Pertemuan	**:**	**6 (Enam)**
Pokok Bahasan	**:**	**Struktur dasar lanjut (Pengulangan/ Looping/ Repetition)**
Tujuan Khusus	**:**	**Mahasiswa mampu menggunakan struktur dasar pengulangan**

---

1. Pendahuluan

Kelebihan yang dimiliki mesin atau dalam hal ini komputer bila dibandingkan manusia adalah keampuannya melakukan tugas yang sama berulang kali tanpa mengeluh lelah dan bosan.Di dalam algoritma, melakukan proses berulang disebut dengan pengulangan atau looping atau repetition jika suatu konsisi dipenuhi atau sebaliknnya.

2. Struktur Pengulangan

Struktur pengulangan terdiri atas dua bagian:

1. kondisi pengulangan, yaitu ekspresi bolean yang harus dipenuhi untuk melaksakan pengulangan.
2. Isi atau badan pengulangan, yaitu satu atau lebih pernyataan (aksi) yang akan diulang.

Di samping itu, struktur pengulangan biasanya disertai dengan bagian:

1. inisialisasi, yaitu pernyataan yang dilakukan sebelum pengulangna dilakukan pertama kali
2. terminasi, yaitu aksi yang dilakukan setelah pengulangan selesai dilaksanakan

inisialisasi dan teminasi tisak selalu harus ada, namun berbagia kasus inisalisasi umumnya diperlukan.

Perintah atau notasi dalam struktur pengulangan adalah:

1. Perintah for
2. Perintah do .. while
3. Perintah while
4. Perintah Continue dan break
5. Peritnah Go To

2.1. Perintah for

Perintah for digunakan untuk menghasilkan pengulangan sejumlah kali tanpa penggunaan kondisi apapun. Perintah ini dapat digunakan bila anda sudah tau berapa kali Andaakan mengulang satu atau beberapa pernyataan. Bentuk umum pernyataan for adalah

Dalam algoritma

```
For peubah nilai ← nilai awal  to nilai akhir di
        Aksi
End
```

Dalam bahasa pemrograman C++

```
For (peubah=nilai awal; peubah=nilai akhir, keadaan)
{
        Aksi;
}
```

keterangan:

a. peubah ← nilai awal mempunyai arti sebagai inisialisasi
b. peubah ← nilai akhir mempunyai arti sebagai terminasi/konsidi untuk keluar dari looping
c. keadaan mempunyai dua pengertian menaik (ascending) atau menurun (descending)
d. Jumlah aksi diulang sebanyak nilai akhir- nilai awal + 1

Kasus 6.1

Buatlah algoritma dan program untuk mencetak nama Anda sebanyak 10 kali

Jawab:

Algoritm untuk memecahkan masalah tersebut di atas adalah:

1. k bertipe integer
2. for k =1 to 10 do
3. cetak “Nama Anda”
4. end

Program

```
#include<iostream.h>

void main()
{
int k;
for(k=1;k<=10;k++)
cout <<"Nama Anda\n";
}

Hasil program:
```

Latihan 1:

1. Buatlah program untuk mencetak deret 1 – 10 ?

   1 3 5 7 9 11 13 15 17 19

2. Buatlah program untuk mencetak deret genap 10 suku?

   2 4 6 8 10 12 14 16 18 20

3. Buatlah program untuk mencetak deret kuadrat 5 suku?

   1 4 9 16 25

2.2. Perintah Do..While

Perintah ini menyatakan pengulangan proses selama kondisi tertentu. Bentuk umumnya adalah sebagai berikut:

```
Do
   Pernyataan;
While (keadaan);

Atau
Do
{
   Pernyataan;
   Pernyataan;
}
while (keadaan)
```

Setiap pengulangan dikerjakan maka kondisi akan dicek. Jika masih benar, proses loop dilakukan lagi dan jika salah maka proses loop berhenti dan berlanjut pada perintah selanjutnya. Lebih jelaskan perhatikan contoh berikut ini:

Kasus 6.2

Buatlah algortima dan pemrogramn untuk mencetak tanggal lahir Anda sebanyak 10 kali.

