	3	5	**4.4721**	1.2649	1.2748	1.2649	**2**	3
9	2	6	**5.0990**	2.6077	2.3717	2.3717	**3**	3
10	3	8	**7.2801**	4.2190	4.2573	4.2190	**2**	2
Total						**19.8474**		

Cek Threshold:

F baru= 19.8474

Delta = |F baru - F lama| = |19.8474 - 0| = 19.8474

Karena Delta > T (0.1), maka Lanjut ke Iterasi 2

Penentuan Cluster (Iterasi 1)

Prinsipnya:

Data masuk ke cluster dengan jarak paling kecil

Contoh:

- **Data 4 (1,2)**
  - Jarak ke C1 = **1.0000** (paling kecil)
    → Masuk **Cluster 1**
- **Data 5 (2,3)**
  - Jarak ke C3 = **1.0607** (paling kecil)
    → Masuk **Cluster 3**
- **Data 8 (3,5)**
  - Jarak ke C2 = **1.2649** (paling kecil)
    → Masuk **Cluster 2**

Kolom Jarak Min menunjukkan jarak terkecil, dan kolom K Baru adalah hasil cluster terbaru.

Perubahan Cluster

Dengan membandingkan K Baru dan K Lama:

- Beberapa data berpindah cluster
- Artinya centroid sebelumnya belum stabil

Ini menandakan proses K-Means belum konvergen.

Total Jarak (Objective Function)

Jumlah total jarak minimum:

$$\sum d_{min} = \mathbf{19.8474}$$

Nilai ini disebut fungsi objektif, digunakan untuk mengecek perubahan antar iterasi.

Cek Threshold / Konvergensi
Diketahui:

- F baru = 19.8474
- F lama = 0
- Delta = |F baru – F lama| = 19.8474

Karena:

$$\Delta > 0.1$$

Maka:
Proses dilanjutkan ke Iterasi 2

3.1.3 Iterasi 2

Iterasi 2
Anggota cluster berdasarkan hasil "K Baru" dari Iterasi 1:
K1: Data 1, 4
K2: Data 2, 3, 6, 8, 10
K3: Data 5, 7, 9

Menghitung Centroid Setiap Cluster (Iterasi 2)

Data	Fx	Fy	K1	K2	K3	K1 Fx	K1 Fy	K2 Fx	K2 Fy	K3 Fx	K3 Fy
1	1	1	*			1	1				
2	4	1		*				4	1		
3	6	1		*				6	1		
4	1	2	*			1	2				
5	2	3			*					2	3
6	5	3		*				5	3		
7	2	5			*					2	5
8	3	5		*				3	5		
9	2	6			*					2	6
10	3	8		*				3	8		
Total			2	5	3	2	3	21	18	6	14

Pada Iterasi 2, pembagian anggota cluster dilakukan berdasarkan centroid baru (K Baru) hasil Iterasi 1. Dari hasil tersebut diperoleh:

- K1 beranggotakan Data 1 dan 4
- K2 beranggotakan Data 2, 3, 6, 8, dan 10
- K3 beranggotakan Data 5, 7, dan 9

Setiap data memiliki koordinat (Fx, Fy). Tanda bintang (*) pada kolom K1, K2, dan K3 menunjukkan bahwa data tersebut menjadi anggota cluster terkait. Nilai Fx dan Fy dari data yang masuk ke suatu cluster kemudian dipindahkan ke kolom Kx Fx dan Kx Fy sesuai clusternya.

Selanjutnya dilakukan perhitungan centroid baru dengan cara menjumlahkan seluruh nilai Fx dan Fy pada masing-masing cluster, lalu dibagi dengan jumlah anggota cluster:

- Cluster K1:
  Total Fx = 2, Total Fy = 3, Jumlah data = 2
  Centroid K1 = (2/2, 3/2) = (1, 1.5)
- Cluster K2:
  Total Fx = 21, Total Fy = 18, Jumlah data = 5
  Centroid K2 = (21/5, 18/5) = (4.2, 3.6)
- Cluster K3:
  Total Fx = 6, Total Fy = 14, Jumlah data = 3
  Centroid K3 = (6/3, 14/3) = (2, 4.67)

Centroid hasil Iterasi 2 inilah yang nantinya digunakan untuk menghitung jarak kembali pada iterasi berikutnya, hingga posisi centroid tidak berubah lagi (konvergen).

Hasil Centroid Setiap Cluster:

Kelompok	Centroid Fitur x	Centroid Fitur y	Koordinat Centroid
1	1	1,5	(1, 1.5)
2	4,2	3,6	(4.2, 3.6)
3	2	4,6667	(2, 4.6667)

Tabel tersebut menampilkan hasil perhitungan centroid setiap cluster pada Iterasi 2 metode K-Means. Centroid merepresentasikan titik pusat dari masing-masing cluster yang diperoleh dengan menghitung rata-rata nilai fitur x (Fx) dan fitur y (Fy) dari seluruh data yang menjadi anggota cluster.

Penjelasannya sebagai berikut:

- Kelompok 1 (K1) memiliki centroid dengan nilai x = 1 dan y = 1,5. Nilai ini diperoleh dari rata-rata koordinat data yang tergabung dalam cluster 1, sehingga koordinat centroidnya adalah (1, 1,5).
- Kelompok 2 (K2) memiliki centroid x = 4,2 dan y = 3,6, yang berasal dari rata-rata seluruh data pada cluster 2. Titik pusat cluster ini berada di (4,2; 3,6).

- Kelompok 3 (K3) memiliki centroid x = 2 dan y = 4,6667, yang merupakan hasil pembagian total nilai fitur dengan jumlah anggota cluster 3. Koordinat centroidnya adalah (2; 4,6667).

