



Penerapan Finite State Automata Pada Transaksi Via Mobile Banking (Aplikasi BRImo)

Januar C. Talahaturuson¹, Magdalena A. Ineke Pakereng²

^{1,2}Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia
e-mail: 672015801@student.uksw.edu¹, ineke.pakereng@uksw.edu²

Abstract

The purpose of this study is the application of Finite State Automata on the BRImo mobile banking application. The result of this study is a simulation of the mobile banking application on the BRImo application related to the application of Finite State Automata which is used in the transaction process via the BRImo mobile banking application. The results show that Finite State Automata can be used as the basic logic to process the BRImo application mobile banking transaction which divides each system into state forms.

Keywords : *Finite State Automata, Brimo Application Mobile Banking Transactions*

Abstrak

Tujuan dari penelitian ini adalah penerapan Finite State Automata pada aplikasi mobile banking BRImo. Hasil dari penelitian ini adalah simulasi aplikasi mobile banking pada aplikasi BRImo terkait penerapan Finite State Automata yang digunakan dalam proses transaksi via mobile banking aplikasi BRImo. Hasil penelitian menunjukkan bahwa Finite State Automata dapat digunakan sebagai logika dasar untuk melakukan proses transaksi i aplikasi BRImo yang membagi tiap sistem ke dalam bentuk state.

Kata kunci: *Finite State Automata, Transaksi Mobile Banking Aplikasi BRImo*

1. PENDAHULUAN

Kemajuan ilmu teknologi membawa perubahan besar dalam kehidupan sehari-hari bagi manusia yang mempunyai dampak positif maupun negatif. Penemuan baru dibutuhkan untuk keberlangsungan hidup saat ini, salah satunya adalah ilmu komputer yang memiliki dua komponen utama, yaitu: model gagasan awal atau dasar terkait komputasi dan teknik rekayasa untuk perancangan suatu sistem komputasi yang meliputi perangkat lunak maupun perangkat keras, terkhusus penerapan rancangan dari teori [1]. Bagian dari keilmuan dari komputer yang pertama merupakan teori bahasa dan automata.

Teori Bahasa dan *Automata* sudah banyak diterapkan dalam hal kompilator, pembuatan bahasa pemrograman, dan perancangan digital [2]. Penggunaan *Automata* dibutuhkan untuk pengembangan ilmu komputer lebih lanjut, baik pada perangkat keras (*hardware*), maupun perangkat lunak (*software*) [3]. *Automata* adalah suatu bentuk yang memiliki fungsi- fungsi dari komputer digital. Menerima *input*, menghasilkan *output*, bisa memiliki penyimpanan sementara, dan mampu membuat keputusan dalam mentransformasikan *input* ke *output* yang selanjutnya membuat keputusan yang mengindikasikan apakah input itu diterima atau tidak. *Automata* merupakan suatu sistem yang terdiri atas sejumlah *state* berhingga setiap *state* menyatakan informasi mengenai *input* sebelumnya, dan dapat pula dianggap sebagai *memory* mesin. *Input* pada mesin *automata* dianggap sebagai bahasa yang harus dikenali oleh mesin (bukan mesin fisik). Selanjutnya mesin

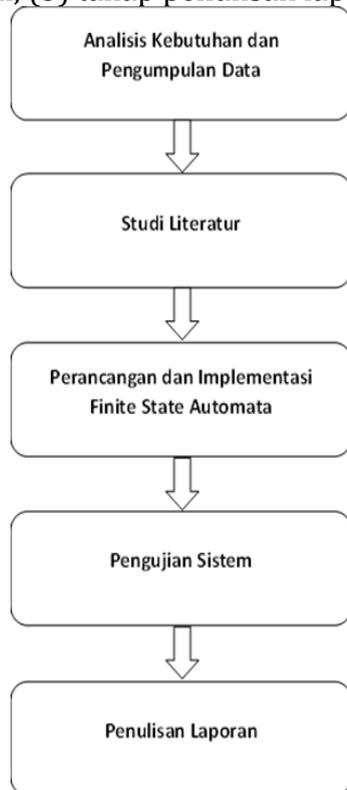


automata membuat keputusan yang dapat mengenali (*recognize*), menerima (*accept*), atau membangkitkan (*generate*) sebuah kalimat dalam bahasa tertentu. Teori *automata* sangat erat kaitannya dengan mesin-mesin abstrak [1,4,5].

