

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT BERGERAK**

**MODUL V
GENERICS**



**Disusun Oleh :
Yogi Hafidh Maulana
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**

BAB I

PENDAHULUAN

A. DASAR TEORI

1. Pengantar Generics

Generics adalah teknik dalam pemrograman yang memungkinkan pembuatan kode yang lebih fleksibel, reusable, dan efisien dengan mendukung berbagai tipe data. Dalam bahasa seperti TypeScript dan C#, generics secara eksplisit didukung, sedangkan dalam JavaScript, generics dapat diterapkan menggunakan pendekatan berbasis class dan function. Penggunaan generics bertujuan untuk:

- a. Menghindari duplikasi kode dengan mendukung berbagai tipe data.
- b. Meningkatkan skalabilitas dan efisiensi dalam pengelolaan struktur data.
- c. Mempermudah pemeliharaan kode dengan meningkatkan keterbacaan dan modularitas.

2. Generic Class

Generic Class adalah konsep yang memungkinkan pembuatan struktur data yang dapat menyimpan berbagai tipe elemen tanpa mendefinisikan class yang berbeda untuk setiap tipe data. Dalam JavaScript, pendekatan ini dapat diterapkan menggunakan array atau class yang fleksibel.

3. Generic Function

Generic Function memungkinkan kita menulis fungsi yang dapat bekerja dengan berbagai jenis tipe data tanpa harus menduplikasi kode. Konsep ini sering digunakan dalam operasi yang melibatkan manipulasi data yang bervariasi.

4. Generic Delegate

Generic Delegate adalah konsep dalam generics yang memungkinkan penerusan fungsi sebagai parameter untuk meningkatkan modularitas kode. Pendekatan ini sering digunakan dalam callback function dalam JavaScript.

B. MAKSUD DAN TUJUAN

1. Maksud

Praktikum ini bertujuan untuk memahami dan mengimplementasikan konsep generics dalam JavaScript, yang memungkinkan pembuatan kode yang lebih fleksibel, reusable, dan efisien dengan mendukung berbagai tipe data tanpa menduplikasi kode.

2. Tujuan

Adapun tujuan dari praktikum ini adalah:

1. Memahami Konsep Generics – Mempelajari prinsip dasar generics dalam pemrograman serta manfaatnya dalam meningkatkan fleksibilitas kode.
2. Menerapkan Generic Class – Mengimplementasikan struktur data yang dapat menangani berbagai jenis tipe data dalam satu class yang sama.
3. Membuat Generic Function – Menulis fungsi generik yang dapat bekerja dengan berbagai tipe data tanpa perlu mendefinisikan fungsi baru untuk setiap tipe data.
4. Menggunakan Generic Delegate – Memahami dan menerapkan konsep delegate dalam generics untuk meningkatkan modularitas dan fleksibilitas kode.
5. Menganalisis Penerapan Generics dalam Aplikasi Nyata – Mengkaji bagaimana generics dapat digunakan dalam pengembangan sistem berbasis data seperti manajemen inventaris, pemrosesan transaksi, atau pembuatan framework.
6. Meningkatkan Keterampilan dalam Pengembangan Perangkat Lunak – Mempraktikkan pola desain generics untuk meningkatkan keterbacaan, pemeliharaan, dan efisiensi kode dalam pengembangan perangkat lunak berbasis JavaScript.

BAB II

IMPLEMENTASI (GUIDED)

A. Generic Class

Code

```
class GenericList {
  constructor() {
    this.items = [];
  }

  add(item) {
    this.items.push(item);
  }

  getAll() {
    return this.items;
  }
}

const list = new GenericList();
list.add(1);
list.add("Hello");
console.log(list.getAll());
```

Output

```
[Running] node "d:\PROJECT\Kontruksi Perangkat Lunak\KPL_YogiHafidhMaulana_2211104061_SE062\05_Generics\GUIDED\genericClass.js"
[ 1, 'Hello' ]

[Done] exited with code=0 in 0.148 seconds
```

Deskripsi Code

Generic Class dalam JavaScript memungkinkan penyimpanan berbagai jenis data dalam satu struktur. Class `GenericList` memiliki sebuah array `items` untuk menyimpan elemen-elemen yang ditambahkan. Method `add(item)` digunakan untuk menambahkan item ke dalam array, sedangkan `getAll()` digunakan untuk mengambil semua item yang telah dimasukkan. Setelah membuat objek `list`, kita menambahkan angka (1) dan string ("Hello") ke dalamnya, lalu mencetak semua isi array menggunakan `console.log(list.getAll())`, yang akan menghasilkan output `[1, "Hello"]`. Dengan cara ini, kita dapat menyimpan dan mengelola berbagai tipe data dalam satu class tanpa perlu membuat class baru untuk tiap jenis data.

