

JURNAL MODUL 4

2311104035

Reza Irawan

1. NamaBuah.js

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

class KodeBuah {
  constructor() {
    this.dataKodeBuah = {
      Apel: 'A00',
      Aprikot: 'B00',
      Alpukat: 'C00',
      Pisang: 'D00',
      Paprika: 'E00',
      Blackberry: 'F00',
      Ceri: 'H00',
      Kelapa: 'I00',
      Jagung: 'J00',
    };
  }

  getKodeBuah(namaBuah) {
    const formattedBuah = namaBuah.charAt(0).toUpperCase() +
      namaBuah.slice(1).toLowerCase();
    const kode = this.dataKodeBuah[formattedBuah];
    return kode ? kode : 'Kode buah tidak ditemukan';
  }

  listBuah() {
    console.log('\nDaftar Buah dan Kode:');
    for (const [buah, kode] of Object.entries(this.dataKodeBuah)) {
      console.log(`- ${buah}: ${kode}`);
    }
  }
}

const kodeBuah = new KodeBuah();

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

kodeBuah.listBuah();

rl.question('\nMasukkan nama buah untuk mencari kode: ', (input) => {
  console.log(`Kode Buah untuk ${input}:`, kodeBuah.getKodeBuah(input));
  rl.close();
});
```

Program Node.js ini menggunakan modul **readline** untuk menerima input dari pengguna dan mendefinisikan kelas **KodeBuah**, yang menyimpan daftar nama buah beserta kode uniknya dalam sebuah objek. Kelas ini memiliki dua metode utama, yaitu **getKodeBuah(namaBuah)** untuk mencari dan menampilkan kode unik berdasarkan nama buah yang dimasukkan pengguna, serta **listBuah()** untuk menampilkan seluruh daftar buah beserta kodenya. Saat dijalankan, program akan meminta pengguna memasukkan nama buah, kemudian mencari dan menampilkan kode uniknya jika ditemukan, atau memberikan pemberitahuan jika kode buah tidak tersedia.

Output:

```
Daftar Buah dan Kode:
- Apel: A00
- Aprikot: B00
- Alpukat: C00
- Pisang: D00
- Paprika: E00
- Blackberry: F00
- Ceri: H00
- Kelapa: I00
- Jagung: J00

Masukkan nama buah untuk mencari kode: Alpukat
Kode Buah untuk Alpukat: C00
PS D:\TUGAS KULIAH SEM 4\KPL PRKTIKUM\04 OOP>
```

2. PosisiKarakterGame.js

```
class PosisiKarakterGame {
  constructor(NIM) {
    this.state = 'Berdiri';
    this.NIM = NIM;
    this.modulus = NIM % 3;
    this.showState();
  }

  changeState(action) {
    const transitions = {
      Berdiri: { TombolW: 'Terbang', TombolS: 'Jongkok' },
      Jongkok: { TombolW: 'Berdiri', TombolS: 'Tengkurap' },
      Tengkurap: { TombolW: 'Jongkok' },
      Terbang: { TombolS: 'Berdiri' },
    };

    const nextState = transitions[this.state]?.[action];
    if (nextState) {
      this.handleSpecialOutput(this.state, nextState, action);
      this.state = nextState;
      this.showState();
    } else {
      console.log('Aksi tidak valid');
    }
  }

  handleSpecialOutput(prevState, nextState, action) {
    if (this.modulus === 0) {
      if (action === 'TombolS') console.log('Tombol arah bawah ditekan');
      if (action === 'TombolW') console.log('Tombol arah atas ditekan');
    } else if (this.modulus === 1) {
      if (nextState === 'Berdiri') console.log('Posisi standby');
      if (nextState === 'Tengkurap') console.log('Posisi istirahat');
    } else if (this.modulus === 2) {
      if (prevState === 'Terbang' && nextState === 'Jongkok') console.log('Posisi landing');
      if (prevState === 'Berdiri' && nextState === 'Terbang') console.log('Posisi take off');
    }
  }

  showState() {
    console.log(`Karakter dalam posisi: ${this.state}`);
  }
}

const NIM = 2311104033;
const karakter = new PosisiKarakterGame(NIM);

karakter.changeState('TombolS');
karakter.changeState('TombolW');
karakter.changeState('TombolS');
karakter.changeState('TombolS');
karakter.changeState('TombolW');
karakter.changeState('TombolW');
```

Kode ini mensimulasikan perubahan posisi karakter dalam game berdasarkan tombol yang ditekan, di mana **TombolW** digunakan untuk naik dan **TombolS** untuk turun. Posisi awal karakter adalah "**Berdiri**", dan setiap perubahan posisi mengikuti aturan transisi yang telah ditentukan. Selain itu, program menampilkan output tambahan berdasarkan hasil perhitungan **NIM % 3**, seperti pesan khusus saat karakter berpindah ke posisi tertentu. Saat dijalankan, karakter akan bergerak sesuai dengan input tombol, menampilkan status terkini, serta mencetak pesan tambahan jika kondisi modulus terpenuhi.

Outputnya:

```
● Karakter dalam posisi: Berdiri
  Tombol arah bawah ditekan
  Karakter dalam posisi: Jongkok
  Tombol arah atas ditekan
  Karakter dalam posisi: Berdiri
  Tombol arah bawah ditekan
  Karakter dalam posisi: Jongkok
  Tombol arah bawah ditekan
  Karakter dalam posisi: Tengkurap
  Tombol arah atas ditekan
  Karakter dalam posisi: Jongkok
  Tombol arah atas ditekan
  Karakter dalam posisi: Berdiri
○ PS D:\TUGAS KULIAH SEM 4\KPL PRKTIKUM\04_OOP>
```