

Analisis Segmentasi Pelanggan dengan Menggunakan *Credit Scoring*



Disusun oleh:

Nabila Agustina Cahyani Putri agustinacahyani5@gmail.com

Bidang Data Analytics

Mentor:

Dede Brahma Arianto

PT. Zona Edukasi Nusantara (Zenius Education)

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	4
LATAR BELAKANG CREDIT SCORING.....	4
1. 1. Pengertian Credit scoring.....	4
1. 2. Tujuan Bisnis.....	4
1. 3. Tujuan Data Mining.....	4
BAB II.....	5
METODE PENELITIAN.....	5
2. 1. Metode yang bisa digunakan.....	5
2. 2. Metode yang Dlgunakan pada Penelitian ini.....	5
BAB III.....	6
ALUR KERJA PENELITIAN.....	6
BAB IV.....	7
HASIL DAN PEMBAHASAN.....	7
4. 1. Business Understanding.....	7
4. 1. 1. Tujuan Bisnis.....	7
4. 1. 2. Tujuan Data MInning.....	7
4. 2. Data Understanding.....	7
4. 2. 1. Mengumpulkan Dataset.....	7
4. 2. 2. Penjelasan Kolom.....	7
4. 2. 3. Deskripsi Kolom.....	8
4. 2. 4. Tipe Data Setiap Kolom.....	9
4. 2. 5. Statistics Describe Setiap Kolom Numeric.....	9
4. 2. 6. Dataset 5 teratas.....	10
4. 2. 7. Dataset 5 terbawah.....	10
4. 2. 8. Identifikasi Nilai NULL.....	10
4. 2. 9. Identifikasi Nilai Pencilan dengan Boxplot Pada Kolom Numeric.....	11
4. 3. Data Preparation.....	11
4. 3. 1. Menghilangkan Kolom yang Tidak Penting.....	11
4. 3. 2. Mengidentifikasi Kembali Data Pencilan Pada Kolom Selain Kolom Credit_Score.....	12
4. 3. 3. Menghapus Data Pencilan dengan Teknik Persentil.....	12
4. 3. 4. Mengidentifikasi Data NULL.....	12
4. 3. 5. Mengidentifikasi Jumlah Data NULL.....	13
4. 3. 6. Menghapus Data NULL.....	13
4. 3. 7. Mengidentifikasi Relasi Setiap Kolom menggunakan Heat Map.....	14
4. 3. 8. Mengidentifikasi 10 Feature Terbaik Menggunakan chi3.....	14
4. 3. 9. Menginisiasi Kembali Dataset dengan 10 Feature Terbaik.....	14
4. 4. Modelling.....	15
4. 4. 1. KNN.....	15
4. 4. 2. Decision Tree.....	15
4. 5. Evaluation.....	15

4. 5. 1. KNN.....	15
4. 5. 2. Decision Tree.....	16
4. 6. Deployment.....	16
BAB V.....	18
KESIMPULAN.....	18

BAB I

LATAR BELAKANG *CREDIT SCORING*

1. 1. Pengertian *Credit scoring*

Credit Scoring adalah suatu analisis yang dilakukan lembaga untuk menghasilkan keputusan akan pemberian pinjaman untuk pelanggan berdasarkan klasifikasi mereka dengan berbagai *feature*.

1. 2. Tujuan Bisnis

Tujuan dari proses data analytics ini adalah mengelompokkan / klasifikasi kredit skor untuk mempermudah dalam identifikasi pelanggan.

1. 3. Tujuan Data Mining

Untuk melakukan klasifikasi pelanggan yang akan didasarkan 10 *feature* yang paling berpengaruh pada pengelompokan pelanggan.

BAB II

METODE PENELITIAN

2. 1. Metode yang bisa digunakan

Metode dalam penelitian ini dapat menggunakan seluruh metode yang terdapat pada kategori *supervised learning* pada *machine learning* seperti KNN, *decision tree*, *random forest*, *naive bayes*, dan SVM.

