

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN ALGORITMA
ARRAY MENGGUNAKAN PYTHON**



Disusun Oleh :

Nama : Muhammad Nuh
NIM : 220504027
Unit : 01
Prodi : Informatika
Mata Kuliah : Struktur Data dan Algoritma
Dosen Pengampu : NURUL FADILLAH, S. ST., M.T

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SAMUDRA
2023**



**KEMENTERIAN PENDIDIKAN KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS SAMUDRA
FAKULTAS TEKNIK**

Jalan Prof. Dr. Syarif Thayeh, Meurandeh, Langsa-Aceh

Telp (0641) 426534, Fax (0641) 426535, 7445155

Laman: <http://www.unsam.ac.id> Email: info@unsa.ac.id Kode Pos 24416

Nama : Muhammad Nuh

NIM : 220504027

Laporan Studi : Informatika

Laporan Praktikum : Struktur Data dan Alogoritma, Array menggunakan Python

Isi dan format laporan ini telah disetujui serta disahkan sebagai syarat untuk melengkapi pelaksanaan praktikum Struktur Data Percabangan (If-Else, Switch-Case, dan Do-While) Menggunakan Python pada Program Studi Informatika, Fakultas Teknik, Universitas Samudra, Pada :

Hari : Senin

Tanggal : 30 Oktober 2023

Nilai :

Mengetahui

Mengetahui

Pembimbing praktikum I

Pembimbing Praktikum II

NURUL FADILLAH, S.ST., M.T.

DEFY SYAFIRA, S.T. , M.Kom.

NIP. 198910012022032009

KATA PENGANTAR

Puji syukur atas kehadirat Allah Swt. atas segala rahmat-Nya sehingga penyusun dapat membuat laporan ini dapat tersusun sampai selesai. Tidak lupa pula shalawat dan salam atas kehadirat Nabi besar Muhammad saw yang telah membawa kita dari alam jahiliah ke alam yang penuh ilmu pengetahuan seperti yang kita rasakan saat ini.

Dengan terselesainya laporan ini, maka tidak lupa saya mengucapkan banyak terima kasih kepada semua pihak yang terlibat dalam penyusunan laporan ini, khususnya kepada Ibu Nurul Fadillah, S.ST.,M.T. selaku Dosen Pengampu.

Saya sangat berharap semoga laporan ini dapat menambah pengetahuan dan pengalaman bagi pembaca. Bahkan saya berharap lebih jauh lagi agar malah ini bisa pembaca praktikkan juga. Bagi saya sebagai penyusun mersa bahwa masih banyak kekurangan dalam penyusunan laporan ini karena keterbatasan pengetahuan dan pengalaman saya. Untuk itu saya sangat mengharapkan kritik dan saran yang membangun dari pembca demi kesempurnaan laporan ini

Langsa, 29 Oktober 2023

Muhammad Nuh

NIM. 220504027

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	1
1.3 Tujuan Laporan	2
BAB II DASAR TEORI.....	3
2.1 Kajian Pustaka	3
BAB III METODOLOGI PRAKTIKUM.....	4
3.1 Algoritma Program 1	4
3.2 Flowchart Program 1	4
3.3 Algoritma Program 2	5
3.4 Flowchart Program 2	6
3.5 Algoritma Program 3	7
3.6 Flowchart Program 3	8
3.7 Algoritma Program 4	9
3.8 Flowchart Program 4	9
3.9 Algoritma Program 5	10
3.10 Flowchart Program 5	11
3.11 Alat dan Bahan	12
BAB IV HASIL PRAKTIKUM DAN ANALISA	13
4.1 Analisis Program 1.....	13
4.2 Analisis Program 2	15
4.3 Analisis Program 3.....	17
4.4 Analisis Program 4.....	19
4.5 Analisis Program 5.....	20

BAB V PENUTUP	23
5.1 Kesimpulan.....	23
5.2 Saran.....	23
LAMPIRAN DOKUMENTASI	24
DAFTAR PUSTAKA	25

DAFTAR GAMBAR

Gambar 4.1 Program 1	13
Gambar 4.2 Output Program 1	14
Gambar 4.3 Program 2	15
Gambar 4.4 Output Program 2	16
Gambar 4.5 Program 3	17
Gambar 4.6 Output Program 3	18
Gambar 4.7 Program 4	19
Gambar 4.8 Output Program 4	20
Gambar 4.9 Program 5	21
Gambar 4.10 Output Program 5	22
Gambar 4.11 Laporan Dokumentasi	24

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penggunaan array dalam pemrograman sangat penting untuk menyimpan dan mengelola kumpulan data. Dalam bahasa pemrograman Python, NumPy (Numerical Python) adalah salah satu pustaka yang menyediakan dukungan untuk array multidimensional. Array NumPy memungkinkan kita untuk melakukan operasi matematika yang efisien pada data numerik, membuatnya menjadi pilihan utama untuk analisis data dan komputasi ilmiah..

Array, sebagai struktur data yang memungkinkan penyimpanan sejumlah elemen data dengan jenis yang sama, menjadi kunci dalam pengelolaan data yang besar dan kompleks. Python, sebagai bahasa pemrograman yang sangat populer, menawarkan dukungan bawaan untuk array melalui struktur data list. Namun, untuk tugas-tugas yang memerlukan kinerja tinggi dan manipulasi array yang efisien, penggunaan NumPy, pustaka array numerik Python, menjadi pilihan yang lebih unggul.

Pustaka NumPy dirancang khusus untuk menangani array multidimensional dengan cepat dan efisien. Dengan dukungan untuk operasi matematika vektorisasi, indeks canggih, dan fungsi aljabar linear, NumPy mampu memberikan kinerja tinggi dalam pengolahan data numerik.

1.2 Rumusan Masalah

Laporan ini akan mencari jawaban untuk beberapa pertanyaan pokok. Pertama, mengapa array penting dalam pemrograman Python? Kedua, apa kelebihan NumPy dalam bekerja dengan array dibandingkan dengan cara lain? Selain itu, bagaimana cara kita melakukan operasi dasar pada array dengan NumPy? Dengan menjawab pertanyaan-pertanyaan ini, laporan ini diharapkan memberikan gambaran yang sederhana dan mudah dipahami tentang array dan penggunaannya dalam python.

1.3 Tujuan Laporan

Laporan ini bertujuan untuk memberikan pemahaman yang jelas tentang penggunaan array dalam pemrograman Python menggunakan pustaka NumPy. Dengan tujuan utama menjelaskan konsep dasar array dan keunggulan penggunaan NumPy.,

BAB II

DASAR TEORI

2.1 Kajian Pustaka

Array adalah struktur data yang bisa menampung banyak nilai dalam satu waktu. Kita bisa melakukan loop atau pengulangan melalui elemen pada array secara mudah dan mengambil nilai yang diharapkan hanya dengan menentukan nilai indeks. Array pun dapat berubah, karena itu, kita bisa melakukan aneka macam manipulasi berdasarkan kebutuhan kita.

