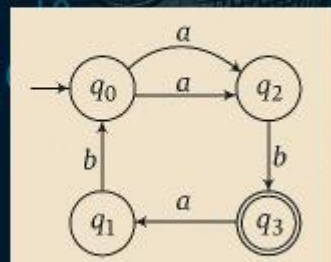
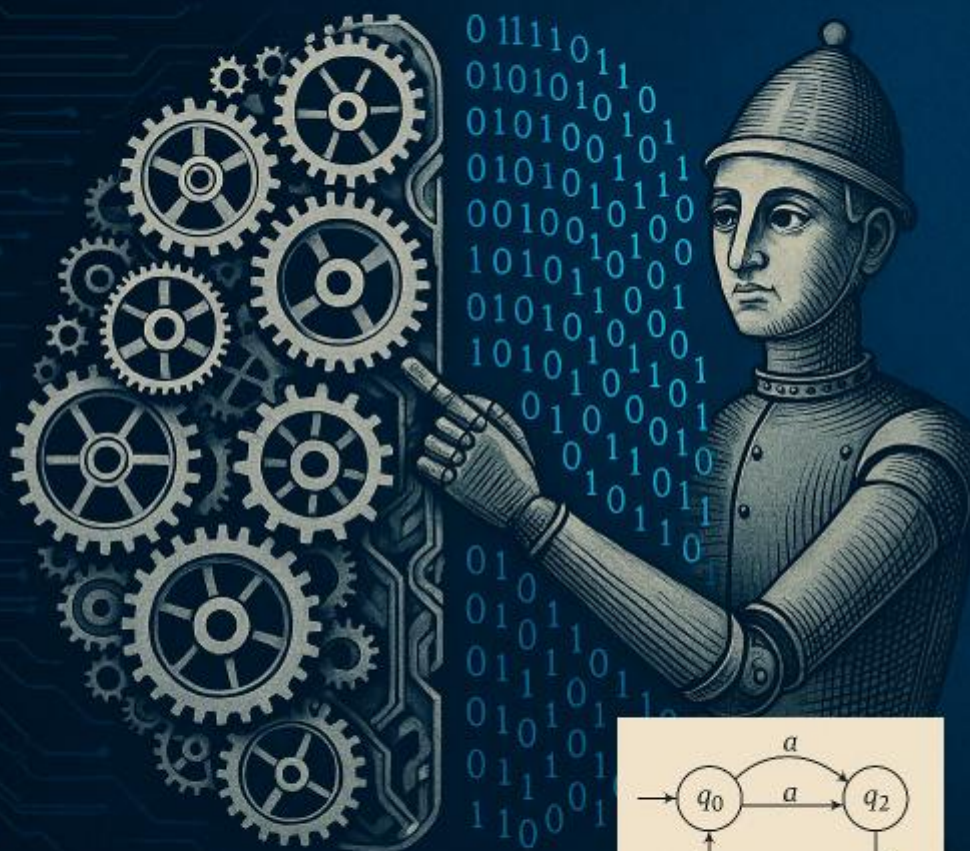


AUTOMATA



RUDY C. TARUMINGKENG

Oleh:

[Prof Ir Rudy C Tarumingkeng, PhD](#)

Professor of Management, NUP: 9903252922

Rector, Cenderawasih State University (1978-1988)

Rector, Krida Wacana Christian University (1991-2000)

© RUDYCT e-PRESS

rudyc75@gmail.com

Bogor, Indonesia

11 May, 2025

AUTOMATA

AUTOMATA (bentuk jamak dari *automaton*) adalah suatu model matematika formal yang digunakan untuk merepresentasikan sistem komputasi atau mesin abstrak yang dapat bergerak melalui serangkaian *state* (keadaan) berdasarkan input yang diterimanya. Automata merupakan dasar penting dalam teori komputasi dan berperan besar dalam pemahaman bahasa formal, compiler, pemrograman, bahkan kecerdasan buatan.

1. Pengertian Umum

Secara sederhana, **automata** adalah model ideal dari mesin yang:

- **Menerima input** berupa simbol dari alfabet tertentu
- **Memproses input** berdasarkan aturan transisi
- **Berpindah dari satu state ke state lain**
- **Menentukan hasil akhir**: apakah input diterima atau ditolak

Model ini menggambarkan **proses komputasi sebagai urutan perubahan keadaan (state transition)**.

2. Sejarah Singkat

Konsep automata dikembangkan dalam konteks **teori bahasa formal dan komputasi**. Tokoh pentingnya adalah:

- **Alan Turing** dengan mesin Turing-nya (1936)
- **Noam Chomsky** (1950-an) dengan teori gramatika formal

- **John von Neumann** yang memperkenalkan *self-replicating automata*

3. Elemen Dasar Automata

Setiap model automata umumnya terdiri dari:

1. **Q** = Himpunan terbatas dari state (keadaan)
2. **Σ** = Alfabet input (simbol-simbol yang bisa diterima)
3. **δ** = Fungsi transisi (aturan perpindahan antar state)
4. **q_0** = State awal
5. **F** = Himpunan state akhir (final/accepting state)

Ditulis dalam bentuk formal:

