

# Loan Credit Risk Analysis

**St S Bintang Pratama Dumatubun**

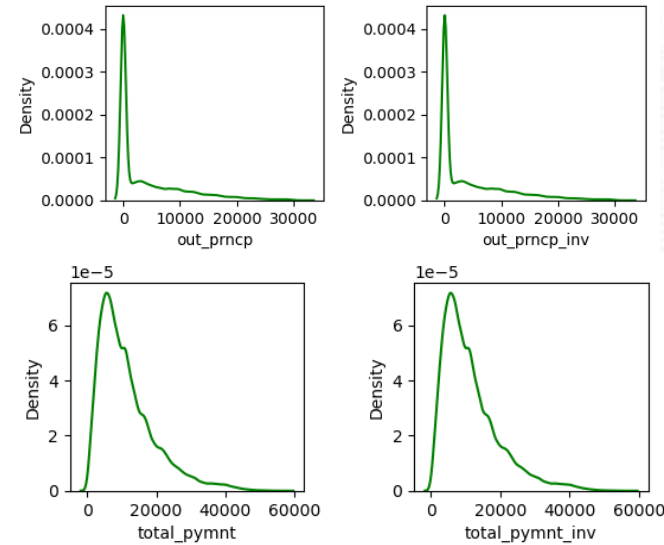
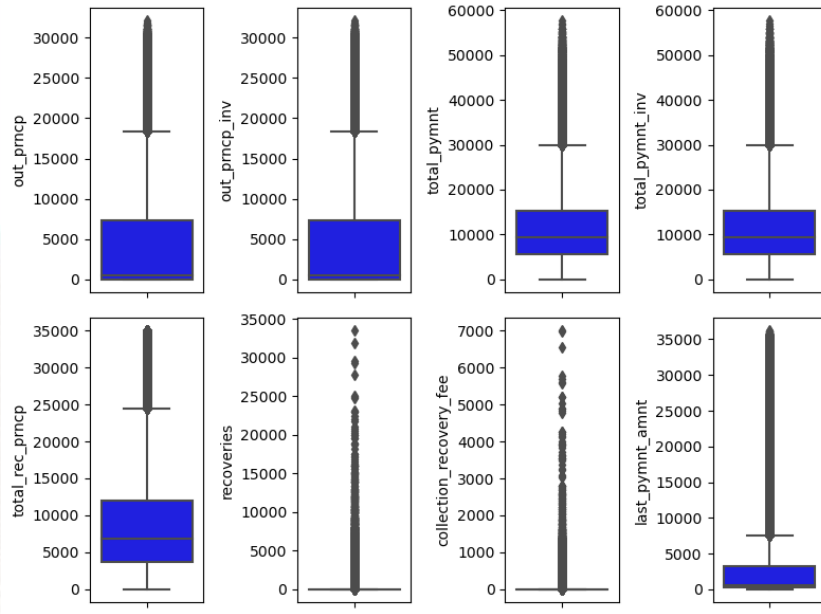
Dokumen Laporan  
Final Task Project  
Based Internship



# 1. Latar Belakang Masalah

Analisis risiko kredit merupakan bagian penting dalam industri perbankan dan keuangan untuk mengukur potensi kerugian yang mungkin terjadi akibat kredit yang gagal dilunasi oleh peminjam. Latar belakang masalah analisis risiko kredit meliputi beberapa poin penting meliputi **Pemberian Kredit, Risiko Kredit, Tujuan Analisis, Pendekatan Analisis,** dan **Manajemen Risiko**. Dalam konteks ini, analisis risiko kredit menjadi penting dalam mendukung pengambilan keputusan yang tepat dalam pemberian kredit, mengelola risiko yang terkait dengan portofolio kredit, dan memastikan kesehatan keuangan dan stabilitas institusi keuangan.

