



Git Remote

Eko Kurniawan Khannedy

Eko Kurniawan Khannedy

- Technical architect at one of the biggest ecommerce company in Indonesia
- 10+ years experiences
- www.programmerzamannow.com
- youtube.com/c/ProgrammerZamanNow





Eko Kurniawan Khannedy

- Telegram : [@khannedy](https://t.me/khannedy)
- Facebook : fb.com/ProgrammerZamanNow
- Instagram : instagram.com/programmerzamannow
- Youtube : youtube.com/c/ProgrammerZamanNow
- Telegram Channel : t.me/ProgrammerZamanNow
- Email : echo.khannedy@gmail.com



Sebelum Belajar

- Git Dasar
- Git Branching



Agenda

- Pengenalan Git Remote
- Git Server
- Git Server Repository
- Push
- Fetch
- Pull
- Dan lain-lain

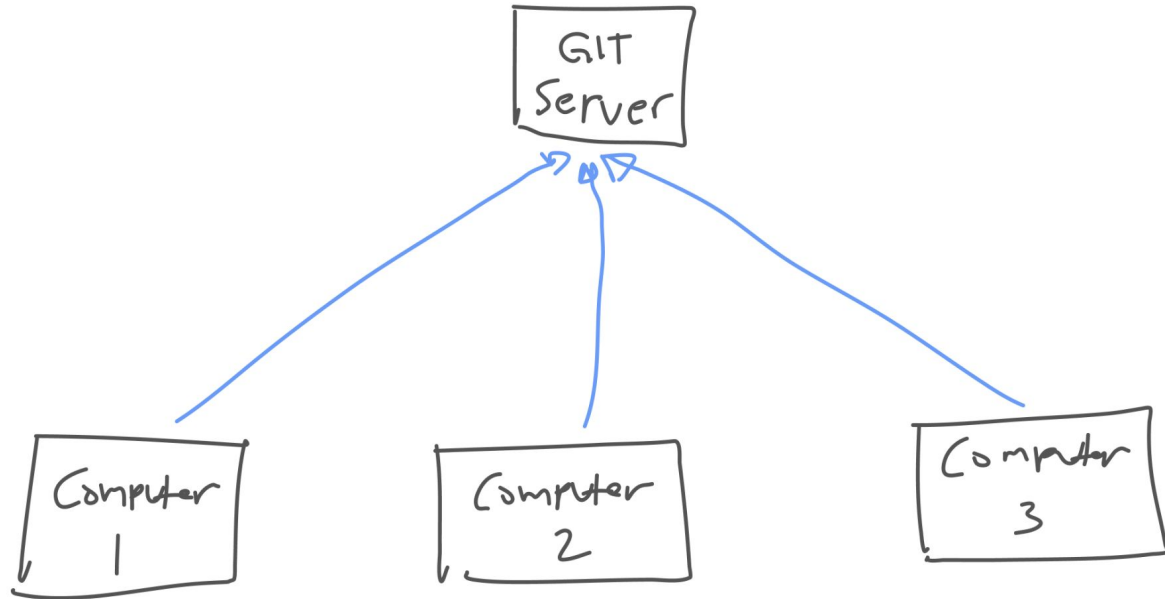
Pengenalan Git Remote



Git Remote

- Sebelumnya kita sudah belajar semua fitur yang terdapat di Git
- Namun semua fitur yang kita bahas, masih kita lakukan di local komputer kita masing-masing
- Saat nanti bekerja dengan tim, atau berkolaborasi, kita tidak hanya akan menyimpan git repository pada satu komputer saja
- Tiap anggota tim, kemungkinan akan meng-copy git repository yang kita buat, dan juga melakukan kontribusi ke git repository nya
- Git mendukung centralized server, dimana kita bisa menyimpan perubahan git repository di komputer ke Git server
- Karena Git merupakan distributed version control, walaupun kita gunakan Git Server, tapi kita tidak wajib terkoneksi ketika ingin mengelola git repository
- Hanya ketika kita ingin mengirim perubahan atau mengambil perubahan, baru kita butuh terkoneksi ke Git Server

