



PERTEMUAN VIII

STRUKTUR LANJUT



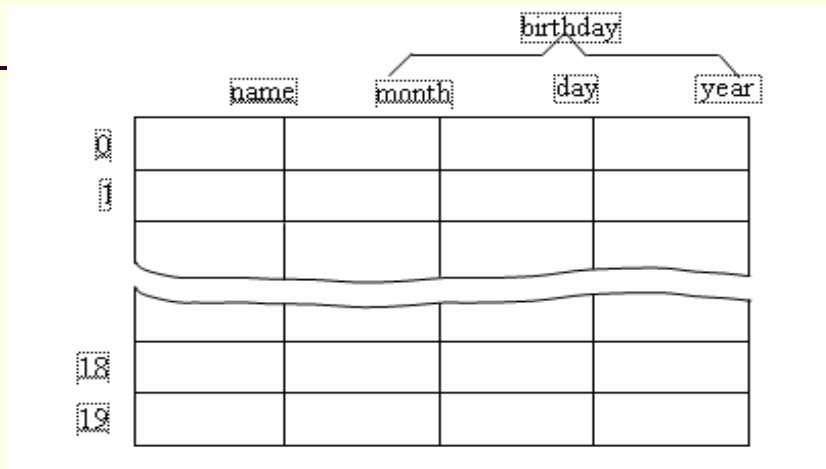
ARRAY & STRUKTUR

- Elemen-elemen dari suatu array juga dapat berbentuk sebuah struktur
- Misalnya array yang dipakai untuk menyimpan sejumlah data siswa (*struct student*) → akan membentuk tabel, dimana barisnya sbg elemen dari array dan kolomnya sbg elemen dari struktur.
- Contoh Deklarasi :

```
#define MAKS 20
struct date {                                /* definisi dari tipe date */
    int month;
    int day;
    int year;};
struct person {                               /* definisi dari tipe person */
    char name[30];
    struct date birthday;};
/* deklarasi dari variabel array student */
struct person student [MAKS];
```

Artinya : mendeklarasikan array **student** yang memiliki elemen yang bertipe *struct person* sebanyak **MAKS**.

ARRAY & STRUKTUR – Cont.



Gambar Array dari Struktur

Cara input data :

```
for (i=0; i<MAKS; i++)
{
    printf("Name      : ");
    fgets(student[i].name, sizeof student[i].name, stdin);
    printf("Birthday (mm-dd-yyyy): ");
    scanf("%d-%d-%d", &student[i].birthday.month,
          &student[i].birthday.day,
          &student[i].birthday.year);
    printf("\n");
    /* hapus sisa data dalam penampung keyboard */
    fflush(stdin);
};
```

Cara output data :

```
printf("DATA SISWA\n");
for (i=0; i<jml; i++)
{
    printf("%d. Name      : %s\n", i+1,
          student[i].name);
    printf("  Birthday : %d-%d-%d\n\n",
          student[i].birthday.month,
          student[i].birthday.day,
          student[i].birthday.year );
};
```

PENDEKLARASIAN STRUKTUR

- Cara lain dalam pendeklarasian struktur :

```
struct date {  
    int month, day, year;  
} today, tomorrow;
```

```
struct person {  
    char name[30];  
    struct date birthday;  
} student[MAKS];
```

```
struct date {  
    int month, day, year;  
} today = {5,14,2001};
```

STRUKTUR & FUNGSI

A. Melewatkan Elemen Struktur ke dalam Fungsi

Dapat dilakukan baik secara nilai (*pass by value*) maupun secara acuan (*pass by reference*).

```
#include <stdio.h>
void cetak_tanggal(int mm, int dd, int yy);
main()
{
    struct date {
        int month;
        int day;
        int year;
    } today;
    printf("Enter the current date (mm-dd-yyyy): ");
    scanf("%d-%d-%d", &today.month, &today.day, &today.year);
    cetak_tanggal(today.month, today.day, today.year);
}
void cetak_tanggal(int mm, int dd, int yy)
{
    static char *nama_bulan[] = {
        "Wrong month", "January", "February", "March", "April", "May", "June",
        "July", "August", "September", "October", "November", "December";
    };
    printf("Todays date is %s %d, %d\n\n", nama_bulan[mm], dd, yy);
}
```

Pass by value

Enter the current date (mm-dd-yyyy): 5-29-2001
Todays date is May 29, 2001

STRUKTUR & FUNGSI – Cont.1

```
#include <stdio.h>
void tukar_xy(int *a, int *b);
main()
{
    struct koordinat {
        int x;
        int y;
    } posisi;
    printf("Masukkan koordinat posisi (x, y) : ");
    scanf("%d, %d", &posisi.x, &posisi.y);
    printf("x, y semula = %d, %d\n", posisi.x, posisi.y);
    tukar_xy(&posisi.x, &posisi.y);
    printf("x, y sekarang = %d, %d\n", posisi.x, posisi.y);
}
void tukar_xy(int *a, int *b)
{
    int z;
    z = *a;
    *a = *b;
    *b = z;
}
```

