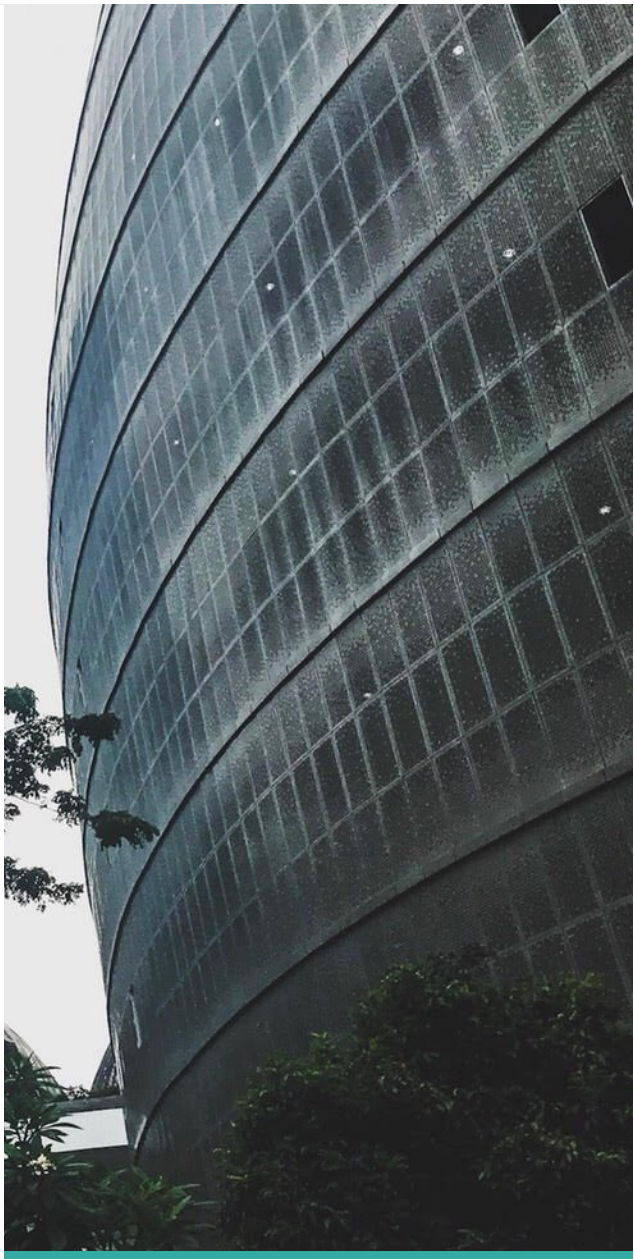
MODUL PRAKTIKUM

IS411 – DATA MODELING
PROGRAM SARJANA S1 SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA



PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA

Gedung B Lantai 5, Kampus UMN
Jl. Scientia Boulevard, Gading Serpong, Tangerang, Banten-15811 Indonesia
Telp: +62-21.5422.0808 (ext. 1803), email: ict.lab@umn.ac.id, web: umn.ac.id

MODUL 3
DM METHODOLOGY

DESKRIPSI TEMA

1. Text Data Preprocessing using Regular Expression (REGEX) using Python

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Students are able to implement Data Modeling Methodology (C_3)

PENUNJANG PRAKTIKUM

1. Anaconda Navigator
2. Jupyter Notebook

(+ Perlengkapan Apd/Alat Pelindung Diri Yang Harus Digunakan, Jika Ada)

LANGKAH-LANGKAH PRAKTIKUM

Import Library

1. Lakukan langkah yang sama seperti minggu sebelumnya untuk meng-import library Numpy dan Pandas.

Membaca Data

2. Gunakan file pendukung yang diberikan, lalu masukkan menjadi dataframe dengan nama "dataset". Perhatikan bahwa file yang digunakan adalah format .csv. (Petunjuk: tambahkan delimeter = ';')
3. Menggunakan function info(), untuk menampilkan deskripsi dari dataset sehingga akan terlihat kolom apa saja yang memiliki data NULL, NaN, maupun None.

```
1 dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Description  522533 non-null  object
 3   Quantity     525461 non-null  int64
 4   InvoiceDate  525461 non-null  object
 5   Price        525461 non-null  object
 6   Customer ID  417534 non-null  float64
 7   Country      525461 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

4. Adapun cara lain untuk menemukan missing value dapat menggunakan function **isnull()** sebagai berikut.

```
1 #menampikan informasi untuk mengetahui kolom/atribut data yang memiliki data null (empty) dalam barisnya
2
3 print(dataset.isnull().sum())
```

```
Invoice            0
StockCode          0
Description     2928
Quantity           0
InvoiceDate        0
Price              0
Customer ID    107927
Country            0
dtype: int64
```

5. Dari informasi di atas diketahui bahwa kolom 'Description' memiliki 2928 data null dan kolom 'Customer ID' memiliki 107927' data null, sedangkan kolom lainnya tidak ada null.

   Untuk menghapus kolom dari dataset , dapat menggunakan function **dropna()**. Untuk menghapus kolom , tambahkan attribut axis pada function dropna(), dimana axis = 1 untuk kolom dan axis = 0 untuk row. Apabila tidak di specified, maka defaultnya adalah axis = 0.

```
#menghapus kolom yang memiliki data null

new_dataset = dataset.dropna(axis = 1)
new_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 6 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Quantity     525461 non-null  int64
 3   InvoiceDate  525461 non-null  object
 4   Price        525461 non-null  object
 5   Country      525461 non-null  object
dtypes: int64(1), object(5)
memory usage: 24.1+ MB
```

Perhatikan bahwa kolom 'Description' dan 'Customer ID' sudah dihapus karena memiliki data null.

6. Lakukan langkah berikut untuk menghapus baris yang memiliki data null.

```
#menghapus baris yang memiliki data null

new_dataset2 = dataset.dropna(axis = 0)
new_dataset2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 417534 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      417534 non-null  object
 1   StockCode    417534 non-null  object
 2   Description  417534 non-null  object
 3   Quantity     417534 non-null  int64
 4   InvoiceDate  417534 non-null  object
 5   Price        417534 non-null  object
 6   Customer ID  417534 non-null  float64
 7   Country      417534 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 28.7+ MB
```

Dengan menghapus baris yang memiliki data null, semua kolom tetap ada tetapi jumlah baris data berkurang karena data yang null terhapus.

7. Untuk cara ketiga, dapat menggunakan function **fillna()**, yaitu mengisi data yang kosong dengan suatu nilai (value). Adapun value yang akan ditambah kedalam kolom dapat berupa default value,

mean value, atau dengan conditional. Cara ini juga dapat digunakan untuk mengisi hanya 1 kolom atau semua kolom.

```
#mengisi data null dengan nilai 0

default_dataset = dataset.fillna(0)
default_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Description  525461 non-null  object
 3   Quantity     525461 non-null  int64
 4   InvoiceDate  525461 non-null  object
 5   Price        525461 non-null  object
 6   Customer ID  525461 non-null  float64
 7   Country      525461 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

Perhatikan bahwa kolom yang memiliki data null sudah memiliki value. Cara ini bisa jadi lebih baik daripada menghapus data null karena data tidak akan hilang.

8. Cara lain mengisi data ke hanya kolom yang memiliki data null, yaitu 'Description' dan 'Customer ID'.

```
#mengisi bagian null pada kolom 'Description' dengan 'no description'

default_dataset_spec = dataset.copy()
default_dataset_spec['Description'] = default_dataset_spec['Description'].fillna('no description')
```

```
#mengisi bagian null pada kolom 'Customer ID' dengan 0

default_dataset_spec['Customer ID'] = default_dataset_spec['Customer ID'].fillna(0)
```

```
#menampilkan informasi data

default_dataset_spec.info()
```

9. Selain dengan value 0, kita dapat mengisi data yang kosong dengan nilai lain. Berikut adalah contoh mengisi data menggunakan mean dari data yang ada. Karena kolom dari dataset saat ini tidak cocok untuk diisi dengan mean, tambahkan satu kolom kosong dengan value nan (menggunakan function nan dari numpy) untuk diisi dengan value mean dari data yang lain.

```
#membuat kolom baru bernama 'mean_dataset'

mean_dataset = dataset.copy()
mean_dataset['mean_quantity'] = np.nan
mean_dataset.info()
```

```
#mengisi kolom 'mean_quantity' dengan nilai rata - rata (mean) dari kolom 'Quantity'

mean_dataset['mean_quantity'] = mean_dataset['mean_quantity'].fillna(mean_dataset['Quantity'].mean())
mean_dataset.info()
mean_dataset
```

Perhatikan perbedaan pada kolom ‘mean quantity’ sebelum dan setelah diisi dengan data. Jumlah data non-null berubah.

