

Analyze the behavior of loan property customers



Rezki Dwi Rahmantlyo

 **rezki dwi rahmantlyo**

 **<https://github.com/Rezki10>**

- Analisa ini berisi analisis yang menjelaskan **tingkah laku** dan **karakteristik pelanggan** di sebuah bank yang menawarkan program pinjaman properti.
- Analisis ini bertujuan untuk memahami preferensi pelanggan terhadap **jenis properti** serta bagaimana **status perkawinan** mempengaruhi **durasi peminjaman** yang dipilih oleh mereka.



Hasil Analisis

Gambar di samping merupakan rangkuman informasi dari dataset yang digunakan yaitu loan_customer.csv. Rangkuman poin - poin utamanya sebagai berikut:

- Terdiri dari **17 kolom/fitur** dan **621 baris**
- Terdapat **11 kolom** yang memiliki missing value
- Fitur **birth_date** dan **phone_number** memiliki kesalahan tipe data

```
# Melihat informasi terkait dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 621 entries, 0 to 620
```

```
Data columns (total 17 columns):
```

#	Column	Non-Null Count	Dtype
0	loan_id	621 non-null	object
1	birth_date	572 non-null	object
2	phone_number	569 non-null	float64
3	gender	608 non-null	object
4	married	618 non-null	object
5	dependents	605 non-null	object
6	education	620 non-null	object
7	self_employed	588 non-null	object
8	applicant_income	612 non-null	float64
9	coapplicant_income	621 non-null	float64
10	loan_amount	621 non-null	int64
11	loan_term_month	621 non-null	int64
12	loan_term_year	621 non-null	int64
13	credit_history	570 non-null	float64
14	has_credit_card	88 non-null	object
15	property_type	620 non-null	object
16	loan_status	621 non-null	object

```
dtypes: float64(4), int64(3), object(10)
```

```
memory usage: 82.6+ KB
```

Hasil Analisis

Gambar di bawah ini menampilkan statistik deskriptif dari dataset. Dari data yang ditampilkan, terlihat bahwa fitur-fitur yang ada memiliki jarak yang signifikan antara nilai **mean** dan **median**. Hal ini bisa menjadi indikasi bahwa terdapat **outliers** dalam fitur-fitur tersebut.

```
# Melihat sebaran data secara statistik
df.describe().T
```

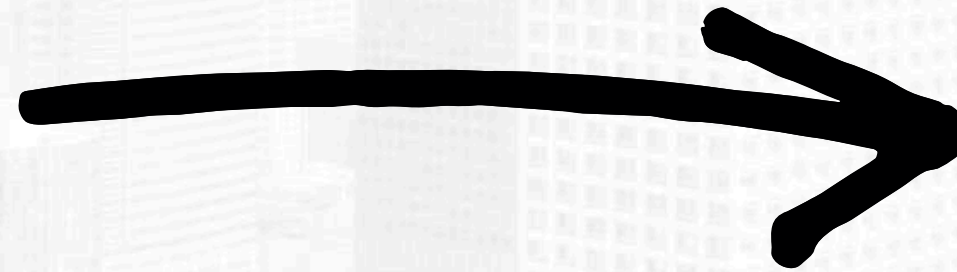
	count	mean	std	min	25%	50%	75%	max
phone_number	569.0	5.468552e+11	1.987860e+11	6.281101e+10	6.281140e+11	6.281343e+11	6.285665e+11	6.287799e+11
applicant_income	612.0	5.681549e+03	9.056454e+03	1.500000e+02	2.874500e+03	3.812500e+03	5.815750e+03	1.705780e+05
coapplicant_income	621.0	1.616874e+03	2.915075e+03	0.000000e+00	0.000000e+00	1.167000e+03	2.302000e+03	4.166700e+04
loan_amount	621.0	1.451337e+05	8.401112e+04	9.000000e+03	1.000000e+05	1.250000e+05	1.650000e+05	7.000000e+05
loan_term_month	621.0	3.426087e+02	6.409084e+01	1.200000e+01	3.600000e+02	3.600000e+02	3.600000e+02	4.800000e+02
loan_term_year	621.0	2.855072e+01	5.340903e+00	1.000000e+00	3.000000e+01	3.000000e+01	3.000000e+01	4.000000e+01
credit_history	570.0	8.421053e-01	3.649626e-01	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00

Merubah Tipe data

```
# Melihat informasi terkait dataset
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 621 entries, 0 to 620
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   loan_id                621 non-null    object
1   birth_date             572 non-null    object
2   phone_number           569 non-null    float64
3   gender                 608 non-null    object
4   married                618 non-null    object
5   dependents             605 non-null    object
6   education              620 non-null    object
7   self_employed          588 non-null    object
8   applicant_income       612 non-null    float64
9   coapplicant_income     621 non-null    float64
10  loan_amount            621 non-null    int64
11  loan_term_month        621 non-null    int64
12  loan_term_year         621 non-null    int64
13  credit_history          570 non-null    float64
14  has_credit_card        88 non-null     object
15  property_type          620 non-null    object
16  loan_status            621 non-null    object
dtypes: float64(4), int64(3), object(10)
memory usage: 82.6+ KB
```

