

UJIAN AKHIR SEMESTER
ALGORITMA DAN PEMROGRAMAN



Disusun oleh:

Rama Pramudya Wibisana

2022320019

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS INFORMATIKA

UNIVERSITAS BINA INSANI

BEKASI

2023

Pada Ujian Akhir Semester kali ini, saya akan membuat program bertema E-Commerce, berikut ini penjelasan dari syntax yang saya buat.

```
1 #include <iostream>
2 #include <conio.h>
3 #define maxstack 5
4 using namespace std;
```

#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 5 berfungsi untuk menentukan maksimal pada stack adalah 5 data.

using namespace std; berarti kita menggunakan 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++.

```
6 struct STACK
7 {
8     int top;
9     string barang[5];
10 };
11 string barang;
```

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

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

Untuk variable **barang** saya menggunakan tipe data **string** yang memiliki 5 index.

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

```
15 bool isfull()
16 {
17     if (newstack.top == maxstack)
18         return true;
19     else
20         return false;
21 }
```

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.

```

23 bool isempty()
24 {
25     if (newstack.top == -1)
26         return true;
27     else
28         return false;
29 }

```

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.

```

31 void push(string barang)
32 {
33     if (isfull() == true)
34     {
35         puts("\nKeranjang belanja sudah penuh!\n");
36         system("pause");
37     }
38     else
39     {
40         newstack.top++;
41         newstack.barang[newstack.top] = barang;
42     }
43 }

```

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.

```

45 void pop()
46 {
47     if (isempty() == true)
48     {
49         cout << "\nKeranjang belanja kosong!\n";
50     }
51     else
52     {
53         cout << "\nBarang yang dibatalkan : " << newstack.barang[newstack.top] << endl;
54         newstack.top--;
55     }
56 }

```

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** di-decrement sehingga posisi teratas pada stack berganti dengan data di bawah top sebelumnya.

```

58 void print()
59 {
60     printf("\nBarang yang anda masukkan ke keranjang : \n");
61     printf("===== \n");
62     for (int x = 0; x <= newstack.top; x++)
63     {
64         cout << newstack.barang[x] << " | ";
65     }
66 }

```

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

```

68 void clear()
69 {
70     newstack.top = -1;
71     printf("\nKeranjang belanja kosong! \n");
72 }

```

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

```

74 int main()
75 {
76     newstack.top = -1;
77
78     char menu;
79     char kembali;

```

newstack.top diinisialisasi berada pada -1.

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

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

```

81  do
82  {
83      system("cls");
84      printf("UJIAN AKHIR SEMESTER\n");
85      printf("NAMA \t\t: Rama Pramudya Wibisana\n");
86      printf("NPM \t\t: 2022320019\n");
87      printf("MATA KULIAH \t: Algoritma dan Pemrograman\n\n");
88      printf("\n\t\tE-COMMERCE\n");
89      printf("\t\t===== \n\n");
90      printf("Menu : \n");
91      puts("1. Barang yang akan dibeli");
92      puts("2. Keranjang");
93      puts("3. Batalkan barang yang akan dibeli");
94      puts("4. Bersihkan keranjang");
95      puts("5. Check Out");
96      puts("6. Exit");
97
98      cout << "\nPilih Menu : ";
99      cin >> menu;

```

Masuk ke perulangan do-while
dengan kondisi ketika

```
ulang == 'y' || ulang 'Y'
```

Pertama saya akan melakukan print mengenai identitas diri saya dalam melaksanakan ujian ini.

Pada perulangan ini akan ditampilkan 6 menu pilihan yang dapat dipilih oleh user. Kemudian ada input yang menggunakan variabel **menu**.

```
101 if (menu == '1')
102 {
103     cout << "\nMasukkan barang : ";
104     cin >> barang;
105     push(barang);
106     kembali = 'y';
107 }
```

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

```

108 else if (menu == '2')
109 {
110     if (isempty() == false)
111     {
112         print();
113         cout << "\n\nKembali? (y/t) ";
114         cin >> kembali;
115     }
116     else
117     {
118         clear();
119         cout << "\n\nKembali? (y/t) ";
120         cin >> kembali;
121     }
122 }

```

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.

```

123 else if (menu == '3')
124 {
125     pop();
126     kembali = 'y';
127     getch();
128 }

```

Jika input `menu == '3'` maka program akan menjalankan seluruh pernyataan yang ada pada kondisi ketiga.

```

129 else if (menu == '4')
130 {
131     clear();
132     cout << "\n\nKembali? (y/t) ";
133     cin >> kembali;
134 }

```

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

```

135 else if (menu == '5')
136 {
137     if (isempty() == false)
138     {
139         cout << "Barang berhasil di-check out\n";
140         clear();
141         cout << "\n\nKembali? (y/t) ";
142         cin >> kembali;
143     }
144     else
145     {
146         cout << "Tidak ada barang di keranjang\n";
147         cout << "\n\nKembali? (y/t) ";
148         cin >> kembali;
149     }
150 }

```

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.

```

151 else if (menu == '6')
152 {
153     exit(0);
154 }

```

Jika input `menu == '6'` maka program akan menjalankan seluruh pernyataan yang ada pada kondisi keenam yaitu `exit` program.

```

155     } while (kembali == 'Y' || kembali == 'y');
156 }

```