**LAPORAN PRAKTIKUM JOBSHEET**

**QUEUE**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**



**Disusun Oleh :**

**Jami’atul Afifah (2341760102)**

**SIB-1F**

**PROGRAM STUDI D4 SISTEM INFOEMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

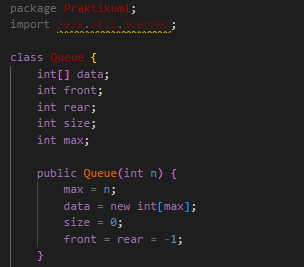
**PRAKTIKUM 1**

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 Jawa.

1. Buat package dengan nama Praktikum1, kemudian buat class baru dengan nama Queue.
2. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.



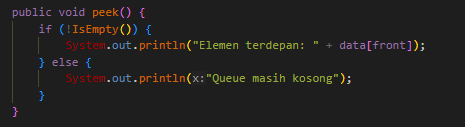
1. Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong.



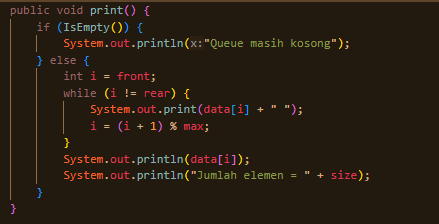
1. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.



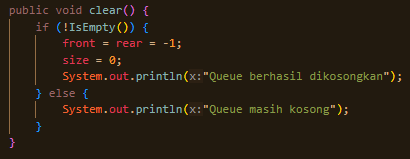
1. Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.



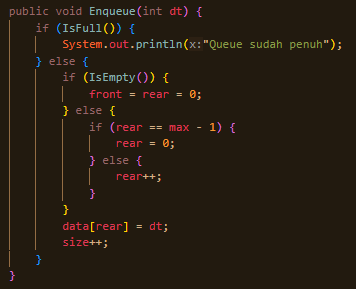
1. Buat method print bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.



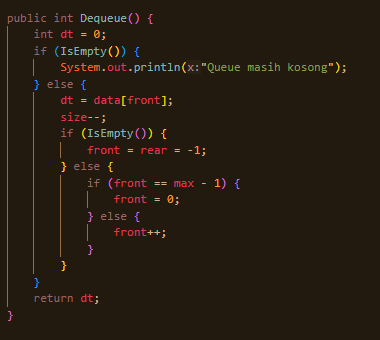
1. Buat method clear bertipe void untuk menghapus semua elemen pada queue



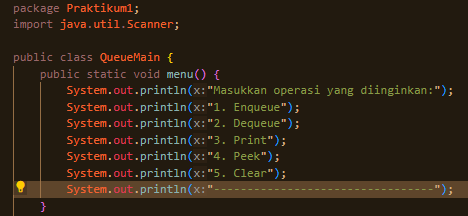
1. Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer



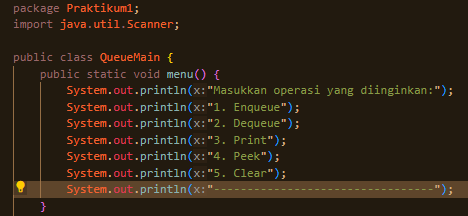
1. Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi paling depan



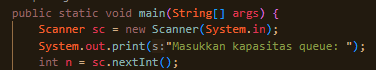
1. Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan.



1. Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc.



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



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

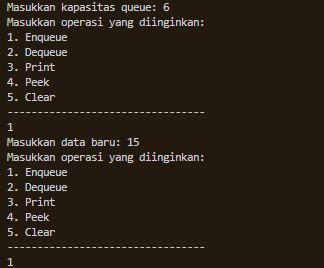
Queue Q = new Queue(n);

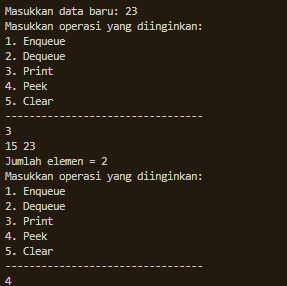
1. Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.
2. Lakukan perulangan menggunakan do-while untuk menjalankan program secara terus menerus sesuai masukan yang diberikan. Di dalam perulangan tersebut, terdapat pemilihan kondisi menggunakan switch-case untuk menjalankan operasi queue sesuai dengan masukan pengguna.

