LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

2022

Praktikan

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi

Praktikum 1

Langkah 1

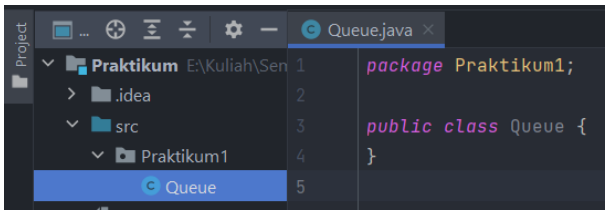
Perhatikan Diagram Class Queue berikut ini:

Queue
data: int[] front: int rear: int size: int max: int
Queue(n: int) isFull(): boolean isEmpty(): boolean enqueue(dt: int): void dequeue(): int peek: void print(): void clear(): void

Berdasarkan diagram class tersebut, akan dibuat program class Queue dalam Java.

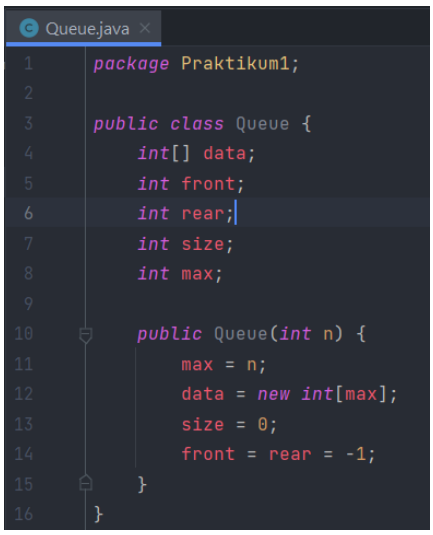
Langkah 2

Buat package dengan nama Praktikum1, kemudian buat class baru dengan nama Queue.



Langkah 3

Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.



Langkah 4

Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong

```
17      public boolean IsEmpty() {
18          if (size == 0) {
19              return true;
20          } else {
21              return false;
22          }
23      }
```

Langkah 5

Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh

```
25      public boolean IsFull() {
26          if (size == max) {
27              return true;
28          } else {
29              return false;
30          }
31      }
```

Langkah 6

Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.

```
33      public void peek() {
34          if (!IsEmpty()) {
35              System.out.println("Elemen terdepan: " + data[front]);
36          } else {
37              System.out.println("Queue masih kosong");
38          }
39      }
```

Langkah 7

Buat method print bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.

```
41      public void print() {
42          if (IsEmpty()) {
43              System.out.println("Queue masih kosong");
44          } else {
45              int i = front;
46              while (i != rear) {
47                  System.out.print(data[i] + " ");
48                  i = (i + 1) % max;
49              }
50              System.out.println(data[i] + " ");
51              System.out.println("Jumlah elemen = " + size);
52          }
53      }
```

Langkah 8

Buat method clear bertipe void untuk menghapus semua elemen pada queue

```
55      public void clear() {
56          if (!IsEmpty()) {
57              front = rear = -1;
58              size = 0;
59              System.out.println("Queue berhasil dikosongkan");
60          } else {
61              System.out.println("Queue masih kosong");
62          }
63      }
```

Langkah 9

Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer

```
65      public void Enqueue(int dt) {
66          if (IsFull()) {
67              System.out.println("Queue sudah penuh");
68          } else {
69              if (IsEmpty()) {
70                  front = rear = 0;
71              } else {
72                  if (rear == max - 1) {
73                      rear = 0;
74                  } else {
75                      rear++;
76                  }
77              }
78              data[rear] = dt;
79              size++;
80          }
81      }
```

Langkah 10

Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi belakang

```
83      public int Dequeue() {
84          int dt = 0;
85          if (IsEmpty()) {
86              System.out.println("Queue masih kosong");
87          } else {
88              dt = data[front];
89              size--;
90              if (IsEmpty()) {
91                  front = rear = -1;
92              } else {
93                  if (front == max - 1) {
94                      front = 0;
95                  } else {
96                      front++;
97                  }
98              }
99          }
100         return dt;
101     }
```

Langkah 11

Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan

```
5   public class QueueMainRidwan {
6       public static void menu() {
7           System.out.println("Masukkan operasi yang diinginkan: ");
8           System.out.println("1. Enqueue");
9           System.out.println("2. Dequeue");
10          System.out.println("3. Print");
11          System.out.println("4. Peek");
12          System.out.println("5. Clear");
13          System.out.println("-----------------------");
14      }
```

Langkah 12

Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc

```
16      public static void main(String[] args) {
17          Scanner sc = new Scanner(System.in);
18      }
```

Langkah 13

Buat variabel n untuk menampung masukan berupa jumlah maksimal elemen yang dapat disimpan pada queue

```
16      public static void main(String[] args) {
17          Scanner sc = new Scanner(System.in);
18          System.out.print("Masukkan kapasitas queue: ");
19          int n = sc.nextInt();
20      }
```

Langkah 14

Lakukan instansiasi objek Queue dengan nama Q dengan mengirimkan parameter n sebagai kapasitas elemen queue

```
21              Queue Q = new Queue(n);
```

Langkah 15

Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

```
22              int pilih;
```

Langkah 16

Lakukan perulangan menggunakan do-while untuk menjalankan program secara terus menerus sesuai masukan yang diberikan. Di dalam perulangan tersebut, terdapat pemilihan kondisi menggunakan switch-case untuk menjalankan operasi queue sesuai dengan masukan pengguna

