

TUGAS STRUKTUR DATA

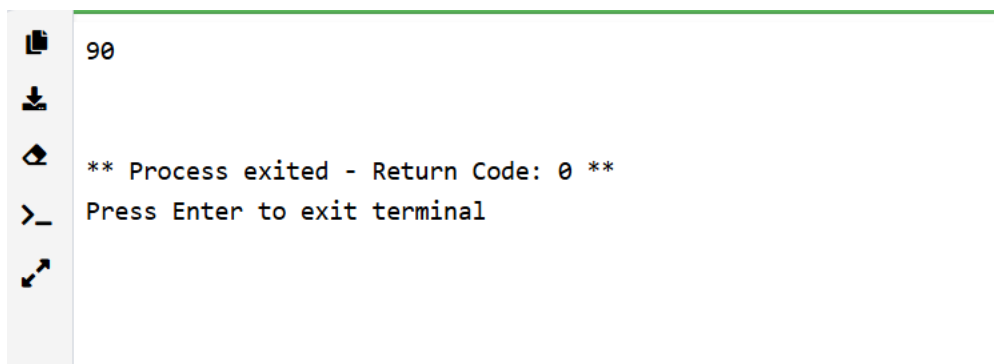
Praktek 1:

```
# 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:



```
90

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

Penjelasan:

Baris ke-1: Baris ini mengimpor library numpy, yaitu library Python yang digunakan untuk operasi numerik, Khususnya array dan matriks. Kata as np digunakan untuk menyingkat penulisan numpy menjadi np, sehingga lebih praktis digunakan.

Baris ke-2: Baris ini membuat sebuah array numpy yang berisi data nilai siswa: 85, 55, 40, dan 90. Fungsi np.array([...]) digunakan untuk membuat array dari daftar (list) Python.

Baris ke-3: Baris ini mencetak elemen ke-4 dari array nilai_siswa, yaitu 90. Perlu diingat bahwa indeks dimulai dari 0, sehingga:

- Nilai_siswa[0] adalah 85
- Nilai_siswa[1] adalah 55
- Nilai_siswa[2] adalah 40
- Nilai_siswa[3] adalah 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:

	75
	40
	88
	70
	Ukuran Array : (4,)
	Ukuran Array : (2, 3)
	Dimensi Array : 2

Penjelasan:

Baris ke-1: Mengimpor library numpy dan disingkat menjadi np agar lebih ringkas saat digunakan.

Baris ke-2: Membuat array dengan numpy.

- nilai_siswa_1: array 1 dimensi yang berisi nilai-nilai siswa.
- nilai_siswa_2: array 2 dimensi (matriks 2 x 3), yang terdiri dari 2 baris dan 3 kolom.

Baris ke-3: Cara akses elemen array

- nilai_siswa_1[0]: mengakses elemen pertama dari array 1 dimensi (75).
- nilai_siswa_2[1][1]: mengakses elemen baris ke-2, kolom ke-2 dari array 2 dimensi (40 sebelum diubah)

Baris ke-4: Mengubah Nilai elemen array

- mengubah nilai pertama di nilai_siswa_1 menjadi 88.
- Mengubah nilai pada baris ke-2, kolom ke-2 di nilai_siswa_2 menjadi 70.

Baris ke-5: Mengecek perubahannya dengan akses elemen array

- Menampilkan nilai yang sudah diperbarui 88 dan 70.

Baris ke-6: Mengecek ukuran dan dimensi array

- shape: menunjukkan ukuran/tata letak array.
 - o nilai_siswa_1. shape = (4,) artinya 1 dimensi dengan 4 elemen.
 - o nilai_siswa_2. shape = (2, 3) artinya 2 baris dan 3 kolom.
- ndim: menunjukkan jumlah dimensi array.
 - o nilai_siswa_2. Ndim = 2 karena array 2 dimensi.

