**LAPORAN HASIL PRAKTIKUM ALSD**

**JOBSHEET 10 QUEUE**



Faiva Puspa Sahara

244107020036

TI – 1E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

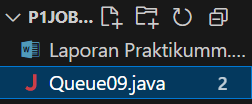
POLITEKNIK NEGERI MALANG

2025

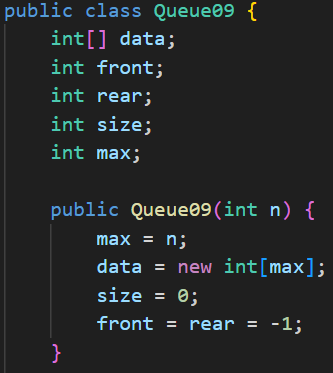
**HASIL PRAKTIKUM**

* 1. **Percobaan 1 (Mahasiswa Mengumpulkan Tugas)**

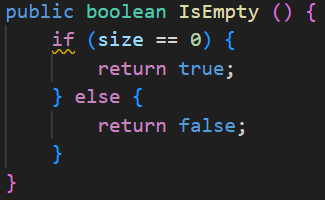
1. Buat folder baru bernama **P1Jobsheet10**. Buat file baru, beri nama **Queue09.java**



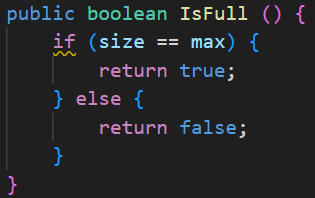
1. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya



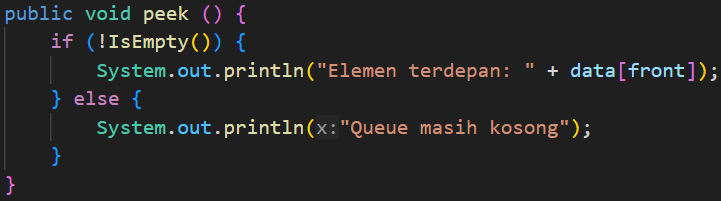
1. Buat method **IsEmpty** bertipe boolean yang digunakan untuk mengecek apakah queue kosong.



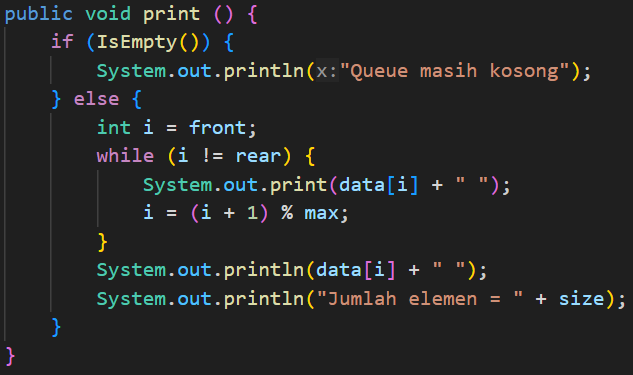
1. Buat method **IsFull** bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.



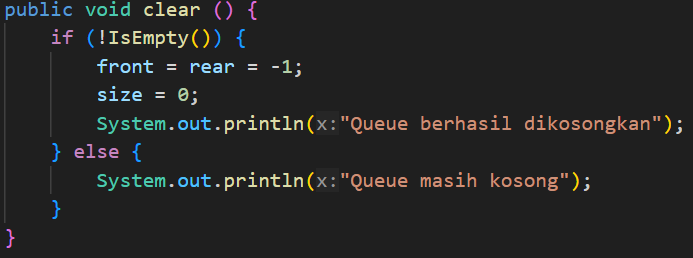
1. Buat method **peek** bertipe void untuk menampilkan elemen queue pada posisi paling depan.



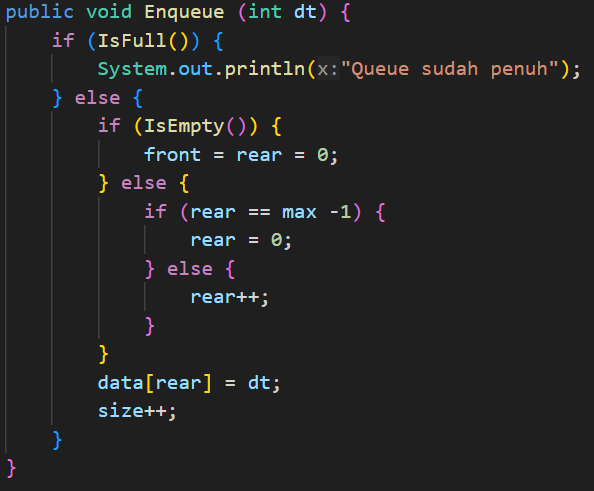
1. Buat method **print** bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.



