

# **PENERAPAN FINITE STATE AUTOMATA (FSA) PADA APLIKASI BUPASTA (BUMI PASUNDAN SARAN TRANSPORTASI AMAN) IMPLEMENTASI ALGORITMA DIJKSTRA**

**Muhammad Fauzan Lubada<sup>1</sup>, Muhammad Rahardian Baihaqi<sup>2</sup>,  
Muhammad Jalalullail<sup>3</sup>, Nazwa Yulianti<sup>4</sup>, Rafly Nur Ramadhan<sup>5</sup>**  
Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia

## **Article Info**

### **Article history:**

Received, 2024

Revised, 2024

Accepted, 2024

### **Keywords:**

Finite State Automata

Algoritma Dijkstra

Transportasi Umum

Kota Bandung

## **ABSTRACT (10 PT)**

Penelitian ini bertujuan untuk menerapkan Finite State Automata (FSA) pada aplikasi BUPASTA (Bumi Pasundan Saran Transportasi Aman) yang menggunakan algoritma Dijkstra. Aplikasi ini dirancang untuk memberikan rekomendasi rute terpendek transportasi umum di Kota Bandung. FSA digunakan untuk merepresentasikan alur kerja algoritma Dijkstra dalam menentukan rute perjalanan. Alur kerja tersebut meliputi input lokasi awal dan tujuan, inisialisasi dan perhitungan jarak, evaluasi node dan rute, hingga pemilihan transportasi. Hasil penelitian menunjukkan bahwa penerapan FSA pada aplikasi BUPASTA berhasil dilakukan. FSA memudahkan pengguna dalam memahami alur kerja aplikasi dan memperjelas logika yang digunakan dalam menentukan rute terpendek. Diharapkan aplikasi BUPASTA ini dapat menjadi solusi yang efektif dalam mengatasi permasalahan transportasi di Kota Bandung.

## **Corresponding Author:**

Name of Corresponding Author,

Informatics Department, Faculty of Science & Technology, UIN Sunan Gunung Djati Bandung

Jl. A. H. Nasution No. 105, Cibiru, Bandung, Indonesia. 40614

Email: join@uinsgd.ac.id

## **1. Pendahuluan**

Perkembangan teknologi yang semakin pesat dalam beberapa dekade terakhir, terutama sejak ditemukannya internet dan computer pada tahun 1980-an (Frictarani dkk, 2023). Perkembangan teknologi telah membawa perubahan signifikan pada berbagai bidang seperti transportasi, komunikasi, Kesehatan, Pendidikan dan hiburan. Perkembangan teknologi yang dilakukan para peneliti telah memberikan sumber berupa informasi dan komunikasi yang mendukung pengembangan ilmu-ilmu lainnya.

Teori komputasi adalah cabang dari ilmu komputer yang mempelajari kemampuan dan batasan dari mesin serta model komputasi. Dalam konteks teori bahasa dan automata, teori komputasi fokus pada pemahaman tentang apa yang dapat dan tidak dapat dicapai oleh berbagai model komputasi. Salah satu konsep kunci dalam teori komputasi adalah teori automata, yang mempelajari mesin-mesin abstrak atau sistem matematis yang dapat memecahkan masalah komputasional. Automata adalah model teoritis yang menggambarkan mesin yang bekerja secara otomatis, dan sering dikaitkan dengan teori bahasa formal karena kemampuannya mengenali bahasa tertentu. Automata dapat merepresentasikan bahasa formal terbatas, bahkan jika himpunan bahasa tersebut tidak terbatas. Model automata ini digunakan sebagai dasar untuk memahami kemampuan komputasi dan batasan dari mesin-mesin komputasi. (Vinawanda Khodijah, dkk. 2024). Pada penelitian kali ini Finite State Automata dibutuhkan sebagai teori dasar dan juga sebagai acuan dalam penelitian dimana untuk penggambaran state pada Algoritma Dijkstra. Penilitan Anggun Yuli Asih,dkk (2021). perancangan FSA, digunakan pada desain mesin penjualan beras otomatis dapat bermanfaat bagi pengguna untuk lebih mengefisienkan waktu berbelanja khususnya belanja beras, dan juga sebagai pemutus rantai penyebaran virus karena dapat menghindari dari kerumunan seperti di pasar. Pada penelitian kali ini rancangan FSA digunakan untuk Algoritma Dijkstra menghitung rute terpendek dari titik(node) asal ke titik tujuan

Kota Bandung merupakan sebuah kota sekaligus ibukota provinsi Jawa Barat. Kota Bandung merupakan kota terbesar keempat di Indonesia dan menjadi kota terpadat di Indonesia setelah Jakarta dengan kepadatan mencapai 15.051 jiwa/km<sup>2</sup>. Jumlah penduduk kota Bandung sebanyak 2.569.107 pada Pada akhir tahun 2023, selain itu kota Bandung juga gencar dalam pembangunan infrastruktur yang ditujukan untuk menjalankan fungsi kota pintar atau smart city (Mulyana, 2021). Sebagai kota terpadat ke 2 di Indonesia tentunya terdapat banyak permasalahan yang terjadi salah satunya ialah kemacetan. Kemacetan merupakan salah satu masalah terbesar yang dihadapi oleh masyarakat urban di kota-kota dunia termasuk Indonesia. Masalah ini pula yang kini turut menjangkit dan sayangnya belum dapat terselesaikan dengan baik. Terdapat berbagai upaya yang dilakukan untuk mengatasi kemacetan yang terjadi di kota Bandung, salah satunya dengan meningkatkan pelayanan angkutan umum.

## 2. Metode Penelitian

### Teori Dasar

#### Algoritma Dijkstra

Didalam penelitian yang dilakukan oleh Maria C. B (2022). Algoritma Dijkstra dirancang untuk menyelesaikan masalah jalur terpendek pada graf dengan bobot bernilai non-negatif di setiap simpul. Algoritma ini juga termasuk ke dalam kategori algoritma serakah (greedy) karena fokus utamanya adalah mencari solusi optimal. Algoritma ini menentukan jalur terpendek dengan mencari bobot minimum pada graf berbobot, sehingga dapat menemukan jarak total terkecil antara dua atau lebih titik dalam graf. Sebagai contoh, jika G adalah graf berarah dengan simpul-simpul yang berlabel, tujuan algoritma ini adalah menemukan jalur terpendek dari simpul v<sub>1</sub> ke simpul v<sub>n</sub>. Algoritma dimulai dari simpul v<sub>1</sub>, dan pada setiap iterasi, algoritma memilih simpul dengan jumlah bobot kumulatif terkecil dari titik awal. Simpul-simpul yang telah dipilih akan diberi tanda dan tidak diperhitungkan lagi dalam iterasi selanjutnya. (Maria Chatrin Bunaenm dkk. 2022)

1. Identifikasi titik awal, yang akan berfungsi sebagai node pertama atau node awal. Selanjutnya, hitung bobot (jarak) dari node awal ke setiap node terdekat satu per satu. Algoritma Dijkstra akan mencari bobot terkecil dari satu node ke node lainnya, kemudian melanjutkan ke node berikutnya secara berurutan.
2. Tentukan bobot (jarak) dari setiap node ke node lainnya, dengan menetapkan nilai 0 pada node awal dan nilai tak terhingga pada semua node lainnya.
3. Tandai semua node yang belum dilalui dan tetapkan node awal sebagai "node keberangkatan."

