

LAPORAN AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMECAHAN MASALAH I
MODUL 3
PERULANGAN FOR DAN WHILE



Nama : Ikhwanul Hakim
No. BP : 2111512016
Hari/Tanggal : Jum'at/ 20 Mei 2022
Shift : IIB

Dosen : Rifki Suwandi, M.T.

LABORATORIUM KOMPUTER DAN JARINGAN
JURUSAN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2022

BAB I

PENDAHULUAN

1.1 Tujuan

1. Pratikan mengenali perulangan program C++.
2. Pratikan dapat menganalisa pengulangan dengan statement “ for “.
3. Pratikan dapat menganalisa pengulangan dengan statement ” while ”.
4. Pratikan dapat menganalisa pengulangan Dengan statement ” do while ”.
5. Pratikan dapat menganalisa pernyataan Break.
6. Pratikan dapat menganalisa pernyataan Continue.
7. Pratikan dapat menganalisa Nested Loop.
8. Pratikan dapat membuat program menjadi program yang execute.

1.2 Landasan Teori

1.2.1 Input

Pada C++ terdapat beberapa instruksi dalam memasukan data, dimana beberapa fungsi yang dapat kita gunakan adalah :

a. cin;

- Fungsi pustaka cin digunakan untuk menginput data berupa data numerik, karakter dan string. (Sedangkan pada C digunakan fungsi scan() memakai penentu format contoh : %i (integer), %c (char), dan fungsi scanf() tidak memerlukan penentu lebar field. Scanf() akan memberi pergantian baris secara otomatis). Pemakaian fungsi scanf(), variabelnya harus menggunakan operator alamat &.

b. gets()

- Fungsi gets() digunakan untuk memasukkan data bertipe karakter dan tidak dapat digunakan untuk memasukkan data numerik.
- Harus diakhiri dengan penekanan tombol enter
- Cursor secara otomatis akan pindah baris
- Tidak memerlukan penentu format

c. getchar()

- Fungsi getchar() digunakan untuk membaca data yang bertipe karakter
- Harus diakhiri dengan penekanan tombol enter
- Karakter yang dimasukkan terlihat pada layar

- Pergantian baris secara otomatis

d. `getch()` dan `getche()`

- Fungsi `getch()` dan `getche()` digunakan untuk membaca data karakter.
- Karakter yang dimasukkan tidak perlu diakhiri dengan penekanan tombol enter.
- Tidak memberikan efek pergantian baris secara otomatis
- Jika menggunakan fungsi `getch()` karakter yang dimasukkan tidak akan ditampilkan pada layar sehingga sering digunakan untuk meminta inputan berupa password.
- Sedangkan pada `getche()` karakter yang dimasukkan akan ditampilkan pada layar.

1.2.2. Output

Dalam C++ instruksi untuk menampilkan data yaitu : `cout`.

Contoh :

```
cout<<nama;  
cout<<"Bahasa C++";
```

Sedangkan pada C untuk menampilkan data ada beberapa cara, yakni :

- a. Fungsi `printf()` digunakan untuk menampilkan semua jenis data (numeric dan karakter).
- b. Fungsi `puts()` digunakan untuk menampilkan data string dan secara otomatis akan diakhiri dengan perpindahan baris.
- c. Fungsi `putchar()` digunakan untuk menampilkan sebuah karakter. Dalam menampilkan bilangan pecahan, terdapat cara dalam mengatur tampilan digitnya, dimana bentuk umum pengaturan bilangan pecahan tersebut adalah :

Bentuk Umum :

```
printf("%x.yf", argument);
```

dimana :

x : menyatakan panjang range

y : menyatakan jumlah digit di belakang koma.

Argument : nilai atau variable yang akan ditampilkan.

Contoh : `printf("%4.2f", nilai);`

Artinya variable nilai akan ditampilkan sebanyak 4 digit dengan 2 digit di belakang koma.

1.2.3. Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat diaktifkan atau dipanggil di manapun di dalam program. Umumnya dalam bahasa pemrograman, beberapa fungsi yang sudah disediakan bahasa tersebut. Sebagai contoh dalam bahasa C++, fungsi pustaka cout, cin atau printf(), scanf() (untuk bahasa C), getch() dan lain sebagainya. Dalam bahasa Pascal seperti write() atau writeln(), read atau readln(). Untuk menggunakan fungsi tersebut programmer dapat langsung menuliskan fungsi tersebut.

