Tugas 1: Soal Praktikum Git

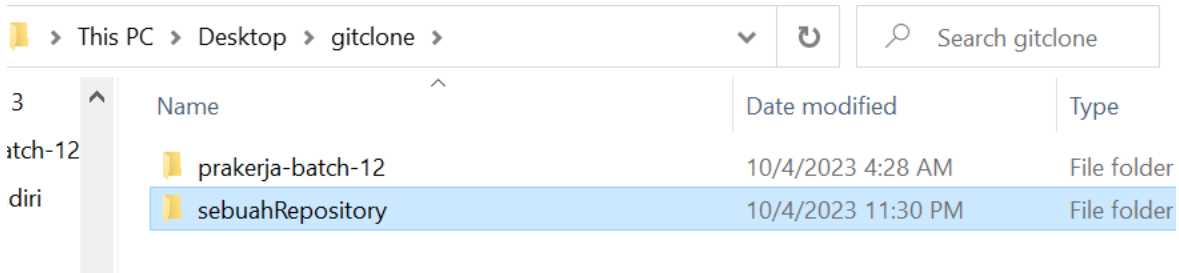## 1. Buat sebuah repository di Github

Seusai selesai membuat sebuah Repository pada github, saya melakukan Pull dengan cara Git Clone **(Sekalian menjawab pertanyaan No. 3 tentang Git Pull pada 3B):**

Oke, lanjut ke Nomor 2

2. Implementasikan penggunaan branching yang terdiri dari master, development, featureA, dan featureB

NOTE: Sampai disini masih belum di Push ke Github, alias masih Lokal

3. Implementasikan intruksi git untuk push, pull, stash dan merge

3a. Git Push

Git push branch master:

Git push branch development:

Git push branch featureA:

Git push branch featureB:

3b. Git Pull

Sudah dijawab pada Soal No. 1 ketika melakukan Git Clone seusai membuat Repository baru.

3c. Git Stash

Skenario 1: terdapat sebuah file namanya test.txt dengan isi "just a testing"

```
PS C:\Users\lenovo\Desktop\writers>
```

Skenario 2: user melakukan git init & commit ke dalam file (contoh folder adalah writers git)

```
PS C:\Users\lenovo\Desktop\writers> git init
Reinitialized existing Git repository in C:/Users/lenovo/Desktop/writers/.git/
PS C:\Users\lenovo\Desktop\writers> git commit -m "Insert file test.txt to git main"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\lenovo\Desktop\writers>
```

Skenario 3: user melakukan update pada file test.txt dengan tambahan kata "test testtt", lalu melakukan commit ulang

```
≡ Test.txt.txt M ✕

  ≡ Test.txt.txt
    1  just a testing test testtt
```

```
nothing to commit, working tree clean
PS C:\Users\lenovo\Desktop\writers> git commit -m "Insert file test.txt to git main"
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Test.txt.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\lenovo\Desktop\writers>
Postgres Server                                                    Ln 1, Col 27
```

Skenario 4: User melihat conflict oleh sebab tidak dapat dicommit, lalu ingin mengembalikan codingan sebelumnya, maka ia melakukan git stack untuk mengembalikan codingan sebelumnya

```
≡ Test.txt.txt ●

  ≡ Test.txt.txt
    1   just a testing
```

```
PS C:\Users\lenovo\Desktop\writers> git stash
Saved working directory and index state WIP on main: 7fce17c descriptive comment for code changes
PS C:\Users\lenovo\Desktop\writers>
```

Skenario 5: Pada akhirnya user dapat melakukan commit dengan lancar dan berhasil

```
PS C:\Users\lenovo\Desktop\writers> git init
Reinitialized existing Git repository in C:/Users/lenovo/Desktop/writers/.git/
PS C:\Users\lenovo\Desktop\writers> git commit -m "commit new file and ordinary information"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```
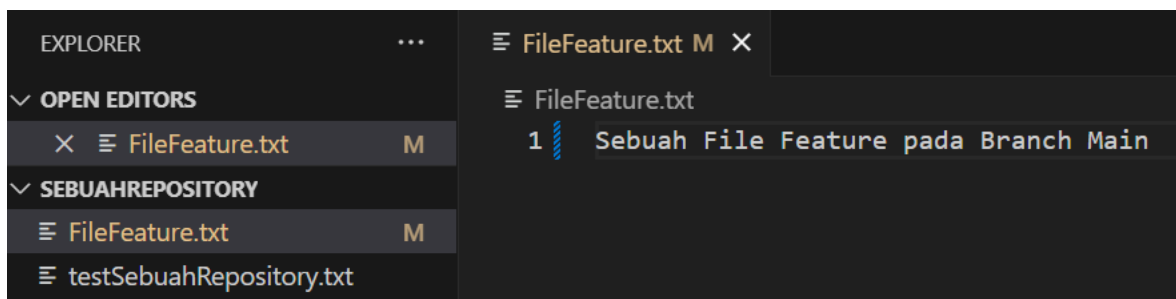
3d. Git Merge

Contoh, memiliki sebuah file namanya FileFeatureA.txt pada Branch FeatureA, lalu ingin di merge file tersebut ke dalam Branch Master (dalam contoh saya adalah Branch Main):

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    COMMENTS

PS C:\Users\lenovo\Desktop\writers> git branch
* featureA
  main
PS C:\Users\lenovo\Desktop\writers> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\lenovo\Desktop\writers> git merge featureA
Updating 7fce17c..fa23ecf
Fast-forward
 FileFeatureA.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 FileFeatureA.txt
PS C:\Users\lenovo\Desktop\writers>
```

