

Workbook Git Branch dan Merge



Endy Muhardin
endy@artivisi.com
last updated : 11 Jan 2011

Daftar Isi

Membuat branch.....	3
Penjelasan.....	3
Panduan.....	3
Kesimpulan.....	3
Compare Branch.....	4
Penjelasan.....	4
Panduan.....	4
Kesimpulan.....	4
Merge fast-forward.....	5
Penjelasan.....	5
Panduan.....	5
Kesimpulan.....	5
Merge rekursif.....	6
Penjelasan.....	6
Panduan.....	6
Kesimpulan.....	6
Rebase	7
Penjelasan.....	7
Panduan.....	7
Kesimpulan.....	7
Merge Conflict	8
Penjelasan.....	8
Panduan.....	8
Kesimpulan.....	9
Menghindari Merge Fast-forward	10
Penjelasan.....	10
Panduan.....	10
Kesimpulan.....	10
Membatalkan Merge.....	11
Penjelasan.....	11
Panduan.....	11
Kesimpulan.....	11
Menghapus Branch	12
Penjelasan.....	12
Panduan.....	12
Kesimpulan.....	12

Membuat branch

Penjelasan

Branch kita gunakan untuk paralel development. Best practicesnya adalah selalu bekerja di topic branch, pada saat sudah selesai, baru merge ke branch utama.

Panduan

No	Aktifitas	Hasil
1	Lihat branch apa saja yang tersedia git branch	* master hanya ada 1 branch, namanya master, sedang aktif (ada * nya)
2	Buat branch bernama fitur-kalender git branch fitur-kalender git branch	fitur-kalender * master
3	Pindah ke branch fitur-kalender git checkout fitur-kalender git branch	* fitur-kalender master
4	Buat file baru Kalender.java, kemudian commit file tersebut. git add Kalender.java git commit -m "template class kalender, belum diimplementasikan"	1 files changed, 7 insertions(+), 0 deletions(-) create mode 100644 BelajarGit/src/belajar/git/Kalender.java
5	File Kalender.java hanya ada di branch fitur-kalender, tidak ada di master. git checkout master Refresh file explorer	Switched to branch 'master' File Kalender.java menghilang dari folder.
6	Kembali ke branch fitur-kalender git checkout fitur-kalender Refresh file explorer	Switched to branch 'fitur-kalender' File Kalender.java kembali ada di dalam folder.

Kesimpulan

Dengan menggunakan branch, kita dapat memisahkan perubahan yang kita buat supaya tidak mengganggu branch utama (master).

Compare Branch

Penjelasan

Agar mengetahui perbandingan perubahan antara 1 branch dengan branch lainnya, pada latihan ini kita akan membuat 1 branch baru yaitu fitur-calculasi, dimana nantinya kita akan melakukan perubahan di branch fitur-calculasi, kemudian membandingkan perubahan antara branch master dan fitur-calculasi.

Panduan

No	Aktifitas	Hasil
1	Lihat branch apa saja yang tersedia git branch	* master hanya ada 1 branch, namanya master, sedang aktif (ada * nya)
2	Buat branch bernama fitur-calculasi git branch fitur-calculasi git branch	fitur-calculasi * master
3	Pindah ke branch fitur-calculasi git checkout fitur-calculasi git branch	* fitur-calculasi master
4	Buat folder baru src, kemudian didalam folder buat file baru FiturKalkulasi.java, kemudian commit file tersebut. git add . git commit -m "tambah 1 fitur kalkulasi untuk tes compare branch"	1 files changed, 10 insertions(+), 0 deletions(-) create mode 100644 src/FiturKalkulasi.java
5	Push branch local baru kita ke server git push origin fitur-calculasi	Counting objects: 35, done. Delta compression using up to 4 threads. Compressing objects: 100% (24/24), done. Writing objects: 100% (35/35), 3.49 KiB, done. Total 35 (delta 2), reused 20 (delta 1) * [new branch] fitur-calculasi -> fitur-calculasi
6	Bandingkan antara branch master dan branch fitur-calculasi. Git log master..fitur-calculasi	commit 965acad4a3fa54e414589cd030c5257348cc0fd6 Author: Jimmy Rengga <jimmy@artivisi.com> Date: Sat Jun 18 00:13:27 2011 +0700 tambah 1 fitur kalkulasi untuk tes compare branch

Kesimpulan

Dengan menggunakan branch, kita dapat memisahkan perubahan yang kita buat supaya tidak mengganggu branch utama (master).

Merge fast-forward

Penjelasan

Pada suatu ketika, kita ingin menggabungkan perubahan di branch yang satu dengan branch yang lain. Pada latihan ini, kita akan memasukkan perubahan di branch fitur-kalender ke dalam master.