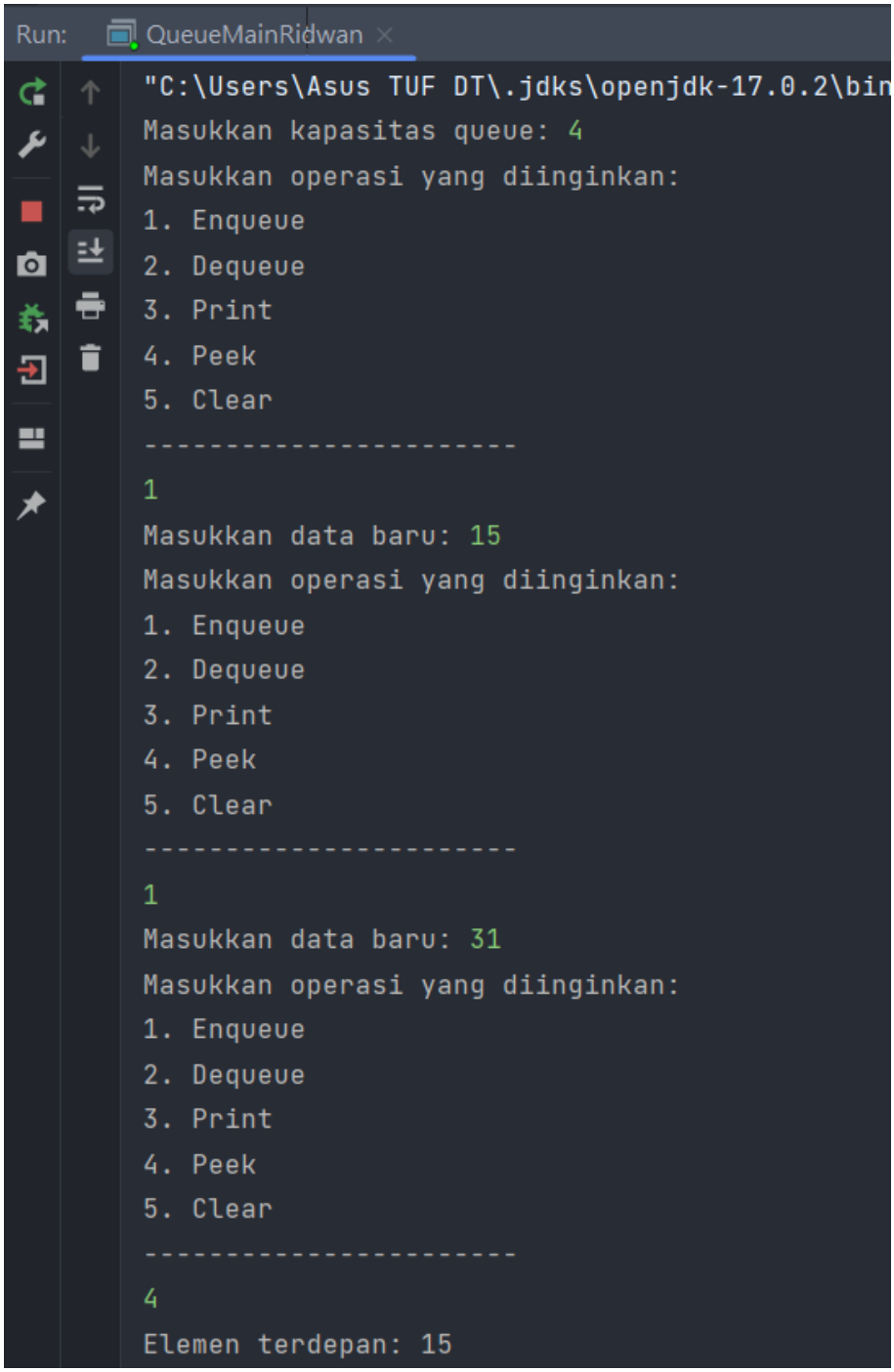
```
24          do {
25              menu();
26              pilih = sc.nextInt();
27              switch (pilih) {
28                  case 1:
29                      System.out.println("Masukkan data baru: ");
30                      int dataMasuk = sc.nextInt();
31                      Q.Enqueue(dataMasuk);
32                      break;
33                  case 2:
34                      int dataKeluar = Q.Dequeue();
35                      if (dataKeluar != 0) {
36                          System.out.println("Data yang dikeluarkan: " + dataKeluar);
37                          break;
38                      }
39                  case 3:
40                      Q.print();
41                      break;
42                  case 4:
43                      Q.peek();
44                      break;
45                  case 5:
46                      Q.clear();
47                      break;
48              }
49          } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
50      }
```

Langkah 17

Compile dan jalankan class QueueMain, kemudian amati hasilnya.

Verifikasi Hasil Percobaan

Samakan hasil compile kode program Anda dengan gambar berikut ini



Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

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

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

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

```
i = (i + 1) % max;
```

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

Jawaban

1. Karena pada awal pembuatan object queue front dan rear tidak menunjuk indeks manapun pada array maka diisi -1 sedangkan untuk size bernilai 0 karena pada saat pembuatan queue belum ada value yang menempatinya.
2. Kode tersebut berguna untuk jika rear berada pada ujung array maka rear akan di set ke indeks 0.
3. Kode tersebut berguna untuk jika front berada pada ujung array maka rear akan di set ke indeks 0.
4. Karena pada queue data awal tidak selalu di indeks ke 0 maka yang perlu di print mulai dari front sampai rear.
5. Kode tersebut digunakan untuk mengisi nilai iterasi I dengan nilai I pada iterasi tersebut ditambahkan 1 kemudian modulus max.
6. Kode untuk mencegah terjadinya queue overflow adalah berikut ini.

```
65      public void Enqueue(int dt) {
66          if (IsFull()) {
67              System.out.println("Queue sudah penuh");
68          } else {
```

