

**PROJEK UAS DATA MINING DAN DATA WAREHOUSE  
NAIVE BAYES CLASSIFIER (GAUSSIAN NAIVE BAYES)**



**Disusun Oleh:**

1. Sholihat Briliana Marshush	2110511139	Kelas D
2. Salma Ashiila Rabbani	2110511141	Kelas A
3. Desi Ratnasari	2110511152	Kelas A
4. Zakiyya Halimatus Sa'diyah	2110511156	Kelas D
5. Amanda Najwa Perak A	2110511158	Kelas D

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA  
2022**

# PENDAHULUAN

## 1. Data Mining

Data mining adalah suatu proses pengerukan atau pengumpulan informasi penting dari suatu data yang besar. Proses data mining seringkali menggunakan metode statistika, matematika, hingga memanfaatkan teknologi artificial intelligence. Nama lainnya yaitu Knowledge discovery (mining) in databases (KDD).

### Fungsi Data Mining

#### 1. Deskriptif

Fungsi deskripsi dalam data mining adalah sebuah fungsi untuk memahami lebih jauh tentang data yang diamati. Dengan melakukan sebuah proses diharap bisa mengetahui perilaku dari sebuah data tersebut. Data tersebut itulah yang nantinya dapat digunakan untuk mengetahui karakteristik dari data yang dimaksud. Dengan menggunakan Fungsi descriptive Data mining, Maka nantinya bisa menemukan pola tertentu yang tersembunyi dalam sebuah data. Dengan kata lain jika pola yang berulang dan bernilai itulah karakteristik sebuah data bisa diketahui.

#### 2. Prediktif

Fungsi prediksi merupakan sebuah fungsi bagaimana sebuah proses nantinya akan menemukan pola tertentu dari suatu data. Pola-pola tersebut dapat diketahui dari berbagai variabel-variabel yang ada pada data. Ketika sudah menemukan pola, Maka pola yang didapat tersebut bisa digunakan untuk memprediksi variabel lain yang belum diketahui nilai ataupun jenisnya. Karena itulah fungsi satu ini dikatakan sebagai fungsi prediksi sama halnya dengan melakukan predictive analisis. Fungsi ini juga bisa digunakan untuk memprediksi sebuah variabel tertentu yang tidak ada dalam suatu data. Sehingga fungsi ini memudahkan dan menguntungkan bagi siapapun yang memerlukan prediksi yang akurat untuk membuat hal penting tersebut menjadi lebih baik.

Fungsi Data mining yang lainnya yaitu : characterization, discrimination, association, classification, clustering, outlier and trend analysis, dll.

### Metode Data Mining

#### 1. Proses pengambilan Data

Proses atau tahapan-tahapan tersebut dimulai dari data mentah dan berakhir dengan pengetahuan atau informasi yang telah diolah. Nah proses tersebut sebagai berikut :

- Data Cleansing, Proses dimana data-data yang tidak lengkap, mengandung error dan tidak konsisten dibuang dari koleksi data. Ketahui juga data lifecycle management untuk mengetahui tentang pengolahan data.

- Data Integration, Proses integrasi data dimana yang berulang akan dikombinasikan.
- Selection, Proses seleksi atau pemilihan data yang relevan terhadap analisis untuk diterima dari koleksi data yang ada.
- Data Transformation, Proses transformasi data yang sudah dipilih ke dalam bentuk mining procedure melalui cara dan agresi data.
- Data Mining, Proses yang paling penting dimana akan dilakukan berbagai teknik yang diaplikasikan untuk mengekstrak berbagai pola-pola potensial untuk mendapatkan data yang berguna.
- Pattern Evolution, Sebuah proses dimana pola-pola menarik yang sebelumnya sudah ditemukan dengan identifikasi berdasarkan measure yang telah diberikan
- Knowledge Presentation, Merupakan proses tahap terakhir, Dalam hal ini digunakan teknik visualisasi yang bertujuan membantu user dalam mengerti dan menginterpretasikan hasil dari penambangan data.

