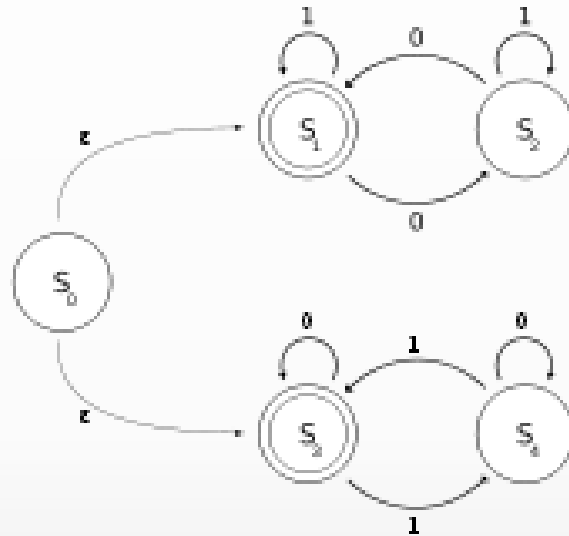


# Otomata dan Teori Bahasa Formal



Dr. Hanny Hikmayanti, M.Kom

Hilda Yulia Novita, M.Kom

✉ : hanny.hikmayanti@ubpkarawang.ac.id

✉ : hilda.yulia@ubpkarawang.ac.id

☎ : +62 857 1414 4026



**PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS BUANA PERJUANGAN  
KARAWANG**

# Push Down Automata (PDA)

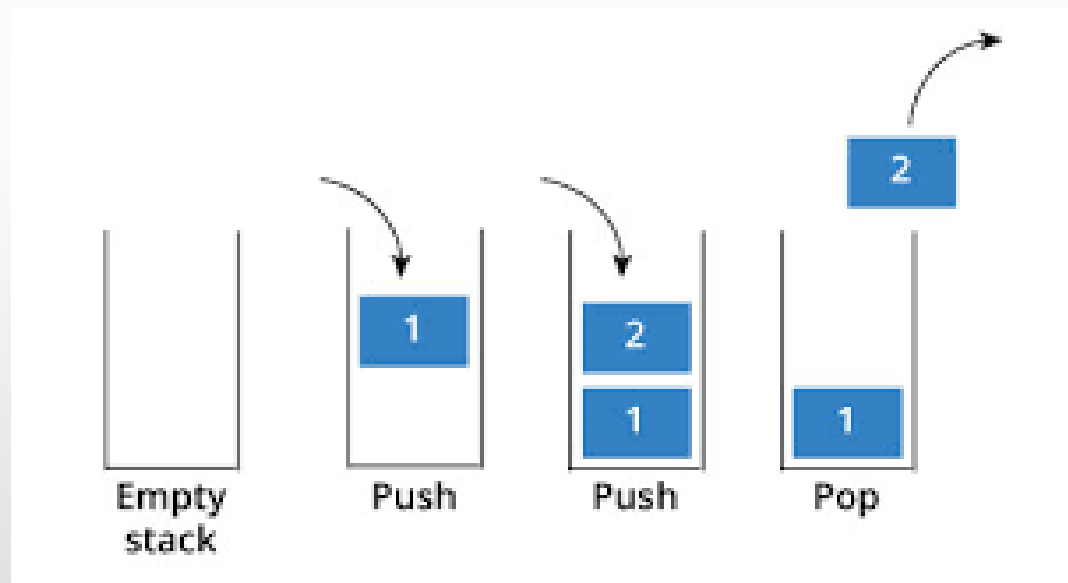
## Video 12.1

- Push Down Automata ( PDA) merupakan mesin otomata dari Tata Bahasa Bebas Konteks.
- FSA punya kemampuan “memory” terbatas sedangkan PDA didefinisikan sebuah tempat penyimpanan yang tidak terbatas berupa stack/ tumpukan

- Stack ialah kumpulan dari elemen-elemen sejenis dengan sifat penambahan elemen dan pengambilan elemen melalui suatu tempat yang disebut top of stack ( puncak stack).
- Aturan pengisian dalam stack adalah LIFO ( Last In First Out)
- Pengambilan elemen dari stack dinyatakan dalam operasi POP.
- Memasukkan elemen dari stack dinyatakan dalam operasi PUSH.

