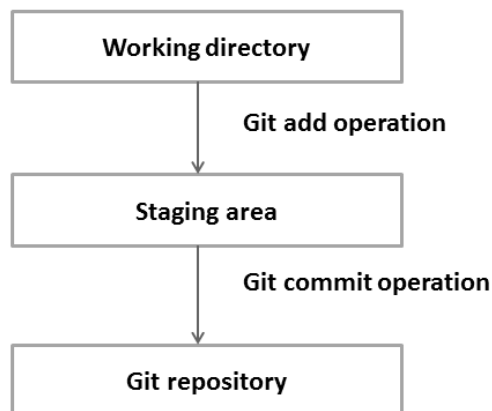

Git

1. Spesifikasi Perangkat Lunak

Git merupakan Pengendali Revisi dan Sistem Manajemen Source Code Terdistribusi yang menitikberatkan pada kecepatan. Git dirancang dan dikembangkan oleh Linus Torvalds untuk pengembangan Kernel Linux. Git merupakan Software Gratis yang didistribusikan di bawah lisensi GNU General Public License versi 2.

Pada ulasan ini, kita berkonsentrasi pada VCS yang terdistribusi dan fokus pada Git. Hal yang perlu diketahui tentang VCS adalah sebagaimana terlihat pada gambar berikut.



Gambar 1 VCS

Working Directory adalah tempat dimana kita secara lokal bekerja (apakah di notebook atau di PC atau di tempat lainnya).

Git add adalah perintah untuk menyalin dan memindahkan status dari file dari Working Directory ke Staging Area atau Index. Sedangkan Git commit adalah perintah untuk menyalin dan memindahkan status dari Staging Area atau Index ke Git Repository. Fokus kita saat ini adalah melakukan eksplorasi pada Github, yang merupakan software CVS terdistribusi.

2. Instalasi

Bagian ini akan menjelaskan secara sederhana, instalasi Git pada berbagai sistem operasi seperti Windows, Ubuntu, CentOS, dan MacOS.

1. Windows

Instalasi git pada Windows dapat dilakukan dengan mengunduh file installer di

<https://git-scm.com/download/win>

2. Ubuntu

Instalasi git pada Ubuntu dapat dilakukan dengan mengeksekusi perintah: `apt-get install git`

3. CentOS

Instalasi git pada CentOS dapat dilakukan dengan mengeksekusi perintah: `Yum install git`

4. MacOS

Instalasi git pada MacOS dapat dilakukan dengan mendownload installer yang ada di

<https://git-scm.com/download/mac>

3. Flow

Pada bagian ini, dijelaskan teknis untuk melakukan versioning menggunakan git. Untuk memulai tutorial, buatlah sebuah direktori khusus untuk belajar dan masuk ke direktori tersebut menggunakan terminal. Secara umum flow untuk menggunakan Git adalah sebagai berikut:

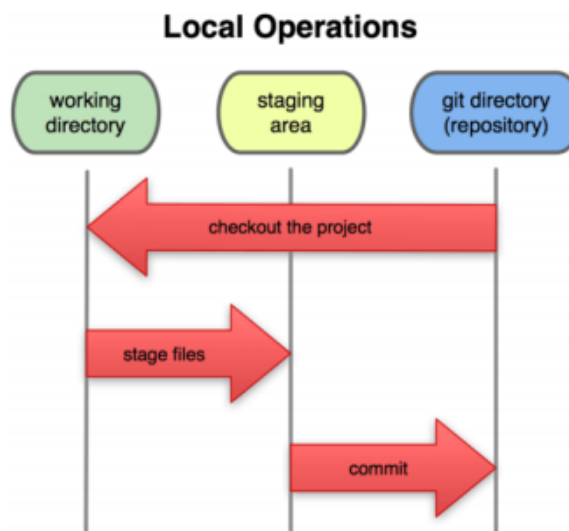
1. Inisiasi repository untuk git
2. Mengatur remote repository yang ingin digunakan
3. Melakukan pull file dari remote repository
4. Menambahkan atau mengubah file pada local repository
5. Menambahkan file ke staging state
6. Melakukan push ke remote repository

Atau juga dapat dilakukan dengan menggunakan fitur cloning sehingga alur kerja menjadi seperti berikut:

1. Clone remote repository
2. Menambah / mengubah file
3. Menambahkan file ke staging
4. Melakukan push ke remote repository

Git memiliki 3 keadaan utama di mana berkas anda dapat berada: committed, modified dan staged. Committed berarti data telah tersimpan secara aman pada basisdata lokal. Modified berarti anda telah melakukan perubahan pada berkas namun anda belum melakukan commit pada basisdata. Staged berarti anda telah menandai berkas yang telah diubah pada versi yang sedang berlangsung untuk kemudian dilakukan commit.

