

Git

Bangsa Cerdas Institute Batch 1 : January 26, 2017



Trenggana Natadirja

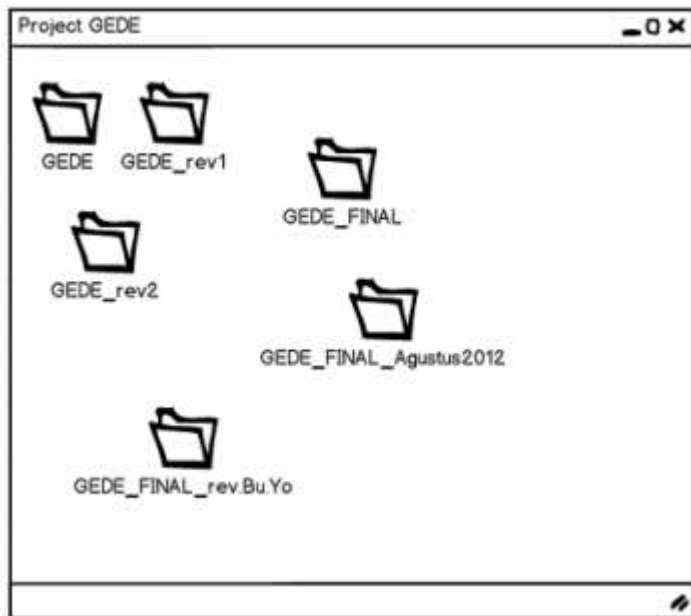
<https://facebook.com/garenk02>

@trenggana_n

Bahasan Materi

- Intro
- Instalasi
- Perintah-perintah Dasar
- Shared Repository
- Hosting

Intro



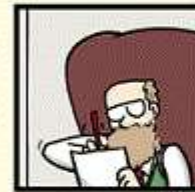
"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



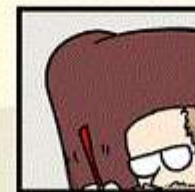
FINAL_rev.6.COMMENTS.doc




FINAL_rev.8.comments5.
CORRECTIONS.doc

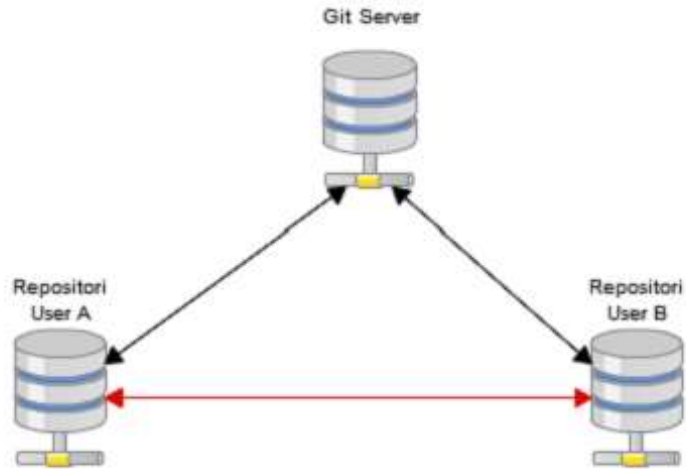


FINAL_rev.18.comments7.
corrections9.MORE.30.doc

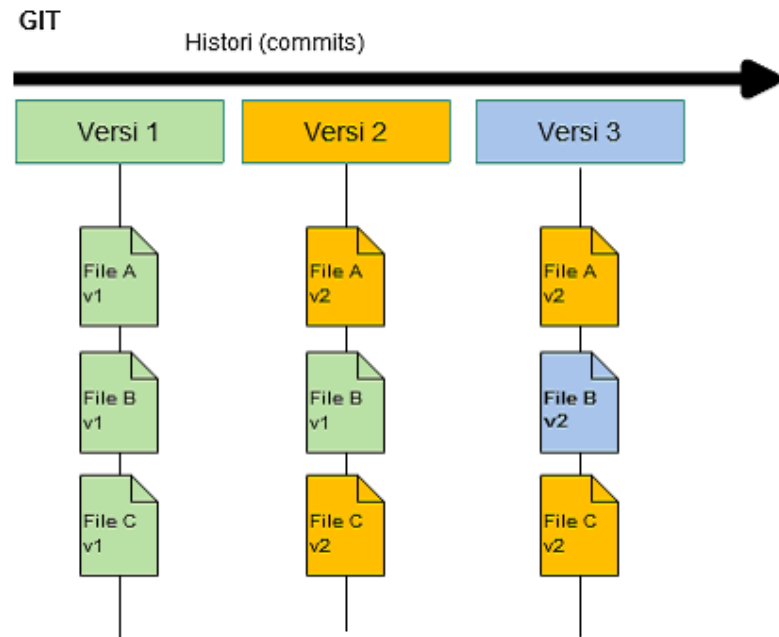


FINAL_rev.22.comments49.
corrections.10. #@\$%WHYDID
ICOMETOGRADSCOOL?????.doc

- *Version Control* atau *Revision Control* adalah sebuah sistem yang merekam perubahan (revisi) yang terjadi dari waktu ke waktu pada sebuah atau sekumpulan file sehingga dapat kembali ke revisi tertentu dengan mudah
 - Tracking Changes
 - Sandboxing
 - Kolaborasi
 - Distributed VCS
- 
- A blue L-shaped graphic is located in the bottom left corner of the slide.