Praktek 3:

```
# impor library numpy
import numpy as np

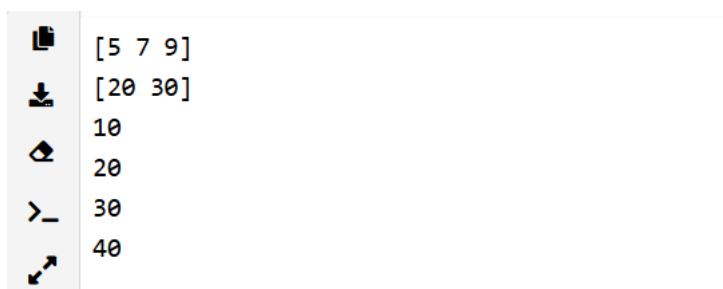
# membuat array
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# 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:



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

Penjelasan:

Baris ke-1: Komentar bahwa baris selanjutnya akan mengimpor library numpy.

Baris ke-2: Mengimpor library numpy dan menyingkatnya menjadi np agar lebih ringkas saat dipakai.

Baris ke-3: Komentar bahwa bagian ini akan membuat array menggunakan numpy.

Baris ke-4: Membuat array a berisi elemen [1,2,3].

Baris ke-5: Membuat array b berisi elemen [4,5,6].

Baris ke-6: Komentar bahwa bagian ini melakukan operasi penjumlahan antar array.

Baris ke-7: Menjumlahkan elemen-elemen array a dan b satu per satu:

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

Baris ke-8: Komentar bahwa bagian ini akan menunjukkan cara mengambil Sebagian elemen dari array (slicing).

Bagian ke-9: Membuat array arr berisi elemen [10,20,30,40].

Bagian ke-10: Mengambil elemen dari indeks ke-1 sampai sebelum indeks ke-3 (slicing)

- Indeks ke-1 = 20
- Indeks ke-2 = 30

Bagian ke-11: Komentar bahwa bagian ini akan melakukan perulangan pada elemen array.

Bagian ke-12: Melakukan iterasi (perulangan) untuk setiap elemen x di dalam array arr.

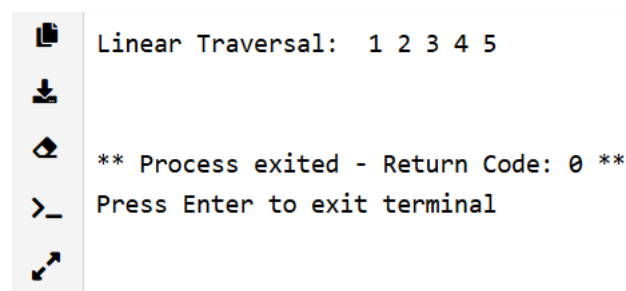
Bagian ke-13: Menampilkan setiap nilai x (elemen array).

Praktek 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()
```

Hasilnya:



```
Linear Traversal: 1 2 3 4 5

** Process exited - Return Code: 0 **

Press Enter to exit terminal
```

Penjelasan:

Baris ke-1: Komentar yang menjelaskan bahwa baris berikutnya akan membuat array atau daftar.

Baris ke-2: Membuat array (list) bernama arr yang berisi lima elemen angka, yaitu 1 sampai 5.

Baris ke-3: Komentar yang menjelaskan bahwa bagian ini akan melakukan penelusuran (traversal) secara linear ke setiap elemen array.

Baris ke-4: Mencetak teks "Linear Traversal:" ke layar tanpa berpindah ke baris baru, agar elemen-elemen array dapat dicetak di baris yang sama.

Baris ke-5: Memulai perulangan for untuk menelusuri setiap elemen di dalam array arr.

Baris ke-6: Mencetak elemen saat ini (i) tanpa berpindah baris, dan memberi spasi setelah setiap angka.

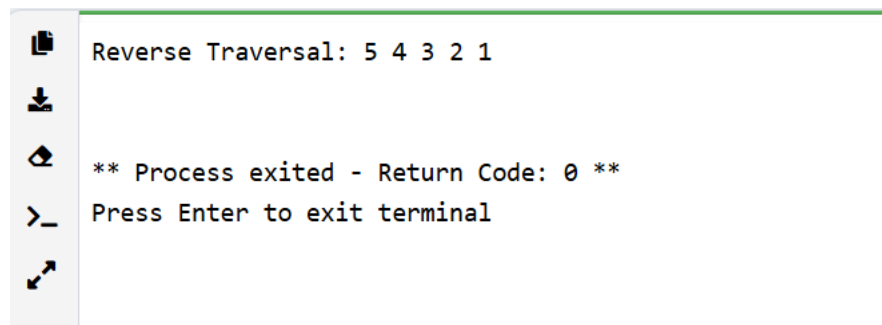
Baris ke-7: Mencetak baris baru setelah semua elemen array ditampilkan, agar tampilan rapi dan tidak menempel ke output berikutnya.

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()
```

Hasilnya:



```
Reverse Traversal: 5 4 3 2 1

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

Penjelasan:

Baris ke-1: Komentar yang menjelaskan bahwa baris selanjutnya akan membuat array atau list.

Baris ke-2: Membuat array (list) bernama arr yang berisi lima elemen: 1, 2, 3, 4, dan 5.

Baris ke-3: Komentar yang menjelaskan bahwa bagian ini akan menelusuri (traversal) elemen array dari belakang ke depan (reverse).

Baris ke-4: Mencetak teks "Reverse Traversal:" ke layar tanpa ganti baris, supaya elemen-elemen bisa muncul di baris yang sama.

Baris ke-5: Melakukan perulangan for untuk mengambil indeks dari elemen terakhir (`len(arr) - 1`) sampai indeks pertama (0) secara mundur (-1 sebagai langkah).

Baris ke-6: Mencetak elemen array berdasarkan indeks `i`, dengan spasi di antaranya dan tetap di baris yang sama.

Baris ke-7: Mencetak baris kosong setelah semua elemen ditampilkan agar output rapi dan tidak menempel ke output berikutnya.

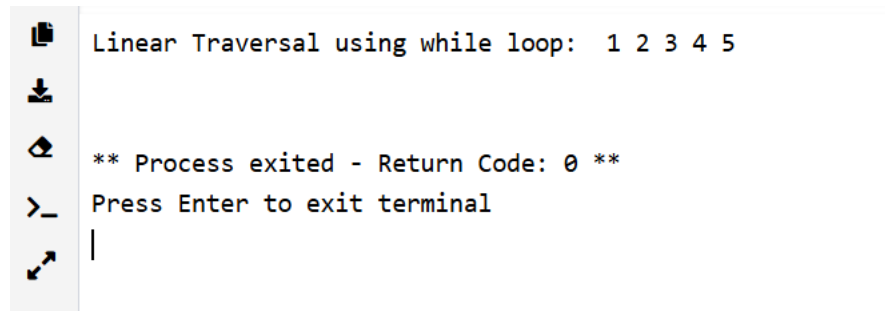
Praktek 7:

```
# 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:



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

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

Penjelasan:

Baris ke-1: Komentar yang menjelaskan bahwa baris berikutnya akan membuat array (list).

Baris ke-2: Membuat array (list) bernama `arr` yang berisi lima elemen: 1, 2, 3, 4, dan 5.

Baris ke-3: Komentar yang menjelaskan bahwa nilai awal akan dideklarasikan sebelum melakukan perulangan.

Baris ke-4: Menyimpan panjang array `arr` ke dalam variabel `n`. Ini akan digunakan sebagai batas akhir perulangan.

Baris ke-5: Menginisialisasi variabel penghitung `i` dengan nilai 0. Ini sebagai indeks awal untuk mengakses elemen array.

Baris ke-6: Mencetak teks "Linear Traversal using while loop:" ke layar tanpa ganti baris, agar hasil traversal muncul di baris yang sama.

Baris ke-7: Komentar yang menjelaskan bahwa bagian ini menggunakan while loop untuk traversal array.

Baris ke-8: Memulai perulangan while selama i masih lebih kecil dari n (panjang array).

Baris ke-9: Mencetak elemen array pada indeks ke-i di baris yang sama, dipisahkan oleh spasi.

Baris ke-10: Menambahkan nilai i sebanyak 1 untuk melanjutkan ke elemen berikutnya.

Baris ke-11: Mencetak baris kosong setelah traversal selesai, supaya output rapi dan tidak menempel pada baris selanjutnya.

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:

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

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

Penjelasan:

Baris ke-1: Komentar yang menjelaskan bahwa baris berikutnya membuat array (list).

Baris ke-2: Membuat array bernama arr yang berisi lima elemen: 1, 2, 3, 4, dan 5.

Baris ke-3: Komentar yang menunjukkan bahwa kita akan mendeklarasikan nilai awal sebelum perulangan.

Baris ke-4: Membuat variabel start dengan nilai 0 sebagai indeks awal.

Baris ke-5: Membuat variabel end dengan nilai indeks terakhir array (panjang array dikurangi 1).

Baris ke-6: Mencetak teks "Reverse Traversal using while loop:" di layar tanpa ganti baris, agar output berada di baris yang sama.

Baris ke-7: Komentar yang menjelaskan bahwa perulangan dilakukan secara mundur menggunakan while.

Baris ke-8: Memulai perulangan while selama nilai start masih lebih kecil dari end.

Baris ke-9: Menukar elemen array pada indeks start dan end. Ini akan membalik posisi elemen dari ujung ke tengah.

Baris ke-10: Menambahkan nilai start sebanyak 1 untuk maju ke indeks berikutnya dari depan.

Baris ke-11: Mengurangi nilai end sebanyak 1 untuk mundur ke indeks sebelumnya dari belakang.

Baris ke-12: Mencetak array setelah proses pembalikan selesai.

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:


```
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
```

Penjelasan:

Baris ke-1: Komentar yang menjelaskan bahwa baris berikutnya membuat array.

Baris ke-2: Membuat array bernama arr yang berisi enam elemen: 12, 16, 20, 40, 50, dan 70.

Baris ke-3: Komentar yang menjelaskan bahwa baris selanjutnya akan mencetak array sebelum dilakukan penyisipan elemen.

Baris ke-4: Mencetak array arr sebelum elemen baru disisipkan.

Baris ke-5: Komentar yang menyatakan bahwa baris berikut akan mencetak panjang array sebelum penyisipan.

Baris ke-6: Mencetak jumlah elemen dalam array arr sebelum penyisipan.

Baris ke-7: Komentar yang menjelaskan bahwa penyisipan akan dilakukan di akhir array menggunakan metode .append().

Baris ke-8: Menyisipkan elemen 26 ke akhir array menggunakan metode .append().

Baris ke-9: Komentar yang menyatakan bahwa array akan dicetak setelah elemen disisipkan.

Baris ke-10: Mencetak array arr setelah elemen 26 ditambahkan.

Baris ke-11: Komentar yang menyatakan bahwa panjang array akan dicetak setelah penyisipan.

Baris ke-12: Mencetak jumlah elemen array setelah elemen baru ditambahkan.

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))

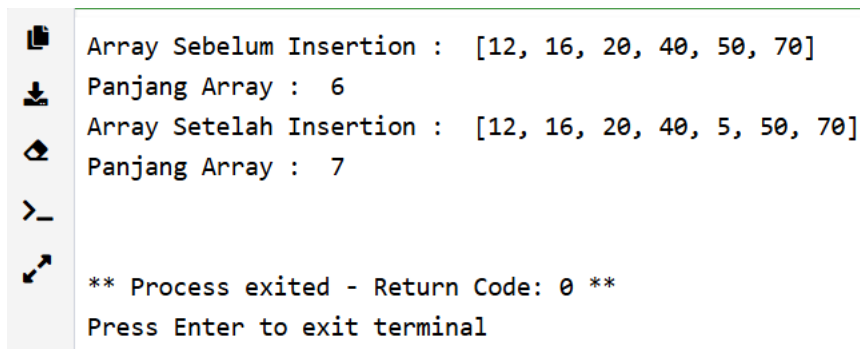
# menyisipkan array pada tengah elemen menggunakan .insert(pos, x)
arr.insert(4, 5)

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

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

```

Hasilnya:



```

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

```

Penjelasan:

Baris ke-1: Komentar yang menjelaskan bahwa array (list) akan dibuat.

Baris ke-2: Membuat array bernama arr dengan isi: 12, 16, 20, 40, 50, 70.

Baris ke-3: Komentar bahwa array akan dicetak sebelum elemen baru disisipkan.

Baris ke-4: Mencetak array sebelum proses penyisipan dilakukan.

Baris ke-5: Komentar bahwa panjang array akan dicetak sebelum penyisipan.

Baris ke-6: Mencetak jumlah elemen array sebelum disisipkan elemen baru.

Baris ke-7: Komentar yang menjelaskan bahwa elemen akan disisipkan di tengah array menggunakan metode .insert(posisi, nilai).

Baris ke-8: Menyisipkan angka 5 ke indeks ke-4 (yaitu sebelum angka 50). Jadi, 5 akan berada di antara 40 dan 50.

Baris ke-9: Komentar bahwa array akan dicetak setelah elemen berhasil disisipkan.

Baris ke-10: Mencetak array setelah elemen 5 disisipkan.

Baris ke-11: Komentar bahwa panjang array akan dicek kembali setelah penyisipan.

Baris ke-12: Mencetak panjang array setelah elemen 5 ditambahkan, kini bertambah menjadi 7 elemen.

Praktek 11:

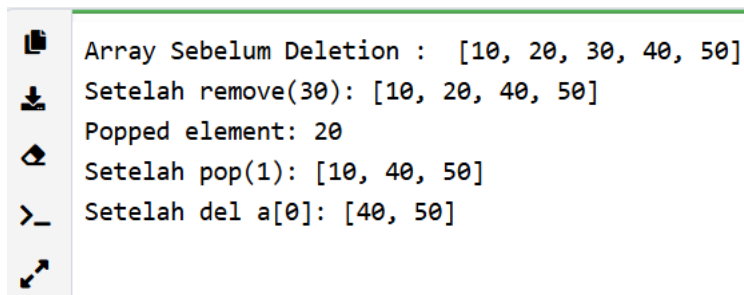
```
# membuat array
a = [10, 20, 30, 40, 50]
print("Array Sebelum Deletion : ", a)

# 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 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]
```

Penjelasan:

Baris 1: Komentar: Membuat array dengan lima elemen.

Baris 2: Mencetak array sebelum dilakukan penghapusan elemen.

