

Program Berstruktur Perulangan

Perulangan merupakan struktur dimana terdapat proses pengulangan perintah yang sama sebanyak n kali. Struktur ini merupakan salah satu kelebihan yang dimiliki oleh mesin komputer. Sebagai contoh untuk menampilkan teks "Belajar Pemrograman" sebanyak 10 kali pada layar monitor, hanya diperlukan beberapa baris perintah menggunakan teknik atau struktur perulangan tersebut. Tanpa harus menuliskan perintah yang sama sebanyak 10 kali.

Contoh program di bawah ini adalah untuk menampilkan deret bilangan 0 1 2 3 4 5 menggunakan perintah for – do. Dimana perintah akan berulang selama nilai yang tersimpan dalam variabel (bil) berada dalam rentang 0 sampai dengan 5. Untuk lebih memahami materi, simak kembali pembahasan perintah for – do tersebut pada bab sebelumnya.

```
#include <iostream>
using namespace std;
main() {
    int bil;
    for(bil=0 ; bil<=5 ; bil++){
        cout << bil << " "; }
}
```

Program selanjutnya merupakan pengembangan dari program di atas, yaitu dengan memasukkan jumlah deret yang ingin ditampilkan. Terdapat dua variabel yang digunakan yaitu bil dan deret. Salinlah program di bawah ini pada aplikasi Dev C++ dan perhatikan outputnya.

```
#include <iostream>
using namespace std;
main() {
    int bil, deret;
    cout << "Masukkan jumlah deret angka : ";
    cin >>deret;
    for(bil=0 ; bil<=deret ; bil++){
        cout << bil << " "; }
}
```

Untuk menampilkan deret bilangan bulat tertentu, misal deret bilangan genap diantara 0 sampai dengan 100, maka diperlukan rumus deret yang ditulis dalam ekspresi numerik atau notasi algoritmik tertentu. Pada program di bawah ini, terdapat ekspresi numerik yang menggunakan operator modulus untuk mencari sisa pembagian (simbol %). Ekspresi numerik "**bil % 2 == 0**" menyatakan bilangan yang habis dibagi 2 merupakan bilangan genap.

```

#include <iostream>
using namespace std;
main()
{
    int bil;
    for(bil=0 ; bil<=100 ; bil++)
    {
        bil % 2 == 0; bil++;
        cout << bil << " ";
    }
}

```

Contoh program selanjutnya berstruktur kombinasi pemilihan dan perulangan, yang merupakan pengembangan dari program deret bilangan genap di atas.

```

#include <iostream>
using namespace std;
int main() {
    int pil, batas;

    cout<<"Masukan Pilihan [1.Ganjil] [2.Genap] : ";
    cin>>pil;
    cout<<"Masukan Batas : ";
    cin>>batas;
    cout<<endl;

    if (pil==1)
    {
        cout<<"Deret Bilangan Ganjil"<<endl;
        for(int i=1;i<=batas;i++)
        {
            if (i % 2 != 0)
                cout<<i<<" ";
        }
    }
    else {
        cout<<"Deret Bilangan Genap"<<endl;
        for(int i=1;i<=batas;i++){
            if (i % 2 == 0)
                cout<<i<<" ";
        }
    }
}

```

Pada program di atas terdapat pilihan di dalamnya, jika pilihan = 1 maka akan ditampilkan deret bilangan ganjil, dan jika pilihan =2 maka akan ditampilkan deret bilangan genap. Dan terdapat dua variabel dengan "pil" dan "batas", yang berfungsi untuk menyimpan nilai pilihan dan batas deret bilangan yang ingin ditampilkan.

Ekspresi numerik "***bil % 2 == 0***" menyatakan bilangan yang habis dibagi 2 alias bilangan genap. Sebaliknya untuk menyatakan bilangan yang tidak habis dibagi 2 alias bilangan ganjil, maka ekspresi numerik menjadi "***bil % 2 != 0***" (menggunakan operator relasional ***!=*** yang berarti "**tidak sama dengan**").

Selain untuk menampilkan deret bilangan yang berurutan secara menaik (ascending) di atas, juga dapat ditampilkan secara menurun (descending), misal deret bilangan bulat 10 9 8 7 6 5 4 3 2 1, atau deret bilangan ganjil 9 7 5 3 1.

