

BANKS

```
In [1]: 1 %load_ext autoreload
        2 %autoreload 2
        3 %matplotlib inline
```

```
In [2]: 1 from collections import OrderedDict
        2 import datetime
        3 import glob
        4 #import ipywidgets as widgets
        5 from IPython.display import display, Markdown, HTML
        6 import matplotlib as mpl
        7 import matplotlib.pyplot as plt
        8 import numpy as np
        9 import os
       10 import pandas as pd
       11 from pandas.plotting import register_matplotlib_converters
       12 from scipy.stats import pearsonr
       13 from statsmodels.tsa.stattools import coint, adfuller
       14 import sys
       15 from typing import Dict, List
       16
       17 p = os.path.abspath('../tradelib')
       18 if p not in sys.path:
       19     sys.path.insert(0, p)
       20
       21 from tradelib.dataset import config, StockSummary, IndexSummary
       22 import tradelib.pipeline as P
       23 import tradelib.plotting as pt
       24
       25 register_matplotlib_converters()
       26
       27 MILLION = 1000000
       28 BILLION = 1000 * MILLION
       29
```

```
In [3]: 1 TICKERS = ['BMRI']
```

In [4]:

```
1 def read_monthly(ticker: str) -> pd.DataFrame:
2     monthly = pd.read_csv(f'{ticker}/{ticker}.csv', parse_dates=['
3     monthly.insert(0, 'ticker', ticker)
4     #monthly['date'] = monthly['date'] + pd.Timedelta(days=1)
5     monthly = monthly.set_index(['ticker', 'date'], drop=True)
6
7     # Convert income statement columns to monthly
8     pos = monthly.columns.get_loc('pendapatan bunga')
9     income_columns = monthly.columns[pos:]
10    for i in range(len(monthly)-1, 0, -1):
11        if monthly.index[i][1].month > 1:
12            monthly.loc[monthly.index[i], income_columns] -= month
13
14    if monthly.index[0][1].month > 1:
15        monthly = monthly.iloc[1:]
16
17    # Convert to milyar
18    pos = monthly.columns.get_loc('pembulatan')
19    pembulatan = monthly.iloc[0, pos]
20    value_cols = monthly.columns[pos:]
21    monthly[value_cols] *= (pembulatan / 1e9)
22    monthly = monthly.drop(columns=['pembulatan'])
23
24    # calculate tahunan (rolling 12 month)
25    monthly['kredit avg'] = monthly['kredit'].rolling(12).mean()
26    for col in income_columns:
27        monthly[col + ' ttm'] = monthly[col].rolling(12).sum()
28    return monthly
29
30
31 def read_monthlies():
32     dfs = []
33     for ticker in TICKERS:
34         df = read_monthly(ticker)
35         dfs.append(df)
36     return pd.concat(dfs)
```

```
In [5]: 1 master = read_monthlies()  
        2 master.columns
```

```
Out[5]: Index(['kredit', 'cadangan kerugian kredit', 'kol1', 'kol2', 'kol3',  
              'kol4',  
              'kol5', 'giro', 'tabungan', 'deposito', 'e-money', 'debt', 'li  
abilitas',  
              'ekuitas', 'pendapatan bunga', 'beban bunga', 'pendapatan bers  
ih',  
              'pendapatan premi bersih', 'beban nilai aset keuangan',  
              'beban nilai liabilitas keuangan',  
              'beban nilai penjualan aset keuangan',  
              'beban nilai transaksi spot derivatif', 'beban nilai penyertaa  
n equity',  
              'beban nilai valas', 'pendapatan dividen', 'pendapatan komis  
i',  
              'pendapatan lainnya', 'beban impairment', 'beban risiko operas  
ional',  
              'beban tenaga kerja', 'beban promosi', 'beban lainnya',  
              'total pendapatan beban operasional', 'laba operasional',  
              'laba sebelum pajak', 'laba bersih', 'laba komprehensif', 'kre  
dit avg',  
              'pendapatan bunga ttm', 'beban bunga ttm', 'pendapatan bersih  
ttm',  
              'pendapatan premi bersih ttm', 'beban nilai aset keuangan tt  
m',  
              'beban nilai liabilitas keuangan ttm',  
              'beban nilai penjualan aset keuangan ttm',  
              'beban nilai transaksi spot derivatif ttm',  
              'beban nilai penyertaan equity ttm', 'beban nilai valas ttm',  
              'pendapatan dividen ttm', 'pendapatan komisi ttm',  
              'pendapatan lainnya ttm', 'beban impairment ttm',  
              'beban risiko operasional ttm', 'beban tenaga kerja ttm',  
              'beban promosi ttm', 'beban lainnya ttm',  
              'total pendapatan beban operasional ttm', 'laba operasional tt  
m',  
              'laba sebelum pajak ttm', 'laba bersih ttm', 'laba komprehensi  
f ttm'],  
          dtype='object')
```

