

MODUL KULIAH
PEMROGRAMAN KOMPUTER
(MAM1204)



DISUSUN OLEH :

1. Dr. Imran M., M.Sc
2. Zulkarnain, M.Si

PRODI S1 MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS RIAU
2016

February 15, 2016

Contents

0	Pendahuluan	3
1	Bagian Pertama	3
1.1	Sasaran Perkuliahan	3
1.2	Menjalankan Dev-C++	3
1.3	Program Sederhana	3
1.4	Latihan	5
2	Bagian Kedua	7
2.1	Karakter khusus di bahasa C++	7
2.2	Konstanta	7
2.3	Variabel	7
2.4	Mendeklarasikan Variabel dan Memberinya Nilai	8
2.5	Meng-input Bilangan atau Karakter dari Keyboard serta Operasi Aritmatika	10
3	Bagian Ketiga	16
3.1	Compound Assignment 1	16
3.2	Compound Assignment 2	16
3.3	Menampilkan data ke layar monitor	18
4	Bagian keempat	23
4.1	Statement Blok	23
4.2	Latihan	23
4.3	Meng-input Karakter	24
4.4	Latihan	26
4.5	Review	27
5	Bagian kelima	29
5.1	Menggunakan <code>if</code> , <code>else</code>	29
5.2	Operator Relasi pada bahasa C++	29
5.3	Latihan	33
6	Bagian keenam	34
6.1	Penggunaan Perintah <code>switch - case - default</code>	34
6.2	<code>for</code> Loop	37
6.3	Latihan	40
7	Bagian Ketujuh	41
7.1	<code>while</code> loop	41
7.2	<code>do-while</code> loop	44
8	Bagian Kedelapan : Review	45
9	Bagian Kesembilan	47
9.1	Array satu dimensi	47
9.2	Latihan	49

10 Bagian Kesepuluh	51
10.1 Array Dua Dimensi	51
10.2 Latihan	53
11 Bagian Kesebelas	56
11.1 Fungsi	56
11.2 Latihan	60
12 Bagian Keduabelas	61
12.1 Array dalam Statistik	61
12.2 Latihan	62
13 Bagian Ketigabelas	63
13.1 Integral Riemann	63
13.2 Latihan	64

0 Pendahuluan

Hand out kuliah ini merupakan modifikasi dari hand out kuliah *Bahasa C untuk Matematikawan* oleh Dr. Imran M., M. Sc. yang digunakan pada kuliah Pengantar Pemrograman pada Program Studi S1 Matematika FMIPA Universitas Riau tahun 2013.

1 Bagian Pertama

1.1 Sasaran Perkuliahan

Ada 3 sasaran perkuliahan pemrograman komputer yang akan dicapai, yaitu

1. Diakhir perkuliahan anda dapat membuat program untuk masalah yang ditentukan.
2. Diakhir perkuliahan anda dapat membaca program dan menebak output dari program.
3. Diakhir perkuliahan anda dapat menemukan kesalahan dari suatu program yang diberikan.

Untuk mengukur keberhasilan perkuliahan Pemrograman komputer bahasa C++ ini akan dilihat ketiga sasaran perkuliahan ini, melalui Quis, UTS, dan UAS.

1.2 Menjalankan Dev-C++

Untuk memulai menulis program dalam bahasa C++, lakukan langkah-langkah berikut :

1. Klik Start ⇒ All Programs ⇒ Bloodshed Dev-C++ ⇒ Dev-C++ untuk menjalankan Dev-C++. Anda dapat juga menjalankan Dev-C++ dengan mengklik icon Dev-C++ pada desktop.
2. Untuk memulai menulis program, klik File ⇒ New ⇒ Source File.
3. Untuk mengcompile program, klik Execute ⇒ Compile atau cukup menekan tombol F9 pada keyboard anda.
4. Untuk me run program, klik Execute ⇒ Run atau cukup dengan menekan tombol F10 pada keyboard anda.

1.3 Program Sederhana

1. Ketik ulang program berikut ini. Beri nama `Latihan1.cpp`

```
#include <stdio>
#include <stdlib>
#include <iostream>
using namespace std;
int main()
{
    cout << "Ini adalah program C++ sederhana";
    system("PAUSE");
    return 0;
}
```

2. Compile program yang anda buat diatas dengan memilih tab `execute > compile` atau cukup dengan menekan tombol F9.
3. Jika program anda tidak mempunyai kesalahan, maka akan ada tulisan `Compilation succeeded`.
4. Untuk melihat hasil, run program tersebut dengan cara memilih tab `execute > run` atau dengan menekan tombol F10.
5. Amati hasil yang diperoleh.

Secara umum, semua program yang kita buat dalam bahasa C++ mempunyai kerangka dasar (framework) sebagai berikut :

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{

// Mulai tulis program anda di sini

system("PAUSE");
return 0;
}
```

Bagian utama program anda adalah bagian yang berada di dalam tanda kurung kurawal { dan }. Selanjutnya kita akan melakukan beberapa modifikasi terhadap program Latihan1.cpp

1. Coba anda ganti baris ke 7

```
cout << "Ini adalah program C++ sederhana\n";
```

dengan

```
cout << " Ini adalah program C++ sederhana\n";
```

Amati perbedaan outputnya.

2. Coba hapus \n pada baris ke 7, kemudian compile dan run kembali program anda. Perhatikan perbedaan outputnya!
3. Coba hapus salah satu tanda " pada baris ke 7, kemudian compile kembali program anda. Perhatikan pesan error apa yang muncul!
4. Coba hapus ; pada baris ke 7, kemudian compile kembali program anda. Perhatikan pesan error apa yang muncul!
5. Coba hapus { atau } kemudian compile kembali program anda. Perhatikan pesan error yang muncul!
6. Coba hapus #include <cstdlib> pada baris ke 2. Perhatikan pesan error yang muncul!
7. Coba hapus #include <iostream> pada baris ke 3. Perhatikan pesan error yang muncul!
8. Coba ganti #include <iostream> pada baris ke 3 dengan #include (iostream). Perhatikan pesan error yang muncul!
9. Coba ganti main pada baris ke 5 dengan Main. Perhatikan pesan error yang muncul!
10. Ketik program berikut ini dan beri nama Latihan2.cpp kemudian perhatikan outputnya!

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
cout << "Ini adalah program C++";
cout << "sangat";
cout << "sederhana";
system("PAUSE");
return 0;
}
```

11. Ketik program berikut ini dan beri nama `Latihan2a.cpp` kemudian perhatikan perbedaannya dengan program `Latihan2.cpp`

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
cout << "Ini adalah program C++\n";
cout << "sangat\n";
cout << "sederhana\n";
system("PAUSE");
return 0;
}
```

1.4 Latihan

1. Buatlah program sehingga outputnya berspasi ganda (terdapat satu baris kosong diantara dua baris kalimat). Contoh output :

```
Inilah latihan ku yang ketiga
membuat program dalam bahasa C++
```

2. Tanpa membuat programnya terlebih dahulu, tentukan output dari program dibawah ini! Konfirmasi jawaban anda dengan membuat programnya!

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
cout << "Besar ke kecil ...";
cout << "10....";
cout << "9...";
cout << "8..";
cout << "7.";
cout << "\n";
system("PAUSE");
return 0;
}
```

3. Buatlah program dengan output sebagai berikut :

```
*
***
*****
*****
*****
```

4. Buatlah program dengan output sebagai berikut :

```
*****
*****
**      **
**      **
**      **
*****
*****
```

5. Buatlah program dengan output sebagai berikut :

```
BBBB   H   H   SSSSS   CCCC
B   B   H   H   S       C   C
BBBB   HHHHH   SSSSS   C
B   B   H   H       S   C   C
BBBB   H   H   SSSSS   CCCC
```

2 Bagian Kedua

2.1 Karakter khusus di bahasa C++

Pada pertemuan sebelumnya kita telah menggunakan karakter khusus `\n` yang berguna untuk pindah ke baris baru. Selain `\n` bahasa C++ juga menyediakan karakter khusus yang disebut karakter escape, antara lain :

```
\a : untuk bunyi bell (alert)
\b : mundur satu spasi (backspace)
\n : ganti baris baru (new line)
\r : ke kolom pertama, baris yang sama (carriage return)
\t : tab horizontal
\v : tab vertikal
\0 : nilai kosong (null)
\' : karakter petik tunggal
\" : karakter petik ganda
\\ : karakter backslash
\? : karakter tanda tanya
\ooo : bilangan octal
\xhh : bilangan hexadecimal
```

Coba ketik program berikut ini dan perhatikan outputnya.

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    cout << "\t\"Membingungkan\b\b\b\b0000\n\"\\n";
    system("PAUSE");
    return 0;
}
```

Silahkan anda bereksperimen dengan karakter-karakter lain. Coba buat program yang outputnya adalah

```
"So what?" said she.
```

2.2 Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string. Dalam bahasa C konstanta dideklarasikan menggunakan preprocessor `#define`. Contoh pendeklarasiannya adalah :

```
#define Itmax 100
#define PHI 3.14
#define nim "0111500382"
#define nama "Sukainah"
```

2.3 Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan. Nama dari suatu variable dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf atau *underscore* (`_`).

2. Tidak boleh mengandung spasi.
3. Tidak boleh menggunakan simbol \$, ?, %, #, !, &, *, (,), -, +, =
4. Panjang bebas, tetapi hanya 32 karakter pertama yang terpakai.
5. Tidak boleh menggunakan 32 kata kunci yang ada pada bahasa C++, yaitu

```

auto      double  int      struct
break    else     long     switch
case     enum     register typedef
char     extern  return   union
const    float    short    unsigned
continue for      signed   void
default  goto     sizeof   volatile
do       if      static   while

```

2.4 Mendeklarasikan Variabel dan Memberinya Nilai

1. Pada tabel dibawah ini, diberikan tipe-tipe variabel dalam C++ beserta kegunaannya:

Tipe Variabel	Keterangan
<code>int</code>	Bilangan bulat bertanda.
<code>unsigned int</code>	Bilangan bulat non negatif.
<code>long</code>	Bilangan bulat yang lebih besar dari <code>int</code> , tetapi pada dev-C++ tidak ada perbedaan antara <code>long</code> dan <code>int</code> .
<code>unsigned long</code>	long integer non negatif.
<code>float</code>	Bilangan real dengan <i>single precision</i> .
<code>double</code>	Bilangan real dengan <i>double precision</i> .
<code>char</code>	Variabel yang digunakan ununtuk menyimpan karakter tunggal.
<code>string</code>	Variable yang digunakan untuk menyimpan sekumpulan karakter.
<code>bool</code>	Variable yang hanya bernilai true atau false.

2. Bilangan bulat bertanda (`int`) dapat bernilai antara $-2.147.483.647$ sampai $2.147.483.647$. Sementara itu, `unsigned int` dapat bernilai dari 0 sampai $4.294.967.295$.
3. Variabel di bahasa C++ ada dua, variabel global dan variabel lokal. Variabel global dideklarasikan diluar fungsi dan nilainya tetap dapat diakses oleh setiap fungsi. Sedangkan variabel lokal dideklarasikan di dalam suatu fungsi dan tidak ada hubungan dengan variabel lokal fungsi lain. Variabel lokal dideklarasikan ketika fungsi dipanggil.
4. Bilangan pecahan konstan dapat juga disajikan dengan *scientific notation*. Contoh bilangan pecahan konstan dalam sajian ini adalah $1.7e4$, yang berarti 1.7×10^4 . Contoh lainnya adalah $2.25e-3$ yang berarti 2.25×10^{-3} atau 0.00225.
5. Untuk mengetahui berapa *byte* diperlukan masing-masing tipe data diatas untuk storagenya dapat dilihat dari hasil program berikut:

```

#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
cout << "Ukuran storage masing-masing tipe data (dalam bytes) : \n";
cout << "char = " << sizeof(char) << "\n";
cout << "bool = " << sizeof(bool) << "\n";
cout << "int = " << sizeof(int) << "\n";
cout << "long = " << sizeof(long) << "\n";
cout << "float = " << sizeof(float) << "\n";
cout << "double = " << sizeof(double) << "\n";
system("PAUSE");
return 0;
}

```

Bentuk umum pendeklarasian variabel adalah :

Tipe_variabel nama_variabel

Contoh pendeklarasian variabel :

```

int x; // deklarasi variabel x bertipe integer
char y, huruf, nim[10]; // deklarasi variabel bertipe char
float nilai; // deklarasi variabel bertipe float
double beta; // deklarasi variabel bertipe double
int array[5][4]; // deklarasi array bertipe integer

```

Pada contoh deklarasi variabel diatas, anda temukan kalimat yang diawali //, yang merupakan *comment* dari program yang anda tulis. *Comment* ini tidak wajib ada, tetapi membantu untuk mengingat apa maksud kalimat yang anda tulis. Setiap *comment* harus dimulai dengan // dan boleh terdiri dari beberapa baris yang nanti akan anda lihat pada berbagai program yang anda temui pada perkuliahan ini.

Catatan : Dalam memberi nama variabel sebaiknya menggunakan nama yang memberi arti variabel yang digunakan. Hal ini memudahkan kita mengenal kerja dari program yang kita buat. Jika kita membuat program untuk menentukan luas empat persegi panjang dengan sisi-sisinya bilangan bulat, katakan variabelnya `int panjang`, `int lebar` dan `int luas`, untuk menyatakan panjang, lebar, dan luas persegi panjangnya, begitu juga variabel-variabel di program lainnya.

2.5 Meng-input Bilangan atau Karakter dari Keyboard serta Operasi Aritmatika

1. Untuk meng-input bilangan atau karakter dari keyboard, kita gunakan perintah `cin >>` seperti pada contoh dibawah ini

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    cout << "Masukkan nilai a = ";
    cin >> a;
    cout << "Masukkan nilai b = ";
    cin >> b;
    cout << "Nilai variabel a adalah " << a
    << " dan nilai variabel b adalah " << b << endl;
    system("PAUSE");
    return 0;
}
```

Keterangan : Pada program diatas, kita deklarasikan dua variabel, yaitu `a` dan `b` yang masing-masing bertipe integer. Setelah itu, nilai masing-masing variabel ini diperoleh dari input yang diberikan melalui keyboard. Perhatikan apa yang terjadi jika input yang diberikan bukan bilangan integer!

