

**PENGEMBANGAN APLIKASI PERMAINAN “PILAH  
SAMPAH” MENGGUNAKAN PEMODELAN *FINITE STATE  
MACHINE***

*Sahrul<sup>1</sup>, Fitri Karimah<sup>2</sup>, Alzahid Muhazabah<sup>3</sup>, Aries Dwi  
Prasetyo<sup>4</sup>, Ariana Yunita<sup>5</sup>, Nurulbaiti Listyendah Zahra<sup>6</sup>*

<sup>1, 2, 3, 4, 5</sup>Program Studi Ilmu Komputer, Fakultas Sains dan Komputer, Universitas Pertamina, Jakarta, Indonesia

<sup>6</sup>Program Studi Teknik Lingkungan, Universitas Pertamina, Jakarta, Indonesia

*E-mail: yunita.ariana@universitaspertamina.ac.id*

**ABSTRACT**

*This paper looks at how Finite State Machine (FSM) can be applied to design a game application namely “Pilah Sampah” as a business model diagram. “Pilah Sampah” aims to classify waste into organic and non-organic by using touch screen. In this game, scores will be added if an user can put the waste into the right place and vice versa. The type of FSM used in this paper is non- deterministic finite automaton (NFA) with epsilon ( $\epsilon$ ) transition or known as  $\epsilon$ -NFA (Epsilon NFA). Basically, each transition, input and state of NFA shows the behaviour of the game. The importance of this research is to show that automata theory can be used to help a game developer to design a game. After doing blackbox test for “Pilah Sampah” game, it results that the game application can be developed by the  $\epsilon$ -NFA modeling.*

**Keywords:** *game application, game design, Finite State Machine,  $\epsilon$ -NFA*

**ABSTRAK**

*Paper ini membahas bagaimana Finite State Machine (FSM) dapat diaplikasikan untuk mendesain aplikasi permainan bernama Pilah Sampah sebagai diagram model bisnis. Pilah Sampah bertujuan untuk mengklasifikasikan sampah menjadi organik dan non-organik dengan menggunakan sentuhan tangan. Jenis FSM yang digunakan adalah non- deterministic finite automaton dengan transisi epsilon ( $\epsilon$ ) atau dikenal dengan  $\epsilon$ -NFA. Pada dasarnya, tiap transisi, input dan state dari NFA menunjukkan karakteristik atau keadaan-keadaan pada sebuah permainan. Pentingnya penelitian ini adalah untuk menunjukkan bahwa teori automata dapat digunakan untuk membantu pembuat permainan untuk mendesain permainan tersebut. Setelah melakukan percobaan untuk permainan “Pilah Sampah”, penelitian ini menghasilkan bahwa aplikasi permainan dapat dibangun dari pemodelan  $\epsilon$ -NFA.*

**Kata kunci:** *aplikasi permainan, desain permainan, Finite State Machine,  $\epsilon$ -NFA*

## 1. PENDAHULUAN

Seiring berkembangnya zaman, dunia pendidikan pun kian berkembang pesat. Salah satu bidang yang terbilang baru jika dibandingkan dengan bidang sains lainnya seperti matematika dan fisika, yaitu ilmu komputer atau teknologi informasi, yang selama beberapa tahun belakang ini terus membuat terobosan. Salah satu bahan pelajaran yang dikaji dalam bidang ilmu komputer yaitu teori automata. Teori automata telah banyak digunakan dalam dunia ilmu komputer, terutama dalam industri aplikasi permainan. Di balik sebuah aplikasi permainan yang sering kali dijadikan bahan hiburan bagi banyak orang, terdapat model automata yang menggambarkan keadaan – keadaan yang ada di dalam sistem aplikasi permainan tersebut.