Plots

In [20]:

```
1 def plot_everything(tickers: List[str]=TICKERS, normalize=True):
2     start_date = pd.Timestamp('2022-12-31').date()
3     columns = []
4
5     for ticker in tickers:
6         display(Markdown(f'### {ticker}'))
7         df = master.loc[ticker].copy()
8         df = df.loc[start_date,: ]
9
10        # kredit disalurkan
11        kredit = df['kredit'].rename(f'{ticker} kredit disalurkan')
12        columns.append((kredit / kredit.iloc[0]) if normalize else kredit)
13
14        # total CKPN
15        ckpn = df['cadangan kerugian kredit'].rename(f'{ticker} total CKPN')
16        columns.append((ckpn / ckpn.iloc[0]) if normalize else ckpn)
17
18        # rasio CKPN
19        rasio_ckpn = (df['cadangan kerugian kredit'] / df['kredit']).rename(f'{ticker} rasio CKPN')
20        columns.append(rasio_ckpn)
21
22        # kol 2-5
23        kol25 = df[['kol2', 'kol3', 'kol4', 'kol5']].sum(axis=1).rename(f'{ticker} kol 2-5')
24        kol25 = kol25.replace(0, np.NaN).fillna(method='pad')
25        columns.append((kol25 / kol25.iloc[0]) if normalize else kol25)
26
27        # NPL gross
28        npl_gross = (kol25 / kredit).rename(f'{ticker} NPL Gross')
29        columns.append(npl_gross)
30
31        # kol 3-5
32        kol35 = df[['kol3', 'kol4', 'kol5']].sum(axis=1).rename(f'{ticker} kol 3-5')
33        kol35 = kol35.replace(0, np.NaN).fillna(method='pad')
34        columns.append((kol35 / kol35.iloc[0]) if normalize else kol35)
35
36        # NPL net
37        npl_net = (kol35 / kredit).rename(f'{ticker} NPL Net')
38        columns.append(npl_net)
39
40        # Rasio CKPN thd kol 2-5
41        ckpn_to_kol25 = (ckpn / kol25).rename(f'{ticker} rasio ckpn thd kol 2-5')
42        columns.append(ckpn_to_kol25)
43
44        # Rasio CKPN thd kol 3-5
45        ckpn_to_kol35 = (ckpn / kol35).rename(f'{ticker} rasio ckpn thd kol 3-5')
46        columns.append(ckpn_to_kol35)
47
48        # dana simpanan
49        simpanan = df[['giro', 'tabungan', 'deposito', 'e-money']].sum(axis=1).rename(f'{ticker} dana simpanan')
50        columns.append((simpanan / simpanan.iloc[0]) if normalize else simpanan)
51
52        # CASA
53        casa = df[['giro', 'tabungan', 'e-money']].sum(axis=1).rename(f'{ticker} CASA')
54        columns.append((casa / casa.iloc[0]) if normalize else casa)
55
56        # CASA ratio
57        casa_ratio = (casa / simpanan).rename(f'{ticker} Rasio CASA')
58        columns.append(casa_ratio)
```

```

59
60 # LDR
61 ldr = (kredit / simpanan).rename(f'{ticker} LDR')
62 columns.append(ldr)
63
64 # Pendapatan bunga
65 rev_b = df['pendapatan bunga'].rename(f'{ticker} pendapat
66 columns.append((rev_b/rev_b.iloc[0]) if normalize else re
67
68 rev_ttm = df['pendapatan bunga ttm'].rename(f'{ticker} pe
69 columns.append((rev_ttm/rev_ttm.iloc[0]) if normalize els
70
71 beban_bunga_b = df['beban bunga'].rename(f'{ticker} beban
72 columns.append((beban_bunga_b/beban_bunga_b.iloc[0]) if n
73
74 beban_bunga_ttm = df['beban bunga ttm'].rename(f'{ticker}
75 columns.append((beban_bunga_ttm/beban_bunga_ttm.iloc[0])
76
77 # GPM
78 gpm_b = (df['pendapatan bersih']/rev_b).rename(f'{ticker}
79 columns.append(gpm_b)
80
81 gpm_ttm = (df['pendapatan bersih ttm']/rev_ttm).rename(f'
82 columns.append(gpm_ttm)
83
84 # NIM ttm
85 nim_ttm = (df['pendapatan bersih ttm']/df['kredit avg']).
86 columns.append(nim_ttm)
87
88 # NIM bulanan annualized
89 nim_b_ann = (df['pendapatan bersih']*12/df['kredit']).ren
90 columns.append(nim_b_ann)
91
92 # beban impairment
93 beban_impairment_b = df['beban impairment'].rename(f'{tic
94 columns.append((beban_impairment_b/beban_impairment_b.ilo
95
96 beban_impairment_ttm = df['beban impairment ttm'].rename(
97 columns.append((beban_impairment_ttm/beban_impairment_ttm
98
99 rasio_ckpn_thd_pend_b = (beban_impairment_b / rev_b).rena
100 columns.append(rasio_ckpn_thd_pend_b)
101
102 rasio_ckpn_thd_pend_ttm = (beban_impairment_ttm / rev_ttm
103 columns.append(rasio_ckpn_thd_pend_ttm)
104
105
106 # Laba operasional
107 laba_opr_b = df['laba operasional'].rename(f'{ticker} lab
108 columns.append((laba_opr_b/laba_opr_b.iloc[0]) if normali
109
110 laba_opr_ttm = df['laba operasional ttm'].rename(f'{ticke
111 columns.append((laba_opr_ttm/laba_opr_ttm.iloc[0]) if nor
112
113 # OPM
114 opm_b = (laba_opr_b/rev_b).rename(f'{ticker} OPM bulanan'
115 columns.append(opm_b)
116

