

VERSION 2.1  
JANUARI, 2023



# MANAJEMEN PROYEK PERANGKAT LUNAK

GIT & JIRA CONFLUENCE

MODUL 6

TIM PENYUSUN: - ILYAS NUR S.KOM, M.KOM  
- DINDA ARINAWATI WIYONO  
- MUHAMMAD NUR ICHSAN

PRESENTED BY: LAB. TEKNIK INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## MANAJEMEN PROYEK PERANGKAT LUNAK

---

### CAPAIAN PEMBELAJARAN MATA KULIAH

Dapat memahami git dan github

Dapat memahami jira confluence

---

### SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa dapat memahami dan mengimplementasikan git dan github

Mahasiswa dapat memahami dan mengimplementasikan jira confluence

---

### KEBUTUHAN HARDWARE & SOFTWARE
















JIRA Software

---

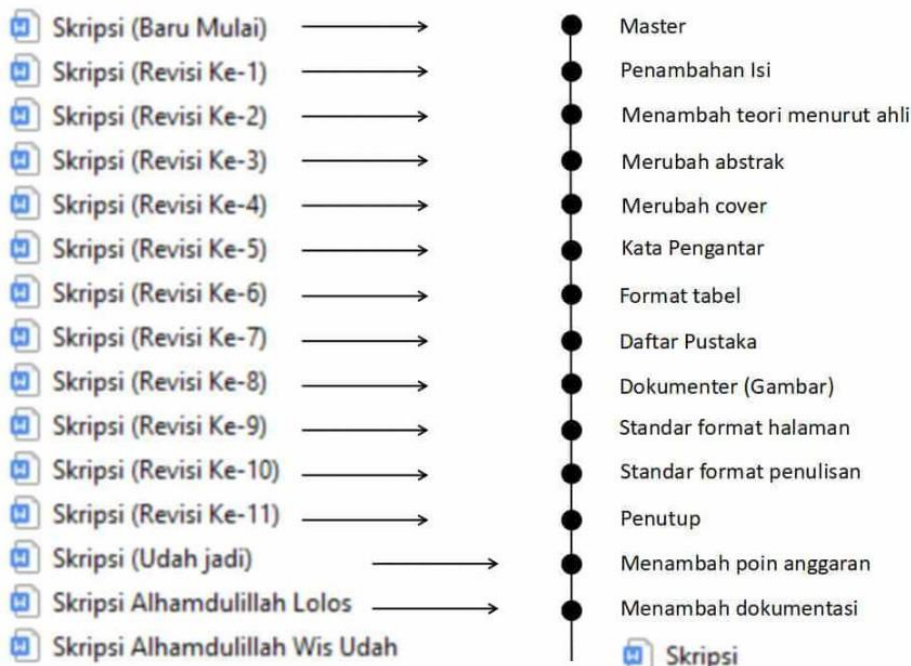
### MATERI POKOK

#### **GIT & GITHUB**

Git adalah salah satu sistem pengontrol versi (Version Control System) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds. Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri. Git dikenal juga dengan distributed revision control (VCS terdistribusi), artinya penyimpanan database Git tidak hanya berada dalam satu tempat saja. Version Control System (VCS) adalah sebuah sistem yang melakukan source code management (SCM) untuk mengelola perubahan di setiap dokumen, program komputer, website, dan kumpulan pemrograman lainnya.

 Skripsi (Baru Mulai)  
 Skripsi (Revisi Ke-1)  
 Skripsi (Revisi Ke-2)  
 Skripsi (Revisi Ke-3)  
 Skripsi (Revisi Ke-4)  
 Skripsi (Revisi Ke-5)  
 Skripsi (Revisi Ke-6)  
 Skripsi (Revisi Ke-7)  
 Skripsi (Revisi Ke-8)  
 Skripsi (Revisi Ke-9)  
 Skripsi (Revisi Ke-10)  
 Skripsi (Revisi Ke-11)  
 Skripsi (Udah jadi)  
 Skripsi Alhamdulillah Lolos  
 Skripsi Alhamdulillah Wis Udah

Lebih jelasnya kamu bisa merasakan contoh kasus yang biasa dilakukan oleh seorang mahasiswa dalam mengerjakan skripsinya. Setiap melakukan revisi, file yang telah lalu tidak akan dibuang dan akan disimpan dengan nama yang berbeda. Sedangkan yang terbaru akan disimpan dengan nama; misal “Skripsi (Revisi Ke-2)”. Kegiatan tersebut akan dilakukan secara terus menerus hingga dalam satu folder skripsi terdapat file Ms. Word dengan kuantitas yang banyak. Tujuan tersebut tidak lain adalah untuk menyimpan history pekerjaan mahasiswa. Hingga akhirnya file terakhir selesai dengan nama “Skripsi Alhamdulillah Wis Udah”.