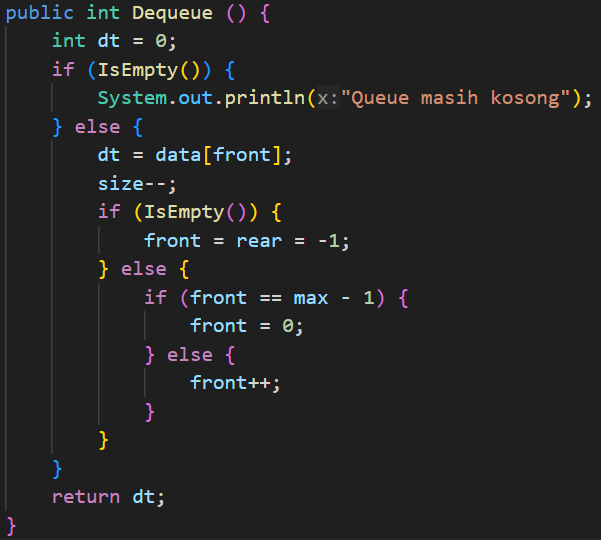
1. Buat method **clear** bertipe void untuk menghapus semua elemen pada queue



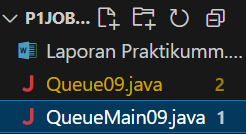
1. Buat method **Enqueue** bertipe void untuk menambahkan isi queue dengan parameter **dt** yang bertipe integer

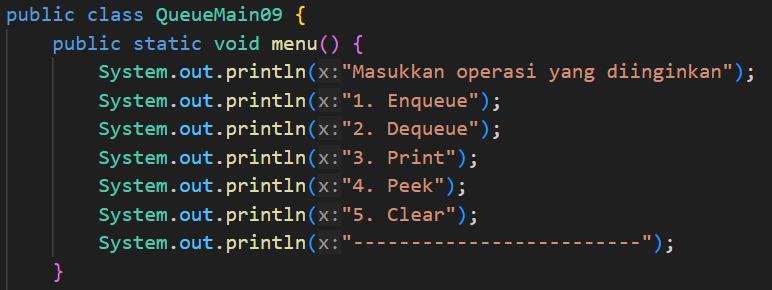


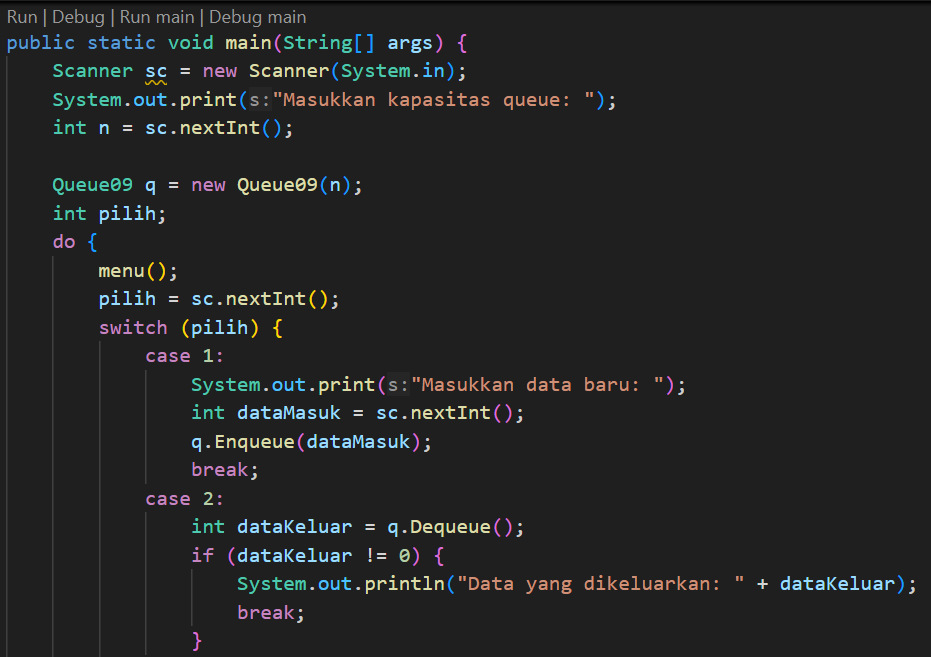
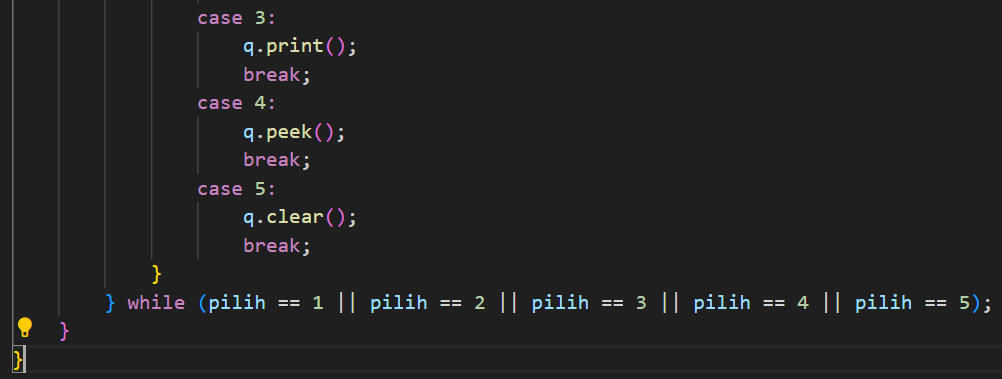
1. Buat method **Dequeue** bertipe int untuk mengeluarkan data pada queue di posisi belakang



