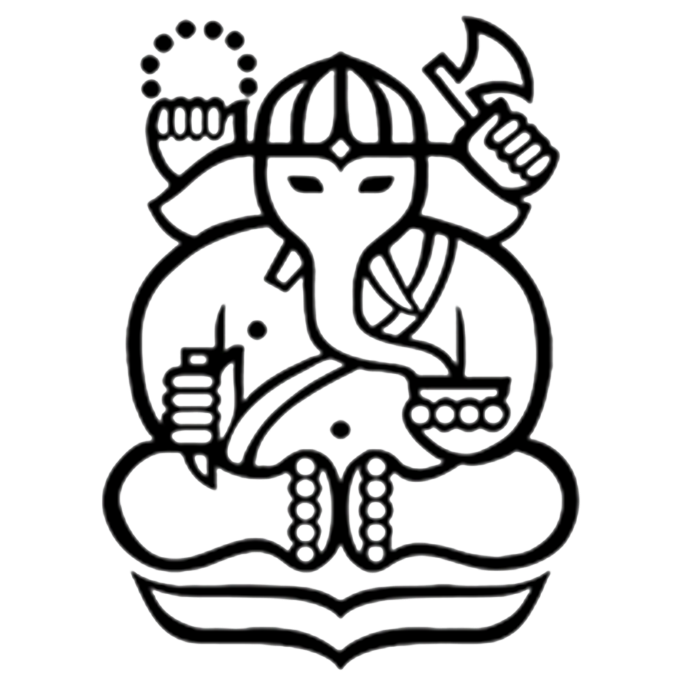
**Tugas Besar 1 IF 2124 Teori Bahasa Formal dan Otomata**

**The Sims Simulator**



**Disusun Oleh :**

Radhinansyah Hemsa Ghaida 13518087

Gregorius Jovan Kresnadi 13518135

**Teknik Informatika**

**Institut Teknologi Bandung**

**Semester Ganjil**

**Tahun 2019/2020**

**BAB I**

**TEORI DASAR**

1. Pengertian Deterministic Finite Automata

Deterministic Finite Automata (DFA) adalah mesin abstrak berupa sistem model matematika dengan masukan dan keluaran diskrit yang dapat mengenali bahasa paling sederhana (bahasa reguler) dan dapat diimplementasikan secara nyata di mana sistem dapat berada di salah satu dari beberapa konfigurasi internal yang disebut state. DFA menggunakan suatu model komputer dengan jumlah memori yang sangat terbatas.

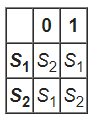
1. Definisi formal dari Deterministic Finite Automata

Sebuah Finite Automata (DFA) didefinisikan dalam 5 pasangan variabel (Q, ∑, δ, q0, dan F), di mana:

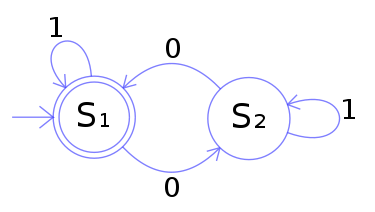
1. Q adalah himpunan terbatas dari states.
2. ∑ adalah himpunan terbatas dari alphabet.
3. δ: Q × ∑ -> Q adalah fungsi transisi dinotasikan sebagai δ(q,a) -> p yang berarti jika state q diberikan string a maka akan berpindah state menjadi p.
4. q 0 ∈ adalah start state.
5. F ⊆ Q adalah himpunan accepted states atau bisa juga disebut final states di mana states ini berisi states yang diterima oleh sistem tersebut (komputer).
6. Ciri – ciri sebuah DFA
7. Mesin automata yang terdiri dari finite number of states.
8. Salah satu state sebagai start state (initial state).
9. Terdapat setidaknya satu state yang diterima oleh komputer (final state / accepted state).
10. Mesin akan menerima input stream berupa alphabet yang akan diproses secara sekuensial.
11. Mesin akan berubah dari state satu ke state yang lain berdasarkan simbol input dan current state.
12. Pada suatu state dari DFA ada tepat satu state berikutnya untuk setiap simbol input (masukan) yang diterima.
13. Esensi dari DFA ada sebagai string recognizer (menerima atau menolak sebuah string) berdasarkan definisi bahasa dari DFA tersebut.
14. Penyajian dari DFA

DFA dapat disajikan dalam dua bentuk :

1. Transition Table



1. Transition Diagram



1. Contoh DFA

b

b

b

a

a

a

Q = {q0, q1, q2}

Σ = {a, b}

S = q0

F = {q2}

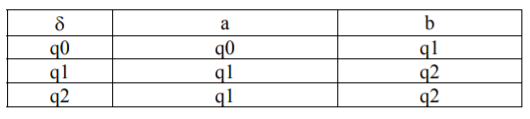
Fungsi-fungsi transisinya sebagai berikut :

δ (q0, a) = q0, δ (q0, b) = q1,

δ (q1, a) = q1, δ (q1, b) = q2,

δ (q2, a) = q1, δ (q2, b) = q2.

Jika disajikan dalam tabel transisi :



**BAB II**

**ANALISIS PERSOALAN**

1. Deskripsi Permasalahan

*The Sims* merupakan permainan yang dibuat oleh *Maxis* dan didistribusikan oleh *Electronic Arts*. *The Sims* memberikan sebuah pengalaman untuk mengatur setiap karakter dalam sebuah kota untuk membangun kota tersebut. Permainan akan dilakukan secara *real time* dan mengharuskan pemainnya untuk benar-benar memperhatikan setiap karakter dalam kota tersebut, dan bisa memilih *goal-goal* yang ingin dicapai. Tugas besar ini adalah simplifikasi dari permainan *The Sims*. Setiap pemain akan mengatur kegiatan seseorang dalam satu hari. Pemain dapat memilih satu diantara banyak aksi yang dapat dilakukan, dimana tiap aksi yang dipilih memiliki konsekuensi baik positif maupun negatif. Seperti layaknya manusia, aktor pada *The Sims* dapat mengalami perubahan kondisi, seperti lapar, bosan, lelah, dan lainnya. Kondisi-kondisi tersebut digambarkan menjadi beberapa atribut. Terdapat beberapa atribut dalam permainan *The Sims* yang dapat mempengaruhi keberjalanan permainan. Atribut tersebut adalah sebagai berikut.

1. *Hygiene*, merupakan tingkat kebersihan dari pemain
2. *Bladder*, merupakan tingkat keinginan pemain untuk membuang air besar atau air kecil.
3. *Hunger*, merupakan tingkat kelaparan pemain.
4. *Energy*, merupakan tingkat energi yang dimiliki pemain.
5. *Social*, merupakan tingkat kesosialan yang dimiliki pemain.
6. *Fun*, merupakan tingkat kesenangan dari pemain.

Untuk mempermudah tugas besar ini, **atribut-atribut tersebut dimodifikasi** sebagai berikut:

1. ***Hygiene***, gabungan dari atribut *Hygiene* dan *Bladder* dalam *The Sims*.
2. ***Energy***, gabungan dari atribut *Energy* dan *Hunger* dalam *The Sims*.
3. ***Fun***, gabungan dari atribut *Fun* dan *Social* dalam *The Sims*.

Setiap atribut memiliki nilai maksimum 15 dan nilai minimum 0. Setiap atribut dapat bertambah (sampai nilai maksimum) atau berkurang (sampai nilai minimum) jika pemain melakukan aksi tertentu. Untuk mempermudah tugas besar ini, elemen waktu tidak diperhitungkan dalam simulasi ini. **Kondisi awal pemain selalu dalam keadaan sudah bangun tidur dengan atribut *Hygiene* bernilai 0, *Energy* bernilai 10, dan *Fun* bernilai 0. Permainan dinyatakan selesai jika semua atribut bernilai 0 atau semua atribut bernilai 15.**

Berikut ini adalah tabel aksi yang dapat dilakukan, rentang waktu untuk melakukan aksi tersebut, dan konsekuensinya:

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Aksi** | **Jenis** | **Konsekuensi** |
| 1 | Tidur <jenis\_tidur> | Siang | +10 *energy* |
| Malam | +15 *energy* |
| 2 | Makan <jenis\_makanan> | * *Hamburger* * *Pizza* * *Steak and Beans* | *Hamburger*:  +5 *energy*  *Pizza:*  +10 *energy*  *Steak and Beans*:  +15 *energy* |
| 3 | Minum <jenis\_minuman> | * Air * Kopi * Jus | Air:  *-*5 *hygiene*  Kopi:  +5 *energy*  -10 *hygiene*  Jus:  +10 *energy*  -5 *hygiene* |
| 4 | Buang Air <jenis> | Kecil | *+*5 *hygiene* |
| Besar | +10 *hygiene*  -5 *energy* |
| 5 | Bersosialisasi ke Kafe | - | +15 *fun*  *-*10 *energy*  *-*5 *hygiene* |
| 6 | Bermain Media Sosial | - | +10 *fun*  *-*10 *energy* |
| 7 | Bermain komputer | - | +15 *fun*  *-*10 *energy* |
| 8 | Mandi | - | +15 *hygiene*  *-*5 *energy* |
| 9 | Cuci Tangan | - | +5 *hygiene* |
| 10 | Mendengarkan Musik di Radio | - | *+*10 *fun*  *-*5 *energy* |
| 11 | Membaca <jenis\_bacaan> | Jenis Bacaan:   * Koran * Novel | Koran:  *+*5 *fun*  *-*5 *energy*  Novel:  *+*10 *fun*  *-*5 *energy* |

Program yang dibuat tidak memerlukan GUI, sehingga masukan hanya berasal dari *keyboard* di *command line*. Masukan berupa satu aksi. **Jika sebuah aksi menambah atau mengurangi satu atau lebih atribut sehingga bernilai lebih kecil dari 0 atau lebih besar dari 15 merupakan aksi yang tidak valid dan mengeluarkan tulisan “Aksi tidak valid”.** Aksi yang valid akan mengeluarkan tulisan yang berisi nilai dari ketiga atribut dengan *format* sebagai berikut:

|  |
| --- |
| Hygiene = <Total\_Hygiene>  Energy = <Total\_Energy>  Fun = <Total\_Fun> |

Untuk mempermudah pemahaman, diberikan contoh konteks dan dua contoh masukan beserta keluarannya sebagai berikut:

**Contoh konteks**

|  |
| --- |
| Hygiene = 10  Energy = 15  Fun = 15 |

**Contoh Masukan 1**

|  |
| --- |
| Cuci Tangan |

**Keluaran 1**

|  |
| --- |
| Hygiene = 15  Energy = 15  Fun = 15 |

“Cuci Tangan” merupakan aksi yang valid karena menambah *Hygiene* sebanyak 5 sehingga total *Hygiene*-nya 15, seperti yang ditunjukkan pada bagian yang berwarna merah.

**Contoh Masukan 2**

|  |
| --- |
| Makan Hamburger |

**Keluaran 2**

|  |
| --- |
| Aksi tidak valid |

“Makan Hamburger” merupakan aksi yang valid, tetapi aksi ini menambah *Energy* sebanyak 5 sehingga total *Energy*-nya 20. Berarti aksi ini menghasilkan total *Energy* yang berada di luar batas permainan sehingga aksinya tidak valid dan mengeluarkan tulisan, “Aksi tidak valid”.

1. Pemecahan masalah

Jumlah state didapatkan dari jumlah kemungkinan yang dimungkinkan dari masalah ini. Pada kasus kali ini, terdapat 3 buah keadaan yaitu Hygiene, Energy, dan Fun. Dari setiap keadaan tersebut, masing – masing memiliki 4 jenis keadaan yang lain yaitu 0, 5 ,10, dan 15 yang melambangkan angka dari masing – masing keadaan. Dari hal tersebut, dapat disimpulkan terdapat 43 states yaitu sebanyak 64 states.

Untuk setiap states, kami memberi nama states tersebut q0, q1, q2, …, q63. Pada DFA ini, terdapat satu buah start state yaitu q8 dimana q8 ini memiliki kondisi yaitu Hygiene sebesar 0, Energy sebesar 10, dan Fun sebesar 0. Karena pada persoalan ini terdapat dua kemungkinan akhir dari simulasi yaitu jika pemain memiliki Hygiene 15, Energy 15, dan Fun 15 (kemungkinan menang) ataupun Hygiene 0, Energy 0, dan Fun 0 (kemungkinan kalah), maka DFA ini memiliki 2 Final States yaitu pada q0 (states yang berisi kemungkinan kalah) dan q 63 (states yang berisi kemungkinan menang). Setiap states akan berubah dengan sebuah masukan dari pengguna (alphabet) yaitu 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, dan Membaca Novel.

