Implementasi Transisi Finite State Automata dengan Aplikasi Mesin Abstrak DFA dan NFA Berbasis Android

[1]Syahbani Farhan, [2]Iskandar Fitri, [3]Fauziah
[1,2,3]Program Studi Informatika
[1,2,3] Jl. Sawo Manila Pasar Minggu, Pejaten, Jakarta Selatan
email : [1]syahbanifarhan@gmail.com, [2]iskandar.fitri@civitas.unas.ac.id,[3]fauziah@civitas.unas.ac.id

ABSTRACT
Language and automata are important sciences that make them the main components in making software. If one of them does not have these components, there will be an imbalance or even fail to make a software. Because language is a tool used daily in communicating both to humans and computers. While automata is a process that can operate alone, by having a theory about abstract machines and is closely related to language. Automata has a relationship in the form of grammar, where the grammar is an abstract form that can reject if the input string we entered is wrong and accept if the input string we entered is correct to generate an output that matches the input. The purpose of this study is to test the theories of Language and Automata to be verified by making DFA and NFA abstract machine applications using Android Studio. The results shown by the Abstract Machine application that have been made are able to read and process numbers with an integer range of 2,147,483,647 and also the level of accuracy of reading data in each state process is 90.00% or 36 data that were successfully tested without experiencing errors from 40 data that have been tested on application.

Keywords - Finite State Automata, Abstract Machines, DFA, NFA, Android Studio, Automata

ABSTRAK
Bahasa dan automata merupakan suatu ilmu penting yang menjadikannya sebagai komponen-komponen utama dalam membuat suatu perangkat lunak atau software. Apabila salah satu diantaranya tidak memiliki komponen tersebut, maka akan terjadi ketidakseimbangan atau bahkan gagal untuk membuat suatu perangkat lunak atau software. Karena bahasa yaitu suatu alat yang digunakan sehari-hari dalam berkomunikasi baik itu kepada manusia maupun komputer. Sedangkan automata merupakan proses yang dapat beroperasi sendiri, dengan memiliki teori tentang mesin-mesin abstrak dan sangat berkaitan dengan bahasa. Automata memiliki keterkaitan yaitu berupa grammar, dimana grammar itu merupakan suatu bentuk abstrak yang dapat menolak jika input string yang kita masukkan salah dan menerima jika input string yang kita masukkan benar untuk membangkitkan suatu output yang sesuai dengan inputannya. Tujuan penelitian ini adalah untuk menguji teori-teori Bahasa dan Automata untuk dibuktikan kebenarannya dengan membuat aplikasi mesin abstrak DFA dan NFA dengan menggunakan Android Studio. Hasil yang ditunjukkan oleh aplikasi Abstract Machine yang telah dibuat yaitu dapat membaca dan memproses angka dengan range integer yaitu 2,147,483,647 dan juga tingkat akurasi pembacaan data pada setiap proses state sebesar 90,00% atau 36 data yang berhasil diuji tanpa mengalami error dari 40 data yang telah diujikan pada aplikasi.

Kata Kunci - Finite State Automata, Mesin Abstrak, DFA, NFA, Android Studio, Automata

1. Introduction

Automata merupakan suatu teori di bidang sistem logika matematika dimana seorang ilmuwan David Hilbert mencoba menciptakan algoritma umum untuk pembuktian persoalan tentang matematika secara otomatis. Pada zaman sekarang dimana perkembangan teknologi sangat pesat bahasa dan

Attribution-NonCommercial-ShareAlike CC BY-NC-SA

automata masih digunakan untuk membuat suatu perangkat yang dapat beroperasi secara otomatis, contohnya seperti vending machine, ATM dan lain sebagainya. Bahasa dan Automata merupakan suatu ilmu penting yang menjadikannya sebagai komponen-komponen utama dalam membuat suatu perangkat lunak atau software.

