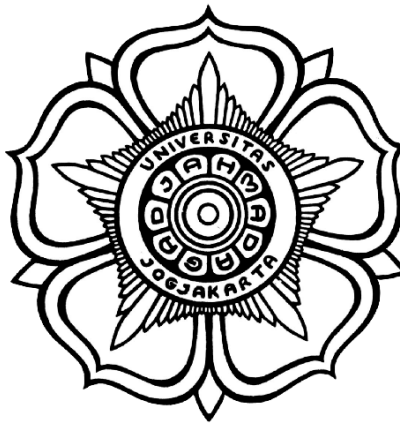


**LAPORAN PRAKTIKUM
PEMROGRAMAN DASAR**

**BAB VI
PERULANGAN**



Disusun Oleh:

FAIZAH EL JOASA ARIESTA (22/499164/SV/21287)

**PROGRAM STUDI DIPLOMA TEKNOLOGI REKAYASA
INSTRUMENTASI DAN KONTROL
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2022

I. Dasar Teori

C++ merupakan bahasa pemrograman yang merupakan salah satu Bahasa pemrograman populer yang digunakan oleh para developer saat ini, dengan bahan modul yang telah dipaparkan oleh dosen dan asisten dosen, saya dapat memahami struktur sintaksis yang dimiliki oleh Bahasa pemrograman C++, salah satunya adalah “Perulangan”, perulangan ini sendiri memiliki berbagai kategori mulai dari, “While”, “Do While”, dan “For”. Penggunaan dari masing-masing perintah tersebut memiliki perbedaan sesuai dengan kondisi dan permasalahan yang sedang kita hadapi. Untuk memudahkan kita dalam melakukan program yang bersifat looping maka cara tersebut sangat efisien untuk mempermudah kita dalam mencetak sesuatu dengan program perulangan.

II. Tujuan

1. Memahami fungsi dari masing-masing struktur perulangan.
2. Memahami penggunaan perulangan sesuai dengan kondisi yang dialami.
3. Pengimplementasian struktur perulangan atau looping pada suatu program dengan struktur yang sesuai.

III. Hasil Praktikum

1. Program Mencetak Bilangan Prima

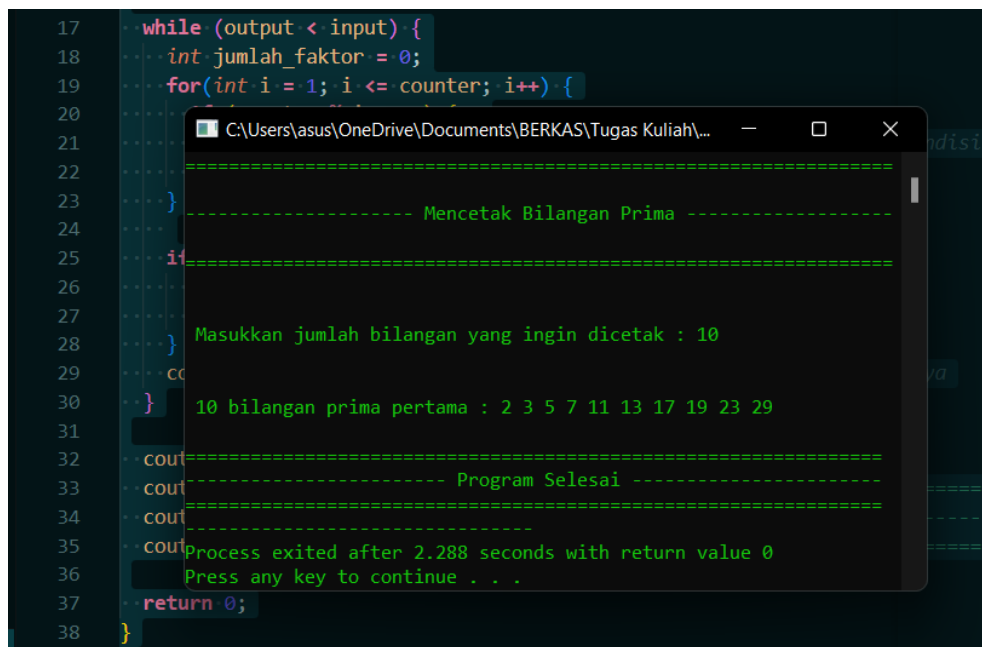
```
1  #include <iostream> // package library yang digunakan
2  using namespace std;
3
4  int main() {
5      int input; // variable untuk menyimpan jumlah bilangan yang ingin dicetak
6      cout << "===== " << endl;
7      cout << "----- Mencetak Bilangan Prima ----- " << endl;
8      cout << "===== " << endl << endl;
9      cout << "Masukkan jumlah bilangan yang ingin dicetak : "; cin >> input;
10     cout << endl << endl;
11
12     int counter = 2; // variable yang digunakan untuk menyimpan bilangan prima
13     int output = 0; // variable penentu untuk menjalankan program sesuai dengan input
14
15     cout << " " << input << " bilangan prima pertama : ";
16
17     while (output < input) {
18         int jumlah_faktor = 0;
19         for(int i = 1; i <= counter; i++) {
20             if (counter % i == 0) {
21                 jumlah_faktor++; // menaikkan jumlah faktor ketika memenuhi kondisi
22             }
23         }
24
25         if (jumlah_faktor == 2) {
26             cout << counter << " "; // menampilkan bilangan prima
27             output++; // setelah menampilkan counter, output dinaikan
28         }
29         counter++; // menaikkan counter, untuk mengecek bilangan selanjutnya
30     }
31
32     cout << endl << endl;
33     cout << "===== " << endl;
34     cout << "----- Program Selesai ----- " << endl;
35     cout << "===== ";
36
37     return 0;
38 }
39
```

