

▼ Nama : Rangga Pebrianto

NIM : G6601231006

Dosen Pengampu : Dr. Toto Haryanto S.Kom., M.Si.

Mata Kuliah Praktikum Pengenalan Pola

Tujuan Praktikum Mahasiswa mampu menterjemahkan representasi data dalam Python dan manipulasinya sehingga bisa menjadi input bagi sistem pengenalan Pola

Review – Representasi Data dalam Python

1. Vektor Deklarasi variabel diawali dengan Vektor dapat dituliskan dengan dan tanpa bantuan library

```
# Import pustaka NumPy dengan alias np
import numpy as np

# Mencetak judul untuk vektor default Python
print("vektor default python\n")

# Membuat vektor a dengan nilai dari 1 hingga 19 dengan langkah 1
a = np.arange(1, 20, 1)

# Membuat vektor b dengan nilai dari 1 hingga 19 dengan langkah 2
b = np.arange(1, 20, 2)

# Mengimpor pustaka NumPy lagi (tidak diperlukan, karena sudah diimpor di atas)
import numpy as np

# Mencetak judul untuk vektor melalui NumPy
print(" \n vektor via numpy \n")

# Membuat vektor c dengan menggunakan np.array() dari daftar nilai
c = np.array([1, 2, 3, 4, 5])

# Membuat vektor d dengan menggunakan np.array() dari daftar nilai desimal
d = np.array([1.5, 2.5, 5, 6, 7])

# Mencetak vektor a
print(a)

# Mencetak vektor b
print(b)

# Mencetak jumlah dimensi dari vektor a
print(a.ndim)

# Mencetak bentuk (shape) dari vektor a
print(a.shape)
```

vektor default python

vektor via numpy

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
[ 1  3  5  7  9 11 13 15 17 19]
1
(19,)
```

Kodingan diatas penggunaan pustaka NumPy dalam bahasa pemrograman Python untuk bekerja dengan vektor. NumPy adalah pustaka yang populer digunakan untuk komputasi numerik dan ilmiah, terutama untuk bekerja dengan array multidimensi. Hasil keluaran dari kodingan ini akan terdiri dari dua bagian. Bagian pertama akan mencetak vektor a dan vektor b yang dihasilkan menggunakan fungsi `np.arange()`. Bagian kedua akan mencetak vektor c dan vektor d yang dihasilkan dengan fungsi `np.array()`.

Perlu diperhatikan bahwa mengimpor pustaka NumPy dua kali seperti dalam contoh di atas tidak diperlukan. Cukup melakukan impor sekali di awal kodingan.

2. Matrix Matriks adalah basic 2D table dari data dan dapat berisi nilai numerik dan/atau karakter. Dapat dibuat dengan cara sederhana dengan membuat urutan dari vektor, mengubah dari vektor atau membaca dari file

```
# Import pustaka NumPy dengan alias np (sebagai asumsi, pustaka ini sudah diimpor sebelumnya)
import numpy as np

# Membuat vektor a dengan nilai dari 1 hingga 20 dengan langkah 1
a = np.arange(1, 21, 1)

# Mengubah vektor a menjadi matriks 2D dengan bentuk (4, 5)
c = a.reshape((4, 5))

# Mencetak matriks c
print(c)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]]
```

3. List merupakan merupakan representasi struktur data yang dapat menyimpan data dengan nilai numerik, karakter, dan lain-lain secara bersamaan

```
# Membuat beberapa contoh list
list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
list4 = ["abc", 34, True, 40, "male"]

# Mencetak isi dari list1
print(list1)

['apple', 'banana', 'cherry']
```

Penjelasan baris per baris:

1. list1 didefinisikan sebagai sebuah list yang berisi tiga elemen string: "apple", "banana", dan "cherry".
2. list2 didefinisikan sebagai sebuah list yang berisi lima elemen angka: 1, 5, 7, 9, dan 3.
3. list3 didefinisikan sebagai sebuah list yang berisi tiga elemen boolean: True, False, dan False.
4. list4 didefinisikan sebagai sebuah list yang berisi lima elemen dengan tipe data yang berbeda-beda: "abc" (string), 34 (angka), True (boolean), 40 (angka), dan "male" (string).
5. print(list1) digunakan untuk mencetak isi dari list1.

Hasil keluaran dari kodingan ini akan mencetak isi dari list1, yaitu:

```
['apple', 'banana', 'cherry']
```

Ini adalah tampilan list yang berisi tiga elemen string yang sebelumnya telah didefinisikan dalam list1.

4. data.Frame Data frame adalah spesialisasi dari tipe list untuk menyimpan vektor ke dalam bentuk frame (menyerupai basis data). Kelebihan dibandingkan matriks adalah dapat dimanipulasi dalam berbagai bentuk dan cara.

```
# Mengimpor pustaka pandas dengan alias pd
import pandas as pd

# Membuat sebuah DataFrame menggunakan pustaka pandas
df = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]),
                  columns=['a', 'b', 'c'])

# Mencetak DataFrame yang telah dibuat
print(df)
```

```
   a  b  c
0  1  2  3
1  4  5  6
2  7  8  9
```

Penjelasan baris per baris:

1. Mengimpor pustaka pandas dengan alias pd. Pandas adalah pustaka yang digunakan untuk analisis data yang menyediakan struktur data dan alat untuk manipulasi data yang efisien.

2. Membuat sebuah DataFrame dengan menggunakan fungsi `pd.DataFrame()`. DataFrame adalah struktur data 2D yang mirip dengan tabel, terdiri dari baris dan kolom. Dalam kasus ini, kita membuat DataFrame dengan tiga baris dan tiga kolom, diisi dengan nilai-nilai dari array 2D.
3. `np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])` adalah array 2D (matriks) yang berisi nilai-nilai untuk mengisi DataFrame. Matriks ini terdiri dari tiga baris dan tiga kolom.
4. `columns=['a', 'b', 'c']` digunakan untuk memberi nama kolom pada DataFrame yang telah dibuat. Kolom pertama akan dinamai 'a', kolom kedua 'b', dan kolom ketiga 'c'.
5. `print(df)` digunakan untuk mencetak DataFrame yang telah dibuat.

Hasil keluaran dari kodingan ini akan mencetak DataFrame sebagai berikut:

```

   a  b  c
0  1  2  3
1  4  5  6
2  7  8  9

```

Ini adalah tampilan DataFrame yang terdiri dari tiga baris dan tiga kolom, sesuai dengan nilai-nilai yang telah diisi dalam array 2D pada awalnya. Kolom 'a', 'b', dan 'c' mewakili nama kolom yang telah ditentukan.

▼ Import dan Export data serta manipulasinya

Menghubungkan Google drive yang berisi file yang akan dipakai dengan google colab

```

from google.colab import drive
drive.mount('/content/drive')

```

Mounted at /content/drive

```

# Membaca data dari file dengan format CSV
import pandas as pd
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Data.csv", sep=";")
print(data)

```

```

# Membaca data dari file dengan format text (delimiter)
print("\n read text data with tab delimiter")
with open ('/content/drive/MyDrive/Colab Notebooks/Data.txt') as data:
    print(data.read())

```

```

# Membaca data dari URL
import pandas as pd
f = pd.read_csv('http://www.exploredata.net/ftp/Spellman.csv')
print(f)

```

	Nama Pasien	Umur	Gender	Diagnosa	Sakit
0	Anto	24	L		Tidak
1	Budi	35	L		Ya
2	Adi	55	L		Ya
3	Delima	32	P		Ya
4	Dodi	21	L		Tidak
5	Tukiyem	19	P		Tidak
6	Rama	23	L		Tidak
7	Santi	35	P		Tidak
8	Mery	44	P		Ya
9	Yanti	27	P		Tidak
10	Parto	43	L		Ya
11	Dea	24	P		Tidak