Before



```
# Ubah tipe data untuk fitur birth_date dan juga phone_number

df['birth_date'] = pd.to_datetime(df['birth_date'])
df['phone_number'] = df['phone_number'].astype(str)
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 621 entries, 0 to 620
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   loan_id                621 non-null    object
1   birth_date             572 non-null    datetime64[ns]
2   phone_number           621 non-null    object
3   gender                 608 non-null    object
4   married                618 non-null    object
5   dependents             605 non-null    object
6   education              620 non-null    object
7   self_employed          588 non-null    object
8   applicant_income       612 non-null    float64
9   coapplicant_income     621 non-null    float64
10  loan_amount            621 non-null    int64
11  loan_term_month        621 non-null    int64
12  loan_term_year         621 non-null    int64
13  credit_history          570 non-null    float64
14  has_credit_card        88 non-null     object
15  property_type          620 non-null    object
16  loan_status            621 non-null    object
dtypes: datetime64[ns](1), float64(3), int64(3), object(10)
memory usage: 82.6+ KB
```

After

Handling Missing Value

Gambar di samping merupakan informasi terkait jumlah missing value pada masing - masing fitur. Langkah - langkah yang akan dilakukan dalam handling missing value yaitu:

- Drop fitur has_credit_card (dikarenakan memiliki jumlah missing value yang banyak)
- Mengisi missing value dengan nilai 0 pada fitur phone_number (dikarenakan satu orang hanya satu nomor hp saja)
- Mengisi nilai median untuk fitur credit_history dan applicant_income (dikarenakan bertipe data numerik dan distribusi datanya skewed)
- Mengisi nilai modus untuk fitur self_employed, dependents, gender, birth_date, married, education, dan property_type (dikarenakan bertipe data kategorik)

```
# Melihat jumlah missing value pada masing masing fitur
```

```
df.isnull().sum()
```

```
loan_id          0
birth_date       49
phone_number     0
gender           13
married          3
dependents       16
education         1
self_employed    33
applicant_income  9
coapplicant_income 0
loan_amount      0
loan_term_month  0
loan_term_year   0
credit_history    51
has_credit_card  533
property_type     1
loan_status       0
dtype: int64
```

Handling Missing Value

```
# Melihat jumlah missing value pada masing masing fitur
```

```
df.isnull().sum()
```

loan_id	0
birth_date	49
phone_number	0
gender	13
married	3
dependents	16
education	1
self_employed	33
applicant_income	9
coapplicant_income	0
loan_amount	0
loan_term_month	0
loan_term_year	0
credit_history	51
has_credit_card	533
property_type	1
loan_status	0
dtype: int64	

Before



```
# Menghapus fitur 'has_credit_card' dari DataFrame
df.drop('has_credit_card', axis=1, inplace=True)

# Mengisi missing value 0 untuk fitur phone_number
df['phone_number'].fillna(0, inplace=True)

# Mengisi nilai median untuk fitur credit_history dan applicant_income
df['applicant_income'].fillna(df['applicant_income'].median(), inplace=True)
df['credit_history'].fillna(df['credit_history'].median(), inplace=True)

# Mengisi nilai modus untuk fitur self_employed, dependents, gender, birth_date
df['gender'].fillna(df['gender'].mode()[0], inplace=True)
df['married'].fillna(df['married'].mode()[0], inplace=True)
df['dependents'].fillna(df['dependents'].mode()[0], inplace=True)
df['education'].fillna(df['education'].mode()[0], inplace=True)
df['self_employed'].fillna(df['self_employed'].mode()[0], inplace=True)
df['property_type'].fillna(df['property_type'].mode()[0], inplace=True)
df['birth_date'].fillna(df['birth_date'].mode()[0], inplace=True)

# Menampilkan jumlah missing value setelah diisi
print(df.isnull().sum())
```

loan_id	0
birth_date	0
phone_number	0
gender	0
married	0
dependents	0
education	0
self_employed	0
applicant_income	0
coapplicant_income	0
loan_amount	0
loan_term_month	0
loan_term_year	0
credit_history	0
property_type	0
loan_status	0

After

Handling Duplicated Data

- Terdapat **5 data** yang terdeteksi duplikat
- Kemungkinan dikarenakan **input data double**
- Mengatasi duplicate data dengan cara **menghapus baris** yang terdeteksi duplicated

```
[10] # Melihat jumlah data yang duplikat  
df.duplicated().sum()
```

