

## Kisi-kisi UTS-P

- Dua soal
  1. Soal yang jawabannya memerlukan *conditional* tanpa *loop*
  2. Soal yang jawabannya memerlukan *conditional* dan *loop*
- Jawaban akan dinilai sesuai persentasi keluaran yang sesuai dengan *testcase*
  - *Testcase*: pasangan masukan dan keluaran untuk pengujian program jawaban
  - *Testcase* dibuat untuk mencakup berbagai kemungkinan masukan dan keluaran

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Kisi-kisi UTS-T

- Materi UTS sampai dengan fungsi non-rekursif
- 20 soal pilihan ganda
- 5 soal isian singkat

	PG	Isian
1. Dasar program C	2	
2. Tipe data	2	1
3. Conditional	4	1
4. Loop while	2	1
5. Loop do while	2	
6. Loop for	4	1
7. Nested loop	2	
8. Fungsi	2	1

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Contoh soal

- Buatlah suatu program yang membaca sebuah bilangan bulat,  $x$ . Program akan mencetak -1 jika  $x$  adalah bilangan negatif; 0 jika  $x$  adalah 0; 1 jika  $x$  adalah bilangan positif bukan nol. *Output* diakhiri oleh *newline*.

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

No	Masukan	Keluaran
1	0	0
2	-2	-1
3	-19	-1
4	-12	-1
5	-1	-1
6	231	1
7	15	1
8	99	1
9	1214	1
10	285	1

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Berapa nilai program berikut?

```
#include <stdio.h>
#include <conio.h>
int main() {
    int x;
    scanf("%d", &x);
    if (0 == x) printf("0");
    else if (0 < x) printf("1");
    else printf("-1");
    getch();
    return 0;
}
```

0 dengan pesan  
Compile Error  
karena  
menggunakan  
conio.h

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Berapa nilai program berikut?

```
#include <stdio.h>
int main() {
    int x, y;
    scanf("%d %d", &x, &y);
    if (0 == x) printf("0");
    else if (0 < x) printf("1");
    else printf("-1");
    return 0;
}
```

0 dengan pesan Time Limit  
Exceeded karena program  
meminta masukan lebih dari yang  
diberikan sehingga program tidak  
berhenti.

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Berapa nilai program berikut?

```
#include <stdio.h>
int main() {
    int x;
    scanf("%d", &x);
    if (0 == x) printf("0");
    else printf("1");
    else printf("-1");
    return 0;
}
```

0 dengan pesan  
Wrong Answer  
karena keluaran  
tidak cocok

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Berapa nilai program berikut?

```
#include <stdio.h>
int main() {
    int x;
    scanf("%d", &x);
    if (0 == x) printf("0\n");
    else printf("1\n");
    return 0;
}
```

60 dengan pesan  
Wrong Answer  
karena ada 40%  
keluaran yang salah

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

# Algoritme dan Pemrograman

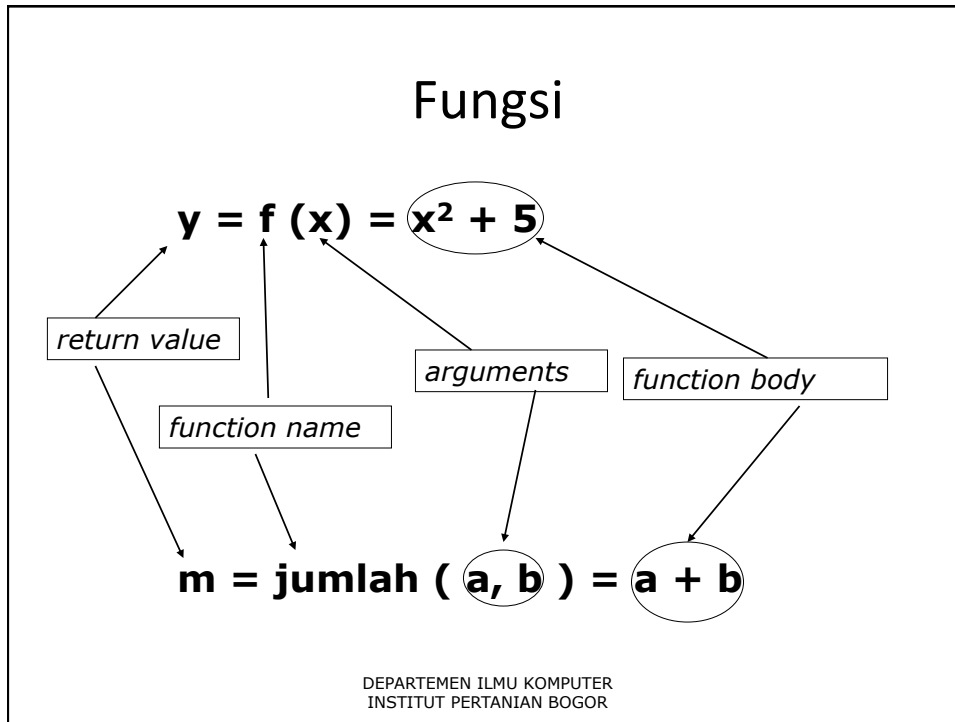
Kuliah #7

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



## Bentuk umum fungsi dalam C

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

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

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Contoh: Fungsi menjumlahkan 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:
- $$\text{jumlah}(a,b) = a+b$$
- Jika  $b=0$ , maka  $\text{jumlah}(a,b)=a$ .
  - Jika  $b \neq 0$ , maka

$$\begin{aligned} \text{jumlah}(a,b) &= a+b \\ &= 1+(a+(b-1)) \\ &= 1+\text{jumlah}(a,b-1) \end{aligned}$$

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

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Fungsi rekursif

- Untuk  $b \neq 0$ , 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

## Kekurangan Fungsi Rekursif

- Setiap pemanggilan fungsi mengharuskan prosesor menyimpan nilai variabel-variabel lokal ke *stack* di memori utama.
- Rekursi yang dalam memperlambat proses karena akses memori utama lambat dan juga menghabiskan *stack* (*stack overflow*).

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

## Perbandingan sederhana

### Non-rekursif

```
int fnr(int n) {
    int i=1, x=0;
    for(;i<=n;i++)
        x += i;
    return x;
}
```

### Rekursif

```
int fr(int n) {
    int i=1;
    if (n<=1) return 1;
    else
        return n + fr(n-1);
}
```

Coba panggil dengan berbagai nilai n (mis. 10000 dan 100000)

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>
#include <math.h>

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

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR

```
int jumFaktor (int x) {
    int i, n, jum = 0;

    for(i=2; i < x; i++)
        if (0 == x%i) jum++;

    return jum;
}
```

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

```
int jumDigit (int x) {
    int jum = 0;

    while (x) {
        jum += x%10;
        x /= 10;
    }
    return jum;
}
```

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

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

DEPARTEMEN ILMU KOMPUTER  
INSTITUT PERTANIAN BOGOR