

Tugas Struktur Data

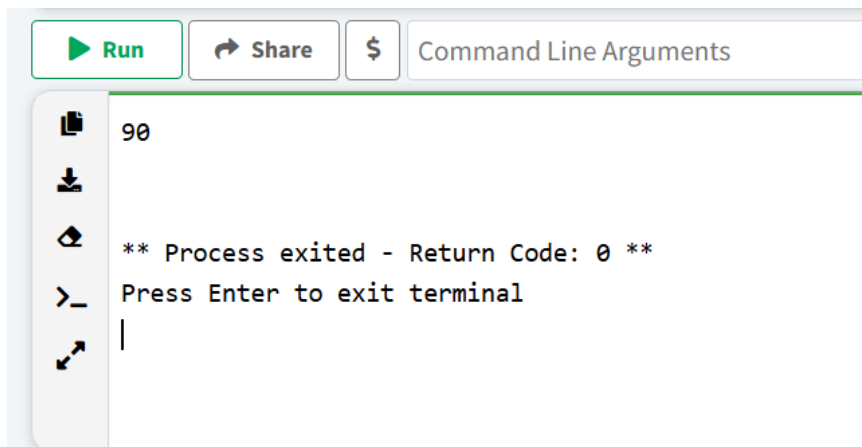
Praktek 1 :

```
# 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])
```

Hasilnya :



The screenshot shows a code execution interface with a top bar containing 'Run', 'Share', and 'Command Line Arguments' buttons. The main area displays the output of the code: the number '90' is printed, followed by a message '** Process exited - Return Code: 0 **' and a prompt 'Press Enter to exit terminal' with a cursor line below it.

Penjelasan :

Baris -1 : Mengimpor library NumPy dengan alias np.

Baris -2 : Membuat array NumPy berisi nilai-nilai siswa

Baris -3 : Menampilkan nilai pada indeks ke-3, yaitu **90**.

Praktek 2 :

```
# impor library numpy
import numpy as np

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

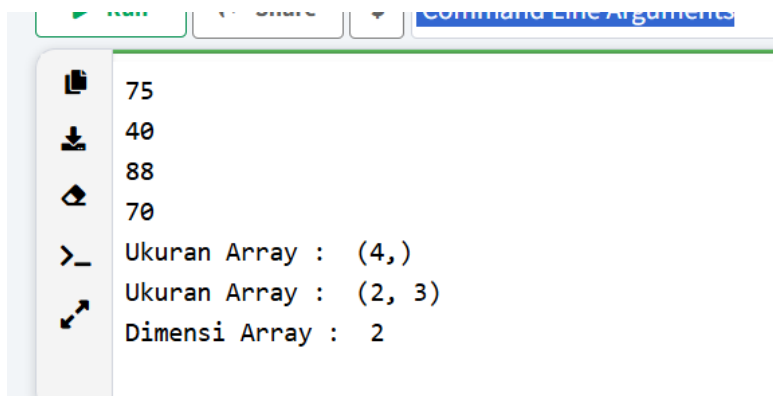
# cara akses elemen array
print(nilai_siswa_1[0])
print(nilai_siswa_2[1][1])

# 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)
print("Ukuran Array : ", nilai_siswa_2.shape)
print("Dimensi Array : ", nilai_siswa_2.ndim)
```

Hasilnya :



Penjelasannya :

Baris -1 : Impor library NumPy.

Baris -2 : Membuat array 1 dimensi (nilai_siswa_1).

Baris -3 : Membuat array 2 dimensi (nilai_siswa_2).

Baris -4 : Menampilkan elemen pertama array 1D → 75.

Baris -5 : Menampilkan elemen baris ke-2, kolom ke-2 \rightarrow 40.

Baris -6 : Mengubah elemen pertama nilai_siswa_1 menjadi 88.

Baris -7 : Mengubah elemen baris ke-2, kolom ke-2 nilai_siswa_2 menjadi 70.

Baris -8 : Menampilkan elemen pertama nilai_siswa_1 setelah diubah \rightarrow 88.

Baris -9 : Menampilkan elemen nilai_siswa_2[1][1] setelah diubah \rightarrow 70.

Baris -10 : Menampilkan ukuran (jumlah elemen) array 1D \rightarrow (4,).

Baris -11 : Menampilkan ukuran array 2D \rightarrow (2, 3) (2 baris, 3 kolom).

Baris -12 : Menampilkan jumlah dimensi array \rightarrow 2.

Praktek 3 :

```
# impor library numpy
import numpy as np

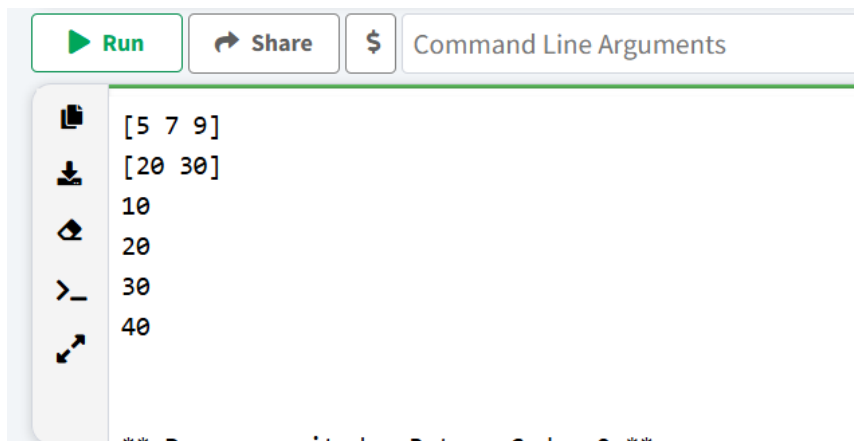
# membuat array
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# 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)
```

Hasilnya :



The screenshot shows a Jupyter Notebook interface. At the top, there are buttons for 'Run' (a green play icon), 'Share' (a share icon), and a '\$' icon for 'Command Line Arguments'. Below these is a code cell with the following Python code:

```
[5 7 9]
[20 30]
10
20
30
40
```

Below the code cell, the output is displayed as a list of values: [5 7 9], [20 30], 10, 20, 30, and 40. At the bottom of the code cell, there is a comment: `** Proses selesai - Return Code: 0 **`.

