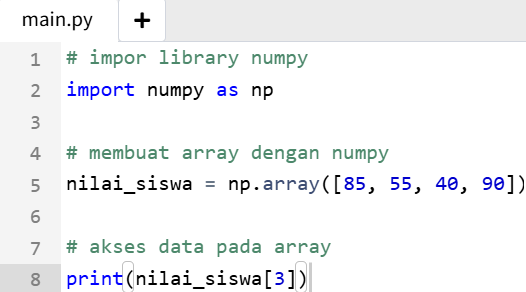
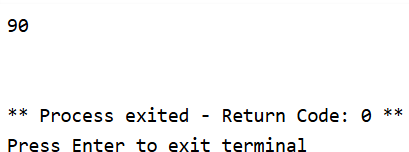
**NAMA:ARIEL NANDA SAFUTRA**

**NIM : 24241021**

**MATKUL : STRUKTUR DATA**

***Praktek 1***

**Outputnya :**

# Penjelasannya :

**Baris 1:**

bahwa baris berikutnya akan mengimpor library NumPy, yang digunakan untuk operasi dan array.

# Baris 2:

Mengimpor library NumPy dan memberi alias np supaya lebih ringkas saat digunakan. Setelah Bisa menggunakan np.array() untuk membuat array,bukan menulis numpy.array().

# Baris 3:

menjelaskan bahwa baris di bawah akan membuat array menggunakan NumPy.

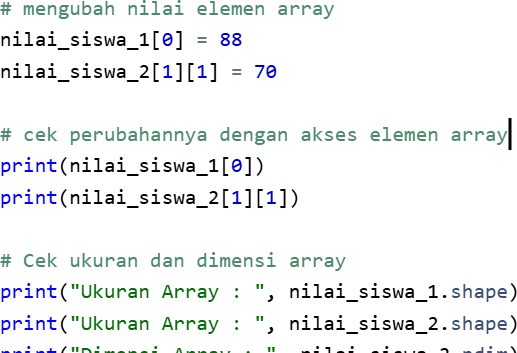
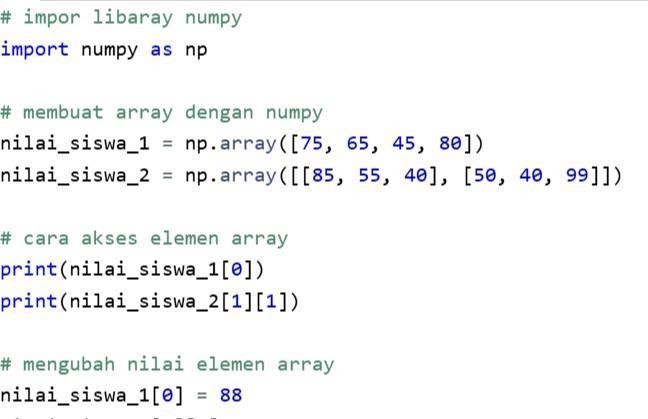
# Baris 4:

Membuat sebuah array NumPy berisi nilai siswa: 85, 55, 40, dan 90. Array ini disimpan dalam Variable.

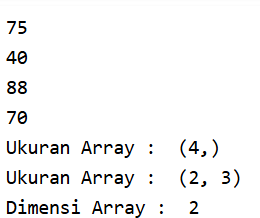
# Penjelasan ouputnya :

Program mencetak 90 ke layar karena itu adalah nilai pada indeks ke-3 dari array cara perhitunganya dari nol mulai dari indeks tersebut maka hasil akhirnya 90.

***Praktek 2***



**Outputnya :**



# Penjelasannya :

**Baris 1**

yang menunjukkan bahwa baris selanjutnya akan mengimpor library NumPy

# Baris 2

Mengimpor library NumPy dan memberinya alias np agar lebih ringkas saat digunakan dalam Kode.

# Baris 3

bahwa kita akan membuat array menggunakan NumPy.

# Baris 4

Membuat array 1 dimensi dengan 4 elemen, lalu disimpan ke variabel nilai\_siswa\_1.

# Baris 5

Membuat array 2 dimensi (seperti matriks 2x3) yang disimpan dalam nilai\_siswa\_2.

# Baris 6

Yang menandai bahwa kita akan mengakses nilai-nilai dalam array.

# Baris 7

Menampilkan elemen pertama dari nilai\_siswa\_1, yaitu 75.

# Baris 8

Menampilkan baris ke-2, kolom ke-2 dari nilai\_siswa\_2, yaitu 40.

# Baris 9

bahwa kita akan mengubah isi array.

# Baris 10

Mengubah elemen pertama dari nilai\_siswa\_1 menjadi 88

# Baris 11

Mengubah elemen baris ke-2, kolom ke-2 dari nilai\_siswa\_2 menjadi 70.

# Baris 12

bahwa kita akan melihat apakah perubahan berhasil.

# Baris 13

Menampilkan elemen pertama dari nilai\_siswa\_1 yang sekarang sudah diubah menjadi 88.

# Baris 14

Menampilkan nilai pada nilai\_siswa\_2[1][1] yang sekarang menjadi 70.

# Baris 15

bahwa kita akan mengecek bentuk dan dimensi array.

# Baris 16

Menampilkan ukuran (jumlah elemen per dimensi) dari nilai\_siswa\_1. Hasil: (4,) → array 1 dimensi dengan 4 elemen.

# Baris 17

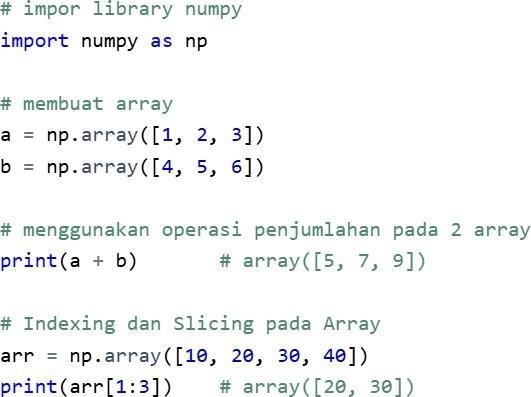
Menampilkan ukuran dari nilai\_siswa\_2.

Hasil: (2, 3) → array 2 dimensi, 2 baris dan 3 kolom.

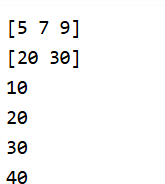
# Baris 18

Menampilkan jumlah dimensi dari nilai\_siswa\_2, yaitu 2 (karena bentuknya seperti tabel/baris- Kolom).

***Praktek 3***

******

**Outputnya :**

****

# Penjelasannya:

**Baris 1**

Komentar yang menjelaskan bahwa library NumPy akan diimpor.

# Baris 2

Mengimpor library NumPy dan memberi alias np supaya lebih singkat saat digunakan.

# Baris 3

bahwa kita akan membuat array NumPy.

# Baris 4

Membuat array a dengan elemen [1, 2, 3].

# Baris 5

Membuat array b = np.array([4, 5, 6]

# Baris 6

bahwa kita akan melakukan penjumlahan antar array

# Baris 7

Menambahkan array a dan b secara elemen (element-wise):

[1+4, 2+5, 3+6] → [5, 7, 9].

# Baris 8

bahwa baris berikut akan menunjukkan teknik mengambil sebagian isi array.

# Baris 9

Membuat array arr dengan 4 elemen: [10, 20, 30, 40]

# Baris 10

Mengambil elemen dari indeks ke-1 hingga sebelum ke-3 (slicing): arr[1:3] → [20, 30].