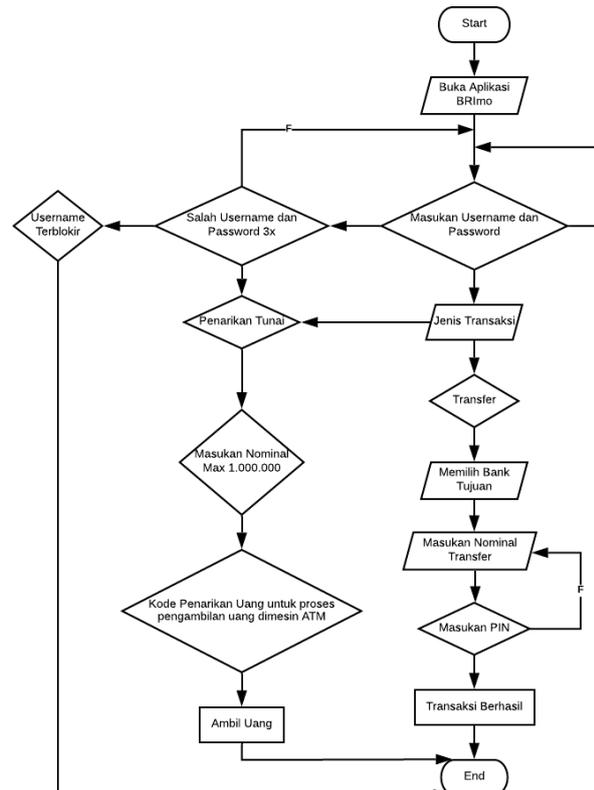
Penelitian terkait *Finite State Automata* untuk transaksi via *Mobile Banking* belum banyak dilakukan khususnya di Indonesia, sehingga perlunya penelitian ini untuk dilakukan. Fenomena penggunaan *Mobile Banking* saat ini dapat mempermudah nasabah sehingga nasabah merasakan kenyamanan dengan adanya layanan tersebut, dengan adanya *Mobile Banking* dapat memungkinkan nasabah mendapatkan informasi serta menyelesaikan berbagai urusan perbankan secara cepat, mudah dan tanpa batas tempat dan waktu [6]. Tujuan dari penelitian ini adalah untuk menerapkan *Finite State Automata* pada aplikasi *Mobile Banking* (Aplikasi BRImo).

2. METODOLOGI PENELITIAN

Tahapan yang digunakan dalam penelitian penerapan *Finite State Automata* pada aplikasi *Mobile Banking* (BRImo) dibagi ke dalam 5 (lima) tahapan yaitu: (1) tahap analisis kebutuhan dan pengumpulan data, (2) tahap studi literatur, (3) tahap perancangan dan implementasi *Finite State Automata*, (4) tahap pengujian sistem, (5) tahap penulisan laporan.



