

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK**

**MODUL II
AUTOMATA DAN TABLE-DRIVEN CONSTRUCTION**



Disusun Oleh :
Dwi Candra Pratama/2211104035
S1SE-06-02

Asisten Praktikum :
Muhamad Taufiq Hidayat

Dosen Pengampu :
Riyan Dwi Yulian Prakoso, S.Kom., M.Kom.

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
DIREKTORAT TELKOM KAMPUS PURWOKERTO
2025**

PENDAHULUAN

A. GUIDED

1. Automata-based Construction (FSM)

Automata Construction atau Automata-based programming adalah salah satu paradigma pemrograman dimana program dianggap seperti finite-state machine(FSM) atau formal automaton lainnya yang memiliki berbagai state-state yang saling berkaitan dan memiliki aturan tertentu yang jelas. Berikut ini indikator utama dalam Automata-based programming:

1. Jangka waktu eksekusi program dipisahkan dengan jelas pada state yang ada dan tidak terjadinya eksekusi yang overlapping pada state state yang ada.
2. Semua komunikasi antara state-state yang ada (perpindahan antar state) hanya dapat dilakukan secara eksplisit yang disimpan pada suatu global variable.

```
1  const readline = require("readline");
2
3  const rl = readline.createInterface({
4    input: process.stdin,
5    output: process.stdout
6  });
7
8  const State = {
9    START: "START",
10   GAME: "GAME",
11   PAUSE: "PAUSE",
12   EXIT: "EXIT"
13 };
14
15 let state = State.START;
16
17 Tabnine | Edit | Test | Explain | Document
18 function runStateMachine() {
19   console.log(`${state} SCREEN`);
20   rl.question("Enter Command: ", (command) => {
21     switch (state) {
22       case State.START:
23         if (command === "ENTER") state = State.GAME;
24         else if (command === "QUIT") state = State.EXIT;
25         break;
26       case State.GAME:
27         if (command === "ESC") state = State.PAUSE;
28         break;
29       case State.PAUSE:
30         if (command === "BACK") state = State.GAME;
31         else if (command === "HOME") state = State.START;
32         else if (command === "QUIT") state = State.EXIT;
33         break;
34     }
35     if (state !== State.EXIT) {
36       runStateMachine();
37     } else {
38       console.log("EXIT SCREEN");
39       rl.close();
40     }
41   });
42 }
43 runStateMachine();
```

2. Table-driven Construction

Table-driven Construction adalah metode yang memungkinkan pencarian informasi menggunakan tabel sebagai pengganti pernyataan logika seperti *if* dan *case*. Pendekatan ini berguna dalam mengorganisir data dengan lebih terstruktur dan efisien, terutama ketika logika yang digunakan semakin kompleks. Dalam kasus sederhana, penggunaan pernyataan logika mungkin lebih praktis, tetapi ketika jumlah kondisi meningkat, pendekatan berbasis tabel dapat menyederhanakan kode, meningkatkan keterbacaan, serta mempermudah pemeliharaan.

Table-driven memiliki dua hal yang perlu diperhatikan sebelum diimplementasikan. Pertama kita perlu menentukan bagaimana mencari entri pada tabel. Kedua beberapa tipe data tidak bisa digunakan untuk mencari entri pada table secara langsung. Secara umum cara pencarian entri pada table-driven dibagi menjadi tiga yaitu :

1. Direct Access.
2. Indexed Access.
3. Stair-step Access.

```
Tabnine | Edit | Test | Explain | Document
1 function getDaysPerMonth(month) {
2     const daysPerMonth = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
3     return daysPerMonth[month - 1] || "Invalid month";
4 }
5
6 console.log(getDaysPerMonth(7)); // Output: 28
7 console.log(getDaysPerMonth(13)); // Output: Invalid month
```

3. Stair-step Access:

Salah satu metode dalam table-driven yang berguna untuk mempermudah pencarian entri apabila terdapat kasus dimana nilai yang didapatkan berdasarkan range yang telah ada seperti contoh index nilai mahasiswa. Untuk lebih jelasnya perhatikan potongan kode berikut.

```
1 function getGradeByScore(studentScore) {
2     const grades = ["A", "AB", "B", "BC", "C", "D", "E"];
3     const rangeLimit = [80, 70, 65, 60, 50, 40, 0];
4
5     for (let i = 0; i < rangeLimit.length; i++) {
6         if (studentScore >= rangeLimit[i]) {
7             return grades[i];
8         }
9     }
10    return "E";
11 }
12
13 console.log(getGradeByScore(75)); // Output: AB
14 console.log(getGradeByScore(45)); // Output: D
```

BAB II

PENUGASAN (UNGUIDED)

Soal 1: Automata-based Construction (FSM)

Sebuah game memiliki tiga state utama:

- START (awal permainan)
- PLAYING (sedang bermain)
- GAME OVER (permainan berakhir)

Aturan transisi antar state:

1. Dari START, jika pemain mengetik "PLAY", permainan masuk ke state PLAYING.
2. Dari PLAYING, jika pemain mengetik "LOSE", permainan masuk ke state GAME OVER.
3. Dari GAME OVER, jika pemain mengetik "RESTART", permainan kembali ke state START.
4. Pemain bisa keluar kapan saja dengan mengetik "EXIT".

SourCodenya

```
14  const readline = require("readline");
15
16  const rl = readline.createInterface({
17    input: process.stdin,
18    output: process.stdout
19  });
20
21  const State = {
22    START: "START",
23    PLAYING: "PLAYING",
24    GAME_OVER: "GAME_OVER",
25    EXIT: "EXIT"
26  };
27
28  let state = State.START;
29
30  function runStateMachine() {
31    console.log(`${state} SCREEN`);
32    rl.question("Enter Command: ", (command) => {
33      switch (state) {
34        case State.START:
35          if (command === "PLAY") state = State.PLAYING;
36          else if (command === "EXIT") state = State.EXIT;
37          break;
38        case State.PLAYING:
39          if (command === "LOSE") state = State.GAME_OVER;
40          else if (command === "EXIT") state = State.EXIT;
41          break;
42        case State.GAME_OVER:
43          if (command === "RESTART") state = State.START;
44          else if (command === "EXIT") state = State.EXIT;
45          break;
46      }
47      if (state !== State.EXIT) {
48        runStateMachine();
49      } else {
50        console.log("EXIT SCREEN");
51        rl.close();
52      }
53    });
54  }
55
56  runStateMachine();
```

OutPutnya:

```
1104035_SE02\PraktikumPert2\UnGuided\Soal1.js"
START SCREEN
Enter Command: PLAY
PLAYING SCREEN
Enter Command: LOSE
GAME_OVER SCREEN
Enter Command: RESTART
START SCREEN
Enter Command: EXIT
EXIT SCREEN
PS D:\Semester 6\KONSTRUKSI PERANGKAT LUNAK\Praktikum\KPL_DwiCandraPratama_2211104035_SE02>
```