# Baris 11

bahwa kita akan melakukan iterasi (perulangan) pada elemen array

# Baris 12–13

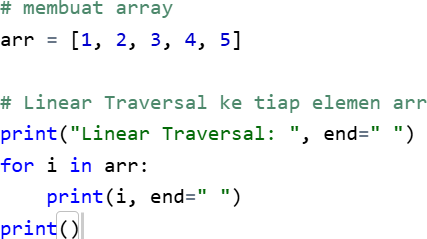
for x in arr:

print(x)

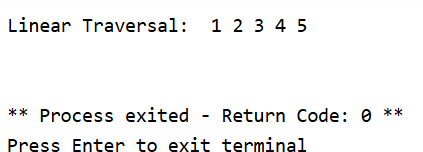
Melakukan loop untuk mencetak setiap elemen dalam array arr.

***Praktek 4***

**Metode Traversal**

****

**Outputnya :**

****

# Penjelasannya :

**Baris 1**

Komentar yang menjelaskan bahwa kamu akan membuat array (dalam bentuk list di Python, Bukan numpyarray).

# Baris 2

Membuat list bernama arr yang berisi lima elemen: [1, 2, 3, 4, 5].

# Baris 3

Komentar bahwa kamu akan melakukan traversal linear, yaitu mengunjungi dan memproses Elemen satu persatu dari ke kiri ke kanan.

# Baris 4

Mencetak teks "Linear Traversal: " tanpa pindah ke baris baru (karena end=" " membuat kursor Tetap dibaris yang sama dan menambahkan spasi).

# Baris 5–6

for i in arr:

print(i, end="

print(i, end=" ") mencetak setiap elemen diikuti oleh spasi, bukan pindah baris.

# Baris 7

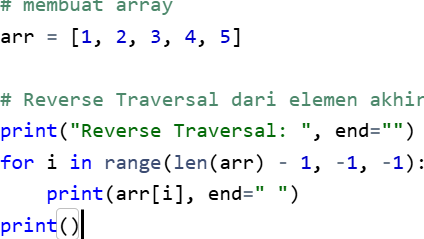
Mencetak baris kosong untuk pindah ke baris baru setelah selesai Mencetak semua elemen baru.

Linear Traversal: 1 2 3 4 5

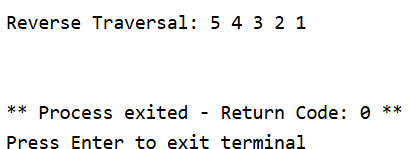
Teks "Linear Traversal: " dicetak terlebih dahulu.

Kemudian setiap elemen 1 2 3 4 5 dicetak di baris yang sama, dipisahkan oleh spasi. Setelah selesai, baris kosong ditambahkan dengan print() untuk menjaga format tampilan.

***Praktek 5***

******

**Outputnya :**



# Penjelasannya :

**Baris 1**

Komentar bahwa kamu akan membuat array (list) di baris berikutnya.

# Baris 2

Membuat list arr yang berisi lima elemen dari 1 sampai 5.

# Baris 3

Komentar bahwa kamu akan mencetak elemen dari list arr secara terbalik (dari belakang Ke depan.

# Baris 4

Mencetak teks "Reverse Traversal: " tanpa pindah baris karena end="" menjaga agar output Selanjutnya dicetak dibaris yang sama.

# Baris 5

len(arr) - 1 = 4 → indeks terakhir (karena jumlah elemen 5 dan indeks mulai dari 0).

-1 adalah batas akhir (exclusive) → berarti iterasi akan berhenti sebelum mencapai indeks -1, Alias berhenti di 0

-1 adalah langkah (step) → artinya mundur satu per satu. Jadi, range(4, -1, -1) menghasilkan urutan indeks: 4, 3, 2, 1, 0

# Baris 6

Untuk setiap indeks i, ambil elemen arr[i] lalu cetak di baris yang sama, dipisahkan dengan spasi.

# Baris7

Pindah ke baris baru setelah mencetak semua elemen, agar output rapi.

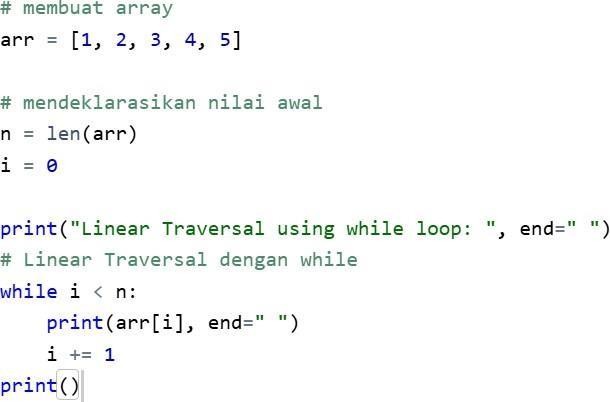
# Penjelasan outputnya :

Reverse Traversal: 5 4 3 2 1

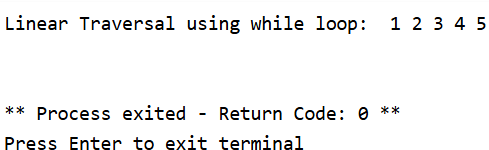
Program mencetak elemen dari indeks terakhir (arr[4] = 5) sampai indeks pertama (arr[0] = 1) Secara mundur

Semuanya dicetak dalam satu baris setelah teks "Reverse Traversal: ".

***Praktek 7***

******

**Outputnya :**

****

# Penjelasannya :

**Baris 1**

bahwakamuakanmembuat array (list).

# Baris 2

Membuat list arrberisi 5 elemen: [1, 2, 3, 4, 5].

# Baris 3

bahwavariabel-variabelawalakandidefinisikan.

# Baris 4

n menyimpanpanjang (jumlahelemen) dari array arr, yaitu 5.

# Baris 5

Variabelidigunakansebagaiindeksawaluntukperulangan. Dimulaidari 0 (indekspertama Array.

# Baris 6

Mencetaktekspembuka, tanpapindah baris, karena end=" " menjaga agar output berikutnya Tetep di baris yang sama

# Baris 7

akanmenggunakanperulangan while untuk traversal.

# Baris 8–10

Perulanganakanberjalanselamaikurangdari n (panjang array). arr[i] mencetakelemenke-idari array.

end=" " agar semuaelemendicetakdalamsatu baris, dipisahkanspasi. i += 1 menaikkanindeks agar pindahkeelemenberikutnya.

Loop inimencetak 1 2 3 4 5

# Baris 11

Pindahke baris barusetelahselesaimencetakelemen array.

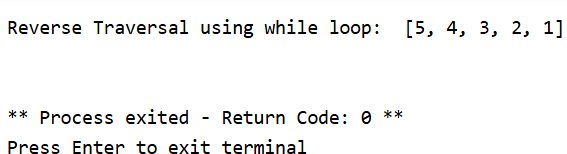
# Penjelasanoutputnya :

Linear Traversal using while loop: 1 2 3 4 5

Program menelusuri list darielemenpertamahinggaterakhirmenggunakan while, dan Mencetaksemuaelemensecaraberurutan.

***Praktek 8***

**Outputnya :**

****

# Penjelasannya :

**Baris 1**

membuatsebuah array (list).

# Baris 2

Membuat list bernamaarrberisielemen [1, 2, 3, 4, 5].

# Baris 3

menetapkanvariabelawaluntukindeks traversal.

# Baris 4–5

start disetkeindekspertama (0).

end disetkeindeksterakhir (len(arr) - 1 = 4). Variabeliniakandigunakanuntukmenukarelemendariujungketengah. **Baris 6**

Mencetaktekssebagaiketerangan, tanpapindahke baris baru (end=" ").

# Baris 7

melakukanpembalikanisi array denganperulangan while.

# Baris 8–11

Loop akanterusberjalanselama start < end. Di dalam loop:

Elemen pada posisi start dan end ditukar (swap).

