**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**



**Oleh:**

**DZULFIKAR MUHAMMAD AL GHIFARI**

**NIM. 2341760071**

**SIB-1F / 08**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

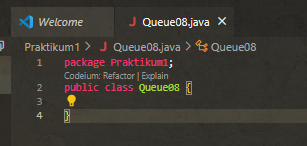
**POLITEKNIK NEGERI MALANG**

**PRAKTIKUM 8**

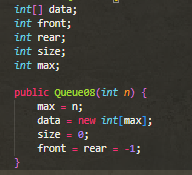
**8.2 Percobaan 1**

**8.2.1 Langkah Langkah percobaan**

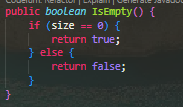
1. Membuat package dan class



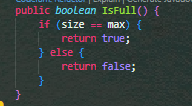
1. Deklarasi variable dan constructor



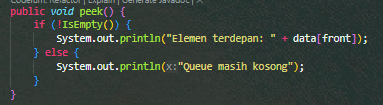
1. Menambahkan method isEmpty



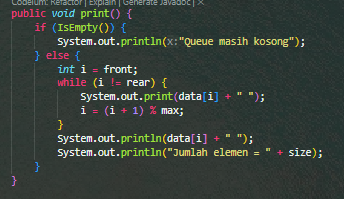
1. Menambahkan method isFull



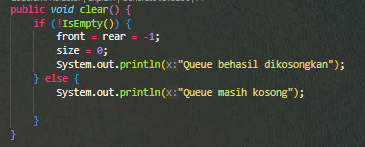
1. Menambahkan method peek



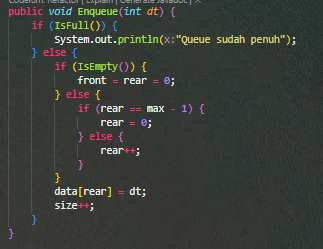
1. Menambahkan method print



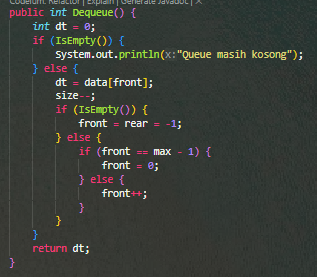
1. Menambahkan method clear



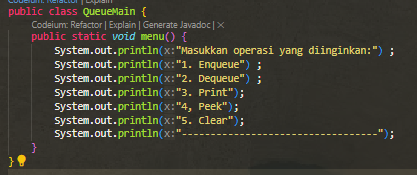
1. Menambahkan method Enqueue



9. Membuat method Dequeue



10. Membuat class QueueMain



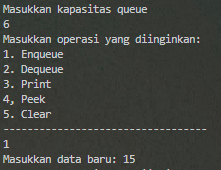
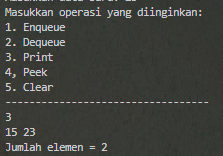
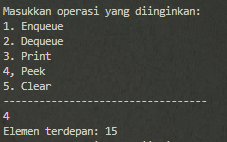
11. Membuat variabel n untuk menampung masukan

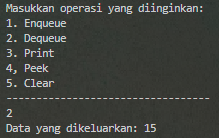
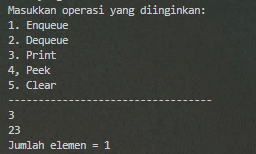


12. instansiasi object Q



**VERIFIKASI HASIL PERCOBAAN 8.2.2**

**PERTANYAAN 3.2.3**

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!



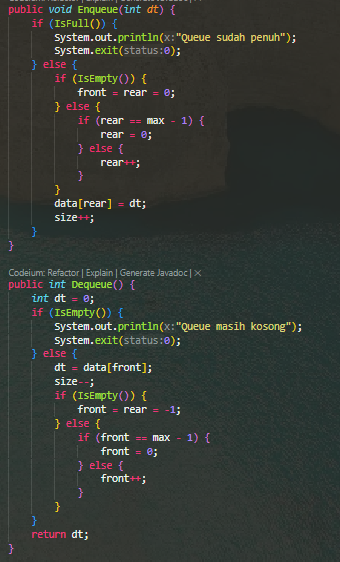
1. Tunjukkan potongan kode program yang merupakan queue overflow!
2. Pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan!

