**-- GIT DAN GITHUB –**

Git dan Github merupakan dua hal yang berbeda.

Kita bisa saja bekerja dengan git atau tanpa github, ataupun sebaliknya. Kita bisa menggunakan github meski dikomputer kita tidak terinstall git. Penggunaan keduanya bisa benar-benar terpisah atau bisa juga digabungkan.

--- GIT ---

Apa itu GIT?

Sesuatu yg disebut VCS = Version Control System

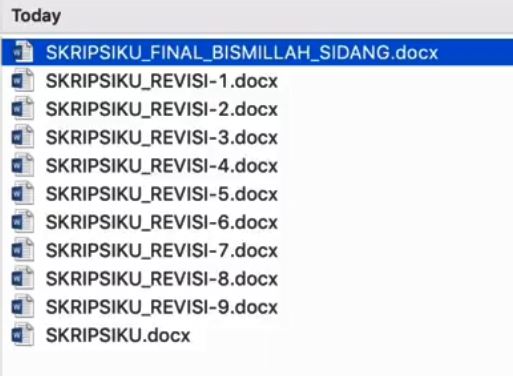
Apa itu VCS?

“Version Control System, disebut juga revision control system atau source code management adalah system yang mengelola perubahan dari sebuah dokumen, program computer, website dan kumpulan informasi lain.” By Wikipedia

Problems?

Ada beberapa masalah yg sering kita hadapi ketika tidak menggunakan VCS.

1. Ketika mengerjakan tugas, skripsi dll :



File yang dikerjakan akan menumpuk karena mengalami beberapa kali revisi atau perubahan.

1. Ketika kita bekerja secara bersama atau tim, maka akan sangat merepotkan jika ingin mengumpulkan/menggabungkan hasil dari pekerjaan kita.
2. Tidak bisa sharing/berbagi document secara cepat meski dibatasi oleh waktu dan jarak

Defenisi dari kesimpulan :

Version Control System :

* sebuah system yang menyimpan ‘rekaman/ snapshot’ perubahan pada source code
* Memungkinkan bekerja berkolaborasi dengan lebih baik.
* Mengetahui siapa yang melakukan dan kapan sebuah perubahan terjadi
* Memungkinkan kita untuk kembali ke keadaan sebelum perubahan(checkout).

Ada beberapa jenis version control system :

1. GIT
2. Subversion
3. Mercurial
4. CVS

# GIT

Git adalah “Sebuah VCS terdistribusi untuk mengelola perubahan file di dalam folder” atau

“Sebuah Software terdistribusi untuk mengelola perubahan file di dalam folder (Repository / Repo)“ By Wikipedia

“Riwayat perubahan file disimpan menggunakan serangkaina **commit**”. Commit ini berfungsi utk merekam/snapshot setiap perubahan yg dilakukan baik itu menambah, menghapus, menyimpan dll”

Semuanya akan disimpan ke dalam computer kita secara local.

#Contoh hal yg pertama dilakukan pada GIT

1. Inisialisasi project
2. Menambahkan view untuk login dan registrasi
3. Membuat controller user
4. Membuat model Auth

Branch 🡺 Melakukan percabangan terhadap perubahan dr repo kita

Merge 🡺 Proses menggabungkan dua buah buah branch menjadi satu pada repo setelelah melakukan beberapa kali commit.

## GITHUB ##

Github adalah “Layanan cloud untuk meyimpan dan mengelola project / repo git”

Sederhananya github adalah sebuah website yg didalamnya menggunakan git.

--- Gabungan antara GIT & GITHUB ---

Ketika kita mengistall Git secara local dan menggunakan Github secara cloud maka akan sangat terasa bermanfaat ketika kita gabungin keduanya.

Push 🡺 Proses mengirimkan source code/project/document kita ke repo Github

Pull 🡺 Proses menerima atau mengambil source code/project/repuokita ke computer local(Git)

Tapi syaratnya adalah :

“kita harus membuat github menjadi sesuatu yg biasa disebut ‘Remote’, yaitu sebagai sumber dari repo kita”

Maka setelah remote dilakukan barulah kita dapat mengambil atau menerima repo dari github untuk di clone ke computer local(git) kita.

Apakah yg di push dan di pull? -> Yaitu commit.