Penjelasannya :

Baris -1 : Mengimpor library NumPy.

Baris -2 &-3 : Membuat dua array 1 dimensi a dan b.

Baris -4 : Menjumlahkan dua array secara elemen-per-elemen $\rightarrow [1+4, 2+5, 3+6]$.

Baris -5 : Membuat array arr dengan empat elemen.

Baris -6 : Menampilkan elemen dari indeks 1 sampai sebelum 3 $\rightarrow [20, 30]$.

Baris -7 & -8 : Melakukan iterasi (perulangan) dan mencetak setiap elemen dalam array arr.

Praktek 4 :

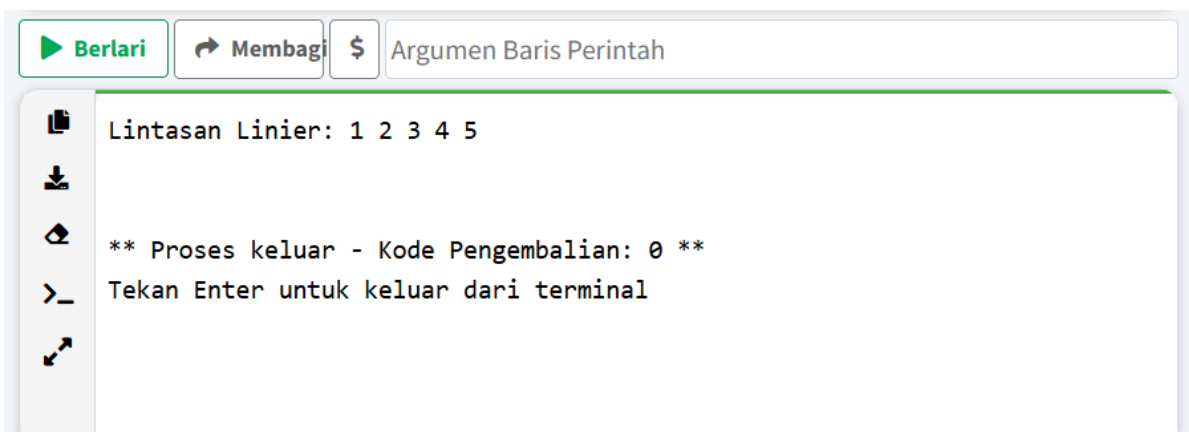


The screenshot shows a code editor with a dark background. The code is as follows:

```
# 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()
```

Hasilnya :



The screenshot shows a terminal window with a light background. At the top, there are buttons for 'Berlari' (a green play icon), 'Membagi' (a share icon), and a '\$' icon for 'Argumen Baris Perintah'. The terminal output is as follows:

```
Lintasan Linier: 1 2 3 4 5
```

Below the output, there is a message: `** Proses keluar - Kode Pengembalian: 0 **`. At the bottom, there is a prompt: `>_ Tekan Enter untuk keluar dari terminal`.

Penjelasannya ;

Baris -1 : Membuat list arr berisi angka 1 sampai 5.

Baris -2 : Mencetak teks "**Linear Traversal:** " tanpa pindah baris.

Baris -3&-4 : Melakukan perulangan untuk mencetak setiap elemen arr dalam satu baris, dipisah spasi.

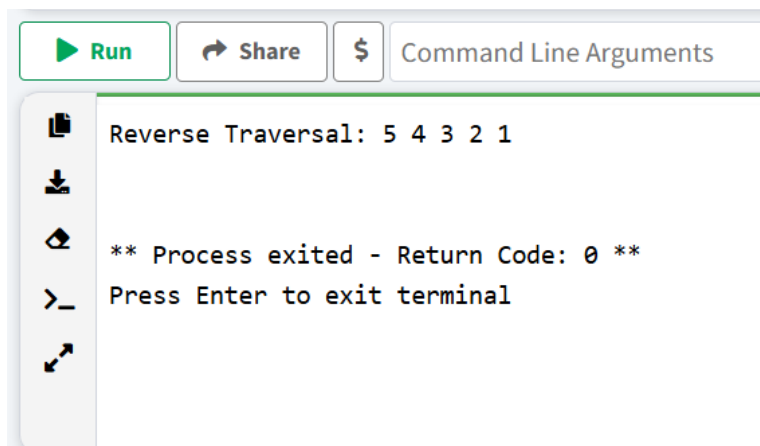
Baris -5 : Pindah baris setelah semua elemen dicetak.

Praktek 5 :

```
# 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()
```

Hasilnya :



```
Run Share $ Command Line Arguments

Reverse Traversal: 5 4 3 2 1

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Penjelasannya :

Baris -1 : Membuat list arr berisi angka 1 sampai 5.

Baris -2 : Mencetak teks "**Reverse Traversal:** " tanpa pindah baris.

Baris -3 : Melakukan perulangan dari indeks terakhir ke pertama:

- `len(arr) - 1` → indeks terakhir (4)
- `-1` → batas akhir (tidak termasuk)
- `-1` → langkah mundur

Baris -4 : Mencetak elemen arr secara terbalik (dari belakang ke depan), dipisah spasi.

Baris -5 : Pindah baris setelah selesai mencetak semua elemen.

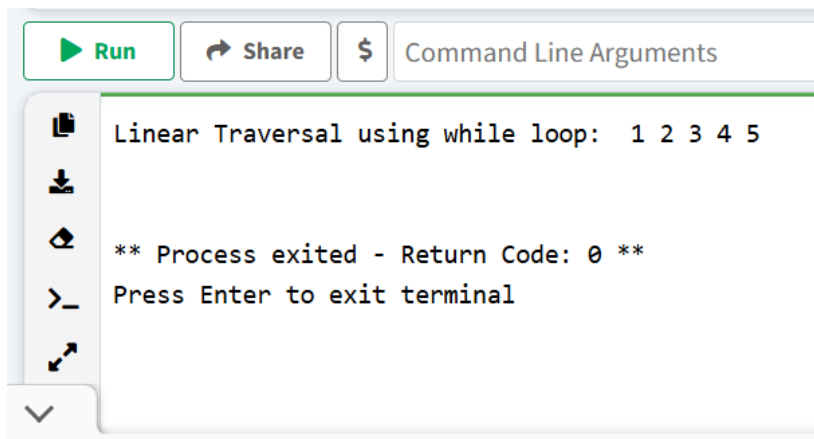
Praktek 7 :

