

# PRAKTIKUM 29

## Struct 1

### A. Dasar Teori

Struktur adalah pengelompokan variabel-variabel yang bernaung dalam satu nama yang sama. Berbeda dengan array yang berisi kumpulan variabel -variabel yang bertipe sama dalam satu nama, maka suatu struktur dapat terdiri atas variabel -variabel yang berbeda tipenya dalam satu nama struktur. Struktur biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi sebuah.

Variabel-variabel yang membentuk suatu struktur, selanjutnya disebut sebagai elemen dari struktur atau field. Dengan demikian dimungkinkan suatu struktur dapat berisi elemen-elemen data berbeda tipe seperti char, int, float, double, dan lain -lain.

Contoh sebuah struktur adalah informasi data tanggal (date) yang berisi:

- day, month dan year

### Mendefinisikan & Mendeklarasikan Struktur

Suatu struktur didefinisikan dengan menggunakan kata kunci struct. Contoh pendefinisian sebuah tipe data struktur :

```
struct date {
    int month;
    int day;
    int year;
};
```

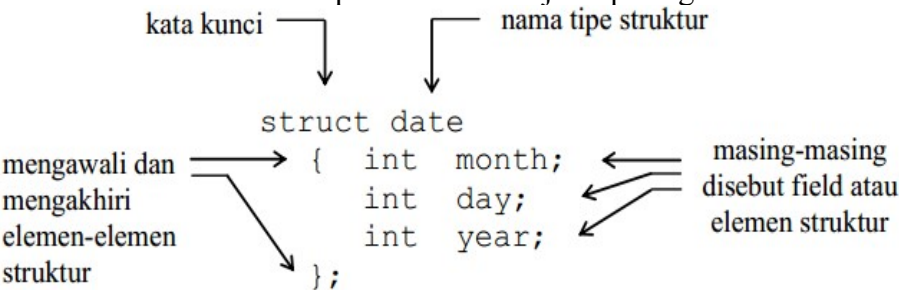
Karena semua field-nya bertipe sama, bisa juga ditulis sbb :

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

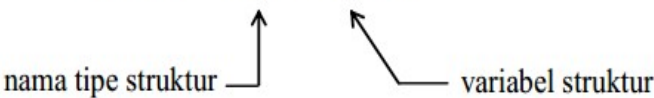
yang mendefinisikan sebuah tipe data struktur bernama date yang memiliki tiga buah elemen (field) berupa :

- day, month dan year

Keterangan untuk definisi struktur ini dapat dilihat lebih jelas pada gambar 29.1.



Selanjutnya untuk mendeklarasikan sebuah variabel `today` yang bertipe struktur `date` pernyataan yang diperlukan seperti diilustrasikan pada gambar 29.2.



gambar 29.2 Pendeklarasian variabel bertipe struktur

Pernyataan di atas menyatakan bahwa variabel `today` bertipe struktur `date`

Mengakses Elemen Struktur

Elemen dari suatu variabel struktur dapat diakses dengan menyebutkan nama variabel struktur diikuti dengan operator titik (‘.’) dan nama dari elemen strukturnya. Cara penulisannya sebagai berikut :

```
variabel_struktur.nama_field
```

Menginisialisasi Struktur

Sebuah struktur juga bisa diinisialisasi pada saat dideklarasikan. Hal ini serupa dengan inisialisasi array, yaitu elemen-elemennya dituliskan di dalam sepasang kurung kurawal (‘{ }’) dengan masing-masing dipisahkan dengan koma. Deklarasi struktur didahului dengan kata kunci static, contoh

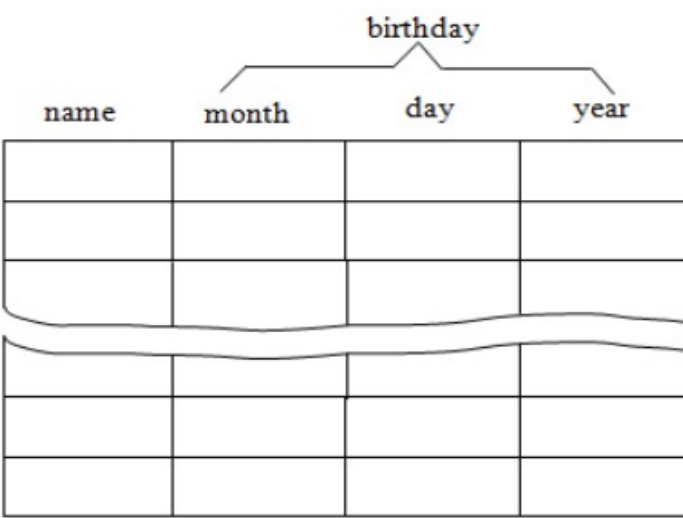
```
static struct zodiak bintang = {"Sagitarious", 22, 11, 21, 12};
```

Array dan Struktur

Elemen-elemen dari suatu array juga dapat berbentuk sebuah struktur. Misalnya array yang dipakai untuk menyimpan sejumlah data siswa (struct student). Array struktur berdimensi satu ini membentuk suatu tabel, dengan barisnya menunjukkan elemen dari array-nya dan kolomnya menunjukkan elemen dari struktur. Dalam hal ini maka deklarasi yang dibutuhkan adalah sebagai berikut :

```
#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];
```

yang artinya, mendeklarasikan array student yang memiliki elemen yang bertipe struct person sebanyak MAKS. Setelah array student dideklarasikan, maka ruang yang disediakan ditunjukkan dalam gambar 29.4.



Gambar 29.4 Array dari struktur

## B. Percobaan

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

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

a. Listing Program

```
#include <stdio.h>
#define MAKS 20

struct date{
    int month, day, year;
};

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

main() {
    struct student data_mhs[MAKS];
    int i=0, jml;
    char lagi;

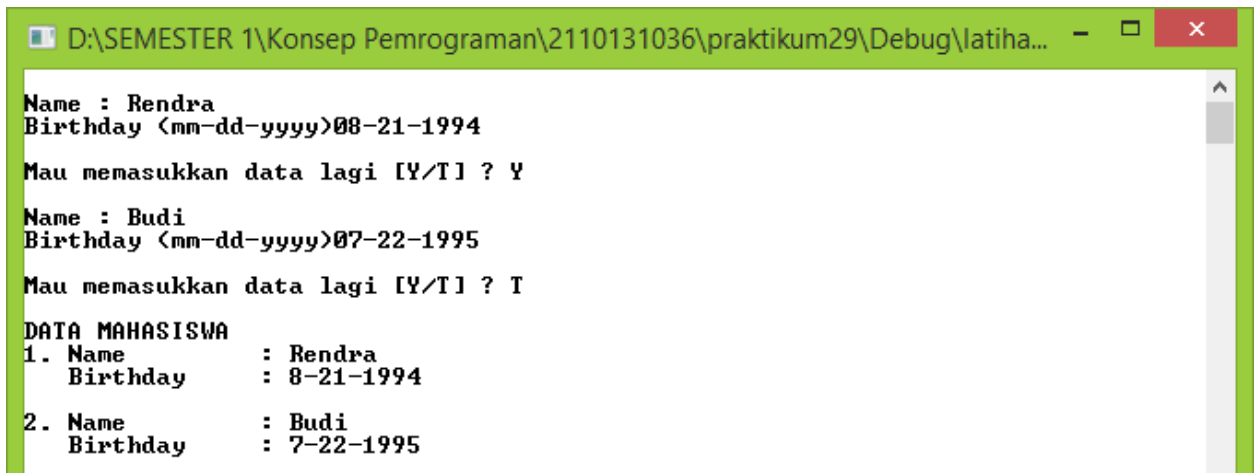
    do
    {
        fflush(stdin);          //hapus sisa data dalam buffer keyboard
        printf("\nName : ");
        gets(data_mhs[i].name);

        fflush(stdin);          //hapus sisa data dalam buffer keyboard
        printf("Birthday (mm-dd-yyyy)");
        scanf("%d-%d-
%d",&data_mhs[i].birthday.month,&data_mhs[i].birthday.day,&data_mhs[i].birthday.year)
        ;

        i++;
        fflush(stdin);          //hapus sisa data dalam buffer keyboard
        printf("\nMau memasukkan data lagi [Y/T] ? ");
        lagi = getchar();       //hapus susa data dalam buffer keyboard
    } while(lagi == 'Y' || lagi == 'y');

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

## b. Capture Program