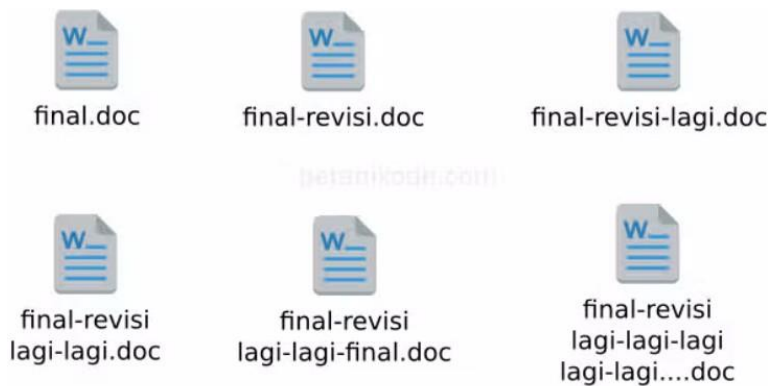


Seperti halnya pada gambar diatas, setiap perubahan data secara manual akan menghasilkan lebih banyak file. Sedangkan pada VCS mengusung konsep menyimpan rekaman perubahan dengan satu file

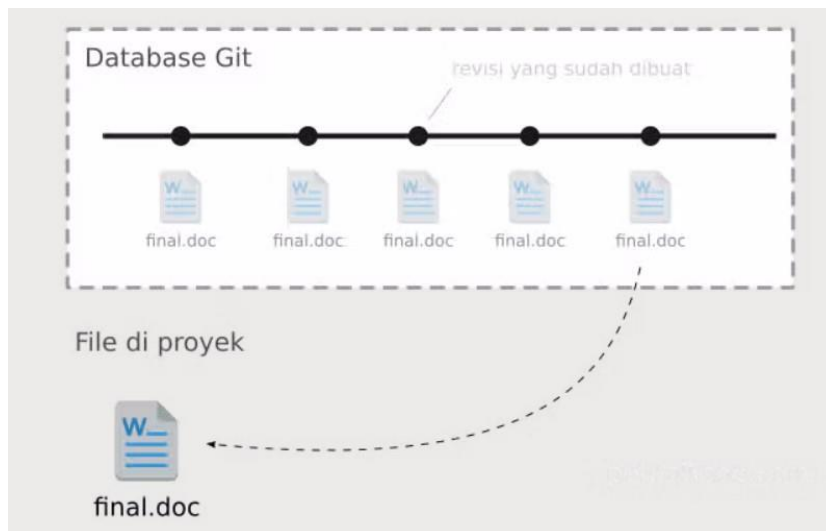
saja. Untuk mengoperasikan Git, kamu perlu menginstall software terlebih dahulu sehingga pekerjaan ini dapat dilakukan secara offline (tidak terkoneksi internet). Software ini juga tersedia secara gratis melalui web unduhan resminya di <https://git-scm.com/downloads>.

Di bawah ini merupakan ilustrasi lain dari konsep history konvensional dan git

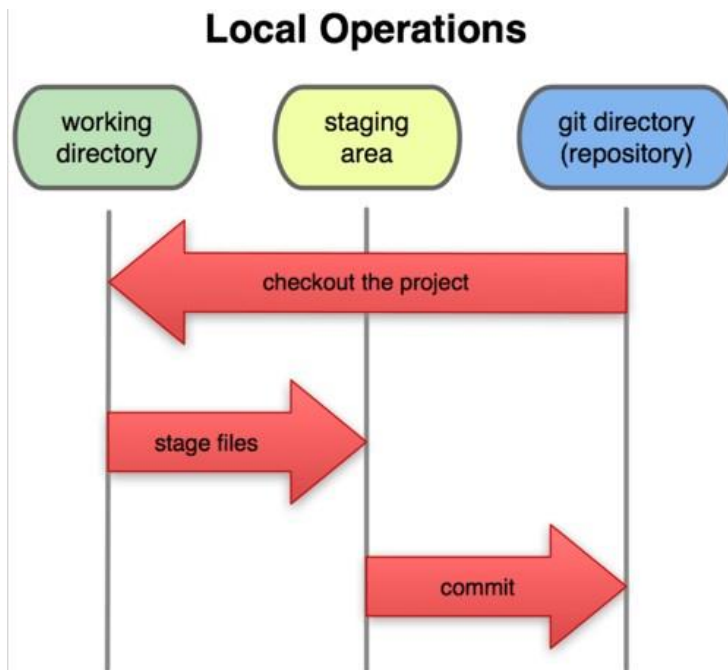
Konvensional :



Git:



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



1. Direktori Git adalah dimana Git menyimpan metadata dan database objek untuk project anda. Ini adalah bagian terpenting dari Git, dan inilah yang disalin ketika anda melakukan kloning sebuah repository dari komputer lain.
2. Direktori kerja adalah sebuah checkout tunggal dari satu versi dari proyek. Berkas-berkas ini kemudian ditarik keluar dari basis data yang terkompresi dalam direktori Git dan disimpan pada disk untuk anda gunakan atau modifikasi.
3. 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.

