

Implementasi NFA dan DFA untuk pemodelan alur layanan pada sistem informasi bengkel berbasis web

Adi Israel Willhelm Coendrad Rontgen Peday¹, Heru Sutejo^{2*}

¹Teknik Informatika Universitas Sepuluh Nopember Papua, Kota Jayapura

²Sistem Informasi Universitas Sepuluh Nopember Papua, Kota Jayapura

*Penulis korespondensi: heru.sutejo01@gmail.com;

Riwayat artikel:

Diterima: 17 Januari 2026

Direvisi: 26 Januari 2026

Disetujui: 26 Januari 2026

Diterbitkan: 27 Januari 2026

Kata kunci:

Automata

Deterministic finite automata

Nondeterministic finite automata

Sistem informasi

Website

ABSTRAK

Permasalahan utama pada manajemen operasional bengkel konvensional adalah ketidakakuratan data akibat pencatatan manual. Penelitian ini bertujuan merancang model alur layanan pada sistem informasi bengkel di 21Motoshop Jayapura menggunakan pendekatan *Nondeterministic Finite Automata* dan *Deterministic Finite Automata*. Metode yang digunakan adalah formalisasi logika bisnis ke dalam *Finite State Automata* 5-tuple, dimana model awal dikonversi menjadi deterministik untuk menghilangkan ambiguitas pada tahap pengambilan keputusan estimasi biaya. Implementasi sistem dilakukan menggunakan teknologi berbasis web dengan Node.js dan Vue.js. Hasil penelitian menunjukkan bahwa penerapan model automata mampu memvalidasi setiap transisi input secara ketat, menjamin keteraturan alur kerja dari pendaftaran hingga pembayaran, serta menyediakan transparansi status layanan secara waktu nyata. Sistem ini terbukti efektif meminimalisir kesalahan logika prosedur yang sering terjadi pada sistem manual.

DOI:

<https://doi.org/10.31315/jiitecs.vxxix.xxx>

This is an open access article under the [CC-BY](https://creativecommons.org/licenses/by/4.0/) license.



1. PENDAHULUAN

Di era digital saat ini, efisiensi operasional bisnis jasa seperti bengkel sangat bergantung pada manajemen data yang akurat dan terstruktur[1]. serta Audrilia & Budiman (2020) mencatat bahwa banyak bengkel konvensional masih mengandalkan pencatatan manual yang rentan terhadap ketidakakuratan data dan kesulitan pelacakan riwayat servis[2]. Rizza dkk (2021) menambahkan bahwa metode manual ini sering menyebabkan kesalahan penulisan keluhan pelanggan dan kekeliruan perhitungan biaya[3]. Sebagai solusi, Nurlaila & Mulyono (2023) membuktikan bahwa sistem informasi manajemen berbasis *web* mampu mengintegrasikan data transaksi secara *real-time* dan meminimalisir kesalahan manusia, meskipun tantangan utama tetap pada pemodelan alur layanan yang kompleks agar logika sistem berjalan *valid*[4].

Untuk memodelkan alur layanan yang kompleks tersebut secara formal, teori automata menawarkan pendekatan yang presisi. Farhan dkk (2021) menjelaskan bahwa *Finite State Automata* (FSA) adalah model matematika yang efektif untuk memverifikasi validitas *input* dalam sebuah sistem[5]. Hal ini didukung oleh Chandra dkk. (2024) dan Nabhan dkk. (2023) yang menekankan bahwa FSA memungkinkan perancang sistem memetakan setiap tahapan layanan ke dalam model matematis, sehingga logika program dapat diuji

kebenarannya sebelum implementasi kode dilakukan[6][7]. Akbar (2025) juga memperlihatkan bahwa pendekatan *Finite State Machine* sangat efektif untuk menggambarkan karakteristik keadaan (*state*) dalam aplikasi interaktif[8].

