

# Implementasi Finite State Automata dalam Proses Pengisian Kartu Rencana Studi Mahasiswa

Ridwan Ahmad Ma'arif<sup>1</sup>, Fauziah<sup>2</sup>

<sup>1</sup>ridwan.22693@gmail.com, <sup>2</sup>fauziah@civitas.unas.ac.id

Program Studi Teknik Informatika, Universitas Nasional Jakarta

**Abstract**—Filling out the Study Plan Card is an obligatory activity for active students at a university. The number of courses and there are conditions in the taker is a common thing among students, but did not rule out that there are still many students who are wrong in taking the course so that the denied access to a course because there are compulsory courses that have not been taken. This paper discusses how Finite State Automata method of NFA type can be implemented in filling out the study plan card (KRS). With the implementation of this method, the user is expected to be helpful in filling out the study plan card and can better understand how the process of filling out the study plan card is correct.

**Intisari**—Mengisi Kartu Rencana Studi merupakan kegiatan yang wajib dilakukan untuk mahasiswa aktif di sebuah universitas. Banyaknya matakuliah dan terdapat Syarat dalam pengambilannya merupakan suatu hal yang lumrah kalangan mahasiswa, akan tetapi tidak menutup kemungkinan bahwa masih banyak mahasiswa yang salah dalam pengambilan matakuliah sehingga menyebabkan ditolaknya akses pengambilan suatu matakuliah disebabkan ada matakuliah wajib yang belum di ambil. Pada paper ini membahas tentang bagaimana metode Finite State Automata jenis NFA dapat diimplementasikan dalam mengisi kartu rencana studi (KRS). Dengan di terapkan metode ini, user diharapkan dapat terbantu dalam mengisi kartu rencana studi dan dapat lebih paham bagaimana proses mengisi kartu rencana studi benar.

**Kata kunci**—Mengisi Kartu Rencana Studi, Mengisi KRS, Finite State Automata, Mesin NFA.

## I. PENDAHULUAN

Teori bahasa dan automata merupakan bagian ilmu komputer. Teori ini sangat berguna untuk pengembangan ilmu komputer lebih lanjut baik dalam perangkat keras (hardware), dan perangkat lunak (software) [1].

Teori bahasa bertindak sebagai sarana komunikasi baik sesama manusia ataupun antara manusia dan mesin.

Sedangkan teori automata adalah teori mengenai mesin-mesin abstrak, dan berkaitan erat dengan teori bahasa formal. Automata dapat diterapkan pada aplikasi pengisian KRS mahasiswa.

Kartu rencana studi (KRS) adalah kartu yang berisi daftar mata kuliah yang akan diikuti oleh setiap mahasiswa dalam satu semester. Dalam KRS tercantum data mahasiswa (NPM, Nama, Kelas, Fakultas, Jurusan, Jumlah Semester dan Tahun Akademik yang diikuti), Kode Mata Kuliah, Mata Kuliah, SKS dan Kelas yang diikuti [2].

Pengisian KRS haruslah sangat diperhatikan, karena di beberapa mata kuliah terdapat syarat dalam pengambilan mata kuliah. Contohnya, jika mahasiswa ingin mengambil mata kuliah Bahasa dan Utama maka mahasiswa tersebut harus mengambil Algoritma Pemrograman terlebih dahulu. Sebagai mahasiswa hal ini haruslah dipahami, karena sering kali mahasiswa lupa mengambil mata kuliah yang menjadi syarat untuk mata kuliah lain bahkan untuk syarat skripsi, oleh karena itu dibutuhkanlah sebuah aplikasi yang mengatur pengisian kartu rencana studi. Pada penelitian ini penerapan automata digunakan untuk mengenal dan menangkap pola dalam proses aplikasi bantu ini. Finite State Automata (FSA) jenis NFA merupakan model yang digunakan untuk merancang aplikasi yang dapat membantu dalam pengisian kartu rencana studi [3].

