

# **Laporan Praktikum 6**

## **Dasar Dasar Pemrograman**

“Looping 2”



Muhammad Azhar Rasyad

0110217029

Teknik Informatika 1

**Sekolah Tinggi Teknologi Terpadu Nurul Fikri**  
**2017**

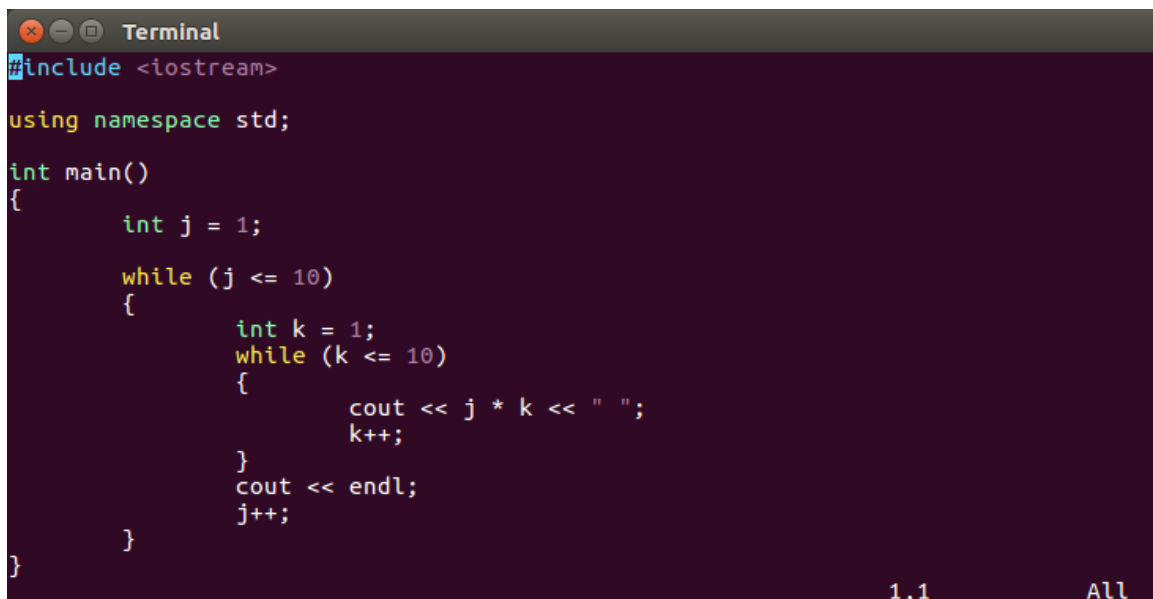
# Laporan Praktikum Looping 2

## Nested While

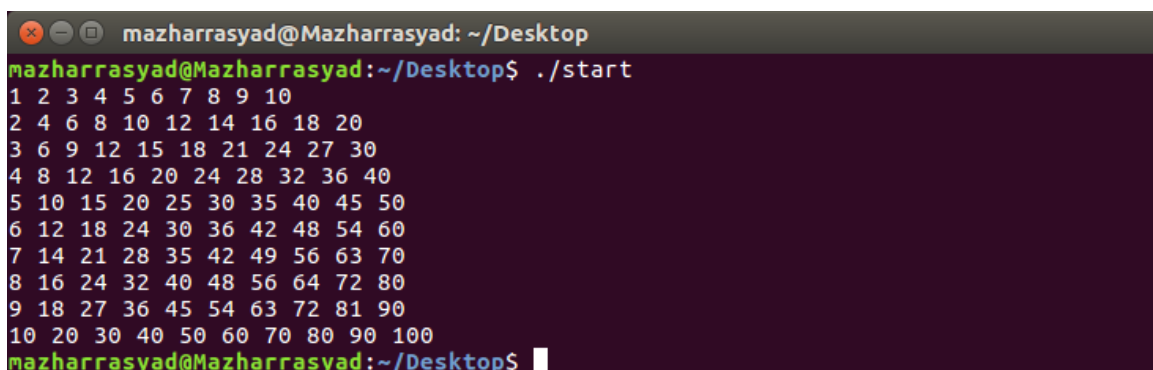
Nested while merupakan kondisi pengulangan while dan dalam while tersebut terdapat while juga untuk menampilkan baris dan kolom, contoh dalam nested while sebagai berikut :

```
while (batasan)  
{  
    kondisi  
    while (batasan)  
    {  
        kondisi  
    }  
}
```

