

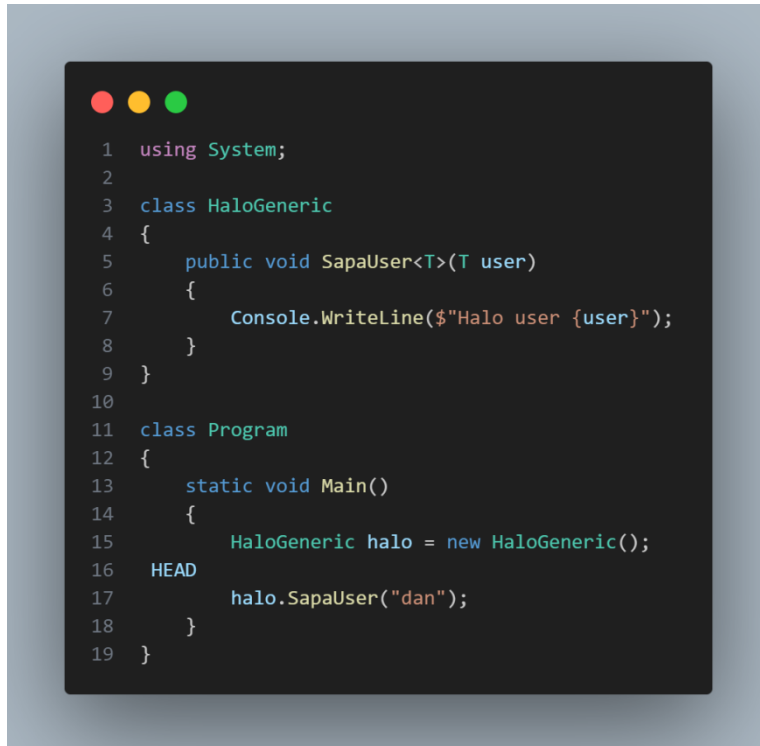
Nama : Pradana Argo Pangestu  
Nim : 2311104079  
Kelas: SE0702

## BRANCH BARU PADA GIT PROJECT

Lakukan  
Git branch generic-method

Setelah uty git checkout generic-method

## MENAMBAHKAN METHOD GENERIC DAN COMMIT

A screenshot of a code editor with a dark background and light-colored text. The code is in C# and defines a generic class `HaloGeneric` with a method `SapaUser<T>(T user)` that prints "Halo user {user}" to the console. A `Program` class with a `Main()` method is also shown, which creates an instance of `HaloGeneric` and calls `SapaUser("dan")`. The code is numbered from 1 to 19.

```
1  using System;
2
3  class HaloGeneric
4  {
5      public void SapaUser<T>(T user)
6      {
7          Console.WriteLine($"Halo user {user}");
8      }
9  }
10
11 class Program
12 {
13     static void Main()
14     {
15         HaloGeneric halo = new HaloGeneric();
16         HEAD
17         halo.SapaUser("dan");
18     }
19 }
```

konsep **generic method**. Program ini memiliki class `HaloGeneric` dengan method `SapaUser<T>(T user)`, di mana `T` adalah **parameter generic** yang memungkinkan method menerima berbagai tipe data sebagai input. Method ini mencetak pesan "**Halo user X**", dengan `X` sebagai nilai yang diberikan saat pemanggilan. Pada `Main()`, objek `halo` dari class `HaloGeneric` dibuat, lalu method `SapaUser` dipanggil dengan argumen "dan", sehingga akan menampilkan output "**Halou ser Nama Panggilan**" di console.

## Pindah Branch

Lakukan  
Git branch generic-class  
Git checkout generic-class

## MENAMBAHKAN METHOD DENGAN GENERIC DAN COMMIT

```
1  using System;
2
3  class HaloGeneric
4  {
5      public void SapaUser<T>(T user)
6      {
7          Console.WriteLine($"Halo user {user}");
8      }
9  }
10
11 class Program
12 {
13     static void Main()
14     {
15         HaloGeneric halo = new HaloGeneric();
16     HEAD
17         halo.SapaUser("dan");
18     }
19 }
20
21 class DataGeneric<T>
22 {
23     private T data;
24
25     public DataGeneric(T data)
26     {
27         this.data = data;
28     }
29
30     public void PrintData()
31     {
32         Console.WriteLine($"Data yang tersimpan adalah: {data}");
33     }
34 }
35
36 class Program
37 {
38     static void Main()
39     {
40         DataGeneric<string> data = new DataGeneric<string>("2311104079");
41         data.PrintData();
42     }
43 }
```

**generic class**, memungkinkan penggunaan berbagai tipe data tanpa mendefinisikan ulang class. Class `DataGeneric<T>` memiliki property `data` bertipe generic `T`, yang diinisialisasi melalui konstruktor saat objek dibuat. Method `PrintData()` digunakan untuk mencetak nilai `data` ke console dengan format **"Data yang tersimpan adalah: X"**, di mana `X` adalah nilai dari `data`. Pada method `Main()`, objek `data` dibuat dari class `DataGeneric<string>` dengan nilai `"2311104079"`, lalu method `PrintData()` dipanggil untuk menampilkan data yang tersimpan di console.

MELAKUKANGITMERGEDARIKEDUABRANCHBARU

Lakukan

Git merge generic-method

Git merge generic-class

Git checkout main

