

MODUL 4 : STRUCT, UNION, DAN TYPEDEF

LAB 4.1: TUJUAN PRAKTIKUM

Learning outcome untuk praktikum pemrograman dasar pada modul 4 ini adalah bagian dari *LO4* khususnya mengenai structure, union, dan typedef

LO4 Mahasiswa mampu menggunakan dan menganalisis array, structure, union dan tipe data dinamis dalam pemrograman

LAB 4.2: DASAR TEORI

STRUCTURE

Structure adalah kumpulan variabel yang direferensikan dalam satu nama, sebagai cara yang nyaman untuk menyimpan informasi yang terkait secara bersama-sama. Structure disebut sebagai tipe data tidak terstruktur atau tipe data majemuk (*compound data types*) yang terdiri dari beberapa variabel yang memiliki tipe data yang berbeda, yang belum terhubung secara logis (belum terstruktur). Variables yang ada di dalam structure disebut sebagai **member**, **element**, atau **fields**

Cara mendefinisikan structure adalah sebagai berikut :

```
struct nama_struct {  
  
    tipe data structure_member1;  
  
    tipe data structure_member2;  
  
    ...  
  
    tipe data structure_memberN;  
};
```

Cara mendeklarasikan variabel bertipe data structure

Nama_struct nama_variabel;

```
struct nama_struct {  
  
    tipe data structure_member1;  
  
    tipe data structure_member2;  
  
    ...  
  
    tipe data structure_memberN;  
} instance_1, instance_2, instance_N;
```

instance_1, instance_2 instance_n;

Contoh : Membuat structure tentang mahasiswa

```
struct mahasiswa{  
  
    char NIM[20];  
  
    char nama[40];  
  
    int angkatan;
```

```
struct mahasiswa{  
  
    char NIM[20];  
  
    char nama[40];  
  
    int angkatan;
```

Cara mendeklarasikan variabel bertipe data structure

Mahasiswa mhs;

mhs1, mhs2, mhs3;

Ukuran structure ditentukan dari jumlah ukuran byte dari elemen-elemennya. Karena tiap elemen memiliki jenis tipe data sendiri maka tinggal dijumlahkan ukuran byte masing-masing elemen.



Ukuran variabel mhs = 20+40+4 = 64 byte

Cara mengakses elemen dari structure

Member selection operator (.) digunakan untuk mengakses secara individual elemen dari structure. Caranya dengan meletakkan operator di antara nama variable dan nama elemen.

Contohnya :

mhs.NIM

mhs.nama

mhs.angkatan

Operasi pada structure

1. Memberi nilai pada elemen structure

```
cout << "Masukkan NIM : \n";  
cin.get(mhs.NIM,20);  
cout << "Masukkan Nama: \n";  
cin.get(mhs.nama,40);  
cout << "Masukkan angkatanr salary: \n";  
cin >> mhs.angkatan;
```

2. Menampilkan nilai pada elemen structure

```
cout << "NIM: " << mhs.NIM << endl;
cout << "Nama: " << mhs.nama << endl;
cout << "Angkatan: " << mhs.angkatan << endl;
```

UNION

Union mirip dengan structure, merupakan kumpulan variabel yang memiliki tipe data yang berbeda, yang direferensikan dalam satu nama, sebagai cara yang nyaman untuk menyimpan informasi yang terkait secara bersama-sama. Bedanya dengan structure adalah ukuran byte-nya. Ukuran Union didasarkan kepada ukuran byte yang paling besar dari tipe data elemennya.