```
D:\SEMESTER 1\Konsep Pemrograman\2110131036\praktikum29\Debug\latiha...
Name : Rendra
Birthday <mm-dd-yyyy>08-21-1994
Mau memasukkan data lagi [Y/T] ? Y
Name : Budi
Birthday <mm-dd-yyyy>07-22-1995
Mau memasukkan data lagi [Y/T] ? T
DATA MAHASISWA
1. Name      : Rendra
   Birthday   : 8-21-1994
2. Name      : Budi
   Birthday   : 7-22-1995
```

## c. Analisis Program

Program ini adalah program untuk mendata tanggal lahir seseorang. Dalam prosesnya, digunakanlah struct yang berada pada struct. Struct pertama adalah struct untuk menyimpan data mahasiswa yaitu berupa nama(name[30]) dan tanggal lahir(date). Tanggal lahir(date) sendiri sebenarnya adalah sebuah struct tersendiri yang memiliki bagian-bagian sendiri pula yaitu bulan(month), hari(day) dan tahun(year).

Pertama untuk meminta input data, program melakukannya secara berulang atau looping menggunakan do while dikarenakan proses input data pasti terjadi minimal sekali. Selanjutnya untuk menampilkan data yang telah diinputkan ke struct tadi, digunakanlah looping for sebanyak jumlah data yang ada pada struct.

2. Buatlah 2 variabel (today & tomorrow) bertipe struct date. Isilah variabel today dengan tanggal hari ini. Temukan tanggal untuk variabel tomorrow.

Keterangan:

- Berikan pengecekan untuk akhir bulan
- Berikan pengecekan untuk akhir tahun
- Berikan pengecekan untuk bulan Februari di tahun kabisat

Tahun kabisat adalah :

- Kelipatan 4, tapi bukan kelipatan 100
- Kelipatan 100 yang juga merupakan kelipatan 400

## a. Listing Program

```
#include <stdio.h>
#define MAKS 20

struct date{
    int day, month, year;
};

int cariJumHari(int month,int year);

main() {

    struct date today,tomorrow;
    int jmlHari = 30;
    char lagi;

    printf("\nPROGRAM PENCARI TANGGAL BESOK");
    printf("\n=====");
```

```

do
{
    printf("\nMasukkan tanggal hari ini (mm-dd-yyyy)\t: ");
    scanf("%d-%d-%d",&today.day,&today.month,&today.year);
    if(today.day + 1 > cariJumHari(today.month,today.year))
    {
        if(today.month == 12)
        {
            //Ganti tahun
            tomorrow.day = 1;
            tomorrow.month = 1;
            tomorrow.year= today.year + 1;
        }
        else
        {
            //Ganti bulan
            tomorrow.day = 1;
            tomorrow.month = today.month + 1;
            tomorrow.year= today.year;
        }
    }
    else
    {
        tomorrow.day = today.day + 1;
        tomorrow.month= today.month;
        tomorrow.year= today.year;
    }

    printf("Maka tanggal besok adalah \t\t: %2d-%2d-%d",tomorrow.day,tomorrow.month,tomorrow.year);
    fflush(stdin); //hapus sisa data dalam buffer keyboard
    printf("\n\nMau memasukkan tanggal lagi [Y/T] ? ");
    lagi = getchar(); //hapus sisa data dalam buffer keyboard
} while(lagi == 'Y' || lagi == 'y');
}

int cariJumHari(int bulan,int tahun)
{
    int daftarHari[] = {31,28,31,30,31,30,31,31,30,31,30,31};
    int jumHari = daftarHari[bulan-1];

    if(bulan == 2 && tahun % 4 == 0)
    {
        if(tahun % 100 != 0 || tahun % 400 == 0)
            jumHari = 29;
    }

    return jumHari;
}

```

## b. Capture Program

```

D:\SEMESTER 1\Konsep Pemrograman\2110131036\praktikum29\Debug\soal1....
PROGRAM PENCARI TANGGAL BESOK
=====
Masukkan tanggal hari ini <mm-dd-yyyy> : 31-12-2013
Maka tanggal besok adalah : 1- 1-2014
Mau memasukkan tanggal lagi [Y/T] ? Y
Masukkan tanggal hari ini <mm-dd-yyyy> : 28-02-2012
Maka tanggal besok adalah : 29- 2-2012
Mau memasukkan tanggal lagi [Y/T] ? Y
Masukkan tanggal hari ini <mm-dd-yyyy> : 28-02-1900
Maka tanggal besok adalah : 1- 3-1900
Mau memasukkan tanggal lagi [Y/T] ? T

```

c. Analisis Program

Program pencari tanggal esok hari menggunakan struct date yang digunakan sebagai penampung tanggal hari ini(today) dan tanggal besok(tomorrow). Pada dasarnya program ini cukup mudah karena kita hanya perlu menambahkan 1 di struct date today yang bagian hari(day). Namun hal tersebut menjadi agak rumit apabila tanggal yang dimasukkan adalah tanggal di akhir bulan seperti 30 atau 31.

Karena itu, perlu diketahui dulu berapa jumlah hari dalam bula tersebut. Disini dibuatlah fungsi untuk mencari jumlah hari dalam satu bulan yaitu fungsi cariJumHari(). Apabila bulan yang dimasukkan adalah bulan 2(Februari), maka perlu juga dicek apakah tahun yang dimasukkan adalah tahun kabisat atau bukan sesuai kriteria pada soal karena bulan Februari jumlahnya berubah-ubah sesuai tahun kabisat tersebut.

3. Sebuah Perusahaan Ayam goreng dengan nama Gerobak Fried Chicken yang telah lumayan banyak pelanggannya, ingin dibantu dibuatkan program demi kelancaran usahanya. Gerobak Fried Chicken mempunyai Daftar Harga Ayam sebagaimana disamping :

Kode	Jenis	Harga
D	Dada	Rp. 5.000,-
P	Paha	Rp. 4.000,-
S	Sayap	Rp. 3.000,-

Ketentuannya : setiap pembeli dikenakan pajak sebesar 10% dari total pembelian

Petunjuk :

- Gunakan tipe data struct
- Deklarasikan variabel bertipe array of struct
- Gunakan \t sebagai tabulasi untuk mengatur tampilan
- Gunakan <string.h> jika diperlukan

a. Listing Program

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

struct datamenu {
    char kode;
    int jumlah;
};

main()
{
    struct datamenu keranjang[30];
    int i = 0,j,harga,total=0;
    char jwb,jenis[6];

    printf("Toko GFC-Gerobak Fried Chicken\n");
    printf("\nDAFTAR HARGA");
    printf("\nKode    Jenis    Harga Per Potong");
    printf("\nD        Dada    Rp. 5000.00");
    printf("\nP        Paha    Rp. 4000.00");
    printf("\nS        Sayap   Rp. 3000.00");
    printf("\n\nHarga belum termasuk pajak : 10 %");
    printf("\n\nMasukkan Pesanan Anda");

    do
    {
        printf("\nPesanan ke-%d",i+1);
        fflush(stdin);
        printf("\nJenis [D/P/S]\t: ");
        scanf("%c",&keranjang[i].kode);
        fflush(stdin);
        printf("Jumlah \t\t: ");
        scanf("%d",&keranjang[i].jumlah);
```

```

        i++;
        fflush(stdin);
        printf("\nMau pesan lagi [y/n] ?");
        jwb = getchar();
    }while(jwb == 'Y' || jwb == 'y');

    printf("\n\nNota Belanja Anda : \n");
    printf("\n          GEROBAK FRIED CHICKEN\n");
    printf("\n=====");
    printf("\nNo\tJenis Potong\tHarga Satuan\tQty\tJumlah Harga");
    printf("\n=====");

    for(j=0;j<i;j++)
    {
        switch(keranjang[j].kode)
        {
            case 'D': case 'd' :
                strcpy(jenis,"DADA");
                harga = 5000;
                break;
            case 'P': case 'p' :
                strcpy(jenis,"PAHA");
                harga = 4000;
                break;
            case 'S': case 's' :
                strcpy(jenis,"SAYAP");
                harga = 3000;
                break;
        }

        printf("\n%d\t%s\t\tRp.   %.2d\t\t%d\t\tRp.   %.2d",j+1,    jenis,    harga,
keranjang[j].jumlah, harga * keranjang[j].jumlah);
        total = total +  harga * keranjang[j].jumlah;
    }
    printf("\n=====");
    printf("\n          Jumlah bayar\tRp. %.2d",total);
    printf("\n          Pajak 10%\tRp. %.2d",total / 10);
    printf("\n          Total bayar\tRp. %.2d",total - total / 10);
    printf("\n          TERIMA KASIH ATAS KUNJUNGANNYA\n");
}