Baris 4: Komentar: Menghapus elemen pertama yang bernilai 30 dari array.

Baris 5: Fungsi remove(30) akan mencari dan menghapus elemen dengan nilai 30 pertama yang ditemukan dalam array.

Baris 6: Mencetak array setelah elemen 30 dihapus.

Baris 8: Komentar: Menghapus elemen pada indeks ke-1, yaitu nilai 20.

Baris 9: `pop(1)` akan menghapus elemen di indeks ke-1 dan menyimpannya ke variabel `popped_val`.

Baris 10: Mencetak elemen yang telah dihapus menggunakan `pop()`.

Baris 11: Mencetak array setelah penghapusan dengan `pop(1)`.

Baris 13: Komentar: Menghapus elemen pertama dari array (sisa sekarang adalah 10 di indeks 0).

Baris 14: `del a[0]` menghapus elemen di indeks ke-0 tanpa mengembalikan nilainya.

Baris 15: Mencetak array setelah elemen pertama dihapus menggunakan `del`.

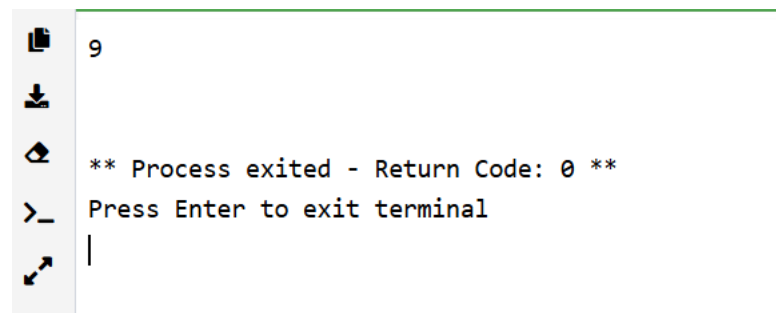
Praktek 12:

```
# 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])
```

Hasilnya:



```
9
** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Penjelasan:

Baris 1: Komentar: Mengimpor library numpy untuk manipulasi array/matriks.

Baris 2: Mengimpor library numpy dan memberi alias `np` agar penulisan lebih ringkas.

Baris 4: Komentar: Membuat matriks 3x3 menggunakan numpy.

Baris 5–7: Membuat array 2 dimensi dengan 3 baris dan 3 kolom, lalu menyimpannya dalam variabel `matriks_np`.

Baris 8: Mencetak elemen pada baris ke-3 dan kolom ke-3 (indeks `[2][2]`), yaitu nilai **9**.

Praktik 13:

```
# 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]]

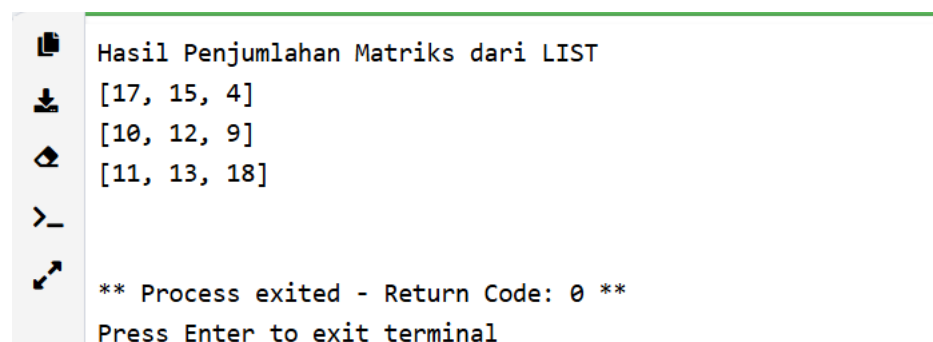
# proses penjumlahan dua matriks menggunakan nested loop
# mengulang sebanyak row (baris)
for i in range(len(X)):
    # mengulang sebanyak column (kolom)
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]

print("Hasil Penjumlahan Matriks dari LIST")

# cetak hasil penjumlahan secara iteratif
for r in result:

# 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]
>_
** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Penjelasan:

Baris 1: Komentar: Program untuk menjumlahkan dua buah matriks yang dibuat dari struktur data list.

Baris 3–5: Membuat matriks X berukuran 3x3 yang berisi angka.

Baris 7–9: Membuat matriks Y berukuran 3x3 sebagai pasangan dari matriks X untuk proses penjumlahan.

Baris 11–13: Membuat matriks kosong result berukuran 3x3 yang akan menyimpan hasil penjumlahan dari X dan Y.

Baris 15: Komentar: Menjelaskan bahwa proses penjumlahan dilakukan dengan menggunakan nested loop (perulangan bersarang).

