

PREDIKSI PELAKU NASABAH PERBANKAN DALAM BERLANGGAN  
DEPOSITO MENGGUNAKAN MODEL MACHINE LEARNING  
K-NEAREST NEIGHBORS (KNN)

---

---

❖ *Latar Belakang*

Dalam industri perbankan, memahami perilaku nasabah sangat krusial untuk meningkatkan efektivitas kampanye pemasaran. Salah satu tantangan terbesar adalah mengetahui siapa saja yang berpotensi merespons positif terhadap penawaran produk keuangan tertentu, seperti deposito berjangka. Dengan memanfaatkan data historis yang mencakup informasi demografis nasabah, riwayat interaksi, serta hasil kampanye sebelumnya, diharapkan dapat membangun model klasifikasi yang mampu memprediksi secara akurat kemungkinan seorang nasabah untuk membeli produk deposito.

Analisis yang komprehensif ini tidak hanya akan membantu dalam memahami preferensi nasabah, tetapi juga memungkinkan bank untuk:

- **Menyesuaikan Penawaran:** Mengembangkan penawaran produk yang lebih relevan dengan kebutuhan dan keinginan nasabah.
- **Meningkatkan Hubungan:** Membangun hubungan yang lebih baik antara bank dan nasabah melalui komunikasi yang lebih terarah.
- **Mengoptimalkan Sumber Daya:** Mengalokasikan anggaran pemasaran dengan lebih efisien untuk mencapai hasil yang maksimal.

Dengan pendekatan ini, diharapkan bank dapat meningkatkan tingkat konversi dan kepuasan nasabah, serta mendorong pertumbuhan bisnis yang berkelanjutan.

❖ *Tujuan*

Membuat model machine learning yang dapat mengklasifikasikan nasabah kedalam dua kategori :

- 1 : Nasabah diprediksi akan membeli deposito berjangka

- 2 : Nasabah diprediksi tidak akan membeli deposito berjangka

#### ❖ *Dataset*

Dalam membangun model klasifikasi yang mampu memprediksi secara akurat kemungkinan seorang nasabah untuk membeli produk deposito. Dataset yang digunakan “training\_dataset.csv” yang diperoleh dari Data Science Indonesia. Dataset ini berisi informasi berbagai nasabah. Dataset ini telah diproses dan disiapkan untuk melatih model Machine Learning.

#### ❖ *Teknologi*

1. Python – Pandas, Numpy, Matplotlib, Seaborn
2. Machine Learning - Scikit-learn
3. Vscode
4. AI – Membantu memberikan Solusi

# PREDIKSI PELAKU NASABAH PERBANKAN DALAM BERLANGGAN DEPOSITO MENGGUNAKAN MODEL MACHINE LEARNING K-NEAREST NEIGHBORS (KNN)

## Data Preparation

Pada tahap awal ini, saya melakukan preparation data seperti import library – library yang digunakan dan dilanjutkan dengan membaca dataset sebagai data train dengan library yang telah diimport terlebih dahulu.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder, Normalizer, StandardScaler
from sklearn.model_selection import train_test_split, learning_curve
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.preprocessing import MinMaxScaler
from imblearn.over_sampling import SMOTE
```

Pada tahap preparation data ini, saya harus mengenal apa saja kolom-kolom yang terdapat pada dataset beserta dengan type data setiap kolomnya ini agar saya lebih memudahkan dalam menganalisisnya. Diketahui bahwa jumlah baris pada dataset ini berjumlah *22916 baris dan 22 kolom*.