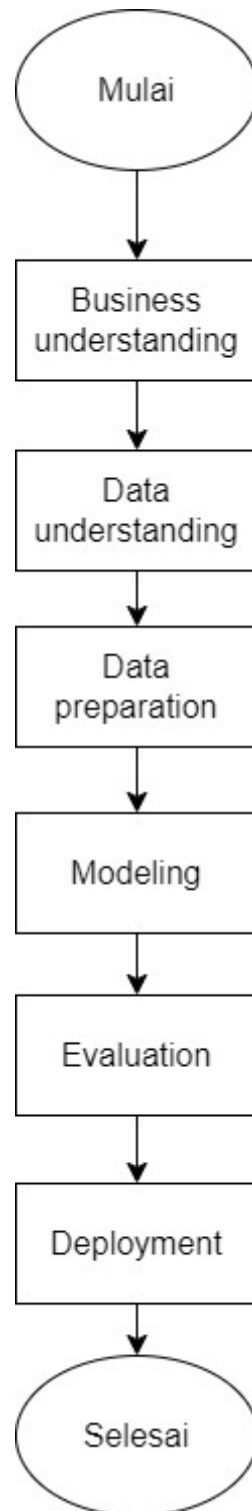
2. 2. Metode yang Digunakan pada Penelitian ini

Metode yang digunakan pada penelitian ini menggunakan metode KNN dan *decision tree* yang diidentifikasi selanjutnya dipilih metode yang paling besar dari segi *precision*.

BAB III

ALUR KERJA PENELITIAN

Pada penelitian ini menggunakan metode CRISP-DM dengan alur kerja seperti berikut:



BAB IV

HASIL DAN PEMBAHASAN

4. 1. *Business Understanding*

Business understanding dilakukan untuk mengetahui kebutuhan bisnis terhadap data. Berdasarkan judul pada penelitian ini *Business understanding* dapat dijabarkan sebagai berikut:

4. 1. 1. Tujuan Bisnis

Tujuan dari proses data analytics ini adalah mengelompokkan / klasifikasi kredit skor untuk mempermudah dalam identifikasi pelanggan.

4. 1. 2. Tujuan Data Mining

Untuk melakukan klasifikasi pelanggan yang akan didasarkan 10 *feature* yang paling berpengaruh pada pengelompokan pelanggan.

4. 2. *Data Understanding*

Data Understanding dilakukan untuk mengetahui data apa saja yang diperlukan untuk memenuhi tujuan dari *analytics* data. Pada penelitian ini *Data Understanding* dapat dijabarkan sebagai berikut:

4. 2. 1. Mengumpulkan Dataset

Dataset yang digunakan pada project ini adalah dataset dari Credit Score Classification Clean Data dari kaggle. Link dataset <https://www.kaggle.com/datasets/clkmuhamed/creditscoreclassification?select=train.csv>.

4. 2. 2. Penjelasan Kolom

- ID: identitas yang unik untuk setiap baris.
- Customer_ID: identitas yang unik untuk setiap data orang yang berada pada dataset.
- Month: berisi data bulan.
- Name: berisikan data nama orang pada dataset.
- Age: berisikan data umur orang pada dataset.
- SSN: berisikan data nomor jaminan sosial orang pada dataset.
- Occupation: berisikan pekerjaan orang pada dataset.
- Annual_Income: berisikan pendapatan orang pada dataset.
- Monthly_Inhand_Salary: berisikan gaji pokok bulanan orang pada dataset.
- Num_Bank_Accounts: berisikan jumlah rekening bank yang dimiliki manusia.
- Num_Credit_Card: berisikan jumlah kartu kredit lain yang dimiliki oleh orang pada dataset.
- Interest_Rate: berisikan tingkat bunga pada kartu kredit.
- Num_of_Loan: berisikan jumlah pinjaman yang diambil pada bank.
- Type_of_Loan: berisikan jenis pinjaman yang diambil oleh orang pada dataset.

- Delay_from_due_date: berisikan jumlah rata-rata hari keterlambatan pembayaran.
- Num_of_Delayed_Payment: berisikan jumlah rata-rata pembayaran yang tertunda.
- Changed_Credit_Limit: berisikan persentase perubahan limit kartu kredit.
- Num_Credit_Inquiries: berisikan jumlah pertanyaan kartu kredit.
- Credit_Mix: berisi klasifikasi bauran kredit.
- Outstanding_Debt: berisi sisa hutang yang harus dibayar.
- Credit_Utilization_Ratio: berisi rasio pemanfaatan kartu kredit.
- Credit_History_Age: berisi usia sejarah kredit orang pada dataset.
- Payment_of_Min_Amount: berisi jumlah minimum yang harus dibayarkan.
- Total_EMI_per_month: berisikan pembayaran EMI perbulan.
- Amount_invested_monthly: berisikan jumlah yang diinvestasikan oleh orang pada dataset.
- Payment_Behaviour: berisi perilaku pembayaran pelanggan.
- Monthly_Balance: berisikan jumlah saldo bulanan orang.
- Credit_Score: berisikan kelompok kartu kredit.