1. List

List atau daftar adalah sebuah perpaduan atau koleksi data yang berisi nomor dan juga teks. List dituliskan di antara tanda kurung siku “[.....]” dan setiap data pada list dipisahkan dengan koma, contoh lis1=[3, 2, 1, “saya”]. List1 sendiri adalah sebuah variabel list tersebut, dan kita memiliki 4 buah data didalamnya.

Di dalam list setiap data mempunyai alamat yang disebut sebagai indeks. Indeks itu sendiri di hitung dari arah kiri menuju ke kanan dan dimulai dengan 0.

2. Tuple

Tuple sebenarnya sama saja dengan list, tetapi yang membedakan merupakan bahwa Tuple value-nilai nya tidak berubah. Cara mengakses valuenya sama halnya dengan kita menggunakan List. Dan juga tuple menggunakan “(.....)”.

3. Dictionary Python

Dictionary Python berbeda dengan List ataupun Tuple. Lantaran setiap urutannya berisi key dan value. Setiap key dipisah dari valuenya dengan titik dua (:), item dipisahkan dengan koma, dan semuanya berada di dalam kurung kurawal {.....}. Dictionary kosong tanpa barang ditulis hanya menggunakan dua kurung kurawal {}.

NumPy adalah pustaka sumber terbuka yang dikembangkan komunitas, yang menyediakan objek array Python multidimensi bersama dengan fungsi-fungsi array-aware yang beroperasi di dalamnya. Karena kesederhanaannya, array NumPy adalah format pertukaran de facto untuk data array dengan Python..

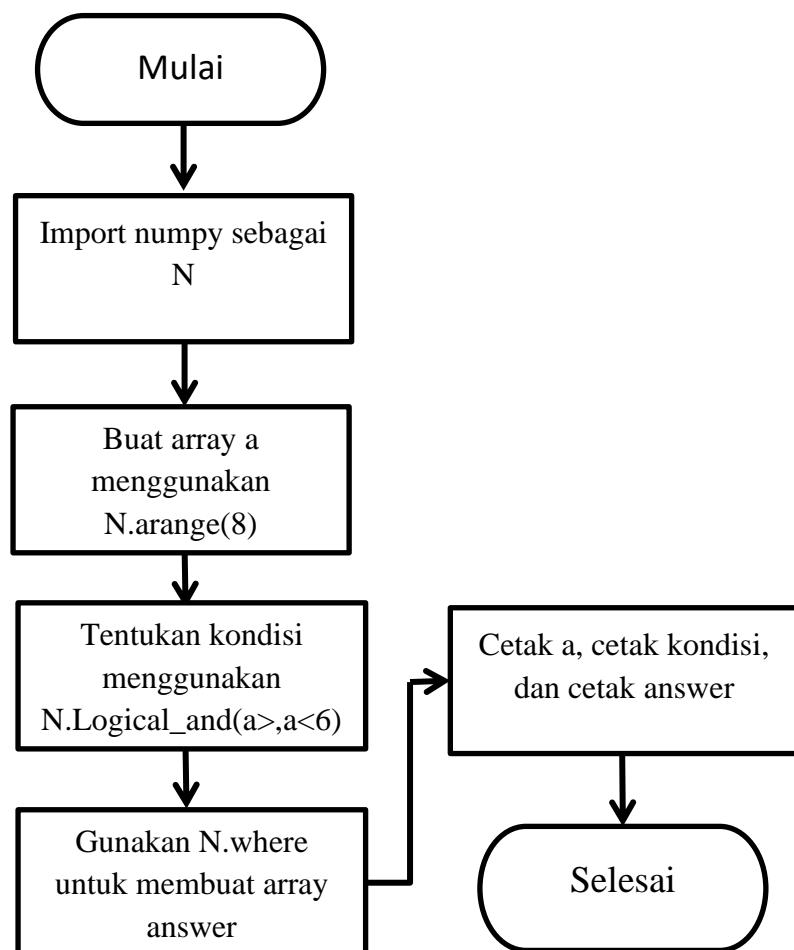
BAB III

METODOLOGI PENELITIAN

3.1 Algoritma Program 1

1. Mulai.
2. Import NumPy sebagai N
3. Buat array a menggunakan N.arange(8)
4. Tentukan kondisi condition menggunakan N.logical_and(a > 3, a < 6)
5. Gunakan N.where untuk membuat array answer dengan aturan: jika condition True, gunakan a*2, jika False, gunakan 0
6. Cetak array a, array condition, dan array answer.
7. Selesai.

