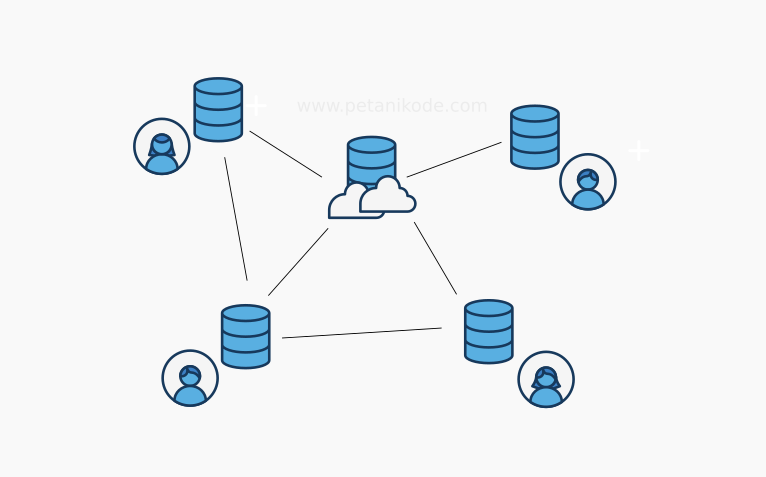
**BELAJAR GIT**

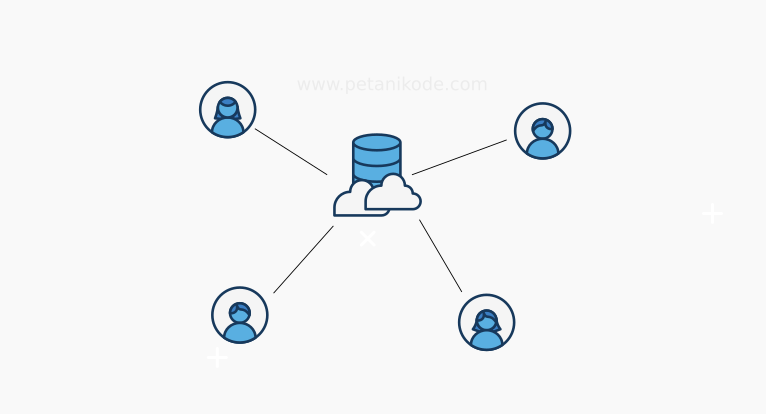
1. **Deskripsi Git**

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.



Semua user yang terlibat dalam pengkodean proyek akan menyimpan database Git, sehingga akan memudahkan dalam mengelola proyek baik online maupun offline. Dalam Git terdapat merge untuk menyebutkan aktifitas penggabungan kode.

Sedangkan pada VCS (Version Control System) yang terpusat… database disimpan dalam satu tempat dan setiap perubahan disimpan ke sana.



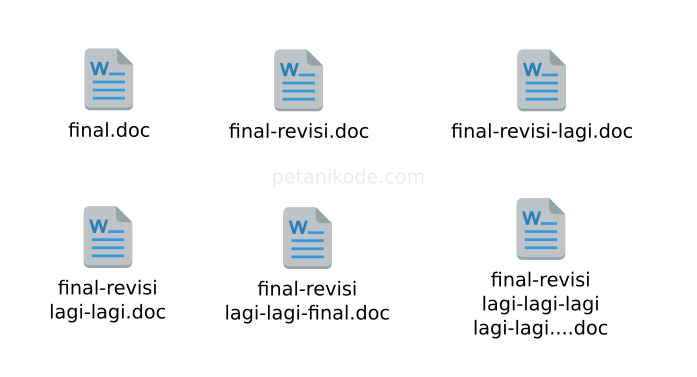
VCS terpusat memiliki beberapa kekurangan:

* Semua tim harus terkoneksi ke jaringan untuk mengakses source-code;
* Tersimpan di satu tempat, nanti kalau server bermasalah bagaimana?

