

PERTEMUAN 13

PUSHDOWN AUTOMATA (PDA)

A. TUJUAN PEMBELAJARAN

Pada bab ini akan dibahas mengenai definisi, pola transisi hingga bagaimana proses input string untuk PDA. Setelah mengikuti perkuliahan dari bab ini, mahasiswa diharapkan mampu :

- 1) Menjabarkan definisi dari PDA
- 2) Menjelaskan transisi PDA
- 3) Menganalisa proses input string ke PDA

B. URAIAN MATERI

1. Definisi Push down Automata (PDA)

Push Down Automata (PDA) merupakan mesin otomata dari bahasa bebas konteks. PDA di gambarkan sebagai tempat penyimpanan yang tidak terbatas berupa stack. Mengulang kembali konsep dari stack dalam struktur data, bahwa stack merupakan suatu tumpukan data dengan konsep utamanya LIFO (Last In First Out). Artinya LIFO adalah data yang pertama dikeluarkan dari suatu stack adalah data yang terakhir masuk. Misalnya kita dapat melihat pada tumpukan suatu piring, jika ingin mengambil atau mengeluarkan piring dari tumpukannya maka seharusnya mengambil piring yang paling atas yakni piring yang terakhir dimasukkan dalam tumpukan.

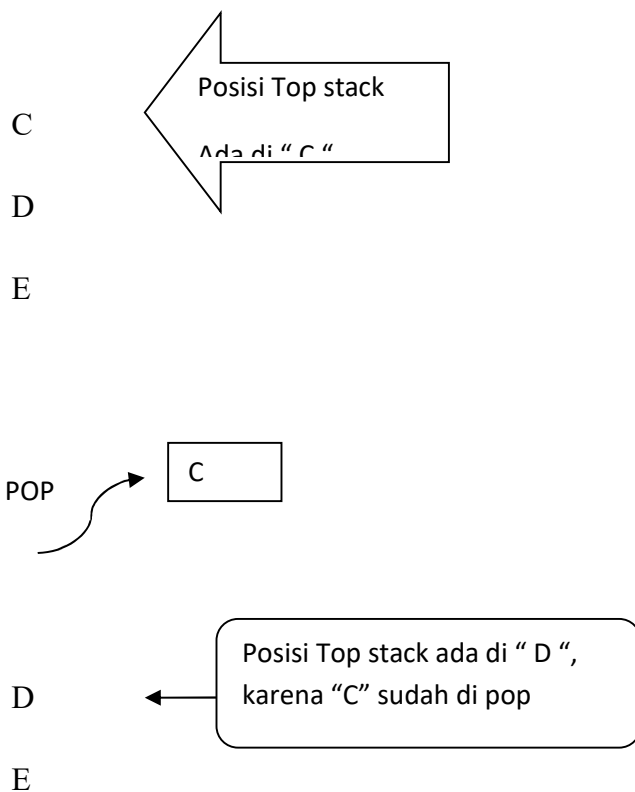
Dalam konsep stack inilah kita juga mengenal operasi “push” maupun “pop” suatu data dalam stack, juga selalu memiliki top of stack atau posisi paling atas dari stack. Setiap elemen akan masuk ke dalam stack dengan metode “push” dan elemen akan keluar dari stack menggunakan metode “pop”.

Dengan asosiasi terhadap konsep stack maka mekanisme mesin PDA mampu menyimpan sejumlah informasi yang proses penyimpanannya dilakukan pada saat pembacaan simbol-simbol di bagian depan string, dan dapat dipergunakan pada saat pembacaan simbol -simbol berikutnya dari string. Artinya kemampuan mesin meningkat karena meningkatnya kemampuan mengenali suatu bahasa. Agar kita lebih memahami konsepnya maka perhatikan contoh berikut :

