

LAPORAN
Struktur Data Linear
Praktikum 8 : Queue (Antrian)

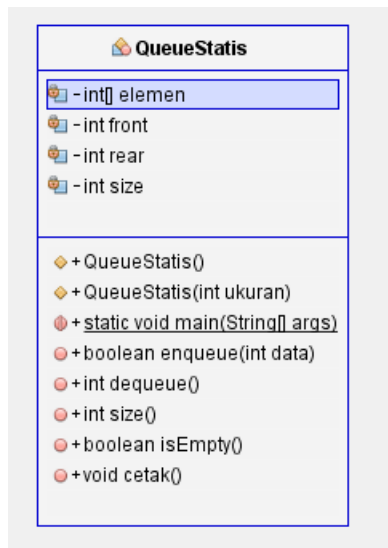


NAMA : Johanes Yogtan Wicaksono Rahaharja

NIM : 215314105

Program Studi INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA

A. DIAGRAM UML



B. METODE

1. Queue()

- Screenshot Listing Program

```
public QueueStatis() {
    elemen = new int[10];
}
```

- Penjelasan

Konstruktur kelas tanpa parameter, berfungsi untuk memberi nilai default apabila tidak diisi. Disitu dideklarasikan array elemen yang berisi 10

2. Queue(int ukuran)

- Screenshot Listing Program

```
public QueueStatis(int ukuran) {
    elemen = new int[ukuran];
}
```

- Penjelasan

Konstruktur kelas yang ada parameter, parameter ukuran berfungsi untuk atribut array yang akan diisi dengan array yang telah diisi dalam parameter, sedangkan untuk atribut.

3. Enqueue()

- Screenshot Listing Program

```

public boolean enqueue(int data) {
    if (size < elemen.length) {
        elemen[rear] = data;
        if (rear == elemen.length - 1) {
            rear = 0;
        } else {
            rear++;
        }
        size++;
        return true;
    }
    return false;
}

```

- Penjelasan

Metode bertipe boolean digunakan untuk menambahkan sebuah data ke dalam antrian (array antrian). Method ini pertama – tama akan menjalankan percabangan untuk menambahkan data di array, dengan variabel size yang dinamis sesuai dengan inputan user itu juga yang akan dijalankan, dan selama percabangan berlangsung parameter data akan masuk ke array elemen dengan pertambahan sizenya, dilanjutkan lagi di dalam perbangan ini ada percabangan dengan variabel size sama dengan panjang array kurang 1, maka varianel rear = 0, jika salah rear akan terus ditambah 1, ketika dalam percabangan selesai size akan ditambah 1.

4. Dequeue()

- Screenshot Listing Program

```

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

```

- Penjelasan

Method yang bertipe int yang digunakan untuk mengeluarkan sebuah data dari dalam antrian(array antrian).Method ini pertama – tama akan menjalankan percabangan untuk mengeluarkan data di array, jika panjang array tidak sama dengan 0 maka data array di indek saat itu akan disimpan ke variabel hapus, dalam percabangan array tersebut dibuat lagi percabngan jika variabel front sama dengan panjang array dikurangi 1 maka front akan disimpan 0, jika tidak maka front ditambah 1, ketika dalam percabangan selesai maka aizenya dikurangi 1.

5. Size()

- Screenshot Listing Program

```

public int size() {
    return size;
}

```

- Penjelasan
Metode yang digunakan untuk mengembalikan nilai atribut size, variabel ini untuk melihat jumlah data array secara visualisasi tidak seperti indeks

6. isEmpty

- Screenshot Listing Program

```

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

```

- Penjelasan
Metode yang digunakan untuk mengecek apakah stack tersebut kosong atau masih memiliki isi. Jika kosong maka dijalankan atau true, jika tidak akan bernilai false atau tidak dijalankan

7. Cetak

- Screenshot Listing Program

```

public void cetak() {
    System.out.print("");
    for (int i = front; i < rear; i++) {
        System.out.print(element[i] + " ");
    }
    System.out.println("");
}

```

- Penjelasan
Sebuah method yang digunakan untuk mencetak semua data yang terdapat pada antrian. Method ini pertama – tama akan memberi sebuah space, setelah tu menjalankan percaabngan dengan front yang bersifat dinamis, front di dalam antrian dialiaskan sebagai data pertama yang ada dalam antirian dan jika lebih kecil dari rear yang dialiaskan sebagai data terakhir dalam antrian akan mberulang terus hingga sesuai dengan kondisi. Selanjutnya akan mencetak data di array elemen sesuai dengan I yang dinamis tadiya hingga memberi space.