Dalam penerapannya, penggunaan *Nondeterministic Finite Automata* (NFA) dan *Deterministic Finite Automata* (DFA) memiliki peran yang saling melengkapi. Richardson dkk. (2019) dan Arifudin dkk. (2023) menyatakan bahwa NFA memberikan fleksibilitas tinggi pada tahap desain awal untuk menangkap berbagai kemungkinan keputusan dalam alur bisnis[9][10]. Namun, untuk kebutuhan eksekusi sistem yang pasti, Robyanto dkk. (2021) menegaskan perlunya konversi ke DFA untuk menjamin determinisme, di mana setiap status hanya memiliki satu transisi pasti untuk setiap input[11]. Keberhasilan metode pemodelan alur kerja (*workflow*) ini juga dikonfirmasi oleh Proboyekti dkk. (2022) yang menunjukkan bahwa *Business Process Modeling* berbasis alur formal menghasilkan aplikasi yang lebih stabil dan mudah dipelihara[12].

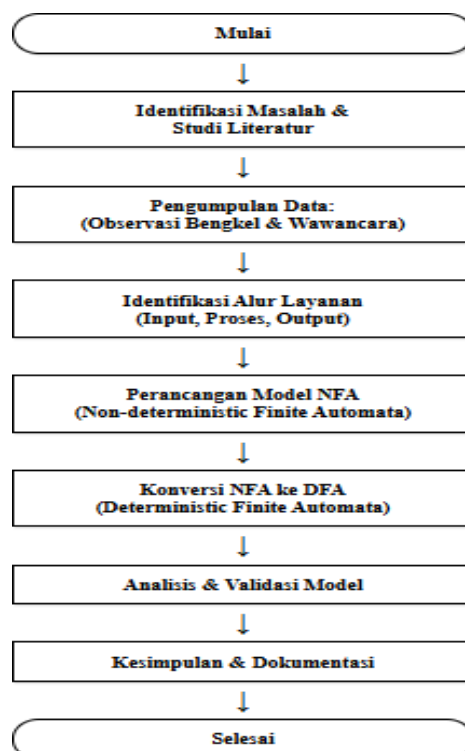
Berdasarkan tinjauan terhadap permasalahan manajemen bengkel dan keandalan teori automata dalam pemodelan sistem, penelitian ini bertujuan untuk merancang model alur layanan pada sistem informasi bengkel di 21Motoshop Jayapura menggunakan pendekatan NFA dan DFA. Penelitian ini berfokus pada analisis alur layanan yang ada, perancangan model NFA untuk merepresentasikan fleksibilitas proses, serta konversi ke model DFA untuk kebutuhan implementasi sistem yang deterministik. Melalui pendekatan ini, diharapkan tercipta sebuah rancangan logika sistem yang tidak hanya mendigitalkan proses manual, tetapi juga memiliki landasan alur kerja yang terstruktur, *valid*, dan terverifikasi secara matematis.

2. TEORI DAN METODE

Metode merupakan pendukung penelitian baik dari segi pengumpulan data, tempat penelitian, jenis dan sumber penelitian, tahapan penelitian, metode pengembangan, dan lainnya yang diperlukan dalam metode. Penelitian ini menggunakan pendekatan formal dengan menerapkan model *Finite State Automata* (FSA) untuk merancang alur layanan pada sistem informasi bengkel di 21Motoshop Jayapura.

2.1. Tahap Penelitian

Penelitian ini dilaksanakan melalui serangkaian tahapan sistematis yang dimulai dari identifikasi masalah, pengumpulan data, perancangan model automata, hingga konversi model NFA ke DFA. Alur tahapan penelitian diilustrasikan pada [Gambar 1](#).



