

IMPLEMENTASI KONSEP FINITE STATE AUTOMATA (FSA) UNTUK PENGOLAHAN DATA SANTRI PADA PONDOK PESANTREN

Syafiqul Shubuh¹, Laela Kurniawati², Eni Heni Hermaliani³, Windu Gata⁴

¹ Ilmu Komputer Universitas Nusa Mandiri, Jakarta, Indonesia

^{2,3}Sistem Informasi Universitas Nusa Mandiri, Jakarta, Indonesia

⁴Informatika Universitas Nusa Mandiri, Jakarta, Indonesia

¹14210221@nusamandiri.ac.id, ²laela@nusamandiri.ac.id, ³enie_h@nusamandiri.ac.id,

⁴windu@nusamandiri.ac.id.

ABSTRAK

Jumlah data pondok pesantren di Indonesia mencapai 27.722 pada pangkalan data pondok pesantren kementerian agama. Pesantren merupakan lembaga pendidikan islam tradisional, siswa atau yang biasa disebut dengan santri menetap bersama di bimbing dan di kontrol penuh oleh kiai dalam satu lingkungan pesantren yang biasanya terdiri dari ruang asrama, ruang belajar, tempat ibadah, dan lapangan olahraga. Ada beberapa pesantren besar yang jumlah santrinya mencapai ribuan bahkan puluhan ribu dan banyak juga pesantren masih belum memiliki sistem pendataan yang baik, tentu akan sangat merepotkan untuk mengolah data santri untuk berbagai keperluan pondok pesantren. Data santri dapat di gunakan untuk data daftar ulang, pembayaran SPP, penilaian, target hafalan, dan alumnii jika sudah lulus. Pada paper kali ini penulis mencoba mengimplementasikan konsep finite state automata (FSA) jenis Nondeterministic Finite Automata (NDFA) dalam siklus data santri untuk berbagai kebutuhan pondok pesantren dan diharapkan dapat mempermudah pengolahan data santri pada pondok pesantren.

Kata Kunci— Finite State Automata(FSA), Nondeterministic Finite Automata (NDFA), Pengolahan data, santri.

ABSTRACT

The number of data on boarding schools in Indonesia reached 27,722 in the database of boarding schools of the Ministry of Religious Affairs. Pesantren is a traditional Islamic educational institution, students or commonly referred to as santri live together guided and fully controlled by kiai in a pesantren environment which usually consists of dormitory rooms, study rooms, places of worship, and sports fields. There are several large pesantren whose number of students reaches thousands and even tens of thousands and many pesantren still do not have a good data collection system, of course it will be very troublesome to process student data for various purposes of the pesantren. Student data can be used for re-registration data, tuition payments, assessments, memorization targets, and alumni if they have graduated. In this paper, the author tries to implement the concept of finite state automata (FSA) type Nondeterministic Finite Automata (NDFA) in the cycle of student data for various needs of the boarding school and is expected to facilitate the processing of student data at the boarding school.

Keywords— Finite State Automata(FSA), Nondeterministic Finite Automata (NDFA), Data processing, Santri.

1. PENDAHULUAN

Jumlah data pondok pesantren di Indonesia mencapai 27.722 pada pangkalan data pondok pesantren kementerian agama. Pesantren merupakan lembaga pendidikan islam tradisional, siswa atau yang biasa disebut dengan santri menetap bersama di bimbing dan di kontrol penuh oleh kiai dalam satu lingkungan pesantren yang biasanya terdiri dari ruang asrama, ruang belajar, tempat ibadah, dan lapangan olahraga. Ada beberapa pesantren besar yang jumlah santrinya mencapai ribuan bahkan puluhan ribu dan banyak juga pesantren masih belum memiliki sistem pendataan yang baik, tentu akan sangat merepotkan untuk mengolah data santri untuk berbagai keperluan pondok pesantren.