Pass by reference

operator alamat ditempatkan di depan nama variabel struktur (bukan di depan nama elemen struktur).

```
Masukkan koordinat posisi (x, y) : 34, 21
x, y semula = 34, 21
x, y sekarang = 21, 34
```

STRUKTUR & FUNGSI – Cont.2

B. Melewatkan Struktur ke dalam Fungsi

Parameter fungsi diubah menjadi bentuk struktur.

```
#include <stdio.h>
struct date {          /* definisi global dari tipe date */
    int month;
    int day;
    int year;
};
void cetak_tanggal(struct date now);
main()
{
    struct date today;
    printf("Enter the current date (mm-dd-yyyy): ");
    scanf("%d-%d-%d", &today.month, &today.day, &today.year);
    cetak_tanggal(today);
}
void cetak_tanggal(struct date now)
{
    static char *nama_bulan[] = {
        "Wrong month", "January", "February", "March", "April", "May", "June",
        "July", "August", "September", "October", "November", "December" };
    printf("Todays date is %s %d, %d\n\n", nama_bulan[now.month], now.day, now.year);
}
```

Enter the current date (mm-dd-yyyy): 5-29-2001
Todays date is May 29, 2001

Struktur dan Pointer (Pointer ke Struktur)

- Jika sebuah struktur mengandung banyak *field* dan diputuskan bahwa keseluruhan *field*-nya akan diubah oleh fungsi, maka cara yang efisien adalah dengan melewati (*passing*) alamat dari struktur.
- Pada pendefinisian fungsi, parameter formalnya berupa pointer yang menunjuk ke struktur.
- Untuk contoh, lihat kembali contoh struktur dan fungsi :

```
void tukar_xy(struct koordinat *pos_xy)
{
    int z;
    z = (*pos_xy).x;
    (*pos_xy).x = (*pos_xy).y;
    (*pos_xy).y = z;
}
```

Pada definisi fungsi di atas,

```
struct koordinat *pos_xy
```

menyatakan bahwa **pos_xy** adalah pointer yang menunjuk ke obyek bertipe struktur **koordinat**.

Struktur dan Pointer – Cont. 1

Penulisan :

```
( *pos_xy ) . x
```

menyatakan : elemen bernama **x** yang ditunjuk oleh pointer **pos_xy**

- Tanda kurung di (*pos_xy).xy harus ada sebab :

```
(*pos_xy.xy) ≠ *pos_xy.xy
```

Ungkapan ***pos_xy.x** mempunyai makna yaitu :

"yang ditunjuk oleh **pos_xy.x** "

(sebab operator titik mempunyai prioritas yang lebih tinggi daripada operator *).

Struktur dan Pointer – Cont. 2

- Bentuk semacam :

`(*pos_xy).x`

dapat ditulis dengan bentuk lain menjadi

`pos_xy->x`

Sehingga untuk program `tukar_xy` dapat ditulis ulang sbb :

```
void tukar_xy(struct koordinat *pos_xy)
{
    int z;
    z = pos_xy->x;
    pos_xy->x = pos_xy->y;
    pos_xy->y = z;
}
```

CONTOH PROGRAM

```
#include <stdio.h>
struct koordinat
    { int x;
      int y;
    };
void tukar_xy(struct koordinat *pos_xy);
main()
{
    struct koordinat posisi;
    printf("Masukkan koordinat posisi (x, y) : ");
    scanf("%d, %d", &posisi.x, &posisi.y);
    printf("x, y semula = %d, %d\n", posisi.x, posisi.y);
    tukar_xy(&posisi);
    printf("x, y sekarang = %d, %d\n", posisi.x, posisi.y);
}
void tukar_xy(struct koordinat *pos_xy)
{
    int z;
    z = (*pos_xy).x;
    (*pos_xy).x = (*pos_xy).y;
    (*pos_xy).y = z;
}
```

Masukkan koordinat posisi (x, y) : 34, 21
x, y semula = 34, 21
x, y sekarang = 21, 34

LATIHAN

1. Definisikan sebuah struktur (misalkan namanya = **data_toko**) yang memiliki 3 buah *field* berupa : sebuah array karakter dengan 5 elemen (misalkan namanya = **nama**), sebuah integer (misalkan namanya = **data**), dan sebuah *float* (misalkan namanya = **harga**). Inputkan data & tampilkan hasilnya !
2. Buat program untuk menghitung luas persegi panjang, dengan variabel-variabelnya didefinisikan dalam sebuah struktur (ada 3 field yaitu int panjang; int lebar & int luas).
3. Buatlah definisi struktur dan programnya dari diagram struktur data berikut ini :