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

4. Jenis-Jenis Automata

Jenis Automata	Ciri Utama	Contoh Aplikasi
Finite State Automaton (FSA)	Memiliki jumlah state terbatas	Pendeteksi pola teks (regex), mesin tiket otomatis
Pushdown Automaton (PDA)	Menggunakan stack	Parsing bahasa pemrograman
Turing Machine	Memiliki tape tak terbatas dan dapat membaca/menulis	Model komputasi umum (semua yang bisa dikomputasi)
Cellular Automata	Grid sel yang berubah berdasar tetangganya	Simulasi biologis, fisika, Game of Life (Conway)

Jenis Automata	Ciri Utama	Contoh Aplikasi
Linear Bounded Automaton (LBA)	Variasi mesin Turing dengan tape terbatas	Bahasa konteks sensitif
Non-deterministic Automata (NFA)	Bisa memilih banyak jalur transisi	Representasi bahasa regular

5. Contoh Naratif: Finite State Automaton (FSA)

Misalkan kita ingin membuat automaton untuk memeriksa apakah sebuah kata berakhiran "ab".

- Alfabet: $\Sigma = \{a, b\}$
- State: $Q = \{q_0, q_1, q_2\}$
- State awal: q_0
- Final state: $F = \{q_2\}$
- Fungsi transisi:
 - $\delta(q_0, a) = q_1$
 - $\delta(q_1, b) = q_2$
 - $\delta(q_2, a/b) = q_0$

Jika input adalah "aaab", maka automata akan:

- $q_0 \rightarrow q_1$ (karena a)
- $q_1 \rightarrow q_1$ (a)
- $q_1 \rightarrow q_2$ (b)

Akhirnya berada di q_2 (final state), berarti input **diterima**.

6. Aplikasi Praktis Automata

1. **Compiler dan Parser** – Analisis sintaks bahasa pemrograman

2. **Regular Expression** – Pencarian pola dalam teks (grep, regex engines)
 3. **Kontrol Otomatis** – Elevator system, vending machine
 4. **Kecerdasan Buatan** – Behavior tree, sistem reaktif (contoh dalam game)
 5. **Simulasi Biologi** – Cellular automata untuk simulasi pertumbuhan
 6. **Sistem Keamanan** – Deteksi urutan aktivitas mencurigakan
-

7. Automata dan Bahasa Formal

Automata erat kaitannya dengan **Chomsky Hierarchy**:

- **Regular Languages** $\Leftrightarrow$ Finite Automata
 - **Context-Free Languages** $\Leftrightarrow$ Pushdown Automata
 - **Context-Sensitive Languages** $\Leftrightarrow$ Linear Bounded Automata
 - **Recursively Enumerable Languages** $\Leftrightarrow$ Turing Machine
-

8. Refleksi dan Diskusi

Automata tidak hanya konsep abstrak, tapi juga **kerangka kerja berpikir tentang proses, logika, dan sistem kompleks**. Dalam pendidikan, automata membantu:

- Mahasiswa memahami dasar pemrograman
- Peneliti membangun sistem yang dapat *meniru kecerdasan*
- Insinyur membuat kontroler sistem fisik dan digital

Di era AI dan IoT, automata tetap relevan sebagai pondasi bagi sistem pemrosesan data, otomatisasi, dan komputasi cerdas yang terstruktur.

Kesimpulan

Automata adalah model fundamental dalam teori komputasi yang menggambarkan bagaimana sistem memproses input dan membuat keputusan berdasarkan urutan aturan tertentu. Ia membentuk tulang punggung dari berbagai sistem komputasi modern, mulai dari pemrosesan bahasa, pengenalan pola, hingga kontrol mesin otomatis.

Penjelasan tentang **AUTOMATA** secara lebih mendalam dengan pendekatan akademik, mencakup akar teoritis, struktur formal, relasi dengan bahasa formal, serta aplikasinya dalam ilmu komputer dan teknologi modern.

1. Automata dalam Konteks Teori Komputasi

Dalam *theoretical computer science*, **Automata** adalah representasi abstrak dari mesin pemroses informasi yang memanipulasi simbol berdasarkan aturan tertentu. Automata digunakan untuk **menyelidiki batas kemampuan komputasi**, yaitu: *apa yang bisa dikomputasi dan bagaimana komputasi dilakukan*.

Automata menjadi alat formal untuk memahami konsep:

- Bahasa formal dan grammar
- Kompleksitas algoritma
- Desain dan verifikasi sistem

Automata merupakan bagian penting dari mata kuliah seperti:

- *Theory of Computation*
 - *Formal Languages and Automata Theory*
 - *Compiler Design*
-

2. Struktur Formal Automata: Penjelasan Matematis

Secara matematis, automata direpresentasikan sebagai **sebuah tuple 5 komponen**:

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

Komponen Penjelasan

Q	Himpunan hingga (finite) dari <i>state</i> atau keadaan
Σ	Himpunan simbol input (<i>alphabet</i>)
δ	Fungsi transisi: menentukan perpindahan state: $\delta: Q \times \Sigma \rightarrow Q$
q_0	State awal ($q_0 \in Q$)
F	Himpunan state akhir (final states) $\subseteq Q$

Contoh Transisi:

Jika automata berada di state q_1 dan menerima simbol a , maka transisi ke q_2 . Ini ditulis sebagai:

$$\delta(q_1, a) = q_2$$

Automata berjalan dengan membaca string simbol dari kiri ke kanan, berpindah antar state sesuai δ , hingga semua input habis. Jika berakhir di state dalam F , maka string **diterima** (accepted); jika tidak, **ditolak** (rejected).