```

```

117     opm_ttm = (laba_opr_ttm/rev_ttm).rename(f'{ticker} OPM tt
118     columns.append(opm_ttm)
119
120     # PPOP
121     ppop_b = (laba_opr_b + beban_impairment_b).rename(f'{tick
122     columns.append((ppop_b/ppop_b.iloc[0]) if normalize else
123
124     ppop_ttm = (laba_opr_ttm + beban_impairment_ttm).rename(f
125     columns.append((ppop_ttm/ppop_ttm.iloc[0]) if normalize e
126
127     # Laba sbl pajak
128     ebt_b = df['laba sebelum pajak'].rename(f'{ticker} ebit b
129     columns.append((ebt_b/ebt_b.iloc[0]) if normalize else eb
130
131     # EBT ratio
132     ebitm_b = (ebt_b/rev_b).rename(f'{ticker} margin ebit bul
133     columns.append(ebitm_b)
134
135     # Laba bersih
136     laba_bersih_b = df['laba bersih'].rename(f'{ticker} laba
137     columns.append((laba_bersih_b/laba_bersih_b.iloc[0]) if n
138
139     laba_bersih_ttm = df['laba bersih_ttm'].rename(f'{ticker}
140     columns.append((laba_bersih_ttm/laba_bersih_ttm.iloc[0])
141
142     # NPM
143     npm_b = (laba_bersih_b/rev_b).rename(f'{ticker} NPM bulan
144     columns.append(npm_b)
145
146     npm_ttm = (laba_bersih_ttm/rev_ttm).rename(f'{ticker} NPM
147     columns.append(npm_ttm)
148
149     # ROE
150     roe_ttm = (laba_bersih_ttm / df['ekuitas']).rename(f'{tic
151     columns.append(roe_ttm)
152
153     # ROA
154     roa_ttm = (laba_bersih_ttm / (df['ekuitas'] + df['liabili
155     columns.append(roa_ttm)
156
157
158     df = pd.concat(columns, axis=1)
159     ylabel = 'Milyar (Rp)' if not normalize else 'Skala Normalisa
160     RH = 3
161     LL = 'upper left'
162
163     # Kredit Disalurkan, LDR, CKPN
164     charts = [
165         pt.Chart([pt.Plot(f'{ticker} kredit disalurkan') for tick
166                 title='Kredit Disalurkan', ylabel=ylabel, legend
167         pt.Chart([pt.Plot(f'{ticker} total CKPN') for ticker in t
168                 title='Total CKPN', ylabel=ylabel, legend_loc=LL
169         pt.Chart([pt.Plot(f'{ticker} LDR', linestyle='--') for ti
170                 title='Load to Deposit Ratio', ylabel='Rasio', l
171         pt.Chart([pt.Plot(f'{ticker} rasio CKPN', linestyle='--')
172                 title='Rasio CKPN thd Kredit', ylabel='Rasio', l
173
174     ]