Gambar 1. Alur tahapan penelitian

2.2. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan informasi mengenai alur layanan yang sedang berjalan serta landasan teori yang relevan. Observasi dilakukan secara langsung di Bengkel 21Motoshop Jayapura untuk mengamati proses bisnis mulai dari pelanggan datang, pendaftaran, servis, hingga pembayaran. Wawancara dengan staf bengkel dilakukan untuk mengonfirmasi aturan-aturan layanan dan keputusan bisnis yang diambil pada setiap tahapan. Studi literatur digunakan untuk memperkuat landasan teori mengenai implementasi automata. Wijaya (2020) menunjukkan bahwa penggunaan model komputasi DFA dapat meningkatkan keamanan dan keteraturan logika proses melalui partisi yang jelas antar tahapan[13]. Konsep ini relevan untuk diterapkan dalam memisahkan status layanan bengkel agar tidak terjadi tumpang tindih logika. Selain itu, Suprpto & Fauziah (2020) menekankan bahwa mesin abstrak FSA, baik NFA maupun DFA, harus dirancang dengan tuple yang lengkap (Q, Σ, δ, S, F) agar dapat diimplementasikan ke dalam logika pemrograman berbasis aplikasi, seperti Android atau Web [14].

2.3. Perancangan Model Automata

Data alur layanan yang telah dikumpulkan kemudian diterjemahkan ke dalam bentuk formal menggunakan teori automata.

a) Identifikasi Tuple FSA

Sesuai dengan definisi formal yang dijelaskan oleh Suprpto & Fauziah (2020), model FSA didefinisikan sebagai 5-tuple $M = (Q, \Sigma, \delta, S, F)$. Dalam konteks bengkel ini:

- Q (Himpunan *State*): Representasi status layanan (misal: Menunggu Antrean, Sedang Dikerjakan, Selesai).
- Σ (Himpunan *Input*): Aksi yang memicu perubahan status (misal: 'Daftar', 'Mulai Servis', 'Bayar').
- δ (Fungsi Transisi): Aturan perpindahan antar status.
- S (State Awal): Status awal saat pelanggan datang.
- F (State Akhir): Status saat layanan selesai sepenuhnya.

b) Pemodelan NFA dan Konversi ke DFA

Pada tahap awal, alur dimodelkan sebagai NFA untuk menangkap semua kemungkinan keputusan yang mungkin terjadi, seperti yang diterapkan oleh Aryanto (2023) dalam mendiagnosa gejala penyakit yang kompleks[15]. Namun, karena sistem informasi membutuhkan kepastian alur, model NFA tersebut dikonversi menjadi DFA. Wijaya (2020) menegaskan bahwa DFA memberikan kepastian transisi (deterministik) yang sangat penting untuk validasi logika sistem, memastikan tidak ada ambiguitas dalam proses eksekusi layanan[13].

Tabel 1. Fungsi transisi model automata

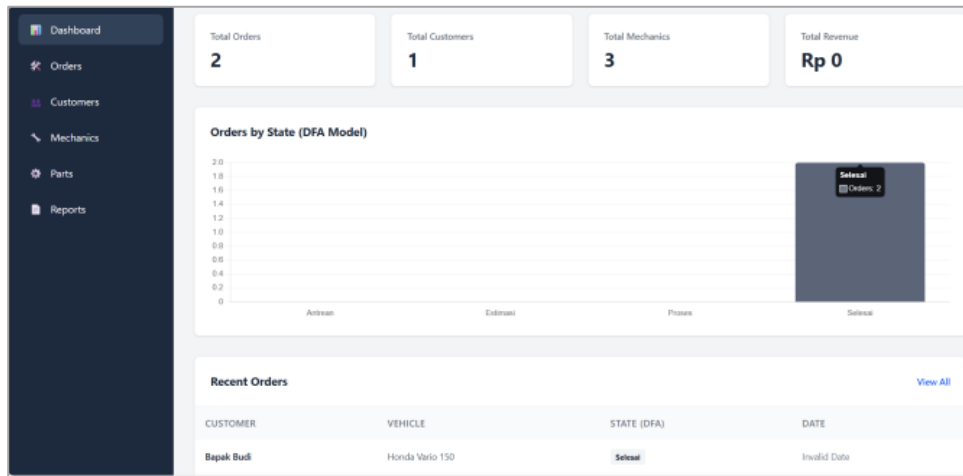
State (<i>q</i>)	Input <i>a</i> (daftar)	Input <i>b</i> (cek)	Input <i>c</i> (setuju)	Input <i>d</i> (bayar)
q ₀ (awal)	q ₁	-	-	-
q ₁ (antrian)	-	q ₂	-	-
q ₂ (estimasi)	-	-	{q ₃ , q ₄ }	-
q ₃ (servis ringan)	-	-	-	q ₅
q ₄ (servis berat)	-	-	-	q ₅
q ₅ (selesai)	-	-	-	-

