**LAPORAN PRAKTIKUM**

**Politeknik Negeri Malang**

**Praktikan**

**ALGORITMA DAN STRUKTUR DATA**

****

2022

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi

[Praktikum 1 4](#_Toc101432422)

[Langkah 1 4](#_Toc101432423)

[Langkah 2 4](#_Toc101432424)

[Langkah 3 4](#_Toc101432425)

[Langkah 4 5](#_Toc101432426)

[Langkah 5 5](#_Toc101432427)

[Langkah 6 5](#_Toc101432428)

[Langkah 7 5](#_Toc101432429)

[Langkah 8 6](#_Toc101432430)

[Langkah 9 6](#_Toc101432431)

[Langkah 10 7](#_Toc101432432)

[Langkah 11 7](#_Toc101432433)

[Langkah 12 7](#_Toc101432434)

[Langkah 13 8](#_Toc101432435)

[Langkah 14 8](#_Toc101432436)

[Langkah 15 8](#_Toc101432437)

[Langkah 16 8](#_Toc101432438)

[Langkah 17 9](#_Toc101432439)

[Verifikasi Hasil Percobaan 9](#_Toc101432440)

[Pertanyaan 10](#_Toc101432441)

[Jawaban 10](#_Toc101432442)

[Praktikum 2 12](#_Toc101432443)

[Langkah 1 12](#_Toc101432444)

[Langkah 2 12](#_Toc101432445)

[Langkah 3 12](#_Toc101432446)

[Langkah 4 13](#_Toc101432447)

[Langkah 5 13](#_Toc101432448)

[Langkah 6 14](#_Toc101432449)

[Langkah 7 15](#_Toc101432450)

[Langkah 8 15](#_Toc101432451)

[Langkah 9 15](#_Toc101432452)

[Langkah 10 15](#_Toc101432453)

[Langkah 11 15](#_Toc101432454)

[Langkah 12 16](#_Toc101432455)

[Verifikasi Hasil Percobaan 16](#_Toc101432456)

[Pertanyaan 18](#_Toc101432457)

[Jawaban 18](#_Toc101432458)

[Tugas 20](#_Toc101432459)

[Soal Nomor 1 20](#_Toc101432460)

[Soal Nomor 2 20](#_Toc101432461)

[Jawaban Nomor 1 21](#_Toc101432462)

[Method peekPosition 21](#_Toc101432463)

[Method peekAt 22](#_Toc101432464)

[Output 22](#_Toc101432465)

[Jawaban Nomor 2 23](#_Toc101432466)

[Source code class MahasiswaRidwan 23](#_Toc101432467)

[Source code class QueueMahasiswa 23](#_Toc101432468)

[Source code class QueueMainMahasiswa 26](#_Toc101432469)

[Output Enqueue 28](#_Toc101432470)

[Output Dequeue 28](#_Toc101432471)

[Output Cek semua antrian 28](#_Toc101432472)

[Output Cek antrian terdepan 29](#_Toc101432473)

[Output Cek antrian paling belakang 30](#_Toc101432474)

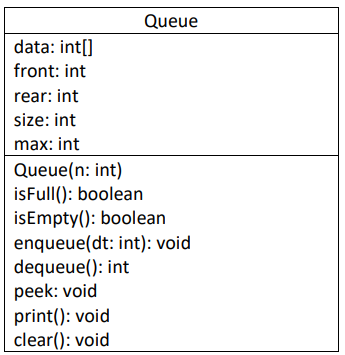
[Output mencari posisi mahasiswa dalam antrian 30](#_Toc101432475)

[Output cek mahasiswa dalam antrian tertentu 30](#_Toc101432476)

# Praktikum 1

## Langkah 1

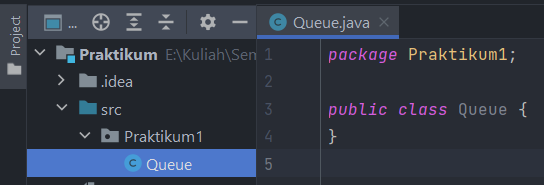
Perhatikan Diagram Class Queue berikut ini:



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

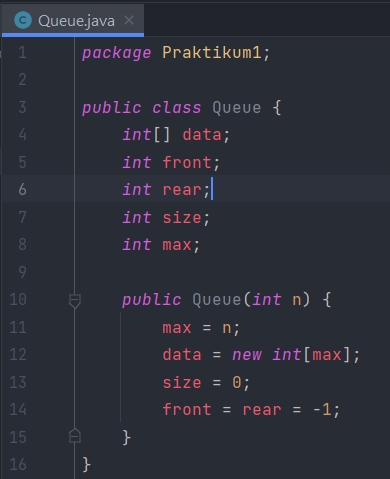
## Langkah 2

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



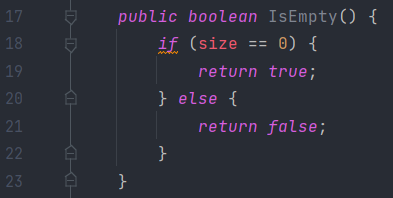
## Langkah 3

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



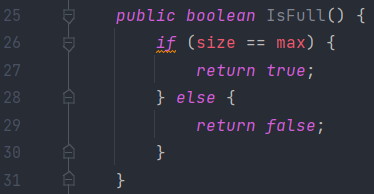
## Langkah 4

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



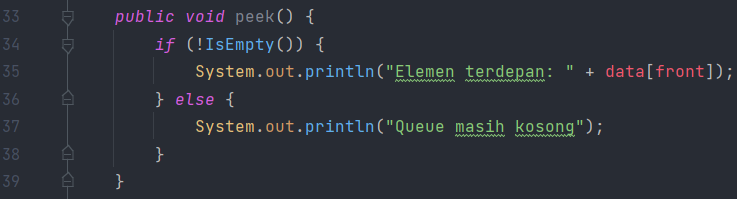
## Langkah 5

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



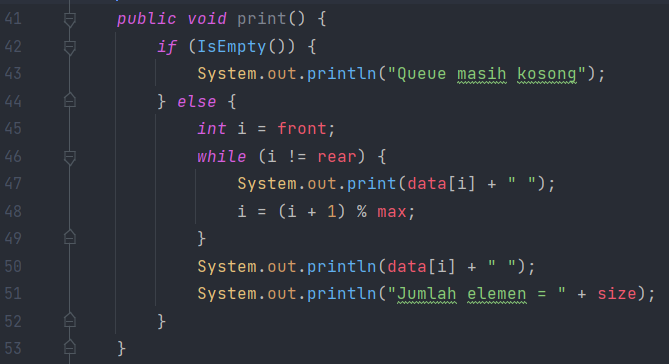
## Langkah 6

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



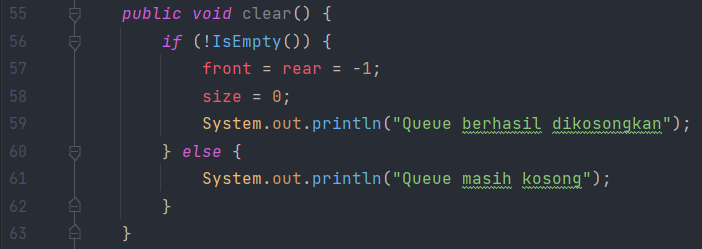
## Langkah 7

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



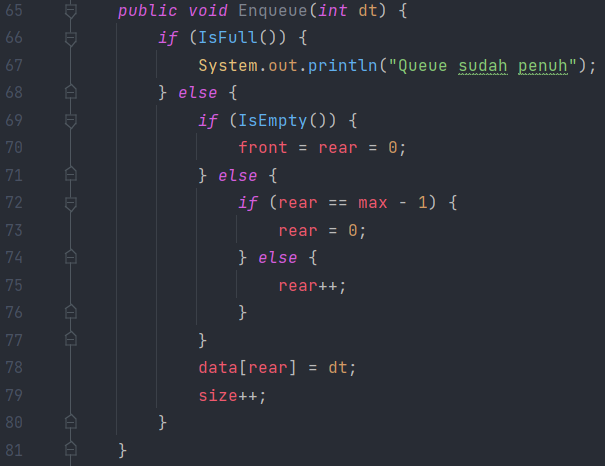
## Langkah 8

Buat method clear bertipe void untuk menghapus semua elemen pada queue



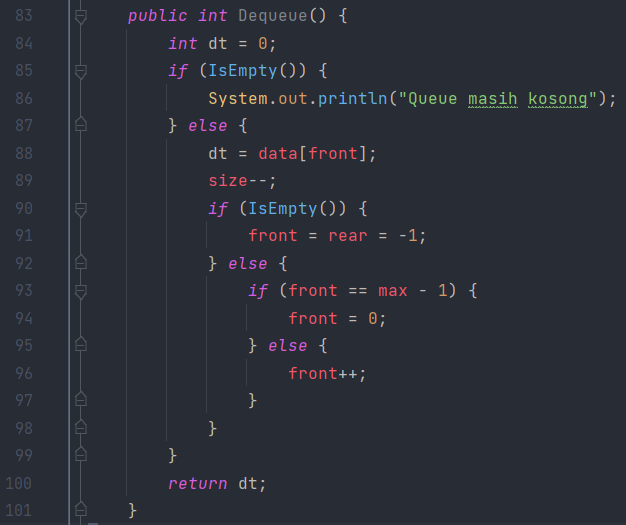
## Langkah 9

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



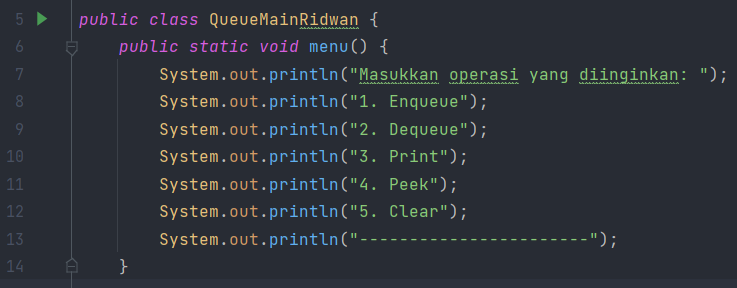
## Langkah 10

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



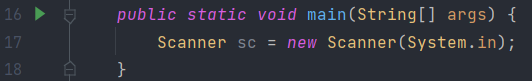
## Langkah 11

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



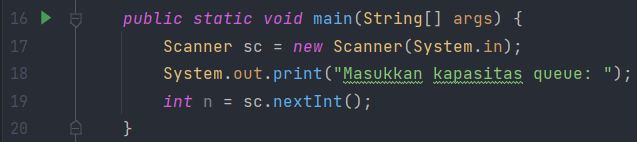
## Langkah 12

Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc



## Langkah 13

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



## Langkah 14

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



## Langkah 15

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



## 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

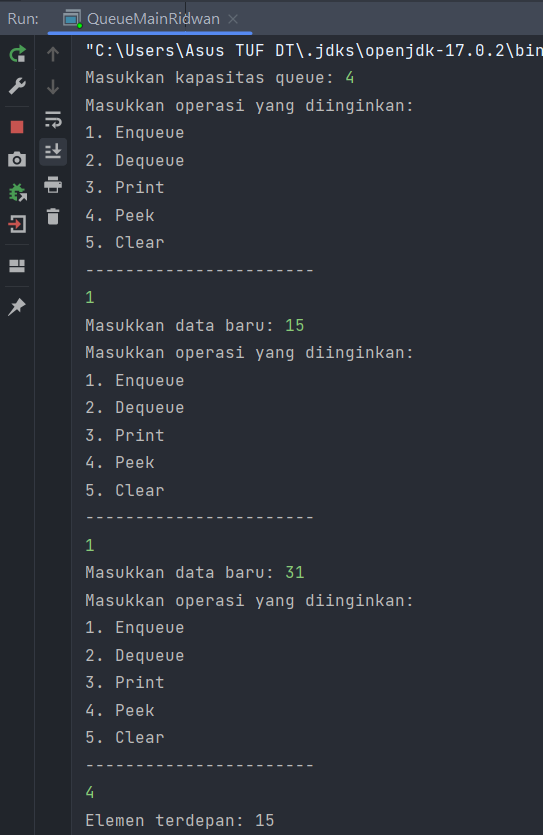


## Langkah 17

Compile dan jalankan class QueueMain, kemudian amati hasilnya.

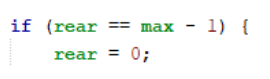
## Verifikasi Hasil Percobaan

Samakan hasil compile kode program Anda dengan gambar berikut ini

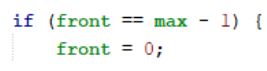


### Pertanyaan

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!



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



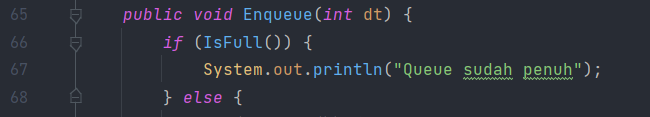
1. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
2. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!



