Company Financial

Clara Sista Widhiastuti - 2206825782
Puti Raissa - 2206830391
Alden Luthfi - 2206028932
Muhammad Faishal Adly Nelwan - 2206030754

Overview

01

Data Description

02

Data Preprocessing

03

EDA

04

Model

Data Description

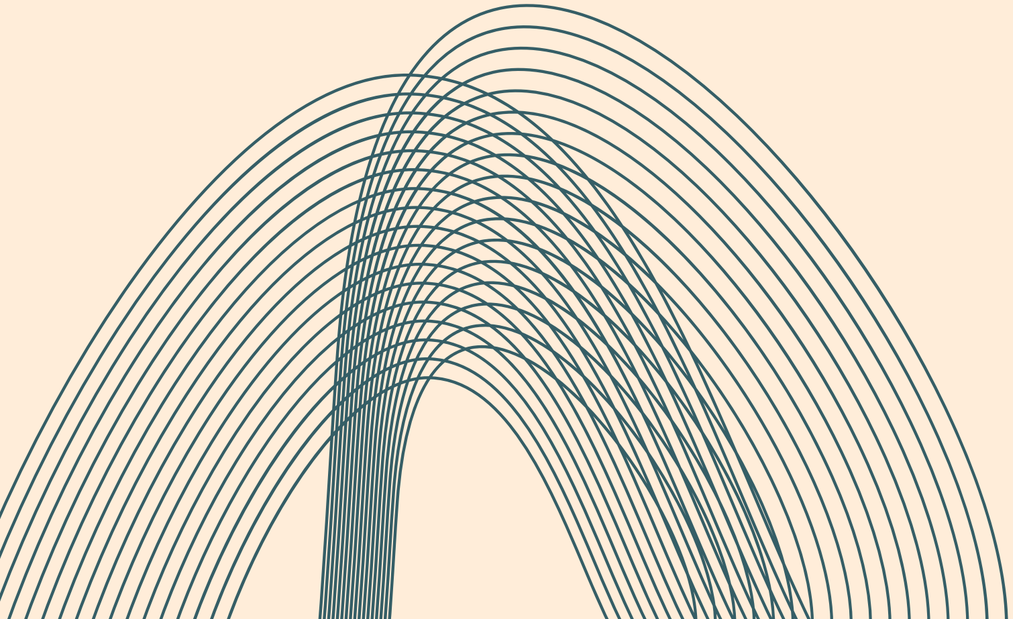
Data Shape

Rows

62896

Columns

20



Deskripsi Kolom

- company_name - nama perusahaan
- status_label - status perusahaan (alive = aktif, failed = bangkrut)
- year - tahun
- cost_of_goods_sold - jumlah total yang dibayar perusahaan sebagai biaya yang terkait langsung dengan penjualan produk
- depreciation_and_amortization - depreciation merujuk pada penurunan nilai aset yang berwujud seiring berjalannya waktu (seperti properti, mesin, gedung, dan pabrik), sedangkan amortization merujuk pada penurunan nilai aset yang tidak berwujud seiring berjalannya waktu
- ebitda (earnings before interest, taxes, depreciation, and amortization) - laba yang diperoleh perusahaan sebelum dikurangi bunga, pajak, depreciation, dan amortization inventory - nilai barang-barang dan bahan baku yang dimiliki perusahaan untuk diproduksi atau dijual

Deskripsi Kolom

- total_receivables - nilai yang harus diterima perusahaan dari barang/jasa yang diberikan tetapi belum dibayar oleh pelanggan
- market_value - harga suatu aset di pasar (dalam dataset ini, merujuk pada kapitalisasi pasar karena perusahaan diperdagangkan secara publik di pasar saham)
- net_sales - jumlah pendapatan dari penjualan setelah dikurangi pengembalian barang, potongan harga, dan diskon
- total_assets - semua aset atau barang berharga milik perusahaan
- total_long_term_debt - jumlah hutang yang harus dibayar perusahaan lebih dari satu tahun dari sekarang
- ebit (earnings before interest and taxes) - laba perusahaan sebelum dikurangi bunga dan pajak
- gross_profit (laba kotor) - laba yang diperoleh bisnis setelah dikurangi semua biaya yang terkait dengan produksi dan penjualan produk/jasa

Deskripsi Kolom

- retained_earnings - jumlah laba perusahaan yang tersisa setelah membayar semua biaya langsung, biaya tidak langsung, pajak penghasilan, dan dividen kepada pemegang saham
- total_liabilities - gabungan hutang dan kewajiban yang dimiliki perusahaan terhadap pihak luar
- total_operating_expenses - total biaya operasional
- current_ratio - rasio keuangan yang membandingkan total aset lancar perusahaan dengan utang lancar perusahaan
- net_profit_margin_category - kategori yang menunjukkan seberapa besar persentase keuntungan bersih perusahaan dibandingkan dengan pendapatannya

Overview

	ID	company_name	status_label	year	cost_of_goods_sold	depreciation_and_amortization	ebitda	inventory	total_receivables	market_value	net_sales	total_assets
0	CQZW3V9G	C_1	alive	1999	833.107	18.373	89.031	336.018	128.348	372.7519	1024.333	740.998
1	CIRJ6YT8	C_1	alive	2000	713.811	18.577	64.367	320.590	115.187	377.1180	874.255	701.854
2	CAHLFH0N	C_1	alive	2001	526.477	22.496	27.207	286.588	77.528	364.5928	638.721	710.199
3	CBU4UE1T	C_1	alive	2002	496.747	27.172	30.745	259.954	66.322	143.3295	606.337	686.621
4	C0DQ4A9M	C_1	alive	2003	523.302	26.680	47.491	247.245	104.661	308.9071	651.958	709.292

total_long_term_debt	ebit	gross_profit	retained_earnings	total_liabilities	total_operating_expenses	current_ratio	net_profit_margin_category
180.447	70.658	191.226	201.026	401.483	935.302	3.12	Low Profit Margin
179.987	45.790	160.444	204.065	361.642	809.888	3.87	Low Profit Margin
217.699	4.711	112.244	139.603	399.964	611.514	2.90	Low Profit Margin
164.658	3.573	109.590	124.106	391.633	575.592	1.95	Low Profit Margin
248.666	20.811	128.656	131.884	407.608	604.467	3.29	Low Profit Margin

Data Preprocessing

Data Duplication

Tidak terdapat data duplikat

Missing Value

Tidak terdapat data yang missing value

Outliers

	Column	Outlier	persentase
0	retained_earnings	14045	0.223305
1	ebit	10419	0.165654
2	total_long_term_debt	10158	0.161505
3	inventory	9915	0.157641
4	total_liabilities	9803	0.155860
5	ebitda	9606	0.152728
6	cost_of_goods_sold	9351	0.148674
7	total_assets	9335	0.148420
8	market_value	9109	0.144826
9	net_sales	9077	0.144318
10	depreciation_and_amortization	9068	0.144175
11	total_operating_expenses	9065	0.144127
12	gross_profit	9021	0.143427
13	total_receivables	9017	0.143364
14	current_ratio	5292	0.084139

Kami memutuskan untuk tidak menangani outlier karena:

1. **Relevansi Bisnis**: Outlier dalam data keuangan bisa mewakili situasi bisnis yang signifikan, seperti perusahaan dengan pertumbuhan luar biasa atau yang mengalami kesulitan keuangan ekstrem. Menghapus outlier berpotensi menghilangkan wawasan berharga tentang performa dan risiko perusahaan.
2. **Keanekaragaman Perusahaan**: Perusahaan dalam dataset ini mungkin sangat bervariasi dalam hal ukuran, industri, dan strategi keuangan. Outlier dapat mencerminkan perbedaan yang alami dan penting, seperti perusahaan besar dengan aset atau pendapatan yang sangat tinggi dibandingkan perusahaan kecil.
3. **Analisis Risiko dan Peluang**: Dalam konteks keuangan, memahami ekstrem sangatlah penting. Outlier dapat membantu mengidentifikasi perusahaan yang menghadapi risiko tinggi (misalnya, beban utang besar) atau yang menunjukkan peluang investasi potensial (seperti peningkatan pasar yang signifikan).