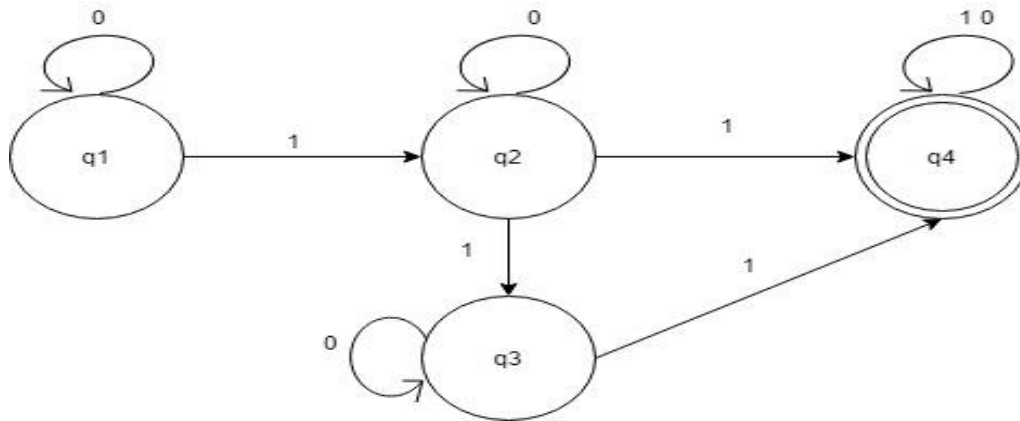
## II. TINJAUAN PUSTAKA

Metode yang digunakan dalam perancangan aplikasi bantu ini yaitu menggunakan FSA. Metode yang digunakan yaitu dengan menggunakan *finite state*, yang mana jika diambil contoh kasus sebagai berikut:

Tuple M pada FSA jenis NFA diantaranya  $(Q, \Sigma, \delta, S, F)$  yang mengartikan untuk  $Q$ =himpunan state,  $\Sigma$ =himpunan input,  $\delta$ =fungsi transisi,  $S$ =state awal,  $F$ =state akhir sebagai contoh sederhana dengan NFA [4].

Dari diagram Gambar 1, kita bisa melihat bahwa tuple-nya sebagai berikut:

$Q = \{q1, q2, q3, q4\}$   
 $\Sigma = \{0, 1\}$   
 $\delta =$  Fungsi transisi  
 $\delta(q1, 0) = q2, \delta(q1, 1) = q2, \delta(q2, 0) = q2,$



Gambar1. Diagram NFA

$$\delta(q2,1)=q3,q4, \delta(q3,0)=q3, \delta(q3,1)=q4, \\ \delta(q4,0)=q4, \delta(q4,1)=q4$$

$$S = \{q1\} \\ F = \{q4\}$$

Dari fungsi transisi tersebut maka tabel transisinya yaitu:

Table 1.  
Tabel Transisi

$\delta$	0	1
q1	q1	q1
q2	q3	(q3,q4)
q3	q3	q4
q4	q4	q4

Kasus pada Tabel 1 dapat diterapkan kedalam *pengisian KRS*, sebagai contoh bila q1 itu adalah mata kuliah Algoritma Pemrograman dan q2 adalah Bahasa dan Automata, dan himpunan *input* 0 sebagai di tidak ambil dan himpunan *input* 1 sebagai ambil, maka untuk bisa mengambil mata kuliah Bahasa dan Utama harus mengambil mata kuliah Algoritma Pemrograman (q1) terlebih dahulu [5].

