Laporan Tugas Praktikum2

Syahrul Giga Wahyudi - 0110224085

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

0110224085@student.nurulfikri.ac.id

Abstrac. Pada praktikum ini diperkenalkan lingkungan pengembangan untuk pemodelan *Machine Learning* menggunakan platform **Google Colab**. Google Colab merupakan aplikasi berbasis web yang memungkinkan pengguna menulis serta mengeksekusi kode Python langsung melalui browser tanpa perlu melakukan instalasi atau konfigurasi khusus.

1.1 pratikum

Load data dari Google drive

Gambar 1.1 kode dan output

Penjelasan

Bagian kode ini digunakan untuk **menghubungkan Google Colab dengan akun Google Drive mu**.. Setelah kode ini dijalankan, akan muncul tautan otorisasi. Kamu harus mengklik tautan tersebut, memilih akun Google, dan memberikan izin agar Colab bisa mengakses file-file yang tersimpan di Google Drive-mu. Proses ini hanya perlu dilakukan satu kali per sesi.

1.2 membaca file csv

Gambar 1.2 kode dan output

Penjelasan

Bagian kode ini memanfaatkan library Pandas untuk melakukan proses pembacaan file data. Variabel path digunakan untuk menyimpan alamat direktori pada Google Drive tempat file 500_Person_Gender_Height_Weight_Index.csv disimpan. Selanjutnya, fungsi pd.read_csv() digunakan untuk membaca isi file tersebut dan memuatnya ke dalam sebuah DataFramedf.head() Menampilkan 5 baris pertama dari dataset untuk melihat struktur dan contoh datanya.

1.3 informasi data set

```
#mencari info data pada file (tipe data nya,non nun, count data,nama kolom)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  500 non-null    object
 1   Height  500 non-null    int64
 2   Weight  500 non-null    int64
 3   Index   500 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

Gambar 1.3 kode dan output

Penjelasan

Metode .info() digunakan untuk melihat gambaran umum struktur data. Fungsi ini menampilkan jumlah baris dan kolom, nama kolom, jumlah data yang terisi, serta tipe data pada setiap kolom, seperti int64 untuk angka dan object untuk teks.

1.4 Menghitung Nilai-Nilai Sentral (Mean, Median, Modus):

```
#menghitung mean semuan kolom numerik
df["Height"].mean()
```

```
np.float64(169.944)
```

```
#menghitung median semua kolo numerik
df["Height"].median()
```

```
170.5
```

```
#mencari modus (hati hati karena bisa lebih dari satu)
df["Height"].mode()
```

	Height
0	188

dtype: int64

Gambar 1.4 kode dan output

Penjelasan

Fungsi mean(), median(), dan mode() digunakan untuk mengetahui ukuran pemusatan data. Nilai mean menunjukkan rata-rata tinggi badan dari seluruh data, median menggambarkan posisi nilai tengah setelah data diurutkan, sedangkan mode menunjukkan nilai yang paling sering muncul. Berdasarkan hasil perhitungan, diperoleh nilai rata-rata sebesar 169.94, nilai tengah 170.5, dan nilai yang paling sering muncul adalah 188.

1.5 Variansi & Standar Deviasi

```
#menghitung variasi & standard deviasi
df.var(numeric_only=True)
```

```
                0
Height  268.149162
Weight 1048.633267
Index     1.836168

dtype: float64
```

```
#menghitung standar deviasi
df.std(numeric_only=True)
```

```
                0
Height  16.375261
Weight  32.382607
Index    1.355053

dtype: float64
```

Gambar 1.5 kode dan output

Penjelasan

agian kode ini digunakan untuk menghitung ukuran penyebaran atau variasi data. Fungsi .var() berfungsi untuk mengetahui variansi, yaitu seberapa jauh nilai-nilai data menyimpang dari rata-ratanya. Sementara itu, fungsi .std() digunakan untuk menghitung standar deviasi, yang merupakan akar kuadrat dari variansi dan menunjukkan tingkat sebaran data dengan satuan yang sama seperti data aslinya.

1.6 Menghitung Kuartil