Analisi Program :

Program memanfaatkan lima buah variable dimana, yang akan berinteraksi dengan user adalah variable input, yang berfungsi untuk menentukan berapa jumlah bilangan prima yang akan ditampilkan (dimulai dari 2). Pada saat sistem telah menerima hasil input dari user maka sistem akan menjalankan program dengan melalui struktur perulangan “While” dimana kondisi ini memungkinkan system untuk mulai menjalankan program Ketika output tidak lebih besar dari input, kondisi ini digunakan untuk mencetak berapa banyak bilangan prima yang akan ditampilkan kepada user.

Kemudian setelah program berhasil lolos kondisi tersebut, maka program berlanjut dengan menetapkan nilai 0 dari variable “jumlah_faktor”, dimana variable ini akan menyeleksi bilangan-bilangan yang merupakan bilangan prima, dikarenakan jumlah faktor dari bilangan prima adalah 2, maka terdapat kondisi dimana variable “jumlah_faktor == 2” yang artinya jika jumlah faktor bernilai sama dengan 2, maka program akan menampilkan bilangan prima yang telah disimpan nilainya ke dalam variable “counter”, setelah menampilkan maka variable output dinaikan agar dapat menyeleksi urutan selanjutnya dan mencapai batas input awal, dan counter dinaikan untuk mengecek bilangan selanjutnya yang akan diseleksi.

Hasil output :



```
17 while (output < input) {
18     int jumlah_faktor = 0;
19     for(int i = 1; i <= counter; i++) {
20         ...
21         ...
22         ...
23     }
24     ----- Mencetak Bilangan Prima -----
25     ...
26     ...
27     ...
28     }
29     cout << "Masukkan jumlah bilangan yang ingin dicetak : 10\n";
30     ...
31     }
32     cout << "10 bilangan prima pertama : 2 3 5 7 11 13 17 19 23 29\n";
33     cout << "----- Program Selesai -----";
34     cout << "\n";
35     cout << "Process exited after 2.288 seconds with return value 0\n";
36     cout << "Press any key to continue . . .";
37     return 0;
38 }
```

