**PERTEMUAN 5**

**STRUKTUR DATA**

**STACK**



**Disusun oleh:**

**Rama Pramudya Wibisana 2022320019**

**PROGRAM STUDI SISTEM INFORMASI**

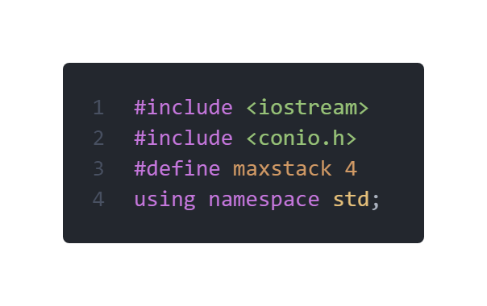
**FAKULTAS INFORMATIKA**

**UNIVERSITAS BINA INSANI**

**BEKASI**

**2022**

* **PENJELASAN SYNTAX**

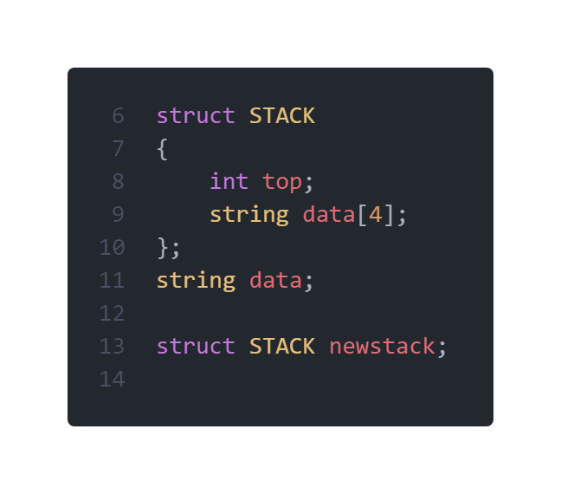
****

**#include <iostream>** berfungsi untuk mengimpor fungsi-fungsi yang sudah didefinisikan pada *header file.*

**#include <conio.h>** berfungsi untuk menampilkan hasil antarmuka kepada pengguna. Fungsi yang terdapat dalam conio.h adalah getch(), getche(), dan clrscr().

**#define maxstack 4** berfungsi untuk menentukan maksimal pada stack adalah 4 data.

**using namespace std**; berarti kita menuliskan perintah ‘c++, saya ingin gunakan semua yang ada dalam namespace std (standart), seperti cin, cout, endl, vector, string, pair, map, queue, deque, dll yang merupakan fitur standard pada library C++.’

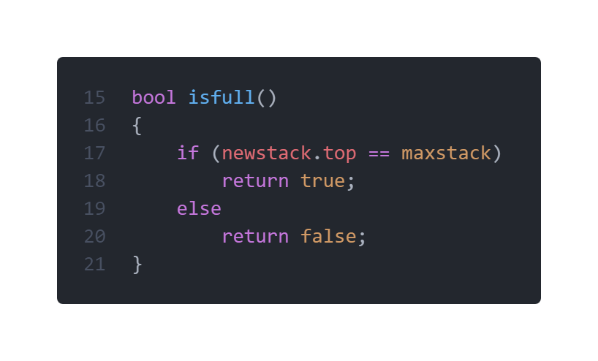
****

**Struct STACK** berarti saya ingin menggunakan tipe data **struct** yang identitasnya ialah **STACK.**

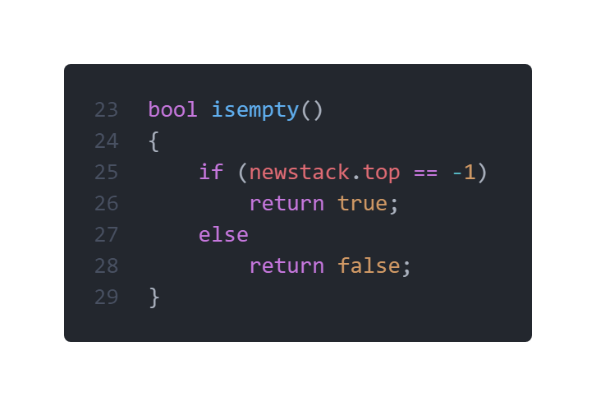
Untuk variable **top** saya menggunakan tipe data **int.**

Untuk variable **data** saya menggunakan tipe data **string** yang memiliki 4 index**.**

Struct **STACK** dijadikan suatu tipe data, dimana disebut tipe data abstrak**.**

