

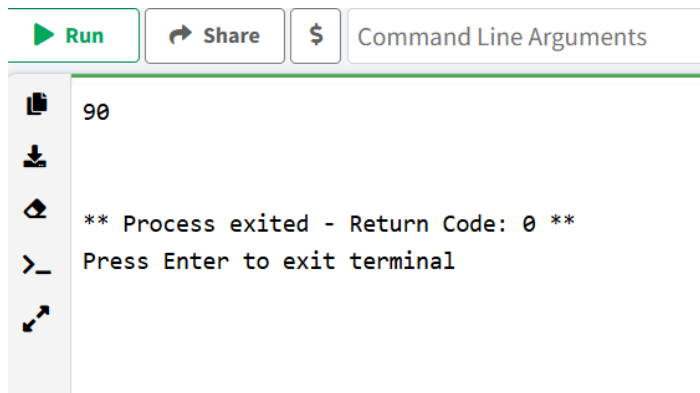
Praktik 1 :

```
# 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])
```

Hasil Run :



Penjelasannya :

Baris -1 Mengimpor library NumPy dan memberi alias np.

Baris -2 Membuat array 1 dimensi berisi angka 1, 2, dan 3.

Baris -3 Membuat array 2 dimensi (matriks) berisi dua baris dan dua kolom.

Praktik 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]])

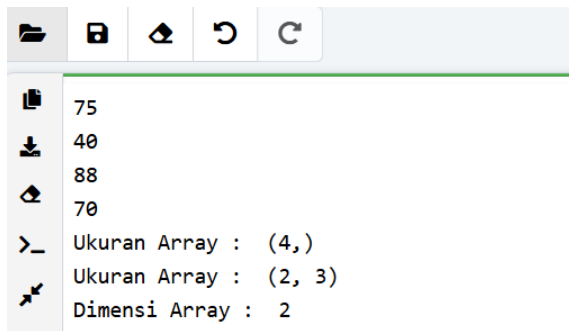
# cara akses elemen array
print(nilai_siswa_1[0])
print(nilai_siswa_2[1][1])

# 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)
```

Hasil run :



Penjelasannya :

Baris -1 Mengimpor NumPy untuk digunakan dalam manipulasi array.

Baris -2 & -3 Membuat array 1 dimensi (nilai_siswa_1) dan array 2 dimensi (nilai_siswa_2).

Baris -4 & -5 Mengakses elemen array: elemen pertama dari nilai_siswa_1 dan baris ke-2 kolom ke-2 dari nilai_siswa_2.

Baris -6 & -7 Mengubah nilai elemen pada array.

Baris -8 & -9 Mengecek nilai yang telah diubah.

Baris -10 , - 11 & -12 Menampilkan ukuran dan jumlah dimensi array:

- shape: menunjukkan ukuran array (jumlah elemen tiap dimensi).
- ndim: menunjukkan jumlah dimensi array.

Praktik 3 :

```
# impor library numpy
import numpy as np

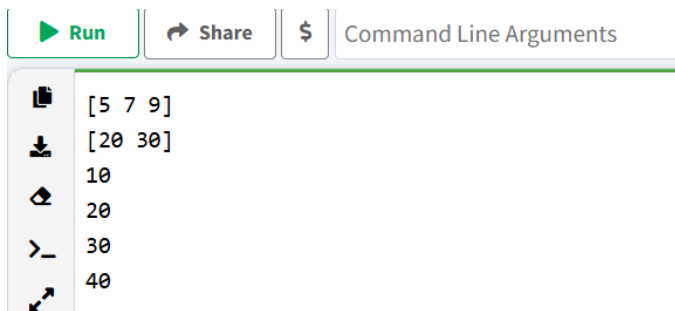
# membuat array
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# 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)
```

Hasil Run :



Penjelasannya :

Baris -1 Mengimpor library NumPy untuk digunakan dalam manipulasi array.

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

Baris -4 Menjumlahkan elemen array a dan b satu per satu:

- $1 + 4 = 5$
- $2 + 5 = 7$
- $3 + 6 = 9$

Baris -5 & -6 Mengambil elemen dari indeks 1 sampai sebelum indeks 3 (yaitu elemen ke-2 dan ke-3).

Baris -7 & -8 Melakukan perulangan untuk mencetak setiap elemen dalam array arr:

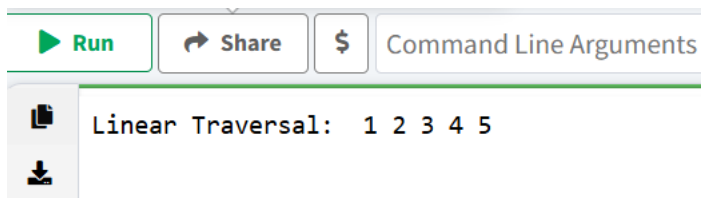
- Output: 10, 20, 30, 40 (dicetak satu per satu).

Praktik 4 :

```
# 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()
```

Hasil run :



Penjelasannya :

Baris -1 Membuat list (array) berisi elemen 1 sampai 5.

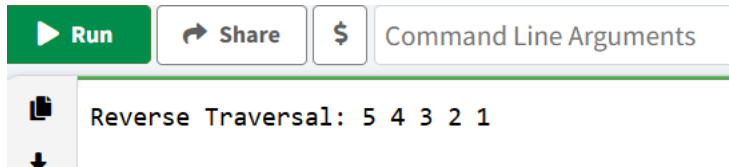
Baris -2 , -3 & -4 Melakukan **linear traversal**, yaitu menelusuri setiap elemen dalam array satu per satu, dan mencetaknya dalam satu baris.

Praktik 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()
```

Hasil run :



The screenshot shows a code editor interface with three buttons at the top: 'Run' (green), 'Share' (blue), and '\$' (grey). Below the buttons, the output of the code is displayed in a terminal window: 'Reverse Traversal: 5 4 3 2 1'.

