

**Worksheet pertemuan 8 - 2**  
**Algoritma dan Struktur Data**  
**Queue Dinamis**

**NIM: 20523164**

**Nama: Fajrun Shubhi**

**A. Membuat Folder Untuk Menyimpan Hasil Praktikum**

1. Siapkan folder kosong dengan nama menggunakan NIM masing-masing. Jika folder NIM pada pertemuan sebelumnya mau dimanfaatkan, jangan lupa pindahkan dulu isinya ke folder lain sebagai arsip.
2. Folder ini akan dijadikan tempat untuk menyimpan semua pdf dari worksheet ini beserta file praktikum lainnya.

**B. Membuat class Queue**

1. Gunakan file *LinkedList.java* yang sudah Anda buat pada pertemuan sebelumnya dan letakkan dalam satu *package* dengan file *Queue.java* yang akan dibuat.
2. Silakan buat sebuah class dengan nama **Queue**
3. Kemudian salin tempel kode program di bawah ini

```
package queuedinamis;

/**
 *
 * @param <E>
 */
public class Queue<E> extends LinkedList {

    public Queue(){

    }

    public Object peek(){
        return getFirst();
    }

}
```

**C. Membuat dan Menjalankan Main Method**

1. Silakan buat sebuah main method class dengan nama **TestQueueDinamis**
2. Kemudian salin tempel kode program di bawah ini

```
package queuedinamis;

public class TestQueueDinamis {
    public static void main(String[] args) {
        Queue<String> mhs = new Queue<>();

        mhs.add("Agus");
    }
}
```

```

        mhs.add("Sinta");
        mhs.add("Bambang");

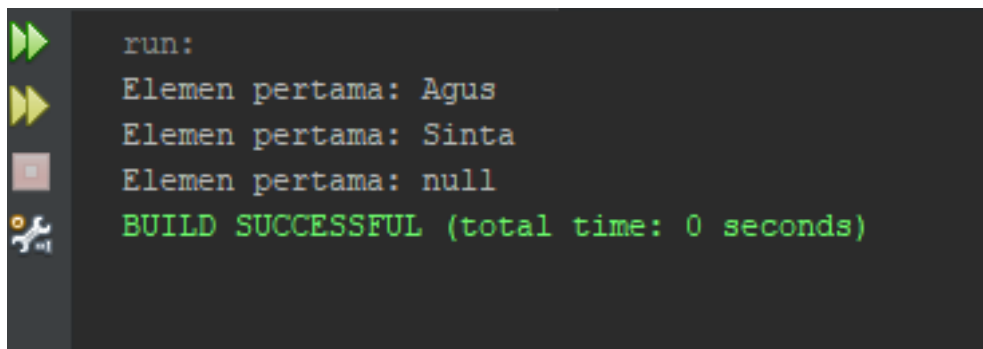
        System.out.println("Elemen pertama: "+mhs.peek());

        mhs.remove();
        System.out.println("Elemen pertama: "+mhs.peek());

        mhs.clear();
        System.out.println("Elemen pertama: "+mhs.peek());
    }
}

```

3. Jalankan *main method* **TestQueueDinamis** dan hasil tangkapan layar keluaran dari program silakan letakkan di bawah ini



```

run:
Elemen pertama: Agus
Elemen pertama: Sinta
Elemen pertama: null
BUILD SUCCESSFUL (total time: 0 seconds)

```

#### D. Membuat Class QueueCircular

1. Silakan buat sebuah class dengan nama **QueueCircular**
2. Kemudian salin tempel kode program di bawah ini!

```

package queuecircular;

public class QueueCircular {
    private Object [] queueCirc;
    private int head;
    private int tail;
    private int maxsize;

    public QueueCircular(int size){
        maxsize = size;
        queueCirc = new Object[maxsize];
        head = 0;
        tail = queueCirc.length - 1;
    }

    public boolean isEmpty(){
        return ((tail+1) % queueCirc.length == head);
    }
}