Disamping fungsi yang sudah disediakan oleh bahasa pemrograman, programmer dapat pula membuat fungsi sendiri sesuai dengan kebutuhannya. Untuk mendefinisikan sendiri fungsi tersebut, harus dideklarasikan terlebih dahulu dalam program yang dibuat kemudian baru bisa menggunakannya. Berikut adalah cara mendeklarasikan fungsi yang dibuat sendiri oleh programmer.

Deklarasi fungsi dalam bahasa C++ :

tipe_fungsi nama_fungsi(parameter_fungsi);

Contoh :

float uas(int nilai);

void output();

int kali(int x, int y);

1.2.4. Definisi Perulangan

Dalam bahasa C++ tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan kita. Misalnya saja, bila kita ingin menginput dan mencetak bilangan dari 1 sampai 100 bahkan 1000, tentunya kita akan merasa kesulitan. Namun dengan struktur perulangan proses, kita tidak perlu menuliskan perintah sampai 100 atau 1000 kali, cukup dengan beberapa perintah saja. Struktur perulangan dalam bahasa C mempunyai bentuk yang bermacam-macam. Sebuah/kelompok instruksi diulang untuk jumlah

pengulangan tertentu. Baik yang terdefinisi sebelumnya ataupun tidak. Struktur pengulangan terdiri atas dua bagian :

- (1). Kondisi pengulangan yaitu ekspresi boolean yang harus dipenuhi untuk melaksanakan pengulangan;
- (2). Isi atau badan pengulangan yaitu satu atau lebih pernyataan (aksi) yang akan diulang.

1.2.5. Statement For

Struktur perulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan for tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Pernyataan for digunakan untuk melakukan looping. Pada umumnya looping yang dilakukan oleh for telah diketahui batas awal, syarat looping dan perubahannya. Selama kondisi terpenuhi, maka pernyataan akan terus dieksekusi. Bentuk umum perulangan for adalah sebagai berikut :

For (inisialisasiNilai; SyaratPerulangan; PerubahanNilai) { Statement yang diulang;

- Ungkapan1 merupakan statement awal (inisialisasi)
- Ungkapan2 merupakan kondisi/syarat perulangan dilakukan
- Ungkapan3 merupakan statement control untuk perulangan
- Statement merupakan pernyataan/perintah yang dijalankan jika syarat terpenuhi.

1.2.6. Statement While

Perulangan WHILE banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar ($\neq 0$) dan akan berhenti bila kondisinya bernilai salah ($=0$).

Berikut gambaran umumnya :

```
while (syarat pengulangan)
{
    statement yang dijalankan;
    statement control:
}
```

Kondisi dalam WHILE juga dapat menggunakan operator logika. Misalnya:

```
while((kondisi 1) && (kondisi2))
{
    Blok Pernyataan;
}
```

1.2.7. Statement Do-While

Perintah DO ... WHILE hampir sama dengan WHILE sebelumnya. Gambaran secara umum:

```
do
{
    Blok Pernyataan;
}
while(kondisi);
```

Perbedaan dengan WHILE sebelumnya yaitu bahwa pada DO WHILE statement perulangannya dilakukan terlebih dahulu baru kemudian di cek kondisinya. Sedangkan WHILE kondisi dicek dulu baru kemudian statement perulangannya dijalankan. Akibat dari hal ini adalah dalam DO WHILE minimal terdapat 1x perulangan. Sedangkan WHILE dimungkinkan perulangan tidak pernah terjadi yaitu ketika kondisinya langsung bernilai FALSE.

1.2.8. Pernyataan Break

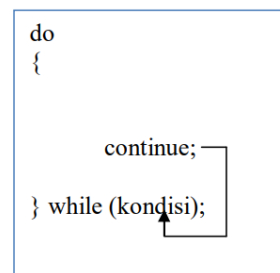
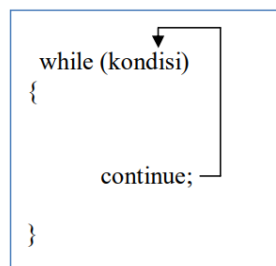
Pernyataan break telah dibahas pada pernyataan pengambilan keputusan switch. Pernyataan break ini berfungsi untuk keluar dari struktur switch. Selain itu pernyataan break berfungsi keluar dari perulangan (for, while dan do-while).