Implementasi

8. Screenshot kelas main

```
1
2 package Modul7_Stack;
3 import java.util.NoSuchElementException;
4 public class QueueStatis {
5     public static void main(String[] args) {
6         QueueStatis antrian = new QueueStatis(7);
7         antrian.enqueue(26);
8         antrian.cetak();
9         antrian.enqueue(15);
10        antrian.cetak();
11        antrian.enqueue(8);
12        antrian.cetak();
13        antrian.enqueue(14);
14        antrian.cetak();
15        System.out.println("-----");
16        System.out.print("Antrian keluar : ");
17        System.out.println(antrian.dequeue());
18        antrian.cetak();
19        System.out.print("Antrian keluar : ");
20        System.out.println(antrian.dequeue());
21        antrian.cetak();
22        System.out.print("Antrian keluar : ");
23        System.out.println(antrian.dequeue());
24        antrian.cetak();
25        System.out.print("Antrian keluar : ");
26        System.out.println(antrian.dequeue());
27        antrian.cetak();
28        System.out.print("Antrian keluar : ");
29        System.out.println(antrian.dequeue());
30        antrian.cetak();
31    }
32
33    private int[] elemen;
34    private int front;
35    private int rear;
36    private int size;
37
38    public QueueStatis() {
39        elemen = new int[10];
40    }
41    public QueueStatis(int ukuran) {
42        elemen = new int[ukuran];
43    }
44 }
```

```

46     public boolean enqueue(int data) {
47         if (size < elemen.length) {
48             elemen[rear] = data;
49             if (rear == elemen.length - 1) {
50                 rear = 0;
51             } else {
52                 rear++;
53             }
54             size++;
55             return true;
56         }
57         return false;
58     }
59     public int dequeue() {
60         if (elemen.length != 0) {
61             int hapus = elemen[front];
62             if (front == elemen.length - 1) {
63                 front = 0;
64             } else {
65                 front++;
66             }
67             size--;
68             return hapus;
69         }
70         throw new NoSuchElementException();
71     }
72
73     public int size() {
74         return size;
75     }
76
77     public boolean isEmpty() {
78         if (size == 0) {
79             return true;
80         } else {
81             return false;
82         }
83     }
84     public void cetak() {
85
86         System.out.print("");
87         for (int i = front; i < rear; i++) {
88
89             System.out.print(elemen[i] + " ");
90         }
91         System.out.println("");
92     }
93 }

```

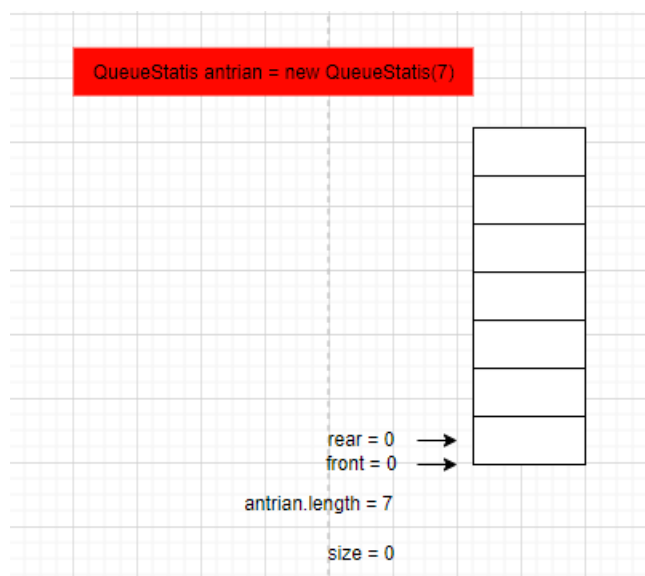
9. Screenshot output

```
run:
26
26 15
26 15 8
26 15 8 14
-----
Antrian keluar : 26
15 8 14
Antrian keluar : 15
8 14
Antrian keluar : 8
14
Antrian keluar : 14

Antrian keluar : 0
```