Konversi ini dilakukan dengan membuat tabel transisi baru yang memetakan setiap kombinasi state dan input ke satu state tujuan yang unik, sehingga siap diimplementasikan ke dalam kode program sistem informasi bengkel.

3. HASIL

Sistem informasi bengkel dikembangkan menggunakan arsitektur *client-server*. Sisi *backend* dibangun menggunakan Node.js dan Express.js untuk menangani logika DFA, sedangkan sisi *frontend* menggunakan Vue.js untuk antarmuka pengguna yang responsif. Basis data MongoDB digunakan untuk menyimpan status (*state*) transaksi secara persisten. Berikut adalah hasil implementasi antarmuka berdasarkan pemodelan DFA

yang telah dirancang. Pada **Gambar 2** merupakan halaman dashboard admin dan monitoring *state*. Halaman ini merupakan pusat kontrol bagi admin untuk memantau seluruh antrian kendaraan. Sistem memvisualisasikan distribusi status servis dalam bentuk grafik dan tabel. Sesuai dengan definisi himpunan *state* Q , setiap kendaraan diberi label status secara *real-time* (Antrean, Estimasi, Proses, atau Selesai).



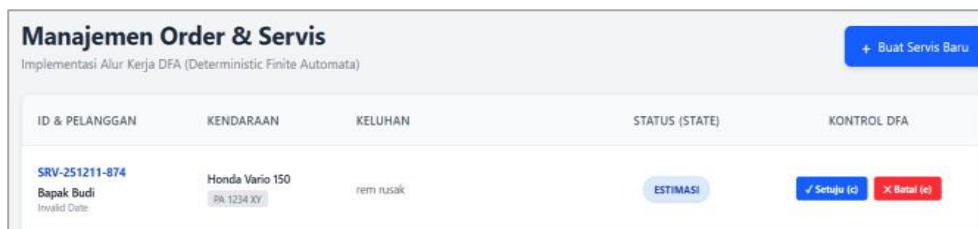
Gambar 2. Tampilan dashboard admin dengan visualisasi status DFA

Sesuai dengan fungsi transisi $\delta(q_0, a) = q_1$, proses pendaftaran servis dilakukan melalui formulir *input* data pelanggan dan keluhan. Ketika admin menekan tombol simpan, sistem secara otomatis menginisialisasi status servis menjadi Antrean (q_1). Sistem memvalidasi bahwa input a hanya dapat diterima ketika sistem berada pada *initial state*. Tampilan antarmuka yaitu pada **Gambar 3**.



