**STRUKTUR DATA**

**PRAKTIKUM 2 REKURSIF**



**D isusun oleh:**

Ridho Riyadi Hartomo

(E3190731)

**D osen Pengampu:**

Mudafiq Riyan Pratama, S.Kom., M.Kom NIP: 19890509 201903 1 010

**PROGRAM STUDI MANAJEMEN INFORMATIKA JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI JEMBER 2020**

**PRAKTIKUM 2 REKURSIF**

|  |  |
| --- | --- |
|  | **Overview** |
|  |
| Bab ini menjelaskan mengenai teori dan implementasi rekursif. Rekursif dalam pemrograman merupakan fungsi / method yang memanggil fungsi / method itu sendiri, sehingga terjadi proses perulangan. Dalam bab ini dibahas perbedaan perulangan biasa dengan perulangan secara rekursif. Juga dibahas studi kasus yang dapat diselesaikan dengan cara rekursif. | |

# TUJUAN PEMBELAJARAN:

* 1. Mengetahui pengertian rekursif dan fungsinya.
  2. Mengetaui perbedaan perulangan biasa dengan rekursif.
  3. Mampu mengaplikasikan konsep rekursif dalam bahasa pemrograman Java.
  4. Mampu menerapkan implementasi rekursif pada beberapa studi kasus.

# DASAR TEORI

* 1. **DEFINISI REKURSIF**

Rekursif dalam pemrograman merupakan sebuah fungsi / method yang memanggil fungsi / method itu sendiri, sehingga terjadi proses perulangan. Oleh karena melakukan proses perulangan, maka perlu adanya *stopping role* atau kondisi yang dapat menghentikan proses perulangan tersebut.

Kode berikut adalah contoh sederhana fungsi rekursif:



Contoh kode di atas menunjukkan pemanggilan fungsi **printBerulang()** pada fungsi itu sendiri, ketika fungsi itu dipanggil / dijalankan, maka akan terjadi proses perulangan. Perulangan yang terjadi tersebut tidak akan berhenti karena pada contoh kode di atas tidak disediakan kondisi *stopping role* nya.

Tidak semua permasalahan komputasi, dapat diselesaikan dengan program rekursif. Contohnya pada kode program diatas, jika hanya menampilkan kalimat secara berulang, tidak perlu menggunakan rekursif. Program rekursif dipakai untuk permasalahan yang memiliki langkah penyelesaian yang terpola atau langkah-langkah yang teratur.

# KELEBIHAN REKURSIF

Berikut kelebihan-kelebihan menggunakan rekursif:

* + - Sintaks program lebih singkat.
    - Pada beberapa kasus, lebih mudah menggunakan fungsi rekursif, contohnya: penghitungan pangkat, faktorial, dan menentukan deret fibonacci.
    - Lebih efisien dan cepat dibandingkan proses perulangan secara iteratif.

# KEKURANGAN REKURSIF

Meskipun bisa dikatakan bahwa menggunakan rekursif itu lebih mempersingkat kode program, akan tetapi ada beberapa kekurangan dari rekursif ini, yaitu:

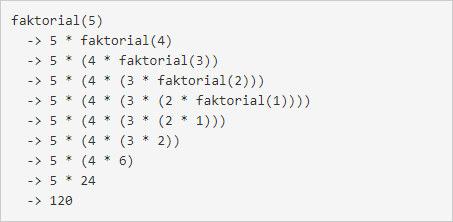
* + - Memakan memori lebih besar, karena setiap bagian dari dirinya dipanggil, maka akan membutuhkan sejumlah ruang memori untuk menampung / menyimpannya.
    - Rekursif yang tidak disediakan *stopping role*, dapat mengakibatkan memori terpakai habis.

# CARA MEMECAHKAN MASALAH DENGAN REKURSIF

1. Tentukan *stopping role* (kondisi penyetopannya), saat pemanggilan fungsinya tidak diperlukan lagi.
2. Terapkan suatu langkah untuk menggiring kasus kompleks ke kasus penyetopan dengan metode yang mencerminkan fungsi.

