PENGANTAR KOMPUTER & SOFTWARE II

PERULANGAN 2

(WHILE & DO-WHILE)

Tim Pengajar KU1202 - Institut Teknologi Sumatera

Tujuan Kuliah

1. Mahasiswa memahami pengulangan (while dan do-while) dan penggunaannya serta memahami elemen-elemen dalam pengulangan.
2. Mahasiswa dapat menggunakan notasi pengulangan (while dan do-while) yang sesuai dengan benar
3. Mahasiswa dapat memanfaatkan jenis-jenis pengulangan dengan tepat dalam menyelesaikan persoalan sederhana yang diberikan.

Pengulangan : Latar Belakang

- **Melakukan** suatu **instruksi**, bahkan **aksi**, secara **berulang-ulang**
  - Komputer: memiliki performansi yang sama
  - Manusia: punya kecenderungan untuk melakukan kesalahan (karena letih atau bosan)

Pengulangan / Looping

- Elemen:
  - **Kondisi pengulangan**: ekspresi lojik
  - **Badan pengulangan**: aksi yang diulang
- Jenis-jenis notasi pengulangan:
  1. Berdasarkan pencacah : **for (pertemuan sebelumnya)**
  2. Berdasarkan kondisi pengulangan di awal : **while**
  3. Berdasarkan kondisi pengulangan di akhir : **do-while**

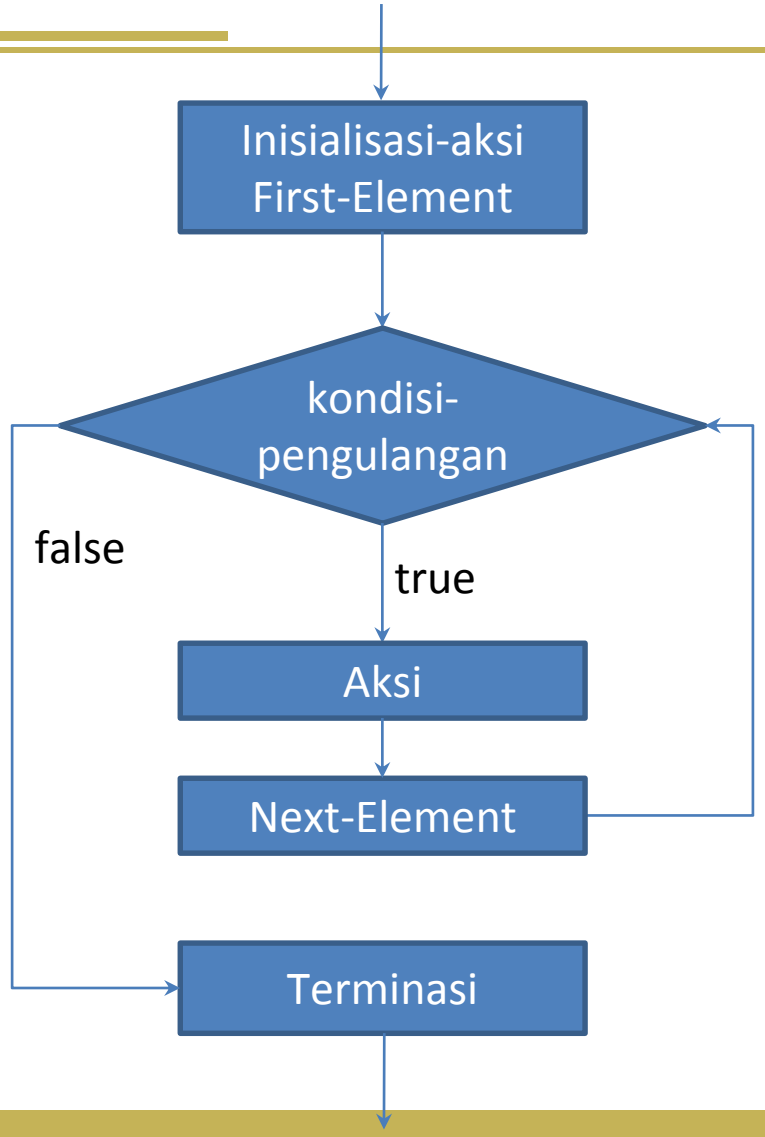
PENGULANGAN BERDASARKAN KONDISI PENGULANGAN DI AWAL (WHILE)

while

```
Inisialisasi-aksi
First-Element

while (kondisi-pengulangan)
{
    Aksi
    Next-Element
}
//Kondisi-pengulangan=false

Terminasi
```



while

- Pengulangan dikendalikan oleh elemen pengulangan yang diinisialisasi sebagai *First-Element* dan diubah nilainya dalam badan pengulangan menjadi *Next-Elem*
- *Aksi* akan dilakukan selama *kondisi-pengulangan* masih dipenuhi (berharga true)
- Tes terhadap *kondisi-pengulangan* dilakukan setiap kali sebelum *aksi* dilaksanakan
- Pengulangan ini berpotensi untuk menimbulkan *Aksi* "kosong" (tidak pernah melakukan apa-apa) karena pada test yang pertama, *kondisi-pengulangan* tidak dipenuhi (berharga false) sehingga langsung ke luar loop

Latihan 1. Berapa Hello di Layar?