3. Klasifikasi Automata (Hierarki Chomsky)

Noam Chomsky mengklasifikasikan bahasa formal dalam empat kelas, dan automata digunakan untuk memprosesnya:

Level Kelas Bahasa		Tipe Automata	Contoh
0	Recursively Enumerable	Turing Machine	Semua masalah komputasi
1	Context-Sensitive	Linear Bounded Automata (LBA)	Bahasa dengan dependensi panjang
2	Context-Free	Pushdown Automata (PDA)	Parsing bahasa pemrograman
3	Regular	Finite Automata (DFA/NFA)	Validasi email, pencocokan pola

4. Determinisme vs Non-Determinisme

◆ **DFA (Deterministic Finite Automaton)**

- Setiap input dari suatu state hanya menuju **satu state tujuan**.
- Lebih mudah diimplementasikan.
- Representasi bahasa **regular deterministik**.

◆ **NFA (Non-deterministic Finite Automaton)**

- Satu input bisa menuju ke **beberapa state berbeda**.
 - Lebih ringkas secara desain, tapi sulit untuk diimplementasi langsung.
 - **Setiap DFA dapat disimulasikan oleh NFA**, dan sebaliknya.
-

5. Automata dalam Pengolahan Bahasa Formal

Automata sangat erat dengan **grammar (tatabahasa formal)** yang membentuk bahasa (string):

Grammar	Automata	Produksi
Regular Grammar	Finite Automata	$A \rightarrow aB$
Context-Free Grammar	Pushdown Automata	$A \rightarrow aBb$
Context-Sensitive Grammar	Linear Bounded Automata	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type-0 Grammar	Turing Machine	$\alpha \rightarrow \beta$

Oleh karena itu, automata **menjadi alat bantu visualisasi dan simulasi grammar**, terutama dalam:

- **Bahasa pemrograman**
- **Desain compiler**
- **Sistem komunikasi formal**

6. Automata dan Compiler

Dalam **compiler**, automata digunakan untuk beberapa tahap:

Tahapan	Peran Automata
Lexical Analysis	Menggunakan Finite Automata untuk mendeteksi token (kata kunci, angka, simbol)
Syntax Analysis	Menggunakan Pushdown Automata untuk parsing grammar bahasa
Semantic Analysis	Melibatkan analisis kontekstual; lebih kompleks dan kadang melibatkan Turing Machines secara teoritis

Automata membantu memastikan bahwa **struktur kode valid** sebelum dijalankan oleh mesin.

7. Cellular Automata: Automata dalam Bentuk Ruang

Cellular automata adalah automata di mana "state" tersebar dalam grid dua dimensi atau lebih. Setiap *sel* berubah berdasarkan keadaan sel-sel di sekitarnya (aturan lokal → perilaku global).

Contoh paling terkenal:

- **Conway's Game of Life:** sistem dengan aturan sederhana, tapi menghasilkan kompleksitas luar biasa (emergence).
 - Digunakan untuk simulasi:
 - Sistem biologis (misalnya pertumbuhan tumor)
 - Pemodelan cuaca
 - Lalu lintas kendaraan
-

8. Penerapan Modern Automata

Dalam Teknologi Informasi:

- **Regex Engines:** Regular expression → Finite Automata
- **Firewall dan IDS:** Pendeteksi pola intrusi
- **Text Parsing:** Pengolahan bahasa alami (NLP)

Dalam Dunia Industri:

- **Sistem Otomatisasi Mesin:** Logic control
- **Simulasi Sistem Produksi:** Manufaktur berbasis aturan logis

Dalam Game dan AI:

- **State Machine AI:** NPC yang berpindah perilaku berdasarkan kondisi
- **Behavior Trees:** Turunan dari finite automata untuk logika cerdas

Dalam Jaringan dan Protokol:

- **Protokol komunikasi** → dapat dimodelkan dengan automata

- Verifikasi protokol = validasi *state machine*
-



9. Studi Kasus Naratif

► Mesin Otomatis Penjual Tiket

Sistem hanya akan mengeluarkan tiket jika pengguna:

1. Memasukkan uang minimal Rp10.000
2. Menekan tombol tujuan

Automata dapat digambarkan sebagai:

- $Q = \{q_0: \text{menunggu uang}, q_1: \text{uang cukup}, q_2: \text{tiket keluar}\}$
 - Input: $\Sigma = \{\text{masukkan 5rb}, \text{masukkan 10rb}, \text{tekan tombol}\}$
 - Transisi:
 - $\delta(q_0, 10\text{rb}) \rightarrow q_1$
 - $\delta(q_1, \text{tombol ditekan}) \rightarrow q_2$
 - $\delta(q_2, -) \rightarrow q_0$
-



10. Penutup dan Refleksi

Teori automata bukan sekadar materi teknis, melainkan juga paradigma berpikir tentang **alur logika, proses transformasi, dan representasi sistem dinamis**. Automata juga merupakan **landasan dari logika pemrograman modern** dan teknologi cerdas.

