

Metode Konversi antara *Infix*, *Prefix*, dan *Postfix* pada Ekspresi Matematika tanpa Menggambar Pohon

Satria Ardhe Kautsar (135 05 004)

Jurusan Teknik Informatika ITB, Jln. Ganesha Bandung, email: sdf_88@yahoo.com

Abstract – Makalah ini membahas tentang metode atau cara dalam konversi dari *infix*, *prefix*, maupun *postfix* ke dalam format penelusuran lainnya tanpa harus menggambar pohon binernya, yaitu dengan menggunakan “alat bantu” berupa tanda kurung.

Kata Kunci : *Infix*, *Prefix*, *Postfix*, Pohon Biner, Metode Konversi, Ekspresi Matematika.

1. PENDAHULUAN

Pada awalnya saat mempelajari tentang *infix*, *prefix*, dan *postfix* di mata kuliah Struktur Diskrit, penulis merasa kesulitan. Hal tersebut dikarenakan selama ini penulis terbiasa dengan penulisan ekspresi matematika *infix* dengan ditambah tanda bantu, sehingga penulisan dengan *postfix* ataupun *infix* terasa sangat asing dan membingungkan, ditambah lagi tanpa tanda bantu berupa kurung.

Oleh karena itu, dengan dituliskannya makalah ini penulis ingin menjelaskan metode atau cara mudah dalam mengkonversi ketiga bentuk tersebut, sekaligus memahami perbedaan yang mendasari ketiganya.

Karena sebagian besar dari kita selama ini terbiasa dengan tanda bantu ‘()’ dalam matematika, maka metode tersebut yang penulis terapkan pada makalah ini, dengan harapan pembaca dapat lebih mudah memahami dan mempelajarinya.

2. METODE

Bab ini menjelaskan satu persatu metode dalam mengubah *infix* menjadi *postfix* dan *prefix*, dan sebaliknya bagaimana mengubah *postfix* atau *prefix* menjadi *infix*.

2.1. *Infix* ke *Postfix*

Misalkan kita memiliki ekspresi matematika dalam bentuk *infix* sebagai berikut :

$$((a * b) + c) / (d - (e + f))$$

Untuk mengubahnya menjadi *postfix*, hal yang pertama kita lakukan adalah mengambil operator yang berada pada tingkat paling atas (yang berada diantara 2 tanda kurung terluar, dalam hal ini ‘/’), lalu menempatkannya di sebelah kanan format *postfix* yang akan kita buat, seperti berikut :

$$((\quad)(\quad)) /$$

Setelah itu, tinjau masing-masing sub-ekspresi yang ada di antara tanda ‘/’ tersebut, yaitu $((a*b)+c)$ dan $(d-(e+f))$.

Ambil ekspresi terluar pada masing-masing sub-ekspresi tersebut, dan tempatkan di sebelah kanan seperti pada langkah sebelumnya, namun kali ini di dalam tanda kurung anaknya.

$$(((\quad)(\quad)) +) ((\quad)(\quad) -) /$$

Ulangi langkah tersebut untuk sub-ekspresi berikutnya, jika sub-ekspresi berikutnya tidak mengandung operator, langsung masukkan huruf yang bersangkutan.

$$(((\underline{a})(\underline{b}) *) (\underline{c}) +) ((\underline{d})(\underline{e})(\underline{f}) +) -) /$$

lalu,

$$(((\underline{a})(\underline{b}) *) (\underline{c}) +) ((\underline{d})(\underline{e})(\underline{f}) +) -) /$$

Sehingga setelah tanda kurung dihilangkan, persamaannya menjadi :

$$a b * c + d e f + -$$

2.2. *Infix* ke *Prefix*

Dalam mengubah dari *infix* menjadi *prefix*, metode yang digunakan sama seperti mengubah *infix* menjadi *postfix*, bedanya hanya pada *prefix* operatornya diletakkan di depan.