5



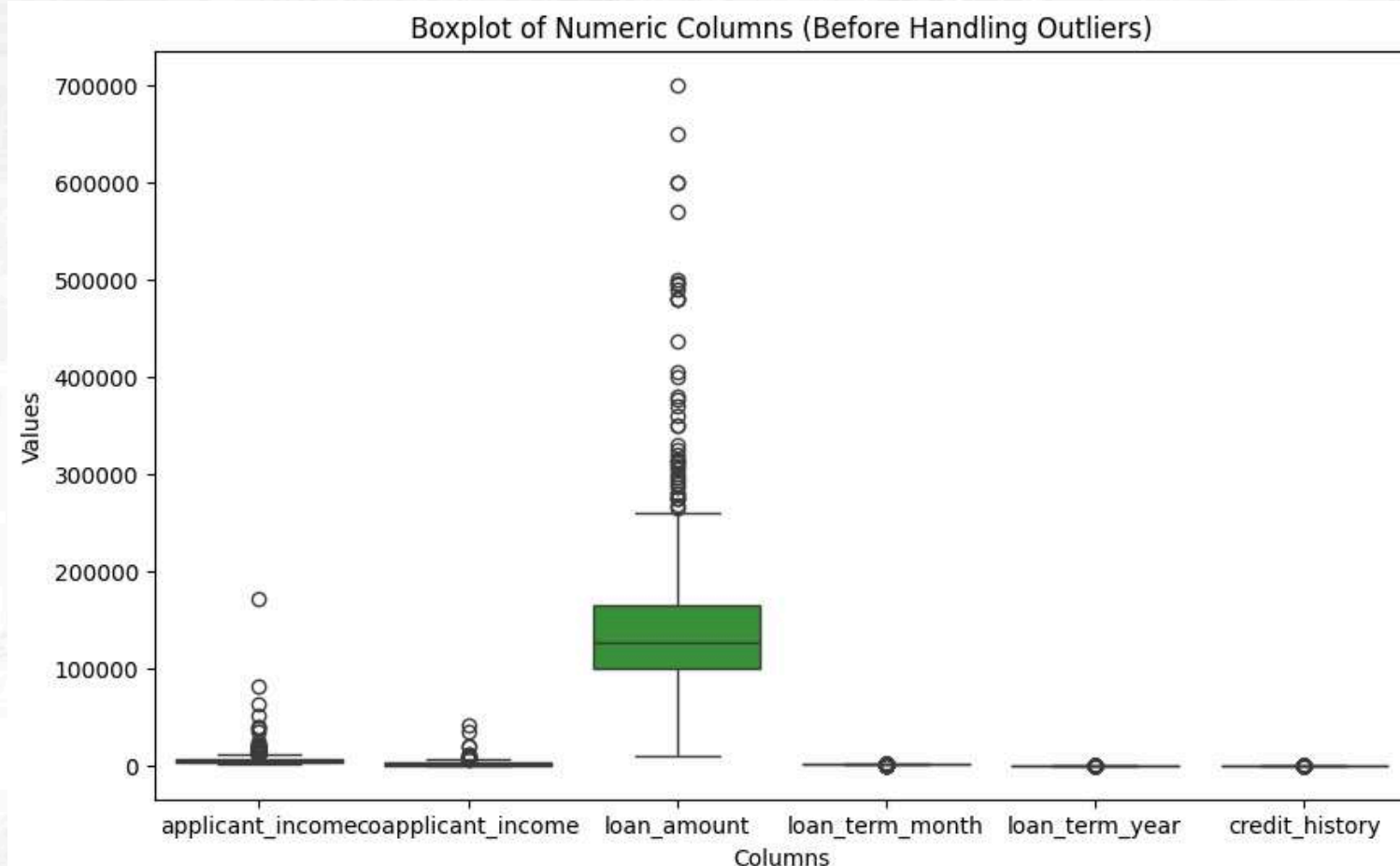
```
[11] # Menghapus baris yang duplikat  
df.drop_duplicates(inplace=True)
```

```
[13] # Mengecek kembali jumlah data yang duplikat  
df.duplicated().sum()
```

