

PRAKTIKUM STRUKTUR DATA

QUIZ 1

Kelas Senin 08.50



Oleh :

ALVIN ANUGERAH PRATAMA

(22343019)

Dosen Pengampu :

RANDI PROSKA SANDRA, S.PD., M.SC.

**PRODI INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG**

2022

PYTHON

- 1. Lists, Dictionary, Tuples, Sets
- Linked List
 - 2. Double List
 - 3. Skip List
- Queue
 - 4. Priority Queue

JAVA

- Array
- Linked List
 - 5. Skip List
- Stack
 - 6. Implementasi Menggunakan Queue
- Queue
 - 7. Circular Queue
 - 8. Priority Queue

1. Python

a) Lists, Dictionary, Tuples, Sets

➤ Python_Lists

```
1. # membuat list kosong
2. my_list = []
3.
4. # menambahkan elemen ke dalam list
5. my_list.append("Lamborghini")
6. my_list.append("Ferrari")
7. my_list.append("Bugatti")
8.
9.     # menampilkan isi dari list
10.    print( )
11.    print("Isi dari list adalah:")
12.    print(my_list)
13.    print( )
14.    # mengakses elemen dari list
15.    print("Elemen pertama dari list adalah:", my_list[0])
16.    print("Elemen kedua dari list adalah:", my_list[1])
17.    print("Elemen ketiga dari list adalah:", my_list[2])
18.    print( )
```

➤ Printscreen

```
Isi dari list adalah:
['Lamborghini', 'Ferrari', 'Bugatti']

Elemen pertama dari list adalah: Lamborghini
Elemen kedua dari list adalah: Ferrari
Elemen ketiga dari list adalah: Bugatti
```

| Baris Program | Source Code | Penjelasan |
|---------------|---|--|
| 1-2 | <pre> 1. # membuat list kosong 2. my_list = [] </pre> | Baris pertama ini membuat sebuah list kosong dengan nama my_list. List ini akan digunakan untuk menyimpan beberapa elemen. |
| 4-7 | <pre> # menambahkan elemen ke dalam list my_list.append("Lamborghini") my_list.append("Ferrari") my_list.append("Bugatti") </pre> | Baris keempat hingga ketujuh menambahkan tiga elemen ke dalam list my_list. Setiap elemen ditambahkan dengan menggunakan metode .append(), dimana metode ini akan menambahkan elemen baru ke akhir dari list. |
| 9-13 | <pre> # menampilkan isi dari list print() print("Isi dari list adalah:") print(my_list) print() </pre> | Baris kesembilan hingga ke tiga belas menampilkan isi dari list my_list dengan menggunakan perintah print. Baris keenam menampilkan sebuah string "Isi dari list adalah:" sebagai judul, kemudian baris ketujuh menampilkan elemen-elemen yang ada di dalam list my_list. |
| 14-18 | <pre> # mengakses elemen dari list print("Elemen pertama dari list adalah:", my_list[0]) print("Elemen kedua dari list adalah:", my_list[1]) print("Elemen ketiga dari list adalah:", my_list[2]) print() </pre> | Baris ke 14 hingga ke 18 mengakses elemen-elemen yang ada di dalam list my_list. Setiap elemen diakses dengan menggunakan indeksinya, yang dimulai dari nol. Setiap elemen kemudian ditampilkan sebagai string dengan menggunakan perintah print. Pada baris terakhir ditambahkan baris kosong untuk memberikan jarak antara output dengan baris kode selanjutnya. |

➤ Python_Dictionary

```

1. # membuat dictionary kosong
2. my_dict = {}
3.
4. # menambahkan elemen ke dalam dictionary
5. my_dict["nama"] = "Alvin Anugerah"
6. my_dict["umur"] = 19
7. my_dict["pekerjaan"] = "Mahasiswa"
8.
9. # menampilkan isi dari dictionary
10.     print( )
11.     print("Isi dari dictionary adalah:")
12.     print(my_dict)
13.     print( )
14.
15.     # mengakses nilai dari dictionary
16.     print("Nama saya adalah", my_dict["nama"])
17.     print("Saya berusia", my_dict["umur"], "tahun")
18.     print("Pekerjaan saya adalah", my_dict["pekerjaan"])
19.     print( )

