

# **LAPORAN AKHIR**

## **ALGORITMA DAN PEMROGRAMAN**

### **LAPORAN KE-5**



### **Disusun Oleh:**

Nama: Andri Firman Saputra

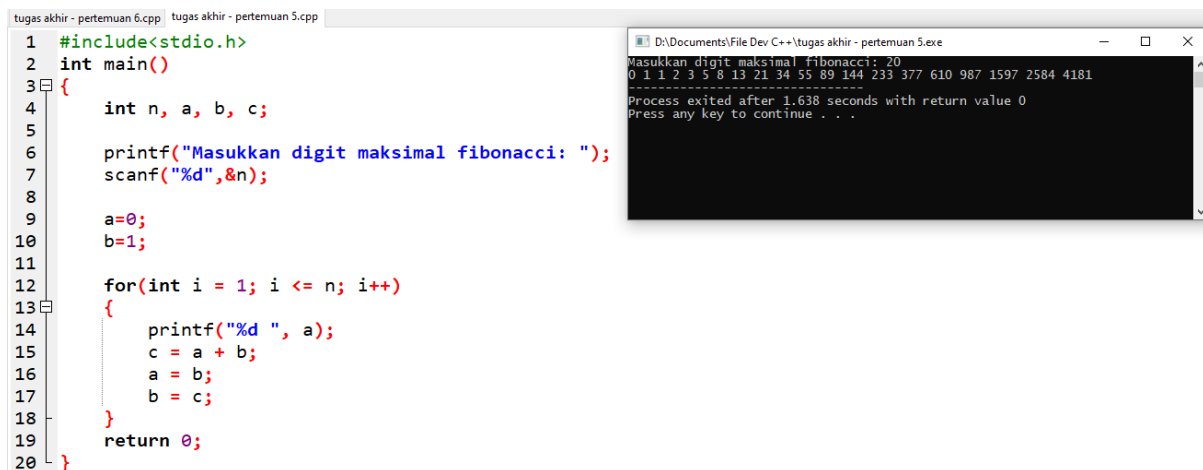
NIM : 201011402125

Kelas : 02TPLP023 – Pagi

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PAMULANG**

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566  
Tangerang Selatan - Banten

## Tugas Akhir – Pertemuan 5



The screenshot shows a C++ IDE with two windows. The left window, titled 'tugas akhir - pertemuan 5.cpp', contains the following code:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n, a, b, c;
5
6     printf("Masukkan digit maksimal fibonacci: ");
7     scanf("%d",&n);
8
9     a=0;
10    b=1;
11
12    for(int i = 1; i <= n; i++)
13    {
14        printf("%d ", a);
15        c = a + b;
16        a = b;
17        b = c;
18    }
19    return 0;
20 }
```

The right window, titled 'D:\Documents\File Dev C++\tugas akhir - pertemuan 5.exe', shows the program's execution output:

```
Masukkan digit maksimal fibonacci: 20
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
Process exited after 1.638 seconds with return value 0
Press any key to continue . . .
```

### Source Code:

```
#include<stdio.h>

int main()
{
    int n, a, b, c;

    printf("Masukkan digit maksimal fibonacci: ");
    scanf("%d",&n);

    a=0;
    b=1;

    for(int i = 1; i <= n; i++)
    {
        printf("%d ", a);
        c = a + b;
        a = b;
        b = c;
    }
    return 0;
}
```

## Kesimpulan

Pada pertemuan kali ini saya dapat menarik kesimpulan, saya memahami tentang *looping* menggunakan function for. fungsi looping berguna untuk melakukan hal yang sama (menginput data, menampilkan data, memanipulasi data) dengan sebuah fungsi looping dapat mencegah redudansi dan membuat program lebih dinamis serta mencegah pemborosan sintaks.

Bentuk umum for:

```
for (pernyataan 1; pernyataan 2; pernyataan 3) {  
    // hal yang ingin melakukan perulangan  
}
```

pernyataan 1 = untuk menginisialisasi variabel dan menentukan nilai awal

pernyataan 2 = untuk melakukan pengkodisian loop tetap dilakukan atau berhenti

pernyataan 3 = untuk menentukan increment atau decrement pada variabel di pernyataan 1

# **LAPORAN AWAL**

## **ALGORITMA DAN PEMROGRAMAN**

### **LAPORAN KE-6**



### **Disusun Oleh:**

Nama: Andri Firman Saputra

NIM : 201011402125

Kelas : 02TPLP023 – Pagi

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PAMULANG**

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566  
Tangerang Selatan - Banten

Nama : Andri Firman Saputra

NIM : 201011402125

Praktikum Algoritma

Laporan Awal-pertemuan 6

### Teori Dasar

#### a) Do... While

Perintah ini menyatakan Pengulangan Proses selama kondisi tertentu. Menekankan do terlebih dahulu, kemudian melakukan Pengkondisian pada While.

Bentuk Umum do While:

do

    Pernyataan;

while (kondisi);

atau

do

{

    Pernyataan;

    Pernyataan;

}

while (kondisi);

#### b) While

Perintah While ini prinsipnya sama dgn perintah do...while hanya Pengkondisian kondisinya terletak pada awal. Langsung melakukan Pengkondisian.

Bentuk Umum While:

while (kondisi)

    Pernyataan;

atau

while (kondisi)

{

    Pernyataan;

    Pernyataan;

}

Pernyataan dapat berupa pernyataan tunggal atau beberapa pernyataan yg dibatasi dgn tanda {}. Pernyataan ini dijalankan bila kondisinya benar.



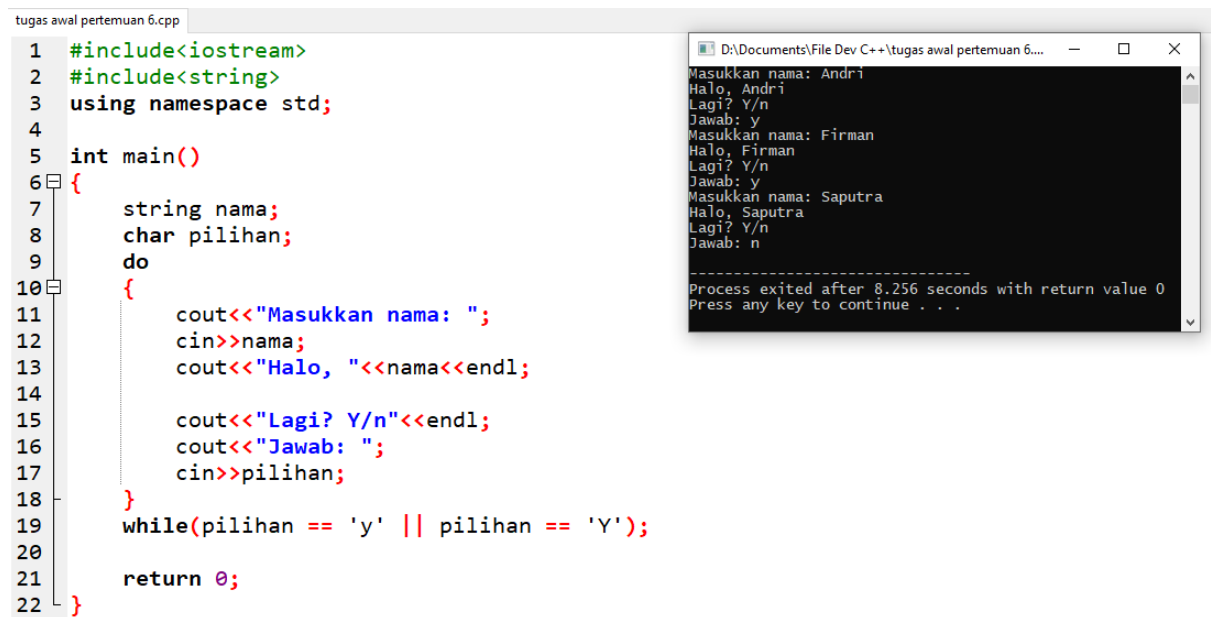
### Tugas Pendahuluan

1. Mengapa anda sebagai Programmer harus menggunakan perintah while atau do... while dalam program?  
Karena Untuk melakukan perintah pengulangan tidak perlu membuat kode secara berulang-ulang, cukup dgn menggunakan perintah while atau do while maka program akan mengulang sesuai dgn perintah.
2. Jelaskan kelebihan dan kekurangan perintah while dan do...while!  
Kelebihan: dapat mengulang kode program tanpa harus menyetik ulang.  
Kekurangan: tidak bisa melakukan kode program yg berbeda
3. Buatlah contoh algoritma dan Program sederhana menggunakan perintah do... while dgn memauai Flowchart!
4. Buatlah contoh algoritma dan program sederhana menggunakan perintah while dgn memauai flowchart!

### 3. Algoritma

1. Masukkan nama
2. Tampilkan nama
3. Masukkan pilihan
4. Jika 'y', maka masukkan nama Kembali
5. Jika 'n', keluar program

### Contoh program



The image shows a C++ program in a code editor and its execution output in a terminal window. The code is a simple loop that asks for a name and a choice to repeat. The output shows the program running with inputs 'Andri', 'y', 'Firman', 'y', 'Saputra', and 'n', resulting in the program exiting after 8.256 seconds.