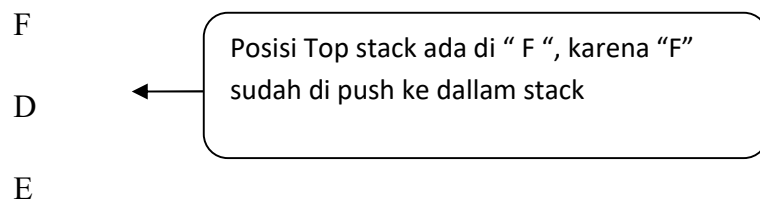
Contoh : jika pada mesin otomata berhingga atau finite state automata mampu menerima bahasa regular yang mempunyai batasan aturan produksi untuk setiap sel kanannya hanya boleh dengan pola non terminal (ingat : alfabet huruf besar) diikuti terminal (alfabet huruf kecil), sedangkan jika sudah terminal maka tidak boleh untuk ditambahkan terminal kembali misal suatu bahasa $L = \{c^i d^j \mid i, j \geq 0\}$, maka jika setelah simbol diproses maka selanjutnya tidak boleh ada simbol c. Sedangkan bahasa $L = \{c^i d^j \mid i \geq 0, j \geq 0\}$, yakni mampu membaca walaupun d sudah dibaca sebelumnya, karena dengan stack yang mempunyai penyimpanan yang tidak terbatas kapasitasnya. Sehingga alfabet c mampu disimpan dan kemudian dipasangkan dengan setiap proses pembacaan d. Jika seluruh simbol dalam string telah dibaca dan tersisa simbol c dalam stack atau kondisi stack kosong maka string diterima.

Sebelum melanjutkan ke definisi dari PDA, marilah kita lihat contoh stack, agar mudah memahami konsep penerapannya.

Contoh stack



Jika dilakukan operasi push B, maka kondisi stack akan menjadi :



Gambar 7.1 Gambar Contoh Stack dari PDA

Selanjutnya dapat kita lanjutkan untuk melakukan definisi dari suatu PDA berdasarkan 7 tuple, yakni :

$M = (Q, \Sigma, q^0, F, \delta, \Gamma, Z_0)$, dimana :

Q : himpunan hingga state,

Σ : alfabet input,

Γ : alfabet/simbol *stack*,

q_0 : state awal, $q^0 \in Q$

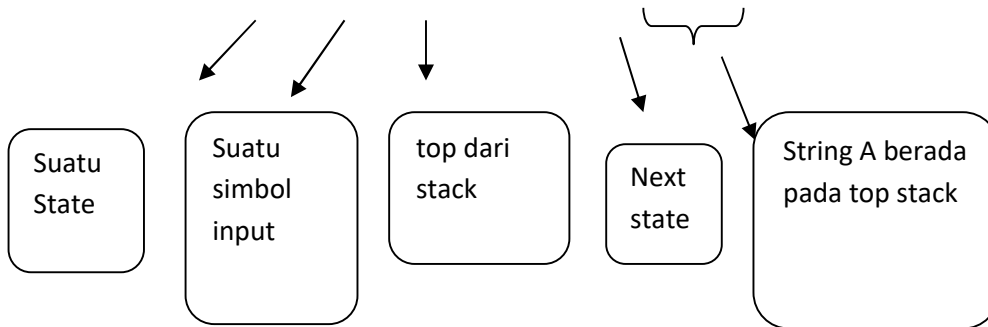
Z_0 : simbol awal *stack*, $Z^0 \in \Gamma$

F : himpunan state penerima, $F \subseteq Q$

δ : fungsi transisi , $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ (himpunan bagian dari $Q \times \Gamma^*$)

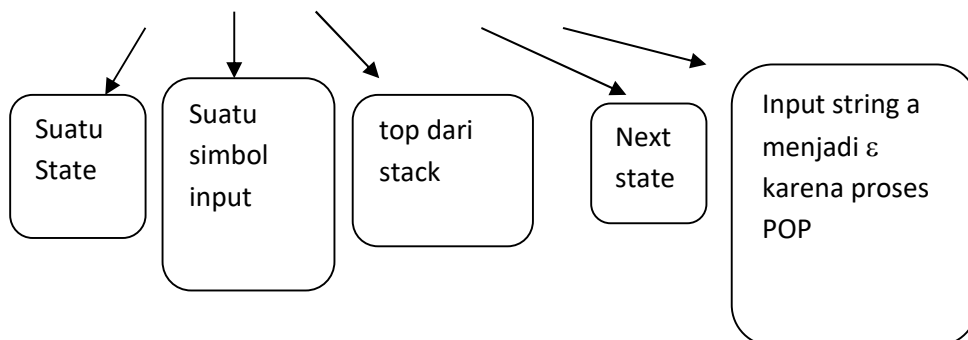
Untuk fungsi Push pada PDA :