2. Untuk operasi aritmatika, ada lima operator yang tersedia, yaitu

Operator	Artinya
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulo

3. Operator `+`, `-`, `*`, dan `/` dapat digunakan pada sembarang tipe data, akan tetapi operator `%` hanya dapat digunakan untuk tipe data integer. Hal lain yang penting adalah jika `a`, `b` dua bilangan bulat, maka `a/b` akan menghasilkan bilangan bulat dari hasil pembagian. Contohnya :

$$15/6 = 2, \quad 1/3 = 0, \quad 42/10 = 4,$$
$$15\%6 = 3, \quad 1\%3 = 1, \quad 42\%10 = 4.$$

4. Untuk menghilangkan keraguan tentang urutan operasi operator sebaiknya digunakan tanda kurung, seperti `a/b/c` dapat berarti `(a/b)/c` atau `a/(b/c)`. Karena pada bahasa C++ `*` dan `/` dimulai dari yang kiri, maka `a/b/c` jika tidak diberi tanda kurung sama artinya dengan `a/(b*c)`.

Perhatikan contoh berikut ini :

```

//Ketik program dibawah ini. Beri nama latihan251.cpp
//compile kemudian run dan perhatikan outputnya.
//Ini merupakan program operasi aritmatika sederhana
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
//mencetak hasil 9+4
cout << "Nilai dari 9 + 4 = " << 9+4 << "\n";
//mencetak hasil 9-4
cout << "Nilai dari 9 - 4 = " << 9-4 << "\n";
//mencetak hasil 9*4
cout << "Nilai dari 9 * 4 = " << 9*4 << "\n";
//mencetak hasil 9/4
cout << "Nilai dari 9 / 4 = " << 9/4 << "\n";
//mencetak hasil 9%4
cout << "Nilai dari 9 % 4 = " << 9%4 << "\n";
system("PAUSE");
return 0;
}

```

```

//Ketik program dibawah ini. Beri nama latihan252.cpp
//compile kemudian run dan perhatikan outputnya.
//Ini merupakan program penggunaan operator / (bagi) dan % (modulo).
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
//mencetak hasil 9+4
cout << "Nilai dari 9 + 4 = " << 9+4 << "\n";
//mencetak hasil 9-4
cout << "Nilai dari 9 - 4 = " << 9-4 << "\n";
//mencetak hasil 9*4
cout << "Nilai dari 9 * 4 = " << 9*4 << "\n";
//mencetak hasil 9/4
cout << "Nilai dari 9 / 4 = " << 9/4 << "\n";
//mencetak hasil 9%4
cout << "Nilai dari 9 % 4 = " << 9%4 << "\n";
system("PAUSE");
return 0;
}

```

```

//Ketik program dibawah ini. Beri nama latihan253.cpp
//compile kemudian run dan perhatikan outputnya.
//Ini merupakan program penggunaan operator / (bagi) dan % (modulo).
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
cout << "5/2 = " << 5/2 << "\n";
cout << "5 modulo 2 = " << 5%2 << "\n";
cout << "4/2 = " << 4/2 << "\n";
cout << "5 modulo 2 = " << 4%2 << "\n";
system("PAUSE");
return 0;
}

```

```

//Ketik program dibawah ini. Beri nama latihan254.cpp
//compile kemudian run dan perhatikan outputnya.
//Ini merupakan program penggunaan operator / (bagi) dan % (modulo).
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int m=-13, n=3, q=m/n, r=m%n;
cout << "m = " << m << "\n";
cout << "n = " << n << "\n";
cout << "q = " << q << "\n";
cout << "r = " << r << "\n";
cout << "nq + r = " << "(" << n << ")" << " x "
    << "(" << q << ")" << " + (" << r << ") = " << m << "\n";
system("PAUSE");
return 0;
}

```

5. Pada contoh dibawah ini, anda diminta mengamati model penyajian program yang berbeda, tetapi menghasilkan hasil yang sama. Pada program ini anda menjumlahkan dua bilangan yaitu 50 dan 25. Pengetahuan ini diperlukan mengingat style penulisan program tergantung pada penulisnya, dan yang paling baik adalah yang punya informasi lengkap.

```

//Ketik program dibawah ini. Beri nama latihan255.cpp
//compile kemudian run dan perhatikan outputnya.
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int jumlah;
jumlah = 50+25;
cout << "Jumlah dari 50 dan 25 adalah " << jumlah << "\n";
system("PAUSE");
return 0;
}

```

```
//Ketik program dibawah ini. Beri nama latihan256.cpp
//compile kemudian run dan perhatikan outputnya.
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int nilai1, nilai2, jumlah;
nilai1=50;
nilai2=25;
jumlah = nilai1+nilai2;
cout << "Jumlah dari " << nilai1 << " dan "
    << nilai2 << " adalah " << jumlah << "\n";
system("PAUSE");
return 0;
}
```

6. Pemberian nilai variabel dapat dilakukan pada saat memberikan nilai pada variabel lain yang dikenal dengan statement berantai, seperti contoh berikut.

```
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int m,n;
m = (n = 25) + 11;
cout << "m = " << m << " dan n = " << n << "\n";
system("PAUSE");
return 0;
}
```

Pada contoh diatas, diberi nilai $n = 25$ kemudian nilai $n = 25$ ditambah dengan 11 untuk mendapatkan nilai m . Akan tetapi statement berantai sebaiknya dihindari. Program diatas sebaiknya ditulis

```
int m,n = 25;
m=n+11;
```

7. Buatlah program yang mempunyai variabel a , b , c yang bertipe bilangan bulat dan bernilai masing-masing $a=4$, $b=5$, $c=2$. Kemudian hitung tiap ekspresi berikut dan tampilkan hasilnya :

- (a) $a\%b-5$
- (b) $c-b*a$
- (c) $c*c+b*a/3$
- (d) $a-b-c*a\%b$

Sebelum anda run, tebaklah hasil yang anda dapatkan.

8. Sekarang kita akan membuat program untuk menghitung luas persegi panjang dengan panjang p dan lebar l . Pertama kita harus asumsikan dulu tipe dari variabel p dan l tersebut. Misalkan sekarang p dan l merupakan bilangan desimal (tipe variabelnya `float` atau `double`). Contoh programnya adalah sebagai berikut

```

//program menghitung luas persegi panjang.
//berinama program ini dengan latihan257.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
float p,l;
cout << "panjang sisi persegi panjang = ";
cin >> p;
cout << "lebar sisi persegi panjang = ";
cin >> l;
cout << "luas persegi panjang = " << p*l << "\n";
system("PAUSE");
return 0;
}

```

Catatan 1 : Pada program diatas, kita deklarasikan panjang dan lebar persegi panjang sebagai variabel bertipe pecahan (`float`).

Catatan 2 : Perhatikan penggunaan `cin >> p`; Disini kita diminta untuk menginputkan bilangan dari keyboard. Bilangan tersebut akan kita simpan sebagai nilai dari variabel `p`. Begitu juga penggunaan `cin >> l`; Disini kita juga menginputkan bilangan dari keyboard dan kita simpan sebagai nilai dari variabel `l`.

9. Untuk membuat program menghitung luas sebuah lingkaran dengan panjang jari jari r , dimana r merupakan suatu bilangan tertentu, kita menentukan terlebih dahulu tipe variabel r yang akan digunakan. Untuk bilangan r berbentuk desimal dapat dilakukan sebagai seperti pada program latihan258.cpp berikut ini. Pada program ini, kita deklarasikan konstanta PHI yang bernilai 3.14 (karena Dev-C++ tidak mengenal konstanta PHI)

```

//program menghitung luas lingkaran.
//berinama program ini dengan latihan258.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#define PHI 3.14
using namespace std;
int main()
{
float r;
cout << "panjang jari-jari lingkaran, r = ";
cin >> r;
cout << "luas lingkaran dengan r = " << r
    << " adalah " << r*r*PHI << "\n";
system("PAUSE");
return 0;
}

```

Catatan 1 : Pada program ini, kita mendeklarasikan suatu konstanta PHI yang bernilai 3.14. Perhatikan bahwa pendeklarasian tidak menggunakan tanda sama dengan (`=`), juga tidak diakhiri dengan titik koma ;

Catatan 2 : Perhatikan penggunaan `cin >> r`; Disini kita diminta untuk menginputkan bilangan dari keyboard. Bilangan tersebut akan kita simpan sebagai nilai dari variabel `r`. Tipe dari variabel `r` ini adalah bilangan desimal (`float`) hal ini terlihat dari deklarasi `float r`;

Catatan 3 : Output program kita adalah kalimat "luas lingkaran dengan r = adalah". Perhatikan baik-baik bagaimana kita melakukan ini.

10. Buatlah program yang dapat digunakan untuk menghitung volume suatu balok dengan panjang p , lebar l , dan tinggi t . Input untuk p , l , t harus bilangan desimal (`float`). Berinama program anda dengan `volumebalok.cpp`
11. Setelah anda sukses membuat program `volumebalok.cpp`, buatlah program yang dapat digunakan untuk menghitung volume dan luas selimut suatu bola, dengan jari-jari r . Input untuk r haruslah bilangan desimal. Berinama program anda dengan `bola.cpp`

$$\begin{aligned}\text{Volume Bola} &= \frac{4}{3}\pi r^3 \\ \text{Luas selimut bola} &= 4\pi r^2\end{aligned}$$

12. Buatlah program yang dapat digunakan untuk menghitung luas sembarang segitiga yang panjang sisi-sisinya masing-masing a , b dan c yang di input dari keyboard dan berbentuk bilangan desimal. Formula untuk menghitung luas sebarang segitiga ini dikenal dengan Formula Heron yang diberikan oleh

$$\begin{aligned}s &= \frac{1}{2}(a + b + c) \\ \text{Luas} &= \sqrt{s(s - a)(s - b)(s - c)}\end{aligned}$$

3 Bagian Ketiga

Disini kembali diingatkan bahwa ada 3 sasaran perkuliahan pemrograman komputer yang harus dicapai, yaitu

1. Diakhir perkuliahan anda dapat membuat program untuk masalah yang ditentukan.
2. Diakhir perkuliahan anda dapat membaca program dan membak output dari program.
3. Diakhir perkuliahan anda dapat menemukan kesalahan dari suatu program yang diberikan.

3.1 Compound Assignment 1

Dalam bahasa C++ pernyataan berbentuk $a = a + 4$ dapat disingkat menjadi $a += 4$. Secara umum,

$$a \text{ op} = b \text{ ekuivalen dengan } a = a \text{ op } b$$

Cara penulisan pernyataan gabungan ini sangat banyak digunakan pada looping yang akan dipelajari pada pertemuan-pertemuan berikutnya. Coba tebak output program berikut kemudian ketik, *compile* dan *run* program anda, bandingkan hasil output program dengan tebakan output yang anda tulis sebelumnya.

```
#include <cstdlib>
#include <stdlib.h>
#include <iostream>
using namespace std;
int main()
{
int a = 0;
a += 4;
cout << "Nilai a adalah " << a << "\n";
a *= 3;
cout << "Nilai a sekarang adalah " << a << "\n";
a -= 4;
cout << "Nilai a sekarang adalah " << a << "\n";
a /= 2;
cout << "Nilai a sekarang adalah " << a << "\n";
a %= 5;
cout << "Nilai a sekarang adalah " << a << "\n";
system("PAUSE");
return 0;
}
```

3.2 Compound Assignment 2

Dalam bahasa C++, terutama dalam *looping* kita sering diharuskan menambah nilai suatu variabel tertentu dengan 1, misalnya kita sering mengevaluasi $a=a+1$. Dalam bahasa C++ cara penulisan ini dapat dilakukan dengan berbagai cara, yaitu

$$a = a + 1 \quad \text{atau} \quad a + = 1 \quad \text{atau} \quad a + +$$

Ketiga cara diatas memberikan hasil yang sama, yaitu menambahkan satu ke variabel a .

