

MODUL 5 : MODULARITAS / FUNGSI

LAB 5.1: TUJUAN PRAKTIKUM

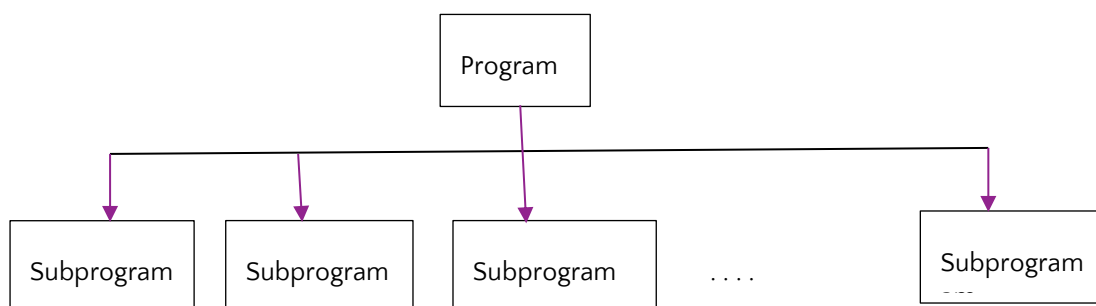
Learning outcome untuk praktikum pemrograman dasar pada modul 5 ini adalah :

LO3 Mahasiswa mampu menerapkan dan menganalisis konsep modularitas dan komunikasi data antar modul dan/atau fungsi dalam C++

LAB 5.2: DASAR TEORI

Konsep modularitas dalam pemrograman dikenal sebagai sebuah metode untuk mempermudah penyusunan dan koreksi terhadap suatu program, khususnya program yang besar dan kompleks. Pendekatan modularitas akan memisahkan atau memecah menjadi bagian-bagian tertentu dalam suatu program yang disebut sebagai komponen program atau subprogram/subrutin/modul, kemudian akan memanggil komponen-komponen program tersebut dalam satu program utama (main program). Salah satu implementasi dari konsep modularitas ini adalah penggunaan Fungsi. Fungsi banyak digunakan dalam bahasa C++ dengan dua alasan utama :

1. Fungsi menjadikan program yang ditulis mempunyai struktur yang jelas. Dengan memisahkan langkah-langkah detail ke satu atau lebih fungsi-fungsi, maka fungsi utama (fungsi main()) akan menjadi lebih pendek, jelas dan mudah dimengerti. Hal seperti ini menunjukkan suatu struktur program yang baik.
2. Fungsi-fungsi dapat digunakan untuk menghindari penulisan yang berulang-ulang. Langkah-langkah program yang sama dan sering digunakan berulang-ulang di program dapat dituliskan sekali saja secara terpisah dalam bentuk fungsi-fungsi. Selanjutnya bagian program yang membutuhkan langkah-langkah ini tidak perlu menuliskannya, tetapi cukup memanggil fungsi-fungsi tersebut.



Bahasa C++ memiliki dua macam fungsi, yaitu *built-in function* dan *user-defined function*. Built-in function adalah fungsi yang sudah disediakan oleh compiler dan siap digunakan oleh pemrogram. Sedangkan User-defined function merupakan fungsi yang didefinisikan sendiri atau dibuat sendiri oleh pemrogram.

BUILT-IN FUNCTION DALAM BAHASA C++

Built-in function dalam bahasa C++ ada banyak sekali. Fungsi-fungsi tersebut dikategorikan dan dikemas dalam file header. Tiap file header mengandung beberapa fungsi yang memiliki kegunaan khusus.

Jika fungsi tersebut akan digunakan, maka harus dipanggil di program dengan menuliskan file headernya dalam bagian *pre-processor* program, diawali dengan tanda `#include<header>`.