4. 2. 3. Deskripsi Kolom

```
In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 28 columns):
#   Column                               Non-Null Count  Dtype  
---  -
0   ID                                    100000 non-null  int64  
1   Customer_ID                          100000 non-null  int64  
2   Month                                100000 non-null  int64  
3   Name                                  100000 non-null  object  
4   Age                                   100000 non-null  float64 
5   SSN                                   100000 non-null  float64 
6   Occupation                            100000 non-null  object  
7   Annual_Income                         100000 non-null  float64 
8   Monthly_Inhand_Salary                 100000 non-null  float64 
9   Num_Bank_Accounts                     100000 non-null  float64 
10  Num_Credit_Card                       100000 non-null  float64 
11  Interest_Rate                         100000 non-null  float64 
12  Num_of_Loan                           100000 non-null  float64 
13  Type_of_Loan                           100000 non-null  object  
14  Delay_from_due_date                   100000 non-null  float64 
15  Num_of_Delayed_Payment                 100000 non-null  float64 
16  Changed_Credit_Limit                   100000 non-null  float64 
17  Num_Credit_Inquiries                   100000 non-null  float64 
18  Credit_Mix                             100000 non-null  object  
19  Outstanding_Debt                       100000 non-null  float64 
20  Credit_Utilization_Ratio               100000 non-null  float64 
21  Credit_History_Age                     100000 non-null  float64 
22  Payment_of_Min_Amount                   100000 non-null  object  
23  Total_EMI_per_month                     100000 non-null  float64 
24  Amount_invested_monthly                 100000 non-null  float64 
25  Payment_Behaviour                       100000 non-null  object  
26  Monthly_Balance                         100000 non-null  float64 
27  Credit_Score                           100000 non-null  object  
dtypes: float64(18), int64(3), object(7)
memory usage: 21.4+ MB
```


4. 2. 4. Tipe Data Setiap Kolom

```
In [15]: df.dtypes

Out[15]: ID                int64
Customer_ID             int64
Month                  int64
Name                   object
Age                   float64
SSN                   float64
Occupation              object
Annual_Income           float64
Monthly_Inhand_Salary   float64
Num_Bank_Accounts       float64
Num_Credit_Card         float64
Interest_Rate           float64
Num_of_Loan             float64
Type_of_Loan            object
Delay_from_due_date     float64
Num_of_Delayed_Payment   float64
Changed_Credit_Limit     float64
Num_Credit_Inquiries     float64
Credit_Mix              object
Outstanding_Debt         float64
Credit_Utilization_Ratio float64
Credit_History_Age       float64
Payment_of_Min_Amount    object
Total_EMI_per_month      float64
Amount_invested_monthly float64
Payment_Behaviour        object
Monthly_Balance          float64
Credit_Score            object
dtype: object
```

4. 2. 5. Statistics Describe Setiap Kolom Numeric

```
In [4]: df.describe().T

Out[4]:
```

	count	mean	std	min	25%	50%	75%	max
ID	100000.0	8.063150e+04	4.330149e+04	5634.000000	4.313275e+04	8.063150e+04	1.181302e+05	1.556290e+05
Customer_ID	100000.0	2.598267e+04	1.434054e+04	1006.000000	1.366450e+04	2.577700e+04	3.838500e+04	5.099900e+04
Month	100000.0	4.500000e+00	2.291299e+00	1.000000	2.750000e+00	4.500000e+00	6.250000e+00	8.000000e+00
Age	100000.0	3.331634e+01	1.076481e+01	14.000000	2.400000e+01	3.300000e+01	4.200000e+01	5.600000e+01
SSN	100000.0	5.004617e+08	2.908267e+08	81349.000000	2.451686e+08	5.006886e+08	7.560027e+08	9.999934e+08
Annual_Income	100000.0	5.050512e+04	3.829942e+04	7005.930000	1.934297e+04	3.699971e+04	7.168347e+04	1.799873e+05
Monthly_Inhand_Salary	100000.0	4.197271e+03	3.186432e+03	303.645417	1.626594e+03	3.095905e+03	5.957715e+03	1.520463e+04
Num_Bank_Accounts	100000.0	5.368820e+00	2.593314e+00	0.000000	3.000000e+00	5.000000e+00	7.000000e+00	1.100000e+01
Num_Credit_Card	100000.0	5.533570e+00	2.067098e+00	0.000000	4.000000e+00	5.000000e+00	7.000000e+00	1.100000e+01
Interest_Rate	100000.0	1.453208e+01	8.741330e+00	1.000000	7.000000e+00	1.300000e+01	2.000000e+01	3.400000e+01
Num_of_Loan	100000.0	3.532880e+00	2.446356e+00	0.000000	2.000000e+00	3.000000e+00	5.000000e+00	9.000000e+00
Delay_from_due_date	100000.0	2.108141e+01	1.480456e+01	0.000000	1.000000e+01	1.800000e+01	2.800000e+01	6.200000e+01
Num_of_Delayed_Payment	100000.0	1.331312e+01	6.237166e+00	0.000000	9.000000e+00	1.400000e+01	1.800000e+01	2.500000e+01
Changed_Credit_Limit	100000.0	1.047032e+01	6.609481e+00	0.500000	5.380000e+00	9.400000e+00	1.485000e+01	2.998000e+01
Num_Credit_Inquiries	100000.0	5.798250e+00	3.867826e+00	0.000000	3.000000e+00	5.000000e+00	8.000000e+00	1.700000e+01
Outstanding_Debt	100000.0	1.426220e+03	1.155129e+03	0.230000	5.660725e+02	1.166155e+03	1.945963e+03	4.998070e+03
Credit_Utilization_Ratio	100000.0	3.228517e+01	5.116875e+00	20.000000	2.805257e+01	3.230578e+01	3.649666e+01	5.000000e+01
Credit_History_Age	100000.0	2.212205e+02	9.968072e+01	1.000000	1.440000e+02	2.190000e+02	3.020000e+02	4.040000e+02
Total_EMI_per_month	100000.0	1.076992e+02	1.322671e+02	0.000000	2.926889e+01	6.646230e+01	1.473926e+02	1.779103e+03
Amount_invested_monthly	100000.0	5.510131e+01	3.900693e+01	0.000000	2.795911e+01	4.515655e+01	7.129580e+01	4.341911e+02
Monthly_Balance	100000.0	3.926976e+02	2.016527e+02	0.007760	2.676160e+02	3.338654e+02	4.632157e+02	1.183931e+03