```
#hitung kuartil pertama (q1)
q1 = df["Height"].quantile(0.25)
print ("Q1:",q1)
#hitung kuartil ketiga (q3)
q3 = df["Height"].quantile(0.75)
print ("q3:",q3)
#hitung iqr (interquartile range)
iqr = q3 - q1
print ("iqr:",iqr)
```

```
Q1: 156.0
q3: 184.0
iqr: 28.0
```

Gambar 1.6 kode dan output

Penjelasan

Perintah df['Height'].quantile(0.25) dan df['Height'].quantile(0.75) digunakan untuk menghitung nilai kuartil pertama (Q1) dan kuartil ketiga (Q3) pada data tinggi badan. Nilai Q1 menunjukkan batas bawah data (25% pertama), sedangkan Q3 menunjukkan batas atas data (75% pertama). Selisih antara Q3 dan Q1 disebut IQR (Interquartile Range), yang digunakan untuk melihat rentang penyebaran data di bagian tengah distribusi.

1.7 Menghitung Statistik Deskriptif Otomatis

```
# untuk membuat statistika deskripsi pada type data int
df.describe()
```

	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

Gambar 1.7 kode dan output

Penjelasan

Fungsi .describe() digunakan untuk menghasilkan ringkasan statistik deskriptif secara otomatis dari seluruh kolom numerik dalam dataset. Melalui perintah ini, pengguna dapat melihat berbagai informasi penting seperti jumlah data (count), nilai rata-rata (mean), standar deviasi (std), nilai minimum (min), serta nilai-nilai kuartil (25%, 50%, 75%) dan maksimum (max). Hasil keluaran dari fungsi ini membantu memberikan gambaran umum mengenai distribusi dan karakteristik data tanpa perlu menghitung satu per satu secara manual.

1.8 Menghitung Korelasi

```
[11]
#menghitung matriks korelasi untuk semua kolom numerik
correlation_matrix = df.corr(numeric_only=True)

