Praktek 1: Membuat Array

```
Modul 2 > Latihan_array.py > ...
1   # Praktek 1 : Membuat array
2   import numpy as np  # Mengimpor library numpy dan memberikan alias 'np'
3
4   # Membuat array dengan numpy
5   nilai_siswa = np.array([85, 55, 40, 90])  # Membuat array satu dimensi dari daftar nilai
6
7   # Akses data pada array
8   print(nilai_siswa[3])  # Mencetak elemen ke-4 (indeks 3), yaitu 90
9
```

Penjelasan:

- Membuat array satu dimensi menggunakan numpy.
- Array ini bisa menyimpan banyak data sekaligus dalam satu variabel, mirip seperti daftar/list biasa di Python, tapi lebih efisien untuk perhitungan numerik.
- print(nilai_siswa[3]) mencetak nilai pada indeks ke-3 (elemen ke-4), yaitu 90.

Praktek 2: Akses, Ubah, Ukuran, dan Dimensi

```
Modul 2 > Latihan_array.py > ...
11  # Praktek 2 : Mengakses, Mengubah, dan Cek Ukuran dan Dimensi Array
12  nilai_siswa_1 = np.array([75, 65, 45, 80])  # Array satu dimensi
13  nilai_siswa_2 = np.array([[85, 55, 40], [50, 40, 99]])  # Array dua dimensi (2 baris, 3 kolom)
14
15  # Akses elemen array
16  print(nilai_siswa_1[0])        # Mencetak elemen pertama dari nilai_siswa_1
17  print(nilai_siswa_2[1][1])     # Mencetak elemen baris ke-2 kolom ke-2 dari nilai_siswa_2
18
19  # Ubah nilai elemen array
20  nilai_siswa_1[0] = 88          # Mengubah elemen pertama menjadi 88
21  nilai_siswa_2[1][1] = 70       # Mengubah elemen [1][1] menjadi 70
22
23  # Cek perubahan
24  print(nilai_siswa_1[0])        # Menampilkan hasil perubahan
25  print(nilai_siswa_2[1][1])     # Menampilkan hasil perubahan
26
27  # Cek ukuran dan dimensi array
28  print("Ukuran Array 1:", nilai_siswa_1.shape)     # Menampilkan ukuran array (jumlah elemen)
29  print("Ukuran Array 2:", nilai_siswa_2.shape)     # Menampilkan dimensi (baris, kolom)
30  print("Dimensi Array 2:", nilai_siswa_2.ndim)     # Menampilkan jumlah dimensi array
31
```

Penjelasan:

- nilai_siswa_1 adalah array 1D (satu dimensi), sedangkan nilai_siswa_2 adalah array 2D (dua dimensi) dengan 2 baris dan 3 kolom.
- Kamu bisa mengakses dan mengubah elemen dengan indeks.
- shape memberi tahu ukuran array (jumlah elemen/baris x kolom).
- ndim memberi tahu jumlah dimensinya (1D atau 2D).

Praktek 3: Operasi Aritmatika, Slicing, Iterasi

Modul 2 > Latihan_array.py > ...

```
33  # Praktek 3 : Operasi Aritmatika Pada Array
34  a = np.array([1, 2, 3])         # Membuat array a
35  b = np.array([4, 5, 6])         # Membuat array b
36
37  # Operasi penjumlahan array
38  print(a + b)                    # Menjumlahkan elemen-elemen pada posisi yang sama
39
40  # Indexing dan Slicing
41  arr = np.array([10, 20, 30, 40])  # Membuat array arr
42  print(arr[1:3])                   # Mengambil elemen dari indeks 1 sampai sebelum 3
43
44  # Iterasi pada array
45  for x in arr:                    # Melakukan perulangan untuk mencetak semua elemen
46      print(x)
47
```

Penjelasan:

- Array a dan b dijumlahkan elemen per elemen → hasil: [5, 7, 9].
- arr[1:3] mengambil sebagian elemen array (slicing) dari indeks 1 sampai sebelum 3 → hasil: [20, 30].
- for x in arr: digunakan untuk mencetak semua elemen array satu per satu.

Praktek 4: Linear Traversal

Modul 2 > Latihan_array.py > ...

```
49  # Praktek 4 : Linear Traversal
50  arr = [1, 2, 3, 4, 5]             # Membuat array list biasa
51  print("Linear Traversal:", end=" ")
52  for i in arr:                     # Menelusuri semua elemen satu per satu
53      print(i, end=" ")
54  print()
55
```

Penjelasan:

- Linear traversal berarti menelusuri array dari awal ke akhir.
- Ini dasar dalam pemrosesan data array — bisa digunakan untuk pencarian, penghitungan, dll.

Praktek 5: Reverse Traversal

```
Modul 2 > Latihan_array.py > ...
57  # Praktek 5 : Reverse Traversal
58  arr = [1, 2, 3, 4, 5]            # Membuat array
59  print("Reverse Traversal:", end=" ")
60  for i in range(len(arr) - 1, -1, -1):  # Perulangan dari belakang ke depan
61      print(arr[i], end=" ")
62  print()
63
```

Penjelasan:

- Menelusuri array dari belakang ke depan.
- Berguna untuk aplikasi seperti pembalikan array atau pemrosesan urutan terbalik.