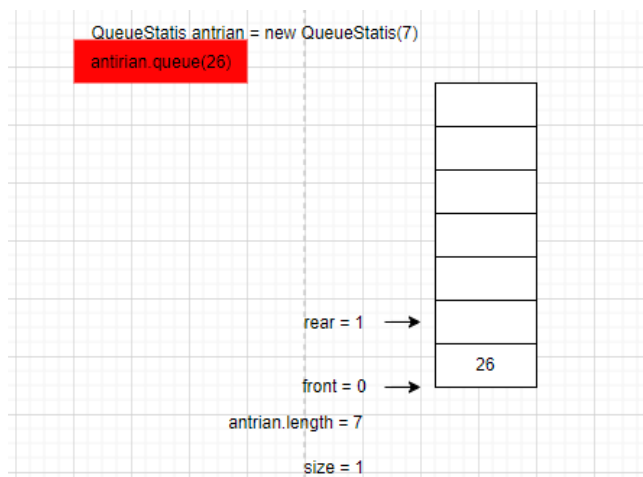
10. Ilustrasi dari Queue dan Penjelasan ilustrasi/program

1. Gambar 1



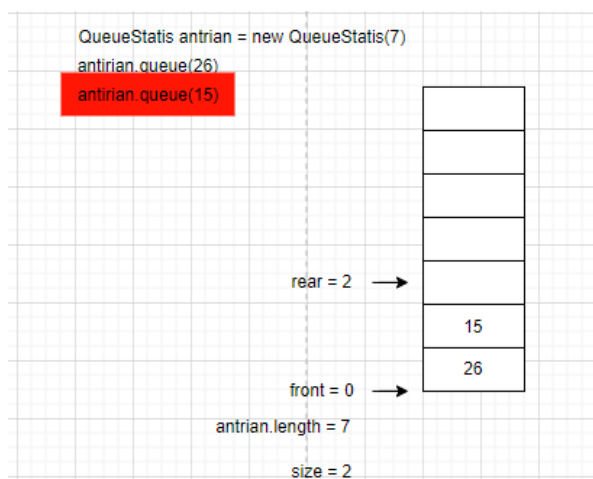
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk membuat jumlah array dari Queue dalam antrian, yaitu berjumlah 7 yang diilustrasikan dalam bentuk table. Selanjutnya ada rear sebagai elemen belakang yang ketika ada penambahan data antrian akan melakukan penambahan sedangkan di front tetap, namun ketika ada pengeluaran data akan melakukan penambahan di front sedangkan di rear tetap. Selanjutnya ada `antrian.length` sebagai deklarasi panjangnya array dan `size` sebagai banyaknya data di array.

2. Gambar 2



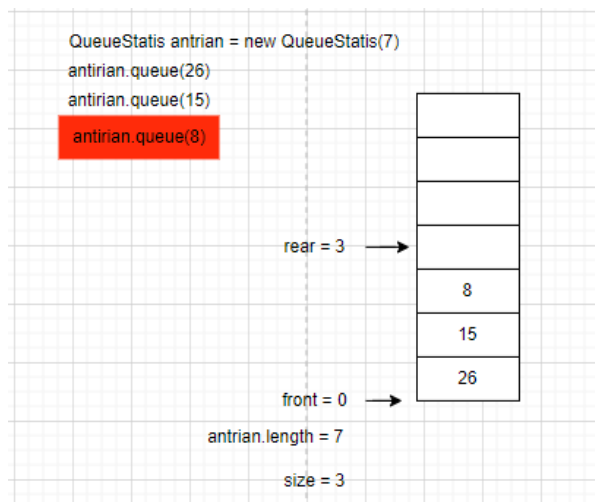
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan pertambahan data dalam array yaitu 26 yang diilustrasikan di dalam tabel di indeks 0, oleh karena itu rearnya menjadi 1 sedangkan frontnya tetap dan size datanya menjadi 2

3. Gambar 3



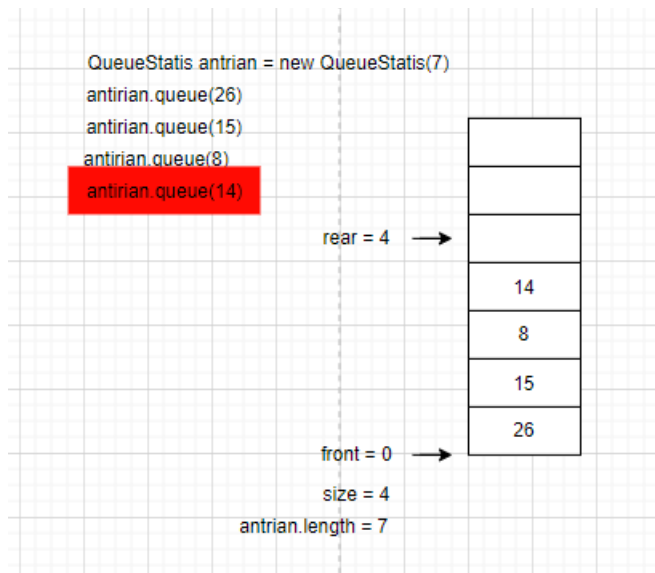
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan pertambahan data dalam array yaitu 15 yang diilustrasikan di dalam tabel di indeks 1, oleh karena itu rearnya menjadi 2 sedangkan frontnya tetap dan size datanya menjadi 2