1. Coba tebak output dari program berikut ini, kemudian verifikasi jawaban anda dengan mengetik, *compile*, dan *run* program tersebut. Bandingkan hasil yang anda peroleh dengan tebakan output yang anda buat.

```

//beri nama program ini dengan latihan321.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int m = 4, n;
n = ++m * --m;
cout << "m=" << m << "dan n=" << n << "\n";
system("PAUSE");
return 0;
}

```

2. Coba tebak output program berikut ini, kemudian ketik, *compile*, dan *run* program anda. Bandingkan hasil yang anda peroleh dengan tebakan anda sebelumnya.

```

//beri nama program ini dengan latihan322.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int a = 3, b;
b = a++;
cout << "b = " << b << "dan a = " << a << "\n";
b = ++a;
cout << "Sekarang b = " << b << "dan a = " << a << "\n";
b = 5 % --a;
cout << "Sekarang b = " << b << "dan a = " << a << "\n";
cout << "Sekarang b = " << ++b << "dan a = " << a-- << "\n";
cout << "Sekarang b = " << b << "dan a = " << a << "\n";
system("PAUSE");
return 0;
}

```

3. Coba tebak output program berikut ini, kemudian ketik, *compile*, dan *run* program anda. Bandingkan hasil yang anda peroleh dengan tebakan anda sebelumnya.

```

//beri nama program ini dengan latihan323.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int int1 = 4, int2 = 7, int3;
int3 = ++int1 * --int2;
cout << ++int1 << int2-- << --int3 << "\n";
cout << int1 << int2 << int3;
system("PAUSE");
return 0;
}

```

3.3 Menampilkan data ke layar monitor

Pada pertemuan-pertemuan sebelumnya kita telah membahas penggunaan `cout <<` yang digunakan untuk menampilkan data di layar monitor. Data yang kita tampilkan dapat berupa kalimat atau nilai suatu variabel. Sekarang kita akan membahas bagaimana cara mengatur format data yang kita tampilkan di layar komputer. Format yang ingin kita atur meliputi banyak digit yang di tampilkan, lebar karakter yang digunakan, serta jenis penyajian bilangan yang di tampilkan. Untuk melakukan ini, kita gunakan deklarasi sebagai berikut

```
cout << flag << setprecision(n) << setw(m) << argument
```

Keterangan dari deklasi di atas dapat dilihat pada tabel berikut ini :

Format	Artinya
<code>flag</code>	Cara penyajian data, boleh dikosongkan saja, atau diberi nilai <code>fixed,scientific,hex,oct,dec</code>
<code>setprecision(n)</code>	menyatakan banyak digit
<code>setw(m)</code>	menyatakan panjang karakter yang digunakan
<code>argument</code>	nilai atau variabel yang akan ditampilkan

Penjelasan tentang jenis-jenis flag beserta kegunaannya dapat dilihat pada tabel di bawah ini :

flag	Kegunaannya
kosong (default)	Jika <code>flag</code> dikosongkan, maka format <code>setprecision(n)</code> akan menampilkan bilangan dengan paling banyak <code>n</code> digit sebelum dan sesudah tanda koma.
<code>fixed</code>	Bilangan yang ditampilkan akan berupa bilangan dengan tepat <code>n</code> digit setelah koma sesuai dengan nilai <code>n</code> pada <code>setprecision(n)</code>
<code>scientific</code>	Bilangan yang ditampilkan selalu mempunyai satu digit sebelum koma, kemudian diikuti dengan <code>n</code> digit dibelakang koma sesuai dengan nilai <code>n</code> pada <code>setprecision(n)</code> , kemudian diikuti dengan bagian eksponensial yang dinyatakan dengan huruf <code>e</code> dan tanda <code>+</code> atau <code>-</code> serta tiga digit yang menyatakan pangkat.
<code>hex,oct,dec</code>	Menampilkan bilangan dengan format berturut-turut hexadecimal, octal, dan decimal.

Catatan : Jika tidak menggunakan `setprecision`, maka

1. `flag` default akan menampilkan bilangan dengan total 6 digit dihitung mulai dari digit pertama dari kiri, jika bilangan sebelum koma lebih dari 6 digit, maka bilangan akan ditampilkan dalam format *scientific* dengan 5 digit setelah koma.
2. `flag` `fixed` akan menampilkan bilangan dengan tepat 6 digit sesudah koma,
3. `flag` `scientific` akan menampilkan bilangan enam digit sesudah koma, diikuti tanda eksponensial `e` dan 3 digit pangkat berserta tanda nya (`+` atau `-`).

Perhatikan program berikut ini

```
//beri nama program ini dengan latihan331.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
int nilai;
cout << "Masukkan nilai anda :";
cin >> nilai;
cout << "Nilai anda dalam bilangan decimal adalah = "
    << dec << nilai << endl;
cout << "Nilai anda dalam bilangan hexadecimal adalah = "
    << hex << nilai << endl;
cout << "Nilai anda dalam bilangan octal adalah = "
    << oct << nilai << endl;
cout << "Anda memperoleh nilai " << setprecision(3)
    << setw(5) << nilai << endl;
cout << "apakah Anda telah puas mendapat nilai "
    << setprecision(2) << setw(6) << nilai << endl;
system("PAUSE");
return 0;
}
```

Ketik kemudian compile dan run program diatas. Perhatikan outputnya untuk input berupa bilangan 8, 12, 123, 2345. Bagaimana jika inputnya berupa bilangan berkoma? Jelaskan hasil pengamatan anda.

Sekarang ganti tipe variabel `nilai` pada program diatas dengan `float`. Perhatikan outputnya untuk input bilangan 20.123, 87.385, 67.976, 67.917, 99.958, 99.928, 15, 25. Coba bereksperimen dengan mengganti nilai pada fungsi `setprecision` dan `setw`.

Sekarang coba anda perhatikan program di bawah ini. Sebelum anda run program tersebut, coba tebak dulu outputnya. Setelah itu baru ketik, compile dan run program tersebut dan bandingkan hasil yang anda peroleh dengan tebakan sebelumnya.

```
//beri nama program ini dengan latihan332a.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
int i=425;
short int j=17;
unsigned int u=0xf179;
long int l=75000;

cout << "BILANGAN BULAT :\n";
cout << setw(4) << dec << i << setw(4)
    << oct << i << setw(4) << hex << i << endl;
cout << setw(4) << hex << i << setw(4)
    << hex << uppercase << i << nouppercase << endl;
cout << setw(4) << dec << j << setw(4)
    << oct << j << setw(4) << hex << j << endl;
cout << setw(8) << dec << u << setw(8)
    << oct << u << setw(8) << hex << u << endl;
cout << setw(8) << dec << l << setw(8)
    << oct << l << setw(8) << hex << l << endl;
system("PAUSE");
return 0;
}
```

```
//beri nama program ini dengan latihan332b.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
float f=12.978;

cout << "\nBILANGAN PECAHAN float dan double :\n";
//berbagai cara penyajian bilangan f dengan tipe float
cout << f << endl;
cout << fixed << f << endl;
cout << scientific << f << endl;
system("PAUSE");
return 0;
}
```

```

//beri nama program ini dengan latihan332c.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
float f=12.978;

//menggunakan setprecision(n)
cout << setprecision(4) << f << endl;
cout << fixed << setprecision(3) << f << endl;
cout << scientific << setprecision(3) << f << endl;
system("PAUSE");
return 0;
}

```

```

//beri nama program ini dengan latihan332d.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
float f=12.978;

//menuliskan bilangan dengan format berbeda pada baris yang sama
//dengan menggunakan setw(n)
cout << setprecision(4) << setw(10) << f
    << fixed << setprecision(3) << setw(10) << f
    << scientific << setprecision(3) << setw(15) << f << endl;
system("PAUSE");
return 0;
}

```

```

//beri nama program ini dengan latihan332e.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
double d=-97.4583;
//berbagai cara penyajian bilangan d dengan tipe double
cout << d << endl;
cout << fixed << d << endl;
cout << scientific << d << endl;
system("PAUSE");
return 0;
}

```

```
//beri nama program ini dengan latihan332f.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
char c='X';
static char s[]="abcdefghijklmnopqrstuvwxy";
double d=-97.4583;
//output dengan justifikasi kiri dan kanan
cout << setw(15) << left << d << endl;
cout << setw(15) << right << d << endl;
cout << "\nKARAKTER :\n";
cout << c << endl;
cout << setw(3) << c << setw(3) << c << endl;
cout << "\nSTRING :\n";
cout << s << endl;
system("PAUSE");
return 0;
}
```

4 Bagian keempat

4.1 Statement Blok

Disini kita akan mempelajari pengaruh penulisan program secara blok. Hal ini diperlukan untuk mengetahui struktur bagaimana Program C++ mengenal nilai suatu variabel pada tiap bloknnya. Untuk memudahkan kita memahami ini pelajari dengan seksama program berikut

```
// beri nama dengan latihan411.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i = 3;
    {
        int i =5;
        cout << "Nilai i di inner blok adalah " << i << endl;
    }
    cout << "Nilai i di outer blok adalah " << i << endl;
    system("PAUSE");
    return 0;
}
```

Catatan 1: Di sini `i` merupakan variabel lokal di inner block. Memory yang dialokasikan untuk `i` pada blok ini hanya aktif ketika blok tersebut dilalui, setelah itu memori untuk variabel `i` di blok ini tidak aktif. Jika ada nama variabel yang sama dengan blok sebelumnya maka nama variabel dalam bloklah yang berlaku.

4.2 Latihan

1. Coba hapus `int i = 5;` pada program di atas. Sebelum anda `run` program tersebut, prediksilah apa yang akan terjadi, misalnya program gagal ter `compile` dan lain-lain. Kemudian `run` program anda tersebut, apa yang anda dapatkan dan bandingkan dengan prediksi anda.
2. Jika pada program di atas, `int i = 3;` dihapus, prediksilah apa yang akan terjadi, misalnya program gagal ter `compile` dan lain-lain. Kemudian `run` program anda tersebut, apa yang anda dapatkan dan bandingkan dengan prediksi anda.
3. Jika pada program di atas, `int i = 5;` diganti dengan `i = 5;`, prediksilah apa yang akan terjadi, misalnya program gagal ter `compile` dan lain-lain. Kemudian `run` program anda tersebut, apa yang anda dapatkan dan bandingkan dengan prediksi anda.

4. Program berikut mengilustrasikan bagaimana pendeklarasian variabel dalam suatu blok tidak dikenal di luar blok :

```
//beri nama dengan latihan412
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    {
        int i;
        cout << "Masukkan nilai i = ";
        cin >> i;
    }
    cout << "Nilai i adalah " << i << endl;
    system("PAUSE");
    return 0;
}
```

Perhatikan program diatas. Perbaiki agar program tersebut dapat di run.

4.3 Meng-input Karakter

Pada pertemuan sebelumnya kita telah membahas fungsi

```
cin >> variabel;
```

yang digunakan untuk menginput nilai variabel. Variabel yang di input dapat mempunyai tipe apapun. Perhatikan program dibawah ini. Ketik, compile dan run program tersebut. Pahami setiap perintah yang dilakukan.

```
// beri nama dengan latihan421.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    string nama;
    cout << "Masukkan nama Anda : ";
    cin >> nama;
    cout << "Hello, Nama Anda adalah " << nama << endl;
    system("PAUSE");
    return 0;
}
```

Sekarang kita akan membahas penggunaan fungsi lain yang juga digunakan untuk menginput nilai variabel dari keyboard. Fungsi yang akan kita bahas adalah `getchar()`, `getch()`, dan `getche()`. Untuk menggunakan fungsi ini, kita harus memasukkan library `<conio.h>` pada header, yaitu menambahkan `#include<conio.h>` pada program.

1. `getchar()`

- (a) Fungsi `getchar()` digunakan untuk membaca data yang bertipe karakter. `getchar()` dapat juga digunakan untuk membaca *satu karakter* dan dapat dipanggil secara berulang, dalam hal ini tidak diperlukan argument.
- (b) Harus diakhiri dengan penekanan tombol enter.

- (c) Karakter yang dimasukkan terlihat pada layar.
- (d) Pergantian baris secara otomatis.
- (e) Penggunaan dua deklarasi dibawah ini mempunyai makna yang sama

```
variabel=getchar()
cin >> variabel;
```

tetapi yang pertama hanya dapat digunakan untuk variabel bertipe karakter.

