



MODUL 9

“Matriks”

A. Tujuan

1. Memahami definisi matriks.
2. Mampu membuat sebuah matriks dari list.
3. Mampu membuat matriks menggunakan perulangan.
4. Memahami operasi dasar pada matriks menggunakan python.

B. Pendahuluan

Matriks adalah suatu bilangan atau simbol yang disusun dalam **baris dan kolom**. Isi dari matriks disebut sebagai elemen-elemen matriks. Penulisan elemen matriks tersebut berada dalam kurung (...) atau [...]. Elemen horizontal disebut sebagai baris (biasa disimbolkan dengan m), kemudian elemen vertikal disebut sebagai kolom (biasa disimbolkan sebagai n). Matriks dalam hal ini sebetulnya merupakan penerapan dari tipe data *collection* yang telah di pelajari di modul sebelumnya dimana dasar dari matriks ini adalah sebuah **array** yang berformat list maupun tuple.

C. Membuat Matrik dari List

Pada bahasa pemrograman python, matriks digambarkan sebagai list bersarang atau **nested list** atau list dua dimensi. Berikut ini ilustrasi pendefinisi-an sebuah matriks.

```
matrks.py > ...
1 # matriks berukuran 2 x 2
2 matrksA = [ [1, 2],
3             [3, 4]
4             ]
5 print(matrksA)
6 print(matrksA[0][0]) #contoh pengakses-an elemen matriks
7 print(matrksA[1][1])
8
9 # matriks berukuran 3 x 2
10 matrksB = [ [1, 2],
11             [3, 4],
12             [5, 6]
13             ]
14 matrksB[1][1] = 40 #contoh update elemen matriks
15 print(matrksB)
16 print(matrksB[2][0])
17
```

Gambar 9.1 Contoh matriks dan cara mengakses

D. Membuat Matriks dari Perulangan

Membuat matriks dapat dilakukan menggunakan blok program perulangan menggunakan for. Seperti contoh di bawah ini.



```
matrks.py > ...
18 #MATRIKS DENGAN PERULANGAN
19 m = 4
20 n = 5
21 x = [0]*m
22 for i in range(m):
23     x[i] = [1]*n
24 print(x)
25
26
27 #OUTPUT
28 [[1, 1, 1, 1, 1],
29  [1, 1, 1, 1, 1],
30  [1, 1, 1, 1, 1],
31  [1, 1, 1, 1, 1]]
```

Gambar 9.2 Contoh matriks dengan perulangan

Pada program di atas kita membuat sebuah matriks dengan ukuran 4×5 dengan menggunakan dua variabel, diantaranya:

- Variabel m yang akan digunakan untuk mendefinisikan **baris**
- Variabel n yang akan digunakan untuk mendefinisikan **kolom**

Selanjutnya membuat sebuah variabel baru (x) untuk proses *parsing* sebanyak m baris. Angka '0' dipilih untuk default definisi awal saja, karena pada akhirnya nya akan di update oleh hasil perulangan nya, jadi sebenarnya '0' atau berapapun sah-sah saja (namun tidak boleh list kosng).

Kemudian pada *syntax* `for i in range(m):`, maknanya indeks i akan berulang sebanyak m kali. Kemudian `x[i]` dimaksudkan untuk memerintahkan mengupdate list awal (`x=[0]`, *line 21*) menjadi `x=[1]`. Maka akan menghasilkan hasil seperti output diatas.

E. Operasi Dasar Matriks

- **Penjumlahan Matriks**

```
matrks.py > ...
33 #PENJUMLAHAN MATRIKS
34 mat1 = [
35     [5, 0],
36     [2, 6],
37 ]
38
39 mat2 = [
40     [1, 0],
41     [4, 2],
42 ]
43 for x in range(len(mat1)):
44     for y in range(len(mat1[0])):
45         print (mat1[x][y] + mat2[x][y], end=' '),
46     print
47 #OUTPUT
48 [6, 0],
49 [6, 8]
```

Gambar 9.3 Contoh **BENAR** penjumlahan matriks.



```
matrks.py > ...
49 mat1 = [
50     [5, 0],
51     [2, 6],
52 ]
53
54 mat2 = [
55     [1, 0],
56     [4, 2],
57 ]
58 print(mat1+mat2)
59
60 #OUTPUT
61 [[5, 0], [2, 6], [1, 0], [4, 2]]
```

Gambar 9.4 Contoh **SALAH** penjumlahan matriks.

