Anggota Kelompok 22:

- Made Baihaqi Aji Kumuda
- Putri Agustina Riadi
- Annisa Syifa Sugaryadi
- Kevin Avicenna Widiarto
- Indira Meutia Khairunnisa

PZSIB Data Analytics
Final Project Report

I. Business Understanding

1.1. Business Objectives

Bisnis adalah serangkaian usaha yang dilakukan individu atau kelompok dengan menawarkan barang dan jasa untuk mendapatkan keuntungan (laba). Arti bisnis juga bisa didefinisikan sebagai menyediakan barang dan jasa guna untuk kelancaran sistem perekonomian. Bisnis merupakan istilah umum yang menggambarkan semua aktivitas dan institusi yang memproduksi barang dan jasa dalam kehidupan sehari-hari. Kesimpulannya, pengertian bisnis memuat 4 aspek yakni, menghasilkan barang dan jasa, mendapatkan laba, suatu kegiatan usaha dan memenuhi kebutuhan masyarakat dalam sehari-hari.

Kredit merupakan fasilitas keuangan yang memungkinkan pelanggan meminjam uang dan membayarnya kembali dalam jangka waktu yang ditentukan. Kredit banyak digunakan untuk memenuhi kebutuhan sehari-hari dan menjadi bisnis yang menjanjikan di masa sekarang. Namun, sayangnya banyak oknum yang menyalahgunakan bisnis ini untuk peminjaman sepihak.

Home Credit merupakan perusahaan pembiayaan yang menyediakan pinjaman bagi pelanggan untuk berbelanja kebutuhan baik itu online maupun offline. Pada dasarnya, ditentukan beberapa kriteria untuk dapat menilai kelayakan peminjam. Dalam penentuannya, digunakan pendekatan statistik dan metode machine learning untuk menilai kemampuan calon peminjam.

Analisis menggunakan data riwayat perilaku calon peminjam seperti riwayat pinjaman, ketepatan waktu membayar tagihan, jumlah pemasukan, jumlah tanggungan, hingga kepemilikan aset. Analisis ini dilakukan untuk menilai kelayakan dan kemampuan calon peminjam sehingga dapat meminimalisir pelanggan yang disetujui tapi sebenarnya tidak layak.

Adapun kriteria kesuksesan bisnis dari kelompok kami yaitu dapat meningkatkan kepercayaan customer terhadap bisnis kami, dapat memberikan customer service yang baik, dapat mendapatkan metode dengan nilai akurasi terbaik untuk menentukan customer yang layak menggunakan pinjaman dari home credit.

1.2. Situation Assesment

Data yang digunakan untuk analisis adalah data riwayat perilaku calon peminjam yang terbagi menjadi beberapa *dataset* berbentuk csv yang diakses pada laman website Kaggle.com, yaitu https://www.kaggle.com/competitions/home-credit-default-risk/overview. Pada analisis digunakan dua istilah yang menjelaskan kondisi calon peminjam, yaitu *defaulted customer* yang digunakan untuk pelanggan yang tidak layak dan *not defaulted customer* yang digunakan untuk pelanggan yang mampu dan layak.

1.3. Data Mining Goals

Tujuan dilakukannya analisis ini adalah untuk mengelompokan jenis *customer* menjadi *defaulted customer* (peminjam tidak layak) dan *not defaulted customer* (peminjam layak). Selain itu, tujuan dari analisis ini adalah untuk dapat membentuk model terbaik dengan nilai akurasi tertinggi dan kecenderungan eror terendah. Model digunakan untuk menentukan kelayakan customer berdasarkan kriteria yang telah ditentukan. Serta dengan model ini harapan kami dapat menumbuhkan kembali kepercayaan masyarakat kepada bisnis home kredit.

1.4. Project Plan

Pada proses analisis menggunakan framework CRISP-DM, dilakukan beberapa tahapan, yaitu pemahaman bisnis (*business understanding*), pemahaman data (*data understanding*), persiapan data (*data preparation*), pemodelan (*modeling*), evaluasi model

(*evaluation*), dan penyajian hasil (*deployment*). Dalam pelaksanaannya, berikut adalah perencanaan projeknya:

II. Data Understanding

2.1. Initial Data

Adapun data yang akan digunakan untuk mencapai tujuan analisis ini yaitu sebagai berikut:

- Application_Train
- Application_Test
- Bureau
- Previous Application

2.2. Data Description

- **Application Train dan Application Test**

  Merupakan data pelatihan dan pengujian utama dengan informasi tentang setiap aplikasi pinjaman di Home Credit. ·Data aplikasi pelatihan dilengkapi dengan TARGET yang menunjukkan 0: pinjaman mudah dilunasi atau 1: pinjaman sulit dilunasi.

Adapun variable yang terdapat di dalam data ini totalnya mencapai 122 variabel. 20 diantaranya dapat dilihat dibawah ini

No	variabel	Deskripsi
1.	SK_ID_CURR	ID peminjam
2.	TARGET	Variable target (1=pelanggan dengan pembayaran yang sulit, 0=pelanggan dengan pembayaran yang mudah)
3.	NAME_CONTRACT_TYPE	Identifikasi apakah pinjaman itu tunai atau bergulir
4.	CODE_GENDER	Gender pelanggan
5.	FLAG_OWN_CAR	Tandai jika klien memiliki mobil
6.	FLAG_OWN_REALTY	Tandai jika klien memiliki rumah atau apartemen
7.	CNT_CHILDREN	Banyak anak yang dimiliki pelanggan
8.	AMT_INCOME_TOTAL	Penghasilan pelanggan
9.	AMT_CREDIT	Jumlah kredit dari pinjaman
10.	AMT_ANNUITY	Anuitas pinjaman
11.	AMT_GOODS_PRICE	Untuk pinjaman konsumen, itu adalah harga barang yang diberikan pinjaman
12.	DAYS_BIRTH	Usia klien dalam hari pada saat pengajuan
13.	DAYS_EMPLOYED	Berapa hari sebelum pengajuan pinjaman orang tersebut memulai pekerjaan saat ini
14.	EXT_SOURCE	Skor yang dinormalisasi dari sumber data eksternal
15.	NAME_EDUCATION_TYPE	Tingkat Pendidikan tertinggi yang dimiliki pelanggan
16.	DAYS_REGISTRATION	Berapa hari sebelum pengajuan klien mengubah pendaftarannya
17.	NAME_FAMILY_STATUS	Status keluarga pelanggan
18.	NAME_HOUSING_TYPE	Bagaimana situasi perumahan klien (sewa, tinggal bersama orang tua, ...)
19.	REGION_POPULATION_RELATIVE	Populasi wilayah tempat tinggal klien yang dinormalisasi (angka yang lebih tinggi berarti klien tinggal di wilayah yang lebih padat penduduknya)
20.	ORGANIZATION_TYPE	Jenis organisasi tempat klien bekerja

- **Bureau**

Berisi data mengenai kredit klien sebelumnya dari lembaga keuangan lain. Setiap kredit sebelumnya memiliki barisnya tersendiri di biro, tetapi satu pinjaman dalam data aplikasi dapat memiliki beberapa kredit sebelumnya. Pada data ini terdapat 19 variabel

No	variabel	Deskripsi
1.	SK_ID_CURR	ID pinjaman - satu pinjaman dalam sampel dapat memiliki 0,1,2 atau lebih kredit terkait sebelumnya di biro kredit
2.	SK_BUREAU_ID	ID yang dikodekan ulang dari kredit Biro Kredit sebelumnya yang terkait dengan pinjaman kami (kode unik untuk setiap aplikasi pinjaman)
3.	CREDIT_ACTIVE	Status Credit Bureau (CB) laporan kredit
4.	CREDIT_CURRENCY	Recoded mata uang kredit Biro Kredit
5.	DAYS_CREDIT	Berapa hari sebelum pengajuan saat ini klien mengajukan kredit Biro Kredit
6.	CREDIT_DAY_OVERDUE	Jumlah hari lewat jatuh tempo kredit CB pada saat pengajuan pinjaman terkait dalam sampel kami
7.	DAYS_CREDIT_ENDDATE	Sisa durasi kredit CB (dalam hari) pada saat pengajuan di Home Credit
8.	DAYS_ENDDATE_FACT	Hari sejak kredit CB berakhir pada saat aplikasi di Home Credit (hanya untuk kredit tertutup)
9.	AMT_CREDIT_MAX_OVERDUE	Jumlah maksimum tunggakan kredit Biro Kredit sejauh ini (pada tanggal permohonan pinjaman dalam sampel kami)
10.	CNT_CREDIT_PROLONG	Berapa kali kredit Biro Kredit diperpanjang
11.	AMT_CREDIT_SUM	Jumlah kredit saat ini untuk kredit Biro Kredit
12.	AMT_CREDIT_SUM_DEBT	Hutang saat ini pada kredit Biro Kredit
13.	AMT_CREDIT_SUM_LIMIT	Batas kredit kartu kredit saat ini dilaporkan di Biro Kredit
14.	AMT_CREDIT_SUM_OVERDUE	Jumlah saat ini tunggakan kredit Biro Kredit
15.	CREDIT_TYPE	Jenis kredit Biro Kredit (Mobil, tunai,...)