Gambar 1. Tahapan Penelitian

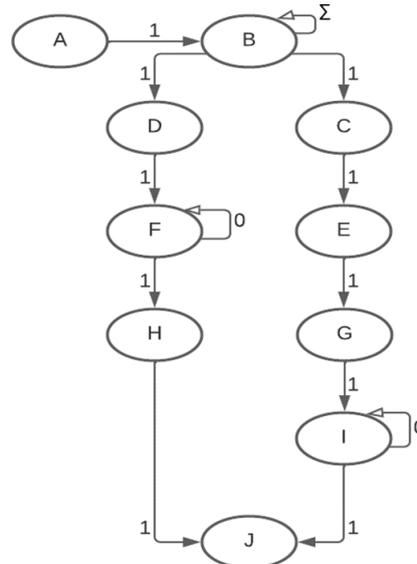


Gambar 2. Proses Perancangan Transaksi *Mobile Banking*

Penjelasan tahapan Penelitian pada Gambar 1, adalah sebagai berikut: Pada tahap analisis Kebutuhan dan Pengumpulan Data, dilakukan analisis tentang sistem pada proses transaksi *Mobile Banking* Aplikasi BRImo. Pada Tahap Studi Literatur dilakukan dengan mengumpulkan sejumlah data dari artikel jurnal terakreditasi terkait penelitian yang dilakukan dan buku; Pada Tahap Perancangan dan Implementasi *Finite State Automata*, dilakukan dengan merancang *Finite State Automata* menggunakan *Non Deterministic Finite Automata*, Pada Tahap Pengujian Sistem dilakukan dengan pengujian penerapan *Finite State Automata* pada proses transaksi via *Mobile Banking* Aplikasi BRImo menggunakan *Non Deterministic Finite Automata* yang telah dirancang. Pengujian dilakukan dengan evaluasi terhadap keseluruhan perancangan *Finite State Automata* pada proses transaksi via *Mobile Banking* BRImo menggunakan *Non Deterministic Finite Automata* yang telah dibuat, dan pada Tahap terakhir adalah tahapan Penulisan Laporan dalam tahap ini dilakukan penulisan laporan penelitian yang telah dilakukan. Gambar 2 menunjukkan rancangan Proses transaksi *mobile banking*.

3. HASIL DAN PEMBAHASAN

State Diagram Sistem Transaksi via *Mobile Banking* (Aplikasi BRImo) ditunjukkan pada Gambar 3.



Gambar 3. State Diagram Sistem Transaksi via Mobile Banking (Aplikasi BRImo)

State diagram pada Gambar 3 dinyatakan dalam 13 tupel, yaitu:

$$Q = \{A, B, C, D, E, F, G, H, I, J\}$$

$$\Sigma = \{\epsilon, 0, 1\} S = \{A\} F = \{L\}$$

$$\Delta = \{(C, \epsilon)C\}, \{(C, 1)D\}, \{(D, 1)E\}, \{(D, 1)F\}, \{(E, 1)G\}, \{(F, 1)H\}, \{(G, 1)I\}, \{(H, 1)J\}, \{(I, 1)K\}, \{(J, 1)L\}, \{(K, 1)L\}$$



Keterangan abjad, himpunan simbol *input* untuk menyatakan apa yang dilakukan oleh nasabah: (ϵ) *string* kosong, (1) melanjutkan proses transaksi, (0) melakukan perulangan pada proses transaksi/membatalkan transaksi. Sedangkan untuk keterangan *state* yang menyatakan proses transaksi : (A) memasukkan ID dan PIN, (B) memilih jenis transaksi, (C) memilih jenis transaksi transfer, (D) memilih jenis transaksi tarik tunai, (E) daftar kode bank, (F) masukkan nominal tarik tunai, (G) masukkan kode bank dan rekening tujuan, (H) mengambil uang, (I) masukkan nominal transfer, (J) proses transaksi berhasil/selesai. Relasi transisi dapat dipetakan dalam tabel, menjadi seperti pada Tabel 1.

Tabel 1. Tabel Relasi Transisi

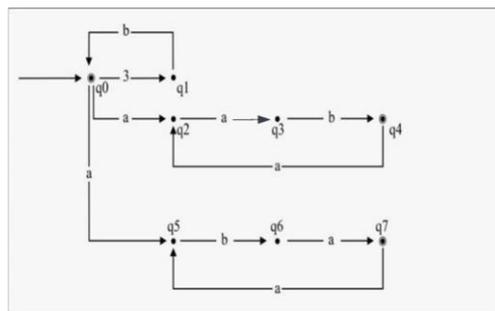
Fungsi Transisi	Simbol Input	Input yang dibaca	State Tujuan
Δ	ϵ	0	1
A	A	\emptyset	B
B	\emptyset	\emptyset	C,D
C	\emptyset	\emptyset	E
D	\emptyset	\emptyset	F
E	\emptyset	\emptyset	G
F	\emptyset	F	H
G	\emptyset	\emptyset	I
H	\emptyset	\emptyset	J
I	\emptyset	\emptyset	J
J	\emptyset	\emptyset	\emptyset