## **2. Teknik dalam Proses Penambangan Data**

- Predictive Modeling, Terdapat dua teknik yaitu Classification dan Value Prediction
- Database Segmentation, Melakukan partisi database menjadi sejumlah segmen, cluster, atau record yang sama
- Link analysis, Sebuah teknik untuk membuat hubungan antara record yang individu atau sekumpulan record dalam database.
- Deviation detection, Sebuah teknik untuk mengidentifikasi outlier yang mengekspresikan sebuah deviasi dari ekspektasi yang sudah diketahui sebelumnya.
- Nearest Neighbour, Yaitu teknik yang memprediksi pengelompokan, Teknik ini sendiri merupakan teknik yang tertua yang digunakan dalam data mining.
- Clustering, merupakan teknik untuk mengklasifikasikan data berdasarkan kriteria masing-masing data.
- Decision Tree, Merupakan teknik generasi selanjutnya, dimana teknik ini adalah sebuah model prediktif yang dapat digambarkan seperti pohon. Setiap node yang terdapat dalam struktur pohon tersebut mewakili sebuah pertanyaan yang digunakan untuk menggolongkan data.

## **2. Naive Bayes Classifier**

Naive Bayes adalah metode pengklasifikasian probabilistik sederhana yang menghitung rentang probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dataset yang ditentukan. Algoritma menggunakan teorema Bayes dan mengasumsikan semua atribut independensi atau tidak saling ketergantungan yang diberikan oleh nilai variabel kelas. Definisi lain menyatakan bahwa Naive Bayes adalah classifier yang menggunakan metode probabilistik dan statistik yang ditemukan oleh ilmuwan

Inggris Thomas Bayes, yaitu memprediksi probabilitas masa depan berdasarkan pengalaman di masa sebelumnya.

Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan Naive Bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (Training Data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Naive Bayes sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

Metode yang juga dikenal sebagai Naive Bayes Classifier ini menerapkan teknik supervised klasifikasi objek di masa depan dengan menetapkan label kelas ke instance/catatan menggunakan probabilitas bersyarat. Probabilitas bersyarat adalah ukuran peluang suatu peristiwa yang terjadi berdasarkan peristiwa lain yang telah (dengan asumsi, praduga, pernyataan, atau terbukti) terjadi. Istilah supervised merujuk pada klasifikasi training data yang sudah diberi label dengan kelas. Misalnya, sebuah transaksi penipuan telah ditandai sebagai data transaksional. Kemudian, jika ingin mengklasifikasikan transaksi di masa depan menjadi fraudulent/non-fraudulent (penipuan/non-penipuan), maka jenis klasifikasi itu akan disebut sebagai supervised. Model machine learning yang diterapkan pada program tersebut menggunakan teorema Bayes yang dirumuskan sebagai berikut:

$$P(A \mid B) = P(B \mid A)P(A)P(B)$$

Keterangan:

- $P(A \mid B)$  : Probabilitas A terjadi dengan bukti bahwa B telah terjadi (probabilitas superior)
- $P(B \mid A)$  : Probabilitas B terjadi dengan bukti bahwa A telah terjadi
- $P(A)$  : Peluang terjadinya A
- $P(B)$  : Peluang terjadinya B

### 3. Gaussian Naive Bayes

Gaussian Naive Bayes merupakan perpanjangan dari naïve Bayes. Sementara fungsi lain digunakan untuk memperkirakan distribusi data, distribusi Gaussian atau normal adalah yang paling sederhana untuk diterapkan karena Anda perlu menghitung rata-rata dan standar deviasi untuk data pelatihan.

Gaussian Naive Bayes adalah algoritma klasifikasi probabilistik berdasarkan penerapan teorema Bayes dengan asumsi independensi yang kuat. Dalam konteks klasifikasi, independensi mengacu pada gagasan bahwa keberadaan satu nilai dari suatu fitur tidak mempengaruhi keberadaan yang lain (tidak seperti independensi dalam teori probabilitas). Naif mengacu pada penggunaan asumsi bahwa fitur suatu objek tidak bergantung satu sama lain. Dalam konteks pembelajaran mesin, pengklasifikasi naif Bayes dikenal sangat ekspresif, dapat diskalakan, dan cukup

akurat, tetapi kinerjanya memburuk dengan cepat seiring pertumbuhan set pelatihan. Sejumlah fitur berkontribusi pada keberhasilan pengklasifikasi naif Bayes. Terutama, mereka tidak memerlukan penyetelan parameter model klasifikasi apapun, mereka menskalakan dengan baik dengan ukuran kumpulan data pelatihan, dan mereka dapat dengan mudah menangani fitur berkelanjutan.