```

## b. Capture Program

```

D:\SEMESTER 1\Konsep Pemrograman\2110131036\praktikum29\Debug\soal2....
Toko GFC-Gerobak Fried Chicken
DAFTAR HARGA
Kode   Jenis   Harga Per Potong
D      Dada    Rp. 5000.00
P      Paha    Rp. 4000.00
S      Sayap   Rp. 3000.00

Harga belum termasuk pajak : 10 %

Masukkan Pesanan Anda
Pesanan ke-1
Jenis [D/P/S]   : D
Jumlah          : 2

Mau pesan lagi [y/n] ?Y

Pesanan ke-2
Jenis [D/P/S]   : P
Jumlah          : 3

Mau pesan lagi [y/n] ?Y

Pesanan ke-3
Jenis [D/P/S]   : S
Jumlah          : 1

Mau pesan lagi [y/n] ?N

```

D:\SEMESTER 1\Konsep Pemrograman\2110131036\praktikum29\Debug\soal2....

Nota Belanja Anda :

GEROBAK FRIED CHICKEN

No	Jenis Potong	Harga Satuan	Qty	Jumlah Harga
1	DADA	Rp. 5000	2	Rp. 10000
2	PAHA	Rp. 4000	3	Rp. 12000
3	SAYAP	Rp. 3000	1	Rp. 3000
Jumlah bayar				Rp. 25000
Pajak 10%				Rp. 2500
Total bayar				Rp. 22500

TERIMA KASIH ATAS KUNYUNGANNYA

- c. Analisis Program
- Untuk membuat program kasir seperti diatas, kita dapat menggunakan struct sebagai penyimpan pesanan barang. Struct pesanan(keranjang) tersebut terdiri dari dua variable masing-masing untuk menyimpan kode barang dan jumlah barang yang akan dibeli. Struct pesanan tersebut berupa array sehingga dapat menyimpan lebih dari 1 data pesanan.
- Selanjutnya, untuk mencetak nota belanja dilakukan looping for sebanyak jumlah struct pesanan yang disimpan sebelumnya. Setiap pesanan akan dicek kode dan jumlah pembeliannya sehingga didapatkan jumlah harga barang tersebut. Terakhir dicari total dari semua barang yang dibeli. Jangan lupa menambahkan pajak sebesar 10% dari total pembelian.

C. Kesimpulan

1. Struktur adalah pengelompokan variabel-variabel yang bernaung dalam satu nama yang sama.
2. Berbeda dengan array yang berisi kumpulan variabel -variabel yang bertipe sama dalam satu nama, maka suatu struktur dapat terdiri atas variabel -variabel yang berbeda tipenya dalam satu nama struktur.
3. Suatu struktur didefinisikan dengan menggunakan kata kunci struct. Contohnya :

```
struct date {  
    int month;  
    int day;  
    int year;  
};
```
4. Elemen dari suatu variabel struktur dapat diakses dengan menyebutkan nama variabel struktur diikuti dengan operator titik (‘.’) dan nama dari elemen strukturnya.
5. Elemen-elemen dari suatu array juga dapat berbentuk sebuah struktur. Misalnya array yang dipakai untuk menyimpan sejumlah data siswa (struct student).



# PRAKTIKUM 30

## Struct 2

---

### A. Dasar Teori

#### Struktur dan Fungsi

Melewatkan sebuah struktur untuk menjadi parameter sebuah fungsi dapat dilakukan sama dengan pengiriman parameter berupa variabel biasa. Fungsi yang mendapat kiriman parameter tersebut juga bisa mengirimkan hasil baliknya yang juga berupa sebuah struktur (pass by reference).

#### Melewatkan Elemen Struktur ke dalam Fungsi

Melewatkan parameter berupa elemen struktur dapat dilakukan sebagaimana pengiriman parameter berupa variabel biasa, dapat dilakukan baik secara nilai (pass by value) maupun secara acuan (pass by reference).

```
main(){
    struct date {
        int month;
        int day;
        int year;
    } today;
    ...
    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",
        nama_bulan[mm],dd,yy);
}
```

Tampak bahwa elemen dari struktur dilewatkan ke fungsi memakai bentuk pengaksesan elemen struktur, berupa :

```
cetak_tanggal(today.month, today.day, today.year);
```

Apabila nilai suatu elemen struktur diharapkan akan diubah oleh fungsi, maka yang dilewatkan haruslah berupa alamat dari elemen struktur (pass by reference). Untuk keperluan ini, operator alamat ditempatkan di depan nama variabel struktur (bukan di depan nama elemen struktur).

```
main() {
    struct koordinat {
        int x;
        int y;
    } posisi;
    ...
    tukar_xy(&posisi.x, &posisi.y);
}
```

```

...
}
void tukar_xy(int *a, int *b) {
    int z;
    z = *a;
    *a = *b;
    *b = z;
}

```

### Melewatkan Struktur ke dalam Fungsi

Pada program di atas misalnya, semua elemen dari struktur dikirimkan ke fungsi cetak\_tanggal(), dengan maksud nilai elemen dari struktur akan ditampilkan di layar. Untuk keadaan seperti ini, lebih baik kalau parameter fungsi diubah menjadi bentuk struktur, sehingga parameter fungsi tidak lagi sebanyak tiga buah, melainkan hanya satu. Selengkapnya, perhatikan program di bawah ini.

```

struct date {
    int month;
    int day;
    int year;
};
void cetak_tanggal(struct date);

main() {
    struct date today;
    ...
    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);
}

