



## Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)

Tahun / Semester : 2023-2024 / Genap

Modul : 4 - Recursion

---

### Tugas Pendahuluan

Tugas pendahuluan dikerjakan secara diketik pada *template* yang telah disediakan dan dikumpulkan secara langsung ke asisten praktikum. Solusi soal pemrograman dapat di-*export* dari *text editor* Notepad++. Tugas pendahuluan wajib dikumpulkan sebelum sesi praktikum dimulai sebagai kehadiran praktikum.

### Soal Konsep

1. Apa yang akan terjadi apabila suatu implementasi *recursion* tidak memiliki *base case* (kasus basis)?
2. Berikan contoh kasus yang **dapat** diselesaikan dengan algoritma rekursif, **tetapi sebaiknya tidak** diselesaikan secara rekursif!
3. Suatu permasalahan dapat diselesaikan secara rekursif dan secara iteratif, dengan jumlah pengulangan/panggilan yang sama antara penyelesaian rekursif dan iteratif. Setelah dikompilasi dari kode bahasa C dan saat dieksekusi, implementasi mana yang bakal lebih cepat antara solusi rekursif dan solusi iteratif? Kenapa?

## Soal Pemrograman

Diberikan suatu bilangan 2 digit, N. Dengan menjumlahkan kuadrat dari digit-digit N, lalu menjumlahkan kuadrat dari digit-digit bilangan yang dihasilkan, lalu meneruskannya lagi dan lagi, dapat dihasilkan suatu *sequence* angka.

Lihat kedua contoh berikut:

$23 \rightarrow 13 \rightarrow 10 \rightarrow 1$

(Keterangan:  $13 = 2^2 + 3^2$ ,  $10 = 1^2 + 3^2$ ,  $1 = 1^2 + 0^2$ )

$69 \rightarrow 117 \rightarrow 51 \rightarrow 26 \rightarrow 40 \rightarrow 17 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow \dots$

(Keterangan:  $117 = 6^2 + 9^2$ , dst.)

Dapat dilihat bahwa ada kedua kemungkinan: *Sequence* berakhir di angka 1 atau berlanjut selamanya.

Buat program yang menerima bilangan N, lalu memberitahukan berapa pengulangan yang diperlukan sebelum dihasilkan angka 1 (jika *sequence* tersebut berakhir) atau memberikan output "infinity" (jika *sequence* tersebut tidak akan berakhir).

Masukkan N: 23

3

Masukkan N: 69

infinity

*Hint:* Perhatikan bahwa *sequence* akan berlanjut selamanya jika didapatkan hasil penjumlahan di antara salah satu dari 4, 16, 37, 58, 89, 145, 42, atau 20.