Tugas Pendahuluan Modul 5

STRUKTUR DATA - Ganjil 2024/2025

"Single\_Linked\_List\_Bagian\_2"

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

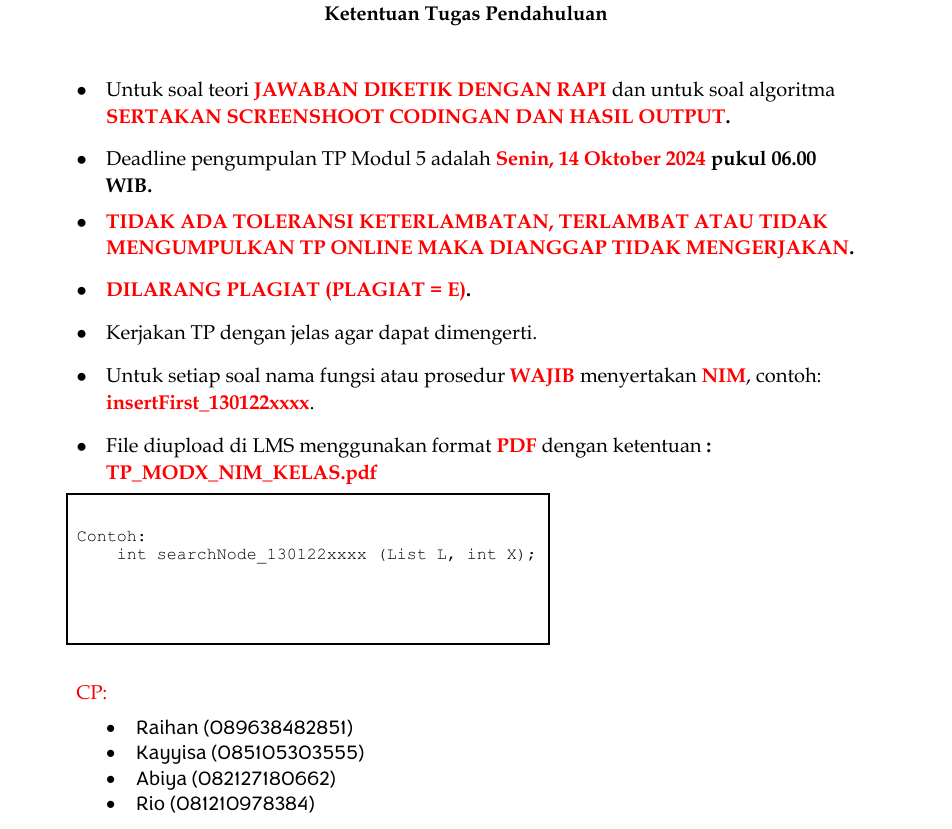
TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.

* 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 5

# 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

**TUGAS PENDAHULUAN – UNGUIDED**

1. **searchElementIdSLL (Cari nilai data tertentu dalam SLL)**

Sintaks program kali adalah untuk mencari nilai tertentu yang ingin kita cari.

Seperti yang biasa dilakukan untuk pembuatan inisial pertama pada barisan SLL beserta dengan isi data yang bertipe Node, kita buat masing-masing constructor untuk Node dan List-nya. Untuk Node perlu kita inisialisasi tipe data integer beserta pembentuk sumber utama dari struktur SLL, yaitu Node. Dalam Node ini kita perlukan deklarasikan pointer ‘next‘ guna pembacaan data dalam SLL dapat berjalan baik.

Selanjutnya buat method untuk memasukkan data yaitu ‘insertLast’ dengan parameter List &L dan pendefinisian tipe data value yang akan mengisi SLL dengan tipe data integer. Dalam method ini kita harus deklarasikan Node baru agar sumber utama untuk membentuk struktur SLL dapat dibuat dahulu. Node tadi harus diatur dengan variabel newNode agar prosesnya dapat berjalan, jangan lupa untuk mendeklarasikan data beserta pointer-nya dengan newNode agar di-set default bahwa data adalah ‘value‘ dan SLL ini masih berstatus kosong dulu.

Buat pengkondisian jika data pertama dalam SLL masih kosong maka SLL harus diinisialisasikan dengan newNode agar data-data eksternal dapat dimasukkan kedalam SLL sewaktu kita inputkan pada sintaks utamanya.

