

PRAKTIKUM PEMROGRAMAN BERBASIS KERANGKA KERJA

GIT



Achmad Wahyu Surya Wahono
1203210026

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
INSTITUT TEKNOLOGI TELKOM SURABAYA
2023**

DAFTAR ISI

DAFTAR ISI	2
1. Perkenalan Git	4
1.1. Mengenal Git	4
1.2. Apa yang dilakukan oleh Git	5
1.3. Mengapa Git penting bagi programmer	5
2. Instalasi dan Konfigurasi Git.....	6
3. Membuat Repositori	6
3.1. git init.....	6
3.2. .gitignore.....	7
4. Git Commit (Simpan Perubahan Revisi).....	7
4.1. Menambahkan Revisi	7
4.2. Tiga Kelompok Kondisi File dalam Git	8
4.3. Membuat Revisi Pertama	8
4.4. Membuat Revisi Kedua.....	10
5. Git Log (Catatan Revisi)	11
6. Git Diff.....	12
7. Membatalkan Revisi (Git Checkout, Git Reset, Git Revert)	14
7.1. git checkout.....	14
7.2. git reset (Membatalkan Perubahan File yang Sudah dalam Kondisi staged)	15
7.3. Membatalkan Perubahan File yang Sudah dalam Kondisi Committed	16
8. Menggunakan Percabangan (branch) untuk mencegah konflik.....	18
8.1. Cara Membuat Cabang (Branch) Baru.....	18
8.3. Menggabungkan Cabang.....	20
8.4. Mengatasi Bentrok.....	20
8.5. Menghapus Cabang.....	21
9. Bekerja dengan Remote Repository.....	21
9.1. Membuat Repository Github	22
9.2. Menambahkan Remote	23
9.3. Mengambil Revisi dari Remote Repository	26
9.4. Mengambil Revisi dengan git fetch	27
9.5. Mengambil Revisi dengan git pull.....	28

9.6.	Clone Repository	29
------	------------------------	----

GIT

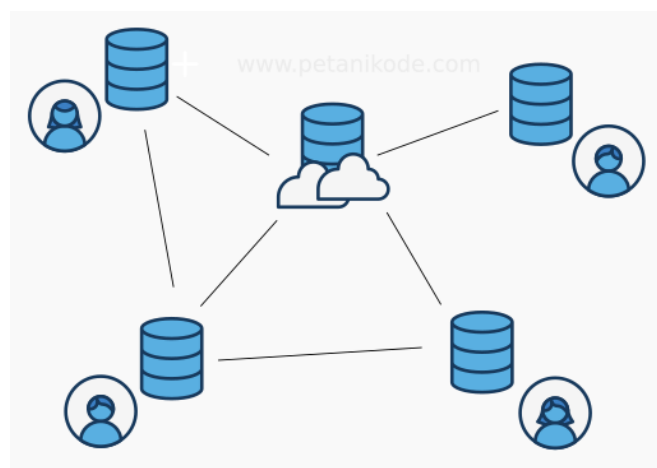
Git adalah salah satu tool yang sering digunakan dalam proyek pengembangan perangkat lunak. Git bahkan menjadi tool yang wajib diapahami oleh programmer, karena banyak digunakan dimana-mana.

1. Perkenalan Git

1.1. Menenal Git

Git adalah salah satu **sistem pengontrol versi (Version Control System)** pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds. Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh **banyak orang maupun sendiri**. Git dikenal juga dengan ***distributed revision control (VCS terdistribusi)***, artinya penyimpanan database Git tidak hanya berada dalam satu tempat saja.

Semua orang yang terlibat dalam pengkodean proyek akan menyimpan database Git, sehingga akan memudahkan dalam mengelola proyek baik online maupun offline. Dalam Git terdapat **merge** untuk menyebut aktifitas penggabungan kode.



VCS Terdistribusi

Sedangkan pada **VCS (Version Control System) yang terpusat**, database disimpan dalam satu tempat dan setiap perubahan disimpan ke sana. VCS terpusat memiliki beberapa kekurangan: Semua tim **harus terkoneksi ke jaringan** untuk mengakses source-code; **Tersimpan di satu tempat**, nanti kalau server bermasalah bagaimana? Karena itu, Git hadir untuk menutupi kekurangan yang dimiliki oleh VCS terpusat.

4. Lebih aman digunakan untuk kolaborasi, karena bisa tahu **apa yang diubah dan siapa yang mengubahnya**.
5. Bisa memahami cara **deploy aplikasi modern**; dan sebagainya...

2. Instalasi dan Konfigurasi Git

- Untuk mengecek apakah di laptop kita sudah terinstall git atau belum bisa melakukan command **git --version** kemudian untuk cek konfigurasi **git config --list**

```
C:\Users\ASUS>git --version
git version 2.41.0.windows.1

C:\Users\ASUS>
```

- Apabila belum, bisa lakukan install dan beberapa konfigurasi dengan mengikuti tutorial ini <https://www.depotcode.com/install-git/>

3. Membuat Repositori

3.1. git init

- Repositori (repository) dalam bahasa indonesia artinya gudang. Repositori merupakan istilah yang digunakan untuk direktori proyek yang menggunakan Git.
- Jika kita memiliki sebuah direktori dengan nama project-01 dan di dalamnya sudah menggunakan git, maka kita sudah punya repositori bernama project-01.
- Pembuatan repositori dapat dilakukan dengan perintah **git init [nama-dir]**

```
C:\Windows\System32\cmd.exe

D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum>git init praktikum-git
Initialized empty Git repository in D:/Maba ITTS/Matkul/SEM 5/Pemrograman Berbasis Kerangka Kerja/Praktikum/praktikum-git/.git/

D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum>
```

- Perintah diatas akan membuat direktori bernama project-01. Kalau direktorinya sudah ada, maka Git akan melakukan inisialisasi di dalam direktori tersebut.
- Contoh lain Perintah berikut ini akan membuat repositori pada direktori saat ini (working directory).

```
C:\Windows\System32\cmd.exe

D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum>mkdir praktikum-12
D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum>cd praktikum-12
D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>git init .
Initialized empty Git repository in D:/Maba ITTS/Matkul/SEM 5/Pemrograman Berbasis Kerangka Kerja/Praktikum/praktikum-12/.git/

D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

- Tanda titik (.) artinya kita akan membuat repository pada direktori tempat kita berada saat ini.

3.2. .gitignore

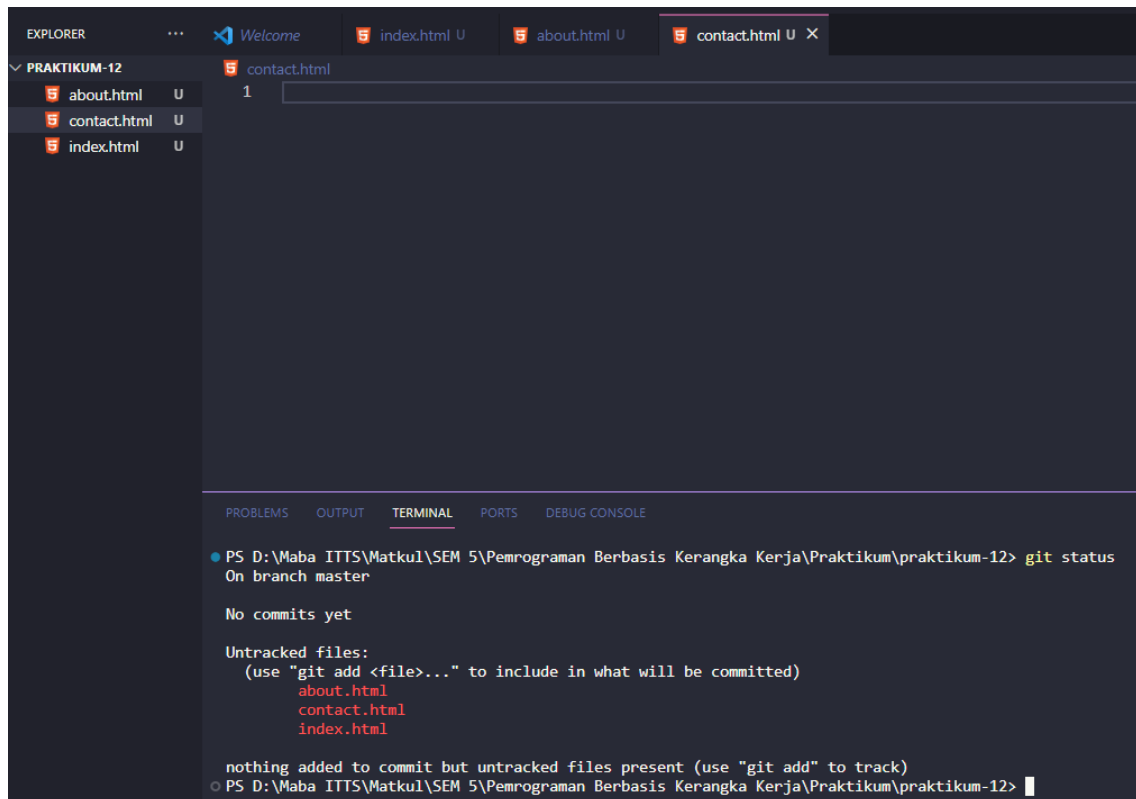
- .gitignore merupakan sebuah file yang berisi daftar nama-nama file dan direktori yang akan diabaikan oleh Git.
- Perubahan apapun yang kita lakukan terhadap file dan direktori yang sudah masuk ke dalam daftar .gitignore tidak akan dicatat oleh Git.
- Cara menggunakan .gitignore, buat saja sebuah file bernama .gitignore dalam root direktori proyek/repositori.
- Sebagai contoh ketika kita membuat project expo terdapat file .gitignore seperti dibawah. Maka file dan direktori tersebut akan diabaikan oleh Git.
- Pembuatan file .gitignore sebaiknya dilakukan di awal pembuatan repositori.