```

read text data with tab delimiter
NamaPasien    Umur    Gender    Diagnosa    Sakit
Anto           24      L         Tidak
Budi           35      L         Ya
Adi            55      L         Ya
Delima         32      P         Ya
Dodi           21      L         Tidak
Tukiyem        19      P         Tidak
Rama           23      L         Tidak
Santi          35      P         Tidak
Mery           44      P         Ya
Yanti          27      P         Tidak
Parto          43      L         Ya
Dea            24      P         Tidak
time          40      50      60      70      80      90      100     110     120     ...  \
0  YAL001C  -0.070  -0.23  -0.100  0.03  -0.04  -0.12  -0.28  -0.44  -0.09  ...
1  YAL014C  0.215  0.09   0.025  -0.04  -0.04  -0.02  -0.51  -0.08  0.00  ...
2  YAL016W  0.150  0.15   0.220  0.29  -0.10  0.15  -0.73  0.19  -0.15  ...

```

```

3    YAL020C -0.350 -0.28 -0.215 -0.15 0.16 -0.12 0.26 0.00 0.13 ...
4    YAL022C -0.415 -0.59 -0.580 -0.57 -0.09 -0.34 0.49 0.32 1.15 ...
...
4376 YPR198W -0.060 0.08 0.210 0.34 0.65 -0.26 0.14 -0.33 0.53 ...
4377 YPR199C 0.155 0.19 0.235 0.28 -0.26 0.21 -0.40 0.34 -0.80 ...
4378 YPR201W -0.255 -0.36 -0.300 -0.24 1.30 -0.07 0.29 -0.20 0.25 ...
4379 YPR203W 0.570 0.12 -0.070 -0.26 -0.44 -0.21 -1.08 0.39 -0.17 ...
4380 YPR204W 0.405 0.17 -0.045 -0.26 -0.60 -0.09 -0.85 0.17 -0.05 ...

170 180 190 200 210 220 230 240 250 260
0    0.59 0.34 -0.28 -0.09 -0.44 0.31 0.03 0.57 0.00 0.010
1    -0.30 -0.38 0.07 -0.04 0.13 -0.06 -0.26 -0.10 0.27 0.235
2    0.12 -0.17 0.11 -0.15 0.03 -0.26 -0.34 -0.34 0.25 0.190
3    0.07 0.61 -0.20 0.49 -0.43 0.80 -0.47 1.01 -0.36 -0.405
4    -0.48 -0.40 -0.59 0.54 -0.09 1.03 0.08 0.57 -0.26 -0.310
...
4376 0.14 -0.64 -0.26 0.53 -0.17 0.59 -0.96 0.40 -0.23 -0.325
4377 0.34 0.15 0.30 -0.06 0.13 -0.44 -1.03 0.14 0.30 0.250
4378 -0.81 0.89 0.07 1.04 -0.32 0.80 -0.13 0.84 -0.39 -0.415
4379 0.12 -0.96 -0.31 -0.81 -0.34 -1.21 -1.36 -0.12 0.69 0.555
4380 0.17 -1.90 -0.21 -0.45 -0.31 -0.39 -0.22 -0.08 0.65 0.520

```

[4381 rows x 24 columns]

Penjelasan kodingan diatas

1. Membaca Data dari File CSV menggunakan Pandas:

```

import pandas as pd
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Data.csv", sep=";")
print(data)

```

- `import pandas as pd`: Ini mengimpor pustaka Pandas dan memberi Anda akses ke fungsionalitasnya.
- `data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Data.csv", sep=";")`: Ini membaca data dari file CSV yang terletak di jalur yang ditentukan. Parameter `sep=";"` mengindikasikan bahwa pemisah (delimiter) dalam file CSV adalah titik koma (;).
- `print(data)`: Ini mencetak data yang telah dibaca dari file CSV ke konsol.

2. Membaca Data dari File Teks dengan Delimiter:

```

print("\n read text data with tab delimiter")
with open('/content/drive/MyDrive/Colab Notebooks/Data.txt') as data:
    print(data.read())

```

- `with open('/content/drive/MyDrive/Colab Notebooks/Data.txt') as data:`: Ini membuka file teks yang terletak di jalur yang ditentukan dan memberikan alias `data` untuk file yang dibuka.
- `print(data.read())`: Ini membaca dan mencetak isi file teks yang dibuka ke konsol. File ini akan dibaca sebagai teks utuh tanpa memisahkan baris atau menguraikannya ke dalam struktur data tertentu.

3. Membaca Data dari URL menggunakan Pandas:

```

import pandas as pd
f = pd.read_csv('http://www.exploredata.net/ftp/Spellman.csv')
print(f)

```

- `import pandas as pd`: Seperti sebelumnya, ini mengimpor pustaka Pandas.
- `f = pd.read_csv('http://www.exploredata.net/ftp/Spellman.csv')`: Ini membaca data dari URL yang diberikan. URL ini mengarah ke file CSV di internet.
- `print(f)`: Ini mencetak data yang telah dibaca dari URL ke konsol.

Kodingan tersebut mengilustrasikan cara membaca data dari berbagai sumber: dari file CSV di Google Drive, dari file teks dengan delimiter, dan dari URL eksternal menggunakan pustaka Pandas. Setelah data dibaca, Pandas memungkinkan Anda untuk melakukan berbagai operasi analisis dan manipulasi pada data tersebut.

Membaca file dan menyajikan dalam bentuk grafik. Untuk dapat menyajikan dalam bentuk grafik terlebih dahulu melakukan instalasi paket matplotlib.

```

import numpy as np
traffic = np.genfromtxt("/content/drive/MyDrive/Colab Notebooks/web_traffic.tsv", delimiter='\t')
print(traffic[:10])
print(traffic.shape)

x = traffic[:,0]
y = traffic[:,1]

x = x[~np.isnan(y)]
y = y[~np.isnan(y)]

import matplotlib.pyplot as plt
plt.scatter(x,y)
plt.title("Web traffic last month")
plt.xlabel("Time")
plt.ylabel("Hits/hour")

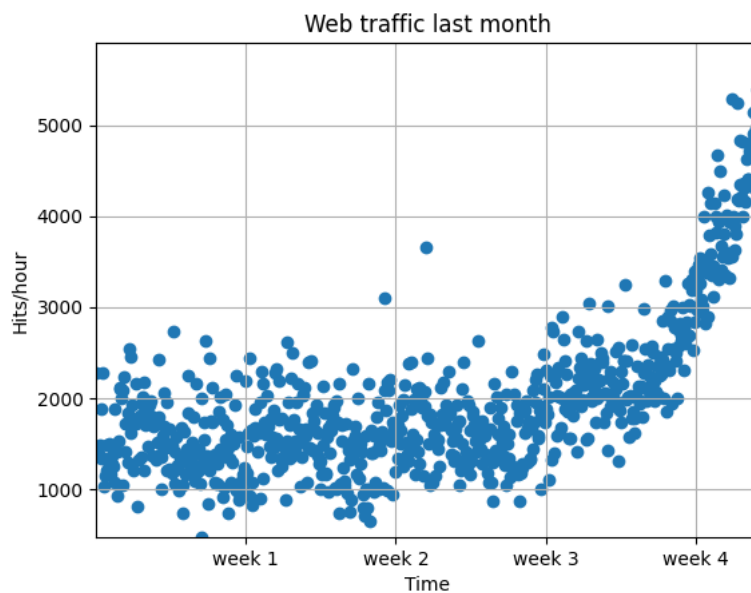
plt.xticks([w*7*24 for w in range(10)], ['week %i' %w for w in range(10)])
plt.autoscale(tight=True)
plt.grid()

```

```

[[1.000e+00 2.272e+03]
 [2.000e+00 nan]
 [3.000e+00 1.386e+03]
 [4.000e+00 1.365e+03]
 [5.000e+00 1.488e+03]
 [6.000e+00 1.337e+03]
 [7.000e+00 1.883e+03]
 [8.000e+00 2.283e+03]
 [9.000e+00 1.335e+03]
 [1.000e+01 1.025e+03]]
(743, 2)

```



Penjelasan dari kodingan diatas

