

Pertemuan 11

Pemrograman Terstruktur

Tipe data Struktur (Struct) dan
Union

Definisi STRUKTUR (*STRUCT*)

- ❖ **Struktur (*struct*)** adalah koleksi dari variabel yang dinyatakan dengan sebuah nama , dengan sifat setiap variabel dapat memiliki tipe yang berlainan.
- ❖ Pendeklarasian struct biasanya dilakukan diawal, setelah pendeklarasian file header dan sebelum fungsi main
- ❖ Cara pendeklarasian struct ada beberapa cara :
 - Deklarasi biasa
 - Deklarasi dengan *typedef*
 - Deklarasi dalam linked list
 - Deklarasi *struct* dalam *struct*

Deklarasi Struktur : Deklarasi biasa

Syntax:

```
struct nama_tipe_struct
{
    tipe field1;
    tipe field2;
    .
    .
    tipe fieldn;
} variabel_strukturl,...,
variabel_strukturn;
```

Contoh (Membuat struct dengan nama person)

```
struct person
{
    char name[25];
    int age;
    char status; /* M =
married, S = single */
};
```

Deklarasi Struktur: Deklarasi dengan

Syntax:

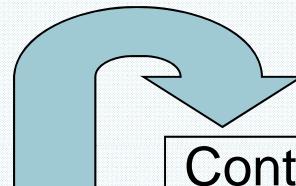
```
typedef struct
{
    tipe field2;
    .
    .
    tipe fieldn;
}tipe struct ;
```

Contoh:

- Dalam proyek untuk penelitian lokal diperlukan database untuk menyimpan data berikut
 - Name : Jupiter
 - Diameter : 142,800 km
 - Moons : 16
 - Orbit time : 11,9
 - Rotation time : 9,925 hr

Typedef

Data yang diperlukan disimpan dalam struck dengan nama **planet_t**



Contoh tersebut dapat diselesaikan dengan membuat struct berikut:

Deklarasi Struktur:Deklarasi dalam Linked List

```
typedef struct {  
    tipe field1;  
    tipe field2;  
    tipe fieldn;  
    tipe_struct *next;  
}tipe_struct;
```

Deklarasi Struktur: Deklarasi struktur dalam Struktur

```
typedef struct {  
    tipe field1;  
    tipe field2;  
    tipe fieldn;  
}tipe_struct1;  
  
typedef struct {  
    tipe_struct1 field1;  
    tipe field2;  
    tipe fieldn;  
    tipe_struct2 *next;  
}tipe_struct2;
```

Dalam hal ini struct kedua juga berisi struct pertama.

Deklarasi Variabel dengan Tipe Struktur

- ❖ Pendeklarasian variabel agar bertipe struktur sesuai dengan struktur yang telah dideklarasikan :

- struct nama_tipe_struct nama_variabel;

Contoh : **struct person descrip;**

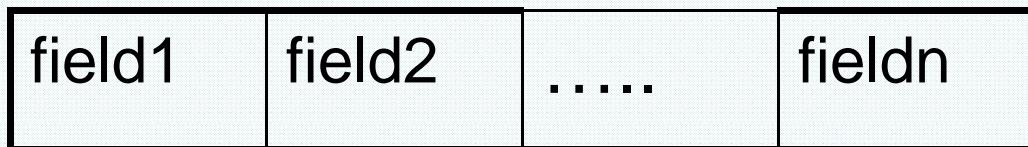
*/*pendeklarasian variabel descrip agar bertipe struktur seperti struktur person*/*

- tipe_struct nama_variabel;(jika pendeklarasian struktur sebelumnya menggunakan *typedef*)

Contoh : **planet_t current_planet;**

*/*pendeklarasian variabel current_planet dengan struktur planet yang didefinisikan dengan typedef*/*

- ❖ Bentuk satu elemen struktur



PENGAKSESAN STRUKTUR

- ❖ Deklarasi untuk mengakses struktur

```
Nama_variabelStruct.nama_field
```

- Ket : antara nama variabel dengan nama field dipisahkan dengan tanda titik.

