

Nama : Selviani Amalia Kartika

Nim : H071211003

Tugas Struktur Data A

Stack dan **Queue** merupakan fungsi yang biasanya diimplementasikan pada program, tidak terkecuali Pemrograman Java. Untuk itu, sebelum membahas program sederhana Stack dan Queue menggunakan Pemrograman Java, kita simak terlebih dahulu pengertian dan penjelasan dari kedua fungsi tersebut dibawah ini.

1. Stack

Stack merupakan perintah pengumpulan data secara linear yang menyusun data seperti tumpukan dan bersifat **Last In First Out (LIFO)** yang berarti data yang terakhir masuk adalah data yang pertama keluar.

Operasi Stack yang biasa digunakan diantaranya yaitu :

1. **Push** untuk memasukkan atau menginputkan data
2. **Pop** untuk mengeluarkan data
3. **IsFull** untuk mengetahui jika tumpukan sudah penuh
4. **IsEmpty** untuk mengetahui tumpukan yang kosong, dan
5. **Clear** untuk menghapus seluruh data atau membersihkan data.

Contoh implementasi stack dalam java

```
2. class NilaiMatkul {
3.     String nim;
4.     String nama;
5.     double nilai;
6.
7.     NilaiMatkul() {
8.     }
9. }
10.
11. class Stack {
12.     int top;
13.     NilaiMatkul[] data = new NilaiMatkul[10];
14.
15.     Stack() {
16.         for (int i = 0; i < 10; i++) {
17.             data[i] = new NilaiMatkul();
18.         }
19.     }
20.
21.     void createEmpty() {
22.         top = -1;
23.     }
24. }
```

```
25.     boolean isEmpty() {
26.         boolean hasil = false;
27.
28.         if (top == -1) {
29.             hasil = true;
30.         }
31.
32.         return hasil;
33.     }
34.
35.     boolean isFull() {
36.         boolean hasil = false;
37.
38.         if (top == data.length - 1) {
39.             hasil = true;
40.         }
41.
42.         return hasil;
43.     }
44.
45.     void push(String nim, String nama, double nilai) {
46.         if (isFull() == true) {
47.             // jika stack penuh
48.             System.out.println("Stacknya penuh bos");
49.         } else {
50.             if (isEmpty() == true) {
51.                 // jika stack kosong
52.                 top = 0;
53.                 data[0].nim = nim;
54.                 data[0].nama = nama;
55.                 data[0].nilai = nilai;
56.             } else {
57.                 // jika stack tidak kosong
58.                 top = top + 1;
59.                 data[top].nim = nim;
60.                 data[top].nama = nama;
61.                 data[top].nilai = nilai;
62.             }
63.         }
64.     }
65.
66.     void pop() {
67.         if (top == 0) {
68.             // jika stack berisi satu elemen
69.             top = -1;
70.         } else {
71.             if (top != -1) {
72.                 // jika stack tidak kosong
```

```

73.         top = top - 1;
74.     }
75. }
76. }
77.
78. void printStack() {
79.     if (top != -1) {
80.         System.out.println("-----");
81.         int i;
82.
83.         for (i = top; i >= 0; i--) {
84.             System.out.println("=====");
85.             System.out.println("elemen ke-" + (i + 1));
86.             System.out.println("nim : " + data[i].nim);
87.             System.out.println("nama : " + data[i].nama);
88.             System.out.println("nilai : " + data[i].nilai);
89.             System.out.println("-----");
90.         }
91.
92.     } else {
93.         // proses jika stacknya kosong
94.         System.out.println("stack kosong");
95.     }
96. }
97. }
98.
99. public class StactStatik {
100.     public static void main(String[] args) {
101.
102.         Stack S = new Stack();
103.
104.         S.createEmpty();
105.         S.printStack();
106.
107.         System.out.println("=====");
108.
109.         S.push("13822", "bahrunnur", 72.43);
110.         S.push("13823", "bahrunnar", 62.21);
111.         S.push("13824", "bahrunyu", 89.45);
112.
113.         S.printStack();
114.
115.         System.out.println("=====");
116.
117.         S.pop();
118.         S.pop();
119.         S.printStack();
120.

```

```

121.         System.out.println("=====");
122.
123.     }
124. }

```

125. Queue

Queue ialah Struktur Data yang mempunyai sifat **FIFO(First In First Out)** yang artinya, data yang pertama kali masuk merupakan data yang akan keluar paing awal.

Queue mempunyai beberapa fungsi operasi diantaranya yaitu :

1. EnQueue untuk Memasukkan data kedalam Antrian
2. DeQueue untuk Mengeluarkan data kedalam Antrian.
3. IsFull untuk memeriksa apakah antrian Penuh
4. IsEmpety untuk memeriksa apakah antrian Kosong
5. Clear untuk Menghapus seluruh Antrian.

Berikut ini adalah implementasi queue menggunakan bahasa pemrograman Java :

- Buat kelas dengan nama Queue

```

6. public class Queue {
7.
8.     int data[];
9.     int head = 0;
10.    int tail = -1;
11.
12.    public Queue(int size) {
13.        data = new int[size];
14.    }
15.
16.    public boolean isEmpty(){
17.        if(tail== -1){
18.            return true;
19.        }else{
20.            return false;
21.        }
22.    }
23.
24.    public boolean isFull(){
25.        if(tail==data.length-1){
26.            return true;
27.        }else{
28.            return false;
29.        }
30.    }
31.
32.    public void Enqueue(int dataBaru){
33.        if(isEmpty()){
34.            tail = head;
35.            data[tail] = dataBaru;
36.        }else if(!isFull()){
37.            tail++;
38.            data[tail] = dataBaru;
39.        }else if(isFull()){
40.            System.out.println("antrian sudah penuh");
41.        }
42.    }
43.

```

```

44.     public int Dequeue(){
45.         int temp = data[head];
46.         for(int i=head;i<=tail-1;i++){
47.             data[i] = data[i+1];
48.         }
49.         tail--;
50.         return temp;
51.     }
52.
53.     public void tampilkan(){
54.         if(!isEmpty()){
55.             int index = head;
56.             while(index <= tail){
57.                 System.out.print("|"+data[index]+"| ");
58.                 index++;
59.             }
60.             System.out.println();
61.         }else{
62.             System.out.println("Kosong");
63.         }
64.     }
65.
66. }

```

- Buat kelas dengan nama QueueApp, kemudian instance objek dari kelas Queue.

```

67. /**
68.  *
69.  * @author Wim Sonevel
70.  */
71. public class QueueApp {
72.     public static void main(String[] args) {
73.         Queue queue = new Queue(3);
74.         queue.Enqueue(1);
75.         queue.Enqueue(2);
76.         queue.Enqueue(3);
77.         queue.tampilkan();
78.         queue.Dequeue();
79.         queue.tampilkan();
80.         queue.Dequeue();
81.         queue.tampilkan();
82.         queue.Dequeue();
83.         queue.tampilkan();
84.     }
85. }

```

Output :

|1| |2| |3|

|2| |3|

|3|

Kosong