

Fungsi Rekursif

Fungsi rekursif adalah suatu fungsi yang memanggil dirinya sendiri. Pada beberapa persoalan, fungsi rekursif sangat berguna karena mempermudah solusi. Namun demikian, fungsi rekursif juga memiliki kelemahan, yakni memungkinkan terjadinya overflow pada stack, yang berarti stack tidak lagi mampu menangani permintaan pemanggilan fungsi karena kehabisan memori (stack adalah area memori yang dipakai untuk variabel lokal untuk mengalokasikan memori ketika suatu fungsi dipanggil). Oleh karena itu, jika bisa diselesaikan dengan metode iteratif, gunakanlah metode iteratif.

Bentuk umum fungsi rekursif.

```
return_data_type function_name(parameter_list)
{
    ...
    function_name(...);
    ...
}
```

Contoh program

Contoh #1: Faktorial

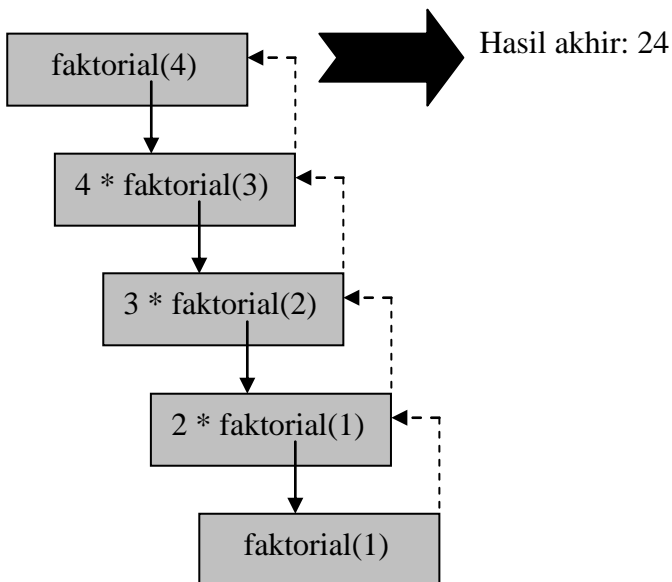
Fungsi rekursif dapat digunakan untuk menghitung faktorial. Berikut penjelasan beserta dengan contoh listing programnya.

Fungsi faktorial dapat dinyatakan dalam bentuk rekursif seperti berikut:

$fak(n) = 1$, untuk $n = 0$ atau $n = 1$

$fak(n) = n \times (n-1)!$, untuk $n > 0$

Berikut contoh proses untuk perolehan hasil 4!.



Berikut implementasi program dengan metode iteratif.

```
#include <stdio.h>

int fak(int n)
{
    int i, fak;
}
```

```
if(n == 0 || n == 1)
    return 1;
else
{
    temp = 1;
    for (i=1; i<=n; i++)
        fak = fak * i;
    return (fak);
}
void main()
{
    int fakt;
    printf("Masukkan berapa faktorial : ");
    scanf("%d",&fakt);
    printf("Hasil faktorial dari adalah : %d\n", fak(fakt));
}
```

Berikut implementasi program untuk proses faktorial dengan metode rekursif:

```
#include <stdio.h>

int fak( int n )
{
    if (n == 0 || n == 1)
        return 1;
    else
        return n * fak(n-1);
}

void main()
{
    int n, hasil;
    printf("Masukkan suatu bilangan bulat : ");
    scanf("%d", &n);
    hasil = fak(n);
    printf("%d! = %d", n, hasil);
}
```

Sekarang cobalah ketikkan program di atas dan masukkan beberapa bilangan untuk mengujinya! Manakah yang memiliki kecepatan yang lebih baik: pendekatan rekursif atau pendekatan iteratif?

Contoh #2: Fibonacci

Fungsi rekursif juga dapat digunakan untuk menyelesaikan bilangan Fibonacci. Fungsi Fibonacci dapat dinyatakan dalam bentuk rekursif seperti berikut:

$\text{fib}(n) = 0$, untuk $n = 0$

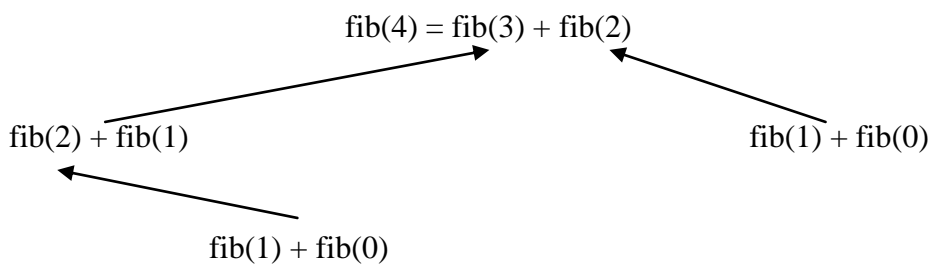
$\text{fib}(n) = 1$, untuk $n = 1$

$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$, untuk $n > 1$

Berikut contoh hubungan antara n dan hasil fungsi:

n	fib(n)
0	0
1	1
2	1
3	2
4	3
5	5
6	8
...	...