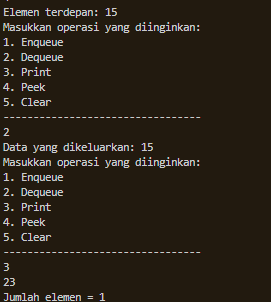


**Verifikasi Hasil Percobaan**

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?

Dalam struktur data queue, nilai awal atribut front dan rear yang disetel ke -1 menunjukkan bahwa queue tersebut kosong. Ini karena dalam representasi array, posisi -1 menunjukkan bahwa tidak ada elemen yang ada.

Sementara itu, atribut size disetel ke 0 karena pada awalnya tidak ada elemen yang disimpan di dalam queue tersebut. Dengan demikian, ketika queue dibuat, tidak ada elemen yang dianggap sebagai elemen pertama atau terakhir di dalamnya, sehingga front dan rear disetel ke -1 dan size disetel ke 0.

Kemudian, saat elemen dimasukkan ke dalam queue, nilai-nilai front, rear, dan size akan diperbarui sesuai dengan operasi-operasi enqueue dan dequeue.

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



Potongan kode tersebut memiliki tujuan dan kegunaan sebagai berikut:

* rear == max - 1: Potongan kode ini memeriksa apakah posisi rear saat ini berada pada indeks terakhir dari array data. Jika demikian, artinya rear sudah berada di ujung maksimum array dan tidak ada lagi slot kosong di belakangnya untuk menambahkan elemen baru.
* Jika kondisi tersebut terpenuhi, maka rear direset ke 0: Jika rear sudah berada di ujung maksimum array, maka elemen selanjutnya harus dimasukkan di awal array (indeks 0), karena queue menggunakan konsep circular array. Dengan mereset rear ke 0, kita akan kembali ke awal array dan mulai menambahkan elemen dari awal.

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



Potongan kode tersebut memiliki maksud dan kegunaan sebagai berikut:

* front == max - 1: Potongan kode ini memeriksa apakah posisi front saat ini berada pada indeks terakhir dari array data. Jika kondisi ini terpenuhi, artinya elemen yang akan di-dequeue adalah elemen terakhir dalam array data.
* Jika kondisi tersebut terpenuhi, maka front direset ke 0: Dalam implementasi circular queue, setelah elemen terakhir di-dequeue, kita perlu kembali ke awal array untuk mencari elemen selanjutnya yang akan menjadi front dari queue. Dengan mereset front ke 0, kita menandakan bahwa elemen pertama dalam array akan menjadi elemen pertama dalam queue setelah proses dequeue.

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

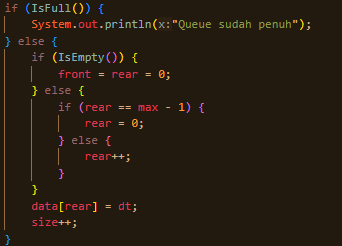
Variabel `i` dimulai dari `front` karena kita ingin mencetak elemen-elemen dalam urutan yang benar sesuai dengan struktur queue. Dengan memulai dari `front`, kita mencetak elemen-elemen dari front queue hingga rear queue, sesuai dengan urutan mereka dalam array data circular.