# PDA

- Setiap elemen stack bisa memuat satu symbol, yang dlm PDA disebut sebagai Simbol Stack.



# DEFINISI

Sebuah PDA dinyatakan dalam 7 tuple

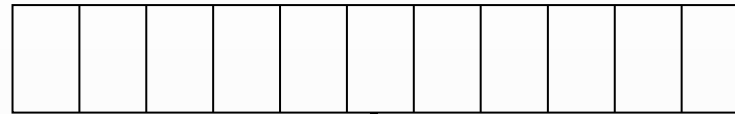
- $M = (Q, \Sigma, \Gamma, \Delta, S, F, Z)$

Dimana:

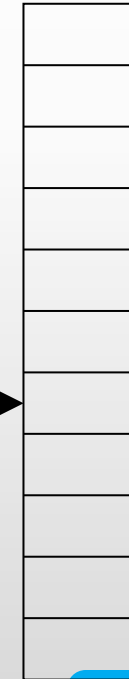
- $Q$  = himpunan state
- $\Sigma$  (Sigma) = himpunan simbol input
- $\Gamma$  (Gamma) = simbol-simbol tumpukan/stack
- $\Delta$  (Delta) = fungsi transisi
- $S$  = state awal ,  $F \in Q$
- $F$  = himpunan final state,  $F \in Q$
- $Z$  = simbol awal tumpukan/top of stack,  $Z \in \Gamma$

# PDA

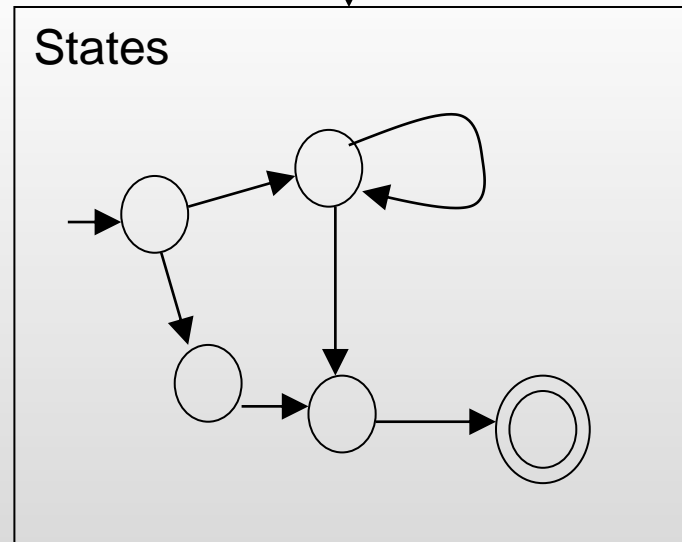
Input String



Stack



States



- PDA bisa pula dianggap sebagai FSA yang dilengkapi stack.
- Sebuah PDA yang menerima suatu input selain bisa berpindah state juga bisa melakukan operasi pada stack.
- Kondisi / konfigurasi PDA pada suatu saat akan dinyatakan dengan state dan stack.

# Pada PDA terdapat 2 jenis transisi ( $\Delta$ )

## 1. Transisi dengan memakai suatu simbol input.

- Bergantung pada simbol input, simbol pada top stack, dan state, terdapat sejumlah pilihan yang mungkin.
- Setiap pilihan terdiri dari state berikutnya dan simbol- simbol (satu, atau beberapa, atau kosong) untuk mengganti simbol pada top stack.
- Penggantian simbol pada top stack bisa berupa push, untuk satu atau beberapa simbol, bisa berupa pop untuk simbol kosong.
- Setelah membuat pilihan, kemudian PDA membaca simbol input berikutnya



## 2. Transisi $\epsilon$

- Mirip dengan yang pertama, kecuali dilakukan tanpa membaca input.
- Transisi ini memungkinkan PDA memanipulasi isi stack atau berpindah state tanpa membaca simbol input.

