

**LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA**



Disusun Oleh :

NAMA : Bagas Afza Joko Ariyanto

NIM : 32601900008

**PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2019**

HALAMAN PENGESAHAN

Laporan Praktikum Algoritma dan Struktur Data

Disusun Oleh :

Bagas Afza Joko Ariyanto

(32601900008)

Telah di setujui sebagai syarat untuk memenuhi mata kuliah Praktikum Algoritma dan Struktur Data. Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Hari :

Tanggal :

Asisten Praktikum

1. Ilham Hidayat

1.....

2. Ardhi Dwi Andika

2.....

Mengetahui,

Laboran

Dosen Praktikum

Werdha Wilubertha H, S.Kom

Deddy Kurniadi, ST, M.Kom

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga laporan Algoritma dan Struktur Data dapat terselesaikan.

Tanpa lupa penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Ir. H. Prabowo Setiyawan, M.T, Ph.D yang mengijinkan penulis menimba ilmu di kampus ini
2. Dekan Fakultas Teknologi Industri Ibu Dr, Hj, Sri Arttini Dwi Prasetyowati, M,Si
3. Dosen pengampu penulis Bapak Dedy Kurniadi, ST, M.Kom yang telah memberi ilmu tentang Algoritma dan Struktur Data
4. Asisten dosen yang telah memebantu menjelaskan mengenai Praktikum Algoritma dan Struktur Data
5. Orang tua penulis yang telah mengijinkan untuk menyelesaikan laporan ini,
6. Dan kepada semua pihak yang tidak dapat saya satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan ini masih terdapat banyak kekurangan, untuk itu penulis mengharap kritik dan saran dari pembaca untuk sempurnanya laporan ini. Semoga dengan ditulisnya laporan ini dapat menjadi sumber ilmu bagi setiap pembaca.

Semarang, 24 Januari 2019

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	x
BAB I STRUKTUR DASAR ALGORITMA.....	1
1.1 Tujuan	1
1.2 Dasar Teori.....	1
1.2.1 Sekuenial.....	1
1.2.2 Seleksi.....	2
1.2.3 Pengulangan	3
1.2.4 Flowchart	5
1.3 Latihan Praktikum	5
1.4 Tugas.....	11
1.5 Kesimpulan.....	30
BAB II ARRAY 2D DAN 3D	32
2.1 Tujuan	32
2.2 Dasar Teori.....	32
2.2.1 Pengertian <i>Array</i>	32
2.2.2 <i>Array</i> Multidimensi	32
2.2.3 <i>Array</i> Yang Bersifat Konstan.....	34
2.2.4 <i>Array</i> Sebagai Tipe Data Bentukan	34

2.3	Latihan	35
2.4	Tugas.....	45
2.5	Kesimpulan.....	60
BAB III METODE <i>SORTING</i> DAN <i>SEARCHING</i>		62
3.1	Tujuan	62
3.2	Dasar Teori.....	62
3.2.1	Pencarian Data.....	62
3.2.2	Pengertian Sorting	62
3.2.3	Bubble Sort.....	63
3.3	Latihan	64
3.4	Tugas.....	70
3.5	Kesimpulan.....	99
BAB IV REKURSI.....		100
4.1	Tujuan	100
4.2	Dasar Teori.....	100
4.2.1	Pengertian Rekursi.....	100
4.2.2	Fibonacci.....	100
4.3	Latihan	102
4.4	Tugas.....	107
4.5	Kesimpulan.....	120
BAB V <i>LINKED LIST, STACK, DAN QUEUE</i>		122
5.1	Tujuan	122
5.2	Dasar Teori.....	122
5.2.1	Metode List	122
5.2.2	Metode Stack.....	123

5.2.3	LIFO.....	123
5.2.4	Metode Queue	124
5.2.5	Pengertian Linked List.....	124
5.2.6	Pengertian Stack	125
5.2.7	Pengertian Queue.....	125
5.3	Latihan	126
5.4	Tugas.....	132
5.5	Kesimpulan.....	148

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 1. 1 Sruktur Sekuential	2
Gambar 1.2 Struktur Seleksi	3
Gambar 1.3 Diagram Alir Untuk Menggambarkan Pengulangan Bentuk Pertama	4
Gambar 1.4 Diagram Alir Untuk Menggambarkan Pengulangan Bentuk Kedua ...	4
Gambar 1.5 <i>Flowchart</i> Latihan 1.....	7
Gambar 1.6 <i>Output</i> Latihan 1 Persegi Panjang	7
Gambar 1.7 <i>Output</i> Latihan 1 Lingkaran	8
Gambar 1.8 <i>Flowchart</i> Latihan 2.....	9
Gambar 1.9 <i>Output</i> Latihan 2.....	10
Gambar 1.10 <i>Flowchart</i> Tugas 1	12
Gambar 1.11 <i>Output</i> Tugas 1	12
Gambar 1.12 <i>Flowchart</i> Tugas 2.....	15
Gambar 1.13 <i>Output</i> Hitung Luas Persegi Panjang Bahasa C++.....	16
Gambar 1.14 <i>Output</i> Keliling Persegi Panjang Bahasa C++	16
Gambar 1.15 <i>Output</i> Diagonal Persegi Panjang Bahasa C++	17
Gambar 1.16 <i>Flowchart</i> Tugas 3.....	20
Gambar 1.17 <i>Output</i> Luas Persegi Panjang Bahasa C.....	21
Gambar 1.18 <i>Output</i> Keliling Persegi Panjang Bahasa C.....	21
Gambar 1.19 <i>Output</i> Diagonal Persegi Panjang Bahasa C	22
Gambar 1.20 <i>Output</i> Tugas 4	24
Gambar 1.21 <i>Output</i> Tugas 4	25
Gambar 1.22 <i>Flowchart</i> Tugas 5.....	28
Gambar 1.23 <i>Output</i> Tugas 5	29
Gambar 2. 1 Array 3 Dimensi	33
Gambar 2. 2 <i>Flowchart</i> Latihan 1	36
Gambar 2. 3 <i>Output</i> Latihan 1	36
Gambar 2. 4 <i>Flowchart</i> Latihan 2.....	40
Gambar 2. 5 <i>Output</i> Latihan 2.....	40
Gambar 2. 6 <i>Flowchat</i> Latihan 3	43

Gambar 2. 7 Output Latihan 3	44
Gambar 2. 8 Flowchart Tugas 1	47
Gambar 2. 9 Output Tugas 1	48
Gambar 2. 10 Flowchart Tugas 2	50
Gambar 2. 11 Output Tugas 2	51
Gambar 2. 12 Flowchart Tugas 3	54
Gambar 2. 13 Output Tugas 3	55
Gambar 2. 14 Flowchart Tugas 4	57
Gambar 2. 15 Output Tugas 4	58
Gambar 2. 16 Flowchart Tugas 5	59
Gambar 2. 17 Output Tugas 5	60
Gambar 3.1 Flowchart Latihan 1	66
Gambar 3.2 Output Latihan 1	67
Gambar 3.3 Flowchart Latihan 2	69
Gambar 3.4 Output Latihan 2	70
Gambar 3.5 Flowchart Tugas 1	74
Gambar 3.6 Output Tugas 1	74
Gambar 3.7 Flowchart Tugas 2	79
Gambar 3. 8 Output Tugas 2	80
Gambar 3 9 Flowchart Tugas 3	82
Gambar 3.10 Output Tugas 3	82
Gambar 3 11 Flowchart Descending Tugas 4	90
Gambar 3.12 Output Bubble Sort Tugas 4	91
Gambar 3.13 Output Selection Sort Tugas 4	91
Gambar 3.14 Output Insertion Sort Tugas 4	92
Gambar 3.15 Flowchart Ascending Tugas 4	94
Gambar 3.16 Output Ascending Tugas 4	94
Gambar 3. 17 Flowchart Tugas 5	98
Gambar 3.18 Output Tugas 5	98
Gambar 4.1 <i>Flowchart</i> Latihan 1 BAB 4	103
Gambar 4.2 <i>Output</i> Latihan 1 BAB 4	103

Gambar 4.3 <i>Flowchart</i> Latihan 2.....	106
Gambar 4.4 <i>Output</i> Latihan 2.....	106
Gambar 4.5 <i>Flowchart</i> Tugas 1.....	109
Gambar 4.6 <i>Output</i> Tugas 1	109
Gambar 4.7 <i>Flowchart</i> Tugas 2.....	111
Gambar 4.8 <i>Output</i> Tugas 2	112
Gambar 4.9 <i>Flowchart</i> Tugas 3.....	114
Gambar 4.10 <i>Output</i> Tugas 3	114
Gambar 4.11 <i>Flowchart</i> Tugas 4.....	117
Gambar 4.12 <i>Output</i> Dari Tugas 4	117
Gambar 4.13 <i>Flowchart</i> Tugas 5.....	119
Gambar 4.14 <i>Output</i> Dari Tugas 5	119
Gambar 5.1 Senarai.....	122
Gambar 5.2 Senarai Tunggal.....	122
Gambar 5.3 Senarai Ganda.....	123
Gambar 5.4 Senarai Sirkular	123
Gambar 5.5 <i>Flowchart</i> Tugas 1	138
Gambar 5.6 <i>Output</i> Tugas 1	138
Gambar 5.7 <i>Flowchart</i> Tugas 2	142
Gambar 5.8 <i>Output</i> Tugas 2	143
Gambar 5.9 <i>Flowchart</i> Tugas 3	147
Gambar 5.10 <i>Output</i> Tugas 3	148

DAFTAR TABEL

Tabel 2. 1 <i>Array</i> 2 Dimensi.....	33
--	----

BAB I

STRUKTUR DASAR ALGORITMA

1.1 Tujuan

Adapun tujuan dari praktikum ini:

1. Mahasiswa dapat mengetahui dan memahami macam struktur dasar algoritma
2. Mahasiswa dapat mengetahui dan memahami struktur sekuensial
3. Mahasiswa dapat mengetahui dan memahami struktur seleksi
4. Mahasiswa dapat mengetahui dan memahami struktur pengulangan

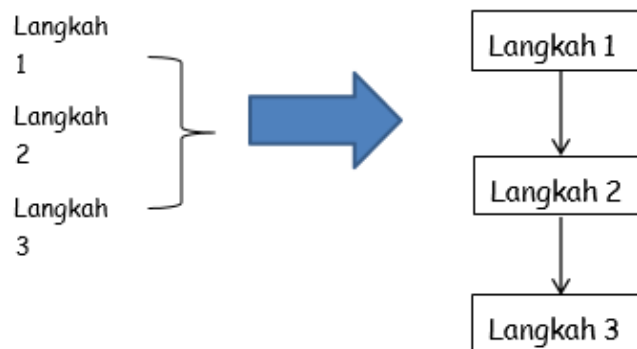
1.2 Dasar Teori

1.2.1 Sekuensial

Suatu pemrograman tidaklah terlepas dari algoritma sebagai acuan untuk membuat program. Pemrograman sekuensial merupakan runtutan (sekuensial / *sequence*) satu atau lebih intruksi, yang berarti bahwa:

- Tiap intruksi di kerjakan satu persatu.
- Tiap intruksi dilaksanakan tepat satu kali; tidak ada intruksi yang diulang.
- Urutan intruksi yang dilaksanakan memproses sama dengan urutan intruksi sebagaimana yang tertulis di dalam teks algoritmanya.
- Akhir dari intruksi terakhir merupakan akhir algoritma.

Pada struktur sekuensial, langkah-langkah yang dilakukan dalam algoritma diproses secara berurutan, sebagaimana diperlihatkan pada gambar berikut:



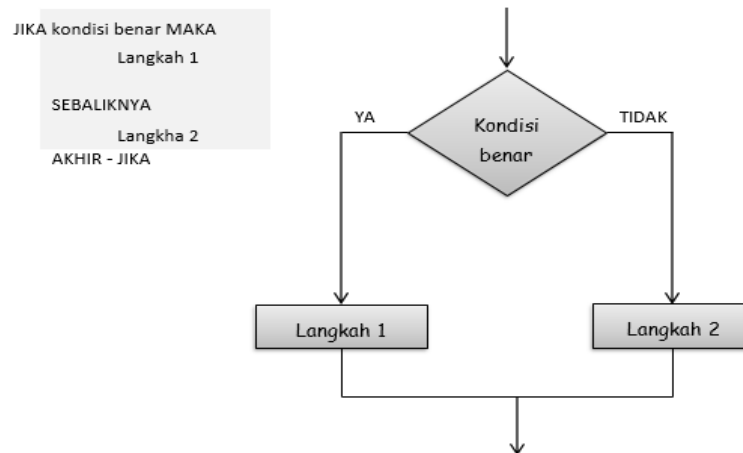
Gambar 1. 1 Struktur Sekuential

Pada contoh tersebut, langkah 1 merupakan langkah yang akan dijalankan pertama kali. Setelah itu, langkah 2 dikerjakan dan diikuti langkah 3.

1.2.2 Seleksi

Seleksi digunakan untuk mengarahkan percabangan aliran eksekusi program. Eksekusi program diarahkan sesuai kondisi yang mengendalikannya. Aliran seleksi dalam C++ dinyatakan dalam bentuk `if` dan `switch`. Aliran ini digunakan untuk menentukan alur jalannya eksekusi program. Alur eksekusi program ditentukan dari nilai kebenaran kondisi yang ditentukan.

Struktur seleksi menyatakan pemilihan langkah yang didasarkan oleh suatu kondisi (pengambilan keputusan). Gambar 2.2 memperlihatkan diagram alur struktur seleksi yang melibatkan dua alternatif. Dalam hal ini, simbol belah ketupat digunakan untuk mewakili langkah pengambilan keputusan.



Gambar 1.2 Struktur Seleksi

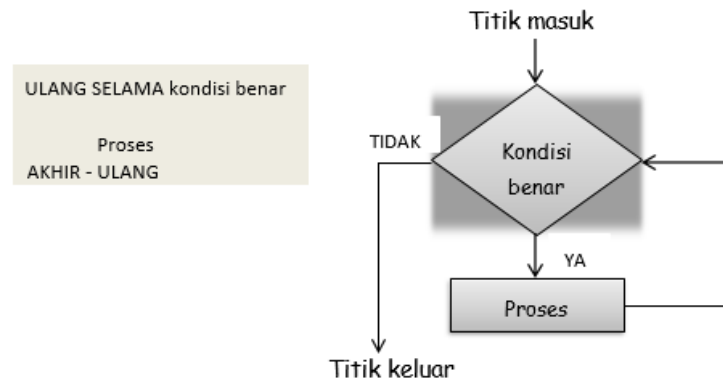
Pada struktur tersebut, langkah 1 hanya akan dijalankan kalau kondisi bernilai benar, sedangkan langkah 2 hanya akan dijalankan kalau kondisi bernilai salah.

1.2.3 Pengulangan

Pengulangan adalah suatu tindakan melakukan hal yang sama berulang-ulang kali. Penting untuk dilakukan pengulangan atau perulangan dalam pembuatan suatu program untuk dengan tujuan untuk mengefisiensikan kerja program ataupun mengefisiensikan listing dari coding dari program tersebut, karena programmer tidak perlu mengulang proses yang sama. Untuk perulangan pada C++ dapat dilakukan dengan menggunakan pernyataan `for`, `while`, `do-while`.

Pengulangan menyatakan suatu tindakan atau langkah yang dijalankan beberapa kali. Struktur pengulangan menyatakan perwujudan keadaan seperti itu. Sebagai contoh, jika anda ingin menampilkan 10 tulisan “SELAMAT BELAJAR”, Anda bisa menuliskannya dengan struktur sekuensial. Hal itu berarti Anda memberikan 10 intruksi untuk menuliskan kesepuluh tulisan tersebut. Cara seperti itu memang praktis untuk jumlah pengulangan yang sedikit (misalnya 2 atau 3 pengulangan), tetapi tidak cocok untuk jumlah yang besar. Agar lebih praktis, Anda bisa menggunakan struktur pengulangan.

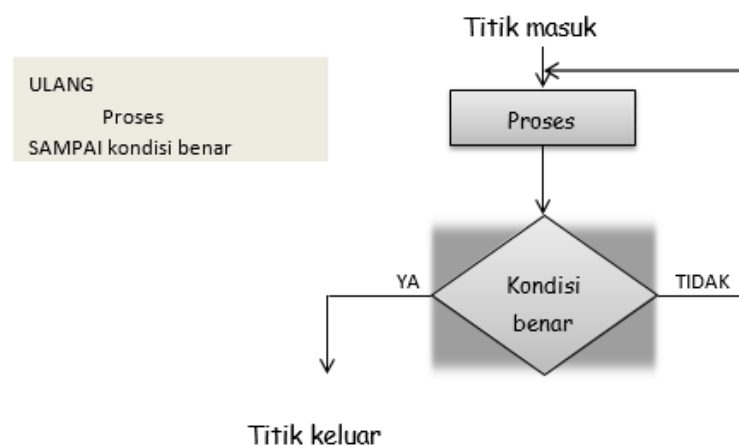
Struktur pengulangan pertama



Gambar 1.3 Diagram Alir Untuk Menggambarkan Pengulangan Bentuk Pertama

Pada struktur tersebut, proses dapat berupa satu atau beberapa langkah. Pada bentuk ini, ada kemungkinan proses tidak dijalankan sama sekali sekiranya di awal kondisi bernilai salah. Diagram alir pada Gambar 2.3 menunjukkan bahwa sebelum proses dijalankan pertama kali, kondisi diuji terlebih dahulu. Sekiranya kondisi bernilai benar maka proses dijalankan, kemudian kondisi diuji lagi. Sepanjang kondisi masih bernilai benar proses masih tetap dijalankan. Namun, begitu kondisi bernilai salah maka pengulangan berakhir.

Struktur pengulangan kedua







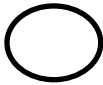
Gambar 1.4 Diagram Alir Untuk Menggambarkan Pengulangan Bentuk Kedua

Pada bentuk kedua, proses paling tidak dijalankan sekali.

1.2.4 Flowchart

Flowchart adalah urutan proses kegiatan yang digambarkan dalam bentuk simbol. *Flowchart* juga didefinisikan sebagai diagram yang menyatakan aliran proses dengan menggunakan anotasianotasi.

Beberapa simbol yang digunakan didalam *flowchart* adalah sebagai berikut:

1.  (**Terminator**) menyatakan titik awal atau titik akhir diagram alir.
2.  (**Proses**) menyatakan sebarang proses, misal untuk menyatakan suatu operasi aritmatika.
3.  (**Input/output** atau kadang disebut data) digunakan untuk menyatakan operasi pemasukan data atau penampilann data.
4.  (**Keputusan**) dilakukan untuk melakukan pengambilan keputusan. Dalam hal ini, yang ada dalam simbol ini berupa suatu pertanyaan yang jawabannya berupa dua kemungkinan, yaitu YA atau TIDAK.
5.  (**Konektor**) digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir. (Kadir, 2010)

1.3 Latihan Praktikum

1. Latihan 1 (Seleksi)

a. Kode Program

```
#include<iostream>

using namespace std;

main() {
```

```
char jenis;

int panjang, lebar, luas, jari;

float luaslingkaran;

cout<< "Pilih Persegi Panjang<P> atau Lingkaran<L> :
";

cin>> jenis;

if(jenis == 'P' || jenis == 'p'){

    cout<< "panjang = ";

    cin>> panjang;

    cout<< "lebar = ";

    cin>> lebar;

    luas = panjang * lebar;

    cout<< "Luas = " << luas << endl;

}else if(jenis == 'L' || jenis == 'l'){

    cout<< "jari-jari = ";

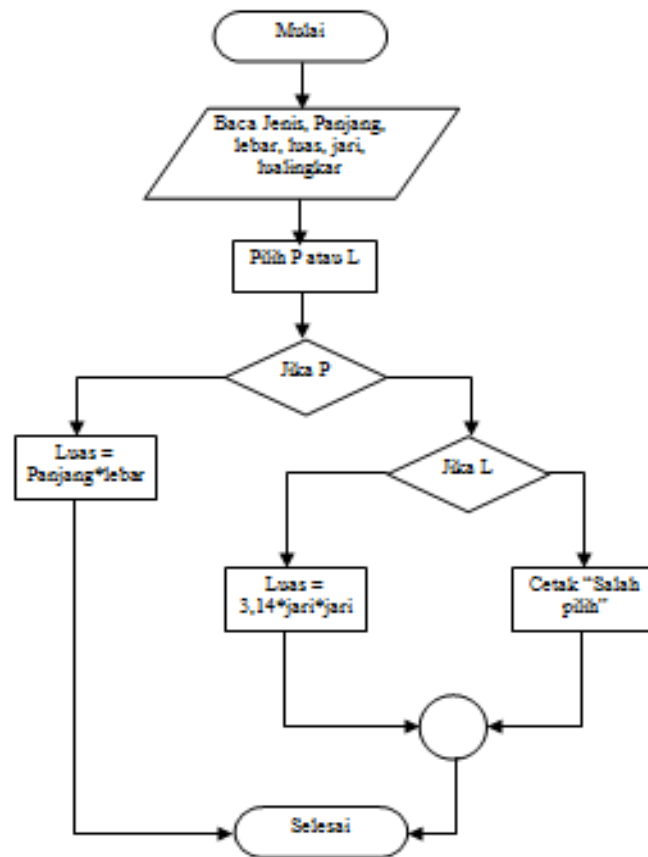
    cin>> jari;

    luaslingkaran = 3.14 * jari * jari;

    cout<< "Luas = " << luaslingkaran << endl;

}else

cout << "Salah pilih" << endl;}
```


b. *Flowchart*Gambar 1.5 *Flowchart* Latihan 1c. *Output*

```

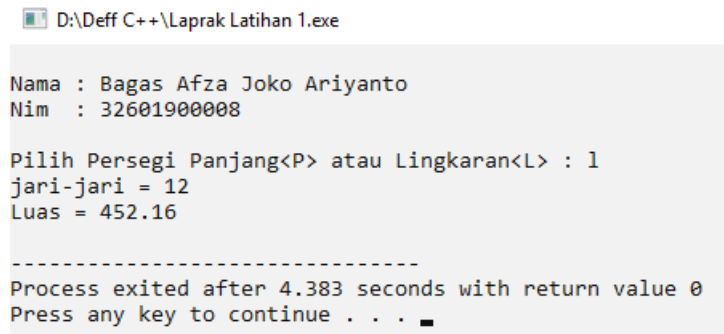
D:\Deff C++\Laprak Latihan 1.exe

Nama : Bagas Afza Joko Ariyanto
Nim : 32601900008

Pilih Persegi Panjang<P> atau Lingkaran<L> : p
panjang = 20
lebar = 15
Luas = 300

-----
Process exited after 58.36 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 1.6 *Output* Latihan 1 Persegi Panjang



```

D:\Deff C++\Laparak Latihan 1.exe

Nama : Bagas Afza Joko Ariyanto
Nim  : 32601900008

Pilih Persegi Panjang<P> atau Lingkaran<L> : l
jari-jari = 12
Luas = 452.16

-----
Process exited after 4.383 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.7 *Output* Latihan 1 Lingkaran

d. Penjelasan

Gambar 3.2 dan gambar 3.3 menunjukkan contoh percabangan *braching* (if) antara pilihan “P” untuk perhitungan luas persegi dan “L” untuk perhitungan luas lingkaran. Dalam coding perhitungan tersebut menggunakan deklarasi variable dengan tipe variable integer (int), float dan karakter (char).

Dalam coding di tiper char yang di masukan adalah bertipe pilihan dimana jika kita ingin menghitung luas persegi kita masukan perintah “P”. Sehingga output yang akan keluar adalah perhitungan persegi dengan integer panjang dan lebar.

Namun, apabila kita ingin menghitung luas lingkaran masukan perintah “L”. Sehingga *output* yang keluar adalah perhitungan luas lingkaran dengan integer jari-jari.

2. Latihan 2 (Perulangan)

a. Kode Program

```

#include<iostream>
using namespace std;

main() {
    int l=0;
    for(int j=0; j<=10; j++){
        for( int k=j; k<=l; k++){
            cout<<k<<' ';

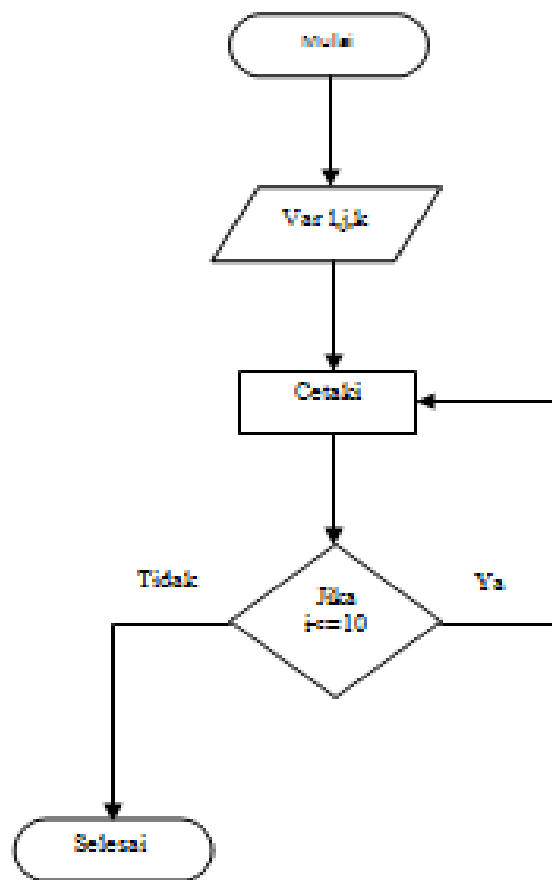
```

```

    }
    l+=2;
    cout<<"\n";
}
}

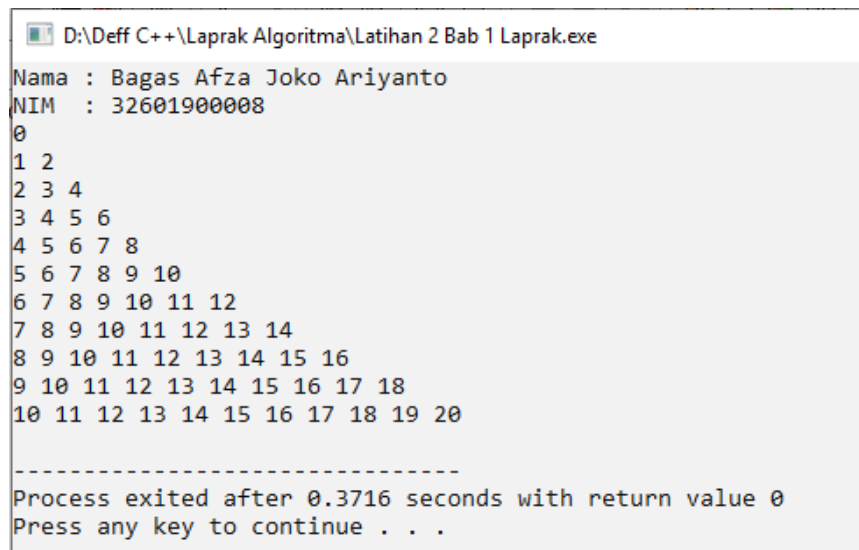
```

b. *Flowchart*



Gambar 1.8 *Flowchart* Latihan 2

c. Output



```

D:\Deff C++\Laprak Algoritma\Latihan 2 Bab 1 Laprak.exe
Nama : Bagas Afza Joko Ariyanto
NIM  : 32601900008
0
1 2
2 3 4
3 4 5 6
4 5 6 7 8
5 6 7 8 9 10
6 7 8 9 10 11 12
7 8 9 10 11 12 13 14
8 9 10 11 12 13 14 15 16
9 10 11 12 13 14 15 16 17 18
10 11 12 13 14 15 16 17 18 19 20

-----
Process exited after 0.3716 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.9 Output Latihan 2

d. Penjelasan

Pada gambar 1.9 merupakan hasil dari perulangan atau *looping*. Kode untuk melakukan *looping* ada *for* dan *while*. Diatas merupakan contoh *looping* dengan *for*. Bisa kita lihat disini dalam membentuk perulangan *for* berlaku komposisi “*for* (inisialisasi; syarat pengulangan; pengubah nilai pecacahan)”.

Inisialisasi adalah bagian untuk memberikan nilai awal untuk variabel-variabel tertentu. Contohnya ketika kita ingin memulai perhitungan dari angka 9 maka pada bagian inisialisasi dimasukan angka 9 (variabel=9).

Kemudian ada syarat pengulangan. Syarat pengulangan adalah sampai mana kita ingin melakukan *looping* atau bisa di artikan angka terakhir yang kita inginkan itu apa. Contoh ketika kita ingin melakukan *looping* sampai angka 10 kita masukan tanda lebih dari sama dengan ≤ 10 (variabel ≤ 10).

Dan yang terakhir adalah pengubah nilai pecacahan. Pengubah nilai pecacahan adalah mengatur hasil *looping* itu naik atau turun. Contoh kita ingin melakukan *looping* dari terkecil ke terbesar kita masukan

variabel++ misal kita ingin kelipatan 2 maka masukan variabel+=2.
Sedangkan jika ingin dari terbesar ke terkecil masukan variable—jika
kita ingin berkelipatan 3 maka masukan variabel-=3.

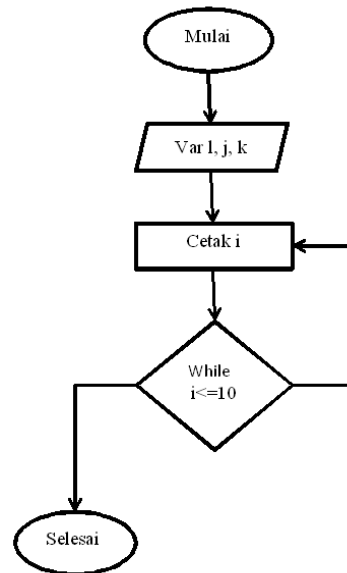
1.4 Tugas

1. Modifikasilah program perulangan pada contoh 2 dengan struktur while menggunakan bahasa C.

Jawaban :

a. Kode Program