Kemudian start majukekanan (+1) dan end mundurke kiri (-1). Proses inimembalikurutanelemendariluarkedalam.

1. start=0, end=4: tukar 1 dan 5 → [5, 2, 3, 4, 1]
2. start=1, end=3: tukar 2 dan 4 → [5, 4, 3, 2, 1]
3. start=2, end=2: kondisi start < end sudahtidakterpenuhi, loop berhenti.

# Baris 12

Mencetakisi array setelahdibalik. Hasil akhirnya:

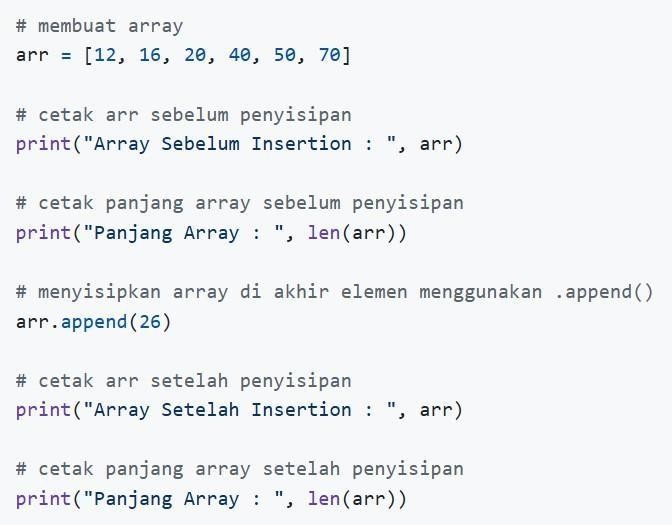
[5, 4, 3, 2, 1]

# Penjelasanoutputnya :

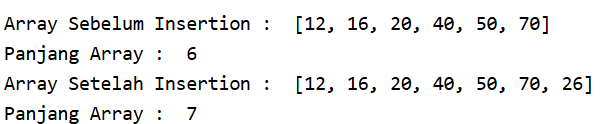
Reverse Traversal using while loop: [5, 4, 3, 2, 1]

Array awal [1, 2, 3, 4, 5] dibalikurutannyamenjadi [5, 4, 3, 2, 1].

Proses inidisebut reverse in-place karenadilakukanlangsung di array yang samatanpa Membuat array baru.

***Praktek 9***

**Outputnya :**



# Penjelasannya :

**Baris 1**

membuat array (dalam Python disebut list).

# Baris 2

Membuat list arrdengan 6 elemenangka: [12, 16, 20, 40, 50, 70]

# Baris 3

mencetakisi array sebelumelemenbarudisisipkan.

# Baris 4

Mencetakisi list arrsebelumadaperubahan:

Output:

Array SebelumInsertion : [12, 16, 20, 40, 50, 70]

# Baris 5

mencetakjumlahelemen list sebelumpenambahan.

# Baris 6

Menggunakanlen(arr) untukmenghitungjumlahelemen, yaitu 6. Output:

Panjang Array : 6

# Baris 7

menambahkanelemen di akhir list denganfungsi .append().

# Baris 8

Menambahkanangka 26 keakhir list arr.

List berubahmenjadi: [12, 16, 20, 40, 50, 70, 26]

# Baris 9

mencetak array setelahpenambahanelemen.

# Baris 10

Mencetakisi array setelahpeambahan output: array setelahinsertion : [12, 16, 20, 40, 50, 70,

26]

**Baris 11** mencetakjumlahelemensetelahpenambahan. **Baris 12**

Mencetakjumlahelemensaatini, yaitu 7. Output:

Panjang Array : 7

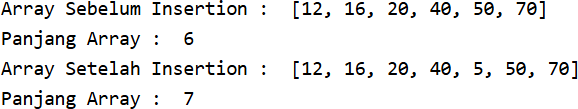
# Penjelasanoutputnya :

Array sebeluminsertion:[12, 16, 20, 40, 50, 70] Panjang Array : 6

Array SetelahInsertion : [12, 16, 20, 40, 50, 70, 26] Panjang Array : 7

***Praktek 10***

**Outputnya :**

****

# Penjelasannya :

**Baris 1**

membuat array (dalam Python disebut list).

# Baris 2

Membuat list arrdengan 6 elemenangka: [12, 16, 20, 40, 50, 70]

# Baris 3

mencetakisi array sebelumelemenbarudisisipkan.

# Baris 4

Mencetakisi list arrsebelumadaperubahan:

Output:

Array SebelumInsertion : [12, 16, 20, 40, 50, 70]

# Baris 5

mencetakjumlahelemen list sebelumpenambahan.

# Baris 6

Menggunakanlen(arr) untukmenghitungjumlahelemen, yaitu 6. Output:

Panjang Array : 6

# Baris 7

menambahkanelemen di akhir list denganfungsi .append().

# Baris 8

Menambahkanangka 26 keakhir list arr.

List berubahmenjadi: [12, 16, 20, 40, 50, 70, 26]

# Baris 9

mencetak array setelahpenambahanelemen.

# Baris 10

Mencetakisi array setelahpeambahan output: array setelahinsertion : [12, 16, 20, 40, 50, 70,

26]

**Baris 11** mencetakjumlahelemensetelahpenambahan. **Baris 12**

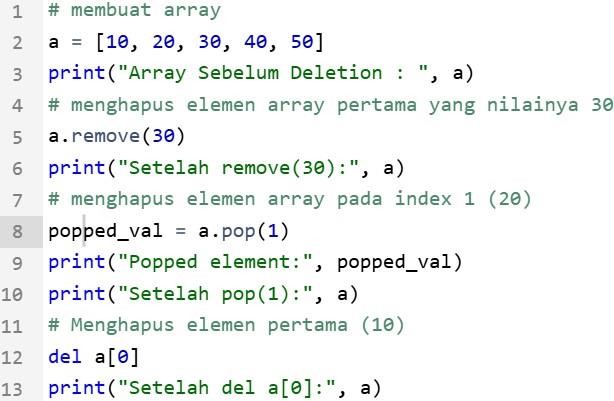
Mencetakjumlahelemensaatini, yaitu 7. Output:

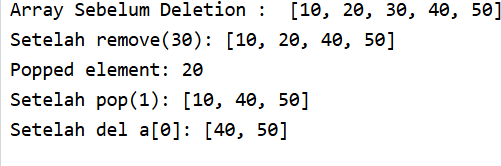
Panjang Array : 7

# Penjelasanoutputnya :

Array sebeluminsertion:[12, 16, 20, 40, 50, 70] Panjang Array : 6

Array SetelahInsertion : [12, 16, 20, 40, 50, 70, 26] Panjang Array : 7

***Praktek 11***

**Output**

**Penjelasan:**

**Baris 1**

membuat sebuah list bernama a. Isinya: [10, 20, 30, 40, 50].

# Baris 2

Menampilkan isi list a sebelum ada perubahan.

# Baris 3

Fungsi remove() digunakan untuk menghapus elemen berdasarkan nilai, bukan berdasarkan posisi.

Python akan mencari nilai 30 dalam list, lalu menghapus elemen

# Baris 4

Menampilkan isi list setelah nilai 30 dihapus.

# Baris 5

* Fungsi pop() digunakan untuk menghapus

elemen berdasarkan indeks dan mengembalikannya.

* pop(1) artinya hapus elemen pada indeks ke- 1, yaitu 20.
* Nilai yang dihapus disimpan dalam variabel popped\_val.

# Baris 6

Menampilkan nilai yang dihapus dengan pop().