Pada gambar dibawah ini merupakan contoh nested while untuk menampilkan bilangan hasil perkalian hingga baris dan kolom terpenuhi. while pertama digunakan untuk baris, while kedua digunakan untuk kolom jadi setiap pengulangan akan menampilkan baris terlebih dahulu kemudian kolomnya akan diselesaikan sampai batasannya kemudian melanjutkan ke baris selanjutnya.



```
Terminal  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    int j = 1;  
    while (j <= 10)  
    {  
        int k = 1;  
        while (k <= 10)  
        {  
            cout << j * k << " ";  
            k++;  
        }  
        cout << endl;  
        j++;  
    }  
}
```




```
mazharrasyad@Mazharrasyad: ~/Desktop  
mazharrasyad@Mazharrasyad:~/Desktop$ ./start  
1 2 3 4 5 6 7 8 9 10  
2 4 6 8 10 12 14 16 18 20  
3 6 9 12 15 18 21 24 27 30  
4 8 12 16 20 24 28 32 36 40  
5 10 15 20 25 30 35 40 45 50  
6 12 18 24 30 36 42 48 54 60  
7 14 21 28 35 42 49 56 63 70  
8 16 24 32 40 48 56 64 72 80  
9 18 27 36 45 54 63 72 81 90  
10 20 30 40 50 60 70 80 90 100  
mazharrasyad@Mazharrasyad:~/Desktop$
```

## Nested Do While

Nested while merupakan kondisi pengulangan while dan dalam while tersebut terdapat while juga untuk menampilkan baris dan kolom, contoh dalam nested while sebagai berikut :

```
do
{
    kondisi
    do
    {
        kondisi
    }
    while (batasan)
}
while (batasan)
```

Pada gambar dibawah ini merupakan contoh nested do while untuk menampilkan bilangan hasil perkalian hingga baris dan kolom habis. do while pertama digunakan untuk baris, do while kedua digunakan untuk kolom jadi setiap pengulangan akan menampilkan baris terlebih dahulu kemudian kolomnya akan diselesaikan sampai batasannya kemudian melanjutkan ke baris selanjutnya.

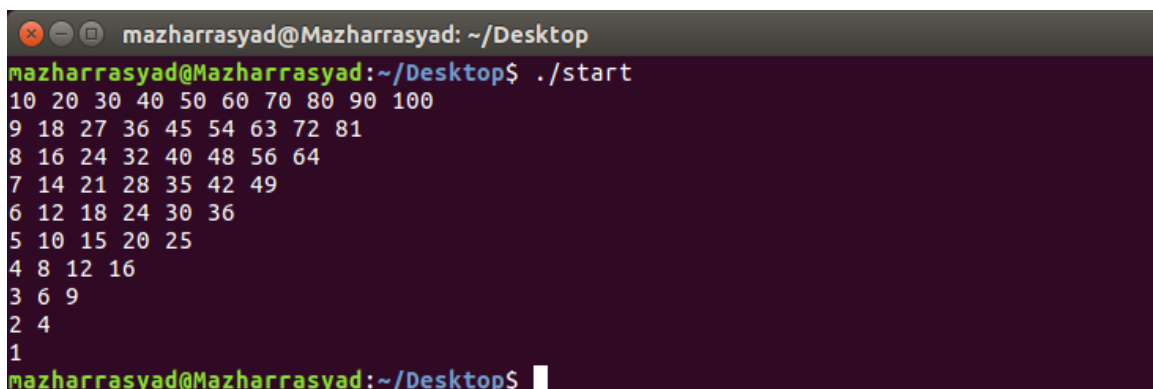


```
#include <iostream>

using namespace std;

int main()
{
    int i = 10;

    do
    {
        int j = 1;
        do
        {
            cout << i * j << " ";
            j++;
        }
        while (j <= i);
        cout << endl;
        i--;
    }
    while (i >= 1);
}
```



```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
10 20 30 40 50 60 70 80 90 100
9 18 27 36 45 54 63 72 81
8 16 24 32 40 48 56 64
7 14 21 28 35 42 49
6 12 18 24 30 36
5 10 15 20 25
4 8 12 16
3 6 9
2 4
1
mazharrasyad@Mazharrasyad:~/Desktop$
```

## Continue Break

continue merupakan kondisi untuk melanjutkan pengulangan jika kondisinya sesuai sedangkan break merupakan kondisi untuk menghentikan pengulangan jika kondisinya sesuai. Contoh dalam continue dan break sebagai berikut :