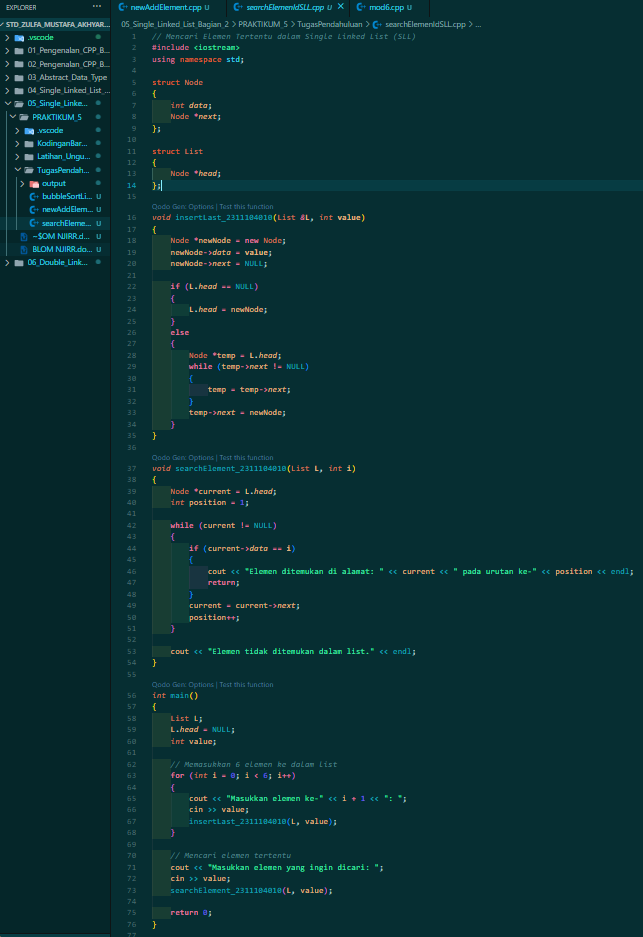
Namun jika data dalam SLL tidak kosong, maka data eksternal yang kita inputkan tadi akan diidentifikasi dengan variabel sementara ‘temp‘ dan akan terisi di posisi belakang setelah data pertama.

Method kedua adalah untuk mencari nilai yang kita inginkan, yaitu ‘searchElement‘ dengan parameter List untuk pemanggilan SLL dan deklarasi ‘i‘ dengan tipe data integer. Didalam method ini SLL harus diintegrasikan dengan fungsi \*current agar Node saat ini akan di-set sebagai data pertama dalam SLL. Nantinya juga posisi pencarian ini juga akan di-set dengan nilai 1 yang berarti default nilainya dimulai dari data pada posisi pertama atau data terdepan dalam SLL.

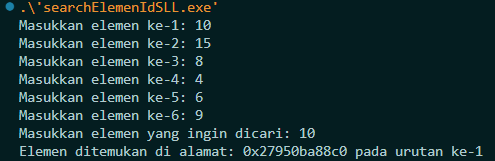
Perulangan while kali ini digunakan untuk mendeteksi keseluruhan data dari Node pertama hingga Node terakhir, artinya keseluruhan data akan di crosscheck oleh perulangan ini. Kondisikan lagi jika data yang dicari oleh fungsi current telah mendeteksi kesesuaian antara nilai ‘i‘ dengan data didalam SLL, maka program akan mengatakan kalo elemen yang kita cari ditemukan pada alamat sekian dan pada urutan sekian.

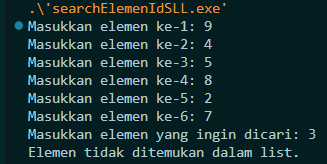
Pengkondisian tersebut akan terpenuhi juga apabila fungsi current tidak menemukan data yang sesuai dengan pencarian yang dilakukan oleh pointer ‘next‘, maka variabel position yang di-set pada nilai 1 tadi akan bertambah iterasinya sesuai jumlah nilai dalam SLL.

Namun jika sampai akhir belum menemukan nilainya, maka program menyatakan elemen gada.



Output :





1. **bubbleSortList (Pencarian SLL dengan Bubble Sort)**

Struktural pembangun SLL-nya masih dengan sintaks deklarasi yang sama. **Untuk method penambahan data maupun method untuk menampilkan data SLL mulai dari deklarasi, parameter dan pengkondisiannya masih sama seperti program yang awal**. Atau bisa dibilang untuk method penambahan data dan method menampilkan data setiap file class C++ tetap menggunakan poin-poin sintaks yang sama. Jadi tidak perlu dijelaskan panjang lebar lagi.

