Tugas Pendahuluan Modul 8

STRUKTUR DATA - Ganjil 2024/2025

"08\_Queue"

Ketentuan Tugas Pendahuluan

* 1. Tugas Pendahuluan dikerjakan secara Individu.
  2. TP ini bersifat WAJIB, tidak mengerjakan = PENGURANGAN POIN JURNAL / TES ASESMEN.
  3. Hanya MENGUMPULKAN tetapi TIDAK MENGERJAKAN = PENGURANGAN POIN JURNAL / TES ASESMEN.
  4. Deadline pengumpulan TP Modul 2 adalah Senin, 30 September 2024 pukul 07.30 WIB.
  5. TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU

TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.

* 1. DILARANG PLAGIAT (PLAGIAT = E).
  2. Kerjakan TP dengan jelas agar dapat dimengerti.
  3. Codingan diupload di Github dan upload Laporan di Lab menggunakan format PDF dengan ketentuan:

TP\_MOD\_[XX]\_NIM\_NAMA.pdf

CP (WA):

* + Andini (082243700965)
  + Imelda (082135374187)

**SELAMAT MENGERJAKAN^^**

# LAPORAN PRAKTIKUM

# PERTEMUAN 8

# STRUKTUR DATA



## Nama :

Zulfa Mustafa Akhyar Iswahyudi (2311104010)

## Dosen :

Yudha Islami Sulistya

# PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

# FAKULTAS INFORMATIKA

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. Tujuan

Untuk melatih kompetensi Mahasiswa untuk memperdalam skill pemrograman C++

1. Tools

Codeblocks, VSCode, Github

**LATIHAN – UNGUIDED**

Dari Kodingan yang dibuat hari senin minggu ke-8 bersama Asprak, didapatkan hasil kodingan berikut :



Sedangkan untuk pengayaan Unguided, diberikan tiga perintah berikut :

* 1. Ubahlah penerapan konsep queue pada bagian guided dari array menjadi linked list
  2. Dari nomor 1 buatlah konsep antri dengan atribut Nama mahasiswa dan NIM Mahasiswa
  3. Modifikasi program pada soal 1 sehingga mahasiswa dapat diprioritaskan berdasarkan NIM (NIM yang lebih kecil didahulukan pada saat output).

Noted : Untuk data mahasiswa dan nim dimasukan oleh user

Dari ketiga soal, saya sudah ubah seluruh struktur kode program Queue menjadi

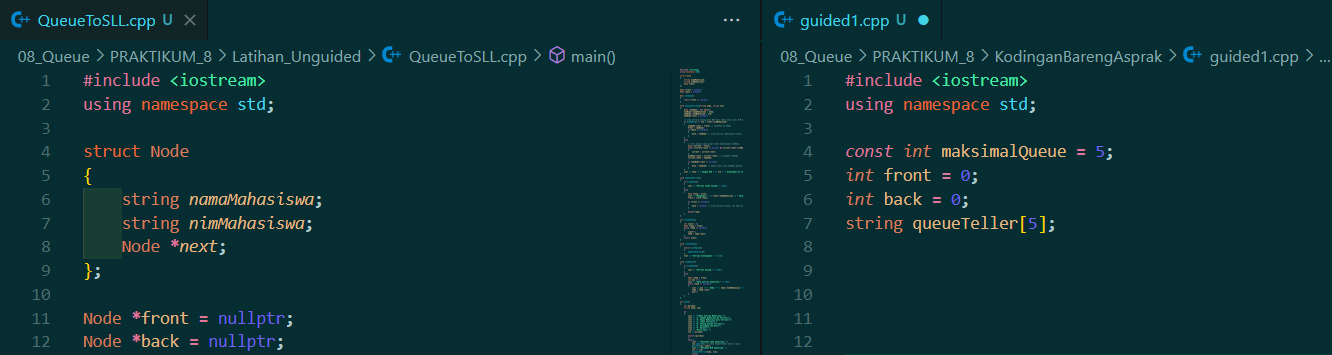
basis SLL dan **mengubah variabel Queue menjadi dua variabel SLL yaitu ‘Nama’ dan**