DFA ini dapat didefinisikan sebagai berikut:

* Q = {q0, q1, q2, q3, …, q63}
* Σ = {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, dan Membaca Novel}
* S = {q8}
* F = {q0, q63}

DFA ini juga dapat didefinisikan dengan sebuah tabel transisi. Di bawah ini terdapat sebuah tabel transisi yang dibagi menjadi 3 bagian sebagai berikut:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| START / END | Aksi | | | State | Tidur Siang E +10 | Tidur Malam E +15 | Makan Hamburger E +5 | Makan Pizza E +10 | Makan Steak and Beans E +15 | Minum Air H -5 |
| Hygiene | Energy | Fun |
| END | 0 | 0 | 0 | q0 | {} | {} | {} | {} | {} | {} |
|  | 0 | 0 | 5 | q1 | q9 | q13 | q5 | q9 | q13v | q1 |
|  | 0 | 0 | 10 | q2 | q10 | q14 | q6 | q10 | q14 | q2 |
|  | 0 | 0 | 15 | q3 | q11 | q15 | q7 | q11 | q15 | q3 |
|  | 0 | 5 | 0 | q4 | q12 | q4 | q8 | q12 | q4 | q4 |
|  | 0 | 5 | 5 | q5 | q13 | q5 | q9 | q13 | q5 | q5 |
|  | 0 | 5 | 10 | q6 | q14 | q6 | q10 | q14 | q6 | q6 |
|  | 0 | 5 | 15 | q7 | q15 | q7 | q11 | q15 | q7 | q7 |
| START | 0 | 10 | 0 | q8 | q8 | q8 | q12 | q8 | q8 | q8 |
|  | 0 | 10 | 5 | q9 | q9 | q9 | q13 | q9 | q9 | q9 |
|  | 0 | 10 | 10 | q10 | q10 | q10 | q14 | q10 | q10 | q10 |
|  | 0 | 10 | 15 | q11 | q11 | q11 | q15 | q11 | q11 | q11 |
|  | 0 | 15 | 0 | q12 | q12 | q12 | q12 | q12 | q12 | q12 |
|  | 0 | 15 | 5 | q13 | q13 | q13 | q13 | q13 | q13 | q13 |
|  | 0 | 15 | 10 | q14 | q14 | q14 | q14 | q14 | q14 | q14 |
|  | 0 | 15 | 15 | q15 | q15 | q15 | q15 | q15 | q15 | q15 |
|  | 5 | 0 | 0 | q16 | q24 | q28 | q20 | q24 | q28 | q0 |
|  | 5 | 0 | 5 | q17 | q25 | q29 | q21 | q25 | q29 | q1 |
|  | 5 | 0 | 10 | q18 | q26 | q30 | q22 | q26 | q30 | q2 |
|  | 5 | 0 | 15 | q19 | q27 | q31 | q23 | q27 | q31 | q3 |
|  | 5 | 5 | 0 | q20 | q28 | q20 | q24 | q28 | q20 | q4 |
|  | 5 | 5 | 5 | q21 | q29 | q21 | q25 | q29 | q21 | q5 |
|  | 5 | 5 | 10 | q22 | q30 | q22 | q26 | q30 | q22 | q6 |
|  | 5 | 5 | 15 | q23 | q31 | q23 | q27 | q31 | q23 | q7 |
|  | 5 | 10 | 0 | q24 | q24 | q24 | q28 | q24 | q24 | q8 |
|  | 5 | 10 | 5 | q25 | q25 | q25 | q29 | q25 | q25 | q9 |
|  | 5 | 10 | 10 | q26 | q26 | q26 | q30 | q26 | q26 | q10 |
|  | 5 | 10 | 15 | q27 | q27 | q27 | q31 | q27 | q27 | q11 |
|  | 5 | 15 | 0 | q28 | q28 | q28 | q28 | q28 | q28 | q12 |
|  | 5 | 15 | 5 | q29 | q29 | q29 | q29 | q29 | q29 | q13 |
|  | 5 | 15 | 10 | q30 | q30 | q30 | q30 | q30 | q30 | q14 |
|  | 5 | 15 | 15 | q31 | q31 | q31 | q31 | q31 | q31 | q15 |
|  | 10 | 0 | 0 | q32 | q40 | q44 | q36 | q40 | q44 | q16 |
|  | 10 | 0 | 5 | q33 | q41 | q45 | q37 | q41 | q45 | q17 |
|  | 10 | 0 | 10 | q34 | q42 | q46 | q38 | q42 | q46 | q18 |
|  | 10 | 0 | 15 | q35 | q43 | q47 | q39 | q43 | q47 | q19 |
|  | 10 | 5 | 0 | q36 | q44 | q36 | q40 | q44 | q36 | q20 |
|  | 10 | 5 | 5 | q37 | q45 | q37 | q41 | q45 | q37 | q21 |
|  | 10 | 5 | 10 | q38 | q46 | q38 | q42 | q46 | q38 | q22 |
|  | 10 | 5 | 15 | q39 | q47 | q39 | q43 | q47 | q39 | q23 |
|  | 10 | 10 | 0 | q40 | q40 | q40 | q44 | q40 | q40 | q24 |
|  | 10 | 10 | 5 | q41 | q41 | q41 | q45 | q41 | q41 | q25 |
|  | 10 | 10 | 10 | q42 | q42 | q42 | q46 | q42 | q42 | q26 |
|  | 10 | 10 | 15 | q43 | q43 | q43 | q47 | q43 | q43 | q27 |
|  | 10 | 15 | 0 | q44 | q44 | q44 | q44 | q44 | q44 | q28 |
|  | 10 | 15 | 5 | q45 | q45 | q45 | q45 | q45 | q45 | q29 |
|  | 10 | 15 | 10 | q46 | q46 | q46 | q46 | q46 | q46 | q30 |
|  | 10 | 15 | 15 | q47 | q47 | q47 | q47 | q47 | q47 | q31 |
|  | 15 | 0 | 0 | q48 | q56 | q60 | q52 | q56 | q60 | q32 |
|  | 15 | 0 | 5 | q49 | q57 | q61 | q53 | q57 | q61 | q33 |
|  | 15 | 0 | 10 | q50 | q58 | q62 | q54 | q58 | q62 | q34 |
|  | 15 | 0 | 15 | q51 | q59 | q63 | q55 | q59 | q63 | q35 |
|  | 15 | 5 | 0 | q52 | q60 | q52 | q56 | q60 | q52 | q36 |
|  | 15 | 5 | 5 | q53 | q61 | q53 | q57 | q61 | q53 | q37 |
|  | 15 | 5 | 10 | q54 | q62 | q54 | q58 | q62 | q54 | q38 |
|  | 15 | 5 | 15 | q55 | q63 | q55 | q59 | q63 | q55 | q39 |
|  | 15 | 10 | 0 | q56 | q56 | q56 | q60 | q56 | q56 | q40 |
|  | 15 | 10 | 5 | q57 | q57 | q57 | q61 | q57 | q57 | q41 |
|  | 15 | 10 | 10 | q58 | q58 | q58 | q62 | q58 | q58 | q42 |
|  | 15 | 10 | 15 | q59 | q59 | q59 | q63 | q59 | q59 | q43 |
|  | 15 | 15 | 0 | q60 | q60 | q60 | q60 | q60 | q60 | q44 |
|  | 15 | 15 | 5 | q61 | q61 | q61 | q61 | q61 | q61 | q45 |
|  | 15 | 15 | 10 | q62 | q62 | q62 | q62 | q62 | q62 | q46 |
| END | 15 | 15 | 15 | q63 | {} | {} | {} | {} | {} | {} |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| START / END | Aksi | | | State | Minum Kopi H -5, E +5 | Minum Jus H -5, E +10 | Buang Air Kecil H +5 | Buang Air Besar H +10, E -5 | Bersosialisasi ke Kafe H -5, E -10, F +15 | Bermain Media Sosial E -10, F +10 |
| Hygiene | Energy | Fun |
| END | 0 | 0 | 0 | q0 | {} | {} | {} | {} | {} | {} |
|  | 0 | 0 | 5 | q1 | q1 | q1 | q17 | q1 | q1 | q1 |
|  | 0 | 0 | 10 | q2 | q2 | q2 | q18 | q2 | q2 | q2 |
|  | 0 | 0 | 15 | q3 | q3 | q3 | q19 | q3 | q3 | q3 |
|  | 0 | 5 | 0 | q4 | q4 | q4 | q20 | q32 | q4 | q4 |
|  | 0 | 5 | 5 | q5 | q5 | q5 | q21 | q33 | q5 | q5 |
|  | 0 | 5 | 10 | q6 | q6 | q6 | q22 | q34 | q6 | q6 |
|  | 0 | 5 | 15 | q7 | q7 | q7 | q23 | q35 | q7 | q7 |
| START | 0 | 10 | 0 | q8 | q8 | q8 | q24 | q36 | q8 | q2 |
|  | 0 | 10 | 5 | q9 | q9 | q9 | q25 | q37 | q9 | q3 |
|  | 0 | 10 | 10 | q10 | q10 | q10 | q26 | q38 | q10 | q10 |
|  | 0 | 10 | 15 | q11 | q11 | q11 | q27 | q39 | q11 | q11 |
|  | 0 | 15 | 0 | q12 | q12 | q12 | q28 | q40 | q12 | q6 |
|  | 0 | 15 | 5 | q13 | q13 | q13 | q29 | q41 | q13 | q7 |
|  | 0 | 15 | 10 | q14 | q14 | q14 | q30 | q42 | q14 | q14 |
|  | 0 | 15 | 15 | q15 | q15 | q15 | q31 | q43 | q15 | q15 |
|  | 5 | 0 | 0 | q16 | q4 | q8 | q32 | q16 | q16 | q16 |
|  | 5 | 0 | 5 | q17 | q5 | q9 | q33 | q17 | q17 | q17 |
|  | 5 | 0 | 10 | q18 | q6 | q10 | q34 | q18 | q18 | q18 |
|  | 5 | 0 | 15 | q19 | q7 | q11 | q35 | q19 | q19 | q19 |
|  | 5 | 5 | 0 | q20 | q8 | q12 | q36 | q48 | q20 | q20 |
|  | 5 | 5 | 5 | q21 | q9 | q13 | q37 | q49 | q21 | q21 |
|  | 5 | 5 | 10 | q22 | q10 | q14 | q38 | q50 | q22 | q22 |
|  | 5 | 5 | 15 | q23 | q11 | q15 | q39 | q51 | q23 | q23 |
|  | 5 | 10 | 0 | q24 | q12 | q24 | q40 | q52 | q3 | q18 |
|  | 5 | 10 | 5 | q25 | q13 | q25 | q41 | q53 | q25 | q19 |
|  | 5 | 10 | 10 | q26 | q14 | q26 | q42 | q54 | q26 | q26 |
|  | 5 | 10 | 15 | q27 | q15 | q27 | q43 | q55 | q27 | q27 |
|  | 5 | 15 | 0 | q28 | q28 | q28 | q44 | q56 | q7 | q22 |
|  | 5 | 15 | 5 | q29 | q29 | q29 | q45 | q57 | q29 | q23 |
|  | 5 | 15 | 10 | q30 | q30 | q30 | q46 | q58 | q30 | q30 |
|  | 5 | 15 | 15 | q31 | q31 | q31 | q47 | q59 | q31 | q31 |
|  | 10 | 0 | 0 | q32 | q20 | q24 | q48 | q32 | q32 | q32 |
|  | 10 | 0 | 5 | q33 | q21 | q25 | q49 | q33 | q33 | q33 |
|  | 10 | 0 | 10 | q34 | q22 | q26 | q50 | q34 | q34 | q34 |
|  | 10 | 0 | 15 | q35 | q23 | q27 | q51 | q35 | q35 | q35 |
|  | 10 | 5 | 0 | q36 | q24 | q28 | q52 | q36 | q36 | q36 |
|  | 10 | 5 | 5 | q37 | q25 | q29 | q53 | q37 | q37 | q37 |
|  | 10 | 5 | 10 | q38 | q26 | q30 | q54 | q38 | q38 | q38 |
|  | 10 | 5 | 15 | q39 | q27 | q31 | q55 | q39 | q39 | q39 |
|  | 10 | 10 | 0 | q40 | q28 | q40 | q56 | q40 | q19 | q34 |
|  | 10 | 10 | 5 | q41 | q29 | q41 | q57 | q41 | q41 | q35 |
|  | 10 | 10 | 10 | q42 | q30 | q42 | q58 | q42 | q42 | q42 |
|  | 10 | 10 | 15 | q43 | q31 | q43 | q59 | q43 | q43 | q43 |
|  | 10 | 15 | 0 | q44 | q44 | q44 | q60 | q44 | q23 | q38 |
|  | 10 | 15 | 5 | q45 | q45 | q45 | q61 | q45 | q45 | q39 |
|  | 10 | 15 | 10 | q46 | q46 | q46 | q62 | q46 | q46 | q46 |
|  | 10 | 15 | 15 | q47 | q47 | q47 | q63 | q47 | q47 | q47 |
|  | 15 | 0 | 0 | q48 | q36 | q40 | q48 | q48 | q48 | q48 |
|  | 15 | 0 | 5 | q49 | q37 | q41 | q49 | q49 | q49 | q49 |
|  | 15 | 0 | 10 | q50 | q38 | q42 | q50 | q50 | q50 | q50 |
|  | 15 | 0 | 15 | q51 | q39 | q43 | q51 | q51 | q51 | q51 |
|  | 15 | 5 | 0 | q52 | q40 | q44 | q52 | q52 | q52 | q52 |
|  | 15 | 5 | 5 | q53 | q41 | q45 | q53 | q53 | q53 | q53 |
|  | 15 | 5 | 10 | q54 | q42 | q46 | q54 | q54 | q54 | q54 |
|  | 15 | 5 | 15 | q55 | q43 | q47 | q55 | q55 | q55 | q55 |
|  | 15 | 10 | 0 | q56 | q44 | q56 | q56 | q56 | q35 | q50 |
|  | 15 | 10 | 5 | q57 | q45 | q57 | q57 | q57 | q57 | q51 |
|  | 15 | 10 | 10 | q58 | q46 | q58 | q58 | q58 | q58 | q58 |
|  | 15 | 10 | 15 | q59 | q47 | q59 | q59 | q59 | q59 | q59 |
|  | 15 | 15 | 0 | q60 | q60 | q60 | q60 | q60 | q39 | q54 |
|  | 15 | 15 | 5 | q61 | q61 | q61 | q61 | q61 | q61 | q55 |
|  | 15 | 15 | 10 | q62 | q62 | q62 | q62 | q62 | q62 | q62 |
| END | 15 | 15 | 15 | q63 | {} | {} | {} | {} | {} | {} |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| START / END | Aksi | | | State | Bermain Komputer E -10, F +15 | Mandi H +15, E -5 | Cuci Tangan H +5 | Mendengarkan Musik di Radio E -5, F +10 | Membaca Koran E -5, F +5 | Membaca Novel E -5, F +10 |
| Hygiene | Energy | Fun |
| END | 0 | 0 | 0 | q0 | {} | {} | {} | {} | {} | {} |
|  | 0 | 0 | 5 | q1 | q1 | q1 | q17 | q1 | q1 | q1 |
|  | 0 | 0 | 10 | q2 | q2 | q2 | q18 | q2 | q2 | q2 |
|  | 0 | 0 | 15 | q3 | q3 | q3 | q19 | q3 | q3 | q3 |
|  | 0 | 5 | 0 | q4 | q4 | q48 | q20 | q2 | q1 | q2 |
|  | 0 | 5 | 5 | q5 | q5 | q49 | q21 | q3 | q2 | q3 |
|  | 0 | 5 | 10 | q6 | q6 | q50 | q22 | q6 | q3 | q6 |
|  | 0 | 5 | 15 | q7 | q7 | q51 | q23 | q7 | q7 | q7 |
| START | 0 | 10 | 0 | q8 | q3 | q52 | q24 | q6 | q5 | q6 |
|  | 0 | 10 | 5 | q9 | q9 | q53 | q25 | q7 | q6 | q7 |
|  | 0 | 10 | 10 | q10 | q10 | q54 | q26 | q10 | q7 | q10 |
|  | 0 | 10 | 15 | q11 | q11 | q55 | q27 | q11 | q11 | q11 |
|  | 0 | 15 | 0 | q12 | q7 | q56 | q28 | q10 | q9 | q10 |
|  | 0 | 15 | 5 | q13 | q13 | q57 | q29 | q11 | q10 | q11 |
|  | 0 | 15 | 10 | q14 | q14 | q58 | q30 | q14 | q11 | q14 |
|  | 0 | 15 | 15 | q15 | q15 | q59 | q31 | q15 | q15 | q15 |
|  | 5 | 0 | 0 | q16 | q16 | q16 | q32 | q16 | q16 | q16 |
|  | 5 | 0 | 5 | q17 | q17 | q17 | q33 | q17 | q17 | q17 |
|  | 5 | 0 | 10 | q18 | q18 | q18 | q34 | q18 | q18 | q18 |
|  | 5 | 0 | 15 | q19 | q19 | q19 | q35 | q19 | q19 | q19 |
|  | 5 | 5 | 0 | q20 | q20 | q20 | q36 | q18 | q17 | q18 |
|  | 5 | 5 | 5 | q21 | q21 | q21 | q37 | q19 | q18 | q19 |
|  | 5 | 5 | 10 | q22 | q22 | q22 | q38 | q22 | q19 | q22 |
|  | 5 | 5 | 15 | q23 | q23 | q23 | q39 | q23 | q23 | q23 |
|  | 5 | 10 | 0 | q24 | q19 | q24 | q40 | q22 | q21 | q22 |
|  | 5 | 10 | 5 | q25 | q25 | q25 | q41 | q23 | q22 | q23 |
|  | 5 | 10 | 10 | q26 | q26 | q26 | q42 | q26 | q23 | q26 |
|  | 5 | 10 | 15 | q27 | q27 | q27 | q43 | q27 | q27 | q27 |
|  | 5 | 15 | 0 | q28 | q23 | q28 | q44 | q26 | q25 | q26 |
|  | 5 | 15 | 5 | q29 | q29 | q29 | q45 | q27 | q26 | q27 |
|  | 5 | 15 | 10 | q30 | q30 | q30 | q46 | q30 | q27 | q30 |
|  | 5 | 15 | 15 | q31 | q31 | q31 | q47 | q31 | q31 | q31 |
|  | 10 | 0 | 0 | q32 | q32 | q32 | q48 | q32 | q32 | q32 |
|  | 10 | 0 | 5 | q33 | q33 | q33 | q49 | q33 | q33 | q33 |
|  | 10 | 0 | 10 | q34 | q34 | q34 | q50 | q34 | q34 | q34 |
|  | 10 | 0 | 15 | q35 | q35 | q35 | q51 | q35 | q35 | q35 |
|  | 10 | 5 | 0 | q36 | q36 | q36 | q52 | q34 | q33 | q34 |
|  | 10 | 5 | 5 | q37 | q37 | q37 | q53 | q35 | q34 | q35 |
|  | 10 | 5 | 10 | q38 | q38 | q38 | q54 | q38 | q35 | q38 |
|  | 10 | 5 | 15 | q39 | q39 | q39 | q55 | q39 | q39 | q39 |
|  | 10 | 10 | 0 | q40 | q35 | q40 | q56 | q38 | q37 | q38 |
|  | 10 | 10 | 5 | q41 | q41 | q41 | q57 | q39 | q38 | q39 |
|  | 10 | 10 | 10 | q42 | q42 | q42 | q58 | q42 | q39 | q42 |
|  | 10 | 10 | 15 | q43 | q43 | q43 | q59 | q43 | q43 | q43 |
|  | 10 | 15 | 0 | q44 | q39 | q44 | q60 | q42 | q41 | q42 |
|  | 10 | 15 | 5 | q45 | q45 | q45 | q61 | q43 | q42 | q43 |
|  | 10 | 15 | 10 | q46 | q46 | q46 | q62 | q46 | q43 | q46 |
|  | 10 | 15 | 15 | q47 | q47 | q47 | q63 | q47 | q47 | q47 |
|  | 15 | 0 | 0 | q48 | q48 | q48 | q48 | q48 | q48 | q48 |
|  | 15 | 0 | 5 | q49 | q49 | q49 | q49 | q49 | q49 | q49 |
|  | 15 | 0 | 10 | q50 | q50 | q50 | q50 | q50 | q50 | q50 |
|  | 15 | 0 | 15 | q51 | q51 | q51 | q51 | q51 | q51 | q51 |
|  | 15 | 5 | 0 | q52 | q52 | q52 | q52 | q50 | q49 | q50 |
|  | 15 | 5 | 5 | q53 | q53 | q53 | q53 | q51 | q50 | q51 |
|  | 15 | 5 | 10 | q54 | q54 | q54 | q54 | q54 | q51 | q54 |
|  | 15 | 5 | 15 | q55 | q55 | q55 | q55 | q55 | q55 | q55 |
|  | 15 | 10 | 0 | q56 | q51 | q56 | q56 | q54 | q53 | q54 |
|  | 15 | 10 | 5 | q57 | q57 | q57 | q57 | q55 | q54 | q55 |
|  | 15 | 10 | 10 | q58 | q58 | q58 | q58 | q58 | q55 | q58 |
|  | 15 | 10 | 15 | q59 | q59 | q59 | q59 | q59 | q59 | q59 |
|  | 15 | 15 | 0 | q60 | q55 | q60 | q60 | q58 | q57 | q58 |
|  | 15 | 15 | 5 | q61 | q61 | q61 | q61 | q59 | q58 | q59 |
|  | 15 | 15 | 10 | q62 | q62 | q62 | q62 | q62 | q59 | q62 |
| END | 15 | 15 | 15 | q63 | {} | {} | {} | {} | {} | {} |

1. Proses pembuatan DFA

Dalam proses mengubah persoalan di tersebt menjadi tabel transisi DFA, kami menggunakan aplikasi Google Sheets dan menggunakan fungsi yang tersedia. Fungsi yang kami gunakan adalah IF, IFERROR, dan FILTER. Proses yang kami lakukan adalah sebagai berikut:

* 1. Membuat sebuah tabel angka yang hanya akan berisi angka kondisi states.
  2. Membuat 3 kolom berisi Hygiene, Energy, dan Fun beserta mengisinya dengan seluruh kondisi untuk setiap state, berarti kami perlu membuat 64 baris.
  3. Membuat 18 kolom untuk setiap aksi, yang tiap baris memiliki 3 subkolom. Subkolom ini berjudul Hygiene, Energy, dan Fun.
  4. Mengisi sub kolom dengan kondisi yang sesuai dengan state dan perlakuan yang akan diberikan. Di sini kami menggunakan fungsi IF untuk mengisi subkolomnya. Misalnya untuk mengisi kolom “Tidur Siang”, jika jumlah Energy kurang dari 10, maka di sel tersebut akan diisi dengan jumlah Energy pada state tersebut ditambah 10. Selain itu, sel akan diisi dengan string “Aksi tidak valid”. Karena jumlah yang mungkin hanyalah 0, 5, 10, dan 15, kami tidak perlu mengantisipasi jika nilainya diantara nilai-nilai tersebut. Berikut contoh kode yang kami implementasikan:

=IF($C3<10,$C3+10,"Aksi tidak valid")

Pada subkolom yang nilainya tidak perlu ditambah, kami hanya menyalin nilai yang sama dengan kondisi.

* 1. Membuat sebuah tabel states yang hanya berisi state q0 sampai q63.
  2. Membuat 19 kolom, dengan kolom pertama berjudul Current States yang berisi q0 sampai q63, dan 18 kolom yang lain berjudul setiap aksinya dan akan berisi transisi yang tepat.
  3. Mengisi setiap sel dengan transisi yang tepat menggunakan fungsi FILTER yang akan di nest di dalam fungsi IFERROR. Fungsi FILTER bekerja dengan menerima sebuah kolom berisi state, mencocokan subkolom Hygiene pada suatu sel di tabel angka dengan kolom kondisi Hygiene, kemudian untuk Energy, dan Fun. Fungsi ini akan mengembalikan state dengan kondisi yang sesuai. Dengan fungsi IFERROR, jika pada sel yang dicocokan terdapat string “Aksi tidak valid”, maka akan fungsi akan mengembalikan state yang sama. Misalnya, state q8 yang diberi aksi “Mandi” akan memiliki nilai Hygiene 15, Energy 5, dan Fun 0, dan akan mengembalikan state q52. Namun jika state q8 diberi aksi “Tidur Malam” akan mengembalikan q8. Berikut contoh kode yang kami implementasikan:

=IFERROR(FILTER($E$3:$E$66,$B$3:$B$66='Sheet Angka'!E4,$C$3:$C$66='Sheet Angka'!F4,$D$3:$D$66='Sheet Angka'!G4),E4)

Sebagai catatan, tabel angka yang kami buat terletak pada sheet yang berbeda untuk mempermudah pembacaan tabel.

* 1. Khusus untuk state q0 dan q63, jika diberi aksi apapun akan mengembalikan himpunan kosong, karena state tersebut merupakan final state.

Tabel state yang telah kami buat akan diexport menjadi sebuah file .csv dan digunakan dalam implementasi program utama kami untuk membaca dan mengembalikan states. Metode ini kami pilih karena terdapat library Pandas pada Python yang menurut kami akan lebih memudahkan pekerjaan kami untuk membaca tabel transisi yang telah kami buat.

**BAB III**

**Implementasi dan Pengujian**

Pada deskripsi masalah yang telah dibahas di bab sebelumnya, kami membuat penyelesaian masalah tersebut dalam sebuah program yang dibuat dengan bahasa Python. Kami memilih bahasa Python karena pada bahasa tersebut memiliki library Pandas yang dapat dengan mudah mengakses tabel transisi yang telah kami buat sebelumnya dengan format file .csv.

Untuk menjalankan program kami, diperlukan bahasa pemrograman Python, dan juga diperlukan untuk mengunduh library Pandas terlebih dahulu. Cara kerja dari program yang kami buat adalah setelah game dimulai, kami meminta input dari pemain yaitu aksi yang akan dilakukan. Setelah itu kami mencocokkan aksi yang diinput oleh pengguna dengan state yang sekarang sebelum dimasukkannya aksi tersebut. Selanjutnya kami mengambil tabel yang bersinggungan dengan state yang sekarang dan aksi yang dilakukan, sehingga didapatkan nilai state yang baru setelah dilakukan aksi tersebut. Setelah state yang baru diketahui, maka kami mengubah state tersebut dengan kondisi yang ada yaitu berapa hygiene, energy, dan fun dari sim tersebut. Jika state yang lama dengan yang baru tidak berubah, maka dapat disimpulkan bahwa aksi yang dilakukan oleh pemain tidaklah valid. Jika aksi tersebut valid, maka proses tersebut akan diulang sampai pemain mencapai final state yaitu jika pemain menang (state q63) atau jika pemain kalah (state q0).

Dari program tersebut, kami mendefinisikan beberapa fungsi dan prosedur yaitu:

1. Start Game

def startgame()

I.S. : Program dimulai.

F.S. : Hygiene =0, Energy = 10, Fun = 0

Fungsi ini digunakan untuk memulai game, dan mengembalikan nilai Hygiene 0, Energy 10, dan Fun 0.

1. Print State

def printState(hygiene,energy,fun)

I.S. : Hygiene, Energy, dan Fun terdefinisi

F.S. : Mencetak nilai Hygiene, Energy, dan Fun ke layar dengan sedikit interface

Prosedur ini digunakan untuk mencetak nilai Hygiene, Energy, dan Fun yang sedang dimiliki oleh pemain.

1. Menang

def menang(hygiene,energy,fun)

I.S. : Hygiene, Energy, dan Fun terdefinisi

F.S. : Mengembalikan true jika Hygiene = 15, Energy = 15, Fun = 15

Fungsi ini akan mengembalikan nilai true jika pemain menang, yaitu memiliki nilai hygiene, energy, dan fun sebesar 15.

1. Kalah

def kalah(hygiene,energy,fun)

I.S. : Hygiene, Energy, dan Fun terdefinisi

F.S. : Mengembalikan true jika Hygiene = 0, Energy = 0, Fun = 0

Fungsi ini akan mengembalikan nilai true jika pemain kalah, yaitu memiliki nilai hygiene, energy, dan fun sebesar 0.

1. Read Action

def readAction(aksi,currstate)

I.S. : Aksi bebas, currstate terdefinisi

F.S. : Mengembalikan state baru sesuai aksi yang dilakukan

Fungsi ini akan mengembalikan nilai state akhir (newstate) dengan input yaitu aksi yang dilakukan dan state awal sebelum aksi dilakukan (currstate). Cara kerja dari fungsi ini adalah fungsi akan mengambil indeks dari currstate pada tabel “Current State” di file tabel transisi. Setelah mendapatkan indeksnya, lalu dia akan mengambil nilai dari “Aksi” yang dilakukan pada indeks tersebut, sehingga akan didapatkan state yang baru. Jika aksi yang dilakukan pengguna tidak terdefinisi (salah pengetikan) maka fungsi akan mengembalikan nilai state awal.

1. Change State

def changeState(condition,awal,akhir)

I.S. : Array kondisi, state awal, dan state akhir terdefinisi

F.S. : Mengembalikan kondisi baru (array berisi hygiene, energy, dan fun) yang dibaca dari table transisi sesuai transisi dari state awal ke akhir. Jika state sama maka mengembalikan pesan “Aksi tidak valid”.

Fungsi ini akan mengubah state menjadi sebuah kondisi. Kondisi adalah keadaan pemain, yaitu jumlah hygiene, energy, dan fun. Awal dan akhir merupakan state dari pemain sebelum dilakukan aksi dan setelah dilakukannya aksi. Pada fungsi ini terdapat syarat jika state awal sama dengan state akhir, maka state tidak berubah dan dapat disimpulkan bahwa aksi yang dilakukan pemain tidak valid dan akan mencetak “Aksi tidak valid” dan mengembalikan nilai condition awal dari sim. Jika state awal tidak sama dengan state akhir, maka state akan berubah dan aksi valid. Lalu state akhir pemain akan diubah ke kondisi hygiene, energy, dan fun yang diambil dari file tabel transisi yang telah kami buat sebelumnya.

1. Menu Utama

# $ ====================== MAIN PROGRAM ======================

# \* Welcome

print("[] ========================================== []")

print("||    Selamat datang di                       ||")

print("||            Simulasi Kehidupan Sederhana    ||")

print("[] ========================================== []")

print("||                 List Aksi                  ||")

print("[] ========================================== []")

print("||  1. Tidur Siang .................. E + 10  ||")

print("||  2. Tidur Malam .................. E + 15  ||")

print("||  3. Makan Hamburger .............. E +  5  ||")

print("||  4. Makan Pizza .................. E + 10  ||")

print("||  5. Makan Steak and Beans ........ E + 15  ||")

print("||  6. Minum Air .................... H -  5  ||")

print("||  7. Minum Kopi ................... H -  5  ||")

print("||                                    E +  5  ||")

print("||  8. Minum Jus .................... H -  5  ||")

print("||                                    E + 10  ||")

print("||  9. Buang Air Kecil .............. H +  5  ||")

print("|| 10. Buang Air Besar .............. H + 10  ||")

print("||                                    E -  5  ||")

print("|| 11. Bersosialisasi ke Kafe ....... H -  5  ||")

print("||                                    E - 10  ||")

print("||                                    F + 15  ||")

print("|| 12. Bermain Media Sosial ......... E - 10  ||")

print("||                                    F + 10  ||")

print("|| 13. Bermain Komputer ............. E - 10  ||")

print("||                                    F + 15  ||")

print("|| 14. Mandi ........................ H + 15  ||")

print("||                                    E -  5  ||")

print("|| 15. Cuci Tangan .................. H +  5  ||")

print("|| 16. Mendengarkan Musik di Radio .. E -  5  ||")

print("||                                    F + 10  ||")

print("|| 17. Membaca Koran ................ E -  5  ||")

print("||                                    F +  5  ||")

print("|| 18. Membaca Novel ................ E -  5  ||")

print("||                                    F + 10  ||")

print("[] ========================================== []")

print("")

# \* Start Game

cond = startgame()

currstate = "q8"

printState(cond[0],cond[1],cond[2])

# \* Start Loop

while (not(menang(cond[0],cond[1],cond[2])) and not(kalah(cond[0],cond[1],cond[2]))):

print("")

aksi = str(input("Mau ngapain bray?? : "))

newstate = readAction(aksi,currstate)

cond = changeState(cond,currstate,newstate)

currstate = newstate

print("")

printState(cond[0],cond[1],cond[2])

# \* Game Selesai

if (menang(cond[0],cond[1],cond[2])):

print("")

print(" \_\_     \_\_\_\_\_\_  \_    \_  \_\_          \_\_\_\_\_\_\_ \_   \_  ")

print(" \ \   / / \_\_ \| |  | | \ \        / /\_   \_| \ | | ")

print("  \ \\_/ / |  | | |  | |  \ \  /\  / /  | | |  \| | ")

print("   \   /| |  | | |  | |   \ \/  \/ /   | | | . ` | ")

print("    | | | |\_\_| | |\_\_| |    \  /\  /   \_| |\_| |\  | ")

print("    |\_|  \\_\_\_\_/ \\_\_\_\_/      \/  \/   |\_\_\_\_\_|\_| \\_| ")

print("[]================================================[]")

print("||  Pesan Sponsor:                                ||")

print("||                  Sehat bet idup lo brayyy :)   ||")

print("[]================================================[]")

if (kalah(cond[0],cond[1],cond[2])):

print("")

print(" \_\_   \_\_\_\_\_\_\_ \_   \_   \_     \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_  ")

print(" \ \ / /  \_  | | | | | |   |  \_  /  \_\_\_|  \_\_\_|")

print("  \ V /| | | | | | | | |   | | | \ `--.| |\_\_   ")

print("   \ / | | | | | | | | |   | | | |`--. \  \_\_|  ")

print("   | | \ \\_/ / |\_| | | |\_\_\_\ \\_/ /\\_\_/ / |\_\_\_  ")

print("   \\_/  \\_\_\_/ \\_\_\_/  \\_\_\_\_\_/\\_\_\_/\\_\_\_\_/\\_\_\_\_/ ")

print("[]============================================[]")

print("||  Pesan Sponsor:                            ||")

print("||            Telah berpulang seorang sim :(  ||")

print("[]============================================[]")

print("")

print("")

print('[]---------------------------------------------[]');

print('||    Terima kasih telah memainkan             ||');

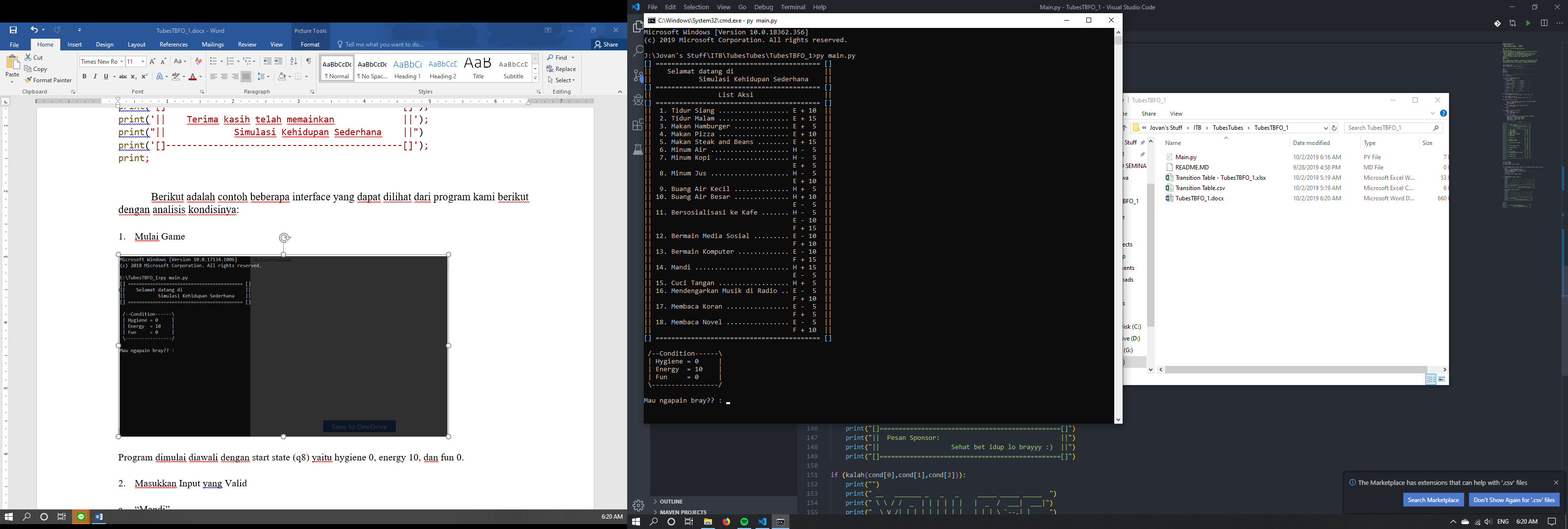
print("||             Simulasi Kehidupan Sederhana    ||")

print('[]---------------------------------------------[]');

print;

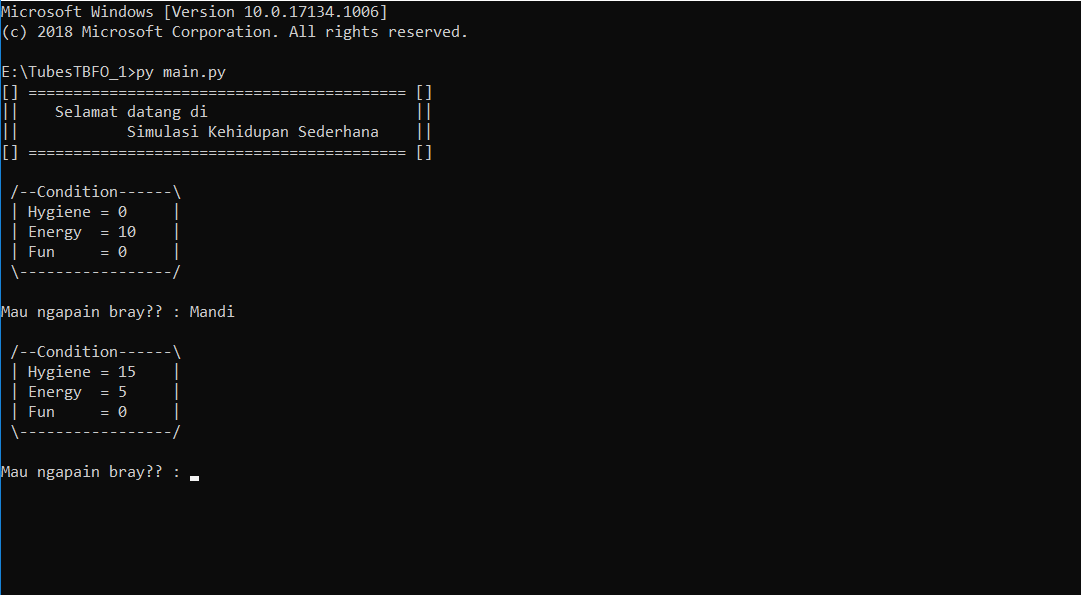
Berikut adalah contoh beberapa interface yang dapat dilihat dari program kami berikut dengan analisis kondisinya:

* 1. Mulai Game



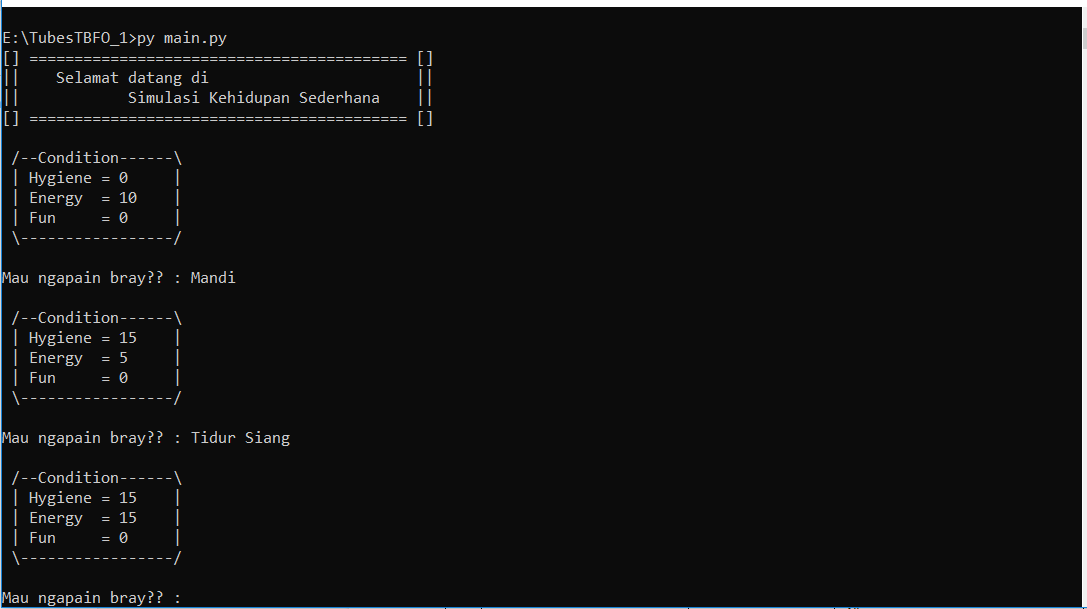
Program dimulai diawali dengan start state (q8) yaitu hygiene 0, energy 10, dan fun 0.

* 1. Masukkan Input yang Valid
     1. “Mandi”



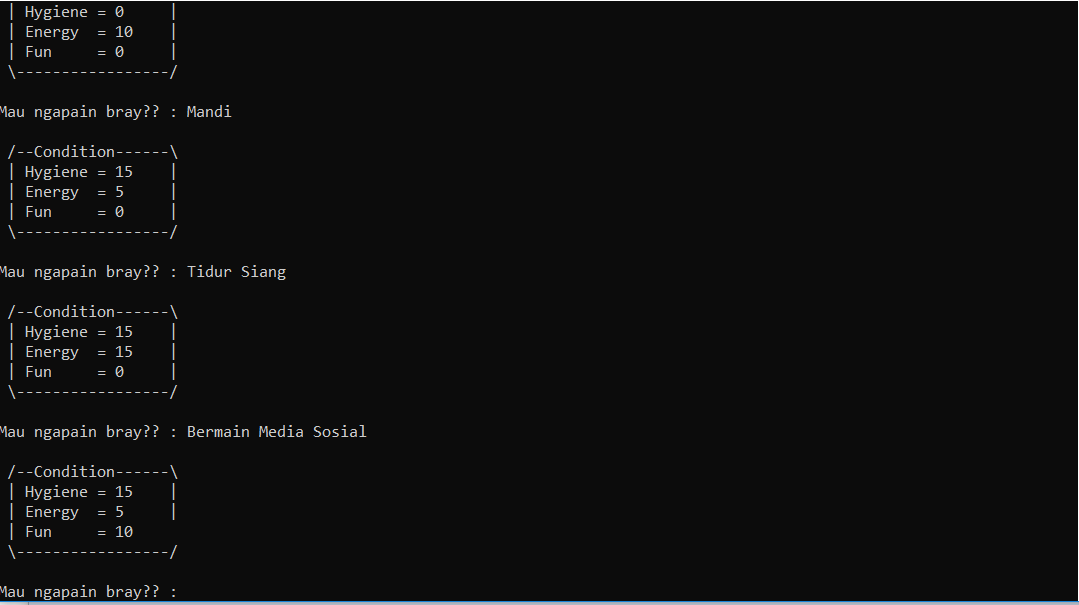
“Mandi” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Tidur Siang”



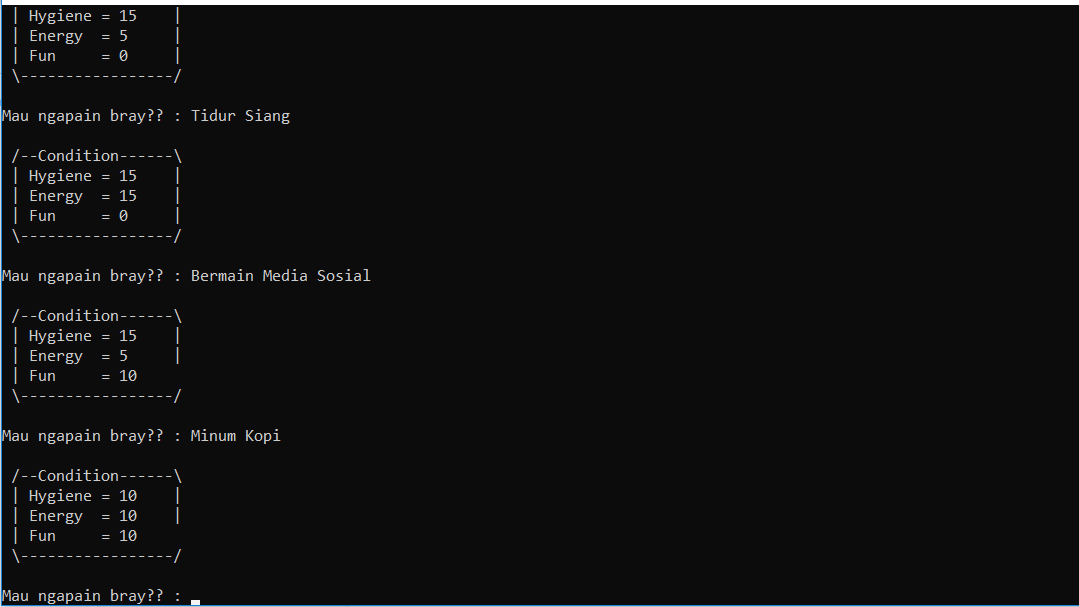
“Tidur Siang” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Bermain Media Sosial”



“Bermain Media Sosial” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Minum Kopi”



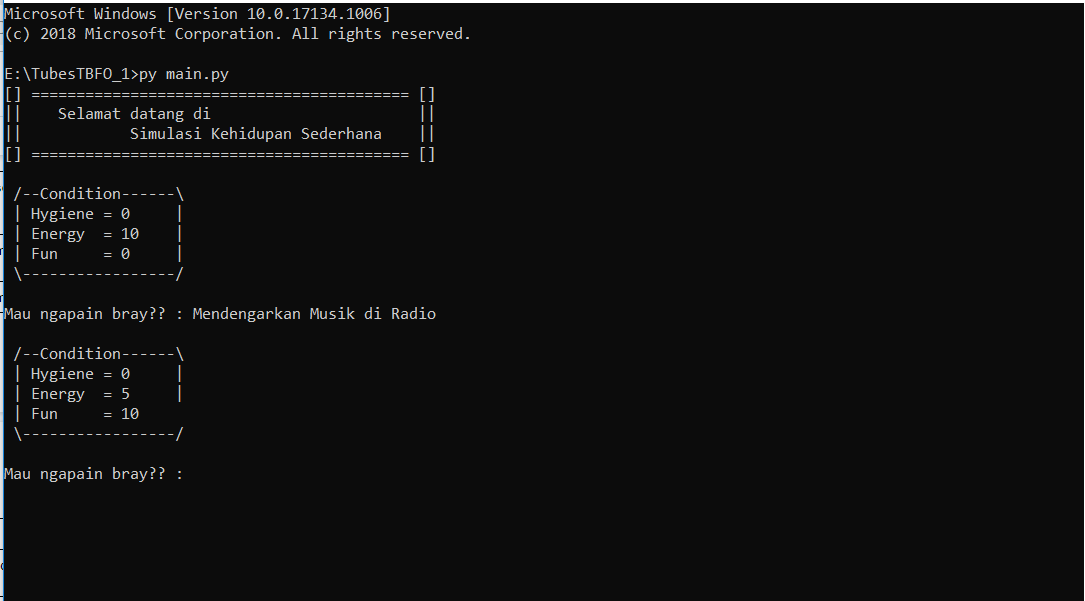
“Minum Kopi” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Membaca Koran”



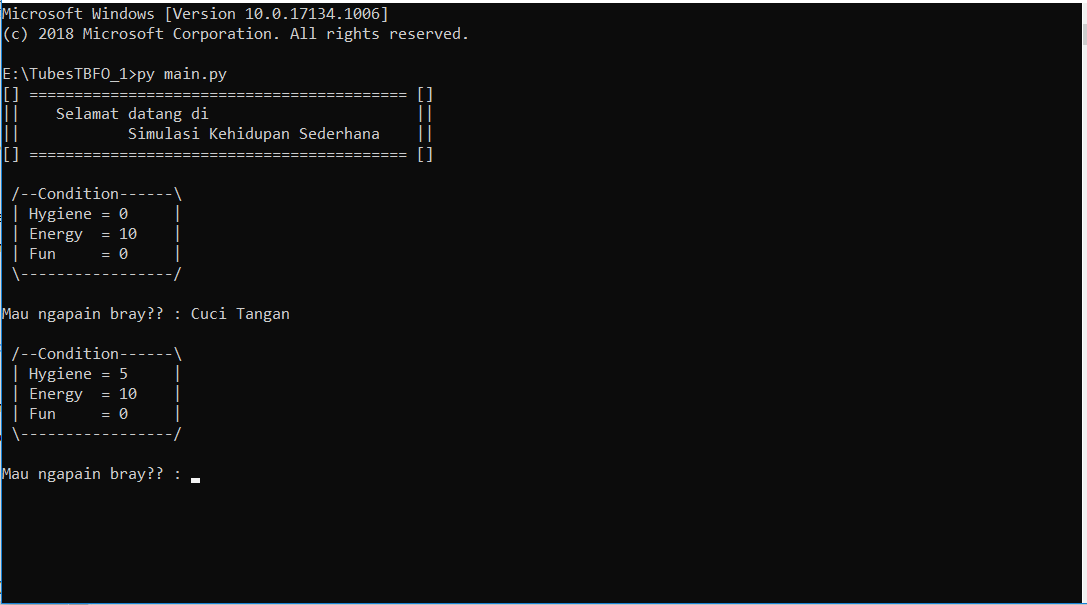
“Membaca Koran” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Mendengarkan Musik di Radio”



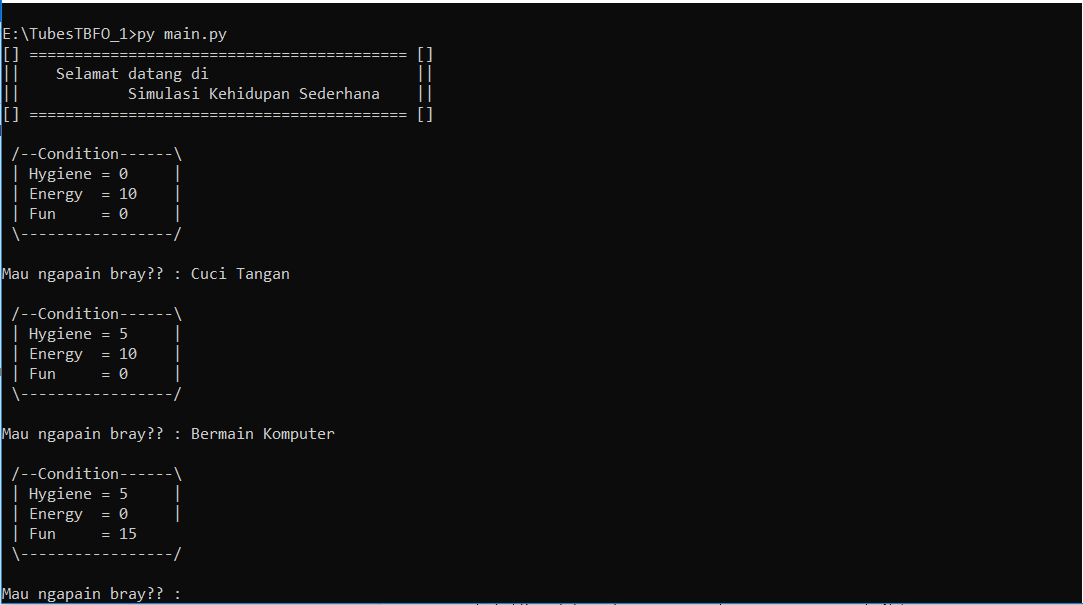
“Mendengarkan Musik di Radio” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Cuci Tangan”



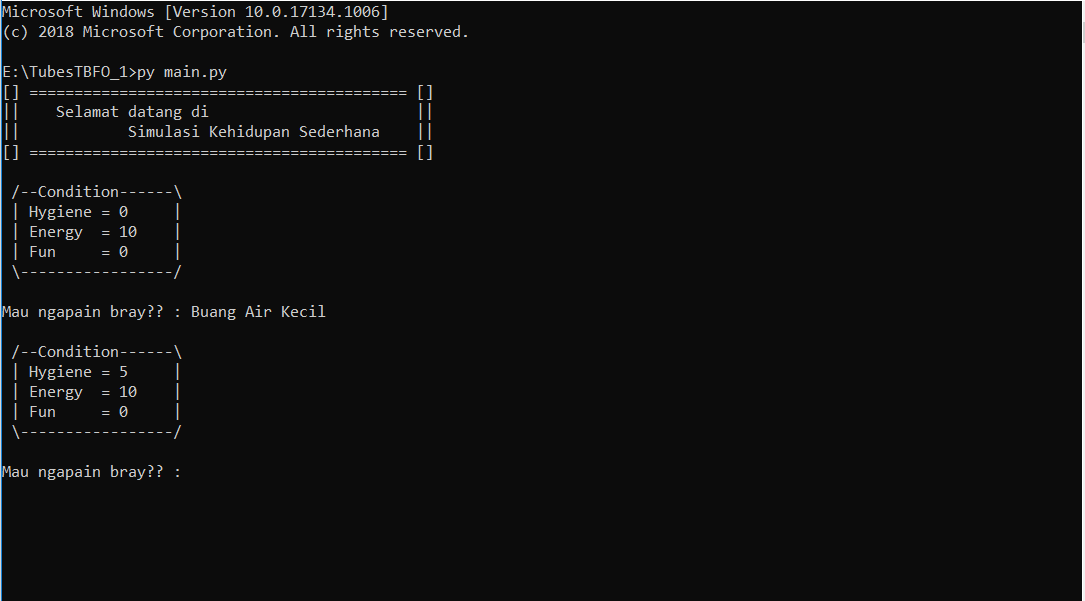
“Cuci Tangan” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. Bermain Komputer



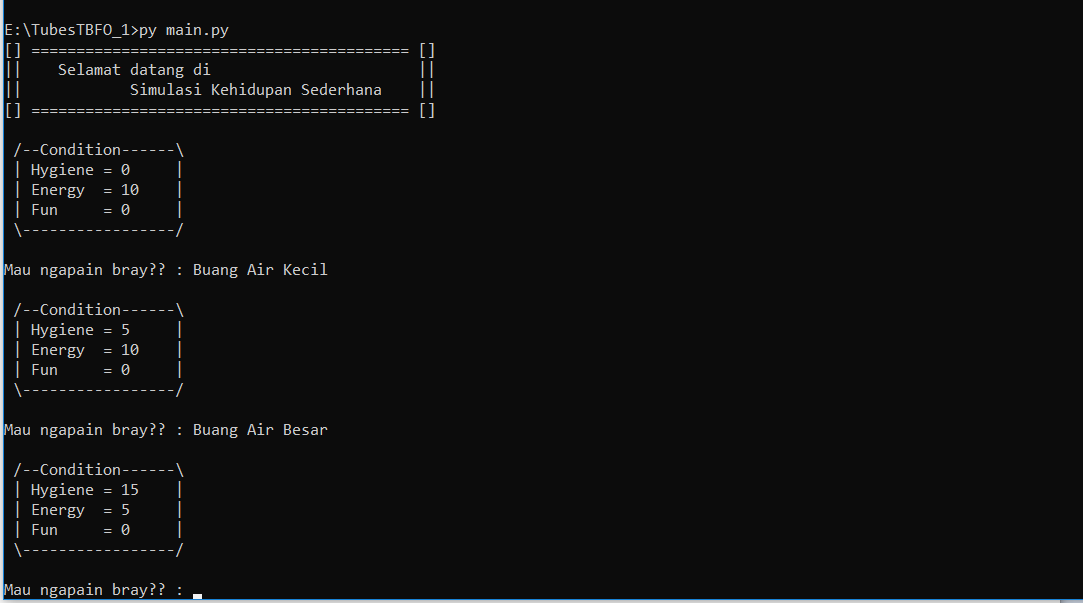
“Bermain Komputer” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Buang Air Kecil”



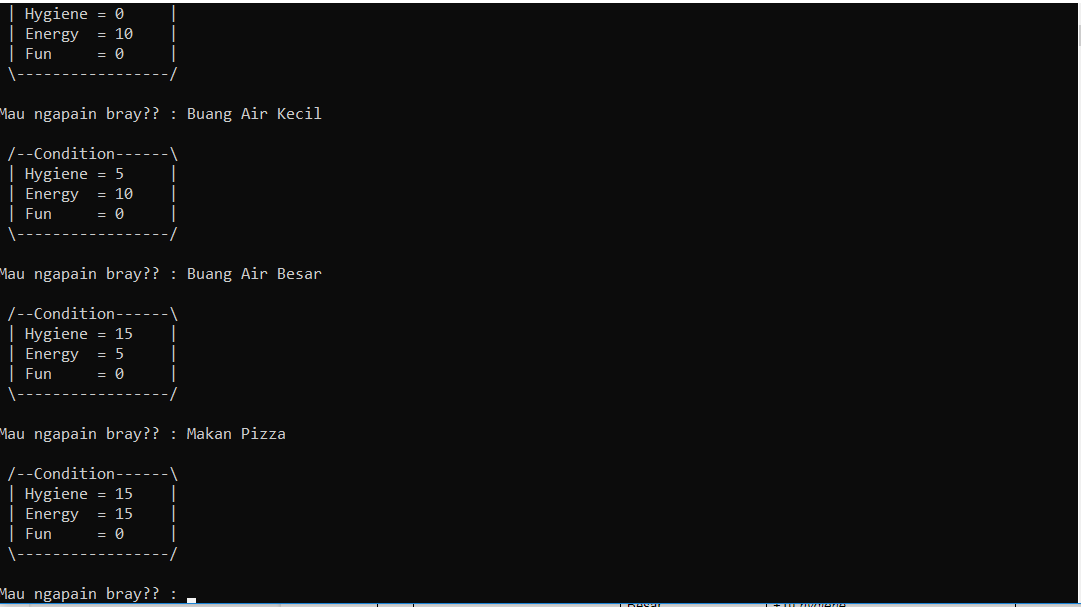
“Buang Air Kecil” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Buang Air Besar”



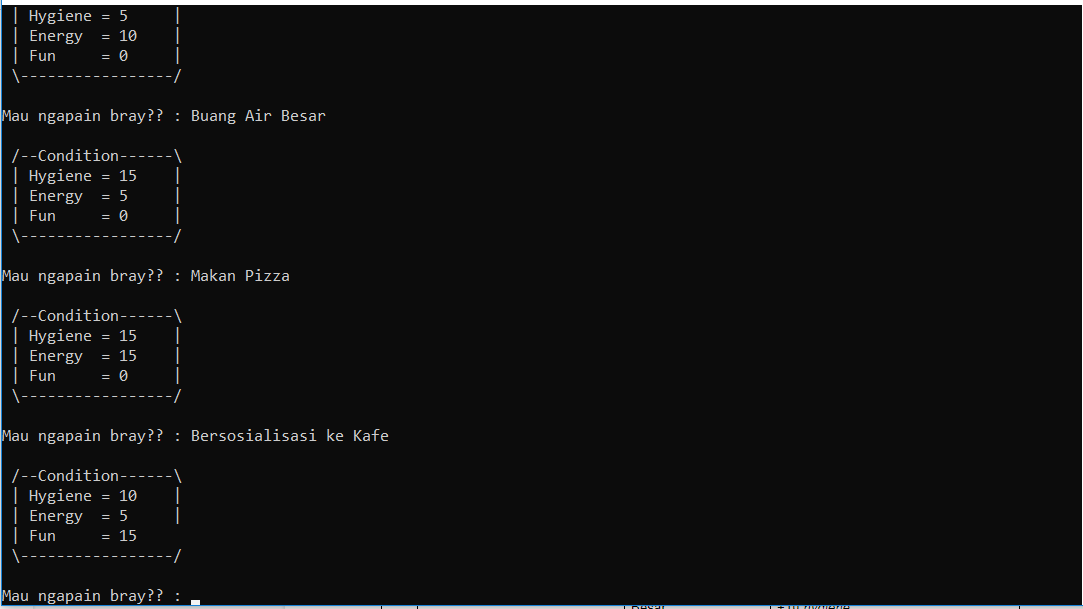
“Buang Air Besar” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Makan Pizza”



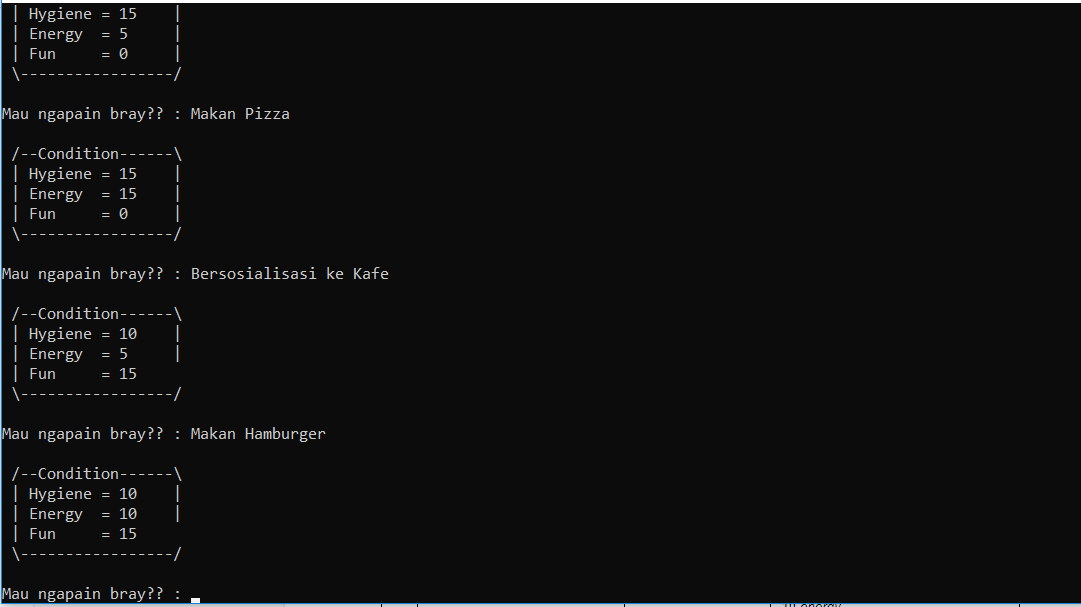
“Makan Pizza” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Bersosialisasi ke Kafe”



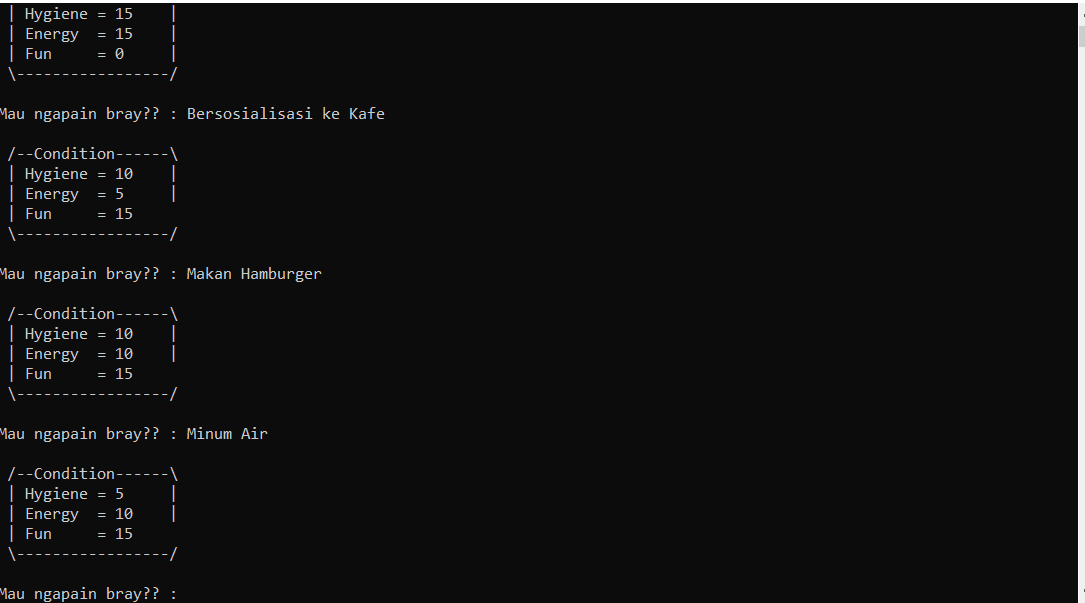
“Bersosialisasi ke Kafe” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. Makan Hamburger