```
# 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()
```

Hasilnya :

The screenshot shows a web-based code editor interface. At the top, there are three buttons: 'Run' (green), 'Share' (blue), and 'Command Line Arguments' (grey). Below these buttons is a terminal window. The terminal output shows the text 'Linear Traversal using while loop: 1 2 3 4 5' on the first line. The second line shows '** Process exited - Return Code: 0 **'. The third line shows '>_ Press Enter to exit terminal'. On the left side of the terminal window, there are icons for file operations (copy, download, upload) and a zoom icon. A dropdown arrow is visible at the bottom left of the terminal window.

Penjelasannya :

Baris -1: Membuat list arr berisi angka 1 sampai 5.

Baris -2 : Menyimpan panjang list arr ke dalam variabel n (hasilnya 5).

Baris -3 : Inisialisasi variabel indeks i dengan nilai awal 0.

Baris -4 : Mencetak teks "**Linear Traversal using while loop:** " tanpa pindah baris.

Baris -5 : Perulangan selama nilai i kurang dari panjang array (n).

Baris -6 : Mencetak elemen ke-i dari array, dipisah spasi.

Baris -7 : Menambah nilai i satu per satu (untuk lanjut ke indeks berikutnya).

Baris -8 : Pindah baris setelah semua elemen dicetak.

Praktek 8 :

```

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

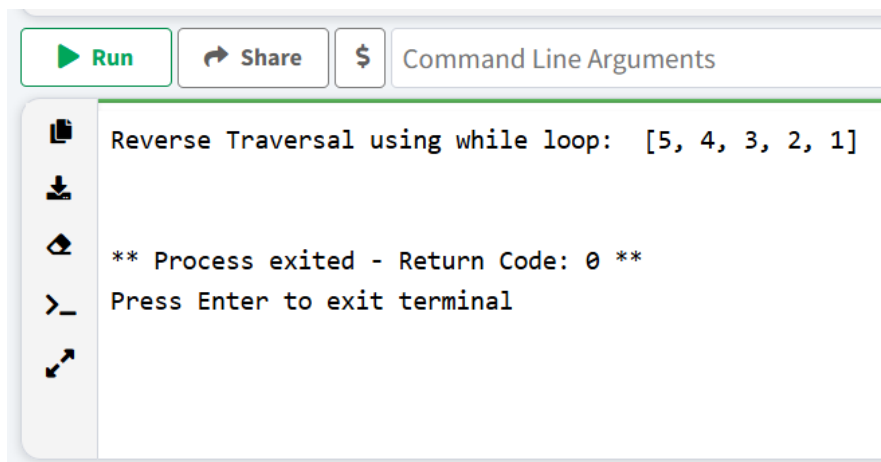
# mendeklarasikan nilai awal
start = 0
end = len(arr) - 1

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)

```

Hasilnya :



The screenshot shows a code execution environment with a toolbar at the top containing 'Run', 'Share', and 'Command Line Arguments' buttons. Below the toolbar, the output of the script is displayed in a terminal window. The output consists of three lines: the first line is 'Reverse Traversal using while loop: [5, 4, 3, 2, 1]', the second line is '** Process exited - Return Code: 0 **', and the third line is 'Press Enter to exit terminal'.

Penjelasannya :

Baris -1 : Membuat list arr berisi angka 1 sampai 5.

Baris -2&-3 : Inisialisasi indeks start (awal) dan end (akhir) untuk proses pembalikan.

Baris -4 : Mencetak teks "**Reverse Traversal using while loop:** " tanpa pindah baris.

Baris -5: Melakukan perulangan selama start lebih kecil dari end.

Baris -6: Menukar elemen di posisi start dengan elemen di posisi end.

Baris -7&-8 : Maju satu langkah dari depan (start) dan mundur satu langkah dari belakang (end).

Baris -9 : Mencetak array yang sudah dibalik.

Output akhir: [5, 4, 3, 2, 1]

Praktek 9 :

```

# 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))

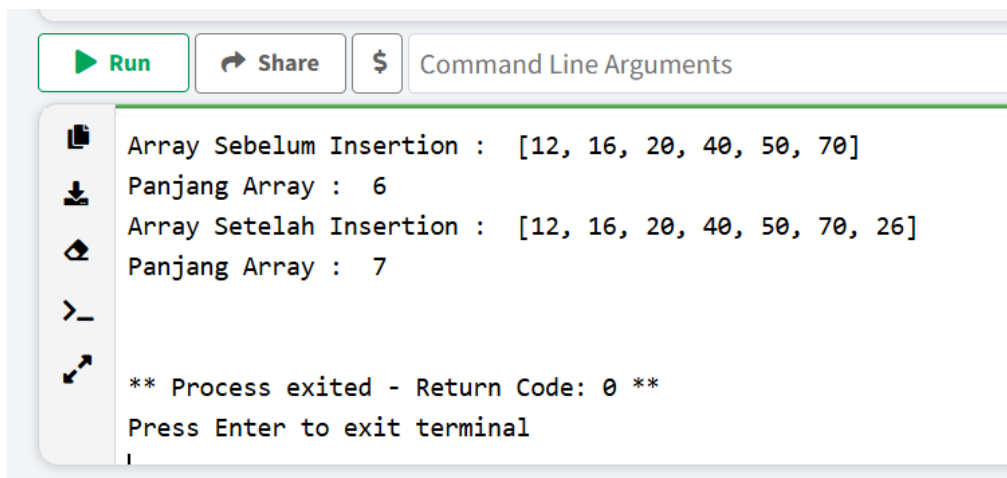
# menyisipkan array di akhir elemen menggunakan .append()
arr.append(26)

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

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

```

Hasilnya :