Jamil et.al (2016) menunjukkan bahwa aplikasi permainan dapat didesain dengan menggunakan teori automata berupa *mealy machine*. Jurnal tersebut juga menjelaskan bahwa teori automata memudahkan *programmer* untuk memahami cara kerja permainan [3]. Selain itu, terdapat pula sebuah penelitian yang memanfaatkan *Finite State Machine* untuk mendesain *elevator* [7] dan juga untuk membuat permainan simulasi untuk manajemen hotel [5].

Pada paper ini akan dibahas mengenai pengembangan aplikasi permainan yang memanfaatkan salah satu materi teori automata yaitu *non-deterministic finite automata*. Aplikasi permainan yang dibuat yaitu simulasi pilah sampah di mana permainan hanya akan berhenti jika pemain gagal memindahkan sampah ke dalam tempat sampah. Pada bab selanjutnya akan dibahas mengenai dasar teori yang digunakan pada penelitian ini yaitu tentang *Finite State Machine* dan NFA, metodologi dan perancangan yang digunakan, hasil dan pembahasan, serta kesimpulan dan saran dari penelitian ini.

## 2. DASAR TEORI

### 2.1. Finite State Machine

*Finite State Machine* (FSM) merupakan metodologi perencanaan *control system* yang mendeskripsikan tingkah laku sistem yang bekerja menggunakan 3 prinsip yaitu *state*, *event*, *action*. FSM terdiri dari 5-tuples yaitu *state* ( $Q$ ), *input* ( $\Sigma$ ), *fungsi transisi* ( $\delta$ ), *start state* ( $q_0$ ), dan *final state* ( $F$ ) [2]. Berikut ini adalah beberapa penjelasan mengenai simbol pada Finite State Machine.

Tabel 1. Simbol Pada Finite State Machine

	<i>State</i>
	<i>End state</i>
	<i>Transition</i>

#### 2.1.1. Non-deterministic Finite Automaton

Sebuah *Nondeterministic Finite Automaton* (NFA) merupakan jenis *finite state machine* (FSM) dimana salah satu *next state* tidak sepenuhnya ditentukan oleh *current state* ataupun *input*. Himpunan dari *possible next states* menyatakan bahwa automata dapat berpindah dari state tertentu ( $q_a$ ) ke state yang lain ( $q_b$ ) sebagai tanggapan terhadap suatu input ( $\alpha$ ). Jenis FSM yang lain, *Deterministic Finite Automaton* (DFA), mempunyai tepat satu *start state* ( $q_0$ ) dan tepat satu transisi ( $\delta$ ) untuk setiap anggota

himpunan input ( $\Sigma$ ). Dalam NFA, jumlah *start state* dan transisi untuk setiap anggota himpunan input ( $\Sigma$ ) tidak harus tepat satu [2]. Notasi bahasa formal dari NFA adalah sebagai berikut:

$$A = \{ Q, \Sigma, \delta, q_0, F \} \quad (1)$$

### 2.1.2. NFA With Epsilon ( $\epsilon$ ) Transition

NFA memiliki beberapa jenis, salah satunya adalah  $\epsilon$ -NFA yang memungkinkan terjadinya transisi dengan tanpa input atau *empty string*. Akibatnya, NFA diperbolehkan melakukan transisi secara spontan tanpa menerima *input*. Notasi formal dari  $\epsilon$ -NFA hampir sama dengan NFA hanya dalam himpunan input ( $\Sigma$ ) bisa terdiri dari epsilon ( $\epsilon$ ) yang memungkinkan transisi dengan *empty string* [4].

