VERSION 2.0

AUGUST, 2022

PEMROGRAMAN DASAR

MODUL 6 – FUNGSI

DISUSUN OLEH:

- Alif Fatwa Ramadhani
- Azka Faza Dzulqarnain

DIAUDIT OLEH:

- Hardianto Wibowo, S.Kom, M.T

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN DASAR

TARGET PRAKTIKUM

1. Mampu menguasai konsep pemrograman
2. Mengiplementasikan fungsi dengan benar
3. Memahami definisi dan kegunaan fungsi
4. Memahami penggunaan fungsi

PERSIAPAN SOFTWARE/APLIKASI

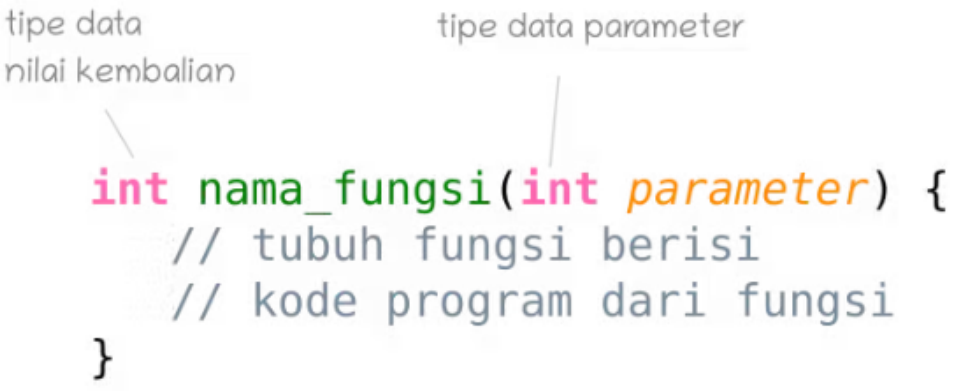
- Komputer/Laptop
- Software (Falcon/Dev C++)
- Bahasa Pemrograman C/C++

MATERI PRAKTIKUM

Dalam pemrograman, fungsi atau prosedur sering digunakan untuk membungkus program menjadi bagian-bagian kecil. Tujuannya agar program tidak menumpuk pada fungsi *main()* saja. Bayangkan saja, kalau program kita tambah besar dan kompleks. Kalau semua kodenya ditulis di dalam fungsi *main()*, maka kita akan kesulitan membacanya. Maka dari itu, kita harus menggunakan fungsi.

APA ITU FUNGSI?

Fungsi adalah sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain dan juga hanya berjalan ketika dipanggil. Fungsi dapat menerima input dan menghasilkan output. Contoh fungsi yang sering kita buat adalah fungsi *main()*. Fungsi ini memang wajib ada di setiap program C++, karena fungsi inilah yang akan dieksekusi pertama kali saat program berjalan. Berikut adalah struktur fungsi



Fungsi biasanya akan mengembalikan sebuah nilai dari hasil prosesnya. Karena itu, kita harus menentukan tipe data untuk nilai yang akan dikembalikan. Apabila fungsi tersebut tidak memiliki nilai kembalian, maka kita harus menggunakan tipe *void* untuk menyatakan kalau fungsi tersebut tidak akan mengembalikan nilai apa-apa. Contohnya seperti ini:

```
void nama_fungsi(){
    cout << "Ini adalah sebuah fungsi\n";
}
```

Lalu untuk parameter bersifat opsional, boleh ada boleh tidak. Tergantung dari fungsi yang dibuat. Jika fungsi itu membutuhkan input, maka kita harus membuatkan paramter. Tapi kalau tidak menerima input apapun, ya tidak perlu dibuat. Fungsi yang tidak menerima input, kadang juga disebut dengan prosedur. Untuk lebih memahami apa itu prosedur, silakan kalian salin kode dibawah ini.

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

// membuat fungsi say_hello()
void say_hello(){
    cout << "Hello Selamat Datang di Laboratorium
Informatika!\n";
}

int main(){
    // memanggil fungsi say_hello()
    say_hello();

    return 0;
}
```

Kalian juga bisa memanggil fungsi *say_hello()* tersebut berkali-kali di fungsi *main().*

DEKLARASI FUNGSI

Pada contoh di atas, kita membuat fungsi dengan cara mendefinisikan langsung fungsinya. Kita juga bisa membuatnya dengan deklarasi.

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

// deklarasi fungsi
void say_hello();

int main(){
    // memanggil fungsi say_hello()
    say_hello();
    say_hello();
    say_hello();

    return 0;
}

// Definisi fungsi
void say_hello(){
    cout << "Hello Selamat Datang di Laboratorium Informatika!\n";
}
```

Apa bedanya dengan yang tadi? Jika kita membuat fungsi secara definisi, kita harus membuat fungsinya di atas fungsi main. Jika dibuat di bawah fungsi main, maka program akan error. Soalnya program C++ dieksekusi dari atas ke bawah. Tapi berkat deklarasi, masalah ini bisa teratasi.

APA ITU PARAMETER?

Parameter adalah variabel yang menyimpan nilai untuk diproses di dalam fungsi. Parameter berfungsi untuk menyimpan nilai yang akan diinputkan ke fungsi. Untuk lebih memahami apa itu parameter, silakan amati kode dibawah ini.