### III. METODE PENELITIAN

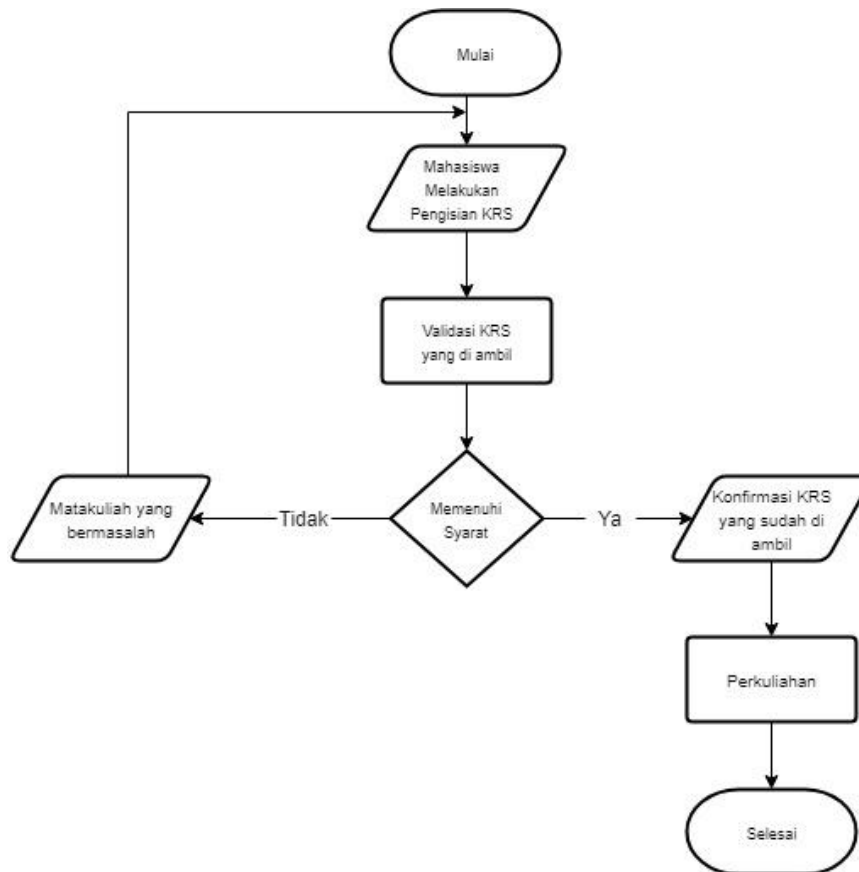
Sistem dirancang dengan menggunakan *flowchart*. Bagan ini menjelaskan tentang gambaran umum pada aplikasi bantu pengambilan matakuliah. Untuk jelasnya dapat dilihat di Aplikasi bantu pengisian *Kartu Rencana Studi* dirancang untuk memvalidasi mata kuliah yang sudah diambil oleh mahasiswa sehingga apabila mahasiswa ingin mengambil suatu mata kuliah maka sistem akan melakukan validasi terhadap mata kuliah yang akan diambil, apakah sudah memenuhi syarat atau tidak. Pada *flowchart* aplikasi ini dapat dijelaskan sebagai berikut:

- Mahasiswa melakukan pengisian *KRS*  
Mahasiswa menentukan jadwal kuliah yang akan diambil di semester berikutnya.
- Sistem akan memvalidasi mata kuliah yang diambil, apakah sudah memenuhi syarat atau belum memenuhi syarat.
- Sistem akan menampilkan laporan diterimanya atau ditolaknya pengisian *krs* tersebut. Bila di tolak maka mahasiswa akan diminta untuk mengisi kembali *krs* dan bila diterima mahasiswa sudah dapat melanjutkan perkuliahan di semester berikutnya.

Sistem dirancang dengan menggunakan *flowchart*. *Flowchart* ini menjelaskan tentang gambaran umum pada aplikasi bantu pengambilan matakuliah. Untuk jelasnya dapat dilihat diaplikasi bantu pengisian *Kartu Rencana Studi* dirancang untuk memvalidasi mata kuliah yang sudah diambil oleh mahasiswa sehingga apabila mahasiswa ingin mengambil suatu mata kuliah maka sistem akan melakukan validasi terhadap mata kuliah yang akan diambil, apakah sudah memenuhi syarat atau tidak.

Pada *flowchart* aplikasi ini dapat di jelaskan sebagai berikut:

- Mahasiswa melakukan pengisian *KRS*  
Mahasiswa menentukan jadwal kuliah yang akan diambil di semester berikutnya.
- Sistem memvalidasi mata kuliah yang diambil, apakah sudah memenuhi syarat atau belum memenuhi syarat.
- Sistem menampilkan laporan diterimanya atau ditolaknya pengisian *KRS* tersebut. Bila di tolak maka mahasiswa akan diminta untuk mengisi kembali *KRS* dan bila diterima mahasiswa sudah dapat melanjutkan perkuliahan disemester berikutnya.