```

➤ Printscreen

```

Isi dari dictionary adalah:
{'nama': 'Alvin Anugerah', 'umur': 19, 'pekerjaan': 'Mahasiswa'}

Nama saya adalah Alvin Anugerah
Saya berusia 19 tahun
Pekerjaan saya adalah Mahasiswa

```

| Baris Program | Source Code | Penjelasan |
|---------------|---|--|
| 1-2 | <pre> 1. # membuat dictionary kosong 2. my_dict = {} </pre> | Baris pertama ini membuat sebuah dictionary kosong dengan nama my_dict. Dictionary ini akan digunakan untuk menyimpan beberapa pasangan kunci-nilai. |
| 4-7 | <pre> # menambahkan elemen ke dalam dictionary my_dict["nama"] = "Alvin Anugerah" my_dict["umur"] = 19 my_dict["pekerjaan"] = "Mahasiswa" </pre> | Baris ke 4 hingga ke 7 menambahkan tiga pasangan kunci-nilai ke dalam dictionary my_dict. Setiap pasangan ditambahkan dengan menggunakan sintaks [kunci] = nilai, dimana kunci adalah string yang digunakan sebagai kunci dan nilai adalah nilai yang akan disimpan di bawah kunci tersebut. |
| 9-13 | <pre> # menampilkan isi dari dictionary </pre> | Baris 9 hingga ke 13 menampilkan isi dari dictionary my_dict dengan |

| | | |
|------|---|---|
| | <pre>print() print("Isi dari dictionary adalah:") print(my_dict) print()</pre> | <p>menggunakan perintah print. Baris keenam menampilkan sebuah string "Isi dari dictionary adalah:" sebagai judul, kemudian baris ketujuh menampilkan pasangan kunci-nilai yang ada di dalam dictionary my_dict.</p> |
| 15-1 | <pre># mengakses nilai dari dictionary ary print("Nama saya adalah", my_dict["nama"]) print("Saya berusia", my_dict["umur"], "tahun") print("Pekerjaan saya adalah", my_dict["pekerjaan"]) print()</pre> | <p>Baris ke 15 hingga ke 19 mengakses nilai-nilai yang ada di dalam dictionary my_dict. Setiap nilai diakses dengan menggunakan kuncinya, yang merupakan string. Setiap nilai kemudian ditampilkan sebagai string dengan menggunakan perintah print. Pada baris terakhir ditambahkan baris kosong untuk memberikan jarak antara output dengan baris kode selanjutnya.</p> |

➤ Python_Tuples

```
1. # membuat tuple
2. my_tuple = ("apel", "jeruk", "mangga")
3.
4. # menampilkan isi dari tuple
5. print( )
6. print("Isi dari tuple adalah:")
7. print(my_tuple)
8. print( )
9.
10. # mengakses elemen dari tuple
11. print("Elemen pertama dari tuple adalah:", my_tuple[0])
12. print("Elemen kedua dari tuple adalah:", my_tuple[1])
13. print("Elemen ketiga dari tuple adalah:", my_tuple[2])
14. print( )
```

➤ Printscreen

```
Isi dari tuple adalah:
('apel', 'jeruk', 'mangga')

Elemen pertama dari tuple adalah: apel
Elemen kedua dari tuple adalah: jeruk
Elemen ketiga dari tuple adalah: mangga
```

| Baris Program | Source Code | Penjelasan |
|---------------|--|---|
| 1-2 | <pre> 1. # membuat tuple 2. my_tuple = ("apel", "jeruk", "mangga") </pre> | Baris ini membuat sebuah tuple yang berisi tiga elemen, yaitu "apel", "jeruk", dan "mangga". Tuple ini disimpan di dalam variabel my_tuple. |
| 4-8 | <pre> # menampilkan isi dari tuple print() print("Isi dari tuple adalah:") print(my_tuple) print() </pre> | Baris ini menampilkan isi dari tuple my_tuple dengan menggunakan perintah print. Baris pertama menampilkan sebuah string sebagai judul, kemudian baris kedua menampilkan elemen-elemen yang ada di dalam tuple. |
| 10-14 | <pre> # mengakses elemen dari tuple print("Elemen pertama dari tuple adalah:", my_tuple[0]) print("Elemen kedua dari tuple adalah:", my_tuple[1]) print("Elemen ketiga dari tuple adalah:", my_tuple[2]) print() </pre> | Baris ini mengakses elemen-elemen yang ada di dalam tuple my_tuple. Setiap elemen diakses dengan menggunakan indeksinya, yang dimulai dari nol. |