# CONTOH KASUS YANG DAPAT DISELESAIKAN DENGAN REKURSIF

1. Faktorial

Fungsi faktorial memiliki pola yang teratur, yaitu proses mengalikan secara terurut dari angka faktorial sampai dikalikan dengan angka 1. Dari sini bisa kita tentukan, bahwa permasalahan ini memiliki *stopping role* yaitu pada kondisi angka 1.

Jika diterapkan pada fungsi, maka:

faktorial(n)

if(n == 1)

return n

else

return n \* faktoril(n-1)

1. Hitung Perkalian

Proses perkalian pada dasarnya adalah penjumlahan yang berulang. Misalnya: perkalian 3 dan 4 maka 3 dijumlahkan dengan angka itu sendiri sebanyak 4 kali. Sehingga 3 + 3 + 3 + 3 = 12. Jika dirincikan adalah sebagai berikut:

kali(3, 4)

-> 3 + kali(3, 3)

-> 3 + (3 + kali(3, 2))

-> 3 + (3 + (3 + kali(3, 1)))

-> 3 + (3 + (3 + 3))

-> 3 + (3 + 6)

-> 3 + 9

-> 12

Jika diterapkan pada fungsi, maka:

kali(a, b)

if(b == 1)

return a

else

return a \* kali(a, b-1)

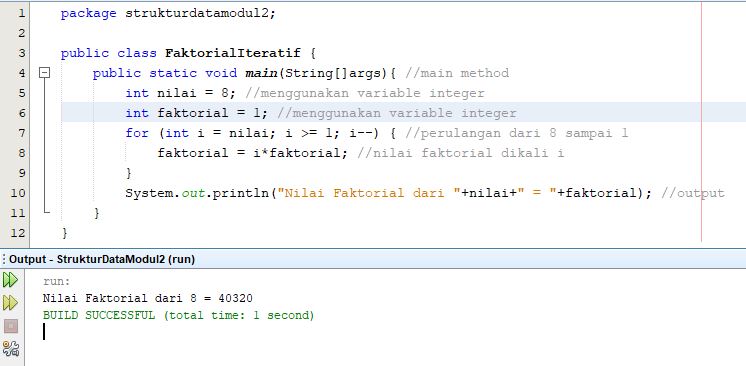
# LATIHAN PRAKTIKUM

Sebelum melakukan praktikum di laboratorium, berikut adalah alat dan bahan yang digunakan dalam praktikum modul ini, yaitu:

* 1. Modul Praktikum
  2. LCD & Komputer
  3. Netbeans
  4. Office Word
* **Buatlah project di Netbeans dengan format nama: Modul2\_NIM\_Kelas. Contoh: Modul2\_E31190001\_A**
* **Ikuti format penamaan file pada praktikum ini.**
* **Harap ditanyakan kepada dosen pengampu jika mengalami kesulitan.**
* **Setiap selesai melaksanakan latihan dan tugas, diharuskan menjalankan kode program dengan cara Run File (Shift + F6).**
* **Setiap percobaan diharuskan melakukan analisis dan memberi penjelasan secara detail terkait kode program yang ditulis.**

1. FAKTORIAL

Berikut ini adalah kode penghitungan **faktorial secara iteratif**. Tuliskan kode berikut ini:

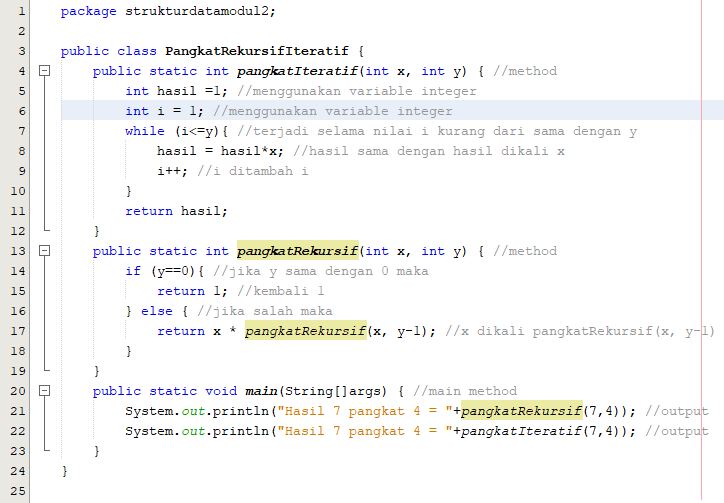


Kemudian buatlah file baru, dan tuliskan kode berikut untuk perhitungan **faktorial secara rekursif**.