## 2. EDA & Insight

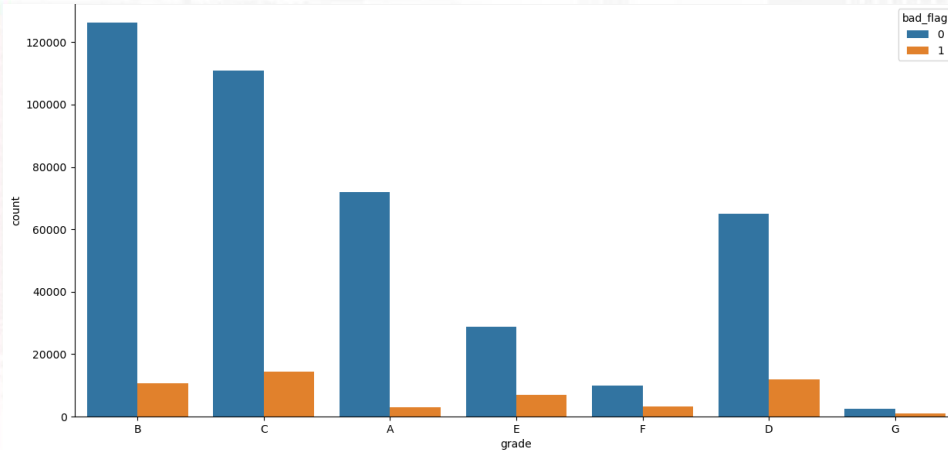


### Analysis

- Sisa pokok terutang untuk jumlah total yang didanai (out\_pnrcp) dan yang didanai oleh investor cenderung rendah
- Total pembayaran yang diterima dan total pembayaran yang diterima didanai oleh investor cukup rendah
- Setiap feature numerical memiliki outlier

## 2. EDA & Insight

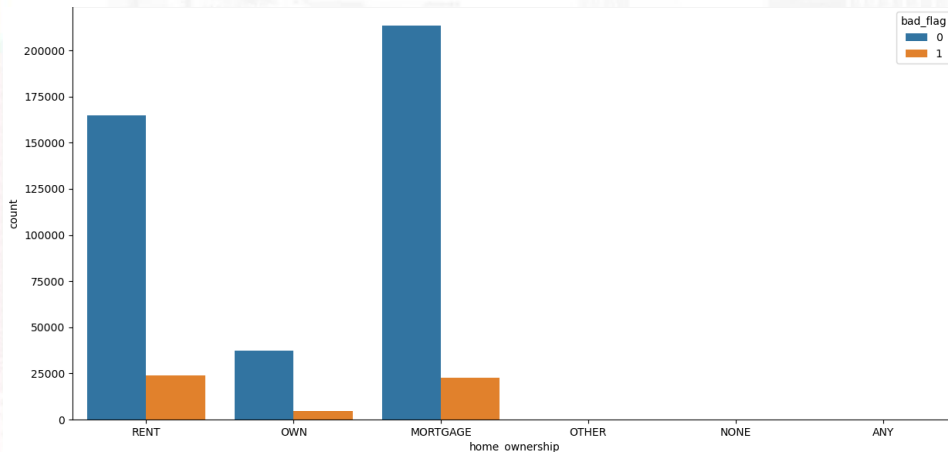
### Grade Based on Bad Flag



Individu dengan **Grade B** merupakan peminjam dengan **good flag** yang paling tinggi dibandingkan grade lainnya, disusul oleh **Grade C** dan **Grade A**. Sedangkan yang terendah ada pada **Grade G**. **Grade C** merupakan individu dengan **bad flag** terbanyak dibandingkan dengan grade lainnya bahkan lebih tinggi dari pada **Grade D**.

