LAPORAN PRAKTIKUM

STRUKTUR DATA LINIER

MODUL VIII

Dosen Pengampu

JB. Budi Darmawan S.T., M.Sc.

DISUSUN OLEH :

AGUSTINUS KEVIN YUDIPRATAMA

235314029

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS SANATA DHARMA

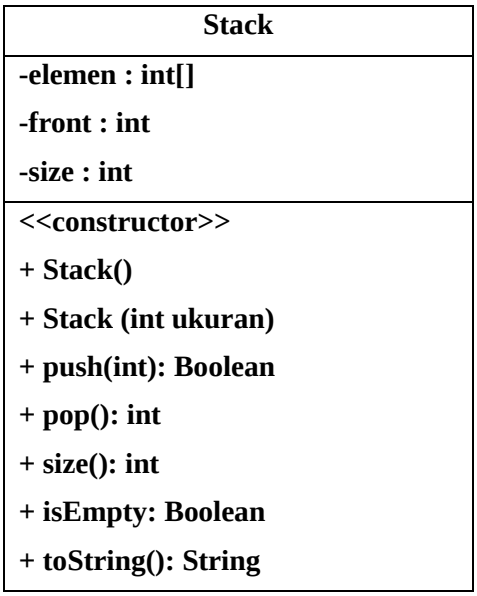
YOGYAKARTA

2024

A. TUJUAN PRAKTIKUM

- Mahasiswa mampu membuat program struktur data Queue (antrian), dengan struktur data statis (array)

B. DIAGRAM UML



C. SOURCE CODE

Class mainnya

```java
package vscode.Modul8;

import java.util.NoSuchElementException;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        Queue antrian = new Queue(ukuran:7);
        antrian.Enqueue(x:26);
        antrian.Enqueue(x:15);
        antrian.Enqueue(x:8);
        antrian.Enqueue(x:14);
        try{
            System.out.println(antrian.dequeue());
            System.out.println(antrian.dequeue());
            System.out.println(antrian.dequeue());
        } catch (NoSuchElementException x){
            System.out.println(x:"amtrian kosong");
        }
    }
}
```

Classnya

```java
package vscode.Modul8;

import java.util.NoSuchElementException;

public class Queue {
    private int[] elemen;
    private int front;
    private int rear;
    private int size;

    public Queue(){
        front = 0;
        size = 0;
        rear = 0;
        elemen = new int[5];

    }
    public Queue (int ukuran){
        front = 0;
        size = 0;
        rear = 0;
        elemen = new int[ukuran];
    }
    public  boolean Enqueue(int x){
        if (size != elemen.length) {
            size ++;
            elemen[rear]= x;
            if (rear < elemen.length -1) {
                rear ++;
            }else{
                rear = 0;
            }
           return true ;
        }
        return false;

    }
    public int size(){
        return size;
    }
    public boolean isEmpty(){
        if (size == 0) {
            return true;

        }
        return false;
    }
    public int dequeue(){
        int temp;
        if (!isEmpty()) {
            size --;
            temp = elemen[front];
            if (front < elemen.length-1) {
                front ++;
            } else {
                front= 0;
            }
            return temp;
        }
        throw new NoSuchElementException();
    }
}
```

D. OUTPUT

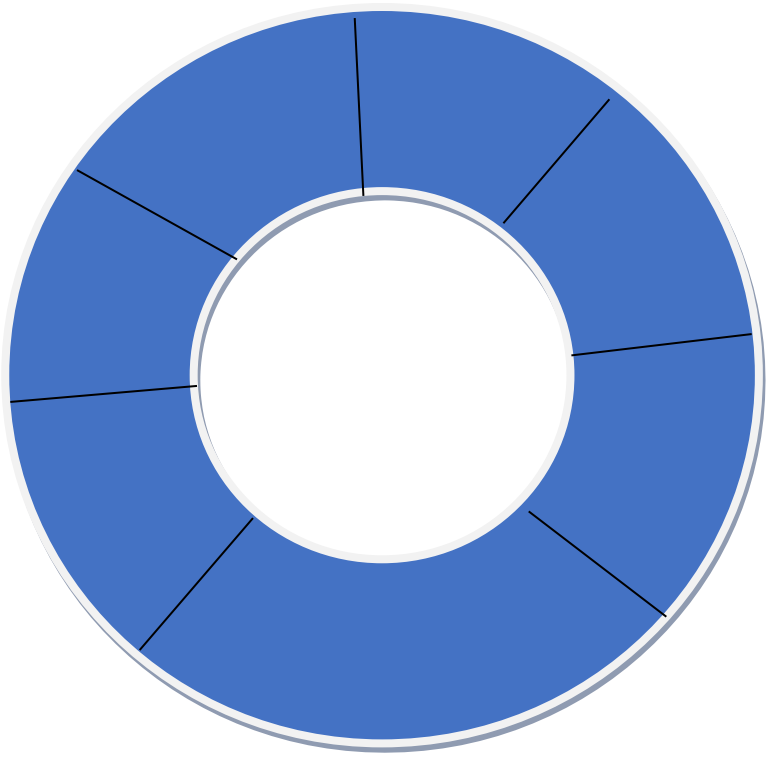
```
26
15
8
```

E. ANALISIS

Queue antrian = new Queue(7)

antrian.length = 7

Size = 0



Rear = 0

Front = 0

Queue antrian = new Queue(7)

antrian.enqueue(26)

