LAPORAN TEORI BAHASA DAN OTOMATA

DOSEN PENGAMPU:

Novi Yusliani, S.Kom., M.T.

DISUSUN OLEH:

KELOMPOK 3

Azka Hukma Tsabita	(09021382328159)
Inayah Khofifah Danis	(09021382328141)
Saravina Zharfa Kelana Putri	(09021382328149)
Fransisca Stevanie Ekawati	(09021382328127)
Putri Alisya Zhafirah	(09021382328153)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2025

1. Grammar CFG (Context Free grammar)

D → xDy | yDx | xDx | yDy | ε

grammar CFG tersebut menggunakan "*Leftmost Derivation"*

- **Variabel (Non-Terminal)**

  D → adalah satu- satunya variabel dalam grammar ini, yang digunakan sebagai simbol awal.

- **Terminal (Simbol Alfabet)**

  terminal x dan terminal y

Aturan Produksi

- D → xDy
- D → yDx
- D → xDx
- D → yDy
- D → ε

2. Contoh String

1. w = xyxxyyxy

   D → xDy → x(yDx)y → x(y(xDy)x)y → x(y(x(xDy)y)x)y → x(y(x(xεy)y)x)y → xyxxyyxy

2. w = xyxy

   D → xDy → x(yDx)y → x(yεx)y → xyxyw = xxxy

3. w = xxxy

   D → xDy → x(xDx)y → x(xεx)y → xxxy

3. CFG ke PDA

D → xDy | yDx | xDx | yDy | ε

PDA = (Q, Σ, Γ, δ, q0, Z0, F)

- Q = { q}
- Σ = { x, y }
- Γ = { x, y, D }
- q: start state
- Initial stack = D

PDA = ({q}, {x,y}, {x,y,D}, δ, q, D)

δ :

- δ(q, ε ,D) = {(q, xDy), (q,y,Dx),(q, xDx),(q,yDy),(q, ε)}
- δ(q,x,x) = {(q, ε)}
- δ(q,y,y) = {(q, ε)}

4. CFG ke CNF

D → xDy | yDx | xDx | yDy | ε

1. ε-*Production*

   D → xDy | yDx | xDx | yDy | xy | yx | xx | yy

2. *Unit Production*

   Tidak memiliki *unit production* karena semua produksinya memiliki lebih dari satu simbol di sisi kanan atau mengandung simbol terminal. Unit production hanya terjadi jika sebuah non-terminal menghasilkan tepat satu non-terminal (misalnya A → B), sedangkan dalam grammar ini, setiap aturan produksi D melibatkan kombinasi terminal dan/atau lebih dari satu simbol, sehingga tidak memenuhi bentuk unit production.

3. *Useless symbol*
   - *Not Generating*: Tidak memiliki *Not Generating*
   - *Not Reachable*: Tidak memiliki *Not Reachable*

4. CNF (*Chomsky Normal Form*)

   Misal :

   A → x

   B → y

   C →DB

   E →DA

   D → xDy | yDx | xDx | yDy | xy | yx | xx | yy

   D → ADB | BDA | ADA | BDB | AB | BA | AA | BB

   D → AC | BE | AE | BC | AB | BA | AA | BB

5. Pembuktian

Code:

```
CFG = {
    "D": ["xDy", "yDx", "xDx", "yDy", ""]
}
```

```
start_symbol = 'D'
```

```
Kedalaman_Maksimum = 20  # Batas kedalaman rekursi
```

```
def left_most_derivation(CFG, start_symbol, string_target):
    hasil = []

    def derivasi(current, step_derivation, kedalaman):
        if kedalaman > Kedalaman_Maksimum:
            return
        string_flat = ''.join([c for c in current if c not in CFG])
        if string_flat == string_target:
            hasil.append(step_derivation + [''.join(current)])
            return
        if len(string_flat) > len(string_target):
            return
```

```
        for i, symbol in enumerate(current):
            if symbol in CFG:
                for produksi in CFG[symbol]:
                    baru = current[:i] + list(produksi) + current[i+1:]
                    derivasi(baru, step_derivation + [''.join(current)],
kedalaman + 1)
                break  # hanya derivasi simbol non-terminal pertama (kiri)

    derivasi([start_symbol], [], 0)
    return hasil
```

```
print("Periksa Leftmost Derivation")
while True:
    inputString = input("\nMasukkan string yang ingin diperiksa (atau
ketik 'selesai' untuk berhenti): ").strip()
    if inputString.lower() == 'selesai':
        print("Program dihentikan.")
        break

    hasil_derivasi = left_most_derivation(CFG, start_symbol, inputString)
    if hasil_derivasi:
        print(f"✅ String '{inputString}' DITERIMA oleh CFG. Contoh
derivasi:")
        for step_derivation in hasil_derivasi[0]:
            print(f" → {step_derivation}")
    else:
        print(f"❌ String '{inputString}' TIDAK DITERIMA oleh CFG.")
```

Output:

Periksa Leftmost Derivation

Masukkan string yang ingin diperiksa (atau ketik 'selesai' untuk berhenti): xyxxyyxy

✅ String 'xyxxyyxy' DITERIMA oleh CFG. Contoh derivasi:

→ D

→ xDy

→ xyDxy

→ xyxDyxy

→ xyxxDyyxy

Masukkan string yang ingin diperiksa (atau ketik 'selesai' untuk berhenti): xyxy

✅ String 'xyxy' DITERIMA oleh CFG. Contoh derivasi:

→ D

→ xDy

→ xyDxy

Masukkan string yang ingin diperiksa (atau ketik 'selesai' untuk berhenti): xxxy

✅ String 'xxxy' DITERIMA oleh CFG. Contoh derivasi:

→ D

→ xDy

→ xxDxy

Masukkan string yang ingin diperiksa (atau ketik 'selesai' untuk berhenti): xyxyxix

❌ String 'xyxyxix' TIDAK DITERIMA oleh CFG.

Masukkan string yang ingin diperiksa (atau ketik 'selesai' untuk berhenti): selesai

Program dihentikan.

link google colab: TBO.ipynb