16.	DAYS_CREDIT_UPDATE	Berapa hari sebelum aplikasi pinjaman, informasi terakhir tentang kredit Biro Kredit datang
17.	AMT_ANNUITY	Anuitas kredit Biro Kredit
18.	SK_BUREAU_ID	Rekode ID kredit Biro Kredit (kode unik untuk setiap aplikasi) - gunakan ini untuk bergabung ke tabel CREDIT_BUREAU
19.	MONTHS_BALANCE	Bulan saldo relatif terhadap tanggal aplikasi (-1 berarti tanggal saldo terbaru)

- **Previous Application**

Berisi data pengajuan pinjaman sebelumnya di Home Credit oleh klien yang memiliki pinjaman dalam data aplikasi. Setiap pinjaman yang ada saat ini di aplikasi data dapat memiliki beberapa pinjaman sebelumnya. Setiap pengajuan sebelumnya memiliki satu baris dan diidentifikasi oleh fitur SK_ID_PREV. Pada data ini terdapat 37 variabel diantaranya

No	variabel	Deskripsi
1.	SK_ID_PREV	ID kredit sebelumnya di Kredit rumah terkait dengan pinjaman dalam sampel
2.	SK_ID_CURR	ID pinjaman dalam sampel kami
3.	NAME_CONTRACT_TYPE	Jenis produk kontrak (Pinjaman tunai, pinjaman konsumen [POS] ,...) dari aplikasi sebelumnya
4.	AMT_ANNUITY	Anuitas aplikasi sebelumnya
5.	AMT_APPLICATION	Untuk berapa banyak kredit yang diminta klien pada aplikasi sebelumnya
6.	AMT_CREDIT	Jumlah kredit akhir pada aplikasi sebelumnya. Ini berbeda dari AMT_APPLICATION dengan cara AMT_APPLICATION adalah jumlah yang awalnya diajukan oleh klien, tetapi selama proses persetujuan kami dia dapat menerima jumlah yang berbeda - AMT_CREDIT
7.	AMT_DOWN_PAYMENT	Uang muka pada aplikasi sebelumnya

8.	AMT_GOODS_PRICE	Harga barang dari barang yang diminta klien (jika ada) pada aplikasi sebelumnya
9.	WEEKDAY_APPR_PROCESS_START	Pada hari apa klien mengajukan aplikasi sebelumnya
10.	HOUR_APPR_PROCESS_START	Kira-kira pada jam berapa klien melamar aplikasi sebelumnya
11.	FLAG_LAST_APPL_PER_CONTRACT	Tandai jika itu adalah aplikasi terakhir untuk kontrak sebelumnya. Terkadang karena kesalahan klien atau petugas kami, mungkin ada lebih banyak aplikasi untuk satu kontrak
12.	NFLAG_LAST_APPL_IN_DAY	Tandai jika aplikasi tersebut adalah aplikasi terakhir per hari dari klien. Terkadang klien mengajukan lebih banyak lamaran dalam sehari. Jarang juga bisa error di sistem kita yang satu aplikasi masuk database dua kali
13.	NFLAG_MICRO_CASH	Tandai Pinjaman keuangan mikro
14.	RATE_DOWN_PAYMENT	Tingkat uang muka dinormalisasi pada kredit sebelumnya
15.	RATE_INTEREST_PRIMARY	Suku bunga dinormalisasi pada kredit sebelumnya
16.	RATE_INTEREST_PRIVILEGED	Suku bunga dinormalisasi pada kredit sebelumnya
17.	NAME_CASH_LOAN_PURPOSE	Tujuan pinjaman tunai
18.	NAME_CONTRACT_STATUS	Status kontrak (disetujui, dibatalkan, ...) dari aplikasi sebelumnya
19.	DAYS_DECISION	Relatif terhadap aplikasi saat ini kapan keputusan tentang aplikasi sebelumnya dibuat

2.3. Explore Data

Explore data adalah proses eksplorasi data yang bertujuan untuk memahami isi dan komponen penyusun data. Adapun langkah langkah yang dilakukan dalam explore data adalah sebagai berikut:

- **Import data dan baca data**: dengan menggunakan pandas dengan task pd.read_csv

```
[ ]  #import and read data
     import pandas as pd
     from matplotlib import pyplot as plt
     import seaborn as sns



     app_train = pd.read_csv('application_train.csv')
     app_test = pd.read_csv('application_test.csv')
     bureau = pd.read_csv('bureau.csv')
     prev_app=pd.read_csv('previous_application.csv')
```

- **Tampilkan data** dengan task head(), info(), columns(), dan shape()

```
[ ] #show top 5 rows
    app_train.head()
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	...
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	...
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	...
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	...
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	...

5 rows × 122 columns

```
[ ] #show data info
    app_train.info()
```

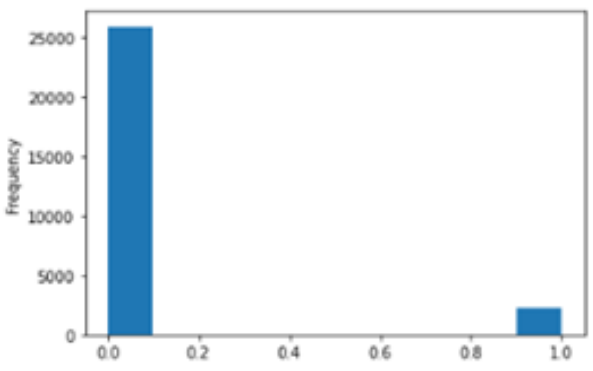
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

```
[ ] app_train.shape
    (307511, 122)

[ ] app_test.shape
    (48744, 121)

[ ] bureau.shape
    (1716428, 17)

[ ] prev_app.shape
    (1670214, 37)
```

Application Train

- **Periksa distribusi dari kolom target** dengan task df['TARGET'].value_counts() dan visualisasikan menggukanan plot histogram

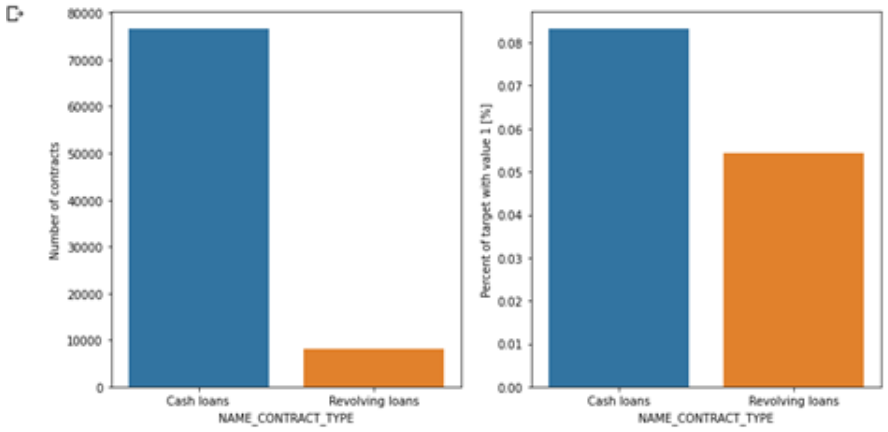
```
[ ]  app_train['TARGET'].astype(int).plot.hist();
```



- **Lihat jenis pinjaman yang diambil**

