**LAPORAN PRAKTIKUM**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**

Dosen Pengampu : Triana Fatmawati, S.T, M.T

**PERTEMUAN - 10 - Queue**

****

**Nama : M. Zidna Billah Faza**

**NIM : 2341760030**

**Prodi : D-IV Sistem Informasi Bisnis**

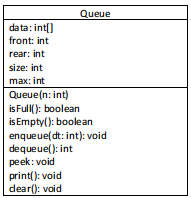
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

**Percobaan 1**

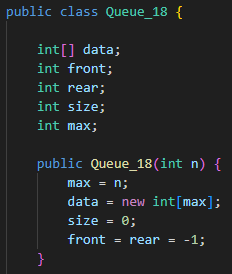
1. Perhatikan Diagram Class Queue berikut ini:



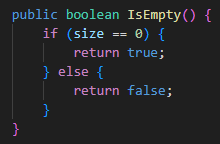
1. Buat package dengan nama Praktikum1, kemudian buat class baru dengan nama Queue.



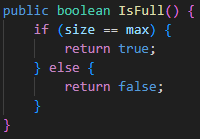
1. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.



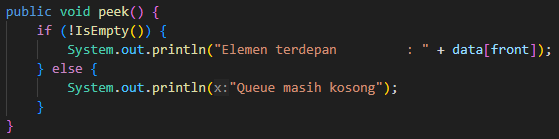
1. Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong.



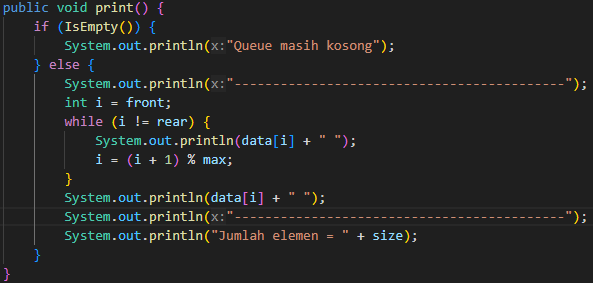
1. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.



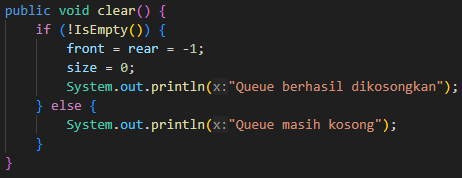
1. Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.



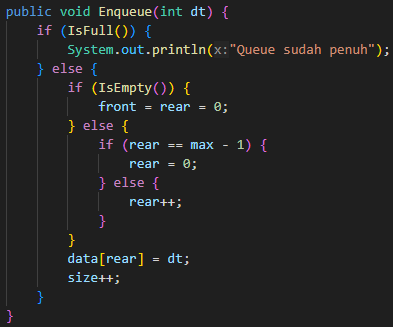
1. Buat method print bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.



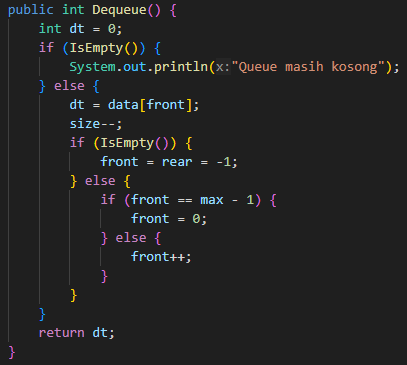
1. Buat method clear bertipe void untuk menghapus semua elemen pada queue



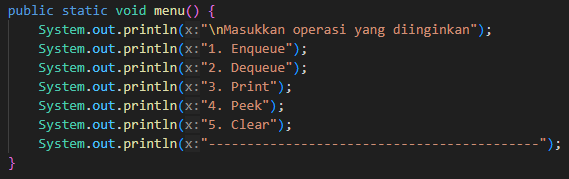
1. Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer



1. Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi paling depan



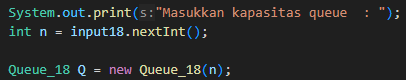
1. Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan.



