



UNS
UNIVERSITAS
SEBELAS MARET



MODUL PRAKTIKUM PROGRAMA KOMPUTER

**PROGRAM STUDI TEKNIK INDUSTRI
UNIVERSITAS SEBELAS MARET**

**TIM ASISTEN LABORATORIUM
PERANCANGAN DAN OPTIMASI
SISTEM INDUSTRI 2020**

MODUL VII

MATRIKS

A. Tujuan

Berikut merupakan tujuan Praktikum Programa Komputer Modul VII.

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

B. Matriks

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). Jika banyak baris suatu matriks adalah m dan banyak kolom suatu matriks adalah n, maka matriks tersebut memiliki ordo matriks atau ukuran m x n.

Berikut merupakan contoh matriks.

$$A = \begin{bmatrix} 4 & 4 \\ 1 & -4 \end{bmatrix} \rightarrow A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

C. Membuat Matriks dari List

Untuk membuat matriks dari sebuah list digunakan *nested list*. *Nested list* adalah list di dalam list, sehingga dapat membentuk struktur matriks dua dimensi. Berikut adalah contoh cara membuat matriks dari list Python.

```
# matriks berukuran 2 x 2
matriksA = [ [1, 3],
              [5, 7] ]

print(matriksA)
```

[1] ✓ 0.0s

... [[1, 3], [5, 7]]
<class 'list'>

Gambar 1 Matriks Ordo 2x2

```
# matriks berukuran 3 x 2
matriksB = [ [6, 3],
              [4, 5],
              [2, 1] ]

print(matriksB)
```

[2] ✓ 0.0s

... [[6, 3], [4, 5], [2, 1]]

Gambar 2 Matriks Ordo 3x2

D. Membuat Matriks Menggunakan Perulangan

Untuk membuat matriks menggunakan perulangan digunakan perintah **for**. Berikut adalah contoh cara membuat matriks menggunakan perulangan Python.

```
# matriks berukuran 3 x 3
m = 3
n = 3
x = [0]*m
for i in range(m):
    x[i] = [1]*n
print(x)
```

[3] ✓ 0.0s

... [[1, 1, 1], [1, 1, 1], [1, 1, 1]]

Gambar 3 Matriks Ordo 3x3

Pada program di atas kita membuat sebuah matriks dengan ukuran 3x3 dengan menggunakan dua variabel, yaitu

1. Variabel *m* yang akan digunakan untuk mendefinisikan baris
2. 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 atau bisa diisi dengan nilai berapa pun (tidak boleh list kosong) karena pada akhirnya nya akan di *update* oleh hasil perulangannya.

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

E. Mengakses Sebuah Elemen Pada Matriks

Untuk mengakses elemen pada matriks Python digunakan indeks baris dan indeks kolom. Indeks dimulai dari 0 untuk baris pertama dan kolom pertama. Berikut adalah contoh cara mengakses elemen matriks pada Python:

```
# matriks berukuran 2 x 2
matriksA = [ [1, 3],
              [5, 7]]

print(matriksA)
print(matriksA[1][1]) # mengakses baris 2 kolom 2
print(matriksA[1][0]) # mengakses baris 2 kolom 1
```

[6] ✓ 0.0s

... [[1, 3], [5, 7]]
7
5

Gambar 4 Mengakses Elemen Matriks

F. Operasi Dasar Matriks

1. Penjumlahan Matriks

Penjumlahan matriks hanya dapat dilakukan ketika ordo atau ukuran matriks sama. Operasi penjumlahan matriks dilakukan dengan menjumlahkan komponen-komponen yang seletak. Hasil operasi penjumlahan adalah matriks baru yang memiliki ordo sama dengan matriks semula dengan elemen-elemennya terdiri dari hasil penjumlahan elemen-elemen pada matriks.

Berikut adalah contoh penjumlahan matriks pada Python:

```
# membuat matriks A 2x2
A = [[1, 2],
      [4, 5]]

# membuat matriks B 2x2
B = [[9, 8],
      [6, 5]]

# membuat matriks kosong untuk hasil penjumlahan
C = [[0, 0],
      [0, 0]]

# menjumlahkan matriks A dan B
for i in range(len(A)):
    for j in range(len(A[i])):
        C[i][j] = A[i][j] + B[i][j]
print(C)
```

[9] ✓ 0.0s

... [[10, 10], [10, 10]]

Gambar 5 Contoh Penjumlahan Matriks

```
# membuat matriks A 2x2
A = [[1, 2],
      [4, 5]]

# membuat matriks B 2x2
B = [[9, 8],
      [6, 5]]

# menjumlahkan matriks A dan B
for i in range(len(A)):
    for j in range(len(A[0])):
        print(A[i][j] + B[i][j], end = ' ')
        print(' ')

[6] ✓ 0.0s

... 10 10
    10 10
```

Gambar 6 Contoh Penjumlahan Matriks

2. Pengurangan Matriks

Pengurangan matriks hanya dapat dilakukan ketika ordo atau ukuran matriks sama. Operasi pengurangan matriks dilakukan dengan mengurangi komponen-komponen yang seletak. Hasil operasi pengurangan adalah matriks baru yang memiliki ordo sama dengan matriks semula, dengan elemen-elemennya terdiri dari hasil pengurangan elemen-elemen pada matriks.