```
#lihat data app_train
#lihat jenis pinjaman yang diambil
def plot_stats(feature,label_rotation=False,horizontal_layout=True):
    temp = app_train[feature].value_counts()
    df1 = pd.DataFrame({feature: temp.index,'Number of contracts': temp.values})

    # Calculate the percentage of target=1 per category value
    cat_perc = app_train[[feature, 'TARGET']].groupby([feature],as_index=False).mean()
    cat_perc.sort_values(by='TARGET', ascending=False, inplace=True)

    if(horizontal_layout):
        fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,6))
    else:
        fig, (ax1, ax2) = plt.subplots(nrows=2, figsize=(12,14))
    sns.set_color_codes("pastel")
    s = sns.barplot(ax=ax1, x = feature, y="Number of contracts",data=df1)
    if(label_rotation):
        s.set_xticklabels(s.get_xticklabels(),rotation=90)

    s = sns.barplot(ax=ax2, x = feature, y='TARGET', order=cat_perc[feature], data=cat_perc)
    if(label_rotation):
        s.set_xticklabels(s.get_xticklabels(),rotation=90)
    plt.ylabel('Percent of target with value 1 [%]', fontsize=10)
    plt.tick_params(axis='both', which='major', labelsize=10)
```

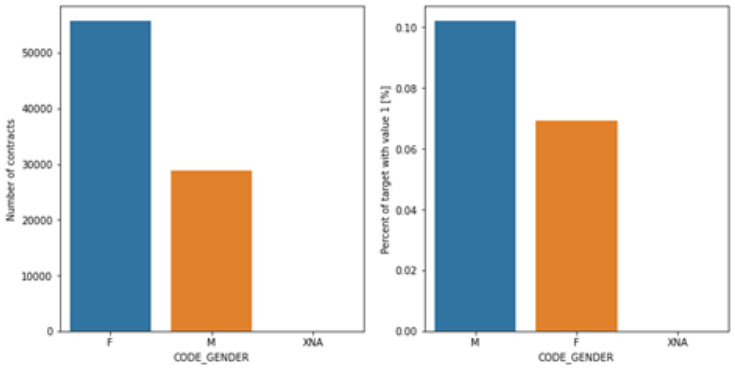
```
plot_stats('NAME_CONTRACT_TYPE')
```



Jenis kontrak Pinjaman bergulir hanyalah sebagian kecil (10%) dari jumlah total pinjaman; pada saat yang sama, sejumlah besar pinjaman Bergulir, dibandingkan dengan frekuensinya, tidak dilunasi.

- **Lihat jenis kelamin klien**

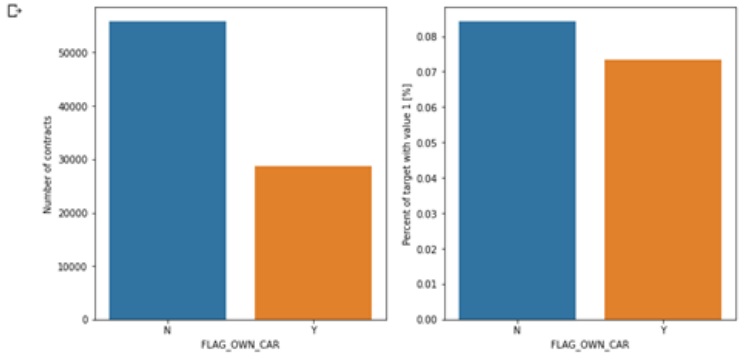
```
[12] plot_stats('CODE_GENDER')
```



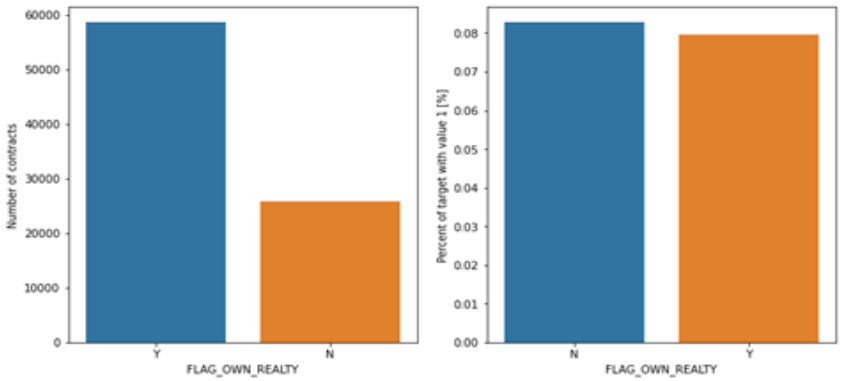
Jumlah klien wanita hampir dua kali lipat jumlah klien pria. Dilihat dari persentase kredit macet, laki-laki memiliki peluang lebih tinggi untuk tidak mengembalikan pinjaman mereka (~10%), dibandingkan dengan perempuan (~7%).

- **Lihat Flag yang memberi tahu kita jika klien memiliki mobil atau real estat**