Centroid-centroid ini selanjutnya digunakan sebagai acuan perhitungan jarak pada iterasi berikutnya. Proses K-Means akan berhenti apabila posisi centroid sudah tidak berubah atau perubahan keanggotaannya sangat kecil, yang menandakan bahwa clustering telah mencapai kondisi konvergen.

Menghitung Jarak Data Ke Centroid - Iterasi 2

Menggunakan centroid baru: C1 (1, 1.5), C2 (4.2, 3.6), C 3 (2, 4.6667)

Data	Fx	Fy	Jarak ke C1	Jarak ke C2	Jarak ke C3	Jarak Min	K Baru	K Lama
1	1	1	**0,5000**	4,1231	3,8006	0,5000	**1**	1
2	4	1	**3,0414**	2,6077	4,1767	2,6077	**2**	2
3	6	1	**5,0249**	3,1623	5,4263	3,1623	**2**	2
4	1	2	**0,5000**	3,5777	2,8480	0,5000	**1**	1
5	2	3	**1,8028**	2,2804	1,6667	1,6667	**3**	3
6	5	3	**4,2720**	1,0000	3,4319	1,0000	**2**	2
7	2	5	**3,6401**	2,6077	0,3333	0,3333	**3**	3
8	3	5	**4,0311**	1,8439	1,0541	1,0541	**3**	2
9	2	6	**4,6098**	3,2558	1,3333	1,3333	**3**	3
10	3	8	**6,8007**	4,5607	3,4801	3,4801	**3**	2
Total						**15,6375**		

Perubahan Cluster:

Data 8 pindah dari K2 ke K3.

Data 10 pindah dari K2 ke K3.

Cek Threshold:

F lama = 19.8474 (dari Iterasi 1)

F baru = 15.6375

Delta = |19.8474 − 15.6375| = 4.21

Karena Delta > T (0.1), maka Lanjut ke Iterasi berikutnya (proses clustering belum selesai).

Berikut penjelasan perhitungan jarak data ke centroid pada Iterasi 2 metode K-Means sesuai tabel yang Anda tampilkan:

Pada iterasi ini, jarak setiap data dihitung terhadap centroid baru hasil Iterasi 2, yaitu C1 (1, 1,5), C2 (4,2; 3,6), dan C3 (2; 4,6667).
Perhitungan jarak menggunakan Euclidean Distance, yaitu akar dari selisih kuadrat koordinat data dengan koordinat centroid.

Setiap baris menunjukkan jarak suatu data ke masing-masing centroid (C1, C2, dan C3). Nilai “Jarak Min” merupakan jarak terkecil dari ketiga jarak tersebut, dan cluster dengan jarak minimum inilah yang menjadi K Baru untuk data tersebut. Kolom K Lama menunjukkan keanggotaan cluster pada iterasi sebelumnya sehingga perubahan cluster dapat diamati.

Dari hasil perhitungan, sebagian besar data tetap berada pada cluster yang sama. Namun terdapat perubahan keanggotaan, yaitu:

- Data 8 berpindah dari K2 ke K3
- Data 10 berpindah dari K2 ke K3

Nilai total fungsi objektif (F baru) pada Iterasi 2 adalah 15,6375, yang merupakan penjumlahan seluruh nilai jarak minimum. Nilai ini dibandingkan dengan F lama dari Iterasi 1 sebesar 19,8474 untuk mengevaluasi konvergensi. Selisihnya (Δ) adalah 4,21.

Karena Δ lebih besar dari nilai threshold (T = 0,1), maka perubahan masih signifikan dan proses K-Means belum konvergen, sehingga algoritma harus dilanjutkan ke iterasi berikutnya untuk memperoleh hasil clustering yang stabil.

3.1.4 Iterasi 3

iterasi 3

Menghitung Centroid Setiap Cluster (Iterasi 3)

Data	Fx	Fy	K1	K2	K3	K1 Fx	K1 Fy	K2 Fx	K2 Fy	K3 Fx	K3 Fy
1	1	1	*			1	1				
2	4	1		*				4	1		
3	6	1		*				6	1		
4	1	2	*			1	2				
5	2	3			*					2	3
6	5	3		*				5	3		
7	2	5			*					2	5
8	3	5			*					3	5
9	2	6			*					2	6
10	3	8			*					3	8
Total			2	3	5	2	3	15	5	12	27

Pada Iterasi 3, keanggotaan cluster sudah diperbarui berdasarkan hasil K Baru pada Iterasi 2, di mana terjadi perpindahan Data 8 dan Data 10 ke cluster K3. Dengan keanggotaan terbaru tersebut, dilakukan kembali perhitungan centroid untuk masing-masing cluster.

Keanggotaan cluster pada Iterasi 3 adalah:

- Cluster K1: Data 1 dan 4
- Cluster K2: Data 2, 3, dan 6
- Cluster K3: Data 5, 7, 8, 9, dan 10

Nilai Fx dan Fy dari setiap data yang masuk ke suatu cluster dijumlahkan, kemudian dicatat pada kolom Kx Fx dan Kx Fy. Hasil penjumlahan total pada baris terakhir digunakan untuk menghitung centroid baru dengan membagi total nilai fitur dengan jumlah anggota cluster.

Hasil perhitungan total tiap cluster adalah:

- K1: Total Fx = 2, Total Fy = 3, Jumlah data = 2
- K2: Total Fx = 15, Total Fy = 5, Jumlah data = 3
- K3: Total Fx = 12, Total Fy = 27, Jumlah data = 5

Sehingga diperoleh centroid baru Iterasi 3 sebagai berikut:

- Centroid K1 = (2/2, 3/2) = (1, 1,5)
- Centroid K2 = (15/3, 5/3) = (5, 1,67)
- Centroid K3 = (12/5, 27/5) = (2,4, 5,4)

Centroid hasil Iterasi 3 ini selanjutnya digunakan untuk menghitung kembali jarak setiap data pada iterasi berikutnya. Jika pada iterasi selanjutnya tidak terjadi perubahan keanggotaan cluster atau perubahan nilai fungsi objektif berada di bawah threshold, maka proses clustering dinyatakan konvergen.