## 2. EDA & Insight

### Home Ownership Based on Bad Flag

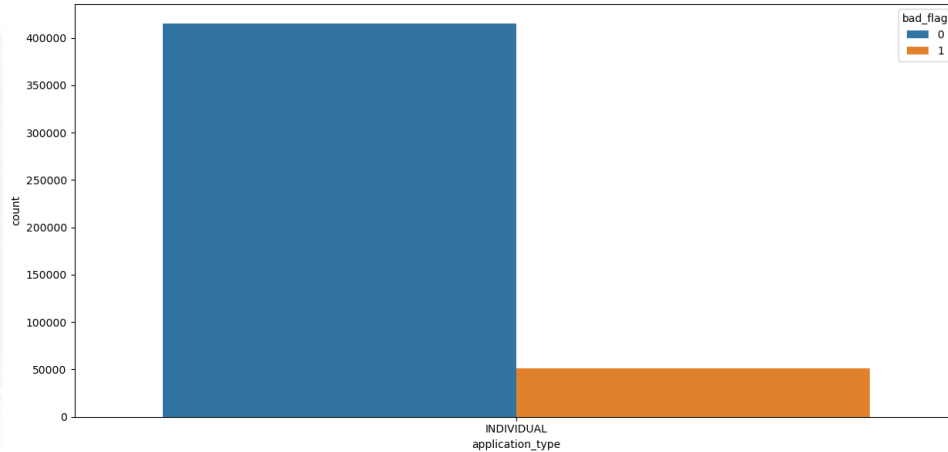


Individu yang **menyicil** rumah merupakan golongan **good flag** yang tertinggi dibandingkan yang lainnya, hal ini mungkin terjadi karena mereka memiliki kewajiban untuk menyicil rumah sehingga tidak ingin gagal bayar dan diberhentikan pinjamannya oleh bank yang menyebabkan mereka cenderung tidak pernah terlambat membayar sedangkan individu yang **memiliki rumah** sangat sedikit yang melakukan pinjaman kredit bahkan dari segi **good flag** dan **bad flag** merupakan yang terendah dibandingkan lainnya. Untuk **bad flag** tertinggi dimiliki oleh individu yang menyewa rumah hal ini mungkin terjadi karena sebagian dari mereka belum memiliki finansial yang stabil sehingga cenderung terjadi gagal bayar.



## 2. EDA & Insight

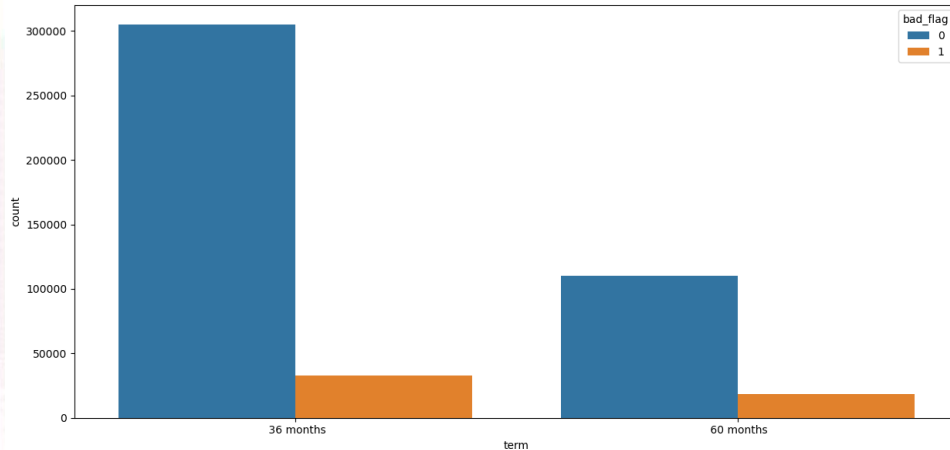
### Application Type Based on Bad Flag



Pada pengajuan pinjaman, hanya terdapat 1 kategori yaitu **perorangan**. Semua peminjam merupakan perorangan dan bukan dari suatu kelompok atau perusahaan. Peminjam perorangan ini sangat banyak yang sudah melakukan pelunasan pinjaman dan saat ini tidak memiliki tunggakan. Peminjam perorangan yang dianggap tidak mampu membayar juga terbilang cukup sedikit.

## 2. EDA & Insight

### Term Based on Bad Flag



Terdapat 2 kategori durasi pembayaran yaitu **3 tahun** dan **5 tahun**. Peminjam dengan durasi **3 tahun** terbilang sangat tertib dalam melakukan pelunasan karena banyak yang telah melunasi pinjaman dan saat ini tidak memiliki tunggakan apapun, akan tetapi peminjam dengan durasi pembayaran **3 tahun** cukup banyak yang dianggap tidak mampu membayar dibandingkan dengan peminjam dengan durasi pembayaran **5 tahun**.

## 3. Pre-processing

### Data Cleaning

- Handle Missing Value
- Handle Duplicated Data
- Handle Outlier

### Feature Engineering

- Feature Selection
- Feature Encoding
- Feature Transformation



# 3.1 Data Cleaning

## Handle Missing Value

### Handle Missing Value

Pada dataset yang diberikan terdapat banyak sekali fitur yang memiliki missing value, sehingga untuk melakukan pengisian data dilakukan dengan menggunakan function.

```
# membuat function fillna
def fill_column_missing_values(df, column_name, method='mean'):
    if method == 'mean':
        filled_column = df[column_name].fillna(df[column_name].mean())
    elif method == 'median':
        filled_column = df[column_name].fillna(df[column_name].median())
    elif method == 'mode':
        filled_column = df[column_name].fillna(df[column_name].mode().iloc[0])
    elif method == 'ffill':
        filled_column = df[column_name].fillna(method='ffill')
    elif method == 'bfill':
        filled_column = df[column_name].fillna(method='bfill')
    else:
        raise ValueError("Metode yang dimasukkan tidak valid. Pilihan: 'mean', 'median', 'mode', 'ffill', 'bfill'.")

    return filled_column
```

