

MODUL 5

OPERASI PERULANGAN

A. Tujuan Praktikum

Setelah mempelajari bab ini mahasiswa diharapkan mampu :

1. Menjelaskan fungsi perulangan for
2. Menjelaskan fungsi perulangan while
3. Menjelaskan fungsi perulangan do while
4. Menjelaskan fungsi break, continue, dan goto

B. Waktu Praktikum

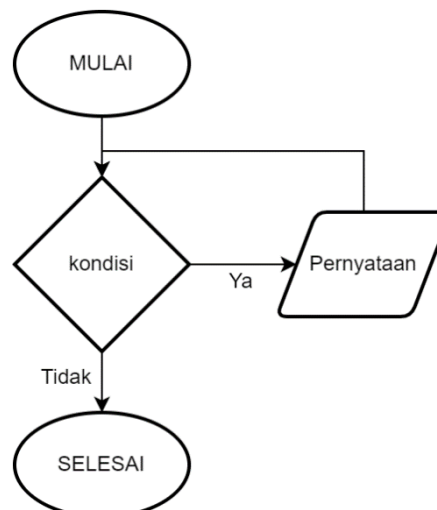
4 x 1 jam

C. Petunjuk

1. Awali setiap kegiatan dengan berdoa dan tekad kuat, agar diberi kelancaran dalam belajar dan mendapatkan ilmu yang bermanfaat.
2. Pahami tujuan praktikum, dasar teori, dan 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

Pada pemrograman komputer, perulangan digunakan untuk mengulang blok kode yang dieksekusi. Misalnya diperlukan untuk menampilkan pesan sebanyak 70 kali. Daripada menulis perintah cout sebanyak 70 kali, kita dapat menggunakan perintah perulangan untuk mengulangi satu perintah cout sebanyak 70 kali.

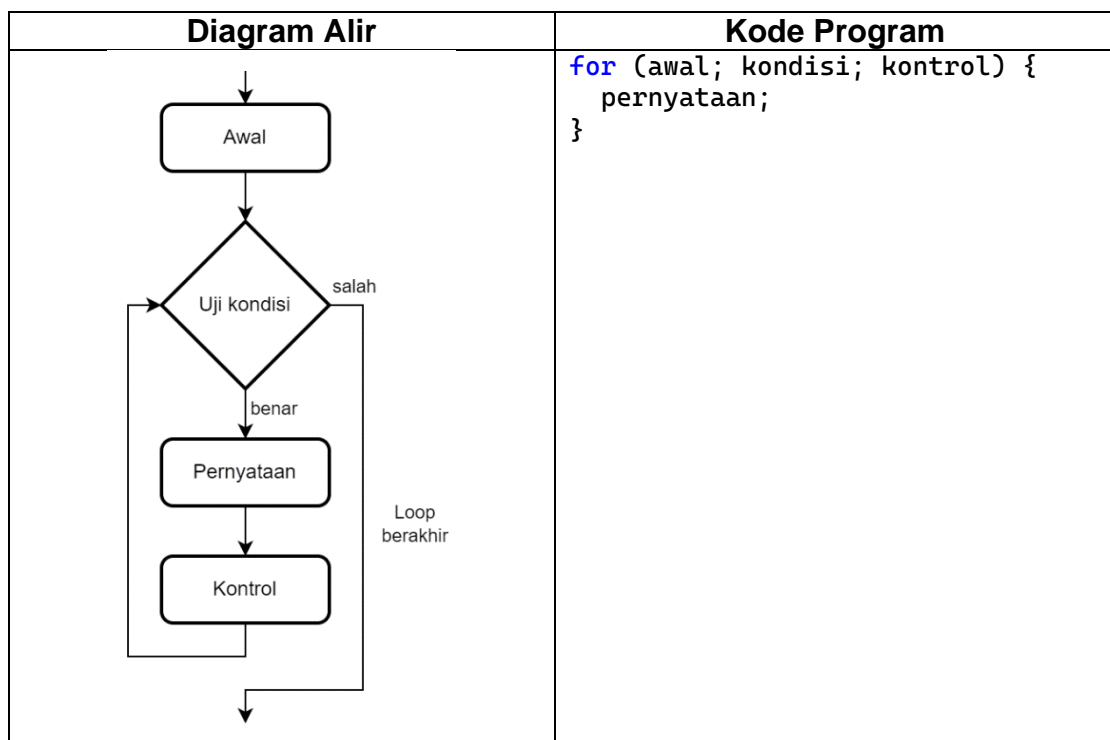


Implementasi perulangan tidak hanya terbatas pada cetak berulang namun fungsi lain seperti menghitung bilangan pangkat. Terdapat 3 jenis perulangan di dalam C++, yaitu :

1. Perulangan for

Perulangan loop adalah jenis perulangan yang memungkinkan untuk melakukan perulangan dalam jumlah yang spesifik. Sebuah variabel perlu digunakan untuk mengontrol perulangan yang akan berhenti apabila memenuhi suatu kondisi.