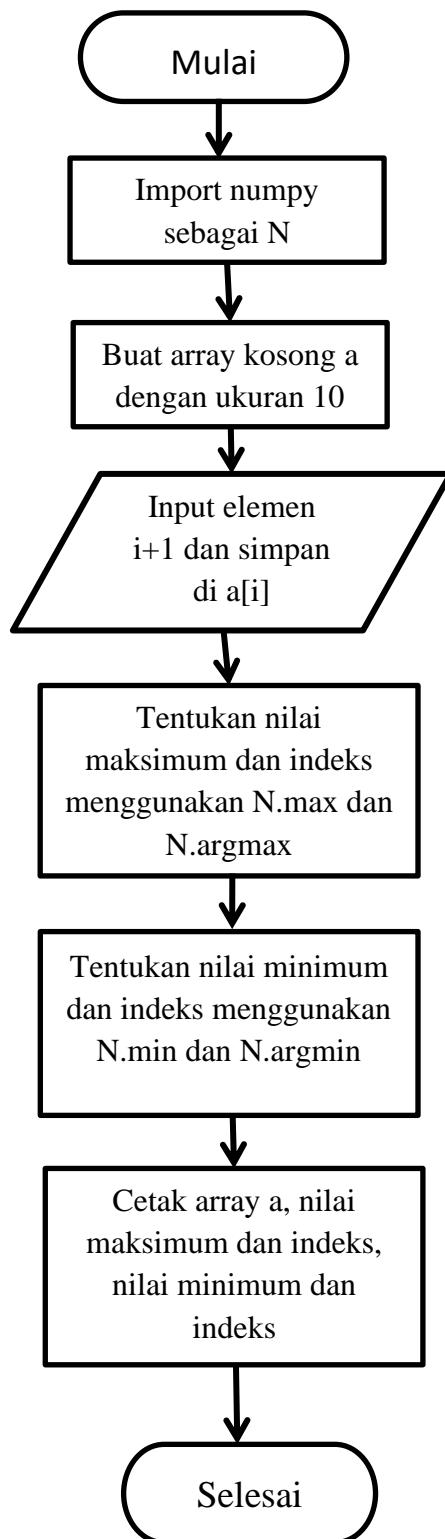
3.2 Flowchart Program 1



3.3 Algoritma Program 2

1. Mulai.
2. Import NumPy sebagai N.
3. Buat array kosong a dengan panjang 10 dan tipe data integer menggunakan N.empty(10, dtype=int).
4. Gunakan loop for untuk mengisi setiap elemen array a melalui input pengguna.
 - a. Iterasi sebanyak 10 kali.
 - b. Pada setiap iterasi, minta pengguna memasukkan nilai untuk elemen ke-(i+1).
 - c. Simpan nilai yang dimasukkan ke dalam array a[i].
5. Gunakan fungsi N.max dan N.argmax untuk menemukan nilai maksimum dan indeks nilai maksimum dalam array a. Simpan hasilnya dalam variabel max_value dan max_index.
6. Gunakan fungsi N.min dan N.argmin untuk menemukan nilai minimum dan indeks nilai minimum dalam array a. Simpan hasilnya dalam variabel min_value dan min_index.
7. Cetak array a, nilai maksimum beserta posisinya, dan nilai minimum beserta posisinya menggunakan pernyataan cetak.
8. Selesai.

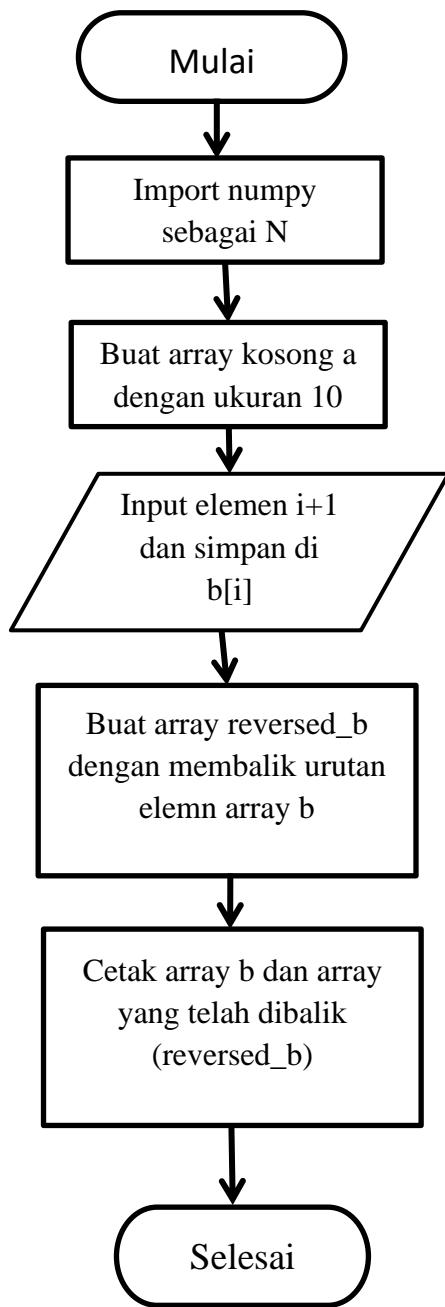
3.4 Flowchart Program 2



3.5 Algoritma Program 3

1. Mulai.
2. Import NumPy sebagai np.
3. Buat array kosong b dengan panjang 10 dan tipe data integer menggunakan `np.empty(10, dtype=int)`.
4. Gunakan loop for untuk mengisi setiap elemen array b melalui input pengguna.
 - a. Iterasi sebanyak 10 kali.
 - b. Pada setiap iterasi, minta pengguna memasukkan nilai untuk elemen ke-($i+1$).
 - c. Simpan nilai yang dimasukkan ke dalam array `b[i]`.
5. Gunakan slicing `[::-1]` untuk membuat array `reversed_b` yang merupakan hasil dari membalikkan urutan elemen array `b`.
6. Cetak array `b` dan array yang telah dibalik (`reversed_b`).
7. Selesai.

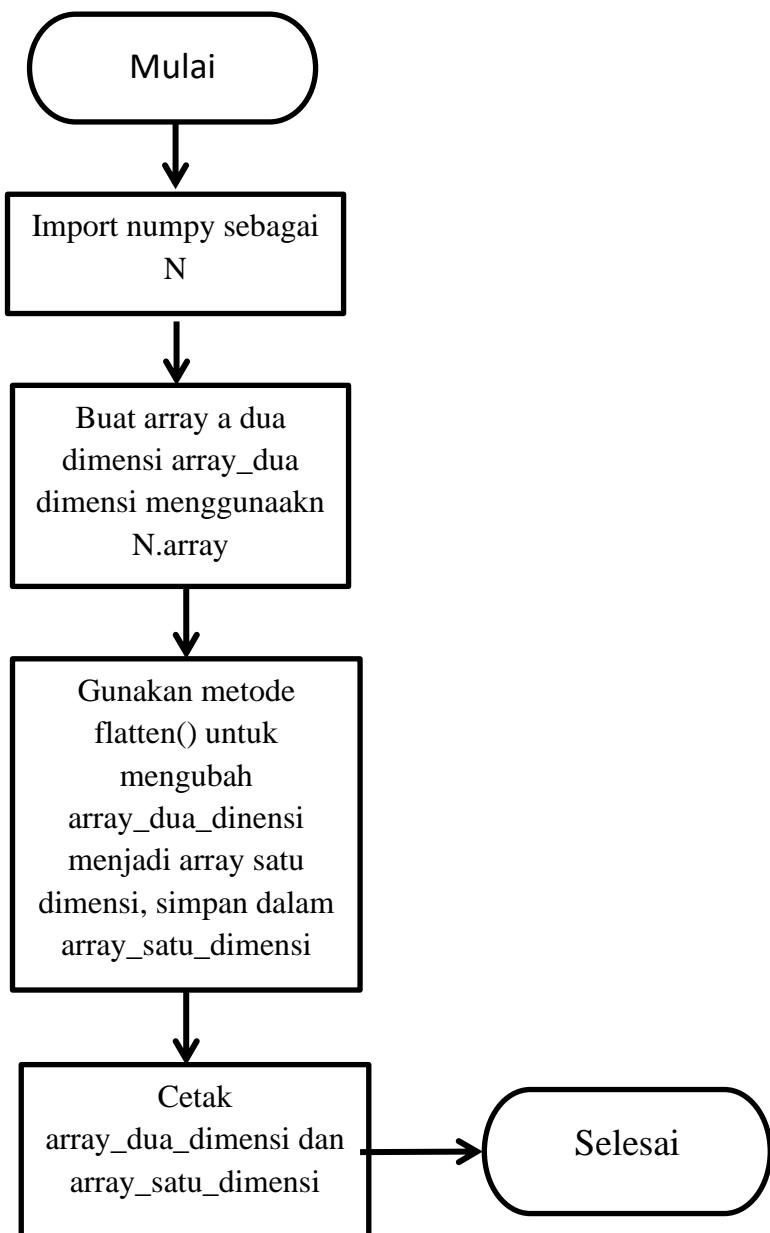
3.6 Flowchart Program 3



3.7 Algoritma Program 4

1. Mulai
2. Import NumPy sebagai N.
3. Buat array dua dimensi array_dua_dimensi menggunakan N.array.
4. Gunakan metode flatten() untuk mengubah array_dua_dimensi menjadi array satu dimensi, simpan dalam variabel array_satu_dimensi.
5. Cetak array_dua_dimensi dan array_satu_dimensi.
6. Selesai.

