

JOBSHEET 8

Nama : Farid Aziz Wicaksono

Kelas : TI/1C

Absen : 14

A. PRAKTIKUM

Praktikum1

No	Queue.java
1	package minggu8;
2	public class Queue {
3	int max, size, front, rear;
4	int[] q;
5	
6	Queue(int m){
7	max = m;
8	create();
9	}
10	void create(){
11	q = new int[max];
12	size = 0;
13	front = rear = -1;
14	}
15	boolean isEmpty(){
16	if(size == 0){
17	return true;
18	}
19	else
20	return false;
21	}
22	boolean isFull(){
23	if(size == max){
24	return true;
25	}
26	else
27	return false;
28	}
29	void enqueue(int data){
30	if(isFull()){
31	System.out.println("Antrian sudah penuh!!");
32	}
33	else{
34	if(isEmpty()){
35	front = rear = 0;
36	}
37	else{
38	if(rear == max-1){
39	rear = 0;

```

40         }
41         else{
42             rear++;
43         }
44     }
45 }
46 q[rear] = data;
47 size++;
48 }
49 int dequeue(){
50     int data = 0;
51     if(isEmpty()){
52         System.out.println("Antrian kosng!!");
53     }
54     else{
55         data = q[front];
56         size--;
57         if(isEmpty()){
58             front = rear = -1;
59         }
60         else{
61             if(front == max-1){
62                 front = 0;
63             }
64             else{
65                 front++;
66             }
67         }
68     }
69     return data;
70 }
71 void print(){
72     if(isEmpty()){
73         System.out.print("Antrian kosong!!");
74     }
75     else{
76         int i = front;
77         while(i!= rear){
78             System.out.print(q[i]+"-");
79             i = (i+1)%max;
80         }
81         System.out.println(q[i]+" ");
82         System.out.println("Jumlah antrian = "+size);
83     }
84 }
85 }

```

Praktikum 2

No	QueueMain.java
1	package minggu8;
2	import java.util.Scanner;
3	
4	public class QueueMain {
5	static void menu(){
6	System.out.println("Plih operasi yang ingin dilakukan");
7	System.out.println("1. Enqueue");
8	System.out.println("2. Dequeue");
9	System.out.println("3. print");
10	System.out.println("4. keluar");
11	}
12	
13	public static void main(String[] args){
14	Scanner sc = new Scanner(System.in);
15	int pil = 0;
16	System.out.print("Masukan berapa maksimal data antrian = ");
17	int m = sc.nextInt();
18	Queue qobj = new Queue(m);
19	do{
20	menu();
21	pil = sc.nextInt();
22	switch(pil){
23	case 1: System.out.print("masukan data baru = ");
24	int dataIn = sc.nextInt();
25	qobj.enqueue(dataIn);
26	break;
27	case 2: int dataOut = qobj.dequeue();
28	if(dataOut != 0)
29	System.out.println("Data yang terambil = "+dataOut);
30	break;
31	case 3: qobj.print();
32	break;
33	}
34	}
35	while(pil != 4);
36	}
37	}

B. PERTANYAAN

1. Jelaskan pengertian dari Queue!
2. Perhatikan class Queue, di dalamnya terdapat atribut q. Untuk apakah atribut tersebut?
3. Jelaskan apa kegunaan dari atribut max, size, front dan rear yang ada di dalam class Queue!
4. Perhatikan konstruktor class Queue, di dalam konstruktor tersebut terdapat statement untuk memanggil method create(). Apa tujuan pemanggilan tersebut method tersebut?
5. Perhatikan isi method create(), kenapa atribut front dan rear diinisialisasi ke -1 dan tidak ke 0?
6. Perhatikan isi dari method isEmpty() dan isFull(), kapan suatu queue dinyatakan kosong? Dan kapan pula queue dinyatakan penuh?
7. Perhatikan kembali isi dari method isFull(), jika kondisi di dalam IF diubah menjadi `size==max-1`, menurut Anda pengaruh apa yang akan terjadi?
8. Perhatikan method enqueue(), di dalamnya terdapat statement sbb:

```
if(rear==max-1){  
    rear = 0;  
}
```

Untuk apakah proses tersebut?