Data santri dapat di gunakan untuk data daftar ulang, pembayaran SPP, penilaian, target hafalan, dan alumn jika sudah lulus. Pada paper kali ini penulis mencoba mengimplementasikan konsep *finite state automata* (FSA) jenis *Nondeterministic Finite Automata* (N DFA) dalam proses pengolahan data santri untuk berbagai kebutuhan pondok pesantren dan diharapkan dapat mempermudah pengolahan data santri pada pondok pesantren nantinya.

Sebagai pengembangan Ilmu Komputer yang meliputi *hardware* dan *software*, Teori Bahasa dan Automata sangatlah berguna. Teori Bahasa sebagai jembatan dan sarana komunikasi antara manusia dan mesin. Sedangkan pembahasan mengenai mesin-mesin abstrak yang berkaitan dengan Bahasa formal dijelaskan secara detail pada pembahasan teori automata [1]. Konsep teori Bahasa dan Automata pada penelitian ini akan diterapkan pada proses pengolahan data santri.

Model komputasi pada automata terdiri dari *Finite Automata* (FA) atau yang sering juga disebut dengan *Finite State Automata* (FSA), *Pushdown Automata* (PA) dan *Turning Machine* (TM). Teori *Finite State Automata* (FSA) adalah mesin otomata dari bahasa regular dengan suatu model matematika dari sebuah sistem yang menerima *input* dan *output* [2].

Finite State Automata (FSA) adalah suatu sistem mesin otomatis yang menerima input dan output.[3][4] Prinsip kerja Finite State Automata yaitu mesin membaca memori masukan yang berbentuk *tape*, yaitu 1 karakter pada tiap penggunaan *head* baca yang dikendalikan oleh kontak kendali *state*, dimana pada mesin tersebut terdapat beberapa *state*. *Finite State Automata* selalu dalam keadaan yang disebut dengan *state* awal pada saat *finite automata* mulai membaca *tape*. Pergantian dari *state* terjadi pada mesin saat sebuah karakter terbaca. Pada saat sebuah *head* telah menuju pada akhir *tape* dan dalam posisi *state* akhir, maka *string* yang terdapat pada *tape* akan dikatakan diterima *Finite Automata* [5].

Berdasarkan definisi pada kemampuan perubahan *state* FSA dapat dikategorikan menjadi dua yaitu *Deterministic Finite Automata* (DFA) dan *Nondeterministic Finite Automata* (NFA). Pada DFA

suatu *state* hanya ada satu *state* selanjutnya untuk tiap simbol masukan (*Input*) yang diterima atau hanya memiliki satu *start state* dan satu transisi untuk tiap anggota dari himpunan *input*. Sebaliknya pada NFA suatu *state* dapat berpindah dari *state* tertentu ke *state* lain sebagai asumsi terhadap suatu *input*, dalam NFA suatu *state* mempunyai 0 ataupun lebih opsi pilihan untuk *state* selanjutnya untuk tiap pasangan *state* *input* [6].

2. TINJAUAN PUSTAKA

Beberapa literatur yang membahas penelitian serupa terakit *Finite State Automata*, Paper karya Ridwan Ahmad Ma'arif membahas mengenai "Implementasi *Finite State Automata* dalam proses pengisian kartu rencana studi mahasiswa", penerapan automata pada paper R. Ahmad ini digunakan untuk membangun sebuah aplikasi yang dapat digunakan dalam pengisian KRS, sehingga dapat membantu mempermudah mahasiswa dalam memahami proses pengisian KRS [7].

Berikutnya paper karya Indra Dharma Wisjaya, Nindy Alvi Fajarina dan Dian Hanifudin Subhi membahas mengenai edukasi cara berternak burung puyuh dengan membangun sebuah game simulator berternak burung puyuh. Game simulator ini mengimplementasikan kecerdasan buatan dengan metode *Finite State Machine* untuk mempermudah observasi pada objek selama proses berlangsung. Simulasi edukasi meliputi proses penetasan telur hingga masa afkir dimana burung puyuh sudah tidak dapat memproduksi telur kembali [6].