```
plot_stats('FLAG_OWN_CAR')
plot_stats('FLAG_OWN_REALTY')
```

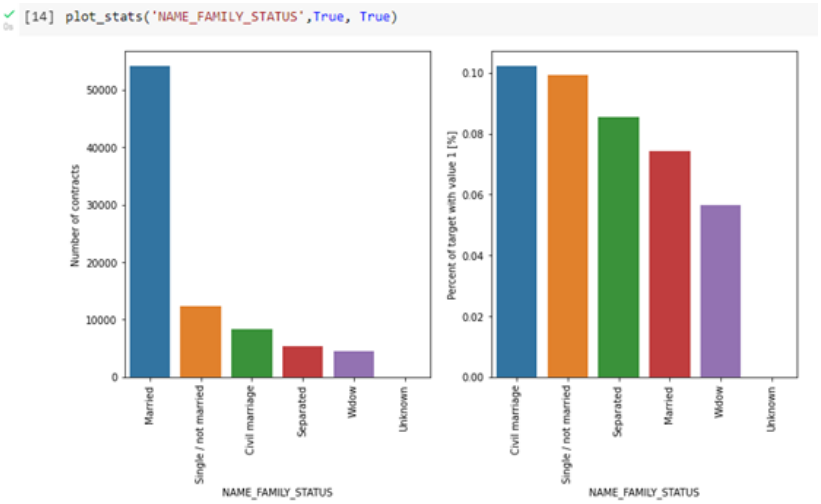


Klien yang memiliki mobil hampir setengah dari klien yang tidak memilikinya. Klien yang memiliki mobil cenderung tidak membayar kembali mobil yang dimilikinya. Kedua kategori tersebut memiliki tingkat tidak-pembayaran sekitar 8%.



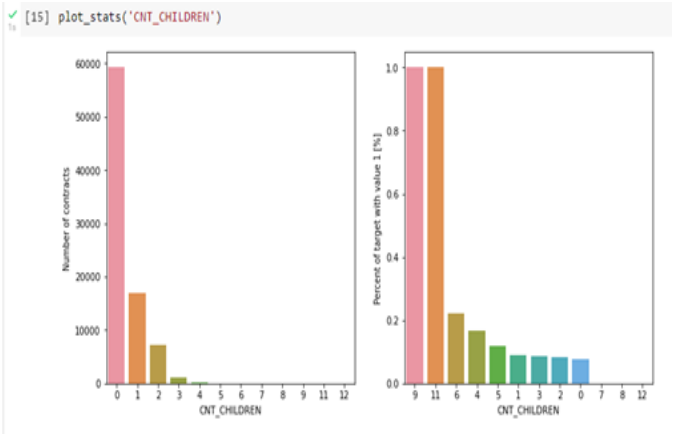
Klien yang memiliki real estat lebih dari dua kali lipat klien yang tidak memiliki. Kedua kategori (memiliki real estat atau tidak memiliki) memiliki tingkat tidak membayar kurang dari 8%.

- **Perhatikan status keluarga klien**



Sebagian besar klien sudah menikah, diikuti Single/belum menikah dan pernikahan sipil. Dalam hal persentase tidak dilunasi pinjaman, Perkawinan sipil memiliki persentase tidak dilunasi tertinggi (10%), dengan variable Janda yang terendah.

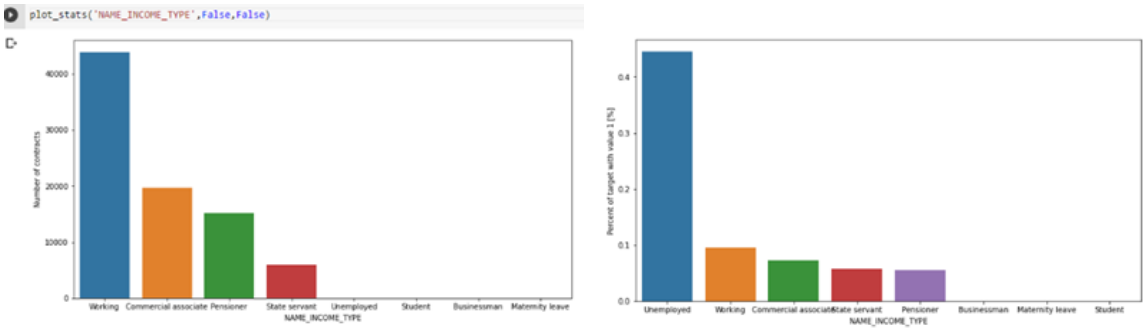
- **Perhatikan distribusi banyak anak dari klien**



Sebagian besar klien yang mengambil pinjaman tidak memiliki anak. Jumlah pinjaman yang terkait dengan klien dengan satu anak 4 kali lebih kecil, jumlah pinjaman yang terkait dengan klien dengan dua anak 8 kali lebih kecil; klien dengan 3 atau 4 anak lebih jarang.

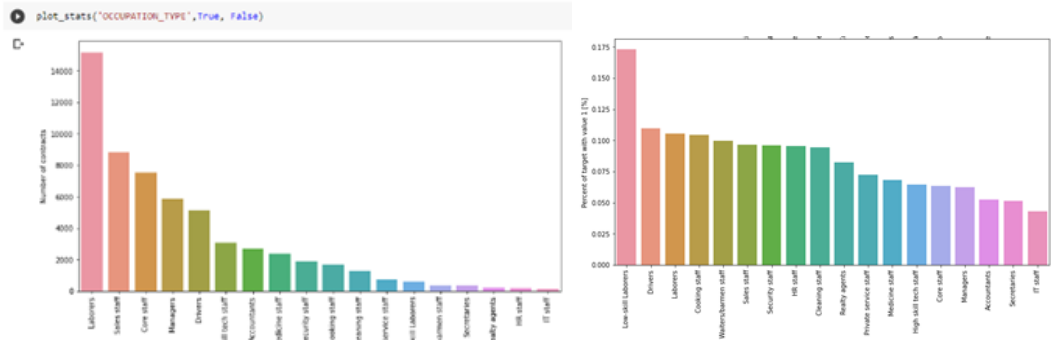
Untuk pelunasan, klien tanpa anak, 1, 2, 3, dan 5 anak memiliki persentase tidak melunasi rata-rata sekitar (10%). Klien dengan 4 dan 6 anak berada di atas rata-rata dalam hal persentase pinjaman yang tidak dibayar kembali (lebih dari 25% untuk keluarga dengan 6 anak). Sedangkan untuk klien dengan 9 atau 11 anak, persentase pinjaman yang tidak dilunasi adalah 100%.

- **Selidiki jumlah klien dengan jenis cara pendapatan yang berbeda.**



Sebagian besar pemohon pinjaman adalah yang berpenghasilan dari Bekerja, diikuti oleh Mitra Usaha, Pensiunan dan Pegawai Negeri Sipil. Pemohon dengan jenis penghasilan pengangguran berada di peringkat paling atas untuk kategori sulit mengembalikan pinjaman. Jenis pendapatan lainnya berada di bawah rata-rata 10% untuk yang sulit mengembalikan pinjaman.

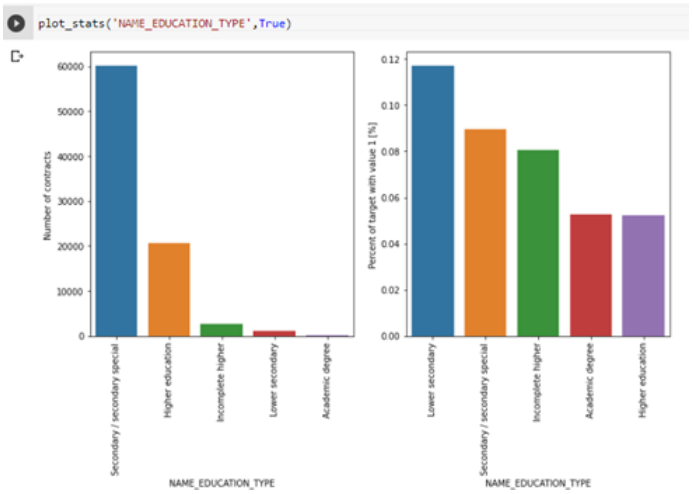
- **Pekerjaan klien**



Sebagian besar pinjaman diambil oleh Buruh, diikuti oleh staf Penjualan. Staf TI mengambil jumlah pinjaman terendah. Kategori dengan persentase pinjaman yang tidak dilunasi tertinggi adalah Tenaga Kerja Keterampilan Rendah (di atas 17%),

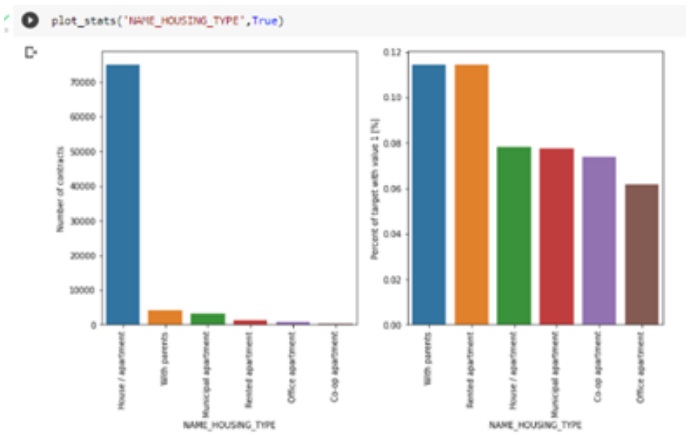
diikuti oleh staf Pengemudi dan Pelayan/barmen, staf Keamanan, Tenaga Kerja dan staf Memasak.

- **Tipe Pendidikan klien**



Mayoritas klien berpendidikan menengah/menengah khusus, diikuti oleh klien berpendidikan tinggi. Hanya sedikit sekali yang bergelar sarjana.Kategori menengah bawah, meskipun jarang, memiliki tingkat tidak mengembalikan pinjaman terbesar (11%). Orang-orang dengan gelar Akademik memiliki tingkat tidak membayar kurang dari 2%.

- **Tipe rumah klien**



Lebih dari 250.000 pemohon kredit mendaftarkan rumah mereka sebagai Rumah/apartemen. Kategori berikut memiliki jumlah klien yang sangat kecil (Dengan

orang tua, apartemen kota). Dari kategori ini, apartemen sewaan dan bersama orang tua memiliki tingkat tidak dapat dilunasi lebih dari 10%.

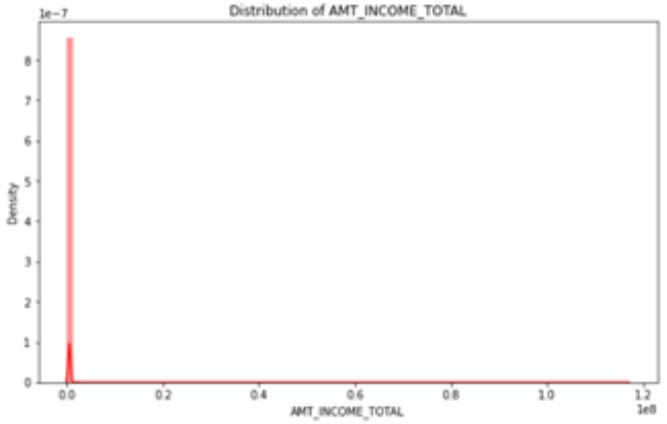
- **plot distribusi pendapatan total** untuk klien, kredit, anuitas, *good prices*, *days birth*, *days employed*, dan *days registration.*

