Nama : Ahmad Franklyn Bima Aquilla

Kelas : SIB 1F – 01

NIM : 2341760027

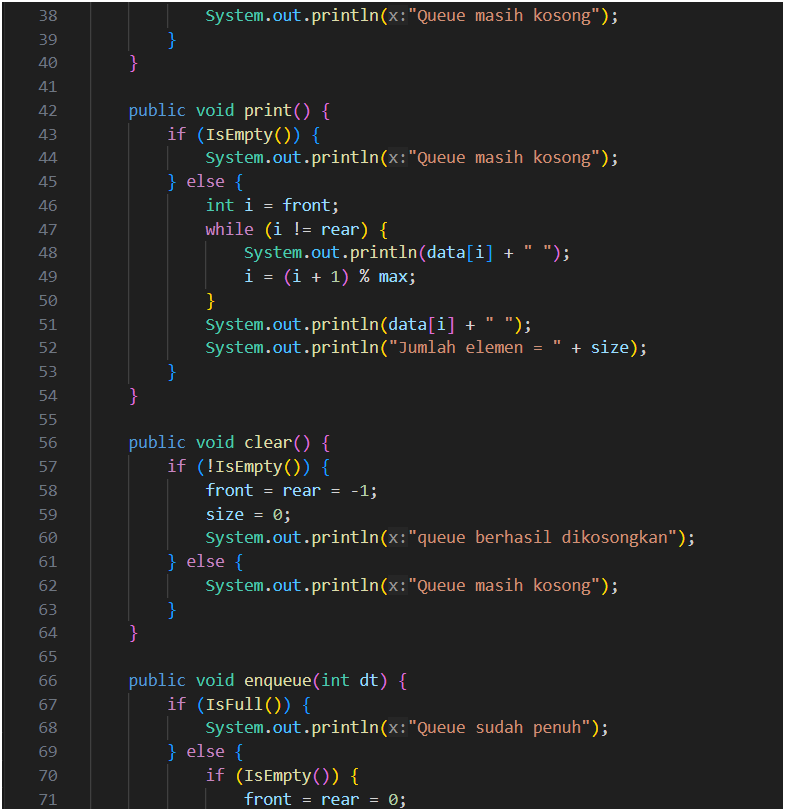
Matkul : Praktikum Algoritma dan Struktur Data

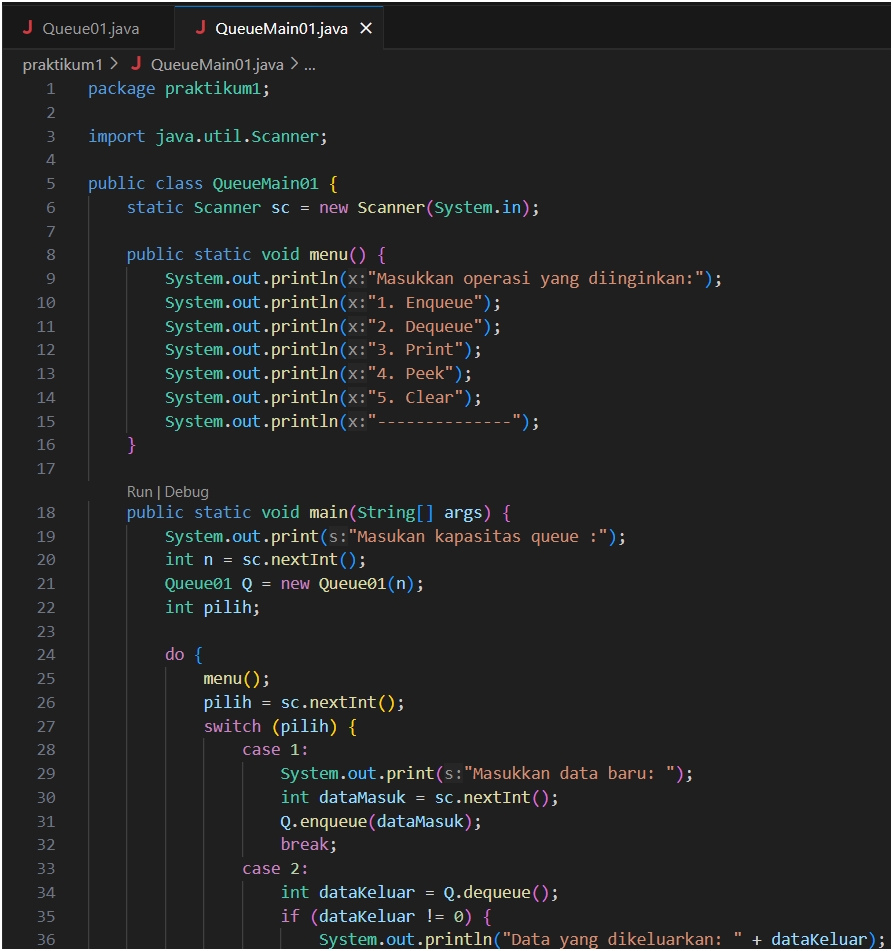
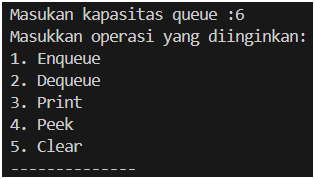
GitHub : <https://github.com/JustQuill25/S2PraktikumASDJS10>

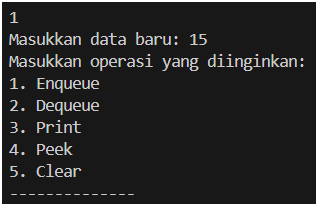
**JOBSHEET 10**

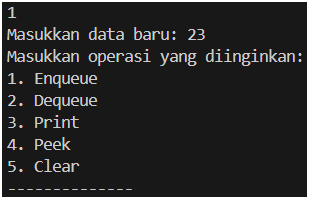
( Praktikum ASD – Queue )

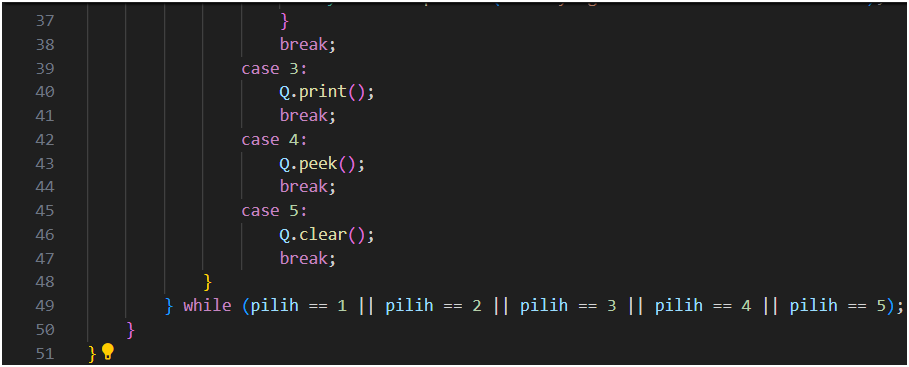
***Code :   
Class Queue01***

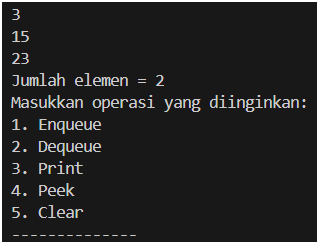


***Class MainQueue01 Output :***









# Pertanyaan

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

* Pada konstruktor, nilai awal atribut front dan rear diatur ke -1 untuk menandakan bahwa antrian (queue) awalnya kosong. Artinya tidak ada elemen yang ada dalam antrian pada saat awal pembuatannya. Sementara itu, nilai awal atribut size diatur ke 0 karena ukuran antrian pada awalnya memang kosong. Meskipun tidak ada elemen dalam antrian, menyimpan nilai 0 pada atribut size membantu dalam mengontrol dan memperbarui jumlah elemen yang ada dalam antrian ketika elemen ditambahkan atau dihapus.