$$\delta(q_0, a, Z_0) = (q_0, AZ_0) \text{ Push/insert}$$



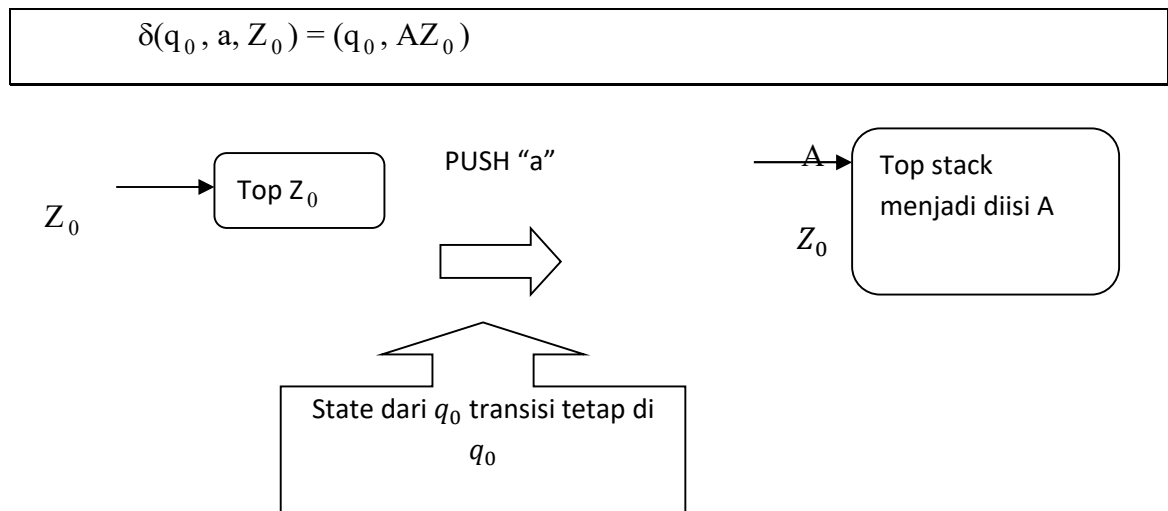
Gambar 7.2 penjelasan fungsi transisi PDA untuk kondisi PUSH/INSERT

$$\delta(q_0, a, A) = (q_1, \epsilon) \text{ Pop/delete}$$



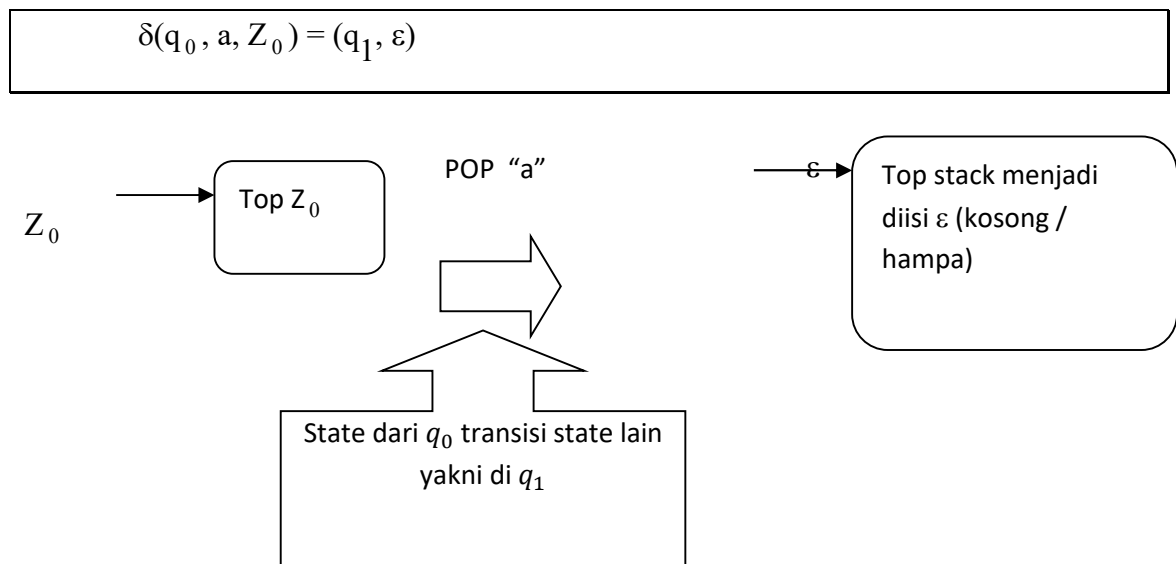
Gambar 7.3 penjelasan fungsi transisi PDA untuk kondisi POP/DELETE