Dengan memahami automata:

- Kita bisa menyusun *algoritma yang robust*
 - Mendesain *sistem yang logis dan dapat ditelusuri*
 - Menelusuri batas kemampuan mesin untuk berpikir dan bertindak
-

Pendalaman

Kelanjutan pembahasan mendalam tentang Automata, yang fokus pada:

1.  **Analisis Automata dalam Perspektif Komputasi Modern**
2.  **Automata dan Kecerdasan Buatan**
3.  **Pengembangan Otomatisasi dan Desain Sistem dengan Automata**
4.  **Automata dalam Sistem Digital dan Embedded Systems**
5.  **Automata sebagai Simulasi Dunia Nyata (Natural Computing)**
6.  **Implikasi Filosofis dan Teoritis: Automata, Kesadaran, dan Mesin**

1. Automata dalam Perspektif Komputasi Modern

Seiring berkembangnya teknologi komputasi modern (cloud, AI, big data), automata tetap relevan sebagai **core engine dalam berbagai sistem digital**. Mengapa?

- Karena **intisari komputasi adalah transformasi informasi**: dari input menjadi output, dan ini adalah prinsip dasar automata.
- Dalam machine learning pun, banyak proses inferensi yang mengasumsikan automata deterministik dalam background-nya, terutama dalam **sequence prediction**.

Contoh:

- Model seperti **RNN (Recurrent Neural Network)** adalah “versi probabilistik dan adaptif” dari automata—mereka memproses sekuens input dan memutuskan output berdasarkan state internal.

2. Automata dan Kecerdasan Buatan (AI)

Automata dalam AI digunakan pada:

Aplikasi	Penjelasan
State Machine AI	Representasi perilaku dalam bentuk transisi antar keadaan. Contoh: musuh dalam game berubah dari “berpatroli” → “mengejar” → “menyerang”
Planning and Decision-Making	Decision tree dan behavior tree sebagai ekspresi automata yang kompleks
Natural Language Processing (NLP)	Analisis morfologis, tokenisasi, dan penguraian struktur kalimat menggunakan Finite Automata dan Pushdown Automata

Dalam **chatbot** misalnya, banyak NLP awal (sebelum deep learning) menggunakan Finite State Transducer (FST) — yaitu automata yang tidak hanya menerima string, tapi juga **menghasilkan keluaran berdasarkan input**.

3. Pengembangan Sistem Otomatis dan Automasi Industri

Automata banyak digunakan dalam *design control system*:

- **PLC (Programmable Logic Controller)** menggunakan prinsip finite state machine.
- Di dunia industri, sistem seperti conveyor, robot industri, dan quality checking system dibangun berdasarkan diagram *state transition*.

Narasi Kasus:

Bayangkan sebuah sistem robot pengemasan:

- State: idle, scan item, tarik item, masukkan ke box, verifikasi berat, tutup box
- Input: sinyal sensor
- Transisi terjadi ketika item terdeteksi, berat sesuai, atau box penuh.

Ini adalah bentuk **automata industri** dalam praktik nyata.

4. Automata dalam Sistem Digital dan Embedded System

Dalam **sistem tertanam (embedded systems)**, automata digunakan sebagai **desain arsitektur perangkat lunak logika**:

Contoh:

- Elevator controller
- Mesin ATM
- Kendaraan otonom sederhana
- Sistem parkir otomatis

Automata membantu membatasi *state* dan memastikan bahwa sistem:

- Tidak macet (deadlock)
- Tidak mengalami *race condition*
- Selalu kembali ke keadaan stabil

Dalam pemrograman seperti di **VHDL** dan **Verilog** untuk FPGA, finite state machine digunakan sebagai pendekatan desain logika digital yang efisien.

✖ 5. Automata Sebagai Simulasi Dunia Nyata (Natural Computing)

Automata juga muncul dalam **natural computing** — pendekatan yang meniru sistem biologis atau fisika alam untuk pemrosesan informasi.

Beberapa contohnya:

- **Cellular Automata** dalam:
 - Simulasi reaksi kimia
 - Penyebaran virus (epidemiologi)
 - Pertumbuhan populasi
- **DNA Computing** dan **Molecular Automata** yang merepresentasikan data dalam bentuk biologis

Kasus:

- **Rule 110** dalam Cellular Automata terbukti **Turing complete**: artinya dengan konfigurasi tertentu, bisa meniru mesin Turing.
- Ini menandakan bahwa **dunia alamiah juga bisa dipandang sebagai automata** dalam skala tertentu.

📖 6. Implikasi Filosofis dan Teoretis: Automata, Kesadaran, dan Mesin

Sejak abad ke-17, konsep *automaton* sudah dibahas oleh filsuf seperti **René Descartes** yang menyebut tubuh manusia sebagai “mesin biologis”, bahkan hewan pun dianggap sebagai automata oleh filsafat mekanistik.

