

NOTASI POLISH - 2

Dosen : Sulistyowati, ST., M.Kom.

Implementasi Stack & Queue pada Notasi Polish

- Untuk merubah ungkapan/notasi Infix menjadi notasi Prefix maupun notasi Postfix, bisa dilakukan dengan menggunakan stack dan queue.
- Yang perlu diperhatikan untuk merubah notasi polish dengan menggunakan stack dan queue ini adalah pembacaan ungkapan harus dilakukan per karakter, dimulai dari kiri ke kanan.

Notasi INFIX → POSTFIX

Algoritma untuk menuliskan/mengubah sebuah ungkapan dari notasi Infix ke dalam notasi Postfix adalah :

1. Baca ungkapan satu per satu (per karakter) dari kiri ke kanan.
2. Jika simbol adalah "(" (kurung buka), maka PUSH ke dalam *stack*
3. Jika simbol adalah ")" (kurung tutup), POP elemen-elemen *stack* dan ENQUEUE sampai ketemu pertama kali simbol "(" . Tapi simbol "(" tidak perlu di ENQUEUE.
4. Jika simbol adalah sebuah *operand*, tanpa melakukan perubahan elemen *stack*, langsung ENQUEUE *operand* tersebut.
5. Jika simbol adalah operator dan *stack* masih kosong, maka PUSH operator ke dalam *stack*.
6. Jika simbol adalah sebuah operator, maka :
 - ✓ Jika TOP *stack* adalah operator dengan level lebih tinggi atau sama, maka POP elemen *stack* dan ENQUEUE. Lanjutkan proses seperti ini sampai TOP merupakan "(" atau operator dengan level lebih rendah. Lalu operator (yang diamati) PUSH ke dalam *stack*.
 - ✓ Jika TOP *stack* adalah operator dengan level lebih rendah, maka PUSH operator.
7. Ulangi langkah 2 sampai 6.
8. Jika ekspresi sudah habis, maka POP semua elemen *stack*, sehingga *stack* menjadi hampa dan ENQUEUE.
9. Kemudian DEQUEUE semua elemen.

Notasi INFIX → PREFIX

Algoritma untuk menuliskan/mengubah sebuah ungkapan dari notasi Infix ke dalam notasi Prefix adalah :

1. Baca ungkapan dari kanan ke kiri.
2. Jika simbol adalah ")" (kurung tutup), maka PUSH ke dalam *stack*
3. Jika simbol adalah "(" (kurung buka), POP elemen-elemen *stack* dan ENQUEUE sampai ketemu pertama kali simbol ")". Tapi simbol ")" tidak perlu di ENQUEUE.
4. Jika simbol adalah sebuah *operand*, tanpa melakukan perubahan elemen *stack*, langsung ENQUEUE *operand* tersebut.
5. Jika simbol adalah operator dan *stack* masih kosong, maka PUSH operator ke dalam *stack*.
6. Jika simbol adalah sebuah operator, maka :
 - ✓ Jika TOP *stack* adalah operator dengan level lebih tinggi, maka POP elemen *stack* dan ENQUEUE. Lanjutkan proses seperti ini sampai TOP merupakan ")" atau operator dengan level lebih rendah atau sama. Lalu operator (yang diamati) PUSH ke dalam *stack*.
 - ✓ Jika TOP *stack* adalah operator dengan level lebih rendah atau sama, maka PUSH operator.
7. Ulangi langkah 2 sampai 6.
8. Jika ekspresi sudah habis, maka POP semua elemen *stack*, sehingga *stack* menjadi hampa dan ENQUEUE.
9. Kemudian DEQUEUE semua elemen *queue* dan PUSH ke dalam *stack* baru
10. Lalu POP semua elemen *stack*

Implementasi Stack & Queue pada Notasi Polish

- Rubahlah notasi Infix berikut menjadi notasi Postfix : $(A + B) / ((C - D) * E ^ F)$
- Ilustrasi penggunaan stack dan queue terlihat sebagai berikut :

Karakter dibaca	Isi stack	Isi queue
((
A	(A
+	(+	
B	(+	A B
)		A B +
/	/	
(/(
(/((
C	/((A B + C
-	/((-	
D	/((-	A B + C D
)	/(A B + C D -
*	/(*	
E	/(*	A B + C D - E
^	/(* ^	
F	/(* ^	A B + C D - E F
)	/(*	A B + C D - E F ^
	/(A B + C D - E F ^ *
	/	
		A B + C D - E F ^ * /

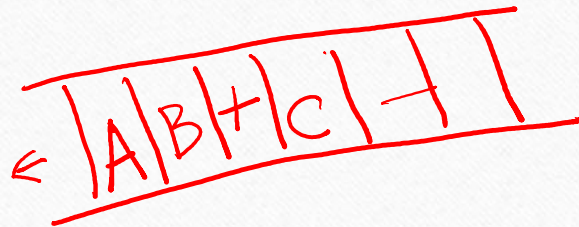
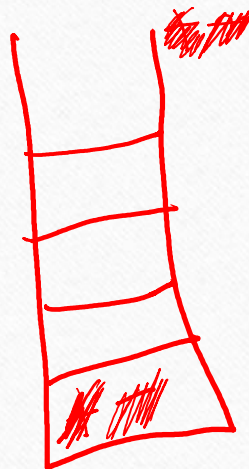
Latihan soal