Gambar 1-4 Repositori Git



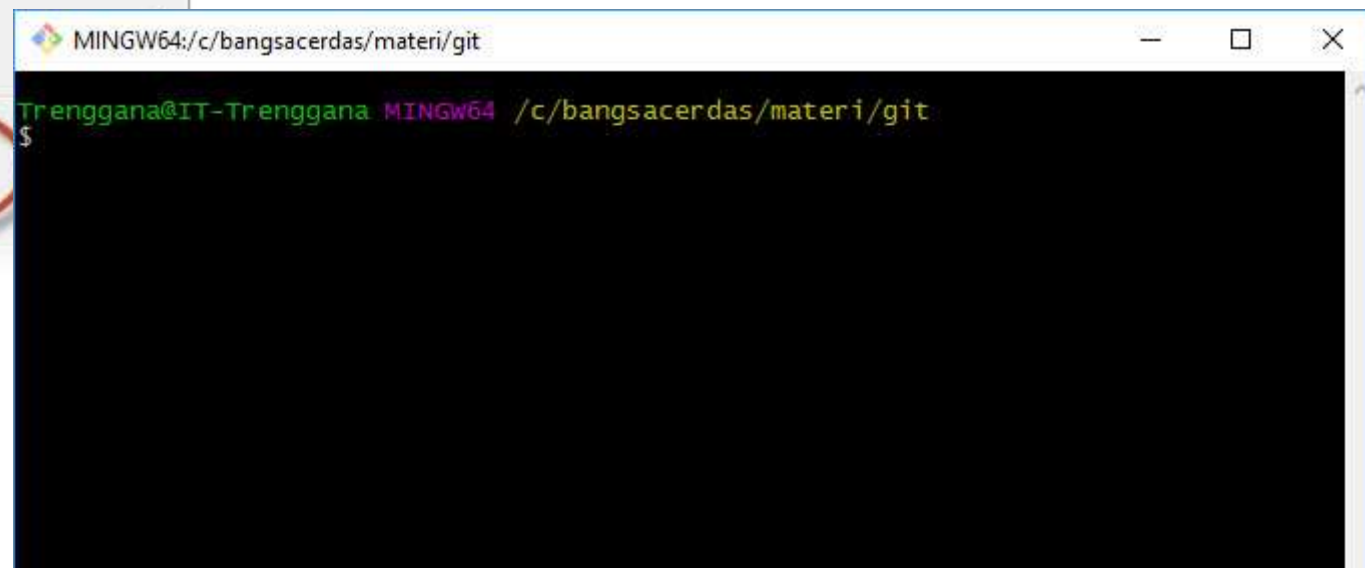
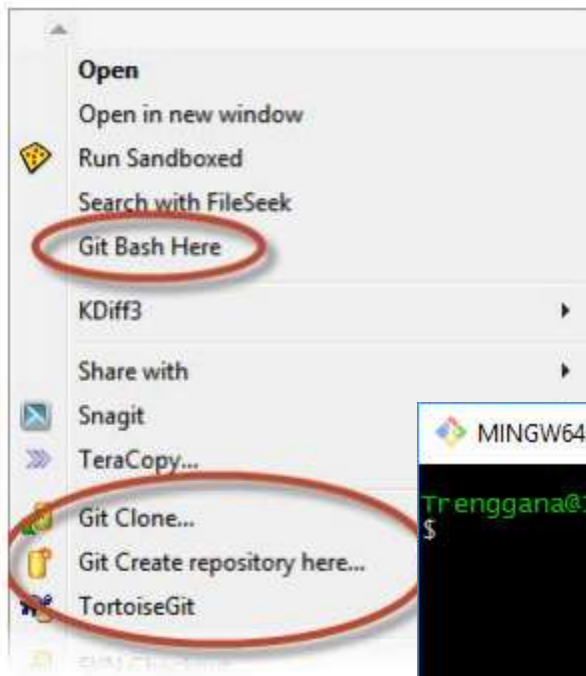
Gambar 1-6 Cara Git Menyimpan Revisi

- Working directory
- Repository
- Commit
- Staging Index
- Staged File
- Unstaged File
- Branch

- Merge
- Untracked File
- Tracked File
- HEAD
- Push
- Fetch
- Pull

Instalasi

- <https://git-scm.com>
- TortoiseGit
- SourceTree



Konfigurasi

- Global

```
git config --global user.name "Nama Anda"
```

```
git config --global user.email  
"email@domain.com"
```

- Lokal

```
git config user.name "Nama Anda"
```

```
git config user.email "email@domain.com"
```

Default Editor

Notepad (Windows)

```
$ git config --global core.editor "notepad"
```

Notepad++ (Windows)

```
$ git config --global core.editor "'C:/Program  
Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession  
-noPlugin"
```

Textmate (Mac)

```
$ git config --global core.editor "mate -w"
```

Nano (Linux)