Rieke Adriati W, Herman Tolle dan Onny Setyawati mengembangkan aplikasi TTS Text-To-Speech Bahasa indonesia yang menggunakan metode FSA berbasis android yang dapat diakses dimana saja dengan memanfaatkan API Google sebagai data voice nya sehingga aplikasi relatif lebih ringan. Aplikasi Text-To-Speech mampu merubah teks menjadi ucapan suara, para penyadang tuna netra dan low vision banyak yang menggunakan aplikasi ini sebagai media komunikasi [8].

Affis Masturina Nisa dan Hafidz Kurniansyah membangun sebuah game maze (labirin) dengan menggunakan adobe flash sebagai media pembelajaran anak. Finite state automata digunakan sebagai pondasi dasar rancangan software dan hardware pada game ini [5].

Sahrul, Fitri Karimah, Aries Dwi Prasetyo, dkk. Merancang sebuah game simulasi latihan dan pembiasaan masyarakat untuk memilah sampah organik dan non organic, dengan teori automata FSA finite state machine non- deterministic dengan kombinasi transisi epsilon (E) atau dikenal dengan E-NFA [9].

Mohamed Hamada dan Sayota Sato dari jepang, menggunakan finite satate automata untuk membangun sebuah Game-based Learning system berbasis robot lego NXT sebagai pembelajaran teori

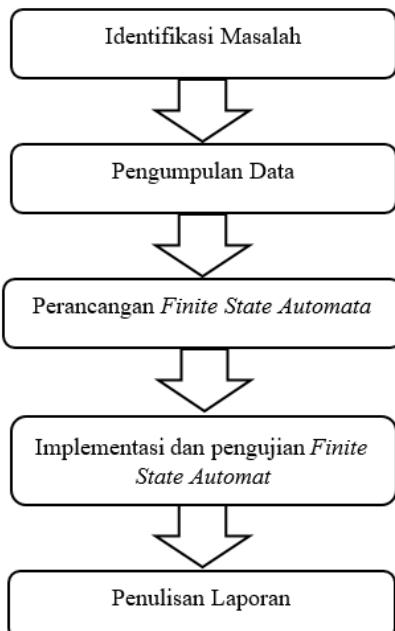
komputasi dengan mengimplementasikan pemrograman Java dan automaton game based robot [10].

Dari china Zhipeng Zhang, Chengyi Xia, Zengqiang Chen, membahas teori non deterministic finate automata NDFA dapat menunjukkan proses yang tidak pasti dalam pemodelan dan analisis, padahal kesetabilan adalah hal yang penting dalam teori riset. Pada paper ini fokus membahas maslah kesetabilan NDFA dalam framework matrix semi-tensor product (matrix STP). Dengan memanfaatkan pendekatan matrix STP dapat memperoleh kondisi yang lebih stabil dan efektif pada NDFA [11].

Terakhir sebuah buku karangan A. W. Aranski merupakan salah satu buku pegangan mahasiswa teknik informatika didalamnya membahas detail teori bahasa dan otomata secara rinci seperti: deskripsi singkat, standar kompetensi, kompetensi dasar, materi pokok, hingga petunjuk pembelajaran.[12]

3. METODE PENELITIAN

Metodologi yang digunakan pada penelitian ini sebagaimana dijelaskan pada Gambar 1, yaitu: Identifikasi Masalah >> Pengumpulan data >> Perancangan *Finite State Automata* >> Implementasi dan pengujian *Finite State Automata* >> Penulisan laporan.

