

**TUGAS PENDAHULUAN  
KONSTRUKSI PERANGKAT LUNAK**

**PERTEMUAN 3  
GENERIC**



**Disusun Oleh :  
Mohammad Dhimas Afrizal  
2211104023  
SE0601**

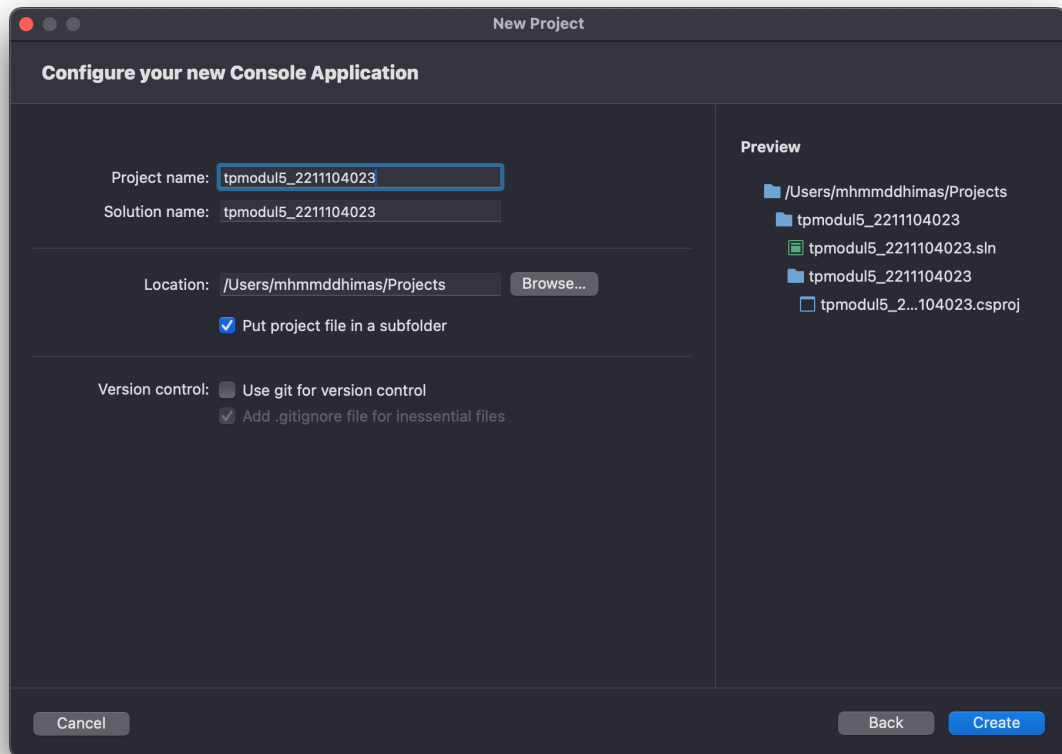
**Asisten Praktikum :  
Naufal El Kamil Aditya Pratama Rahman  
Imelda**

**Dosen Pengampu :  
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## 1. Buat Project Console / Tanpa GUI

- Buat Project baru, lalu beri nama tpmodul5\_2211104023



## 2. Buat branch baru pada Git project dan pindah ke branch tersebut

```
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(main) x git checkout -b generic-method
Switched to a new branch 'generic-method'
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(generic-method) x
```

## 3. Menambahkan method dengan generic

- Buatlah sebuah class bernama “HaloGeneric”.
- Pada class tersebut, tambahkan sebuah method dengan nama “SapaUser” yang memiliki
- generic parameter yang akan melakukan print “Halo user X” dimana X adalah input/nilai argument yang diberikan pada method tersebut.
- Panggil method tersebut pada fungsi/method utama dengan input String dengan isi nilai nama panggilan praktikan.