4. Dari node keberangkatan, hitung jarak ke node terdekat yang belum dilalui. Jika jarak baru ini lebih kecil daripada jarak yang telah dihitung sebelumnya, hapus data lama dan perbarui jarak dengan nilai yang lebih pendek.
5. Setelah menghitung dan mempertimbangkan jarak ke semua node yang berdekatan, beri label pada node yang telah dilalui sebagai "node yang dilalui." Node yang sudah dilalui tidak akan diperiksa kembali, dan jarak yang disimpan akan mencerminkan lintasan terpendek.
6. Pilih "node yang belum dilalui" dengan bobot terkecil (dari node keberangkatan) sebagai "node keberangkatan baru" untuk iterasi berikutnya dan ulangi proses tersebut. (Aldy C, Fauziah, Winarsih . 2020)

### Finite State Automata

Pemodelan matematika yang menerima input dari sebuah himpunan terbatas disebut sebagai Finite State Automaton (FSA). FSA mirip dengan graf, tetapi memiliki jumlah state yang terbatas. FSA memiliki state awal dan fungsi transisi yang memungkinkan perpindahan dari satu state ke state lainnya berdasarkan input yang diberikan. Selain itu, terdapat himpunan state penerima (accepting states) yang menentukan apakah suatu input akan diterima atau ditolak. Prinsip kerja FSA melibatkan perubahan state berdasarkan input hingga mencapai state akhir yang menentukan hasilnya. (Banu W. Y, Teofilus R, dan Saptadi N. 2017). Didalam sebuah Finite State Automata (FSA) terdapat himpunan state yang bisa menghasilkan sebuah Prinsip kerja adalah sebagai berikut:

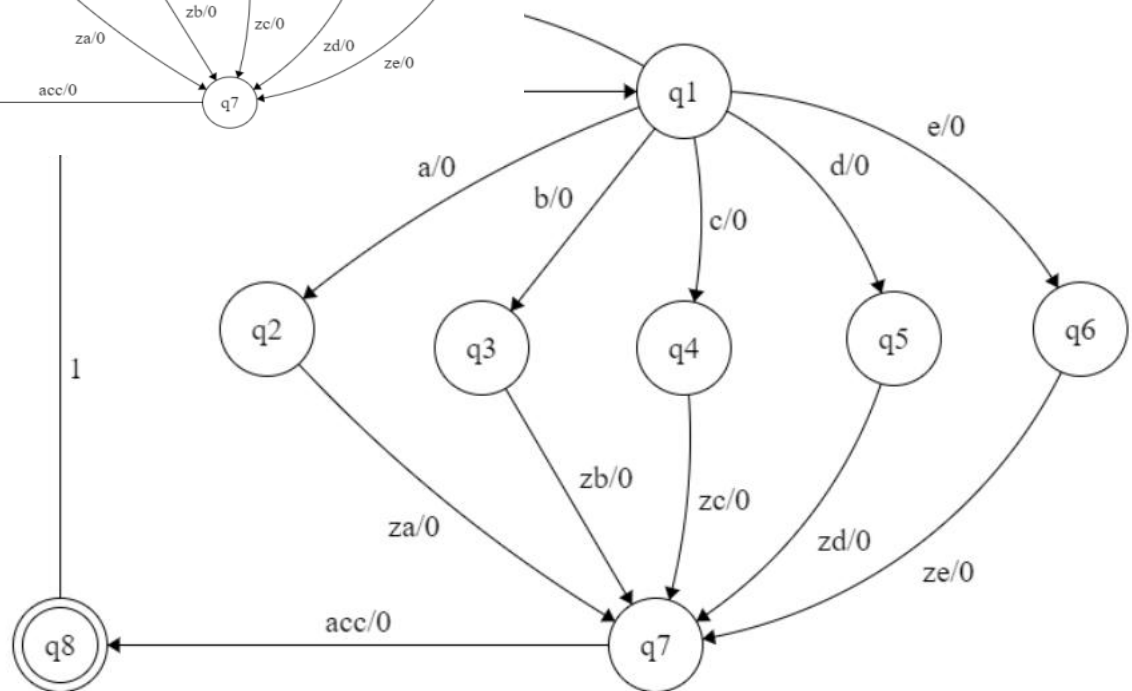
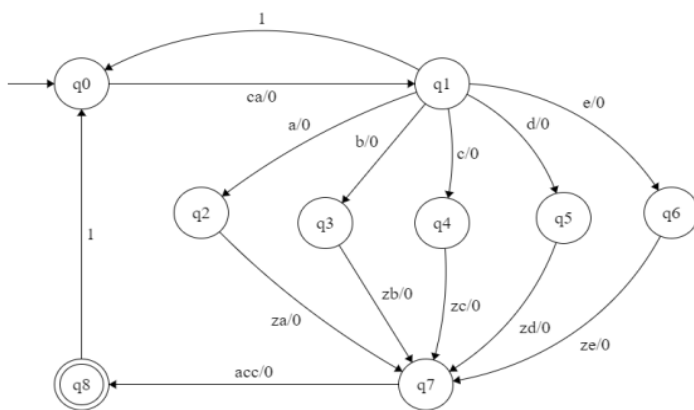
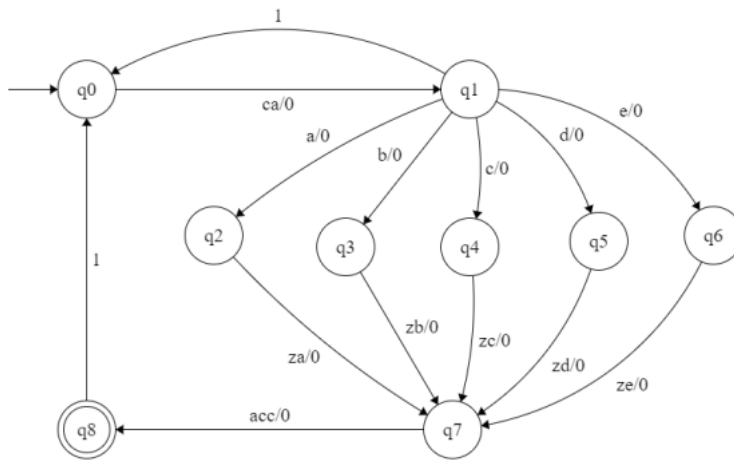
1. Menerima string input
2. Membaca (memindai substring) karakter awal dengan kontrol sesuai dengan kondisi awal
3. Menggunakan kontrol dan karakter awal yang telah dipindai, state akan pindah ke kondisi baru,
4. Proses ini berlanjut sampai semua string dipindai atau dibaca,
5. Jika keadaan terakhir berada di set keadaan akhir yang telah ditentukan, maka string diterima atau dikenali oleh FSA. Jika tidak, maka string ditolak atau tidak dikenali oleh FSA (Fergie Joanda Kaunang. (2019.)

FSA terdiri atas dua jenis, yaitu Deterministic Finite Automata (DFA) dan Non-Deterministic Finite Automata (NFA). Keduanya dibedakan berdasarkan mekanisme transisi state-nya. Pada DFA, setiap state hanya memiliki satu jalur transisi untuk setiap input yang diterima, sehingga jalur yang harus diikuti sangat jelas dan terprediksi. Sebaliknya, pada NFA, sebuah state dapat memiliki beberapa jalur transisi untuk input yang sama, sehingga terdapat lebih dari satu kemungkinan jalur yang bisa diambil, termasuk opsi untuk tidak melakukan transisi. (Nugraha,dkk. 2020)

Selain itu, terdapat perbedaan lain yang penting, yaitu dalam hal penerimaan input kosong (epsilon). DFA tidak dapat menerima input kosong, yang berarti setiap transisi harus dipicu oleh suatu input tertentu. Di sisi lain, NFA dapat menerima input kosong, yang memungkinkan transisi antar state terjadi tanpa adanya input tambahan. Oleh karena itu, NFA sering dianggap lebih fleksibel dalam menentukan transisi. (Suprpto dan Fauziah 2020)

Didalam artikel yang diteliti oleh Refi Riduan Achmad.(2021) Finite State Automata didefinisikan dengan lima tuple.  $M=(Q,\Sigma,\delta,S_0,\lambda)$  dengan:

- $Q$  = himpunan state
- $\Sigma$  = himpunan simbol input
- $\delta$  = fungsi transisi  $\delta : Q \times \Sigma$
- $S_0$  = state awal / initial state ,  $S_0 \in Q$
- $F$  = state akhir,  $F \subseteq Q$



Gambar 1. Finite State Automata VM Peminjaman Alat Praktik

Sumber : Dwi Krisnadi, dkk(2021)

Pada diagram state diatas, konfigurasi mesin dapat dijelaskan sebagai berikut:

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

$\Sigma = \{0, 1, ca, a, b, c, d, e, za, zb, zc, zd, ze, acc\}$

$S = \{q_0\}$

$F = \{q_8\}$

Fungsi Transisinya  $\delta$  :

