

# Judul Proyek: Perancangan dan Implementasi Model Finite Automata pada Sistem Vending Machine Sederhana

## 1. Penjelasan Aturan Sistem (Business Logic)

Sistem ini mensimulasikan Vending Machine yang menerima input uang dan mengeluarkan produk, dengan batasan logika sebagai berikut:

- **Saldo Maksimal:** Mesin memiliki kapasitas penyimpanan saldo (state) maksimal sebesar Rp 20.000 (merepresentasikan keterbatasan memori mesin nyata).
- **Pecahan Uang:** Mesin hanya menerima input pecahan Rp 5.000 dan Rp 10.000.
- **Produk:** Terdapat 4 jenis produk dengan harga bervariasi (Rp 5.000, Rp 10.000, Rp 15.000, Rp 20.000).
- **Mekanisme Transaksi:**
  - Saldo bertambah saat uang dimasukkan.
  - Saldo berkurang saat produk dibeli (tanpa reset ke nol, memungkinkan pembelian beruntun).
  - Jika input uang melebihi batas maksimal state (Rp 20.000), uang ditolak.
- **Pengembalian (Refund):** Pengguna dapat menekan tombol '0' untuk membatalkan proses, mengambil sisa kembalian, dan mematikan mesin.

## 2. Penjelasan Model Mesin Otomata (Formal Definition)

Sistem dimodelkan menggunakan **Deterministic Finite Automata (DFA)** karena setiap state memiliki transisi yang pasti untuk setiap input yang diberikan.

Definisi formal DFA dinyatakan dalam 5-tuple:

$$M = (Q, \Sigma, \delta, S, F)$$

### A. Himpunan State (\$Q\$)

State merepresentasikan jumlah saldo yang tersimpan dalam memori mesin saat ini.

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$

- **\$q\_0\$:** Saldo 0
- **\$q\_1\$:** Saldo 5.000
- **\$q\_2\$:** Saldo 10.000
- **\$q\_3\$:** Saldo 15.000
- **\$q\_4\$:** Saldo 20.000 (Batas Maksimum)
- **\$q\_5\$:** State Akhir (Selesai/Keluar)

B. Himpunan Simbol Input (\$\Sigma\$)

Input dibagi menjadi Input Uang (\$U\$), Input Pilih Produk (\$P\$), dan Input Kontrol (\$K\$).

$\Sigma = \{ '5', '10', '1', '2', '3', '4', '0' \}$

- '5': Input uang Rp 5.000
- '10': Input uang Rp 10.000
- '1': Pilih Air Mineral (Harga 5.000)
- '2': Pilih Teh Botol (Harga 10.000)
- '3': Pilih Susu Kotak (Harga 15.000)
- '4': Pilih Kopi Kaleng (Harga 20.000)
- '0': Tombol Batal/Selesai

C. State Awal (\$S\$)

Mesin dimulai dengan saldo 0.

$S = q_0$

D. Himpunan State Akhir (\$F\$)

Mesin berhenti beroperasi ketika mencapai state ini.

$F = \{ q_5 \}$

E. Fungsi Transisi (\$\delta\$)

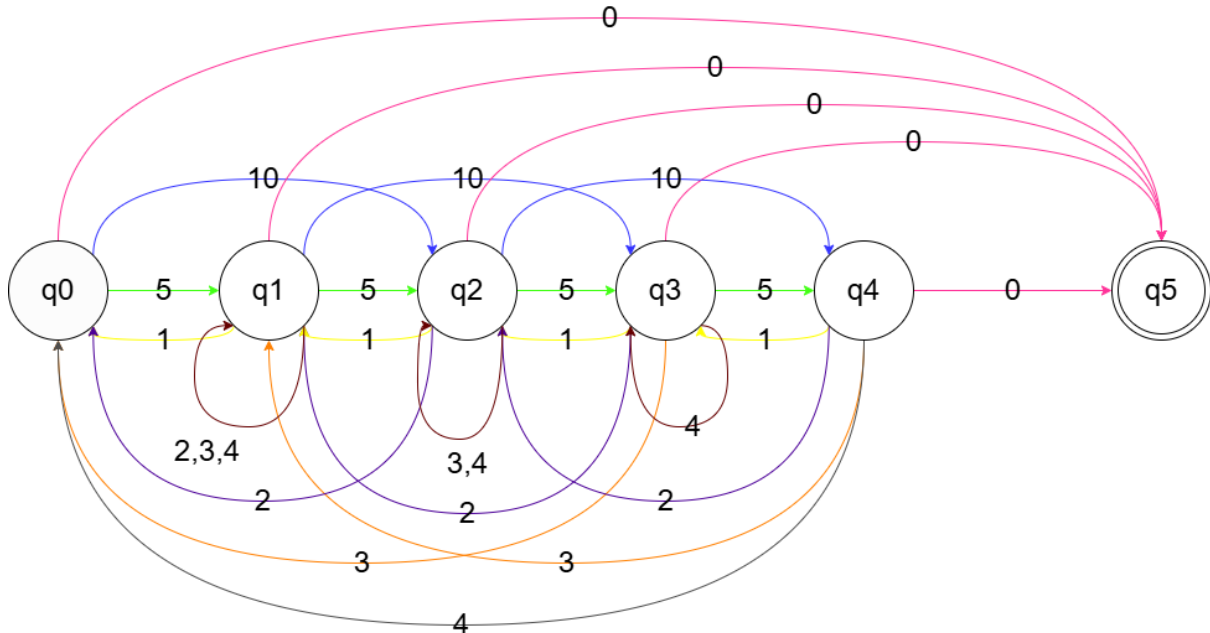
Fungsi transisi  $\delta: Q \times \Sigma \rightarrow Q$  menentukan perpindahan state.

