

MAKALAH TEORI BAHASA DAN AUTOMATA



DISUSUN OLEH KELOMPOK 1 :

Rizmawan Iqbal Lutfi	1810651040
Rizky Irfan Sulthoni	1910651165
Mohammad Firmansyah	1810651057
Alfa Yunitra	1810651042
Wisdayanti	1910651164

UNIVERSITAS MUHAMMADIYAH JEMBER

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

2020

## KATA PENGANTAR

Puja dan puji syukur penulis panjatkan atas kehadiran Allah yang Maha Esa, karena berkat rahmat-Nya penulis dapat menyelesaikan makalah ini yang berjudul “Teori Bahasa dan Automata”. Makalah ini dibuat dengan tujuan memenuhi tugas dari Bapak Ginanjar selaku dosen Teori Bahasa dan Otomata.

Sesuai dengan tugas yang diberikan, penulis menyadari bahwa makalah ini jauh dari sempurna, oleh karena itu penulis mengharapkan kritik dan saran dari Bapak serta rekan-rekan sekalian sehingga penulis dapat memperbaiki kesalahan-kesalahan dalam makalah ini dan menyempurnakannyasehingga menjadi sumber ilmu yang bermanfaat bagi kita semua.

Akhir kata penulis ucapkan terima kasih dan Semoga makalah ini dapat memenuhitugas yang diberikan dan dapat menjadi acuan untuk menghasilkan makalah yang lebih baik lagi.

Jember, 08 Maret 2020

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB 1. PENDAHULUAN.....	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah.....	2
BAB 2 PEMBAHASAN.....	3
2.1    Sejarah Automata.....	3
2.2    Definisi Automata.....	5
2.3    Contoh Terapan Teori Otomata.....	6
2.4    Sifat –Sifat Otomata.....	7
2.5    Konsep Dasar Bahasa Formal.....	10
2.6    Konsep Bahasa dan Automata.....	10
2.7    Elemen Bahasa Formal.....	11
BAB III. PENUTUP.....	13
3.1.    Kesimpulan.....	13
3.2.    Saran.....	14
DAFTAR PUSTAKA.....	15

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Teori bahasa membicarakan bahasa formal (*formal language*), terutama untuk kepentingan perancangan kompilator (*compiler*) dan pemroses naskah (*text processor*). Bahasa formal adalah kumpulan *kalimat*. Semua kalimat dalam sebuah bahasa dibangkitkan oleh sebuah tata bahasa (*grammar*) yang sama. Sebuah bahasa formal bisa dibangkitkan oleh dua atau lebih tata bahasa berbeda. Dikatakan bahasa formal karena grammar diciptakan mendahului pembangkitan setiap kalimatnya. Bahasa manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya 'bahasa formal' akan disebut 'bahasa' saja.

Automata adalah mesin abstrak yang dapat mengenali, menerima atau membangkitkan sebuah kalimat dalam bahasa tertentu. Automata berkaitan erat dengan teori bahasa formal. Selain itu juga ada beberapa hal yang berkaitan dengan Otomata, yaitu Grammar. Grammar adalah bentuk abstrak yang dapat diterima untuk membangkitkan suatu kalimat otomata berdasarkan suatu aturan tertentu. Automata memiliki empat tipe grammar yang disebut dengan Hirarki Chomsky, yaitu Unrestricted Grammar (UG), Context Sensitive Grammar (CSG), Context Free Grammar (CFG), dan Regular Grammar (RG).

Teori bahasa automata merupakan salah satu teori komputasi pada ilmu komputer. Teori komputasi datang dari bahasa dan rekayasa sistem, terutama yang berbasiskan matematika. Teori bahasa automata dapat dijadikan suatu gagasan mendasar dalam komputasi yang menjadi tools untuk mengenali suatu persoalan atau masalah karena dapat memberikan konsep dan prinsip untuk memahami suatu persoalan yang berkorelasi dengan bidang ilmu komputer.

## **1.2 Rumusan Masalah**

1. Bagaimana sejarah otomata bermula?
2. Apa definisi otomata sendiri?
3. Bagaimana contoh terapan teori otomata?
4. Apa saja sifat – sifat otomata?
5. Apa saja konsep dasar Bahasa formal?
6. Apa saja Elemen yang ada di dalam Bahasa formal?