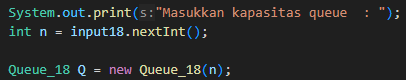
1. Buat fungsi main, kemudian deklarasikan Scanner dengan nama input18.



1. Buat variabel n untuk menampung masukan berupa jumlah maksimal elemen yang dapat disimpan pada queue.



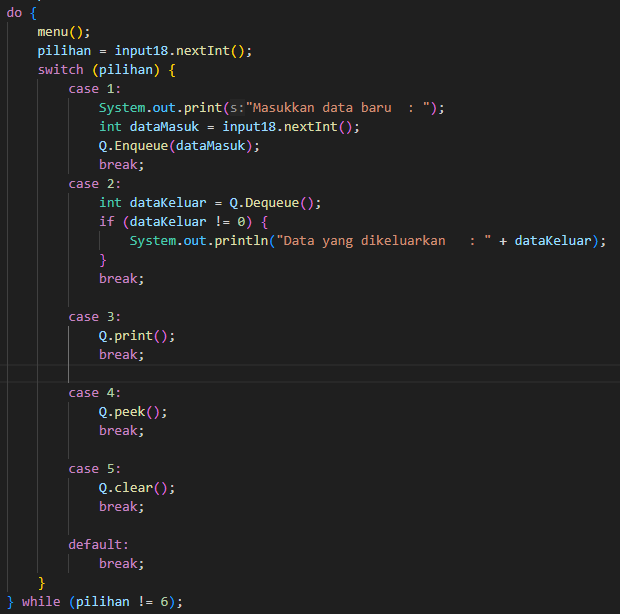
1. Lakukan instansiasi objek Queue dengan nama Q dengan mengirimkan parameter n sebagai kapasitas elemen queue



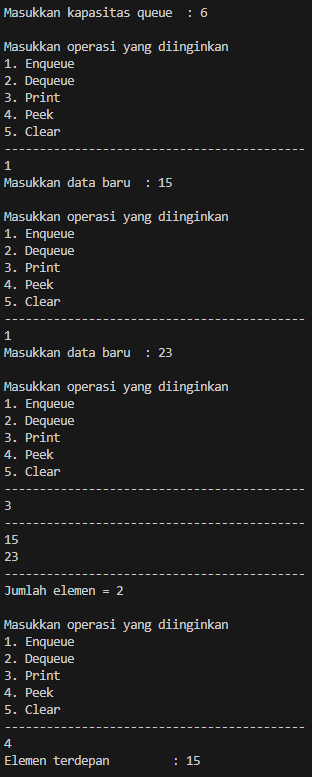
1. Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

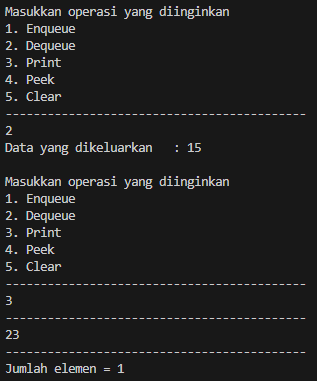


1. Lakukan perulangan menggunakan do-while untuk menjalankan program secara terus menerus sesuai masukan yang diberikan. Di dalam perulangan tersebut, terdapat pemilihan kondisi menggunakan switch-case untuk menjalankan operasi queue sesuai dengan masukan pengguna.



1. Compile dan jalankan class QueueMain, kemudian amati hasilnya dan verifikasi.





Pertanyaan Percobaan 1

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

Karena pada front dan rear ketika diberikan nilai awal -1 menandakan bahwa antrian kosong atau tidak ada elem dalam antrian, sementara pada size diberikan nilai awal 0 karena pada awal program tidak ada antrian yang ada dialamnya.