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

int main()
{
    Deklarasi variabel-variabel
    . . . . .
    . . . . .
    Statement-statementen
    . . . . .
    . . . . .
}
```

Pre processor

Fungsi/program

Badan program

Tabel berikut ini akan merangkum beberapa kategori file header dilengkapi dengan penjelasan operasi dan contoh fungsi-fungsi dan kegunaannya.

No.	Nama file header	Operasi header	Contoh fungsi-fungsi
1	<iostream>	<i>input output stream header</i> yang digunakan sebagai standar input output operasi yang digunakan di C++.	cout : merupakan fungsi keluaran atau untuk menampilkan data dengan tipe apapun ke layar. cin : merupakan fungsi masukan yang dapat memasukkan data berupa numerik dan karakter. endl : merupakan fungsi yang digunakan untuk perintah newline atau pindah baris.
2	<cmath> <math.h>	header untuk operasi-operasi matematika	sqrt() : merupakan fungsi untuk menghitung akar dari suatu bilangan. pow() : fungsi ini untuk menghitung hasil dari perpangkatan bilangan. sin(), cos(), tan() : fungsi ini digunakan untuk menghitung nilai sinus, cosinus, dan tangen dari sudut yang dimasukkan.
3	<stdlib>	Header untuk operasi pembandingan dan operasi konversi	max() : fungsi yang digunakan untuk menentukan nilai maximum dari bilangan yang diinputkan. min() : fungsi yang digunakan untuk menentukan nilai minimum dari bilangan yang diinputkan. atof() : fungsi ini digunakan untuk mengkonversi nilai string bilangan bertipe double. atoi() : fungsi ini digunakan untuk mengkonversi nilai string bertipe integer.

4	<string.h>	header untuk melakukan manipulasi string	strcpy() : fungsi ini digunakan untuk menyalin suatu string ke variabel tujuan. strlen() : fungsi ini digunakan untuk menghitung jumlah karakter yang ada dalam suatu string. strcmp() : fungsi ini digunakan untuk membandingkan 2 buah string. strcat() : fungsi ini digunakan untuk menggabungkan 2 buah string.
5	<stdio.h>	header untuk <i>standar input output header</i> yang digunakan sebagai standar input output operasi yang digunakan oleh bahasa C, akan tetapi bisa juga digunakan didalam bahasa C++.	printf() : merupakan fungsi keluaran yang digunakan untuk menampilkan informasi/pesan kelayar secara terformat (menentukan tipe data yang akan dikeluarkan). puts() : merupakan fungsi keluaran yang digunakan untuk menampilkan informasi/pesan kelayar yang bertipe data string (tanpa melakukan penentuan tipe data terlebih dahulu). putchar() : merupakan fungsi keluaran yang digunakan untuk menampilkan informasi/pesan yang bertipe data char (tanpa melakukan penentuan tipe data terlebih dahulu). scanf() : merupakan fungsi masukan yang digunakan untuk menginputkan data numerik, karakter dan string secara terformat (menentukan tipe data yang akan dimasukkan). gets() : merupakan fungsi masukan yang khusus untuk menerima masukan tipe data string (tanpa harus melakukan penentuan tipe data terlebih dahulu).

USER-DEFINED FUNCTION DALAM BAHASA C++

User-defined function merupakan fungsi yang ditentukan oleh pengguna dan memungkinkan program untuk dibagi menjadi modul-modul (subprogram) yang masing-masing khusus melakukan tugas tertentu. Suatu fungsi terdiri dari tiga terminologi yaitu : Prototipe fungsi (*Function prototype*) untuk mengenalkan atau mendeklarasikan suatu fungsi, pemanggilan fungsi (*Function call*) untuk memanggil suatu fungsi di program utama dan implementasi fungsi (*Function imlementation*) yang digunakan untuk membuat dan mendefinisikan suatu fungsi. Prototipe fungsi menyediakan informasi untuk *compiler* tentang jenis data yang dikembalikan oleh suatu fungsi, jumlah parameter yang diharapkan diterima oleh fungsi, tipe data parameter dan urutan dari parameter.