```
data_train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22916 entries, 0 to 22915
Data columns (total 22 columns):
 #   Column                                Non-Null Count  Dtype  
---  -
 0   customer_number                      22916 non-null  int64  
 1   usia                                22916 non-null  int64  
 2   pekerjaan                           22916 non-null  object  
 3   status_perkawinan                   22916 non-null  object  
 4   pendidikan                          22916 non-null  object  
 5   gagal_bayar_sebelumnya              22916 non-null  object  
 6   pinjaman_rumah                     22916 non-null  object  
 7   pinjaman_pribadi                   22916 non-null  object  
 8   jenis_kontak                        22916 non-null  object  
 9   bulan_kontak_terakhir               22916 non-null  object  
10  hari_kontak_terakhir                22916 non-null  object  
11  jumlah_kontak_kampanye_ini          22916 non-null  int64  
12  hari_sejak_kontak_sebelumnya        22916 non-null  int64  
13  jumlah_kontak_sebelumnya            22916 non-null  int64  
14  hasil_kampanye_sebelumnya           22916 non-null  object  
15  tingkat_variasi_pekerjaan           22916 non-null  float64  
16  indeks_harga_konsumen               22916 non-null  float64  
17  indeks_kepercayaan_konsumen          22916 non-null  float64  
18  suku_bunga_euribor_3bln            22916 non-null  float64  
19  jumlah_pekerja                      22916 non-null  float64  
20  pulau                               22916 non-null  object  
21  berlangganan_deposito               22916 non-null  int64  
dtypes: float64(5), int64(6), object(11)
memory usage: 3.8+ MB
```

## Data Pre-processing

Pada tahap pre-processing ini, ini merupakan Langkah penting sebelum data digunakan untuk analisis model. Pada tahap ini saya melakukan pengecekan data duplikat dan dilanjutkan pengecekan data missing values.

```
# Mengidentifikasi missing values
data_train.isnull().sum()

customer_number      0
usia                 0
pekerjaan            0
status_perkawinan    0
pendidikan           0
gagal_bayar_sebelumnya 0
pinjaman_rumah      0
pinjaman_pribadi     0
jenis_kontak         0
bulan_kontak_terakhir 0
hari_kontak_terakhir 0
jumlah_kontak_kampanye_ini 0
hari_sejak_kontak_sebelumnya 0
jumlah_kontak_sebelumnya 0
hasil_kampanye_sebelumnya 0
tingkat_variasi_pekerjaan 0
indeks_harga_konsumen 0
indeks_kepercayaan_konsumen 0
suku_bunga_euribor_3bln 0
jumlah_pekerja       0
pulau               0
berlangganan_deposito 0
dtype: int64
```

```
# Mengidentifikasi data duplikat
data_train.duplicated()

0      False
1      False
2      False
3      False
4      False
...
22911   False
22912   False
22913   False
22914   False
22915   False
Length: 22916, dtype: bool
```

Diatas dapat diketahui, tidak ada data duplikat dan missing values pada dataset. Ini membuktikan data clean dan siap dianalisis selanjutnya.

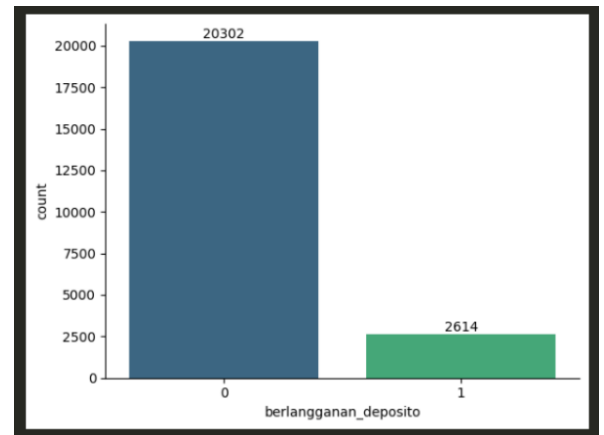
## Exploratory Data Analysis

Pada proses exploratory data analysis ini bertujuan untuk memahami karakteristik dan pola dalam data sebelum melakukan modelling.

**Visualisasi Kategori Berlangganan Deposito** - Pada tahap ini saya akan menganalisis kategori orang yang berlangganan deposito. Pada grafik disamping dapat diketahui bahwa pada dataset tersebut lebih didominasi oleh orang yang tidak membeli deposito dengan angka 20302 dan orang yang membeli deposito sebanyak 2614 orang.

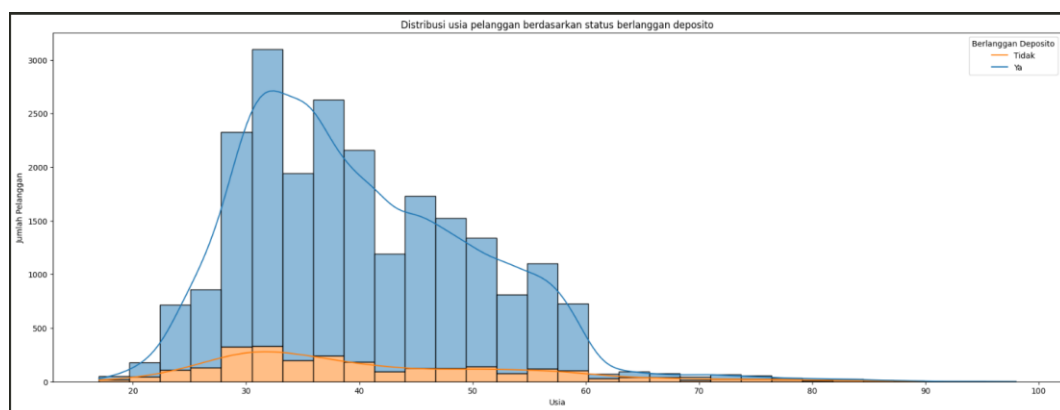
```
ax = sns.countplot(data_train, x='berlangganan_deposito', palette='viridis')
for i in ax.containers:
    ax.bar_label(i)

