

## IMPLEMENTASI FINITE STATE AUTOMATA PADA MESIN ABSTRAK DFA DAN NFA BERBASIS ANDROID

Dimas Dandy Aryarajendra Suprpto<sup>1</sup>, Fauziah<sup>2</sup>  
Program Studi Informatika, Universitas Nasional  
dimazdandy@gmail.com<sup>1</sup>, fauziah@civitas.unas.ac.id<sup>2</sup>

*Submitted March 9, 2020; Revised June 30, 2020; Accepted July 5, 2020*

### Abstrak

Mesin Abstrak FSA (*Finite State Automata*) merupakan mesin yang hanya dilihat dari logic dan proses serta dapat menggunakan berbagai bahasa pemrograman untuk membuatnya. Mesin Abstrak FSA bersifat menerima dan menolak suatu inputan. FSA dibagi menjadi dua yaitu, DFA (*Deterministic Finite Automata*) dan NFA (*Non-deterministic Finite Automata*). Pada paper ini membahas tentang penggunaan algoritma FSA untuk membuat mesin abstrak DFA dan NFA yang berbasis android. Dengan menggunakan aplikasi mesin abstrak FSA berbasis android user dapat menentukan hasil input proses automata diterima atau ditolak secara mudah dan akurat tanpa harus melakukan perhitungan manual.

**Kata kunci:** Mesin Abstrak, FSA, DFA, NFA, Android.

### Abstract

*FSA abstract machine (Finite State Automata) is a machine that is only seen from logic and processes and can use various programming languages to make it. The FSA Abstract Engine is accepting and rejecting an input. FSA is divided into two namely, DFA (Deterministic Finite Automata) and NFA (Non-deterministic Finite Automata). This paper discusses the use of the FSA algorithm to create DFA and NFA abstract machines based on Android. By using an android-based FSA abstract engine application the user can determine the input results of the automata process accepted or rejected easily and accurately without having to do manual calculations.*

**Key Words:** Abstract machine, FSA, DFA, NFA, Android.

### 1. PENDAHULUAN

*Finite State Automata* adalah mesin abstrak yang merupakan bagian dari teori bahasa dan automata. Teori ini sangat berguna untuk perkembangan teknologi baik software maupun hardware[1].

*Finite State Automata/otomata* berhingga state, FSA disebut juga sebagai suatu model matematika dari suatu sistem yang menerima input dan output diskrit. FSA merupakan mesin otomata dari bahas regular. Suatu FSA memiliki *state* yang banyaknya berhingga dan dapat berpindah-pindah dari suatu *state ke state* lain. Perubahan *state* ini dinyatakan oleh fungsi transisi. FSA tidak memiliki tempat penyimpanan, sehingga

kemampuan ‘mengingatnya’ terbatas, hanya bisa mengingat state yang terkini[7].

Pada teori bahasa dan automata, bahasa dapat dikategorikan sebagai penyampaian ide, dan *logic* manusia, Implementasi terhadap mesin abstrak merupakan ilmu dari [5].

Mesin abstrak FSA dibagi menjadi dua yaitu, DFA (*Deterministic Finite Automata*) dan NFA (*Non-deterministic Finite Automata*) dimana terdapat state yang menerima dan menolak suatu proses sesuai dengan input yang diberikan[2].

Perbedaan di antara keduanya adalah bahwa DFA menerima sebuah input

dimana state tujuan dari input tersebut adalah satu, sedangkan NFA dapat menuju beberapa state tujuan untuk input yang sama. Perbedaan lainnya adalah DFA tidak menerima input kosong (*empty*), tidak seperti NFA. FSA yang biasanya pertama kali dirancang adalah NFA, dan kemudian ditransformasi ke bentuk DFA sebagai bentuk yang lebih minimal dan efisien. Dalam proses transformasi, umumnya dilakukan beberapa tahapan khusus dan kompleks[8].

Implementasi mesin abstrak FSA yang dilakukan akan berbasis android sebagai salah satu teknologi yang telah banyak digunakan saat ini, dan penggunaannya yang mudah akan membantu sebagai alat pembelajaran[3].

