

PRAKTIKUM DASAR PEMROGRAMAN
MODUL IX
FUNGSI

	Awal mengerjakan projek
	Mulai ketemu error
	Udah Deadline

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
2022

TUJUAN

1. Praktikan dapat mengerti dan memahami konsep dan struktur Fungsi.
2. Praktikan dapat mengerti perbedaan Fungsi dan Prosedur dan mampu mengimplementasikan didalam kasus nyata.

PENGERTIAN FUNGSI

Fungsi merupakan sebuah modul di dalam pengkodean yang mengerjakan kasus atau pekerjaan tertentu. Sehingga Fungsi tidaklah sama dengan prosedur dimana Fungsi hanya memiliki satu tujuan dan menyelesaikan satu masalah. Cara kerja Fungsi ialah Fungsi akan mengambil data, kemudian memprosesnya, dan mengeluarkan output berupa hasil balikan sesuai tipe datanya. Sebuah Fungsi dapat digunakan berulang ulang kali, maka dari itu salah satu tujuan Fungsi ialah untuk membuat pengkodean menjadi lebih efisien dan rapi.

STRUKTUR FUNGSI

Fungsi memiliki struktur yang tidak jauh berbeda dari prosedur. Struktur Fungsi dapat dijelaskan sebagai berikut:

Deklarasi Fungsi :

```
tipeData_balikan nama_Fungsi(tipeData param1, tipeData param2, dst..);
```

Pemakaian Fungsi :

```
tipeData_balikan nama_Fungsi(tipeData param1, tipeData param2, dst..) {  
    Function body...  
  
    return  
}
```

Keterangan :

1. **tipeData_balikan** – merupakan tipe data dari nilai balikan Fungsi, disini maksudnya nilai balikan adalah nilai / value yang akan dihasilkan dari Fungsi tersebut (int, char, float, bool).

2. **nama_Fungsi** – merupakan nama yang akan digunakan untuk mendeklarasikan Fungsi dan juga untuk pemakaiannya. Nama Fungsi adalah case sensitive dan memiliki format pada umumnya (luasSegitiga, volumeTabung, nilaiTerbesar).
3. **param** – merupakan parameter seperti pada prosedur dimana berfungsi sebagai placeholder atau variable-variable yang akan digunakan di dalam Fungsi tersebut. Parameter di dalam Fungsi merupakan parameter inputan dan tidak memiliki parameter output seperti Prosedur.
4. **Function body** – merupakan isi dari fungsi yang ingin dibuat, berisi kode dan syntax yang relevan dengan nama fungsinya dan mengerjakan suatu task atau perintah khusus tertentu.
5. **return** – merupakan syntax yang harus ada dan wajib ada dalam setiap pembuatan Fungsi, dimana digunakan untuk mengeluarkan nilai atau value dari hasil operasi Fungsi tersebut.

Contoh :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
//Deklarasi
int bilanganKuadrat(int bilangan);
int main(){
    int angka, hasilKuadrat;
    //input
    printf("Input angka yang ingin dikuadratkan : ");scanf("%d",&angka);
    //perhitungan fungsi
    hasilKuadrat = bilanganKuadrat(angka);
    //output
    printf("Hasil bilangan kuadrat dari %d adalah %d", angka, hasilKuadrat);
    return 0;
}
int bilanganKuadrat(int bilangan){
    int total;
    total = bilangan*bilangan;

