



# Keputusan Kredit Thera Bank dengan Metode ANN

Kelompok 3 RA

# ANGGOTA



**Marhanny Zahra N**  
**120450017**



**Fadillah Muhesa L.**  
**120450003**



**Dimas Wahyu S**  
**120450081**



**Bane Rael Sharin**  
**120450101**



**Lisnurani**  
**120450055**

# OUTLINE

**01**

**Pendahuluan**

**02**

**Metode**

**03**

**Hasil dan  
Pembahasan**

**04**

**Kesimpulan**



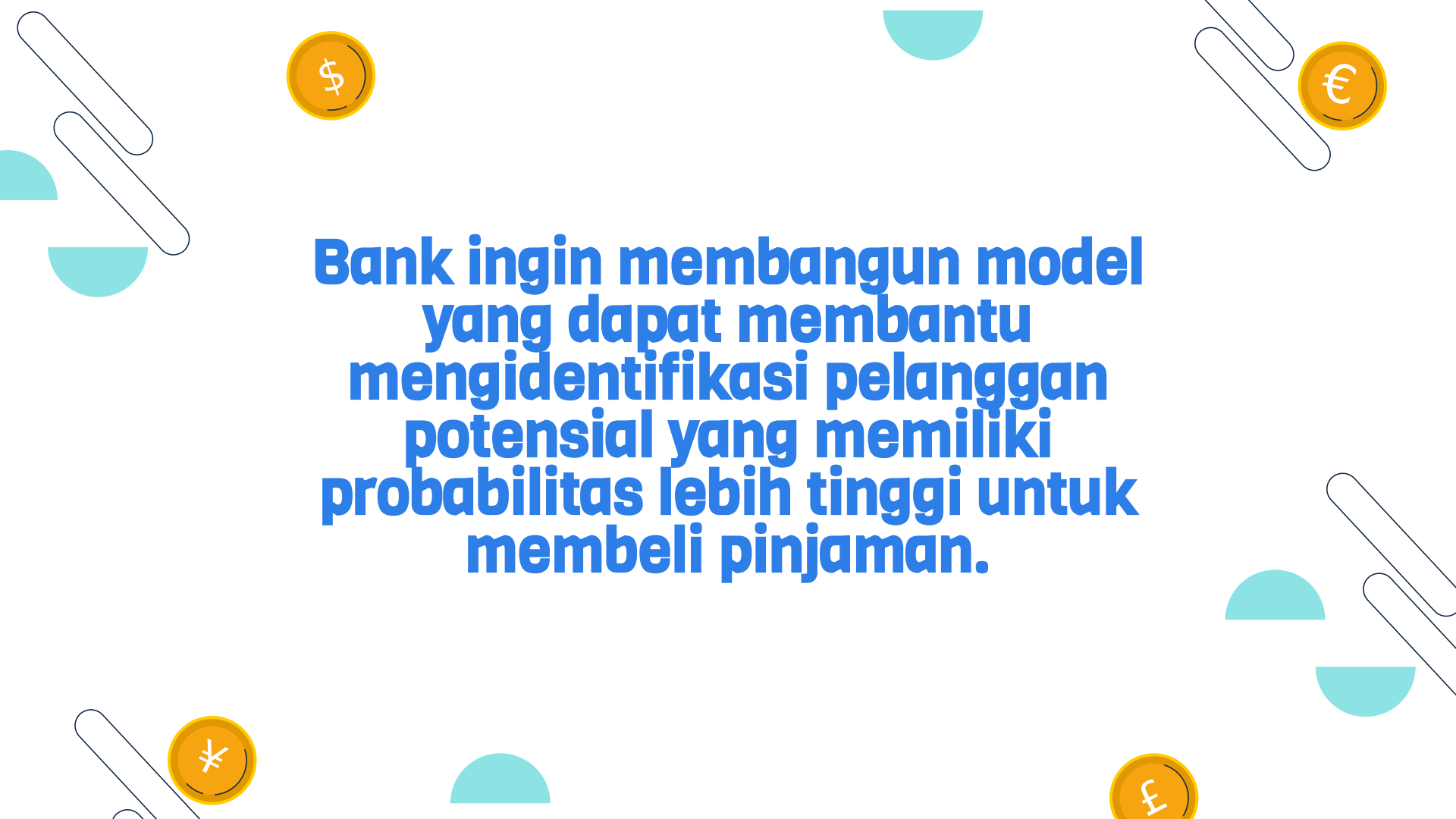
01

# Pendahuluan



# Pendahuluan



The background is white and decorated with several elements: two paper clips in the top-left and top-right corners, two gold coins with currency symbols (\$, €, ¥, £) in the corners, and several light blue semi-circles scattered throughout.