- Blok awal digunakan untuk melakukan inisialisasi variabel loop.
- Blok kondisi digunakan untuk melakukan pengecekan. Jika bernilai benar maka perulangan dieksekusi, jika tidak maka perulangan dihentikan.
- Blok kontrol digunakan untuk melakukan perubahan nilai variabel loop.



Contoh Program :

```

#include <iostream>

using namespace std;

int main() {
    for (int i = 1; i <= 5; i++) {
        cout << "Hitung " << i << endl;
    }
}
        
```

Keluaran Program :

```
Microsoft Visual Studio Debug (x) + -
Hitung 1
Hitung 2
Hitung 3
Hitung 4
Hitung 5
```

2. Perulangan while

Perulangan while adalah perulangan yang tidak membutuhkan jumlah pasti seberapa banyak perulangan dilakukan. Sehingga perulangan while bekerja dengan mengecek kondisi, apabila bernilai benar maka pernyataan di dalam perulangan akan dieksekusi hingga kondisi bernilai salah.

Diagram Alir	Kode Program
<pre>graph TD Entry(()) --> UjiKondisi{Uji kondisi} UjiKondisi -- benar --> Pernyataan[Pernyataan] Pernyataan --> PerubahanKondisi[Perubahan kondisi] PerubahanKondisi --> UjiKondisi UjiKondisi -- salah --> Exit(()) subgraph Loop Pernyataan PerubahanKondisi end Exit --- LoopEnd[Loop berakhir]</pre>	<pre>while (kondisi) { pernyataan; perubahan_kondisi; }</pre>

Contoh Program :

```
#include <iostream>

using namespace std;

int main() {
    int i = 1;

    while (i <= 5) {
        cout << "Hitung " << i << endl;
        i++;
    }
}
```

Keluaran Program :

```
Microsoft Visual Studio Debug ( X + - )
Hitung 1
Hitung 2
Hitung 3
Hitung 4
Hitung 5
```

3. Perulangan do while

Perulangan do while merupakan variasi dari perulangan while dimana letak perbedaannya adalah perulangan ini melakukan eksekusi kode sekali sebelum pengecekan kondisi.

Diagram Alir	Kode Program
<pre>graph TD Entry(()) --> Pernyataan[pernyataan] Pernyataan --> Perubahan[Perubahan kondisi] Perubahan --> Uji{Uji kondisi} Uji -- benar --> Pernyataan Uji -- salah --> Exit[Loop berakhir] Exit --> Exit2(())</pre>	<pre>do { pernyataan; perubahan_kondisi; } while (kondisi);</pre>

Contoh Program :

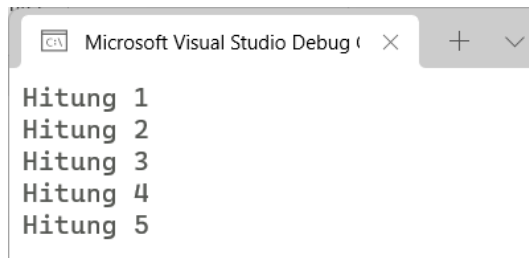
```
#include <iostream>

using namespace std;

int main() {
    int i = 1;

    do {
        cout << "Hitung " << i << endl;
        i++;
    } while (i <= 5);
}
```

Keluaran Program :



```
Microsoft Visual Studio Debug Console
Hitung 1
Hitung 2
Hitung 3
Hitung 4
Hitung 5
```

Selain itu terdapat perintah lain yang di dalam pengulangan yaitu break, continue, dan goto.

1. Pernyataan break

Pernyataan break digunakan untuk menghentikan perulangan. Contohnya program dibawah hanya akan mencetak angka 1 dan 2. Karena ketika bertemu dengan angka 3, perulangan dihentikan.

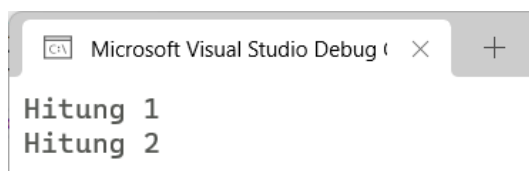
Contoh Program :

```
#include <iostream>

using namespace std;

int main() {
    for (int i = 1; i <= 5; i++) {
        if (i == 3) {
            break;
        }
        cout << "Hitung " << i << endl;
    }
}
```

Keluaran Program :



```
Microsoft Visual Studio Debug Console
Hitung 1
Hitung 2
```

2. Pernyataan continue

Pernyataan continue digunakan untuk melewati atau skip iterasi saat ini dan berpindah ke iterasi berikutnya. Contohnya pada program ini, iterasi dimana i bernilai 3 akan diskip sehingga hanya mencetak angka 1, 2, 4, dan 5.