Ini membawa kita ke tiga bagian utama dari sebuah proyek Git: direktori Git, direktori kerja (working directory), dan staging area.



Direktori Git adalah tempat Git menyimpan metadata dan database objek untuk proyek anda. Ini adalah bahagian terpenting dari Git, dan inilah yang disalin ketika anda melakukan cloning sebuah repository dari komputer lain.

Direktori kerja adalah sebuah checkout tunggal dari satu versi dari proyek. Berkas-berkas ini kemudian ditarik keluar dari basisdata yang terkompresi dalam direktori Git dan disimpan pada disk untuk anda gunakan atau modifikasi.

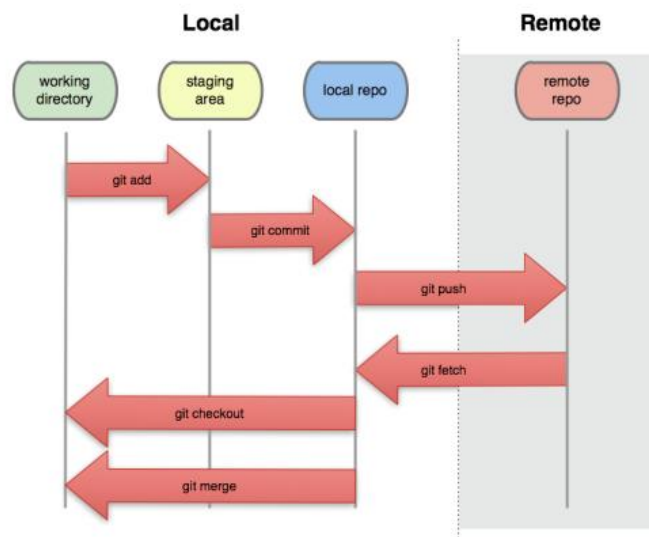
Staging area adalah sebuah berkas sederhana, umumnya berada dalam direktori Git anda, yang menyimpan informasi mengenai apa yang menjadi commit selanjutnya. Ini terkadang disebut sebagai index, tetapi semakin menjadi standard untuk menyebutnya sebagai staging area.

Jika sebuah versi tertentu dari sebuah berkas telah ada di direktori git, ia dianggap 'committed'. Jika berkas diubah (modified) tetapi sudah ditambahkan ke staging area, maka itu adalah 'staged'. Dan jika berkas telah diubah sejak terakhir dilakukan checked out tetapi belum ditambahkan ke staging area maka itu adalah 'modified'. Terakhir, ingat bahwa seluruh proses tersebut terjadi hanya di komputer lokal anda.

Ketika Server Repository Terlibat

Dalam Git, seringkali anda memerlukan suatu server penyedia layanan repository. Server ini dalam terminologi Git disebut sebagai “remote”. Server ini menyediakan tempat terpusat di internet sehingga developer lain dapat berkolaborasi dengan perantara server tersebut.

Ketika anda telah menyelesaikan operasi di komputer lokal anda (add, commit), anda dapat menyimpan keadaan repository lokal anda ke server. Kegiatan ini disebut sebagai “push”. Ketika anda “push”, maka keadaan repository remote akan disamakan dengan keadaan repository lokal anda.



Operasi-operasi Dasar

Init

Perintah init digunakan untuk inisiasi git. Biasanya inisiasi dilakukan oleh pemimpin proyek.

Anggota lain akan melakukan clone setelah pemimpin proyek melakukan inisiasi repository. Init dapat digunakan di proyek baru (masih kosong) atau di proyek yang sudah dikerjakan (sudah ada file source code).

Clone

Perintah clone digunakan untuk menyalin repository dari remote repository ke local repository.

Add

Perintah add digunakan untuk menambahkan file ke staging area.

Commit

Perintah commit digunakan untuk menyimpan perubahan kode di repository local.

Push

Perintah push digunakan untuk mengirim commit dari local repository ke remote server.