Berdasarkan latar belakang diatas, maka akan dilakukan sebuah perancangan aplikasi yang merupakan implementasi dari teori bahasa dan automata berkaitan dengan mesin abstrak FSA. Penerapan mesin abstrak ini, diharapkan dapat membantu mahasiswa untuk memahami secara jelas tentang teori bahasa dan automata dengan adanya contoh mesin abstrak FSA[4].

## 2. METODE PENELITIAN

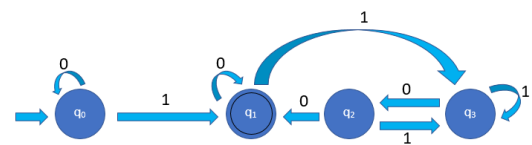
Aplikasi ini merupakan salah satu dari mesin abstrak FSA yaitu, NFA (*Non Deterministic Finite Automata*) dan DFA (*Deterministic Finite Automata*). Dimana mesin ini hanya bisa mengenali, menerima dan membangkitkan dari nilai *input* yang diberikan [1].

Mesin DFA itu sendiri merupakan mesin otomata yang *Finite State Number*. Dimana *finite* berarti terbatas dalam menentukan jumlah statenya. Sementara *deterministic* berarti tidak ambigu dalam menentukan *state*, jika mesin menerima sebuah inputan maka akan ada transisi ke state selanjutnya.

Mesin NFA berbeda dengan mesin DFA. Pada mesin NFA tiap statenya dapat

bertransisi ke lebih dari satu *state*, atau juga pada *state* tidak memiliki suatu nilai *input* atau yang di sebutkan dengan nilai hampa. Maka arti dari *non-deterministic* itu adalah artinya ambigu atau tidak tetap dalam menjalankan suatu nilai *input* yang di berikan.

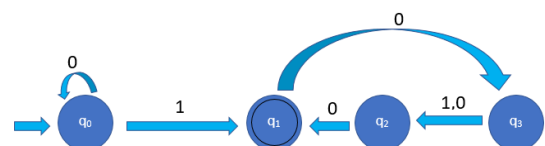
*Tuple* pada metode FSA yaitu  $M = (Q, \Sigma, \delta, S, F)$  yang berarti  $Q$  = himpunan *state*,  $\Sigma$  = himpunan *input*,  $\delta$  = fungsi transisi,  $S$  = *state* awal,  $F$  = *state* akhir[2].



Gambar 1. Diagram Transisi DFA

Pada gambar 1 merupakan diagram transisi DFA pada aplikasi ini, dengan *tuple* sebagai berikut :

$$\begin{aligned}
 Q &= \{q_0, q_1, q_2, q_3\} \\
 \Sigma &= \{0, 1\} \\
 \delta &= \delta(q_0, 0) = q_0, \quad \delta(q_0, 1) = q_1, \\
 &\delta(q_1, 0) = q_1, \quad \delta(q_1, 1) = q_3, \\
 &\delta(q_2, 0) = q_1, \quad \delta(q_2, 1) = q_3, \\
 &\delta(q_3, 0) = q_2, \quad \delta(q_3, 1) = q_3. \\
 S &= \{q_0\} \\
 F &= \{q_1\}
 \end{aligned}$$

