

Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Esa Unggul
2019



Struktur Data

Fungsi Rekursif

M. Bahrul Ulum, M.Kom

PENGERTIAN FUNGSI REKURSIF

Teori

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 variable 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

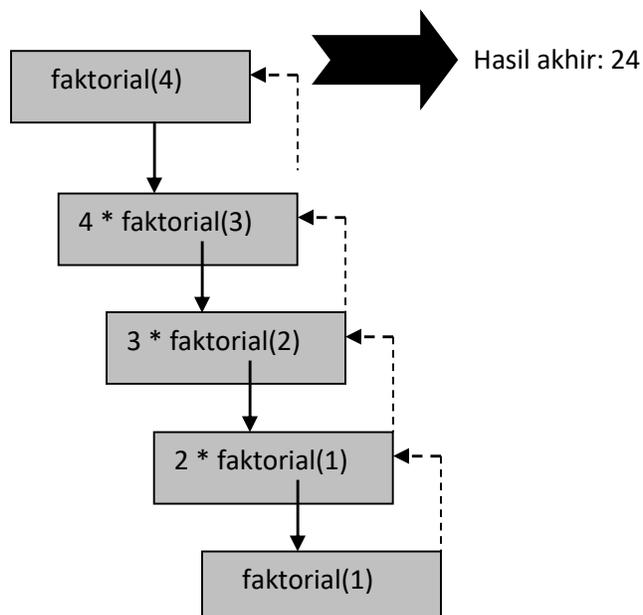
Fungsi rekursif dapat digunakan untuk menghitung faktorial. Berikut penjelasan beserta dengan contoh listing programnya.

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);
}

```

Contoh program *Faktorial*

```

1  #include <iostream>
2  using namespace std;
3  int factorial (int n){
4      if(n==1){
5          return 1;
6      } else {
7          return n*factorial(n-1);
8      }
9  }
10
11 int main(){
12     int bil;
13
14     cout<<"Program Faktorial dengan Rekursif"<<endl;
15     cout<<"======"<<endl;
16     cout<<endl;
17     cout<<"Masukan Sebuah Bilangan Bulat [Integer]"<<endl;
18     cout<<"bilangan = ";cin>>bil;
19     cout<<factorial(bil);
20
21 }

```


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 = fibo(n);
    printf("fib(%d) = %d", n, hasil);
}

```

Contoh program

```

1  #include <iostream>
2  #include <conio.h>
3  using namespace std;
4
5  int fibo(int i)
6  {
7      if(i==0)
8          return 0;
9      else if(i==1)
10         return 1;
11     else
12         //fungsi rekursif
13         return fibo(i-1)+fibo(i-2);
14 }
15
16 int main()
17 {
18     int bil;
19
20     cout<<"Fungsi Rekursif Bilangan Fibonacci"<<endl<<endl;
21     cout<<"Masukkan bilangan : ";
22     cin>>bil;
23     //pemanggilan fungsi
24     cout<<"Fibonacci("<<bil<<" ) = "<<fibo(bil);
25
26     getch();
27 }

```

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!

Kombinasi

```
input n, r : integer
If (n < r) Then
    return (0)
Else
    return (Faktorial(n)/Faktorial(r)*Faktorial(n-r))
Endif
```

Permutasi

```
input n, r : integer

    If (n < r) Then
        return (0)
    Else
        return (Faktorial(n) / Faktorial(n-r))
    Endif
```

Studi Kasus-1:

Output deret $\rightarrow S = 1+2+3+4+5+\dots+n$

```
input n:integer

    If (n==1) Then
        return (n)
    Else
        return (n + S(n-1))
    Endif
```

Studi Kasus-2

Output deret $S = 2+4+6+8+10+\dots+2n$

```
input n:integer

    If (n==1) Then
        return (2)
    Else
        return (2*n + S(n-1))
    Endif
```

Kesimpulan

Fungsi rekursif merupakan fungsi yang memanggil dirinya sendiri. Terdapat dua komponen penting dalam fungsi rekursif, yaitu kondisi kapan berhentinya fungsi dan pengurangan atau pembagian data ketika fungsi memanggil dirinya sendiri. Optimasi fungsi rekursif dapat dilakukan dengan menggunakan teknik tail call, meskipun teknik ini tidak selalu diimplementasikan oleh semua bahasa pemrograman. Selain sebagai fungsi, konsep rekursif juga terkadang digunakan untuk struktur data seperti binary tree atau list.

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 :

$\text{fpb}(x,y) = y$ jika $y \leq x$ dan $\text{sis_pembagian}(x,y) = 0$

$\text{fpb}(x,y) = \text{fpb}(y,x)$ jika $x < y$

$\text{fpb}(x,y) = \text{fpb}(y, \text{sis_pembagian}(x,y))$ untuk keadaan yang lain

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