

LECTURE NOTES

COMP6599 – Algorithm and Programming

Minggu 7

Sesi 11

Structure & Union

LEARNING OUTCOMES

Learning Outcomes (Hasil Pembelajaran) :

Setelah menyelesaikan pembelajaran ini mahasiswa akan mampu :

1. LO 2 : Menerapkan sintaks-sintaks dan fungsi-fungsi bahasa pemrograman C dalam pemecahan masalah.
2. LO 3 : Membuat program dengan menggunakan bahasa C dalam pemecahan masalah.

OUTLINE MATERI(Sub-Topic):Structures & Union

1. Structure
2. Typedef
3. Union
4. Enumeration

Structures & Union

1. Structure

Structure atau *struct* merupakan struktur data yang menggabungkan beberapa data yang berbeda tipe (*heterogen*) tetapi data tersebut saling berkaitan. Misalnya data mengenai *StudentId*, nama, dan IPK seorang mahasiswa. Ketiga data ini memiliki tipe data yang berbeda tetapi masih saling berhubungan yaitu data akademik seorang mahasiswa. Dengan menggunakan *struct* maka data ini bisa diolah per-elemen (per *field*) atau secara keseluruhan (per *struct*, per *record*). Berikut contoh sintaks *struct* pada bahasa C :

```
struct [type_name]
{
    member definition;
    member definition;
    ...
    member definition;
} [one or more structure variables];
```

Type_name : adalah nama tipe *struct*.

Member definition : adalah nama *member* atau *field*.

one or more structure variables : adalah nama variabel *struct*.

Berikut adalah beberapa contoh deklarasi *struct* pada bahasa C :

```
//deklarasi variabel struct.
struct {
    char StudentId[11];
    float ipk;
    char nama[50];
}mhs;

//deklarasi tipe struct dan variabel secara bersamaan.
struct data{
    char StudentId[11];
    float ipk;
    char nama[50];
}mhs;
```

```
//deklarasi tipe struct dan variabel struct secara terpisah.
struct data{
    char StudentId[11];
    float ipk;
    char nama[50];
};

struct data mhs;
```

Pada beberapa *compiler* C, *keyword* *struct* pada deklarasi variabel *struct* secara terpisah dapat dihilangkan. Seperti tipe data yang lain, sebuah variabel *struct* dapat diinisialisasi diawal pada saat pendeklarasian. Berikut contoh inisialisasi *struct* pada C:

```
#include<stdio.h>

struct data{
    char StudentId[11];
    float ipk;
    char nama[50];
};

int main()
{
    struct data mhs = {"1900012223", 3.78, "Adam Syahputra"};
    printf("StudentId : %s\n", mhs.StudentId);
    printf("IPK : %.2f\n", mhs.ipk);
    printf("Nama : %s\n", mhs.nama);

    getchar();
    return 0;
}
```

Untuk mengakses *field struct* gunakan dotoperator yaitu nama variabel *struct* diikuti titik dan nama *field*. Berikut adalah hasil output dari contoh inisialisasi sturct diatas :

```
StudentId : 1900012223
IPK : 3.78
Nama : Adam Syahputra
```

2. Typedef

Pada bahasa pemrograman C disediakan *keyword typedef*, yang dapat digunakan untuk membuat tipe penamaan baru.

Misalnya *typedef* dapat digunakan dengan *struct* untuk mendefinisikan tipe data baru dan kemudian tipe data ini digunakan untuk mendefinisikan variabel *struct*, dengan contoh sebagai berikut :

```
#include <stdio.h>
#include <string.h>
typedef struct Books
{
    char title[50];
    int book_id;
} Book;    //variable struct

int main( )
{
    Book book;    //struct name
    strcpy( book.title, "C Programming");
    book.book_id = 6495407;

    printf( "Book title : %s\n", book.title);
    printf( "Book book_id : %d\n", book.book_id);

    getchar();
    return 0;
}
```