2. Program Menampilkan Deret Bilangan Ganjil dan Genap

```
1 #include <iostream> // package library yang digunakan
2 using namespace std;
3
4 int main() {
5     int pilihan, jumlah, range, i, bilangan;
6     string jenis;
7
8     cout << "===== " << endl;
9     cout << "----- Program Deret Bilangan ----- " << endl;
10    cout << "===== " << endl << endl;
11    cout << " 1. Bilangan Ganjil      2. Bilangan Genap" << endl << endl;
12    cout << "===== " << endl;
13    cout << " Masukkan pilihan deret bilangan      : "; cin >> pilihan; // input untuk memilih program
14
15
16    switch(pilihan) {
17        case 1: jenis = "ganjil"; // program untuk mencetak deret bilangan ganjil
18            cout << " Masukkan panjang deret bilangan ganjil : "; cin >> range; // memilih banyaknya bilangan
19            cout << "===== " << endl << endl;
20            cout << " Deret bilangan " << jenis << " yang didapatkan : ";
21
22            for (i = 1; i <= range; i++) {
23                cout << i * 2 - 1; // ganjil dimulai dari 1, maka 1 x 2 - 1 = 1, begitu seterusnya
24                if (i < range) {
25                    cout << ", "; // mencetak koma pada akhir bilangan, untuk i lebih kecil dari range
26                }
27            }
28
29            cout << endl;
30            cout << " Hasil penjumlahan dari " << range << " deret bilangan : " << endl << endl;
31            cout << " =>  ";
32
33            for (i = 1; i <= range; i++) {
34                cout << i * 2 - 1;
35                if (i < range) {
36                    cout << " + "; // mencetak tanda plus diakhir bilangan, untuk i lebih kecil dari range
37                }
38            }
39
40            jumlah = range; // pemasukan nilai jumlah yang di dapat dari nilai range
41            jumlah = jumlah * jumlah; // penjumlahan deret bilangan ganjil memiliki rumus range kuadrat
42            cout << " = " << jumlah << endl << endl;
43            break;
44
45        case 2: jenis = "genap";
46            bilangan = 0; // pendeklarasian bilangan dengan nilai 0
47            jumlah = bilangan; // pemasukkan nilai jumlah yang di dapat dari nilai bilangan
48
49            cout << " Masukkan panjang deret bilangan genap : "; cin >> range;
50            cout << "===== " << endl << endl;
51            cout << " Deret bilangan " << jenis << " yang didapatkan : ";
52
53            for (i = 1; i <= range; i++) {
54                bilangan = bilangan + 2; // bilangan awalnya 0 agar dapat menjadi 2, ditambahkan 2
55                cout << bilangan;
56                if (i < range) {
57                    cout << ", ";
58                }
59            }
60            bilangan = 0; // pengulangan nilai bilangan menjadi 0, agar dapat digunakan dengan normal
61
62            cout << endl;
63            cout << " Hasil penjumlahan dari " << range << " deret bilangan : " << endl << endl;
64            cout << " =>  ";
65
66            for (i = 1; i <= range; i++) {
67                bilangan = bilangan + 2;
68                jumlah = jumlah + bilangan; // jumlah didapatkan dari nilai jumlah awal ditambah bilangan
69                cout << bilangan;
70                if (i < range) {
71                    cout << " + ";
72                }
73            }
74            cout << " = " << jumlah << endl << endl;
75            break;
76
77        default: cout << "----- " << endl << endl;
78            cout << " Pilihan yang anda masukkan salah. " << endl << endl;
79    }
80
81    cout << "===== " << endl;
82    cout << "----- Program Selesai ----- " << endl;
83    cout << "===== ";
84
85    return 0;
86 }
```

Analisis Program :

