

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK**

**MODUL V
GENERICS**



**Disusun Oleh :
Arzario Irsyad Al Fatih
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

1. DASAR TEORI

1. Generics

Generics adalah fitur yang memungkinkan pembuatan kelas, metode, antarmuka, dan delegasi dengan parameter tipe. Dengan menggunakan generics, kode dapat dibuat lebih fleksibel, aman, dan dapat digunakan kembali tanpa perlu menentukan tipe data secara spesifik. Generics membantu menghindari konversi tipe eksplisit dan meningkatkan performa karena mengurangi unboxing dan boxing pada tipe data dasar.

Tujuan dari penggunaan generics adalah sebagai berikut:

- Meningkatkan fleksibilitas kode.
- Mengurangi duplikasi kode.
- Mendukung keamanan tipe pada waktu kompilasi.
- Mengoptimalkan performa aplikasi.

2. Generic Class

Generic class adalah kelas yang dirancang untuk bekerja dengan berbagai tipe data tanpa menentukan tipe secara spesifik pada saat perancangan. Dengan menggunakan parameter tipe pada deklarasi kelas, generic class dapat mengakomodasi tipe data yang berbeda pada saat instansiasi. Hal ini memungkinkan pengembangan kelas yang lebih dinamis dan mudah digunakan kembali.

3. Generic Function

Generic function adalah fungsi atau metode yang didefinisikan dengan parameter tipe sehingga dapat digunakan untuk berbagai tipe data tanpa mengubah kode fungsi itu sendiri. Generic function memungkinkan penulisan metode yang lebih fleksibel dan efisien. Generic function banyak digunakan pada operasi pemrosesan data atau metode utilitas yang menangani berbagai tipe.

4. Generic Delegate

Generic delegate adalah tipe data yang merepresentasikan metode dengan parameter dan nilai kembalian yang ditentukan secara generik. Generic delegate memungkinkan penggunaan metode dengan parameter tipe yang berbeda tanpa mendefinisikan banyak delegate secara eksplisit. Dengan adanya generic delegate, pengembangan aplikasi menjadi lebih modular dan fleksibel karena metode dapat didelegasikan dengan berbagai tipe data tanpa membuat definisi delegate yang berulang.

2. MAKSUD DAN TUJUAN

1. MAKSUD

Modul 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

Setelah mempelajari dan mempraktikkan modul ini, peserta diharapkan mampu:

- a. Memahami Konsep Generics – Mempelajari prinsip dasar generics dalam pemrograman serta manfaatnya dalam meningkatkan fleksibilitas kode.
- b. Menerapkan Generic Class – Mengimplementasikan struktur data yang dapat menangani berbagai jenis tipe data dalam satu class yang sama.
- c. Membuat Generic Function – Menulis fungsi generik yang dapat bekerja dengan berbagai tipe data tanpa perlu mendefinisikan fungsi baru untuk setiap tipe data.
- d. Menggunakan Generic Delegate – Memahami dan menerapkan konsep delegate dalam generics untuk meningkatkan modularitas dan fleksibilitas kode.
- e. Menganalisis Penerapan Generics dalam Aplikasi Nyata – Mengkaji bagaimana generics dapat digunakan dalam pengembangan sistem berbasis data seperti manajemen inventaris, pemrosesan transaksi, atau pembuatan framework.

- f. 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)

1. Generic Class

a. 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());
```

b. Output

```
\\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\K  
Class.js"  
[ 1, 'Hello' ]
```

c. Penjelasan

GenericList adalah kelas sederhana di JavaScript untuk menyimpan berbagai tipe data dalam array. Metode `add(item)` menambah item, sedangkan `getAll()` mengembalikan semua item dalam daftar. JavaScript tidak mendukung generics eksplisit, tetapi sifat dinamisnya memungkinkan penyimpanan tipe data berbeda dalam satu list.

2. Generic Function

a. Code

```
function swap(a, b) {  
  return [b, a];  
}
```

```
let x = 5,
    y = 10;
[x, y] = swap(x, y);
console.log(x, y);
```

b. Output

```
\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\W
Function.js"
10 5
```

c. Penjelasan

Kode di atas adalah contoh penggunaan fungsi swap dalam JavaScript menggunakan teknik destructuring assignment.

- Fungsi swap(a, b) menerima dua parameter dan mengembalikannya dalam array dengan posisi terbalik: [b, a].
- Variabel x dan y diinisialisasi dengan nilai 5 dan 10.
- swap(x, y) mengembalikan array [10, 5].
- Dengan destructuring assignment [x, y] = swap(x, y);, nilai x dan y ditukar sehingga x menjadi 10 dan y menjadi 5.
- console.log(x, y); mencetak output: 10 5

3. Generic Delegate

a. Code

```
function genericDelegate(callback, value) {
    callback(value);
}

genericDelegate(console.log, "Event Triggered");
```

b. Output

```
\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\W
Delegate.js"
Event Triggered
```

c. Penjelasan

Kode di atas menerima dua parameter, yaitu callback sebagai fungsi yang akan dipanggil dan value sebagai nilai yang diberikan kepada fungsi

tersebut. Saat dipanggil dengan `genericDelegate(console.log, "Event Triggered");`, fungsi `console.log` dijalankan dengan argumen "Event Triggered".

BAB III

PENUGASAN (UNGUIDED)

1. Soal 1 Unguided

a. Code

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

b. Output

```
\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\  
eneric.js"  
Halo user Tuan Rio
```

c. Penjelasan

Kode di atas adalah contoh generic class sederhana di JavaScript. Properti user diinisialisasi melalui constructor, lalu metode SapaUser() mencetak sapaan.

2. Soal 2 Unguided

a. Code

```
class DataGeneric {  
  constructor(data) {  
    this.data = data;  
  }  
  
  PrintData() {  
    console.log(`Data yang tersimpan adalah: ${this.data}`);  
  }  
}  
  
const data = new DataGeneric("2211104032");  
data.PrintData();
```


b. Output

```
\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\  
eneric.js"  
Data yang tersimpan adalah: 2211104032
```

c. Penjelasan

Kode di atas mendefinisikan kelas DataGeneric yang menggunakan konsep generic class sederhana dalam JavaScript. Kelas ini memiliki properti data yang diinisialisasi melalui constructor, serta metode PrintData() untuk menampilkan data yang tersimpan.