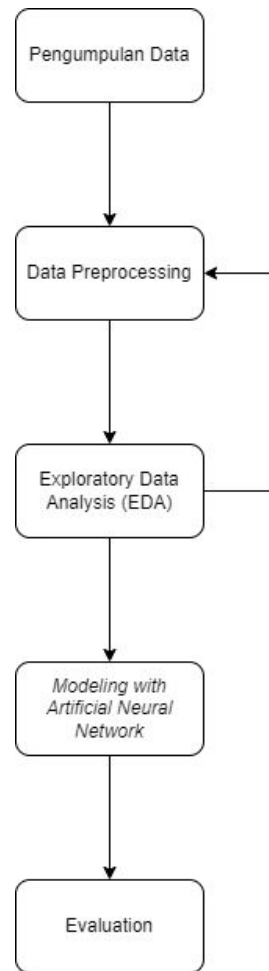
**Bank ingin membangun model  
yang dapat membantu  
mengidentifikasi pelanggan  
potensial yang memiliki  
probabilitas lebih tinggi untuk  
membeli pinjaman.**



02

**Metode**

# Flowchart Metode





# Pengumpulan Data

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	4996	29	3	40	92697	1	1.9	3	0	0	0	0	1	0
4996	4997	30	4	15	92037	4	0.4	1	85	0	0	0	1	0
4997	4998	63	39	24	93023	2	0.3	3	0	0	0	0	0	0
4998	4999	65	40	49	90034	3	0.5	2	0	0	0	0	1	0
4999	5000	28	4	83	92612	3	0.8	1	0	0	0	0	1	1

5000 rows × 14 columns

Sumber: <https://www.kaggle.com/code/dikshasingh19/bank-loan-model-using-decisiontreeclassifier/notebook>

Atribut	Keterangan
ID	ID Pelanggan
Age	Umur Pelanggan
Experience	Jumlah pengalaman kerja dalam beberapa tahun
Income	Jumlah pendapatan tahunan (dalam ribuan)
ZIP Code	Kode pos tempat tinggal pelanggan
Family	Jumlah anggota keluarga
CCAvg	Rata-rata pengeluaran kartu kredit bulanan
Education	Tingkat pendidikan (1: Sarjana, 2: Master, 3: Gelar Lanjutan)
Mortgage	Hipotek rumah (dalam ribuan)
Personal Loan	Ini adalah variabel target (Masalah Klasifikasi Biner)
Securities Account	Apakah nasabah memiliki efek rekening
CD Account	Boolean apakah nasabah memiliki rekening Sertifikat Deposito
Online	Boolean apakah pelanggan menggunakan perbankan online
Credit Card	Apakah nasabah menggunakan kartu kredit yang dikeluarkan oleh bank?

# Data Preprocessing

Proses pengklasifikasian data yang diperlukan untuk membersihkan, menghilangkan, dan atau mengubah bentuk data

Bertujuan untuk meningkatkan kualitas pada data sebelum dilakukan pemodelan agar didapatkan model yang memiliki performa yang baik.



Beberapa data pada tabel yang akan diuji harus ditransformasikan atau diinisiasi menjadi nilai tertentu

Beberapa preprocessing yang dilakukan yaitu membersihkan dan memperbaiki data yang rusak, menghapus data yang tidak penting, serta menggabungkan beberapa data yang sejenis menjadi satu atribut.

# Exploratory Data Analysis (EDA)

Tujuan utama dari langkah ini adalah guna memeriksa data untuk distribusi, outlier, dan lainnya untuk mengarahkan pengujian spesifik dari hipotesis yang dimiliki. Langkah ini juga berguna untuk menjadi alat pembuatan hipotesis melalui visualisasi data sehingga membantu memahami data melalui representasi grafis yang ditampilkan.

