

Algoritme dan Pemrograman

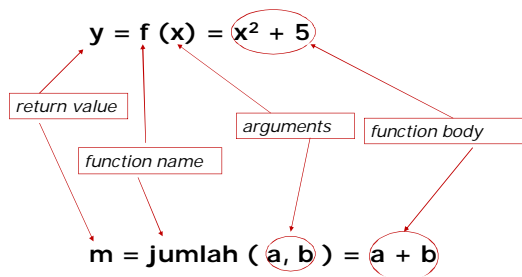
Kuliah #6 - 7
Fungsi

Fungsi

- Contoh fungsi $\rightarrow y=f(x)=x^2+5$.
- Nilai yang mungkin untuk variabel x disebut **daerah fungsi (domain)** (df), dan untuk variabel y atau $f(x)$ disebut **wilayah fungsi (range)** (wf).
- Nilai yang diberikan pada df akan menentukan nilai yang dihasilkan pada wf.
- Pada contoh tersebut:
 - f disebut nama fungsi
 - x disebut argumen fungsi
 - x^2+5 disebut tubuh fungsi (*function body*)
 - Tipe data dari y disebut *return type*.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi



DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Bentuk umum fungsi dalam C

```
return-type function-name parameter-list
{
    local definitions;
    statements;
    return retval;
}
```

- *Return type* \rightarrow tipe nilai yang dihasilkan/dikembalikan fungsi
- *Function name* \rightarrow nama fungsi
- *Parameter list* \rightarrow daftar nama variabel penampung parameter, dituliskan di dalam tanda kurung $()$, dipisahkan oleh tanda koma $(,)$
- *Local definitions* \rightarrow variabel lokal
- *Retvalue* \rightarrow nilai yang dikembalikan fungsi
- Fungsi boleh didefinisikan sebelum atau setelah fungsi main

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh: Fungsi menjumlah dua bilangan

```
#include <stdio.h>
int jumlah(int, int); //Prototipe fungsi jumlah
int main() { //main() adalah fungsi utama
    int a, b, c;
    scanf("%d %d", &a, &b); //Perhatikan bahwa scanf() juga fungsi
    c = jumlah(a, b); //Pemanggilan fungsi jumlah
    // dengan a dan b sebagai parameter
    // Dalam fungsi jumlah, a dan b masing-
    // masing masuk ke x dan y
    // Nilai kembalian fungsi disimpan di c
    printf("%d\n", c);
    return 0;
}
int jumlah(int x, int y) { // Nama fungsi: jumlah
    int t; // x dan y menampung parameter
    t = x+y; // t adalah variabel lokal
    return t; // Isi t menjadi hasil fungsi
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Function prototype

- Dituliskan sebelum fungsi main()
- Prototipe fungsi terdiri atas tiga elemen:
 - *return type*,
 - *function name*, dan
 - daftar *parameter type*.
- Berguna untuk memberitahu tipe parameter dan nilai kembalian fungsi
 - Terutama ketika deklarasi fungsi dilakukan di bawah pemanggilannya

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Variabel lokal

- Variabel yang didefinisikan dan dideklarasikan di dalam suatu fungsi hanya “hidup”/berlaku di dalam fungsi tersebut.
- Disebut sebagai **variabel lokal**.
- Variabel yang dibuat di dalam satu fungsi **tidak ada hubungannya** dengan variabel di fungsi lainnya walaupun menggunakan nama yang sama.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Variabel lokal: Apa output program berikut?

```
#include <stdio.h>
int test(int, int);
int kali(int, int);

int test(int a, int b) {
    int c;
    c = kali(a,b);
    return c;
}

int main()
{
    int a=5, b=10, c=15, d;
    d = test(a,b);
    printf("%d %d %d %d\n", a, b, c, d);
    return 0;
}
```

```
int kali (int x, int y){
    int z;
    z = x*y;
    return z;
}
```

5 10 15 50

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Void

- Jika fungsi tidak mengembalikan nilai apapun, gunakan void sebagai *return type*.
- Contoh (apa yang dilakukan?):

```
void pola(int n) {
    int i, j;
    for (i=1; i<=n; i++) {
        for (j=i; j<=n; j++)
            printf("*");
        printf("\n");
    }
}
```



DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi rekursif

- Perhatikan fungsi berikut:
jumlah(a,b) = a+b
- Jika b=0, maka jumlah(a,b)=a.
- Jika b≠0, maka
jumlah(a,b) = a+b
= 1+(a+(b-1))
= 1+jumlah(a,b-1)

$$jumlah(a,b) = \begin{cases} a & \text{utk } b=0 \\ 1 + jumlah(a,b-1) & \text{utk } b \neq 0 \end{cases}$$

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi rekursif

- Untuk b≠0, fungsi jumlah masih memanggil fungsi yang sama (dirinya sendiri).
- Fungsi yang memanggil dirinya sendiri disebut **fungsi rekursif**.
- Program yang disusun menggunakan fungsi rekursif akan lebih ringkas, tetapi kecepatan program lebih lambat karena komputer membutuhkan waktu **overhead** menyusun dalam struktur **stack**.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi rekursif: Contoh

```
int jumlah(int a, int b) {
    if (b==0)
        return a;
    else
        return 1+jumlah(a,b-1);
}
```

Telusuri fungsi tersebut jika dipanggil dengan jumlah(3,2)

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Elemen fungsi rekursif

Fungsi rekursif membutuhkan tiga elemen:

- **Terminating conditions** (kondisi berhenti), yaitu suatu kondisi untuk menentukan kapan proses rekursif akan dihentikan. Contoh: $b=0$.
- **Recursif conditions** (kondisi rekursif), yaitu suatu kondisi dimana fungsi akan terus memanggil dirinya sendiri. Contoh: $b \neq 0$.
- **Recursive expressions** (ekspresi rekursif), yaitu ekspresi yang ditulis untuk memanggil fungsi itu sendiri. Contoh: $\text{jumlah}(a,b) = 1 + \text{jumlah}(a,b-1)$.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh lain: Fungsi faktorial

- Contoh: $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$

$$faktorial(n) = \begin{cases} 1 & \text{utk } n = 0 \\ n * faktorial(n-1) & \text{utk } n \neq 0 \end{cases}$$

- Implementasi dalam C:

```
long int faktorial(long int n) {
    if (n==0)
        return 1;
    else
        return n*faktorial(n-1);
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan 1

Lengkapi program berikut dengan mendefinisikan fungsi **jumFaktor** untuk menghitung banyaknya faktor pembagi dari bilangan bulat n selain 1 dan bilangan itu sendiri. Sebagai contoh, jika diberi masukan 12 maka keluaran program adalah 4 (ada 4 bilangan yang dapat membagi 12, yaitu 2, 3, 4, dan 6). Sedangkan jika diberi masukan 7 maka keluaran program adalah 0.

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%d\n", jumFaktor(n));
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan 2

Lengkapi program berikut dengan mendefinisikan fungsi **jumDigit** untuk menghitung jumlah dari setiap digit bilangan bulat. Sebagai contoh, jika diberi masukan 125 maka keluaran program adalah 8 (dihitung dari $1+2+5$). Sedangkan jika diberi masukan 20 maka keluaran program adalah 2.

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%d\n", jumDigit(n));
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan 3

Lengkapi program berikut dengan mendefinisikan fungsi **segitiga** untuk mencetak bentuk segitiga berukuran n (bilangan bulat) seperti pada contoh. Jika diberi masukan 4 maka keluaran program adalah:

```
xxxx
xxx
xx
x

#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    segitiga(n);
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR