



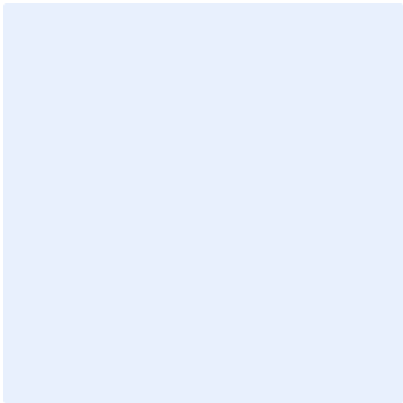
LEMBAR SAMPUL PENGAJUAN TUGAS

Nama Mahasiswa:	Javan Visman
Nomor Mahasiswa:	201855202042
Nama Mata Kuliah:	Data Mining
Nama Pengajar:	MUHAMMAD YUSUF, S.Kom., M.Kom.
Judul Tugas:	UTS
Penugasan Ke:	1
Tanggal Batas Pengumpulan:	29 Mei 2023

PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Tugas ini sepenuhnya adalah hasil pekerjaan saya sendiri. Selain yang tertera pada daftar pustaka, materi yang terkandung dalam tugas ini belum pernah dikirimkan untuk penilaian dalam tugas apa pun.
2. Saya telah memiliki salinan dari tugas ini dan jika diperlukan saya dapat menunjukkan salinan dari tugas tersebut.
3. Saya telah memahami definisi dan konsekuensi dari plagiarisme.

Tanda Tangan Mahasiswa: 	Tanggal: Senin, 29 Mei 2023
---	--



Apa itu Preprocessing Data?

data preprocessing adalah teknik yang digunakan untuk mengubah data mentah dalam format yang berguna dan efisien. Inisiatif ini diperlukan karena data mentah seringkali tidak lengkap dan memiliki format yang tidak konsisten. Kualitas data sendiri memiliki korelasi langsung dengan keberhasilan setiap proyek yang melibatkan analisis data. *Preprocessing* sendiri melibatkan validasi dan imputasi data. Tujuan dari validasi adalah untuk menilai tingkat kelengkapan dan akurasi data yang tersaring.

Dalam *machine learning*, kegiatan ini sangat penting untuk memastikan bahwa big data sudah diformat dan informasi yang dikandungnya dapat ditafsirkan dan algoritma perusahaan.

Berikut adalah keempat tahap kerja *data preprocessing* :

1. *Data cleaning*

Dalam tahap kerja ini, data dibersihkan melalui beberapa proses seperti mengisi nilai yang hilang, menghaluskan *noisy data*, dan menyelesaikan inkonsistensi yang ditemukan. Data juga bisa dibersihkan dengan dibagi menjadi segmen-segmen yang memiliki ukuran serupa lalu dihaluskan (*binning*). Kamu juga bisa menyesuaikannya dengan fungsi regresi linear atau berganda (*regression*), atau dengan mengelompokkannya ke dalam kelompok-kelompok data yang serupa (*grouping*).

2. *Data integration*

Tahap kerja berikutnya dalam proses data preprocessing adalah data integration. Di sini, data dengan representasi yang berbeda disatukan dan semua konflik dalam di dalamnya diselesaikan. Tahap kerja satu ini merupakan proses lanjutan dari data cleaning dengan tujuan untuk membuat data lebih halus.

3. *Data transformation*

Data transformation adalah tahap kerja selanjutnya dalam proses data preprocessing. Pada tahap ini, data akan dinormalisasi dan digeneralisasikan. Normalisasi sendiri adalah sebuah proses di mana perusahaan memastikan bahwa tidak ada data yang berlebihan. Semua data akan disimpan dalam satu tempat dan semua dependensinya haruslah logis. Langkah ini juga diambil untuk mentransformasikan data ke dalam bentuk yang sesuai untuk proses mining.



4. Data reduction

Tahap kerja terakhir dalam proses kerja data preprocessing adalah data reduction. Data mining adalah sebuah teknik yang digunakan untuk menangani data dalam jumlah yang besar. Saat bekerja dengan volume data yang besar, proses analisis akan menjadi lebih sulit. Nah, untuk mempermudah proses data mining, kamu bisa menggunakan teknik data reduction. Sebab, menurut Monkey Learn, inisiatif ini bisa meningkatkan efisiensi penyimpanan dan mengurangi representasi data dalam data warehouse.