```

Run Share $ Command Line Arguments
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
** Process exited - Return Code: 0 **
Press Enter to exit terminal

```

Penjelasannya :

Baris -1 : Membuat list arr berisi beberapa angka.

Baris -2 : Menampilkan array sebelum elemen baru ditambahkan.

Baris -3 : Menampilkan jumlah elemen array sebelum penambahan.

Baris -4 : Menambahkan angka 26 ke akhir array menggunakan fungsi append().

Baris -5 : Menampilkan array setelah elemen 26 disisipkan.

Baris -6 : Menampilkan panjang array setelah penambahan elemen.

Praktek 10 :


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

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

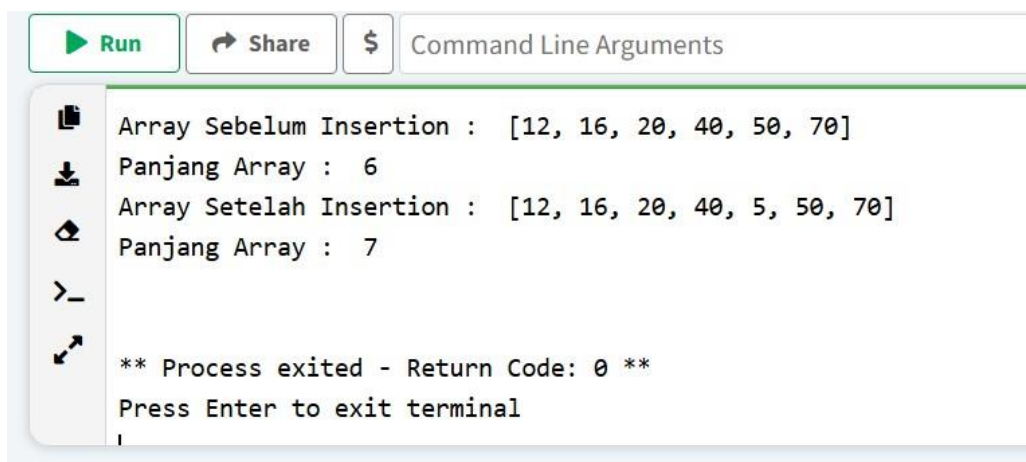
# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

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

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

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

Hasilnya :



```
Run Share $ Command Line Arguments

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

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Penjelasannya :

Baris -1 : Membuat list arr berisi angka.

Baris -2 : Menampilkan array sebelum penambahan elemen.

Baris -3 : Menampilkan jumlah elemen array sebelum penyisipan (hasilnya 6).

Baris -4 : Menyisipkan angka **5** pada indeks ke-4 (sebelum elemen 50).

Baris-5 : Menampilkan array setelah elemen 5 disisipkan.

Baris -6 : Menampilkan panjang array setelah penyisipan (hasilnya 7).

Praktek 11 :

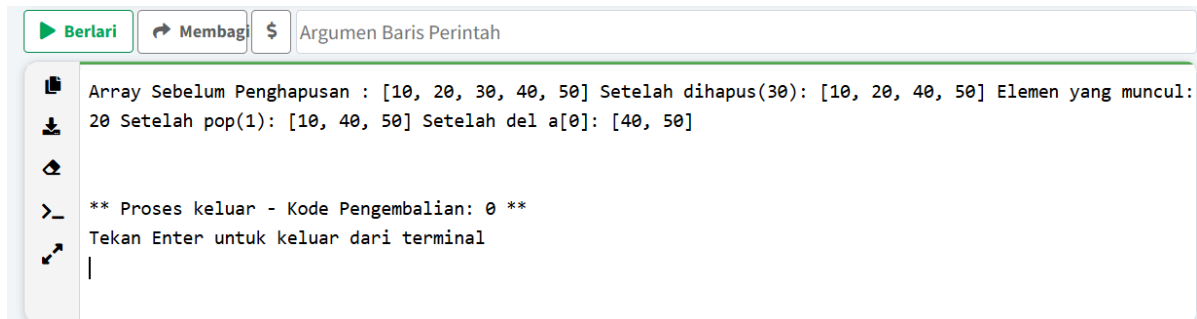
```
# membuat array
a = [10, 20, 30, 40, 50]
print("Array Sebelum Deletion : ", a)

# 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)
```

Hasilnya :



```
Array Sebelum Penghapusan : [10, 20, 30, 40, 50] Setelah dihapus(30): [10, 20, 40, 50] Elemen yang muncul:
20 Setelah pop(1): [10, 40, 50] Setelah del a[0]: [40, 50]

** Proses keluar - Kode Pengembalian: 0 **
Tekan Enter untuk keluar dari terminal
|
```

Penjelasannya :

Baris -1 : **Membuat array/list a** berisi lima elemen: 10, 20, 30, 40, 50.

Baris -2 : **Menampilkan isi array** sebelum ada penghapusan elemen.

Baris -3 : **Menghapus elemen dengan nilai 30** dari array. Metode remove() akan mencari **nilai** (bukan indeks), dan menghapus kemunculan pertamanya.

Baris -4 : Menampilkan isi array **setelah elemen 30 dihapus**.

Baris -5 : **Menghapus elemen pada indeks ke-1** (yaitu 20) dan menyimpannya ke dalam variabel popped_val. Fungsi pop() menghapus **berdasarkan indeks** dan mengembalikan nilainya.

Baris -6 : Menampilkan **nilai yang dihapus** dari array (20).

Baris -7 : Menampilkan isi array **setelah elemen di indeks ke-1 dihapus**.

Baris -8 : **Menghapus elemen di indeks ke-0** (yaitu 10) menggunakan keyword del.

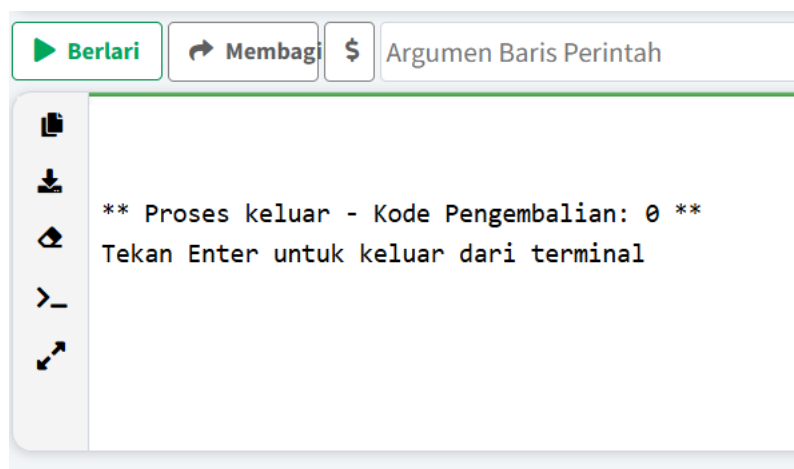
Baris -9 : Menampilkan isi array **setelah elemen pertama dihapus**.