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

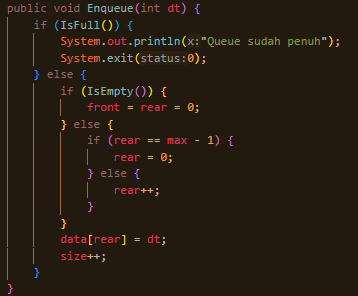


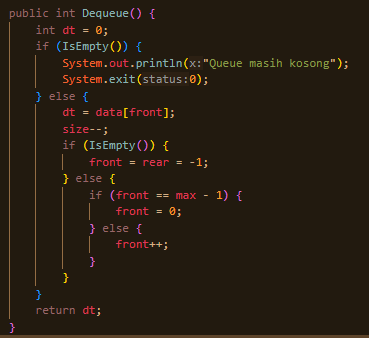
Pada method print, variabel `i` tidak dimulai dari 0 (`i=0`), melainkan dari `front`, karena kita ingin mencetak elemen-elemen queue yang sesungguhnya, yang mungkin dimulai dari indeks selain 0 dalam array circular.

1. Tunjukkan potongan kode program yang merupakan queue overflow!

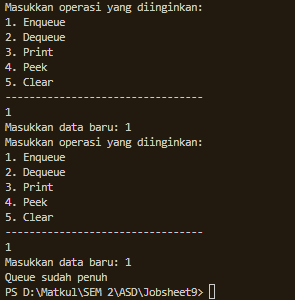


1. Pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan!





Hasil Modifikasi :



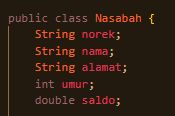
**PRAKTIKUM 2**

1. Perhatikan Diagram Class berikut ini:

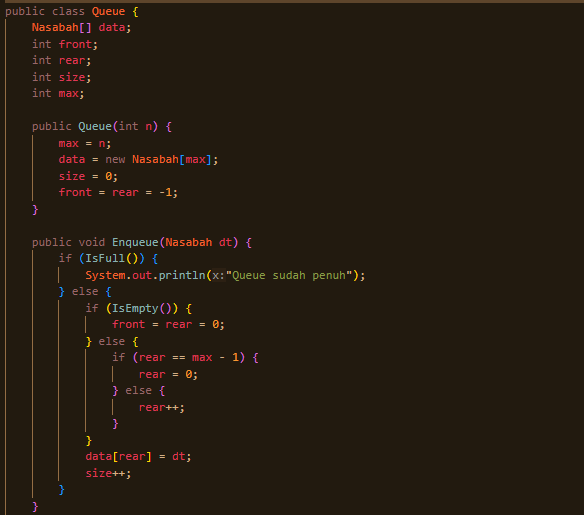


Berdasarkan diagram class tersebut, akan dibuat program class Nasabah dalam Java.

1. Buat package dengan nama Praktikum2, kemudian buat class baru dengan nama Nasabah.
2. Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

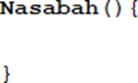


1. Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini. Karena pada Praktikum 1, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada Praktikum 2 data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class Queue tersebut.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

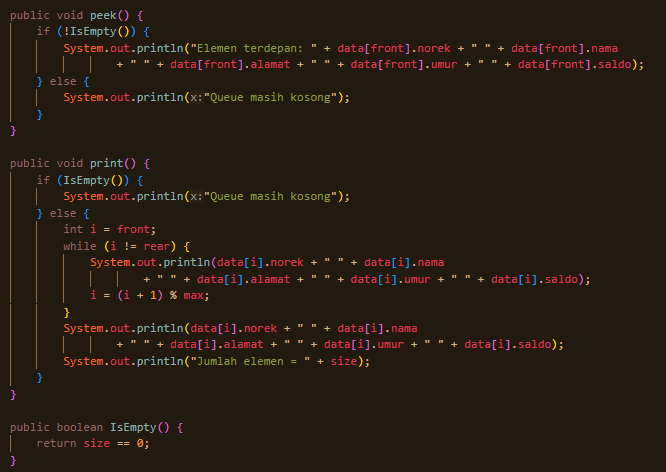




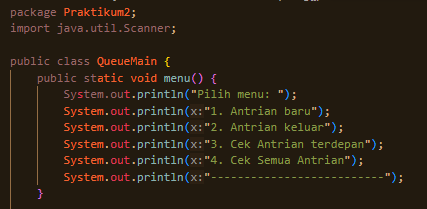
Baris program Nasabah dt = new Nasabah(); akan ditandai sebagai error, untuk mengatasinya, tambahkan konstruktor default di dalam class Nasabah.



