

**TUGAS KONSEP APLIKASI DATA MAINING ASSOCIATION RULE
(MENGHITUNG SUPPORT DAN CONFIDANCE) MENGGUNAKAN JUPYTER
NETBOOK**



**Nama : Himatus Yulva Aprilia Sari
Jurusan :Sistem Infomasi
Nim :20.51.0018**

**KEMENTRIAN RISET DAN TEKNOLOGI PENDIDIKAN TINGGI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA KOMPUTER
PRADNYA PARAMITA
MALANG
2022**

Dalam aturan asosiasi terdapat 3 macam pengukuran penting, yaitu Support, Confidence, dan Lift Rasio. 3 asosiasi di atas sering di gunakan di **Association Rule**.

- Support adalah pencarian jumlah transaksi yang mengandung item berbanding dengan total transaksi Dan indikasi sering muncul di dataset.
- Confidence adalah nilai kepastian atau kuatnya hubungan antar item dalam aturan asosiasi.(indikasi seberapa sering aturan itu terbukti benar)
- Nilai lift rasio merupakan suatu ukuran dalam mengetahui kekuatan suatu aturan asosiasi. (indikasi nilai pengamatan Support , diharapkan jika dua aturan itu independen)

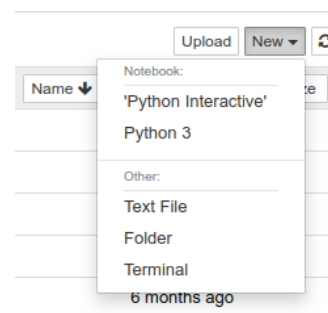
1. Halaman pertama Jupyter Notebook akan muncul di lokasi folder dimana kita menjalankan sintaks *jupyter notebook*.



The screenshot shows the Jupyter Notebook web interface. At the top left is the 'jupyter' logo. At the top right are 'Quit' and 'Logout' buttons. Below is a table listing various notebooks. The first notebook, 'TugasYulvaApriori.ipynb', is marked as 'Running' and shows a progress bar. The other notebooks are in various states (idle, running, or errored) and are listed with their names, last activity, and file sizes.

| Name | Status | Last Activity | Size |
|----------------------------------|---------|--------------------|---------|
| TugasYulvaApriori.ipynb | Running | 17 menit yang lalu | 21.2 kB |
| Twitter Sentiment Analysis.ipynb | | setahun yang lalu | 28.9 kB |
| Untitled.ipynb | | 2 tahun yang lalu | 3.78 kB |
| Untitled1.ipynb | | setahun yang lalu | 555 B |
| Untitled10.ipynb | | 4 bulan yang lalu | 4.49 kB |
| Untitled11.ipynb | | 4 bulan yang lalu | 11 kB |
| Untitled12.ipynb | | 4 bulan yang lalu | 577 B |
| Untitled13.ipynb | | 3 bulan yang lalu | 61.7 kB |
| Untitled14.ipynb | | 3 bulan yang lalu | 9.75 kB |
| Untitled15.ipynb | | 3 bulan yang lalu | 555 B |
| Untitled16.ipynb | | 3 bulan yang lalu | 9.97 kB |
| Untitled17.ipynb | | 3 bulan yang lalu | 8.99 kB |
| Untitled18.ipynb | | 3 bulan yang lalu | 8.85 kB |
| Untitled19.ipynb | | 3 bulan yang lalu | 11 kB |
| Untitled2.ipynb | | setahun yang lalu | 5.05 kB |
| Untitled20.ipynb | | 3 bulan yang lalu | 15.6 kB |
| Untitled21.ipynb | | 3 bulan yang lalu | 3.3 kB |
| Untitled3.ipynb | | setahun yang lalu | 1.86 kB |
| Untitled4.ipynb | | setahun yang lalu | 555 B |

2. Di Jupyter Notebook kita bisa membuat file dan folder seperti kita membuatnya di halaman sistem operasi seperti Windows/Linux/Mac

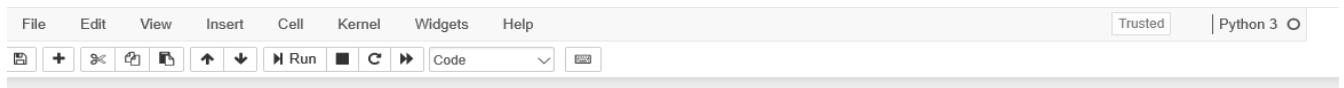


3. Saat membuat file atau folder di Jupyter Notebook secara otomatis akan bernama ‘Untitled’ dan bisa di ubah sesuai Tugas.