Contoh Program :

```
#include <iostream>

using namespace std;

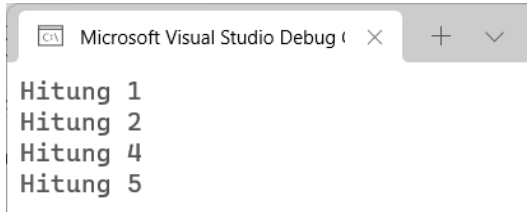
int main() {
    for (int i = 1; i <= 5; i++) {
        if (i == 3) {
            continue;
        }
        cout << "Hitung " << i << endl;
    }
}
```

```

        continue;
    }
    cout << "Hitung " << i << endl;
}
}

```

Keluaran Program :



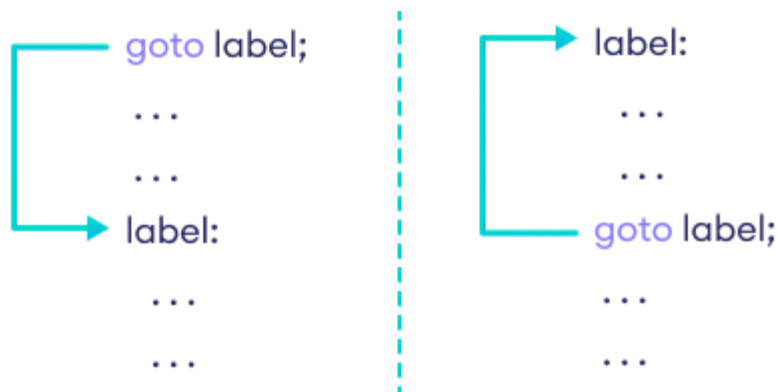
```

Microsoft Visual Studio Debug Console
Hitung 1
Hitung 2
Hitung 4
Hitung 5

```

3. Pernyataan goto

Pernyataan goto digunakan untuk melewati urutan normal eksekusi program dengan melakukan “lompatan” ke bagian lain. Pernyataan goto dapat melompat ke atas maupun ke bawah blok kode.



Pernyataan goto biasanya jarang dipakai dalam pemrograman karena membuat kode menjadi kompleks dan membingungkan. Sehingga beberapa programmer memilih menggunakan break dan continue.

Contoh Program :

```

#include <iostream>

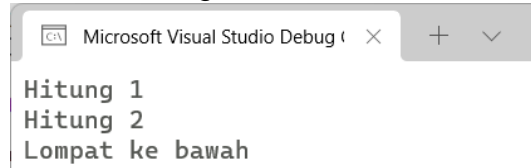
using namespace std;

int main() {
    for (int i = 1; i <= 5; i++) {
        if (i == 3) {
            goto lompat;
        }
        cout << "Hitung " << i << endl;
    }

    cout << "Pernyataan ini tidak akan dieksekusi";
    lompat:
    cout << "Lompat ke bawah";
}

```

Keluaran Program :



E. Latihan Praktikum

1. Ketik kode program di bawah ini dengan menggunakan Visual Studio, kemudian compile dan jalankan program tersebut. Jelaskan dan simpulkan hasil program.

```
#include <iostream>

using namespace std;

int main() {
    int number;

    do {
        cout << "Masukkan sebuah angka positif: ";
        cin >> number;
    } while (number <= 0);

    cout << "Angka berhasil dimasukkan adalah " << number << endl;
}
```

2. Ketik kode program di bawah ini dengan menggunakan Visual Studio, kemudian compile dan jalankan program tersebut. Jelaskan dan simpulkan hasil program.

```
#include <iostream>

using namespace std;

int main() {
    int base, exp, result = 1;

    cout << "Masukkan angka: ";
    cin >> base;

    cout << "Masukkan pangkat: ";
    cin >> exp;

    cout << base << "^" << exp << " adalah ";

    while (exp != 0) {
        result = result * base;
        exp--;
    }

    cout << result << endl;
}
```

3. Ketik kode program di bawah ini dengan menggunakan Visual Studio, kemudian compile dan jalankan program tersebut. Jelaskan dan simpulkan hasil program.

```
#include <iostream>

using namespace std;

int main() {
    for (int i = 1; i < 100; i++) {
        if (i % 2 == 0) {
            cout << i << " ";
        }
    }
}
```

F. Tugas Praktikum

1. Buatlah sebuah program sederhana yang menerima input kalimat yang akan diulang dan jumlah pengulangan (n). Program akan menampilkan kalimat tersebut dalam n kali.
2. Buatlah program yang menerima input integer a dan b. Kemudian program akan menampilkan bilangan prima dari rentang a hingga b.
3. Buatlah program yang menerima input integer n kemudian program akan menampilkan bilangan fibonacci ke-n.
4. Buatlah sebuah program yang menyelesaikan permasalahan di sekitar dengan perulangan. Kemudian jelaskan hasil dari program tersebut.