**(Looping)**

```
{  
    if (condition = true)  
        continue  
}
```

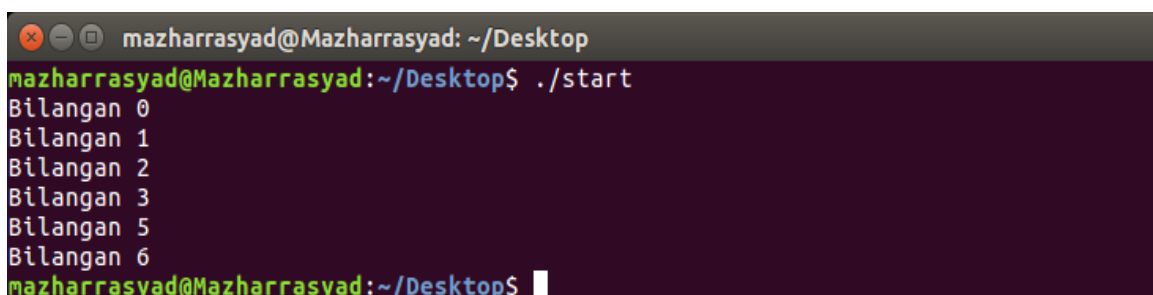
**(Looping)**

```
{  
    if (condition = true)  
        break  
}
```

Pada gambar dibawah ini merupakan contoh continue dan break untuk menampilkan bilangan 1 sampai 10, continue terjadi jika bilangan sampai 4 sedangkan break terjadi jika bilangan sampai 6. Dapat dilihat bahwa ketika continue proses pengulangan akan terus berjalan tetapi jika break maka pengulangan akan berhenti, itulah sebabnya continue diletakkan sebelum break supaya pengulangan dapat berjalan.



```
Terminal  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    int i;  
  
    for (i = 0; i < 10; i++)  
    {  
        if (i == 4)  
            continue;  
  
        cout << "Bilangan " << i << endl;  
  
        if (i == 6)  
            break;  
    }  
}
```



```
mazharrasyad@Mazharrasyad: ~/Desktop  
mazharrasyad@Mazharrasyad:~/Desktop$ ./start  
Bilangan 0  
Bilangan 1  
Bilangan 2  
Bilangan 3  
Bilangan 5  
Bilangan 6  
mazharrasyad@Mazharrasyad:~/Desktop$
```

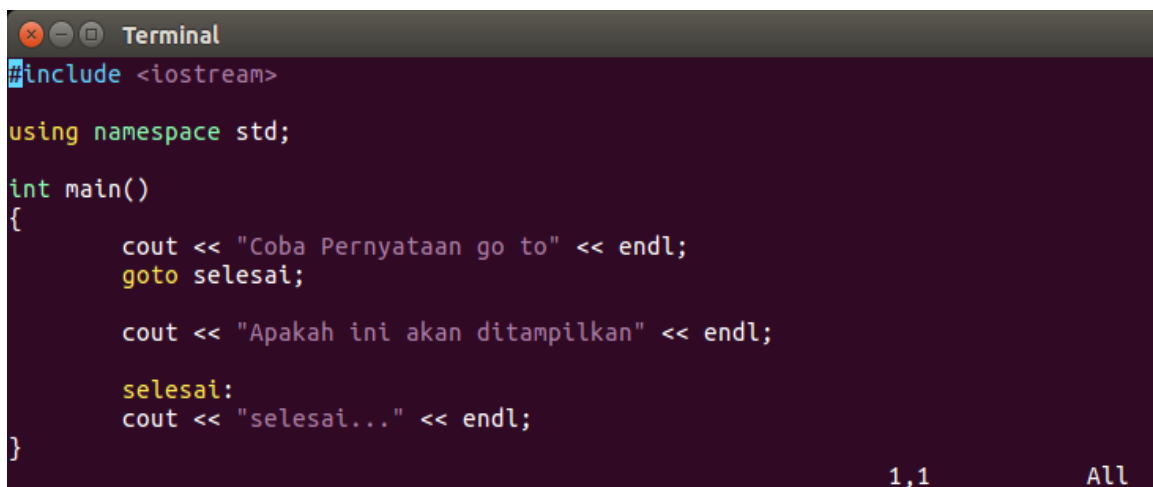
## Goto

goto merupakan kondisi untuk berpindah ke tujuan yang telah ditentukan, kondisi ini dapat menjadi pengulangan jika kondisi tersebut tidak sesuai dengan ketentuan. Contoh goto sebagai berikut :

**goto nama\_tujuan**

**nama\_tujuan :**

Pada gambar dibawah ini merupakan contoh goto untuk langsung berpindah ke tujuan dengan melewati kondisi yang seharusnya lebih dulu dijalankan. Contoh ini mungkin bukan termasuk pengulangan karena tidak adanya kondisi yang dijalankan berulang kali.