Bahasa yang banyak digunakan dalam dunia komputer seperti bahasa alami yang dikenal yaitu seperti bahasa Indonesia atau Inggris, tetapi bahasa dimaksudkan pada sintaks ini merupakan himpunan dari string symbol pada suatu alfabet. Jadi bahasa tersebut bisa saja tidak termasuk kedalam string, atau biasa disebut dengan himpunan yang kosong atau ø.[1] Namun, bahasa kosong ini berbeda dengan bahasa yang terdiri dari string kosong {ε}. Otomata adalah mesin abstrak yang dapat mengenali (recognize), menerima (accept), atau membangkitkan (generate) sebuah kalimat dalam bahasa tertentu. [2]

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 didalam bahasa dapat dibangkitkan oleh tata bahasa (grammar) yang sama. Yang artinya bahasa formal bisa saja dibangkitkan oleh dua atau lebih dengan tata bahasa yang berbeda. Finite State Automata (FSA) adalah suatu bentuk matematika pada sebuah system dengan input dan output yang memiliki jumlah berhingga state dan fungsi-fungsi transisi yang menyajikan perubahan state.[3]

Pada dasarnya Finite State Automata menggunakan pemikiran seperti neural network dan juga switching circuit. Finite State Automata juga melambangkan tool yang bermanfaat dalam membuat rancangan lexical analyzer, yaitu bagian pada kompilator yang berisi kumpulan karakter-karakter yang dikelompokkan menjadi token, atau unit-unit kecil contohnya seperti variable atau keyword. Saat menulis sistem pada kompilator dapat dilakukan secara otomatis yang nantinya akan mentransformasikan ekspresi regular ke finite state automata yang digunakan sebagai lexical analyzer. Finite state automata dan juga ekspresi regular yang dipakai pada text editor , pattern-matching, pemrosesan sebuah teks, dan pada aplikasi file-searching, dan juga sebagai salah satu konsep dari matematika untuk mengaplikasikan disiplin ilmu yang lain seperti logika.[4]

FSA sendiri merupakan kumpulan status dan kontrol yang bergerak pada satu state ke state lain yang memliki bentuk berhingga dan dijalankan automata (otomatis). Kontrol pada FSA itu dapat bersifat deterministic (dimana automata tak akan berada di dalam suatu state yang lebih dari satu disaat yang bersama-sama) atau non-deterministic (dimana automata bisa berada di dalam suatu state yang lebih dari satu disaat yang bersama-sama) sehingga FSA yang terbentuk menjadi 2 buah tipe yaitu : Deterministic Finite State Automata (DFA) dan Non-Deterministic Finite Automata (NFA). [5]

Automata adalah sebuah sistem dimana memiliki suatu fungsi dan juga berlaku di beberapa state, pada setiap state tersebut menjelaskan sebuah informasi yang menyinggung input dan juga dapat disebut dengan memori pada sebuah mesin. Pada dasarnya, konsep automata ini dapat menerima sebuah input , serta menghasilkan sebuah output, yang juga memiliki storage yang bersifat volatile serta mampu membentuk sebuah putusan akhir yang dapat mengubah suatu input beralih ke output.[6]

Oleh karena itu, Bahasa dan Automata merupakan suatu ilmu yang mendasar untuk menciptakan suatu rancangan perangkat lunak atau software serta perangkat kerasnya guna menghasilkan suatu mesin abstrak yang bermanfaat bagi kehidupan masyarakat.

2. Research Method

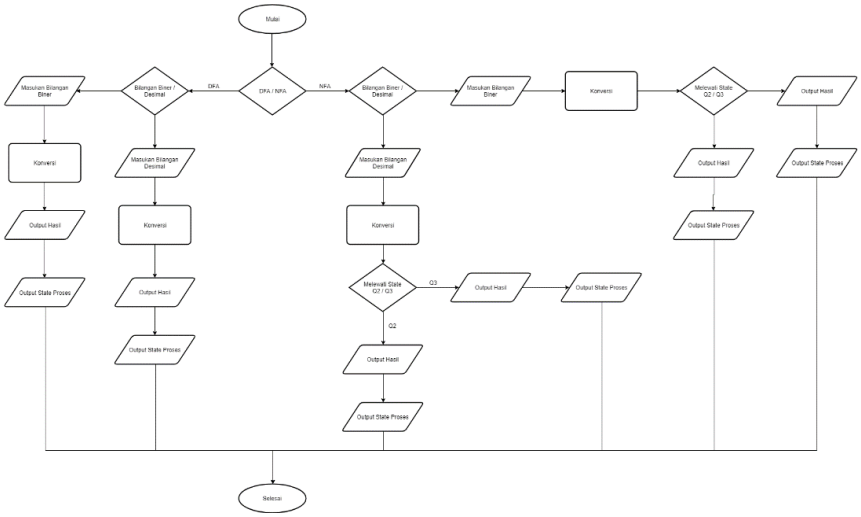
A. Metode Pengumpulan Data

Metode pengumpulan data yang dipakai pada pembuatan aplikasi berikut ini yaitu berawal dengan study literature yang membahas FSA dengan jelas dan nyata, dikarenakan hanya ada beberapa

jurnal serta buku yang membahas teori bahasa dan automata serta mesin-mesin yang mendukung dari teori-teori yang telah ada.

B. Sifat Penelitian

Penelitian ini bersifat percobaan input string dengan menguji coba mesin abstrak Deteriministic Finite Automata (DFA) dan Non-Deterministic Automata (NFA). Pada pengujian ini dengan membuat suatu aplikasi mesin abstrak Deteriministic Finite Automata (DFA) dan Non-Deterministic Automata (NFA), dimana didalam aplikasi tersebut digunakan untuk memverifikasikan bahwa string yang di-input mampu diterima atau ditolak serta menjelaskan tentang proses-proses yang terjadi pada mesin abstrak tersebut melalui model matematika yang telah digambarkan pada diagram transisi.



Gambar 1. Flowchart mesin abstrak DFA dan NFA untuk menentukan output suatu string yang di-input.

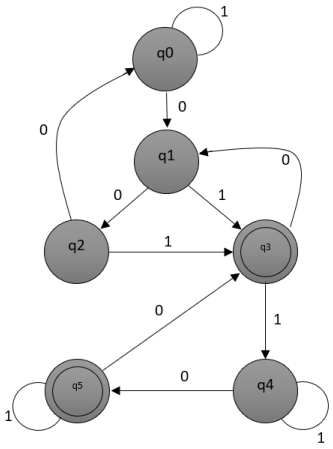
3. Result and Analysis

A. Rancangan Mesin Abstrak DFA

Pada Gambar 1 dibawah ini merupakan suatu diagram transisi dan tabel transisi dari mesin abstrak *Deterministic Finite Automata* (DFA).

tabel transisi dfa

δ	0	1
q0	q1	q0
q1	q2	q3
q2	q0	q3
q3	q1	q4
q4	q5	q4
q5	q3	q5



Gambar 2. Diagram Transisi Mesin Abstrak DFA

Attribution-NonCommercial-ShareAlike CC BY-NC-SA

Berikut merupakan tuple dari mesin abstrak DFA tersebut.
TUPLE Mesin Abstrak DFA

1. Q = Himpunan State
   Q = (q0 , q1 , q2 , q3 , q4 , q5)
2. Σ = Himpunan *Input*
   Σ = (0 , 1)
3. δ = Fungsi Transisi
   δ = δ (q0 , 0) = q1 δ (q0 , 1) = q0 δ (q1 , 0) = q2 δ (q1 , 1) = q3
   δ (q2 , 0) = q0 δ (q2 , 1) = q3 δ (q3 , 0) = q1 δ (q3 , 1) = q4
   δ (q4 , 0) = q5 δ (q4 , 1) = q4 δ (q5 , 0) = q3 δ (q5 , 1) = q5
4. S = State Awal
   S = {q0}
5. F = State Akhir
   F = {q3} dan {q5}

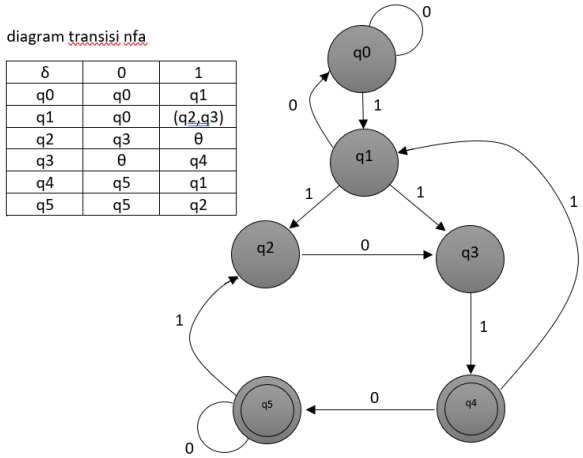
Dari fungsi transisi diatas maka dapat membuat tabel transisinya.