# Baris 7

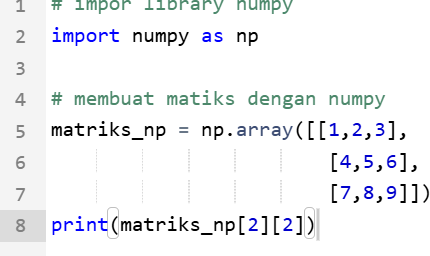
Menampilkan isi list setelah pop(1) dilakukan.

**Baris :**

* del adalah keyword untuk menghapus sesuatu.
* del a[0] artinya menghapus elemen pada indeks ke-0, yaitu 10.

# Baris 9

Menampilkan isi list setelah elemen pertama dihapus.

***Praktek12:***

**Outputnya:**

****

**Penjelasannya:**

**Baris 1:**

* mengimpor library numpy dan memberinya alias np.
* numpy adalah library Python yang sangat kuat untuk manipulasi array dan operasi matematika tingkat lanjut, terutama untuk data berbentuk matriks atau vektor.

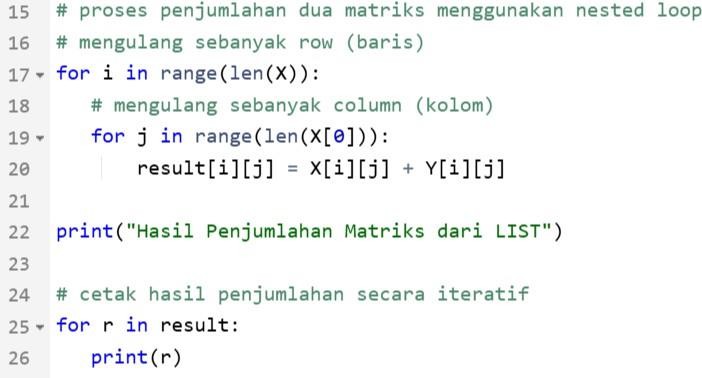
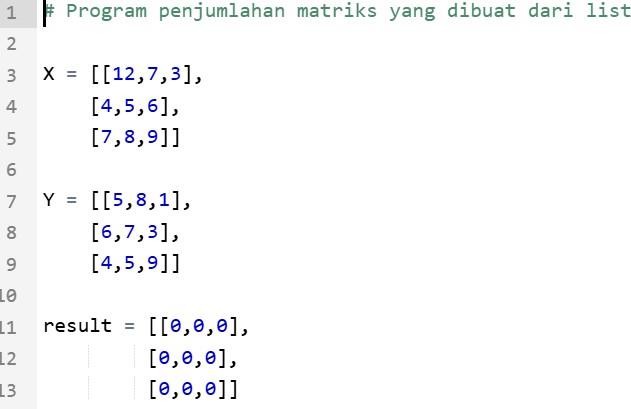
# Baris 2-5:

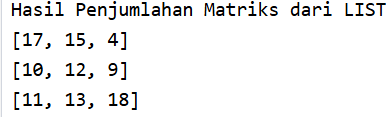
* membuat array 2 dimensi (disebut juga matriks) menggunakan np.array().
* Matriks ini memiliki bentuk (shape) 3 baris × 3 kolom.

# Baris 6:

* mencetak nilai pada baris ke-2 dan kolom ke-2 dari matriks.
* Karena indeks di Python dimulai dari 0, maka:
* matriks\_np[2] mengacu ke baris ketiga → [7, 8, 9]
* matriks\_np[2][2] mengacu ke elemen ketiga dalam baris tersebut → 9

***Praktek 13:***



**Output :**

**Penjelasannya:**

**Baris 1**

Komentar yang menjelaskan tujuan program, yaitu penjumlahan matriks dari list.

# Baris 2–4

Membuat matriks X berukuran 3x3 sebagai list bersarang (list of lists).

# Baris 5–7

Membuat matriks Y berukuran 3x3 dengan nilai berbeda dari X.

# Baris 8–10

Membuat matriks result berisi nol (3x3) yang akan diisi dengan hasil penjumlahan X + Y.

# Baris 11

Komentar yang menjelaskan proses penjumlahan

dilakukan dengan nested loop.

# Baris 12

Loop pertama: mengulang setiap baris (i) dari matriks.

# Baris 13

Loop kedua di dalamnya: mengulang setiap kolom (j) dari baris.

# Baris 14

Menjumlahkan elemen pada posisi [i][j] dari X dan Y, lalu disimpan ke result[i][j].

# Baris 15

Mencetak teks keterangan bahwa hasil penjumlahan akan ditampilkan.

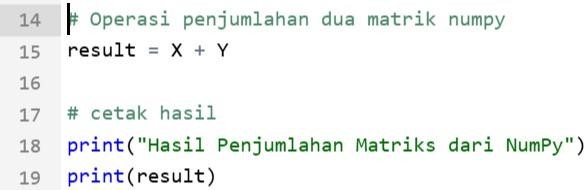
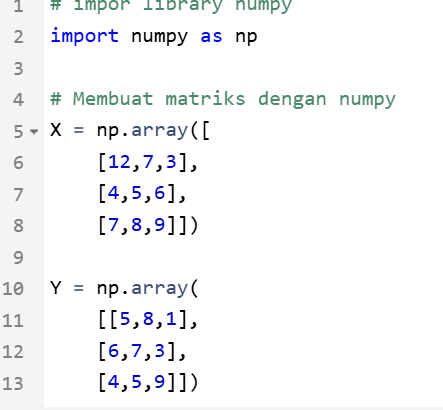
# Baris 16

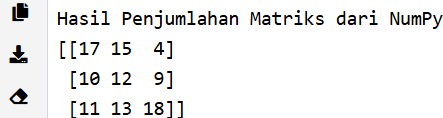
Loop untuk membaca setiap baris dalam matriks result.

# Baris 17

Mencetak setiap baris hasil penjumlahan.

***Praktek 14***



**Output:**

**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2–5

Membuat matriks X berukuran 3x3 menggunakan

# np.array.

**Baris 6–9**

Membuat matriks Y berukuran 3x3 menggunakan np.array.

# Baris 10

Menjumlahkan dua matriks (X + Y) dan menyimpan hasilnya di result.

# Baris 11

Komentar bahwa hasil akan dicetak.

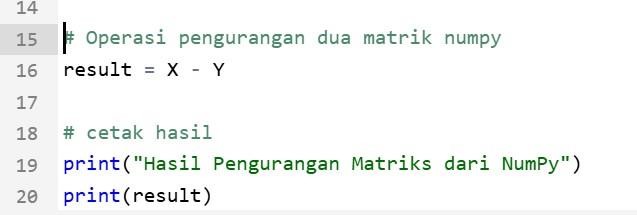
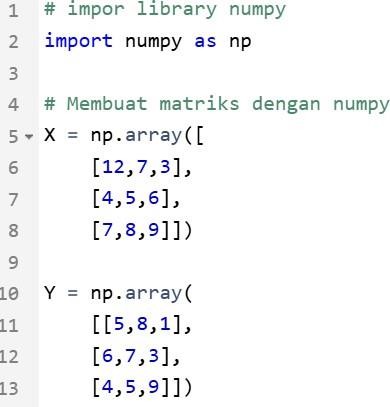
# Baris 12

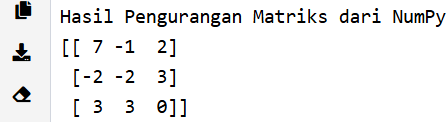
Mencetak keterangan "Hasil Penjumlahan Matriks dari NumPy".

# Baris 13

Mencetak isi dari result, yaitu hasil penjumlahan matriks X dan Y.

***Praktek 15***



**Output:**

**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2–5

Membuat matriks X 3x3 menggunakan np.array.