4. 2. 6. Dataset 5 teratas

```
In [5]: df.head()
```

	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	..
0	5634	3392	1	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1824.843333	3.0	.
1	5635	3392	2	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1824.843333	3.0	.
2	5636	3392	3	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1824.843333	3.0	.
3	5637	3392	4	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1824.843333	3.0	.
4	5638	3392	5	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1824.843333	3.0	.

5 rows × 28 columns

4. 2. 7. Dataset 5 terbawah

```
In [6]: df.tail()
```

	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	..
99995	155625	37932	4	Nicks	25.0	78735990.0	Mechanic	39628.99	3359.415833	4.0	.
99996	155626	37932	5	Nicks	25.0	78735990.0	Mechanic	39628.99	3359.415833	4.0	.
99997	155627	37932	6	Nicks	25.0	78735990.0	Mechanic	39628.99	3359.415833	4.0	.
99998	155628	37932	7	Nicks	25.0	78735990.0	Mechanic	39628.99	3359.415833	4.0	.
99999	155629	37932	8	Nicks	25.0	78735990.0	Mechanic	39628.99	3359.415833	4.0	.

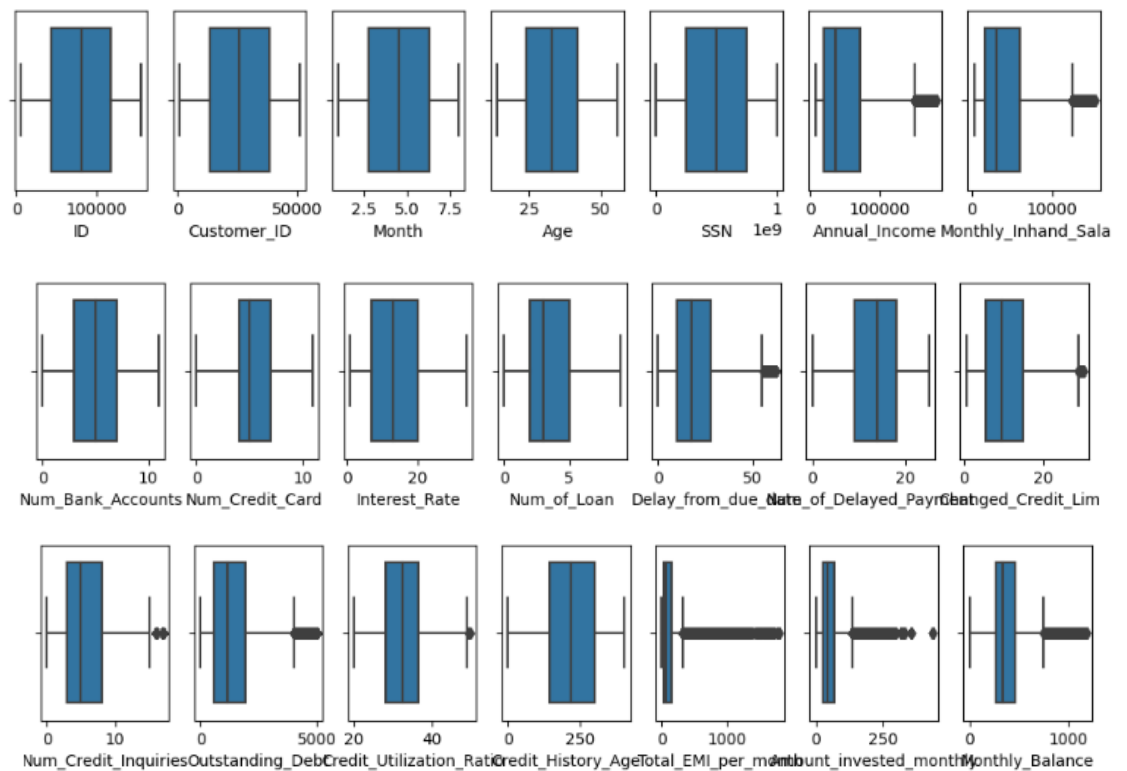
5 rows × 28 columns

4. 2. 8. Identifikasi Nilai NULL

```
In [7]: df.isnull().any()
```

ID	False
