PRAKTIKUM 8

REKURSIF TAIL

A. TUJUAN PEMBELAJARAN

1. Memahami mengenai konsep rekursif tail
2. Mampu memecahkan permasalahan dengan konsep rekursif tail

B. DASAR TEORI

Rekursif adalah suatu proses atau prosedur dari fungsi yang memanggil dirinya sendiri secara berulang-ulang. Karena proses dalam Rekursif ini terjadi secara berulang-ulang maka harus ada kondisi yang membatasi pengulangan persebut, jika tidak maka proses tidak akan pernah berhenti sampai memori yang digunakan untuk menampung proses tersebut tidak dapat menampung lagi/penuh.

Terdapat dua metode rekursif yaitu Tail recursive dan Nontail recursive. Nontail recursive adalah proses rekursif yang sudah kita bahas sebelumnya. Rekursif Tail adalah proses rekursif dengan pemanggilan rekursif di akhir method dan tidak memiliki aktivitas selama fase balik. Method rekursif yang bukan tail recursive disebut non-tail recursive. Contoh dari tail rekursif adalah

```
void tail(int i) {
      if (i>0) {
         system.out.print(i+"")
      tail(i-1)
}
```

Yang bukan merupakan tail rekursif adalah:

```
int fact(int x){
   if (x==0)
      return 1;
   else
```

```
      return x*fact(x-1);
}
```

Bukan tail recursive karena pada saat kembali dari recursive call x*fact(x-1), masih terdapat operasi perkalian.

```
void non prog(int i) {
   if (i>0) {
         prog(i-1);
         System.out.print(i+"");
         prog(i-1);
      }
}
```

Bukan tail rekursif, karena dibaris awal terdapat recursive call. Pada tail recursive, recursive call menjadi statement akhir, dan tidak ada recursive call diatasnya.

Proses mengubah dari Non Tail ke Tail Rekursif

Method non-tail recursive dapat diubah menjadi tail-recursive method dengan menambahkan parameter tambahan untuk menampung hasil.

`method fact(int n)→ fact_aux(int n, int result)`

Teknik yang digunakan biasanya membuat fungsi tambahan (`method fact(int n)`) yang memanggil method tail recursive.

```
int fact_aux(int n, int result) {
   if (n == 1)
      return result;
   return fact_aux(n - 1, n * result)
}

int fact(n) {
   return fact_aux(n, 1);
}
```

Method tail-recursive Fibonacci diimplementasikan menggunakan dua parameter bantuan untuk menampung hasil.

```
int fib_aux ( int n , int next, int result)
{
  if (n == 0)
    return result;
  return fib_aux(n - 1, next + result, next);
}
To calculate fib(n) , call fib_aux(n,1,0)
```

C. TUGAS PENDAHULUAN

Jawablah pertanyaan berikut ini :

1. Apa yang dimaksud dengan rekursif tail ?
2. Tuliskan fungsi untuk menghitung deret fibonanci menggunakan tail rekursif !

D. PERCOBAAN

Percobaan 1 : Fungsi tail rekursif untuk menampilkan i

```
void tail(int i) {
      if (i>0) {
         system.out.print(i+"")
      tail(i-1)
}
```

Percobaan 2 : Tail rekursif untuk menghitung faktorial

```
int fact_aux(int n, int result) {
   if (n == 1)
      return result;
   return fact_aux(n - 1, n * result)
}

int fact(n) {
   return fact_aux(n, 1);
}
```

Percobaan 3 : Tail rekursif untuk menghitung Fibonacci

```
int fib_aux ( int n , int next, int result)
{
```

```
  if (n == 0)
    return result;
  return fib_aux(n - 1, next + result, next);
}
int fib(int n){
   fib_aux(n,1,0);
}
```

E. LATIHAN

1. Ubahlah proses rekursif non tail menjadi tail rekursif untuk program dibawah ini !

```
public static void decToBin(int num)
{
      if (num > 0)
      { decToBin(num / 2);
      System.out.print(num % 2);
      }
}
```

2. Ubahlah proses rekursif non tail menjadi tail rekursif untuk program dibawah ini !

```
public static boolean gcdlike(int p, int q) {
   if (q == 0) return (p == 1);
   return gcdlike(q, p % q);
}
```

3. Ubahlah proses rekursif non tail menjadi tail rekursif untuk program dibawah ini !

```
public static int f(int n) {
   if (n == 0) return 0;
   if (n == 1) return 1;
   if (n == 2) return 1;
   return 2*f(n-2) + f(n-3);
```

4. Ubahlah proses rekursif non tail menjadi tail rekursif untuk program dibawah ini dengan memanggil `square(5), cube(5), cube(123)`?

```
public static int square(int n) {
   if (n == 0) return 0;
   return square(n-1) + 2*n - 1;
}
```

F. LAPORAN RESMI

Kerjakan hasil percobaan(D) dan latihan(E) di atas dan tambahkan analisa.