4. Gambar 4



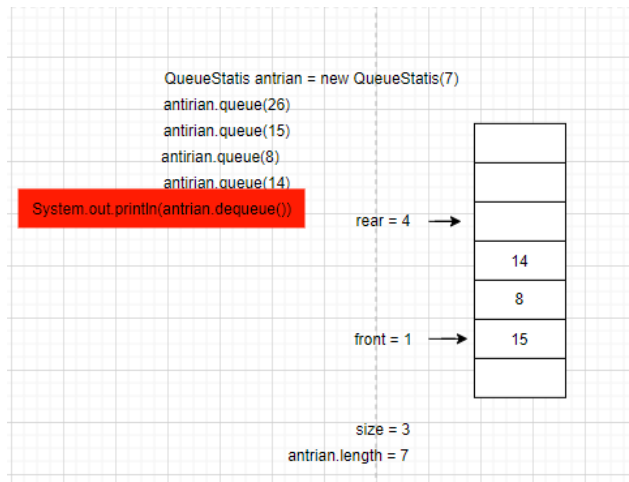
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan penambahan data dalam array yaitu 8 yang diilustrasikan di dalam tabel di indeks 2, oleh karena itu rearnya menjadi 2 sedangkan frontnya tetap dan size datanya menjadi 3

5. Gambar 5



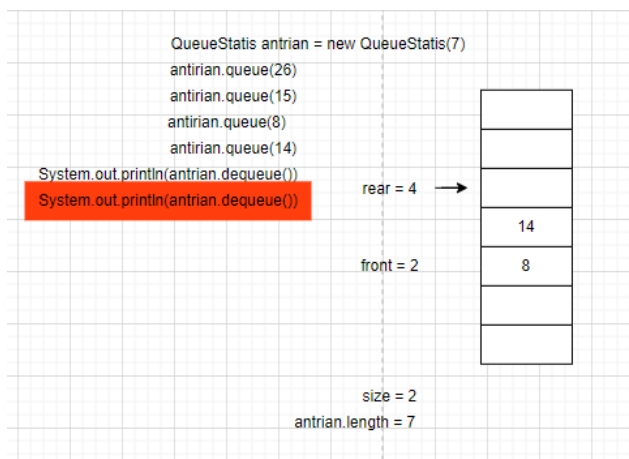
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan penambahan data dalam array yaitu 14 yang diilustrasikan di dalam tabel di indeks 3, oleh karena itu rearnya menjadi 4 sedangkan frontnya tetap dan size datanya menjadi 4

6. Gambar 6



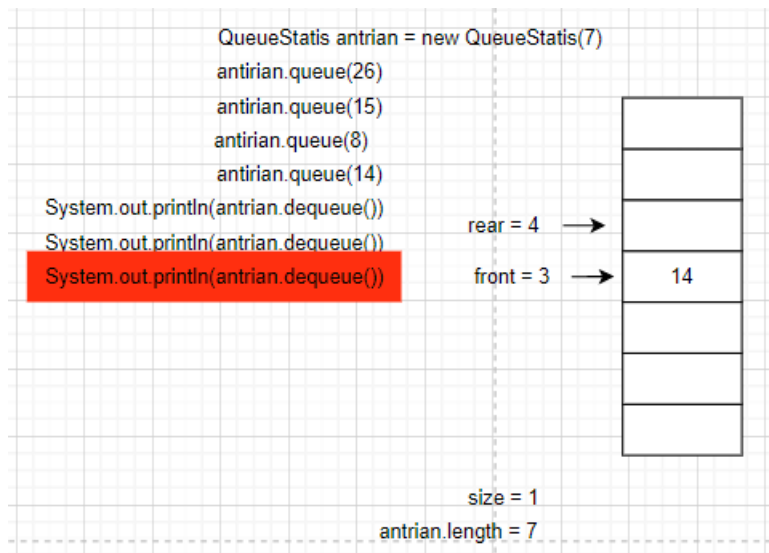
Pada kondisi sebelumnya melakukan penambahan data dan kali ini akan melakukan pengeluaran data dalam antrian seperti dalam kalimat berwarna merah mempunyai fungsi untuk melakukan pengeluaran data dalam array yaitu 26 yang diilustrasikan di dalam tabel di indeks 0 dihilangkan, oleh karena itu frontnya menjadi 1 sedangkan rearnya tetap dan size datanya menjadi 3