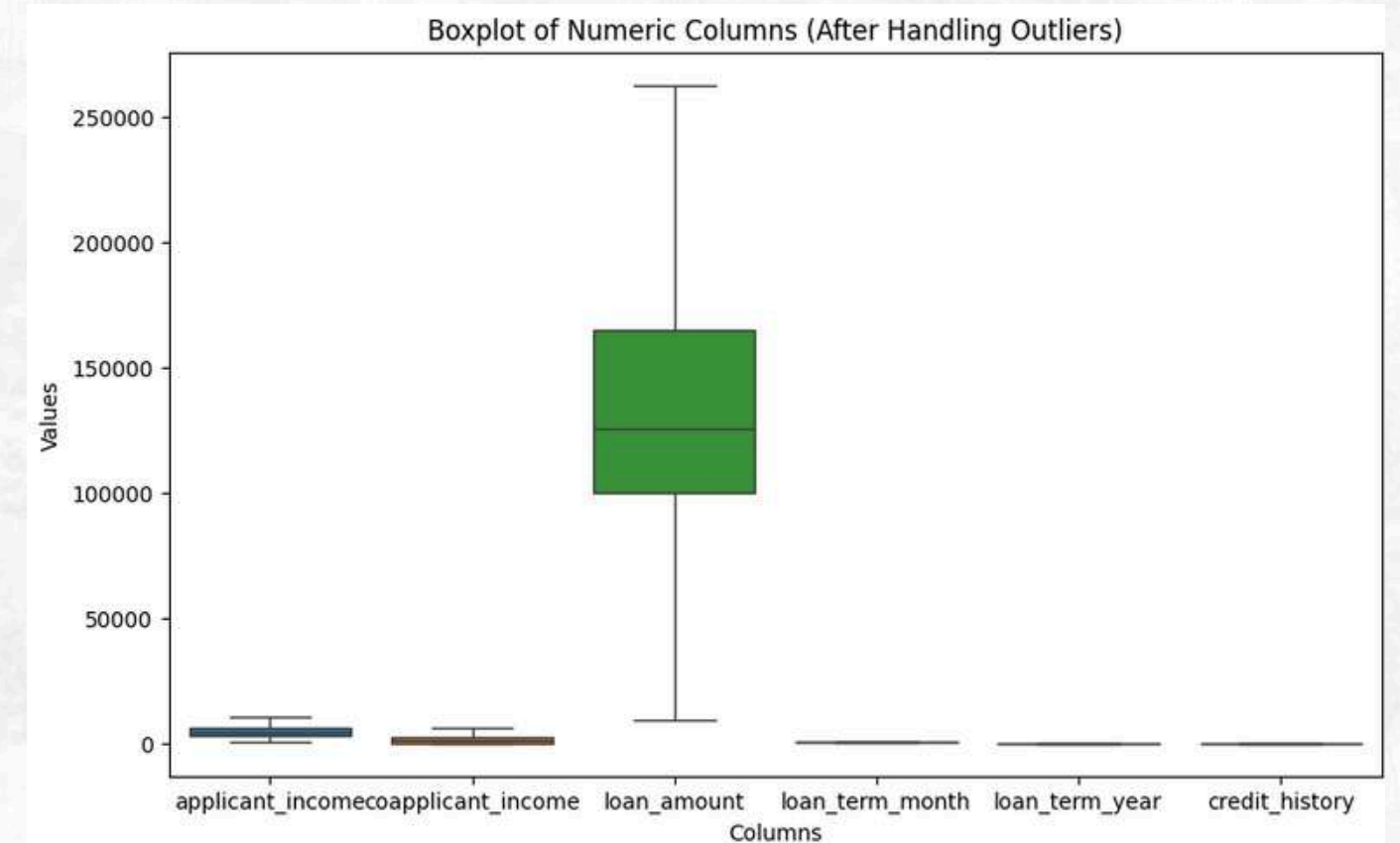
0

Handling Outliers

```
numeric_columns = ["applicant_income", "coapplicant_income",  
                  "loan_amount", "loan_term_month", "loan_term_year",  
                  "credit_history"]  
  
# Membuat boxplot  
plt.figure(figsize=(10, 6))  
sns.boxplot(data=df[numeric_columns])  
plt.title('Boxplot of Numeric Columns (Before Handling Outliers)')  
plt.xlabel('Columns')  
plt.ylabel('Values')  
plt.show()
```



```
# Fungsi untuk mengatasi outliers dengan IQR  
def handle_outliers(df, column):  
    Q1 = df[column].quantile(0.25)  
    Q3 = df[column].quantile(0.75)  
    IQR = Q3 - Q1  
    lower_bound = Q1 - 1.5 * IQR  
    upper_bound = Q3 + 1.5 * IQR  
  
    df[column] = df[column].apply(lambda x: upper_bound if x > upper_bound else  
                                  (lower_bound if x < lower_bound else x))  
    return df  
  
for column in numeric_columns:  
    df = handle_outliers(df, column)  
  
# Membuat boxplot  
plt.figure(figsize=(10, 6))  
sns.boxplot(data=df[numeric_columns])  
plt.title('Boxplot of Numeric Columns (After Handling Outliers)')  
plt.xlabel('Columns')  
plt.ylabel('Values')  
plt.show()
```



Feature Encoding

Feature encoding adalah merubah **nilai kategorik ke numerik** yang tujuannya untuk memudahkan **interpretasi hasil analisis**. Berikut fitur- fitur yang akan di encoding:

- gender
- married
- dependents
- education
- self_employed
- property_type
- loan_status

```
# Mapping masing masing fitur
```

```
mapping_gender = {'Male': 0, 'Female': 1}
mapping_married = {'No': 0, 'Yes': 1}
mapping_dependents = {"0": 0, "1": 1, "2": 2, "3": 3}
mapping_education = {'Not Graduate': 0, 'Graduate': 1}
mapping_self_employed = {'No': 0, 'Yes': 1}
mapping_property_type = {'house': 0, 'studio': 1, "apartment": 2}
mapping_loan_status = {'REJECT': 0, 'APPROVE': 1}

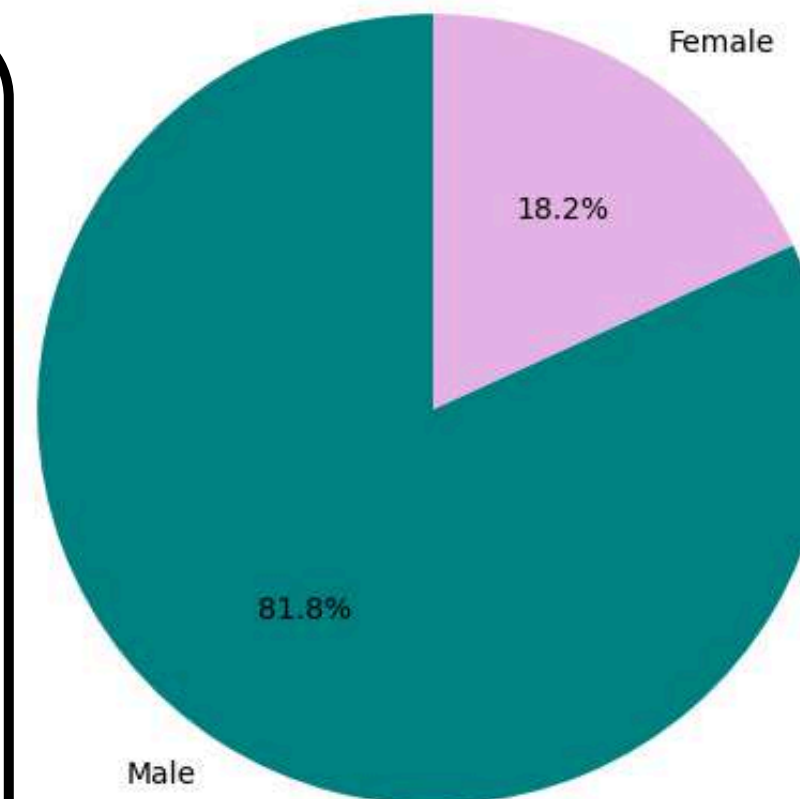
df['gender'] = df['gender'].map(mapping_gender)
df['married'] = df['married'].map(mapping_married)
df['dependents'] = df['dependents'].map(mapping_dependents)
df['education'] = df['education'].map(mapping_education)
df['self_employed'] = df['self_employed'].map(mapping_self_employed)
df['property_type'] = df['property_type'].map(mapping_property_type)
df['loan_status'] = df['loan_status'].map(mapping_loan_status)
```