$\delta(q_0, 1) = q_0$

$\delta(q_0, ca) = q_1$

$\delta(q_1, a) = q_2$

$\delta(q_1, b) = q_3$

$\delta(q_1, c) = q_4$

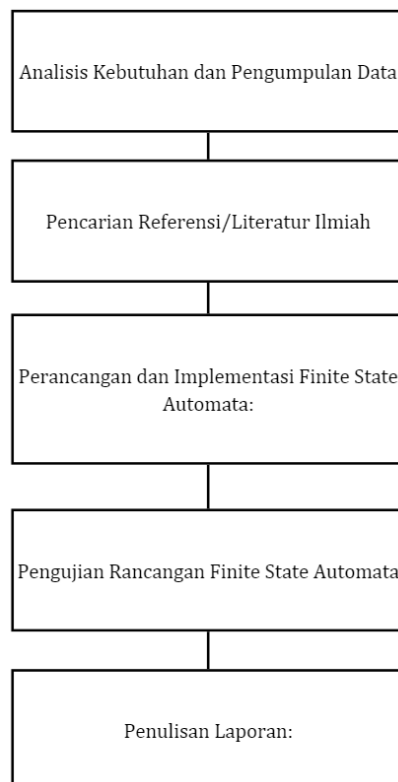
$\delta(q_1, d) = q_5$

$\delta(q_1, e) = q_6$

$\delta(q_2, za) = q_7$

### Metode dan Perancangan

Motode perancangan penelitian ini dibagi menjadi 5 tahap :



### Gambar 2. Tahapan Penelitian

Tahap metode yang diterapkan pada penelitian ini adalah dengan menggunakan FSA pada Rekomendasi Transportasi Umum Di Kota Bandung Menggunakan Algoritma Dijkstra yang dibagi menjadi 5 tahap antara lain :

1. Analisis Kebutuhan dan Pengumpulan Data: Menentukan kebutuhan sistem dan mengumpulkan data yang diperlukan sebagai dasar penelitian. Tahap ini mengumpulkan data bagaimana sistem yang dilakukan pada Algoritma Dijkstra
2. Pencarian Referensi/Literatur Ilmiah: Melakukan studi literatur untuk memahami teori dan konsep melalui artikel jurnal serta sumber mengenai pembahasan terkait penelitian tersebut; Tahap Perancangan dan Implementasi Finite State Automata,
3. Perancangan dan Implementasi Finite State Automata: Pada tahap ini akan dilakukan perancangan finite state automata menggunakan *NFA (Non-deterministic Finite Automaton)* , dan juga melakukan pengujian yang dilakukan dengan evaluasi terhadap keseluruhan perancangan finite state automata pada Rekomendasi Transportasi Umum dengan Algoritma Dijkstra Menggunakan DFA
4. Pengujian Rancangan Finite State Automata: Melakukan uji coba terhadap rancangan FSA untuk memastikan sistem bekerja sesuai dengan spesifikasi dan memperbaiki kesalahan yang ditemukan.
5. Penulisan Laporan: Menyusun laporan akhir yang mencakup keseluruhan proses penelitian, temuan, dan analisis hasil.

### Hasil dan Pembahasan

Rancangan state diagram pada pemetaan sistem saran transportasi umum di Bandung dapat dilihat pada Gambar 3 , yang menjelaskan tentang pemetaan sistem transportasi umum di Bandung berdasarkan tahap-tahap yang dilakukan pengguna dalam memilih transportasi umum. Berikut adalah sistem atau tahap yang dilakukan pengguna dalam menggunakan sarana transportasi umum di Bandung, yaitu:

- Q0: Start (akses aplikasi) – Pengguna membuka aplikasi sarana transportasi untuk memulai proses.  
 Q1: Lokasi Saat Ini (user input lokasi) – Pengguna memasukkan lokasi mereka saat ini sebagai titik awal untuk rute perjalanan.  
 Q2: Lokasi Tujuan (user input tujuan) – Pengguna memasukkan lokasi tujuan yang ingin dicapai sebagai titik akhir perjalanan.  
 Q3: Inisialisasi (Initialization) – Sistem menginisialisasi proses dengan mengumpulkan data lokasi awal dan tujuan, mempersiapkan algoritma Dijkstra.  
 Q4: Mengatur Inisialisasi Jarak (Initializing Distance Calculation) – Sistem mempersiapkan perhitungan jarak ke semua node dari lokasi saat ini (Q1). Algoritma Dijkstra menginisialisasi node awal dengan jarak 0 dan node lainnya dengan jarak tak terhingga.  
 Q5: Inisialisasi & Memilih Node (Initialization & Node Selection) – Sistem mulai memilih node untuk diproses berdasarkan jarak terpendek, mengevaluasi node tetangga dan jaraknya.  
 Q6: Evaluasi (Evaluation) – Sistem mengevaluasi pilihan transportasi yang tersedia (bus, kereta, dll.) berdasarkan rute terpendek dan preferensi pengguna. Algoritma Dijkstra memeriksa rute optimal menuju tujuan.  
 Q7: Memperbarui Jarak (Updating Distance) – Sistem memperbarui jarak ke node tetangga saat jalur baru dievaluasi dan ditemukan jalur yang lebih pendek.  
 Q8: Menandai Jarak yang Sudah Dikunjungi (Marking Visited Nodes) – Sistem menandai node yang sudah dikunjungi setelah mengevaluasi tetangganya untuk menghindari perhitungan yang berulang.

Q9: Node yang Belum Dikunjungi (Unvisited Nodes) – Sistem mempertahankan daftar node yang belum dikunjungi dan melanjutkan pemilihan node terdekat yang belum dikunjungi untuk menyempurnakan rute.

Q10: Semua Node Sudah Dikunjungi (All Nodes Visited) – Semua node telah diproses, dan rute optimal ke tujuan telah ditentukan.

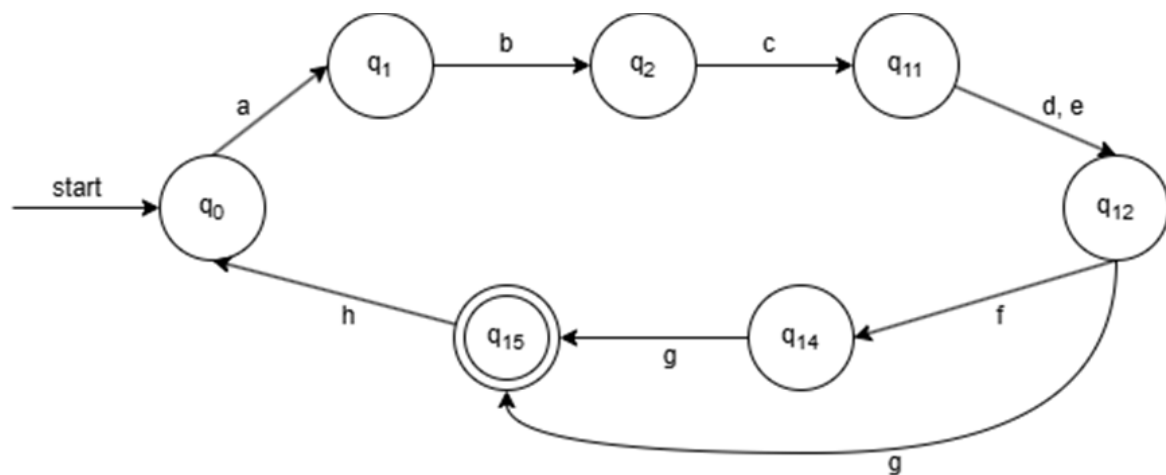
Q11: Pemilihan Transportasi (Transport Selection) – Pengguna memilih moda transportasi yang sesuai dengan pilihan dan preferensi mereka berdasarkan rute yang telah dihitung (misalnya, bus, kereta, ojek).

Q12: Perjalanan Dimulai (Journey Begins) – Setelah pembayaran (jika diperlukan) diverifikasi, pengguna memulai perjalanan dengan moda transportasi yang dipilih.

Q13: Tiba di Tujuan (Arrival at Destination) – Pengguna tiba di tujuan sesuai dengan rute dan moda transportasi yang dipilih sebelumnya.

Q14: Umpan Balik Pengguna/Rating (User Feedback/Rating) – Setelah perjalanan selesai, sistem meminta pengguna untuk memberikan umpan balik atau menilai pengalaman mereka menggunakan aplikasi transportasi.

Q15: End (Akhir) – Pengguna keluar dari aplikasi dan sesi berakhir.



