

Tutorial Git #8

Perbedaan Git checkout, Git Reset, dan Git Revert

Pada tutorial ke-6, kita sudah membahas cara membatalkan revisi menggunakan perintah git checkout, git reset, dan git revert.

Sekilas, tiga perintah tersebut melakukan hal yang sama. Akan tetapi ada perbedaannya.

Apa itu?

Mari kita bahas...

Git Checkout

Perintah git checkout seperti mesin waktu, kita bisa kembalikan kondisi file proyek seperti waktu yang dituju.

Misalnya:

```
git checkout 06f735af7724558164c87f6b1ce3ca7778eb1c1b
```

Maka semua file akan dikembalikan seperti keadaan pada nomer commit tersebut.



Akan tetapi, ini bersifat temporer (sementara). Pengembalian ini tidak disimpan dalam database Git.

Kita bisa sebut perintah git checkout sebagai perintah untuk mengecek kondisi file di setiap commit.

Tips: Untuk kembali dari masa lalu gunakan perintah: git checkout master

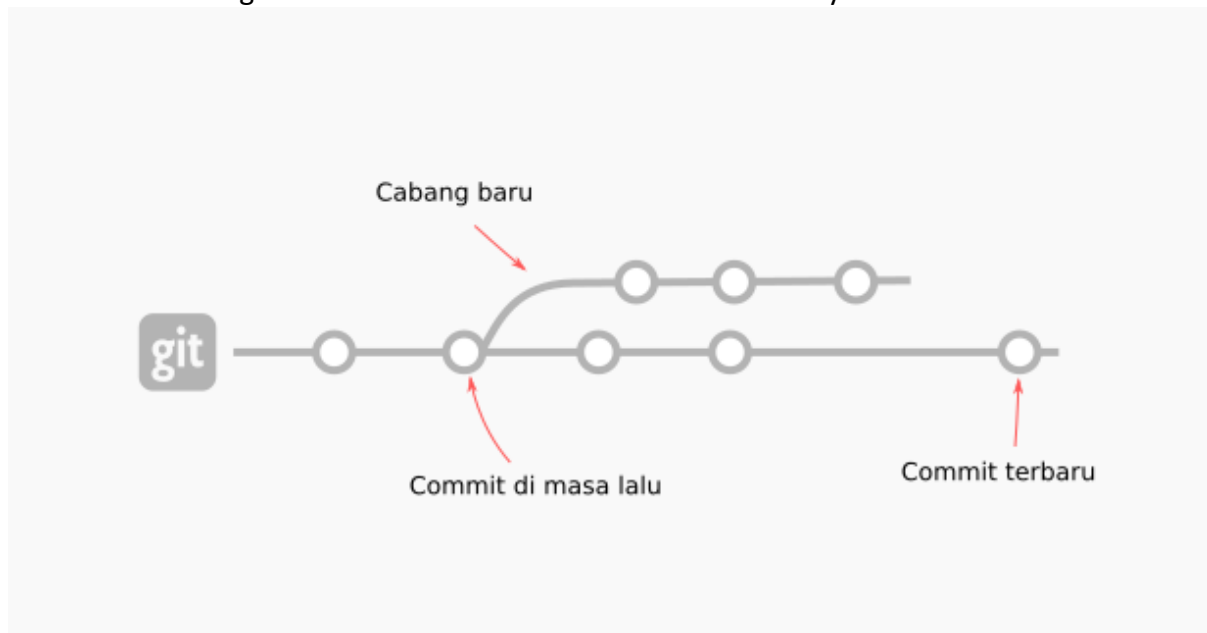
Selain itu juga, perintah ini digunakan untuk berpindah dan membuat cabang. Ini bisa kita gunakan untuk membuat perubahan baru berdasarkan kode di masa lalu.

Contoh:

Misalnya kita ingin membuat cabang baru berdasarkan kondisi kode di masa lalu, maka kita bisa menggunakan perintah:

```
git checkout -b nama_cabang <nomer_commit>
```

Maka nanti cabang baru akan terbentuk dari commit sebelumnya.

