Kode

```
Codeium: Refactor | Explain
class DataGeneric {
    Codeium: Refactor | Explain | Generate JSDoc | ×
    constructor(data) {
        this.data = data;
    }

    Codeium: Refactor | Explain | Generate JSDoc | ×
    PrintData() {
        console.log(`Data yang tersimpan adalah: ${this.data}`);
    }
}

export { DataGeneric };
```

```
Codeium: Refactor | Explain
class HaloGeneric {
    Codeium: Refactor | Explain | Generate JSDoc | ×
    SapaUser(user) {
        console.log(`Halo user ${user}`);
    }
}

export { HaloGeneric };
```

```
import { HaloGeneric } from './HaloGeneric.js';
import { DataGeneric } from './DataGeneric.js';

const halo = new HaloGeneric();
halo.SapaUser("Resita");

const data = new DataGeneric("2311104037");
data.PrintData();
```

Output

```
Halo user Resita
Data yang tersimpan adalah: 2311104037
```

Penjelasan

1. Branch generic-method Pada branch ini dibuat class HaloGeneric yang memiliki method generic SapaUser<T>. Method ini menerima parameter generic dan mencetak ke console dengan format "Halo user X", di mana X adalah nilai dari parameter yang diberikan.
2. Branch generic-class Pada branch ini dibuat class DataGeneric dengan property data yang bertipe generic T. Class ini memiliki konstruktor untuk menginisialisasi data, dan method PrintData() yang mencetak ke console dengan format "Data yang tersimpan adalah: Y", di mana Y adalah isi dari property data.