

PRAKTIKUM MINGGU 1
Pengenalan Struktur Program (C++)

1. Struktur Program

Dalam praktikum pemrograman komputer ini akan dipakai bahasa C++ yaitu Dev C++. Adapun struktur umum dari bahasa ini hampir sama dengan bahasa tingkat tinggi yang lain terdiri dari beberapa bagian yaitu :

- Judul Program
- Daftar Header File
- Deklarasi
- Utama

1.1 Judul Program

Judul Program berisi tentang gambaran umum dari program yang biasanya ditulis dalam bentuk baris komentar dan tidak mempengaruhi jalannya atau eksekusi program. Judul program sifatnya opsional.

Contoh :

```
/*Program Luas Lingkaran, untuk menghitung luas lingkaran dengan jari- jari R*/
/*Program Rata2, menghitung nilai rata-rata dari sejumlah nilai*/
//Program Hitung_volume;
```

1.2 Daftar Header File

Bagian ini berisi sejumlah file judul (*header*) yaitu file fungsi baik yang sudah disediakan oleh compiler C++ (*Library function*) maupun fungsi buatan sendiri (*user defined function*).

Contoh :

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

1.3 Deklarasi

Bagian ini berisi deklarasi: konstanta dan variabel global, prototipe fungsi, serta class

Contoh :

```
const int m = 8;
float total;
void hitung_volume(int panjang, int lebar, int tinggi, int &vol);
```

1.3 Utama (*main*)

Bagian Utama merupakan sebuah fungsi utama (*main*) yang merupakan titik awal dan titik akhir program

Contoh :

```
int main
{
    //baris deklarasi
    //baris instruksi
    .....
}
```

2. Tipe Data

Ada beberapa macam tipe data dasar dalam pemrograman yaitu :

Tipe	Jangkauan	Ukuran
char	-128...127	1 byte
int	-32768...32767	2 byte
long	-2.147.483.648...2.147.483.648	4 byte
float	3.4×10^{-38} ... $3.4 \times 10^{+38}$	4 byte
double	1.7×10^{-308} ... $1.7 \times 10^{+308}$	8 byte
long double	3.4×10^{-4932} ... $3.4 \times 10^{+4932}$	10 byte
bool	0, 1	

Tabel 1.1 Tipe data dasar

3. Variabel

Variabel dipakai untuk menyimpan suatu nilai. Nilai yang tersimpan dalam sebuah variabel dapat berubah-ubah. Setiap variabel mempunyai tipe data.

Contoh :

```
int nilai;
float rata2, luas_lingkaran;
```

4. Konstanta

Konstanta hampir sama dengan variabel yaitu dipakai untuk menyimpan suatu nilai namun nilai yang tersimpan dalam konstanta **tidak dapat** berubah.

Contoh :

```
const int R=5;
const float g=9,8;
#define pph=0.15;
```

5. Pernyataan Keluaran (cout)

Pernyataan keluaran dipakai untuk menampilkan karakter alfanumerik di layar. Pernyataan yang biasanya dipakai dalam C++ adalah : cout.

Contoh :

```
#include<iostream.h>
int main()
{
    cout<<"Selamat Belajar C++";
    cout<<"come and enjoy";
    system("pause");
}
```

```

        return 0;
    }

```

Hasil :

Selamat Belajar C++ come and enjoy

6. Pernyataan Masukan (*cin*)

Pernyataan masukan dipakai untuk menerima data masukan atau input dari keyboard. Pernyataan yang biasanya dipakai dalam C++ adalah : `cout`.

Contoh :

1)

```

/*Program hitung luas segi empat*/
#include <iostream.h>
int main()
{
    int p,l,luas;
    cout<<"Masukkan Panjang : ";
    cin>>p;
    cout<<"Masukkan lebar : ";
    cin>>l;
    luas=p*l;
    cout<<"Luas segi empat = "<<luas;
    system("pause");
    return 0;
}

```

2)

```

//Program luas lingkaran;
#include<iostream.h>
#include<math.h>
int main()
{
    const float pi = 3.14;
    float luas,r ;
    cout<<"Masukkan jari-jari : ";
    cin>>r;
    luas = pi*pow(r,2);
    cout<<"Luas lingkaran = "<<luas;
    system("pause");
    return 0;
}

```

3)

```

/*Program Energi Enstein
program untuk menghitung energi yg terkandung dalam benda*/
#include <iostream.h>
int main()
{
    const float c=3e+8;
    float E,m ;
    cout<<"Masukkan massa benda : ";
    cin>>m;
    E=m*pow(c,2);
    cout<<"Energi yang terkandung dalam benda = "<<E;
    system("pause");
    return 0;
}