```

## B. Percobaan

1. Buatlah program sbb:

#### Input :

Data mahasiswa yang terdiri atas nama, nilai Tugas, nilai UTS, nilai UAS sebanyak n mahasiswa

#### Output :

Tampilkan dalam bentuk tabel

No, NRP, Nama Mhs, Nilai tugas, nilai UTS, nilai UAS, nilai Akhir, Grade

(Nilai Akhir = 20% tugas + 40% UTS + 40%UAS) dengan ketentuan sbb :

- Nilai akhir >= 80 : Grade A
- Nilai akhir >= 70 : Grade B
- Nilai akhir >= 60 : Grade C
- Nilai akhir >= 50 : Grade D
- Nilai akhir < 50 : Grade E

#### Petunjuk :

- Gunakan array of struct
- Perhatikan contoh tampilan
- Nilai tambah, gunakan array sebagai parameter fungsi (misal 2 fungsi : input() dan tampil())

a. Listing Program

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

int jumSiswa;
struct nilai {
    char nama[30];
    float nilaiTugas,nilaiUTS,nilaiUAS;
};
void input(struct nilai data[]);
void tampil(struct nilai data[]);

main()
{
    struct nilai mahasiswa[30];
    printf("\nMENGHITUNG NILAI AKHIR MATA KULIAH KONSEP PEMROGRAMAN");
    printf("\n=====\\n");

    input(mahasiswa);
    tampil(mahasiswa);
}

void input(struct nilai data[])
{
    int i;

    fflush(stdin);
    printf("\nBerapa jumlah mahasiswa ? ");
    scanf("%d",&jumSiswa);
    printf("\nMasukan data mahasiswa");
    printf("\n=====\\n");
    for(i=0;i<jumSiswa;i++)
    {
        printf("\nData ke-%d",i+1);
        fflush(stdin);
        printf("\nNama\\t\\t= ");
        scanf("%s",data[i].nama);
        fflush(stdin);
        printf("Nilai tugas\\t= ");
        scanf("%f",&data[i].nilaiTugas);
        fflush(stdin);
        printf("Nilai UTS\\t= ");
        scanf("%f",&data[i].nilaiUTS);
        printf("Nilai UAS\\t= ");
        scanf("%f",&data[i].nilaiUAS);
    }
}

void tampil(struct nilai data[])
{
    int i,j;
    float akhir;
    char grade;
```

## Capture Program



c. Analisis Program

Pada program penghitung nilai akhir mata kuliah konsep pemrograman ini, digunakan sebuah struct untuk menyimpan data nilai mahasiswa yang terdiri dari field nama, nilai tugas, nilai UTS dan nilai UAS. Struct itu sendiri nantinya akan berupa array sehingga dapat menyimpan data nilai dari beberapa mahasiswa pula.

Untuk memasukkan atau menginput data, dibuatlah fungsi input() dengan parameter array of struct nilai. Array of struct nilai tadi kemudian diisi secara looping satu persatu hingga user menjawab tidak (N).

Terakhir untuk menampilkan data yang telah diinputkan, dibuatlah fungsi tampil() dengan parameter array of struct nilai itu pula. Data ditampilkan per struct, disetiap prosesnya dicari nilai akhir sesuai ketentuan yang telah ada. Kemudian nilai akhir tersebut diseleksi menggunakan IF untuk mencari grade dari mahasiswa tersebut.

2. Toko “LARIS” menjual baju jadi dengan berbagai ukuran dengan daftar harga sbb:

Ukuran Baju	Nama Ukuran	Harga per potong
S	Small	Rp. 36.000,-
M	Medium	Rp. 43.000,-
L	Large	Rp. 62.000,-

**Ketentuan prosesnya sbb :**

- Jika membeli lebih dari 10 potong, akan mendapat diskon 20% dari jumlah harga, jika kurang atau sama dengan 10 potong, maka tidak ada diskon
- Pajak diberikan sebesar 10% dari (harga-diskon)

**Petunjuk :**

- Gunakan \t sebagai tabulasi untuk mengatur tampilan
- Gunakan <string.h> jika diperlukan
- Gunakan array of struct
- Gunakan array sebagai parameter fungsi (misal 2 fungsi : input() dan nota())

a. Listing Program

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

struct databarang {
    char kode;
    int jumlah;
};

int n = 0;

void input(struct databarang data[]);
void tampil(struct databarang data[]);

main()
{
    struct databarang keranjang[30];

    printf("\tToko Laris");
    printf("\n\tDaftar Harga");
    printf("\n=====");
    printf("\nUkuran\tNama\tHarga Per Potong");
    printf("\n=====");
    printf("\nS\tSMALL\tRp. 36.000,-");
```

```

printf("\nM\tMEDIUM\tRp. 43.000,-");
printf("\nL\tLARGE\tRp. 62.000,-");
printf("\n=====");
printf("\n\nDapatkan diskon 20 %% untuk pembelian lebih dari 10 potong!");
printf("\nJangan lupa, pajaknya ya... 10 %% x (harga - diskon)");

input(keranjang);
tampil(keranjang);
}
void input(struct databarang data[])
{
    char jwb;

    printf("\n\n=====");
    printf("\nMasukkan Pesanan Anda");

    do
    {
        printf("\n\nPesanan ke-%d",n+1);
        fflush(stdin);
        printf("\nUkuran [S/M/L]\t: ");
        scanf("%c",&data[n].kode);
        fflush(stdin);
        printf("Jumlah \t\t: ");
        scanf("%d",&data[n].jumlah);

        n++;
        fflush(stdin);
        printf("Beli lagi [y/n] ? ");
        jwb = getchar();
    }while(jwb == 'Y' || jwb == 'y');
}

void tampil(struct databarang data[])
{
    int j,harga,total=0,diskon;
    char jenis[7];

    printf("\n\n=====");
    printf("\n\nNota Belanja Anda : \n");
    printf("\n\t\t\tToko Laris\n");
    printf("\n=====");
    printf("\nNo\tNama Ukuran\tHarga Satuan\tQty\tDiskon\t\tJumlah Harga");
    printf("\n=====");

    for(j=0;j<n;j++)
    {
        switch(data[j].kode)
        {
            case 'S': case 's' :
                strcpy(jenis,"SMALL");
                harga = 36000;
                break;
            case 'M': case 'm' :
                strcpy(jenis,"MEDIUM");
                harga = 42000;
                break;
            case 'L': case 'l' :
                strcpy(jenis,"LARGE");
                harga = 64000;
                break;
        }

        if(data[j].jumlah > 10)
            diskon = harga * data[j].jumlah * 0.1;
        else
            diskon = 0;
    }
}