Pada kode tersebut dilakukan pengecekkan apakah queue sudah penuh/belum, jika sudah penuh maka tidak bisa diisi namun program tetap berjalan

7. Tambahkan baris kode berikut ini pada class Queue maka jika terjadi overflow/underflow maka program akan dihentikan

```
65      public void Enqueue(int dt) {
66          if (IsFull()) {
67              System.out.println("Queue sudah penuh");
68              System.exit( status: 0);
69          } else {
70              if (IsEmpty()) {
71                  front = rear = 0;
72              } else {
73                  if (rear == max - 1) {
74                      rear = 0;
75                  } else {
76                      rear++;
77                  }
78              }
79              data[rear] = dt;
80              size++;
81          }
82      }
83
84      public int Dequeue() {
85          int dt = 0;
86          if (IsEmpty()) {
87              System.out.println("Queue masih kosong");
88              System.exit( status: 0);
89          } else {
```

Output

```
Masukkan data baru: 2
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----------------------
1
Masukkan data baru: 3
Queue sudah penuh

Process finished with exit code 0
```

Praktikum 2

Langkah 1

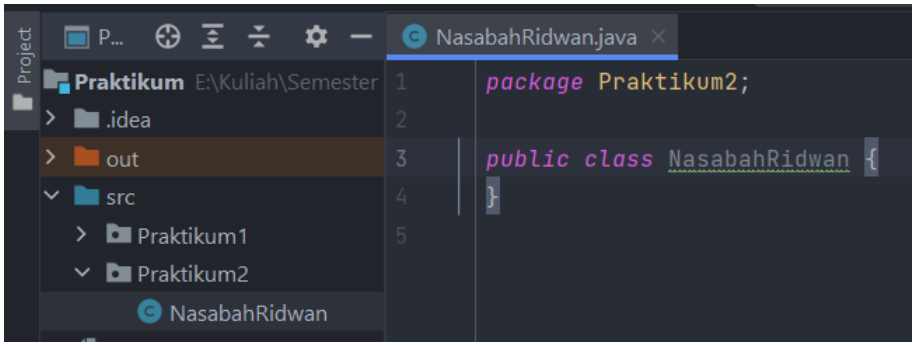
Perhatikan Diagram Class berikut ini:

Nasabah
norek: String nama: String alamat: String umur: int saldo: double
Nasabah(norek: String, nama: String, alamat: String, umur: int, saldo: double)

Berdasarkan diagram class tersebut, akan dibuat program class Nasabah dalam Java.

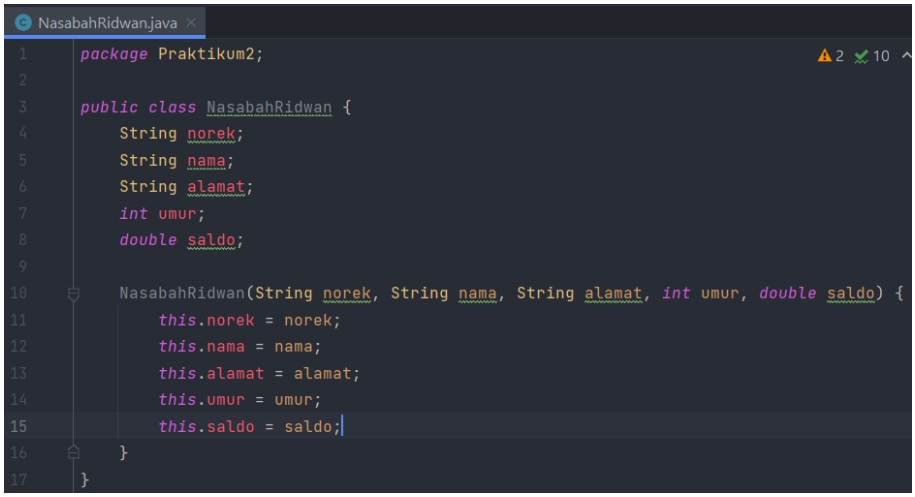
Langkah 2

Buat package dengan nama Praktikum2, kemudian buat class baru dengan nama Nasabah.



Langkah 3

Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini



Langkah 4

Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini. Karena pada Praktikum 1, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada Praktikum 2 data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class Queue tersebut.

Langkah 5

Lakukan modifikasi pada class Queue dengan mengubah tipe int[] data menjadi Nasabah[] data karena pada kasus ini data yang akan disimpan pada queue berupa object Nasabah. Modifikasi perlu dilakukan pada atribut, method Enqueue, dan method Dequeue.

```
public class QueueNasabah {
    NasabahRidwan[] data;
    int front;
    int rear;
    int size;
    int max;

    public QueueNasabah(int n) {
        max = n;
        data = new NasabahRidwan[max];
        size = 0;
        front = rear = -1;
    }

    public void Enqueue(NasabahRidwan dt) {
        if (IsFull()) {
            System.out.println("Queue sudah penuh");
            System.exit( status: 0);
        } else {
            if (IsEmpty()) {
                front = rear = 0;
            } else {
                if (rear == max - 1) {
                    rear = 0;
                } else {
                    rear++;
                }
            }
            data[rear] = dt;
            size++;
        }
    }
```

```
public NasabahRidwan Dequeue() {
    NasabahRidwan dt = new NasabahRidwan();
    if (IsEmpty()) {
        System.out.println("Queue masih kosong");
        System.exit( status: 0);
    } else {
        dt = data[front];
        size--;
        if (IsEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}
```

