

# Merancang *Git Server* dengan Pendekatan *GitHub Social Coding* dalam Peningkatan Pembelajaran Mahasiswa

## SKRIPSI

Diajukan Untuk Memenuhi Sebagian Syarat Guna  
Memperoleh Gelar Sarjana Komputer (S.Kom.)  
Pada Program Studi Teknik Informatika



OLEH:

**M. SAIFUL MUKHAROM**

NPM. 09.1.03.02.0299

FAKULTAS TEKNIK  
UNIVERSITAS NUSANTARA PERSATUAN GURU REPUBLIK INDONESIA  
UNP KEDIRI  
2015

Skripsi oleh :

**M. SAIFUL MUKHAROM**  
NPM. 09.1. 03.02.0299

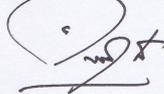
Judul :

**Merancang *Git Server* dengan Pendekatan  
*GitHub Social Coding* dalam Peningkatan  
Pembelajaran Mahasiswa**

Telah disetujui untuk dilanjutkan Kepada  
Panitia Ujian/Sidang Skripsi Program Studi Teknik Informatika  
Fakultas Teknik UNP Kediri

Tanggal : 23 Mei 2015

Pembimbing I



**Dr. M. Muchson, SE., M.M**  
NIDN. 0018126701

Pembimbing II



**Ari Eka Prasetyanto, S.Kom**  
NIDN. 0729048403

Skripsi oleh:

**M. SAIFUL MUKHAROM**  
NPM. 09.1.03.02.0299

Judul:

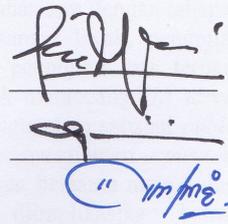
**Merancang *Git Server* dengan Pendekatan  
*GitHub Social Coding* dalam Peningkatan  
Pembelajaran Mahasiswa**

Telah dipertahankan di depan Panitia Ujian/Sidang Skripsi  
Program Studi Teknik Informatika  
Fakultas Teknik UNP Kediri  
Pada Tanggal: 23 Mei 2015

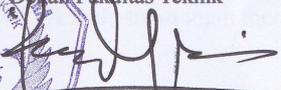
**Dan Dinyatakan telah Memenuhi Persyaratan**

Panitia Penguji :

1. Ketua : Rini Indriati, S.Kom., M.Kom.
2. Penguji I : Nursalim, S.Pd., MH.
3. Penguji II : Patmi Kasih, M.Kom.



Mengetahui,  
Dekan Fakultas Teknik

  
Rini Indriati, S.Kom., M.Kom.  
NIM. 1081001070

# Merancang *Git Server* dengan Pendekatan *GitHub Social Coding* dalam Peningkatan Pembelajaran Mahasiswa

M. Saiful Mukharom

Fakultas Teknik - Program Studi Teknik Informatika  
ritnesaif@gmail.com

Dr. M. Muchson, SE., M.M dan Ari Eka Prasetya, S.Kom  
UNIVERSITAS NUSANTARA PGRI KEDIRI

October 13, 2015

## Abstrak

Selama ini *Git Server* hanya dimanfaatkan sebagai pengembangan aplikasi, belum banyak yang menerapkan sebagai media pembelajaran, sehingga banyak ditemukan media belajar mahasiswa yang masih personal(individu). *Git Server* adalah layanan *Source Code Management* dalam jaringan yang terdistribusi. *Git Server* dengan pendekatan *GitHub Social Coding* dirancang untuk meningkatkan pembelajaran mahasiswa dalam mata kuliah pemrograman.

*Git Server* dirancang dengan layanan *Domain Name System (DNS) server*, *Dynamic Host Configuration Protocol (DHCP) server*, akses protokol SSH dan HTTP, *gitolite* sebagai manajemen *repository* dan *users*, serta *git* sebagai layanan *bare repository*, rancangan layanan tersebut untuk mendapatkan *url* SSH:git@git.gitserver:path/to/repo\_name.git dan *url* HTTP:http://www.gitserver/. Alamat *url* tersebut sebagai akses kolaborasi *repository* yang *read-write* maupun *read-only*. Setelah diujikan kepada mahasiswa dapat disimpulkan bahwa merancang *Git Server* dengan pendekatan *GitHub Social Coding* harus dapat digunakan sebagai *public repository* maupun *privat repository*, kemudian dapat memajemen *repository* dan *users*, serta dapat digunakan sebagai kolaborasi belajar bersama mata kuliah pemrograman yang terdistribusi.