1. 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.



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



1. Buat fungsi main, deklarasikan Scanner dengan nama sc
2. Buat variabel max untuk menampung kapasitas elemen pada queue. Kemudian lakukan instansiasi objek queue dengan nama antri dan nilai parameternya adalah variabel jumlah.

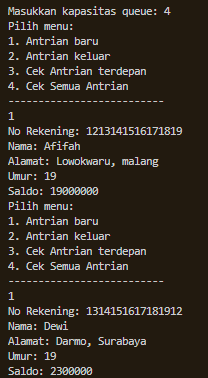


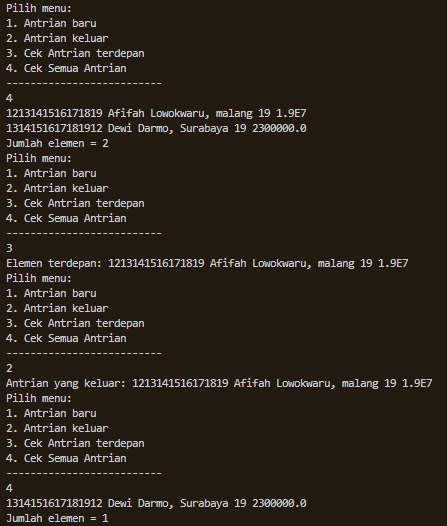
1. Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.
2. Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.



**Verifikasi Hasil Percobaan**

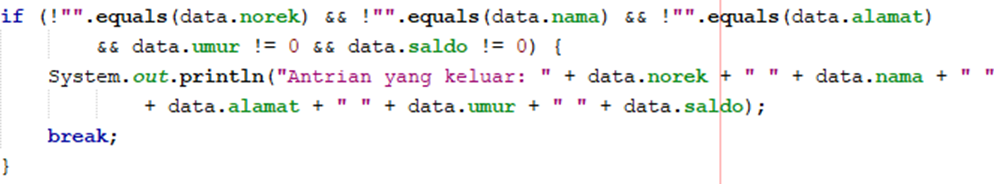
Samakan hasil compile kode program Anda dengan gambar berikut ini





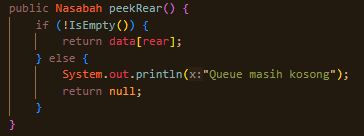
**Pertanyaan**

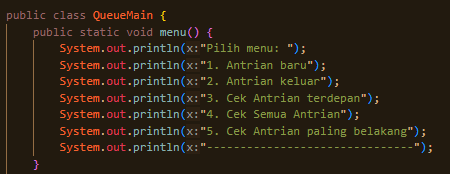
1. Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!

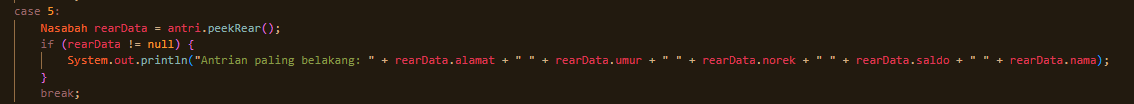


Potongan kode tersebut digunakan untuk memastikan bahwa data yang di-dequeue dari antrian memiliki nilai yang valid sebelum dicetak. Jika semua atribut data seperti nomor rekening, nama, alamat, umur, dan saldo tidak kosong atau tidak nol, maka data tersebut dianggap valid dan dicetak sebagai "Antrian yang keluar" bersama dengan informasi terkait. Setelah itu, pernyataan `break` digunakan untuk keluar dari loop switch case.

