**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 Aplhabet pada DFA 6

2.3 Menyusun Tabel Transisi 7

BAB III IMPLEMENTASI DAN PENGUJIAN 9

3.1 Spesifikasi dan Teknis Program 9

3.2 Kasus – Kasus 17

**BAB I**

**DASAR TEORI**

* 1. **Pengertian DFA**

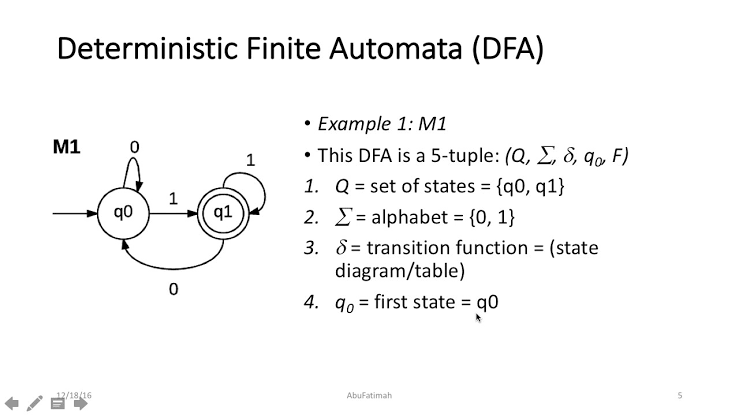
Deterministic Finite Automata (DFA) adalah mesin abstrak berupa sistem model matematika dengan masukan dan keluaran diskrit yang dapat mengenali bahasa paling sederhana (bahasa regular) dan dapat diimplementasikan secara nyata dimana sistem dapat berada di salah satu dari beberapa konfigurasi internal 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,Σ, δ,q0,F) dimana :

1. Q adalah himpunan terbatas dari states,
2. Σ himpunan terbatas alphabet,
3. δ: Q × Σ = Q fungsi transisi, dinotasikan ke δ(q,a) = p
4. q0 ∈ Q adalah start state, dan
5. F⊆Q adalah himpunan accepted states (atau final states).



* 1. **Ciri-Ciri DFA**

Model Dinite 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. **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 :

* 1. Hygiene : tingkat kebersihan pemain
  2. Energy : tingkat energy yang dimiliki pemain
  3. 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 x 4 x 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

* + 1. Hygiene : 0
    2. Energy : 10
    3. 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**

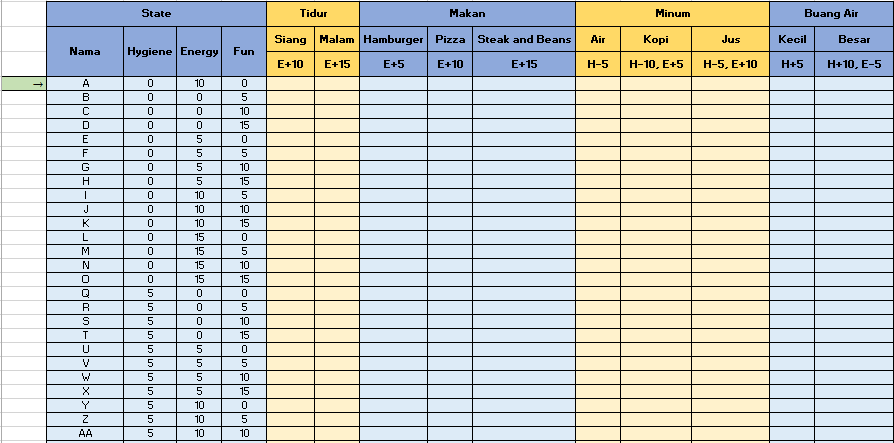
Sesuai dengan deskripsi soal. DFA ini memiliki 18 alphabet aksi.