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]])
```

Hasilnya :



Penjelasannya :

Baris -1 : **Mengimpor library numpy** dan memberi alias np. Numpy adalah library Python yang digunakan untuk perhitungan numerik, khususnya array dan matriks.

Baris -2 : **Membuat array dua dimensi (matriks 3x3)** menggunakan fungsi np.array()

Baris -3 : Baris ini menyimpan hasilnya ke dalam variabel matriks_np.

Praktek 13 :

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

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

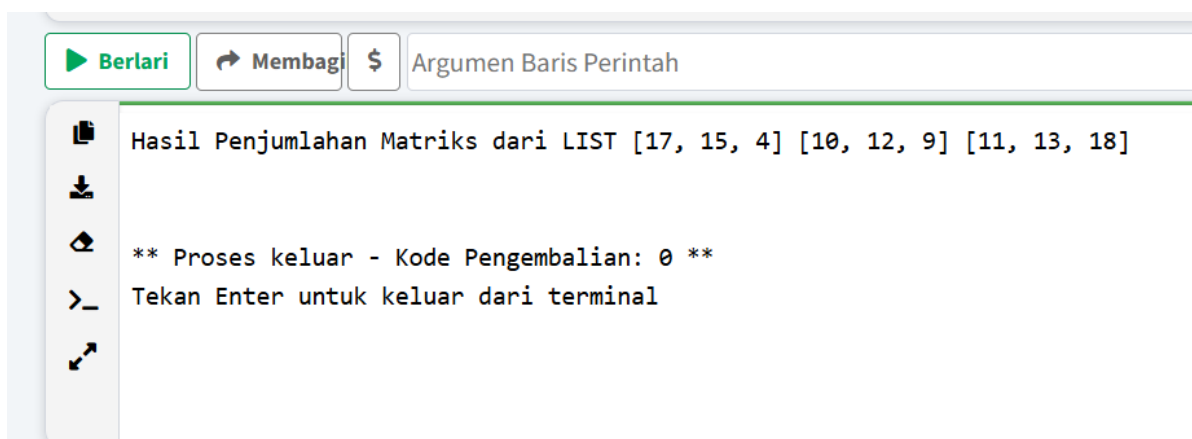
result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

# 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)
```

Hasilnya :



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

** Proses keluar - Kode Pengembalian: 0 **
>_ Tekan Enter untuk keluar dari terminal
```

Penjelasannya :

Baris -1 : pembuka yang menjelaskan bahwa program ini menjumlahkan dua matriks yang dibuat menggunakan list (bukan NumPy).

Baris -2 : Membuat matriks X dalam bentuk list 2D berukuran **3x3**.

Baris -3 : Membuat matriks Y dalam bentuk list 2D **dengan ukuran yang sama** seperti X.

Baris -4 : Membuat matriks result sebagai penampung hasil penjumlahan. Diinisialisasi dengan semua elemen bernilai 0.

Baris -5 : Memulai **loop luar** yang mengakses **baris** (i) dari matriks.

Baris -6 : Memulai **loop dalam** yang mengakses **kolom** (j) dari tiap baris.

Baris -7 : Menjumlahkan elemen pada posisi [i][j] dari X dan Y, lalu menyimpannya di result[i][j].

Baris -8 : Menampilkan teks informasi hasil.

Baris -9 & -10 : **Mencetak hasil akhir** matriks baris demi baris.

Praktek 14 :

```
# impor library numpy
import numpy as np

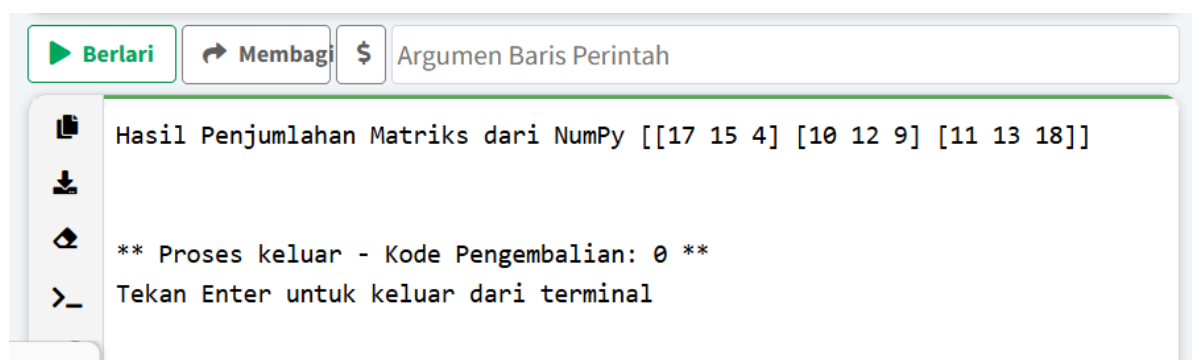
# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

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

# Operasi penjumlahan dua matrik numpy
result = X + Y

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

Hasilnya :



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

** Proses keluar - Kode Pengembalian: 0 **

>_ Tekan Enter untuk keluar dari terminal
```

Penjelasannya :

Baris -1 : Mengimpor library **NumPy** dan memberi alias **np**, agar bisa digunakan untuk operasi numerik seperti matriks/array.

Baris -2 : Membuat **matriks X berukuran 3x3** menggunakan np.array(). Matriks ini bertipe NumPy array.

Baris -3 : membuat **matriks Y berukuran 3x3** sebagai pasangan dari matriks X.

Baris -4 : Melakukan **penjumlahan elemen per elemen** antara matriks X dan Y, lalu menyimpannya dalam variabel result. Karena NumPy mendukung operasi vektor/matriks langsung, tidak perlu pakai loop.

Baris -5 : Menampilkan teks informasi, lalu mencetak isi matriks hasil penjumlahan result.

Praktek 15 :