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



* Kode “if (rear == max - 1) { rear = 0; }” dalam metode Enqueue memiliki maksud dan kegunaan untuk menangani kondisi ketika rear mencapai batas maksimum antrian (queue).

Kode tersebut berfungsi untuk mengecek apakah posisi rear saat ini sudah berada di posisi maksimum yang dapat diindeks dalam array yang merepresentasikan antrian. Jika rear sudah mencapai batas maksimum (yaitu max - 1), artinya array yang digunakan untuk menyimpan elemen antrian sudah penuh dan rear harus "dikembalikan" ke posisi awal array untuk menambahkan elemen baru.

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



* Kode if (front == max - 1) { front = 0; } dalam metode Dequeue memiliki maksud dan kegunaan yang serupa dengan potongan kode dalam metode Enqueue, namun diaplikasikan pada atribut front.

Kode tersebut berfungsi untuk mengecek apakah posisi front saat ini sudah berada di posisi maksimum yang dapat diindeks dalam array yang merepresentasikan antrian. Jika front sudah mencapai batas maksimum (yaitu max - 1), itu berarti bahwa array yang digunakan untuk menyimpan elemen antrian sudah mencapai batasnya di ujung depan, dan elemen yang ditunjuk oleh front harus "dikembalikan" ke posisi awal array.

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

* Pada metode print, proses perulangan dimulai dari posisi front bukan dari indeks 0 karena struktur data queue mungkin memiliki elemen-elemen yang telah dihapus dari depan antrian sebelumnya. Dengan memulai perulangan dari posisi front, kita memastikan bahwa kita hanya mencetak elemen yang masih valid dalam antrian dan melewati elemen yang mungkin sudah dihapus sebelumnya.

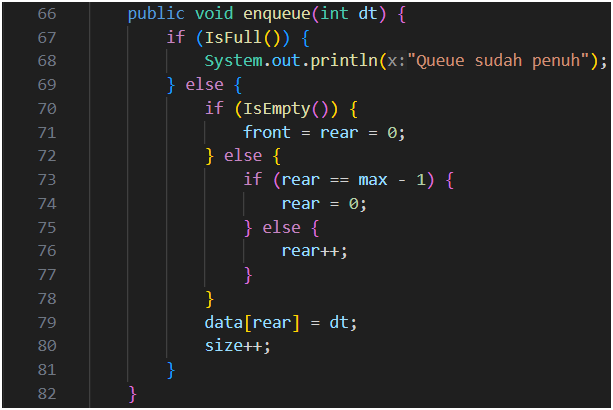
Ketika antrian diimplementasikan sebagai circular queue, elemen-elemen yang dihapus dari depan antrian membuat posisi front maju ke depan di dalam array. Dengan demikian, elemen-elemen yang masih tersisa dalam antrian akan dimulai dari posisi front hingga posisi rear atau hingga posisi max - 1 jika rear berada sebelumnya di depan front dalam array.

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

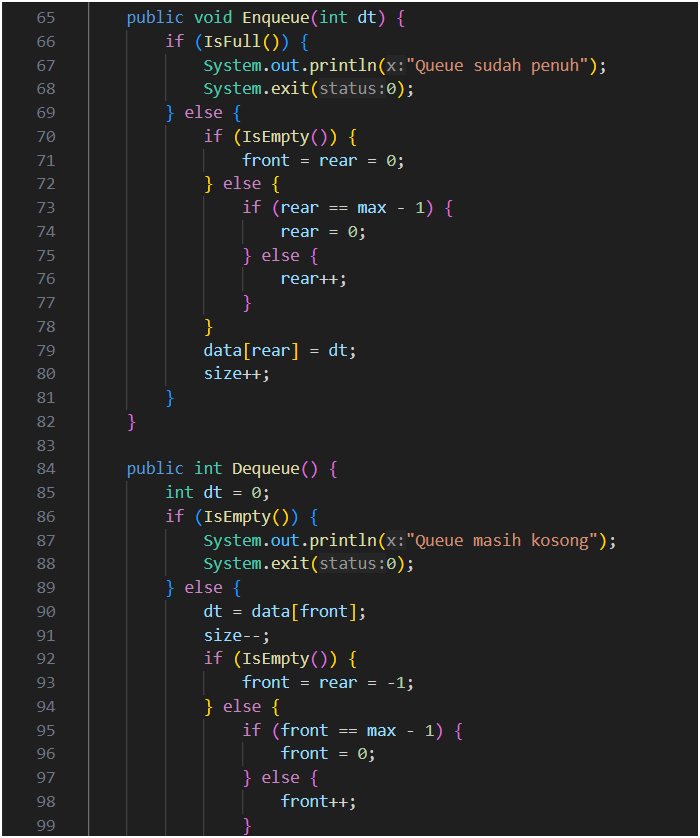


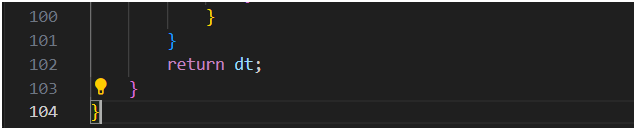
* Kode “i = (i + 1) % max;” pada metode print digunakan untuk mengatur iterasi perulangan melalui array yang merepresentasikan antrian secara sirkular. Dalam struktur data circular queue, ketika mencapai akhir array dan ingin melanjutkan iterasi ke elemen pertama dari antrian yang masih ada, kita perlu "memutar" indeks kembali ke posisi awal array. Potongan kode “(i + 1) % max” dilakukan untuk mencapai efek tersebut.