Setelah kita lihat pada gambar 7.1 dan 7.2 di atas, kondisi suatu PDA untuk setiap transisi yang mengalami “push” atau pun “pop”. Maka proses Push pada PDA akan selalu menghasilkan suatu top stack dengan string yang dimasukkan pada PDA. Seperti gambar berikut :



Gambar 7.4 simulasi dari stack PDA dengan fungsi PUSH

Pada gambar 7.3, kita akan melihat suatu simulasi untuk contoh mengisi ke suatu stack dengan input string “a” berdasarkan transisi $\delta((q_0, a, Z_0) = (q_0, AZ_0)$, posisi top stack Z_0 akan



Gambar 7.5 simulasi dari stack PDA dengan fungsi POP

Jika kita amati gambar 7.4 pada fungsi POP yakni selalu akan menghasilkan ϵ dengan posisi state berpindah ke state tertentu. Dengan asumsi dalam stack tersebut posisi hanya ada top di Z_0 dan tidak ada string lainnya, maka saat diinputkan suatu string misal “a” maka stack akan menjadi kosong atau diisi ϵ .

2. Transisi Pada PDA

PDA memiliki 2 jenis transisi, yaitu Δ yang merima simbol input, simbol top of stack, dan state. Setiap pilihan terdiri dari state berikutnya dan simbol- simbol. Penggantian isi stack dilakukan dengan operasi push dan pop. Jenis transisi yang kedua adalah transisi ϵ . Transisi ϵ tidak melakukan pembacaan input namun hanya menerima simbol top of stack dan state. Transisi ini memungkinkan PDA untuk memanipulasi isi stack dan berpindah antar state tanpa membaca input.

Setelah mengenal definisi dari suatu PDA, maka selanjutnya kita akan mengenal pola transisi dari transisi PDA dengan menerapkan suatu contoh pada tabel berikut berdasarkan buku *John C Martin* p.168, yakni PDA :

Contoh PDA Deterministik

$M1 = (Q, \Sigma, \Gamma, q_0, Z_0, F, \delta)$ dengan

$Q = \{q_0, q_1, q_2, q_3\}$

$F = \{q_0, q_3\}$

Move Number	State	Input	Stack Symbol	Move(s)
1	q_0	a	Z_0	(q_1, aZ_0)
2	q_1	a	a	(q_1, aa)
3	q_1	b	a	(q_2, Λ)
4	q_2	b	a	(q_2, Λ)
5	q_2	Λ	Z_0	(q_3, Z_0)
(all other combinations)				none

Tabel 7.1

contoh tabel transisi dari suatu PDA

Catatan : simbol Λ = simbol ε = hampa/kosong

Berdasarkan tabel 7.1, dapat kita buat transisi PDA secara formal berdasarkan *move number* atau no transisi :

- 1) $\delta(q_0, aZ_0) = (q_1, aZ_0)$
- 2) $\delta(q_1, aa) = (q_1, aa)$
- 3) $\delta(q_1, ba) = (q_2, \Lambda)$
- 4) $\delta(q_2, ba) = (q_2, \Lambda)$
- 5) $\delta(q_2, \Lambda) = (q_3, Z_0)$