	gender	married	dependents	education	self_employed	property_type	loan_status
0	0	0	0.0	1	0	0	1
1	0	1	1.0	1	0	1	0
2	0	1	0.0	1	1	0	1
3	0	1	0.0	0	0	0	1
4	0	0	0.0	1	0	0	1

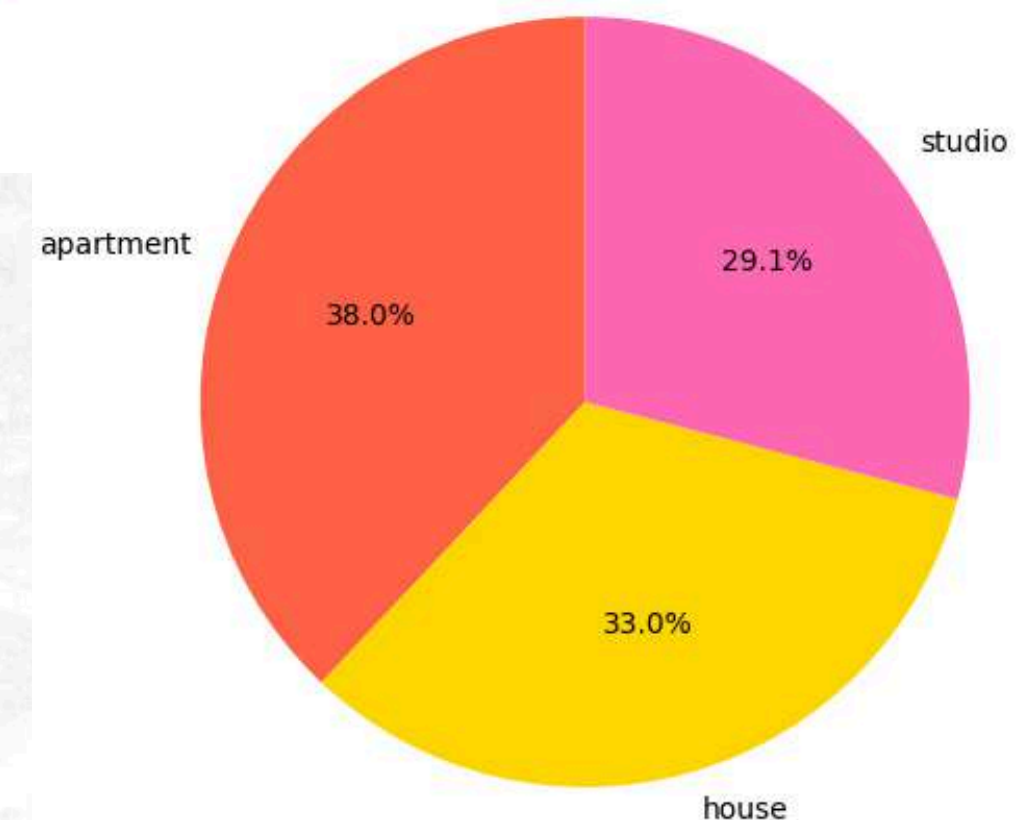
Insight

- Jumlah pelanggan **laki-laki** jauh lebih banyak dibandingkan dengan perempuan. Ini bisa mengindikasikan bahwa program pinjaman properti dari bank ini lebih diminati oleh laki-laki dibandingkan perempuan dikarenakan **kebutuhan laki-laki** lebih banyak dibandingkan perempuan
- Pelanggan yang melakukan pinjaman lebih banyak memilih properti **apartemen** sebesar 38%.