```
tugas awal pertemuan 6.cpp
1 #include<iostream>
2 #include<string>
3 using namespace std;
4
5 int main()
6 {
7     string nama;
8     char pilihan;
9     do
10    {
11        cout<<"Masukkan nama: ";
12        cin>>nama;
13        cout<<"Halo, "<<nama<<endl;
14
15        cout<<"Lagi? Y/n"<<endl;
16        cout<<"Jawab: ";
17        cin>>pilihan;
18    }
19    while(pilihan == 'y' || pilihan == 'Y');
20
21    return 0;
22 }
```

```
D:\Documents\File Dev C++\tugas awal pertemuan 6...
Masukkan nama: Andri
Halo, Andri
Lagi? Y/n
Jawab: y
Masukkan nama: Firman
Halo, Firman
Lagi? Y/n
Jawab: y
Masukkan nama: Saputra
Halo, Saputra
Lagi? Y/n
Jawab: n
-----
Process exited after 8.256 seconds with return value 0
Press any key to continue . . .
```

### Source Code:

```
#include<iostream>

#include<string>

using namespace std;

int main()

{

    string nama;

    char pilihan;
```

```
do
{
    cout<<"Masukkan nama: ";

    cin>>nama;

    cout<<"Halo, "<<nama<<endl;


    cout<<"Lagi? Y/n"<<endl;

    cout<<"Jawab: ";

    cin>>pilihan;

}

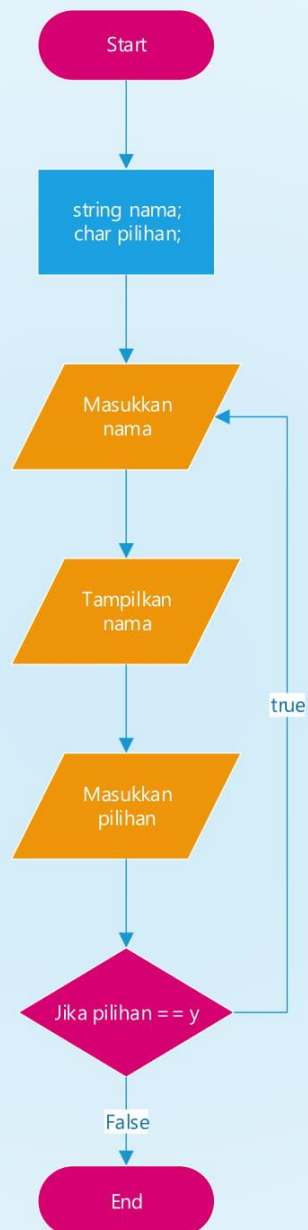
while(pilihan == 'y' || pilihan == 'Y');

return 0;

