

Nama : Elvaretta Anantya Velya

NIM : 2211104074

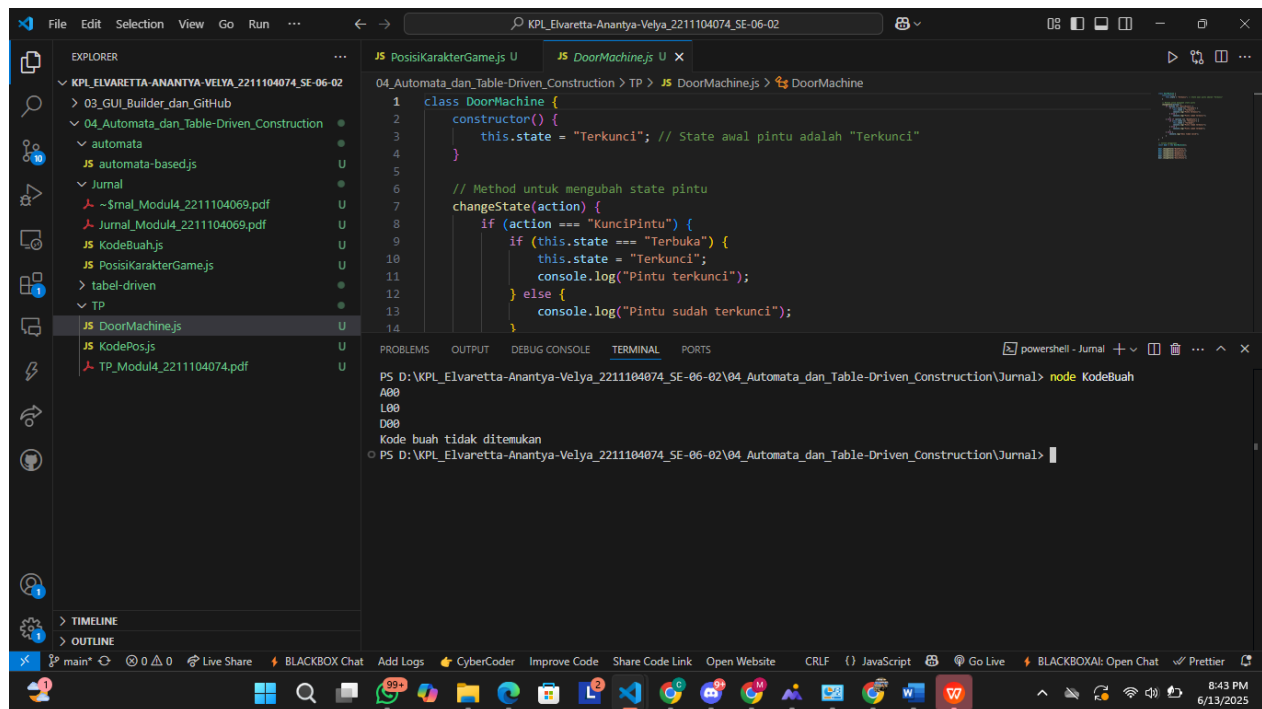
Kelas : SE0602

Kode Buah

Source Code

```
1  class KodeBuah {
2      constructor() {
3          this.kodeBuah = {
4              "Apel": "A00",
5              "Aprikot": "B00",
6              "Alpukat": "C00",
7              "Pisang": "D00",
8              "Paprika": "E00",
9              "Blackberry": "F00",
10             "Ceri": "H00",
11             "Kelapa": "I00",
12             "Jagung": "J00",
13             "Kurma": "K00",
14             "Durian": "L00",
15             "Anggur": "M00",
16             "Melon": "N00",
17             "Semangka": "O00"
18         };
19     }
20
21     getKodeBuah(namaBuah) {
22         return this.kodeBuah[namaBuah] || "Kode buah tidak ditemukan";
23     }
24 }
25
26 // Contoh penggunaan
27 const kodeBuah = new KodeBuah();
28 console.log(kodeBuah.getKodeBuah("Apel"));
29 console.log(kodeBuah.getKodeBuah("Durian"));
30 console.log(kodeBuah.getKodeBuah("Pisang"));
31 console.log(kodeBuah.getKodeBuah("strawberry"));
```