Hasil Centroid Baru:

Kelompok	Centroid Fitur x	Centroid Fitur y	Koordinat Centroid
1	1	1,5	**(1, 1.5)**
2	5	1,6667	**(5, 1.6667)**
3	2,4	5,4	**(2.4, 5.4)**

Tabel Hasil Centroid Baru tersebut menunjukkan posisi centroid hasil perhitungan Iterasi 3 pada metode K-Means. Centroid merupakan titik pusat setiap cluster yang diperoleh dari rata-rata nilai fitur x (Fx) dan fitur y (Fy) seluruh data dalam cluster.

Penjelasannya sebagai berikut:

- Kelompok 1 (K1) memiliki centroid dengan Fx = 1 dan Fy = 1,5, sehingga koordinat centroidnya adalah (1, 1,5). Nilai ini sama dengan iterasi sebelumnya, menandakan bahwa posisi pusat cluster K1 sudah stabil.
- Kelompok 2 (K2) memiliki centroid Fx = 5 dan Fy = 1,6667, yang diperoleh dari rata-rata data yang tergabung dalam cluster 2. Koordinat centroidnya adalah (5, 1,6667).
- Kelompok 3 (K3) memiliki centroid Fx = 2,4 dan Fy = 5,4, hasil perhitungan dari lima data anggota cluster 3. Titik pusat cluster ini berada di (2,4; 5,4).

Centroid-centroid baru ini akan digunakan sebagai acuan perhitungan jarak pada iterasi selanjutnya. Apabila pada iterasi berikutnya tidak terjadi perubahan keanggotaan cluster atau perubahan nilai fungsi objektif berada di bawah threshold yang ditentukan, maka proses K-Means dinyatakan konvergen dan selesai.

Menghitung Jarak Data Ke Centroid - Iterasi 3

Centroid: C1 (1, 1.5), C2 (5, 1.6667), C3 (2.4, 5.4)

Data	Fx	Fy	Jarak ke C1	Jarak ke C2	Jarak ke C3	Min	K Baru	K Lama
1	1	1	**0,5000**	4,0552	4,6174	0,5000	**1**	1
2	4	1	**3,0414**	1,2019	4,6819	1,2019	**2**	2
3	6	1	**5,0249**	1,2019	5,6851	1,2019	**2**	2
4	1	2	**0,5000**	4,0139	3,6770	0,5000	**1**	1
5	2	3	**1,8028**	3,2830	2,4331	1,8028	**1**	3
6	5	3	**4,2720**	1,3333	3,5384	1,3333	**2**	2
7	2	5	**3,6401**	4,4845	0,5657	0,5657	**3**	3
8	3	5	**4,0311**	3,8873	0,7211	0,7211	**3**	3
9	2	6	**4,6098**	5,2705	0,7211	0,7211	**3**	3
10	3	8	**6,8007**	6,6416	2,6683	2,6683	**3**	3
Total						**11,2160**		

Perubahan Cluster:

Data 5 pindah dari K3 ke K1 (Jarak ke C1 1.8 lebih kecil dari jarak ke C3 2.4).

Cek Threshold:

F lama = 15.6375

F baru = 11.2160

Delta = |15.6375 - 11.2160 | = 4.4215

Karena Delta > 0.1, maka Lanjut ke Iterasi

Pada iterasi ke-3 algoritma K-Means, dilakukan perhitungan jarak setiap data terhadap centroid terbaru untuk menentukan keanggotaan cluster yang paling sesuai. Centroid yang digunakan pada iterasi ini adalah sebagai berikut:

- C1 = (1, 1,5)
- C2 = (5, 1,6667)
- C3 = (2,4, 5,4)

Perhitungan jarak dilakukan menggunakan rumus Euclidean Distance, yaitu dengan mengukur jarak terpendek antara masing-masing data dan centroid.

Hasil perhitungan jarak dan pembentukan cluster pada iterasi ke-3 dapat dijelaskan sebagai berikut:

- Setiap data dihitung jaraknya ke C1, C2, dan C3.
- Nilai jarak terkecil (minimum) dipilih sebagai acuan penentuan cluster baru.

- Sebagian besar data tetap berada pada cluster yang sama seperti pada iterasi sebelumnya.
- Data ke-5 mengalami perubahan cluster, yaitu berpindah dari cluster K3 ke cluster K1, karena jarak ke centroid C1 (1,8028) lebih kecil dibandingkan jarak ke centroid C3 (2,4331).

Untuk mengevaluasi konvergensi algoritma, dilakukan perhitungan nilai fungsi objektif (F), dengan hasil sebagai berikut:

- F lama = 15,6375
- F baru = 11,2160
- Δ (Delta) = |15,6375 − 11,2160| = 4,4215

Berdasarkan hasil tersebut, karena nilai Δ lebih besar dari batas threshold yang ditetapkan (0,1), maka proses clustering belum mencapai kondisi konvergen. Oleh sebab itu, algoritma K-Means perlu dilanjutkan ke iterasi berikutnya untuk memperoleh hasil pengelompokan yang lebih optimal.

3.1.5 Iterasi 4

Iterasi 4
Menghitung Centroid Setiap Cluster (Iterasi 4)
Anggota baru:
K1: 1, 4, 5
K2: 2, 3, 6
K3: 7, 8, 9, 10

Data	Fx	Fy	K1	K2	K3	K1 Fx	K1 Fy	K2 Fx	K2 Fy	K3 Fx	K3 Fy
1	1	1	*			1	1				
2	4	1		*				4	1		
3	6	1		*				6	1		
4	1	2	*			1	2				
5	2	3	*			2	3				
6	5	3		*				5	3		
7	2	5			*					2	5
8	3	5			*					3	5
9	2	6			*					2	6
10	3	8			*					3	8
Total			3	3	4	4	6	15	5	10	24

Pada iterasi ke-4, dilakukan perhitungan ulang centroid setiap cluster berdasarkan hasil pengelompokan data pada iterasi sebelumnya. Tujuan dari tahap ini adalah untuk memperoleh posisi centroid terbaru yang merepresentasikan rata-rata anggota pada masing-masing cluster.