## PEMBAHASAN

### CODE

```
import pandas as pd
from sklearn.model_selection import
train_test_split
from sklearn import metrics
from sklearn import preprocessing

df = pd.read_excel("mushroom_dataset.xlsx")
df
```

**Penjelasan:** Program diatas berfungsi untuk mengimport library yang dibutuhkan untuk menjalankan program naive bayes classifier yang sudah kelompok kami buat. Pada program ini membutuhkan

librari:

1. Python Data Analysis (pandas)
2. train\_test\_split (untuk memisahkan data training dengan data testing)
3. metrics (modul untuk penghitungan akurasi)
4. preprocessing (untuk mengimpor modul agar bisa melakukan preprocessing data)

### OUTPUT

Name	Type	Size	Value
clf	naive_bayes.GaussianNB	1	GaussianNB object of sklearn.naive_bayes module
df	DataFrame	(8416, 22)	Column names: cap-shape, cap-surface, cap-color, bruises, odor, gill-a ...
df1	DataFrame	(8416, 22)	Column names: cap-shape, cap-surface, cap-color, bruises, odor, gill-a ...
kriteria	list	21	['cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor', 'gill-att ...
le	preprocessing_label.LabelEncoder	1	LabelEncoder object of sklearn.preprocessing_label module
model	naive_bayes.GaussianNB	1	GaussianNB object of sklearn.naive_bayes module
x	DataFrame	(8416, 21)	Column names: cap-shape, cap-surface, cap-color, bruises, odor, gill-a ...
x_test	DataFrame	(2525, 21)	Column names: cap-shape, cap-surface, cap-color, bruises, odor, gill-a ...
x_train	DataFrame	(5891, 21)	Column names: cap-shape, cap-surface, cap-color, bruises, odor, gill-a ...
y	Series	(8416,)	Series object of pandas.core.series module
y_pred	str	6	EDIBLE
y_pred1	str	9	POISONOUS
y_pred2	str	6	EDIBLE
y_test	Series	(2525,)	Series object of pandas.core.series module
y_train	Series	(5891,)	Series object of pandas.core.series module

index	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	surface-above	surface-below	color-above	color-below	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	CONVEX	SMOOTH	WHITE	BRUISES	ALMOND	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
1	CONVEX	SMOOTH	WHITE	BRUISES	ALMOND	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
2	CONVEX	SMOOTH	WHITE	BRUISES	ALMOND	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
3	CONVEX	SMOOTH	WHITE	BRUISES	ALMOND	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
4	CONVEX	SMOOTH	WHITE	BRUISES	ALMOND	FREE	CROWDED	NARROW	BROWN	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
5	CONVEX	SMOOTH	WHITE	BRUISES	ALMOND	FREE	CROWDED	NARROW	BROWN	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
6	CONVEX	SMOOTH	WHITE	BRUISES	ANISE	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
7	CONVEX	SMOOTH	WHITE	BRUISES	ANISE	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
8	CONVEX	SMOOTH	WHITE	BRUISES	ANISE	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
9	CONVEX	SMOOTH	WHITE	BRUISES	ANISE	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
10	CONVEX	SMOOTH	WHITE	BRUISES	ANISE	FREE	CROWDED	NARROW	BROWN	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
11	CONVEX	SMOOTH	WHITE	BRUISES	ANISE	FREE	CROWDED	NARROW	BROWN	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
12	CONVEX	SMOOTH	YELLOW	BRUISES	ALMOND	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
13	CONVEX	SMOOTH	YELLOW	BRUISES	ALMOND	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
14	CONVEX	SMOOTH	YELLOW	BRUISES	ALMOND	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
15	CONVEX	SMOOTH	YELLOW	BRUISES	ALMOND	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
16	CONVEX	SMOOTH	YELLOW	BRUISES	ALMOND	FREE	CROWDED	NARROW	BROWN	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
17	CONVEX	SMOOTH	YELLOW	BRUISES	ALMOND	FREE	CROWDED	NARROW	BROWN	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
18	CONVEX	SMOOTH	YELLOW	BRUISES	ANISE	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
19	CONVEX	SMOOTH	YELLOW	BRUISES	ANISE	FREE	CROWDED	NARROW	WHITE	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS
20	CONVEX	SMOOTH	YELLOW	BRUISES	ANISE	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	PURPLE	SEVERAL	WOODS
21	CONVEX	SMOOTH	YELLOW	BRUISES	ANISE	FREE	CROWDED	NARROW	PINK	TAPERING	BULBOUS	SMOOTH	SMOOTH	WHITE	WHITE	WHITE	ONE	PENDANT	BROWN	SEVERAL	WOODS

## CODE

```
print(df.isnull().all())
```

**Penjelasan:** Kode program tersebut digunakan untuk mendeteksi adanya missing values atau tidak. Tampilan untuk output menghasilkan informasi False pada tiap kolom yang membuktikan bahwa tidak terdapat missing values pada dataset mushroom.

