```
import pandas as pd
import seaborn as sns
from matplotlib import dates as mpl_date
import matplotlib.pyplot as plt


df = pd.read_csv("RAM.csv")


df.drop(columns=df.columns[0], axis=1, inplace=True)


df.describe()
```

| | Time_Elapsed | RAM_Precentage |
|---|---|---|
| **count** | 162472.000000 | 162471.000000 |
| **mean** | 8235.824516 | 21.626763 |
| **std** | 4752.126699 | 0.743362 |
| **min** | 0.000585 | 12.600000 |
| **25%** | 4121.890290 | 21.000000 |
| **50%** | 8236.790036 | 21.700000 |
| **75%** | 12350.227626 | 22.300000 |
| **max** | 16467.124169 | 23.100000 |

```
dfH = []
dfm = []
dfs = []
for index, row in df.iterrows():
    timestamp = row['Time']
    H,m,s = timestamp.split(':')
    dfH.append(H)
    dfm.append(m)
    dfs.append(s)

df['H'] = dfH
df['m'] = dfm
df['s'] = dfs

print(type(df))
df.head()
```

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

| | Time | Time_Elapsed | RAM_Precentage | H | m | s |
|---|---|---|---|---|---|---|
| **0** | 04:18:35 | 0.000585 | 12.6 | 04 | 18 | 35 |
| **1** | 04:18:35 | 0.101551 | 13.1 | 04 | 18 | 35 |
| **2** | 04:18:35 | 0.202569 | 13.2 | 04 | 18 | 35 |
| **3** | 04:18:35 | 0.303092 | 13.2 | 04 | 18 | 35 |
| **4** | 04:18:36 | 0.406815 | 13.2 | 04 | 18 | 36 |

```
Time = []
for index, row in df.iterrows():
    Time.append(row['H']+':'+row['m']+':'+row['s'])

newDF = pd.DataFrame()
newDF['Time'] = Time
newDF['RAM_Precentage'] = df['RAM_Precentage']
newDF['Time'] = pd.to_datetime(newDF['Time'])
newDF
```

```
<ipython-input-8-a2a4d2ff5a52>:8: UserWarning: Could not infer format, so each element will be parsed individually, falling back to
  newDF['Time'] = pd.to_datetime(newDF['Time'])
```

| | Time | RAM_Precentage |
|---|---|---|
| 0 | 2024-06-22 04:18:35 | 12.6 |
| 1 | 2024-06-22 04:18:35 | 13.1 |
| 2 | 2024-06-22 04:18:35 | 13.2 |
| 3 | 2024-06-22 04:18:35 | 13.2 |
| 4 | 2024-06-22 04:18:36 | 13.2 |
| ... | ... | ... |
| 162467 | 2024-06-22 08:53:02 | 22.9 |
| 162468 | 2024-06-22 08:53:02 | 22.9 |
| 162469 | 2024-06-22 08:53:02 | 22.9 |
| 162470 | 2024-06-22 08:53:02 | 22.9 |
| 162471 | 2024-06-22 08:53:02 | NaN |

162472 rows × 2 columns

```
!pip install pandasql
```

```
Collecting pandasql
  Downloading pandasql-0.7.3.tar.gz (26 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pandasql) (1.25.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from pandasql) (2.0.3)
Requirement already satisfied: sqlalchemy in /usr/local/lib/python3.10/dist-packages (from pandasql) (2.0.30)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->pandasql) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->pandasql) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->pandasql) (2024.1)
Requirement already satisfied: typing-extensions>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from sqlalchemy->pandasql) (4.1
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from sqlalchemy->pandasql) (3.0.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->pandasql)
Building wheels for collected packages: pandasql
  Building wheel for pandasql (setup.py) ... done
  Created wheel for pandasql: filename=pandasql-0.7.3-py3-none-any.whl size=26771 sha256=dbcf2352aae139e29414fde4ead8f3f101f8186c83
  Stored in directory: /root/.cache/pip/wheels/e9/bc/3a/8434bdcccf5779e72894a9b24fecbdcaf97940607eaf4bcdf9
Successfully built pandasql
Installing collected packages: pandasql
Successfully installed pandasql-0.7.3
```

```
from pandasql import sqldf
sql = lambda q: sqldf(q, globals())
newDF = sqldf("SELECT * FROM newDF WHERE Time BETWEEN '2024-06-22 04:18:35' AND '2024-06-22 04:19:35'")
newDF
```