Program dimulai dengan pemilihan jenis operasi yang akan digunakan dengan menggunakan fungsi “Switch” dimana switch akan memilih operasi sesuai dengan input yang dimasukkan oleh user. Kemudian user akan diminta memasukkan panjang deret bilangan yang memiliki arti berapa bilangan yang akan ditampilkan baik bilangan ganjil maupun bilangan genap. Pilihan pertama adalah bilangan ganjil, untuk mencetak bilangan ganjil kita dapat menggunakan cara “jika deret yang ditampilkan awalnya 1, 2, 3,.. dan seterusnya, maka akan kita ganti tampilan tersebut dengan rumus $= (\text{deret} \times 2) - 1$. Dimana rumus ini dapat merubah bilangan yang ditampilkan menjadi bilangan ganjil yang berurutan, contoh 1, 2, 3, ... menjadi 1, 3, 5,... untuk mencetak hasil jumlah dari deret bilangan ganjil kita dapat menggunakan rumus dimana jumlah bilangan ganjil di kuadratkan, contoh apabila ingin menampilkan 3 bilangan ganjil, maka $3^2 = 9$, sama hasilnya dengan $1 + 3 + 5 = 9$. Untuk pilihan kedua adalah bilangan genap kita mencetak bilangan genap dengan cara menentukan terlebih dahulu nilai dari variable bilangan yaitu 0, untuk memulai menampilkan bilangan yang sesuai kita dapat menambahkan angka 2 pada bilangan, yang memiliki arti (range pertama = bilangan pertama) \Rightarrow (urutan pertama = 0) dari pernyataan tersebut kita dapat memulai perulangan yang menampilkan hasil yang sesuai dengan menjumlahkan bilangan dengan dua, sehingga menampilkan output (range pertama = 2) begitu juga seterusnya. Untuk menampilkan jumlah adalah dengan menjumlahkan jumlah awal (jumlah = bilangan) bilangan awal memiliki nilai 0, maka (jumlah = 0) kemudian ditambahkan dengan bilangan yang akan dicetak yang akan disimpan di variable jumlah nantinya.

Hasil Output :

```
et-bila https://gcc.gnu.org/onlinedocs/libstdc++/manual/section4.html#sec4.1
====
jenis
----- Program Deret Bilangan -----
====
1. Bilangan Ganjil      2. Bilangan Genap
=====
Masukkan pilihan deret bilangan      : 1
Masukkan panjang deret bilangan ganjil : 5
=====
Deret bilangan ganjil yang didapatkan : 1, 3, 5, 7, 9
Hasil penjumlahan dari 5 deret bilangan :
=> 1 + 3 + 5 + 7 + 9 = 25
=====
----- Program Selesai -----
=====
Process exited after 5.535 seconds with return value 0
Press any key to continue . . .
```

```

If __sb is NULL, the stream will set failbit in its
racted
input
his ca
an ex
the c
#1
US
in

C:\Users\asus\OneDrive\Documents\BERKAS\Tuga...
=====
----- Program Deret Bilangan -----
=====
1. Bilangan Ganjil      2. Bilangan Genap
=====
Masukkan pilihan deret bilangan      : 2
Masukkan panjang deret bilangan genap : 5
=====
Deret bilangan genap yang didapatkan : 2, 4, 6, 8, 10
Hasil penjumlahan dari 5 deret bilangan :
=>  2 + 4 + 6 + 8 + 10  =  30
=====
----- Program Selesai -----
=====
Process exited after 4.447 seconds with return value 0
Press any key to continue . . .
23
```

