

JOB SHEET 10

Enumerasi & Structure

*"Everybody should learn to program a computer,
because it teaches you how to think."*

TUJUAN PEMBELAJARAN

1. Mampu menjelaskan dan mengimplementasikan *Enum* dan *Structure* dalam pemrograman menggunakan IDE.

POKOK MATERI

1. Pengertian Enum
2. Pengertian Structure
3. Penggunaan Enum
4. Penggunaan Struct
5. Penggunaan typedef pada Struct
6. Struct Bersarang
7. Passing Struct ke dalam fungsi

URAIAN MATERI

A. Pengertian Enumerasi

Enum atau Enumeration merupakan tipe data yang berisi sekumpulan konstanta. Enum termasuk dalam tipe data bentukan, karena tipe data ini dibuat sendiri oleh kita.

Berikut ini cara membuat enum:

nama atau
flag untuk enum

konstanta

Jangan lupa akhiri
dengan titik koma

```
enum nama_enum { const1, const2, ... };
```

konstanta
dipisah dengan
tanda koma

Contoh:

```
enum level {EASY = 1, NORMAL = 2, HARD = 3};  
enum sex {MALE, FEMALE};  
enum state {PENDING, INPROGRESS, DONE};
```

Apabila kita tidak mengisi nilai konstanta pada enum, maka secara default akan diisi dengan mengikuti urutan penulisannya.

Contoh:

```
enum hari {SENIN, SELASA, RABU, KAMIS, JUMAT, SABTU, MINGGU};
```

Maka konstanta yang ada di dalam enum hari akan bernilai:

```
SENIN   = 0  
SELASA  = 1  
RABU    = 2  
KAMIS   = 3  
JUMAT   = 4  
SABTU   = 5  
MINGGU  = 6
```

Membuat program dengan enum

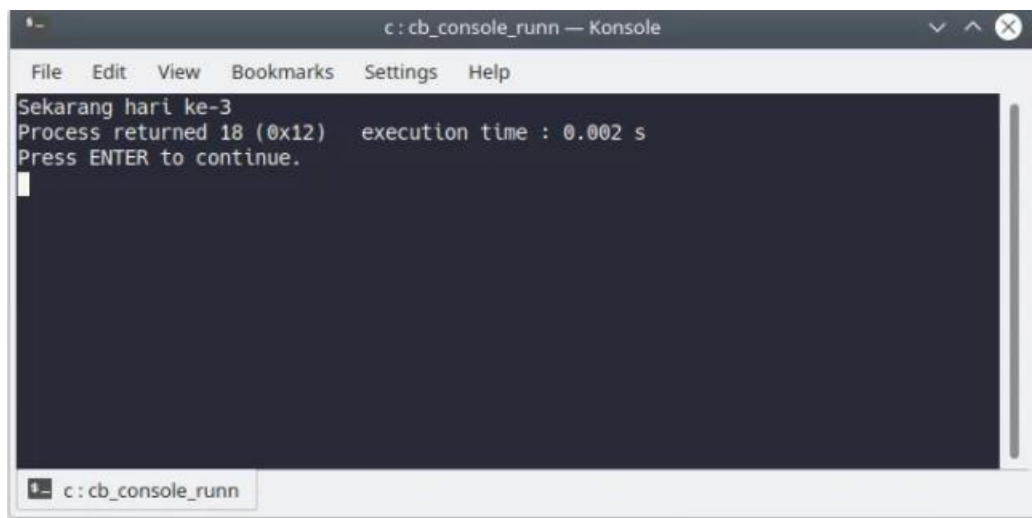
Silahkan buat program baru bernama contoh_enum.c, kemudian isi dengan kode berikut.

```
#include <stdio.h>

enum hari {SENIN, SELASA, RABU, KAMIS, JUMAT, SABTU, MINGGU};

void main() {
    enum hari sekarang;
    sekarang = RABU;
    printf("Sekarang hari ke-%d", sekarang + 1);
}
```

Hasilnya:



Membuat Tipe Data Boolean dari Enum

Bahasa pemrograman C tidak memiliki tipe data boolean seperti bahasa pemrograman lain. Namun, kita bisa membuatnya sendiri. Pada bahasa C, nilai true akan sama dengan 1 dan false sama dengan 0.