Berikut output dari program tersebut adalah :

```
Book title : C Programming
Book book_id : 6495407
_
```

3. Union

Union digunakan untuk kongsi (menggabungkan) memori. Dengan menggunakan union, suatu lokasi memori dapat ditempati oleh dua atau beberapa variabel dengan masing-masing tipe data yang berbeda. Berikut sintaks dalam penggunaan union pada bahasa C :

```
union [type_name]
{
    member definition;
    member definition;
    ...
    member definition;
} [one or more union variables];
```

Type_name : adalah nama tipe *union*.

Member definition : adalah nama-nama variabel yang kongsi memori, minimal memiliki 2 variabel.

one or more union variables : adalah nama variabel *union*.

Berikut contoh penggunaan *union* pada bahasa C :

```
#include <stdio.h>
#include <string.h>

union Data
{
    int i;
    float f;
    char str[20];
};

int main( )
{
    union Data data;

    printf( "Memory size occupied by data : %d\n", sizeof(data));

    getchar();
    return 0;
}
```

Pada contoh program tersebut terlihat bahwa memori terbesar adalah **charstr** sebesar 20 *byte* maka memori yang dibutuhkan adalah sebesar 20 *byte*. Dari program tersebut akan menghasilkan *output* sebagai berikut :

```
Memory size occupied by data : 20
```

4. Enumeration

Enumeration adalah tipe data yang ditetapkan oleh pengguna yang terdiri dari konstanta integral dan setiap konstanta tersebut diberikan nama. Berikut sintaks *enum* pada bahasa C :

```
enumtype_name{ value1, value2,...,valueN };
```

Type_name : adalah nama dari tipe data *enum*.

Value : nilai dari *enum*.

Berikut adalah contoh penggunaan enum pada bahasa C :

```
#include <stdio.h>
#include <string.h>

enum week{ sunday, monday, tuesday, wednesday, thursday, friday, saturday};

int main()
{
    enum week today;
    today=wednesday;
    printf("%d day",today+1);

    getchar();
    return 0;
}
```

Berikut hasil dari *output* program diatas :

```
4 day_
```

SIMPULAN

1. *Structure* atau *struct* merupakan struktur data yang menggabungkan beberapa data yang berbeda tipe (*heterogen*) tetapi data tersebut saling berkaitan.
2. *Typedef*, yang dapat digunakan untuk membuat tipe penamaan baru. Misalnya *typedef* dapat digunakan dengan *struct* untuk mendefinisikan tipe data baru dan kemudian tipe data ini digunakan untuk mendefinisikan variabel *struct*.
3. Union digunakan untuk kongsi (menggabungkan) memori. Dengan menggunakan union, suatu lokasi memori dapat ditempati oleh dua atau beberapa variabel dengan masing-masing tipe data yang berbeda.
4. *Enumeration* adalah tipe data yang ditetapkan oleh pengguna yang terdiri dari konstanta integral dan setiap konstanta tersebut diberikan nama.

DAFTAR PUSTAKA

1. Paul J. Dietel, Harvey M. Deitel,. 2010. C : how to program. PEAPH. New Jersey. ISBN:978-0-13-705966-9 Chapter 10
2. Collecting Data Items of Different Types: <http://aelinik.free.fr/c/ch19.htm>
3. Structs, Enums, and Unions: <http://www.lysator.liu.se/c/c-faq/c-9.html>
4. http://www.tutorialspoint.com/cprogramming/c_structures.htm
5. http://www.tutorialspoint.com/cprogramming/c_typedef.htm
6. http://www.tutorialspoint.com/cprogramming/c_unions.htm
7. <http://www.programiz.com/c-programming/c-enumeration>
8. Thompson Susabda Ngoen, 2006. Pengantar Algoritma dengan Bahasa C. Salemba Teknik. ISBN : 979-9549-25-6. Bagian 6.