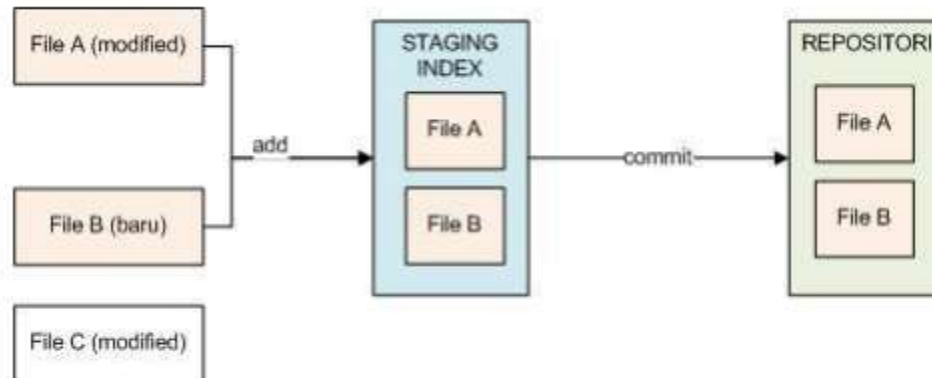
```
$ git config --global core.editor "nano"
```

Perintah-perintah Dasar

•Membuat Repository

➤git init

•Commit



Gambar 3-1 Git Staging & Commit

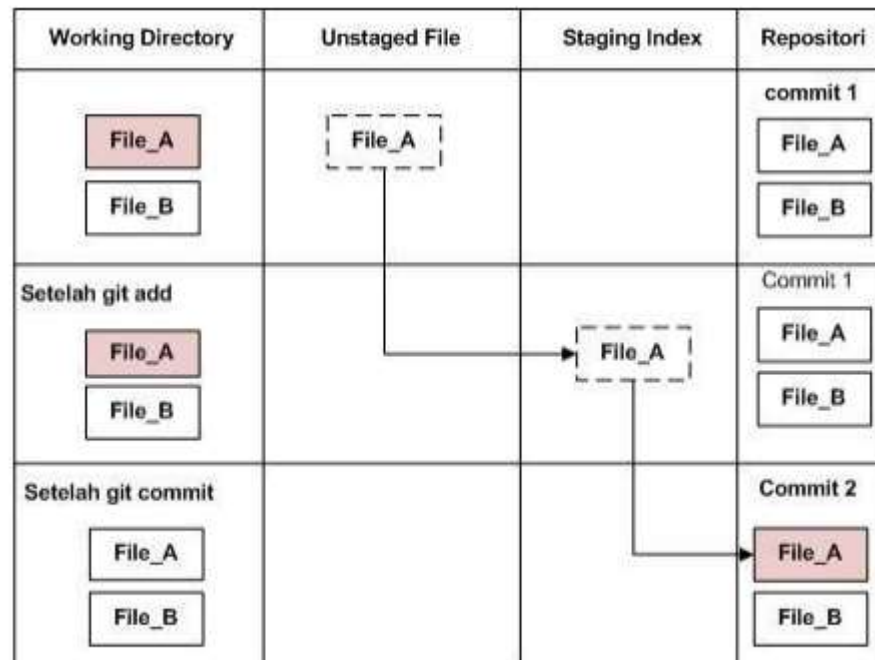
```
// buat direktori
$ mkdir staging-demo
$ cd staging-demo

// inisialisasi repositori
$ git init

// buat file
$ touch File_A File_B

// tambahkan file ke staging index
$ git add .

// initial commit
$ git commit -m "Initial commit"
```





•Membatalkan staging

```
// isi File_A dengan teks hello world
$ echo 'hello world' > File_A

// lihat status
$ git status
# On branch master
# Changes not staged for commit:
#   modified:   File_A

// tambahkan file ke staging index
$ git add File_A

// lihat status working directory
$ git status
# On branch master
# Changes to be committed:
#   modified:   File_A

// soft-reset
$ git reset HEAD
Unstaged changes after reset:
    M       File_A

// File_A dikeluarkan dari staging index,
// tapi tetap dalam kondisi modified
$ git status
# On branch master
# Changes not staged for commit:
#   modified:   File_A
```



- Membatalkan commit

- git commit --amend

- Log

- git log

```
$ git log
commit 35604b7127c9b6319d58244cb18b0fd90e60573a
Author: Rengga <rengga@staff.gramedia.com>
Date: Sat Jan 21 20:43:25 2017 +0700

    Penambahan judul pada bagian 1

commit 4d50b2525c3350b12576eda782d2ef66092a4bb2
Author: Rengga <rengga@staff.gramedia.com>
Date: Sat Jan 21 20:28:19 2017 +0700

    Perbaikan pada langkah 4

commit 4b8f67032daaa8b3a24fba684bafc41f730d16ea
Author: Rengga <rengga@staff.gramedia.com>
Date: Sat Jan 21 20:19:22 2017 +0700

    Penambahan resep polijus
```

- git log --oneline

```
$ git log --oneline
35604b7 Penambahan judul pada bagian 1
4d50b25 Perbaikan pada langkah 4
4b8f670 Penambahan resep polijus
```