Ada beberapa jenis sama seperti layanan Github :

- Bitbucket

- Gitlab

# Istilah-istilah pada GIT

Repo == folder project kita

Commit == rekaman / snapsot dari repo kita

Hash == penanda unik pada sebuah commit

Checkout == berpindah ke sebuah commit

Branch == cabang bebas dari sebuah commit

Merge == menggabungkan branch

Remote == sumber yg memiliki repo

Clone == mengambil repo dari remote

Push == mengirimkan commit ke repo

Pull == mengambil commit dari repo

1. Branch pada Github

Proses melakukan percabangan atau memisahkan diri dari master branch(jalur utama).

\*\* Branching

-Membuat GIt Branch

- Membuat snapshot tanpa mengganggu jalur utama(Master Branch)

- Fitur Experimental

- 2 Orang mengerjakan repo yang sama.

Compare and Pull Request => proses menggabungkan 2 buah branch/commit menjadi satu ke jalur utama(masterbranch)

Checkout => Berpindah ke branch / commit yg lain.

Merge Conflict => Baris yang sama diubah oleh 2 branch yang berbeda

B. Fork / Forking pada Github

- Membuat ‘Copy / duplikat’ dari repo orang lain (beserta history-nya).

- Jembatan antara repo original dan duplikatnya

- Memodifikasi terhadap repo original

- Berkontribusi Pada repo orang lain

- Fork != Clone

##.. G I T ..##

Download App Git di <https://git-scm.com/> dan Ebook di <https://git-scm.com/book/id/v2> .

* Git Command (local)
* $ git init -> utk menginisialisasi/memantau repo git di komp kita
* $ git add <file(s)> >> utk menambah file (staging area)
* $ git status -> utk mengetahui status repo saat ini ?= + -
* $ git commit ->
* $ git config -> utk memasukkan configurasi ke dalam git
* $ git branch -> utk membuat branch baru
* $ git help -> utk menampilkan perintah bantuan
* $ git add . -> utk menyimpan semua perubahan file kedalam repo git
* $ git commit –m “text” -> melakukan history perubahan pada commit
* $ git log -> utk menampilkan seluruh history perubahan yg pernah dilakukan dgn commit. Utk keluar dr git log tekan tombol Q.
* $ git log –“angka” -> utk menampilkan beberapa(angka) history perubahan yg prnh dilakukan
* $ git log – style.css -> utk menampilkan history perubahan yg pernah dilakukan meskipun file sudah dihapus.
* $ git commit –am <text> -> ini gabungan antara $git add . dan $git commit –m . yang bekerja sekaligus.
* $ alias graph="git log --all --decorate --oneline --graph" -> utk melihat daftar branch, commit, dan aktivitas di jalur utama.
* $ git checkout <nama branch> -> utk berpindah ke branch/commit lainnya dgn petunjuk <Head>

# ada dua jenis Merge

1. Fast Forward
2. Three-way merge

* $ git merge <nama branch> -> utk menggabungkan branch yg berbeda menuju branch(master)
* $ git branch --merged -> utk melihat branch mana yg baru saja digabungkan.
* $ git branch –-merged -> utk mengetahui branch yg baru saja digabungkan
* $ git branch –d <nama branch> -> utk menghapus branch yg sudah berhasil di merge ke jalur utama(master).
* git checkout 082cb4d(ambil 7 digit no. hash commit dr depan) -> utk berpindah/mundur ke sebuah commit atau keadaan dan ketika terjadi merge conflict. detached HEAD (head yg lepas dari branchnya).
* 3 area pada repo git
* Working tree = repo beserta isi seperti biasanya()
* Working area = repo beserta isi seperti biasanya(tempat kita bekerja)
* Staging area = utk memberitahu git bahwa kita sudah melakukan perubahan
* History = utk menjadikan commit ketika ada perubahan

# Git Remote

Git Remote yaitu Membuat duplikat repo local(komputer) kita ke suatu tempat lain. Seperti menyimpan ke internet(github), jadi disini bisa dikatakan github adalah remote kita yg akan terhubung ke repo local kita.

Kedepepannnya ketika kita melakukan perubahan kode di local kita(new commmit) setelah itu kita akan mengirim perubahan tsb ke remotenya(github) biasa disebut dengan ‘Push’. Atau sebalikknya. Kalau ada perubahan didalam remotenya(github), misal ada orang(teman) mengerjakan project yg sama dan terhubung ke remote yg sama kemudian ketika repo di remotenya berubah(ada commit baru) maka dengan begitu kita bisa menarik perubahannya dengan melakukan yg namanya ‘fetch’ atau ‘Pull’.