Baris program Nasabah dt = new Nasabah(); akan ditandai sebagai error, untuk mengatasinya, tambahkan konstruktor default di dalam class Nasabah.

```
NasabahRidwan() {

}
```

Langkah 6

Karena satu elemen queue terdiri dari beberapa informasi (norek, nama, alamat, umur, dan saldo), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut, sehingga meodifikasi perlu dilakukan pada method peek dan method print.

```
public void peek() {
    if (!IsEmpty()) {
        System.out.println("Elemen terdepan: " + data[front].norek + " "
        + data[front].nama + " " + data[front].alamat + " " + data[front].umur
        + data[front].saldo);
    } else {
        System.out.println("Queue masih kosong");
    }
}

public void print() {
    if (IsEmpty()) {
        System.out.println("Queue masih kosong");
    } else {
        int i = front;
        while (i != rear) {
            System.out.println(data[i].norek + " " + data[i].nama + " "
                    + data[i].alamat + " " + data[i].umur
                    + data[i].saldo);
            i = (i + 1) % max;
        }
        System.out.println(data[i].norek + " " + data[i].nama + " "
                + data[i].alamat + " " + data[i].umur
                + data[i].saldo);
        System.out.println("Jumlah elemen = " + size);
    }
```

Langkah 7

Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum2. Buat method menu untuk mengakomodasi pilihan menu dari masukan pengguna

```
7           System.out.println("Pilih menu: ");
8           System.out.println("1. Antrian baru");
9           System.out.println("2. Antrian keluar");
10          System.out.println("3. Cek Antrian Terdepan");
11          System.out.println("4. Cek Semua Antrian");
12          System.out.println("------------------------");
```

Langkah 8

Buat fungsi main, deklarasikan Scanner dengan nama sc

```
16      public static void main(String[] args) {
17          Scanner sc = new Scanner(System.in);
18      }
```

Langkah 9

Buat variabel max untuk menampung kapasitas elemen pada queue. Kemudian lakukan instansiasi objek queue dengan nama antri dan nilai parameternya adalah variabel jumlah.

```
19          System.out.print("Masukkan kapasitas queue: ");
20          int jumlah = sc.nextInt();
21          QueueNasabah antri = new QueueNasabah(jumlah);
```

Langkah 10

Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

```
23              int pilih;
```

Langkah 11

Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.

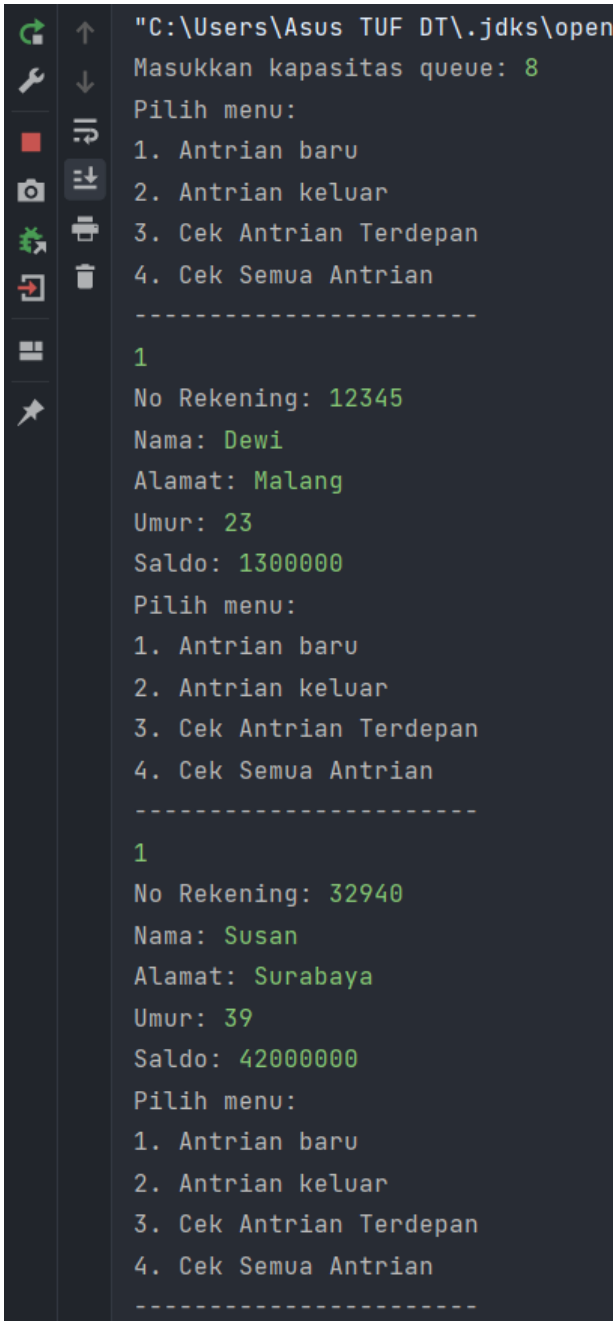
```
        do {
            menu();
            pilih = sc.nextInt();
            sc.nextLine();
            switch (pilih) {
                case 1:
                    System.out.print("No Rekening: ");
                    String norek = sc.nextLine();
                    System.out.print("Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("Alamat: ");
                    String alamat = sc.nextLine();
                    System.out.print("Umur: ");
                    int umur = sc.nextInt();
                    System.out.print("Saldo: ");
                    double saldo = sc.nextDouble();
                    NasabahRidwan nb = new NasabahRidwan(norek, nama, alamat, umur, saldo);
                    sc.nextLine();
                    antri.Enqueue(nb);
                    break;
                case 2:
                    NasabahRidwan data = antri.Dequeue();
                    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);
                        break;
                    }
                case 3:
                    antri.peek();
                    break;
                case 4:
                    antri.print();
                    break;
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4);
    }
```

Langkah 12

Compile dan jalankan class QueueMain, kemudian amati hasilnya.