Customer_ID	False
Month	False
Name	False
Age	False
SSN	False
Occupation	False
Annual_Income	False
Monthly_Inhand_Salary	False
Num_Bank_Accounts	False
Num_Credit_Card	False
Interest_Rate	False
Num_of_Loan	False
Type_of_Loan	False
Delay_from_due_date	False
Num_of_Delayed_Payment	False
Changed_Credit_Limit	False
Num_Credit_Inquiries	False
Credit_Mix	False
Outstanding_Debt	False
Credit_Utilization_Ratio	False
Credit_History_Age	False
Payment_of_Min_Amount	False
Total_EMI_per_month	False
Amount_invested_monthly	False
Payment_Behaviour	False
Monthly_Balance	False
Credit_Score	False
dtype:	bool

4. 2. 9. Identifikasi Nilai Penciran dengan *Boxplot* Pada Kolom *Numeric*



Terdapat outlier pada kolom Annual_Income, Monthly_Inhand_Salary, Delay_from_due_date, Changed_Credit_Limit, Num_Credit_Inquiries, Outstanding_Debt, Credit_Utilization_Ratio, Total_EMI_per_month, Amount_invested_monthly, dan Monthly_Balance.

4. 3. Data Preparation

4. 3. 1. Menghilangkan Kolom yang Tidak Penting

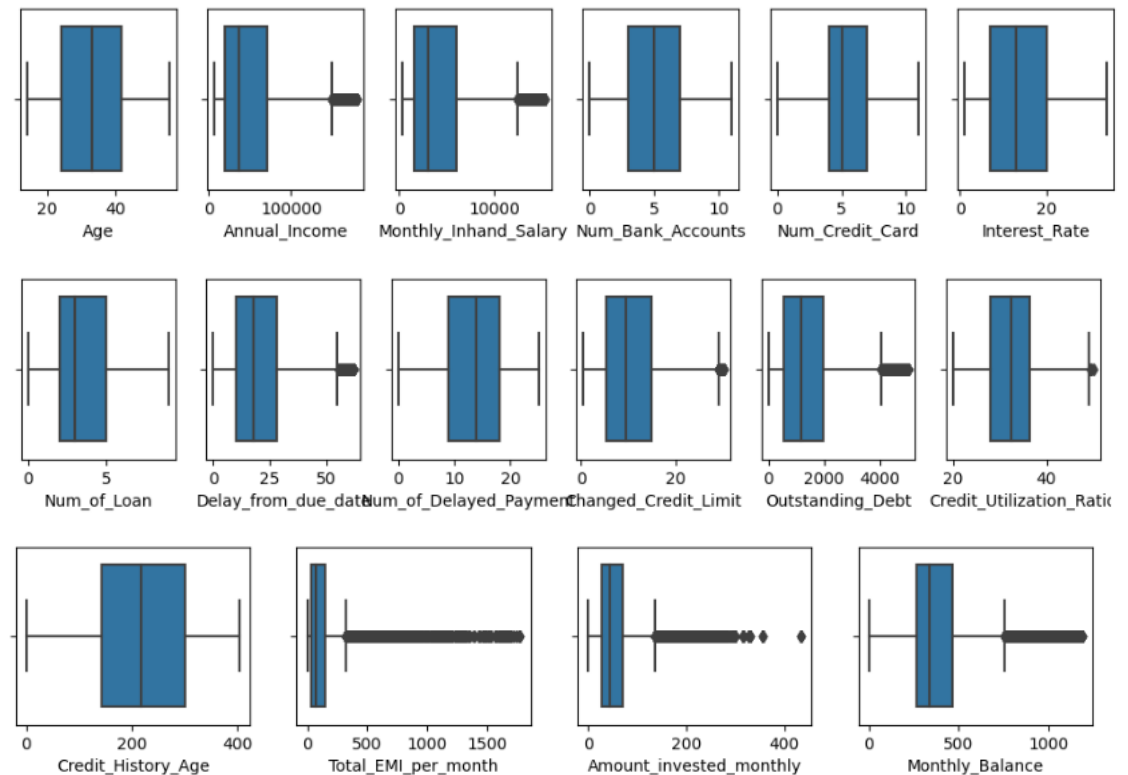
```
In [4]: df = df.drop(['ID', 'Customer_ID', 'Month', 'Name', 'SSN', 'Occupation', 'Num_Credit_Inquiries', 'Payment_Behaviour', 'T'])
```

```
In [5]: df.head()
```

```
Out[5]:
```