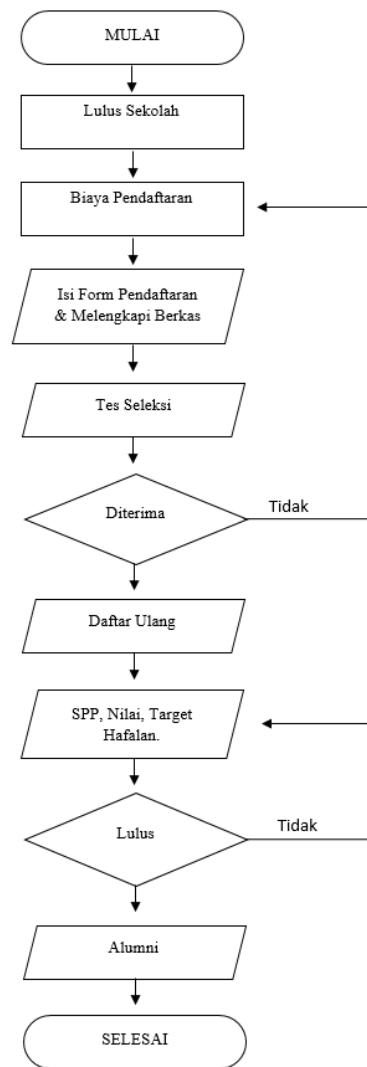


Gambar. 1 Tahap Penelitian

Pada tahap Identifikasi Masalah: dilakukan analisis permasalahan pada proses terkait penerapan *Finite State Automata* pengolahan data santri dipondok pesantren; Tahap Pengumpulan Data: mengakomodir data-data terkait, baik dari penelitian-penelitian sebelumnya, buku, serta sumber lain yang membahas hal serupa; Tahap Perancangan *Finite State Automata*: perancangan *Finite State Automata* menggunakan NDFA (*Non Deterministic Finite*

Automata) dilakukan pada penelitian ini; Tahap Implementasi dan Pengujian *Finite State Automata* : membuat sistem dengan menerapkan *Finite State Automata* untuk pengolahan data santri dan dilakukan pengujian dengan evaluasi terhadap keseluruhan perancangan *Finite State Automata* pengolahan data santri menggunakan NDFA (*Non Deterministic Finite Automata*) yang telah dibuat, apakah sudah berjalan dengan semestinya; Tahap Penulisan Laporan : proses akhir yang dilakukan pada penelitian ini adalah menyusun laporan dari hasil *implementasi* dan pengujian yang telah dilakukan.

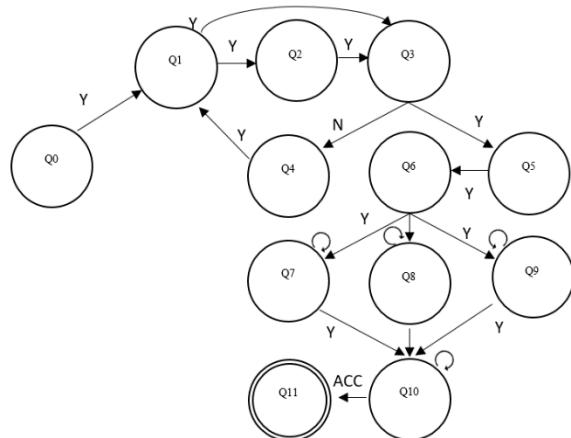
Berikut ini dibuat alur program sebagai pendefinisian diagram state yang digunakan, alur dan sistem yang dibangun dijelaskan pada gambar berikut.



Gambar. 2 Diagram State

Diagram state dari gambar 2 diatas menjelaskan *start state* dari proses adalah lulus sekolah untuk awal proses pengolahan data santri dan menuju *final state* yaitu alumni dengan menerima input dari lulus atau tidak lulus.