1. Tunjukkan potongan kode program yang merupakan queue overflow!

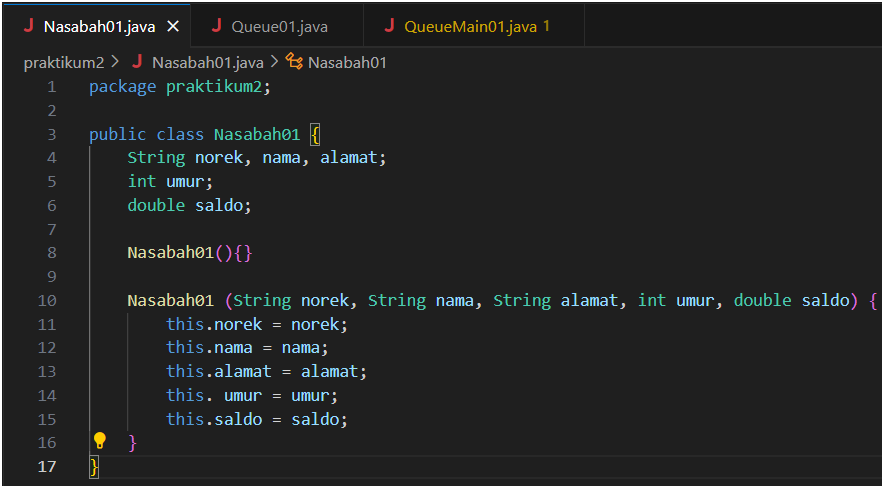


1. Pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan!

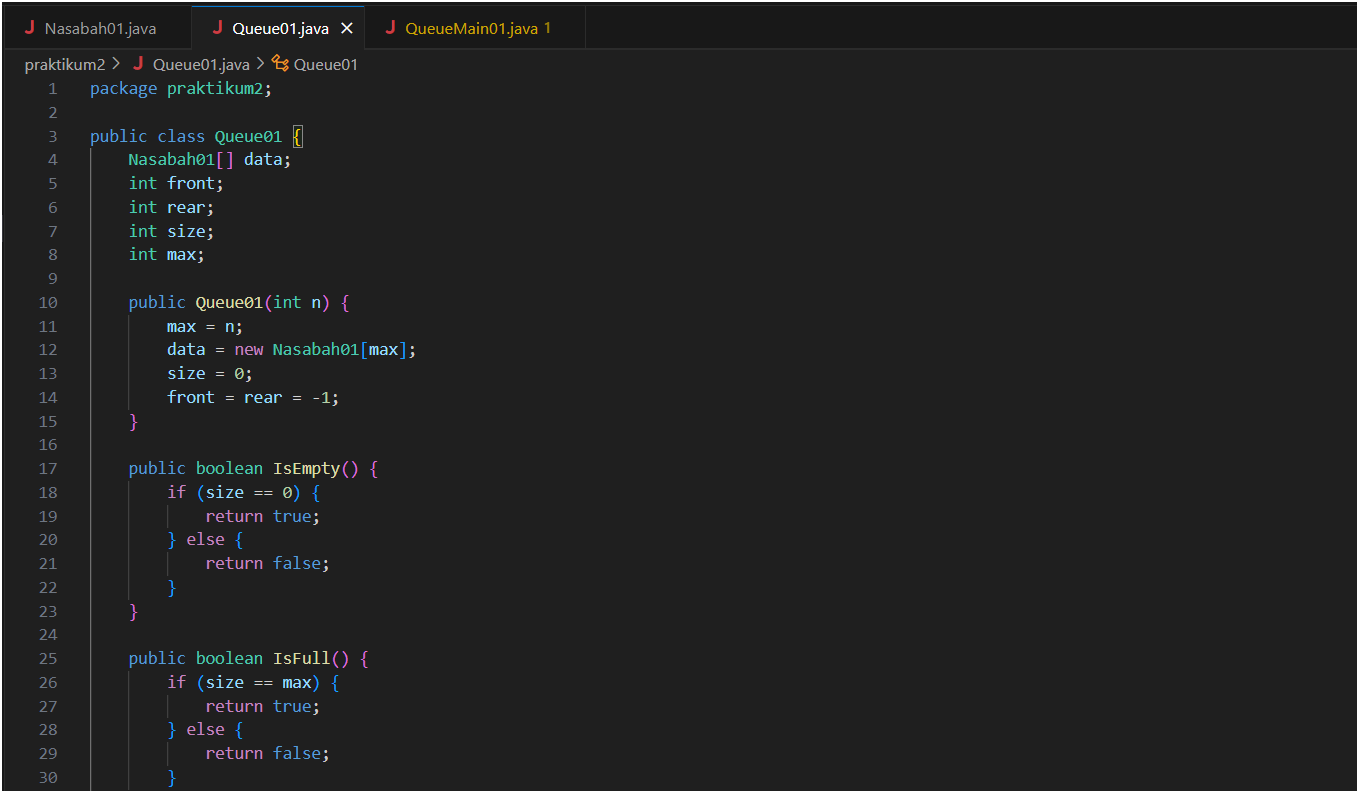
* 

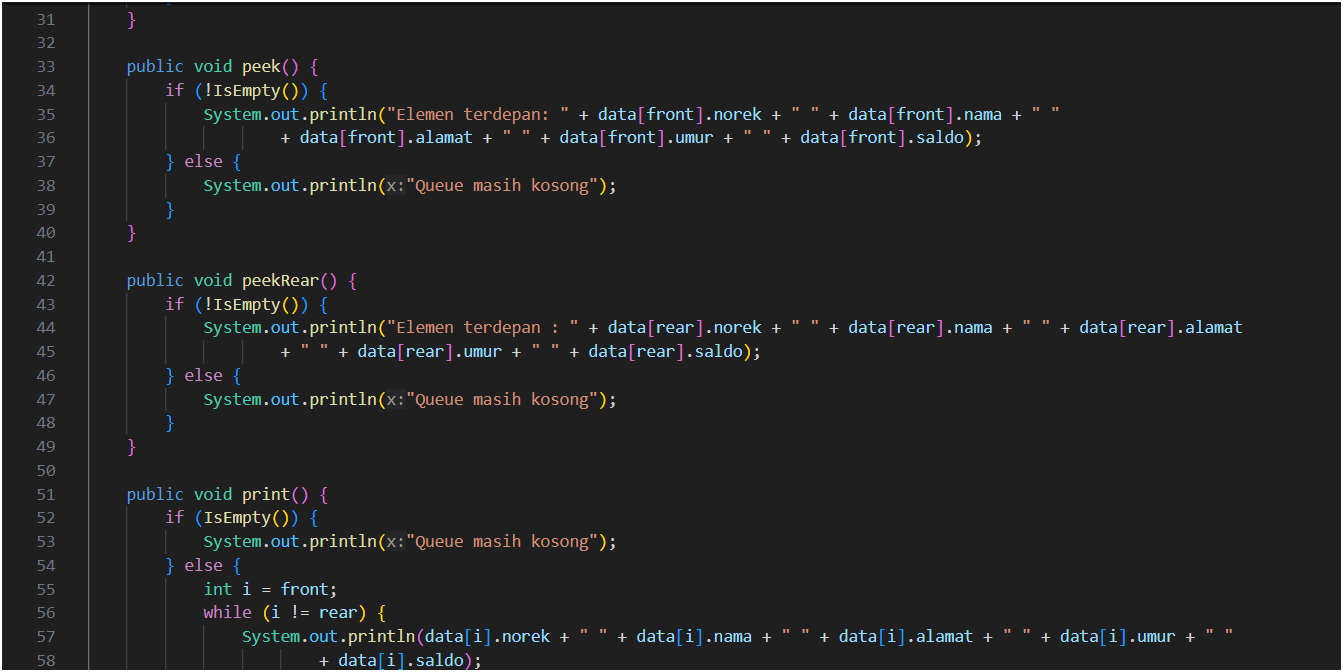


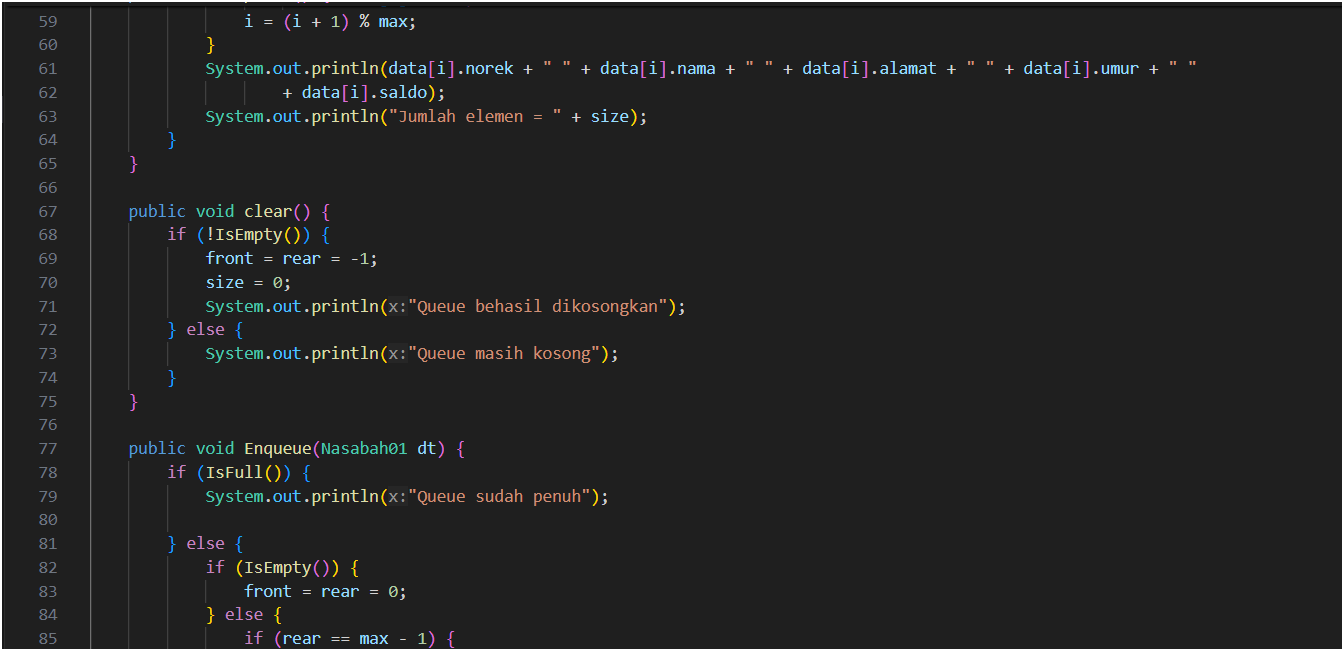