#### Alur kerja dasar Git adalah seperti ini:

1. Anda mengubah berkas dalam direktori kerja anda.
2. Anda membawa berkas ke stage, menambahkan snapshotnya ke staging area.
3. Anda melakukan commit, yang mengambil berkas seperti yang ada di staging area dan menyimpan snapshotnya secara permanen ke direktori Git anda.

**Lalu bagaimana jika project yang sudah terintegrasi dengan git dapat diakses secara online? Salah satunya yaitu menggunakan Github.**

GitHub adalah platform khusus developer yang dibuat karena terinspirasi dari cara bekerja para programmer. Dari open source ke bisnis, kita bisa meng-host dan meninjau sebuah kode, mengelola

project, dan membangun perangkat lunak bersama 56 juta lebih pengguna di seluruh dunia. Gampangnya, GitHub merupakan manajemen project, sistem versioning code, sekaligus platform jaringan sosial bagi para developer seluruh dunia. Banyak sekali fungsi yang bisa digunakan oleh para developer, sehingga melalui platform ini memudahkan developer dalam mengembangkan sebuah karya. GitHub memberikan layanan cloud untuk menyimpan dan mengelola project/repository git. Karena bersifat online, kita meng-edit sebuah repository/project secara bersamaan dengan orang lain di tempat yang berbeda. Oleh karena itu, platform ini sangat membantu tim project dalam menyusun suatu folder yang berisikan files terkait pemrograman.

## Basic Git Commands

### 1. **git config**

Berfungsi untuk menambahkan konfigurasi git, umumnya dipakai untuk setting nama dan email secara global.

Usage: `git config --global user.name "[name]"`

Usage: `git config --global user.email "[email address]"`

### 2. **git init**

Berfungsi untuk inisiasi direktori sebagai *working directory* berbasis git

Usage: `git init [repository name]`

### 3. **git clone**

Berfungsi untuk clone/menggandakan repository, umumnya dipakai untuk clone repository dari server seperti Github, Bitbucket

Usage: `git clone [url]`

### 4. **git add**

Berfungsi untuk menambahkan kode yang dimodifikasi atau baru ditambahkan ke dalam staging area

Usage: `git add [file]`

Usage: `git add .`

### 5. **git commit**

Berfungsi untuk menambahkan snapshot permanent ke dalam git dari staging area

Usage: `git commit -m "[ Type in the commit message]"`

### 6. **git diff**

Berfungsi untuk menampilkan perbedaan file-file yang sudah diubah dengan yang sudah tersimpan dalam commit secara spesifik

Usage: git diff

Usage: git diff --staged

Usage: git diff [first branch] [second branch]

## 7. git reset

Untuk mengembalikan file-file ke kondisi sebelumnya, kemudian menghapus catatan history commit berikutnya

Usage: git reset [file]

Usage: git reset [commit]

Usage: git reset --hard [commit]

## 8. git status

Berfungsi untuk menampilkan perbedaan file-file yang sudah diubah dengan yang sudah tersimpan dalam commit secara umum, biasanya juga digunakan untuk menampilkan daftar lokasi file yang berubah atau file yang belum masuk dalam staging area

Usage: git status

## 9. git rm

Berfungsi untuk menghapus file dari staging area atau direktori kerja

Usage: git rm [file]

## 10. git log

Berfungsi untuk menampilkan daftar log yang sudah tersimpan yang ada di branch/tag/commit

Usage: git log

Usage: git log --follow[file]

## 11. git checkout

Berfungsi untuk pindah branch/tag/commit

Usage: git checkout [branch\_name]

Usage: git checkout -b [branch\_name]

## 12. git branch

Berfungsi untuk melihat daftar branch yang ada di repository

Usage: git branch [branch\_name]

## 13. git push

Berfungsi untuk mengupload commit yang ada di local ke remote repository (ex: Github)

Usage: git push

Usage: git push -u [remote\_name] [branch\_name]

## 14. git remote


Berfungsi untuk menambah dan atau menghapus lokasi remote repository

Usage: git remote -v

Usage: git remote add [remote\_name] [url repository]

Usage: git remote set-url [remote\_name] [url repository]

Daftar git commands dapat diakses di dokumentasi resmi git, disini <https://git-scm.com/docs> dan lalu untuk implementasi penggunaan git secara detailnya, kalian dapat melihat video ini

 Jago Git Dalam 2 Jam

## JIRA Confluence



Confluence adalah aplikasi buatan Atlassian yang ditujukan untuk pendokumentasian. Confluence mempunyai format serupa Wikipedia sehingga memungkinkan kita untuk membuat dokumentasi yang jelas dan terstruktur ketimbang jika kita menggunakan Google Docs atau Microsoft Word. Confluence juga memiliki berbagai macam tipe dokumen untuk keperluan yang berbeda-beda, seperti product requirement atau meeting notes. Selain itu, dengan aplikasi ini kamu juga dapat melihat dua versi langsung, yaitu versi sebelum dan setelah diubah. Hal ini membuat pengguna sangat mudah untuk melihat perkembangan yang terjadi di dalam konten.