## **BAB 2 PEMBAHASAN**

### **2.1 Sejarah Automata**

Otomata bermula sebelum komputer ada pada teori di bidang sistem logika matematika atau formal, ilmuwan David Hilbert telah mencoba menciptakan algoritma umum untuk pembuktian (seluruh) persoalan matematika secara otomatis yaitu mampu menentukan salah benarnya sembarang prosisi matematika.

Tahun 1931, KurtGdel mempublikasikan teori ketidaklengkapan dimana membuktikan prosedur/algoritma yang dikehendaki David Hilbert tersebut tidak akan pernah ada.

KurtGdel membangun rumus di kalkulus predikat yang diterapkan pada bilangan bulat yang memiliki pernyataan-pernyataan definisi yang tidak dapat dibuktikan maupun dibantah di dalam sistem logika yang mungkin dibangun manusia.

Formalisasi argumen teorema ketidaklengkapan KurtGdel ini berikut penjelasan dan formalisasi selanjutnya dari prosedur efektif secara intuitif merupakan salah satu pencapaian intelektual terbesar abad 20, yaitu abad dimana formalisasi berkembang semarak.

Pengembangan teori otomata, komputasi dan teori bahasa berikutnya difasilitasi perkembangan bidang psyo-linguistic. Bidang psyo-linguistic berupaya menjawab pertanyaan-pertanyaan berikut:

- Apakah bahasa secara umum?
- Bagaimana manusia mengembangkan bahasa?
- Bagaimana manusia memahami bahasa?
- Bagaimana manusia mengajarkan bahasa ke anak-anaknya?
- Apa gagasan-gagasan yang dapat dinyatakan dan bagaimana caranya?
- Bagaimana manusia membangun kalimat-kalimat dari gagasan-gagasan yang berada dipikirannya?

Sekitar tahun 1950-an, Noam Chomsky menciptakan model matematika sebagai sarana untuk mendeskripsikan bahasa serta menjawab pertanyaan-pertanyaan di atas. Saat ini dimulai pendalaman bidang bahasa computer.

Sekitar tahun 1950-an, Noam Chomsky menciptakan model matematika sebagai sarana untuk mendeskripsikan bahasa serta menjawab pertanyaan-pertanyaan di atas. Saat ini dimulai pendalaman bidang bahasa komputer.

Perbedaan antara bahasa komputer dan bahasa manusia adalah sampai sekarang belum diketahuinya bagaimana cara manusia mengartikan bahasa, sementara dengan pasti dapat mengartikan bahasa pada komputer.

Noam Chomsky mengemukakan perangkat format disebut grammar untuk memodelkan properti-properti bahasa. Tata bahasa (grammar) bisa didefinisikan secara formal sebagai kumpulan dari himpunan? himpunan variabel, simbol? simbol, terminal, simbol awal, yang dibatasi oleh aturan? aturan produksi. Tingkat bahasa dapat digolongkan menjadi empat yaitu :

### **1. Bahasa : Regular type 3**

Mesin otomata: Finite State Otomata (FSA) meliputi deterministic finite automata dan non deterministic finite automata

Batasan aturan produksi: adalah sebuah simbol variabel maksimal memiliki sebuah simbol variabel yang bila terletak di posisi paling kanan.

### **2. Bahasa : Bebas konteks/context free /type 2**

Mesin otomata: Push down automata (PDA)

Batasan aturan produksi: Berupa sebuah simbol variabel.

### **3. Bahasa : Context sensitive/type 1**

Mesin otomata: Linier bounded automata Batasan aturan produksi.