Adapun anggota cluster terbaru pada iterasi ke-4 adalah sebagai berikut:

- Cluster K1: Data ke-1, 4, dan 5
- Cluster K2: Data ke-2, 3, dan 6

- Cluster K3: Data ke-7, 8, 9, dan 10

Setiap data dikelompokkan sesuai cluster-nya, kemudian nilai koordinat Fx dan Fy masing-masing data dijumlahkan untuk setiap cluster. Proses perhitungan centroid dilakukan dengan cara membagi total nilai koordinat dengan jumlah anggota cluster.

Hasil perhitungan masing-masing cluster adalah sebagai berikut:

- Cluster K1
  - Jumlah anggota = 3 data
  - Total Fx = 4
  - Total Fy = 6
  - Centroid K1 diperoleh dari rata-rata nilai Fx dan Fy anggota cluster.
- Cluster K2
  - Jumlah anggota = 3 data
  - Total Fx = 15
  - Total Fy = 5
  - Centroid K2 dihitung dari rata-rata koordinat Fx dan Fy data pada cluster K2.
- Cluster K3
  - Jumlah anggota = 4 data
  - Total Fx = 10
  - Total Fy = 24
  - Centroid K3 diperoleh dari hasil rata-rata koordinat anggota cluster K3.

Dari hasil tersebut, diperoleh centroid baru yang selanjutnya akan digunakan pada tahap berikutnya, yaitu perhitungan jarak data ke centroid pada iterasi selanjutnya. Proses ini dilakukan secara berulang hingga tidak terjadi perubahan signifikan pada nilai centroid atau keanggotaan cluster, yang menandakan algoritma telah mencapai kondisi konvergen.

Hasil Centroid Baru:

Kelompok	Centroid Fitur x	Centroid Fitur y	Koordinat Centroid
1	1,3333	2	(1.3333, 2)
2	5	1,6667	(5, 1.6667)
3	2,5	6	(2.5, 6)

Setelah dilakukan pengelompokan ulang data pada iterasi ke-4, selanjutnya dihitung centroid baru untuk masing-masing cluster. Centroid diperoleh dengan cara menghitung rata-rata nilai fitur x (Fx) dan fitur y (Fy) dari seluruh anggota dalam satu cluster. Nilai centroid ini digunakan sebagai representasi pusat data pada setiap kelompok.

Adapun hasil perhitungan centroid baru adalah sebagai berikut:

- Cluster 1 (K1)
  - Centroid fitur x = 1,3333
  - Centroid fitur y = 2
  - Koordinat centroid = (1,3333 , 2)
  - Nilai ini diperoleh dari rata-rata koordinat data yang tergabung dalam cluster K1.
- Cluster 2 (K2)
  - Centroid fitur x = 5
  - Centroid fitur y = 1,6667
  - Koordinat centroid = (5 , 1,6667)
  - Centroid ini merepresentasikan pusat sebaran data pada cluster K2.
- Cluster 3 (K3)
  - Centroid fitur x = 2,5
  - Centroid fitur y = 6
  - Koordinat centroid = (2,5 , 6)
  - Nilai centroid ini menunjukkan posisi rata-rata data yang berada pada cluster K3.

Centroid baru yang diperoleh pada iterasi ini selanjutnya digunakan untuk menghitung jarak data ke centroid pada iterasi berikutnya. Proses iterasi akan terus dilakukan hingga posisi centroid stabil dan tidak terjadi perubahan keanggotaan cluster secara signifikan.

Menghitung Jarak Data Ke Centroid - Iterasi 4

Centroid: C1 (1.3333, 2), C2 (5, 1.6667), C3 (2.5, 6)

Data	Fx	Fy	Jarak ke K1	Jarak ke K2	Jarak ke K3	Min	K Baru	K Lama
1	1	1	**1,0541**	4,0552	5,2202	1,0541	**1**	1
2	4	1	**2,8480**	1,2019	5,2202	1,2019	**2**	2
3	6	1	**4,7726**	1,2019	6,1033	1,2019	**2**	2
4	1	2	**0,3333**	4,0139	4,2720	0,3333	**1**	1
5	2	3	**1,2019**	3,2830	3,0414	1,2019	**1**	1
6	5	3	**3,8006**	1,3333	3,9051	1,3333	**2**	2
7	2	5	**3,0732**	4,4845	1,1180	1,1180	**3**	3
8	3	5	**3,4319**	3,8873	1,1180	1,1180	**3**	3
9	2	6	**4,0552**	5,2705	0,5000	0,5000	**3**	3
10	3	8	**6,2272**	6,6416	2,0616	2,0616	**3**	3
Total						**11,1239**		

Perubahan Cluster: Tidak ada data yang berpindah kelompok (K Baru = K Lama).

Cek Threshold:

F lama = 11.2160

F baru = 11.1239

Delta = |11.2160 - 11.1239 | = 0.0921

Karena Delta (0.0921) < Threshold (0.1), maka iterasi BERHENTI.

Pada iterasi ke-4, dilakukan perhitungan jarak setiap data terhadap **centroid terbaru** hasil iterasi sebelumnya. Centroid yang digunakan pada tahap ini adalah C1 (1,3333; 2), C2 (5; 1,6667), dan C3 (2,5; 6). Perhitungan jarak menggunakan rumus Euclidean Distance untuk menentukan kedekatan setiap data terhadap masing-masing centroid.