Rubahlah notasi Infix berikut ke dalam notasi **Postfix** dan **Prefix** dengan menggunakan stack dan queue (seperti contoh di slide 10)

- $A + B - C / D + E$
- $A / B - C ^ D + E * F / G - H$
- $A / (B + C ^ D) * E - F$
- $(A / B * C) - (D + E * (F ^ G - H) * I) ^ J$
- $A - B - C / D / E + F + G * H * I ^ J$

$A + B - C \rightarrow$ Prefix: $- + A B C$
 Postfix: $A B + C -$

Postfix:

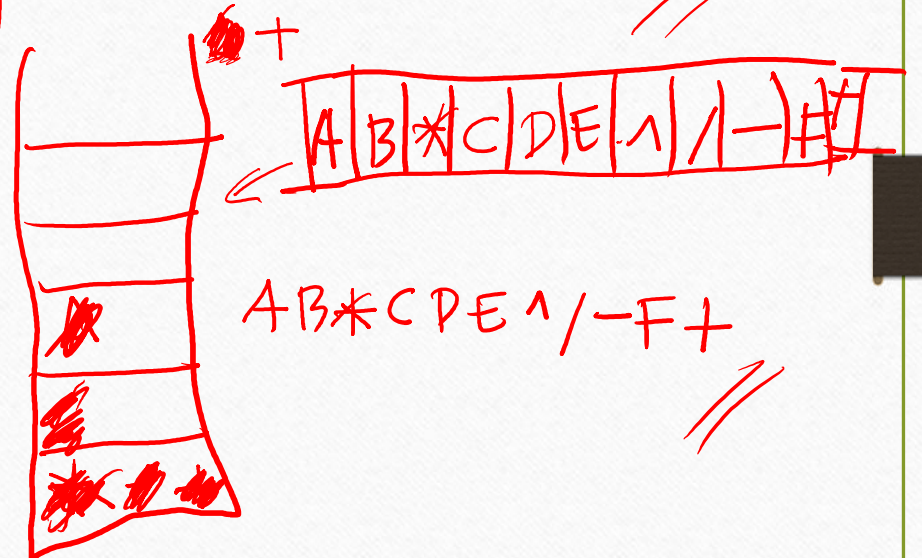