***Code :   
Class Nasabah01***

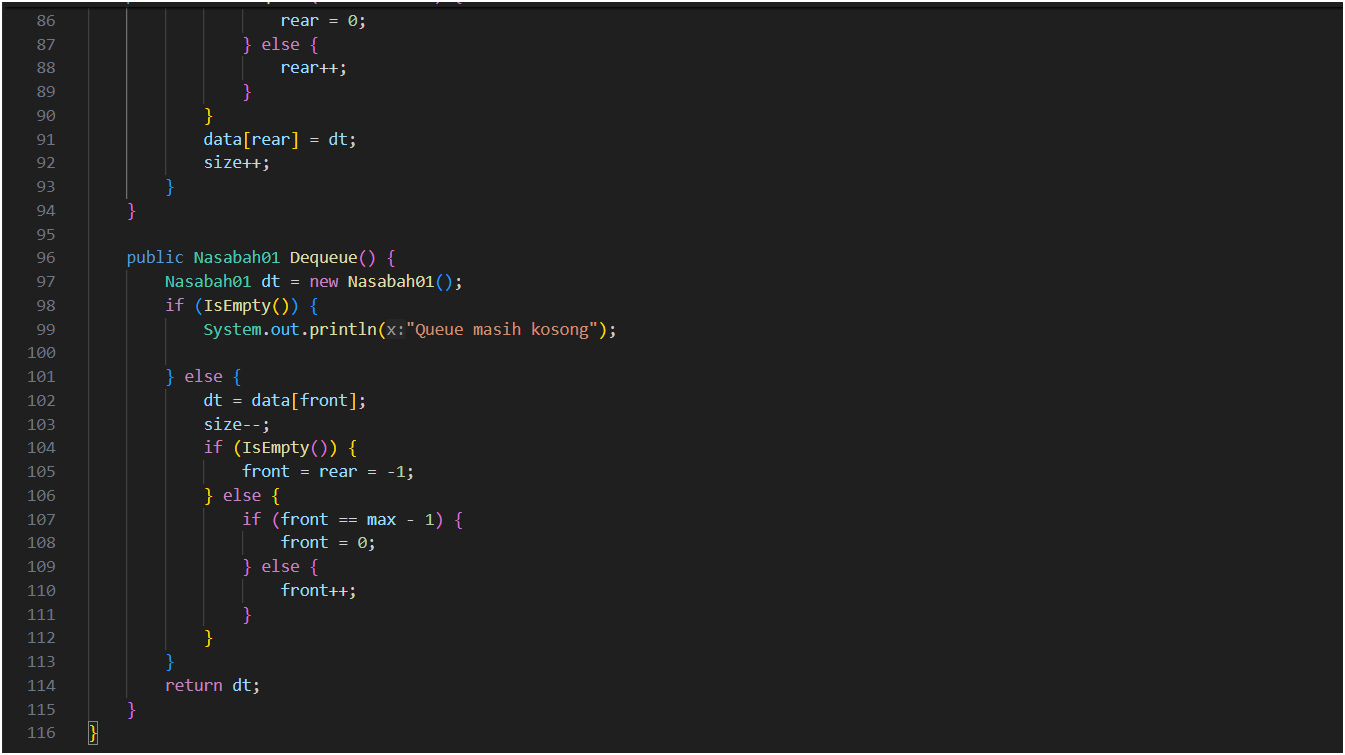
******

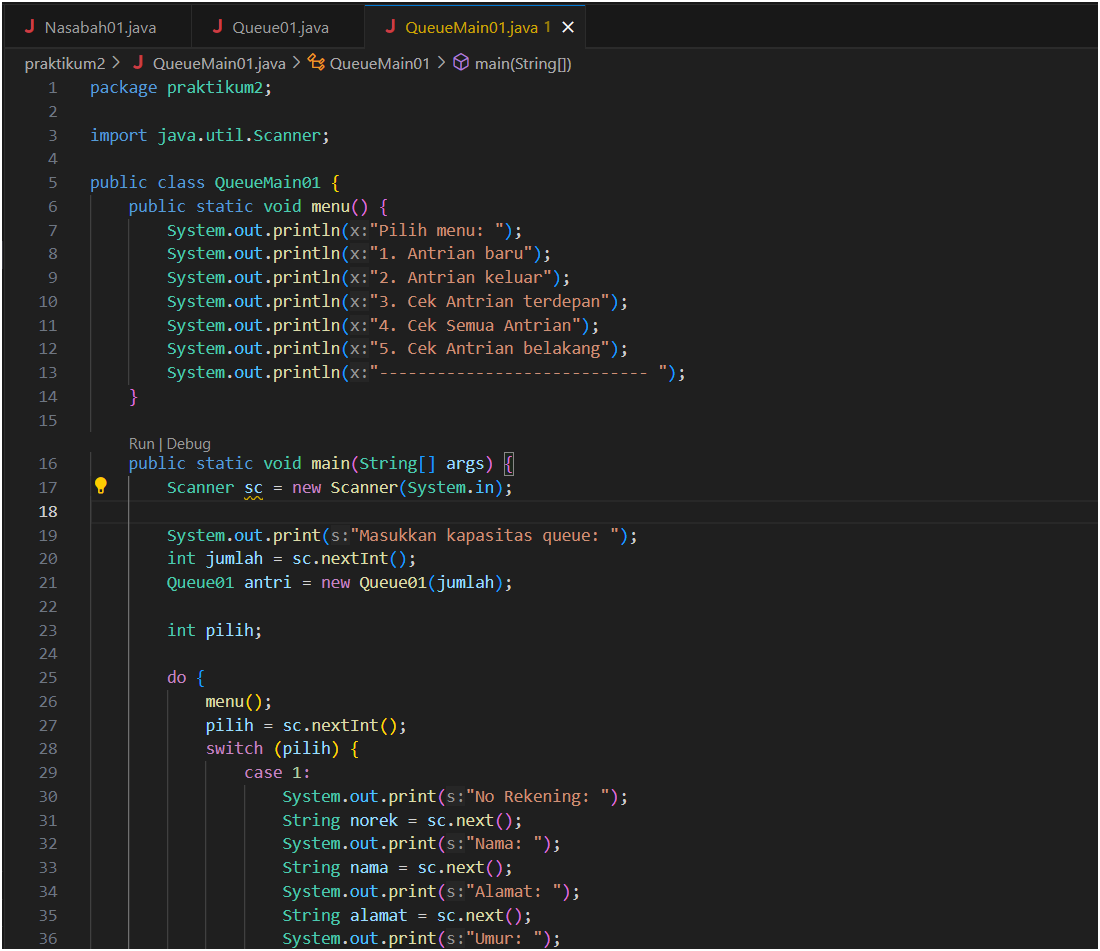
***Class Queue01***

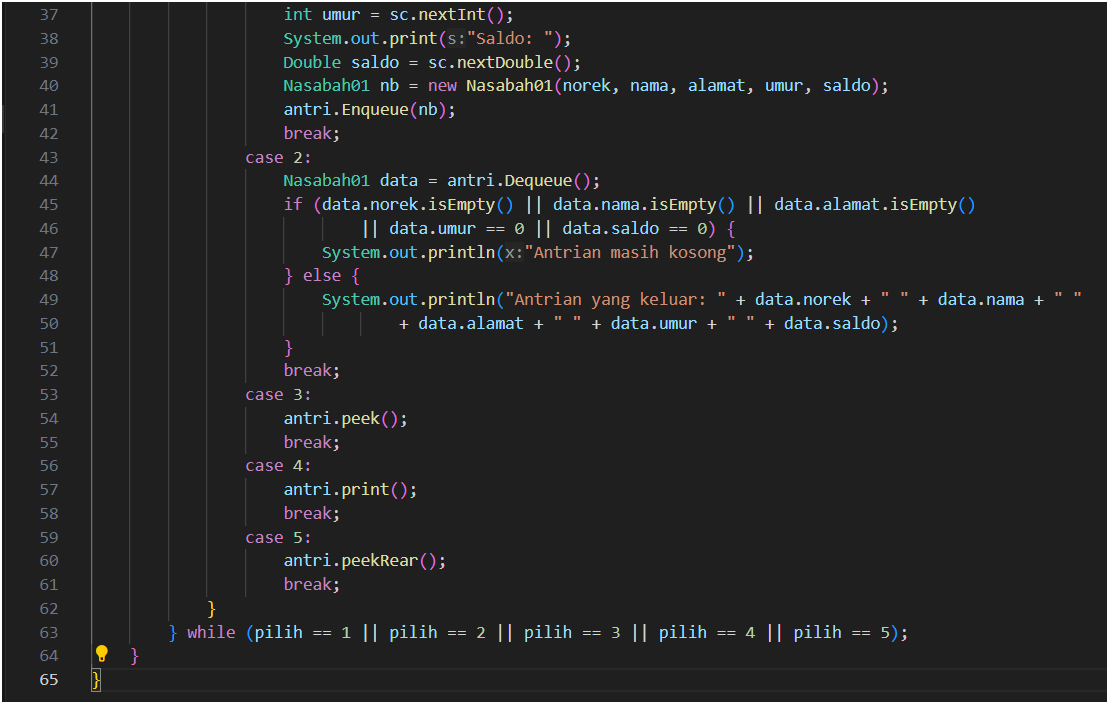
******



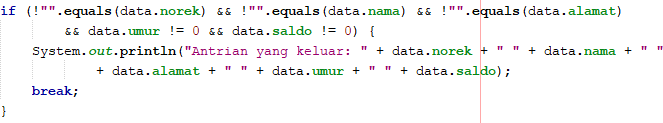