Hasil perhitungan jarak dan penentuan cluster dapat dijelaskan sebagai berikut:

- Setiap data dihitung jaraknya ke centroid K1, K2, dan K3.
- Nilai jarak minimum dipilih sebagai acuan penentuan cluster baru.
- Hasil menunjukkan bahwa seluruh data memiliki K Baru yang sama dengan K Lama, sehingga tidak terjadi perpindahan cluster pada iterasi ini.

Nilai fungsi objektif (F) pada iterasi ke-4 diperoleh dari penjumlahan seluruh jarak minimum data ke centroid terdekat, dengan hasil sebagai berikut:

- F lama = 11,2160
- F baru = 11,1239
- Δ (Delta) = |11,2160 − 11,1239| = 0,0921

Berdasarkan hasil pengecekan threshold, karena nilai Δ lebih kecil dari batas toleransi yang ditentukan (0,1), maka dapat disimpulkan bahwa algoritma K-Means telah mencapai kondisi konvergen. Oleh karena itu, proses iterasi dihentikan pada iterasi ke-4, dan hasil clustering yang diperoleh dianggap sebagai hasil akhir.

3.1.6 Kesimpulan Akhir

KESIMPULAN AKHIR

Proses clustering selesai di Iterasi 4 dengan hasil akhir sebagai berikut:

Data	Fitur x	Fitur y	K baru
1	1	1	**1**
2	4	1	**2**
3	6	1	**2**
4	1	2	**1**
5	2	3	**1**
6	5	3	**2**
7	2	5	**3**
8	3	5	**3**
9	2	6	**3**
10	3	8	**3**

Proses clustering menggunakan algoritma K-Means telah berhasil dilaksanakan dan mencapai kondisi konvergen pada iterasi ke-4. Penghentian iterasi dilakukan karena tidak terjadi lagi perubahan keanggotaan cluster dan nilai selisih fungsi objektif (Δ) telah memenuhi batas threshold yang ditentukan.

Adapun hasil akhir pengelompokan data adalah sebagai berikut:

- Cluster 1 (K1)
  - Anggota: Data 1, 4, dan 5
  - Karakteristik umum: Data dengan nilai fitur x dan fitur y relatif rendah.
- Cluster 2 (K2)
  - Anggota: Data 2, 3, dan 6
  - Karakteristik umum: Data dengan nilai fitur x sedang dan fitur y rendah hingga sedang.
- Cluster 3 (K3)
  - Anggota: Data 7, 8, 9, dan 10
  - Karakteristik umum: Data dengan nilai fitur y lebih tinggi dibandingkan cluster lainnya.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa algoritma K-Means mampu mengelompokkan data secara optimal berdasarkan kedekatan jarak terhadap centroid. Hasil clustering ini menunjukkan bahwa data dalam satu cluster memiliki karakteristik yang relatif serupa, sedangkan antar cluster memiliki perbedaan yang cukup jelas.

Hasil Terminal:

```
PS D:\Uas_Kecerdasan_Komputasi> & C:/laragon/bin/python/python-3.10/python.exe d:/Uas_Kecerdasan_Komputasi/k_means.py

=== HASIL AKHIR ===
Data 1 (np.int64(1), np.int64(1)) -> Cluster K1
Data 2 (np.int64(4), np.int64(1)) -> Cluster K2
Data 3 (np.int64(6), np.int64(1)) -> Cluster K2
Data 4 (np.int64(1), np.int64(2)) -> Cluster K1
Data 5 (np.int64(2), np.int64(3)) -> Cluster K1
Data 6 (np.int64(5), np.int64(3)) -> Cluster K2
Data 7 (np.int64(2), np.int64(5)) -> Cluster K3
Data 8 (np.int64(3), np.int64(5)) -> Cluster K3
Data 9 (np.int64(2), np.int64(6)) -> Cluster K3
Data 10 (np.int64(3), np.int64(8)) -> Cluster K3

Centroid akhir:
K1 = (1.3333, 2.0000)
K2 = (5.0000, 1.6667)
K3 = (2.5000, 6.0000)
```

3.2 Analisis Entropy dan Information Gain

3.2.1 Dataset dan Karakteristik Data

Analisis entropy dilakukan pada dataset dengan 20 sampel data yang memiliki tiga atribut prediktor dan satu atribut target:

Atribut Prediktor:

1. Wind (Angin)**:** Strong atau Weak
2. Temperature (Suhu)**:** High, Medium, atau Low
3. Weather (Cuaca)**:** Sunny, Cloudy, atau Rain

Atribut Target:

- Keputusan biner: Yes (11 data) atau No (9 data)

Distribusi Data:

Atribut	Kategori	Total	Yes	No
Wind	Strong	9	3	6
	Weak	11	8	3
Temperature	High	6	2	4
	Medium	8	4	4
	Low	6	5	1
Weather	Sunny	7	1	6
	Cloudy	7	7	0
	Rain	6	3	3

3.2.2 Perhitungan Entropy Root (Dataset Keseluruhan)

Entropy mengukur tingkat ketidakpastian dalam dataset. Untuk dataset keseluruhan dengan 11 data "Yes" dan 9 data "No":

Formula Entropy:

$$\text{Entropy}(S) = -\Sigma [p_i \times \log_2(p_i)]$$

Perhitungan:

P(Yes) = 11/20 = 0.55
P(No) = 9/20 = 0.45

$$\begin{aligned}\text{Entropy}(S) &= -(0.55 \times \log_2(0.55)) - (0.45 \times \log_2(0.45)) \\ &= -(0.55 \times -0.8625) - (0.45 \times -1.152) \\ &= 0.4744 + 0.5184 \\ &= 0.9927\end{aligned}$$

Interpretasi:

Nilai entropy 0.9927 (mendekati 1) menunjukkan bahwa dataset sangat heterogen atau tidak homogen. Ini berarti ada tingkat ketidakpastian yang tinggi dalam dataset, dan kita tidak bisa dengan mudah memprediksi kelas target tanpa informasi tambahan dari atribut-atribut prediktor.

