

Typedef dan Struct

Pendahuluan

Memahami konsep typedef dan struct. Perintah *typedef* berguna untuk membuat alias dari suatu tipe data. *Struct* merupakan perintah yang dapat digunakan untuk membuat struktur dari suatu record. Sebuah record bisa memiliki domain yang heterogen. Sebagai contoh, sebuah record yang merepresentasikan tentang seseorang memiliki atribut berupa nama yang bertipe string, umur yang bertipe integer, dan tinggi yang bertipe double. Biasanya penggunaan *typedef* dikombinasikan dalam pendeklarasian sebuah *struct*.

Objektif

- Memperkenalkan penggunaan typedef
- Memperkenalkan penggunaan struct
- Menunjukkan cara mendeklarasi struct yang dikombinasikan dengan perintah typedef

Bahan Bacaan

Buku pegangan atau bahan lain yang menjelaskan tentang typedef dan struct dalam bahasa pemrograman C.

Typedef

Typedef

Perhatikan dan pahami beberapa contoh di bawah ini:

```
typedef int Time;
Time hours, minutes, seconds;
```

Pada contoh di atas, **Time** merupakan alias dari tipe data primitif **int**. Selanjutnya, pendeklarasian variabel **hours**, **minutes** dan **seconds** dapat secara langsung menggunakan aliasnya saja, bukan lagi menggunakan tipe data primitif **int**.

Sama halnya juga terjadi pada contoh berikut ini:

```
typedef char * String;
String namaMhs, alamatMhs;
```

Dalam C, pointer ke character merupakan satu-satunya cara mendeklarasi tipe data string. Pada contoh di atas, dengan menggunakan *typedef*, sebuah tipe data baru dengan nama **String** dideklarasikan. Kemudian, menggunakan tipe data **String** yang baru tersebut, variabel **namaMhs** dan **alamatMhs** dideklarasikan.

Typedef Pada Tipe Data Array

Typedef Pada Array

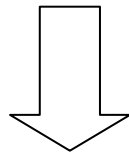
Pendeklarasian sebuah array bertipe int dapat dilakukan seperti yang terlihat berikut ini.

```
int x[100]
```

Alias dari array bertipe int berukuran 100 di atas dapat dilakukan menggunakan typedef sebagai berikut:

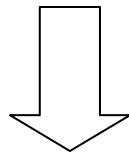
Langkah 1: Ganti variabel x dengan nama alias, misalnya Larik sehingga menjadi:

```
int Larik[100];
```



Langkah 2: Tambahkan statemen **typedef** sehingga menjadi

```
typedef int Larik[100];
```



Langkah 3: Deklarasi variabel dengan tipe Larik ukuran 100 adalah:

```
Larik a, b, c;
```

Typedef Dengan Contoh

Contoh 1:

```
#define N 3
typedef float Scalar;
typedef Scalar Vector[N];

void addVector(Vector a, Vector b, Vector c) {
    int i;
    for(i=0; i<N; ++i)
        c[i] = a[i] + b[i];
}
```

Contoh 2:

```
#define N 3
typedef float Scalar;
typedef Scalar Matrix[N][N];

void addMatrix(Matrix a, Matrix b, Matrix c) {
    int i,j;
    for(i=0; i<N; ++i)
        for(j=0; j<N; ++j)
            c[i][j] = a[i][j] + b[i][j];
}
```

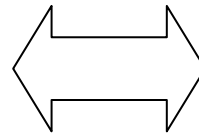
Struct

Struct

Struct atau *structure* dalam C digunakan untuk membentuk tipe data dengan anggota (*member*) bertipe tertentu. Cara mendeklarasi sebuah struct adalah sebagai berikut:

```
struct tgl{
    int hari;
    int bulan;
    int tahun;
};

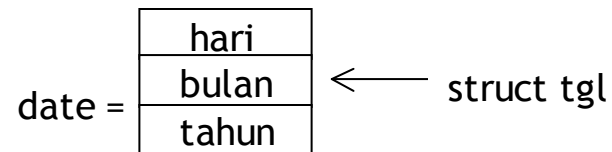
typedef struct tgl Tanggal;
Tanggal date;
```



```
typedef struct tgl{
    int hari;
    int bulan;
    int tahun;
}Tanggal;

Tanggal date;
```

Bila divisualisasikan kira-kira sebagai berikut:



Struct (Lanjutan)

Penjelasan Pendeklarasian Struct

Contoh pada halaman sebelumnya memperlihatkan bagaimana sebuah struct dengan nama **struct tgl** yang memiliki tiga member bertipe **int** yaitu **hari**, **bulan** dan **tahun** dideklarasikan. Kemudian, **struct tgl** tersebut dibuat aliasnya dengan nama **Tanggal** menggunakan perintah *typedef*. Selanjutnya, variabel **date** yang bertipe alias **Tanggal** dideklarasikan.

Untuk mengakses anggota (*member*) dari sebuah struct digunakan operator *titik*. Contoh

```
date.hari = 2;  
date.tahun = 2001;
```

Mengakibatkan member dari struct **date** di atas memiliki nilai sebagai berikut ini:

date =	2
	?
	2001

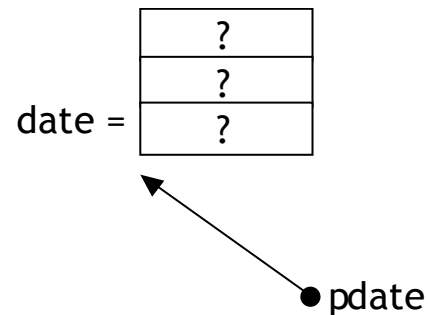
Pointer ke Struct

Pointer ke Sebuah Struct

Pendeklarasian sebuah pointer ke **struct tgl** dilakukan dengan cara sebagai berikut:

```
Tanggal date;  
Tanggal * pdate = &date;
```

Karena **pdate** merupakan pointer ke alias **Tanggal** dan **date** adalah sebuah variabel bertipe **Tanggal**, maka perintah **pdate = &date;** mengakibatkan pointer **pdate** tunjuk ke alamat dari variabel **date**.



Untuk mengakses member dalam struct **date** menggunakan pointer **pdate** di atas, digunakan operator panah (**->**) sebelum nama member. Contoh:

```
pdate->hari = 2  
pdate->bulan = 11  
pdate->tahun = 1999;
```