```

```

    }

    public boolean isFull(){
        if ((tail+2) % queueCirc.length == head){
            return true;
        }
        if(tail+1 == queueCirc.length && head != 0){
            return true;
        }
        return false;
    }

    public void enqueue(Object x){
        if (!isFull()) {
            tail = ((tail+1) % queueCirc.length);
            queueCirc[tail] = x;
            System.out.println("Enqueue -> "+x);
        }else{
            System.out.println("Queue is full");
        }
    }

    public Object dequeue(){
        Object temp = null;
        if( !isEmpty() ) {
            temp = queueCirc[head];
            head = ((head+1) % queueCirc.length);
        }
        return temp;
    }
}

```

### E. Membuat Main Method TestQueueCircular

1. Silakan buat sebuah main method class dengan nama **TestQueueCircular**
2. Kemudian salin tempel kode program di bawah ini

```

package queuecircular;

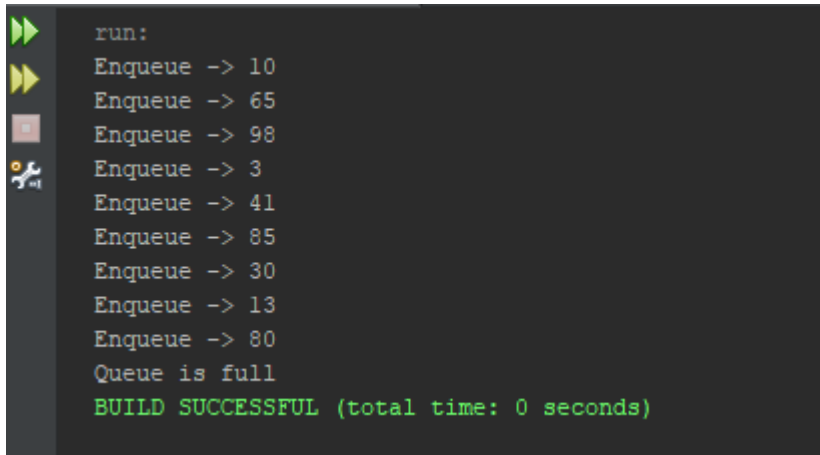
public class TesQueueCircular {
    public static void main(String[] args) {
        QueueCircular q = new QueueCircular(10);
        int i;
        Integer j;

        for(i=0;i<10;i++){
            j = (int) (Math.random()*100);
            q.enqueue(j);
        }
    }
}

```

```
    }  
}  
}
```

3. Jalankan file *TestQueueCircular.java* dan hasil tangkapan layar keluaran dari program silakan letakkan di bawah ini



```
run:  
Enqueue -> 10  
Enqueue -> 65  
Enqueue -> 98  
Enqueue -> 3  
Enqueue -> 41  
Enqueue -> 85  
Enqueue -> 30  
Enqueue -> 13  
Enqueue -> 80  
Queue is full  
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Buat sebuah *method* tambahan di dalam class **QueueCircular** yang digunakan untuk menampilkan isi dari queue mulai dari elemen pertama yang dimasukkan hingga elemen terakhir. Salin tempel kode program *method* yang Anda buat di bagian di bawah ini!

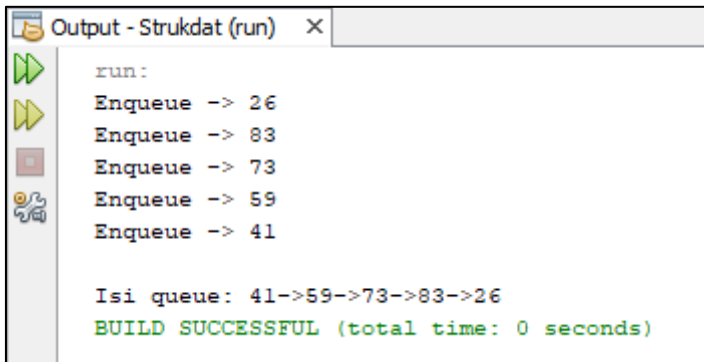
```
public void Display (){  
    if (isEmpty()){  
        System.out.println("Queue is Empty");  
    }  
    else{  
        System.out.print("isi queue adalah: ");  
        for (int i = queueCirc.length - 2; i >= 0; i--){  
            System.out.print(" ==>> " + queueCirc[i]);  
        }  
        System.out.println();  
    }  
}
```

