

Laporan Tugas Program TBA

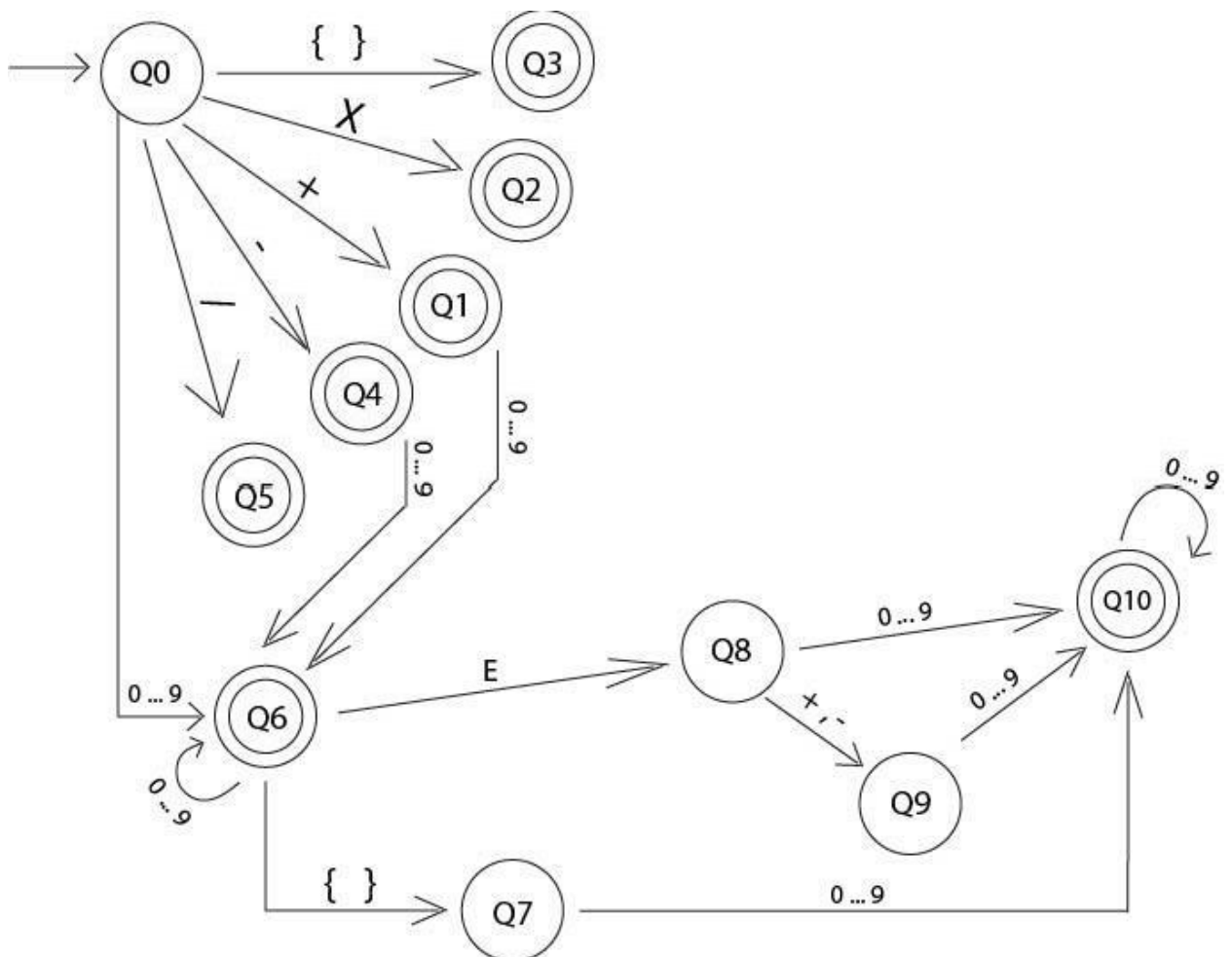
1. Kelompok:

No	NIM	Nama

2. Spesifikasi Program

Program yang akan dijalankan pada parser sederhana untuk ekspresi aritmatika hanya terbatas pada kemampuan operator aritmatika +, -, x. Serta penulisan pada program yang akan dijalankan haruslah sesuai dengan ekspresi matematika yang ada.

3. Rancangan Finite Automata



Tabel Transisi	+	-	X	/	{,}	0 ... 9	E	+,-
{q0}	{q1}	{q4}	{q2}	{q5}	{q3}	{q6}	{}	{}
{q1}	{}	{}	{}	{}	{}	{q6}	{}	[]
{q2}	{}	{}	{}	{}	{}	[]	{}	{}
{q3}	{}	{}	{}	{}	{}	{}	{}	{}
{q4}	{}	{}	{}	{}	{}	[q6]	{}	{}
{q5}	{}	{}	{}	{}	{}	{}	{}	{}
{q6}	{}	{}	{}	{}	{q7}	{q6}	{q8}	{}
{q7}	{}	{}	{}	{}	{}	{q10}	{}	{}
{q8}	{}	{}	{}	{}	{}	{q10}	{}	{q9}
{q9}	{}	{}	{}	{}	{}	{q10}	{}	{}
{q10}	{}	{}	{}	{}	{}	{q10}	{}	{}