Gambar 2. Flowchart untuk pengambilan Mata kuliah

Cara kerja dari simulasi pengisian kartu rencana studi dapat dilihat pada Gambar 3.

Gambar 3 menjelaskan alur pengambilan dari sebuah matakuliah dengan tujuan akhir pada peminatan sistem cerdas. Pada desain di atas terdapat simbol {Ambil,Tidak} simbol itu diumpamakan untuk mahasiswa apakah matakuliah terkait akan di ambil atau tidak. Secara formal finite state automata (FSA) pada Gambar 3 dinyatakan dalam 5 tupel yaitu  $M=(Q,\Sigma,\delta,S,F)$ . yang dapat di jelaskan sebagai berikut:

$Q$  = {Pengantar Teknologi, Fisika Dasar, Komputer dan Masyarakat, Logika Informatika, Algoritma Pemrograman 1, Algoritma Pemrograman 2, Rangkaian Sistem Digital, Pemrograman Visual, Struktur Data, Bahasa & Automata, Basis Data, Interaksi Manusia dan Komputer, Kriptografi, Sistem Jaringan Syaraf Tiruan, Text Mining, Sistem Cerdas }

$\Sigma$  = {Ambil, Tidak} himpunan input (1,0) 1: ambil, 0 : tidak ambil

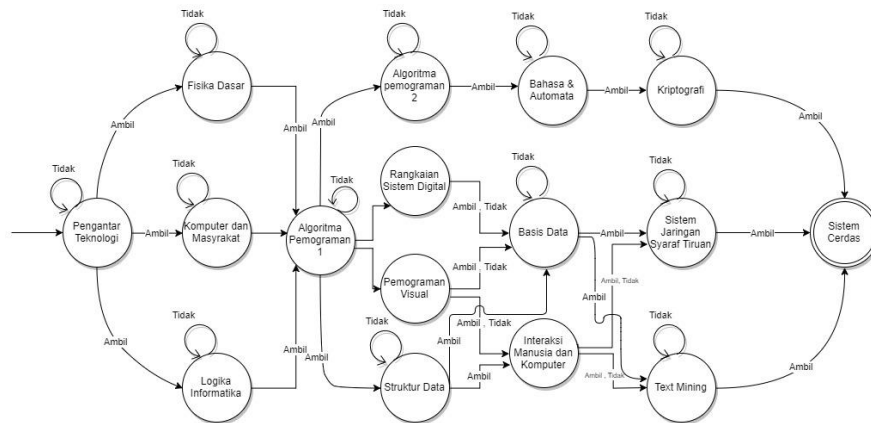
$\delta$  = Fungsi transisi untuk pengambilan matakuliah (Tabel 2)

$S$  = Pengantar Teknologi

$F$  = Sistem Cerdas

Terdapat simbol  $\emptyset$  dalam Tabel 2. Simbol  $\emptyset$  adalah simbol hampa[1]. Pada kasus ini, simbol ini diartikan bahwa tidak ada matakuliah lagi setelah sampai di state sistem cerdas. Dari Tabel 2 tersebut, kita mengetahui bahwa jika kita ingin mengambil suatu matakuliah maka ada matakuliah yang di ambil terlebih dahulu. Contoh jika kita ingin mengambil minat sistem cerdas dengan mempelajari matakuliah kriptografi maka kita harus mengambil matakuliah: Pengantar Teknologi – Fisika Dasar – Algoritma dan Pemrograman 1 – Algoritma dan Pemrograman 2 - Bahasa & Automata - Kriptografi – Sistem Cerdas. Contoh inilah yang disebut syarat dalam pengambilan suatu matakuliah.

Hasil dari aplikasi bantu ini adalah daftar matakuliah yang harus di ambil untuk bisa mengambil suatu matakuliah yang di inginkan oleh mahasiswa.