Panduan

No	Aktifitas	Hasil
1	Pindah ke branch tujuan, yaitu master. git checkout master	Switched to branch 'master'
2	Merge branch yang ingin digabungkan git merge fitur-kalender	Updating 70002d2..24e0dc1 Fast-forward BelajarGit/src/belajar/git/Kalender.java 10 +++++ 1 files changed, 10 insertions(+), 0 deletions(-) create mode 100644 BelajarGit/src/belajar/git/Kalender.java

Kesimpulan

Merge fast-forward adalah merge di mana branch tujuannya belum mengalami perubahan. Sehingga yang dilakukan Git hanyalah memajukan master sehingga sama dengan fitur-kalender.

Merge rekursif

Penjelasan

Merge rekursif terjadi bila di branch tujuan sudah terjadi perubahan, sehingga fast-forward tidak bisa dilakukan.

Panduan

No	Aktifitas	Hasil
1	Pindah ke branch fitur-kalender	
2	Buat beberapa commit	
3	Ada bug, kerjakan di branch baru, yaitu bugfix, yang diambil dari posisi master. git branch bugfix master git checkout bugfix	
4	Lakukan beberapa kali commit di branch bugfix, setelah selesai merge ke master. Harusnya akan terjadi fast-forward git checkout master git merge bugfix	Updating 24e0dc1..2b91ac6 Fast-forward BelajarGit/src/belajar/git/Kalender.java 4 +++ 1 files changed, 3 insertions(+), 1 deletions(-)
5	Penambahan fitur di fitur-kalender sudah selesai, mari kita merge ke master. git merge fitur-kalender	Auto-merging BelajarGit/src/belajar/git/Kalender.java Merge made by recursive. BelajarGit/src/belajar/git/Kalender.java 6 +++++ 1 files changed, 6 insertions(+), 0 deletions(-)

Kesimpulan

Merge rekursif dilakukan bila kedua branch sudah mengalami perubahan. Bila dilihat di tampilan grafis, merge ini akan terlihat sebagai percabangan. Sedangkan merge fast-forward akan terlihat seperti garis lurus biasa.

Rebase

Penjelasan

Bila kita ingin history tetap garis lurus tanpa cabang, kita bisa melakukan rebase di topic branch supaya selaras dengan master. Ini juga dilakukan bila ada perubahan di master yang ingin kita gunakan di topic branch.

Panduan

No	Aktifitas	Hasil
1	Undo merge pada latihan sebelumnya. git checkout fitur-kalender git reset ORIG_HEAD	
2	Lakukan rebase terhadap master git rebase master	First, rewinding head to replay your work on top of it... Applying: tambah fitur selisih hari Using index info to reconstruct a base tree... <stdin>:21: trailing whitespace. warning: 1 line adds whitespace errors. Falling back to patching base and 3-way merge... Auto-merging BelajarGit/src/belajar/git/Kalender.java
3	Lihat history git log --pretty=oneline --abbrev-commit Terlihat bahwa commit tambah fitur selisih hari pindah ke atas validasi invalid input	af6bc92 tambah fitur selisih hari 2b91ac6 tambahkan validasi untuk mencegah invalid input 24e0dc1 implement hitung hari 0d7e727 template class kalender, belum diimplementasikan 70002d2 Undo perubahan "ganti jadi bahasa indonesia" e2e84fa ganti jadi bahasa indonesia d6a8455 tambahkan fitur pembagian 8a4fbcf implement fitur tambah, kurang, kali 07a38f2 tambah satu file, fitur kalkulator c70c58c commit pertama
4	Setelah rebase, branch fitur-kalender dapat dimerge secara fast-forward ke master git checkout master git merge fitur-kalender	Updating 2b91ac6..af6bc92 Fast-forward BelajarGit/src/belajar/git/Kalender.java 6 ++++++ 1 files changed, 6 insertions(+), 0 deletions(-)

Kesimpulan

Dengan rebase, kita bisa mengakomodasi perubahan yang terjadi di master. Perhatikan bahwa ini mengubah history commit, sehingga tidak boleh dilakukan di branch yang sudah dipublish.

Merge Conflict

Penjelasan

Merge conflict terjadi apabila ada perubahan di lokasi yang sama di dua branch yang berbeda. Pada saat dilakukan merge, git tidak tahu perubahan mana yang akan diambil, sehingga diserahkan pada user untuk memilih.