## OUTPUT

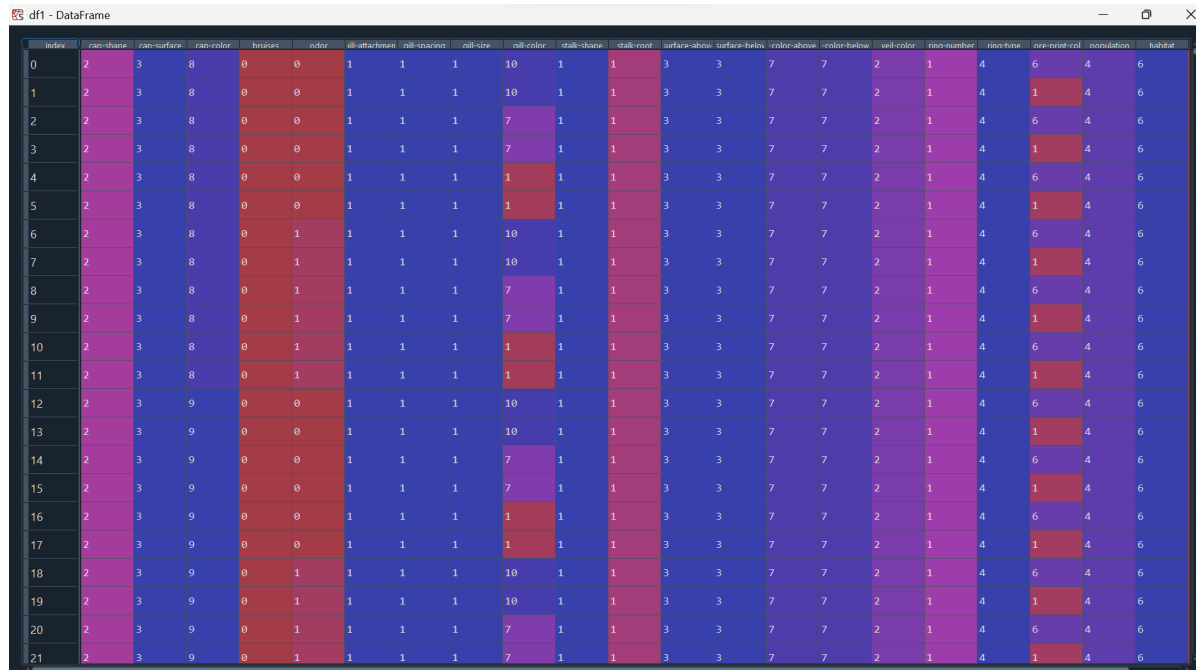
```
cap-shape           False
cap-surface         False
cap-color           False
bruises             False
odor                False
gill-attachment     False
gill-spacing        False
gill-size           False
gill-color          False
stalk-shape         False
stalk-root          False
stalk-surface-above-ring False
stalk-surface-below-ring False
stalk-color-above-ring False
stalk-color-below-ring False
veil-color          False
ring-number         False
ring-type           False
spore-print-color   False
population          False
habitat             False
mushroom            False
dtype: bool
```

## CODE

```
from sklearn.preprocessing import LabelEncoder
df1 = df.apply(LabelEncoder().fit_transform)
df1
```

**Penjelasan:** Kode program diatas digunakan untuk mengubah semua kategori nilai numerik menjadi nilai numerik spesifik.

## OUTPUT



index	ran-shape	ran-surface	ran-color	hntacs	odor	all-attachmen	rail-usares	rail-size	rail-color	stalk-shape	stalk-root	surface-above	surface-below	color-above	color-below	veil-color	ring-number	ring-shape	ring-root-col	nonulation	habitat
0	2	3	8	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
1	2	3	8	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
2	2	3	8	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
3	2	3	8	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
4	2	3	8	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	6	4	6
5	2	3	8	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	1	4	6
6	2	3	8	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
7	2	3	8	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
8	2	3	8	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
9	2	3	8	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
10	2	3	8	0	1	1	1	1	1	1	1	3	3	7	7	2	1	4	6	4	6
11	2	3	8	0	1	1	1	1	1	1	1	3	3	7	7	2	1	4	1	4	6
12	2	3	9	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
13	2	3	9	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
14	2	3	9	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
15	2	3	9	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
16	2	3	9	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	6	4	6
17	2	3	9	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	1	4	6
18	2	3	9	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
19	2	3	9	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
20	2	3	9	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
21	2	3	9	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6