9. Perhatikan kembali method enqueue(), statement mana yang menunjukkan bahwa data baru disimpan di dalam posisi terakhir queue?
10. Perhatikan method dequeue(), mengapa method tersebut tidak dibuat ber-tipe data void?
11. Pada method dequeue(), statement mana yang menunjukkan bahwa pada proses pengambilan data, data paling depan yang terambil?
12. Pada method dequeue(), di dalamnya terdapat statement sbb:

```
if(front==max-1){  
    front = 0;  
}
```

Untuk apakah proses tersebut?

13. Statement mana yang menunjukkan bahwa ketika data terambil dari suatu queue, maka jumlah data yang ada di dalam berkurang satu?
14. Perhatikan method print(), mengapa pada awal proses perulangan, variabel i dimulai dari front (`int i=front`, bukan `int i=0`)?
15. Perhatikan method print(), untuk apakah proses `i=(i+1)%max` ?
16. Perhatikan method print(), kapan proses perulangan di dalam method print() akan berhenti?
17. Perhatikan method main(), statement mana saja yang menunjukkan:
 - a. Proses pembuatan objek dari class Queue
 - b. Proses pemanggilan method enqueue()
 - c. Proses pemanggilan method dequeue
 - d. Proses pemanggilan method print()

Jawab :

1. **Queue (antrian)** adalah sekumpulan elemen/data dimana proses **memasukkan/menambah** elemen/datadilakukan padaposisi **belakang** (rear) dan proses **mengeluarkan/mengambil** elemen/data dilakukan pada elemen/data di posisi **depan** (front).
2. Atribut **Q** digunakan untuk menyimpan data queue
3. **Max** : atribut/variabel yang akan digunakan untuk menyimpan banyak data maksimal yang bisa disimpan di dalam queue
Size : atribut/variabel yang akan digunakan untuk menyimpan berapa banyak data yang ada dalam antrian
Front : atribut/variabel yang akan digunakan untuk menyimpan nilai indeks array, dimana data terdepan dari antrian berada
Rear : atribut/variabel yang akan digunakan untuk menyimpan nilai indeks array, dimana data paling belakang dari antrian berada
4. Digunakan untuk membuat objek queue
5. Agar queue dalam keadaan kosong, jika di set menjadi 0 maka queue tersebut sudah ada isinya yaitu terdapat di index ke 0
6. Queue dinyatakan kosong ketika size atau queue == 0, dinyatakan penuh ketika size sudah dalam kondisi max (ukuran maksimal queue telah di set oleh inputan di method main)
7. Jika size == max - 1 maka isi dari maksimal yang telah diinputkan user akan dikurang -1, misalkan input 3 maka maksimalnya adalah $3 - 1 = 2$
8. Untuk mengurangi isi dari queue
9. Ada pada statemen `q[rear] = data; size++;`
10. Karena dequeue mempunyai nilai kembalian
11.

```
else if (front == max - 1) {  
    front = 0;  
} else {  
    front++;  
}
```

12. Untuk proses pengurangan nilai yang paling atas

13. if(isEmpty())

front = rear = -1;

14. dimulai dari nilai yang paling awal (rear)

15. Sebagai parameter dari i yang terus bertambah 1, selama nilai i tidak sama dengan nilai rear maka proses tersebut akan berjalan

16. Ketika queue sedang dalam keadaan kosong maupun sedang terisi

17.

a. System.out.print("Masukkan berapa maksimal data

antrian : ")

Int m = sc.nextInt();

Queue qobj = new

Queue(m);

b. Case 1:

System.out.print("Masukkan data baru :");

Int dataIn = sc.nextInt();

Qobj.enqueue(dataIn);

break;

c. Case 2:

int dataOut =

qobj.dequeue();

if(dataOut!=0)

System.out.println("Data yang terambil : "+dataOut);

break;

d. if(isEmpty()){

System.out.println("Antrian kosong!!");

}else{

int i = front;

while(i!=rear){

System.out.print(q[i]+"_");

i = (i+1)%max;