Jika pernyataan break dikerjakan, maka eksekusi akan dilanjutkan ke pernyataan yang terletak sesudah akhir dari badan perulangan (loop).

1.2.9. Pernyataan Continue

Kegunaannya kebalikannya dari break. Dipakai untuk mengarahkan eksekusi ke putaran atau iterasi berikutnya pada pernyataan perulangan. Efek dari perintah continue:

- 1) Pada for: Ungkapan ke tiga pada for (ungkapan terkanan yg terletak di dalam () pada for) akan dijalankan dan kemudian ungkapan ke dua diuji lagi.
- 2) Pada while dan do- while : Pengujian terhadap ungkapan pada while dilakukan kembali.



BAB II PERCOBAAN

2.1 Percobaan 3.1

```
bug  [*] Untitled1  percobaan 3.1.cpp
1  #include <iostream>
2  using namespace std;
3  main(){
4      for (int i=0; i<4; i++){
5          cout << i;
6      }
7      cout<<endl;
8      for (int i=3; i>0; i--){
9          cout<<i;
10     }
11     cout<<endl;
12     cout<<endl;
13
14     int n, sigma = 0;
15     cout<<"Masukkan bilangan integer positif : "; cin>>n;
16     for (int i=1; i <= n; i++){
17         sigma += i;
18     }
19     cout<< "Jumlah dari "<<n<<" bilangan pertama adalah : "<<sigma<<endl;
20 }
```

```
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.1.exe
0123
321

Masukkan bilangan integer positif : 100
Jumlah dari 100 bilangan pertama adalah : 5050

-----
Process exited after 2.842 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.1.exe
0123
321

Masukkan bilangan integer positif : 3
Jumlah dari 3 bilangan pertama adalah : 6

-----
Process exited after 1.72 seconds with return value 0
Press any key to continue . . .
```

2.2 Percobaan 3.2

```
[*] Untitled1  percobaan 3.1.cpp  percobaan 3.2.cpp
1  #include <iostream>
2  using namespace std;
3  main(){
4      int i=1, n, sigma = 0;
5      cout<<" Masukkan bilangan integer positif : ";cin>>n;
6      while (i<=n){
7          sigma += i;
8          i++;
9      }
10     cout<<"Jumlah dari "<<n<<" bilangan pertama adalah "<<sigma<<endl;
11 }
```



```
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.2.exe

Masukkan bilangan integer positif : 5
Jumlah dari 5 bilangan pertama adalah 15

-----
Process exited after 5.229 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.2.exe

Masukkan bilangan integer positif : 10
Jumlah dari 10 bilangan pertama adalah 55

-----
Process exited after 2.505 seconds with return value 0
Press any key to continue . . .
```

2.3 Percobaan 3.3

```
g  [*] Untitled1  percobaan 3.1.cpp  percobaan 3.2.cpp  percobaan 3.3.cpp

1  #include <iostream>
2  using namespace std;
3  main(){
4      int n, f = 1;
5      cout<<"Masukkan bil. integer positif "; cin>>n;
6      do {
7          f *=n;
8          n--;
9      } while (n >= 1);
10     cout<<n<<"Faktorial adalah : "<<f<<endl;
11 }
```

```
(globals)
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.3.exe
Masukkan bil. integer positif : 3
0Faktorial adalah : 6

-----
Process exited after 3.133 seconds with return value 0
Press any key to continue . . .
```

```
2 using namespace std;
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.3.exe
Masukkan bil. integer positif : 5
0Faktorial adalah : 120

-----
Process exited after 1.593 seconds with return value 0
Press any key to continue . . .
```

2.4 Percobaan 3.4

```
Classes
C:\Users\TOSHIBA\Documents\anjay\percobaan 3.4.exe
Masukkan bilangan integer positif : 5
1
2
3
4
5

-----
Process exited after 3.948 seconds with return value 0
Press any key to continue . . .
```

```
globals)
perbug  percobaan 3.1.cpp  percobaan 3.2.cpp  percobaan 3.3.cpp  percobaan 3.4.cpp

1  #include <iostream>
2  using namespace std;
3  main(){
4      int i=1, n, sigma = 0;
5      cout << " Masukkan bilangan integer positif : ";cin>>n;
6      while (1){
7          if (i > n) break; // loop berhenti disini ketika i>n
8          cout<< i <<endl;
9          i++;
10     }
11 }
```

C:\Users\TOSHIBA\Documents\anjay\percobaan 3.4.exe

Masukkan bilangan integer positif : 10

1
2
3
4
5
6
7
8
9
10

Process exited after 7.633 seconds with return value 0
Press any key to continue . . .