## CODE

```
from sklearn.model_selection import train_test_split
x = df1.iloc[:, :21]
y = df1.iloc[:, 21]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=42)
```

**Penjelasan:** Pada program diatas adalah untuk membagi dataset menjadi dua bagian yakni bagian yang digunakan dalam data training dan testing dengan proporsi tertentu.



## OUTPUT

x - DataFrame

Index	cap-shape	cap-surface	cap-color	brnks	odor	ill-attachmen	ill-ocasion	ill-size	ill-color	stalk-shape	stalk-root	surface-above	surface-below	color-above	color-below	veil-color	ring-number	ring-shape	ore-ment-col	nonulation	habitat
0	2	3	8	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
1	2	3	8	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
2	2	3	8	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
3	2	3	8	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
4	2	3	8	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	6	4	6
5	2	3	8	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	1	4	6
6	2	3	8	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
7	2	3	8	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
8	2	3	8	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
9	2	3	8	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
10	2	3	8	0	1	1	1	1	1	1	1	3	3	7	7	2	1	4	6	4	6
11	2	3	8	0	1	1	1	1	1	1	1	3	3	7	7	2	1	4	1	4	6
12	2	3	9	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
13	2	3	9	0	0	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
14	2	3	9	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
15	2	3	9	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
16	2	3	9	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	6	4	6
17	2	3	9	0	0	1	1	1	1	1	1	3	3	7	7	2	1	4	1	4	6
18	2	3	9	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	6	4	6
19	2	3	9	0	1	1	1	1	10	1	1	3	3	7	7	2	1	4	1	4	6
20	2	3	9	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
21	2	3	9	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6

Format   Resize   Background color   Column min/max   Save and Close   Close

Tabel diatas merupakan tabel(x) yaitu untuk menyimpan dataset mushroom yang akan menjadi kriteria untuk mencari prediksi target di dalam dataset mushroom yang sudah di encoding (dari string ke data numerik) yang terdiri dari 21 kolom.

x\_test - DataFrame

Index	cap-shape	cap-surface	cap-color	brnks	odor	ill-attachmen	ill-ocasion	ill-size	ill-color	stalk-shape	stalk-root	surface-above	surface-below	color-above	color-below	veil-color	ring-number	ring-shape	ore-ment-col	nonulation	habitat
7223	3	2	0	1	8	1	0	1	2	1	0	3	2	7	5	2	1	0	7	4	1
4591	2	2	9	1	4	1	0	0	4	0	1	2	2	1	5	2	1	2	3	4	0
2522	2	0	0	0	6	1	0	0	7	1	1	3	3	5	3	2	1	4	1	5	6
6223	2	3	7	1	8	1	0	1	2	1	0	3	2	5	5	2	1	0	7	4	6
2436	2	0	7	0	6	1	0	0	1	1	1	3	3	7	3	2	1	4	0	5	6
4264	2	0	9	1	4	1	0	0	4	0	1	2	2	1	5	2	1	2	3	5	0
7897	0	3	8	1	6	1	1	0	4	0	0	3	3	7	7	2	2	4	7	2	0
2698	2	2	3	0	6	1	0	0	8	1	1	3	3	5	5	2	1	4	1	5	6
2391	2	0	7	0	6	1	0	0	7	1	1	3	3	3	3	2	1	4	1	4	6
6827	3	3	0	1	4	1	0	1	2	1	0	3	3	7	5	2	1	0	7	4	1
263	0	2	8	0	1	1	0	0	10	0	2	3	3	7	7	2	1	4	1	2	0
4397	2	0	9	1	4	1	0	0	3	0	1	2	2	5	0	2	1	2	3	5	3
1656	2	0	0	1	6	1	1	0	1	1	3	0	3	7	7	2	1	0	0	3	0
33	2	0	8	0	1	1	1	0	7	1	1	3	3	7	7	2	1	4	1	4	6
828	2	0	0	1	6	1	0	1	1	0	3	3	3	7	7	2	1	4	0	5	4
5457	2	3	8	0	4	1	0	0	7	1	1	3	3	7	7	2	1	4	3	3	0
6705	3	3	7	1	4	1	0	1	2	1	0	2	3	7	7	2	1	0	7	4	3
4778	3	0	3	1	4	1	0	0	4	0	1	2	2	5	0	2	1	2	3	4	3
7803	4	2	0	1	8	1	0	1	2	1	0	3	2	5	7	2	1	0	7	4	3
1046	3	3	8	0	7	1	0	1	0	0	3	3	3	7	7	2	1	4	1	4	4
5686	3	3	1	0	4	1	0	0	10	1	1	3	0	7	7	2	1	4	3	4	4
586	2	2	9	0	1	1	0	0	4	0	2	3	3	7	7	2	1	4	0	2	2

