Nama : Angga Saputra Ramadhan

No.Absen : 05

Kelas : 1B

NIM : 244107020065

JOBSHEET X

QUEUE

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengenal struktur data Queue

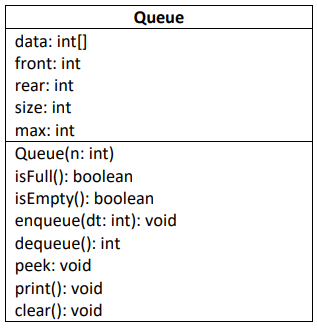
2. Membuat dan mendeklarasikan struktur data Queue

3. Menerapkan algoritma Queue dengan menggunakan array

1. Praktikum 1
   1. Percobaan 1: Operasi Dasar Queue

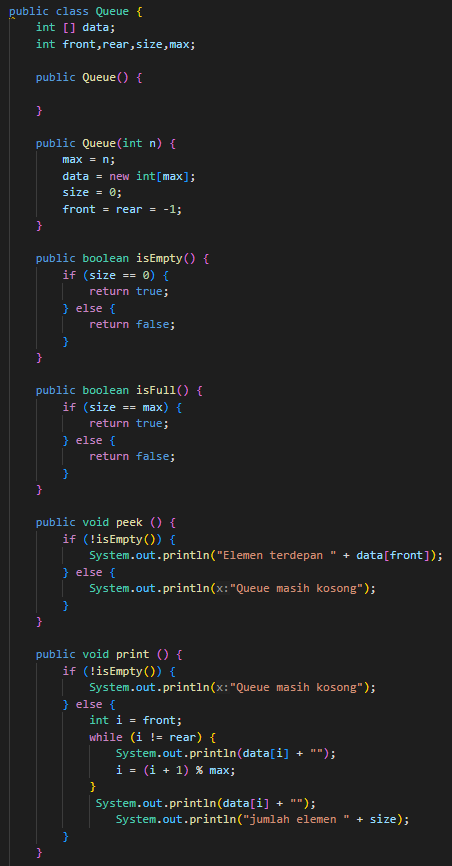
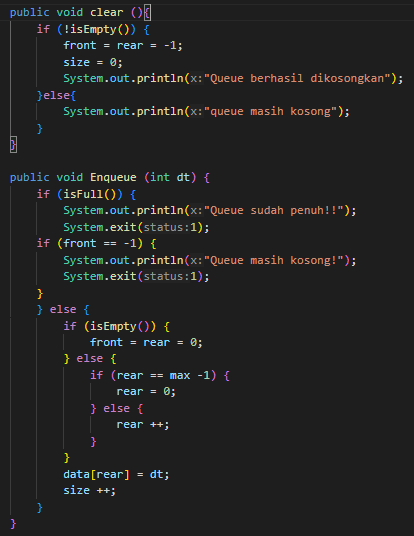
Waktu Percobaan : 90 Menit

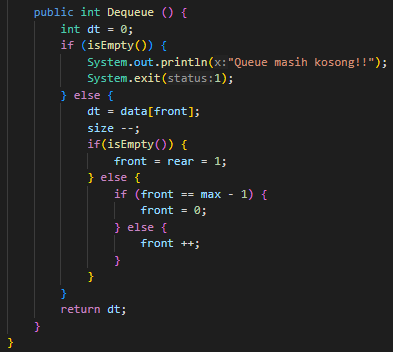
Pada percobaan ini, kita akan menerapkan operasi dasar pada algoritma Queue. Perhatikan diagram Class Queue berikut ini:



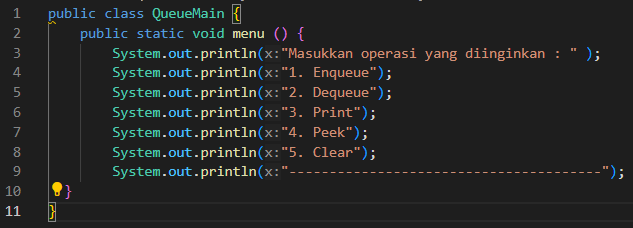
2.2.1 Langkah-Langkah Percobaan

a. Class Queue

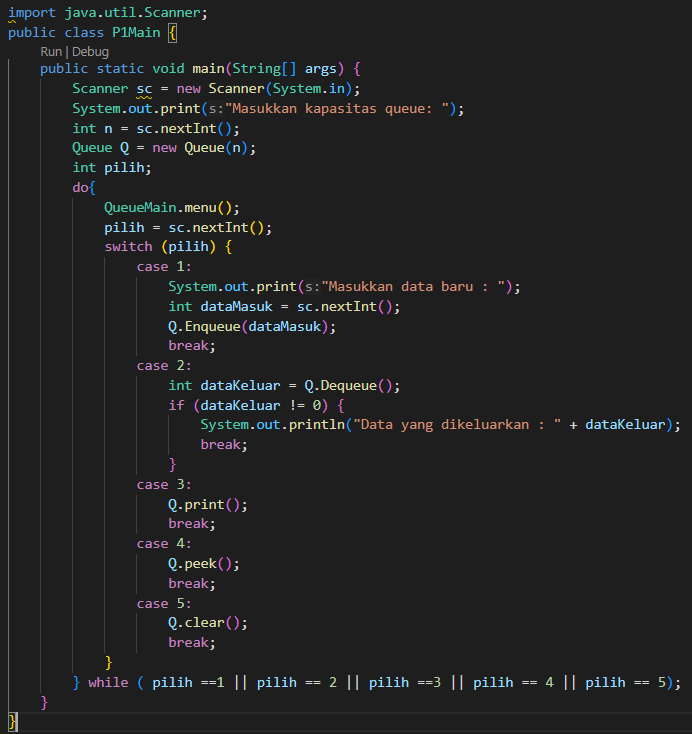
 



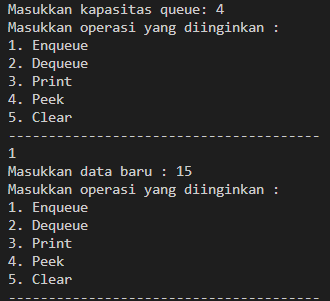
b. Class QueueMain.java



c. Class P1Main.java



d. Verifikasi Hasil Percobaan



e. Pertanyaan

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

bernilai 0?

Jwb:

- front = -1 dan rear = -1 menandakan antrian kosong karena belum ada alemen masuk

- size = 0 menyatakan bahwa belum ada data yang ditambahkan

- Nilai -1 berguna untuk **inisialisasi awal** agar saat pertama kali enqueue, posisi bisa

ditentukan dengan benar.

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

Jwb:

* rear = (rear + 1) % max: perhitungan indeks **sirkuler**, agar antrian bisa berputar kembali ke indeks 0 setelah mencapai akhir array.
* data[rear] = mhs: menambahkan data ke posisi rear.
* size++: menambah jumlah elemen dalam antrian.

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

Jwb:

* Mengambil data mahasiswa dari indeks front.
* front = (front + 1) % max: memindahkan posisi front secara sirkuler ke elemen berikutnya.
* size--: mengurangi jumlah elemen karena satu telah diambil dari antrian.

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0),

melainkan int i=front?

Jwb:

- Karena data dimulai dari front (bukan selalu dari indeks 0).

- Jika front = 2, berarti elemen pertama di antrian ada di data[2], bukan data[0].

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

Jwb:

- Ini untuk mendapatkan indeks data **secara melingkar (circular)**.

- Jika front + i lebih besar dari panjang array (max), maka kembali ke awal (0), agar array bisa

digunakan penuh seperti queue lingkaran.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Jwb:

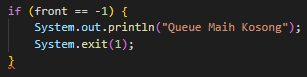
Ini menandakan bahwa kapasitas array sudah penuh (size == max), dan **tidak bisa enqueue lagi**.

7. 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!