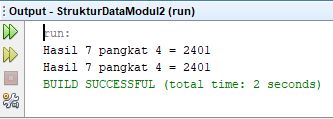


1. PANGKAT

Tuliskan kode berikut ini untuk penghitungan **pangkat secara iteratif dan rekursif**.

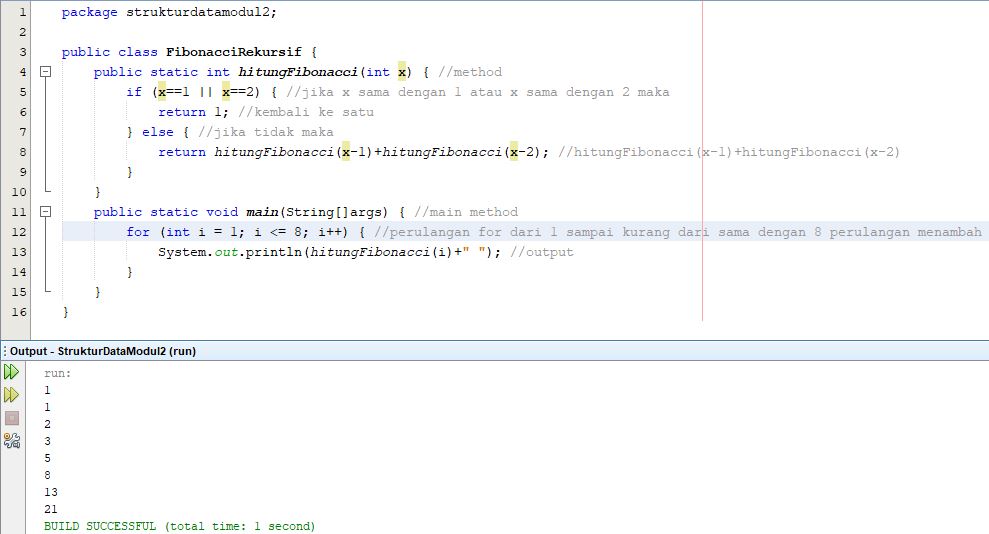


Output



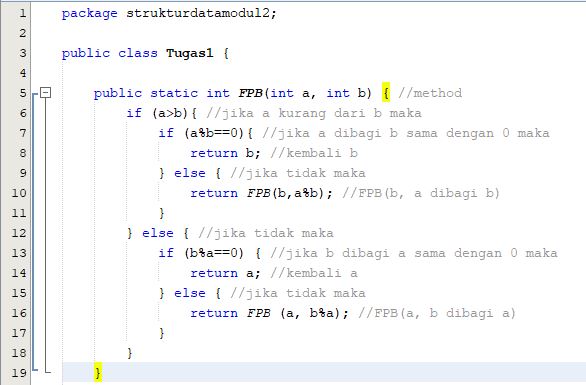
1. DERET FIBONACCI

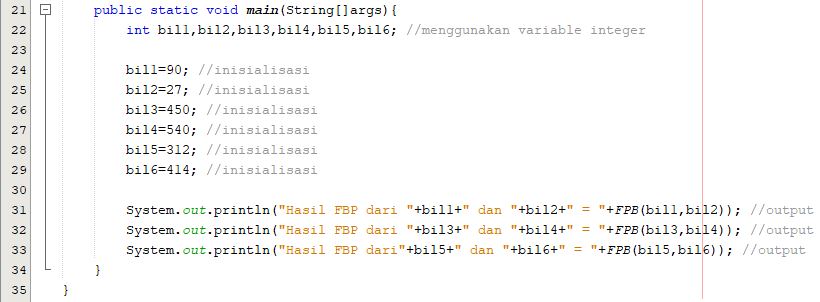
Tuliskan kode berikut ini untuk menampilkan **deret fibonacci secara rekursif**.