# 3.1 Data Cleaning

## Handle Missing Value

Berikut adalah fitur-fitur yang memiliki nilai kosong dan diisi sesuai dengan tipe data dan distribusi data.

```
df['emp_title'] = fill_column_missing_values(df, 'emp_title', method='mode')
df['emp_length'] = fill_column_missing_values(df, 'emp_length', method='mode')
df['title'] = fill_column_missing_values(df, 'title', method='mode')
df['delinq_2yrs'] = fill_column_missing_values(df, 'delinq_2yrs', method='mean')
df['earliest_cr_line'] = fill_column_missing_values(df, 'earliest_cr_line', method='mode')
df['inq_last_6mths'] = fill_column_missing_values(df, 'inq_last_6mths', method='mean')
df['mths_since_last_delinq'] = fill_column_missing_values(df, 'mths_since_last_delinq', method='median')
df['mths_since_last_record'] = fill_column_missing_values(df, 'mths_since_last_record', method='median')
df['open_acc'] = fill_column_missing_values(df, 'open_acc', method='median')
df['pub_rec'] = fill_column_missing_values(df, 'pub_rec', method='median')
df['revol_util'] = fill_column_missing_values(df, 'revol_util', method='mean')
df['total_acc'] = fill_column_missing_values(df, 'total_acc', method='median')
df['last_pymnt_d'] = fill_column_missing_values(df, 'last_pymnt_d', method='mode')
df['next_pymnt_d'] = fill_column_missing_values(df, 'next_pymnt_d', method='mode')
df['last_credit_pull_d'] = fill_column_missing_values(df, 'last_credit_pull_d', method='mode')
df['collections_12_mths_ex_med'] = fill_column_missing_values(df, 'collections_12_mths_ex_med', method='mean')
df['mths_since_last_major_derog'] = fill_column_missing_values(df, 'mths_since_last_major_derog', method='mean')
df['acc_now_delinq'] = fill_column_missing_values(df, 'acc_now_delinq', method='mean')
df['tot_coll_amt'] = fill_column_missing_values(df, 'tot_coll_amt', method='median')
df['tot_cur_bal'] = fill_column_missing_values(df, 'tot_cur_bal', method='mean')
df['total_rev_hi_lim'] = fill_column_missing_values(df, 'total_rev_hi_lim', method='mean')
```

# 3.1 Data Cleaning

## Handle Duplicated Rows

Pada pemeriksaan baris duplikat, ternyata tidak ada baris duplikat

```
# Pemeriksaan baris duplikat  
df.duplicated().sum()
```

0

# 3.1 Data Cleaning

## Handle Outlier

Pada fitur numerical hampir semua fiturnya memiliki outlier, mengingat distribusi masing-masing fitur yang *skew* sehingga digunakan metode IQR untuk menghapus outlier.

```
# Hitung Q1 dan Q3
Q1 = df1[['out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'recoveries', 'collection_recovery_fee', 'last_pymnt_amnt']]
Q3 = df1[['out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'recoveries', 'collection_recovery_fee', 'last_pymnt_amnt']]

# Hitung IQR
IQR = Q3 - Q1

# Hitung Batas Bawah dan Batas Atas
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Hapus outlier
df1_clean = df1[~((df1[['out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'recoveries', 'collection_recovery_fee', 'last_pymnt_amnt']] < lower_bound) | (df1[['out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'recoveries', 'collection_recovery_fee', 'last_pymnt_amnt']] > upper_bound)).any(axis=1)]
```