Pertanyaan mendalam:

- Jika otak kita adalah sistem yang berubah dari satu state ke state lain karena input sensorik, **apakah kita adalah automaton biologis?**
- Apakah **kesadaran** bisa muncul dari sistem automata yang cukup kompleks?

Dalam konteks ini:

- **Turing Machine** dianggap sebagai batas paling bawah dari sistem "yang bisa berpikir".
- Para peneliti AI mencoba menciptakan *self-adaptive automata* (automata yang bisa belajar dari lingkungan dan mengubah state transition-nya sendiri).

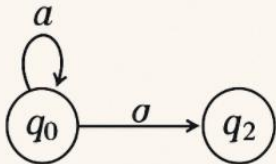
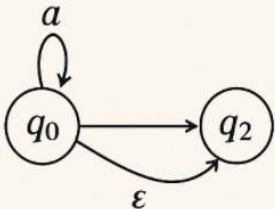
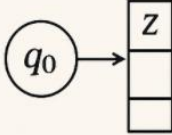
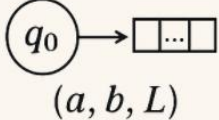
 Penutup: Automata sebagai Bahasa Umum Sistem Cerdas

Dapat dikatakan bahwa automata:

- Adalah **abstraksi dari sistem pemroses informasi**
- Menjadi **model mental dalam mendesain sistem cerdas**
- Menjembatani **matematika murni, rekayasa, dan logika komputasi**

Automata memungkinkan manusia **mendisiplinkan logika, memformalkan perilaku, dan mengimplementasikan kecerdasan dalam sistem fisik dan digital.**

Diagram perbandingan visual jenis automata: DFA, NFA, PDA, dan TM.

DETERMINISTIC FINITE AUTOMATON (DFA)	NONDETERMINISTIC FINITE AUTOMATON (NFA)	PUSHDOWN AUTOMATON (PDA)	TURING MACHINE (TM)
			
• Finite set of states • Single initial state • Accepts regular languages	• Finite set of states • Single initial state • Accepts context-free languages	• Finite set of states • Stack for memory • Accepts context-free languages	• Finite set of states • Infinite tape for memory • Accepts recursively enumerable languages

Studi Kasus

Berikut adalah dua **studi kasus penerapan automata** secara naratif dan mendalam—satu di bidang **Natural Language Processing (NLP)** dan satu lagi di **industri manufaktur**. Keduanya menunjukkan bagaimana automata (DFA, PDA, hingga konsep turunan) digunakan dalam sistem nyata.

STUDI KASUS 1: Automata dalam NLP — *Lexical Analysis pada Proses Pemrosesan Bahasa*

Latar Belakang:

Dalam Natural Language Processing (NLP), proses analisis kata-kata (tokenisasi dan klasifikasi) merupakan tahap awal dan krusial dalam pipeline pemrosesan teks. Salah satu aplikasi utamanya adalah **compiler** atau sistem pemroses bahasa buatan (chatbot, text analyzer), yang mengidentifikasi kata berdasarkan pola (misalnya kata benda, kata kerja, dll.).

Automata yang Digunakan:

- **Deterministic Finite Automaton (DFA)**
 - **Non-deterministic Finite Automaton (NFA)**
 - **Finite State Transducer (FST)** (varian automata yang juga menghasilkan output)
-

Skenario Proses:

Misalnya kita ingin mengenali kata kerja bentuk lampau dalam bahasa Inggris yang diakhiri dengan “-ed”:

Contoh kata: walked, played, jumped

Automata dibentuk dengan transisi state yang mengenali struktur akhir ed. Setiap karakter yang dibaca akan membawa automata dari state awal ke state penerimaan jika struktur sesuai.

Struktur DFA:

$q_0 \xrightarrow{w,a,l,k} q_1 \xrightarrow{e} q_2 \xrightarrow{d} q_3 \text{ (accepting)}$

Aplikasi di NLP:

- **Token recognition:** mengenali kata dasar dan imbuhan
- **Spelling correction engine:** memeriksa pola huruf secara deterministik
- **Speech recognition:** menggunakan probabilistic automata (Hidden Markov Models) untuk menebak transisi fonem
- **Morphological Analyzer:** FST digunakan untuk mengurai kata menjadi morfem (contoh: unhappiness $\rightarrow$ un + happy + ness)

Hasil dan Implikasi:

- Meningkatkan akurasi parsing kalimat secara struktural
- Memungkinkan analisis cepat terhadap struktur bahasa dalam volume besar (big text corpus)
- DFA bekerja cepat dan efisien dalam mengklasifikasikan pola leksikal

STUDI KASUS 2: Automata dalam Industri Manufaktur — *Sistem Kendali Otomatis Conveyor Barang*

Latar Belakang:

Sebuah pabrik pengemasan minuman ringan ingin mengotomatisasi proses **pengemasan dan pengecekan berat** botol yang lewat di atas conveyor. Sistem ini harus mengatur:

1. Sensor botol masuk

2. Pengisian otomatis
3. Verifikasi berat
4. Penyisihan produk cacat

Automata yang Digunakan:

- **Finite State Machine (FSM/DFA)** untuk sistem logika transisi
 - **Embedded Controller Automata** dalam PLC (Programmable Logic Controller)
 - Diagram ladder yang diterjemahkan ke dalam model automata
-

Desain Sistem Automata:

State:

- q0: idle
- q1: sensor mendeteksi botol
- q2: proses pengisian
- q3: pengecekan berat
- q4: validasi
- q5: botol dilepas
- q6: botol disisihkan jika cacat

Transisi:

- Sensor mendeteksi → q0 → q1
 - Isi botol → q1 → q2
 - Cek berat → q2 → q3
 - Jika berat benar → q3 → q4 → q5
 - Jika salah → q3 → q6
-

Aplikasi Industri:

- **Pemrograman PLC berbasis state diagram**
 - Sistem logika kontrol untuk **robotik produksi**
 - Deteksi dan respon terhadap error/penyimpangan dalam lini produksi
-

Hasil dan Implikasi:

- **Efisiensi meningkat** karena sistem dapat memproses tanpa intervensi manusia
 - **Reliabilitas tinggi**, karena setiap langkah mengikuti pola terstruktur dan bisa dilacak
 - **Automata memudahkan debugging**: cukup telusuri state mana yang menyebabkan kegagalan
-

Perbandingan Singkat Kedua Studi Kasus

Aspek	NLP	Industri Manufaktur
Tipe Automata	DFA, NFA, FST	DFA/FSM
Input	Teks (karakter)	Sensor dan sinyal mesin
Output	Label token, struktur kata	Aksi fisik mesin
Manfaat	Pengenalan bahasa	Otomasi dan efisiensi produksi
Kompleksitas	Abstraksi linguistik	Proses kontrol real-time

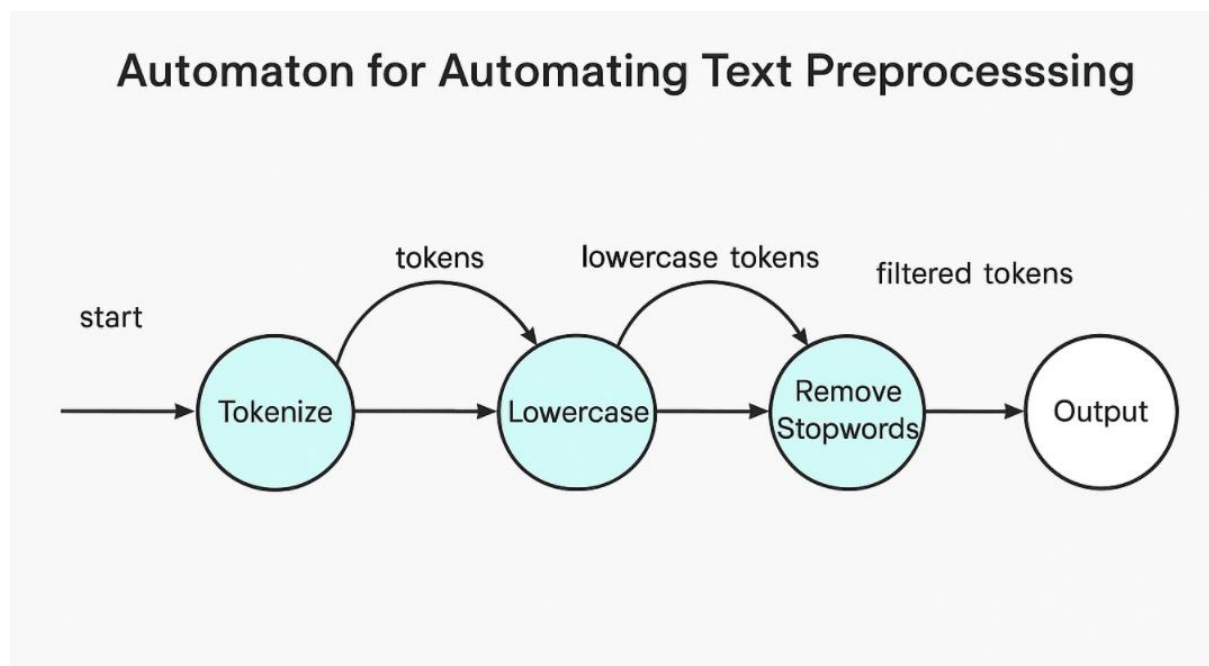
Refleksi Penutup

Automata tidak hanya berada di dunia teori komputer, tetapi menjadi **kerangka kerja universal** dalam sistem logika, bahasa, dan industri. Di NLP, automata memungkinkan komputer "mengerti" bahasa; di

manufaktur, automata menjamin proses berjalan sesuai urutan dan logika.

Kedua studi kasus ini menunjukkan **adaptabilitas dan skalabilitas automata dalam domain yang berbeda**, dengan prinsip yang sama: *state*, *transisi*, dan *reaksi* terhadap input.

Berikut adalah diagram proses automata untuk automasi pemrosesan teks dalam NLP. Selanjutnya saya akan buat modul ajarnya.