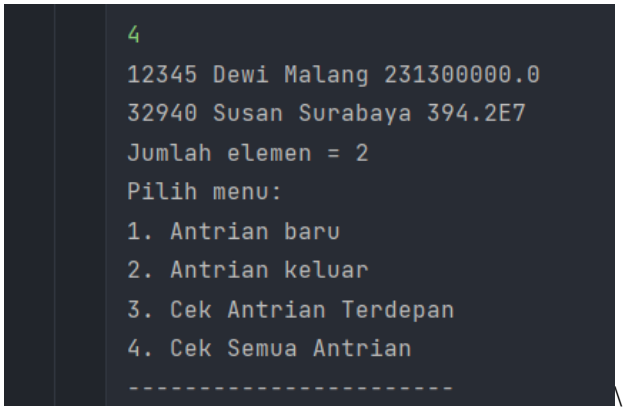
Verifikasi Hasil Percobaan

Samakan hasil compile kode program Anda dengan gambar berikut ini.


"C:\Users\Asus TUF DT\.jdks\open
Masukkan kapasitas queue: 8
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian

1
No Rekening: 12345
Nama: Dewi
Alamat: Malang
Umur: 23
Saldo: 1300000
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian

1
No Rekening: 32940
Nama: Susan
Alamat: Surabaya
Umur: 39
Saldo: 42000000
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian


4
12345 Dewi Malang 231300000.0
32940 Susan Surabaya 394.2E7
Jumlah elemen = 2
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian

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);
    break;
}
```

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!

Jawaban

1. Kode program tersebut berfungsi untuk mengecek apakah nilai dari norek, nama, dan alamat tidak sama dengan string kosong dan mengecek apakah nilai umur dan saldo tidak sama dengan 0 jika iya maka akan ditampilkan antrian yang keluar.
2. Modifikasi source code class QueueNasabah

```
83      public void peekRear() {
84          if (IsEmpty()) {
85              System.out.println("Queue masih kosong");
86          } else {
87              System.out.println("Elemen paling belakang: " + data[rear].norek + " "
88                      + data[rear].nama + " " + data[rear].alamat + " " + data[rear].umur
89                      + " " + data[rear].saldo);
90          }
91      }
```

Modifikasi source code class QueueMainNasabah

```
6       public static void menu() {
7           System.out.println("Pilih menu: ");
8           System.out.println("1. Antrian baru");
9           System.out.println("2. Antrian keluar");
10          System.out.println("3. Cek Antrian Terdepan");
11          System.out.println("4. Cek Semua Antrian");
12          System.out.println("5. Cek Antrian Paling Belakang");
13          System.out.println("------------------------");
14      }
```

```
58                      case 5:
59                          antri.peekRear();
60                          break;
```

Output

```
"C:\Users\Asus TUF DT\.jdks\openjdk-17.0.2\bin\j
Masukkan kapasitas queue: 5
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
----------------------
1
No Rekening: 101
Nama: Ridwan
Alamat: Kediri
Umur: 19
Saldo: 1000000
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
----------------------
1
No Rekening: 102
Nama: Rizqi
Alamat: Malang
Umur: 19
Saldo: 1000000
```

```
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
----------------------
3
Elemen terdepan: 101 Ridwan Kediri 19 1000000.0
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
----------------------
5
Elemen paling belakang: 102 Rizqi Malang 19 1000000.0

Process finished with exit code 0
```

Tugas

Soal Nomor 1

Tambahkan dua method berikut ke dalam class Queue pada Praktikum 1:

a) Method peekPosition(data: int) : void

   Untuk menampilkan posisi dari sebuah data di dalam queue, misalnya dengan mengirimkan data tertentu, akan diketahui posisi (indeks) data tersebut berada di urutan ke berapa

b) Method peekAt(position: int) : void

   Untuk menampilkan data yang berada pada posisi (indeks) tertentu

Sesuaikan daftar menu yang terdapat pada class QueueMain sehingga kedua method tersebut dapat dipanggil!

Soal Nomor 2

Buatlah program antrian untuk mengilustasikan mahasiswa yang sedang meminta tanda tangan KRS pada dosen DPA di kampus. Ketika seorang mahasiswa akan mengantri, maka dia harus menuliskan terlebih dulu NIM, nama, absen, dan IPK seperti yang digambarkan pada Class diagram berikut:

Mahasiswa
nim: String nama: String absen: int ipk: double
Mahasiswa(nim: String, nama: String, absen: int, ipk: double)

Class diagram Queue digambarkan sebagai berikut:

Queue
antrian: Mahasiswa[] front: int

rear: int size: int max: int
Queue(n: int) isEmpty(): boolean isFull(): boolean enqueue(antri: Mahasiswa): void dequeue(): int print(): void peek(): void peekRear(): void peekPosition(nim: String): void printMahasiswa(posisi: int): void

Keterangan:

- Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada Praktikum
- Method peek(): digunakan untuk menampilkan data Mahasiswa yang berada di posisi antrian paling depan
- Method peekRear(): digunakan untuk menampilkan data Mahasiswa yang berada di posisi antrian paling belakang
- Method peekPosition(): digunakan untuk menampilkan posisi antrian ke berapa, seorang Mahasiswa berada. Pengecekan dilakukan berdasarkan NIM
- Method printMahasiswa(): digunakan untuk menampilkan data mahasiswa pada suatu posisi tertentu dalam antrian

Jawaban Nomor 1

Method peekPosition