7. Gambar 7



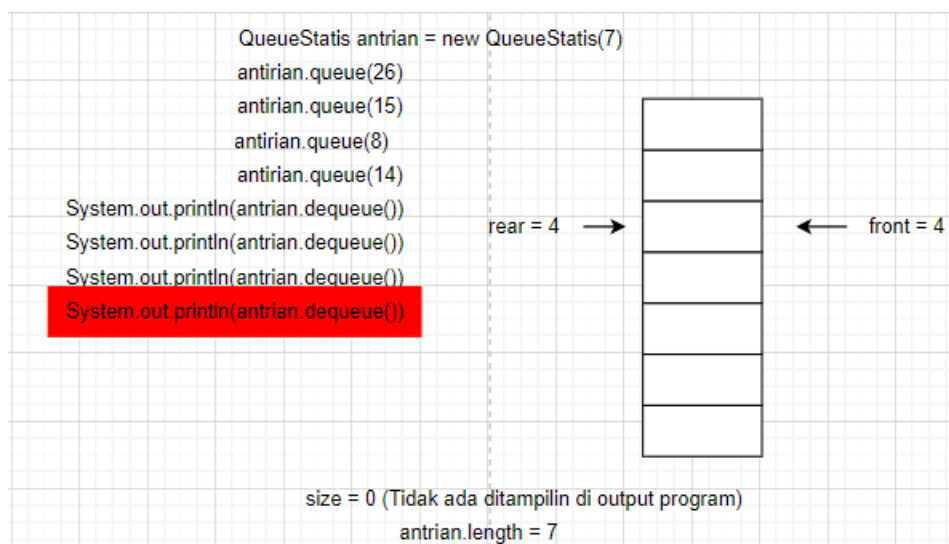
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan pengeluaran data dalam array yaitu 15 yang diilustrasikan di dalam tabel di indeks 1 dihilangkan, oleh karena itu frontnya menjadi 2 sedangkan rearnya tetap dan size datanya menjadi 2

8. Gambar 8



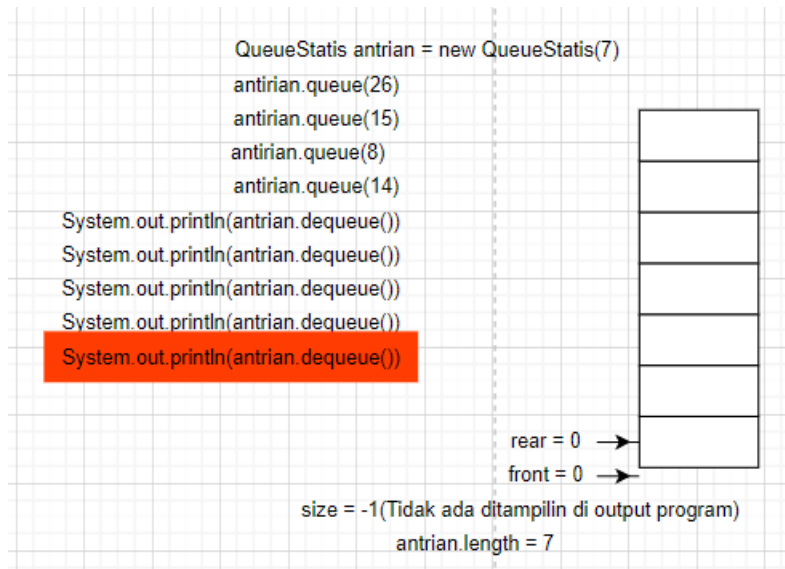
Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan pengeluaran data dalam array yaitu 8 yang diilustrasikan di dalam tabel di indeks 2 dihilangkan, oleh karena itu frontnya menjadi 4 sedangkan rearnya tetap dan size datanya menjadi 1

9. Gambar 9



Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan pengeluaran data dalam array yaitu 14 yang diilustrasikan di dalam tabel di indeks 3 dihilangkan, oleh karena itu frontnya menjadi 5 sedangkan rearnya tetap dan size datanya menjadi 0

10. Gambar 10



Pada kalimat berwarna merah tersebut mempunyai fungsi untuk melakukan pengeluaran data dalam array, karena data dalam array sudah kosong jadi yang dikeluarkan tidak ada oleh karena itu frontnya menjadi 6 sedangkan rearnya tetap dan size datanya menjadi -1, tetap dijalankan karena kondisinya adalah $<$ bukannya \leq