

MAKALAH STRUKTUR DATA

Tipe Data Turunan



GOLONGAN TIF B BWS

Disusun oleh :

Nama : Fahmi Dafrin Maulana

Nim : E41201302

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI POLITEKNIK
NEGERI JEMBER
2021**

A. Tipe data structure

a. Pengertian struct

Dalam bahasa C, struct adalah tipe data bentukan yang terdiri dari kumpulan tipe data lain. Struct mirip seperti array, tapi struct bisa menampung lebih dari 1 jenis tipe data. Jika sebelumnya anda pernah belajar bahasa pemrograman Pascal, struct dalam bahasa C sangat mirip seperti record di Pascal.

Tipe data struct cocok dipakai untuk menampung data berkelompok. Misalnya saya ingin membuat biodata siswa. Setiap siswa memiliki data nama, nama sekolah serta jumlah uang saku. Menggunakan cara biasa, semua data ini bisa ditampung ke dalam 3 variabel berikut:

```
1 char nama_siswa01[50] = "Andi Permana Nugroho";  
2 char nama_sekolah_siswa01[50] = "SMA 1 Lumut Ijo";  
3 unsigned int uang_saku_siswa01 = 10000;
```

Untuk siswa kedua, tinggal membuat penamaan yang sama, yakni nama_siswa02, nama_sekolah_siswa02 dan uang_saku_siswa02.

Cara ini tidak salah, namun setiap variabel seolah-olah terpisah satu sama lain. Satu-satunya yang menghubungkan ketika variabel hanya dari penamaan saja. Kita juga tidak bisa menggunakan array karena 3 data di atas memiliki tipe data yang berbeda. Dalam kasus seperti inilah tipe data struct cocok digunakan.

b. Cara Penggunaan Tipe Data Struct

Tipe data struct adalah tipe data bentukan, dimana kita harus buat dulu seperti apa isi komponen dari struct, baru kemudian diakses dari kode program utama.

Kita akan bahas dengan contoh kode program berikut ini:

```

#include <stdio.h>

struct Siswa {
    char  nama[50];
    char  sekolah[50];
    unsigned int  uangSaku;
};

int main(void)
{
    struct Siswa siswa01;

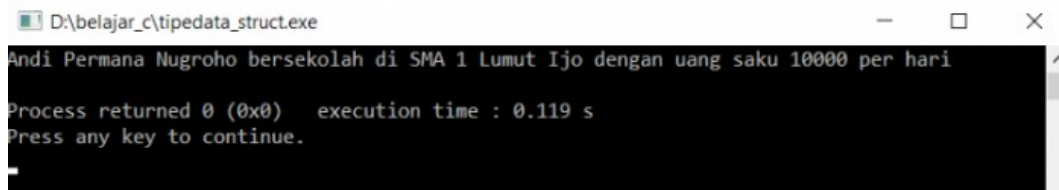
    strcpy(siswa01.nama, "Andi Permana Nugroho");
    strcpy(siswa01.sekolah, "SMA 1 Lumut Ijo");
    siswa01.uangSaku = 10000;

    printf("%s bersekolah di %s ", siswa01.nama, siswa01.sekolah);
    printf("dengan uang saku %i per hari\n", siswa01.uangSaku);

    return 0;
}

```

Hasil kode program :



```

D:\belajar_c\tipedata_struct.exe
Andi Permana Nugroho bersekolah di SMA 1 Lumut Ijo dengan uang saku 10000 per hari
Process returned 0 (0x0) execution time : 0.119 s
Press any key to continue.

```

Proses pembuatan tipe data struct ditulis sebelum main(), yang dalam contoh ada di baris 3 – 7. Setelah keyword struct, diikuti dengan nama identitas atau nama tipe data yang kita inginkan. Dalam contoh ini saya membuat tipe data Siswa. Tentu saja anda bisa menggunakan nama lain.