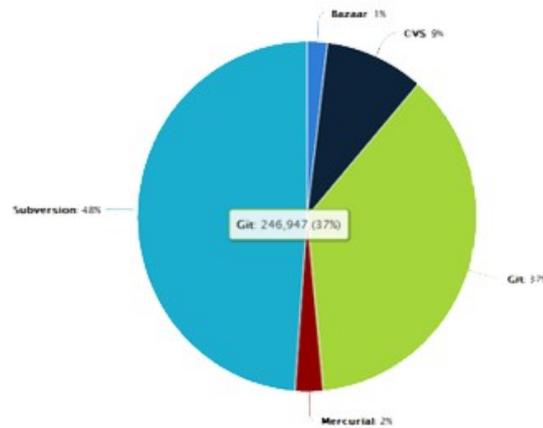
## 1 Pendahuluan

### 1.1 Latar Belakang Masalah

Didalam proses kegiatan Komunias CAH UNP (KCU) mahasiswa yang tergabung dalam komunitas sedikit sekali yang aktif, padahal berkominutas telah dimodelkan oleh Wahono,

R.S.(2007:15) mengatakan bahwa model motivasi komunitas cukup efektif digunakan untuk pengembangan dan penerapan sistem pembelajaran elektronik.

Dalam pengembangan *software*, *Git* adalah sebuah *Distributed Version Control* (DVC) dan *Source Code Management* (SCM) dengan penekanan pada kecepatan (Torvalds, 2005: list.linux-kernel). Lima tahun terakhir ini *Git* sangat populer, dipopulerkan oleh *GitHub*. Statistik yang diambil dari [ohloh.net](http://ohloh.net) *Git* menunjukkan perkembangan sangat cepat:



Gambar 1: Perbandingan *Repository*.

Sejalan dengan pendekatan konstruktivisme dalam pembelajaran, salah satu model pembelajaran yang kini banyak mendapat respon adalah pembelajaran kooperatif Sumarno (2011:homepage). *Git Server* dapat dimanfaatkan seperti, mengerjakan tugas kelompok mata kuliah pemrograman, mengembangkan aplikasi secara bersama-sama, berbagi *source code* dengan cepat dan mudah, serta siapapun dapat melihat dan men-*download source code* yang diunggah ke *Git Server*.

## 1.2 Identifikasi Masalah

Perlunya sebuah rancangan *Git Server* dengan pendekatan *GitHub Social Coding* yang dapat digunakan sebagai media belajar bersama mata kuliah pemrograman sehingga dapat meningkatkan pembelajaran mahasiswa.

## 1.3 Rumusan Masalah

1. Bagaimana merancang *Git Server* dengan pendekatan *GitHub Social Coding* dalam peningkatan pembelajaran mahasiswa.
2. Apakah rancangan *Git Server* dengan pendekatan *GitHub Social Coding* dapat meningkatkan pembelajaran mahasiswa.



## 1.4 Tujuan Penelitian

1. Untuk merancang *Git Server* dengan pendekatan *GitHub Social Coding* dalam peningkatan pembelajaran mahasiswa.
2. Untuk mengetahui rancangan *Git Server* dengan pendekatan *GitHub Social Coding* dapat meningkatkan pembelajaran mahasiswa pada mata kuliah pemrograman.

## 2 Landasan Teori

### 2.1 Ikhtisar

*Version Control* adalah sebuah sistem yang mencatat setiap perubahan terhadap sebuah berkas atau kumpulan berkas sehingga pada suatu saat anda dapat kembali pada salah satu versi dari bekas tersebut (Chacon, 2009:1). Sedangkan *Git* itu sendiri merupakan sekian dari beberapa *version control* yang ada.

Menurut Gadjaja (2013:1-6) untuk menginstal dan memulai *Git* seperti berikut:

#### 2.1.1 Instal *Git*

##### Instal *Git* di *Windows*

Pergi ke alamat [msysgit.github.io](https://msysgit.github.io), kemudian unduh versi *installer* terbaru untuk *windows*, jalankan berkas *installer* dan abaikan semua pilihan, atur ke nilai *default*. Setelah itu *Git* siap dijalankan pada sistem anda. menginstal *Git* untuk *windows* dapat menggunakan paket *cygwin* yang tersedia di [cygwin.com](https://cygwin.com).