**‘NIM‘, serta mengaturnya agar program mengurutkan Nama/NIM Mahasiswa**

**secara *ascending* (urut terkecil -> teratas)**

Berikut penjelasan perbandingan antar struktural program :

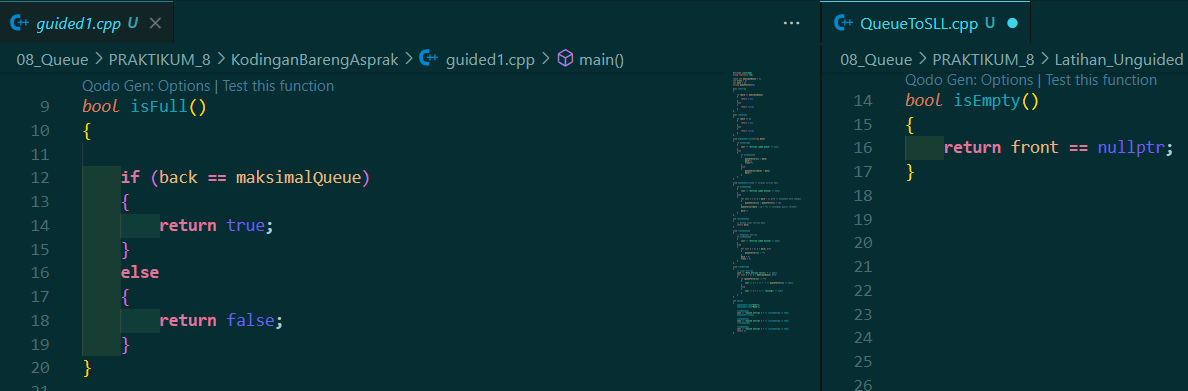
1. **Inisialisasi Awal**



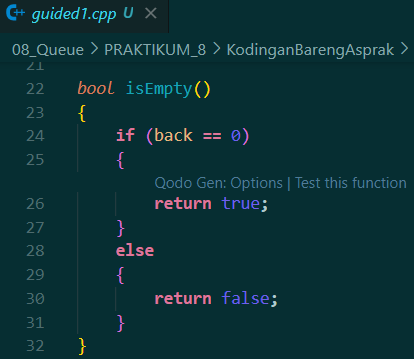
Bisa kita lihat perbedaannya cukup banyak antara basis Queue dan SLL. Pada SLL, kita membuat struktur dari linkedlist yang berisikan tipe data dan inisialisasi Node \*next agar data pada linkedlist dapat terus bertambah.

Sedangkan pada Queue, kita menginisialisasikan agar antrian berjumlah 5 elemen data dengan set default data ‘front’ dan data ‘back’ adalah nol. Buat deklarasi parameter ‘queueTeller’ dengan tipe data string nantinya.