sns.despine()
```

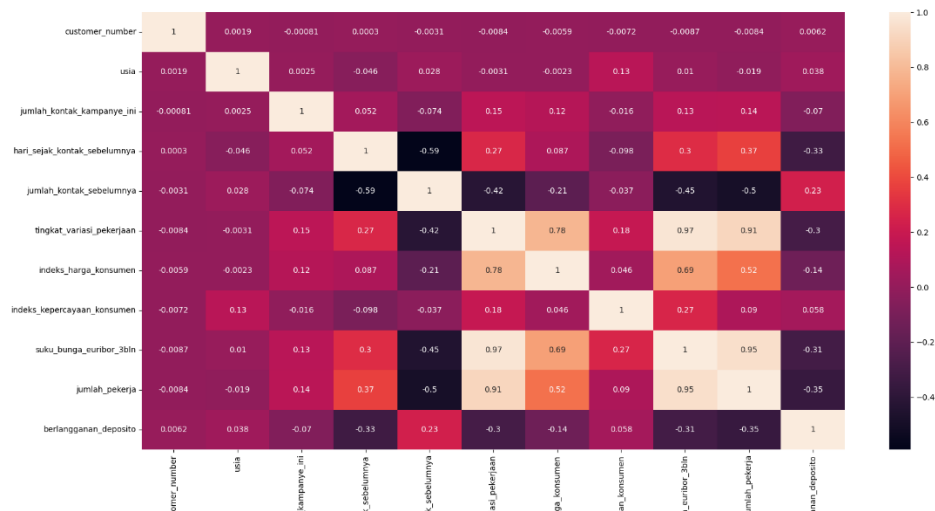


**Visualisasi Distribusi Usia** – Saya melakukan visualisasi dengan melihat histogram pada kolom usia, diketahui antara usia 25 sampai 40 jauh lebih mendominasi untuk membeli deposito. Hal ini menandakan usia tersebut lebih mendominasi.

```
plt.figure(figsize=(23,8))
sns.histplot(data_train, x='usia', hue='berlangganan_deposito', multiple='stack', bins=30, kde=True, color='viridis')
plt.title('Distribusi usia pelanggan berdasarkan status berlangganan deposito')
plt.xlabel('Usia')
plt.ylabel('Jumlah Pelanggan')
plt.legend(title="Berlanggan Deposito", labels=['Tidak', 'Ya'])
plt.show()
```



**Corelation Heatmap** - Pada tahap ini saya mengidentifikasi kolom-kolom apa saja yang memiliki keterhubungan yang kuat. Hal ini bisa menjadi bahan pertimbangan untuk kolom-kolom apa saja yang bisa dijadikan kolom-kolom penting untuk dianalisis.



## Label Encoder

Pada step ini merupakan step untuk pemrosesan data yang mana bertujuan untuk mengubah kolom yang bertipe object menjadi kolom bertipe numerik. Hal ini dilakukan agar machine learning dapat belajar dengan input data bertipe numerik, machine learning tidak dapat belajar dengan kolom bertipe data object. Label encoder merupakan cara yang efektif untuk mempersiapkan data kategorikal sebelum pelatihan model. Berikut kodenya dibawah :