➤ Python_Sets

```

1. # membuat set kosong
2. my_set = set()
3.
4. # menambahkan elemen ke dalam set
5. my_set.add("apel")
6. my_set.add("jeruk")
7. my_set.add("mangga")
8.
9. # menampilkan isi dari set
10.     print( )
11.     print("Isi dari set adalah:")
12.     print(my_set)
13.     print( )
14.
15.     # menghapus elemen dari set
16.     my_set.remove("jeruk")
17.
18.     # menampilkan isi dari set setelah dihapus
19.     print( )
20.     print("Isi dari set setelah dihapus adalah:")
21.     print(my_set)

```

➤ **Printscreen**

```
Isi dari set adalah:
{'apel', 'jeruk', 'mangga'}
```

```
Isi dari set setelah dihapus adalah:
{'apel', 'mangga'}
```

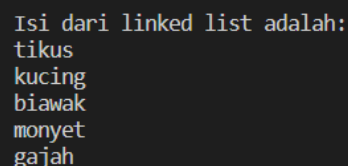
| Baris Program | Source Code | Penjelasan |
|---------------|---|--|
| 1-2 | <pre>1. # membuat set kosong 2. my_set = set()</pre> | Baris ini membuat sebuah set kosong dan menyimpannya di dalam variabel my_set. |
| 4-7 | <pre># menambahkan elemen ke dalam set my_set.add("apel") my_set.add("jeruk") my_set.add("mangga")</pre> | Baris ini menambahkan tiga elemen ke dalam set my_set dengan menggunakan perintah add. yaitu apel, jeruk, mangga |
| 9-13 | <pre># menampilkan isi dari set print() print("Isi dari set adalah:") print(my_set) print()</pre> | Baris ini menampilkan isi dari set my_set dengan menggunakan perintah print. |
| 15-16 | <pre># menghapus elemen dari set my_set.remove("jeruk")</pre> | Baris ini menghapus elemen "jeruk" dari set my_set dengan menggunakan perintah remove. |
| 18-21 | <pre># menampilkan isi dari set se telah dihapus print() print("Isi dari set setelah d ihapus adalah:") print(my_set)</pre> | Baris ini menampilkan isi dari set my_set setelah elemen "jeruk" dihapus. |

b) Linked List

➤ Python_Double Linked List

```
1. # definisi kelas Node
2. class Node:
3.     def __init__(self, data):
4.         self.data = data
5.         self.next = None
6.         self.prev = None
7.
8. # definisi kelas DoubleLinkedList
9. class DoubleLinkedList:
10.     def __init__(self):
11.         self.head = None
12.
13.     # method untuk menambahkan node di awal linked list
14.     def add_beginning(self, new_data):
15.         new_node = Node(new_data)
16.         if self.head is None:
17.             self.head = new_node
18.         else:
19.             new_node.next = self.head
20.             self.head.prev = new_node
21.             self.head = new_node
22.
23.     # method untuk menampilkan isi dari linked list
24.     def display(self):
25.         current_node = self.head
26.         while current_node is not None:
27.             print(current_node.data, end=" ")
28.             print( )
29.             current_node = current_node.next
30.
31. # membuat objek dari kelas DoubleLinkedList
32. dll = DoubleLinkedList()
33.
34. # menambahkan beberapa node ke dalam linked list
35. dll.add_beginning("gajah")
36. dll.add_beginning("monyet")
37. dll.add_beginning("biawak")
38. dll.add_beginning("kucing")
39. dll.add_beginning("tikus")
40.
41. # menampilkan isi dari linked list
42. print( )
43. print("Isi dari linked list adalah:")
44. dll.display()
```

