**ALGORITMA DAN STRUKTUR DATA**

**MODUL 3**

**COLLECTIOS, ARRAYS, AND LINKED STRUCTURE**



**Disusun oleh:**

Muhammad Ferizal Fadhli

L200210119

D

**TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2022/2023**

Tugas Mahasiswa

1. Terkait array 2 dimensi kita akan membuat tipe data sebuah matrix yang berisi angka-angka. Untuk itu buatlah fungsi-fungsi

* untuk memastikan bahwa isi dan ukuran matrix-nya konsisten (karena tiap anggota dari list-luar-nya bisa saja mempunyai ukuran yang berbeda-beda, dan bahkan bisa saja berbeda tipe!)

Berikut kode program :

def is\_matrix\_consistent(matrix):

    if len(matrix) == 0:

        return True

    row\_length = len(matrix[0])

    for row in matrix:

        if len(row) != row\_length:

            return False

        if any(type(elem) != type(matrix[0][0]) for elem in row):

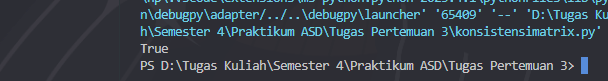
            return False

    return True

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

print(is\_matrix\_consistent(matrix))

Berikut SS Ketika program dijalankan :



* Untuk mengambil ukuran matrix.

Berikut kode program :

def ambilUkuranMatrix(matrix):

    row = len(matrix)

    col = len(matrix[0])

    print(f"Ukuran matrix adalah {row}x{col}")

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

ambilUkuranMatrix(matrix)

Berikut SS Ketika program dijalankan :



* Untuk menjumlahkan dua matrix.

Berikut kode program :

def tambahMatrix(matrix1,matrix2):

    matrix3 = []

    for i in range(len(matrix1)):

        baris = []

        for j in range(len(matrix1[0])):

            baris.append(matrix1[i][j] + matrix2[i][j])

        matrix3.append(baris)

    return matrix3

matrix1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

matrix2 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

print(tambahMatrix(matrix1,matrix2))

Berikut SS Ketika program dijalankan :



* Untuk mengkalikan dua matrix

Berikut kode program :

def kalikanMatrix(matrix1,matrix2):

    matrix3 = []

    for i in range(len(matrix1)):

        baris = []

        for j in range(len(matrix1[0])):

            baris.append(matrix1[i][j] \* matrix2[i][j])

        matrix3.append(baris)

    return matrix3

matrix1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

matrix2 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

print(kalikanMatrix(matrix1,matrix2))

Berikut SS Ketika program dijalankan :



* Untuk menghitung determinan sebuah matrix bujur sangkar.

Berikut kode program :

def calculate\_determinant(matrix):

*#Mengambil ukuran matrix*

    rows = len(matrix)

    cols = len(matrix[0])

*#mengambil apakah matrix bujur sangkar*

    if rows != cols:

        raise ValueError("Matrix harus bujur sangkar")

    if rows == 1:

        return matrix[0][0]

    if rows == 2:

        return matrix[0][0]\*matrix[1][1] - matrix[0][1]\*matrix[1][0]

    determinant = 0

    for j in range(cols):

        cofactor = (-1)\*\*(0+j) \* calculate\_minor(matrix, 0, j)

        determinant += matrix[0][j] \* cofactor

    return determinant

def calculate\_minor(matrix, i, j):

    rows = len(matrix)

    cols = len(matrix[0])

    submatrix = [[matrix[r][c] for c in range(cols) if c != j] for r in range(rows) if r != i]

    minor = calculate\_determinant(submatrix)

    return minor

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

print(“Deteminannya adalah”, calculate\_determinant(matrix))

Berikut SS Ketika program dijalankan :



1. Terkait matrix dan list comphrehension. Buatlah (dengan memanfaatkan list comprehension) fungsi-fungsi.

* untuk membangkitkan matrix berisi nol semua, dengan diberikan ukurannya. Pemanggilan: buatNol(m,n) dan buatNol(m). Pemanggilan dengan cara terakhir akan memberikan matrix bujursangkar ukuran m × m.

Berikut kode program :

def buatMatrixnol(m,n):

    M = []

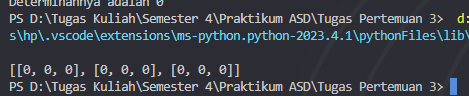
    for i in range(m):

        M.append([0]\*n)

    return M

print(buatMatrixnol(3,3))

Berikut SS Ketika program dijalankan :



* untuk membangkitkan matrix identitas, dengan diberikan ukurannya. Pemanggilan: buatIdentitas(m).