Gambar 3. Diagram State Buspasta

$Q = \{Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q14, Q15\}$

$\odot = \{a, b, c, d, e, f, g, h\}$

$S = \{Q0\}$

$F = \{Q15\}$

$\delta = \{((Q0, a) \rightarrow Q1), ((Q1, b) \rightarrow Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10), ((Q10, c) \rightarrow Q11), ((Q11, d) \rightarrow Q12, Q13), ((Q11e, ) \rightarrow Q12, Q13), ((Q12, Q13, f) \rightarrow Q14), ((Q12, Q13, g) \rightarrow Q15), ((Q14, g) \rightarrow Q15), ((Q15, h) \rightarrow Q0)\}$

Untuk keterangan abjad, himpunan simbol input yang menyatakan apa yang dilakukan oleh user adalah sebagai berikut

Q0: Start (akses aplikasi)

Q1: Lokasi Saat Ini (user input lokasi)

Q2: Lokasi Tujuan (user input tujuan)

Q3: Inisialisasi

Q4: Mengatur Inisialisasi Jarak

Q5: Inisialisasi & Memilih Node

Q6: Evaluasi

Q7: Memperbarui Jarak

Q8: Menandai Jarak yang Sudah Dikunjungi

Q9: Node yang Belum Dikunjungi  
 Q10: Semua Node Sudah Dikunjungi  
 Q11: Pemilihan Transportasi  
 Q12: Perjalanan dimulai  
 Q13 Tiba di Tujuan  
 Q14: Umpan Balik Pengguna/ Rating  
 Q15: End

a: User Input Lokasi  
 b: User Input Lokasi Tujuan  
 c: Input Transportasi  
 d: Input Pembayaran (Transportasi dengan pesanan)  
 e: User tidak melakukan proses pembayaran, (Transportasi tanpa pesanan)  
 f: Pemberian Umpan Balik  
 g: User keluar aplikasi  
 h: Menginput untuk kembali ke lokasi tujuan baru

Untuk keterangan abjad, himpunan simbol input yang menyatakan apa yang dilakukan oleh user adalah sebagai berikut:

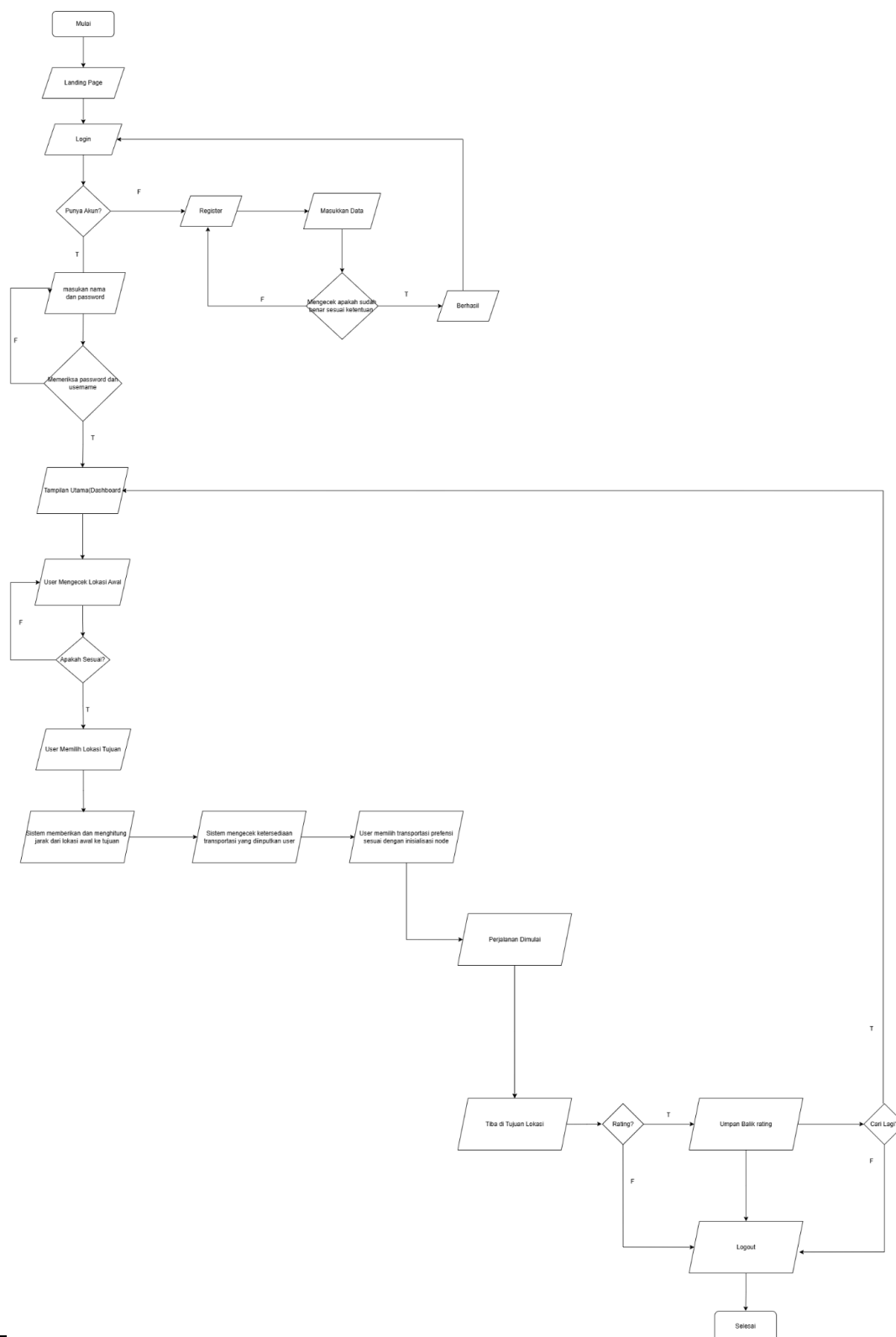
$\delta'$	a	b	c	d	e	f	g	h
q1	q1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q2	$\emptyset$	q2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q3	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q4	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q5	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q6	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q7	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q8	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q9	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q10	$\emptyset$	$\emptyset$	q11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q11	$\emptyset$	$\emptyset$	$\emptyset$	q12	q12	$\emptyset$	$\emptyset$	$\emptyset$
q12	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	q14	q15	$\emptyset$



q13	∅	∅	∅	∅	∅	q14	q15	∅
q14	∅	∅	∅	∅	∅	∅	q15	∅
q15	∅	∅	∅	∅	∅	∅	∅	q0

Tabel 1. Relasi Transisi State  
(Sumber : Hasil Penelitian . 2024)

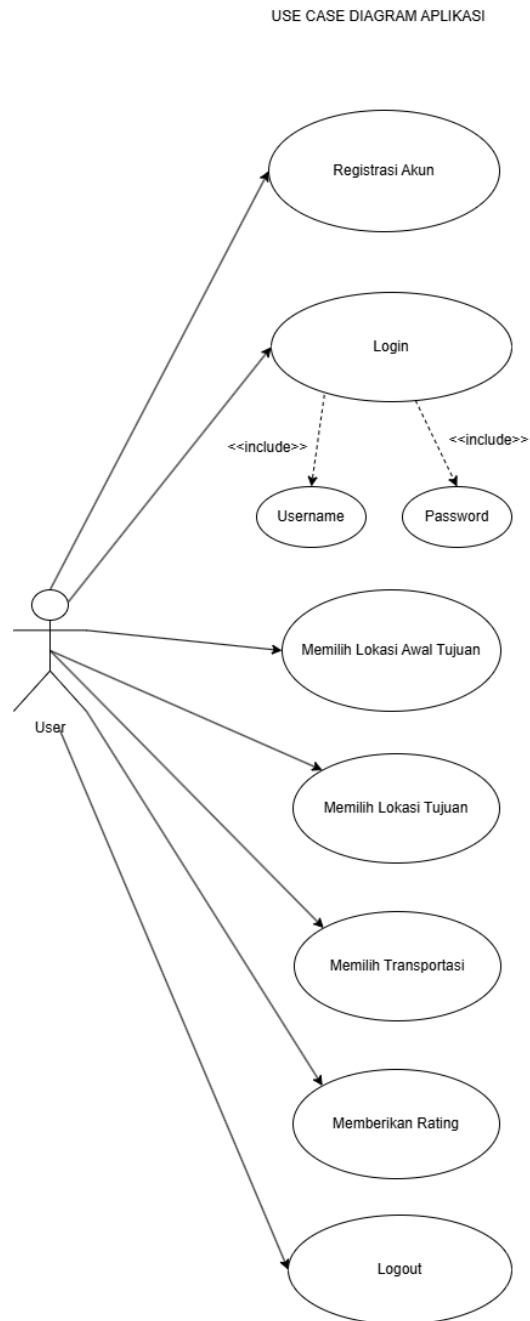
Tabel 1 menggambarkan perpindahan state ketika menerima suatu input. Sebagai contoh, jika state Q1 menerima input `a`, maka akan beralih ke state Q2, seperti yang ditunjukkan pada Gambar 3. Selanjutnya, ketika state Q2 menerima input `b`, state tersebut akan berpindah ke state Q3. Setiap perpindahan state akan menghasilkan ∅ (bahasa kosong) jika input yang masuk tidak sesuai. Misalnya, state S1 yang menerima input `b` akan menghasilkan ∅ (bahasa kosong). Hal ini juga berlaku untuk state lainnya yang menerima input yang tidak sesuai.