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



Potongan kkode tersebut bertujuan untuk menangani situasi ketika variale ‘rear’ mencapai batas maximalnya yang telah ditentukan oleh ‘max - 1’. Ketika kondisi ‘rear == max -1’ terpenuhi maka elemen terakhir dari array sudah diisi dan apabila akan menambahkan elemen baru maka dimulai dari indeks 0 lagi

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



Potongan kkode tersebut bertujuan untuk menangani situasi ketika variale ‘front’ mencapai batas maximal antrian. Ketika kondisi ‘rear == max -1 terpenuhi maka elemen terakhir dalam array telah diakses dan akan dikeluarkan dari antrian namun apabila kita ingin menggunakan array pada indeks awal jika memungkinkan program akan ‘front’ pada indeks awal yaitu 0

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

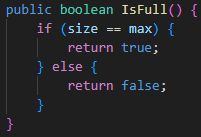
Karena program ‘print’ digunakan untuk menngeprint elemen dari awal antrian bukan awal indeks karena pada queue urutan antrian tidak mesti dari indeks ke 0

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



Potongan kode `i = (i + 1) % max;` digunakan untuk memperbarui nilai `i` secara bertahap dari `front` ke `rear` dalam array antrian. Operasi modulus (`%`) memastikan bahwa jika `i` mencapai batas maksimum `max`, ia akan kembali ke 0, sehingga perulangan dapat berlanjut secara siklikal melalui array antrian.

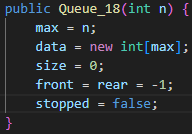
1. Tunjukkan potongan kode program yang merupakan queue overflow!

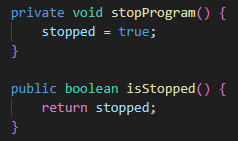


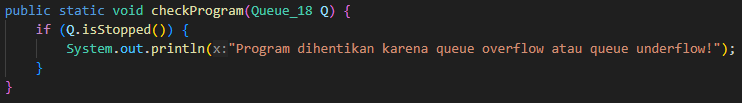


1. Pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan!

Modifikasi dilakukan dengan menambahkan boolean dapat dilihat pada github

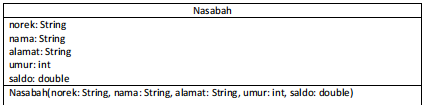
 



**Percobaan 2**

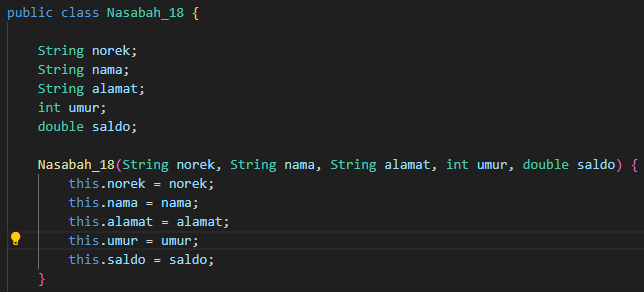
1. Perhatikan Diagram Class berikut ini:



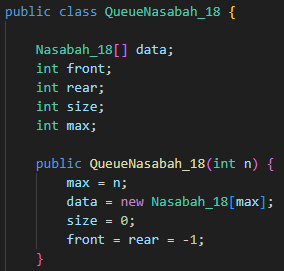
1. Buat package dengan nama Praktikum2, kemudian buat class baru dengan nama Nasabah.