Oleh karena itu, kami akan menggunakan model klasifikasi dan regresi yang robust terhadap outlier, seperti Random Forest, untuk memastikan hasil analisis dapat diandalkan.

Feature Engineering

```python
# Define a function to create new features
def create_new_features(df):
    # Avoid division by zero by replacing zeros with a small number
    df['net_sales'].replace(0, 1e-6, inplace=True)
    df['total_assets'].replace(0, 1e-6, inplace=True)
    df['total_assets_minus_liabilities'] = df['total_assets'] - df['total_liabilities']
    df['total_assets_minus_liabilities'].replace(0, 1e-6, inplace=True)
    df['total_receivables'].replace(0, 1e-6, inplace=True)

    # Gross Profit Margin
    df['gross_profit_margin'] = df['gross_profit'] / df['net_sales']

    # EBITDA Margin
    df['ebitda_margin'] = df['ebitda'] / df['net_sales']

    # EBIT Margin
    df['ebit_margin'] = df['ebit'] / df['net_sales']

    # Asset Turnover
    df['asset_turnover'] = df['net_sales'] / df['total_assets']

    # Debt-to-Equity Ratio
    df['debt_to_equity'] = df['total_liabilities'] / df['total_assets_minus_liabilities']

    # Receivables Turnover
    df['receivables_turnover'] = df['net_sales'] / df['total_receivables']

    # Operating Expense Ratio
    df['operating_expense_ratio'] = df['total_operating_expenses'] / df['net_sales']

    # Depreciation and Amortization Ratio
    df['depreciation_amortization_ratio'] = df['depreciation_and_amortization'] / df['total_assets']

    # Drop intermediate columns if not needed
    df.drop(['total_assets_minus_liabilities'], axis=1, inplace=True)

    return df
```

1. Gross Profit Margin
Tujuan: Mengukur efisiensi perusahaan dalam menghasilkan laba kotor dari penjualan.
Manfaat: Menilai performa operasional dasar.

2. EBITDA Margin
Tujuan: Menunjukkan efisiensi operasional tanpa pengaruh pajak, bunga, atau depresiasi.
Manfaat: Membandingkan kinerja operasional antar perusahaan.

3. EBIT Margin
Tujuan: Mengukur profitabilitas operasional dari penjualan.
Manfaat: Analisis laba sebelum pengaruh pajak dan bunga.

4. Asset Turnover
Tujuan: Mengukur efisiensi penggunaan aset untuk menghasilkan penjualan.
Manfaat: Menilai efektivitas pemanfaatan aset.

5. Debt-to-Equity Ratio
Tujuan: Mengukur struktur pendanaan perusahaan (hutang vs modal).
Manfaat: Menilai risiko keuangan perusahaan.

6. Receivables Turnover
Tujuan: Mengukur kecepatan penagihan piutang dari penjualan.
Manfaat: Menilai likuiditas dan manajemen piutang.

7. Operating Expense Ratio
Tujuan: Mengukur proporsi biaya operasional terhadap penjualan.
Manfaat: Menilai efisiensi pengendalian biaya.

8. Depreciation and Amortization Ratio
Tujuan: Mengukur beban depresiasi terhadap total aset.
Manfaat: Menilai investasi dan umur aset perusahaan.

EDA

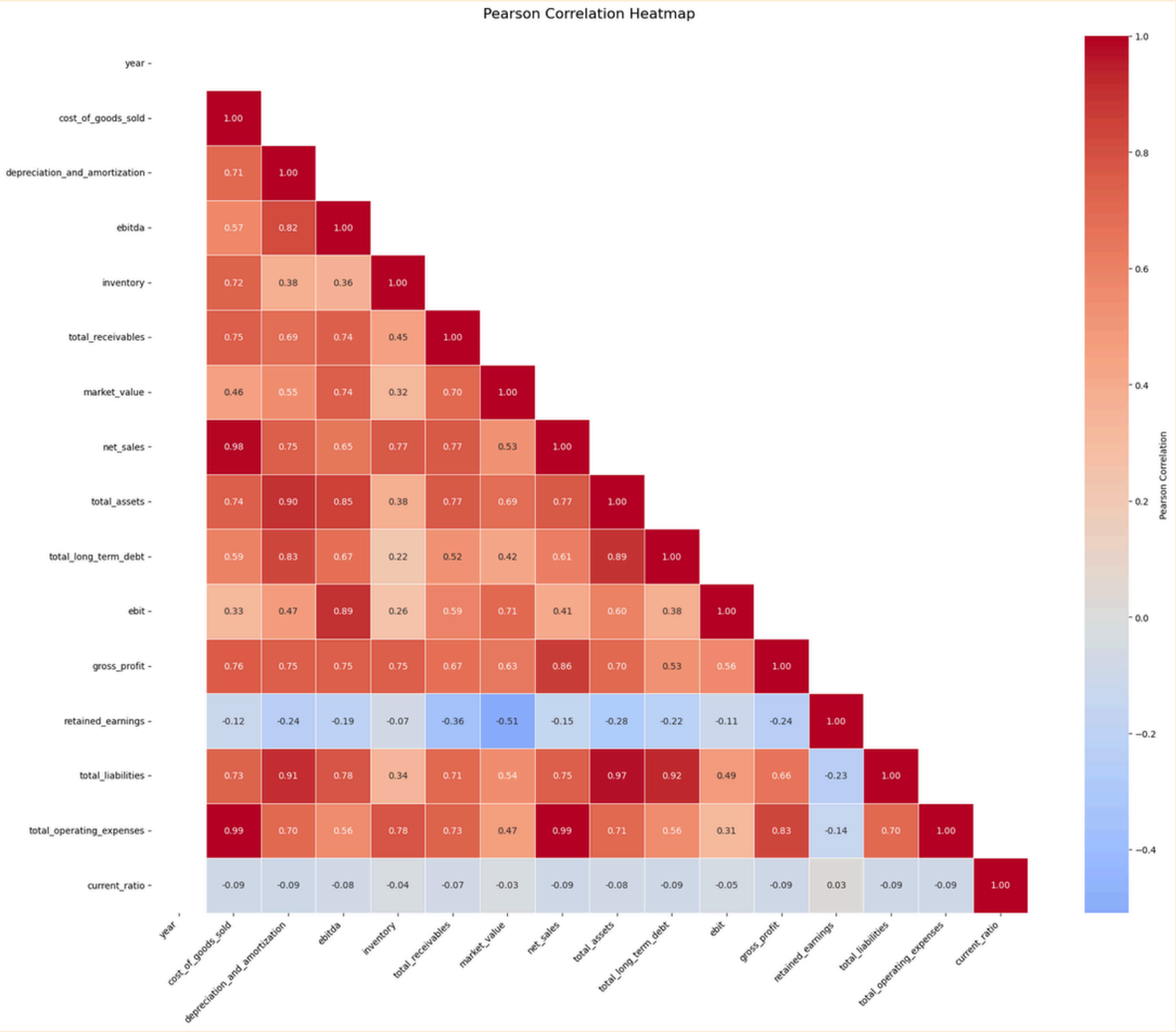
Karakteristik perusahaan yang bangkrut pada tahun dengan jumlah kebangkrutan tertinggi

Tahun dengan jumlah kebangkrutan tertinggi

2003

Banyak perusahaan bangkrut pada tahun tersebut

415



Beban Operasional Tinggi

total_operating_expenses memiliki korelasi yang sangat tinggi dengan **net_sales**. Ini menunjukkan bahwa biaya operasional meningkat seiring penjualan meningkat.