#menampilkan matriks korelasi
print ("matriks korelasi: ")
print(correlation_matrix)
```

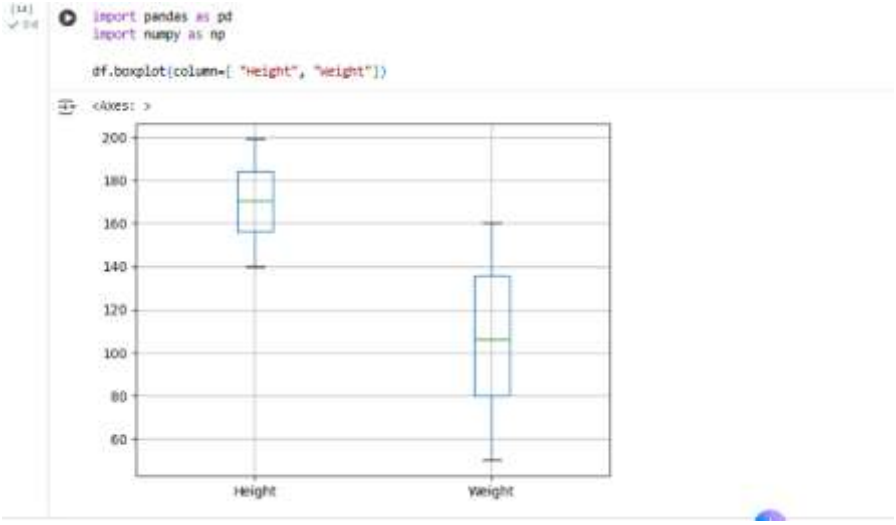
```
matriks korelasi:
          Height    Weight     Index
Height  1.000000  0.000446 -0.422223
Weight  0.000446  1.000000  0.804569
Index  -0.422223  0.804569  1.000000
```

Gambar 1.8 kode dan output

Penjelasan

Kode ini menghitung korelasi Pearson untuk melihat hubungan linear antar variabel numerik di DataFrame. Nilainya berkisar antara -1 sampai 1. Nilai positif menunjukkan hubungan searah, nilai negatif menandakan hubungan berlawanan arah, sedangkan nilai mendekati nol berarti tidak ada hubungan linear yang kuat.

2.1 visualisasi Box plot



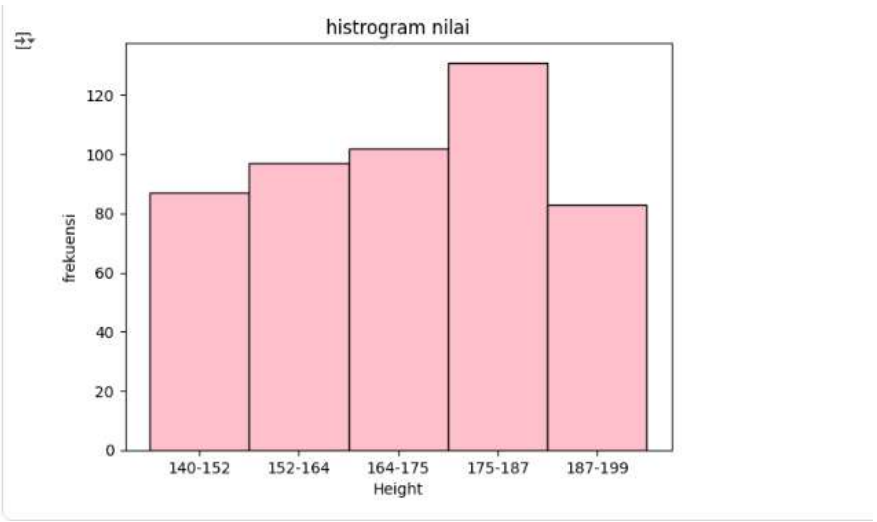
Gambar 2.1 kode dan output

Penjelasan

Kode ini menggunakan fungsi df.boxplot() untuk menampilkan boxplot pada kolom Height dan Weight. Garis tengah pada kotak menunjukkan median, sedangkan ukuran kotak menggambarkan IQR (Interquartile Range). Bagian garis di ujung (whiskers) menunjukkan batas nilai data yang dianggap normal, dan perintah plt.suptitle() digunakan untuk memberi judul pada keseluruhan grafik.

2.2 Histogram

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#ambil data height

data_height = df["Height"]

#buat histrogram
n, bins, patches = plt.hist(data_height, bins=5, color='pink' , edgecolor='black')

#tambah label
plt.title("histrogram nilai")
plt.xlabel("Height")
plt.ylabel("frekuensi")

#tampilkan rentang frekuensi di sumbu x
bin_centers = 0.5 * (bins[1:] + bins[:-1])
plt.xticks(bin_centers,['{:.0f}-{:.0f}'.format(bins[i], bins[i+1]) for i in range(len(bins)-1)])


#tampilkan histrogram
plt.show()
```