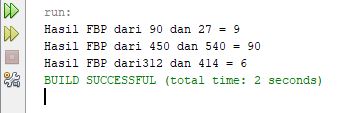
# TUGAS PRAKTIKUM

1. Buatlah sebuah program dengan fungsi rekursif untuk penghitungan FPB (Faktor Persekutuan terBesar). Misalkan: FPB dari 228 dan 90 adalah 6. Sehingga ketika dijalankan, hasilnya adalah seperti berikut:





Output



Jika sudah selesai, tampilkan hasil penghitungan FPB berikut melalui program tersebut:

* + FPB (90, 27) = 9
  + FPB (450, 540) = 90
  + FPB (312, 414) = 6

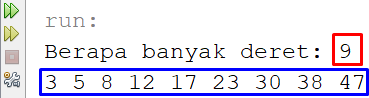
1. Buatlah sebuah program untuk menghitung rumus berikut secara rekursif: (a+b)y

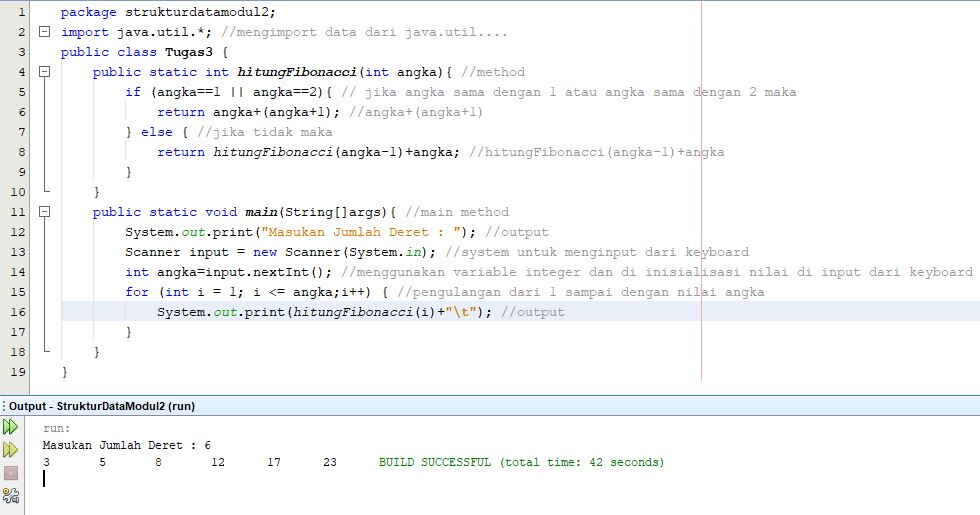
Misalkan: (2+3)4 = 54 = 625



1. Buatlah program yang dapat menampilkan deret angka dengan pola seperti berikut: 3 5 8 12 17 23

menggunakan rekursif, dan buatlah program tersebut menggunakan inputan user. Sehingga contoh input-output nya sebagai berikut:

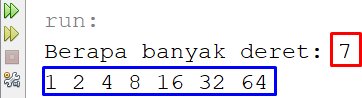


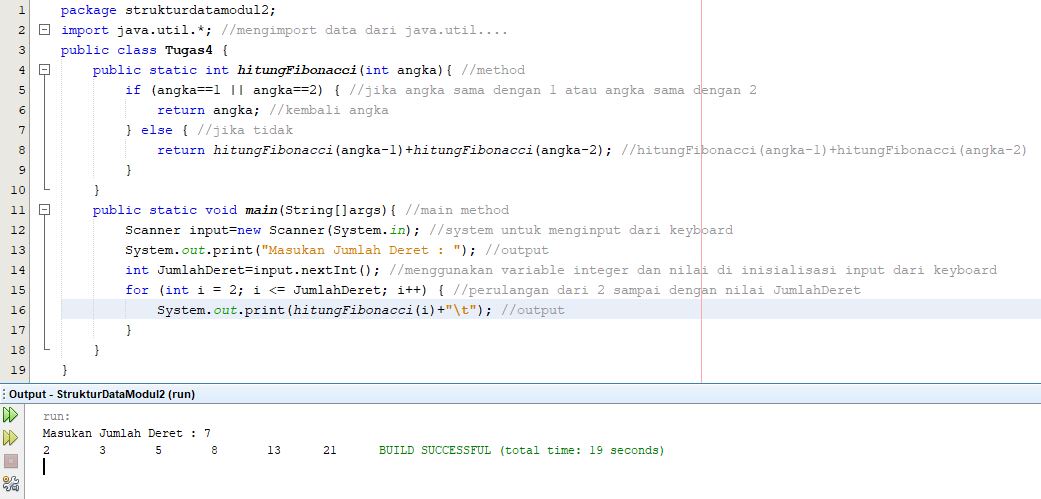


*Keterangan: kotak merah merupakan inputan user, sedangkan kotak biru merupakan hasil outputnya. Jika inputan 9, maka akan menghasilkan 6 deret angka dengan pola yang sama*

1. Buatlah program yang dapat menampilkan deret angka dengan pola seperti berikut: 1 2 4 8 16 32 64

menggunakan rekursif dan gunakan inputan user dalam program tersebut. Sehingga contoh input-outpunya seperti berikut:

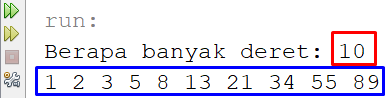


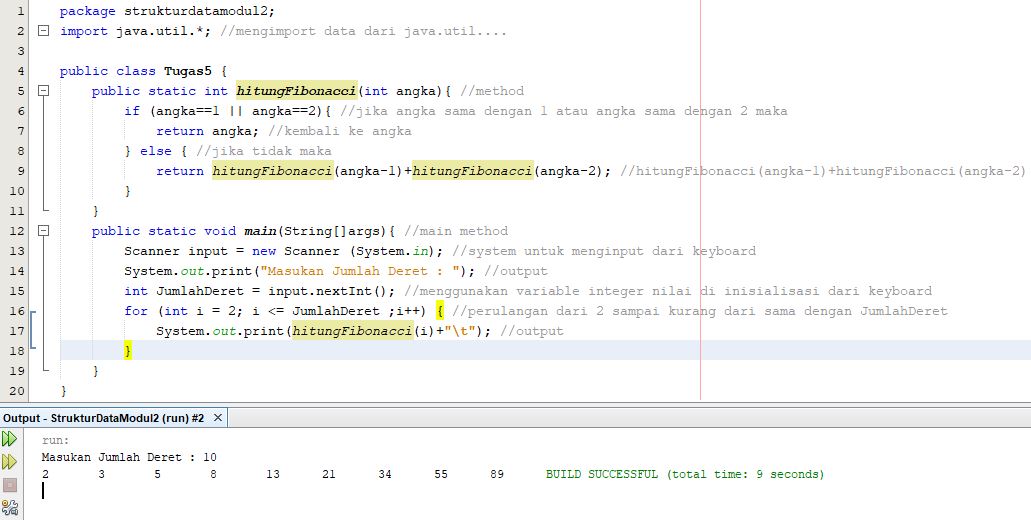


*Keterangan: kotak merah merupakan inputan user, sedangkan kotak biru merupakan hasil outputnya. Jika inputan 7, maka akan menghasilkan 7 deret angka dengan pola yang sama.*

1. Buatlah program yang dapat menampilkan deret angka dengan pola seperti berikut: 1 2 3 5 8 13 21 34 55 89

menggunakan rekursif dan gunakan inputan user dalam program tersebut. Sehingga contoh input-outpunya seperti berikut:





*Keterangan: kotak merah merupakan inputan user, sedangkan kotak biru merupakan hasil outputnya. Jika inputan 10, maka akan menghasilkan 10 deret angka dengan pola yang sama.*

1. **KESIMPULAN**

Salah satu konsep fungsi pada matematika yang ditemui pada ilmu komputer adalah fungsi rekursif: sebuah fungsi yang memanggil dirinya sendiri.

Fungsi rekursif selalu memiliki kondisi yang menyatakan kapan fungsi tersebut berhenti. Kondisi ini harus dapat dibuktikan akan tercapai, karena jika tidak tercapai maka kita tidak dapat membuktikan bahwa fungsi akan berhenti, yang berarti algoritma kita tidak benar.