Praktek 1

```python
# impor library numpy
import numpy as np

# membuat array dengan numpy
nilai_siswa = np.array([85, 55, 40, 90])

# akses data pada array
print(nilai_siswa[3])
```

Import adalah cara untuk mengimport library dan numpy adalah sebuah library yang digunakan untuk melakukan komputasi numerik.

Array digunakan untuk membuat objek array multidimensi yang efisien.

Print(nilai_siswa[3]) maksudnya untuk mencetak indeks ke 3 pada data variable, indeks dimulai dari angka 0 dan seterusnya.

Output

```
_________ praktek 1 _________
90
```

Praktek 2

```python
# membuat array dengan numpy
nilai_siswa_1 = np.array([75, 65, 45, 80])
nilai_siswa_2 = np.array([[85, 55, 40], [50, 40, 99]])

# cara akses elemen array
print(nilai_siswa_1[0])
print(nilai_siswa_2[1][1])

# mengubah nilai elemen array
nilai_siswa_1[0] = 88
nilai_siswa_2[1][1] = 70

# cek perubahannya dengan akses elemen array
print(nilai_siswa_1[0])
print(nilai_siswa_2[1][1])

# Cek ukuran dan dimensi array
print("Ukuran Array : ", nilai_siswa_1.shape)
#kolom 4 baris 0
print("Ukuran Array : ", nilai_siswa_2.shape)
#kolom 2 baris 3
print("Dimensi Array : ", nilai_siswa_2.ndim)
#ada 2 dimensi
```

print(nilai_siswa_1[0]) mencetak indeks ke 0

print(nilai_siswa_2[1][1]) mencetak indek ke 1 pada baris ke 1 dan elemen ke 1

penjelasan

baris ke 0 = [85, 55, 40]

{[0], [0]} {[0], [1]} {[0], [2]}

baris ke 1 = [50, 40, 99]

{[1], [0]} {[1], [1]} {[1], [2]}

Nilai (nilai_siswa_1[0]) di ubah menjadi 88

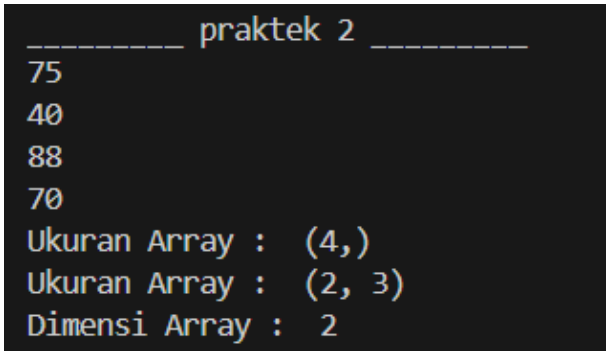
Nilai (nilai_siswa_1[1], [1]) di ubah menjadi 70

print("Ukuran Array : ", nilai_siswa_1.shape) mengukur seberapa panjang nilai_siswa_1

print("Ukuran Array : ", nilai_siswa_2.shape) mengukur seberapa panjang nilai_siswa_2 yakni 3 baris dan 2 kolom

print("Dimensi Array : ", nilai_siswa_2.ndim) menghitung jumlah dimensi

Output



Praktek 3

```python
import numpy as np

# membuat array
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# menggunakan operasi penjumlahan pada 2 array
print(a + b)       # array([5, 7, 9])

# Indexing dan Slicing pada Array
arr = np.array([10, 20, 30, 40])
print(arr[1:3])    # array([20, 30])


# iterasi pada array
for x in arr:
    print(x)
```

Print (a + b) mencetak dan menambahkan data yang ada di dalam variable a dan b

Print (arr[1:3]) mencetak indeks ke 1 dan ke 2, Indeks akhir [3] tidak termasuk, sehingga elemen yang diambil adalah dari indeks 1 hingga 2. Jadi, hasil dari slicing ini adalah array [20, 30].

Print x untuk mencetak perelemen

Output

```
_________ praktek 3 _________
[5 7 9]
[20 30]
10
20
30
40
```

Praktek 4

```python
# membuat array
arr = [1, 2, 3, 4, 5]

# Linear Traversal ke tiap elemen arr
print("Linear Traversal: ", end=" ")
for i in arr:
    print(i, end=" ")
print()
```

end= “ “ artinya menginstruksikan untuk tidak membuat baris baru setelah mencetak, melainkan melanjutkan di baris yang sama.