Setelah mengetahui kegunaannya, saatnya kamu memahami beberapa fitur yang disediakan oleh aplikasi ini. Fitur-fitur dari Confluence ini di antaranya adalah:

### 1. Kolaborasi editor

Dengan Confluence kamu dapat melakukan pengeditan secara bersamaan dengan rekan kerjamu. Hal ini membuat berbagai perubahan yang terjadi dapat dilihat secara bersamaan. Hal ini tentu memudahkan ketika sedang membuat proyek bersama atau membuat kerangka strategi. Pasalnya, orang lain, misalnya dari tim lain, dapat langsung mengeditnya dan kamu bisa melihat perubahannya dalam sekejap.

### 2. Dapat disinkronkan ke handphone

Fitur lainnya yang disediakan oleh Confluence adalah kamu dapat menyinkronkan aplikasi ini ke dalam handphone Android ataupun iOS. Hal ini memungkinkan kamu untuk bekerja dari mana saja. Selain itu, kamu juga dapat melacak aktivitas tim, memberikan feedback, serta tetap sinkron antara handphone dan desktop.



### 3. Membuat grup

Mungkin ini adalah salah satu fitur yang tidak dimiliki oleh Google Docs. Confluence membuat penggunanya dapat membuat grup sehingga dapat mengatur pekerjaan dengan baik. Kamu bisa membuat grup antartim ataupun ketika ada proyek besar lewat aplikasi ini. Hal ini tentu memudahkan kamu dalam mengatur pekerjaan sebaik mungkin.

### 4. Feedback yang cepat

Confluence memungkinkan penggunanya untuk memberikan feedback yang cepat setelah melakukan perubahan dalam suatu dokumen. Di dalam aplikasi ini terdapat fitur komentar. Dengan begitu, pengguna hanya tinggal memberikan komentar mengenai dokumen apa yang perlu diubah. Tidak hanya memberikan feedback yang cepat, Confluence juga dapat membuat penggunanya memberikan feedback secara langsung melalui berbagai file, seperti PDF, spreadsheet, dan lain-lain.

---

## MATERI PRAKTIKUM

1. <https://yunwuxin1.gitbooks.io/git/content/id/b46a67ff77cde65a0bc213ca0d452e25/9927549e7f2237140af757a0a6d60075.html>
2. <https://www.dicoding.com/blog/apa-itu-github/>
3. <https://git-scm.com/docs/git>
4. <https://www.dicoding.com/blog/perbedaan-git-dan-github/>
5. <https://glints.com/id/lowongan/apa-itu-git-adalah/>
6. <https://glints.com/id/lowongan/confluence-adalah/>
7. <https://www.atlassian.com/software/confluence>
8. <https://www.petanikode.com/git-untuk-pemula/>
9. <https://www.youtube.com/watch?v=KGSfUgaiVNI&t=3188s>

---

## LEMBAR KERJA

### KEGIATAN 1

Buatlah repository di Github dengan nama repo menyesuaikan dari nama project atau pengembangan perangkat lunak yang telah kalian rencanakan dari modul 1 sampai 5. Kemudian upload semua file yang berhubungan dengan pengerjaan tugas dari modul 1 hingga 5 menggunakan konsep git yang kalian pahami. **(minimal add, commit, checkout, branch, pull, push)**

NB:

1. akan mendapatkan nilai lebih jika menerapkan command-command lainnya dan atau menerapkan dan memahami fitur pull request.

## KEGIATAN 2

Buatlah sebuah JIRA Confluence dan dokumentasikan seluruh hasil pekerjaan modul kalian dari modul 1 hingga modul 6 ke dalam Confluence dengan ketentuan sebagai berikut:

1. Setiap dokumentasi modul harus dibuat dalam **Page** terpisah.
  2. Penulisan dokumentasi pada setiap **Page** harus kreatif, tidak monoton, dan menarik untuk dibaca.
  3. Dokumentasikan juga peran setiap anggota kelompok dari manager hingga closing secara lengkap dan menarik pada setiap **Page** yang dibuat.
- 

## RUBRIK PENILAIAN

Kegiatan 1 : 40%

Kegiatan 2 : 60%