

# Aplikasi Graf pada Representasi Deterministic Finite Automata

I Gede Govindabhakta - 13519139<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13519139@std.stei.itb.ac.id

**Abstract**—Deterministic Finite Automata atau DFA adalah salah satu dari Finite State Machine yang dipelajari pada cabang ilmu Teori Bahasa Formal dan Automata. DFA digunakan untuk berbagai mesin dan automata sederhana seperti *vending machine*. Secara formal, sebuah DFA didefinisikan dengan 5 tuple yang melingkupi keseluruhan state, alfabet, fungsi transisi, state awal, dan state akhir. Untuk membantu memahami suatu DFA lebih baik, digunakan graf untuk memvisualisasi kerja dan fungsi dari DFA tersebut.,

**Keywords**—Automata, DFA, Graf, Visualisasi

## I. PENDAHULUAN

Teori Bahasa Formal dan Automata (TBFO) adalah cabang ilmu yang mempelajari perangkat komputasi abstrak atau “mesin” [1]. TBFO muncul pada abad ke-20 sebagai cabang ilmu komputer teoritis dengan tujuan mengabstraksikan komputasi menjadi automata hingga dapat melakukan atau mengimitasi cara kerja manusia.

Mesin yang dibahas didalam TBFO berfungsi untuk menerima sebuah *language* seperti *Regular Language*, *Context Free Language*, dll. Contoh dari penerapan TBFO ini adalah compiler, yang membaca kode dalam bahasa tertentu dan mengubahnya menjadi program yang bisa dijalankan oleh komputer.

Deterministic Finite Automata atau DFA adalah salah satu mesin yang paling sederhana yang dipelajari dalam TBFO. Karena keterbatasannya, DFA bisa digunakan untuk compiler untuk bahasa yang sangat sederhana. Penerapannya terdapat pada mesin yang sederhana seperti *vending machine*. Akan tetapi, DFA adalah mesin yang sangat mendasar yang perlu dipelajari dan dikuasai untuk mendapati fondasi kuat untuk dapat memahami mesin-mesin lainnya yang lebih rumit.

## II. LANDASAN TEORI

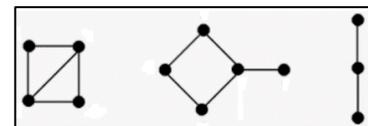
### A. Graf

#### a. Definisi Graf

Graf terdiri dari simpul (vertex atau node) dan sisi (edge) yang digunakan bersama untuk menunjukkan objek diskrit dan hubungan antara objek-objek tersebut. Secara matematis, graf didefinisikan dalam bentuk  $G(V, E)$  dimana  $V$  adalah himpunan dari semua vertex dan  $E$  adalah himpunan dari semua edge pada graf. Sebuah edge menghubungkan dua buah vertex.

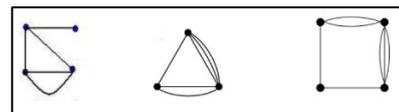
Graf dapat dikelompokkan berdasarkan berbagai atribut dan sifat. Berdasarkan ada atau tidaknya gelang (edge yang menghubungkan sebuah vertex dengan dirinya sendiri) dan edge ganda (dua atau lebih edge yang identik), graf dibagi menjadi graf sederhana dan graf tidak sederhana.

Graf sederhana adalah graf yang tidak memiliki gelang maupun edge ganda.

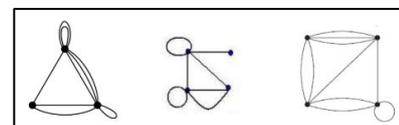


Gambar 2.1 Graf Sederhana [2]

Graf tak sederhana adalah yang memiliki gelang atau edge ganda. Berdasarkan adanya gelang atau sisi ganda, graf tak sederhana dibagi menjadi graf ganda (memiliki 1 atau lebih edge ganda) dan graf semu (memiliki 1 atau lebih gelang).

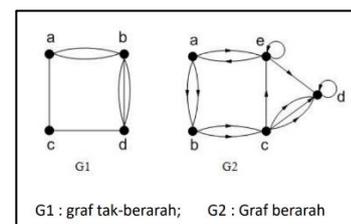


Gambar 2.2 Graf Ganda [2]



Gambar 2.3 Graf Semu [2]

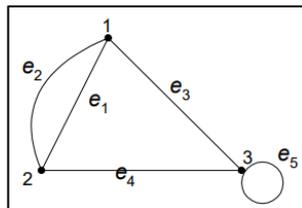
Berdasarkan adanya atribut arah pada edge, graf dibagi menjadi graf tidak berarah dan graf berarah.



