

## 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	}

Output :

```
run:
Masukan berapa maksimal data antrian = 3
Plih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. keluar
1
masukan data baru = 1
Plih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. keluar
1
masukan data baru = 2
Plih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. keluar
1
masukan data baru = 3
Plih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. keluar
1
masukan data baru = 4
Antrian sudah penuh!!
Plih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. keluar
2
Data yang terambil = 1
Plih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. keluar
3
2-4
Jumlah antrian = 3
Plih operasi yang ingin dilakukan
1. Enqueue
```

## 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 diambil?
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 diambil 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 diset 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;
```

### C. TUGAS

1. Tambahkan method-method di bawah ini di dalam class Queue di atas.
  - a. Method printFront() : void > untuk menampilkan data/nilai yang ada di posisi antrian paling depan
  - b. Method printRear() : void > untuk menampilkan data/nilai yang ada di posisi antrian paling belakang
  - c. Method printPosition(data: int) : void > untuk menampilkan di posisi ke berapa dalam antrian, suatu data/nilai berada
  - d. Method printDataByPos(position: int) : void > untuk menampilkan data/nilai yang ada di suatu posisi antrian tertentu. Kemudian lakukan penyesuaian di dalam class QueueMain untuk mensimulasikan pemanggilan method-method yang baru ditambahkan tersebut!
2. Buatlah program antrian nasabah di suatu bank. Ketika seorang nasabah akan antri, maka ia harus menuliskan terlebih dulu no. rekening, dan nama. Jadi, antrian yang akan dibuat, berisi data-data nasabah berupa no. rekening dan nama. Sehingga pertama kali yang harus dibuat adalah class Nasabah sbb:

Nasabah
Nasabah noRekening: String nama: String
Nasabah(noRek: String, nm: String) Nasabah() print(): void

Selanjutnya buatlah class Queue sbb

Queue
Queue max: int front: int rear: int size: int q: Nasabah[]
Queue(max: int) create(): void isEmpty(): Boolean isFull(): Boolean enqueue(data: int): void dequeue(): int print(): void printFront(): void printRear(): void printPosition(nas: Nasabah): vo id printNasabah(posisi: int): void



Catatan:

- Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada sesi praktikum
- Method printFront(): digunakan untuk menampilkan data Nasabah yang ada di posisi antrian paling depan
- Method printRear(): digunakan untuk menampilkan data Nasabah yang ada di posisi antrian paling belakang
- Method printPosition(): digunakan untuk menampilkan posisi antrian ke berapa, seorang Nasabah berada
- Method printNasabah(): digunakan untuk menampilkan data nasabah pada suatu posisi tertentu dalam antrian

Jawab :

1. Tugas 1

No	Queue.java
1	package minggu8;
2	
3	public class Queue {
4	int max, size, front, rear;
5	int[] q;
6	
7	Queue(int m){
8	max = m;
9	create();
10	}
11	
12	void create(){
13	q = new int[max];
14	size = 0;
15	front = rear = -1;
16	}
17	
18	boolean isEmpty(){
19	if(size == 0){
20	return true;
21	}
22	else
23	return false;
24	}
25	

```

26 boolean isFull(){
27     if(size == max){
28         return true;
29     }
30     else
31         return false;
32 }
33
34 void equeue(int data){
35     if(isFull()){
36         System.out.println("Antrian sudah penuh!!");
37     }
38     else{
39         if(isEmpty()){
40             front = rear = 0;
41         }
42         else{
43             if(rear == max-1){
44                 rear = 0;
45             }
46             else{
47                 rear++;
48             }
49         }
50     }
51     q[rear] = data;
52     size++;
53 }
54
55 int dequeue(){
56     int data = 0;
57     if(isEmpty()){
58         System.out.println("Antrian kosng!!");
59     }
60     else{
61         data = q[front];
62         size--;
63         if(isEmpty()){
64             front = rear = -1;
65         }
66         else{
67             if(front == max-1){
68                 front = 0;
69             }
70             else{
71                 front++;

```

