

TUGAS BESAR I
IF2124 TEORI BAHASA FORMAL DAN OTOMATA
THE SIMS SIMULATOR

LAPORAN TUGAS BESAR

Diajukan untuk memenuhi tugas mata kuliah IF2124 Teori Bahasa Formal dan Otomata

Oleh

KELOMPOK I/O

KELAS 01

INDRA FEBRIO NUGROHO

13518016

MUHAMMAD FAUZAN AL-GHIFARI

13518112



INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2019

DAFTAR ISI

BAB I DASAR TEORI.....	1
1.1 Pengertian DFA.....	1
1.2 Ciri-Ciri DFA	2
1.3 Cara Kerja DFA	2
BAB II TEORI DASAR	3
2.1 Menentukan State DFA.....	3
2.2 Menentukan Alphabet pada DFA	6
2.3 Menyusun Tabel Transisi.....	7
BAB III IMPLEMENTASI DAN PENGUJIAN	9
3.1 Spesifikasi dan Teknis Program.....	9
3.2 Capture Program	17

BAB I

DASAR TEORI

1.1 Pengertian DFA

Deterministic Finite Automata (DFA) adalah mesin abstrak berupa sistem model matematika dengan masukan dan keluaran diskrit yang dapat mengenali bahasa paling sederhana (bahasa regular) dan dapat diimplementasikan secara nyata dimana sistem dapat berada di salah satu dari beberapa konfigurasi internal yang disebut *state*. Sistem *state* merupakan ringkasan informasi yang berkaitan dengan beberapa input sebelumnya yang diperlukan untuk menentukan perilaku sistem pada inputan selanjutnya.

Deterministic Finite Automata adalah :

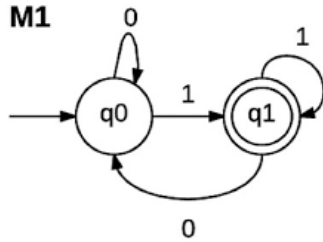
- Mesin yang terdiri dari *finite number of State*.
- Salah satu sebagai initial state.
- Minimal satu accepted state.
- Mesin akan menerima input stream berupa symbol / alphabet yang datang secara sequential.
- Mesin akan berubah dari state satu ke state lain berdasar simbol input dan current state.
- Deterministic artinya tidak diperkenankan ambigu.
- Finite artinya mesin terdiri dari state yang finite.
- Esensi DFA adalah recognizer string (menerima atau menolak).
- Mungkin saja di DFA menerima string kosong. Jika hal ini terjadi, maka initial state adalah accepted state.

Sebuah finite automata didefinisikan dalam 5-tuple $(Q, \Sigma, \delta, q_0, F)$ dimana :

1. Q adalah himpunan terbatas dari states,
2. Σ himpunan terbatas alphabet,
3. $\delta: Q \times \Sigma \rightarrow Q$ fungsi transisi, dinotasikan ke $\delta(q,a) = p$
4. $q_0 \in Q$ adalah start state, dan
5. $F \subseteq Q$ adalah himpunan accepted states (atau final states).

Deterministic Finite Automata (DFA)

M1



• *Example 1: M1*

• This DFA is a 5-tuple: $(Q, \Sigma, \delta, q_0, F)$

1. Q = set of states = $\{q_0, q_1\}$
2. Σ = alphabet = $\{0, 1\}$
3. δ = transition function = (state diagram/table)
4. q_0 = first state = q_0

1.2 Ciri-Ciri DFA

Model Deterministic Finite Automata memiliki ciri-ciri:

1. Tidak memiliki memori sementara
2. Head hanya bergerak 1 arah
3. Hanya berisi memori masukan berupa tape berisi string masukan dan sejumlah kendali berhingga.

1.3 Cara Kerja DFA

Finite Automata bekerja dengan cara mesin membaca memori masukan berupa tape yaitu 1 karakter tiap saat (dari kiri ke kanan) menggunakan head baca yang dikendalikan oleh kotak kendali state berhingga dimana pada mesin terdapat sejumlah state berhingga.

Finite Automata selalu dalam kondisi yang disebut state awal (initial state) pada saat Finite Automata mulai membaca tape. Perubahan state terjadi pada mesin ketika sebuah karakter berikutnya dibaca.

Ketika head telah sampai pada akhir tape dan kondisi yang ditemui adalah state akhir, maka string yang terdapat pada tape dikatakan diterima Finite Automata (String-string merupakan milik bahasa bila diterima Finite Automata bahasa tersebut).