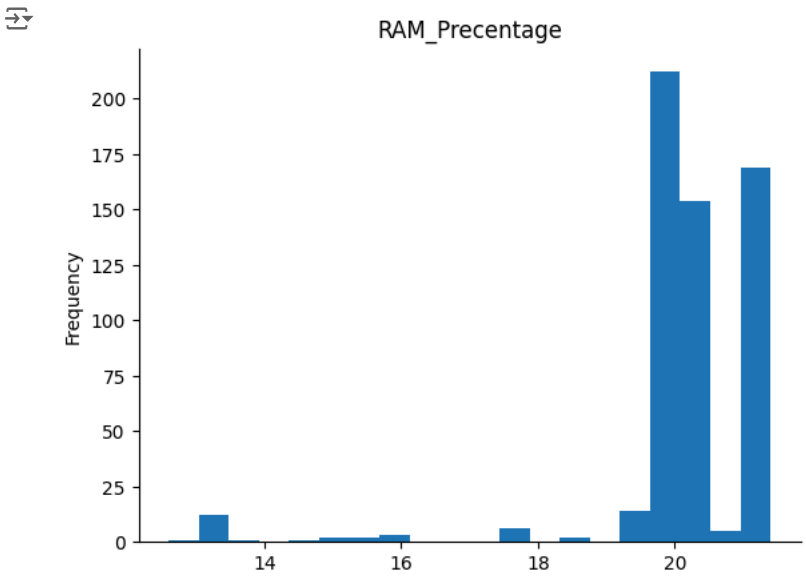
| | Time | RAM_Precentage |
|---|---|---|
| 0 | 2024-06-22 04:18:35.000000 | 12.6 |
| 1 | 2024-06-22 04:18:35.000000 | 13.1 |
| 2 | 2024-06-22 04:18:35.000000 | 13.2 |
| 3 | 2024-06-22 04:18:35.000000 | 13.2 |
| 4 | 2024-06-22 04:18:36.000000 | 13.2 |
| ... | ... | ... |
| 579 | 2024-06-22 04:19:34.000000 | 21.4 |
| 580 | 2024-06-22 04:19:34.000000 | 21.4 |
| 581 | 2024-06-22 04:19:34.000000 | 21.4 |
| 582 | 2024-06-22 04:19:34.000000 | 21.4 |
| 583 | 2024-06-22 04:19:34.000000 | 21.4 |