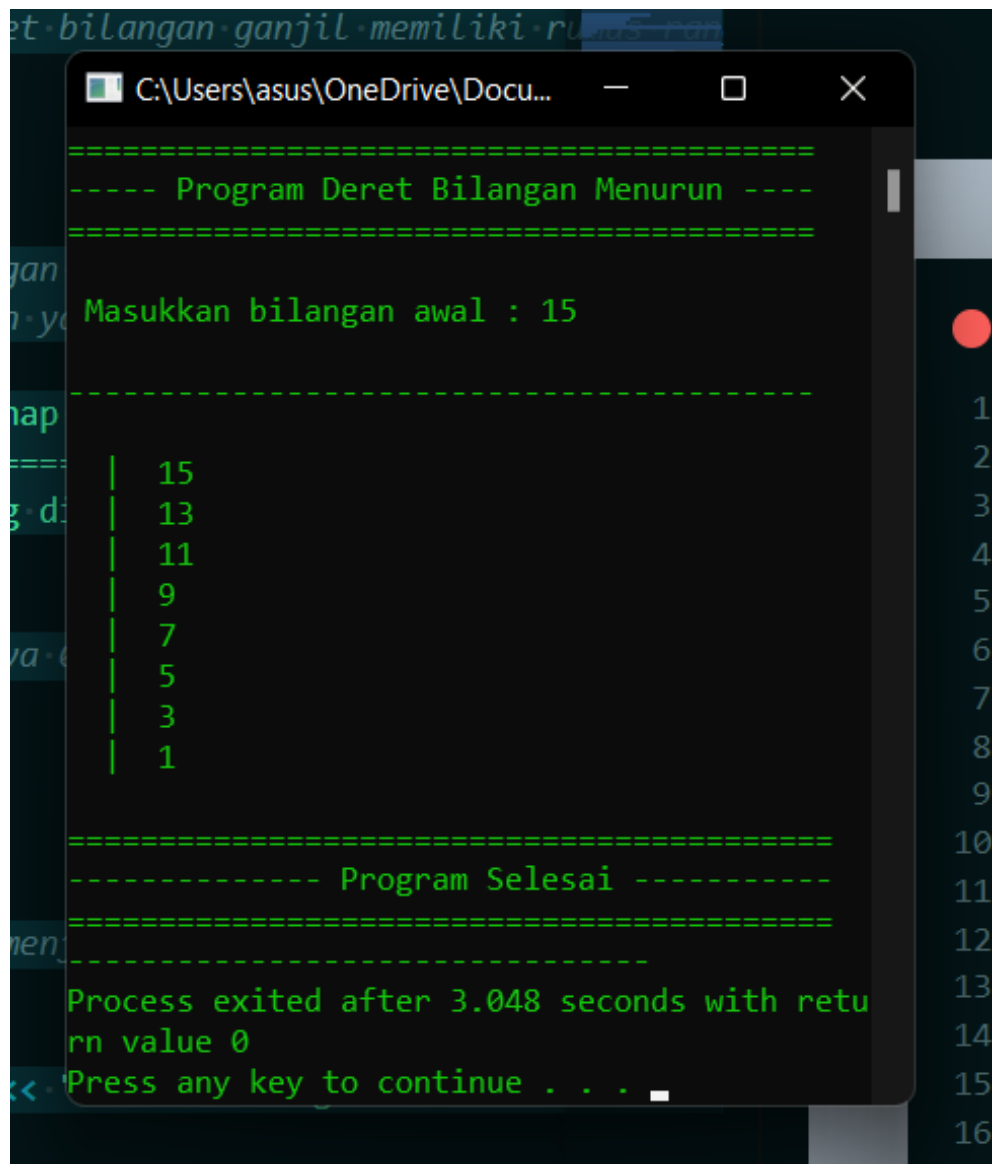
3. Program Deret Bilangan Menurun

```
1  #include <iostream> // package library yang digunakan
2  using namespace std;
3
4  int main() {
5      int awal;
6
7      cout << "===== " << endl;
8      cout << "----- Program Deret Bilangan Menurun ----" << endl;
9      cout << "===== " << endl << endl;
10     cout << " Masukkan bilangan awal : "; cin >> awal;
11     cout << endl;
12     cout << "-----" << endl << endl;
13
14     for(int i = awal; i > 0; i-=2) { // i mengurangi nilai sebesar 2
15         cout << " | " << i << endl;
16     }
17
18     cout << endl;
19     cout << "===== " << endl;
20     cout << "----- Program Selesai -----" << endl;
21     cout << "===== ";
22
23     return 0;
24
25 }
```

Analisis Program :

Program tersebut merupakan program pencetak deret bilangan menurun dengan penurunan sebesar dua kali, contoh (10, 8, 6, 4, 2) dan (9, 7, 5, 3, 1). Dapat berupa bilangan ganjil maupun bilangan genap sesuai dengan bilangan awal yang dimasukkan oleh user. Program diawali dengan memasukkan input bilangan awal yang dilakukan oleh user, untuk memberikan perintah kepada program yang berfungsi sebagai pendeteksi bilangan pertama yang dicetak dan kemudian akan menurunkannya. Program yang digunakan untuk menurunkannya adalah menggunakan increment ($i-=2$) yang memiliki arti i mengurangi nilai dari i sebesar 2. Untuk bilangan awal yang dicetak adalah bilangan yang dimasukkan oleh user, dan batas berjalanya program adalah jika nilai i (bilangan) lebih besar dari 0, apabila bilangan kurang dari 0 maka program looping akan dihentikan. Jadi untuk nilai akhir dari bilangan ganjil adalah 1, sedangkan nilai akhir bilangan genap adalah 2.