BAB II

ANALISIS PERSOALAN

2.1 Menentukan State DFA

Pada DFA ini satu state terdiri dari tiga buah atribut yaitu :

- a. Hygiene : tingkat kebersihan pemain
- b. Energy : tingkat energy yang dimiliki pemain
- c. Fun : tingkat kesenangan dari pemain

Setiap atribut memiliki nilai maksimum 15 dan nilai minimum 0. Setiap atribut bertambah (sampai nilai maksimum) atau berkurang (sampai nilai minimum) sejumlah kelipatan 5. Berarti setiap atribut memiliki 4 kemungkinan nilai yaitu 0, 5, 10 dan 15.

Menentukan jumlah state pada DFA.

Satu state pada DFA terdiri dari 3 atribut dan setiap atribut memiliki 4 kemungkinan nilai, sehingga jumlah state pada DFA ini adalah $4 \times 4 \times 4 = 64$ buah state.

Menentukan Set of States dari DFA (Q)

DFA ini memiliki 64 buah states, kami memutuskan untuk memberi nama states dengan huruf sebagai berikut

$Q = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, \\ AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, \\ AS, AT, AU, AV, AW, AX, AY, AZ, BA, BB, BC, BD, BE, BF, BG, BH, BI, BJ, \\ BK, BL, BM\}$

Setiap state memiliki nilai atribut yang berbeda sesuai dengan tabel di bawah ini

State			
Nama	Hygiene	Energy	Fun
A	0	10	0
B	0	0	5
C	0	0	10
D	0	0	15
E	0	5	0
F	0	5	5
G	0	5	10

H	0	5	15
I	0	10	5
J	0	10	10
K	0	10	15
L	0	15	0
M	0	15	5
N	0	15	10
O	0	15	15
Q	5	0	0
R	5	0	5
S	5	0	10
T	5	0	15
U	5	5	0
V	5	5	5
W	5	5	10
X	5	5	15
Y	5	10	0
Z	5	10	5
AA	5	10	10
AB	5	10	15
AC	5	15	0
AD	5	15	5
AE	5	15	10
AF	5	15	15
AG	10	0	0
AH	10	0	5
AI	10	0	10
AJ	10	0	15
AK	10	5	0
AL	10	5	5
AM	10	5	10
AN	10	5	15
AO	10	10	0
AP	10	10	5
AQ	10	10	10
AR	10	10	15
AS	10	15	0
AT	10	15	5
AU	10	15	10
AV	10	15	15
AW	15	0	0
AX	15	0	5
AY	15	0	10
AZ	15	0	15

BA	15	5	0
BB	15	5	5
BC	15	5	10
BD	15	5	15
BE	15	10	0
BF	15	10	5
BG	15	10	10
BH	15	10	15
BI	15	15	0
BJ	15	15	5
BK	15	15	10
BL	15	15	15
BM	0	0	0

Menentukan Start State dan Final State

Sesuai dengan deskripsi soal. Kondisi awal pemain selalu dalam keadaan sudah bangun tidur dengan rincian atribut sebagai berikut

- Hygiene : 0
- Energy : 10
- Fun : 0

Ketiga atribut di atas ada pada state A

State A adalah start state.

Sesuai dengan deskripsi soal. Permainan dinyatakan selesai jika semua atribut bernilai 0 atau semua atribut bernilai 15. Semua atribut bernilai 0 saat mencapai state BM dan semua atribut bernilai 15 saat mencapai state BL.

State BM dan BL adalah Final state.

2.2 Menentukan Alphabet pada DFA

Sesuai dengan deskripsi soal. DFA ini memiliki 18 alphabet aksi.

$\Sigma = \{\text{Tidur Siang, Tidur Malam, Makan Hamburger, Makan Pizza, Makan Steak and Beans, Minum Air, Minum Kopi, Minum Jus, Buang Air Kecil, Buang Air Besar, Bersosialisasi ke Kafe, Bermain Media Sosial, Bermain komputer, Mandi, Cuci Tangan, Mendengarkan Musik di Radio, Membaca Koran, Membaca Novel}\}$

Setiap aksi yang dilakukan akan mengubah nilai current state sesuai dengan tabel di bawah ini