Sebuah struct terdiri dari berbagai variabel yang ditulis di antara tanda kurung kurawal. Untuk tipe data struct Siswa, saya membuat variabel nama, sekolah dan uangSaku. Setiap variabel tetap ditulis tipe data asalnya. Perintah di baris 3 – 7 dipakai untuk membuat struktur dasar struct. Agar bisa digunakan, kita harus membuat variabel baru dengan tipe data Siswa, yakni tipe data bentukan dari struct. Masuk kode kode program utama (main), saya mendefinisikan variabel siswa01 sebagai struct Siswa di baris 11.

Sekarang, variabel siswa01 adalah tipe data bentukan yang terdiri dari 3 element, yakni:

siswa01.nama

siswa01.sekolah

siswa01.uangSaku

Tanda titik dipakai untuk mengakses sub struktur dari siswa01. Semua ini sesuai dengan pendefinisian struct Siswa. Proses pengisian data ke dalam variabel siswa01 dilakukan pada baris 13 – 15. Karena nama dan sekolah adalah string, maka saya harus menggunakan fungsi strcpy. Sedangkan untuk uangSaku bisa langsung diisi sebagaimana cara pengisian variabel biasa. Di bagian akhir kode program, saya menampilkan ketiga data siswa01. Caranya sama seperti variabel biasa, hanya saja sekarang kita menggunakan struct Siswa sehingga cara aksesnya adalah siswa01.nama, siswa01.sekolah, dan siswa01.uangSaku.

B. Tipe data pointer

Pointer (dalam pemrograman C) adalah suatu variable yang berisikan suatu alamat lokasi tertentu. Yang disimpan oleh variabel yang didefinisikan sebagai pointer adalah ‘alamat’ variabel tersebut. Seperti yang kita ketahui, variabel pointer berfungsi sebagai penunjuk alamat lokasi variabel lain, dan mengganti nilai variable lyang ditunjuk tersebut Pada operasi pointer, kita menggunakan 2 buah operator dalam bahasa C yaitu “*” dan “&”. Dalam pembuatan program menggunakan pointer, tanda asterisk “*” digunakan untuk menentukan variable yang menjadi suatu pointer.

Pada deklarasi, penulisan awal deklarasi dapat dibuat sebagai berikut :

Tipe_data

*nama_variabel_komputer

Seperti contoh, kita membuat deklarasi int *b, berarti sebuah pointer hanya dapat menunjuk sebuah variable dengan tipe data integer

Untuk mendapatkan alamat dari suatu variabel, maka kita perlu menggunakan operator “&” di depan variable tersebut. Adapun contoh dalam penggunaannya :

```
printf("Alamat dari variabel a = %d", &a);
```

Maka pada saat program dijalankan, akan muncul alamat dari variabel a tersebut. Untuk mendapatkan nilai dari suatu variable, maka perintah yang digunakan cukup hanya dengan memanggil variabel nya saja.

Adapun contoh penggunaannya :

```
Printf("Isi dari variabel a = %d", a);
```

Maka pada saat program dijalankan, akan muncul nilai dari variabel a tersebut

Untuk mendapatkan nilai dari suatu alamat yang terdapat pada isi pointer, maka perintah yang digunakan adalah mendambah tanda asterisk “*” di depan variabel. Adapun contoh penggunaannya :

```
*b=&a
```

```
Printf("Isi dari variabel yang ditunjuk poiter adalah %d", *b);
```

Maka pada saat program dijalankan, akan muncul nilai dari variabel yang ditunjuk oleh pointer b, yaitu nilai dari variabel a tersebut pemrograman C) adalah suatu variable yang berisikan suatu alamat lokasi tertentu. Yang disimpan oleh variabel yang didefinisikan sebagai pointer adalah ‘alamat’ variabel tersebut. Seperti yang kita ketahui, variabel pointer berfungsi sebagai penunjuk alamat lokasi variabel lain, dan mengganti nilai variable lyang ditunjuk tersebut Pada operasi pointer, kita menggunakan 2 buah operator dalam bahasa C yaitu “*” dan “&”. Dalam pembuatan program menggunakan pointer, tanda asterisk “*” digunakan untuk menentukan variable yang menjadi suatu pointer.

