01158 - ALGORITMA DAN STRUKTUR DATA 2

STACK

04

OLEH : ANDRI HERYANDI, M.T.

CONTOH IMPLEMENTASI STACK

OLEH : ANDRI HERYANDI, M.T.

IMPLEMENTASI STACK DI DUNIA KOMPUTER

- Bagaimana melakukan perhitungan rumus di bawah ini dengan memperhatikan prioritas operator.

```
Q=5+8*4-6^2+(9+1)/3
```

IMPLEMENTASI STACK DI DUNIA KOMPUTER

- Penyelesaian secara manual

1. Hitung yang berada di dalam kurung

   Q=5+8*4-6^2+(9+1)/3 → Q=5+8*4-6^2+10/3

2. Hitung yang mempunyai prioritas paling tinggi (pangkat)

   Q=5+8*4-6^2+10/3 → Q=5+8*4-36+10/3

3. Hitung dari kiri dan kanan untuk prioritas selanjutnya (kali dan bagi)

   Q=5+8*4-36+10/3 → Q=5+32-36+10/3

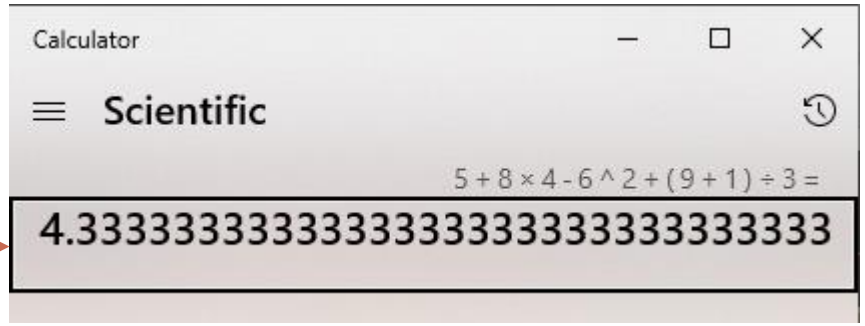
   Q=5+32-36+10/3 → Q=5+32-36+3.33

4. Hitung dari kiri dan kanan untuk prioritas selanjutnya (tambah dan kurang)

   Q=5+32-36+3.33 → Q=37-36+3.33

   Q=37-36+3.33 → Q=1+3.33

   Q=1+3.33 → Q=4.33



IMPLEMENTASI STACK DI DUNIA KOMPUTER

- Bagaimana melakukan perhitungan rumus secara manual agar bisa dilakukan oleh komputer?
- Kesulitan yang terjadi adalah :
  - Proses perhitungan dilakukan kadang dari kanan (karena prioritas), tapi kemudian kembali lagi ke kiri, lalu pindah lagi ke kanan dan mungkin pindah lagi ke kiri. Ini sulit untuk dibuat programnya.
- Solusinya adalah :

Postfix Expression
atau
Polish Notation

EKSPRESI ARITMATIKA

Ekspresi aritmatika dapat disusun dalam salah satu dari 3 cara berikut :

1. Ekspresi Infix

   Ekspresi aritmatika Infix adalah ekspresi yang ditulis dengan cara menuliskan operator aritmatika pada bagian dalam (in) ekspresi diapit oleh 2 operannya.

   Contoh : A + B (dibaca : A dijumlahkan dengan B)

2. Ekspresi Prefix

   Ekspresi aritmatika Prefix adalah ekspresi yang ditulis dengan cara menuliskan operator aritmatika pada bagian depan/sebelum (pre) disusul dengan 2 operannya.

   Contoh : + A B (dibaca : Jumlahkan A dengan B)

3. Ekspresi Postfix

   Ekspresi aritmatika postfix adalah ekspresi yang ditulis dengan cara menuliskan 2 operan terlebih dahulu, kemudian disusul dengan operatornya (post).