Gambar 4. Flowchart penerapan FSA untuk rekomendasi transportasi

(Sumber : Hasil Penelitian. 2024)

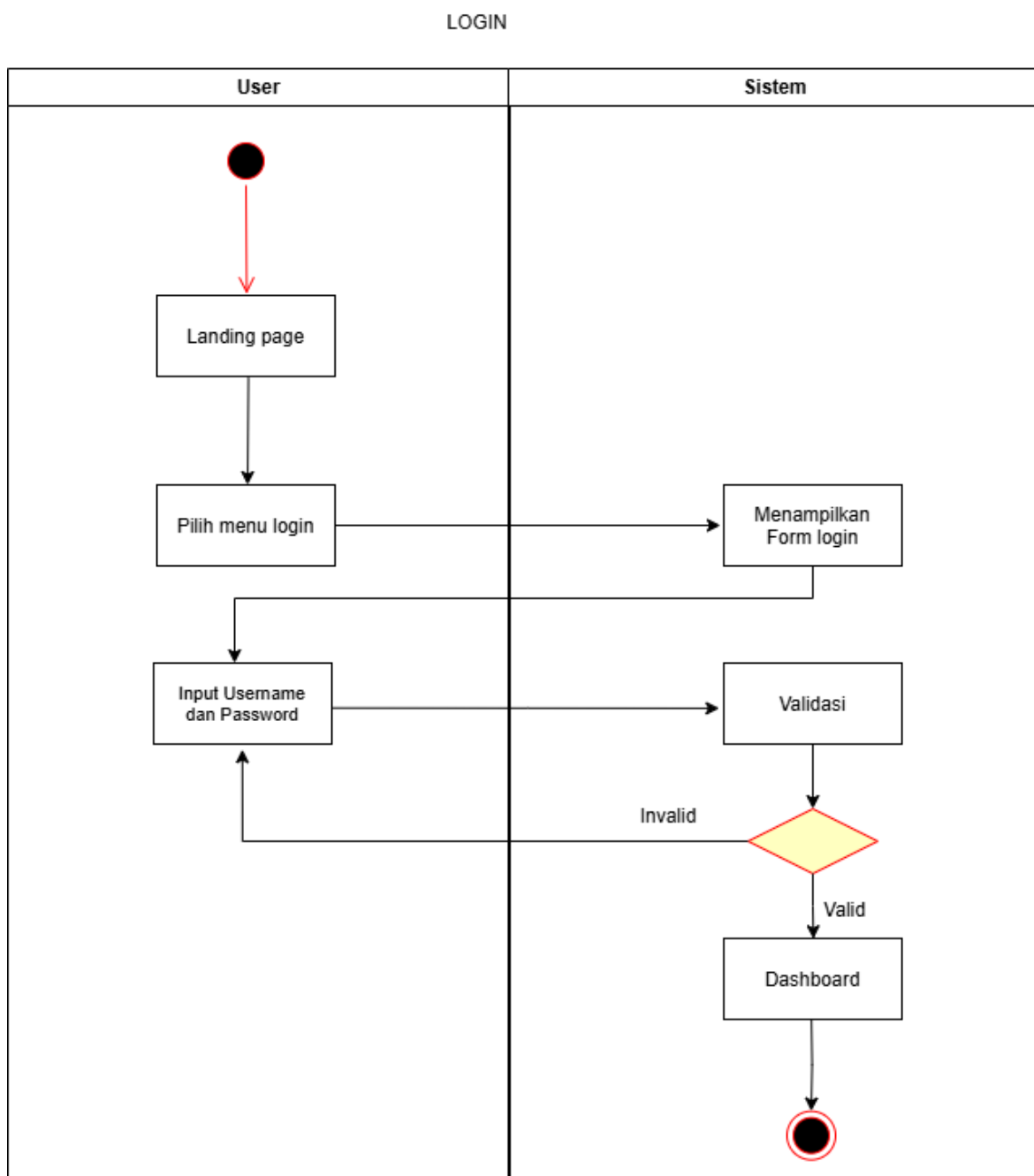
### Use Case aplikasi



Gambar 5.Use Case Aplikasi

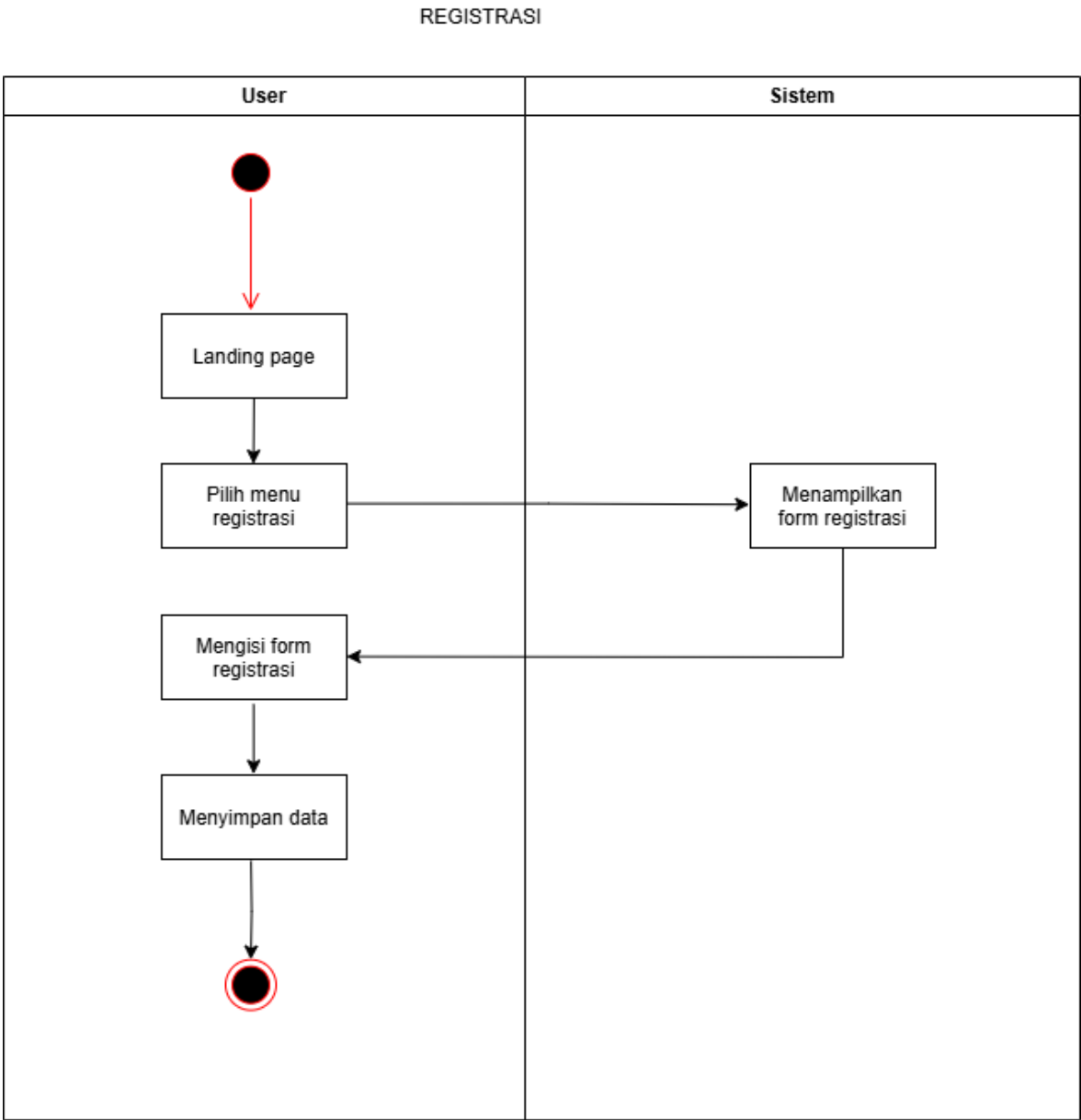
(Sumber : Hasil Penelitian. 2024)