Maka kita bisa membuat enum untuk boolean seperti ini:

```
enum boolean {false, true};
```

Buatlah program baru bernama contoh_boolean.c, kemudian isi dengan kode berikut:

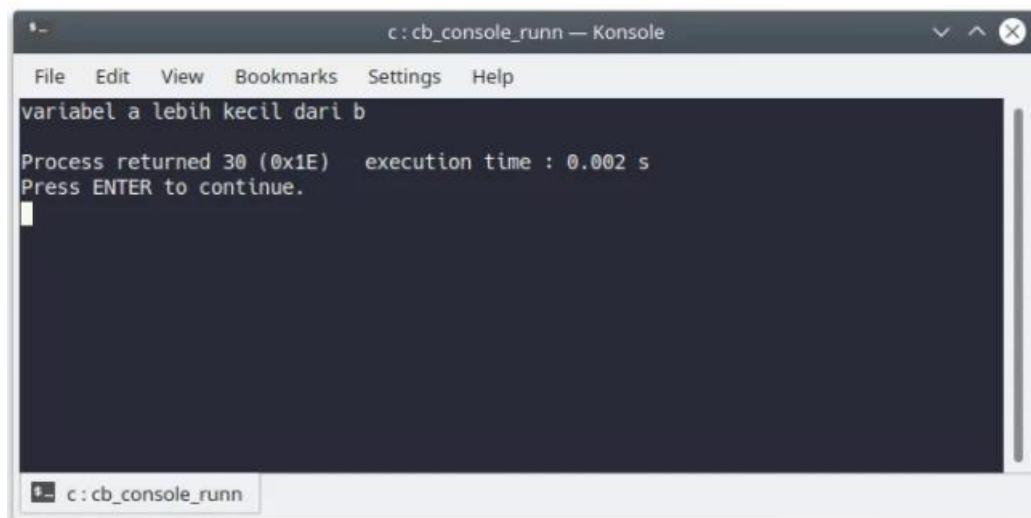
```
#include <stdio.h>

enum boolean {false, true};

void main() {
    int a = 4;
    int b = 5;

    if(a < b == true){
        printf("variabel a lebih kecil dari b\n");
    }
}
```

Hasilnya:

A screenshot of a console window titled "c : cb_console_runn — Konsole". The window has a menu bar with "File", "Edit", "View", "Bookmarks", "Settings", and "Help". The main area shows the output "variabel a lebih kecil dari b" on the first line. Below it, it says "Process returned 30 (0x1E) execution time : 0.002 s" and "Press ENTER to continue." The status bar at the bottom shows "c : cb_console_runn".

Mengapa Membutuhkan Enum?

Karena enum akan memberikan efisiensi dan fleksibilitas dalam penggunaan memori.

Variabel enum hanya akan mengambil satu nilai dari konstanta yang ada.

Contoh:

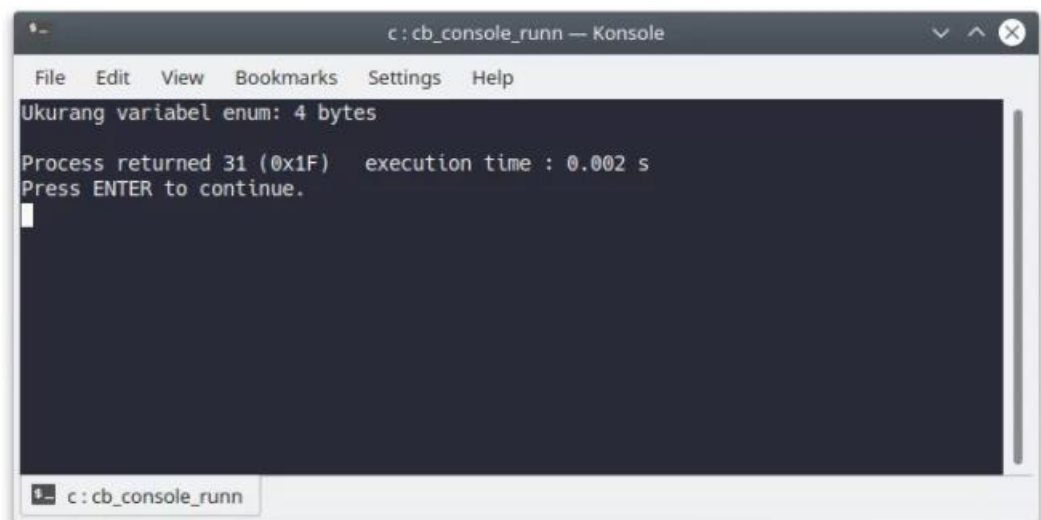
```
#include <stdio.h>

enum hari {SENIN, SELASA, RABU, KAMIS, JUMAT, SABTU, MINGGU};

void main() {
    enum hari sekarang;
    sekarang = RABU;

    printf("Ukuran variabel enum: %d bytes\n", sizeof(sekarang));
}
```

Hasilnya:

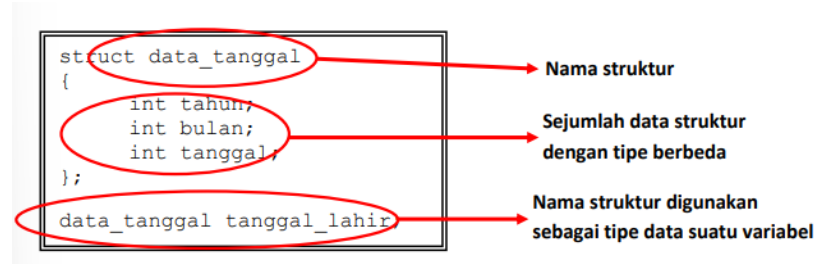


Sedangkan apabila kita menggunakan array atau struct, ukuran dari variabelnya akan mengikuti banyaknya isi dari array dan struct.

B. Structure

Structure atau struct adalah kumpulan dari beberapa variabel dengan beragam tipe data yang dibungkus dalam satu variabel.

Struct juga dikenal dengan records dalam bahasa pemrograman lain seperti Pascal.



Pada Contoh di atas, struktur bernama `data_tanggal`, yang tersusun atas 3 unit penyusunnya yaitu `tahun`, `tanggal`, dan `bulan`. Setelah dibuat struktur tanggal, selanjutnya struktur tersebut dapat digunakan sebagai tipe data suatu variabel, dalam hal ini disebut tipe data abstrak.

Mengapa kita membutuhkan struct?

Misalnya kita ingin menyimpan data mahasiswa. Kita bisa saja melakukannya seperti ini:

```

char name[] = "Dian";
char address[] = "Mataram";
int age = 22;

```

Lalu bagaimana kalau ada lebih dari satu mahasiswa?

Mungkin bisa saja kita buat seperti ini:

```

char name[] = "Dian";
char address[] = "Mataram";
int age = 22;

char name2[] = "Bambang";
char address2[] = "Surabaya";
int age2 = 23;

char name3[] = "Bimo";
char address3[] = "Jakarta";
int age3 = 23;

```

Terlihat kurang bagus.

Cara terbaik adalah menyimpan data ini adalah menggunakan struct.

Penggunaan Struct

Struct dapat kita buat dengan kata kunci struct kemudian diikuti dengan nama struct dan isinya.



Contoh:

```
struct Mahasiswa
{
    char *name;
    char *address;
    int age;
};
```

Catatan: kita menggunakan pointer * untuk data string

Agar struct dapat digunakan, kita harus membuat variabel untuknya.