Contoh struktur mahasiswa :

```
typedef struct {  
    char NRP[10];  
    char *nama;  
    char kelas;  
}mahasiswa;
```

Mahasiswa data_mahasiswa

Contoh pengaksesan struktur data_mahasiswa :

```
data_mahasiswa.NRP;  
data_mahasiswa.nama;
```

PENGIRIMAN STRUKTUR ANTAR

FUNGSI

- ❖ Melalui parameter fungsi
 - untuk mengirim struktur
 nama_fungsi (nama_variabel_struct);
 contoh : tulis (data_mahasiswa);
 - untuk menerima struktur di dalam parameter
 tipe nama_fungsi (struct variabel)
 contoh: void tulis (struct mahasiswa)

- ❖ Dengan nilai kembalian
 - struct nama_fungsi ()

CONTOH-CONTOH STRUKTUR

PEMBAHASAN CONTOH :

Melewatkan Elemen Struktur ke dalam Fungsi
Melewatkan Struktur ke dalam Fungsi

Contoh 1 : Melewatkkan Elemen Struktur ke dalam Fungsi

```
#include <stdio.h>
void tukar (int *x, int *y),
void main ()
{
    struct koordinat
    {
        int x;
        int y;
    };
    struct koordinat posisi = { 18, 257 };
    printf ("Nilai isi variable struct \"sebelum\" masuk fungsi \n\n");
    printf ("\t\t x ==> %d\n", posisi.x);
    printf ("\t\t y ==> %d\n", posisi.y);
    tukar (&posisi.x, &posisi.y);
    printf ("\n\nNilai isi variable struct \"setelah\" memasuki Fungsii\n\n");
    printf ("\t\t x ==> %d\n", posisi.x);
    printf ("\t\t y ==> %d\n", posisi.y);
}
void tukar (int *x, int *y)
{
    int      z = *x;
    *x = *y;
    *y = z;
}
```

■ (Inactive C:\TCWIN45\BIN\NONAME00.EXE)

Nilai isi variable struct "sebelum" masuk fungsi

x ==> 18
y ==> 257

Nilai isi variable struct "setelah" memasuki Fungsi

x ==> 257
y ==> 18



Contoh 2 : Melewatkkan Struktur ke dalam Fungsi



```
printf ("\t\t y ==> %d\n", posisi.y);
proses (posisi);
}

void proses (struct koordinat posisi)
{
    printf ("\n\nNilai isi variable struct didalam fungsi \"proses\"\n\n");
    printf ("\t\t x ==> %d\n", posisi.x);
    printf ("\t\t y ==> %d\n", posisi.y);
}
```

■ (Inactive C:\TCWIN45\BIN\NONAME00.EXE)

Nilai isi variable struct didalam fungsi "main"

x ==> 18
y ==> 257

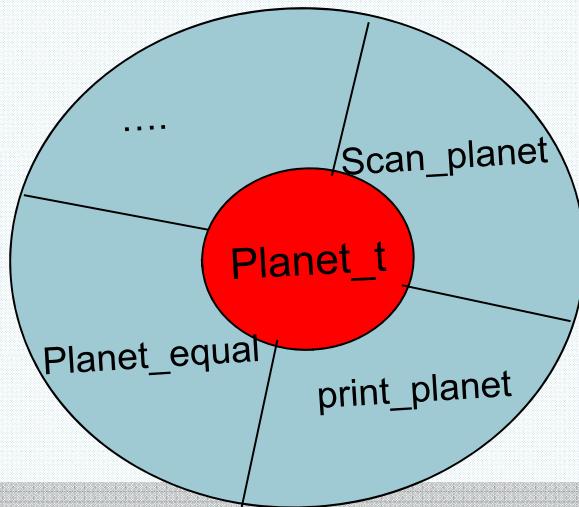
Nilai isi variable struct didalam fungsi "proses"

x ==> 18
y ==> 257



Pemecahan Masalah dengan Tipe Terstruktur

- ❖ Untuk memanipulasi data, C menyediakan semua operasi dasar yang dapat digunakan untuk memecahkan masalah dengan menggunakan tipe data standar
- ❖ Kombinasian type tertentu yang didefinisikan user dengan sekumpulan operasi-operasi dasar yang membolehkan satu kebenaran untuk melihat tipe sebagai konsep kesatuan disebut **Abstract Data Type (ADT)**
- ❖ Contoh tipe data planet_t dan operasi-operasi dasarnya digambarkan sbb:



Contoh Studi Kasus untuk ADT

- ❖ Tipe yang didefinisikan user untuk angka yang komplek
- ❖ Analisa:
 - Angka yang komplek adalah angka dengan bagian real dan imaginer
 - Contoh : $a+bi$ (a : real, b :imaginer)
 - Definisikan fungsi untuk melakukan operasi aritmetika dasar (penjumlahan, pengurangan, perkalian dan pembagian)



Desain

- ❖ 2 aspek utama dari solusi masalah ini adalah :
 - Mendefinisikan struktur dari tipe yang didefinisikan user
 - Menentukan nama fungsi, parameter, dan tujuan dari tiap operasi

Desain (cont..)

- ❖ Nama struktur yang didefinisikan sendiri :
`complex_t`
- ❖ Fungsi :
 - `Multiply_complex` (untuk perkalian bil. komplek)
 - `Divide_complex` (untuk pembagian bil. kompleks)
 - `Abs_complex` (untuk mendapatkan nilai absolut)
 - `Print_complex` (untuk menampilkan hasil)
 - `Scan_complex` (untuk melakukan pembacaan thd bil. Kompleks)
 - `Add_complex` (untuk menjumlahkan bil. Kompleks)
 - `Substract_complex` (untuk mengurangkan bil. Kompleks)

Spesifikasi dari Tipe complex_t dan operasi-

STRUCTURE: Angka Kompleks adalah suatu object dari tipe complex_t yang terdiri dari sepasang nilai bertipe double

OPERATOR:

```
/*Complex number input fuction returns standard scanning*/
int scan_complex(complex_t *c) /*output-address of complex
variable to fill*/
```