Push => mengirim perubahan yg terjadi pada repo local ke dalam remotenya(github).

Pull => menarik/menerima perubahan yg terjadi pada remotenya(github) ke dalam repo local kita.

Kita tahu bahwa git dilocal kita juga bisa terhubung ke remote/tempat lain juga, misal terhubung ke GitLab, BitBucket. Bahkan satu repo local bisa dapat terhubung ke beberapa remote sekaligus(repolokal->Github->GitLab->BitBucket).

Atau misalnya didalam github itu sendiri, misal kita sedang bekerja didalam github dan kita melihat ada repo orang lain yg menurut kita bagus dan kita perlu itu maka kita bisa melakukan ‘Fork’.

Fork itu artinya repo orang lain yg kita fork bisa juga menjadi remote untuk repo github kita.

Misal repo sandhikagalih adalah remote utk webprogrammingunpas.

Misal dikantor ada beberapa computer yg terhubung ke satu server, sehingga server tsb bisa menjadi remote dari repo kita.

Misal pada komputer sendiri. Didalam komputer kita memiliki sebuah folder yg nantinya kita memiliki duplikatnya ditempat lain di dalam satu computer tapi di folder yg lain. Dan folder tsb bisa juga kita jadikan sebagai remote.

Kesimpulannya kita bisa menyimpan source code, tugas, pembelajaran dan dokumen lain pada repo local langsung ke dalam remote(kegithub).

\*Account hak akses user git bash pada computer kita.

Untuk Ganti user per satu project :

- git config --local user.name "irwanto siregar"

- git config --local user.email [irwantosiregar@ymail.com](mailto:irwantosiregar@ymail.com)

Untuk Ganti user secara global pada setiap project :

- git config --global user.name "irwanto siregar"

- git config --global user.email [irwantosiregar@ymail.com](mailto:irwantosiregar@ymail.com)

Clonning(Colone or download) yaitu melakukan duplikat sebuah repo yg ada dlm akun github kita ke dalam computer yg akan kita jadikan sebagai repo local.

Ada dua cara utk cloning pada github

1. Clone dengan HTTPS(Colone with https). Yg biasa dilakukan dan mudah.
2. Clone dengan SSH(Use SSH).pertama harus lakukan konfigurasi key nya dulu di komputer kita. Key Berfungsi Utk menghubungkan antara remote(github) dengan computer local.

* $ git clone <alamat url> -> utk menduplikat/copy repo dari github ke dalam computer local kita.
* $ git remote -> utk mengetahui nama branch dari sumber repo kita(Default : Origin)
* $ git remote -v-> utk menampilkan alamat url asal dari sumber repo kita(yaitu, github).
* $ git remote add sandhikagalih <alamat url(<https://github.com/sandhikagalih/NGOBAR.git>)> --> utk menambahkan remote(sumber repo->utama) baru ke dalam git komp local kita, dimana sebelumnya dilakukan fork terhadap repo di github. Misal akun github irwantosiregar melakukan fork terhadap salah satu repo dari akun sandhika galih, hal ini biasa disebut “Multiple Remotes”.
* $ git fetch <nama\_remote(sandhikagalih)> -> utk mengambil/menarik history commit dari sumber repo yg difork(sandhikagalih) ke dalam git komp local kita.
* $ git checkout -b <nama branch> -> utk membuat branch baru pada repo dan kemudian akan langsung masuk ke branch tsb.
* $ git branch -d <nama branch(features)>-> utk menghapus branch cadangan(features) pada repo git local kita.
* $ git push origin --delete <nama branch(features)>-> utk Menghapus 'branch cadangan(features)' dengan melakukan push/mengirim perubahan 'delete' menuju ke repo github kita.

# Git REBASE

Apa itu Rebase?

Rebase merupakan salah satu cara kita utk memilih yg namanya Git Workflow.

Seperti di video-video sebelum ini cara kerja kita menggunakan git/git workflow kita adalah menggunakan

teknik merging, saat kita berkolaborasi dgn team atau bahkan dgn repo kita sendiri saat kita mau menambahkan fitur baru

dan fitur itu mau kita terapkan kemaster itu kita lakukan merging. Nah Rebase adalah cara lain didalam pengerjaan git nya

yg tidak menggunakan merging, jadi ini workflow yg berbeda. Berikutnya kita akan melihat apasih bedanya antara merging dan rebase ini(Merge Vs Rebase).