

**Nama** : Ifham Syafwan Fikri  
**Github** : <https://github.com/FikriSyafwan/Penugasan-Praktikum-Pemrograman>  
**Nomor Mahasiswa** : 24/545184/PA/23161  
**Kelas** : KOM B  
**Dosen Pengampu** : Muhammad Husni Santriaji

---

## LAPORAN PRAKTIKUM PEMROGRAMAN

### PERTEMUAN 6

#### Number 1: Implementasi Struct & Array pada pencatatan barang

Pada praktikum pemrograman pertemuan keenam, permasalahan yang diangkat berupa pencatatan barang swalayan. Pencatatan yang dimaksud adalah pendataan barang dengan beberapa *variable*, antara lain kode barang, nama barang, harga barang, dan jumlah barang serta perhitungan jumlah barang sebanyak “n” dan harga rata-rata dari “n” jumlah barang.

Pada permasalahan ini, struct dan array digunakan untuk mempermudah pengerjaan. Struct itu apa? Struct atau *structures* merupakan struktur yang berfungsi untuk menyimpan atau memuat berbagai macam tipe data dalam satu bagian. Setiap *variable* pada struct disebut *member of structure*. Berikut merupakan program yang menghasilkan *output* atau keluaran dari permasalahan di atas:

1. struct, expected output, dan input,

```
11 #include <iostream>
12 #include <cmath>
13 using namespace std;
14
15
16 struct idBarang {
17     string kodeBarang;
18     string namaBarang;
19     float hargaBarang;
20     float jumlahBarang;
21 };
22
23 int main () {
24     int exp_output1 = 7,           //expected output
25     exp_output2 = 13000;
26
27     int n = 3;                    //input
28
29     cout << "===== " << endl;
30     cout << "    Selamat Datang di Toko XXX    " << endl;
31     cout << "===== " << endl;
32
33     cout << endl;
34 }
```

Struct akan diberikan nama “idBarang” dalam program ini dengan beberapa *member of structure* berupa

- **string** kodeBarang : kode unik untuk mempresentasikan suatu barang,
- **string** namaBarang : nama dari barang,
- **float** hargaBarang : harga barang, dan
- **float** jumlahBarang : jumlah barang.

Pada **int main()** akan diberikan *input* = 3 dan *expected output* = exp\_output1: 7 (jumlah barang) dan exp\_output2: 13.000 (harga rata-rata barang) untuk menguji apakah program akan berjalan lancar dan menghasilkan *output* atau keluaran yang sesuai dengan hipotesis. Apabila sesuai dengan expected output, program akan menghasilkan keluaran “[Test succeed!! :>]”, sedangkan sebaliknya program akan menghasilkan keluaran “[Test failed?! :<].”

## 2. input barang menggunakan array,

```
38     idBarang Barang[n];
39     Barang[0].kodeBarang = "111";
40     Barang[0].namaBarang = "esKrim";
41     Barang[0].hargaBarang = 3000;
42     Barang[0].jumlahBarang = 2;
43
44     Barang[1].kodeBarang = "112";
45     Barang[1].namaBarang = "kopiBubuk";
46     Barang[1].hargaBarang = 35000;
47     Barang[1].jumlahBarang = 1;
48
49     Barang[2].kodeBarang = "113";
50     Barang[2].namaBarang = "looseLeaf";
51     Barang[2].hargaBarang = 12500;
52     Barang[2].jumlahBarang = 4;
53
56     cout << "    Masukkan data barang! (for testing) " << endl;
57
58     for (int i = 0; i < n; i++) {
59         cout << "Data barang " << i + 1 << endl;
60         cout << "===== " << endl;
61
62         cout << "Kode barang : " <<
63         Barang[i].kodeBarang << endl;
64         cout << "Nama barang : " <<
65         Barang[i].namaBarang << endl;           //tanpa spasi
66         cout << "Harga barang : " <<
67         Barang[i].hargaBarang << endl;         //tanpa titik
68         cout << "Jumlah barang: " <<
69         Barang[i].jumlahBarang << endl;         //maksimal 50 barang
70         cout << endl;                           //untuk tes barang di-set
71         //menjadi 3.
72     }
73     cout << endl;
74
```