1. Method True/False

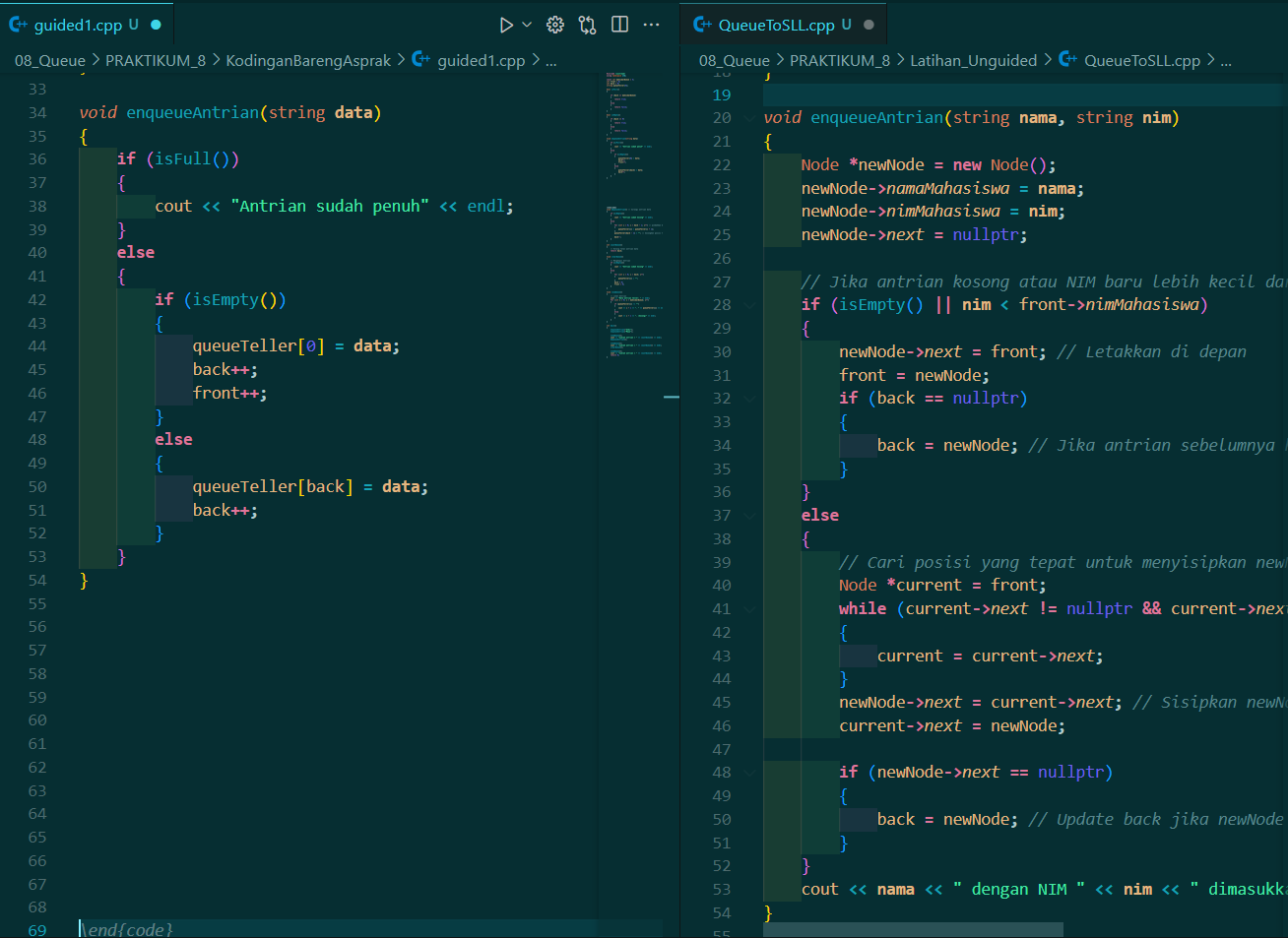


Seperti pada gambar, pada ‘Queue’ kita prioritaskan mthod boolean untuk mengecek kapasitas pada struktur Queue dengan pengkondisian nilai terakhir dengan konstanta. Sedangkan pada File Unguided ‘QueueToSLL‘, kita cukup melakukan method True/False untuk mengecek apakah linkedlist kosong atau tidak dengan patokan data depan sebagai default-nya.



Namun pada ‘Queue‘ juga ada Method boolean untuk mengecek apakah struktur antrian kosong atau tidak. Jika data paling belakang dalam antrian adalah nol, maka akan bernilai true.