##### Instal *Git* di *Linux*

Tergantung *system* yang anda jalankan, lakukan perintah :

```
# untuk ubuntu
$ sudo apt-get install git
# untuk fedora
$ sudo yum install git
```

#### 2.1.2 Memulai *Git*

Untuk mengkonfigurasi *Git* agar dapat bekerja dan berjalan, jalankan perintah:

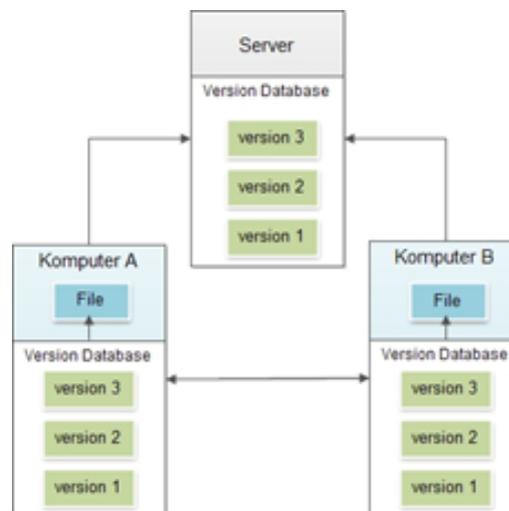
```
$ git config --global user.name "Nama Lengkap"
$ git config --global user.email "email@example.com"
```

Perintah tersebut harus dikonfigurasi karena pengaturan *user.name* dan *user.email* akan disimpan dalam setiap *commit* yang anda buat.

## 2.2 *Git Server*

*Git* adalah sistem manajemen versi dan kode program, *Git* didistribusikan dan mendukung percabangan yang rumit, *Git* dirancang untuk proyek-proyek *open source* oleh Linus Torvalds (Brazdil, 2013:5), Sedangkan *Server* adalah sebuah komputer yang menjadi pusat kegiatan suatu jaringan yang dapat memproses satu atau lebih layanan jaringan (M. Doss, 1999:1), Jadi *Git Server* dapat diartikan sebagai penyedia layanan manajemen kode program yang terdistribusi dalam jaringan.

Agar *Git* dapat melakukan *version control* secara terdistribusi dan dapat menangani sejumlah *remote repository*, maka perlu adanya *Git Server* untuk menyediakan layanan tersebut. seperti diperlihatkan pada gambar berikut:



Gambar 2: Diagram *Distributed Version Control*(Chacon, 2009:3)

## 2.3 *GitHub Social Coding*

*GitHub* adalah layanan untuk *hosting git repository*. *GitHub* memiliki web kode peramban yang memungkinkan mengomentari kode. *GitHub* bukan *open source* tetapi proyek-proyek *open source* dapat di-*hosting* disana dengan gratis. Untuk proyek-proyek lain, *GitHub* menyediakan *privat repository* prabayar dan menawarkan *GitHub Enterprise* untuk instalasi rumahan. (Brazdil, 2013:10). *GitHub* sangat populer memiliki lebih dari tiga juta pengguna (Sanheim, 2013:homepage).

Aktifitas *Social Coding* yang dimaksud adalah dengan konsep *whatching*: dengan *whatching* pengguna lain akan tertarik untuk berkontribusi, *forking repository*: untuk mengizinkan siapapun menyalin atau menggandakan proyek, *pull request*: sebagai sinyal bagi pemilik utama proyek yang mana programmer lain memiliki perubahan kode yang menarik untuk berpotensi bergabung didalamnya, dan *line-level comments* pada commit tertentu untuk diijinkan perbaikan ulang sederhana untuk turut serta berkontribusi (Loeliger, 2012:390).

Jadi *GitHub Social Coding* dapat diartikan sebagai *hosting git repository* yang memiliki layanan *whatching, forking, pull request, dan line-level comments*, sehingga memungkinkan pengguna dapat melakukan aktivitas *social coding* dalam proyek *open source* maupun *close source*. Langkah-langkah mengoperasikan *GitHub*, anda dapat mengakses alamat [help.github.com](http://help.github.com).

## 2.4 Alat *Git Server*

### 2.4.1 Kebutuhan *Hardware*

Kebutuhan *hardware* untuk dapat diinstal sistem operasi *linux* Ubuntu Server 12.04.1 LTS, Menurut Wiliamson(2012:4) seperti berikut:

Tabel 1: Persyaratan Minimum yang Direkomendasikan

CPU	RAM	Hard Drive Space	
		Base System	All Task Installed
Server Standar			
1GHz	512MB	1GB	1,75GB
Server Minimal			
300MHz	256MB	700MB	1,4GB

### 2.4.2 Kebutuhan *Software*

*Software* yang diinstal sebagai kebutuhan merancang *Git Server*, menurut Loeliger (2012:10) yaitu:

Tabel 2: *Packet Installed*

Sistem Operasi	Packet Installed
Linux Ubuntu Server 12.04.01 LTS	Git, gitweb, git-daemon-run, gitolite, openssh-server, bind9, isc-dhcp-server, apache2

### 2.4.3 Mengoperasikan *Git*

Untuk dapat mengoperasikan *Git* menggunakan *command-line* dibutuhkan keahlian operasi *file* dan *directory* berbasis *unix* dan *linux*. Sedangkan untuk perintah dasar *git* anda dapat mengetikkan perintah `git --help` atau `git <command> --help`.