Berikut adalah mengolah Data baru menggunakan googlescollab :

Import library dan dataset yang digunakan

```
[1] import pandas as pd
import numpy as np

[2] df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/data-mining/Dataset.xlsx')
df.head()
```

	Tanggal	Tavg	RH_avg	RR	ff_avg
0	01-03-2020	28.5	84.0	8888.0	2
1	02-03-2020	NaN	NaN	6.4	2
2	03-03-2020	28.8	81.0	3.4	3
3	04-03-2020	27.8	84.0	8888.0	2
4	05-03-2020	28.3	81.0	0.7	2

Sebelumnya kita import dahulu library yang akan dipakai seperti pandas dan numpy, kemudian akses dataset yang akan diolah. Disini dataset yang akan diolah telah saya upload pada G-Drive agar lebih mudah untuk pengaksesan datasetnya.

Menampilkan info dan missing value pada data

```
[3] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 5 columns):
#   Column    Non-Null Count  Dtype  
---  -
0   Tanggal   31 non-null     object  
1   Tavg      27 non-null     float64  
2   RH_avg    27 non-null     float64  
3   RR        21 non-null     float64  
4   ff_avg    31 non-null     int64  
dtypes: float64(3), int64(1), object(1)
memory usage: 1.3+ KB

df.isna().sum()

Tanggal    0
Tavg       4
RH_avg     4
RR        10
ff_avg     0
dtype: int64
```

Selanjutnya kita tampilkan info pada dataset menggunakan perintah `df.info()` dan cek missing values atau nilai-nilai yang hilang pada dataframe/dataset kita menggunakan perintah `df.isna().sum()`. Dari sini dapat kita ketahui jumlah kolom dan baris hingga type data pada dataset kita dan juga jumlah missing values yang terdapat pada data kita.

Mengubah nama atribut

```
[5] df = df.rename(columns = {"Tavg": "Suhu",
                             "RH_avg": "Kelembapan",
                             "RR": "Curah_Hujan",
                             "ff_avg": "Kec_Angin"})

[6] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   Tanggal     31 non-null    object  
1   Suhu        27 non-null    float64  
2   Kelembapan  27 non-null    float64  
3   Curah_Hujan 21 non-null    float64  
4   Kec_Angin  31 non-null    int64  
dtypes: float64(3), int64(1), object(1)
memory usage: 1.3+ KB
```

Berikutnya kita ubah dahulu nama atribut atau kolom pada datasetnya agar lebih mudah dipahami, dimana atribut yang kita ubah namanya adalah Tavg, RH_avg, RR, ff_avg menjadi Suhu, Kelembapan, Curah_Hujan, Kec_Angin.

Menangani missing value

```
[7] df.isna().sum()

Tanggal      0
Suhu         4
Kelembapan   4
Curah_Hujan 10
Kec_Angin    0
dtype: int64

[8] df["Suhu"].fillna(df["Suhu"].median(), inplace=True)
df["Kelembapan"].fillna(df["Kelembapan"].median(), inplace=True)
df["Curah_Hujan"].fillna(df["Curah_Hujan"].median(), inplace=True)

[9] df.isna().sum()

Tanggal      0
Suhu         0
Kelembapan   0
Curah_Hujan 0
Kec_Angin    0
dtype: int64
```

Pada tahap ini kita akan menangani missing values atau nilai-nilai yang hilang pada datasetnya yaitu 4 data Suhu, 4 data Kelembapan, dan 10 Data Curah_hujan. Dan untuk mengisi data-data yang kosong tersebut kita akan menggunakan nilai median dari masing-masing atributnya.

