



Modul 6 : Function

PENGANTAR FUNCTION (20 poin)

Perhatikan blok code di bawah ini!

PROGRAM 1

```
#include <stdio.h>

int main(void){
    int a, b, hasil;
    printf("Masukkan angka pertama: ");
    scanf("%d", &a);
    printf("Masukkan angka kedua: ");
    scanf("%d", &b);
    hasil = a+b;
    printf("Jumlah angka: ");
    printf("%d", hasil);
    return 0;
}
```

1. Apa yang dilakukan oleh kode tersebut? (10 poin)

Kode di atas dapat ditulis ulang dengan menggunakan function menjadi

PROGRAM 2

```
#include <stdio.h>

int penjumlahan(int a, int b){
    return a + b;
}

int main(void){
    int a, b, hasil;
    printf("Masukkan angka pertama: ");
    scanf("%d", &a);
    printf("Masukkan angka kedua: ");
    scanf("%d", &b);
    hasil = penjumlahan(a, b);
    printf("Jumlah angka: ");
    printf("%d", hasil);
    return 0;
}
```

Silahkan coba untuk run kedua program. Output yang akan tampil pasti sama. Pada program di atas, terdapat 1 buah function yang bernama **penjumlahan()**. Di dalam tanda kurung function disebut sebagai **parameter**. Pada program di atas, parameter dari function penjumlahan() adalah int a, int b.

Parameter digunakan untuk memindahkan nilai dari variabel yang ada di main() ke function yang dipanggil. Pada program di atas, nilai dari variabel a dan b yang ada di main() akan dipindah ke function penjumlahan().

Setiap function memiliki tipe data. Pada program di atas, tipe data dari function di atas adalah int atau integer (bilangan bulat). Hal ini dapat dilihat dari

```
int penjumlahan(int a, int b){
    return a + b;
}
```

Karena bertipe int, maka function akan mengembalikan nilai integer. Coba perhatikan di dalam main(), ada baris kode `hasil = penjumlahan(a, b);`. Yang terjadi pada baris tersebut adalah nilai yang dikembalikan oleh function penjumlahan (a,b) akan disimpan pada variabel hasil.

Jadi, secara kesimpulan kalian dapat melakukan penjumlahan di dalam program dengan dua cara, yaitu secara langsung (program 1) atau dengan menggunakan function (program 2).

2. Coba modifikasi program 2, dengan menambahkan function perkalian(int a, int b). Hasil dari function tersebut di print dalam main. Source code di copas. (10 poin)

VOID FUNCTION (20 poin)

Sekarang kita akan belajar tentang function yang bertipe void. Sebelumnya anda telah membuat function yang bertipe int. Ini berarti bahwa function tersebut akan mengembalikan nilai. Akan tetapi, function tidak akan selalu mengembalikan nilai. Terdapat beberapa function yang tidak mengembalikan nilai ketika dipanggil dalam main(). Function ini disebut sebagai **void function**. Coba perhatikan kode di bawah ini!

PROGRAM 3

```
#include <stdio.h>

int main(void){
    int a, b;

    printf("Masukkan nilai a: ");
    scanf("%d", &a);
    printf("Masukkan nilai b: ");
    scanf("%d", &b);

    if (a > b){
        printf("a lebih besar daripada b");
    }
    else if (a < b){
        printf("a lebih kecil daripada b");
    }
    else {
        printf("a sama dengan b");
    }
    return 0;
}
```

3. Apa yang dilakukan oleh kode tersebut? (10 poin)

Agar kode yang ditulis dalam main() dapat terlihat lebih sederhana, maka dapat diimplementasikan sebuah function. Function yang akan digunakan bertipe void karena di dalam program tersebut **tidak terjadi perubahan nilai pada variabel apapun**, sehingga function tidak perlu mengembalikan nilai. Coba perhatikan kode berikut!

PROGRAM 4

```
#include <stdio.h>

void compare(int a, int b){
    if (a > b){
        printf("a lebih besar daripada b");
    }
    else if (a < b){
        printf("a lebih kecil daripada b");
    }
    else {
        printf("a sama dengan b");
    }
}

int main(void){
    int a, b;

    printf("Masukkan nilai a: ");
    scanf("%d", &a);
    printf("Masukkan nilai b: ");
    scanf("%d", &b);

    compare(a, b);

    return 0;
}
```

Percabangan if yang sebelumnya ada di dalam main() dapat dipindahkan ke sebuah function yang diberi nama compare(a,b). Hal ini tentu memberikan tampilan yang lebih sederhana di dalam main() daripada program sebelumnya. Sebuah function juga dapat dipanggil berulang kali. Coba perhatikan program berikut!

PROGRAM 5

```
#include <stdio.h>

void compare(){
    int a, b;

    printf("Masukkan nilai a: ");
    scanf("%d", &a);
    printf("Masukkan nilai b: ");
    scanf("%d", &b);

    if (a > b){
        printf("a lebih besar daripada b");
    }
    else if (a < b){
        printf("a lebih kecil daripada b");
    }
    else {
        printf("a sama dengan b");
    }
}

int main(void){
    compare();
    compare();
    compare();

    return 0;
}
```

Pada program di atas, function `compare()` **tidak memiliki parameter**, dikarenakan variabel `a` dan `b` dideklarasikan di dalam function `compare()`, sehingga tidak perlu dipindahkan dari `main()` terlebih dahulu. Di dalam program tersebut dapat dilakukan perbandingan antara nilai `a` dan `b` sebanyak 3 kali, hanya dengan memanggil function yang sama sebanyak 3 kali. Dengan demikian, function dapat digunakan untuk mempersingkat penggunaan blok kode yang sama.

4. Sekarang coba buat program baru yang akan menampilkan output penjelasan tentang diri anda sendiri. Penjelasan dapat berupa nama, tempat tanggal lahir, hobi, aslab favorit, cita-cita, dsb. **Gunakan function void** untuk menampilkan teks. (10 poin)

THEORY (Gunakan Referensi) (20 poin)

5. Jelaskan apa itu function dalam bahasa C dan tujuan penggunaannya ! (10 poin)
6. Sebutkan dan jelaskan macam-macam variabel berdasarkan tempat dideklarasikannya (global variable dan local variable) ! (10 poin)

PRACTICE (40 poin)

7. Masih ingatkah kalian dengan soal post test Modul 5 “Kalkulator Sederhana” ? Sekarang coba implementasikan function untuk setiap pilihan operasi yang ada. Tampilan menu juga dapat menggunakan function void. Kode program dapat di copas.

ADDITIONAL INFORMATION (Gunakan Referensi) (10 poin)

8. Jelaskan apa itu recursive function! Coba cari salah satu contoh program dan jelaskan apa yang dilakukan oleh program tersebut. (10 poin)