Gambar 2.4 Graf Tak-berarah dan Graf Berarah [2]

*b. Terminologi Graf*

1. **Adjacent (bertetangga)**  
Dua buah vertex a dan b dikatakan bertetangga jika terdapat sebuah edge yang menghubungkan a dengan b.
2. **Incidency (bersisian)**  
Pada sebuah edge yang menghubungkan a dengan b, edge tersebut dikatakan bersisian dengan a dan juga bersisian dengan b.
3. **Isolated Vertex**  
Isolated Vertex adalah vertex dengan derajat 0.
4. **Graf Kosong**  
Sebuah graf dimana E adalah himpunan kosong disebut sebagai graf kosong. Graf kosong direpresentasi dengan  $N_n$ .
5. **Derajat**  
Jumlah edge yang terhubung dengan suatu vertex adalah derajat dari vertex tersebut. Pada graf berarah, derajat masuk dan keluar dibedakan. Berdasarkan Lemma Jabat Tangan, jumlah total derajat dari semua vertex harus genap, atau dalam kata lain, jumlah vertex dengan derajat ganjil harus genap.
6. **Lintasan (Path)**  
Lintasan adalah rangkaian vertex dan edge yang dimana setiap vertex diikuti oleh sebuah edge yang bersisian kemudian dilanjutkan dengan vertex bersisian lainnya dari edge tersebut. Panjang sebuah lintasan adalah banyaknya edge pada lintasan tersebut. Sebagai contoh, lintasan yang terdiri dari 2,  $e_2$ , 1,  $e_3$ , 3,  $e_5$ , 3 pada gambar 2.5 dibawah adalah 3.

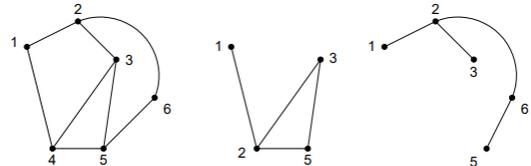


Gambar 2.5 [2]

7. **Siklus (Cycle) atau Sirkuit (Circuit)**  
Siklus atau sirkuit adalah sebuah lintasan yang berawal dan berakhir pada vertex yang sama. Sebagai contoh, lintasan yang terdiri dari 2,  $e_2$ , 1,  $e_1$ , 2 adalah sebuah sirkuit dengan panjang 2.
8. **Keterhubungan (Connected)**  
Dua buah vertex dikatakan terhubung jika terdapat sebuah lintasan yang menghubungkan kedua vertex tersebut. Sebuah graf dikatakan terhubung jika semua simpul pada graf terhubung (connected graph). Jika terdapat sepasang simpul yang tidak terhubung, graf tersebut disebut graf tak-terhubung (disconnected graph).  
Graf berarah adalah graf terhubung jika graf tidak berarahnya (diperoleh dengan menghilangkan arah dari setiap edgenya) adalah graf terhubung. Dua simpul a dan b dikatakan terhubung kuat jika terdapat lintasan dari a ke b dan juga dari b ke a, jika tidak maka dikatakan terhubung lemah. Pada graf terhubung kuat,

setiap vertex memiliki lintasan ke semua vertex lainnya.

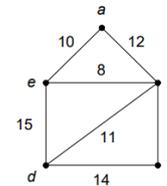
9. **Upagraf (Subgraf) dan Komplemen Upagraf**  
Graf G adalah upagraf dari graf A jika  $V_G$  adalah subset dari  $V_A$  dan  $E_G$  adalah subset dari  $E_A$ . Komplemen dari upagraf G, P, adalah graf dimana  $V_P + V_G = V_A$  dan  $E_P + E_G = E_A$ . Jika Upagraf G hanya memiliki himpunan vertex yang sama, G dikatakan upagraf rentang dari A.



(a) Graf  $G_1$  (b) Sebuah upagraf (c) komplemen dari upagraf (b)

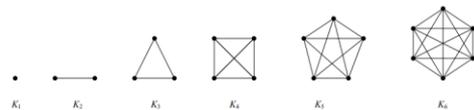
Gambar 2.6 Contoh upagraf dan komplemen upagraf [2]

10. **Cut-Set**  
Cut-Set dari graf terhubung G adalah himpunan sisi yang perlu dihilangkan agar G menjadi graf tidak terhubung.
11. **Graf Berbobot (Weighted Graph)**  
Graf berbobot adalah graf yang semua sisinya diberi suatu nilai bobot yang berfungsi untuk memberi kuantitas/nilai skalar pada hubungan antara vertex-vertex yang bersisian dengan edge.



Gambar 2.7 Contoh Graf Berbobot

12. **Graf Khusus**
  - a. **Graf Lengkap**  
Graf sederhana dimana setiap simpul bersisian dengan semua simpul lainnya. Memiliki jumlah sisi sebanyak  $n(n-1)/2$  dimana n adalah jumlah simpul. Oleh karena itu, derajat dari semua simpul pada graf adalah sama. Dilambangkan dengan  $K_n$ .



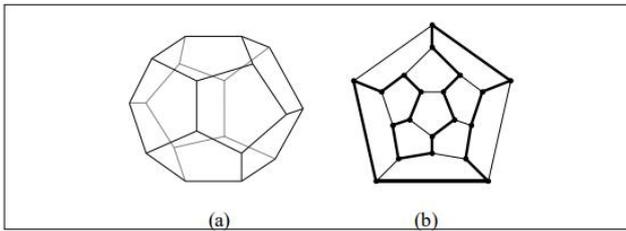
Gambar 2.8 Contoh Graf Lengkap

- b. **Graf Lingkaran**  
Graf lingkaran adalah graf dimana derajat semua simpulnya adalah 2. Graf lingkaran dilambangkan dengan  $C_n$  dimana n adalah jumlah simpul.
- c. **Graf Teratur (Regular Graphs)**  
Jika pada suatu graf, derajat semua simpulnya sama, graf tersebut disebut graf teratur. Jumlah sisi pada graf teratur adalah  $nr/2$  dimana n adalah jumlah simpul dan r adalah derajat

semua simpul. Semua graf lingkaran adalah graf teratur.

13. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton adalah lintasan yang melalui semua vertex dalam graf tepat sekali. Sirkuit Hamilton adalah lintasan Hamilton yang mulai dan berakhir pada vertex yang sama. Graf yang memiliki lintasan Hamilton tetapi tidak mempunyai sirkuit Hamilton. [3]

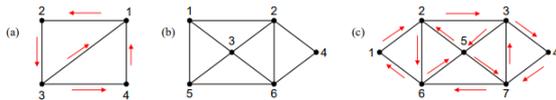


(a) Dodecahedron Hamilton,  
(b) graf yang mengandung sirkuit Hamilton

Gambar 2.9 Sirkuit dan Lintasan Hamilton.

14. Lintasan dan Sirkuit Euler

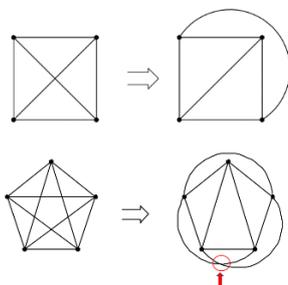
Lintasan Euler adalah lintasan yang melalui semua sisi tepat sekali. Sirkuit Euler adalah lintasan Euler yang mulai dan berakhir pada simpul yang sama. Graf yang mempunyai lintasan Euler dinamakan graf Euler dan graf yang mempunyai hanya lintasan Euler disebut graf semi-Euler.



Gambar 2.10 Contoh lintasan dan sirkuit euler [3]

15. Graf Planar

Graf Planar adalah graf yang dapat digambarkan pada bidang datar dengan sisi-sisi tidak saling memotong (bersilangan)



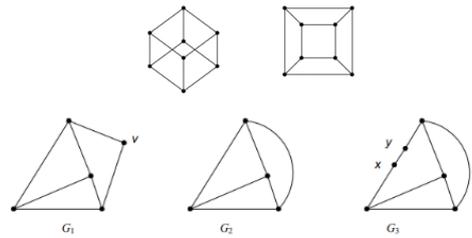
Gambar 2.9 Contoh graf planar dan tidak planar

16. Isomorfisme dan Homeomorfisme

Dua buah graf disebut isomorfis jika kedua graf adalah graf yang sama tetapi dengan representasi geometris berbeda.

Dua buah graf A dan B disebut homeomorfik jika terdapat himpunan vertex pada A yang semuanya berderajat 2 yang jika dihilangkan (dan vertex yang

bersisian disambungkan) dapat menghasilkan graf B.



Gambar 2.10 Dua Buah Graf Isomorfik (atas) [2] dan 3 Buah Graf Homeomorfik (bawah) [2]

B. Deterministic Finite Automata (DFA)

a. Definisi DFA

Sebuah DFA didefinisikan dengan 5 buah tuple yaitu himpunan state yang terhingga (Q), alfabet dari input ( $\Sigma$ ), fungsi transisi ( $\delta$ ), state awal (umumnya  $q_0$ , anggota Q), dan himpunan final state (F yang merupakan subset dari Q). Fungsi dari DFA adalah menerima sebuah string input dan menerima atau menolak string tersebut berdasarkan language yang diterima oleh DFA tersebut. Cara kerja DFA adalah ia memulai dari karakter pertama pada string input, memasukkannya kedalam fungsi transisi, dan berpindah state hingga semua karakter pada string input telah diproses. Beda dengan pushdown automata dan mesin turing, DFA sebagai sebuah finite state machine tidak memiliki memori seperti stack ataupun Random Access Memory (RAM).

$$Q = \{ 000, 001, 011y, 011n, 111y, 111n, 110, 100y, 100n, 101, 010 \}$$

Dimana untuk state  $x_1x_2x_3$ , x dan z menunjukkan jika  $x_1$  dan  $x_3$  menghadap ke  $x_2$  (1 jika iya dan 0 jika tidak) dan y menunjukkan jika  $x_2$  menghadap ke D atau c (1 jika D dan 0 jika C)

$$\Sigma = \{ A, B \}$$

$\delta$	A	B
001	101n	010
101n	011n	110
101y	011n	110
010	110	011y
011n	111n	000
011y	111n	000
110	000	111y
111n	001	100y
111y	001	100y
000	100n	001
100n	010	101y
100y	010	101y

$$q_0 = 001 \text{ ( } x_1 \text{ tidak menghadap } x_2, x_2 \text{ menghadap C, } x_3 \text{ menghadap } x_2 \text{)}$$

$$F = \{ 001, 011y, 000, 111y, 100y \}$$

Dapat direpresentasi sebagai fungsi  $f(a_n, \delta^A(a_1a_2 \dots a_{n-1}))$  untuk semua string dengan panjang 0 sampai 5.

Dimana

$$f(a_n, y) = \begin{cases} \text{rejected, jika } (a_n = A \text{ dan } y = 0x_2x_3) \\ \text{atau } (a_n = A \text{ dan } y = 10x_3) \\ \text{atau } (a_n = B \text{ dan } y = x_101) \\ \text{accepted, jika } (a_n = B \text{ dan } y = x_1x_20) \\ \text{atau } (a_n = A \text{ dan } y = 11x_3) \\ \text{atau } (a_n = B \text{ dan } y = x_111) \end{cases}$$

Gambar 2.6 Contoh DFA

1. State

Sebuah state menunjukkan kondisi, kombinasi, atau konfigurasi mesin pada suatu saat dalam beroperasi. Sebagai contoh, sebuah state dapat menunjukkan bahwa input terakhir dari string yang telah diproses adalah sebuah 0 atau sebuah state juga dapat menandakan bahwa semua input yang sudah diproses telah membentuk suatu string yang diterima oleh language dari DFA tersebut.

Pada saat DFA mulai beroperasi dan membaca string input, ia mulai pada start state (yang biasanya ditandai dengan  $q_0$ ) yang merupakan anggota dari  $Q$ . Sebuah DFA hanya bisa memiliki 1 start state.

Final state menunjukkan bahwa semua input yang telah dibaca sudah membentuk sebuah string yang diterima oleh language DFA. Jika pembacaan input selesai dengan state akhir dari mesin DFA adalah final state, maka string input diterima oleh DFA. Sebuah DFA dapat memiliki lebih dari satu final state dan start state dari sebuah DFA boleh merupakan sebuah final state.

2. Alfabet dan String

Alfabet adalah sebuah himpunan simbol yang terhingga yang digunakan oleh language yang diterima oleh DFA seperti alfabet biner ( $\Sigma = \{0, 1\}$ ).

Sebuah string adalah sebuah rangkaian dari simbol.  $\Sigma^*$  menunjukkan himpunan dari semua string yang mungkin dari alphabet  $\Sigma$ .

Power dari sebuah alfabet ( $\Sigma^k$ ) adalah himpunan dari semua string dengan panjang  $k$  dari himpunan simbol  $\Sigma$  seperti  $\Sigma^2$  dari alfabet biner adalah  $\{01, 11, 10, 00\}$ .

3. Fungsi Transisi

Fungsi transisi,  $\delta(q, a)$  (ditandai dengan delta), menerima input  $q$ , suatu state, dan  $a$ , sebuah simbol, dan memberikan output suatu state tujuan. Sebuah DFA harus memiliki fungsi transisi untuk semua simbol pada setiap state. Contoh representasi fungsi transisi adalah dengan tabel.

Final states starred		0	1	← Columns = input symbols
→ * A	A	B		
Arrow for start state * B	A	C		
C	C	C		
↑ Rows = states				

Gambar 2.7 Tabel transisi untuk sebuah automata

Hasil sebuah fungsi transisi dapat dimasukkan kedalam fungsi transisi lainnya sehingga menjadi extended-delta, seperti  $\delta(\delta(q, a), b)$ . Untuk meringkas penulisannya, extended-delta dapat ditulis sebagai delta-hat, seperti  $\hat{\delta}(q, ab)$ .

b. Language

Language adalah sebuah subset dari  $\Sigma^*$ , biasa dilambangkan dengan  $L$ . Terdapat berbagai macam language yang dapat diterima oleh suatu mesin seperti *Regular Language* dan

*Context-Free Language*. DFA sebagai sebuah *finite automata* menerima  $L$  yang merupakan *Regular Language*. Contoh dari suatu language adalah semua string dari alfabet biner yang berawal dengan 00.

Karena *Regular Language* adalah language yang diterima oleh finite state machine, ia tidak bisa memanfaatkan memori dalam bentuk apapun. Salah satu contoh kata yang bukan bagian dari *Regular Language* adalah palindrom (kata yang bisa dibaca dari depan dan belakang).

III. APLIKASI GRAF PADA REPRESENTASI DETERMINISTIC FINITE AUTOMATA

A. Translasi DFA Menjadi Graf

Manfaat terbesar dari representasi DFA dengan graf adalah dengan memvisualisasikan cara kerja dan alur pikir sebuah DFA. Komponen untuk membangun sebuah graf dari sebuah DFA sudah ada pada tuple yang mendefinisikan DFA tersebut. Himpunan  $Q$ , yang merupakan himpunan semua state yang ada pada DFA berperan sebagai  $V$ , himpunan semua vertex, pada graf. Fungsi transisi ( $\delta$ ) berperan untuk mengisi himpunan  $E$  dimana berlaku:

$$E = \{ (q, \delta(q, X)) \mid q \in Q \text{ dan } X \in \Sigma \}$$

Untuk semua  $q$  dalam  $Q$  dan  $X$  dalam  $\Sigma$ . Mirip dengan graf berbobot, setiap edge ditandai dengan simbol input  $X$  yang berkorespondensi dengan edge tersebut.

Konvensi yang digunakan dalam representasi DFA dengan graf adalah untuk menandakan start state dengan satu tanda panah menuju ke vertex start state tersebut. Selain itu, semua final state ditandai dengan vertex khusus yang dilingkari dua kali.

Graf yang merepresentasi DFA sudah melingkupi kelima dari tuple yang mendefinisikan DFA, akan tetapi representasi graf juga dapat digunakan sebagai representasi fungsi transisi saja meskipun melingkupi keempat tuple lainnya.