Entropy maksimum adalah 1.0 (ketika distribusi kelas seimbang 50:50), sehingga nilai 0.9927 menunjukkan dataset hampir mencapai tingkat ketidakpastian maksimal.

3.2.3 Analisis Entropy Berdasarkan Atribut Wind

Distribusi Data Wind:

- Strong: 9 data (3 Yes, 6 No)
- Weak: 11 data (8 Yes, 3 No)

Perhitungan Entropy(Wind = Strong):

P(Yes|Strong) = 3/9 = 0.333
P(No|Strong) = 6/9 = 0.667

$$\begin{aligned}\text{Entropy}(\text{Strong}) &= -(0.333 \times \log_2(0.333)) - (0.667 \times \log_2(0.667)) \\ &= -(0.333 \times -1.585) - (0.667 \times -0.585)\end{aligned}$$

= 0.528 + 0.390
= 0.9182

Perhitungan Entropy(Wind = Weak):

P(Yes|Weak) = 8/11 = 0.727
P(No|Weak) = 3/11 = 0.273

Entropy(Weak) = $-(0.727 \times \log_2(0.727)) - (0.273 \times \log_2(0.273))$
= $-(0.727 \times -0.460) - (0.273 \times -1.873)$
= 0.334 + 0.511
= 0.8453

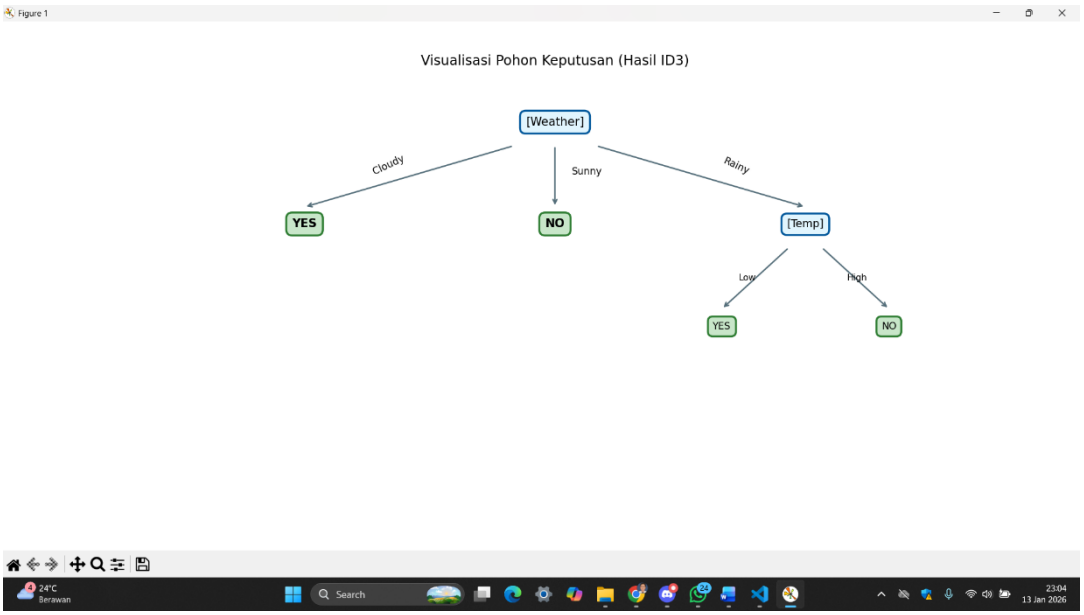
Perhitungan Information Gain(Wind):

Gain(S, Wind) = Entropy(S) - $\Sigma[(|S_v|/|S|) \times \text{Entropy}(S_v)]$
= 0.9927 - [(9/20 × 0.9182) + (11/20 × 0.8453)]
= 0.9927 - [0.4132 + 0.4649]
= 0.9927 - 0.8781
= 0.14535

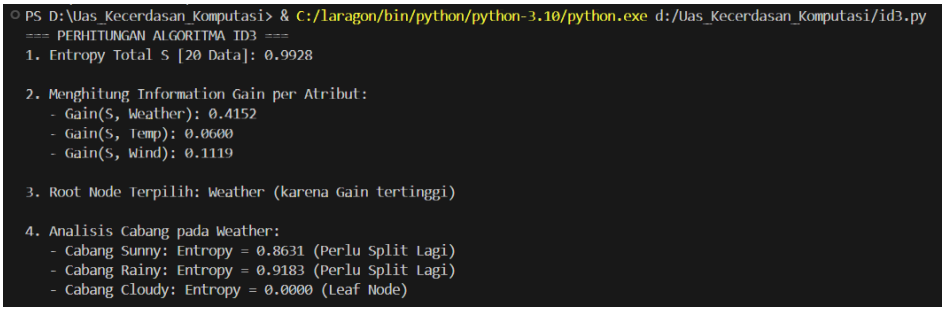
Interpretasi:

Information Gain sebesar 0.14535 menunjukkan bahwa atribut Wind memberikan pengurangan entropy (ketidakpastian).

