**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA STRUKTUR DATA**

**QUEUE**



**Oleh:**

**FAUZIYYAH ADELIA RAMANDA**

**NIM. 2341760145**

**SIB-1F / 10**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

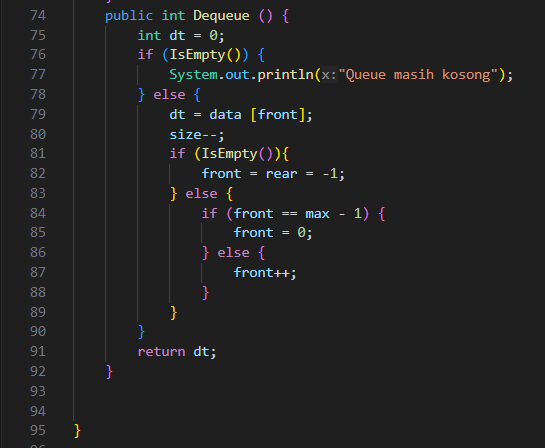
**8.2 Praktikum 1**

**8.2.1 Langkah-langkah Percobaan**

Queue10.java





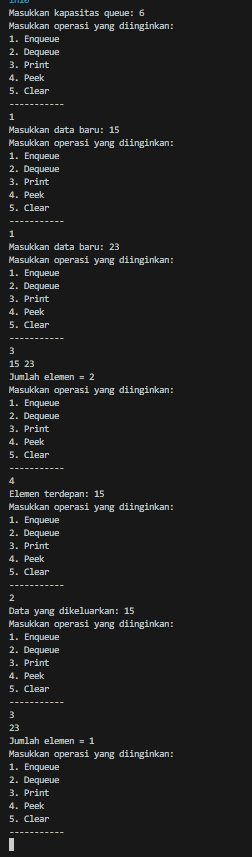


QueueMain10.java



**8.2.2 Verifikasi Hasil Percobaan**

Output :



**8.2.3 Pertanyaan**

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

Nilai awal -1 untuk front dan rear dan 0 untuk size merupakan praktik umum dalam implementasi antrian melingkar, hal ini membantu dalam menandakan antrian kosong, memudahkan perhitungan indeks, dan meniingkatkan efisiensi operasi antrian.

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



Potongan kode yang ditunjukkan berfungsi untuk memperbarui nilai rear setelah elemen baru ditambahkan ke antrian.

* Jika elemen baru ditambah ke elemen akhir, nilai rear diubah menjadi 0 untuk memulai Kembali dari elemen pertama
* Jika elemen baru ditambahkan ke indeks lain, nilai rear incremented untuk menunjuk ke elemen yang baru ditambahkan

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



* front == max -1 : potongan kode ini mengecek apakah posisi front saat ini berada pada indeks terakhir dari array data. Jika ini sudah terpenuhi, artinya elemen yang akan di dequeue adalah elemen terakhir dalam array data
* Jiika kondisi tersebut terpenuhi, maka front direset ke 0

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

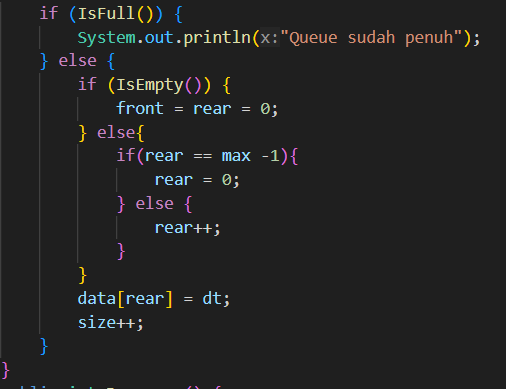
Variable I dimulai dari front karena disini ingin mencetak elemen-elemen dalam urutan yang benar sesuai dengan struktur queue. Dengan memulai dari front kita mencetak elemen dari front queue sehingga rear queue bisa sesuai dengan urutan mereka dalam array data circular

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



Pada method print, variable, I tidak memulai dari 0, melainkan dari front, karena kita ingin mencetak elemen queue yang sesungguhnya, yang mungkin dimulai dari indeks selain 0 dalam array circular