## 2.5 Peningkatan Pembelajaran Mahasiswa

Mahasiswa merupakan insan dewasa yang memiliki kesadaran sendiri dalam mengembangkan potensi diri di perguruan tinggi untuk menjadi intelektual, ilmuan, praktisi, dan/atau profesional (UU No.12,2012:Tentang Pendidikan Tinggi).

Menurut Suskie(2002:1) strategi meningkatkan pembelajaran mahasiswa ada sepuluh strategi, tiga diantaranya yaitu:

1. Mereka menghabiskan lebih banyak waktu aktif terlibat dalam pembelajaran dan sedikit waktu untuk mendengarkan kuliah.
2. Mereka berpartisipasi dalam kegiatan diluar kelas, seperti kegiatan ko-kurikuler.
3. Mereka merefleksikan apa dan bagaimana mereka telah belajar.

Meskipun belum bisa dikatakan sempurna berdasarkan sepuluh strategi tersebut, akan tetapi dengan tiga strategi diatas dapat membantu mahasiswa dalam meningkatkan pembelajaran mereka.

## 3 Metode Pengembangan

Metode yang digunakan dalam penelitian ini lebih tepatnya menggunakan metode penelitian dan pengembangan.

### 3.1 Model Pengembangan

Model pengembangan ini menggunakan model ADDIE yang merupakan singkatan dari *Analysis, Design, Development, Implementation, Evaluasion*, yang telah lama dirumuskan oleh Reiser dan Mollenda ditahun 1990.

### 3.2 Prosedur Pengembangan

#### 3.2.1 Analisis

Kurangnya partisipasi mahasiswa dalam kegiatan ko-kurikuler, pembelajaran yang terpusat pada dosen, serta media belajar mahasiswa yang masih personal. Berdasarkan analisis kinerja tersebut, maka dibutuhkan (1) materi ajar pemrograman sebagai simulasi pembelajaran tersidistribusi menggunakan *Git Server*, (2) media belajar yang dapat mensosialisasikan kode program secara terbuka, (3) serta dapat berkolaborasi antara mahasiswa dengan mahasiswa kemudian mahasiswa dengan dosen.

#### 3.2.2 Perancangan

Merancang perangkat pembelajaran mata kuliah pemrograman java dengan penekanan pada pendekatan *social coding* dengan menambah media belajar *Git Server*.

Merancang perangkat sistem pada sisi *server*:



1. *Openssh-server*, untuk menginstalnya lakukan perintah:

```
$ sudo apt-get install openssh-server
```

2. *git-1.9.2.tar.gz*, Ekstensi *\*.tar.gz* merupakan bentuk menginstal aplikasi melalui *source-nya*, caranya:

```
# instal library
$ sudo apt-get install libcurl4-gnutls-dev \
  libexpat1-dev gettext libz-dev libssl-dev \
  build-essential
# instal Git
$ tar -zxf git-1.9.2.tar.gz
$ cd git-1.9.2/
$ make prefix=/usr/local all
$ make prefix=/usr/local install
```

3. *Gitolite-admin*, untuk menginstal *gitolite* dari *source-nya clone* dari *GitHub* dengan asumsi direktori *\$HOME/bin* sudah dibuat, ini adalah jalan agar *gitolite* dapat diinstal:

```
$ git clone git://github.com/sitaramc/gitolite
$ gitolite/install -to /home/git/bin
$ /home/git/bin/gitolite-setup -pk \
  id_rsa_administrator.pub
```

4. *Gitweb*, *gitweb* menyediakan antarmuka *web* untuk *git repository*, untuk menginstal *gitweb* dari *source-nya*:

```
$ cd git-1.9.2/
$ cp -Rf gitweb/ /var/www/
$ nano /var/www/gitweb/gitweb.cgi
Edit gitweb.cgi sesuai kebutuhan.
```

5. *git-daemon*, *git-daemon-run* memungkinkan pengguna dapat *clone repository* melalui protokol *git* port 9418, protokol ini akan bekerja jika *rule firewall* diaktifkan dan dibuktikan pada *repository* yang memiliki *file* *git-daemon-export-ok*. Perintah mengaktifkan *git-daemon*:

```
$ git daemon --bash-path=/opt/git --detach-syslog \
  --export-all
```

6. *bind9*, *apache2*, kedua paket tersebut sudah terinstall ketika menginstal sistem operasi *linux* Ubuntu Server 12.04.1 LTS. *bind9* sebagai *DNS server* dan *apache2* sebagai *web server*. Yang dengan kedua paket tersebut akan didapatkan alamat *url* *http://www.gitserver/* mengarah ke *path* */opt/git/gitweb/* dan *domain* *git.gitserver* mengarah *\$Home/git/repositories* atau */opt/git*.