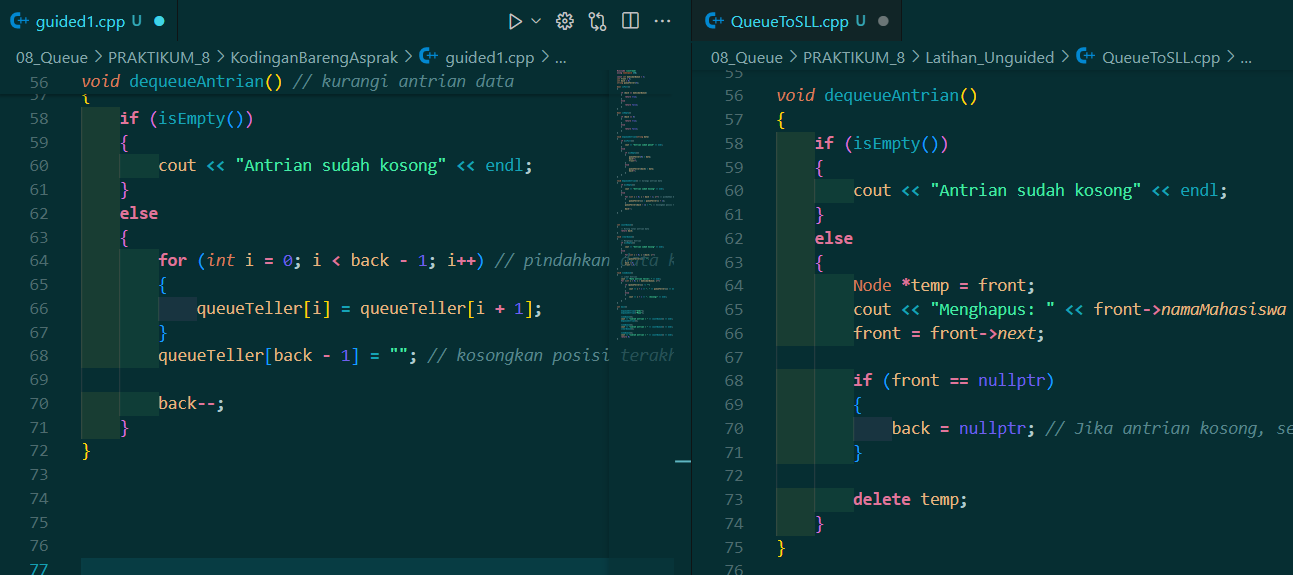
1. Method Enqueue



Pada Queue biasa method enqueue data diparameterkan dengan String data lalu lanjut ke pengkondisian fungsi isFull dan isEmpty yang bertujuan untuk pengecekan menyeluruh pada antrian yang akan diisikan data inputan.

Sementara pada SLL, method didefinisikan dengan Node yang membawa newNode untuk menginisialisasi variabel nama dan nim mahasiswa. Kemudian berlanjut pada pengecekan linkedlist beserta settingan Node pada pengkondisian ‘while true’ untuk memastikan bahwa data bisa terus bertambah selama pengkondisian bernilai benar.

