

### Evaluate Postfix

1. Buat tumpukan untuk menyimpan operan.
2. Pindai ekspresi yang diberikan dari kiri ke kanan.
3. a) Jika karakter yang dipindai adalah operand, dorong ke dalam tumpukan.  
b) Jika karakter yang dipindai adalah operator, POP 2 operand dari stack dan melakukan operasi dan PUSH hasilnya kembali ke stack.
4. Ulangi langkah 3 hingga semua karakter dipindai.
5. Ketika ekspresi diakhiri, nomor dalam tumpukan adalah hasil akhir.

Contoh: Biarkan ekspresi yang diberikan menjadi "456 \* +". Kami memindai semua elemen satu per satu.

Step	Input Symbol	Operation	Stack	Calculation
1.	4	Push	4	
2.	5	Push	4,5	
3.	6	Push	4,5,6	
4.	*	Pop(2 elements) & Evaluate	4	$5*6=30$
5.		Push result(30)	4,30	
6.	+	Pop(2 elements) & Evaluate	Empty	$4+30=34$
7.		Push result(34)	34	
8.		No-more elements(pop)	Empty	34(Result)

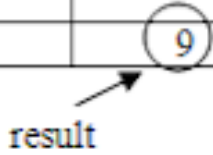
### Evaluate Prefix

Biarkan opndstack menjadi tumpukan operan.

1. opndstack = tumpukan kosong
2. Pindai karakter satu per satu dari kanan ke kiri dalam string.

3. Jika scan char adalah operand, dorong ke dalam tumpukan.
4. Jika scan char adalah operator, pop opnd1, opnd2 dan lakukan operasi yang ditentukan oleh operator. Dorong hasilnya ke dalam tumpukan.
5. Ulangi langkah 2, 3 dan 4 sampai string awalan masukan berakhir.
6. Pop opndstack dan tampilan, yang merupakan nilai yang dibutuhkan dari ekspresi yang diberikan.

Contoh :  $_{*+4325}$

Symbol	opnd1	opnd2	value	opndstack
5				5
2				5, 2
3				5, 2, 3
4				5, 2, 3, 4
+	4	3	7	5, 2
				5, 2, 7
*	7	2	14	5
				5, 14
-	14	5	9	
				9
				

Kode program ada digithub : [sahrialihsani/EX\\_SDA\\_STACK \(github.com\)](https://github.com/sahrialihsani/EX_SDA_STACK)