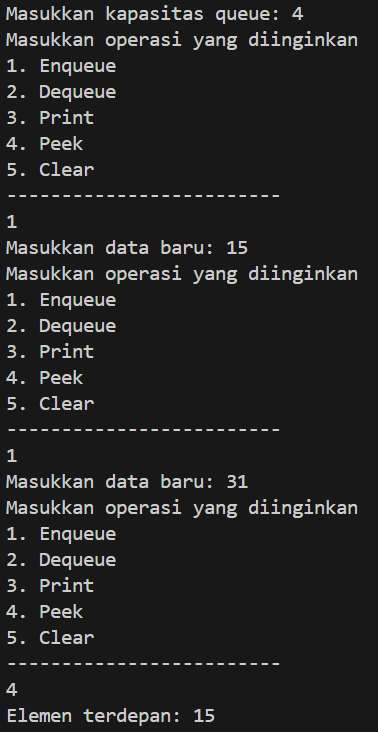
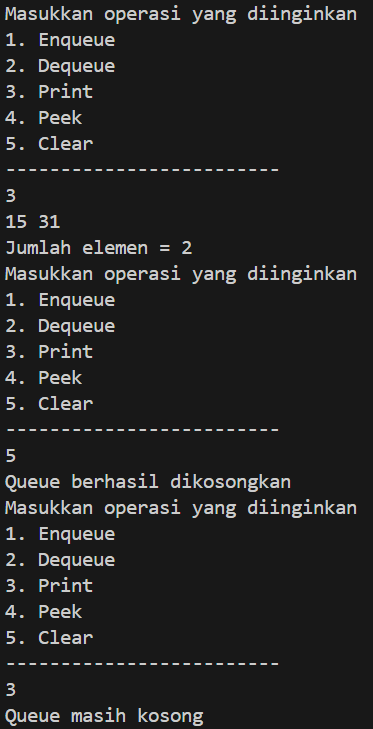
1. Selanjutnya, buat class baru dengan nama **QueueMain** tetap pada package **Praktikum1**. Buat method **menu** bertipe void untuk memilih menu program pada saat dijalankan.

* 

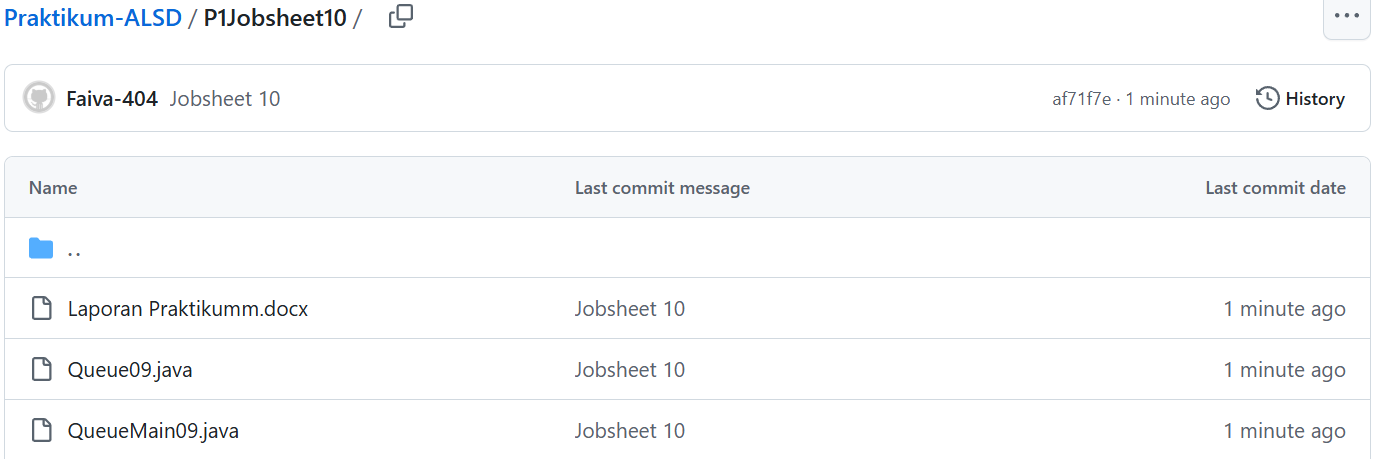


* y
* y
* y

1. Run Program

1. Commit dan push ke Github



* **Pertanyaan**

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

* Pada front dan rear bernilai -1 karena sebagai penanda jika queue masih kosong, pada array digunakan untuk queue tidak ada index -1, sementara size 0 digunakan karena queue masih kosong.

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



* Sebagai kondisi jika rear sudah mencapai posisi yg paling belakang (max -1) maka rear akan berpindah ke posisi depan (index 0) dengan catatan queue belum penuh.

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



* Sebagai kondisi jika front sudah mencapai posisi paling belakang(max -1) maka front akan dipindahkan ke posisi depan(index 0).

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

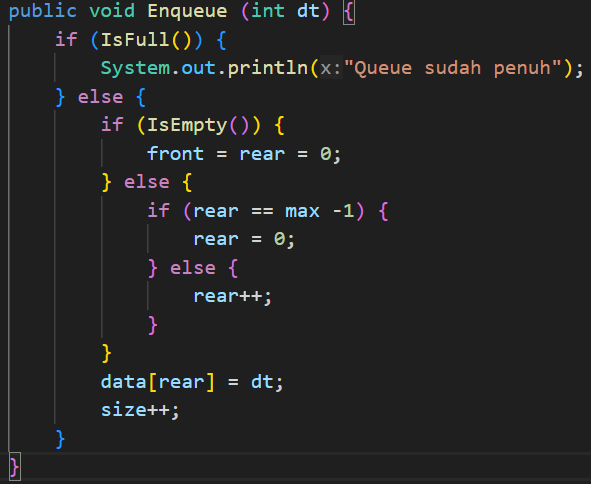
* Karena dalam queue front merupakan bagian paling depan dari queue dan juga tidak selalu berada di index 0, maka perulangan dimulai dari front agar bisa mengecek semua elemen yang ada pada queue