}
```



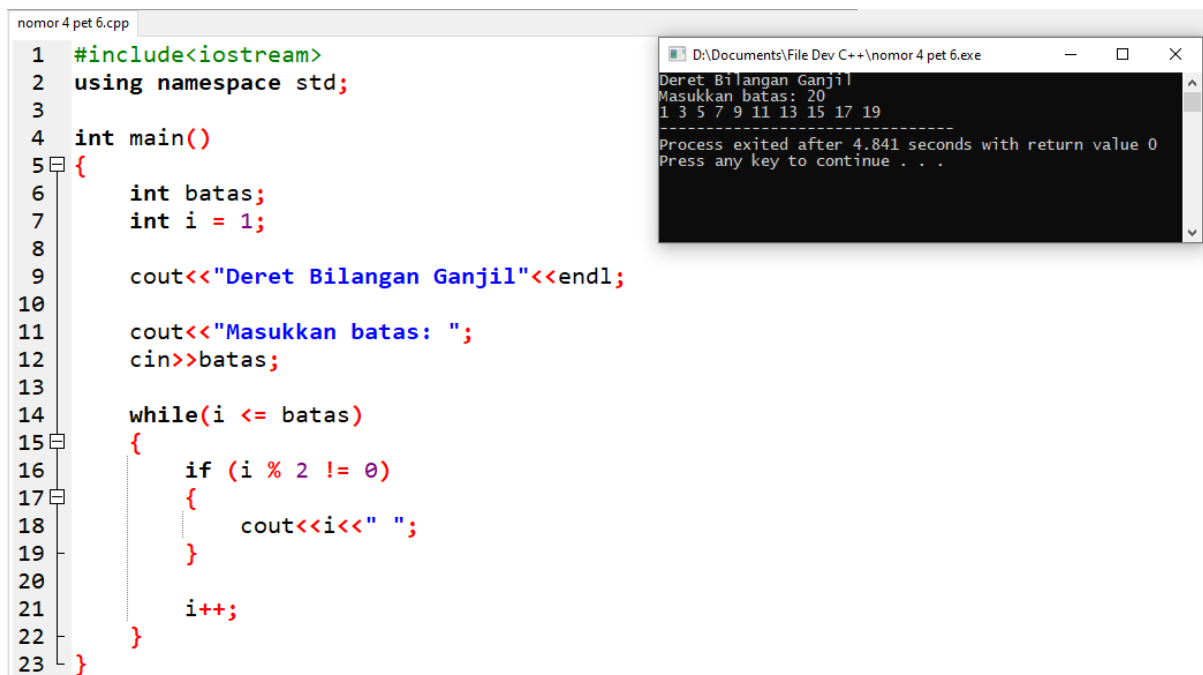
## Flowchart



## 4. Algoritma

1. Masukkan batas maksimal nilai deret ganjil
2. Melakukan perulangan while dengan batas
3. Melakukan pengkondisian ganjil
4. Tampilkan deret ganjil

## Contoh Program



The screenshot shows a C++ IDE with a file named 'nomor 4 pet 6.cpp'. The code is as follows:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int batas;
7     int i = 1;
8
9     cout<<"Deret Bilangan Ganjil"<<endl;
10
11     cout<<"Masukkan batas: ";
12     cin>>batas;
13
14     while(i <= batas)
15     {
16         if (i % 2 != 0)
17         {
18             cout<<i<<" ";
19         }
20
21         i++;
22     }
23 }
```

The execution window shows the following output:

```
Deret Bilangan Ganjil
Masukkan batas: 20
1 3 5 7 9 11 13 15 17 19
-----
Process exited after 4.841 seconds with return value 0
Press any key to continue . . .
```

## Source Code:

```
#include<iostream>

using namespace std;

int main()

{

    int batas;

    int i = 1;


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

```
cout<<"Masukkan batas: ";  
cin>>batas;  
  
while(i <= batas)  
{  
    if (i % 2 != 0)  
    {  
        cout<<i<<" ";  
    }  
  
    i++;  
}  
}
```

## Flowchart