584 rows × 2 columns

Langkah berikutnya: Buat kode dengan newDF | Lihat plot yang direkomendasikan

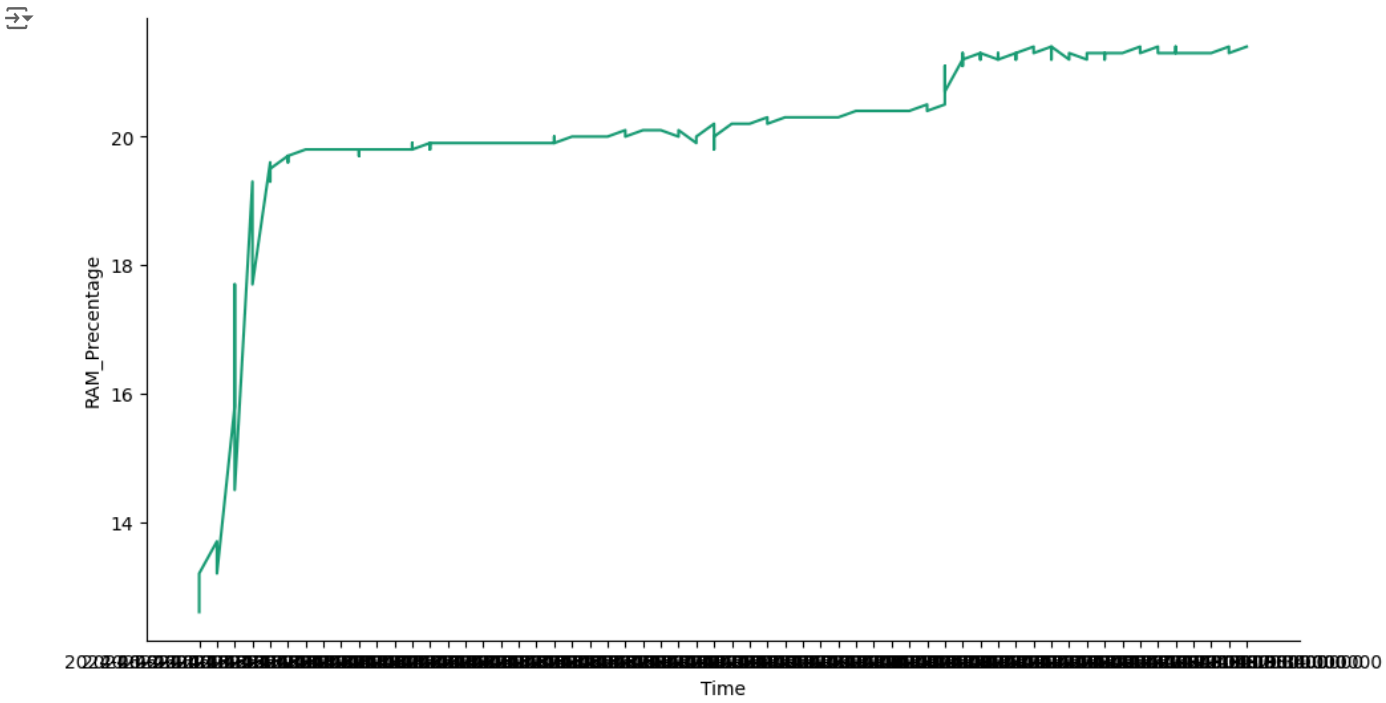
## RAM_Precentage

Tampilkan kode



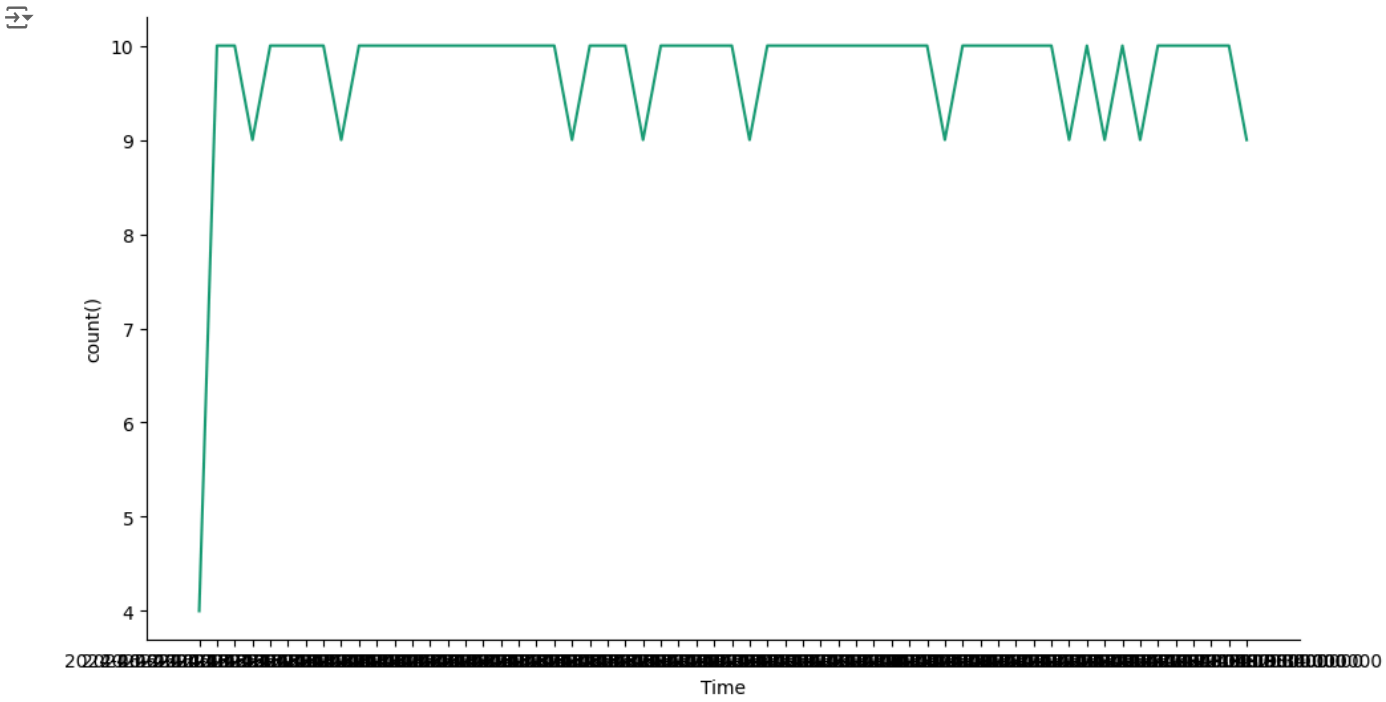

## Time vs RAM_Precentage

Tampilkan kode




## Time vs count()

Tampilkan kode




## RAM_Precentage

Tampilkan kode