```
#include <stdio.h>
int main(){
    int l=0;
    int j=0;
    printf("Nama : Bagas Afza Joko Ariyanto");
    printf("\nNIM   : 32601900008");
    printf("\n");
    while (j<=10){
        int k=j;
        while (k<=1){
            printf("%d ",k);
            k++;
        }
        j++;
        l+=2;
        printf("\n");
    }
}
```

b. *Flowchart*Gambar 1.10 *Flowchart* Tugas 1c. *Output*

```

D:\Deff C++\Laprak Algoritma\Tugas 1 Bahasa C BAB 1 L...
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008
0
1 2
2 3 4
3 4 5 6
4 5 6 7 8
5 6 7 8 9 10
6 7 8 9 10 11 12
7 8 9 10 11 12 13 14
8 9 10 11 12 13 14 15 16
9 10 11 12 13 14 15 16 17 18
10 11 12 13 14 15 16 17 18 19 20

-----
Process exited after 0.1676 seconds with return value 10
Press any key to continue . . .
  
```

The screenshot shows a Windows command prompt window with a title bar. The output displays a pattern of numbers from 0 to 20 arranged in rows of increasing length (1 to 10 numbers per row). Below the pattern, it shows the process exit information and a prompt to press any key to continue.

Gambar 1.11 *Output* Tugas 1d. *Penjelasan*

Dari *output*, dan *flowchart* memang sama dengan contoh soal di latihan 2. Namun, pada kali ini saya menggunakan bahasa C dimana kode programnya pun akan sedikit berbeda dengan kode program di latihan 2. Dalam bahasa C *header* yang kita pakai adalah `#include<stdio.h>`. *Stdio.h* adalah salah satu *header* dari bahasa C.

Fungsi dari *header stdio.h* untuk membaca *input* dan *output* dari kode program bahasa C.

Di bahasa C kode *input* dan *output*nya berbeda dengan bahasa C++ *input* yang di pakai untuk bahasa C adalah *printf* sedangkan untuk *output* dari bahasa C adalah *scanf*.

Dalam bahasa C++ setelah kita menuliskan *output (cout)* kita masukan kode << untuk kode *output* dan untuk kode *input (cin)* kita masukan kode >> itu berbeda dengan di bahasa C. Di dalam bahasa C setelah kita memasukan kode *output (printf)* kita masukan kode () dan untuk kode *input (scanf)* kita masukan kode yang sama juga dengan kode *output* yaitu (). Kemudian dalam bahasa C++ subjek kita masukan setelah kode *input (>>)* atau kode *output (<<)*, berbeda dengan bahasa C dalam bahasa C subjek kita masukan di dalam kode *input ()* atau kode *output ()*. Dan pastinya menggunakan tanda petik dua di awal dan penutupnya (“”).

2. Buatlah program yang dapat menghitung luas, keliling, dan diagonal persegi panjang. Diagonal dihitung dengan rumus:

Diagonal = akar($\text{panjang}^2 + \text{lebar}^2$). Akar dapat dikerjakan dengan fungsi *sqr()*. Masukan berupa panjang dan lebar. Program berupa pilihan untuk luas, keliling, dan diagonal.

Jawaban :

a. Kode *Program*

```
#include<iostream>
#include<math.h>
using namespace std;

main() {
    char jenis;
    int panjang, lebar, luas, keliling;
    float diagonal;
    cout<< "\nNama : Bagas Afza Joko Ariyanto";
    cout<< "\nNIM : 32601900008";
    cout<< "\n\nKode Hitung Luas Persegi Panjang = L";
```

```

cout<< "\n\nKode Hitung Keliling Persegi Panjang = K";
cout<< "\n\nKode Hitung Diagonal Persegi Panjang = D";
cout<< "\n\nMasukan Kode Perhitungan = ";
cin>> jenis;
cout<<
"=====";

if(jenis == 'L' || jenis == 'l' ){
    cout<< "\n>>>>>Penghitungan Luas Persegi
Panjang<<<<<";
    cout<< "\n\nMasukan Panjang = ";
    cin>> panjang;
    cout<< "\nMasukan Lebar = ";
    cin>> lebar;

    luas = panjang * lebar;
    cout<< "=====";
    cout<< "\nLuas Persegi Panjang Adalah = "<<luas;
}else if(jenis == 'K' || jenis == 'k' ){
    cout<< "\n>>>>>Penghitungan Keliling Persegi
Panjang<<<<<";
    cout<< "\n\nMasukan Panjang = ";
    cin>> panjang;
    cout<< "\nMasukan Lebar = ";
    cin>> lebar;

    keliling = 2 * (panjang + lebar);
    cout<< "\n=====";
    cout<< "\nKeliling Persegi Panjang =
"<<keliling;
}else if(jenis == 'D' || jenis == 'd' ){
    cout<< "\n>>>>>Penghitungan Diagonal Persegi
Panjang<<<<<";
    cout<< "\n\nMasukan Panjang = ";
    cin>> panjang;
    cout<< "\nMasukan Lebar = ";
    cin>> lebar;

```

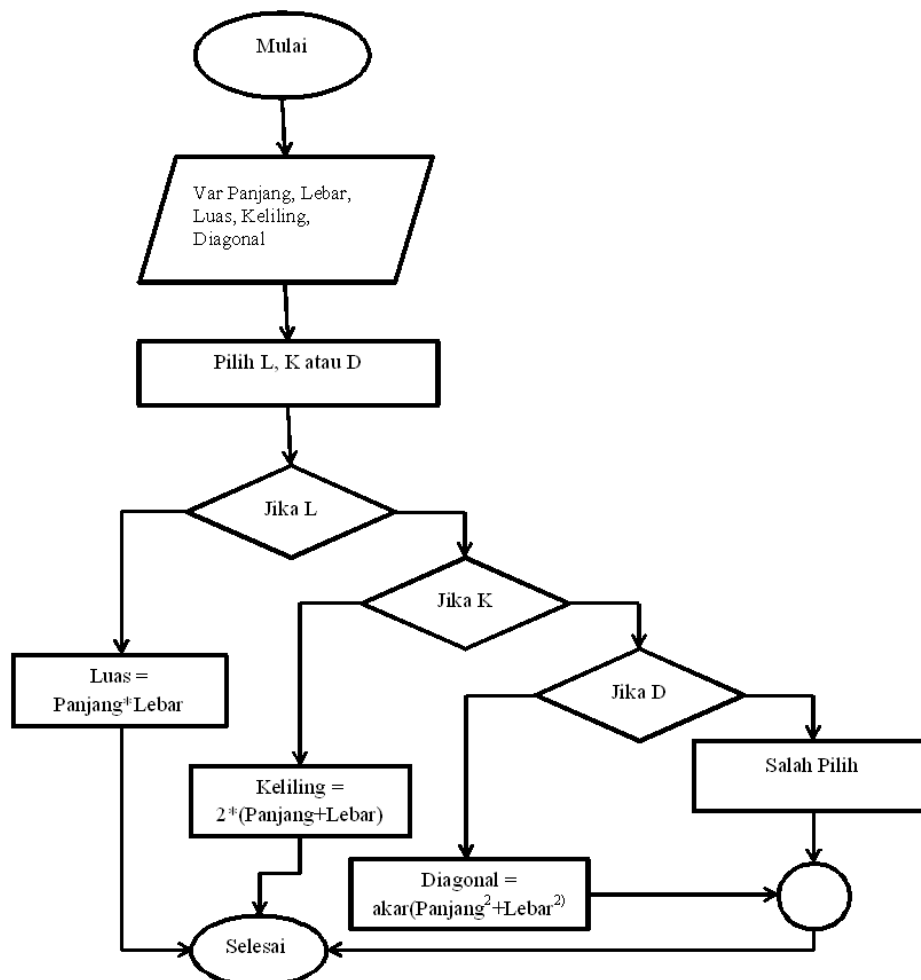


```

        diagonal=sqrt (panjang*panjang + lebar*lebar);
        cout<< "\n===== ";
        cout<< "\nDiagonal Persegi Panjang = "<<
        diagonal;
    }else
        cout<< "\nKode Tidak Valid";
}

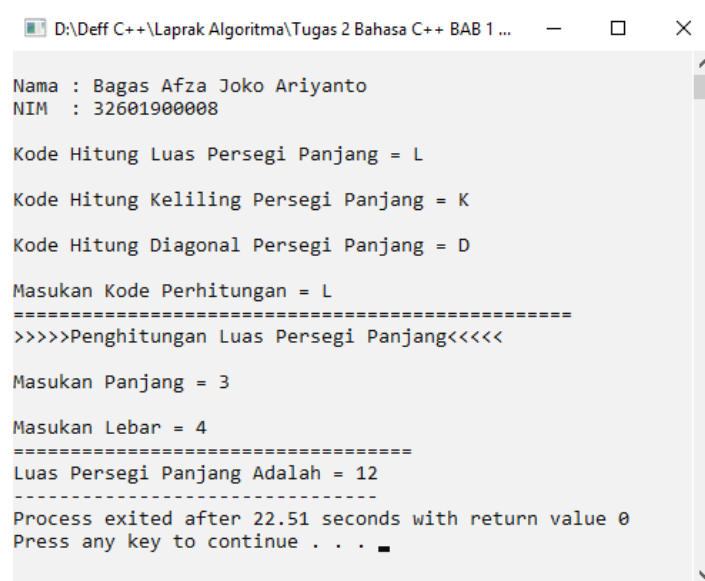
```

b. *Flowchart*



Gambar 1.12 *Flowchart* Tugas 2

c. *Output*



```

D:\Deff C++\Laprak Algoritma\Tugas 2 Bahasa C++ BAB 1 ...
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008

Kode Hitung Luas Persegi Panjang = L
Kode Hitung Keliling Persegi Panjang = K
Kode Hitung Diagonal Persegi Panjang = D

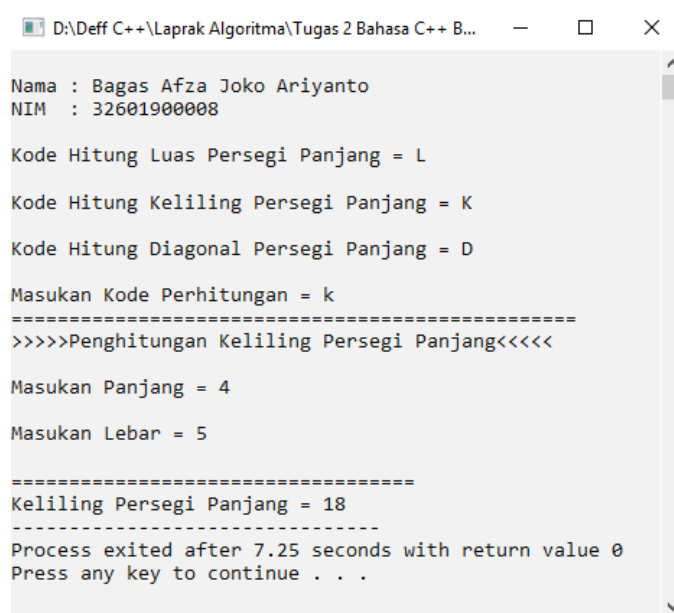
Masukan Kode Perhitungan = L
=====
>>>>Penghitungan Luas Persegi Panjang<<<<

Masukan Panjang = 3

Masukan Lebar = 4
=====
Luas Persegi Panjang Adalah = 12
-----
Process exited after 22.51 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.13 *Output* Hitung Luas Persegi Panjang Bahasa C++



```

D:\Deff C++\Laprak Algoritma\Tugas 2 Bahasa C++ B...
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008

Kode Hitung Luas Persegi Panjang = L
Kode Hitung Keliling Persegi Panjang = K
Kode Hitung Diagonal Persegi Panjang = D

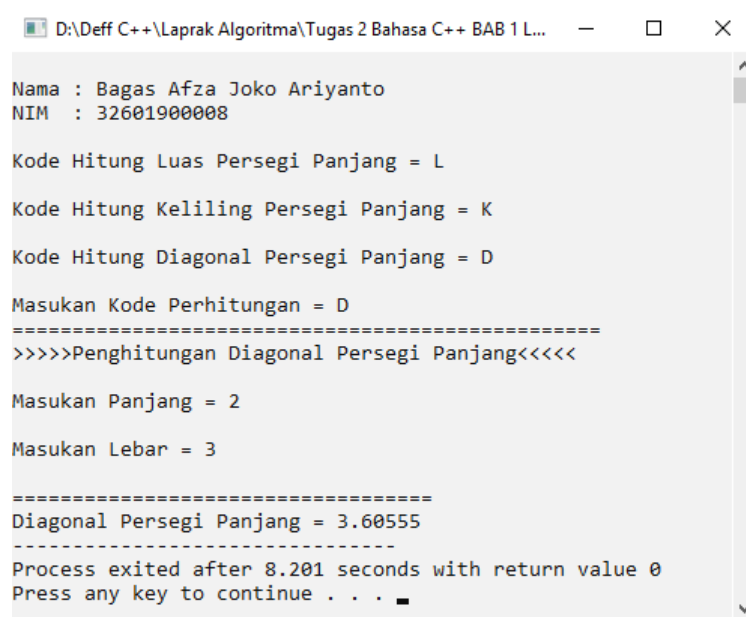
Masukan Kode Perhitungan = k
=====
>>>>Penghitungan Keliling Persegi Panjang<<<<

Masukan Panjang = 4

Masukan Lebar = 5
=====
Keliling Persegi Panjang = 18
-----
Process exited after 7.25 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.14 *Output* Keliling Persegi Panjang Bahasa C++



```

D:\Deff C++\Laprak Algoritma\Tugas 2 Bahasa C++ BAB 1 L...
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Kode Hitung Luas Persegi Panjang = L
Kode Hitung Keliling Persegi Panjang = K
Kode Hitung Diagonal Persegi Panjang = D
Masukan Kode Perhitungan = D
=====
>>>>Penghitungan Diagonal Persegi Panjang<<<<
=====
Masukan Panjang = 2
Masukan Lebar = 3

=====
Diagonal Persegi Panjang = 3.60555
=====
Process exited after 8.201 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.15 Output Diagonal Persegi Panjang Bahasa C++

d. Penjelasan

Mengenai kode program ada tipe data berjenis *char*, *int*, dan *float*. Tipe data *char* digunakan untuk membuat program pilihan dalam program kali ini saya menggunakan tipe data *char* karena saya akan membuat pilihan penghitungan dari luas persegi panjang, keliling persegi panjang dan juga diagonal persegi panjang maka dari itu saya menggunakan tipe data *char*.

Tidak hanya menggunakan tipe data *char* saya juga menggunakan tipe data *int* atau *integer*. Tipe data *integer* digunakan untuk hasil *output* tidak akan desimal atau bilangan bulat. Karena saya akan merencanakan untuk memberikan *input* bilangan bulat saya menggunakan tipe data *integer*.

Dan tipe data yang saya pakai terakhir adalah tipe data *float*. Tipe data *float* digunakan apabila ingin menghitung atau mengeluarkan *output* bilangan desimal atau bilangan koma. Dalam menghitung diagonal menggunakan rumus akar dan biasanya hasil akar akan berakhir bilangan koma maka dari itu saya memakai tipe data *float*.

Dalam membuat program pilihan kita harus menggunakan percabangan karena percabanganlah yang akan menyeleksi pilihan kita.

Bila anda ingin membuat pilihan yang lebih dari satu bisa tiga atau dua gunakanlah seleksi majemuk atau bisa di bilang fungsi dalam fungsi atau dalam *if* ada *if*.

3. Modifikasilah program nomor 2 menjadi bahasa C.

Jawaban :

a. Kode Program

```
#include<stdio.h>
#include<math.h>

main() {
    char jenis;
    int panjang, lebar, luas, keliling;
    float diagonal;
    printf("\nNama : Bagas Afza Joko Ariyanto");
    printf("\nNIM : 32601900008");
    printf("\n\nKode Hitung Luas Persegi Panjang = L");
    printf("\n\nKode Hitung Keliling Persegi Panjang = K");
    printf("\n\nKode Hitung Diagonal Persegi Panjang = D");
    printf("%s \n\nMasukan Kode Perhitungan = ");
    scanf("%s", &jenis);
    printf("\n=====
=====");

    if(jenis == 'L' || jenis == 'l' ){
        printf("\n>>>>Penghitungan Luas Persegi
Panjang<<<<");
        printf("\n\nMasukan Panjang = ");
        scanf("%i", &panjang);
        printf("\nMasukan Lebar = ");
        scanf("%i", &lebar);
```

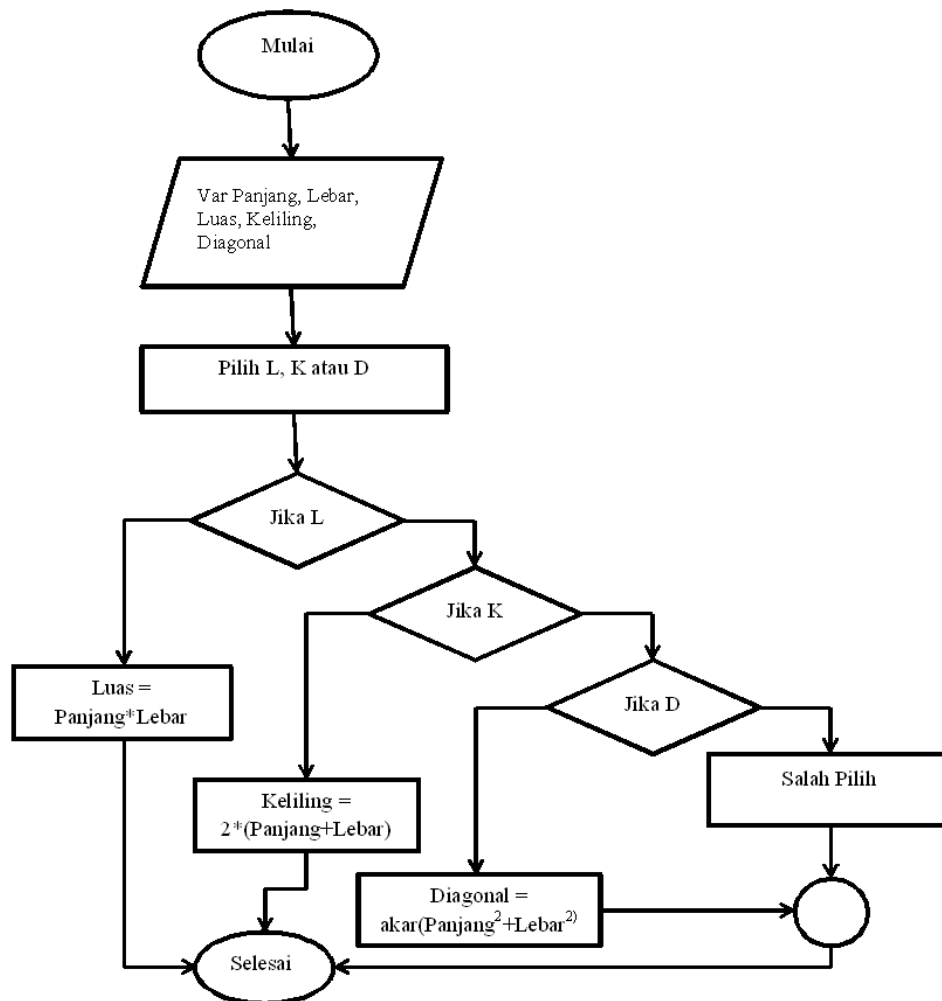
```

        luas = panjang * lebar;
        printf("=====");
        printf("\nLuas Persegi Panjang Adalah =
%d",luas);
    }else if(jenis == 'K' || jenis == 'k' ){
        printf("\n>>>>Penghitungan Keliling Persegi
Panjang<<<<");
        printf("\n\nMasukan Panjang = ");
        scanf("%i", &panjang);
        printf("\nMasukan Lebar = ");
        scanf("%i", &lebar);

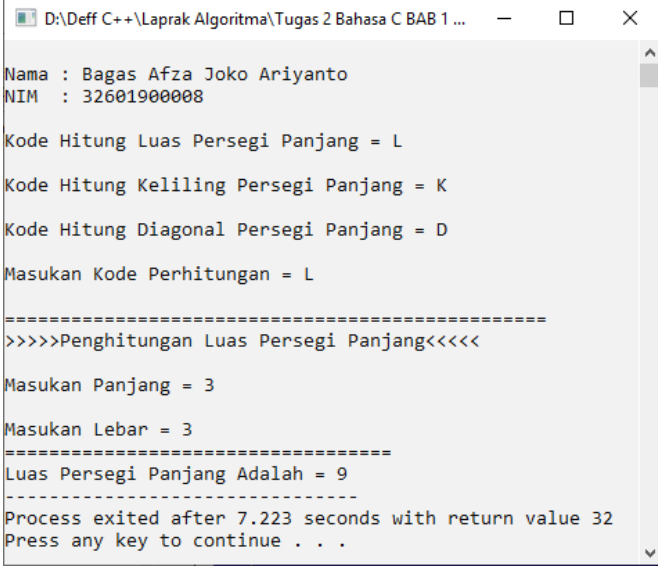
        keliling = 2 * (panjang + lebar);
        printf("\n=====");
        printf("\nKeliling Persegi Panjang =
%i",keliling);
    }else if(jenis == 'D' || jenis == 'd' ){
        printf("\n>>>>Penghitungan Diagonal Persegi
Panjang<<<<");
        printf("\n\nMasukan Panjang = ");
        scanf("%i", &panjang);
        printf("\nMasukan Lebar = ");
        scanf("%i", &lebar);

        diagonal=sqrt(panjang*panjang + lebar*lebar);
        printf("\n=====");
        printf("\nDiagonal Persegi Panjang =
%f",diagonal,panjang,lebar);
    }else
        printf("\nKode Tidak Valid");
}

```

b. *Flowchart*Gambar 1.16 *Flowchart* Tugas 3

c. *Output*



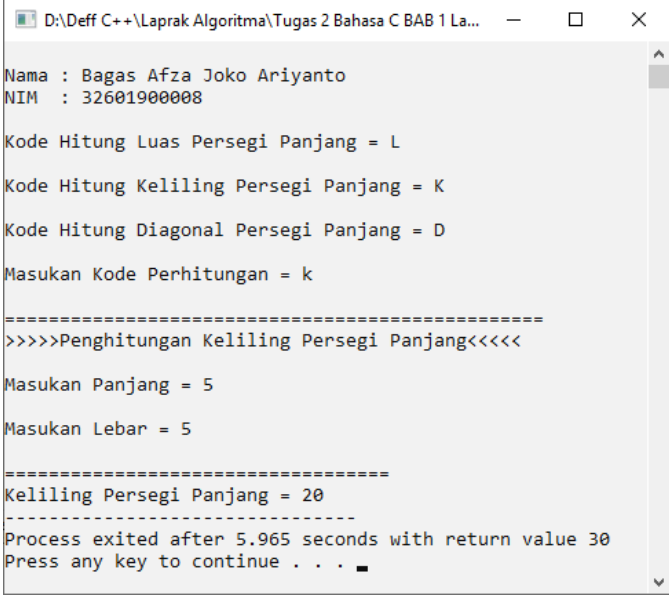
```

D:\Deff C++\Laprak Algoritma\Tugas 2 Bahasa C BAB 1 ...
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Kode Hitung Luas Persegi Panjang = L
Kode Hitung Keliling Persegi Panjang = K
Kode Hitung Diagonal Persegi Panjang = D
Masukan Kode Perhitungan = L
=====
>>>>Penghitungan Luas Persegi Panjang<<<<
Masukan Panjang = 3
Masukan Lebar = 3
=====
Luas Persegi Panjang Adalah = 9
-----
Process exited after 7.223 seconds with return value 32
Press any key to continue . . .

```

Gambar 1.17 *Output* Luas Persegi Panjang Bahasa C



```

D:\Deff C++\Laprak Algoritma\Tugas 2 Bahasa C BAB 1 La...
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Kode Hitung Luas Persegi Panjang = L
Kode Hitung Keliling Persegi Panjang = K
Kode Hitung Diagonal Persegi Panjang = D
Masukan Kode Perhitungan = k
=====
>>>>Penghitungan Keliling Persegi Panjang<<<<
Masukan Panjang = 5
Masukan Lebar = 5
=====
Keliling Persegi Panjang = 20
-----
Process exited after 5.965 seconds with return value 30
Press any key to continue . . .

```

Gambar 1.18 *Output* Keliling Persegi Panjang Bahasa C

```

D:\Deff C++\Laprak Algoritma\Tugas 2 Bahasa C BAB 1...
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Kode Hitung Luas Persegi Panjang = L
Kode Hitung Keliling Persegi Panjang = K
Kode Hitung Diagonal Persegi Panjang = D
Masukan Kode Perhitungan = D

=====
>>>>Penghitungan Diagonal Persegi Panjang<<<<

Masukan Panjang = 9
Masukan Lebar = 9

=====
Diagonal Persegi Panjang = 12.727922
-----
Process exited after 7.361 seconds with return value 37
Press any key to continue . . .

```

Gambar 1.19 Output Diagonal Persegi Panjang Bahasa C

d. Penjelasan

Sebelum saya membahas untuk soal nomor 3 kita ingat pada penjelasan soal nomor 1. Penjelasan pada nomor 1 dalam bahasa C ada *header* yang bernama *stdio.h* dan kita sudah mengenal apa itu *header* bahasa C *stdio.h* jika ada yang lupa baca kembali penjelasan pada nomor 1. Sekarang pada soal nomor 3 saya akan menjelaskan mengenai salah satu *header* dari bahasa C yaitu `#include<math.h>`. *Header* bernama *math.h* dalam bahasa C berfungsi untuk menerjemahkan kode rumus matematika. Dalam kode program saya, saya akan menggunakan fungsi matematika *sqrt()*.

Fungsi *sqrt()* digunakan untuk menghitung akar. Bila di bayangkan bisa diibaratkan kata *sqrt* adalah simbol akar $\sqrt{\quad}$. Dan tanda () di masukan subjek angka yang akan di akarkan.