Jawab:

Algoritma dari permasalahan di atas adalah:

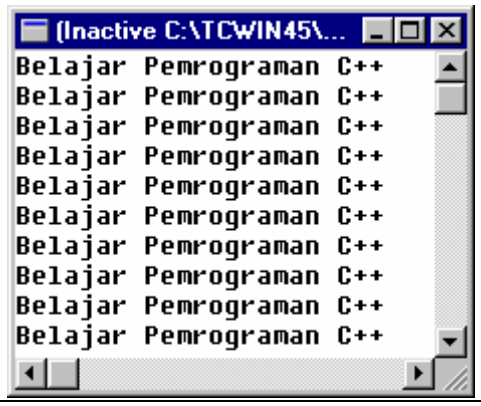
1. Tentukan nilai awal perhitungan = 0
2. Lakukan pengulangan
3. Cetak tanggal lahir
4. Nilai perhitungan bertambah Satu
5. Ulangi langkah sampai nilai perhitungan kurang dari 10

Program menggunakan Do..While

```
#include<iostream.h>
void main()
{
int A=0;
do
{
cout<<"Belajar Pemrograman C++ \n";
A++;
}
while (A<10);
```

```
}
```

Hasil Program:



Latihan 2

Kerjakan latihan 1 di atas dengan menggunakan perintah Do..while, tapi deret tersebut menurun.

2.3. Perintah While

Perintah While ini prinsipnya sama dengan perintah Do..While hanya pengujian kondisinya terletak pada awal loop.

Bentuk umumnya adalah sebagai berikut:

```
While (keadaan)
{
        Pernyataan;
        Pernyataan;
        …
}
```

Pernyataan dapat berupa pernyataan tunggal atau beberapa pernyataan yang dibatasi dengan tanda {}. Pernyataan itu akan dijalankan jika kondisinya masih benar.

Kasus 3:

Buatlah algoritma dan pemrograman untuk mencetak deret berikut ini :

10 9 8 7 6 5 4 3 2 1

Algoritma untuk kasus di atas:

1. Tentukan nilai awal
2. Lakukan proses pengulangan selama i > 0

3. Cetak bilangan
4. Ulangi langkah 2 sampai batas akhir

Program untuk masalah tersebut adalah:

```
#include<iostream.h>
void main()
{
int A=10;
while (A>0)
{
cout<<A<<"    ";
A--;
}
}
```

2.4. Perintah continue dan break

Pada beberapa kasus, Anda mungkin ingin menghentikan suatu pengulangan di tengah jalan untuk menghentikan pengulangan, gunakanlah pernyataan break. Anda bisa menggunakan pernyataan break pada pernyataan for, do..while dan while do. Perhatikan program berikut ini:

```
#include<iostream.h>
void main()
{
int x=0;
while (x>=0)
{
x++;
cout<<x<<"\n";
if (x>100)
break;
}
}
```

Bila dilihat sekilas, looping di atas seperti tidak akan pernah berakhir karena x selalu ditambah 1 padahal kondisi yang harus dipenuhi adalah (x>=0). Namun bila dijalankan, setelah x lebih besar dari pada 100, pengulangan akan berhenti. Hal ini dikarenakan adanya pernyataan break.

Pernyataan continue adalah bentuk pernyataaan khusus lain yang digunakan pada pengulangan. Pernyataan ini digunakan untuk mengembalikan aliran program ke

pengujian kondisi pengulangan. Dengan kata lain, pernyataan di bawah continue akan diabaikan.

Berikut ini contoh program yang menggunakan perintah Continue:

```
#include<iostream.h>
void main()
{
int i=0;
for (i=0;i<=50;i++)
{
if(i%3!=0)
continue;
cout<<i<<"   ";
}
}
```

Program tersebut di atas akan menghasilkan semua bilangan kelipatan iga yang kurang dari 50. Mengapa bisa demikian? Karena pada saat nilai pencacah pengulangan atidak habis dibagi 3, program akan memeanggil pernyataan continue, akibatnya cout<<i<<"   "; tidak dilaksanakan.

