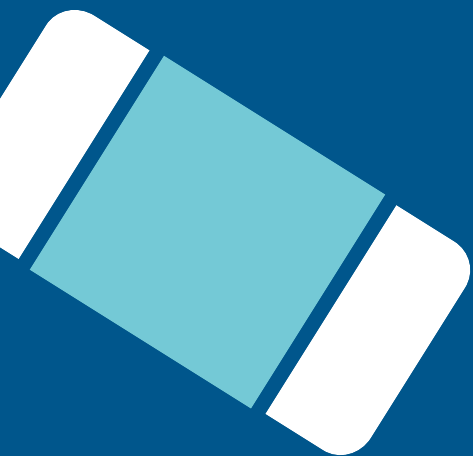
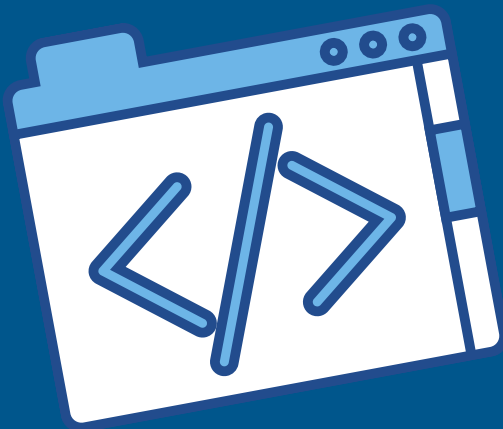
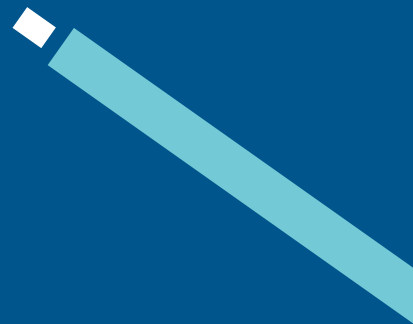
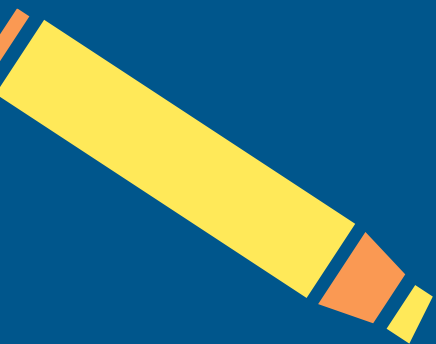


RUANG LINGKUP VARIABEL



VARIABEL PADA PROGRAM

Dalam sebuah program dengan fungsi dan prosedur dideklarasikan pula variabel pada program. Baik variabel yang dideklarasikan **di dalam fungsi**, maupun variabel yang dideklarasikan **di luar fungsi**.

Pada contoh program berikut, program mendefinisikan fungsi hitungLuas. Selain itu program mendeklarasikan beberapa variabel antara lain: variabel luas yang dideklarasikan **di dalam fungsi**, serta variabel p, l dan luasPersegi yang dideklarasikan **di luar fungsi** (program utama).

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

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

int main(){
    int p, l, luasPersegi;

    luasPersegi = hitungLuas(p, l);
    cout << "Luas = " << luas << endl;
}
```

Perhatikan contoh program berikut!

```
#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;
}
```

Hasil eksekusi:

Error: undefined

Pada program tersebut dideklarasikan beberapa variabel salah satunya adalah variabel luas yang dideklarasikan **di dalam fungsi** hitung Luas. Lalu, program utama mengakses variabel luas yang dideklarasikan di dalam fungsi.

Sehingga, ketika program tersebut dieksekusi program menampilkan pesan *error*. Kata-kata **error: undefined** mengimplikasikan bahwa variabel luas telah dipanggil di program utama namun variabel ini belum dideklarasikan. Artinya, variabel luas yang dideklarasikan didalam fungsi fungsi tidak berlaku di bagian program lain di luar fungsi.

Variabel luas dalam program tersebut merupakan **variabel lokal**. Oleh karena itu, ketika variabel lokal digunakan oleh bagian program lain di luar fungsi maka variabel tidak dapat dikenali. Variabel lokal tidak akan dikenali oleh bagian program lain di luar fungsi yang mendeklarasikannya.

Perhatikan contoh program berikut!

```
#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;
}
```

Hasil eksekusi:
Luas = 35

Dengan program yang sama, kini variabel luas dideklarasikan **di luar** fungsi maupun program utama.

Ketika program utama akan mengakses variabel luas tersebut maka hasil eksekusi program adalah berupa hasil perhitungan luas. Variabel ini dapat diakses program utama, sehingga nilai dari variabel luas dapat tercetak ke dalam layar.

Pada program jika suatu variabel dideklarasikan **di luar fungsi**, maka variabel tersebut dapat dikenali secara global oleh seluruh bagian program termasuk program utama. Variabel yang dideklarasikan di luar fungsi maupun prosedur dinamakan sebagai **variabel global**.

RUANG LINGKUP VARIABEL

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

A. VARIABEL LOKAL

Variabel lokal merupakan variabel yang dideklarasikan **di dalam** fungsi maupun prosedur. Variabel lokal **hanya dikenali 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.**

B. VARIABEL GLOBAL

Variabel global merupakan variabel yang dideklarasikan **di luar** sub program (fungsi maupun prosedur). Variabel global **dapat dikenali oleh seluruh bagian program.**

Variabel global **dapat dipanggil, diakses dan diubah oleh bagian program yang lain termasuk program utama.**

Penggunaan variabel global pada program memang terlihat mudah, karena satu variabel dapat dikenali oleh seluruh bagian dari program dan transfer data antar fungsi menjadi lebih mudah. Namun penggunaan variabel global dalam fungsi juga memiliki beberapa 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;
}
```

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
```

Ketika nilai variabel a dan b pada fungsi hitung() berubah nilainya, 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.

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;
}
```

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, program utama dan fungsi luas mendeklarasikan variabel yang sama (a dan b) secara lokal, tetapi antara variabel yang ada di program utama dan fungsi hitung **tidak saling mempengaruhi**.

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. Penggunaan variabel lokal memberikan keuntungan sebab variabel lokal dapat meminimumkan usaha pencarian kesalahan yang disebabkan oleh variabel tersebut.