- Buatlah program yang dapat menampilkan 10 Hello di layar.
- Gunakan notasi while untuk melakukan pengulangan.

```
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
```

Solusi - Latihan 1

```
#include <iostream>
using namespace std;
int main () {
// KAMUS
      int i;

// ALGORITMA
    i = 1; //first element
    while (i<=10) {//kondisi perulangan
       cout << “Hello” << endl;//aksi
       i++; //next element
    }
    return 0;
}
```

Perulangan akan terus dijalankan jika kondisi perulangan masih bernilai true (masih memenuhi batas yang ditentukan)

- Ubah nilai i = 1 menjadi i = 15.
- Apakah ada Hello yang muncul di layar?
- Why??

  Pengulangan while berpotensi untuk menimbulkan *Aksi* "kosong" (tidak pernah melakukan apa-apa) karena pada test yang pertama, *kondisi-pengulangan* tidak dipenuhi (berharga false) sehingga langsung ke luar loop

Latihan 2. Menghitung Angka

- Buatlah program yang dapat menghitung jumlah angka, dari 1 hingga N. Gunakan notasi while untuk melakukan pengulangan.
- Misal diinputkan N = 3.
- Output/tampilan yang muncul di layar.

```
1
2
3
Jumlah = 6
```

Solusi – Latihan 2

```
// Program JumlahAngka
// Menghitung 1+2+3+...+N; N > 0
#include <iostream>
using namespace std;
int main () {
// KAMUS
        int N, i, sum;
// ALGORITMA
        cin >> N;
        sum = 0; //Inisialisasi
        i = 1;   //First-Element
        while (i <= N) { //Kondisi-pengulangan
              cout << i << endl; //Aksi
              sum = sum + i;     //Aksi
              i = i + 1;         //Next-Element
        } // i > N
        cout << "Jumlah = " << sum << endl;  //Terminasi
        return 0;
}
```

```
// Alternatif ekspresi
while (i <= N) {
    cout << i << endl;
    sum+=i;
    i++;
} // i > N
```

PENGULANGAN BERDASARKAN KONDISI PENGULANGAN DI AKHIR (DO-WHILE)

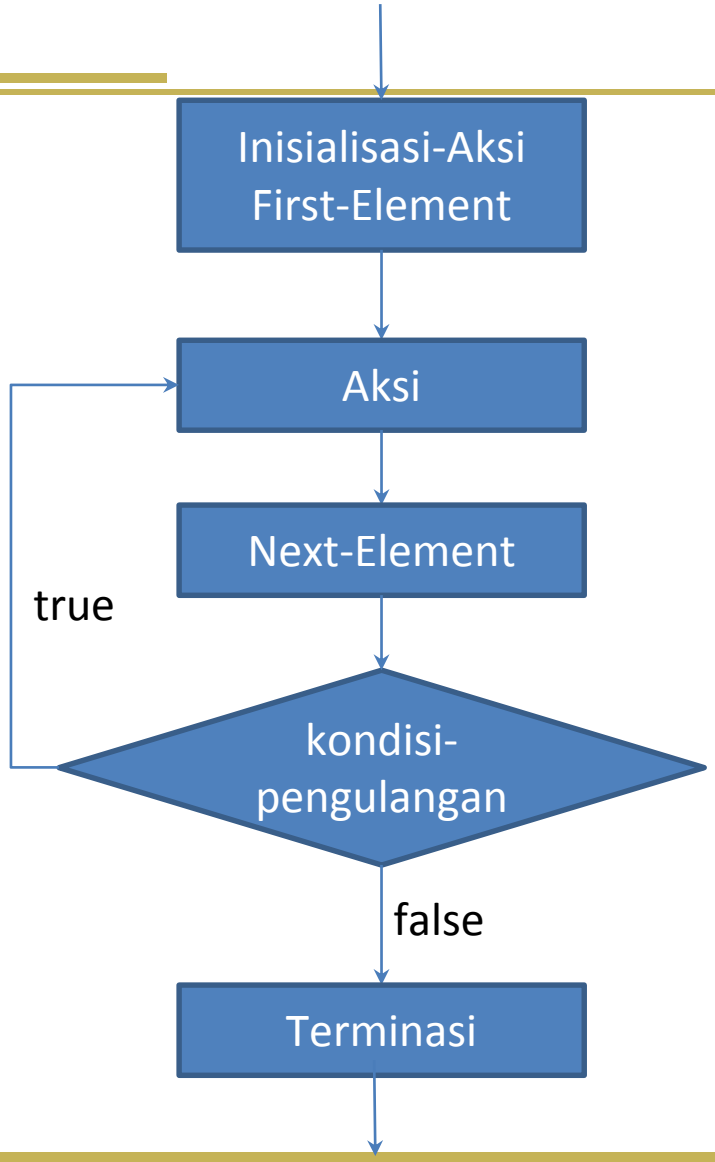
do-while

```
Inisialisasi-Aksi
First-Element

do
{
        Aksi
        Next-Element

} while (kondisi-pengulangan);

Terminasi
```