```
#include<header>
#include<header>
```

}

pre

```
<tipe><namafungsi_1>(tipe argumen1,tipe argumen2,..., tipe argumenN);
```

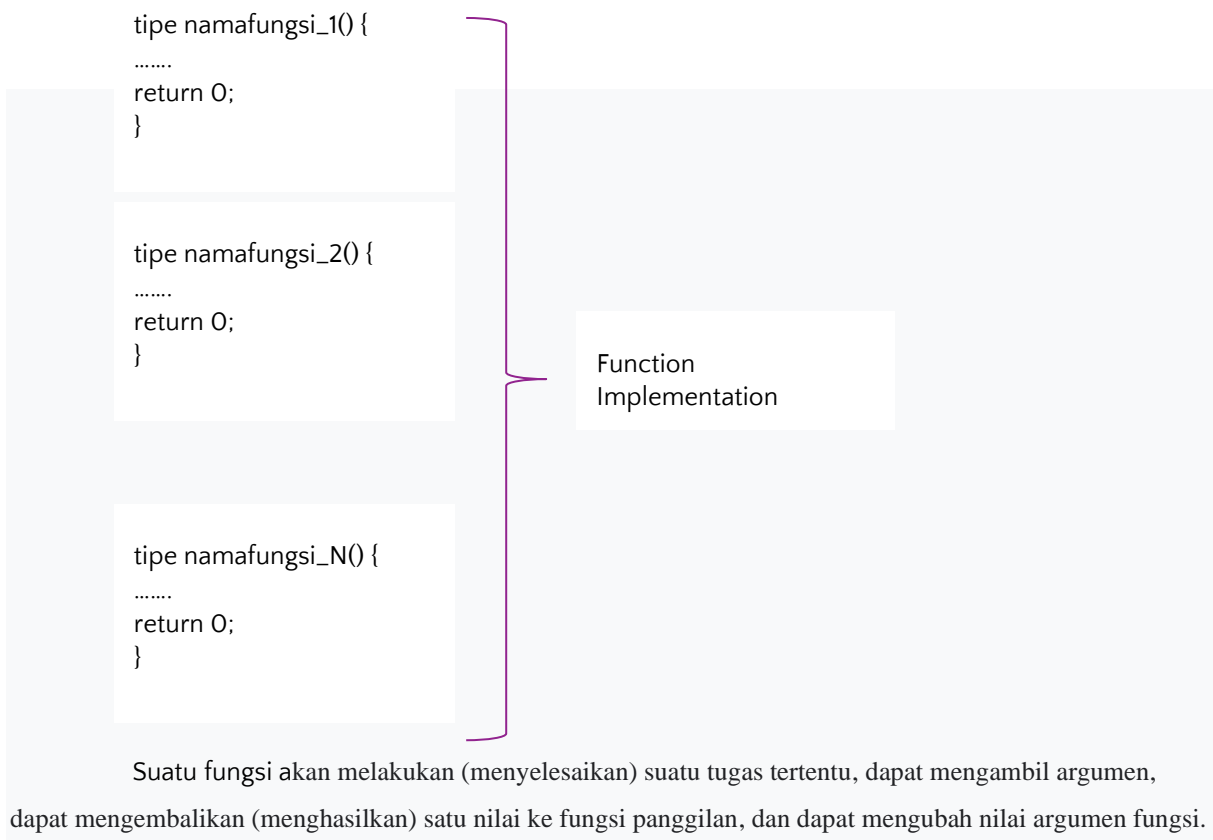
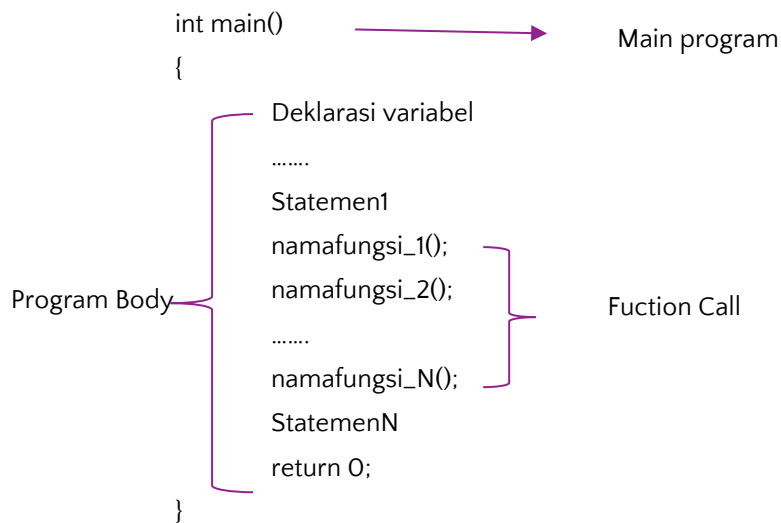
}

Fuction Prototype

<type><namafungsi_2>(tipe argumen1,tipe argumen2,..., tipe argumenN);

...

<type><namafungsi_N>(tipe argumen1,tipe argumen2,..., tipe argumenN);



HINT

Modularitas adalah satu konsep yang mengawali adanya pemrograman berorientasi objek (Object Oriented Programming), yang akan dibahas lebih lanjut dalam mata kuliah dan praktikum PBO.

PENDEFINISIAN FUNGSI

Cara mendefinisikan fungsi adalah sebagai berikut :