“Makan Hamburger” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. Minum Air



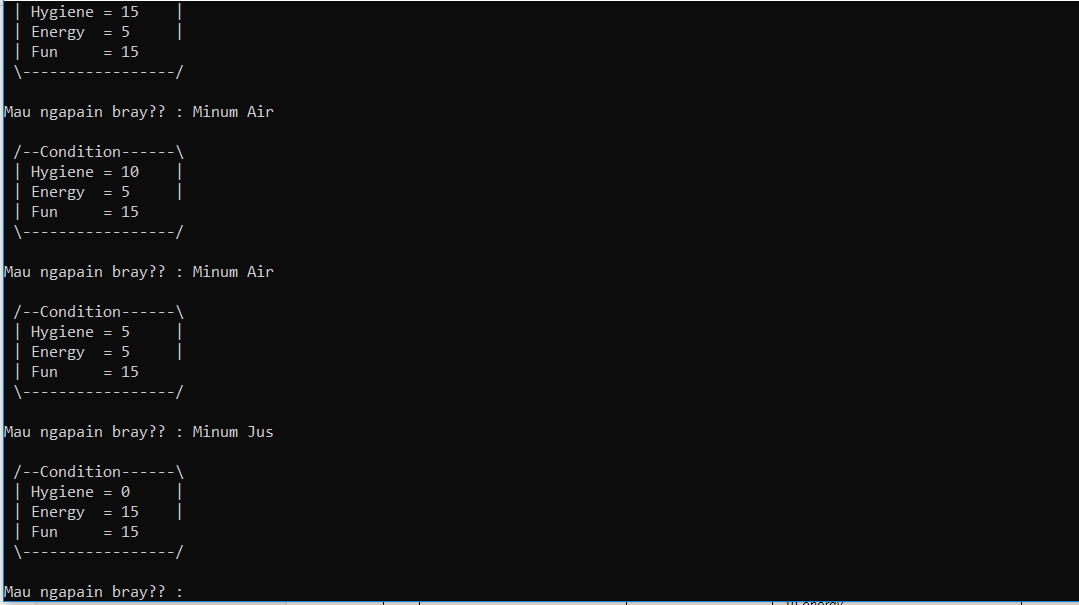
“Minum Air” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Makan Steak and Beans”



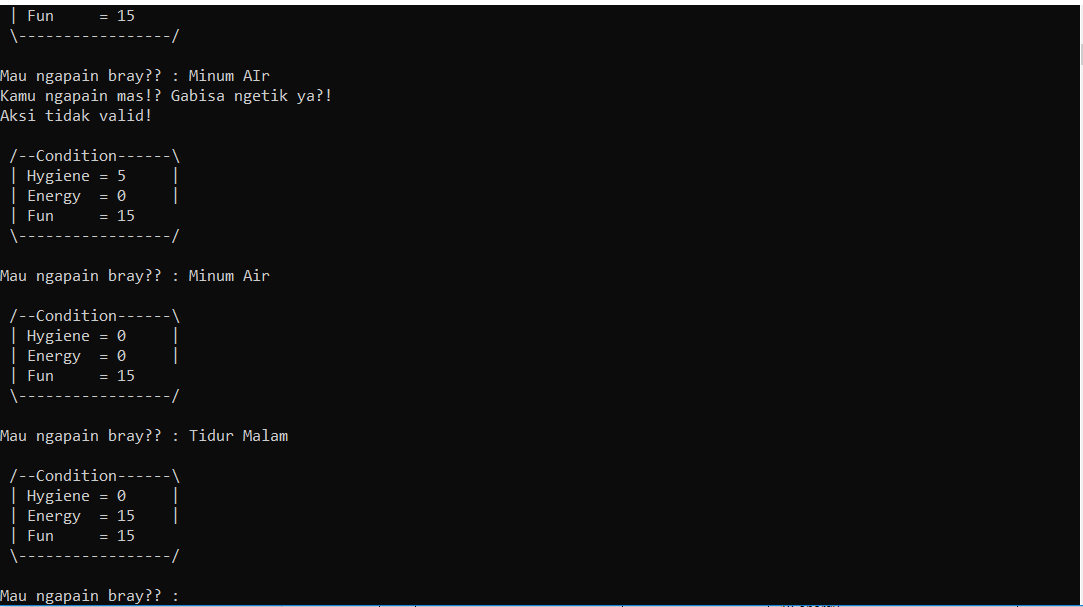
“Makan Steak and Beans” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Minum Jus”



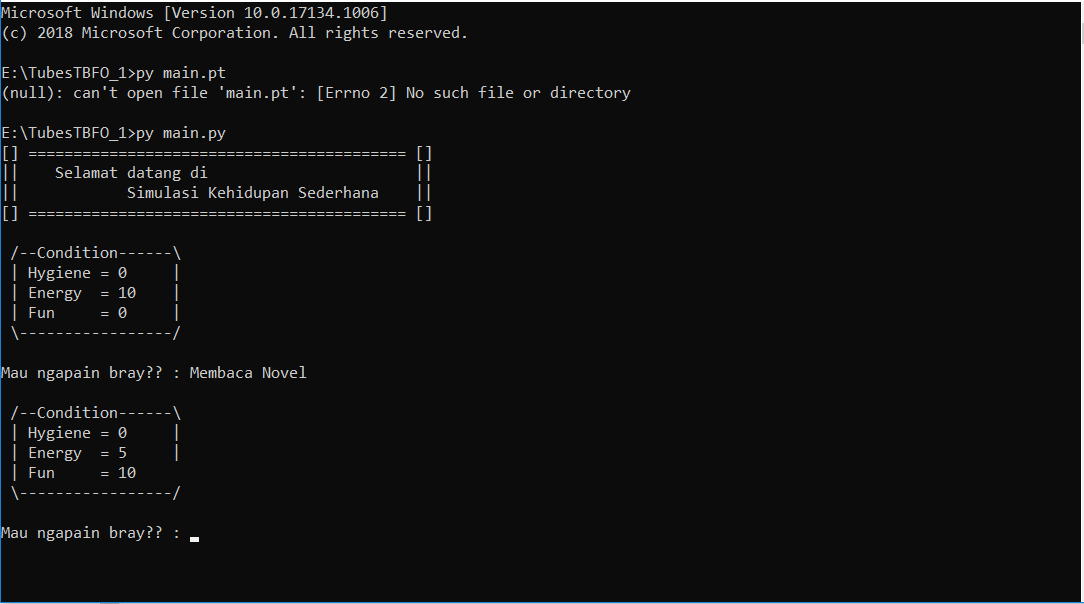
“Minum Jus” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Tidur Malam”



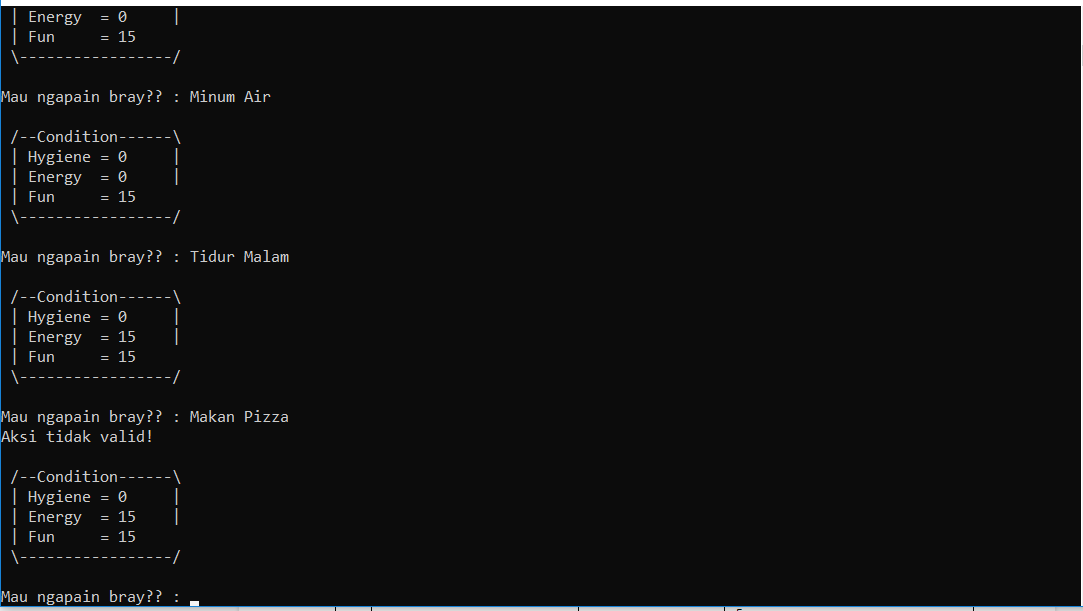
“Tidur Malam” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* + 1. “Membaca Novel”



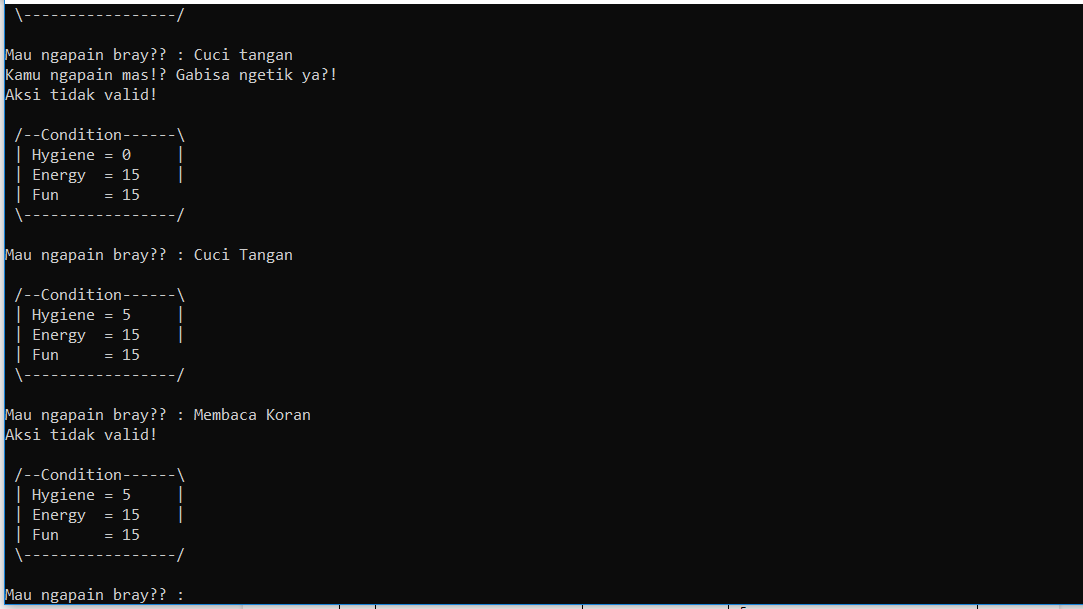
“Membaca Novel” merupakan aksi yang valid dan memenuhi syarat karena setelah dilakukan aksi tersebut, seluruh keadaan (hygiene, energy, fun) masih lebih dari 0 dan kurang dari 15.