Berikut program untuk menampilkan deret bilangan menurun 10 sampai dengan 1 :

```
#include <iostream>
using namespace std;
main() {
    int bil;
    for(bil=10 ; bil<=1 ; bil--)
    {
        cout << bil << " "; }
    }
```

Perhatikan perintah for yang tercetak tebal (***bil=10; bil<=1; bil--***) pada program di atas, yang berarti perintah perulangan dimulai dari 10 (bil=10) sebagai nilai awal, dilakukan berulang secara menurun satu persatu (bil--), dan berakhir saat variable bil <=1 (nilai akhir). Sehingga jika program tersebut dijalankan dapat ditampilkan deret bilangan bulat menurun 10 9 8 7 6 5 4 3 2 1.

Selain perintah for – do, untuk membuat program berstruktur perulangan juga dapat menggunakan perintah while dan do – while. Berikut contoh program berstruktur perulangan menggunakan while dan do-while.

```
#include <iostream>
using namespace std;
int main() {
    int a = 1;
    while(a <= 10) {
        cout<<a<<endl;
        a++; }
    }
```

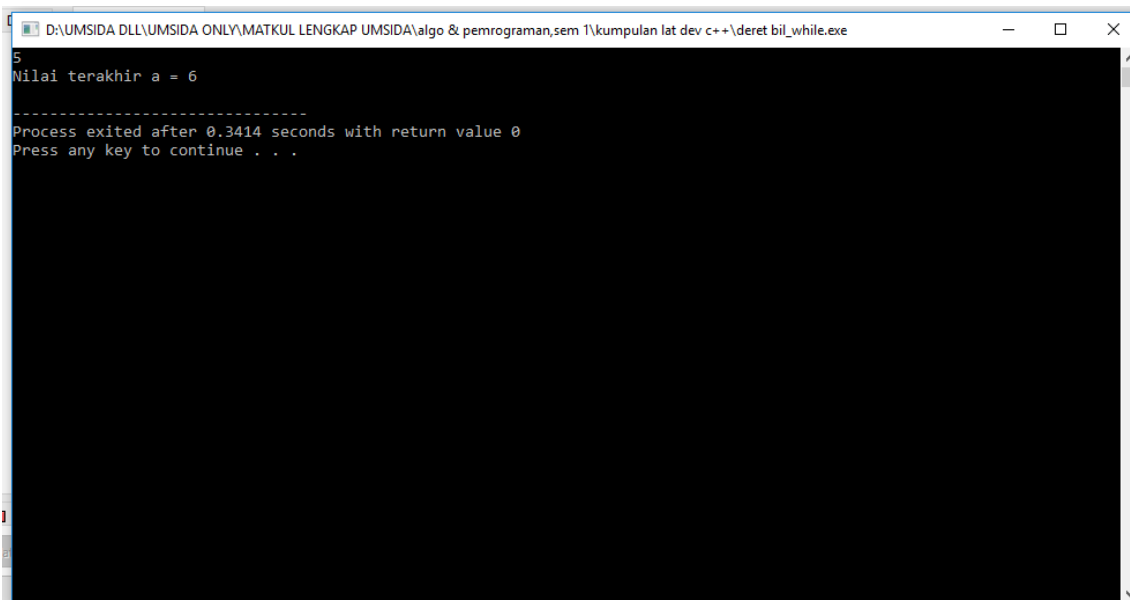
Program while di atas untuk menampilkan deret bilangan bulat dari 1 sampai dengan 10. Berbeda dengan perintah for-do dimana nilai awal dan nilai akhir ditulis dalam satu kesatuan perintah, maka pada perintah while diberi nilai awal saja (int a =1). Setelah itu terdapat perintah untuk mengecek kondisi yang ditentukan, dan perintah berikutnya akan dikerjakan selama kondisi tersebut bernilai benar (***while(a<=10)***).

Sedangkan pada program do-while, semua perintah dikerjakan berulang terlebih dahulu, dan dihentikan jika kondisi yang ditentukan sudah terpenuhi atau bernilai benar. Seperti yang tampak dalam pembahasan program di bawah ini.

