**JOBSHEET X QUEUE**

**OPERASI DASAR ENQUEUE DAN DEQUEUE**

**Praktikum Algoritma Struktur Data**



**Krisnahadi Jayawardana**

**NIM. 244107060001**

**Kelas 1C**

**Jurusan Teknologi Informasi**

**Sistem Informasi Bisnis**

**POLITEKNIK NEGERI MALANG**

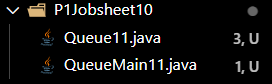
**TAHUN 2025**

**2.1 Percobaan 1 : Operasi Dasar Queue**

**Link GitHub :** [**https://github.com/KrisnahadiJ/Praktikum-ASD.git**](https://github.com/KrisnahadiJ/Praktikum-ASD.git)

**2.1.1. Langkah-langkah Percobaan**

1. **Buat folder baru bernama P1Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Queue.**

****

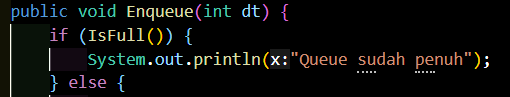
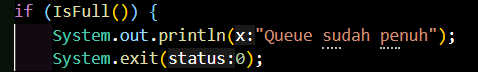
1. **Code Queue11.java:**

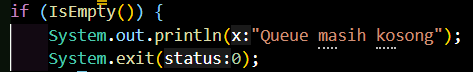
|  |  |
| --- | --- |
|  |  |

1. **Code QueueMain11.java:  
   **
2. **Output percobaan 1 :**

|  |  |
| --- | --- |
|  |  |
|  |  |

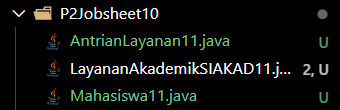
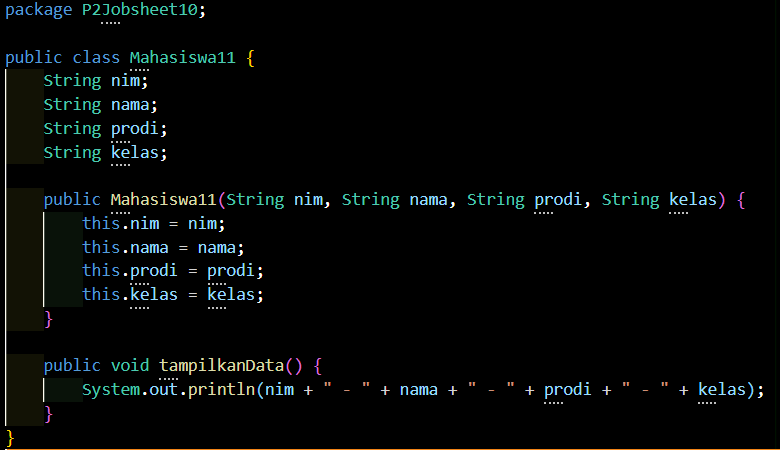
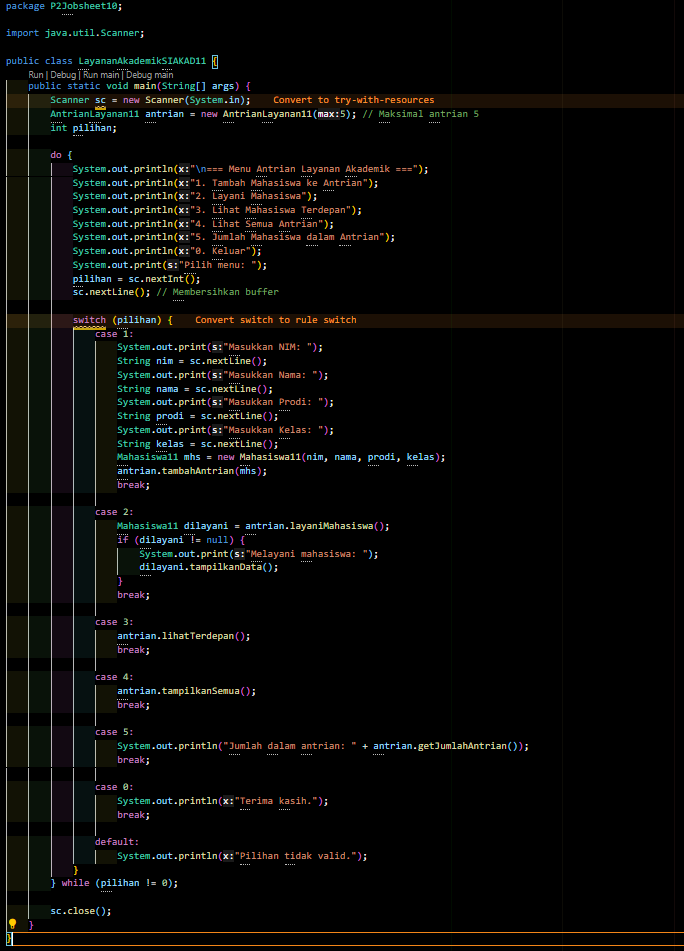
**2.1.3. Pertanyaan**