2. getch(), dan getche()

- (a) Fungsi getch() dan getche() digunakan untuk membaca data karakter.
- (b) Karakter yang dimasukkan tidak perlu diakhiri dengan penekanan tombol enter.
- (c) Jika menggunakan fungsi getch() karakter yang dimasukkan tidak akan ditampilkan pada layar sehingga sering digunakan untuk meminta inputan berupa password.
- (d) Sedangkan pada getche() karakter yang dimasukkan akan ditampilkan pada layar.

Untuk lebih memahami dua fungsi diatas, perhatikan program di bawah ini

```
// beri nama dengan latihan422.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
char huruf1, huruf2, huruf3;
cout << "Masukkan sebuah karakter : ";
huruf1=getchar(); // karakter diinput terlihat di layar
cout << "\nKarakter yang Anda masukkan adalah " << huruf1;
cout << "\nMasukkan sebuah karakter lagi : ";
huruf2=getch(); // karakter diinput tidak terlihat di layar
cout << "\nKarakter yang Anda masukkan adalah : " << huruf2 << endl;
cout << "\nMasukkan sebuah karakter lagi : ";
huruf3=getche(); // karakter diinput terlihat di layar
cout << "\nKarakter yang Anda masukkan adalah : " << huruf3 << endl;
system("PAUSE");
return 0;
}
```

4.4 Latihan

Ketik program berikut ini. Program ini memiliki berbagai kesalahan, perbaiki semua kesalahan tersebut, dan sebelum anda compile dan run tebaklah hasil yang akan anda peroleh. Bandingkan hasil yang anda peroleh.

```
// beri nama dengan latihan43.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <conio.h>
using namespace std
int main{}
{
int umur
char nama;
cout << "Masukkan nama anda : ";
cin << nama;
cout << "Masukkan umur anda : ";
cin << umur;
cout << "Nama anda : " << nama;
cout << "Umur anda : " << umur;
system("PAUSE");
return 0;
}
```

4.5 Review

1. Untuk menyegarkan kembali sebagian materi yang sudah didiskusikan pada pertemuan sebelumnya, coba tuliskan output program berikut dan kemudian, ketik dieditor dan run. Coba bandingkan dengan seksama hasil output yang anda tuliskan dengan output program.

```
// beri nama dengan latihan441.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int x=1, y, z;
cout << "Nilai x = " << x << endl;
cout << "Nilai 2*x + 5 = " << 2*x+5 << endl;
x=5;
cout << "Nilai x sekarang adalah = " << x << endl;
y=2*x++ + 1;
cout << "Nilai y adalah = " << y << endl;
cout << "x = " << x << " dan y = " << y << endl;
z=y=4*x+5;
cout << "y = " << y << " dan z = " << z << endl;
system("PAUSE");
return 0;
}
```

2. Bila pada suatu program diberi inisiasi `int m = -123` dan `double n = 234.1234567`, coba anda tulis output dari fungsi `cout` berikut ini (asumsikan tiap baris merupakan penggalan dari program yang berbeda)

```
cout << "m = " << m << endl;
cout << "m = " << setprecision(7) << setw(6) << m << endl;
cout << "m = " << left << setprecision(7) << setw(6) << m << endl;
cout << "n = " << n << endl;
cout << "n = " << fixed << n << endl;
cout << "n = " << scientific << n << endl;
cout << "n = " << setprecision(5) << n << endl;
cout << "n = " << setprecision(5) << fixed << n << endl;
cout << "n = " << setprecision(5) << scientific << n << endl;
cout << "n = " << setprecision(5) << setw(12) << n << endl;
cout << "n = " << setprecision(5) << setw(12) << fixed << n << endl;
cout << "n = " << setprecision(5) << setw(12) << scientific << n << endl;
```

3. Coba teliti program berikut ini. Bagaimana output yang anda harapkan?

```
// beri nama dengan latihan443.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    char c,d;
    c='d';
    d=c;
    cout << "d = " << d << endl;
    system("PAUSE");
    return 0;
}
```

4. Tulislah program untuk menghitung nilai polinomial

$$3x^3 - 5x^2 + 6$$

untuk $x = 2.55$. Kemudian modifikasi program tersebut sehingga seseorang dapat menghitung nilai polinomial tersebut pada nilai x yang ditentukannya, dimana nilai x tersebut di inputkan oleh user.

5. Tulislah program untuk menghitung nilai expresi berikut:

$$(3.31 \times 10^{-8} + 2.10 \times 10^{-7}) / (7.16 \times 10^6 + 2.10 \times 10^8)$$

kemudian tampilkan output anda di screen.

5 Bagian kelima

5.1 Menggunakan if, else

Struktur umum penggunaan if, else adalah

```
if (kondisi)
    pernyataan1
else {
    pernyataan2
    pernyataan3
}
```

```
if (kondisi){
    pernyataan1
    pernyataan2
}
else
    pernyataan3
```

dan sesudah else masih bisa ditambahkan kondisi if,else sehingga bentuknya menjadi bersarang (*nested*) seperti berikut

```
if (kondisi1){
    pernyataan1
    pernyataan2
}
else if(kondisi2){
    pernyataan3
    pernyataan4
    pernyataan5
}
else
    pernyataan6
```

Perhatikan bahwa bahwa jika pada satu kondisi terdapat dua pernyataan yang berbentuk statement block maka kedua pernyataan tersebut harus terletak antara dua kurung kurawal { · · }. Perhatikan juga hierarki dalam penulisan program tersebut.

5.2 Operator Relasi pada bahasa C++

Operator logika yang tersedia di C++ adalah

Operator	Artinya
==	Sama dengan
!=	Tidak sama
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama
<=	Lebih kecil atau sama
&&	AND
	OR
!	NOT

Untuk memahami penggunaan if,else, perhatikan beberapa contoh berikut ini :

1. Masalah menentukan nilai maksimum dari 3 buah bilangan pecahan yang diinput dari keyboard: Program berikut meminta seseorang untuk mengentri 3 bilangan dari keyboard. Setelah ketiga bilangan tersebut dientri, program akan mengeluarkan output bilangan terbesar yang diinput dan bila bilangan yang maksimum ini positif katakan bilangan tersebut adalah bilangan positif, dan sebaliknya katakan bilangan tersebut adalah bilangan negatif. Untuk memudahkan kita membuat program kita buat dahulu algoritma dari masalah yang akan kita buat programnya ini.

Algoritma: Menentukan maksimum dari 3 bilangan

Katakan tiga bilangan yang akan kita cari maksimumnya adalah a, b, c dan nilai maksimumnya dengan max.

1.	INPUT a, b, c;
2.	OUTPUT max;
3.	PROSES
4.	if((a>b)&&(a>c))
5.	max=a;
6.	if((b>a)&&(b>c))
7.	max=b;
8.	if((c>a)&&(c>b))
9.	max=c;
10.	outputkan max;
11.	if(max>0)
12.	outputkan Bil tsb adalah bil positif;
13.	if(max<0)
14.	outputkan Bil tsb adalah bil negatif

Bila algoritma diatas diterjemahkan kedalam bahasa C++, kita peroleh program berikut ini :

```
// beri nama dengan latihan521.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    float a,b,c,max;
    cout << "Masukkan bilangan 1 dan tekan enter: ";
    cin >>a;
    cout << "Masukkan bilangan 2 dan tekan enter: ";
    cin >>b;
    cout << "Masukkan bilangan 3 dan tekan enter: ";
    cin >>c;
    if((a>b)&&(a>c))
        max=a;
    if((b>a)&&(b>c))
        max=b;
    if((c>a)&&(c>b))
        max=c;
    cout << "Bilangan terbesar : " << max;
    if(max>0)
        cout << "\nBil tsb adalah bil positif" << endl;
    if(max<0)
        cout << "\nBil tsb adalah bil negatif" << endl;
    system("PAUSE");
    return 0;
}
```

Output program berikut ini adalah jika suatu bilangan di-input di keyboard lebih besar dari nol dikatakan positif, jika bilangan tersebut kecil dari nol dikatakan negatif dan bila sama dengan nol dikatakan nol. Jadi disini ada kondisi lebih besar, lebih kecil dan sama dengan.

```
// beri nama dengan latihan522.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int a;
    cout << "Silahkan input bilangan integer : ";
    if (a>0){
        cout << a << " adalah positif\n";
    }
    else if(a<0){
        cout << a << " adalah negatif\n";
    }
    else {
        cout << a << " adalah nol\n";
    }
    system("PAUSE");
    return 0;
}
```

Perhatikan bahwa bila program tersebut dijalankan kemudian kita memasukan nilai bilangan 60 maka perintah mencetak 60 adalah bilangan positif, bila yang kita masukkan -45, perintah mencetak -45 adalah bilangan negatif dan bila kita masukkan 0 perintah mencetak 0 adalah nol. Juga diperhatikan bagaimana penulisan susunan perintah sehingga dapat terbaca dengan jelas.

Untuk memahami bagaimana ekspresi dan relasi di Program berlaku, perhatikan output program berikut :

```
// beri nama dengan latihan523.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int x = 3;
bool a,b,c,d,e,f;
//set output bool menjadi true atau false. Output defaultnya adalah 0 atau 1.
cout.setf(cout.boolalpha);
a = x==3;
b = x != 3;
c = 3*x - 4 <=3;
d = x >= 3;
e = 2*x % 3 > 3;
f = 25 / (2*x) < 3;
cout << "Nilai (x == 3) adalah " << a;
cout << "\nNilai (x != 3) adalah " << b;
cout << "\nNilai (3*x - 4 <= 3) adalah " << c;
cout << "\nNilai (x >= 3) adalah " << d;
cout << "\nNilai (2*x % 3 > 3) adalah " << e;
cout << "\nNilai (25 / (2*x) < 3) adalah " << f << endl;
system("PAUSE");
return 0;
}
```

5.3 Latihan

1. Buatlah program dengan input tiga bilangan oleh user dan mengoutputkan bilangan terkecil dari ketiga bilangan tersebut.
2. Buatlah program untuk menentukan nilai mutlak suatu bilangan desimal yang diinput oleh user. Nilai mutlak suatu bilangan didefinisikan dengan

$$|x| = \begin{cases} x, & \text{jika } x \geq 0 \\ -x, & \text{jika } x < 0 \end{cases}$$

3. Buatlah program untuk menentukan nilai mutlak suatu fungsi $f(x) = \frac{x(x-5)}{2}$ yang diberi input bilangan desimal oleh user yang didefinisikan sebagai berikut:

$$|f(x)| = \begin{cases} f(x), & \text{jika } f(x) \geq 0 \\ -f(x), & \text{jika } f(x) < 0 \end{cases}$$

4. Buatlah program dengan input oleh user nilai yang bersangkutan dan mengoutputkan hasil konversi nilai ujian user tersebut dari angka ke huruf dengan aturan sebagai berikut:

nilai ≥ 85	A
$70 \leq$ nilai < 85	B
$55 \leq$ nilai < 70	C
$40 \leq$ nilai < 55	D
nilai < 40	E

5. Buatlah program yang menginput nama seseorang, kemudian umurnya. Kemudian keluarkan outputnya jika umur yang dientrikan kurang dari 5 tahun dikatakan yang bersangkutan balita, jika umur yang dientrikan tersebut antara 5 sampai 12 yang bersangkutan dikatakan anak-anak dan jika umur yang dientrikan 13-17 dikatakan remaja, jika umur yang dientrikan 18-25 yang bersangkutan dikatakan pemuda, jika umur yang dientrikan 26-60 yang bersangkutan dikatakan Dewasa, dan jika umur yang dientrikan lebih besar dari 60 tahun yang bersangkutan dikatakan lansia. Contoh output:

Masukkan nama anda: Sukainah
Masukkan umur anda (dlm tahun) : 2.6
Anda adalah seorang balita.

6. Buatlah program untuk menentukan akar-akar dari fungsi kuadrat yang berbentuk $ax^2 + bx + c = 0$ dengan menggunakan rumus ABC, untuk kasus $b^2 - 4ac \geq 0$, bila nilai a, b, c di input oleh user. Berikan informasi kepada user bila $b^2 - 4ac < 0$ dengan mengatakan "Program ini khusus untuk kasus $b^2 - 4ac \geq 0$ ". Rumus ABC untuk menentukan akar-akar dari fungsi kuadrat $ax^2 + bx + c = 0$ diberikan oleh

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{dan} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

7. Kembangkan kasus program nomor 6 untuk kasus $b^2 - 4ac > 0$, $b^2 - 4ac = 0$, dan $b^2 - 4ac < 0$, tes program anda untuk setiap kasus.
8. Buatlah program untuk menerima input dari user berupa bilangan bulat yang selanjutnya diuji untuk menentukan bilangan tersebut genap atau ganjil. Hasil pengujian ditampilkan di layar, misalnya "26 adalah bilangan genap."
9. Buatlah sebuah program yang menerima input dari keyboard dua buah bilangan bulat. Kemudian jika hasil kali kedua bilangan yang di input itu bilangan genap, maka akan di output kan di layar monitor **HASIL KALI BILANGAN TERSEBUT GENAP**, jika hasil kali kedua bilangan di input itu bilangan ganjil, maka akan di output kan di layar monitor **HASIL KALI BILANGAN TERSEBUT GANJIL**, dan jika hasil kali kedua bilangan di input itu NOL, maka akan di output kan di layar monitor **SALAH SATU BILANGAN YANG ANDA INPUT NOL**.