## 2.2 Sampah Organik dan Non-Organik

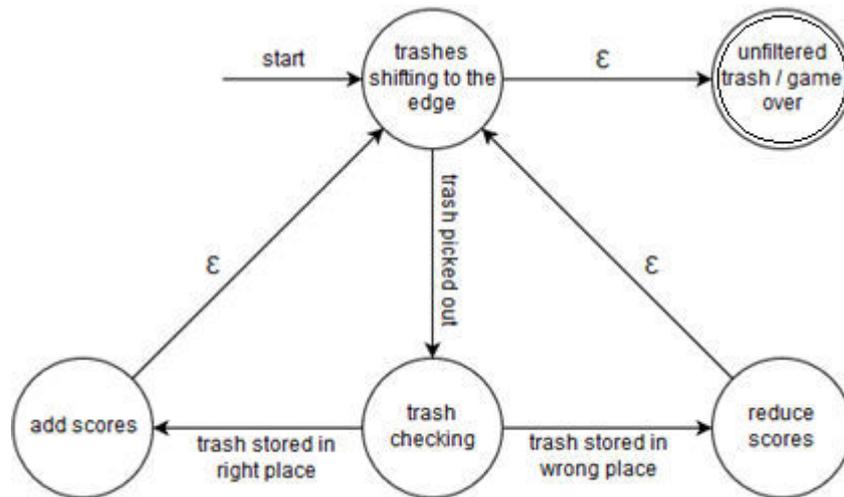
Menurut UU No 18 tahun 2008, sampah merupakan sisa kegiatan sehari-hari manusia dan/atau proses alam yang berbentuk padat [6]. Jika tidak dikelola dengan baik, sampah akan menjadi permasalahan besar, tidak hanya mengganggu secara estetika namun juga dapat mencemari lingkungan dan mengancam kesehatan manusia.

Untuk memudahkan proses pengolahan sampah, timbulan sampah yang dihasilkan oleh manusia sebaiknya dipilah ke dalam beberapa golongan. Secara umum, penggolongan jenis sampah dapat dibagi menjadi dua, yaitu sampah mudah terdekomposisi secara alami (*biodegradable*) dan sampah sulit atau tidak terdekomposisi secara alami (*nonbiodegradable*). Contoh sampah bersifat *biodegradable* diantaranya sampah sisa dapur, sisa makanan, sampah sisa sayur dan kulit buah-buahan, dedaunan, kayu dan sampah kebun. Contoh sampah *nonbiodegradable* adalah kertas, kardus, tisu, plastik, tekstil, tembikar, kaca, kaleng dan logam[1].

Di Indonesia, sampah *biodegradable* dikenal dengan istilah sampah organik sedangkan sampah *nonbiodegradable* dikenal dengan istilah sampah nonorganik. Penggolongan inilah yang digunakan dalam permainan Pilah Sampah ini. Walaupun sebenarnya penggunaan istilah tersebut kurang tepat karena kertas, kardus, tisu dan plastik walaupun sulit untuk terdekomposisi secara alami namun tergolong organik karena bahan bakunya berasal dari makhluk hidup.

## 3. METODOLOGI DAN PERANCANGAN

Pemodelan permainan Pilah Sampah yang dikembangkan dalam penelitian ini menggunakan  $\epsilon$ -NFA. Berikut ini adalah desain automata yang menjelaskan karakteristik dan keadaan-keadaan dari permainan “Pilah Sampah”.



**Gambar 1.** Desain  $\epsilon$ -NFA

Berikut ini merupakan deskripsi bahasa formal dari automata diatas:

**3.1. Himpunan State**

$$Q = \{trash\ es\ shifting, trash\ checking, add\ scores, reduce\ scores, unfiltered\ trash/game\ over\}$$