- Printscreen



```
Isi dari linked list adalah:
tikus
kucing
biawak
monyet
gajah
```


| Baris Program | Source Code | Penjelasan |
|---------------|---|--|
| 1-6 | <pre># definisi kelas Node class Node: def __init__(self, data): self.data = data self.next = None self.prev = None</pre> | Baris ini mendefinisikan kelas Node dengan tiga atribut, yaitu data untuk menyimpan data dari node, next untuk menyimpan pointer ke node selanjutnya, dan prev untuk menyimpan pointer ke node sebelumnya. |
| 8-29 | <pre># definisi kelas DoubleLinkedList class DoubleLinkedList: def __init__(self): self.head = None # method untuk menambahkan # node di awal linked list def add_beginning(self, new_data): new_node = Node(new_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 # method untuk menampilkan # isi dari linked list</pre> | Baris ini mendefinisikan kelas DoubleLinkedList dengan dua metode, yaitu add_beginning untuk menambahkan node di awal linked list dan display untuk menampilkan isi dari linked list. |

| | | |
|-------|---|--|
| | <pre> def display(self): current_node = self.head while current_node is not None: print(current_node.data, end=" ") print() current_node = current_node.next </pre> | |
| 31-44 | <pre> # membuat objek dari kelas DoubleLinkedList dll = DoubleLinkedList() # menambahkan beberapa node ke dalam linked list dll.add_beginning("gajah") dll.add_beginning("monyet") dll.add_beginning("biawak") dll.add_beginning("kucing") dll.add_beginning("tikus") # menampilkan isi dari linked list print() print("Isi dari linked list adalah:") dll.display() </pre> | <p>Baris ini membuat objek dll dari kelas DoubleLinkedList dan menambahkan beberapa node ke dalam linked list menggunakan metode add_beginning. Kemudian, isi dari linked list ditampilkan menggunakan metode display.</p> |

➤ Python_Skip Linked List

```

1. my_list = ["januari", "febuari", "maret", "april", "maret"]
   # List
2.
3. skip_index = 0 # Index untuk diskip
4.
5. my_list.pop(skip_index) # dihapus oleh index
6.
7. print()
8. print("Isi Dari Skip List : ")
9. for i in my_list: # Loop pada new_list
10.     print(i)

```

- Printscreen

```

Isi Dari Skip List :
febuari
maret
april
maret

```

| Baris Program | Source Code | Penjelasan |
|---------------|--|---|
| 1 | <pre> my_list = ["januari", "febuari", "maret", "april", "maret"] # List </pre> | Sebuah list dengan nama my_list dibuat dan diisi dengan lima elemen, yaitu "januari", "febuari", "maret", "april", dan "maret". |
| 3 | <pre> skip_index = 0 # Index untuk diskip </pre> | Sebuah variabel dengan nama skip_index dibuat dan diinisialisasi dengan nilai 0. |
| 5 | <pre> my_list.pop(skip_index) # dihapus oleh index </pre> | Elemen pada indeks yang disimpan dalam variabel skip_index (yaitu indeks ke-0) dihapus dari list my_list. |
| 7-10 | <pre> print() print("Isi Dari Skip List : ") for i in my_list: # Loop pada new_list print(i) </pre> | Sebuah loop for dibuat dengan variabel iterasi i, yang akan mengulang sebanyak elemen pada my_list. Setiap iterasi pada loop akan mencetak nilai dari variabel iterasi i. |

c) Queue

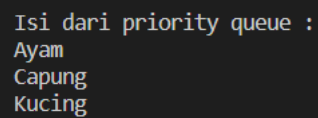
➤ Python_Priority Queue

```

1. import heapq
2.
3. class PriorityQueue:
4.     def __init__(self):
5.         self._queue = []
6.         self._index = 0
7.
8.     def push(self, item, priority):
9.         heapq.heappush(self._queue, (priority, self._index,
10. item))
11.         self._index += 1
12.
13.     def pop(self):
14.         return heapq.heappop(self._queue)[-1]
15.
16. # contoh penggunaan
17. q = PriorityQueue()
18. q.push("Capung", 2)
19. q.push("Kucing", 3)
20. q.push("Ayam", 1)
21.
22. print( )
23. print("Isi dari priority queue : ")
24. print(q.pop()) # Item 2
25. print(q.pop()) # Item 3
26. print(q.pop()) # Item 1
27. print( )