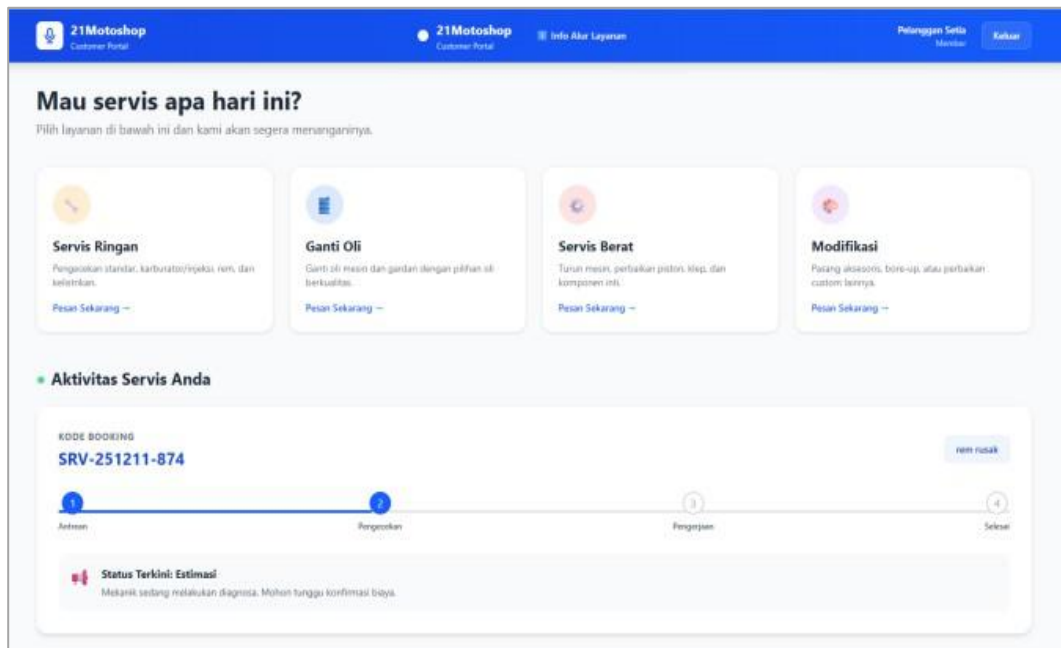
Gambar 3. Formulir pendaftaran servis sebagai input transisi awal

Salah satu penerapan krusial DFA terdapat pada tahap Estimasi (q_2). Pada tahap ini, sistem membatasi aksi admin hanya pada dua kemungkinan *input* deterministik: "Setuju" (c) yang membawa transisi ke Proses Servis (q_{34}), atau "Batal" (e) yang langsung menuju Selesai (q_5). Implementasi ini mencegah ambiguitas alur yang sering terjadi pada pencatatan manual, tampilannya pada **Gambar 4** yang merupakan kontrol logika deterministik (*state* estimasi).



Gambar 4. Implementasi control percabangan deterministik pada tahap estimasi

Transparansi alur layanan disajikan kepada pelanggan melalui antarmuka *tracking*. Pelanggan dapat melihat posisi kendaraan mereka dalam diagram alur. Hal ini membuktikan bahwa model DFA $(Q, \Sigma, \delta, S, F)$ telah berhasil diterjemahkan menjadi indikator visual yang dapat dipahami pengguna awam.

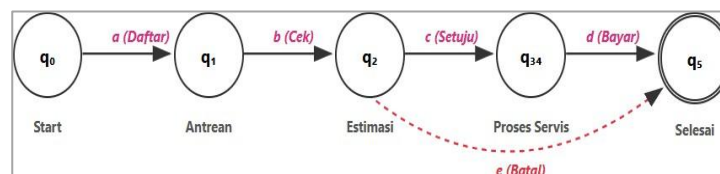


Gambar 5. Tampilan pelacakan status servis berbasis alur untuk pelanggan

4. PEMBAHASAN

Pembahasan difokuskan pada analisis transformasi alur layanan bengkel manual menjadi model formal automata. Berdasarkan observasi lapangan di 21Motoshop, alur layanan memiliki karakteristik keputusan yang dinamis, terutama pada tahap estimasi biaya di mana pelanggan dapat menyetujui atau membatalkan layanan. Pada tahap awal perancangan menggunakan NFA, ditemukan titik ambiguitas pada *state* Estimasi (q_2). *Input* "Setuju" dari pelanggan secara teori dapat memicu dua transisi berbeda: menuju *state* Servis Ringan (q_3) atau Servis Berat (q_4). Kondisi percabangan ini valid dalam pemodelan NFA untuk menggambarkan variasi proses bisnis, sebagaimana Aryanto (2023) menerapkan NFA untuk menangani variasi gejala penyakit yang kompleks. Namun, dalam konteks pengembangan perangkat lunak, ambiguitas ini dapat menyebabkan ketidakpastian logika di mana sistem tidak dapat memutuskan secara otomatis status mana yang harus dituju hanya dengan satu *input* "Setuju". Untuk mengatasi masalah tersebut, dilakukan konversi model menjadi DFA dengan dua strategi utama:

- 1) Penggabungan *State* (*State Merging*): *State* Servis Ringan (q_3) dan Servis Berat (q_4) digabungkan menjadi satu *state* komposit baru, yaitu Proses Servis (q_{34}). Strategi ini menyederhanakan alur servis menjadi linear tanpa mengurangi makna proses, sehingga *input* "Setuju" kini memiliki satu tujuan transisi yang pasti.
- 2) Penanganan Alur Pembatalan: Ditambahkan transisi eksplisit untuk *input* "Batal" (e) dari *state* Estimasi (q_2) yang langsung menuju ke *state* Selesai (q_5). Penambahan ini memastikan bahwa sistem dapat menangani kondisi penolakan layanan secara deterministik.



Gambar 6. Diagram transisi DFA hasil koneversi

Wijaya (2020) menegaskan bahwa determinisme yang dihasilkan dari model DFA menjamin keamanan dan keteraturan eksekusi algoritma. Dengan model DFA hasil revisi (Gambar 3), setiap pasangan *state* dan *input* kini memiliki tepat satu transisi yang terdefinisi. Model logika formal ini kemudian diimplementasikan menggunakan teknologi Node.js dan Express.js di sisi server serta MongoDB sebagai basis data, yang dipadukan dengan Vue.js dan Tailwind CSS di sisi client untuk menjamin responsivitas sistem terhadap

transisi status yang cepat dan akurat. Hal ini memberikan landasan yang kokoh bagi sistem informasi untuk memvalidasi setiap tindakan pengguna dan memastikan integritas data transaksi.

5. KESIMPULAN

Simpulan penelitian ini berhasil merancang dan mengimplementasikan sistem informasi manajemen bengkel di 21Motoshop Jayapura menggunakan pendekatan *Deterministic Finite Automata* (DFA) yang terbukti efektif mengatasi permasalahan ketidakakuratan data pada sistem manual. Transformasi model dari *Non-deterministic Finite Automata* (NFA) ke DFA melalui penggabungan *state* servis mampu menghilangkan ambiguitas logika alur layanan, sehingga menghasilkan sistem yang deterministik dan *valid* dalam memverifikasi setiap transisi *input* (Q, Σ, δ, S, F) mulai dari pendaftaran hingga penyelesaian. Implementasi berbasis *web* menggunakan Node.js dan Vue.js menunjukkan bahwa pemodelan formal ini tidak hanya mendigitalkan proses bisnis, tetapi juga menjamin integritas data dan transparansi status servis secara *real-time*. Sebagai pengembangan selanjutnya, sistem dapat diintegrasikan dengan fitur notifikasi otomatis berbasis *WhatsApp Gateway* yang memicu pesan setiap kali terjadi perpindahan *state*, serta penambahan algoritma penjadwalan cerdas untuk mengoptimalkan alokasi mekanik berdasarkan beban kerja antrean.