Berikut adalah contoh pengurangan matriks pada Python:

```
# membuat matriks A 2x2
A = [[1, 2],
      [4, 5]]

# membuat matriks B 2x2
B = [[9, 8],
      [6, 5]]

# membuat matriks kosong untuk hasil pengurangan
C = [[0, 0],
      [0, 0]]

# mengurangi matriks A dan B
for i in range(len(A)):
    for j in range(len(A[i])):
        C[i][j] = A[i][j] - B[i][j]
    print(C)

[1] ✓ 0.0s

... [[-8, -6], [-2, 0]]
```

Gambar 7 Contoh Pengurangan Matriks

```
# membuat matriks A 2x2
A = [[1, 2],
      [4, 5]]

# membuat matriks B 2x2
B = [[9, 8],
      [6, 5]]

# mengurangkan matriks A dan B
for i in range(len(A)):
    for j in range(len(A[0])):
        print(A[i][j] - B[i][j], end = ' ')
        print(' ')

[7] ✓ 0.0s

... -8 -6
    -2 0
```

Gambar 8 Contoh Pengurangan Matriks

3. Perkalian Matriks

Perkalian matriks dengan matriks hanya dapat dilakukan jika jumlah kolom di suatu matriks sama dengan jumlah baris pada matriks lainnya yang akan dikalikan. Misalnya, suatu matriks memiliki 2 kolom. Maka, matriks tersebut hanya bisa dikalikan dengan matriks lain yang memiliki 2 baris. Perkalian matriks dilakukan dengan mengalikan setiap elemen kedua 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.

Berikut merupakan contoh perkalian matriks pada Python.

```
# membuat matriks A 2x2
A = [[1, 2],
      [4, 5]]

# membuat matriks B 2x2
B = [[9, 8],
      [6, 5]]

# membuat matriks kosong untuk hasil perkalian
C = []

for x in range(0, len(A)):
    row = []
    for y in range(0, len(A[0])):
        total = 0
        for z in range(0, len(A)):
            total = total + (A[x][z] * B[z][y])
        row.append(total)
    C.append(row)

for x in range(0, len(C)):
    for y in range(0, len(C[0])):
        print (C[x][y], end=' ')
    print ()
```

[14] ✓ 0.0s

... 21 18
66 57

Gambar 9 Contoh Perkalian Matriks

Penjelasan:

1. **$A = [[1, 2], [4, 5]]$ dan $B = [[9, 8], [6, 5]]$**
Ini adalah deklarasi dan inisialisasi matriks A dan B dengan ukuran atau ordo 2x2.
2. **$C = []$**
Ini adalah inisialisasi matriks kosong yang akan digunakan untuk menyimpan hasil perkalian matriks A dan B.
3. **$\text{for } x \text{ in range}(0, \text{len}(A))$**
Ini adalah loop pertama yang akan mengiterasi setiap baris di matriks A. Variabel **x** akan berisi indeks baris.
4. **$\text{row} = []$**
Ini adalah inisialisasi list kosong **row** yang akan menyimpan elemen pada baris hasil perkalian.
5. **$\text{for } y \text{ in range}(0, \text{len}(A[0]))$**
Ini adalah *loop* kedua yang akan mengiterasi setiap kolom di matriks A. Variabel **y** akan berisi indeks kolom.
6. **$\text{total} = 0$**
Ini adalah inisialisasi variabel **total** yang akan menyimpan nilai total dari perkalian elemen matriks.
7. **$\text{for } z \text{ in range}(0, \text{len}(A))$**
Ini adalah *loop* ketiga yang akan mengiterasi setiap elemen pada kolom matriks A dan baris matriks B. Variabel **z** akan berisi indeks yang digunakan untuk mengakses elemen di matriks.
8. **$\text{total} = \text{total} + (A[x][z] * B[z][y])$**
Ini adalah perhitungan perkalian elemen matriks A dan B yang akan dijumlahkan pada variabel **total**.
9. **$\text{row.append}(\text{total})$**
Ini akan menambahkan nilai total ke dalam list **row**.
10. **$C.append(\text{row})$**
Ini akan menambahkan list **row** ke dalam matriks hasil perkalian C.