```

• Printscreen



```

Isi dari priority queue :
Ayam
Capung
Kucing

```

| Baris Program | Source Code | Penjelasan |
|---------------|--|--|
| 1 | <code>import heapq</code> | Mengimport modul heapq yang merupakan implementasi dari queue priority di Python menggunakan heap binary. |
| 3-13 | <pre> class PriorityQueue: def __init__(self): self._queue = [] self._index = 0 def push(self, item, priority): </pre> | Membuat sebuah class PriorityQueue yang memiliki dua method yaitu push dan pop. Pada method push, elemen dan nilai prioritasnya dimasukkan ke dalam sebuah tuple yang kemudian dimasukkan ke dalam list _queue. Nilai _index akan diincrement setiap kali terdapat |

| | | |
|-------|--|---|
| | <pre> heapq.heappush(self._queue, (priority, self._index, item)) self._index += 1 def pop(self): return heapq.heappop(self._queue)[-1] </pre> | <p>penambahan elemen ke dalam _queue. Pada method pop, elemen dengan nilai prioritas tertinggi akan diambil dari _queue menggunakan heappop dari modul heapq.</p> |
| 15-19 | <pre> # contoh penggunaan q = PriorityQueue() q.push("Capung", 2) q.push("Kucing", 3) q.push("Ayam", 1) </pre> | <p>Membuat sebuah instance dari class PriorityQueue dan menambahkan 3 elemen ke dalamnya dengan nilai prioritas masing-masing 2, 3, dan 1.</p> |
| 21-26 | <pre> print() print("Isi dari priority queue : ") print(q.pop()) # Item 2 print(q.pop()) # Item 3 print(q.pop()) # Item 1 print() </pre> | <p>Menampilkan elemen-elemen yang ada pada priority queue dalam urutan nilai prioritasnya yang sudah diatur sebelumnya dengan menggunakan method pop.</p> |

2. Java

1. Java_Array

```
1. package array;
2. public class Array {
3.     public static void main(String[] args) {
4.         // membuat array kosong dengan panjang 5
5.         int[] myArray = new int[5];
6.
7.         // mengisi elemen pada array
8.         myArray[0] = 10;
9.         myArray[1] = 20;
10.        myArray[2] = 30;
11.        myArray[3] = 40;
12.        myArray[4] = 50;
13.
14.        // menampilkan isi dari array
15.        System.out.println("Isi dari array adalah:");
16.        for (int i = 0; i < myArray.length; i++) {
17.            System.out.println(myArray[i]);
18.        }
19.    }
20. }
```

- Printscreen

```
Isi dari array adalah:
10
20
30
40
50
```

| Baris Program | Source Code | Penjelasan |
|---------------|---|---|
| 1 | package array; | Baris kode ini mendeklarasikan paket Java dengan nama "array". Paket ini akan berisi kelas-kelas Java yang berhubungan dengan array. |
| 2 | public class Array { | Baris kode ini mendeklarasikan kelas Java dengan nama "Array". Kelas ini akan berisi metode untuk membuat, mengisi, dan menampilkan isi dari array. |
| 3 | public static void main(String [] args) { | Baris kode ini mendeklarasikan metode utama (main) dari program Java. Metode ini akan dijalankan ketika program dijalankan, dan akan |

| | | |
|-------|--|---|
| | | memulai eksekusi program. |
| 4-5 | <pre>// membuat array kosong dengan panjang 5 int[] myArray = new int[5];</pre> | Baris kode ini membuat sebuah array kosong dengan panjang 5, dan menyimpannya dalam variabel myArray. Tipe data dari array tersebut adalah integer (int). |
| 7-12 | <pre>// mengisi elemen pada array myArray[0] = 10; myArray[1] = 20; myArray[2] = 30; myArray[3] = 40; myArray[4] = 50;</pre> | Baris kode ini mengisi elemen-elemen pada array myArray dengan nilai 10, 20, 30, 40, dan 50. |
| 14-18 | <pre>// menampilkan isi dari array System.out.println("Isi dari a rray adalah:"); for (int i = 0; i < myArray.le ngth; i++) { System.out.println(myArray[i]) ; } }</pre> | Baris kode ini menampilkan sebuah pesan teks ke konsol, yaitu "Isi dari array adalah:". Baris kode ini memulai sebuah loop for untuk menampilkan isi dari array myArray. Loop tersebut akan mengeksekusi blok kode yang di dalamnya sebanyak myArray.length (yaitu 5) kali. Setiap kali loop dieksekusi, nilai dari elemen ke-i pada myArray akan ditampilkan ke konsol menggunakan metode println dari kelas System. |