Aksi	Aksi	Konsekuensi		
		Hygiene	Energy	Fun
1	Tidur Siang	-	+10	-
2	Tidur Malam	-	+15	-
3	Makan Hamburger	-	+5	-
4	Makan Pizza	-	+10	-
5	Makan Steak and Beans	-	+15	-
6	Minum Air	-5	-	-
7	Minum Kopi	-10	+5	-
8	Minum Jus	-5	+10	-
9	Buang Air Kecil	+5	-	-
10	Buang Air Besar	+10	-5	-
11	Bersosialisasi ke Kafe	-5	-10	+15
12	Bermain Media Sosial	-	-10	+10
13	Bermain Komputer	-	-10	+15
14	Mandi	+15	-5	-
15	Cuci Tangan	+5	-	-
16	Mendengarkan Musik di Radio	-	-5	+15
17	Membaca Koran	-	-5	+5
18	Membaca Novel	-	-5	+10

2.3 Menyusun Tabel Transisi

	State				Tidur		Makan			Minum			Buang Air	
	Nama	Hygiene	Energy	Fun	Siang	Malam	Hamburger	Pizza	Steak and Beans	Air	Kopi	Jus	Kecil	Besar
					E+10	E+15	E+5	E+10	E+15	H-5	H-10, E+5	H-5, E+10	H+5	H+10, E-5
→	A	0	10	0										
	B	0	0	5										
	C	0	0	10										
	D	0	0	15										
	E	0	5	0										
	F	0	5	5										
	G	0	5	10										
	H	0	5	15										
	I	0	10	5										
	J	0	10	10										
	K	0	10	15										
	L	0	15	0										
	M	0	15	5										
	N	0	15	10										
	O	0	15	15										
	P	5	0	0										
	Q	5	0	5										
	R	5	0	10										
	S	5	0	15										
	T	5	5	0										
	U	5	5	5										
	V	5	5	10										
	W	5	5	15										
	X	5	10	0										
	Y	5	10	5										
	Z	5	10	10										
	AA	5	10	15										

Buat tabel transisi kosong yang memuat baris States (dari state A sampai state BM) dan kolom Alphabet (dari tidur siang sampai membaca novel) . Tabel di atas adalah sedikit cuplikan dari Tabel transisi.

Untuk mengisi tabel transisi di baris States A dan kolom Tidur Siang, kelompok kami menggunakan kode dengan bahasa python di bawah. Untuk kolom-kolom yang lain kode disesuaikan dengan kondisi konsekuensi yang dihasilkan dari aksi yang dilakukan (tergantung +/- dari atribut Hygiene, Energy dan Fun).

```
import pandas as pd
import numpy as np
from pandas import ExcelWriter
from pandas import ExcelFile

df = pd.read_excel("DFA.xlsx")

for j in range (3, 66):
    if ((int(df.iloc[j,3]) + 10) >= 20):
        df.iloc[j,5] = df.iloc[j,1]
    else:
        for k in range (3,66):
            if (int(df.iloc[k,3]) == int(df.iloc[j,3] + 10) and int(df.
iloc[k,2] == int(df.iloc[j,2]) and int(df.iloc[k,4] == int(df.iloc[j,4]
)))) :
                df.iloc[j,5] = df.iloc[k,1]

print(df.head())
writer = pd.ExcelWriter('DFA.xlsx')
df.to_excel(writer,'Sheet1')
writer.save()
```

	State Nama	Hygiene	Energy	Fun	Tidur Siang	Malam	Makan Hamburger	Pizza	Sak and Be	Minum Air	Kopi	Jus	Buang Air Kecil
→	A	0	10	0	-								
	B	0	0	5	I								
	C	0	0	10	J								
	D	0	0	15	K								
	E	0	5	0	L								
	F	0	5	5	M								
	G	0	5	10	N								
	H	0	5	15	O								
	I	0	10	5	-								
	J	0	10	10	-								
	K	0	10	15	-								
	L	0	15	0	-								
	M	0	15	5	-								
	N	0	15	10	-								
	O	0	15	15	-								
	Q	5	0	0	Y								
	R	5	0	5	Z								
	S	5	0	10	AA								
	T	5	0	15	AB								

Kode akan mencari state yang sesuai dengan aksi yang dilakukan, jika atribut dari state melebihi 15 atau kurang dari 0 maka tabel akan diisi dengan statenya sendiri. Kode diulangi sebanyak 18 kali dan disesuaikan dengan kondisi alphabet dan state sampai semua tabel terisi