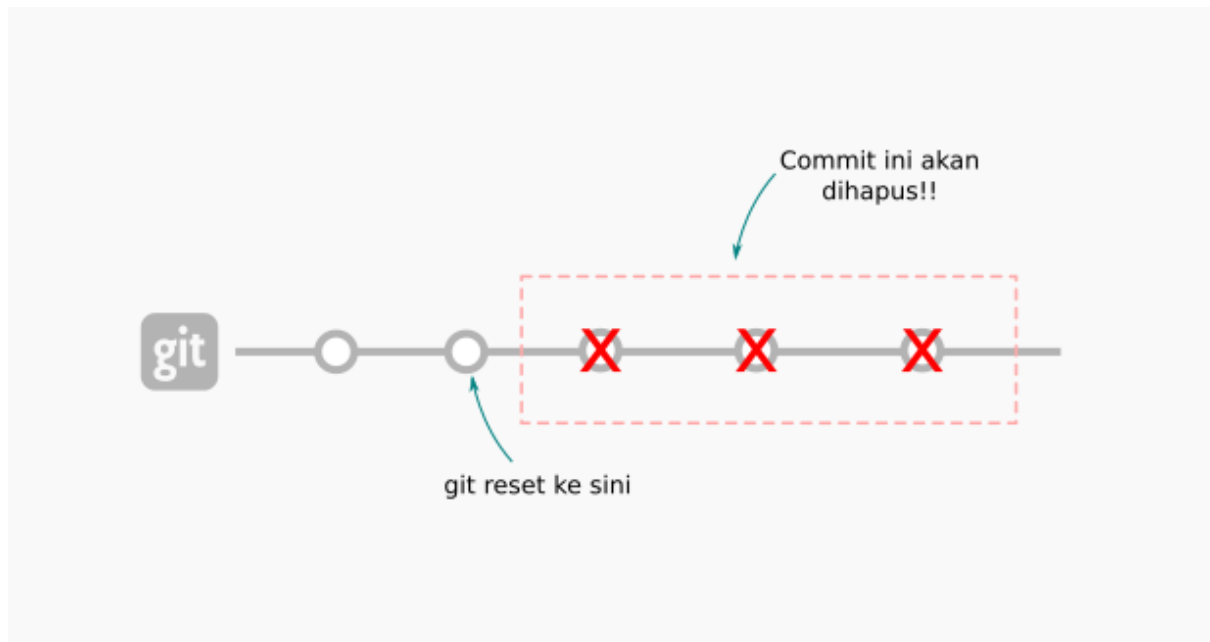


Cara ini bisa kita ibaratkan seperti menulis cerita baru dengan plot yang berbeda.

Git Reset

Perintah git reset sering disebut sebagai perintah berbahaya yang dapat menghancurkan catatan sejarah perubahan.

Hati-hati! Perintah ini membuat kita tidak bisa kembali lagi ke masa depan. Mau tidak mau, kita harus menulis ulang sejarah.



Perintah ini memiliki tiga argumen atau opsi utama, yaitu **--soft**, **--mixed**, dan **--hard**.

- **--soft** akan mengembalikan dengan kondisi file dalam keadaan *staged*
- **--mixed** akan mengembalikan dengan kondisi file dalam keadaan *modified*
- **--hard** akan mengembalikan dengan kondisi file dalam keadaan *committed*

Contoh penggunaan:

```
git reset --soft a85ce393eacbffa0ec4bee18f8340f54d3d95d8a
```

Maka kita akan dikembalikan dengan file dalam keadaan *staged*.

```
proyek-01 — -bash — 76x10
[Firdausa-2:proyek-01 Firdausa$ git reset --soft a85ce393eacbffa0ec4bee18f834]
0f54d3d95d8a
[Firdausa-2:proyek-01 Firdausa$ git status
On branch halaman_login
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   login.html

Firdausa-2:proyek-01 Firdausa$
```

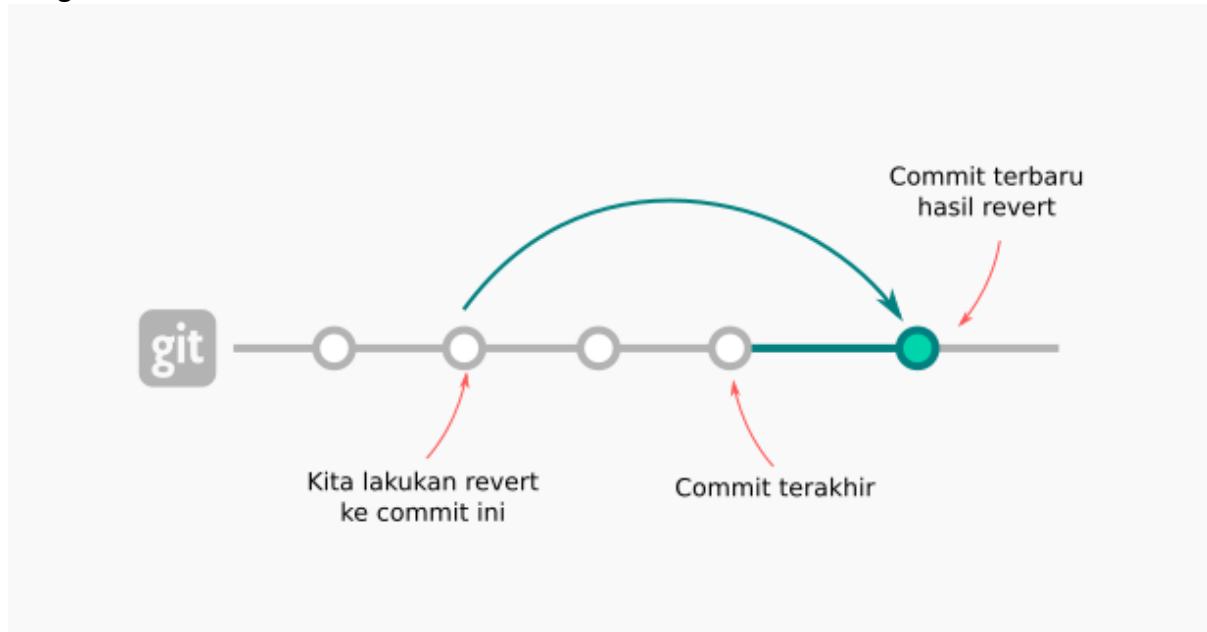
Coba periksa catatan perubahan dengan perintah **git log**, pasti ada yang hilang dan kita tidak akan bisa kembali lagi ke masa depan. Ini seperti melakukan Re:Zero .