REFERENSI

- [1] L. Siallagan and M. Ropianto, "PERANCANGAN DAN PENGEMBANGAN SISTEM INFORMASI MANAJEMEN BENGKEL BERBASIS WEB DI BENGKEL CV.TIO MEGAH JAYA MOBIL."
- [2] M. Audrilia and A. Budiman, "Perancangan Sistem Informasi Manajemen Bengkel Berbasis Web (Studi Kasus : Bengkel Anugrah)," *Jurnal Madani : Ilmu Pengetahuan, Teknologi, dan Humaniora*, vol. 3, no. 1, pp. 1–12, Mar. 2020, doi: 10.33753/madani.v3i1.78.
- [3] P. Studi Manajemen Informatika, I. Rizza, Z. Ritonga, D. Naingolan, and S. Zuhri Harahap, "SITEM INFORMASI PENJUALAN PRODUK PERTANIAN PADA TOKO TRIPUTRI AEK NABARA BERBASIS WEB," 2021.
- [4] D. Nurlaila and H. Mulyono, "Sistem Informasi Manajemen Bengkel Berbasis Web pada Bengkel Ikhsan Jaya Motor," 2023.
- [5] S. Farhan, I. Fitri, F. Infoman, P. Studi Informatika, J. Sawo Manila Pasar Minggu, and J. Selatan, "Implementasi Transisi Finite State Automata dengan Aplikasi Mesin Abstrak DFA dan NFA Berbasis Android."
- [6] E. Avilia Chandra and M. Ikhsan, "IMPLEMENTASI FINITE STATE AUTOMATA PADA APLIKASI PEMBELAJARAN MENGASAH OTAK ANAK MENGGUNAKAN DFA DAN NFA BERBASIS ANDROID," 2024. [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [7] J. Sistem Komputer dan Kecerdasan Buatan et al., "Simulator Mesin Deterministic Finite Automata (DFA) Berdasarkan Diagram Transisi Menggunakan Python."
- [8] A. Fawwaz Akbar, A. Jl Riau Ujung No, A. Hitam Kec Payung Sekaki, and K. Pekanbaru, "Pengembangan Aplikasi Permainan Pisah Menggunakan Metode Finite State Machine untuk Anak-Anak."
- [9] B. Richardson, K. Hendy, V. Andiyani, and W. Philips, "Penerapan Konsep Non- Deterministic Finite Automata (NFA) pada Aplikasi Simulasi Mesin Kopi Vending," *Maret*, vol. 4, no. 1, pp. 2622–4615, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika>
- [10] M. B. Arifudin, Y. Mulyani, and W. E. Sulistiono, "SIMULASI SMART VENDING MACHINE MINUMAN KOPI DENGAN NFA BERBASIS ANDROID," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 1, Jan. 2023, doi: 10.23960/jitet.v11i1.2803.
- [11] D. Robyanto, A. Priyatna, E. H. Hermaliani, W. Gata, and E. Heni Hermaliani, "Deny Robyanto Implementasi Deterministic Finite Automata (DFA) Implementasi Deterministic Finite Automata (DFA) Pada Perancangan Aplikasi Perhitungan Uang Harian Perjalanan Dinas Kementerian."
- [12] U. Proboyekti, C. Septa, and B. Susanto, "Implementasi Business Process Modeling dalam Workflow Aplikasi Monitoring Electronic Data Capturer," *Jurnal Terapan Teknologi Informasi*, vol. 6, no. 1, pp. 75–82, Apr. 2022, doi: 10.21460/jutei.2022.61.208.
- [13] A. Wijaya, "Modifikasi Algoritma Kriptografi Klasik dengan Implementasi Deterministic Finite Automata melalui Partisi Pesan Asli berdasarkan Kriteria Pesan Bagian," *Journal of Science and Applicative Technology*, vol. 4, no. 2, p. 133, Dec. 2020, doi: 10.35472/jsat.v4i2.346.

- [14] D. Dandy Aryarajendra Suprpto, "STRING (Satuan Tulisan Riset dan Inovasi Teknologi) IMPLEMENTASI FINITE STATE AUTOMATA PADA MESIN ABSTRAK DFA DAN NFA BERBASIS ANDROID."
- [15] R. P. Aryanto, "Penerapan Konsep Non- deterministic Finite Automata Dalam Diagnosa Penyakit ISPA."