

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK**

**MODUL III
GENERICS**



Disusun Oleh :
Dwi Candra Pratama/2211104035
S1SE-06-02

Asisten Praktikum :
Muhamad Taufiq Hidayat

Dosen Pengampu :
Riyan Dwi Yulian Prakoso, S.Kom., M.Kom.

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
DIREKTORAT TELKOM KAMPUS PURWOKERTO
2025**

PENDAHULUAN

Generics merupakan sebuah teknik dalam pemrograman yang memungkinkan pembuatan kode yang lebih fleksibel, dapat digunakan kembali, dan efisien dengan mendukung berbagai jenis data. Dalam JavaScript, meskipun tidak memiliki dukungan generics seperti yang ada di TypeScript atau C#, konsep ini masih dapat diterapkan melalui pendekatan berbasis class dan function. Teknik ini sangat bermanfaat dalam pengembangan perangkat lunak karena memungkinkan penggunaan kembali kode tanpa perlu menuliskan ulang fungsi atau class untuk setiap tipe data yang berbeda. Dengan menerapkan generics, pengembang dapat menghindari duplikasi kode, meningkatkan skalabilitas aplikasi, serta menciptakan struktur kode yang lebih terorganisir dan mudah dikelola.

A. GUIDED

1. Generic Class

Generic Class memungkinkan pengembang membuat struktur data yang fleksibel, mampu menyimpan berbagai jenis elemen dalam satu class yang sama. Hal ini sangat bermanfaat dalam pengembangan aplikasi karena mengurangi kebutuhan untuk membuat banyak class khusus bagi setiap tipe data yang berbeda. Salah satu keuntungan utama dari penggunaan Generic Class adalah efisiensi dalam mengelola data. Dengan memanfaatkan class generik, pengembang tidak perlu mendefinisikan class terpisah untuk setiap jenis data yang berbeda. Sebagai contoh, satu class dapat dirancang untuk menangani baik data numerik maupun string secara bersamaan, sehingga meningkatkan fleksibilitas dan kemudahan pemrograman.

SourceCode:

```
PraktikumPert3 > Guided > JS GenericClass.js > ...
1  class GenericList {
2      constructor() {
3          this.items = [];
4      }
5
6      add(item) {
7          this.items.push(item);
8      }
9
10     getAll() {
11         return this.items;
12     }
13 }
14
15 const list = new GenericList();
16 list.add(1);
17 list.add("Hello");
18 console.log(list.getAll());
```

OutPut:

```
um\KPL_DwiCandr
[ 1, 'Hello' ]
PS D:\Semester
```

2. Generic Function

Generic Function memungkinkan kita menulis fungsi yang dapat bekerja dengan berbagai jenis tipe data tanpa harus menduplikasi kode.

Dalam banyak kasus, fungsi generik dapat digunakan untuk meningkatkan fleksibilitas kode, misalnya dalam operasi pertukaran nilai atau pemrosesan koleksi data yang berbeda jenis.

```
1 function swap(a, b) {  
2     return [b, a];  
3 }  
4  
5 let x = 5, y = 10;  
6 [x, y] = swap(x, y);  
7 console.log(x, y); // Output: 10, 5
```

3. Generic Delegate

Generic Delegate adalah konsep dalam generics yang memungkinkan kita meneruskan fungsi sebagai parameter, sehingga meningkatkan modularitas dan fleksibilitas kode. Delegate sering digunakan dalam pengembangan perangkat lunak untuk mendukung desain berbasis event-driven, seperti callback function dalam JavaScript.

```
1 function genericDelegate(callback, value) {  
2     callback(value);  
3 }  
4  
5 genericDelegate(console.log, "Event Triggered");
```

BAB II PENUGASAN (UNGUIDED)

Tujuan:

Untuk memahami konsep penggunaan Git Branching serta penerapan generic class dan method dalam JavaScript secara terstruktur dan terpisah menggunakan branch.

