

# PRAKTIKUM 31

---

## Struct 3

---

### A. TUJUAN PEMBELAJARAN

1. Memahami penggunaan Struktur pada pointer (*pointer to struct*)

### B. DASAR TEORI

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. Dengan demikian pada pendefinisian fungsi, parameter formalnya berupa pointer yang menunjuk ke struktur.

Masalah pointer ke struktur dapat diterapkan dalam program sebelumnya. Argumen dari fungsi `tukar_xy()` dapat disederhanakan menjadi satu argumen saja, yakni sebagai berikut :

```
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**. Adapun penulisan :

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

menyatakan : elemen bernama **x** yang ditunjuk oleh pointer **pos\_xy**

Perlu diperhatikan bahwa penulisan tanda kurung seperti pada contoh `(*pos_xy).x` merupakan suatu keharusan.

Sebab

```
*pos_xy.x
```

mempunyai makna yang berbeda dengan

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

Ungkapan `*pos_xy.x` mempunyai makna yaitu : "yang ditunjuk oleh `pos_xy.x` " (sebab operator titik mempunyai prioritas yang lebih tinggi daripada operator \*).

---

```
#include <stdio.h>

struct koordinat
{
    int x;
    int y;
};

void tukar_xy(struct koordinat *);

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;
}
```

### **Contoh eksekusi :**

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

---

Bentuk semacam :

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

dapat ditulis dengan bentuk lain menjadi

```
pos_xy->x
```

Dalam C operator `->` (berupa tanda minus - diikuti dengan tanda lebih dari `>`) disebut sebagai **operator panah**. Dengan menggunakan operator panah, maka fungsi **tukar\_xy()** dalam program **posisi2.c** dapat ditulis menjadi

```
void tukar_xy(struct koordinat *pos_xy){
    int z;