4. HASIL PENELITIAN



Gambar. 3 Rancangan Diagram State

Berdasarkan diagram diatas ketahui bahwa:

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

$\Sigma = \{0,1\}$ 1 = Ambil, 0 = Tidak Ambil

δ = Fungsi Transisi

$$S = Q0$$

Tabel. 1 Daftar State

State	Deskripsi
Q0	Lulus sekolah
Q1	Biaya pendaftaran
Q2	Isi form pendaftaran & melengkapi berkas
Q3	Hasil tes
Q4	Tidak diterima
Q5	Diterima
Q6	Daftar ulang
Q7	Spp
Q8	Nilai
Q9	Hafalan
Q10	Syarat kelulusan
Q11	Alumni

Tabel. 2 Daftar Simbol

Tabel. 2 Daftar Simbol	
Simbol Input	Deskripsi
Y	<i>Yes</i> untuk menuju ke state berikutnya
N	<i>No</i> untuk menuju ke state berikutnya
ACC	<i>Accept</i>

Tabel. 3 Fungsi Relasi Transisi

Table 3: Target Relation Transfer			
Δ	Y	N	ACC
Q0	Q1	Q11	0
Q1	Q2	0	0
Q2	Q3	0	0
Q3	Q5	Q4	0
Q4	Q1	0	0
Q5	Q6	0	0
Q6	Q7, Q8, Q9	0	0
Q7	Q10	0	0
Q8	Q10	0	0
Q9	Q10	0	0
Q10	Q11	Q7, Q8, Q9	Q11
Q11	0	0	0

Proses perpindahan *state* input yang masuk dijelaskan pada Tabel 3. Sedangkan pada gambar 3 rancangan *diagram state* dijelaskan *state* awal Q0 mendapat input Y maka akan beralih ke *state* Q1, dan selanjutnya hingga pada *state* Q3 mendapat *input* N maka selanjutnya akan berpindah ke *state* Q4 dan akan kembali ke *state* Q1, dan jika *state* Q3 mendapat input Y maka akan berpindah ke *state* berikutnya hingga ke *state* Q10. Kemudian pada *state* Q10 mendapat *input* ACC maka akan sampai ke *state* terakhir yaitu Q11. Perpindahan *state* akan bernilai ϵ_c (String Kosong) jika input yang masuk tidak sesuai.

Agar dapat dengan mudah mengimplementasikan proses sebelumnya (rancangan N-DFA) proses berikutnya adalah membuat detail rancangan aplikasi.

Proses Pendaftaran

1. Untuk mendaftar calon santri baru harus dinyatakan lulus dalam menyelesaikan pendidikan sebelumnya.
 2. Calon santri membayar biaya pendaftaran
 3. Calon santri melengkapi form pendaftaran & melengkapi berkas pada website pesantren.
 4. Calon santri mengikuti tes seleksi
 5. Hasil tes, jika dinyatakan lulus/diterima calon santri melakukan daftar ulang, dan jika dinyatakan tidak lulus/diterima calon santri dapat mendaftar kembali dan mengikuti tes seleksi pada gelombang berikutnya tanpa harus mengisi form dan melengkapi berkas.

Proses Penerimaan

1. Jika calon santri dinyatakan lulus/diterima pada tes seleksi maka harus melakukan daftar ulang.
 2. Status berubah menjadi santri aktif.
 3. Santri dapat mengakses halaman pembayaran spp, nilai, dan target hafalan pada website pesantren.
 4. Santri harus memenuhi syarat kelulusan yang telah ditentukan dari pembayaran spp, target nilai, dan target hafalan dari syarat kelulusan.
 5. Status alumni jika santri dinyatakan lulus, jika tidak santri dapat memenuhi point 4 sebelumnya.

5. KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa konsep *Finite State Automata* (FSA) dapat diimplementasikan sebagai logika dasar dalam pengolahan data santri dipondok pesantren. Untuk mempermudah dalam tahap pengolahan data santri dilakukan dengan membagi sistem ke dalam beberapa bentuk *state*.

Hasil dari penelitian ini hanya berupa rancangan pengolahan data santri dengan memanfaatkan konsep *Finite State Automata* (FSA), sehingga diharapkan nantinya dapat dikembangkan menjadi aplikasi desktop atau online.