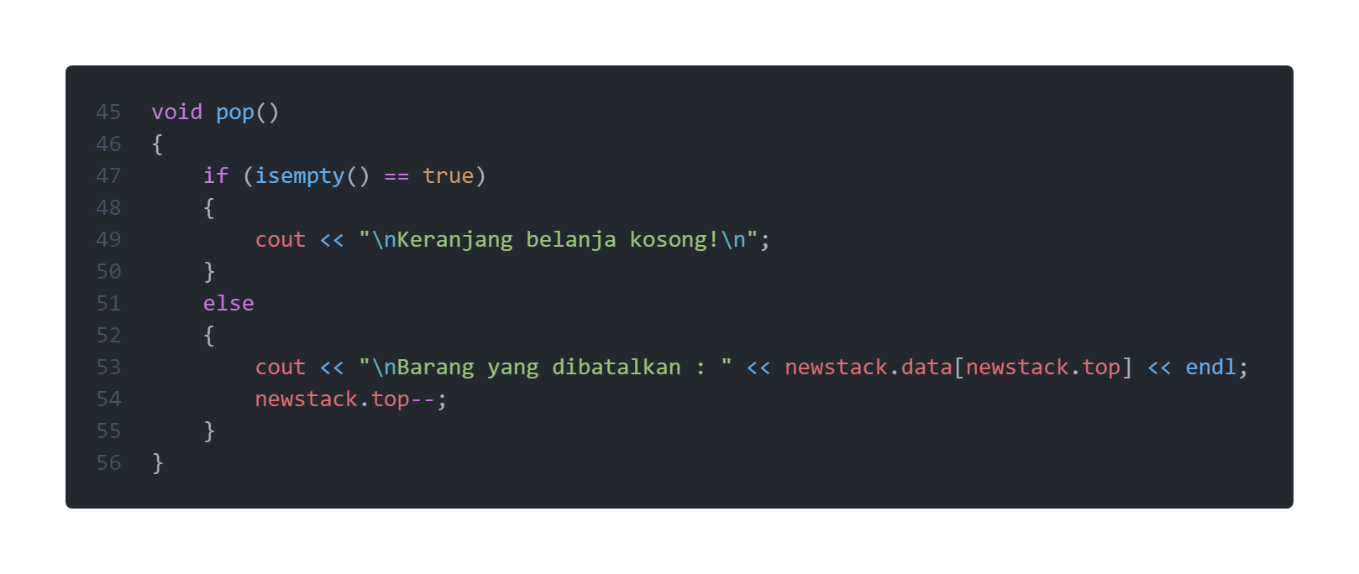
****

Berfungsi untuk mengetahui apakah suatu stack sedang penuh adalah dengan membandingkan **newstack.top** dengan **maxstack**, jika kondisi true maka stack dalam posisi penuh, dan sebaliknya.

