

Materi Lab 7

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)
Base case merupakan 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 **menghentikan proses rekursi**.
- **Recursive case**
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.

```
def factorial(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** kurang dari atau sama dengan 1 (keterangan: $1! = 1$ dan $0! = 1$, diasumsikan juga **number** tidak bernilai 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 `RecursionError`. Untuk menghindarinya, **pastikan ada base case**, dan **setiap langkah rekursif selalu mengarah ke base case**.