    z = pos_xy->x;
    pos_xy->x = pos_xy->y;
    pos_xy->y = z;
}
```

### C. PERCOBAAN

1. Untuk semua program yang ada di modul teori Struktur 2 :

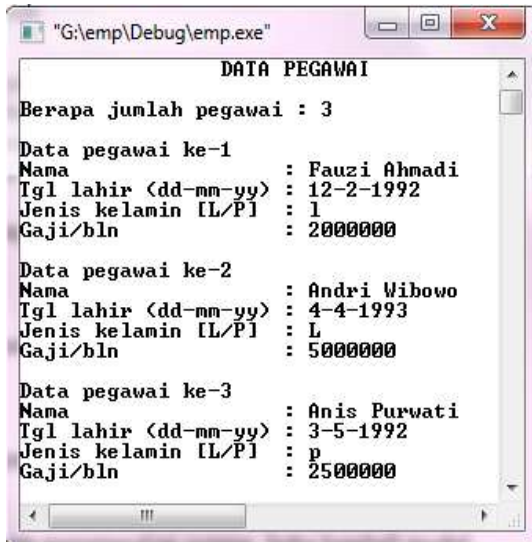
- Ketikkan kembali program-program tsb
- Jalankan programnya
- Analisis & buatlah kesimpulan terhadap program-program tsb

2. Dengan menggunakan typedef, buatlah tipe data struct untuk menyimpan informasi pegawai (misal tipe pegawai) berupa : no ID, nama, tgl lahir, jenis kelamin (L/P) dan gaji/bulan.

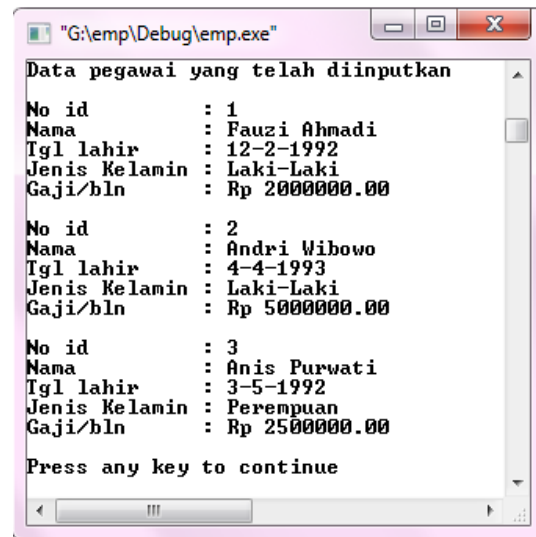
- di `main()` deklarasikan sebuah var bertipe array of pegawai (misal `emp[]`)
- panggil fungsi `input()` untuk memasukkan data-datanya dengan pengiriman parameter secara *pass by reference*
- panggil fungsi `tampil()` untuk menampilkan semua data yang telah diinputkan dengan pengiriman parameter secara *pass by reference*

### Petunjuk :

- Untuk no id, gunakan auto increment (tanpa input) → perhatikan contoh tampilan
- Jika diperlukan, gunakan fungsi strcpy( ) dari <string.h>
- Contoh input bisa dilihat pada gambar 31.1 dengan output pada gambar 31.2



Gambar 31.1 Contoh Input Soal no.2



Gambar 31.2 Contoh Output Soal no.2

### 3. Modifikasi program dari modul STRUCT 2 soal no 1.

- Ubah nama tipe data dengan menggunakan typedef
- Panggilah fungsi input( ) dan tampil( ) dengan cara *pass by reference*

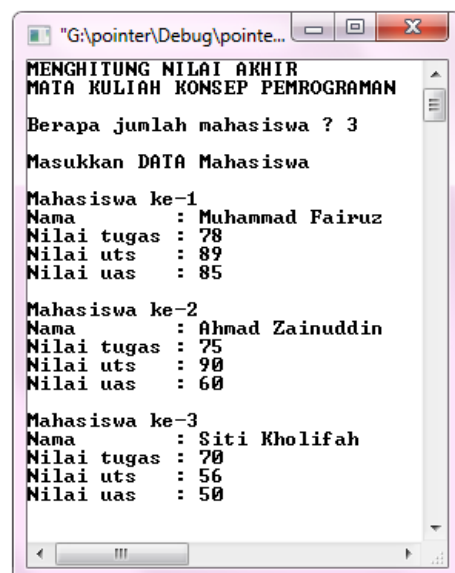
### Petunjuk :

- Untuk soal nomor 2 & 3, perhatikan cara membaca array dengan menggunakan pointer, buka kembali modul teori *pointer to array*

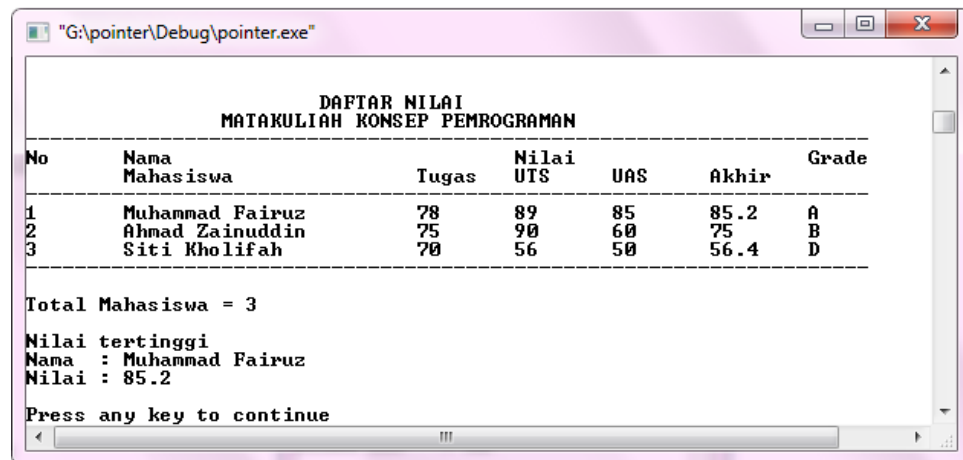
### 4. Untuk soal no 3 diatas, pada tampilan output, tambahkan informasi : nama & nilai akhir mahasiswa yang tertinggi (lihat contoh)

### Petunjuk:

- buatlah tipe data struct baru misal dengan nama maks dengan 2 field : nama & nilai untuk menyimpan informasi mahasiswa dengan nilai tertinggi



- Sertakan `<string.h>` agar bisa menggunakan fungsi `strcpy()` untuk mengcopy nama
- Contoh input bisa dilihat pada gambar 31.3 dengan output pada gambar 31.4



Gambar 31.4 Contoh Output Soal no.3

#### D. LAPORAN RESMI

1. Untuk setiap listing program dari percobaan-percobaan di atas, ambil *capture* outputnya.
2. Tuliskan kesimpulan dari percobaan yang telah anda lakukan.