```
PModul4 > .gitignore
1  # Learn more https://docs.github.com/en/
2
3  # dependencies
4  node_modules/
5
6  # Expo
7  .expo/
8  dist/
9  web-build/
10
```

4. Git Commit (Simpan Perubahan Revisi)

4.1. Menambahkan Revisi

- Setelah membuat repositori kosong, silahkan coba buat 3 file html kosong pada repositori "project-01" yang telah dibuat sebelumnya.
- Buat file kosong "index.html", "about.html", dan "contact.html"
- Setelah ditambahkan, coba ketik perintah **git status** untuk melihat status repositorinya



```
EXPLORER  Welcome  index.html U  about.html U  contact.html U X
PRAKTIKUM-12
  about.html U
  contact.html U
  index.html U

contact.html
1

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    about.html
    contact.html
    index.html

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Berdasarkan keterangan gambar di atas, saat ini kita berada **cabang (branch) master** dan ada **tiga file yang belum ditambahkan ke Git**.

4.2. Tiga Kelompok Kondisi File dalam Git

1. Modified

Modified adalah kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di version control. Contohnya pada gambar di atas, ada tiga file HTML yang dalam kondisi modified.

2. Staged

Staged adalah kondisi dimana revisi sudah ditandai, tetapi belum disimpan di version control. Untuk mengubah kondisi file dari modified ke staged gunakan perintah **git add nama_file**. Contoh: **git add index.html**

3. Committed

Committed adalah kondisi dimana revisi sudah disimpan di version control. perintah untuk mengubah kondisi file dari staged ke committed adalah **git commit**.

4.3. Membuat Revisi Pertama

Selanjutnya, silahkan ubah kondisi tiga file HTML tadi menjadi staged dengan perintah **git add** ada beberapa cara :

git add index.html

git add about.html

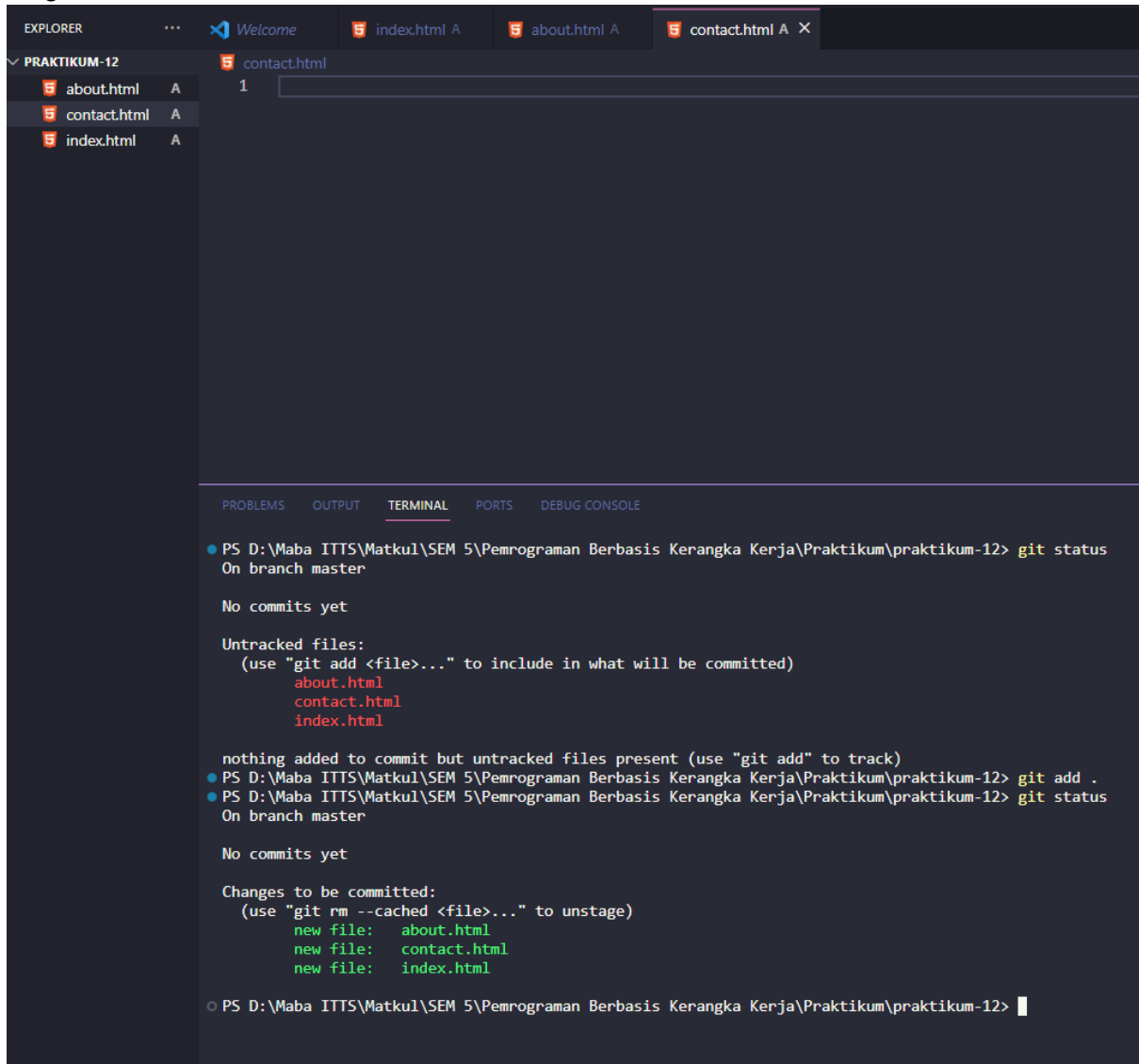
git add contact.html

git add index.html about.html contact.html

git add *.html

Atau seperti ini (semua file dan direktori): **git add .**

Setelah itu, cobalah ketik perintah **git status** lagi. Kondisi filenya sekarang akan menjadi staged.

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a folder named 'PRAKTIKUM-12' containing three files: 'about.html', 'contact.html', and 'index.html', each with a status icon 'A'. The main editor area shows the 'contact.html' file. The bottom panel contains a terminal window with the following output:

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        about.html
        contact.html
        index.html

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add .
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master

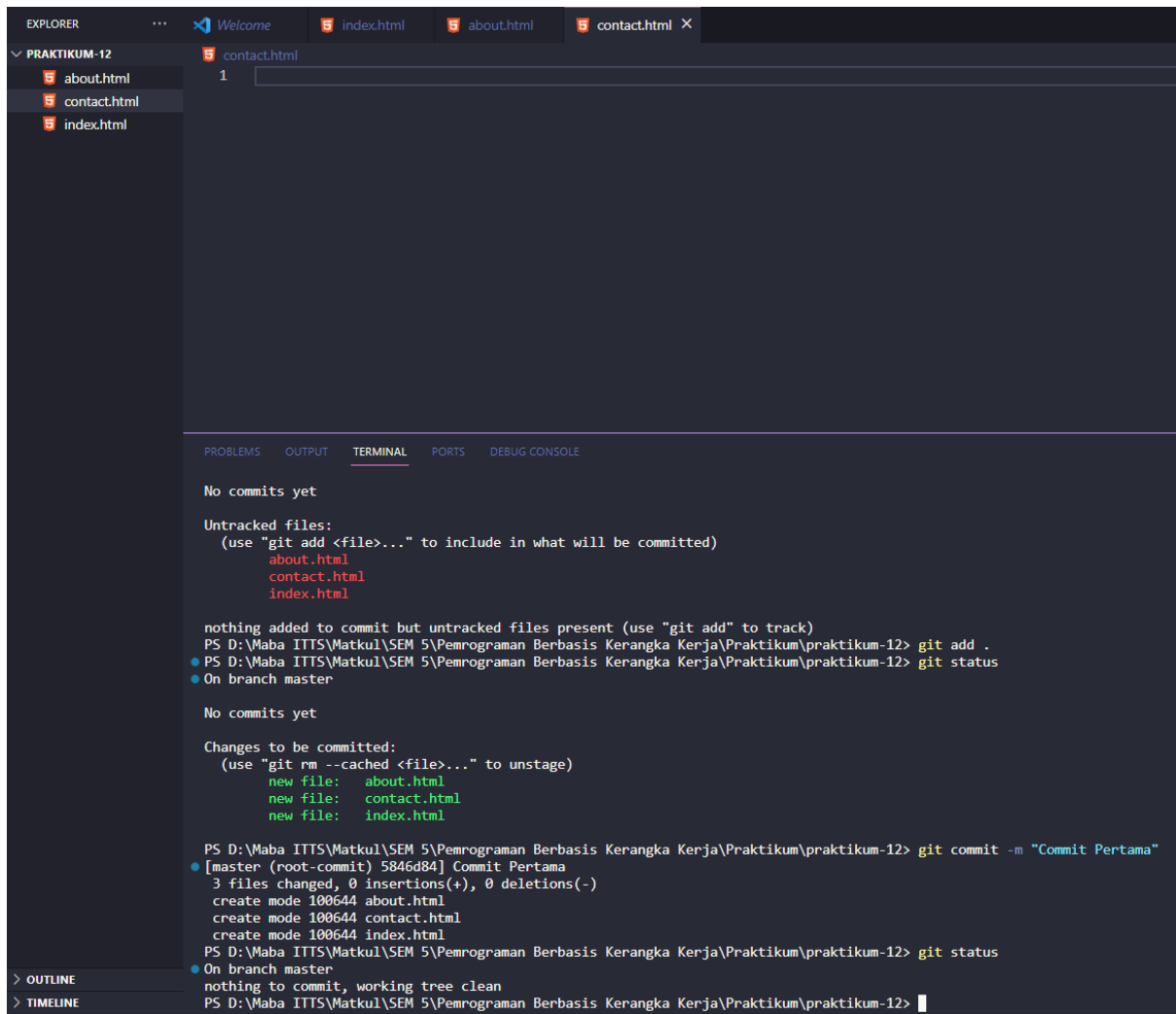
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   about.html
        new file:   contact.html
        new file:   index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Setelah itu, ubah kondisi file tersebut ke committed agar semua perubahan disimpan oleh Git.