```
<type><namafungsi>(tipe argumen1,tipe argumen2,... tipe argumenN)

{

    Deklarasi variabel-variabel;

    Statement-statement ; .....    // tubuh fungsi

}
```

Keterangan :

Tipe pada fungsi tergantung dari tipe data hasil balik (return type) yang akan diberikan oleh fungsi. Jika suatu fungsi memberikan hasil balik, maka nilai hasil balik yang diberikan oleh fungsi dapat dilakukan oleh statement sebelum return yang diikuti oleh nilai hasil baliknya yang ditulis dalam kurung. Jika fungsi tidak memberikan hasil balik, maka tipenya adalah void.

Contoh Fungsi :

```
int tambah(int a, int b)
{
    int sum;
    sum = a+b;
    return sum;
}
```

```
int tambah(int a, int b)
{
    int sum;
    sum = a+b;
    cout<<"Hasil penjumlahan a dan b = "<<sum<<endl;
    return 0;
}
```

PARAMETER FUNGSI DAN *PASSING PARAMETER*

Passing parameter merupakan metode yang digunakan untuk menentukan nilai argumen berhubungan dengan parameter. Parameter formal ditulis dalam prototipe fungsi dan header fungsi dari definisi. Parameter formal adalah variabel lokal yang diberi nilai dari argumen saat fungsi dipanggil. Ketika suatu fungsi dipanggil, nilai (ekspresi) yang diteruskan dalam panggilan disebut argumen atau parameter aktual (kedua istilah memiliki arti yang sama). Pada saat panggilan, setiap parameter aktual ditetapkan ke parameter formal yang sesuai dalam definisi fungsi. Untuk parameter nilai (default), nilai parameter aktual ditetapkan ke variabel parameter formal. Untuk parameter referensi, alamat memori dari parameter aktual ditetapkan ke parameter formal.

Function prototype

```
int func1 (int a, char b, float& c);
```

Function call

```
int main {
    func1(5 * 3, 'A', z);
}
```

Parameters

argument

Function definition

```
int func1 (int a, char b, float& c) {
    int x,
    char y;
    ...
}
```

Parameters

Local variable

Sebuah variabel diasosiasikan dengan sebuah nilai dan a lokasi memori. Dalam C++, ada dua metode untuk melewati parameter (*passing parameter*):

- Melewati nilai (*call-by-value*). Ini dilambangkan dengan mendeklarasikan jenis dan nama parameter.
- Melewati lokasi memori (*call-by-reference*). Ini dilambangkan dengan menambahkan simbol & di sebelah tipe parameter

```
void p(int x, int& y) { ... }
```

Call-by-value

Call-by-reference

Metode *call-by-value* akan menyalin nilai argumen dan salinan nilai diteruskan ke dalam daftar parameter formal daftar. Perubahan yang dilakukan pada nilai daftar parameter tidak berpengaruh pada nilai-nilai variabel yang telah dipanggil.

Contoh :

```
int fungsi (int x);
int main(void) {
    int a = 2;
    int b= fungsi (a);
}
int fungsi (int x)
{
    x= x+ 1;
    return x;
}
```

CONTOH FUNGSI

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

void add_print(int , int); //function declaration

int main() //main program
{
    int a=4;
    int b=5;
    cout<<"Fungsi untuk mencetak hasil penambahan dua bilangan"<<endl;
    add_print(a,b); //function call

    return 0;
}

//function definition
void add_print(int val1,int val2){
    int c;
    cout<<"Bilangan yang dijumlahkan : "<<val1<<" dan "<<val2<<endl;
    c=val1+val2;
    cout<<"Hasil penjumlahannya adalah : "<<c<<endl;
}
```

LAB 5.3: LATIHAN

PERCOBAAN 1 : *BUILD-IN FUNCTION*

Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut.