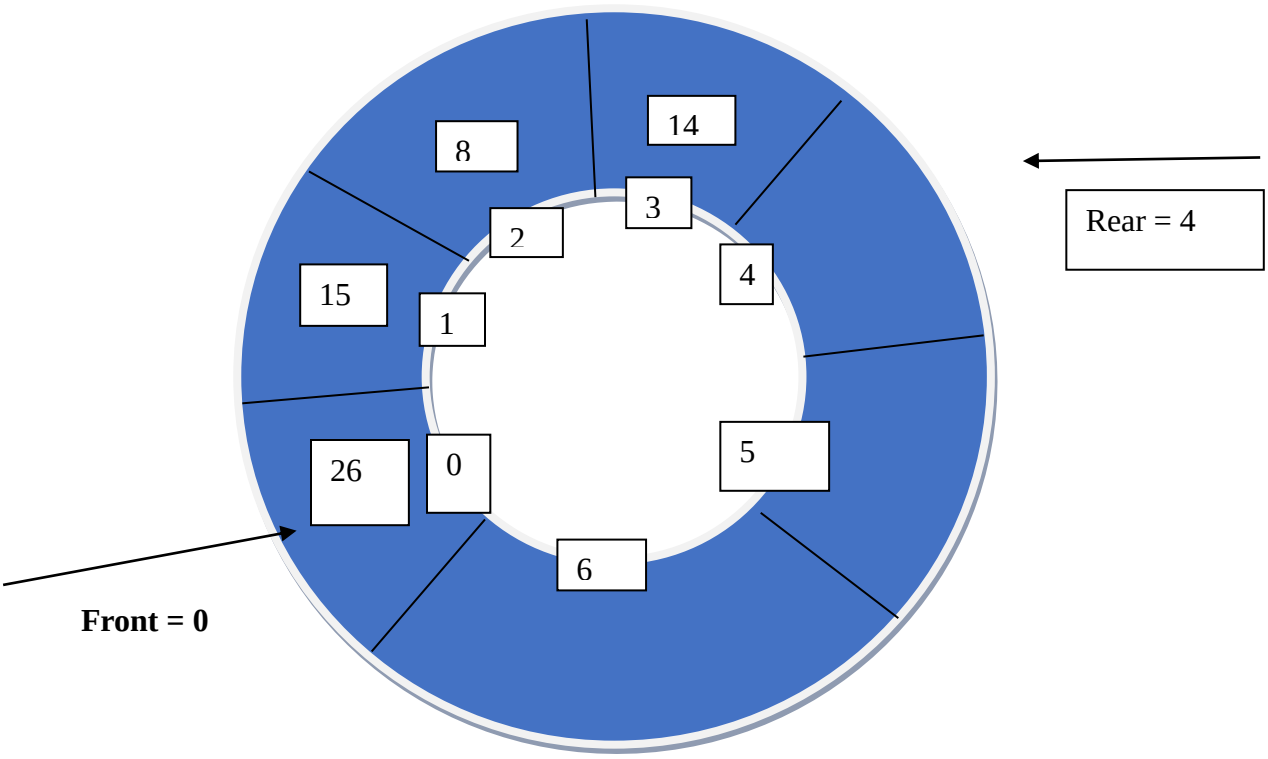
antrian.enqueue(15)

antrian.enqueue(8)

antrian.enqueue(14)

antrian.length = 7

Size = 4



Queue antrian = new Queue(7)

Antrian.enqueue(26)