Checkout

Perintah checkout digunakan untuk berpindah dari satu branch ke branch lain. Checkout juga digunakan untuk mengembalikan file yang diubah tapi belum dicommit ke versi sebelum diedit.

Fetch

Perintah fetch digunakan untuk menyamakan keadaan remote repo dengan local repo (mengupdate local repo).

Merge

Perintah merge digunakan untuk menggabungkan branch.

Pull

Perintah pull digunakan untuk menarik commit dari remote server ke lokal. Aslinya, pull ini melakukan fetch yang diikuti merge secara otomatis.

4. Perintah-perintah pada Git

Pada pembahasan ini menggunakan Gitlab yang merupakan suatu tools yang menggunakan system Git. Terdapat tools lain juga, diantaranya Github dan Bitbucket.

1.4.1 Cloning Repository

Cloning sebuah repository adalah aksi untuk melakukan penggandaan working remote repository dengan tujuan untuk mendapatkan working copy dari sebuah repository agar dapat dilakukan

versioning di local. Cloning repository dapat dilakukan dengan perintah `git clone {url}` dengan `{url}` adalah URL remote repository.

1.4.2 Inisiasi Git

Cara lain untuk memulai menggunakan Git adalah dengan melakukan inisiasi git pada local folder. Inisiasi git pada local folder dilakukan dengan mengeksekusi perintah `git init`. Perintah ini akan menghasilkan satu buah hidden folder dengan nama `.git`.

1.4.3 Mengambil File dari Remote Repository

Langkah selanjutnya setelah mengatur remote repository yang digunakan adalah melakukan pull terhadap file yang ada di remote repository. Perintah yang digunakan untuk melakukan hal ini adalah `git remote pull {name} {branch}` yang akan mengambil file pada remote repository dengan branch `{branch}`.

1.4.4 Mengecek Perubahan Status File

Dari waktu ke waktu, file yang ada mungkin mengalami perubahan. Untuk mengecek file apa saja yang berubah, dapat digunakan perintah `git status`.

Dapat dilihat bahwa, file `newfile` yang notabene adalah file baru terdeteksi sebagai perubahan.

Namun, file tersebut belum dapat dikirim ke remote repository karena belum masuk ke index local repository.

1.4.5 Menambah File pada Local Repository

Untuk melakukan penambahan file, digunakan perintah `git add {pathToFile}` atau dengan `git add -A` untuk menambahkan semua file yang sudah dibuat pada repository local.

Dengan demikian, file `newfile` sudah masuk pada index file yang ditrack oleh git pada local repository.

1.4.6 Melakukan Staging / Commit Perubahan

Untuk dapat di-push ke remote repository, harus dilakukan commit perubahan pada file yang ada di local repository. Commit dapat dilakukan dengan perintah `git commit -m {message}` yang akan menggunakan `{message}` sebagai deskripsi commit yang dilakukan. Hasil eksekusi dapat dilihat pada

Dengan demikian, file tersebut sudah masuk staging dan siap untuk dikirim ke remote repository. Perlu diperhatikan bahwa setiap commit menghasilkan kode seperti `8caab8a` yang dapat digunakan

untuk “kembali” ke kondisi saat itu. Hal ini sangat berguna untuk meng-undo perubahan yang mungkin menyebabkan kode tidak berfungsi.

1.4.7 Mengunggah File ke Remote Repository

Perubahan yang ada di local repository harus di-push ke remote repository agar anggota tim lain bias mendapatkan file terbaru hasil pengubahan yang dilakukan. Pengunggahan ini dapat dilakukan dengan perintah `git push {name} {branch}` yang mana akan melakukan push file ke branch `{branch}`.

Dengan demikian, file yang ada pada local repository telah tersedia di remote repository dan dapat digunakan oleh anggota tim yang lain.

1.4.8 Membuat dan Berpindah ke Branch

Seperti yang diterangkan sebelumnya, kelebihan version control adalah dimungkinkan adanya beberapa line of development. Line of development dapat diidentikkan dengan branch yang ada pada sebuah repository seperti master / development. Untuk membuat branch baru dapat digunakan perintah `git branch {branch}` yang akan membuat branch dengan nama `{branch}`. Untuk berpindah ke branch tersebut, digunakan perintah `git checkout {branch}`.

Sekarang, working branch pada local directory menjadi development. Perubahan pada branch ini tidak akan berpengaruh pada branch master.