Dalam kode program bahasa C pada program kali ini ada perbedaan yang sangat mencolok dari bahasa C++ yaitu adanya kode `%i`, `%s`, dan `%f`. Arti kode kode tersebut menjurus pada tipe *input* dan tipe


```

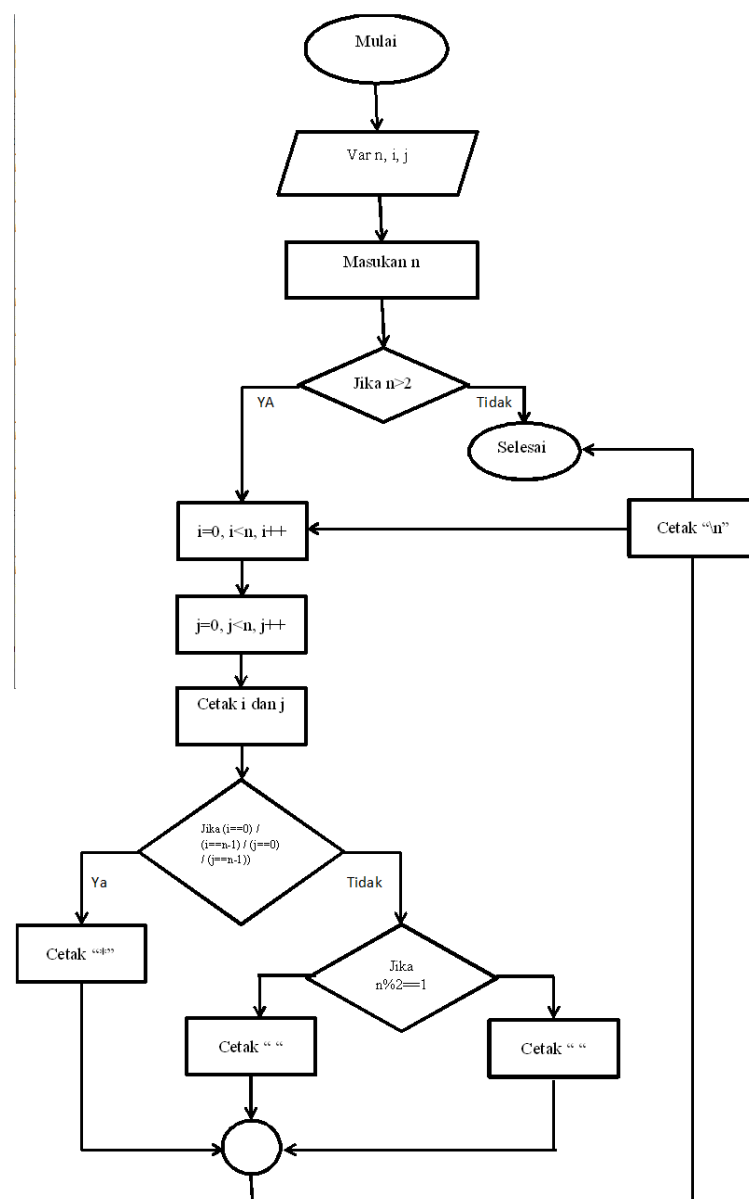
        printf(" ");
    }

    }
    printf("\n");
}

}
}

```

b. *Flowchart*



Gambar 1.20 *Output Tugas 4*

c. *Output*



```

D:\Deff C++\Laprak Algoritma\Tugas 4 BAB 1 Laprak....
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008
>>>>>>>>Program Membentuk Bingkai<<<<<<<<<

Input = 20
*****
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*****
-----
Process exited after 2.4 seconds with return value 0
Press any key to continue . . .

```

Gambar 1.21 *Output* Tugas 4

d. *Penjelasan*

Kode program di atas menggunakan bahasa C, menggunakan kombinasi percabangan (*branching*) dan perulangan (*looping*). Pada gambar 1.21 adalah *output* dari kode program yaitu membuat bingkai persegi dengan memasukan *inputannya*.

Dalam pembuatan bingkai bisa kita atur ingin ke bentuk bintang atau bentuk pagar, plus dan simbol lainnya yang ada di *keyboard*. Caranya dengan memasukan simbol tersebut ke dalam kode program bagian *printf*("*"). Tanda * dapat di rubah sesuka kalian.

5. Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

MENU PILIHAN

1. Dangdut
2. Pop
3. Rock
4. Exit

Pilihan Anda (1/2/3/4) ? ...

Apabila dipilih menu nomor 1, maka akan tampil teks “Anda memilih menu 1”. Demikian pula untuk nomor 2 dan 3. Kemudian setelah itu muncul teks “Tekan ENTER untuk kembali ke menu utama”. Artinya begitu kita tekan ENTER menu pilihan akan muncul kembali, dst. Akan tetapi bila yang dipilih menu 4 (EXIT), program langsung berhenti.

Jawaban :

a. Kode Program

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

main() {
    int sel;
    cout << "Nama : Bagas Afza Joko Ariyanto";
    cout << "\nNIM : 32601900008\n";
    menu:
        cout<< "\nMENU PILIHAN : ";
        cout<< "\n1. Dangdut";
        cout<< "\n2. Pop";
        cout<< "\n3. Rock";
        cout<< "\n4. Exit";
        cout<< "\nMasukan Pilihan Anda(1/2/3/4)? ";
        cin >> sel;
        do {
            switch(sel) {
```

```

        case 1 : cout<<
"===== " << "\nAnda memilih menu
satu" << "\nTekan ENTER untuk kembali ke menu utama" <<
"\n===== "; goto menu;

        case 2 : cout<<
"===== " << "\nAnda memilih menu
dua" << "\nTekan ENTER untuk kembali ke menu utama" <<
"\n===== "; goto menu;

        case 3 : cout<<
"===== " << "\nAnda memilih menu
tiga" << "\nTekan ENTER untuk kembali ke menu utama" <<
"\n===== "; goto menu;

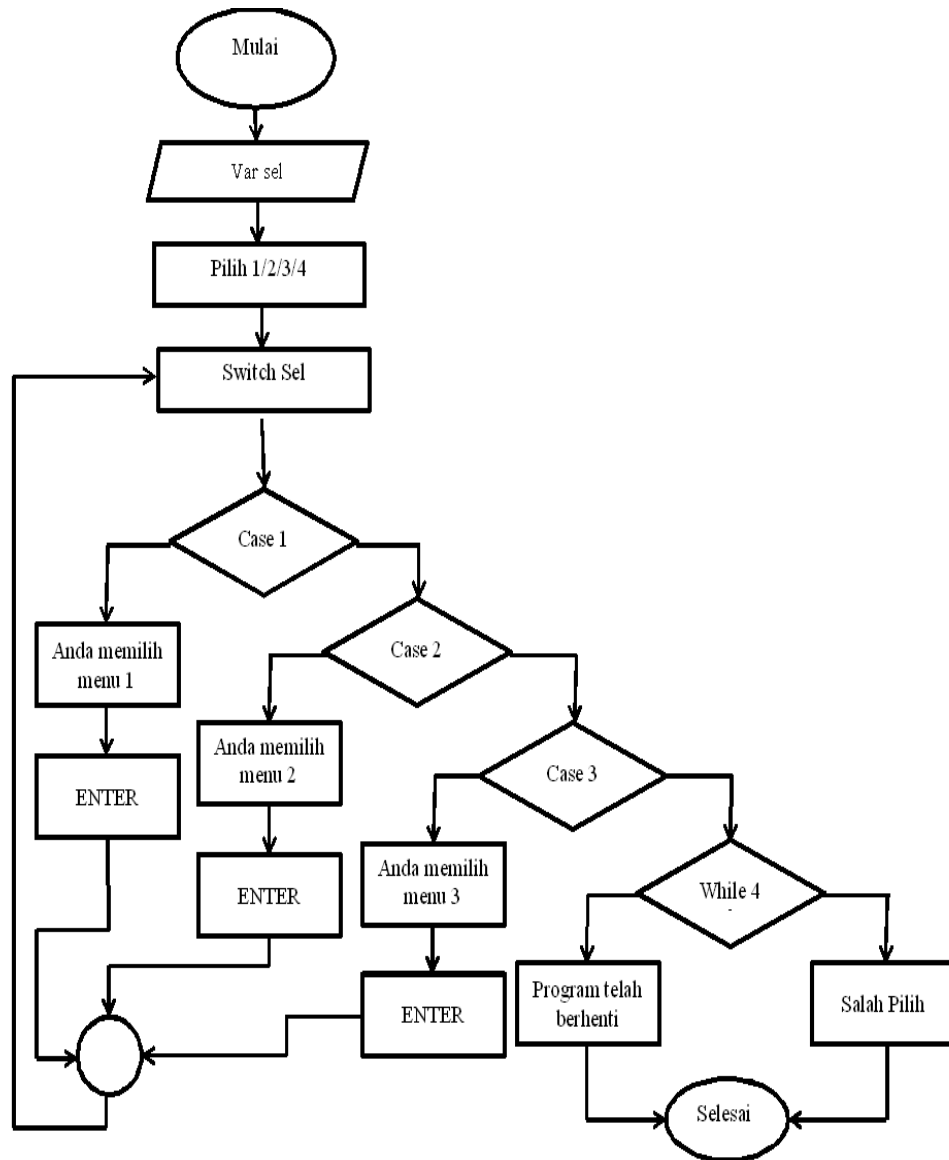
    }

    while (sel!=4){
        cout<< "===== " <<
"\nProgram telah berhenti" <<
"\n===== ";

    }

}

```

b. *Flowchart*Gambar 1.22 *Flowchart* Tugas 5

c. *Output*



```

D:\Deff C++\Laprak Algoritma\Tugas 5 BAB 1 Laprak....
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

MENU PILIHAN :
1. Dangdut
2. Pop
3. Rock
4. Exit
Masukan Pilihan Anda(1/2/3/4)? 1
=====
Anda memilih menu satu
Tekan ENTER untuk kembali ke menu utama
=====
MENU PILIHAN :
1. Dangdut
2. Pop
3. Rock
4. Exit
Masukan Pilihan Anda(1/2/3/4)? 2
=====
Anda memilih menu dua
Tekan ENTER untuk kembali ke menu utama
=====
MENU PILIHAN :
1. Dangdut
2. Pop
3. Rock
4. Exit
Masukan Pilihan Anda(1/2/3/4)? 3
=====
Anda memilih menu tiga
Tekan ENTER untuk kembali ke menu utama
=====
MENU PILIHAN :
1. Dangdut
2. Pop
3. Rock
4. Exit
Masukan Pilihan Anda(1/2/3/4)? 4
=====
Program telah berhenti
=====
-----

```

Gambar 1 23 *Output* Tugas 5

d. Penjelasan

Pada program di atas saya menggunakan fungsi switch-case fungsi switch-case sebenarnya sama dengan fungsi if-else perbedaan antara switch-case dan if-else terdapat pada cakupan ruangnya.

Maksudnya untuk switch-case seleksinya lebih kecil dari pada fungsi if-else. Dan pada program di atas saya menggunakan perintah do. Perintah do memungkinkan untuk menyambungkan perintah lain di akhir program seperti di atas di akhir program case 1 saya masukan goto menu artinya setelah case 1 selesai kita bisa kembali ke menu awal tanpa menghilangkan hasil dari program case 1.

1.5 Kesimpulan

Kesimpulannya dalam membuat program kita harus mengetahui kode-kode penting dalam bahasa pemrograman bisa dari bahasa C ataupun bahasa C++. Sebenarnya setiap bahasa pemrograman memiliki ciri khas sendiri-sendiri namun pada dasarnya semua bahasa pemrograman tidak jauh berbeda satu sama lain. Saran saya sebelum anda membuat program kenallilah bahasa pemrograman yang akan anda pakai karena itu sangat penting mulai dari *header*, *input output* dan tipe-tipe datanya.

Tidak hanya itu ketika anda akan membuat program anda harus mengerti bagaimana struktur program tersebut setiap perintah pasti memiliki strukturnya sendiri sendiri namun biasanya strukturnya sama persis beda pada penempatannya saja.

Disini kita telah mempelajari tentang perintah perintah utama dalam bahasa pemrograman seperti *sequential* yaitu dimana langsung di selesaikan, kemudian percabangan/penyeleksian (*branching*) yaitu dimana ada seleksi di dalamnya dan lagi ada perulangan *looping* dimana yang paling susah disini, dimana kita harus memasukan beberapa syarat untuk bisa menyelesaikan masalah pemrograman kita.

Dan tidak kita lupakan pula tentang pembuatan *flowchart* sebenarnya bagi yang masih awal memandang *flowchart* hanyalah hal yang

membingungkan dan tidak ada fungsinya namun ketika sudah mengerti apa fungsi sebenarnya *flowchart* anda pasti akan lebih mengutamakan membuat *flowchart* dari pada mencoba-coba dalam koding karena *flowchart* sendiri berfungsi untuk membuat alur dalam anda membuat koding agar bisa terstruktur. Karena algoritma sendiri adalah kumpulan perintah untuk menyelesaikan masalah secara runtut dan terstruktur.

BAB II

ARRAY 2D DAN 3D

2.1 Tujuan

Adapun tujuan dari praktikum ini:

1. Mengeal *array* 2 dimensi dan *array* 3 dimensi
2. Mengerti dan memahami penggunaan *array* 2 dimensi dan 3 dimensi

2.2 Dasar Teori

2.2.1 Pengertian Array

Array adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe yang sama. Setiap data tersebut menempati lokasi atau alamat memori yang berbeda- beda dan selanjutnya disebut dengan elemen *array*. Elemen *array* itu kemudian dapat kita akses melalui indeks yang terdapat di dalamnya.

Untuk mendeklarasikan sebuah *array* dalam C++, kita harus menggunakan tanda [] (*bracket*).

2.2.2 Array Multidimensi

Array Multi dimensi yaitu *array* yang terdiri dari beberapa subskrip *array*. Sebagai contoh, *array* 2 dimensi adalah *array* yang mempunyai 2 subskrip *array*, 3 dimensi mempunyai 3 subskrip *array* dan seterusnya. *Array* seperti ini sering digunakan untuk pemrosesan matriks.

1. Array Dua Dimensi

Array dua dimensi adalah *array* yang memiliki dua buah elemen bertipe *array*. Dengan kata lain, *array* dua dimensi memiliki dua buah subskrip, yang biasanya dipresentasikan dengan baris dan kolom.

Contoh sederhana adalah data yang tertulis dalam tabel berikut ini :

Tabel 2. 1 *Array 2 Dimensi*

Merk HP	1992	1993	1994	1995
1. Nokia	35	45	80	120
2. Samsung	100	110	70	101
3. SONY	10	15	20	17

Jika dibuat programnya adalah sebagai berikut :

```
Int data_hp [3] [4];
```

Bentuk umum pendeklarasian sebuah array dua dimensi adalah sebagai berikut :

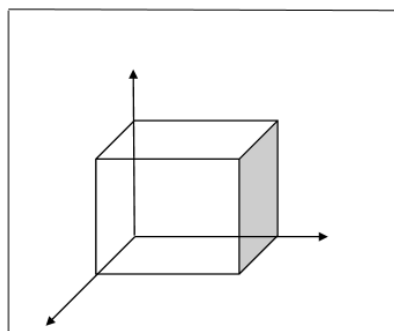
```
tipe_data
```

```
nama_array [jumlah_elemen_baris] [jumlah_elemen_kolom] ;
```

2. *Array Tiga Dimensi*

Array tiga dimensi merupakan array yang memiliki tiga buah subskrip dan lebih kompleks apabila dibandingkan dengan array dua dimensi. Di sini, subskrip yang ada akan dipresentasikan dengan sumbu x, y, z atau panjang, lebar, dan tinggi seperti yang ditunjukkan oleh gambar berikut.

Bentuk umum pendeklarasian sebuah *array* 3 dimensi adalah sebagai berikut :



Gambar 2. 1 *Array 3 Dimensi*

```
tipe_data

nama_array      [jumlah_elemen_x]      [jumlah_elemen_y]
[jumlah_elemen_z] ;
```

2.2.3 Array Yang Bersifat Konstan

Nilai dalam elemen *array* dapat dibuat tetap, yaitu dengan mendefinisikannya sebagai konstanta. Caranya sama seperti pada saat kita membuat konstanta dari *identifier*, yaitu dengan menggunakan kata kunci `const` didepan nama *array* yang didefinisikan.

2.2.4 Array Sebagai Tipe Data Bentukan

Dalam C++, *array* juga dapat digunakan sebagai tipe data bentukan seperti halnya struktur dan enumerasi. Untuk melakukan hal ini kita harus menggunakan kata kunci `typedef`, dimana berfungsi untuk memberikan nama lain dari *array* yang dideklarasikan. Berikut ini bentuk umum untuk membuat *array* sebagai tipe data bentukan.

```
Typedef tipe_data nama_array[jumlah_elemen];
```

2.2.5 Array dari karakter

Dalam C++, kumpulan karakter disebut dengan *string* (teks). Dan untuk mendeklarasikan *array* dari tipe karakter kita tentu akan menuliskannya ke dalam bentuk umum seperti dibawah ini :

```
Char nama_array[jumlah_elemen];
```

Dengan demikian, apabila kita ingin melakukan deklarasi variabel *string* (misalnya dengan nama TEKS) yang terdiri dari 5 buah karakter, maka sintaksnya adalah sebagai berikut :

```
Char TEKS[5] = {'B', 'U', 'D', 'I', '\0'};
```

Karakter terakhir, `'\0'` disebut dengan karakter *null*, yaitu karakter yang digunakan sebagai terminator dari sebuah *string* didalam C++. Namun, karena bentuk tersebut susah untuk ditulis dan riskan terhadap terjadinya sebuah kesalahan, maka C++ memperbolehkan kita untuk dapat menuliskan

karakter-karakter tersebut dalam sebuah string yaitu dengan cara seperti dibawah ini :

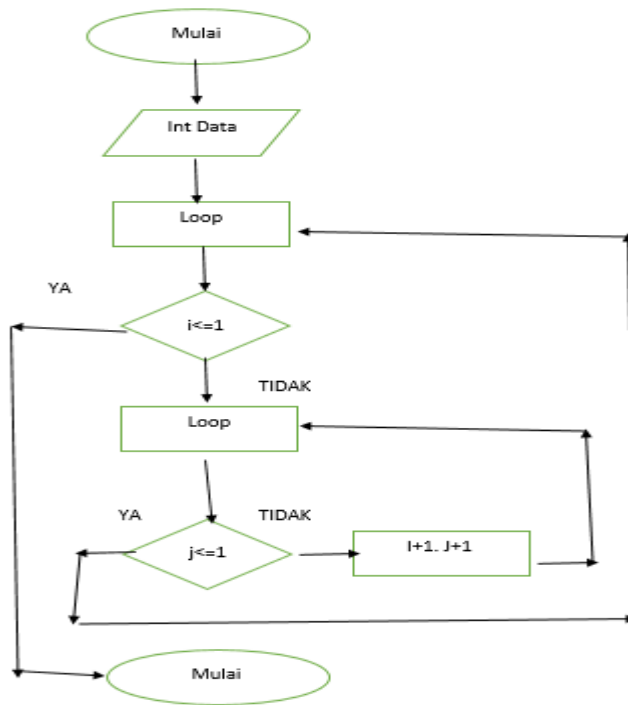
```
Char TEKS[5] = "BUDI";
```

2.3 Latihan

1. Latihan 1

a. Kode *Program*

```
#include <iostream>
using namespace std;
main(){
    cout<< " Nama : Bagas Afza Joko Ariyanto\n";
    cout<< " NIM   : 326019000008\n\n";
    int i, j;
    int data [2] [2];
    cout<<"Input Data"<<endl;
    for(i=0 ; i<=1 ;i++)
    {
        for(j=0 ;j<=1 ;j++)
        {
            cout<<"data["<<i+1<<"] ["<<j+1<<"]="";
            cin >> data [i] [j];
        }
    }
    cout<<"Matriks yang diinputkan : "<<endl;
    for(i=0 ;i<=1 ;i++)
    {
        for(j=0 ;j<=1 ;j++)
        {
            cout<<data[i][j]<<" ";
        }
        cout<<endl;
    }
}
```

b. *Flowchart*Gambar 2. 2 *Flowchart* Latihan 1c. *Output*

```

D:\Deff C++\Laprak Algoritma\Latihan 1 BAB 2.exe
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008

Input Data
data[1] [1]=1
data[1] [2]=2
data[2] [1]=3
data[2] [2]=4
Matriks yang diinputkan :
1 2
3 4

-----
Process exited after 5.187 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 3 *Output* Latihan 1

d. Penjelasan

Program pada latihan 1 ini menampilkan program *array* yang berupa matrik 2x2 (2 baris dan 2 kolom). *Header* yang digunakan adalah *iostream* untuk menampilkan *cin*, *cout*, *endl*, dan *endl*. Lalu menggunakan tipe data *integer* karena menunjukkan sebuah variabel untuk menyimpan nilai yang dimasukkan, dalam pemasukan nilai harus menggunakan tanda [] (*bracket*) sebagai pendeklarasian bentuk *array*. Pertama kita menginput nilai data yang diinginkan, menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya. Lalu membuat *output* data matriks yang telah diinputkan.

2. Latihan 2

a. Kode Program

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

int main(){
    cout<<"Nama : Bagas Afza Joko Ariyanto \n";
    cout<<"NIM : 32601900008\n\n";
    float nilai [2] [3];
    int baris, kolom;
    for(baris=0; baris<2; baris++)
    {
        for(kolom=0; kolom<3; kolom++)
        {
            cout<<"Nilai   [baris]   [kolom]   ke-   ["<<baris+1<<"]
["<<kolom+1<<"] = ";
            cin>>nilai[baris][kolom];
        }
    }
    cout<<endl;
    cout<<"Matrik Nilai yang dimasukan : "<<endl<<endl;
    for(baris=0; baris<2; baris++)
    {
        for(kolom=0; kolom<3; kolom++)
```

```

    {
        cout<<nilai [baris] [kolom]<<" ";
    }
    cout<<endl;
}
cout<<endl;

float jml_nilai1, jml_nilai2;
float rerata1, rerata2;
jml_nilai1 =0;
jml_nilai2 =0;

float jml_nilaicol1, jml_nilaicol2, jml_nilaicol3;
float reratacol1, reratacol2, reratacol3;
jml_nilaicol1=0;
jml_nilaicol2=0;
jml_nilaicol3=0;

for(baris=0; baris<2; baris++)
{
    if(baris==0)
    {
        for(kolom=0; kolom<3; kolom++)
            jml_nilai1=jml_nilai1 + nilai[baris][kolom];
    }
    else
    {
        for(kolom=0; kolom<3; kolom++)
            jml_nilai2=jml_nilai2 + nilai[baris][kolom];
    }
}
rerata1=jml_nilai1/3;
rerata2=jml_nilai2/3;

cout<<"rerata baris ke 1 = "<<rerata1<<endl;
cout<<"rerata baris ke 2 = "<<rerata2<<endl;

//per kolom

```



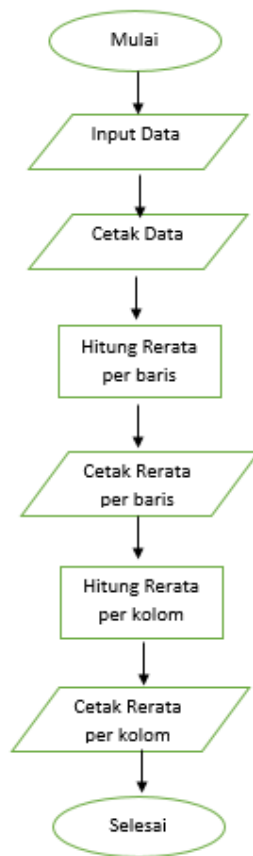
```

cout<<endl;
for(kolom=0; kolom<3; kolom++)
{
    if(kolom==0)
    {
        for(baris=0; baris<2; baris++)
            jml_nilaicol1=jml_nilaicol1 +
nilai[baris][kolom];
    }
    else
    if(kolom==1)
    {
        for (baris=0; baris<2; baris++)
            jml_nilaicol2=jml_nilaicol2 + nilai [baris][kolom];
    }
    else
    {
        for(baris=0; baris<2; baris++)
            jml_nilaicol3=jml_nilaicol3 +
nilai[baris][kolom];

    }
}
reratacol1=jml_nilaicol1/2;
reratacol2=jml_nilaicol2/2;
reratacol3=jml_nilaicol3/2;

cout<<"rerata kolom ke 1 = "<<reratacol1<<endl;
cout<<"rerata kolom ke 2 = "<<reratacol2<<endl;
cout<<"rerata kolom ke 3 = "<<reratacol3<<endl;
}

```

b. *Flowchart*Gambar 2. 4 *Flowchart* Latihan 2c. *Output*

```

D:\Deff C++\Laprak Algoritma\Latihan 2 BAB 2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Nilai [baris] [kolom] ke- [1] [1] = 4
Nilai [baris] [kolom] ke- [1] [2] = 3
Nilai [baris] [kolom] ke- [1] [3] = 2
Nilai [baris] [kolom] ke- [2] [1] = 5
Nilai [baris] [kolom] ke- [2] [2] = 7
Nilai [baris] [kolom] ke- [2] [3] = 8

Matrik Nilai yang dimasukan :
4 3 2
5 7 8

rerata baris ke 1 = 3
rerata baris ke 2 = 6.66667

rerata kolom ke 1 = 4.5
rerata kolom ke 2 = 5
rerata kolom ke 3 = 5

-----
Process exited after 9.863 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 2. 5 *Output* Latihan 2

d. Penjelasan

Program ini menampilkan array matriks 2x3 (2 baris dan 3 kolom) beserta rerata baris dan kolomnya. *Header* yang digunakan adalah *iostream* yang merupakan *header* pada program C++ untuk menampilkan *cin*, *cout*, *endl*, dan *ends*. Tipe data yang digunakan adalah *float* sebagai deklarasi nilai desimal. Menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya. Menggunakan fungsi (*if*) untuk menyeleksi kondisi, yaitu jika pernyataan pertama tidak sesuai maka diseleksi dipernyataan kedua dan seterusnya. Maka disertakan pula fungsi (*else*) yang merupakan lanjutan dari fungsi (*if*) dimana jika kondisi benar maka statement pertama yang akan dilaksanakan, tetapi jika kondisi salah maka statement yang ada pada blok *else* yang dilaksanakan. Lalu memasukkan inputan baris dan kolom serta nilai matriks yang diinginkan. Memasukkan input rerata baris dan kolom. Jika telah memasukkan input dengan benar maka akan menghasilkan output matriks baris dan kolom yaitu 2 baris dan 3 kolom serta menghasilkan output rerata matriks sesuai dengan baris dan kolomnya.

3. Latihan 3

a. Kode Program

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

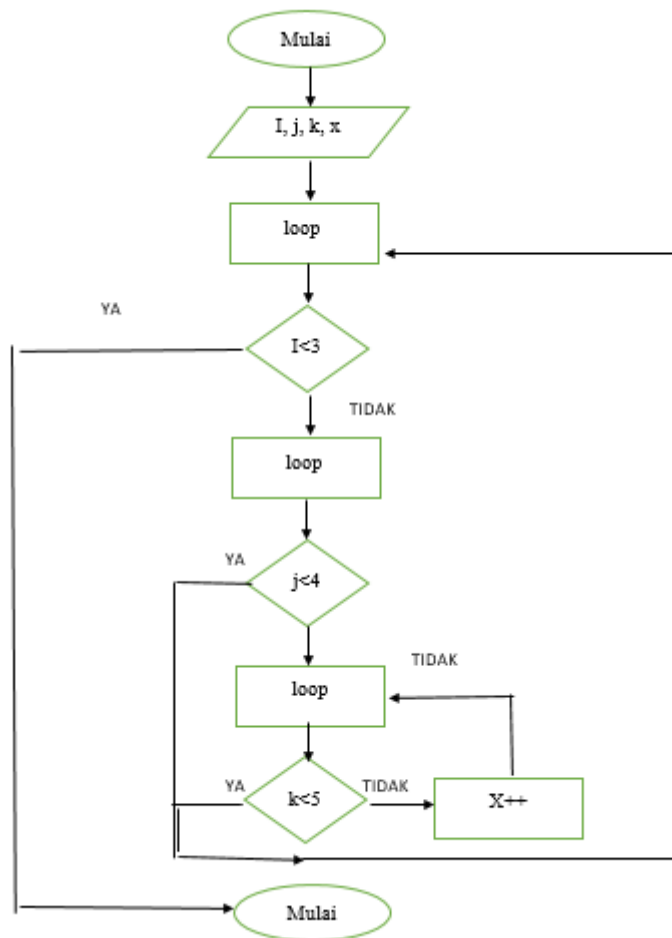
main() {
    cout<<"Nama : Bagas Afza Joko Ariyanto\n";
    cout<<"NIM : 32601900008\n\n";

    int i, j, k, x=1;
    int matrik [3] [4] [4];

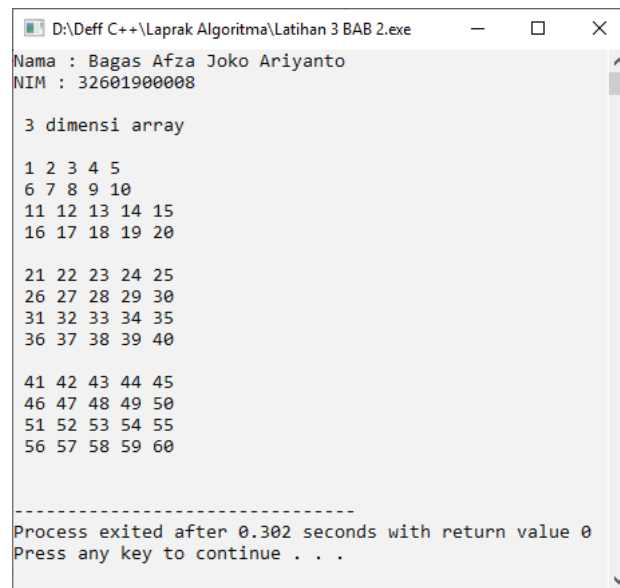
    cout<<" 3 dimensi array "<<endl<<endl;

    for(i=0; i<3; i++)
```

```
{  
    for(j=0; j<4; j++)  
    {  
        for(k=0; k<5; k++)  
        {  
            matrik [i] [j] [k]= x;  
            cout<<" "<<matrik[i] [j] [k];  
            x++;  
        }  
        cout<<endl;  
    }  
    cout<<endl;  
}
```

b. *Flowchart*Gambar 2. 6 *Flowchat* Latihan 3

c. Output



```

D:\Deff C++\Laprak Algoritma\Latihan 3 BAB 2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

3 dimensi array

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

21 22 23 24 25
26 27 28 29 30
31 32 33 34 35
36 37 38 39 40

41 42 43 44 45
46 47 48 49 50
51 52 53 54 55
56 57 58 59 60

-----
Process exited after 0.302 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 7 Output Latihan 3

d. Penjelasan

Program ini menampilkan *array* 3 dimensi. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ yaitu untuk menampilkan *cin*, *cout*, *endl*, dan *ends*. Tipe data yang digunakan adalah *integer* yang menampilkan variabel untuk menyimpan nilai yang dimasukkan. Matriks *array* 3 dimensi dapat digambarkan sebagai benda ruang. Program diatas di-*input*-kan 3 dimensi dimana angka-angkanya selalu berselisihan. Dan program tersebut menggunakan fungsi (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya, lalu ada juga fungsi matriks yang berfungsi sebagai kumpulan bilangan, simbol, atau ekspresi.