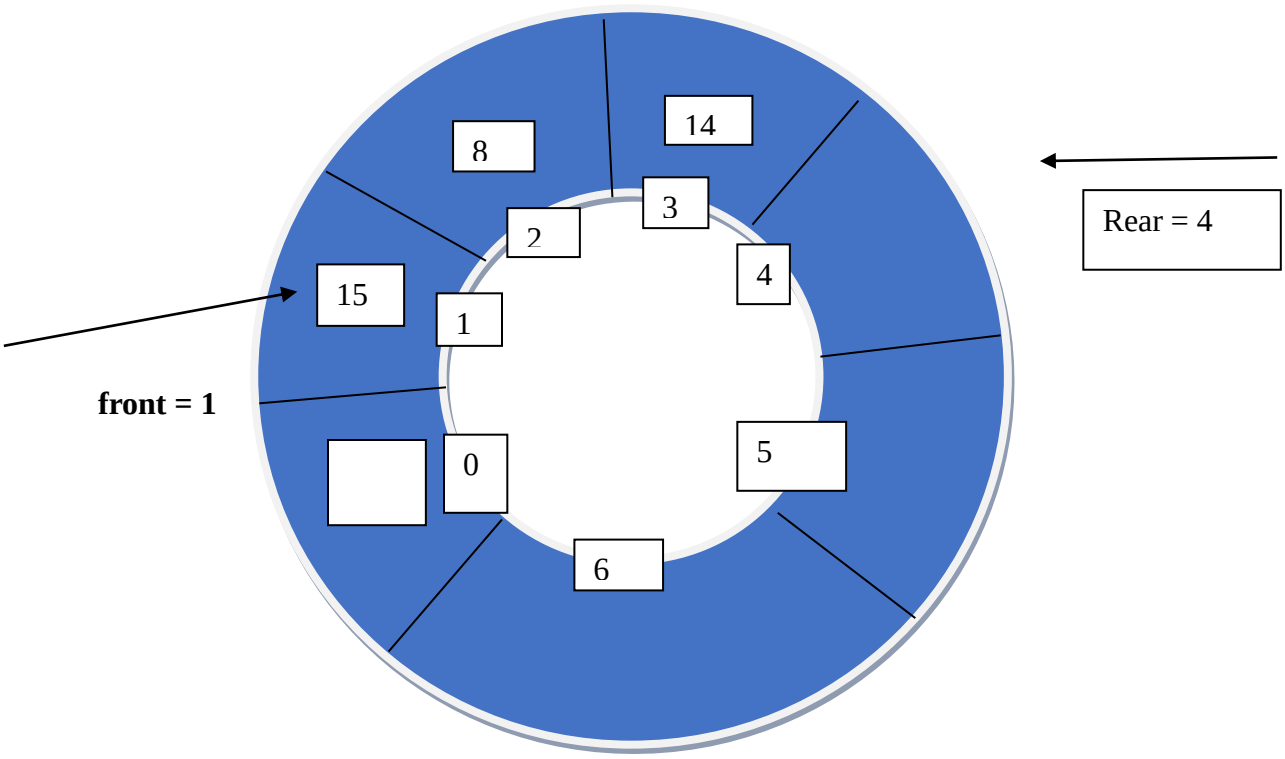
antrian.enqueue(15)

antrian.enqueue(8)

antrian.enqueue(14)

antrian.length = 7

Size = 3



Queue antrian = new Queue(7)

Antrian.enqueue(26)

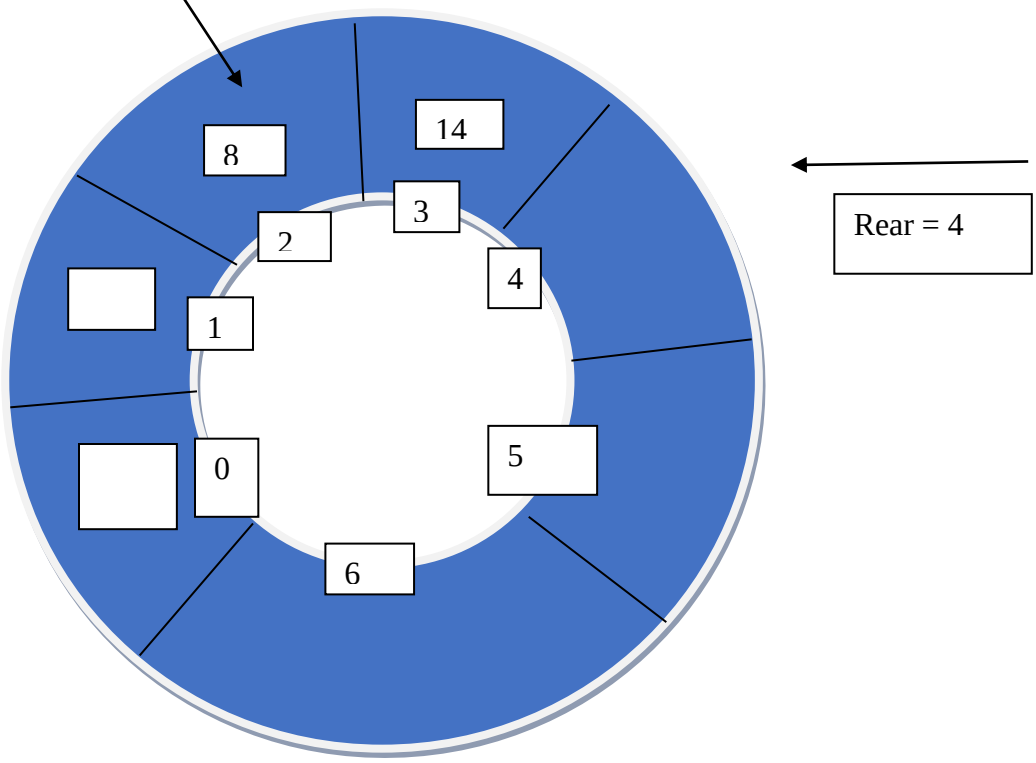
antrian.enqueue(15)