7. *ist-dhcp-server*, paket ini digunakan sebagai *DHCP server*, untuk memberi *ip address* otomatis. Perintah menginstal:

```
$ apt-get install isc-dhcp-server
```

Memasang *Git* pada sisi *client* dan *administrator*:

### **Linux**

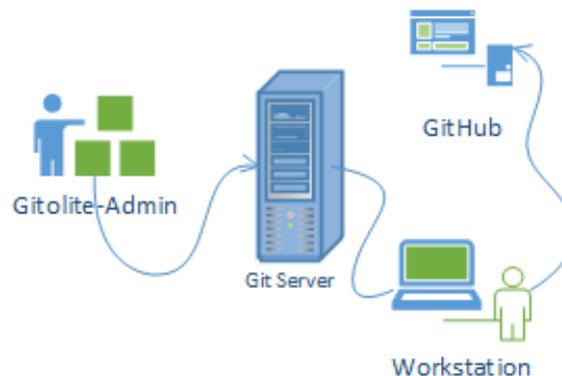
```
$ apt-get install openssh-client  
$ apt-get install git-core
```

Paket *openssh-client* digunakan untuk *generating publickey* dan untuk komunikasi protokol SSH.

**Windows** Versi *installer git* dapat diunduh pada alamat [msysgit.github.io](https://msysgit.github.io). Untuk windows paket aplikasi SSH sudah tersinstal ketika menginstal *git installer*.

Spesifikasi *Hardware* untuk *server* Pentium (R) Dual-Core CPU E5300 @ 2.60GHz, RAM 2GB.

Rancangan Jaringan digambarkan sebagai berikut:



Gambar 3: *Workflow System*.

### **3.2.3 Pengembangan**

Sebelum mengerjakan latihan dan tugas java maka mahasiswa perlu latihan mengoperasikan *git*, seperti yang digambarkan:

Lihat Gambar 4.

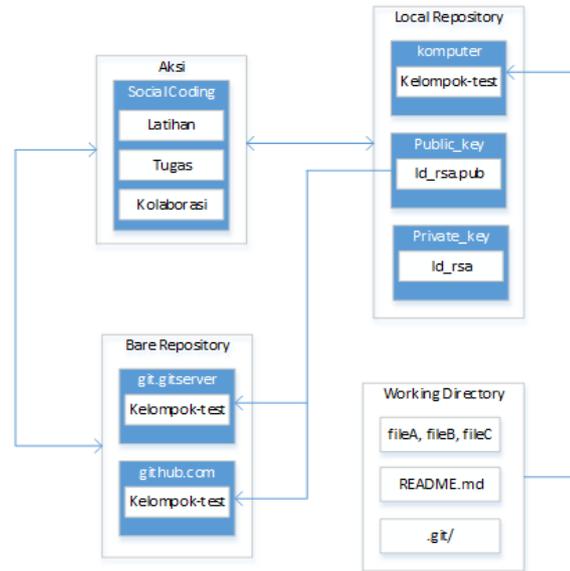
Pada Gambar diatas *repository* di letakkan di alamat tersendiri, khusus mahasiswa untuk melakukan latihan<sup>1</sup> mengoperasikan *git*. Dalam latihan tersebut pada berkas `README.md` menjelaskan latihan dan tugas untuk mengoperasikan *git*.

Ketika menyampaikan materi java tetap menggunakan *slide* dan ceramah sedangkan untuk silabus<sup>2</sup>, latihan dan tugas menggunakan *git repository*, sebagai asumsi kita buat *repository* latihan-java-oop dengan alamat *url*.<sup>3</sup>

<sup>1</sup><http://www.gitserver/kelompok-test.git>

<sup>2</sup><https://github.com/saifulindo/syllabus>

<sup>3</sup><http://www.gitserver/dosen/latihan-java-opp.git>

Gambar 4: Latihan Mengoperasikan *git*.

### 3.2.4 Implementasi

Instalasi sisi *Server* telah dibahas pada bagian perancangan sedangkan sisi *client* juga sudah dibahas pada landasan teori. Sebelumnya Instal dan Konfigurasi `public_key`.<sup>4</sup> Agar instruktur dapat mengakses *terminal server* dan mengkonfigurasi *gitolite-admin*, *Administrator* sebelumnya telah memberikan akses *read-write*.