```

encoder = LabelEncoder()

data_kategorikal = data_train.select_dtypes(include='object').columns

for kal in data_kategorikal :
    data_train[kal] = data_train[kal].astype(str) # merubah semua nilai menjadi type string
    data_train[kal] = encoder.fit_transform(data_train[kal])

```

✓ 0.0s

## ***Splitting Data***

Splitting data merupakan proses membagi dataset menjadi beberapa bagian untuk tujuan analisis atau pelatihan model. Dalam hal ini dataset akan dipisahkan menjadi dua yaitu data training dan data uji.

```
# Proses membagi data menjadi dua fitur (X dan y)
X = data_train.drop(columns=['berlangganan_deposito'], axis=1)
y = data_train['berlangganan_deposito']
✓ 0.0s
```

```
# Membagi data latih 80% dan data uji 20%
X_train, X_test, y_train, y_test = train_test_split(X_new, y_new, test_size=0.2, random_state=69)
✓ 0.0s
```

## ***Train Model***

Pada tahap ini model akan dilatih dengan data train, si model akan belajar dari data untuk mengidentifikasi pola dan hubungan. Pada hal ini saya menggunakan model K-NEAREST NEIGHBORS (KNN) untuk si modelnya si mesin akan belajar berdasarkan data train yang diberikan.

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
✓ 0.1s
```

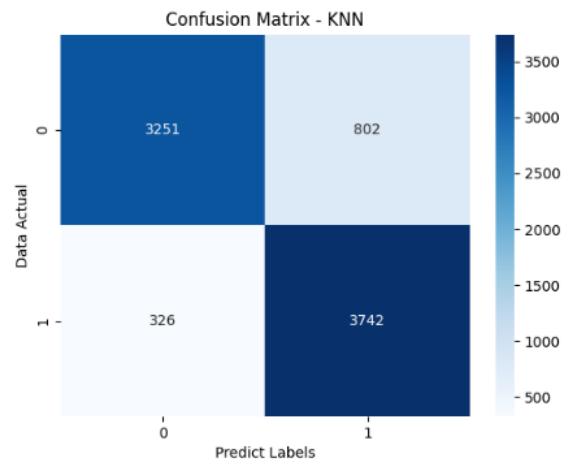
KNeighborsClassifier

KNeighborsClassifier(n\_neighbors=3)

Classification Report =				precision	recall	f1-score	support
	0	0.91	0.80	0.85			4053
	1	0.82	0.92	0.87			4068
accuracy				0.86			8121
macro avg				0.87	0.86	0.86	8121
weighted avg				0.87	0.86	0.86	8121

Dengan model K-Nearest Neighbors berhasil belajar dengan data train dan mendapatkan skor sebesar **0.86%** hal ini dapat dikatakan baik.

Visualisasi confusion matriks disamping merupakan evaluasi kinerja model klasifikasi dalam machine learning. Pada confusion matriks jumlah prediksi yang benar dan salah yang dilakukan model. Model berhasil memprediksi True Positif 3251 dikelas 0 dan True Positif 3742 dikelas 1



## **Evaluation Model**

Pada proses evaluasi model ini, memastikan model dapat melakukan prediksi dengan baik pada data. Evaluasi model merupakan proses untuk menilai kinerja model setelah model dilatih. Melakukan evaluasi model membantu mengidentifikasi kekuatan dan kelemahan model, serta memberikan wawasan untuk perbaikan di masa depan.