# Baris 6–9

Membuat matriks Y 3x3 menggunakan np.array.

# Baris 10

Melakukan operasi pengurangan elemen-elemen matriks: X - Y, hasil disimpan di result.

# Baris 11

Komentar untuk memberi tahu bahwa hasil akan dicetak.

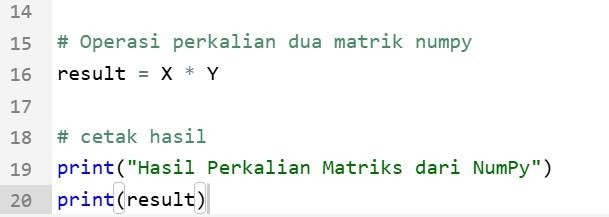
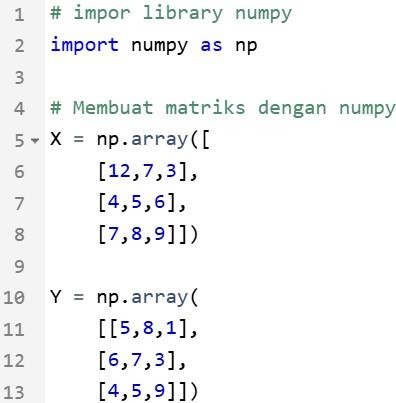
# Baris 12

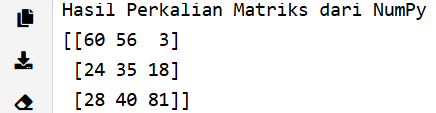
Mencetak teks: "Hasil Pengurangan Matriks dari NumPy".

# Baris 13

Mencetak isi result, yaitu hasil dari X - Y.

***Praktik 16:***



**Output:**

**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2–5

Membuat matriks X berukuran 3x3 menggunakan np.array.

# Baris 6–9

Membuat matriks Y berukuran 3x3 menggunakan np.array.

# Baris 10

Melakukan perkalian elemen per elemen (disebut juga

*element-wise multiplication*) antara X dan Y, hasil

disimpan di result.

# Baris 11

Komentar penjelas bahwa hasil akan dicetak.

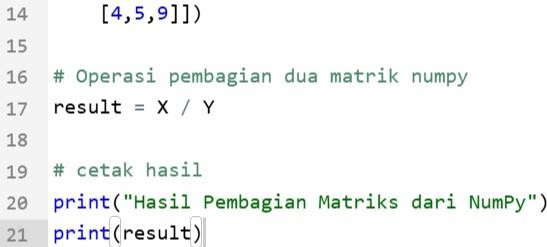
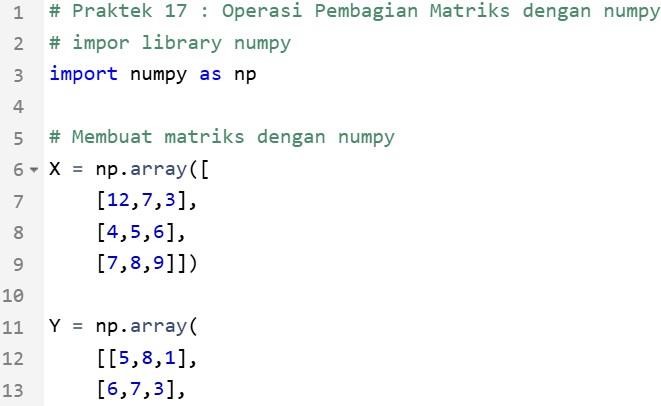
# Baris 12

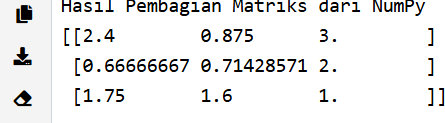
Mencetak teks keterangan: "Hasil Perkalian Matriks dari NumPy".

# Baris 13

Mencetak result, yaitu hasil dari X \* Y (bukan perkalian matriks biasa, tapi perkalian per elemen).

***Praktek 17***



**Output:**

**Penjelasan:**

**Baris 1**

Komentar: Menjelaskan bahwa ini adalah praktek operasi pembagian matriks dengan NumPy.

# Baris 2

Impor library NumPy sebagai np.

# Baris 3–6

Membuat matriks X berukuran 3x3 menggunakan np.array.

# Baris 7–10

Membuat matriks Y berukuran 3x3 menggunakan np.array.

# Baris 11

Melakukan pembagian elemen per elemen antara X dan Y menggunakan X / Y, hasil disimpan di result. **Baris 12**

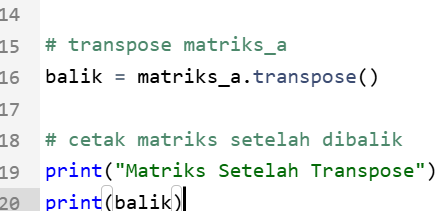
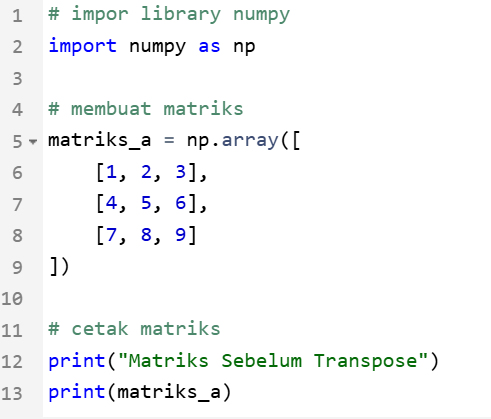
Komentar: Akan mencetak hasil pembagian**. Baris 13**

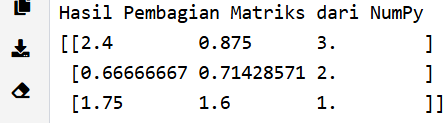
Mencetak teks: "Hasil Pembagian Matriks dari NumPy".

# Baris 14

Mencetak result, yaitu hasil pembagian elemen- elemen X dibagi Y.

**Praktek 18**



**Output:**

**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2–5

Membuat matriks matriks\_a berukuran 3x3 menggunakan np.array.

# Baris 6

Akan mencetak isi matriks sebelum transpose.

# Baris 7

Mencetak teks "Matriks Sebelum Transpose".

# Baris 8

Mencetak isi matriks\_a (sebelum di-transpose).

# Baris 9

Melakukan transpose (membalik baris jadi kolom) dengan matriks\_a.transpose() dan menyimpannya ke variabel balik.

# Baris 10

Komentar: Akan mencetak matriks setelah ditranspose.

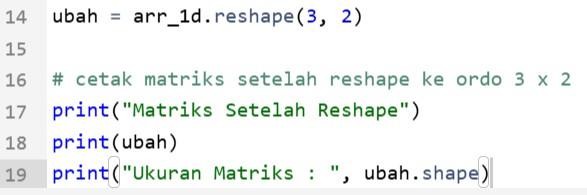
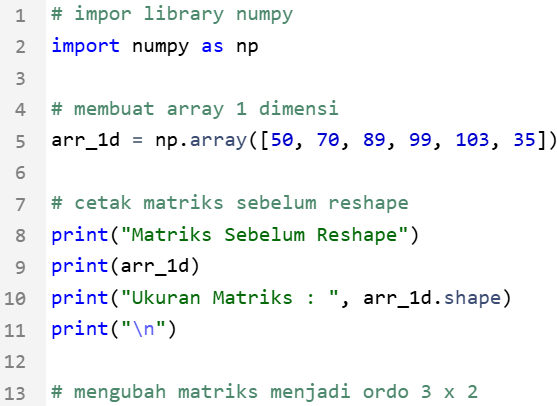
# Baris 11

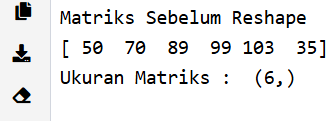
Mencetak teks "Matriks Setelah Transpose".

