

PERBANDINGAN SUBPROGRAM PADA BAHASA C DAN JAVA

Rachmat Selamat

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI
Jl. Ir. H. Juanda 96 Bandung 40132

E-mail : rachmatselametskom@gmail.com

ABSTRAK

Pemrograman Java dan C adalah bahasa yang paling banyak digunakan saat ini. Bahasa C adalah induk dari bahasa Java. Walaupun sebagai induk bahasa, ke dua bahasa pemrograman ini memiliki banyak perbedaan penulisan, terutama dalam pembuatan subprogram. Fitur untuk subprogram di Java lebih kompleks dibandingkan dengan bahasa C karena mengusung pemrograman berbasis objek di mana penerapan subprogram lebih beragam.

Salah satu materi mengenai subprogram adalah parameter. Kasus parameter yang cukup ramai dibicarakan adalah tidak adanya parameter alamat di Java dalam penulisan subprogram, sedangkan bahasa C memilikinya. Melalui dokumen resmi dari Java disimpulkan bahwa parameter alamat ada di Java, tetapi bukan untuk tipe data primitive melainkan khusus untuk class.

Kata-kata kunci: *Java, C, subprogram, parameter*

1 PENDAHULUAN

Prosedural programming atau membagi program menjadi bagian-bagian kecil adalah konsep pemrograman yang sering digunakan sampai saat ini. Kelebihan konsep ini selain membuat program menjadi lebih pendek (karena dapat dipanggil berulang-ulang) juga lebih sederhana. Konsep prosedural programming ini pasti ada di dalam bahasa pemrograman modern saat ini. Semua bahasa pemrograman memiliki bentuk penulisan yang berbeda-beda dalam pembuatannya. Bahasa pemrograman C dan Java adalah bahasa yang paling banyak digunakan saat ini. Dalam jurnal ini akan dibahas penulisan subprogram untuk kedua bahasa tadi.

2 SUBPROGRAM

Subprogram adalah bagian kecil hasil pembagian program dalam konsep prosedural programming. Keuntungan membuat subprogram adalah mampu membuat program menjadi lebih sederhana (dibandingkan dengan seluruh program di main) dan lebih ringkas (untuk proses-proses yang sering dipanggil dapat disingkat menjadi 1 pemanggilan subprogram).

Ada 2 jenis dari subprogram, yaitu :

- a. procedure adalah subprogram yang digunakan untuk mengerjakan sesuatu (mungkin menghasilkan nilai dan mungkin tidak)
- b. function adalah subprogram yang pasti menghasilkan 1 nilai karena itu diakhiri dengan return (pengembalian nilai hasil)

Istilah-istilah umum dalam subprogram, yaitu :

- a. variabel global adalah variabel yang dapat diakses di seluruh subprogram manapun
- b. variabel lokal adalah variabel yang hanya diakses oleh subprogram yang memilikinya
- c. parameter nilai/passing by value adalah parameter yang nilainya akan kembali ke nilai sebelum subprogram dipanggil
- d. parameter alamat/passing by address adalah parameter yang nilainya dapat berubah karena pemanggilan subprogram
- e. parameter input adalah parameter yang berguna untuk sebagai nilai input/masukan yang digunakan untuk proses dalam subprogram (umumnya bersifat parameter nilai)
- f. parameter output adalah parameter yang digunakan sebagai hasil dari proses subprogram (umumnya bersifat parameter alamat)
- g. parameter input/output adalah parameter yang digunakan sebagai input maupun output (umumnya bersifat parameter alamat)
- h. parameter formal adalah parameter yang ada pada saat subprogram dibuat
- i. parameter actual adalah parameter yang ada pada saat subprogram dipanggil

3 SUBPROGRAM DALAM BAHASA C

Penulisan procedure dan function dalam bahasa C hanya dibedakan dengan tipe data(menggunakan void untuk procedure) dan return (khusus di function dan tidak ada return di procedure).