	State				Tidur		Makan		
	Nama	Hygiene	Energy	Fun	Siang	Malam	Hamburger	Pizza	Steak and Beans
					E+10	E+15	E+5	E+10	E+15
→	A	0	10	0	A	A	L	A	A
	B	0	0	5	I	M	F	A	M
	C	0	0	10	J	N	G	J	N
	D	0	0	15	K	O	H	K	O
	E	0	5	0	L	E	A	L	E
	F	0	5	5	M	F	I	M	F
	G	0	5	10	N	G	J	N	G
	H	0	5	15	O	H	K	O	H
	I	0	10	5	I	I	M	I	I
	J	0	10	10	J	J	N	J	J
	K	0	10	15	K	K	O	K	K
	L	0	15	0	L	L	L	L	L
	M	0	15	5	M	M	M	M	M
	N	0	15	10	N	N	N	N	N
	O	0	15	15	O	O	O	O	O
	Q	5	0	0	Y	AC	U	Y	AC
	R	5	0	5	Z	AD	V	Z	AD
	S	5	0	10	AA	AE	W	AA	AE
	T	5	0	15	AB	AF	X	AB	AF
	U	5	5	0	AC	U	Y	AC	U
	V	5	5	5	AD	V	Z	AD	V
	W	5	5	10	AE	W	AA	AE	W
	X	5	5	15	AF	X	AB	AF	X
	Y	5	10	0	Y	Y	AC	Y	Y
	Z	5	10	5	Z	Z	AD	Z	Z

BAB III

IMPLEMENTASI DAN PENGUJIAN

3.1 Spesifikasi dan Teknis Program

Program pada kelompok kami terbagi menjadi 4 buah file

- a. boolean.h, berisi definisi dari type boolean
- b. sims.h, berisi ADT SIMS
- c. sims.c, berisi implementasi dari ADT SIMS
- d. main.c, realisasi dari main program

Isi boolean.h

```
/* Definisi type boolean */

#ifndef _BOOLEAN_h
#define _BOOLEAN_h

#define boolean unsigned char
#define true 1
#define false 0

#endif
```

Fungsi dan Procedure pada sims.h

```
// File : sims.h
// DESKRIPSI ABSTRACT DATA TYPE SIMS

#ifndef SIMS_H
#define SIMS_H

#include "boolean.h"

#define ValMin 0
#define ValMax 15

//TIPE BENTUKAN
typedef struct {
    int Hygiene;
    int Energy;
    int Fun;
} SIMS;
```

```

//SELEKTOR
#define Hygiene(S) (S).Hygiene
#define Energy(S) (S).Energy
#define Fun(S) (S).Fun

void Start (SIMS *s);
// Menentukan initial state Sims saat bangun tidur,
// yaitu Hygiene=0, Energy=10, Fun=0.

boolean IsFinal (SIMS s);
// Menghasilkan true jika semua atribut bernilai ValMin
// atau semua atribut bernilai ValMax.

void ShowStats (SIMS s);
// Mencetak ke layar status dari SIMS saat ini.

boolean IsHygieneValid (SIMS s, int x, char op);
// Menghasilkan true apabila nilai hygiene awal
// ditambah/dikurang x valid, yaitu ValMin<=(Hygiene(S) +- x) <= ValMax

boolean IsEnergyValid (SIMS s, int x, char op);
// Menghasilkan true apabila nilai energy awal
// ditambah/dikurang x valid, yaitu ValMin<= (Energy(S) +- x) <= ValMax

boolean IsFunValid (SIMS s, int x, char op);
// Menghasilkan true apabila nilai fun awal
// ditambah/dikurang x valid, yaitu ValMin <= (Fun(S) +- x) <= ValMax

void Nap (SIMS *s);
// Menambah 10 Energy apabila nilai Energy valid.
// Mengeluarkan pesan error apabila nilai Energy tidak valid.

void Sleep (SIMS *s);
// Menambah 15 Energy apabila nilai Energy valid.
// Mengeluarkan pesan error apabila nilai Energy tidak valid.

void EatHamburger (SIMS *s);
// Menambah 5 Energy apabila nilai Energy valid.
// Mengeluarkan pesan error apabila nilai Energy tidak valid.

void EatPizza (SIMS *s);
// Menambah 10 Energy apabila nilai Energy valid.
// Mengeluarkan pesan error apabila nilai Energy tidak valid.