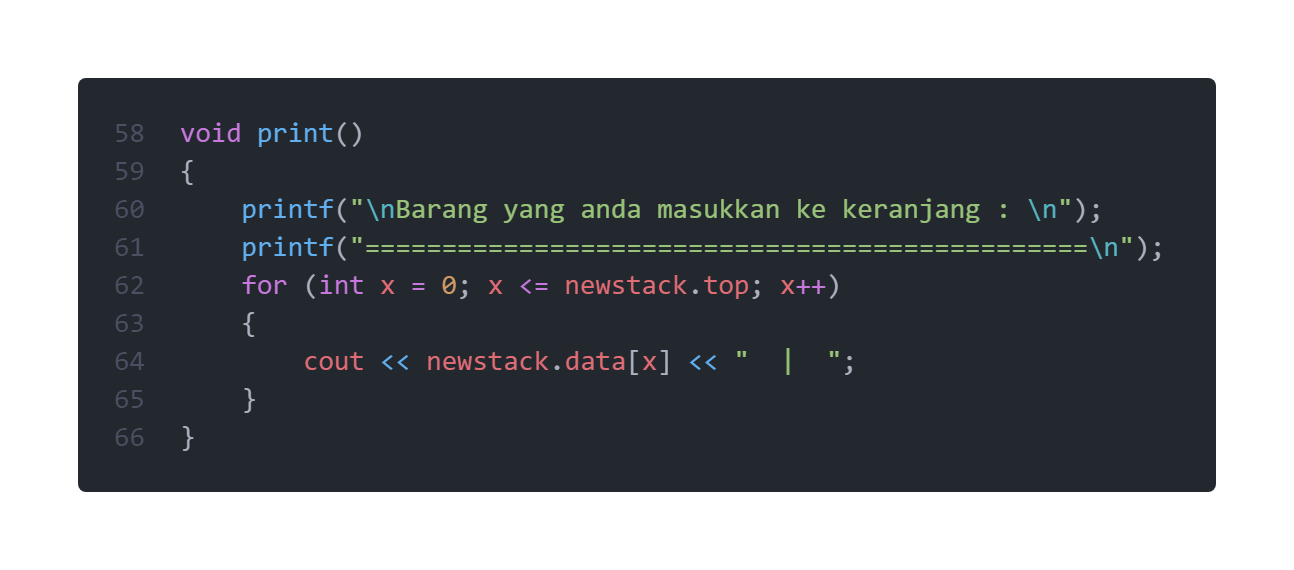


Befungsi untuk mengetahui apakah suatu stack dalam keadaan kosong adalah dengan membandingkan **newstack.top** dengan -1, jika kondisi true maka stack dalam posisi kosong, dan sebaliknya.



Pertama kali akan dicek apakah stack dalam keadaan penuh, jika true maka akan tercetak string pada layar “Keranjang belanja sudah penuh!”. Jika bernilai false maka **newstack.top** akan diincrement kemudian data yang tadi diinputkan ditambahkan pada stack.

Pertama kali yang akan dilakukan program adalah mengecek apakah stack dalam keadaan kosong, jika true maka akan tercetak string pada layar “Keranjang belanja kosong!". Jika bernilai false maka data pada posisi teratas akan diambil, dan kemudian nilai **newstack.top** didecrement sehingga posisi teratas pada stack berganti dengan data di bawah top sebelumnya.

Dengan memanfaatkan perulangan for, fungsi ini akan mencetak seluruh data yang berada di dalam stack.

Saat fungsi ini dipanggil maka posisi **newstack.top** diinisialisasi berada pada -1. Seperti halnya mereset ulang suatu stack yang membuat isinya akan hilang.



**newstack.top** diinisialisasi berada pada -1.

Variabel **menu** sebagai tipe data **char.**

Variabel **kembali** sebagai tipe data **char.**

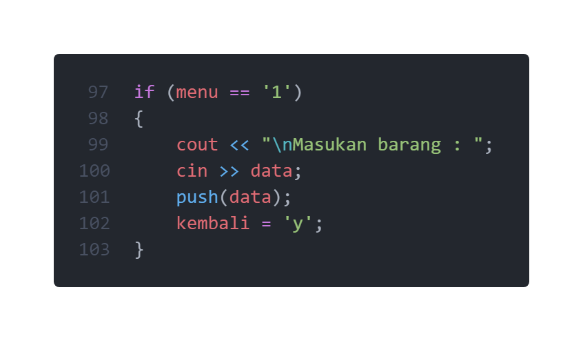


Masuk ke perulangan do-while dengan kondisi ketika

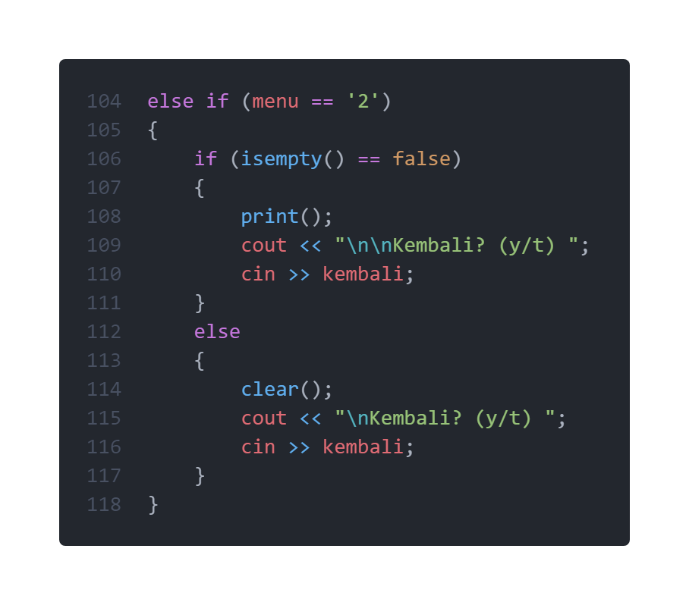
**ulang == ‘y’ || ulang ‘Y’**

Pada perulangan ini akan ditampilkan 6 menu pilihan yang dapat dipilih oleh user.

Kemudian ada input yang menggunakan variabel **menu**.