1. Tunjukkan potongan kode program yang merupakan queue overflow!
2. 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!

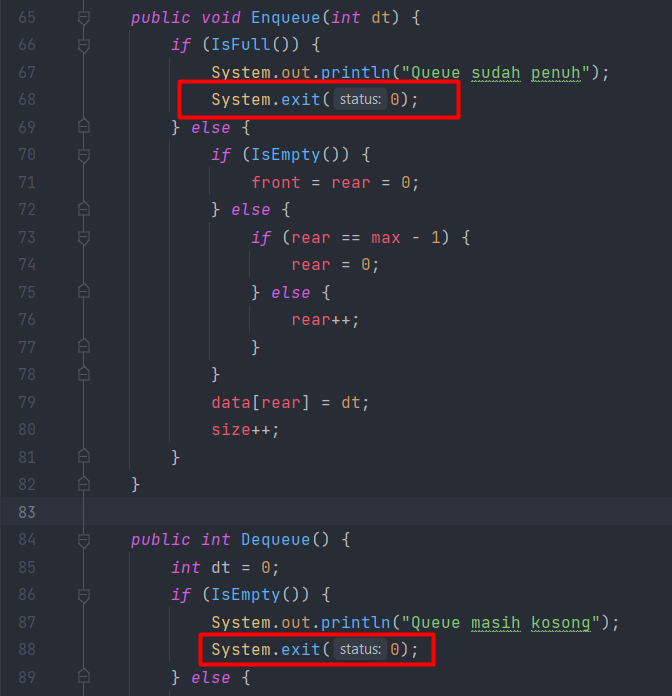
### Jawaban

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.

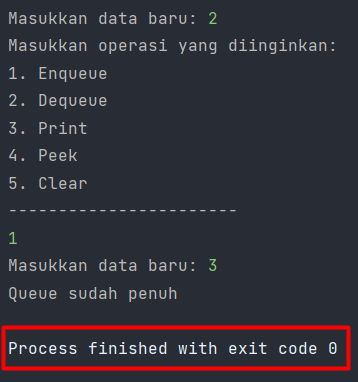


Pada kode tersebut dilakukan pengecekkan apakah queue sudah penuh/belum, jika sudah penuh maka tidak bisa diisi namun program tetap berjalan

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



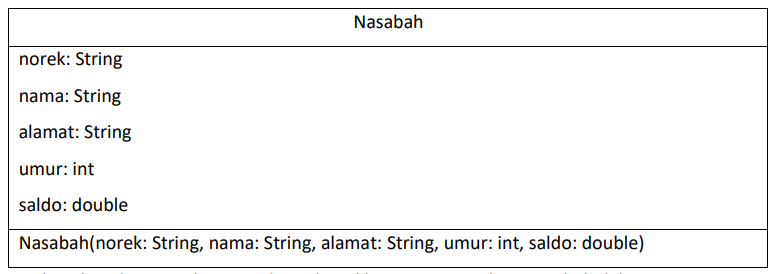
Output



# Praktikum 2

## Langkah 1

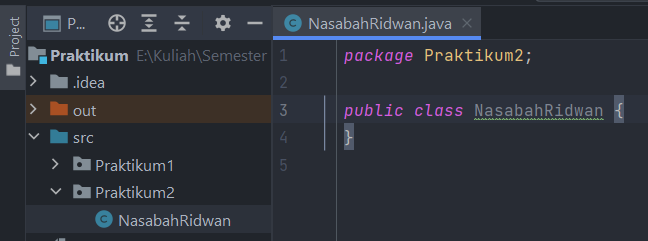
Perhatikan Diagram Class berikut ini:



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

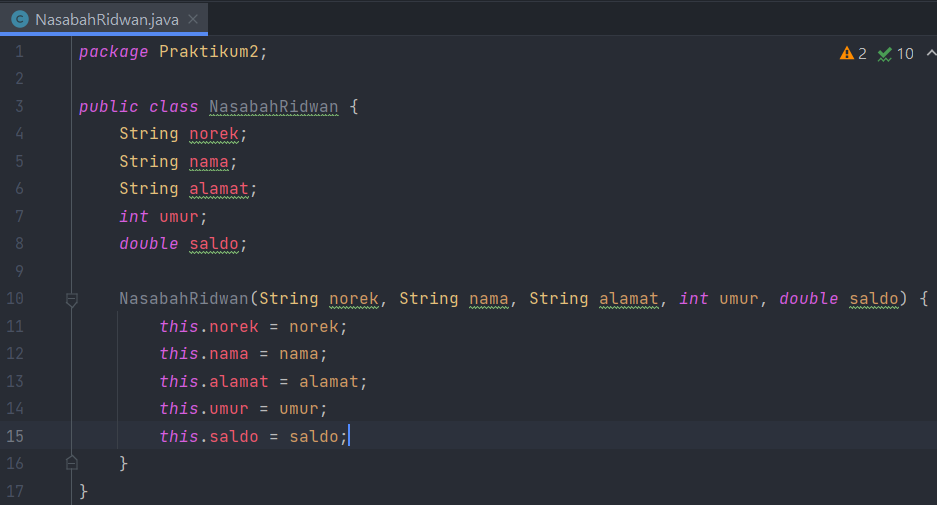
## Langkah 2

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



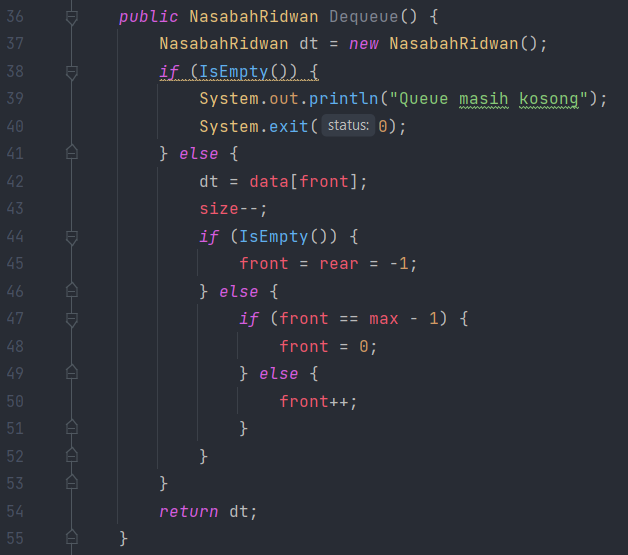
## Langkah 4

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.

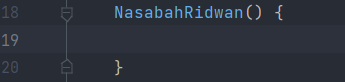
## Langkah 5

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.





