

KODE BUAH

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

class KodeBuah {
  constructor() {
    // data kode buah yang tersedia yee
    this.dataKodeBuah = {
      "Apel": "A00",
      "Aprikot": "B00",
      "Alpukat": "C00",
      "Pisang": "D00",
      "Paprika": "E00",
      "Kurma": "K00",
      "Durian": "L00",
      "Anggur": "M00",
      "Melon": "N00",
      "Semangka": "O00",
      "Mangga": "P00",
      "Stroberi": "Q00",
      "Nanas": "R00",
      "Jeruk": "S00",
      "Blueberry": "T00"
    };
  }

  getKodeBuah(namaBuah) {
    let formattedNamaBuah = namaBuah.charAt(0).toUpperCase() +
namaBuah.slice(1).toLowerCase();

    return this.dataKodeBuah[formattedNamaBuah] || "Kode buah tidak ditemukan. Coba
lagi!";
  }
}
```

```

tampilkanDaftarBuah() {
    console.log("Daftar buah yang tersedia:");
    console.log(Object.keys(this.dataKodeBuah).join(", "));
}

run() {
    const rl = readline.createInterface({
        input: process.stdin,
        output: process.stdout
    });

    console.log("Selamat datang di sistem pencarian kode buah!");
    this.tampilkanDaftarBuah();

    const tanya = () => {
        rl.question("\nMasukkan nama buah (atau ketik 'exit' untuk keluar): ", (namaBuah) => {
            if (namaBuah.toLowerCase() === "exit") {
                console.log("Terima kasih telah menggunakan sistem ini!");
                rl.close();
            } else {
                console.log("Kode Buah:", this.getKodeBuah(namaBuah));
                tanya();
            }
        });
    };

    tanya();
}
}

```

```
module.exports = KodeBuah;
```

```
if (require.main === module) {  
  const kb = new KodeBuah();  
  kb.run();  
}
```

```
Daftar buah yang tersedia:  
Apel, Aprikot, Alpukat, Pisang, Paprika, Kurma, Durian, Anggur, Melon, Semangka, Mangga, Stroberi, Nanas, Jeruk, Blue  
berry  
  
Masukkan nama buah (atau ketik 'exit' untuk keluar): Alpukat  
Kode Buah: C00  
  
Masukkan nama buah (atau ketik 'exit' untuk keluar):
```

Penjelasan: Kode di atas adalah program pencarian kode buah menggunakan JavaScript. Kelas KodeBuah menyimpan daftar nama buah beserta kodenya dalam sebuah objek. Pengguna dapat memasukkan nama buah, lalu program akan menampilkan kode yang sesuai atau pesan jika buah tidak ditemukan. Program juga menyediakan daftar buah yang tersedia dan menggunakan readline untuk menerima input secara interaktif. Jika pengguna mengetik "exit", program akan berhenti.

POSISI KARAKTER GAME

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

```
class PosisiKarakterGame {  
  constructor(NIM) {  
    this.currentState = "Berdiri";  
    this.validStates = ["Berdiri", "Jongkok", "Tengkurap", "Terbang"];  
    this.NIM = NIM;  
    console.log("State awal:", this.currentState);  
  }  
  
  ubahState(newState) {  
    if (!this.validStates.includes(newState)) {  
      console.log("State tidak valid! Pilih antara:", this.validStates.join(", "));  
      return;  
    }  
  }  
}
```

```

    if (this.currentState !== newState) {
        console.log(`Transisi dari ${this.currentState} ke ${newState}`);

        if (this.NIM % 3 === 0) {
            console.log(newState === "Jongkok" || newState === "Tengkurap" ? "Tombol arah bawah ditekan" : "Tombol arah atas ditekan");
        } else if (this.NIM % 3 === 1) {
            if (newState === "Berdiri") console.log("Posisi standby");
            if (newState === "Tengkurap") console.log("Posisi istirahat");
        } else if (this.NIM % 3 === 2) {
            if (this.currentState === "Terbang" && newState === "Jongkok") console.log("Posisi landing");
            if (this.currentState === "Berdiri" && newState === "Terbang") console.log("Posisi take off");
        }

        this.currentState = newState;
    } else {
        console.log("Karakter sudah dalam posisi", newState);
    }
}

run() {
    const rl = readline.createInterface({
        input: process.stdin,
        output: process.stdout,
    });

    console.log("\n=== Simulasi PosisiKarakterGame ===");
    console.log("State yang tersedia:", this.validStates.join(", "));

```

```

const tanya = () => {
  rl.question("\nMasukkan state baru (atau ketik 'exit' untuk keluar): ", (state) => {
    if (state.toLowerCase() === "exit") {
      console.log("Terima kasih telah menggunakan simulator!");
      rl.close();
    } else {
      this.ubahState(state.charAt(0).toUpperCase() + state.slice(1).toLowerCase());
      tanya();
    }
  });
};

tanya();
}
}

// masukan nim nya disini
const NIM = 2311104007;
if (require.main === module) {
  const pg = new PosisiKarakterGame(NIM);
  pg.run();
}

```

```

State awal: Berdiri

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

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

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

```

Penjelasan : Kode di atas adalah program simulasi perubahan posisi karakter dalam game berdasarkan NIM pengguna. Kelas PosisiKarakterGame memiliki empat state: "Berdiri", "Jongkok", "Tengkurap", dan "Terbang". Saat state berubah, program akan menampilkan transisi yang terjadi dan memberikan respon sesuai dengan aturan berdasarkan NIM. Jika NIM habis dibagi 3, transisi akan menampilkan tombol arah yang ditekan. Jika sisa 1, ada pesan seperti "Posisi standby" atau "Posisi istirahat". Jika sisa 2, terdapat pesan "Posisi take off" atau "Posisi landing". Program menggunakan readline untuk menerima input pengguna dan memungkinkan perubahan state secara interaktif.