

Modul 10: Teori Automata & Bahasa Formal

Pemodelan Mesin Keadaan Terbatas dan Analisis Sintaksis Compiler

Kusuma Web

June 18, 2026

Konsep Dasar Bahasa Formal

Definisi Teoretis

- **Abjad (Σ):** Himpunan berhingga karakter/symbol tak-kosong. Contoh: abjad biner $\Sigma = \{0, 1\}$.
- **String (Kata):** Barisan berhingga simbol-simbol dari suatu abjad. Panjang string w dinotasikan $|w|$. String kosong dinyatakan dengan ϵ ($|\epsilon| = 0$).
- **Bahasa (L):** Himpunan string-string yang dipilih dari Σ^* (himpunan semua kemungkinan string dari Σ).

Hierarki Chomsky

Klasifikasi pengelompokan tata bahasa (grammar) formal ke dalam 4 tingkatan penting:

- **Tipe 3 (Regular Grammars):** Dikenali oleh **Finite State Automata**.
- **Tipe 2 (Context-Free Grammars):** Dikenali oleh **Pushdown Automata** (mendasari struktur kode bahasa pemrograman).
- **Tipe 1 (Context-Sensitive Grammars):** Dikenali oleh **Linear Bound Automata**.

Deterministic Finite Automata (DFA)

Definisi Formal DFA

DFA adalah mesin keadaan abstrak yang didefinisikan secara matematis oleh 5-tuple:

$$M = (Q, \Sigma, \delta, q_0, F)$$

- Q : Himpunan berhingga state/keadaan.
- Σ : Abjad input.
- δ : Fungsi transisi pemetaan keadaan $\delta : Q \times \Sigma \rightarrow Q$.
- $q_0 \in Q$: State awal mesin.
- $F \subseteq Q$: Himpunan state akhir (Accepting States).

Karakteristik Deterministik

Untuk setiap state pada DFA, ketika menerima suatu karakter input, dipastikan ada **tepat**

Nondeterministic Finite Automata (NFA)

Perbedaan NFA dengan DFA

Pada NFA, untuk satu input karakter pada suatu state, mesin dapat bertransisi ke **nol, satu, atau lebih dari satu** state berikutnya secara bersamaan, serta mendukung transisi instan tanpa input menggunakan transisi epsilon (ϵ). Fungsi transisinya menghasilkan himpunan kuasa dari Q :

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$$

Teorema Ekuivalensi NFA-DFA

Setiap bahasa yang dapat dikenali oleh NFA dipastikan dapat dikenali oleh DFA. NFA dapat dikonversi menjadi DFA yang ekuivalen menggunakan metode **Subset Construction** (Algoritma Konstruksi Subset). Meskipun DFA hasil konversi memiliki perilaku yang sama, jumlah state-nya secara teoritis dapat membengkak hingga $2^{|Q|}$ state.

Implementasi Teori Automata dalam Praktik Komputasi

1. Mesin Pencocokan Pola (Regular Expression / Regex Engine)

Regex yang kita gunakan dalam rekayasa perangkat lunak (seperti `/^[a-zA-Z0-9_]+$`) di balik layar dikonversi oleh sistem menjadi diagram transisi NFA, lalu dioptimasi menjadi DFA minimal untuk melakukan validasi input teks dalam kecepatan linier $\mathcal{O}(n)$.

2. Tahap Analisis Leksikal (Lexer) pada Kompiler

Kompiler bahasa pemrograman (seperti gcc, clang, atau kompiler Java) menggunakan Finite Automata untuk memindai karakter demi karakter dari kode sumber dan mengelompokkannya menjadi kategori token yang valid (seperti kata kunci `if`, nama variabel, konstanta angka, atau operator).