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



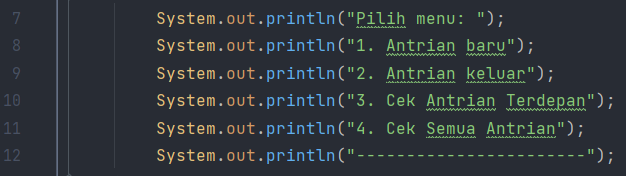
## Langkah 6

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.



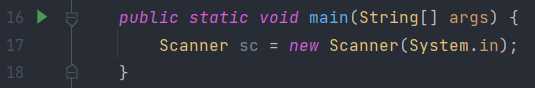
## Langkah 7

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



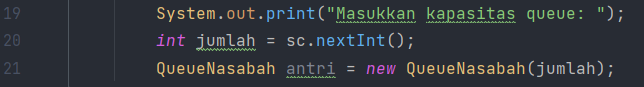
## Langkah 8

Buat fungsi main, deklarasikan Scanner dengan nama sc



## 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.



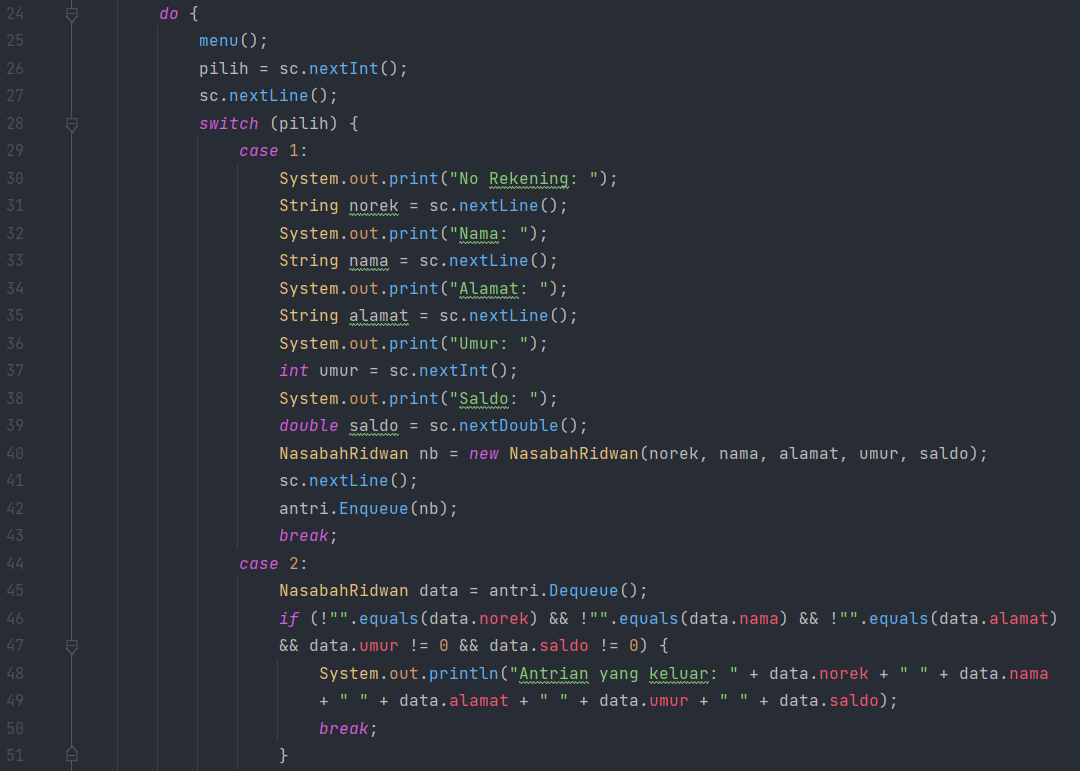
## Langkah 10

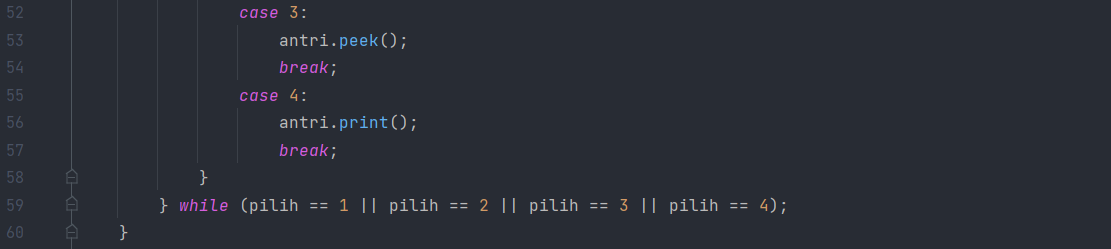
Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.



## Langkah 11

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





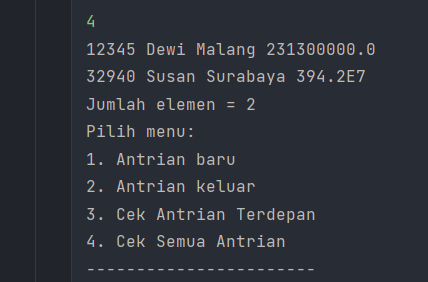
## Langkah 12

Compile dan jalankan class QueueMain, kemudian amati hasilnya.

## Verifikasi Hasil Percobaan

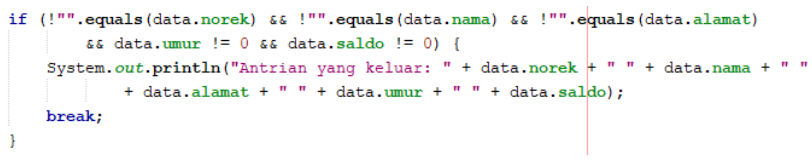
Samakan hasil compile kode program Anda dengan gambar berikut ini.



\

### Pertanyaan

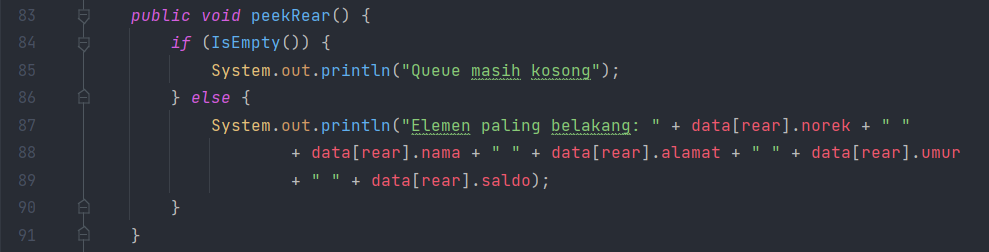
1. Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!