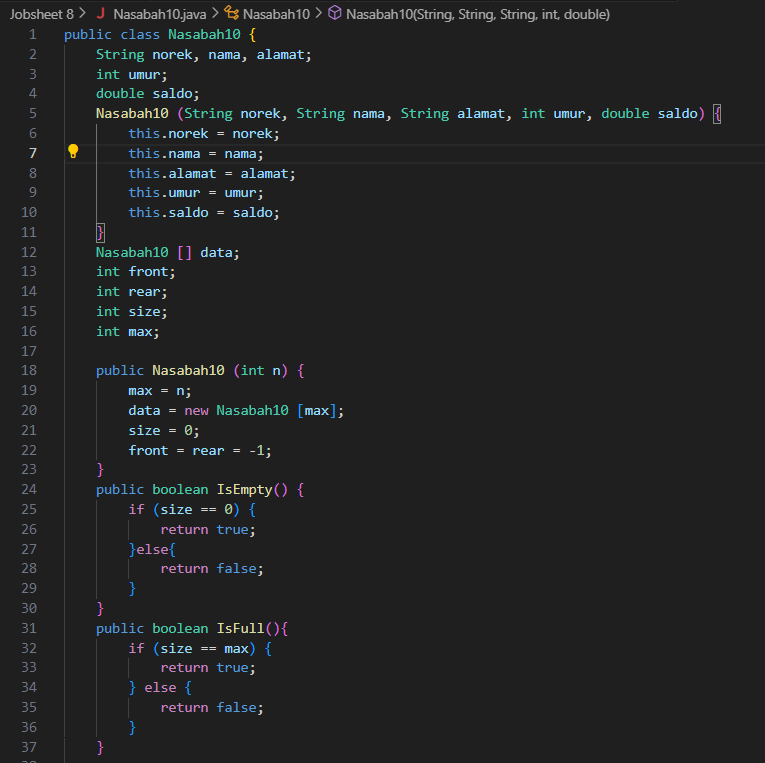
1. Tunjukkan potongan kode program yang merupakan queue overflow!