   Contoh : A B + (dibaca : A dan B dijumlahkan)

EKSPRESI ARITMATIKA

Contoh Lain :

	Ekspresi	Urutan eksekusi
Infix	A + B * C	[A + [B * C]]
Prefix	+ A * B C	[+ A [* B C]]
Postfix	A B C * +	[A [B C *] +]

	Ekspresi	Urutan eksekusi
Infix	A + B * (C - D)	[A + [B * [C – D]]]
Prefix	+ A * B - C D	[+ A [* B [– C D]]]
Postfix	A B C D - * +	[A [B [C D -] *] +]

MENGEVALUASI RUMUS ARITMATIKA (ARITHMETIC EXPRESSION EVALUATION)

- INPUT : Rumus Aritmatika dalam bentuk Infix
- OUTPUT : Hasil Perhitungannya

- Langkah-langkahnya adalah :
  1. Konversikan rumus dalam notasi Infix ke dalam bentuk notasi Postfix
  2. Lakukan perhitungan pada notasi Postfix

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Asumsikan Q adalah ekspresi dalam bentuk Infix, dan P adalah ekspresi dalam bentuk Postfix. Algoritma konversi notasi Infix ke notasi Postfix adalah :

1. Push "(" ke dalam stack, dan tambahkan ")" ke akhir dari Q.
2. Scan Q dari kiri ke kanan dan ulangi Langkah 3 s.d 6 untuk setiap elemen Q sampai stack Kosong.
3. Jika yang discan adalah operan, maka tambahkan ke P.
4. Jika yang discan adalah "(", maka push ke stack.
5. Jika yang discan adalah operator (+,-,*,/,^) (asumsikan namanya OPR), maka
   A. Periksa isi di top stack. Selama operator yang ada di top mempunyai prioritas sama dengan atau lebih besar dari OPR maka lakukan pop dari stack (simpan di variable T) dan tambahkan T ke P.
   B. Tambahkan OPR ke stack.
   [Akhir dari IF Langkah 5]
6. Jika yang discan adalah ")" maka :
   A. Ulangi pop dari stack (simpan di variable T) dan tambahkan T ke P sampai ditemukan tanda "(".
   B. Hilangkah "(" (jangan ditambahkan ke P).
   [Akhir dari IF Langkah 6]
   [Akhir dari pengulangan Langkah 2]
7. Keluar/Selesai.

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Contoh 1 :

$$Q=5+(2-3)*4$$

1. Tambahkan tanda "(" ke stack, dan tambahkan tanda ")" ke dalam Q
   Q=5+(2-3)*4), Isi stack: [**(**], P:[kosong]
2. Scan Q satu persatu dari kiri ke kanan
   1) Operan "5", karena operan maka tambahkan ke P
      Isi Stack: [**(**], P: [**5**]
   2) Operator "+", karena operator dan isi top stack bukan operator yang prioritasnya lebih besar atau sama dengan operator "+" maka tambahkan "+" ke stack.
      Isi Stack: [**(+**], P: [**5**]
   3) Tanda "(", maka push ke stack.
      Isi Stack: [**(+ (**], P: [**5**]
   4) Tanda "2", karena operan maka tambahkan ke P.
      Isi Stack : [**(+ (**], P: [**5 2**]
   5) Operator "-", karena operator dan isi top stack bukan operator yang prioritasnya lebih besar atau sama dengan operator "-" maka tambahkan "-" ke stack.
      Isi Stack : [**(+ (-**], P: [**5 2**]

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

6) Operan “3”, karena operan maka tambahkan ke P

   Isi Stack: [**(+ (-**], P: [**5 2 3**]

7) Tanda “)”, maka ulangi pop dari stack sampai menemukan “(“. Setiap hasil pop tambahkan ke P kecuali “(“.

   Pop stack, dapat “-”, tambahkan ke P. Isi Stack: [**(+ (**], P: [**5 2 3 -**]

   Pop stack, dapat “(“, abaikan, tidak usah ditambahkan ke P. Isi Stack: [**(+**], P: [**5 2 3 -**]

8) Operator “*”, karena operator dan isi top stack bukan operator yang lebih besar atau sama dengan operator “*” maka tambahkan “*” ke stack.

   Isi Stack: [**(+ ***], P: [**5 2 3 -**]

9) Operan “4”, karena operan maka tambahkan ke P

   Isi Stack: [**(+ ***], P: [**5 2 3 - 4**]

10) Tanda “)”, maka ulangi pop dari stack sampai menemukan “(“. Setiap hasil pop tambahkan ke P kecuali “(“.