2.4 Tugas

1. Ubahlah latihan 2 kedalam bahasa C!

Jawaban:

a. Kode Program

```
#include <stdio.h>

int main(){
printf(" Nama : Bagas Afza Joko Ariyanto\n");
printf(" NIM : 32601900008\n\n");
int nilai [2] [3], baris, kolom;
for(baris=0; baris<2; baris++)
{
    for(kolom=0; kolom<3; kolom++)
    {
        printf("Nilai [baris] [kolom] ke-[%d] [%d]
= ", baris+1, kolom+1);
        scanf("%d", &nilai[baris] [kolom]);
    }
}
printf("\n");
printf("Matriks Nilai yang dimasukkan :\n\n");
for(baris=0; baris<2; baris++)
{
    for(kolom=0; kolom<3; kolom++)
    {
        printf("%d" " " ", nilai [baris] [kolom]);

    }
    printf("\n");
}
float jml_nilai1, jml_nilai2;
float rerata1, rerata2;
jml_nilai1=0;
jml_nilai2=0;

float jml_nilaicol1, jml_nilaicol2, jml_nilaicol3;
float reratacol1, reratacol2, reratacol3;
```

```

jml_nilaicol1=0;
jml_nilaicol2=0;
jml_nilaicol3=0;
for(baris=0; baris<2; baris++)
{
    if (baris==0)
    {
        for(kolom=0; kolom<3; kolom++)
jml_nilai1=jml_nilai1 + nilai[baris][kolom];
    }
    else
    {
        for(kolom=0; kolom<3; kolom++)
jml_nilai2=jml_nilai2 + nilai [baris][kolom];
    }
}
rerata1=jml_nilai1/3;
rerata2=jml_nilai2/3;
printf("rerata baris ke 1 = %f \n",rerata1);
printf("rerata baris ke 2 = %f \n",rerata2);
printf("\n");
for(kolom=0; kolom<3; kolom++)
{
    if(kolom==0)
    {
        for(baris=0; baris<2;baris++)
jml_nilaicol1=jml_nilaicol1 +
nilai[baris][kolom];

    }
    else if(kolom==1)
    {
        for(baris=0; baris<2; baris++)
jml_nilaicol2=jml_nilaicol2 +
nilai[baris][kolom];

    }
    else

```



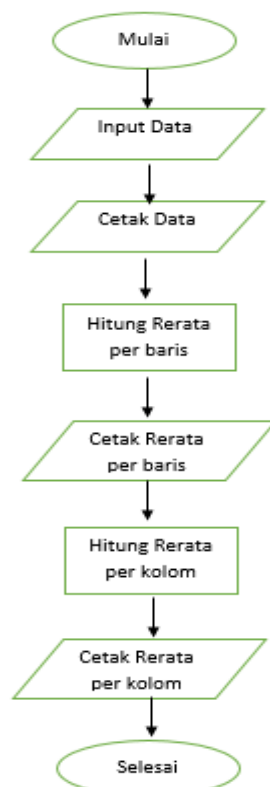
```

{
    for(baris=0; baris<2; baris++)
        jml_nilaicol3=jml_nilaicol3 + nilai
        [baris][kolom];

}
}
reratacol1=jml_nilaicol1/2;
reratacol2=jml_nilaicol2/2;
reratacol3=jml_nilaicol3/2;
printf("rerata kolom ke 1 = %f \n", reratacol1);
printf("rerata kolom ke 2 = %f \n", reratacol2);
printf("rerata kolom ke 3 = %f \n", reratacol3);
return 0;
}

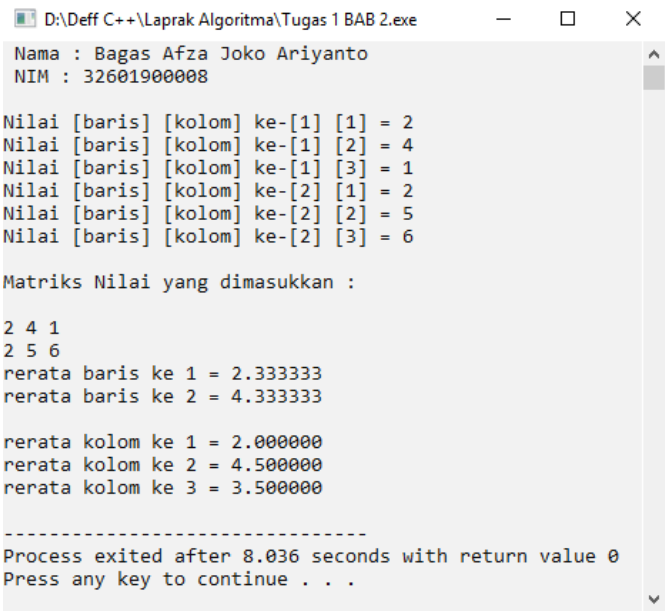
```

b. *Flowchart*



Gambar 2. 8 *Flowchart* Tugas 1

c. Output



```

D:\Deff C++\Laprak Algoritma\Tugas 1 BAB 2.exe
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008

Nilai [baris] [kolom] ke-[1] [1] = 2
Nilai [baris] [kolom] ke-[1] [2] = 4
Nilai [baris] [kolom] ke-[1] [3] = 1
Nilai [baris] [kolom] ke-[2] [1] = 2
Nilai [baris] [kolom] ke-[2] [2] = 5
Nilai [baris] [kolom] ke-[2] [3] = 6

Matriks Nilai yang dimasukkan :

2 4 1
2 5 6
rerata baris ke 1 = 2.333333
rerata baris ke 2 = 4.333333

rerata kolom ke 1 = 2.000000
rerata kolom ke 2 = 4.500000
rerata kolom ke 3 = 3.500000

-----
Process exited after 8.036 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 9 Output Tugas 1

d. Penjelasan

Program ini menampilkan *array* matriks 2x3 (2 baris dan 3 kolom) beserta rerata baris dan kolomnya. *Header* yang digunakan adalah *stdio.h* yang merupakan *header* bahasa C yaitu *header* yang berisi definisi *makro*, *konstanta*, dan deklarasi fungsi serta jenis yang digunakan untuk berbagai *standard input output* operasi. Tipe data yang digunakan adalah *integer* untuk menunjukkan sebuah variabel. Kita menginputkan data matriks. Menggunakan fungsi *printf* untuk menampilkan keluaran data serta hasil rerata baris kolom yang telah diinputkan. Menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya.

2. Jalankan program berikut :

Jawaban :

a. Kode Program

```

#include<iostream>

using namespace std;

```

```

main() {
    cout<<"Nama : Bagas Afza Joko Ariyanto\n";
    cout<<"NIM : 32601900008\n\n";
    typedef int matrik [3][2];
    int elemen1[3][3];
    int elemen2[3][3];
    int hasil_jumlah[3][3], clrscr[3][2];
    int baris, kolom;
    for (baris=1; baris<=3; baris++)
    {
        for(kolom=1; kolom<=2; kolom++)
        {

            cout<<"elemen A["<<baris<<"]["<<kolom<<"] = ";

            cin>>elemen1 [baris][kolom];
        }
    }
    cout<<endl;
    for(baris=1; baris<=3; baris++)
    {
        for(kolom=1; kolom<=2; kolom++)
        {

            cout<<"elemen B["<<baris<<"]["<<kolom<<"] = ";

            cin>>elemen2 [baris][kolom];

        }
    }
    cout<<endl;
    for(baris=1; baris<=3; baris++)
    {
        for(kolom=1; kolom<=2; kolom++)
        {
            hasil_jumlah[baris][kolom]=elemen1[baris][kolom]      +
            elemen2[baris][kolom];
        }
    }
}

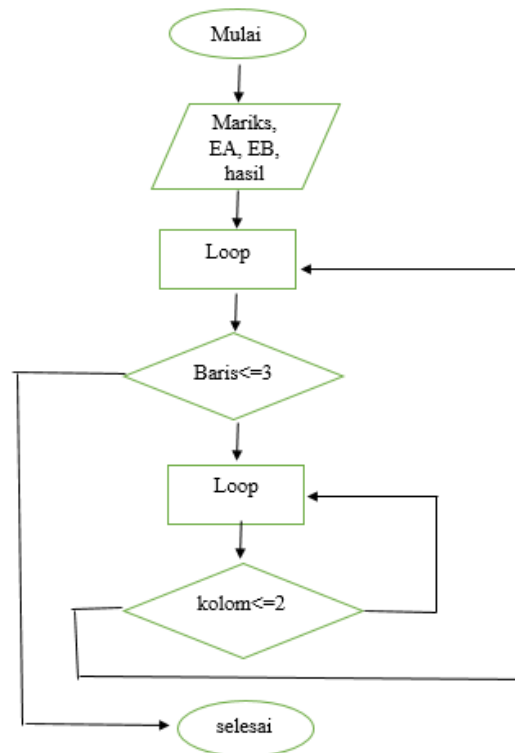
```

```

cout<<"hasil penjumlahan elemen A dan elemen B adalah :
"<<endl;
for(baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
cout<<"hasil_jumlah      ["<<baris<<"] ["<<kolom<<"]      =
"<<hasil_jumlah[baris][kolom]<<endl;
}
}
}

```

b. *Flowchart*



Gambar 2. 10 *Flowchart* Tugas 2

c. Output



```

D:\Deff C++\Laparak Algoritma\Tugas 2 BAB 2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

elemen A[1][1] = 1
elemen A[1][2] = 2
elemen A[2][1] = 3
elemen A[2][2] = 4
elemen A[3][1] = 5
elemen A[3][2] = 6

elemen B[1][1] = 7
elemen B[1][2] = 8
elemen B[2][1] = 9
elemen B[2][2] = 2
elemen B[3][1] = 3
elemen B[3][2] = 4

hasil penjumlahan elemen A dan elemen B adalah :
hasil_jumlah [1][1] = 8
hasil_jumlah [1][2] = 10
hasil_jumlah [2][1] = 12
hasil_jumlah [2][2] = 6
hasil_jumlah [3][1] = 8
hasil_jumlah [3][2] = 10

-----
Process exited after 10.92 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 11 Output Tugas 2

d. Penjelasan

Program ini menampilkan hasil penjumlahan 2 elemen (elemen A dan elemen B), dengan menginputkan elemen A dan elemen B serta matriks baris dan kolom, lalu menyertakan *output* hasil penjumlahan kedua elemen tersebut. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ berfungsi untuk menampilkan *cin*, *cout*, *endl*, dan *ends*. Tipe data yang digunakan adalah *typedef* yang digunakan untuk mendeklarasikan tipe data dengan nama lain. Serta menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya.

3. Modifikasi tugas 2 dengan menginputkan jumlah baris dan kolom.

Kemudian tambahkan operasi pengurangan dan perkalian didalamnya!

Jawaban :

a. Kode Program

```

#include<iostream>
using namespace std;
main() {
    cout<<"Nama : Bagas Afza Joko Ariyanto\n";
    cout<<"NIM : 32601900008\n\n";

```

```

typedef int matrik [3][2];
int elemen1[3][3];
int elemen2[3][3];
int hasil_jumlah[3][3];
int hasil_kurang[3][3];
int hasil_kali[3][5];
int baris, kolom;
for (baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
cout<<"elemen A["<<baris<<"]["<<kolom<<"] = ";
cin>>elemen1 [baris][kolom];
}
}
cout<<endl;
for(baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
cout<<"elemen B["<<baris<<"]["<<kolom<<"] = ";
cin>>elemen2 [baris][kolom];
}
}
cout<<endl;
for(baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
hasil_jumlah[baris][kolom]=elemen1[baris][kolom]      +
elemen2[baris][kolom];

}
}
cout<<"hasil penjumlahan elemen A dan elemen B adalah :
"<<endl;

for(baris=1; baris<=3; baris++)
{

```

```

for(kolom=1; kolom<=2; kolom++)
{
cout<<"hasil_jumlah      ["<<baris<<"] ["<<kolom<<"]      =
"<<hasil_jumlah[baris][kolom]<<endl;
}
}
cout<<endl;
for(baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
hasil_kurang[baris][kolom]=elemen1[baris][kolom]      -
elemen2[baris][kolom];
}
}
cout<<"hasil pengurangan elemen A dan elemen B adalah :
"<<endl;
for(baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
cout<<"hasil_kurang      ["<<baris<<"] ["<<kolom<<"]      =
"<<hasil_kurang[baris][kolom]<<endl;
}
}
cout<<endl;
for(baris=1; baris<=3; baris++)
{
for(kolom=1; kolom<=2; kolom++)
{
hasil_kali[baris][kolom]=elemen1[baris][kolom]      *
elemen2[baris][kolom];
}
}
cout<<"hasil perkalian elemen A dan elemen B adalah :
"<<endl;
for(baris=1; baris<=3; baris++)
{

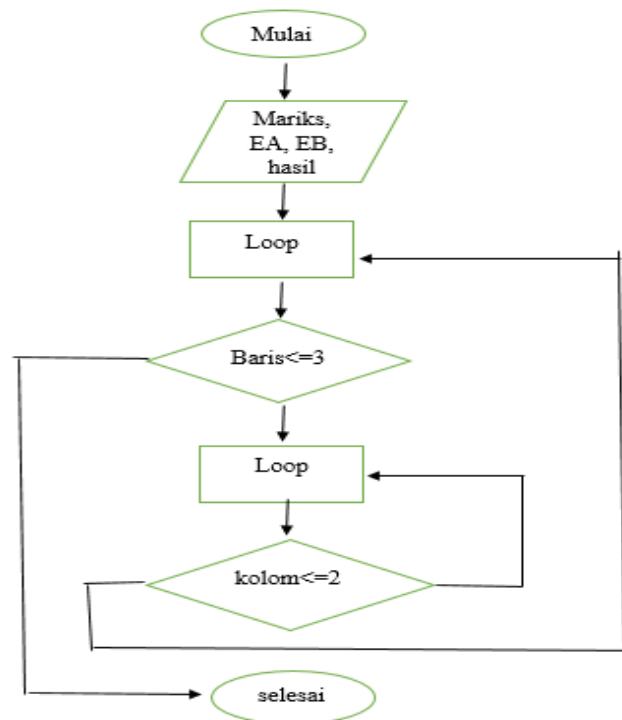
```

```

for(kolom=1; kolom<=2; kolom++)
{
    cout<<"hasil_kali["<<baris<<"] ["<<kolom<<"]
    "<<hasil_kali[baris][kolom]<<endl;
}
}
}

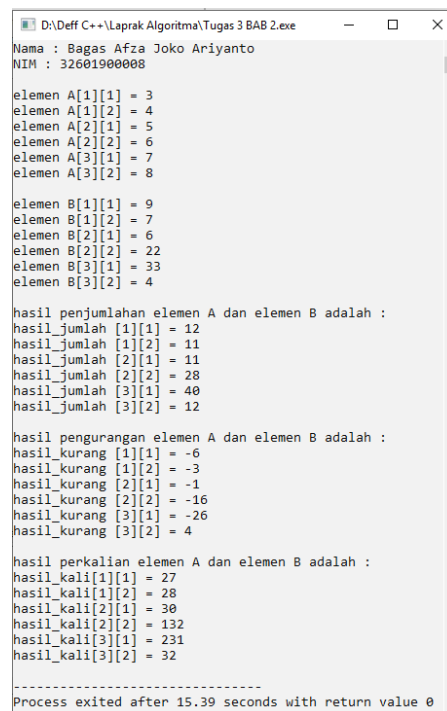
```

b. *Flowchart*



Gambar 2. 12 *Flowchart* Tugas 3

c. Output



```

D:\Deff C++\Laprak Algoritma\Tugas 3 BAB 2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

elemen A[1][1] = 3
elemen A[1][2] = 4
elemen A[2][1] = 5
elemen A[2][2] = 6
elemen A[3][1] = 7
elemen A[3][2] = 8

elemen B[1][1] = 9
elemen B[1][2] = 7
elemen B[2][1] = 6
elemen B[2][2] = 22
elemen B[3][1] = 33
elemen B[3][2] = 4

hasil penjumlahan elemen A dan elemen B adalah :
hasil_jumlah [1][1] = 12
hasil_jumlah [1][2] = 11
hasil_jumlah [2][1] = 11
hasil_jumlah [2][2] = 28
hasil_jumlah [3][1] = 40
hasil_jumlah [3][2] = 12

hasil pengurangan elemen A dan elemen B adalah :
hasil_kurang [1][1] = -6
hasil_kurang [1][2] = -3
hasil_kurang [2][1] = -1
hasil_kurang [2][2] = -16
hasil_kurang [3][1] = -26
hasil_kurang [3][2] = 4

hasil perkalian elemen A dan elemen B adalah :
hasil_kali[1][1] = 27
hasil_kali[1][2] = 28
hasil_kali[2][1] = 30
hasil_kali[2][2] = 132
hasil_kali[3][1] = 231
hasil_kali[3][2] = 32

-----
Process exited after 15.39 seconds with return value 0

```

Gambar 2. 13 Output Tugas 3

d. Penjelasan

Program ini menampilkan hasil penjumlahan, pengurangan, dan perkalian matriks dengan 2 elemen. Yaitu dengan menginputkan elemen A dan elemen B. Lalu menyertakan *output* hasil dari penjumlahan, pengurangan, dan perkalian kedua elemen tersebut. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ berfungsi untuk menampilkan *cin*, *cout*, *endl*, dan *ends*. Tipe data yang digunakan adalah *typedef* yang digunakan untuk mendeklarasikan tipe data dengan nama lain. Serta menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya.

4. Jalankan program berikut dan kedalam bahasa C++ !

```

#include<iostream>

Using namespace std;

main() {
    cout<<"Nama : Bagas Afza Joko Ariyanto\n";

```

```

cout<<"NIM : 32601900008\n\n";
Char h=70, nama [4][3][13] =
{
"Angsa", "Anjing", "Anoa",
"Banteng", "Badak", "Beruang", "Camar", "Capung",
"Cendrawasih", "Dara", "Domba", "Duyung",
};
printf("Daftar Hewan Sesuai Abjad : \n\n");
for(int i=0; i<4; i++)
{
++h;
printf("kelompok %c \n", h);
for(s=1; s<=3; s++)
{
printf("%d. %s \n", s, nama [i][s]);
}
printf("\n");
}
}

```

Jawaban :

a. Kode Program

```

#include<iostream>
using namespace std;
int main(){
cout<<"Nama : Bagas Afza Joko Ariyanto\n";
cout<<"NIM : 32601900008\n\n";
cout<<"daftar hewan sesuai abjad\n\n";
cout<<"Kelompok A"<<endl;
cout<<"1.Angsa"<<endl;
cout<<"2.Anjing"<<endl;
cout<<"3.Anoa\n\n";
cout<<"Kelompok b"<<endl;
cout<<"1.Banteng"<<endl;
cout<<"2.Badak"<<endl;
cout<<"3.Beruang\n\n";
cout<<"Kelompok C"<<endl;
cout<<"1.Camar"<<endl;

```


```
cout<<"2.Camar"<<endl;  
cout<<"3.Capung\n\n";  
cout<<"Kelompok D"<<endl;  
cout<<"1.Dara"<<endl;  
cout<<"2.Domba"<<endl;  
cout<<"3.Duyung\n\n";  
}
```

b. *Flowchart*



Gambar 2. 14 *Flowchart* Tugas 4

c. Output



```

D:\Deff C++\Laprak Algoritma\Tugas 4 BAB 2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

daftar hewan sesuai abjad

Kelompok A
1.Angsa
2.Anjing
3.Anoa

Kelompok b
1.Banteng
2.Badak
3.Beruang

Kelompok C
1.Camar
2.Camar
3.Capung

Kelompok D
1.Dara
2.Domba
3.Duyung

-----
Process exited after 0.2386 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 15 Output Tugas 4

d. Penjelasan

Program pada tugas 4 ini menampilkan *output* daftar hewan yang *diinputkan* sesuai dengan abjad. *Header* yang digunakan adalah *iostream* untuk menampilkan *cin*, *cout*, *endl*, dan *ends*. Tipe data yang digunakan adalah *integer* karena berfungsi menunjukkan sebuah variabel untuk menyimpan nilai yang dimasukkan. Kita menampilkan nama-nama hewan yang sesuai dengan abjad, lalu membuat *output* untuk menampilkan hasil penyesuaian nama-nama hewan tersebut.

5. Buatlah program untuk membentuk sebuah kelompok dimana jumlah kelompok dan anggota *diinputkan*. Kemudian tampilkan kembali kelompok dan anggota yang *diinputkan*.

Jawaban :

a. Kode Program

```

#include<iostream>
using namespace std;
main(){
cout<<" Nama : Bagas Afza Joko Ariyanto\n";
cout<<" NIM : 32601900008\n\n";
    int x,y;

```

```

float f,j;
string anggota;
cout<<"masukan jumlah kelompok : ";
cin>>j;
for(x=0; x<j;){
    cout<<"kelompok"<<" "<<x+1<<endl;
    cout<<"jumlah anggota kelompok : ";
    cin>>f;
    for(y=1; y<=f;){
        cout<<" "<<y<<".";
        cin>>anggota;
        y++;
    }
    x++;
}
}

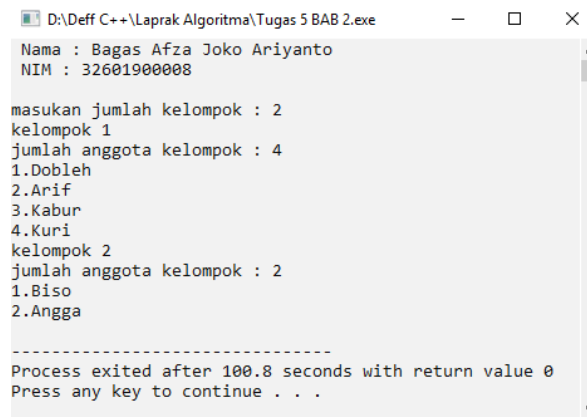
```

b. *Flowchart*



Gambar 2. 16 *Flowchart* Tugas 5

c. Output



```

D:\Deff C++\Laprak Algoritma\Tugas 5 BAB 2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

masukan jumlah kelompok : 2
kelompok 1
jumlah anggota kelompok : 4
1.Dobleh
2.Arif
3.Kabur
4.Kuri
kelompok 2
jumlah anggota kelompok : 2
1.Biso
2.Angga

-----
Process exited after 100.8 seconds with return value 0
Press any key to continue . . .

```

Gambar 2. 17 Output Tugas 5

d. Penjelasan

Program ini menampilkan sebuah kelompok dan anggotanya. Yaitu dengan menginputkan jumlah kelompok dan jumlah anggotanya. Kemudian pada *output* disertakan kelompok dan anggota yang telah diinputkan, agar dapat ditampilkan kembali kelompok dan anggotanya. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ untuk menampilkan *cin*, *cout*, *endl*, dan *ends*. Menggunakan tipe data *integer* yang menunjukkan sebuah variabel. Sedangkan *string* digunakan untuk menyimpan data berbentuk kalimat.

2.5 Kesimpulan

Pada BAB II ini membahas mengenai *Array* 2D dan 3D. *Array* merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe yang sama. Terdapat beberapa jenis *array* yaitu *array* 1 dimensi, *array* 2 dimensi, dan *array* 3 dimensi. Suatu elemen bisa dikatakan *array* 1 dimensi jika banyaknya indeks hanya ada satu. *Array* 2 dimensi yaitu *array* yang memiliki dua buah elemen bertipe *array*. Dan *array* 3 dimensi yaitu *array* yang memiliki tiga buah subskrip dan lebih kompleks daripada *array* 2 dimensi. *Array* biasa digunakan untuk penerapan matriks. Pada BAB ini menjadikan kita lebih memahami mengenai *array* serta penerapannya dalam sebuah program terutama dalam bahasa pemrograman

C++ baik itu *array* 1D, 2D, maupun 3D. Dan untuk mendeklarasikan sebuah *array* dalam C++ menggunakan tanda [] (*bracket*). Kita juga lebih memahami bagaimana membuat *flowchart* yang benar menggunakan *array*.

BAB III

METODE *SORTING* DAN *SEARCHING*

3.1 Tujuan

Adapun tujuan dari praktikum ini:

1. Mengerti dan Memahami Metode Pencarian Data.
2. Dapat Menggunakan Metode Pencarian Data dalam Program.
3. Memahami penggunaan metode *Sorting*.

3.2 Dasar Teori

3.2.1 Pencarian Data

Pencarian (*searching*) merupakan tindakan untuk mendapatkan suatu data dalam kumpulan data. Dalam kehidupan sehari-hari, seringkali kita berurusan dengan pencarian, misalnya untuk menemukan nomor telepon seseorang pada buku telepon atau mencari suatu istilah dalam kamus.

Terdapat beragam algoritma pencarian untuk keperluan mencari data. Perlu diketahui, yang dimaksud dengan algoritma pencarian adalah “algoritma yang menerima sebuah *argument* a dan mencoba untuk menemukan sebuah rekaman yang memiliki kunci a”.

Pencarian data dilakukan terhadap data dapat dilakukan terhadap data yang secara keseluruhan berada dalam *memory computer* ataupun terhadap data yang berada dalam penyimpanan eksternal.

3.2.2 Pengertian Sorting

Pengurutan data dalam struktur data sangat penting terutama untuk data yang bertipe data numerik ataupun karakter. Pengurutan dapat dilakukan secara *ascending* (urut naik) dan *descending* (urut turun). Pengurutan (*Sorting*) adalah proses pengurutan data yang sebelumnya disusun secara acak sehingga tersusun secara teratur menurut aturan tertentu.

Sorting juga bias diartikan sebagai proses penyusunan kembali sekumpulan obyek ke dalam urutan tertentu. (Sukrisno & Utami, 2005)

Sorting merupakan suatu proses untuk menyusun kembali himpunan obyek menggunakan aturan tertentu. *Sorting* disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data kedalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen.(Putra, Andreswari, & Susilo, 2015)

3.2.3 Bubble Sort

Sorting merupakan suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. *Sorting* disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data kedalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen. Pada dasarnya ada dua macam urutan yang biasa digunakan dalam suatu proses *sorting*:

1. Urut naik (*ascending*) Mengurutkan dari data yang mempunyai nilai paling kecil sampai paling besar.
2. Urut turun (*descending*) Mengurutkan dari data yang mempunyai nilai paling besar sampai paling kecil.

Metode-metode sorting yang akan dibahas kali ini meliputi:

1. *Bubble sort* (Metode Gelembung)
2. *Insertion Sort* (Metode Penyisipan)
3. *Selection Sort* (Metode Seleksi)
4. *Quick Sort* (Metode *Quick*)

Metode gelembung (*bubble sort*) sering juga disebut dengan metode penukaran (*exchange sort*) adalah metode yang mengurutkan data dengan cara membandingkan masing-masing elemen, kemudian melakukan penukaran bila perlu. Metode ini mudah dipahami dan diprogram, tetapi bila dibandingkan dengan metode lain yang kita pelajari, metode ini merupakan metode yang paling tidak efisien.

Algoritma *Bubble Sort*

1. Membandingkan data ke- i dengan data ke- $(i+1)$ (tepat bersebelahan). Jika tidak sesuai maka tukar (data ke- i = data ke $(i+1)$ dan data ke- $(i+1)$ = data ke- i). Apa maksudnya tidak sesuai? Jika kita menginginkan algoritma menghasilkan data dengan urutan *ascending* (A-Z) kondisi tidak sesuai

adalah data $ke-i > data\ ke-i+1$, dan sebaliknya untuk urutan *descending* (A-Z).

2. Membandingkan data $ke-(i+1)$ dengan data $ke-(i+2)$. Kita melakukan perbandingan ini sampai data terakhir. Contoh: 1 dgn 2; 2 dgn 3; 3 dgn 4; 4 dgn 5 ... ; $n-1$ dgn n .
3. Selesai satu iterasi, adalah jika kita sudah selesai membandingkan antara $(n-1)$ dgn n . Setelah selesai satu iterasi kita lanjutkan lagi *iterasi* berikutnya sesuai dengan aturan ke-1. mulai dari data ke-1 dgn data ke-2, dst.
4. Proses akan berhenti jika tidak ada pertukaran dalam satu *iterasi*.

3.3 Latihan

1. Latihan 1

a. Kode Program

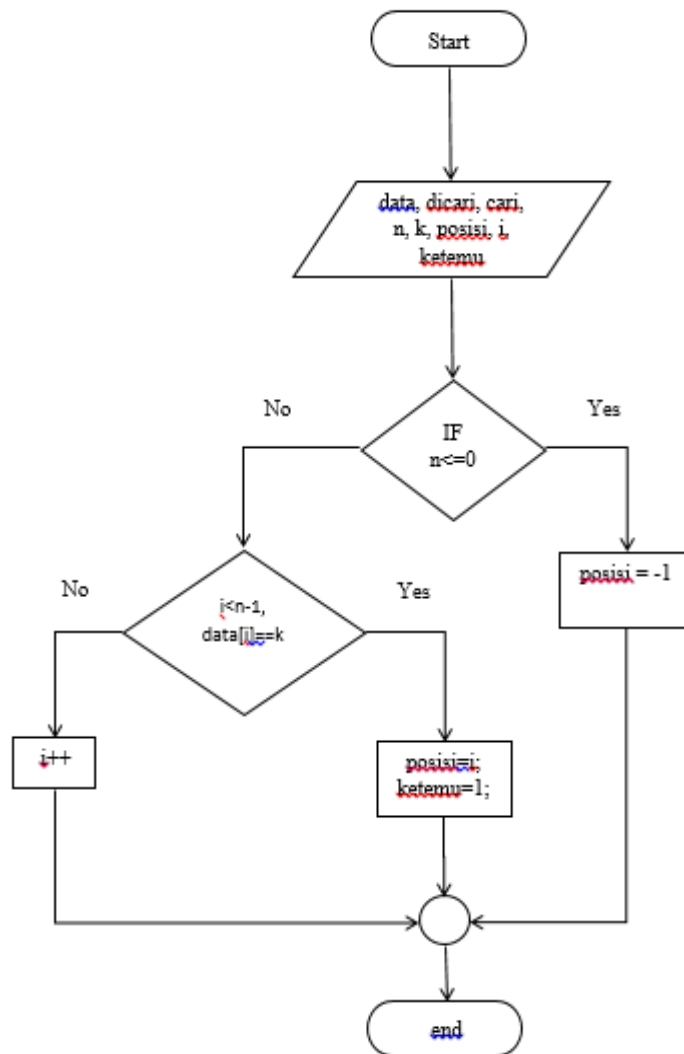
```
#include <iostream>
#include <conio.h>
using namespace std;