Baris 16: Komentar: Perulangan luar digunakan untuk mengakses baris-baris pada matriks.

Baris 17: Perulangan dalam digunakan untuk mengakses kolom-kolom pada setiap baris.

Baris 18: Menjumlahkan elemen-elemen yang sesuai dari X dan Y, lalu disimpan ke dalam result.

Baris 20: Mencetak judul hasil penjumlahan matriks.

Baris 22: Komentar: Mencetak hasil penjumlahan matriks secara baris per baris.

Baris 23: Mencetak setiap baris dari matriks hasil (result).

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]]

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

Penjelasan:

Baris 1: Komentar: Mengimpor pustaka/library numpy untuk manipulasi array dan matriks.

Baris 2: Mengimpor modul numpy dengan alias np agar lebih singkat saat digunakan.

Baris 4–7: Komentar: Membuat matriks X berukuran 3x3 menggunakan fungsi np.array().

Baris 9–12: Membuat matriks Y berukuran 3x3 menggunakan np.array() sebagai pasangan untuk dijumlahkan dengan X.

Baris 14: Komentar: Melakukan penjumlahan elemen matriks X dan Y secara langsung dengan operator +.

Baris 15: Variabel result menyimpan hasil penjumlahan matriks X dan Y.

Baris 17: Komentar: Mencetak hasil penjumlahan matriks ke layar.

Baris 18: Menampilkan isi dari matriks result yang merupakan hasil penjumlahan dari X dan Y.

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]])

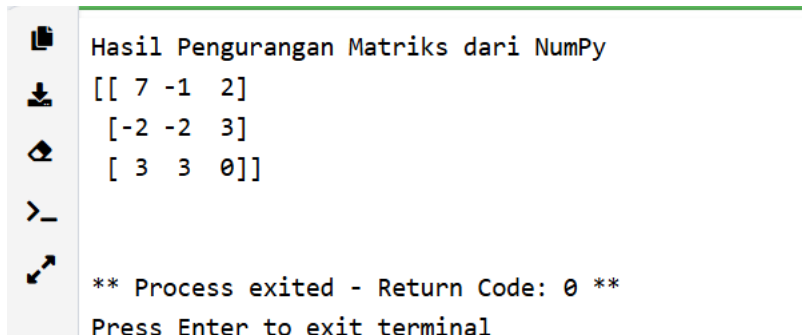
Y = np.array(
    [[5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi pengurangan dua matrik numpy
result = X - Y

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

```

Hasilnya:



```

Hasil Pengurangan Matriks dari NumPy
[[ 7 -1  2]
 [-2 -2  3]
 [ 3  3  0]]

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

```

Penjelasan:

Baris 1: Komentar: Mengimpor pustaka/library numpy untuk memproses array dan matriks.

Baris 2: Mengimpor library numpy dengan alias np agar penulisan lebih ringkas.

Baris 4–7: Komentar: Membuat matriks X berukuran 3x3 menggunakan np.array() berisi elemen-elemen integer.

Baris 9–12: Membuat matriks Y berukuran 3x3 menggunakan np.array(), yang akan digunakan sebagai pengurang matriks X.

Baris 14: Komentar: Melakukan pengurangan elemen-elemen dua matriks NumPy secara langsung.

Baris 15: Melakukan operasi pengurangan X - Y dan menyimpannya dalam variabel result.

Baris 17: Komentar: Mencetak teks sebagai judul hasil pengurangan matriks.

Baris 18: Menampilkan hasil matriks result yang berisi hasil pengurangan setiap elemen X dengan elemen yang sesuai pada Y.

Praktik 16:

```
# impor library numpy
import numpy as np

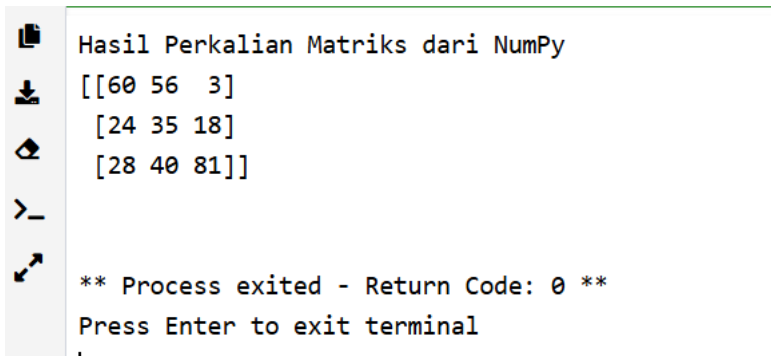
# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

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:



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

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

Penjelasan:

Baris 1: Komentar: Mengimpor pustaka/library numpy untuk memproses array dan matriks.