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?   
   Nilai -1 menandakan queue kosong (tidak ada elemen), sedangkan size bernilai 0 untuk menunjukkan bahwa tidak ada elemen dalam queue.
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!   
   ****  
   Untuk membuat queue melingkar (circular queue), sehingga rear kembali ke indeks 0 jika sudah mencapai batas akhir array.
3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!   
   ****  
   Untuk membuat queue melingkar, sehingga front kembali ke indeks 0 jika sudah mencapai batas akhir array.
4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?  
   Karena elemen pertama yang harus dicetak adalah elemen di posisi front, bukan dari indeks 0.
5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!   
   ****  
   Untuk melanjutkan iterasi ke indeks berikutnya dalam array melingkar, kembali ke indeks 0 jika sudah mencapai batas akhir array.
6. Tunjukkan potongan kode program yang merupakan queue overflow!   
   ****
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!  
   Menambahkan [System.exit(0)](vscode-file://vscode-app/c:/Users/Krisnahadi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) setelah menampilkan pesan overflow/underflow:  
   ****

****

**2.2. Percobaan 2 : Antrian Layanan Akademik**

**2.2.1. Langkah-langkah Percobaan**

1. **Buat folder baru bernama P2Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Mahasiswa.  
   **
2. **Code Mahasiswa11.java:  
   **
3. **Code AntrianLayanan11.java:  
   **
4. **Code LayananAkademikSIAKAD.java:  
   **
5. **Output Percobaan 2:**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

2.2.3 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!

|  |  |
| --- | --- |
| Menambahkan method Lihat Akhir | Menambahkan menu 6 |
|  |  |
| Output: |  |

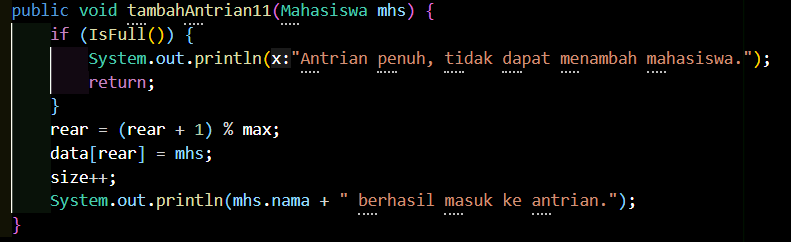
2.3 Tugas

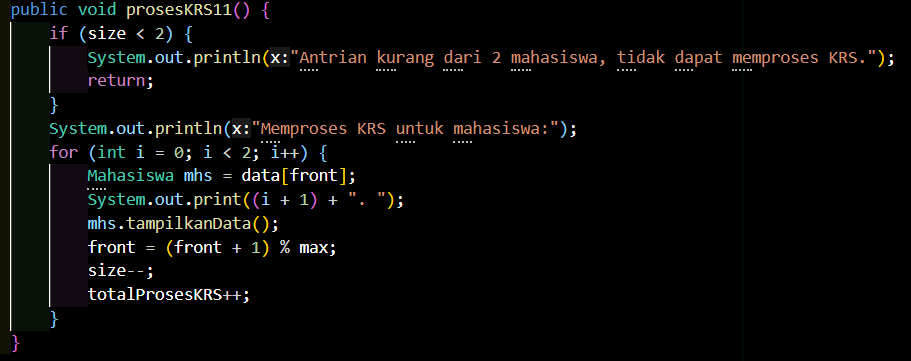
Buatlah program antrian untuk mengilustasikan antrian persetujuan Kartu Rencana Studi (KRS) Mahasiswa oleh Dosen Pembina Akademik (DPA). Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya (data mahasiswa seperti pada praktikum 2). Gunakan class untuk antrian seperti pada Praktikum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

• Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.