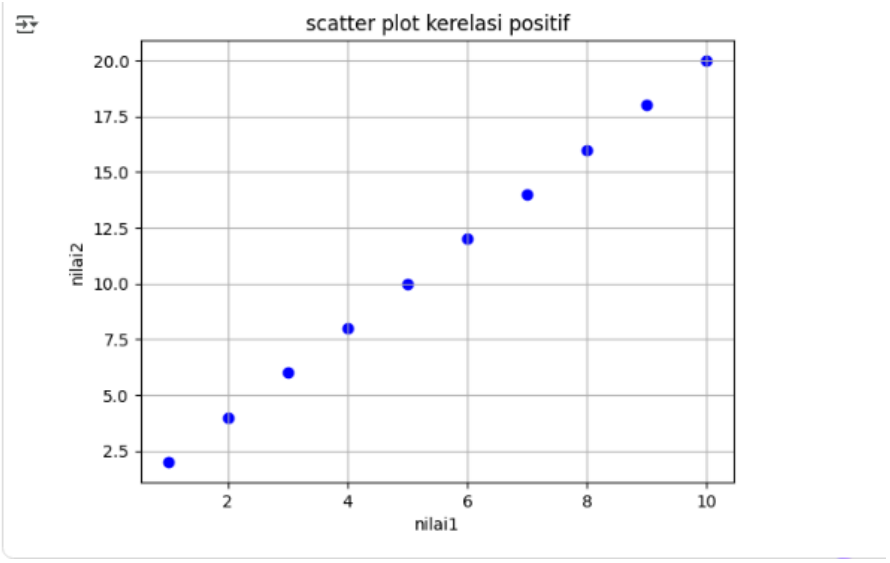
Gambar 2.2 kode dan output

Penjelasan

Kode ini digunakan untuk menampilkan histogram pada kolom Height. Grafik tersebut memperlihatkan distribusi frekuensi dari data tinggi badan. Hasilnya berupa lima batang (bins = 5) yang menunjukkan jumlah data pada tiap rentang nilai tinggi badan.

2.3 Scatter Plot (Korelasi Positif)

```python
import pandas as pd
import matplotlib.pyplot as plt

#buat data fram contoh
data = {
    'nilai1': [1,2,3,4,5,6,7,8,9,10],
    'nilai2': [2,4,6,8,10,12,14,16,18,20]
}
df2 = pd.DataFrame(data)

#buat scatter plot
plt.scatter(df2['nilai1'], df2['nilai2'], color='blue', marker='o')

#tambakah label
plt.title('scatter plot kerelasi positif')
plt.xlabel('nilai1')
plt.ylabel('nilai2')

#tambahkan grid
plt.grid(True)

#tampilkan scatter plot
plt.show()
```



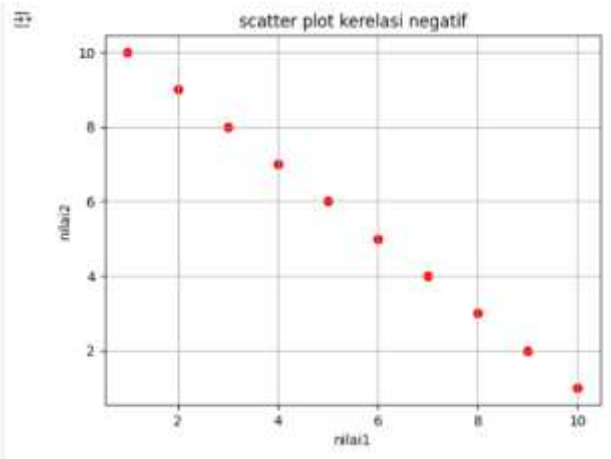
Gambar 2.3 kode dan output

Penjelasan

Kode ini digunakan untuk membuat scatter plot atau diagram pencar guna memperlihatkan hubungan antara dua variabel numerik. Terdapat dua plot yang dihasilkan: plot pertama menunjukkan korelasi positif antara Nilai1 dan Nilai2, sedangkan plot kedua memperlihatkan korelasi negatif di mana peningkatan satu variabel diikuti penurunan variabel lainnya.