Pada deklarasi, penulisan awal deklarasi dapat dibuat sebagai berikut :

```
Tipe_data *nama_variabel_komputer
```

Seperti contoh, kita membuat deklarasi `int *b`, berarti sebuah pointer hanya dapat menunjuk sebuah variable dengan tipe data integer. Untuk mendapatkan alamat

dari suatu variabel, maka kita perlu menggunakan operator “&” di depan variable tersebut

C. Tipe data Enumeration

Tipe data enum atau tipe data enumerated adalah tipe data khusus yang isinya bisa kita tentukan sendiri. Sebagai contoh, kita bisa membuat tipe enum warna yang hanya bisa diisi nilai: merah, kuning, biru dan ungu. Atau tipe enum nama_hari yang hanya bisa di isi nama-nama hari seperti senin, Selasa, Rabu, dst.

Setelah di definisikan, data yang bisa diinput hanya nilai yang telah ditentukan saja, tidak bisa di isi nilai lain. Format Penulisan Tipe Data Enum C++ yaitu sama seperti tipe data struct, tipe data enum adalah tipe data bentukan. Kita harus buat dulu seperti apa isi komponen dari enum, baru kemudian diakses dari function main(). Tipe data enum C++ dibuat menggunakan keyword enum. Format dasarnya adalah sebagai berikut:

```
enum nama_tipe {nilai1, nilai2, nilai3... };
```

Sebagai contoh, jika saya ingin membuat tipe enum nama_hari yang berisi nama-nama hari, bisa menggunakan perintah berikut:

```
enum nama_hari {minggu, senin, Selasa, Rabu, Kamis, Jumat, Sabtu};
```

Dalam kode diatas, enum nama_hari terdiri dari nama seperti minggu, senin, Selasa... Sabtu. Agar bisa dipakai, tipe data enum harus di deklarasikan oleh sebuah variabel di dalam function main(). Penjelasan ini akan lebih mudah dengan contoh kode program.

```

1  #include <iostream>
2
3  using namespace std;
4
5  enum nama_hari {minggu, senin,selasa,rabu,kamis,jumat,sabtu};
6
7  int main()
8  {
9      nama_hari var1;
10
11     var1 = senin;
12     cout << var1 << endl;
13
14     var1 = rabu;
15     cout << var1 << endl;
16
17     var1 = sabtu;
18     cout << var1 << endl;
19
20     return 0;
21 }

```

Hasil kode program:

```

1
3
6

```

Dalam kode program ini pendefinisian enum nama_hari dilakukan pada baris 5. Kemudian di dalam function main(), variabel var1 di deklarasikan agar bertipe data nama_hari di baris 9. Cara pendeklarasian ini mirip seperti tipe data biasa, hanya saja sekarang menggunakan tipe data enum nama_hari. Selanjutnya var1 di isi dengan nilai senin, rabu, dan sabtu secara bergantian. Setelah itu isi variabel var1 ditampilkan dengan perintah cout. Dari hasil kode program, terlihat bahwa nilai yang tersimpan dalam var1 adalah angka, atau lebih tepatnya index dari posisi setiap nilai tipe data enum. Sebagai contoh, pada saat pendefinisian enum nama_hari, hari senin berada di posisi kedua, yakni di index 1 (ingat, index mulai dari 0). Maka hasil dari cout << var1 adalah 1. Begitu juga saat var1 diisi dengan hari sabtu, perintah cout menampilkan angka 6. Ciri khas dari tipe data enum adalah, kita tidak bisa menginput nilai selain yang sudah didaftarkan