$A B + C -$

$A * B - C / D ^ E + F$

↓

$A B * C D E ^ / - F +$

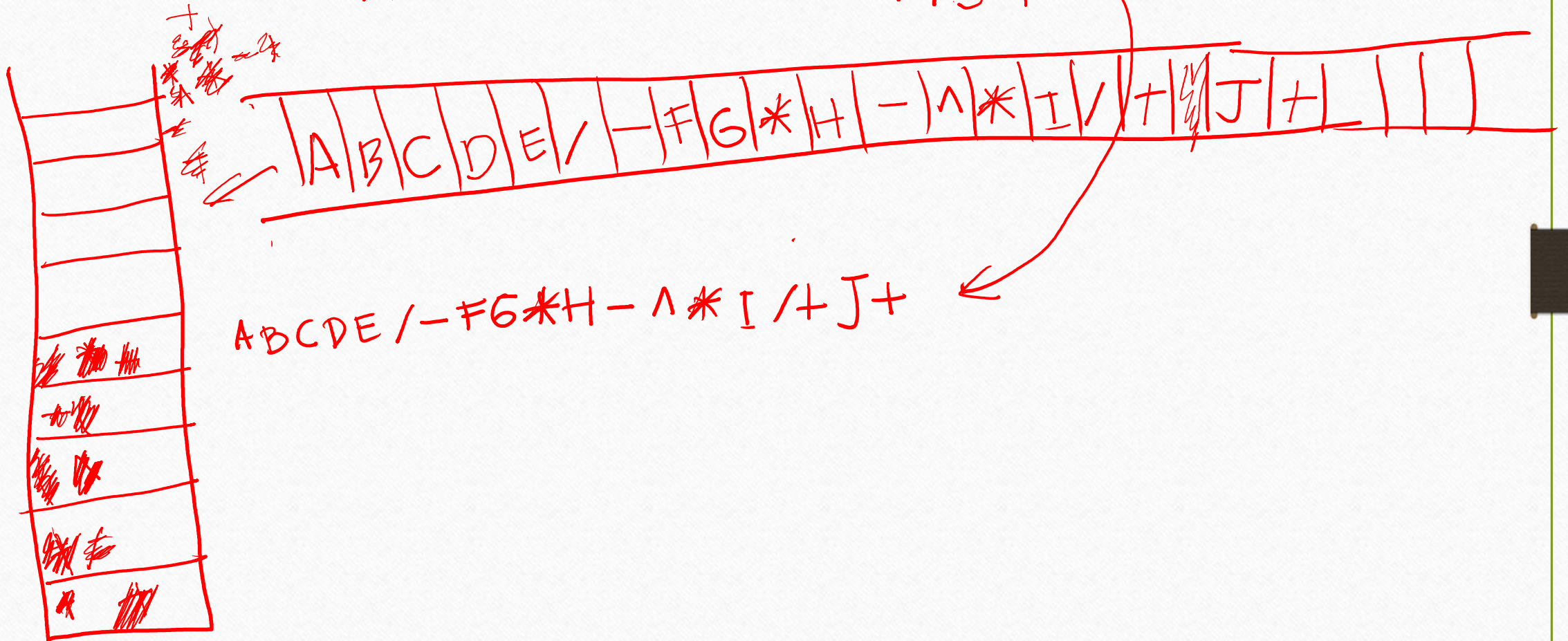


$A B * C D E ^ / - F +$

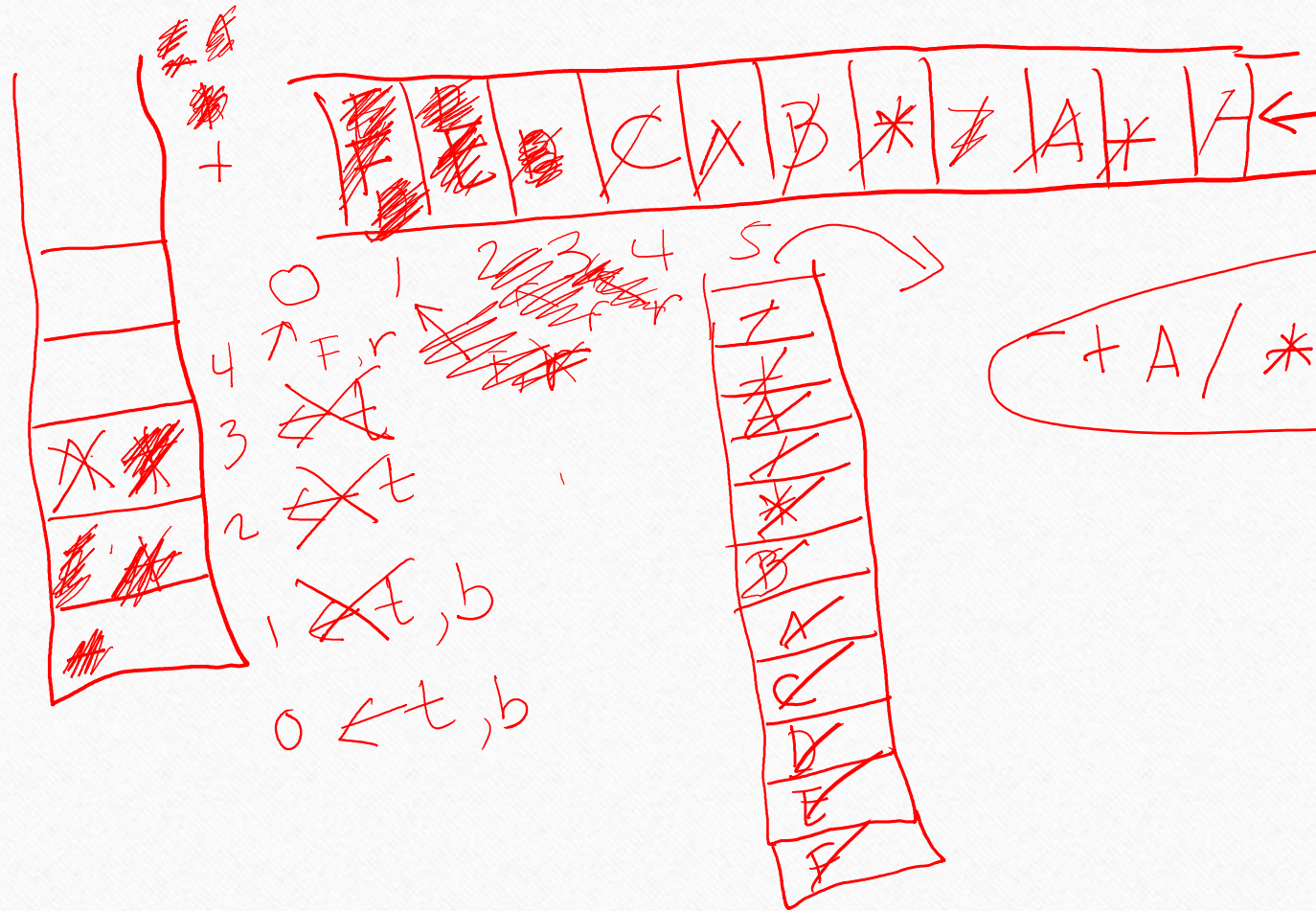
$$A + B * (C - D / E) ^ (F * G - H) / I + J$$

Postfix:

$$A B C D E / - F G * H - ^ * I / + J +$$



$A + B * C^{\wedge} D / E - F \rightarrow \text{Prefik: } - + A / * B^{\wedge} C D E F$



$- + A / * B^{\wedge} C D E F$