2.5 Percobaan 3.5

```
globals)
perbug  percobaan 3.1.cpp  percobaan 3.2.cpp  percobaan 3.3.cpp  percobaan 3.4.cpp

1  #include <iostream>
2  using namespace std;
3  main(){
4      int n;
5      for (; ;) {
6          cout<<"Masukkan integer : "; cin>>n;
7          if (n % 2 ==0) continue;
8          else if (n % 2 > 0) break;
9      }
10     cout << "\tAkhir Loop" << endl;
11 }
```

```
2 using namespace std;

C:\Users\TOSHIBA\Documents\anjay\percobaan 3.4.exe

Masukkan integer : 10
Masukkan integer : 10
Masukkan integer : 10
Masukkan integer : 10
Masukkan integer : 6
Masukkan integer : 8
Masukkan integer : 4
Masukkan integer : 3
    Akhir Loop

-----
Process exited after 11.34 seconds with return value 0
Press any key to continue . . .
```

```
Edit Search View Project Execute Tools AStyle Window Help

C:\Users\TOSHIBA\Documents\anjay\percobaan 3.4.exe

Masukkan integer : 5
    Akhir Loop

-----
Process exited after 0.8144 seconds with return value 0
Press any key to continue . . .
```

2.6 Percobaan 3.6

```
C:\Users\TOSHIBA\Documents\anjay\Untitled8.exe

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

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

-----
Process exited after 0.1134 seconds with return value 0
Press any key to continue . . .
```

```

1  #include <iostream>
2  using namespace std;
3  main(){
4      for (int i=1; i<=3; i++) {
5          for (int j=1; j<=5; j++) {
6              cout << " * ";
7          }
8          cout << endl;
9      }
10     cout << endl;
11
12     int i=1;
13     while(i<=3){
14         int j=1;
15         while (j<=5 ){
16             cout << " * ";
17             j++;
18         }
19         cout<<endl;
20         i++;
21     }
22 }

```

BAB III

TUGAS

3.1 Tugas 1

```

(globals)
sses Debug tugas modul 3 (1).cpp
1  #include<iostream>
2  using namespace std;
3  main(){
4      for(int i=1; i<=3; i++){
5          for(int j=1;j<=i;j++){
6              cout<<" * ";
7          }
8          cout<<endl;
9      }
10
11     for(int i=1; i<=2; i++){
12         for(int j=i;j<=2;j++){
13             cout<<" * ";
14         }cout<<endl;
15     }cout<<endl;
16
17     for (int i=1;i<=3;i++){
18         for(int j=1; j<=3; j++){
19             cout<<" * ";
20         }cout<<endl;
21     }
22     for(int i=1;i<=2;i++){
23         for(int j=1;j<=i+1;j++){
24             cout<<" * ";
25         }
26         cout<<endl;
27     }
28

```

```
Project
C:\Users\TOSHIBA\Documents\anjay\tugas modul 3 (1).exe
*
* *
* * *
*
*
* * *
* *
*
* *
* * *
* * *

-----
Process exited after 0.1228 seconds with return value 0
Press any key to continue . . .
```

3.2 Tugas 2

```
ig tugas modul 3 (1).cpp tugas modul 3 (2).cpp
1 #include <iostream>
2 using namespace std;
3 main(){
4     cout << "Tugas 2 Modul 3 Ikhwanul Hakim" << endl;
5     cout << " " << endl;
6
7     for(int i=1; i<=5; i++){
8         for(int j=1; j<=5; j++){
9             if((i==j) || (i+j==6)){
10                 cout << " * ";
11             }
12             else{
13                 cout << "  ";
14             }
15         }
16         cout << endl;
17     }
18     cout << endl;
19 }
20 }

C:\Users\TOSHIBA\Documents\anjay\tugas modul 3 (2).exe
Tugas 2 Modul 3 Ikhwanul Hakim
*
* *
*
* *
* *

-----
Process exited after 0.156 seconds with return value 0
Press any key to continue . . .
```