antrian.enqueue(8)

antrian.enqueue(14)

antrian.length = 7

Size = 2



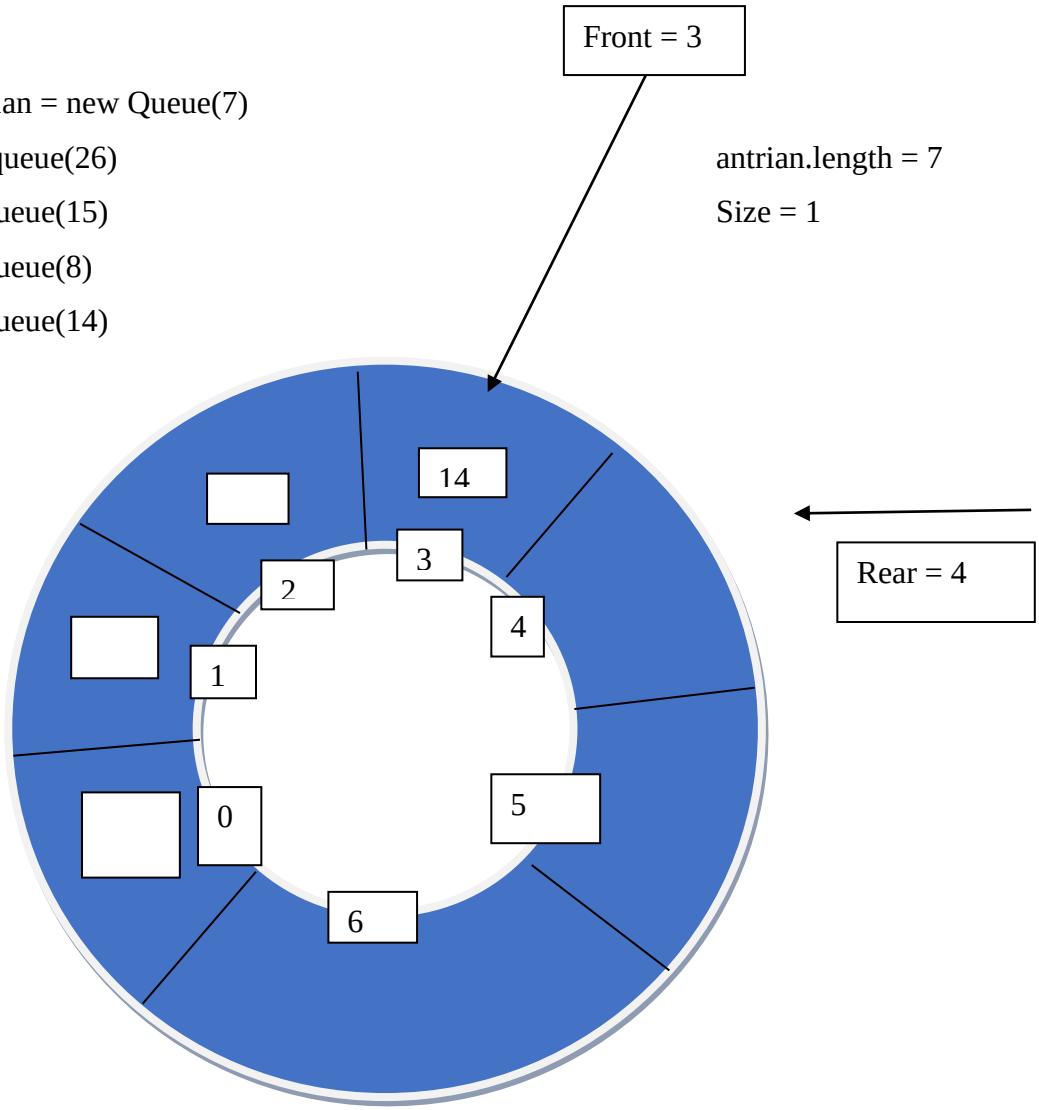
Queue antrian = new Queue(7)

Antrian.enqueue(26)

antrian.enqueue(15)

antrian.enqueue(8)

antrian.enqueue(14)



Queue antrian = new Queue(7)

Antrian.enqueue(26)

Enqueue(15)

Antrian.enqueue(8)

Antrian.enqueue(14)