# Exploratory Data Analysis (EDA)

```
bank_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype  
---  --
 0   ID                   5000 non-null   int64  
 1   Age                  5000 non-null   int64  
 2   Experience            5000 non-null   int64  
 3   Income                5000 non-null   int64  
 4   ZIP Code              5000 non-null   int64  
 5   Family                5000 non-null   int64  
 6   CCAvg                 5000 non-null   float64
 7   Education             5000 non-null   int64  
 8   Mortgage              5000 non-null   int64  
 9   Personal Loan         5000 non-null   int64  
10   Securities Account    5000 non-null   int64  
11   CD Account            5000 non-null   int64  
12   Online                5000 non-null   int64  
13   CreditCard            5000 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
```

## Informasi data

Berisikan informasi detail mengenai dataframe seperti jumlah baris, fitur, tipe data, dsb.

```
bank_df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
ID	5000.0	2500.500000	1443.520003	1.0	1250.75	2500.5	3750.25	5000.0
Age	5000.0	45.338400	11.463166	23.0	35.00	45.0	55.00	67.0
Experience	5000.0	20.104600	11.467954	-3.0	10.00	20.0	30.00	43.0
Income	5000.0	73.774200	46.033729	8.0	39.00	64.0	98.00	224.0
ZIP Code	5000.0	93152.503000	2121.852197	9307.0	91911.00	93437.0	94608.00	96651.0
Family	5000.0	2.396400	1.147663	1.0	1.00	2.0	3.00	4.0
CCAvg	5000.0	1.937938	1.747659	0.0	0.70	1.5	2.50	10.0
Education	5000.0	1.881000	0.839869	1.0	1.00	2.0	3.00	3.0
Mortgage	5000.0	56.498800	101.713802	0.0	0.00	0.0	101.00	635.0
Personal Loan	5000.0	0.096000	0.294621	0.0	0.00	0.0	0.00	1.0
Securities Account	5000.0	0.104400	0.305809	0.0	0.00	0.0	0.00	1.0
CD Account	5000.0	0.060400	0.238250	0.0	0.00	0.0	0.00	1.0
Online	5000.0	0.596800	0.490589	0.0	0.00	1.0	1.00	1.0
CreditCard	5000.0	0.294000	0.455637	0.0	0.00	0.0	1.00	1.0

## Ringkasan

Berisikan ringkasan deskriptif dataframe

# Exploratory Data Analysis (EDA)

```
bank_df.isnull().sum()
```

ID	0
Age	0
Experience	0
Income	0
ZIP Code	0
Family	0
CCAvg	0
Education	0
Mortgage	0
Personal Loan	0
Securities Account	0
CD Account	0
Online	0
CreditCard	0
dtype: int64	

## Deteksi data NULL

Memeriksa apakah terdapat nilai yang hilang/tidak diketahui pada kolom. Pada penelitian ini tidak terdeteksi ada nilai null.

```
avg_age = bank_df["Age"].mean()  
print ("The average age of this dataset is {:.1f}.".format(avg_age))
```

The average age of this dataset is 45.3.

## Rata-rata umur customer

Hasil analisis rata-rata umur customer adalah sekitar 45,3

