

Konsep Pemrograman

14. Struktur - 2

Umi Sa'adah

Entin Martiana Kusumaningtyas

Tri Hadiah Muliawati

2020



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Overview

- Struktur dan Fungsi
 - Melewatkan Elemen Struktur ke dalam Fungsi
 - Melewatkan Struktur ke dalam Fungsi
1. Melewatkan Elemen Struktur ke dalam Fungsi
 - Pass by value
 - Pass by reference
 2. Melewatkan Struktur ke dalam Fungsi
 - Pass by value
 - Pass by reference (*pointer to struct*)

Struktur dan Fungsi

- Melewatkan sebuah struktur untuk menjadi parameter dalam 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*)

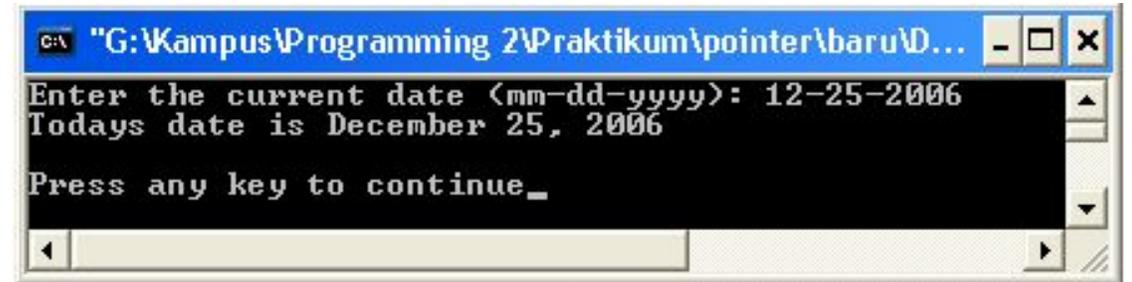
Melewatkan Elemen Struktur ke dalam Fungsi

- Melewatkan parameter berupa elemen struktur dapat dilakukan sebagaimana pengiriman parameter berupa variabel biasa, dapat dilakukan baik secara :
 - Pengiriman nilai (*pass by value*)
 - Pengiriman *acuan/address* (*pass by reference*).

Melewatkan Elemen Struktur ke dalam Fungsi

Pass by value

```
1 #include <stdio.h>
2 void cetakTanggal(int, int, int);
3 int main() {
4     struct date {
5         int month, day, year;
6     } today;
7
8     printf("Enter the current date (mm-dd-yyyy): ");
9     scanf("%d-%d-%d", &today.month, &today.day, &today.year);
10
11     cetakTanggal(today.month, today.day, today.year);
12     return 0;
13 }
14
15 void cetakTanggal(int mm, int dd, int yy){
16     char *nama_bulan[] = {
17         "Wrong month", "January", "February", "March",
18         "April", "May", "June", "July", "August",
19         "September", "October", "November", "December"
20     };
21     printf("Todays date is %s %d, %d\n\n", nama_bulan[mm],dd,yy);
22 }
```



```
C:\ "G:\Kampus\Programming 2\Praktikum\pointer\baru\D...
Enter the current date (mm-dd-yyyy): 12-25-2006
Todays date is December 25, 2006
Press any key to continue_
```

- Pengiriman parameter dari elemen / field struct ke dalam fungsi dilakukan pada baris ke-11.

Melewatkan Elemen Struktur ke dalam Fungsi

Pass by value

- Tampak bahwa nilai dari field `struct` dilewatkan ke fungsi memakai bentuk pengaksesan field `struct`, berupa :

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



Melewatkan Struktur ke dalam Fungsi

Pass by value

- Pada program sebelumnya, nilai dari semua field pada variabel `today` yang memiliki tipe data `struct date` dikirimkan sebagai input parameter fungsi `cetakTanggal()`.
- Untuk keadaan seperti ini (dimana semua field dikirim sebagai input parameter), akan lebih baik apabila input parameter dari fungsi `cetakTanggal()` diubah ke dalam bentuk `struct`, sehingga input parameter yang dibutuhkan hanya 1.