Cara mendefinisikan union adalah sebagai berikut :

```
union nama_union {
    tipe data union_member1;
    tipe data union_member2;
    ...
    tipe data union_memberN;
};
```

Cara mendeklarasikan variabel bertipe data structure

Nama_union nama_variabel;

Contoh : Membuat union tentang mahasiswa

```
union mahasiswa{
    char NIM[20];
    char nama[40];
    int angkatan;
};
```

```
} mhs1,mhs2, mhs3;
```

mahasiswa mhs ;

```
struct nama_union {
    tipe data union_member1;
    tipe data union_member2;
    ...
    tipe data union_memberN;
} instance_1,instance_2, instance_N;
```

instance_1,instance_2 instance_n;

```
union mahasiswa{
    char NIM[20];
    char nama[40];
    int angkatan;
} mhs1,mhs2, mhs3;
```

```
} mhs1,mhs2, mhs3;
```

mhs1, mhs2, mhs3;

Ukuran union ditentukan oleh ukuran byte terbesar dari elemen-elemennya. Karena tiap elemen memiliki jenis tipe data sendiri maka tinggal dicari mana ukuran byte yang paling besar.

 Char
nama[40]

Ukuran variabel mhs = 40 = 40 byte

Cara mengakses elemen dari union

Member selection operator (.) digunakan untuk mengakses secara individual elemen dari union.

Caranya dengan meletakkan operator di antara nama variable dan nama elemen.

Contohnya :

```
mhs.NIM  
mhs.nama  
mhs.angkatan
```

Operasi pada union

1. Memberi nilai pada elemen union

```
cout << "Masukkan NIM : \n";  
cin.get(mhs.NIM,20);  
cout << "Masukkan Nama: \n";  
cin.get(mhs.nama,40);  
cout << "Masukkan angkatan salary: \n";  
cin >> mhs.angkatan;
```

2. Menampilkan nilai pada elemen union

```
cout << "NIM: " << mhs.NIM << endl;  
cout << "Nama: " << mhs.nama << endl;  
cout << "Angkatan: " << mhs.angkatan << endl;
```

TYPEDEF

Semua contoh program structure di bagian sebelumnya dapat disederhanakan menggunakan typedef. Kata kunci typedef dapat digunakan untuk membuat sinonim tipe data atau untuk menyederhanakan penamaan dan penggunaan structure.

Contoh : Membuat typedef structure tentang mahasiswa

```
typedef struct mahasiswa{  
    char NIM[20];  
    char nama[40];  
    int angkatan;  
};
```

```
} mhs1, mhs2, mhs3;
```

```
typedef struct mahasiswa{  
    char NIM[20];  
    char nama[40];  
    int angkatan;  
} mhs;
```

```
} mhs1, mhs2, mhs3;
```

mahasiswa student;

mhs student;

LAB 4.3: LATIHAN

PERCOBAAN 1 : STRUCTURE

Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut.

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

struct pegawai
{
    char nama[20];
    int id;
    float gaji;
};

int main()
{
    pegawai pgw;
    cout << "Masukkan Nama : \n";
    cin.get (pgw.nama,20);
    cout << "Masukkan id Pegawai : \n";
    cin>>pgw.id;
    cout << "Masukkan gaji : \n";
    cin >> pgw.gaji;

    cout << "\nTampilkan Information Pegawai" << endl;
    cout << "Nama: " << pgw.nama << endl;
    cout << "ID: " << pgw.id << endl;
    cout << "Gaji: " << pgw.gaji;

    return 0;
}
```

1. Ada berapa elemen yang ada pada structure pegawai ?
2. Informasi apakah yang dikeluarkan program di atas ?
3. Modifikasilah program di atas untuk menambahkan 2 elemen structure, yaitu alamat dan umur pegawai
4. Modifikasi program sehingga keluarannya sesuai dengan jumlah elemen yang sudah dimodifikasi sebelumnya

PERCOBAAN 2 : STRUCTURE

Berikut ini adalah sebuah program dalam bahasa C++. Salinlah program berikut menggunakan

text editor dan simpan dengan nama `binary.cpp`. Kemudian *compile* dan *run* (eksekusi) program tersebut. Contoh di bawah ini hanya terdiri dari sebuah fungsi, yaitu fungsi `main`.

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

struct Produk
{
    int kode;
    float harga;
} peg1, peg2;

int main()
{
    cout << "Masukkan Produk Pertama : \n";
    cout << "Masukkan kode Produk : \n";
    cin >> peg1.kode;
    cout << "Masukkan harga : \n";
    cin >> peg1.harga;

    cout << "Masukkan Produk Kedua: \n";
    cout << "Masukkan kode Produk : \n";
    cin >> peg2.kode;
    cout << "Masukkan harga : \n";
    cin >> peg2.harga;

    cout << "\nTampilkan Information Produk Pertama" << endl;

    cout << "Kode: " << peg1.kode << endl;
    cout << "Harga: " << peg1.harga;

    cout << "\nTampilkan Information Produk Kedua" << endl;
    cout << "Kode: " << peg2.kode << endl;
    cout << "Harga: " << peg2.harga;

    return 0;
}
```