Bentuk penulisan procedure dalam bahasa C :

```
void NamaSub (Parameter){
    isi perintah
}
```

Bentuk penulisan function dalam bahasa C :

```
tipeData NamaFungsi (Parameter){
    isi perintah
    return(hasil);
}
```

Penulisan parameter dalam bahasa C hanya dibedakan dengan simbol * untuk parameter alamat dan tidak ada simbol apapun untuk parameter nilai. Sedangkan untuk pemanggilan parameter menggunakan symbol & untuk parameter alamat. Contoh subprogram dalam bahasa C :

```
int x,y,z; //variable global

void hitung(int a,int *b){ //parameter formal
    int z; //variable lokal
    a=100;
    *b=200;
    z=300;
    printf("%d %d %d\n", a, *b ,z);
}
```

```

int main()
{
    x=10;y=20;z=5;
    printf("%d %d %d\n", x,y ,z);
    hitung(x,&y); //parameter aktual
    printf("%d %d %d\n", x,y ,z);
}

```

Apabila contoh program dijalankan, maka akan muncul 3 baris dengan masing-masing 3 angka yang mewakili x,y,z. Baris 1 berisi 10 20 5, karena merupakan angka yang dideklarasikan di awal program. Baris 2 muncul 100 200 300 karena menampilkan angka yang ada di subprogram hitung. Baris 3 muncul 10 200 5, karena menampilkan x,y,z setelah pemanggilan hitung. Variabel x tetap 10, karena merupakan parameter nilai. Variabel y berubah dari 20 menjadi 200 karena parameter alamat. Variabel z kembali dari 300 menjadi 5, karena yang berubah 300 adalah z yang ada di lokal subprogram hitung, sedangkan yang ditampilkan adalah z yang dideklarasikan global.

4 SUBPROGRAM DALAM BAHASA JAVA

Penulisan subprogram di java hampir sama dengan bahasa C tetapi terdapat beberapa fitur tambahan. Secara konsep, subprogram di java akan menjadi method dari class yang ada di java. Method dari class lain dapat digunakan atau tidak tergantung dari hak aksesnya. Karena menggunakan konsep OOP (tidak memiliki pointer), maka tipe data primitif yang dijadikan parameter, hanya dapat menjadi parameter nilai saja tidak bias menjadi parameter alamat.

Salah satu fitur yang tidak ada dalam bahasa C tetapi ada di java adalah hak akses yang dituliskan di depan subprogram. Hak akses ini yang menghubungkan pengaksesan subprogram antar class dan package. Hak akses tersebut terdiri dari :

- public* dapat diakses di mana saja
- private* hanya dapat diakses di class yang sama
- protected* dapat diakses di class yang sama atau masih dari subclass atau di package yang sama
- default* (tidak dituliskan apa-apa hak aksesnya) hanya diakses di class yang sama atau di package yang sama

Tabel 1. Tingkatan akses dari hak akses

Modifier	Class	Package	Subclass	World
Public	Y	Y	Y	Y
protected	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
Private	Y	N	N	N

Class artinya ada di class yang sama

Package artinya ada di package yang sama

Subclass artinya ada di turunan dari class yang sama

World artinya beda package dan bukan sub class

Fitur tambahan subprogram di java :

- a. *Static* adalah subprogram yang dimiliki oleh class dan dapat dipanggil tanpa perlu diinstance menjadi object.
- b. *Final* adalah subprogram yang tidak dapat diubah lagi / *override*.

Contoh subprogram di java :

```
public class contohsub{
int x,y,z;    //variable global

    public void hitung(int a,contohsub c){
//parameter formal
        int z; //variable lokal
        a=100;
        c.y=200;
        z=300;
        System.out.printf("%d %d %d\n", a,t.y ,z);
    }

    public static void main(String[] args) {
        x=10;y=20;z=5;
        System.out.printf("%d %d %d\n", x,y ,z);
        hitung(x,this); //parameter aktual
        System.out.printf("%d %d %d\n", x,y ,z);
    }
}
```

Hasil program ini sama dengan contoh program bahasa C. Yang perlu diperhatikan adalah parameter alamat yang diwakili oleh class. Untuk sisanya sama dengan contoh bahasa C.

5 KESIMPULAN

Kesimpulan yang dapat ditarik dari jurnal ini adalah sebagai berikut:

- a. Java menyederhanakan konsep parameter bahwa parameter untuk variabel / tipe data primitif pasti merupakan parameter nilai dan parameter kelas pasti merupakan parameter alamat
- b. Java tidak memiliki tipe data pointer jadi tidak dapat memaksakan parameter variabel menjadi parameter alamat, tidak seperti pada bahasa C yang menuliskan simbol * sebagaipointer
- c. Urutan subprogram di java tidak mempengaruhi subprogram lain, berbeda dengan bahasa C
- d. Karena OOP lebih kompleks, maka fitur subprogram di java lebih banyak dari C

6 DAFTAR PUSTAKA

- [1] <https://docs.oracle.com>.
- [2] <http://www.javatpoint.com>.
- [3] <http://beginnersbook.com>.
- [4] <http://www.tutorialspoint.com>.
- [5] <https://www.mkkyong.com>
- [6] <https://examples.javacodegeeks.com>
- [7] <http://www.java2novice.com>