

## Praktikum 3

### PERULANGAN

#### A. TUJUAN

1. Memahami kapan keadaan pengulangan digunakan.
2. Mengetahui beberapa perintah pengulangan dalam C
3. Mengetahui masing-masing format perintah pengulangan dalam C.
4. Dapat menerapkan perintah pengulangan dalam algoritma penyelesaian suatu masalah
5. Dapat membuat program dengan perintah pengulangan.

#### B. DASAR TEORI

##### Struktur Pengulangan `while`

Pengulangan `while` digunakan untuk mengeksekusi beberapa instruksi jika kondisinya bernilai `true`. Sehingga jika kondisi awal bernilai `false` maka perintah-perintah dalam `while` tidak akan pernah dieksekusi.

Bentuk umum:

```
while (kondisi)
    instruksi
```

Ketika dieksekusi, instruksi `while` pertama memeriksa kondisi. Jika kondisi memberikan nilai `true`, sistem akan mengeksekusi instruksi. Kemudian mengulangi tes kondisi dan mengerjakan instruksi jika hasilnya masih `true`. Hal ini diulangi sampai kondisi mengembalikan nilai `false`.

Contoh penggunaan pengulangan dengan 'while' dalam program untuk menjumlahkan 10 angka yang dimasukkan:

```
int main()    {
    int number, sum = 0, count = 0;

    while (count < 10){
        cout<<"Masukkan angka";
        cin>>number;
        sum += number;
        count++;
    }
    cout<<"jumlah: "<<sum;
}
```

Dalam contoh program diatas, perulangan dilakukan 10 kali dengan instruksi `while (count < 10)`, `count < 10` adalah tes kondisi. Dalam hal ini tes adalah: "apakah count lebih kecil dari 10 ? ". Jika jawabannya "ya" maka instruksi di pengulangan 'while' dieksekusi. Jika "tidak" maka perulangan tidak dilakukan lagi dan menuju perintah selanjutnya yaitu `cout<<"jumlah: "<<sum;`

### **Struktur Pengulangan do while**

Pengulangan '`do while`' juga mengeksekusi beberapa instruksi berulang jika kondisi yang telah ditentukan dipenuhi. Dalam perulangan '`do while`' tes kondisi dilakukan pada akhir pengulangan. Bentuk umum pengulangan '`do while`' :

```
do {
    Instruksi instruksi
} while (kondisi
```

Berikut ini adalah contoh program untuk menghitung nilai count dimulai dengan 0, dengan penambahan 2 dan akan berhenti pada 15

```

int main()
{
    int Count = 0;
    do {
        cout<< Count * 2;
        Count++;
    } while( Count <= 15 );
}

```

Pada contoh program diatas, pertama kali akan dicetak `Count * 2`, kemudian `Count` ditambahkan satu sebagai perubahan tes kondisi untuk menghitung berapa kali pengulangan sudah dilakukan pengulangan. Tes kondisi dilakukan setelah instruksi dalam pengulangan dilakukan dengan instruksi sebagai berikut ( `Count <= 15` ), jika hasil tes kondisi bernilai *true* maka pengulangan dikerjakan jika bernilai *false* maka pengulangan berhenti.

### Struktur Pengulangan For

Struktur pengulangan ketiga adalah pengulangan 'for'. Pengulangan 'for' dapat mengeksekusi instruksi berulang sejumlah yang sudah ditentukan. Bentuk umumnya adalah sebagai berikut:

```

for (inisialisasi;tes kondisi;perubahan nilai)
{
    instruksi;
}

```

Inisialisasi merupakan pemberian nilai awal pada variabel yang akan dilakukan tes kondisi. Tes kondisi digunakan untuk menentukan apakah pengulangan bisa dikerjakan. Dan yang terakhir perubahan nilai yaitu perubahan nilai untuk tes kondisi sehingga pengulangan bisa mencapai batas untuk berhenti.

Berikut adalah contoh pengulangan 'for' untuk menghitung dari 1 sampai 10:

```

for (int count = 1; count <= 10; count++)
{
    printf("%d\n", count);
}

```

Tes kondisi mungkin tidak berhubungan dengan variabel yang diinisialisasi dan diperbaharui. Pada contoh ini pengulangan menghitung sampai pemakai memberikan respon untuk menghentikan pengulangan. Contoh seperti berikut:

```

for (count = 1; response != 'T'; count++)
{
    printf("%d\n", count);
    printf("Lanjut (Y/T): ");
    scanf("%c", &response);
}

```

### C. TUGAS PENDAHULUAN

1. Program dibawah ini seharusnya menjumlahkan sejumlah bilangan integer yang dimasukkan dan berhenti ketika bilangan yang dimasukkan adalah -1. Kemudian menampilkan banyaknya data yang dimasukkan kecuali -1 dan hasil penjumlahan semua data tersebut kecuali -1. Perbaikilah program tersebut sehingga memberikan hasil yang benar. Cobalah dengan memasukkan data 1 2 3 4 5 -1, jumlah data yang ditampilkan harusnya 5 dan hasil penjumlahan adalah 15. apakah hasilnya sudah benar?

```

int main()
{
    int total = 0, count = 0, number;
    do{
        cin >> number;
        total = total + number;
        count++;
    }while (number != -1);
    cout<<"The number of data read is "<< count << endl;
    cout<<"The sum of the numbers entered is " << total;
    return 0;
}

```

2. Rubahlah program berikut menjadi bentuk while loop

```

int main()

```

```
{
    int counter,sum,N;
    cout << "Enter the number of positive integers :";
    cin >> N;
    sum = 0;
    for (counter = 1; counter <= N; counter++)
        sum = sum + counter;

    cout << "The sum of the first " << N
    << " positive integers is " << sum << endl;
    return 0;
}
```