Hasil Tampilan Pohon Keputusan:



Hasil Terminal:



BAB IV
PENUTUP

4.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian algoritma K-Means dan ID3 yang telah dilakukan, dapat disimpulkan bahwa kedua algoritma tersebut mampu diterapkan dengan baik untuk menyelesaikan permasalahan yang diberikan. Algoritma K-Means berhasil mengelompokkan data numerik ke dalam beberapa cluster berdasarkan kedekatan jarak menggunakan perhitungan Euclidean, sehingga pola data dapat terlihat secara jelas. Sementara itu, algoritma ID3 mampu membangun pohon keputusan berdasarkan nilai entropy dan information gain, sehingga atribut yang paling berpengaruh dapat ditentukan sebagai root node dalam proses pengambilan keputusan. Dengan adanya visualisasi hasil clustering dan pohon keputusan, pemahaman terhadap cara kerja kedua algoritma menjadi lebih mudah. Oleh karena itu, penerapan K-Means dan ID3 dalam program Python ini dapat membantu dalam memahami konsep dasar kecerdasan komputasi serta penerapannya pada data nyata.

DAFTAR PUSTAKA

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). San Francisco: Morgan Kaufmann.

Larose, D. T., & Larose, C. D. (2015). *Discovering Knowledge in Data: An Introduction to Data Mining* (2nd ed.). New Jersey: John Wiley & Sons.

Prasetyo, E. (2012). *Data Mining: Konsep dan Aplikasi Menggunakan MATLAB*. Yogyakarta: Andi Offset.

Santosa, B. (2007). *Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Burlington: Morgan Kaufmann.

Suyanto. (2018). *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: Informatika Bandung.

LAMPIRAN

Coding beserta Penjelasan:

1. k_means.py

```python
import numpy as np

data = np.array([
    [1, 1],
    [4, 1],
    [6, 1],
    [1, 2],
    [2, 3],
    [5, 3],
    [2, 5],
    [3, 5],
    [2, 6],
    [3, 8]
])

K = 3
threshold = 0.1

centroids = np.array([
    data[0],
    data[2],
    data[1]
], dtype=float)

def euclidean(a, b):
    return np.sqrt(np.sum((a - b) ** 2))

for iteration in range(100):
    clusters = [[] for _ in range(K)]
    labels = []

    for point in data:
        dists = [euclidean(point, c) for c in centroids]
        idx = np.argmin(dists)
        labels.append(idx)
        clusters[idx].append(point)

    new_centroids = []
    for i in range(K):
        if clusters[i]:
            new_centroids.append(np.mean(clusters[i], axis=0))
        else:
            new_centroids.append(centroids[i])

    new_centroids = np.array(new_centroids)
    delta = sum(euclidean(centroids[i], new_centroids[i]) for i in range(K))

    if delta < threshold:
        centroids = new_centroids
        break

    centroids = new_centroids

print("\n=== HASIL AKHIR ===")
for i, p in enumerate(data):
    print(f"Data {i+1} {tuple(p)} -> Cluster K{labels[i]+1}")

print("\nCentroid akhir:")
for i, c in enumerate(centroids):
    print(f"K{i+1} = ({c[0]:.4f}, {c[1]:.4f})")

```

Penjelasan:

Program ini menggunakan algoritma K-Means untuk mengelompokkan data dua dimensi. Pertama, library NumPy diimpor untuk mendukung pengolahan data numerik. Dataset kemudian didefinisikan dalam bentuk array yang berisi 10 data dengan dua atribut. Nilai K = 3 menunjukkan jumlah cluster yang akan dibentuk, sedangkan threshold digunakan sebagai batas perubahan centroid untuk menentukan kapan iterasi dihentikan. Centroid awal ditentukan secara manual dari beberapa data yang ada. Selanjutnya dibuat fungsi Euclidean untuk menghitung jarak antara data dan centroid. Proses K-Means dilakukan secara iteratif hingga maksimal 100 iterasi, di mana setiap data dihitung jaraknya ke seluruh centroid dan dimasukkan ke cluster terdekat. Setelah itu, centroid baru dihitung berdasarkan rata-rata data dalam setiap cluster. Perubahan posisi centroid dihitung untuk mengecek konvergensi, dan jika perubahan tersebut lebih kecil dari threshold, maka proses dihentikan. Terakhir, program menampilkan hasil pengelompokan setiap data ke dalam cluster serta posisi centroid akhir dari masing-masing cluster.

2. id3.py

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# 1. Dataset berdasarkan soal UAS
data = {
    'Weather': ['Sunny', 'Rainy', 'Cloudy', 'Rainy', 'Sunny', 'Rainy', 'Cloudy', 'Sunny', 'Sunny', 'Rainy',
                'Sunny', 'Cloudy', 'Cloudy', 'Rainy', 'Cloudy', 'Sunny', 'Rainy', 'Cloudy', 'Sunny', 'Cloudy'],
    'Temp': ['High', 'High', 'High', 'Medium', 'Low', 'Low', 'Low', 'Medium', 'Low', 'Medium',
             'Medium', 'Medium', 'High', 'High', 'Medium', 'Medium', 'Low', 'Low', 'High', 'Medium'],
    'Wind': ['Weak', 'Weak', 'Weak', 'Weak', 'Weak', 'Strong', 'Weak', 'Strong', 'Weak', 'Weak',
             'Weak', 'Strong', 'Weak', 'Strong', 'Weak', 'Strong', 'Strong', 'Strong', 'Weak', 'Weak'],
    'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes',
             'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes']
}

df = pd.DataFrame(data)

def calculate_entropy(data_label):
    values, counts = np.unique(data_label, return_counts=True)
    entropy = 0
    for i in range(len(values)):
        prob = counts[i] / np.sum(counts)
        entropy -= prob * np.log2(prob)
    return entropy

def calculate_gain(data, attribute, target_name):
    total_entropy = calculate_entropy(data[target_name])
    values, counts = np.unique(data[attribute], return_counts=True)

    weighted_entropy = 0
    for i in range(len(values)):
        subset = data[data[attribute] == values[i]]
        prob = counts[i] / np.sum(counts)
        weighted_entropy += prob * calculate_entropy(subset[target_name])