∑ = {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**



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] = '-'

    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])))) :

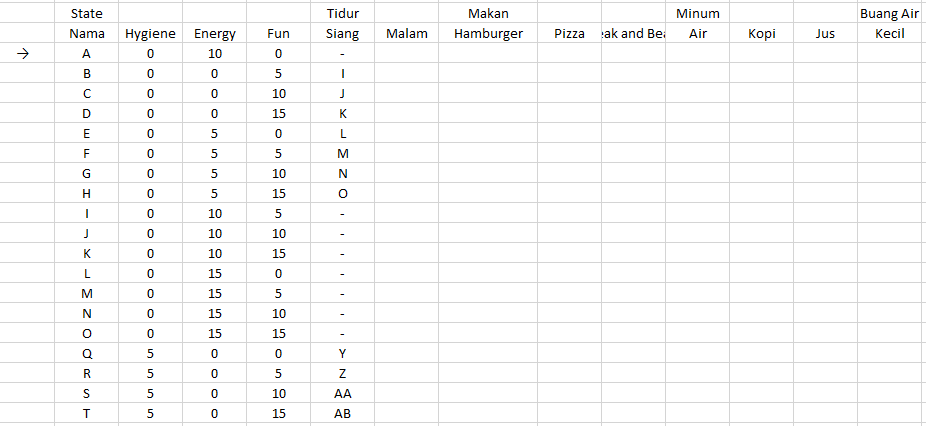
                df.iloc[j,5] = df.iloc[k,1]

print(df.head())

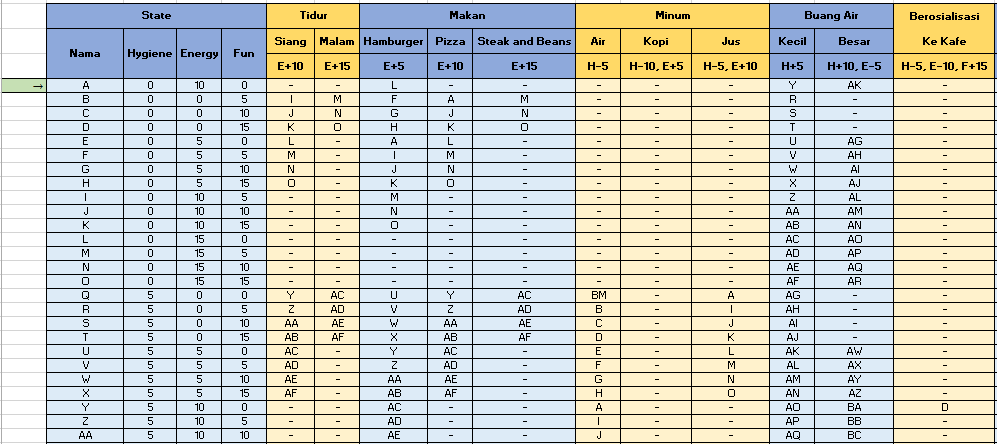
writer = pd.ExcelWriter('DFA.xlsx')

df.to\_excel(writer,'Sheet1')

writer.save()



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 “–“. Kode diulangi sebanyak 18 kali dan disesuaikan dengan kondisi alphabet dan state sampai semua tabel terisi



**BAB III**

**IMPLEMENTASI DAN PENGUJIAN**

**3.1 Spesifikasi dan Teknis Program**

Program pada kelompok kami terbagi menjadi 4 buah file

1. boolean.h, berisi defenisi dari type boolean
2. sims.h, berisi ADT SIMS
3. sims.c, berisi implementasi dari ADT SIMS
4. 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 dan Energy valid.*

*// Mengeluarkan pesan error apabila nilai Hygiene dan Energy tidak valid.*

void DrinkJuice (SIMS \*s);

*// Mengurangi 5 Hygiene dan Menambah 10 Energy apabila nilai Hygiene dan Energy valid.*

*// Mengeluarkan pesan error apabila nilai Hygiene dan Energy tidak valid.*

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 dan Energy valid.*

*// Mengeluarkan pesan error apabila nilai Hygiene dan Energy tidak valid.*

void GoToCafe (SIMS \*s);