```
percent_cc = sum(bank_df["CreditCard"] == 1)/len(bank_df)  
print ("The percentage of customers that own the bank's credit card is {:.2%}.".format(percent_cc))
```

The percentage of customers that own the bank's credit card is 29.40%.

## Customer yang memiliki kartu kredit

Hasil analisis, didapatkannya persentase dari customer yang memiliki kartu kredit sebesar 29,4%

# Exploratory Data Analysis (EDA)

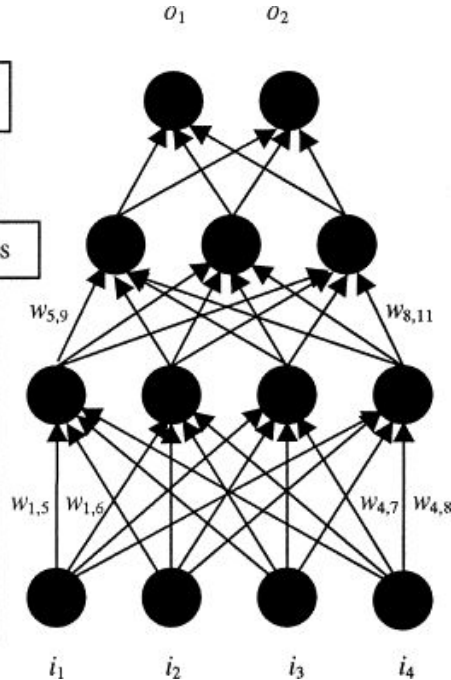
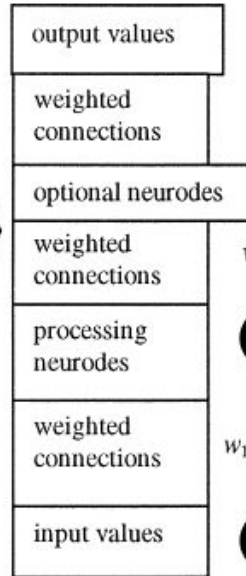
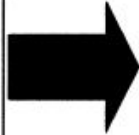
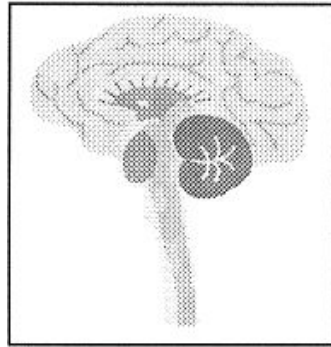
```
percent_loan = sum(bank_df["Personal Loan"] == 1)/len(bank_df)
print ("The percentage of customers that took out a personal loan is {:.2%}.".format(percent_loan))
```

The percentage of customers that took out a personal loan is 9.60%.

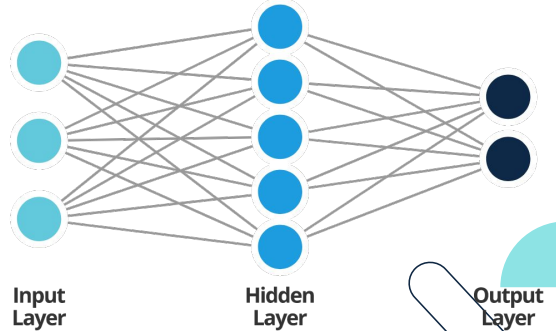
## Customer yang mengambil pinjaman pribadi

Berdasarkan hasil analisis, didapatkan hasil persentase customer yang mengambil pinjaman pribadi mencapai 9,6%

# Artificial Neural Network (ANN)



## Artificial Neural Network Architecture





03

## Hasil dan Pembahasan



```
[ ] 1 bank_df.head()
```

01

## Cek Data

```
[ ] 1 bank_df.shape
```

```
(5000, 14)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
#   Column             Non-Null Count  Dtype  
---  -
0   ID                  5000 non-null  int64  
1   Age                 5000 non-null  int64  
2   Experience           5000 non-null  int64  
3   Income              5000 non-null  int64  
4   ZIP Code            5000 non-null  int64  
5   Family              5000 non-null  int64  
6   CCAvg               5000 non-null  float64 
7   Education            5000 non-null  int64  
8   Mortgage            5000 non-null  int64  
9   Personal Loan       5000 non-null  int64  
10  Securities Account  5000 non-null  int64  
11  CD Account          5000 non-null  int64  
12  Online              5000 non-null  int64  
13  CreditCard          5000 non-null  int64  
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
```

02

## Cek Dimensi dan Info Data

```
1 bank_df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
ID	5000.0	2500.500000	1443.520003	1.0	1250.75	2500.5	3750.25	5000.0
Age	5000.0	45.338400	11.463166	23.0	35.00	45.0	55.00	67.0
Experience	5000.0	20.104600	11.467954	-3.0	10.00	20.0	30.00	43.0
Income	5000.0	73.774200	46.033729	8.0	39.00	64.0	98.00	224.0
ZIP Code	5000.0	93152.503000	2121.852197	9307.0	91911.00	93437.0	94608.00	96651.0
Family	5000.0	2.396400	1.147663	1.0	1.00	2.0	3.00	4.0
CCAvg	5000.0	1.937938	1.747659	0.0	0.70	1.5	2.50	10.0

```
1 bank_df.isnull().sum()
```