    gain = total_entropy - weighted_entropy
    return gain

# --- PROSES DI TERMINAL ---
print("=== PERHITUNGAN ALGORITMA ID3 ===")
entropy_total = calculate_entropy(df['Play'])
print(f"1. Entropy Total S [20 Data]: {entropy_total:.4f}")

print("\n2. Menghitung Information Gain per Atribut:")
attributes = ['Weather', 'Temp', 'Wind']
gains = {}
for attr in attributes:
    gain = calculate_gain(df, attr, 'Play')
    gains[attr] = gain
    print(f"   - Gain(S, {attr}): {gain:.4f}")

root_node = max(gains, key=gains.get)
print(f"\n3. Root Node Terpilih: {root_node} (karena Gain tertinggi)")

print("\n4. Analisis Cabang pada Weather:")
for val in df['Weather'].unique():
    subset = df[df['Weather'] == val]['Play']
    ent = calculate_entropy(subset)
    print(f"   - Cabang {val}: Entropy = {ent:.4f} ({'Leaf Node' if ent == 0 else 'Perlu Split Lagi'})")

# --- VISUALISASI MATPLOTLIB ---
def draw_tree():
    fig, ax = plt.subplots(figsize=(10, 7))
    node_style = dict(boxstyle="round,pad=0.5", fc="#e1f5fe", ec="#01579b", lw=2)
    leaf_style = dict(boxstyle="round,pad=0.5", fc="#c8e6c9", ec="#2e7d32", lw=2)
    arrow_props = dict(arrowstyle="->", lw=1.5, color="#546e7a")

    # Root
    ax.annotate(f"[{root_node}]", xy=(0.5, 0.9), ha='center', bbox=node_style, fontsize=12)

    # Cabang Cloudy (Leaf)
    ax.annotate("", xy=(0.2, 0.7), xytext=(0.45, 0.85), arrowprops=arrow_props)
    ax.text(0.28, 0.78, "Cloudy", rotation=25)
    ax.annotate("YES", xy=(0.2, 0.65), ha='center', bbox=leaf_style, fontsize=12, fontweight='bold')

    # Cabang Sunny (Split - Berdasarkan data Anda)
    ax.annotate("", xy=(0.5, 0.7), xytext=(0.5, 0.85), arrowprops=arrow_props)
    ax.text(0.52, 0.78, "Sunny")
    ax.annotate("NO", xy=(0.5, 0.65), ha='center', bbox=leaf_style, fontsize=12, fontweight='bold')

    # Cabang Rainy (Split ke Temperature)
    ax.annotate("", xy=(0.8, 0.7), xytext=(0.55, 0.85), arrowprops=arrow_props)
    ax.text(0.7, 0.78, "Rainy", rotation=-25)
    ax.annotate("[Temp]", xy=(0.8, 0.65), ha='center', bbox=node_style, fontsize=11)

    # Sub-cabang Rainy
    # Low -> Yes
    ax.annotate("", xy=(0.7, 0.45), xytext=(0.78, 0.6), arrowprops=arrow_props)
    ax.annotate("YES", xy=(0.7, 0.4), ha='center', bbox=leaf_style)
    ax.text(0.72, 0.52, "Low", fontsize=9)

    # High -> No
    ax.annotate("", xy=(0.9, 0.45), xytext=(0.82, 0.6), arrowprops=arrow_props)
    ax.annotate("NO", xy=(0.9, 0.4), ha='center', bbox=leaf_style)
    ax.text(0.85, 0.52, "High", fontsize=9)

    ax.set_axis_off()
    plt.title("Visualisasi Pohon Keputusan (Hasil ID3)", fontsize=14, pad=20)
    plt.show()

draw_tree()
```

Penjelasan:

Program ini mengimplementasikan algoritma Decision Tree ID3 untuk menentukan keputusan Play berdasarkan kondisi cuaca. Pertama, library Pandas digunakan untuk mengelola data dalam bentuk tabel, NumPy untuk perhitungan matematika seperti logaritma, dan Matplotlib untuk menampilkan visualisasi pohon keputusan. Dataset yang digunakan terdiri dari 20 data dengan atribut Weather, Temp, dan Wind sebagai variabel input, serta Play sebagai target keputusan. Data tersebut kemudian diubah menjadi DataFrame agar mudah diolah. Selanjutnya dibuat fungsi calculate_entropy untuk menghitung nilai entropy, yang berfungsi mengukur tingkat ketidakpastian data, serta fungsi calculate_gain untuk menghitung information gain setiap atribut. Program kemudian menghitung entropy total dari seluruh data dan menghitung information gain untuk setiap atribut, yaitu Weather, Temp, dan Wind. Atribut dengan nilai information gain tertinggi dipilih sebagai root node, yang pada program ini adalah atribut Weather. Setelah root node ditentukan, setiap cabang Weather dianalisis untuk melihat nilai entropy-nya guna menentukan apakah cabang tersebut sudah menjadi leaf node atau masih perlu dilakukan pemisahan lanjutan. Terakhir, struktur pohon keputusan divisualisasikan menggunakan Matplotlib dengan node keputusan dan leaf node yang ditampilkan secara jelas, sehingga hasil algoritma ID3 dapat dipahami secara visual.

KONTRIBUSI ANGGOTA KELOMPOK

1. Isra Naswa Reyka Swahili (23104410006): Mengerjakan Excel Soal No. 2.
2. Agistha Ardha Sulistyo P. (23104410009): Mengerjakan Excel Soal No. 1 bagian Iterasi 1 dan 2.
3. Zaki Zakaria Zakse (23104410010): Mengerjakan Excel Soal No. 1 bagian Iterasi 3 dan 4.
4. Jusafa Ido Ad'hareza (23104410022): Mengerjakan program Python Soal No. 2 (ID3).
5. Jovanda Kelvin Wibawa P. (23104410035): Mengerjakan program Python Soal No. 1 (K-Means).