git commit -m "Commit pertama"

Setelah itu, coba cek dengan perintah git status lagi.



```
EXPLORER
... Welcome index.html about.html contact.html X
PRAKTIKUM-12
  about.html
  contact.html
  index.html

contact.html
1

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

No commits yet

Untracked files:
(use "git add <file>..." to include in what will be committed)
  about.html
  contact.html
  index.html

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add .
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:   about.html
  new file:   contact.html
  new file:   index.html

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "Commit Pertama"
[master (root-commit) 5846d84] Commit Pertama
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 about.html
 create mode 100644 contact.html
 create mode 100644 index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
nothing to commit, working tree clean
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

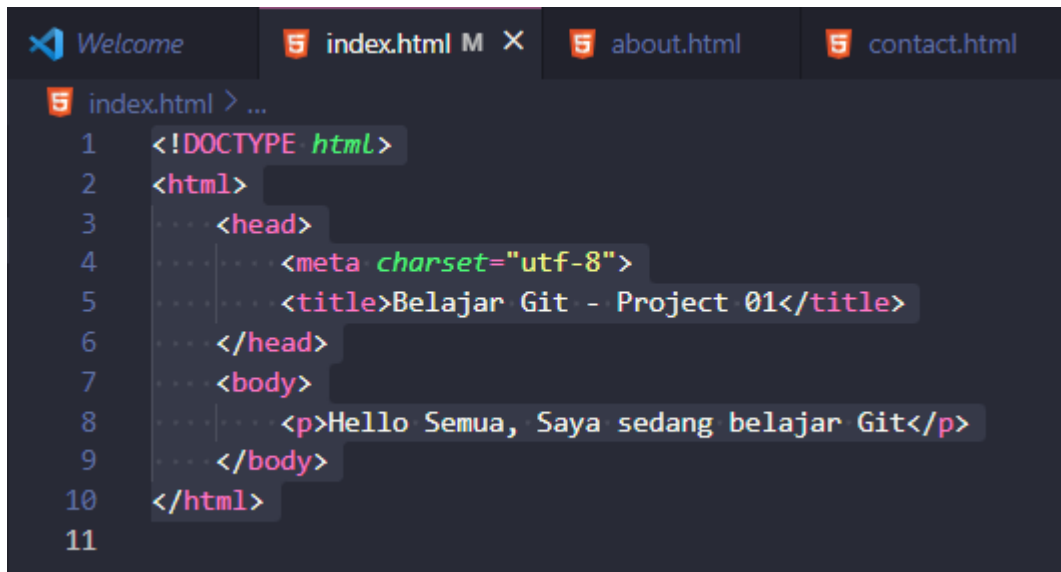
4.4. Membuat Revisi Kedua

Ceritanya ada perubahan yang akan kita lakukan pada file **index.html**. Silahkan modifikasi isi file index.html Menjadi seperti ini.

index.html	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Belajar Git - Project 01</title> </head> <body> <p>Hello Semua, Saya sedang belajar Git</p> </body> </html></pre>
------------	--

Setelah itu ketik lagi perintah git status.

Terlihat di sana, file index.html sudah dimodifikasi. Kondisinya skarang berada dalam modified.



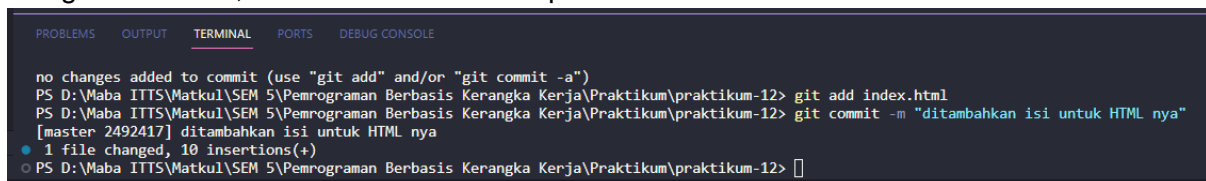
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Belajar Git - Project 01</title>
6   </head>
7   <body>
8     <p>Hello Semua, Saya sedang belajar Git</p>
9   </body>
10 </html>
11
```

Lakukan commit lagi seperti revisi pertama.

git add index.html

git commit -m "ditambahkan isi"

Dengan demikian, revisi kedua sudah disipan oleh Git.



```
no changes added to commit (use "git add" and/or "git commit -a")
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "ditambahkan isi untuk HTML nya"
[master 2492417] ditambahkan isi untuk HTML nya
1 file changed, 10 insertions(+)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Dalam commit ada argumen -m, argumen tersebut untuk menambahkan pesan setiap menyimpan revisi.

Sekarang Git sudah mencatat dua revisi yang sudah kita lakukan. Kita bisa ibaratkan revisi-revisi ini sebagai checkpoint pada Game. Apabila nanti ada kesalahan, kita bisa kembali ke checkpoint ini.



5. Git Log (Catatan Revisi)

- Git menyediakan perintah git log untuk melihat catatan log perubahan pada repositori. Contoh penggunaannya:
git log
git log --oneline
- Maka kita akan melihat log perubahan apa saja yang sudah dilakukan dalam repositori.

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log
commit 249241797562c13a118bb43c711b99b7d67b2854 (HEAD -> master)
Author: WithStarup <withstarup@gmail.com>
Date: Fri Dec 15 15:42:42 2023 +0700

    ditambahkan isi untuk HTML nya

commit 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b
Author: WithStarup <withstarup@gmail.com>
Date: Fri Dec 15 15:36:51 2023 +0700

    Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log --oneline
2492417 (HEAD -> master) ditambahkan isi untuk HTML nya
5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

Terdapat juga beberapa command git log seperti dibawah ini :

Untuk melihat log pada revisi tertentu, kita bisa memasukan nomer revisi/commit.

git log 249241797562c13a118bb43c711b99b7d67b2854

Log Pada file tertentu

git log index.html

Log Revisi yang dilakukan oleh Author Tertentu

git log --author="WithStarup"

6. Git Diff

Git Diff berfungsi untuk melihat perbedaan perubahan pada revisi.

Gunakan perintah berikut ini untuk melihat perubahan yang dilakukan pada revisi tertentu.

git diff 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git diff 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b
diff --git a/index.html b/index.html
index e69de29..481cc02 100644
--- a/index.html
+++ b/index.html
@@ -0,0 +1,10 @@
+<!DOCTYPE html>
+<html>
+  <head>
+    <meta charset="utf-8">
+    <title>Belajar Git - Project 01</title>
+  </head>
+  <body>
+    <p>Hello Semua, Saya sedang belajar Git</p>
+  </body>
+</html>
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

Lihatlah hasil dari git diff, simbol plus (+) artinya kode yang ditambahkan. Sedangkan kalau ada kode yang dihapus simbolnya akan menggunakan minus (-). Contoh :

Ditambahkan:

+ <p>ini kode yang ditambahkan</p>

Dihapus:

- <i>ini kode yang dihapus</i>

Dimodifikasi/diubah:

- ini kode sebelum diubah
- + ini kode sesudah diubah

Coba ubah file index.html

Index.html	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Belajar Git - Project 01</title> </head> <body> <p>Hello Dunia!, Saya sedang belajar Git</p> </body> </html></pre>
------------	---

Setelah itu lakukan jalankan perintah **git diff**

Apa yang dilakukan git diff? Perintah git diff akan membandingkan perubahan yang baru saja dilakukan dengan revisi/commit terakhir.

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git diff
diff --git a/index.html b/index.html
index 481cc02..c5082e6 100644
--- a/index.html
+++ b/index.html
@@ -5,6 +5,6 @@
     <title>Belajar Git - Project 01</title>
   </head>
   <body>
-    <p>Hello Semua, Saya sedang belajar Git</p>
+    <p>Hello Dunia!, Saya sedang belajar Git</p>
   </body>
 </html>
```