ID	0	321	0.0	0.00	0.0	0.00	1.0
Age	0	309	0.0	0.00	0.0	0.00	0.00
Experience	0	309	0.0	0.00	0.0	0.00	0.00
Income	0	250	0.0	0.00	0.0	0.00	0.00
ZIP Code	0	589	0.0	0.00	0.0	1.00	0.00
Family	0	337	0.0	0.00	0.0	0.00	1.0
CCAvg	0	337	0.0	0.00	0.0	0.00	1.0
Education	0	337	0.0	0.00	0.0	0.00	1.0
Mortgage	0	337	0.0	0.00	0.0	0.00	1.0
Personal Loan	0	337	0.0	0.00	0.0	0.00	1.0
Securities Account	0	337	0.0	0.00	0.0	0.00	1.0
CD Account	0	337	0.0	0.00	0.0	0.00	1.0
Online	0	337	0.0	0.00	0.0	0.00	1.0
CreditCard	0	337	0.0	0.00	0.0	0.00	1.0
dtype: int64							

03

## Statistika Deskriptif dan Cek Null

```
[ ] 1 avg_age = bank_df["Age"].mean()
    2 print ("The average age of this dataset is {:.1f}.".format(avg_age))
```

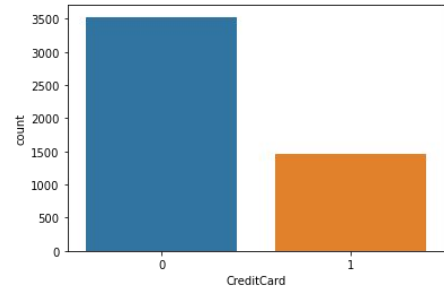
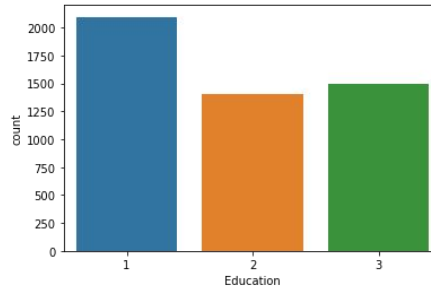
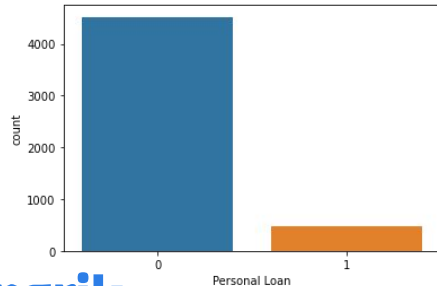
The average age of this dataset is 45.3.

```
[ ] 1 percent_cc = sum(bank_df["CreditCard"] == 1)/len(bank_df)
    2 print ("The percentage of customers that own the bank's credit card is {:.2%}.".format(percent_cc))
```

The percentage of customers that own the bank's credit card is 29.40%.