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

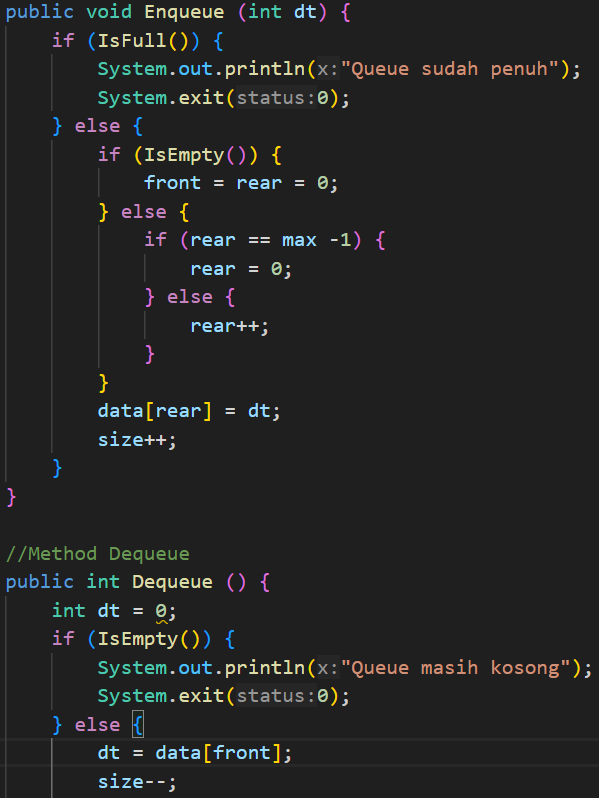
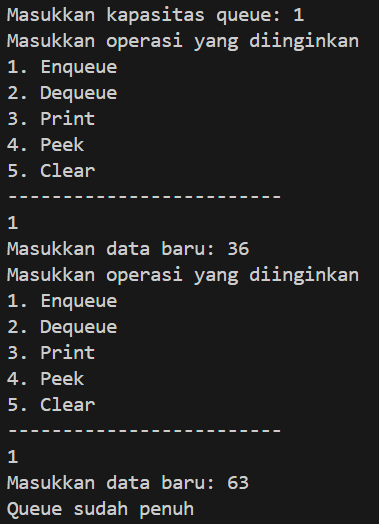
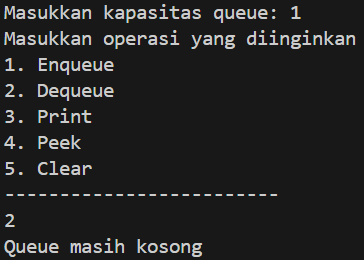


* Proses mengubah iterasi I ke elemen berikutnya. **i = (i+1) %** max memungkinkan jika i+1 sudah mencapai max maka I akan bernilai 0 kembali sesuai prinsip queue

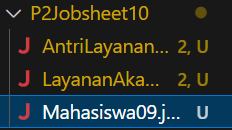
1. Tunjukkan potongan kode program yang merupakan queue overflow!

* 
* Jika rear sudah diindeks max – 1 (indeks terakhir), ketika ingin menambahkan elemen baru harus dimulai kembali dari indeks 0.

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!

* 
*  
  1. **Percobaan 2 (Antrian Layanan Akademik)**

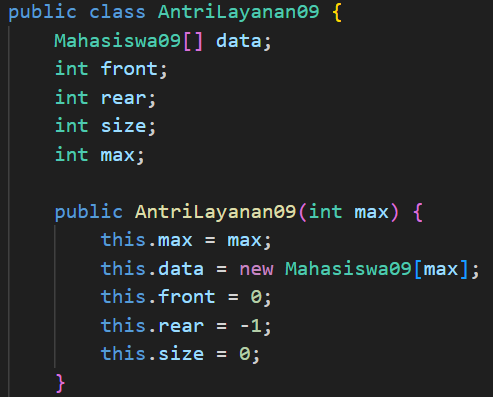
1. Buat folder baru **P2Jobsheet10**, kemudian buar class baru bernama **Mahasiswa09**



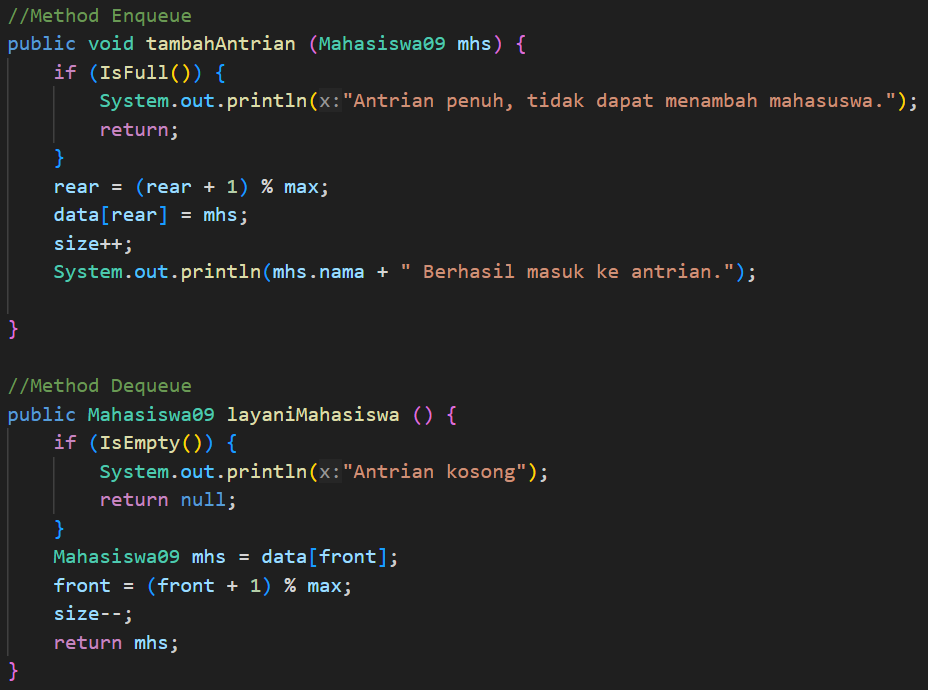
1. Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.