    return total;
}
```

Prosedur vs Fungsi

Fungsi	Prosedur
<p>Didefinisikan dengan tipe data yang nilainya akan dikembalikan. Contoh:</p> <pre>int bilanganKuadrat(int bilangan);</pre>	<p>Didefinisikan dengan void. Contoh:</p> <pre>void bilanganKuadrat(int *hasil, int bilangan);</pre>
<p>Pada body fungsi, alur logika harus berakhir dengan RETURN untuk mengembalikan sebuah nilai. Contoh:</p> <pre>int bilanganKuadrat(int bilangan) { return bilangan*bilangan; }</pre>	<p>Tidak membutuhkan return untuk mengembalikan sebuah nilai, cukup menggunakan parameter output. Contoh:</p> <pre>void bilanganKuadrat(int *hasil, int bilangan) { (*hasil) = bilangan*bilangan; }</pre>
<p>Pemanggilan fungsi harus menempel pada suatu statement dan diperlakukan sebagai value</p> <p>Contoh:</p> <pre>hasilKuadrat = bilanganKuadrat(bil);</pre> <p>atau</p> <pre>printf("Hasilnya adalah %d", bilanganKuadrat(bil));</pre>	<p>Pemanggilan harus berdiri sendiri dan diperlakukan sebagai statement Contoh:</p> <pre>bilanganKuadrat(&hasilKuadrat, bil);</pre>
Hanya terdapat parameter input	Terdapat parameter input dan output.
Menghasilkan sebuah nilai.	Menghasilkan efek netto.

GUIDED

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <math.h>
#define phi 3.14

typedef char string[20];

float volumeTabung(float jari, float tinggi);
float luasPermukaanTabung(float jari, float tinggi);
float bilanganKuadrat(float bilangan);

int main(int argc, char *argv[])
{
    int menu;
    float jari, tinggi;
    string ans;

    printf("\n\n\n\t\tAPAKAH KAMU MANUSIA [Y/N] ? ");
    fflush(stdin);
    gets(ans);

    if (strcmpi(ans, "Y") == 0)
    {
        system("color 20");
        printf("\n\n\n\t\tYEAY ANDA MANUSIA :)\n\n");
        printf("\n\n\n\t\ttekan apapun untuk melanjutkan guided :D Semangat\n");
        getch();
        do
        {
            system("color 60");
            system("cls");
            printf("\n\t\t===== GUIDED FUNGSI =====\n\n");
            printf("\t[1] Input Data Tabung\n");
            printf("\t[2] Hitung Volume Tabung\n");
            printf("\t[3] Hitung Luas Permukaan Tabung\n");
            printf("\t[0] EXIT\n");
            printf("\tMasukan Pilihan >>> ");
            scanf("%d", &menu);
            switch (menu)
```

```

switch (menu)
{
case 1:
    printf("\n\n\tInput Data Tabung");
    printf("\n\n\tMasukan Jari-jari      : ");
    scanf("%f", &jari);
    printf("\tMasukan Tinggi          : ");
    scanf("%f", &tinggi);
    printf("\n\tInput Data Berhasil");
    break;

case 2:
    printf("\n\n\tMenghitung Volume Tabung");
    printf("\n\t=====");
    printf("\n\tVolume Tabung adalah : %.2f ", volumeTabung(jari, tinggi));
    break;

case 3:
    printf("\n\n\tMenghitung Luas Permukaan Tabung");
    printf("\n\t=====");
    printf("\n\tLuas Permukaan Tabung adalah : %.2f ", luasPermukaanTabung(jari, tinggi));
    break;
}
printf("\n\n\n\t\tTEKAN APAPUN UNTUK MELANJUTKAN :D !\n\n");
getch();
} while (menu != 0);
}
else if (strcmpi(ans, "N") == 0)
{
    system("color 40");
    printf("\n\n\n\t\tOH TIDAK, SYSTEM OUT!! :(\n\n");
}

printf("\n\n\n\t\tSAMPAI JUMPA DI UGD :D, Asik BANGET !\n\n");
return 0;
}

float volumeTabung(float jari, float tinggi)
{
    return phi * bilanganKuadrat(jari) * tinggi;
}

float luasPermukaanTabung(float jari, float tinggi)
{
    return (2 * phi * bilanganKuadrat(jari)) + (2*phi*jari*tinggi);
}

float bilanganKuadrat(float bilangan)
{
    return bilangan * bilangan;
}

```

Informasi tambahan:

Format Pengumpulan : GD9_X_YYYYY.zip

X = Kelas; Y = 5 digit terakhir NPM

Dikumpul file lengkap !

~MASIH BELUM TERLAMBAT UNTUK BELAJAR LAGI~