

---

### EXERCISE 3: ALGORITMA REKURSIF

dikerjakan saat perkuliahan tatap muka

---

#### Aturan pengerjaan tugas:

1. Kerjakan secara berkelompok (1 kelompok terdiri dari 6 orang, lintas kelas, 2 orang dari masing-masing kelas A, B, IKI).
2. Kerjakan **semua** soal yang ada. Naskah ini terdiri dari 2 soal. Soal nomor 1 memuat pemrograman.
3. Tugas teori ditulis tangan (pastikan bisa dibaca).
4. Tugas pemrograman dibuat dengan jupyter notebook. Harap mengurutkan program sesuai dengan urutan pada soal, dan berikan keterangan nomor soal pada setiap program. Penjelasan dapat ditulis dengan “markdown” yang tersedia pada jupyter. Untuk setiap program, tambahkan “**annotation**” atau “**comment**” pada bagian yang Anda rasa perlu, untuk dokumentasi/memudahkan pembacaan program Anda.
5. Saya menganjurkan Anda menulis program dalam python dengan jupyter notebook. File pemrograman disimpan dalam 1 folder yang kemudian dikompres menjadi ekstensi zip. File terdiri dari (1) source code; (2) readme file yang menjelaskan bagaimana program Anda dapat dijalankan melalui terminal, serta rangkuman tentang apa yang Anda kerjakan dalam program tersebut; (3) file tambahan yang dibutuhkan untuk mengetes program Anda. Penamaan setiap file harus rapi.
6. Setiap anggota kelompok **wajib** memahami solusi yang dituliskan oleh kelompoknya.
7. Di akhir perkuliahan, **setiap mahasiswa** wajib memberikan ulasan tentang hasil diskusi kelompoknya dalam bentuk video **mandiri (tidak berkelompok)** berdurasi  $\pm 10$  menit yang diunggah di Youtube. Ulasan memuat hasil diskusi kelompok, apa yang Anda pelajari, apa yang Anda tidak pahami, dan hal-hal lain yang Anda pandang perlu.

\_\_\_\_\_  
*Dengan ini, Anda menyatakan bahwa Anda siap menerima segala konsekuensi  
jika nantinya ditemukan adanya kecurangan dalam pengerjaan tugas ini.*  
\_\_\_\_\_

## Problems

### 1. (Have fun with recursion)

Tulis program dalam bahasa python (buat di jupyter notebook) untuk latihan berikut (kerjakan soal secara berurutan). Pastikan untuk memperhatikan kasus dasar (*base case*) dan pemanggilan rekursi (*recursive call*) Anda!

Tambahkan tabel evaluasi sebagai berikut di laporan file pdf Anda.

Nama program	Poin 1	Poin 2	Poin 3	Poin 4	Keterbatasan program
a. Integer multiplication					
b. Powering					
c. Print Down					
d. Print Up					
e. Reverse string					
f. Prime checking					
g. Fibonacci					

Poin penilaian pada tabel (diisi dengan *Ya/Tidak*)

1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)
2. Program berhasil *running*
3. Program dapat membaca file masukan dan menuliskan luaran.
4. Program dapat mengatasi ketika input tidak sesuai dengan kriteria

### Spesifikasi program:

- (a) Misal  $a$  dan  $b$  adalah bilangan bulat tak negatif. Saat di SD, kita diajarkan bahwa nilai  $b \times a$  ekuivalen  $\underbrace{a + a + \dots + a}_{\text{sebanyak } b}$ . Manfaatkan sifat penjumlahan tersebut untuk membuat fungsi yang mengambil input dua bilangan bulat tak negatif dan mengalikannya secara rekursif.
- (b) Buat fungsi yang memberikan input bilangan bulat  $X$  dan  $n \geq 0$ , dan menghitung  $X^n$  secara rekursif. Anda tidak diperbolehkan menggunakan operator  $**$  (operator pangkat pada python)!
- (c) Buat fungsi menggunakan rekursi untuk mencetak angka dari  $n$  ke 0.
- (d) Modifikasi fungsi sebelumnya untuk membuat sebuah fungsi menggunakan rekursi untuk mencetak angka dari 0 hingga  $n$ .
- (e) Tulis fungsi rekursif yang mengambil input sebuah string dan memberikan return string dalam urutan terbalik (Contoh: input = "Salam" maka output = "malaS"). Satu-satunya operasi string yang boleh Anda gunakan adalah penggabungan string (atau *concatenation*, dengan menggunakan operasi "+").
- (f) Tulis sebuah fungsi rekursif untuk mengecek apakah suatu bilangan  $n$  adalah bilangan prima (Anda harus memeriksa apakah  $n$  habis dibagi dengan bilangan di bawah  $n$ ).
- (g) Tulis fungsi rekursif yang mengambil satu argumen  $n \geq 1$  dan menghitung  $F(n)$ , yakni nilai ke- $n$  dari barisan Fibonacci. Barisan Fibonacci didefinisikan oleh relasi:

$$F(n) = \begin{cases} 1, & n = 1 \\ 1, & n = 2 \\ F_{n-1} + F_{n-2}, & n \geq 3 \end{cases}$$

2. (*Experiment with tower of Hanoi*)

- (a) (*5 poin*) Dalam versi asli “Tower of Hanoi problem”, seperti yang diterbitkan pada tahun 1890-an oleh Édouard Lucas, seorang ahli matematika Prancis, dunia akan berakhir setelah 64 cakram dipindahkan dari Menara Brahma yang mistis. Berdasarkan algoritma yang didiskusikan di kelas, perkirakan berapa tahun yang diperlukan jika para bhikkhu dapat memindahkan satu cakram per menit. (Asumsikan bahwa para bhikkhu dapat bekerja sepanjang waktu, tidak makan, tidur, atau mati.)
- (b) (*5 poin*) Berapa banyak gerakan yang dilakukan oleh cakram terbesar ke- $i$  ( $1 \leq i \leq n$ ) dalam algoritma ini? Jelaskan jawaban Anda!