# Baris 12

Mencetak isi matriks balik, yaitu hasil dari transpose.

**Praktik 19**



**Output:**

**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2

Membuat array 1 dimensi arr\_1d berisi 6 elemen.

# Baris 3

Komentar: Akan mencetak array sebelum diubah bentuknya.

# Baris 4

Mencetak teks "Matriks Sebelum Reshape".

# Baris 5

Mencetak isi array arr\_1d.

# Baris 6

Mencetak ukuran array menggunakan .shape → hasilnya (6,).

# Baris 7

Mencetak baris kosong (newline) untuk pemisah tampilan.

# Baris 8

Melakukan reshape array 1 dimensi menjadi matriks berukuran 3 baris × 2 kolom, disimpan dalam ubah. **Baris 9**

Komentar: Akan mencetak hasil setelah reshape ke ordo 3×2.

# Baris 10

Mencetak teks "Matriks Setelah Reshape".

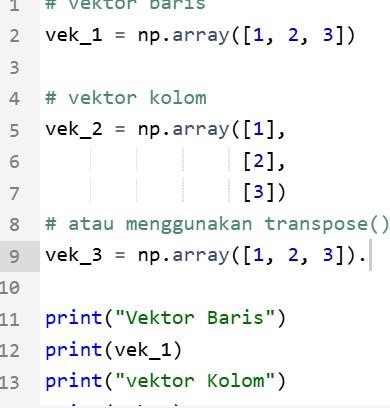
# Baris 11

Mencetak isi array setelah reshape (ubah).

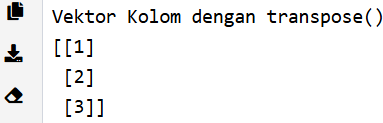
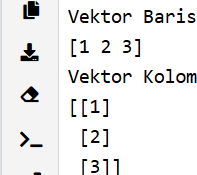
# Baris 12

Mencetak ukuran array hasil reshape dengan .shape, yaitu (3, 2).

***Praktek 20***



**Output:**



**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2

Membuat vektor baris vek\_1, yaitu array 1D [1, 2, 3].

# Baris 3–5

Membuat vektor kolom vek\_2 dengan bentuk 3×1 menggunakan list bersarang.

# Baris 6

Membuat vek\_3 dengan cara reshape dari array 1D menjadi bentuk (3,1) — vektor kolom juga.

# Baris 7

Mencetak teks "Vektor Baris".

# Baris 8

Mencetak isi vek\_1.

# Baris 9

Mencetak teks "Vektor Kolom".

# Baris 10

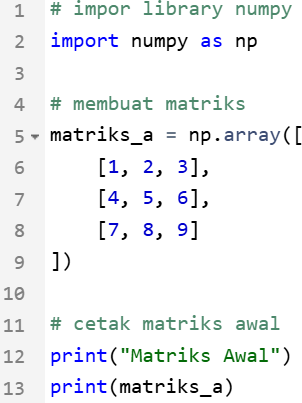
Mencetak isi vek\_2.

# Baris 11

Mencetak teks "Vektor Kolom dengan transpose()".

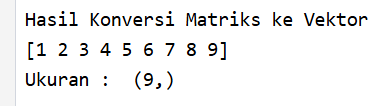
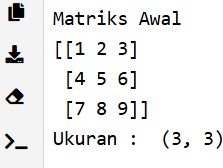
# Baris 12

Mencetak isi vek\_3.

***Praktek 21***



**Output:**



**Penjelasan:**

**Baris 1**

Impor library NumPy sebagai np.

# Baris 2–5

Membuat matriks matriks\_a berukuran 3×3 menggunakan np.array.

# Baris 6

Komentar bahwa matriks awal akan ditampilkan.

# Baris 7

Mencetak teks "Matriks Awal".

# Baris 8

Mencetak isi dari matriks\_a.

# Baris 9

Mencetak ukuran matriks menggunakan .shape, hasilnya (3, 3).

# Baris 10

Mencetak newline (\n) untuk pemisah visual di output.

# Baris 11

Mengubah matriks menjadi vektor 1 dimensi menggunakan .flatten(), disimpan ke variabel jd\_vektor.

# Baris 12

Komentar bahwa hasil konversi akan dicetak.

# Baris 13

Mencetak teks "Hasil Konversi Matriks ke Vektor".