6 Bagian keenam

6.1 Penggunaan Perintah `switch - case - default`

Dalam komputasi matematika penggunaan `switch - case` tidaklah terlalu sering, tetapi untuk mengenal strukturnya kita diskusikan pada pertemuan kali ini, sehingga anda mengenalnya. Struktur kondisi `switch-case-default` digunakan untuk penyeleksian kondisi dengan kemungkinan yang terjadi cukup banyak. Struktur ini akan melaksanakan salah satu dari beberapa pernyataan `case` tergantung nilai kondisi yang ada di dalam `switch`. Selanjutnya proses diteruskan hingga ditemukan pernyataan `break`. Jika tidak ada nilai pada `case` yang sesuai dengan nilai kondisi, maka proses akan diteruskan kepada pernyataan yang ada di bawah `default`.

Bentuk umum dari struktur kondisi ini adalah

```
switch(kondisi)
{
    case 1 : pernyataan-1;
    break;
    case 2 : pernyataan-2
    break;
    :
    case n : pernyataan-n
    break;
    default : pernyataan-m
}
```

Coba perhatikan program berikut ini. Pahami struktur yang digunakan.

```
// beri nama dengan latihan611.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int hari;
    cout << "Menentukan nama hari\n";
    cout << "1=Senin,2=Selasa,3=Rabu,4=Kamis,5=Jumat,6=Sabtu,7=Minggu";
    cout << "\nMasukkan kode hari (1-7) : "; cin >> hari;
    switch(hari)
    {
        case 1 : cout << "Hari Senin\n" ;
                break;
        case 2 : cout << "Hari Selasa\n" ;
                break;
        case 3 : cout << "Hari Rabu\n" ;
                break;
        case 4 : cout << "Hari Kamis\n" ;
                break;
        case 5 : cout << "Hari Jumat\n" ;
                break;
        case 6 : cout << "Hari Sabtu\n" ;
                break;
        case 7 : cout << "Hari Minggu\n" ;
                break;
        default : cout << "Kode hari yang anda masukkan SALAH\n";
    }
    system("PAUSE");
    return 0;
}
```

Keterangan : Bila program di atas kita run, dan kita memasukkan kode hari =1, maka akan tercetak "Hari Senin", jika kode hari =2, maka akan tercetak Selasa, dan seterusnya. Perhatikan bahwa jika kode hari yang kita masukkan bukan merupakan bilangan antara 1-7, maka akan tercetak "Kode hari yang anda masukkan SALAH".

Selanjutnya perhatikan dengan seksama contoh kedua di bawah ini :

```
// beri nama dengan latihan612.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int nilai;
    cout << "Masukkan bilangan 0-9 : ";
    cin >> nilai;
    switch(nilai)
    {
        case 9 : cout << "9 9 9 9 9 9 9 9\n";
        case 8 : cout << " 8 8 8 8 8 8 8 8\n";
        case 7 : cout << "   7 7 7 7 7 7 7\n";
        case 6 : cout << "    6 6 6 6 6 6\n";
        case 5 : cout << "     5 5 5 5\n";
        case 4 : cout << "      4 4 4 4\n";
        case 3 : cout << "       3 3 3\n";
        case 2 : cout << "        2 2\n";
        case 1 : cout << "         1\n";
        case 0 : cout << "0000000000000000\n";
        default : cout << "-----\n";
    }
    system("PAUSE");
    return 0;
}
```

Perhatikan output program di atas. Lihat apa yang terjadi jika tidak ada perintah `break`.

6.2 for Loop

for loop biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah loop yang akan dilakukan atau diketahui maksimum loop yang diizinkan. Dari segi penulisannya, for loop tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Bentuk umum for loop adalah sebagai berikut :

```
for (inisiasi; kondisi; increment)
{
    \\. . . body loop
}
```

Perhatikan bentuk umum for loop diatas. **Inisiasi** merupakan kondisi awal untuk memulai looping. **Kondisi** merupakan syarat untuk melakukan looping. Jika syarat terpenuhi maka looping akan dilakukan, sementara jika tidak, maka looping akan berhenti. **Increment** berguna untuk pengendali looping, yang dapat berupa peningkatan nilai atau penurunan nilai variabel yang digunakan pada **inisiasi**. Setelah loop berjalan, nilai variabel pada **inisiasi** akan bertambah atau berkurang sesuai dengan **increment**. Perubahan ini akan menjadi nilai baru bagi variabel yang diuji lagi apakah masih memenuhi **kondisi**. Jika masih memenuhi, maka proses for loop akan diulangi lagi, jika tidak terpenuhi, maka for loop akan berakhir.

Perhatikan contoh-contoh berikut ini. Pahami setiap struktur program yang diberikan.

1. Ketik dan run dua program berikut ini. Perhatikan penulisan program serta output yang dihasilkan.

```
//latihan621a.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=1;i<11;i++)
        cout << i << endl;
    cout << "Berhenti\n";
    system("PAUSE");
    return 0;
}
```

```
//latihan621b.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=10;i>0;i--)
        cout << i << endl;
    cout << "Berhenti\n";
    system("PAUSE");
    return 0;
}
```

2. Sebelum anda ketik dan run dua program berikut ini, coba anda tebak terlebih dahulu outputnya. Bandingkan hasil yang anda peroleh.

```
//latihan622a.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=0;i<51;i+=5)
        cout << i << endl;
    cout << "Berhenti\n";
    system("PAUSE");
    return 0;
}
```

```
//latihan622b.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=20;i>0;i-=2)
        cout << i << endl;
    cout << "Berhenti\n";
    system("PAUSE");
    return 0;
}
```

3. Tuliskan output program berikut sebelum menetik, mengcompile dan me runnya. Amati dengan seksama proses yang terjadi.

```
//latihan623a.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int i;
for (i=11;i<11;i++)
    cout << i << endl;
cout << "Berhenti\n";
system("PAUSE");
return 0;
}
```

```
//latihan623b.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int i;
for (i=1;i>0;i--)
    cout << i << endl;
cout << "Berhenti\n";
system("PAUSE");
return 0;
}
```

4. Sebagaimana kita ketahui dalam belajar induksi matematika bahwa jumlah n buah bilangan asli pertama adalah $\frac{n(n+1)}{2}$, atau sering disajikan dengan

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

Dua program berikut ini meminta input suatu bilangan bulat dan menghitung jumlah bilangan bulat dari 1 sampai dengan bilangan yang diinputkan.

- (a) Berikan input bilangan 10, 20, 41, 100, 200. Amati nilai yang dihasilkan dan verifikasi nilai yang anda peroleh.

```
//latihan624a.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int i,num;
long int sum;
cout << "Silahkan input bilangan bulat (0-500) : ";
cin >> num;
sum=0;
for (i=1;i<=num;i++)
    sum=sum+i;
cout<<"Jumlah "<<num<<" bilangan bulat pertama = "<<sum<<endl;
system("PAUSE");
return 0;
}
```

- (b) Berikan input bilangan 10, 20, 41, 100, 200 amati nilai yang dihasilkan dan verifikasi nilai yang anda peroleh.

```
//latihan624b.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i,num;
    long int sum;
    cout << "Silahkan input bilangan bulat (0-500) : ";
    cin >> num;
    sum=(num*(num+1))/2;
    cout<<"Jumlah "<<num<<" bilangan bulat pertama = "<<sum<<endl;
    system("PAUSE");
    return 0;
}
```

5. Coba anda tuliskan output program berikut sebelum anda ketik, compile dan run di komputer.

```
//latihan625.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#define LAST 7
using namespace std;
int main()
{
    int count,factorial;
    cout << "Counting\n";
    for (count=1;count<LAST;count++)
        cout << count << endl;
    cout << "Counting by 2\n";
    for (count=0;count<2*LAST;count+=2)
        cout << count<< endl;

    cout << "Calculating factorial\n";
    for (factorial=1,count=1;count<=LAST;count++)
        factorial*=count;
        cout << LAST <<"! adalah "<<factorial<<endl;
    system("PAUSE");
    return 0;
}
```

Agar anda terampil dalam memahami jalannya for loop coba ceritakan proses berjalannya loop tersebut dan sesuaikan dengan output yang anda peroleh. Jika analisa anda benar maka output dari analisa anda akan sama dengan output program sesudah di run.

7 Bagian Ketujuh

7.1 while loop

`while` loop digunakan bila kita mempunyai perulangan yang jumlah maksimum perulangan yang akan dilakukan tidak diketahui, tetapi kita mempunyai kondisi yang disyaratkan dipenuhi. `while` loop banyak digunakan dalam komputasi matematika.

Bentuk umum dari struktur kondisi ini adalah :

```
while (kondisi)
{
    pernyataan-1;
    pernyataan-2

    :
    pernyataan-n
}
```

Cara kerja `while` adalah pertama program akan memeriksa apakah `kondisi` bernilai benar (`true`) atau salah (`false`). Jika benar (`true`), maka blok pernyataan akan dieksekusi. Setelah blok pernyataan selesai dieksekusi, maka program akan kembali memeriksa apakah `kondisi` bernilai benar (`true`) atau salah (`false`), begitu seterusnya sampai `kondisi` bernilai salah (`false`). Hati-hati menggunakan `while` loop ini, karena jika `kondisi` selalu bernilai benar, maka loop tersebut tidak akan pernah berhenti (`infinite loop`). Kita dapat mempertukarkan penyajian `while` loop dan `for` loop mengikuti teknis berikut:

```
for (inisial;kondisi;increment){
    pernyataan
}
```

```
inisial;
while (kondisi){
    pernyataan;
    increment;
}
```

Secara umum jika suatu perulangan diketahui jumlah iterasi yang diperlukan maka `for` loop adalah pilihan yang tepat. Demikian juga jika `inisial`, `kondisi`, `increment` melibatkan variabel yang sama maka pilihan yang bijak adalah `for` loop. Sementara `while` loop digunakan jika kita tidak tahu berapa iterasi yang ingin dilakukan, tetapi kita ingin melakukan iterasi sampai kondisi yang kita inginkan tercapai. Biasanya, kita hanya membatasi banyak iterasi maksimum dalam menggunakan `while` loop ini.

Berikut ini disajikan beberapa contoh penggunaan `while` loop. Anda diharapkan memahami setiap struktur programnya.

1. Output program berikut adalah kata `WHILE LOOPS` sebanyak 5 baris, dimana didepan kata tersebut diberi nomor urut bilangan bulat.

```
//latihan711.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int x;
    x=1;
    while (x<=5)
    {
        cout << x << " WHILE LOOPS\n";
        x++;
    }
    system("PAUSE");
    return 0;
}
```

Output yang anda dapatkan haruslah berbentuk

```
1 WHILE LOOPS
2 WHILE LOOPS
3 WHILE LOOPS
4 WHILE LOOPS
5 WHILE LOOPS
```

Coba amati dengan seksama, bagaimana proses mendapatkan output tersebut baris perbarisnya.

- Perhatikan kembali program `latihan711.cpp`. Coba anda tuliskan program tersebut dengan menggunakan `for loop`.
- Buatlah program dengan menggunakan `while loop` yang outputnya berbentuk sebagai berikut :

```
1    1
2    4
3    9
4   16
5   25
```

- Buatlah program dengan menggunakan `while loop` yang outputnya berupa 10 bilangan ganjil pertama.
- Buatlah program dengan menggunakan `while loop` yang outputnya berupa 10 bilangan genap pertama
- Amati dengan seksama program `latihan716.cpp` yang mempunyai output

```
1
1 2
1 2 3
1 2 3 4
```

Sebelum anda coba mengetik programnya pelajari proses jalannya program tersebut dengan membandingkan dengan output yang ada. Sesuaikan prediksi anda dengan output yang ada?

```
//latihan716.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
int i=1,x;
while (i<=4)
{
    x=1;
    while (x<=i){
        cout << setw(3) << x;
        x=x+1;
    }
    cout << "\n";
    i=i+1;
}
system("PAUSE");
return 0;
}
```

7. Sebagai latihan anda, sebelum menetik dan menjalankan program berikut, coba tebak outputnya terlebih dahulu.

```
//latihan717.cpp
#include <stdio>
#include <stdlib>
#include <iostream>
using namespace std;
int main()
{
int i=2,j=1;
while ((i>-5) || (j-->0)){
    if ((j%2) && (i--%3))
        cout << j;
    else
        cout << i << endl;
}
system("PAUSE");
return 0;
}
```

8. Buatlah program yang outputnya seperti tabel berikut menggunakan for atau while loop :

i	x	$\sin(x)$
1	1.00000	0.841471
2	2.00000	0.909297
3	3.00000	0.141120
4	4.00000	-0.756802
5	5.00000	-0.958924
6	6.00000	-0.279415
7	7.00000	0.656987
8	8.00000	0.989358
9	9.00000	0.412118
10	10.00000	-0.544021

7.2 do-while loop

Versi loop berikutnya adalah **do-while**. Bentuk umumnya adalah

```
do
{
    pernyataan-1;
    pernyataan-2

    :
    pernyataan-n
} while (kondisi);
```

Perbedaannya dengan **while** loop adalah blok pernyataan dieksekusi terlebih dahulu setelah itu baru kondisi diperiksa. Jika kondisi benar, maka proses eksekusi akan diulangi kembali. Oleh karena itu, pada **do-while** loop, blok pernyataan pasti dieksekusi **minimal satu kali**, sementara di **while** loop, mungkin saja blok pernyataan tidak pernah dieksekusi sama sekali.