Output :



The screenshot shows the Visual Studio Code editor with a file named `DoorMachine.js` open. The code defines a `DoorMachine` class with a constructor and a `changeState` method. The terminal output shows the command `node KodeBuah` being executed, resulting in the message `Kode buah tidak ditemukan`.

```
1 class DoorMachine {
2   constructor() {
3     this.state = "Terkunci"; // State awal pintu adalah "Terkunci"
4   }
5
6   // Method untuk mengubah state pintu
7   changeState(action) {
8     if (action === "KunciPintu") {
9       if (this.state === "Terbuka") {
10        this.state = "Terkunci";
11        console.log("Pintu terkunci");
12      } else {
13        console.log("Pintu sudah terkunci");
14      }
15    }
16  }
17 }
```

```
PS D:\KPL_Elvareta-Anantya-Velya_2211104074_SE-06-02\04_Automata_dan_Table-Driven_Construction\Jurnal> node KodeBuah
AB0
LB0
DB0
Kode buah tidak ditemukan
PS D:\KPL_Elvareta-Anantya-Velya_2211104074_SE-06-02\04_Automata_dan_Table-Driven_Construction\Jurnal>
```

Penjelasan :

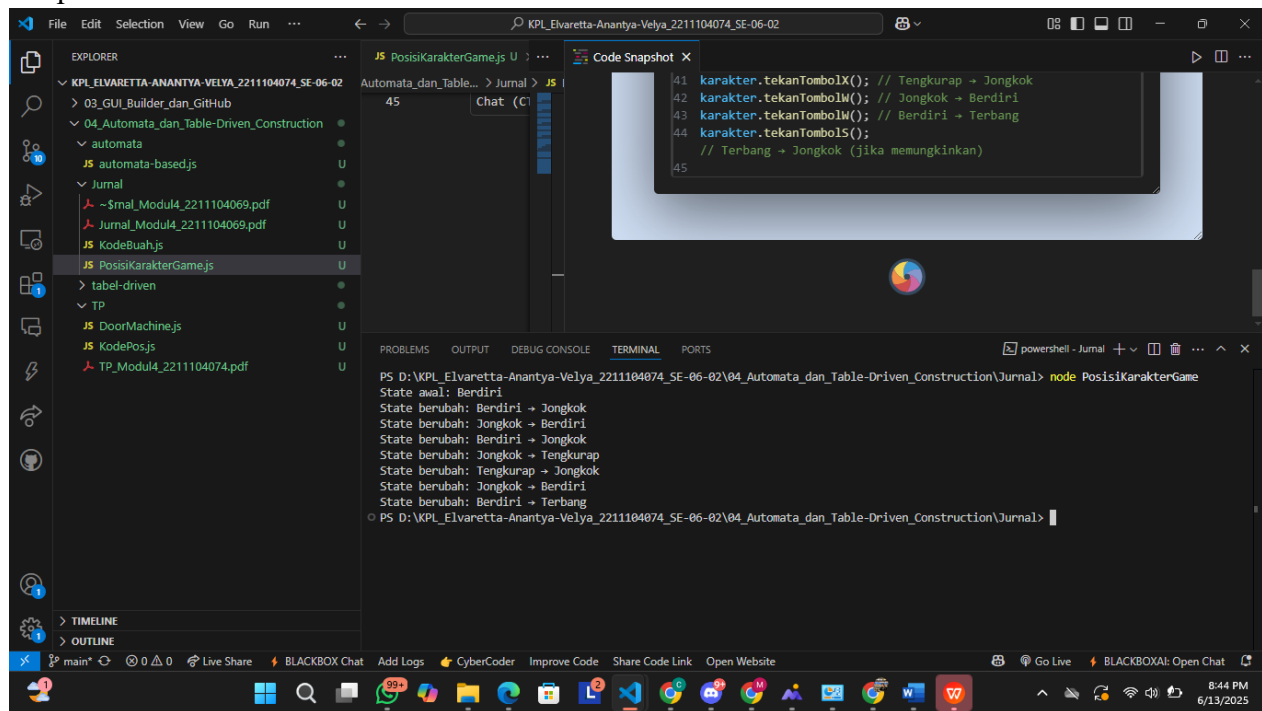
Program ini mengimplementasikan untuk mencari kode buah dengan menggunakan nama buahnya, method `getKodeBuah(namaBuah)` digunakan untuk mengembalikan kode berdasarkan dari nama yang diberikan, jika nama buahnya sesuai maka akan mengembalikan dengan kode buahnya, dan jika tidak sesuai maka akan mengembalikan pesan `'Kode buah tidak ditemukan'`.