```
106     public void peekPosition(int data) {
107         int indeks = -1;
108         if (IsEmpty()) {
109             System.out.println("Queue Masih Kosong");
110         } else {
111             int i = front;
112             while (i != rear) {
113                 if (data == this.data[i]) {
114                     indeks = i;
115                     System.out.println("Nilai " + data + " ditemukan pada indeks ke " + indeks);
116                 }
117                 i = (i + 1) % max;
118             }
119             if (data == this.data[i]) {
120                 indeks = i;
121                 System.out.println("Nilai " + data + " ditemukan pada indeks ke " + indeks);
122             }
123             if (indeks == -1) {
124                 System.out.println("Data tidak ditemukan");
125             }
126         }
127 
128     }
```

Method peekAt

```
public void peekAt(int posisi) {
    int indeks = -1;
    if (IsEmpty()) {
        System.out.println("Queue Masih Kosong");
    } else {
        int i = front;
        while (i != rear) {
            if (posisi == i) {
                indeks = i;
                System.out.println("Nilai pada indeks ke " + posisi + " adalah " + data[indeks]);
            }
            i = (i + 1) % max;
        }
        if (posisi == i) {
            indeks = i;
            System.out.println("Nilai pada indeks ke " + posisi + " adalah " + data[indeks]);
        }
        if (indeks == -1) {
            System.out.println("Data tidak ditemukan");
        }
    }
}
```

Output

Sebelumnya sudah saya masukkan data ke dalam queue

```
3. Print
4. Peek
5. Clear
6. PeekPosition
7. PeekAt
-----------------------
3
10 20 30 40
Jumlah elemen = 4
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. PeekPosition
7. PeekAt
-----------------------
6
Masukkan angka yang anda cari : 20
Nilai 20 ditemukan pada indeks ke 1
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. PeekPosition
7. PeekAt
-----------------------
7
Masukkan posisi indeks yang anda cari : 2
Nilai pada indeks ke 2 adalah 30
```

Jawaban Nomor 2

Source code class MahasiswaRidwan

MahasiswaRidwan.java QueueMahasiswa.java QueueMainMahasiswa.java

```java
package Tugas;

public class MahasiswaRidwan {
    String nim;
    String nama;
    int absen;
    double ipk;

    MahasiswaRidwan(String nim, String nama, int absen, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.absen = absen;
        this.ipk = ipk;
    }

    MahasiswaRidwan() {

    }

}
```

Source code class QueueMahasiswa

MahasiswaRidwan.java QueueMahasiswa.java QueueMainMahasiswa.java

```java
package Tugas;

public class QueueMahasiswa {
    MahasiswaRidwan[] antrian;
    int front;
    int rear;
    int size;
    int max;

    QueueMahasiswa(int n) {
        max = n;
        antrian = new MahasiswaRidwan[max];
        front = rear = -1;
        size = 0;
    }

    boolean IsEmpty() {
        return size == 0;
    }

    boolean IsFull() {
        return size == max;
    }

    void enqueue(MahasiswaRidwan antri) {
        if (IsFull()) {
            System.out.println("Antrian sudah penuh");
            System.exit( status: 0);
        } else {
            if (IsEmpty()) {
                front = rear = 0;
            } else {
                if (rear == max -1) {
```

```
                    rear = 0;
                } else {
                    rear++;
                }
            }
            antrian[rear] = antri;
            size++;
        }
    }

    MahasiswaRidwan dequeue() {
        MahasiswaRidwan data = new MahasiswaRidwan();
        if (IsEmpty()) {
            System.out.println("Antrian masih kosong");
            System.exit( status: 0);
        } else {
            data = antrian[front];
            size--;
            if (IsEmpty()) {
                front = rear = -1;
            } else {
                if (front == max - 1) {
                    front = 0;
                } else {
                    front++;
                }
            }
        }
        return data;
    }
```

```
    void print() {
        int count = 0;
        if (IsEmpty()) {
            System.out.println("Antrian masih kosong");
        } else {
            int i = front;
            while (i != rear) {
                count++;
                System.out.println("Mahasiswa Antrian ke " + count);
                System.out.println("Nim\t\t\t : " + antrian[i].nim);
                System.out.println("Nama\t\t : " + antrian[i].nama);
                System.out.println("Absen\t\t : " + antrian[i].absen);
                System.out.println("IPK\t\t\t : " + antrian[i].ipk);
                System.out.println("==================================");
                i = (i + 1) % max;
            }
            count++;
            System.out.println("Mahasiswa Antrian ke " + count);
            System.out.println("Nim\t\t\t : " + antrian[i].nim);
            System.out.println("Nama\t\t : " + antrian[i].nama);
            System.out.println("Absen\t\t : " + antrian[i].absen);
            System.out.println("IPK\t\t\t : " + antrian[i].ipk);
            System.out.println("Jumlah elemen = " + size);
        }
    }
```

```
91      void peek() {
92          if (IsEmpty()) {
93              System.out.println("Antrian masih kosong");
94          } else {
95              System.out.println("Antrian paling depan : ");
96              System.out.println("Nim\t\t\t : " + antrian[front].nim);
97              System.out.println("Nama\t\t : " + antrian[front].nama);
98              System.out.println("Absen\t\t : " + antrian[front].absen);
99              System.out.println("IPK\t\t\t : " + antrian[front].ipk);
100         }
101     }
102
103     void peekRear() {
104         if (IsEmpty()) {
105             System.out.println("Antrian masih kosong");
106         } else {
107             System.out.println("Antrian paling belakang : ");
108             System.out.println("Nim\t\t\t : " + antrian[rear].nim);
109             System.out.println("Nama\t\t : " + antrian[rear].nama);
110             System.out.println("Absen\t\t : " + antrian[rear].absen);
111             System.out.println("IPK\t\t\t : " + antrian[rear].ipk);
112         }
113     }
114
```

