

## **MODUL 3**

### **PERCABANGAN TINGKAT LANJUT DAN PERULANGAN**

#### **A. Tujuan Praktikum :**

Setelah mempelajari bab ini mahasiswa diharapkan mampu :

1. Menjelaskan tentang fungsi perulangan
2. Menjelaskan tentang fungsi percabangan tingkat lanjut

#### **B. Waktu Praktikum**

4 X 1 jam

#### **C. Petunjuk**

1. Awali setiap kegiatan dengan berdoa, agar diberi kelanjutan dalam belajar dan mendapatkan ilmu yang bermanfaat.
2. Pahami tujuan praktikum, dasar teori, dan latihan – latihan praktikum dengan baik
3. Kerjakan tugas praktikum dengan baik, jujur, sabar dan teliti
4. Tanyakan kepada asisten laboratorium apabila ada hal-hal yang kurang jelas.

#### **D. Dasar Teori**

##### **Operator**

Dalam bahasa C++ tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan kita. Misalnya saja, bila kita ingin menginput dan mencetak bilangan dari 1 sampai 100 bahkan 1000, tentunya kita akan merasa kesulitan. Namun dengan struktur perulangan proses, kita tidak perlu menuliskan perintah sampai 100 atau 1000 kali, cukup dengan beberapa perintah saja. Struktur perulangan dalam bahasa C mempunyai bentuk yang bermacam-macam. Sebuah / kelompok instruksi diulang untuk jumlah pengulangan tertentu. Baik yang terdefiniskan sebelumnya ataupun tidak. Struktur pengulangan terdiri atas dua bagian : (1) Kondisi pengulangan yaitu ekspresi boolean yang harus dipenuhi untuk melaksanakan pengulangan; (2) Isi atau badan pengulangan yaitu satu atau lebih pernyataan (aksi) yang akan diulang.

Perintah atau notasi dalam struktur pengulangan adalah : 1) Pernyataan while

2) Pernyataan do..while

3) Pernyataan for

4) Pernyataan continue dan break

5) Pernyataan go to

## 1. Struktur perulangan WHILE

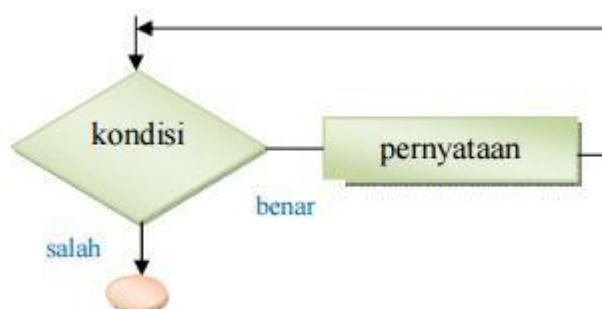
Perulangan WHILE banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar ( $\neq 0$ ) dan akan berhenti bila kondisinya bernilai salah ( $=0$ ).

Bentuk Umumnya :

```
while (kondisi)
{
    Pernyataan ;
}
```

Pengujian ungkapan pada while dilakukan sebelum bagian pernyataan, Oleh karena itu ada kemungkinan bagian pernyataan pada while tidak dijalankan sama sekali, yaitu kalau kondisi yang pertama kali bernilai salah.

Perhatikan gambar flowchart berikut :



Catatan :

*Jika menggunakan while pastikan bahwa pada suatu saat ungkapan pada while bernilai salah. Jika tidak demikian pernyataan yang mengikutinya akan dijalankan seterusnya.*

## 2. Struktur perulangan DO-WHILE

Pada dasarnya struktur perulangan do....while sama saja dengan struktur while, hanya saja pada proses perulangan dengan while, seleksi berada di while yang letaknya di atas sementara pada perulangan do....while, seleksi while berada di bawah batas perulangan. Jadi dengan menggunakan struktur do...while sekurang-kurangnya akan terjadi satu kali perulangan.

Pada struktur do-while kondisi pengecekan ditempatkan di bagian akhir. Hal ini menyebabkan struktur pengulangan ini minimal akan melakukan satu kali proses walaupun kondisi yang didefinisikan tidak terpenuhi (bernilai salah).