# 3.2 Feature Engineering

## Feature Selection

Berdasarkan matrix correlation, dipilih fitur-fitur yang dianggap relevan terhadap pembuatan model *machine learning* sehingga dilakukan penghapusan terhadap fitur-fitur yang dianggap tidak relevan untuk menghindari terjadinya *curse dimensionality*.

```
df3.drop(['id', 'member_id', 'sub_grade', 'emp_title', 'emp_length', 'issue_d', 'url', 'title', 'zip_code', 'addr_state', 'earliest_cr_line', 'revol_bal',  
         'revol_util', 'total_rec_int', 'total_rec_late_fee', 'last_pymnt_d', 'next_pymnt_d', 'collections_12_mths_ex_med', 'mths_since_last_major_derog',  
         'policy_code', 'application_type', 'acc_now_delinq', 'tot_coll_amt', 'tot_cur_bal', 'total_rev_hi_lim'], axis=1, inplace=True)
```

```
df3.drop('last_credit_pull_d', axis=1, inplace=True)
```



# 3.2 Feature Engineering

## Feature Encoding

Dilakukan dua metode untuk feature encoding yaitu label encoding dan one hot encoding.

Label encoding dilakukan pada

```
from sklearn.preprocessing import LabelEncoder

def label_encode_column(df, column_name):
    le = LabelEncoder()
    df[column_name] = le.fit_transform(df[column_name])
    return df

df_encoded = label_encode_column(df2, 'term')
df_encoded = label_encode_column(df2, 'grade')
df_encoded = label_encode_column(df2, 'verification_status')
df_encoded = label_encode_column(df2, 'pymnt_plan')
df_encoded = label_encode_column(df2, 'initial_list_status')
```

One hot encoding dilakukan pada

```
onehot = pd.get_dummies(df2['home_ownership'], prefix = 'home')
df2 = df2.join(onehot)
df2.drop('home_ownership', axis=1, inplace=True)

onehot1 = pd.get_dummies(df2['purpose'], prefix = 'pur')
df2 = df2.join(onehot1)
df2.drop('purpose', axis=1, inplace=True)
```

# 3.2 Feature Engineering

## Feature Transformation

Pada feature transformation digunakan standardisasi. Feature transformation hanya dilakukan pada fitur numerikal. Tidak dilakukan terhadap fitur kategorikal.

```
num = ['loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'int_rate', 'installment', 'annual_inc',  
       'dti', 'delinq_2yrs', 'inq_last_6mths', 'mths_since_last_delinq', 'mths_since_last_record', 'open_acc', 'pub_rec', 'total_acc',  
       'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'recoveries', 'collection_recovery_fee', 'last_pymnt_amnt']
```

```
from sklearn.preprocessing import StandardScaler  
ss = StandardScaler()
```

```
for col in num:  
    df_std[col] = ss.fit_transform(df_std[[col]])
```

```
display(df_std.shape, df_std.head(3))
```

```
(341207, 48)
```

## 4.1 Modelling - Pemilihan Metrik untuk Evaluasi

### Logistic Regression

```
Accuracy (Test Set): 0.98  
Precision (Test Set): 1.00  
Recall (Test Set): 0.73  
F1-Score (Test Set): 0.84  
roc_auc (test-proba): 0.94  
roc_auc (train-proba): 0.94
```

### Recall

**Metrik recall dipilih karena pada kasus ini, true positive rate menjadi prioritas dan meminimalisir nilai false negative yang besar**