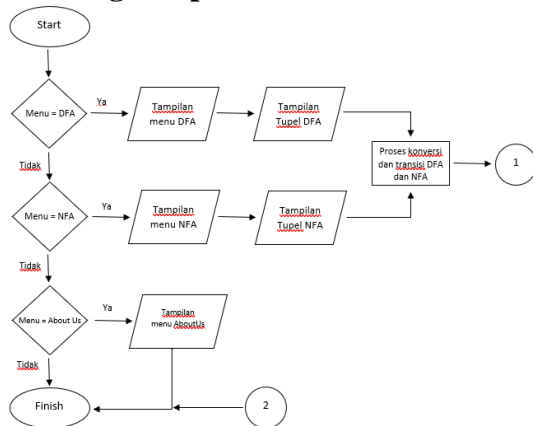


Gambar 2. Diagram Transisi NFA

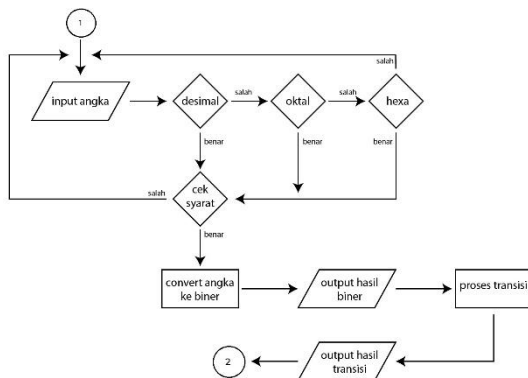
Pada gambar 2 merupakan diagram transisi NFA pada aplikasi ini, dengan *tuple* sebagai berikut :

$$\begin{aligned}
 Q &= \{q_0, q_1, q_2, q_3\} \\
 \Sigma &= \{0, 1\} \\
 \delta &= \delta(q_0, 0) = q_0, \quad \delta(q_0, 1) = q_1, \quad \delta(q_1, 0) = \\
 &q_3, \quad \delta(q_1, 1) = \emptyset, \quad \delta(q_2, 0) = q_1, \\
 &\delta(q_2, 1) = \emptyset, \quad \delta(q_3, 0) = q_2, \\
 &\delta(q_3, 1) = q_2. \\
 S &= \{q_0\} \\
 F &= \{q_1\}
 \end{aligned}$$

**Rancangan Aplikasi**



**Gambar 3. FlowChart Alur Aplikasi Android Mesin FSA**



**Gambar 4. Flowchart Proses Konversi Bilangan Biner ke Bilangan Decimal, Oktal, dan Hexadecimal, Beserta Proses Transisi pada Mesin FSA**

bilangan yang ada (dalam bentuk decimal, octal dan hexadecimal). Setelah itu program akan mengecek apakah nilai tersebut memenuhi syarat atau tidak, misalkan pada desimal angka maksimal yang ada pada aplikasi ini adalah 255 dan pada hexa dan oktal angka maksimalnya adalah 2 digit.

Jika syarat terpenuhi maka akan mengkonversi nilai tersebut ke dalam bilangan biner. selanjutnya program akan menghitung berapa jumlah digit pada bilangan biner tersebut. Program akan menambahkan angka 0 jika nilai biner tersebut kurang dari 8 digit untuk desimal dan hexa, sementara untuk oktal adalah 6 digit. Lalu hasilnya akan di tampilkan.

Ketika user menekan tombol hasil maka hal pertama yang program jalankan adalah membuat string untuk menampung nilai pada setiap digit nilai biner dan menghitung jumlah digit pada biner tersebut.



**Gambar 5. Penyimpanan Bilangan Biner pada Array**

Aplikasi ini dibuat dengan menggunakan *software* android studio. Bahasa pemrograman yang di gunakan pada aplikasi tersebut ialah java dan juga XML.

Bahasa pemrograman java disini digunakan untuk membuat *logic*, alur program atau perintah-perintah pada aplikasi ini, dan xml digunakan untuk mendesain aplikasi ini.

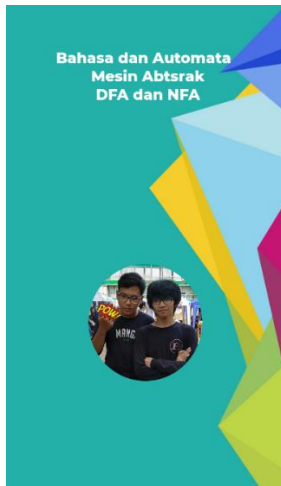
Aplikasi ini berjalan dengan awalan *user* akan menginput nilai angka dan memilih jenis bilangan yang ingin di gunakan dari desimal, oktal ataupun hexa. Langkah pertama yang di lakukan adalah membuat desain input sesuai pilihan untuk jenis

Selanjutnya program akan menyimpan nilai pada setiap digitnya pada array yang di telah dibuat. Dimana digit pertama akan disimpan pada array 0, digit kedua akan disimpan pada array 1 dan begitu seterusnya seperti yang ada pada gambar 5.

Langkah selanjutnya program akan membaca array dari 0 dan berpindah ke array selanjutnya hingga akhir untuk menentukan alur transisinya. Program akan mengeluarkan hasilnya sesuai dimana lokasi state itu berakhir. Jika program berhenti pada state akhir maka muncul pesan diterima, jika tidak maka akan muncul pesan ditolak.

### 3. HASIL DAN PEMBAHASAN

Gambar 6 sampai gambar 24 merupakan hasil atau aplikasi android mesin abstrak DFA dan NFA. Ketika membuka aplikasi, akan ada *splash screen* atau tampilan awal aplikasi.

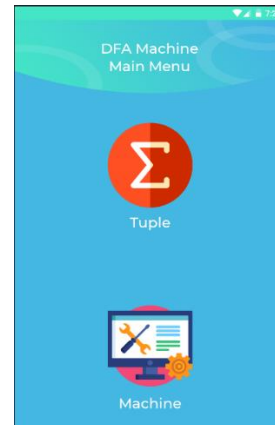


**Gambar 6. Tampilan Awal atau Splash Screen**



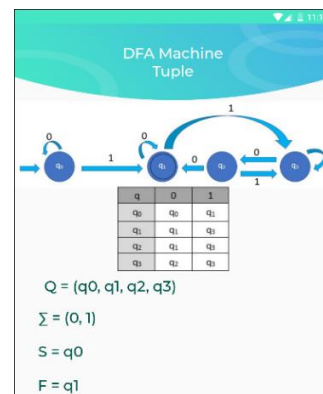
**Gambar 7. Desain Tampilan Menu**

Pada gambar 7 terdapat 3 menu yaitu DFA, NFA, dan About us. Dibawah bar menu terdapat penjelasan singkat apa itu Mesin Automata.



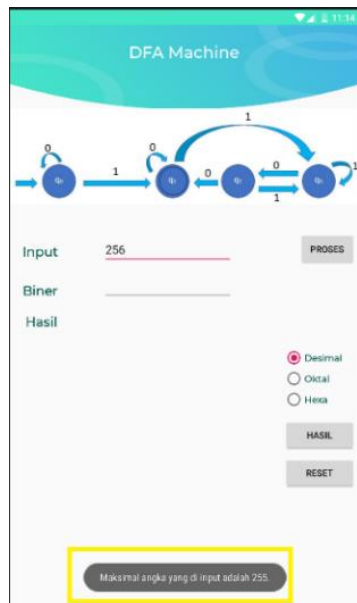
**Gambar 8. Desain Tampilan Menu DFA**

Pada gambar 8 terdapat 2 pilihan menu yaitu *Tuple* dan *Machine*. Dimana isi pada menu *Tuple* adalah *Tuple* dari Diagram DFA yang telah dibuat, dan *Machine* adalah mesin untuk menghitung DFA.



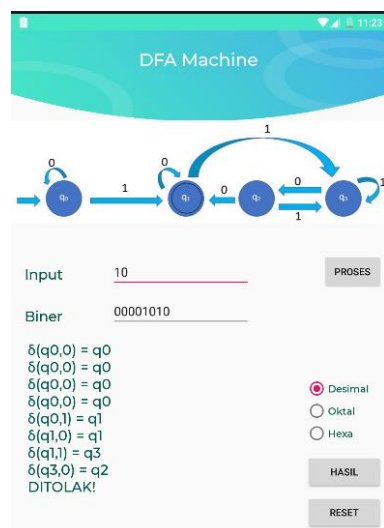
**Gambar 9. Desain Tampilan Menu DFA**

Pada gambar 9 terdapat diagram DFA, tabel transisi, serta *tuple-tuple* nya.



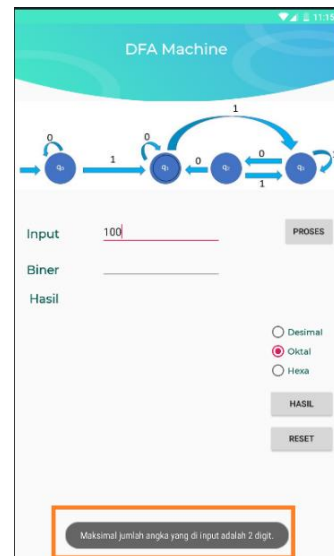
**Gambar 10. Batasan Angka pada Konversi Bilangan Biner ke Desimal pada Mesin DFA**

Pada gambar 10 Pada mesin abstrak DFA, ketika akan mengkonversi bilangan biner (pada kolom input) ke bilangan desimal, akan ada toast atau pesan singkat bahwa bilangan biner yang diinput tidak boleh lebih dari 255. 256 merupakan batasan input bilangan biner.



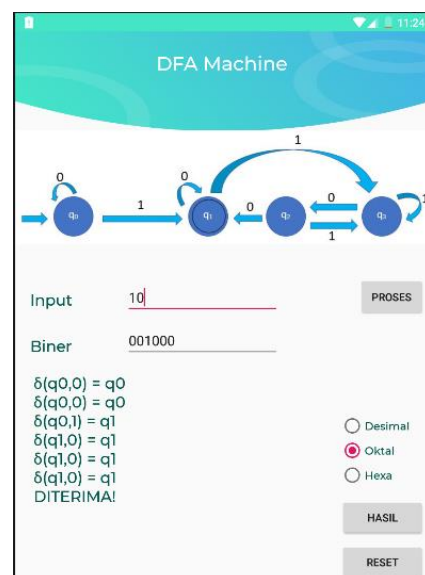
**Gambar 11. Konversi Input (Bilangan Biner) ke Desimal dan Menentukan Hasil dari Nilai Input Desimal pada Mesin DFA**

Pada gambar 11 nilai input yang di berikan adalah 10 desimal, nilai tersebut di konversi ke biner menjadi 00001010. Hasilnya nilai tersebut adalah ditolak.



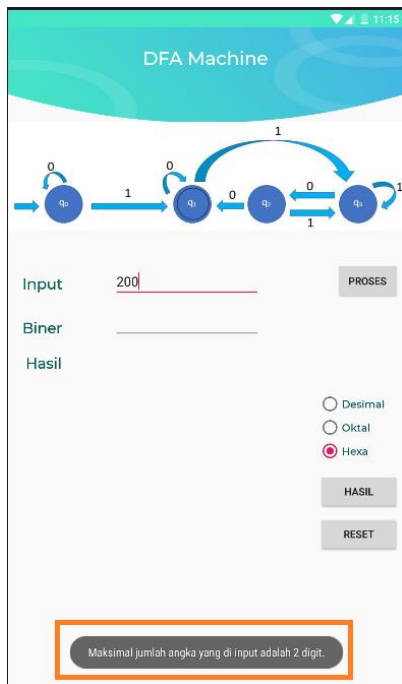
**Gambar 12. Batasan Angka pada Konversi Bilangan Biner ke Oktal pada Mesin DFA**

Pada gambar 12 ketika akan mengkonversi bilangan biner (pada kolom input) ke bilangan oktal, akan ada toast atau pesan singkat bahwa bilangan biner yang diinput tidak boleh lebih dari 2 digit angka. 100 terdiri dari 3 digit angka, sehingga tidak dapat melakukan proses konversi.



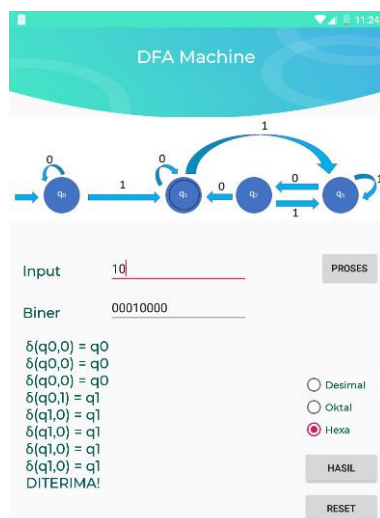
**Gambar 13. Menentukan Hasil dari Nilai Input Oktal pada Mesin DFA**

Pada gambar 13 nilai input yang di berikan adalah 10 oktal, nilai tersebut di konversi ke biner menjadi 001000. Hasilnya nilai tersebut adalah diterima.



**Gambar 14. Batasan Angka pada Konversi Bilangan Biner ke Hexadesimal pada Mesin DFA**

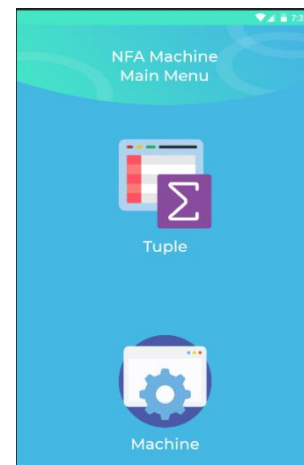
Pada gambar 14 ketika akan mengkonversi bilangan biner (pada kolom input) ke bilangan hexadesimal, akan ada toast atau pesan singkat bahwa bilangan biner yang diinput tidak boleh lebih dari 2 digit angka. 200 terdiri dari 3 digit angka, sehingga tidak dapat melakukan proses konversi.



**Gambar 15. Menentukan Hasil dari Nilai Input Hexa pada Mesin DFA**

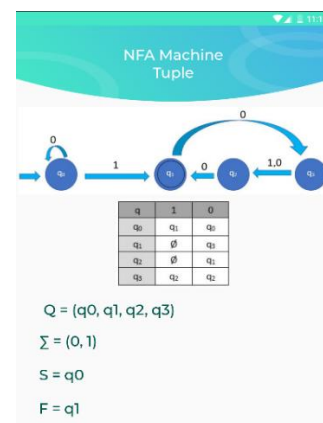
Pada gambar 15 nilai input yang diberikan adalah 10 hexa, nilai tersebut di

konversi ke biner menjadi 00010000. Hasilnya nilai tersebut adalah diterima.



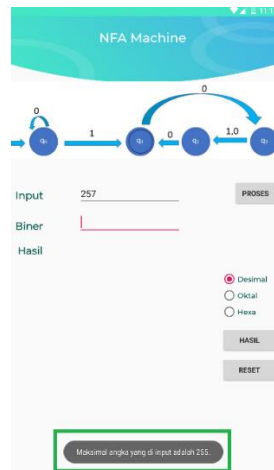
**Gambar 16. Tampilan Menu Mesin NFA**

Pada gambar 16 terdapat 2 pilihan menu yaitu *Tuple* dan *Machine*. Dimana isi pada menu *Tuple* adalah *Tuple* dari Diagram NFA yang telah dibuat, dan *Machine* adalah mesin untuk menghitung NFA.



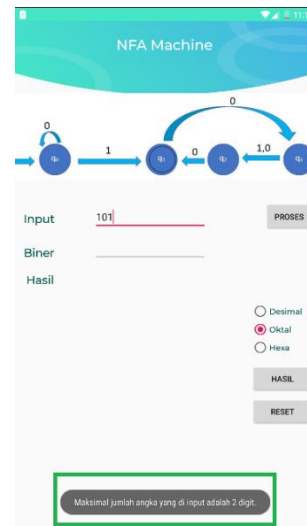
**Gambar 17. Desain Tampilan Menu NFA**  
Pada gambar 17 terdapat diagram NFA, tabel transisi, serta *tuple-tuple* nya.





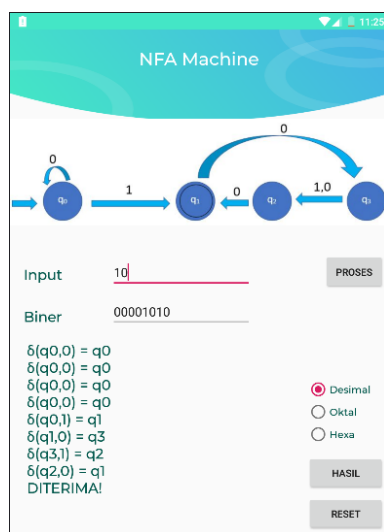
**Gambar 18. Batasan Konversi Bilangan Biner ke Desimal pada Mesin NFA**

Pada gambar 18 Pada mesin abstrak NFA, ketika akan mengkonversi bilangan biner (pada kolom input) ke bilangan desimal, akan ada toast atau pesan singkat bahwa bilangan biner yang diinput tidak boleh lebih dari 255. 257 merupakan batasan input bilangan biner.



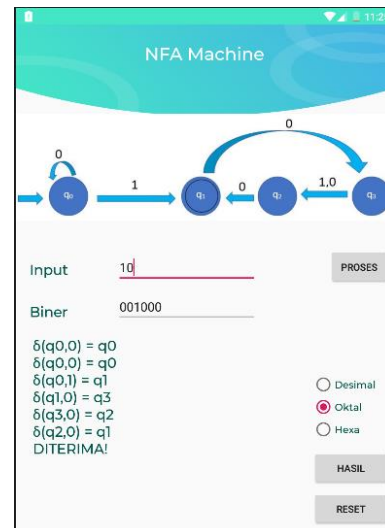
**Gambar 20. Batasan Konversi Bilangan Biner ke Oktal**

Pada gambar 20 Pada mesin abstrak NFA, ketika akan mengkonversi bilangan biner (pada kolom input) ke bilangan oktal, akan ada toast atau pesan singkat bahwa bilangan biner yang diinput tidak boleh lebih dari 2 digit. Angka 101 terdiri dari 3 digit, sehingga tidak dapat melakukan konversinya.



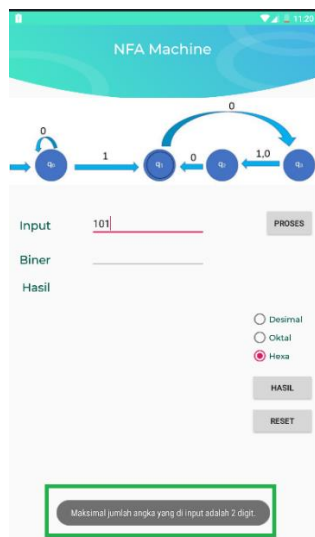
**Gambar 19. Menentukan Hasil dari Nilai Input Desimal pada Mesin DFA**

Pada gambar 19 nilai input yang di berikan adalah 10 desimal, nilai tersebut di konversi ke biner menjadi 00001010. Hasilnya nilai tersebut adalah diterima.



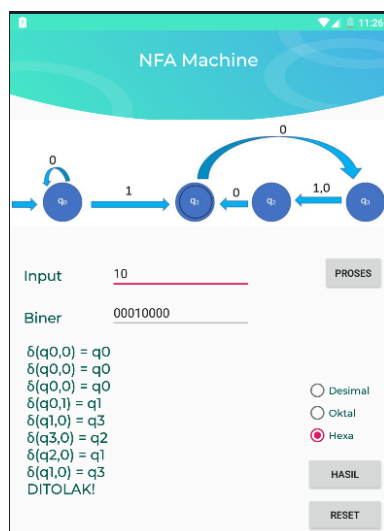
**Gambar 21. Menentukan Hasil dari Nilai Input Oktal pada Mesin NFA**

Pada gambar 21 nilai input yang di berikan adalah 10 oktal, nilai tersebut di konversi ke biner menjadi 001000. Hasilnya nilai tersebut adalah diterima.



**Gambar 22. Batasan Konversi Bilangan Biner ke Hexa**

Pada gambar 22 Pada mesin abstrak NFA, ketika akan mengkonversi bilangan biner (pada kolom *input*) ke bilangan hexadesimal, akan ada toast atau pesan singkat bahwa bilangan biner yang diinput tidak boleh lebih dari 2 digit. Angka 101 terdiri dari 3 digit, sehingga tidak dapat melakukan konversinya.



**Gambar 23. Menentukan Hasil dari Nilai Input Hexa pada Mesin NFA**

Pada gambar 23 nilai input yang di berikan adalah 10 hexa, nilai tersebut di konversi ke biner menjadi 00010000. Hasilnya nilai tersebut adalah ditolak.



**Gambar 24. Tampilan Pada Menu About**

Pada gambar 24 berisikan tentang data diri pembuat aplikasi.

#### 4. SIMPULAN

1. Mesin Abstrak FSA merupakan mesin yang hanya dilihat dari logic dan proses serta dapat menggunakan berbagai bahasa pemrograman untuk membuatnya.
2. Mesin abstrak FSA dibagi menjadi dua yaitu, DFA (*Deterministic Finite Automata*) dan NFA (*Non-deterministic Finite Automata*) dimana terdapat state yang menerima dan menolak suatu proses.
3. Mesin DFA merupakan *deterministic* berarti tidak ambigu dalam menentukan state, jika mesin menerima sebuah inputan maka akan ada transisi ke state selanjutnya.
4. Mesin NFA merupakan ambigu atau tidak tetap dalam menjalankan suatu nilai input yang di berikan. Pada mesin ini tiap statenya dapat bertransisi ke lebih dari satu state, atau juga pada state tidak memiliki suatu nilai input atau yang di sebutkan dengan nilai hampa.
5. Tuple pada metode FSA yaitu  $M = (Q, \Sigma, \delta, S, F)$  yang berarti  $Q$  = himpunan state,  $\Sigma$  = himpunan input,  $\delta$  = fungsi transisi,  $S$  = state awal,  $F$  = state akhir.



6. Aplikasi ini merupakan contoh mesin NFA dan DFA dalam menentukan suatu string jika di inputkan akan di terima atau ditolak.
7. Aplikasi ini menggunakan 8 digit bilangan biner.
8. Saat menginput nilai desimal, angka maksimalnya adalah 255.
9. Saat menginput nilai oktal dan hexa, jumlah digitnya ialah 2 digit.

#### DAFTAR PUSTAKA

- [1].R. A. Ma'arif and F. Fauziah, "Implementasi Finite State Automata (FSA) dalam Proses Pengisian Kartu Rencana Studi," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 3, no. 3, pp. 255–260, 2018.
- [2].Saputra, Tri Ichsan, dkk, "Simulasi Vending Machine Dengan Mengimplementasikan Finite State Automata," vol. 3, no. 3, 2018.
- [3].M. Y. F. Zaelani and Lelah, "Implementasi Finite State Automata Pada Aplikasi Pembelajaran Aksara Sunda," vol. 8, no. 2, 2018.
- [4].Leimena, Yessy Willy., Pakereng, Magdalena A. Ineke. (2017). Penerapan Finite Automata Pada Pergantian Desktop Background. Salatiga: Universitaas Kristen Satya Wacana.
- [5].Mauboy, Raymond Elias. (2017). Penerapan *Finite State Automata* Pada Proses Peminjaman Buku di Perpustakaan Universitas Kristen Satya Wacana. Salatiga: Universitaas Kristen Satya Wacan
- [6].Suparyanto, "Simulator Pengenal String Yang Diterima Sebuah *Deterministic Finite Automata (DFA)*," pp. 377–381, 2017.
- [7].Widyasari, "Telaah Teoritis Finite State Automata Dengan Pengujian Hasil Pada Mesin Otomata," *Sisfotenika*, vol. 1, no. 1, pp. 59–67, 2011.
- [8].Sinurat, S. Simulasi Transformasi Regular Expression Terhadap Finite State Automata. *Pelita Informatika: Informasi dan Informatika*, vol. 4, no. 1, 2013.