```
[23] # Plot distribution of one feature
     def plot_distribution(feature,color):
         plt.figure(figsize=(10,6))
         plt.title("Distribution of %s" % feature)
         sns.distplot(app_train[feature].dropna(),color=color, kde=True,bins=100)
         plt.show()
```

```
[24] # Plot distribution of multiple features, with TARGET = 1/0 on the same graph
     def plot_distribution_comp(var,nrow=2):

         i = 0
         t1 = app_train.loc[app_train['TARGET'] != 0]
         t0 = app_train.loc[app_train['TARGET'] == 0]

         sns.set_style('whitegrid')
         plt.figure()
         fig, ax = plt.subplots(nrow,2,figsize=(12,6*nrow))

         for feature in var:
             i += 1
             plt.subplot(nrow,2,i)
             sns.kdeplot(t1[feature], bw=0.5,label="TARGET = 1")
             sns.kdeplot(t0[feature], bw=0.5,label="TARGET = 0")
             plt.ylabel('Density plot', fontsize=12)
             plt.xlabel(feature, fontsize=12)
             locs, labels = plt.xticks()
             plt.tick_params(axis='both', which='major', labelsize=12)
         plt.show();
```

```
plot_distribution('AMT_INCOME_TOTAL','red')
```

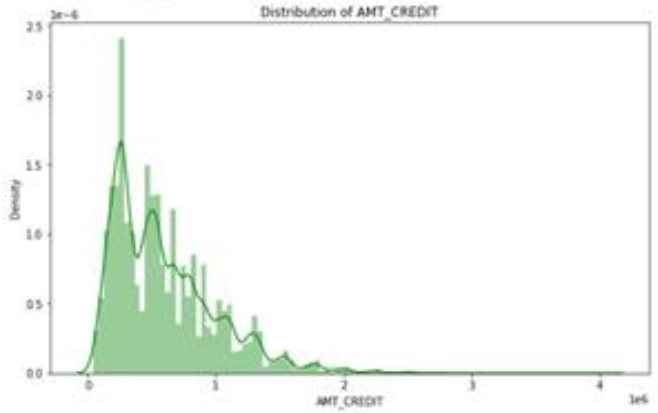
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarni
  warnings.warn(msg, FutureWarning)
```



Klien

```
[27] plot_distribution('AMT_CREDIT','green')
```

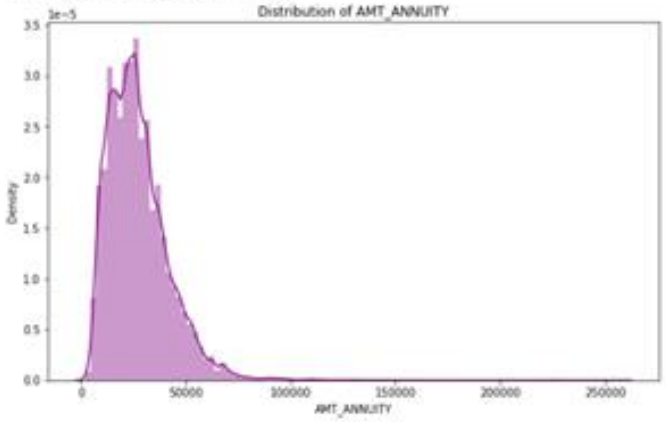
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
```



Kredit

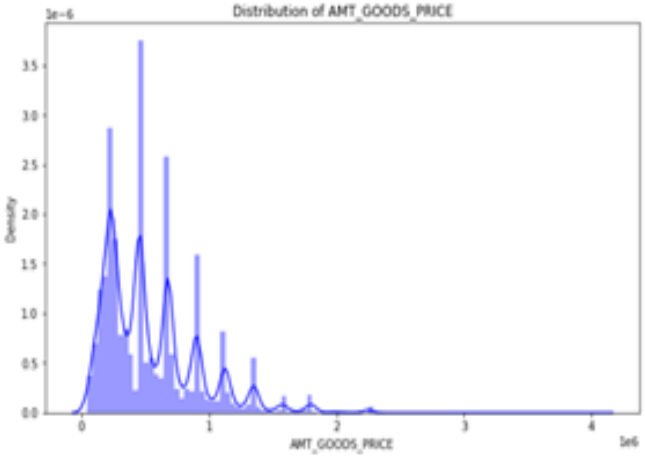
```
[28] plot_distribution('AMT_ANNUITY','purple')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarni
  warnings.warn(msg, FutureWarning)
```



Anuitas

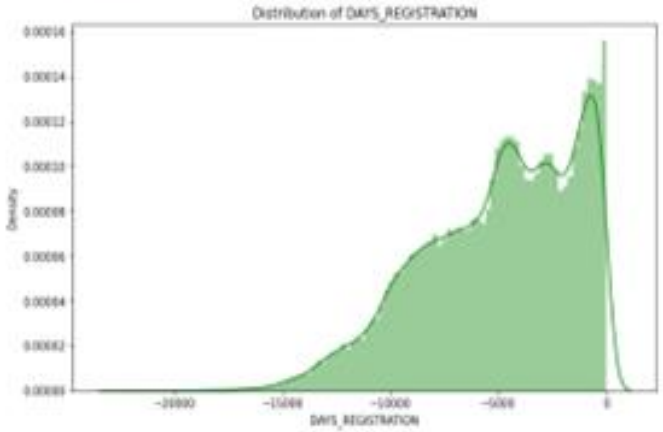
```
[30] plot_distribution('AMT_GOODS_PRICE','blue')
```



Good Price

```
[33] plot_distribution('DAYS_REGISTRATION','green')
```

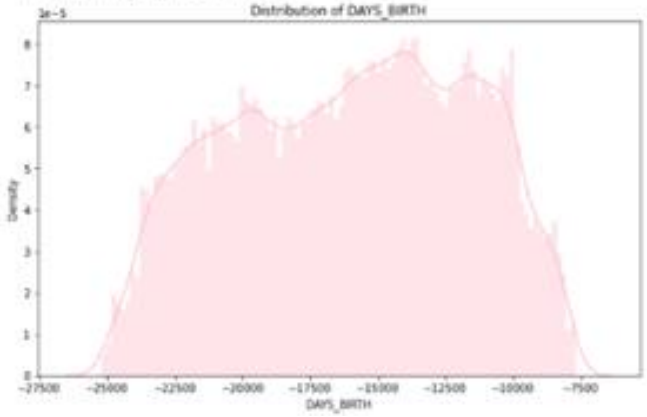
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```



Days Registration

```
plot_distribution('DAYS_BIRTH','pink')
```

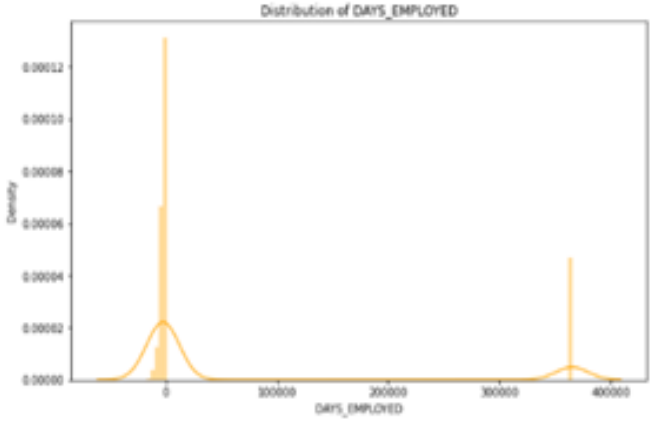
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarn
  warnings.warn(msg, FutureWarning)
```