Gambar 3. Diagram state NFA Untuk pengambilan Mata Kuliah

$\delta$	Ambil	Tidak
Pengantar Teknologi	{Fisika Dasar, Komputer dan Masyarakat, Logika Informatika}	Pengantar Teknologi
Fisika Dasar	Algoritma Pemrograman 1	Fisika Dasar
Komputer dan Masyarakat	Algoritma Pemrograman 2	Komputer dan Masyarakat
Logika Informatika	Algoritma Pemrograman 3	Logika Informatika
Algoritma Pemrograman 1	{Algoritma Pemrograman 2, Rangkaian Sistem Digital, Pemrograman Visual, Struktur Data}	Algoritma Pemrograman 1
Algoritma Pemrograman 2	Bahasa & Automata	Algoritma Pemrograman 2
Rangkaian Sistem Digital	Basis Data	Basis Data
Pemrograman Visual	{Basis Data, Interaksi Manusia dan Komputer}	Basis Data
Struktur Data	{Basis Data, Interaksi Manusia dan Komputer}	Struktur Data
Bahasa & Automata	Kriptografi	Bahasa & Automata
Basis Data	{Sistem Jaringan Syaraf Tiruan, Text Mining}	Basis Data
Interaksi Manusia dan Komputer	{Sistem Jaringan Syaraf Tiruan, Text Mining}	{Sistem Jaringan Syaraf Tiruan, Text Mining}
Kriptografi	Sistem Cerdas 1	
Sistem Jaringan Syaraf Tiruan	Sistem Cerdas 2	Sistem Jaringan Syaraf Tiruan
Text Mining	Sistem Cerdas 3	Text Mining
Sistem Cerdas	$\emptyset$	$\emptyset$

#### IV. HASIL DAN PEMBAHASAN

Aplikasi yang dibuat merupakan aplikasi berbasis *desktop*. Dengan tampilan yang sederhana, sehingga memudahkan user dalam menggunakannya.

*Form* awal pada aplikasi ini berisikan tampilan *login*, dimana mahasiswa diwajibkan *login* dengan memasukkan NIM dan *password*, agar sistem dapat mendapatkan data KRS

mahasiswa di semester sebelumnya. *Form login* aplikasi bantu pengisian *kartu rencana studi* dapat dilihat di Gambar 4.

Setelah mahasiswa melakukan login, selanjutnya manusia di minta untuk mengisi *kartu rencana studi* dan mahasiswa juga dapat memilih peminatan akhir yang diinginkan. Untuk kasus ini kita memilih peminatan sistem cerdas. Tampilan *form* ini dapat dilihat di Gambar 5.

**Login Mahasiswa**  
Universitas Nasional

Username  
Password

**LOGIN**

Gambar 4. Form Login Aplikasi Bantu Pengisian KRS

**My Bio**  
Ridwan Ahmad Ma'arif  
Teknik Informatika  
163112706450051

**Kartu Rencana Studi**

Status	Country
<input checked="" type="checkbox"/>	Pengantar Teknologi
<input checked="" type="checkbox"/>	Fisika Dasar
<input checked="" type="checkbox"/>	Komputer Masyarakat
<input checked="" type="checkbox"/>	Logika Informatika