Pengelolaan Laba Ditahan Tidak Optimal

retained_earnings berkorelasi negatif dengan banyak variabel lain, menunjukkan bahwa perusahaan mungkin tidak berhasil menabung atau menginvestasikan kembali labanya secara efektif.

Profitabilitas yang Tidak Stabil

Jika terjadi penurunan penjualan atau peningkatan biaya barang yang dijual (COGS), perusahaan bisa mengalami penurunan profitabilitas yang signifikan, mempercepat kebangkrutan.

Apakah perusahaan dengan gross profit di bawah rata-rata dan long term debt di atas rata-rata lebih cenderung mengalami kebangkrutan?

Perusahaan bangkrut: 422

Perusahaan tidak bangkrut: 2429

JIka melihat perbandingan antara perusahaan yang bangkrut dan tidak, dapat dibilang kalau perusahaan dengan ketentuan tersebut tidak cenderung mengalami kebangkrutan

Bagaimana hubungan antara total operating expenses, total assets, dan gross profit terhadap market value dalam perusahaan?

	market_value	total_operating_expenses	total_assets	gross_profit
market_value	1.000000	0.655469	0.791588	0.852560
total_operating_expenses	0.655469	1.000000	0.746548	0.815820
total_assets	0.791588	0.746548	1.000000	0.888621
gross_profit	0.852560	0.815820	0.888621	1.000000

Berdasarkan matriks korelasi tersebut, kita dapat melihat hubungan antara market value dan variabel lainnya:

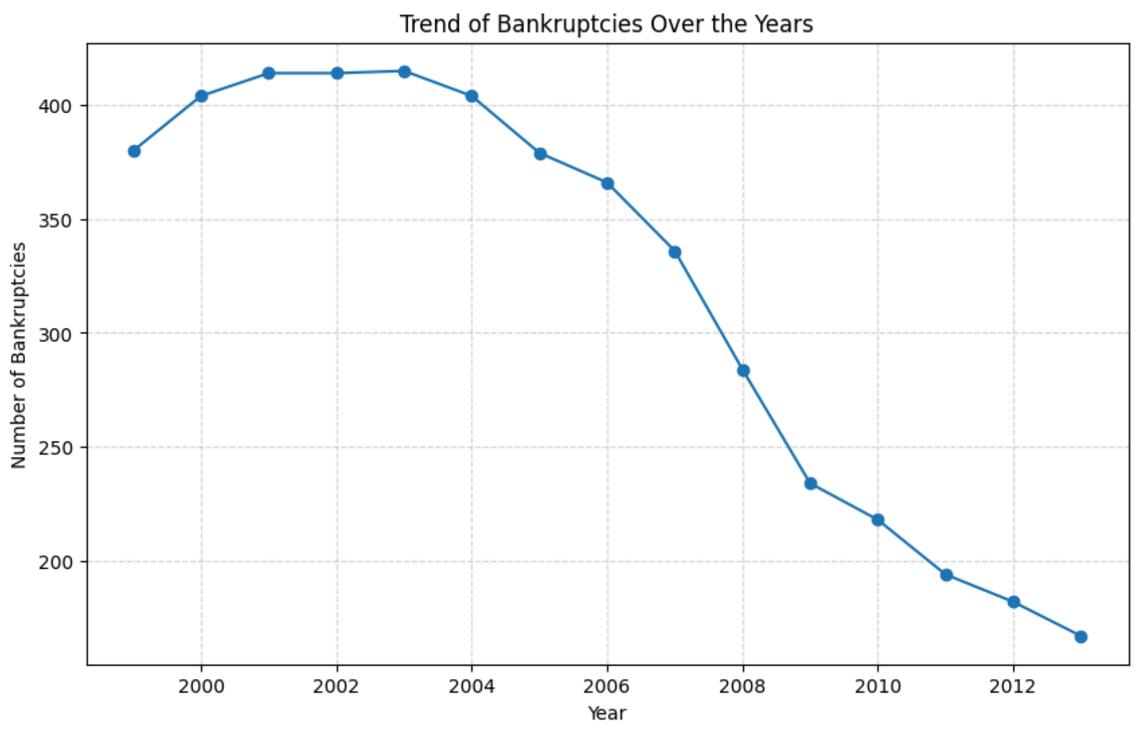
Market value memiliki korelasi yang cukup tinggi dengan gross profit (0.85), total assets (0.79), dan total operating expenses (0.66). Ini menunjukkan bahwa ketika gross profit, total assets, atau total operating expenses meningkat, ada kecenderungan market value juga meningkat. Kesimpulannya market value memiliki hubungan yang kuat dengan gross profit, serta total assets dan total operating expenses.

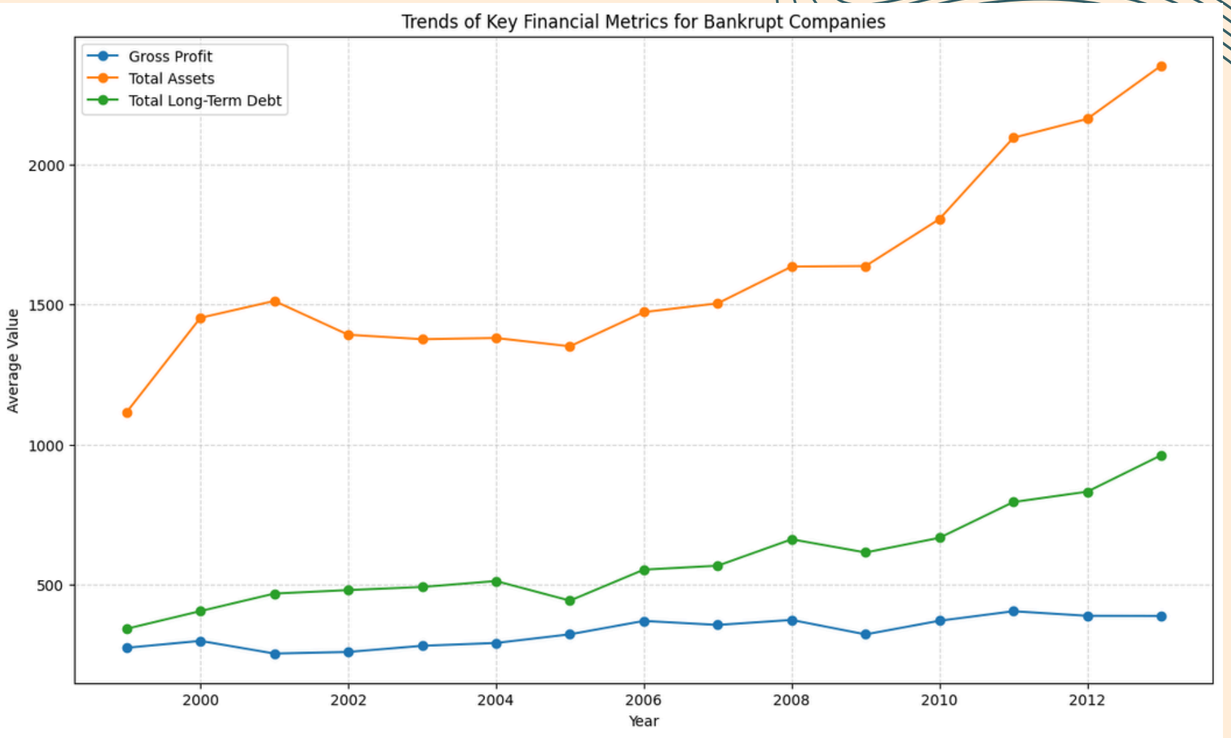
	net_profit_margin_category
net_profit_margin_category	1.000000
market_value	0.180679
ebit	0.170725
ebitda	0.158235
gross_profit	0.133256
total_assets	0.132676
retained_earnings	0.115553
total_liabilities	0.113091
total_long_term_debt	0.102686
total_receivables	0.101973
depreciation_and_amortization	0.100295
year	0.081317
net_sales	0.054917
inventory	0.039486
total_operating_expenses	0.032996
cost_of_goods_sold	0.018714
current_ratio	0.000448

Faktor-faktor apa yang mempengaruhi tingkat net profit margin perusahaan?

