



Pertemuan – 10

Tumpukan (*Stack*)

Dipersiapkan oleh : Boldson Herdianto. S., S.Kom., MMSI.

Definisi

- ☀ Tumpukan adalah kumpulan elemen-elemen data yang disimpan dalam satu lajur linier.
- ☀ Kumpulan elemen-elemen data hanya boleh diakses pada satu lokasi saja yaitu pada posisi ATAS (TOP)
- ☀ Tumpukan disebut juga “Push Down Stack” yaitu penambahan elemen baru (PUSH) dan penghapusan elemen dari tumpukan (POP).
- ☀ Sistem pada pengaksesan pada tumpukan menggunakan sistem LIFO (Last In First Out), artinya elemen yang terakhir masuk itu yang akan pertama dikeluarkan dari tumpukan.

KAMUS DATA

Tumpukan (Stack)

Const

MAKSTUM = 80; {Kapasitas maksimal dari tumpukan}

Type

JenisElemen = char;

Tumpukan = record

Elemen : Array[1..MAKSTUM]of JenisElemen;

Atas : 0..MAKSTUM;

End;

Operasi-operasi Dasar

- ★ **CREATESTACK(S)** : Membuat Tumpukan baru S, dengan jumlah elemen kosong.
- ★ **MAKENULL (S)** : Mengosongkan Tumpukan S, jika ada elemen maka semua elemen dihapus.
- ★ **EMPTY** : Tumpukan kosong?- menguji apakah tumpukan kosong.
- ★ **PUSH (x, S)** : memasukkan elemen baru x ke dalam tumpukan S.
- ★ **POP (S)** : mengeluarkan elemen posisi atas pada tumpukan S

NOTASI ARITMETIK (INFIX, PREFIX, POSTFIX)

- ✦ Infix adalah penulisan notasi aritmatik baku
- ✦ Prefix adalah keadaan dimana simbol operator diletakkan sebelum dua operand.
- ✦ Postfix adalah keadaan dimana simbol operator diletakkan sesudah dua operand.

Bentuk Umum

Notasi	Bentuk Umum	Contoh
Infix	Operand operator operand	$A + B$
Prefix	Operator operand operand	$+ A B$
Postfix	Operand operand operator	$A B +$

NOTASI INFIX

Notasi Infix

Prioritas pengerjaan dalam notasi infix:

1. Tanda kurung : (.....)
2. Eksponensial atau pangkat : ^
3. Perkalian, Pembagian : * , /
4. Penjumlahan, Pengurangan : +, -

Contoh pengerjaan Infix

$$(A - B) * (C + D)$$

Prioritas pengerjaan soal di atas adalah:

- a. Dalam kurung yang paling kiri : (A-B)
- b. Dalam kurung yang kedua : (C+D)
- c. Perkalian hasil pengurangan dengan hasil penjumlahan

INFIX ke PREFIX : $(A + B) - (C * D)$

Urutan pengerjaan infix:

- a. Pengerjaan dalam kurung ke-1 : $(A + B)$,
prefix-nya adalah $+AB$
- b. Pengerjaan dalam kurung ke-2 : $(C * D)$,
prefix-nya adalah $*CD$
- c. Terakhir adalah operator $-$: $+AB - *CD$,
prefix-nya adalah **$- +AB *CD$**

INFIX ke POSTFIX : $(A + B) - (C * D)$

Urutan pengerjaan Infix:

- a. Pengerjaan dalam kurung ke-1 : $(A + B)$, postfix-nya adalah $AB+$
- b. Pengerjaan dalam kurung ke-2 : $(C * D)$, postfix-nya adalah CD^*
- c. Terakhir adalah operator $-$: $AB+ - CD^*$, postfix-nya adalah **$AB+ CD^* -$**

PREFIX ke INFIX : +/*A B C D

Untuk pengerjaan prefix, mencari operator dimulai dari operand terkanan.

- a. Cari operator ke-1: *****, ambil dua operand sebelumnya A dan B, sehingga infix-nya adalah $(A * B)$
- b. Cari operator ke-2: **/**, ambil dua operand sebelumnya $(A * B)$ dan C, sehingga infix-nya adalah $((A * B) / C)$
- c. Cari operator ke-3: **+**, ambil dua operand sebelumnya $((A * B) / C)$ dan D, sehingga infix-nya adalah $((A * B) / C) + D$

PREFIX ke POSTFIX : +/*A B C D

Untuk pengerjaan prefix, mencari operator dimulai dari operand terkanan.

- a. Cari operator ke-1: $*$, ambil dua operand sebelumnya A dan B, sehingga postfix-nya adalah AB^*
- b. Cari operator ke-2: $/$, ambil dua operand sebelumnya AB^* dan C, sehingga postfix-nya adalah $AB^*C/$
- c. Cari operator ke-3: $+$, ambil dua operand sebelumnya $AB^*C/$ dan D, sehingga postfix-nya adalah **$AB^*C/D+$**

POSTFIX ke INFIX : A B C D * / -

Untuk pengerjaan postfix, mencari operator dimulai dari operand terkiri.

- a. Cari operator ke-1: *, ambil dua operand sebelumnya C dan D, sehingga infix-nya adalah $(C * D)$
- b. Cari operator ke-2: /, ambil dua operand sebelumnya B dan $(C * D)$, sehingga infix-nya adalah $(B / (C * D))$
- c. Cari operator ke-3: -, ambil dua operand sebelumnya A dan $(B / (C * D))$, sehingga infix-nya adalah **$- A - (B / (C * D))$**

POSTFIX ke PREFIX : A B C D * / -

Untuk pengerjaan postfix, mencari operator dimulai dari operand terkiri.

- a. Cari operator ke-1: *, ambil dua operand sebelumnya C dan D, sehingga prefix-nya adalah *CD
- b. Cari operator ke-2: /, ambil dua operand sebelumnya B dan *CD, sehingga prefix-nya adalah / B *CD
- c. Cari operator ke-3: -, ambil dua operand sebelumnya A dan / B *CD, sehingga prefix-nya adalah - **A / B *CD**