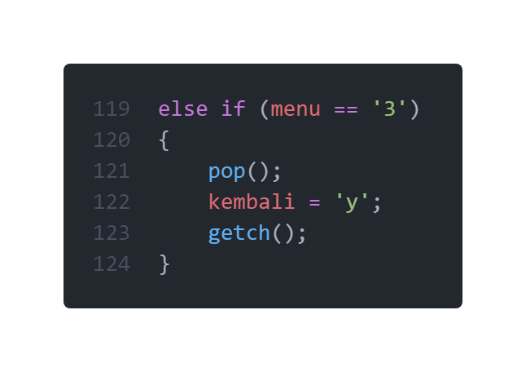


Jika input **menu == ‘1’** maka program akan menjalankan seluruh pernyataan yang ada pada kondisi pertama yaitu fungsi **push**.

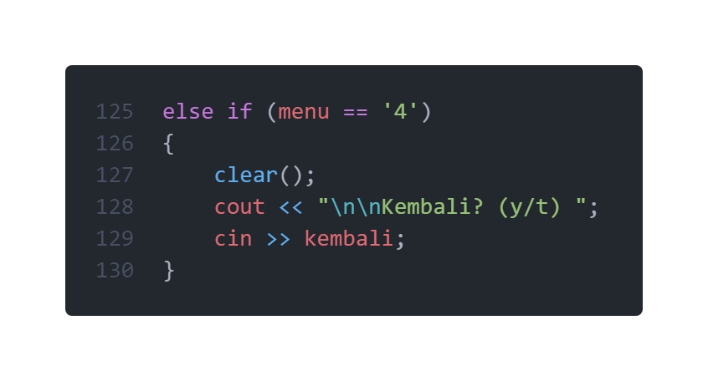


Jika input **menu == ‘2’** maka program akan menjalankan seluruh pernyataan yang ada pada kondisi kedua.

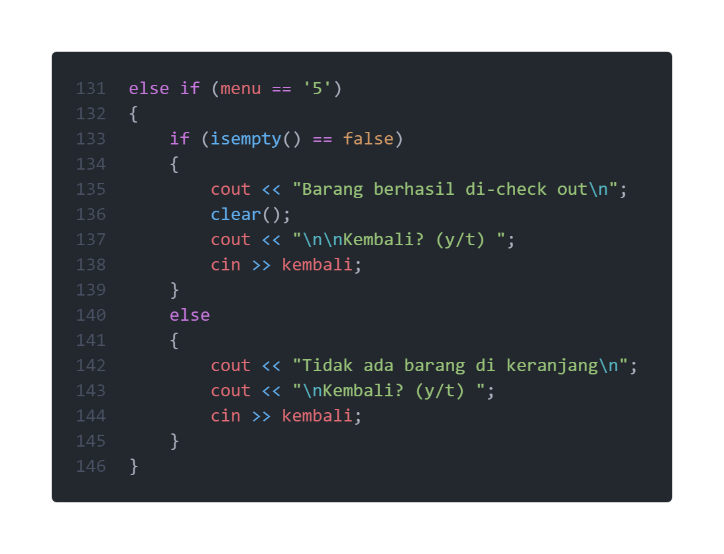
Kemudian program akan mengecek apakah stack dalam keadaan kosong, jika false maka program akan memanggil fungsi **clear()** dan akan tercetak string pada layar “Keranjang belanja anda kosong". Jika bernilai true maka program akan memanggil fungsi **print()** yang nantinya akan dicetak.



Jika input **menu == ‘3’** maka program akan menjalankan seluruh pernyataan yang ada pada kondisi ketiga.



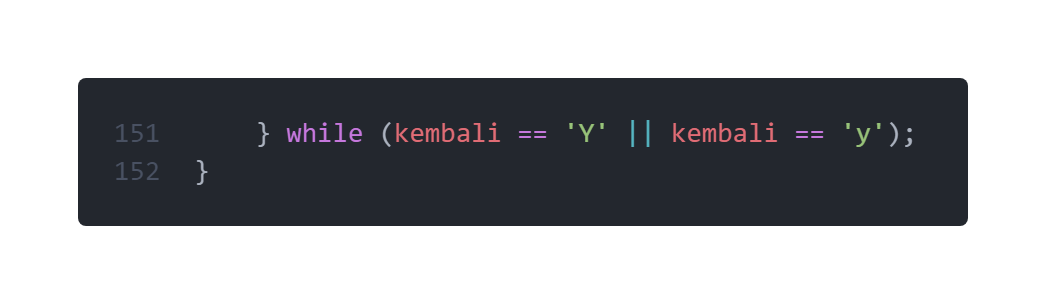
Jika input **menu == ‘4’** maka program akan menjalankan seluruh pernyataan yang ada pada kondisi keempat yaitu fungsi **pop**.