Hati-hati! Jangan lakukan git reset pada repositori yang sudah di bagikan ke publik, karena dapat merusaknya.

Git Revert

Revert artinya mengembalikan. Perintah ini lebih aman daripada git reset, karena tidak akan menghapus catatan sejarah revisi.

Revert akan mengambil kondisi file yang ada di masa lalu, kemudian menggabungkannya dengan commit terakhir.



Masih belum paham?

Mari kita coba lakukan sedikit eksperimen.

Sebelumnya, berikut ini adalah kondisi repositori yang dijadikan bahan percobaan

```
proyek-01 — less • git log — 76x18
commit 74b88a3d80ed53e5e8b8498af01f14616dc6b94b
Merge: 6fbcc67 c4871f6
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
Date: Wed Jul 31 07:47:00 2019 +0700

    perbaiki konflik

commit 6fbcc678d2b0cb59fe6843acde8819849af1db0a
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
Date: Wed Jul 31 07:39:45 2019 +0700

    ubah isi login.html di cabang master

commit c4871f69c242f9479e9884f91c4b7ab91677f8be
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
Date: Wed Jul 31 07:36:19 2019 +0700
```

Ada beberapa commit di sana, kemudian kita ingin melakukan **revert** ke commit ke-2 yang memiliki message “ubah isi login.html di cabang master”.

```
proyek-01 — -bash — 76x7
[Firdausa-2:proyek-01 Firdausa$ git revert 6fbcc678d2b0cb59fe6843acde8819849a]
f1db0a
error: could not revert 6fbcc67... ubah isi login.html di cabang master
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
Firdausa-2:proyek-01 Firdausa$
```

Terjadi error, karena ada file yang bentrok.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Login</title>
5  </head>
6  <body>
7      <<<<<<< HEAD
8      Ini Halaman Login diubah oleh cabang halaman_login
9      =====
10     Ini Halaman Login
11     >>>>>>> parent of 6fbcc67... ubah isi login.html di cabang master
12 </body>
13 </html>
```

Silahkan atasi konflik ini, setelah itu lakukan commit lagi.

```
proyek-01 — less - git log — 76x22
[Firdausa-2:proyek-01 Firdausa$ git log
commit b399aeb49e082d39c3dd2dc17ea1167b678120a9 (HEAD -> master)
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
Date:   Wed Jul 31 10:57:02 2019 +0700

    update login.html hasil revert

commit 07c23427c382650c548183f4726afc8d7a0af334
Merge: 74b88a3 c1eb01e
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
Date:   Wed Jul 31 10:49:52 2019 +0700

    commit login from master

commit c1eb01e66264cddb3f6cd18e499b3d3b62ad2c3c (halaman_login)
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
Date:   Wed Jul 31 10:48:18 2019 +0700

    commit

commit 710a315ef18e198c825f5875af71a9cb01df84e9
Author: Firdausa <firdausa@smktelkom-mlg.sch.id>
```

Nah, sekarang kita sudah punya catatan revisi/commit baru di dalam repositori.

Kesimpulan

- Perintah **git checkout** mengembalikan file dalam kondisi sebelumnya, tapi bersifat sementara.
- Perintah **git reset**, akan mengembalikan file ke kondisi sebelumnya, kemudian menghapus catatan sejarah commit berikutnya.
- Perintah **git revert** mengembalikan file dengan tidak menghapus sejarah commit.