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

void say_hello(string name){
    cout << "Hello " << name << "!\n";
}

int main(){
    say_hello("Azka");
    say_hello("Alif");
    return 0;
}
```

Silakan diketik kembali di text editor masing-masing agar lebih paham tentang parameter. Kita juga bisa menggunakan 2 parameter sekaligus di sebuah fungsi, berikut contohnya

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

void fungsiTambah(int a, int b){
    printf("%d + %d = %d\n", a, b, a+b);
}

int main(){
    fungsiTambah(1, 4);
    fungsiTambah(8, 2);
    fungsiTambah(3, 2);
    return 0;
}
```

Pada contoh-contoh fungsi diatas, kita menggunakan fungsi bertipe void dimana fungsi ini tidak mengembalikan apa-apa, karena tipe data yang diberikan pada nilai kembalian adalah void. Fungsi juga terkadang harus menghasilkan output. Mengapa demikian? Karena kita membutuhkan hasil dari fungsi tersebut untuk digunakan pada proses berikutnya. Kita bisa menggunakan kata kunci *return* untuk mengembalikan nilai dari fungsi. Berikut contohnya:

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

float bagi(int a, int b){
    float hasil = (float)a / (float)b;
    return hasil;
}

int main(){
    printf("Hasil 5/2: %.2f\n", bagi(5, 2));
    return 0;
}
```

PASS BY VALUE DAN PASS BY REFERENCE

Pass by value dan pass by reference adalah cara untuk memberikan nilai pada paramaeter. Perhatikan contoh dibawah ini

Contoh 1

```cpp
kali_dua(4);
```

Contoh 2

```cpp
kali_dua(&nama_variabel);
```

Perbedaanya adalah di parameter kedua fungsi tersebut, dimana contoh 1 disebut pass by value karena disana kita memberikan nilai 4 secara langsung. Sedangkan contoh 2 adalah pass by reference dimana kita memberikan alamat memori. Untuk lebih memahami silakan ketik ulang kode di bawah ini dan jalankan.

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

void kali_dua(int *num){
    *num = *num * 2;
}

int main(){
    int angka = 9;

    // memanggil fungsi
    kali_dua(&angka);

    // mencetak isi variabel
    // setelah fungsi dipanggil
    cout << "isi variabel angka = " << angka << endl;

    return 0;
}
```

FUNGSI REKURSIF

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Maksudnya bagaimana? biasanya kita memanggil fungsi pada fungsi main atau fungsi yang lainnya. Namun, pada fungsi rekursif hal tersebut akan memanggil dirinya sendiri di dalam tubuh fungsi. Untuk lebih memahami Rekursif, silakan perhatikan kode dibawah ini

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

// deklarasi fungsi
int sum(int n);

int main(){
    int number, result;

    printf("Enter a positive integer: ");
    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

    return 0;
}

// definisi fungsi
int sum(int num){
    if (num!=0)
        return num + sum(num-1); // fungsi sum( ) memanggil dirinya sendiri
    else
        return num;
}
```

Dimana program tersebut jika kita input angka 5 akan mengeluarkan hasil output 15. Mengapa demikian? Karena dari 5 tersebut akan dijumlah menjadi 5+4+3+2+1= 15. Silakan dicoba ketik ulang dan menjalankan kode tersebut.

LAB ACTIVITY

Cobalah dengan memperbaiki program di bawah ini. (Pilih salah satu bahasa program yang kalian gunakan).

1) Memanggil Fungsi

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

int main() {
  myFunction();
  return 0;
}

void myFunction() {
  cout << "I just got executed!";
}
```

2) Memanggil Fungsi (2)

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

void myFunction() {
  cout << "I just got executed!";
}

int main() {
  myFunction("Hallo");
  return 0;
}
```

3) Implementasi Parameter

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

void myFunction(string fname) {
  cout << fname << " Refsnes\n";
}

int main() {
  myFunction('Azka');
  myFunction(1987);
  myFunction("Alif");
  return 0;
}
```

4) Default Parameter (Silakan kalian cek, apakah kode dibawah ini error atau tidak)

```cpp
void myFunction(string country = "Norway") {
  cout << country << "\n";
}

int main() {
  myFunction("Sweden");
  myFunction("India");
  myFunction();
  myFunction("USA");
  return 0;
}
```

5) Multiple Parameter

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

void myFunction(string fname, int age) {
  cout << fname << " Refsnes. " << age << " years old. \n";
}

int main() {
  myFunction(22 ,"Azka");
  myFunction("Jenny", 21);
  myFunction(21, "Alif");
  myFunction("Putro", 20);
  return 0;
}
```

TUGAS PRAKTIKUM

Bedasarkan tema tugas besar yang kalian pilih, silakan untuk memulai dan mengembangkan program kalian dengan kriteria sebagai berikut:

- Telah mengimplementasikan fungsi pada program dengan benar
- Telah menggunakan fungsi dengan benar
- Telah mendeklarasikan fungsi dengan benar
- Telah mengimplementasikan parameter pada program dengan benar\
- Telah menggunakan parameter dengan benar

DETAIL PENILAIAN PRAKTIKUM

Ketentuan	Bobot Penilaian
Dapat mengiplementasikan materi dari modul praktikum	20%
Dapat menjelaskan program dan materi dari modul praktikum	40%
Program berhasil berjalan	10%
Menjawab pertanyaan asisten	10%