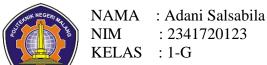
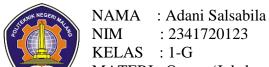


1. Praktikum 1



MATERI: Queue (Jobsheet 8)

```
Masukkan operasi yang diinginkan :
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_16_01> & 'C:\Progr
                                                                                       1. Enqueue.
                                                                                       2. Dequeue.
                                                                                       3. Print.
4
Masukkan operasi yang diinginkan :
1. Enqueue.
2. Dequeue.
3. Print.
4. Peek
5. Clear.
                                                                                       4. Peek
                                                                                       5. Clear.
    ukkan data baru :
                                                                                       Jumlah elemen = 1
                                                                                       Masukkan operasi yang diinginkan :
    sukkan operasi yang diinginkan :
Masukkan ope
1. Enqueue.
2. Dequeue.
3. Print.
4. Peek
5. Clear.
                                                                                       1. Enqueue.
                                                                                       2. Dequeue.
                                                                                       Print.
                                                                                       4. Peek
                                                                                       5. Clear.
   sukkan data baru :
31
Masukkan operasi yang diinginkan :
1. Enqueue.
2. Dequeue.
3. Print.
4. Peek
5. Clear.
                                                                                       Elemen terdepan : 31
                                                                                       Masukkan operasi yang diinginkan :
                                                                                       1. Engueue.
                                                                                       Dequeue.
      en terdepan : 15
kkan operasi yang diinginkan :
                                                                                       Print.
                                                                                       4. Peek
                                                                                       Clear.
                                                                                       Queue berhasil dikosongkan.
```



KELAS: 1-G

MATERI: Queue (Jobsheet 8)

Pertanyaan!

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?

Jawab : Nilai awal atribut front dan rear nilainya -1 karena untuk menunjukkan queue awalnya tidak memiliki elemen. Berarti, saat queue dibuat, belum ada elemen yang dimasukkan, sehingga tidak ada indeks yang valid untuk elemen pertama dan terakhir dalam queue. Sedangkan nilai awal atribut size diatur menjadi 0 karena untuk melacak jumlah elemen yang saat ini ada dalam queue. Pada awalnya, saat queue baru dibuat, tidak ada elemen yang dimasukkan, sehingga ukuran queue adalah 0.

2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

```
if (rear == max -1) {
rear = 0:
```

Jawab : Potongan kode tersebut bertanggung jawab untuk menangani kasus di mana rear telah mencapai indeks maksimum dari array yang digunakan untuk menyimpan elemen dalam queue. Potongan kode tersebut memastikan bahwa queue dapat terus berputar dan tidak terbatas oleh batas indeks maksimum array yang digunakan untuk menyimpan elemen-elemen queue.

3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

Jawab: Potongan kode tersebut untuk menangani front telah mencapai indeks maksimum dari array yang digunakan untuk menyimpan elemen dalam queue.

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?

Jawab : Karena queue yang diimplementasikan menggunakan konsep circular queue. Dalam circular queue, elemen-elemen queue mungkin tidak selalu berurutan dari indeks 0 hingga max - 1. Jadi, front sebagai awal perulangan memungkinkan kita untuk mencetak elemen-elemen queue dengan benar dan memperlakukan queue seperti circular queue yang diharapkan.

5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

i = (i + 1) % max;

Jawab: Potongan kode tesebut digunakan untuk menggeser variabel i ke indeks berikutnya dalam array yang digunakan untuk menyimpan elemen-elemen queue. Dengan demikian, potongan kode tersebut memastikan bahwa perulangan mencetak elemen-elemen queue akan berlanjut secara benar dari indeks pertama hingga terakhir.

6. Tunjukkan potongan kode program yang merupakan queue overflow! Jawab:

```
case 1:
System.out.println("Masukkan data baru: ");
int dataMasuk = sc.nextInt();
```



KELAS : 1-G

MATERI : Queue (Jobsheet 8)

Q.Enqueue(dataMasuk); break;

7. Pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan! Jawab:

```
public class Queue {
            public void Enqueue(int dt) {
   if (!IsFull()) {
                         if (IsEmpty()) {
   front = rear = 0;
                         } else {
   if (rear == max - 1) {
                                      rear = 0;
                              rear++;
                          data[rear] = dt;
                         size++;
                          System.exit(status:0);
           public int Dequeue() {
                  int dt = 0;
if (!IsEmpty()) {
    dt = data[front];
                         size--;
if (size == 0) [
front = rear = -1;
                         front++:
                    return dt;
System.out.println(x:"Masukkan kapasitas queue : ");
int n = sc.nextInt();
     menu();
pilih = sc.nextInt();
switch (pilih) {
         tch (pllih) {
   case 1:
   if (!Q.IsFull()) {
        System.out.println(x:"Masukkan data baru : ");
        int dataMasuk = sc.nextInt();
        Q.Enqueue(dataMasuk);
        } else {
        System.out.println(x:"Queue sudah penuh. Tidak dapat menambahkan elemen baru.");
        }
}
```



