

Tugas 1: Implementasi DFA pada Vending Machine

MII212202 - Bahasa dan Otomata

March 4, 2025

1 Deskripsi Tugas

Implementasikan sebuah DFA (Deterministic Finite Automaton) yang mensimulasikan aplikasi mesin penjual minuman kaleng (vending machine). Mesin ini menerima sejumlah uang dalam pecahan tertentu, memprosesnya melalui state transitions, dan menentukan apakah suatu produk bisa dibeli atau tidak.

Tugas dikerjakan per kelompok dengan anggota kelompok berjumlah 2 orang.

2 Spesifikasi Sistem

Sebuah vending machine menjual tiga jenis minuman:

- Minuman A (Rp3.000)
- Minuman B (Rp4.000)
- Minuman C (Rp6.000)

Berikut adalah aturan pembelian vending machine tersebut:

- Mesin hanya menerima uang pecahan Rp1.000, Rp2.000, Rp5.000, dan Rp10.000.
- Jika jumlah uang yang dimasukkan tepat, tombol ON pada minuman yang sesuai akan menyala.
- Jika uang yang dimasukkan lebih dari harga minuman, mesin akan menolak pembelian (jika tidak ada fitur pengembalian uang).
- Mesin memiliki fitur penukaran uang, di mana uang Rp2.000, Rp5.000, atau Rp10.000 dapat ditukarkan ke pecahan Rp1000.
- Jumlah maksimal nominal uang yang dapat dimasukkan adalah Rp10.000.
- (Spesifikasi bonus) Implementasikan fitur kembalian jika uang yang dimasukkan ke dalam mesin lebih dari harga minuman yang dipilih.

3 Instruksi Pengerjaan

1. Buat model DFA dalam vending machine dalam format `.txt` (detail format dapat dilihat pada contoh di bawah) sesuai dengan spesifikasi sistem. Gunakan file `vending_dfa.txt` sebagai konfigurasi DFA.
2. Buat implementasi DFA, bahasa pemrograman dibebaskan (C, C++, Python, Java, atau lainnya).
3. Tambahkan validasi untuk mencegah input yang tidak sesuai.
4. (Soal bonus) Tambahkan fitur pengembalian uang untuk transaksi dengan nominal lebih dari harga minuman.

4 Format Input dan Output

Program akan membaca file berformat `.txt` yang berisi deskripsi DFA.

Contoh isi file `vending_dfa.txt`:

```
States: S0, S1000, S2000, ...
Alphabet: 1000, 2000, ...
Start: S0
Accept: S3000, S4000, S6000
Transitions:
S0 1000 S1000
S0 2000 S2000
S0 5000 S5000
...
```

Kemudian, berdasarkan DFA yang telah dibaca, program akan dapat menerima uang dan mengeluarkan output sesuai dengan DFA yang telah dirancang.

Berikut adalah contoh input dan output program:

Contoh Input 1 (Pembelian berhasil dengan jumlah uang pas):

```
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): 2000
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): 1000
ON: Minuman A
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): A
```

Output:

```
Lintasan DFA: S0 → S2000 → S3000
Minuman A dapat dibeli. Status: ACCEPTED.
```

Contoh Input 2 (Pembelian gagal karena uang kurang):

```
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): 1000
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): 1000
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): B
```

Output:

```
Lintasan DFA: S0 → S1000 → S2000
Uang tidak cukup. Status: REJECTED.
```

Contoh Input 3 (Pembelian dengan jumlah uang lebih dari harga minuman, tanpa fitur kembalian):

```
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): 5000
ON: Minuman A, Minuman B
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): B
```

Output:

```
Lintasan DFA: S0 → S5000
Uang lebih dari harga minuman. Status: REJECTED.
```

Contoh Input BONUS (Pembelian dengan jumlah uang lebih dari harga minuman, dengan fitur kembalian):

```
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): 5000
ON: Minuman A, Minuman B
Masukkan uang atau beli minuman (1000, 2000, 5000, 10000, A, B, C): B
```

Output:

```
Lintasan DFA: S0 → S5000
Minuman A dapat dibeli. Status: ACCEPTED.
Kembalian: 1000
```

5 Deliverables

1. Dokumen laporan berisi:

- Deskripsi permasalahan
- Notasi DFA lengkap (states, alphabet, start state, accepting states, transition table, diagram)
- Penjelasan implementasi program
- Contoh masukan dan keluaran (screenshot)

- Detail kontribusi masing-masing anggota tim

2. Source code berisi:

- Kode program (.cpp, .py, .java, dll.)
- File konfigurasi DFA (.txt)
- File executable (.exe atau .out jika relevan)
- File README.md berisi tatacara atau petunjuk menjalankan program

Deadline: Selasa, **25 Maret 2025, 23:59**. Tugas dikumpulkan di Simaster.