```
# impor library numpy
import numpy as np

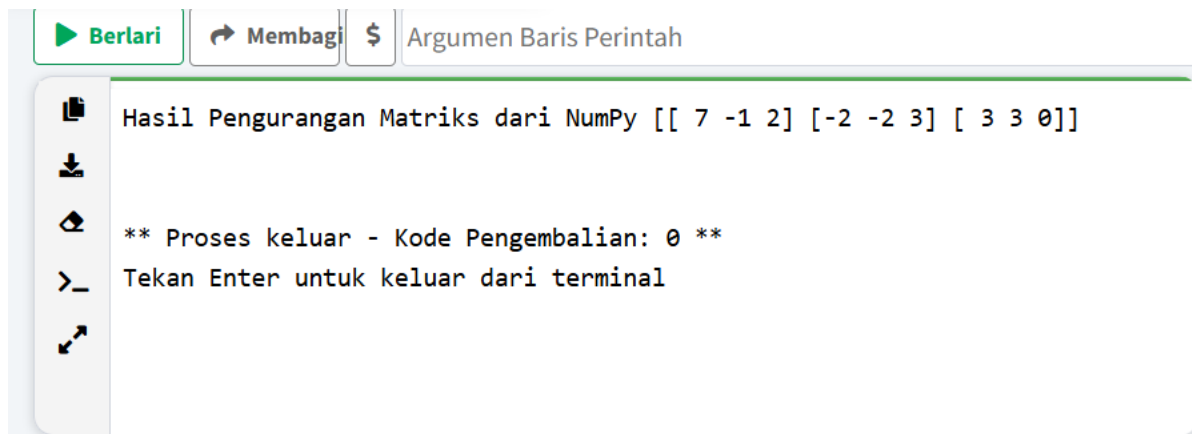
# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

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)
```

Hasilnya :



The screenshot shows a Jupyter Notebook interface with a terminal output. At the top, there are buttons for 'Berlari', 'Membagi', and '\$ Argumen Baris Perintah'. The terminal output displays the result of matrix subtraction: 'Hasil Pengurangan Matriks dari NumPy [[7 -1 2] [-2 -2 3] [3 3 0]]'. Below this, there is a message: '** Proses keluar - Kode Pengembalian: 0 **' and a prompt: '>_ Tekan Enter untuk keluar dari terminal'.

Penjelasannya :

Baris -1 : Mengimpor **library NumPy** dengan alias np. NumPy digunakan untuk operasi numerik seperti array dan matriks.

Baris -2 : Membuat **matriks X berukuran 3x3** dalam bentuk NumPy array.

Baris -3 : Membuat **matriks Y berukuran 3x3** sebagai pasangan dari X yang akan dikurangkan.

Baris -4 : Melakukan **pengurangan elemen per elemen** antara X dan Y. Misalnya, $12 - 5 = 7$, $7 - 8 = -1$, dst. Hasilnya disimpan dalam variabel result.

Baris -5 : Menampilkan teks informasi dan mencetak hasil pengurangan result.

Praktek 16 :

```
# impor library numpy
import numpy as np

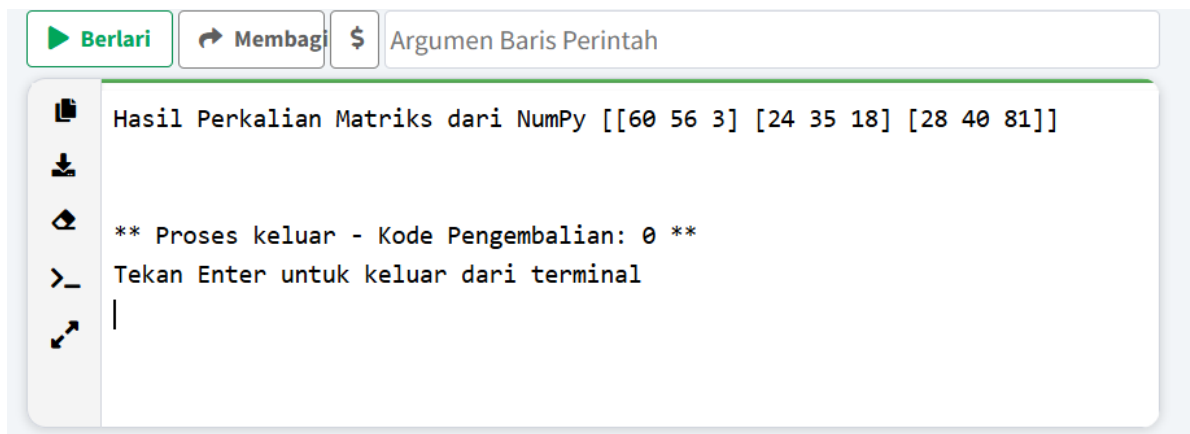
# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

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)
```

Hasilnya :



The screenshot shows a Jupyter Notebook interface. At the top, there are three buttons: 'Berlari' (with a green play icon), 'Membagi' (with a share icon), and a button with a dollar sign icon. Below these is a text input field containing 'Argumen Baris Perintah'. The main area of the notebook is a terminal window. It displays the output of a code cell: 'Hasil Perkalian Matriks dari NumPy [[60 56 3] [24 35 18] [28 40 81]]'. Below this, there is a prompt for the user to press Enter to exit the terminal. The terminal also shows some boilerplate text: '** Proses keluar - Kode Pengembalian: 0 **' and 'Tekan Enter untuk keluar dari terminal'.

Penjelasannya :

Baris -1 : Mengimpor **library NumPy** dengan alias np untuk mendukung operasi array dan matriks.

Baris -2 : Membuat **matriks X 3x3** dalam bentuk NumPy array.

Baris -3 : Membuat **matriks Y 3x3** yang akan dikalikan dengan X.

Baris -4 ; **Melakukan perkalian elemen-per-elemen** (bukan perkalian matriks biasa). Ini disebut **perkalian elemenwise** atau **Hadamard product**:

- $\text{result}[0][0] = 12 * 5 = 60$
- $\text{result}[0][1] = 7 * 8 = 56$
- dst.

Baris -5 : Menampilkan teks informasi dan mencetak hasil dari perkalian elemenwise antar matriks X dan Y.

Praktek 17 :

```
# Praktek 17 : Operasi Pembagian Matriks dengan numpy
# impor library numpy
import numpy as np