1.4.9 Menggabungkan Branch

Setelah pekerjaan pada suatu branch selesai maka perlu dilakukan penggabungan branch tersebut ke mainline development. Merge dapat dilakukan dengan perintah `git merge {branch}` yang akan menggabungkan branch `{branch}` dengan current branch.

Setelah merge dilakukan maka perubahan yang ada pada branch development akan muncul pada branch master. Hal ini terlihat dengan adanya newfiledev yang dibuat pada branch development.

1.4.10 Revert Changes

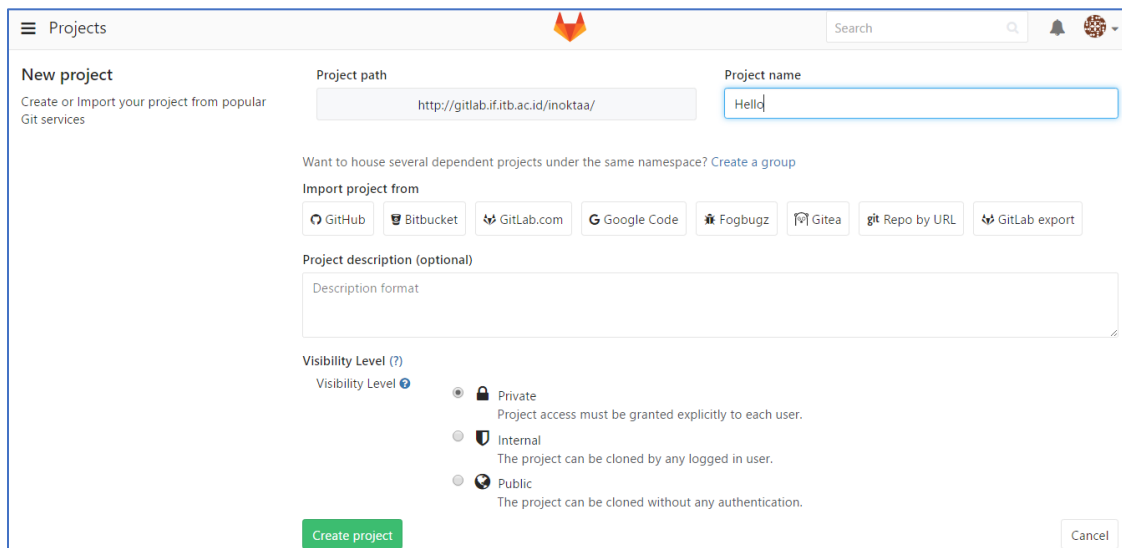
Revert changes berfungsi untuk kembali ke versi file atau branch tertentu. Biasanya, revert dilakukan untuk melakukan undo terhadap perubahan yang mengakibatkan program tidak berfungsi. Perintah umum untuk melakukan revert adalah `git checkout commit_hash`.

5. Cara Penggunaan Perangkat Lunak

Create Project

Create New Project adalah aksi untuk membuat project baru, dimana dapat melakukan import dari repository lain yaitu GitHub, Bitbucket, Google Code, Fogbugz, Gitea dan lain lain. Kemudian Pilih visibility level yang diinginkan, diantaranya:

1. Private, proyek hanya bisa diakses oleh member di proyek itu saja.
2. Internal, proyek bisa diakses oleh semua member yang terdaftar di CS IPB GitLab.
3. Public, proyek bisa diakses semua orang tanpa otentikasi.



Gambar 2 Membuat Repository

Setelah membuat repository, tambahkan member. Apa perbedaan member dengan non member? Intinya member punya akses menulis sesuai levelnya, sedangkan non member hanya bisa clone saja. Jika proyek bersifat private proyek hanya bisa di-clone oleh member saja.

Untuk menambahkan member, tekan menu Members -> Add members.

Ketik username member yang akan dimasukkan, kemudian berikan project access yang diinginkan. Sesama developer sebaiknya memiliki project access developer atau master. Setelah itu tekan Add user to Project.

Setup

Setup berikut ini adalah untuk mengkoneksikan repository local ke remote repository.

Git global setup

```
git config --global user.name "Intan Oktafiani"

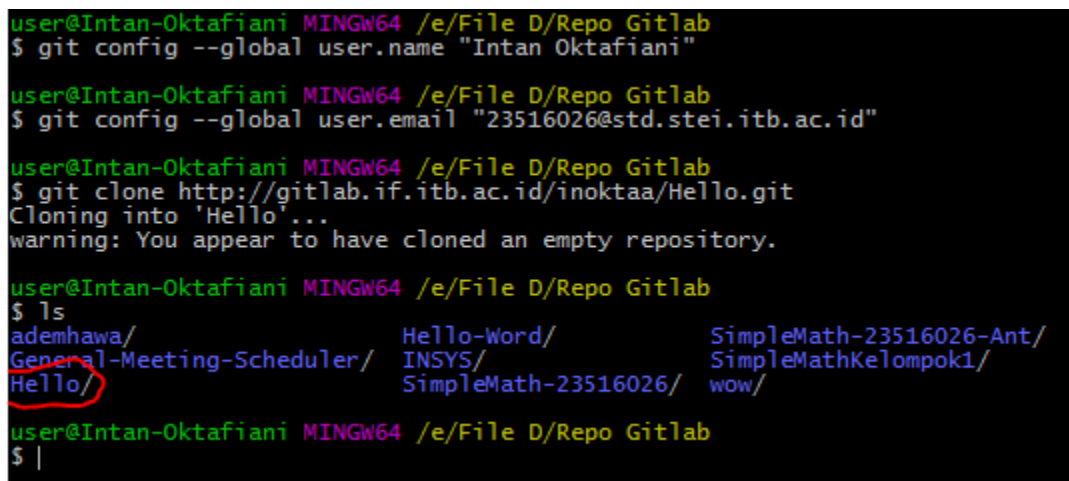
git config --global user.email "23516026@std.stei.itb.ac.id"

git clone http://gitlab.if.itb.ac.id/inoktaa/Hello-Word.git
```

Clone ke repository local

```
git clone http://gitlab.if.itb.ac.id/inoktaa/Hello.git
```

Cloning sebuah repository adalah aksi untuk melakukan penggandaan working remote repository dengan tujuan untuk mendapatkan working copy dari sebuah repository agar dapat dilakukan versioning di local. Cloning repository dapat dilakukan dengan perintah `git clone {url}` dengan `{url}` adalah URL remote repository. Perintah diatas akan menghasilkan folder baru hasil cloning remote repository seperti yang dapat dilihat pada Gambar 5 Contoh Cloning Remote Repository.



```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab
$ git config --global user.name "Intan Oktafiani"

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab
$ git config --global user.email "23516026@std.stei.itb.ac.id"


user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab
$ git clone http://gitlab.if.itb.ac.id/inoktaa/Hello.git
Cloning into 'Hello'...
warning: You appear to have cloned an empty repository.

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab
$ ls
adempawa/      Hello-Word/      SimpleMath-23516026-Ant/
General-Meeting-Scheduler/  INSYS/           SimpleMathKelompok1/
Hello/         SimpleMath-23516026/  wow/

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab
$ |
```

Pada gambar di atas, repository local sudah bertambah dengan nama folder sesuai dengan nama project pada gitlab yaitu Hello. Kemudian kita dapat menambahkan file readme untuk inisiasi file. Dengan cara sebagai berikut:

Buat file pada direktori lokal

Name	Date modified	Type	Size
 Readme	3/2/2017 8:44 AM	Text Document	1 KB

Lakukan git add, git commit dan git push untuk menambahkan file ke gitlab.

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab
$ cd Hello

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git add -A

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git commit -m "Initiate"
[master (root-commit) 2f53eba] Initiate
1 file changed, 1 insertion(+)
create mode 100644 Readme.txt

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git push
Counting objects: 3, done.
Writing objects: 100% (3/3), 224 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To http://gitlab.if.itb.ac.id/inoktaa/Hello.git
 * [new branch]      master -> master

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ |
```

Kita sudah berhasil menambahkan file dan dapat menambahkan file lain untuk kemudian dikerjakan oleh branch lain yang merupakan copy dari master, untuk kemudian dapat dimodifikasi.

New Branch

Branch name

satu

Create from

master

Existing branch name, tag, or commit SHA

Create branch

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git checkout satu
error: pathspec 'satu' did not match any file(s) known to git.

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git fetch
From http://gitlab.if.itb.ac.id/inoktaa/Hello
* [new branch]      satu      -> origin/satu

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ |
```

Gambar di atas adalah percobaan untuk pindah dari branch master ke branch satu. Percobaan gagal karena belum melakukan perbaruan status dengan git fetch. Setelah git fetch dilakukan master baru mendapat update bahwa saat ini sudah ada branch baru bernama “satu”.