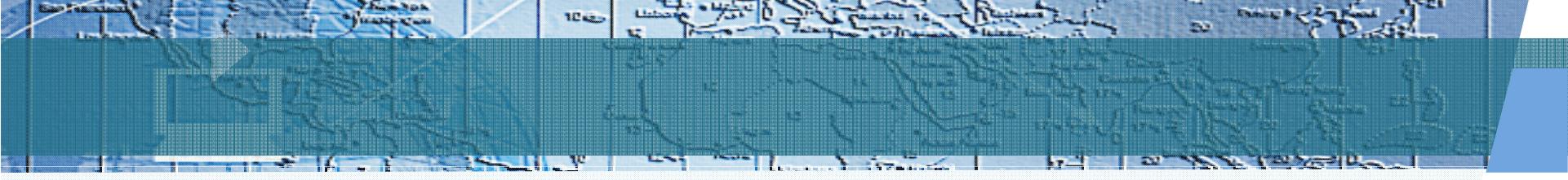
```
/*
```

Complex output fuction displays value as a+bi or a-bi

Displays only a if imaginary part is 0

Display only bi if real part is 0 */

```
void print_complex(complex_t c) /* input-complex number to
display */
```



```
/*Returns sum of complex value c1 and c2*/
complex_t add_complex(complex_t c1,complex_t c2) /*input*/

/*Returns difference c1-c2*/
complex_t subtract_complex(complex_t c1,complex_t c2) /*input*/

/*Returns product of complex values c1 and c2*/
complex_t multiply_complex(complex_t c1,complex_t c2)
/*input*/

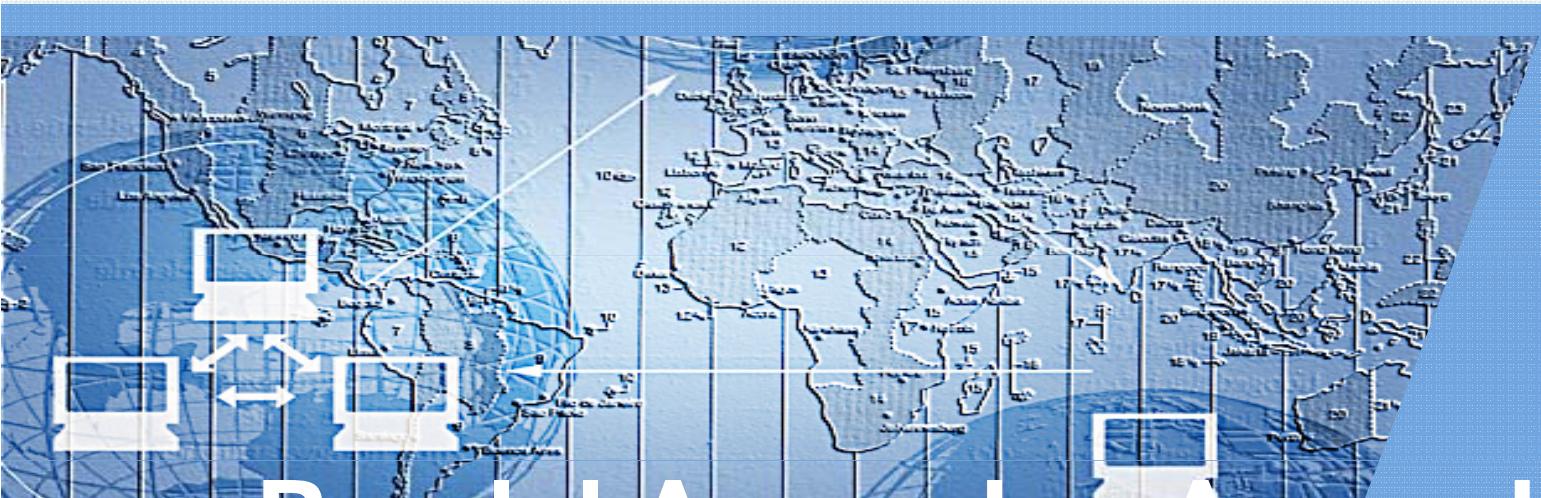
/*Returns quotient of complex values(c1/c2)*/
complex_t divide_complex(complex_t c1,complex_t c2) /*input*/

/*Returns absolute value of complex number c*/
abs_complex(complex_t c) /*input*/
```



Implementasi

- ❖ Hasil Implementasi dapat dilihat di buku referensi halaman 559-563



Paralel Array dan Array dari Struktur

Paralel Array

- ❖ Contoh representasi koleksi data menggunakan paralel array :

```
int np[50];  
double nilai[50];
```

array np dan nilai disebut paralel array karena isi data dengan index yang sama diperuntukkan bagi mahasiswa pada index yang sama.

Array dari struktur

- ❖ Dengan menggunakan array dari struktur contoh data nilai mahasiswa akan menjadi lebih terorganisasi dibandingkan dengan menggunakan paralel array.