	Age	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	Num_Credit_Card	Interest_Rate	Num_of_Loan	Delay_from_due
0	23.0	19114.12	1824.843333	3.0	4.0	3.0	4.0	
1	23.0	19114.12	1824.843333	3.0	4.0	3.0	4.0	
2	23.0	19114.12	1824.843333	3.0	4.0	3.0	4.0	
3	23.0	19114.12	1824.843333	3.0	4.0	3.0	4.0	
4	23.0	19114.12	1824.843333	3.0	4.0	3.0	4.0	

4. 3. 2. Mengidentifikasi Kembali Data Pencilan Pada Kolom Selain Kolom Credit_Score



4. 3. 3. Menghapus Data Pencilan dengan Teknik Persentil

```
In [13]: trim_percent = 10
for data in data_num.columns:
    lower_bound = np.percentile(data_num[f'{data}'], trim_percent/2)
    upper_bound = np.percentile(data_num[f'{data}'], 100 - trim_percent/2)
    print(f'{i}.', lower_bound, upper_bound)
    df[f'{data}'] = df[f'{data}'][(df[f'{data}'] > lower_bound) & (df[f'{data}'] < upper_bound)]
```

4. 3. 4. Mengidentifikasi Data NULL

```
In [14]: df.isnull().any()
```

```
Out[14]: Age                True
Annual_Income              True
Monthly_Inhand_Salary      True
Num_Bank_Accounts          True
Num_Credit_Card            True
Interest_Rate              True
Num_of_Loan                True
Delay_from_due_date        True
Num_of_Delayed_Payment     True
Changed_Credit_Limit       True
Outstanding_Debt           True
Credit_Utilization_Ratio   True
Credit_History_Age        True
Total_EMI_per_month        True
Amount_invested_monthly    True
Monthly_Balance            True
Credit_Score               False
dtype: bool
```

4. 3. 5. Mengidentifikasi Jumlah Data NULL

```
In [15]: df.isnull().sum()

Out[15]: Age                11766
Annual_Income              10000
Monthly_Inhand_Salary      10008
Num_Bank_Accounts          14295
Num_Credit_Card            27706
Interest_Rate              11552
Num_of_Loan                18600
Delay_from_due_date        11286
Num_of_Delayed_Payment     12785
Changed_Credit_Limit       10016
Outstanding_Debt           10000
Credit_Utilization_Ratio   10000
Credit_History_Age        10351
Total_EMI_per_month        15991
Amount_invested_monthly    10000
Monthly_Balance            10000
Credit_Score               0
dtype: int64
```