Namun ada method yang berbeda untuk membuat operasi urutan bubblesort, yaitu method bubbleSort. Didalamnya ada fungsi baru yaitu ‘swapped’. Fungsi tersebut adalah untuk melakukan cek iterasi dalam proses pengecekan nilai-nilai dalam SLL apakah ada pertukaran data atau tidak. Sementara kita inisiasi fungsi tersebut dengan nilai false yang menandakan bahwa pertukaran data belum terjadi. Pengkondisian didalamnya adalah proses pertukaran data depan-belakang maupun belakang-depan.

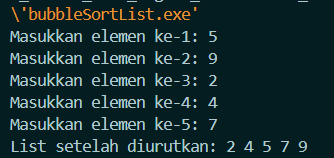
Yang membedakan hanya proses operasinya sekarang. Namun ada variabel yang harus di set default sebelum mengoperasikan SLL, yaitu ‘List L‘ dan value yang bertipe integer.

Buat perulangan sebuah inputan yang dimana berjumlah 5 dengan pemanggilan method ‘insertLast‘ supaya data dapat masuk dalam SLL.

Setelah selesai, panggil method bubbleSort untuk menukar-nukar urutan posisi nilai data dan method displayList untuk menampilkan keseluruhan SLL setelah ditukar-tukar posisinya.



Output :



1. **newAddElementIDSLL (Penambahan Nilai Data Baru)**

Kali ini yang berbeda hanya method insertSorted. Ini adalah method untuk memasukkan nilai data baru kedalam SLL. Panggil parameter ‘List &L’ dan variabel value yang bertipe integer.

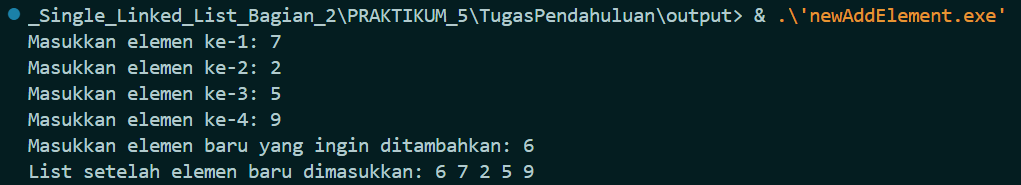
Kondisikan jika data pertama kosong dalam SLL, otomatis data baru ini bisa lebih besar atau sama dengan nilai data pertama dan akan berada diposisi pertama. Jika list tidak kosong, kondisikan selama proses pengecekan interasi data dalam SLL tidak kosong dan data yang diiterasi tidak lebih besar dari data baru, program akan terus mengiterasi keseluruhan data sampai data baru ini ada diposisi yang sesuai nilainya.

Dalam operasi programnya, buat perulangan untuk 4 buah inputan dengan method ‘insertLast’. Lalu kita bisa memasukkan nilai baru yang akan ditampung variabel value dan kita bisa gunakan variabel ‘insertSorted’.

Terakhir kita sudah bisa melihat tampilan keseluruhan nilai data SLL dengan method displayList.



Output :



**LATIHAN – UNGUIDED**

1. **Mod\_5**

Kita percepat tempo penjelasannya.

Variabel mahasiswa harus berupa constructor yang isinya Nim dan nama, kemudian untuk deklarasi SLL-nya adalah ‘Mahasiswa’ yang diberi pointer next supaya data-data baru bisa masuk kedalamnya.

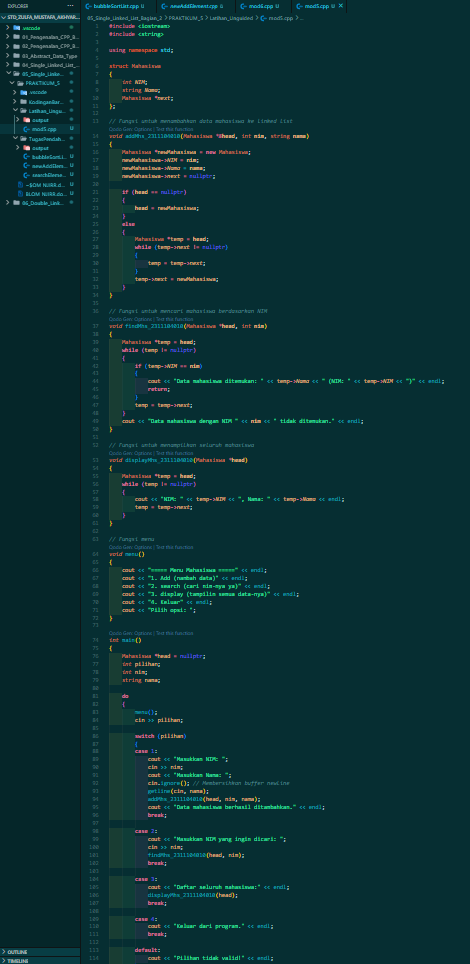
Masuk ke method untuk menambahkan data mahasiswa yaitu **addMhs**, parameterkan dulu SLL-nya dan dua variabel sebelumnya yaitu ‘nim’ dan ‘nama’. Ingat jangan lupa kita set default data pertama SLL-nya juga. Kondisikan apabila data pertama bisa terdeklarasi sebagai data baru untuk data seterusnya yang masuk ke dalam SLL.