- Source Code

```
< > x Program.cs
Program > M Main()
1  using System;
2
3  class HaloGeneric
4  {
5      public void SapaUser<T>(T user)
6      {
7          Console.WriteLine($"Halo Bro {user}");
8      }
9  }
10
11 class Program
12 {
13     static void Main()
14     {
15         HaloGeneric halo = new HaloGeneric();
16         halo.SapaUser("Mohammad Dhimas Afrizal");
17     }
18 }
```

#### - Output

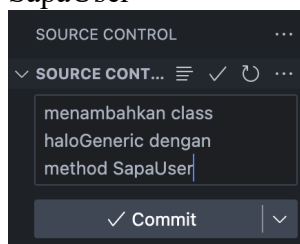
```
Terminal - tpmodul5_2211104023  
  
Halo Bro Mohammad Dhimas Afrizal
```

#### - Penjelasan

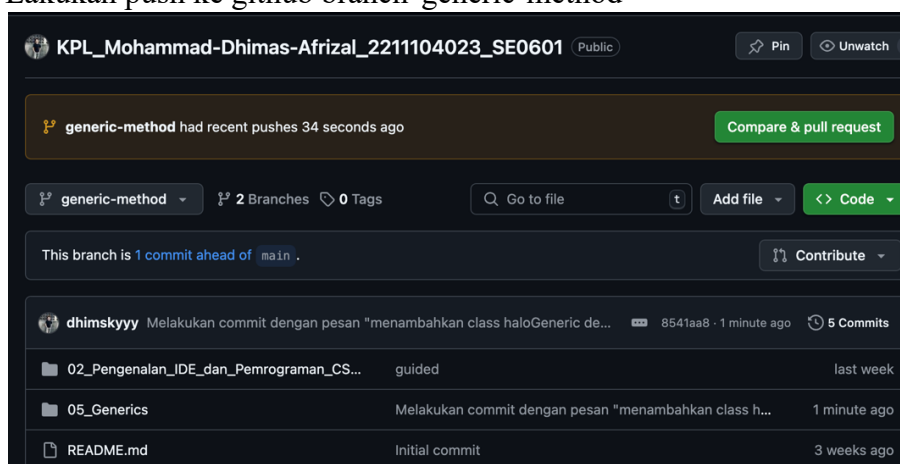
Kode ini merupakan implementasi Generic Method dalam bahasa C# untuk menyapa pengguna dengan fleksibilitas tipe data. Program ini terdiri dari dua class utama, yaitu HaloGeneric dan Program. Class HaloGeneric memiliki method SapaUser<T>(T user), di mana T adalah tipe data generic yang memungkinkan method ini menerima berbagai tipe data saat dipanggil. Method ini mencetak teks "Halo user X", di mana X adalah nilai dari parameter user. Dalam contoh ini, method dipanggil dengan string "Mohammad Dhimas Afrizal", sehingga output yang dihasilkan adalah "Halo bro Mohammad Dhimas Afrizal". Class Program berisi method utama (Main), yang berfungsi sebagai titik awal eksekusi program. Di dalamnya, dibuat objek HaloGeneric yang kemudian digunakan untuk memanggil SapaUser dengan input "Mohammad Dhimas Afrizal". Dengan penggunaan Generic, method ini dapat digunakan untuk berbagai tipe data tanpa perlu membuat method baru, sehingga meningkatkan efisiensi dan fleksibilitas kode.

### 4. Melakukan commit push dan pindah ke branch

- Melakukan commit dengan pesan "menambahkan class haloGeneric dengan method SapaUser"



- Lakukan push ke github branch "generic-method"



- Pada branch master/main, buat branch baru dengan nama "generic-class"

```
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(generic-method) ✖ git checkout -b generic-class  
Switched to a new branch 'generic-class'  
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(generic-class) |
```

### 5. Menambahkan method dengan generic

Tanpa membuat file baru (gunakan file yang dibuat saat membuat project dan pastikan branch aktif adalah pada branch generic-class):

a. Buatlah sebuah class bernama “DataGeneric” dengan mengikuti class model yang ditunjukkan pada gambar/tabel di bawah ini. Class tersebut memiliki property “Data” yang bertipe generic “T” dan memiliki konstruktor dengan parameter data.

DataGeneric
- data: T
+ DataGeneric(T)
+ PrintData(): void

b. Class tersebut juga memiliki method bernama PrintData yang melakukan print di console dengan output “Data yang tersimpan adalah: Y”, dengan “Y” adalah nilai dari property “data” dari kelas tersebut.

c. Panggil method PrintData() setelah mengisi “data” dengan NIM pada fungsi/method utama.

