

# Praktikum Struktur Data

## Modul 3 - Ekspresi Aritmatika Infix, Prefix, Postfix

Senin, 11 April 2022

Tujuan dari modul ini agar mahasiswa memahami konsep struktur data Stack dan implementasinya untuk konversi dan evaluasi ekspresi aritmatika Infix-Postfix. Kerjakan tugas-tugas yang terdapat dalam modul ini, dengan ketentuan sebagai berikut :

1. Semua jawaban modul dikerjakan dalam format \*.ipynb (baik konsep maupun implementasi), jangan lupa diberikan **heading** di setiap cell jawaban, misalkan untuk jawaban konsep Nomor 1, diberikan heading **Konsep\_1**. Silahkan dilihat contoh *lecture notes* yang saya telah bagikan melalui website.
2. Penamaan file ipynb adalah : NPM\_ModulX\_TopikModul.ipynb, misalkan, 200411100077\_Modul3\_StackInfix.ipynb
3. Print menjadi file pdf, dokumen ipynb tersebut dengan nama yang sama, hanya saja berekstensi pdf, misalkan, 200411100077\_Modul3\_StackInfix.pdf
4. Submit **link collaboratory** yang berisi file ipynb tersebut, dan submit **file pdf**
5. Pilih salah satu nomor untuk dibuat video pembelajaran dan upload ke channel youtube masing-masing, dengan hashtag #PraktikumStrukturDataModul2. Format video bebas, wajib diberikan voice over. Akan lebih baik jika mode camera adalah ON.
6. Kejujuran selalu jadi yang utama, kerjakan sendiri, tidak diperkenankan plagiarism

## 1 Konsep

Tulis ringkasan atau penjelasan hal-hal berikut, dengan kata-kata kalian sendiri mengenai, Ekspresi Aritmatika Infix, Prefix, dan Postfix.

## 2 Implementasi

### 2.1 *Konversi Infix ke Postfix*

Buatlah Fungsi dan code yang diperlukan untuk mengkonversi ekspresi aritmatika infix menjadi ekspresi aritmatika postfix, dengan menggunakan struktur data **stack**, dengan output seperti yang ditunjukkan pada Gambar 1 (terdapat visualisasi token yang dibaca dan isi dari stack).

### 2.2 *Evaluasi Postfix*

Buatlah Fungsi dan code yang diperlukan untuk evaluasi ekspresi aritmatika postfix, dengan menggunakan struktur data **stack**, dengan output seperti yang ditunjukkan pada Gambar 2 (terdapat visualisasi operasi matematika yang dikerjakan tiap tahap).

Selamat Mengerjakan, Selalu Latihan, Jujur  
harus dimulai kapanpun, Bertanya jika kurang  
mengerti, #StayAtHome,  
#LearningFromHome

---

*Struktur Data - 2022*  
*Indah Agustien Siradjuddin*

```

M print('((a+b)*(c-d))/f : ',inf2Post('((a+b)*(c-d))/f'))

read token-1 :(
stack : ['(']

read token-2 :(
stack : ['(', '(']

read token-3 :a
stack : ['(', '(', 'a']

read token-4 :+
stack : ['(', '(', '+']

read token-5 :b
stack : ['(', '(', '+', 'b']

read token-6 :)
stack : ['(', '+']

read token-7 :*
stack : ['(', '*']

read token-8 :(
stack : ['(', '*', '(']

read token-9 :c
stack : ['(', '*', '(', 'c']

read token-10 :-
stack : ['(', '*', '(', '-', 'c']

read token-11 :d
stack : ['(', '*', '(', '-', 'd']

read token-12 :)
stack : ['(', '*']

read token-13 :)
stack : []

read token-14 :/
stack : ['/']

read token-15 :f
stack : ['/']

((a+b)*(c-d))/f : ab+cd-*f/

```

(a)

```

M print('a+b*c/d-f+g*h : ',inf2Post('a+b*c/d-f+g*h'))

read token-1 :a
stack : []

read token-2 :+
stack : ['+']

read token-3 :b
stack : ['+', 'b']

read token-4 :*
stack : ['+', '*', 'b']

read token-5 :c
stack : ['+', '*', 'b', 'c']

read token-6 :/
stack : ['+', '/', 'b', 'c']

read token-7 :d
stack : ['+', '/', 'b', 'c', 'd']

read token-8 :-
stack : ['+', '-', 'b', 'c', 'd']

read token-9 :f
stack : ['+', '-', 'b', 'c', 'd', 'f']

read token-10 :+
stack : ['+', '+']

read token-11 :g
stack : ['+', '+', 'g']

read token-12 :*
stack : ['+', '*', 'g']

read token-13 :h
stack : ['+', '*', 'g', 'h']

a+b*c/d-f+g*h : abc*d/+f-gh*+

```

(b)

```

M print('(A*(B+C))-D : ',inf2Post('(A*(B+C))-D'))

read token-1 :(
stack : ['(']

read token-2 :A
stack : ['(', 'A']

read token-3 :*
stack : ['(', '*', 'A']

read token-4 :(
stack : ['(', '*', 'A', '(']

read token-5 :B
stack : ['(', '*', 'A', '(', 'B']

read token-6 :+
stack : ['(', '*', 'A', '(', '+', 'B']

read token-7 :C
stack : ['(', '*', 'A', '(', '+', 'C']

read token-8 :)
stack : ['(', '*', 'A']

read token-9 :)
stack : []

read token-10 :-
stack : ['-']

read token-11 :D
stack : ['-']

(A*(B+C))-D : ABC+*D-

```

(c)

Gambar 1: Konversi Infix<sup>3</sup> ke Postfix dengan Stack

```
▶ print(evaluatePostFix("2 6 +"))
```

```
[1]: 2 + 6 = 8.0  
8.0
```

```
▶ print(evaluatePostFix("10 2 3 + * 4 -"))
```

```
[1]: 2 + 3 = 5.0  
[2]: 10 * 5.0 = 50.0  
[3]: 50.0 - 4 = 46.0  
46.0
```

```
▶ print(evaluatePostFix("2 3 4 * 10 / + 11 - 9 2 * +"))
```

```
[1]: 3 * 4 = 12.0  
[2]: 12.0 / 10 = 1.2  
[3]: 2 + 1.2 = 3.2  
[4]: 3.2 - 11 = -7.8  
[5]: 9 * 2 = 18.0  
[6]: -7.8 + 18.0 = 10.2  
10.2
```

Gambar 2: Evaluasi Ekspresi Aritmatika Postfix