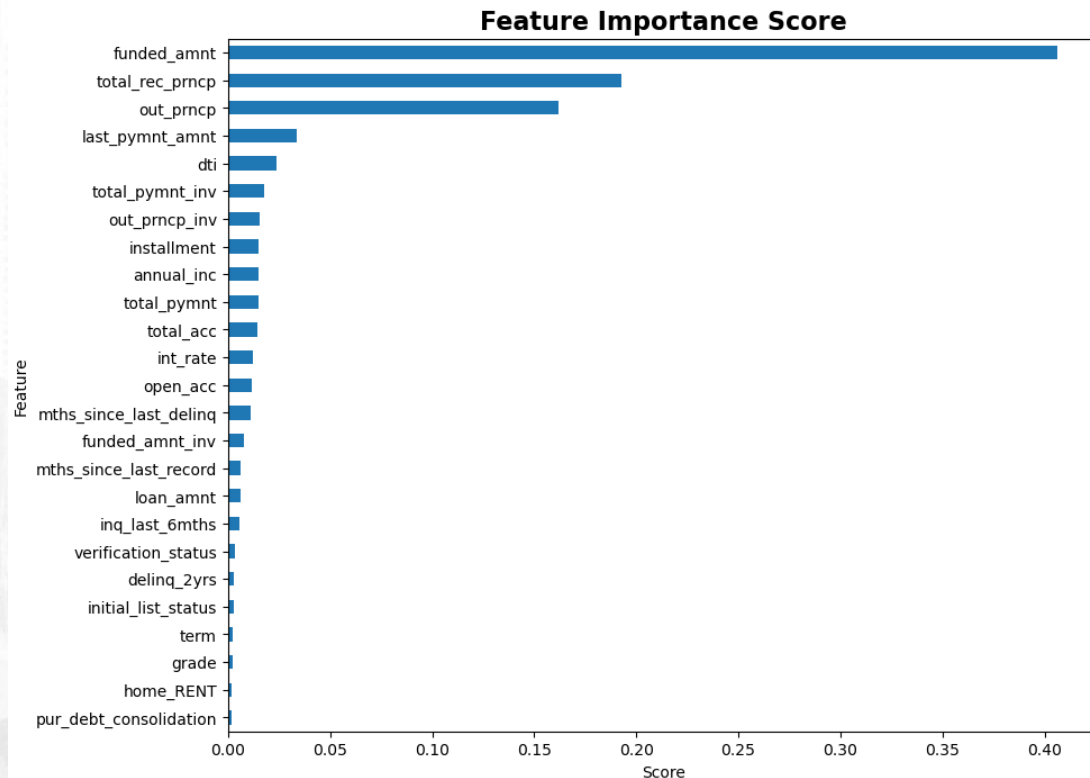
## Random Forest

```
Accuracy (Test Set): 0.94  
Precision (Test Set): 1.00  
Recall (Test Set): 0.22  
F1-Score (Test Set): 0.36  
roc_auc (test-proba): 0.87  
roc_auc (train-proba): 0.88
```

## Decision Tree

```
Accuracy (Test Set): 0.97  
Precision (Test Set): 0.75  
Recall (Test Set): 0.79  
F1-Score (Test Set): 0.77  
roc_auc (test-proba): 0.89  
roc_auc (train-proba): 1.00
```

# 4.2 Feature Importance





## 4.3 Summary

Berdasarkan model prediksi dan *feature importance* yang telah dibuat, disimpulkan bahwa algoritma yang sesuai yaitu Logistic Regression karena menghasilkan **recall** dan **F-1 Score** yang cukup tinggi, sedangkan pada algoritma lainnya metrik recall terbilang cukup rendah. Metrik recall digunakan karena kami berfokus pada **True Positive Rate** dan meminimalisir **False Negative** yang besar.

Pada *feature importance* ditetapkan threshold yaitu **0.05** sehingga fitur dengan score  $> 0.05$  dianggap sebagai fitur yang berpengaruh terhadap target. Sehingga dihasilkan top 3 *feature importance* yaitu funded amount, total rec prncp, dan out prncp.

## 4.4 Recommendations

### 1. Funded Amount (Jumlah Dana yang Diberikan)

- Perhatikan besarnya jumlah dana yang diberikan kepada peminjam pada saat pemberian pinjaman. Jumlah ini mencerminkan tingkat risiko yang diambil oleh pemberi pinjaman, karena semakin besar jumlah dana yang diberikan, semakin besar juga risiko kredit yang dihadapi.
- Evaluasi kembali kebijakan penetapan jumlah dana pinjaman untuk memastikan bahwa tingkat risiko kreditnya sesuai dengan toleransi risiko perusahaan.

### 2. Total Received Principal (Total Pokok yang Diterima)

- Monitor total pokok yang telah diterima dari peminjam hingga saat ini. Informasi ini membantu dalam memahami seberapa baik peminjam memenuhi kewajibannya dalam membayar kembali pinjaman.
- Analisis tren dalam total pokok yang diterima dapat memberikan wawasan tentang perilaku pembayaran peminjam dan kemungkinan risiko pembayaran yang mungkin timbul di masa depan.

### 3. Outstanding Principal (Pokok yang Masih Belum Lunas)

- Perhatikan jumlah pokok yang masih harus dilunasi oleh peminjam. Informasi ini penting untuk memahami tingkat kewajiban keuangan yang masih harus dihadapi oleh peminjam dan risiko kredit yang terkait.
- Evaluasi ulang proses penagihan dan kebijakan penanganan kredit untuk memastikan bahwa langkah-langkah yang tepat diambil untuk memastikan pengembalian pembayaran pokok yang masih belum lunas.

**TERIMA KASIH!**