

1. Kode
kodebuah

```
1  class KodeBuah {
    Codeium: Refactor | Explain | Generate JSDoc | X
2  constructor() {
3      this.tabel = {
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 }
    Codeium: Refactor | Explain | Generate JSDoc | X
16 getCodeBuah(buah) {
17     return this.tabel.hasOwnProperty(buah) ? this.tabel[buah] : "Kode buah tidak di
18 }
19 }
    Codeium: Refactor | Explain | Generate JSDoc | X
21 function main() {
22     const kodeBuahObj = new KodeBuah();
23     console.log("Kode Buah Apel:", kodeBuahObj.getCodeBuah("Apel"));
24     console.log("Kode Buah Aprikot:", kodeBuahObj.getCodeBuah("Aprikot"));
25     console.log("Kode Buah Alpukat:", kodeBuahObj.getCodeBuah("Alpukat"));
26     console.log("Kode Buah Pisang:", kodeBuahObj.getCodeBuah("Pisang"));
27     console.log("Kode Buah Paprika:", kodeBuahObj.getCodeBuah("Paprika"));
28     console.log("Kode Buah Kurma:", kodeBuahObj.getCodeBuah("Kurma"));
29     console.log("Kode Buah Durian:", kodeBuahObj.getCodeBuah("Durian"));
30     console.log("Kode Buah Anggur:", kodeBuahObj.getCodeBuah("Anggur"));
31     console.log("Kode Buah Melon:", kodeBuahObj.getCodeBuah("Melon"));
32     console.log("Kode Buah Semangka:", kodeBuahObj.getCodeBuah("Semangka"));
33 }
34
35 main();
```

class posisikaraktergame

```
class PosisiKarakterGame {
  Codeium: Refactor | Explain | Generate JSDoc | X
  constructor(nim) {
    this.nimMod = nim % 3;
    this.state = "Berdiri";
    if (this.nimMod === 1) {
      console.log("posisi standby");
    } else {
      console.log("State awal: " + this.state);
    }
  }
  Codeium: Refactor | Explain | Generate JSDoc | X
  changeState(newState) {
    const oldState = this.state;
    this.state = newState;
    if (this.nimMod === 1) {
      if (newState === "Berdiri") {
        console.log("posisi standby");
      } else if (newState === "Tengkurap") {
        console.log("posisi istirahat");
      } else {
        console.log("State changed from " + oldState + " to " + newState);
      }
    } else if (this.nimMod === 2) {
      if (oldState === "Terbang" && newState === "Jongkok") {
        console.log("posisi landing");
      } else if (oldState === "Berdiri" && newState === "Terbang") {
        console.log("posisi take off");
      } else {
        console.log("State changed from " + oldState + " to " + newState);
      }
    } else {
      console.log("State changed from " + oldState + " to " + newState);
    }
  }
  Codeium: Refactor | Explain | Generate JSDoc | X
  tombolS() {
    if (this.nimMod === 0) {
      console.log("tombol arah bawah ditekan");
    }
  }
  Codeium: Refactor | Explain | Generate JSDoc | X
  tombolW() {
    if (this.nimMod === 0) {
      console.log("tombol arah atas ditekan");
    }
  }
}

function main() {
  const nim = 2311104037;
  const game = new PosisiKarakterGame(nim);
  game.changeState("Tengkurap");
  game.changeState("Berdiri");
  game.changeState("Terbang");
}

main();
```

Output:

```
Kode Buah Apel: A00  
Kode Buah Aprikot: B00  
Kode Buah Alpukat: C00  
Kode Buah Pisang: D00  
Kode Buah Paprika: E00  
Kode Buah Kurma: K00  
Kode Buah Durian: L00  
Kode Buah Anggur: M00  
Kode Buah Melon: N00  
Kode Buah Semangka: O00
```

```
posisi standby  
posisi istirahat  
posisi standby  
State changed from Berdiri to Terbang
```

Penjelasan: KodeBuah menyimpan data nama buah dan kode buah. PosisiKarakterGame adalah kelas state-based untuk simulasi perubahan posisi karakter sesuai NIM (dengan NIM 2311104037 menghasilkan nimMod 1, sehingga aturan: "Berdiri" = "posisi standby", "Tengkurap" = "posisi istirahat").