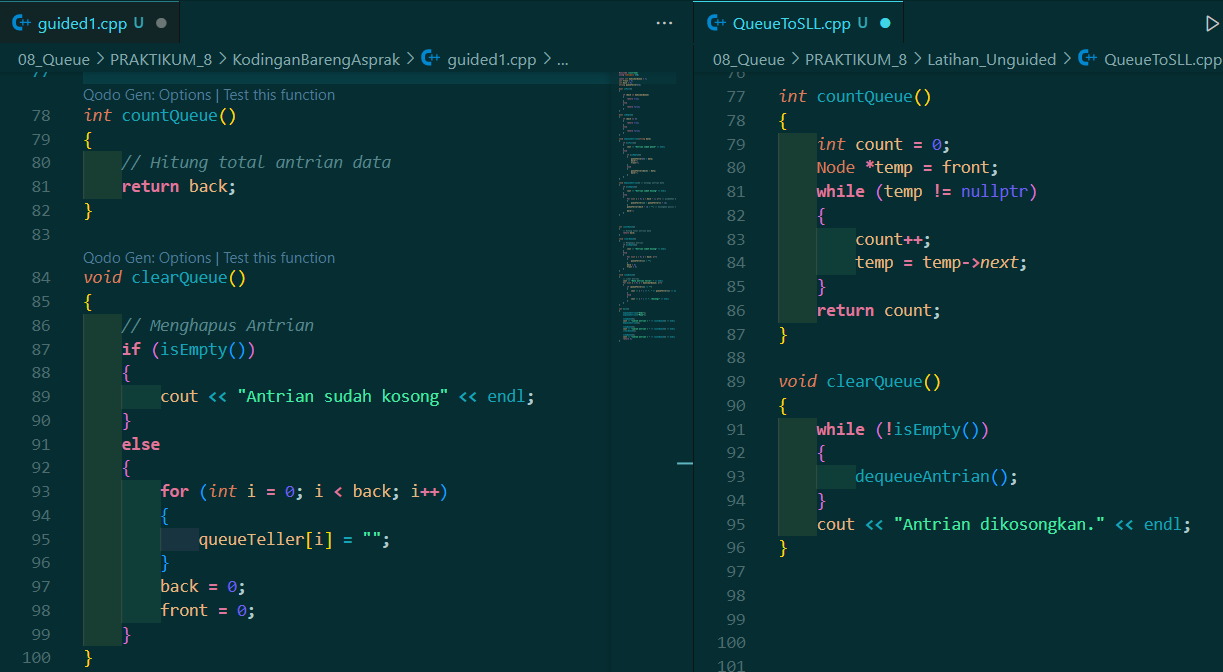
1. Method Dequeue



Untuk Queue biasa method ini memberikan output bahwa antrian kosong pada function isEmpty. Untuk penghapusan data pada antrian dilakukan perulangan yang berjalan melalui data paling belakang dulu yang akan dihapus atau data terakhir yang masuk.

Sedangkan untuk linkedlist, untuk menghapus data dilakukan peng-iterasian yang akan mendeteksi data depan dan seterusnya sampai data paling belakang yang akan dideteksi terakhir.

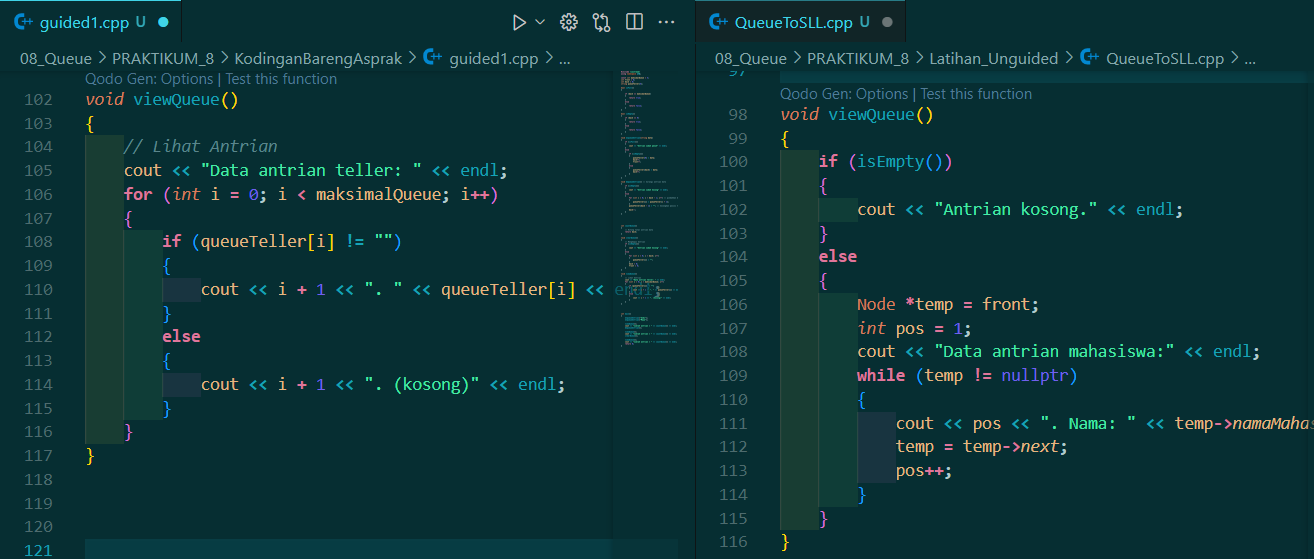
1. Counting



Pada method khusus ini kita menghitung keseluruhan data antrian yang masuk. Pada Queue biasa kita hanya melakukan return sederhana. Sedangkan pada linkedlist, kita mendeklarasikan Node sebagai data sementara yang berada di posisi depan, kemudian dibuat looping ‘while’ selama linkedlist tidak kosong, data akan dikalkulasikan dari data paling depan hingga seterusnya.