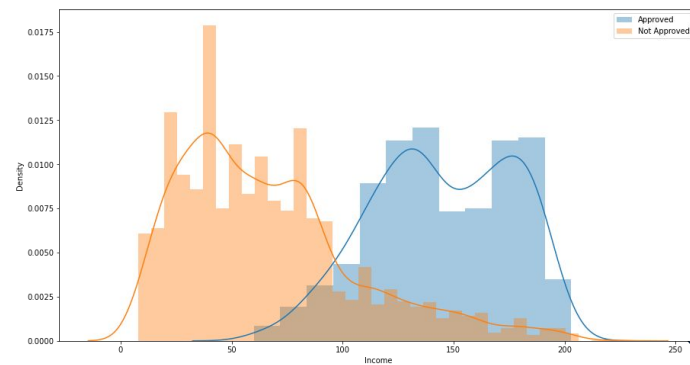
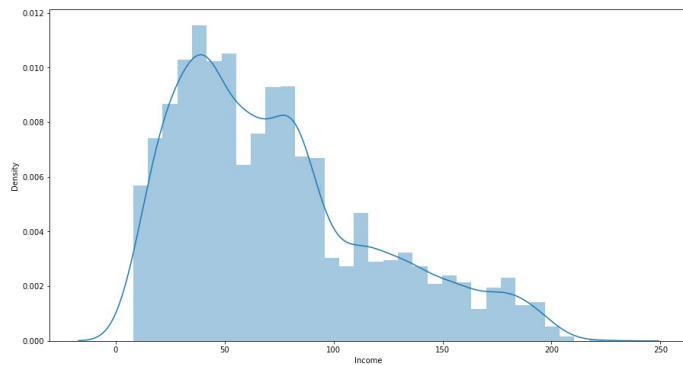
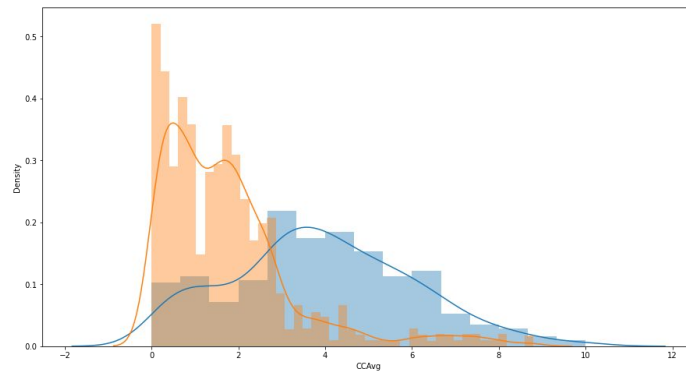
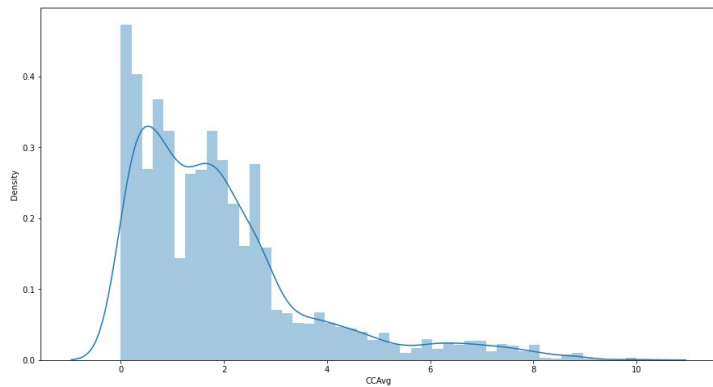
```
[ ] 1 percent_loan = sum(bank_df["Personal Loan"] == 1)/len(bank_df)
    2 print ("The percentage of customers that took out a personal loan is {:.2%}.".format(percent_loan))
```

The percentage of customers that took out a personal loan is 9.60%.