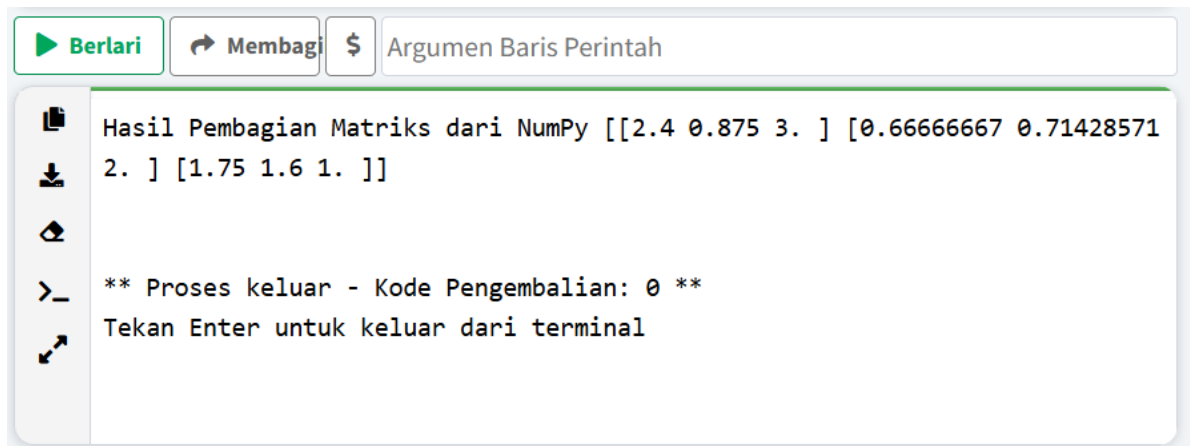
# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

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)
```


Hasilnya :



The screenshot shows a Jupyter Notebook interface with a terminal output. At the top, there are buttons for 'Berlari', 'Membagi', and a dollar sign icon, followed by a text input field labeled 'Argumen Baris Perintah'. The terminal output displays the result of a matrix division operation using NumPy. It shows two 3x3 matrices: the first matrix is $\begin{bmatrix} 2.4 & 0.875 & 3. \\ 2. & 1.75 & 1.6 \\ 1. & 1. & 1. \end{bmatrix}$ and the second matrix is $\begin{bmatrix} 0.66666667 & 0.71428571 & 0.66666667 \\ 0.66666667 & 0.71428571 & 0.66666667 \\ 0.66666667 & 0.71428571 & 0.66666667 \end{bmatrix}$. The output also includes a message: '** Proses keluar - Kode Pengembalian: 0 **' and 'Tekan Enter untuk keluar dari terminal'.

Penjelasannya :

- Baris -1: Mengimpor library **NumPy** dengan alias np, yang digunakan untuk operasi array/matriks.
- Baris -2 : Membuat **matriks X** berukuran 3x3 dalam bentuk numpy.ndarray.
- Baris -3 : Membuat **matriks Y** berukuran 3x3 yang akan digunakan sebagai pembagi elemen-elemen matriks X.
- Baris -4 : Melakukan **pembagian elemen-per-elemen (element-wise)** antara X dan Y.
- Baris -5 : Menampilkan teks informasi, lalu mencetak hasil pembagian matriks X dan Y.

Praktek 18 :

```
# impor library numpy
import numpy as np

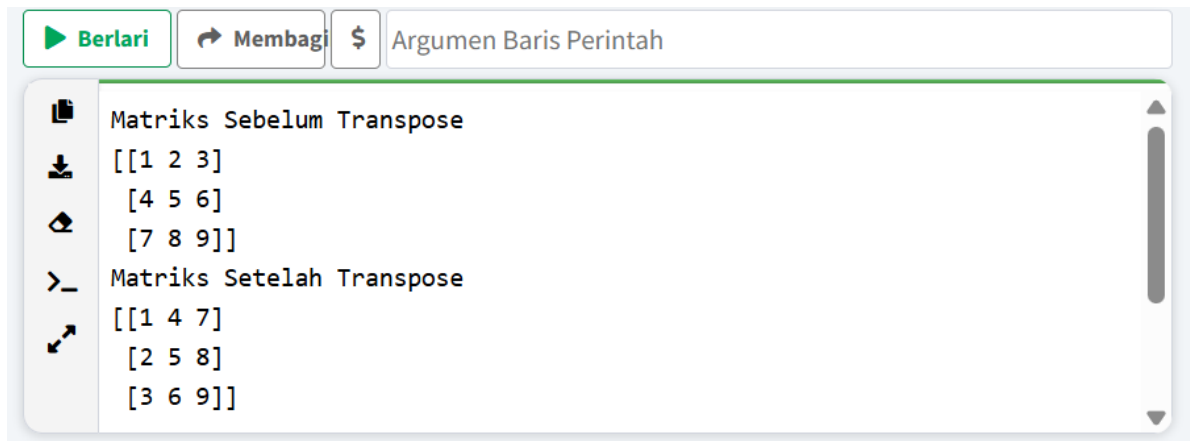
# membuat matriks
matriks_a = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])

# 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)
```

Hasilnya :



Penjelasannya :

Baris -1 : Mengimpor library **NumPy** dengan alias np untuk mendukung operasi array dan matriks.

Baris -2 : Membuat **matriks matriks_a berukuran 3x3** dengan elemen dari 1 hingga 9 menggunakan np.array().

Baris -3 : Menampilkan teks penjas lalu mencetak isi matriks sebelum dilakukan operasi transpose.

Baris -4 : Melakukan **transpose matriks**, yaitu membalik baris menjadi kolom dan sebaliknya.

Baris -5 : Menampilkan teks penjas lalu mencetak matriks hasil transpose.

Praktek 19 :