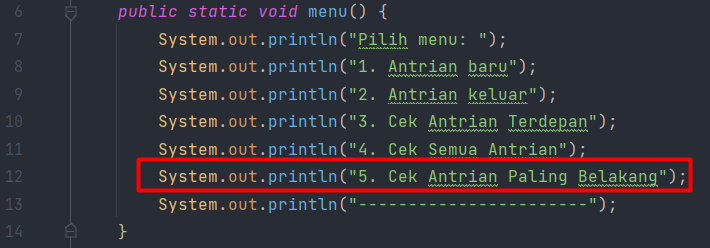
1. Lakukan modifikasi program dengan menambahkan method baru bernama peekRear pada class Queue yang digunakan untuk mengecek antrian yang berada di posisi belakang! Tambahkan pula daftar menu 5. Cek Antrian paling belakang pada class QueueMain sehingga method peekRear dapat dipanggil!

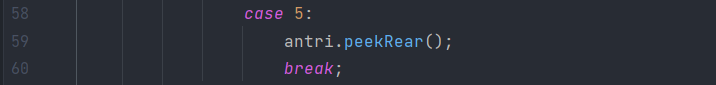
### Jawaban

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

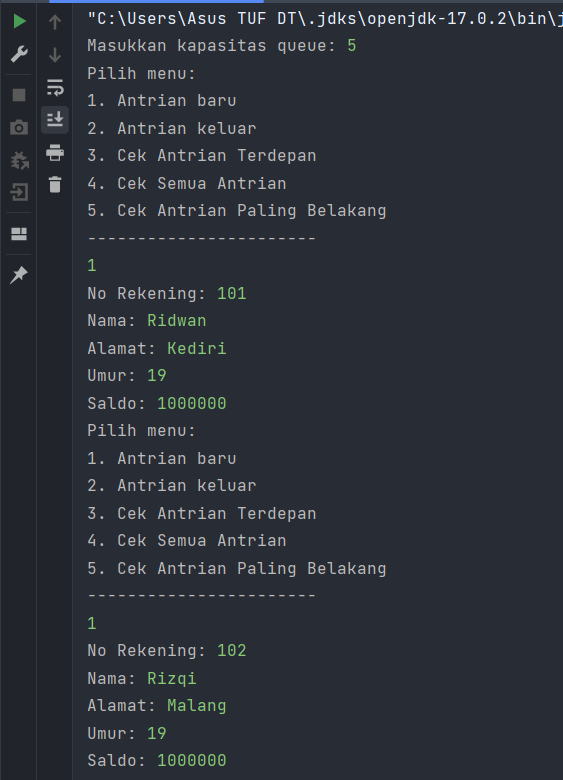


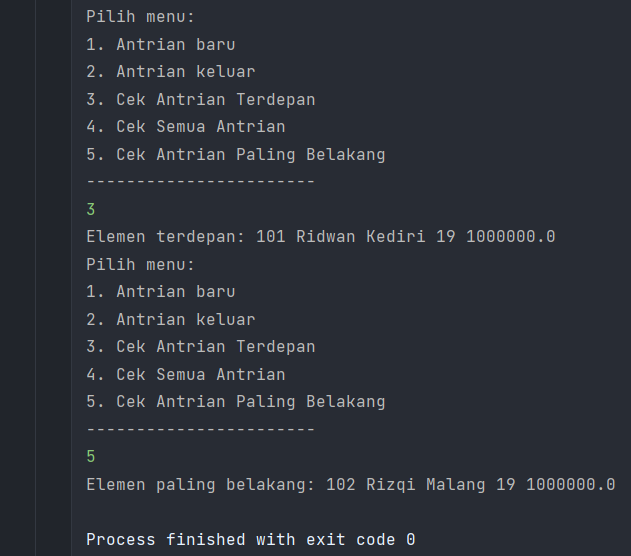
Modifikasi source code class QueueMainNasabah





Output





# Tugas

## Soal Nomor 1

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

1. 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

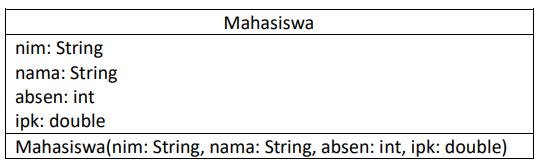
1. Method peekAt(position: int) : void

Untuk menampilkan data yang berada pada posisi (indeks) tertentu

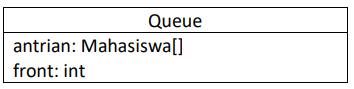
Sesuaikan daftar menu yang terdapat pada class QueueMain sehingga kedua method tersebut dapat dipanggil!

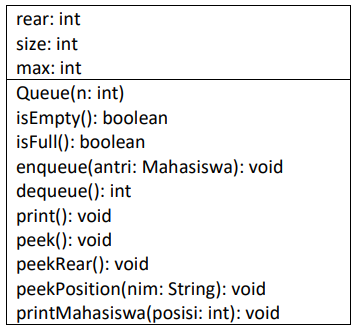
## Soal Nomor 2

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:



Class diagram Queue digambarkan sebagai berikut:



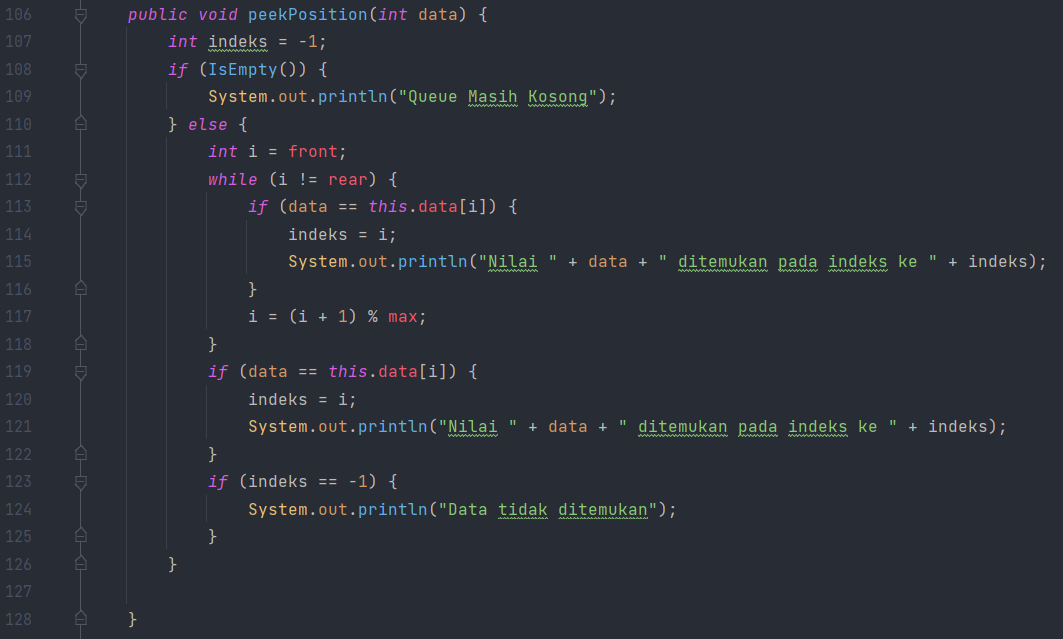


Keterangan:

* 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

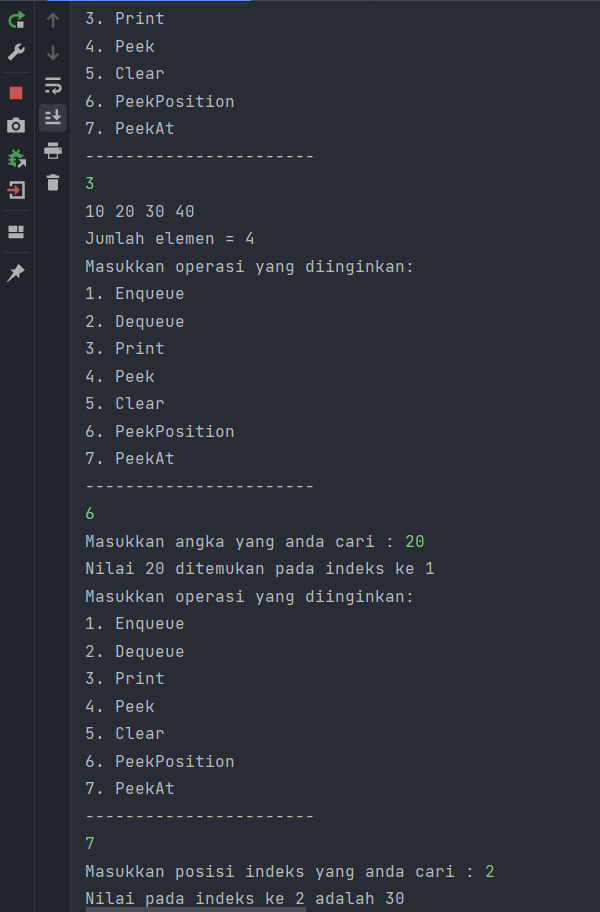


### Method peekAt



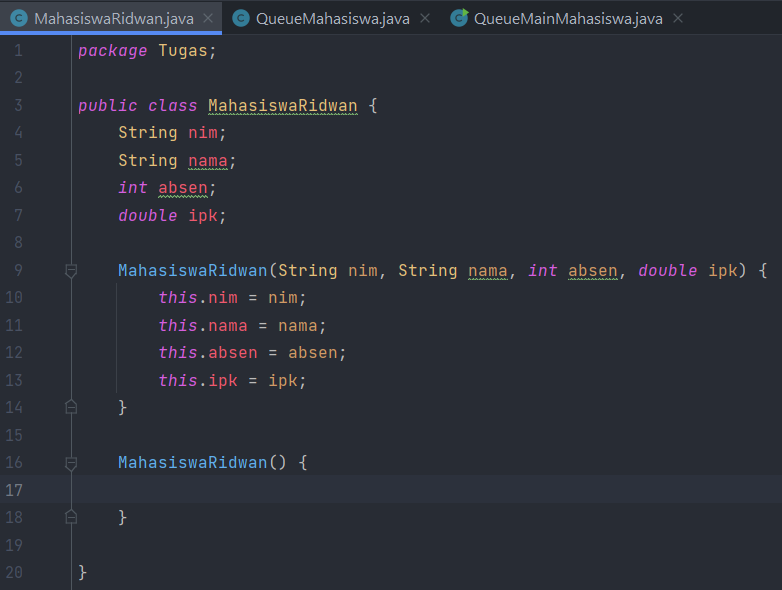
### Output

Sebelumnya sudah saya masukkan data ke dalam queue



## Jawaban Nomor 2

### Source code class MahasiswaRidwan

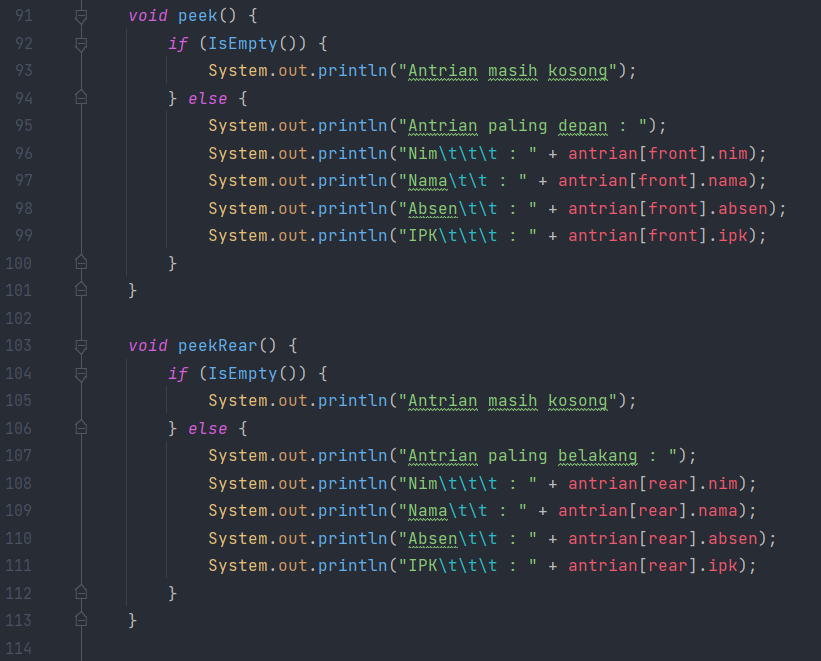


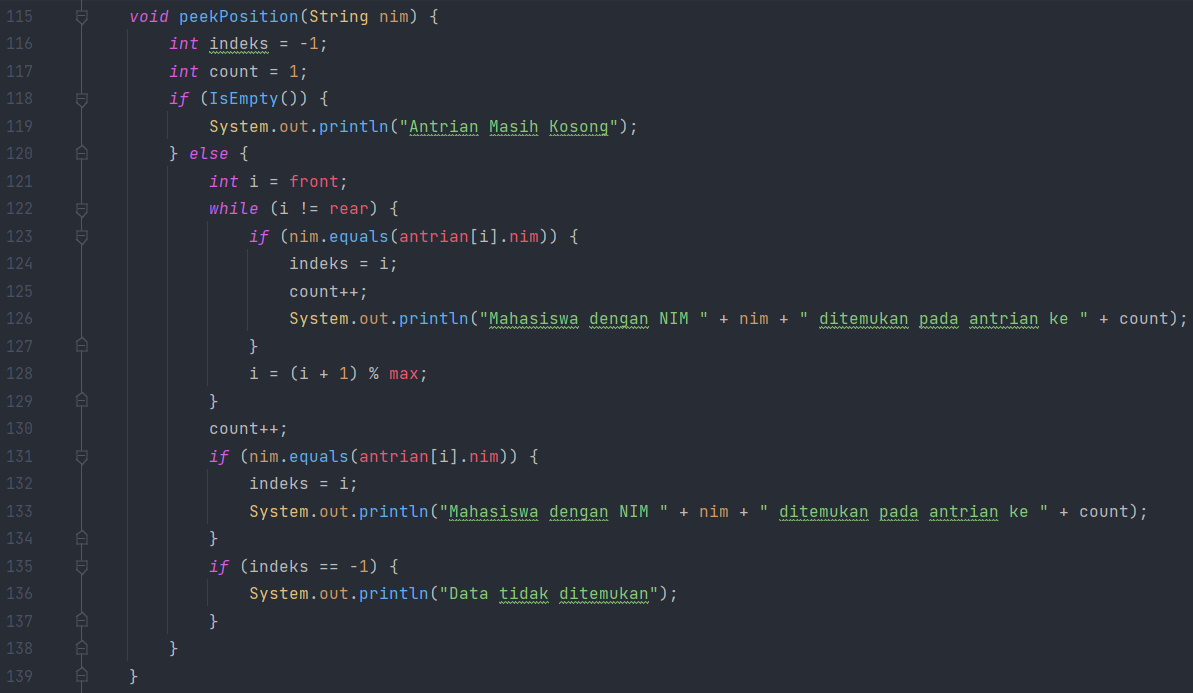