Penerapan Automata dalam NLP dan Industri Manufaktur

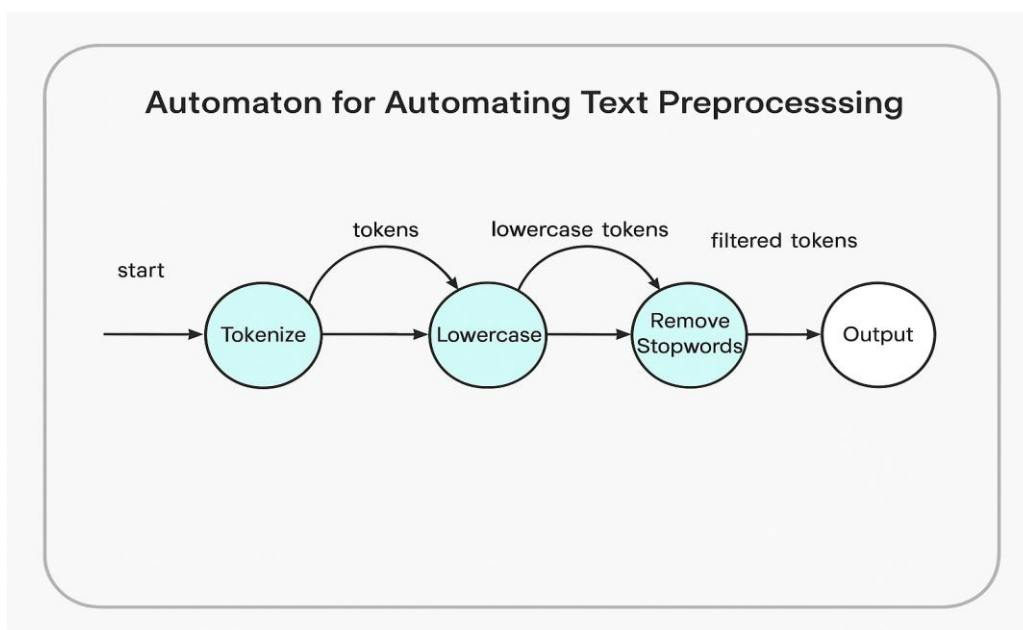
1. Pendahuluan

Automata adalah model matematis dari sistem yang memproses input secara bertahap melalui serangkaian state. Automata banyak digunakan dalam berbagai bidang, termasuk Natural Language Processing (NLP) dan industri manufaktur. Modul ini membahas dua studi kasus penerapan automata secara praktis dan aplikatif.

2. Studi Kasus NLP: Automasi Preprocessing Teks

Dalam NLP, automata digunakan untuk membentuk alur kerja pemrosesan teks seperti tokenisasi, transformasi huruf kecil, dan penghapusan stopwords. Proses ini dapat direpresentasikan sebagai automata deterministik, di mana setiap tahap pemrosesan adalah satu state.

Diagram proses automata:



Keterangan proses:

- Tokenize: memecah kalimat menjadi token (kata-kata).
- Lowercase: mengubah semua token menjadi huruf kecil.
- Remove Stopwords: menghapus kata-kata umum yang tidak bermakna spesifik (seperti 'dan', 'yang', 'di').
- Output: hasil akhir adalah token yang sudah difilter.

3. Studi Kasus Industri Manufaktur: Sistem Otomatisasi Conveyor

Pada industri manufaktur, automata digunakan dalam sistem kendali otomatis berbasis PLC. Contoh aplikasinya adalah sistem conveyor untuk pengepakan botol, di mana tiap state mewakili tahapan proses seperti mendeteksi botol, mengisi cairan, memverifikasi berat, dan menyisihkan jika cacat.

State utama:

- q0: idle (menunggu botol)
- q1: mendeteksi botol
- q2: pengisian
- q3: pengecekan berat
- q4: validasi dan pelepasan
- q5: penyisihan jika cacat

Transisi terjadi berdasarkan input dari sensor dan logika kontrol.

4. Penutup

Kedua studi kasus ini menunjukkan fleksibilitas automata dalam merepresentasikan proses di bidang yang berbeda. Dengan pendekatan berbasis state, sistem menjadi lebih terstruktur, mudah dipantau, dan dapat dioptimalkan.

Berikut adalah **Glosarium** untuk modul ajar "*Penerapan Automata dalam NLP dan Industri Manufaktur*", disusun secara alfabetis dan relevan dengan konteks automata, NLP, dan sistem manufaktur:

GLOSARIUM

Istilah	Definisi
Accepting State	Keadaan akhir dalam automata di mana input diterima sebagai valid.
Alphabet (Σ)	Himpunan simbol dasar yang digunakan sebagai input dalam automata.
Automata	Model matematis untuk sistem yang bertransisi antar state berdasarkan input tertentu.
Compiler	Program yang menerjemahkan kode sumber menjadi kode mesin, memanfaatkan automata dalam tahap tokenisasi dan parsing.
Conveyor System	Sistem mekanik yang digunakan untuk mengangkut barang secara otomatis dalam lini produksi industri.
DFA (Deterministic Finite Automaton)	Jenis automata di mana setiap kombinasi state dan input memiliki tepat satu transisi ke state berikutnya.
FST (Finite State Transducer)	Varian automata yang selain menerima input, juga menghasilkan output pada setiap transisi.
FSM (Finite State Machine)	Nama umum untuk sistem automata terbatas yang digunakan dalam pemrograman logika dan kontrol.