Berdasarkan tabel korelasi tersebut, dapat disimpulkan bahwa market_value merupakan variabel yang memiliki korelasi paling tinggi dibanding variabel lainnya. Namun, jika dilihat dari nilai korelasinya juga tidak dapat dikatakan sebagai variabel yang memiliki korelasi yang kuat

Apakah tren kebangkrutan tahun ke tahun dipengaruhi oleh variabel total_long_term_debt?



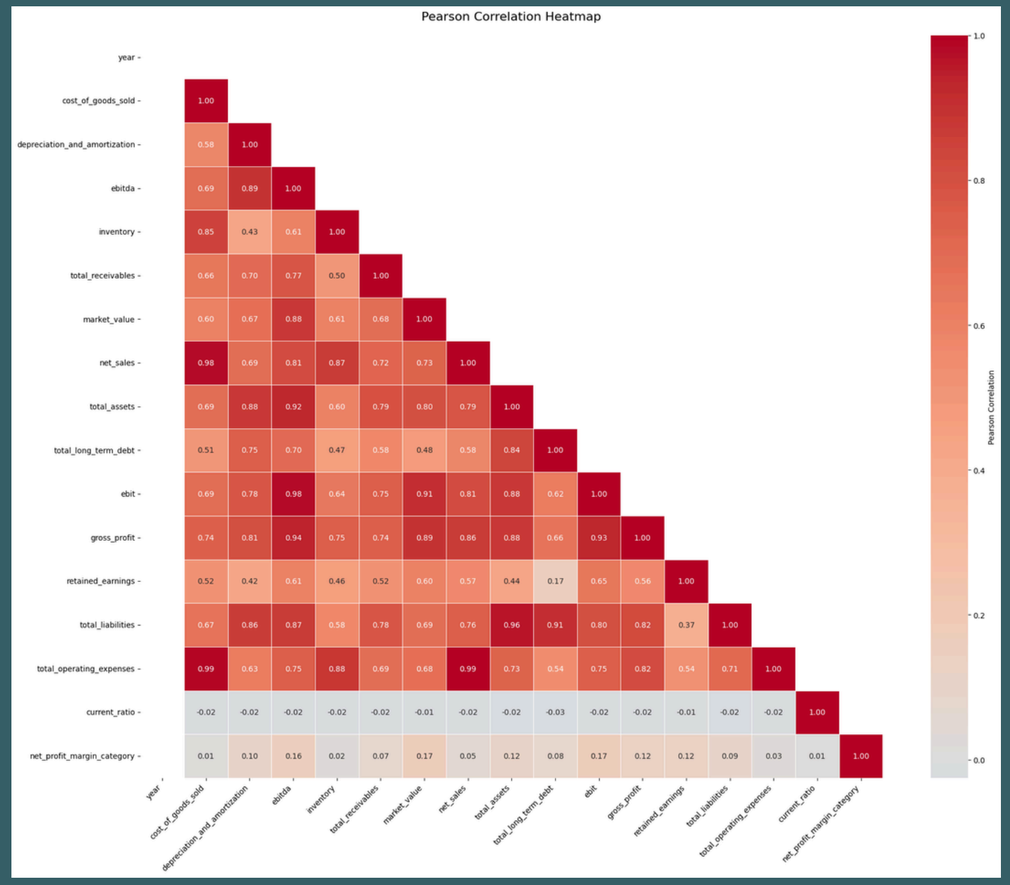


Kesimpulan

- Data menunjukkan bahwa stagnasi laba kotor mungkin memiliki dampak yang lebih signifikan terhadap risiko kebangkrutan dibandingkan dengan tingkat utang jangka panjang. Meskipun perusahaan membawa lebih banyak utang dari waktu ke waktu, mereka mungkin berhasil mengelola utang tersebut dengan cara yang tidak langsung menyebabkan kegagalan, kemungkinan karena kondisi ekonomi yang menguntungkan atau strategi keuangan yang mengurangi risiko utang.

- Penurunan jumlah kebangkrutan dari waktu ke waktu mungkin lebih terkait dengan faktor ekonomi eksternal atau perbaikan kondisi pasar secara keseluruhan, daripada faktor internal seperti utang

Bagaimana karakteristik perusahaan yang bertahan pada tahun dengan jumlah kebangkrutan tertinggi?

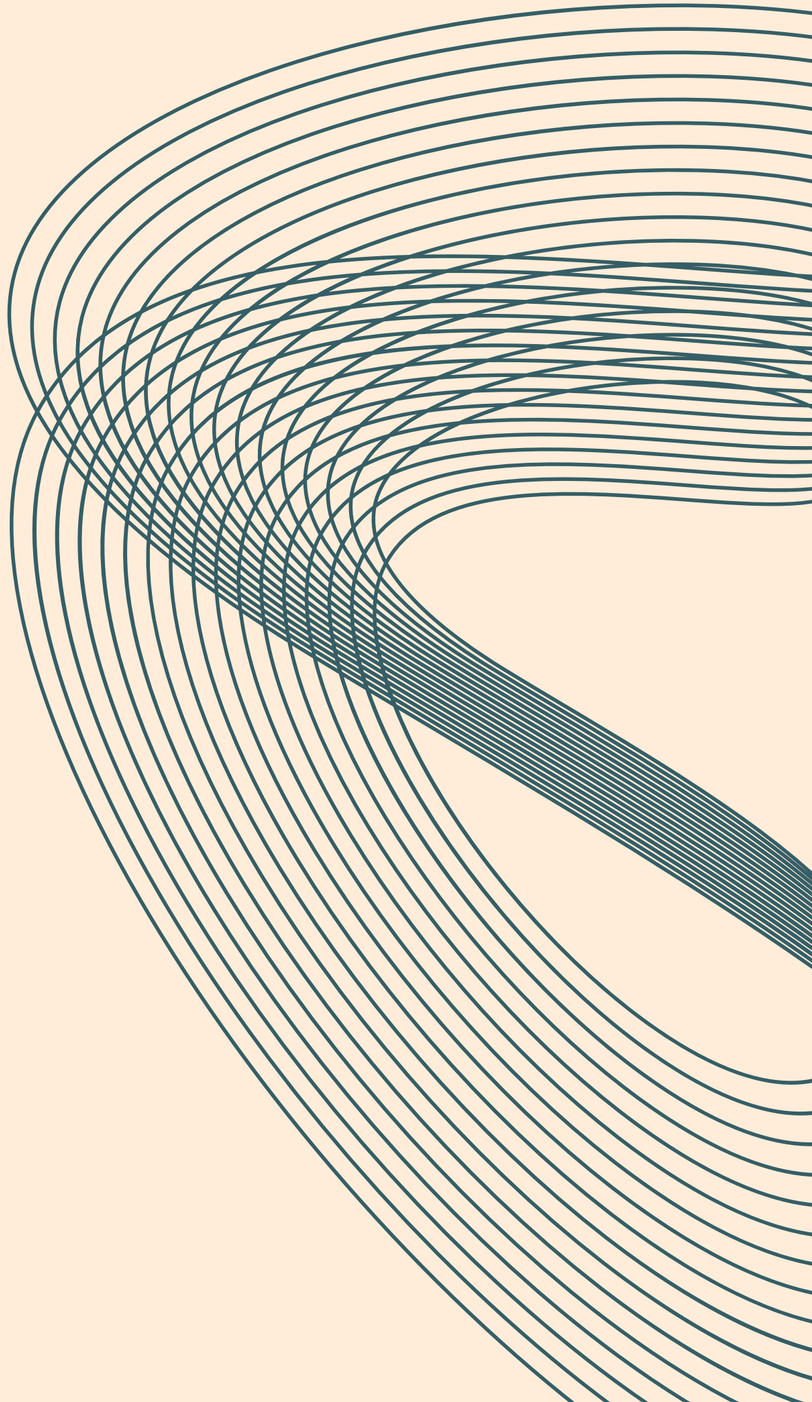


Analisis berdasarkan diagram korelasi tersebut:
Perusahaan yang cenderung bertahan dibanding perusahaan yang bangkrut adalah fitur "retained_earnings" yang cenderung berkorelasi positif dengan semua fitur lainnya.