Perhatikan beberapa contoh berikut ini. Pahami struktur program yang diberikan.

1. Program dibawah ini adalah contoh penggunaan **do-while** loop. Perhatikan output yang dihasilkan.

```
//latihan721.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int x;
    x=1;
    do
    {
        cout << x << " BAHASA C++\n";
        x++;
    }
    while(x<=10);
    system("PAUSE");
    return 0;
}
```

2. Ubah program `latihan721.cpp` diatas dengan menggunakan **for** loop dan **while** loop. Beri nama program anda dengan `latihan722a.cpp` dan `latihan722b.cpp`.
3. Buatlah program yang outputnya seperti tabel berikut menggunakan **do-while** loop.

i	x	$\tan(x)$
1	1.00000	1.557408
2	2.00000	-2.185040
3	3.00000	-0.142547
4	4.00000	1.157821
5	5.00000	-3.380515
6	6.00000	-0.291006
7	7.00000	0.871448
8	8.00000	-6.799711
9	9.00000	-0.452316
10	10.00000	0.648361

8 Bagian Kedelapan : Review

1. Buatlah program yang outputnya adalah kalimat berikut :

```
1. Belajar bahasa pemrograman C++ terkadang mudah/sulit.
2. Dalam bahasa C++, penggunaan huruf kecil sangat penting.
3. int main adalah tempat program eksekusi dimulai.
4. Semua statement harus diakhiri dengan tanda titik koma.
5. Hati-hati memilih tipe variabel yang digunakan!.
6. secara umum double mempunyai panjang bilangan dua kali float.
```

2. Buatlah program dengan output sebagai berikut :

```
Percobaan ...
..1
...2
....3
```

3. Periksalah program berikut: jika anda temukan kesalahan perbaikilah, kemudian ketiklah hasil perbaikan anda dan run program tersebut.

```
#include<cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main[]
{
    int sum:
    sum=34+56:
    cout << "Jumlah 34 dan 56 adalah << sum << endl;
    system("PAUSE");
    return 0;
}
```

4. Tuliskan output program berikut ini.

```
#include<cstdio>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    cout << "Percobaan...";
    cout << "....1";
    cout << "...2";
    cout << "..3";
    cout << "\n";
    system("PAUSE");
    return 0;
}
```

5. Buatlah program pembagian dua bilangan yaitu 13 dibagi dengan 5. Yakinkan hasil program anda benar.

6. Buatlah program dengan menggunakan berbagai format sehingga outputnya sebagai berikut:

```
variabel_int = 100
variabel_float = 331.789978
variabel_double = 8.440000E+011
variabel_char = Z
```

7. Buatlah program yang menghitung hasil operasi berikut dengan benar

$$30000 \times 20000 / 6000 = \dots$$

8. Buatlah program dengan input $a=100$, $b=2$, $c=25$, $d=4$ yang outputnya

```
a - b = 98
b * c = 50
a / c = 4
a + b * c = 150
a * b + c * d = 300
```

9. Diantara nama variabel berikut manakah yang dapat digunakan sebagai nama variabel dibahasa C++, berikan alasan anda?

Int	char	6_05
_calloc	Xx	alpha_beta_routine
floating	_1312	z
doubling	-	A

10. Diantara nilai konstanta berikut manakah yang dapat digunakan sebagai konstanta dibahasa C++, berikan alasan anda?

123.456	0x10.5	0x0G1
0001	0xFFFF	123L
0XAB05	0L	-597.25
123.5e2	.0001	+12.5
0996	-12E-12	07777

11. Tulislah program yang mengkonversi nilai 27 derajat Fahrenheit (F) ke derajat Celcius (C) dengan formula

$$C = (F - 32) / 1.8$$

12. Tulislah program untuk mengevaluasi nilai polinomial $5x^4 + 4x^2 - 7x + 10$ di $x = 2.4$.

13. Tulislah program untuk mengevaluasi nilai ekspresi berikut:

$$\frac{3.31 \times 10^{-8} + 2.10 \times 10^{-7}}{7.16 \times 10^6 + 2.01 \times 10^8}$$

14. Tulislah program yang menerima input bilangan $n = 1 \dots 10$ dan outputnya adalah tabel n faktorial yang di input. Contoh

```
n=4
Tabel faktorial suatu bilangan
1  1
2  2
3  6
4 24
```

15. Buatlah program dengan input jumlah detik dan outputnya dalam jam, menit dan detik. Contoh hasil run program anda adalah :

```
Enter jumlah detik yang anda inginkan: 3920
1 jam, 5 menit, 20 detik
```

16. Buatlah program dengan input suatu bilangan. Jika bilangan yang di input misalnya bilangan bulat bertanda negatif maka program akan mengoutputkan kata "Bilangan yang anda input adalah bilangan bulat negatif", Jika yang diinput bilangan bulat positif maka program akan mengoutputkan kata "Bilangan yang anda input adalah bilangan bulat positif" Jika bilangan yang di input bilangan pecahan desimal bertanda negatif maka program akan mengoutputkan kata "Bilangan yang anda input adalah bilangan pecahan negatif", dan Jika bilangan yang di input bilangan pecahan desimal bertanda positif maka program akan mengoutputkan kata "Bilangan yang anda input adalah bilangan pecahan positif".

9 Bagian Kesembilan

9.1 Array satu dimensi

Array adalah koleksi elemen data yang bertipe sama. Contoh yang tepat di bidang matematika adalah vektor dan matriks. Mungkin anda berfikir kenapa "string" tidak termasuk dalam daftar tipe data di bahasa C++. Hal ini karena "string" adalah array karakter yang berakhir dengan karakter null ("`\0`").

Syntax untuk mendeklarasikan suatu array dalam bahasa C++ adalah

```
tipe nama_variabel[ukuran]
```

Contoh: misalkan kita ingin melakukan komputasi dengan vektor $x = (1.0, 2.0, 3.5)$. Di `int main()` kita deklarasikan dan beri nilai vektor x seperti berikut:

```
double x[3] = {1.0, 2.0, 3.5}; //jika bertipe double
float x[3] = {1.0, 2.0, 3.5}; //jika bertipe float
```

Dalam bahasa C++ indeks array mulai dari nol (*zero offset*). Seperti pada contoh kita ini, ketiga element dari x adalah $x[0]=1.0$, $x[1]=2.0$ dan $x[2]=3.5$. Untuk lebih memahami cara array bekerja perhatikan program `latihan911a.cpp` berikut:

```
//latihan911a.cpp
#include <iomanip>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int i;
    float x[4];
    for (i=0;i<4;i++) {
        cout << "x[" << i << "] = ";
        cin >> x[i];
    }
    cout << "x = [";
    for (i=0;i<4;i++) {
        cout << setw(5) << setprecision(2) << x[i];
    }
    cout << " ]\n";
    system("PAUSE");
    return 0;
}
```

Pada program ini kita deklarasikan vektor x dengan jumlah elemen 4 dengan menggunakan

```
float x[4];
```

tetapi kita tidak memberikan nilai untuk setiap elemen dari vektor x . Pemberian nilainya kita input sesuai dengan permintaan dari keyboard, yang dituliskan sebagai berikut

```
for (i=0;i<4;i++) {
    cout << "x[" << i << "] = ";
    cin >> x[i];
}
```


Output dari program `latihan911a.cpp` untuk input 4.5, 3.5, 2.6 dan 2.5 adalah

```
x[0] = 4.5
x[1] = 3.5
x[2] = 2.6
x[3] = 2.5
x = [ 4.50 3.50 2.60 2.50 ]
```

Baris terakhir pada output diatas diperoleh dari potongan program berikut ini

```
cout << "x = [";
for (i=0;i<4;i++) {
    cout << setw(5) << setprecision(2) << x[i];
}
cout << " ]\n";
```

Pahami dengan baik setiap potongan program tersebut dan output yang dihasilkan. Cara pendefinisian lain dari array pada program `latihan911a.cpp` adalah seperti pada program `latihan911b.cpp` berikut

```
//latihan911b.cpp
#include <iomanip>
#include <cstdlib>
#include <iostream>
#define N 4
using namespace std;
int main()
{
    int i;
    float x[N];
    for (i=0;i<N;i++) {
        cout <<"x[" << i << "] = ";
        cin >> x[i];
    }
    cout << "x = [";
    for (i=0;i<N;i++) {
        cout << setw(5) << setprecision(2) << x[i];
    }
    cout << " ]\n";
    system("PAUSE");
    return 0;
}
```

Pada program `latihan911b.cpp` kita menggunakan variabel global untuk mendefinisikan konstanta untuk ukuran array. Secara umum ukuran array harus didefinisikan dengan konstanta. Cara pendefinisian ukuran array dapat dilakukan dengan dua cara diatas.

9.2 Latihan

1. Modifikasi program `latihan911b.cpp` untuk dapat menerima input vektor $x = [4, 5, 6, 7, 8, 1, 2, 5]$.
2. Buatlah program yang menerima input dari keyboard, kemudian mengeluarkan outputnya seperti berikut :

```
x[0] = 4.56
x[1] = 4.51
x[2] = 4.50
x[3] = 3.10
x[4] = 3.21
x[5] = 2.10
x[6] = 2.11
x = [
    4.56
    4.51
    4.50
    3.10
    3.21
    2.10
    2.11
]
```

3. Buatlah program yang menerima input dari user berupa sebuah vektor

$$x = [4.56 \ 4.51 \ 4.50 \ 3.10 \ 3.21 \ 2.10 \ 2.11]$$

dengan output berbentuk :

```
Input vektor
x[0] = 4.56
x[1] = 4.51
x[2] = 4.50
x[3] = 3.10
x[4] = 3.21
x[5] = 2.10
x[6] = 2.11
Susunan vektor tersebut dibalik adalah
x[6] = 2.11
x[5] = 2.10
x[4] = 3.21
x[3] = 3.10
x[2] = 4.50
x[1] = 4.51
x[0] = 4.56
```

4. Buatlah program yang menerima input dari user berupa vektor $x = [1\ 2\ 3\ 4]$ dan $y = [7\ 8\ 9\ 10]$ dengan output mempertukarkan nilai setiap elemen vektor x dan y sebagai berikut :

```

vektor x awal
x[0] = 1
x[1] = 2
x[2] = 3
x[3] = 4
vektor y awal
y[0] = 7
y[1] = 8
y[2] = 9
y[3] = 10
vektor x akhir
x[0] = 7
x[1] = 8
x[2] = 9
x[3] = 10
vektor y akhir
y[0] = 1
y[1] = 2
y[2] = 3
y[3] = 4

```

5. Perkalian skalar a dengan suatu vektor $v = [v_0\ v_1 \cdots v_n]$ didefinisikan dengan

$$av = [av_0\ av_1 \cdots av_n]$$

Buatlah program untuk menghitung perkalian skalar $a = 5$ dengan vektor $x = [2\ 3\ 4\ 5]$

6. Penjumlahan vektor $v = [v_0\ v_1 \cdots v_n]$ dengan vektor $w = [w_0\ w_1 \cdots w_n]$ didefinisikan sebagai

$$v + w = [v_0 + w_0\ v_1 + w_1 \cdots v_n + w_n]$$

Buatlah program untuk menghitung jumlah dua vektor $x = [2\ 3\ 4]$ dan $y = [3\ 4\ 5]$.

7. Buatlah program untuk menghitung vektor $z = 2x + y$ jika $x = [2\ 3\ 4]$ dan $y = [3\ 4\ 5]$
8. Perkalian titik (dot product) dari vektor $v = [v_0\ v_1 \cdots v_n]$ dengan vektor $w = [w_0\ w_1 \cdots w_n]$ didefinisikan sebagai

$$v \cdot w = \sum_{i=1}^n v_i w_i = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n$$

Buatlah program untuk menghitung perkalian titik dua vektor $x = [2\ 3\ 4]$ dan $y = [3\ 4\ 5]$.