Variable i untuk mencetak perelemen

Output

```
_________ praktek 4 _________
Linear Traversal:  1 2 3 4 5
```

Praktek 5

```python
# membuat array
arr = [1, 2, 3, 4, 5]

# Reverse Traversal dari elemen akhir
print("Reverse Traversal: ", end="")
for i in range(len(arr) - 1, -1, -1):
    print(arr[i], end=" ")
print()
```

for i in range(len(arr) - 1, -1, -1):

len(arr) - 1 memberikan indeks elemen terakhir.

-1 adalah batas akhir yang menunjukkan iterasi berakhir ketika i kurang dari 0.

-1 pada langkah menunjukkan pengurangan satu pada setiap iterasi.

Output

```
_________ praktek 5 _________
Reverse Traversal: 5 4 3 2 1
```

Praktek 7

```python
# membuat array
arr = [1, 2, 3, 4, 5]

# mendeklarasikan nilai awal
n = len(arr)
i = 0

print("Linear Traversal using while loop: ", end=" ")
# Linear Traversal dengan while
while i < n:
    print(arr[i], end=" ")
    i += 1
print()
```

Len di gunakan untuk menhitung Panjang

While i < n akan membuat perulangan sampai nilai i lebih dari n (Panjang)

`i += 1`

Ini menambah nilai i sebanyak 1 setiap kali loop berjalan, untuk mengakses elemen selanjutnya dalam array.

Output

```
_________ praktek 7 _________
Linear Traversal using while loop:  1 2 3 4 5
```

Praktek 8

```python
# membuat array
arr = [1, 2, 3, 4, 5]

# mendeklarasikan nilai awal
start = 0
end = len(arr) - 1

print("Reverse Traversal using while loop: ", end=" ")
# Reverse Traversal dengan while
while start < end:

    arr[start], arr[end] = arr[end], arr[start]
    start += 1
    end -= 1
print(arr)
```

end = len(arr) - 1: Menginisialisasi indeks akhir pada posisi terakhir dari array. Fungsi len(arr) memberikan panjang array, dan dikurangi 1 untuk mendapatkan indeks terakhir.

while start < end: Loop akan berjalan selama indeks awal (start) lebih kecil dari indeks akhir (end).

arr[start], arr[end] = arr[end], arr[start]: Menukar elemen pada indeks awal dengan elemen pada indeks akhir.

start += 1: Menambahkan 1 pada indeks awal.

end -= 1: Mengurangi 1 pada indeks akhir.

Output

```
_________ praktek 8 _________
Reverse Traversal using while loop:  [5, 4, 3, 2, 1]
```

Praktek 9

```python
# membuat array
arr = [12, 16, 20, 40, 50, 70]

# cetak arr sebelum penyisipan
print("Array Sebelum Insertion : ", arr)

# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

# menyisipkan array di akhir elemen menggunakan .append()
arr.append(26)

# cetak arr setelah penyisipan
print("Array Setelah Insertion : ", arr)

# cetak panjang array setelah penyisipan
print("Panjang Array : ", len(arr))
```

Arr.append( ) di gunakan untuk menyisipkan elemen

Len digunakan untuk menghitung elemen yang ada

Output

```
_________ praktek 9 _________
Array Sebelum Insertion :  [12, 16, 20, 40, 50, 70]
Panjang Array :  6
Array Setelah Insertion :  [12, 16, 20, 40, 50, 70, 26]
Panjang Array :  7
```

Praktek 10

```python
# membuat array
arr = [12, 16, 20, 40, 50, 70]

# cetak arr sebelum penyisipan
print("Array Sebelum Insertion : ", arr)

# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

# menyisipkan array pada tengah elemen menggunakan .insert(pos, x)
arr.insert(4, 5)

# cetak arr setelah penyisipan
print("Array Setelah Insertion : ", arr)

# cetak panjang array setelah penyisipan
print("Panjang Array : ", len(arr))
```

Arr.insert(4, 5) digunakan untuk menyisipkan sebuah elemen di indeks ke (4) dan (5) adalah elemen yang ingin di masukan

```
_________ praktek 10 _________
Array Sebelum Insertion :  [12, 16, 20, 40, 50, 70]
Panjang Array :  6
Array Setelah Insertion :  [12, 16, 20, 40, 5, 50, 70]
Panjang Array :  7
```