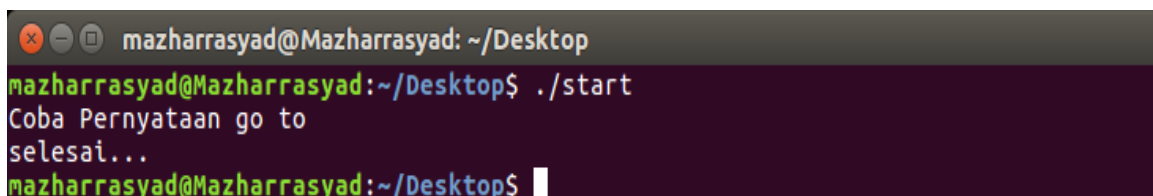
```
Terminal
#include <iostream>

using namespace std;

int main()
{
    cout << "Coba Pernyataan go to" << endl;
    goto selesai;

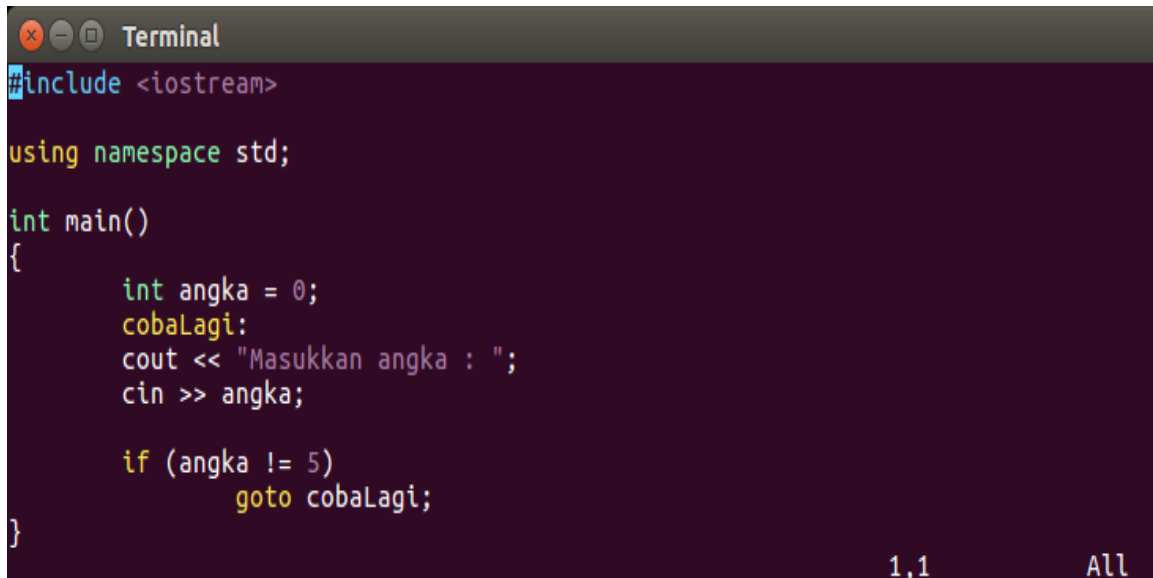
    cout << "Apakah ini akan ditampilkan" << endl;

    selesai:
    cout << "selesai..." << endl;
}
```



```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
Coba Pernyataan go to
selesai...
mazharrasyad@Mazharrasyad:~/Desktop$
```

Contoh goto berikutnya untuk memasukkan angka sesuai dengan ketentuan, contoh ini termasuk pengulangan karena jika angka yang dimasukkan tidak sesuai dengan ketentuan maka kondisi tersebut akan terus diulang sampai sesuai ketentuan, dan jika angkanya sudah sesuai maka kondisi tersebut akan berpindah ke tujuan yang telah ditentukan.



