Praktikum 11 (3/3)
Struktur

A. TUJUAN

1. Mengerti tentang 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 melewatkan *(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. Buatlah program sebagai berikut dengan menggunakan array dari struktur:

   Input:

   Nama barang

   Jumlah

   Harga Satuan

   sebanyak n barang dapat di tentukan oleh user

2. Dengan menggunakan fungsi dan pointer buatlah fasilitas untuk menampilkan Nama Barang yang di inputkan, jika tidak ada maka tampilkan pesan!

3. Dengan menggunakan fungsi dan pointer buatlah fasilitas untuk menampilkan data yang memmiliki nilai jumlah terbesar!

4. Dengan menggunakan fungsi dan pointer buatlah fasilitas untuk menampilkan barang dengan nilai rupiah tertinggi

D. LAPORAN

1. Tulis listing program dari semua percobaan yang dilakukan.

2. Kemudian tuliskan outputnya. Terangkan kenapa demikian.

3. Dengan menggunakan fungsi dan pointer serta melewatkan struktur pada fungsi, urutkan array dari fungsi tersebut berdasarkan Nilai jual!