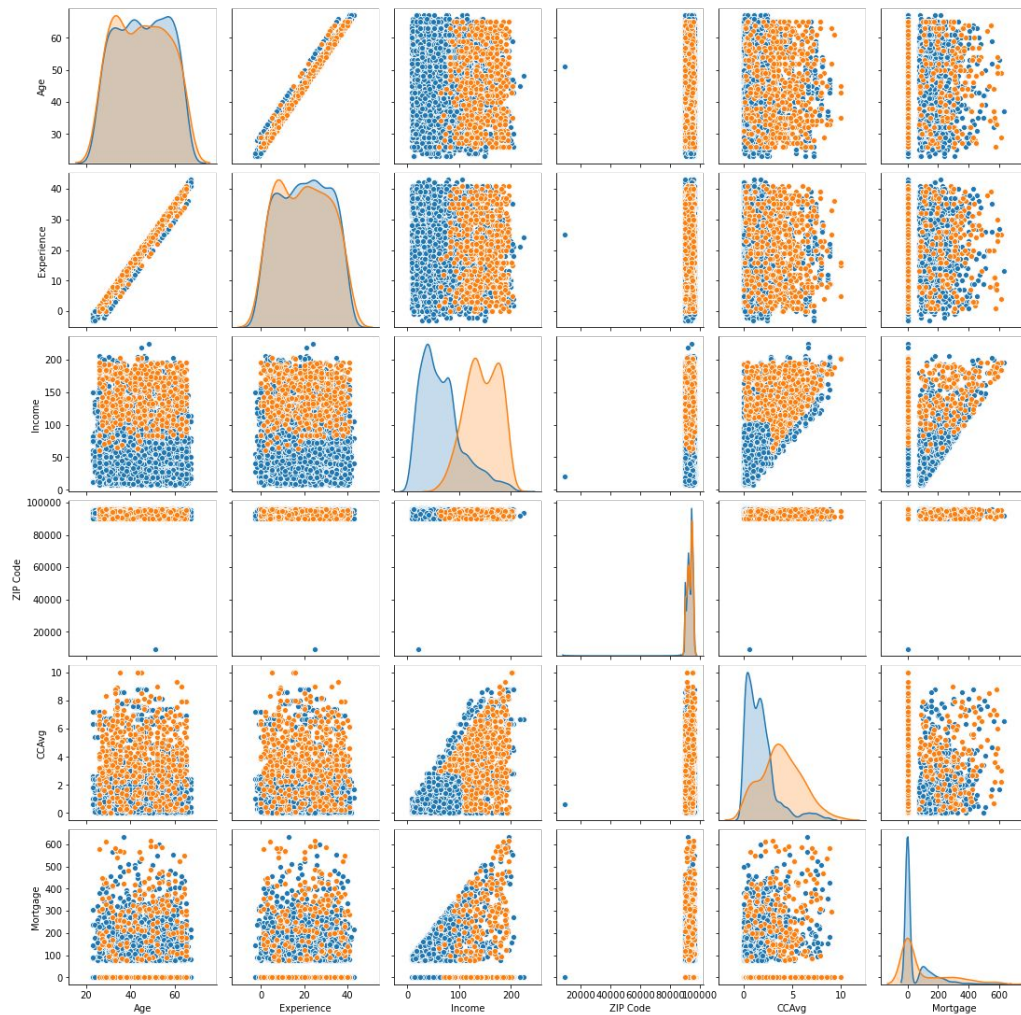
04

Info Menarik



04

Info Menarik



```
[ ] 1 bank_df.head()
```

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1

05

X

y

## Data Preparation

06

```
1 from sklearn.preprocessing import StandardScaler, MinMaxScaler
2 from sklearn.model_selection import train_test_split
3
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
5
6 sc = StandardScaler()
7 X_train = sc.fit_transform(X_train)
8 X_test = sc.transform(X_test)
9
10 X_train.shape, X_test.shape, y_train.shape, y_test.shape
(4500, 13), (500, 13), (4500, 2), (500, 2))
```

## Train Test Split, Standardization



07

## Bangun Model

```

1  # sequential model
2  ann_model = keras.Sequential()
3
4  # adding dense layer
5  ann_model.add(Dense(250, input_dim=13, kernel_initializer='normal', activation='relu'))
6  ann_model.add(Dropout(0.3))
7  ann_model.add(Dense(500, activation='relu'))
8  ann_model.add(Dropout(0.3))
9  ann_model.add(Dense(500, activation='relu'))
10 ann_model.add(Dropout(0.3))
11 ann_model.add(Dense(500, activation='relu'))
12 ann_model.add(Dropout(0.4))
13 ann_model.add(Dense(250, activation='linear'))
14 ann_model.add(Dropout(0.4))
15
16 # adding dense layer with softmax activation/output layer
17 ann_model.add(Dense(2, activation='softmax'))
18 ann_model.summary()

```

```

[ ] 1  ann_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=[f1_m]) # metrics=['accuracy']