```

7. Latihan Praktikum :

1. Buatlah program untuk menghitung kecepatan sebuah benda v . Dimana persamaannya adalah :

$$v = s/t$$

2. Buatlah program untuk menghitung momen gaya τ . Dimana persamaannya adalah :

$$\tau = F.d$$

3. Buatlah program untuk menghitung Tekanan P dalam gas ideal. Dimana persamaannya adalah :

$$PV = nRT$$

4. Buatlah program untuk menghitung Gaya F yang dialami oleh sebuah benda. Dimana rumusnya adalah :

$$F = m.a$$

5. Buatlah program untuk menghitung Hambatan total R_s dan R_p dari suatu rangkaian hambatan yang disusun secara seri dan paralel. Dimana persamaannya adalah :

$$\text{Rangkaian Seri : } R_s = R_1 + R_2 + R_3$$

$$\text{Rangkaian Paralel : } \frac{1}{R_p} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

6. Buatlah program untuk menghitung daya listrik P . Dimana persamaannya adalah :

$$P = VI$$

7. Buatlah program untuk menghitung Intensitas dari sebuah sinar I ketika melewati sebuah bahan. Dimana persamaannya adalah :

$$I = I_0 e^{-\mu x}$$

8. Buatlah program untuk menghitung Energi photon. Dimana persamaannya adalah :

$$E = hc / \lambda$$

9. Buatlah program untuk menghitung panjang sisi miring c sebuah segitiga siku-siku. Dimana rumusnya adalah :

$$c^2 = a^2 + b^2$$

10. Buatlah program untuk menghitung akar x_1 dan x_2 dari persamaan kuadrat :

$$Ax^2 + Bx + C$$

dengan memakai rumus ABC :

$$x_1 = \frac{-B + \sqrt{B^2 - 4.A.C}}{2.A}$$

$$x_2 = \frac{-B - \sqrt{B^2 - 4.A.C}}{2.A}$$

PRAKTIKUM MINGGU 2
Operator

Dalam pemrograman komputer ini dikenal 3 jenis operator :

- Operator Aritmatika
- Operator Logika
- Operator Relasi

1. Operator Aritmatika

Yang termasuk operator aritmatika adalah : penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan modulus (%).

Contoh :

```
#include<iostream.h>
int main ()
{
    int a = 3;
    int b = 5;
    int c,d,e,g;
    float f;
    c = a + b ;
    d = a - b ;
    e = c * d;
    f = c / d;
    g = c % a;
    cout<<c<<endl;
    cout<<d<<endl;
    cout<<g<<endl;
    system("pause");
    return 0;
}
```

2. Operator Logika

Yang termasuk operator logika adalah : AND (&&), OR (||), dan NOT (!). Operator logika dipakai untuk membandingkan 2 pernyataan yang mempunyai nilai logika benar (TRUE) atau salah (FALSE). Hasil perbandingan dari operator logika dinyatakan dalam bentuk tabel kebenaran berikut ini :

A	B	A && B
0 (FALSE)	0 (FALSE)	0 (FALSE)
1 (TRUE)	0 (FALSE)	0 (FALSE)
0 (FALSE)	1 (TRUE)	0 (FALSE)
1 (TRUE)	1 (TRUE)	1 (TRUE)

Tabel 2.1 Tabel kebenaran operator AND (&&)

A	B	A B
0 (FALSE)	0 (FALSE)	0 (FALSE)
1 (TRUE)	0 (FALSE)	1 (TRUE)
0 (FALSE)	1 (TRUE)	1 (TRUE)
1 (TRUE)	1 (TRUE)	1 (TRUE)

Tabel 2.2 Tabel kebenaran operator OR (||)

Contoh :

```
#include<iostream.h>
int main()
{
    int nilai=7;
    int absen = 40;
    bool x,y,z,q;
    x=nilai>5;
    y=absen>75;
    z=x&& y;
    q=x||y;
    cout<<z<<endl;
    cout<<q<<endl;
    system("pause");
    return 0;
}
```

3. Operator Relasi

Yang termasuk operator Relasi adalah : lebih besar (>), lebih kecil (<), lebih besar sama dengan (>=), lebih kecil sama dengan (<=), sama dengan (==), dan tidak sama dengan (!=). Operator logika dipakai untuk membandingkan 2 buah bilangan dan hasil perbandingannya bisa mempunyai nilai logika benar (TRUE) atau salah (FALSE).

Contoh :

```
#include<iostream.h>
int main()
{
    int nilai=90;
    bool x,y,z;
    x=nilai>=80;
    y=nilai<=100;
    cout<<x<<endl;
    cout<<y<<endl;
    z=x&&y;
    cout<<z<<endl;
    system("pause");
    return 0;
}
```

PRAKTIKUM MINGGU 3

Pernyataan Pengambilan Keputusan

Dalam pemrograman komputer terdapat pernyataan yang dipakai untuk pengambilan keputusan dari sebuah kondisi yaitu : pernyataan *if*, pernyataan *if-else*, dan pernyataan *switch-case*.