2.4 Scatter Plot (Korelasi negative)

```
import pandas as pd
import matplotlib.pyplot as plt

#buat data frame contoh
data = {
    'nilai1': [1,2,3,4,5,6,7,8,9,10],
    'nilai2': [10,9,8,7,6,5,4,3,2,1]
}

df3 = pd.DataFrame(data)

#buat scatter plot
plt.scatter(df3['nilai1'], df3['nilai2'], color='red', marker='o')

#tambahkan label
plt.title('scatter plot kerelasi negatif')
plt.xlabel('nilai1')
plt.ylabel('nilai2')

#tambahkan grid
plt.grid(True)

#tampilkan scatter plot
plt.show()
```



Gambar 2.4 kode dan output

Hasil:

- Korelasi Positif: Titik-titik pada grafik membentuk pola yang naik, menandakan bahwa saat *Nilai1* bertambah, *Nilai2* juga ikut meningkat.
- Korelasi Negatif: Titik-titik membentuk pola menurun, yang berarti peningkatan *Nilai1* diikuti penurunan pada *Nilai2*.

3.1 tugas pratikum mandiri

1. Jalankan seluruh kode program yang terdapat pada modul ini dan kumpulkan hasilnya sebagai laporan praktikum minggu ke-2.
2. Buatlah program untuk **membagi dataset day.csv menjadi tiga bagian**, yaitu:
    - **Data Training (80%)** dari total data,
    - **Data Validation (10%)** dari data training, dan
    - **Data Testing (20%)** dari total dataset.
3. Tampilkan jumlah data pada masing-masing bagian serta lima baris pertama dari setiap set (Training, Validation, dan Testing) sebagai bukti bahwa pembagian data telah berhasil dilakukan.

3.2 Langkah Penyelesaian

1. **Mengimpor Library yang Diperlukan**
   Langkah pertama yaitu mengimpor pustaka yang akan digunakan, seperti pandas untuk membaca dan mengelola dataset, serta scikit-learn khususnya fungsi train_test_split untuk proses pembagian data.
2. **Memuat Dataset day.csv**
   Dataset dibaca menggunakan pandas melalui perintah pd.read_csv(). File day.csv dapat disimpan di folder ../data/day.csv agar mudah diakses dari notebook.
3. **Membagi Data Menjadi Training dan Testing**
   Dataset kemudian dibagi menjadi dua bagian utama, yaitu **Training (80%)** dan **Testing (20%)**, menggunakan fungsi train_test_split. Pembagian ini bertujuan agar sebagian data digunakan untuk melatih model dan sebagian lainnya untuk menguji hasilnya.
4. **Membuat Data Validation dari Data Training**
   Dari data training yang sudah ada, diambil **10%** untuk dijadikan **data validation**.
   Misalnya, jika total data 1000 baris, maka:
    - Data Training awal = 800 baris
    - Data Validation = 80 baris (10% dari Training)
    - Sisa Training akhir = 720 baris
    - Data Testing = 200 baris
5. **Menampilkan Jumlah Data dan Sampel Awal**
   Setelah pembagian selesai, tampilkan jumlah data dari setiap set (Training, Validation, dan Testing), beserta **lima baris pertama** dari masing-masing set sebagai bukti hasil pembagian dataset.

```
[27]
✓ 0d
```

```python
import pandas as pd
from sklearn.model_selection import train_test_split

df4 = pd.read_csv("/content/drive/MyDrive/Pratikum_Ml/pratikum02/data/day.csv")
print("Jumlah total data: ", len(df4))

train,test = train_test_split(df4, test_size=0.2, random_state=42)

train,val = train_test_split(train, test_size=0.1, random_state=42)

print("\nJumlah data train      : ",len(train))
print("Jumlah data validation : ",len(val))
print("Jumlah data test       : ",len(test))

print("\nData train:")
print(train.head())

print("\nData validation:")
print(val.head())

print("\nData test:")
print(test.head())
```