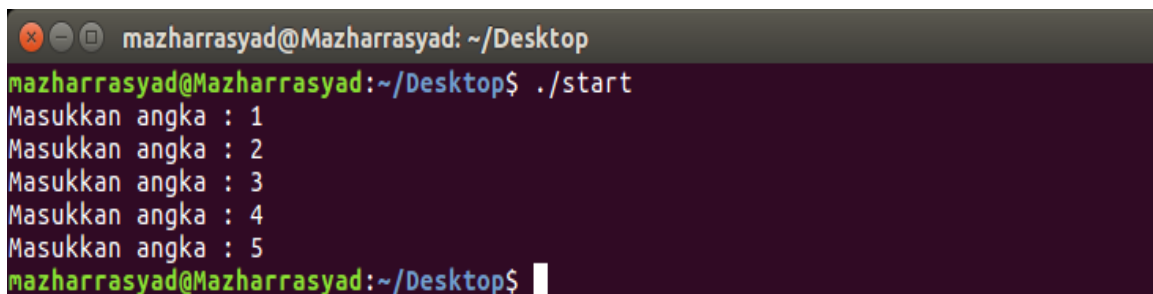
```
Terminal
#include <iostream>

using namespace std;

int main()
{
    int angka = 0;
    cobaLagi:
    cout << "Masukkan angka : ";
    cin >> angka;

    if (angka != 5)
        goto cobaLagi;
}
```

1,1 All



```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
Masukkan angka : 1
Masukkan angka : 2
Masukkan angka : 3
Masukkan angka : 4
Masukkan angka : 5
mazharrasyad@Mazharrasyad:~/Desktop$
```

Buat program dengan **nested for** perkalian 1 sampai 5

```
Terminal
#include <iostream>

using namespace std;

int main()
{
    int i,j,k;

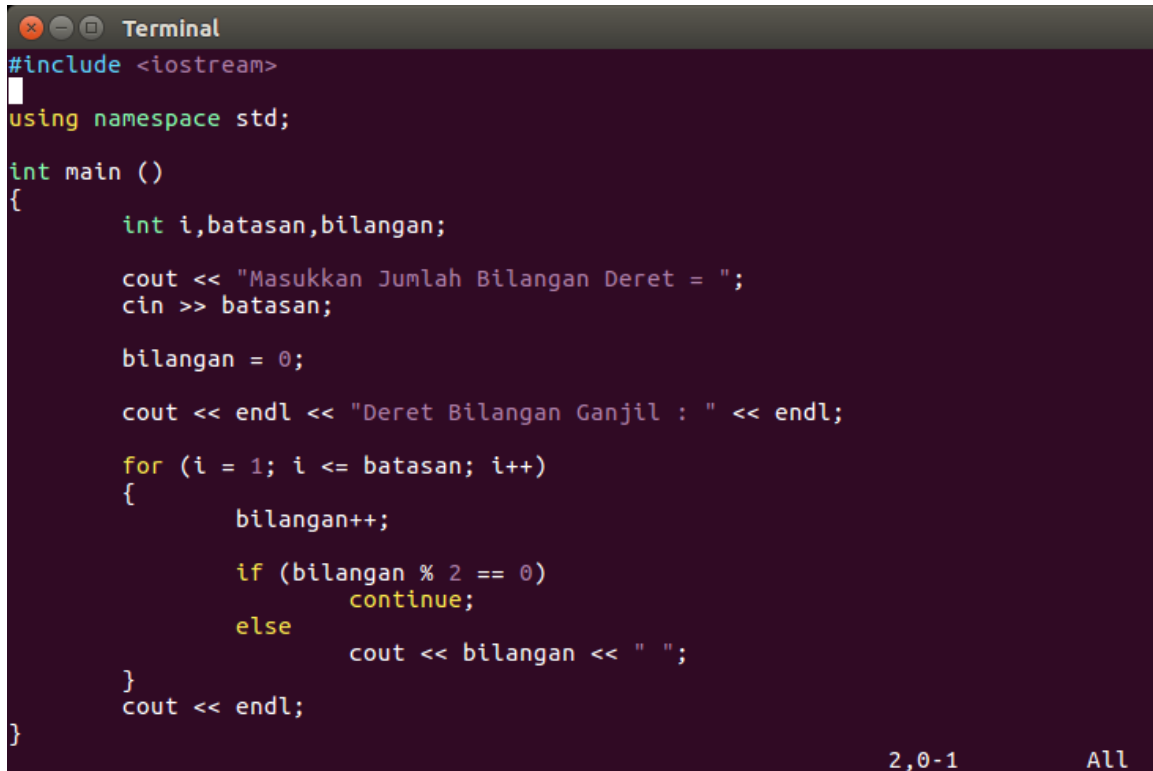
    for (i = 1; i <= 5; i++)
    {
        for (j = 1; j <= 5; j++)
        {
            k = i * j;
            cout << j << " * " << i << " = " << k << "\t";
        }
        cout << endl;
    }
}
```

