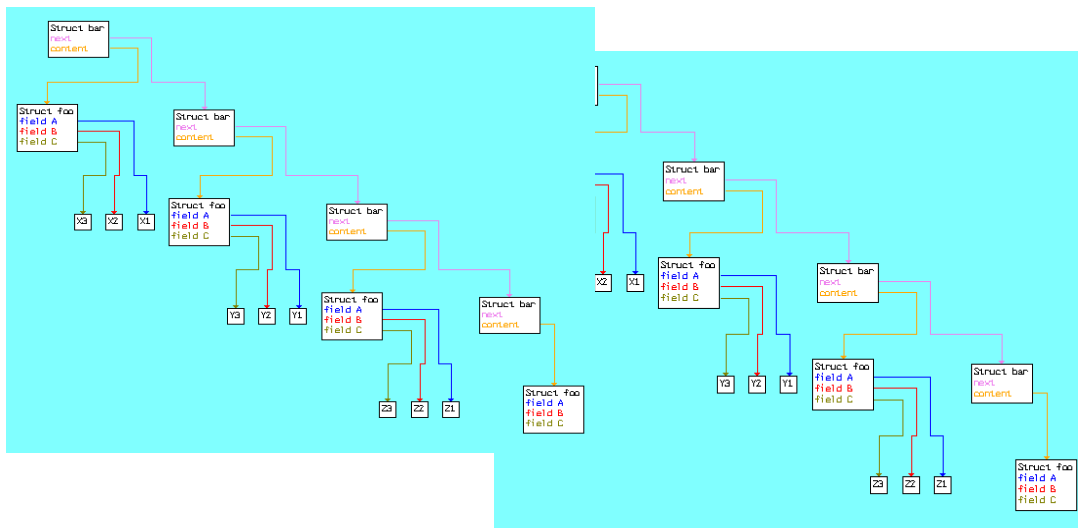


PENGENALAN STRUCT



Apa Itu Struct?

A. Pengenalan Struct

Latar Belakang

Struct sebenarnya mirip seperti array. Seperti yang telah kita ketahui, array sekumpulan variabel bertipe sama yang dikelompokkan dan kemudian kelompok tersebut memiliki nama yang sama. Maka, struct seperti array namun membolehkan beberapa variabel yang berbeda tipe dikelompokkan menjadi satu dalam nama yang sama. Dalam java, struct berperan sama seperti *object*.

Contoh

Kamu dapat membuat structure bernama 'person' yang disusun dari string 'name' dan integer 'age' :

```
struct person
{
    char *name;
    int age;
};
```

Contoh di atas adalah **deklarasi** dari tipe. Kamu masih harus membuat variabel dari tipe tersebut agar dapat menggunakannya.

Untuk itu, lakukan langkah berikut ini:

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

```
struct person
```

```
{
```

```
    char *name;
```

```
    int age;
```

```
};
```

```
int main()
```

```
{
```

```
    struct person p;
```

```
    return 0;
```

```
}
```

Variabel p

Untuk mengakses string atau integer dari structure, kamu harus menggunakan **dot (.)** di antara nama structure dan nama variabel

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

```
struct person
```

```
{
```

```
    char *name;
```

```
    int age;
```

```
};
```

```
int main()
```

```
{
```

```
    struct person p;
```

```
    p.name = "John Smith";
```

```
    p.age = 25;
```

```
    printf("%s",p.name);
```

```
    printf("%d",p.age);
```

```
    return 0;
```

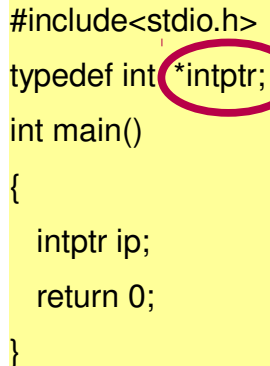
```
}
```

Pengaksesan struct p untuk diberi nilai

B. Pendefinisian Tipe

Kamu dapat memberi namamu sendiri ke sebuah variabel menggunakan sebuah pendefinisian tipe. Contoh berikut ini untuk membuat pendefinisian tipe yang disebut `intptr` untuk sebuah **pointer ke sebuah integer**:

```
#include<stdio.h>
typedef int *intptr;
int main()
{
    intptr ip;
    return 0;
}
```