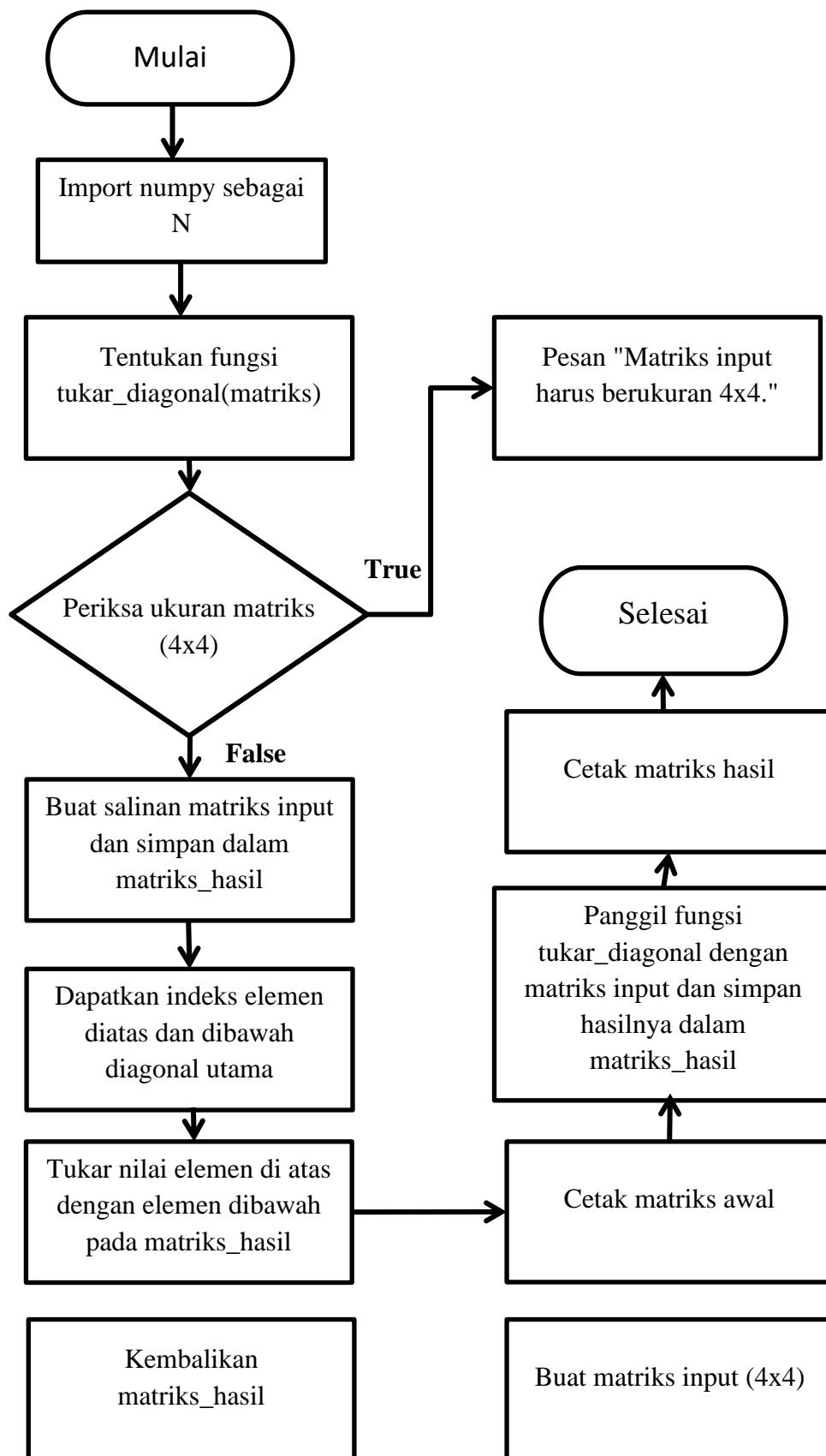
3.8 Flowchart Program 4



3.9 Algoritma Tugas Program

1. Mulai
2. Import NumPy sebagai N.
3. Tentukan fungsi tukar_diagonal(matriks).
4. Periksa apakah matriks memiliki ukuran 4x4.
5. Buat salinan matriks input dan simpan dalam variabel matriks_hasil.
6. Gunakan N.triu_indices dan N.tril_indices untuk mendapatkan indeks elemen di atas dan di bawah diagonal utama.
7. Tukar nilai elemen di atas dengan nilai elemen di bawah pada matriks_hasil.
8. Kembalikan matriks_hasil.
9. Buat matriks input sebagai contoh dengan ukuran 4x4.
10. Cetak matriks awal.
11. Panggil fungsi tukar_diagonal dengan matriks input dan simpan hasilnya dalam variabel matriks_hasil.
12. Cetak matriks hasil.
13. Selesai

3.10 Flowchart Tugas Program



3.11 Alat dan Bahan

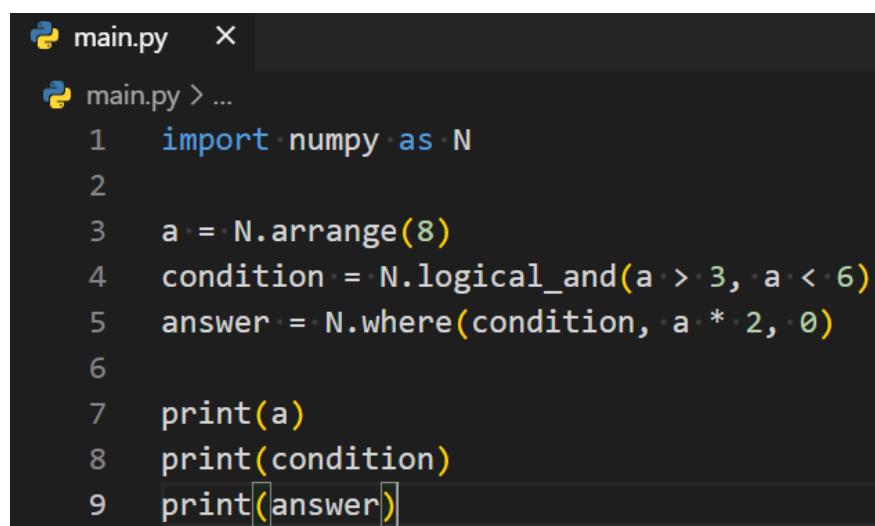
1. Komputer: Diperlukan sebuah komputer (laptop atau desktop) dengan sistemoperasi seperti Windows, macOS, atau Linux.
2. Text editor atau Integrated Development Environment (IDE) yang mendukung penulisan dan eksekusi kode python.
3. Koneksi internet jika ada kebutuhan untuk mengakses referensi program
4. Contoh-contoh kasus atau scenario yang akan digunakan sebagai dasar praktikum.

