

## **PRAKTIKUM 5**

### *Record/Structure*

Tujuan Praktikum :

Mempraktekkan konsep *record/structure* pada bahasa pemrograman C++

**Record/Structure** adalah suatu tipe data yang merupakan kumpulan dari atribut-atribut (field) suatu objek. Pada record tipe elemen bisa berbeda-beda tidak seperti array yang mengharuskan mempunyai tipe elemen yang sama.

**Struktur** (*Structure*) C++ adalah koleksi variabel dibawah sebuah nama, variabel – variabel ini dapat

berbentuk berbagai type, yaitu sebagai berikut :

- Int
- Float
- Char
- Dan lain-lain

Perbedaan utama antara struktur dan array adalah bahwa dalam array memiliki tipe data yang sama, sedangkan struktur adalah sebuah koleksi dari variabel – variabel dibawah nama yang sama, dimana setiap elemen dapat saja memiliki tipe yang berbeda.

#### **Percobaan 5.1: Deklarasi Record/Structure 1**

Struktur dalam C++ dideklarasikan menggunakan keyword **struct** diikuti dengan nama struktur atau sering disebut dengan tag. Variabel – variabel struktur dideklarasikan dalam kurung kurawal { }, setiap elemennya dipisahkan dengan tanda titik koma atau semi colom.

```
struct namaRecord {  
    tipe_atribut1 atribut1  
    tipe_atribut2 atribut2  
    tipe_atribut3 atribut3  
    .....  
}
```

Contoh:

```
struct mahasiswa { //deklarasi struktur  
    string nama;    //atribut/member  
    int nim;        //atribut/member  
    double nilai;   //atribut/member  
} mhs;             //variabel struktur
```

Cara mengakses variabel di dalam record/struct adalah dengan operator dot (.). Misalkan terdapat nama record/struct mhs, dan variabel yang akan diakses di dalamnya adalah nim, maka cara mengaksesnya adalah “ mhs.nim “yang artinya kita mengakses nim yang merupakan satu atribut dari data mhs.

```
#include <iostream>
#include <string>
using namespace std;

main() {
    struct car {
        string brand;
        string model;
        int year;
    } myCar1, myCar2;

    myCar1.brand = "BMW";
    myCar1.model = "X5";
    myCar1.year = 1999;

    myCar2.brand = "Ford";
    myCar2.model = "Mustang";
    myCar2.year = 1969;

    cout <<"Car 1 : "<<endl;
    cout <<"\tBrand : "<<myCar1.brand<<endl;
    cout <<"\tModel : "<<myCar1.model<<endl;
    cout <<"\tYear   : "<<myCar1.year<<endl;
    cout <<"Car 2 : "<<endl;
    cout <<"\tBrand : "<<myCar2.brand<<endl;
    cout <<"\tModel : "<<myCar2.model<<endl;
    cout <<"\tYear   : "<<myCar2.year<<endl;
}
```



## Percobaan 5.2: Deklarasi Record/Structure 2

```
#include<iostream>
using namespace std;

main(){
    struct data_tanggal{
        int tanggal;
        int bulan;
        int tahun;
    };

    data_tanggal tgl1, tgl2;

    cout<<"Input Data Tanggal 1: "<<endl;
    cout<<"\tTanggal : "; cin>>tgl1.tanggal; cout<<endl;
    cout<<"\tBulan   : ";cin>>tgl1.bulan;cout<<endl;
    cout<<"\tTahun    : ";cin>>tgl1.tahun;cout<<endl;

    tgl2.tanggal = tgl1.tanggal;
    tgl2.bulan = tgl1.bulan;
    tgl2.tahun = tgl1.tahun;

    cout<<"Tanggal 1: "<<tgl1.tanggal<<"/"<<tgl1.bulan<<"/"<<tgl1.tahun<<endl;
    cout<<"Tanggal 2: "<<tgl2.tanggal<<"/"<<tgl2.bulan<<"/"<<tgl2.tahun<<endl;
}
```