3.3 Tugas 3

```
tugas modul 3 (1).cpp  tugas modul 3 (2).cpp  tugas modul 3 (3).cpp
1  //menghitung banyak bilangan genap
2  #include <iostream>
3  using namespace std;
4  main(){
5      cout <<"Tugas 3 Modul 3 Ikhwanul Hakim" <<endl;
6      cout<<" " <<endl;
7
8      int n, m;
9      m==0;
10     for(int i=1; i<=10; i++){
11         cout<<" Masukkan bilangan acak : ";cin>>n;
12         if(n%2==0){m=m+1;
13         }
14     }
15     cout<<endl;
16     cout<<"Banyaknya bilangan genap : " <<m<<endl;
17 }
```

C:\Users\TOSHIBA\Documents\anjay\tugas modul 3 (3).exe

Tugas 3 Modul 3 Ikhwanul Hakim

Masukkan bilangan acak : 10
Masukkan bilangan acak : 9
Masukkan bilangan acak : 8
Masukkan bilangan acak : 7
Masukkan bilangan acak : 6
Masukkan bilangan acak : 5
Masukkan bilangan acak : 4
Masukkan bilangan acak : 3
Masukkan bilangan acak : 2
Masukkan bilangan acak : 1

Banyaknya bilangan genap : 5

Process exited after 11.6 seconds with return value 0
Press any key to continue . . .

LKJ

3.4 Tugas 4

```
Debug | tugas modul 3 (1).cpp | tugas modul 3 (2).cpp | tugas modul 3 (3).cpp | tugas modul 3 (4).
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      cout<<" TABEL PERKALIAN 1 - 10 "<<endl;
6      cout<<" _____"<<endl;
7
8
9      for (int i=1; i<=10; i++){
10         for (int j=1; j<=5; j++){
11             cout<<j<<" X "<<i<<" = "<<j*i<<" \t ";
12         }
13         cout<<endl;
14     }
15     cout<<endl;
16     cout<<endl;
17
18     for (int i=1; i<=10; i++){
19         for (int j=6; j<=10; j++){
20             cout<<j<<" X "<<i<<" = "<<j*i<<" \t ";
21         }
22         cout<<endl;
23     }
24 }
```

```
==== C:\Users\PC31161\Documents\angry\tugas modul 3 (1)\tks
TABEL PERKALIAN 1 - 10
1 X 1 = 1      2 X 1 = 2      3 X 1 = 3      4 X 1 = 4      5 X 1 = 5
1 X 2 = 2      2 X 2 = 4      3 X 2 = 6      4 X 2 = 8      5 X 2 = 10
1 X 3 = 3      2 X 3 = 6      3 X 3 = 9      4 X 3 = 12     5 X 3 = 15
1 X 4 = 4      2 X 4 = 8      3 X 4 = 12     4 X 4 = 16     5 X 4 = 20
1 X 5 = 5      2 X 5 = 10     3 X 5 = 15     4 X 5 = 20     5 X 5 = 25
1 X 6 = 6      2 X 6 = 12     3 X 6 = 18     4 X 6 = 24     5 X 6 = 30
1 X 7 = 7      2 X 7 = 14     3 X 7 = 21     4 X 7 = 28     5 X 7 = 35
1 X 8 = 8      2 X 8 = 16     3 X 8 = 24     4 X 8 = 32     5 X 8 = 40
1 X 9 = 9      2 X 9 = 18     3 X 9 = 27     4 X 9 = 36     5 X 9 = 45
1 X 10 = 10    2 X 10 = 20    3 X 10 = 30    4 X 10 = 40    5 X 10 = 50

6 X 1 = 6      7 X 1 = 7      8 X 1 = 8      9 X 1 = 9      10 X 1 = 10
6 X 2 = 12     7 X 2 = 14     8 X 2 = 16     9 X 2 = 18     10 X 2 = 20
6 X 3 = 18     7 X 3 = 21     8 X 3 = 24     9 X 3 = 27     10 X 3 = 30
6 X 4 = 24     7 X 4 = 28     8 X 4 = 32     9 X 4 = 36     10 X 4 = 40
6 X 5 = 30     7 X 5 = 35     8 X 5 = 40     9 X 5 = 45     10 X 5 = 50
6 X 6 = 36     7 X 6 = 42     8 X 6 = 48     9 X 6 = 54     10 X 6 = 60
6 X 7 = 42     7 X 7 = 49     8 X 7 = 56     9 X 7 = 63     10 X 7 = 70
6 X 8 = 48     7 X 8 = 56     8 X 8 = 64     9 X 8 = 72     10 X 8 = 80
6 X 9 = 54     7 X 9 = 63     8 X 9 = 72     9 X 9 = 81     10 X 9 = 90
6 X 10 = 60    7 X 10 = 70    8 X 10 = 80    9 X 10 = 90    10 X 10 = 100

-----
Process exited after 0.202 seconds with return value 0
```

BAB IV

ANALISA

4.1 Tugas 1

4.1.1 Poin A

Untuk dapat menampilkan program dengan output pola seperti pada soal tugas, kita dapat menggunakan program c++ dengan perulangan for.