#### 4. Bahasa : Unrestricted /phase /natural language/type 0

Mesin otomata: Mesin turing

Batasan aturan produksi: Tidak ada batasan

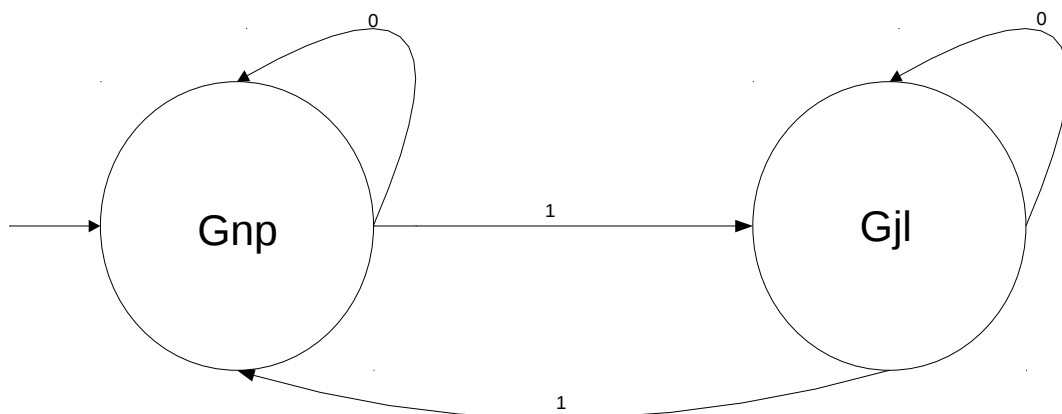
### 2.2 Definisi Automata

#### Teori Bahasa

- Teori bahasa membicarakan bahasa formal (*formal language*), terutama untuk kepentingan perancangan kompilator (*compiler*) dan pemroses naskah (*text processor*).
- Bahasa formal adalah kumpulan *kalimat*. Semua kalimat dalam sebuah bahasa dibangkitkan oleh sebuah tata bahasa (*grammar*) yang sama.
- Sebuah bahasa formal bisa dibangkitkan oleh dua atau lebih tata bahasa berbeda.
- Dikatakan bahasa formal karena grammar diciptakan mendahului pembangkitan setiap kalimatnya.
- Bahasa Natural/manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya 'bahasa formal' akan disebut 'bahasa' saja.

Otomata (Automata)

Otomata adalah mesin abstrak yang dapat mengenali (*recognize*), menerima (*accept*), atau membangkitkan (*generate*) sebuah kalimat dalam bahasa tertentu.





## 2.3 Contoh Terapan Teori Otomata

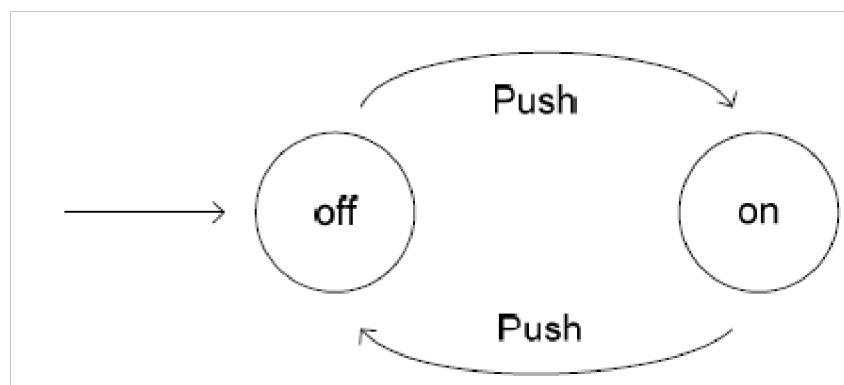
Contoh Penerapan Teori Bahasa otomata Model switch on/off digambarkan sebagai berikut :

### Contoh 1:

Model tersebut mengingat apakah *switch* berada dalam state "*on*" atau state "*off*". Model memungkinkan user untuk menekan tombol yang memiliki pengaruh berbeda tergantung pada keadaan *switch*:

- *switch* berada dalam state "*off*" maka setelah tombol ditekan state berubah menjadi "*on*".
- Jika *switch* berada dalam state "*on*" maka setelah tombol ditekan state berubah menjadi "*off*".

Model pada Gambar 1 dapat dipandang sebagai model *finite automata* sederhana.



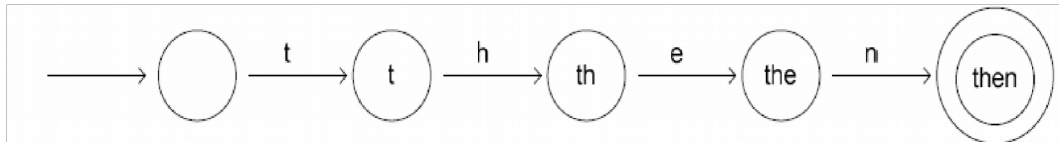
Dalam *finite automata*, **state** dinyatakan oleh lingkaran, dan dalam Contoh 1 *state* diberi nama "*on*" dan "*off*". **Arc** diantara *state* diberi label "input " yang menyatakan pengaruh eksternal pada sistem. Dalam Contoh 1 kedua *arc* diberilabel 'push" yang menyatakan *user* menekan tombol tertentu.

Salah satu *state* dinyatakan sebagai **start state** atau **initial state** yang merupakan *state* dimana sistem berada dalam keadaan awal. Dalam Contoh *start state* adalah off. Dalam pembahasan selanjutnya, *start state* ditunjukkan oleh kata *start* dan panah menuju *start state* tersebut. Dalam Gambar 1 *state* on dinyatakan sebagai **final** atau **accepting** state.

Dalam *state* tersebut, peralatan yang sedang dikontrol oleh *switch* akan beroperasi. Dalam pembahasan selanjutnya, *final State* dinyatakan dalam lingkaran ganda.

### Contoh 2:

*Finite automaton* berikut dapat dinyatakan sebagai bagian dari *lexical analyzer*.



Tugas dari *automaton* tersebut adalah mengenali *keyword* "then" sehingga diperlukan lima *state* masing-masing menyatakan posisi yang berbeda dalam kata "then" yang telah dicapai sejauh ini. Posisi ini berhubungan dengan prefix dari kata yang berkisar dari kata string kosong (tidak ada kata yang dikenali sejauh ini) sampai dengan kata lengkap. Dalam Gambar 2, input dinyatakan oleh huruf. *Start state* merupakan string kosong, dan setiap *state* memiliki transisi pada huruf selanjutnya dari kata then ke *state* yang menyatakan prefix selanjutnya yang lebih besar. *State* yang diberi nama "then" dimasuki ketika input mengeja kata "then". Karena fungsi dari model dalam Gambar 2 adalah mengenali kata then, maka *state* "then" dinyatakan sebagai *accepting state*.

## 2.4 Sifat –Sifat Otomata

Automata adalah suatu mesin sekuensial (otomatis), yang menerima input (dari pita masukan) dan mengeluarkan output, keduanya dalam bentuk diskrit. Automata mempunyai sifat-sifat

- Kelakuan mesin bergantung pada rangkaian masukan yang diterima mesin tersebut.
- Setiap saat, mesin dapat berada pada satu status tertentu dan dapat berpindah ke status baru karena adanya perubahan input.
- Rangkaian input (diskrit) pada mesin automata dapat dianggap sebagai bahasa yang harus "dikenali" oleh sebuah automata. Setelah pembacaan input selesai, mesin automata kemudian membuat "keputusan". Jenis- jenis automata :

Jenis	Pita masukan	Arah Head	Memori
<b>Finite State</b>	Read Only	1 arah	-
<b>Push Down</b>	Read Only	1 arah	stack
<b>Linear-Bounded</b>	R/W	2 arah	(bounded)
<b>Turing Machine</b>	R/W	2 arah	(unbounded)

Pada bahasan ini jenis automata yang akan dipakai adalah Finite State Automata (FSA). FSA adalah mesin yang dapat mengenali kelas bahasa reguler dan memiliki sifat-sifat:

1. Pita masukan (input tape) berisi rangkaian simbol (string) yang berasal dari himpunan simbol / alfabet.
2. Setiap kali setelah membaca satu karakter, posisi read head akan berada pada symbol berikutnya.
3. Setiap saat, FSA berada pada status tertentu
4. Banyaknya status yang berlaku bagi FSA adalah berhingga.

Suatu FSA didefinisikan sebagai  $F = (Q, S, q_0, d, F)$  dengan

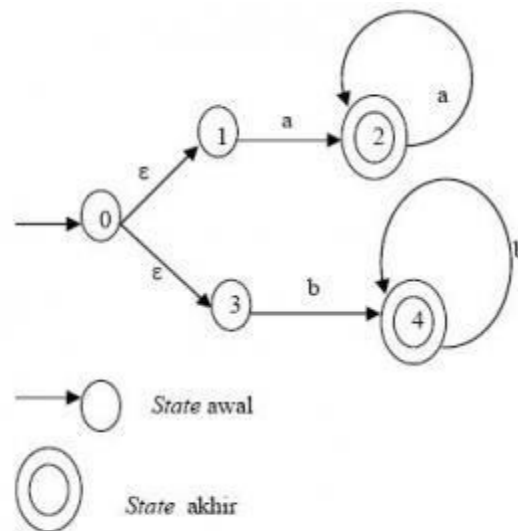
$Q$  = himpunan state(keadaan)

$\Sigma$  = Himpunan input  $q_0 \in Q$  adalah keadaan awal

$\delta = Q \times S \rightarrow Q$  adalah tabel transisi

$F$  = keadaan akhir

Suatu NFA dapat direpresentasikan dalam bentuk bagan sebagai suatu graf yang diberi label dan disebut dengan graf transisi. Dalam graf transisi ini nodal adalah state dan label dari sisi menyatakan fungsi transisi, contoh Graf transisi NFA dapat dilihat pada gambar 1.