## Sifat penerimaan bahasa:

- PDA null stack
- PDA final state



# Contoh PDA dengan final state: Penjelasan di **Video 12.2**

- $Q = \{q1, q2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{A, B, Z\}$
- $S = q1$
- $F = \{q2\}$
- $Z = Z$

dengan  $\Delta$

- $\Delta(q1, \varepsilon, Z) = \{(q2, Z)\}$
  - $\Delta(q1, a, Z) = \{(q1, AZ)\}$
  - $\Delta(q1, b, Z) = \{(q1, BZ)\}$
  - $\Delta(q1, a, A) = \{(q1, AA)\}$
  - $\Delta(q1, b, A) = \{(q1, \varepsilon)\}$
  - $\Delta(q1, a, B) = \{(q1, \varepsilon)\}$
  - $\Delta(q1, b, B) = \{(q1, BB)\}$
- Cek apakah string “abba” diterima oleh PDA tersebut?

# Contoh PDA dengan null stack:

## Penjelasan di **Video 12.3**

- $Q = \{q1, q2\}$
- $\Sigma = \{0, 1, 2\}$
- $\Gamma = \{Z, B, G\}$
- $S = q1$
- $F = \emptyset$
- $Z = Z$

dengan  $\Delta$  :

$$\Delta(q1, 0, Z) = \{(q1, BZ)\}$$

$$\Delta(q1, 0, B) = \{(q1, BB)\}$$

$$\Delta(q1, 0, G) = \{(q1, BG)\}$$

$$\Delta(q1, 2, Z) = \{(q2, Z)\}$$

$$\Delta(q1, 2, B) = \{(q2, B)\}$$

$$\Delta(q1, 2, G) = \{(q2, G)\}$$

$$\Delta(q2, 0, B) = \{(q2, \epsilon)\}$$

$$\Delta(q2, \epsilon, Z) = \{(q2, \epsilon)\}$$

$$\Delta(q1, 1, Z) = \{(q1, GZ)\}$$

$$\Delta(q1, 1, B) = \{(q1, GB)\}$$

$$\Delta(q1, 1, G) = \{(q1, GG)\}$$

$$\Delta(q2, 1, G) = \{(q2, \epsilon)\}$$

Cek untuk string "020" apakah diterima oleh PDA tsb?

# PDA vs CFG

Aturan untuk mengkonstruksi transisi:

1.  $\Delta(q_1, \varepsilon, Z) = \{(q_2, S, Z)\} \rightarrow$  untuk mempush simbol awal  $S$  ke stack
2.  $\Delta(q_2, \varepsilon, A) = \{(q_2, w) \mid A \rightarrow w \text{ adalah sebuah produksi dalam CFG tsb}\}$  untuk semua  $A$
3.  $\Delta(q_2, a, a) = \{(q_2, \varepsilon)\}$  untuk setiap simbol terminal untuk mempop pembandingan terminal yang sama
4.  $\Delta(q_2, \varepsilon, Z) = \{(q_3, Z)\}$  sukses, stack kosong dan sampai pada final state.

# PDA vs CFG

- Mesin ini dimulai dengan mem-*push* Z pada top-stack. Pada setiap Langkah berikutnya dilakukan **salah satu** dari 2 hal berikut:
  1. Jika top stack dari symbol tumpukan adalah suatu variable (misal A), kita menggantinya dengan dengan ruas kanan dari A, misal  $A \rightarrow w$ , maka kita ganti dengan w
  2. Jika top stack dari simbol tumpukan adalah sebuah terminal dan jika ia menyamai symbol masukan berikutnya, kita pop dari tumpukan



- CFG :

$$S \rightarrow aSa \mid bSb \mid c$$

Dapat dikonstruksi PDAnyanya:

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{S, a, b, c, Z\}$
- $S = q_1$
- $F = \{q_3\}$
- $Z = Z$

- dengan  $\Delta$  :

$$\Delta(q1, \varepsilon, Z) = \{(q2, SZ)\}$$

$$\Delta(q2, \varepsilon, S) = \{(q2, aSa), (q2, bSb), (q2, c)\}$$

$$\Delta(q2, a, a) = \Delta(q2, b, b) = \Delta(q2, c, c) = \{(q2, \varepsilon)\}$$

$$\Delta(q2, \varepsilon, Z) = \{(q3, Z)\}$$

- Cek apakah aturan produksi dan PDA tsb menurunkan string “aca”



PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS BUANA PERJUANGAN  
KARAWANG

# THANK YOU