

Agiftsany Azhar
152011513020/D3-Sistem Informasi

-----Main-----

```
package tugas4_152011513020;
```

```
public class Tugas4_152011513020 {
    public static void main(String[] args) {

        System.out.println("-----");
        System.out.println("Praktikum 4 - Queue");
        System.out.println("-----");
        Queue a;

        a = new Queue(5);

        System.out.println("-----");
        System.out.println("Nomor 1 Is Empty");
        System.out.println("-----");

        System.out.println("Queue kosong? " + a.isEmpty());

        System.out.print("\n");
        System.out.println("-----");
        System.out.println("Nomor 2 Is Full");
        System.out.println("-----");

        System.out.println("Queue penuh? " + a.isFull());

        System.out.print("\n");
        System.out.println("-----");
        System.out.println("Nomor 3&4 - End Q&De Q");
        System.out.println("-----");

        a.endq(92);
        a.endq(23);
        a.endq(46);
        a.endq(51);
        a.endq(64);
        //    a.endq(53);
        a.display();
        a.deq();
        a.display();
        a.deq();
        a.display();
        a.deq();
        a.display();
        a.deq();
        a.display();

        System.out.print("\n\n");
        System.out.println("-----");
        System.out.println("Praktikum 4 - Node");
        System.out.println("-----");
    }
}
```

```

        Node x = new Node(5);
        Node y = new Node(10);
        Node z = new Node(15);

        x.print();
        y.print();
        z.print();

        x.setNext(y);
        y.setNext(z);

        System.out.println(x.getInfo());
        System.out.println(x.getNext().getInfo());
        System.out.println(x.getNext().getNext().getInfo());
    }
}

```

-----Class-----

```
package tugas4_152011513020;
```

```

public class Queue {
    int[] data;        // array yang menampung data dalam queue
    int maxSize;       // ukuran array data
    int front;         // pointer menandai posisi terdepan
    int rear;          // pointer menandai posisi terakhir

    public Queue(int max){
        this.maxSize    = max;
        this.data        = new int[this.maxSize];
        this.front       = 0;
        this.rear        = 0;
    }

    // Nomor 1
    public boolean isEmpty(){
        if (this.rear == this.front){
            return true;
        }
        else return false;
    }

    // Nomor 2
    public boolean isFull(){
        if(this.rear == maxSize){
            return true;
        }
        else return false;
    }

    // Nomor 3
    public void endq(int element){
        if (isFull() == false){
            this.data[this.rear]    = element;

```

```

        this.rear++;
    }
}

// Nomor 4
public int deq(){
    if(isEmpty() == false){
        this.front++;

        return this.data[this.front];
    }
    else return 0;
}

public void display(){
    for (int i=front; i<this.maxSize; i++) {
        System.out.print(" " + this.data[i] + " ");
    }
    System.out.print("\n\n");
}
}

```

-----Class-----

```
package tugas4_152011513020;
```

```

public class Node {
    int info;          // memuat satu data integer
    Node next;        // pointer to next node

    /**
     * constructor dengan parameter info
     * @param info
     */
    public Node(int info){
        this.info    = info;
        this.next    = null;
    }

    /**
     * mengubah nilai variable info dengan nilai tertentu yang dimasukkan
     * dari luar melalui parameter input
     * @param info
     */
    public void setInfo(int info){
        this.info    = info;
    }

    /**
     * mengubah variable pointer next menunjuk ke object tertentu sesuai nilai
     * parameter input
     * @param next
     */
    public void setNext(Node next){

```

```

        this.next    = next;
    }

    /**
     * mengambil nilai info dari sebuah node
     * mengembalikan sebuah nilai integer
     * @return
     */
    public int getInfo(){
        return this.info;
    }

    /**
     * mengambil nilai pointer next, nilainya mungkin null atau merefers
     * pada address/ alamat yang merujuk pada node lain
     * mengembalikan nilai pointer of Node
     * @return
     */
    public Node getNext(){
        return this.next;
    }

    /**
     * mencetak nilai yang termuat di dalam info
     */
    public void print(){
        System.out.println("Info    = " + this.info);
    }
}

```