    Pop stack, dapat “*”, tambahkan ke P. Isi Stack: [**(+**], P: [**5 2 3 - 4***]

    Pop stack, dapat “+”, tambahkan ke P. Isi Stack: [**(**], P: [**5 2 3 - 4 * +**]

    Pop stack, dapat “(“, abaikan, tidak usah ditambahkan ke P. Isi Stack: [ ], P: [**5 2 3 - 4 * +**]

11) Q habis, dan stack kosong maka proses selesai. Hasil akhir ada di P = **5 2 3 - 4 * +**

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Hasil Akhir Contoh 1 :

$$Q=5+(2-3)*4$$

Dikonversikan menjadi Postfix

$$P = 5\ 2\ 3\ -\ 4\ *\ +$$

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Contoh 2 :

$$Q = 7 + \frac{9-3}{2+4} * 10^{8-6} + (5 - 1)$$

Konversikan ke notasi Infix

Q=7+(9-3)/(2+4)*10^(8-6)+(5-1)

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Contoh 2 :

Q=7+(9-3)/(2+4)*10^(8-6)+(5-1)

Kondisi Awal :

Tambahkan “(“ ke stack, dan tambahkan “)” ke Q

Q=7+(9-3)/(2+4)*10^(8-6)+(5-1))

Q	Stack	P (Postfix)
	(	
7	(	7
+	(+	7
(	(+ (	7

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Q	Stack	P (Postfix)
9	(+ (	7 9
–	(+ (–	7 9
3	(+ (–	7 9 3
)	(+	7 9 3 –
/	(+ /	7 9 3 –
(	(+ / (	7 9 3 –
2	(+ / (	7 9 3 – 2
+	(+ / (+	7 9 3 – 2

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Q	Stack	P (Postfix)
4	(+ / (+	7 9 3 – 2 4
)	(+ /	7 9 3 – 2 4 +
*	(+ *	7 9 3 – 2 4 + /
10	(+ *	7 9 3 – 2 4 + / 10
^	(+ * ^	7 9 3 – 2 4 + / 10
(	(+ * ^ (	7 9 3 – 2 4 + / 10
8	(+ * ^ (	7 9 3 – 2 4 + / 10 8
–	(+ * ^ (–	7 9 3 – 2 4 + / 10 8

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Q	Stack	P (Postfix)
6	(+ * ^ (-	7 9 3 - 2 4 + / 10 8 6
)	(+ * ^	7 9 3 - 2 4 + / 10 8 6 -
+	(+	7 9 3 - 2 4 + / 10 8 6 - ^ * +
(	(+ (	7 9 3 - 2 4 + / 10 8 6 - ^ * +
5	(+ (	7 9 3 - 2 4 + / 10 8 6 - ^ * + 5
-	(+ (-	7 9 3 - 2 4 + / 10 8 6 - ^ * + 5
1	(+ (-	7 9 3 - 2 4 + / 10 8 6 - ^ * + 5 1

KONVERSI NOTASI INFIX KE NOTASI POSTFIX

Q	Stack	P (Postfix)
)	(+	7 9 3 - 2 4 + / 10 8 6 - ^ * + 5 1 -
)		7 9 3 - 2 4 + / 10 8 6 - ^ * + 5 1 - +
		7 9 3 - 2 4 + / 10 8 6 - ^ * + 5 1 - +

Hasil Akhir Contoh 2

Notasi Infix

```
7+(9-3)/(2+4)*10^(8-6)+(5-1))
```

Notasi Postfix

```
7 9 3 - 2 4 + / 10 8 6 - ^ * + 5 1 - +
```

PERHITUNGAN NOTASI POSTFIX

- Perhitungan notasi postfix adalah proses melakukan perhitungan rumus yang telah tersusun dalam notasi postfix.
- Hasil akhir dari proses ini adalah hasil perhitungan rumus tersebut (angka).

PERHITUNGAN NOTASI POSTFIX

Algoritmanya adalah :

Asumsikan bahwa P adalah notasi postfix.