Penjelasan :

Secara matematis dalam penjumlahan matriks, objek yang dioperasikan adalah **jumlah dari masing-masing elemen kedua matriks**. Dalam mekanisme python, matriks di wujudkan dalam struktur data **list**, yang mana untuk mengakses elemen nya harus meng-indeks dari tiap-tiap elemen itu sendiri. Dalam operasi aritmatika python, yang berhasil dioperasikan adalah data yang telah bertipe integer.

Perulangan adalah solusi untuk mengakses tiap elemen nya, karena perulangan akan **mem - parsing** sebanyak baris matriks. Supaya semakin memahami, perhatikan contoh kedua. *Output* kedua menunjukkan apabila matriks langsung dijumlahkan secara variabel maka akan mengikuti kaidah **operasi list**. Alih-alih menghasilkan penjumlahan matriks, melainkan menghasilkan **penggabungan list**.

- **Pengurangan Matriks**

Tidak berbeda jauh dengan penjumlahan matriks, pada pengurangan matriks kita hanya mengganti operatornya saja dengan tanda kurang (-).

```
matrks.py > ...
33 #PENGURANGAN MATRIKS
34 mat1 = [
35     [5, 0],
36     [2, 6],
37 ]
38
39 mat2 = [
40     [1, 0],
41     [4, 2],
42 ]
43 for x in range(len(mat1)):
44     for y in range(len(mat1[0])):
45         print (mat1[x][y] - mat2[x][y], end=' '),
46     print
47 #OUTPUT
48 [4, 0],
49 [-2 4]
```

Gambar 9.5 Contoh pengurangan matriks.



- Perkalian Matriks

Perkalian matriks dilakukan dengan menjumlahkan hasil perkalian suatu baris matriks pertama (X) ke kolom matriks kedua (Y). Proses perkalian matriks menggunakan **nested loop for** di dalam nested loop yang kedua. Perulangan ketiga digunakan untuk melakukan proses penjumlahan hasil perkalian baris dan kolom. Hasilnya elemen matriks baru akan ditempatkan pada koordinat tersebut.

```
matrks.py > ...
63 #PERKALIAN MATRIKS
64 X = [[1,2],
65      |
66      |
67      |
68      |
69      |
70      |
71      |
72      |
73 # memparsing baris matriks X
74 for i in range(len(X)):
75     # memparsing kolom matriks Y
76     for j in range(len(Y[0])):
77         # memparsing baris matriks Y
78         for k in range(len(Y)):
79             A[i][j] += X[i][k] * Y[k][j]
80
81 for hasil in A:
82     print(hasil)
83
84 #OUTPUT
85 [14, 16]
86 [28, 32]
```

Gambar 9.6 Contoh perkalian matriks.

Penjelasan :

1. Perhatikan indeks yang didefinisikan yaitu **i, j, k**. **Penting diingat bahwa struktur matriks adalah list of list.**
2. Indeks i berperan membaca sebanyak len(X) dimana len(X) merupakan **jumlah dari baris matriks X**.
3. Indeks j berperan membaca sebanyak len(Y[0]) dimana len(Y[0]) merupakan **jumlah dari kolom matriks Y**.
4. Indeks k berperan membaca sebanyak len(Y) dimana len(Y) merupakan **jumlah dari baris matriks Y**.
5. Kilas balik dahulu terhadap konsep perkalian matriks bahwa baris ke-i kolom ke-k matriks X dikali dengan baris ke-k kolom ke-j matriks Y. Mengapa bisa demikian?
6. Baris X dibaca oleh indeks i.
7. Kolom X dibaca oleh indeks k.
8. Baris Y dibaca oleh indeks k.
9. Kolom Y dibaca oleh indeks j.



10. Dua buah matriks dapat dikali apabila **kolom matriks X == baris matriks Y**.
11. Hal tersebut yang menyebabkan poin no 7 dan 8 berkaitan.
Bahwa **pembacaan kolom matriks X == pembacaan baris matriks Y**.