Berikut kode program :

def matrixidentitas(m):

    M = []

    for i in range(m):

        M.append([0]\*m)

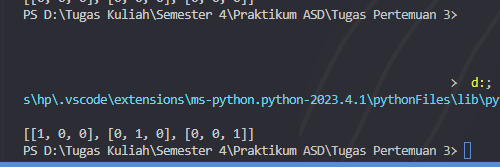
    for i in range(m):

        M[i][i] = 1

    return M

print(matrixidentitas(3))

Berikut SS Ketika program dijalankan :



1. Terkait linked list, buatlah fungsi untuk

* Mencaari data yang isinya tertentu

Berikut kode program :

def cari(head, search):

    current = head

    while current is not None:

        if current.data == search:

            return current

        current = current.next

    return None

head = Node(1)

head.**next** = Node(2)

head.**next**.**next** = Node(3)

head.**next**.**next**.**next** = Node(4)

head.**next**.**next**.**next**.**next** = Node(5)

search\_value = 3

hasil = cari(head, search\_value)

if hasil is not None:

    print(f"Data {search\_value} ditemukan")

else:

    print(f"Data {search\_value} tidak ditemukan")

Berikut SS Ketika program dijalankan :



* menambah suatu simpul di awal

Berikut kode program :

def tambahDepan(head):

    new\_node = Node(6)

    new\_node.**next** = head

    head = new\_node

    return head

* menambah suatu simpul di akhir

Berikut kode program :

def tambahBelakang(head):

    new\_node = Node(7)

    current = head

    while current.next is not None:

        current = current.next

    current.next = new\_node

    return head

* menyisipkan suatu simpul di mana saja

Berikut kode program :

def tambah(head, posisi):

    new\_node = Node(8)

    current = head

    if posisi == 0:

        new\_node.**next** = head

        head = new\_node

    else:

        for i in range(posisi-1):

            current = current.next

        new\_node.**next** = current.next

        current.next = new\_node

    return head

* menghapus suatu simpul di awal, di akhir, atau di mana saja: hapus(posisi)

Berikut kode program :

def hapus(head, posisi):

    current = head

    if posisi == 0:

        head = current.next

    else:

        for i in range(posisi-1):

            current = current.next

        current.next = current.next.next

    return head

1. Terkait doubly linked list, buatlah fungsi untuk

* mengunjungi dan mencetak data tiap simpul dari depan dan dari belakang.

Berikut adalah kode program :

class Node:

    def \_\_init\_\_(self, data):

        self.**data** = data

        self.**prev** = None

        self.**next** = None

class DoublyLinkedList:

    def \_\_init\_\_(self):

        self.**head** = None

    def add\_node(self, data):

        new\_node = Node(data)

        if self.**head** is None:

            self.**head** = new\_node

        else:

            current = self.**head**

            while current.**next**:

                current = current.**next**

            current.**next** = new\_node

            new\_node.**prev** = current

    def print\_forward(self):

        current = self.**head**

        while current:

            print(current.**data**)

            current = current.**next**

    def print\_backward(self):

        current = self.**head**

        while current.**next**:

            current = current.**next**

        while current:

            print(current.**data**)

            current = current.**prev**

dll = DoublyLinkedList()

dll.add\_node(1)

dll.add\_node(2)

dll.add\_node(3)

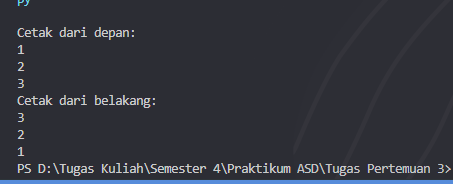
print("Cetak dari depan:")

dll.print\_forward()

print("Cetak dari belakang:")

dll.print\_backward()

Berikut SS Ketika program dijalankan :



* menambah simpul di awal.

Berikut kode program :

class Node:

    def \_\_init\_\_(self, data):

        self.**data** = data

        self.**prev** = None

        self.**next** = None

class DoublyLinkedList:

    def \_\_init\_\_(self):

        self.**head** = None

    def add\_node\_at\_beginning(self, data):

        new\_node = Node(data)

        if self.**head** is None:

            self.**head** = new\_node

        else:

            new\_node.**next** = self.**head**

            self.**head**.**prev** = new\_node

            self.**head** = new\_node

    def print\_list(self):

        current = self.**head**

        while current:

            print(current.**data**)

            current = current.**next**

dll = DoublyLinkedList()

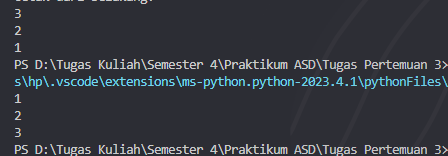
dll.add\_node\_at\_beginning(3)

dll.add\_node\_at\_beginning(2)

dll.add\_node\_at\_beginning(1)

dll.print\_list()

Berikut SS Ketika program dijalankan :



* menambah simpul di akhir

berikut kode program :

class Node:

    def \_\_init\_\_(self, data):

        self.**data** = data

        self.**prev** = None

        self.**next** = None

class DoublyLinkedList:

    def \_\_init\_\_(self):

        self.**head** = None

    def add\_node\_at\_end(self, data):

        new\_node = Node(data)

        if self.**head** is None:

            self.**head** = new\_node

        else:

            current = self.**head**

            while current.**next**:

                current = current.**next**

            current.**next** = new\_node

            new\_node.**prev** = current

    def print\_list(self):

        current = self.**head**

        while current:

            print(current.**data**)

            current = current.**next**

dll = DoublyLinkedList()

dll.add\_node\_at\_end(1)

dll.add\_node\_at\_end(2)

dll.add\_node\_at\_end(3)

*# print the list*

dll.print\_list()

Berikut SS Ketika program dijalankan :