1. Tambahkan “)” ke ujung P sebagai penanda sentinel.
2. Scan elemen P dari kiri ke kanan, ulangi Langkah 3 dan 4 untuk setiap elemen P sampai ditemukan tanda “)” (sentinel).
3. Jika elemennya berupa operan maka tambah ke stack.
4. Jika elemennya berupa operator (asumsikan disimpan dalam variable OPR) maka :
   a) Pop dari top stack, simpan di variable A, kemudian Pop lagi dari top stack, simpan di variable B.
   b) Lakukan perhitungan dengan urutan B OPR A.
   c) Hasil perhitungan langsung di-push ke stack.

   [akhir dari IF Langkah 4]

   [akhir dari pengulangan Langkah 2]
5. Setelah perulangan 2 selesai, maka hasil akhir ada di top dari stack.
6. Selesai.

PERHITUNGAN NOTASI POSTFIX

Contoh 1 :

Notasi Infix

$$Q=5+(2-3)*4$$

Notasi Postfix

$$P = 5\ 2\ 3\ -\ 4\ *\ +$$

PERHITUNGAN NOTASI POSTFIX

Contoh 1:

P= 5 2 3 – 4 * +

Proses perhitungan :

1. Tambahkan “)” ke akhir P

   P= 5 2 3 - 4 * +), Isi stack: [ ]

2. Scan elemen dari P

   1) Dapat “5”, karena operan maka tambahkan ke stack.

      1) Isi Stack: [5]

   2) Dapat “2”, karena operan maka tambahkan ke stack.

      Isi Stack: [5 2]

   3) Dapat “3”, karena operan maka tambahkan ke stack

      Isi Stack: [5 2 3]

PERHITUNGAN NOTASI POSTFIX

4) Dapat “-” (asumsikan disimpan di variable OPR), karena operator maka :
   a. Pop dari stack, simpan di variable A, maka isi Stack menjadi : [5 2] dan variable A: 3
   b. Pop dari stack, simpan di variable B, maka isi Stack menjadi : [5] dan variable B: 2
   c. Hitung B OPR A sehinga “2 – 3”, menghasilkan “-1”.
   d. Tambahkan ke stack, sehingga isi stack menjadi [5 -1]
5) Dapat “4”, karena operan maka tambahkan ke stack.

   Isi stack: [5 -1 4]
6) Dapat “*” (asumsikan disimpan di variable OPR), karena operator maka :
   a. Pop dari stack, simpan di variable A, maka isi Stack menjadi : [5 -1] dan variable A: 4
   b. Pop dari stack, simpan di variable B, maka isi Stack menjadi : [5] dan variable B: -1
   c. Hitung B OPR A sehinga “-1 * 4”, menghasilkan “-4”.
   d. Tambahkan ke stack, sehingga isi stack menjadi [5 -4]

PERHITUNGAN NOTASI POSTFIX

7) Dapat “+” (asumsikan disimpan di variable OPR), karena operator maka :
   a. Pop dari stack, simpan di variable A, maka isi Stack menjadi : [5] dan variable A: -4
   b. Pop dari stack, simpan di variable B, maka isi Stack menjadi : [ ] dan variable B: 5
   c. Hitung B OPR A sehinga “5 + -4”, menghasilkan “1”.
   d. Tambahkan ke stack, sehingga isi stack menjadi [1]
8) Dapat “)”. Karena dapat sentinel maka pop dari stack (asumsikan disimpan di variable HASIL), maka variable HASIL bernilai 1.
9) Selesai.

PERHITUNGAN NOTASI POSTFIX

Contoh 2 :

Notasi Infix

7+(9-3)/(2+4)*10^(8-6)+(5-1)

Notasi Postfix

7 9 3 - 2 4 + / 10 8 6 - ^ * + 5 1 - +

PERHITUNGAN NOTASI POSTFIX

Contoh 2:

P= 7 9 3 – 2 4 + / 10 8 6 – ^ * + 5 1 – +

Proses perhitungan :

1. Tambahkan “)” ke akhir P

   P= 7 9 3 – 2 4 + / 10 8 6 – ^ * + 5 1 – +)

2. Scan elemen dari P

P	Keterangan	Stack
7	Operan, maka push ke stack.	7
9	Operan, maka push ke stack.	7 9
3	Operan, maka push ke stack.	7 9 3