Buatlah program baru dengan nama contoh_struct.c kemudian isi dengan kode berikut.

```
#include <stdio.h>

// membuat struct
struct Mahasiswa {
    char *name;
    char *address;
    int age;
};

void main(){

    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;

    // mengisi nilai ke struct
    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;

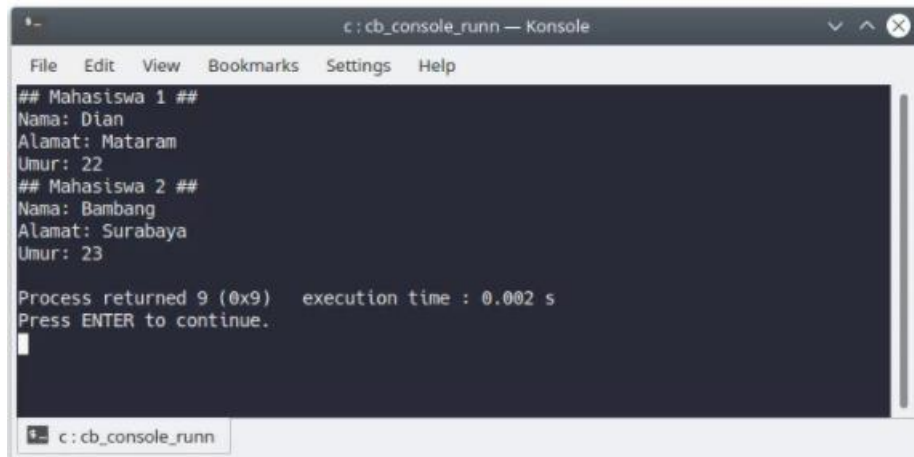
    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;

    // mencetak isi struct
    printf("## Mahasiswa 1 ##\n");
    printf("Nama: %s\n", mhs1.name);
    printf("Alamat: %s\n", mhs1.address);
    printf("Umur: %d\n", mhs1.age);

    printf("## Mahasiswa 2 ##\n");
    printf("Nama: %s\n", mhs2.name);
    printf("Alamat: %s\n", mhs2.address);
    printf("Umur: %d\n", mhs2.age);

}
```


Hasilnya:



```
c:cb_console_runn — Konsole
File Edit View Bookmarks Settings Help

## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23

Process returned 9 (0x9)  execution time : 0.002 s
Press ENTER to continue.

c:cb_console_runn
```

Menggunakan typedef pada Struct

Kata kunci typedef adalah kata kunci untuk mendefinisikan tipe data baru.

Kita bisa menggunakan kata kunci ini di depan struct untuk menyatakannya sebagai tipe data baru.

Contoh:

Tanpa typedef kita akan menggunakan struct seperti ini.

```
// membuat struct
struct Distance{
    int feet;
    float inch;
};

void main() {
    // menggunakan struct
    struct Distance d1, d2;
}
```

Kalau menggunakan typedef akan menjadi seperti ini:

```
// membuat struct dengan typedef
typedef struct Distance{
    int feet;
    float inch;
} distances;

void main() {
    // menggunakan struct
    distances dist1, dist2, sum;
}
```

Struct Bersarang

Struct dapat dibuat bersarang (nested). Bersarang artinya ada struct di dalam struct.

Contoh:

```
struct complex
{
    int imag;
    float real;
};

struct number
{
    struct complex comp;
    int integers;
} num1, num2;
```

Lalu cara menggunakannya akan seperti ini:

```
num1.integer = 12;
num1.comp.real = 44.12;
num2.comp.imag = 11;
```

Passing Struct ke dalam fungsi

Struct dapat kita buat sebagai parameter untuk fungsi.