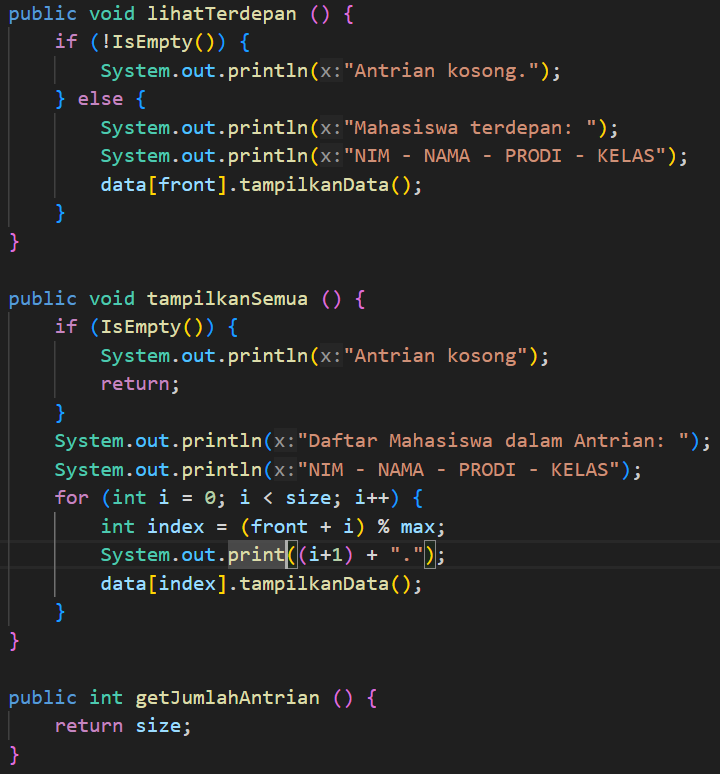
1. Salin kode program class **Queue09** pada **Praktikum 1** untuk digunakan kembali pada **Praktikum 2** ini, ganti nama class-nya dengan **AntrianLayanan09.** Karena pada **Praktikum 1**, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada **Praktikum 2** data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class **AntrianLayanan09** tersebut.



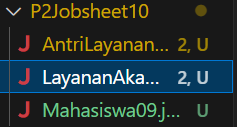
1. Lakukan modifikasi pada class **AntrianLayanan** dengan mengubah tipe **int[] data** menjadi **Mahasiswa[] data** karena pada kasus ini data yang akan disimpan berupa object Mahasiswa. Modifikasi pada **atribut**, **Enqueue**, dan **Dequeue**.



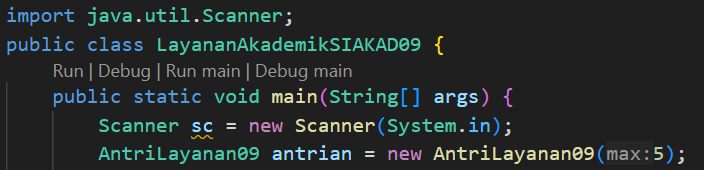
1. Berikutnya method peek dan print yaitu untuk menampilkan data antrian layanan paling depan dan menampilkan semua data antrian layanan.



1. Selanjutnya, buat class baru dengan nama **LayananAkademikSIAKAD** tetap pada package yang sama. Buat fungsi **main**, deklarasikan Scanner dengan nama **sc.**



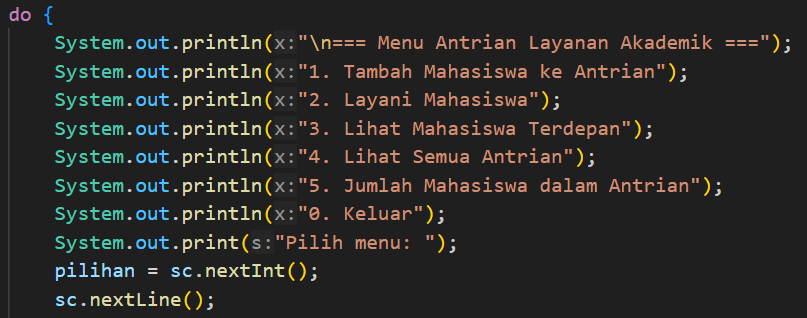
1. Kemudian lakukan instansiasi objek **AntrianLayanan** dengan nama **antrian** dan nilai parameternya adalah nilai maksimal antrian yang ditentukan (misal sama dengan 5).