Praktek 11

```python
# membuat array
a = [10, 20, 30, 40, 50]
print("Array Sebelum Deletion : ", a)

# menghapus elemen array pertama yang nilainya 30
a.remove(30)
print("Setelah remove(30):", a)

# menghapus elemen array pada index 1 (20)
popped_val = a.pop(1)
print("Popped element:", popped_val)
print("Setelah pop(1):", a)

# Menghapus elemen pertama (10)
del a[0]
print("Setelah del a[0]:", a)
```

a.remove() di gunakan untuk menghapus elemen yang ingin di hapus

a.pop() di gunakan untuk menghapus indeks yang ingin di hapus

del a[0] di gunakan untuk menghapus indeks yang ingin di hapus

a.pop
Menghapus dan mengembalikan elemen dari daftar berdasarkan indeks yang diberikan. Jika indeks tidak ditentukan, elemen terakhir akan dihapus dan dikembalikan.
del
Menghapus elemen atau suatu objek (misalnya, variabel) tanpa mengembalikannya. Dapat digunakan untuk menghapus elemen dengan indeks atau untuk menghapus seluruh daftar.

Output

```
_________ praktek 11 _________
Array Sebelum Deletion :  [10, 20, 30, 40, 50]
Setelah remove(30): [10, 20, 40, 50]
Popped element: 20
Setelah pop(1): [10, 40, 50]
Setelah del a[0]: [40, 50]
```

Praktek 12

```
# impor library numpy
import numpy as np

# membuat matiks dengan numpy
matriks_np = np.array([[1,2,3],
                       [4,5,6],
                       [7,8,9]])
print(matriks_np[2][2])
```

Print(matriks_np[2][2]) maksudnya mencetak elemen dari variable matriks_np

penjelasan

baris ke 0 = [1, 2, 3]

{[0], [0]} {[0], [1]} {[0], [2]}

baris ke 1 = [4, 5, 6]

{[1], [0]} {[1], [1]} {[1], [2]}

Baris ke 2 = [7, 8, 9]

{[2], [0]} {[2], [1]} {[2], [2]}

Output

```
_________ praktek 12 _________
9
```

Praktek 13

```python
# Program penjumlahan matriks yang dibuat dari list

X = [[12,7,3],
    [4,5,6],
    [7,8,9]]

Y = [[5,8,1],
    [6,7,3],
    [4,5,9]]

result = [[0,0,0],
         [0,0,0],
         [0,0,0]]

# proses penjumlahan dua matriks menggunakan nested loop
# mengulang sebanyak row (baris)
for i in range(len(X)):
   # mengulang sebanyak column (kolom)
   for j in range(len(X[0])):
       result[i][j] = X[i][j] + Y[i][j]

print("Hasil Penjumlahan Matriks dari LIST")

# cetak hasil penjumlahan secara iteratif
for r in result:
   print(r)
```

For I in range(len(x)) adalah sebuah loop untuk baris

For j in range(len(x[0])) adalah sebuah loop untuk kolom

`result[i][j] = X[i][j] + Y[i][j]` penjumlahan elemen matriks

Output

```
_________ praktek 13 _________
Hasil Penjumlahan Matriks dari LIST
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]
```

Praktek 14

```python
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array(
    [[5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi penjumlahan dua matrik numpy
result = X + Y

# cetak hasil
print("Hasil Penjumlahan Matriks dari NumPy")
print(result)
```

`Result = X + Y` adalah sebuah penjumlahan elemen matriks yang hasilnya disimpan pada variable result

Output

```
________ praktek 14 ________
Hasil Penjumlahan Matriks dari NumPy
[[17 15  4]
 [10 12  9]
 [11 13 18]]
```

Praktek 15

```python
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array(
    [[5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi pengurangan dua matrik numpy
result = X - Y

# cetak hasil
print("Hasil Pengurangan Matriks dari NumPy")
print(result)
```

`Result = X - Y` adalah sebuah operasi pengurangan elemen matriks yang hasilnya disimpan pada variable result

Output

```
_________ praktek 15 _________
Hasil Pengurangan Matriks dari NumPy
[[ 7 -1  2]
 [-2 -2  3]
 [ 3  3  0]]
```

Praktek 16

```python
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array(
    [[5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi perkalian dua matrik numpy
result = X * Y

# cetak hasil
print("Hasil Perkalian Matriks dari NumPy")
print(result)
```

`Result = X * Y` adalah sebuah operasi perkalian elemen matriks yang hasilnya disimpan pada variable result

Output

```
_________ praktek 16 _________
Hasil Perkalian Matriks dari NumPy
[[60 56  3]
 [24 35 18]
 [28 40 81]]
```

Praktek 17

```python
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array(
    [[5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi pembagian dua matrik numpy
result = X / Y

# cetak hasil
print("Hasil Pembagian Matriks dari NumPy")
print(result)
```

`Result = X / Y` adalah sebuah operasi pembagian elemen matriks yang hasilnya disimpan pada variable result