4. 3. 6. Menghapus Data NULL

```
In [16]: df.dropna(inplace=True)

In [17]: df.isnull().any()

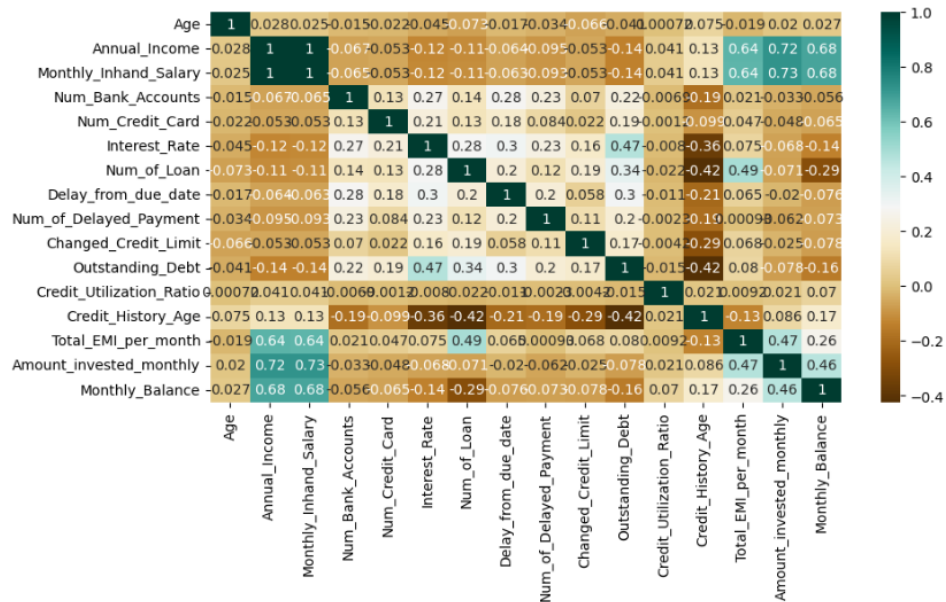
Out[17]: Age                False
Annual_Income              False
Monthly_Inhand_Salary      False
Num_Bank_Accounts          False
Num_Credit_Card            False
Interest_Rate              False
Num_of_Loan                False
Delay_from_due_date        False
Num_of_Delayed_Payment     False
Changed_Credit_Limit       False
Outstanding_Debt           False
Credit_Utilization_Ratio   False
Credit_History_Age        False
Total_EMI_per_month        False
Amount_invested_monthly    False
Monthly_Balance            False
Credit_Score               False
dtype: bool
```

4. 3. 7. Mengidentifikasi Relasi Setiap Kolom menggunakan *Heat Map*

Heat Map

```
In [18]: plt.figure(figsize=(10,5))
c= df.corr()
sns.heatmap(c,cmap="BrBG",annot=True)
```

Out[18]: <Axes: >



4. 3. 8. Mengidentifikasi 10 *Feature* Terbaik Menggunakan chi3

```
In [19]: X = df.iloc[:, :-1]
y = df.iloc[:, -1]
```

```
In [20]: bestfeature = SelectKBest(score_func=chi2, k=10)
fit = bestfeature.fit(X,y)
```

```
In [21]: dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
```

```
In [22]: featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score']
```

```
In [24]: best_feat = featureScores.nlargest(10,'Score')
print(best_feat)
```

	Specs	Score
10	Outstanding_Debt	1.765561e+06
1	Annual_Income	1.648950e+06
2	Monthly_Inhand_Salary	1.321903e+05
12	Credit_History_Age	5.841267e+04
5	Interest_Rate	1.126581e+04
15	Monthly_Balance	7.541405e+03
7	Delay_from_due_date	7.114007e+03
13	Total_EMI_per_month	5.133993e+03
9	Changed_Credit_Limit	1.281642e+03
6	Num_of_Loan	9.016268e+02

4. 3. 9. Menginisiasi Kembali Dataset dengan 10 *Feature* Terbaik

```
In [28]: df_clean = pd.merge(left=datas, right=y, how='inner', left_on='index', right_on='index')
df_clean.drop('index', axis=1, inplace=True)
```

```
In [29]: df_clean.to_csv(path_or_buf='../Dataset/data_clean.csv')
```

4. 4. Modelling

Pada penelitian ini menggunakan model KNN dan *Decision Tree*.

4. 4. 1. KNN

```
In [5]: X = df.iloc[:, :-1]
        y = df.iloc[:, -1]

In [6]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

In [7]: modelKNN = KNeighborsClassifier(n_neighbors=3, weights='uniform', metric='euclidean')
        modelKNN.fit(X_train, y_train)

Out[7]: KNeighborsClassifier(metric='euclidean', n_neighbors=3)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

4. 4. 2. Decision Tree

```
In [5]: X = df.iloc[:, :-1]
        y = df.iloc[:, -1]

In [6]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

In [13]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

In [14]: modelDT = DecisionTreeClassifier(criterion = "entropy", random_state = 1234, max_depth = 5, min_samples_leaf = 10)
        modelDT.fit(X_train, y_train)