1. Pernyataan *if*

Adapun bentuk umum atau syntax dari pernyataan ini dalam C++ adalah :

```
if (kondisi)
{
    aksi_1
    aksi_2
    ...
    aksi_n
}
```

Pernyataan *aksi_1*, *aksi_2*,...,*aksi_n* akan dikerjakan jika *kondisi* bernilai TRUE.

Contoh :

```
#include <iostream.h>
int main ()
{
    int nilai_ujian,absen;
    cout << "Masukkan nilai ujian = ";
    cin >> nilai_ujian;
    cout << "Masukkan prosentase absen = ";
    cin >> absen;
    if (nilai_ujian >= 5 && absen>=75)
    {
        cout << " Anda Lulus ";
        cout <<endl;
        cout << " Selamat ";
    }
    if (nilai_ujian < 5 && absen<75)
    {
        cout << " Anda Tidak Lulus ";
        cout <<endl;
        cout << " Belajar Lagi ";
    }
    if (nilai_ujian >= 5 && absen<75)
    {
        cout << " Anda Tidak Lulus ";
        cout <<endl;
        cout << " Belajar Lagi ";
    }
    if (nilai_ujian < 5 && absen>=75)
    {
        cout << " Anda Tidak Lulus ";
        cout <<endl;
        cout << " Belajar Lagi ";
    }
    system("pause");
    return 0;
}
```

2. Pernyataan *if-else*

Adapun bentuk umum atau syntax dari pernyataan ini dalam C++ adalah :

```

if (kondisi)
{
    aksi_1
    aksi_2
    ...
    aksi_n
}
else
{
    aksi_a
    aksi_b
    ...
    aksi_z
}
    
```

Jika *kondisi* bernilai TRUE maka yang dikerjakan adalah aksi_1,aksi_2,...aksi_n sedangkan jika kondisi bernilai FALSE maka yang dikerjakan adalah aksi_a, aksi_b, ...aksi_z.

Contoh :

```

#include <iostream.h>
int main ()
{
    int nilai_ujian,absen;
    cout << "Masukkan nilai ujian = ";
    cin >> nilai_ujian;
    cout << "Masukkan prosentase absen = ";
    cin >> absen;
    if (nilai_ujian >= 5 && absen>=75)
    {
        cout << " Anda Lulus ";
        cout <<endl;
        cout << " Selamat ";
    }
    else
    {
        cout << " Anda Tidak Lulus ";
        cout <<endl;
        cout << " Belajar Lagi ";
    }
    system("pause");
    return 0;
}
    
```

3. Pernyataan *switch-case*

Adapun bentuk umum atau syntax dari pernyataan ini dalam C++ adalah :

```

switch (pilihan)
{
    case(pilihan_1) :
    {
        Aksi_1
    }
    case(pilihan_2) :
    {
    
```



```

        Aksi_2
    }

    case(pilihan_n) :
    {
        Aksi_n
    }
    Default :
    {
        Aksi_z
    }
}

```

Jika *pilihan* bernilai pilihan_1 maka yang dikerjakan adalah aksi_1, Jika *pilihan* bernilai pilihan_2 maka yang dikerjakan adalah aksi_2, Jika *pilihan* bernilai pilihan_n maka yang dikerjakan adalah aksi_n. Tapi jika *pilihan* bernilai diluar pilihan_1,pilihan_2,...pilihan_n maka yang dikerjakan adalah aksi_z.

Contoh :

1)

```

#include <iostream.h>
int main()
{
    int no_channel;
    cout << "masukkan nomer channel TV (1 - 4) = ";
    cin >> no_channel;
    switch (no_channel)
    {
        case (1) :
        {
            cout << "TVRI";
            break;
        }
        case (2) :
        {
            cout << "RCTI";
            break;
        }
        case (3) :
        {
            cout << "SCTV";
            break;
        }
        case (4) :
        {
            cout << "ANTV";
            break;
        }
        default :
        {
            cout << "Maaf, anda salah channel ! ";
            break;
        }
    }
    system("pause");
    return 0;
}

```

2)

```
#include<iostream.h>
#include<string.h>
using namespace std;
int main()
{
    string nama, ket;
    char kode;
    cout<<"Masukkan nama mahasiswa: ";
    getline(cin,nama);
    cout<<"Pilih kode jurusan [A/B/C/D] : ";
    cin>>kode;
    switch (toupper(kode))
    {
        case 'A' :
        {
            ket = "Jurusan Fisika";
            break;
        }
        case 'B' :
        {
            ket ="Jurusan Kimia";
            break;
        }
        case 'C' :
        {
            ket="Jurusan Biologi";
            break;
        }
        case 'D' :
        {
            ket="Jurusan Matematika";
            break;
        }
        default :
        {
            ket="Salah Jurusan";
            break;
        }
    }

    cout<<"Nama mahasiswa : "<<nama<<endl;
    cout<<"Kode jurusan : "<<kode<<endl;
    cout<<"Nama jurusan : "<<ket<<endl;
    system("pause");
    return 0;
}
```