```

```

175 fig = pt.Figure(charts, row_height=RH, title='Kredit Disalurk
176 fig.plot(df, tight_layout=True)
177
178 # NPL
179 charts = [
180     pt.Chart([pt.Plot(f'{ticker} Kol 2-5') for ticker in tick
181                 title='Total Kolektibilitas 2-5', ylabel=ylabel,
182     pt.Chart([pt.Plot(f'{ticker} Kol 3-5') for ticker in tick
183                 title='Total Kolektibilitas 3-5', ylabel=ylabel,
184     pt.Chart([pt.Plot(f'{ticker} NPL Gross', linestyle='--')
185                 title='NPL Gross', ylabel=ylabel, legend_loc=LL)
186     pt.Chart([pt.Plot(f'{ticker} NPL Net', linestyle='--') fo
187                 title='NPL Net', ylabel=ylabel, legend_loc=LL),
188     pt.Chart([pt.Plot(f'{ticker} rasio ckpn/kol25', linestyle
189                 title='Rasio CKPN/Kol 2-5', ylabel='Rasio', lege
190     pt.Chart([pt.Plot(f'{ticker} rasio ckpn/kol35', linestyle
191                 title='Rasio CKPN/Kol 3-5', ylabel='Rasio', lege
192 ]
193 fig = pt.Figure(charts, row_height=RH, title='Kolektibilitas,
194 fig.plot(df, tight_layout=True)
195
196 # Dana Simpanan
197 charts = [
198     pt.Chart([pt.Plot(f'{ticker} dana simpanan') for ticker i
199                 title='Total Dana Simpanan', ylabel=ylabel, lege
200     pt.Chart([pt.Plot(f'{ticker} CASA') for ticker in tickers
201                 title='CASA & E-Money', ylabel=ylabel, legend_lo
202     pt.Chart([pt.Plot(f'{ticker} Rasio CASA', linestyle='--')
203                 title='CASA Ratio', ylabel='Rasio', legend_loc=L
204 ]
205
206 fig = pt.Figure(charts, row_height=RH, title='Dana Simpanan',
207 fig.plot(df, tight_layout=True)
208
209 # Pendapatan bunga
210 charts = [
211     pt.Chart([pt.Plot(f'{ticker} pendapatan bunga ttm') for t
212                 title='Pendapatan Bunga TTM', ylabel=ylabel, leg
213     pt.Chart([pt.Plot(f'{ticker} pendapatan bunga bulanan') f
214                 title='Pendapatan Bunga Bulanan', ylabel=ylabel,
215
216     pt.Chart([pt.Plot(f'{ticker} beban bunga ttm') for ticker
217                 title='Beban Bunga TTM', ylabel=ylabel, legend_l
218     pt.Chart([pt.Plot(f'{ticker} beban bunga bulanan') for ti
219                 title='Beban Bunga Bulanan', ylabel=ylabel, lege
220
221     pt.Chart([pt.Plot(f'{ticker} GPM ttm', linestyle='--') fo
222                 title='Margin Pendapatan Bunga TTM', ylabel='Mar
223     pt.Chart([pt.Plot(f'{ticker} GPM bulanan', linestyle='--'
224                 title='Margin Pendapatan Bunga Bulanan', ylabel=
225
226     pt.Chart([pt.Plot(f'{ticker} GPM ttm', linestyle='--') fo
227                 title='Margin Pendapatan Bunga TTM', ylabel='Mar
228     pt.Chart([pt.Plot(f'{ticker} GPM bulanan', linestyle='--'
229                 title='Margin Pendapatan Bunga Bulanan', ylabel=
230
231     pt.Chart([pt.Plot(f'{ticker} NIM ttm', linestyle='--') fo
232                 title='NIM TTM', ylabel='Margin', legend_loc=LL)

```



```

233         pt.Chart([pt.Plot(f'{ticker} NIM bulanan annz', linestyle
234                        title='NIM Bulanan Annualized', ylabel='Margin',
235                ]
236    fig = pt.Figure(charts, row_height=RH, title='Pendapatan Bung
237    fig.plot(df, tight_layout=True)
238
239    # PPOP
240    charts = [
241        pt.Chart([pt.Plot(f'{ticker} PPOP ttm') for ticker in tic
242                  title='PPOP TTM', ylabel=ylabel, legend_loc=LL),
243        pt.Chart([pt.Plot(f'{ticker} PPOP bulanan') for ticker in
244                  title='PPOP Bulanan', ylabel=ylabel, legend_loc=
245                ]
246    fig = pt.Figure(charts, row_height=RH, title='Pre-Provision 0
247    fig.plot(df, tight_layout=True)
248
249    # beban CKPN
250    charts = [
251        pt.Chart([pt.Plot(f'{ticker} beban impairment ttm') for t
252                  title='Beban Impairment TTM', ylabel=ylabel, leg
253        pt.Chart([pt.Plot(f'{ticker} beban impairment bulanan') f
254                  title='Beban Impairment Bulanan', ylabel=ylabel,
255
256        pt.Chart([pt.Plot(f'{ticker} rasio bbn ckpn ttm', linestyle
257                  title='Rasio Beban Impairment thd Pendapatan Bun
258        pt.Chart([pt.Plot(f'{ticker} rasio bbn ckpn bulanan', lin
259                  title='Rasio Beban Impairment thd Pendapatan Bun
260
261    ]
262    fig = pt.Figure(charts, row_height=RH, title='Beban Impairmen
263    fig.plot(df, tight_layout=True)
264
265    # Operating profit
266    charts = [
267        pt.Chart([pt.Plot(f'{ticker} laba operasional ttm') for t
268                  title='Laba Operasional TTM', ylabel=ylabel, leg
269        pt.Chart([pt.Plot(f'{ticker} laba operasional bulanan') f
270                  title='Laba Operasional Bulanan', ylabel=ylabel,
271
272        pt.Chart([pt.Plot(f'{ticker} OPM ttm', linestyle='--') fo
273                  title='Operating Profit Margin TTM', ylabel='Ras
274        pt.Chart([pt.Plot(f'{ticker} OPM bulanan', linestyle='--'
275                  title='Operating Profit Margin Bulanan', ylabel=
276    ]
277    fig = pt.Figure(charts, row_height=RH, title='Laba Operasiona
278    fig.plot(df, tight_layout=True)
279
280    # Laba-laba
281    charts = [
282        pt.Chart([pt.Plot(f'{ticker} laba bersih ttm') for ticker
283                  title='Laba Bersih TTM', ylabel=ylabel, legend_l
284        pt.Chart([pt.Plot(f'{ticker} laba bersih bulanan') for ti
285                  title='Laba Bersih Bulanan', ylabel=ylabel, lege
286
287        pt.Chart([pt.Plot(f'{ticker} NPM ttm', linestyle='--') fo
288                  title='Net Profit Margin TTM', ylabel='Rasio', l
289        pt.Chart([pt.Plot(f'{ticker} NPM bulanan', linestyle='--'
290                  title='Net Profit Margin Bulanan', ylabel='Rasio