Gender Distribution

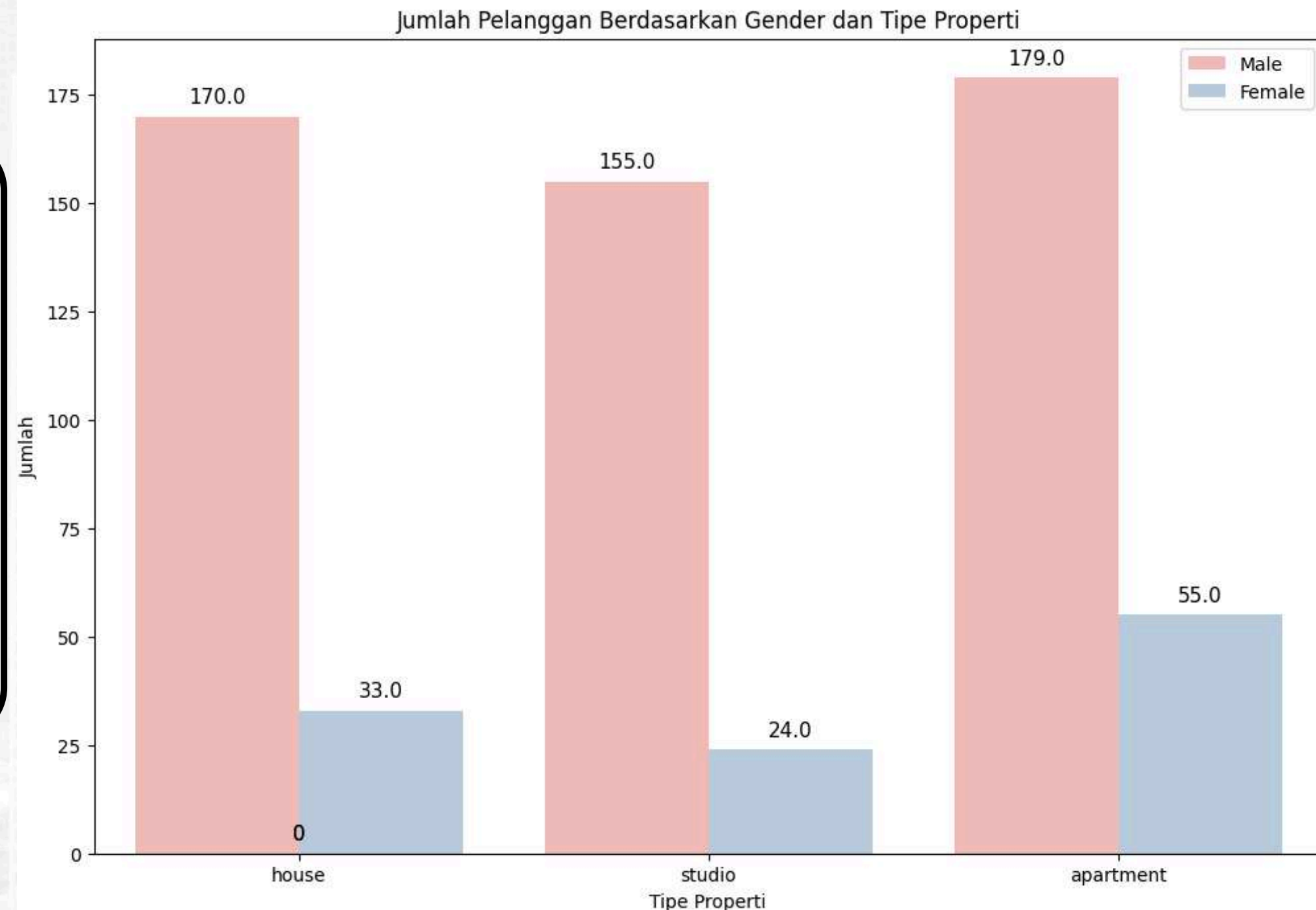


Property Type Distribution



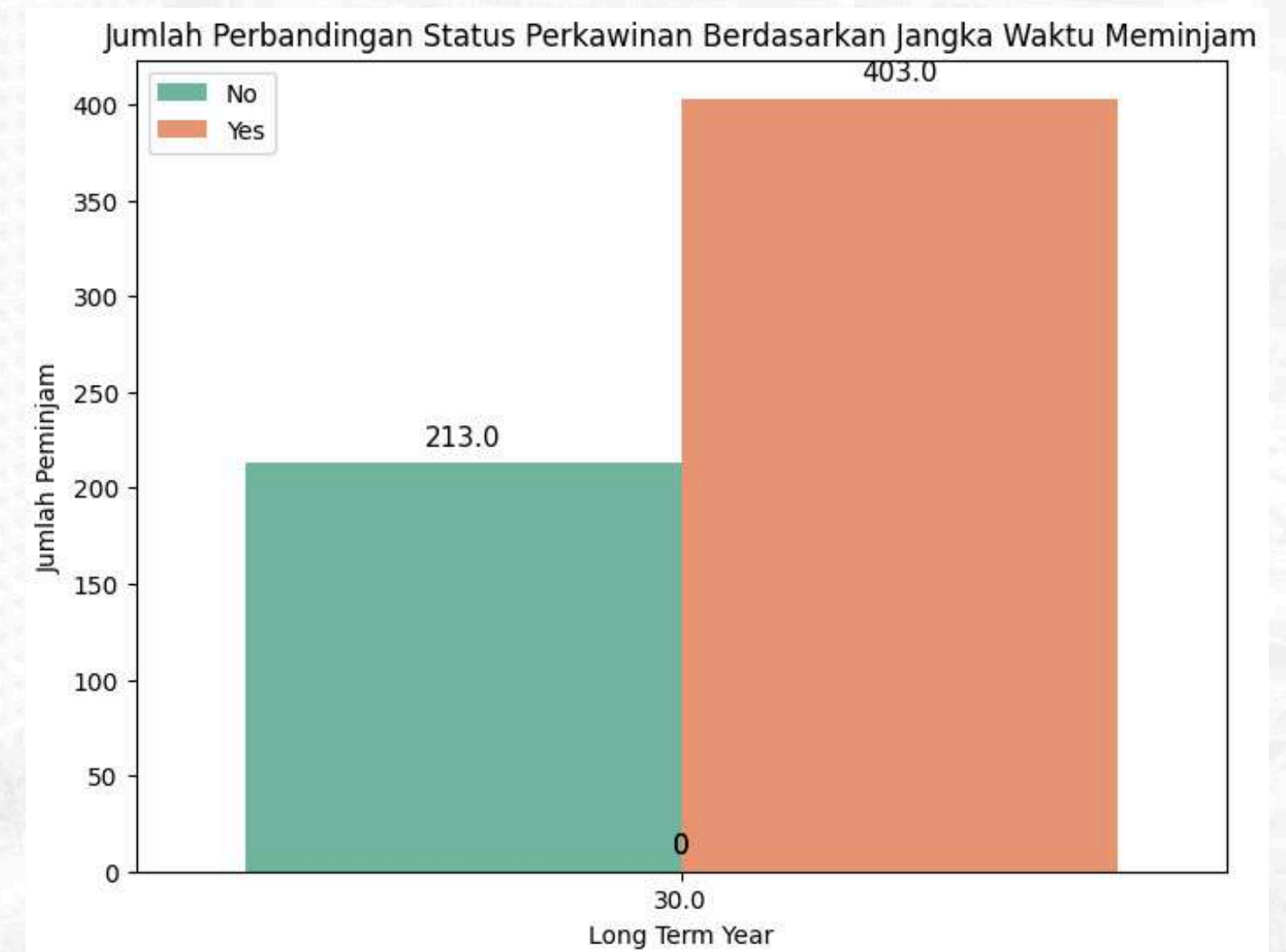
Insight

- Jenis properti yang paling diminati dari kedua gender adalah **apartement**
- Jenis properti **house** lebih diminati gender **perempuan** dibandingkan dengan laki - laki, begitupun sebaliknya untuk properti studio



Insight

Pada grafik barchart di bawah menunjukkan bahwa pelanggan yang sudah **menikah** lebih tertarik dalam melakukan **pinjaman**. Hal ini kemungkinan terjadi dikarenakan kebutuhan untuk **rumah tangga**. Perbedaan dalam jumlah peminjam antar keduanya cukup significant meskipun durasi peminjamannya sama.