Terdapat juga beberapa command git diff seperti dibawah ini :
Melihat Perbandingan pada File.

git diff index.html

Melihat Perbandingan antar Revisi/Commit

git diff <nomer commit> <nomer commit>

**git diff 74348572a4a5e0af5b35b31aafd05e9cca745494
e5785d7d34498869a1d143539c655b07de6fd484**

Perbandingan Antar Cabang (Branch)

git diff <nama cabang> <nama cabang>

Simpan Kembali revisi ketiga dengan git add kemudian git commit

git add index.html

git commit -m "perubahan sudah dilakukan"

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "perubahan sudah dilakukan"
[master 04d33a4] perubahan sudah dilakukan
1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

7. Membatalkan Revisi (Git Checkout, Git Reset, Git Revert)

Terkadang pada perubahan yang kita lakukan terjadi kesalahan dan kita ingin mengembalikannya seperti keadaan sebelumnya. Maka kita perlu menyuruh git untuk mengembalikannya. Ada beberapa perintah yang digunakan diantaranya: **git checkout**, **git reset**, dan **git revert**.

- **git checkout** -> Membatalkan revisi dalam kondisi revisi **belum staged** ataupun **committed**
- **git reset** -> Membatalkan revisi yang sudah dalam **kondisi staged**
- **git revert** -> Membatalkan revisi yang sudah dalam **kondisi committed**

7.1. git checkout

Jika revisi kita belum staged ataupun committed, kita bisa mengembalikannya menggunakan perintah **git checkout nama_file**

Contoh: Misalkan kita akan merubah isi dari file index.html pada repositori project-01

Index.html	<pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Belajar Git - Project 01</title> </head> <body> <p>Hello Dunia!, Saya sudah belajar Git</p> <p>Belajar git ternyata cukup menyenangkan</p> </body> </html> </pre>
------------	---

Pastikan hasil dari **git log** dan **git diff** seperti dibawah ini

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log --oneline
04d33a4 (HEAD -> master) perubahan sudah dilakukan
2492417 ditambahkan isi untuk HTML nya
5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git diff
diff --git a/index.html b/index.html
index c5082e6..19757c1 100644
--- a/index.html
+++ b/index.html
@@ -5,6 +5,7 @@
     <title>Belajar Git - Project 01</title>
   </head>
   <body>
-     <p>Hello Dunia!, Saya sedang belajar Git</p>
+     <p>Hello Dunia!, Saya sudah belajar Git</p>
+     <p>Belajar git ternyata cukup menyenangkan</p>
   </body>
- </html>
+ </html>
\ No newline at end of file
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Sekarang kita akan membatalkan perubahan tersebut. Sebelum melakukan pembatalan revisi silahkan coba **git status** terlebih dahulu, kemudian batalkan. Karena kita belum melakukan stage dan commit, maka kita bisa menggunakan perintah:

git checkout index.html

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Kemudian **git status** kembali untuk memastikan revisi kita telah dibatalkan

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git checkout index.html
Updated 1 path from the index
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
nothing to commit, working tree clean
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

7.2. git reset (Membatalkan Perubahan File yang Sudah dalam Kondisi staged)

Kondisi staged merupakan kondisi file yang sudah di add (git add), namun belum disimpan (git commit) ke dalam Git.

Sebagai contoh, kita lakukan perubahan lagi di file index.html seperti pada contoh sebelumnya.

Index.html	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Belajar Git - Project 01</title> </head> <body> <p>Hello Dunia!, Saya sudah belajar Git</p> <p>Belajar git ternyata cukup menyenangkan</p></pre>
------------	---

```
</body>
</html>
```

Setelah itu, kita ubah kondisi file menjadi staged dengan perintah:

Git add index.html

Cek statusnya terlebih dahulu dengan git status

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Nah, file index.html sudah masuk ke dalam kondisi staged. Untuk mengubahnya menjadi kondisi modified, kita bisa menggunakan perintah git reset.

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git reset index.html
Unstaged changes after reset:
  M       index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
On branch master
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

Sekarang file index.html sudah dalam kondisi modified, apabila ingin membatalkan perubahan, kita bisa membatalkan perubahannya dengan perintah git checkout seperti contoh sebelumnya pada poin 7.1. Namun saat ini kita akan lakukan commit untuk contoh pembatalan file yang sudah dalam kondisi committed.

7.3. Membatalkan Perubahan File yang Sudah dalam Kondisi Committed

Untuk melakukan pembatalan perubahan file yang sudah dalam kondisi committed, kita harus mengetahui nomer commit, kemudian mengembalikan perubahannya seperti pada nomer commit tersebut.

Lakukan commit file index.html pada poin terakhir

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "belajar git greget!"
[master 517d7f5] belajar git greget!
1 file changed, 1 insertion(+), 10 deletions(-)
```

Sekarang, lihat nomer git dengan git log


```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log
commit 517d7f57d0673d6acaa28394d3344d76770e316a (HEAD -> master)
Author: WithStarup <withstarup@gmail.com>
Date:   Fri Dec 15 18:26:51 2023 +0700

    belajar git greget!

commit 04d33a43f58a0d97032ddfd15654c2313671fd0b
Author: WithStarup <withstarup@gmail.com>
Date:   Fri Dec 15 17:56:02 2023 +0700

    perubahan sudah dilakukan

commit 249241797562c13a118bb43c711b99b7d67b2854
Author: WithStarup <withstarup@gmail.com>
Date:   Fri Dec 15 15:42:42 2023 +0700

    ditambahkan isi untuk HTML nya

commit 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b
Author: WithStarup <withstarup@gmail.com>
Date:   Fri Dec 15 15:36:51 2023 +0700

    Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

Yang berwarna merah adalah commit yang baru saja dilakukan, kemudian kita akan mengembalikan kondisi file index.html menjadi seperti pada commit sebelumnya (berwarna hijau). Maka kita bisa lakukan hal tersebut dengan perintah

git checkout [nomor commit]

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git checkout 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b
Note: switching to '5846d8469ee81c38d420f0c35a6c9e42aacbbd9b'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
HEAD is now at 5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log
commit 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b (HEAD)
Author: WithStarup <withstarup@gmail.com>
Date:   Fri Dec 15 15:36:51 2023 +0700

    Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
HEAD detached at 5846d84
nothing to commit, working tree clean
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

Perintah diatas akan mengembalikan semua file dalam kondisi pada nomer commit yang diberikan, namun bersifat temporer. Jika kita ingin mengembalikan semua file ke suatu commit, kita bisa melakukannya dengan perintah:

git revert -n [nomer commit]

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git revert -n 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log
commit 5846d8469ee81c38d420f0c35a6c9e42aacbbd9b (HEAD)
Author: WithStartup <withstartup@gmail.com>
Date:   Fri Dec 15 15:36:51 2023 +0700

    Commit Pertama

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git status
HEAD detached at 5846d84
You are currently reverting commit 5846d84.
(all conflicts fixed: run "git revert --continue")
(use "git revert --skip" to skip this patch)
(use "git revert --abort" to cancel the revert operation)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    about.html
        deleted:    contact.html
        deleted:    index.html

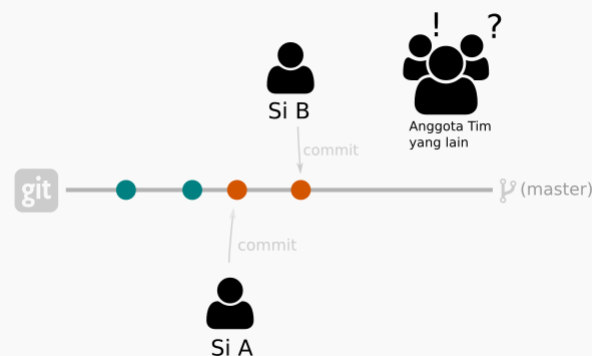
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

8. Menggunakan Percabangan (branch) untuk mencegah konflik

Apabila sedang bekerja dengan tim pada suatu repositori Git. Repositori ini dikerjakan secara bersama-sama. Terkadangkadang akan terjadi konflik, karena kode yang kita tulis berbeda dengan yang lain.

Misalnya, Si A menulis kode untuk fitur X dengan algoritma yang ia ketahui. Sedangkan si B menulis dengan algoritma yang berbeda. Lalu mereka melakukan commit, dan kode sumber jadi berantakan. Anggota tim yang lain menjadi pusing.



Agar tidak terjadi hal yang seperti ini, kita harus membuat cabang (branch) tersendiri. Misalnya, si A akan mengerjakan fitur X, maka dia harus membuat cabang sendiri. Si A akan bebas melakukan apapun di cabangnya tanpa mengganggu cabang utama (master).

8.1. Cara Membuat Cabang (Branch) Baru

Perintah untuk membuat cabang adalah `git branch`, kemudian diikuti dengan nama cabangnya.

```

git branch [nama branch]
git branch halaman_login

```

untuk pindah cabang, gunakan perintah

git checkout halaman_login

untuk melihat cabang apa saja yang ada di repository, gunakan perintah

git branch

Tanda bintang artinya cabang tempat kita berada saat ini.