int cari (int data[], int n, int k)
{
    int posisi, i, ketemu;
    if (n <= 0)
        posisi=-1;
    else{
        ketemu = 0;
        i=1;
        while ((i<n-1)&&! ketemu)
            if (data[i] == k)
            {
                posisi=i;
                ketemu=1;
            }
        else
            i++;
        if (!ketemu)
```

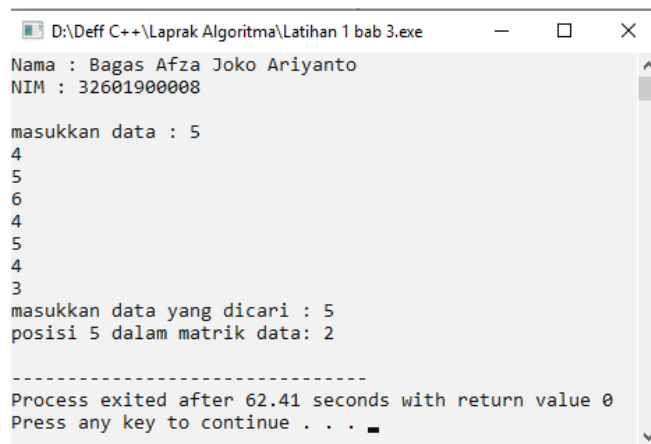
```

        posisi = -1;
    }
    return posisi;
}
int main()
{
    int data [8];
    int dicari;
    cout << "Nama : Bagas Afza Joko Ariyanto";
    cout << "\nNIM : 32601900008\n\n";
    cout <<"masukkan data : ";
    for(int a=0; a<8;a++)
    {
        cin >> data[a];
    }
    cout <<"masukkan data yang dicari : ";
    cin >> dicari;
    cout <<"posisi " <<dicari<<" dalam matrik data: "
    <<(cari(data, 8, dicari))<<"\n";
    getch();
}

```

b. *Flowchat*Gambar 3.1 *Flowchart* Latihan 1

c. *Output* :



```

D:\Deff C++\Laprak Algoritma\Latihan 1 bab 3.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

masukkan data : 5
4
5
6
4
5
4
3
masukkan data yang dicari : 5
posisi 5 dalam matrik data: 2

-----
Process exited after 62.41 seconds with return value 0
Press any key to continue . . .

```

Gambar 3.2 *Output* Latihan 1

d. *Penjelasan*

Pada gambar 3.2 merupakan hasil dari program diatas. Program diatas adalah program yang menampilkan Masukkan data. Program tersebut memerintahkan agar memasukkan data selama 8 kali. Kemudian program akan menampilkan masukkan data yang dicari agar dapat mencari posisi dalam matrik data.

Header yang digunakan dalam program tersebut adalah *iostream* dan *conio.h*. *iostream* adalah *header* yang menjalankan perintah *cin* dan *cout*. Sedangkan *conio.h* merupakan *header* untuk membersihkan tampilan, menahan tampilan

2. **Latihan 2**

a. *Kode Program*

```

#include <iostream>
using namespace std;
int main()
{
    int i, j, n, data[10],simpan, k;
    cout << "Nama : Bagas Afza Joko Ariyanto";
    cout << "\nNIM : 32601900008\n\n";
    cout<<"masukkan banyak data= ";cin>>n;
    for(i=1;i<=n;i++)
    {

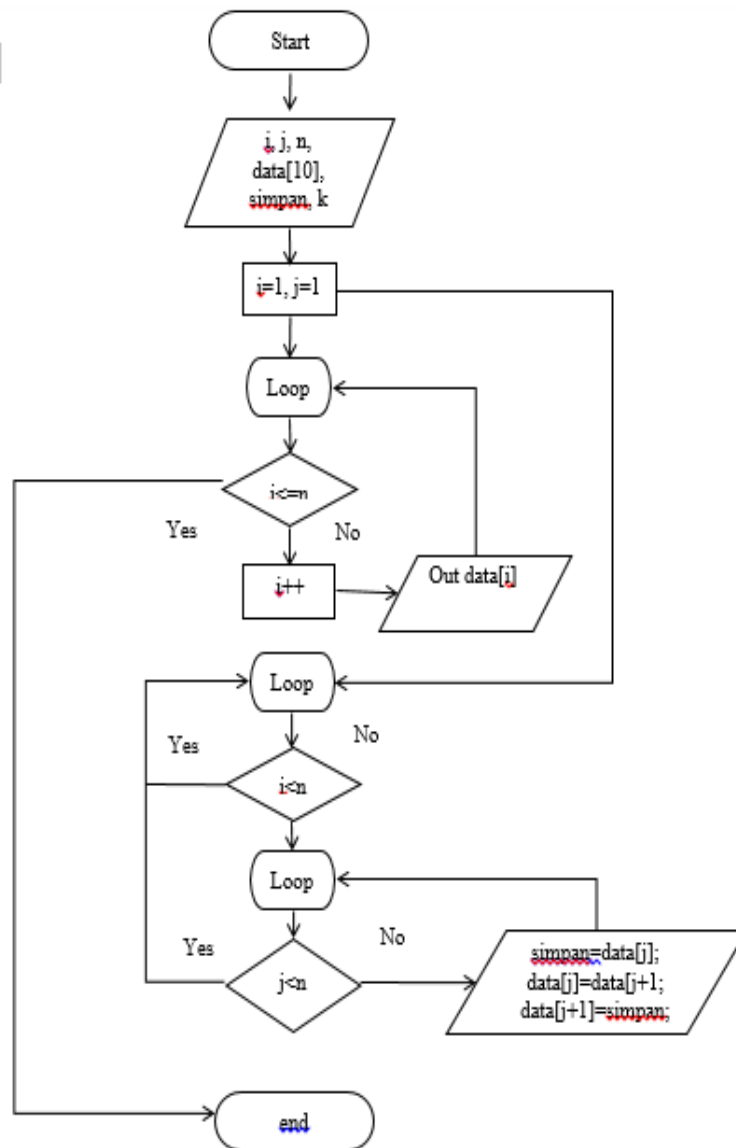
```

```

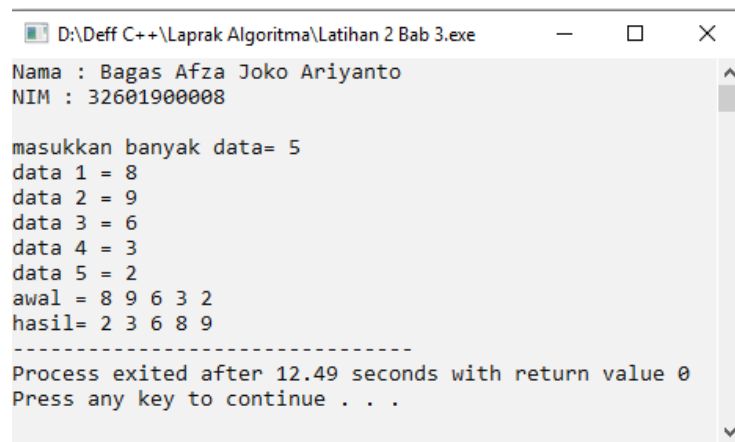
        cout<<"data "<<i<<" = ";cin>>data[i];
    }
    cout<<"awal = ";
    for (i=1;i<=n;i++)
        cout<<data[i]<<" ";
    cout<<endl;

    for (i=1;i<n;i++)
    {
        for (j=1;j<n;j++)
        {
            if (data[j]>data[j+1])
            {
                simpan=data[j];
                data[j]=data[j+1];
                data[j+1]=simpan;
            }
        }
    }
    cout<<"hasil= ";
    for (i=1;i<=n;i++)
        cout<<data[i]<<" ";
    return 0;
}

```

b. *Flowchat*Gambar 3.3 *Flowchart* Latihan 2

c. Output



```

D:\Deff C++\Laprak Algoritma\Latihan 2 Bab 3.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

masukkan banyak data= 5
data 1 = 8
data 2 = 9
data 3 = 6
data 4 = 3
data 5 = 2
awal = 8 9 6 3 2
hasil= 2 3 6 8 9
-----
Process exited after 12.49 seconds with return value 0
Press any key to continue . . .

```

Gambar 3.4 Output Latihan 2

d. Penjelasan

Pada gambar 3.4 merupakan hasil dari program di atas. Program diatas merupakan program yang menampilkan masukkan banyak data. Untuk mencari hasil dari *sorting* yang dilakukan pada program tersebut.

Header yang digunakan dalam program tersebut adalah *iostream*. *Iostream* merupakan *header* yang menjalankan perintah *cin* dan *cout*. Dan *main* merupakan program yang dijalankan paling utama.

3.4 Tugas

1. Tugas 1.

Ubahlah *script* di bawah ini menggunakan bahasa C.

```

#include <iostream>
using namespace std;
int data[10],data2[10];
int n;
void tukar(int a,int b) {
    int t;
    t = data[b];
    data[b] = data[a];
    data[a] = t;
}

```



```

    }

void Input() {
    cout<<"Masukkan jumlah data = ";cin>>n;
    cout<<"-----"<<endl;

    for(int i=0;i<n;i++){
        cout<<"Masukkan data ke-"<<(i+1)<<" = ";cin>>data[i];
        data2[i] = data[i];
    }
    cout<<endl;
}

void Tampil() {
    for(int i=0;i<n;i++){
        cout<<data[i]<<" ";
    }
    cout<<endl;
}

void bubble_sort() {
    for(int i=1;i<n;i++){

        for(int j=n-1;j>=i;j--){
            if(data[j]<data[j-1]) tukar(j,j-1);

        }Tampil();
        cout<<endl;
    }
}

main() {
    cout<<"* Bubble Sort *"<<endl;
    Input();
    cout<<"Proses Bubble Sort,,,,,,,,,"<<endl;
    cout<<"-----"<<endl;
    Tampil();
    bubble_sort();
    cout<<"-----"<<endl;
}

```

Jawaban :

a. Kode Program

```
#include <stdio.h>
int data[10], data2[10], n;
void tukar(int a,int b) {
    int t;
    t = data[b];
    data[b] = data[a];
    data[a] = t;
}

void input(){
    printf("Masukkan jumlah data : "); scanf("%d", &n);
    printf("-----\n");

    int i;
    for(i=0; i<n; i++){
        printf("Masukkan data ke- %d = ", i+1);
        scanf("%d", &data[i]); data2[i] = data[i];
    }
    printf("\n");
}

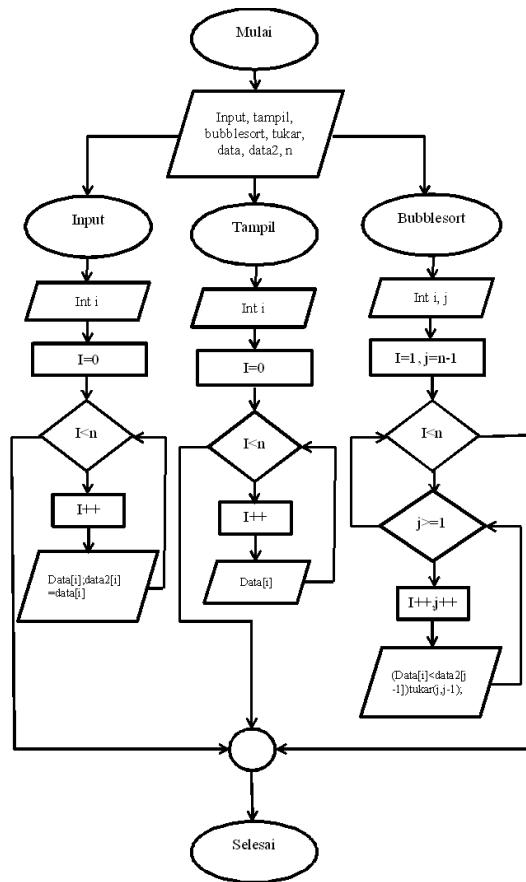
void tampil(){
    int i;
    for(i=0; i<n; i++){
        printf("%d ", data[i]);
    }
    printf("\n");
}

void bubblesort(){
    int i, j;
    for(i=1; i<n; i++){
        for(j=n-1; j>=i; j--){
            if(data[j]<data[j-1]){
                tukar(j,j-1);
            }
        }
    }
}
```

```
        }

    }
}
tampil();
}

main(){
    printf("Nama : Bagas Afza Joko Ariyanto");
    printf("\nNIM : 32601900008\n\n");
    printf("* Bubble Sort *");
    printf("\n");
    input();
    tampil();
    bubblesort ();
}
```

b. *Flowchart*

Gambar 3.5 Flowchart Tugas 1

c. *Output*

```

D:\Deff C++\Laprak Algoritma\Tugas 1 BAB 3.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

* Bubble Sort *
Masukkan jumlah data : 4
-----
Masukkan data ke- 1 = 2
Masukkan data ke- 2 = 4
Masukkan data ke- 3 = 3
Masukkan data ke- 4 = 1

2 4 3 1
1 2 3 4

-----
Process exited after 5.763 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 3.6 Output Tugas 1

d. Penjelasan

Pada gambar 3.6 merupakan hasil dari program di atas. Program diatas merupakan program yang menggunakan bahasa C. Program diatas menampilkan **Bubble sort** kemudiam masukkan jumlah data. Program ini akan menampilkan nilai yang belumurut kemudian akan mengurutkan nilai dari terbesar ke terkecil. Program tersebut dijalankan dengan maksud untuk mengurutkan data dengan menggunakan *Bubble sort*. Program tersebut meminta user untuk memasukkan atau menginputkan banayak data terlebih dahulu. Setelah menginputkan banyak data maka sdengan pengulangan, kata “masukkan data” akan diuangi sebanyak data yang diinginkan kemudian user akan diminta menginputkan data sebanyak data yang dimasukkan tadi.

Setelah data terkumpul maka sorting akan dilakukan dengan metode *bubble sort* yang mengurutkan data daroi terbesar ke terkecil.

Header yang digunakan dalam program tersebut adalah *stdio.h*. *Stdio.h* merupakan header yang menjalankan perintah *printf* dan *scanf*.

2. Tugas 2.

Jalankan program akan menghasilkan tampilan gambar seperti berikut:

```
#include <iostream>
using namespace std;
int cari (int data[], int n,int k,int m)  {
int posisi,i,ketemu,pencacah=0;
if (n <= 0)
posisi=-1;
else{
ketemu = 0;
i=1;
while ((i<n+1)&&! ketemu)
if (data[i] == k)          {
pencacah++;
cout<<"Pencacah " <<pencacah <<endl;
if (pencacah != m)          {
cout<<"Ketemu"<<pencacah << endl;
posisi=i;
ketemu = 1;
}
Else
I--;
}
Else
i++;
if (!ketemu)
}
return posisi;
}
i nt main()  {
int data [9]={10,9,4,6,3,4,2,5,0};
int dicari;
int ke;
dicari=4;
ke =2;
cout<<"Posisi " <<dicari<<" yang ke-"<<ke<<" dalam larik
data: " <<(cari(data,8,dicari,ke))<<"\n";
```

```
ke =1;
cout<<"Posisi " <<dicari<<" yang ke-"<<ke<<" dalam larik
data: " <<(cari(data,8,dicari,ke))<<"\n";
dicari = 0;
ke=1;
cout<<"Posisi " <<dicari<<" yang ke-"<<ke<<" dalam larik
data: " <<(cari(data,9,dicari,ke))<<"\n";
return 0;
}
```

Jawaban :

a. Kode *Program*

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

int cari (int data[], int n, int k, int m){
    int posisi,i,ketemu,pencacah=0;
    if(n<=0){
        posisi=-1;
    }else{
        ketemu=0;
        i=1;
        while((i<n+1)&&! ketemu)
            if(data[i]==k){
                pencacah++;
                cout<<"pencacah "<<pencacah<<endl;
                if(pencacah !=m){
                    cout<< "ketemu
" <<pencacah<<endl;

                    posisi=i;
                    ketemu=1;
                }else{
                    i--;
                }
            }else{
                i++;
            }
        if(!ketemu){
            return posisi;
        }
    }
}

int main(){
    int data[9]={10,4,9,6,3,4,2,5,0};
    int dicari;
    int ke;
```



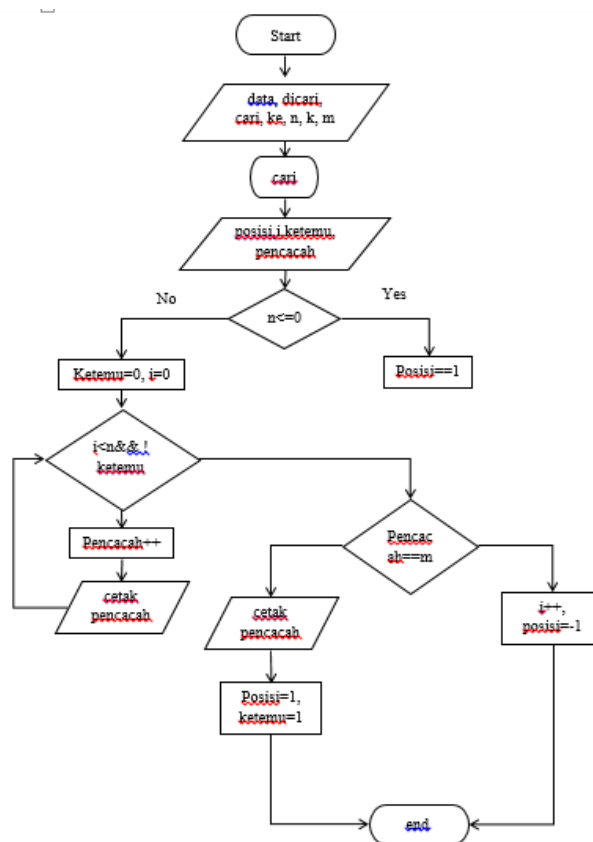
```

cout<< "Nama : Bagus Afza Joko Ariyanto";
cout<< "\nNIM : 32601900008\n\n";

dicari=4;
ke=2;
cout<< "posisi "<<dicari<< " yang ke- "<<ke<<" dalam
larik data : "<<(cari(data,4,dicari,ke))<<"\n";
ke=1;
cout<< "posisi "<<dicari<< " yang ke- "<<ke<<" dalam
larik data : "<<(cari(data,1,dicari,ke))<<"\n";
dicari=0;
ke=1;
cout<< "posisi "<<dicari<< " yang ke- "<<ke<<" dalam
larik data : "<<(cari(data,2,dicari,ke))<<"\n";
return 0;
}

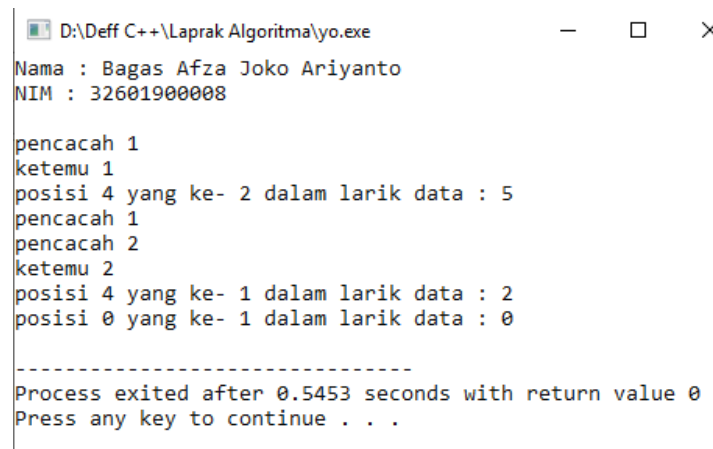
```

b. Flowchart



Gambar 3.7 Flowchart Tugas 2

c. Output



```

D:\Deff C++\Laprak Algoritma\yo.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

pencacah 1
ketemu 1
posisi 4 yang ke- 2 dalam larik data : 5
pencacah 1
pencacah 2
ketemu 2
posisi 4 yang ke- 1 dalam larik data : 2
posisi 0 yang ke- 1 dalam larik data : 0

-----
Process exited after 0.5453 seconds with return value 0
Press any key to continue . . .

```

Gambar 3. 8 Output Tugas 2

d. Penjelasan

Pada gambar 3.8 merupakan hasil dari program diatas. Program diatas merupakan program yang menampilkan pencacah 1, pencacah 2, poisi 4 yang ke 2 dalam larik data, pencacah 1, ketemu 1, posisi 4 yang ke-1 dalam larik data, posisi 0 yang ke-1 dalam larik data.

Kemudian pencacah akan menampilkan pencacah berada dalam posisi mana pada data yang ada. Pencacah ini ada dua kali yaitu pencacah satu dan 2. Sehingga output yang dihasilkan pun ada dua pencacah. Header yang digunakan dalam program tersebut adalah `iostream`. `iostream` merupakan header untuk menjalankan perintah `cin` dan `cout`. `Main` merupakan perintah pertama yang dijalankan oleh program tersebut.

3. Tugas 3.

Ubahlah *Script* pada latihan 1 dengan menggunakan metode *Insertion Sort* secara *Decending*.

a. Kode Program

```

#include<iostream>
using namespace std;
int A[]={2,3,4,6,7,8,9,1,5,0};
int n=10;
void SortingDesc(){

```

```

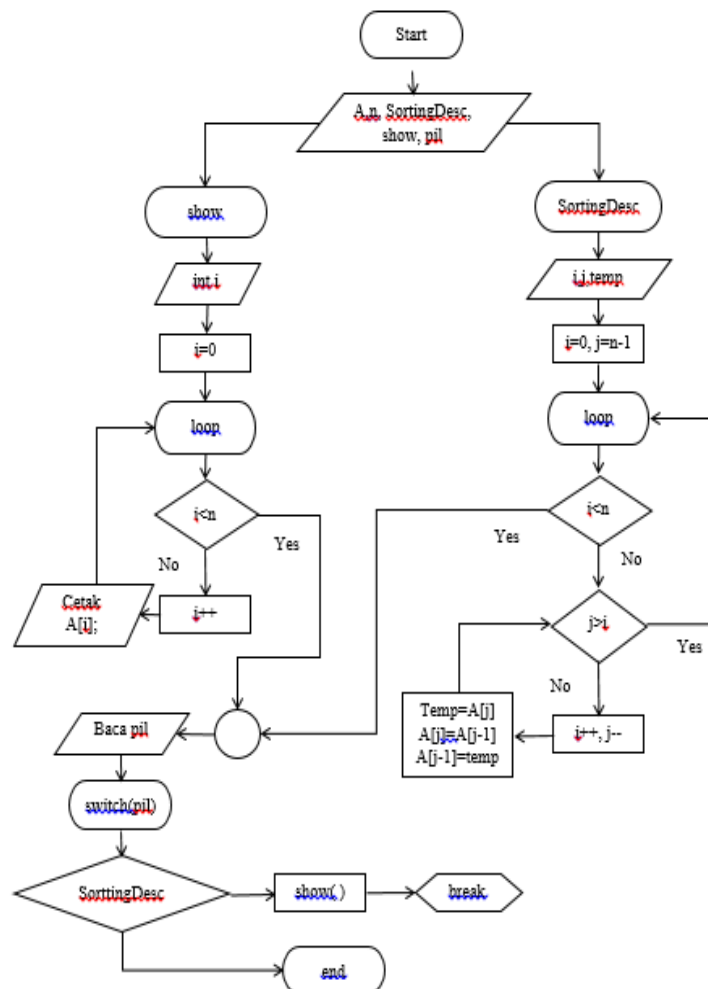
int i, j, temp;
for(i=0; i<n; i++){
    for(j=n-1; j>i; j--){
        if(A[j]>A[j-1]){
            temp = A[j];
            A[j] = A[j-1];
            A[j-1] = temp;
        }
    }
}

void show(){
    int i;
    for(i=0; i<n; i++){
        cout << A[i]<<" ";
    }
}

int main(){
    int pil;
    cout<< "Nama : Bagas Afza Joko Ariyanto\n";
    cout<< "NIM :32601900008\n\n";
    cout<< "Program Shorting Array\n"<<"\n";
    show();
    cout<< "\nData Decensing Tekan 1 dan Enter\n\n";
    cout<< "Input Pilihan Anda : ";
    cin>> pil;
    switch(pil){
        case 1 :
            SortingDesc();
            cout<< "\nData Setelah di
Urutkan\n\nDecending : ";
            show();
            break;
    }
}

```

b. Flowchart



Gambar 3.9 Flowchart Tugas 3

c. Output

```

D:\Deff C++\Laprak Algoritma\Tugas 3 BAB 3.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Program Shorting Array

2 3 4 6 7 8 9 1 5 0
Data Decensing Tekan 1 dan Enter

Input Pilihan Anda : 1

Data Setelah di Urutkan

Decending : 9 8 7 6 5 4 3 2 1 0
-----
Process exited after 4.078 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 3.10 Output Tugas 3

d. Penjelasan

Pada gambar 3.8 merupakan hasil dari program diatas. Program diatas merupakan program yang menghasilkan tampilan Program Shorting Array. Yang menghasilkan urutan nomor secara descending dengan menggunakan intersection sort. Pada program ini akan menampilkan data setelah diurutkan dari bilangan terbesar kemudian seterusnya sampai bilangan terkecil. Pada program ini user akan melihat tampilan angka yang belum urut. Untuk mengurutkan angka tersebut secara descending yaitu dari besar ke kecil dengan memasukkan angka satu kemudian menekan enter. Program akan mengurutkan angka tersebut dengan metode intersection sort.

Program diatas adalah program yang menggunakan bahasa C++. Dengan menggunakan header iostream. Iostream adalah header yang digunakan untuk menjalankan perintah cin dan cout. Dan main adalah yang pertama dijalankan oleh iostream.

4. Tugas 4.

Buatlah *program* pengurutan data dengan menggunakan semua metode pengurutan untuk mengurutkan angka secara *ascending* dan *descending*.