Model

Classification

Random Forest



Klasifikasi - Company Financial

Submit Prediction

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

This leaderboard is calculated with approximately 30% of the test data. The final results will be based on the other 70%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Join
1	Urip Mulyo		0.86510	10	3h	
2	Mas Thirza Silahkan ke Depan		0.86509	39	1h	
3	**CRUD**		0.85868	35	2h	
4	mas iqza tolong ambilin sembak		0.85783	33	15m	
5	Mikir Dulu		0.85616	20	5h	
6	!1!! kasDUAR !!1!		0.85529	51	6h	
	clas_pred_1.csv		0.85179			

Your Best Entry!
Your submission scored 0.85045, which is not an improvement of your previous score. Keep trying!

Agar mencegah *overfit,* fitur company_name kita drop

```
df.drop('company_name', axis=1, inplace=True)
```

Untuk fitur target ‘net_profit_margin_category’ dan ‘status_label’, kita lakukan *ordinal encoding*

```
df['net_profit_margin_category'] = df['net_profit_margin_category'].map({'Low Profit Margin': 0, 'Healthy Profit Margin': 1, 'High Profit Margin': 2})
df['status_label'] = df['status_label'].map({'alive': 1, 'failed': 0})
```

Setelah itu kita cek juga, jumlah data yang memiliki atribut masing-masing kelas target.

```
print(len(df[df["net_profit_margin_category"] == 0].index))
print(len(df[df["net_profit_margin_category"] == 1].index))
print(len(df[df["net_profit_margin_category"] == 2].index))
```

```
42240
10210
10446
```

Didapat jumlah perusahaan terhadap net_profit_margin_category:

1. Low Profit Margin : 42240
2. Healthy Profit Margin : 10210
3. High Profit Margin : 10446

Berdasarkan informasi terhadap jumlah data terhadap masing kelas target, kita bisa spesifikasikan weight masing kelas target dari model menjadi:

1. Low Profit : 1
2. Healthy Profit : 1.25
3. High Profit : 1.25

```
rfc = RandomForestClassifier(
    n_estimators=135,
    max_depth=20,
class_weight= {0: 1, 1: 1.25, 2: 1.25},
)
```

```python
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score

skf = StratifiedKFold(n_splits=5)

display(cross_val_score(rfc, X, y, cv=skf, scoring='accuracy', verbose=3).mean())
display(cross_val_score(rfc, X, y, cv=skf, scoring='f1_macro', verbose=3).mean())
```

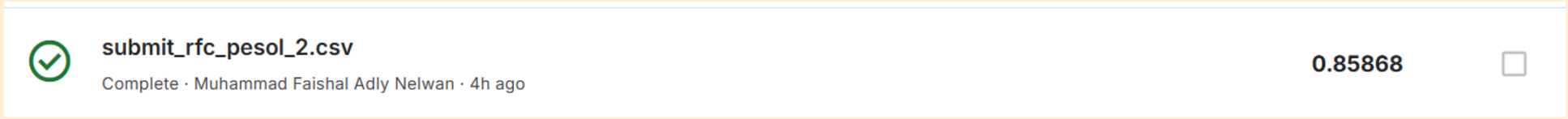
```
[CV] END ................................. score: (test=0.838) total time=  50.7s
[CV] END ................................. score: (test=0.856) total time=  47.8s
[CV] END ................................. score: (test=0.861) total time=  46.0s
[CV] END ................................. score: (test=0.853) total time=  47.0s
[CV] END ................................. score: (test=0.855) total time=  46.1s

0.8528207584978673

[CV] END ................................. score: (test=0.772) total time=  46.5s
[CV] END ................................. score: (test=0.787) total time=  45.9s
[CV] END ................................. score: (test=0.781) total time=  46.6s
[CV] END ................................. score: (test=0.764) total time=  46.3s


0.776
```

Setelah melakukan *cross-validating* terhadap keseluruhan training data, didapat skor model yaitu:

1. Accuracy : 0.8528
2. F1 Macro : 0.776

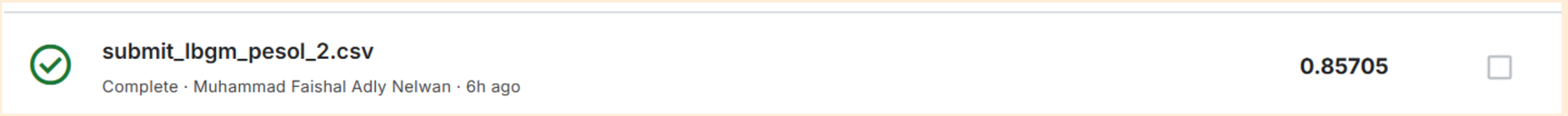
```
test = pd.read_csv('company_test_classif.csv')

test_x_ori = test.drop(["ID"], axis=1)
id = test["ID"]

test_x = test_x_ori.copy()
test_x['status_label'] = test_x_ori['status_label'].map({'alive': 1, 'failed': 0})
test_x = create_new_features(test_x)

test_x.drop('company_name', axis=1, inplace=True)

rfc.fit(X, y)

y = pd.Series(rfc.predict(test_x)).map({0: 'Low Profit Margin', 1: 'Healthy Profit Margin', 2: 'High Profit Margin'})

prediction = pd.concat([id, y], axis=1).rename(columns={0: 'net_profit_margin_category'})

display(prediction)

prediction.to_csv('submit_rfc_pesol_2.csv', index=False)
```

Setelah kita fitting model dan predict data test, didapat skor akhir kita pada 30% keseluruhan data yaitu:



Dari segala percobaan kita memakai model lainnya, didapat hasil skor performance terhadap F1 Score data kaggle antara berikut: (Hanya mengambil submisi dengan skor terdekat dengan hasil tertinggi)

1. LGBMClassifier dengan kolom company_name

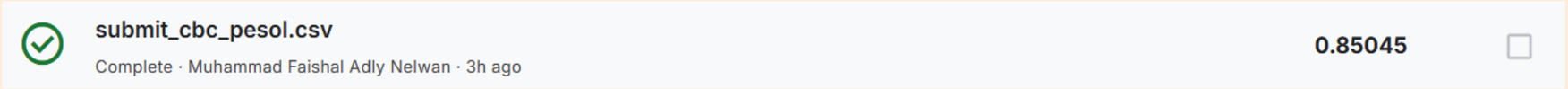


2. LGBMClassifier tanpa kolom company_name

3. ExtraTreeClassifier dengan kolom company_name

4. ExtraTreeClassifier tanpa kolom company_name

5. Catboost dengan parameter auto_balanced_weight



Regression

ExtraTreeRegression dengan approach one-hot encoding ditambah moving average

Regresi - Company Financial

Submit Prediction ...

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

This leaderboard is calculated with approximately 30% of the test data. The final results will be based on the other 70%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Join
1	Mas Thirza Silahkan ke Depan		0.89333	20	3h	
2	mas iqza tolong ambilin sembak		0.86555	28	5h	
3	**CRUD**		0.82322	52	10m	
4	Mikir Dulu		0.71575	34	1h	
5	Urip Mulyo		0.71456	33	36m	
6	!1!! kasDUAR !!1!		0.64719	63	6h	

Your Best Entry!
Your submission scored 0.82043, which is not an improvement of your previous score. Keep trying!