```

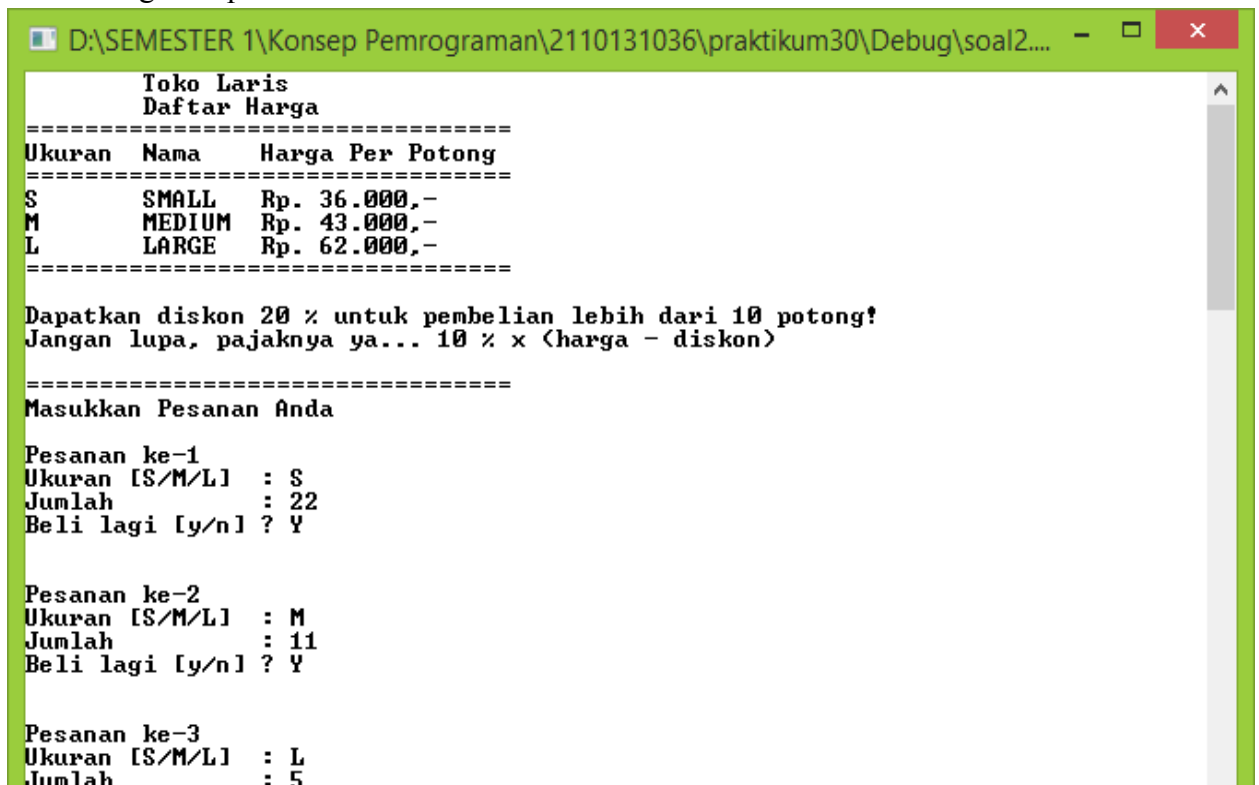
```

        printf("\n%d\t%s\tRp.      %d\t%d\tRp.      %5d\tRp.      %8d",j+1,      jenis,      harga,
data[j].jumlah,diskon, harga * data[j].jumlah - diskon);
        total = total + harga * data[j].jumlah - diskon;
    }
    printf("\n=====");
    printf("\n\t\t\t\t\tJumlah bayar\tRp. %8d",total);
    printf("\n\t\t\t\t\tPajak 10%\tRp. %8d",total / 10);
    printf("\n\t\t\t\t\tTotal bayar\tRp. %8d",total + total / 10);
    printf("\n\n\t\t\tTERIMA KASIH ATAS KUNJUNGANNYA\n");
}
}

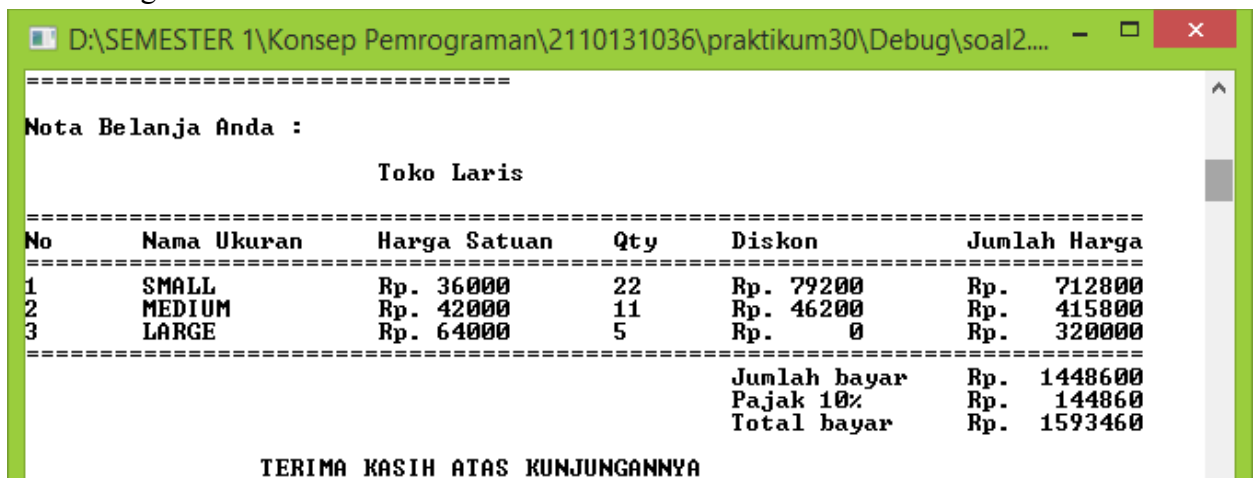
```

### b. Capture Program

Bagian input data :



Bagian cetak nota :



### c. Analisis Program

Untuk program no 2 ini, struct digunakan sebagai penyimpan pesanan yang telah dimasukkan oleh user. Struct pesanan atau keranjang belanja(struct databarang) tersebut terdiri dari dua field yaitu kode dan jumlah dari barang yang akan dibeli tersebut. Hampir sama dengan program pada soal pertama, pada program kasir Toko Laris ini menggunakan dua

buah fungsi buatan kita sendiri yaitu fungsi input untuk memasukkan data dan fungsi tampil untuk menampilkan data.

Perbedaannya di dalam program ini adalah pada proses penampilan data atau cetak notanya. Pertama, pada proses ini dicek berapa jumlah barang yang dibeli yang sebelumnya sudah tersimpan pada array of struct databarang apakah lebih dari 10 atau tidak. Jika iya berarti mendapatkan diskon. Selanjutnya dilakukan berbagai perhitungan matematis sehingga ketemu total yang harus dibayar oleh pembeli.

### C. Kesimpulan

1. Melewatkan sebuah struktur untuk menjadi parameter sebuah fungsi dapat dilakukan sama dengan pengiriman parameter berupa variabel biasa.
2. Fungsi yang mendapat kiriman parameter tersebut juga bisa mengirimkan hasil baliknya yang juga berupa sebuah struktur (pass by reference).
3. Melewatkan parameter berupa elemen struktur dapat dilakukan sebagaimana pengiriman parameter berupa variabel biasa, dapat dilakukan baik secara nilai (*pass by value*) maupun secara acuan (*pass by reference*).
4. Sebuah struct juga dapat dijadikan seluruhnya sebagai parameter dalam fungsi.