```

72     }
73     }
74     }
75     return data;
76 }
77
78 void print(){
79     if(isEmpty()){
80         System.out.print("Antrian kosong!!");
81     }
82     else{
83         int i = front;
84         while(i!= rear){
85             System.out.print(q[i]+"-");
86             i = (i+1)%max;
87         }
88         System.out.println(q[i]+" ");
89         System.out.println("Jumlah antrian = "+size);
90     }
91 }
92
93 void printFront() {
94     int elemenFront = q[front];
95     System.out.println("Menampilkan data depan : "+ elemenFront);
96 }
97
98 void printRear() {
99     int elemenRear = q[rear];
100     System.out.println("Menampilkan data belakang : " +
101         elemenRear);
102 }
103
104 void printPosition(int data) {
105     int count = 0;
106     if (isEmpty()) {
107         System.out.println("\n Data tidak ditemukan ");
108     }
109     else {
110         int i = front;
111         while (i != rear) {
112             count++;
113             if (data == q[i]) {
114                 System.out.println("\n Data ditemukan diIndex: " + count);
115                 break;
116             }
117             i = (i + 1) % max;

```

118	}
119	count++;
120	if (data == q[i]) {
121	}
122	System.out.println("\n Data ditemukan di Index : " + count);
123	}
124	}
125	
126	void printDataByPos(int position) {
127	int count = 0;
128	if (isEmpty()) {
129	System.out.println("\n Data tidak ditemukan ");
130	}
131	else{
132	int i = front;
133	while (i != rear) {
134	count++;
135	if (position == count) {
136	System.out.println("\n Indeks " + (i + 1) + " : " + q[i]);
137	break;
138	}
139	i = (i + 1) % max;
140	}
141	count++;
142	System.out.println("\n Indeks " + (i + 1) + ":" + q[i]);
143	}
144	}
145	}

No	QueueMain.java
1	package minggu8;
2	import java.util.Scanner;
3	
4	public class QueueMain {
5	static void menu(){
6	System.out.println("Pilih 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. PrintFront");
11	System.out.println("5. PrintRear");
12	System.out.println("6. PrintPosition");
13	System.out.println("7. PrintDataByPos");
14	System.out.println("8. Keluar");
15	}
16	

```
17 public static void main(String[] args){
18     Scanner sc = new Scanner(System.in);
19     int pil = 0;
20     System.out.print("Masukan berapa maksimal data antrian = ");
21     int m = sc.nextInt();
22     Queue qobj = new Queue(m);
23     do{
24         menu();
25         pil = sc.nextInt();
26         switch(pil){
27             case 1: System.out.print("masukan data baru = ");
28                 int dataIn = sc.nextInt();
29                 qobj.enqueue(dataIn);
30                 break;
31             case 2: int dataOut = qobj.dequeue();
32                 if(dataOut != 0)
33                     System.out.println("Data yang terambil = "+dataOut);
34                 break;
35             case 3: qobj.print();
36                 break;
37             case 4: qobj.printFront();
38                 break;
39             case 5: qobj.printRear();
40                 break;
41             case 6: System.out.println("Masukan angka : ");
42                 int data = sc.nextInt();
43                 qobj.printPosition(data);
44                 break;
45             case 7: System.out.println("Masukan Indeks : ");
46                 int position = sc.nextInt();
47                 qobj.printDataByPos(position);
48                 break;
49         }
50     } while (pil != 8);
51 }
52 }
```

Output :

```
masukan data baru = 3
Pilih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. PrintFront
5. PrintRear
6. PrintPosition
7. PrintDataByPos
8. Keluar
↓
masukan data baru = 3
Pilih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. PrintFront
5. PrintRear
6. PrintPosition
7. PrintDataByPos
8. Keluar
↓
Menampilkan data depan : 1
Pilih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. PrintFront
5. PrintRear
6. PrintPosition
7. PrintDataByPos
8. Keluar
↓
Menampilkan data belakang : 3
Pilih operasi yang ingin dilakukan
1. Enqueue
2. Dequeue
3. print
4. PrintFront
5. PrintRear
6. PrintPosition
7. PrintDataByPos
8. Keluar
```

## 2. Tugas 2

No	Nasabah.java
1	package minggu8;
2	
3	public class Nasabah {
4	String NoRekening, Nama;
5	
6	Nasabah(String nma, String nr) {
7	this.NoRekening = nr;
8	this.Nama = nma;
9	}
10	
11	void printNasabah() {
12	System.out.println(Nama + " No Rekening : " + NoRekening);
13	}
14	}

No	NasabahQueue.java
1	package minggu8;
2	
3	public class NasabahQueue {
4	int max, size, front, rear;
5	Nasabah[] nasabahh;
6	int idx;
7	
8	NasabahQueue(int max) {
9	this.max = max;
10	create();
11	}
12	
13	void create() {
14	nasabahh = new Nasabah[max];
15	size = 0;
16	front = rear = -1;
17	}
18	
19	boolean isEmpty() {
20	if (size == 0) {
21	return true;
22	}
23	else{
24	return false;
25	}
26	}
27	
28	boolean isFull() {