1. *Descending*.

a. Kode Program

```
#include<iostream>
#include<conio.h>
#include<iomanip>
#include <time.h>
#include <windows.h>
#define MAX 1
using namespace std;
int A[]={2,3,4,6,7,8,9,1,5,0};
int n=10;
void SortingDesc()
{
```

```

int i,j,temp;
for (i=0;i<n;i++)
{
for (j=n-1;j>i;j--)
{
if (A[j]>A[j-1])
{
temp = A[j];
A[j] = A[j-1];
A[j-1] = temp;
}
}
}
void SortingAscen ()
{
int i,j,temp;
for (i=0;i<n;i++)
{
for (j=n;j>i;j--)
{

if (A[j]<A[j-1])
{
temp = A[j];
A[j] = A[j-1];
A[j-1] = temp;
}
}
}
}
void show()
{
int i;
for (i=0;i<n;i++)
{
cout<<A[i]<<" ";
}
}

```

```

}
int bubblesort();
int selectionsort();
int insertionsort();
int pilihan, hitung, i;
char ulang, pilih;
char nx[10];
int data[MAX];
int data1[100];
int temp;
int pil;
int cek(char str[])
{
    int i, panjang;
    panjang=strlen(str);
    for (i=0; i<panjang; i++)
    {
        if ((i==0)&&(str[i]=='-'))
            i++;
        if (!isdigit(str[i]))
        {

            cout<<"input salah ! ulangi input !\n";
            return 0;
        }
    }
    return 1;
}
int main()
{
    int pil;
    awal:
    system("cls");
    do
    {
        system("cls");
        int pil;
        int n;

```

```

for (hitung=0; hitung<8; hitung++)
data1[hitung]=data[hitung];
cout<<"Nama : Bagas Afza Joko Ariyanto\n";
cout<<"NIM : 32601900008\n\n";
cout<<"Program full metode sorting ";
cout<<endl<<endl;
cout<<"Masukan banyak data: ";
cin>>nx;
cout<<endl;
}
while (!cek(nx));
n=atoi(nx);
int data[100];
srand(time(NULL));
int i;
for (i=0; i<n; i++)
{
data[i]=rand()%n;
cout<<data[i]<<" ";
}
cout<<endl;
{

for (hitung=0; hitung<n; hitung++)
data1[hitung]= data[hitung];
cout<<"\n Metode sorting \n"<<endl;
cout<<"1. Bubble sort\n"<<endl;
cout<<"2. selection sort\n"<<endl;
cout<<"3. insertion sort\n"<<endl;
cout<<"masukkan pilihan :";
cin>>pil;
cout<<"-----\n";
cout<<"pilihan pengurutan Ascending Descending[1/2]?
";
cin>>pil;
switch(pil)
{
case 1:

```



```

SortingDesc();
cout<<"\nData Setelah di urutkan\n\nDescending :";
show();
break;
case 2:
SortingAscen();
cout<<"\nData Setelah di urutkan\n\nAscending :";
show();
break;
} cout<<endl;
}
cout<<endl;
}
int bubblesort ()
{
cout<<"=====\n";
for (int i=0; i<n-1; i++)
for (int ii=0; ii<n-1; ii++)
if (data1[ii]>=data1[ii+1])
{

temp=data1[ii];
data1[ii]=data1[ii+1];
data1[ii+1]=temp;
}
if (pilih=='a')
{
cout<<"menjalankan metode bubblesort dengan pengurutan
Ascending ";
cout<<"\n\nHasil pengurutan : \n\n";
for (int iii=0; iii<n; iii++)
cout<<data1[iii]<<" ";
}
if (pilih=='d')
{
cout<<"menjalankan metode bubblesort dengan pengurutan
Descending ";
cout<<"\n\nHasil pengurutan : \n\n";

```

```

for (int iii=0; iii<n-1; iii--)
cout<<data1[iii]<<" ";
}
cout<<"\n";
}
int selectionsort()
{
cout<<"=====\n";
int kecil;
for (int i=0; i<n; i++)
{
kecil=i;
for (int j=kecil+1; j<n; j++)
{
if (data1[kecil]>data1[j])
{
kecil=j;
}
}
temp=data1[i];
data1[i]=data1[kecil];

data1[kecil]=temp;
}
if (pilih=='a')
{
cout<<"menjalankan metode bubblesort dengan pengurutan
Descending ";
cout<<"\n\nHasil pengurutan : \n\n";
for (int iii=0; iii<n; iii++)
cout<<data1[iii]<<" ";
}
if (pilih=='d')
{
cout<<"menjalankan metode bubblesort dengan pengurutan
Descending ";
cout<<"\n\nHasil pengurutan : \n\n";
for (int iii=0; iii<n-1; iii--)

```

```

cout<<data1[iii]<<" ";
}
cout<<"\n";
}
int insertionsort()
{
cout<<"=====\n";
int gap=n/2;
do
{
int swap;
do
{
swap=0;
for (int i=0; i<n-gap; i++)
if (data1[i]>data1[i+gap])
{
int t=data1[i];
data1[i]=data1[i+gap];
data1[i+gap]=t;
swap=1;

}
}
while (swap);
while (gap=gap/2);
if (pilih=='a')
{
cout<<"menjalankan metode bubblesort dengan pengurutan
Descending ";
cout<<"\n\nHasil pengurutan : \n\n";
for (int i=0; i<n; i++)
cout<<data1[i]<<" ";
}
if (pilih=='d')
{

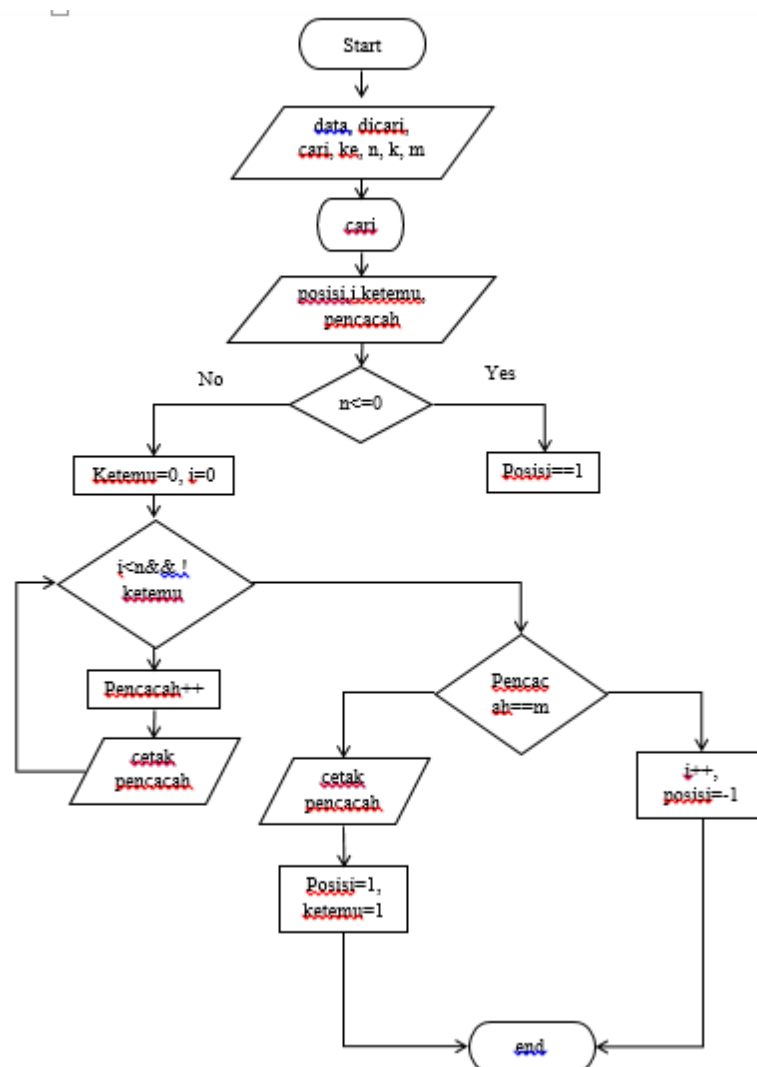
```

```

cout<<"menjalankan metode bubblesort dengan pengurutan
Descending ";
cout<<"\n\nHasil pengurutan : \n\n";
for (int i=0; i<n-1; i--)
cout<<data1[i]<<" ";
}
cout<<endl;
return 0;
}

```

b. *Flowchart*



Gambar 3 11 *Flowchart Descending Tugas 4*

c. Output

```

D:\Deff C++\Laprak Algoritma\TUGAS BAB 3 NO 4(1)...
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Program full metode sorting

Masukan banyak data: 5

0 1 4 2 0

Metode sorting

1. Bubble sort
2. selection sort
3. insertion sort

masukkan pilihan :1
-----
pilihan pengurutan Ascending Descending[1/2]? 2

Data Setelah di urutkan

Ascending :2 3 4 6 7

-----
Process exited after 165.5 seconds with return value 0
Press any key to continue . . . █

```

Gambar 3.12 *Output Bubble Sort* Tugas 4

```

D:\Deff C++\Laprak Algoritma\TUGAS BAB 3 NO 4(1)....
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Program full metode sorting

Masukan banyak data: 5

1 0 2 0 4

Metode sorting

1. Bubble sort
2. selection sort
3. insertion sort

masukkan pilihan :2
-----
pilihan pengurutan Ascending Descending[1/2]? 2

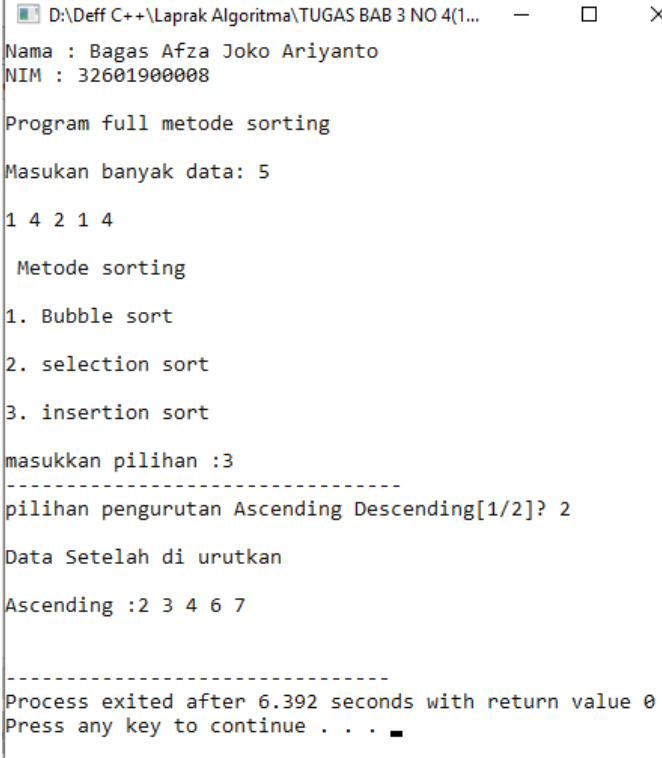
Data Setelah di urutkan

Ascending :2 3 4 6 7

-----
Process exited after 5.775 seconds with return value 0
Press any key to continue . . . █

```

Gambar 3.13 *Output Selection Sort* Tugas 4



```

D:\Deff C++\Laprak Algoritma\TUGAS BAB 3 NO 4(1...  -  □  ×
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Program full metode sorting

Masukan banyak data: 5

1 4 2 1 4

Metode sorting

1. Bubble sort
2. selection sort
3. insertion sort

masukkan pilihan :3
-----
pilihan pengurutan Ascending Descending[1/2]? 2

Data Setelah di urutkan

Ascending :2 3 4 6 7

-----
Process exited after 6.392 seconds with return value 0
Press any key to continue . . . █

```

Gambar 3.14 *Output Insertion Sort Tugas 4*

d. Penjelasan

Pada gambar 3.10, 3.11, 3.12 merupakan hasil dari program yang dijalankan diatas. Program diatas adalah program yang menggunakan bahasa C++. Program diatas merupakan program yang menampilkan data sebelum diurutkan dan memberikan perintah untuk menekan 1 dan enter untuk mencari data yang telah diurutkan. Awalnya user akan diminta untuk menginputkan banyaknya data yang akan digunakan untuk metode sorting. Kemudian akan ada pilihan metode sorting yaitu 1 untuk bubble sorting, 3 untuk intersection sorting, 2 untuk selection sorting. Kemudian setelah memilih metode pengurutan maka akan ada pilihan untuk menampilkan progra tersebut secara ascending atau descending. Header yang digunakan dalam program tersebut adalah iostream, conio.h, time.h, windows.h dan define MAX 1.

2. Ascending

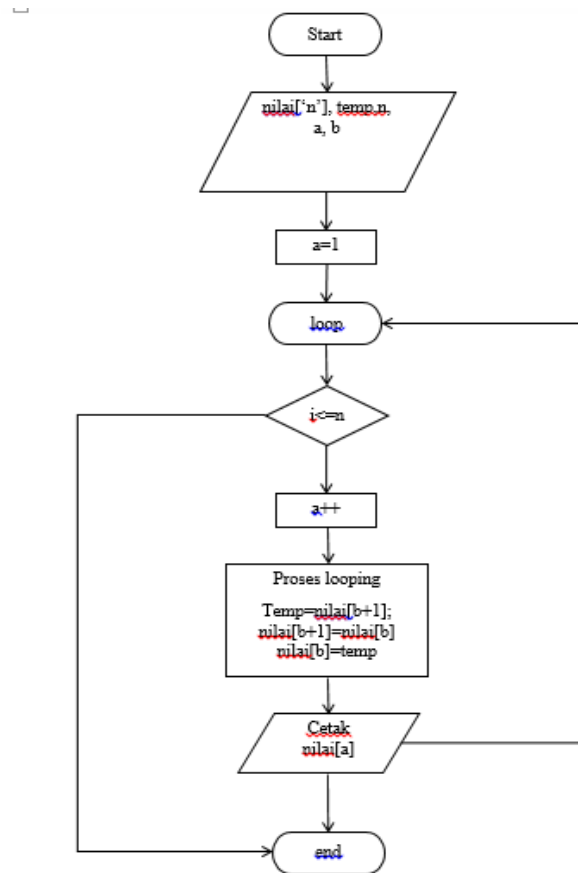
a. Kode Program

```
#include <iostream>
using namespace std;
main ()
{
    cout<<"Nama : Bagas Afza Joko Ariyanto\n";
    cout<<"NIM : 32601900008\n\n";

    int nilai ['n'], temp, n;
    cout<< "banyak data: ";
    cin>> n; cout<<endl;
    for(int a=1; a<=n; a++){
        cout<< "nilai["<<a<<"] : "; cin>>nilai [a];
    }
    cout<<"\n\n";
    cout<<"data sebelum diurutkan"<<endl;
    for(int a=1; a<=n; a++){
        cout<<nilai[a]<<" ";
    }

    for (int a=n-1; a>=1; a--){
        for (int b=1; b<=a; b++){
            if (nilai[b]>nilai[b+1]){
                temp=nilai [b+1];
                nilai[b+1]=nilai[b];
                nilai[b]=temp;
            }
        }
    }

    cout<<"\n\nData Setelah Diurutkan (Ascending)"<<endl;
    for(int a=1; a<=n; a++){
        cout<<nilai[a]<<" ";
    }
}
```

b. *Flowchart*

Gambar 3.15 Flowchart Ascending Tugas 4

c. *Output*

```

D:\Deff C++\Laprak Algoritma\TUGAS BAB 3 NO 4(2...
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

banyak data: 4

nilai[1]: 1
nilai[2]: 3
nilai[3]: 4
nilai[4]: 5

data sebelum diurutkan
1 3 4 5

Data Setelah Diurutkan (Ascending)
1 3 4 5

-----
Process exited after 9.721 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 3.16 Output Ascending Tugas 4

d. Penjelasan

Pada gambar 3.14 merupakan hasil dari program yang dijalankan diatas. Program diatas adalah program yang menggunakan bahasa C++. Program diatas merupakan program yang menampilkan data sebelum diurutkan dan memberikan perintah untuk menekan 1 dan enter untuk mencari data yang telah diurutkan. Data ini diurutkan dari nilai terbesar ke nilai yang paling terkecil (Ascending). Pengguna akan diminta memasukkan banyaknya data yang akan digunakan untuk pengurutan data. Pengguna selanjutnya memasukkan nilai sebanyak data yang dimasukkan sebelumnya. Kemudian akan disusun data itu tapi belum urut. Setelah itu program akan mengurutkan angka itu secara otomatis dari terkecil ke terbesar. Header yang digunakan dalam program tersebut adalah `iostream`, `conio.h`, `time.h`, `windows.h` dan `define MAX 1`.

5. Tugas 5.

Ubahlah *script* dibawah ini dengan menggunakan bahasa C++.

```
#include <stdio.h>

void tampilkan_larik (int data[],int n) {
    int i;
    for(i=0;i< n ;i++)
        printf("%d ",data[i]);
    printf("\n");
}

int partisi(int data[],int p,int r){
    int pivot= data[r];
    int i = p-1;
    int j;
    for (j=p;j<r;j++){
        if (data[j]<= pivot) {
            i++;
            int tmp = data[i];
            data[i]= data[j];
            data[j]= tmp;
        }
    }
}
```

```

    }
}
int tmp = data[i+1];
data[i+1]= data[r];
data[r]= tmp;
return i+1;
}
void quick_sort(int data[],int p,int r) {
    int q ;
    if (p<r){
        q = partisi(data,p,r);
        quick_sort(data,p,q-1);
        quick_sort(data,q+1,r);
    }
}

int main(){
    int jum_data = 9;
    int i;
    int data[]={25,57,48,37,12,92,80,33,1};
    quick_sort(data,0,jum_data-1);
    printf("hasil pengurutan : \n");
    tampilkan_larik(data,jum_data);
    return 0;
}

```

Jawaban:

a. Kode Program

```

#include <iostream>
#include <conio.h>
#define max 20
using namespace std ;

void quick_sort(int data[max], int l, int b) {
    int a;
    int up,down;
    int temp;

    if (l>=b)        return;

```

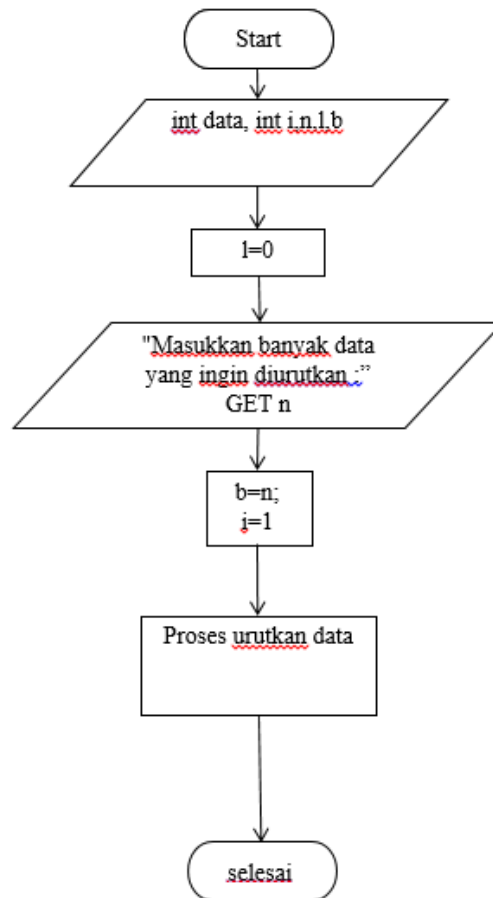
```

a=data[l];
up=b;
down=l;

while (down < up)    {
    while (data[down] <= a)        down++;
    while (data[up]>a)            up--;
    if(down<up)                {
        temp=data[down];
        data[down]=data[up];
        data[up]=temp;
    }
}
data[l]=data[up];
data[up]=a;
quick_sort(data,l,up-1);
quick_sort(data,up+1,b);
}

int main() {
    int data[max];
    int i,n,l,b;
    l=0;
    cout<<"Masukkan banyak data yang ingin diurutkan: ";
    cin>>n;
    b=n;
    cout<<"Masukkan data-datanya: \n\n";
    for(i=1;i<=n;i++)    {
        cout<<"\tdata ke- "<<i<<" : "; cin>>data[i];
    }
    quick_sort(data,l,b);
    cout<<"\nHasil pengurutan data: ";
    for(i=0; i<n;i++)
        cout<<" "<<data[i];
    getch();
}

```

b. *Flowchart*Gambar 3. 17 *Flowchart* Tugas 5c. *Output*

```

D:\Deff C++\Laprak Algoritma\Tugas 5 BAB 3.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

Masukkan banyak data yang ingin diurutkan: 4
Masukkan data-datanya:

data ke- 1 : 1
data ke- 2 : 3
data ke- 3 : 5
data ke- 4 : 2

Hasil pengurutan data: 1 2 3 5
-----
Process exited after 7.258 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 3.18 *Output* Tugas 5

d. Penjelasan

Program diatas merupakan program yang menggunakan bahasa c++ yang menampilkan data seelum dan sesudah diurutkan. Dan memberikan perintah untuk memassukkan data agar mendapatkan hasil pengurutan yang benar. Pengguna akan memasukkan banyak data yang akan digunakan. Kemudian data yang sama akan dimasukkan oleh pengguna. Data inilah yang akan dignakan sebagai acuan untuk melakukan sorting. Sorting ini kan menjalankan sorting dari bilangan terkecil ke terbesar atau biasa disebut descending.

Header yang digunakan pada program tersebut adalah iostream dan conio.h. iostream merupakan header yang melakukan perintah cin dan cout.

3.5 Kesimpulan

Sorting merupakan suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. Sorting disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data kedalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen. Pada dasarnya ada dua macam urutan yang biasa digunakan dalam suatu proses *sorting*:

1. Urut naik (*ascending*) mengurutkan dari data yang mempunyai nilai paling kecil sampai paling besar.
2. Urut turun (*descending*) Mengurutkan dari data yang mempunyai nilai paling besar sampai paling kecil.

Metode-metode sorting yang akan dibahas kali ini meliputi:

1. *Bubble sort*(Metode Gelembung)
2. *Selection Sort* (Metode Seleksi)
3. *Quick Sort* (Metode Quick)
4. *Quick Sort* (Metode *Quick*)

BAB IV

REKURSI

4.1 Tujuan

Adapun tujuan dilakukannya praktikum ini:

1. Memahami mengenai konsep rekursif
2. Mampu memecahkan permasalahan dengan konsep rekursif

4.2 Dasar Teori

4.2.1 Pengertian Rekursi

Rekursi adalah suatu kemampuan subrutin untuk memanggil dirinya sendiri. Adapun suatu subrutin yang memanggil dirinya seperti itu dinamakan subrutin rekursi. Pada beberapa persoalan, kemampuan seperti itu sangat berguna karena mempermudah solusi. Namun demikian rekursi juga memiliki kelemahan, yakni memungkinkan terjadinya *overflow* pada *stack* (*stack* tidak lagi mampu menangani permintaan pemanggilan subrutin karena kehabisan memori). Itu sebabnya harus ada jaminan bahwa proses rekursi akan berhenti pada suatu waktu tertentu, yang menyebabkan pemanggilan fungsi berakhir.

4.2.2 Fibonacci

Fibonacci merupakan sebuah pola bilangan yang didapatkan dari penjumlahan dua bilangan sebelumnya pada sebuah deret tersebut.

Berikut contoh program dari fibonacci

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

int fibonacci(int n) {
    if (n == 0 || n == 1) {
        return n;
    } else {
        return (fibonacci(n-1) + fibonacci(n-2));
    }
}
```

```

    }
}

int main() {

    int n, i, j = 0;

    cout << "Masukkan batas jumlah bilangan fibonacci: ";
    cin >> n;

    cout << "Hasil bilangan fibonacci: \n";
    for (i = 1; i <= n; i++){
        cout << "fibonacci ke-"<< i <<": " << fibonacci(j) << endl;
        j++;
    }
    return 0;
}

```

Berikut adalah penjelasan source code programnya :

- Baris 5: Membuat fungsi bernama fibonacci yang bertipe integer dan menerima masukkan sebuah integer yang diberi nama variable n.
- Baris 6-7: Cek kondisi jika nilai dari variable n adalah 0 atau 1 maka akan mengembalikan nilai n.
- Baris 8-9: Jika variable n bernilai selain 0 dan 1 maka akan melakukan operasi rekursif dimana hasil operasi fibonacci n-1 ditambah dengan hasil operasi fibonacci n-2.
- Baris 15: mendeklarasikan variable n, i, dan j dimana variable n digunakan untuk batas jumlah bilangan fibonacci yang akan ditampilkan, variable i digunakan untuk perulangan dan variable j digunakan untuk memasukkan nilai ke fungsi fibonacci secara berurutan.
- Baris 17-18: menginputkan nilai kedalam variable n.
- Baris 20-14: melakukan perulangan sebanyak nilai n dan menginputkan nilai j kedalam fungsi fibonacci serta menampilkan hasil operasi fibonacci. (S Safwandi, 2014)

4.3 Latihan

1. Latihan 1

Tuliskan algoritma untuk menghitung nilai suatu faktorial dan beserta programnya!

a. Algoritma

SUBROUTIN faktor

JIKA $n = 0$ ATAU 1 MAKA

NILAI-BALIK 1 SEBALIKNYA

NILAI-BALIK $n \times \text{faktorial}(n-1)$

AKHIR-JIKA AKHIR-SUBROUTIN

b. Kode Program

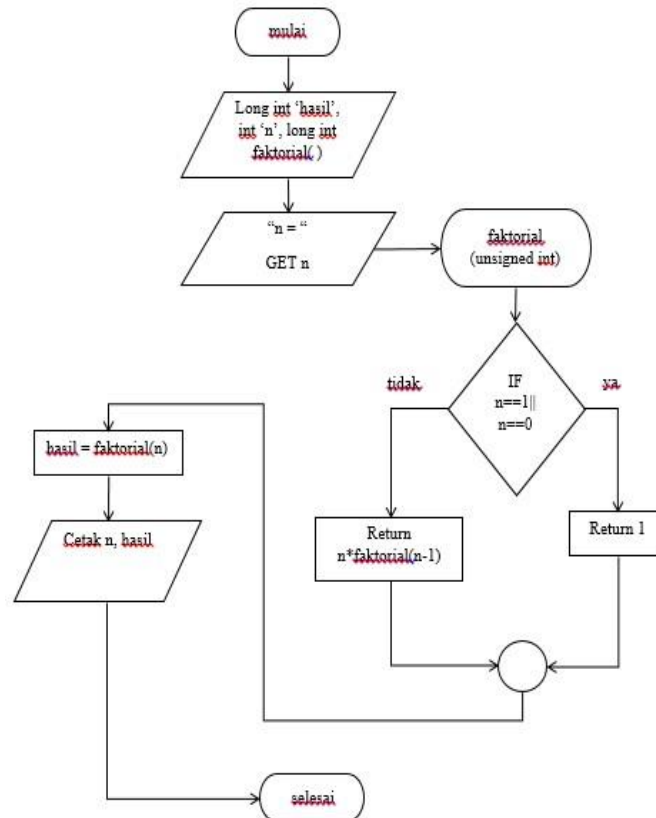
Berikut ini merupakan implementasi program dalam bahasa C.

```
#include <stdio.h>

long int faktorial (unsigned int n){
    if (n == 0 || n == 1){
        return 1;
    }else{
        return n * faktorial(n-1);
    }
}

int main(){
    int n; long int hasil;
    printf("Nama : Bagas Afza Joko Ariyanto\n");
    printf("NIM : 32601900008\n\n");
    printf("n = ");
    scanf("%d", &n);
    hasil = faktorial(n);
    printf("%d! = %ld", n, hasil);
    return 0;
}
```


c. Flowchart



Gambar 4.1 Flowchart Latihan 1 BAB 4

d. Output

```

D:\Deff C++\Laprak Algoritma\Untitled1.exe
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008

n = 5
5! = 120
-----
Process exited after 4.106 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 4.2 Output Latihan 1 BAB 4

e. Penjelasan

Pada gambar 4.2 merupakan hasil dari program diatas.
 Program diatas merupakan program yang menggunakan bahasa
 C. Program tersebut menghasilkan tampilan “n =” yang

dimaksudkan agar kita mengisi angka yang ingin difaktorialkan, kemudian tekan enter untuk mengetahui hasil dari faktorial.

Header yang digunakan pada program ini adalah *stdio.h*. *stdio.h* merupakan *header* yang berfungsi untuk operasi *input* dan *output*.

Stdio.h merupakan singkatan dari standar *input output header*.

