

Tutorial Git #7

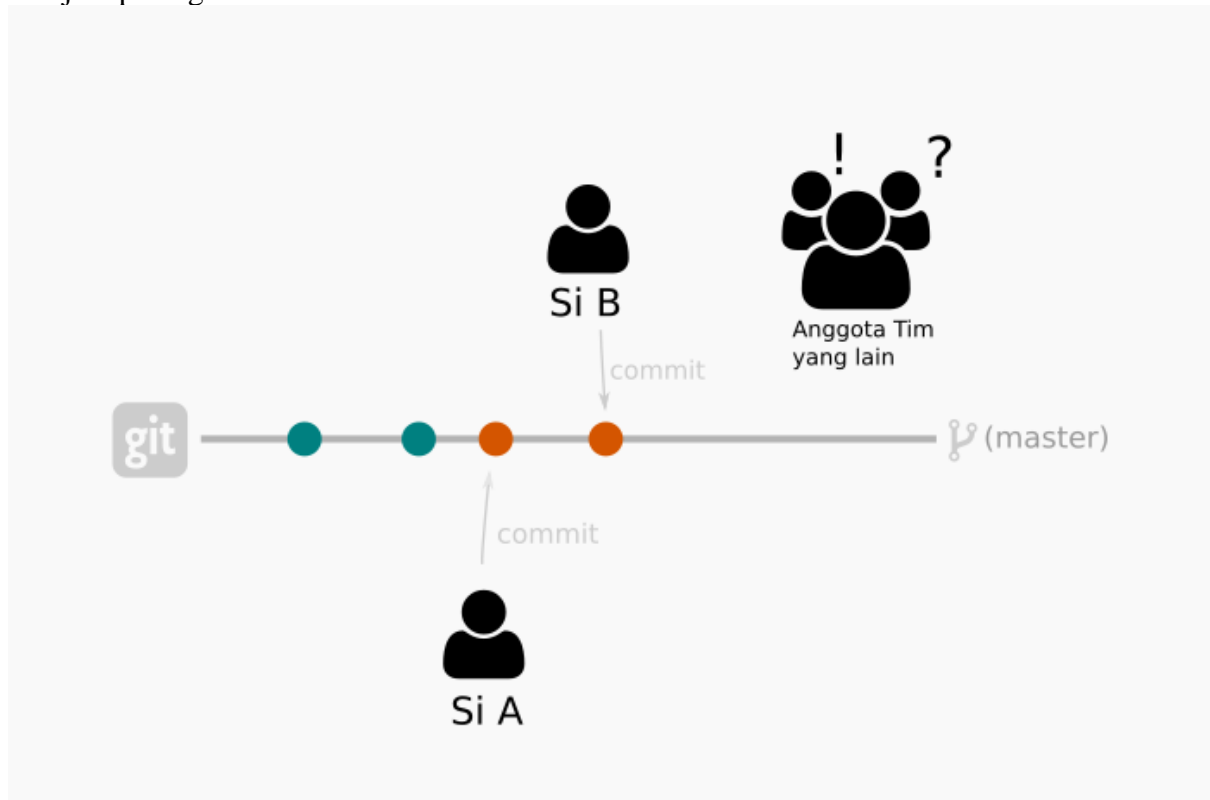
Menggunakan Percabangan untuk Mencegah Konflik

Bayangkan anda sedang bekerja dengan tim pada suatu repositori Git. Repositori ini dikerjakan secara bersama-sama.

Kadang... 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*).