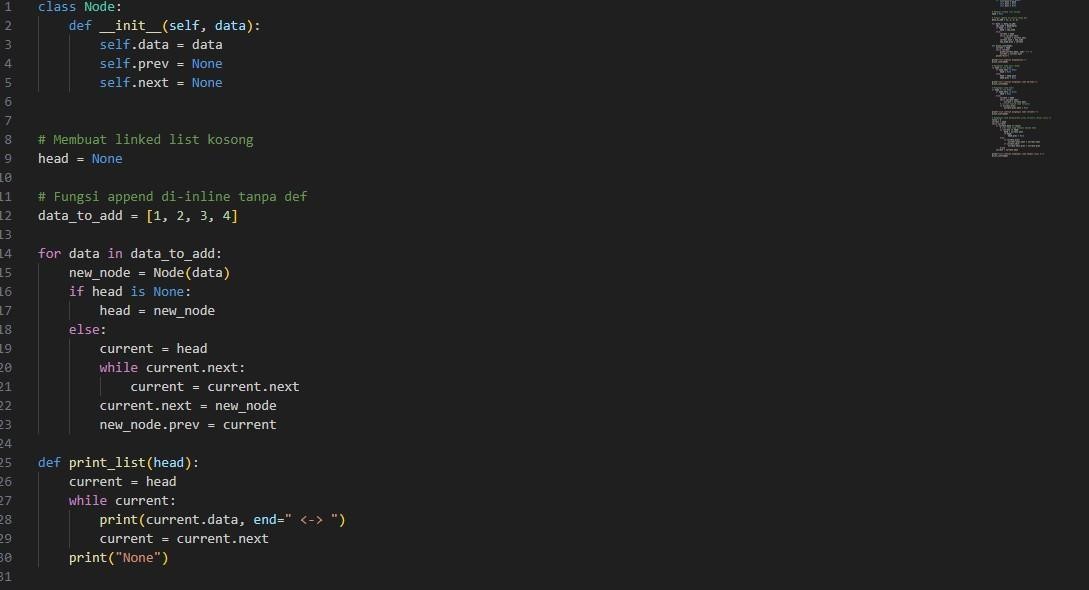
# Baris 14

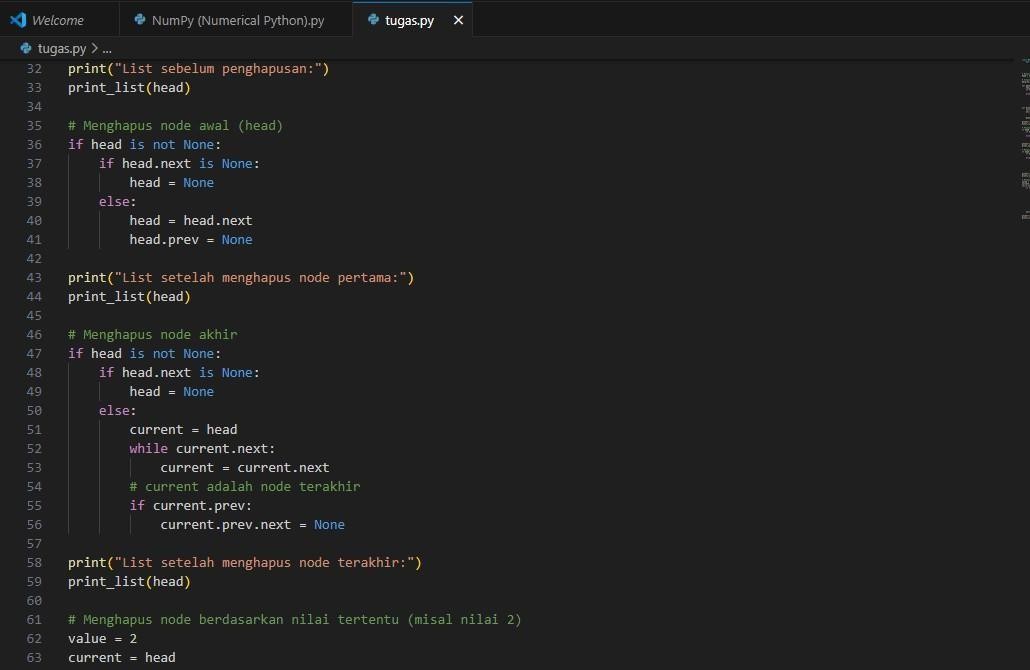
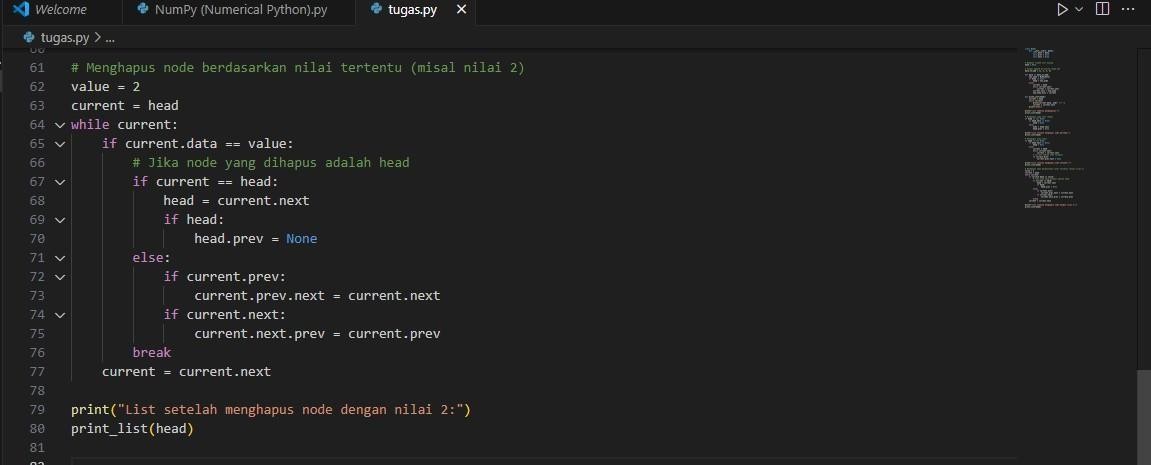
Mencetak isi jd\_vektor, hasil dari flatten().

# Baris 15

Mencetak ukuran jd\_vektor dengan .shape, hasilnya (9,) → berarti vektor 1 dimensi dengan 9 elemen.

***B. Tugas Modul 2:***

******



# Penjelasa setiap baris:

* 1. Class node: sebuah class Node yang akan digunakan untuk membuat node dalam linked list. def init (self, data): : Mendefinisikan method constructor untuk class Node, yang akan dipanggil ketika sebuah node dibuat.

self.data = data : Mengatur atribut data dari node dengan nilai yang diberikan. Atribut data ini digunakan untuk menyimpan nilai yang akan disimpan dalam node.

self.prev = None : Mengatur atribut prev dari node dengan nilai None. Atribut prev ini digunakan untuk menunjuk ke node sebelumnya dalam linked list.

self.next = None : Mengatur atribut next dari node dengan nilai None. Atribut next ini digunakan untuk menunjuk ke node berikutnya dalam linked list.

* 1. Membuat Linked List Kosong: - head = None : Membuat variabel head yang akan digunakan untuk menunjuk ke node pertama dalam linked list, dan mengaturnya dengan nilai None.
  2. Fungsi Append di inline: bekerja dengan cara yang sama seperti fungsi append biasa. membuat node baru dan menambahkannya ke akhir linked list.
     + data\_to\_add = [1, 2, 3, 4] : Membuat list data yang akan ditambahkan ke dalam linked list.
     + for data in data\_to\_add: : Melakukan loop untuk setiap data dalam list data\_to\_add.
     + new\_node = Node(data) : Membuat node baru dengan data yang diberikan.
     + if head is None: : Mengecek apakah linked list masih kosong (head adalah None).
     + head = new\_node : Jika linked list kosong, maka node baru menjadi head.
     + else: : Jika linked list tidak kosong, maka node baru akan ditambahkan ke akhir linked list.
     + current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
     + while current.next: : Melakukan loop sampai node terakhir dalam linked list.
     + current = current.next : Mengatur current ke node berikutnya.
     + current.next = new\_node : Mengatur next dari node terakhir ke node baru.
     + new\_node.prev = current : Mengatur prev dari node baru ke node terakhir.
  3. Menghapus node awal (head): proses menghapus node pertama dalam linked list. Ketika node awal dihapus, maka node berikutnya akan menjadi node awal yang baru.
     + if head is not None: : Mengecek apakah linked list tidak kosong.
     + if head.next is None: : Mengecek apakah linked list hanya memiliki satu node.
     + head = None : Jika linked list hanya memiliki satu node, maka head diatur ke None.
     + else: : Jika linked list memiliki lebih dari satu node.
     + head = head.next : Mengatur head ke node berikutnya.
     + head.prev = None : Mengatur prev dari node baru menjadi None.
  4. Fungsi print list: sebuah fungsi yang digunakan untuk mencetak isi dari linked list. Fungsi ini memungkinkan kita untuk melihat data yang ada dalam linked list.
     + def print\_list(head): : Mendefinisikan fungsi print\_list yang akan digunakan untuk mencetak linked list.
     + current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
     + while current: : Melakukan loop sampai node terakhir dalam linked list.
     + print(current.data, end=" <-> ") : Mencetak data dari node saat ini.
     + current = current.next : Mengatur current ke node berikutnya.
     + print("None") : Mencetak None untuk menandakan akhir linked list.
  5. Menghaous node akhir: proses menghapus node terakhir dalam linked list.

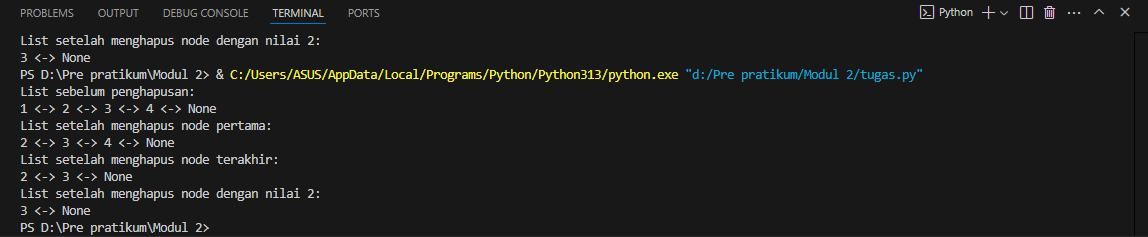
Ketika node akhir dihapus, maka node sebelumnya menjadi node terakhir yang baru.