4. Latihan Praktikum

1. Buatlah program untuk menghitung hambatan total dari 2 buah hambatan yaitu R1 dan R2. Program memiliki pilihan untuk menentukan apakah hambatan-hambatan akan disusun secara seri atau paralel. Gunakan pernyataan *if* atau *if-else*.
2. Buatlah program untuk mengoperasikan 2 buah bilangan yaitu bil1 dan bil2. Kita akan mengoperasikan kedua bilangan tersebut dengan pilihan sebagai berikut :
 1. Penjumlahan
 2. Pengurangan

3. Perkalian

4. Pembagian

misal jika kita pilih 1 maka program akan menjumlahkan bil1 dan bil2. Dan jika kita pilih 3 maka program komputer akan melakukan perkalian antara bil1 dan bil2. Dalam hal ini gunakanlah pernyataan *switch*.

3. Buatlah program yang dapat memilih untuk menghitung diameter, luas atau keliling dari sebuah lingkaran dengan jari-jari R. Kita akan menghitung besaran-besaran tersebut dengan menu pilihan sebagai berikut :

1. Diameter
2. Keliling
3. Luas

Dalam hal ini gunakanlah pernyataan *switch-case*.

4. Buatlah program komputer yang dapat mengubah nilai angka menjadi nilai huruf dengan ketentuan sbb :

Nilai angka	Nilai huruf
80<=nilai angka<=100	A
65<=nilai angka<80	B
50<=nilai angka<65	C
35<=nilai angka<50	D
0<=nilai angka<35	E

Gunakan pernyataan *if-else*

5. Buatlah program komputer yang dapat menghitung fungsi f(x) berikut ini :

$$f(x) = \begin{cases} 0 & ;x \leq 5 \\ (x-5)/3 & ;5 \leq x \leq 8 \\ 1 & ;x \geq 8 \end{cases}$$

Gunakan pernyataan *if-else*

6. PT. Citra menggaji karyawannya berdasarkan golongannya. Upah per jam tiap karyawannya ditentukan seperti berikut

Golongan	Upah/jam
A	Rp. 7000,-
B	Rp. 8000,-
C	Rp. 9000,-
D	Rp. 10.000,-

Jumlah jam kerja normal dalam 1 minggu adalah 48 jam. Jika terdapat kelebihan jam kerja dianggap lembur dengan upah lembur adalah Rp.5000,- per jam untuk semua golongan karyawan. Buatlah program untuk menghitung gaji karyawan mingguan. Data yang dimasukan(cin) adalah nama karyawan, golongan, jumlah jam kerja. Sedangkan Data yang dicetak (cout) adalah nama karyawan dan gajinya. Gunakan pernyataan kombinasi antara *if* dan *switch-case*

PRAKTIKUM MINGGU 4

Pernyataan Pengulangan (*looping*)

Dalam pemrograman komputer ada beberapa pernyataan yang biasanya dipakai untuk menangani suatu kejadian yang terjadi secara berulang yaitu : pernyataan *for*, pernyataan *while*, dan pernyataan *do-while*.

1. Pernyataan *for*

Bentuk umum atau *syntax* dari pernyataan *for* dalam C++ adalah :

```
for (ekspresi_awal;kondisi;ekspresi_pencacah)
{
    aksi_1
    aksi_2
    ...
    aksi_n
}
```

Selama *kondisi* bernilai TRUE maka *aksi_1*, *aksi_2*,...,*aksi_n* akan dikerjakan secara berulang-ulang dimulai dari *ekspresi_awal*

Contoh :

```
#include <iostream.h>
int main()
{
    int i;
    for (i=1; i <= 5 ; i++)
    {
        cout << i;
        cout <<endl;
    }
    system("pause");
    return 0;
}
```

2. Pernyataan *while*

Bentuk umum atau *syntax* dari pernyataan *while* dalam C++ adalah :

```
While (kondisi)
{
    aksi_1
    aksi_2
    ...
    aksi_n
}
```

Selama *kondisi* bernilai TRUE maka *aksi_1*,*aksi_2*,...,*aksi_n* akan dikerjakan secara berulang

Contoh :

```
#include <iostream.h>
int main()
{
    int i;
    i=1;
```

```

while (i <= 5)
{
    cout << i;
    cout <<endl;
    i++;
}
system("pause");
return 0;
}