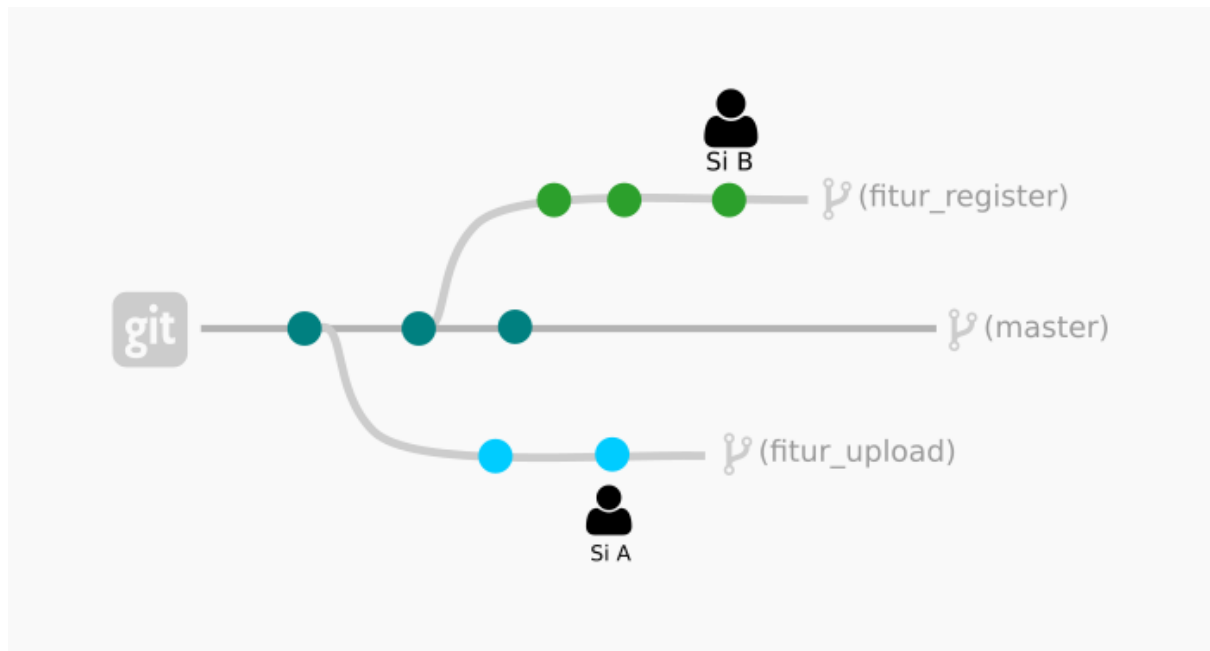
Cara Membuat Cabang Baru

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

Contoh:

```
git branch fitur_register
```

Maka Git akan membuat cabang bernama `fitur_register`.



Sekarang setiap orang memiliki cabangnya masing-masing. Mereka bebas bereksperimen. Untuk melihat cabang apa saja yang ada di repositori, gunakan perintah `git branch`.

Contoh:

```
projek-01 — -bash — 76x5
[Firdausa-2:projek-01 Firdausa$ git branch fitur_register
[Firdausa-2:projek-01 Firdausa$ git branch
  fitur_register
* master
Firdausa-2:projek-01 Firdausa$
```

Tanda bintang (*) artinya cabang yang sedang aktif atau Kita sedang berada di sana.

Latihan

Untuk memantapkan pemahaman tentang percabangan Git, mari kita coba praktik. Pada repositori, buatlah sebuah cabang baru.

```
git branch halaman_login
```

Setelah itu, pindah ke cabang yang baru saja kita buat dengan perintah:

```
git checkout halaman_login
```

Lalu tambahkan file `login.html`, isinya terserah anda.

```
proyek-01 — -bash — 76x14
[Firdausa-2:proyek-01 Firdausa$ git branch fitur_register
[Firdausa-2:proyek-01 Firdausa$ git branch
  fitur_register
* master
[Firdausa-2:proyek-01 Firdausa$ git branch halaman_login
[Firdausa-2:proyek-01 Firdausa$ git status
On branch master
nothing to commit, working tree clean
[Firdausa-2:proyek-01 Firdausa$ git checkout halaman_login
Switched to branch 'halaman_login'
[Firdausa-2:proyek-01 Firdausa$ git status
On branch halaman_login
nothing to commit, working tree clean
Firdausa-2:proyek-01 Firdausa$
```

