

**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 :

```
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> python konsistensimatrix.py  
True  
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3>
```

- 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 :

```
py
Ukuran matrix adalah 3x3
PS D:\Tugas Kuliah\Semester 4\
```

- 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 :

```
[[2, 4, 6], [8, 10, 12], [14, 16, 18]]
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> []
```

- 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 :

```
[[1, 4, 9], [16, 25, 36], [49, 64, 81]]
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> []
```

- 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 :

```
s\hp\.vscode\extensions\ms-python.pyth  
Determinannya adalah 0  
PS D:\Tugas Kuliah\Semester 4\Praktiku
```

2. Terkait matrix dan list comprehension. 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 $m \times m$.

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 :

```
Determinannya adalah 0  
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> d  
s\hp\.vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib  
  
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]  
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> |
```

- 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 :

```
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3>

> d;;

s\hp\.vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib\py

[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> |
```

3. 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 :

```
Data 3 ditemukan
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> |
```

- 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
```

4. 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 :

```
Cetak dari depan:
1
2
3
Cetak dari belakang:
3
2
1
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3>
```

- 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 :

```

PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3>
s\hp\.vscode\extensions\ms-python.python-2023.4.1\pythonFiles\
1
2
3
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3>

```

- 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 :

```
3
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> d:; cd 'd:\T
s\hp\.vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib\python\debu
1
2
3
PS D:\Tugas Kuliah\Semester 4\Praktikum ASD\Tugas Pertemuan 3> 
ord
```