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git branch
● * (HEAD detached at 5846d84)
  master
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> |
```

8.2. Perubahan File pada Cabang

Pastikan saat ini kita berada pada branch halaman_login, kemudian tambahkan file login.html

login.html	<p>Ini Halaman Login</p>
------------	--------------------------

Selanjutnya commit file login.html

```
git add login.html
git commit -m "membuat file login.html"
```

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add login.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "membuat file login.html"
[detached HEAD 1333d92] membuat file login.html
● 4 files changed, 1 insertion(+)
  delete mode 100644 about.html
  delete mode 100644 contact.html
  delete mode 100644 index.html
  create mode 100644 login.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> |
```

Revisi pada cabang halaman_login sudah disimpan. Sekarang coba kembali ke cabang master, kita lihat maka file login.html tidak ada. Kemudian kembali lagi ke cabang halaman_login ternyata ada. Dengan begitu, bisa kita ambil kesimpulan bahwa perubahan pada cabang halaman_login tidak akan berpengaruh di cabang master

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git branch
● * (HEAD detached from 5846d84)
  halaman_login
  master
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> |
```

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> dir

Directory: D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12

Mode                LastWriteTime         Length Name
----                -
-a----           12/15/2023   6:30 PM             199 index.html
-a----           12/15/2023   6:31 PM              24 login.html

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> |
```

8.3. Menggabungkan Cabang

Anggaplah kita sudah selesai membuat fitur login di cabang halaman_login. Sekarang kita ingin Menggabungkannya dengan cabang master (utama).

Pertama, kita harus pindah dulu ke cabang master.

git checkout master

Setelah itu, barulah kita bisa menggabungkan dengan perintah git merge.

git merge halaman_login

Sekarang lihat, file login.html sudah ada di cabang master.

Hati-hati! kadang sering terjadi bentrok ketika menggabungkan cabang.

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git checkout master
error: The following untracked working tree files would be overwritten by checkout:
    index.html
Please move or remove them before you switch branches.
Aborting
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git merge halaman_login
● Already up to date.
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> |
```

8.4. Mengatasi Bentrok

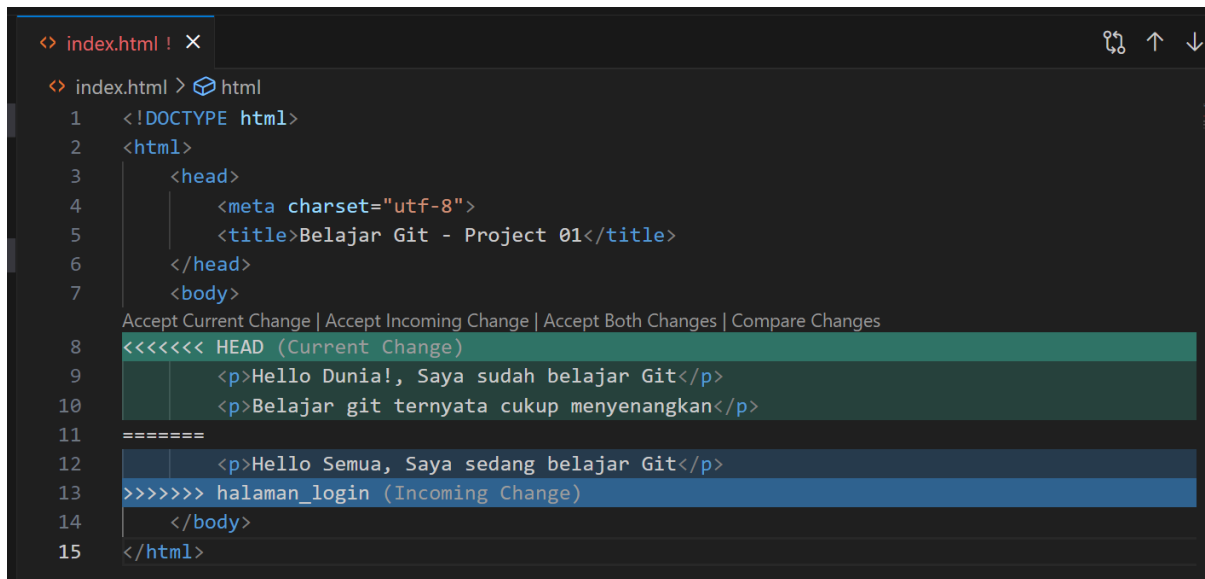
Bentrok biasanya terjadi jika ada dua orang yang mengedit file yang sama. Kenapa bisa begitu, 'kan mereka sudah punya cabang masing-masing?

Bisa jadi, di cabang yang mereka kerjakan ada file yang sama dengan cabang lain. Kemudian, saat digabungkan terjadi bentrok.

Mengatasi bentrok adalah tugas dari pemilik atau pengelola repostori. Dia harus bertindak adil, kode mana yang harus diambil.

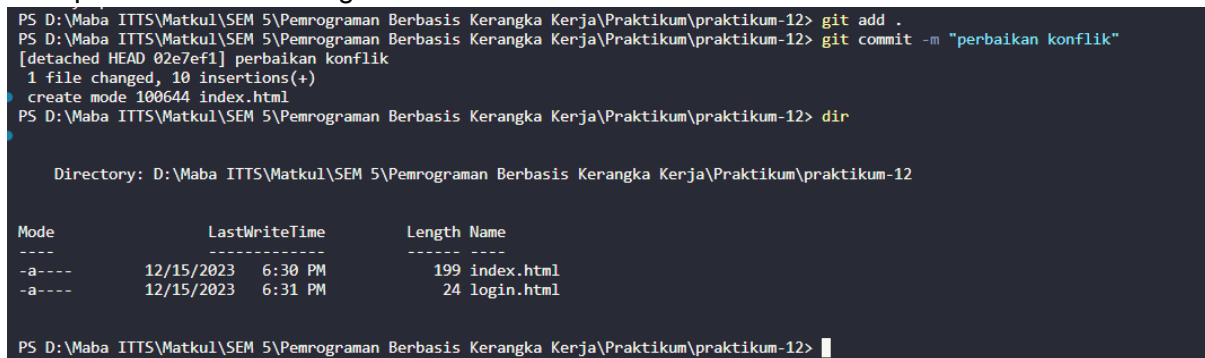
Biasanya akan ada proses diskusi dulu dalam mengambil keputusan.

Untuk memperbaiki conflict, buka text editor, kemudian pilih perubahan mana yang akan dipilih.



```
<> index.html ! x
<> index.html > html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Belajar Git - Project 01</title>
6      </head>
7      <body>
8          <<<<<< HEAD (Current Change)
9              <p>Hello Dunia!, Saya sudah belajar Git</p>
10             <p>Belajar git ternyata cukup menyenangkan</p>
11         =====
12             <p>Hello Semua, Saya sedang belajar Git</p>
13         >>>>>> halaman_login (Incoming Change)
14             </body>
15     </html>
```

Apabila sudah memilih, kemudian lakukan commit. Kemudian kita lihat di cabang master terdapat tambahan file login.html



```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add .
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "perbaikan konflik"
[detached HEAD 02e7ef1] perbaikan konflik
1 file changed, 10 insertions(+)
create mode 100644 index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> dir

Directory: D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12

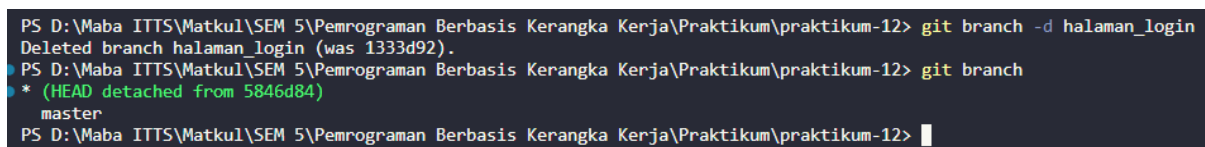
Mode                LastWriteTime         Length Name
----                -
-a----           12/15/2023   6:30 PM             199 index.html
-a----           12/15/2023   6:31 PM              24 login.html

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

8.5. Menghapus Cabang

Cabang yang sudah mati atau tidak ada pengembangan lagi, sebaiknya dihapus. Agar repository kita bersih dan rapi. Cara menghapus cabang, gunakan perintah git branch dengan argumen -d dan diikuti dengan nama cabangnya.