Buat sebuah file **login.html** baru kemudian masukan perintah *git status*

```
proyek-01 — -bash — 76x9
[Firdausa-2:proyek-01 Firdausa$ git status
On branch halaman_login
Untracked files:
  (use "git add <file>..." to include in what will be committed)

  login.html

nothing added to commit but untracked files present (use "git add" to track)
Firdausa-2:proyek-01 Firdausa$
```

*Tips: Jangan lupa untuk menggunakan perintah **git status** untuk melihat status repositori.*

Kita sudah menambahkan file login.html. Selanjutnya kita lakukan *commit*.

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

Bagus! revisi kita pada cabang halaman_login sudah disimpan. Sekarang coba kembali ke cabang master.

```
git checkout master
```

```
proyek-01 — -bash — 76x5
[Firdausa-2:proyek-01 Firdausa$ git checkout master
Switched to branch 'master'
[Firdausa-2:proyek-01 Firdausa$ ls
about.html      index.html
Firdausa-2:proyek-01 Firdausa$
```

Apakah anda menemukan file login.html?

Pasti tidak!

Sekarang kembali lagi ke cabang halaman_login.

```
git checkout halaman_login
```

Cek lagi, apakah sekarang file login.html sudah ada?

```
proyek-01 — -bash — 76x7
[Firdausa-2:proyek-01 Firdausa$ git branch
  fitur_register
* halaman_login
  master
[Firdausa-2:proyek-01 Firdausa$ ls
about.html      index.html      login.html
Firdausa-2:proyek-01 Firdausa$
```

Ternyata ada. kita bisa mengambil kesimpulan, kalau perubahan pada cabang halaman_login tidak akan berpengaruh di cabang master.

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.