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!

**Menambahkan peekRear**  


**Menambahkan Cek antrian paling belakang**  




**Hasil Modifikasi:**



**TUGAS**

Buatlah program antrian untuk mengilustrasikan antrian pasien di sebuah klinik. Ketika seorang pasien akan mengantri, maka dia harus mendaftarkan nama, nomor identitas, jenis kelamin dan umur seperti yang digambarkan pada Class diagram berikut:

|  |
| --- |
| Pasien |
| nama: String noID: int jenisKelamin: char  umur: int |
| Pasien (nama: String, noID: int, jenisKelamin: char, umur: int) |

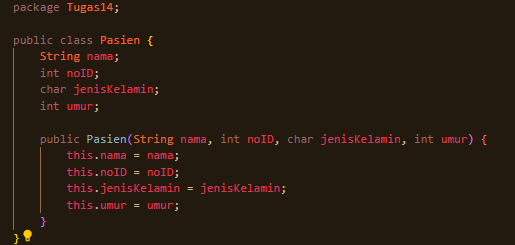
|  |
| --- |
| Queue |
| antrian: Pasien[] front: int  rear: int size: int max: int |
| Queue(n: int) isEmpty(): boolean isFull(): boolean  enqueue(antri: Pasien): void dequeue(): int  print(): void peek(): void peekRear(): void  peekPosition(nama: String): void daftarPasien(): void |

Keterangan method:

* Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada Praktikum
* Method peek(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling depan
* Method peekRear(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling belakang
* Method peekPosition(): digunakan untuk menampilkan seorang pasien (berdasarkan nama) posisi antrian ke berapa
* Method daftarPasien(): digunakan untuk menampilkan data seluruh pasien

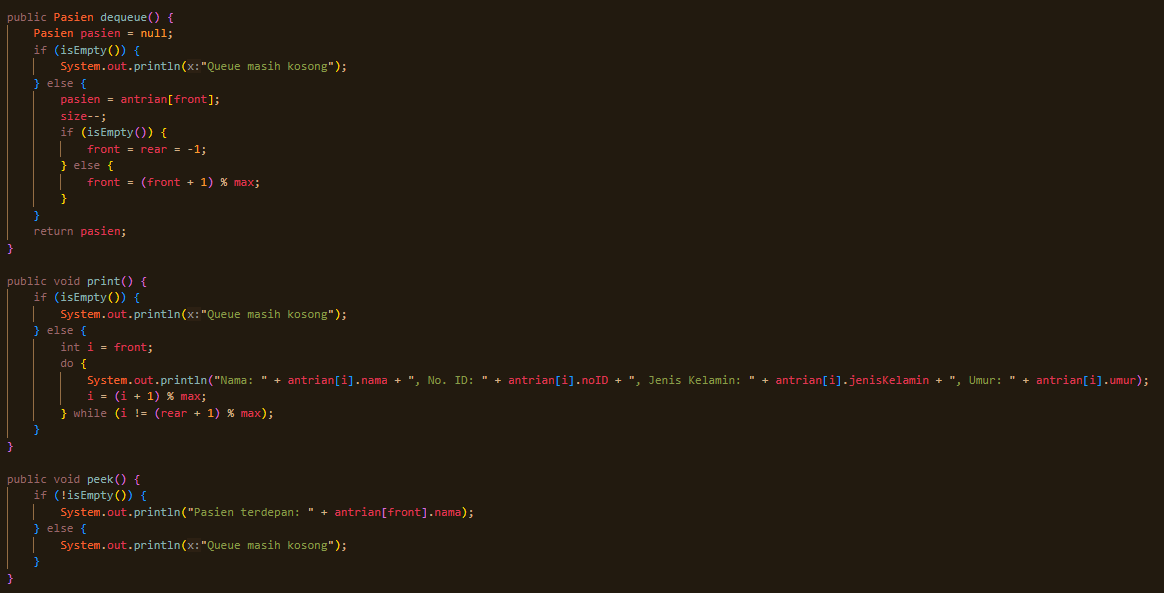
Buat Folder Tugas14 yang berisi:

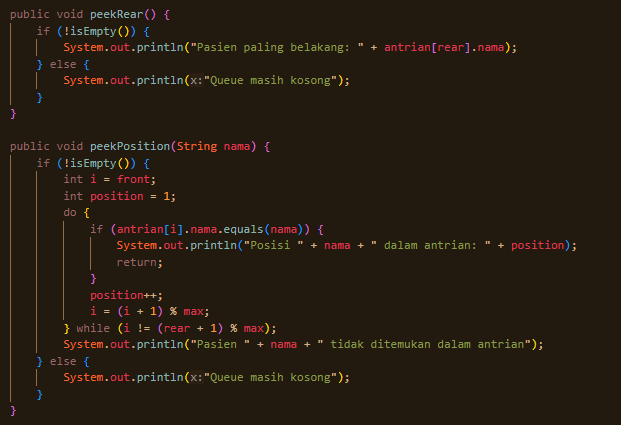
Pasien.java

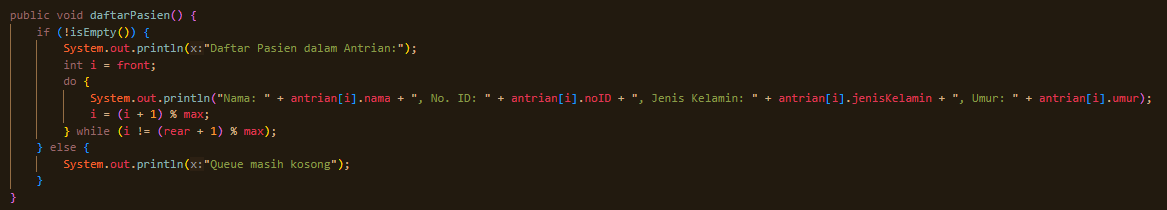


Queue.java

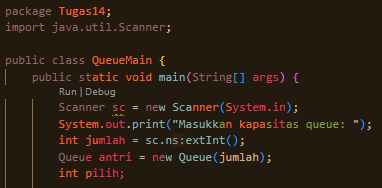


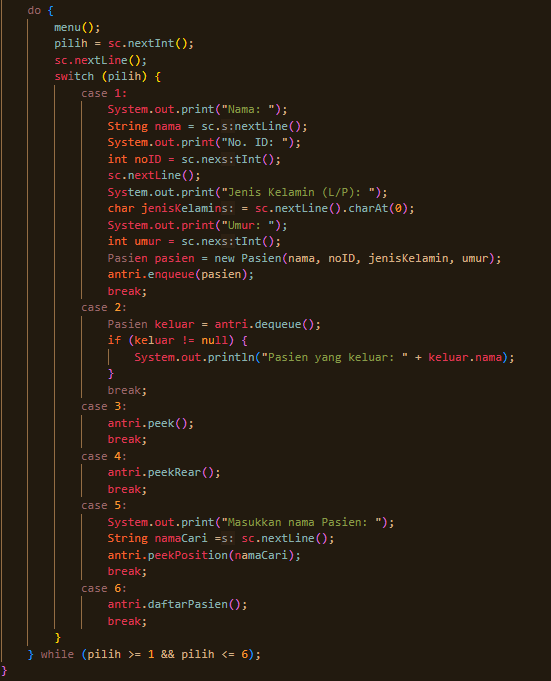


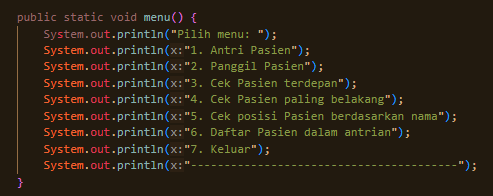




QueueMain.java







**Hasil Program:**