4.1.2 Poin B

Dengan menginputkan inisialisasi dan diikuti syarat perulangan for ke program dan ditambah pengubah nilai pencacah, kita dapat menciptakan sebuah program dengan output pola bintang seperti yang dimaksud pada tugas.

4.2 Tugas 2

4.2.1. Poin A

Untuk dapat menampilkan output dari program C++ dengan pola silang bintang, kita dapat menggunakan program C++ dengan perulangan for.

4.2.2 Poin B

Dengan menginputkan inisialisasi dan diikuti syarat perulangan for ke program dan ditambah pengubah nilai pencacah, dan juga dengan menambahkan cout baris kosong, kita dapat menciptakan sebuah program dengan output pola bintang silang seperti yang dimaksud pada tugas.

4.3 Tugas 3

4.3.1 Poin A

Untuk dapat membuat program dengan input 10 buah bilangan bulat sembarang dari keyboard dan output keterangan bilangan mana yang genap

4.3.2 Poin B

Dengan menggunakan program C++ dan dengan tambahan perulangan for, dan dengan memasukkan kondisi dan syarat, yaitu (`int i=1; i<=10; i++`) dan bilangan genap apabila `n%2==0` (n habis dibagi 2), kita dapat membuat program dengan output menentukan bilangan genap dari 10 bilangan bulat yang kita inputkan dari keyboard.

4.4 Tugas 4

4.4.1 Poin A

Untuk dapat membuat program C++ dengan output tabel perkalian 1 – 10, kita dapat memanfaatkan perulangan for.

4.4.2 Poin B

Dengan memasukkan inisialisasi / memberi nilai awal $i=1$ pada program kemudian menambahkan syarat pengulangan $i \leq 10$, dan dengan memberi pengubah nilai pencacah $++$ untuk mengatur kenaikan nilai pencacah, dan kita juga memberi nilai for kedua yaitu ($j=1; j \leq 5; j++$) agar dalam satu baris pertama hanya menampilkan perkalian 1 – 5. Kemudian lakukan langkah yang sama, hanya saja berbeda di for yang kedua yaitu ($int j=6; j \leq 10; j++$) agar di baris kedua atau baris bawah menampilkan perkalian 6-10. Kita dapat membuat sebuah program dengan output perkalian 1 – 10 seperti yang kita inginkan.

BAB V

PENUTUP

5.1 Bentuk Perulangan for dan while memiliki banyak sekali penerapan dan kegunaan. Dan juga sangat ditekankan ketelitian disini, karena salah sedikit saja bisa membuat program gagal dijalankan, dan pada kasus loop dapat membuat program menjadi endless loop. Dan juga sangat penting untuk memilih kondisi perulangan yang tepat dengan kasus program yang ingin kita ciptakan.

5.2 Saran

Dalam melakukan perulangan, yang terpenting adalah mengetahui dan menguasai bentuk umum dari perulangan for dan while itu sendiri terlebih dahulu, karena itu merupakan pegangan dasar dari materi kita kali ini. Jika masih belum menguasai bentuk umum dari for dan while maka akan menyulitkan saat proses pemrograman.

DAFTAR PUSTAKA

- Anonim. Modul Praktikum Algoritma dan Pemrograman. Padang:
Laboratorium Komputer dan Jaringan, Jurusan Teknik
Komputer, 2022.
- Anonim. Modul Pengantar Bahasa Pemrograman C++. Padang: Laboratorium
Komputer dan Jaringan, Jurusan Teknik Komputer, 2022.
- Anonim. Modul Pratikum C++ Pemrograman Dasar. Malang, Jurusan
Teknik Elektro, 2017.
- Anonim. Modul Perulangan Dalam C++. Yogyakarta, 2022
- Friyadie, dkk. Modul Praktikum Pemrograman C++ Menggunakan Borland
C++ 5.02. Jakarta, 2014.