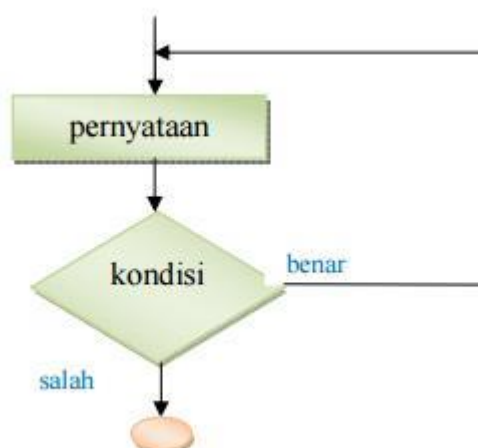
Bentuk umum dari struktur do-while

```
do
{
  pernyataan 1;
  pernyataan 2;
  ....
  pernyataan n ;
}while (ungkapan)
```

Catatan :

- Bagian pernyataan1 hingga N dijalankan secara berulang sampai dengan ungkapan bernilai salah (=0).
- Berbeda dengan while, pengujian ungkapan dilakukan di bagian belakang (setelah bagian pernyataan).
- Dengan demikian bagian pernyataan pada pengujian do – while minimal akan dijalankan satu kali.

Perhatikan bentuk flowchart do .. while berikut ini:



### 3. Struktur Perulangan FOR

Struktur perulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan for tampaknya lebih efisien karena susunannya lebih simpel dan sederhana.

Pernyataan for digunakan untuk melakukan looping. Pada umumnya looping yang dilakukan oleh for telah diketahui batas awal, syarat looping dan perubahannya. Selama kondisi terpenuhi, maka pernyataan akan terus dieksekusi. Bentuk umum perulangan for adalah sebagai berikut :

```
for ( ungkapan1; ungkapan2; ungkapan3)
    Pernyataan;
```

Keterangan :

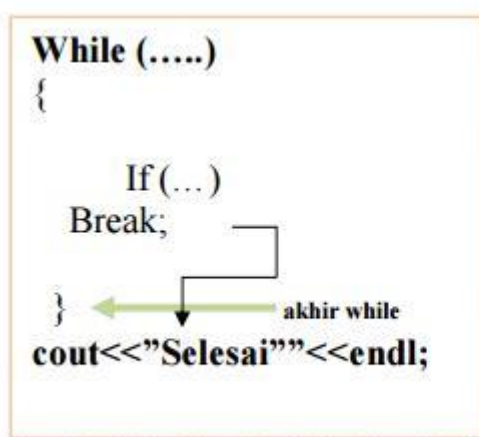
- *Ungkapan1 merupakan pernyataan inisialisasi*
- *Ungkapan 2 sebagai kondisi yang menentukan pengulangan*
- *Ungkapan 3 sebagai pengatur variabel yang digunakan di ungkapan1*

Pernyataan for dapat berada di dalam pernyataan for lainnya yang biasa disebut nested for.

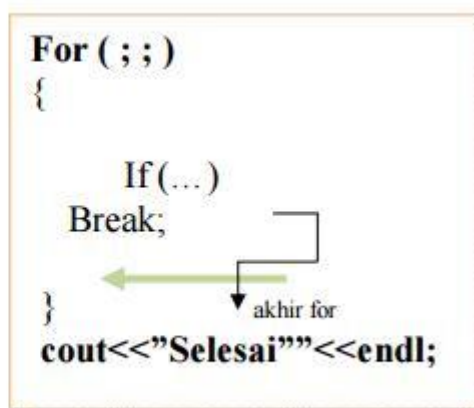
```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
{
    for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah)
        pernyataan / perintah;
}
```

#### 4. Pernyataan Break (dalam Perulangan)

Di samping break digunakan untuk pernyataan switch, juga bias digunakan pada For, while dan do-while. Intinya sama yakni untuk keluar dari pernyataan-pernyataan itu.



(a) While - break

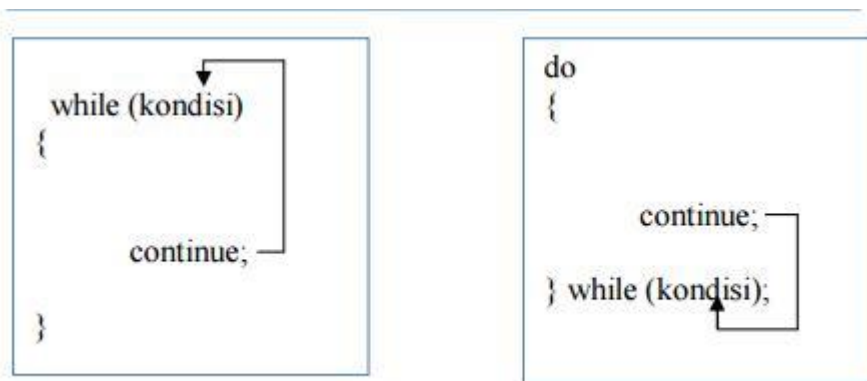


(b) For - Break

## 5. Pernyataan CONTINUE

Kegunaannya kebalikannya dari break. Dipakai untuk mengarahkan eksekusi ke putaran atau iterasi berikutnya pada pernyataan perulangan. Efek dari perintah continue:

- 1) Pada for: Ungkapan ke tiga pada for (ungkapan terkanan yg terletak di dalam () pada for) akan dijalankan dan kemudian ungkapan ke dua diuji lagi.
- 2) Pada while dan do- while : Pengujian terhadap ungkapan pada while dilakukan kembali.



## 6. Pernyataan Goto

Pernyataan goto, diperlukan untuk melakukan suatu lompatan ke suatu pernyataan berlabel yang ditandai dengan tanda " : ". Bentuk umumnya adalah :

```
goto bawah;
    pernyataan1;
    pernyataan2;
bawah : pernyataan 3;
```

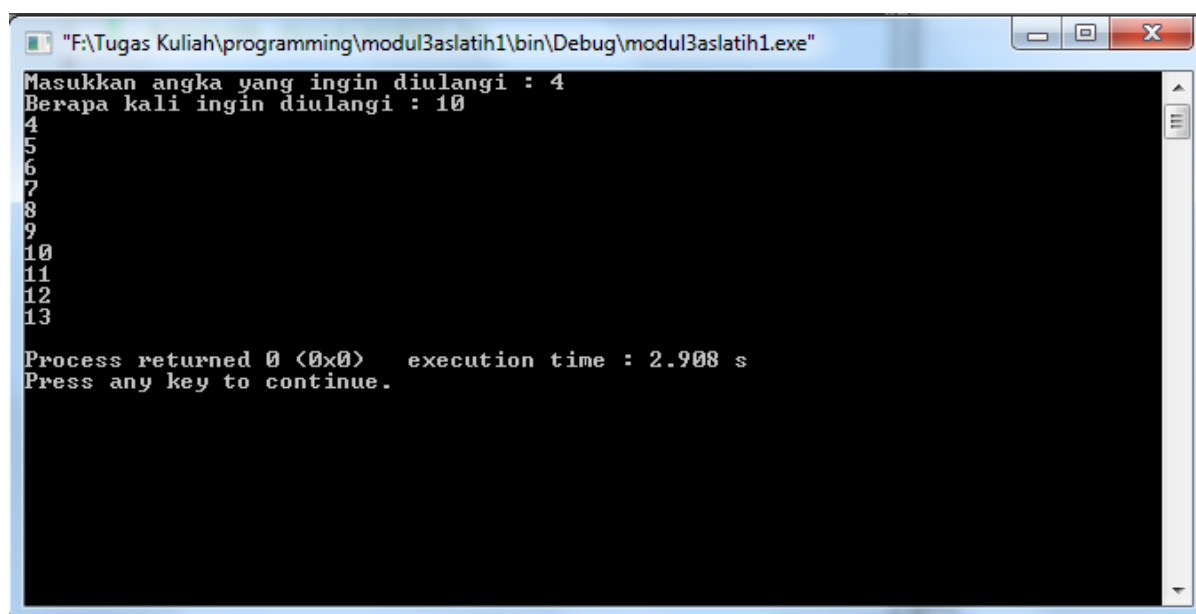
### E. Latihan Praktikum

1. Tuliskan kode program di bawah ini dengan menggunakan IDE masing masing, kemudian compile dan simpan dengan nama Modul3Latih1, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int angka, ulang, i = 0;
8      cout << "Masukkan angka yang ingin diulangi : ";
9      cin >> angka;
10     cout << "Berapa kali ingin diulangi : ";
11     cin >> ulang;
12
13     while (i < ulang){
14         cout << angka << endl;
15         i++;
16     }
17 }
18
```

Pertanyaan:

1. Coba hilangkan i++, Apa yang terjadi dan simpulkan fungsi i++!
2. Jelaskan maksud dari while(i < ulang), apa yang terjadi jika i < ulang di rubah menjadi ulang < i?
3. Rubah sintax sehingga output menjadi seperti ini:



2. Tuliskan kode program di bawah ini dengan menggunakan IDE masing-masing, kemudian compile dan simpan dengan nama Modul3Latih2, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
1 | #include <bits/stdc++.h>
2 |
3 | using namespace std;
4 |
5 | int main()
6 | {
7 |     int a, b = 1, c = 1;
8 |     cout << "Masukkan nilai a : ";
9 |     cin >> a;
10 |    cout << "Angka c bernilai : "<< c << endl;
11 |    cout << "Angka b bernilai : "<< b << endl;
12 |    cout << "Nilai a akan diulangi sebanyak perbandingan c dengan b : " << endl;
13 |    do {
14 |        cout << a << endl;
15 |        c++;
16 |    } while (c < b);
17 | }
18 |
```