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git checkout satu
Branch satu set up to track remote branch satu from origin.
Switched to a new branch 'satu'

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ |
```

Pada gambar dibawah ini, branch satu melakukan penambahan class “Pembagian” pada project aritmatika. Sebelum melakukan penambahan file pada branch satu, dilakukan git status untuk mengetahui perubahan ada yang ada.

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ git status
On branch satu
Your branch is up-to-date with 'origin/satu'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Aritmatika/src/aritmatika/Pembagian.java

nothing added to commit but untracked files present (use "git add" to track)
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ |
```

Penambahan class Pembagian.java oleh branch satu dengan perintah git add, git commit dan git push.

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ git add -A

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ git commit -m "Adding Pembagian Method"
[satu 6a89d25] Adding Pembagian Method
1 file changed, 25 insertions(+)
create mode 100644 Aritmatika/src/aritmatika/Pembagian.java

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ git push
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 686 bytes | 0 bytes/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote:
remote: To create a merge request for satu, visit:
remote: http://gitlab.if.itb.ac.id/inoktaa/Hello/merge_requests/
est%5Bsource_branch%5D=satu
remote:
To http://gitlab.if.itb.ac.id/inoktaa/Hello.git
b28db1c..6a89d25 satu -> satu

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ |
```

Lakukan pindah branch ke master.

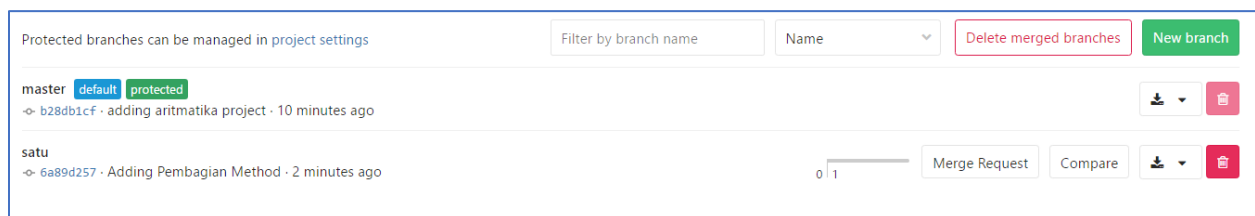
```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (satu)
$ git checkout master
Your branch is up-to-date with 'origin/master'.
Switched to branch 'master'

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git fetch

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git pull
Already up-to-date.

user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ |
```

Dibawah ini adalah tahap untuk melakukan merge melalui antarmuka gitlab.



Source branch

satu

Target branch

master

Change branches

☐ Remove source branch when merge request is accepted.

Submit merge request

Adding Pembagian Method

Request to merge `satu` into `master`

Accept Merge Request

☐ Remove source branch


Adding Pembagian Method

Edited in 8 minutes

Request to merge `satu` into `master`

Merged by  Intan Oktafiani in 8 minutes

The changes were merged into `master`. You can remove the source branch now.

 Remove Source Branch

Revert

Cherry-pick

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git fetch
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (1/1), done.
From http://gitlab.if.itb.ac.id/inoktaa/Hello
 b28db1c..6b9d903  master    -> origin/master
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ |
```

```
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ git pull
Updating b28db1c..6b9d903
Fast-forward
 Aritmatika/src/aritmatika/Pembagian.java | 25 ++++++
 1 file changed, 25 insertions(+)
 create mode 100644 Aritmatika/src/aritmatika/Pembagian.java
user@Intan-Oktafiani MINGW64 /e/File D/Repo Gitlab/Hello (master)
$ |
```

6. Kelebihan dan Kekurangan Perangkat Lunak

A. Keunggulan dari Git :

1. Design yang sederhana
2. Mendukung dalam pengembangan non-linear(pengembangan paralel)
3. Sistem terdistribusi, peer to peer
4. Mendukung dalam proyek besar seperti Kernel Linux
5. Akses menggunakancommand line (tidak harusclient-server)
6. Penyimpanan murni berbasis file (tidak menggunakan database/SQL)

B. Kelemahan dari Git :

1. Tidak Optimal untuk pengembang tunggal
2. Dukungan untuk Windows terbatas dibandingkan Linux