1. Ada berapa produk yang dikeluarkan program di atas ?
2. Modifikasilah program di atas untuk menambahkan dan mengambil 2 jenis Produk lain

PERCOBAAN 3 : STRUCTURE

Berikut ini adalah sebuah program dalam bahasa C++. Salinlah program berikut menggunakan text editor dan simpan dengan nama `binary.cpp`. Kemudian *compile* dan *run* (eksekusi) program tersebut. Contoh di bawah ini hanya terdiri dari sebuah fungsi, yaitu fungsi `main`.

```
#include <iostream>
#include <cstring>

using namespace std;

struct Books {
```

```
char title[50];
char author[50];
char subject[100];
int book_id;
};

int main() {
    struct Books Book1;    // Declare Book1 of type Book
    struct Books Book2;    // Declare Book2 of type Book

    // book 1 specification
    strcpy( Book1.title, "Learn C++ Programming");
    strcpy( Book1.author, "Chand Miyan");
    strcpy( Book1.subject, "C++ Programming");
    Book1.book_id = 6495407;

    // book 2 specification
    strcpy( Book2.title, "Telecom Billing");
    strcpy( Book2.author, "Yakit Singha");
    strcpy( Book2.subject, "Telecom");
    Book2.book_id = 6495700;

    // Print Book1 info
    cout << "Book 1 title : " << Book1.title << endl;
    cout << "Book 1 author : " << Book1.author << endl;
    cout << "Book 1 subject : " << Book1.subject << endl;
    cout << "Book 1 id : " << Book1.book_id << endl;

    // Print Book2 info
    cout << "Book 2 title : " << Book2.title << endl;
    cout << "Book 2 author : " << Book2.author << endl;
    cout << "Book 2 subject : " << Book2.subject << endl;
    cout << "Book 2 id : " << Book2.book_id << endl;

    return 0;
}
```

1. Ada berapa buku yang dikeluarkan program di atas ?
2. Modifikasilah program di atas untuk menambahkan 2 elemen structure dan menampilkannya

PERCOBAAN 4 : STRUCTURE

Pada percobaan ini, salinlah program berikut menggunakan *text editor* dan simpan masing-masing dengan nama *strukturmhscpp*. Kemudian *compile* dan eksekusi program tersebut.

```
#include <iostream>
using namespace std;
struct Mahasiswa
{
    char nama[40];
    int angkatan;
};
```

```
int main()
{
    mhs;
    cout<<"Masukkan Nama Mahasiswa :";
    cin.get(nama,40);
    cout<<"Masukkan Angkatan :";
    cin>>angkatan;

    cout<<"Data Mahasiswa"<<endl;
    cout<<"Nama : "<<mhs.nama<<endl;
    cout<<"Angkatan : "<<mhs.angkatan<<endl;

    return 0;
}
```

1. Apakah terdapat kesalahan pada program tersebut? Tunjukkan kesalahannya dan jelaskan
2. Bila terdapat kesalahan, cobalah perbaiki program tersebut hingga dapat di-*compile* dan memberikan hasil.