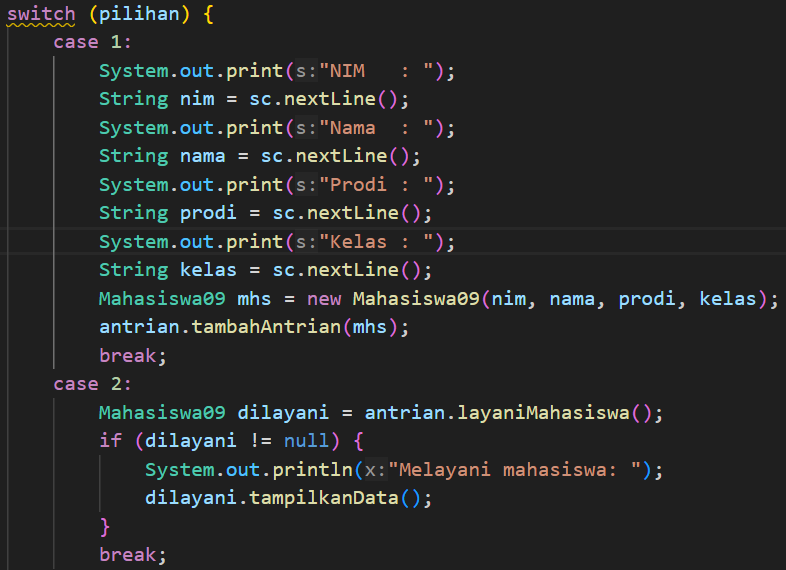


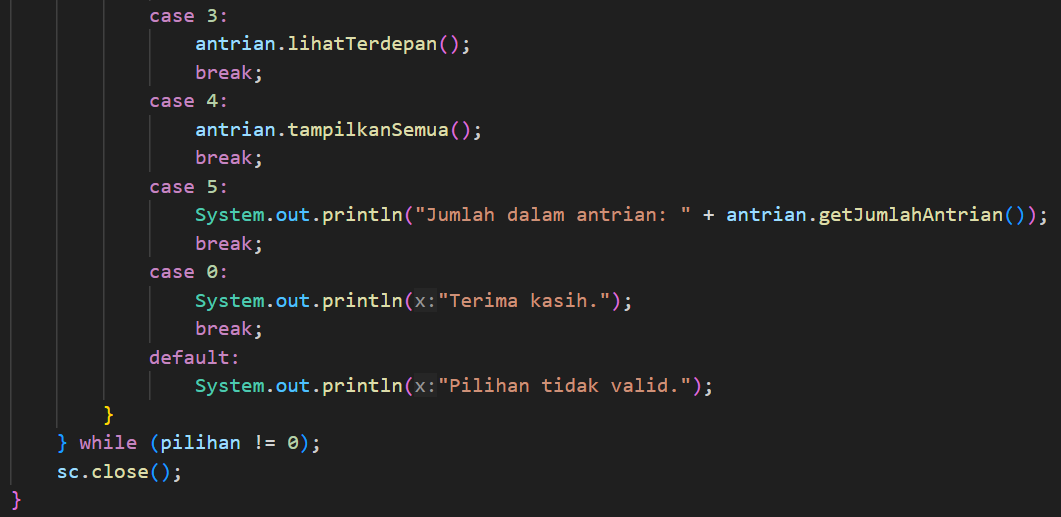
1. Deklarasikan variabel dengan nama **pilihan** bertipe integer untuk menampung pilih menu dari pengguna



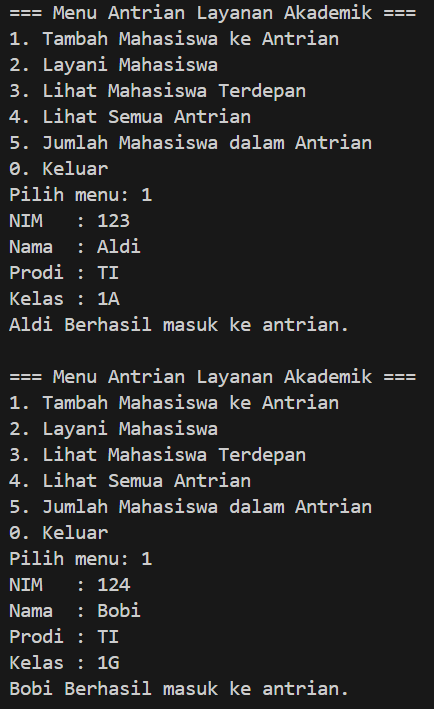
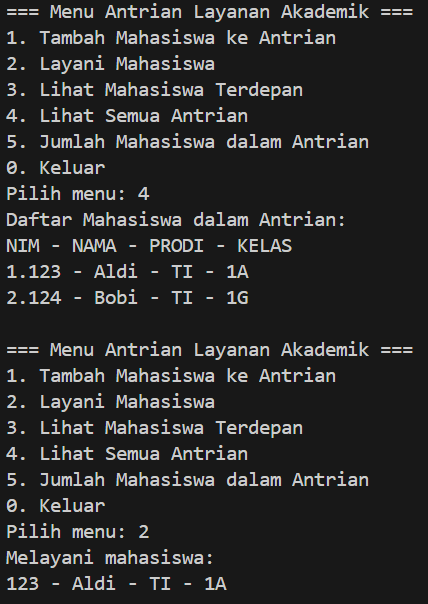
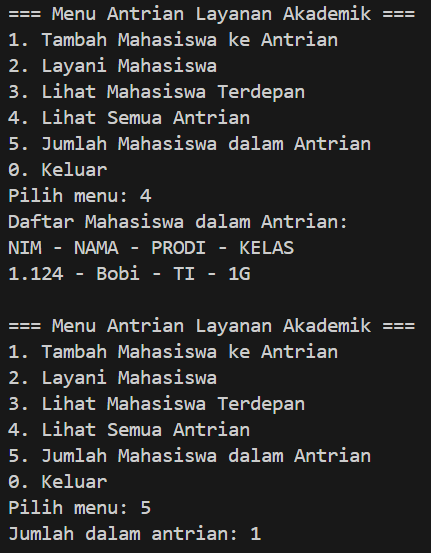
1. Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.