b) Linked List

➤ Java_Skip List

c) Stack

➤ Java_Implementasi menggunakan Queue

```
1. import java.util.LinkedList;
2. import java.util.Queue;
3.
4. public class StackWithQueue {
5.
6.     private Queue<Integer> q1 = new LinkedList<>();
7.     private Queue<Integer> q2 = new LinkedList<>();
8.     private int top;
9.
10.    // Constructor
11.    public StackWithQueue() {
12.        top = 0;
13.    }
14.
15.    // Push operation
16.    public void push(int x) {
17.        q1.add(x);
18.        top = x;
19.    }
20.
21.    // Pop operation
22.    public int pop() {
23.        if (q1.isEmpty()) {
24.            return -1;
25.        }
26.
27.        while (q1.size() != 1) {
28.            top = q1.remove();
29.            q2.add(top);
30.        }
31.
32.        int popValue = q1.remove();
33.
34.        Queue<Integer> temp = q1;
35.        q1 = q2;
36.        q2 = temp;
37.
38.        return popValue;
39.    }
40.
41.    // Top operation
42.    public int top() {
43.        return top;
44.    }
45.
46.    // Check if the stack is empty
47.    public boolean isEmpty() {
48.        return q1.isEmpty();
49.    }
50.
51.    // Main method
52.    public static void main(String[] args) {
53.        StackWithQueue stack = new StackWithQueue();
54.
55.        stack.push(1);
56.        stack.push(2);
57.        stack.push(3);
58.    }
```



```

59.         System.out.println("Elemen paling atas: " + st
      ack.top());
60.         System.out.println(stack.pop() + " diambil dar
      i stack");
61.
62.         System.out.println("Elemen paling atas: " + st
      ack.top());
63.         System.out.println(stack.pop() + " diambil dar
      i stack");
64.         System.out.println(" ")
65.         System.out.println("Apakah stack kosong? " + s
      tack.isEmpty());
66.     }
67. }

```

- Printscreen

```

Elemen paling atas: 3
3 diambil dari stack
Elemen paling atas: 2
2 diambil dari stack

Apakah stack kosong? false

```

| Baris Program | Source Code | Penjelasan |
|---------------|---|---|
| 1-2 | <pre>import java.util.LinkedList; import java.util.Queue;</pre> | Mendefinisikan package dan mengimport kelas LinkedList dan Queue dari pustaka java.util |
| 4-13 | <pre>public class StackWithQueue { private Queue<Integer> q1 = new LinkedList<>(); private Queue<Integer> q2 = new LinkedList<>(); private int top; // Constructor public StackWithQueue() { top = 0; } }</pre> | Mendefinisikan kelas StackWithQueue dan variabel-variabelnya. StackWithQueue memiliki dua antrian q1 dan q2 yang masing-masing bertipe LinkedList, variabel top yang menyimpan nilai elemen teratas, dan konstruktor yang menginisialisasi nilai top menjadi 0. |
| 15-19 | <pre>// Push operation public void push(int x) {</pre> | Mendefinisikan metode push() untuk menambah elemen ke dalam stack menggunakan antrian q1. Metode ini juga |