1. *Trashes shifting*: Pada *state* ini, sampah akan muncul dan mulai mengalir ke tepi selama sampah tersebut tidak di pilih. Ketika suatu jenis sampah pada *trashes shifting state* dipilih maka dedefinisikan sebagai *picked out input* dan akan melakukan transisi dari *trashes shifting state* ke *trash checking state*.
2. *Trash checking*: Setelah suatu sampah dipilih dari *trashes shifting state*, *player* akan memilih tempat pembuangan sampah yang sesuai, organik atau *non-organik*. Ketika sampah yang dipilih ditempatkan sesuai dengan tempatnya, maka didefinisikan sebagai *right place input* dan akan melakukan transisi dari *trash checking state* ke *add scores state*. Begitupun sebaliknya, ketika *player* menempatkan sampah tidak sesuai, maka akan didefinisikan sebagai *wrong place input* dan akan melakukan transisi dari *trash checking state* ke *reduce scores state*.
3. *Add scores*: Pada *state* ini, setiap penempatan sampah yang benar, *player scores* akan bertambah +25 poin. Selama permainan belum berhenti, maka dari *add scores state* akan terjadi transisi ke *trashes shifting state* dengan epsilon ( $\epsilon$ ) atau *empty string input*.
4. *Reduce scores*: Begitupun ketika *player* melakukan kesalahan dalam penempatan sampah, *player* akan diberi pengurangan *score* sebanyak -5 poin. Selama permainan belum berhenti, maka dari *add scores state* akan terjadi transisi ke *trashes shifting state* dengan *epsilon* ( $\epsilon$ ) atau *empty string input*.
5. *Unfiltered trash / game over*: Selama jangka waktu tertentu sampah tidak dipilih dari *trashes shifting state* maka akan terjadi transisi dari *trashes shifting state* ke *unfiltered trash / game over state* dengan epsilon ( $\epsilon$ ) atau *empty string input*.

### 3.2. Himpunan *Input*

$$\Sigma = \{\epsilon, \textit{picked out}, \textit{right place}, \textit{wrong place}\}$$

1. *Epsilon* ( $\epsilon$ ): Didefinisikan sebagai transisi antar *state* dengan tanpa *input*.
2. *Picked out*: Didefinisikan sebagai *input* ketika *player* telah memilih suatu jenis sampah di *trashes shifting state*.
3. *Right place*: Didefinisikan sebagai *input* ketika *player* melakukan penempatan sampah sesuai dengan jenisnya.
4. *Wrong place*: Didefinisikan sebagai *input* ketika *player* tidak melakukan penempatan sampah sesuai dengan jenisnya.

### 3.3. Fungsi Transisi

$$Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$$

Tabel 2. Transisi untuk setiap input

<i>State</i> \ <i>Input</i>	<i>Picked Out</i>	<i>Right Place</i>	<i>Wrong Place</i>	$\epsilon$
<i>Trashes Shifting</i>	{ <i>Trash Checking</i> }	$\emptyset$	$\emptyset$	{ <i>Game Over</i> }
<i>Trash Checking</i>	$\emptyset$	{ <i>Add Scores</i> }	{ <i>Reduce Scores</i> }	$\emptyset$
<i>Add Scores</i>	$\emptyset$	$\emptyset$	$\emptyset$	{ <i>Trashes Shifting</i> }
<i>Reduce Scores</i>	$\emptyset$	$\emptyset$	$\emptyset$	{ <i>Trashes Shifting</i> }
<i>Game Over</i>	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

### 3.4 *Initial State* dan *Final State*

$q_0$  : *Trashes shifting*

F : *Unfiltered trash / game over*

## 4. HASIL DAN PEMBAHASAN

Berdasarkan percobaan yang telah dilakukan didapatkan aplikasi permainan pilah sampah yang menyesuaikan model NFA yang telah dibuat. Berikut merupakan gambar dari aplikasi permainan yang dikembangkan.



**Gambar 2.** Keadaan awal

Pada Gambar 2 dapat dilihat bahwa keadaan awal dari *game* yaitu *conveyor* akan memindahkan sampah organik dan non-organik ke arah samping, ini merupakan *initial state Trashes Shifting*.



**Gambar 3.** Pengguna Memindahkan Sampah

Pada Gambar 3 menunjukkan ketika pengguna mengambil sebuah sampah dan memindahkannya. Ini merupakan salah satu *input Trash Picked Out* yang akan mengantarkan pada *state Trash Checking* dari *state Trashes Shifting*.



**Gambar 4.** Score Bertambah

Pada Gambar 4 dapat dilihat bahwa *score* pengguna dapat bertambah atau sistem berada di keadaan *Add Scores* jika setelah *Trash Checking*, masukan berada di posisi yang benar atau *input* merupakan *right place*.



