

**MAKALAH**  
**TEORI BAHASA DAN OTOMATA**

**Dosen Pengampu :**  
**Ihsan Fathoni Amri, M.Stat**



**Disusun oleh :**

**Fanni Tyasari**

**C2C021054**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MUHAMMADIYAH SEMARANG**  
**TAHUN 2022/2023**

## **KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Allah SWT, yang telah memberikan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan makalah ini dengan baik. Kami ingin mengungkapkan rasa terima kasih yang sangat besar kepada pihak yang memberikan kami tugas ini.

Makalah ini disusun oleh Fanni Tyasari (C2C021054), Makalah ini bertujuan untuk menjelajahi bagaimana teori bahasa dan otomata dapat diterapkan untuk mendapatkan pemahaman yang lebih mendalam tentang struktur bahasa, mengenali pola atau pola bahasa dalam data, dan mengembangkan model sistem komputasi dan komunikasi yang efisien.

Saya menyadari bahwa makalah ini masih jauh dari kesempurnaan, oleh karena itu, kritik dan saran yang membangun sangat kami harapkan demi perbaikan makalah ini. Kami mengucapkan terima kasih kepada dosen yang telah memberikan arahan dan bimbingan dalam pembuatan makalah ini, semoga makalah ini bermanfaat dan dapat menambah wawasan kita semua mengenai bidang Teori Bahasa dan Otomata dalam ilmu komputer.

Wassalamualaikum Wr. Wb.

Semarang, 11 Mei 2023

Tim Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>BAB I PENDAHULUAN.....</b>	<b>4</b>
1.1. Latar Belakang .....	4
1.2. Rumusan Masalah .....	5
1.3. Tujuan.....	5
1.4. Manfaat.....	5
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>6</b>
2.1. Non-Deterministic Finite Automata (NFA) .....	6
2.1.1. Definisi Non-Deterministic Finite Automata (NFA) .....	6
2.1.2. Transisi Lambda.....	7
2.2. Deterministic Finite Automata (DFA) .....	8
2.2.1. Definisi Deterministic Finite Automata (DFA) .....	8
2.2.2. Bahasa Reguler (Reguler Language) .....	9
2.3. Epsilon Move .....	10
2.3.1. Definisi Epsilon Move .....	10
2.3.2. Mengubah Epsilon Move menjadi tanpa Epsilon Move. ....	11
<b>BAB III PEMBAHASAN .....</b>	<b>15</b>
3.1. Penerapan Non – Deterministic Finite Automata (NFA) .....	15
3.1.1. Mesin Aplikasi Wilkerstat.....	15
3.2. Penerapan Deterministic Finite Automata (DFA).....	18
3.2.1. Mesin Aplikasi Wikerstat.....	18
3.3. Penerapan Epsilon Move ke tanpa Epsilon Move.....	20
3.3.1. Mesin Aplikasi Wikerstat.....	20
<b>PENUTUP.....</b>	<b>31</b>
4.1. Kesimpulan.....	31
<b>DAFTAR PUSTAKA .....</b>	<b>32</b>

## **BAB I PENDAHULUAN**

### **1.1. Latar Belakang**

Teori Bahasa dan Otomata adalah cabang penting dalam ilmu komputer yang mempelajari formalisme matematis untuk menggambarkan, mengenali, dan menghasilkan bahasa-bahasa yang dihasilkan oleh mesin komputer. Otomata adalah model matematika yang digunakan untuk menggambarkan sistem yang berubah dari satu keadaan ke keadaan lainnya berdasarkan serangkaian aturan.

Teori otomata mulai dikenal pada abad ke-20, Salah satu kontributor utama dalam pengembangan teori ini adalah Noam Chomsky, seorang ahli linguistik dan filsuf yang dikenal dengan teori tata bahasa dan teori bahasa generatif. Selain itu, sejumlah peneliti lainnya seperti Alan Turing, Stephen Cole Kleene, dan John Backus juga memberikan kontribusi penting dalam memahami dan mengembangkan teori bahasa dan otomata..

Badan Pusat Statistik (BPS) pada tanggal 15 Okt - 14 Nov 2022 akan melaksanakan Pendataan Awal Registrasi Sosial Ekonomi (Regsosek) di seluruh provinsi di Indonesia. Pendataan Regsosek adalah pengumpulan data seluruh penduduk yang terdiri atas profil, kondisi sosial, ekonomi, dan tingkat kesejahteraan. Data Regsosek dapat dimanfaatkan untuk meningkatkan kualitas berbagai layanan pemerintah seperti pendidikan, bantuan sosial, kesehatan, hingga administrasi kependudukan.

Dalam kegiatan Regsosek ini menggunakan aplikasi yang dibangun untuk melakukan tagging keluarga pada pada kegiatan registrasi sosial dan ekonomi yaitu wilkerstat. Wilkerstat adalah wilayah kerja statistik untuk kegiatan sensus dan survei yang diselenggarakan oleh BPS. Pesatnya pembangunan, tentunya menyebabkan banyak perubahan muatan pada wilkerstat . Pemutakhiran bentuk dan posisi peta digital Wilkerstat kali ini dipersiapkan untuk mendukung pendataan yang diselenggarakan BPS.

## **1.2. Rumusan Masalah**

1. Apa definisi dan perbedaan antara NFA, DFA, dan Epsilon Move dalam teori bahasa dan otomata?
2. Bagaimana penerapan NFA, DFA, dan Epsilon Move pada Mesin Aplikasi wilkerstat dan Berikan contoh-contoh penerapannya dalam kinerja Mesin?
3. Bagaimana penggunaan graph dalam menggambarkan struktur dan alur Mesin Aplikasi wilkerstat berbasis NFA, DFA, dan Epsilon Move dan Berikan langkah-langkah untuk membuat representasi grafis dari otomata tersebut pada NFA, DFA, dan Epsilon Move?

## **1.3. Tujuan**

1. Menjelaskan definisi dan perbedaan antara NFA (Nondeterministic Finite Automaton), DFA (Deterministic Finite Automaton), dan Epsilon Move dalam teori bahasa dan otomata.
2. Menganalisis penerapan NFA, DFA, dan Epsilon Move pada Mesin Aplikasi wilkerstat (contoh aplikasi atau sistem yang menggunakan otomata) serta memberikan contoh-contoh konkret mengenai bagaimana ketiga jenis otomata ini digunakan dalam kinerja Mesin tersebut.
3. Menjelaskan penggunaan graph (grafik) dalam menggambarkan struktur dan alur Mesin Aplikasi wilkerstat berbasis NFA, DFA, dan Epsilon Move serta memberikan langkah-langkah yang jelas untuk membuat representasi grafis dari otomata tersebut pada NFA, DFA, dan Epsilon Move.

## **1.4. Manfaat**

1. Menyediakan pemahaman yang mendalam tentang NFA, DFA, dan Epsilon Move dalam teori bahasa dan otomata. Makalah ini akan menjelaskan definisi masing-masing konsep dan perbedaan antara mereka. Hal ini akan membantu pembaca untuk memahami dasar-dasar teori ini dan membedakan karakteristik kunci dari setiap jenis otomata.
2. Memberikan wawasan tentang penerapan NFA, DFA, dan Epsilon Move dalam Mesin Aplikasi wilkerstat. Makalah ini akan menjelaskan bagaimana konsep-konsep ini diterapkan dalam praktek, terutama dalam konteks mesin aplikasi wilkerstat.
3. Menggambarkan penggunaan graph dalam menggambarkan struktur dan alur Mesin Aplikasi wilkerstat berbasis NFA, DFA, dan Epsilon Move. Makalah ini akan menjelaskan bagaimana graf digunakan sebagai representasi visual dari otomata ini. Langkah-langkah untuk membuat representasi grafis dari otomata NFA, DFA, dan Epsilon Move akan diuraikan secara terperinci

## **BAB II TINJAUAN PUSTAKA**

### **2.1. Non-Deterministic Finite Automata (NFA)**

#### **A. Algoritma NFA**

1. Tentukan himpunan keadaan (states)  $Q$ .
2. Tentukan himpunan simbol input (input symbols)  $\Sigma$ .
3. Tentukan fungsi transisi (transition function)  $\delta$ .
4. Tentukan keadaan awal (start state)  $q_0$ .
5. Tentukan himpunan keadaan akhir (final states)  $F$ .
6. Baca masukan (input) dari pengguna.
7. Mulai keadaan awal ( $q_0$ ).
8. Untuk setiap simbol dalam masukan, himpunan baru untuk menyimpan keadaan berikutnya (next states).
9. Tambahkan dan Perbarui keadaan berikutnya ke dalam himpunan next states.
10. Setelah semua simbol dalam masukan telah diproses, periksa apakah ada keadaan saat ini yang merupakan keadaan akhir (final state)
11. Tampilkan hasil akhir (masukan diterima atau ditolak) kepada pengguna.
12. Selesai

#### **2.1.1. Definisi Non-Deterministic Finite Automata (NFA)**

Non-deterministic Finite Automata, selanjutnya disebut sebagai NFA. Pada NFA, dari suatu state bisa terdapat 0 atau 1 atau lebih busur keluar (transisi) berlabel simbol input yang sama. Suatu string diterima oleh NFA bila terdapat suatu urutan transisi sehubungan dengan input string tersebut dari state awal menuju state akhir. Untuk NFA, maka harus mencoba semua kemungkinan yang ada sampai terdapat satu yang mencapai state akhir.

NFA hampir sama seperti DFA, namun yang membedakan adalah pada beberapa hal berikut:

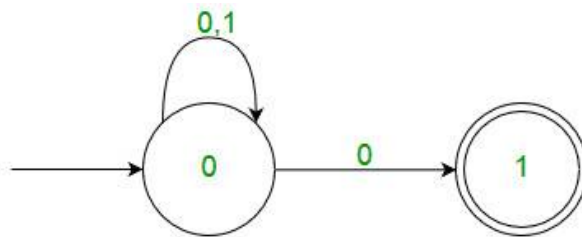
- Diperbolehkan terjadi perpindahan null (or  $\epsilon$ ), yang berarti NFA dapat berpindah ke state berikutnya tanpa membaca simbol input.
- NFA dapat mengirimkan beberapa state untuk input tertentu.

Pada dasarnya beberapa hal di atas tidak membuat NFA lebih unggul dari DFA. Jika kita membandingkan keduanya dalam hal kekuatan, keduanya setara.

Karena fitur tambahan di atas, NFA memiliki fungsi transisi yang berbeda, selebihnya sama dengan DFA. Fungsi transisi pada NFA dapat didefinisikan sebagai berikut:

$$\delta: Q \times (\Sigma \cup \epsilon) \rightarrow 2^Q.$$

Berikut adalah contoh dari NFA:



Satu hal penting yang perlu diperhatikan pada NFA adalah jika ada jalur untuk string input yang mengarah ke state akhir, maka string input diterima. Misalnya, di NFA di atas, ada beberapa jalur untuk string input "00". Karena salah satu jalur mengarah ke state akhir, "00" diterima oleh NFA.

### 2.1.2. Transisi Lambda

Transisi lambda, juga dikenal sebagai transisi epsilon ( $\epsilon$ ), adalah jenis transisi dalam otomata yang memungkinkan mesin untuk beralih ke keadaan baru tanpa membaca simbol masukan apa pun. Dalam hal ini, mesin dapat "melompat" langsung ke keadaan baru tanpa adanya input eksternal. Berikut adalah contoh transisi lambda ( $\lambda$ ).

State Awal	Input	State Tujuan
q0	$\lambda$	q1
q1	$\lambda$	q2
q2	$\lambda$	q3

Dalam contoh di atas, terdapat tiga transisi lambda yang terjadi dari state awal ke state tujuan. Transisi pertama terjadi dari state awal q0 dengan input  $\lambda$  (lambda) menuju state tujuan q1. Transisi kedua terjadi dari state awal q1 dengan input  $\lambda$  menuju state tujuan q2. Terakhir, transisi ketiga terjadi dari state awal q2 dengan input  $\lambda$  menuju state tujuan q3.

Perlu diperhatikan bahwa transisi lambda tidak memerlukan simbol masukan yang spesifik dan dapat digunakan untuk memindahkan mesin antara keadaan-keadaan tanpa membaca simbol masukan apa pun.

## **2.2.Deterministic Finite Automata (DFA)**

### **A. Algoritma DFA**

1. Tentukan himpunan keadaan ( $Q$ ).
2. Tentukan himpunan simbol masukan ( $\Sigma$ ).
3. Tentukan fungsi transisi ( $\delta$ ) yang menghubungkan keadaan saat ini dengan keadaan berikutnya berdasarkan simbol masukan.
4. Tentukan keadaan awal ( $q_0$ ).
5. Tentukan himpunan keadaan akhir ( $F$ ).
6. Set keadaan saat ini menjadi keadaan awal ( $q_0$ ).
7. Baca simbol masukan berikutnya.
8. Gunakan fungsi transisi ( $\delta$ ) untuk menentukan keadaan berikutnya berdasarkan keadaan saat ini dan simbol masukan yang dibaca.
9. Set keadaan saat ini menjadi keadaan berikutnya yang ditentukan oleh fungsi transisi ( $\delta$ ).
10. Cek keadaan akhir, Jika keadaan saat ini adalah keadaan akhir ( $F$ ), maka kata diterima oleh DFA. Jika tidak, kata ditolak oleh DFA.
11. Selesai

### **2.2.1. Definisi Deterministic Finite Automata (DFA)**

Deterministic Finite Automata/otomata berhingga deterministik, selanjutnya disebut sebagai DFA. Pada DFA, dari suatu state ada tepat satu state berikutnya untuk setiap simbol masukan yang diterima. Suatu string  $x$  dinyatakan diterima bila  $\delta(S, x)$  berada pada state akhir/final state.

Dalam Deterministic Finite Automata (DFA), untuk karakter input tertentu, mesin hanya dapat menuju satu state dan fungsi transisi dipakai pada setiap state untuk setiap simbol input. Selain itu, pada DFA, perpindahan state null (atau  $\epsilon$ ) tidak diperbolehkan, artinya, DFA tidak dapat mengubah state tanpa karakter input sama sekali.

Deterministic Finite Automata (DFA) terdiri atas 5 tuple, yakni:

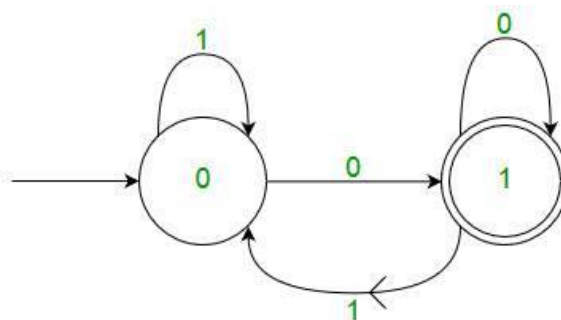
$$\{Q, \Sigma, q, F, \delta\}$$

- $Q$  : himpunan semua state



- $\Sigma$  : himpunan simbol input (simbol yang digunakan oleh mesin untuk mengambil nilai input)
- $q$  : initial state (state atau keadaan awal dari mesin)
- $F$  : himpunan state akhir
- $\delta$  : fungsi transisi, yang didefinisikan  $\delta : Q \times \Sigma \rightarrow Q$ .

Sebagai contoh berikut adalah DFA  $\Sigma = \{0, 1\}$  yang menerima semua string yang diakhiri dengan 0.



Satu hal penting yang perlu diperhatikan pada jenis DFA adalah terdapat banyak kemungkinan pada sebuah pola. Namun secara umum, DFA dengan jumlah state minimum cenderung lebih baik.

### 2.2.2. Bahasa Reguler (Reguler Language)

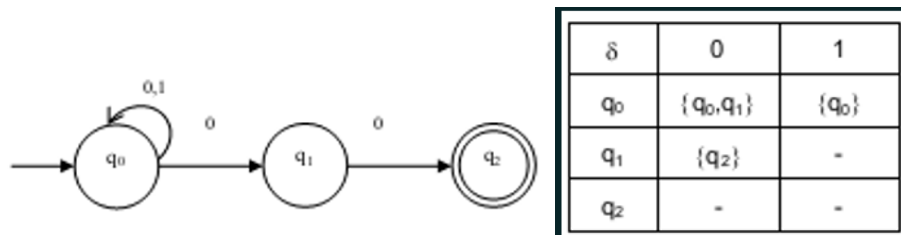
Dalam ilmu komputer teoretis dan teori bahasa formal, regular language adalah bahasa formal yang dapat didefinisikan dengan regular expression.

Dengan kata lain, Regular Language dapat didefinisikan sebagai bahasa yang dikenali oleh finite automaton. Dalam membuat mesin Deterministic Finite Automata yang menerima bahasa yang berupa semua string yang berakhiran dengan '00'.

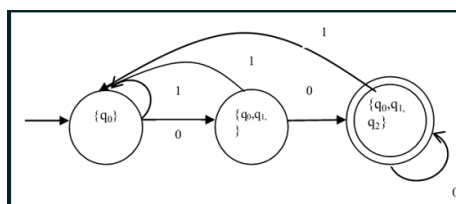
Diketahui  $\Sigma = (0,1)$

Pertama kita buat ekspresi reguler:  $(0+1)^*00$

Dari ekspresi reguler tersebut kita lebih mudah membuat Non-deterministic Finite Automata lebih dahulu, daripada langsung Deterministic Finite Automata. Mesin NFA-nya dapat kita bisa lihat pada gambar berikut: NFA yang menerima string yang berakhiran dengan '00'.



Akhirnya Deterministic Finite Automata yang ekivalen dengan Non-deterministic Finite Automata tersebut. Bisa di cek dengan untai yang harus diterima oleh mesin itu, seperti : 00, 100, 000, 0100, 0000, 1100.



## 2.3.Epsilon Move

### A. Algoritma Epsilon Move

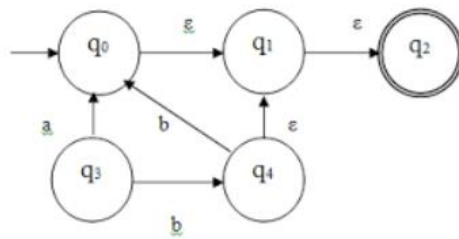
1. Mulai dengan sebuah keadaan awal.
2. Periksa apakah keadaan saat ini memiliki transisi epsilon ( $\epsilon$ ) yang terhubung ke keadaan lain.
3. Jika ada transisi epsilon, tambahkan keadaan tersebut ke dalam himpunan keadaan yang dijangkau melalui transisi epsilon.
4. Periksa apakah ada transisi epsilon lagi dari keadaan-keadaan yang baru ditambahkan.
5. Jika ada, tambahkan keadaan-keadaan tersebut juga ke dalam himpunan keadaan yang dijangkau melalui transisi epsilon.
6. Lanjutkan proses ini hingga tidak ada lagi transisi epsilon yang dapat ditemukan.
7. Setelah semua keadaan yang dijangkau melalui transisi epsilon telah ditambahkan ke dalam himpunan, kembalikan himpunan keadaan tersebut sebagai hasil epsilon move.

### 2.3.1. Definisi Epsilon Move

Epsilon move, juga dikenal sebagai transisi lambda ( $\lambda$ ) atau transisi kosong, adalah jenis transisi dalam otomata yang memungkinkan mesin berpindah antara keadaan tanpa membaca simbol masukan apa pun. Epsilon move memungkinkan mesin melakukan transisi tambahan yang tidak bergantung pada simbol masukan.

Penggunaan epsilon move memperluas kemampuan otomata untuk mengenali bahasa yang lebih kompleks dan memodelkan berbagai tipe transisi. Ini memungkinkan otomata untuk menjadi lebih ekspresif dalam definisi bahasa yang diterima dan memberikan fleksibilitas dalam pemrosesan string atau teks.

Contoh:



- Dari q0 tanpa membaca input dapat berpindah ke q1
- Dari q1 tanpa membaca input dapat berpindah ke q2
- Dari q4 tanpa membaca input dapat berpindah ke q1
- $\epsilon$ -move adalah suatu transisi antara dua status tanpa adanya input. Contoh gambar : Transisi anatara status q1 ke q3
- $\epsilon$ -Closure adalah himpunan state-state yang dapat dicapai dari suatu state tanpa membaca input. Misalkan saja  $\epsilon$ -closure(q0) = himpunan state-state yang dapat dicapai dari state q0 tanpa membaca input. Maka dengan melihat contoh diatas  $\epsilon$ -closure(q0) = { q0, q1, q2 }, artinya dari state q0 tanpa membaca input dapat mencapai state q0, q1, q2.

$$\epsilon\text{-closure}(q0) = \{ q0, q1, q2 \},$$

$$\epsilon\text{-closure}(q1) = \{ q1, q2 \},$$

$$\epsilon\text{-closure}(q2) = \{ q2 \},$$

$$\epsilon\text{-closure}(q3) = \{ q3 \},$$

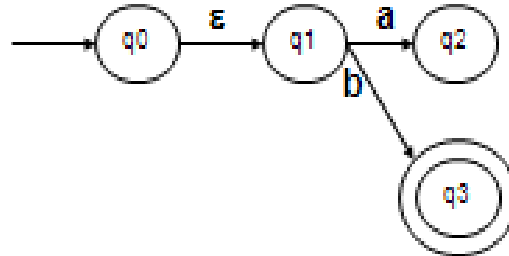
$$\epsilon\text{-closure}(q4) = \{ q1, q2, q4 \}$$

### 2.3.2. Mengubah Epsilon Move menjadi tanpa Epsilon Move.

1. Buat table transisi NFA dengan  $\epsilon$  - Move
2. Tentukan  $\epsilon$  - Closure untuk setiap state
3. Mencari setiap fungsi hasil perubahan dari NFA  $\epsilon$  - Move ke NFA tanpa  $\epsilon$  - Move
4. Tabel Transisi dan Diagram Transisi NFA tanpa  $\epsilon$  - Move yang ekuivalensi dengan NFA  $\epsilon$  - Move  $\epsilon$

5. State akhir untuk NFA tanpa  $\epsilon$  - Move

Contoh



1. Tabel Transisi

$\delta$	a	b
q0	$\phi$	$\phi$
q1	q2	q3
q2	$\phi$	$\phi$
q3	$\phi$	$\phi$

2.  $\epsilon$  - Closure untuk setiap state

- $\epsilon$ -closure(q0) = { q0, q1, q2 },
- $\epsilon$ -closure(q1) = { q1, q2 },
- $\epsilon$ -closure(q2) = { q2 },
- $\epsilon$ -closure(q3) = { q3 },
- $\epsilon$ -closure(q4) = { q1, q2, q4 }

3. Fungsi hasil perubahan dari NFA  $\epsilon$  - Move ke NFA tanpa  $\epsilon$  - Move

Carilah setiap fungsi hasil perubahan dari NFA  $\epsilon$  - Move ke NFA tanpa  $\epsilon$  - Move dengan rumus :  $\delta' ( \text{state}, \text{input} ) = \epsilon - \text{closure} ( \delta ( \epsilon - \text{closure} ( \text{state} ), \text{input} ) )$

$$\delta' ( \text{state}, \text{input} ) = \epsilon - \text{closure} ( \delta ( \epsilon - \text{closure} ( \text{state} ), \text{input} ) )$$

$\delta' ( q0, a )$ $= \epsilon - \text{closure} ( \delta ( \epsilon - \text{closure} ( q0 ), a ) )$ $= \epsilon - \text{closure} ( \delta ( \{ q0, q1 \}, a ) )$ $= \epsilon - \text{closure} ( q2 )$ $= \{ q2 \}$	$\delta' ( q0, b )$ $= \epsilon - \text{closure} ( \delta ( \epsilon - \text{closure} ( q0 ), b ) ) = \epsilon - \text{closure} ( \delta ( \{ q0, q1 \}, b ) )$ $= \epsilon - \text{closure} ( q3 )$ $= \{ q3 \}$
$\delta' ( q1, a )$ $= \epsilon - \text{closure} ( \delta ( \epsilon - \text{closure} ( q1 ), a ) )$	$\delta' ( q1, b )$ $= \epsilon - \text{closure} ( \delta ( \epsilon - \text{closure} ( q1 ), b ) )$

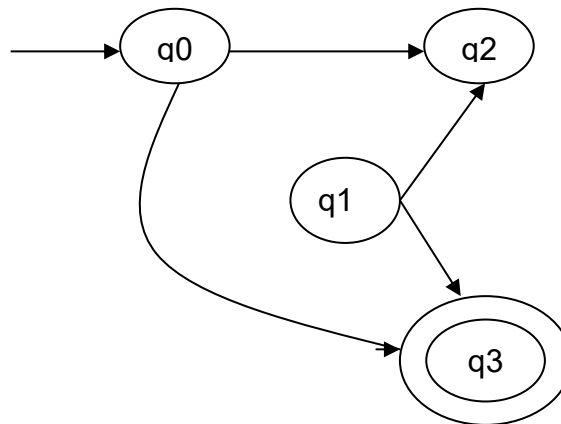
$= \varepsilon - \text{closure} ( \delta ( \{ q1 \}, a ) )$ $= \varepsilon - \text{closure} ( q2 )$ $= \{ q2 \}$	$= \varepsilon - \text{closure} ( \delta ( \{ q1 \}, b ) )$ $= \varepsilon - \text{closure} ( q3 )$ $= \{ q3 \}$
<b><math>\delta' ( q2, a )</math></b> $= \varepsilon - \text{closure} ( \delta ( \varepsilon - \text{closure} ( q2 ), a ) )$ $= \varepsilon - \text{closure} ( \delta ( \{ q2 \}, a ) )$ $= \varepsilon - \text{closure} ( \phi )$ $= \phi$	<b><math>\delta' ( q2, b )</math></b> $= \varepsilon - \text{closure} ( \delta ( \varepsilon - \text{closure} ( q2 ), b ) )$ $= \varepsilon - \text{closure} ( \delta ( \{ q2 \}, b ) )$ $= \varepsilon - \text{closure} ( \phi )$ $= \phi$
<b><math>\delta' ( q3, a )</math></b> $= \varepsilon - \text{closure} ( \delta ( \varepsilon - \text{closure} ( q3 ), a ) )$ $= \varepsilon - \text{closure} ( \delta ( \{ q3, a \} ) )$ $= \varepsilon - \text{closure} ( \phi )$ $= \phi$	<b><math>\delta' ( q3, b )</math></b> $= \varepsilon - \text{closure} ( \delta ( \varepsilon - \text{closure} ( q3 ), b ) )$ $= \varepsilon - \text{closure} ( \delta ( \{ q3 \}, b ) )$ $= \varepsilon - \text{closure} ( \phi )$ $= \phi$

4. Berdasarkan hasil no. 3 buat tabel transisi & diagram transisi NFA tanpa  $\varepsilon$  - Move yang ekuivalensi dengan NFA  $\varepsilon$  - Move  $\varepsilon$

$\delta'$	a	b
q0	q2	q3
q1	q2	q3
q2	$\phi$	$\phi$
q3	$\phi$	$\phi$

5. State akhir untuk NFA tanpa  $\epsilon$  - Move  
state akhir untuk NFA tanpa  $\epsilon$  - Move tersebut, yaitu state – state akhir semua  
& tambah dengan state – state yang  $\epsilon$  - closurenya menuju ke salah satu dari  
state akhir semula.

NFA tanpa  $\epsilon$  - Move



## BAB III PEMBAHASAN

### 3.1. Penerapan Non – Deterministic Finite Automata (NFA)

#### 3.1.1. Mesin Aplikasi Wilkerstat

A. Algoritma NFA:

1. Buka Aplikasi Wilkerstat yang sudah terinstall di smartphone Anda.
2. Tekan tombol “Plus” untuk menambahkan project.
3. Mulai melakukan sensus dengan memasukkan data yang diperlukan, seperti nama responden, nomor rumah, dan jenis kelamin.
4. Jika lokasi rumah yang dituju telah tercapai, centang tanda di depan rumah tersebut.
5. Jika responden termasuk keluarga miskin, ambil foto rumahnya dan jika tidak, tekan tombol “Location”.
6. Masuk ke menu Perekaman Lokasi untuk memastikan akurasi lokasi yang dicatat (harus sekitar 20 untuk keluarga miskin dan kaya).
7. Setelah selesai melakukan sensus untuk satu RT, data tersebut harus di-upload ke server Wilkerstat.
8. Setelah semua data ter-upload, maka sensus selesai.

Wilkestat NFA				
$\delta$	Input			
	A	B	C	$\varepsilon$
q0	q1	$\emptyset$	$\emptyset$	$\emptyset$
q1	$\emptyset$	$\emptyset$	$\emptyset$	q2
q2	q3	$\emptyset$	$\emptyset$	$\emptyset$
q3	$\emptyset$	q4	q5	$\emptyset$
q4	$\emptyset$	q5	$\emptyset$	$\emptyset$
q5	$\emptyset$	$\emptyset$	$\emptyset$	q6
q6	q7, q8	$\emptyset$	$\emptyset$	$\emptyset$
q7	q9	$\emptyset$	$\emptyset$	$\emptyset$
q8	q9	$\emptyset$	$\emptyset$	$\emptyset$
<u>q9</u>	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

- Langkah pertama adalah memulai aplikasi Wilkerstat, sehingga kita memulai dari state q0.
- Selanjutnya, kita menekan tombol "plus" untuk menambahkan project. Ini direpresentasikan oleh transisi dari q0 ke q1 dengan input A.
- Setelah itu, kita memulai melakukan sensus, yang direpresentasikan oleh transisi dari q1 ke q2 dengan input  $\varepsilon$  (epsilon). Epsilon merepresentasikan

langkah kosong, yang artinya tidak ada input yang dibutuhkan untuk melakukan transisi dari satu state ke state lainnya.

- Jika kita berada di depan rumah yang dituju, kita menandai dengan centang. Ini direpresentasikan oleh transisi dari q2 ke q5 dengan input C dan epsilon.
- Namun, jika rumah yang kita tuju adalah rumah miskin, kita harus mengambil foto rumahnya terlebih dahulu. Ini direpresentasikan oleh transisi dari q2 ke q3 dengan input A. Kemudian, jika tidak, kita dapat langsung menekan tombol "Location", yang direpresentasikan oleh transisi dari q3 ke q5 dengan input C dan epsilon.
- Setelah itu, kita memasuki menu perekaman lokasi, yang direpresentasikan oleh transisi dari q5 ke q6 dengan input epsilon.
- Selanjutnya, kita harus memastikan akurasi lokasi yang direkam sekitar 20 untuk miskin dan kaya. Ini direpresentasikan oleh transisi dari q6 ke q7 dan q8 dengan input A.
- Setelah selesai melakukan sensus untuk satu RT, data tersebut di-upload. Ini direpresentasikan oleh transisi dari q7 dan q8 ke q9 dengan input A.
- Langkah terakhir adalah berada di accepting state, yaitu state q9, yang berarti string input yang merepresentasikan langkah-langkah sensus dengan aplikasi Wilkerstat telah diterima dan valid.

Satu rangkaian penuh pada kerja Aplikasi Wilkestat NFA

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$$

$$\Sigma = \{A, B, C, \text{ dan } \epsilon\}$$

$$\delta = \{$$

$$(q_0, A) \rightarrow q_1,$$

$$(q_1, \epsilon) \rightarrow q_2,$$

$$(q_2, A) \rightarrow q_3,$$

$$(q_3, B) \rightarrow q_4,$$

$$(q_3, C, \epsilon) \rightarrow q_5,$$

$$(q_4, B) \rightarrow q_5,$$

$$(q_5, \epsilon) \rightarrow q_6,$$

$$(q_6, A) \rightarrow q_7, q_8,$$

$$(q_7, q_8, A) \rightarrow q_9,$$

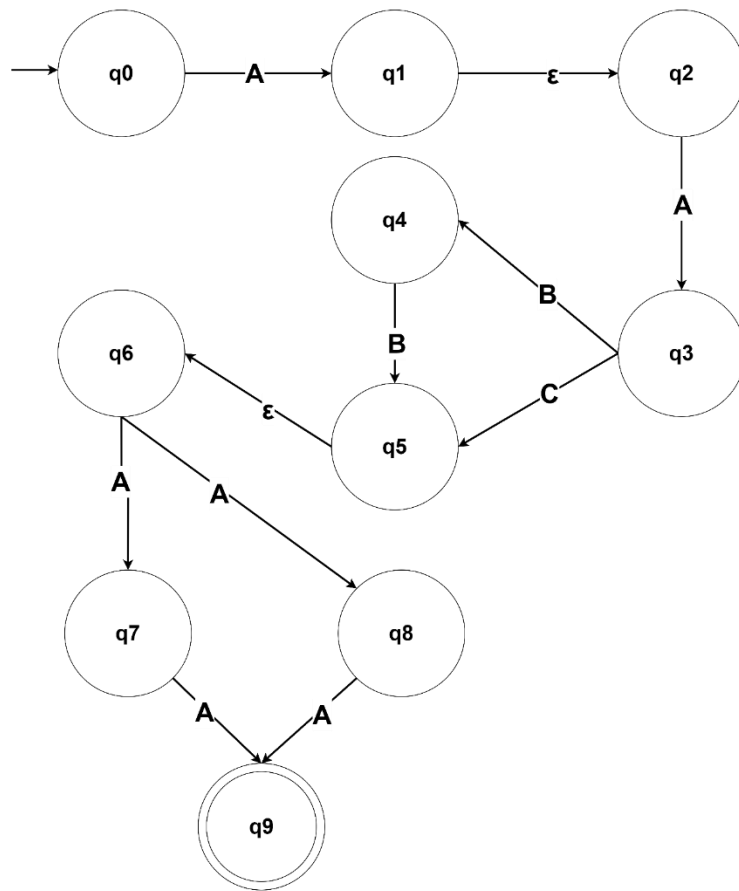
$$\}$$



$q_0$  = Start State

$F = \{q_9\}$  = Accepting State

Wilkestat NFA



### 3.2. Penerapan Deterministic Finite Automata (DFA)

#### 3.2.1. Mesin Aplikasi Wikerstat

##### A. Algoritma DFA :

1. Pertama, buka aplikasi Wilkerstat di perangkat Anda.
2. Setelah itu, tekan tombol "Plus" untuk menambahkan project baru.
3. Mulailah melakukan sensus dengan memasukkan data yang diminta oleh aplikasi. Pastikan Anda mengikuti instruksi dengan cermat dan memasukkan data yang akurat.
4. Jika Anda berada di depan rumah yang dituju, tandai lokasi dengan tanda centang. Jika tidak, lanjutkan ke langkah berikutnya.
5. Jika rumah yang Anda tuju adalah rumah miskin, ambillah foto rumahnya terlebih dahulu. Jika tidak, tekan tombol "Location" untuk merekam lokasi rumah.
6. Setelah Anda selesai melakukan sensus untuk satu RT, pastikan data yang telah Anda masukkan telah disimpan dan di-upload.

Terakhir, selesaikan proses sensus dengan menyelesaikan project atau keluar dari aplikasi

Wilkestat DFA			
$\delta$	Input		
	A	B	C
q0	q1	$\emptyset$	$\emptyset$
q1	q2	$\emptyset$	$\emptyset$
q2	q3	$\emptyset$	$\emptyset$
q3	$\emptyset$	q4	q5
q4	$\emptyset$	q5	$\emptyset$
q5	q6	$\emptyset$	$\emptyset$
q6	q7	$\emptyset$	$\emptyset$
q7	q8	$\emptyset$	$\emptyset$
q8	$\emptyset$	$\emptyset$	$\emptyset$

- Membuka aplikasi Wilkerstat, maka automata ini memulai pada state awal q0. Setelah pengguna menekan tombol "Plus", maka automata berpindah ke state q1 dengan input A, yang menunjukkan bahwa pengguna telah memulai project baru.
- Pengguna memasukkan data yang diminta oleh aplikasi dengan benar, automata berpindah ke state selanjutnya, yaitu q2, q3, q4, q5, q6, q7, dan q8 secara berurutan. Pada state q3, automata memiliki dua kemungkinan transisi, yaitu beralih ke state q4 dengan input B, atau beralih ke state q5 dengan input C.

- Jika pengguna berada di depan rumah yang dituju, maka pengguna dapat menandai lokasi dengan tanda centang dan automata akan berpindah ke state q5. Namun jika pengguna tidak berada di depan rumah yang dituju, maka automata akan langsung berpindah ke state q5.
- Jika rumah yang dituju adalah rumah miskin, maka pengguna harus mengambil foto rumah terlebih dahulu. Jika tidak, pengguna dapat merekam lokasi rumah dengan menekan tombol "Location".
- Setelah pengguna selesai melakukan sensus untuk satu RT, automata akan mengecek apakah data yang telah dimasukkan telah disimpan dan di-upload. Jika ya, automata berpindah ke state akhir q8 dan menganggap project tersebut telah selesai. Namun jika tidak, pengguna harus memastikan bahwa data telah disimpan dan di-upload terlebih dahulu.

Satu rangkaian penuh pada kerja Aplikasi Wilkestat DFA

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$

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

$\delta = \{$

$(q_0, A) \rightarrow q_1,$

$(q_1, A) \rightarrow q_2,$

$(q_2, A) \rightarrow q_3,$

$(q_3, B) \rightarrow q_4,$

$(q_3, C) \rightarrow q_5,$

$(q_4, B) \rightarrow q_5,$

$(q_5, A) \rightarrow q_6,$

$(q_6, A) \rightarrow q_7,$

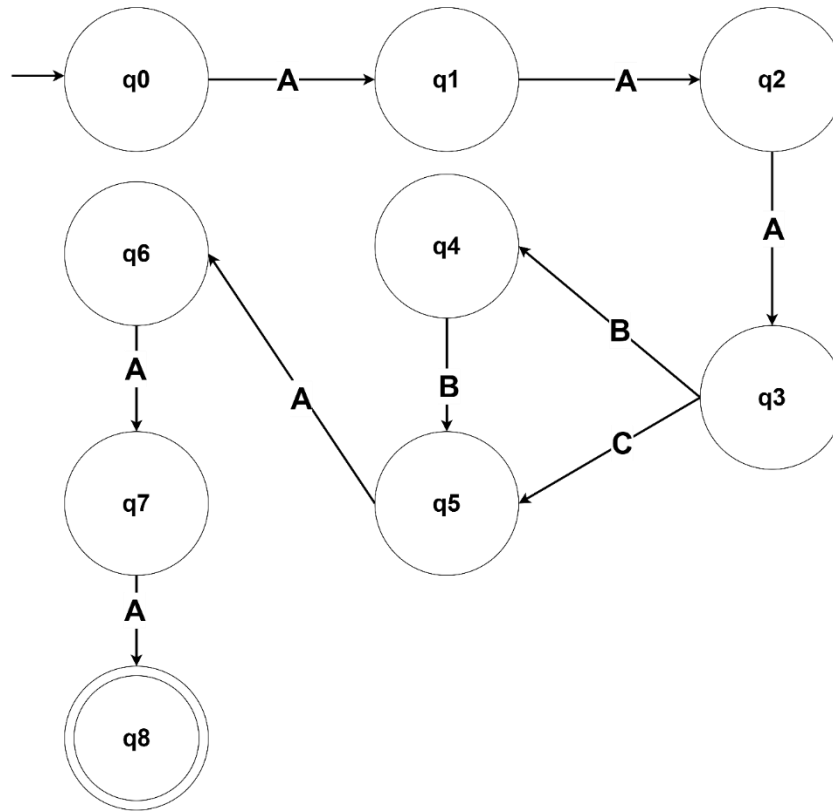
$(q_7, A) \rightarrow q_8,$

$\}$

$q_0 = \text{Start State}$

$F = \{q_8\} = \text{Accepting State}$

### Wilkestat DFA



### 3.3. Penerapan Epsilon Move ke tanpa Epsilon Move

#### 3.3.1. Mesin Aplikasi Wikerstat

A. Algoritma NFA:

1. Buka Aplikasi Wilkerstat yang sudah terinstall di smartphone Anda.
2. Tekan tombol “Plus” untuk menambahkan project.
3. Mulai melakukan sensus dengan memasukkan data yang diperlukan, seperti nama responden, nomor rumah, dan jenis kelamin.
4. Jika lokasi rumah yang dituju telah tercapai, centang tanda di depan rumah tersebut.
5. Jika responden termasuk keluarga miskin, ambil foto rumahnya dan jika tidak, tekan tombol “Location”.
6. Masuk ke menu Perekaman Lokasi untuk memastikan akurasi lokasi yang dicatat (harus sekitar 20 untuk keluarga miskin dan kaya).
7. Setelah selesai melakukan sensus untuk satu RT, data tersebut harus di-upload ke server Wilkerstat.
8. Setelah semua data ter-upload, maka sensus selesai.

Wilkestat NFA				
$\delta$	Input			
	A	B	C	$\epsilon$
q0	q1	$\emptyset$	$\emptyset$	$\emptyset$
q1	$\emptyset$	$\emptyset$	$\emptyset$	q2
q2	q3	$\emptyset$	$\emptyset$	$\emptyset$
q3	$\emptyset$	q4	q5	$\emptyset$
q4	$\emptyset$	q5	$\emptyset$	$\emptyset$
q5	$\emptyset$	$\emptyset$	$\emptyset$	q6
q6	q7, q8	$\emptyset$	$\emptyset$	$\emptyset$
q7	q9	$\emptyset$	$\emptyset$	$\emptyset$
q8	q9	$\emptyset$	$\emptyset$	$\emptyset$
q9	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

- Langkah pertama adalah memulai aplikasi Wilkerstat, sehingga kita memulai dari state q0.
- Selanjutnya, kita menekan tombol "plus" untuk menambahkan project. Ini direpresentasikan oleh transisi dari q0 ke q1 dengan input A.
- Setelah itu, kita memulai melakukan sensus, yang direpresentasikan oleh transisi dari q1 ke q2 dengan input  $\epsilon$  (epsilon). Epsilon merepresentasikan langkah kosong, yang artinya tidak ada input yang dibutuhkan untuk melakukan transisi dari satu state ke state lainnya.
- Jika kita berada di depan rumah yang dituju, kita menandai dengan centang. Ini direpresentasikan oleh transisi dari q2 ke q5 dengan input C dan epsilon.
- Namun, jika rumah yang kita tuju adalah rumah miskin, kita harus mengambil foto rumahnya terlebih dahulu. Ini direpresentasikan oleh transisi dari q2 ke q3 dengan input A. Kemudian, jika tidak, kita dapat langsung menekan tombol "Location", yang direpresentasikan oleh transisi dari q3 ke q5 dengan input C dan epsilon.
- Setelah itu, kita memasuki menu perekaman lokasi, yang direpresentasikan oleh transisi dari q5 ke q6 dengan input epsilon.
- Selanjutnya, kita harus memastikan akurasi lokasi yang direkam sekitar 20 untuk miskin dan kaya. Ini direpresentasikan oleh transisi dari q6 ke q7 dan q8 dengan input A.
- Setelah selesai melakukan sensus untuk satu RT, data tersebut di-upload. Ini direpresentasikan oleh transisi dari q7 dan q8 ke q9 dengan input A.
- Langkah terakhir adalah berada di accepting state, yaitu state q9, yang berarti string input yang merepresentasikan langkah-langkah sensus dengan aplikasi Wilkerstat telah diterima dan valid.

## B. Mesin Aplikasi Wilkerstat

Satu rangkaian penuh pada kerja Aplikasi Wilkestat NFA

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$$

$$\Sigma = \{A, B, C, \text{ dan } \varepsilon\}$$

$$\delta = \{$$

$$(q_0, A) \rightarrow q_1,$$

$$(q_1, \varepsilon) \rightarrow q_2,$$

$$(q_2, A) \rightarrow q_3,$$

$$(q_3, B) \rightarrow q_4,$$

$$(q_3, C, \varepsilon) \rightarrow q_5,$$

$$(q_4, B) \rightarrow q_5,$$

$$(q_5, \varepsilon) \rightarrow q_6,$$

$$(q_6, A) \rightarrow q_7, q_8,$$

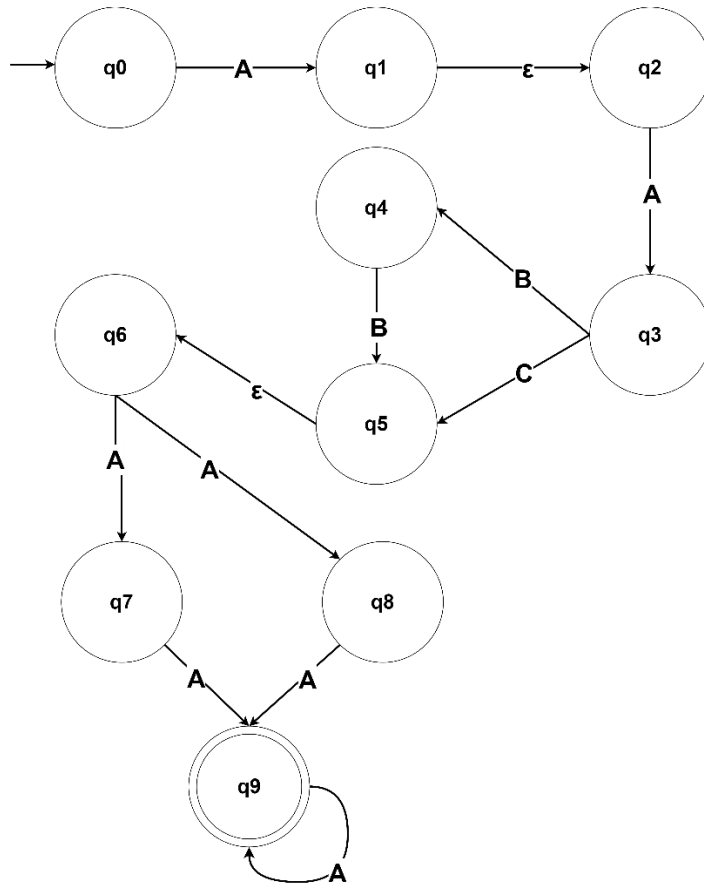
$$(q_7, q_8, A) \rightarrow q_9,$$

}

$$q_0 = \text{Start State}$$

$$F = \{q_9\} = \text{Accepting State}$$

Wilkestat NFA



- Tanpa e-move

a. Tabel Transisi

Wilkestat NFA				
$\delta$	Input			
	A	B	C	$\epsilon$
q0	q1	$\emptyset$	$\emptyset$	$\emptyset$
q1	$\emptyset$	$\emptyset$	$\emptyset$	q2
q2	q3	$\emptyset$	$\emptyset$	$\emptyset$
q3	$\emptyset$	q4	q5	$\emptyset$
q4	$\emptyset$	q5	$\emptyset$	$\emptyset$
q5	$\emptyset$	$\emptyset$	$\emptyset$	q6
q6	q7, q8	$\emptyset$	$\emptyset$	$\emptyset$
q7	q9	$\emptyset$	$\emptyset$	$\emptyset$
q8	q9	$\emptyset$	$\emptyset$	$\emptyset$
q9	q9	$\emptyset$	$\emptyset$	$\emptyset$

b. Tentukan  $\epsilon$ \_closure untuk setiap state

1.  $\epsilon\_cl(q0)=\{q0,\}$

2.  $\epsilon\_cl(q1)=\{q1, q2\}$

3.  $\epsilon\_cl(q2)=\{q2\}$

4.  $\epsilon\_cl(q3)=\{q3\}$

5.  $\epsilon\_cl(q4)=\{q4\}$

6.  $\epsilon\_cl(q5)=\{q5,q6\}$

7.  $\epsilon\_cl(q6)=\{q6\}$

8.  $\epsilon\_cl(q7)=\{q7\}$

9.  $\epsilon\_cl(q8)=\{q8\}$

10.  $\epsilon\_cl(q9)=\{q9\}$



c. Tentukan  $\delta'$  :

$$\begin{aligned} 1. \delta'(q_0, A) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_0), A)) \\ &= \varepsilon\_closure(\delta(\{q_0\}, A)) \\ &= \varepsilon\_closure(q_1) \\ &= \{q_1, q_2\} \end{aligned}$$

$$\begin{aligned} 2. \delta'(q_0, B) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_0), B)) \\ &= \varepsilon\_closure(\delta(\{q_0\}, B)) \\ &= \varepsilon\_closure(\emptyset) \\ &= \{\emptyset\} \end{aligned}$$

$$\begin{aligned} 3. \delta'(q_0, C) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_0), C)) \\ &= \varepsilon\_closure(\delta(\{q_0\}, C)) \\ &= \varepsilon\_closure(\emptyset) \\ &= \{\emptyset\} \end{aligned}$$

$$\begin{aligned} 4. \delta'(q_1, A) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_1), A)) \\ &= \varepsilon\_closure(\delta(\{q_1, q_2\}, A)) \\ &= \varepsilon\_closure(q_3) \\ &= \{q_3\} \end{aligned}$$

$$\begin{aligned} 5. \delta'(q_1, B) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_1), B)) \\ &= \varepsilon\_closure(\delta(\{q_1, q_2\}, B)) \\ &= \varepsilon\_closure(\emptyset) \\ &= \{\emptyset\} \end{aligned}$$

$$\begin{aligned} 6. \delta'(q_1, C) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_1), C)) \\ &= \varepsilon\_closure(\delta(\{q_1, q_2\}, C)) \\ &= \varepsilon\_closure(\emptyset) \\ &= \{\emptyset\} \end{aligned}$$

$$\begin{aligned} 7. \delta'(q_2, A) &= \varepsilon\_closure(\delta(\varepsilon\_closure(q_2), A)) \end{aligned}$$

$$\begin{aligned}
&= \varepsilon\_closure(\delta(\{q2\}, A) \\
&= \varepsilon\_closure(q3) \\
&= \{q3\}
\end{aligned}$$

$$\begin{aligned}
8. \delta'(q2, B) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q2), B) \\
&= \varepsilon\_closure(\delta(\{q2\}, B) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
9. \delta'(q2, C) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q2), C) \\
&= \varepsilon\_closure(\delta(\{q2\}, C) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
10. \delta'(q3, A) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q3), A) \\
&= \varepsilon\_closure(\delta(\{q3\}, A) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
11. \delta'(q3, B) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q3), B) \\
&= \varepsilon\_closure(\delta(\{q3\}, B) \\
&= \varepsilon\_closure(q4) \\
&= \{q4\}
\end{aligned}$$

$$\begin{aligned}
12. \delta'(q3, C) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q3), C) \\
&= \varepsilon\_closure(\delta(\{q3\}, C) \\
&= \varepsilon\_closure(q5) \\
&= \{q5, q6\}
\end{aligned}$$

$$\begin{aligned}
13. \delta'(q4, A) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q4), A) \\
&= \varepsilon\_closure(\delta(\{q4\}, A) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
14. \delta'(q4, B) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q4), B) \\
&= \varepsilon\_closure(\delta(\{q4\}, B)
\end{aligned}$$

$$= \varepsilon\_closure(q5)$$

$$= \{q5, q6\}$$

$$15. \delta'(q4, C)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q4), C))$$

$$= \varepsilon\_closure(\delta(\{q4\}, C))$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

$$16. \delta'(q5, A)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q5), A))$$

$$= \varepsilon\_closure(\delta(\{q5, q6\}, A))$$

$$= \varepsilon\_closure(q7, q8)$$

$$= \{q7, q8\}$$

$$17. \delta'(q5, B)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q5), B))$$

$$= \varepsilon\_closure(\delta(\{q5, q6\}, B))$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

$$18. \delta'(q5, C)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q5), C))$$

$$= \varepsilon\_closure(\delta(\{q5, q6\}, C))$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

$$19. \delta'(q6, A)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q6), A))$$

$$= \varepsilon\_closure(\delta(\{q6\}, A))$$

$$= \varepsilon\_closure(q7, q8)$$

$$= \{q9\}$$

$$20. \delta'(q6, B)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q6), B))$$

$$= \varepsilon\_closure(\delta(\{q6\}, B))$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

$$\begin{aligned}
21. \delta'(q6, C) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q6), C)) \\
&= \varepsilon\_closure(\delta(\{q6\}, C)) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
22. \delta'(q7, A) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q7), A)) \\
&= \varepsilon\_closure(\delta(\{q7\}, A)) \\
&= \varepsilon\_closure(q9) \\
&= \{q9\}
\end{aligned}$$

$$\begin{aligned}
23. \delta'(q7, B) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q7), B)) \\
&= \varepsilon\_closure(\delta(\{q7\}, B)) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
24. \delta'(q7, C) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q7), C)) \\
&= \varepsilon\_closure(\delta(\{q7\}, C)) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
25. \delta'(q8, A) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q8), A)) \\
&= \varepsilon\_closure(\delta(\{q8\}, A)) \\
&= \varepsilon\_closure(q9) \\
&= \{q9\}
\end{aligned}$$

$$\begin{aligned}
26. \delta'(q8, B) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q8), B)) \\
&= \varepsilon\_closure(\delta(\{q8\}, B)) \\
&= \varepsilon\_closure(\emptyset) \\
&= \{ \emptyset \}
\end{aligned}$$

$$\begin{aligned}
27. \delta'(q8, C) \\
&= \varepsilon\_closure(\delta(\varepsilon\_closure(q8), C)) \\
&= \varepsilon\_closure(\delta(\{q8\}, C))
\end{aligned}$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

$$28. \delta'(q9, A)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q9), A))$$

$$= \varepsilon\_closure(\delta(\{q9\}, A))$$

$$= \varepsilon\_closure(q9)$$

$$= \{q9\}$$

$$29. \delta'(q9, B)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q9), B))$$

$$= \varepsilon\_closure(\delta(\{q9\}, B))$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

$$30. \delta'(q9, C)$$

$$= \varepsilon\_closure(\delta(\varepsilon\_closure(q9), C))$$

$$= \varepsilon\_closure(\delta(\{q9\}, C))$$

$$= \varepsilon\_closure(\emptyset)$$

$$= \{ \emptyset \}$$

d. Tabel transisi dari hasil diatas

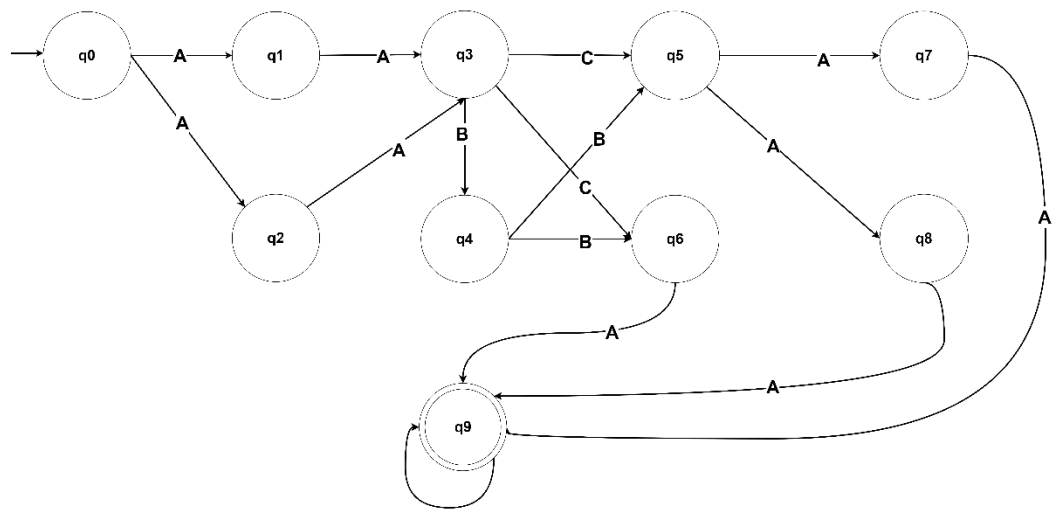
Wilkestat NFA			
$\delta$	Input		
	A	B	C
q0	q1, q2	$\emptyset$	$\emptyset$
q1	q3	$\emptyset$	$\emptyset$
q2	q3	$\emptyset$	$\emptyset$
q3	$\emptyset$	q4	q5, q6
q4	$\emptyset$	q5, q6	$\emptyset$
q5	q7, q8	$\emptyset$	$\emptyset$
q6	q9	$\emptyset$	$\emptyset$
q7	q9	$\emptyset$	$\emptyset$
q8	q9	$\emptyset$	$\emptyset$
q9	q9	$\emptyset$	$\emptyset$

e. Himpunan State akhir dari NFA tanpa  $\epsilon$  – move

- Himpunan State akhir semula adalah {q4}

f. Hasil tanpa E-Move

Wilkestat NFA



## PENUTUP

### 4.1.Kesimpulan

Dalam teori bahasa dan otomata, terdapat konsep-konsep penting seperti NFA (Nondeterministic Finite Automaton), DFA (Deterministic Finite Automaton), dan Epsilon Move. NFA adalah model otomata yang memungkinkan keadaan non-deterministik, sementara DFA adalah model otomata yang hanya memperbolehkan keadaan deterministik. Epsilon Move adalah transisi otomata yang dapat dilakukan tanpa membaca simbol masukan.

Penerapan NFA pada Mesin Aplikasi Wilkerstat memberikan kontribusi signifikan dalam meningkatkan efisiensi dan akurasi pemutakhiran peta digital Wilkerstat. Hal ini akan berdampak positif pada kualitas data yang dikumpulkan oleh BPS dalam kegiatan sensus dan survei, serta memfasilitasi proses pengambilan keputusan yang berbasis data yang lebih baik di masa mendatang.

Penerapan DFA dalam aplikasi Wilkerstat untuk kegiatan sensus dan survei merupakan langkah yang penting untuk meningkatkan efisiensi dan akurasi pendataan. Pemutakhiran bentuk dan posisi peta digital Wilkerstat juga menjadi bagian penting dalam mendukung pengembangan aplikasi ini. Dengan menggunakan DFA, aplikasi Wilkerstat dapat mengenali dan memvalidasi data yang masuk, sehingga proses pengolahan data dapat dilakukan dengan lebih cepat dan akurat

Penerapan Epsilon Move pada Mesin Aplikasi Wikerstat merupakan pendekatan yang dapat mempermudah pemutakhiran wilkerstat dengan mempertahankan kontinuitas data. Namun, perlu dipertimbangkan juga penerapan Wikerstat tanpa Epsilon Move agar dapat memenuhi kebutuhan pemutakhiran yang mungkin memiliki kendala teknis atau keinginan untuk menyederhanakan proses. Pemutakhiran bentuk dan posisi peta digital Wilkerstat yang akurat dan terbaru sangat penting untuk mendukung kegiatan pendataan BPS dalam kegiatan REGSOSEK 2022.

Dalam Kesimpulannya, pemahaman tentang NFA, DFA, dan Epsilon Move merupakan landasan penting dalam teori bahasa dan otomata. Penerapannya dalam Mesin Aplikasi wilkerstat dapat meningkatkan kinerja mesin dan mengoptimalkan pengenalan pola serta pemrosesan bahasa alami. Penggunaan graf sebagai representasi visual juga membantu dalam memahami struktur dan alur otomata.

## DAFTAR PUSTAKA

- [1]<https://www.trivusi.web.id/2022/08/finite-state-automata.html>
- [2] <https://media.neliti.com/media/publications/226255-telaah-teoritis-finite-state-automata-de-8b056b07.pdf>
- [3]<https://www.studocu.com/id/document/universitas-lambung-mangkurat/teori-bahasa-dan-otomata/materi-finite-state-automata/37374241>
- [4][https://elearning.sttindonesia.ac.id/pluginfile.php/557/mod\\_resource/content/1/ Teori% 20Bahasa% 20Dan% 20Otomata% 20% 28Handout% 29.pdf](https://elearning.sttindonesia.ac.id/pluginfile.php/557/mod_resource/content/1/Teori%20Bahasa%20Dan%20Otomata%20%28Handout%29.pdf)
- [5]<https://repository.unikom.ac.id/36690/1/Pertemuan5.pdf>
- [6]<http://raudhatuljannah97.blogspot.com/2017/05/nfa-dengan-move.html>
- [7] Heru Cahya Rustamaji, S. M. (2004). Teori Bahasa dan Otomata. Yogyakarta: Universitas Pembangunan Nasional “ Veteran “ Yogyakarta.