

12 Maret 2022

TUGAS PENDAHULUAN MODUL 4

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 tpmodul4_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.

5. MELAKUKAN COMMIT, PUSH DAN PINDAH BRANCH

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

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

- D. Pada branch master/main, buat branch baru dengan nama “generic-class”.
- E. Setelah itu, lakukan pindah branch ke branch “generic-class” yang sudah dibuat.

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):

- 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.

7. MELAKUKAN COMMIT, PUSH DAN PINDAH BRANCH BAGIAN 2

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” di remote/github repo.
- C. 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 :

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

9. PENGUMPULAN FILE/TUGAS PENDAHULUAN

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

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