### Source code class QueueMahasiswa





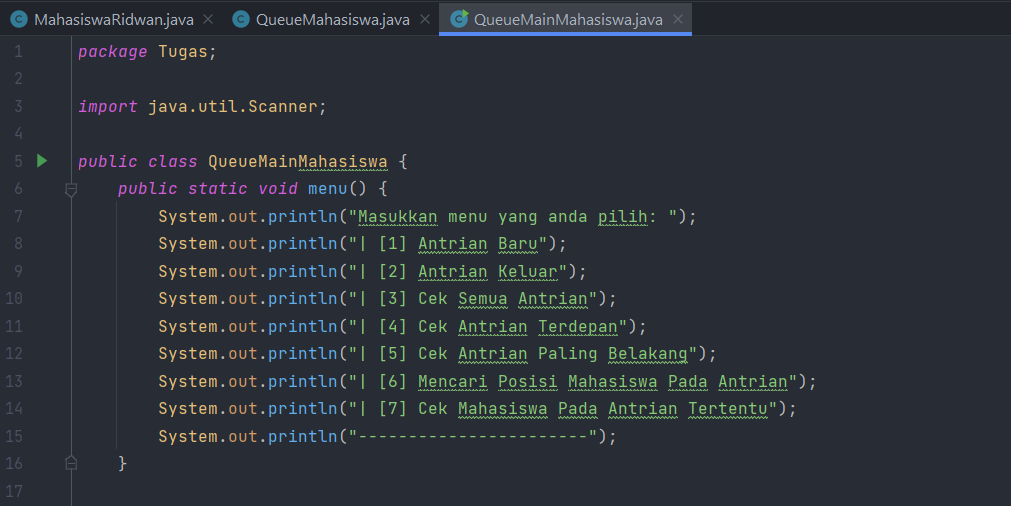




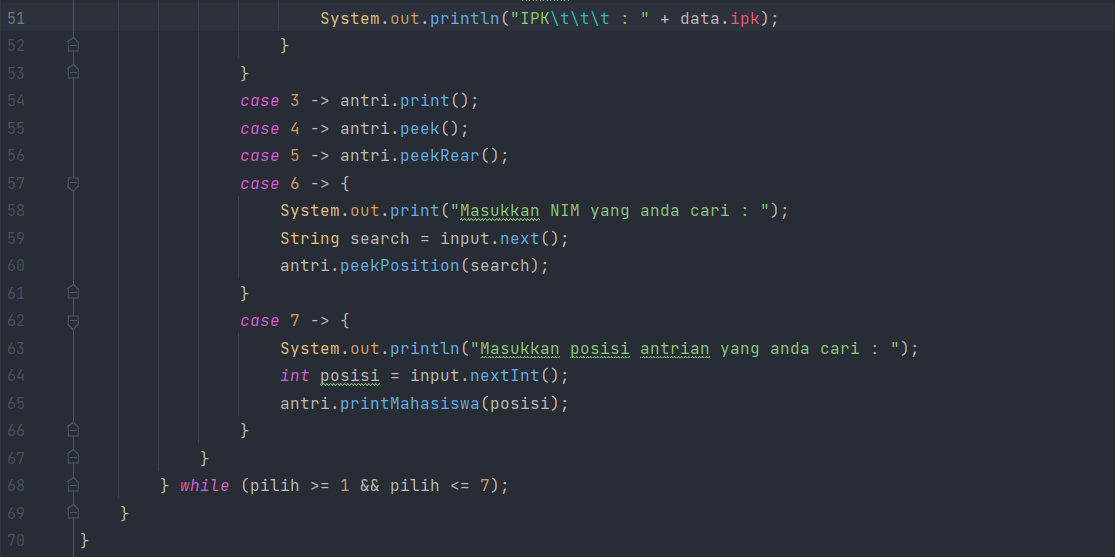




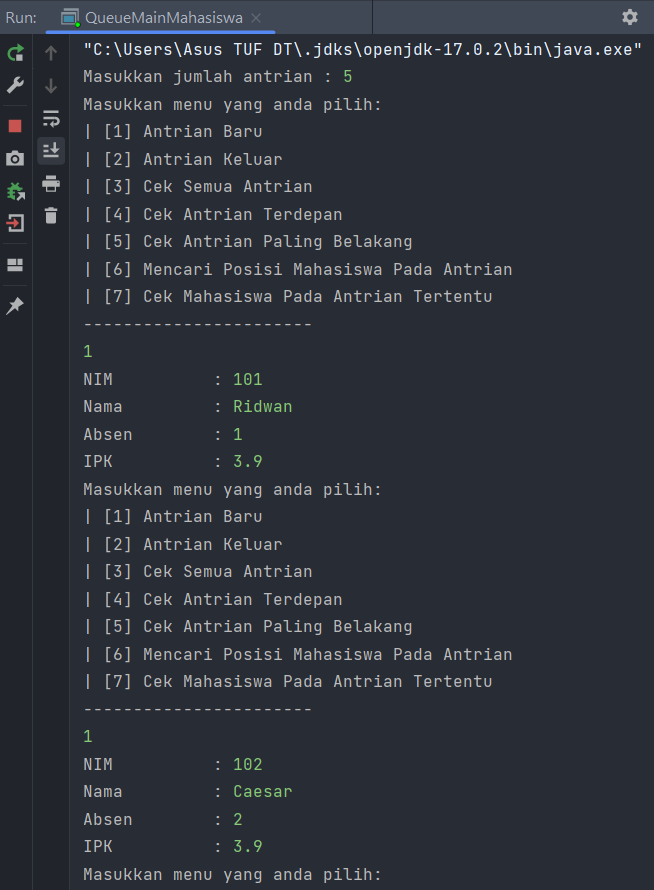
### Source code class QueueMainMahasiswa



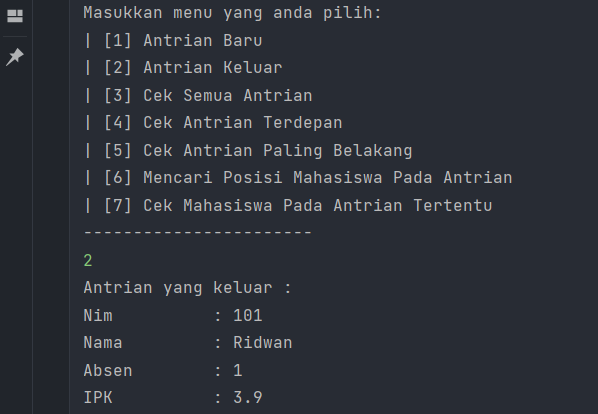




### Output Enqueue

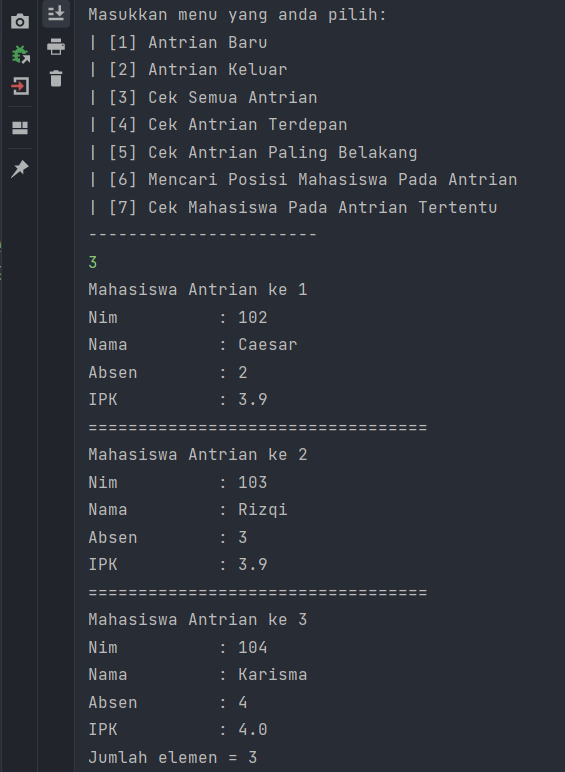


### Output Dequeue

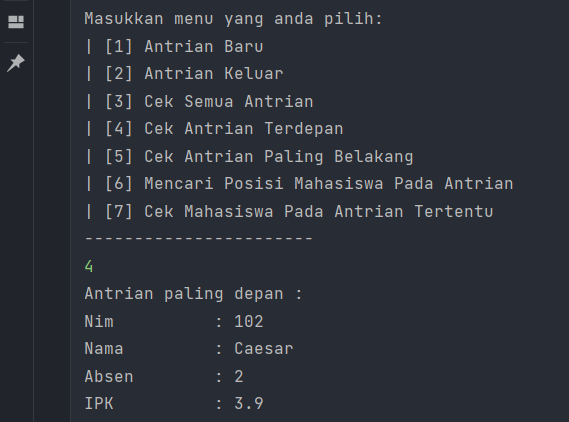


### Output Cek semua antrian

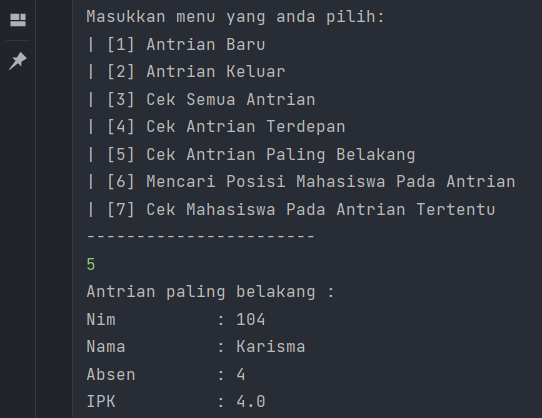
Sebelumnya sudah saya tambahkan antrian baru



### Output Cek antrian terdepan

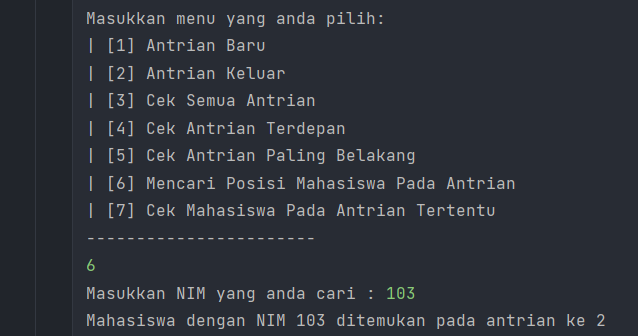


### Output Cek antrian paling belakang



### Output mencari posisi mahasiswa dalam antrian

Disini menampilkan antrian dalam kehidupan nyata bukan dari indeks array.



### Output cek mahasiswa dalam antrian tertentu

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