Penambahan *derived features* dari fitur-fitur yang sudah ada dan class weights. Kemudian digunakan ExtraTreeRegression sebagai model terbaik.

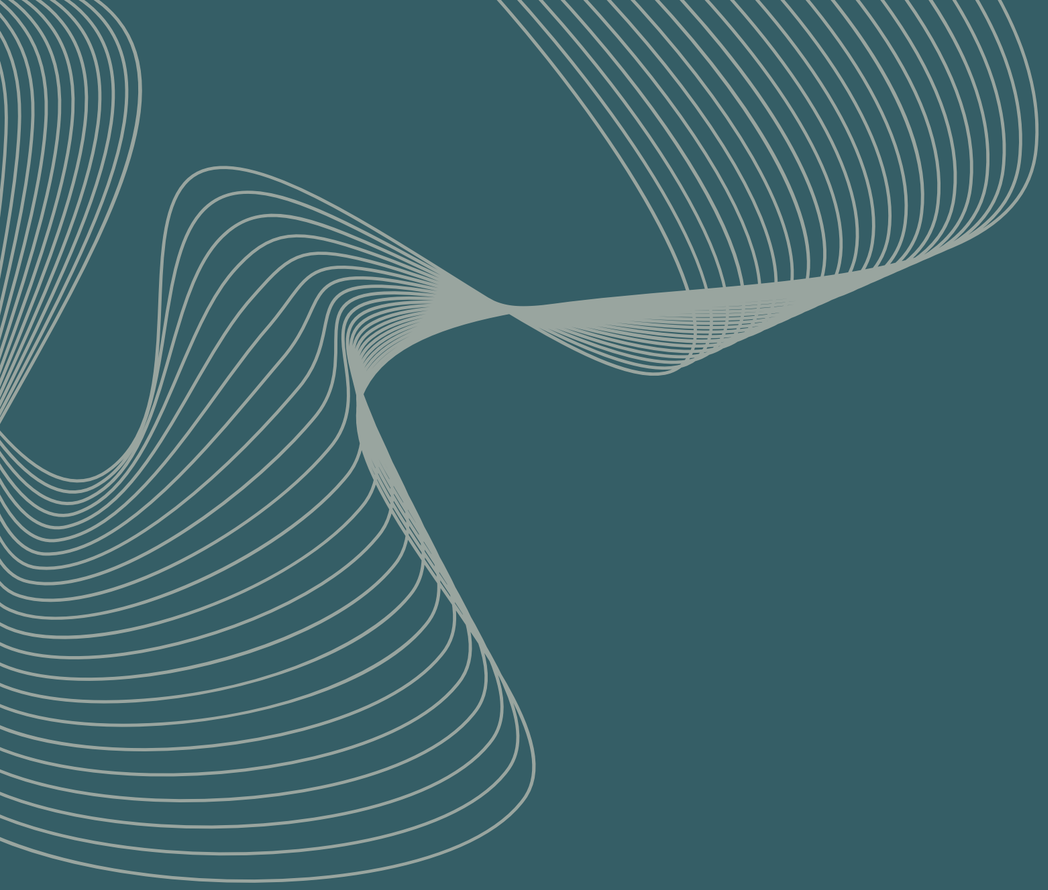
```python
anchor1 = "gross_profit"
anchor2 = "market_value"
anchor3 = "gross_profit"
anchor4 = "total_long_term_debt"

rolling_n = 3
df[f'{anchor1}_moving_avg'] = df.groupby('company_name')[anchor1].transform(lambda x: x.rolling(rolling_n, 1).mean())
df[f'{anchor2}_moving_avg'] = df.groupby('company_name')[anchor2].transform(lambda x: x.rolling(rolling_n, 1).mean())
df[f'{anchor3}_moving_avg'] = df.groupby('company_name')[anchor3].transform(lambda x: x.rolling(rolling_n, 1).mean())
df[f'{anchor4}_moving_avg'] = df.groupby('company_name')[anchor4].transform(lambda x: x.rolling(rolling_n, 1).mean())

df['net_sales'] = df['net_sales'].replace(0, 1e-6)
df['total_assets'] = df['total_assets'].replace(0, 1e-6)
df['total_assets_minus_liabilities'] = df['total_assets_minus_liabilities'] = df['total_assets'] - df['total_liabilities']
df['total_assets_minus_liabilities'] = df['total_assets_minus_liabilities'].replace(0, 1e-6)
df['total_receivables'] = df['total_receivables'].replace(0, 1e-6)

df['gross_profit_margin'] = df['gross_profit'] / df['net_sales']
df['ebitda_margin'] = df['ebitda'] / df['net_sales']
df['ebit_margin'] = df['ebit'] / df['net_sales']
df['asset_turnover'] = df['net_sales'] / df['total_assets']
df['debt_to_equity'] = df['total_liabilities'] / df['total_assets_minus_liabilities']
df['receivables_turnover'] = df['net_sales'] / df['total_receivables']
df['operating_expense_ratio'] = df['total_operating_expenses'] / df['net_sales']
df['depreciation_amortization_ratio'] = df['depreciation_and_amortization'] / df['total_assets']
df.drop(['total_assets_minus_liabilities'], axis=1)

df = pd.get_dummies(df, columns=['company_name'])
```

	mean r2	var r2	fit time	fold 1	fold 2	fold 3	fold 4	fold 5
extra_trees	0.953839	0.000127	13145.915026	0.975628	0.947259	0.948892	0.944198	0.953217
xgboost	0.946611	0.00019	423.653524	0.954943	0.925183	0.963965	0.951876	0.937086
catboost	0.945056	0.000137	433.806422	0.959861	0.928861	0.939724	0.957115	0.93972
random_forest	0.913426	0.000357	7334.193715	0.948048	0.892854	0.901109	0.914661	0.910457
ridge	0.861055	0.001526	731.393552	0.876502	0.832603	0.901614	0.799544	0.895011
lightgbm	0.757707	0.003888	103.621433	0.695635	0.68591	0.749218	0.818175	0.839596
hist_gradient_boosting	0.699294	0.00362	889.618777	0.692487	0.661511	0.721729	0.799474	0.621268

```python
# 7. Create and Run the Optuna Study
print("\n=== Starting Optuna Hyperparameter Optimization ===")
study = optuna.create_study(direction='maximize', sampler=TPESampler(seed=RANDOM_SEED))
study.optimize(objective, n_trials=10, timeout=None)  # You can set a timeout in seconds

# 8. Display the Best Hyperparameters
print("\n=== Optuna Hyperparameter Optimization Completed ===")
print(f"Best Trial: {study.best_trial.number}")
print("Best Parameters:")
for key, value in study.best_trial.params.items():
    print(f"  {key}: {value}")
print(f"Best R² Score: {study.best_trial.value:.4f}")

# 9. Train the Best Model on the Entire Training Set
best_params = study.best_trial.params
best_params.update({
    'objective': 'reg:squarederror',
    'random_state': RANDOM_SEED,
    'verbosity': 0,
    'gamma': 0,
    'min_child_weight': 1,
    'reg_alpha': 0,
    'reg_lambda': 1.0
})

# Initialize the best model with early stopping
best_model = xgb.XGBRegressor(
    **best_params,
    eval_set=[(X_test, y_test)],  # Ideally, use a separate validation set
    tree_method='auto'  # Use 'gpu_hist' if GPU is available
)

# Fit the best model
print("\nTraining the Best Model on the Entire Training Set...")
best_model.fit(X_train, y_train)
```

```
=== Applying Custom Cross-Validation Function on Best Model ===
Fold 1: R² Score = 0.9584
Fold 2: R² Score = 0.9053
Fold 3: R² Score = 0.9400
Fold 4: R² Score = 0.9290
Fold 5: R² Score = 0.9206

=== Fold-wise R² Scores ===
Fold 1: R² Score = 0.9584
Fold 2: R² Score = 0.9053
Fold 3: R² Score = 0.9400
Fold 4: R² Score = 0.9290
Fold 5: R² Score = 0.9206


0.9307
```

Best parameter untuk model kedua terbaik ternyata masih belum bisa mengalahkan model terbaik yang kami temukan.