Format   Resize   Background color   Column min/max   Save and Close   Close

Tabel diatas merupakan tabel(x\_test) menampung data target mushroom yang akan dilatih.

x\_train - DataFrame

index	cap-shape	cap-surface	cap-color	brnks	odor	st-attachment	cell-structure	cell-size	cell-color	stalk-shape	stalk-root	surface-above	surface-below	color-above	color-below	veil-color	ring-number	ring-type	zone-root-col	nonulation	habitat
8010	2	3	3	1	6	1	1	0	10	0	0	3	2	7	7	2	2	4	7	3	0
7284	4	3	7	1	4	1	0	1	2	1	0	2	3	7	5	2	1	0	7	4	3
44	2	0	9	0	1	1	1	1	7	1	1	3	3	7	7	2	1	4	6	4	6
3411	3	0	0	0	6	1	0	0	8	1	1	3	3	7	3	2	1	4	1	4	6
7314	4	3	7	1	3	1	0	1	2	1	0	3	3	5	5	2	1	0	7	4	3
7599	4	2	7	1	3	1	0	1	2	1	0	3	3	5	7	2	1	0	7	4	3
5728	0	3	8	0	6	1	0	0	10	0	1	3	3	7	7	2	2	4	4	4	2
5364	3	2	9	1	4	1	0	0	3	0	1	2	2	5	5	2	1	2	3	5	6
3539	3	2	3	0	6	1	0	0	7	1	1	3	3	5	5	2	1	4	1	4	6
7174	3	2	0	1	3	1	0	1	2	1	0	3	2	7	7	2	1	0	7	4	6
1683	3	3	8	1	6	1	1	0	3	1	3	3	3	7	7	2	1	0	1	0	0
6449	2	2	7	1	3	1	0	1	2	1	0	3	3	5	5	2	1	0	7	4	1
3434	3	0	0	0	6	1	0	0	8	1	1	3	3	3	3	2	1	4	1	5	6
6114	2	3	7	1	4	1	0	1	2	1	0	3	3	5	5	2	1	0	7	4	3
63	3	3	9	0	0	1	1	1	7	1	1	3	3	7	7	2	1	4	1	4	6
6734	3	3	7	1	3	1	0	1	2	1	0	3	3	5	7	2	1	0	7	4	1
6926	3	3	0	1	8	1	0	1	2	1	0	3	3	5	7	2	1	0	7	4	1
3949	2	3	5	1	2	1	0	1	1	0	1	3	3	7	7	2	1	4	0	4	6
227	0	2	8	0	0	1	0	0	10	0	2	3	3	7	7	2	1	4	0	2	0
3994	2	3	3	1	2	1	1	1	8	0	1	3	3	7	7	2	1	4	1	3	6
6817	3	3	7	1	8	1	0	1	2	1	0	2	2	7	5	2	1	0	7	4	6
4329	2	0	9	1	4	1	0	0	7	0	1	2	2	1	0	2	1	2	3	4	6

Format   Resize   Background color   Column min/max   Save and Close   Close

Tabel diatas merupakan tabel(x\_train) yaitu untuk menampung semua dataset mushroom yang akan dilatih.

y - Series

index	mushroom
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0

Format   Resize   Background color   Column min/max   Save and Close   Close

Tabel diatas merupakan tabel(y) yaitu untuk menampung data target yang akan di testing, dari dataset mushroom ini, data target ada di kolom mushroom yang berisi nilai 0= edible, dan nilai 1= poisonous.

index	mushroom
7223	1
4591	1
2522	0
6223	1
2436	0
4264	1
7897	0
2698	0
2391	0
6827	1
263	0
4397	1
1656	0
33	0
828	0
5457	1
6705	1
4778	1
7803	1
1046	1
5686	1
586	0

Tabel diatas merupakan tabel(y\_test) yaitu untuk menampung dataset mushroom target yang akan digunakan untuk testing.

index	mushroom
8010	0
7284	1
44	0
3411	0
7314	1
7599	1
5728	1
5364	1
3539	0
7174	1
1683	0
6449	1
3434	0
6114	1
63	0
6734	1
6926	1
3949	1
227	0
3994	1
6817	1
4329	1

Tabel diatas merupakan tabel (y\_train) yaitu menampung semua dataset mushroom yang akan ditesting.