Menampilkan seluruh isi pada dataset

✓ [10] df

	Tanggal	Suhu	Kelembapan	Curah_Hujan	Kec_Angin
0	01-03-2020	28.5	84.0	8888.0	2
1	02-03-2020	27.9	82.0	6.4	2
2	03-03-2020	28.8	81.0	3.4	3
3	04-03-2020	27.8	84.0	8888.0	2
4	05-03-2020	28.3	81.0	0.7	2
5	06-03-2020	27.8	84.0	3.4	2
6	07-03-2020	26.4	88.0	2.6	2
7	08-03-2020	27.0	84.0	31.1	2
8	09-03-2020	27.9	82.0	8888.0	2
9	10-03-2020	28.9	78.0	7.4	2
10	11-03-2020	28.3	79.0	7.4	3
11	12-03-2020	28.3	80.0	7.4	3
12	13-03-2020	28.4	82.0	7.4	2
13	14-03-2020	27.9	82.0	17.4	2
14	15-03-2020	28.0	78.0	7.4	2
15	16-03-2020	27.9	77.0	7.4	2
16	17-03-2020	27.3	82.0	7.4	2
17	18-03-2020	27.9	78.0	7.4	2
18	19-03-2020	27.8	82.0	7.4	2
19	20-03-2020	27.9	82.0	7.4	2
20	21-03-2020	27.1	84.0	32.5	2
21	22-03-2020	27.9	82.0	6.5	2
22	23-03-2020	26.2	88.0	9.0	2
23	24-03-2020	26.1	88.0	17.7	1
24	25-03-2020	28.0	83.0	0.2	2

Disini dapat dilihat pada datanya terdapat nilai yang tidak beraturan dan terpaut jauh seperti 8888.0 yang mana ini merupakan data rusak atau outliers pada isi datasetnya yang juga akan berpengaruh bilah nantinya data ini akan kita gunakan.

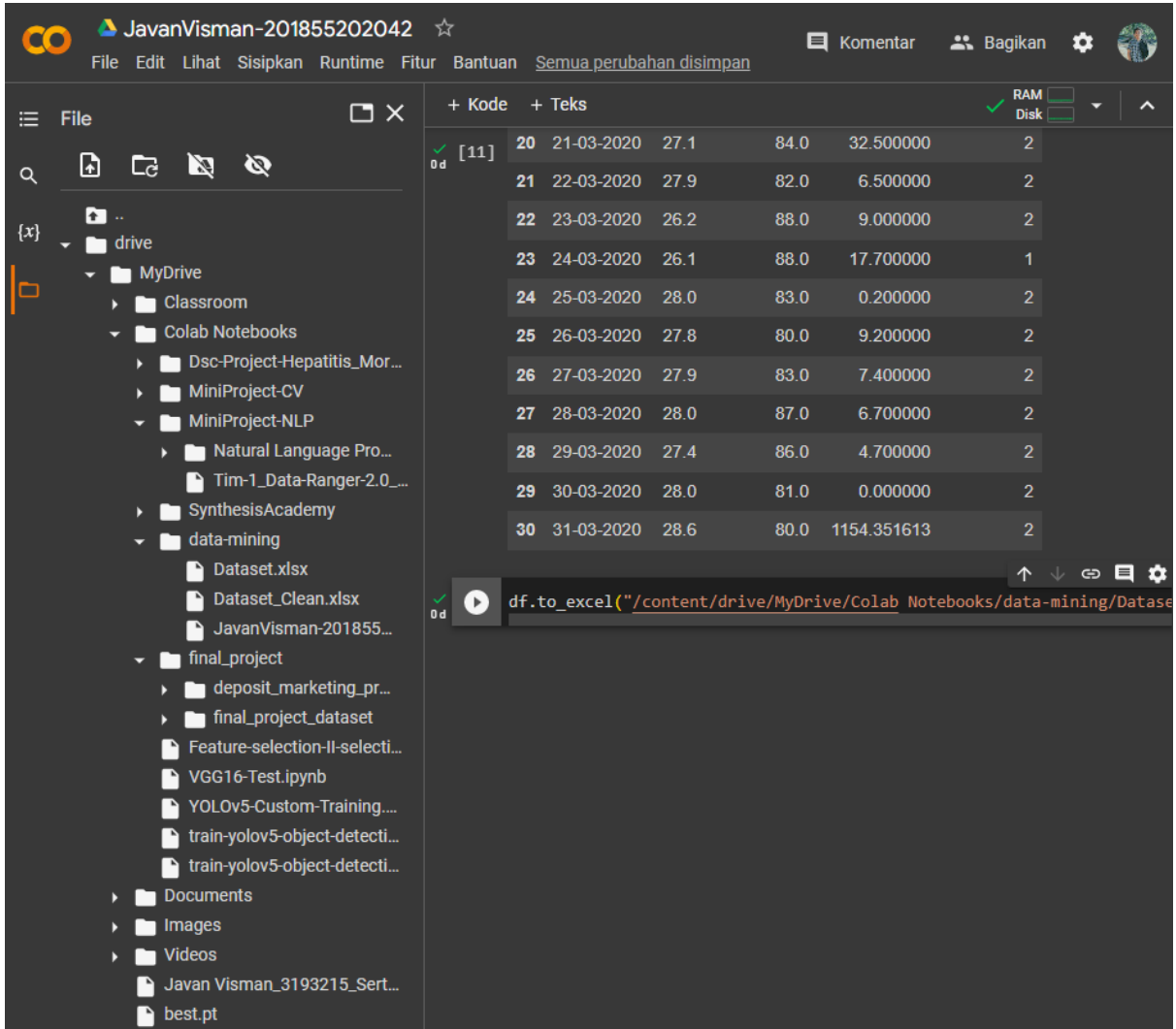
Menangani nilai 8888 yang merupakan outliers

```
[11] mean1 = df["Curah_Hujan"].mean()
df["Curah_Hujan"] = df["Curah_Hujan"].replace([8888.0], mean1)
df
```