```



```

291 ]
292 fig = pt.Figure(charts, row_height=RH, title='Laba Bersih', n
293 fig.plot(df, tight_layout=True)
294
295 # Ratios
296 charts = [
297     pt.Chart([pt.Plot(f'{ticker} ROE ttm', linestyle='--') fo
298                 title='ROE TTM', ylabel='Rasio', legend_loc=LL),
299     pt.Chart([pt.Plot(f'{ticker} ROA ttm', linestyle='--') fo
300                 title='ROA TTM', ylabel='Rasio', legend_loc=LL),
301 ]
302 fig = pt.Figure(charts, row_height=RH, title='Rasio-Rasio', n
303 fig.plot(df, tight_layout=True)
304
305

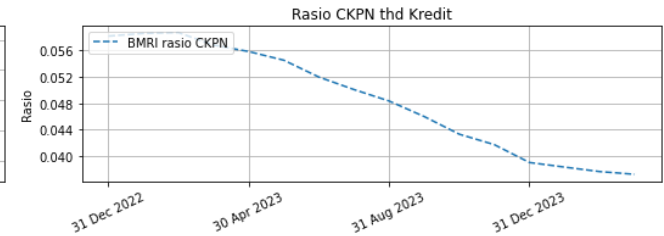
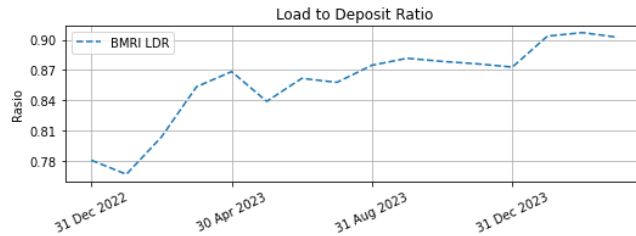
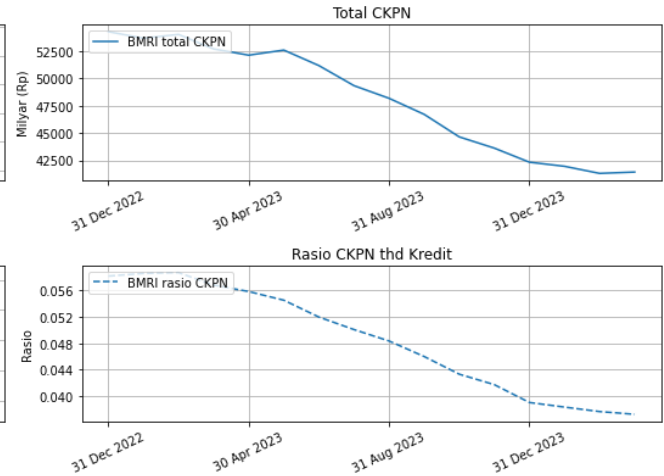
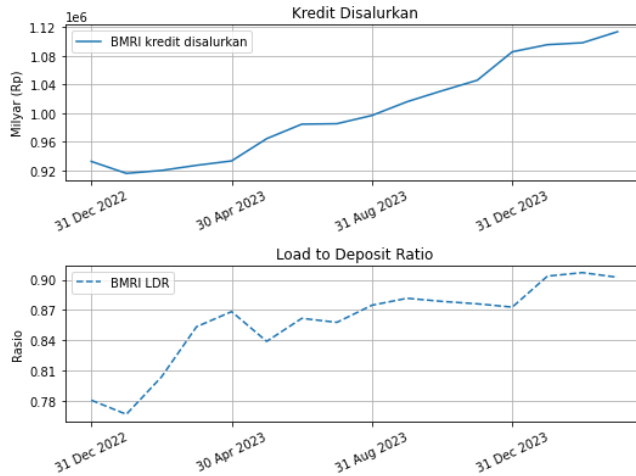
```

In [21]:

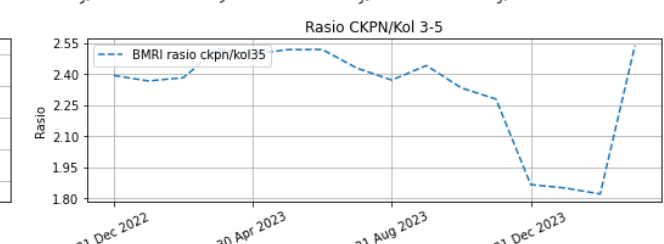
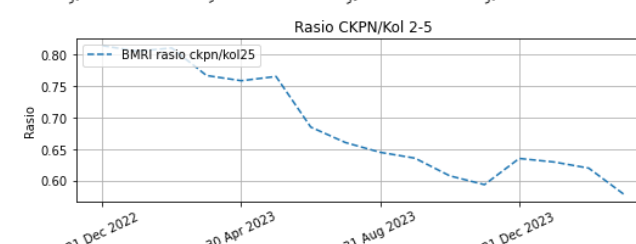
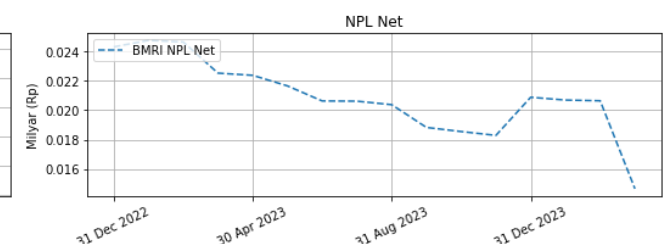
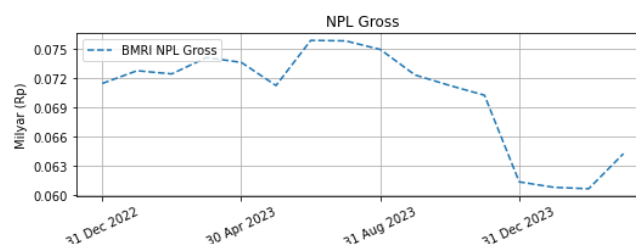
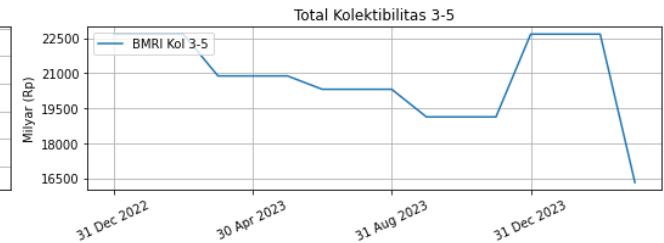
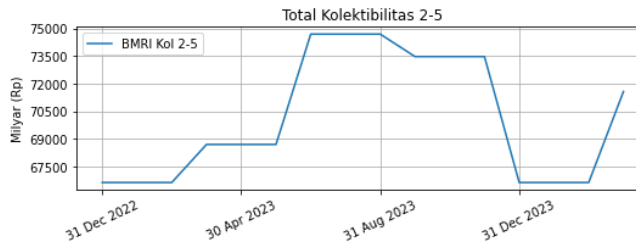
```
1 #plot_everything(normalize=True)
2 plot_everything(['BMRI'], normalize=False)
```

BMRI

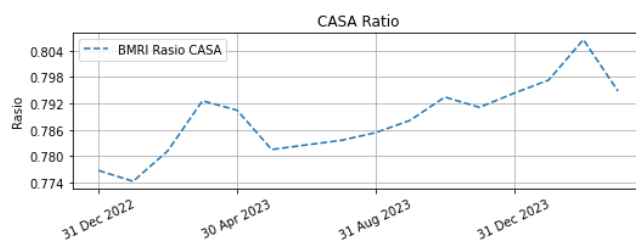
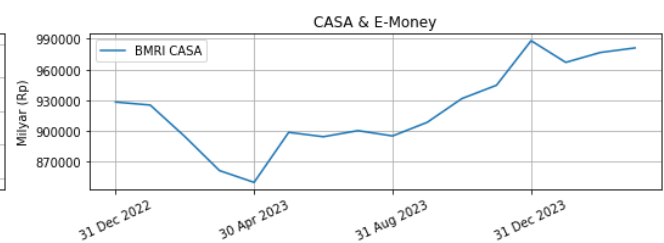
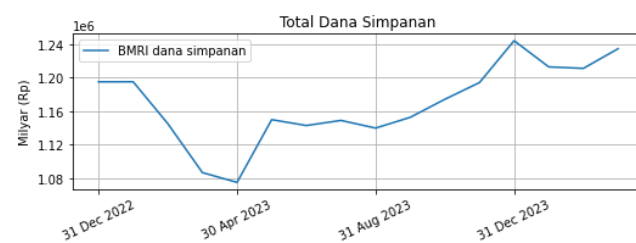
Kredit Disalurkan