```

3. Pernyataan *do-while*

Adapun bentuk umum atau syntax dari pernyataan ini dalam C++ adalah :

```

do
{
    aksi_1
    aksi_2
    ...
    aksi_n
}
while (kondisi);

```

Selama *kondisi* bernilai TRUE maka aksi_1, aksi_2,...aksi_n akan dikerjakan secara berulang-ulang

Contoh :

```

#include <iostream.h>
int main()
{
    int i =1;
    do
    {
        cout <<"Save our planet, keep violence away!"<<endl;
        i++ ;
    }
    while (i <=5);
    system("pause");
    return 0;
}

```

4. Latihan Praktikum

1. Buatlah program untuk menghitung hambatan total dari n buah hambatan yaitu R1,R2,...Rn. Program memiliki pilihan untuk menentukan apakah hambatan-hambatan akan disusun secara seri atau paralel.
2. Buatlah program komputer yang dapat memasukkan n buah nilai dan mencari nilai rata-ratanya. Selanjutnya mengubah nilai rata-rata yang berupa nilai angka tersebut menjadi nilai huruf dengan ketentuan sbb :

Nilai angka	Nilai huruf
80<=nilai angka<=100	A
65<=nilai angka<80	B
50<=nilai angka<65	C

35<=nilai angka<50	D
0<=nilai angka<35	E

- Buatlah program komputer yang dapat mencetak bilangan ganjil dan genap antara 0 sampai n.
- Buatlah program komputer yang dapat memasukkan n buah bilangan dan menghitung hasil perkalian dari n buah bilangan tersebut.
- Buatlah program komputer yang dapat menghitung nilai faktorial dari sebuah bilangan.
- Buatlah program komputer yang dapat menghitung fungsi y berikut ini :

$$y = \sum_{i=1}^n y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Contoh untuk n=3 :

$$y = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

- Buatlah program komputer yang dapat menghitung fungsi y berikut ini :

$$y = mx + c$$

dimana :

$$m = \frac{N \sum_{n=1}^N x_n y_n - \left(\sum_{n=1}^N x_n \right) \left(\sum_{n=1}^N y_n \right)}{N \sum_{n=1}^N x_n^2 - \left(\sum_{n=1}^N x_n \right)^2}$$

$$c = \frac{\sum_{n=1}^N y_n}{N} - m \frac{\sum_{n=1}^N x_n}{N}$$

Contoh data untuk N=7

x_n	y_n
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5

PRAKTIKUM MINGGU 5

Larik (*Array*)

Array merupakan sekumpulan variabel yang memiliki nama dan tipe data yang sama. Ada dua jenis array yaitu : 1 dimensi dan multi dimensi.

1. Array 1 Dimensi

Cara mendeklarasikan variabel array 1 dimensi adalah :

```
tipe_data nama_array[indeks];
```

contoh :

```
int data[5];
```

mendeklarasikan *array* dengan nama *data* dan tipe data integer. *Array* data terdiri dari 5 elemen yaitu : data[0], data[1], data[2], data[3], dan data[4].

Contoh :

- 1)

```
#include <iostream.h>
int main()
{
    int bil[5];
    int jumlah;
    cout << "masukkan bilangan ke-0 =";
    cin >>bil[0];
    cout <<"masukkan bilangan ke-1=";
    cin>>bil[1];
    cout << "masukkan bilangan ke-2 =";
    cin >>bil[2];
    cout <<"masukkan bilangan ke-3=";
    cin>>bil[3];
    cout << "masukkan bilangan ke-4 =";
    cin >>bil[4];
    jumlah = bil[0]+bil[1]+bil[2]+bil[3]+bil[4];
    cout <<"Hasil penjumlahan adalah = " <<jumlah;
    system("pause");
    return 0;
}
```
- 2)

```
#include <iostream.h>
int main()
{
    int bil[5];
    int jumlah,i;
    jumlah=0;
    for (i=0;i<5;i++)
    {
        cout<<"Masukkan bilangan ke-"<<i<<" = ";
        cin>>bil[i];
        jumlah=jumlah+bil[i];
    }
    cout <<"Hasil penjumlahan adalah = " <<jumlah<<endl;
    system("pause");
    return 0;
}
```


2. Array Multi Dimensi

Cara mendeklarasikan variabel array multi dimensi adalah :

tipe_data nama_array[indeks_1][indeks_2];

contoh :

```
int data[2][3];
```

mendeklarasikan *array* 2 dimensi dengan nama *data* dan tipe data integer. *Array* data terdiri dari 6 elemen yaitu : data[0][0], data[0][1], data[0][2], data[1][0], data[1][1], dan data[1][2].