Berikut contoh proses pemanggilan fib(4)



Implementasi program dengan metode iteratif

```
#include <stdio.h>

int fib(int n)
{
    int f1 = 0, f2 = 1, fibo;
    if(n == 0)
        return 0;
    else if(n == 1)
        return 1;
    else
    {
        for(int i = 0; i < n; i++)
        {
            fibo = f1 + f2;
            f2 = f1;
            f1 = fibo;
        }
        return fibo;
    }
}

void main()
{
    int n, hasil;
    printf("Bilangan Fibonacci ke-");
    scanf("%d", &n);
    hasil = fib(n);
    printf("fib(%d) = %d", n, hasil);
}
```

Implementasi program dengan metode rekursif.

```
#include <stdio.h>

int fib(int n)
{
    if(n == 0)
        return 0;
    else if(n == 1)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}

void main()
{
    int n, hasil;
    printf("Bilangan Fibonacci ke-");
    scanf("%d", &n);
    hasil = fib(n);
    printf("fib(%d) = %d", n, hasil);
}
```

Cobalah ketikkan implementasi program dari metode rekursif di atas dan jalankan program tersebut! Uji dengan memasukkan beberapa bilangan! Selanjutnya, kembangkan program tersebut sehingga dapat untuk menampilkan n buah bilangan fibonacci!

Contoh #3: Menara Hanoi

Menara Hanoi adalah persoalan untuk memindahkan tumpukan piring dari suatu tonggak ke tonggak lain dengan bantuan sebuah tonggak perantara. Penyelesaian secara rekursif untuk persoalan ini untuk n buah piring:

1. Pindahkan n-1 piring teratas pada tonggak A ke tonggak B, dengan menggunakan tonggak C sebagai perantara.
2. Pindahkan 1 piring tersisa pada tonggak A ke tonggak C.
3. Pindahkan n-1 piring teratas pada tonggak B ke tonggak C, dengan menggunakan tonggak A sebagai perantara.

Berikut implementasi program dengan metode rekursif.

```
#include <stdio.h>
void tonggak(int n, char a, char b, char c)
{
    if(n == 1)
        printf("Pindahkan piring dari %c ke %c\n", a, c);
    else
    {
        tonggak(n-1, a, c, b);
        tonggak(1, a, b, c);
        tonggak(n-1, b, a, c);
    }
}

void main()
{
    int jml_piring;
    printf("Jumlah piringan: ");
    scanf("%d", &jml_piring);
    tonggak(jml_piring, 'A', 'B', 'C');
}
```

Latihan

Berikut ini diberikan contoh program untuk menghitung pangkat sebuah bilangan dengan menggunakan metode iteratif.

```
#include <stdio.h>

int pangkat (int a,int b)
{
    int i, bil = a;
    if(b==1)
        return a;
    else
    {
        for (i=2;i<=b;i++)
            a = a * bil;
        return a;
    }
}

void main()
{
    int x,y,hasil;
    printf("masukan bilangan:");
    scanf("%i",&x);
    printf("masukan pangkat:");
    scanf("%i",&y);
    hasil = pangkat (x,y);
    printf("%i",hasil);
}
```

Ketikkan program di atas dan jalankan!Ujilah dengan memasukkan beberapa bilangan. Kemudian ubahlah program tersebut sehingga program tersebut menggunakan metode rekursif!

Soal

1. Buatlah program untuk menghitung FPB(faktor persekutuan terbesar) dari 2 bilangan dengan menggunakan metode rekursif! Untuk membantu, berikut diberikan penyelesaian secara rekursif dari FPB :

$fpb(x,y) = y$ jika $y \leq x$ dan $sisa_pembagian(x,y) = 0$

$fpb(x,y) = fpb(y,x)$ jika $x < y$

$fpb(x,y) = fpb(y, sisa_pembagian(x,y))$ untuk keadaan yang lain

2. Buatlah suatu fungsi untuk membalik suatu bilangan dengan cara rekursif. Sebagai contoh, bilangan 1238 ditampilkan menjadi 8321!