10. Kita juga dapat mengisi data null dengan berdasarkan kondisi tertentu. Misalnya kita ingin mengisi data null pada kolom 'Description' dengan value 'bulk product' jika nilai pada kolom 'Quantity' lebih besar dari 10, selain itu akan diisi dengan value 'not clear'.

```
data_cond = dataset.copy()

data_cond['Description'] = np.where(np.logical_and(data_cond['Description'].isnull(), data_cond['Quantity'] > 10), 'bulk',
                                    np.where(data_cond['Description'].isnull(), 'not clear', data_cond['Description']))
data_cond.info()

```

```
#memastikan data sudah terisi sesuai kondisi yang ditentukan

data_cond[data_cond['Description'] == 'bulk']
```

```
data_cond[data_cond['Description'] == 'not clear']
```

Manipulasi Data menggunakan Regex (Regular Expression)

Regular expression (Regex) adalah beberapa karakter yang dapat membentuk atau sering digunakan untuk pencarian dari sebuah teks, seperti pada function "find" atau "find and replace".

11. Pada Python, untuk memproses regular expression sudah disediakan library khusus dan dapat dipanggil dengan import **re**.

```
#mengimport library Regex

import re
```

Beberapa function yang terdapat pada library re adalah:

- **re.findall()** mengembalikan list kata yang ditemukan dalam sebuah teks atau kalimat.

- **re.search()** untuk menjadi kata atau huruf tertentu, hanya mengembalikan kata pertama kali ditemukan dari semua kata yang mungkin cocok.
- **re.split()** untuk memisahkan sebuah kata berdasarkan separator yang ditentukan.
- **re.sub()** untuk mengganti beberapa teks menjadi teks yang lain.

12. Menggunakan **re.findall()**

```
var = "Contoh teks yang digunakan untuk uji coba function dari library re, pencarian teks, memisah teks, dan mengganti teks"

print(re.findall('teks', var))
```

13. Menggunakan **re.search()**

```
print(re.search('teks', var))
```

14. Menggunakan **re.split()**

```
print(re.split(' ',var))
```

15. Regular expression pada Pandas memiliki kemiripan dengan regular expression pada python, berikut adalah contoh penggunaan regular expression pada Pandas.

```
dataset.info()
dataset
```

```
#function count dapat digunakan untuk menghitung jumlah data yang muncul dalam satu kolom.

dataset['Description'].str.count('LIGHTS').sum()
```

CONTAINS vs. MATCH

16. Untuk mencari data yang memliki kata mirip atau persis, dapat menggunakan **match**(), sedangkan mencari kata yang mengandung atau memiliki kata tertentu dapat menggunakan **contains()**.

```
data_contain = dataset['Description'].str.contains('LIGHTS')
print(data_contain)
```

```
data_match = data['Description'].str.match('LIGHTS')
print(data_match)
```

```
data_match2 = dataset['Description'].str.match('FELTCRAFT PRINCESS LOLA DOLL')
print(data_match2)
```

Jelaskan perbedaan hasil output menggunakan **contains()** dan **match()!**

17. **Replace** memiliki fungsi yang sama seperti **re.sub()**, yaitu untuk mengganti teks yang sesuai ditentukan menjadi teks lain.

```
data_replace = dataset['Description'].str.replace('LIGHTS','LAMP')
data_replace
```

18. Pandas juga memiliki function **split** yang fungsinya seperti **re.split()** dan function split lain yang berkerja dari belakang yaitu **rsplit()**. Secara hasil, jika hanya untuk memisahkan kata akan sama.

```
#menggunakan split()

data_split = data['Description'].str.split()
print(data_split)
```

```
#menggunakan resplit()

data_resplit = dataset['Description'].str.rsplit()
print(data_resplit)
```

PENGUMPULAN

1. File yang dikumpulkan terdiri dari:
    a. File project (.ipynb)
    b. File PDF berisi screenshot output dan jawaban (jika ada pertanyaan)
2. File di-compress (.zip) dan diberi nama **KODEMATAKULIAH_KELAS_NIM_NAMA_WEEK KE-XX.zip** (contoh: IS5411_A_13110310017_Monika Evelin Johan_Week-01.zip).

REFERENSI