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int a = 69;
    int b = 4;
    int c = 2;
    int d = 0;
    float pangkat, akar, vsin ,vcos, vtan;
    akar = sqrt(a);
    cout<<"Akar dari "<<a<<" adalah"<<akar<<endl;
    pangkat = pow(b,c);
    cout<<b<<" pangkat "<<c<<" adalah"<<pangkat<<endl;
    vsin = sin(d);
    cout<<"sin sudut "<<d<<" derajat adalah"<<vsin<<endl;
```

```
vcos = cos(d);
cout<<"cos sudut "<<d<<" derajat adalah"<<vcos<<endl;
vtan = tan(d);
cout<<"tan sudut "<<d<<" derajat adalah"<<vtan;
return 0;
}
```

1. Jenis fungsi apa yang digunakan dalam program di atas ?
2. Apakah keluaran dari program di atas ?

PERCOBAAN 2 : BUILD-IN FUNCTION

Berikut ini adalah sebuah program dalam bahasa C++. Salinlah program berikut menggunakan text editor dan simpan, kemudian *compile* dan *run* (eksekusi) program tersebut. Contoh di bawah ini hanya terdiri dari sebuah fungsi, yaitu fungsi main.

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

int main ()
{
char angka[4]; // variabel angka untuk menyimpan angka bertipe data char
int a; // variabel a bertipe data integer
cout<<"Masukan karakter angka = "<<endl;
cin.get(angka,4);
a = atoi(angka); //variabel angka bertipe data char dikonversikan
menjadi variabel a yang bertipe data int
cout<<"Angka dalam tipe character = "<<angka<<endl;
cout<<"Angka dalam tipe integer = "<<a<<endl;
return 0;
}
```

1. Isilah variabel angka dengan karakter 561, apakah keluaran dari program ?
2. Ulangi, isilah variabel angka dengan karakter 1234, apakah keluaran dari program ? Apakah terdapat kesalahan pada program tersebut? Jelaskan jika ada kesalahan. Bila terdapat kesalahan, cobalah perbaiki program tersebut hingga dapat di-*compile* dan memberikan hasil yang sesuai.

PERCOBAAN 3 : BUILD-IN FUNCTION

Pada percobaan ini, salinlah program berikut menggunakan *text editor* dan simpan, kemudian *compile* dan eksekusi program tersebut.

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

int main()
{
string nama;
```



```
int ukur,ukur2;
cout << "Masukkan namamu:"<<endl;
getline(cin,nama);
cout << "Nama yang dimasukkan: " <<nama<<endl;
ukur = nama.size();
cout<<"Ukuran dari string nama dengan perintah size : "<<ukur<<endl;
ukur2 = nama.length();
cout<<"Ukuran dari string nama dengan perintah length : "<<ukur2<<endl;
}
```

1. Isilah dengan namamu, apakah keluaran dari program ?
2. Apakah hasil nilai dariukur dan ukur2 sama ? Jelaskan

PERCOBAAN 4 : BUILD-IN FUNCTION

Pada percobaan ini, salinlah program berikut menggunakan *text editor* dan simpan, kemudian *compile* dan eksekusi program tersebut.

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1,str2, str3, str4, str5;
    int len1, len2, len3,len4,len5 ;

    cout << "Masukkan 3 kata yang berbeda:"<<endl;
    cout << "Masukkan kata pertama:"<<endl;
    getline(cin,str1);
    cout << "Masukkan kata kedua:"<<endl;
    getline(cin,str2);
    cout << "Masukkan kata ketiga:"<<endl;
    getline(cin,str3);

    cout << "Kata pertama : " << str1 << endl;
    cout << "Kata kedua : " << str2 << endl;
    cout << "Kata ketiga : " << str3 << endl;

    // copy of string
    str4 = str1;

    // concatenates string
    str4 = str4.append(str2);
    cout << "Kata keempat : " << str4 << endl;
    str5 = str4 + str3;
    cout << "Kata kelima : " << str5 << endl;

    // copy of substring
    string str6(str5,0,8);
    cout << "Kata keenam : " << str6 << endl;

    //size of string
    len1 = str1.size();
    cout << "Ukuran Kata Pertama : " << len1 << endl;
    len2 = str2.size();
```

```
cout << "Ukuran Kata Kedua : " << len2 << endl;
len3 = str3.size();
cout << "Ukuran Kata Ketiga : " << len3 << endl;
len4 = str4.size();
// total length of str3 after concatenation
cout << "Ukuran Kata Keempat : " << len4 << endl;
len5 = str5.size();
cout << "Ukuran Kata Kelima : " << len5 << endl;
return 0;
}
```

1. Isilah 3 kata yang berbeda, apakah keluaran dari program ? Jelaskan
2. Apa perbedaan fungsi copy dan copy substring ?

PERCOBAAN 5 : BUILD-IN FUNCTION

Pada percobaan ini, salinlah program berikut menggunakan *text editor* dan simpan , kemudian *compile* dan eksekusi program tersebut.

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1 = "Happy Year";
    string str2 = "New ";
    string str3, str4, str5, str6, str7;

    str3 = str1.insert(6, str2 );
    cout << "str3 : " << str3 << endl;

    str4 = str1.replace(6, 3, str2, 0, 3 );
    cout << "str4 : " << str4 << endl;

    str5 = str3.substr(6, 4);
    cout << "str5 : " << str5 << endl;
    str6 = str3.substr(0, 5);
    cout << "str6 : " << str6 << endl;

    cout << str3.find("New") << endl;
    cout << str3.find("ar") << endl;
    cout << str3.find("ro") << endl;

    str7 = str3.erase(6);
    cout << "str7 : " << str7 << endl;
    str7 = str1.replace(6, 3, str2, 1, 2 );
    cout << "str7 : " << str7 << endl;

    return 0;
}
```

1. Apa perbedaan fungsi insert dan replace ?
2. Apa kegunaan fungsi find dan erase ?
3. Apa kegunaan fungsi substract ?
4. Jelaskan keluaran program di atas !

PERCOBAAN 6 : USER-DEFINED FUNCTION

Program di bawah ini adalah program yang memanfaatkan *user-defined function*. Salin, *compile*, dan eksekusi program di berikut ini.

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

void add_print(int , int); //function declaration

int main() //main program
{
    int a=4;
    int b=5;
    cout<<"Fungsi untuk mencetak hasil penambahan dua bilangan"<<endl;
    add_print(a,b); //function call

return 0;
}

//function definition
void add_print(int val1,int val2){
    int c;
    cout<<"Dua Bilangan yang akan dijumlahkan adalah: "<<val1<<" dan
"<<val2<<endl;
    c=val1+val2;
    cout<<"Hasil penjumlahannya adalah : "<<c<<endl;
}
```

1. Apakah terdapat kesalahan pada program tersebut? Bila terdapat kesalahan, cobalah perbaiki program tersebut hingga dapat di-*compile* dan memberikan hasil yang benar!
2. Apakah tujuan dari program di atas?

PERCOBAAN 7 : PARAMETER FUNGSI DAN PASSING PARAMETER

Program di bawah ini adalah program yang memanfaatkan *user-defined function*. Salin, *compile*, dan eksekusi program di berikut ini. Jelaskan keluarannya.

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

void area (float);

int main ( )
{
    float radius;
    cin >> radius;
    area (radius ) ;

}

void area (float r)
```

```
{
float A;
A = 3.1416 *r* r;
cout << "The area of the circle is " << A << endl;

}
```

1. Sebutkan mana yang disebut variabel lokal
2. Sebutkan mana yang disebut variabel aktual
3. Sebutkan mana yang disebut variabel formal

LAB 1.4: CEK POIN 1

1. Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut. Jika terdapat syntax error atau error karena kesalahan dalam hal penggunaan tipe data, operator, ataupun kesalahan logika, perbaiki hingga dapat dikompilasi dan dijalankan sehingga menghasilkan luaran seperti yang diharapkan.

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int a = 69;
    int b = 4;
    int c = 2;
    int d = 0;
    int e = -4;
    float hasil1, hasil2, hasil3, hasil4;
    hasil1 = sqrt(a)/d;
    cout<<"Hasil dari sqrt(a)/d adalah "<<hasil1<<endl;
    hasil2 = sqrt(e);
    cout<<" Hasil dari sqrt(e) adalah "<<hasil2<<endl;
    hasil3 = min(c)
    cout<<" Hasil dari min(c) adalah "<<hasil3<<endl;
    hasil4 = abs(e);
    cout<<" Hasil dari abs(e) adalah "<<hasil4<<endl;

    return 0;
}
```

2. Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut. Jika terdapat error karena kesalahan dalam hal penggunaan tipe data, operator, ataupun kesalahan logika, perbaiki hingga dapat dikompilasi dan dijalankan sehingga menghasilkan luaran seperti yang diharapkan. Jelaskan keluaran program tersebut.