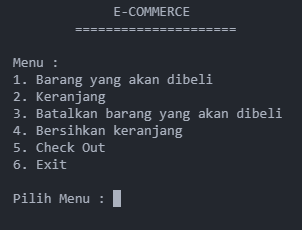
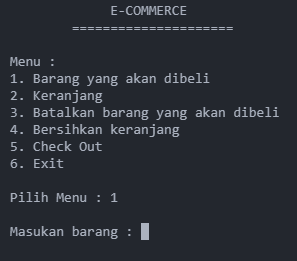
Jika input **menu == ‘5’** maka program akan menjalankan seluruh pernyataan yang ada pada kondisi kelima.

Kemudian program akan mengecek apakah stack dalam keadaan kosong, jika false maka program akan menampilkan print “Tidak ada barang di keranjang”. Jika bernilai true maka program akan menampilkan print “Barang berhasil di-check out” dan memanggil fungsi **clear()** yang nantinya akan dicetak.



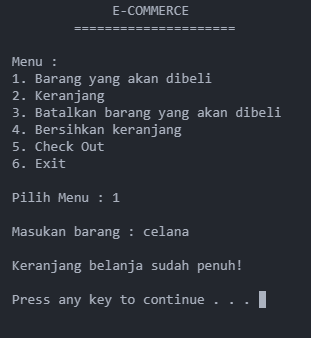
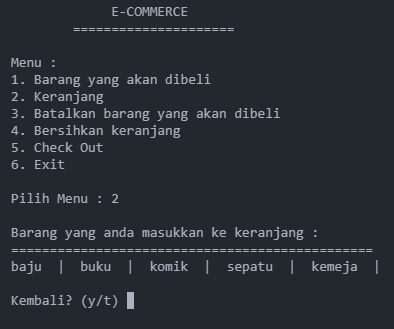
Jika input **menu == ‘6’** maka program akan menjalankan seluruh pernyataan yang ada pada kondisi keenam yaitu exit program.

* **OUTPUT SYNTAX**



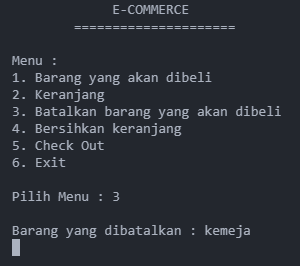
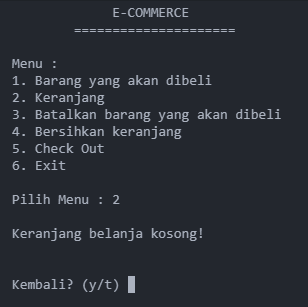
**Menu 1 push stack**

**Tampilan utama**



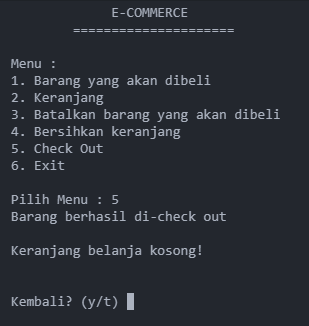
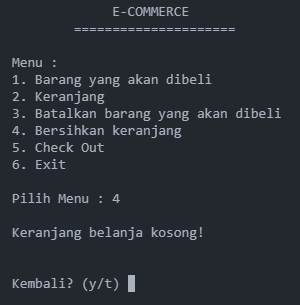
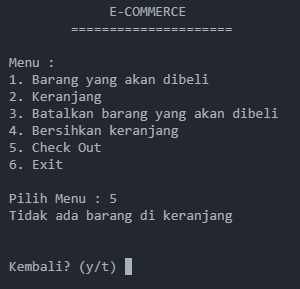
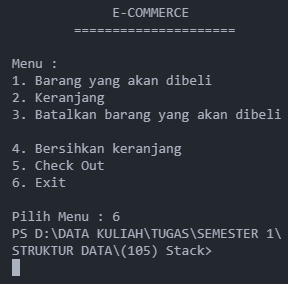
**Menu 1 apabila stack penuh**

**Menu 2 apabila stack terisi**



**Menu 2 apabila stack kosong**

**Menu 3 pop stack**



**Menu 6 exit**

**Menu 5 apabila stack kosong**

**Menu 4**

**Menu 5 apabila stack terisi**