Method berikutnya adalah **findMhs**. Method ini berfungsi untuk mencari data mahasiswa berdasarkan nim. Parameterkan data pertama dalam SLL Mahasiswa dan variabel nim agar method ini bisa mendeteksi data berupa integer dari suku data dalam SLL.

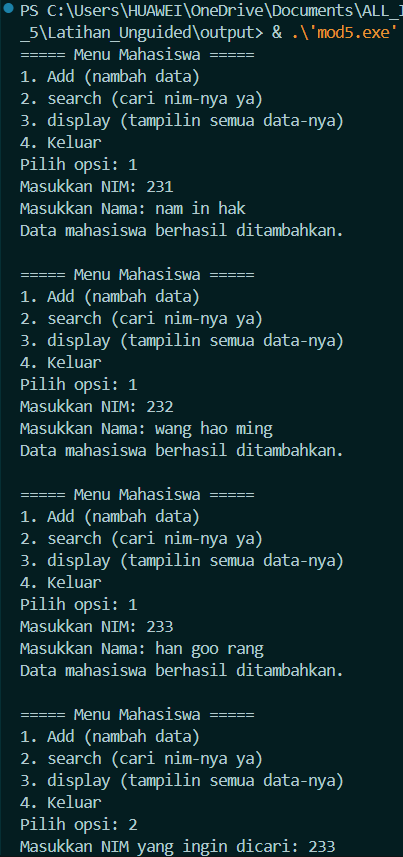
Kondisikan selama pengecekan data dimulai dari data pertama tidak kosong maka method akan terus cari nim yang sesuai dengan inputan user, namun jika masih ga ketemu, maka program menyatakan kalo nim tidak ditemukan.

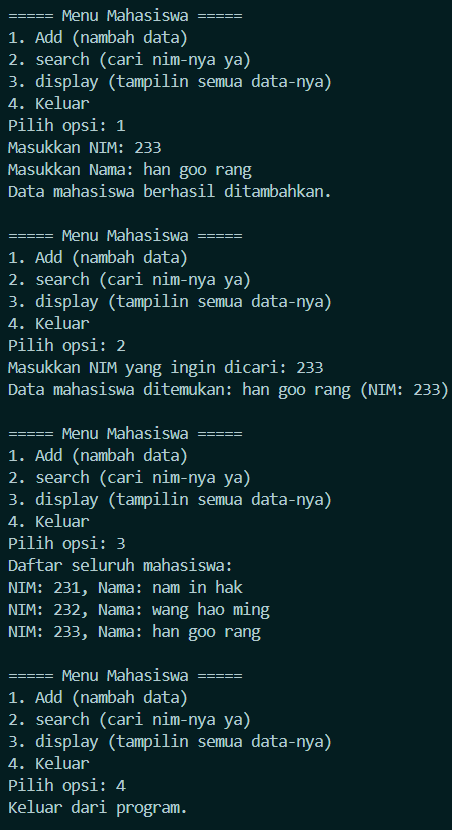
Method terakhir yaitu **displayMhs** untuk menampilkan seluruh data mahasiswa dari nim sampai nama. Pengkondisiannya sama seperti method findMhs, jadi tak perlu dijelaskan ulang.

Oh iya, jangan lupa method **menu**, tak perlu parameter apa-apa. Cukup masukkan output untuk menampilkan semacam menu untuk add, search data, display data, dan keluar.



Output :





**SOURCE CODE WAKTU DI KELAS – KODINGAN ASPRAK**





**Semoga Selalu diberi kemudahan^^**