BAB IV

HASIL PRAKTIKUM DAN ANALISA

4.1 Analisis Program 1

Program ini adalah implementasi dalam Python menggunakan NumPy untuk membuat array dengan 8 elemen menggunakan N.arange, kemudian membuat kondisi dengan N.logical_and untuk memeriksa elemen yang berada di antara 3 dan 6. Selanjutnya, fungsi N.where digunakan untuk menggandakan nilai elemen yang memenuhi kondisi, sementara elemen yang tidak memenuhi kondisi diatur menjadi 0. Hasilnya dicetak untuk menampilkan array asli (a), kondisi, dan array yang dihasilkan setelah penerapan kondisi.



```
main.py > ...
1 import numpy as N
2
3 a = N.arange(8)
4 condition = N.logical_and(a > 3, a < 6)
5 answer = N.where(condition, a * 2, 0)
6
7 print(a)
8 print(condition)
9 print(answer)
```

Gambar 4.1 Program 1

Analisis:

1. import numpy as N: Mengimpor pustaka NumPy dan menetapkan alias N.
2. a = N.arange(8): Membuat array a dengan nilai 0 hingga 7 menggunakan fungsi arange.
3. condition = N.logical_and(a > 3, a < 6): Membuat array condition yang berisi nilai True jika elemen a lebih besar dari 3 dan kurang dari 6, dan False untuk elemen lainnya.

4. `answer = N.where(condition, a*2, 0)`: Menggunakan fungsi `where` untuk membuat array baru (`answer`). Jika kondisi terpenuhi, nilai diambil dari `a*2`, jika tidak, diambil nilai 0.
5. `print(a)`: Mencetak array `a`.
6. `print(condition)`: Mencetak array `condition`.
7. `print(answer)`: Mencetak array `answer`.

Hasil yang dicetak adalah nilai array `a`, kondisi `condition`, dan hasil dari `N.where`. Program ini menciptakan array baru (`answer`) dengan menggandakan nilai elemen yang memenuhi kondisi, dan mengganti nilai elemen yang tidak memenuhi kondisi dengan 0.

Contoh Eksekusi:

```
D:\codingan\UNSAM\SEMESTER 3\mk-struktur-data-dan-algoritma\array>py main.py
[0 1 2 3 4 5 6 7]
[False False False False True True False False]
[ 0  0  0  0  8 10  0  0]
```

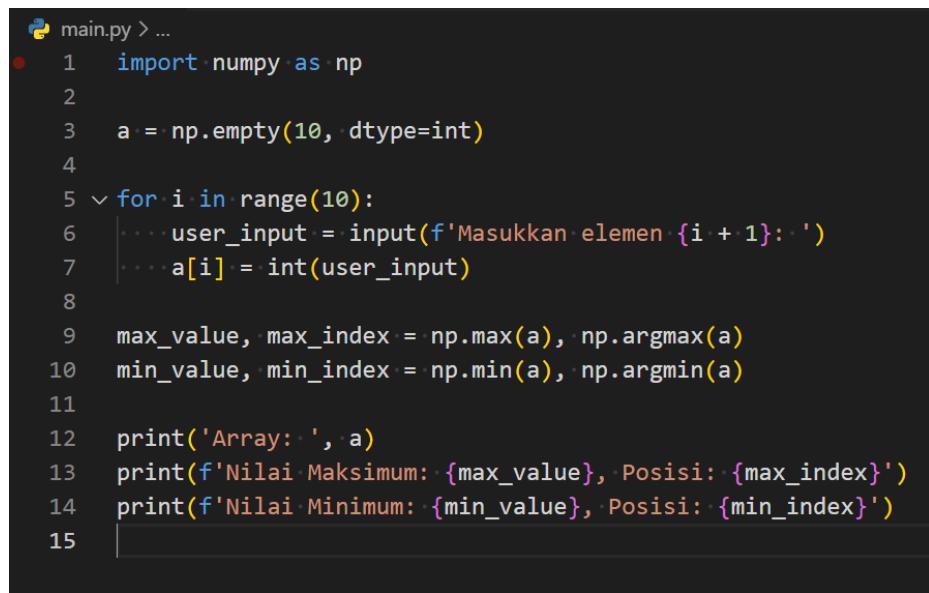
Gambar 4.2 Output program 1

Analisis Output:

1. `[0 1 2 3 4 5 6 7]`: Ini adalah array `a` yang dibuat dengan `N.arange(8)`, sehingga berisi nilai dari 0 hingga 7.
2. `[False False False False True True False False]`: Ini adalah array `condition` yang berisi nilai boolean (True atau False) untuk setiap elemen `a`. Nilai True menunjukkan bahwa elemen tersebut memenuhi kondisi ($a > 3$ dan $a < 6$), sedangkan nilai False menunjukkan sebaliknya.
3. `[0 0 0 0 8 10 0 0]`: Ini adalah array `answer`, yang dihasilkan menggunakan `N.where`. Nilai-nilai diambil dari `a*2` untuk elemen-elemen yang memenuhi kondisi, dan diambil nilai 0 untuk elemen-elemen yang tidak memenuhi kondisi. Jadi, elemen keempat dan kelima dari `answer` memiliki nilai 0, sementara elemen keenam dan ketujuh memiliki nilai yang dihasilkan dari `a*2`.