Contoh:

```
#include <stdio.h>
struct student
{
    char name[50];
    int age;
};

void main() {
    struct student s1;

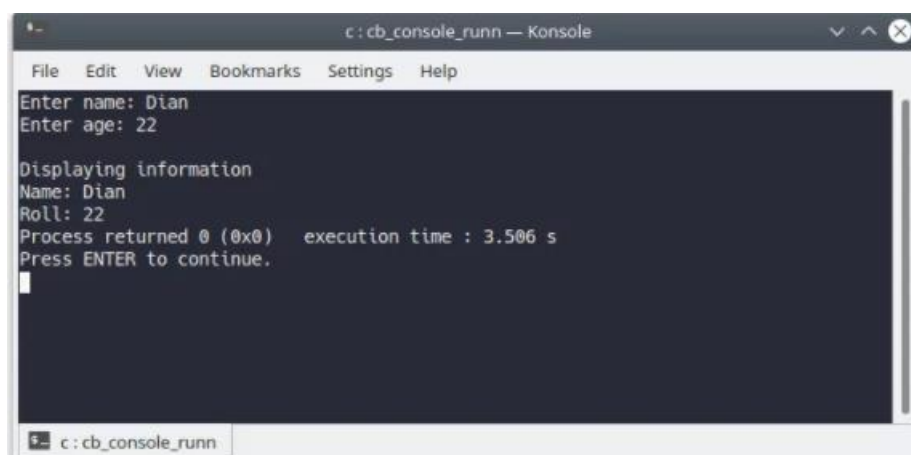
    printf("Enter name: ");
    scanf("%[^\n]*c", s1.name);

    printf("Enter age: ");
    scanf("%d", &s1.age);

    display(s1); // passing structure as an argument
}

// membuat fungsi dengan struct sebagai parameter
void display(struct student s) {
    printf("\nDisplaying information\n");
    printf("Name: %s", s.name);
    printf("\nRoll: %d", s.age);
}
```

Hasilnya:



```
c:cb_console_runn — Konsole
File Edit View Bookmarks Settings Help
Enter name: Dian
Enter age: 22

Displaying information
Name: Dian
Roll: 22
Process returned 0 (0x0)   execution time : 3.506 s
Press ENTER to continue.
c:cb_console_runn
```

LATIHAN

1. Buatlah game sederhana, di mana komputer memilih nomor secara acak dalam rentang antara 1 dan 20. Anda kemudian akan diminta untuk menebak nomor yang dipilih oleh Komputer.

Untuk memberi Anda petunjuk, komputer akan menjawab tebakan dengan kondisi berikut:

- a) Jika tebakan nomor anda lebih rendah dari nomor komputer, outputnya harus: **Nomor saya lebih besar!**
- b) Jika tebakan nomor anda lebih tinggi dari nomor komputer, outputnya: **Nomor saya lebih kecil!**
- c) Ketika Anda menemukan jawaban yang tepat, komputer harus memberi selamat kepada Anda. **Selamat, anda benar!** Kemudian Program juga menampilkan jumlah percobaan yang diperlukan untuk menebak nomor tersebut.
- d) Program menyediakan kesempatan untuk bermain lagi 'y' atau menghentikan permainan dengan menekan 'n' pada keyboard.

Dalam generate nomor acak anda bisa menggunakan `srand(time(NULL));`

Pelajari tentang penggunaan tersebut!

2. Buat program untuk menentukan zodiak kelahiran dengan menggunakan Array dengan Struktur.

Bentuk keluaran:

Tanggal Lahir Anda [tgl-bln-tahun] : 24-12-1970

Zodiak Anda adalah : CAPRICORN

3. Buat program untuk menyimpan data mahasiswa dengan menggunakan struktur.

Bentuk tampilan masukan:

NPM : 10400570
NAMA : Muh. Rizki Putra
TGL LAHIR : 09-19-1980
ALAMAT : Padang
HP : 7874528

Mau memasukkan data lagi [y/t]?

Bentuk tampilan output:

10400570	Muh. Rizky Putra	09-09-1980	Padang	7875428
10400572	Pratama Sutisna	11-08-1978	Pariaman	7987555

REFERENCES

1. The C Programming Language. 2nd Edition
2. [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
3. https://en.wikipedia.org/wiki/Imperative_programming
4. <https://www.petanikode.com/tutorial/c/>
5. <https://www.learn-c.org/>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.programiz.com/>
8. <https://www.dicoding.com/>
9. <https://data-flair.training/blogs/c-tutorials-home/>