B. Generic Function

Code

```
function swap(a, b) {  
  return [b, a];  
}  
  
let x = 5,  
    y = 10;  
[x, y] = swap(x, y);  
console.log(x, y);
```

Output

```
[Running] node "d:\PROJECT\Kontruksi Perangkat  
Lunak\KPL_YogiHafidhMaulana_2211104061_SE062\05_Generics\GUIDED\genericFunction.js"  
10 5  
  
[Done] exited with code=0 in 0.086 seconds
```

Deskripsi Code

Generic Function dalam JavaScript yang digunakan untuk menukar nilai dua variabel. Fungsi `swap(a, b)` menerima dua parameter dan mengembalikannya dalam urutan terbalik sebagai sebuah array `[b, a]`. Kemudian, dua variabel `x` dan `y` diinisialisasi dengan nilai 5 dan 10. Saat fungsi `swap(x, y)` dipanggil, hasilnya adalah array `[10, 5]`. Dengan menggunakan destructuring assignment `[x, y] = swap(x, y);`, nilai `x` menjadi 10 dan `y` menjadi 5. Akhirnya, `console.log(x, y);` mencetak hasilnya, yaitu 10 5. Kode ini berguna untuk menukar nilai variabel tanpa memerlukan variabel sementara tambahan.

C. Generic Delegate

Code

```
function genericDelegate(callback, value) {  
    callback(value);  
}  
  
genericDelegate(console.log, "Event Triggered");
```

Output

```
[Running] node "d:\PROJECT\Kontruksi Perangkat Lunak\KPL_YogiHafidhMaulana_2211104061_SE062\05_Generics\GUIDED\genericDelegate.js"  
Event Triggered  
  
[Done] exited with code=0 in 0.146 seconds
```

Deskripsi Code

Generic Delegate dalam JavaScript memungkinkan sebuah fungsi diteruskan sebagai parameter ke fungsi lain. Fungsi `genericDelegate(callback, value)` menerima dua parameter: `callback` (sebuah fungsi) dan `value` (nilai yang akan diproses). Di dalam fungsi, `callback(value)` dipanggil untuk mengeksekusi fungsi yang diberikan dengan nilai tersebut. Saat kode dijalankan, `genericDelegate(console.log, "Event Triggered")` akan memanggil `console.log("Event Triggered")`, sehingga mencetak "Event Triggered" ke konsol. Teknik ini sering digunakan dalam callback function untuk meningkatkan fleksibilitas dan modularitas kode.

BAB III PENUGASAN (UNGUIDED)

1. Soal 1 - Generic Method

Code

```
class HaloGeneric {  
  constructor(user) {  
    this.user = user;  
  }  
  
  SapaUser() {  
    console.log(`Halo user ${this.user}`);  
  }  
}  
  
const sapa = new HaloGeneric("Yogi");  
sapa.SapaUser();
```

Output

```
[Running] node "d:\PROJECT\Kontruksi Perangkat Lunak\KPL_YogiHafidhMaulana_2211104061_SE062\05_Generics\UNGUIDED\haloGeneric.js"  
Halo user Yogi  
  
[Done] exited with code=0 in 0.224 seconds
```

Deskripsi Code

Class HaloGeneric memiliki sebuah constructor yang menerima parameter user dan menyimpannya sebagai properti. Method SapaUser() digunakan untuk mencetak pesan "Halo user X", di mana X adalah nilai dari user. Saat objek sapa dibuat dengan nilai "Yogi", kemudian memanggil sapa.SapaUser();, program akan mencetak "Halo user Yogi" ke konsol. Kode ini menunjukkan bagaimana sebuah class dapat digunakan secara fleksibel dengan berbagai input tanpa perlu mengubah strukturnya.

2. Soal 2 - Generic Class

Code

```
class DataGeneric {  
  constructor(data) {  
    this.data = data;  
  }  
  
  PrintData() {  
    console.log(`Data yang tersimpan adalah: ${this.data}`);  
  }  
}  
  
const data = new DataGeneric("2211104061");  
data.PrintData();
```

Output

```
[Running] node "d:\PROJECT\Kontruksi Perangkat Lunak\KPL_YogiHafidhMaulana_2211104061_SE062\05_Generics\UNGUIDED\dataGeneric.js"  
Data yang tersimpan adalah: 2211104061  
  
[Done] exited with code=0 in 0.18 seconds
```

Deskripsi Code

Class DataGeneric memiliki sebuah constructor yang menerima parameter data dan menyimpannya sebagai properti. Method PrintData() digunakan untuk mencetak pesan "Data yang tersimpan adalah: X", di mana X adalah nilai dari data. Saat objek data dibuat dengan nilai "2211104061" dan method PrintData() dipanggil, program akan mencetak "Data yang tersimpan adalah: 2211104061" ke konsol. Kode ini menunjukkan bagaimana class dapat digunakan untuk menangani berbagai jenis data tanpa perlu mengubah strukturnya.