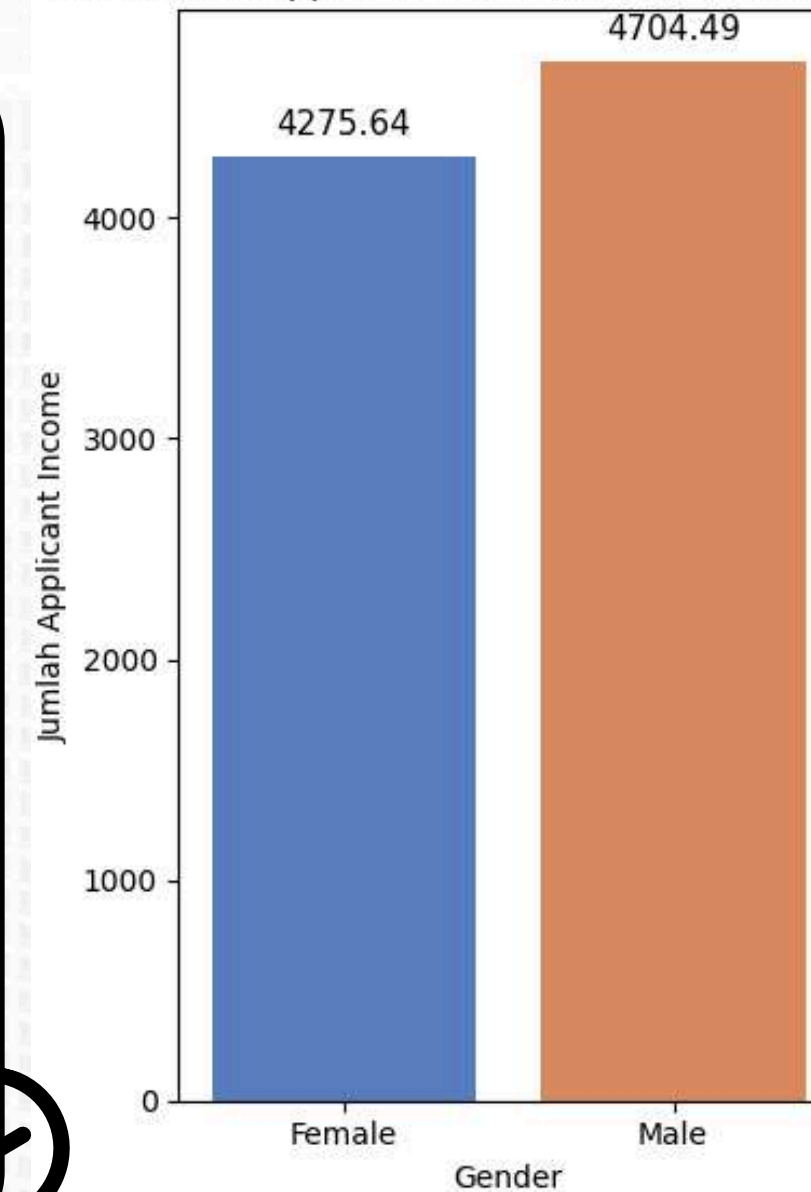


Rekomendasi

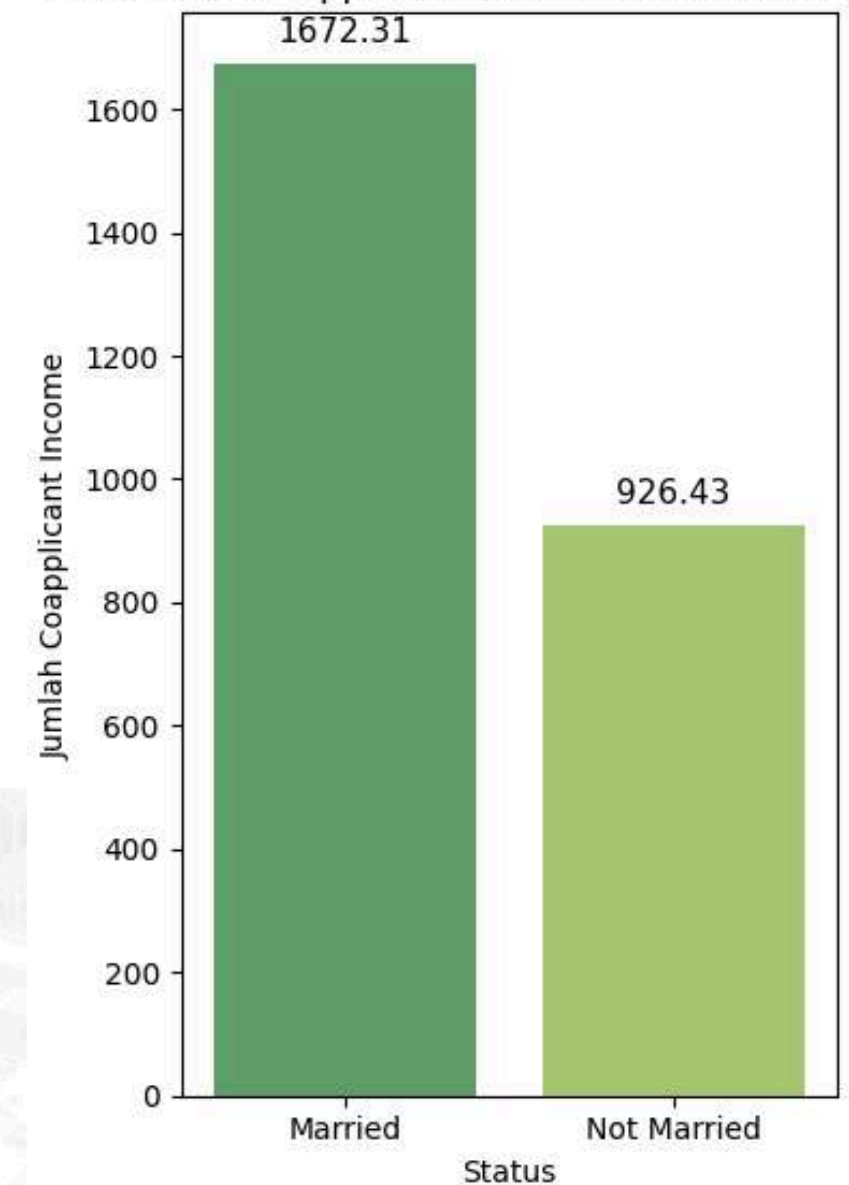
Memberikan pertimbangan yang lebih kepada nasabah **laki - laki** dikarenakan memiliki **applicant income** yang lebih baik dan juga kepada nasabah yang sudah **menikah** dikarenakan memiliki **coapplicant income** yang lebih besar. Sehingga peminjam **tidak terkendala** dalam melakukan pengembalian pinjamannya



Rata - rata Applicant Income Berdasarkan Gender

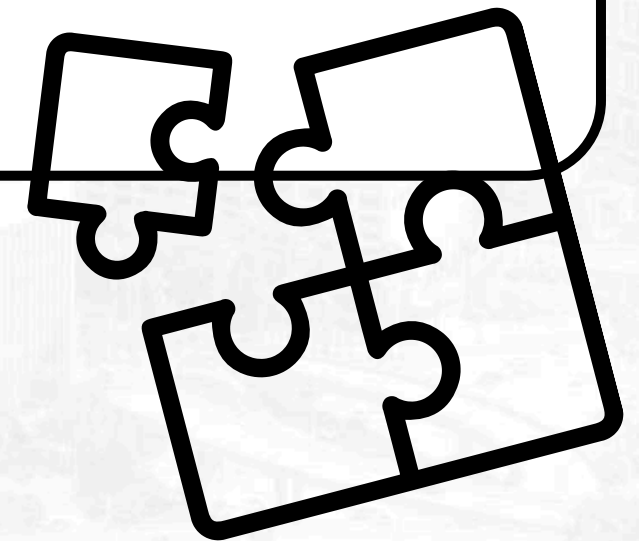


Rata - rata Coapplicant Income Berdasarkan Status



Kesimpulan

Secara keseluruhan, data menunjukkan bahwa aspek - aspek seperti **jenis kelamin** dan **status pernikahan** memiliki **dampak** pada preferensi dan ketertarikan individu terhadap **minat properti**. Selain itu, analisis data pinjaman memberikan perspektif atau wawasan tentang karakteristik peminjam dan faktor-faktor yang memengaruhi **keputusan** pemberian pinjaman.



The background of the slide is a faded, grayscale aerial photograph of a dense urban skyline with numerous skyscrapers and buildings.

TERIMA KASIH