**Peminatan**  
Sistem Cerdas ☒ Selected  
Jaringan ☐ Unselected  
Multimedia ☐ Unselected

**Submit Result**

**Logout**

Gambar 5. Form Utama Aplikasi Bantu Pengisian KRS

**Perhatian**  
Kartu Rencana Studi anda di terima.  
Silahkan klik tombol result untuk melihat KRS anda.

**Close**

Gambar 6. Pop Up Yang Muncul Apabila KRS Diterima

**Result**

Hari	Mulai	Akhir	Ruang	Kode MK	KRS	Nama MK	SKS
Senin	08:00:00	09:40:00	005/4	1708030208	R03	Komunikasi Data dan Jaringan Komputer	4
Senin	09:50:00	11:30:00	006/1-1	1708030208	R03	Komunikasi Data dan Jaringan Komputer	4
Senin	11:40:00	13:20:00	Lab Informatika 1	1708030225	R03	Praktikum Pengelolaa Instalasi Komputer	1
Senin	13:30:00	15:10:00	309/3	1708030326	R04	Pengelolaan Instalasi Komputer	2
Senin	17:00:00	18:40:00	LAB JARINGAN KOMPUTER	1708020318	R02	Praktikum Pemrograman Berorientasi Objek	1
Selasa	08:00:00	09:40:00	Lab Perangkat Keras	1708030329	R03	Pemrograman Web	2
Rabu	13:30:00	15:10:00	Lab Informatika 1	1708030326	R03	Praktikum Pemrograman Web	1
Kamis	09:50:00	11:30:00	006/1-1	1708030302	R02	Bahasa dan Automata	3
Kamis	11:40:00	13:20:00	401/4	1708020317	R02	Pemrograman Berorientasi Objek	2
Kamis	13:30:00	15:10:00	108/4	1708020222	R03	Teknik Simulasi dan Pemodelan	3
Kamis	15:20:00	17:00:00	LAB JARINGAN KOMPUTER	1708030310	R04	Pemrograman Berorientasi Objek	2
Jumat	08:00:00	09:40:00	003/1	1708030312	R01	Metode Numerik	3

**Logout**

Gambar 7. Tampilan Result KRS Yang Aktif

**Perhatian**  
Kartu Rencana Studi anda di TOLAK.  
Ada mata kuliah wajib yang harus di ambil terlebih dahulu.

**Close**

Gambar 8. Pop Up Yang Muncul Apabila KRS Ditolak

**Result**

Status Pengambilan	Mata Kuliah Yang Wajib Di Ambil
<input type="checkbox"/>	Algoritma Pemrograman 1
<input type="checkbox"/>	Algoritma Pemrograman 2
<input type="checkbox"/>	Fisika Dasar

**Logout**

Gambar 9. Tampilan Result Yang Berisi Mata Kuliah Yang Harus Diambil

Data yang telah di *input* kedalam aplikasi ini akan divalidasi oleh sistem, apakah *krs* yang di isi memenuhi syarat atau tidak. Jika proses validasi selesai, maka *user* akan di tampilkan laporan diterimanya *krs* yang di isikan. Jelasnya lihat pada gambar-gambar program di atas:

Apabila di tolak, maka *user* akan di tampilkan laporan kenapa ditolaknya *krs* tersebut dan akan muncul rekomendasi mata kuliah yang harus di ambil. Lihat Gambar 8. Pada Gambar 9, mahasiswa diharuskan memilih mata kuliah yang tertera di *result*. Sehingga sistem dapat mengkonfirmasi *krs* yang di *input*.

## V. KESIMPULAN DAN SARAN

*Finite state automata* berjenis *NFA* sangat sesuai dalam gambaran kerja sistem pada aplikasi bantu ini. Penempatan matakuliah dilakukan menggunakan state yang telah diurutkan berdasarkan ketentuan dari masing-masing matakuliah. Dengan aplikasi ini, user diharapkan dapat terbantu dan mengetahui alur matakuliah yang benar sesuai dengan kebijakan di universitas masing-masing.

## DAFTAR PUSTAKA

- [1] P. Adenis and A. Ray, "State Splitting and State Merging in Probabilistic Finite State Automata  $\star$ ," pp. 5145–5150, 2011.
- [2] X. Zhao, J. Xue, C. Hu, R. Ma, and S. Zhang, "RESEARCH ON SOFTWARE BEHAVIOR MODELING BASED ON EXTENDED FINITE STATE AUTOMATA."
- [3] R. O. Reddy, "A probabilistic finite state architecture to classify texture images," 2016.
- [4] S. Manko, S. Diane, and V. Lokhin, "Task Planning in Robot Groups for Problems with Implicitly Defined Scenarios Based on Finite-State Automata Technique," no. 1, pp. 348–351, 2017.
- [5] N. D. Switching, N. D. Switching, and E. Power, "MDC-DFA : A Multi-dimensional Cube Deterministic Finite Automata - Based Feature Matching Algorithm," no. 2012, pp. 1119–1124, 2015.