## Activity Diagram



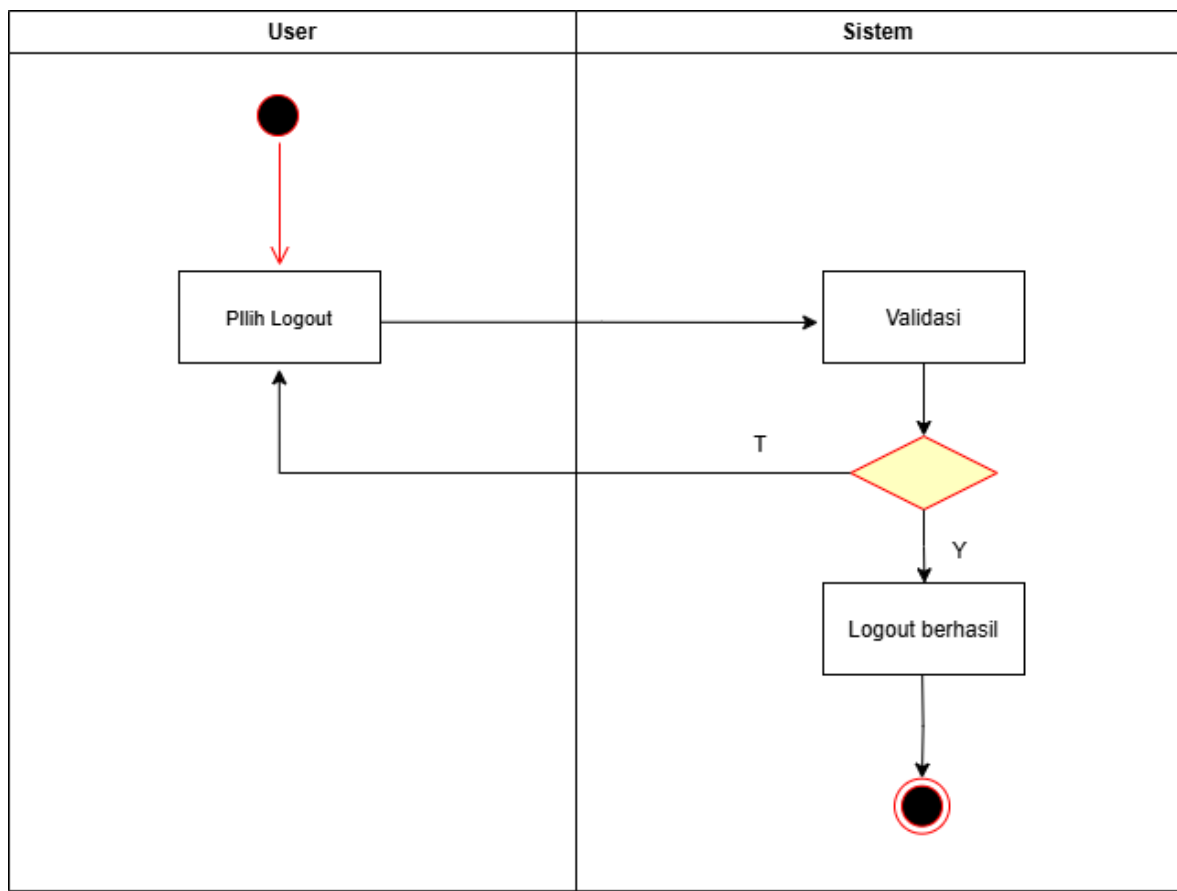
Gambar 6 Activity Diagram Login

(Sumber : Hasil Penelitian. 2024)



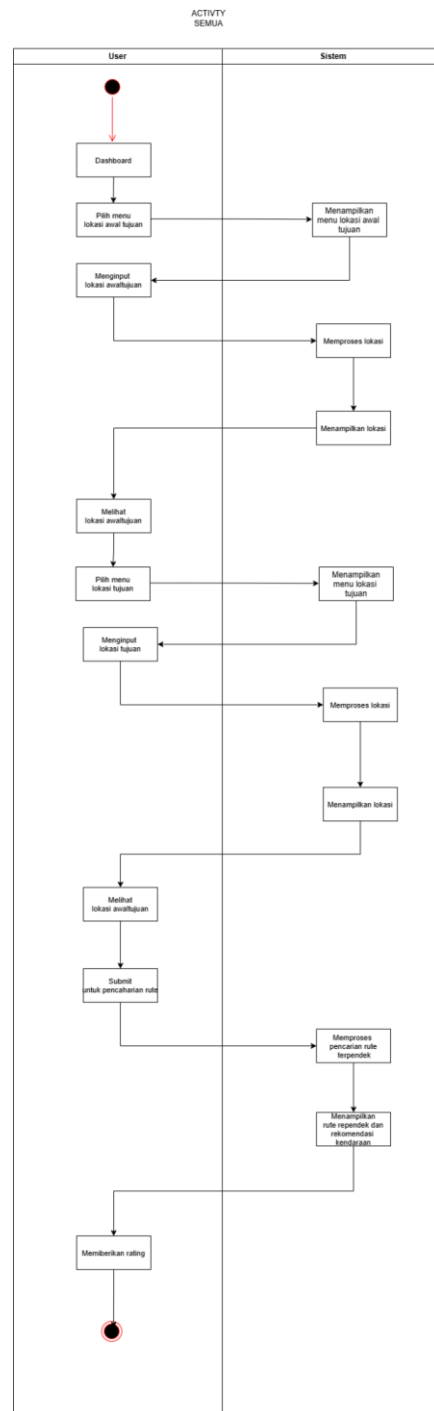
Gambar 7 Activity Diagram Registrasi

(Sumber : Hasil Penelitian. 2024)



Gambar 8 Activity Diagram Logout

(Sumber: Hasil Penelitian . 2024)



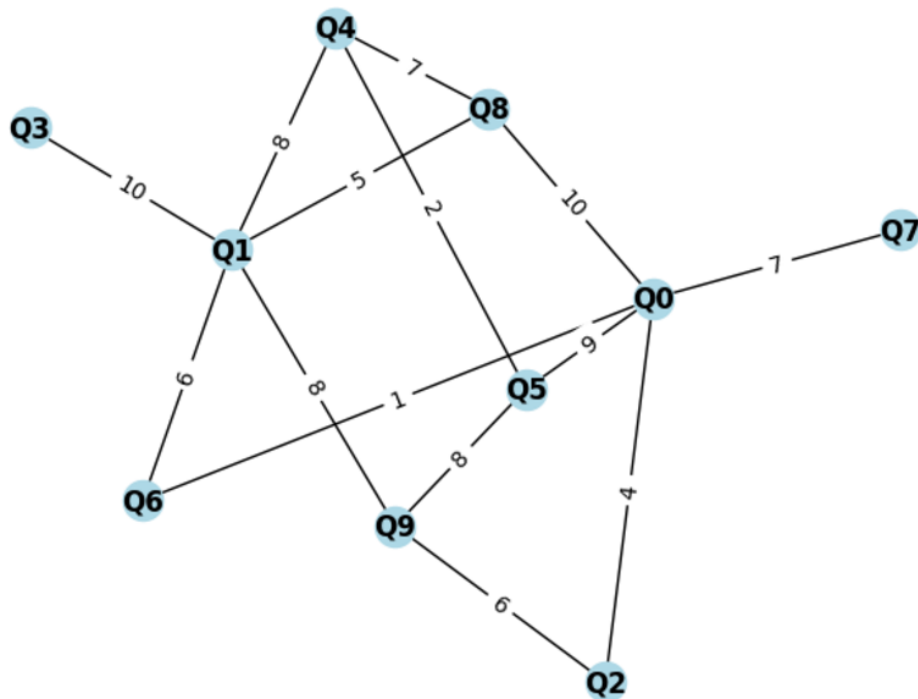
Gambar 9 Activity Diagram Aplikasi



(Sumber : Hasil Penelitian. 2024)

Rancangan ini merupakan gambaran konsep aplikasi yang dibuat, berguna untuk membantu user atau pengguna menentukan rute terpendek dan mengetahui rute apa yang akan dilalui