Kasus 4

Buatlah algoritma dan program untuk menampilkan bilangan dari 0 sampai 6, ketika proses pencetakan bilangan pada 4, bilangan 4 ini akan dilewati dan dilanjutkan dengan bilangan 5 dan pada bilangan 6 program akan berhenti sehingga hasil terakhir adalah 0, 1, 2, 3, 5, 6

Algoritma dari kasus di atas adalah sebagai berikut:

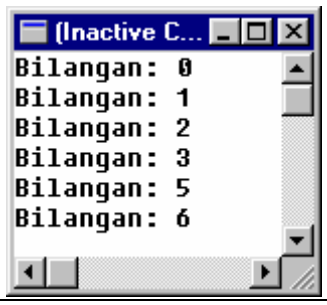
1. Tentukan nilai awal, batas akhir dan pertambahan nilai
2. Lakukan pengulangan sesuai dengan langkai
3. Jika dalam pengulangan bilangan bernilai dengan 4, maka pengulangan berhenti dan dilanjutkan ke bilangan selanjutnya.
4. Cetak bilangan
5. jika dalam pengulangan bernila 6, maka pengulangan dihentikan

Program untuk kasus di atas adalah:

```
#include<iostream.h>
void main()
{
```

```
int i=0;
for (i=0;i<10;i++)
{
if(i==4)
continue;
cout<<"Bilangan: "<<i<<"\n";
if (i==6)
break;
}
}
```

Hasil dari program itu adalah:



Dari program di atas, dapat dilihat pengulangan sari suatu bilangan sebannyak 10 kali. Etapi pada pengulangan i=4, ada perintah continue. Dengan perintah ini, maka program langsung meloncat ke loop berikutnya dan ketika sampai pengulangan i=6 ada perintah break. Otomatis program akan berhenti dan tidak sampai ke i=10.

2.5. Perintah GOTO

Perintah GOTO digunakan untuk mengalihkan proses menuju ke suatu tabel tertentu. Lebih jelaskan perhatikan contoh berikut ini:

Kasus 5

Buatlah algoritma dan program untuk menampilkan tulisan “C++ sangat mudah sekali” sebanyak 10 kali.

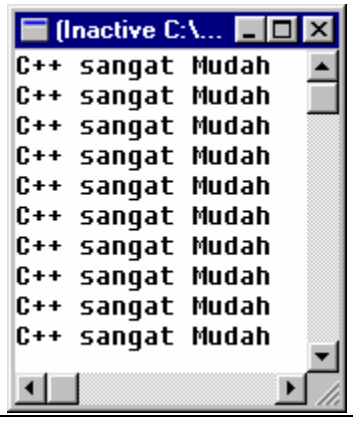
Algoritma kasus tersebut adalah:

1. Tentukan nilai awal pencetakan = 1
2. Tentukan nama label dimana kondisi untuk pencetakan tulisan dikenakan padanya
3. Cetak tulisan
4. Lakukan pengujian, jika nilai awal pencetakan kurang dari 10 ulangi langkah 3
5. Jika nilai pencetakan lebih dari 10, proses pencetakan dihentikan dan program akan selesai.

Program untuk kasus di atas adalah

```
#include<iostream.h>
void main()
{
int n=1;
cetak:
cout<<"C++ sangat Mudah\n";
if (n++<10)
goto cetak;
}
```

Hasil dari program tersebut di atas adalah:



LATIHAN 3

1. Rancanglah algoritma dan program untuk menampilkan semua bilangan ganjil kurang dari 100?
2. Tuliskan program untuk menampilkan pola-pola bintang seperti berikut ini:

   *

   * *

   * * *

   * * * *

   n = 4
3. Buatlah algoritma dan program untuk menentukan vokal dan konsonan dari kalimat yang diinputkan?
4. Buatlah program untuk mencari bilangan terbesar dari 5 buah bilangan yang diinputkan?
5. Buatlah algoritma dan program untuk menentukan sisa hasil bagi pembagian antara bilangan yang dimasukkan dengan bilangan pembagi. Apabila sisa baginya=0 maka dicetak tidak ada dan kalu ada sisa baginya, maka sisa bagi tersebut ditampilkan.?
6. Tuliskan program untuk menampilkan pola-pola bintang seperti berikut ini:

   1 = 1

   1+2 = 3

   1+2+3 = 6

   1+2+3+4 = 10