U





1. Run Program

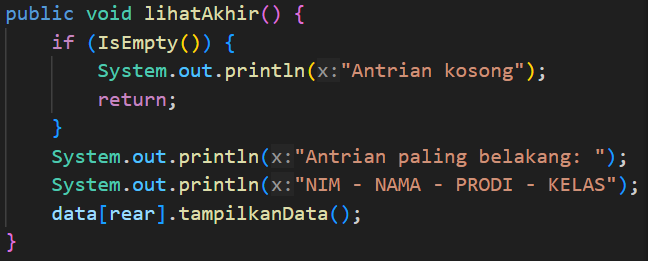
1. Commit dan Push ke Github



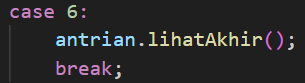
* **Pertanyaan**

1. Lakukan modifikasi program dengan menambahkan method baru bernama **LihatAkhir** pada class **AntrianLayanan09** yang digunakan untuk mengecek antrian yang berada di posisi belakang. Tambahkan pula daftar menu **6. Cek Antrian paling belakang** pada class **LayananAkademikSIAKAD09** sehingga method **LihatAkhir** dapat dipanggil!

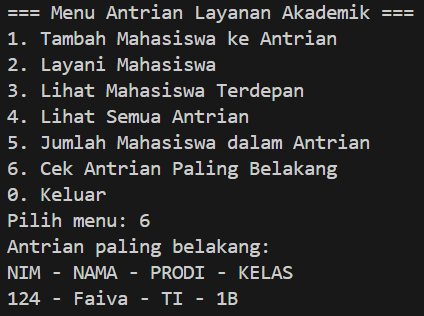
* Kode program



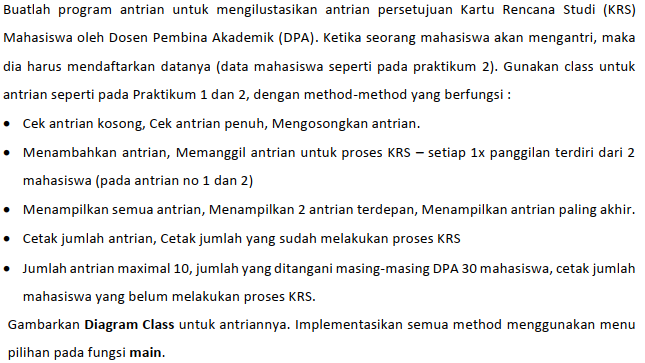




* Run



* 1. **Tugas**



* **Class Surat09.java**

package Jobsheet9;

public class Surat09 {

String idSurat;

String namaMahasiswa;

String kelas;

char jenisIzin;

int durasi;

public Surat09() {

}

public Surat09(String idSurat, String namaMahasiswa, String kelas, char jenisIzin, int durasi) {

this.idSurat = idSurat;

this.namaMahasiswa = namaMahasiswa;

this.kelas = kelas;

this.jenisIzin = jenisIzin;

this.durasi = durasi;

}

}

* **Class StackSurat09.java**

package Jobsheet9;

public class StackSurat09 {

Surat09[] stack;

int size;

int top;

public StackSurat09(int size) {

this.size = size;

stack = new Surat09[size];

top = -1;

}

public boolean isFull() {

return top == size - 1;

}

public boolean isEmpty() {

return top == -1;

}

public void push(Surat09 surat) {

if (!isFull()) {

top++;

stack[top] = surat;

} else {

System.out.println("Stack penuh! Tidak bisa menambahkan surat lagi.");

}

}

public Surat09 pop() {

if (!isEmpty()) {

Surat09 surat = stack[top];

top--;

return surat;

} else {

System.out.println("Stack kosong! Tidak ada surat untuk diambil.");

return null;

}

}

public Surat09 peek() {

if (!isEmpty()) {

return stack[top];

} else {

System.out.println("Stack kosong! Tidak ada surat yang dikumpulkan.");

return null;

}

}

public boolean search(String namaMahasiswa) {

for (int i = top; i >= 0; i--) {

if (stack[i].namaMahasiswa.equals(namaMahasiswa)) {

return true;

}

}

return false;

}

}

System.out.println("Stack kosong! Tidak ada surat untuk diambil.");

return null;

}

}

public Surat09 peek() {

if (!isEmpty()) {

return stack[top];

} else {

System.out.println("Stack kosong! Tidak ada surat yang dikumpulkan.");

return null;

}

}

public boolean search(String namaMahasiswa) {

for (int i = top; i >= 0; i--) {

if (stack[i].namaMahasiswa.equals(namaMahasiswa)) {

return true;

}

}

return false;

}

}

* **Class SuratDemo09.java**

package Jobsheet9;

import java.util.Scanner;

public class SuratDemo09 {

public static void main(String[] args) {

StackSurat09 stack = new StackSurat09(5);

Scanner sc = new Scanner(System.in);

int pilih;

do {

System.out.println("\nMenu: ");

System.out.println("1. Terima Surat Izin");

System.out.println("2. Proses Surat Izin");

System.out.println("3. Lihat Surat Izin Terakhir");

System.out.println("4. Cari Surat Izin");

System.out.print("Pilih : ");

pilih = sc.nextInt();

sc.nextLine();

switch (pilih) {

case 1 :

if (stack.isFull()) {

System.out.println("Stack penuh! Tidak bisa menambahkan surat lagi.");

} else {

System.out.print("ID Surat : ");

String idSurat = sc.nextLine();

System.out.print("Nama Mahasiswa : ");