Gambar 1. NFA penerima  $aa^* | bb^*$

Gambar 1. diatas mempunyai definisis formal sebagai berikut :

$Q = \{0,1,2,3,4\}$

$\Sigma = ,a,b- q_0 = 0$

$F = \{2, 4\}$

& = diagram transisi dapat dilihat pada tabel 1

Tabel 1. Diagram Transisi

$\delta$	a	b
0	$\emptyset$	$\emptyset$
1	2	$\emptyset$
2	2	$\emptyset$
3	$\emptyset$	4
4	$\emptyset$	4

## 2.5 Konsep Dasar Bahasa Formal

1. Teori bahasa membicarakan bahasa formal (*formal language*), terutama untuk kepentingan perancangan kompilator (*compiler*) dan pemroses naskah (*text processor*).
2. Bahasa formal adalah kumpulan *kalimat*. Semua kalimat dalam sebuah bahasa dibangkitkan oleh sebuah tata bahasa (*grammar*) yang sama.
3. Sebuah bahasa formal bisa dibangkitkan oleh dua atau lebih tata bahasa berbeda.
4. Dikatakan bahasa formal karena grammar diciptakan mendahului pembangkitan setiap kalimatnya.
5. Bahasa Natural/manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya 'bahasa formal' akan disebut 'bahasa' saja.

## 2.6 Konsep Bahasa dan Automata

Ada beberapa konsep-konsep bahasa yang ada didalam teori automata, yaitu :

1. Anggota alfabet dinamakan simbol terminal
2. Kalimat adalah deretan hingga simbol - simbol terminal
3. Bahasa adalah himpunan Kalimat – Kalimat
4. String adalah suatu deretan berhingga dari simbol - simbol , contoh : 'a','b','c' adalah simbol dan 'abc' adalah sebuah string
5. Simbol -Simbol terminal . Seperti : Huruf kecil (a,b,c) , Simbol Operator (+ dan \*) , Simbol tanda baca ( , dan ; ) dan String yang bercetak tebal . Contohnya adalah if , then dan else
6. Simbol - simbol non terminal / Variabel .Seperti : Huruf besar (A, B, C) , huruf S sebagai simbol awal , String yang tercetak miring .
7. Huruf yunani melambangkan string yang tersusun atas simbol - simbol terminal atau simbol - simbol non terminal atau campuran keduanya , misalnya pada gambar di bawah ini :

$\alpha, \beta$ , dan  $\epsilon$  .

8. Sentensial adalah string yang tersusun atas simbol-simbol terminal atau simbol-simbol non terminal atau campuran keduanya.
9. Derivasi adalah proses pembentukan sebuah kalimat atau sentensial . Sebuah derivasi dilambangkan sebagai berikut :

$$\alpha \Rightarrow \beta$$

## 2.7 Elemen Bahasa Formal

Elemen-elemen Bahasa adalah Alphabet, Grammar (Tata Bahasa) dan Semantic.

- **Alphabet** adalah himpunan terhingga dari token-token dimana kalimat dibentuk dalam suatu bahasa. **Alpabet** Adalah himpunan simbol (karakter) tak kosong yang berhingga. Alpabet digunakan untuk membentuk kata-kata (string-string) di bahasa. Bahasa dimulai dengan alpabet. Pada beberapa buku, alpabet dilambangkan dengan

$\Sigma$  Istilah huruf, karakter dan simbol adalah *sinonim* menunjukkan elemen alpabet. Jika simbol berbaris bersebelahan, maka diperoleh "string simbol". Istilah kalimat, kata dan string adalah *sinonim*

Contoh :

$\{a,b\} \rightarrow$  Himpunan yang terdiri dari simbol "a" dan "b".

- **Grammar** (Tata Bahasa) adalah himpunan dari aturan-aturan structural yang didefinisikan yang berlaku dalam suatu kalimat pada token-token.