Diagram Git Remote

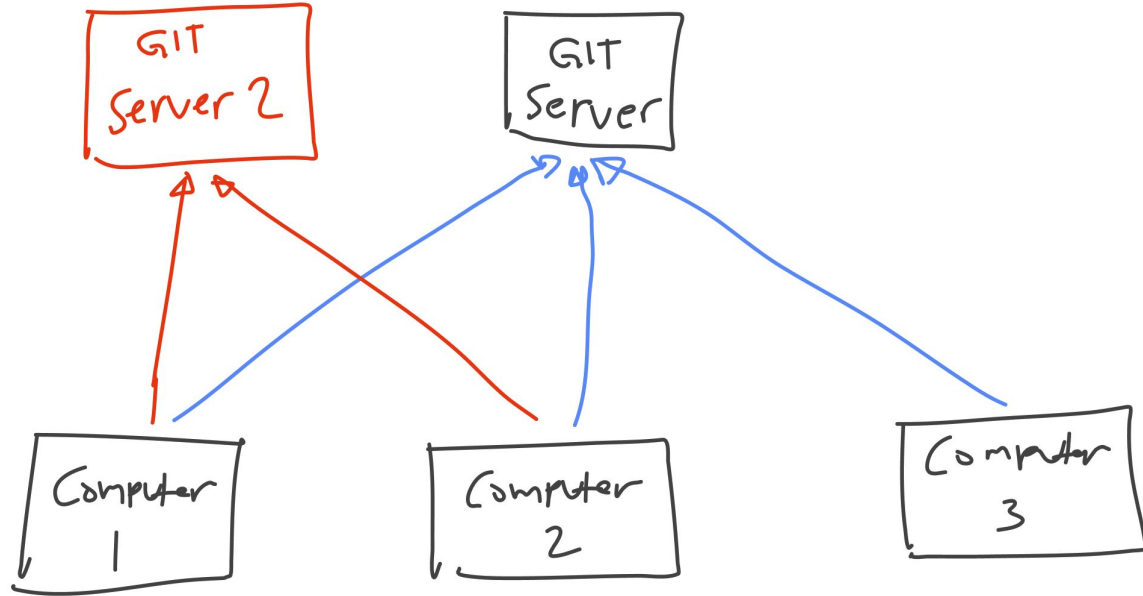




Multiple Git Server

- Karena Git merupakan distributed system, jadi kita tidak hanya bisa menggunakan satu Git Server
- Jika kita mau, kita juga bisa menggunakan beberapa Git Server sekaligus ketika membuat git repository
- Nanti kita bisa memilih, mau mengirim perubahan atau mengambil perubahan ke Git Server yang mana

Diagram Multiple Git Server



Git Server



Git Server

- Git Server adalah Server yang bisa digunakan untuk menyimpan repository git yang sudah kita buat di local
- Ada banyak sekali Git Server yang tersedia, baik itu yang gratis atau pun berbayar
- Kelebihan ketika kita menggunakan Git Server adalah, data repository aman tersimpan di Server, sehingga jika kita hapus git repository nya di local, repository akan tetap aman di Server



Contoh Git Server

- Ada banyak sekali Git Server yang tersedia, kita bisa gunakan salah satunya
- Github : <https://github.com/>
- Gitlab : <https://gitlab.com/>
- Bitbucket : <https://bitbucket.org/>
- Dan lain-lain



Github

- Pada kelas ini, kita akan menggunakan Git Server yang paling populer dan banyak digunakan, yaitu Github
- Github memiliki versi gratis dan versi berbayar
- Banyak sekali project Open Source yang menggunakan Github untuk menyimpan git repository nya
- Silahkan membuat akun terlebih dahulu di Github
- <https://github.com/>

Git Server Repository



Git Server Repository

- Tiap Git Server biasanya memiliki fitur untuk membuat Git Repository
- Tidak wajib kita harus membuat Git Repository di local dulu jika ingin membuat Git Repository di Git Server
- Sekarang silahkan buat Git Repository di Github dengan nama belajar-git-remote
- <https://github.com/new>

SSH



SSH

- SSH adalah singkatan dari Secure Shell
- SSH merupakan protokol jaringan untuk komunikasi jaringan yang aman dan terenkripsi
- Pengguna sistem operasi Linux atau Mac biasanya sudah sangat familiar dengan SSH
- SSH merupakan aplikasi berbasis terminal
- Di Linux dan Mac, SSH sudah terinstall secara otomatis, sedangkan di Windows, ketika kita menginstall Git, secara otomatis Git akan menginstall SSH juga



Git SSH

- Git sendiri memiliki beberapa mekanisme untuk berkomunikasi dengan Git Server, seperti http dan SSH
- Pada kelas ini, kita akan menggunakan SSH untuk berkomunikasi dengan Git Server
- Hal ini karena SSH merupakan protokol yang direkomendasikan ketika berkomunikasi dengan Git Server



SSH Key

- Hal pertama yang perlu kita lakukan ketika menggunakan SSH adalah, membuat SSH Key
- SSH Key merupakan kunci yang digunakan untuk autentikasi ke SSH Server
- Untuk membuat SSH Key, kita bisa gunakan perintah ssh-keygen di terminal
- Setelah selesai, maka secara otomatis akan terdapat 2 key di local kita, yaitu private key dan public key
- Kita bisa melihatnya di dalam folder .ssh di Home directory kita
- File id_rsa adalah private key, dan id_rsa.pub adalah public key



Menambahkan SSH Public Key ke Github

- Setelah kita membuat SSH Key, selanjutnya kita perlu mendaftarkan SSH public key ke Github
- Hal ini dilakukan, agar ketika kita nanti terkoneksi ke Git Server di Github, kita tidak perlu melakukan autentikasi lagi
- <https://github.com/settings/keys>



Test SSH ke Github

- Selanjutnya untuk memastikan apakah kita sudah bisa terkoneksi ke Github menggunakan SSH, kita bisa gunakan perintah :
`ssh -T git@github.com`

```
→ ~ ssh -T git@github.com
```

```
Hi khannedy! You've successfully authenticated, but GitHub does not provide shell access.
```

```
→ ~
```

Remote Repository



Remote Repository

- Ketika kita membuat Git Project, secara default, Git tidak tahu tentang Remote Repository
- Kita perlu memberi tahu ke git project yang sudah kita buat tentang lokasi git repository



Menambah Remote Repository

- Untuk menambah remote repository, kita bisa gunakan perintah :
`git remote add nama ssh-url`
- Salah satu kebiasaan di Git, biasanya memberi nama untuk remote repository dengan nama origin



Melihat Remote Repository

- Untuk melihat remote repository yang ada di git project, kita bisa gunakan perintah :
git remote
- Untuk melihat URL detail remote repository, kita bisa gunakan perintah
git remote get-url nama



Menghapus Remote Repository

- Untuk menghapus remote repository, kita bisa gunakan perintah :
`git remote rm nama`

—

Push



Push

- Walaupun kita sudah menyimpan perubahan di Git Project di local, tapi tidak secara otomatis akan di sync dengan Remote Repository
- Hal ini karena sejak awal Git di desain sebagai distributed version control, artinya kita bisa melakukan perubahan dimanapun dan kapanpun, tanpa harus terkoneksi ke Git Server
- Oleh karena itu, jika kita ingin mengirim perubahan yang terjadi di Git Project di local kita, kita perlu mengirimnya secara manual ke Git Server
- Untuk mengirim perubahan di local ke git server, kita bisa gunakan perintah yang bernama push



Push Branch

- Untuk mengirim perubahan branch ke remote repository dengan nama branch sama:
`git push namaremotelocalbranch`
- Untuk mengirim perubahan branch ke remote repository dengan nama branch yang berbeda, kita bisa gunakan perintah :
`git push namaremotelocalbranch:remotebranch`



Push Semua Branch

- Jika kita ingin mengirim semua perubahan di semua branch ke remote repository, kita bisa gunakan perintah :
`git push origin --all`



Menghapus Branch

- Perintah Push juga bisa digunakan untuk menghapus branch yang ada di remote repository
- Kita bisa gunakan perintah
`git push --delete namaremove namabbranch`
- Perlu diingat, menghapus remote branch bukan berarti menghapus branch di local, jadi jika kita ingin hapus di local, kita harus lakukan secara manual

Clone



Clone

- Apa yang harus dilakukan jika misal kita ingin download project Git yang ada di Server ke komputer baru?
- Hal ini dinamakan perintah clone
- Dengan perintah clone, kita bisa download project di remote repository ke local dan secara otomatis di download sebagai git project



Melakukan Clone

- Untuk melakukan clone, kita bisa gunakan perintah :
`git clone urlremoterepository`
- Secara default, clone akan membuat project dengan nama folder sama dengan nama project remote repository
- Jika kita ingin melakukan clone, dengan nama folder yang berbeda dengan nama project remote repository, kita bisa gunakan perintah :
`git clone urlremoterepository namafolder`



Default Hasil Clone

- Default clone akan berisi remote repository origin ke git remote repository yang kita clone
- Default clone akan berisikan branch utama di remote repository



Tugas

- Simulasikan terdapat dua pengguna
- Clone project Git di remote di folder yang berbeda

Remote Branch



Melihat Remote Branch

- Secara default, saat kita melakukan clone, tidak semua remote branch akan di buat di local git project
- Jika kita ingin melihat semua daftar branch yang ada di remote repository, kita bisa gunakan perintah :
`git branch -r`
- Atau jika kita ingin melihat semua branch di local dan remote, kita bisa gunakan perintah :
`git branch -a`



Membuat Branch dari Remote Branch

- Karena secara default hanya branch utama yang buat di local git project ketika kita clone, kadang kita ingin membuat local branch yang berisikan data dari remote branch
- Untuk melakukan hal itu, kita bisa gunakan perintah :
`git checkout -b localbranch namaremove/remotetbranch`



Fetch



Fetch

- Fetch adalah perintah di Git yang digunakan untuk mendapatkan perubahan terakhir dari remote repository
- Hal ini berguna ketika misal kita ingin mengetahui perubahan apa aja yang sudah terjadi pada remote repository, mungkin rekan kerja kita sudah menambah perubahan ke remote repository
- Fetch itu tidak akan mengubah isi local Git Project kita



Melakukan Fetch

- Untuk melakukan fetch untuk semua branch di remote repository, kita bisa gunakan perintah :
`git fetch namareMOTE`
- Atau jika kita ingin melakukan fetch untuk branch tertentu saja di remote, kita bisa gunakan :
`git fetch namareMOTE remotebranch`



Tugas

- Lakukan commit dan push di user pertama
- Lakukan fetch di user kedua
- Di user kedua Lakukan git diff dari branch local dan branch remote

Pull



Pull

- Jika fetch hanya melakukan untuk mendapatkan perubahan di remote repository tanpa mengubah local repository
- Dengan menggunakan pull, kita akan mengambil perubahan yang ada di remote repository dan disimpan di local repository
- Hati-hati jangan sampai terjadi merge conflict, jika terjadi merge conflict, kita harus perbaiki secara manual seperti yang sudah kita pelajari di materi merge
- Saat kita melakukan pull, secara otomatis kita akan melakukan fetch terlebih dahulu



