

Ruang Lingkup Variabel (*Variabel Scope*)

Pada pembelajaran dalam sub bab fungsi dan prosedur kita telah mempelajari bagaimana cara membuat suatu fungsi maupun prosedur dalam program. Dalam suatu fungsi dan prosedur dideklarasikan pula beberapa variabel, baik variabel yang dideklarasikan **di dalam sub program** fungsi maupun prosedur, maupun variabel yang dideklarasikan **di luar sub program** fungsi maupun prosedur.

Masing-masing variabel tersebut, baik yang dideklarasikan di dalam maupun luar fungsi maupun prosedur memiliki **ruang lingkup** atau jangkauan tertentu atau sering disebut ***variable scope***. *Variabel scope* atau ruang lingkup variabel adalah jangkauan akses suatu variabel.

Contoh 1:

```
#include <iostream>
using namespace std;

int hitungLuas (int panjang, int lebar) {
    int luas;
    luas = panjang * lebar;
    return luas;
}

int main() {
    int p = 7; int l = 5;
    int luasPersegi;
    luasPersegi = hitungLuas(p, l);
    cout << "Luas = " << luas << endl;
}
```

Variabel dideklarasikan di dalam fungsi

Program utama mengakses variabel luas yang terdapat pada fungsi

Pada program tersebut dideklarasikan beberapa variabel salah satunya adalah variabel luas yang dideklarasikan **di dalam fungsi** hitung Luas. Lalu, pada contoh tersebut program utama mengakses variabel luas yang dideklarasikan **di dalam fungsi**. Ketika program tersebut dieksekusi maka nilai keluaran program adalah sebagai berikut:

Hasil eksekusi:

```
Error: undefined
```

Berdasarkan contoh program tersebut, ketika program dieksekusi program tersebut menampilkan *error*. Kata-kata **error: undefined** menunjukkan bahwa variabel luas telah diakses oleh program utama namun variabel ini belum didefinisikan dalam program utama, sehingga variabel tersebut tidak dikenali oleh program utama. Artinya, variabel luas yang didefinisikan didalam fungsi tidak berlaku pada program utama.

Variabel luas dalam program tersebut merupakan **variabel lokal**. Variabel lokal merupakan variabel yang dideklarasikan **didalam fungsi maupun prosedur**. Oleh karena itu, ketika variabel luas digunakan oleh bagian program lain di luar fungsi dalam hal ini program utama, variabel luas tidak dapat dikenali. Sebab variabel lokal hanya dapat variabel dapat dikenali dan diakses oleh fungsi maupun prosedur yang mendeklarasikannya.

Contoh 2:

```
#include <iostream>
using namespace std;

int luas;

int hitungLuas(int panjang, int lebar){
    luas = panjang * lebar;
    return luas;
}

int main(){
    int p = 7; int l = 5;
    int luasPersegi;
    luasPersegi = hitungLuas(p, l);
    cout << "Luas = " << luas << endl;
}
```

Variabel dideklarasikan di luar fungsi

Program utama mengakses variabel luas yang dideklarasikan di luar fungsi

Dengan program yang seperti pada contoh 1, namun kini variabel luas dideklarasikan di luar fungsi maupun program utama. Ketika program utama akan mengakses variabel luas tersebut maka hasil eksekusi program sebagai berikut;

```
Hasil eksekusi:
Luas = 35
```

Ketika variabel dideklarasikan di luar sub program variabel tersebut kini dapat diakses oleh program utama, sehingga nilai dari variabel luas dapat tercetak ke dalam layar. Pada program jika suatu variabel dideklarasikan **di luar fungsi maupun prosedur**, maka variabel tersebut dapat dikenali dan diakses secara global oleh seluruh bagian program baik fungsi maupun prosedur termasuk program utama. Variabel yang dideklarasikan di luar fungsi maupun prosedur dinamakan sebagai **variabel global**.

RUANG LINGKUP VARIABEL

Berdasarkan beberapa contoh program yang telah diuraikan, dapat diketahui bahwa suatu variabel memiliki jangkauan atau *scope* tertentu. Jangkauan atau ruang lingkup variabel tersebut antara lain:

1. VARIABEL LOKAL

Variabel lokal merupakan variabel yang dideklarasikan di dalam fungsi maupun prosedur. Variabel lokal hanya dikenali dan diakses secara lokal dalam sebuah sub program (fungsi atau prosedur) yang mendeklarasikannya. **Variabel lokal tidak dapat dipanggil, diakses dan diubah oleh bagian program yang lain termasuk program utama.** Sifat variabel lokal adalah sebagai berikut;

- Variabel lokal berada di dalam blok suatu fungsi maupun prosedur.
- Variabel lokal tidak dapat dikenali dan digunakan oleh bagian program lain di luar blok atau bagian program yang mendeklarasikannya.
- Variabel lokal hanya bisa diakses oleh bagian program tempat variabel tersebut dideklarasikan

2. VARIABEL GLOBAL

Variabel global merupakan variabel yang dideklarasikan di luar sub program (fungsi maupun prosedur) maupun program utama. Variabel global dapat dikenali oleh seluruh bagian program. **Variabel global dapat dipanggil, diakses dan diubah oleh bagian program yang lain termasuk program utama.** Sifat variabel global adalah sebagai berikut;

- Variabel global berada di luar blok suatu fungsi dan prosedur maupun program utama.
- Variabel global dapat dikenali dan digunakan oleh seluruh bagian blok program.
- Variabel global dapat diakses oleh seluruh bagian blok program.