Jawaban

1. Atribut front dan rear bernilai -1 untuk menyatakan bahwa kondisi masih dalam keadaan kosong. Attribut size bernilai 0, karena indexs dimulai dari ke 0.
2. Jika rear berada pada posisi indexs terakhir, penambahan data baru akan di tempatkan pada index ke 0
3. Jika front berada pada posisi indexs terakhir, penambahan data baru akan di tempatkan pada index ke -0.
4. Karena front tidak selalu pada indeks ke-0, sedangkan perulangan dimulai dengan posisi front.
5. Jika nilai i tidak berposisi sebagai rear, maka akan dilakukan increment dan dimodulus dengan nilai max atau kapasitas dari Queue tersebut.
6. Potongan kode queue overflow



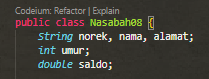
1. Dengan melakukan perubahan seprti berikut



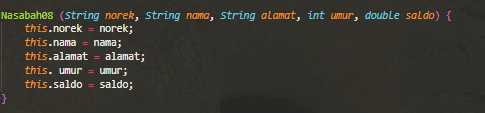
**8.3 Percobaan 2**

**8.3.1 Langkah-langkah Percobaan**

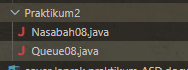
1. Menambahkan class nasabah dan instansiasi variabel



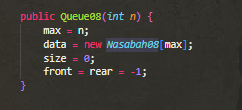
1. Menambahkan konstruktor



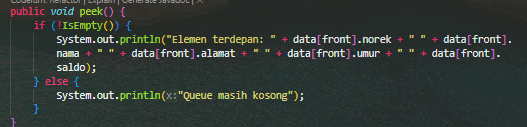
1. Menyalin kode program queue pada praktikum 1

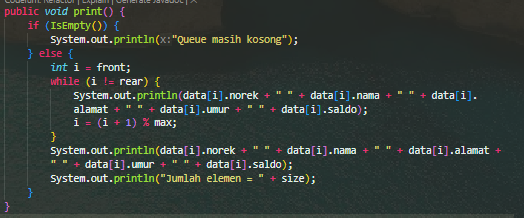


1. Melakukan perumahan pada queue menyesuaikan menggunakan object

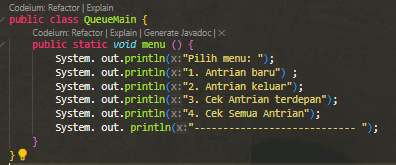
   

1. Mengubah method peek dan print





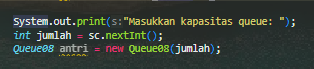
1. Menambahkan class queue main



1. Menambahkan class main



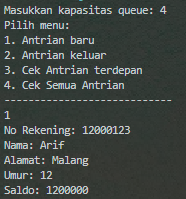
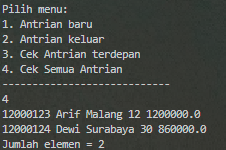
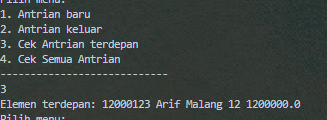
1. Menambahkan input dan instansiasi objek

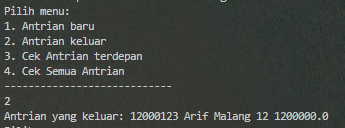
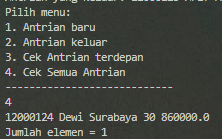


1. Menambahkan deklarasi variable pilih



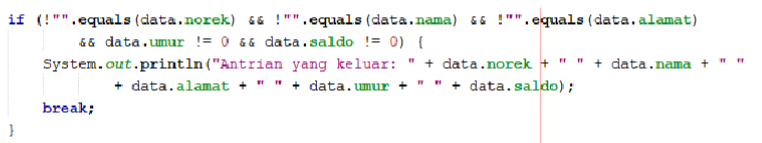
**VERIFIKASI HASIL PERCOBAAN 7.3.2**

**** **** 

**** ****

**PERTANYAAN 8.3.3**

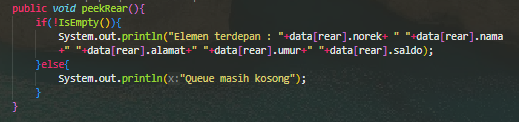
1. Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!

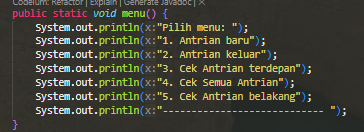


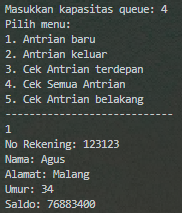
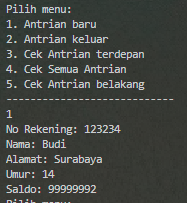
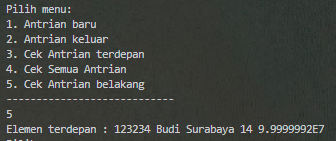
1. Lakukan modifikasi program dengan menambahkan method baru bernama peekRear pada class Queue yang digunakan untuk mengecek antrian yang berada di posisi belakang! Tambahkan pula daftar menu 5. Cek Antrian paling belakang pada class QueueMain sehingga method peekRear dapat dipanggil!