```

public void Display () {
    if (isEmpty()) {
        System.out.println("Queue is Empty");
    }
    else {
        System.out.print("isi queue adalah: ");
        for (int i = queueCirc.length - 2; i >= 0; i--) {
            System.out.print(" ==>> " + queueCirc[i]);
        }
        System.out.println();
    }
}

```

\*Contoh tampilan isi Queue



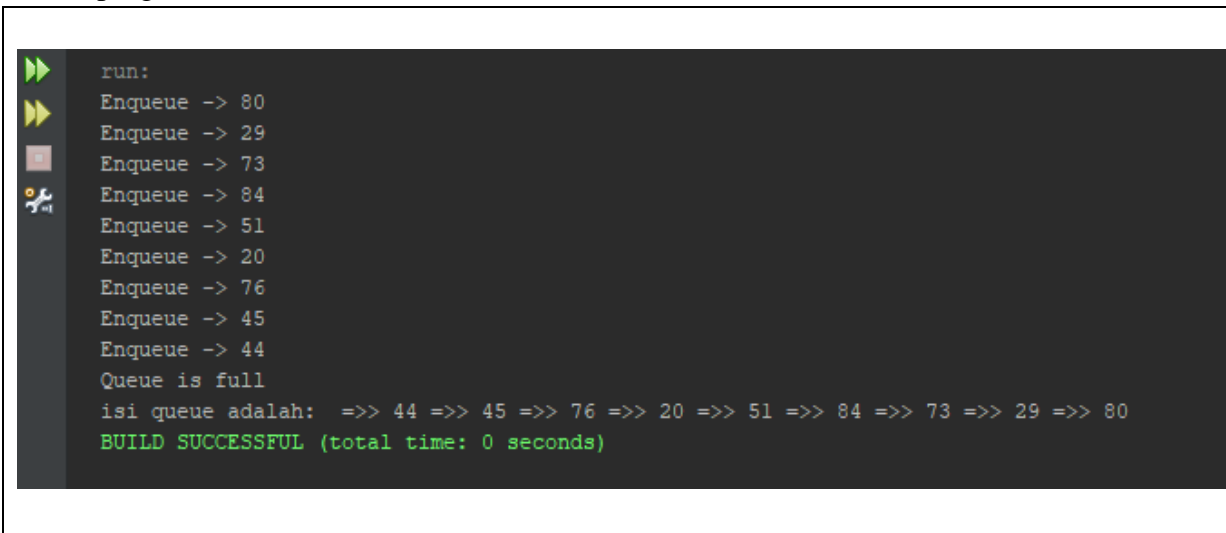
```

Output - Strukdat (run) X
run:
Enqueue -> 26
Enqueue -> 83
Enqueue -> 73
Enqueue -> 59
Enqueue -> 41

Isi queue: 41->59->73->83->26
BUILD SUCCESSFUL (total time: 0 seconds)

```

5. Jalankan file *TestQueueCircular.java* sekali lagi dan hasil tangkapan layar keluaran dari program silakan letakkan di bawah ini



```

run:
Enqueue -> 80
Enqueue -> 29
Enqueue -> 73
Enqueue -> 84
Enqueue -> 51
Enqueue -> 20
Enqueue -> 76
Enqueue -> 45
Enqueue -> 44
Queue is full
isi queue adalah: ==>> 44 ==>> 45 ==>> 76 ==>> 20 ==>> 51 ==>> 84 ==>> 73 ==>> 29 ==>> 80
BUILD SUCCESSFUL (total time: 0 seconds)

```

\*Catatan

- Jangan lupa simpan juga file worksheet ini (yang sudah diisi) sebagai file pdf di folder NIM anda.
- Sertakan juga file-file di bawah ini di dalam folder yang Anda gunakan
  - [LinkedList.java](#)

- Queue.java
- TestQueueDinamis.java
- QueueCircular.java
- TestQueueCircular.java
- **Kompres folder ini sebagai file ZIP kemudian kumpulkan di classroom atau ruang pengumpulan lain di kelas masing-masing.**