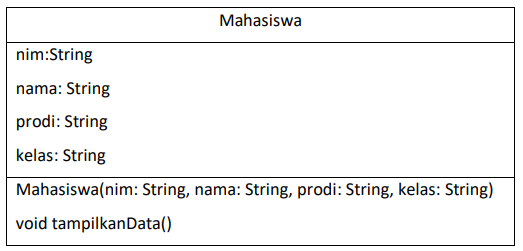
Jwb:



* 1. Percobaan 2: Antrian Layanan Akademik

Waktu Percobaan : 90 Menit

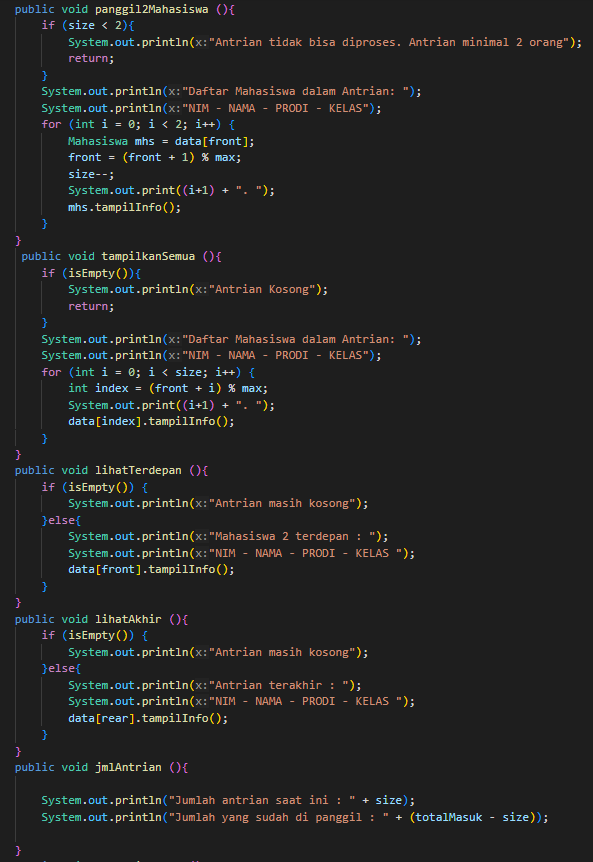
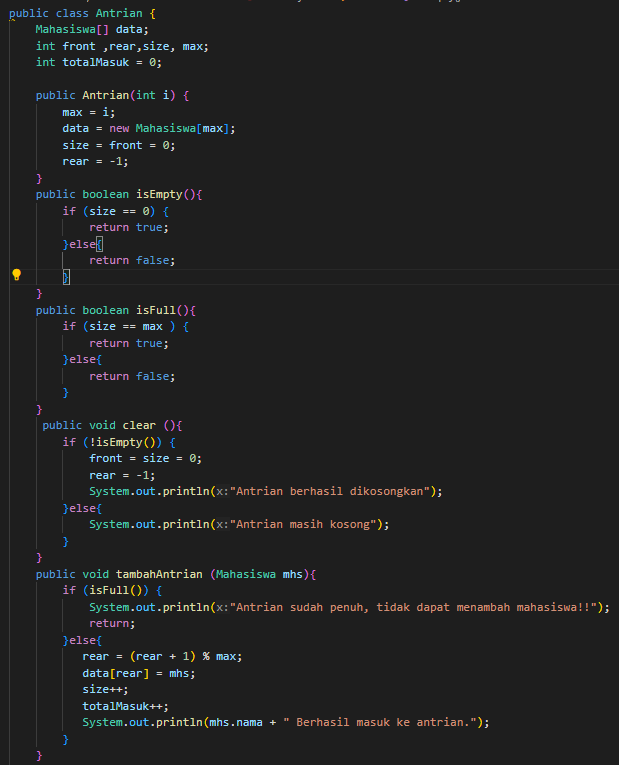
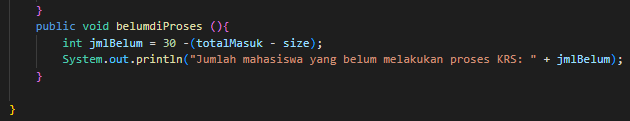
Pada Percobaan ini, kita akan membuat program yang mengilustrasikan layanan pada admin akademik. Perhatikan Diagram Class Berikut Ini:



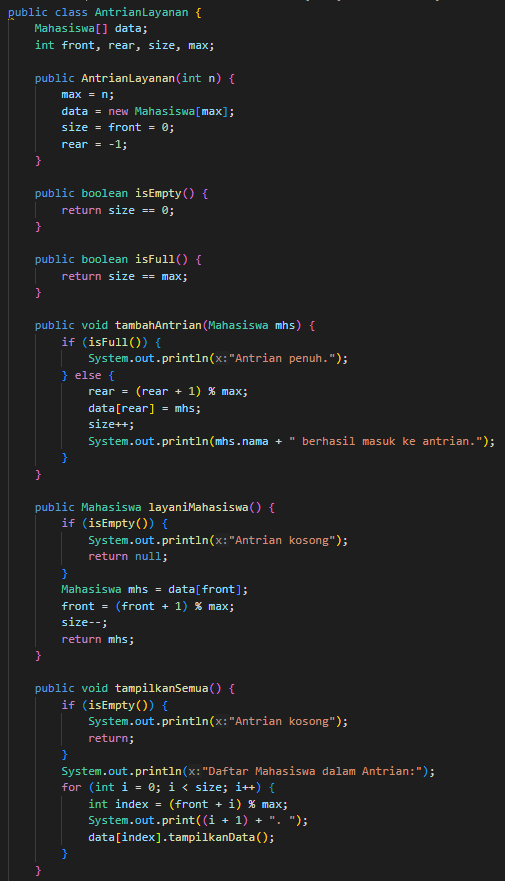
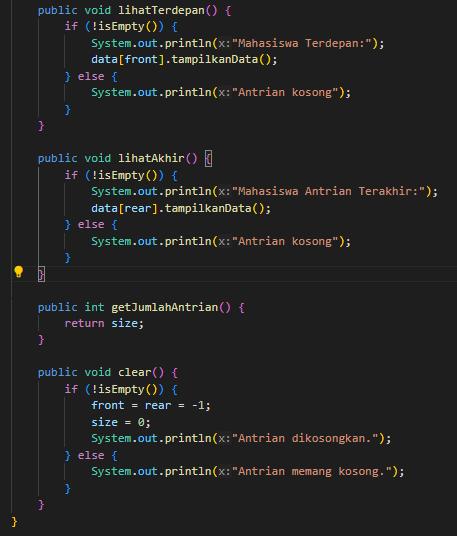
* + 1. Langkah-Langkah Percobaan

Berdasarkan diagram class tersebut, akan dibuat program class Mahasiswa dalam java.

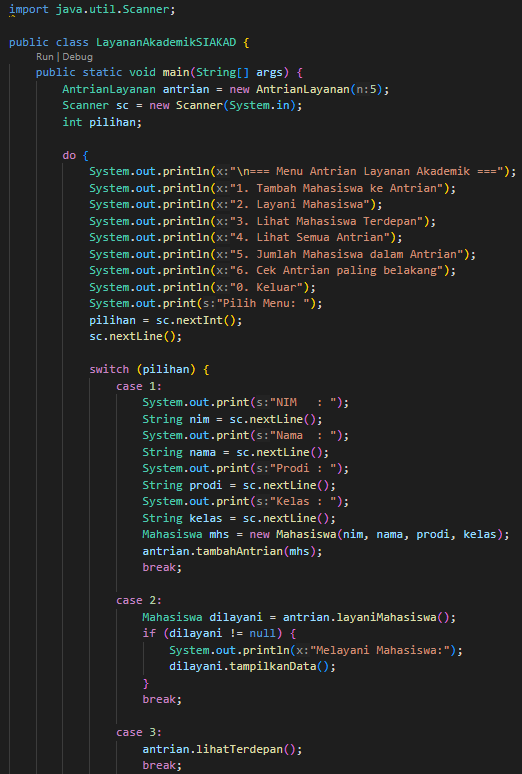
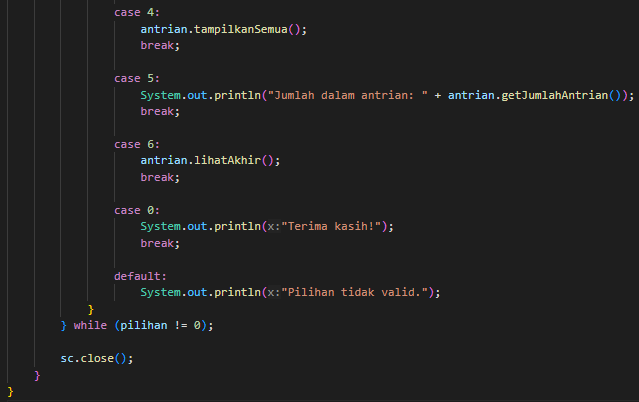
1. Class Antrian.java