$M = \{Q, \Sigma, A, F\}$

- $Q = \{q1, q2, q3, q4, q5, q6, q7, q8, q9, q10\}$
- $\Sigma = \{+, -, x, /, \{, \}, 0..9, E\}$
- $A = \{q0\}$
- $F = \{q1, q2, q3, q4, q5, q6, q10\}$

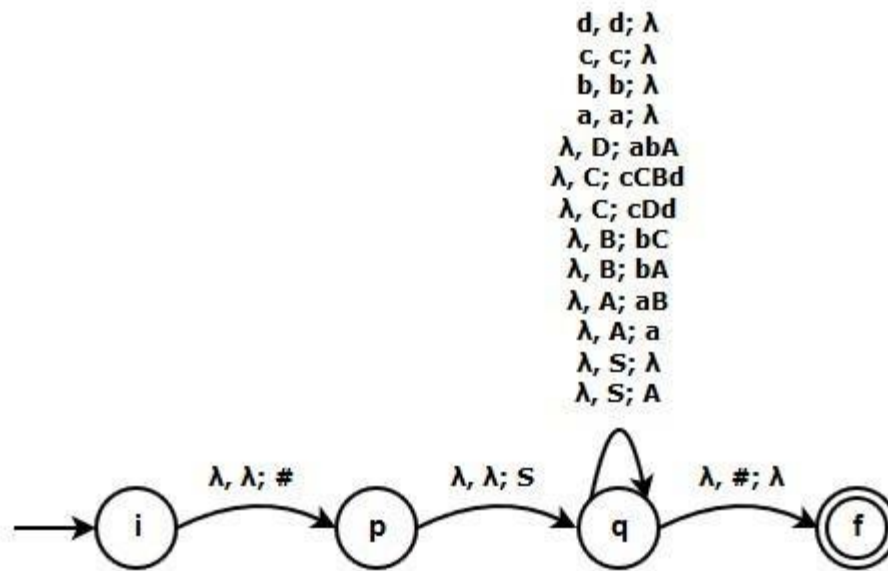
4. Rancangan Context Free Grammar dan Pushdown Automata

a. Rancangan Context Free Grammar

$a : \text{num}$
 $b : \text{opr } (+, -, /, *)$
 $c : \text{kurbuka } ($
 $d : \text{kurtutup })$
 $S \rightarrow A | \epsilon$
 $A \rightarrow a | aB$
 $B \rightarrow bA | bC$
 $C \rightarrow cDd | cCBd$
 $D \rightarrow abA$

b. Rancangan Pushdown Automata

$\Sigma = \{a, b, c, d\}$
 $\Gamma = \{a, b, c, d, S, A, B, C, D, \#\}$
 $S = \{i, p, q, f\}$



5. Cara Kerja Program

Tahap 1

Alur kerja program parser sederhana untuk ekspresi matematika :

1. Terdapat variabel status yang di inisialisasikan dengan nilai boolean true.
2. Input dari user berupa ekspresi matematika.
3. Input dari user akan disimpan ke dalam sebuah pita karakter.
4. String dari user akan disalin ke sebuah array per angka atau simbol.
5. Setelah proses input selesai akan terjadi perulangan sampai pada array terakhir yang disalinkan string dari pita karakter sebelumnya untuk melakukan pengecekan. Perulangan hanya dilakukan apabila nilai variable status true.
6. Pengecekan pada program ini dilakukan berdasarkan angka pertama dari setiap string dimana terjadi kemungkinan angka-angka selanjutnya sesuai dengan operasi pada logika preposisi. Pengecekan dapat dijabarkan sebagai berikut :
 - a. Jika isi array pertama adalah 0 sampai 9 atau -, + maka akan dilakukan pengecekan ke array setelahnya.
 1. Apabila isi array selanjutnya adalah 0 sampai 9 atau E atau , maka akan dilakukan pengecekan ke array selanjutnya.
 - ▢ Jika isi array selanjutnya kosong atau spasi maka nilai status akan berubah TRUE. Dan output 'num'.
 2. Apabila isi array selanjutnya adalah 0 sampai 9 atau E atau , maka akan dilakukan pengecekan ke array selanjutnya.
 - ▢ Jika isi array kosong atau spasi maka nilai status akan berubah TRUE. Dan output 'num'.