Tabel 1 menjelaskan tentang perpindahan *state* apabila ada *input* yang masuk. Misalnya pada *state* E mendapat *input* 1 maka akan berpindah ke *state* G, sesuai Gambar 3. Begitu seterusnya untuk cara membaca transisi dari *state* yang lain. Perpindahan *state* akan bernilai \emptyset (bahasa kosong) apabila tiap *input* yang masuk tidak sesuai. Misalnya *state* B apabila mendapat *input* ϵ (*string* kosong), maka akan bernilai \emptyset (bahasa kosong), begitu juga pada contoh *state* F mendapat *input* 0 maka akan melakukan perulangan sampai data yang diinput telah sesuai baru akan mendapat *input* 1 agar berpindah ke *state* H. Begitu juga seterusnya untuk *state* yang lain, yang mendapat *input* tidak sesuai.

Algoritma transaksi nasabah pada mobile banking BRImo adalah sebagai berikut:

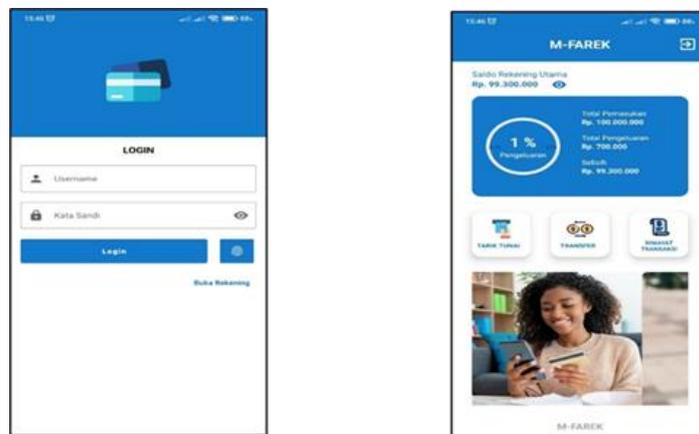
- a) Memasukkan ID dan PIN akun BRImo
- b) Memilih jenis transaksi
- c) Jika nasabah ingin melakukan transfer uang, maka nasabah memilih fitur transfer
- d) Selanjutnya nasabah diminta untuk memilih bank tujuan sebelum melakukan transfer ke rekening tujuan
- e) Masukkan nominal transfer sesuai kebutuhan nasabah
- f) Nasabah diminta memasukkan kode PIN yang telah dibuat ketika di awal pendaftaran di aplikasi BRImo

- g) Transaksi Berhasil dengan dikirimnya struk digital pada halaman aplikasi BRImo
- h) Jika nasabah ingin melakukan transaksi tarik tunai, maka nasabah memilih menu tarik tunai
- i) Masukkan nominal penarikan sesuai kebutuhan nasabah
- j) Nasabah diminta untuk mengkonfirmasi transaksi yang telah dilakukan
- k) Nasabah diminta memasukkan kode PIN yang telah dibuat di awal pendaftaran aplikasi BRImo
- l) Tampilan di aplikasi BRImo akan mengarahkan nasabah ke kode penarikan uang
- m) Kode tersebut digunakan untuk proses pengambilan uang di mesin ATM pada fitur penarikan tanpa kartu
- n) Nasabah diminta memilih Bahasa, kemudian memasukkan kode, setelah itu diminta untuk memasukkan nomor telepon yang telah terdaftar pada akun BRImo nasabah.
- o) Ambil uang, transaksi selesai.