***Class QueueMain01***

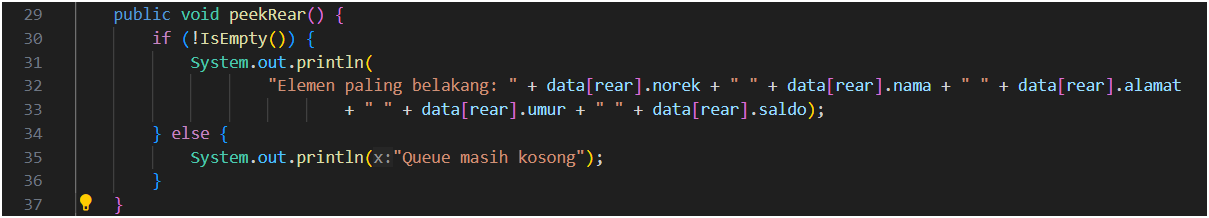


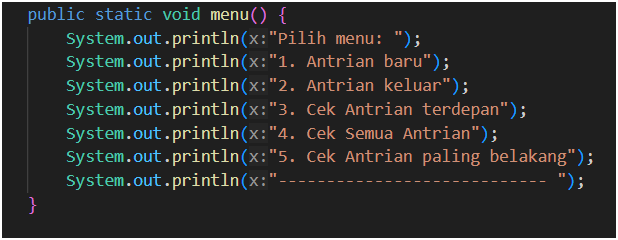
# Pertanyaan

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

* Kode tersebut berfungsi untuk memeriksa apakah data yang dikeluarkan dari antrian memiliki nilai yang valid sebelum mencetaknya.