Melakukan Pull

- Untuk melakukan pull ke local branch saat ini, kita bisa gunakan perintah `git pull namareMOTE namareMOTEbranch`



Tugas

- Lakukan commit dan push di user pertama
- Lakukan pull di user kedua
- Bandingkan local branch dan remote branch di user kedua

Tag



Tag

- Saat kita melakukan push data ke remote repository, data Tag tidak akan di push juga ke remote repository
- Jika kita ingin mengirim data Tag juga, kita perlu melakukan push tag tersebut



Mengirim Tag

- Untuk mengirim Tag ke remote repository, kita bisa gunakan perintah `git push namareMOTE namatag`
- Atau jika kita ingin mengirim semua Tag ke remote repository, kita bisa gunakan perintah `git push namareMOTE --tags`



Mengambil Tag

- Untuk mengambil Tag, kita bisa gunakan perintah
`git fetch namareMOTE namatag`
- Atau jika kita ingin mengambil semua Tag, kita bisa gunakan perintah
`git fetch namareMOTE`



Menghapus Tag

- Saat kita menghapus Tag di local repository, data tag tersebut tetap ada di remote repository
- Jika kita ingin menghapus Tag di remote repository, kita bisa gunakan perintah :
`git push --delete namaremove namatag`

Pull Request



Pull Request

- Saat kita bekerja dengan tim, biasanya saat melakukan perubahan kita akan membuat di branch yang berbeda dengan branch utama
- Setelah selesai, kita bisa melakukan yang namanya merge ke branch utama
- Namun, beberapa fitur di remote git repository mendukung fitur yang namanya Pull Request atau Merge Request
- Fitur ini sangat cocok untuk melakukan review otomatis di remote git repository, tanpa harus manual melakukan review dengan git diff

Merge Conflict



Merge Conflict

- Ketika melakukan Pull Request, kadang sering terjadi Merge Conflict yang menyebabkan tidak bisa lakukan proses merge otomatis
- Pada kasus seperti itu, kita perlu melakukan merge dari branch asal nya terlebih dahulu, lalu memperbaiki conflict nya

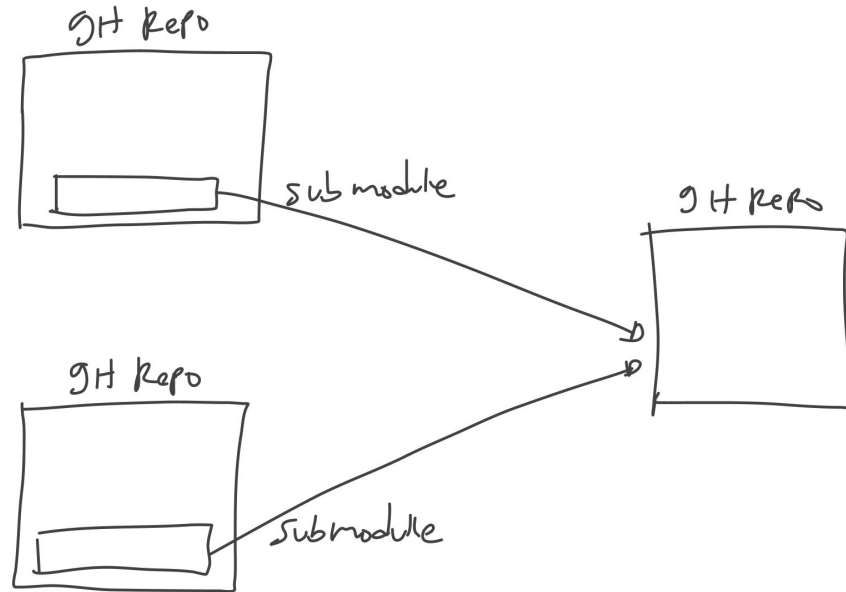
Submodule



Submodule

- Saat kita membuat project aplikasi, kadang kita ingin membuat library yang berisi kode program yang ingin digunakan di beberapa aplikasi
- Pada kasus seperti ini, fitur Git Submodule sangat bermanfaat
- Git Submodule memungkinkan kita menambahkan git repository ke git repository yang lain, atau dinamakan submodule
- Dengan menggunakan submodule, kita bisa melakukan management git repository terpisah, namun kita bisa mengakses git repository lain