PERCOBAAN 5 : UNION

Program di bawah ini adalah program yang memanfaatkan union. Salin, *compile*, dan eksekusi program di berikut ini.

```
#include <iostream>
using namespace std;
union Mahasiswa
{
    char nama[40];
    int angkatan;
};

int main()
{
    Mahasiswa mhs;
    cout<<"Masukkan Nama Mahasiswa :";
    cin.get(mhs.nama,40);
    cout<<"Masukkan Angkatan :";
    cin>>mhs.angkatan;

    cout<<"Data Mahasiswa"<<endl;
    cout<<"Nama : "<<mhs.nama<<endl;
    cout<<"Angkatan : "<<mhs.angkatan<<endl;

    cout<<"Alokasi memori untuk elemen nama : " <<&mhs.nama<<endl;
    cout<<"Alokasi memori untuk elemen angkatan : " <<&mhs.angkatan<<endl;

    cout<<"Ukuran dari variabel union : " << sizeof(mhs)<<endl;

    return 0;
}
```


1. Apakah tujuan dari program di atas? Jelaskan

PERCOBAAN 6 : TYPEDEF

Program di bawah ini adalah program yang memanfaatkan pointer. Salin, *compile*, dan eksekusi program di berikut ini.

```
#include <iostream>
using namespace std;
typedef struct mahasiswa
{
    char nama[40];
    int angkatan;
}mhs;

int main()
{
    mhs aku;
    cout<<"Masukkan Nama Mahasiswa :";
    cin.get(aku.nama,40);
    cout<<"Masukkan Angkatan : "<<endl;
    cin>>aku.angkatan;
    cout<<"Data Mahasiswa"<<endl;
    cout<<"Nama : "<<aku.nama<<endl;
    cout<<"Angkatan : "<<aku.angkatan<<endl;
    return 0;
}
```

1. Apakah maksud dari program di atas ?
2. Buatlah bentuk typedef yang lainnya

LAB 1.4: CEK POIN 1

Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut. Jika terdapat error, perbaiki hingga dapat dikompilasi dan dijalankan sehingga menghasilkan keluaran seperti yang diharapkan.

```
#include <iostream>
using namespace std;
struct Mahasiswa
{
    char nama[40];
    int angkatan;
};
```

```
int main()
{
    mahasiswa mhs;
    cout<<"Masukkan Nama Mahasiswa :";
    cin.get(mhs.nama,40);
    cout<<"Masukkan Angkatan :";
    cin>>angkatan;

    cout<<"Data Mahasiswa"<<endl;
    cout<<"Nama : "<<mhs.nama<<endl;
    cout<<"Angkatan : "<<mhs.angkatan<<endl;

    cout<<"Alokasi memori untuk elemen nama : " <<&mhs.nama<<endl;
    cout<<"Alokasi memori untuk elemen angkatan : " <<&mhs.angkatan<<endl;

    cout<<"Ukuran dari variabel union : " << sizeof(mahasiswa)<<endl;
    return 0;
}
```

LAB 1.5: CEK POIN 2

Buatlah program menampilkan data 5 jenis buku menggunakan struct, tentukan setidaknya mempunyai 4 elemen struct dengan tipe data yang berbeda beda.

LAB 1.6: CEK POIN 3

Buatlah program menggunakan union untuk menampilkan 3 data jenis film, tentukan setidaknya mempunyai 5 elemen struct dengan tipe data yang berbeda beda. Lengkapi juga dengan menampilkan ukuran byte tiap elemen dan ukuran byte dari variabel union tersebut.

LAB 1.7: REFERENSI

Di bawah ini adalah referensi online dari Modul 4. Apabila mengalami kesulitan dalam memahami materi, silahkan buka referensi di bawah ini :

1. https://www.tutorialspoint.com/cplusplus/cpp_data_structures.htm

LAB 1.8: SEKILAS MATERI UNIT BERIKUTNYA

Materi praktikum pemrograman dasar unit selanjutnya akan lebih banyak membahas beberapa hal, terkait dengan pendekatan pemrograman modular.

1. Pengertian dasar program modular.
2. Perbedaan dasar antara program modular dan tidak modular.
3. Implementasi program modular