

LEARNING JOURNAL

Nama : Bilhaq Avi Dewantara

NIM : 120140141

Kelas : RC

Minggu Ke - : 5

Materi : Perulangan

A. Pokok pikiran (Teorinya) :

Di dalam suatu pemrograman, kita pasti tidak menulis kalimat yang sama di tiap barisnya karena itu sangat membuat kita lelah. Maka dari itu, munculah suatu fungsi pengulangan dalam pemrograman. Pengulangan dalam pemrograman merupakan suatu proses eksekusi *source* program yang dilakukan berulang-ulang apabila telah memenuhi suatu syarat tertentu. Apabila syarat tersebut tidak memenuhi maka program akan berhenti atau tidak melakukan eksekusi, sehingga akan melanjutkan ke program lainnya bila ada suatu kondisi yang lain. Di pemrograman menggunakan bahasa C++ ada 3 jenis notasi perulangan, yaitu :

1. FOR

Perulangan menggunakan *for* biasanya digunakan untuk kondisi data tipe integer atau angka. Perulangan ini juga merupakan perulangan yang berdasarkan pencacahan. Suatu pencacah harus memiliki variabel yang terdefinisi sebagai suksesor dan predesesor. Nilai dari suatu pencacah adalah “*hmim*” dan juga “*hmaks*”. Aksi yang dilakukan ialah dengan memperhitungkan harga-harga dari pencacah yang telah di-*explore*. Apabila suatu pencacah = *hmaks*, maka pengulangan akan berhenti.

2.. WHILE

Perulangan menggunakan syntax *while* biasanya hanya berfokus kepada suatu kondisi benar dan juga salah saja. Nilai itu dapat dihasilkan dari notasi operator aritmatika ataupun dari yang lain agar mengembalikan data dengan boolean *true* ataupun *false*. Pengambilan data dari fungsi yang lain ini bisa menjadi perbedaan antara notasi pengulangan *for* dan *while*.

3. DO-WHILE

Do-while merupakan pengulangan yang terjadi di awal. Pengendalian dari pengulangan ini di proses oleh elemen pengulangan yang eksekusi sebagai First-Element dan diubah ke dalam pengulangan yang menjadi Next-Element.

nt. Maksud dari sini ialah kode di dalam *do* akan dijalankan terlebih dahulu sebelum terjadi pengecekan kondisi *while*. Program akan dieksekusi bila kondisi *while* terpenuhi. Dengan demikian, apabila suatu kondisi *while* tidak terpenuhi, maka perulangan ini akan berlanjut mengeksekusi kode programnya satu kali.

B. Penerapan (Contoh) :

1. Notasi For

```
//
Nama      : Bilhaq Avi Dewantara
NIM       : 120140141
TPB       : 52
Prodi     : Teknik Informatika
*/

#include <iostream>
using namespace std;

int main()
//
{
    int x;
    int n=0;

    cout << "\t\t*****\n";
    cout << "\t\tPROGRAM PENGULANGAN BILANGAN NEGATIF\n";
    cout << "\t\t*****\n\n";

    // Command Input Angka Negatif
    cout << "Berhenti di bilangan ke~> ";
    cin >> x;

    cout << "\n\t_____STARTING_____ \n\n";

    // Pengulangan
    while (n != x)
    {
        n--;
        cout << "Angka ke~> " << n << endl;
    }
    cout << "\n\t_____END_____ \n";

    return false;
}
```

2. Notasi While

```
/*
Nama      : Bilhaq Avi Dewantara
NIM       : 120140141
Prodi     : Teknik Informatika
TPB       : 52
*/

#include <iostream>
using namespace std;

int main()
// ALGORITMA
{
    int n, t , i;

    cout << "Masukkan nilai Maks -> ";
    cin >> n;
    cout << "Masukkan nilai Min -> ";
    cin >> t;
    i = t;

    while (i <= n){ // Kondisi Perulangan
        cout << i << endl; // Aksi
        i++;
    }
    cout << "Minimal " << t << endl;
    cout << "Maksimal " << n;

    return 0;
}
```

3. Notasi Do-While

```
#include <iostream>
using namespace std;

int main()
{
    int sum, i, n;

    cout <<"Masukkan nilai -> ";
    cin >> n;
    sum = 0; // Inisialisasi
    i = 1; // First-element
    do {
        cout << i << endl; // Aksi
        sum = sum + i;      // Aksi
        i = i + 1;          // Next-Element

    } while (i <= n);
    cout << "Jumlah'";
    cout << sum << endl;

    return 0;
}
```