Masukkan ID Anda: 123  
Masukkan nama Anda: fauzan



Gambar 10. Node yang dapat dilewati

(Sumber : Hasil Penelitian . 2024)

```

Masukkan simpul awal (Q0, Q1, dst.): Q2
Masukkan simpul tujuan (Q0, Q1, dst.): Q1

Rute yang tersedia:
1. Rute: ['Q2', 'Q0', 'Q8', 'Q1'], Jarak: 19, Transportasi: Bus
2. Rute: ['Q2', 'Q0', 'Q6', 'Q1'], Jarak: 11, Transportasi: Jalan Kaki
3. Rute: ['Q2', 'Q9', 'Q5', 'Q0', 'Q8', 'Q1'], Jarak: 38, Transportasi: Angkot
4. Rute: ['Q2', 'Q9', 'Q5', 'Q0', 'Q6', 'Q1'], Jarak: 30, Transportasi: Shuttle
Pilih rute (1, 2, dst.): 1
Apakah Anda ingin memberi rating? (ya/tidak): ya
Berikan rating bintang 1-5: 5

Detail Perjalanan:
ID: 123
Nama: fauzan
Rute: ['Q2', 'Q0', 'Q8', 'Q1']
Jenis Kendaraan: Bus
Pemesanan Tiket: Tidak
Rating: 5 bintang
Apakah Anda ingin melanjutkan perjalanan ke node lain? (ya/tidak): ya
Masukkan simpul awal (Q0, Q1, dst.): Q2
Masukkan simpul tujuan (Q0, Q1, dst.): Q9

Rute yang tersedia:
1. Rute: ['Q2', 'Q9'], Jarak: 6, Transportasi: Jalan Kaki
2. Rute: ['Q2', 'Q0', 'Q6', 'Q1', 'Q9'], Jarak: 19, Transportasi: Shuttle

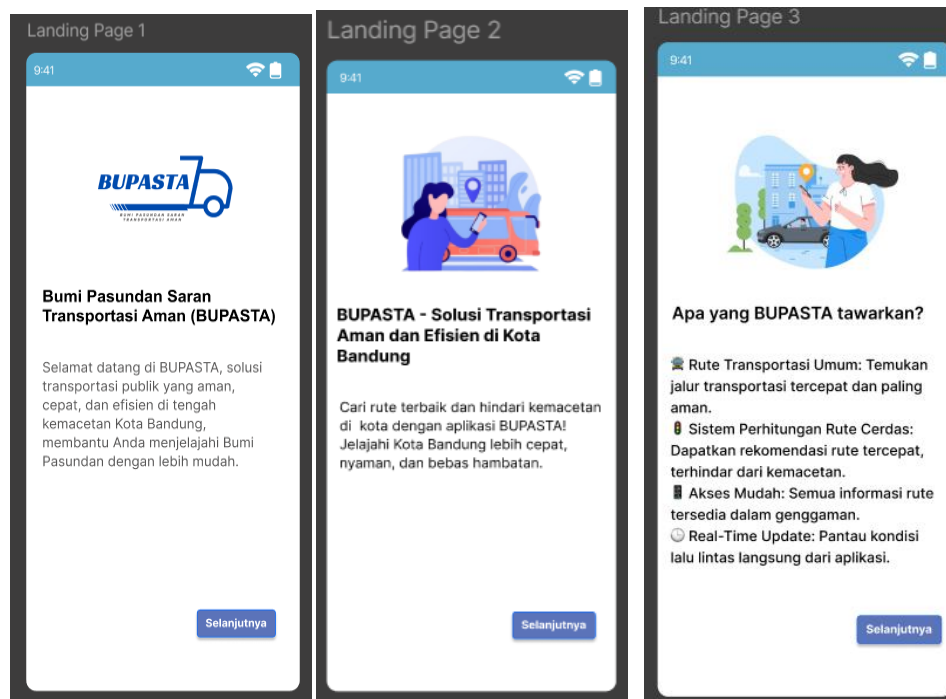
```

Gambar 11. Output implementasi Algoritma Dijkstra

(Sumber : Hasil Penelitian . 2024)

## Mockup Design Aplikasi

### 1) Landing Page

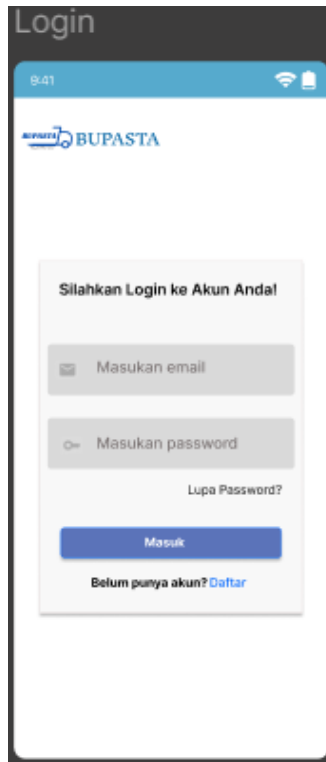


Gambar 12. Landing Page

(Sumber : Hasil Penelitian . 2024)

Desain tampilan awal ini merupakan halaman pertama yang dilihat pengguna saat memasuki aplikasi Bupasta. Halaman ini berfungsi sebagai pengantar utama yang menyambut pengguna dengan elemen visual yang menarik dan informatif. Tampilan awal ini menampilkan logo aplikasi, slogan, serta tombol navigasi utama yang memudahkan akses ke fitur-fitur inti aplikasi, seperti pencarian rute, rekomendasi transportasi.

## 2) Login

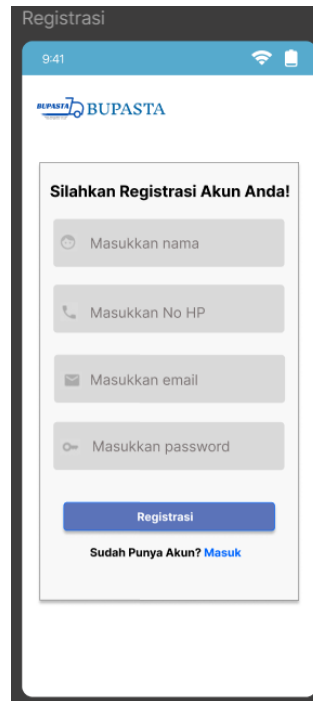


Gambar 13. Login

(Sumber : Hasil Penelitian . 2024)

Tampilan pada login pada gambar kedua berfungsi sebagai langkah pertama untuk mengakses aplikasi BUPASTA . Pengguna diminta untuk memasukkan alamat email dan kata sandi; kemudian mereka menggunakan tombol " Masuk " untuk melanjutkan. Jika pengguna tidak memiliki akun , mereka dapat membeli operasi "Daftar " . Selain itu , ada opsi yang disebut " Lupa Kata Sandi ? " yang membantu pengguna untuk login jika mereka mengetik kata sandi . Setelah berhasil login , pengguna akan diarahkan ke layar aplikasi utama untuk mulai menggunakan fitur - fitur yang tersedia .

### 3) Registrasi



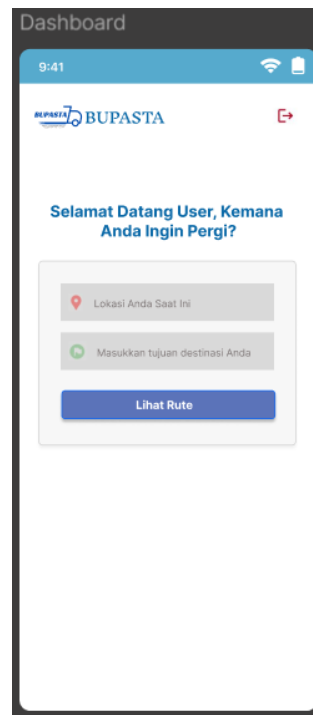
The screenshot shows a mobile application interface for registration. At the top, the status bar displays the time 9:41 and signal icons. The app header is blue with the BUPASTA logo. The main content area is white with a grey border. It contains the title 'Silahkan Registrasi Akun Anda!' followed by four input fields with icons: a person for 'Masukkan nama', a phone for 'Masukkan No HP', an envelope for 'Masukkan email', and a key for 'Masukkan password'. Below these fields is a blue button labeled 'Registrasi' and a link 'Sudah Punya Akun? Masuk'.

Gambar 14. Registrasi

(Sumber : Hasil Penelitian . 2024)

Tampilan registrasi pada gambar ini merupakan langkah awal bagi pengguna baru untuk membuat akun di aplikasi BUPASTA. Pengguna diminta mengisi informasi pribadi seperti nama, nomor HP, email, dan password, kemudian menekan tombol "Registrasi" untuk menyelesaikan proses pendaftaran. Jika pengguna sudah memiliki akun, mereka dapat memilih opsi "Masuk" untuk diarahkan ke halaman login. Setelah registrasi berhasil, pengguna dapat langsung login dan mengakses fitur-fitur aplikasi.