- gitk

Menangani File

- Unstaged files

```
// ubah isi fileA
$ echo 'hello' > fileA

// lihat modifikasi yang kita lakukan
// terhadap fileA
$ git diff fileA
diff --git a/fileA b/fileA
index e69de29..ce01362 100644
--- a/fileA
+++ b/fileA
@@ -0,0 +1 @@
+hello
```

```
// tambahkan semua file ke dalam staging index
$ git add .

// initial commit
$ git commit -m 'update fileA'

// ubah isi fileA
$ echo 'hello world' > fileA

// lihat perbedaan isi fileA
$ git diff fileA
diff --git a/fileA b/fileA
index ce01362..14be0d4 100644
--- a/fileA
+++ b/fileA
@@ -1,1 +1,1 @@
-hello // baris yang dihapus
+hello world // baris baru
```

•Staged files

```
$ git add fileA

// lihat modifikasi isi fileA yang
// sudah ada di dalam staging index
$ git diff --staged fileA

diff --git a/fileA b/fileA
index ce01362..14be0d4 100644
--- a/fileA
+++ b/fileA
@@ -1,1 @@
-hello
+hello world
```

```
$ git diff 5acdc27..053ea41 fileA
diff --git a/fileA b/fileA
index ce01362..14be0d4 100644
--- a/fileA
+++ b/fileA
@@ -1,1 @@
-hello
+hello world
```



- Membandingkan dua file

```
$ git diff

diff --git a/fileA b/fileA
index 3b18e51..3432579 100644
--- a/fileA
+++ b/fileA
@@ -1,1 @@
-hello world
+ini isi fileA

diff --git a/fileB b/fileB
index e69de29..98d3e2f 100644
--- a/fileB
+++ b/fileB
@@ -0,0 +1 @@
+ini isi fileB
```