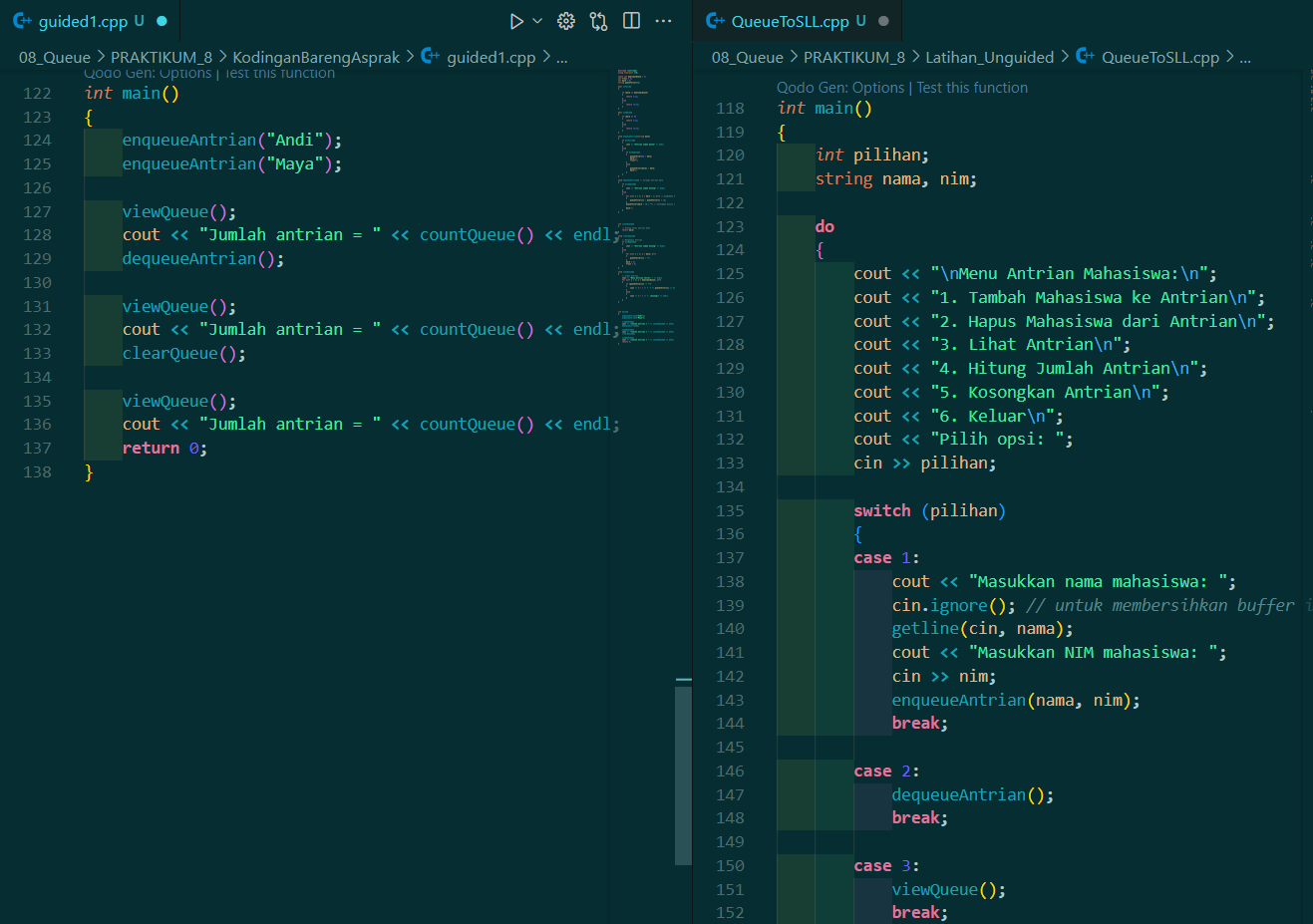
Method Clear berguna untuk menghapus seluruh data baik pada Queue maupun linkedlist. Keduanya saling menggunakan konstanta yang sesuai function untuk melakukan pembersihan data.