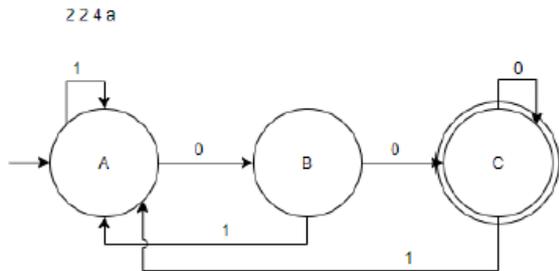
Semua string yang berakhir dengan 00

DFA  $(Q, \Sigma, \delta, q_0, F)$

$Q = \{A, B, C\}$

$\Sigma = \{0, 1\}$

$\delta$



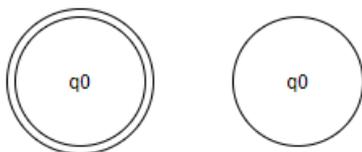
$q_0 = A$

$F = \{C\}$

Gambar 3.1 Representasi DFA sebagai Graf

**B. Sifat-Sifat Graf pada DFA**

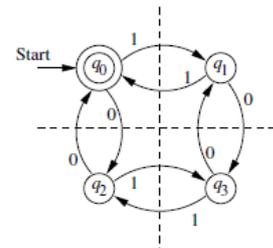
1. Graf Sederhana dan Tidak Sederhana  
Graf representasi DFA dapat membentuk graf sederhana maupun graf tidak sederhana.
2. Graf Berarah  
Semua graf representasi DFA adalah graf berarah.
3. Lintasan dan Sirkuit  
Setiap lintasan pada representasi graf DFA dapat direpresentasikan sebagai sebuah delta-hat  $\hat{\delta}(q, x)$  dimana q adalah state atau vertex awal dari lintasan dan x adalah string input yang diperlukan untuk mencapai state tujuan dari lintasan. Sebagai contoh, pada Gambar 3.1, lintasan  $A \rightarrow B \rightarrow C$  sama dengan  $\hat{\delta}(A, 00)$ .
4. Keterhubungan dan Isolated Vertex  
Semua graf representasi DFA adalah graf terhubung, tetapi tidak semuanya adalah graf terhubung kuat. Karena semua graf representasi DFA adalah graf terhubung, tidak mungkin ada isolated vertex.
5. Graf Kosong  
Hanya ada dua kemungkinan untuk graf representasi DFA yang merupakan graf kosong yaitu DFA yang menerima semua string pada alphabet atau yang menolak semua string.



Gambar 3.2 Graf Kosong Representasi DFA

6. Sirkuit dan Siklus  
Jika sebuah language sebuah DFA tidak memiliki restriksi dari panjang string yang diterima, maka untuk representasi graf DFA tersebut, dijamin bahwa minimal ada 1 sirkuit/siklus. Ini dikarenakan adanya

keperluan untuk looping yang jumlahnya tidak tertentu sehingga adanya siklus atau sirkuit adalah sebuah keharusan.



Gambar 3.3 Representasi Graf dari DFA yang menerima string dengan jumlah 0 dan 1 yang genap mempunyai lebih dari satu siklus atau sirkuit

7. Isomorfisme dan Homeomorfisme  
Isomorfisme dan Homeomorfisme pada graf hasil representasi DFA tidak bermakna.
8. Lintasan dan Sirkuit Hamilton  
Graf representasi DFA dapat membentuk graf Hamilton tetapi tidak semua graf representasi DFA adalah graf Hamilton.
9. Lintasan dan Sirkuit Euler  
Graf representasi DFA dapat membentuk graf euler tetapi tidak semua graf representasi DFA adalah graf euler.
10. Graf Planar  
Graf representasi DFA dapat membentuk graf planar dan graf tidak planar.
11. Graf Lingkaran dan Graf Teratur  
Graf DFA dapat membentuk graf teratur tetapi tidak mungkin (dengan syarat alfabet minimal 2) membentuk graf lingkaran karena jumlah sisi dari graf representasi DFA pasti  $n(i)$  dimana n adalah jumlah state dan i adalah banyaknya simbol didalam alfabet.

**V. KESIMPULAN**

Representasi DFA dengan graf membantu memahami cara kerja dan alur pikir dari sebuah DFA. Membaca sebuah graf representasi DFA jauh lebih mudah dibanding membaca tabel transisi dari sebuah DFA. Oleh karena itu, representasi DFA dengan graf jauh lebih menguntungkan.

**VII. UCAPAN TERIMA KASIH**

Penulis berterima kasih kepada Tuhan dan seluruh Dosen pada Matkul Matematika Diskrit karena atas bimbingan dan restunya dapat menyelesaikan makalah ini.

**REFERENCES**

[1] Hopcroft, John E.; Rajeev, Jeffrey, "Automata Theory, Languages, and Computation," Boston, MA, Addison-Wedly, 2006.  
 [2] <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf2020-Bagian1> diakses tanggal 8 Desember 2020  
 [3] <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf2020-Bagian3> diakses tanggal 8 Desember 2020

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2020

Ttd (scan atau foto ttd)

A handwritten signature in blue ink on a light gray background. The signature is stylized and appears to be 'L. S. J.' with a horizontal line underneath.

Nama dan NIM