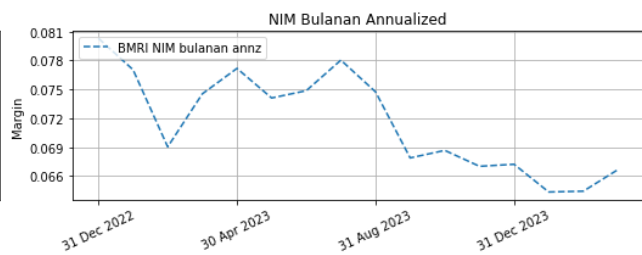
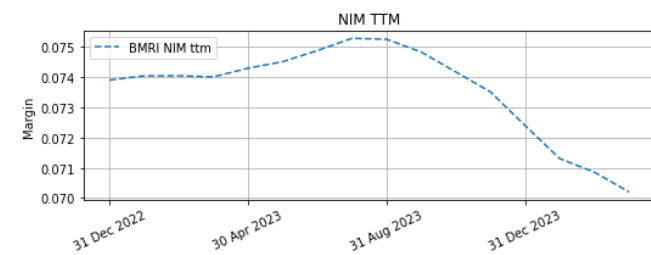
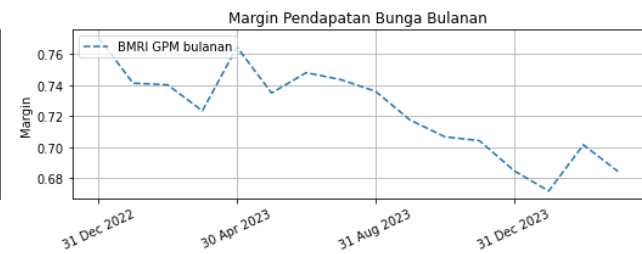
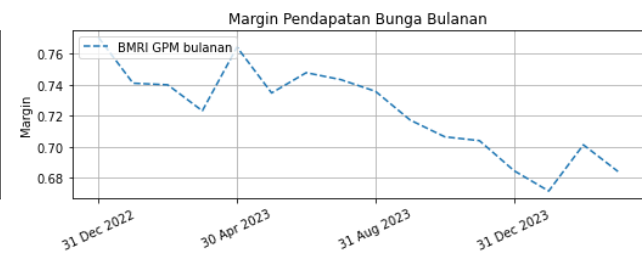
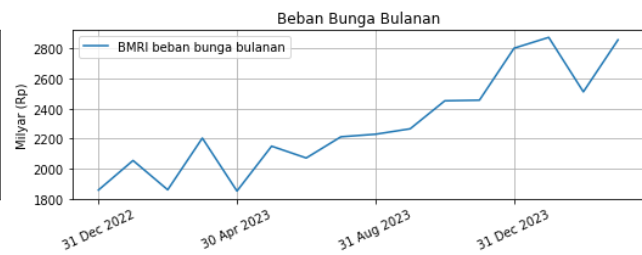
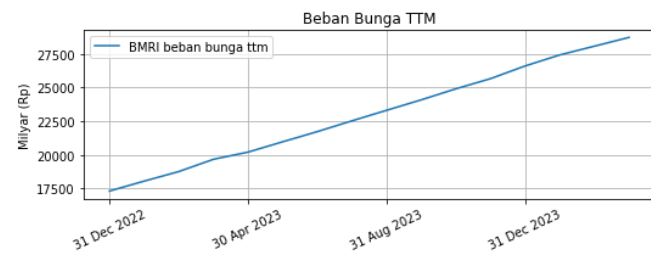
Kolektibilitas, NPL, CKPN



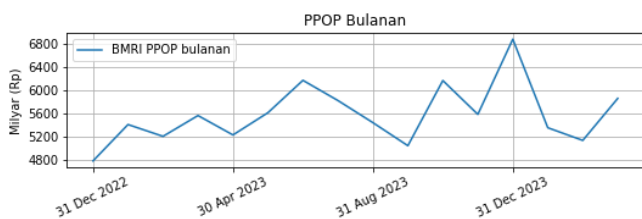
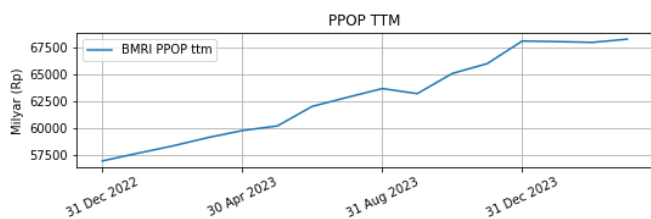
Dana Simpanan



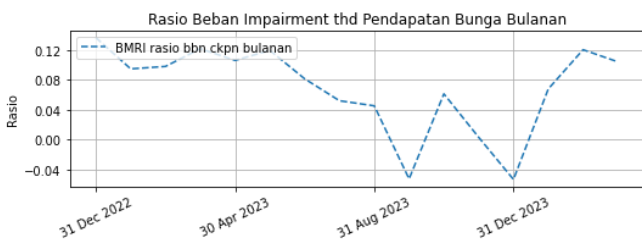
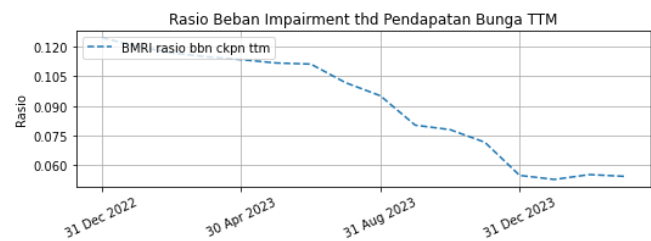
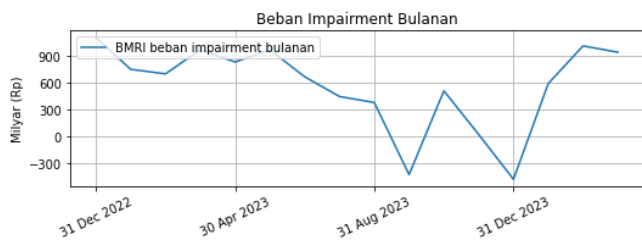
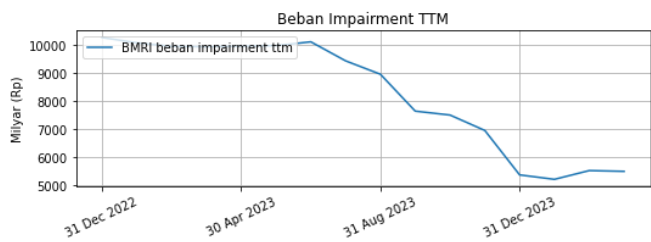
Pendapatan Bunga