## CODE

```

from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model = model.fit(x_train,y_train)

print(model.predict(x_test))
print(y_test)

```

**Penjelasan:** Kode di atas berisi program yang berfungsi untuk mengimport library Gaussian Naive Bayes, data training yang nantinya akan digunakan untuk melatih algoritma dalam mencari model yang sesuai dan data testing akan dipakai untuk menguji dan mengetahui performa model yang didapatkan pada tahapan testing.

## OUTPUT

```

[1 1 0 ... 1 0 1]
7223    1
4591    1
2522    0
6223    1
2436    0
..
1891    0
2268    0
7779    1
3282    0
6855    1
Name: mushroom, Length: 2525, dtype: int32

```

## CODE

```

from sklearn.metrics import accuracy_score
y_pred = model.predict(x_test)
accuracy_score(y_test, y_pred)
print("Nilai Akurasi:", accuracy_score(y_test, y_pred))

```

**Penjelasan:** Kode diatas diawali dengan mengimportkan fungsi accuracy\_score kedalam program, kemudian menyimpan variabel 'y\_pred' yang berisikan fungsi yang dapat memberikan prediksi dari data yang ingin diketahui. Kemudian dilanjutkan mencetak nilai akurasi dari prediksi data yang ingin diketahui.

## OUTPUT

```

Nilai Akurasi: 0.8681188118811881

```

## CODE

```

from sklearn.metrics import confusion_matrix
confusion_matrix(y_pred, y_test)
print("Confusion Matriks:\n", confusion_matrix(y_pred, y_test))

```

**Penjelasan:** Kode program di atas diawali dengan mengimport fungsi confusion matrix yang berguna untuk menghitung kinerja atau tingkat kebenaran dari proses klasifikasi. Kemudian diikuti dengan

mencetak atau menampilkan confusion matriks dengan ketentuan yang diilustrasikan pada tabel berikut.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Keterangan:

- TP (True Positive) = Jumlah data aktual yang sebenarnya True diprediksi True
- TN (True Negative) = Jumlah data aktual yang sebenarnya False diprediksi False
- FP (False Positive) = Jumlah data aktual yang sebenarnya True diprediksi False
- FN (False Negative) = Jumlah data aktual yang sebenarnya False diprediksi True

## OUTPUT

```
Confusion Matriks:  
[[1176 154]  
 [ 179 1016]]
```

## CODE

```
from sklearn.metrics import classification_report  
  
print(classification_report(y_test,y_pred))
```

**Penjelasan:** Pada kode program diatas terdapat fungsi classification report yang bertujuan untuk membuat laporan teks yang menunjukkan metrik klasifikasi utama. Sehingga akan muncul hasil rata-rata Macro (hitung metrik untuk setiap label, dan temukan rata-rata tak berbobotnya. Hal ini tidak memperhitungkan ketidakseimbangan label) dan Weighted (hitung metrik untuk setiap label, dan dapat menentukan bobot rata-ratanya dengan dukungan jumlah instance sebenarnya untuk setiap label) dari metrics classification report sama seperti metrics accuracy score mereka memiliki perhitungan yang sama dari accuracy tetapi di dalam metrics classification report menghitung macro dan weightednya.

Dalam menentukan akurasi, presisi, dan recall dapat menggunakan rumus:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (1)$$

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (2)$$

$$Recall = \frac{TP}{FN+TP} * 100\% \quad (3)$$

## OUTPUT

	precision	recall	f1-score	support
0	0.88	0.87	0.88	1355
1	0.85	0.87	0.86	1170
accuracy			0.87	2525
macro avg	0.87	0.87	0.87	2525
weighted avg	0.87	0.87	0.87	2525