```

```
void EatSteaknBeans (SIMS *s);  
// Menambah 15 Energy apabila nilai Energy valid.  
// Mengeluarkan pesan error apabila nilai Energy tidak valid.
```

```
void DrinkWater (SIMS *s);  
// Mengurangi 5 Hygiene apabila nilai Hygiene valid.  
// Mengeluarkan pesan error apabila nilai Hygiene tidak valid.
```

```
void DrinkCoffee (SIMS *s);  
// Mengurangi 10 Hygiene dan Menambah 5 Energy apabila nilai Hygiene da  
n Energy valid.  
// Mengeluarkan pesan error apabila nilai Hygiene dan Energy tidak vali  
d.
```

```
void DrinkJuice (SIMS *s);  
// Mengurangi 5 Hygiene dan Menambah 10 Energy apabila nilai Hygiene da  
n Energy valid.  
// Mengeluarkan pesan error apabila nilai Hygiene dan Energy tidak vali  
d.
```

```
void PeePee (SIMS *s);  
// Menambah 5 Hygiene apabila nilai Hygiene valid.  
// Mengeluarkan pesan error apabila nilai Hygiene tidak valid.
```

```
void PooPoo (SIMS *s);  
// Menambah 10 Hygiene dan mengurangi 5 Energy apabila nilai Hygiene da  
n Energy valid.  
// Mengeluarkan pesan error apabila nilai Hygiene dan Energy tidak vali  
d.
```

```
void GoToCafe (SIMS *s);  
// Menambah 15 Fun, mengurangi 10 Energy, dan mengurangi 5 Hygiene apab  
ila ketiga nilai tersebut valid.  
// Mengeluarkan pesan error apabila ketiga nilai tersebut tidak valid.
```

```
void SocMed (SIMS *s);  
// Menambah 10 Fun dan mengurangi 10 Energy apabila kedua nilai tersebu  
t valid.  
// Mengeluarkan pesan error apabila kedua nilai tersebut tidak valid.
```

```
void PlayComputer (SIMS *s);  
// Menambah 15 Fun dan mengurangi 10 Energy apabila kedua nilai tersebut valid.  
// Mengeluarkan pesan error apabila kedua nilai tersebut tidak valid.
```

```
void TakeAShower (SIMS *s);  
// Menambah 15 Hygiene dan mengurangi 5 Energy apabila kedua nilai tersebut valid.  
// Mengeluarkan pesan error apabila kedua nilai tersebut tidak valid.
```

```
void WashHands (SIMS *s);  
// Menambah 5 Hygiene apabila nilai tersebut valid.  
// Mengeluarkan pesan error apabila nilai tersebut tidak valid.
```

```
void ListenToRadio (SIMS *s);  
// Menambah 10 Fun dan mengurangi 5 Energy apabila kedua nilai tersebut valid.  
// Mengeluarkan pesan error apabila kedua nilai tersebut tidak valid.
```

```
void ReadNewspaper (SIMS *s);  
// Menambah 5 Fun dan mengurangi 5 Energy apabila kedua nilai tersebut valid.  
// Mengeluarkan pesan error apabila kedua nilai tersebut tidak valid.
```

```
void ReadNovel (SIMS *s);  
// Menambah 10 Fun dan mengurangi 5 Energy apabila kedua nilai tersebut valid.  
// Mengeluarkan pesan error apabila kedua nilai tersebut tidak valid.
```

```
#endif // !SIMS_H
```

Main Program

```
#include <stdio.h>
#include "sims.h"
#include <string.h>

int main () {
    //Kamus
    SIMS s;
    char inwords[100];
    char confirm[100];
    boolean IsFinish = false;
    boolean IsExit = false;
    boolean IsConfirm;

    Start(&s);
    ShowStats(s); printf("\n");

    while (!(IsExit)) {
        while (!IsFinish) {
            IsConfirm = false;
            printf("What will you do? (type answer in Bahasa)\n");
            printf(">>> ");
            scanf("%[^\n]*c", inwords);
            if (strcmp(inwords, "Tidur Siang") == 0) {
                Nap(&s);
            }
            else if (strcmp(inwords, "Tidur Malam") == 0) {
                Sleep(&s);
            }
            else if (strcmp(inwords, "Makan Hamburger") == 0){
                EatHamburger(&s);
            }
            else if (strcmp(inwords, "Makan Pizza") == 0) {
                EatPizza(&s);
            }
            else if (strcmp(inwords, "Makan Steak and Beans") == 0) {
                EatSteaknBeans(&s);
            }
            else if (strcmp(inwords, "Minum Air") == 0) {
                DrinkWater(&s);
            }
            else if (strcmp(inwords, "Minum Kopi") == 0) {
                DrinkCoffee(&s);
            }
            else if (strcmp(inwords, "Minum Jus") == 0) {
                DrinkJuice(&s);
            }
            else if (strcmp(inwords, "Buang Air Kecil") == 0) {
```