3. Apabila isi array terakhir adalah kosong atau spasi, maka di outputkan 'num', dan nilai statusnya TRUE.
- b. Jika isi pada array pertama = '+' atau '-' atau 'x' atau ';' maka dilakukan pengecekan ke array setelahnya.
 1. Apabila isi array selanjutnya spasi maka akan di outputkan 'opr' dan nilai status di set TRUE.
 - ▢ Jika isi array selanjutnya selain spasi maka status berubah menjadi FALSE, dan di output 'error'.
- c. Jika isi pada array pertama = '(' maka dilakukan pengecekan ke array setelahnya.
 1. Apabila isi array selanjutnya spasi maka akan di outputkan 'opr' dan nilai status di set TRUE.
 - ▢ Jika isi array selanjutnya selain spasi maka status berubah menjadi FALSE, dan di output 'error'.
- d. Jika isi pada array pertama = ')' maka dilakukan pengecekan ke array setelahnya.
 1. Apabila isi array selanjutnya spasi maka akan di outputkan 'opr' dan nilai status di set TRUE.
 - ▢ Jika isi array selanjutnya selain spasi maka status berubah menjadi FALSE, dan di output 'error'.

Tahap 2

Alur kerja program :

1. Program meminta inputan berupa string.
2. String inputan diubah menjadi char untuk pengecekan.
3. Inputan akan di periksa sesuai dengan kondisi yang ada sampai string habis atau kondisi tidak terpenuhi :
 - 1) Kondisi Z, jika inputan :
 - ▢ Kali atau bagi atau tanda kurung, maka akan pindah ke kondisi G
 - ▢ Tambah atau kurang, maka akan pindah ke kondisi A
 - ▢ Angka, maka pindah ke kondisi B
 - ▢ Spasi, maka akan tetap di kondisi Z
 - ▢ Untuk yang lainnya akan error
 - 2) Kondisi A, jika inputan selanjutnya :
 - ▢ Angka, maka pindah ke kondisi B
 - ▢ Spasi, maka akan ke kondisi Z dan output opr
 - ▢ Untuk yang lainnya akan error
 - 3) Kondisi B, jika inputan selanjutnya :
 - ▢ Angka, maka akan tetap di kondisi B
 - ▢ Koma, maka akan pindah ke kondisi C
 - ▢ E, maka pindah ke kondisi D

- ▮ Spasi, maka akan ke kondisi Z dan output num
- ▮ Untuk yang lainnya akan error
- 4) Kondisi C, jika inputan selanjutnya :
 - ▮ Angka, maka akan tetap di kondisi C
 - ▮ Spasi, maka akan ke kondisi Z dan output num
 - ▮ Untuk yang lainnya akan error
- 5) Kondisi D, jika inputan selanjutnya :
 - ▮ Angka, maka pindah ke kondisi F
 - ▮ Tambah atau kurang, maka akan pindah ke kondisi E
 - ▮ Untuk yang lainnya akan error
- 6) Kondisi E, jika inputan selanjutnya :
 - ▮ Angka, maka pindah ke kondisi F
 - ▮ Untuk yang lainnya akan error
- 7) Kondisi F, jika inputan selanjutnya :
 - ▮ Spasi, maka akan ke kondisi Z dan output num
 - ▮ Angka, maka tetap di kondisi F
 - ▮ Untuk yang lainnya akan error
- 8) Kondisi G, jika inputan selanjutnya :
 - ▮ Spasi, maka akan ke kondisi Z. Jika setelah spasi inputannya :
 - ▮ Kali atau bagi, maka akan mengoutput opr
 - ▮ Kurung buka, maka akan mengoutput kurbuk
 - ▮ Kurung tutup, maka akan mengoutput kurtup
 - ▮ Untuk yang lainnya akan error

6. Pengujian Program

Tahap 1

Input	Output Program	Keterangan
2 + 3 x 10	num opr num opr num	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Program membaca tiap notasi dan dapat menghasilkan output-an.
-3 x (7 + 10)	num opr kurbuka num opr num kurtutup	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Program membaca tiap notasi dan dapat menghasilkan output-an.
3 x 7 % 2 (2 + 3)	num opr num error	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Terjadi error karena pembacaan pada "%" bukan merupakan string lexic yang

		dapat dikenali pada input ekspresi.
$97 \times 5 + 19$ _____	num opr num opr num	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Program membaca tiap notasi dan dapat menghasilkan output-an.
$30 + 5 (\% 4 \times 2)$	num opr num kurbuka error	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Terjadi error karena pembacaan pada “%” bukan merupakan string lexic yang dapat dikenali pada input ekspresi.

Tahap 2

Input	Output Program	Keterangan
$2 + 3 \times 10$	num opr num opr num VALID	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Program membaca tiap notasi dan dapat menghasilkan output-an.
$-3 \times (7 + 10)$	num opr kurbuka num opr num kurtutup VALID	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Program membaca tiap notasi dan dapat menghasilkan output-an.
$3 \times 7 \% 2 (2 + 3)$	num opr num error TIDAK VALID	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Terjadi error karena pembacaan pada “%” bukan merupakan string lexic yang dapat dikenali pada input ekspresi.
$97 \times 5 + 19$	num opr num opr num VALID	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Program membaca tiap notasi dan dapat menghasilkan output-an.
$30 + 5 (\% 4 \times 2)$	num opr num kurbuka error TIDAK VALID	Sesuai dengan yang seharusnya dengan contoh program yang tertera. Terjadi error karena pembacaan pada “%” bukan merupakan string lexic yang dapat dikenali pada input ekspresi.