Istilah	Definisi
Input Symbol	Karakter atau sinyal yang dimasukkan ke dalam automata untuk memicu transisi state.
Lexical Analysis	Proses analisis teks atau kode menjadi token; merupakan tahap awal dalam NLP atau compiler.
Lowercase	Proses mengubah huruf kapital menjadi huruf kecil dalam preprocessing teks NLP.
NFA (Non-deterministic Finite Automaton)	Jenis automata yang memungkinkan lebih dari satu transisi untuk kombinasi state dan input tertentu.
Output	Hasil dari automata setelah memproses input berdasarkan urutan transisi state.
Parsing	Analisis struktur kalimat atau kode untuk memastikan kesesuaian dengan aturan grammar.
PLC (Programmable Logic Controller)	Komputer industri yang menggunakan logika berbasis automata untuk mengontrol proses manufaktur.
PDA (Pushdown Automaton)	Automata yang menggunakan stack untuk menyimpan dan memproses input, digunakan untuk bahasa bebas konteks.
Preprocessing	Tahap awal dalam NLP untuk membersihkan dan menyiapkan data teks sebelum dianalisis lebih lanjut.
Sensor	Perangkat yang mendeteksi perubahan fisik (seperti keberadaan botol) dan menghasilkan sinyal input untuk automata.

Istilah	Definisi
Stack	Struktur data LIFO (Last In First Out) yang digunakan dalam PDA untuk menyimpan simbol.
State (Keadaan)	Representasi dari suatu kondisi sistem pada saat tertentu dalam automata.
Stopword	Kata-kata umum dalam bahasa alami (seperti "dan", "itu", "di") yang biasanya dihapus dalam proses NLP.
Tokenisasi	Pemisahan kalimat atau teks menjadi unit-unit kata (token) dalam NLP.
Transition Function (δ)	Fungsi yang menentukan perpindahan state dalam automata berdasarkan input tertentu.
Turing Machine	Model automata paling kuat yang dapat meniru proses komputasi apapun, menggunakan pita tak terbatas sebagai memori.
Validation	Proses pengecekan akhir untuk memastikan hasil akhir sesuai dengan kriteria (misalnya berat botol sesuai).

Berikut adalah **Daftar Pustaka** untuk modul "*Penerapan Automata dalam NLP dan Industri Manufaktur*", mencakup literatur utama dalam teori automata, NLP, dan sistem industri. Daftar ini mencakup sumber akademik, buku teks, dan referensi teknis.

DAFTAR PUSTAKA

1. **Hopcroft, J. E., Motwani, R., & Ullman, J. D.** (2006). *Introduction to Automata Theory, Languages, and Computation* (3rd ed.). Pearson Education.

Buku rujukan klasik yang menjadi standar dalam pembelajaran teori automata dan bahasa formal.

2. **Jurafsky, D., & Martin, J. H.** (2021). *Speech and Language Processing* (3rd ed.). Draft. Stanford University.

Referensi utama dalam NLP yang mencakup penggunaan automata untuk analisis leksikal dan morfologis.

3. **Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D.** (2006). *Compilers: Principles, Techniques, and Tools* (2nd ed.). Pearson.

Dijuluki "Dragon Book", membahas penerapan automata dalam lexing dan parsing bahasa pemrograman.

4. **Sipser, M.** (2012). *Introduction to the Theory of Computation* (3rd ed.). Cengage Learning.

Buku referensi mendalam untuk teori komputasi, Turing machine, dan batas-batas automata.

5. **Russell, S., & Norvig, P.** (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Buku teks AI modern yang membahas automata dalam konteks sistem berbasis aturan dan agent.

6. **Zhou, L., & Su, J.** (2019). *Industrial Automation and Control System Security Principles*. Springer.

Buku yang membahas sistem kendali otomatisasi dan bagaimana automata digunakan dalam sistem kontrol.

7. **IEC 61131-3 Standard** – *Programmable Controllers – Part 3: Programming Languages* (2013). International Electrotechnical Commission.

Standar internasional untuk pemrograman PLC yang berdasarkan desainnya pada automata dan state machine.

8. **Conway, J. H.** (1970). *The Game of Life*. Scientific American.

Artikel populer yang memperkenalkan cellular automata dalam konteks simulasi sistem kompleks.

9. **Chomsky, N.** (1956). *Three Models for the Description of Language*. IRE Transactions on Information Theory.

Tulisan fundamental yang menjadi dasar hierarki Chomsky dan kaitannya dengan automata.

10. **OpenAI Documentation** – *Natural Language Processing and Automata Applications*. (2024).

Panduan teknis dan eksperimen praktis dengan automata dan NLP dalam platform AI modern.

11. ChatGPT 4o (2025). Access date: 11 May 2025. Prompting by [Rudy C Tarumingkeng](https://chatgpt.com/c/681fce48-cb28-8013-957c-25fb0118f52c) on Writer's account.
<https://chatgpt.com/c/681fce48-cb28-8013-957c-25fb0118f52c>