Contoh: **git branch -d halaman_login**



```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git branch -d halaman_login
Deleted branch halaman_login (was 1333d92).
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git branch
* (HEAD detached from 5846d84)
master
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

9. Bekerja dengan Remote Repository

Pada proyek pengembangan software yang melibatkan banyak orang (tim), kita tidak hanya akan menyimpan sendiri repository proyeknya. Semua tim yang terlibat dalam pengkodean (coding) akan menyimpan repository lokal di komputernya masing-masing.

Setelah itu, akan dilakukan penggabungan ke repository inti atau remote. Biasanya akan ada repository pusat atau untuk menyimpan source code yang sudah digabungkan (merge) dari beberapa orang.

Remote repository bisa disimpan di server kantor atau bisa juga menggunakan layanan seperti Github, Gitlab, Bitbucket, dll. Github adalah layanan yang paling populer untuk menyimpan (hosting) repository secara remote. Banyak proyek open source tersimpan di sana. Kita akan menggunakan Github pada praktik kali ini, pastikan kamu sudah memiliki akun Github.

9.1. Membuat Repository Github

Silahkan buka Github, kemudian buat sebuah repository dengan nama belajar-git seperti berikut ini.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

GueJuks /

Great repository names are short and memorable. Need inspiration? How about [urban-octo-chainsaw](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

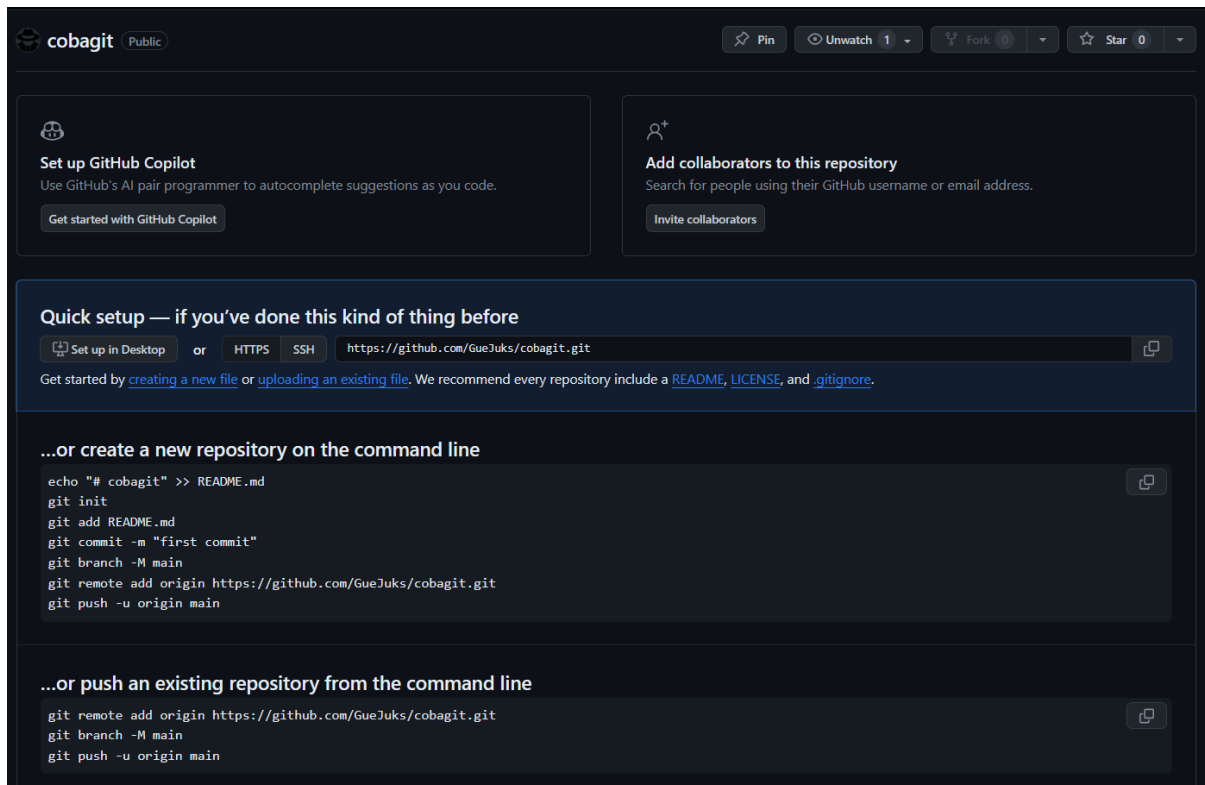
License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

Create repository

Maka sekarang kita punya repository kosong di Github.



9.2. Menambahkan Remote

Sebelum kita bisa upload semua revisi yang ada di repository lokal, kita harus menambahkan remote-nya terlebih dahulu.

Remote dapat kita tambahkan dengan perintah seperti ini:

Git remote add origin [url remote]

Git remote add origin https://github.com/dauditts/belajar-git.git

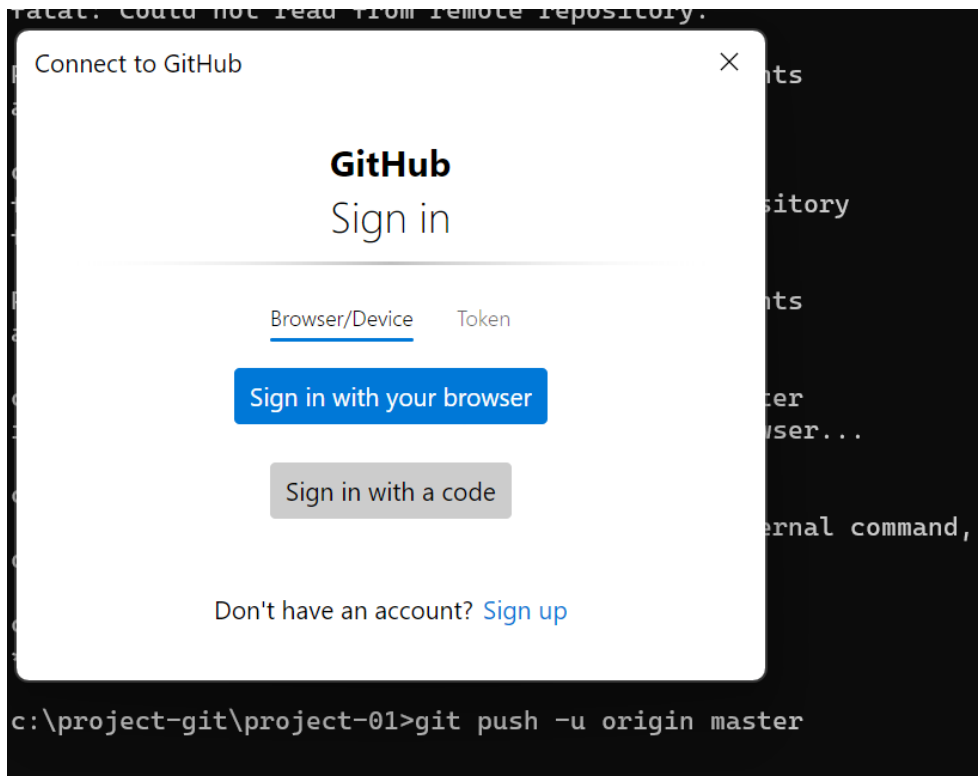
Kemudian cek apakah remote nya sudah ditambahkan dengan perintah **git remote -v**

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git remote add origin https://github.com/GueJuks/cobagit.git
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git remote -v
origin https://github.com/GueJuks/cobagit.git (fetch)
origin https://github.com/GueJuks/cobagit.git (push)
```

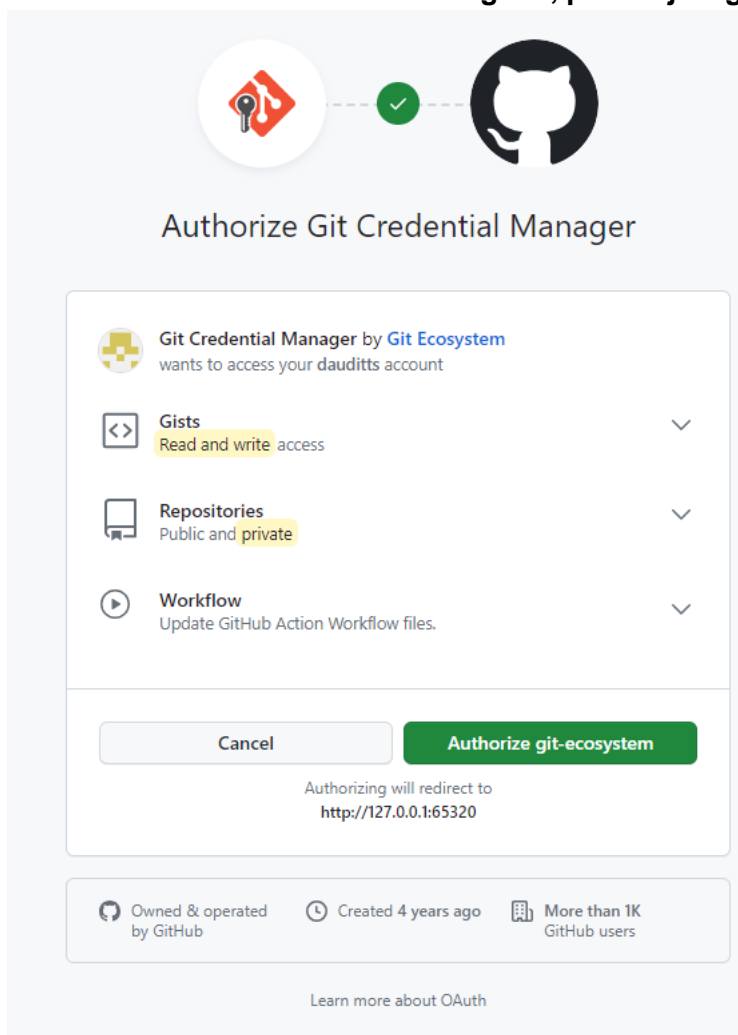
Mengirim Revisi ke Remote Repository

Perintah yang kita gunakan untuk mengirim revisi ke repository remote adalah git push.