* + - if head is not None: : Mengecek apakah linked list tidak kosong.
    - if head.next is None: : Mengecek apakah linked list hanya memiliki satu node.
    - head = None : Jika linked list hanya memiliki satu node, maka head diatur ke None.
    - else: : Jika linked list memiliki lebih dari satu node.
    - current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
    - while current.next: : Melakukan loop sampai node terakhir dalam linked list.
    - current = current.next : Mengatur current ke node berikutnya.
    - if current.prev: : Mengecek apakah node terakhir memiliki node sebelumnya.
    - current.prev.next = None : Mengatur next dari node sebelumnya menjadi None.
  1. Menghapus node berdasarkan nilai: proses menghapus node yang memiliki nilai tertentu dalam linked list.

Ketika node dengan nilai tertentu dihapus, maka node sebelumnya dan node berikutnya dihubungkan kembali.

* + - value = 2 : Membuat variabel value yang akan digunakan untuk mencari node dengan nilai tertentu.
    - current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
    - while current: : Melakukan loop sampai node terakhir dalam linked list.
    - if current.data == value: : Mengecek apakah data dari node saat ini sama dengan nilai yang dicari.
    - if current == head: : Mengecek apakah node yang dihapus adalah head.
    - head = current.next : Mengatur head ke node berikutnya.
    - if head: : Mengecek apakah head tidak None.
    - head.prev = None : Mengatur prev dari node baru menjadi None.
    - else: : Jika node yang dihapus bukan head.
    - if current.prev: : Mengecek apakah node yang dihapus memiliki node sebelumnya.
    - current.prev.next = current.next : Mengatur next dari node sebelumnya ke node berikutnya.

# output:

****

1. Penjelasa setiap baris:
   1. Class node: sebuah class Node yang akan digunakan untuk membuat node dalam linked list. def init (self, data): : Mendefinisikan method constructor untuk class Node, yang akan dipanggil ketika sebuah node dibuat.

self.data = data : Mengatur atribut data dari node dengan nilai yang diberikan. Atribut data ini digunakan untuk menyimpan nilai yang akan disimpan dalam node.

self.prev = None : Mengatur atribut prev dari node dengan nilai None. Atribut prev ini digunakan untuk menunjuk ke node sebelumnya dalam linked list.

self.next = None : Mengatur atribut next dari node dengan nilai None. Atribut next ini digunakan untuk menunjuk ke node berikutnya dalam linked list.

* 1. Membuat Linked List Kosong: - head = None : Membuat variabel head yang akan digunakan untuk menunjuk ke node pertama dalam linked list, dan mengaturnya dengan nilai None.
  2. Fungsi Append di inline: bekerja dengan cara yang sama seperti fungsi append biasa. membuat node baru dan menambahkannya ke akhir linked list.
     + data\_to\_add = [1, 2, 3, 4] : Membuat list data yang akan ditambahkan ke dalam linked list.
     + for data in data\_to\_add: : Melakukan loop untuk setiap data dalam list data\_to\_add.
     + new\_node = Node(data) : Membuat node baru dengan data yang diberikan.
     + if head is None: : Mengecek apakah linked list masih kosong (head adalah None).
     + head = new\_node : Jika linked list kosong, maka node baru menjadi head.
     + else: : Jika linked list tidak kosong, maka node baru akan ditambahkan ke akhir linked list.
     + current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
     + while current.next: : Melakukan loop sampai node terakhir dalam linked list.
     + current = current.next : Mengatur current ke node berikutnya.
     + current.next = new\_node : Mengatur next dari node terakhir ke node baru.
     + new\_node.prev = current : Mengatur prev dari node baru ke node terakhir.
  3. Menghapus node awal (head): proses menghapus node pertama dalam linked list. Ketika node awal dihapus, maka node berikutnya akan menjadi node awal yang baru.
     + if head is not None: : Mengecek apakah linked list tidak kosong.
     + if head.next is None: : Mengecek apakah linked list hanya memiliki satu node.
     + head = None : Jika linked list hanya memiliki satu node, maka head diatur ke None.
     + else: : Jika linked list memiliki lebih dari satu node.
     + head = head.next : Mengatur head ke node berikutnya.
     + head.prev = None : Mengatur prev dari node baru menjadi None.
  4. Fungsi print list: sebuah fungsi yang digunakan untuk mencetak isi dari linked list. Fungsi ini memungkinkan kita untuk melihat data yang ada dalam linked list.
     + def print\_list(head): : Mendefinisikan fungsi print\_list yang akan digunakan untuk mencetak linked list.
     + current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
     + while current: : Melakukan loop sampai node terakhir dalam linked list.
     + print(current.data, end=" <-> ") : Mencetak data dari node saat ini.
     + current = current.next : Mengatur current ke node berikutnya.
     + print("None") : Mencetak None untuk menandakan akhir linked list.
  5. Menghaous node akhir: proses menghapus node terakhir dalam linked list.

Ketika node akhir dihapus, maka node sebelumnya menjadi node terakhir yang baru.

* + - if head is not None: : Mengecek apakah linked list tidak kosong.
    - if head.next is None: : Mengecek apakah linked list hanya memiliki satu node.
    - head = None : Jika linked list hanya memiliki satu node, maka head diatur ke None.
    - else: : Jika linked list memiliki lebih dari satu node.
    - current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
    - while current.next: : Melakukan loop sampai node terakhir dalam linked list.
    - current = current.next : Mengatur current ke node berikutnya.
    - if current.prev: : Mengecek apakah node terakhir memiliki node sebelumnya.
    - current.prev.next = None : Mengatur next dari node sebelumnya menjadi None.
  1. Menghapus node berdasarkan nilai: proses menghapus node yang memiliki nilai tertentu dalam linked list.

Ketika node dengan nilai tertentu dihapus, maka node sebelumnya dan node berikutnya dihubungkan kembali.

* + - value = 2 : Membuat variabel value yang akan digunakan untuk mencari node dengan nilai tertentu.
    - current = head : Membuat variabel current yang akan digunakan untuk menunjuk ke node saat ini dalam linked list.
    - while current: : Melakukan loop sampai node terakhir dalam linked list.
    - if current.data == value: : Mengecek apakah data dari node saat ini sama dengan nilai yang dicari.
    - if current == head: : Mengecek apakah node yang dihapus adalah head.
    - head = current.next : Mengatur head ke node berikutnya.
    - if head: : Mengecek apakah head tidak None.
    - head.prev = None : Mengatur prev dari node baru menjadi None.
    - else: : Jika node yang dihapus bukan head.
    - if current.prev: : Mengecek apakah node yang dihapus memiliki node sebelumnya.
    - current.prev.next = current.next : Mengatur next dari node sebelumnya ke node berikutnya.

1. Penjelasan output:
   1. List sebelum penghapusan: 1 <-> 2 <-> 3 <-> 4 <-> None:

Ini adalah kondisi awal linked list sebelum penghapusan. Linked list memiliki 4 node dengan nilai 1, 2, 3, dan 4.

Node terakhir memiliki next yang bernilai None, menandakan akhir linked list.

* 1. List setelah menghapus node pertama: 2 <-> 3 <-> 4 <-> None:

Node pertama dengan nilai 1 telah dihapus dari linked list. Node kedua dengan nilai 2 menjadi node pertama yang baru. Linked list sekarang memiliki 3 node dengan nilai 2, 3, dan 4.

* 1. List setelah menghapus node terakhir: 2 <-> 3 <-> None

Node terakhir dengan nilai 4 telah dihapus dari linked list. Node dengan nilai 3 menjadi node terakhir yang baru.

Linked list sekarang memiliki 2 node dengan nilai 2 dan 3.

* 1. List setelah menghapus node dengan nilai 2: 3 <-> None: Node dengan nilai 2 telah dihapus dari linked list.

Linked list sekarang hanya memiliki 1 node dengan nilai 3.

Node dengan nilai 3 menjadi node pertama dan terakhir dalam linked list.