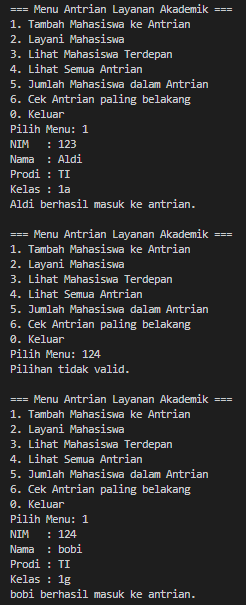
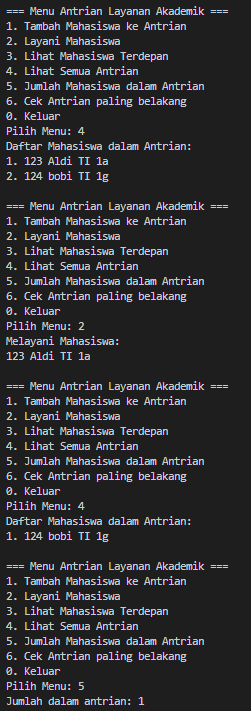
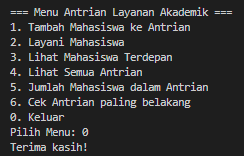
1. Class AntrianLayanan.java

1. Class LayananAkademikSIAKAD.java

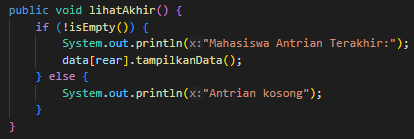
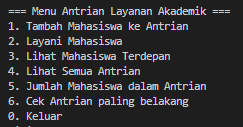
 

1. Verifikasi Hasil Percobaan

* + 1. Pertanyaan

Lakukan modifikasi program dengan menambahkan method baru bernama LihatAkhir pada class AntrianLayanan yang digunakan untuk mengecek antrian yang berada di posisi belakang. Tambahkan pula daftar menu 6. Cek Antrian paling belakang pada class LayananAkademikSIAKAD sehingga method LihatAkhir dapat dipanggil!

* 1. Tugas

Waktu : 120 Menit

Buatlah program antrian untuk mengilustasikan antrian persetujuan Kartu Rencana Studi (KRS) Mahasiswa oleh Dosen Pembina Akademik (DPA). Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya (data mahasiswa seperti pada praktikum 2). Gunakan class untuk antrian seperti pada Praktikum 1 dan 2, dengan method-method yang berfungsi :

* Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
* Menambahkan antrian, Memanggil antrian untuk proses KRS – setiap 1x panggilan terdiri dari 2 mahasiswa (pada antrian no 1 dan 2)
* Menampilkan semua antrian, Menampilkan 2 antrian terdepan, Menampilkan antrian paling akhir.
* Cetak jumlah antrian, Cetak jumlah yang sudah melakukan proses KRS
* Jumlah antrian maximal 10, jumlah yang ditangani masing-masing DPA 30 mahasiswa, cetak jumlah mahasiswa yang belum melakukan proses KRS.

Diagram Class untuk antriannya. Implementasikan semua method menggunakan menu pilihan pada fungsi main.