Gambar 3.2 kode tugas mandiri

Hasil dari data train

```
Jumlah total data:  731

Jumlah data train      :  525
Jumlah data validation :  59
Jumlah data test       :  147

Data train:
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
657      658  2012-10-19       4   1    10        0        5           1
163      164  2011-06-13       2   0     6        0        1           1
305      306  2011-11-02       4   0    11        0        3           1
111      112  2011-04-22       2   0     4        0        5           1
538      539  2012-06-22       3   1     6        0        5           1

     weathersit      temp     atemp       hum  windspeed  casual  registered  \
657           2  0.563333  0.537896  0.815000   0.134954     753        4671
163           1  0.635000  0.601654  0.494583   0.305350     863        4157
305           1  0.377500  0.390133  0.718750   0.082092     370        3816
111           2  0.336667  0.321954  0.729583   0.219521     177        1506
538           1  0.777500  0.724121  0.573750   0.182842     964        4859

      cnt
657  5424
163  5020
305  4186
111  1683
538  5823
```

Gambar 3.2 output tugas mandiri

Hasil dari data validation

```
Data validation:
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
325      326  2011-11-22       4   0    11        0        2           1
410      411  2012-02-15       1   1     2        0        3           1
92        93  2011-04-03       2   0     4        0        0           0
47        48  2011-02-17       1   0     2        0        4           1
508      509  2012-05-23       2   1     5        0        3           1

     weathersit      temp     atemp       hum  windspeed  casual  registered  \
325           3  0.416667  0.421696  0.962500   0.118792      69        1538
410           1  0.348333  0.351629  0.531250   0.181600     141        4028
92            1  0.378333  0.378767  0.480000   0.182213    1651        1598
47            1  0.435833  0.428658  0.505000   0.230104     259        2216
508           2  0.621667  0.584612  0.774583   0.102000     766        4494

      cnt
325  1607
410  4169
92   3249
47   2475
508  5260
```

Gambar 3.2 output tugas mandiri

Hasil dari Data Test

```
--- ----

Data test:
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
703      704  2012-12-04       4   1    12        0        2           1
33        34  2011-02-03       1   0     2        0        4           1
300      301  2011-10-28       4   0    10        0        5           1
456      457  2012-04-01       2   1     4        0        0           0
633      634  2012-09-25       4   1     9        0        2           1

     weathersit      temp     atemp       hum  windspeed  casual  registered  \
703           1  0.475833  0.469054  0.733750   0.174129     551        6055
33            1  0.186957  0.177878  0.437826   0.277752      61        1489
300           2  0.330833  0.318812  0.585833   0.229479     456        3291
456           2  0.425833  0.417287  0.676250   0.172267    2347        3694
633           1  0.550000  0.544179  0.570000   0.236321     845        6693

      cnt
703  6606
33   1550
300  3747
456  6041
633  7538
```

Gambar 3.2 output tugas mandiri

Kesimpulan

Berdasarkan hasil praktikum ini, dapat disimpulkan bahwa:

1. **Google Colab** merupakan platform yang efektif untuk menjalankan eksperimen *Machine Learning* karena mudah digunakan, tidak memerlukan instalasi tambahan, dan mendukung akses GPU gratis.
2. Proses **analisis statistik deskriptif** seperti perhitungan mean, median, modus, variansi, dan standar deviasi membantu memahami karakteristik data sebelum dilakukan pemodelan.

3. Visualisasi seperti **boxplot, histogram, dan scatter plot** sangat membantu dalam mengidentifikasi pola serta hubungan antar variabel.
4. Pembagian dataset menjadi **Training, Validation, dan Testing** penting dilakukan agar model *Machine Learning* dapat dilatih, dievaluasi, dan diuji secara seimbang.
5. Dengan langkah-langkah ini, mahasiswa dapat memahami dasar pengolahan data sebelum melangkah ke tahap pemodelan dan evaluasi performa model *Machine Learning* secara lebih mendalam.

Link github https://github.com/SyahrulGigaWahyudi/Machine-Learning-pagi/tree/714fa02313a073da23679cd1627eddca44dbec22/pratikum02

Link google colab:
https://drive.google.com/drive/folders/1UiBb5Vyb62N5SjHypGODfQH0ydTTiAqO?usp=sharing