Membatalkan Modifikasi

- Hard Reset

- git checkout -- File_A

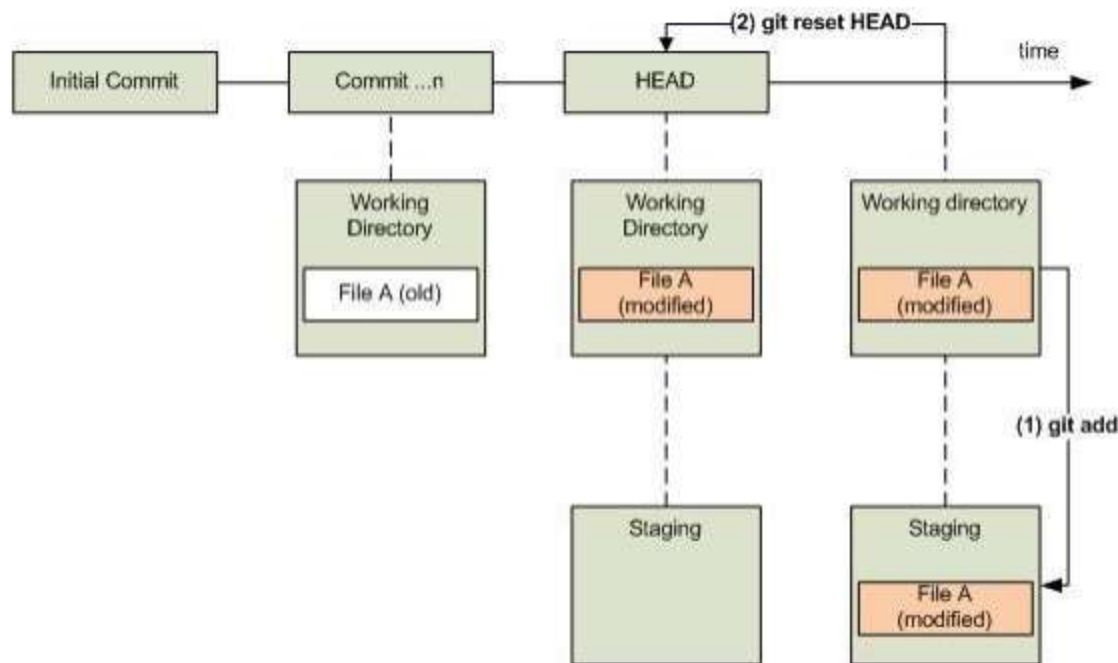
- Soft Reset

- git reset HEAD File_A

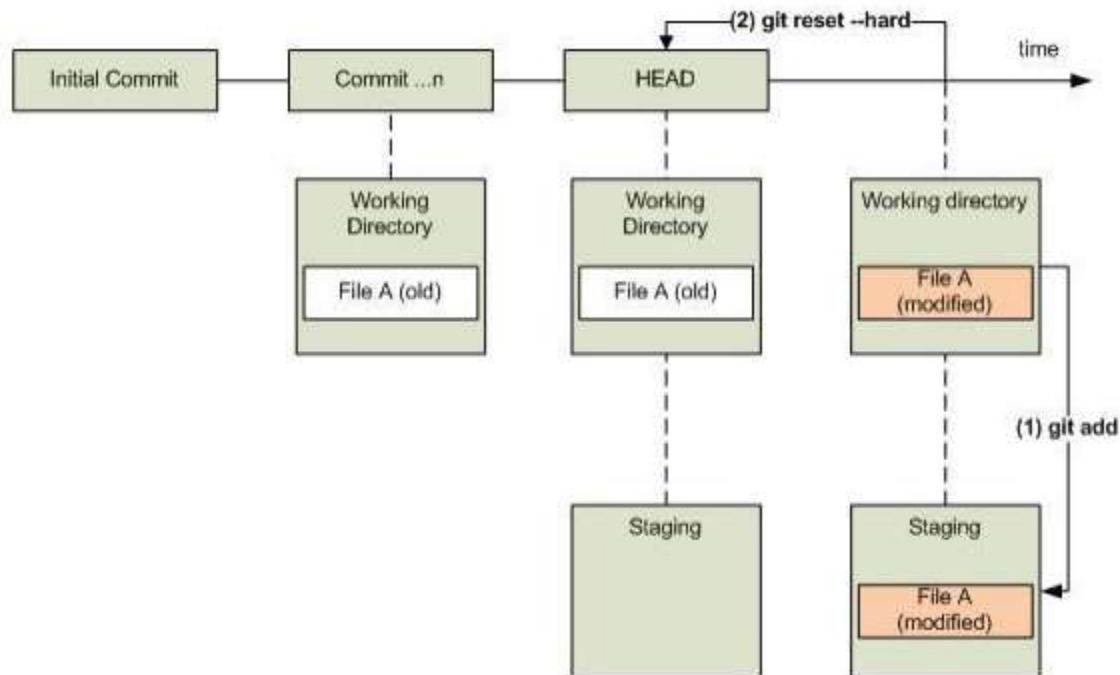
- git checkout -- File_A

- Hard Reset Working copy

- git reset --hard



Gambar 3-6 Soft Reset



Gambar 3-7 Hard Reset

•Berpindah Versi

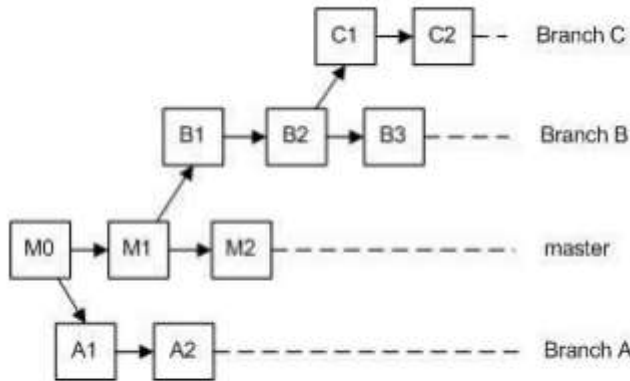
➤git reset <id-commit>

➤git log --oneline

➤git reset <id-commit>

➤git log --oneline

Branching & Merging



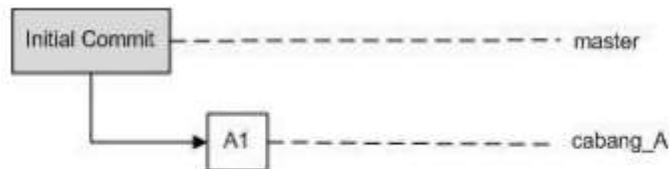
Gambar 3-8 Git Branch

- Membuat Branch
 - `git branch <nama branch>`
- Melihat Branch
 - `git branch`

- Berpindah Branch
 - `git checkout <nama branch>`
- Membuat branch baru dan langsung berpindah
 - `git checkout -b <nama branch>`
- Membuat child branch dari branch manapun
 - `git checkout parent-branch -b branch-baru`

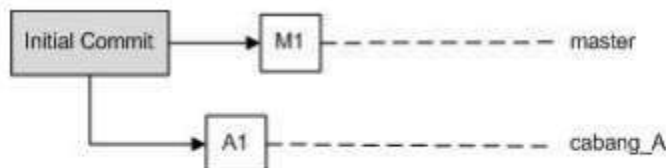
Berpindah Branch

```
$ git checkout cabang_A
$ touch fileB.txt
$ git add .
$ git commit -m 'A1'
```



Gambar 3-9 Cabang Baru

```
$ git checkout master
$ touch fileC.txt
$ git add fileC.txt
$ git commit -m "M1"
```



Gambar 3-10 Diverging Branch

```
// ubah isi fileA.txt
$ echo 'update file A di master' > fileA.txt

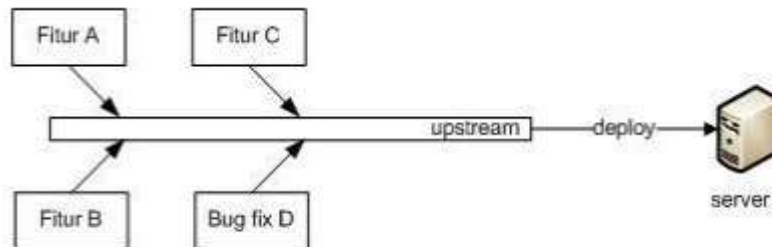
// cek status
$ git status
# On branch master
# Changes not staged for commit:
#   modified:   fileA.txt
#
no changes added to commit

// buat cabang baru dan checkout
$ git checkout -b cabang_B

// cek status
$ git status
# On branch cabang_B
# Changes not staged for commit:
#   modified:   fileA.txt
#
no changes added to commit
```

Upstream Branch

- Kita perlu menentukan *upstream branch* yaitu branch dari mana file-file tersebut berasal