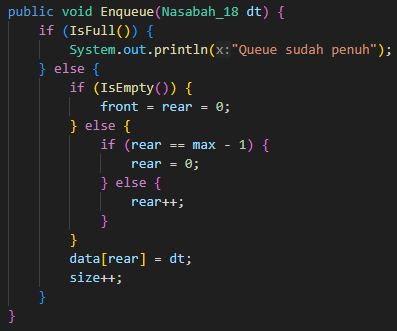


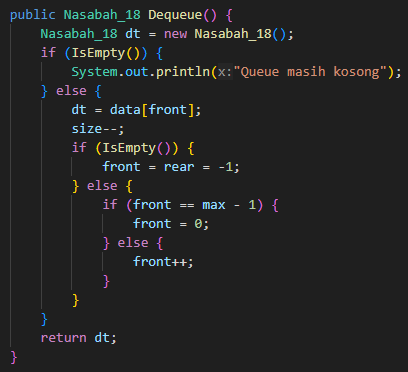
1. Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini

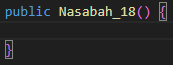


1. Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini. Karena pada Praktikum 1, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada Praktikum 2 data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class Queue tersebut.
2. Lakukan modifikasi pada class Queue dengan mengubah tipe int[] data menjadi Nasabah[] data karena pada kasus ini data yang akan disimpan pada queue berupa object Nasabah. Modifikasi perlu dilakukan pada atribut, method Enqueue, dan method Dequeue. Baris program Nasabah dt = new Nasabah(); akan ditandai sebagai error, untuk mengatasinya, tambahkan konstruktor default di dalam class Nasabah.

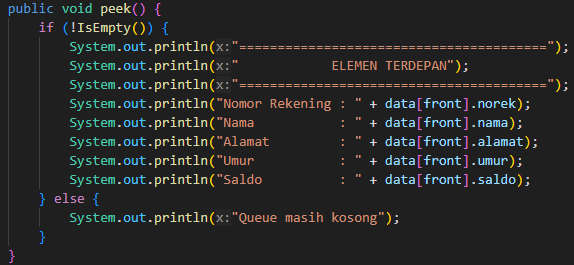


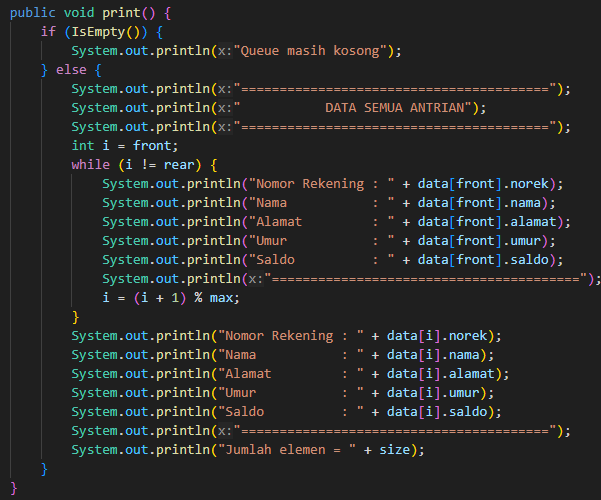




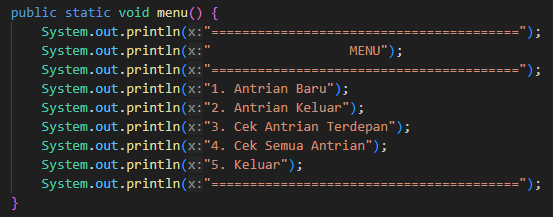


1. Karena satu elemen queue terdiri dari beberapa informasi (norek, nama, alamat, umur, dan saldo), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut, sehingga meodifikasi perlu dilakukan pada method peek dan method print.





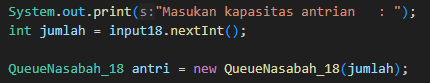
1. Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum2. Buat method menu untuk mengakomodasi pilihan menu dari masukan pengguna



1. Buat fungsi main, deklarasikan Scanner dengan nama input18



1. Buat variabel jumlah untuk menampung kapasitas elemen pada queue. Kemudian lakukan instansiasi objek queue dengan nama antri dan nilai parameternya adalah variabel jumlah.