Jawaban

1. .equals adalah perintah untuk membandingkan dua string. Pada kode program tersebut di bandingkan apakah list data pada object data berisikan string kosong jika tidak berisi string kosong maka akan print antrian yang keluar. Jika tidak maka tidak akan menghasilkan output apapun

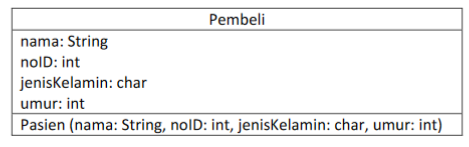


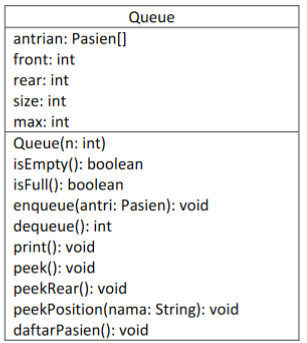
  

**8.4 TUGAS**

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:



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

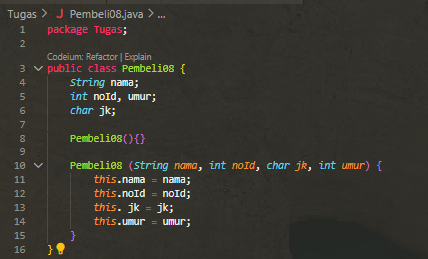
• Method peekRear(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling belakang

• Method peekPosition(): digunakan untuk menampilkan seorang pasien (berdasarkan nama) posisi antrian ke berapa

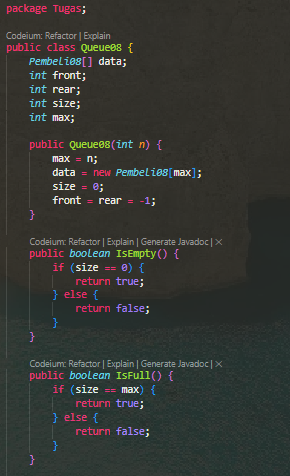
• Method daftarPasien(): digunakan untuk menampilkan data seluruh pasien

Jawaban

Membuat class pembeli sebagai berikut



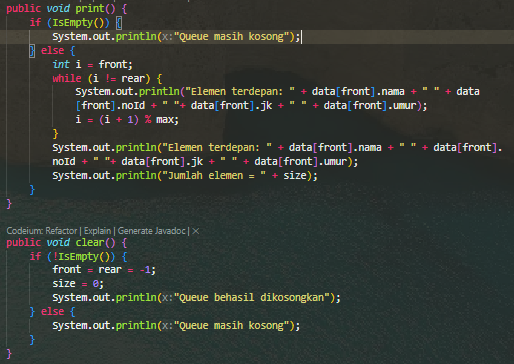
Membuat class queue sebagai berikut



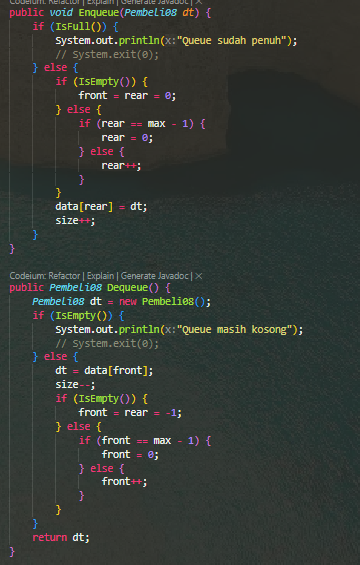
Method peek dan peek rear



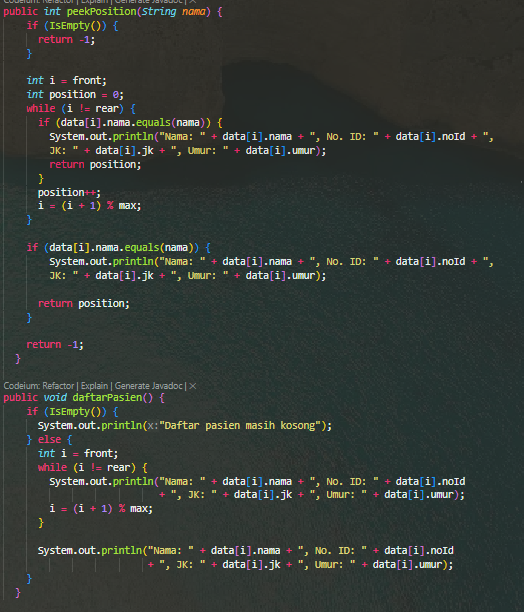
Method print & clear



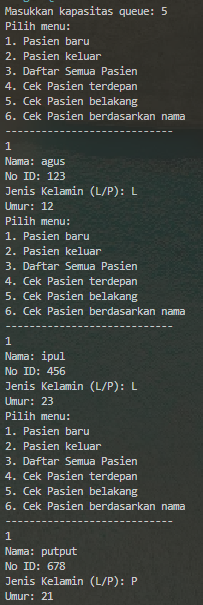
Method Enqueue & Dequeue

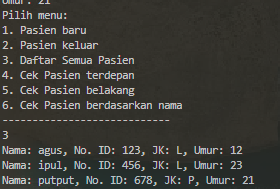


Method peekPosition & daftarPasien



Hasil compile

****

****

