

TUGAS JURNAL

MODUL 4

2311104004

Marsella Dwi Julianti

SE07-01

Source Code:

```
1  const readline = require("readline");
```

Bagian ini, readline itu bawaan Node.js untuk membaca input dari user di terminal.

```
1  class KodeBuah {
2    constructor() {
3      this.buahDict = {
4        "Apel": "001", "Aprikot": "002", "Alpukat": "003", "Pisang": "004",
5        "Paprika": "005", "Kurma": "006", "Durian": "007", "Anggur": "008",
6        "Melon": "009", "Semangka": "000", "Mangga": "010", "Stroberi": "011",
7        "Nanas": "012", "Jeruk": "013", "Blueberry": "014"
8      };
9    }
10 }
```

Kode diatas merupakan kelas KodeBuah dimana terdapat this.buahDict yaitu objek yang menyimpan daftar buah dan kode buahnya.

```
1  cariKodeBuah(rl, callback) {
2    console.log("\n=== Sistem Pencarian Kode Buah ===");
3    console.log("Daftar buah yang tersedia: ", Object.keys(this.buahDict).join(", "));
4
5    const tanyaBuah = () => {
6      rl.question("\nMasukkan nama buah (atau ketik 'exit' untuk lanjut ke game): ", (input) => {
7        if (input.toLowerCase() === "exit") {
8          console.log("Pencarian kode buah selesai! Sekarang lanjut ke game.");
9          callback();
10         return;
11        }
12
13        const kode = this.buahDict[input] || "Kode tidak ditemukan";
14        console.log(`Kode Buah: ${kode}`);
15        tanyaBuah();
16      });
17    };
18
19    tanyaBuah();
20  }
21 }
22 }
```

Kode diatas merupakan metode cariKodeBuah, terdapat rl.question(...) yang membaca input user. Jika user mengetik exit, maka program beralih ke game menggunakan callback(). Dan jika user menginput nama buah, program akan mencocokkannya di buahDiet dan menampilkan kode buahnya. Lalu jika buah tidak ditemukan, akan mencetak "Kode tidak ditemukan".

```
1 class PosisiKarakterGame {
2   constructor() {
3     this.state = "Berdiri";
4     this.states = ["Berdiri", "Jongkok", "Tengkurap", "Terbang"];
5     this.NIM = 2311104004;
6   }
}
```

Kode diatas merupakan kelas PosisiKarakterGame, terdapat this.state yang menyimpan posisi awal karakter ("Berdiri"). Lalu ada this.states yang daftar state yang tersedia. Dan this.NIM yang disimpan dalam kelas, digunakan untuk menentukan aturan tambahan.

```
1 ubahState(stateBaru) {
2   if (!this.states.includes(stateBaru)) {
3     console.log("State tidak valid! Pilih: Berdiri, Jongkok, Tengkurap, Terbang");
4     return;
5   }
6
7   if (this.state === stateBaru) {
8     console.log(`Karakter sudah dalam posisi ${stateBaru}`);
9   } else {
10    console.log(`Transisi dari ${this.state} ke ${stateBaru}`);
11    this.state = stateBaru;
12
13    if (this.NIM % 3 === 1) {
14      if (stateBaru === "Berdiri") {
15        console.log("Posisi standby");
16      } else if (stateBaru === "Tengkurap") {
17        console.log("Posisi istirahat");
18      }
19    }
20  }
21 }
```

Kode diatas menggunakan metode ubahState, dimana nantinya bisa mengecek apakah state valid atau tidak. Jika state sudah sama dengan state sekarang, tidak akan perubahan. Jika berbeda, karakter akan berpindah ke state baru dan menampilkan transisi. Terdapat juga aturan tambahan berdasarkan NIM % 3, dimana jika NIM % 3 = 1, transisi ke "Berdiri" lalu tampilkan "Posisi standby". Jika NIM % 3 = 1, transisi ke "Tengkurap" lalu tampilkan "Posisi istirahat". Nah, NIM saya 2311104004 % 3 = 1.

```

1  start(r1) {
2      console.log("\n=== Simulasi PosisiKarakterGame ===");
3      console.log(`State awal: ${this.state}`);
4      console.log(`State yang tersedia: ${this.states.join(", ")}`);
5
6      const tanyaState = () => {
7          rl.question("\nMasukkan state baru (atau ketik 'exit' untuk keluar): ", (input) => {
8              if (input.toLowerCase() === "exit") {
9                  console.log("Game selesai. Terima kasih!");
10                 rl.close();
11                 return;
12             }
13             this.ubahState(input);
14             tanyaState();
15         });
16     };
17
18     tanyaState();
19 }
20 }