Contoh :

```
#include <iostream.h>
int main()
{
int A[2][3];
int i,j;
//memasukkan elemen array
for (i=0;i<2;i++)
for (j=0;j<3;j++)
{
cout<<"Masukkan A["<<i<<j<<" = ";
cin>>A[i][j];
}
//mencetak elemen array
for (i=0;i<2;i++)
for (j=0;j<3;j++)
{
cout<<"A["<<i<<j<<" = "<<A[i][j]<<endl;
}
system("pause");
return 0;
}
```

3. Latihan Praktikum

1. Buatlah program untuk menghitung hambatan total dari n buah hambatan yaitu R1,R2,...Rn. Program memiliki pilihan untuk menentukan apakah hambatan-hambatan akan disusun secara seri atau paralel. Nyatakan hambatan R1, R2,...Rn dengan *array* R yaitu : R[1],R[2],...R[n]
2. Buatlah program komputer yang dapat memasukkan n buah nilai dan mencari nilai rata-ratanya. Selanjutnya mengubah nilai rata-rata yang berupa nilai angka tersebut menjadi nilai huruf dengan ketentuan sbb :

Nilai angka	Nilai huruf
80<=nilai angka<=100	A
65<=nilai angka<80	B
50<=nilai angka<65	C

$35 \leq \text{nilai angka} < 50$	D
$0 \leq \text{nilai angka} < 35$	E

Nyatakan n buah nilai dengan array nilai yaitu : nilai[1], nilai[2], ... nilai[n]

- Buatlah program komputer yang dapat mencetak bilangan ganjil dan genap antara 0 sampai n. Nyatakan bilangan ganjil dan genap tersebut dalam array genap[n] dan ganjil[n]
- Buatlah program komputer yang dapat memasukkan n buah bilangan dan menghitung hasil perkalian dari n buah bilangan tersebut. Nyatakan n buah bilangan tersebut dengan array yaitu : bil[n].
- Buatlah program komputer yang dapat menghitung fungsi y berikut ini :

$$y = \sum_{i=1}^n y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Contoh untuk n=3 :

$$y = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Nyatakan data x dan y dalam bentuk array yaitu : x[i] dan y[i]

- Buatlah program komputer yang dapat menghitung fungsi y berikut ini :

$$y = mx + c$$

dimana :

$$m = \frac{N \sum_{n=1}^N x_n y_n - \left(\sum_{n=1}^N x_n \right) \left(\sum_{n=1}^N y_n \right)}{N \sum_{n=1}^N x_n^2 - \left(\sum_{n=1}^N x_n \right)^2}$$

$$c = \frac{\sum_{n=1}^N y_n}{N} - m \frac{\sum_{n=1}^N x_n}{N}$$

Contoh data untuk N=7

x_i	y_i
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5

Nyatakan data x dan y dalam bentuk array : x[i] dan y[i]

7. Buatlah program komputer untuk menghitung hasil penjumlahan antara 2 buah matrik $A[2][2]$ dan $B[2][2]$. Hasil penjumlahan dari kedua matrik tersebut disimpan dalam matrik $C[2][2]$
8. Buatlah program komputer untuk menghitung hasil perkalian antara 2 buah matrik $A[3][3]$ dan $B[3][3]$. Hasil penjumlahan dari kedua matrik tersebut disimpan dalam matrik $C[3][3]$

PRAKTIKUM MINGGU 6

Fungsi (*Function*)

Function adalah sebuah teknik dalam pemrograman dimana program yang besar atau panjang akan dipecah menjadi beberapa sub program yang kecil. Dimana masing-masing sub program akan bertugas untuk menjalankan fungsi atau tugas tertentu.

Dalam program maka fungsi terdiri dari dua bagian yaitu : prototipe fungsi dan definisi fungsi. Bentuk umum atau *sintax* dari function dalam C++ adalah :

```
void nama_fungsi(parameter) ; -Prototipe fungsi

int main()                    -Fungsi utama (main)
{
    nama_fungsi(argumen);    -pemanggilan fungsi
}
void nama_fungsi(parameter) -Definisi fungsi
{
    //pernyataan fungsi
}
```

Contoh :

```
#include <iostream.h>
void cetak_tulisan(); //prototipe fungsi
int main()
{
    cetak_tulisan(); //panggil fungsi cetak_tulisan tanpa argumen
    system("pause");
    return 0;
}
void cetak_tulisan() //definisi fungsi
{
    for (int i=1; i <=5; i++)
    {
        cout <<"Manusia yang besar adalah manusia yang tidak pernah mengeluh";
        cout <<endl;
    }
}
```

Dalam pemrograman dengan fungsi maka ada 3 cara pemanggilan fungsi dalam fungsi utama yaitu : pemanggilan secara nilai (*call by value*), pemanggilan secara alamat (*call by address*), dan pemanggilan secara referensi (*call by reference*).