4.2 Analisis Program 2

Program ini adalah implementasi dalam Python menggunakan NumPy untuk membuat array dengan 10 elemen yang diisi oleh pengguna melalui input, dan kemudian mencari nilai maksimum, indeks maksimum, nilai minimum, dan indeks minimum dari array tersebut. Dalam konteks ini, array a dibuat menggunakan N.empty dengan tipe data integer, dan setiap elemennya diisi melalui loop for yang meminta input dari pengguna. Setelah itu, fungsi-fungsi NumPy seperti N.max, N.argmax, N.min, dan N.argmin digunakan untuk mencari nilai dan indeks maksimum serta minimum dari array. Hasilnya dicetak untuk menampilkan array, nilai maksimum, posisi maksimum, nilai minimum, dan posisi minimum.



```
main.py > ...
1 import numpy as np
2
3 a = np.empty(10, dtype=int)
4
5 for i in range(10):
6     user_input = input(f'Masukkan elemen {i + 1}: ')
7     a[i] = int(user_input)
8
9 max_value, max_index = np.max(a), np.argmax(a)
10 min_value, min_index = np.min(a), np.argmin(a)
11
12 print('Array:', a)
13 print(f'Nilai Maksimum: {max_value}, Posisi: {max_index}')
14 print(f'Nilai Minimum: {min_value}, Posisi: {min_index}')
15
```

Gambar 4.3 Program 2

Analisis:

1. import numpy as N: Mengimpor pustaka NumPy dan menetapkan alias N.
2. a = N.empty(10, dtype=int): Membuat array a dengan ukuran 10 dan tipe data integer, namun masih kosong (belum diisi).
3. for i in range(10): a[i] = int(input(f"Masukkan elemen {i + 1}: ")): Mengisi array a dengan input dari pengguna sebanyak 10 kali.

4. `max_value, max_index = N.max(a), N.argmax(a)`: Menggunakan fungsi `N.max` untuk mendapatkan nilai maksimum dari array `a` dan `N.argmax` untuk mendapatkan indeks/posisi nilai maksimum.
5. `min_value, min_index = N.min(a), N.argmin(a)`: Menggunakan fungsi `N.min` untuk mendapatkan nilai minimum dari array `a` dan `N.argmin` untuk mendapatkan indeks/posisi nilai minimum.
6. `print("Array:", a)`: Mencetak array `a`.
7. `print(f"Nilai maksimum: {max_value}, posisi: {max_index}")`: Mencetak nilai maksimum dan posisinya.
8. `print(f"Nilai minimum: {min_value}, posisi: {min_index}")`: Mencetak nilai minimum dan posisinya.

Contoh Eksekusi:

```
D:\codingan\UNSAM\SEMESTER 3\mk-struktur-data-dan-algoritma\array>py main.py
Masukkan elemen 1: 1
Masukkan elemen 2: 3
Masukkan elemen 3: 4
Masukkan elemen 4: 2
Masukkan elemen 5: 5
Masukkan elemen 6: 4
Masukkan elemen 7: 6
Masukkan elemen 8: 8
Masukkan elemen 9: 6
Masukkan elemen 10: 11
Array: [ 1  3  4  2  5  4  6  8  6 11]
Nilai Maksimum: 11, Posisi: 9
Nilai Minimum: 1, Posisi: 0
```

Gambar 4.4 Output Program 2

Analisis Output:

1. Meminta user untuk memasukkan 10 elemen, dan nilai-nilai tersebut disimpan dalam ‘`a`’.
2. `Array: [4 6 3 2 3 4 5 6 9 7]`: Mencetak array `a` yang telah diisi dengan nilai dari user.
3. `Nilai maksimum: 9, posisi: 8`: Menggunakan fungsi `N.max` dan `N.argmax`, program menemukan nilai maksimum dari array ‘`a`’ (yaitu 9) dan posisinya (indeks 8).

- Nilai maksimum: 2, posisi: 3: Menggunakan fungsi N.min dan N.argmin, program menemukan nilai minimum dari array ‘a’ (yaitu 2) dan posisinya (indeks 3).

4.3 Analisis Program 3

Program ini adalah implementasi dalam Python menggunakan NumPy untuk membuat array dengan 10 elemen yang diisi oleh pengguna melalui input, dan kemudian membalikkan urutan elemen-elemen array tersebut. Dalam konteks ini, array awal b dibuat menggunakan N.empty dengan tipe data integer, dan setiap elemennya diisi melalui loop for yang meminta input dari pengguna. Setelah itu, array tersebut dibalikkan urutannya menggunakan slicing dengan langkah -1 (b[::-1]). Hasilnya dicetak untuk menampilkan array asli dan array yang telah dibalik.



```
in.py  x
ain.py > ...
import numpy as N

b = N.empty(10, dtype=int)
for i in range(10):
    b[i] = int(input(f'Masukkan elemen {i + 1}: '))

reversed_b = b[::-1]

print('Array asli:', b)
print('Array yang telah dibalik:', reversed_b)
```

Gambar 4.5 Program 3

Analisis:

1. import numpy as N: Mengimpor pustaka NumPy dan menetapkan alias N.
2. b = N.empty(10, dtype=int): Membuat array b dengan ukuran 10 dan tipe data integer, namun masih kosong (belum diisi).
3. for i in range(10): b[i] = int(input(f"Masukkan elemen {i + 1}: ")): Mengisi array b dengan input dari pengguna sebanyak 10 kali.
4. reversed_b = b[::-1]: Membuat array reversed_b yang merupakan salinan dari array b dengan elemen-elemen yang dibalik urutannya menggunakan slicing [::-1].
5. print("Array asli:", b): Mencetak array b.
6. print("Array yang telah dibalik:", reversed_b): Mencetak array reversed_b.

Contoh Eksekusi:

```
Masukkan elemen 1: 3
Masukkan elemen 2: 2
Masukkan elemen 3: 4
Masukkan elemen 4: 1
Masukkan elemen 5: 2
Masukkan elemen 6: 4
Masukkan elemen 7: 6
Masukkan elemen 8: 6
Masukkan elemen 9: 6
Masukkan elemen 10: 4
Array asli: [3 2 4 1 2 4 6 6 6 4]
Array yang telah dibalik: [4 6 6 6 4 2 1 4 2 3]
```

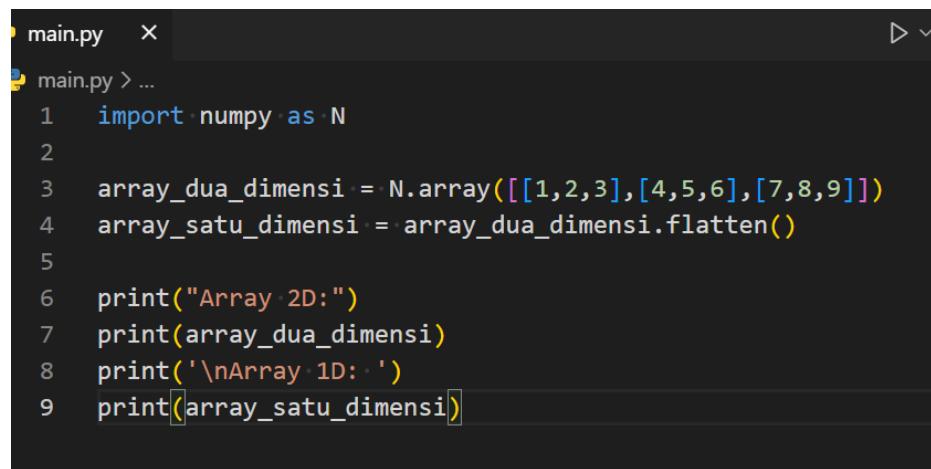
Gambar 4.6 Output Program 3

Analisis Output:

1. User diminta untuk memasukkan 10 elemen, dan nilai-nilai tersebut disimpan dalam array 'b'.
2. Array asli: [3 2 4 1 2 4 6 6 6 4]: Mencetak array 'b' yang telah diisi oleh user.
3. Array yang telah dibalik: [4 6 6 6 4 2 1 4 2 3]: Mencetak array 'reversed_b' yang merupakan salinan dari array 'b' dengan elemen-elemen yang dibalik urutannya menggunakan slicing [::-1].