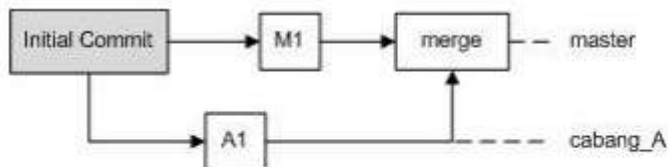
Gambar 3-12 Upstream Branch & Deployment

- Untuk melakukan sinkronisasi dengan branch master
 - git merge master

Merging

```
// pindah ke master
$ git checkout master

// gabungkan cabang_A ke dalam master
$ git merge cabang_A
Merge made by the 'recursive' strategy.
0 files changed
create mode 100644 fileB.txt
```



Gambar 3-13 Merge cabang A dengan master

•Merge Conflict

```
// pindah ke cabang_A
$ git checkout cabang_A
// ubah isi fileB.txt
$ echo "hello cabang A" > fileB.txt
$ git add .
$ git commit -m "A2"
```

```
// pindah ke master
$ git checkout master
// ubah isi fileB.txt
$ echo "hello master" > fileB.txt
$ git add .
$ git commit -m "M1"

// merge cabang_A ke dalam master
$ git merge cabang_A
Auto-merging fileB.txt
CONFLICT (content): Merge conflict in fileB.txt
Automatic merge failed; fix conflicts and then commit the
result.
```

```
$ git diff
diff --cc fileB.txt
index 22a5dfd,093991f..0000000
--- a/fileB.txt
+++ b/fileB.txt
@@@ -1,1 -1,1 +1,5 @@@
++<<<<<< HEAD
+hello master
++=====
+ hello cabang A
++>>>>>> cabang_A
```

Git Stash

1. Menyimpan modifikasi yang belum kita **commit** ke dalam sebuah tempat penyimpanan sementara (**stash**)
2. Melakukan **hard-reset** terhadap **working copy** sehingga kita bisa berpindah **branch**

```
$ mkdir stash-demo
$ cd stash-demo
// inisialisasi repositori
$ git init
$ touch fileA
$ git add .
$ git commit -m 'initial commit'

// buat branch baru
$ git branch dev
```

```
// ubah isi fileA
$ echo 'teks ini dari master' > fileA
// commit master
$ git commit -a -m 'M1'
```

```
// pindah ke cabang dev
$ git checkout dev
// ubah isi fileA
$ echo 'Teks ini dari dev'> fileA
```

```
$ git checkout master
error: Your local changes to the following files would be
overwritten by checkout:
    fileA
Please, commit your changes or stash them before you can
switch branches.
Aborting
```

```
$ git stash
Saved working directory and index state WIP on dev: f46e549
initial commit
HEAD is now at f46e549 initial commit
```

```
$ git checkout master
$ echo 'teks ini dari master v2' > fileA
$ git commit -a -m 'M2'
```

```
$ git checkout dev
$ git stash apply
```


Tagging

- Sebagai penanda revisi-revisi penting seperti upload/deployment ke server, final release, beta release

```
$ mkdir tag-demo  
$ cd tag-demo  
$ git init  
$ touch fileA  
$ git add .  
$ git commit -m 'initial commit'
```

```
$ echo 'Judul : Tag demo' > fileA  
$ git commit -a -m 'tambah judul'  
$ echo 'Deskripsi : Ini file untuk demo tagging' >> fileA  
$ git commit -a -m 'tambah deskripsi'
```

```
$ git tag 'v1'
```

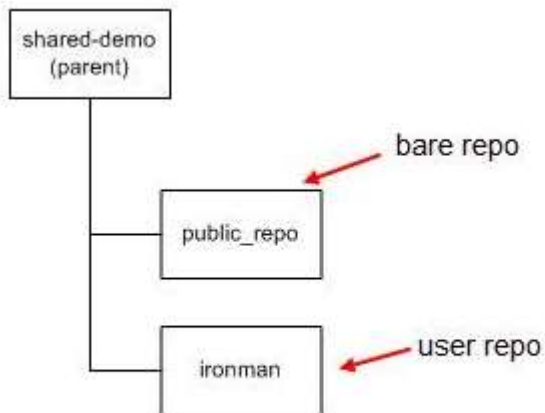
Shared Repository

Bare Repository

- Sebuah repositori yang tidak memiliki *working directory*

```
$ mkdir public_repo
$ cd public_repo/

//inisialisasi bare repo
$ git init --bare
Initialized empty Git repository in
/Users/boss/Desktop/belajargit/shared-demo/public_repo/
```



Gambar 4-2 Struktur direktori shared-demo

```
$ git clone ../public_repo/ .
Cloning into ....
done.
warning: You appear to have cloned an empty repository.
```

```
$ git config user.name "Ironman"
$ git config user.email "tonystark@avengers.com"
```

```
$ touch readme
$ echo 'contoh shared repository' > readme
$ git add .
$ git commit -m 'initial commit'

// push = update remote repository
// origin = nama alias untuk remote repository
// master = default branch di remote repository
$ git push origin master
```

```
// kembali ke parent directory
$ cd ..
// buat directory untuk user Hulk
$ mkdir hulk
$ cd hulk
// clone public repository
$ git clone ../public_repo/ .
// konfigurasi identitas user
$ git config user.name "Hulk"
$ git config user.email "dr.banner@avengers.com"

// kembali ke parent directory
$ cd ..
// buat directory untuk user Thor
$ mkdir thor
$ cd thor
// clone public repository
$ git clone ../public_repo/ .
// konfigurasi identitas user
$ git config user.name "Thor"
$ git config user.email "thor@avengers.com"
```