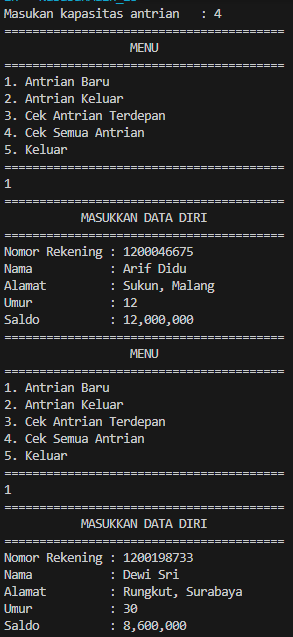
1. Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

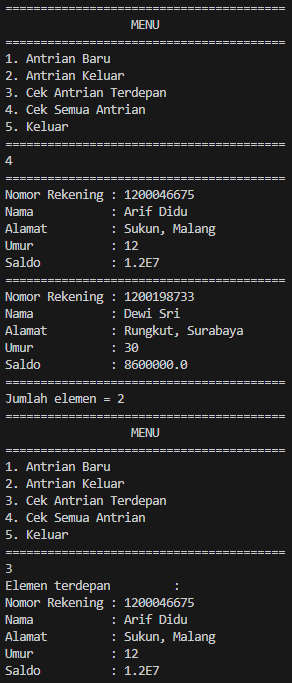


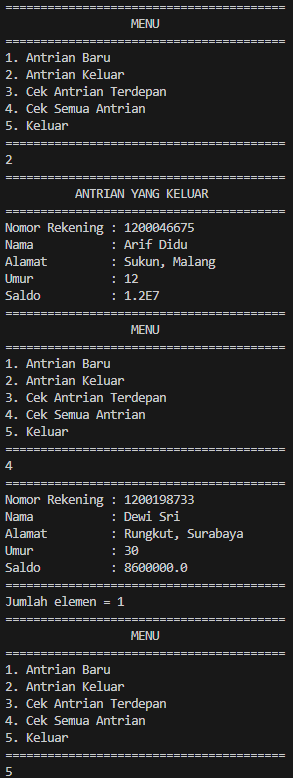
1. Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.



1. Compile dan jalankan class QueueMain, kemudian amati hasilnya dan verifikasi.





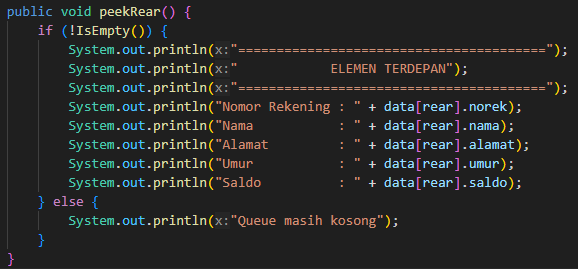


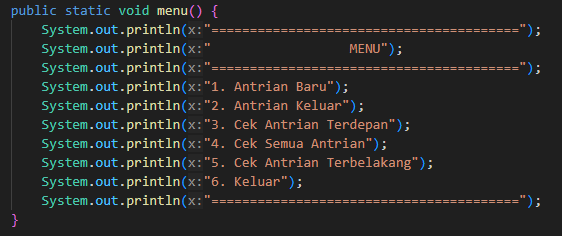
**Pertanyaan Percobaan 2**

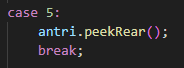
1. Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!

Kode ini memeriksa apakah data nasabah yang diambil dari antrian memiliki nilai yang valid untuk nomor rekening (norek), nama, alamat, umur, dan saldo. Jika salah satu dari data tersebut tidak kosong (tidak sama dengan string kosong) atau umur dan saldo tidak sama dengan 0, maka kondisi tersebut dianggap sebagai data yang valid dan akan dijalankan perintah untuk menampilkan data nasabah yang keluar dari antrian.

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!

