

PRAKTIKUM PEMROGRAMAN 1

TUGAS PERTEMUAN 9

Nama : Muhammad Faris Fathur Rohman

NRP : 223040126

Kelas : D

Link github : https://github.com/Riss27/PP1_223040126_Pertemuan9.git

Latihan 1 dan 2

```
1 public class StrukturQueue {
2     private Node FRONT;
3     private Node REAR;
4
5     // enqueue
6     public void enqueue(int data) {
7         Node newNode = new Node(data);
8         if (isEmpty()) {
9             FRONT = newNode;
10            REAR = newNode;
11        } else {
12            REAR.setNext(newNode);
13            REAR = newNode;
14        }
15    }
16    /*
17     Method ini bertujuan untuk menambahkan data baru ke dalam antrian. Jika antrian masih kosong, maka
18     data baru akan menjadi elemen pertama (FRONT) dan juga elemen terakhir (REAR) dalam antrian. jika antrian
19     sudah berisi elemen, data baru akan ditambahkan di belakang (REAR) antrian, dan REAR akan diperbarui menjadi elemen baru.
20     Parameter 'data' adalah nilai yang akan disimpan dalam elemen baru yang akan ditambahkan ke dalam antrian.
21     */
}
```

```

23 // dequeue
    Codeium: Refactor | Explain
24 public void dequeue() {
25     Node temp;
26     if (FRONT != null) {
27         if (FRONT == REAR) {
28             FRONT = null;
29             REAR = null;
30         } else {
31             temp = FRONT;
32             FRONT = FRONT.getNext();
33             temp = null;
34         }
35     }
36 }
37 /*
38 Method ini berfungsi untuk menghapus elemen dari depan (FRONT) antrian.
39 Jika antrian tidak kosong, maka elemen pertama (FRONT) akan dihapus.
40 Jika setelah penghapusan, antrian hanya memiliki satu elemen, maka FRONT dan REAR akan diatur menjadi null, menandakan antrian kosong.
41 Namun, jika antrian memiliki lebih dari satu elemen, FRONT akan dipindahkan ke elemen berikutnya, dan elemen pertama sebelumnya akan dihapus.
42 Variabel 'temp' digunakan untuk menyimpan referensi ke elemen pertama (FRONT) sebelum dihapus.
43 */

```

```

    Codeium: Refactor | Explain | Generate Java
83 public boolean isEmpty() {
84     return FRONT == null;
85 }
86 }
87

```

Latihan 3

```
59 // size
Codeium: Refactor | Explain
60 public int size() {
61     int size = 0;
62     Node curNode = FRONT;
63     while (curNode != null) {
64         size++;
65         curNode = curNode.getNext();
66     }
67     return size;
68 }
69 /*
70     Menghitung jumlah elemen dalam antrian dengan mengiterasi dari FRONT hingga ke elemen terakhir.
71     Setiap kali menemukan node, variabel 'size' akan ditambahkan satu.
72     Jumlah total elemen kemudian dikembalikan.
73 */
74
Codeium: Refactor | Explain | Generate Javadoc
75 public int front() {
76     return FRONT != null ? FRONT.getData() : -1;
77 }
78 /*
79     Mendapatkan nilai data dari elemen depan (FRONT) antrian tanpa menghapusnya.
80     Mengembalikan nilai data dari FRONT jika antrian tidak kosong, dan -1 jika kosong.
81 */
```

Latihan 4

```
1 public class QueueMain {  
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc  
2     public static void main(String[] args) {  
3         StrukturQueue queue = new StrukturQueue();  
4  
5         // sebelum enqueue 3x  
6         System.out.println(x: "### sebelum enqueue 3x ###");  
7         System.out.println("size : " + queue.size());  
8         System.out.println("IsEmpty : " + queue.isEmpty());  
9         System.out.println();  
10  
11        // Enqueue 3x  
12        System.out.println(x: "### Enqueue 3x ###");  
13        queue.enqueue(data:3);  
14        queue.enqueue(data:2);  
15        queue.enqueue(data:1);  
16        System.out.println("size : " + queue.size());  
17        System.out.println("IsEmpty : " + queue.isEmpty());  
18        System.out.println("Front : " + queue.front());  
    }
```

```
[Running] cd "d:\UNPAS\SEMESTER 4\PRAKTIKUM PEMROGRAMAN 1\PP1_223040126_Pertemuan9\" && javac QueueMain.java && java QueueMain
```

```
### sebelum enqueue 3x ###
```

```
size : 0
```

```
IsEmpty : true
```

```
### Enqueue 3x ###
```

```
size : 3
```

```
IsEmpty : false
```

```
Front : 3
```

Latihan 5

```
1  public class QueueMain2 {
2      public static void main(String[] args) {
3          StrukturQueue queue = new StrukturQueue();
4
5          // sebelum enqueue 4x
6          System.out.println(x: "### Sebelum Enqueue 4x ###");
7          System.out.println("size : " + queue.size());
8          System.out.println("IsEmpty : " + queue.isEmpty());
9          System.out.print(s: "Elemen Queue : ");
10         queue.displayElements();
11         System.out.println();
12
13         // Enqueue 4x
14         System.out.println(x: "### Enqueue 4x ###");
15         queue.enqueue(data: 2);
16         queue.enqueue(data: 7);
17         queue.enqueue(data: 6);
18         queue.enqueue(data: 1);
19         System.out.println("size : " + queue.size());
20         System.out.println("IsEmpty : " + queue.isEmpty());
21         System.out.print(s: "Elemen Queue : ");
22         queue.displayElements();
23         System.out.println("Front : " + queue.front());
24
25         // menampilkan semua elemen
26         Codeium: Refactor | Explain
27         public void displayElements() {
28             Node curNode = FRONT;
29             if (curNode == null) {
30                 System.out.println(x: "Queue Kosong");
31             } else {
32                 while (curNode != null) {
33                     System.out.print(curNode.getData() + " ");
34                     curNode = curNode.getNext();
35                 }
36                 System.out.println();
37             }
38         }
39     }
40 }
```

```
[Running] cd "d:\UNPAS\SEMESTER 4\PRAKTIKUM PEMROGRAMAN 1\PP1_223040126_Pertemuan9\" && javac QueueMain2.java && java QueueMain2  
### Sebelum Enqueue 4x ###  
size : 0  
IsEmpty : true  
Elemen Queue : Queue Kosong  
  
### Enqueue 4x ###  
size : 4  
IsEmpty : false  
Elemen Queue : 2 7 6 1  
Front : 2
```

Tugas

Latihan 4

```
20 // Dequeue
21 System.out.println(x: "\n### Dequeue ###");
22 queue.dequeue();
23 System.out.println("size : " + queue.size());
24 System.out.println("IsEmpty : " + queue.isEmpty());
25 System.out.println("Front : " + queue.front());
```

```
### Dequeue ###
size : 2
IsEmpty : false
Front : 2
```

Latihan 5

```
25 // Dequeue
26 System.out.println(x: "\n### Dequeue ###");
27 queue.dequeue();
28 System.out.println("size : " + queue.size());
29 System.out.println("IsEmpty : " + queue.isEmpty());
30 System.out.print(s: "Elemen Queue : ");
31 queue.displayElements();
32 System.out.println("Front : " + queue.front());
```

```
### Dequeue ###
size : 3
IsEmpty : false
Elemen Queue : 7 6 1
Front : 7
```