Pertanyaan:

1. Simpulkan kenapa perulangan yang di tampilkan hanya 1?
  2. Ubah syntax dari do-while menjadi while dengan kondisi yang sama, apa bedanya?
3. Tuliskan kode program di bawah ini dengan menggunakan minGW, kemudian compile dan simpan dengan nama LATIH3, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
1 | #include <bits/stdc++.h>
2 |
3 | using namespace std;
4 |
5 | int main()
6 | {
7 |     int batas, perulangan;
8 |     string kata;
9 |
10 |    cout << "Masukkan kata : ";
11 |    cin >> kata;
12 |    cout << "Masukkan angka untuk mengulangi kata : ";
13 |    cin >> batas;
14 |
15 |    for ( perulangan = 0 ; perulangan < batas ; perulangan++){
16 |        cout << kata << endl;
17 |    }
18 | }
```

Pertanyaan:

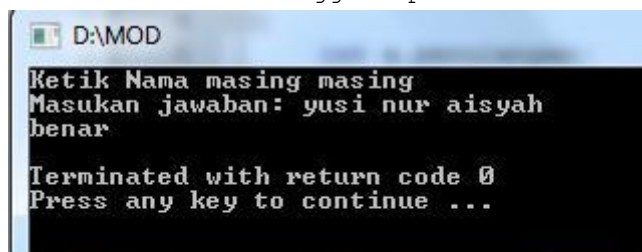
1. Apa yang terjadi jika angka yang dimasukan sama dengan 0? Apabila tidak muncul, bagaimana supaya tetap tampil jika di masukan angka 0 ? (Jelaskan dengan kedua cara while dan for)
2. Apa perbedaan perulangan menggunakan for dan while? Terletak di variable mana yang membedakan?

4. Tuliskan kode program di bawah ini dengan menggunakan IDE masing-masing, kemudian compile dan simpan dengan nama Modul3Latih4, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

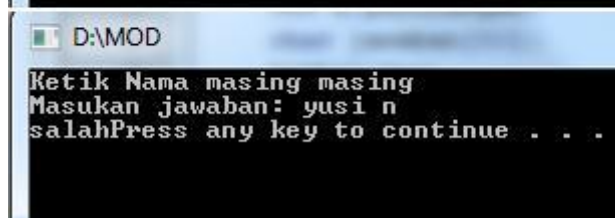
```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int a, perulangan, jawaban;
8      kembalikan :
9          system("pause");
10         system("cls");
11         cout << "Pertanyaan : 1 + 1 = " << endl;
12         for (perulangan = 0; perulangan < 10; perulangan++){
13             cout << "Masukkan jawaban : ";
14             cin >> jawaban;
15             if (jawaban == 2){
16                 cout << "benar";
17                 break;
18             }
19             else {
20                 cout << "salah" << endl;
21                 goto kembalikan;
22             }
23         }
24     }
25 }
```

Pertanyaan:

1. Apa yang terjadi jika break di hilangkan? Apa fungsi break?
2. Apa maksud goto kembalikan? Coba label kembalikan di pindah di bawah system("cls")? Apa yang terjadi?
3. Apa fungsi dari system("cls") dan system("pause"), apa yang terjadi jika salah satu atau keduanya di hilangkan?
4. Coba break di ganti dengan continue?
5. Rubah syntax hingga seperti ini?