1. Pemanggilan secara nilai (*call by value*)

Dalam Pemanggilan fungsi secara nilai maka data yang dikirimkan lewat argumen akan diterima oleh parameter dalam fungsi. Data yang diterima oleh parameter adalah berupa nilai langsung atau isi dari argumen. Selanjutnya data tersebut akan diproses

dalam fungsi. Ketika fungsi selesai dikerjakan dan kembali ke fungsi utama maka data-data tersebut **tidak** akan mengalami perubahan.

Contoh :

```
#include <iostream.h>
void cetak_variabel(int x,char y); //prototype fungsi
int main()
{
    int x;
    char y;
    x = 5;
    y = 'A';
    cetak_variabel(x,y); //panggil fungsi cetak_variabel dgn argumen x,y
    cout<<"nilai x dalam fungsi utama "<<x<<endl;
    cout<<"nilai y dalam fungsi utama "<<y<<endl;
    system("pause");
    return 0;
}

void cetak_variabel(int x, char y) //definisi fungsi
{
    x=x+5;
    y='B';
    cout <<"nilai x = "<< x <<endl;
    cout <<"nilai y = " << y <<endl;
    return;
}
```

2. Pemanggilan secara alamat (*call by address*)

Dalam Pemanggilan fungsi secara alamat maka data yang dikirimkan lewat argumen akan diterima oleh parameter dalam fungsi. Data yang diterima oleh parameter adalah berupa alamat dari argumen. Selanjutnya data tersebut akan diproses dalam fungsi. Ketika fungsi selesai dikerjakan dan kembali ke fungsi utama maka data-data tersebut akan mengalami perubahan.

Contoh :

```
#include <iostream.h>
void cetak_variabel(int *x,int *y); //prototype fungsi
int main()
{
    int x,y;
    x = 5;
    y = 10;
    cetak_variabel(&x,&y); //panggil fungsi
    cout<<"nilai x dalam fungsi utama "<<x<<endl;
    cout<<"nilai y dalam fungsi utama "<<y<<endl;
    system("pause");
    return 0;
}

void cetak_variabel(int *x, int *y) //definisi fungsi
{
    *x=*x+5;
    *y=*y+5;
    cout <<"nilai x = "<< *x <<endl;
}
```

```

    cout <<"nilai y = " << *y <<endl;
    return;
}

```

3. Pemanggilan secara referensi (*call by reference*)

Dalam Pemanggilan fungsi secara referensi maka data yang dikirimkan lewat argumen akan diterima oleh parameter dalam fungsi. Data yang diterima oleh parameter adalah berupa alamat dari argumen. Selanjutnya data tersebut akan diproses dalam fungsi. Ketika fungsi selesai dikerjakan dan kembali ke fungsi utama maka data-data tersebut akan mengalami perubahan.

Contoh :

```

#include <iostream.h>
void cetak_variabel(int &x,int &y); //prototype fungsi
int main()
{
    int x,y;
    x = 5;
    y = 10;
    cetak_variabel(x,y);      //panggil fungsi
    cout<<"nilai x dalam fungsi utama "<<x<<endl;
    cout<<"nilai y dalam fungsi utama "<<y<<endl;
    system("pause");
    return 0;
}

void cetak_variabel(int &x, int &y)      //definisi fungsi
{
    x=x+5;
    y=y+5;
    cout <<"nilai x = " << x <<endl;
    cout <<"nilai y = " << y <<endl;
    return;
}

```

4. Fungsi *Inline*

Fungsi *Inline* merupakan sebuah fungsi yang dipakai untuk mengembalikan nilai lewat pernyataan return. Fungsi ini ditandai dengan menyebutkan tipe data di depan nama fungsi.

Contoh :

```

#include <iostream.h>
int luas_segi4(int panjang, int lebar);
int main()
{
    int panjang,lebar,luas;
    cout << "masukkan panjang = ";
    cin >> panjang;
    cout << "masukkan lebar = ";
    cin >>lebar;
    luas= luas_segi4(panjang,lebar);
    cout<<"Luas segi 4 = "<<luas<<endl;
    system("pause");
    return 0;
}

```

```

}

int luas_segi4(int panjang,int lebar)
{
    int L;
    L=panjang*lebar;
    return L;
}

```

5. Memecah Program dengan Fungsi

Fungsi berguna untuk memecah program yang besar menjadi beberapa sub program yang kecil. Masing-masing sub program akan bertugas untuk mengerjakan tugas atau fungsi tertentu. Misalkan kita mempunyai program yang belum dibagi menjadi sub program seperti berikut ini :

```

#include <iostream.h>
int main()
{
    int panjang,lebar, tinggi, vol;
    cout <<"masukkan panjang = ";
    cin >>panjang;
    cout <<"masukkan lebar = ";
    cin >>lebar;
    cout <<"masukkan tinggi = ";
    cin >>tinggi;
    vol = panjang*lebar*tinggi;
    cout << " Volume dari kotak adalah = " <<vol;
    cout <<endl;
    system("pause");
    return 0;
}