Tabel Transisi:

State Awal	Input '5' (+5k)	Input '10' (+10k)	Input '1' (-5k)	Input '2' (-10k)	Input '3' (-15k)	Input '4' (-20k)	Input '0' (Exit)
$q_0$ (0)	$q_1$	$q_2$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$q_5$
$q_1$ (5k)	$q_2$	$q_3$	$q_0$	$\emptyset$	$\emptyset$	$\emptyset$	$q_5$
$q_2$ (10k)	$q_3$	$q_4$	$q_1$	$q_0$	$\emptyset$	$\emptyset$	$q_5$
$q_3$ (15k)	$q_4$	$q_3^*$	$q_2$	$q_1$	$q_0$	$\emptyset$	$q_5$
$q_4$ (20k)	$q_4^*$	$q_4^*$	$q_3$	$q_2$	$q_1$	$q_0$	$q_5$
$q_5$ (Out)	-	-	-	-	-	-	-

> Catatan ( $\emptyset$ ): Input tidak valid (saldo kurang), state tidak berubah.> Catatan (0): Saldo penuh, uang dikembalikan, state tidak berubah.\*

### 3. Visualisasi Diagram (Untuk Slide Presentasi)



### 4. Penjelasan Kode (Implementasi)

Program diimplementasikan menggunakan Python dengan pendekatan **Object-Oriented Programming (OOP)** untuk merepresentasikan struktur DFA.

#### 1. Representasi State:

Variabel `self.state` menyimpan string 'q0' sampai 'q5'. Ini berbeda dengan pemrograman prosedural biasa yang langsung menyimpan angka integer.

Python

```
self.state = 'q0' # State awal
```

#### 2. Mapping (Kamus Data):

Kita menggunakan Dictionary untuk memetakan label state abstrak ke nilai riil agar mesin dapat memproses logika bisnis (validasi harga).

Python

```
self.state_map = {'q0': 0, 'q1': 5000, ..., 'q5': 'KELUAR'}
```

### 3. Fungsi Transisi (transition):

Fungsi ini adalah implementasi dari  $\delta$ . Ia menerima input, melihat state saat ini, menghitung state tujuan, dan memperbarui `self.state`.

- *Logika Input Uang*: Mencari state yang nilainya = (Nilai State Sekarang + Input).
  - *Logika Beli Barang*: Mencari state yang nilainya = (Nilai State Sekarang - Harga Barang).
  - *Logika Exit*: Memaksa perpindahan ke  $q_5$  dan memicu fungsi `refund`.
- ### 4. Validasi Otomata:

Kode memeriksa apakah transisi valid (apakah state tujuan ada di dalam himpunan  $Q$ ). Jika hasil perhitungan (misal 25.000) tidak ada di `state_map`, maka transisi ditolak, sesuai prinsip DFA terbatas.

## 5. Referensi

Berikut adalah referensi yang digunakan untuk menyusun landasan teori:

1. **Buku Utama:** Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation* (3rd ed.). Pearson. (Bab tentang Deterministic Finite Automata).
2. **Buku Pendukung:** Linz, P. (2016). *An Introduction to Formal Languages and Automata* (6th ed.). Jones & Bartlett Learning.
3. **Jurnal Terkait:** Jessica Christiani Irawan, M. A. Ineke Pakereng, Ramos Somya, *Perancangan dan Implementasi Finite Automata pada Simulasi Vending Machine*, d'Cartesian.