Gambar 4. Rangkaian NFA (*Non Deterministic Finite Automata*)[15]

Rangkaian pada Gambar 4 tergolong dalam NFA (*Non Deterministic Finite Automata*) karena beberapa transisi yang berasal dari satu kondisi yaitu kondisi q_0 memiliki input yang sama yaitu 'a'. Rangkaian tersebut akan menerima *string* ab, aab, aabaab, aba, dan abaaba, tetapi tidak akan menerima *string* abb dan aabb.



Gambar 5. Tampilan Awal Penginputan Login dan Menu Dashboard.

Gambar 5 adalah tampilan awal pada aplikasi simulasi *mobile banking*. Dimana sebelum masuk ke dalam *menu*, *user* diminta agar memasukkan *username* dan *password* yang sudah terdaftar, dan ketika sudah melakukan *login*, akan muncul tampilan menu.

Kode Program 1 Perintah untuk Proses *penginputan* dan Validasi

```

69 private void formValidation() {
70     String username = binding.username.getText().toString().trim();
71     String password = binding.password.getText().toString().trim();
72
73     // di halaman login, ada 2 kolom, yaitu username, dan password, sistem harus dapat memvalidasi apakah kolom dalam keadaan kosong atau terisi
74     if (username.isEmpty()) {
75         Toast.makeText(LoginActivity.this, "Username tidak boleh kosong!", Toast.LENGTH_SHORT).show();
76         return;
77     } else if (password.isEmpty()) {
78         Toast.makeText(LoginActivity.this, "Kata Sandi tidak boleh kosong!", Toast.LENGTH_SHORT).show();
79         return;
80     }
81
82     // jika klik login dan kolom inputan tidak kosong, maka tampilkan loading bar ketika klik login
83     binding.progressBar.setVisibility(View.VISIBLE);
84
85     // mula mula cek dulu username yang dimasukkan user, apakah sudah terdaftar di dalam database collection users atau belum,
86     FirebaseFirestore
87         .getInstance()
88         .collection("users")
89         .whereEqualTo("username", username)
90         .limit(1)
91         .get()
92         .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
93             @Override
94             public void onComplete(@NonNull Task<QuerySnapshot> task) {
95
96                 if(task.getResult().size() == 0) {
97                     // jika tidak terdapat di database dan email serta password, maka tidak bisa login
98                     binding.progressBar.setVisibility(View.GONE);
99                     showFailureDialog();
100                    return;
101                }
102
103                // jika terdaftar maka ambil email di database, kemudian lakukan autentikasi menggunakan email & password dari user
104                for(QueryDocumentSnapshot snapshot : task.getResult()) {
105                    String email = "" + snapshot.get("email");
106
107                    // fungsi untuk mengecek, apakah email yang di inputkan ketika login sudah terdaftar di database atau belum
108                    FirebaseAuth
109                        .getInstance()
110                        .signInWithEmailAndPassword(email, password)
111                        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
112                            @Override
113                            public void onComplete(@NonNull Task<AuthResult> task) {
114                                if(task.isSuccessful()) {
115                                    // jika terdapat di database dan email serta password sama, maka masuk ke homepage
116                                    binding.progressBar.setVisibility(View.GONE);
117                                    startActivity(new Intent(LoginActivity.this, HomeActivity.class));
118                                } else {
119                                    // jika tidak terdapat di database dan email serta password, maka tidak bisa login
120                                    binding.progressBar.setVisibility(View.GONE);
121                                    showFailureDialog();
122                                }
123                            }
124                        });
125                }
126            }
127        });
128    }

```

Kode Program 1 adalah perintah untuk proses *login* yang digunakan untuk memvalidasi *inputan user*, yang berada pada *line 74* sampai *line 80*. Setelah melakukan validasi selanjutnya akan dilakukan pengecekan *user account*, dimana jika tersedia dan *account* benar dibawa ke menu *dashboard* ke akun *mobile banking*.