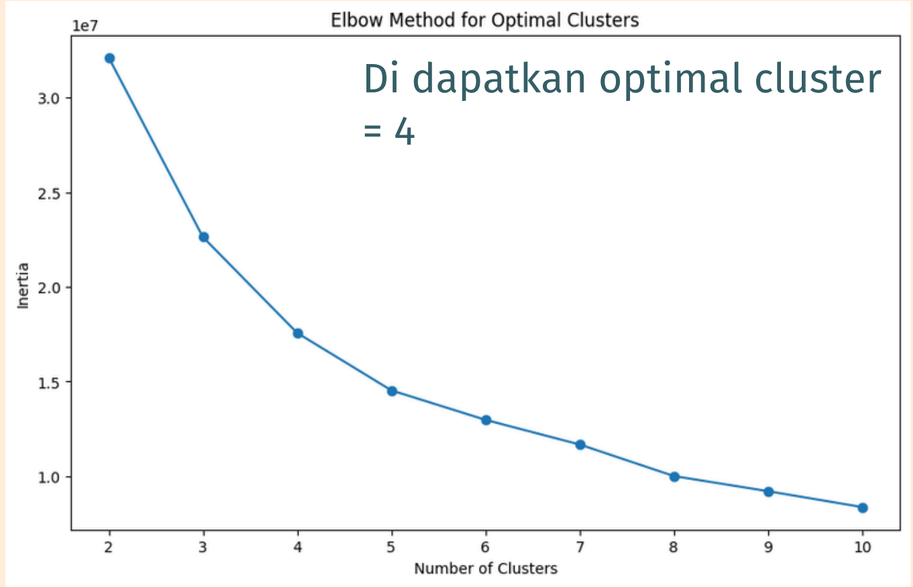


Clustering

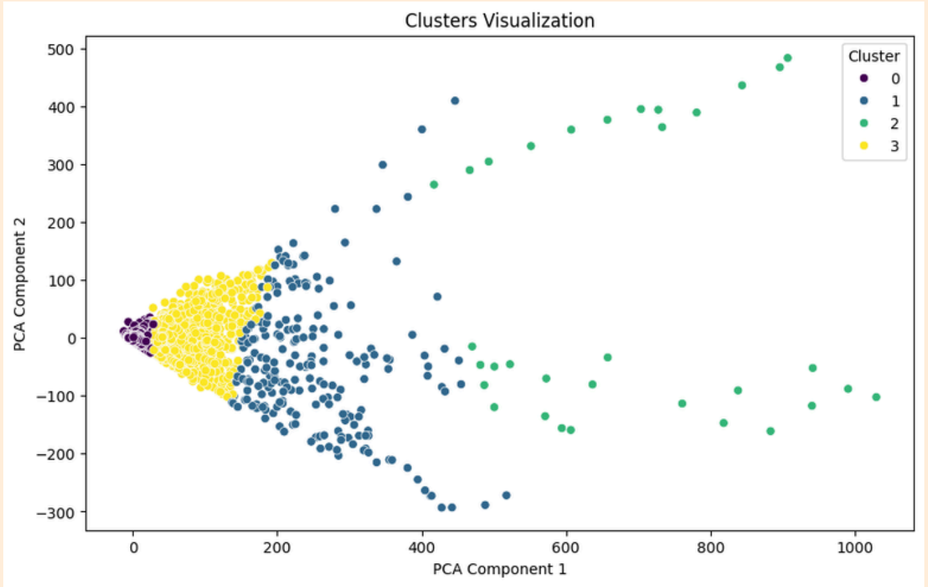
K-means

Kami mencoba clustering dengan menggunakan K-means dengan memilih 5 feature numeric saja. Disini kami memilih feature cost_of_goods_sold, ebitda, inventory, 'total_receivables, total_assets

Elbow Method



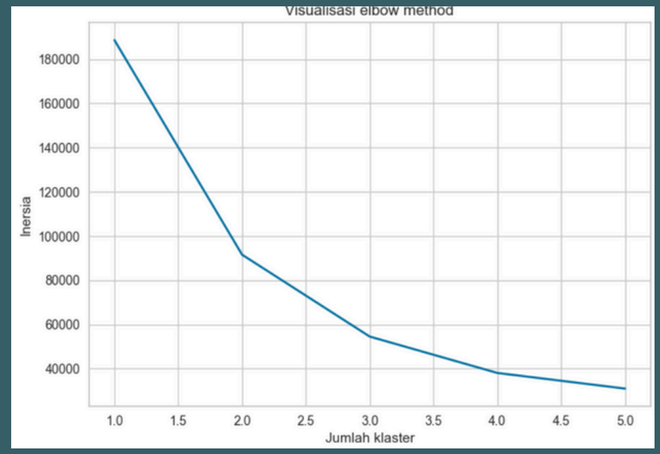
PCA visualisasi



K-Means Silhouette Score: 0.9012

K-means

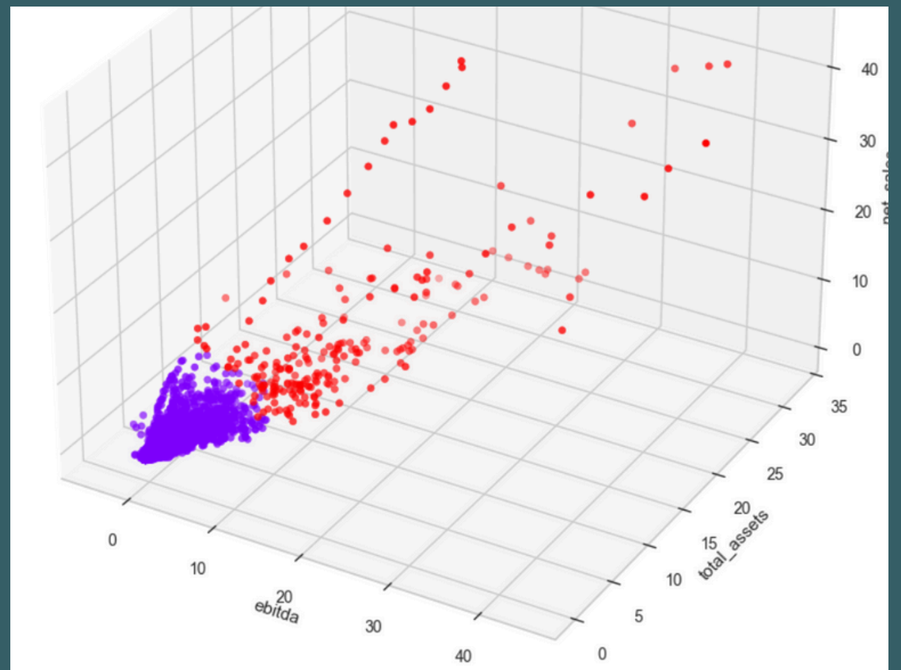
Kami mencoba membuat cluster dengan fitur EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization), Total Assets, dan Net Sales dengan alasan:



1. EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization):
   - Alasan: Menunjukkan kinerja operasional perusahaan tanpa pengaruh biaya finansial dan akuntansi non-tunai, memfasilitasi perbandingan yang adil antar perusahaan dalam klaster yang sama.

2. Total Assets:
   - Alasan: Mencerminkan ukuran dan kapasitas perusahaan, membantu pengelompokan berdasarkan skala operasi.