Pada transisi di atas, mesin M1 state penerima berada pada state awal $\{q_0\}$ dan state $\{q_3\}$ artinya M1 akan selalu menerima pada kondisi awal. Proses pertama menerima input string “a” yang akan membuat mesin M1 akan bertransisi ke q_1 serta “a” akan PUSH ke dalam stack. Saat proses input string “b” state akan berpindah ke q_2 dan POP string yang ada di top stack. String input diterima ketika “b” mencapai no transisi ke 4 dimana dari state q_2 akan bernilai hampa Λ pada stack, sehingga jumlah b dan jumlah a sama. Pada saat itu, M1 membuat transisi Λ untuk keadaan menerima lainnya, serta tidak ada langkah yang mungkin.

Contoh transisi lainnya adalah sebagai berikut :

No.	Stata	Input	TopStack	Hasil
1	q_0	a	Z_0	(q_0, aZ_0)
2	q_0	b	Z_0	(q_0, bZ_0)
3	q_0	a	a	(q_0, aa)
4	q_0	b	a	(q_0, ba)
5	q_0	a	b	(q_0, ab)
6	q_0	b	b	(q_0, bb)

No.	Stata	Input	TopStack	Hasil
7	q_0	c	Z_0	(q_1, Z_0)
8	q_0	c	a	(q_1, a)
9	q_0	c	b	(q_1, b)
10	q_1	a	a	(q_1, ε)
11	q_1	b	b	(q_1, ε)
12	q_1	ε	Z_0	(q_2, ε)

Tabel 7.2 contoh tabel transisi PDA Deterministik

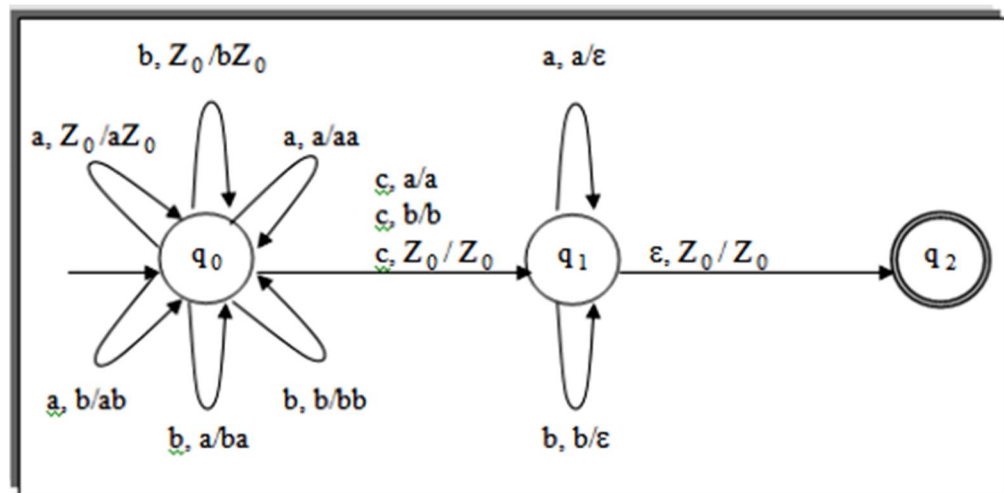
Definisi mesin PDA sesuai transisi pada tabel 7.2 adalah PDA $M_2 = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ pengenalan palindrome (Palindrome adalah sebuah kata, frasa, angka maupun susunan lainnya yang dapat dibaca dengan sama baik dari depan maupun belakang (spasi antara huruf-huruf biasanya diperbolehkan). Kata "palindrom" berasal

dari bahasa Yunani: palin ("balik") dan dromos ("pacuan kuda") berdasarkan wikipedia.co.id).

$L = \{xcx^T \mid x \in (a|b)^*\}$, dimana x^T adalah cermin(x), mempunyai tuple : $Q = \{q_0, q_1, q_2\}$, $F = \{q_2\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{a, b, Z_0\}$.