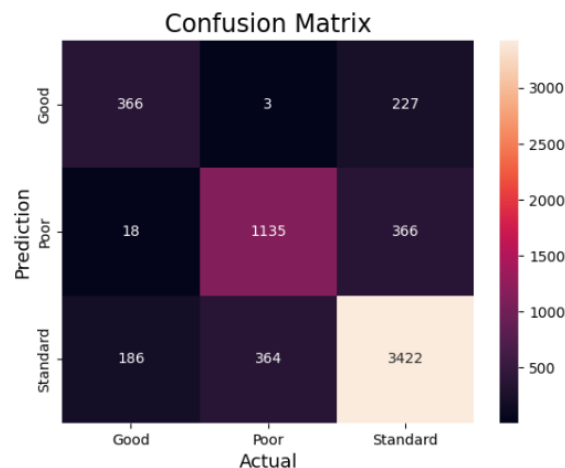
Out[14]: DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_leaf=10,
                                random_state=1234)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

4. 5. Evaluation

Pada penelitian ini evaluasi model dilakukan dengan menggunakan *confusion matrices* yang divisualisasikan dengan menggunakan *seaborn*.

4. 5. 1. KNN

```
In [11]: cm = confusion_matrix(y_test, y_pred)
        sns.heatmap(cm,
                    annot=True,
                    fmt='g',
                    xticklabels=['Good', 'Poor', 'Standard'],
                    yticklabels=['Good', 'Poor', 'Standard'])
        plt.ylabel('Prediction', fontsize=13)
        plt.xlabel('Actual', fontsize=13)
        plt.title('Confusion Matrix', fontsize=17)
        plt.show()
```

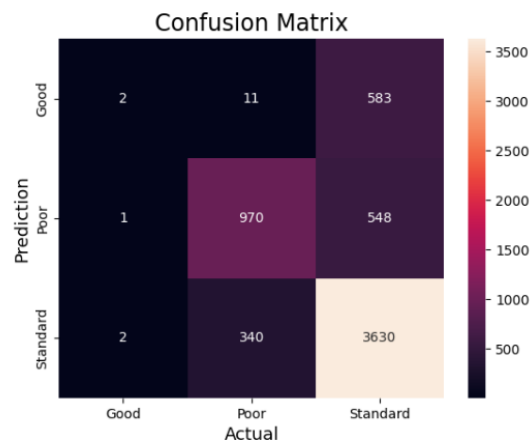


```
In [9]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.8087727944800395

4. 5. 2. Decision Tree

```
In [17]: cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['Good', 'Poor', 'Standard'],
            yticklabels=['Good', 'Poor', 'Standard'])
plt.ylabel('Prediction', fontsize=13)
plt.xlabel('Actual', fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
```



```
In [16]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.756037456875308

Setelah melakukan evaluasi diketahui bahwa model dari accuracy lebih tinggi dengan nilai sebesar 0.808 dan selanjutnya diekstraksi untuk tahap *Deployment*.

4. 6. Deployment

Deployment menggunakan model yang telah dibuat sebelumnya dan akan di-*deploy* menggunakan *library* streamlit dapat diakses dalam link github

<https://github.com/Nabilaagustina/Essay/blob/main/Deployment/Deployment.py>.

← → ↻ localhost:8501 🔍 🌐 ⭐ 🗂️ 📄 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📙 📚 📛 📜 📝 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📿

Credit Score Classification

Pengertian hash

Insert a Outstanding Debt	Insert a Annual Income
<input type="text" value="0"/>	<input type="text" value="0"/>
The current number is 0	The current number is 0
Insert a Monthly Inhand Salary	Insert a Credit History Age
<input type="text" value="0"/>	<input type="text" value="0"/>
The current number is 0	The current number is 0
Insert a Monthly Interest Rate	Insert a Monthly Balance
<input type="text"/>	<input type="text"/>

Insert a Monthly Interest Rate

0

–

+

The current number is 0

Insert a Monthly Balance

0

–

+

The current number is 0

Insert a Delay from due date

0

–

+

The current number is 0

Insert a Total EMI per month

0

–

+

The current number is 0

Insert a Changed Credit Limit

0

–

+

The current number is 0

Insert a Num of Loan

0

–

+

The current number is 0

	input_data
Outstanding_Debt	0
Annual_Income	0
Monthly_Inhand_Salary	0
Credit_History_Age	0
Interest_Rate	0
Monthly_Balance	0
Delay_from_due_date	0
Total_EMI_per_month	0
Changed_Credit_Limit	0
Num_of_Loan	0

Buat Prediksi

BAB V

KESIMPULAN

Penelitian ini dapat mengklasifikasi pelanggan dengan label Good, Poor, dan Standard yang didasarkan oleh pemrograman yang telah dilakukan sebelumnya. *Coding* yang digunakan pada penelitian ini dapat diakses pada link github: <https://github.com/Nabilaagustina/Essay>.