```
D:\MOD
Ketik Nama masing masing
Masukan jawaban: yusi nur aisyah
benar
Terminated with return code 0
Press any key to continue ...
```



```
D:\MOD
Ketik Nama masing masing
Masukan jawaban: yusi n
salahPress any key to continue . . .
```



Clue:

- Cin di ganti dengan gets
- Kondisi di if di rubah dengan if(strcmp(variable jawaban,"yang di bandingkan")==0)
- Ganti yang di bandingkan menjadi nama kalian masing masing
- Simpulkan strcmp

5. Tuliskan kode program di bawah ini dengan menggunakan IDE masing-masing, kemudian compile dan simpan dengan nama Modul3Latih5, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int tinggi, i, j, k;
8      cout << "Masukkan tinggi segitiga : ";
9      cin >> tinggi;
10
11     for (i = tinggi; i >= 0; i--){
12         for (j = i; j >= 0; j--){
13             cout << " ";
14         }
15
16         for (k = 1; k <= tinggi-i; k++){
17             cout << "*";
18         }
19
20         for (k = 1; k < tinggi-i; k++){
21             cout << "*";
22         }
23         cout << endl;
24     }
25 }
```

Pertanyaan:

1. Bagaimana proses menjadi segitiga dari perulangan tsb?
2. Jelaskan fungsi perulangan yang berada di dalam perulangan?

## F. Tugas Praktikum

1. Pak Dengklek memiliki  $n$  kandang bebek. Kandang ke- $i$  berisi  $x$  ekor bebek. Berapa jumlah total bebek yang Pak Dengklek miliki?

### Contoh Input

Jumlah kandang : 5

Jumlah bebek di kandang ke-1 : 10

Jumlah bebek di kandang ke-2 : 44

Jumlah bebek di kandang ke-3 : 22

Jumlah bebek di kandang ke-4 : 55

Jumlah bebek di kandang ke-5 : 99

### Contoh Output

Jumlah bebek pak Dengklek adalah : 230

```
Masukkan jumlah kandang : 5
Masukkan jumlah bebek yang ada di kandang ke-1 : 10
Masukkan jumlah bebek yang ada di kandang ke-2 : 44
Masukkan jumlah bebek yang ada di kandang ke-3 : 22
Masukkan jumlah bebek yang ada di kandang ke-4 : 55
Masukkan jumlah bebek yang ada di kandang ke-5 : 99
Jumlah bebek pak Dengklek adalah : 230
Process returned 0 (0x0)   execution time : 10.869 s
Press any key to continue.
```

2. Buat sebuah bangun dengan bintang menggunakan sebuah perulangan dengan pilihan sebagai berikut :

1. Persegi
2. Segitiga siku
3. Persegi panjang
4. Segitiga terbalik

Contoh :

```
-----
Pilihan bangun datar :
-----
1. Persegi
2. Segitiga siku
3. Persegi panjang
4. Segitiga terbalik
-----
Pilihan anda adalah : 1
-----
Masukkan panjang sisi persegi : 4
-----
*****
*****
*****
*****
-----
Process returned 0 (0x0)   execution time : 3.464 s
Press any key to continue.
```

```
-----  
Pilihan bangun datar :  
-----  
1. Persegi  
2. Segitiga siku  
3. Persegi panjang  
4. Segitiga terbalik  
-----  
Pilihan anda adalah : 2  
-----  
Masukkan tinggi segitiga : 4  
****  
***  
**  
*  
  
Process returned 0 (0x0)   execution time : 2.861 s  
Press any key to continue.
```

```
-----  
Pilihan bangun datar :  
-----  
1. Persegi  
2. Segitiga siku  
3. Persegi panjang  
4. Segitiga terbalik  
-----  
Pilihan anda adalah : 3  
-----  
Masukkan panjang persegi panjang : 4  
Masukkan lebar persegi panjang : 3  
****  
****  
****  
  
Process returned 0 (0x0)   execution time : 3.881 s  
Press any key to continue.
```

```
-----  
Pilihan bangun datar :  
-----  
1. Persegi  
2. Segitiga siku  
3. Persegi panjang  
4. Segitiga terbalik  
-----  
Pilihan anda adalah : 4  
-----  
Masukkan tinggi segitiga : 4  
*****  
*****  
***  
*  
  
Process returned 0 (0x0)   execution time : 2.312 s  
Press any key to continue.
```

3. Buat program yang bisa menemukan nilai rata rata dari semua bilangan prima yang berada di antara bilangan inputan user.

Contoh :

Input awal : 1

Input akhir : 10

Bilangan prima yang ada di antara 1 dan 10 adalah :2, 3, 5, 7

Sehingga rata ratanya adalah :  $17 / 4 = 4.25$

```
Program penghitung rata rata bilangan prima
-----
Masukkan batas awal : 1
Masukkan batas akhir : 10
-----
Rata rata bilangan prima antara bilangan 1 dan 10 adalah 4.25
-----
Process returned 0 (0x0)   execution time : 2.588 s
Press any key to continue.
```