2. Latihan 2

Tuliskan algoritma untuk menampilkan nilai satu deret *fibonaci* tertentu beserta programnya.

a. Algoritma

```

SUBROUTIN fib(n)

JIKA n = 0 MAKA

NILAI-BALIK 0

SEBALIKNYA JIKA n = 1

NILAI-BALIK 1

SEBALIKNYA

NILAI-BALIK fib(n-1) + fib(n-2)

AKHIR-JIKA

AKHIR-SUBROUTIN

```

b. Kode Program

Berikut ini merupakan implementasi program dalam bahasa C++.

```

#include <iostream>

using namespace std;

long int fib (unsigned int n){

    if (n == 0){

        return 0;

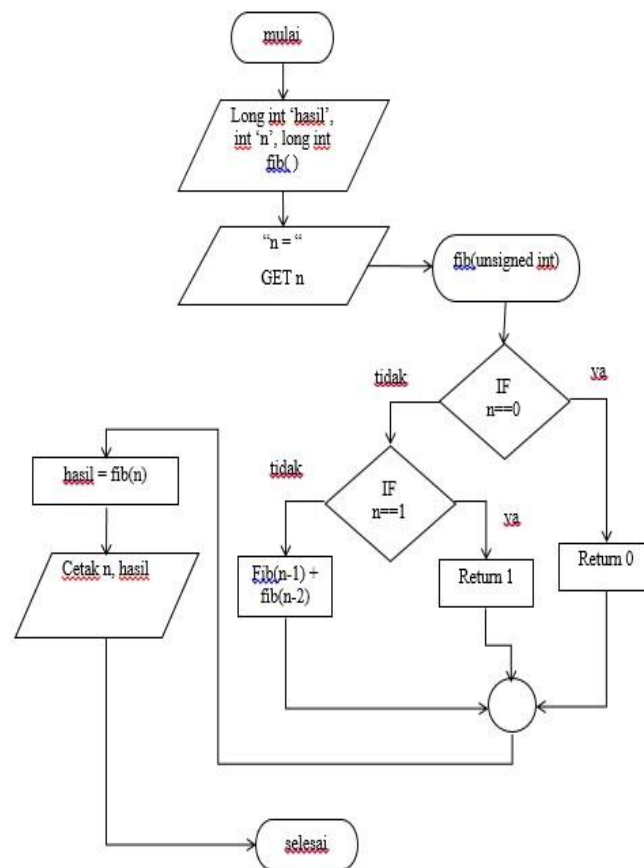
    }else if (n == 1){

        return 1;

```

```
}else{  
    return fib(n-1) + fib(n-2);  
}  
  
}  
  
int main(){  
    int n;  
  
    long int hasil;  
  
    cout << "Nama : Bagas Afza Joko Ariyanto\n";  
  
    cout <<"NIM : 32601900008\n\n";  
  
    cout<<"n = ";  
  
    cin>>n;  
  
    hasil = fib(n);  
  
    cout<<"fibonaci("<<n<<" ) = "<<hasil;  
  
    return 0;  
}
```

c. Flowchart



Gambar 4.3 Flowchart Latihan 2

c. Output

```

D:\Deff C++\Laprak Algoritma\Latihan 2 BAB 4.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

n = 7
fibonacci(7) = 13
-----
Process exited after 3.983 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 4.4 Output Latihan 2

d. Penjelasan

Pada gambar 4.4 merupakan hasil dari program diatas. Program diatas merupakan program yang menggunakan bahasa C++. Yang menghasilkan tampilan untuk mencari deret

fibonanci. Dengan cara memasukkan nilai n kemudian tekan enter untuk mengetahui hasilnya.

Pada program diatas merupakan program yang menggunakan *header iostream*. *Iostream* merupakan *header* yang melakukan perintah *cin* dan *cout* dan *main* merupakan fungsi yang pertama dijalankan.

4.4 Tugas

1. Berikut ini merupakan algoritma untuk menghitung nilai Y^n dengan berupa bilangan bulat lebih besar dari pada nol yang dihitung secara rekursif.

SUBROUTIN pangkat(y,n)

JIKA $n = 1$ MAKA

NILAI BALIK y

SEBALIKNYA

NILAI BALIK y x pangkat(y,n-1)

AKHIR JIKA AKHIR SUBROUTIN

Buatlah program menggunakan bahasa C/C++ untuk menghitung nilai Y pangkat n tersebut berdasarkan algoritma di atas.

Jawaban:

a. Kode Program

```
#include <iostream>
#include <conio.h>
using namespace std;
int pangkat (int y, int b){
    int hasil=1;

    for(int i=1;i<=b;i++){
```

```
        hasil=hasil*y;

    }

    return hasil;

}

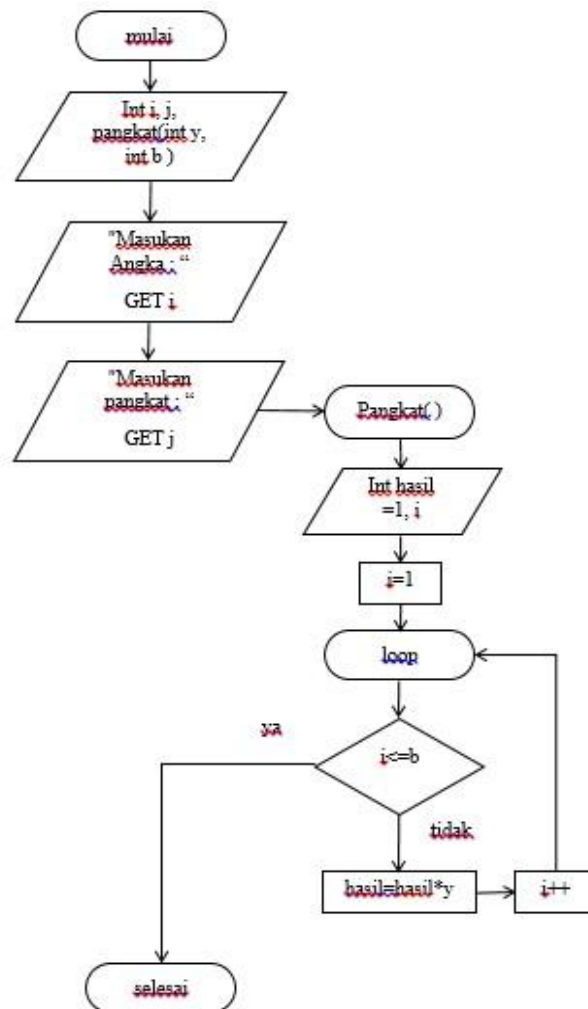
int main(){

    int i,j;

    cout<< "Nama : Bagas Afza Joko Ariyanto\n";
    cout<< "NIM : 32601900008\n\n";
    cout<<"**program Menghitung Pangkat**\n";
    cout<<"=====\n";
    cout<<endl<<"Masukan Angka :";
    cin>>i;
    cout<<endl<<"Masukkan Pangkat :";
    cin>>j;
    cout<<endl<<"Hasilnya :";
    cout<<pangkat(i,j);

}
```

b. Flowchart



Gambar 4.5 Flowchart Tugas 1

c. Output

```

D:\Deff C++\Laprak Algoritma\Tugas 1 BAB 4.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

**program Menghitung Pangkat**
=====

Masukan Angka :3

Masukkan Pangkat :4

Hasilnya :81
-----
Process exited after 9.038 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 4.6 Output Tugas 1

d. Penjelasan

Pada gambar 4.6 merupakan hasil dari program diatas. Program tersebut merupakan program yang menampilkan program menghitung pangkat. Lalu hasil dari program tersebut memberi perintah agar memasukkan angka dan pangkat. Kemudian program akan mencetak hasil pangkat dari angka dan pangkat yang dimasukkan.

Header yang digunakan adalah `<iostream>` dan `conio.h`. *iostream* merupakan *header* yang menjalankan perintah *cin* dan *cout*. Sedangkan *conio.h* merupakan program untuk membersihkan layar dan menahan tampilan.

2. Ubahlah program yang telah dibuat di nomor 1 kedalam bahasa Java.

Jawaban:

a. Kode Program

```
#include <stdio.h>

#include <conio.h>

using namespace std;

int pangkat(int y, int n){
    int pangkat=1;
    for (int i=1;i<=n;i++){
        pangkat=pangkat*y;
    }return pangkat;
}

int main(){
    int i,j;

    printf("Nama : Bagas Afza Joko Ariyanto");
    printf("NIM : 32601900008");
    printf("**program Menghitung Pangkat**\n");
```



```

printf("=====\n");

printf("\nMasukan Angka :");

scanf("%d",&i);

printf("\nMasukan pangkat :");

scanf("%d",&j);

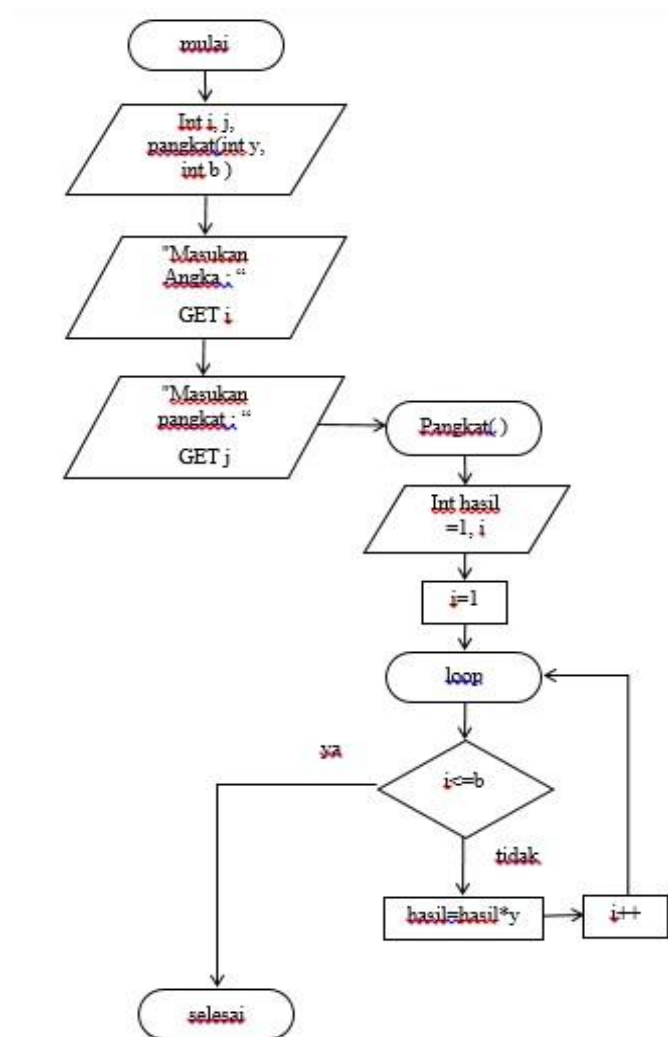
printf("\nhasil pangkat :");

printf("%d", pangkat(i,j));

}

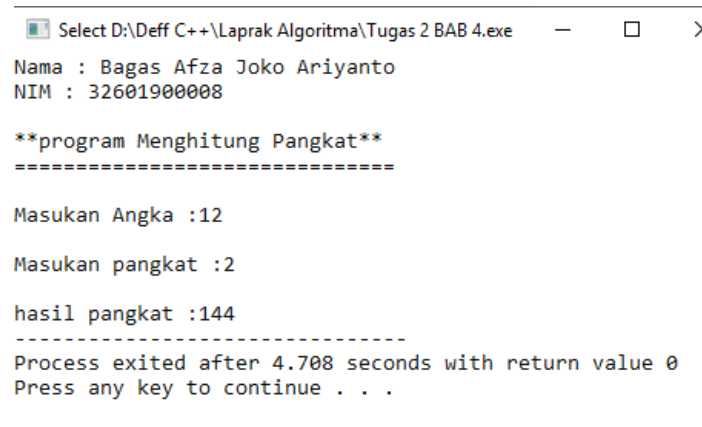
```

b. Flowchart



Gambar 4.7 Flowchart Tugas 2

c. Output



```

Select D:\Deff C++\Laprak Algoritma\Tugas 2 BAB 4.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

**program Menghitung Pangkat**
=====

Masukan Angka :12

Masukan pangkat :2

hasil pangkat :144
-----
Process exited after 4.708 seconds with return value 0
Press any key to continue . . .

```

Gambar 4.8 Output Tugas 2

d. Penjelasan

Pada gambar 4.8 merupakan hasil dari program diatas. Program diatas merupakan program yang menghasilkan tampilan program menghitung pangkat. Yang artinya program diatas dibuat untuk mencari sebuah pangkat dari suatu bilangan. Agar dapat mendapatkan hasil dari pangkat suatu bilangan, anda harus memasukkan angka dan pangkat kemudian tekan enter untuk mengetahui hasilnya.

3. Buatlah algoritma dari subrutin untuk membalik suatu bilangan dengan cara rekursi. Sebagai contoh, bilangan 1261 ditampilkan menjadi 1621.

Jawaban:

a. Kode Program

```

#include <iostream>

#include <conio.h>

using namespace std;

class membalik{

public:

    int proses();

    void keluaran ();

```

```

        private:
            int x,balik;
    };

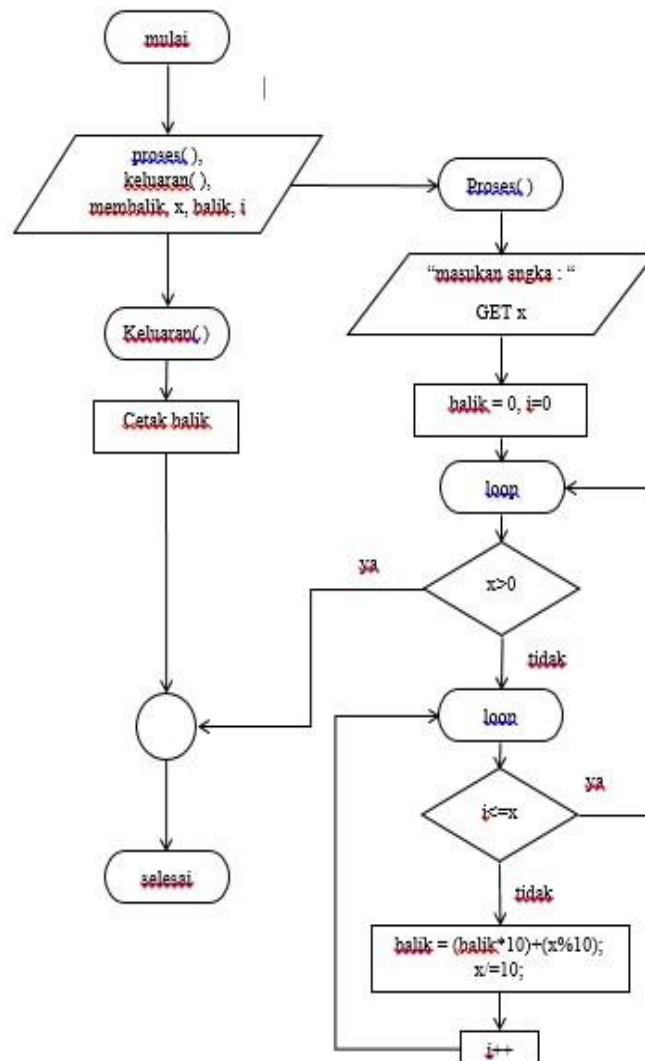
    int membalik::proses(){
        cout << "Nama : Bagas Afza Joko Ariyanto\n";
        cout << "NIM : 32601900008\n\n";
        cout<<"~<membalik angka yang ingin
dibalik>~"<<endl<<endl;
        cout<<"masukan angka : ";
        cin>>x;
        balik=0;
        while (x>0)
            for (int i=0; i<=x; i++){
                balik = (balik*10)+(x%10);
                x/=10;
            }
        return balik;
    }

    void membalik::keluaran(){
        cout<<"angka yang telah dibalik menjadi
:"<<balik<<endl;
    }

    int main(int argc, char**argv){
        membalik bilangan;
        bilangan.proses();
        bilangan.keluaran();
        return 0;
    }

```

b. Flowchart



Gambar 4.9 Flowchart Tugas 3

c. Output

```

D:\Deff C++\Laprak Algoritma\Tugas 3 BAB 4.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

~<membalik angka yang ingin dibalik>~

masukan angka : 1698
angka yang telah dibalik menjadi :8961

-----
Process exited after 12.17 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 4.10 Output Tugas 3

d. Penjelasan

Pada gambar 4.10 merupakan hasil dari program diatas. Gambar diatas adalah gambar yang menampilkan masukkan angka. Kemudian memerintahkan kita untuk memasukkan data secara acak dan jumlah sesuai keinginan. Dalam gambar tersebut menunjukkan adanya pertukaran data, sehingga menghasilkan angka yang urut.

Program diatas merupakan program yang menggunakan *header iostream* dan *conio.h*. *iostream* merupakan *header* yang berfungsi menjalankan perintah *cin* dan *cout*. Sedangkan *conio.h* merupakan *header* untuk menahan tampilan.

4. Buatlah program menggunakan bahasa C/C++ berdasarkan algoritma pada soal nomor 3.

a. Kode Program

```
#include <iostream>

#include <conio.h>

using namespace std;

class membalik{
public:
    int proses();
    void keluaran ();
private:
    int x,balik;
};

int membalik::proses(){
    cout << "Nama : Bagas Afza Joko Ariyanto\n";
    cout << "NIM : 32601900008\n\n";
    cout<<"~<membalik angka yang ingin
dibalik>~"<<endl<<endl;

    cout<<"masukan angka : ";
```

```
        cin>>x; balik=0;

        while (x>0)

            for (int i=0; i<=x; i++){

                balik = (balik*10)+(x%10);

                x/=10;

            }

        return balik;

    }

    void membalik::keluaran(){

        cout<<"angka yang telah dibalik menjadi

        :"<<balik<<endl;

    }

    int main(int argc, char**argv){

        membalik bilangan;

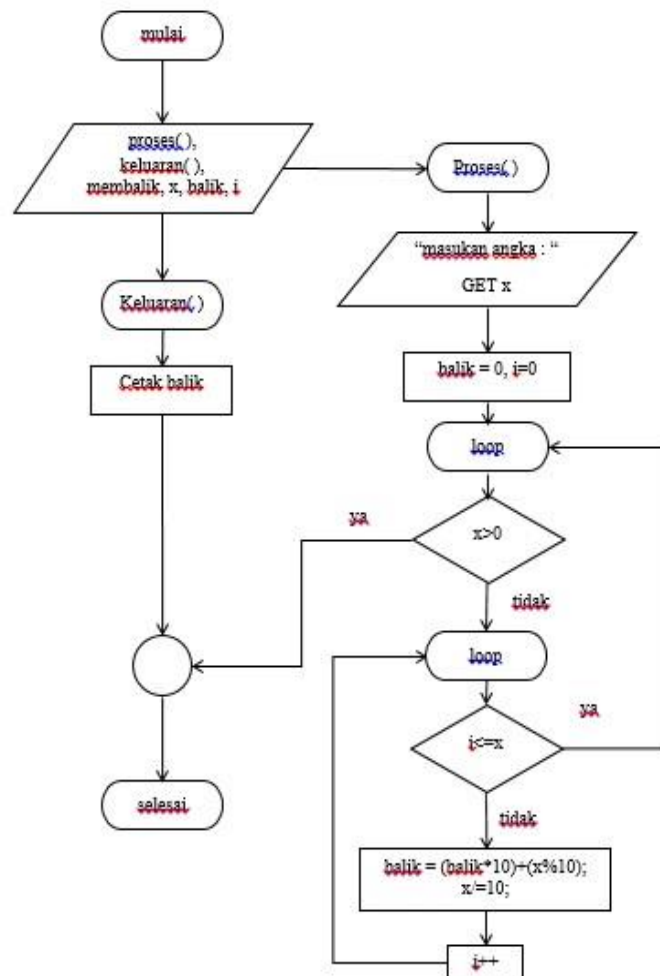
        bilangan.proses();

        bilangan.keluaran();

        return 0;

    }
```

b. Flowchart



Gambar 4.11 Flowchart Tugas 4

c. Output

```

Select D:\Deff C++\Laprak Algoritma\Tugas 4 BAB 4.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

~<membalik angka yang ingin dibalik>~

masukan angka : 1236598
angka yang telah dibalik menjadi :8956321

-----
Process exited after 20.49 seconds with return value 0
Press any key to continue . . .

```

Gambar 4.12 Output Dari Tugas 4

d. Penjelasan

Pada gambar 4.12 merupakan hasil dari program diatas.

Program diatas merupakan program yang menampilkan membalik angka yang ingin dibalik. Kemudian terdapat tampilan masukkan angka, dalam tampilan ini anda diperintahkan untuk memasukkan angka yang ingin dibalik kemudian tekan enter dan akan menghasilkan angka yang sudah dibalik.

5. Ubahlah program yang telah dibuat pada soal nomor 4 ke dalam bahasa C.

Jawaban :

a. Kode program

```
#include <stdio.h>
#include <conio.h>
using namespace std;
class membalik{
public:
    int proses();
    void keluaran ();
private:
    int x,balik;
};
int membalik::proses(){
printf("Nama : Bagas Afza Joko Ariyanto\n");
printf("NIM : 32601900008\n\n");
printf("\t ***PROGAM MEMBALIK ANGKA***\n\n");
printf("Masukkan angka yang ingin dibalik : ");
scanf("%d",&x);
    balik=0; while (x>0)
        for (int i=0; i<=x; i++){
            balik = (balik*10)+(x%10);
            x/=10;
        }
    return balik;
}
void membalik::keluaran(){
```

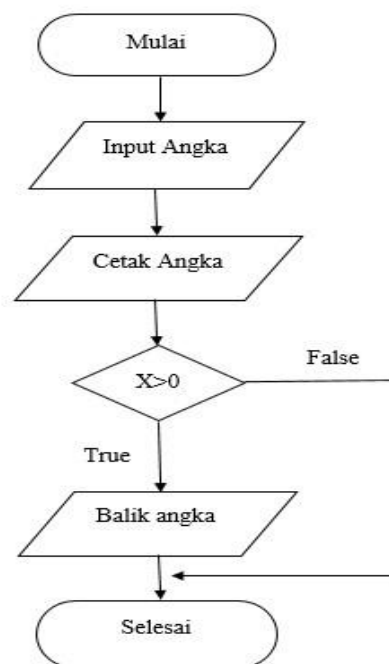


```

printf("\nAngka yang telah dibalik :");
printf("%d", balik);
}
int main(int argc, char**argv){
    membalik bilangan;  bilangan.proses();
    bilangan.keluaran();
    return 0;
}

```

b. *Flowchart*



Gambar 4.13 *Flowchart* Tugas 5

c. *Output*

```

D:\Deff C++\Laprak Algoritma\Tugas 5 BAB 4.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

***PROGAM MEMBALIK ANGKA***

Masukkan angka yang ingin dibalik : 192837465

Angka yang telah dibalik :564738291
-----
Process exited after 9.083 seconds with return value 0
Press any key to continue . . .

```

Gambar 4.14 *Output* Dari Tugas 5

b. Penjelasan

Pada gambar 4.14 merupakan hasil dari program diatas. Program diatas merupakan program yang menggunakan bahasa java. Program diatas menampilkan program membalik angka, masukkan angka yang ingin dibalik (untuk memasukkan angka yang ingin dibalik), angka yang telah dibalik (untuk menampilkan angka yang sudah dibalik, yang dihasilkan oleh program tersebut.

4.5 Kesimpulan

Rekursi adalah suatu kemampuan subrutin untuk memanggil dirinya sendiri. Adapun suatu subrutin yang memanggil dirinya seperti itu dinamakan subrutin rekursi. Pada beberapa persoalan, kemampuan seperti itu sangat berguna karena mempermudah solusi. Namun demikian rekursi juga memiliki kelemahan, yakni memungkinkan terjadinya *overflow* pada *stack* (*stack* tidak lagi mampu menangani permintaan pemanggilan subrutin karena kehabisan memori). Itu sebabnya harus ada jaminan bahwa proses rekursi akan berhenti pada suatu waktu tertentu, yang menyebabkan pemanggilan fungsi berakhir. Berikut adalah kelebihan dan kekurangan rekursi :

a. Kelebihan Rekursif

1. Sangat mudah untuk melakukan perulangan dengan batasan yang luas dalam artian melakukan perulangan dalam skala yang besar.
2. Dapat melakukan perulangan dengan batasan fungsi.

b. Kekurangan Rekursif

1. Tidak bisa melakukan nested loop atau looping bersarang.
2. Biasanya membuat fungsi sulit untuk dipahami, hanya cocok untuk persoalan tertentu saja.
3. Memerlukan *stack* yang lebih besar, sebab setiap kali fungsi dipanggil, variabel lokal dan parameter formal akan ditempatkan ke

stack dan ada kalanya akan menyebabkan *stack* tak cukup lagi (*Stack Overrun*).

4. Proses agak berbelit-belit karena terdapat pemanggilan fungsi yang berulang-ulang dan pemanggilan data yang ditumpuk.

BAB V

LINKED LIST, STACK, DAN QUEUE

5.1 Tujuan

Adapun tujuan dari praktikum ini adalah sebagai berikut :

1. Mengetahui penggunaan metode *LINKED LIST*
2. Mengetahui penggunaan metode *STACK*
3. Mengetahui penggunaan metode *QUEUE*

5.2 Dasar Teori

5.2.1 Metode List

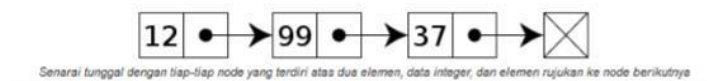
List atau senarai adalah sebuah pemikiran/ konsep struktur data yang sangat dasar pada pemrograman agar lebih fleksibel, dimana setiap elemen akan ditambahkan saat dibutuhkan. struktur data ini digunakan untuk menyimpan sejumlah objek data biasanya secara terurut sehingga memungkinkan penambahan, pengurangan, dan pencarian atas elemen data yang tersimpan dalam senarai dilakukan secara lebih efektif.



Gambar 5.1 Senarai

Jenis– jenis Senarai berantai ada 3:

1. Senarai Tunggal Bila struktur data sebuah *node* hanya memiliki satu tautan atas *node* berikutnya dalam sebuah senarai, maka senarai tersebut dinamakan sebagai senarai tunggal.



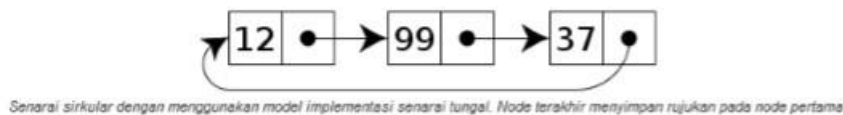
Gambar 5.2 Senarai Tunggal

2. Senarai Ganda Berbeda halnya dengan senarai tunggal, pada senarai ganda, struktur data atas tiap-tiap node memiliki rujukan pada *node* sebelum dan berikutnya. Sebagian algoritma membutuhkan taut ganda, contohnya *sorting* dan *reverse traversing*.



Gambar 5.3 Senarai Ganda

3. Senarai Sirkular Pada dua jenis senarai sebelumnya, *node* terakhir dalam senarai tersebut merujuk pada *null* yang artinya akhir dari sebuah senarai, begitu pula *null* sebagai rujukan *node* sebelumnya pada *node* pertama bila senarai yang dimaksudkan adalah senarai ganda. Pada senarai sirkular, informasi rujukan pada *node* terakhir akan merujuk pada *node* pertama, dan rujukan pada *node* pertama akan merujuk pada *node* terakhir bila yang digunakan sebagai dasar implementasi adalah senarai ganda.



Gambar 5.4 Senarai Sirkular

5.2.2 Metode Stack

Sebuah koleksi objek yang menggunakan prinsip *LIFO* (*Last In First Out*), yaitu data yang terakhir kali dimasukkan akan pertama kali keluar dari tumpukan tersebut. Tumpukan dapat diimplementasikan sebagai representasi berkait atau kontigu (dengan tabel *fix*).

Ciri tumpukan :

- 1) *TOP* merupakan sebutan untuk elemen paling atas dari suatu stack
- 2) Elemen *TOP* merupakan elemen yang paling akhir ditambahkan
- 3) Elemen *TOP* diketahui
- 4) Penambahan dan penghapusan elemen selalu dilakukan di *TOP*

5.2.3 LIFO

Pemanfaatan tumpukan :

- 1) Perhitungan ekspresi aritmatika (*posfix*)
- 2) Algoritma *backtracking* (runut balik)

3) Algoritma rekursif

5.2.4 Metode Queue

Queue/antrian adalah *ordered list* dengan penyisipan di satu ujung, sedang penghapusan di ujung lain. Ujung penyisipan biasa disebut *rear/tail*, sedang ujung penghapusan disebut *front/head*. Fenomena yang muncul adalah elemen yang lebih dulu disisipkan akan juga lebih dulu diambil. *Queue* berdisiplin *FIFO (First In, First Out)*. *Queue* merupakan kasus khusus *ordered list*. Dengan karakteristik terbatas itu maka kita dapat melakukan optimasi representasi *ADT Queue* untuk memperoleh kerja paling optimal. (Putra, Andreswari, & Susilo, 2015)

5.2.5 Pengertian Linked List

Linked List atau dikenal juga dengan sebutan senarai berantai adalah struktur data yang terdiri dari urutan *record* data dimana setiap *record* memiliki *field* yang menyimpan alamat/referensi dari *record* selanjutnya (dalam urutan). Elemen data yang dihubungkan dengan link pada *Linked List* disebut Node. Biasanya didalam suatu *linked list*, terdapat istilah *head* dan *tail*. (Ahmad Fali Oklilas1, 2014)

- 1) *Head* adalah elemen yang berada pada posisi pertama dalam suatu *linked list*
- 2) *Tail* adalah elemen yang berada pada posisi terakhir dalam suatu *linked list*.

Ada beberapa macam *Linked List*, yaitu :

- a) Single Linked List
- b) Double Linked List
- c) Circular Linked List
- d) Multiple Linked List

5.2.6 Pengertian Stack

Stack adalah suatu tumpukan dari benda. Konsep utamanya adalah *LIFO (Last In First Out)*, benda yang terakhir masuk dalam *stack* akan menjadi benda pertama yang dikeluarkan dari *stack*. Tumpukan disebut juga “*Push Down Stack*” yaitu penambahan elemen baru (*PUSH*) dan penghapusan elemen dari tumpukan (*POP*). Contoh pada PDA (*Push Down Automaton*). Sistem pada pengaksesan pada tumpukan menggunakan *system LIFO (Last In First Out)*, artinya elemen yang terakhir masuk itu yang akan pertama dikeluarkan dari tumpukan (*Stack*). Ilustrasi tumpukan (*Stack*) dapat digambarkan seperti tumpukan CD atau tumpukan sate. *Stack* merupakan suatu susunan koleksi data dimana dapat ditambahkan dan dihapus selalu dilakukan pada bagian akhir data, yang disebut dengan *Top Of Stack*. (Andriani, Munawaroh, & Nursuprianah, 2015)

5.2.7 Pengertian Queue

Queue merupakan suatu struktur data linear. Konsepnya hampir sama dengan *Stack*, perbedaannya adalah operasi penambahan dan penghapusan pada ujung yang berbeda. Penghapusan dilakukan pada bagian depan (*front*) dan penambahan berlaku pada bagian belakang (*Rear*). Elemen-elemen di dalam antrian dapat bertipe *integer*, *real*, *record* dalam bentuk sederhana atau terstruktur.

Tumpukan disebut juga “*Waiting Line*” yaitu penambahan elemen baru dilakukan pada bagian belakang dan penghapusan elemen dilakukan pada bagian depan. Sistem pada pengaksesan pada *Queue* menggunakan sistem *FIFO (First In First Out)*, artinya elemen yang pertama masuk itu yang akan pertama dikeluarkan dari *Queue*. *Queue* jika diartikan secara harfiah, *queue* berarti antrian. *Queue* merupakan salah satu contoh aplikasi dari pembuatan *double linked list* yang cukup sering kita temui dalam kehidupan sehari-hari, misalnya saat anda mengantri diloket untuk membeli tiket.

Istilah yang cukup sering dipakai apabila seseorang masuk dalam sebuah antrian adalah *enqueue*. Sedang istilah yang sering dipakai bila seseorang keluar dari antrian adalah *dequeue*. (Kristanto, 2007)

5.3 Latihan

1. Latihan 1