Tabel 1. Transisi Mesin Abstrak DFA

δ	**0**	**1**
q0	q1	q0
q1	q2	q3
q2	q0	q3
q3	q1	q4
q4	q5	q4
q5	q3	q5

B. Rancangan Mesin Abstrak NFA

Pada Gambar 2 dibawah ini merupakan suatu diagram transisi dan tabel transisi dari mesin abstrak *Non-Deterministic Finite State* Automata (NFA).



Gambar 2. Diagram Transisi Mesin Abstrak NFA

Berikut merupakan tuple dari mesin abstrak DFA tersebut.
TUPLE Mesin Abstrak NFA

1. Q = Himpunan State
   Q = (q0 , q1 , q2 , q3 , q4 , q5)
2. Σ = Himpunan Input
   Σ = (0 , 1)

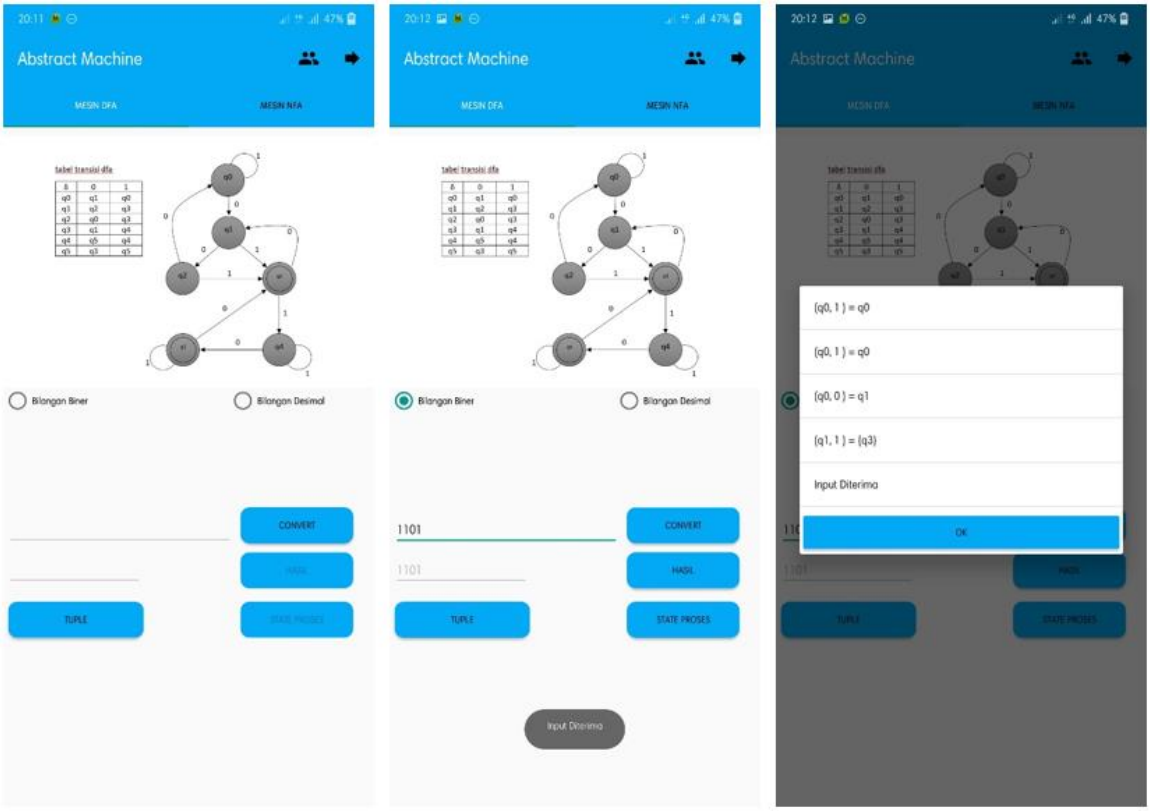
3. δ = Fungsi Transisi
   δ = δ (q0 , 0) = q0 δ (q0 , 1) = q1 δ (q1 , 0) = q0 δ (q1 , 1) = q2 / q3
   δ (q2 , 0) = q3 δ (q2 , 1) = θ δ (q3 , 0) = θ δ (q3 , 1) = q4
   δ (q4 , 0) = q5 δ (q4 , 1) = q1 δ (q5 , 0) = q5 δ (q5 , 1) = q2
4. S = State Awal
   S = {q0}
5. F = State Akhir
   F = {q4} dan {q5}

Dari Dari fungsi transisi diatas maka dapat membuat tabel transisinya.

Tabel 2. Transisi Mesin Abstrak NFA

δ	**0**	**1**
q0	q0	q1
q1	q0	(q2,q3)
q2	q3	θ
q3	θ	q4
q4	q5	q1
q5	q5	q2

C. Hasil Rancangan Mesin Abstrak DFA Berbasis Android



Gambar 3. Tampilan proses aplikasi *Abstract Machine Deterministic Finite State Automata* (DFA)

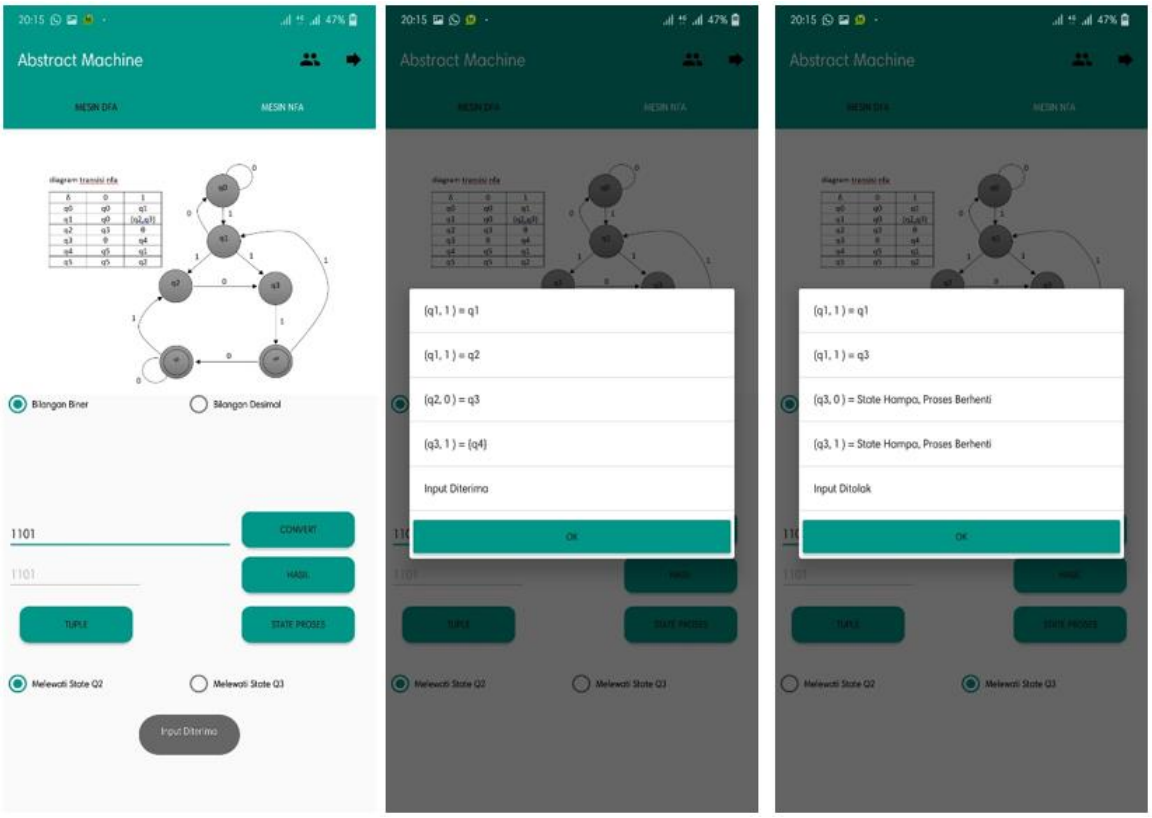
Cara Kerja dari Aplikasi mesin abstrak DFA yaitu sebagai berikut.

1. Pada tampilan awal dari aplikasi, pilih dari salah satu antara bilangan biner atau bilangan desimal, apabila user memilih bilangan biner maka input string berupa angka 1 dan 0. Sedangkan apabila user memilih bilangan desimal maka input string berupa angka 0 sampai 9.

Attribution-NonCommercial-ShareAlike CC BY-NC-SA

2. Lalu selanjutnya setelah memilih bilangan, user diharuskan meng-input bilangan yang user pilih sebelumnya lalu klik pada button convert. Button convert berfungsi sebagai pengubah dari bilangan desimal ke bilangan biner, apabila user memilih bilangan desimal sebagai input-an user.
3. Setelah itu klik button hasil, nanti akan muncul apakah inputan string user diterima oleh mesin abstrak atau tidak.
4. Lalu apabila user ingin mengetahui bagaimana proses state yang berjalan, cukup klik button state proses, nanti akan muncul proses dari setiap state yang dilewati.

D. Hasil Rancangan Mesin Abstrak NFA Berbasis Android



Gambar 3. Tampilan proses aplikasi *Abstract Machine Non-Deterministic Finite State Automata* (NFA)

Cara Kerja dari Aplikasi mesin abstrak DFA yaitu sebagai berikut.

1. Pada tampilan awal dari aplikasi, pilih dari salah satu bilangan biner atau bilangan desimal, apabila user memilih bilangan biner maka input string berupa angka 1 dan 0. Sedangkan apabila user memilih bilangan desimal maka input string berupa angka 0 sampai 9.
2. Lalu selanjutnya setelah memilih bilangan, user diharuskan menginput bilangan yang user pilih sebelumnya lalu klik pada button convert. Button convert berfungsi sebagai pengubah dari bilangan desimal ke bilangan biner, apabila user memilih bilangan desimal sebagai inputan user.
3. Kita diharuskan memilih jalur state mana yang akan user lewati melewati state q2 atau q3. Apabila user tidak memilih maka hasil output akan ditolak dan memiliki keterangan “Tidak memilih State yang ingin dilewati!”.
4. Setelah itu klik button hasil, nanti akan muncul apakah input-an string user diterima oleh mesin abstrak atau tidak.

5. Lalu apabila user ingin mengetahui bagaimana proses state yang berjalan, cukup klik button state proses, nanti akan muncul proses dari setiap state yang dilewati. User akan melihat input-an string user ditolak apabila user meng-input string tertentu yang menghasilkan state hampa.

E. Hasil Pengujian

Hasil pengujian dilakukan pada beberapa input yang telah diuji pada sebuah aplikasi Abstract Machine. Pada aplikasi Abstract Machine DFA, string input diterima pada state q3 dan q5. Sedangkan pada aplikasi Abstract Machine NFA, string input diterima pada state q4 dan q5.

Tabel 3. Hasil Pengujian *Abstract Machine Deterministic Finite State Automata* (DFA)

No	Desimal	State Akhir	Hasil Pengujian
1	38	q5	String Diterima
2	108	q3	String Diterima
3	3.000	q2	String Ditolak
4	40.000	q0	String Ditolak
5	500.000	q2	String Ditolak
6	3.000.000	q2	String Ditolak
7	20.000.000	q1	String Ditolak
8	100.000.000	q2	String Ditolak
9	2.000.000.000	q0	String Ditolak
10	60.000.000.000	-	Aplikasi Berhenti

Tabel 4. Hasil Pengujian *Abstract Machine Non-Deterministic Finite State Automata* (NFA)

No	Desimal	Melalui State	State Akhir	Hasil Pengujian
1	38	q3	q3	String Ditolak
2	108	q2	State Hampa	String Ditolak
3	3.000	q3	q3	String Ditolak
4	40.000	q2	State Hampa	String Ditolak
5	500.000	q2	q5	String Diterima
6	3.000.000	q3	q5	String Diterima
7	20.000.000	q2	State Hampa	String Ditolak
8	100.000.000	q3	State Hampa	String Ditolak
9	2.000.000.000	q3	q5	String Diterima
10	60.000.000.000	-	-	Aplikasi Berhenti

Pada Tabel 3 dan Tabel 4 dilakukan pengujian pada setiap angka desimal yang ada di dalam data uji dengan 10 kali pengujian terhadap Abstract Machine DFA dan 10 kali pengujian terhadap Abstract Machine NFA, Adapun keseluruhan pengujian dari data uji yang telah disediakan dari pengujian keseluruhan Abstract Machine baik itu DFA ataupun NFA sebanyak 40 kali pengujian dengan hasil persentase sebagai berikut :

$$\frac{Jumlah\ Data\ Pengujian\ Berhasil}{Jumlah\ Data\ Pengujian\ Keseluruhan}\ x\ 100\% = \frac{36}{40}\ x\ 100\% = \mathbf{90,00}\%$$

