

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT BERGERAK**

**MODUL V
GENERICS**



Disusun Oleh :

Dhiya Ulhaq R

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. Konsep ini sangat berguna dalam pengembangan perangkat lunak karena memungkinkan penggunaan kembali kode tanpa harus menuliskan ulang fungsi atau class untuk setiap tipe data yang berbeda. Dengan generics, kita dapat menghindari duplikasi kode, meningkatkan skalabilitas aplikasi, dan membuat struktur kode yang lebih mudah dipahami serta dikelola.

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

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.

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.

BAB II

IMPLEMENTASI (GUIDED)

A. Generic Class

Source 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

```
C:\Program Files\nodejs\node.exe .\05_Generics\Guided\Class.js
> (2) [1, 'Hello']
```

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: 10, 5
```

Output

```
C:\Program Files\nodejs\node.exe .\05_Generics\Guided\Function.js  
10 5
```

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

```
C:\Program Files\nodejs\node.exe .\05_Generics\Guided\Delegate.js  
Event Triggered
```

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)

Soal 1 – Halo Generic

Code

```
class HaloGeneric {  
  SapaUser(x) {  
    console.log(`Halo user ${x}`);  
  }  
}  
  
const halo = new HaloGeneric();  
  
halo.SapaUser("Dhiya");  
  
console.log("=== Code Execution Successful ===");
```

Output

```
C:\Program Files\nodejs\node.exe .\05_Generics\Unguided\HaloGeneric.js  
Halo user Dhiya  
=== Code Execution Successful ===
```

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 "Dhiya", kemudian memanggil sapa.SapaUser(), program akan mencetak "Halo user Dhiya" ke konsol. Kode ini menunjukkan bagaimana sebuah class dapat digunakan secara fleksibel dengan berbagai input tanpa perlu mengubah strukturnya.

Soal 2 – Data Generic

Code

```
class DataGeneric {  
  constructor(data) {  
    this.data = data;  
  }  
  
  PrintData() {  
    console.log(`Data yang tersimpan adalah: ${this.data}`);  
  }  
}  
  
const dataGeneric = new DataGeneric("Dhiya Ulhaq Ramadhan  
dengan nim : 2211104053");  
  
dataGeneric.PrintData();  
  
console.log("=== Code Execution Successful ===");
```

Output

```
C:\Program Files\nodejs\node.exe .\05_Generics\Unguided\DataGeneric.js  
Data yang tersimpan adalah: Dhiya Ulhaq Ramadhan dengan nim : 2211104053  
=== Code Execution Successful ===
```

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 "Dhiya Ulhaq Ramadhan dengan nim : 2211104053" dan method PrintData() dipanggil, program akan mencetak "Data yang tersimpan adalah: Dhiya Ulhaq Ramadhan dengan nim : 2211104053" ke konsol. Kode ini menunjukkan bagaimana class dapat digunakan untuk menangani berbagai jenis data tanpa perlu mengubah strukturnya.