Melewatkan Struktur ke dalam Fungsi

Pass by value

```

1 #include <stdio.h>
2 void cetakTanggal(int, int, int);
3 int main() {
4     struct date {
5         int month, day, year;
6     } today;
7
8     printf("Enter the current date (mm-dd-yyyy): ");
9     scanf("%d-%d-%d", &today.month, &today.day, &today.year);
10
11     cetakTanggal(today.month, today.day, today.year);
12     return 0;
13 }
14
15 void cetakTanggal(int mm, int dd, int yy){
16     char *nama_bulan[] = {
17         "Wrong month", "January", "February", "March",
18         "April", "May", "June", "July", "August",
19         "September", "October", "November", "December"
20     };
21     printf("Todays date is %s %d, %d\n\n", nama_bulan[mm], dd, yy);
22 }

```

```

1 #include <stdio.h>
2 struct date {
3     int month, day, year;
4 };
5 void cetakTanggal(struct date);
6
7 int main() {
8     struct date today;
9     printf("Enter the current date (mm-dd-yyyy): ");
10    scanf("%d-%d-%d", &today.month, &today.day, &today.year);
11    cetakTanggal(today);
12    return 0;
13 }
14
15 void cetakTanggal(struct date now){
16     char *nama_bulan[] = {
17         "Wrong month", "January", "February", "March", "April", "May", "June",
18         "July", "August", "September", "October", "November", "December"
19     };
20     printf("Todays date is %s %d, %d\n\n", nama_bulan[now.month], now.day, now
        .year);
21 }

```

Melewatkan Elemen Struktur ke dalam Fungsi

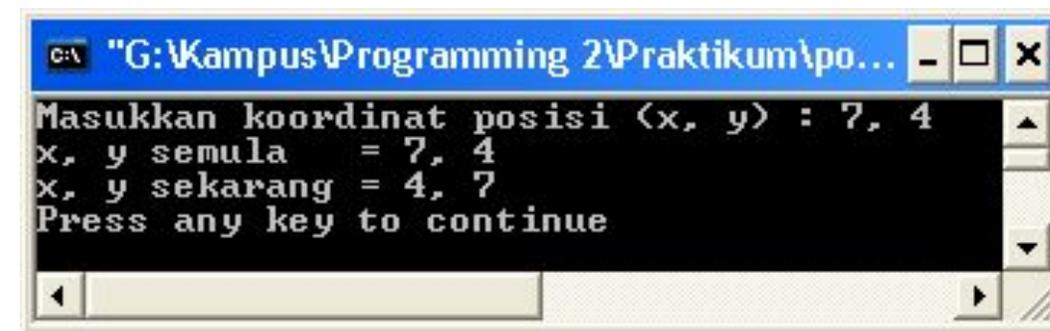
Pass by reference

- 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 `struct` (bukan di depan nama field `struct`).

Melewatkan Elemen Struktur ke dalam Fungsi

Pass by reference

```
1  #include <stdio.h>
2  void tukar_xy(int *, int *);
3
4  int main() {
5      struct koordinat {
6          int x, y;
7      } posisi;
8
9      printf("Masukkan koordinat posisi (x, y) : ");
10     scanf("%d, %d", &posisi.x, &posisi.y);
11     printf("x, y semula = %d, %d\n", posisi.x, posisi.y);
12     tukar_xy(&posisi.x, &posisi.y);
13     printf("x, y sekarang = %d, %d\n", posisi.x, posisi.y);
14 }
15
16 void tukar_xy(int *a, int *b) {
17     int z;
18     z = *a;
19     *a = *b;
20     *b = z;
21 }
```



```
G:\Kampus\Programming 2\Praktikum\po...
Masukkan koordinat posisi (x, y) : 7, 4
x, y semula = 7, 4
x, y sekarang = 4, 7
Press any key to continue
```

- Pengiriman alamat dari elemen / field struct ke dalam fungsi dilakukan pada baris ke-12.
- Karena parameter yang dikirimkan ke dalam fungsi berupa alamat dari struct, maka perubahan nilai field struct yang dilakukan di dalam fungsi `tukar_xy()` akan berdampak pada nilai field struct yang dikirimkan.