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!

* 



# Tugas

1. 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:

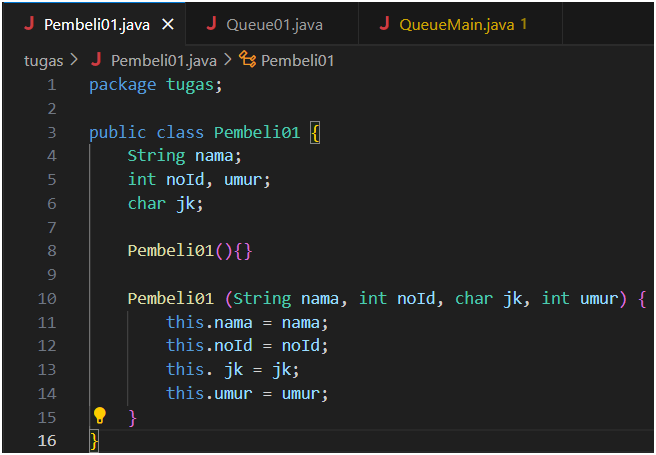
|  |
| --- |
| Pembeli |
| nama: String noID: int jenisKelamin: char  umur: int |
| Pasien (nama: String, noID: int, jenisKelamin: char, umur: int) |

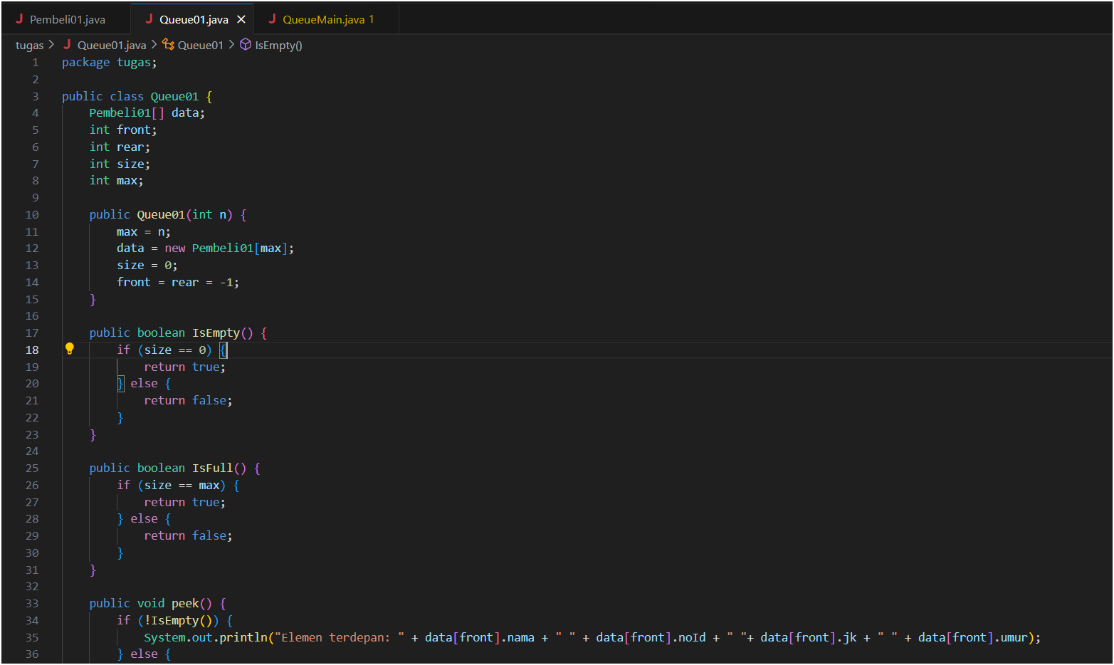
|  |
| --- |
| Queue |
| antrian: Pasien[] front: int  rear: int size: int max: int |
| Queue(n: int) isEmpty(): boolean isFull(): boolean  enqueue(antri: Pasien): void dequeue(): int  print(): void peek(): void peekRear(): void  peekPosition(nama: String): void daftarPasien(): void |

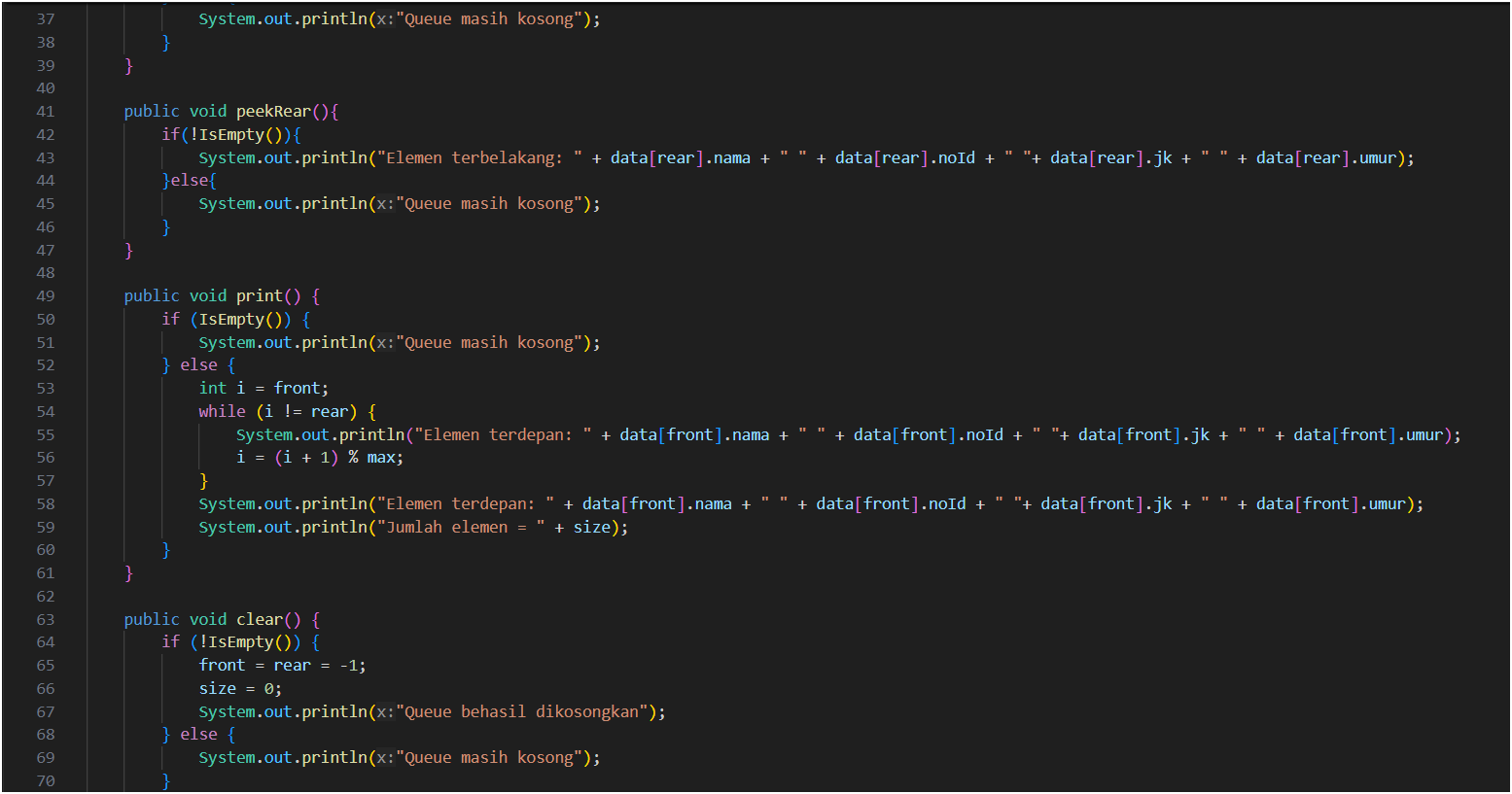
Class diagram Queue digambarkan sebagai berikut