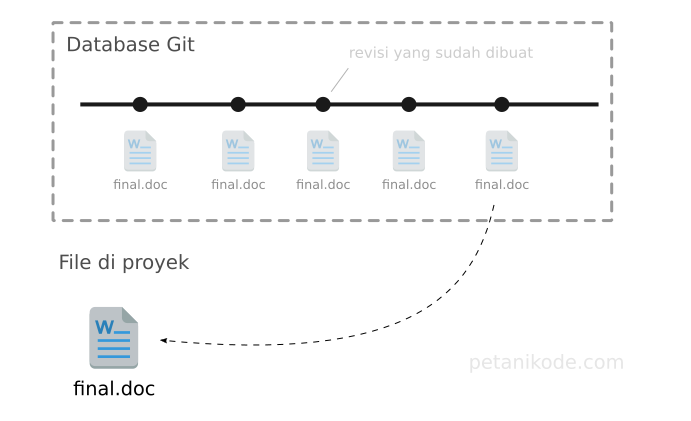
Karena itu, Git hadir untuk menutupi kekurangan yang dimiliki oleh VCS terpusat.

Git sebenarnya akan memantau semua perubahan yang terjadi pada file proyek. Lalu menyimpannya ke dalam database.

Sebelum menggunakan Git :



Setelah menggunakan Git :



Saat kita ingin menyimpan semua perubahan pada file, biasanya kita membuat file baru dengan *“save as”*. Lalu, file akan menumpuk dalam direktori proyek seperti pada ilustrasi di atas.

Tapi setelah menggunakan Git…

Hanya akan ada satu file dalam proyek dan perubahannya disimpan dalam database.

Git hanya akan menyimpan delta perubahannya saja, dia tidak akan menyimpan seluruh isi file yang akan memakan banyak memori.

Git memungkinkan kita kembali ke versi revisi yang kita inginkan.

1. **Cara Membuat Repositori Git pada Proyek**

Repositori (repository) dalam bahasa indonesia artinya gudang. Repositori merupakan istilah yang digunakan untuk direktori proyek yang menggunakan Git.

Jika kita memiliki sebuah direktori dengan nama proyek-01 dan di dalamnya sudah menggunakan git, maka kita sudah punya repositori bernama proyek-01.

1. Membuat Repositori

Pembuatan repositori dapat dilakukan dengan perintah git init nama-dir

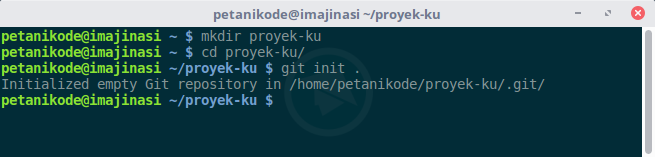


Perintah tersebut akan membuat direktori bernama **proyek-01**. Kalau direktorinya sudah ada, maka Git akan melakukan inisialisasi di dalam direktori tersebut.

Perintah **git init** akan membuat sebuah direktori bernama **.git** di dalam proyek kita. Direktori ini digunakan Git sebagai database untuk menyimpan perubahan yang kita lakukan.

Dan jika kita menghapus direktori ini, maka semua rekaman atau catatan yang dilakukan oleh Git akan hilang.

Perintah git init jika ditambahkan titik artinya kita membuat repository pada direktori tempat berada saat ini.



Perintah berikut ini akan membuat repositori pada direktori /var/ww/html/proyekweb/



1. .gitignore

.gitignore merupakan sebuah file yang berisi daftar nama-nama file dan direktori yang akan diabaikan oleh Git.

Perubahan apapun yang kita lakukan terhadap file dan direktori yang sudah masuk ke dalam daftar .gitignore tidak akan dicatat oleh Git.

Cara menggunakan .gitignore, buat saja sebuah file bernama .gitignore dalam root direktori proyek/repositori.



Pada contoh file **.gitignore** di atas, saya memasukkan direktori **vendor**, **upload**, **cache** dan file **test.php**. File dan direktori tersebut akan diabaikan oleh Git.

Pembuatan file **.gitignore** sebaiknya dilakukan di awal pembuatan repositori.