Daftar Pustaka

- [1] Budyanaita Asrun, "KONSEP FINITE STATE AUTOMATA DALAM PROSES PENDAFTARAN UJIAN SKRIPSI DI FAKULTAS TEKNIK KOMPUTER UNCP," *J. Ilm. Inf. Technol. d'Computare*, pp. 5–9, 2020.
- [2] A. P. Giovani, F. Zamachsari, E. D. Agustono, M. I. Prasetya, and W. Gata, "IMPLEMENTASI FINITE STATE AUTOMATA DALAM SIKLUS PEMBELAJARAN MAGISTER ILMU KOMPUTER STMIK NUSA MANDIRI," 2020.
- [3] Anggun Yuli Asih, Rini Novi Ambarwati, Eni Heni Hermaliani, Tuti Haryanti, and Windu Gata, "Penerapan Konsep Finite State Automata Pada Aplikasi Simulasi Vending Machine Beras," *Elkom J. Elektron. dan Komput.*, vol. 14, no. 1, pp. 130–140, 2021, doi: 10.51903/elkom.v14i1.442.
- [4] S. Permana, "Penerapan Konsep Finite State Automata Pada Layanan Pengajuan Open Proxy Access Di Lingkungan Provinsi Dki Jakarta," *J. Informatics Commun. Technol.*, vol. 3, no. 1, pp. 42–49, 2021, doi: 10.52661/j_ict.v3i1.72.
- [5] A. M. Nisa and H. Kurniansyah, "Perancangan dan Implementasi Finite State Automata pada Pusheen Cat Maze Game dengan Adobe Flash," *Res. Comput. Inf. Syst. Technol. Manag.*, vol. 2, no. 1, p. 13, 2019, doi: 10.25273/research.v2i1.4268.
- [6] I. D. Wisjaya, N. A. Fajarina, and D. H. Subhi, "Simulasi Pembelajaran Berternak Burung Puyuh Menggunakan Metode Finite State Machine Berbasis Android," *J. Teknol. Inf. dan Terap.*, vol. 6, no. 1, pp. 12–18, 2019, doi: 10.25047/jtit.v6i1.103.
- [7] R. A. Ma'arif and F. Fauziah, "Implementasi Finite State Automata (FSA) dalam Proses Pengisian Kartu Rencana Studi," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 3, no. 3, pp. 115–120, 2018, doi: 10.31328/jointecs.v3i3.816.
- [8] R. A. W, H. Tolle, and O. Setyawati, "Pengembangan Aplikasi Text-to -," *Pengemb. Apl. Text-to-Speech Bhs. Indonesia Menggunakan Metod. Finite State Autom. Berbas. Android*, vol. 5, no. 1, 2016, [Online]. Available: https://www.academia.edu/61451518/Pengembangan_Aplikasi_Text_to_Speech_Bahasa_Indonesia_Menggunakan_Metode_Finite_State_Automata_Berbasis_Android
- [9] A. Perguruan and T. Aperti, "Jurnal Teknologi PENGEMBANGAN APLIKASI PERMAINAN ' PILAH SAMPA H ' MENGGUNAKAN PEMODELAN FINITE STATE Jurnal Teknologia," vol. 1, no. 1, pp. 38–46, 2018.
- [10] M. Hamada and S. Sato, "A game-based learning system for theory of computation using Lego NXT robot," *Procedia Comput. Sci.*, vol. 4, pp. 1944–1952, 2011, doi: 10.1016/j.procs.2011.04.212.
- [11] Z. Zhang, C. Xia, and Z. Chen, "On the stabilization of nondeterministic finite automata via static output feedback," *Appl. Math. Comput.*, vol. 365, p. 124687, 2020, doi: 10.1016/j.amc.2019.124687.
- [12] A. W. Aranski, *Teori Bahasa dan Otomata*, 2018th ed. Pustaka Galeri Mandiri.