4. Implementasikan sebuah penanganan conflict di branch developement ketika setelah merge dari branch featureA lalu merge dari branch featureB (Conflict bisa terjadi jika kedua branch mengerjakan di file dan line code yang sama)

Contoh kita memiliki sebuah File namanya FileFeature.txt pada Branch Main sebagai berikut:

```
EXPLORER                    ...      ≡ FileFeature.txt M ×

∨ OPEN EDITORS                          ≡ FileFeature.txt
    ×  ≡ FileFeature.txt      M          1│    Sebuah File Feature pada Branch Main
∨ SEBUAHREPOSITORY
    ≡ FileFeature.txt         M
    ≡ testSebuahRepository.txt
```

FileFeature.txt kita add ke dalam Branch Main sebelum membuat 2 prototype:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS                        >_ powers

PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git init
Reinitialized existing Git repository in C:/Users/lenovo/Desktop/gitclone/sebuahRepository/.git/
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git add .
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git commit -m "Add FileFeature.txt to Branch Main"
[main c1c278a] Add FileFeature.txt to Branch Main
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository>
```

Kita ingin melakukan 2 prototyping yakni melalui branch FeatureA dan FeatureB:

Pertama: menuju ke Branch FeatureA

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS

PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git branch
  development
  featureA
  featureB
* main
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git checkout featureA
Switched to branch 'featureA'
Your branch is ahead of 'origin/featureA' by 1 commit.
  (use "git push" to publish your local commits)
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository>
```

Melakukan perubahan pada FileFeature.txt di branch FeatureA

```
≡ FileFeature.txt ×

≡ FileFeature.txt
  1    Sebuah perubahan file feature pada Branch FeatureA
```

Kedua: menuju ke Branch FeatureB
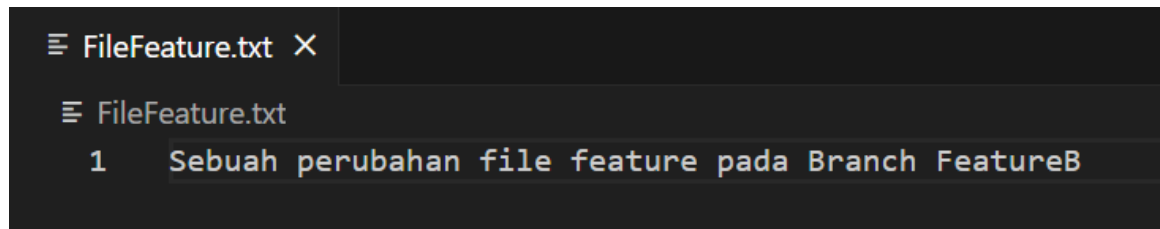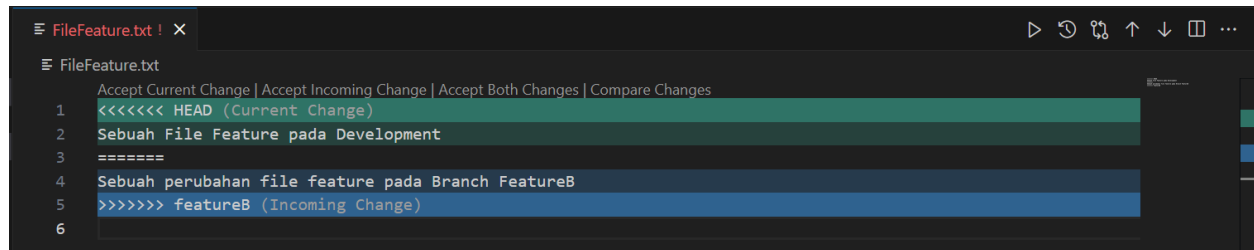
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS

PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git branch
  development
* featureA
  featureB
  main
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository> git checkout featureB
Switched to branch 'featureB'
Your branch is ahead of 'origin/featureB' by 1 commit.
  (use "git push" to publish your local commits)
PS C:\Users\lenovo\Desktop\gitclone\sebuahRepository>
```

Lalu semuanya dilakukan merge menuju branch Development (ambil contoh branch featureB ketika di merge menuju ke branch A):

Terdapat 4 pilihan solusi yakni accept current change, accept incoming change, accept both changes, dan compare changes, dimana kita bisa memilih salah satu sesuai dengan kenyamanan, sebelum lanjut melakukan commit.

5. Gunakan Merge no fast forward

6. Screenshot hasil network di github