Posisi Karakter Game

Source Code

```
1 class PosisiKarakterGame {
2   constructor() {
3     this.state = "Berdiri";
4     this.nim = 2211104074;
5   }
6
7   ubahState(stateBaru) {
8     console.log(`State berubah: ${this.state} → ${stateBaru}`);
9     this.state = stateBaru;
10  }
11
12  tekanTombolS() {
13    if (this.nim % 3 === 0) console.log(
14      "Tombol arah bawah ditekan.");
15    if (this.state === "Berdiri") this.ubahState("Jongkok");
16    else if (this.state === "Jongkok") this.ubahState(
17      "Tengkurap");
18  }
19  tekanTombolW() {
20    if (this.nim % 3 === 0) console.log(
21      "Tombol arah atas ditekan.");
22    if (this.state === "Jongkok") this.ubahState("Berdiri");
23    else if (this.state === "Berdiri") this.ubahState(
24      "Terbang");
25  }
26  tekanTombolX() {
27    if (this.state === "Tengkurap") {
28      this.ubahState("Jongkok");
29    }
30  }
31 }
32
33 // **Simulasi Program**
34 const karakter = new PosisiKarakterGame();
35 console.log(`State awal: ${karakter.state}`);
36
37 karakter.tekanTombolS(); // Berdiri → Jongkok
38 karakter.tekanTombolW(); // Jongkok → Berdiri
39 karakter.tekanTombolS(); // Berdiri → Jongkok
40 karakter.tekanTombolS(); // Jongkok → Tengkurap
41 karakter.tekanTombolX(); // Tengkurap → Jongkok
42 karakter.tekanTombolW(); // Jongkok → Berdiri
43 karakter.tekanTombolW(); // Berdiri → Terbang
44 karakter.tekanTombolS();
45 // Terbang → Jongkok (jika memungkinkan)
```

Output :



```
41 karakter.tekanTombolX(); // Tengkurap -> Jongkok
42 karakter.tekanTombolW(); // Jongkok -> Berdiri
43 karakter.tekanTombolW(); // Berdiri -> Terbang
44 karakter.tekanTombolS();
   // Terbang -> Jongkok (jika memungkinkan)
45
```

```
PS D:\KPL_Elvaretta-Anantya-Velya_2211104074_SE-06-02\04_Automata_dan_Table-Driven_Construction\Jurnal> node PosisiKarakterGame
State awal: Berdiri
State berubah: Berdiri -> Jongkok
State berubah: Jongkok -> Berdiri
State berubah: Berdiri -> Jongkok
State berubah: Jongkok -> Tengkurap
State berubah: Tengkurap -> Jongkok
State berubah: Jongkok -> Berdiri
State berubah: Berdiri -> Terbang
PS D:\KPL_Elvaretta-Anantya-Velya_2211104074_SE-06-02\04_Automata_dan_Table-Driven_Construction\Jurnal>
```

Penjelasan :

State awal pada program ini adalah 'berdiri', lalu ada transisi berdasarkan tombolnya seperti :

S -> 'berdiri' -> 'jongkok' -> 'tengkurap'

W -> 'jongkok' -> 'berdiri' -> 'terbang'

X -> 'tengkurap' -> 'jongkok'

Dengan aturan tambahan menggunakan NIM ($2211104074 \% 3 = 0$). Jadi memastikan setiap perubahan state dan aturan tambahannya sesuai dengan gambar diagram yang diberikan.