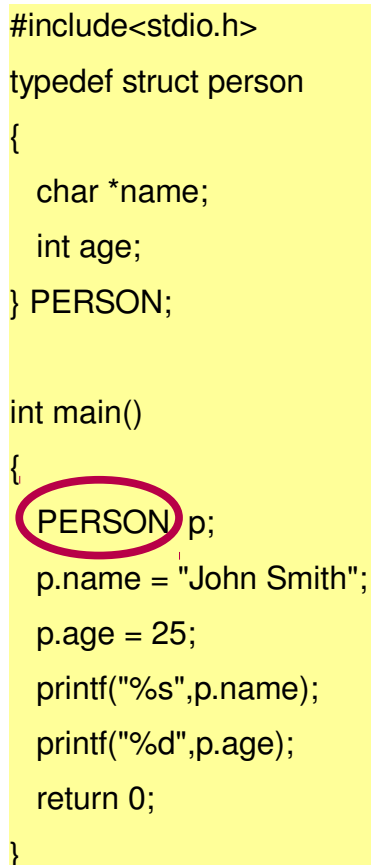
C. Pendefinisian Tipe untuk Structure

Bila kamu tidak ingin memakai kata "struct" saat mendeklarasikan sebuah variabel structure maka kamu dapat membuat definisi structure itu sendiri.

Perhatikan bahwa structure biasa ditulis dalam HURUF BESAR.

```
#include<stdio.h>
typedef struct person
{
    char *name;
    int age;
} PERSON;

int main()
{
    PERSON p;
    p.name = "John Smith";
    p.age = 25;
    printf("%s",p.name);
    printf("%d",p.age);
    return 0;
}
```

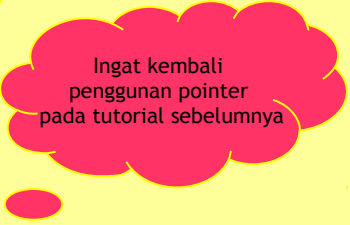


D. Pointer menjadi Structure

Ketika kamu menggunakan pointer menjadi sebuah structure kamu harus menggunakan → daripada dot.

```
#include<stdio.h>
typedef struct person
{
    char *name;
    int age;
} PERSON;

int main()
{
    PERSON p;
    PERSON *pptr;
    PERSON pptr = &p;
    pptr->name = "John Smith";
    pptr->age = 25;
    printf("%s",pptr->name);
    printf("%d",pptr->age);
    return 0;
}
```



Ingat kembali
penggunaan pointer
pada tutorial sebelumnya

E. Unions

Apa itu unions?

Union hampir seperti structures, namun mereka memakan memori yang lebih sedikit. Bila kamu membuat structures dengan tipe data double(8 bytes) dan integer (4 bytes) maka dengan unions total memori tetap hanya membutuhkan 8 bytes (bukan 12). Hal ini disebabkan karena double berada di atas prioritas integer. Bisa dijelaskan dengan kata lain, integer adalah bagian dari double itu sendiri. Mereka saling berhubungan. Bila kamu mengubah nilai dari double maka nilai dari integer juga berubah. Kemudian, bila kamu mengubah nilai dari integer, nilai double akan berubah pula.

Contoh penggunaan

```
#include<stdio.h>
typedef union num
{
    double d;
    int i;
} NUM;

int main()
{
    NUM n;
    n.d = 3.14;
    n.i = 5;
    return 0;
}
```

Union Num dengan variabel n
ada yg bernilai double dan yang bernilai integer

Recommended Link :

Learn C programming – C for dummies – tutorial Lesson 9 – structures. (n.d.). July 12, 2012.
http://www.programmerprogramming.com/programming_c_learn09.html

**Revisi Terakhir :**

<http://bebas.vlsm.org/v24/sosp12/root/tutorial-lab/struct-r04.pdf>

Disusun Oleh : Annisa Prida Rachmadianty dari tim SOSP12

Kontributor:

Pembina :

1. Rahmat M Samik-Ibrahim
2. Ramot Stephanus
3. Rizki Mandala Putra

Tim SOSP12 (<http://sosp.vlsm.org/>) :

1. Aditya Try Anggoro
2. Annisa Prida Rachmadianty
3. Ardanella Nelwan
4. Farah Nuraini
5. Nisrina Luthfiyati
6. Siska Eka Putriani
7. Zulfikar Ian Ifetayo



Copyright 2012 by SOSP12. all rights reserved. This material may be freely copied and distributed subject to inclusion of this copyright notice and our World Wide Web URL <http://sosp.vlsm.org>