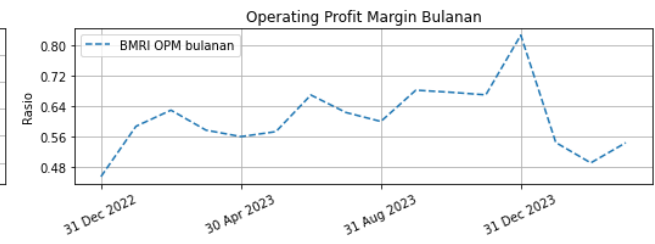
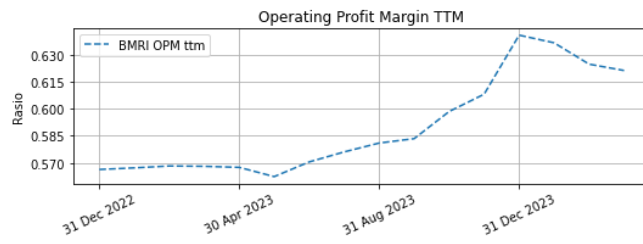
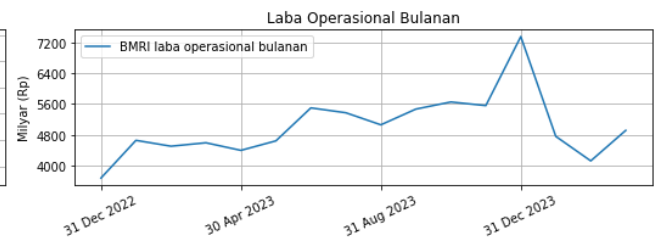
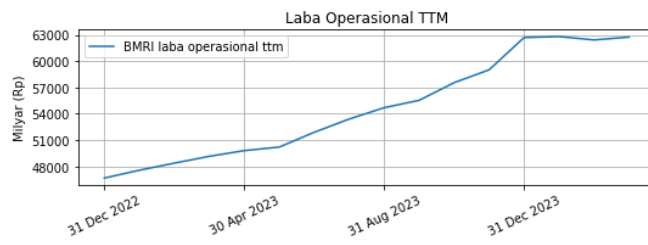
Pre-Provision Operating Profit (PPOP)



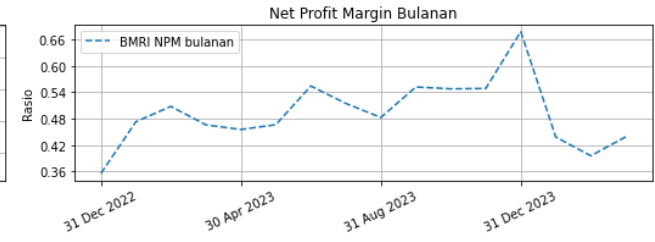
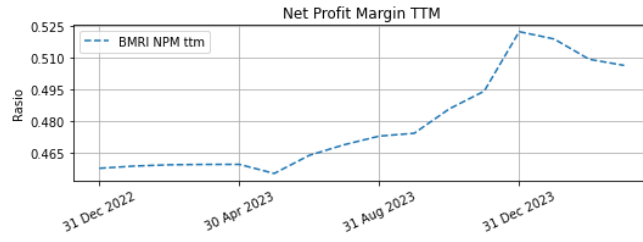
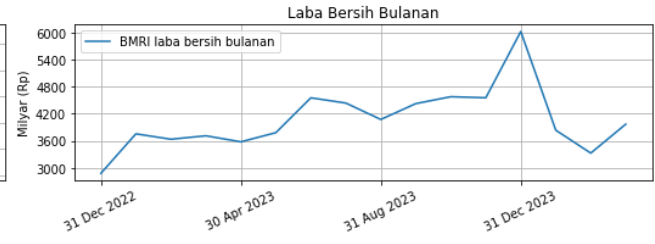
Beban Impairment



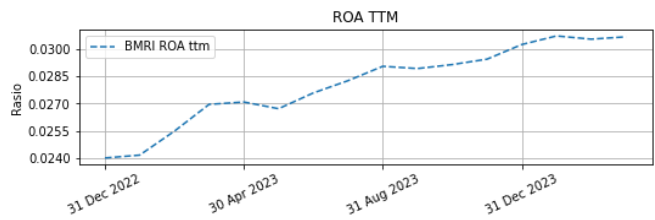
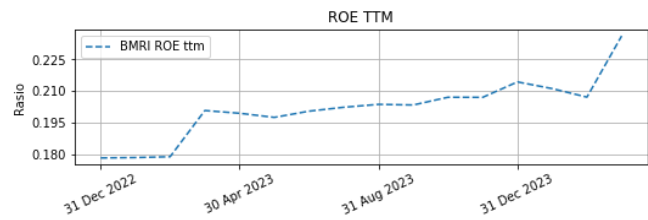
Laba Operasional



Laba Bersih



Rasio-Rasio



In []:

1