Days of Birth

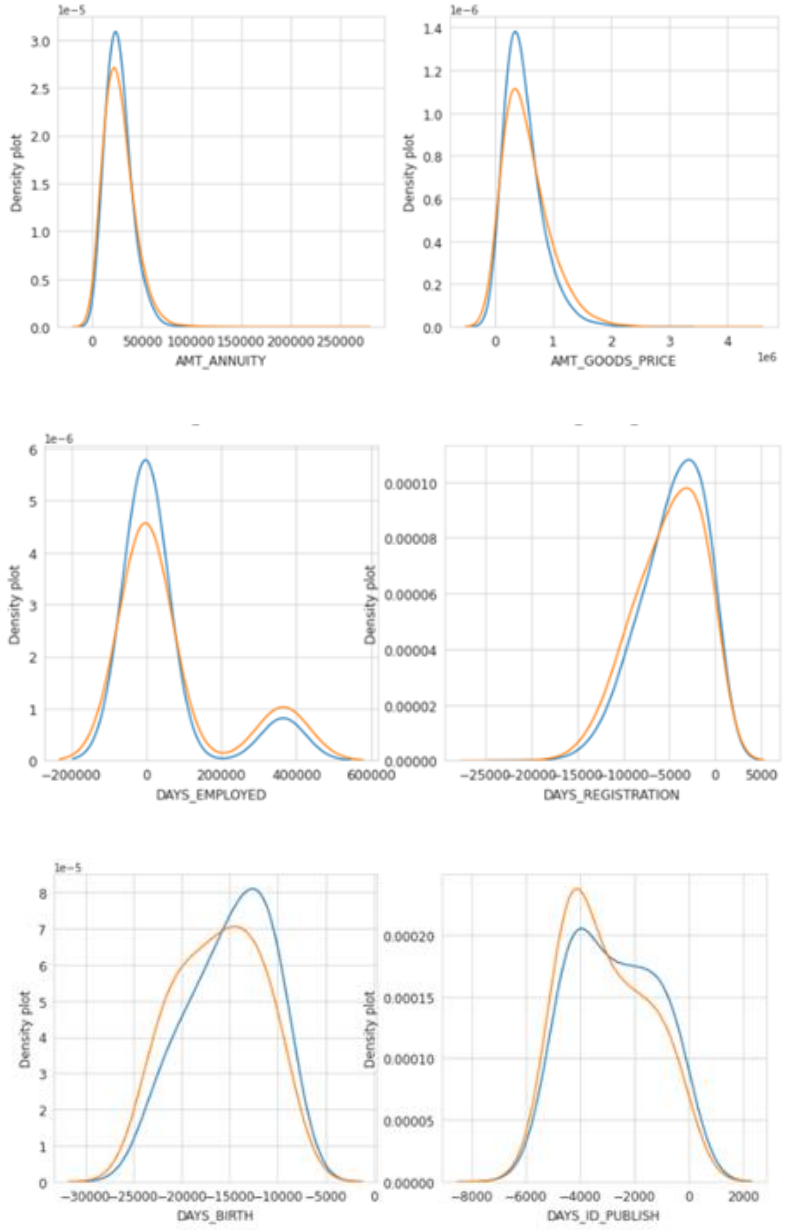
```
plot_distribution('DAYS_EMPLOYED','orange')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```



Days Employed

Lihatlah Perbandingan nilai interval dengan TARGET



- ***Check null value*** dengan menggunakan task df..isnull().sum() kemudian urutkan 20 teratas berdasarkan penurunan missing value dengan tabel

	Missing Values	% of Total Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4
NONLIVINGAPARTMENTS_AVG	213514	69.4
FONDKAPREMONT_MODE	210295	68.4
LIVINGAPARTMENTS_MODE	210199	68.4
LIVINGAPARTMENTS_MEDI	210199	68.4
LIVINGAPARTMENTS_AVG	210199	68.4
FLOORSMIN_MODE	208642	67.8
FLOORSMIN_MEDI	208642	67.8
FLOORSMIN_AVG	208642	67.8
YEARS_BUILD_MODE	204488	66.5
YEARS_BUILD_MEDI	204488	66.5
YEARS_BUILD_AVG	204488	66.5
OWN_CAR_AGE	202929	66.0
LANDAREA_AVG	182590	59.4
LANDAREA_MEDI	182590	59.4
LANDAREA_MODE	182590	59.4

2.4. Verifying Data Quality

Data quality terbagi dalam 6 dimensi:

- Akurasi: informasi yang diberikan data sudah realitas
- Kelengkapan: data yang disediakan Home Credit sudah lengkap
- Konsistensi: data tersebut konsisten
- Ketepatan waktu: Apakah informasi Anda tersedia saat Anda membutuhkannya?
- Validitas (alias Kesesuaian): Apakah informasi dalam format, jenis, atau ukuran tertentu? Apakah itu mengikuti aturan bisnis/praktik terbaik?=Jenis data yang diberikan dalam bentuk csv, format dan ukurannya sudah sesuai
- Integritas: kumpulan data yang berbeda digabungkan dengan benar untuk mencerminkan gambaran yang lebih besar. hubungan didefinisikan dan diimplementasikan dengan baik.

III. Data Preparation

3.1. Data Sets

Terdapat empat data set yang digunakan, yaitu data application_train, application_test, bureau, dan previous application

3.2. Data Selection

Tahap *selection data* dilakukan dengan mencari korelasi terlebih dahulu dan dijelaskan lebih lanjut didalam proses *modeling* dengan menggunakan random forest untuk menemukan *feature important* dari data

3.3. Cleaning Data

Adapun tahapan yang dilakukan dalam *cleansing* data adalah sebagai berikut:

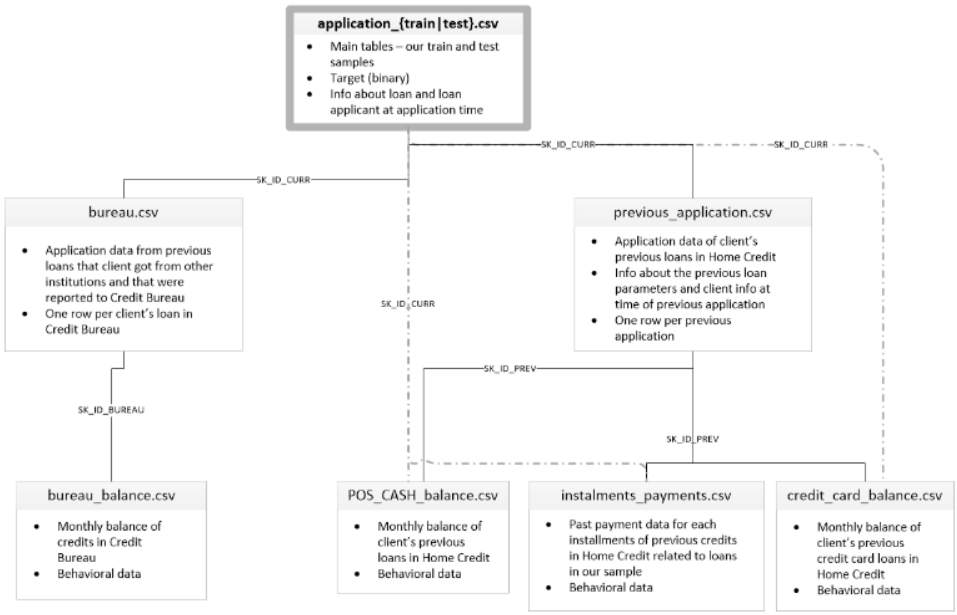
- Periksa missing value pada data dengan task df.isnull().sum()
- Drop columns yang memiliki missing value lebih dari 60% dengan task df = df.drop(columns=[])
- Fill missing value pada tabel yang masih diperlukan
- Encoding data, pada tahap ini kami menggunakan one hot encoding dengan task pd.get_dummies(df)

3.4. Construct Data

Setelah data melalui tahap *cleansing*, maka data diperiksa kembali dan dilakukan corelasi dengan task `df.corr()`

3.5. Integrating Data

Berikut deksripsi data dari Kaggle.com:



Merge data yaitu menggabungkan Data digunakan untuk menggabungkan dua dataset secara horizontal, berdasarkan nilai atribut yang dipilih (kolom). Dalam input widget Merge Data, diperlukan dua set data, data dan data ekstra. Adapun task yang digunakan kelompok kami yaitu Pandas Join 3 DataFrame

- Link google collab data understanding dan data preparation: https://colab.research.google.com/drive/132DOdHT-RkstZq97sjDkK3_i74BLczVE?usp=sharing
- EDA **https://colab.research.google.com/drive/1Y4xymSMtBcPDwR3DXqi7RhrNXPHm2_A6?usp=sharing**

IV. Modeling

4.1. Modeling Technique

Tujuan kita membuat modeling ialah untuk memprediksi menentukan pantas dan tidak pantas nya seseorang layak mendapatkan kredit. Maka teknik yang dipilih ialah *Random Forest Classifier* untuk membuat prediksi.

Random forest adalah teknik modeling yang menggunakan sekumpulan pohon keputusan (*decision trees*). *Random Forest* adalah algoritma dalam machine learning yang digunakan untuk pengklasifikasian data set dalam jumlah besar. Karena fungsinya bisa digunakan untuk banyak dimensi dengan berbagai skala dan performa yang tinggi.

Klasifikasi ini dilakukan melalui penggabungan tree dalam decision tree dengan cara training dataset yang Anda miliki.Setiap pohon keputusan dibangun menggunakan sebagian dari data training, sehingga pohon-pohon keputusan tersebut tidak saling bergantung satu sama lain. Target dari dari model ialah menentukan layak atau tidak layaknya seseorang mendapatkan pinjaman Home Credit.

4.2. Generate Test Design

- **Identifikasi tujuan**

  Tujuan dari pembuatan model ialah menentukan layak dan tidak layak nya seseorang dengan pengujian dengan berbagai feature

- **Metode Pengujian**

  Menggunakan 70.000 Sample dataset yang terdapat pada application_train dengan modeling Random forest classifier

- **Skenario**

  a. Menyiapkan data input Target