10 Bagian Kesepuluh

10.1 Array Dua Dimensi

Array dua dimensi dalam bahasa pemrograman C++ mempunyai peranan untuk menyajikan suatu matriks dibidang matematika. Oleh karena itu, array dua dimensi memungkinkan kita melakukan berbagai operasi matriks. Sintax untuk mendeklarasikan array dua dimensi dalam bahasa C++ adalah

```
tipe nama_variabel[ukuran baris][ukuran kolom]
```

Contohnya :

- `int A[3][2]` : mendeklarasikan array dua dimensi bertipe integer bernama A berukuran 3×2 .
- `float B[4][2]` : mendeklarasikan array dua dimensi bertipe float bernama B berukuran 4×2 .
- `double C[2][2]` : mendeklarasikan array dua dimensi bertipe double bernama C berukuran 2×2 .

Dengan cara yang sama dapat didefinisikan array bertipe lainnya.

Untuk memberi nilai awal dari array dua dimensi, seperti untuk matrik berukuran 2×2 berikut

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

dapat dilakukan dengan cara sebagai berikut

```
float A[2][2] = {{1.0,2.0},{3.0,4.0}};
```

Karena dalam bahasa C++ indeks array mulai dari nol (*zero offset*), maka nilai elemen-elemen array pada contoh diatas adalah

```
A[0][0]=1.0  A[0][1]=2.0  
A[1][0]=3.0  A[1][1]=4.0
```

Untuk memahami cara array bekerja perhatikan program `latihan101a.cpp` berikut dengan memberi input matriks

$$A = \begin{bmatrix} 2.5 & 3.3 \\ 2.0 & 5.2 \end{bmatrix}$$

```

//latihan101a.cpp
#include <iomanip>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
int i,j;
float A[2][2];
for (i=0;i<2;i++) {
    for (j=0;j<2;j++){
        cout <<"A[" << i << "]"[" << j << "] = ";
        cin >> A[i][j];
    }
}
cout << "A =\n";
for (i=0;i<2;i++) {
    for (j=0;j<2;j++){
        cout << setw(5) << setprecision(2) << A[i][j];
    }
    cout << "\n";
}
cout << "\n";
system("PAUSE");
return 0;
}

```

Program latihan101a.cpp diatas dapat juga disajikan seperti program latihan101b.cpp berikut

```

//latihan101b.cpp
#include <iomanip>
#include <cstdlib>
#include <iostream>
#define N 2
#define M 2
using namespace std;
int main()
{
int i,j;
float A[M][N];
for (i=0;i<M;i++) {
    for (j=0;j<N;j++){
        cout <<"A[" << i << "]"[" << j << "] = ";
        cin >> A[i][j];
    }
}
cout << "A =\n";
for (i=0;i<M;i++) {
    for (j=0;j<N;j++){
        cout << setw(5) << setprecision(2) << A[i][j];
    }
    cout << "\n";
}
cout << "\n";
system("PAUSE");
return 0;
}

```

Pahami setiap struktur program diatas beserta outuput yang dihasilkan.

10.2 Latihan

1. Modifikasi program `latihan101a.cpp` untuk dapat menerima input matriks

$$B = \begin{bmatrix} 5 & 2 & 6 \\ 3 & 4 & 5 \\ 4 & 2 & 3 \end{bmatrix}$$

2. Modifikasi program `latihan101a.cpp` untuk dapat menerima input matriks

$$C = \begin{bmatrix} 5 & 5 & 2 & 6 \\ 3 & 4 & 5 & 5 \\ 4 & 5 & 2 & 3 \end{bmatrix}$$

3. Modifikasi program `latihan101a.cpp` untuk dapat menerima input matriks

$$D = \begin{bmatrix} 5 & 7 & 2 \\ 8 & 8 & 5 \\ 4 & 5 & 2 \\ 0 & 7 & 3 \end{bmatrix}$$

4. Perkalian skalar α dengan matriks A didefinisikan sebagai

$$\alpha A = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \cdots & \alpha a_{1n} \\ \alpha a_{21} & \alpha a_{22} & & \alpha a_{2n} \\ \vdots & & \ddots & \vdots \\ \alpha a_{n1} & \alpha a_{n2} & & \alpha a_{nn} \end{bmatrix}$$

Buatlah program untuk menghitung perkalian skalar $\alpha = 5$ dengan matriks

$$A = \begin{bmatrix} \frac{1}{5} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{3} \end{bmatrix}$$

5. Jumlah dua matriks

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} \quad \text{dan} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & & b_{2n} \\ \vdots & & \ddots & \vdots \\ b_{n1} & b_{n2} & & b_{nn} \end{bmatrix}$$

didefinisikan sebagai

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & & a_{2n} + b_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & & a_{nn} + b_{nn} \end{bmatrix}$$

Buatlah program untuk menghitung jumlah dua matriks dengan input

$$A = \begin{bmatrix} \frac{1}{5} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{3} \end{bmatrix} \quad B = \begin{bmatrix} 5 & 2 & 6 \\ 3 & 4 & 5 \\ 4 & 2 & 3 \end{bmatrix}$$

6. Buatlah program untuk menghitung $C = 2A + B$ dengan

$$A = \begin{bmatrix} 2.5 & 3.3 & 4.2 \\ 2.0 & 5.2 & 3.4 \\ 1.0 & 7.2 & 8.4 \end{bmatrix} \quad B = \begin{bmatrix} 5 & 2 & 6 \\ 3 & 4 & 5 \\ 4 & 2 & 3 \end{bmatrix}$$

7. Modifikasi program `latihan101b.cpp` agar dapat menerima input matriks

$$\text{a) } A = \begin{bmatrix} 5 & 2 & 6 \\ 3 & 4 & 5 \\ 4 & 2 & 3 \end{bmatrix} \quad \text{b) } B = \begin{bmatrix} \frac{1}{5} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

8. Diberikan dua vektor $x = [x_1 \ x_2 \ \dots \ x_n]$ dan $y = [y_1 \ y_2 \ \dots \ y_n]$. Definisikan $x^T y$ dengan

$$x^T y = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \dots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \dots & x_2 y_n \\ \vdots & & \ddots & \vdots \\ x_n y_1 & x_n y_2 & & x_n y_n \end{bmatrix}$$

Buatlah program untuk menghitung nilai $x^T y$ jika $x = [2 \ 3 \ 4]$ dan $y = [3 \ 4 \ 5]$.

9. Misalkan

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix}$$

transpose dari matriks A didefinisikan sebagai

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & & a_{n2} \\ \vdots & & \ddots & \vdots \\ a_{1n} & a_{2n} & & a_{nn} \end{bmatrix}$$

Buatlah program dengan input suatu matriks misalnya

$$A = \begin{bmatrix} 4.2 & 2.1 & 6.7 \\ 3.3 & 4.2 & 5.8 \\ 4.5 & 2.3 & 3.9 \end{bmatrix}$$

dan outputnya adalah matriks A^T .

10. Buatlah program dengan input dua matriks dari keyboard, misalnya

$$A = \begin{bmatrix} 4.2 & 2.1 & 6.7 \\ 3.3 & 4.2 & 5.8 \\ 4.5 & 2.3 & 3.9 \end{bmatrix} \quad \text{dan} \quad B = \begin{bmatrix} 5.2 & 7.1 & 12.7 \\ 6.3 & 4.8 & 3.8 \\ 4.9 & 2.4 & 10.9 \end{bmatrix}$$

dan outputnya adalah mempertukarkan transpose dari kedua matriks tersebut, yaitu $A = B^T$ dan $B = A^T$.

11. Misalkan

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} \quad \text{dan} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & & b_{2n} \\ \vdots & & \ddots & \vdots \\ b_{n1} & b_{n2} & & b_{nn} \end{bmatrix}$$

Perkalian matriks A dan B yang ditulis dengan $C = AB$, dengan elemen ke ij dari C didefinisikan dengan

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}, \quad \text{dengan } i, j = 1, 2, \dots, n.$$

Buatlah program dengan input dua matriks misalnya

$$A = \begin{bmatrix} 4.2 & 2.1 & 6.7 \\ 3.3 & 4.2 & 5.8 \\ 4.5 & 2.3 & 3.9 \end{bmatrix} \quad \text{dan} \quad B = \begin{bmatrix} 5.2 & 7.1 & 12.7 \\ 6.3 & 4.8 & 3.8 \\ 4.9 & 2.4 & 10.9 \end{bmatrix}$$

dan outputnya adalah matriks $C = AB$.

11 Bagian Kesebelas

11.1 Fungsi

Fungsi merupakan blok program yang mempunyai tugas tersendiri. Kegunaan fungsi adalah untuk membagi suatu permasalahan yang kompleks menjadi beberapa bagian sehingga mudah untuk diselesaikan. Tiga bagian penting dari fungsi adalah apa **nama fungsinya**, apa saja nilai yang bisa menjadi **inputnya**, dan apa **outputnya**. Bentuk umum fungsi dalam C++ adalah

```
Tipe_Fungsi Nama_Fungsi(Tipe_Argumen Nama_Argumen)
{
    // Pernyataan
}
```

Perhatikan contoh fungsi berikut ini :

```
int mult ( int x, int y )
{
    int z;
    z=x*y;
    return z;
}
```

Penjelasan mengenai fungsi ini adalah :

1. Tipe Fungsi adalah `int`. Tipe lain yang dapat digunakan adalah `float`, `double`, atau `char`.
2. Nama fungsi adalah `mult`. Pada dasarnya, kita bebas untuk memberi nama fungsi.
3. Ada dua argumen pada fungsi tersebut yang merupakan input fungsi, yaitu `x` dan `y`, masing-masing bertipe `int`. Input ini boleh menggunakan semua tipe data yang sudah kita bahas pada pertemuan-pertemuan sebelumnya.
4. Hasil dari fungsi tersebut adalah variabel `z`, yang merupakan nilai perkalian bilangan `x` dan `y`.
5. `return` merupakan perintah untuk menghasilkan output fungsi.
6. Dalam kalimat, kita dapat mengatakan bahwa : "Fungsi `mult` merupakan fungsi yang menerima input dua bilangan bulat dan outputnya adalah hasil kali dari dua bilangan tersebut".

Contoh berikut ini merupakan pendeklarasian fungsi serta penggunaannya dalam suatu program. Perhatikan dengan seksama dan pahami strukturnya.

```

//latihan111.cpp
#include <iostream>
#include <cstdlib>
using namespace std;

int mult ( int x, int y )
{
int z;
z=x*y;
return z;
}

int main()
{
int x,y;
cout<<"Program ini menghitung hasil kali dua bilangan\n";
cout<<"Masukkan bilangan pertama : ";
cin>> x;
cout<<"Masukkan bilangan kedua : ";
cin>> y;
cout<<"Hasil kali kedua bilangan tersebut adalah "<< mult ( x, y ) <<"\n";
system("PAUSE");
return 0;
}

```

Perhatikan contoh selanjutnya. Berdasarkan penjelasan diatas, coba tebak apa hasil dari program berikut ini sebelum anda run dikomputer.

```

//Latihan112.cpp
#include <iostream>
#include <cstdlib>
using namespace std;

int penjmlhn (int a, int b)
{
int r;
r=a+b;
return r;
}

int main () {
int z;
z = penjmlhn (5,3);
cout << "Hasilnya adalah " << z << endl;
system("PAUSE");
return 0;
}

```

Coba andan modifikasi program Latihan111.cpp dan Latihan112.cpp agar dapat menerima input bilangan bertipe float.