```
#include <iostream>
#include <cstring>
#include <stdio.h>
using namespace std;

typedef struct{
    char nim[10];
    char nama[50];
    float nilai;
}nilaiMatKul;

typedef struct{
    int first;
    int last;
    nilaiMatKul data[10];
}queue;

void createEmpty(queue*Q) {
    (*Q).first = -1; (*Q).last = -1;
}

int isEmpty(queue Q){
    int hasil = 0;

    if(Q.first == -1){
        hasil=1;
    }
    return hasil;
}

int isFull(queue Q){
    int hasil = 0;

    if(Q.last==9){
        hasil=1;
```



```

    }
    return hasil;
}

void add(char nim[], char nama[], float nilai, queue *Q){

    if(isEmpty(*Q) == 1 ){

        (*Q).first=0;
        (*Q).last=0;
        strcpy((*Q).data[0].nim,nim);
        strcpy((*Q).data[0].nama,nama);
        (*Q).data[0].nilai=nilai;
    }else{
        if(isFull(*Q) !=1){
            (*Q).last=(*Q).last+1;
            strcpy((*Q).data[(*Q).last].nim,nim);
            strcpy((*Q).data[(*Q).last].nama,nama);
            (*Q).data[(*Q).last].nilai = nilai;
        } else{
            printf("queue penuh\n");
        }
    }
}

void del(queue *Q){

    if((*Q).last ==0){

        (*Q).first ==-1;
        (*Q).last ==-1;
    }else{
        int i;
        for(i=(*Q).first+1;i<=(*Q).last;i++){

            strcpy((*Q).data[i-1].nim, (*Q).data[i].nim);

```

```

strcpy((*Q).data[i-1].nama, (*Q).data[i].nama);
(*Q).data[i-1].nilai=(*Q).data[i].nilai;
}
(*Q).last=(*Q).last-1;
}
}

void printfQueue(queue Q){
if (Q.first != -1){
printf("-----isi queue-----\n");

int i;
for(i=Q.last;i>=Q.first;i--){
printf("elemen ke : %d\n",i);
printf("nim : %s\n",Q.data[i].nim);
printf("nama: %s\n",Q.data[i].nama);
printf("nilai: %f\n",Q.data[i].nilai);
}
printf("-----\n");
} else{
printf("queue kosong\n");
}
}

int main(){
queue Q;
createEmpty(&Q);
printfQueue(Q);

printf("-----\n");
add("13507701","Rangga","64.67", &Q);
add("13507702","Anggy","75.11", &Q);
add("13507703","Intan","84.63", &Q);
printfQueue(Q);
printf("-----\n");
del(&Q);
del(&Q);
printfQueue(Q);
printf("-----\n");

```

```

return 0;
}

```

2. Latihan 2

```

#include <iostream>

#include <cstring>
#include <iomanip>
using namespace std;

    struct DAT {
int id;  char
fname[20]; char
mname[20]; char
lname[20]; char
address[80];
double salary;
char
tele_no[15];

};

struct NODE {
DAT data;

NODE *N;

NODE*P;

NODE(const int i , const char *f, const char *m, const char
*l, const char *ad, const double s, const char
*tel){      data.id = i;      strcpy(data.fname,f);
      strcpy(data.mname,m);      strcpy(data.lname,l);
      strcpy(data.address,ad);      data.salary = s;
      strcpy(data.tele_no,tel);

      N = NULL;

      P = NULL;

}

};

```

```

class StackLinkedList {
private:
    NODE *front;
public:
    StackLinkedList() {front = NULL;}
    ~StackLinkedList() {destroyList();}
    void push(NODE *);      NODE*
    pop();      void destroyList();

};

void StackLinkedList::push(NODE *n) {
    if(front == NULL) {

        front = n;

    }

    else{

        front->P = n;
        n->N = front;
        front = n;

    }

}

NODE* StackLinkedList::pop()
{ NODE *temp; if( front ==
NULL )//no nodes      return
NULL;

else if(front->N == NULL)//there is only one node
{

    NODE * temp2 =
front;      temp = temp2;
    front = NULL;
    delete temp2;
    return temp;
}
}

```

```

    }

    else//there are more than one node
    {

        NODE * temp2 =
front;    temp = temp2;
        front = front->N;
        front->P = NULL;
        delete temp2;
        return temp;

    }

}

void
StackLinkedList::destroyList(){
while(front != NULL){          NODE
*temp = front;    front = front->N;
    delete temp;

}

}

void disp(NODE *N){
if( N == NULL ){

    cout << "\nStack is Empty!!!" << endl;

}
else{

    cout << "\nId No.      : " << N->data.id << " ";
    cout << "\nFull Name  : " << N->data.fname << "
";    cout << N->data.mname << " ";        cout << N-
>data.lname << endl;

    cout << "Address      : " << N->data.address << endl;
    cout << "Salary       : " << setprecision(15) <<
N->data.salary << endl;

    cout << "Tele_no      : " << N->data.tele_no<< endl << endl;

}
}

```

```

    }    int
main() {
    StackLinkedList *Stack = new StackLinkedList();

    NODE No1(101,"Anggy","Eka","P","Semarang
120",7851243.9475,"07502334121");

    NODE No2(102,"Wa Ode","Kanartia","Ningsi","Timor leste
121",5681125.9457,"07507534127");

    NODE No3(103,"Fatimah","Nurul","Intan","Irian Jaya
123",2344003.48345,"075078654129");

    Stack->push(&No1);

    Stack->push(&No2);
    Stack->push(&No3);
    disp(Stack->pop());
    disp(Stack->pop());
    disp(Stack->pop());
    disp(Stack->pop());
    delete Stack;          return
0;

}

```

5.4 Tugas

1. Benarkan *listing* program sehingga dapat di-*compile* !

```

#include
<iostream> using
namespace std
class node {    int
data;    Node* next;
public:    Node()
{};

void SetData(int aData) { data = aData; };
void SetNext(Node* aNext) { next = aNext; };
int Data() { return data; };

Node* Next() { return next; }

```

```

}; class
List {
Node *head;
public:

List() { head = NULL;
}; void Print(); void
Append(int data); void
Delete(int data);

}; void List::Print()
{ Node *tmp = head;
if ( tmp == NULL ) {
cout << "NULL" <<
endl; return; }

if ( tmp->Next() == NULL )
{ cout << tmp->Data();
cout << " --> ";
        cout << "EMPTY" << endl;
} else { do {
cout << tmp->
Data(); cout << "
--> ";

tmp = tmp->Next(); }
while ( tmp != NULL );
cout << "NULL" <<
endl;

} } void List::Append(int
data) { if(data != 300) {
Node* newNode = new Node();
newNode->SetData(data);
newNode->SetNext(NULL); Node
*tmp = head; if ( tmp != NULL
) { while ( tmp->Next() !=
NULL ) { tmp = tmp->Next();

```

```

    } tmp->SetNext(newNode); }
else { head = newNode;

} } } void List::Delete(int
data) { if(data != 300) { Node
*tmp = head; if ( tmp == NULL )
return; if ( tmp->Next() ==
NULL ) { delete tmp; head =
NULL; } else { Node *prev;
do { if ( tmp->Data() == data )
break; prev = tmp; tmp = tmp-
>Next();

} while ( tmp != NULL );
// Adjust the pointers
prev->SetNext(tmp-
>Next()); // Delete the
current node delete tmp;
}

}

int main() {
List list;
list.Append(100)
; list.Print();
list.Append(200)
; list.Print();
list.Append(300)
; list.Print();
list.Append(400)
; list.Print();
list.Append(500)
; list.Print();
list.Delete(400)
; list.Print();
list.Delete(300)
; list.Print();

```



```
list.Delete(200)
; list.Print();
list.Delete(500)
; list.Print();
list.Delete(100)
; list.Print();
```

Jawaban:

a. Kode Program

```
#include <iostream>
using namespace std;
class node {
    int data;
    node* next;
    public:
    node() {
        };
    void SetData(int aData) {
        data = aData;
    };
    void SetNext(node* aNext) {
        next = aNext;
    };
    int Data() {
        return data;
    };
    node* Next() {
        return next;
    }
};
class List {
    node *head; public:
    List() {
        head = NULL;
    };
    void Print();
    void Append(int data);
    void Delete(int data);
```

```

};

void List::Print() {
    node *tmp = head;
    if ( tmp == NULL ) {
        cout << "NULL" << endl;
        return;
    }
    if ( tmp->Next() == NULL ) {
        cout << tmp->Data();
        cout << " --> ";
        cout << "EMPTY" << endl;
    } else {
        do {
            cout << tmp->Data();
            cout << " --> ";
            tmp = tmp->Next();
        }
        while ( tmp != NULL );
        cout << "NULL" << endl;
    }
}

void List::Append(int data)
{ if(data != 300) {
    node* newnode = new node();
    newnode->SetData(data);
    newnode->SetNext(NULL);
    node *tmp = head;
    if ( tmp != NULL ) {
        while ( tmp->Next() != NULL ) {
            tmp = tmp->Next();
        } tmp->SetNext(newnode);
    } else {
        head = newnode;
    }
}
}

void List::Delete(int data) {
    if(data != 300) {

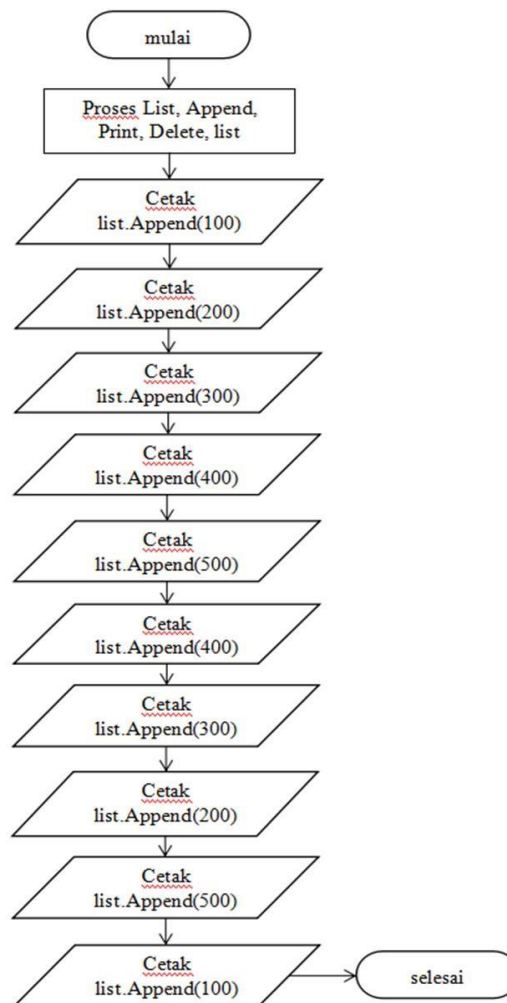
```

```

node *tmp = head; if ( tmp == NULL ) return;
if ( tmp->Next() == NULL ) {
    delete tmp; head = NULL;
} else {
    node *prev;
    do {
        if ( tmp->Data() == data ) break;
        prev = tmp;
        tmp = tmp->Next(); }
    while ( tmp != NULL );
// Adjust the pointers prev->SetNext(tmp->Next()); //
Delete the current node delete tmp; } }
}
}
}

int main(){
    cout << "Nama : Bagas Afza Joko Ariyanto\n";
    cout << "NIM : 32601900008\n\n";
    List list; list.Append(100);
    list.Print();
    list.Append(200);
    list.Print();
    list.Append(300);
    list.Print();
    list.Append(400);
    list.Print();
    list.Append(500);
    list.Print();
    list.Delete(400);
    list.Print();
    list.Delete(300);
    list.Print();
    list.Delete(200);
    list.Print();
    list.Delete(500);
    list.Print();
    list.Delete(100);
    list.Print();}

```

b. *Flowchart*Gambar 5.5 *Flowchart* Tugas 1c. *Output*

```

D:\Deff C++\Laprak Algoritma\Untitled3.exe
Nama : Bagus Afza Joko Ariyanto
NIM : 32601900008

100 --> EMPTY
100 --> 200 --> NULL
100 --> 200 --> NULL
100 --> 200 --> 400 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 400 --> 500 --> NULL

-----
Process exited after 0.2256 seconds with return value 0
Press any key to continue . . .
  
```

Gambar 5.6 *Output* Tugas 1

d. Penjelasan

Pada *script* pertama kita disuruh mencari kesalahannya untuk menghasilkan *output* seperti gambar diatas,sebenarnya penulisan *scrip*.

Cuma mengandung sedikit semisal kurangnya “;”,dan juga lagi penusilan pada pemanggilan *node*, yang seharusnya ditulis “*node*” Akan tetapi pada *script* pertama ditulis “*Node*”,karena penggunaan huruf besar dan kecil sangat berpengaruh terhadap pemrograman.

2. Ubah program sehingga menjadi seperti hasil *output* dibawah ini
(Gunakan *Listing* program 1).

Jawaban :

a. Kode Program

```
#include <iostream>
using namespace std;
class node{
    int data;
    node* next;
    public:
    node() {
    };
void SetData(int aData) {
    data = aData;
};
void SetNext(node* aNext) {
    next = aNext;
};
int Data() {
    return data;
};
node* Next() {
    return next;
}
};
class List {
    node *head;
```

```

    public:
List() {
    head = NULL;
};
void Print();
    void Append(int data);
    void Delete(int data);
};
void List::Print() {
    node *tmp = head;
    if ( tmp == NULL ) {
        cout << "EMPTY" << endl; return;
    }
    if ( tmp->Next() == NULL ) {
        cout << tmp->Data();
        cout << " --> ";
        cout << "NULL" << endl;
    } else {
        do {
            cout << tmp->Data();
            cout << " --> ";
            tmp = tmp->Next();
        }
        while ( tmp != NULL );
        cout << "NULL" << endl;
    }
}

void List::Append(int data) {
    if(data != 300) {
        node* newnode = new node();
        newnode->SetData(data);
        newnode->SetNext(NULL);
        node *tmp = head;
        if ( tmp != NULL ) {
            while ( tmp->Next() != NULL ) {
                tmp = tmp->Next();
            } tmp->SetNext(newnode);
        } else {

```

```

        head = newnode;
    }
}

void List::Delete(int data) {
    if(data != 300) {
        node *tmp = head;
        if ( tmp == NULL ) return;
        if ( tmp->Next() == NULL ) {
            delete tmp; head = NULL;
        } else {
            node *prev;
            do {
                if ( tmp->Data() == data ) break;
                prev = tmp;
                tmp = tmp->Next();
            }
            while ( tmp != NULL );
            // Adjust the pointers
            prev->SetNext(tmp->Next());
            // Delete the current node
            delete tmp;
        }
    }
}

int main() {
    cout << "Nama : Bagas Afza Joko
Ariyanto\n";
    cout << "NIM : 32601900008\n\n";
    List list;
    list.Append(100);
    list.Print();
    list.Append(200);
    list.Print();
    list.Append(300);
    list.Print();
    list.Append(400);
    list.Print();
}

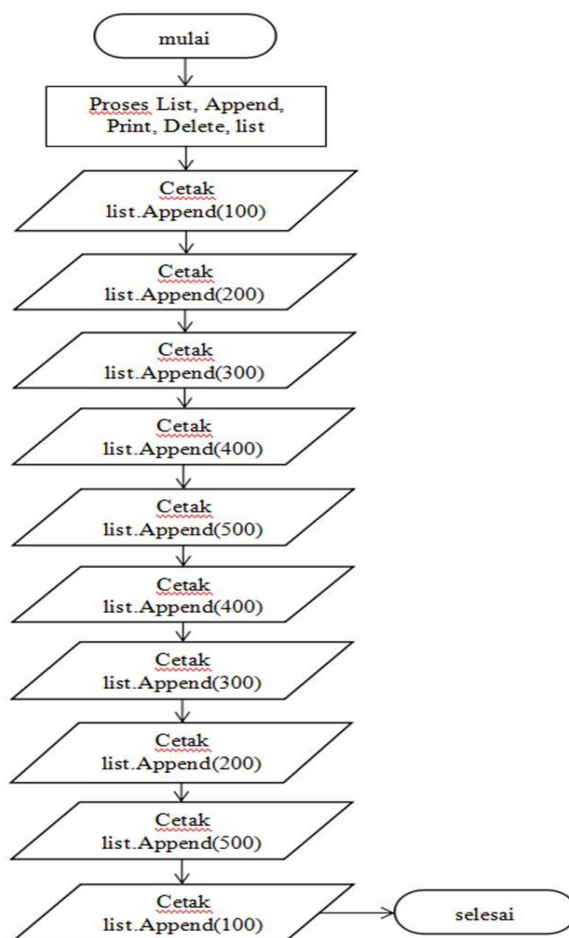
```

```

list.Append(500);
list.Print();
list.Delete(400);
list.Print();
list.Delete(300);
list.Print();
list.Delete(200);
list.Print();
list.Delete(500);
list.Print();
list.Delete(100);
list.Print();
}

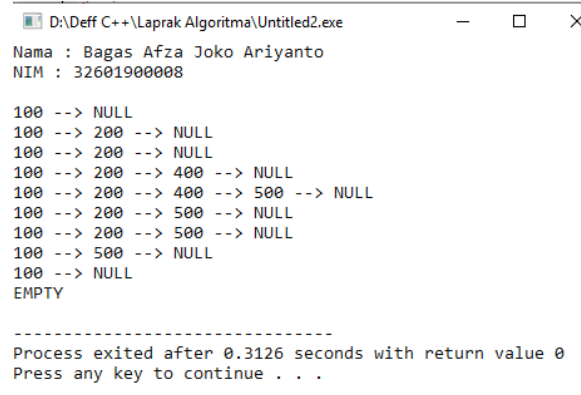
```

b. *Flowchart*



Gambar 5.7 *Flowchart* Tugas 2

c. Output



```

D:\Deff C++\Laprak Algoritma\Untitled2.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

100 --> NULL
100 --> 200 --> NULL
100 --> 200 --> NULL
100 --> 200 --> 400 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 500 --> NULL
100 --> 200 --> 500 --> NULL
100 --> 500 --> NULL
100 --> NULL
EMPTY

-----
Process exited after 0.3126 seconds with return value 0
Press any key to continue . . .

```

Gambar 5.8 Output Tugas 2

d. Penjelasan

Pada gambar 5.8 merupakan hasil dari program diatas. Program diatas merupakan program yang menggunakan bahasa C++. Yang menghasilkan tampilan untuk mencari *Linked List*. Dengan cara memasukkan program kemudian tekan *enter* untuk mengetahui hasilnya.

Pada program diatas merupakan program yang menggunakan *header iostream*. *Iostream* merupakan *header* yang melakukan perintah *int main()* dan *main* merupakan fungsi yang pertama dijalankan. Untuk menampilkan *list* tersebut

3. Ubah program menjadi bahasa C

Jawaban :

a. Kode Program

```

#include <stdio.h>

using namespace std;

class node{

int data;

node* next;

public:

node() {

};

```

```

void SetData(int aData) {
    data = aData;
};

void SetNext(node* aNext) {
    next = aNext;
};

int Data() {
    return data;
};

node* Next() {
    return next;
}

};

class List {
    node *head; public:
    List() {
        head = NULL;
    };

    void Print();
    void Append(int data);
    void Delete(int data);
}; void List::Print() {
    node *tmp = head;
    if ( tmp == NULL ) {
        printf("EMPTY\n");
        return;
    }

    if ( tmp->Next() == NULL ) {
        printf("%d", tmp->Data());
        printf(" --> ");
    }
}

```

```

printf("NULL\n");

} else {

do {

printf("%d", tmp->Data());

printf(" --> ");

tmp = tmp->Next();

}

while ( tmp != NULL );

printf("NULL\n");

}

}

void List::Append(int data) {

if(data != 300) {

node* newnode = new node();

newnode->SetData(data);

newnode->SetNext(NULL);

node *tmp = head;

if ( tmp != NULL ) {

while ( tmp->Next() != NULL ) {

tmp = tmp->Next();

}

tmp->SetNext(newnode);

} else {

head = newnode;

}

}

}

void List::Delete(int data) {

if(data != 300) {

node *tmp = head;

```

```

if ( tmp == NULL ) return;
if ( tmp->Next() == NULL ) {
delete tmp; head = NULL;
} else {
node *prev;
do {
if ( tmp->Data() == data ) break;
prev = tmp; tmp = tmp->Next();
}
while ( tmp != NULL );
// Adjust the pointers
prev->SetNext(tmp->Next());
// Delete the current node
delete tmp;
}
}
}

int main() {
printf("Nama : Bagas Afza Joko Ariyanto\n");
printf("NIM : 32601900008\n\n");
List list;
list.Append(100);
list.Print();
list.Append(200);
list.Print();
list.Append(300);
list.Print();
list.Append(400);
list.Print();
list.Append(500);

```

```

list.Print();

list.Delete(400);

list.Print();

list.Delete(300);

list.Print();

list.Delete(200);

list.Print();

list.Delete(500);

list.Print();

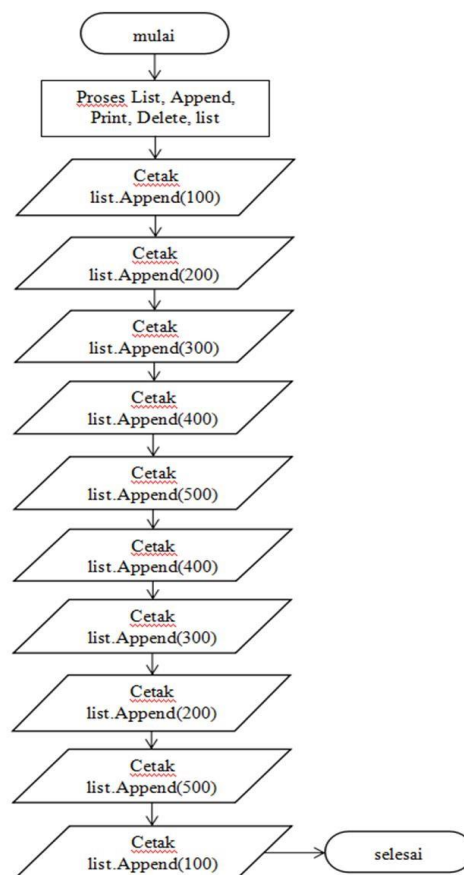
list.Delete(100);

list.Print();

}

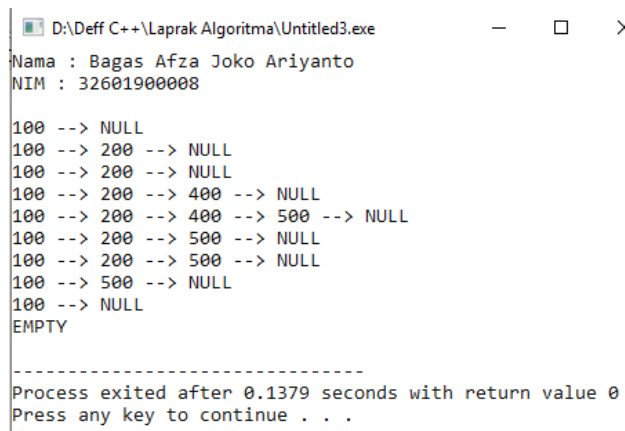
```

b. *Flowchart*



Gambar 5.9 *Flowchart* Tugas 3

c. Output



```

D:\Deff C++\Laprak Algoritma\Untitled3.exe
Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008

100 --> NULL
100 --> 200 --> NULL
100 --> 200 --> NULL
100 --> 200 --> 400 --> NULL
100 --> 200 --> 400 --> 500 --> NULL
100 --> 200 --> 500 --> NULL
100 --> 200 --> 500 --> NULL
100 --> 500 --> NULL
100 --> NULL
EMPTY

-----
Process exited after 0.1379 seconds with return value 0
Press any key to continue . . .

```

Gambar 5.10 Output Tugas 3

d. Penjelasan

Pada gambar 5.10 merupakan hasil dari program diatas. Program diatas merupakan program yang menggunakan bahasa C++. Yang menghasilkan tampilan untuk mencari *Linked List*. Dengan cara memasukkan program kemudian tekan *enter* untuk mengetahui hasilnya.

5.5 Kesimpulan

Dengan mempelajari BAB V, kita dapat mengetahui rumus logika serta fungsi-fungsi dari program *linked list*, *stack*, dan *queue*. Perbandingan antara *linked list*, *queue*, dan *stack* adalah untuk *stack* dan *queue* yang berukuran besar, terutama jumlah maksimal data tidak diketahui, menggunakan *linked list*, untuk perangkat yang memiliki memori terbatas, seperti *small handheld devices*, *linked list* memiliki performa yang bagus. *Linked List* atau dikenal juga dengan sebutan senarai berantai adalah struktur data yang terdiri dari urutan *record* data dimana setiap *record* memiliki *field* yang menyimpan alamat/referensi dari *record* selanjutnya (dalam urutan).

DAFTAR PUSTAKA

Kadir, a. (2010). *ALGORITMA DAN PEMROGRAMAN MENGGUNAKAN C & C++*. YOGYAKARTA.

S Safwandi, N. N. (2014). *Bahan Ajar Struktur Data*. aceh: FAKULTAS TEKNIK UNIVERSITAS MALIKUSSALEH.

Sukrisno, & Utami, E. (2005). *10 Langkah Belajar Logika dan Algoritma, Menggunakan Bahasa C dan C++ di GNU/Linux*. Yogyakarta: ANDI.