```
# Memisah Target dan meng-align data dari application_train, application_test
train_target_labels = application_train['TARGET']
application_train_align,application_test_align = application_train.align(application_test,join='inner',axis=1)
application_train_align['TARGET'] = train_target_labels
```

0.1s

  b. Split data Training dan data testing

```
# Menggunakan metode train test split untuk membagi data training dan data testing
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(train,train_target_labels,random_state=10)
```

1.8s

4.3. Building Model

- **Models & Parameter**

  a. *Random Forest Classifier*

```
# Modeling Random Forest dengan n_estimators 50 , random state 10, verbose 1 dan max_features 100
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=50,random_state=10,verbose=1,n_jobs=-1,max_features=100)
```

0.3s

  b. *Training data*

```
#Train data
rf.fit(X_train,y_train)
```

1m 20.2s

  c. Prediksi probabiltas

```
# Prediksi dengan probabilitas
rf_pred = rf.predict_proba(X_test)[:,1]
```

0.3s

d. Lalu menyimpan di sebuah dataframe

```
# Hasil akhir akhir
submit = pd.DataFrame()
tes = pd.DataFrame(X_test)
submit['SK_ID_CURR'] = tes.index
submit['TARGET'] = rf_pred
print(submit.head())
print(submit.shape)
```

✓ 0.5s

```
   SK_ID_CURR  TARGET
0           0    0.04
1           1    0.12
2           2    0.30
3           3    0.16
4           4    0.04
(76878, 2)
```

- **Model Description**

  Random forest adalah Teknik modeling yang menggunakan sekumpulan pohon keputusan (decision trees). Target dari dari model ialah menentukan pantas atau tidak dipantas nya seseorang mendapatkan kelayakan. Parameter yang digunakan n_estimator =50 , random_state=10 ,verbose=1,n_jobs=-1 dan menggunakan max_feature=100 untuk membatasi feature

4.4. Assess Model

Penilaian dari Model Random Forest Classifier

- Pertama menentukan seberapa akurat model tersebut, atau membandingkannya dengan model lain
- Lalu bisa menggunakan MAE,MSE, dan RMSE sebagai score performa model
- Selanjutnya, tentukan metrik yang akan digunakan untuk mengevaluasi model. Metrik ini bisa berupa akurasi, presisi, recall, atau metrik lain seperti contoh menggunakan confusion matrix dan classification report

```
train_target_labels.value_counts()
```
✔ 0.3s

```
TARGET
0        64373
1         5627
dtype: int64
```

```
# Membuat dataframe prediksi kita dan menyeleksi 20.000 baris
rf_pred = pd.DataFrame(rf_pred)
rf_pred = rf_pred.head(70000)
rf_pred = round(rf_pred)
rf_pred.value_counts()
```
✔ 0.5s

```
0.0    69867
1.0      133
dtype: int64
```

```
# Evaluation Using accuracy score ,
from sklearn import metrics as m
print('                 Evaluation Score               ')
print('------------------------------------------------')
print(f'Score Akurasi : {m.accuracy_score(train_target_labels,rf_pred)}')
print(f'Score MSE     : {m.mean_squared_error(train_target_labels,rf_pred)}')
print(f'Score MAE     : {m.mean_absolute_error(train_target_labels,rf_pred)}')
print(f'Score RMSE    : {np.sqrt(m.mean_squared_error(train_target_labels,rf_pred))}')
print(f'Nilai Confusion Matrix : {m.confusion_matrix(train_target_labels,rf_pred)}')
print(f'Classification Report :{m.classification_report(train_target_labels,rf_pred)})')

# print(m.r2_score(train_target_labels,rf_pred))
```

```
                 Evaluation Score
------------------------------------------------
Score Akurasi : 0.9179714285714285
Score MSE     : 0.08202857142857142
Score MAE     : 0.08202857142857142
Score RMSE    : 0.2864063047989192
Nilai Confusion Matrix : [[64249   124]
 [ 5618     9]]
Classification Report :              precision    recall  f1-score   support

           0       0.92      1.00      0.96     64373
           1       0.07      0.00      0.00      5627