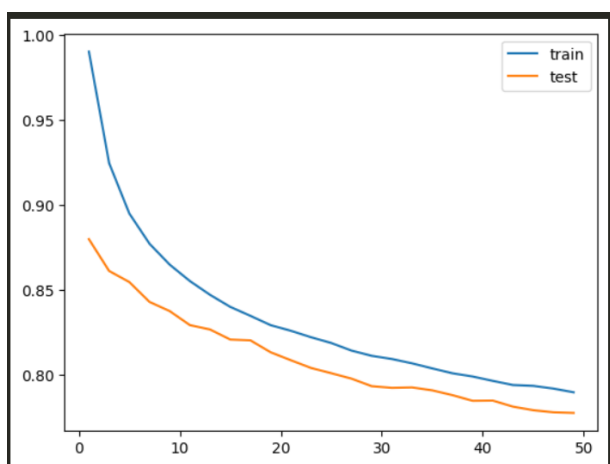
**Feature Scaling – Min Max Scaler**, Proses ini dilakukan untuk menskalakan fitur dalam dataset ke dalam rentang tertentu seperti 0 dan 1.

```
scaler = MinMaxScaler()
scaler.fit(X_train)
```

✓ 0.0s

MinMaxScaler ⓘ ?

MinMaxScaler()



**Tuning Model** - Tuning model adalah proses untuk mengoptimalkan hyperparameter dari model machine learning agar dapat mencapai kinerja terbaik. Hyperparameter adalah parameter yang tidak dipelajari secara otomatis selama pelatihan dan perlu diatur sebelum pelatihan dimulai.



## Test Model

Tahap ini saya melakukan test model dengan dataset yang disediakan dengan catatan kolom targetnya telah dihapus. Pada dataset test model ini terdapat 5729 baris dan 22 kolom. Test model bertujuan untuk roses pengembangan machine learning untuk mengevaluasi kinerjanya pada data yang belum pernah dilihat sebelumnya.

Sama seperti dataset train sebelum type data disetiap kolom sama hal ini bisa menandakan proses analisis dapat dilanjutkan.

```
pred_test = knn.predict(data_test)
pred_test

✓ 1.0s

c:\Users\Deon Harefa\AppData\Local\Programs\Python\Python312\
warnings.warn(

array([0, 0, 0, ..., 0, 0, 0], shape=(5729,))
```

Diatas merupakan proses prediksi data dengan model machine learning yang telah dilatih sebelumnya dengan dataset uji.

```
df.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5729 entries, 0 to 5728
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype
---  -
0   customer_number     5729 non-null  int64
1   usia                5729 non-null  int64
2   pekerjaan           5729 non-null  object
3   status_perkawinan   5729 non-null  object
4   pendidikan          5729 non-null  object
5   gagal_bayar_sebelumnya 5729 non-null  object
6   pinjaman_rumah     5729 non-null  object
7   pinjaman_pribadi    5729 non-null  object
8   jenis_kontak        5729 non-null  object
9   bulan_kontak_terakhir 5729 non-null  object
10  hari_kontak_terakhir 5729 non-null  object
11  jumlah_kontak_kampanye_ini 5729 non-null  int64
12  hari_sejak_kontak_sebelumnya 5729 non-null  int64
13  jumlah_kontak_sebelumnya 5729 non-null  int64
14  hasil_kampanye_sebelumnya 5729 non-null  object
15  tingkat_variasi_pekerjaan 5729 non-null  float64
16  indeks_harga_konsumen 5729 non-null  float64
17  indeks_kepercayaan_konsumen 5729 non-null  float64
18  suku_bunga_euribor_3bln 5729 non-null  float64
19  jumlah_pekerja      5729 non-null  float64
20  pulau              5729 non-null  object
dtypes: float64(5), int64(5), object(11)
memory usage: 948.0+ KB
```

Dapat dilihat grafik batang diatas, berdasarakan data uji model berhasil melakukan prediksi kategori 0 dengan jumlah 4433 orang yang tidak membeli deposito dan 1296 orang yang membeli depoito.