1,1 All

```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
1 * 1 = 1    2 * 1 = 2    3 * 1 = 3    4 * 1 = 4    5 * 1 = 5
1 * 2 = 2    2 * 2 = 4    3 * 2 = 6    4 * 2 = 8    5 * 2 = 10
1 * 3 = 3    2 * 3 = 6    3 * 3 = 9    4 * 3 = 12   5 * 3 = 15
1 * 4 = 4    2 * 4 = 8    3 * 4 = 12   4 * 4 = 16   5 * 4 = 20
1 * 5 = 5    2 * 5 = 10   3 * 5 = 15   4 * 5 = 20   5 * 5 = 25
mazharrasyad@Mazharrasyad:~/Desktop$
```

## Latihan

2. Buatlah program untuk menampilkan bilangan ganjil sampai deret ke-n, dengan n diinputkan sendiri dari keyboard. Jika inputan = 10, maka hasil 1 3 5 7



```
Terminal
#include <iostream>
using namespace std;
int main ()
{
    int i,batasan,bilangan;

    cout << "Masukkan Jumlah Bilangan Deret = ";
    cin >> batasan;

    bilangan = 0;

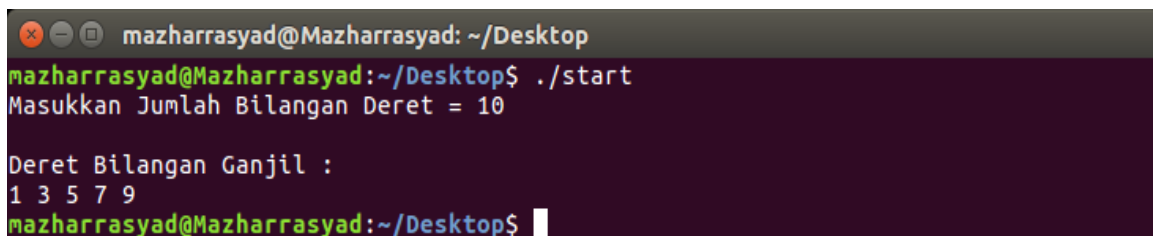
    cout << endl << "Deret Bilangan Ganjil : " << endl;

    for (i = 1; i <= batasan; i++)
    {
        bilangan++;

        if (bilangan % 2 == 0)
            continue;
        else
            cout << bilangan << " ";

    }
    cout << endl;
}
```

2,0-1 All



```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
Masukkan Jumlah Bilangan Deret = 10

Deret Bilangan Ganjil :
1 3 5 7 9
mazharrasyad@Mazharrasyad:~/Desktop$
```



3. Dengan menggunakan pernyataan break dan continue, buatlah program untuk membuat program dengan input n, dan output, bilangan ganjil kecuali kelipatan 7 dan 11 mulai dari 1 sampai < n atau bilangan tersebut <100

Contoh Input : 20

Output : 1 3 5 9 13 15 17 19

```
Terminal
#include <iostream>
using namespace std;
int main ()
{
    int i,j,batasan,bilangan;

    cout << "Masukkan Jumlah Bilangan Deret = ";
    cin >> batasan;

    bilangan = 0;
    i = 0;

    cout << endl << "Deret Bilangan Ganjil Kecuali 7 dan 11 : " << endl;
    while (batasan)
    {
        i++;
        if (batasan != i)
        {
            bilangan++;

            if (bilangan % 2 == 0)
                continue;
            else
            {
                if (bilangan % 7 == 0 || bilangan % 11 == 0)
                    continue;
                else
                    cout << bilangan << " ";
            }
        }
        else if (batasan == i)
            break;
    }
    cout << endl;
}
```

```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
Masukkan Jumlah Bilangan Deret = 20

Deret Bilangan Ganjil Kecuali 7 dan 11 :
1 3 5 9 13 15 17 19
mazharrasyad@Mazharrasyad:~/Desktop$
```