VARIABEL GLOBAL

Variabel global merupakan variabel yang dideklarasikan di luar sub program (fungsi maupun prosedur) maupun program utama dan dapat diakses oleh seluruh bagian program. Penggunaan variabel global pada program memang terlihat mudah, karena satu variabel dapat dikenali oleh seluruh bagian program sehingga transfer data antara fungsi maupun prosedur menjadi lebih mudah. Namun penggunaan variabel global memiliki kekurangan. Sebagai contoh, perhatikan program berikut:

```
#include <iostream>
using namespace std;

int a, b;

int hitung(){
    a = 10;
    b = 16

    cout << "Nilai a pada prosedur: " << a
    <<endl;
    cout << "Nilai b pada prosedur : " << b
    <<endl;
}
int main (){
    a = 5;
    b = 9;

    cout << "Nilai awal a: " << a << endl;
    cout << "Nilai awal b: " << b << endl;

    hitung();

    cout << "Nilai akhir a: " << a << endl;
    cout << "Nilai akhir b: " << b << endl;
}
```

Variabel Global

Ketika variabel a dan b dideklarasikan secara global di luar fungsi. Maka nilai keluaran dari program adalah:

Hasil eksekusi:

```
Nilai awal a: 5  
Nilai awal b: 9  
Nilai a pada prosedur: 10  
Nilai a pada prosedur: 16  
Nilai akhir a: 10  
Nilai akhir b: 16
```

Variabel a dan b pada program utama memiliki nilai awal 10 dan 16. Ketika fungsi dipanggil dan dijalankan nilai variabel a dan b pada fungsi `hitung()` dirubah nilainya, lalu nilai variabel a dan b pada program utama pun berubah. Pada variabel global, perubahan nilai variabel pada suatu bagian fungsi akan berpengaruh pada nilai variabel pada bagian program lain.

Sehingga penggunaan variabel cenderung menyebabkan terjadinya *bug*, karena nilai variabel dapat dirubah oleh bagian blok program yang lain dalam suatu program.

VARIABEL LOKAL

```
#include <iostream>
using namespace std;

int hitung(a, b){
    int a, b;
    a = 10; b = 16;

    cout << "Nilai a pada prosedur: " << a
    << endl;
    cout << "Nilai b pada prosedur: " << b
    << endl;
}

int main (){
    int a, b;
    a = 5; b = 9;

    cout << "Nilai awal a: " << a << endl;
    cout << "Nilai awal b: " << b << endl;

    hitung();

    cout << "Nilai akhir a: " << a << endl;
    cout << "Nilai akhir b: " << b << endl;
}
```

Variabel
Lokal

Variabel
Lokal

Hasil eksekusi:

```
Nilai awal a: 5
Nilai awal b: 9
Nilai a pada prosedur: 10
Nilai a pada prosedur: 16
Nilai akhir a: 5
Nilai akhir b: 9
```

Pada contoh program tersebut, baik program utama maupun fungsi mendeklarasikan variabel lokal yang bernama sama yaitu variabel (a dan b), tetapi antara variabel yang ada di program utama dan fungsi hitung **tidak saling mempengaruhi** meskipun bernama sama karena variabel tersebut hanya berlaku pada tempat ia dideklarasikan.

Setelah program utama memanggil fungsi hitung, perubahan variabel a dan b pada fungsi hitung() tidak berpengaruh pada program utama. Sehingga nilai variabel a dan b pada program utama sebelum dan sesudah pemanggilan fungsi bernilai tetap. Dengan pendeklarasian variabel secara lokal, perubahan nilai variabel pada satu bagian program tidak berpengaruh pada nilai variabel dalam bagian program lainnya.

VARIABEL LOKAL ATAU VARIABEL GLOBAL?

Keputusan apakah suatu variabel dideklarasikan secara global atau lokal bergantung kepada penggunaan variabel tersebut. Bila suatu variabel digunakan di seluruh bagian program, maka variabel dideklarasikan secara global. Sebaliknya, bila variabel hanya digunakan dalam fungsi atau prosedur saja sebaiknya dideklarasikan secara lokal.

Usahakanlah menggunakan variabel global sesedikit mungkin, karena penggunaan variabel global cenderung tidak aman dan akan membuat *bug* pada program. Hal itu disebabkan karena perubahan nilai pada suatu bagian program akan mempengaruhi nilai variabel pada bagian program lain. Penggunaan variabel lokal memberikan keuntungan sebab variabel lokal dapat meminimumkan usaha pencarian kesalahan atau *error* yang disebabkan oleh variabel tersebut.