Diagram Submodule





Menambah Submodule

- Untuk menambah submodule ke dalam git repository, kita bisa gunakan perintah :
`git submodule add urlgitrepo namafolder`
- Saat kita menambah submodule, secara otomatis semua git repository submodule tersebut akan di clone ke dalam folder yang kita tentukan
- Bagaimana git mendeteksi bahwa folder itu adalah submodule?
- Git akan mendeteksinya dari file `.gitmodules`



Submodule Repository

- Saat kita masuk ke folder submodule, sebenarnya akan seperti berada di project git repository submodule tersebut
- Kita bisa melakukan management git repository di submodule tersebut, seperti melakukan commit, mengelola branch, dan lain-lain
- Namun tidak disarankan untuk melakukan perubahan di submodule, lakukanlah perubahan langsung di git repository nya, anggap submodule hanya digunakan untuk menduplikasi perubahan yang terjadi di git repository sumber asli nya



Update Submodule

- Jika terjadi perubahan di repository submodule, kita bisa mendapat update perubahannya dengan menggunakan perintah :
`git submodule update --remote namafolder`
- Atau jika ingin semua submodule di update, kita bisa gunakan perintah :
`git submodule update --remote`
- Atau jika ingin melakukan update manual, kita bisa masuk ke dalam folder submodule, lalu melakukan pull ke git remote submodule tersebut



Clone Submodule

- Saat kita melakukan clone sebuah git repository yang terdapat submodule nya, tidak secara default repository submodule akan di clone juga
- Setelah kita melakukan clone, kita perlu melakukan perintah init untuk mengaktifkan module nya dengan perintah :
`git submodule init`
- Perintah ini akan membaca dari file `.gitmodule` lalu membuat submodule sesuai dengan file tersebut
- Selanjutnya untuk download submodule git repository nya, kita bisa gunakan perintah `submodule update`



Mengubah Branch Submodule

- Untuk mengubah branch di submodule, sebenarnya kita bisa lakukan secara manual menggunakan perintah `git branch`
- Namun hal ini tidak akan membuat konsisten, karena bisa jadi tiap orang bisa berbeda-beda branch submodule nya
- Agar sama, kita bisa gunakan perintah `submodule set-branch` untuk mengubah branch submodule, sehingga secara otomatis semua git repo akan menggunakan branch yang sudah ditentukan
- Caranya kita bisa gunakan perintah :
`git submodule set-branch --branch namabbranch namafolder`



Menghapus Submodule

- Untuk menghapus submodule, kita bisa hapus folder submodule nya, lalu ubah file `.gitmodules` nya



Fork



Fork

- Salah satu git workflow yang sebelumnya kita jelaskan di kelas Git Branching adalah Forking Workflow
- Forking Workflow biasanya digunakan ketika kita ingin kontribusi ke project opensource, dimana kita tidak punya akses langsung ke git repository nya, sehingga kita melakukan forking (duplikasi) git repository ke tempat kita
- Pada kasus menggunakan github, misal kita melakukan forking dari ProgrammerZamanNow ke github profile kita



Pull Request

- Semua perubahan nanti dilakukan di git repository profile kita
- Dan setelah selesai, kita bisa melakukan pull request ke repository sumber forking kita
- Seperti inilah cara kerja ketika kita berkontribusi ke project opensource

Materi Selanjutnya



Materi Selanjutnya

- Mulai gunakan Git untuk semua Project Aplikasi