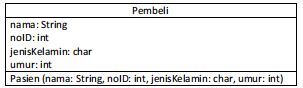


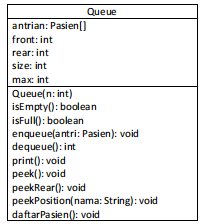




**Tugas Praktikum**

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:





Keterangan method:

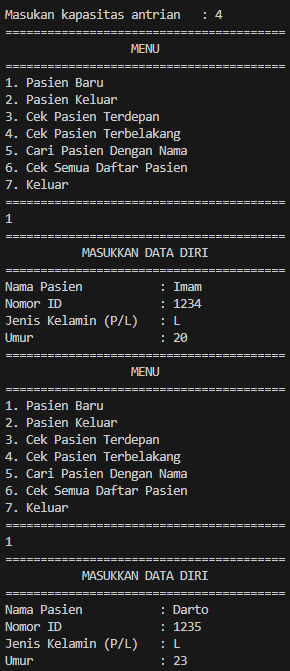
* Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada Praktikum
* Method peek(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling depan
* Method peekRear(): digunakan untuk menampilkan data Pasien yang berada 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

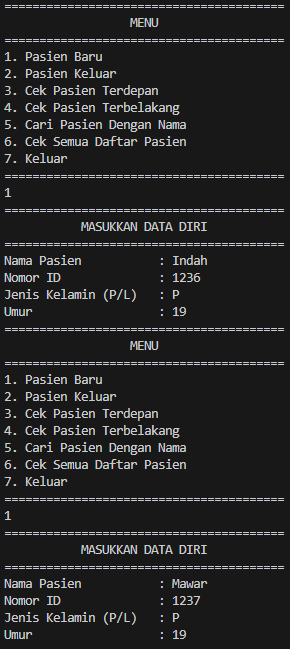
**Jawaban**

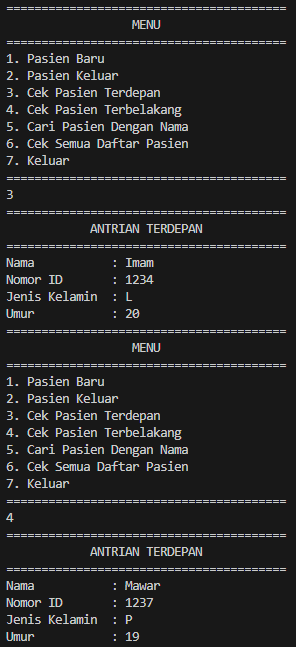
1. a. Program

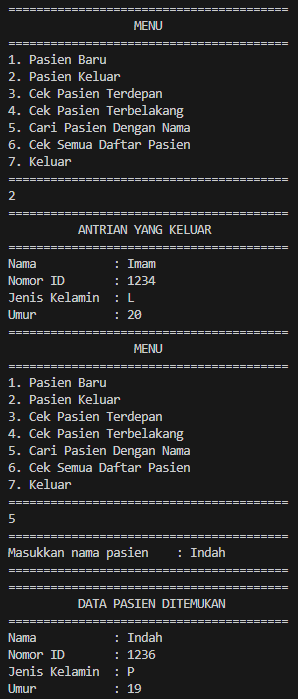


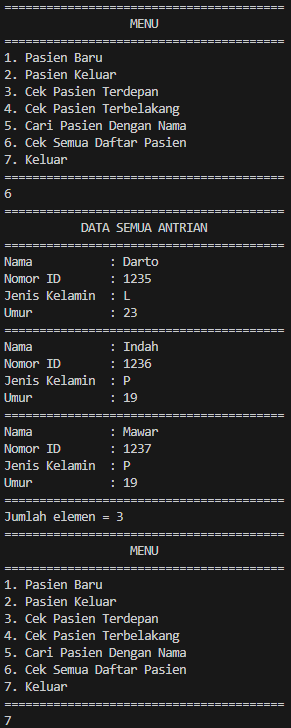
1. b. Output











**Repository : <https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data>**