

1. kodeBuah.js

```
1  class KodeBuah {
2    constructor() {
3      this.tableBuah = {
4        apel: "A00",
5        aprikot: "B00",
6        alpukat: "C00",
7        pisang: "D00",
8        paprika: "E00",
9        kurma: "K00",
10       durian: "L00",
11       anggur: "M00",
12       melon: "N00",
13       semangka: "O00",
14     };
15   }
16
17   getKodeBuah(namaBuah) {
18     return this.tableBuah[namaBuah.toLowerCase()] || "kode tidak ditemukan";
19   }
20 }
21
22 const kodeBuah = new KodeBuah();
23
24 console.log(kodeBuah.getKodeBuah("apel"));
25 console.log(kodeBuah.getKodeBuah("jeruk"));
26 console.log(kodeBuah.getKodeBuah("melon"));
27
```

Kode tersebut mendefinisikan sebuah kelas bernama KodeBuah yang berfungsi untuk menyimpan dan mengambil kode buah berdasarkan nama buahnya. Kelas ini memiliki sebuah konstruktor yang menginisialisasi sebuah objek tableBuah. Objek ini berisi pasangan kunci-nilai, di mana kunci adalah nama buah dalam huruf kecil, dan nilai adalah kode buah yang sesuai. Misalnya, "apel" memiliki kode "A00", "aprikot" memiliki kode "B00", dan seterusnya.

Kelas KodeBuah juga memiliki sebuah metode bernama getKodeBuah yang menerima nama buah sebagai argumen. Metode ini mengubah nama buah menjadi huruf kecil dan mengembalikan kode buah yang sesuai dari objek tableBuah. Jika nama buah tidak ditemukan dalam objek tableBuah, metode ini akan mengembalikan string "kode tidak ditemukan". Kode di luar kelas KodeBuah membuat sebuah objek kodeBuah dari kelas KodeBuah dan memanggil metode getKodeBuah dengan beberapa nama buah sebagai argumen. Hasil dari setiap pemanggilan metode kemudian dicetak ke konsol.

## 2. Posisi KarakterGame.js

```
1 class PosisiKarakterGame {
2   constructor() {
3     this.state = "Berdiri"; // State awal
4     this.transitions = {
5       Berdiri: { TombolS: "Jongkok", TombolW: "Terbang" },
6       Jongkok: { TombolW: "Berdiri", TombolS: "Tengkurap" },
7       Tengkurap: { TombolW: "Jongkok" },
8       Terbang: { TombolS: "Berdiri", TombolX: "Terbang" },
9     };
10  }
11
12  tekanTombol(tombol) {
13    console.log(`tombol ${tombol} ditekan`);
14    const nextState = this.transitions[this.state][tombol];
15    if (nextState) {
16      this.state = nextState;
17      console.log(
18        `Tombol arah ${tombol} ditekan, karakter sekarang dalam posisi: ${this.state}`
19      );
20    } else {
21      console.log(
22        `Tombol arah ${tombol} ditekan, tetapi karakter gagal bergerak`
23      );
24    }
25  }
26 }
27
28 // Contoh penggunaan
29 const karakter = new PosisiKarakterGame();
30 karakter.tekanTombol("TombolS");
31 karakter.tekanTombol("TombolS");
32 karakter.tekanTombol("TombolW");
33 karakter.tekanTombol("TombolW");
34 karakter.tekanTombol("TombolW");
35 karakter.tekanTombol("TombolS");
36
```

Kode tersebut mendefinisikan sebuah kelas bernama `PosisiKarakterGame` yang merepresentasikan posisi karakter dalam sebuah permainan. Karakter memiliki beberapa state (posisi) seperti "Berdiri", "Jongkok", "Tengkurap", dan "Terbang". State awal karakter adalah "Berdiri". Kelas ini juga memiliki objek `transitions` yang mendefinisikan transisi antar state berdasarkan tombol yang ditekan. Misalnya, jika karakter dalam state "Berdiri" dan tombol "TombolS" ditekan, maka state karakter akan berubah menjadi "Jongkok".

Metode `tekanTombol` dalam kelas ini menerima input berupa tombol yang ditekan. Metode ini kemudian mencari transisi state yang sesuai berdasarkan state karakter saat ini dan tombol yang ditekan. Jika transisi ditemukan, state karakter akan diubah dan pesan log akan ditampilkan. Jika tidak, pesan log yang menyatakan bahwa karakter gagal bergerak akan ditampilkan. Contoh penggunaan di bagian bawah kode menunjukkan bagaimana objek `PosisiKarakterGame` dibuat dan bagaimana metode `tekanTombol` digunakan untuk mengubah state karakter.