String namaMahasiswa = sc.nextLine();

System.out.print("Kelas : ");

String kelas = sc.nextLine();

System.out.print("Jenis Izin (S/I): ");

char jenisIzin = sc.next().charAt(0);

System.out.print("Durasi Izin : ");

int durasi = sc.nextInt();

Surat09 surat = new Surat09(idSurat, namaMahasiswa, kelas, jenisIzin, durasi);

stack.push(surat);

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diterima.");

}

break;

case 2 :

Surat09 surat = stack.pop();

if (surat != null) {

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diproses.");

}

break;

case 3 :

Surat09 suratTerakhir = stack.peek();

if (suratTerakhir != null) {

System.out.println("Surat izin terakhir: ");

System.out.println("ID Surat : " + suratTerakhir.idSurat);

System.out.println("Nama Mahasiswa: " + suratTerakhir.namaMahasiswa);

System.out.println("Jenis Izin : " + suratTerakhir.jenisIzin + " (S/I)");

System.out.println("Durasi Izin : " + suratTerakhir.durasi + " hari");

}

break;

case 4 :

System.out.print("Nama Mahasiswa yang dicari: ");

String cariNama = sc.nextLine();

boolean ditemukan = stack.search(cariNama);

if (ditemukan) {

System.out.println("Surat izin untuk " + cariNama + " ditemukan.");

} else {

System.out.println("Surat izin untuk " + cariNama + " tidak ditemukan.");

}

break;

default:

System.out.println("Pilihan tidak valid. Silakan coba lagi.");

break;

}

} while (pilih >= 1 && pilih <= 4);

}

}

char jenisIzin = sc.next().charAt(0);

System.out.print("Durasi Izin : ");

int durasi = sc.nextInt();

Surat09 surat = new Surat09(idSurat, namaMahasiswa, kelas, jenisIzin, durasi);

stack.push(surat);

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diterima.");

}

break;

case 2 :

Surat09 surat = stack.pop();

if (surat != null) {

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diproses.");

}

break;

case 3 :

Surat09 suratTerakhir = stack.peek();

if (suratTerakhir != null) {

System.out.println("Surat izin terakhir: ");

System.out.println("ID Surat : " + suratTerakhir.idSurat);

System.out.println("Nama Mahasiswa: " + suratTerakhir.namaMahasiswa);

System.out.println("Jenis Izin : " + suratTerakhir.jenisIzin + " (S/I)");

System.out.println("Durasi Izin : " + suratTerakhir.durasi + " hari");

}

break;

case 4 :

System.out.print("Nama Mahasiswa yang dicari: ");

String cariNama = sc.nextLine();

boolean ditemukan = stack.search(cariNama);

if (ditemukan) {

System.out.println("Surat izin untuk " + cariNama + " ditemukan.");

} else {

System.out.println("Surat izin untuk " + cariNama + " tidak ditemukan.");

}

break;

default:

System.out.println("Pilihan tidak valid. Silakan coba lagi.");

break;

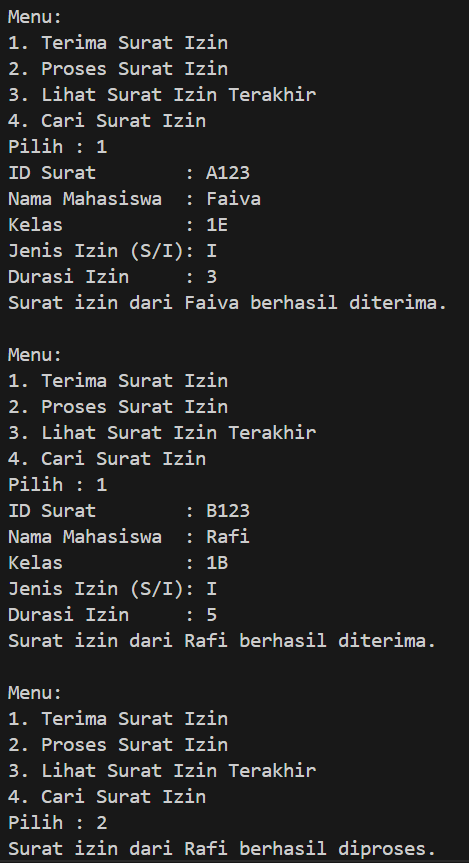
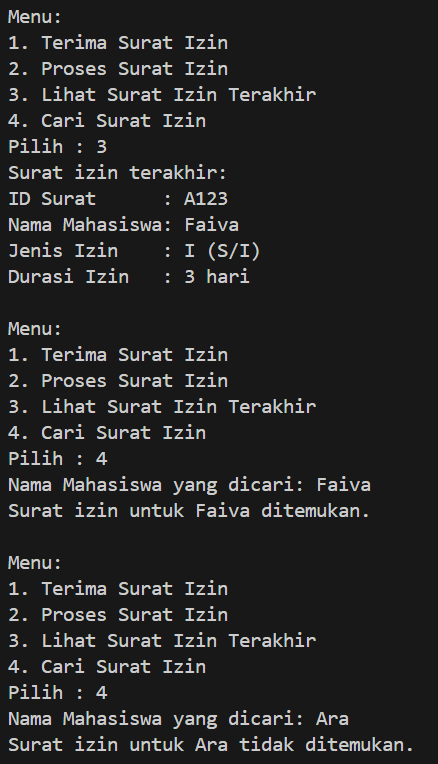
}

} while (pilih >= 1 && pilih <= 4);

}

}

* Run Program

* Commit dan Push ke Github