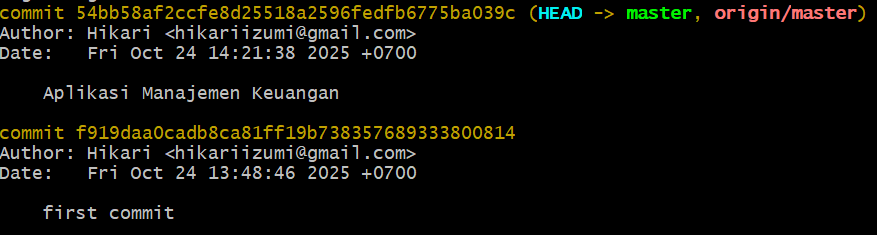
1. **Melihat Catatan Log Revisi**

Pada tutorial sebelumnya, kita sudah membuat dua revisi pada repositori project-01. Sekarang bagaimana caranya kita melihat catatan log dari revisi-reivisi tersebut?

Git sudah menyediakan perintah git log untuk melihat catatan log perubahan pada repositori. Contoh penggunaannya:



Akan terdapat melihat log perubahan apa saja yang sudah dilakukan dalam repositori.



1. Log yang Lebih Pendek

Untuk menampilkan log yang lebih pendek, kita bisa menambahkan argumen –online



Yang akan menghasilkan output :

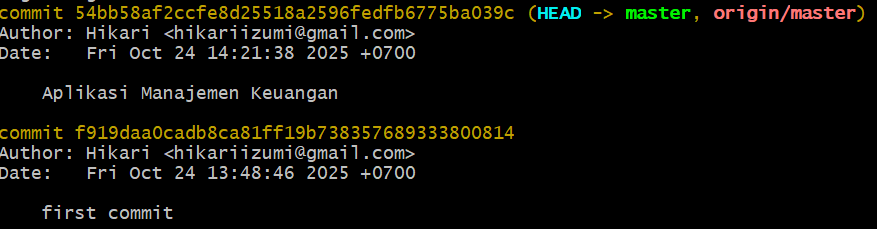


1. Log pada Nomer Revisi/Commit

Untuk melihat log pada revisi tertentu, kita bisa memasukkan nomer revisi/commit



Maka akan menghasilkan output :

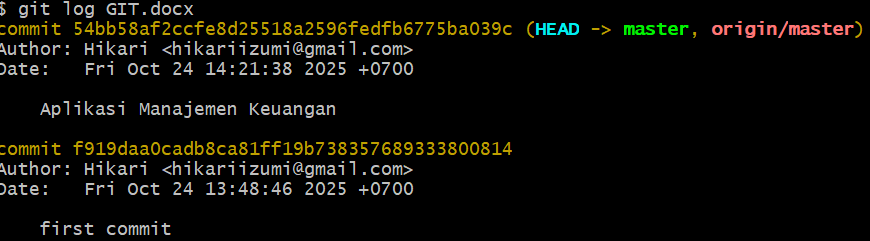


1. Log pada File Tertentu

Untuk melihat revisi pada file tertentu, kita dapat memasukkan nama filenya.



Maka menghasilkan output :

****

1. **Meihat Perbandingan Revisi dengan Git Diff**

Git diff berfungsi untuk melihat perbedaan perubahan di revisi.

1. Melihat Perbandingan Perubahan yang Dilakukan pada Revisi



54bb58af2ccfe8d25518a2596fedfb6775ba039c adalah nomor revisi yang ingin dilihat.



Simbol plus (+) artinya kode yang ditambahkan. Sedangkan jika terdapat simbil minum (-) artinya kode yang dihapus.

1. Melihat Perbandingan pada File

Apa bila kita melakukan banyak perubahan, maka akan banyak sekali tampil output. Karena itu, kita mungkin hanya perlu melihat perubahan untuk file tertentu saja. Untuk melihat perbandingan perubahan pada file tertentu, gunakan perintah berikut.



Perintah di atas akan melihat perbedaan perubahan pada file **index.html** saja.

1. Melihat Perbandingan anatar Revisi/Commit

Perintah untuk membandingkan perubahan pada revisi dengan revisi yang lain adalah sebagai berikut.