3. Net Sales:
   - Alasan: Menunjukkan pendapatan bersih dari penjualan, memberikan indikasi volume bisnis dan permintaan pasar.

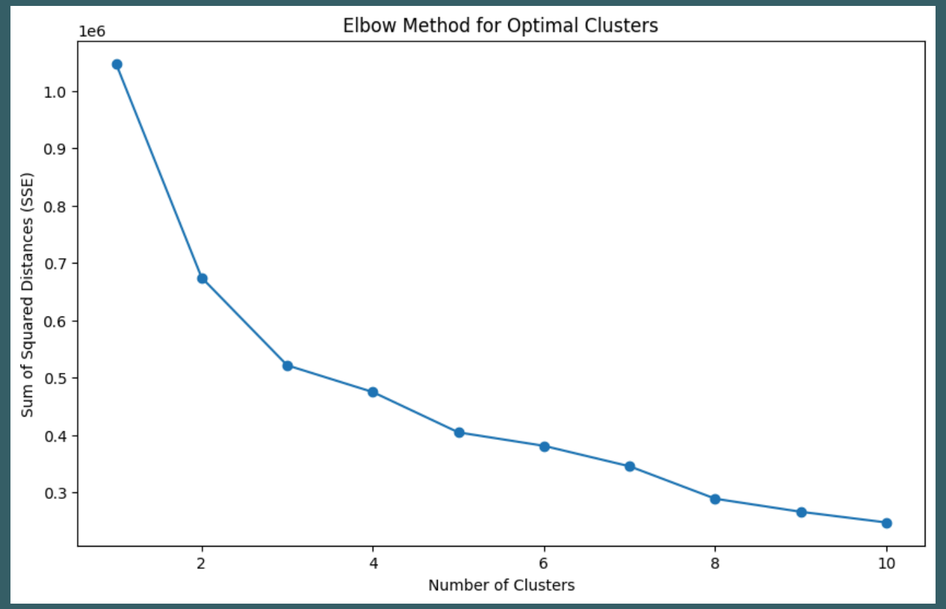


2 cluster
silhouette_coefficient adalah: 0.97203355539606

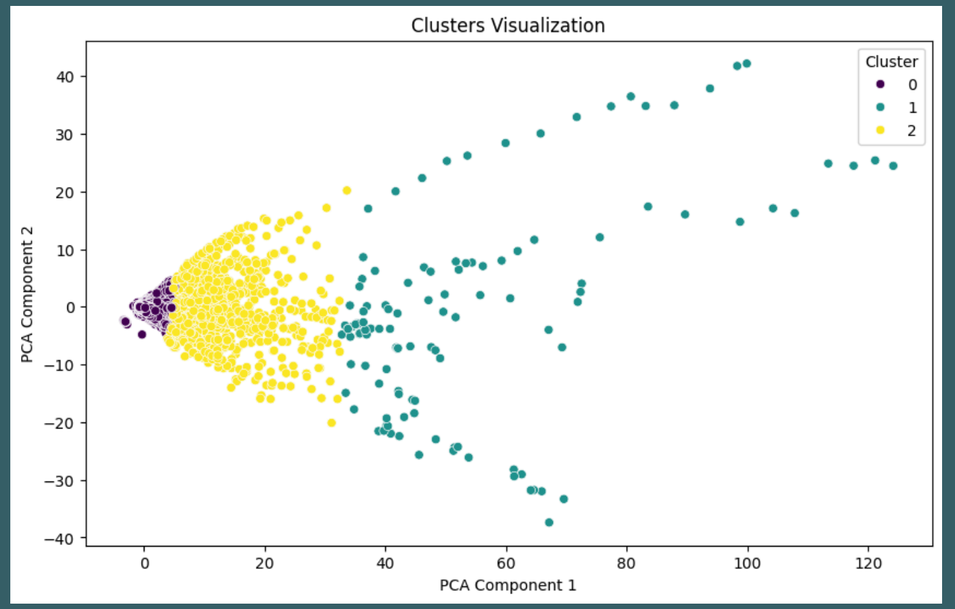
K-means

Kami mencoba membuat cluster dengan memanfaatkan semua fitur untuk mencoba menangkap pola yang lebih kompleks

3 Cluster

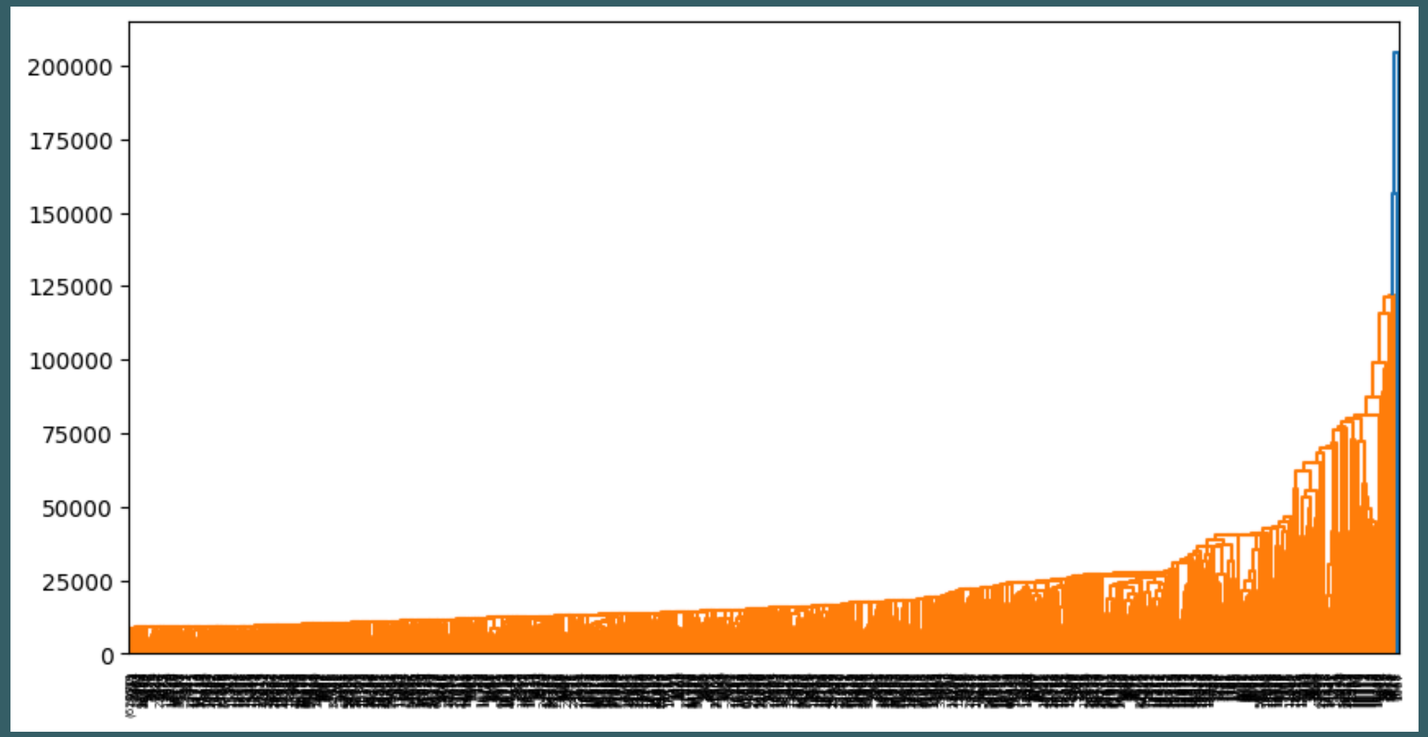


Silhouette Score: 0.8198



Agglomerative Clustering

Metode ini tidak kami lakukan karena metode ini memerlukan O(n^2) space dan O(n^3) kompleksitas waktu. Sehingga tidak cocok digunakan untuk dataset yang besar



THANK YOU!