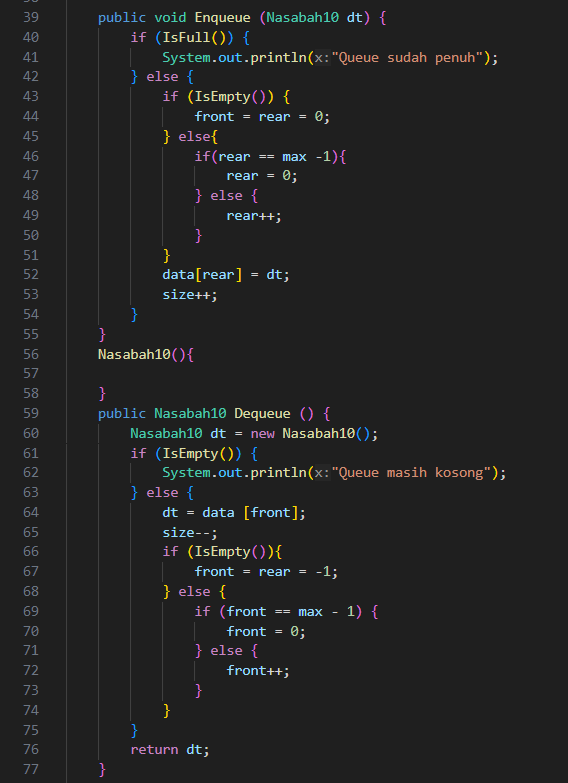


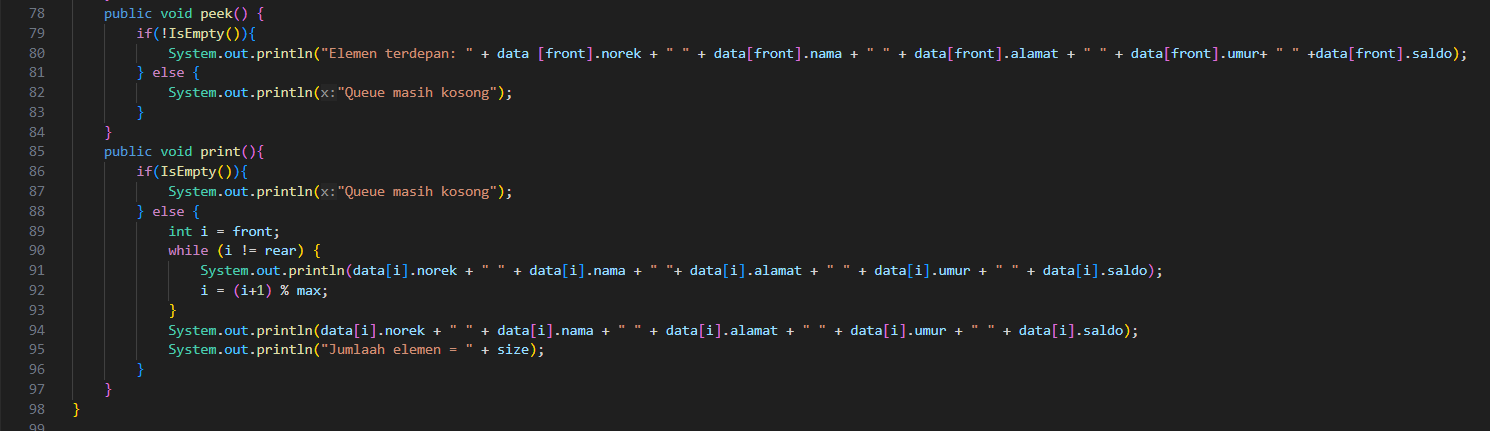
**8.3 Praktikum 2**

**8.3.1 Langkah-langkah Percobaan**

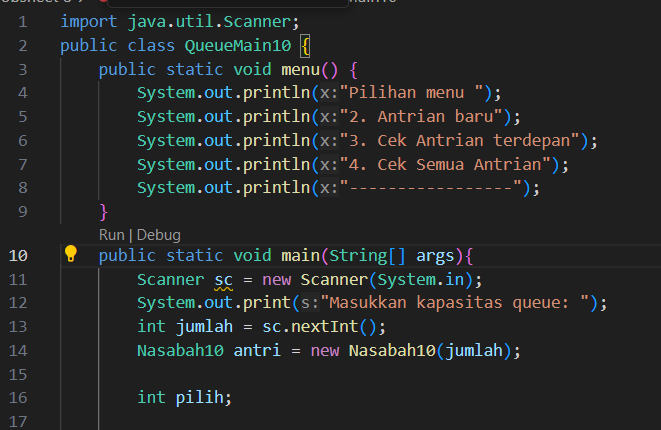
Nasabah10.java

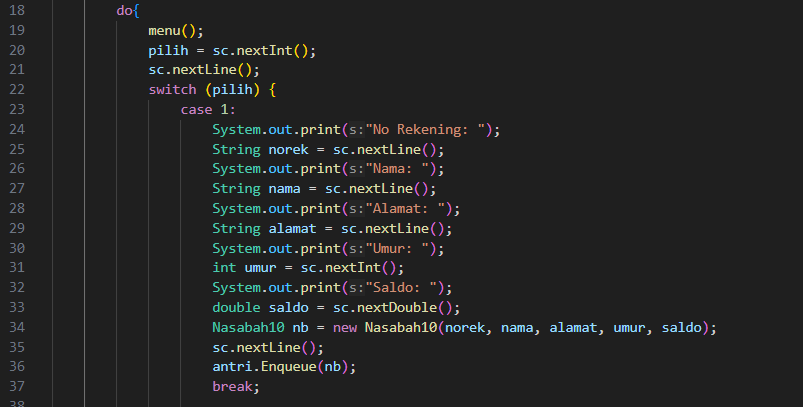


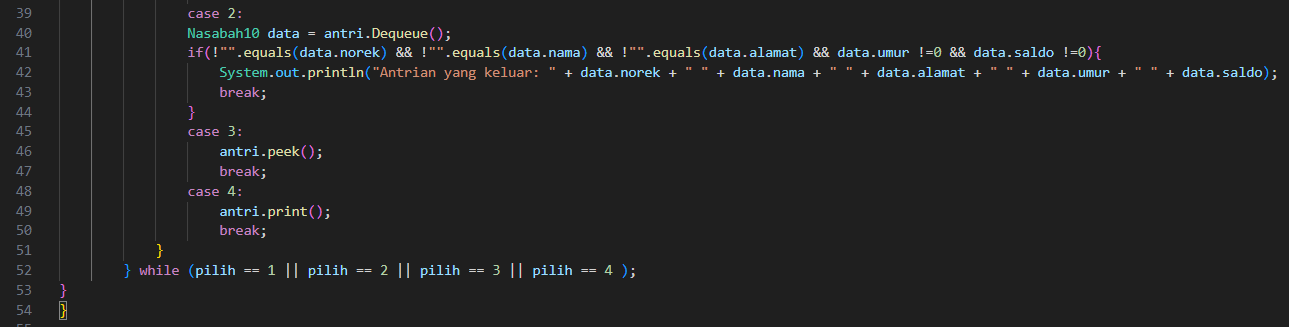




NasabahMain10.java

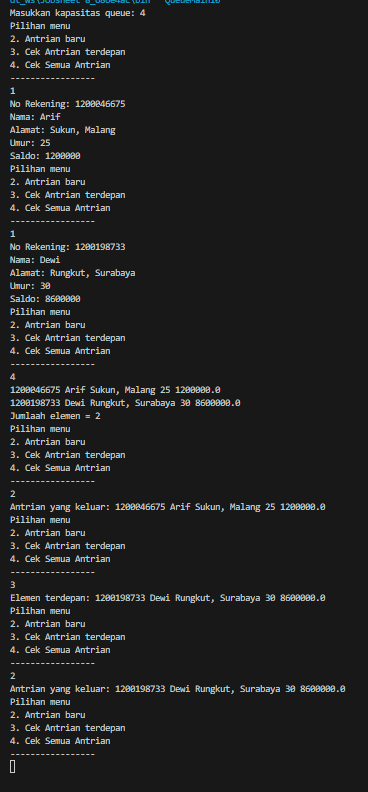




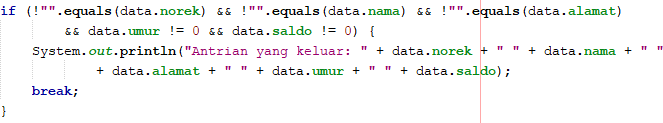


**8.3.2 Verifikasi Hasil Percobaan**

Output



**8.3.3 Pertanyaan**

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

Digunakan untuk mengecek apakah data yang di dequeue dari antrian memiliki nilai yang valid sebelum dicetak. Jika semua atribut data seperti nomor