

# **HANDS-ON LABS 3**

## **STRUKTUR DATA**



**Oleh:**

Martogi Jekson C. Siagian 2105541054

**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2023**

### Hands-On Labs 3

#### Tugas

1. Terdapat sebuah peti dengan volume tertentu, dimana peti ini dapat menampung beberapa kotak kayu dengan maksimal ukuran volume dari keseluruhan kotak kayu yang dimasukkan tidak melebihi dari volume peti tersebut. Mekanisme memasukkan kotak kayu ke dalam peti, hanya dapat dilakukan satu per satu dengan konsep LIFO (Last In First Out) atau Stack. Buatlah program untuk mensimulasikan proses memasukkan kotak kayu ke dalam peti tersebut, dengan ketentuan:

- Program menampilkan menu pilihan: masukkan ukuran peti, masukkan satu kotak kayu ke peti, keluarkan satu kotak kayu dari peti, kondisi peti, keluar.
- Pada pilihan menu: masukkan peti maka pengguna dapat menginputkan besar volume peti
- Pada pilihan menu: masukkan satu kotak kayu ke peti maka pengguna dapat menginputkan besar volume kotak kayu yang akan dimasukkan, jika melebihi volume peti maka tidak dapat dimasukkan
- Pada pilihan menu: keluarkan satu kotak kayu dari peti maka pengguna dapat menginputkan besar volume kotak kayu yang akan dikeluarkan, kemudian menampilkan sisa volume dari peti
- Pada pilihan menu: kondisi peti maka pengguna dapat menampilkan data seluruh kotak kayu yang ada di dalam peti
- Pada pilihan menu: keluar maka pengguna dapat keluar dari program.
- 

Jawab :

Berikut merupakan kode program yang saya buat untuk membuat program tersebut

```

tugasmod3.cpp X
1  #include <iostream>
2  #include <stack>
3  using namespace std;
4
5  int main() {
6      stack<int> peti;
7      int volume_peti = 0;
8
9      while (true) {
10         cout << "===== MENU =====" << endl;
11         cout << "1. Masukkan ukuran peti" << endl;
12         cout << "2. Masukkan satu kotak kayu ke peti" << endl;
13         cout << "3. Keluarkan satu kotak kayu dari peti" << endl;
14         cout << "4. Kondisi peti" << endl;
15         cout << "5. Keluar" << endl;
16         cout << "Pilihan: ";
17
18         int pilihan;
19         cin >> pilihan;
20
21         switch (pilihan) {
22             case 1:
23                 cout << "Masukkan volume peti maksimum: ";
24                 cin >> volume_peti;
25                 break;
26
27             case 2:
28                 if (volume_peti <= 0) {
29                     cout << "Volume peti belum diatur. Silakan atur volume peti terlebih dahulu." << endl;
30                 } else {
31                     int volume_kotak;
32                     cout << "Masukkan volume kotak yang akan dimasukkan ke peti: ";
33                     cin >> volume_kotak;
34
35                     if (volume_peti - volume_kotak >= 0) {
36                         peti.push(volume_kotak);
37                         volume_peti -= volume_kotak;
38                         cout << "Volume kotak yang dimasukkan: " << volume_kotak << " m3" << endl;
39                         cout << "Volume peti yang tersisa: " << volume_peti << " m3" << endl;
40                     } else {
41                         cout << "Volume kotak terlalu besar untuk dimasukkan ke peti." << endl;
42                     }
43                 }
44                 break;
45
46             case 3:
47                 if (!peti.empty()) {
48                     int volume_kotak = peti.top();
49                     peti.pop();
50                     volume_peti += volume_kotak;
51                     cout << "Kotak kayu yang dikeluarkan adalah kotak ke-" << peti.size() + 1 << " dengan volume " << volume_kotak << " m3" << endl;
52                     cout << "Sisa volume peti sekarang: " << volume_peti << " m3" << endl;
53                 } else {
54                     cout << "Peti sudah kosong." << endl;
55                 }
56                 break;
57
58             case 4:
59                 cout << "1) Jumlah kotak kayu pada peti besar : " << peti.size() << " kotak" << endl;
60                 if (!peti.empty()) {
61                     cout << "2) Volume masing-masing kotak kayu : " << endl;
62                     int kotak_ke = 1;
63                     stack<int> tempStack = peti;
64                     while (!tempStack.empty()) {
65                         cout << "Kotak ke- " << kotak_ke << " ber-volume: " << tempStack.top() << " m3" << endl;
66                         tempStack.pop();
67                         kotak_ke++;
68                     }
69                 } else {
70                     cout << "2) Peti kosong." << endl;
71                 }
72                 cout << "3) Volume maksimum peti besar yang tersisa: " << volume_peti << " m3" << endl;
73                 break;
74             case 5:
75                 return 0;
76             default:
77                 cout << "Pilihan tidak valid. Silakan pilih kembali." << endl;
78                 break;
79         }
80     }
81     return 0;
82 }
83
84
85
86

```

tur Data\2105541054\_Martogi Jekson C. Siagian\_Tugas Modul 3 Struktur Data\tugasmod3.cpp C/C++ | Windows (CR+LF) WINDOWS-1252 Line 72, Col 68, Pos 2955 Insert Read/Write default

Berikut Penjelasan dari kode yang telah saya buat:

Program ini menggunakan struktur data stack untuk menyimpan kotak kayu yang ada di dalam peti. Stack adalah struktur data yang dapat digunakan untuk menyimpan data secara LIFO (Last In First Out).