4.4 Analisis Program 4

Program ini adalah implementasi dalam Python menggunakan NumPy untuk mengubah array dua dimensi menjadi array satu dimensi. Dalam konteks ini, array dua dimensi array_dua_dimensi didefinisikan, dan kemudian fungsi flatten() dari NumPy digunakan untuk mengonversinya menjadi array satu dimensi. Hasilnya dicetak dengan menggunakan perintah print untuk menampilkan kedua versi array tersebut, baik yang dua dimensi maupun yang satu dimensi.



```
main.py > ...
1 import numpy as N
2
3 array_dua_dimensi = N.array([[1,2,3],[4,5,6],[7,8,9]])
4 array_satu_dimensi = array_dua_dimensi.flatten()
5
6 print("Array 2D:")
7 print(array_dua_dimensi)
8 print('\nArray 1D:')
9 print(array_satu_dimensi)
```

Gambar 4.7 Program 4

Analisis:

1. import numpy as N: Mengimpor pustaka NumPy dan menetapkan alias N.
2. array_dua_dimensi = N.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]): Membuat array dua dimensi array_dua_dimensi dengan menggunakan fungsi N.array.
3. array_satu_dimensi = array_dua_dimensi.flatten(): Menggunakan fungsi flatten() untuk membuat array satu dimensi array_satu_dimensi dari array dua dimensi array_dua_dimensi.
4. print("Array 2D:"): Mencetak pesan untuk array dua dimensi.
5. print(array_dua_dimensi): Mencetak array dua dimensi.
6. print("\nArray 1D:"): Mencetak pesan untuk array satu dimensi.
7. print(array_satu_dimensi): Mencetak array satu dimensi.

Contoh Eksekusi:

```
Array 2D:  
[[1 2 3]  
 [4 5 6]  
 [7 8 9]]  
  
Array 1D:  
[1 2 3 4 5 6 7 8 9]
```

Gambar 4.8 Output Program 4

Analisis Output:

1. Array 2D: ini adalah array dua dimensi yang telah dibuat dengan nilai-nilai tertentu.
2. [1 2 3 4 5 6 7 8 9]: Ini adalah array satu dimensi yang dihasilkan dari array dua dimensi dengan menggunakan fungsi flatten().

Program ini berhasil menghasilkan array dua dimensi dan kemudian meratakan (flatten) array tersebut menjadi satu dimensi. Hasil cetakan menunjukkan array dua dimensi dan array satu dimensi yang telah dihasilkan dengan benar.

4.5 Analisis Program 5

Program ini adalah implementasi dalam Python menggunakan NumPy untuk menukar elemen-elemen di atas dan di bawah diagonal utama suatu matriks 4x4. Dalam konteks ini, fungsi tukar_diagonal menerima matriks sebagai argumen dan mengembalikan matriks baru yang elemen-elemen di atas diagonal utama telah ditukar dengan elemen-elemen di bawah diagonal utama. Program kemudian mencetak matriks awal dan matriks setelah dilakukan pertukaran di bagian diagonal.

```

1 import numpy as N
2
3 def tukar_diagonal(matrix):
4     if matrix.shape != (4, 4):
5         raise ValueError("Matriks input harus berukuran 4x4.")
6
7     matriks_hasil = matrix.copy()
8
9     matriks_hasil[N.triu_indices(4, k=1)] = matrix[N.tril_indices(4, k=-1)]
10    matriks_hasil[N.tril_indices(4, k=-1)] = matrix[N.triu_indices(4, k=1)]
11
12    return matriks_hasil
13
14 matriks_input = N.array([[1, 2, 3, 4],
15                         [5, 6, 7, 8],
16                         [9, 10, 11, 12],
17                         [13, 14, 15, 16]])
18
19 print("Matriks Awal:")
20 print(matriks_input)
21
22 matriks_hasil = tukar_diagonal(matriks_input)
23
24 print("\nMatriks setelah ditukar:")
25 print(matriks_hasil)
26

```

Gambar 4.9 Program 5

Analisis:

1. import numpy as N: Mengimpor pustaka NumPy dan menetapkan alias N.
2. def tukar_diagonal(matrix):: Mendefinisikan fungsi tukar_diagonal yang menerima sebuah matriks sebagai argumen.
3. if matrix.shape != (4, 4):: Mengecek apakah matriks input memiliki ukuran 4x4. Jika tidak, program akan menaikkan ValueError.
4. matriks_hasil = matrix.copy(): Membuat salinan dari matriks input untuk dimodifikasi.
5. matriks_hasil[N.triu_indices(4, k=1)] = matrix[N.tril_indices(4, k=-1)]: Menggunakan indeks NumPy untuk menukar elemen di atas diagonal utama dengan elemen di bawah diagonal utama.
6. matriks_hasil[N.tril_indices(4, k=-1)] = matrix[N.triu_indices(4, k=1)]: Menggunakan indeks NumPy untuk menukar elemen di bawah diagonal utama dengan elemen di atas diagonal utama.
7. return matriks_hasil: Mengembalikan matriks hasil yang telah dimodifikasi.

8. matriks_input = N.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]): Membuat matriks input sebagai contoh dengan ukuran 4x4.
9. print("Matriks Awal:"): Mencetak pesan untuk matriks awal.
10. print(matriks_input): Mencetak matriks awal.
11. matriks_hasil = tukar_diagonal(matriks_input): Memanggil fungsi tukar_diagonal untuk menukar diagonal matriks.
12. print("\nMatriks setelah ditukar di bagian diagonal:"): Mencetak pesan untuk matriks setelah ditukar diagonal.
13. print(matriks_hasil): Mencetak matriks setelah ditukar diagonal.