# PRAKTIKUM 31

## Struct 3

---

### A. 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;
}
```

## B. 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

### cetak1.c

#### a. Listing Program

```
#include <stdio.h>
void cetak_tanggal(int, int, int);
main()
{
    struct date {
        int month, day, 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);

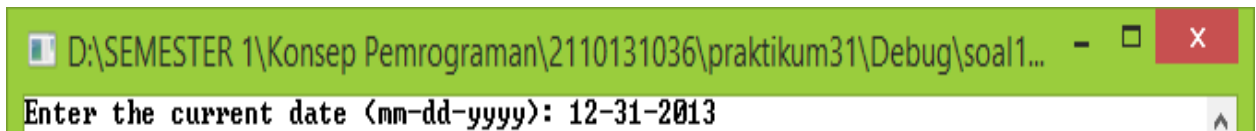
    getch();
}

void cetak_tanggal(int mm, int dd, int yy){
    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);
}
```



## b. Capture Program



## c. Analisis Program

Pada percobaan ini struct digunakan untuk menyimpan tanggal sekarang yang terdiri dari field bulan, hari dan tahun(struct date). Selanjutnya struct date dilewatkan ke dalam fungsi cetak\_tanggal() dengan cara menjadikannya parameter satu persatu untuk masing-masing elemen dalam struct date.

### cetak2.c

#### a. Listing Program

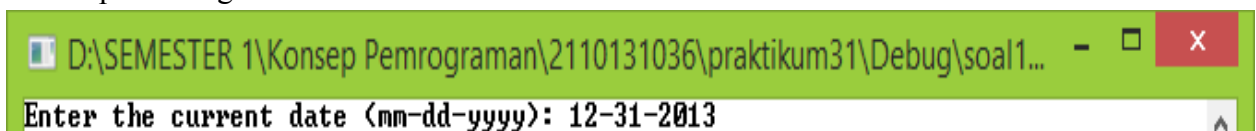
```
#include <stdio.h>
struct date {
    int month, day, year;
};

void cetak_tanggal(struct date);

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)
{
    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);
}
```

## b. Capture Program



## c. Analisis Program

Pada percobaan ini masih sama dengan percobaan sebelumnya(cetak1.c). Struct masih digunakan untuk menyimpan tanggal sekarang yang terdiri dari field bulan, hari dan tahun(struct date). Perbedaannya disini adalah pada pemanggilan fungsi cetak\_tanggal(), struct date langsung dijadikan parameter tanpa memecah elemennya satu persatu.

### posisi1.c

#### a. Listing Program

```
#include <stdio.h>

void tukar_xy(int *, int *);
```

```

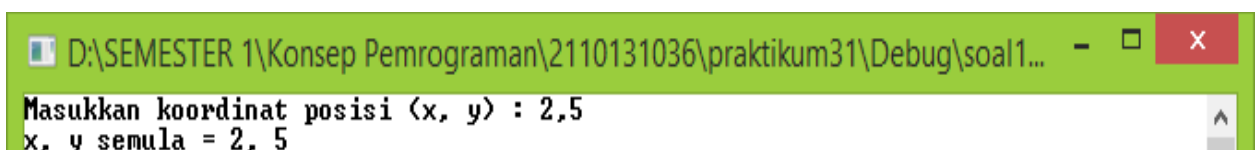
main()
{
    struct koordinat {
        int x, y;
    } posisi;

    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);
    getch();
}

void tukar_xy(int *a, int *b)
{
    int z;
    z = *a;
    *a = *b;
    *b = z;
}

```

#### b. Capture Program



#### c. Analisis Program

Pada program ini, struct digunakan untuk menyimpan koordinat yang terdiri dari nilai x dan y (struct koordinat). Selanjutnya, program ini akan menukar isi dari masing-masing x dan y tadi. Karena itu digunakanlah pointer agar perubahan variabel pada fungsi lain ikut merubah nilai aslinya di fungsi main().

Yang dikirim ke fungsi tukar\_xy() adalah alamat dari masing-masing element struct data. Yang kemudian pada fungsi tukar\_xy() nilai yang ditunjuk keduanya ditukar menggunakan bantuan variabel ketiga yaitu z.

### posisi2.c

#### a. Listing Program

```

#include <stdio.h>
struct koordinat {
    int x, y;
};

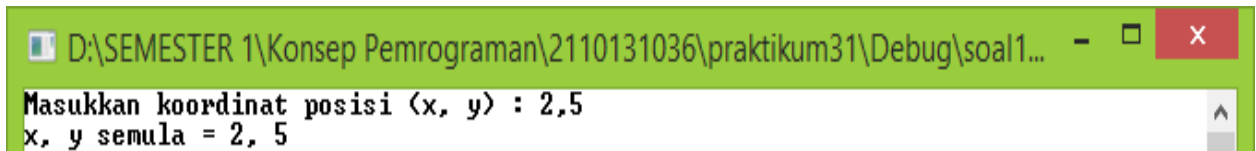
void tukar_xy(struct koordinat *);
main() {
    struct koordinat posisi;
    printf("Masukkan koordinat posisi (x, y) : ");
    scanf("%d, %d", &posisi.x, &posisi.y);
    printf("x, y semula = %d, %d\n", posisi.x, posisi.y);
    tukar_xy(&posisi);
    printf("x, y sekarang = %d, %d\n", posisi.x, posisi.y);
}

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

```

}

b. Capture Program



c. Analisis Program

Pada program ini, struct masih digunakan untuk menyimpan koordinat yang terdiri dari nilai x dan y(struct koordinat). Selanjutnya, program ini akan menukar isi dari masing-masing x dan y tadi. Karena itu digunakanlah pointer agar perubahan variabel pada fungsi lain ikut merubah nilai aslinya di fungsi main().

Perbedaannya apabila pada soal posisi1.c yang dikirim sebagai parameter ke fungsi tukar\_xy() adalah alamat dari masing-masing element struct date, pada percobaan ini yang dikirim sebagai parameter ke fungsi tukar\_xy() adalah alamat dari struct koordinat tersebut. Yang selanjutnya pada fungsi tukar\_xy() nilai yang ditunjuk oleh struct koordinat pada kedua fieldnya ditukar menggunakan bantuan variabel ketiga yaitu z.

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)
- Jika diperlukan, gunakan fungsi strcpy() dari <string.h>

a. Listing Program

```
#include <stdio.h>

int n;

struct date {
    int hari, bulan, tahun;
};
struct datakaryawan {
    int id,gaji;
    char nama[32], jenis;
    struct date tglLahir;
};

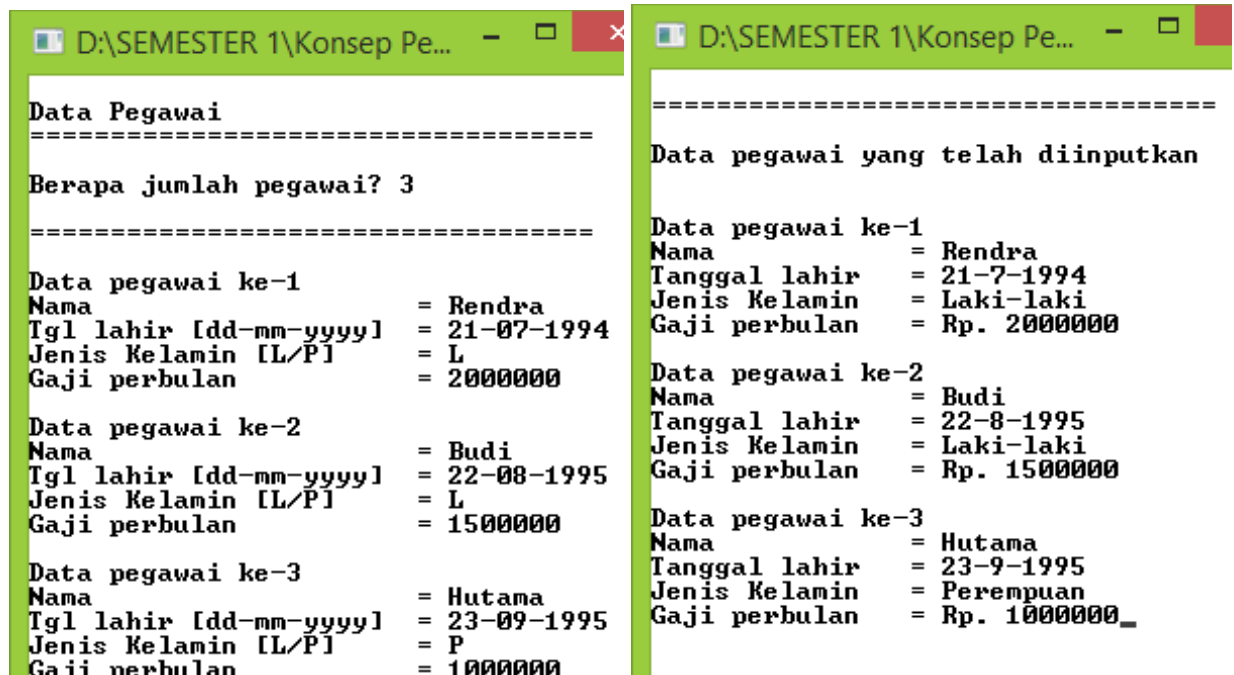
typedef struct datakaryawan pegawai;

