

Dasar-Dasar Pemrograman 2

Tutorial 4 Kelas C & F

Recursion

Selasa, 5 Maret 2019 - 16:00 WIB



FAKULTAS
ILMU
KOMPUTER

Recursion

Pada tutorial kali ini, Anda akan belajar suatu konsep di dalam pemrograman yang disebut dengan **recursive method**. Sebuah *method*/fungsi dapat dikatakan **rekursif** jika fungsi tersebut dapat memanggil (meng-*invoke*) dirinya sendiri baik secara langsung maupun tidak langsung. Biasanya, fungsi rekursif digunakan untuk menyelesaikan permasalahan-permasalahan yang dapat direduksi menjadi masalah serupa yang lebih kecil. Selain itu, penggunaan teknik rekursi juga dapat membuat kode menjadi lebih ringkas, elegan, dan simpel, ketika digunakan dengan baik.

Dalam fungsi rekursif, terdapat dua komponen utama yang saling berkaitan :

- **Base case (kasus dasar atau kasus penyetop)**
Yaitu permasalahan terkecil yang bisa diselesaikan dengan mudah tanpa melibatkan pemecahan masalah menjadi masalah yang lebih kecil lagi. Dalam fungsi rekursif, *base case* tidak melakukan sebuah pemanggilan rekursif lagi (*recursive call*), melainkan bertujuan untuk **memberhentikan proses rekursi**.
- **Recursive case**
Yaitu sebuah proses yang melakukan pemecahan **permasalahan lebih besar** menjadi **permasalahan kecil** yang **identik**. Pastikan proses ini akan mengarah kepada *base case*.

Supaya kamu mendapat gambaran lebih jelas mengenai rekursi, perhatikan contoh fungsi perhitungan faktorial sebuah bilangan di bawah ini.

```
static int factorial(int number) {
    if (number <= 1) {
        return 1; // Base case
    } else {
        return number * factorial(number - 1); // Recursive case
    }
}
```

Pada fungsi di atas, *base case* dari fungsi tersebut adalah ketika angka pada parameter **number**

Feel free to use, reuse, and share this work: the more we share, the more we have!



kurang dari atau sama dengan 1 ($1! = 1$ dan $0! = 1$, diasumsikan juga **number** tidak negatif). *Base case* ini tidak melakukan *recursive call* sama sekali karena bertujuan untuk menghentikan proses rekursi.

Kemudian, proses rekursif pada fungsi di atas akan melakukan komputasi dengan mengembalikan nilai **number** itu sendiri dan dikalikan dengan pemanggilan kembali fungsi **factorial** dengan parameter yang berubah (perhatikan bahwa parameter mengecil dan mengarah ke *base case*).

Misalnya, pemanggilan **factorial(3)** akan mengembalikan nilai 3, kemudian dikali dengan **factorial(2)**, yang akan mengembalikan nilai 2 dikali dengan **factorial(1)**, dan akhirnya berakhir ke *base case*, proses rekursi berhenti, dan akhirnya fungsi mengembalikan nilai $3 * 2 * 1 = 6$.

Untuk diingat:

Hati-hati ketika mendefinisikan fungsi rekursif, jangan sampai terjadi rekursi yang tak hingga (*infinite recursion*) yang akan menyebabkan `StackOverflowError`. Untuk menghindarinya, pastikan ada *base case*, dan setiap langkah rekursif selalu mengarah ke *base case*.

Kenangan Masa Lalu



Source: [https://en.wikipedia.org/wiki/Night_\(hieroglyph\)#/media/File:Hieroglyphe_karnak.jpg](https://en.wikipedia.org/wiki/Night_(hieroglyph)#/media/File:Hieroglyphe_karnak.jpg)

License: CC BY-SA 3.0

Kamu tahu enggak? Dulu Dek Depi sebenarnya pernah ingin menjadi arkeolog lho! Sebenarnya keinginan ini datang dari melihat kakak-kakak Dek Depi yang pernah turun langsung ke sebuah gua kuno beramai-ramai dengan kerja sama yang sangat baik dan memecahkan teka teki dari *hieroglyph* yang ditemukan di gua tersebut.

Dek Depi sangat kagum dengan hal itu, dan ingin mencoba untuk terjun di bidang arkeolog juga. Namun, kemampuan Dek Depi tidak seberapa. Terlebih lagi, Dek Depi tidak tau apa itu *hieroglyph*, yang Dek Depi tau adalah *Relief*. Jadi, dengan inisiatif dan motivasi yang tinggi, Dek Depi membuat suatu program untuk men-*generate* suatu *Relief*. Apa arti dari gambar *Relief* yang dibuat Dek Depi? Hanya Dek Depi yang tahu.

Hal tersebut memang sudah terjadi lama. Tetapi, baru - baru ini Dek Depi ingin terjun kembali ke bidang arkeolog! Dengan tahap pertama membuat *Relief Generator* yang pernah dibuatnya lagi. Masalahnya, Dek Depi lupa dengan implementasi kodenya. Yang Dek Depi ingat hanya format input, output dan penggunaan rekursi. Lalu, Dek Depi meminta bantuan kamu untuk menyelesaikan *Relief Generator* ini. Kamu pasti mau kan?

Kamu diminta untuk membuat program *Relief Generator* yang menerima input sebuah bilangan bulat dan mengeluarkan pola tertentu yang bentuknya akan bergantung pada bilangan bulat yang diinput (untuk lebih jelasnya, lihat contoh output). **Program hanya akan meminta input satu kali saja.**

Note: Kamu wajib menggunakan implementasi rekursi untuk menyelesaikan program dan dilarang untuk menggunakan *loop* dalam bentuk apapun.

Spesifikasi Program :

INPUT :

Program akan meminta sebuah integer **N** yang akan membuat bentuk dari reliefnya, dijamin bahwa input $N \geq 1$.

OUTPUT :

Mencetak bentuk seperti berikut untuk input **N** berbeda

N = 1

*

N = 2

*

**

*

N = 3

*

**

*

*

**

*

dst.

Contoh Input/Output :

```

Input  :
4
Output :
*
**
*
***
*
**
*
****
*
**
*
***
*
**
*

```

Notes : karakter bercetak **tebal/miring** merupakan input dari user

Komponen Penilaian :

Komponen	Penjelasan	Bobot
Tampilan Program	Interaksi program yang dibuat sesuai dengan contoh.	10%
Implementasi Fungsi Hieroglyph	Mengimplementasikan fungsi yang mencetak pola hieroglyph seperti deskripsi soal dengan menggunakan rekursi .	40 %
Kebenaran Program	Kesesuaian Output program dengan hasil yang diinginkan	35%
Kerapian Kode	Penulisan program mengikuti kaidah dan konvensi yang telah diajarkan. Program ditulis dengan rapi, terstruktur, dan disertakan oleh dokumentasi secukupnya	15 %

Deadline :Selasa, 5 Maret 2019

Pukul 17:20 WIB

Format Pengumpulan :

Program dibuat dalam sebuah file Java yang bernama **[NAMA_FILE].java**

Kumpulkan di slot pengumpulan yang telah disediakan di SCeLE dengan format :

[Kode Asdos]_[Nama]_[Kelas]_[NPM]_Lab[X].zip yang berisi **[NAMA_FILE].java**

Contoh :

TES_DemoSuremo_C_1234567891_Lab4.zip yang berisi **[NAMA_FILE].java** di dalamnya.

Acknowledged Lecturers :

- Fariz Darari, S.Kom, M.Sc., Ph.D.
- Raja Oktovin Parhasian Damanik S.Kom., M.Sc.

Selamat Bekerja !