Hasil Output :



```
C:\Users\asus\OneDrive\Docu...  
=====   
----- Program Deret Bilangan Menurun -----   
=====   
Masukkan bilangan awal : 15   
-----   
| 15   
| 13   
| 11   
| 9   
| 7   
| 5   
| 3   
| 1   
-----   
----- Program Selesai -----   
-----   
Process exited after 3.048 seconds with return value 0   
Press any key to continue . . .
```


4. Program Pola Piramid

```
1  #include <iostream> // package library yang digunakan
2  using namespace std;
3
4  int main() {
5      int tinggi, baris, jarak, icon;
6
7      cout << "===== " << endl;
8      cout << "----- Program Pola Piramid -----" << endl;
9      cout << "===== " << endl << endl;
10     cout << "Masukkan tinggi piramid : "; cin >> tinggi;
11     cout << endl;
12     cout << "----- " << endl << endl;
13
14     for(baris = 1; baris <= tinggi; baris++) { // mengatur banyak baris
15         cout << " "; // memberi jarak dari kiri layar
16         for (jarak = 1; jarak <= tinggi - baris; jarak++) { // memberi spasi tambahan
17             cout << " "; // spasi untuk menengahkan icon bintang
18         }
19
20         for (icon = 1; icon <= baris; icon++) {
21             cout << " *"; // memberi jarak antar icon, perbandingan 2 : 1 dari jarak
22         }
23         cout << endl;
24     }
25
26     cout << endl;
27     cout << "===== " << endl;
28     cout << "----- Program Selesai -----" << endl;
29     cout << "===== ";
30
31     return 0;
32 }
```

Analisis Program :

Program tersebut merupakan program untuk mencetak pola berbentuk segitiga dengan menggunakan icon symbol bintang untuk menampilkanya. Program diawali dengan perintah memasukkan tinggi segitiga oleh user dimana perintah ini akan menjalankan berapa banyak baris yang akan dibuat pada pola tersebut. Perulangan pertama adalah meneksekusi banyaknya baris dari input tinggi yang telah dimasukkan oleh user. Perulangan kedua adalah mencetak banyaknya jarak spasi yang digunakan, semakin menurun barisnya, jumlah spasi yang dicetak akan semakin sedikit, fungsi dari spasi ini untuk membuat posisi icon bintang agar membentuk layaknya segitiga. Kemudian perulangan ketiga adalah pencetakan icon symbol bintang, banyaknya icon symbol yang dicetak sesuai dengan urutan baris atau tinggi yang dimasukkan oleh user pada awal program. Pada pencetakan icon symbol bintang tersebut kita sebaiknya memberikan spasi terlebih dahulu sebelum icon symbol bintang, dikarenakan hal tersebut mempengaruhi bentuk pola, pola bintang memiliki perbandingan antara (jarak : icon) yaitu (1 : 2), angka dua ini dihitung dari spasi dan symbol bintang.

Hasil Output :

```

=====
----- Program Pola Piramid -----
=====

Masukkan tinggi piramid : 15

-----

          *
        * *
      * * *
    * * * *
  * * * * *
* * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *

=====
----- Program Selesai -----
=====

Process exited after 3.497 seconds with return value 0

```

IV. Kesimpulan

Program perulangan tersebut merupakan salah satu program yang sangat dibutuhkan untuk pengelolaan data pada suatu database sistem ataupun sistem yang lainnya. Dengan adanya praktikum modul perulangan kita dapat belajar bagaimana cara untuk mengefisiensi pemasukan sebuah data dan penghematan waktu pengerjaan. Dengan memberikan perintah yang sesuai dengan algoritma yang dapat dipahami bahasa pemrograman, program ini akan sangat bermanfaat untuk bekal kedepannya. Dan dengan pelaksanaan ini kita dapat membedakan atau menempatkan struktur perulangan yang sesuai dengan kondisi yang sedang kita alami, baik menggunakan for, while, maupun do while.