* 1. Masukkan Input yang Valid Namun Tidak Memenuhi Syarat
     1. “Makan Pizza”



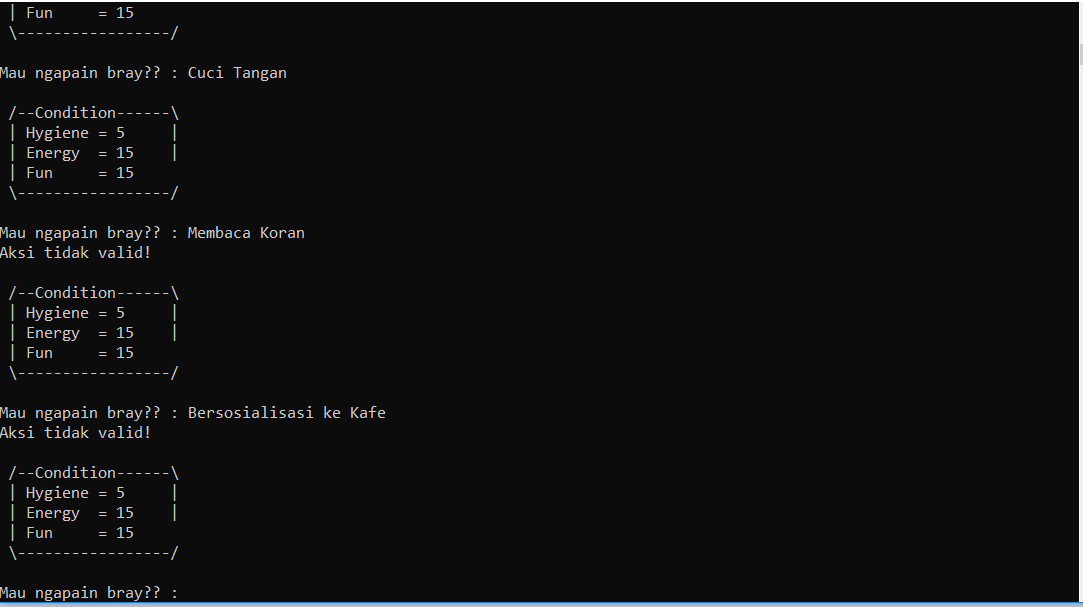
“Makan Pizza” merupakan input yang valid, namun “Makan Pizza” akan menambah 10 energy, sedangkan jika aksi tersebut dilakukan, maka energy akan melebihi 15, sehingga aksi yang dilakukan tidak valid dan state tidak berubah.

* + 1. “Membaca Koran”



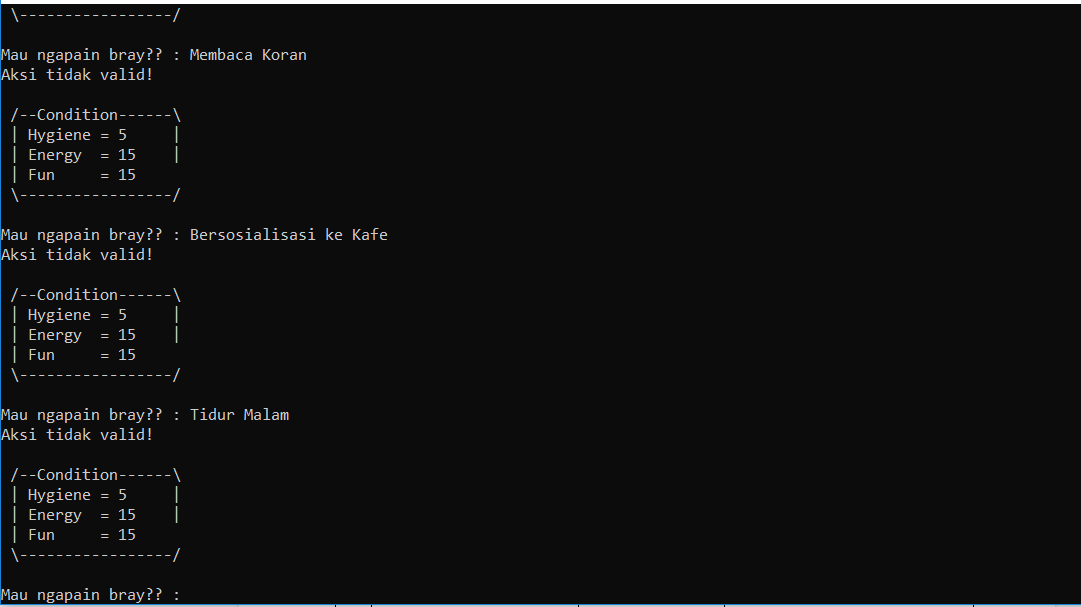
“Membaca Koran” merupakan input yang valid, namun “Membaca Koran” akan menambah 5 fun dan mengurangi 5 energy, sedangkan jika aksi tersebut dilakukan, maka fun akan melebihi 15, sehingga aksi yang dilakukan tidak valid dan state tidak berubah.

* + 1. “Bersosialisasi ke Kafe”



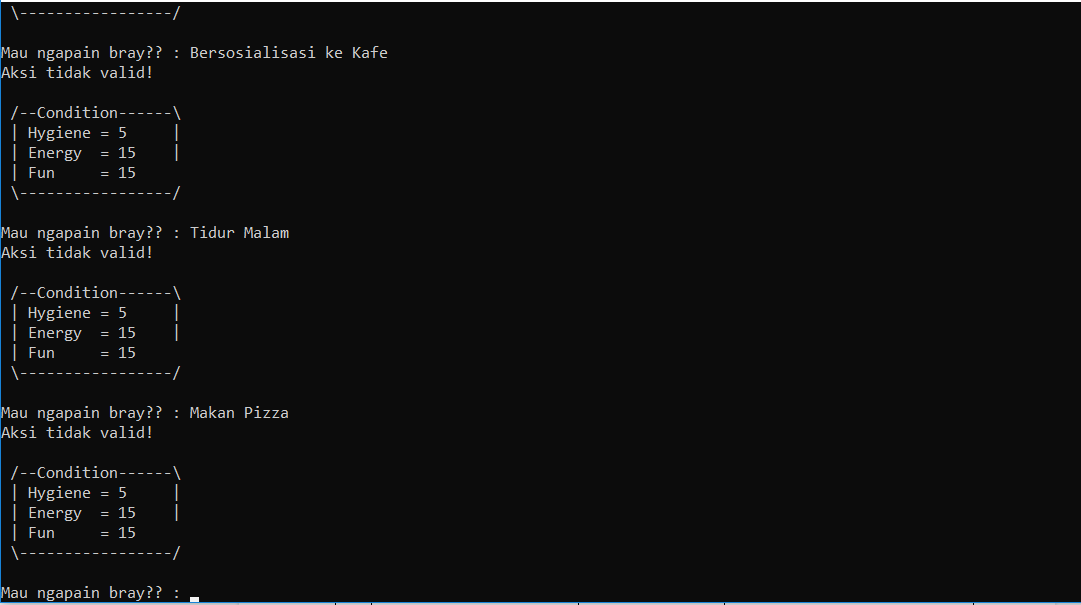
“Bersosialisasi ke Kafe” merupakan input yang valid, namun “Bersosialisasi ke Kafe” akan menambah 15 fun, mengurangi 10 energy dan mengurangi 5 hygiene, sedangkan jika aksi tersebut dilakukan, maka fun akan melebihi 15, sehingga aksi yang dilakukan tidak valid dan state tidak berubah.

* + 1. “Tidur Malam”



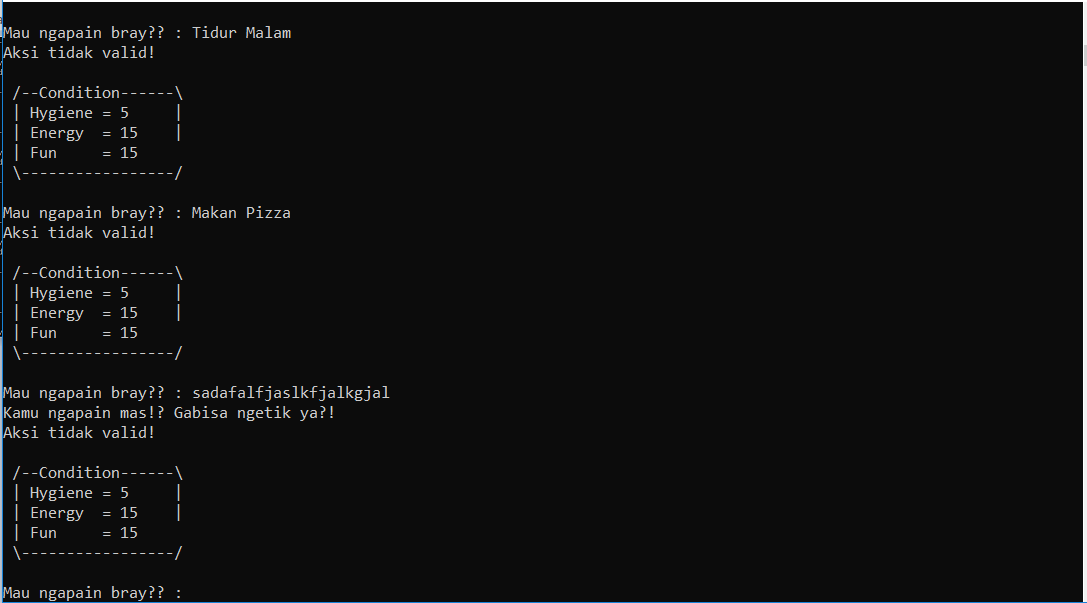
“Tidur Malam” merupakan input yang valid, namun “Tidur Malam” akan menambah 10 energy sedangkan jika aksi tersebut dilakukan maka energy akan melebihi 15, sehingga aksi yang dilakukan tidak valid dan state tidak berubah.

* + 1. “Makan Pizza”



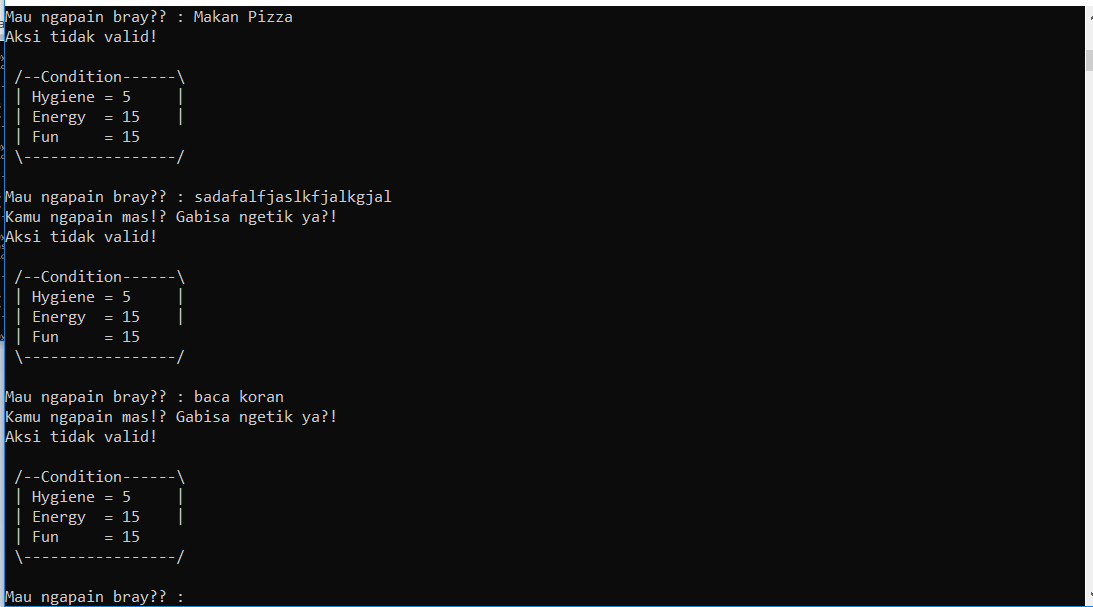
“Makan Pizza” merupakan input yang valid, namun “Makan Pizza” akan menambah 10 energy sedangkan jika aksi tersebut dilakukan maka energy akan melebihi 15, sehingga aksi yang dilakukan tidak valid dan state tidak berubah.

* 1. Aksi Tidak Valid
     1. “sadafalfjaslkfjalkgjal”



“sadafalfjaslkfjalkgjal” merupakan input yang tidak valid, karena aksi tersebut tidak terdefinisi, sehingga state dari keadaannya tidak berubah.

* + 1. “baca koran”



“baca koran” merupakan input yang tidak valid, karena aksi tersebut tidak terdefinisi, sehingga state dari keadaannya tidak berubah.

* 1. Menang



Kondisi ini akan tercapaai saat pemain mendapatkan hygiene 15, energy 15, dan fun 15. Saat state berubah ke Final State yang berisi kondisi menang (q63), interface akan berubah menjadi seperti gambar diatas dan loop akan berhenti.

* 1. Kalah



Kondisi ini akan tercapaai saat pemain mendapatkan hygiene 0, energy 0, dan fun 0. Saat state berubah ke Final State yang berisi kondisi kalah (q0), interface akan berubah menjadi seperti gambar diatas dan loop akan berhenti.

**DAFTAR PUSTAKA**

<https://www.academia.edu/33302393/Deterministic_Finite_Automata_DFA_Non-Deterministic_Automata_NFA>

<https://www.academia.edu/11881230/DETERMINISTIC_FINITE_AUTOMATA>

<https://guruinformatika.blogspot.com/2015/04/contoh-nfa-dan-dfa-di-automata.html>

<https://en.wikipedia.org/wiki/Deterministic_finite_automaton>

<https://repository.unikom.ac.id/36690/1/Pertemuan5.pdf>

<http://cdndata.telkomuniversity.ac.id/pjj/15161/CSG3D3/MDS/COURSE_MATERIAL/z1443568886f3a5e2a2d30275753fe19d0b88da12fc.pdf>

Hopcroft, John E dkk. 2001. *Introduction to Automata Theory, Languages, and Computation, 2nd Edition.* Inggris.