```

Kode diatas merupakan metode start, dimana nantinya menampilkan state awal dan state yang tersedia. Lalu meminta input user state yang baru. Jika input “exit”, program akan berhenti (rl.close()). Dan jika input valid, memanggil ubahState.

```

1  const rl = readline.createInterface({
2      input: process.stdin,
3      output: process.stdout
4  });
5

```

Kode diatas merupakan bagian menjalankan program dimana readline yang Membuat interface untuk membaca input user.

```

1  const kodeBuah = new KodeBuah();
2  kodeBuah.cariKodeBuah(rl, () => {
3      // **Langkah 2: Simulasi Game Karakter**
4      const game = new PosisiKarakterGame();
5      game.start(rl);
6  });

```

Kode diatas untuk memanggil cariKodeBuah(), dan jika user selesai (exit), lanjut ke PosisiKarakterGame().start().

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\SEMESTER 4 GENAP 2025\KONTRUKSI PERANGKAT LUNAK\KPL_MARSELLADWIJULIANTI_2311104004_SE0701> ls

Directory: D:\SEMESTER 4 GENAP 2025\KONTRUKSI PERANGKAT LUNAK\KPL_MARSELLADWIJULIANTI_2311104004_SE0701

Mode                LastWriteTime         Length Name
----                -
d-----         17/02/2025         16.08      01_Pengantar_Praktikum_KPL
d-----         27/02/2025         14.27      02_Pengenalan_IDE_dan_Pemrograman_NodeJS
d-----         08/03/2025         17.29      03_GUI
d-----         14/03/2025         17.12      04_OOP

PS D:\SEMESTER 4 GENAP 2025\KONTRUKSI PERANGKAT LUNAK\KPL_MARSELLADWIJULIANTI_2311104004_SE0701> cd 04_OOP
PS D:\SEMESTER 4 GENAP 2025\KONTRUKSI PERANGKAT LUNAK\KPL_MARSELLADWIJULIANTI_2311104004_SE0701\04_OOP> node jurnal_index.js

=== Sistem Pencarian Kode Buah ===
Daftar buah yang tersedia: Apel, Aprikot, Alpukat, Pisang, Paprika, Kurma, Durian, Anggur, Melon, Semangka, Mangga, Stroberi
, Nanas, Jeruk, Blueberry

Masukkan nama buah (atau ketik 'exit' untuk lanjut ke game): Semangka
Kode Buah: 000

Masukkan nama buah (atau ketik 'exit' untuk lanjut ke game): exit
Pencarian kode buah selesai! Sekarang lanjut ke game.

=== Simulasi PosisiKarakterGame ===
State awal: Berdiri
State yang tersedia: Berdiri, Jongkok, Tengkurap, Terbang

Masukkan state baru (atau ketik 'exit' untuk keluar): Jongkok
Transisi dari Berdiri ke Jongkok

Masukkan state baru (atau ketik 'exit' untuk keluar): Tengkurap
Transisi dari Jongkok ke Tengkurap
Posisi istirahat

Masukkan state baru (atau ketik 'exit' untuk keluar): exit
Game selesai. Terima kasih!
PS D:\SEMESTER 4 GENAP 2025\KONTRUKSI PERANGKAT LUNAK\KPL_MARSELLADWIJULIANTI_2311104004_SE0701\04_OOP> |
```

Pada output ini cara kerjanya yaitu nantinya user akan menginput dan mencari nama buah terlebih dahulu. Jika user menginput “exit”, pencarian nama buah selesai, dan langsung masuk ke game posisi karakter. Setelah selesai bermain game, program akan berhenti dan selesai.