### 4) Dashboard

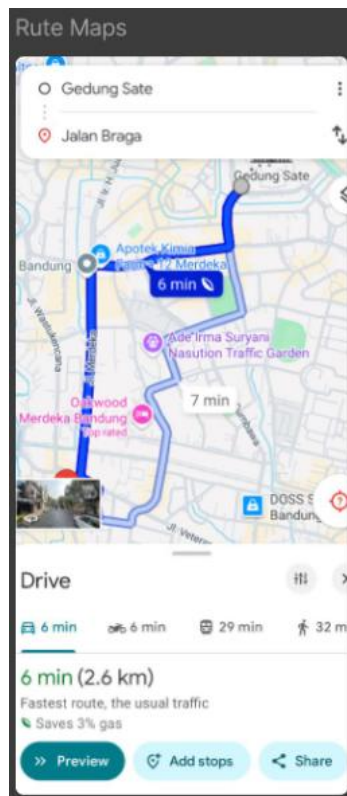


Gambar 15. Dashboard

(Sumber : Hasil Penelitian . 2024)

Tampilan Dashboard aplikasi BUPASTA ini berfungsi sebagai halaman utama yang menyambut pengguna. Desainnya menampilkan pesan sambutan 'Selamat Datang User, Kemana Anda Ingin Pergi?', disertai dengan kolom isian sederhana. Pengguna diminta untuk memasukkan lokasi saat ini dan tujuan destinasi mereka, kemudian menekan tombol 'Lihat Rute' untuk mendapatkan informasi rute yang diinginkan.

## 5) Rute

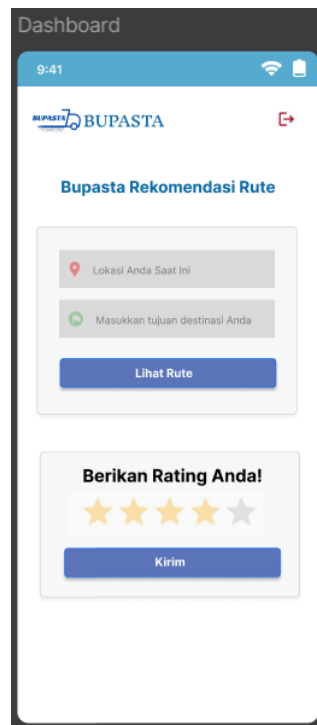


Gambar 17. Rute

(Sumber : Hasil Penelitian . 2024)

Tampilan Rute ini adalah tampilan setelah pengguna memilih lokasi awal tujuan dan lokasi destinasi yang dituju, dimana di halaman ini pengguna bisa melihat rekomendasi transportasi yang lebih efisien dan juga rute rekomendasi atau jaraknya lebih dekat.

## 6) Rating



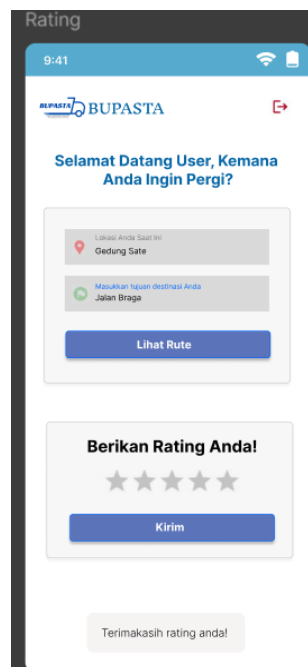
Gambar 18. Rating

(Sumber : Hasil Penelitian . 2024)

Tampilan rating ini adalah tampilan setelah pengguna mendapatkan rute dan transportasi rekomendasi dan pengguna diminta untuk memberikan umpan balik dengan memberi rating pada aplikasi guna kepuasan pengguna dalam penggunaan aplikasi Bupasta.



## 7) Sesudah Rating



Gambar 19. Setelah Rating

(Sumber : Hasil Penelitian . 2024)

Tampilan ini adalah tampilan dimana berhasil memberikan rating pada aplikasi yang nantinya sistem akan memberikan pesan “terimakasih atas rating anda”.

## Kesimpulan

Finite State Automata (FSA) pada aplikasi BUPASTA (Bumi Pasundan Saran Transportasi Aman) digunakan untuk merepresentasikan alur kerja algoritma Dijkstra dalam menentukan rute terpendek transportasi umum di Kota Bandung. Pemodelan FSA ini mencakup 16 state yang menggambarkan tahapan-tahapan dalam penggunaan aplikasi, mulai dari akses aplikasi hingga pengguna memberikan umpan balik. Penerapan FSA ini diharapkan dapat memberikan kemudahan bagi pengguna dalam memahami alur kerja aplikasi dan memperjelas logika yang digunakan dalam menentukan rute terpendek. Dengan demikian, aplikasi BUPASTA yang diimplementasikan dengan algoritma Dijkstra dan dimodelkan menggunakan FSA ini diharapkan dapat menjadi solusi yang efektif dalam mengatasi permasalahan transportasi di Kota Bandung, khususnya dalam hal kemacetan.

## REFERENSI

- Achmad, R. R., Septiana, F. F., Syamsi, N., Prakoso, B. S., & Novitasari, H. B. (2021). Penerapan Finite State Automata pada Vending Machine dalam Melakukan Transaksi Pengembalian Buku di Perpustakaan. *METIK*, 5(1). <https://doi.org/10.47002/metik.v5i1.219>
- Bunaen, M. C., Pratiwi, H., & Riti, Y. F. (2022). Penerapan algoritma Dijkstra untuk menentukan rute terpendek dari pusat kota Surabaya ke tempat bersejarah. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 4(1). <https://doi.org/10.47233/jteksis.v4i1.407>
- Cantona, A., Fauziah, & Winarsih. (2020). Implementasi algoritma Dijkstra pada pencarian rute terpendek ke museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika*, 6(1). <https://doi.org/10.26905/jtmi.v6i1.3837>
- Friticarani, A. Hayati, A. Ramdani. Hoirunisa, I. Rosdalina, G, M. (2023). Strategi Pendidikan Untuk Sukses Di Era Teknologi 5.0. *Jurnal Inovasi Pendidikan dan Teknologi Informasi*. 4 (1).F201
- Kaunang, F. J. (2019). Penerapan Konsep Finite State Automata (FSA) pada Mesin Pembuat Ice Cream Otomatis. *TelKa*, 9(2), 129-137. <https://doi.org/10.36342/teika.v9i02.2200>
- Khodijah, V., Thoyyibah, T., & Yunial, A. H. (2024). *Teori Bahasa dan Automata*. Purbalingga: CV. Eureka Media Aksara. ISBN 978-623-120-070-9. Diakses pada tanggal 17 oktorber 2024. Dari [567294-buku-ajar-teori-bahasa-dan-automata-3eb74a0f.pdf \(penerbiteureka.com\)](https://doi.org/10.36342/teika.v9i02.2200)
- Krisnandi, D., Fatiha, Z. D., Putra, I. L., Saputra, S. A., & Gata, W. (2021). *Konsep Finite State Automata pada Desain Vending Machine Alat Praktik di Sekolah Menengah Kejuruan*. *Explore IT! : Jurnal Keilmuan dan Aplikasi Teknik Informatika*. <https://doi.org/10.35891/explorit>
- Mulyana, D. (2021). Kajian Perkembangan Kota Bandung Melalui Aspek Elemen-Elemen Kota. *Jurnal Pendidikan Teknik Bangunan Dan Sipil*. 7 (1)
- Nugraha, R. A., Yanto, Mulyani, A., & Gata, W. (2020). Desain vending machine rujak buah dengan finite state automata. *Indonesian Journal on Computer and Information Technology*, 5(2), 198-207. <https://doi.org/10.31294/ijcit.v5i2.8169>
- Suprpto, D. D. A., & Fauziah. (2020). Implementasi Finite State Automata pada Mesin Abstrak DFA dan NFA Berbasis Android. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 5(1), 28. <https://doi.org/10.30998/string.v5i1.6196>.
- Wirawan, B., Robert, T., & Nugroho, S. (2017). *Sistem Penerjemah Bahasa Jawa-Aksara Jawa Berbasis Finite State Automata*. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 6(2), 127. <https://doi.org/10.22146/jnteti.v6i2.306>