Tutorial Git #6

Perintah untuk Membatalkan Revisi

Pada tutorial git 05, kita sudah belajar cara melihat perbedaan di setiap revisi. Sekarang kita akan belajar, cara membatalkan sebuah revisi.

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

Membatalkan Perubahan

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

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

Sebelum diubah:

```
<!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>
```

Setelah diubah:

```
<!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>
```

Hasil **git diff**:

```
$ git diff
diff --git a/index.html b/index.html
index c5082e6..115efcb 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>
```

Sekarang kita akan membatalkan perubahan tersebut. Karena kita belum melakukan *stage* dan *commit*, maka kita bisa menggunakan perintah:

```
git checkout index.html
```

Perubahan yang baru saja kita lakukan akan dibatalkan. Kalau tidak percaya, coba saja periksa file yang sudah dirubah tadi atau cek dengan perintah ***git status***.

```
$ git status
On branch master
nothing to commit, working directory clean
```

Hati-hati! Terkadang perintah ini sangat berbahaya, karena akan menghapus perubahan yang baru saja dilakukan.

Bila kita sudah merubah banyak hal, maka itu akan sia-sia setelah menjalankan perintah ini.

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.

```
$ git diff
diff --git a/index.html b/index.html
index c5082e6..c99aa5b 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 gampang-gampang susah</p>
    </body>
 </html>
```

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

```
git add index.html
```

Cek statunya dulu:

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   index.html
```

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

```
git reset index.html
```

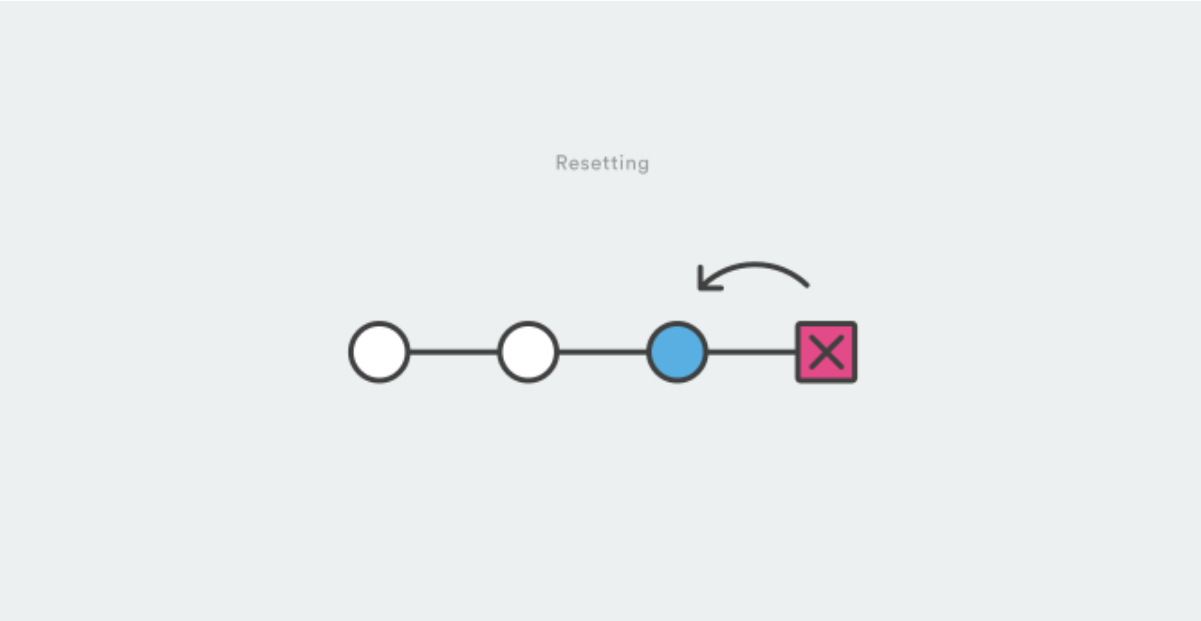
Cek statusnya lagi:

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Sekarang file index.html sudah dalam kondisi *modified*, kita bisa membatalkan perubahannya dengan perintah git checkout seperti contoh sebelumnya.