## CODE

```
le = preprocessing.LabelEncoder()
kriteria = ['cap-shape','cap-surface','cap-color','bruises','odor','gill-attachment',
            'gill-spacing','gill-size','gill-color','stalk-shape','stalk-root',
            'stalk-surface-above-ring','stalk-surface-below-ring','stalk-color-above-ring',
            'stalk-color-below-ring','veil-color','ring-number','ring-type','spore-print-color',
            'population','habitat']

x = df1[kriteria]
y = df1['mushroom']
clf = GaussianNB()
clf = clf.fit(x.values,y.values)
y_pred = clf.predict([[4,3,8,0,0,1,1,1,1,1,10,1,3,7,7,2,1,4,6,4,6]])
y_pred1 = clf.predict([[2,3,0,0,7,1,0,1,0,0,3,3,3,7,7,2,1,4,0,3,0]])
y_pred2 = clf.predict([[2,3,3,1,6,1,1,0,3,1,3,0,3,7,7,2,1,0,1,3,0]])

if y_pred == 0:
    y_pred = "EDIBLE"
else:
    y_pred = "POISONOUS"

if y_pred1 == 0:
    y_pred1 = "EDIBLE"
else:
    y_pred1 = "POISONOUS"

if y_pred2 == 0:
    y_pred2 = "EDIBLE"
else:
    y_pred2 = "POISONOUS"

print("Jenis Mushroom 1:",y_pred)
print("Jenis Mushroom 2:",y_pred1)
print("Jenis Mushroom 3:",y_pred2)
```

**Penjelasan:** Berikut adalah contoh perhitungan menggunakan Gaussian Naive Bayes dimana sebelumnya dilakukan proses preprocessing dengan label encoder yang dimana kriterianya diambil dari dataset mushroom yang disebutkan dalam program (didalam variabel kriteria). Dimana y\_pred, y\_pred1 dan y\_pred2 apabila y\_pred = 0 berarti menghasilkan edible, dan jika y\_pred selain 0 berarti menghasilkan poisonous. Maka dari program tersebut dapat dicetak hasil perkiraannya edible atau poisonous.

## OUTPUT

```
Jenis Mushroom 1: EDIBLE
Jenis Mushroom 2: POISONOUS
Jenis Mushroom 3: EDIBLE
```

## **PENUTUP**

### **1. Simpulan**

Data mining merupakan suatu proses pengerukan atau pengumpulan informasi penting dari suatu data yang besar. Proses data mining seringkali menggunakan metode statistika, matematika, hingga memanfaatkan teknologi artificial intelligence. Nama lainnya yaitu Knowledge discovery (mining) in databases (KDD). Naive Bayes adalah metode pengklasifikasian probabilistik sederhana yang menghitung rentang probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dataset yang ditentukan. Berdasarkan apa yang telah dilakukan kita mengimplementasikan pengklasifikasi Naive Bayes dan mencoba menyesuaikannya dengan kumpulan data klasifikasi jamur untuk memprediksi apakah jamur itu beracun atau dapat konsumsi dengan metode naive bayes.

### **2. Hasil Analisa**

Setelah melakukan analisa Data Mushroom menggunakan metode klasifikasi Naive Bayes dan algoritma Gaussian Naive Bayes kami mendapatkan nilai akurasi sebesar 0.8681188118811881. Di dalam program juga diberikan sebuah confusion matriks yang berfungsinya untuk menghitung kinerja atau tingkat kebenaran dari proses klasifikasi program yang kami jalankan. Confusion matrix memiliki empat tipe yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Maka diperoleh 1176(TP), 154(FP), 179(FN), dan 1016(TN) sebagai confusion matrix. Selain itu diketahui juga kriteria dari jamur dapat konsumsi (Edible) dan beracun (Poisonous) yang terdapat di dataset mushroom. Sehingga kami dapat melakukan percobaan program menentukan suatu jamur dapat konsumsi (Edible) atau beracun (Poisonous) dengan menggunakan metode Gaussian Naive Bayes seperti diatas. sebagai contoh diketahui sebuah jamur dengan kriteria sebagai berikut:

4(KNOBBED), 3(SMOOTH), 8(WHITE), 0(BRUISES), 0(ALMOND), 1(FREE),  
1(CROWDED), 1(NARROW), 1(PINK), 1(TAPERING), 1(BULBOUS),  
1(SMOOTH), 3(SMOOTH), 7(WHITE), 7(WHITE), 2(WHITE), 1(ONE),  
4(PENDANT), 6(PURPLE), 4(SEVERAL), 6(WOODS)

akan menghasilkan bahwa jamur tersebut dapat dikonsumsi (Edible).