Berikut merupakan input dari tiga barang sebagai tes dengan menggunakan array pada program. Array itu apa? Berbeda dengan struct, array adalah kumpulan-kumpulan *variable* yang menyimpan data dengan tipe yang sama. Indeks dari elemen array dimulai dari angka 0, bukan angka 1. Dalam array, setiap *variable* dapat dibedakan dengan *variable* lain berdasarkan *subscript*. Bilangan dalam kurung siku [...] disebut *subscript* dengan masing-masing elemen dapat diakses.

Array pada program akan dinamakan sebagai “Barang” dan dihubungkan dengan struct idBarang menjadikan setiap elemen pada Barang[i] memiliki *variable* dari struct idBarang. Untuk mengubah nilai dari masing-masing *variable*, dot syntax (.) diperlukan, sebagai contoh Barang[0]. kodeBarang pada program akan diberi string “111”.

Setelah memberikan nilai pada masing-masing *variable* untuk tiap elemen Barang[i], program akan mengeluarkan data dari input yang telah disimpan.

### 3. dan perhitungan jumlah dan harga rata-rata barang

```
90
91     cout << "-Jumlah dan harga rata-rata barang-" << endl;
92     float p, q, r;
93     for (int i = 0; i < n; i++) {
94         p += Barang[i].jumlahBarang;
95         q += Barang[i].hargaBarang * Barang[i].jumlahBarang;
96         r = q / p;
97     }
98
99     cout << " <Jumlah barang> " << endl;
100    cout << "> " << p << " <" << endl;
101
102    cout << endl;
103
104    cout << " <Harga rata-rata barang> " << endl;
105    cout << "> " << static_cast<int>(round(r)) << " <" << endl;
106
107    if (p == exp_output1 && q / p == exp_output2) {
108        cout << "[Test succeed!! :>]" << endl;
109    }
110    else {
111        cout << "[Test failed?! :<]" << endl;
112    }
113
114
115    return 0;
116 }
```

Untuk menghitung jumlah barang tiap elemen dari array Barang[i], tipe perulangan for akan digunakan sebanyak tiga kali, nilai i akan dimulai dari 0 karena indeks elemen dari array dimulai dari 0. Pada program ini juga, untuk memudahkan perhitungan, *variable* float p, q, dan r akan diberikan untuk menyimpan nilai. *Variable* p sebagai jumlah dari semua barang, q sebagai jumlah dari semua barang dikali harga barang, dan float r sebagai harga rata-rata dari barang yang telah di-*input* sebelumnya.

Kemudian, pembulatan akan diperlukan untuk menyamakan hasil dengan expected output. Pembulatan ditandai dengan `static_cast<int>(round(r))` menyebabkan `#include <cmath>` diperlukan.

Berikut merupakan hasil dari program yang telah diberi *input*:

```
./number_1_test
=====
    Selamat Datang di Toko XXX
=====

Masukkan banyak barang = 3 (for testing)

    Masukkan data barang! (for testing)
Data barang 1
=====
Kode barang : 111
Nama barang : esKrim
Harga barang : 3000
Jumlah barang: 2

Data barang 2
=====
Kode barang : 112
Nama barang : kopiBubuk
Harga barang : 35000
Jumlah barang: 1

Data barang 3
=====
Kode barang : 113
Nama barang : looseLeaf
Harga barang : 12500
Jumlah barang: 4

    -Data barang-
Data barang 1
=====
Kode barang : 111
Nama barang : esKrim
Harga barang : 3000
Jumlah barang: 2

Data barang 2
=====
Kode barang : 112
Nama barang : kopiBubuk
Harga barang : 35000
Jumlah barang: 1

Data barang 3
=====
Kode barang : 113
Nama barang : looseLeaf
Harga barang : 12500
Jumlah barang: 4

    -Jumlah dan harga rata-rata barang-
    <Jumlah barang>
    > 7 <

    <Harga rata-rata barang>
    > 13000 <
[Test succeed!! :>]
Testing complete!° ✧ °
```