*Administrator* memberikan hak akses *read-write* kepada instruktur:

```
repo gitolite-admin
RW+   = saifulindo @dosen
```

```
repo dosen/[a-z].*
C     = @dosen
```

Instruktur membuat silabus, latihan, dan tugas:

```
$ git clone git@git.gitserver:dosen/latihan-java-oop.git
```

buat atau kalau sudah ada salin berkas latihan, tugas dan silabus pada *directory* latihan -java-oop/, setelah itu lakukan pengunggahan:

```
$ git add .
$ git commit -m "Initial commit"
$ git push -u origin master
```

Instruktur mengizinkan mahasiswa untuk *cloning repository* latihan-java-oop dengan mengizinkan akses protokol HTTP:

<sup>4</sup><https://github.com/komunitas-cahump/InstallGit>



```
$ ssh git@git.gitserver
# masukkan password
$ cd /opt/git/dosen/latihan-java-oop.git
$ sudo mv hooks/post-update.sample hooks/post-update
$ chmod a+x hooks/post-update
```

berikutnya instruktur mengkonfigurasi *gitolite-admin*:

```
$ git clone git@git.gitserver:gitolite-admin.git
$ cd gitolite-admin/
$ sublime conf/gitolite.conf
# Konfigurasi
repo dosen/latihan-java-oop
  RW+ = @instruktur
  desc = "Repo lat.java"
  R    = gitweb daemon
# simpan dan keluar
$ git add .
$ git commit -m "Initial commit"
$ git push -u origin master
```

Mahasiswa melihat dan duplikasi *repository* latihan-java-oop dengan meng-akses alamat <http://www.gitserver/dosen/latihan-java-oop/> untuk menduplikasi lakukan:

```
$ git clone http://git.gitserver/dosen/latihan-java-oop.git
```

Instruktur mengkonfigurasi *gitolite-admin* dengan menambahkan hak akses:

```
$ sublime conf/gitolite.conf
repo mahasiswa/[a-z].*
  C    = @mahasiswa
# simpan dan keluar
# lakukan pengunggahan seperti
  sebelumnya
```

Mahasiswa membentuk kelompok dan yang ditunjuk sebagai ketua melakukan *cloning repository* untuk mengerjakan tugas:

```
$ git clone git@git.gitserver:mahasiswa/tugas-java-oop.git
```

mahasiswa melaporkan kepada instruktur, untuk memberitahukan alamat *repository* tugas yang telah dibuat.

Instruktur mengkonfigurasi hak akses *repository* tugas:

```
# jika close source
repo mahasiswa/tugas-java-oop
  RW+ = John Doe Smith Alice
```



```
# jika open source
repo mahasiswa/tugas-java-oop
RW+ = John Doe Smith Alice
desc = "Repo Tugas Java"
R = gitweb daemon
```

Mahasiswa dapat berkolaborasi mengerjakan tugas kelompok sesuai batas waktu pengumpulan yang telah ditentukan oleh instruktur.

Instruktur dapat memantau keaktifan dan perkembangan mahasiswa dalam mengerjakan latihan dan tugasnya.

### 3.2.5 Evaluasi

Dalam rancangan *Git Server* ini beberapa hal yang penting untuk diketahui:

1. Alamat domain melalui *browser* yaitu `http://www.gitserver/`
2. Alamat hostname yang di *cloning* melalui protokol SSH itu `git@git.gitserver`, sedangkan protokol HTTP `http://git.gitserver`.
3. *Repository* yang *close source* kodenya tidak dapat di akses melalui domain `http://www.gitserver` dan sebaliknya kalau *Repository* yang *open source* maka dapat diakses melalui domain.

Jika *git server* diimplementasi dalam skala luas yaitu menangani satu kampus, maka perlu penanganan khusus dalam manajemen *repository* dan *users*.

## 3.3 Uji Coba

Uji coba dilakukan di kampus Fakultas Teknik jurusan Teknik Informatika, mahasiswa tingkat dua. Uji coba terbatas pertama dilakukan kepada tujuh mahasiswa, berikutnya uji coba terbatas kedua dilakukan kepada sepuluh mahasiswa, dan uji coba luas dilakukan kepada dua kelas yaitu kelas 2a dan 2g tahun akademik 2014/2015.

Bersama dengan uji coba juga dilakukan uji validasi kepada para ahli dan praktisi yaitu para dosen pengajar sebanyak lima validator.

## 4 Hasil Penelitian dan Pembahasan

### 4.1 Uji Coba Pertama

Uji coba pertama terukur 87,79% kategori sangat layak, perbaikan pada sisi *server* dengan menambahkan nameserver seperti:

```
$ nano /etc/resolve.conf
# Menambah nameserver disini
nameserver 192.168.92.1
nameserver 192.168.91.1
# simpan dan keluar
```

serta mengkonfigurasi *server* sebagai NAT.