**Percobaan 5.3: Deklarasi Record/Structure 3**

```
#include <iostream>
#include <cstring>
using namespace std;

struct books {
    char title[50];
    char author[50];
    char subject[100];
    int book_id;
};

main() {
    cout<<"Program Daftar Nilai Mahasiswa"<<endl;
    cout<<"-----"<<endl;

    struct books book[5]; //dapat juga menggunakan tipe array

    strcpy( book[1].title, "Learn C++ Programming");
    strcpy( book[1].author, "Chand Miyan");
    strcpy( book[1].subject, "C++ Programming");
    book[1].book_id = 6495407;

    book[2] = book[1];

    cout<<"Buku ke-1 :"<<endl;
    cout << "\tTitle   : " << book[1].title <<endl;
    cout << "\tAuthor   : " << book[1].author <<endl;
    cout << "\tSubject  : " << book[1].subject <<endl;
    cout << "\tID       : " << book[1].book_id <<endl;

    cout<<"Buku ke-2 :"<<endl;
    cout << "\tTitle   : " << book[2].title <<endl;
    cout << "\tAuthor   : " << book[2].author <<endl;
    cout << "\tSubject  : " << book[2].subject <<endl;
    cout << "\tID       : " << book[2].book_id <<endl;
}
```

**Percobaan 5.4: Nested Record/Structure**

```
#include<iostream>
using namespace std;

main() {
    struct Tgl_Lahir
    {
        int Tanggal;
        int Bulan;
        int Tahun;
    };

    struct Mahasiswa
    {
        char Nim[9];
        char Nama[25];
        Tgl_Lahir Lahir;
    };

    Mahasiswa Mhs;
    cout<<"NIM   : "; cin.getline(Mhs.Nim,9);
    cout<<"Nama   : "; cin.getline(Mhs.Nama,25);
    cout<<"Tanggal Lahir   :\n";
    cout<<"\tTanggal   : "; cin>>Mhs.Lahir.Tanggal;
    cout<<"\tBulan    : "; cin>>Mhs.Lahir.Bulan;
    cout<<"\tTahun     : "; cin>>Mhs.Lahir.Tahun;
    cout<<"\n\nMencetak Kembali Nilai Anggota\n\n";
    cout<<"NIM   : "<<Mhs.Nim;
    cout<<"\nNama   : "<<Mhs.Nama;
    cout<<"\nTanggal Lahir   : "<<Mhs.Lahir.Tanggal<<"-";
    cout<<Mhs.Lahir.Bulan<<"- "<<Mhs.Lahir.Tahun;

}
```

### Sumber / Referensi :

- [1] R. Munir, *Algoritma dan Pemrograman*, Jilid 1, Bandung: Penerbit Informatika, 2012.
- [2] N. Wirth, *Algorithms + Data Structures = Programs*, India: Prentice-Hall Of India Pvt. Limited, 1990.
- [3] J. R. Hubbard, *Programming With C++, Schaum's outlines Series*, USA: McGraw Hill, 1996.
- [4] M. Suarga, M.Math., *Algoritma Dan Pemrograman* (Edisi 2), Yogyakarta: Penerbit Andi, 2012.
- [5] M. Shalahuddin and A. S. Rosa, *Belajar Bahasa Pemrograman dengan C++ dan Java*, Penerbit Informatika, 2007.
- [6] J. Adams, S. Leestma, and L. Nyhoof, *C++ An Introduction To Computing*, Prentice-Hall, Inc., 1995.
- [7] J. P. Cohoon and J. W. Davidson, *C++ Programming Design*, McGraw-Hill, 1997.
- [8] B. Raharjo, *Mengungkap Rahasia Pemrograman Dalam C++*, Penerbit Informatika, 2004.
- [9] A. Kadir, *Pemrograman C++*, Andi, Yogyakarta, 2003.
- [10] Y. O. Suheru, *Trik Memecahkan Masalah Dengan Tiga bahasa Pemrograman – C++, Pascal dan Visual Basic*, Gava Media, Yogyakarta, 2004.