Melewatkan Struktur ke dalam Fungsi

Pass by reference

- Jika sebuah struktur mengandung banyak field dan diputuskan bahwa keseluruhan field-nya akan diubah oleh fungsi, maka cara yang efisien adalah dengan melewati alamat dari `struct` (*pass by reference*).
- Dengan demikian pada saat pendefinisian fungsi, parameter formalnya berupa pointer yang menunjuk ke `struct` (*pointer to struct*). Sedangkan parameter aktualnya berupa alamat dari `struct` yang akan dikirimkan.



Melewatkan Struktur ke dalam Fungsi

Pass by reference

```

1  #include <stdio.h>
2  void tukar_xy(int *, int *);
3
4  int main() {
5      struct koordinat {
6          int x, y;
7      } posisi;
8
9      printf("Masukkan koordinat posisi (x, y) : ");
10     scanf("%d, %d", &posisi.x, &posisi.y);
11     printf("x, y semula = %d, %d\n", posisi.x, posisi.y);
12     tukar_xy(&posisi.x, &posisi.y);
13     printf("x, y sekarang = %d, %d\n", posisi.x, posisi.y);
14 }
15
16 void tukar_xy(int *a, int *b) {
17     int z;
18     z = *a;
19     *a = *b;
20     *b = z;
21 }

```

```

1  #include <stdio.h>
2  struct koordinat {
3      int x, y;
4  };
5
6  void tukar_xy(struct koordinat *);
7
8  int main() {
9      struct koordinat posisi;
10     printf("Masukkan koordinat posisi (x, y) : ");
11     scanf("%d, %d", &posisi.x, &posisi.y);
12     printf("x, y semula = %d, %d\n", posisi.x, posisi.y);
13     tukar_xy(&posisi);
14     printf("x, y sekarang = %d, %d\n", posisi.x, posisi.y);
15 }
16 void tukar_xy(struct koordinat *pos_xy) {
17     int z;
18     z = (*pos_xy).x;
19     (*pos_xy).x = (*pos_xy).y;
20     (*pos_xy).y = z;
21 }

```

Melewatkan Struktur ke dalam Fungsi

Pass by reference

- Parameter dari fungsi `tukar_xy()` disederhanakan menjadi satu parameter saja, yakni sebagai berikut :

```
void tukar_xy(struct koordinat *pos_xy) {  
...  
}
```

- Pada definisi fungsi di atas, `struct koordinat *pos_xy` menyatakan bahwa `pos_xy` adalah pointer yang menunjuk ke obyek bertipe data `struct koordinat`.
- Adapun penulisan `(*pos_xy).x` menyatakan field bernama `x` yang ditunjuk oleh pointer `pos_xy`
- Perlu diperhatikan bahwa penulisan tanda kurung seperti pada `(*pos_xy).x` merupakan **suatu keharusan**, sebab `*pos_xy.x` mempunyai makna yang berbeda dengan `(*pos_xy).x`
- Ungkapan `*pos_xy.x` mempunyai makna "yang ditunjuk oleh `pos_xy.x`". Hal ini dikarenakan operator titik mempunyai prioritas yang lebih tinggi daripada operator `*`.

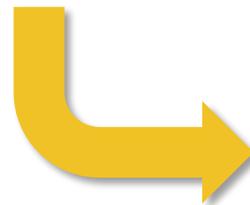


Melewatkan Struktur ke dalam Fungsi

Pass by reference

- Bentuk semacam `(*pos_xy).x` dapat ditulis dengan bentuk lain menjadi `pos_xy->x`
- Dalam C operator `->` disebut sebagai operator panah.
- Dengan menggunakan operator panah, maka fungsi `tukar_xy()` dalam program sebelumnya dapat ditulis ulang menjadi sbb:

```
16 void tukar_xy(struct koordinat *pos_xy) {  
17     int z;  
18     z = (*pos_xy).x;  
19     (*pos_xy).x = (*pos_xy).y;  
20     (*pos_xy).y = z;  
21 }
```



```
16 void tukar_xy(struct koordinat *pos_xy) {  
17     int z;  
18     z = pos_xy->x;  
19     pos_xy->x = pos_xy->y;  
20     pos_xy->y = z;  
21 }
```

bridge to the future

<http://www.eepis-its.edu>