Praktek 7: Linear Traversal dengan While

```
Modul 2 > Latihan_array.py > ...
65  # Praktek 7 : Linear Traversal dengan While
66  arr = [1, 2, 3, 4, 5]             # Membuat array
67  i = 0                             # Inisialisasi variabel penghitung
68  print("Linear Traversal (while):", end=" ")
69  while i < len(arr):               # Selama i lebih kecil dari panjang array
70      print(arr[i], end=" ")        # Cetak elemen ke-i
71      i += 1                        # Tambah i
72  print()
73
```

Penjelasan:

- Sama seperti traversal biasa, tapi menggunakan while alih-alih for.
- Berguna ketika kamu ingin kontrol lebih manual terhadap iterasi (misalnya melangkah dua-dua).

Praktek 8: Reverse Traversal dengan While

```
Modul 2 > Latihan_array.py > ...
75  # Praktek 8 : Reverse Traversal dengan While
76  arr = [1, 2, 3, 4, 5]              # Membuat array
77  start = 0                          # Inisialisasi indeks awal
78  end = len(arr) - 1                 # Inisialisasi indeks akhir
79  while start < end:                 # Tukar elemen dari luar ke dalam
80      arr[start], arr[end] = arr[end], arr[start]
81      start += 1
82      end -= 1
83  print("Reverse Traversal (while):", arr)  # Menampilkan array setelah dibalik
84
```

Penjelasan:

- Membalik isi array dengan menukar elemen dari ujung ke tengah.
- Teknik ini umum dalam algoritma pembalikan array tanpa membuat array baru.

Praktek 9: Insertion di Akhir Array

```
Modul 2 > Latihan_array.py > ...
86  # Praktek 9 : Insertion di Akhir Array
87  arr = [12, 16, 20, 40, 50, 70]    # Membuat array awal
88  print("Array Sebelum Insertion:", arr)
89  print("Panjang Array:", len(arr))  # Menampilkan panjang array
90
91  arr.append(26)                    # Menambahkan elemen 26 di akhir array
92
93  print("Array Setelah Insertion:", arr)
94  print("Panjang Array:", len(arr))
95
```

Penjelasan:

- Menambahkan elemen di akhir array menggunakan append().
- Metode ini efisien dan paling sering dipakai dalam menambah data baru.

Praktek 10: Insertion di Tengah Array

Modul 2 > Latihan_array.py > ...

```
97  # Praktek 10 : Insertion di Tengah Array
98  arr = [12, 16, 20, 40, 50, 70]
99  print("Array Sebelum Insertion:", arr)
100 print("Panjang Array:", len(arr))
101
102 arr.insert(4, 5)                 # Menyisipkan angka 5 di indeks ke-4
103
104 print("Array Setelah Insertion:", arr)
105 print("Panjang Array:", len(arr))
106
107
108 # Insertion tanpa fungsi insert()
109 arr = [12, 16, 20, 40, 50, 70]
110 print("Array Sebelum Penyisipan:", arr)
111
112 pos = 4                          # Posisi penyisipan
113 x = 5                            # Nilai yang disisipkan
114
115 arr.append(0)                    # Tambah elemen dummy agar panjang array cukup
116
117 # Geser elemen dari belakang ke posisi 'pos'
118 for i in range(len(arr) - 2, pos - 1, -1):
119     arr[i + 1] = arr[i]
120
121 arr[pos] = x                     # Sisipkan nilai pada posisi yang diinginkan
122 print("Array Sesudah Penyisipan:", arr)
123
```

Penjelasan:

- Menyisipkan elemen baru ke posisi tertentu menggunakan insert().
- Juga dijelaskan teknik manual **tanpa fungsi** insert(), yaitu dengan menggeser elemen secara manual ke kanan, lalu menyisipkan nilai baru.

Praktek 11: Menghapus Elemen dari Array

Modul 2 > Latihan_array.py > ...

```
125  # Praktek 11 : Menghapus Array
126  a = [10, 20, 30, 40, 50]
127  print("Array Sebelum Deletion:", a)
128
129  a.remove(30)                    # Menghapus nilai 30
130  print("Setelah remove(30):", a)
131
132  popped_val = a.pop(1)           # Menghapus dan menyimpan elemen indeks ke-1 (yaitu 40 setelah remove)
133  print("Popped element:", popped_val)
134  print("Setelah pop(1):", a)
135
136  del a[0]                        # Menghapus elemen pertama (indeks 0)
137  print("Setelah del a[0]:", a)
138
```

Penjelasan:

- remove(value) menghapus elemen berdasarkan nilainya (pertama yang ditemukan).
- pop(index) menghapus elemen berdasarkan indeks dan **mengembalikannya**.
- del digunakan untuk menghapus elemen berdasarkan indeks **tanpa mengembalikannya**.