*// Menambah 15 Fun, mengurangi 10 Energy, dan mengurangi 5 Hygiene apabila 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 tersebut 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;

    boolean IsConfirm;

    Start(&s);

    ShowStats(s); printf("\n");

    while (!(IsExit)){

        while (!IsFinish) {

            IsConfirm = false;

            IsExit = 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) {

                PeePee(&s);

            }

            else if (strcmp(inwords,"Buang Air Besar") == 0) {

                PooPoo(&s);

            }

            else if (strcmp(inwords,"Bersosialisasi ke Kafe") == 0) {

                GoToCafe(&s);

            }

            else if (strcmp(inwords, "Bermain Media Sosial") == 0) {

                SocMed(&s);

            }

            else if (strcmp(inwords, "Bermain komputer") == 0) {

                PlayComputer(&s);

            }

            else if (strcmp(inwords, "Mandi") == 0) {

                TakeAShower(&s);

            }

            else if (strcmp(inwords, "Cuci Tangan") == 0) {

                WashHands(&s);

            }

            else if (strcmp(inwords,"Mendengarkan Musik di Radio") == 0) {

                ListenToRadio(&s);

            }

            else if (strcmp(inwords,"Membaca Koran") == 0) {

                ReadNewspaper(&s);

            }

            else if (strcmp(inwords, "Membaca Novel") == 0) {

                ReadNovel(&s);

            }

            else {

                printf("  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n\n");

                printf("  !!! You put the wrong input. Try again !!!\n\n");

                printf("  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n\n");

            }

            if(IsFinal(s)){

                IsFinish = true;

            }

        }

        if (IsFinish) {

            if (Hygiene(s) == 15) {

                printf("                       /^\'                             \n");

                printf("            /         (/^\')     /                      \n");

                printf("       \'   ( \'         \' /     ( \'     /^\'             \n");

                printf("      / )   \' |        \_|\_      \' |   |/^\'|            \n");

                printf("     | /    \_|\_        | |      \_|\_    \' /             \n");

                printf("     \_|\_    | |        | |      | |    \_|\_             \n");

                printf("     | |    | |        | |      | |    | |             \n");

                printf("     | |    | |    \*\*\*\*| |\*\*\*\*\*\*| |    | |             \n");

                printf("     | |\*\*\*\*| |\*\*\*\*    | |      | |\*\*\*\*| |             \n");

                printf("    \*| |    | |                 | |    | |\*\*\*\*\*        \n");

                printf("  \*  | |   Y O U   W I N               | |      \*      \n");

                printf(" \*                                               \*     \n");

                printf(" | \*            Y O U   W I N             !      \* |   \n");

                printf(" |  \*\*\*\*\*                                 \*\*\*\*\*  |     \n");

                printf(" |@      \*\*\*\*\*\*\*\*\*\*             \*\*\*\*\*\*\*\*\*\*      @|     \n");

                printf(" | @   @           \*\*\*\*\*\*\*\*\*\*\*\*\*           @   @ |     \n");

                printf(" |  @@@ @    @                       @    @ @@@  |     \n");

                printf(" |       @@@@ @      @       @      @ @@@@       |     \n");

                printf("  \*            @@@@@@ @     @ @@@@@@            \*      \n");

                printf("   \*                   @@@@@                   \*       \n");

                printf("    \*\*\*\*\*                                 \*\*\*\*\*        \n");

                printf("         \*\*\*\*\*\*\*\*\*\*             \*\*\*\*\*\*\*\*\*\*             \n");

                printf("                   \*\*\*\*\*\*\*\*\*\*\*\*\*                       \n\n\n");