- Source Code

```

1  using System;
2
3  public class HaloGeneric
4  {
5      public static void SapaUser<T>(T user)
6      {
7          Console.WriteLine($"Halo Bro {user}");
8      }
9  }
10
11 public class DataGeneric<T>
12 {
13     private T data;
14     public DataGeneric(T data)
15     {
16         this.data = data;
17     }
18     public void PrintData()
19     {
20         Console.WriteLine($"Data yang tersimpan adalah: {data}");
21     }
22 }
23
24 class Program
25 {
26     static void Main()
27     {
28         HaloGeneric.SapaUser("Mohammad Dhimas Afrizal");
29         DataGeneric<string> data = new DataGeneric<string>("2211104023");
30         data.PrintData();
31     }
32 }

```

- Output

```

Terminal - tpmodul5_2211104023

Halo Bro Mohammad Dhimas Afrizal
Data yang tersimpan adalah: 2211104023

```

- Penjelasan

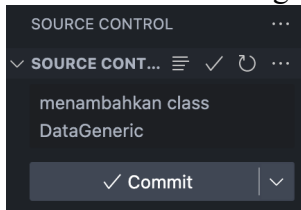
Kode ini merupakan implementasi Generic Method dan Generic Class dalam bahasa C#, yang memungkinkan penggunaan berbagai tipe data secara fleksibel. Class HaloGeneric memiliki method SapaUser<T>(T user), yang dapat menerima parameter dengan tipe data apa pun dan mencetak pesan "Halo Bro X", di mana X adalah nilai dari parameter user. Dalam program ini, method tersebut dipanggil dengan string "Mohammad Dhimas Afrizal", sehingga output yang dihasilkan adalah "Halo Bro Mohammad Dhimas Afrizal".

Selain itu, terdapat class DataGeneric<T> yang menggunakan konsep Generic Class untuk menyimpan data dengan tipe yang fleksibel. Class ini memiliki konstruktor yang menerima parameter data bertipe generic dan method PrintData(), yang mencetak nilai dari data yang tersimpan. Pada method Main(), objek DataGeneric<string> dibuat dengan menyimpan data "2211104023", dan ketika PrintData() dipanggil, program akan mencetak "Data yang tersimpan adalah: 2211104023". Dengan penggunaan generics, kode menjadi lebih fleksibel dan dapat digunakan untuk berbagai tipe data tanpa harus membuat method atau class baru, sehingga meningkatkan efisiensi dan keterbacaan program.

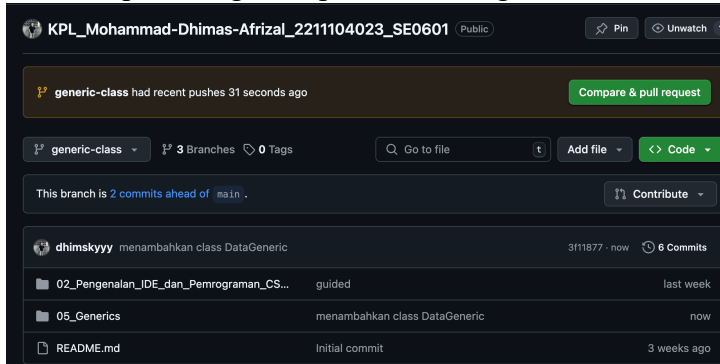
## 6. Melakukan commit, push, dan pindah ke branch bagian kedua

Pada branch yang sedang aktif saat ini ( branch "generic-class" )

- a. Lakukan commit dengan pesan "menambahkan class DataGeneric"



- b. Lakukan push ke github pada branch "generic-class"



## 7. Melakukan git merge dari

Pastikan branch aktif adalah branch master/main:

- a. Lakukan git merge branch "generic-method" ke branch master/main.

```
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(generic-class) git merge generic-method
Already up to date.
```

- b. Lakukan git merge branch "generic-class" ke branch master/main, dan jika terjadi merge conflict, pastikan semua baris yang conflict sudah diperbaiki.

```
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(generic-class) git merge generic-class
Already up to date.
```

- c. Lakukan git push untuk branch master/main ke github repository.

```
> KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601 git:(main) git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/dhimsky/KPL_Mohammad-Dhimas-Afrizal_2211104023_SE0601.git
e1f6db7..3f11877 main -> main
```