•Remote Branch

```
$ git branch -a
* master                //local branch
remotes/origin/master    //remote branch
```

```
$ cd thor
// buat branch
$ git checkout -b thor_branch
// push branch ke remote repository sebagai thor branch
$ git push -u origin thor_branch
```

```
// pindah ke working directory Hulk
$ cd ../hulk/
// update repository lokal
$ git pull
From /Users/boss/Desktop/belajargit/shared-
demo/hulk/../../public_repo
* [new branch]      thor_branch -> origin/thor_branch
Already up-to-date.

// pindah ke working directory Ironman
$ cd ../ironman

// update repositori lokal
$ git pull
From /Users/boss/Desktop/belajargit/shared-
demo/ironman/../../public_repo
* [new branch]      thor_branch -> origin/thor_branch
Already up-to-date.
```

Tracking dan Non-tracking Branch

- ***Tracking branch*** adalah local branch yang memiliki referensi ke remote branch
- ***Non-tracking branch*** tidak memiliki hubungan dengan remote branch
- ***git pull*** & ***git push*** dilakukan di tracking branch

```
// buat tracking branch dengan nama thor_branch  
// yang berhubungan dengan remote branch : origin/thor_branch  
$ git branch --track thor_branch origin/thor_branch  
Branch thor_branchA set up to track remote branch thor_branch  
from origin.
```

Sinkronisasi

- ***push*** adalah proses untuk melakukan *update database remote repository*
- ***pull*** men
repository
(*merge*) c

```
// pindah ke branch thor_branch di repositori thor
$ git checkout thor_branch

// pull thor_branch dari shared repository
$ git pull
remote: Counting objects: 5, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /Users/boss/Desktop/belajargit/shared-
demo/thor/../../public_repo
    8e2blbd..64e75b4  thor_branch -> origin/thor_branch
Updating 8e2blbd..64e75b4
Fast-forward
  readme |      1 +
  1 files changed, 1 insertions(+), 0 deletions(-)

// lihat isi file readme
$ cat readme
contoh shared repository
teks ini dari ironman
```

Non Fast-forward & Merging Conflict

```
// HULK
$ echo 'teks ini dari hulk' >> readme
$ git commit -a -m 'update teks hulk'
$ git push
```

```
// THOR
$ echo 'teks ini dari thor' >> readme
$ git commit -a -m 'update teks thor'

$ git push
To /Users/boss/Desktop/belajargit/shared-
demo/thor/../../public repo/
! [rejected] thor_branch -> thor_branch (non-fast-forward)
```

```
$ git pull
Auto-merging readme
CONFLICT (content): Merge conflict in readme
Automatic merge failed; fix conflicts and then commit the re-
sult.
```

```
$ git commit -a -m 'merge readme'
```

```
$ git push
```