void input(pegawai *data);
void tampil(pegawai *data);

main()
{
    pegawai dataPegawai[30];
```



b. Capture Program



c. Analisis Program

Pada program input data pegawai ini sebenarnya hamper mirip dengan program input data yang lain. Namun perbedaanya disini struct karyawan terlebih dahulu diubah namanya sehingga menjadi lebih singkat yaitu pegawai saja menggunakan perintah typedef. Hal ini tentu saja memudahkan kita dalam penyebutan struct selanjutnya di dalam program. Selanjutnya pegawai tadi dibuat sebagai array agar dapat menyimpan banyak data pegawai.

Perbedaan lainnnya yang cukup mencolok disini adalah array of pegawai dikirimkan ke fungsi secara *pass by reference* sehingga pada fungsi array of pegawai yang merupakan nama lain dari array of struct karyawan tidak lagi diakses dengan indeksnya, namun langsung menggunakan pointer contohnya `(*data).nama`

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

- Ubah nama tipe data dengan menggunakan typedef
- Panggillah 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

a. Listing Program

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

int jumSiswa;

struct nilai {
    char nama[30];
    float nilaiTugas, nilaiUTS, nilaiUAS;
};

typedef struct nilai daftarNilai;
```

```

void input(daftarNilai *);
void tampil(daftarNilai *);

main()
{
    struct nilai mahasiswa[30];
    printf("\nMENGHITUNG NILAI AKHIR MATA KULIAH KONSEP PEMROGRAMAN");
    printf("\n=====");

    input(mahasiswa);
    tampil(mahasiswa);
}

void input(daftarNilai *data)
{
    int i;

    fflush(stdin);
    printf("\nBerapa jumlah mahasiswa ? ");
    scanf("%d",&jumSiswa);
    printf("\nMasukan data mahasiswa");
    printf("\n=====");
    for(i=0;i<jumSiswa;i++)
    {
        printf("\nData ke-%d",i+1);
        fflush(stdin);
        printf("\nNama\t\t= ");
        scanf("%s",(*data).nama);
        fflush(stdin);
        printf("Nilai tugas\t= ");
        scanf("%f",&(*data).nilaiTugas);
        fflush(stdin);
        printf("Nilai UTS\t= ");
        scanf("%f",&(*data).nilaiUTS);
        printf("Nilai UAS\t= ");
        scanf("%f",&(*data).nilaiUAS);
        data++;
    }
}

void tampil(daftarNilai *data)
{
    int i,j;
    float akhir;
    char grade;

    printf("\n\t\t\t\tDAFTAR NILAI");
    printf("\n\t\t\t\tMATAKULIAH KONSEP PEMROGRAMAN");
    printf("\n=====");
    printf("\nNo\tNama Mahasiswa\tTugas\tUTS\tUAS\tAkhir\tGrade");
    printf("\n=====");
    for(j=0;j<jumSiswa;j++)
    {
        akhir = (0.2 * (*data).nilaiTugas) + (0.4 * (*data).nilaiUAS) + + (0.4 *
(*data).nilaiUTS);

        if(akhir>=80)
            grade = 'A';
        else if(akhir>=70)
            grade = 'B';
        else if(akhir>=60)
            grade = 'C';
        else if(akhir>=50)
            grade = 'D';
        else
            grade = 'E';
    }
}

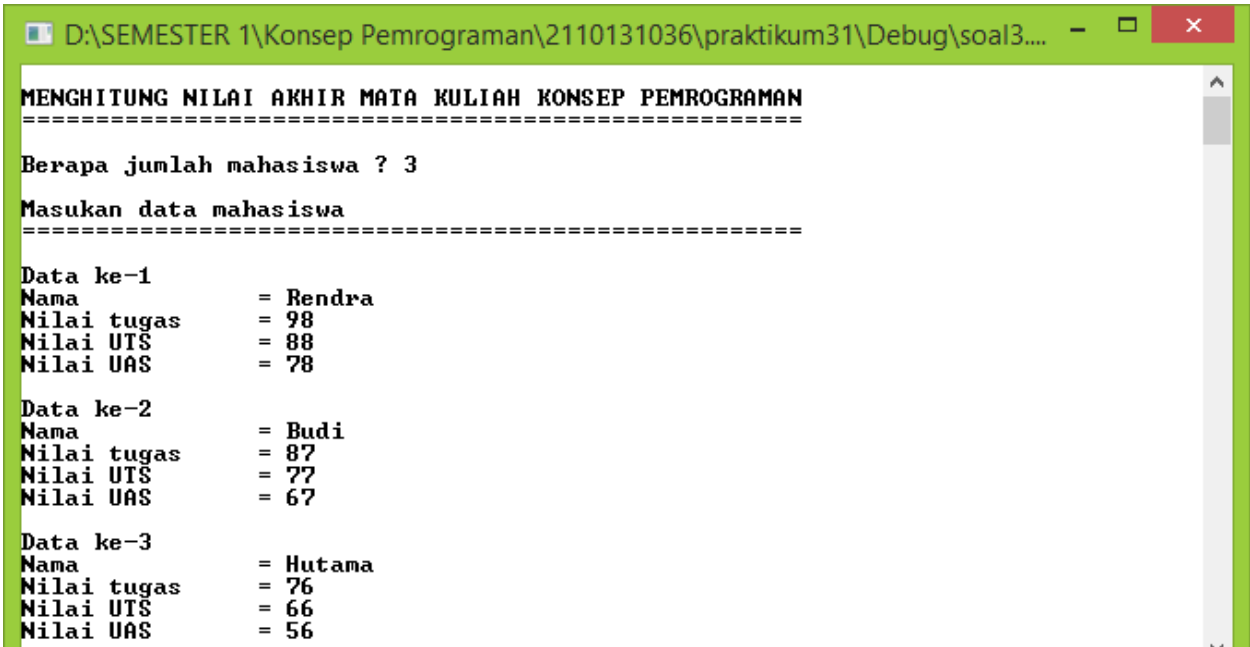
```