```

import numpy as np
traffic = np.genfromtxt("web_traffic.tsv", delimiter='\t')
print(traffic[:10])
print(traffic.shape)

x = traffic[:, 0]
y = traffic[:, 1]

x = x[~np.isnan(y)]
y = y[~np.isnan(y)]

import matplotlib.pyplot as plt
plt.scatter(x, y)
plt.title("Web traffic last month")
plt.xlabel("Time")
plt.ylabel("Hits/hour")

plt.xticks([w * 7 * 24 for w in range(10)], ['week %i' % w for w in range(10)])

```

```
plt.autoscale(tight=True)
plt.grid()
```

1. Mengimpor Numpy:

```
import numpy as np
```

Ini mengimpor pustaka NumPy dan memberi Anda akses ke berbagai fungsi untuk pemrosesan array dan komputasi numerik.

2. Membaca Data dari File TSV (Tab-Separated Values):

```
traffic = np.genfromtxt("web_traffic.tsv", delimiter='\t')
print(traffic[:10])
print(traffic.shape)
```

- `traffic = np.genfromtxt("web_traffic.tsv", delimiter='\t')`: Ini membaca data dari file TSV yang disebut "web_traffic.tsv" dengan menggunakan fungsi `genfromtxt` dari NumPy. Parameter `delimiter='\t'` mengindikasikan bahwa file tersebut memiliki delimiter berupa tab.
- `print(traffic[:10])`: Ini mencetak 10 baris pertama dari data yang telah dibaca ke konsol.
- `print(traffic.shape)`: Ini mencetak dimensi (shape) dari array data yang telah dibaca ke konsol.

3. Memisahkan Data x dan y:

```
x = traffic[:, 0]
y = traffic[:, 1]

x = x[~np.isnan(y)]
y = y[~np.isnan(y)]
```

- `x = traffic[:, 0]` dan `y = traffic[:, 1]`: Ini memisahkan kolom pertama (x) dan kolom kedua (y) dari data ke dalam array terpisah.
- `x = x[~np.isnan(y)]` dan `y = y[~np.isnan(y)]`: Ini menghilangkan nilai NaN (Not a Number) dari array y dan x, karena NaN bisa mengganggu visualisasi dan analisis data.

4. Menggunakan Matplotlib untuk Visualisasi Data:

```
import matplotlib.pyplot as plt
plt.scatter(x, y)
plt.title("Web traffic last month")
plt.xlabel("Time")
plt.ylabel("Hits/hour")

plt.xticks([w * 7 * 24 for w in range(10)], ['week %i' % w for w in range(10)])
plt.autoscale(tight=True)
plt.grid()
```

- Ini menggunakan Matplotlib untuk membuat scatter plot dari data x dan y.
- Fungsi-fungsi seperti `plt.title`, `plt.xlabel`, dan `plt.ylabel` digunakan untuk memberikan judul dan label sumbu pada plot.
- `plt.xticks` digunakan untuk mengatur tanda sumbu x, dan `plt.autoscale` untuk menyesuaikan skala plot secara otomatis.
- `plt.grid` menambahkan grid pada plot untuk membantu visualisasi.

Kodingan tersebut mengilustrasikan bagaimana menggunakan NumPy dan Matplotlib untuk membaca data dari file TSV, membersihkan data, dan membuat scatter plot yang menggambarkan hubungan antara waktu (x) dan jumlah hits per jam (y) dari lalu lintas web.

```
# Operasi image pada Python
# Instalasi paket open cv
!pip install opencv-python

import matplotlib.pyplot as plt
import cv2

print("read images using opencv")
logo = cv2.imread("/content/drive/MyDrive/Colab Notebooks/ipb.png")
print(logo.shape)
print(logo.size)
```

```
five = cv2.imread("/content/drive/MyDrive/Colab Notebooks/5.png")
print(five.shape)
print(five.size)
```

```
# Menampilkan dua gambar dalam subplot
plt.figure(figsize=(10, 5))
```

```
# Gambar pertama
plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(logo, cv2.COLOR_BGR2RGB))
plt.title("Logo IPB")
plt.axis('off')
```

```
# Gambar kedua
plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(five, cv2.COLOR_BGR2RGB))
plt.title("Number 5")
plt.axis('off')
```

