# Bab 6 Queue

## Identitas

**Kajian**

Queue

**Topik**

1. Queue

**Referensi**

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms Third Edition. Cambridge: The MIT Press.
2. Sedgewick, R. (2002). Algorithm in Java: parts 1-4 Third Edition. Boston: Pearson Education, Inc.
3. P. Deitel and H. Deitel, Java How To Program 9th Edition, Prentice Hall, 2011.
4. Poo, Danny, Derek Kiong, and Swarnalatha Ashok. Object-Oriented Programming and Java, 2nd Edition. Springer, 2010.
5. Wu, C. Thomas. An Introduction to Object-Oriented Programming with Java. McGraw-Hill, 2009.

**Kompetensi Utama**

1. Mahasiswa mampu menggunakan bahasa java untuk membangun sebuah aplikasi sederhana yang digunakan untuk mengelola data dalam struktur Queue dengan menggunakan kelas object

**Lama Kegiatan Kajian**

1. Pertemuan Terbimbing : 2 x 500 menit
2. Kegiatan Mandiri : 1 x 70 menit

**Parameter Penilaian**

1. Jurnal 60%
2. Tugas Akhir 40%

**Pengumpulan**

Kumpulkan jawaban dari Jurnal tembimbing dan jurnal mandiri pada LMS yang telah disediakan

## Jurnal Terbimbing

Lengkapi kode program berikut untuk menyusun sebuah program untuk mengelola data dengan menggunakan Queue

```
import java.io.*;

class Node {
    int data;
    Node next;

    Node(int d) {
        data = d;
        next = null;
    }
}

public class QueueList {
    Node head; // head of list
    Node tail; // tail of list

    /* Insert last  */
    public int enqueue(int data) {
      Node nn = new Node(data);
        if (________) {
            head=nn;
            tail = nn;
        } else{
            tail.next = ________;
            tail = ________;
        }
        System.out.println("Node baru "+data+" ditambahkan");
        return 0;
    }
    /*     Dequeue    */
    public int dequeue() {
        // Jika list kosong
        Node current = null;
        if (________) {
            System.out.println("List kosong");
            return 0;
        } else if (________){
            head = null;
            tail = null;
        } else{
            current = ________;
            head = current.next;
            current.next = ________;
        }
        System.out.println("Data diambil "+current.data);
        return 0;
    }
```

```
    public void printList() {
        Node current = ________;
        System.out.print("Daftar antrian : ");

        while (current != ________) {
            System.out.print(current.data + " ");
            current = ________;
        }
        System.out.println("");
    }

    public static void main(String[] args) {
        QueueList sll = new QueueList();
        sll.runThis();
    }
    void runThis(){
        enqueue(1);
        enqueue(5);
        enqueue(3);
        enqueue(2);
        enqueue(7);
        enqueue(6);
        enqueue(9);
        enqueue(8);
        printList();

        dequeue();
        printList();
        dequeue();
        printList();
        dequeue();
        printList();

    }
}
```

## Jurnal Mandiri

Modifikasi program diatas, sehingga data yang dikelola bukan dalam bentuk data String saja, namun untuk mengelola kelas berikut ini

```
import java.io.*;
class Pasien{
    int noUrut;
    String nama;

    Pasien(int nu, String nama){
        this.noUrut = nu;
        this.nama=nama;
    }
    String info(){
        return noUrut+" "+nama;
    }
    int getNoUrut(){
        return noUrut;
    }

}
```

```
class Node {
    Pasien data;
    Node next;

    Node(Pasien d)
    {
        data = d;
        next = null;
    }
}

public class QueueListPasien {
    Node head; // head of list
    Node tail; // tail of list

    /* Tambah antrian  */
    public int enqueue(String nama) {
        //Pastikan bahwa no urut pasien akan selalu terurut, sehingga
       // Jika Queue kosong, no urut dari 1
       // Jika Queue tidak kosong, no urut diambil dari no urut last node +1

    }
    /*  Ambil Urutan */
    public int dequeue() {
           //Pengambilan node urutan dari yang paling ujung
    }
```

```
/* jika no antrian depan tidak ada, maka ambil antrian yang ada*/
    public int ambilAntrianTengah(int noUrut) {
        // Jika list kosong
        Node current = null;
        Node prev = null;
        if (head == null) {
            //Jika List kosong
            return 0;
        } else if ((head.data.getNoUrut() == noUrut) && (head==tail)){
            //jika hanya ada 1 node
                return 0;
            } else{
                //Jika pasien yang datang sesuai no urut tidak ada, maka akan diambil nextnya sampai ketemu
                //Lakukan pemanggilan pasien sesuai urutan hingga ketemu atau hingga akhir Queue
                return 0;
            }

    public void printList() {
           Node current = head;
           System.out.println("Daftar antrian : ");
           while (current != null) {
                System.out.println(current.data.info());
                current = current.next;
           }
           System.out.println("");
    }

     public static void main(String[] args) {
           QueueListPasien sll = new QueueListPasien();
           sll.runThis();
     }
```

```
void runThis(){
                enqueue("Gandalf The Grey");
                enqueue("Aragorn");
                enqueue("Legolas");
                enqueue("Gimli");
                enqueue("Boromir");
                enqueue("Frodo Baggins");
                enqueue("Samwise Gamgee");
                enqueue("Meri");
                enqueue("Pippin");
                printList();

                ambilAntrianTengah(5);
                printList();
                dequeue();
                printList();
                enqueue("Gandalf The White");
                printList();
            }
}
```