

PERULANGAN

OLEH : AKHI HARUNI



Mengapa diciptakan algoritma perulangan?

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     cout << "SAYA BISA";
6     cout << "SAYA BISA";
7     cout << "SAYA BISA";
8     cout << "SAYA BISA";
9     cout << "SAYA BISA";
10    cout << "SAYA BISA";
11    cout << "SAYA BISA";
12    cout << "SAYA BISA";
13    .....
14    .....
15    hingga 1000 kali?
16 }
```

Ketika kamu akan menampilkan motivasi diri "SAYA BISA" sebanyak 1000 kali, mungkinkah kamu menuliskannya sebanyak 1000 kali juga?

Jawabannya adalah MUNGKIN, namun TIDAK EFISIEN KODE PROGRAM. Oleh karena itu, untuk meringkas kode program kita, diciptakanlah algoritma perulangan.

Perulangan FOR

- ❑ Ada banyak macam algoritma perulangan. Untuk saat ini akan dijelaskan perulangan FOR
- ❑ Struktur dasar :

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     for (batasAwal ; batasAkhir ; increment/decrement) {
6         .....
7         pernyataan;
8     }
```

batasAwal merupakan batas awal perulangan

batasAkhir merupakan batas akhir perulangan (mau diulang berapa kali? Dihitung dari batas awal)

Increment/decrement merupakan penambahan atau pengurangan

Contoh

Tampilkan warna favoritmu sebanyak 7 kali dengan batas awal 0! Maka sintaknya...

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     int a; //mendeklarasikan variabel untuk FOR
6     for (a = 0 ; a < 7 ; a++) {
7         cout << "Maroon";
8     }
9 }
```



```
Maroon
Maroon
Maroon
Maroon
Maroon
Maroon
Maroon
-----
Process exited af
Press any key to
```

Penjelasan KODE PROGRAM

- ❑ **int a** merupakan deklarasi variabel untuk for. Jika tidak ada deklarasi tersebut, maka program akan error dengan tulisan **a was not declared in this scope**. Alternatif lain adalah memasukkannya langsung ke dalam sintak FOR menjadi for (int a = 0; a
- ❑ **a = 0** berarti batas awal perulangan adalah 0. Batas tersebut nantinya digunakan sebagai patokan. **a < 7** berarti batas akhirnya adalah kurang dari 7 (atau bisa kita sebut 6) sehingga jumlah pernyataan yang akan tampil sebanyak : 0, 1, 2, 3, 4, 5, 6 (7 kali)
- ❑ **a++** berarti setiap proses selesai dilakukan, maka akan ada penambahan 1 pada variabel a, maksudnya seperti ini :
 - ❑ a = 0, apakah a kurang dari 7? Ya, maka tambahkan 1 pada a (a++) kemudian tampilkan pernyataan
 - ❑ a = 1, apakah a kurang dari 7? Ya, maka tambahkan 1 pada a (a++) kemudian tampilkan pernyataan
 - ❑ a = 2dan seterusnya sampai a < 7

Bagaimana dengan bilangan kelipatan?

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     int a; //mendeklarasikan variabel untuk FOR
6     //menampilkan kelipatan 5 antara 0 sampai 25
7     for (a = 5 ; a < 25 ; a=a+5) {
8         cout << a << endl;
9     }
10 }
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     int a; //mendeklarasikan variabel untuk FOR
6     //menampilkan kelipatan 5 antara 1 sampai 25
7     for (a = 1 ; a < 25 ; a++) {
8         if (a%5==0) {
9             cout << a << endl;
10        }
11    }
12 }
```

Bagaimana dengan bilangan ganjil genap?

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     int a; //mendeklarasikan variabel untuk FOR
6     //menampilkan bilangan ganjil antara 0 sampai 25
7     for (a = 0 ; a < 25 ; a++) {
8         if (a%2==1) {
9             cout << a << endl;
10        }
11    }
12 }
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main (){
5     int a; //mendeklarasikan variabel untuk FOR
6     //menampilkan bilangan genap antara 1 sampai 25
7     for (a = 1 ; a < 25 ; a++) {
8         if (a%2==0) {
9             cout << a << endl;
10        }
11    }
12 }
```