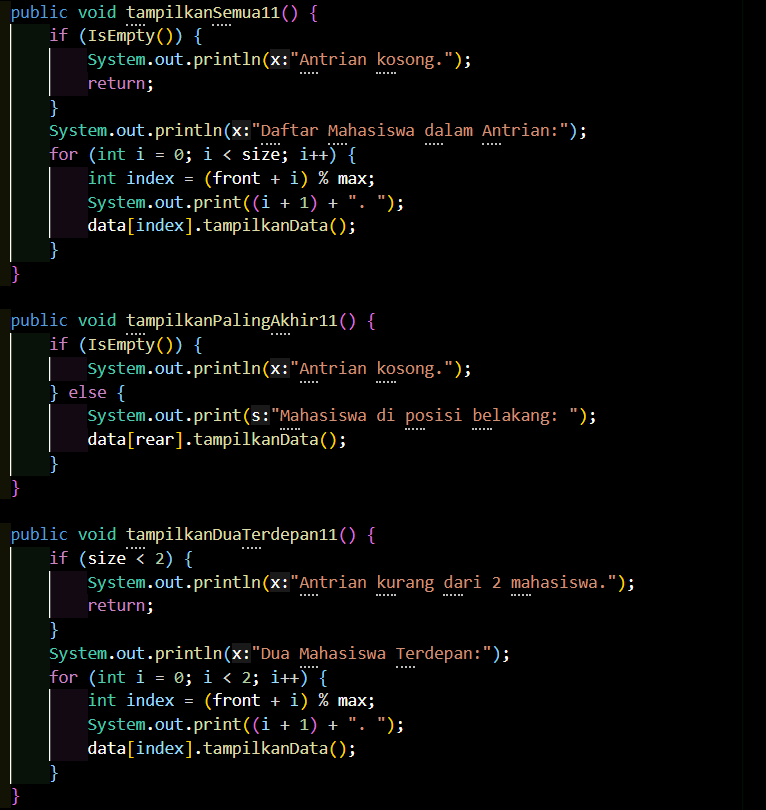


• Menambahkan antrian, Memanggil antrian untuk proses KRS – setiap 1x panggilan terdiri dari 2 mahasiswa (pada antrian no 1 dan 2)

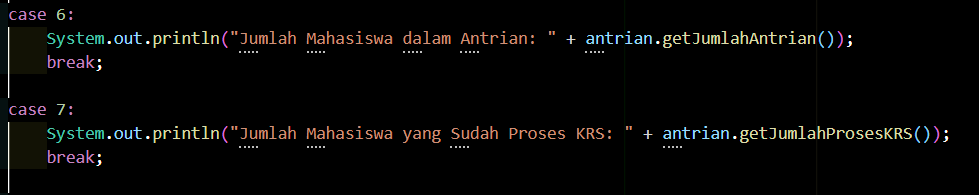




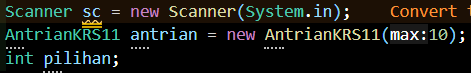
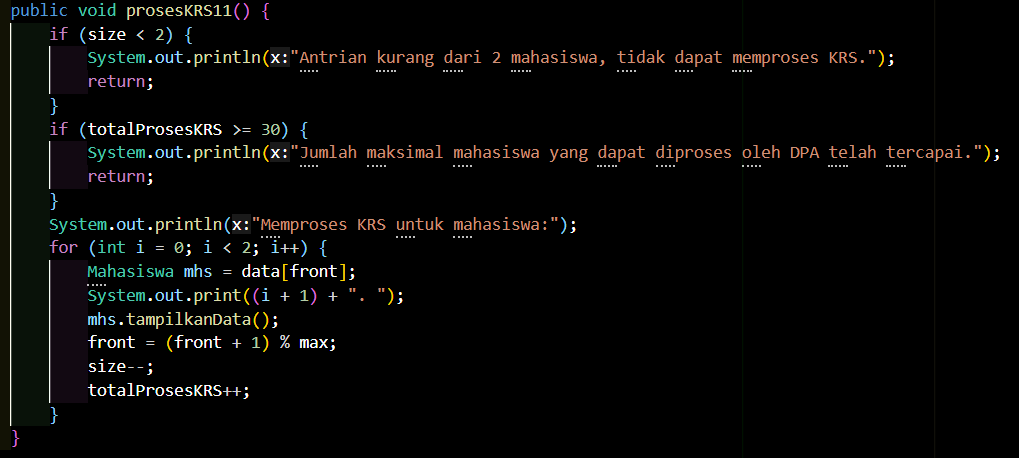
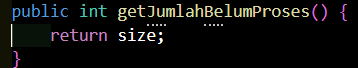
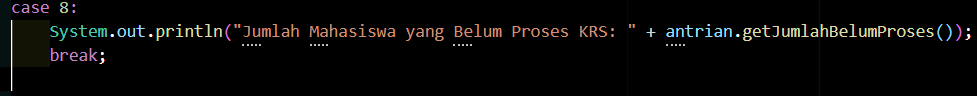
• Menampilkan semua antrian, Menampilkan 2 antrian terdepan, Menampilkan antrian paling akhir.



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

1. jumlah Antrian Maksimal 10  
   
2. Jumlah yang Ditangani Masing-Masing DPA Maksimal 30 Mahasiswa  
   
3. Mencetak Jumlah Mahasiswa yang Belum Melakukan Proses KRS  
     
   

• Diagram Class untuk AntrianKRS11

|  |
| --- |
| AntrianKRS11 |
| - data: Mahasiswa[]  - front: int  - rear: int  - size: int  - max: int  - totalProsesKRS: int |
| + AntrianKRS11(max: int)  + IsEmpty(): boolean  + IsFull(): boolean  + clear(): void  + tambahAntrian11(mhs: Mahasiswa): void  + prosesKRS11(): void  + tampilkanSemua11(): void  + tampilkanDuaTerdepan11(): void  + tampilkanPalingAkhir11(): void  + getJumlahAntrian(): int  + getJumlahProsesKRS(): int  + getJumlahBelumProses(): int |