Contoh Eksekusi:

```
Matriks Awal:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]

Matriks setelah ditukar:
[[ 1  5  9 10]
 [ 2  6 13 14]
 [ 3  4 11 15]
 [ 7  8 12 16]]
```

Gambar 4.10 Output Program 5

Analisis Output:

1. Matriks Awal: Ini adalah matriks input yang telah didefinisikan.
2. Matriks setelah ditukar: Matriks setelah ditukar menunjukkan bahwa elemen di atas diagonal utama telah ditukar dengan elemen di bawah diagonal utama, sesuai dengan implementasi dalam fungsi ‘tukar_diagonal’.

Jadi, output sesuai dengan fungsi yang diharapkan. Program ini bertujuan untuk menukar elemen-elemen di atas dan di bawah diagonal utama matriks, dan output menunjukkan bahwa pertukaran tersebut telah berhasil dilakukan

BAB V

PENUTUP

5.1 Kesimpulan

Dalam penutup laporan ini, dapat disimpulkan bahwa penggunaan NumPy dalam pemrograman Python memberikan keuntungan signifikan dalam manipulasi dan analisis data berukuran besar. NumPy menyediakan struktur data array yang efisien, operasi vektorisasi, dan berbagai fungsi matematika yang kuat, memungkinkan pengguna untuk dengan mudah melakukan operasi numerik kompleks. Kecepatan dan efisiensi NumPy membuatnya menjadi pilihan utama dalam pengolahan data ilmiah, pembelajaran mesin, dan berbagai aplikasi lainnya.

Dengan demikian, laporan ini diharapkan dapat memberikan wawasan yang cukup untuk memahami dasar-dasar penggunaan NumPy dan memberikan dasar yang kuat bagi para pembaca untuk menjelajahi lebih lanjut fitur-fitur canggih yang ditawarkan oleh pustaka ini. NumPy tidak hanya sebagai alat bantu, tetapi juga sebagai fondasi yang kokoh untuk eksplorasi lebih lanjut dalam analisis data dan pengolahan numerik menggunakan Python.

5.2 Saran

Untuk pengembangan lebih lanjut, disarankan agar pengguna mendalami lebih lanjut fitur-fitur lanjutan NumPy seperti operasi linier aljabar, fungsi-fungsi statistik, dan pemahaman lebih mendalam tentang vektorisasi kode.

Laporan Dokumentasi



Gambar 4.11 Dokumentasi Praktikum

LAMPIRAN 2 : SHORTCUT

1. Program 1

```
import numpy as np  
  
a = np.arange(8)  
  
condition = np.logical_and(a > 3, a < 6)  
  
answer = np.where(condition, a * 2, 0)  
  
print(a)  
  
print(condition)  
  
print(answer)
```

Output

```
[0 1 2 3 4 5 6 7]  
  
[False False False False  True  True False False]  
  
[ 0  0  0  0  8 10  0  0]
```

2. Program 2

```
import numpy as np

a = np.empty(10, dtype=int)

for i in range(10):

    user_input = input(f'Masukkan elemen {i + 1}: ')

    a[i] = int(user_input)

max_value, max_index = np.max(a), np.argmax(a)

min_value, min_index = np.min(a), np.argmin(a)

print('Array: ', a)

print(f'Nilai Maksimum: {max_value}, Posisi: {min_index}')

print(f'Nilai minimum: {min_value}, Posisi: {min_index}')
```

Output

```
Masukkan elemen 1: 3
Masukkan elemen 2: 5
Masukkan elemen 3: 3
Masukkan elemen 4: 2
Masukkan elemen 5: 6
Masukkan elemen 6: 1
Masukkan elemen 7: 7
Masukkan elemen 8: 2
Masukkan elemen 9: 11
Masukkan elemen 10: 9
Array: [ 3  5  3  2  6  1  7  2 11  9]
Nilai Maksimum: 11, Posisi: 5
Nilai minimum: 1, Posisi: 5
```

3. Program 3

```
import numpy as N  
  
b = N.empty(10, dtype=int)  
  
for i in range(10):  
  
    b[i] = int(input('Masukkan elemen {i + 1}: '))  
  
reversed_b = b[::-1]  
  
print('Array Asli: ', b)  
  
print('Array yang telah dibalik: ', reversed_b)
```

Output

```
Masukkan elemen 1: 2  
  
Masukkan elemen 2: 3  
  
Masukkan elemen 3: 4  
  
Masukkan elemen 4: 5  
  
Masukkan elemen 5: 6  
  
Masukkan elemen 6: 2  
  
Masukkan elemen 7: 8  
  
Masukkan elemen 8: 9  
  
Masukkan elemen 9: 1  
  
Masukkan elemen 10: 8  
  
Array Asli: [2 3 4 5 6 2 8 9 1 8]  
  
Array yang telah dibalik: [8 1 9 8 2 6 5 4 3 2]
```

4. Program 4

```
import numpy as N

array_dua_dimensi = N.array([[1,2,3], [4,5,6], [7,8,9]])

array_satu_dimensi = array_dua_dimensi.flatten()

print('Array 2D:')

print(array_dua_dimensi)

print('\nArray 1D:')

print(array_satu_dimensi)
```

Output

```
Array 2D:

[[1 2 3]
 [4 5 6]
 [7 8 9]]

Array 1D:

[1 2 3 4 5 6 7 8 9]
```

5. Program 5

```
import numpy as N

def tukar_diagonal(matrix):

    if matrix.shape != (4, 4):

        raise ValueError('Matrix input harus berukuran 4x4')

    matriks_hasil = matrix.copy()

    matriks_hasil[N.triu_indices(4, k=1)] = matrix[N.tril_indices(4, k=1)]

    matriks_hasil[N.triu_indices(4, k=1)] = matrix[N.triu_indices(4, k=1)]

    return matriks_hasil

matriks_input = N.array([[1,2,3,4],  
                        [5,6,7,8],  
                        [9,10,11,12],  
                        [13,14,15,16]])

print('Matriks Awal:')

print(matriks_input)

matrks_hasil = tukar_diagonal(matriks_input)

print('\nMatriks Setelah Ditukar: ')

print(matrks_hasil)
```

Output

Matriks Awal:	Matriks Setelah Ditukar:
[[1 2 3 4]	[[1 5 9 10]
[5 6 7 8]	[2 6 13 14]
[9 10 11 12]	[3 4 11 15]
[13 14 15 16]]	[7 8 12 16]]

DAFTAR PUSTAKA

- Taruna, Gregorius Kennard, and Setia Budi. "Penerapan Data Science pada Dataset Olympics." *Jurnal STRATEGI-Jurnal Maranatha* 4.2 (2022): 278-296.
- Dubois, P. F., Hinsen, K. & Hugunin, J. Numerical Python. *Comput. Phys.* **10**, 262–267 (1996).
- Ascher, D., Dubois, P. F., Hinsen, K., Hugunin, J. & Oliphant, T. E. *An Open Source Project: Numerical Python* (Lawrence Livermore National Laboratory, 2001).