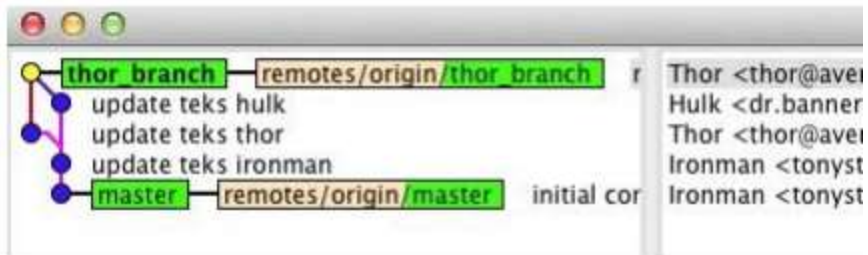


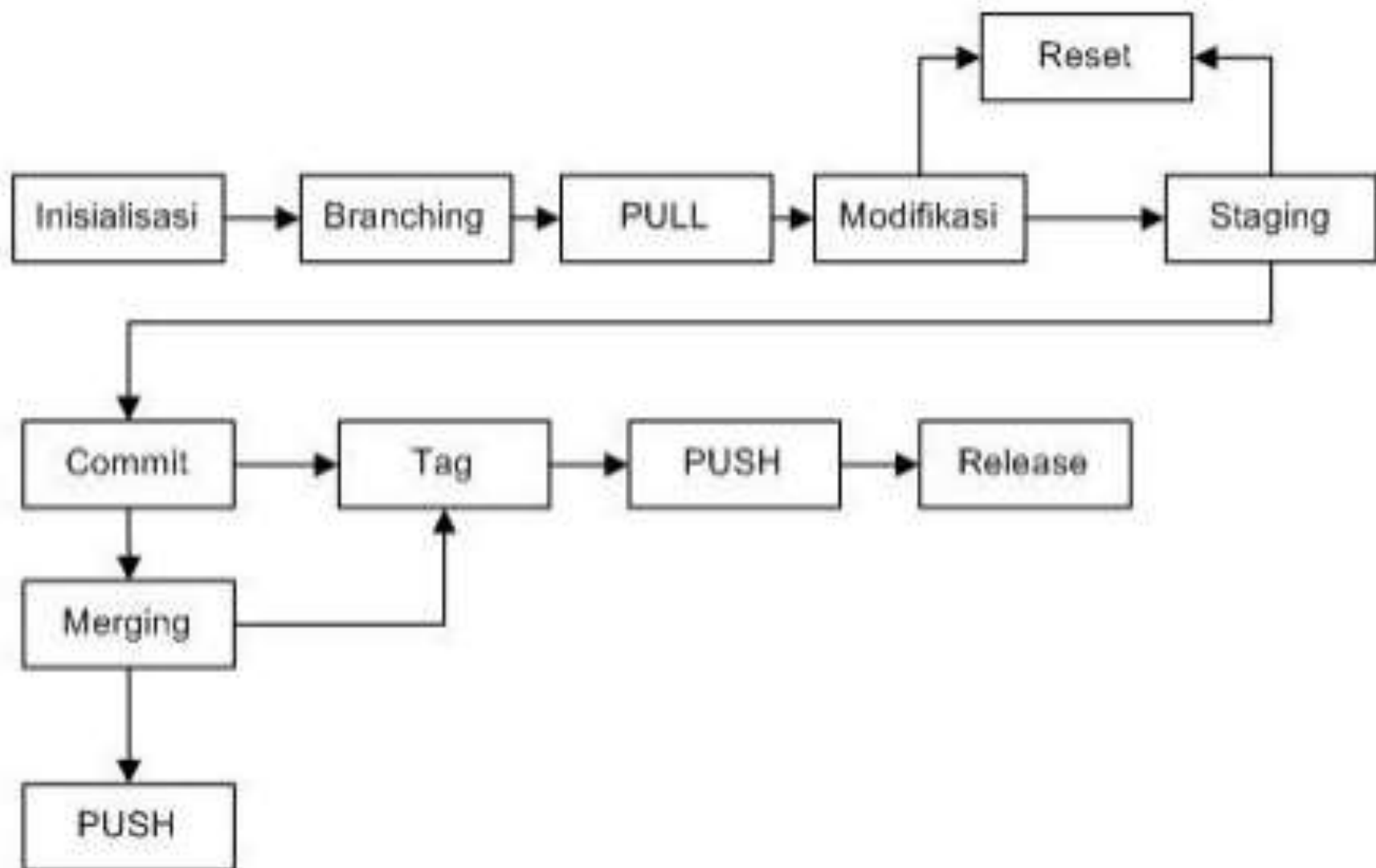
```

$ git checkout thor branch
$ git pull
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (9/9), done.
From /Users/boss/Desktop/belajargit/shared-
demo/ironman/../../public_repo
    e70f329..e58a466  thor_branch -> origin/thor_branch
Updating e70f329..e58a466
Fast-forward
  readme |      2 ++
  1 files changed, 2 insertions(+), 0 deletions(-)

// lihat isi readme
$ cat readme
contoh shared repository
ini teks dari ironman
teks ini dari thor
teks ini dari hulk

```





Gambar 4-5 Workflow untuk Kolaborasi

Git Hosting

- Github

- Bitbucket

- Gitlab

- SSH Key

 - ssh-keygen -t rsa

 - id_rsa

 - id_rsa.pub





Personal Open source Business Explore

Pricing Blog Support

Search GitHub

Sign in

Sign up

How people build software

Millions of developers use
personal projects, support
and work together on open



Download

Features

Products

Community

Explore

Docs

Blog

Contact



Sign in

Sign Up

Get paid to write for GitLab! Join the Community Writers Program!



Tools for modern developers

GitLab unifies issues, code review, CI and CD into a single UI



Features

Integrations

Server

Data Center

Pricing

Log in

Get started

Code, Manage, Collaborate

Bitbucket is the Git solution for professional teams

Get started for free

Host it yourself with Bitbucket Server

Built for professional teams

Distributed version control system that makes it easy for you to collaborate with your team.
The only collaborative Git solution that massively scales.

...or create a new repository on the command line

```
echo "# github-essentials" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:axilleas/github-essentials.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:axilleas/github-essentials.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project

[Import code](#)

Unwatch 1 Star 0 Fork 0

Notifications

Not watching

Be notified when participating or @mentioned.

✓ Watching

Be notified of all conversations.

Ignoring

Never be notified.

Issues Pull requests Labels Milestones

0 Open 0 Closed

You haven't created any Milestones

Use Milestones to create collections of issues

Release Tag

mark of latest release

Latest release

tag version → v1.5

latest commit SHA of the tag → 7780ee2

release title

axilleas released this 3 days ago

latest commit date of the tag

We are finally out of beta

The long awaited v1.5 release has finally come out! After a long time developing and bug squashing we are pleased to announce that a great number of bugs are fixed and an even greater number of features have been implemented 🎉

Bugs fixed

- Closed issue #3
- Fixed #1

description box

Downloads

Source code (zip)

Source code (tar.gz)

download links of the source code at the given tag

Github for dummies <http://githubessentials.com> — Edit

2 commits 3 branches 0 releases 1 contributor

branch: master github-essentials / +

Switch branches/tags

Post a tag

Latest commit 7780ee2c18

4 months ago

3 days ago

~ Terima Kasih ~