```
plt.tight_layout()
plt.show()
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-pac
read images using opencv
(1200, 1600, 3)
5760000
(402, 410, 3)
494460
```

Number 5

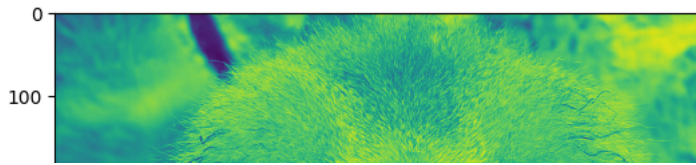
Logo IPB



```
# konversi image
import cv2
babon = cv2.imread("/content/drive/MyDrive/Colab Notebooks/babon.jpg")
babon_gray = cv2.cvtColor(babon, cv2.COLOR_BGR2GRAY)

plt.imshow(babon)
plt.imshow(babon_gray)
```

<matplotlib.image.AxesImage at 0x7d72a9fd3700>



penjelasan kodingan diatas berfokus pada konversi citra dan penggunaan Matplotlib untuk menampilkan gambar. Saya akan menjelaskan setiap bagian dari kodingan tersebut:

```
# konversi image
import cv2
babon = cv2.imread("/content/drive/MyDrive/Colab Notebooks/babon.jpg")
babon_gray = cv2.cvtColor(babon, cv2.COLOR_BGR2GRAY)

plt.imshow(babon)
plt.imshow(babon_gray)
```

1. `import cv2`: Ini mengimpor pustaka OpenCV yang digunakan untuk operasi gambar.
2. `babon = cv2.imread("/content/drive/MyDrive/Colab Notebooks/babon.jpg")`: Ini membaca gambar "babon.jpg" dari path yang diberikan dan menyimpannya dalam variabel `babon`.
3. `babon_gray = cv2.cvtColor(babon, cv2.COLOR_BGR2GRAY)`: Ini mengubah gambar berwarna (`babon`) menjadi citra grayscale (`babon_gray`) menggunakan fungsi `cv2.cvtColor()`. Konversi dari BGR (format warna default OpenCV) ke grayscale dilakukan dengan menggunakan `cv2.COLOR_BGR2GRAY`.
4. `plt.imshow(babon)`: Ini menggunakan Matplotlib untuk menampilkan gambar berwarna `babon`.
5. `plt.imshow(babon_gray)`: Ini menggunakan Matplotlib untuk menampilkan gambar grayscale `babon_gray`.

Namun, perlu diingat bahwa Anda hanya melihat hasil dari operasi ini di dalam notebook atau lingkungan pengembangan. Untuk menampilkan gambar secara terpisah, Anda perlu menambahkan perintah `plt.show()` setelah setiap pemanggilan `plt.imshow()`.

Jika Anda ingin menampilkan kedua gambar dalam satu tampilan subplot, Anda dapat menggunakan perbaikan yang sama seperti yang telah saya jelaskan sebelumnya.

```
# mengambil nilai matriksnya

# acces pixel of images per postion
pixels = five[100,100]
print(pixels)
```

[255 255 255]

Kode yang Anda berikan adalah contoh pengambilan nilai piksel dari suatu citra (gambar). Di bawah ini adalah penjelasan langkah demi langkah dari kode tersebut:

1. `pixels = five[100, 100]`: Pada baris ini, kita mengakses nilai piksel dari suatu citra yang disimpan dalam variabel `five`. `five` seharusnya adalah representasi matriks/citra yang ingin Anda olah. `[100, 100]` adalah posisi piksel yang ingin diakses. Angka pertama (100) mengacu pada koordinat baris (tinggi), dan angka kedua (100) mengacu pada koordinat kolom (lebar) dalam matriks citra.
2. `print(pixels)`: Ini akan mencetak nilai piksel yang telah diakses pada koordinat (100, 100) ke konsol.

Jadi, potongan kode tersebut sebenarnya hanya mengambil nilai piksel pada posisi (100, 100) dari citra yang disimpan dalam variabel `five` dan mencetak nilainya ke layar.

Double-click (or enter) to edit

✓ 0s completed at 5:06 PM

● ✕