git push -u origin master

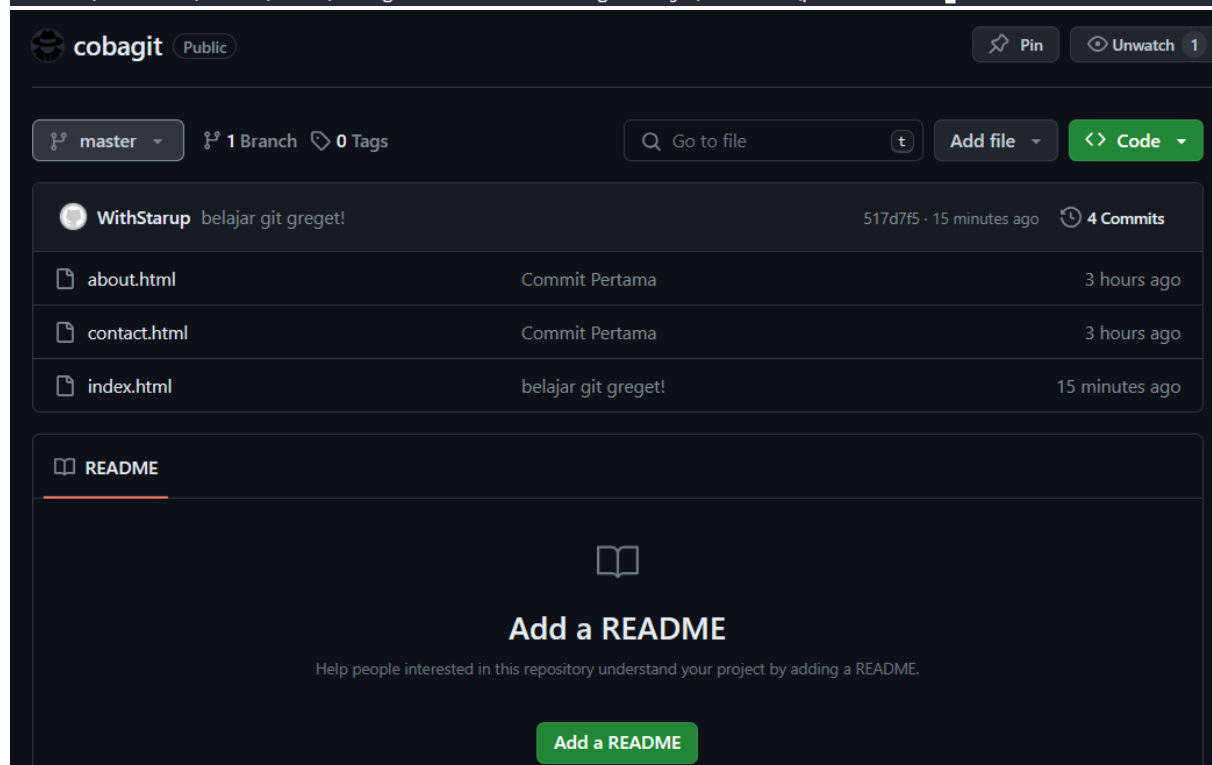


Akan diminta untuk melakukan sign in, pilih saja sign in dengan browser



Setelah itu akan diminta memasukkan password, bila berhasil file akan di upload ke repository

```
origin https://github.com/GueJuks/cobagit.git (push)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git push -u origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 1.00 KiB | 1.00 MiB/s, done.
Total 12 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/GueJuks/cobagit.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```



The screenshot shows the GitHub interface for a repository named 'cobagit'. At the top, there's a header with the repository name, a 'Public' badge, and buttons for 'Pin' and 'Unwatch'. Below the header, there's a navigation bar with 'master' selected, '1 Branch', '0 Tags', a search bar, and buttons for 'Add file' and 'Code'. The main content area shows a commit history table with columns for the commit message, commit hash, time ago, and number of commits. The table lists three commits: 'belajar git greget!' (517d7f5, 15 minutes ago, 4 commits), 'about.html' (Commit Pertama, 3 hours ago), and 'contact.html' (Commit Pertama, 3 hours ago). Below the table, there's a 'README' section with a large 'Add a README' button and a message: 'Help people interested in this repository understand your project by adding a README.'

Coba lakukan revisi pada file index.html kemudian commit dan push

index.html	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Belajar Git - Remote Repository</title> </head> <body> <p>Hello Dunia!, Saya sudah belajar Git</p> <p>Saya sedang belajar git Repository</p> </body> </html></pre>
------------	---

git add index.html

git commit -m "mengubah judul dan teks di body"

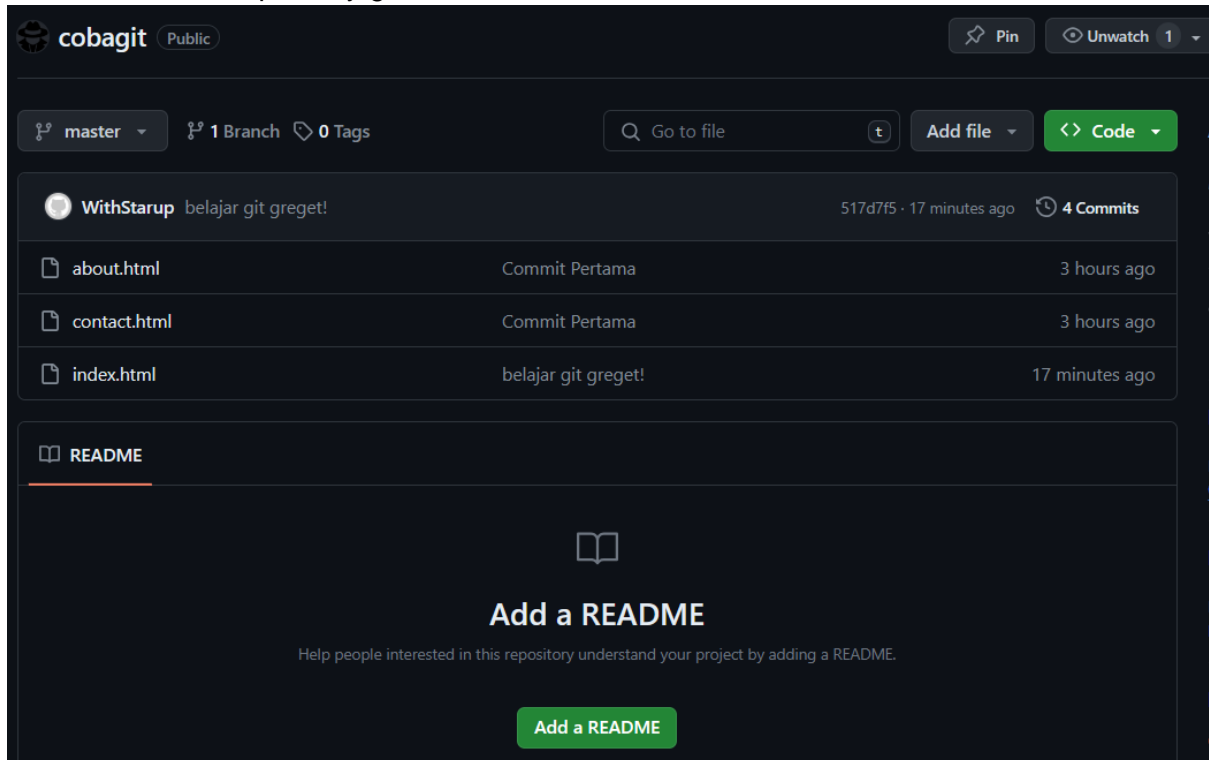
git push -u origin master

```

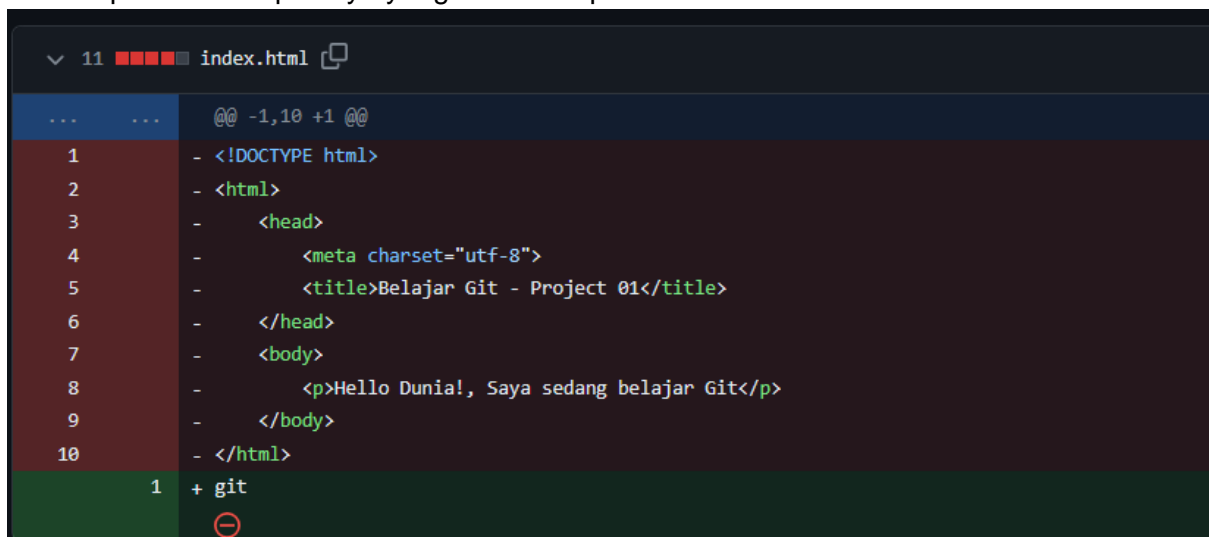
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git add index.html
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git commit -m "mengubah judul dan teks di body"
[detached HEAD 6f9a6e2] mengubah judul dan teks di body
1 file changed, 3 insertions(+), 2 deletions(-)
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git push -u origin master
Everything up-to-date
branch 'master' set up to track 'origin/master'.
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