Output

```
_________ praktek 17 _________
Hasil Pembagian Matriks dari NumPy
[[2.4        0.875      3.        ]
 [0.66666667 0.71428571 2.        ]
 [1.75       1.6        1.        ]]
```

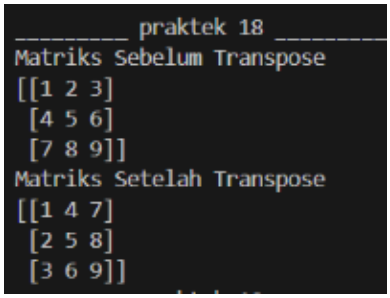
Praktek 18

```python
# impor library numpy
import numpy as np

# membuat matriks
matriks_a = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])

# cetak matriks
print("Matriks Sebelum Transpose")
print(matriks_a)

# transpose matriks_a
balik = matriks_a.transpose()

# cetak matriks setelah dibalik
print("Matriks Setelah Transpose")
print(balik)
```

Transpose adalah operasi dalam matriks yang mengubah baris menjadi kolom dan kolom menjadi baris. Dalam konteks kode yang diberikan.

`Matriks_a.transpose()` digunakan untuk membalik matriks sehingga elemen-elemen di dalamnya tertranspose

Output



Praktek 19

```
# impor library numpy
import numpy as np

# membuat array 1 dimensi
arr_1d = np.array([50, 70, 89, 99, 103, 35])

# cetak matriks sebelum reshape
print("Matriks Sebelum Reshape")
print(arr_1d)
print("Ukuran Matriks : ", arr_1d.shape)
print("\n")

# mengubah matriks menjadi ordo 3 x 2
ubah = arr_1d.reshape(3, 2)

# cetak matriks setelah reshape ke ordo 3 x 2
print("Matriks Setelah Reshape")
print(ubah)
print("Ukuran Matriks : ", ubah.shape)
```

Print(“\n”) untuk mencetak sebuah baris baru

Variable.reshape digunakan untuk mengubah bentuk array tanpa mengubah data yang ada. Dalam contoh di atas, array satu dimensi diubah menjadi dua dimensi dengan 3 baris dan 2 kolom

Output

```
_________ praktek 19 _______
Matriks Sebelum Reshape
[ 50  70  89  99 103  35]
Ukuran Matriks :  (6,)


Matriks Setelah Reshape
[[ 50  70]
 [ 89  99]
 [103  35]]
Ukuran Matriks :  (3, 2)
```

Praktek 20

```python
# vektor baris
vek_1 = np.array([1, 2, 3])

# vektor kolom
vek_2 = np.array([[1],
                  [2],
                  [3]])
# atau menggunakan transpose()
vek_3 = np.array([1, 2, 3]).T

print("Vektor Baris")
print(vek_1)
print("vektor Kolom")
print(vek_2)
print("Vektor Kolom dengan transpose()")
print(vek_3)
```

`print("Vektor Baris")`: Menampilkan judul untuk vektor baris.

`print(vek_1)`: Menampilkan elemen dari vek_1.

`print("vektor Kolom")`: Menampilkan judul untuk vektor kolom.

`print(vek_2):` Menampilkan elemen dari vek_2.

`print("Vektor Kolom dengan transpose()")`: Menampilkan judul untuk vektor kolom hasil transpose.

`print(vek_3):` Menampilkan elemen dari vek_3.

Output

```
_________ praktek 20 _________
Vektor Baris
[1 2 3]
vektor Kolom
[[1]
 [2]
 [3]]
Vektor Kolom dengan transpose()
[1 2 3]
```

Praktek 21

```python
# impor library numpy
import numpy as np

# membuat matriks
matriks_a = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])

# cetak matriks awal
print("Matriks Awal")
print(matriks_a)
print("Ukuran : ", matriks_a.shape)
print("\n")

# ubah matriks menjadi vektor
jd_vektor = matriks_a.flatten()

# cetak vektor
print("Hasil Konversi Matriks ke Vektor")
print(jd_vektor)
print("Ukuran : ", jd_vektor.shape)
```

`Variable.flatten()` digunakan untuk Mengubah matriks 2 atau lebih dimensi menjadi vektor 1D. Semua elemen dari matriks akan diambil dan disusun dalam satu dimensi.

Output

```
_________ praktek 21 _________
Matriks Awal
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Ukuran :  (3, 3)


Hasil Konversi Matriks ke Vektor
[1 2 3 4 5 6 7 8 9]
Ukuran :  (9,)
```