Perhatikan kembali bentuk umum deklarasi fungsi diatas. Tipe fungsi merupakan tipe variabel yang dihasilkan oleh fungsi, dan merupakan output fungsi (return value). Namun adakalanya fungsi tidak mempunyai return value. Fungsi seperti ini disebut fungsi **void**. Selain itu, fungsi juga tidak selalu memerlukan input. Jika input tidak ada, maka bentuk umum fungsi cukup ditulis

Tipe_Fungsi Nama_Fungsi()

Berikut ini merupakan beberapa contoh deklarasi fungsi yang dapat digunakan

```
//fungsi tanpa return value
void Nama_Fungsi(Tipe_Argumen Nama_Argumen)
{
    // Pernyataan
}
```

```
//fungsi tanpa input argumen
Tipe_Fungsi Nama_Fungsi()
{
    // Pernyataan
}
```

```
//fungsi tanpa input dan tanpa return value
void Nama_Fungsi()
{
    // Pernyataan
}
```

Berikut ini adalah contoh fungsi tanpa return value dan tanpa input :

```
//Latihan113.cpp
#include <iostream>
#include<cstdlib>
using namespace std;

void printmessage ()
{
    cout << "Ini adalah fungsi void\n";
}

int main () {
    printmessage ();
    system("PAUSE");
    return 0;
}
```

Program berikut merupakan permainan tebakkan suatu bilangan yang telah ditetapkan besarnya oleh program. Penyajian program ini menggunakan blok fungsi bertipe void. Pahami bagaimana struktur fungsi void tersebut.

```
//Latihan114.cpp
#include <cstdio>
#include <cstdlib>
#include <iostream>
#define target 17
using namespace std;

//definisikan fungsi yang diinginkan disini
//fungsi tersebut kita beri nama test
//fungsi test ini akan memeriksa input bilangan yang diberikan
void test(int tebakkan)
{
    if (tebakkan<target)
        cout << "Tebakan terlalu kecil. \n";
    else if (tebakkan > target)
        cout << "Tebakan terlalu besar. \n";
    else
        cout << "Tebakan anda tepat. \n";
}

int main()
{
    int tebakkan;
    cout << "Saya mengingat suatu bilangan.\n";
    cout << "Cobalah anda tebak bilangan tersebut : ";
    cin >> tebakkan;
    test(tebakkan);
    cout << "Bilangan yang saya ingat adalah " << target << endl;
    system("PAUSE");
    return 0;
}
```

11.2 Latihan

1. Buat program yang menerima input sebuah bilangan r bertipe `float`, kemudian buat fungsi yang menerima input r dengan output luas persegi dengan sisi r .
2. Buat program yang menerima input sebuah bilangan r bertipe `float`, kemudian buat fungsi yang menerima input r dengan output luas lingkaran dengan jari-jari r .
3. Buat program yang menerima input sebuah bilangan r bertipe `float`, kemudian buat fungsi bertipe `void` yang menerima input r kemudian menampilkan luas persegi dengan sisi r .
4. Buat program yang menerima input sebuah bilangan r bertipe `float`, kemudian buat fungsi bertipe `void` yang menerima input r kemudian menampilkan luas lingkaran dengan jari-jari r .
5. Buat program yang menerima input sebuah bilangan $n > 0$ bertipe `int`, kemudian buat fungsi yang menerima input n dengan output jumlah n bilangan asli pertama.

Contoh :

input = 5

output = 15 (karena $1 + 2 + 3 + 4 + 5 = 15$)

6. Buat program yang menerima input sebuah bilangan $n > 0$ bertipe `int`, kemudian buat fungsi yang menerima input n dengan output jumlah kuadrat n bilangan asli pertama.

Contoh :

input = 3

output = 14 (karena $1^2 + 2^2 + 3^2 = 14$)

7. Buat program yang menerima input dua bilangan $m, n > 0$ bertipe `int`, kemudian buat fungsi yang menerima input m, n dengan output luas segi empat dengan panjang m dan lebar n .
8. Buat program yang menerima input dua bilangan $m, n > 0$ bertipe `int`, kemudian buat fungsi bertipe `void` yang menerima input m, n kemudian menampilkan luas segi empat dengan panjang m dan lebar n .
9. Buat program yang memuat fungsi `void` tanpa input argumen yang menampilkan kalimat "Saya adalah pekerja keras dan pantang menyerah" sebanyak 100 baris.
10. Buatlah program untuk menentukan nilai mutlak suatu fungsi $f(x) = \frac{x(x-5)}{2}$ yang diberi input bilangan desimal oleh user yang didefinisikan sebagai berikut:

$$|f(x)| = \begin{cases} f(x), & \text{jika } f(x) \geq 0 \\ -f(x), & \text{jika } f(x) < 0 \end{cases}$$

Coba buat program dengan menggunakan fungsi seperti pada program `Latihan114.cpp`. Definisikan fungsi yang diinginkan sebagai berikut

```
double f(double x)
{
    return (x*(x-5)/2);
}
```

12 Bagian Keduabelas

12.1 Array dalam Statistik

Di Statistik sering kita diberi sekumpulan data. Data ini terkadang perlu diurutkan dari yang kecil ke yang besar atau sebaliknya. Disamping itu kita memerlukan nilai rata-rata data, simpangan baku, variansi dan lain-lain. Disinilah peran Array yang sangat dirasakan dibidang statistik.

Bila kita diberikan sekumpulan data, katakan $x_1, x_2, x_3, \dots, x_n$ maka nilai rata rata aritmatik data tersebut, \bar{x} , diberikan oleh formula

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$$

Selanjutnya bila data yang diberikan berupa nilai $x_1, x_2, x_3, \dots, x_n$ dan frekuensi yang bersesuaian $f_1, f_2, f_3, \dots, f_n$, maka nilai rata rata aritmatik data tersebut, \bar{x} , diberikan oleh formula

$$\bar{x} = \frac{1}{N} \sum_{i=1}^n f_i x_i = \frac{f_1 x_1 + f_2 x_2 + \dots + f_n x_n}{N}, \quad N = \sum_{i=1}^n f_i$$

Bila kita diberikan sekumpu lan data, katakan $x_1, x_2, x_3, \dots, x_n$ dan $\omega_1, \omega_2, \omega_3, \dots, \omega_n$ adalah bobot yang bersesuaian untuk setiap $x_i, i = 1, 2, 3, \dots, n$ maka nilai rata rata aritmatik berbobot data tersebut, \bar{x} , diberikan oleh formula

$$\bar{x} = \frac{\sum_{i=1}^n \omega_i x_i}{\sum_{i=1}^n \omega_i} = \frac{\omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n}{\omega_1 + \omega_2 + \dots + \omega_n}$$

Bila kita diberikan sekumpulan data, katakan $x_1, x_2, x_3, \dots, x_n$ maka nilai rata-rata geometri data tersebut, \bar{x} , diberikan oleh formula

$$\bar{x} = \sqrt[n]{x_1 x_2 x_3 \dots x_n}$$

sementara nilai rata-rata harmonik didefinisikan sebagai :

$$\bar{x} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

Jika data-data $x_1, x_2, x_3, \dots, x_n$ mempunyai frekuensi $f_1, f_2, f_3, \dots, f_n$, maka nilai rata-rata harmonik data-data tersebut adalah

$$\bar{x} = \frac{\sum_{i=1}^n f_i}{\sum_{i=1}^n \frac{f_i}{x_i}} = \frac{f_1 + f_2 + f_3 + \dots + f_n}{\frac{f_1}{x_1} + \frac{f_2}{x_2} + \dots + \frac{f_n}{x_n}}$$

Standar deviasi dari sekumpulan data $x_1, x_2, x_3, \dots, x_n$ adalah

$$s = \sqrt{\frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)}}$$

Jika data tersebut mempunyai frekuensi $f_1, f_2, f_3, \dots, f_n$, maka standar deviasinya adalah

$$s = \sqrt{\frac{N \sum_{i=1}^n f_i x_i^2 - (\sum_{i=1}^n f_i x_i)^2}{N(N-1)}}, \quad N = \sum_{i=1}^n f_i$$

12.2 Latihan

1. Buatlah program untuk menghitung rata-rata aritmatik, geometri, harmonik dan standar deviasi dengan data berikut di input dari keyboard.

(a) $x = [2, 4, 6, 1, 3, 5]$

(b) $x = [10, 8, 6, 4, 2, 12, 13]$

2. Buatlah program untuk menghitung rata-rata aritmatik, geometri, harmonik dan standar deviasi dengan menerima input dari keyboard data berikut.

(a)

No	x	Frekuensi
1.	3	15
2.	5	10
3.	7	25
4.	8	40

(b)

No	x	Frekuensi
1.	4	30
2.	6	10
3.	7	25
4.	8	35

3. Buatlah program untuk menghitung rata-rata aritmatik berbobot dengan menerima input dari keyboard data berikut.

No	x	Frekuensi
1.	20	23/100
2.	25	12/100
3.	37	13/100
4.	27	22/100
5.	43	30/100

13 Bagian Ketigabelas

13.1 Integral Riemann

Pada kalkulus telah dibahas bahwa integral tentu dari suatu fungsi merupakan luas bertanda dari daerah yang dibatasi oleh fungsi tersebut. Nilai integral tentu $\int_a^b f(x) dx$ dapat dihampiri dengan jumlah Riemann kiri, kanan, atau tengah. Misalkan interval $[a, b]$ dibagi menjadi n subinterval dengan lebar $\Delta x = \frac{b-a}{n}$, yaitu

$$a = x_0 < x_1 < x_2 < \dots < x_n = b$$

dimana $x_i = x_0 + i\Delta x$. Hampiran jumlah Riemann kiri, kanan, dan tengah bagi integral tentu $\int_a^b f(x) dx$ adalah

$$L_n = \sum_{i=0}^{n-1} f(x_i) \Delta x \approx \int_a^b f(x) dx$$

$$R_n = \sum_{i=1}^n f(x_i) \Delta x \approx \int_a^b f(x) dx$$

$$M_n = \sum_{i=1}^n f\left(\frac{x_i + x_{i-1}}{2}\right) \Delta x \approx \int_a^b f(x) dx$$

Berikut ini merupakan program untuk menghitung hampiran jumlah Riemann kiri untuk menghampiri

$$\int_0^1 x^2 dx$$

```
#include<iostream>
#include<cstdlib>
using namespace std;
float f(float x)
{
    return x*x;
}
int main()
{
    int i;
    float delta_x,Ln=0;
    delta_x=1.0/4;
    for(i=0;i<4;i++)
    {
        Ln=Ln+f(i*delta_x)*delta_x;
    }
    cout << Ln << endl;
    system("PAUSE");
    return 0;
}
```

```
#include<iostream>
#include<cstdlib>
using namespace std;
float f(float x)
{
    return x*x;
}
int main()
{
    int i;
    float delta_x,Rn=0;
    delta_x=1.0/4;
    for(i=1;i<=4;i++)
    {
        Ln=Ln+f(i*delta_x)*delta_x;
    }
    cout << Ln << endl;
    system("PAUSE");
    return 0;
}
```



```

#include<iostream>
#include<cstdlib>
using namespace std;
float f(float x)
{
    return x*x;
}
int main()
{
    int i;
    float delta_x,Mn=0;
    delta_x=1.0/4;
    for(i=0;i<4;i++)
    {
        Mn=Mn+f((i*delta_x+(i+1)*delta_x)/2)*delta_x;
    }
    cout << Mn << endl;
    system("PAUSE");
    return 0;
}

```

13.2 Latihan

- Modifikasi tiga contoh diatas untuk menghitung jumlah Riemann L_n, R_n, M_n dimana nilai n diinput oleh user. Tampilkan nilai hampiran untuk $i = 1, 2, \dots, n$ dalam bentuk tabel untuk masing-masing metode.
- Untuk setiap fungsi dibawah ini, hampiri luas daerah dibawah kurva tersebut dengan menggunakan tiga metode diatas. Banyak subinterval n diinput oleh user. Tampilkan hasilnya dalam bentuk tabel.
 - $y = \sqrt{x}$ dari $x = 0$ sampai $x = 4$.
 - $y = 4 - 2x$ dari $x = -2$ sampai $x = 2$.
 - $y = 1 + x^2$ dari $x = -1$ sampai $x = 2$.
 - $y = |x|$ dari $x = -1$ sampai $x = 2$.
 - $y = |x - 5|$ dari $x = 0$ sampai $x = 10$.
 - $y = \frac{1}{1 + x^2}$ dari $x = -2$ sampai $x = 2$.
 - $y = \frac{x}{x + 2}$ dari $x = 1$ sampai $x = 4$.
 - $y = \frac{2x}{x^2 + 1}$ dari $x = 1$ sampai $x = 3$.
 - $y = x^2 + \sqrt{1 + 2x}$ dari $x = 4$ sampai $x = 7$.
 - $y = x^4 + 5x^2 + x$ dari $x = 2$ sampai $x = 7$.
 - $y = \frac{1}{x}$ dari $x = 1$ sampai $x = 2$.
 - $y = e^{x^2}$ dari $x = 0$ sampai $x = 4$.
 - $y = \sqrt{1 + x^4}$ dari $x = 0$ sampai $x = 2$.
 - $y = 3x\sqrt{4 + x^2}$ dari $x = 0$ sampai $x = 2$.

Referensi

- [1] Handout Kuliah *Bahasa C untuk Matematikawan* Oleh Dr. M. Imran, M. Sc.
- [2] S. R. Davis. 2004. *C++ FOR DUMMIES*. 5th ed. Wiley Publicing, Inc, USA.

Rencana Kuliah :

Pertemuan 1	:	Kontrak Kuliah
Pertemuan 2	:	
Pertemuan 3	:	
Pertemuan 4	:	
Pertemuan 5	:	
Pertemuan 6	:	
Pertemuan 7	:	
Pertemuan 8	:	UTS
Pertemuan 9	:	
Pertemuan 10	:	
Pertemuan 11	:	
Pertemuan 12	:	
Pertemuan 13	:	
Pertemuan 14	:	
Pertemuan 15	:	
Pertemuan 16	:	UAS