4. Conclusion

Dari apa yang telah dikaji sebelumnya, pengujian serta penelitian yang sudah dilaksanakan sebelumnya, dapat disimpulkan bahwa :

1. Fungsi dan tabel perubahan/transisi dapat mengklarifikasikan string yang di-input itu diterima atau ditolak dari mesin abstrak. Pada mesin abstrak NFA, apabila kita meng-input string tertentu yang menyebabkan kita tidak ada pilihan input yang sesuai, maka string tersebut hasilnya dinyatakan ditolak dikarenakan state hampa.
2. Hasil dari pengujian dengan aplikasi Abstract Machine dengan menggunakan Android Studio menghasilkan keputusan dapat diterima dan ditolak oleh mesin abstrak.

Attribution-NonCommercial-ShareAlike CC BY-NC-SA

3. Apabila suatu input-an string pada proses terakhir berhenti pada state akhir maka input-an diterima, namun apabila suatu input-an string pada proses terakhir berhenti tidak pada state akhir maka input-an ditolak.
4. Dapat diketahui bahwa dari Aplikasi Abstract Machine, mesin abstrak DFA dan NFA itu berbeda. Mesin abstrak DFA yaitu automata tidak dapat menimbulkan transisi ke lebih dari satu kondisi state disaat yang bersamaan, sedangkan untuk mesin abstrak NFA yaitu automata tidak dapat menimbulkan transisi apabila lebih dari satu kondisi state disaat bersama-sama.. Yang artinya mesin abstrak DFA untuk setiap state hanya dapat dilewati oleh satu proses saja, sedangkan untuk mesin abstrak NFA pada state q1 kita dapat memilih jalur state mana yang ingin kita lewati, apakah ingin melewati state q2 atau melewati state q3.
5. Hasil pengujian aplikasi Abstract Machine dengan menggunakan Android Studio yaitu pada aplikasi Abstract Machine dapat membaca data berupa angka hingga jangkauan maksimal dari nilai integer yaitu 2,147,483,647. Pada Tabel 3 dan Tabel 4 dilakukan pengujian pada setiap angka desimal yang ada di data uji dengan 10 kali pengujian terhadap Abstract Machine DFA dan 10 kali pengujian terhadapt Abstract Machine NFA. Adapun keseluruhan pengujian dari data uji yang telah disediakan yaitu sebanyak 40 kali pengujian dengan hasil 36 yang berhasil diujikan dari 40 pengujian data dengan tingkat akurasi sebesar 90,00%.

References

[1] Widyasari, "Telaah Teoritis Finite State Automata Dengan Pengujian Hasil Pada Mesin Otomata," *Sisfotenika*, vol. 1, no. 1, pp. 59–67, 2011, [Online]. Available: https://media.neliti.com/media/publications/.

[2] A. H. Pratomo, O. S. Simanjuntak, and D. C. N. Putra, "Bisindo Sign Language Transliteration Using Automata," *Compiler*, vol. 8, no. 1, p. 57, 2019, doi: 10.28989/compiler.v8i1.415.

[3] T. H. Wicaksono, F. D. Amrizal, and H. A. Mumtahana, "Pemodelan Vending Machine dengan Metode FSA (Finite State Automata)," *DoubleClick J. Comput. Inf. Technol.*, vol. 2, no. 2, pp. 66–69, 2019, [Online]. Available: http://e-journal.unipma.ac.id/index.php/doubleclick.

[4] V. Atina, S. Palgunadi, and W. Widiarto, "Program Transliterasi Antara Aksara Latin dan Aksara Jawa dengan Metode FSA," *J. Teknol. Inf. ITSmart*, vol. 1, no. 2, p. 60, 2016, doi: 10.20961/its.v1i2.592.

[5] D. Suhaedi, "EKSPRESI REGULAR PADA SUATU DETERMINISTIC FINITE STATE AUTOMATA," *J. Mat.*, vol. 6, no. 1, pp. 63–70, 2006.

[6] B. Richardson, K. Hendy, V. Andiyani, and W. Philips, "Penerapan Konsep Non-Deterministic Finite Automata (NFA) pada Aplikasi Simulasi Mesin Kopi Vending," *J. Inform. Univ. Pamulang*, vol. 4, no. 1, p. 1, 2019, doi: 10.32493/informatika.v4i1.2062.