do-while

- Pengulangan dikendalikan oleh elemen pengulangan yang diinisialisasi sebagai *First-Element* dan diubah nilainya dalam badan pengulangan menjadi *Next-Element*
- *Aksi* minimal akan dilakukan **satu kali** karena pada waktu eksekusi pengulangan yang pertama tidak dilakukan test terhadap *kondisi-pengulangan*
- *Aksi* akan dihentikan jika *kondisi-pengulangan* tidak dipenuhi (berharga false), akan diulang jika *kondisi-pengulangan* tercapai

do-while

- Test terhadap kondisi pengulangan dilakukan setelah *Aksi* dilaksanakan
- Pengulangan berpotensi mengalami "kebocoran", jika ada kemungkinan bahwa seharusnya *Aksi* tidak pernah boleh dilakukan untuk kasus tertentu

Latihan 1. Berapa Hello di Layar?

- Sama dengan latihan sebelumnya.
- Buatlah program yang dapat menampilkan 10 Hello di layar.
- Akan tetapi, gunakan notasi do-while untuk melakukan pengulangan.

```
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
```

Solusi – Latihan 1

```
#include <iostream>
using namespace std;
int main () {
// KAMUS
      int i;

// ALGORITMA
   i = 1; //first element
   do {
      cout << "Hello" << endl;//aksi
      i++; //next element

   } while (i<=10);//kondisi perulangan

      return 0;
}
```

Perulangan akan terus dijalankan jika kondisi perulangan masih bernilai true (masih memenuhi batas yang ditentukan)

- Ubah nilai i = 1 menjadi i = 15.
- Apakah ada Hello yang muncul di layar?
- Apakah output yang dihasilkan berbeda dengan notasi while?
- Why??

Pengulangan do-while berpotensi mengalami “kebocoran”, jika ada kemungkinan bahwa seharusnya *Aksi* tidak pernah boleh dilakukan untuk kasus tertentu. Karena *Aksi* minimal akan dilakukan **satu kali** karena pada waktu eksekusi pengulangan yang pertama tidak dilakukan test terhadap *kondisi-pengulangan.*

Latihan 2. Menghitung Angka

- Sama dengan latihan sebelumnya.
- Buatlah program yang dapat menghitung jumlah angka, dari 1 hingga N.
- Akan tetapi, gunakan notasi do-while untuk melakukan pengulangan.
- Misal diinputkan N = 3.
- Output/tampilan yang muncul di layar.

```
1
2
3
Jumlah = 6
```

Solusi – Latihan 2

```
// Program JumlahAngka
// Menghitung 1+2+3+...+N; N > 0
#include <iostream>
using namespace std;
int main () {
// KAMUS
        int N, i, sum;
// ALGORITMA
        cin >> N;
        sum = 0; //Inisialisasi-aksi
        i = 1;   //First-element
        do {
              cout << i << endl; //Aksi
              sum = sum + i;     //Aksi
              i = i + 1;         //Next-Element
        } while (i <= N); //Kondisi Pengulangan
        cout << "Jumlah = " << sum << endl;   //Terminasi
        return 0;
}
```

```
// Alternatif ekspresi
do {
        cout << i << endl;
        sum+=i;
        i++;
} while ( i <= N );
```

Latihan 3. Menghitung Rata-Rata

- Buatlah program yang dapat menerima input bilangan integer sebanyak 5 kali dari pengguna. Kemudian program dapat menampilkan nilai rata-ratanya.
- Gunakan notasi do-while.
- Contoh tampilan program

Bilangan yang berwarna hijau diinputkan oleh pengguna

```
Input Nilai ke-1 = 2
Input Nilai ke-2 = 5
Input Nilai ke-3 = 3
Input Nilai ke-4 = 6
Input Nilai ke-5 = 9
Nilai rata-rata = 5
```

Tambahan

- Ubah solusi 3 tersebut, dengan menggunakan notasi while.

Latihan 4

Buatlah program yang meminta user untuk menginputkan tiga bilangan bulat a, b dan c (satu per satu). Apabila di antara a, b, dan c ada yang bernilai sama maka program akan kembali meminta user menginputkan a, b dan c seterusnya hingga ketiganya berbeda nilai.

Contoh tampilan :

```
Masukkan nilai a = 6
Masukkan nilai b = 4
Masukkan nilai c = 4
Masukkan nilai a = 1
Masukkan nilai b = 1
Masukkan nilai c = 2
Masukkan nilai a = 8
Masukkan nilai b = 8
Masukkan nilai c = 8
Masukkan nilai a = 3
Masukkan nilai b = 4
Masukkan nilai c = 5
```

TERIMA KASIH