Baris 2: Mengimpor library numpy dengan alias np agar penulisan lebih ringkas.

Baris 4–7: Komentar: Membuat matriks X berukuran 3x3 menggunakan np.array() dengan elemen-elemen integer.

Baris 9–12: Membuat matriks Y berukuran 3x3 menggunakan np.array(), yang akan digunakan dalam operasi perkalian.

Baris 14: Komentar: Melakukan operasi perkalian dua matriks secara elemen-wise (bukan perkalian matriks biasa).

Baris 15: Melakukan perkalian elemen demi elemen antara X dan Y menggunakan operator * dan menyimpannya dalam variabel result.

Baris 17: Komentar: Menampilkan teks judul untuk hasil perkalian matriks.

Baris 18: Mencetak isi dari matriks result yang merupakan hasil perkalian elemen-elemen pada X dan Y.

Praktik 9:

```
# 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:

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

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

Baris 1: Komentar: Menandai latihan/praktik ke-17 yaitu tentang operasi pembagian matriks menggunakan NumPy.

Baris 2: Komentar: Mengimpor library numpy.

Baris 3: Mengimpor library numpy dengan alias np agar penulisan lebih singkat.

Baris 5–8: Komentar: Membuat matriks X berukuran 3x3 dengan menggunakan np.array() dan mengisinya dengan angka-angka tertentu.

Baris 10–13: Membuat matriks Y berukuran 3x3 dengan menggunakan np.array() yang akan dibagi dengan matriks X secara elemen-wise.

Baris 15: Komentar: Melakukan pembagian antar dua matriks secara elemen demi elemen.

Baris 16: Membagi masing-masing elemen dari matriks X dengan elemen yang sesuai di Y menggunakan operator / (pembagian elemen-wise), lalu menyimpan hasilnya dalam variabel result.

Baris 18: Komentar: Mencetak judul hasil pembagian matriks.

Baris 19: Menampilkan isi dari matriks result, yaitu hasil dari X / Y secara elemen-wise.

Praktik 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:

```

Matriks Sebelum Transpose
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Matriks Setelah Transpose
[[1 4 7]
 [2 5 8]
 [3 6 9]]

```

Penjelasan:

Baris 1: Komentar: Mengimpor library NumPy untuk melakukan operasi numerik.

Baris 2: Mengimpor numpy dan memberi alias np agar mudah digunakan dalam kode.

Baris 4–8: Komentar: Membuat sebuah matriks 3x3 bernama matriks_a menggunakan np.array() dan mengisinya dengan angka 1 sampai 9.

Baris 10: Komentar: Mencetak teks keterangan sebelum matriks ditranspose.

Baris 11: Menampilkan isi matriks_a sebelum dilakukan operasi transpose.

Baris 13: Komentar: Melakukan operasi transpose terhadap matriks_a.

Baris 14: Menggunakan .transpose() untuk membalik baris menjadi kolom dan sebaliknya, hasil disimpan ke variabel balik.

Baris 16: Komentar: Mencetak teks keterangan setelah matriks ditranspose.

Baris 17: Menampilkan isi matriks setelah dilakukan transpose (hasilnya: baris menjadi kolom dan kolom menjadi baris).

Praktik 19:

```
# impor library numpy
import numpy as np

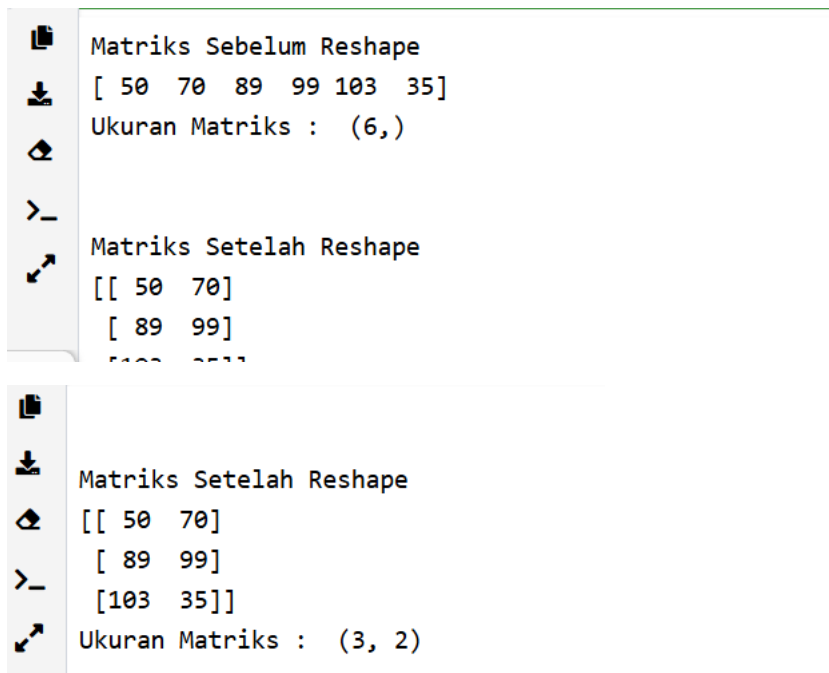
# membuat array 1 dimensi
arr_1d = np.array([50, 70, 89, 99, 103, 35])