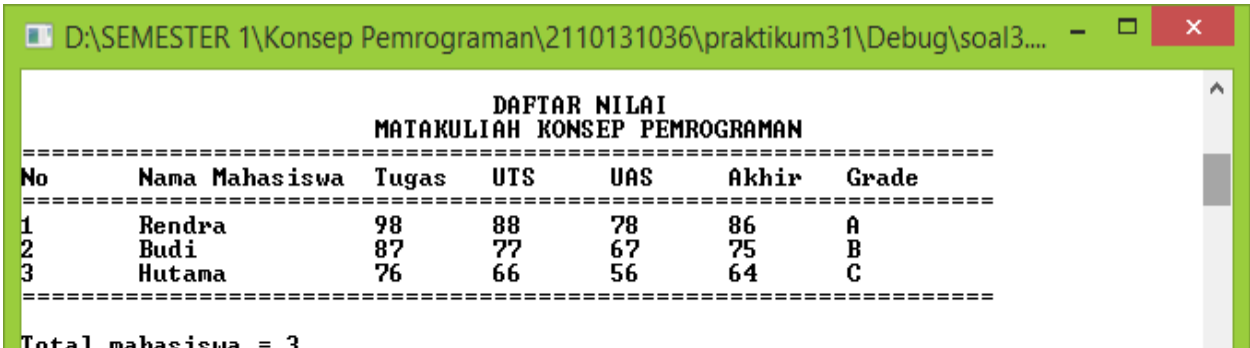
```
        printf("\n%d\t%s\t\t%.0f\t%.0f\t%.0f\t%.0f\t%c",j+1,(*data).nama,
(*data).nilaiTugas,(*data).nilaiUTS,(*data).nilaiUAS,akhir,grade);
        data++;
    }
    printf("\n=====");
    printf("\n\nTotal mahasiswa = %d",jumSiswa);
}
```

b. Capture Program

Bagian input data :



Bagian Output data :



c. Analisis Program

Merupakan modifikasi dari soal struct 2 praktikum 30 soal no 1 program ini sama-sama berfungsi untuk menghitung nilai akhir dari mata kuliah konsep pemrograman. Perbedaannya dalam program ini sudah menggunakan typedef sebagai peyingkat nama struct. Selain itu dalam pemanggilan fungsi, yang dikirim ke fungsi sebagai parameter adalah alamat dari struct yang telah ditypedef namanya tersebut.

Karena yang dikirim adalah alamat, maka yang menerima dalam fungsi adalah pointer. Dengan bantuan pointer tersebut, pemanggilan nilai struct tidak perlu lagi menggunakan indeks dari arraynya, cukup dengan pemanggilan nilai yang ditunjuk pointer tersebut.

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

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

a. Listing Program

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

int jumSiswa;

struct nilai {
    char nama[30];
    float nilaiTugas,nilaiUTS,nilaiUAS;
};

struct maks {
    char nama[30];
    float nilai;
};

typedef struct nilai daftarNilai;

void input(daftarNilai *);
void tampil(daftarNilai *);

main()
{
    struct nilai mahasiswa[30];
    printf("\nMENGHITUNG NILAI AKHIR MATA KULIAH KONSEP PEMROGRAMAN");
    printf("\n=====");

    input(mahasiswa);
    tampil(mahasiswa);
}

void input(daftarNilai *data)
{
    int i;

    fflush(stdin);
    printf("\nBerapa jumlah mahasiswa ? ");
    scanf("%d",&jumSiswa);
    printf("\nMasukan data mahasiswa");
    printf("\n=====");
    for(i=0;i<jumSiswa;i++)
    {
        printf("\nData ke-%d",i+1);
        fflush(stdin);
        printf("\nNama\t\t= ");
        scanf("%s",(*data).nama);
        fflush(stdin);
        printf("Nilai tugas\t= ");
        scanf("%f",&(*data).nilaiTugas);
        fflush(stdin);
        printf("Nilai UTS\t= ");
        scanf("%f",&(*data).nilaiUTS);
        printf("Nilai UAS\t= ");
        scanf("%f",&(*data).nilaiUAS);
        data++;
    }
}

void tampil(daftarNilai *data)
{
    int i,j;
    float akhir;
    char grade;
```

```

struct maks daftarMaks;

printf("\n\t\t\t\t\tDAFTAR NILAI");
printf("\n\t\t\t\t\tMATAKULIAH KONSEP PEMROGRAMAN");
printf("\n=====");
printf("\nNo\tNama Mahasiswa\tTugas\tUTS\tUAS\tAkhir\tGrade");
printf("\n=====");
for(j=0;j<jumSiswa;j++)
{
    akhir = (0.2 * (*data).nilaiTugas) + (0.4 * (*data).nilaiUAS) + + (0.4 *
(*data).nilaiUTS);

    if(j == 0 || daftarMaks.nilai < akhir)
    {
        strcpy(daftarMaks.nama,(*data).nama);
        daftarMaks.nilai = akhir;
    }

    if(akhir>=80)
        grade = 'A';
    else if(akhir>=70)
        grade = 'B';
    else if(akhir>=60)
        grade = 'C';
    else if(akhir>=50)
        grade = 'D';
    else
        grade = 'E';

    printf("\n%d\t%s\t\t%.0f\t%.0f\t%.0f\t%.0f\t%c",j+1,(*data).nama,
(*data).nilaiTugas,(*data).nilaiUTS,(*data).nilaiUAS,akhir,grade);
    data++;
}
printf("\n=====");
printf("\n\nTotal mahasiswa\t= %d",jumSiswa);
printf("\n\nNilai Tertinggi");
printf("\nNama mahasiswa\t= %s",daftarMaks.nama);
printf("\nNilai\t= %.2f",daftarMaks.nilai);
}

```

## b. Capture Program

Bagian input data :

```

D:\SEMESTER 1\Konsep Pemrograman\2110131036\praktikum31\Debug\soal4....
MENGHITUNG NILAI AKHIR MATA KULIAH KONSEP PEMROGRAMAN
=====
Berapa jumlah mahasiswa ? 3
Masukan data mahasiswa
=====
Data ke-1
Nama          = Rendra
Nilai tugas   = 98
Nilai UTS     = 88
Nilai UAS     = 78
Data ke-2
Nama          = Budi
Nilai tugas   = 87
Nilai UTS     = 77
Nilai UAS     = 67
Data ke-3
Nama          = Hutama
Nilai tugas   = 76
Nilai UTS     = 66
Nilai UAS     = 56

```

Bagian output data :

DAFTAR NILAI MATAKULIAH KONSEP PEMROGRAMAN						
No	Nama Mahasiswa	Tugas	UTS	UAS	Akhir	Grade
1	Rendra	98	88	78	86	A
2	Budi	87	77	67	75	B
3	Hutama	76	66	56	64	C
Total mahasiswa = 3						
Nilai Tertinggi						
Nama mahasiswa = Rendra						

c. Analisis Program

Masih merupakan modifikasi dari soal struct 2 praktikum 30 soal no 1. Program ini merupakan program untuk mencari nilai akhir dari mata kuliah konsep pemrograman. Perbedaannya dalam program ini sudah menggunakan *pass by reference* sesuai soal no 3 pada pengiriman array of struct nilainya.

Sedikit tambahan pada program ini adalah adanya fitur untuk mengetahui nilai tertinggi dari mahasiswa. Untuk itu diperlukanlah sebuah struct baru yang nantinya digunakan untuk menyimpan data nilai tertinggi. Struct baru tersebut adalah struct maks. Defaultnya struct maks akan bernilai sama dengan nilai element struct nilai pertama. Namun nantinya akan berubah jika menemui nilai yang lebih besar darinya hingga pengulangan untuk penampilan struct berakhir. Dan terakhir tinggal kita tampilkan isi dari masing-masing element dari struct maks tadi.

C. Kesimpulan

1. Suatu struct dapat diakses menggunakan variabel pointer.
2. Dalam penulisan pointer pada struct 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 \*).