| | | |
|-------|---|--|
| | <pre> q1.add(x); top = x; } </pre> | memperbarui nilai top dengan nilai elemen terbaru. |
| 21-39 | <pre> // Pop operation public int pop() { if (q1.isEmpty()) { return -1; } while (q1.size() != 1) { top = q1.remove(); q2.add(top); } int popValue = q1.remove(); Queue<Integer> temp = q1; q1 = q2; q2 = temp; return popValue; } </pre> | Mendefinisikan metode pop() untuk menghapus dan mengembalikan elemen teratas dari stack. Jika q1 kosong, metode ini mengembalikan nilai -1. Selain itu, metode ini memindahkan semua elemen dari q1 ke q2 kecuali elemen terakhir yang merupakan elemen teratas stack. Kemudian, metode ini mempertukarkan q1 dengan q2 dan mengembalikan nilai elemen teratas yang dihapus dari q1. |
| 41-44 | <pre> // Top operation public int top() { return top; } </pre> | Mendefinisikan metode top() untuk mengembalikan nilai top. |
| 46-49 | <pre> // Check if the stack is empty public boolean isEmpty() { return q1.isEmpty(); } </pre> | Mendefinisikan metode isEmpty() untuk memeriksa apakah q1 kosong. |
| 51-67 | <pre> // Main method public static void main(String [] args) { </pre> | Metode main() yang berfungsi sebagai pengujian kelas StackWithQueue. Dalam metode ini, objek stack |

| | | |
|--|--|--|
| | <pre> StackWithQueue stack = new StackWithQueue(); stack.push(1); stack.push(2); stack.push(3); System.out.println("Elemen paling atas: " + stack.top()); System.out.println(stack.pop() + " diambil dari stack"); System.out.println("Elemen paling atas: " + stack.top()); System.out.println(stack.pop() + " diambil dari stack"); System.out.println(" ") System.out.println("Apakah stack kosong? " + stack.isEmpty()); } } </pre> | <p>dibuat dan tiga elemen ditambahkan ke dalamnya menggunakan metode push(). Kemudian, elemen teratas stack dan elemen pertama yang dihapus dari stack ditampilkan menggunakan metode top() dan pop(). Akhirnya, metode isEmpty() dipanggil untuk memeriksa apakah stack kosong.</p> |
|--|--|--|

d) Queue

➤ Java_Circular Queue

➤ Java_Priority Queue

```

1. import java.util.PriorityQueue;
2.
3. public class PriorityQueueExample {
4.     public static void main(String[] args) {
5.         PriorityQueue<String> pq = new PriorityQueue<>();
6.
7.         pq.add("Nanas");
8.         pq.add("Pepaya");
9.         pq.add("Apel");
10.        pq.add("Jambu");
11.        pq.add("Semangka");
12.
13.        System.out.println("Isi dari Priority Queue: "
14.    );
15.        while (!pq.isEmpty()) {
16.            System.out.println(pq.poll());
17.        }
18.    }

```

- Printscreen

```

Isi dari Priority Queue:
Apel
Jambu
Nanas
Pepaya
Semangka

```

| Baris Program | Source Code | Penjelasan |
|---------------|--|---|
| 1 | <code>import java.util.PriorityQueue;</code> | Mengimport kelas PriorityQueue dari Java Collections Framework. |
| 3 | <code>public class PriorityQueueExample {</code> | Deklarasi kelas PriorityQueueExample. |
| 4-5 | <code>public static void main(String[] args) {</code> <code>PriorityQueue<String> pq = new PriorityQueue<>();</code> | Deklarasi objek PriorityQueue dengan tipe elemen String yang dinamakan "pq". |
| 7-11 | <code>pq.add("Nanas");</code> <code>pq.add("Pepaya");</code> <code>pq.add("Apel");</code> <code>pq.add("Jambu");</code> <code>pq.add("Semangka");</code> | Menambahkan beberapa elemen ke PriorityQueue "pq" menggunakan metode add(). |
| 13-15 | <code>System.out.println("Isi dari Priority Queue: ");</code> <code>while (!pq.isEmpty())</code> <code>{</code> <code>System.out.println(pq.poll());</code> <code>}</code> | Mencetak string "Isi dari Priority Queue: ". Mengeluarkan elemen-elemen dari PriorityQueue "pq" menggunakan metode poll() dan mencetaknya ke layar sampai PriorityQueue kosong. PriorityQueue akan mengeluarkan elemen dengan prioritas tertinggi terlebih dahulu, yaitu elemen dengan urutan alfabet dalam kasus ini. |