Kemudian cek di repository github



Jika kita klik commit terakhir, maka kita akan dibawa ke git diff-nya Github. Di sana kita bisa melihat perubahan apa saja yang dilakukan pada commit tersebut.



9.3. Mengambil Revisi dari Remote Repository

Saat kita bekerja dengan repository yang memiliki banyak kontributor, kita seharusnya mengambil dulu revisi terbaru dari repository inti agar tidak bentrok. Misalnya begini. Pada

repository remote ada kontributor lain yang sudah menambahkan dan merubah sesuatu di sana. Maka kita harus mengambil perubahan tersebut, agar repository lokal kita tetap ter-update atau sama persis seperti repository remote.

Ada dua perintah untuk mengambil revisi dari repository remote:

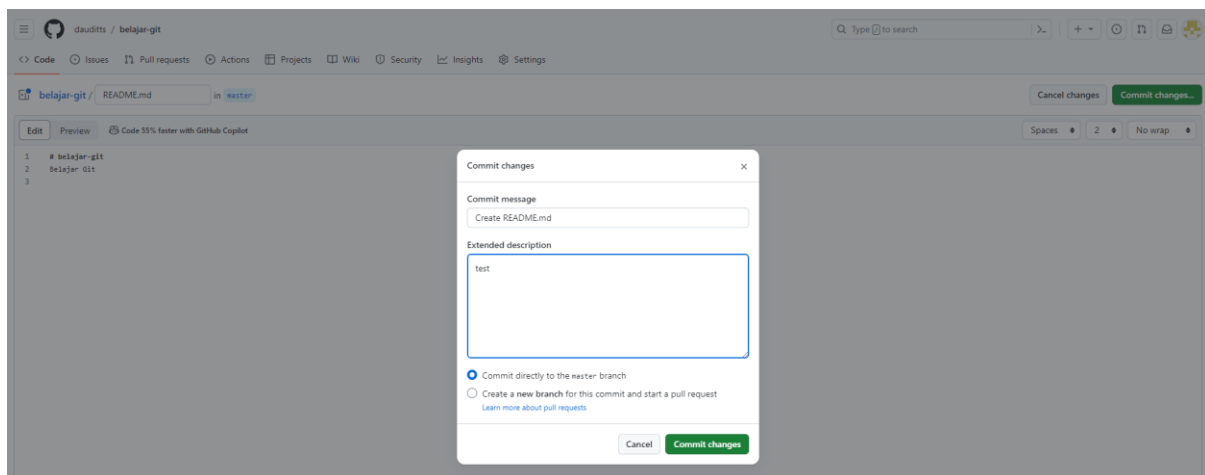
- `git fetch [nama remote] [nama cabang]`
- `git pull [nama remote] [nama cabang]`

Perintah `git fetch` hanya akan mengambil revisi (commit) saja dan tidak langsung melakukan penggabungan (merge) terhadap repository lokal. Sedangkan `git pull` akan mengambil revisi (commit) dan langsung melakukan penggabungan (merge) terhadap repository lokal.

Lalu kita harus pakai yang mana? Tergantung dari situasi dan kondisi. Bila kita sudah membuat perubahan di repository lokal, maka sebaiknya menggunakan `git fetch` agar perubahan yang kita lakukan tidak hilang. Namun, bila kita tidak pernah melakukan perubahan apapun dan ingin mengambil versi terakhir dari repository remote, maka gunakanlah `git pull`.

9.4. Mengambil Revisi dengan `git fetch`

Silahkan buka github, dan tambahkan file README.md melalui Github. Klik tombol add README. Setelah itu, isilah file README.md dengan apapun yang kamu inginkan. Kemudian lakukan commit. Maka di dalam remote repository akan ada tambahan file `readme.md`



Kemudian kita ambil file perubahan pada remote dengan perintah **git fetch**. File revisi sudah diambil, tapi belum ada file `readme.md` di lokal, karena `git fetch` hanya mengambil revisi saja, tidak langsung menggabungkannya dengan repository lokal.

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git fetch
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

```

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git fetch
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log --oneline
6f9a6e2 (HEAD) mengubah judul dan teks di body
02e7ef1 perbaikan konflik
1333d92 membuat file login.html
5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log origin/master --online
fatal: unrecognized argument: --online
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log origin/master --oneline
517d7f5 (origin/master, master) belajar git greget!
04d33a4 perubahan sudah dilakukan
2492417 ditambahkan isi untuk HTML nya
5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

Melihat perbedaan antara lokal master dan remote master dengan git diff

```

c:\project-git\project-01>git diff master origin/master
diff --git a/README.md b/README.md
new file mode 100644
index 0000000..c4592bd
--- /dev/null
+++ b/README.md
@@ -0,0 +1,2 @@
+# belajar-git
+Belajar Git
c:\project-git\project-01>

```

Lakukan perintah untuk merge seperti dibawah ini, kemudian kita cek bahwa readme.md sudah ada di dalam lokal.

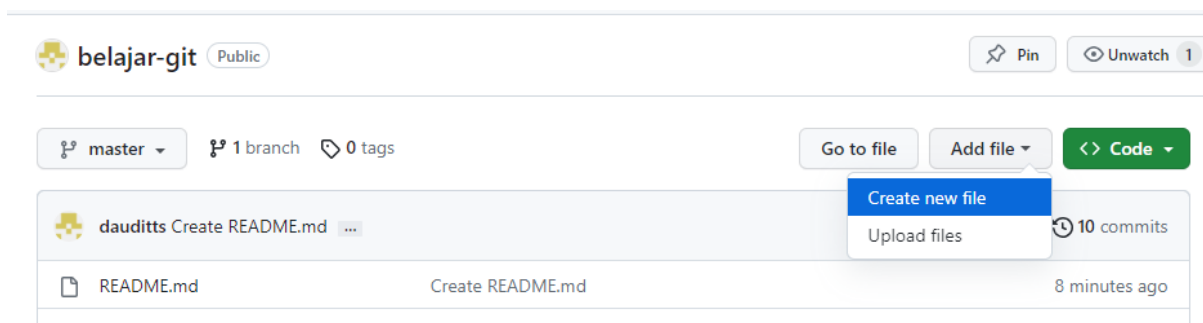
```

5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git diff master origin/master
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git merge master origin/master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>

```

9.5. Mengambil Revisi dengan git pull

Buat file register.html melalui Github. Kemudian commit.





Kemudian jalankan perintah git pull, maka file register.html akan langsung ada di lokal

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git pull
error: Pulling is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> git log --oneline
6f9a6e2 (HEAD) mengubah judul dan teks di body
02e7ef1 perbaikan konflik
1333d92 membuat file login.html
5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
5846d84 Commit Pertama
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> dir

Directory: D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12

Mode                LastWriteTime         Length Name
----                -
-a----           12/15/2023   6:47 PM             320 index.html
-a----           12/15/2023   6:31 PM              24 login.html

PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12>
```

9.6. Clone Repository

Clone repository bisa kita bilang seperti copy repository dari remote ke lokal. Perintah untuk melakukan clone adalah git clone [url remote] [directory]

Kita buat folder baru untuk menaruh project kita, kemudian kita jalankan perintah clone seperti dibawah ini

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> mkdir project-clone
```

```
Directory: D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12
```

Mode	LastWriteTime	Length	Name
d----	12/15/2023 6:51 PM		project-clone

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12> cd .\project-clone\
```

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12\project-clone> git clone https://github.com/GueJuks/cobagit.git
```

```
Cloning into 'cobagit'...
```

```
• remote: Enumerating objects: 15, done.  
remote: Counting objects: 100% (15/15), done.  
remote: Compressing objects: 100% (8/8), done.  
remote: Total 15 (delta 5), reused 11 (delta 4), pack-reused 0  
Receiving objects: 100% (15/15), done.  
Resolving deltas: 100% (5/5), done.
```

```
PS D:\Maba ITTS\Matkul\SEM 5\Pemrograman Berbasis Kerangka Kerja\Praktikum\praktikum-12\project-clone> |
```