rek, nama, alamat, umur, dan saldo tidak kosong, maka data tersebut dianggap valid dan dicetak sebagai antrian yang keluar

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!

**8.4 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) |

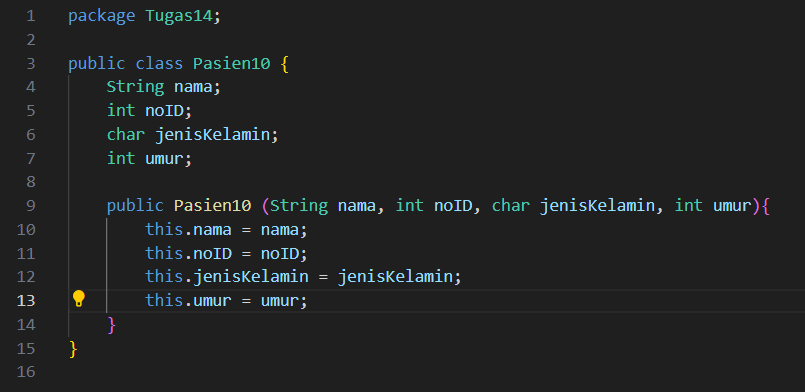
|  |
| --- |
| Queue |
| antrian: Pasien[] front: int  rear: int size: int max: int |
| Queue(n: int) isEmpty(): boolean isFull(): boolean  enqueue(antri: Pasien): void dequeue(): int  print(): void peek(): void peekRear(): void  peekPosition(nama: String): void daftarPasien(): void |

Class diagram Queue digambarkan sebagai berikut:

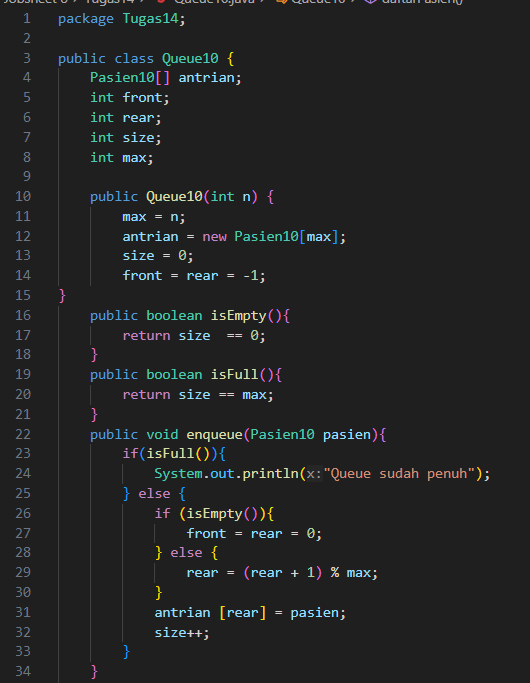
Keterangan method:

* 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

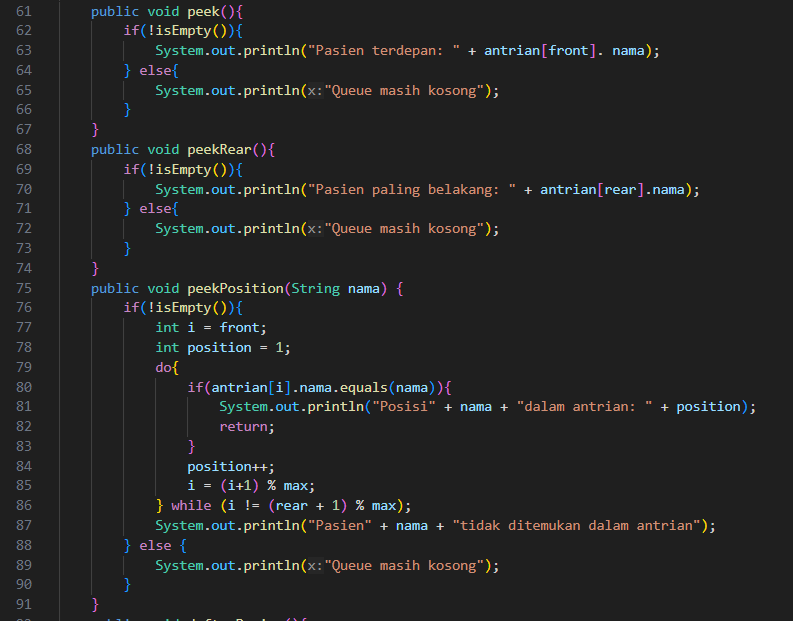
Pasien10.java



Queue10.java

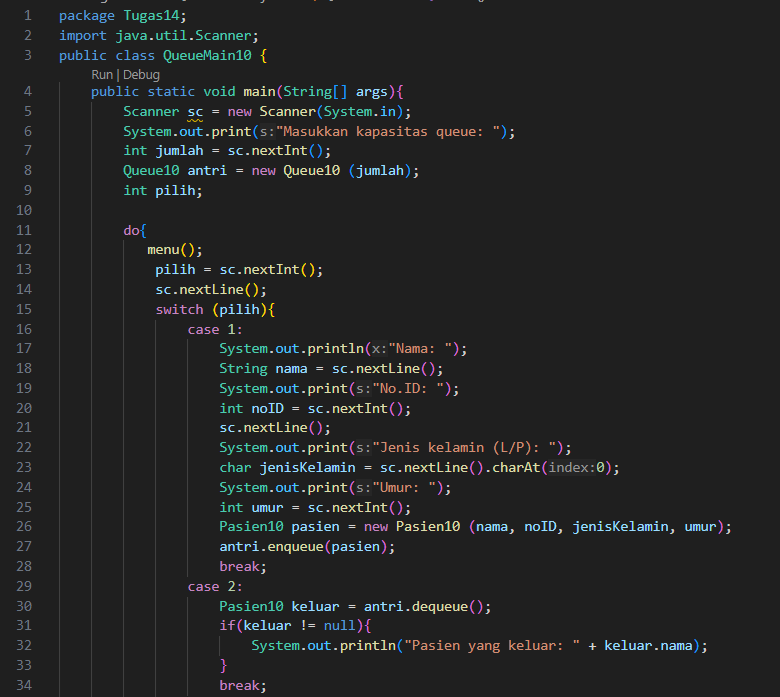


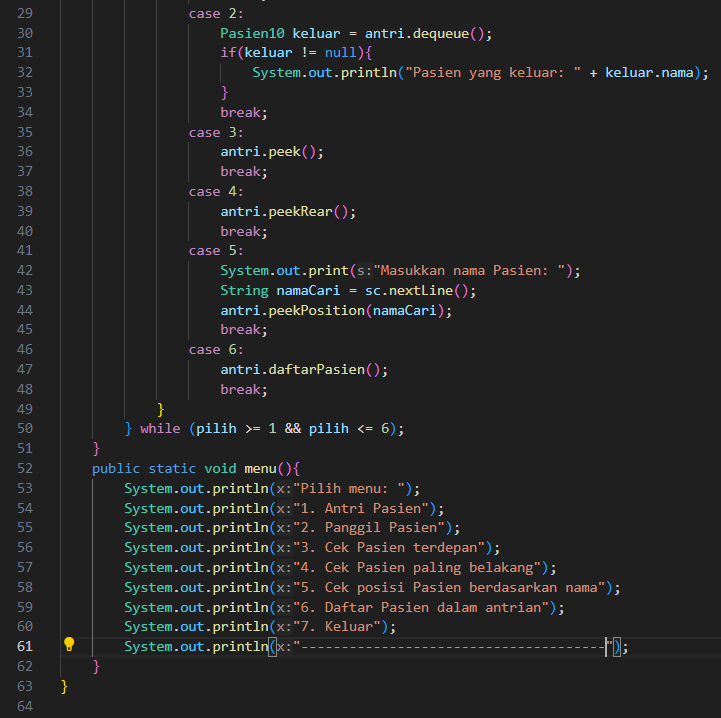






QueueMain10.java



****

Output :

