

Pertemuan 10



Fungsi Rekursi, Contoh program Fungsi dan Rekursi

Mata Kuliah : Algoritma & Pemrograman
Dosen : Tessy Badriyah, SKom., MT., PhD.



Tujuan Pembelajaran

- Memahami konsep rekursi
- Berpikir secara rekursif
- Menyelesaikan persoalan pemrograman yang membutuhkan cara rekursif



Konsep Rekursi

- Rekursi adalah program yang memanggil dirinya sendiri
- Contoh persoalan rekursi : penyelesaian permasalahan faktorial
- Penyelesaian faktorial dapat dituliskan sbb :
$$n! = (n)(n-1)(n-2) \dots (1)$$
- Dapat juga dinyatakan dengan :

$$F(n) = \begin{cases} 1 & \text{jika } n=0, n=1 \\ nF(n-1) & \text{jika } n>1 \end{cases}$$



Syarat Rekursi

- Syarat yang harus dipenuhi agar suatu permasalahan dapat diselesaikan secara rekursif:

1. Permasalahan dapat dipecah menjadi lebih sederhana.

Seperti dicontohkan untuk permasalahan faktorial pada slide sebelumnya, $4! = 3! * 4$.

2. Harus ada kondisi yang mengakhiri proses rekursi. Disebut dengan terminate condition
3. Ada bagian program yang melakukan pemanggilan terhadap dirinya sendiri.



Persoalan Fibonacci secara Rekursi


- Deret fibonaci yang mempunyai nilai suku-suku bilangan sebagai berikut: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...
- Cara rekursif untuk permasalahan deret fibonacci, secara umum dapat kita detailkan sebagai berikut

$$F(n) = \begin{cases} n & \text{jika } n=0, n=1 \\ F(n-2) + F(n-1) & \text{jika } n>1 \end{cases}$$



Latihan 1 : Fungsi Rekursi tanpa akhir

```
#include <stdio.h>
void Tidak_Berhenti();
main()
{
    Tidak_Berhenti();
}
void Tidak_Berhenti()
{
    printf("Ctrl-Break untuk berhenti.\n");
    Tidak_Berhenti();
}
```



Latihan 2 : Fungsi Rekursi dengan batas akhir

```
#include <stdio.h>
void Berhenti_N_Kali(int n);
main()
{
    int N=3;
    Berhenti_N_Kali(N);
}
void Berhenti_N_Kali(int n)
{
    static int i=0;
    if (n<=0) return;
    printf("%d kali\n",++i);
    Berhenti_N_Kali(n-1);
}
```

Latihan 3 : Faktorial tanpa rekursi

- ```
#include <stdio.h>
int fact_it (int n)
{
 int i,fak=1;
 /* hitung faktorial */
 for (i=1; i<=n; i++)
 { fak = fak * i; }
 return (fak);
}
void main()
{
 int fac;
 printf("Masukkan berapa faktorial : ");
 scanf("%d",&fac);
 printf("Hasil faktorial dari %d adalah : %d ", fac, fact_it(fac));
 printf('\n');
```





## Latihan 4 : Faktorial dengan Rekursi

```
#include <stdio.h>
int fact_rec(int n)
{
 /* Menghitung faktorial dengan rekursi */
 if (n < 0)
 return 0;
 else if (n == 0 || n == 1)
 return 1;
 else
 return n * fact_rec(n-1);
}
void main()
{
 int fac;
 printf("Masukkan berapa faktorial : ");
 scanf("%d",&fac);
 printf("Hasil faktorial dari adalah : %d ", fact_rec(fac));
 printf('\n');
}
```



## Latihan 5 : Deret Fibonacci dengan rekursi

```
#include <stdio.h>
int Fibonaci(int N);
void main()
{
 int i, N;
 printf("Masukkan batas akhir dari bilangan fibonaci : ");
 scanf("%d",&N);
 for (i=0; i<=N; i++)
 printf("%d, ",Fibonaci(i));
}
int Fibonaci(int N)
{
 if (N<2) return (N);
 else
 return(Fibonaci(N-2)+Fibonaci(N-1));
}
```



## Tugas 4

Kerjakan 4 latihan soal berikut, kemudian submit ke elearning

1. Buatlah sebuah fungsi yang menulis angka dari  $n$  ke 0 dengan menggunakan proses rekursi.
2. Tuliskan sebuah fungsi untuk menulis angka dari 0 ke  $n$  dengan menggunakan proses rekursi.



## Tugas 4

3. Tuliskan sebuah fungsi rekursi yang melakukan pengecekan apakah sebuah elemen  $X$  merupakan anggota dari sebuah array  $a[n]$
4. Tuliskan fungsi rekursi untuk membalik suatu kalimat. Sebagai contoh, kalimat 'Struktur Data' dibalik menjadi 'ataD rutkurtS'.



## Referensi

- Robertson, Lesley Anne. (1992). *Students' guide to program design*. Oxford : Newnes
- Santner, Williams, and Notz (2003), *Design and Analysis of Computer Experiments*, Springer.
- Deitel & Deitel, *C How to Program*, Prentice Hall 1994 (2<sup>nd</sup> edition)
- Brookshear, J.G., *Computer Science: An Overview*, Benjamin-Cummings 2000 (6<sup>th</sup> edition)
- Kernighan & Ritchie, *The C Programming Language*, Prentice Hall