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

```

int main()
{
    char s1[10],s2[20],s3[20],i;
    cout<<"Masukkan string pertama untuk ditemukan panjang string tersebut :";
    cin>>s1;
    cout<<"\nString yang dimasukkan : "<<s1;

    cout<<"\nPanjang String tersebut adalah : "<<strlen(s1);
    cout<<"\nMasukkan string kedua untuk ditemukan panjang string tersebut :";
    cin>>s2;
    cout<<"\nString yang dimasukkan : "<<s2;

    cout<<"\nPanjang String tersebut adalah : "<<strlen(s2);

    strcpy(s3,s2);
    cout<<"\nSalin string kedua dalam string ketiga ";
    cout<<"\nString ketiga adalah : " <<s3;

    cout<<"\nSambungkan string pertama dan kedua:";
    strcat(s1,s2);
    cout<<"\nHasil penyambungan string pertama dan kedua adalah : "<<s1<<endl;

    cout<<endl<<"\nBandingkan string pertama dan kedua :";

    i=strcmp(s1,s2);
    if(i==0)
        cout<<"\nBoth strings are equal\n";
    else if(i<0)
        cout<<s1<<" is less than "<<s2<<endl;
    else
        cout<<s1<<" is greater than "<<s2;

    cout<<endl<<"\nBandingkan string kedua dan ketiga :";

    i=strcmp(s2,s3);
    if(i==0)
        cout<<"\nBoth strings are equal\n";
    else if(i<0)
        cout<<s2<<" is less than "<<s3<<endl;
    else
        cout<<s2<<" is greater than "<<s3;

    return 0;
}

```

- Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut. Jika terdapat syntax error atau error karena kesalahan dalam hal penggunaan tipe data, operator, ataupun kesalahan logika, perbaiki hingga dapat dikompilasi dan dijalankan sehingga menghasilkan luaran seperti yang diharapkan. Jelaskan tentang passing parameter yang terjadi di program ini.

```

#include <iostream>
using namespace std;

```

```
float area (float);

int main ( )
{
    int radius, y;
    cin >> radius;
    y = area (radius);
    cout << "The area of the circle is " << y;
}

float area (float r)
{
    return ( 3.1416 *r* r );
}
```

LAB 5.4: CEK POIN 2

1. Buatlah program C++ untuk mendapatkan nomor telpon lengkap dengan memasukkan tiga komponen yaitu kode negara, kode kota dan nomor telepon dalam bentuk string, lalu menggabungkan menjadi nomor telpon lengkap. Kemudian hapuslah kode negara dan kode kota dengan memanfaatkan fungsi fungsi yang ada di header string.
2. Buatlah program C++ yang terdiri dari program utama dan 3 fungsi/modul/subprogram. Fungsi pertama untuk memasukkan data panjang diameter suatu bola dan data penunjang lainnya, fungsi kedua untuk menghitung luas permukaan bola dan volume bola, fungsi ketiga untuk mencetak diameter bola, luas permukaan bola dan volume bola. Sebutkan yang mana yang termasuk parameter formal, parameter aktual dan variabel lokal.

LAB 5.4: CEK POIN 3

1. Buatlah Buatlah program C++ untuk memasukkan NIM anda dalam bentuk string, lalu memecah NIM menjadi 4 luaran, yaitu : Angkatan, NIU, Kode Fakultas, NIF dengan memanfaatkan fungsi fungsi yang ada di header string.
2. Buatlah program C++ untuk menentukan IPK mahasiswa, program yang terdiri dari program utama dan 3 fungsi/modul/subprogram. Fungsi pertama untuk memasukkan 5 data NIM, Matakuliah, sks, Nilai. Fungsi kedua untuk memasukkan menghitung IPK, fungsi ketiga untuk mencetak NIM dan IPK.

LAB 5.5: REFERENSI

Di bawah ini adalah referensi online dari Modul 5. Apabila mengalami kesulitan dalam memahami materi, silahkan buka referensi di bawah ini :

- <https://www.cplusplus.com/doc/tutorial/functions/>

LAB 5.6: SEKILAS MATERI MODUL BERIKUTNYA

Materi terakhir dari pemrograman dasar adalah Struktur Data sederhana, berupa Array.