PERHITUNGAN NOTASI POSTFIX

P	Keterangan	Stack
-	Operator (OPR), maka Pop stack A = 3, Pop stack B = 9, Hitung B OPR A => 9 - 3 = 6, Push 6 ke Stack	7 6
2	Operan, maka push ke stack.	7 6 2
4	Operan, maka push ke stack.	7 6 2 4
+	Operator (OPR), maka Pop stack A = 4, Pop stack B = 2, Hitung B OPR A => 2 + 4 = 6, Push 6 ke Stack	7 6 6

PERHITUNGAN NOTASI POSTFIX

P	Keterangan	Stack
/	Operator (OPR), maka Pop stack A = 6, Pop stack B = 6, Hitung B OPR A => 6 / 6 = 1, Push 1 ke Stack	7 1
10	Operan, maka push ke stack.	7 1 10
8	Operan, maka push ke stack.	7 1 10 8
6	Operan, maka push ke stack.	7 1 10 8 6
-	Operator (OPR), maka Pop stack A = 6, Pop stack B = 8, Hitung B OPR A => 8 - 6 = 2, Push 2 ke Stack	7 1 10 2

PERHITUNGAN NOTASI POSTFIX

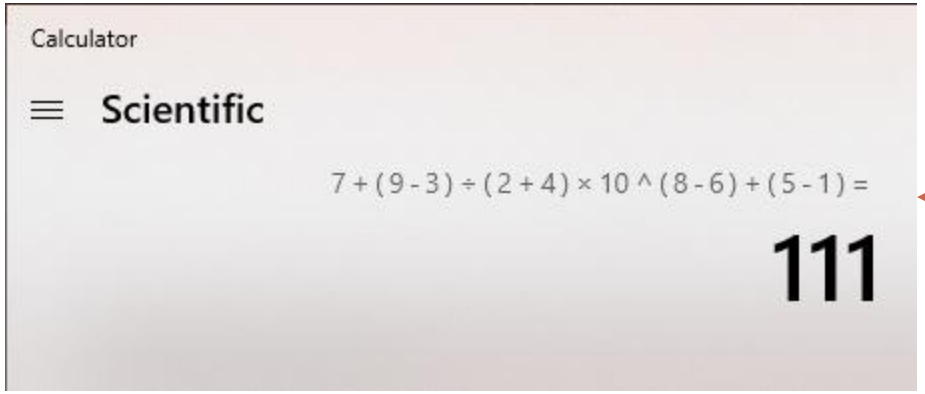
P	Keterangan	Stack
^	Operator (OPR), maka Pop stack A = 2, Pop stack B = 10, Hitung B OPR A => 10 ^ 2 = 100, Push 100 ke Stack	7 1 100
*	Operator (OPR), maka Pop stack A = 100, Pop stack B = 1, Hitung B OPR A => 100 * 1 = 100, Push 100 ke Stack	7 100
+	Operator (OPR), maka Pop stack A = 100, Pop stack B = 7, Hitung B OPR A => 7 + 100 = 107, Push 107 ke Stack	107

PERHITUNGAN NOTASI POSTFIX

P	Keterangan	Stack
5	Operan, maka push ke stack.	107 5
1	Operan, maka push ke stack.	107 5 1
-	Operator (OPR), maka Pop stack A = 1, Pop stack B = 5, Hitung B OPR A => 5 - 1 =, Push 4 ke Stack	107 4
+	Operator (OPR), maka Pop stack A = 4, Pop stack B = 107, Hitung B OPR A => 107 + 4 = 111, Push 111 ke Stack	111

PERHITUNGAN NOTASI POSTFIX

P	Keterangan	Stack
)	**Sentinel,** **Pop dari stack, simpan di variable HASIL = 111**	



LATIHAN

Buatkan bentuk notasi infix, notasi postfix dan hasil perhitungan dari rumus berikut

1. Rumus 1 :

$$Q = 5 + \left(\frac{2+3}{5-2} + 7\right) - 9$$

2. Rumus 2 :

$$Q = 5.5 * \frac{3+2.5}{11} - \frac{2(6-1)3^2}{9-4}$$

SEKIAN

Ada pertanyaan?

Silahkan tanyakan melalui Group Whatsapp, email, LMS.

FORUM DISKUSI

LMS UNIKOM
https://lms.unikom.ac.id

Group Whatsapp Perkuliahan

Youtube Comments
https://unikom.id/YTCStrukturData