	Tanggal	Suhu	Kelembapan	Curah_Hujan	Kec_Angin
0	01-03-2020	28.5	84.0	1154.351613	2
1	02-03-2020	27.9	82.0	6.400000	2
2	03-03-2020	28.8	81.0	3.400000	3
3	04-03-2020	27.8	84.0	1154.351613	2
4	05-03-2020	28.3	81.0	0.700000	2
5	06-03-2020	27.8	84.0	3.400000	2
6	07-03-2020	26.4	88.0	2.600000	2
7	08-03-2020	27.0	84.0	31.100000	2
8	09-03-2020	27.9	82.0	1154.351613	2
9	10-03-2020	28.9	78.0	7.400000	2
10	11-03-2020	28.3	79.0	7.400000	3
11	12-03-2020	28.3	80.0	7.400000	3
12	13-03-2020	28.4	82.0	7.400000	2
13	14-03-2020	27.9	82.0	17.400000	2
14	15-03-2020	28.0	78.0	7.400000	2
15	16-03-2020	27.9	77.0	7.400000	2
16	17-03-2020	27.3	82.0	7.400000	2
17	18-03-2020	27.9	78.0	7.400000	2
18	19-03-2020	27.8	82.0	7.400000	2
19	20-03-2020	27.9	82.0	7.400000	2
20	21-03-2020	27.1	84.0	32.500000	2
21	22-03-2020	27.9	82.0	6.500000	2
22	23-03-2020	26.2	88.0	9.000000	2
23	24-03-2020	26.1	88.0	17.700000	1

Untuk menangani nilai tersebut kita akan menggunakan cara yang hampir sama dengan mengisi missing values sebelumnya, yaitu kita akan menggantikan nilai 8888 tersebut menjadi nilai meannya. Lalu cek kembali dataset yang telah olah tersebut.

Menyimpan Dataset



The screenshot shows the JupyterLab interface for a user named JavanVisman-201855202042. The left sidebar displays a file explorer with a directory structure. The main area shows a code editor with a DataFrame being converted to an Excel file.

File Explorer Structure:

- drive
 - MyDrive
 - Classroom
 - Colab Notebooks
 - Dsc-Project-Hepatitis_Mor...
 - MiniProject-CV
 - MiniProject-NLP
 - Natural Language Pro...
 - Tim-1_Data-Ranger-2.0_...
 - SynthesisAcademy
 - data-mining
 - Dataset.xlsx
 - Dataset_Clean.xlsx
 - JavanVisman-201855...
 - final_project
 - deposit_marketing_pr...
 - final_project_dataset
 - Feature-selection-II-selecti...
 - VGG16-Test.ipynb
 - YOLOv5-Custom-Training...
 - train-yolov5-object-detecti...
 - train-yolov5-object-detecti...
 - Documents
 - Images
 - Videos
 - Javan Visman_3193215_Sert...
 - best.pt

Code Editor Content:

```
df.to_excel("/content/drive/MyDrive/Colab Notebooks/data-mining/Datase")
```

Sampai sini dataset yang sudah diolah tersebut sudah bersih dan lebih baik dari sebelumnya, selanjutnya simpan data tersebut dengan format yang diinginkan.