

Nama : Prajna paramitha Wardhany

NIM : 2311104016

Kelas : S1 SE 07 01

https://github.com/270505Nana/KPL_Prajna-paramitha_2311104016_SE0701/tree/Generic-method/05_Generics_dbc

TP MODUL 5

1. MEMBUAT PROJECT CONSOLE/TANPA GUI

Buka IDE misalnya dengan Visual Studio

A. Pilih “Create a new project” kemudian pilih “Console App”. Pada IDE lain pada umumnya hanya perlu membuat project baru saja.

B. Masukkan project baru dengan nama tpmodul5_NIM.

2. MELAKUKAN GIT COMMIT PADA PROJECT YANG DIBUAT

Task atau langkah-langkah yang perlu dikerjakan adalah sebagai berikut:

A. Buatlah github public repository kosong (pastikan bagian “Initialize this repository with” tidak ada yang dicentang pada saat membuat repository baru) melalui <https://github.com/>

B. Melakukan inisialisasi dengan “git init” di folder project yang dibuat.

C. Membuat commit untuk versi pertama dari project yang dibuat dengan pesan commit bebas.

D. Melakukan git push ke github repo.

3. MEMBUAT BRANCH BARU PADA GIT PROJECT

Pada folder project yang sedang aktif, langkah-langkah yang perlu dikerjakan:

A. Membuat git branch baru dengan nama “generic-method”.

B. Berpindah ke branch yang sudah dibuat sebelumnya yaitu “generic-method”.



4. MENAMBAHKAN METHOD DENGAN GENERIC

Tanpa membuat file baru (gunakan file yang dibuat saat membuat project):

A. Buatlah sebuah class bernama “HaloGeneric”.

B. 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.

C. Panggil method tersebut pada fungsi/method utama dengan input String dengan isi nilai nama panggilan praktikan.

SELAMAT MENGERJAKAN!

5. MELAKUKAN COMMIT, PUSH, DAN PINDAH KE BRANCH

Pada branch yang sedang aktif saat ini (branch “generic-method”):

- Lakukan commit dengan pesan “menambahkan class HaloGeneric dengan method SapaUser”.
- Lakukan push ke github pada branch “generic-method” di remote/github repo.
- Setelah proses push berhasil, ganti branch yang aktif ke master/main branch.
- Pada branch master/main, buat branch baru dengan nama “generic-class”.
- Setelah itu, lakukan pindah branch ke branch “generic-class” yang sudah dibuat.

```
class HaloGeneric {
  // Method dengan parameter generic
  SapaUser(user) {
    console.log(`Halo user ${user}`);
  }
}

// Fungsi utama
function main() {
  // Membuat objek dari kelas HaloGeneric
  const halo = new HaloGeneric();

  // Memanggil method SapaUser dengan input String nama nana
  halo.SapaUser("Nana");
}

main();
```

Penjelasan :

- Pertama membuat class HaloGeneric lalu membuat method sapauser yang memiliki user sebagai parameter.
- Console.log untuk menampilkan output dari ‘Halo user, nana’
- Membuat function main dimana memanggil method sapa user dan memasukkan nama praktikan

Output :

```
paramitha_231110401
Halo user Nana
```

6. MENAMBAHKAN METHOD DENGAN GENERIC

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

- 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

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

SELAMAT MENERJAKAN!

7. MELAKUKAN COMMIT, PUSH, DAN PINDAH BRANCH BAGIAN 2

Pada branch yang sedang aktif saat ini (branch “generic-class”):

- Lakukan commit dengan pesan “menambahkan class DataGeneric”.
- Lakukan push ke github pada branch “generic-class” di remote/github repo.
- Setelah proses push berhasil, ganti branch yang aktif ke master/main branch.

8. MELAKUKAN GIT MERGE DARI KEDUA BRANCH BARU

Pastikan branch aktif adalah branch master/main :

- Lakukan git merge branch “generic-method” ke branch master/main.
- Lakukan git merge branch “generic-class” ke branch master/main, dan jika terjadi merge conflict, pastikan semua baris yang conflict sudah diperbaiki.
- Lakukan git push untuk branch master/main ke github repository.

```
class DataGeneric {
  constructor(data) {
    this.data = data;
  }

  // B.Class tersebut juga memiliki method bernama PrintData yang melakukan print di console
  PrintData() {
    console.log(`Data yang tersimpan adalah: ${this.data}`);
  }
}

// Fungsi utama
function main() {
  // Membuat objek dari kelas DataGeneric dengan NIM sebagai data
  const dataObj = new DataGeneric("2311104016");

  // Mengambil method PrintData
  dataObj.PrintData();
}

main();
```

Penjelasan :

- Membuat class DataGeneric sesuai perintah dimana didalamnya terdapat method printData sesuai dengan point B untuk melakukan print out menggunakan console.log
- Pada fungsi main const dataObj membuat objek dari class datagenericnya nim sebagai datanya

Output

```
paramitha_2311104016_SE0701\05_Gener
Data yang tersimpan adalah: 23111040
```

9. PENGUMPULAN FILE/TUGAS PENDAHULUAN

Kumpulkan semua file berikut dalam bentuk file zip/rar/7zip:

- Source code dari project yang dibuat
- File docx/pdf yang berisi:
 - Link github repository
 - Screenshot hasil run (hasil console output dari table driven dan state-based)
 - Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

SELAMAT MENGERJAKAN!