```

29     if (size == max) {
30         return true;
31     }
32     else {
33         return false;
34     }
35 }
36
37 void enqueue(Nasabah nasabah) {
38     if (isFull()) {
39         System.out.println("Queue sudah penuh! ");
40     }
41     else {
42         if (isEmpty()) {
43             front = rear = 0;
44         }
45         else if (rear == max - 1) {
46             rear = 0;
47         }
48         else {
49             rear++;
50         }
51         nasabahh[rear] = nasabah;
52         size++;
53     }
54 }
55
56 Nasabah dequeue() {
57     Nasabah data = null;
58     if (isEmpty()) {
59         System.out.println("Queue kosong ! ");
60     }
61     else {
62         data = nasabahh[front];
63         size--;
64         if (isEmpty()) {
65             front = rear = -1;
66         }
67         else if (front == max - 1) {
68             front = 0;
69         }
70         else {
71             front++;
72         }
73     }
74     return data;

```



```

75     }
76
77     void print() {
78         System.out.println();
79         if (isEmpty()) {
80             System.out.println("Queue kosong! ");
81         }
82         else {
83             int i = front;
84             while (i != rear) {
85                 nasabahh[i].printNasabah();
86                 System.out.println(" ===== ");
87                 i = (i + 1) % max;
88             }
89             nasabahh[i].printNasabah();
90             System.out.println("Queue size : " + size+ "\n");
91         }
92     }
93
94     void printFront() {
95         System.out.println("Data Nasabah paling depan : ");
96         nasabahh[front].printNasabah();
97     }
98
99     void printRear() {
100         System.out.println("Data Nasabah paling terakhir: ");
101         nasabahh[rear].printNasabah();
102     }
103
104     void printPosition(String nama) {
105         int hitung = 0;
106         if (isEmpty()) {
107             System.out.println("\n Queue Kosong ");
108         }
109         else {
110             int i = front;
111             while (i != rear) {
112                 hitung++;
113                 if (nama.equalsIgnoreCase(nasabahh[i].Nama)) {
114                     System.out.print("\n Nasabah yang ada dalamqueue:" + hitung);
115                     break;
116                 }
117                 i = (i + 1) % max;
118             }
119             System.out.println("\n Nasabah yang ada dalam queue :"+ hitung);
120             nasabahh[i].printNasabah();
121

```

122	}
123	}
124	
125	void printDataByPosition(int position) {
126	int hitung = 0;
127	if (isEmpty()) {
128	System.out.println("\n Queue kosong ");
129	}
130	else {
131	int i = front;
132	while (i != rear) {
133	hitung++;
134	if (position == hitung) {
135	System.out.println("Yang ada dalam queue "+ hitung + " : ");
136	nasabahh[i].printNasabah();
137	break;
138	}
139	i = (i + 1) % max;
140	}
141	}
142	}
143	}

No	NasabahMain.java
1	package minggu8;
2	import java.util.*;
3	public class NasabahMain {
4	static Scanner sc = new Scanner(System.in);
5	
6	static Nasabah Informasi() {
7	System.out.print("Nama Nasabah : ");
8	String nama = sc.next();
9	System.out.print("Bank akun nasabah : ");
10	String NoRekening = sc.next();
11	Nasabah data = new Nasabah(nama, NoRekening);
12	return data;
13	}
14	
15	public static void main(String[] args) {
16	System.out.println("Berapa Banyak Nasabah ? ");
17	System.out.print("Jumlah Nasabah : ");
18	int banyak = sc.nextInt();
19	NasabahQueue list = new NasabahQueue(banyak);
20	
21	for (int i = 0; i < banyak; i++) {
22	list.nasabahh[i] = Informasi();

```

23         list.enqueue(list.nasabahh[i]);
24     }
25
26     list.print();
27     list.printFront();
28     list.printRear();
29     System.out.println();
30
31     System.out.print("Cari nama nasabah dalam queue : ");
32     String nama = sc.next();
33     list.printPosition(nama);
34     System.out.print("Indeks Queue : ");
35     int position = sc.nextInt();
36     list.printDataByPosition(position);
37 }
38 }

```

Output :

```

run:
Berapa Banyak Nasabah ?
Jumlah Nasabah : 2
Nama Nasabah : Farid
Bank akun nasabah : BNI
Nama Nasabah : Aziz
Bank akun nasabah : Bank DKI

Farid No Rekening : BNI
=====
Aziz No Rekening : Bank
Queue size : 2

Data Nasabah paling depan :
Farid No Rekening : BNI
Data Nasabah paling terakhir:
Aziz No Rekening : Bank

Cari nama nasabah dalam queue :
Nasabah yang ada dalam queue :1
Aziz No Rekening : Bank
Indeks Queue : 1
Yang ada dalam queue 1 :
Farid No Rekening : BNI
BUILD SUCCESSFUL (total time: 1 minute 50 seconds)

```