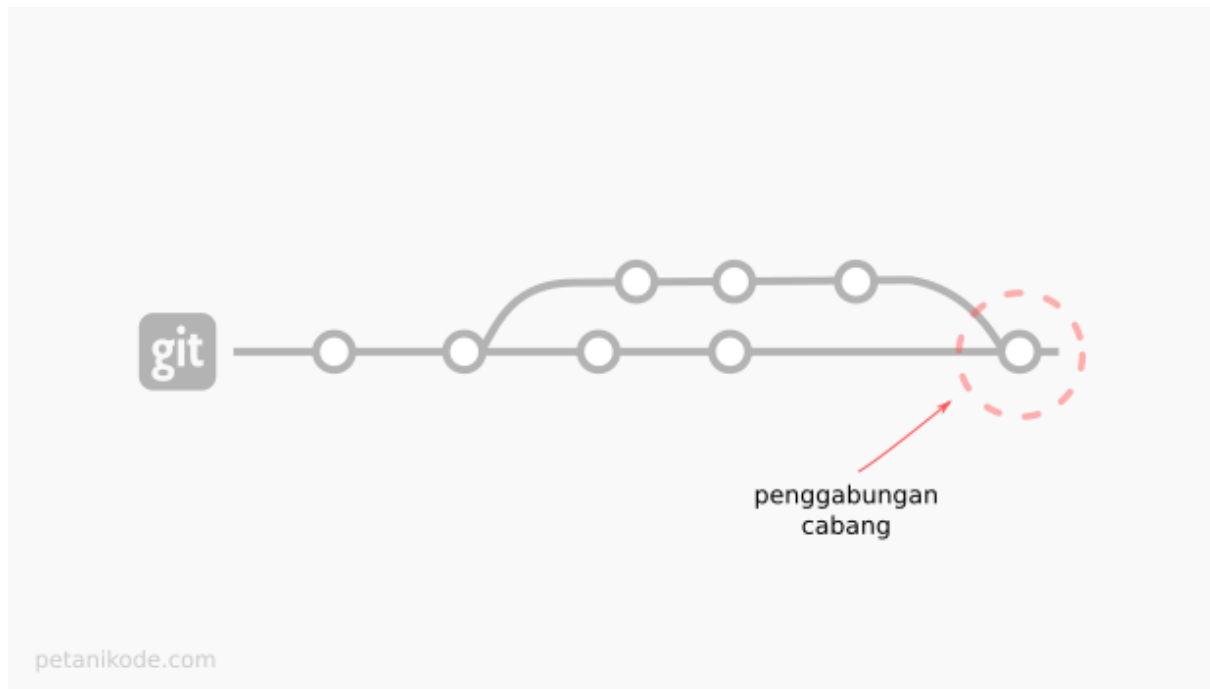
```
git checkout master
```

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

```
git merge halaman_login
```

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

```
proyek-01 — -bash — 76x9
[Firdausa-2:proyek-01 Firdausa$ git merge halaman_login
Updating c8ca710..a85ce39
Fast-forward
 login.html | 9 ++++++++
 1 file changed, 9 insertions(+)
 create mode 100644 login.html
[Firdausa-2:proyek-01 Firdausa$ ls
about.html      index.html      login.html
Firdausa-2:proyek-01 Firdausa$
```



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

Mengatasi Bentrok/Konflik

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.

Baiklah, sekarang kita akan coba membuat bentrokan .

Pindah dulu ke **branch halaman_login...**

```
git checkout halaman_login
```

Setelah itu, edit file **login.html** , karena sekarang kedua file tersebut ada di kedua cabang yang akan kita gabungkan.

Berikut hasil perubahan script pada file **login.html**

```
proyek-01 — -bash — 76x14
[Firdausa-2:proyek-01 Firdausa$ git diff
diff --git a/login.html b/login.html
index 2274dfe..7acdd8a 100644
--- a/login.html
+++ b/login.html
@@ -4,6 +4,6 @@
<title>Login</title>
</head>
<body>
-   Ini Halaman Login
+   Ini Halaman Login diubah oleh cabang halaman_login
</body>
</html>
Firdausa-2:proyek-01 Firdausa$
```

Setelah itu, lakukan *commit* lagi:

```
git add login.html
git commit -m "ubah isi login.html"
```

Selanjutnya pindah ke **branch master** dan lakukan perubahan juga di cabang ini. Ubah file yang sama seperti di cabang halaman_login yaitu file **login.html**.

```
proyek-01 — -bash — 76x16
diff --git a/login.html b/login.html
index 31ac298..a5f2789 100644
--- a/login.html
+++ b/login.html
@@ -4,6 +4,6 @@
<title>Login</title>
</head>
<body>
-   Ini Halaman Login
+   Ini Halaman Login diubah oleh cabang master
</body>
- </html>
\ No newline at end of file
+ </html>
~
Firdausa-2:proyek-01 Firdausa$
```

Setelah itu, lakukan *commit* di cabang master

```
git add login.html
git commit -m "ubah isi login.html di cabang master"
```

Terakhir, coba gabungkan cabang halaman_login dengan cabang master, maka akan terjadi bentrok.

```
$ git merge halaman_login
Auto-merging login.html
CONFLICT (content): Merge conflict in login.html
Automatic merge failed; fix conflicts and then commit the result.
```

Nah, kita disuruh perbaiki kode yang bentrok. Sekarang buka login.html dengan teks editor.

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

Kedua kode cabang dipisahkan dengan tanda =====.

Sekarang.. tugas kita adalah memperbaikinya dengan cara mengeliminasi atau menentukan salah satu dari kode tersebut yang akan digunakan.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Login</title>
5  </head>
6  <body>
7    Ini Halaman Login diubah oleh cabang halaman_login
8  </body>
9  </html>
```

Setelah itu lakukan *commit* untuk menyimpan perubahan ini.

```
git add login.html
git commit -m "perbaiki konflik"
```

Mantappp! Sekarang bentrokan kode antar programmer sudah teratasi.

Menghapus Cabang

Cabang yang sudah mati atau tidak ada pengembangan lagi, sebaiknya dihapus.

Agar repositori 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
```