Penjelasannya :

Baris -1 Membuat list (array) dengan elemen 1 sampai 5.

Baris -2 , -3 & -4 Melakukan **reverse traversal**, yaitu menelusuri array dari elemen terakhir ke pertama.

range(len(arr) - 1, -1, -1) artinya mulai dari indeks terakhir (4) hingga indeks pertama (0), mundur satu per satu.

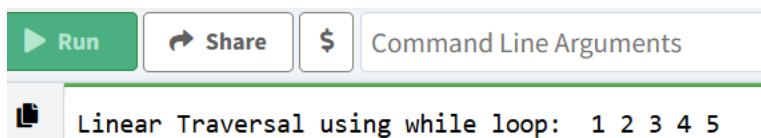
Praktik 7 :

```
# 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()
```

Hasil run :



The screenshot shows a code editor interface with three buttons at the top: 'Run' (green), 'Share' (blue), and '\$' (grey). Below the buttons, the output of the code is displayed in a terminal window: 'Linear Traversal using while loop: 1 2 3 4 5'.

Penjelasannya :

Baris -1 Membuat array berisi angka 1 hingga 5.

Baris -2 & -3 Mendefinisikan n sebagai panjang array dan i sebagai indeks awal.

Baris -4 , -5 , -6 & -7 Melakukan **linear traversal** menggunakan perulangan while. Elemen array dicetak satu per satu dari indeks 0 hingga akhir.

Praktik 8 :

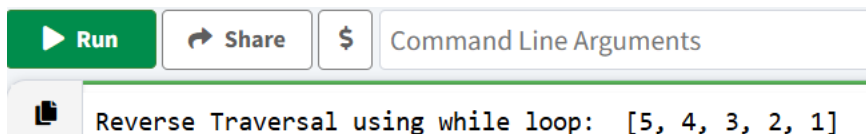
```
# membuat array
arr = [1, 2, 3, 4, 5]

# 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)
```

Hasil run :



Penjelasannya :

Baris -1 Membuat array berisi angka 1 hingga 5.

Baris -2 & -3 start adalah indeks awal, dan end adalah indeks akhir array.

Baris -4 , -5 , -6 , -7 & -8 Melakukan **reverse traversal dan membalik isi array** menggunakan while loop. Elemen paling depan dan paling belakang ditukar, lalu indeks digeser ke tengah hingga selesai.

Baris -9 Menampilkan array yang telah dibalik

Praktik 9 :

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

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

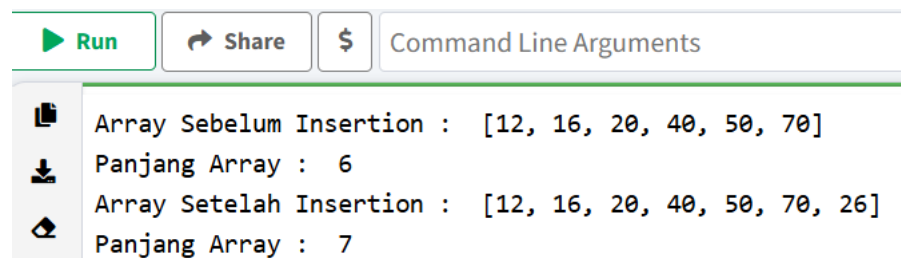
# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

# 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))
```

Hasil run :



```
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
```

Penjelasannya :

Baris -1 Membuat array (list) awal berisi beberapa angka.

Baris -2 & -3 Menampilkan isi array dan panjangnya sebelum penyisipan.

Baris -4 Menyisipkan (menambahkan) elemen 26 di **akhir array** menggunakan metode `.append()`.

Baris -5 & -6 Menampilkan isi array dan panjangnya setelah elemen ditambahkan.

Praktik 10 :

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

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

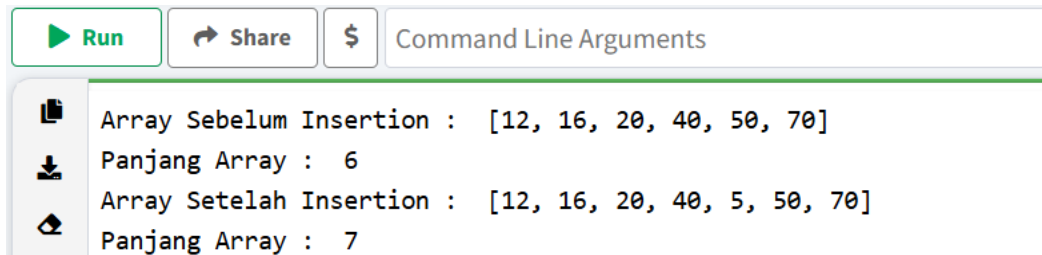
# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

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

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

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

Hasil run :



```
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
```

Penjelasannya :

Baris – 1 Membuat array awal dengan beberapa angka.

Baris -2 & -3 Menampilkan isi dan panjang array sebelum penyisipan.

Baris -4 Menyisipkan elemen 5 pada indeks ke-4 (di antara elemen 50 dan 40).

Metode .insert(pos, x) menambahkan elemen x di posisi pos.

Baris -5 & -6 Menyisipkan elemen 5 pada indeks ke-4 (di antara elemen 50 dan 40).

Metode .insert(pos, x) menambahkan elemen x di posisi pos.