```
git checkout index.html
```

Maka perubahan yang kita lakukan akan dibatalkan, .



Membatalkan Perubahan File yang Sudah dalam Kondisi *Commited*

Sekarang bagaimana kalau filenya sudah dalam kondisi *commited* dan kita ingin mengembalikannya? Untuk melakukan ini, kita harus mengetahui nomer *commit*, kemudian mengembalikan perubahannya seperti pada nomer *commit* tersebut.

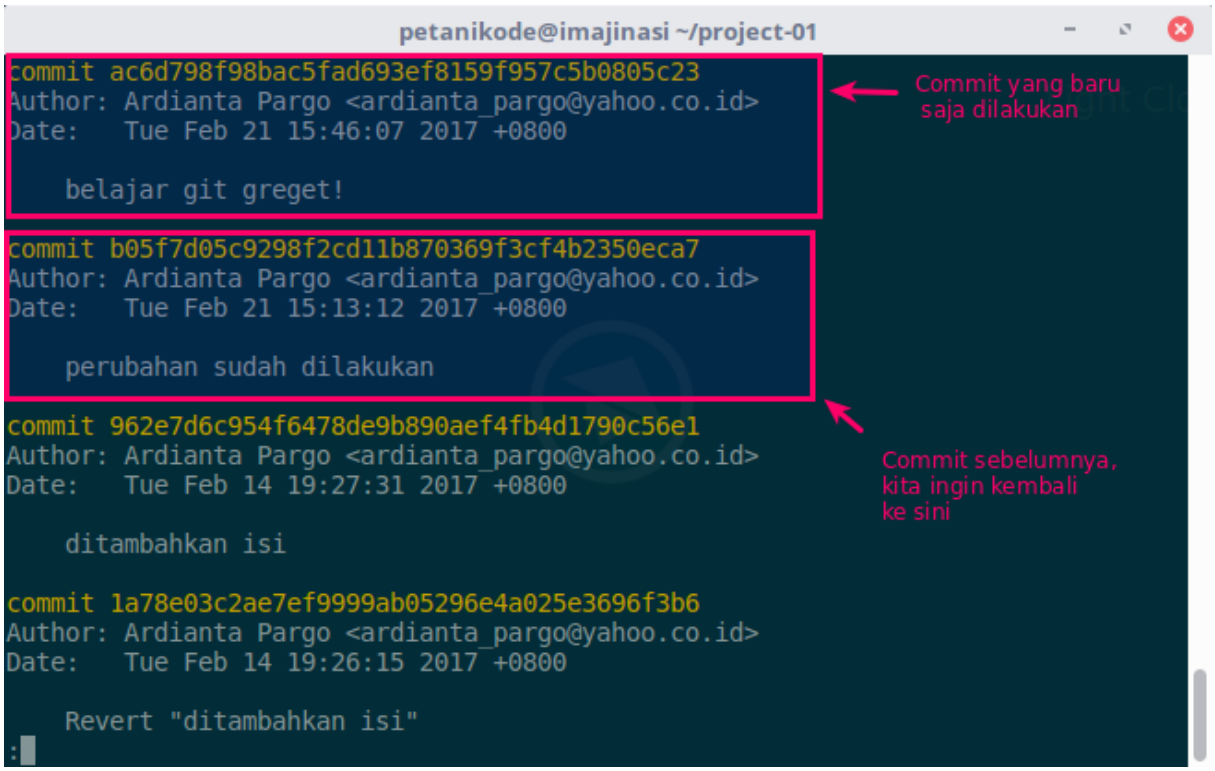
Misalkan, kita ubah kembali file index.html.

```
$ git diff
diff --git a/index.html b/index.html
index c5082e6..3c150a8 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 Greget!</p>
    </body>
 </html>
```

Kemudian kita melakukan *commit*.

```
git add index.html
git commit -m "belajar git greget!"
```

Sekarang kita akan melihat nomer commit dengan perintah git log.



Kita akan mengembalikan kondisi file index.html, seperti pada *commit* sebelumnya. Maka kita bisa menggunakan perintah:

```
git checkout b05f7d05c9298f2cd11b870369f3cf4b2350eca7 index.html
```

kita sudah mengembalikan keadaan file index.html seperti keadaan saat *commit* tersebut. Namun, saat ini kondisi index.html dalam keadaan *staged*. Kita bisa kembalikan ke dalam kondisi *modified* dengan perintah git reset.

```
git reset index.html
```

Pada contoh tersebut, kita sudah berhasil mengembalikan file index.html ke dalam keadaan seperti *commit* sebelumnya.

Apabila kita ingin mengembalikan seluruh file dalam commit, kita cukup melakukan *checkout* ke nomer *commit* saja, tanpa diikuti nama file. Contoh:

```
git checkout ac6d798f98bac5fad693ef8159f957c5b0805c23
```

Catatan: Perintah ini akan mengembalikan semua file dalam kondisi pada nomer commit yang diberikan, namun bersifat temporer.

Kembali ke 3 Commit sebelumnya

Untuk kembali ke 3 *commit* sebelumnya, kita bisa menggunakan perintah berikut.

```
git checkout HEAD~3 index.html
```

Membatalkan Semua Perubahan yang ada

Jika kita ingin mengembalikan semua file ke suatu commit, kita bisa melakukannya dengan perintah:

```
git revert -n <nomer commit>
```

Contoh:

```
git revert -n 2400ba0e258bd6a144caa273012b130d6baa5e42
```