Perhatikan outputnya pada gambar di bawah ini, mengapa hasilnya demikian ?

```
#include <iostream>
using namespace std;
int main() {
    int a = 5;
    do {
        cout<<a<<endl;
        a++;
    } while(a <= 4);
    cout<<"Nilai terakhir a = "<<a<<endl;
}
```

Output :
nilai terakhir a = 6



Output Program Do-While 1

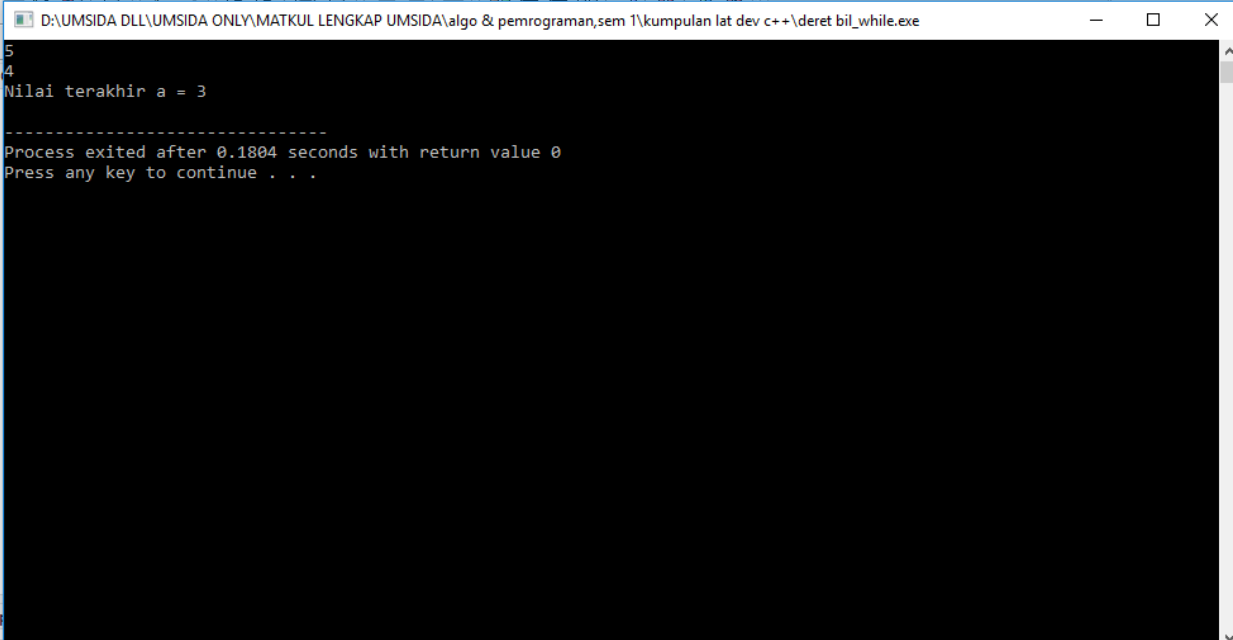
Jika beberapa perintah pada program di atas diganti menjadi sebagai berikut, maka tampak perbedaan outputnya seperti yang tampak dalam gambar di bawah ini.

```
#include <iostream>
using namespace std;
int main() {
    int a = 5;
    do { cout<<a<<endl;
        a--;
    } while(a >= 4);
    cout<<"Nilai terakhir a = "<<a<<endl; }
```

Output:
5
4
Nilai terakhir a = 3

Perhatikan perintah yang dicetak tebal, counter / pencacahnya diganti menjadi **a--** (decrement / menurun), dimana pada program sebelumnya pencacahnya menaik **a++** (increment), dan kondisi yang ditentukan menjadi (**a>=4**). Maka outputnya menjadi seperti dalam gambar di bawah ini.

Mengapa demikian?



```
D:\UMSIDA DLL\UMSIDA ONLY\MATKUL LENGKAP UMSIDA\algo & pemrograman,sem 1\kumpulan lat dev c++\deret bil_while.exe
5
4
Nilai terakhir a = 3
-----
Process exited after 0.1804 seconds with return value 0
Press any key to continue . . .
```

Output Program Do-While 2