```
115     void peekPosition(String nim) {
116         int indeks = -1;
117         int count = 1;
118         if (IsEmpty()) {
119             System.out.println("Antrian Masih Kosong");
120         } else {
121             int i = front;
122             while (i != rear) {
123                 if (nim.equals(antrian[i].nim)) {
124                     indeks = i;
125                     count++;
126                     System.out.println("Mahasiswa dengan NIM " + nim + " ditemukan pada antrian ke " + count);
127                 }
128                 i = (i + 1) % max;
129             }
130             count++;
131             if (nim.equals(antrian[i].nim)) {
132                 indeks = i;
133                 System.out.println("Mahasiswa dengan NIM " + nim + " ditemukan pada antrian ke " + count);
134             }
135             if (indeks == -1) {
136                 System.out.println("Data tidak ditemukan");
137             }
138         }
139     }
```

```
    void printMahasiswa(int posisi) {
        int indeks = -1;
        if (IsEmpty()) {
            System.out.println("Queue Masih Kosong");
        } else {
            int i = front;
            while (i != rear) {
                if (posisi == i) {
                    indeks = i;
                    System.out.println("Mahasiswa pada antrian ke " + posisi + " adalah ");
                    System.out.println("Nim\t\t\t : " + antrian[indeks].nim);
                    System.out.println("Nama\t\t : " + antrian[indeks].nama);
                    System.out.println("Absen\t\t : " + antrian[indeks].absen);
                    System.out.println("IPK\t\t\t : " + antrian[indeks].ipk);
                }
                i = (i + 1) % max;
            }
            if (posisi == i) {
                indeks = i;
                System.out.println("Mahasiswa pada antrian ke " + posisi + " adalah ");
                System.out.println("Nim\t\t\t : " + antrian[indeks].nim);
                System.out.println("Nama\t\t : " + antrian[indeks].nama);
                System.out.println("Absen\t\t : " + antrian[indeks].absen);
                System.out.println("IPK\t\t\t : " + antrian[indeks].ipk);
            }
            if (indeks == -1) {
                System.out.println("Data tidak ditemukan");
            }
        }
    }
}
```

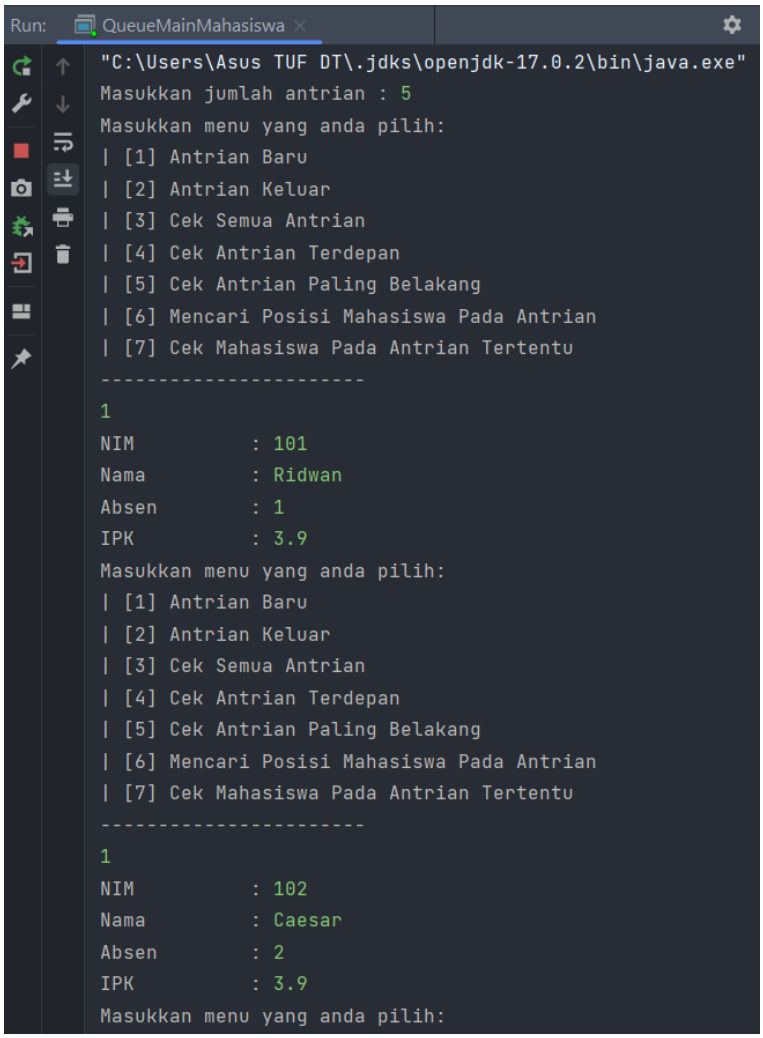
Source code class QueueMainMahasiswa

MahasiswaRidwan.java | QueueMahasiswa.java | QueueMainMahasiswa.java

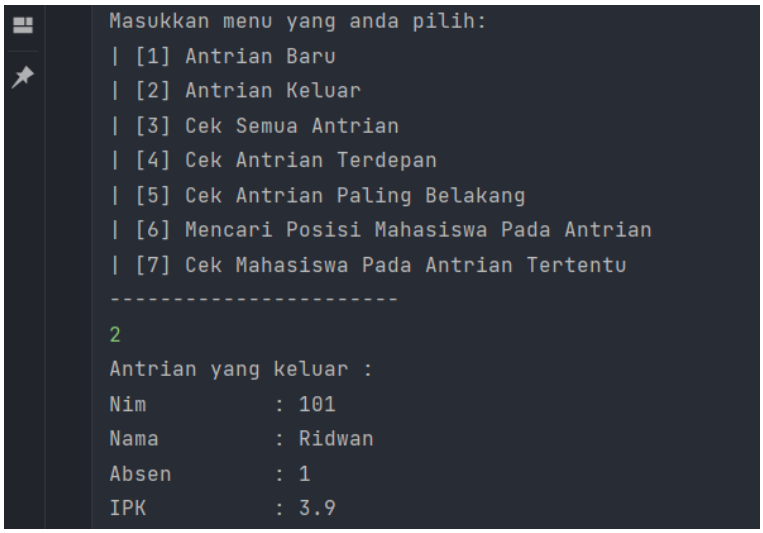
```
package Tugas;