1. Program dimulai dengan mendeklarasikan dan menginisialisasi stack (tumpukan) yang disebut `peti` untuk merepresentasikan peti kayu dan

variabel `volume_peti` untuk menyimpan volume maksimum peti dan menghitung sisa volume yang tersedia dalam peti.

2. Program utama berjalan dalam loop `while (true)`, yang memungkinkan pengguna untuk terus menggunakan menu hingga pengguna memilih untuk keluar.
3. Terdapat lima opsi menu yang tersedia untuk pengguna:
  - Menu 1:
    - Menerima input volume maksimum peti dari pengguna.
    - Menginisialisasi variabel `volume_peti` dengan nilai input.
  - Menu 2
    - Memeriksa apakah volume peti sudah diatur.
    - Jika sudah, menerima input volume kotak kayu dari pengguna.
    - Memeriksa apakah volume kotak kayu tidak melebihi volume peti.
    - Jika tidak, menambahkan kotak kayu ke `stack_peti`.
  - Menu 3
    - Memeriksa apakah `stack_peti` kosong
    - Jika tidak, mengeluarkan kotak kayu dari `stack_peti`.
  - Menu 4
    - Menampilkan jumlah kotak kayu di dalam peti.
    - Jika `stack_peti` tidak kosong, menampilkan volume masing-masing kotak kayu.
    - Menampilkan volume maksimum peti yang tersisa.
  - Menu 5
    - Pengguna dapat keluar dari program.
4. Program melakukan validasi untuk memastikan bahwa volume peti sudah diatur sebelum pengguna mencoba memasukkan kotak kayu. Selain itu, program memeriksa apakah volume kotak kayu yang dimasukkan tidak melebihi sisa volume peti yang tersedia.
5. Saat mengeluarkan kotak kayu dari peti, program menampilkan nomor kotak yang dikeluarkan dan volume yang sesuai.

6. Pada menu 4, program menggunakan stack sementara (`tempStack`) untuk menampilkan volume masing-masing kotak kayu di dalam peti tanpa menghilangkan mereka dari peti.
7. Kode ini menggunakan struktur data stack yang dapat digunakan untuk menyimpan data kotak kayu secara efisien.
8. Kode ini memiliki menu pilihan yang memudahkan pengguna untuk berinteraksi dengan program.
9. Kode ini dilengkapi dengan laporan kondisi peti yang dapat digunakan untuk memeriksa status peti.

Berikut merupakan hasil dari program yang telah saya jalankan:

### Menu 1

```
"D:\Semester 5\Struktur Data" X + v
===== MENU =====
1. Masukkan ukuran peti
2. Masukkan satu kotak kayu ke peti
3. Keluarkan satu kotak kayu dari peti
4. Kondisi peti
5. Keluar
Pilihan: 1
Masukkan volume peti maksimum: 20|
```

### Menu 2

```
===== MENU =====
1. Masukkan ukuran peti
2. Masukkan satu kotak kayu ke peti
3. Keluarkan satu kotak kayu dari peti
4. Kondisi peti
5. Keluar
Pilihan: 2
Masukkan volume kotak yang akan dimasukkan ke peti: 15
Volume kotak yang dimasukkan: 15 m3
Volume peti yang tersisa: 5 m3
```

### Menu 3

```
===== MENU =====
1. Masukkan ukuran peti
2. Masukkan satu kotak kayu ke peti
3. Keluarkan satu kotak kayu dari peti
4. Kondisi peti
5. Keluar
Pilihan: 3
Kotak kayu yang dikeluarkan adalah kotak ke-1 dengan volume 15 m3
Sisa volume peti sekarang: 20 m3
```

### Menu 4

```
===== MENU =====
1. Masukkan ukuran peti
2. Masukkan satu kotak kayu ke peti
3. Keluarkan satu kotak kayu dari peti
4. Kondisi peti
5. Keluar
Pilihan: 4
1) Jumlah kotak kayu pada peti besar : 1 kotak
2) Volume masing-masing kotak kayu :
- kotak 1 ber-volume: 15 m3
3) Volume maksimum peti besar yang tersisa : 5 m3
```