## 4.2 Uji Coba Kedua

Uji coba kedua terukur 85,70% kategori sangat layak, dengan perbaikan membuat modul panduan mengoperasikan *Git Server* untuk mahasiswa.

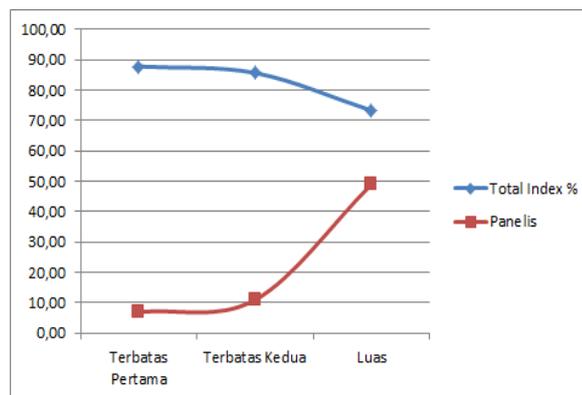
## 4.3 Uji Coba Luas

Uji coba luas terukur 73,30% kategori layak, dengan perbaikan membuat modul panduan mengoperasikan *Git Server* untuk Instruktur/Dosen.

## 4.4 Pembahasan

Berdasarkan hasil uji coba maka rancangan *Git Server* layak untuk dioperasikan sebagai media pembelajaran mata kuliah pemrograman. Untuk mengimplikasikan bahwa rancangan *Git Server* dapat meningkatkan pembelajaran mahasiswa, maka dilakukan uji kemanfaatan yang terukur 77,60% kategori baik, yaitu yang diukur: keaktifan mahasiswa dalam mengikuti mata kuliah, sikap mahasiswa untuk mengasah kemampuan pemrograman dengan mahasiswa lain, dan sikap mahasiswa untuk merefleksikan apa dan bagaimana mereka belajar di kelas.

Hasil statistik uji kelayakan digambar seperti berikut:



Gambar 5: Output *Hasil Analisis*

Setelah dilakukan uji coba kelayakan, kemanfaatan, kevalidtan, serta berbagai perbaikan maka didapatkan model simulasi komputer:

Lihat Gambar 6:

## Kesimpulan

Berdasarkan hasil penelitian yang telah diuraikan, dapat disimpulkan bahwa:

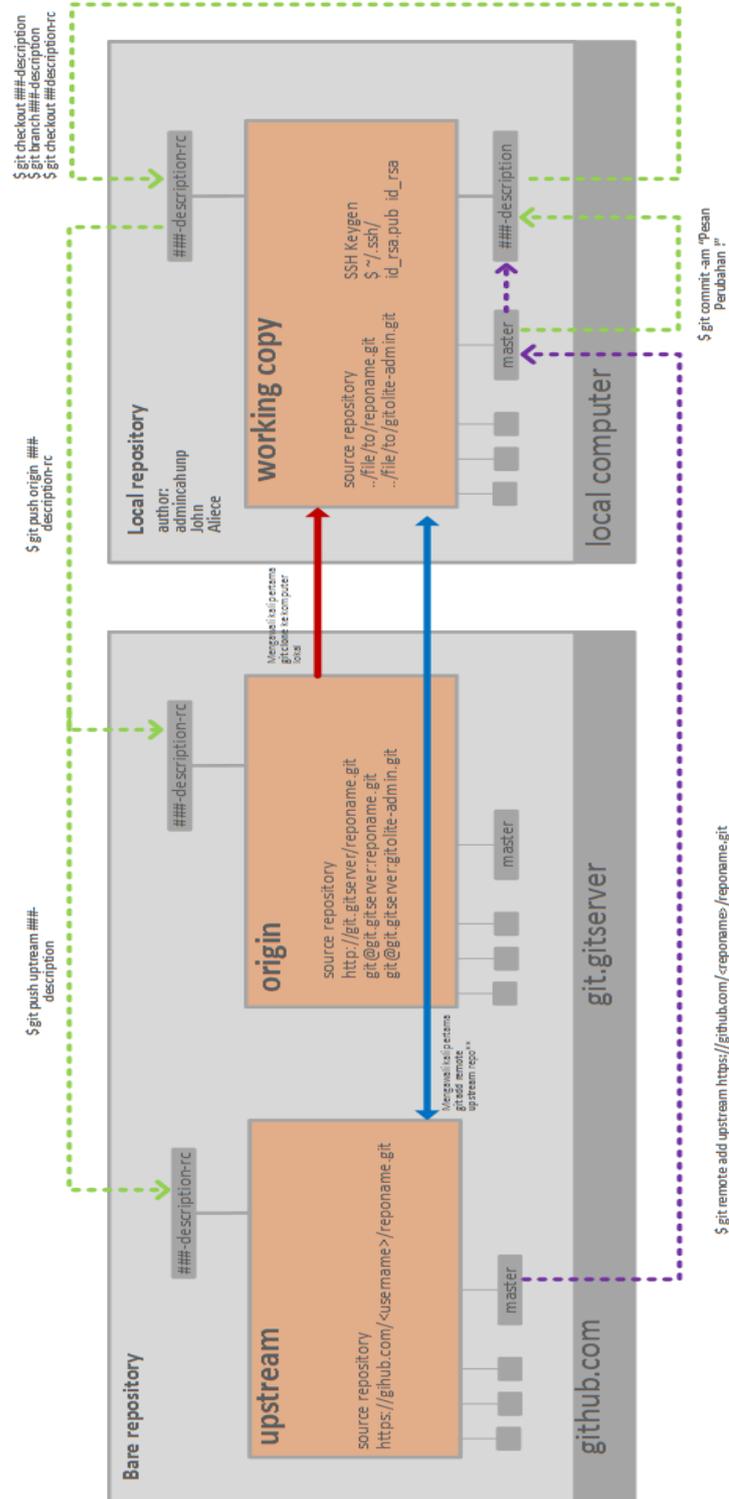
1. Untuk merancang *Git Server* dibutuhkan keahlian dalam bidang jaringan dan pemrograman, *Git Server* dirancang harus dapat digunakan sebagai *public repository* maupun *private repository*, harus dapat memajemen *repository* dan *users*, serta harus dapat digunakan sebagai kolaborasi pembelajaran yang terdistribusi