import java.util.Scanner;

public class QueueMainMahasiswa {
    public static void menu() {
        System.out.println("Masukkan menu yang anda pilih: ");
        System.out.println("| [1] Antrian Baru");
        System.out.println("| [2] Antrian Keluar");
        System.out.println("| [3] Cek Semua Antrian");
        System.out.println("| [4] Cek Antrian Terdepan");
        System.out.println("| [5] Cek Antrian Paling Belakang");
        System.out.println("| [6] Mencari Posisi Mahasiswa Pada Antrian");
        System.out.println("| [7] Cek Mahasiswa Pada Antrian Tertentu");
        System.out.println("-----------------------");
    }

```

```
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan jumlah antrian : ");
        int jumlah = input.nextInt();
        QueueMahasiswa antri = new QueueMahasiswa(jumlah);

        int pilih;
        do {
            menu();
            pilih = input.nextInt();
            input.nextLine();
            switch (pilih) {
                case 1 -> {
                    System.out.print("NIM\t\t\t : ");
                    String nim = input.nextLine();
                    System.out.print("Nama\t\t : ");
                    String nama = input.nextLine();
                    System.out.print("Absen\t\t : ");
                    int absen = input.nextInt();
                    System.out.print("IPK\t\t\t : ");
                    double ipk = input.nextDouble();
                    input.nextLine();
                    MahasiswaRidwan mhs = new MahasiswaRidwan(nim, nama, absen, ipk);
                    antri.enqueue(mhs);
                }
                case 2 -> {
                    MahasiswaRidwan data = antri.dequeue();
                    if (!"".equals(data.nim) && !"".equals(data.nama) && data.absen != 0
                    && data.ipk != 0) {
                        System.out.println("Antrian yang keluar : ");
                        System.out.println("Nim\t\t\t : " + data.nim);
                        System.out.println("Nama\t\t : " + data.nama);
                        System.out.println("Absen\t\t : " + data.absen);
                        System.out.println("IPK\t\t\t : " + data.ipk);
                    }
                }
                case 3 -> antri.print();
                case 4 -> antri.peek();
                case 5 -> antri.peekRear();
                case 6 -> {
                    System.out.print("Masukkan NIM yang anda cari : ");
                    String search = input.next();
                    antri.peekPosition(search);
                }
                case 7 -> {
                    System.out.println("Masukkan posisi antrian yang anda cari : ");
                    int posisi = input.nextInt();
                    antri.printMahasiswa(posisi);
                }
            }
        } while (pilih >= 1 && pilih <= 7);
    }
}
```

Output Enqueue



Output Dequeue



Output Cek semua antrian

Sebelumnya sudah saya tambahkan antrian baru

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru
| [2] Antrian Keluar
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
----------------------
3
Mahasiswa Antrian ke 1
Nim          : 102
Nama         : Caesar
Absen        : 2
IPK          : 3.9
================================
Mahasiswa Antrian ke 2
Nim          : 103
Nama         : Rizqi
Absen        : 3
IPK          : 3.9
================================
Mahasiswa Antrian ke 3
Nim          : 104
Nama         : Karisma
Absen        : 4
IPK          : 4.0
Jumlah elemen = 3
```

Output Cek antrian terdepan

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru
| [2] Antrian Keluar
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
----------------------
4
Antrian paling depan :
Nim          : 102
Nama         : Caesar
Absen        : 2
IPK          : 3.9
```

Output Cek antrian paling belakang

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru
| [2] Antrian Keluar
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
----------------------
5
Antrian paling belakang :
Nim          : 104
Nama         : Karisma
Absen        : 4
IPK          : 4.0
```

Output mencari posisi mahasiswa dalam antrian

Disini menampilkan antrian dalam kehidupan nyata bukan dari indeks array.

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru
| [2] Antrian Keluar
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
----------------------
6
Masukkan NIM yang anda cari : 103
Mahasiswa dengan NIM 103 ditemukan pada antrian ke 2
```

Output cek mahasiswa dalam antrian tertentu

Disini mencari berdasarkan antrian dalam kehidupan nyata bukan dari indeks array.

```
Masukkan menu yang anda pilih:
| [1] Antrian Baru
| [2] Antrian Keluar
| [3] Cek Semua Antrian
| [4] Cek Antrian Terdepan
| [5] Cek Antrian Paling Belakang
| [6] Mencari Posisi Mahasiswa Pada Antrian
| [7] Cek Mahasiswa Pada Antrian Tertentu
----------------------
7
Masukkan posisi antrian yang anda cari :
2
Mahasiswa pada antrian ke 2 adalah
Nim          : 103
Nama         : Rizqi
Absen        : 3
IPK          : 3.9
```