Kode Program 2 Perintah untuk Proses Transaksi Tarik Tunai

```
80 private void inputPin() {
81     Dialog dialog;
82     Button btnSubmit;
83     EditText etPin;
84     dialog = new Dialog(this);
85     dialog.setContentView(R.layout.popup_pin);
86     dialog.setCanceledOnTouchOutside(false);
87     btnSubmit = dialog.findViewById(R.id.submit);
88     etPin = dialog.findViewById(R.id.pin);
89
90     btnSubmit.setOnClickListener(new View.OnClickListener() {
91         @Override
92         public void onClick(View view) {
93             String inputPin = etPin.getText().toString().trim();
94
95             // validasi pin oleh sistem
96             if(inputPin.isEmpty()) {
97                 Toast.makeText(WithdrawConfirmationActivity.this, "PIN tidak boleh kosong", Toast.LENGTH_SHORT).show();
98                 return;
99             } else if (!inputPin.equals(model.getPin()) && chance > 0) {
100                 chance--;
101                 Toast.makeText(WithdrawConfirmationActivity.this, "PIN Salah!, Kesempatan " + chance + " kali lagi", Toast.LENGTH_SHORT).show();
102                 return;
103             } else if (chance == 0) {
104                 showBlockedUser();
105                 return;
106             }
107
108             // jika pin benar maka akan lanjut ke halaman result tarik tunai
109             Intent intent = new Intent(WithdrawConfirmationActivity.this, WithdrawResultActivity.class);
110             intent.putExtra(WithdrawResultActivity.EXTRA_USER, model);
111             intent.putExtra(WithdrawResultActivity.EXTRA_NOMINAL, getIntent().getLongExtra(EXTRA_NOMINAL, 0));
112             startActivity(intent);
113
114         }
115     });
116     dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
117     dialog.show();
118 }
119
120 }
```

Kode Program 2 menjelaskan perintah untuk proses transaksi tarik tunai dimana *user* harus memilih jumlah nominal yang sudah disediakan, kemudian *user* diminta untuk memasukkan pin. Jika pada saat memasukkan pin benar maka akan dibawa ke *view* konfirmasi penarikan yang berada pada *file WithdrawResultActivit*.

Kode Program 3 Perintah untuk Proses Transaksi Transfer

```
52 protected void onCreate(Bundle savedInstanceState) {
53     super.onCreate(savedInstanceState);
54     binding = ActivityTransferConfirmationBinding.inflate(getLayoutInflater());
55     setContentView(binding.getRoot());
56
57     // halaman ini merupakan halaman konfirmasi transfer, dimana user memasukkan nominal transfer dan memasukkan pin
58
59
60     // inisiasi nama tujuan, rekening tujuan, dan bank tujuan
61     model = getIntent().getParcelableExtra(EXTRA_NASABAH);
62     binding.name.setText(model.getName());
63     binding.bank.setText(model.getBank());
64     binding.rekeningTujuan.setText(model.getRekening());
65
66
67     // inisiasi nama pengirim, rekening pengirim
68     getUserData();
69
70
71     // kembali ke halaman sebelumnya
72     binding.backButton.setOnClickListener(new View.OnClickListener() {
73         @Override
74         public void onClick(View view) {
75             onBackPressed();
76         }
77     });
78
79
80     // klik tombol transfer dan masukkan pin
81     binding.transferBtn.setOnClickListener(new View.OnClickListener() {
82         @Override
83         public void onClick(View view) {
84             inputPin();
85         }
86     });
87 }
```

Kode Program 3 menjelaskan pada saat ingin melakukan transaksi transfer, *user* diminta untuk memasukan jumlah nominal dan memilih bank tujuan, kemudian akan diminta untuk memasukkan pin.

4. SIMPULAN

Berdasarkan hasil penelitian Penerapan *Finite State Automata* Transaksi via *Mobile Banking* (Aplikasi BRImo) yang telah dilakukan maka dapat disimpulkan bahwa : (1) *Finite State Automata* adalah sebuah logika dasar yang digunakan sebagai dasar rancangan untuk melakukan proses transaksi via ATM; (2) Proses transaksi Aplikasi BRImo dilakukan dengan membagi tiap sistem ke dalam bentuk *state* hal ini bertujuan untuk mempermudah dalam penyelesaian sistem; (3) Bentuk algoritma dan *console application* aplikasi BRImo yang dibuat merupakan simulasi atau gambaran dari aplikasi BRImo yang telah digunakan nasabah saat ini, untuk menjelaskan proses penggunaan transaksi via *Mobile Banking* (Aplikasi BRImo).