2. Rancangan *Git Server* dengan pendekatan *GitHub Social Coding* diimplikasikan terbukti dapat meningkatkan pembelajaran mahasiswa

## References

- [1] Romi, Satria, W. 2007. *Sistem e-Learning Berbasis Model Motivasi Komunitas*. Jurnal Teknodik, Vol 1 (No. 1) 2007.
- [2] Linus, Torvalds. 2007. *Re: Kernel SCM Saga. Linux-kernel (Mailling list)*. (online). Tersedia : <http://marc.info/?l=linux-kernel&m=111288700902396>, diunduh 18 Januari 2013.
- [3] Sumarno, Alim, M.Pd, 2011. *Model Pembelajaran Kooperatif*. (online). tersedia: <http://blog.elearning.unesa.ac.id/alim-sumarno/model-pembelajaran-kooperatif>, diunduh Januari 2014.
- [4] Scott, Chacon. 2009. *Pro Git*. (online). tersedia : <https://github.s3.amazonaws.com/media/progit.en.pdf>, diunduh 16 Januari 2013.
- [5] Wlodzimierz, Gajda. 2013. *Git Recieps*. New York : Apress. (online). tersedia : <https://github.com/kospiootr/kospiootr.github.io/blob/master/resources/books/Git%20Recipes%20-%20Wlodzimierz%20Gajda.pdf>, diunduh 16 Oktober 2014.
- [6] Jan, Brazdil. 2013. *Automatic Pull Request Integration*. Disertasi. (online). tersedia : [http://is.muni.cz/th/374346/fi.b/bakalarka\\_1\\_1.pdf](http://is.muni.cz/th/374346/fi.b/bakalarka_1_1.pdf), diunduh 20 Maret 2014.
- [7] George, M. Doss. 1999. *Tipe Server RED HAT Linux*. Terjemahan Imam, Musthaqim. Jakarta : PT. Elex Media Komputindo.
- [8] Jon, Loeliger. dan Matthew, McCollough. 2012. *Version Control with Git 2nd Edition*. United State of America : OReilly Media. OReilly, (online). tersedia : <http://oreilly.com/>, diunduh 1 September 2014.
- [9] Contributor, Team. 2012. *Ubuntu Server Guide*. (online). tersedia : <https://help.ubuntu.com/12.04/serverguide/serverguide.pdf>, diunduh 31 Agustus 2013.
- [10] Direktorat Jendral Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan. 2012. *Undang undang Nomor 12 Tahun 2012 Tentang Pendidikan Tinggi*. (online). tersedia : [http://sipuu.setkab.go.id/PUUdoc/17624/UU0122012\\_Full.pdf](http://sipuu.setkab.go.id/PUUdoc/17624/UU0122012_Full.pdf), diunduh 10 November 2014.
- [11] Linda, Suskie. 2002. *Strategies to Improve Student Learning*. (online). tersedia : <https://www.mc.uky.edu/healthsciences/docs/Strategies-for-Improving-Student-Learning.pdf>, diunduh 10 November 2014.



Gambar 6: Model Simulasi Komputer