1. View



Kedua struktur baik itu Queue dan linkedlist sama-sama melakukan pengkondisian untuk mengecek keseluruhan data yang diiterasikan secara runtut sesuai struktur masing-masing.

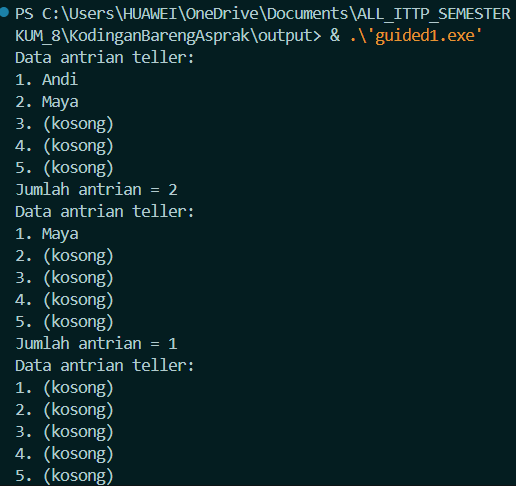
1. Main Program



Masing-masing Queue dan linkedlist berbeda jauh dalam pengoperasian programnya. Pada Queue kita melakukan inout data secara otomatis dengan method enqueue dan untuk menampilkan keseluruhan datanya menggunakan method viewQueue.

Sedangkan linkedlist, kita memasukkan inputan data secara manual yang dimana kita bisa memilih untuk Tambah, Hapus, Lihat, Hitung Total, Mengosongkan SLL, dan keluar. Disini kita buat percabangan masing-masing menu sesuai dengan function masing-masing method dengan ‘switch case’.

Output :

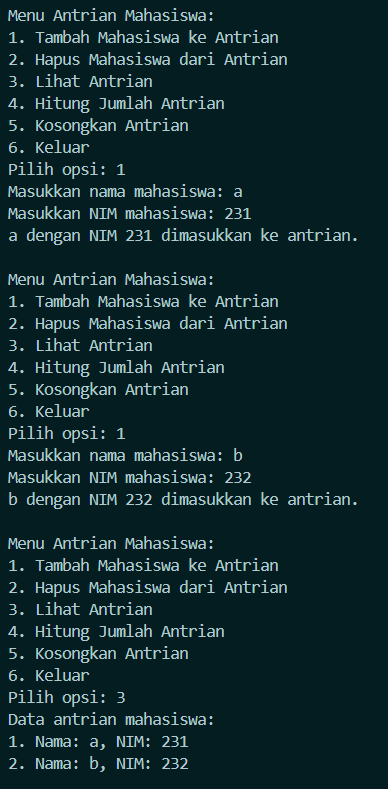


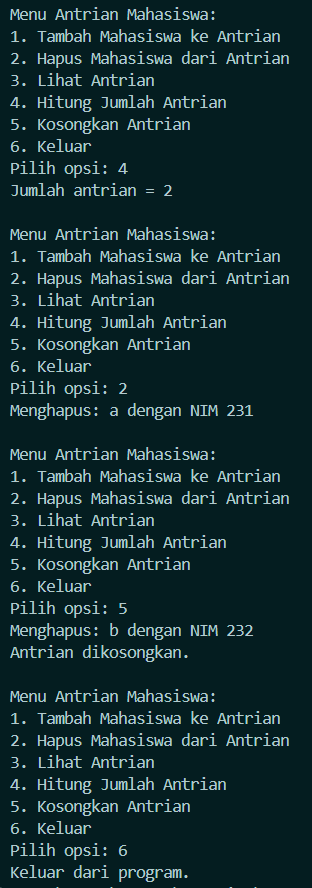
Queue

.

.

.





SLL

**Semoga Selalu diberi kemudahan^^**