Sebagai contoh, perhatikan bahwa fungsi transisi No. 1 dapat dinyatakan sebagai : $\delta(q_0, a, Z_0) = (q_0, aZ_0)$. Pada tabel transisi tersebut terlihat bahwa pada state q_0 PDA akan melakukan PUSH jika mendapat input a atau b dan melakukan transisi state ke state q_1 jika mendapat input c. Pada state q_1 PDA akan melakukan POP.

Gambar transisi dapat dilihat sebagai berikut :



Gambar 7.6 diagram transisi sesuai tabel transisi PDA Deterministik

Notasi $(p, ay, X\beta) \Rightarrow (q, y, \gamma\beta)$ dapat diperluas menjadi : $(p, x, \alpha) \Rightarrow^* (q, y, \beta)$, yang berarti konfigurasi (q, y, β) dicapai melalui sejumlah (0 atau lebih) transisi. Ada dua cara penerimaan sebuah kalimat oleh PDA, yang masing-masing terlihat dari konfigurasi akhir, sebagaimana penjelasan berikut :

Jika $M_2 = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, A)$ adalah PDA dan $x \in \Sigma^*$, maka x diterima dengan state akhir (*accepted by final state*) oleh PDA M jika : $(q_0, x, Z_0) \Rightarrow^* (q, \epsilon, \alpha)$ untuk $\alpha \in \Gamma^*$ dan $q \in F$. x diterima dengan stack hampa (*accepted by empty stack*) oleh PDA M_2 jika : $(q_0, x, Z_0) \Rightarrow^* (q, \epsilon, \epsilon)$ untuk $q \in Q$

Contoh PDA Non Deterministik

PDA $M_3 = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, A)$ pengenalan palindrome $L = \{xx^T \mid x \in (a|b)^*\}$ mempunyai komponen tuple berikut : $Q = \{q_0, q_1, q_2\}$, $F = \{q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, Z_0\}$, dan fungsi transisi δ terdefinisi melalui tabel berikut :

No.	St.	In.	TS	Hasil	No.	St.	In.	TS	Hasil
1	q_0	a	Z_0	$(q_0, aZ_0), (q_1, Z_0)$	7	q_0	ϵ	Z_0	(q_1, Z_0)
2	q_0	b	Z_0	$(q_0, bZ_0), (q_1, Z_0)$	8	q_0	ϵ	a	(q_1, a)
3	q_0	a	a	$(q_0, aa), (q_1, a)$	9	q_0	ϵ	b	(q_1, b)
4	q_0	b	a	$(q_0, ba), (q_1, a)$	10	q_1	a	a	(q_1, ϵ)
5	q_0	a	b	$(q_0, ab), (q_1, b)$	11	q_1	b	b	(q_1, ϵ)
6	q_0	b	b	$(q_0, bb), (q_1, b)$	12	q_1	ϵ	Z_0	(q_2, ϵ)

Tabel 7.3 transisi PDA non deterministik

Pada tabel transisi tersebut terlihat bahwa pada state q_0 PDA akan melakukan PUSH jika mendapat input a atau b dan melakukan transisi state ke state q_1 jika mendapat input ϵ . Pada state q_1 PDA akan melakukan POP. Contoh 14 dan 15 menunjukkan bahwa PDA dapat dinyatakan sebagai mesin PUSH – POP.

3. Analisa Proses Input String PDA

Saat melakukan proses input string ke PDA perlu diperhatikan tahap – tahapannya sehingga dapat diterima atau tidak dapat dianalisa dengan baik. Berikut ini kita akan membahas proses input string menggunakan mesin PDA yang ada dalam pembahasan transisi PDA sebelum nya.

Mari kita coba menganalisa input string **aabb** ke mesin PDA dengan transisi pada tabel 7.1, berikut ini tahap input string :

$$\begin{aligned}
 (q_0, aabb, Z_0) &\Rightarrow (q_1, abb, aZ_0) \text{ (no transisi 1)} \\
 &\Rightarrow (q_1, bb, aaZ_0) \text{ (no transisi 2)} \\
 &\Rightarrow (q_2, b, aZ_0) \text{ (no transisi 3)} \\
 &\Rightarrow (q_2, \Lambda, Z_0) \text{ (no transisi 4)}
 \end{aligned}$$