Inisialisasi Proyek

- Membuat folder tpmodul5_2211104035 LINKNYA : https://github.com/DCandraPratama/tpmodul5_2211104035
- Inisialisasi Git: git init
- Membuat file README.md dan melakukan commit awal

Menghubungkan ke GitHub

- Membuat repository di GitHub
- Menambahkan remote:

```
git remote add origin https://github.com/DCandraPratama/tpmodul5_2211104035.git
```

- Push awal ke branch main:

```
git push -u origin main
```

Pembuatan Branch dan Implementasi

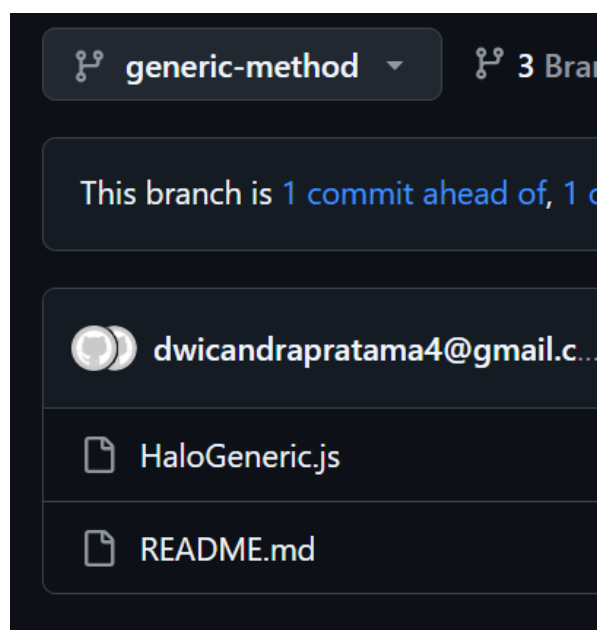
Branch: generic-method

- Membuat file HaloGeneric.js
- Isi:

```
class HaloGeneric {  
  SapaUser(user) {  
    console.log(`Halo user ${user}`);  
  }  
}  
  
function main() {  
  const halo = new HaloGeneric();  
  const nama = "Candra";  
  halo.SapaUser(nama);  
}  
  
main();
```

- Commit dan push:

```
git add HaloGeneric.js  
git commit -m "Menambahkan class HaloGeneric dan method SapaUser"  
git push origin generic-method
```



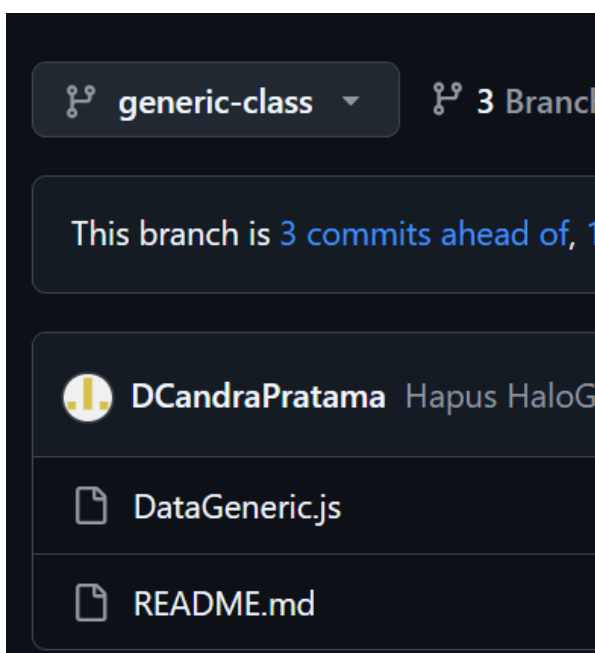
Branch: generic-class

- Membuat file DataGeneric.js
- Isi:

```
class DataGeneric {  
  constructor() {  
    this.data = null;  
  }  
  
  setData(data) {  
    this.data = data;  
  }  
  
  PrintData() {  
    console.log(`Data: ${this.data}`);  
  }  
}  
  
function main() {  
  const obj = new DataGeneric();  
  obj.setData("Candra");  
  obj.PrintData();  
}  
  
main();
```

- Commit dan push:

```
git add DataGeneric.js  
git commit -m "Menambahkan class DataGeneric dan method  
PrintData"  
git push origin generic-class
```



Linknya: https://github.com/DCandraPratama/tpmodul5_2211104035