Sebagai contoh, kita memiliki ekspresi matematika dalam bentuk *infix* seperti pada subbab 2.1 :

$$((a * b) + c) / (d - (e + f))$$

Seperti dalam mengubah *infix* menjadi *postfix*, hal yang pertama kita lakukan adalah mengambil operator yang berada pada tingkat paling atas, namun saat ini kita menaruhnya di sebelah kiri.

$$/((\quad) (\quad))$$

Lalu kita lanjutkan untuk sub-ekspresi sub-ekspresi selanjutnya sampai tidak ada lagi sub-ekspresi yang bisa ditinjau.

$$/((+ (\quad) (\quad)) (- (\quad) (\quad)))$$

$$/((+ (* (\quad) (\quad)) (\underline{c})) (- (\underline{d}) (+ (\quad) (\quad))))$$

$$(/ (+ (* (\underline{a}) (\underline{b})) (\underline{c})) (- (\underline{d}) (+ (\underline{e}) (\underline{f}))))$$

Sehingga persamaannya menjadi :

$$/ + * a b c - d + e f$$

2.3. Prefix / Postfix ke Infix

Untuk mengubah kembali dari *prefix* maupun *postfix* menjadi *infix*, pada dasarnya memiliki metode yang sama dengan *infix* ke *postfix* ataupun *infix* ke *prefix*, yang perlu diperhatikan hanyalah penempatan operator dan tanda bantuannya.

Jika pada *prefix* operator berada di depan tanda bantu dan pada *postfix* operator berada di belakang tanda bantu, pada *infix* operator berada di tengah-tengah tanda bantu.

Ambil contoh persamaan dalam bentuk *postfix* yang sudah kita dapatkan pada bab 2.2 :

$$/ + * a b c - d + e f$$

Hal yang pertama kali kita lakukan adalah menambahkan tanda bantu agar lebih mudah dalam mengubahnya menjadi bentuk lain, dengan cara menambahkan tanda kurung di awal sub-ekspresi, dan menutupnya di akhir sub-ekspresi.

$$\begin{aligned} & (/ + * a b c - d + e f) \\ & (/ (+ * a b c) - d + e f) \\ & (/ (+ * a b c) (- d + e f)) \\ & (/ (+ (* a b) c) (- d + e f)) \\ & (/ (+ (* a b) c) (- d (+ e f))) \end{aligned}$$

Setelah terbentuk persamaan dengan tanda bantu, maka lebih mudah untuk mengubahnya menjadi *infix*. Caranya sama seperti mengubah *infix* menjadi *postfix* maupun *prefix*, namun kali ini operatornya kita tempatkan di tengah-tengah kedua tanda kurung.

$$\begin{aligned} & ((\quad) / (\quad)) \\ & (((\quad) + (\quad)) / ((\quad) - (\quad))) \\ & ((((\quad) * (\quad)) + (\underline{c})) / ((\underline{d}) - ((\quad) + (\quad)))) \\ & (((\underline{a}) * (\underline{b})) + (\underline{c})) / ((\underline{d}) - ((\underline{e}) + (\underline{f}))) \end{aligned}$$

Sehingga didapatkan persamaan seperti pada awal bab 2.1 :

$$((a*b) + c) / (d - (e+f))$$

3. KESIMPULAN

Kesimpulan yang dapat diambil dari isi makalah ini adalah :

- Membaca persamaan dalam bentuk *prefix*, *infix*, maupun *postfix* akan lebih mudah dengan tanda bantu.
- Perbedaan bentuk *prefix*, *infix*, dan *postfix* hanya terletak pada penempatan operatornya saja, *postfix* di belakang, *prefix* di depan, dan *infix* di tengah sub-ekspresi.

DAFTAR REFERENSI

1[1] Munir, Rinaldi. 2008. "Diktat IF2153 Struktur Diskrit". Departemen Teknik Informatika, Institut Teknologi Bandung.