            }

            else {

               printf("                    \_\_\_\_\_  \_\_\_\_\_                        \n");

               printf("                   <     `/     |                       \n");

               printf("                    >          (                        \n");

               printf("                   |   \_     \_  |                       \n");

               printf("                   |  |\_) | |\_) |                       \n");

               printf("                   |  | \' | |   |                       \n");

               printf("                   |            |                       \n");

               printf("    \_\_\_\_\_\_.\_\_\_\_\_\_\_\_|            |\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_      \n");

               printf("  \_/                                       \'|     |     \n");

               printf(" |               T U B E S   T B F O               <     \n");

               printf(" |\_\_\_\_\_.-.\_\_\_\_\_\_\_\_\_              \_\_\_\_/|\_\_\_\_\_\_\_\_\_\_\_|     \n");

               printf("                   | \* 18/09/19 |                       \n");

               printf("                   | + 02/10/19 |                       \n");

               printf("                   |            |                       \n");

               printf("                   |            |                       \n");

               printf("                   |   \_        <                       \n");

               printf("                   |\_\_/         |                       \n");

               printf("                    / `--.      |                       \n");

               printf("                  0|            |0                      \n");

               printf("              |/.00|          -< @000                   \n");

               printf("              `\'0`@|     v      |@@0@00                 \n");

               printf("            .000@@@|0    |    0 @@@00@0000              \n");

               printf("       \_.000000@@@@@@00\_/0\'\_0@@00@@@@@@@000000          \n\n\n");

            }

        }

        printf("You have finished your game. \nDo You Want to continue this game? (Yes/No)\n");

        printf(">>> ");

        while (!(IsConfirm)){

            scanf("%[^\n]%\*c", confirm);

            if (strcmp(confirm, "Yes") == 0) {

                IsExit = false;

                IsConfirm = true;

                IsFinish = false;

                ShowStats(s);

                printf("\n");

            }

            else if (strcmp(confirm, "No") == 0) {

                IsExit = true;

                IsConfirm = true;

                IsFinish = true;

                printf("See you next time!!!\n");

            }

            else{

                printf("Input Yes/No only, please\n");

                printf(">>> ");

            }

        }

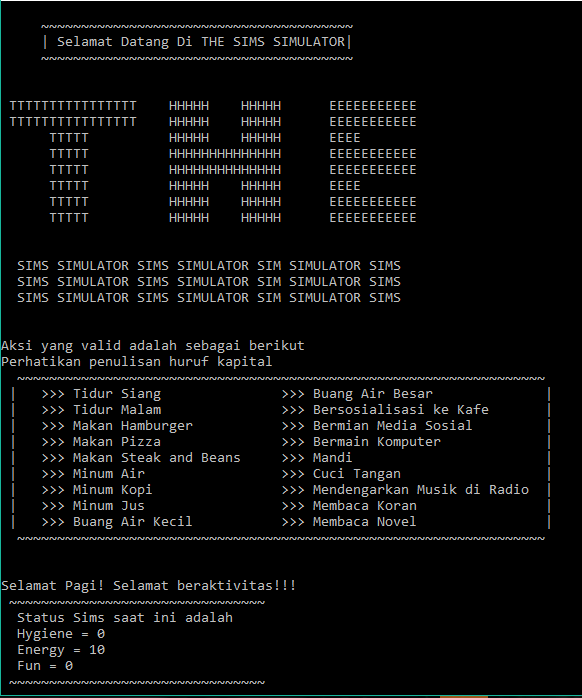
    }

    return 0;

}

**3.2 Kasus-Kasus**

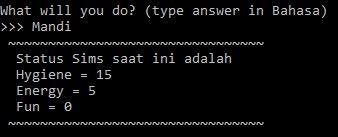
Saat program baru dijalalankan



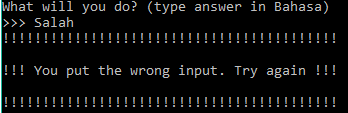
Saat aksi tidak valid



Saat aksi valid



Saat input yang dimasukkan salah



Saat program mencapai finish

