

Nama : Andika Rifki Pratama

Kelas : SE-07-01

NIM : 2311104011

1. State Based

```
04_OOP > JurnalMOD3_2311104011 > StateBased > JS CharacterGamePos.js > ...
1  class CharacterGamePos {
2      constructor() {
3          this.state = "Berdiri";
4          this.transitions = {
5              "Berdiri": { "TombolS": "Jongkok", "TombolW": "Terbang" },
6              "Jongkok": { "TombolW": "Berdiri", "TombolS": "Tengkurap" },
7              "Tengkurap": { "TombolW": "Jongkok", "TombolX": "Terbang" },
8              "Terbang": { "TombolS": "Berdiri", "TombolW": "Jongkok" }
9          };
10     }
11
12     transition(tombol) {
13         const nextState = this.transitions[this.state]?.[tombol];
14
15         if (nextState) {
16             if (this.state === "Terbang" && nextState === "Jongkok") {
17                 console.log("Posisi landing");
18             } else if (this.state === "Berdiri" && nextState === "Terbang") {
19                 console.log("Posisi take off");
20             }
21
22             this.state = nextState;
23         } else {
24             console.log("Transisi tidak valid");
25         }
26     }
27 }
28
29 const sm = new CharacterGamePos();
30 sm.transition("TombolW");
31 sm.transition("TombolS");
32 sm.transition("TombolS");
33 sm.transition("TombolS");
34 sm.transition("TombolW");
35 sm.transition("TombolW");
36 sm.transition("TombolW");
```

Penjelasan :

Pada kode ini terdapat kelas CharacterGamePos di mana di dalamnya terdapat konstruktor untuk menginisialisasi state awal sebagai Berdiri serta mendefinisikan transisi antar state melalui tombol yang ditekan, seperti TombolS, TombolW, dan TombolX. Kemudian terdapat method position yang berfungsi untuk menangani perpindahan state, menampilkan status seperti "Posisi take off" saat bertransisi dari Berdiri ke Terbang dan "Posisi landing" saat bertransisi dari Terbang ke Jongkok, serta memperbarui state saat ini. Terakhir, terdapat instance sm yang digunakan untuk mensimulasikan perubahan state berdasarkan tombol yang ditekan.

Output :

```
PS D:\Campuss\SM T 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Andika Rifki Pratama_231104011_SE0701\04_OOP\JurnalMOD3_2311104011\StateBased> node CharacterGamePos.js
Posisi take off
Posisi take off
Posisi take off
Posisi landing
```

2. Table Driven

```

04_OOP > JurnalMOD3_2311104011 > Table > JS KodeBuah.js > ...
1 class KodeBuah {
2   constructor() {
3     this.fruitCodes = [
4       ["Apel", "A00"],
5       ["Aprikot", "B00"],
6       ["Alpukat", "C00"],
7       ["Pisang", "D00"],
8       ["Paprika", "E00"],
9       ["Blackberry", "F00"],
10      ["Ceri", "H00"],
11      ["Kelapa", "I00"],
12      ["Jagung", "J00"],
13      ["Kurma", "K00"],
14      ["Durian", "L00"],
15      ["Anggur", "M00"],
16      ["Melon", "N00"],
17      ["Semangka", "O00"]
18    ];
19  }
20
21  getKodeBuah(buah) {
22    return this.fruitCodes.find(item => item[0] === buah)?.[1] || "Kode buah tidak ditemukan";
23  }
24 }
25
26 const kodeBuah = new KodeBuah();
27 console.log(kodeBuah.getKodeBuah("Apel"));
28 console.log(kodeBuah.getKodeBuah("TidakAda"));

```

Penjelasan :

Pada kode ini terdapat kelas KodeBuah yang berfungsi untuk mencari kode buah berdasarkan nama buah yang diberikan, di mana di dalam konstruktor terdapat array fruitCodes yang menyimpan daftar nama buah beserta kode buahnya, kemudian terdapat method getKodeBuah yang mencari kode buah dengan find() dan mengembalikan kode buah jika ditemukan, atau menampilkan pesan "Kode buah tidak ditemukan" jika nama buah tidak ada dalam daftar, serta penggunaan instance kodeBuah untuk menguji kode dengan input "Apel" yang menghasilkan "A00" dan "TidakAda" yang menghasilkan pesan "Kode buah tidak ditemukan".

Output :

```

PS D:\Campuss\SMT_4\Konstruksi Perangkat Lunak\Praktikum\KPL_Andika Rifki Pratama_231104011_SE0701\04_OOP\JurnalMOD3_2311104011\Table> node KodeBuah.js
A00
Kode buah tidak ditemukan
PS D:\Campuss\SMT_4\Konstruksi Perangkat Lunak\Praktikum\KPL_Andika Rifki Pratama_231104011_SE0701\04_OOP\JurnalMOD3_2311104011\Table>

```