```

08

## Compile Model

```

1  history = ann_model.fit(X_train, y_train, epochs=20, validation_split=0.2, verbose=1)

```

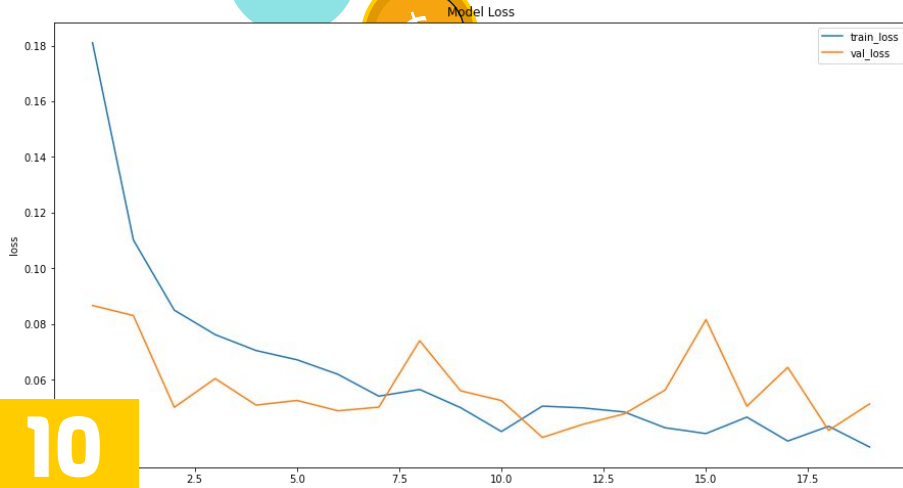
Epoch 1/20  
113/113 [=====] - 4s 17ms/step - loss: 0.1811 - f1\_m: 0.9336 - val\_loss: 0.0867 - val\_f1\_m: 0.9655  
Epoch 2/20  
113/113 [=====] - 2s 15ms/step - loss: 0.1102 - f1\_m: 0.9560 - val\_loss: 0.0830 - val\_f1\_m: 0.9644  
Epoch 3/20  
113/113 [=====] - 2s 15ms/step - loss: 0.0850 - f1\_m: 0.9699 - val\_loss: 0.0501 - val\_f1\_m: 0.9784  
Epoch 4/20  
113/113 [=====] - 2s 14ms/step - loss: 0.0762 - f1\_m: 0.9751 - val\_loss: 0.0604 - val\_f1\_m: 0.9806  
Epoch 5/20  
113/113 [=====] - 2s 15ms/step - loss: 0.0705 - f1\_m: 0.9773 - val\_loss: 0.0509 - val\_f1\_m: 0.9849  
Epoch 6/20  
113/113 [=====] - 2s 15ms/step - loss: 0.0672 - f1\_m: 0.9770 - val\_loss: 0.0526 - val\_f1\_m: 0.9838  
Epoch 7/20  
113/113 [=====] - 2s 15ms/step - loss: 0.0620 - f1\_m: 0.9790 - val\_loss: 0.0489 - val\_f1\_m: 0.9817  
Epoch 8/20  
113/113 [=====] - 2s 15ms/step - loss: 0.0541 - f1\_m: 0.9820 - val\_loss: 0.0502 - val\_f1\_m: 0.9806  
Epoch 9/20  
113/113 [=====] - 2s 17ms/step - loss: 0.0565 - f1\_m: 0.9829 - val\_loss: 0.0740 - val\_f1\_m: 0.9838  
Epoch 10/20  
113/113 [=====] - 2s 15ms/step - loss: 0.0500 - f1\_m: 0.9817 - val\_loss: 0.0560 - val\_f1\_m: 0.9795

09

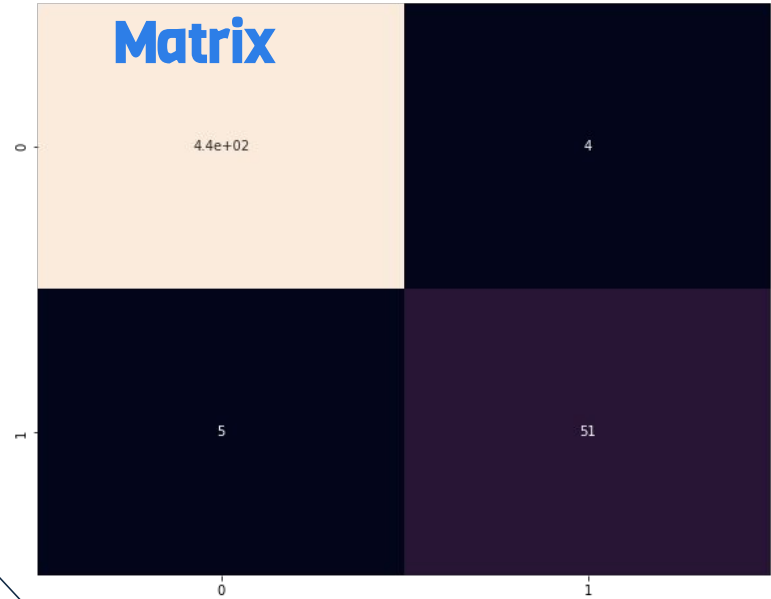
## Latih Model

10

## Visualisasi Loss



## 11 Confussion Matrix



```
1 print(metrics.classification_report(y_test, predict))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	444
1	0.93	0.91	0.92	56
accuracy			0.98	500
macro avg	0.96	0.95	0.95	500
weighted avg	0.98	0.98	0.98	500

## 12 Classification Report





04

## Kesimpulan



Rancangan Artificial Neural Network (ANN) berhasil diterapkan untuk keputusan kredit Thera Bank



Akurasi dari hasil prediksi yang didapatkan mencapai 98%





**Do you have any questions?**

**Thanks!**



**A picture is worth than a thousand words**