☐ TugasYulvaApriori.ipynb

☐ Twitter Sentiment Analysis.ipynb

4. Toolbar mempunyai beberapa tombol shortcuts penting.



Dari kiri ke kanan:

save and checkpoint = menyimpan file dan melakukan checkpoint
insert cell below = menambah cell dibawah cell aktif
cut selected cell = memotong cell aktif
copy selected cell = menyalin cell aktif
paste cell below = menempel ke cell dibawah cell aktif
move selected cell up = memindah cell aktif ke atas
move selected cell down = memindah cell aktif ke bawah
run = menjalankan cell aktif
interupt kernel = interupsi kernel
restart the kernel = memulai ulang kernel
restart the kernel and re-run the whole notebook = memulai ulang kernel dan menjalankan seluruh kode notebook
cell type = Tipe cell (code / markdown)
command palette = shortcuts perintah

5. Yang pertama adalah melakukan import library, yang pertama adalah pandas nanti di sebut sebagai pd, matplotlib.pyplot nanti di sebut sebagai plt, Kemudian mlxtend melakukan import fungsi apriori dan association_rules.

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from apyori import apriori
```

6. Pertama-tama adalah membaca file excel menggunakan library pandas. Dimana nama file excel yang di gunakan adalah 'data.xls' file ini di simpan 1 folder dengan lokasi script yang sedang kita jalankan, jika sudah berhasil maka akan muncul head tabel dari data tersebut.

```
store_data= pd.read_excel('data.xls')
```

```
store_data.head()
```

Output:

| | Barang 1 | Barang 2 | Barang 3 | Barang 4 | Barang 5 | Barang 6 |
|---|----------|----------|----------|----------|----------|----------|
| 0 | Wine | Chips | Bread | Butter | Milk | Apple |
| 1 | Wine | NaN | Bread | Butter | Milk | NaN |
| 2 | NaN | NaN | Bread | Butter | Milk | NaN |
| 3 | NaN | Chips | NaN | NaN | NaN | Apple |
| 4 | Wine | Chips | Bread | Butter | Milk | Apple |

7.Selanjutnya ingin melihat baris 5 yang paling bawah sesuai data yang kita buat.

```
store_data.tail()
```

Output:

| | Barang 1 | Barang 2 | Barang 3 | Barang 4 | Barang 5 | Barang 6 |
|----|----------|----------|----------|----------|----------|----------|
| 17 | NaN | Chips | NaN | Butter | Milk | Apple |
| 18 | Wine | Chips | Bread | Butter | Milk | Apple |
| 19 | Wine | NaN | Bread | Butter | Milk | Apple |
| 20 | Wine | Chips | Bread | NaN | Milk | Apple |
| 21 | NaN | Chips | NaN | NaN | NaN | NaN |

8. Selanjutnya jika kita ingin menemukan jumlah number recordnya yang diexecute oleh Library apriori. Jalankan skrip berikut:

```
store_data.shape
```

(22, 6)

9. Selajutnya, Library Apriori yang akan kita gunakan membutuhkan dataset dalam bentuk list di dalam list, di mana seluruh dataset adalah daftar besar dan setiap transaksi dalam dataset adalah List dalam di dalam List yang besar luar. Saat ini kami memiliki dataset dalam bentuk dataset pandas. Untuk mengonversi dataset dalam pandas kami ke List di dalam list, jalankan skrip berikut:

```
records = []
for i in range (0, 22):
    records.append([str(store_data.values[i,j])for j in range (0,6)])
```

10. Langkah selanjutnya adalah menerapkan algoritma Apriori pada dataset. Untuk melakukannya, kita bisa menggunakan kelas apriori yang kita impor dari perpustakaan apyori. Jalankan skrip berikut :

```
association_rules = apriori(records, min_support=0.08, min_confidence=0.9, min_lift=1.1, min_length=4)
association_results = list(association_rules)
```

11. Pertama mari kita temukan jumlah total aturan yang diexecute oleh Library apriori. Jalankan skrip berikut:

```
print(len(association_results))
```

15

12. Selanjutnya mari menampilkan item pertama di association_rules untuk melihat apa aturan pertama. Jalankan skrip berikut:

```
print(association_results[0])
```

Outputnya:

```
RelationRecord(items=frozenset({'Bread', 'Wine', 'Apple'}), support=0.45454545454545453, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Apple'}), items_add=frozenset({'Bread'}), confidence=0.9090909090909091, lift=1.25)])
```

13. Script untuk menampilkan support, confidence, dan lift, maka jalankan script dibawah ini:

```
results = []
for item in association_results:
    pair = item[0]
    items = [x for x in pair]

    value0 = str(items[0])
    value1 = str(items[1])
    value2 = str(item[1])[0:6]
    value3 = str(item[2][0][2])[0:6]
    value4 = str(item[2][0][3])[0:6]

    rows = (value0, value1, value2, value3, value4)

    results.append(rows)

label = ['title1'], ['title2'], ['support'], ['confidence'], ['lift']
store_suggestion = pd.DataFrame.from_records(results, columns=label)

print(store_suggestion)
```

| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
|-----|-----------|-----------|------------|---------------|---------|
| 0 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
| 0 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 1 | Bread | Butter | 0.4090 | 0.9000 | 1.3200 |
| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
| 0 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 1 | Bread | Butter | 0.4090 | 0.9000 | 1.3200 |
| 2 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
| 0 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 1 | Bread | Butter | 0.4090 | 0.9000 | 1.3200 |
| 2 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 3 | Wine | Milk | 0.4545 | 0.9090 | 1.1764 |
| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
| 0 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 1 | Bread | Butter | 0.4090 | 0.9000 | 1.3200 |
| 2 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 3 | Wine | Milk | 0.4545 | 0.9090 | 1.1764 |
| ... | ... | ... | ... | ... | ... |
| 11 | Bread | Wine | 0.2727 | 1.0 | 1.375 |
| 12 | Bread | Wine | 0.1363 | 1.0 | 1.375 |
| 13 | Apple | Bread | 0.1818 | 1.0 | 1.375 |
| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
| 0 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 1 | Bread | Butter | 0.4090 | 0.9000 | 1.3200 |
| 2 | Bread | Wine | 0.4545 | 0.9090 | 1.25 |
| 3 | Wine | Milk | 0.4545 | 0.9090 | 1.1764 |
| 4 | Chips | Milk | 0.4090 | 0.9000 | 1.1647 |
| 5 | Bread | Wine | 0.2727 | 1.0 | 1.375 |
| 6 | Bread | Wine | 0.3181 | 1.0 | 1.375 |
| 7 | Bread | Milk | 0.4090 | 0.9000 | 1.1647 |
| 8 | Apple | Bread | 0.2272 | 1.0 | 1.375 |
| 9 | Apple | Bread | 0.2272 | 1.0 | 1.375 |
| 10 | Apple | Bread | 0.0909 | 1.0 | 1.375 |
| 11 | Bread | Wine | 0.2727 | 1.0 | 1.375 |
| 12 | Bread | Wine | 0.1363 | 1.0 | 1.375 |
| 13 | Apple | Bread | 0.1818 | 1.0 | 1.375 |
| 14 | Bread | Wine | 0.0909 | 1.0 | 1.375 |

14.Selanjutnya jalankan script berikut ini untuk menampilkan kesimpulan

```
store_suggestion.describe()
```

Output:

| | (title1,) | (title2,) | (support,) | (confidence,) | (lift,) |
|--------|-----------|-----------|------------|---------------|---------|
| count | 15 | 15 | 15 | 15 | 15 |
| unique | 4 | 4 | 8 | 3 | 5 |
| top | Bread | Wine | 0.4090 | 1.0 | 1.375 |
| freq | 9 | 7 | 3 | 9 | 9 |

15.Selanjutnya meneksport hasilnya ke bentuk excel dengan menggunakan script berikut ini :

```
In [26]: store_suggestion.to_excel('C:/Users/yulvavi/outputsyulvatugas.xls')
```

Kesimpulan:

Algoritma Association rule mining seperti Apriori sangat fleksibel dan mudah di gunakan karena sangat sederhana . Mereka mudah diimplementasikan dan memiliki kemampuan menjelaskan yang tinggi. Pada tugas ini saya mencoba (min_support=0.08, min_confidence=0.9, min_lift=1.1, min_length=24) dan mendapatkan 15 list dan yang tertinggi adalah bread dan bersamaan dengan Wine dengan Support 0,4090 dan Confidence 1.0.