DAFTAR PUSTAKA

- [1] Suharsih, R dan Atqiya, F. (2019). Penerapan Konsep Finite State Automata (FSA) pada Aplikasi Simulasi Vending Machine Yoghurt Walagri. *Jurnal Pendidikan Multimedia*. 1(2): 71-78.
- [2] Utdirartatmo, F. (2005). Teori Bahasa dan Otomata. Yogyakarta: Graha Ilmu.
- [3] Zimmermann, G., Vanderheiden, G., Gilman, A, S. (2002). Universal Remote Console - Prototyping for the Alternate Interface Access Standard. *Lecture Notes in Computer Science*. 26(15): 524-531.
- [3] Adenis, P., Mukherjee, K., & Ray, A. (2011). State splitting and state merging in probabilistic finite state automata. In *american control conference (acc)*. *Journal IEEE*. 5145-5150.
- [4] Kim dan Fainekos. (2014). Revision of Specification Automata under Quantitative Preferences. *Proceedings - IEEE International Conference on Robotics and Automation*. 71-72.
- [5] Widyasari, 2011. Telaah Teoritis Finite State Automata Dengan Pengujian Hasil Pada Mesin Otomata. *Jurnal Ilmiah Sisfoteknika*. 1(1): 59-67.
- [6] Sulistriyani, Suci. (2013). Pengaruh Minat Individu terhadap Penggunaan Mobile Banking. *Jurnal Universitas Brawijaya*.
- [7] Nendissa dan Pakereng. (2018). Penerapan Finite State Automata Pada Proses Transaksi Via ATM". Skripsi. Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga.
- [8] Asih, A, Y., Ambarwati, R, N., Hermaliani, E, H., Haryanti, T, Gata, W. Penerapan Konsep Finite State Automata Pada Aplikasi Simulasi Vending Machine Beras. *Jurnal Ilmiah Elektronika Dan Komputer*. 14(1): 130 - 140.
- [9] Achmad, R, R., Septiana, F, F., Syamsi, N., Prakoso, B, S., Novitasari, B. Penerapan Finite State Automata pada Vending Machine dalam Melakukan Transaksi Pengembalian Buku di Perpustakaan. *Journal Metik*. 5(1): 63-70.
- [10] Ambarwati, R, N., Asih, A, Y., Rahayu, S., Novitasari, H, B., Gata, W. 2021. Penerapan Konsep Finite State Automata Pada Vending Machine Dalam Pembelian Kartu E-Money Semua Bank Konvensional. *Journal Methomika*. 5(2): 87-90.

- [11] Maulana, A. S., Azizah, H. N., Kirana, K. C. (2019). Implementasi Finite State Automata (Fsa) Dengan Simulasi Vending Machine Pada Aplikasi Android. *Jurnal Edukasi Elektro*. 3(2): 110-120.
- [12] Putri, S. A dan Gata. (2020). Penerapan Konsep Finite State Automata Pada Aplikasi Simulasi Vending Machine Jamu Tradisional. *Jurnal Informatika*. 7(2):141-147.
- [13] Ma'arif, R. A., & Fauziah, F. (2018). Implementasi Finite State Automata (FSA) dalam Proses Pengisian Kartu Rencana Studi. *Journal of Information Technology and Computer Science*. 3(3): 115-120.
- [14] Nugraha, R. A., Yanto, Mulyani, A., Gata, W. (2020). Desain Vending Machine Rujak Buah Dengan Finite State Automata. *IJCIT (Indonesian Journal on Computer and Information Technology)*. 5(2): 198-207.
- [15] Sutopo, M. (2022, Oktober 22). Retrieved from Mursids Blog: <http://mursids.blogspot.com/2010/02/nondeterministic-finite-automata-nfa.html>