Grammar G didefinisikan sebagai pasangan 4 tuple :  $V_t$  ,  $V_n$  , S, dan P, dan dituliskan sebagai  $G(V_t, V_n, S, P)$ , dimana :

- $V_t$  : himpunan simbol-simbol terminal (alfabet) = kamus
- $V_n$  : himpunan simbol-simbol non terminal
- S C V : simbol awal (atau simbol start)
- P : himpunan produksi

Contoh :

- $G_1 : V_T = \{I, \text{want}, \text{need}, \text{You}\}, V = \{S, A, B, C\}, P = \{S \rightarrow ABC, A \rightarrow I, B \rightarrow \text{want} \mid \text{need}, C \rightarrow \text{You}\}$   
 $S \rightarrow ABC$

→ IwantYou

→  $L(G1) = \{IwantYou, IneedYou\}$

2.  $G2 : VT = \{a\}, V = \{S\}, P = \{S \rightarrow aS \mid a\}$

$S \rightarrow aS$

$\rightarrow aaS$

$\rightarrow aaa \quad L(G2) = \{a^n \mid n \geq 1\}$

$L(G2) = \{a, aa, aaa, aaaa, \dots\}$

- **Semantic** adalah himpunan aturan-aturan yang didefinisikan yang mempunyai efek operasional pada setiap program yang ditulis dalam bahasa apabila ditranslasi dan dieksekusi pada suatu mesin.

Kalimat dalam bahasa Inggris dikonstruksi dari himpunan karakter yang terdiri dari huruf, angka, spasi dan tanda-tanda baca. Karakter-karakter ini dibentuk menjadi kata melalui aturan-aturan ejaan dan kamus, dan kemudian kata-kata dibentuk menjadi kalimat berdasarkan aturan-aturan tata bahasa.

Suatu program komputer dapat dikonstruksi dengan cara yang sama dari urutan karakter yang terdapat pada himpunan karakter dari komputer tersebut. Perbedaan yang prinsip antara Bahasa Inggris dengan Bahasa pemrograman komputer adalah bahwa aturan-aturan ejaan dan tata bahasa dalam bahasa Inggris sangat kompleks dan banyak pengecualian dan keragu-raguan, sementara dalam bahasa pemrograman harus mempunyai struktur yang tepat dan pasti.

## BAB III. PENUTUP

### 3.1. Kesimpulan

Kesimpulan yang dapat di ambil dari makalah ini dibagi menjadi beberapahal dari sejarah hingga dari konesp otomata itu sendiri.

Otomata bermula sebelum komputer ada pada teori di bidang sistem logika matematika atau formal, ilmuwan David Hilbert telah mencoba menciptakan algoritma umum untuk pembuktian (seluruh) persoalan matematika secara otomatis yaitu mampu menentukan salah benarnya sembarang prosisi matematika.

Di teori bahasa otomata juga dijelaskan beberapa hal sebagai berikut :

- Teori bahasa membicarakan bahasa formal (*formal language*), terutama untuk kepentingan perancangan kompilator (*compiler*) dan pemroses naskah (*text processor*).
- Bahasa formal adalah kumpulan *kalimat*. Semua kalimat dalam sebuah bahasa dibangkitkan oleh sebuah tata bahasa (*grammar*) yang sama.
- Sebuah bahasa formal bisa dibangkitkan oleh dua atau lebih tata bahasa berbeda.
- Dikatakan bahasa formal karena grammar diciptakan mendahului pembangkitan setiap kalimatnya.
- Bahasa Natural/manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya ‘bahasa formal’ akan disebut ‘bahasa’ saja.

Automata adalah suatu mesin sekuensial (otomatis), yang menerima input (dari pita masukan ) dan mengeluarkan output, keduanya dalam bentuk diskrit. Automata mempunyai sifat-sifat



Terdapat juga Konsep Dasar Bahasa Formal Konsep Bahasa dan Automata Elemen Bahasa Formal yang di jelaskan pada bab sebelumnya yang dimana sudah dijelaskan secara rinci sesuai dengan sub bab yang di berikan.

### **3.2. Saran**

Sebagai mana yang sudah di jelaskan saran dari kami kedepannya tentang teori bahasa dan otomata adalah lebih menjurus pada dasar dasar pengetahuan. Karena jika tidak di bekali pengetahuan dasar sulit untuk mengembangkan kembali apa yang sudah ada.

(Rizqia, 2012)

## DAFTAR PUSTAKA

Heru, C. (2004). *Teori bahasa dan otomata*. 1–63.

Rizqia, C. (2012). *Modul Teori Bahasa Dan Automata*. Retrieved from  
<https://aqilvsibnu.files.wordpress.com/2013/10/teori-bahasaautomata-2.pdf>