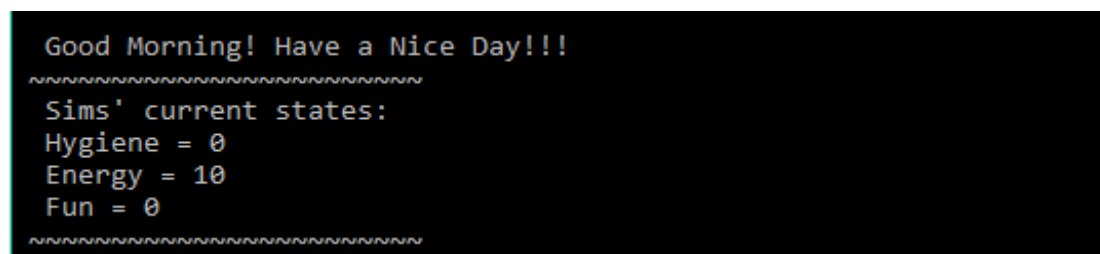
```

while (!(IsConfirm)){
    printf("Do You Want to continue? (Yes/No)\n");
    printf(">>> ");
    scanf("%[^\n]*c", confirm);
    if (strcmp(confirm, "Yes") == 0) {
        IsConfirm = true;
        IsFinish = false;
        ShowStats(s);
        printf("\n");
    }
    else if (strcmp(confirm, "No") == 0) {
        IsConfirm = true;
        IsExit = true;
        printf("See you next time!!!\n");
    }
    else{
        printf("Invalid Input\n\n");
    }
}
return 0;
}

```

3.2 Capture Program

Saat program baru dijalankan



Saat aksi tidak valid karena atribut melebihi 15 atau kurang dari 0

```
What will you do? (type answer in Bahasa)
>>> Minum Air
Aksi tidak valid
```

Saat aksi valid

```
What will you do? (type answer in Bahasa)
>>> Mandi
~~~~~
Status Sims saat ini adalah
Hygiene = 15
Energy = 5
Fun = 0
~~~~~
```

Saat input yang dimasukkan salah

```
What will you do? (type answer in Bahasa)
>>> Minum GoMilk
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!! You put the wrong input. Try again !!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Saat aksi mencapai final state 0 0 0

```

What will you do? (type answer in Bahasa)
>>> Minum Air

Sims' current states:
Hygiene = 0
Energy = 0
Fun = 0

TUBES T B F O
* 18/09/19
+ 02/10/19

0|0
|/.00| -< @000
`'0`@| v |@00000
.000000|0 | 0 @0000000000
_.000000000000_ /0' _00000000000000000000

You have finished your game.
Do You want to continue? (Yes/No)
>>>

```

User dapat memilih untuk melanjutkan string atau tidak.

Jika user memilih untuk tidak melanjutkan string, permainan akan berakhir.

```
You have finished your game.  
Do You Want to continue? (Yes/No)  
>>> No  
See you next time!!!
```

Jika user memilih untuk melanjutkan string, maka permainan akan dilanjutkan dan user dapat memilih aksi selanjutnya.

```
You have finished your game.
Do You Want to continue? (Yes/No)
>>> Yes

~~~~~
Sims' current states:
Hygiene = 0
Energy = 0
Fun = 0
~~~~~

What will you do? (type answer in Bahasa)
>>>
```

Saat aksi mencapai final state 15 15 15

[illegible]

User dapat memilih untuk melanjutkan string atau tidak.

Jika user memilih untuk tidak melanjutkan string, permainan akan berakhir.

```
You have finished your game.  
Do You Want to continue? (Yes/No)  
>>> No  
See you next time!!!
```

Jika user memilih untuk melanjutkan string, maka permainan akan dilanjutkan dan user dapat memilih aksi selanjutnya.

```
You have finished your game.  
Do You Want to continue? (Yes/No)  
>>> Yes  
~~~~~  
Sims' current states:  
Hygiene = 15  
Energy = 15  
Fun = 15  
~~~~~  
What will you do? (type answer in Bahasa)  
>>>
```