

*Modul***4****STACK AND QUEUE****Tujuan**

Mahasiswa diharapkan dapat menerapkan penggunaan stack dan queue pada suatu kasus. Secara khusus program yang mengandung instruksi-instruksi yang menerapkan bagian dari stack dan queue. Mahasiswa juga dapat menggunakan penerapan pop dan push pada stack dan queue. Di akhir praktikum ini mahasiswa dapat:

- Menjelaskan konsep dasar stack and queue
- Memahami bagaimana menggunakan pop dan push pada stack
- Memahami bagaimana menggunakan pop dan push pada queue

**Pendahuluan**

Praktikum ini mengasumsikan bahwa mahasiswa telah dapat mengoperasikan Netbeans. Mahasiswa juga diharapkan sudah memahami dengan baik materi-materi yang telah diberikan sebelumnya (pemrograman dasar I). Agar mahasiswa dapat mencapai tujuan dalam pertemuan praktikum yang pertama ini, mahasiswa harus memiliki pengetahuan (minimal telah membaca) penggunaan konsep dasar stack dan queue Bagian dari stack dan queue yaitu: pop dan push. Pemahaman tentang penggunaan array dan linkedlist juga dibutuhkan dalam praktikum ini.

**Proses**

Praktikan membaca buku/diktat yang dilakukan selama 20 menit termasuk membuat ringkasan penting, kemudian berdiskusi sesuai dengan panduan aktifitas yang dilakukan selama 20 menit, sedangkan waktu untuk mengerjakan *project* (membuat program/*coding*) secara individual harus diselesaikan di dalam laboratorium dalam waktu 30 menit. Berikutnya untuk sesi latihan, ada soal tentang pengembangan program yang juga harus diselesaikan di laboratorium dalam waktu 50 menit. Terakhir adalah bagian tugas, yaitu *project* yang dikerjakan di rumah dan wajib dikumpulkan pada pertemuan berikutnya.

**Aktifitas**

1. Mahasiswa membaca buku ajar (jika ada)/diktat kuliah/materi dari sumber lain tentang konsep dasar stack dan queue sebagai materi bab pertama. Temukan bagian penting dalam topik stack dan queue, kemudian tulis sebagai ringkasan hasil belajar.
2. Mahasiswa berdiskusi tentang definisi stack dan queue, tujuan adanya push dan pop sebagai bahan diskusi

- a. Lengkapilah bagian yang kosong pada class LinkedListNode yang menerapkan konsep stack dan queue sesuai dengan perintah / command yang tertera.

```
public class LinkedListNode{
    LinkedListNode next;
    LinkedListNode prev;
    int data;

    /* Constructor
     * set this.data into new_data
     * set this.prev into null
     * set this.next into null
     */
    LinkedListNode(int new_data){
        .....
        .....
        .....
    }

    /* set this.prev into other
     * if other is not null, set other.next into this
     */
    void set_prev(LinkedListNode other){
        .....
        if( .....){
            .....
        }
    }

    /* set this.next into other
     * if other is not null, set other.prev into this
     */
    void set_next(LinkedListNode other){
        .....
        if( .....){
            .....
        }
    }
}
```

- b. Lengkapilah bagian yang kosong pada class LinkedList di bawah ini sesuai dengan perintah / command yang tertera.

```
public class LinkedList{
    LinkedListNode head;
    LinkedListNode tail;

    LinkedList(){
        this.head = null;
        this.tail = null;
    }

    /* First set a Node named current into head
     * while current is not null, print current.data, set current
     = current.next
     * print end of line
```

```

        */
        void print() {
            .....
            while (.....) {
                System.out.print (.....);
                .....;
            }
            System.out.println();
        }

        /* if LinkedList is empty, set new_node as head and tail
        * if LinkedList is not empty, set tail.next into new_node,
        set new_node.prev into tail, and make new_node a new tail
        */
        void push(LinkedListNode new_node) {
            if (.....) {
                .....
            }
            .....
        }
    }

    /* declare a node named taken
    * if LinkedList is empty, set taken into null
    * if linkedList consists only one node, set taken = head, set
    both head and tail into null
    * if linkedList consists of two or more nodes, set taken =
    head, set head.next into new head, cut off all the link between
    taken and head
    * return taken
    */
    LinkedListNode qpop() {
        LinkedListNode taken;
        if (.....) {
            .....
        }
        .....
    }
    return taken;
}

/* declare a node named taken
* if LinkedList is empty, set taken into null
* if linkedList consists only one node, set taken = tail, set
both head and tail into null
* if linkedList consists of two or more nodes, set taken =
tail, set tail.prev into new tail, cut off all the link between
taken and tail
* return taken
*/
LinkedListNode spop() {
    LinkedListNode taken;
    if (.....) {
        .....
    }
}

```

```

        }else if(.....){
            .....
            .....
            .....
        }else{
            .....
            .....
            .....
            .....
        }
        return taken;
    }
}

```

- c. Simpan class `LinkedListNode.java` dan `LinkedList.java` pada satu package yang sama. Selanjutnya buat class baru pada package yang sama ketikkan class `Test.java` di bawah ini dan jalankan.

```

public class Test{
    public static void main(String Args[]){
        // Test case for linked list, queue and stack
        LinkedList a = new LinkedList();
        a.print(); // should show null
        System.out.println(a.qpop()); // should show null
        System.out.println(a.spop()); // should show null
        a.push(new LinkedListNode(5));
        a.push(new LinkedListNode(4));
        a.print(); // should show "5 4"
        System.out.println(a.spop().data); // should show "4"
        a.push(new LinkedListNode(4));
        System.out.println(a.qpop().data); // should show "5"
        a.print(); // should show "4"}
    }
}

```

- d. Setelah dijalankan akan muncul output seperti di bawah ini :

```

null
null
5 4
4
5
4

```

## Kesimpulan

Buatlah Interface penerapan stack dan queue pada program diatas

## Penutup

### Tugas

1. Buatlah sebuah aplikasi antrian sederhana menggunakan stack and queue