**Gambar 5.** Score Minus



**Gambar 6.** Permainan Berakhir

Kemudian Gambar 6 merupakan keadaan di mana permainan berakhir jika selama *Trash Shifting* pengguna tidak melakukan apa-apa dan *input* akan berupa *epsilon* sehingga sampah akan terlewat.

Setelah aplikasi permainan ini dibangun, pengujian berupa blackbox testing dilakukan untuk menguji fungsionalitas dari aplikasi ini. Hasil dari pengujian ditunjukkan pada tabel 3.

Tabel 3. Hasil Pengujian Blackbox

No	Input	Kondisi yang diharapkan	Sesuai dengan kondisi?
1	Tanpa input	Permainan selesai atau skor berkurang	Ya
2	Sampah diletakkan di tempat yang benar	Skor bertambah	Ya
3	Sampah diletakkan di tempat yang salah	Skor berkurang	Ya

Dari hasil pengujian *black box*, didapatkan bahwa aplikasi permainan ini telah sesuai dengan desain automata yang dijadikan model bisnis.

## 5. KESIMPULAN DAN SARAN

Pengembangan aplikasi permainan dengan menggunakan pemodelan *finite state machine* (FSM) jenis  $\epsilon$ -NFA lebih cepat dan terstruktur. Hubungan dan transisi antar fungsional dalam aplikasi permainan dengan mudah dapat didefinisikan dengan transisi antar *state*, sehingga dalam proses pengembangan tidak terjadi kesalahan pendefinisian hubungan antar fungsional.

Pada paper ini telah ditunjukkan bagaimana pengembangan aplikasi permainan “Pilah Sampah” dengan pemodelan teori automata yaitu  $\epsilon$ -NFA. Penelitian selanjutnya dapat menggunakan jenis automata lainnya, misalnya *Push Down Automata* yang dapat menyimpan berapa sampah organik yang benar dimasukkan ke tempatnya. Selain itu, dari sisi pengembangan aplikasi permainan, kecerdasan buatan dapat ditambahkan pada desain permainan ini.

## UCAPAN TERIMA KASIH

Dengan selesainya penulisan paper ini, tim penulis mengucapkan terima kasih sebesar-besarnya kepada Ibu Ariana Yunita, M.I.T., MBA selaku dosen pengampu mata kuliah Teori Automata yang telah memberikan penulis kesempatan untuk menulis paper ini. Tidak lupa juga penulis ucapkan terima kasih kepada teman-teman kelas CS ‘16 yang senantiasa menemani dalam keadaan susah dan senang

## DAFTAR PUSTAKA

- [1] Damanhuri, E & Padmi, T. (2010). Diktat Pengelolaan Sampah (pp. 7). Bandung. ITB Press.
- [2] Hopcroft J. E. (2007). Introduction to automata theory, languages, and computation (3rd ed.). Boston: Greg Tobin.
- [3] Jamil, A., Engr. AsadUllah, & Rehman, M. (2016). An Infinite Runner Game Design using Automata Theory. In International Journal of Computer Science and Software Engineering (IJCSSE), 5(7), pp. 119-125.
- [4] Kozen D.C. (1977). Nondeterministic Finite Automata. In Automata and Computability (pp.25-31). [https://doi.org/10.1007/978-3-642-85706-5\\_5](https://doi.org/10.1007/978-3-642-85706-5_5)
- [5] Setyadi A.T., Kuswardayan I., & Hariadi R.R. (2018). Hotel Manajer: Permainan Simulasi Manajemen Operasi Hotel dengan Pemodelan Finite State Machine. Jurnal Teknik ITS, 7(1), pp. 10-14.
- [6] Undang Undang Republik Indonesia No. 18 Tahun 2008 Tentang Pengelolaan Sampah.
- [7] Wibowo, F.W. (2011). Finite State Machine Untuk Pengendali Elevator Berbasis Field Programmable Gate Array. Jurnal Dasi, (Vol. 12).