Panduan

No	Aktifitas	Hasil
1	Lakukan perubahan di branch fitur-kalender git checkout fitur-kalender git commit	
2	Lakukan perubahan di file yang sama, baris yang sama di branch master git checkout master git commit	
3	Merge fitur-kalender ke master, akan terjadi conflict git merge fitur-kalender	Auto-merging BelajarGit/src/belajar/git/Kalender.java CONFLICT (content): Merge conflict in BelajarGit/src/belajar/git/Kalender.java Automatic merge failed; fix conflicts and then commit the result.
4	Status conflict bisa dilihat dari perintah status git status	# On branch master # Unmerged paths: # (use "git add/rm <file>..." as appropriate to mark resolution) # # both modified: BelajarGit/src/belajar/git/Kalender.java # no changes added to commit (use "git add" and/or "git commit -a")
5	Buka file yang conflict, kita akan melihat conflict marker. Tanda <<<< menunjukkan awal area konflik Tanda >>>> menunjukkan akhir area konflik. Tanda ===== menunjukkan pembatas antara versi di HEAD (dalam hal ini master) dan versi di branch fitur-kalender.	<<<<<< HEAD if(awal == null akhir == null) throw new IllegalArgumentException("Input tidak boleh null"); return 0; ===== throw new UnsupportedOperationException("Belum diimplement"); >>>>>> fitur-kalender
6	Selesaikan konfliknya dan hapus markernya. Kita dapat menggunakan merge tool yang grafis seperti meld, kdiff, atau TortoiseMerge. Setelah selesai mengedit, add ke staging area dan commit. git add Kalender.java git commit	[master c582408] resolve conflict, langsung throw aja toh belum diimplement

Kesimpulan

Merge conflict adalah suatu hal yang wajar dan tidak perlu dikhawatirkan. Cara resolvenya juga tidak sulit. Cukup berkomunikasi antar programmer untuk menentukan versi mana yang ingin digunakan.

Menghindari Merge Fast-forward

Penjelasan

Biasanya kebijakan perusahaan menghendaki setiap fitur dikembangkan di topic branch dan dimerge ke master pada saat sudah matang dan siap rilis. Apabila master belum ada perubahan sejak branch diawali, maka merge ini akan menjadi fast-forward. Ini akan menyulitkan penentuan ujung awal implementasi fitur baru, sehingga kalau ada kebutuhan untuk undo akan sulit melacak commit mana saja yang ingin dibatalkan.

Untuk menghindari hal ini, kita bisa melarang Git melakukan fast forward.

Panduan

No	Aktifitas	Hasil
1	Tambahkan beberapa commit di branch fitur-kalender. Pastikan posisi master tidak bergeser. git checkout fitur-kalender git commit git commit	
2	Merge fitur-kalender ke master git checkout master git merge --no-ff fitur-kalender	Merge made by recursive. BelajarGit/src/belajar/git/Kalender.java 17 +++++ 1 files changed, 17 insertions(+), 0 deletions(-)
3	Lihat historynya git log --pretty=oneline --graph --abbrev-commit	* ac24e69 Merge branch 'fitur-kalender' \n * 4341221 tambah fitur pengurangan tanggal * 2ae738b tambah fitur penjumlahan tanggal * f5457a9 add handling untuk bulan februari * c582408 resolve conflict, langsung throw aja toh belum diimplement

Kesimpulan

Dengan menggunakan opsi --no-ff merge dari topic branch menjadi lebih mudah dibatalkan bila diinginkan.

Membatalkan Merge

Penjelasan

Adakalanya kita ingin membatalkan merge. Kita bisa mereset posisi HEAD ke posisi terakhir sebelum dilakukan merge.

Panduan

No	Aktifitas	Hasil
1	<p>Pindah dulu ke branch tujuan merge. Misalnya kita merge fitur-kalender ke master, maka kita pindah ke master</p> <pre>git checkout master git reset ORIG_HEAD</pre> <p>Ini akan menghasilkan perubahan di branch fitur-kalender yang tidak jadi dimasukkan ke master.</p>	<p>Unstaged changes after reset: M</p> <p>BelajarGit/src/belajar/git/Kalender.java</p>
2	<p>Hapus saja perubahan tersebut, karena tidak jadi kita gunakan.</p> <pre>git reset --hard</pre>	<p>HEAD is now at c582408 resolve conflict, langsung throw aja toh belum diimplement</p>

Kesimpulan

Seperti halnya commit, merge juga bisa direset. Kita bisa gunakan variabel ORIG_HEAD untuk mengacu pada posisi HEAD sebelum terjadinya merge.

Menghapus Branch

Penjelasan

Branch yang sudah tidak digunakan bisa dihapus supaya tidak membingungkan.

Panduan

No	Aktifitas	Hasil
1	Hapus branch bugfix git branch -d bugfix	Deleted branch bugfix (was 2b91ac6)

Kesimpulan

Branch yang sudah dimerge bisa dihapus. Sedangkan branch yang belum dimerge akan menimbulkan warning, karena menghapus branch tersebut bisa menghilangkan historynya. Gunakan opsi -D untuk memaksa hapus branch yang belum dimerge.