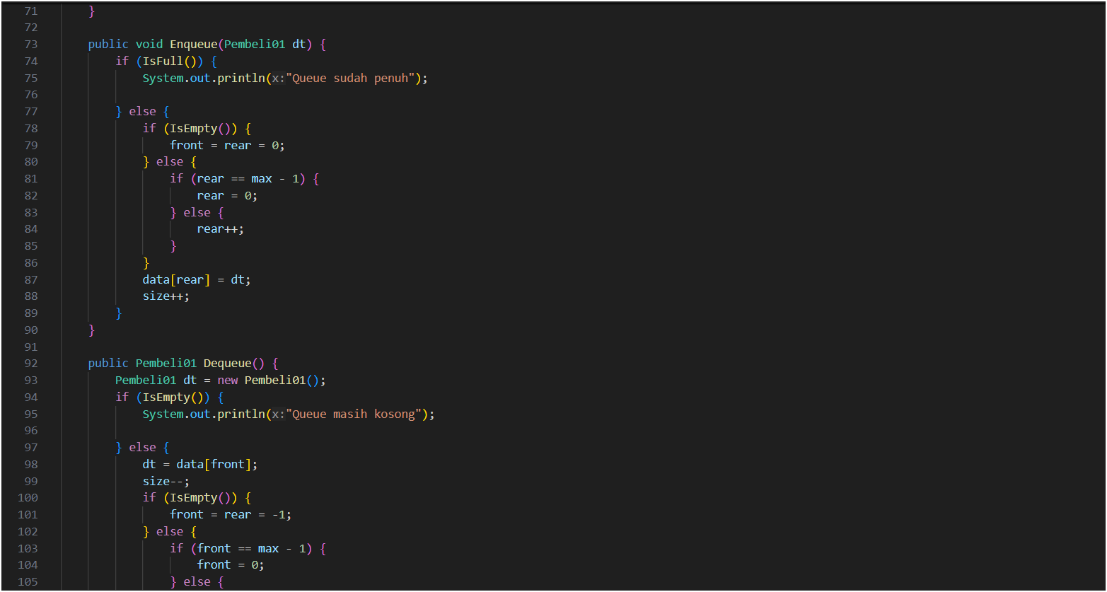
Keterangan method:

* 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

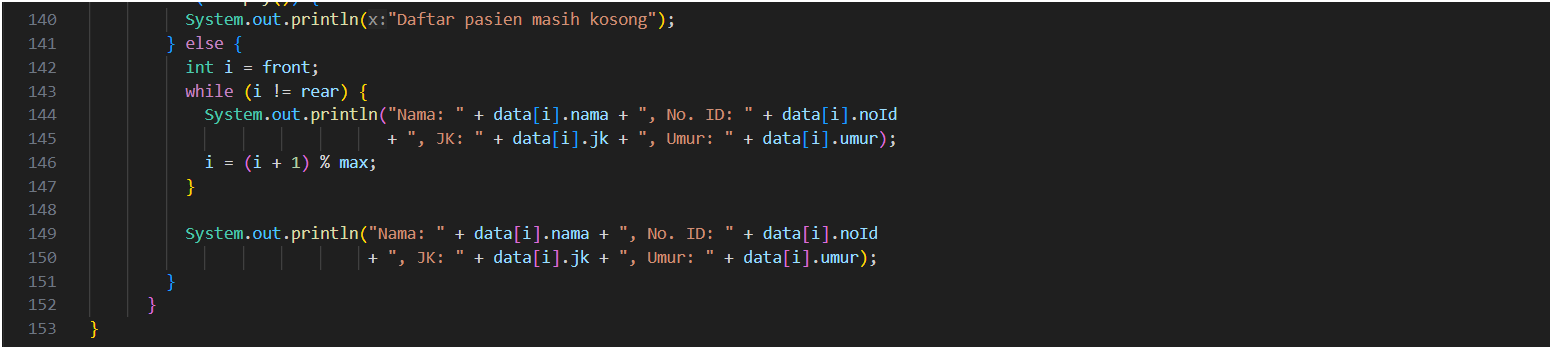
***Class Pembeli01***

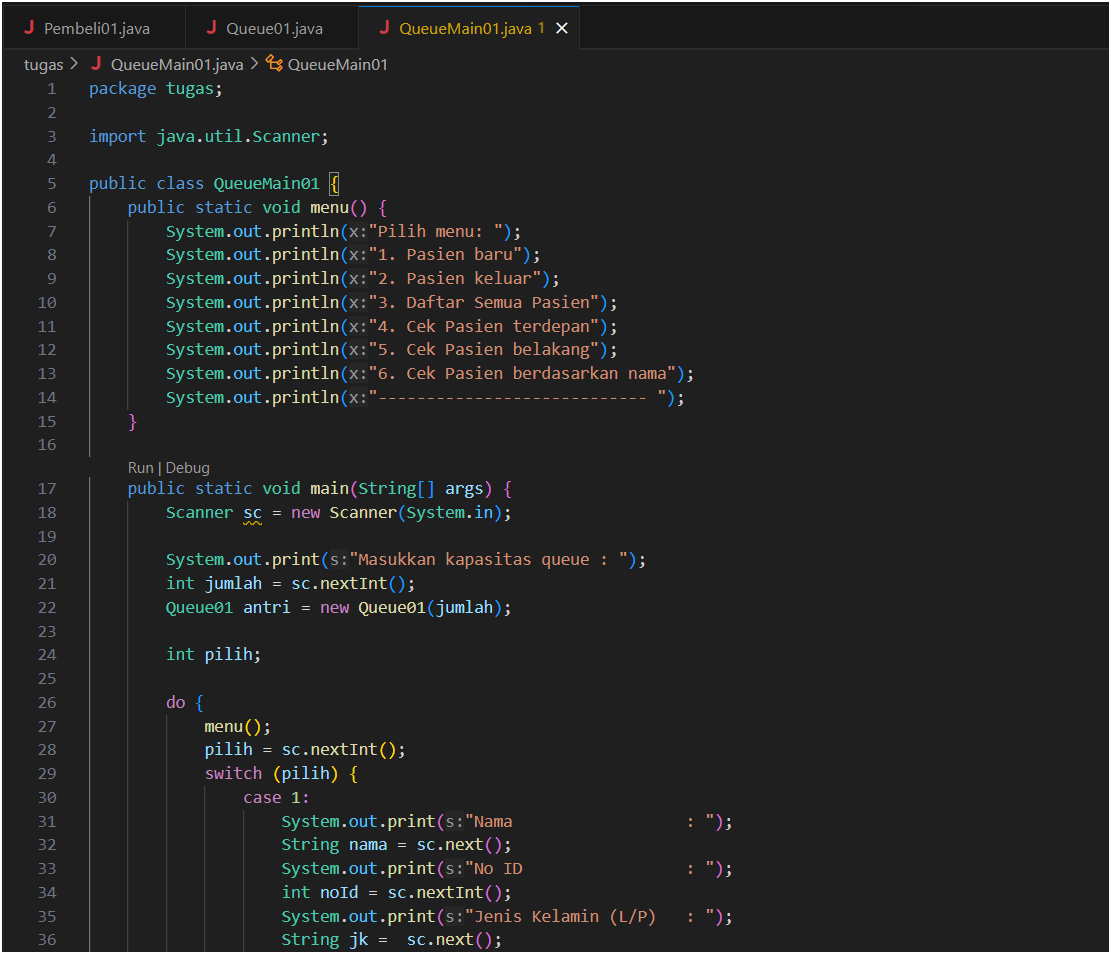
***Class Queue01***

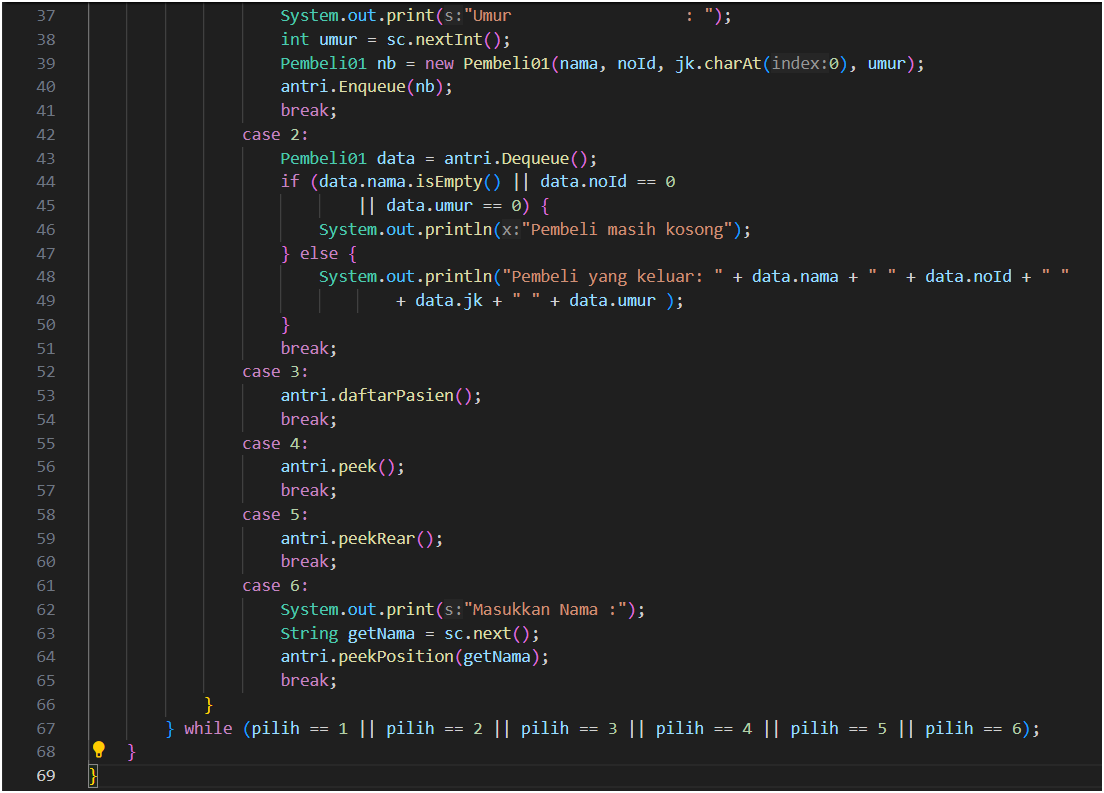


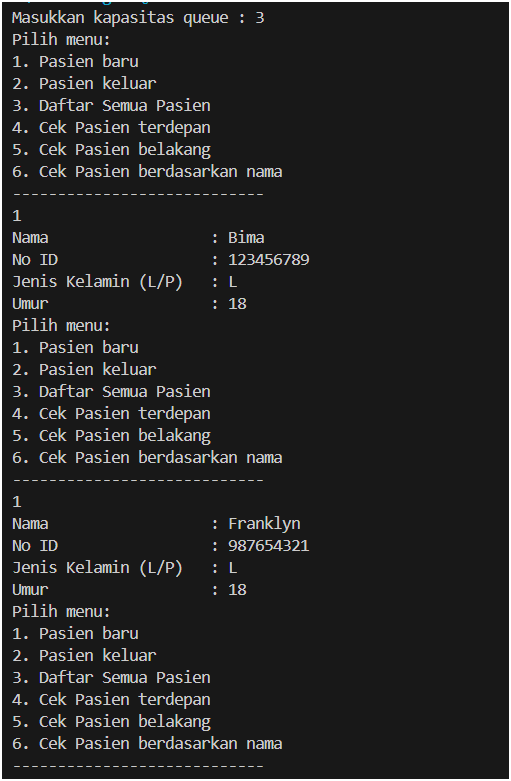
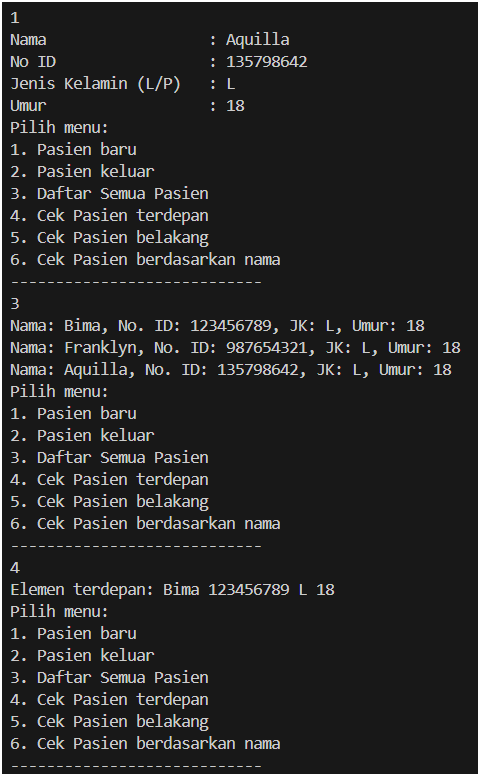






***Class MainQueue01***



***Output***