Contoh penggunaan array dari struktur

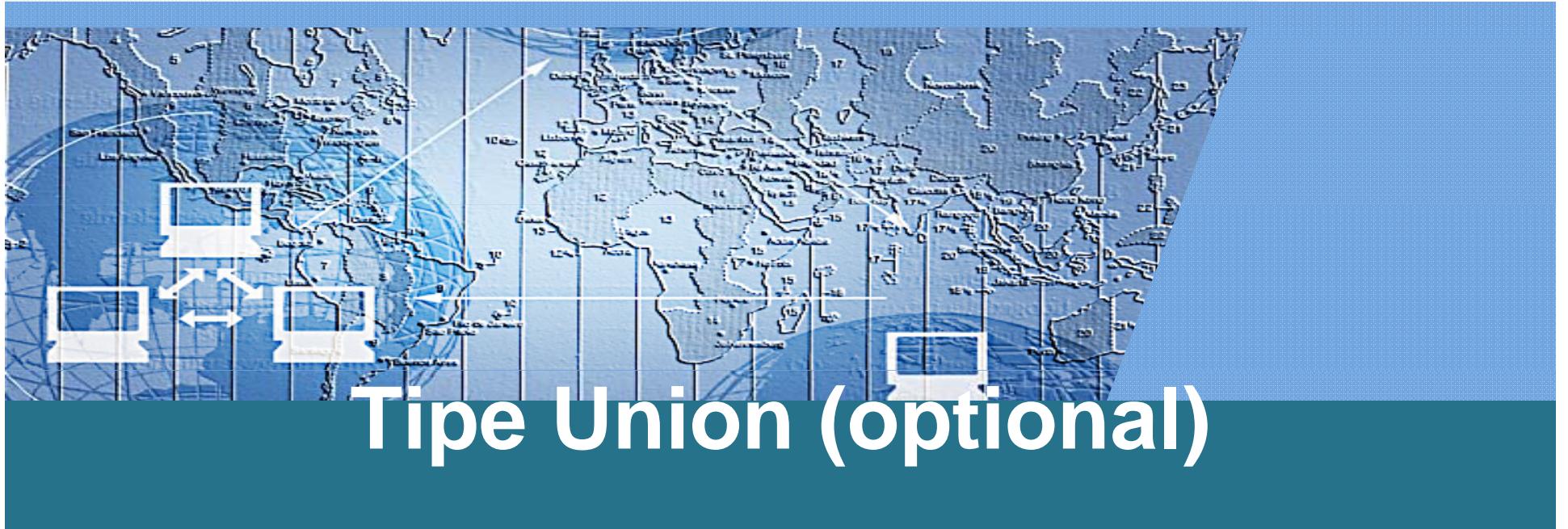
```
typedef struct {  
    int nrp;  
    double nilai;  
}nilai;
```

```
nilai mahasiswa[50];
```

Contoh

- ❖ Data mahasiswa pertama dimasukkan ke dalam struktur mahasiswa[0].
 - Data nrp dimasukkan dalam mahasiswa[0].nrp
 - Data nilai dimasukkan dalam mahasiswa[0].nilai

| | nrp | nilai |
|---------------|------------|-------|
| mahasiswa[0] | 5106100001 | 8.01 |
| mahasiswa[1] | 5106100002 | 8.02 |
| mahasiswa[2] | 5106100003 | 8.03 |
| mahasiswa[] | | |
| mahasiswa[49] | 5106100049 | 8.49 |



Type Union (optional)

- ❖ Union memungkinkan suatu lokasi memori ditempati oleh 2 atau lebih variabel dengan tipe data berlainan
- ❖ Contoh cara mendefinisikan
union bil_bulat
 { unsigned int di ;
 unsigned char dc[2] ;
 };

Union (cont..)

- ❖ Cara mengakses elemen dari sebuah union:
`variabel_union.nama elemen`
- ❖ Seperti halnya pada struktur, variabel union juga dapat dilewatkan ke dalam suatu fungsi sebagai parameter

Contoh program Union

```
/* Contoh pendeklarasian dan pengaksesan variabel union */
```

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

```
main()
```

```
{
```

```
union{
```

```
    unsigned int di;
```

```
    unsigned char dc[2];
```

```
 }bil_x; //varibel union
```

```
bil_x.di =321 ;
```

```
printf("di=%d dc[0] =%d dc [1] = %d \n",bil_x.di,
bil_x.dc , bil_x.dc[1]);
```

```
}
```

Contoh hasil eksekusi :

di=321 dc[0] =65 dc [1] =1

Contoh program mengubah nilai variabel union melalui fungsi

```
#include <stdio.h>
union bil_bulat
{
    unsigned int di ;
    unsigned char dc[2] ;
}; //definisi tipe union
void beri_nilai(union bil_bulat *x); //deklarasi fungsi
main()
{
    union bil_bulat bil_x ; //deklarasi variabel union
    beri_nilai(&bil_x);
    printf("di=%d dc [0] = %d dc [1] = %d \n",bil_x.di,bil_x.dc[0],bil_x.dc[1]);
    void beri_nilai(union bil_bulat *x);
    {
        x->di=321; //elemen yang ditunjuk oleh x diberi nilai 321
    }
}
```

Contoh hasil eksekusi :

di=321 dc [0] = 65 dc [1] = 1

Referensi

- ❖ Bab 11, “Structure and Union Types”, *Problem Solving and Program Design in C*, Jeri R. Hanly dan Elliot B. Koffman, Addison Wesley, 2002