```
# impor library numpy
import numpy as np

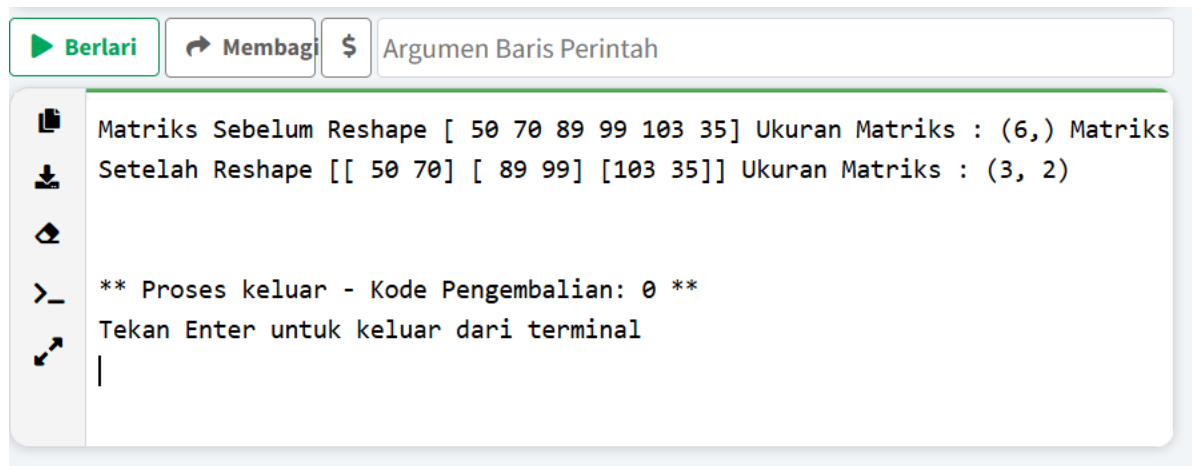
# membuat array 1 dimensi
arr_1d = np.array([50, 70, 89, 99, 103, 35])

# 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)
```

Hasilnya :



The screenshot shows a Jupyter Notebook interface with a terminal output. At the top, there are buttons for 'Berlari', 'Membagi', and a dollar sign icon, followed by a text input field containing 'Argumen Baris Perintah'. The terminal output displays the following text: 'Matriks Sebelum Reshape [50 70 89 99 103 35] Ukuran Matriks : (6,)' followed by 'Matriks Setelah Reshape [[50 70] [89 99] [103 35]] Ukuran Matriks : (3, 2)'. Below this, it shows the output of a print statement: '** Proses keluar - Kode Pengembalian: 0 **' and a prompt 'Tekan Enter untuk keluar dari terminal' with a cursor on the next line.

Penjelasannya :

Baris -1 : Mengimpor library **NumPy** sebagai np untuk operasi array/matriks.

Baris -2 : Membuat **array 1 dimensi** (vektor) dengan 6 elemen.

Baris -3, -4, -5, -6 : Menampilkan isi dan **ukuran array sebelum diubah bentuknya**.

- arr_1d.shape akan menghasilkan (6,) karena array masih 1 dimensi dengan 6 elemen.

Baris -7: Mengubah array 1D menjadi **matriks 2D berukuran 3 baris x 2 kolom** ($3 \times 2 = 6$ elemen, cocok dengan ukuran awal array).

Baris -8 : Menampilkan matriks hasil reshape dan bentuk ukurannya yang baru (3, 2).

Praktek 20 :

```
import numpy as np # WAJIB

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

# vektor kolom
vek_2 = np.array([[1], [2], [3]])

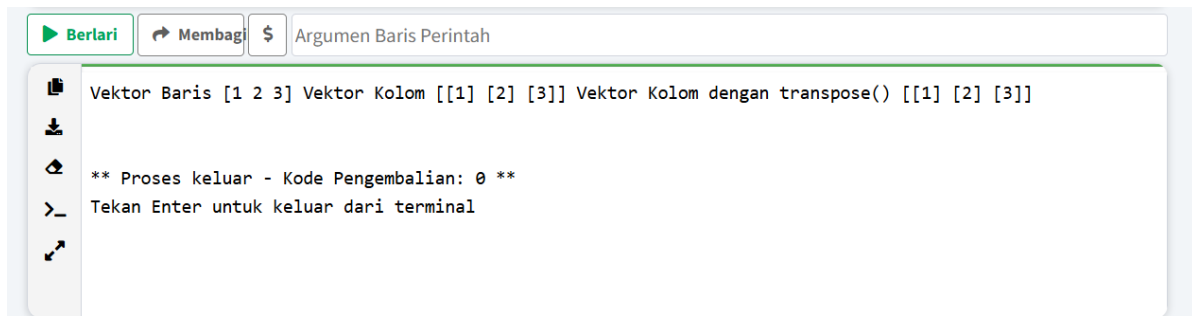
# atau menggunakan reshape untuk transpose
vek_3 = np.array([1, 2, 3]).reshape(-1, 1)

print("Vektor Baris")
print(vek_1)

print("Vektor Kolom")
print(vek_2)

print("Vektor Kolom dengan transpose()")
print(vek_3)
```

Hasilnya :



```
Vektor Baris [1 2 3] Vektor Kolom [[1] [2] [3]] Vektor Kolom dengan transpose() [[1] [2] [3]]

** Proses keluar - Kode Pengembalian: 0 **

Tekan Enter untuk keluar dari terminal
```

Penjelasannya :

Baris -1 : Mengimpor library **NumPy** sebagai np, digunakan untuk manipulasi array dan matriks.

Baris -2 : Membuat **matriks matriks_a berukuran 3x3** menggunakan np.array().

Baris -3, -4, -5, -6 : Menampilkan isi matriks awal dan ukurannya (shape), yaitu (3, 3).

Baris -7: **Mengubah matriks 2D menjadi vektor 1D (rata)** menggunakan flatten(). Semua elemen dijadikan satu baris: [1 2 3 4 5 6 7 8 9].

Baris -8, -9, -10, -11 : Menampilkan isi vektor hasil konversi dan ukurannya. shape akan berubah menjadi (9,), artinya array 1 dimensi dengan 9 elemen.

Praktek 21 ;

```
# 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)
```

Hasilnya :

Berlari

Membagi

Argumen Baris Perintah

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,)

>_

** Proses keluar - Kode Pengembalian: 0 **

Tekan Enter untuk keluar dari terminal

Penjelasannya :

Baris -1 : Mengimpor library **NumPy** sebagai np, digunakan untuk manipulasi array dan matriks.

Baris -2 : Membuat **matriks matriks_a berukuran 3x3** menggunakan np.array().

Baris -3 : Menampilkan isi matriks awal dan ukurannya (shape), yaitu (3, 3).

Baris -4 : **Mengubah matriks 2D menjadi vektor 1D (rata)** menggunakan flatten(). Semua elemen dijadikan satu baris: [1 2 3 4 5 6 7 8 9].

Baris -5, -6, -7 : Menampilkan isi vektor hasil konversi dan ukurannya.

shape akan berubah menjadi (9,), artinya array 1 dimensi dengan 9 elemen.