```

Jika program tersebut kita pecah menjadi 3 sub program atau fungsi akan menjadi :

- input_data
- hitung_vol
- cetak_vol

Maka program tersebut akan menjadi :

```

#include <iostream.h>
void input_data(int &panjang, int &lebar, int &tinggi);
void hitung_vol(int panjang, int lebar, int tinggi, int &vol);
void cetak_vol(int vol);
int main()
{
    int panjang,lebar,tinggi,vol;
    input_data(panjang,lebar,tinggi);
    hitung_vol(panjang,lebar,tinggi,vol);
    cetak_vol(vol);
    system("pause");
    return 0;
}

void input_data(int &panjang, int &lebar, int &tinggi)
{

```

```

cout <<"masukkan panjang = ";
cin >>panjang;
cout <<"masukkan lebar = ";
cin >>lebar;
cout <<"masukkan tinggi = ";
cin >>tinggi;
}

void hitung_vol(int panjang, int lebar, int tinggi, int &vol)
{
    vol = panjang*lebar*tinggi;
    return;
}

void cetak_vol(int vol)
{
    cout <<" volume kotak adalah = "<<vol;
    return;
}

```

6. Latihan Praktikum

1. Buatlah program untuk menghitung hambatan total dari n buah hambatan yaitu R1,R2,...Rn. Program memiliki pilihan untuk menentukan apakah hambatan-hambatan akan disusun secara seri atau paralel. Bagilah program tersebut menjadi 3 fungsi yaitu : input_data, hitung_R, dan cetak_R
2. Buatlah program komputer yang dapat memasukkan n buah nilai dan mencari nilai rata-ratanya. Selanjutnya mengubah nilai rata-rata yang berupa nilai angka tersebut menjadi nilai huruf dengan ketentuan sbb :

Nilai angka	Nilai huruf
80<=nilai angka<=100	A
65<=nilai angka<80	B
50<=nilai angka<65	C
35<=nilai angka<50	D
0<=nilai angka<35	E

Bagilah program tersebut menjadi 3 sub program atau fungsi yaitu : input_data, hitung_konversi, dan cetak_hasil

3. Buatlah program komputer yang dapat memasukkan n buah bilangan dan menghitung hasil perkalian dari n buah bilangan tersebut. Nyatakan n buah bilangan tersebut dengan array yaitu : bil[n]. Bagilah program tersebut menjadi 3 fungsi yaitu : input_data, hitung_perkalian, dan cetak_hasil.
4. Buatlah program komputer yang dapat menukarkan isi dari dua buah variabel seperti contoh berikut ini :

A = 5 B = 10 setelah ditukar menjadi A = 10 B = 5

Bagilah program tersebut menjadi 3 fungsi yaitu : input, tukar, dan cetak_hasil.

5. Buatlah program komputer yang dapat menghitung fungsi y berikut ini :

$$y = \sum_{i=1}^n y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Contoh untuk n=3 :

$$y = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Nyatakan data x dan y dalam bentuk array yaitu : x[i] dan y[i]. Bagilah program tersebut menjadi 3 fungsi yaitu : input_data, hitung_fungsi, dan cetak_fungsi

6. Buatlah program komputer yang dapat menghitung fungsi y berikut ini :

$$y = mx + c$$

dimana :

$$m = \frac{N \sum_{n=1}^N x_n y_n - \left(\sum_{n=1}^N x_n \right) \left(\sum_{n=1}^N y_n \right)}{N \sum_{n=1}^N x_n^2 - \left(\sum_{n=1}^N x_n \right)^2}$$

$$c = \frac{\sum_{n=1}^N y_n}{N} - m \frac{\sum_{n=1}^N x_n}{N}$$

Contoh data untuk N=7

x_i	y_i
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5

Nyatakan data x dan y dalam bentuk array : x[i] dan y[i]. Bagilah program tersebut menjadi 3 fungsi yaitu : input_data, hitung_fungsi, dan cetak_fungsi

7. Buatlah program komputer untuk menghitung hasil penjumlahan antara 2 buah matrik A[2][2] dan B[2][2]. Hasil penjumlahan dari kedua matrik tersebut disimpan dalam matrik C[2][2]. Bagilah program tersebut menjadi 3 fungsi yaitu : input_matrik, jumlah_matrik, dan cetak_jumlah
8. Buatlah program komputer untuk menghitung hasil perkalian antara 2 buah matrik A[3][3] dan B[3][3]. Hasil penjumlahan dari kedua matrik tersebut disimpan dalam

matrik C[3][3]. Bagilah program tersebut menjadi 3 fungsi yaitu : input_matrik, kali_matrik, dan cetak_kali