KELAS: 1-G

MATERI : Queue (Jobsheet 8)

2. Praktikum 2



KELAS : 1-G

MATERI : Queue (Jobsheet 8)

```
| Profit | Post | Post
```

```
Or_Data_10.6(100n) 'Porrestitund_Question'
Plankann possolitus dume:

Plint menu :

1. Antriam Baru.

2. Antriam Schuar

3. Antriam Schuar

4. Gas Seman Antriam

1. Antriam Antriam

1. Antriam Baru.

3. Antriam Baru.

4. Gas Seman Antriam

5. Saldo:

1. Antriam Baru.

2. Antriam Schuar

5. Saldo:

1. Antriam Baru.

2. Antriam Schuar

1. Antriam Baru.

3. Antriam Baru.

4. Gas Seman Antriam

1. Antriam Baru.

5. Antriam Baru.

1. Antriam Baru.

1. Antriam Baru.

1. Antriam Baru.

2. Antriam Schuar

1. Antriam Baru.

2. Antriam Schuar

1. Antriam Baru.

2. Antriam Schuar

2. Antriam Schuar

3. Saldo:

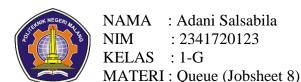
4. Gas Seman Antriam

5. Cas Antriam Baru.

5. Cas Seman Antriam

6. Cas Seman Antriam

7. Cas Se
```



Pertanyaan!

1. Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!

```
if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat) &&
data.umur != 0 && data.saldo != 0) {
       System.out.println("Antrian yang keluar : " + data.norek + " " + data.nama +
data.alamat + " " + data.umur + " " + data.saldo):
```

Jawab: Kode program di atas digunakan untuk memeriksa apakah data antrian yang keluar memiliki nilai yang valid sebelum mencetaknya. Keseluruhan `if` statement tersebut akan mengevaluasi apakah semua atribut dari objek `data` memiliki nilai yang valid. Jika semua syarat terpenuhi, maka pernyataan `println` akan dijalankan untuk mencetak data antrian yang keluar.

Sebagai berikut:

- 1. !"".equals(data.norek) memeriksa apakah atribut `norek` dari objek `data` tidak kosong. Jika `norek` tidak kosong, maka kondisi ini bernilai `true`.
- 2. !"".equals(data.nama) memeriksa apakah atribut `nama` dari objek `data` tidak kosong.
- 3. !"".equals(data.alamat) memeriksa apakah atribut `alamat` dari objek `data` tidak kosong.
- 4. data.umur != 0 apakah atribut `umur` dari objek `data` tidak sama dengan 0. Sebuah nilai umur yang valid dianggap bukan 0.
- 5. data.saldo != 0 memeriksa apakah atribut `saldo` dari objek `data` tidak sama dengan 0.
- 2. Lakukan modifikasi program dengan menambahkan method baru bernama peekRear pada class Queue yang digunakan untuk mengecek antrian yang berada di posisi belakang! Tambahkan pula daftar menu 5. Cek Antrian paling belakang pada class QueueMain sehingga method peekRear dapat dipanggil!

Jawab:

```
f (IlsEmpty()) {

System.out.println("Antrian paling belakang : " + data[rear].norek + " " + data[rear].nama + " " + data[rear].alamat + " " + data[rear].umur + " " + data[rear].saldo);
    System.out.println(x:"Queue masih kosong.");
```



KELAS: 1-G

MATERI: Queue (Jobsheet 8)

```
mc > Potentions2 > 1 Quantataryse > 2 Quantatary Q managering
pages = n.crastine;
```

```
2. Antrian Keluar.
3. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang

1 No Rekening:
123123
Nama:
Kiak
Alamat:
Candi
Umur:
34
Saldo:
23000000
Pilih menu:
1. Antrian Baru.
2. Antrian Keluar.
3. Cek Antrian Paling Belakang

1. Cek Semua Antrian
5. Cek Antrian Terdepan.
4. Cek Semua Reluar.
5. Soultan
Alamat:
Kalimattan
Umur:
55
Saldo:
33000000
Pilih menu:
1. Antrian Baru.
2. Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Reluar.
3. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Reluar.
3. Cek Antrian Reluar.
5. Cek Antrian Reluar.
5. Cek Antrian Faling Belakang
5. Cek Antrian Reluar.
5. Cek Antrian Terdepan.
6. Cek Semua Antrian
7. Cek Semua Antrian
7. Cek Semua Antrian
7. Cek Cet Antrian Terdepan.
7. Cek Cettrian Terdepan.
```