# 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:



```
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)
```

Penjelasan:

Baris 1: Komentar: Mengimpor library NumPy untuk operasi array dan matriks.

Baris 2: Mengimpor numpy dan memberi alias np untuk mempersingkat penulisan fungsi NumPy.

Baris 4: Komentar: Membuat array 1 dimensi berisi enam elemen menggunakan np.array.

Baris 5: Membuat array arr_1d dengan elemen [50, 70, 89, 99, 103, 35].

Baris 7: Komentar: Mencetak matriks sebelum dilakukan reshape (perubahan bentuk).

Baris 8: Menampilkan array arr_1d ke layar.

Baris 9: Menampilkan ukuran array (jumlah baris dan kolom) menggunakan .shape.

Baris 10: Menambahkan baris kosong sebagai pemisah.

Baris 12: Komentar: Mengubah bentuk array menjadi matriks berukuran 3 baris \times 2 kolom.

Baris 13: Menggunakan reshape(3, 2) untuk mengubah array 1 dimensi menjadi 2 dimensi.

Baris 15: Komentar: Menampilkan matriks hasil reshape ke ukuran 3 \times 2.

Baris 16: Menampilkan matriks ubah hasil perubahan bentuk.

Baris 17: Menampilkan ukuran (shape) dari matriks hasil reshape menggunakan .shape.











Praktek 20:

```
# 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)
```

Hasilnya:

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

Penjelasan:

Baris 1: Komentar: Mengimpor library NumPy dengan alias np untuk memudahkan penulisan fungsi.

Baris 2: Menjalankan perintah impor NumPy agar dapat digunakan dalam program.

Baris 4: Komentar: Membuat vektor baris.

Baris 5: Membuat array 1 dimensi vek_1 berisi elemen [1, 2, 3].

Baris 7: Komentar: Membuat vektor kolom.

Baris 8: Membuat array 2 dimensi vek_2 berukuran 3x1, yaitu [1], [2], [3], membentuk vektor kolom.

Baris 10: Komentar: Menunjukkan cara lain membuat vektor kolom menggunakan transpose.

Baris 11: Membuat array 2 dimensi [1, 2, 3] lalu mentranspose (.T) sehingga bentuknya menjadi 3 baris dan 1 kolom.

Baris 13: Komentar: Menampilkan label "Vektor Baris".

Baris 14: Menampilkan isi dari vek_1 (array baris).

Baris 15: Komentar: Menampilkan label "vektor Kolom".

Baris 16: Menampilkan isi dari vek_2 (array kolom manual).

Baris 17: Komentar: Menampilkan label "Vektor Kolom dengan transpose()".

Baris 18: Menampilkan isi dari vek_3 (array kolom hasil transpose).

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:

```

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,)
```

Penjelasan:

Baris 1: Komentar: Mengimpor library NumPy dengan alias np.

Baris 2: Menjalankan perintah impor agar bisa menggunakan fungsi-fungsi NumPy.

Baris 4: Komentar: Membuat sebuah matriks.

Baris 5–9: Membuat array 2 dimensi matriks_a berisi 3 baris dan 3 kolom (3x3).

Baris 11: Komentar: Mencetak matriks awal sebelum diubah ke vektor.

Baris 12: Menampilkan label "Matriks Awal".

Baris 13: Menampilkan isi dari matriks_a.

Baris 14: Menampilkan ukuran matriks menggunakan .shape.

Baris 15: Mencetak baris kosong sebagai pemisah tampilan.

Baris 17: Komentar: Mengubah matriks menjadi vektor.

Baris 18: Mengubah matriks_a menjadi array 1 dimensi menggunakan .flatten() dan disimpan di variabel jd_vektor.

Baris 20: Komentar: Mencetak hasil konversi ke vektor.

Baris 21: Menampilkan label "Hasil Konversi Matriks ke Vektor".

Baris 22: Menampilkan isi array jd_vektor.

Baris 23: Menampilkan ukuran vektor menggunakan .shape.