    accuracy                           0.92     70000
   macro avg       0.49      0.50      0.48     70000
weighted avg       0.85      0.92      0.88     70000
)
```

Modeling Colab Page:
https://colab.research.google.com/drive/1GLEsC9PnqqhMlxadFFEq34S63FzMkLZk?usp=sharing

V. Evaluation

5.1. Evaluating Results

Berdasarkan 70.000 sampel dataset yang digunakan dari application_train, didapatkan nilai RMSE sebesar 0,286. Hal ini dapat diartikan bahwa model Random Forest yang terbentuk untuk melakukan klasifikasi pada data Home Credit memiliki nilai error yang kecil, sehingga dapat dikatakan model yang terbentuk cukup baik. Lalu, untuk memastikan lebih lanjut lagi, didapatkan nilai *Score Accuracy* sebesar 0,92 atau bisa disebut sebesar 92%. Nilai akurasi 92% ini dapat membuktikan bahwa ketepatan model *Random Forest* yang terbentuk mampu melakukan klasifikasi data Home Credit dengan baik, sehingga ditemukan *defaulted customer* (peminjam tidak layak) sebanyak 69.887 dan *non defaulted customer* (peminjam layak) sebanyak 113.

5.2. Process Review

Dalam melakukan analisis untuk mendapatkan hasil yang diinginkan, dilakukan beberapa tahap sebagai berikut:

- **Business Understanding**

  Pada tahap ini kami mendeksripsikan latar belakang masalah, bisnis objektif, tujuan bisnis, kriteria sukses dan plan dari project kami. Bisnis homekredit merupakan perusahaan yang menyediakan pinjaman bagi pelanggan untuk memenuhi kebutuhan hidup yang memiliki kesulitan dalam menentukan customer mana yang layak mendapatkan pinjaman kredit. Dataset yang digunakan diambil dari kaggle.com. Kriteria untuk dapat menilai kelayakan peminjam yaitu 0 (defaulted/tidak layak) dan 1 (not defaulted/layak). Dalam penentuannya, digunakan pendekatan statistik dan metode machine learning untuk menilai kemampuan calon peminjam. Planning project dilakukan agar dapat memanage waktu dengan baik serta memudahkan proses pengerjaan.

- **Data Understanding & Preparation**

  Pada tahap proses ini, data dideskripsikan agar mudah dipahami dan dikelola seperti di explor, cleansing dan merge sehingga siap digunakan pada tahap selanjutnya

- **Modeling**
- **Evaluation**
- **Deployment**

VI. Deployment

6.1. Deployment Plan

Pada deployment plan proyek kali ini memiliki rencana untuk menampilkan visulasisasi antara variabel top feature. Top feature variabel merupakan variabel yang memiliki pengaruh besar dalam skoring kredit untuk menilai kelayakan client dalam menerima kredit. Untuk top feature diambil dari 2 tabel yaitu “train” dan “brueu”. Pada tabel “train” kita mengambil variabel: “SK_ID_CURR”, “TARGET”, “AMT_INCOME_TOTAL”, “AMT_CREDIT”, “AMT_ANNUITY”, “AMT_GOODS_PRICE”, “DAYS_BIRTH”, “DAYS_EMPLOYED”, “DAYS_REGISTRATION”, “DAYS_ID_PUBLISH”, “EXT_SOURCE_1”, “EXT_SOURCE_2”, dan “EXT_SOURCE_3”. Pada tabel “bureau” mengambil nilai: “SK_ID_CURR”, “SK_ID_BUREAU”, “CREDIT_ACTIVE”, “CREDIT_DAY_OVERDUE”.

6.2. Plan Monitoring and Maintenance

Pada monitoring dan maintenance untuk dashboard kita menggunakan persektif dari pihak penyedia jasa kredit. Yang dimana data yang ditampilkan pada dashboard nanti memuat berbagai informasi (top feature) tentang client. Sehingga harapannya pihak penyedia jasa kredit menilai dan memutuskan clientnya akan menerima kredit atau tidaknya.

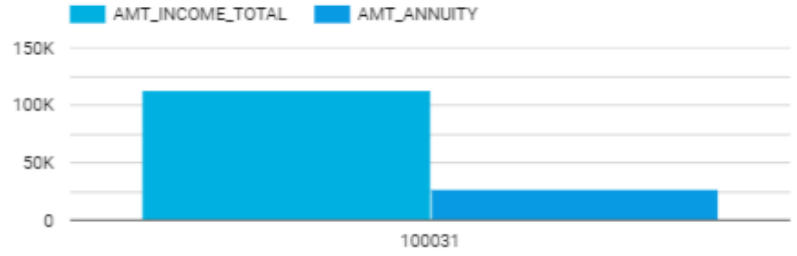
6.3. Final Report

- **Link Google Data Studio :** https://datastudio.google.com/reporting/7563a095-180b-48ef-abfc-d68fa81e9f81

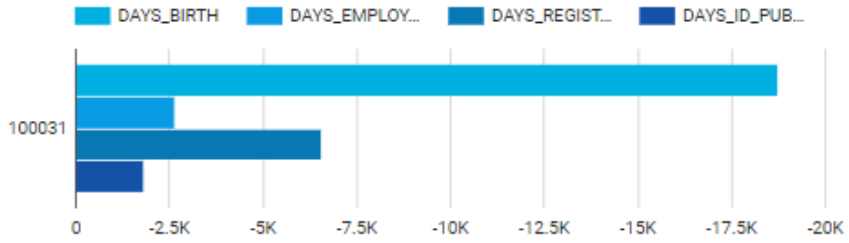
DASHBOARD HOME CREDIT DEFAULT RISK

ID Customer : 100,031

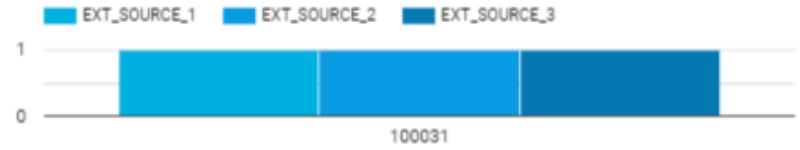
Perbandingan Pendapatan & Cicilan :

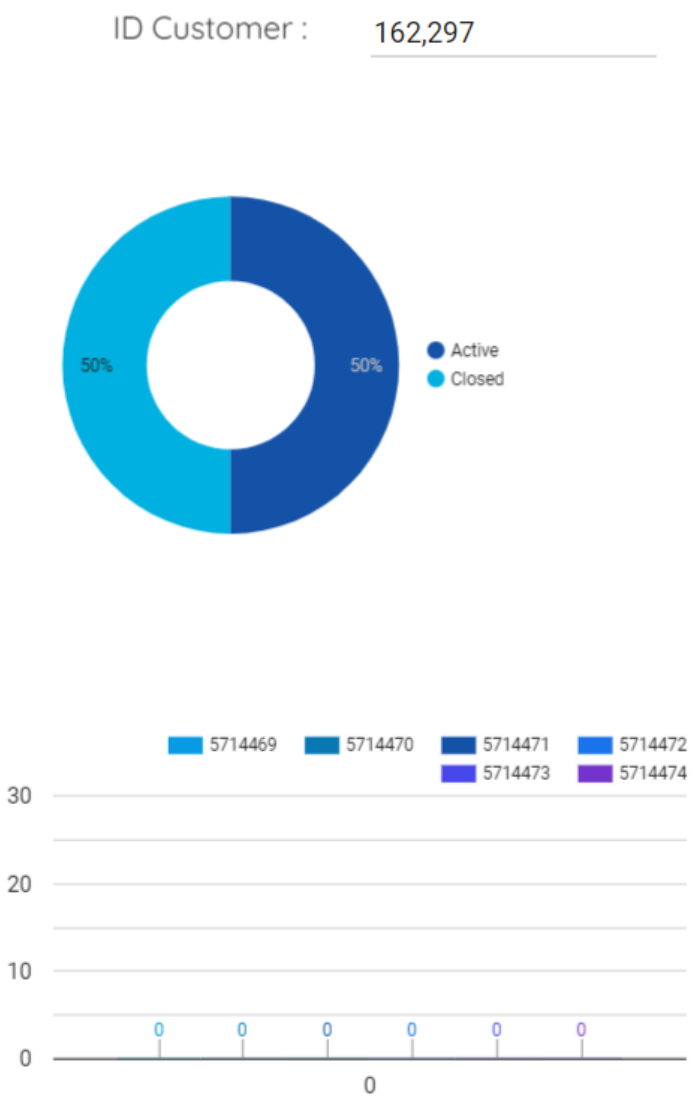


Perbandingan usia, jumlah hari kerja, registrasi, dan publikasi



Ketersediaan jumlah aset





6.4. Review Project

Hasil pembuatan dashboard dapat dibagi menjadi beberapa bagian untuk bisa dijelaskan secara proses seperti berikut :

- Flied ID Customer berfungsi untuk memanggil data client mana yang akan kita lihat dengan memanggil IDnya
- Perbandingan antara income total dan amt_annuity berfungsi untuk melihat apakah client sanggup tidaknya membayar cicilan berdasarkan pemasukannya
- Perbandingan usia, lama bekerja, dan lama registrasi menunjukan umur dan rekam jejak client dalam pengajuan kredit

- Ketersediaan jumlah aset menunjukan nilai aset yang dimiliki oleh client yang menjadi salah satu nilai plus dalam pengajuan kredit
- Penggunaan pie chart untuk menampilkan riwayat dari peminjaman client, dimana menjelaskan saat ini client sedang menerima banyaknya peminjaman atau tidaknya
- Penggunaan barchart selanjutnya berfungsi untuk melihat durasi berapa lamanya jika client menunggak kredit

```
# Evaluation Using accuracy score ,
from sklearn import metrics as m
print('                 Evaluation Score                ')
print('-------------------------------------------------')
print(f'Score Akurasi : {m.accuracy_score(train_target_labels,rf_pred)}')
print(f'Score MSE     : {m.mean_squared_error(train_target_labels,rf_pred)}')
print(f'Score MAE     : {m.mean_absolute_error(train_target_labels,rf_pred)}')
print(f'Score RMSE    : {np.sqrt(m.mean_squared_error(train_target_labels,rf_pred))}')
print(f'Nilai Confusion Matrix : {m.confusion_matrix(train_target_labels,rf_pred)}')
print(f'Classification Report :{m.classification_report(train_target_labels,rf_pred)})')

# print(m.r2_score(train_target_labels,rf_pred))
```

```
                 Evaluation Score
-------------------------------------------------
Score Akurasi : 0.9179714285714285
Score MSE     : 0.08202857142857142
Score MAE     : 0.08202857142857142
Score RMSE    : 0.2864063047989192
Nilai Confusion Matrix : [[64249   124]
 [ 5618     9]]
Classification Report :              precision    recall  f1-score   support

           0       0.92      1.00      0.96     64373
           1       0.07      0.00      0.00      5627

    accuracy                           0.92     70000
   macro avg       0.49      0.50      0.48     70000
weighted avg       0.85      0.92      0.88     70000
)
```

Link Publikasi/Deployment:

https://colab.research.google.com/drive/1GLEsC9PnqqhMlxadFFEq34S63FzMkLZk?usp=sharing