**Maulana Ardiansyah**

**1610320001**

**Perintah dasar GIT**

1. **git config**  
   Salah satu perintah git yang paling banyak digunakan adalah **git config**, yang bisa digunakan untuk mengatur konfigurasi tertentu sesuai keinginan pengguna, seperti email, algoritma untuk diff, username, format file, dll. Contohnya, perintah berikut bisa digunakan untuk mengatur email:
   1. git config --global user.email sam@google.com
2. **git init**  
   Perintah ini digunakan untuk membuat repositori baru. Caranya:
   1. git init
3. **git add**  
   Perintah git add bisa digunakan untuk menambahkan file ke index. Contohnya, perintah berikut ii akan menambahkan file bernama temp.txt yang ada di direktori lokal ke index:
   1. git add temp.txt
4. **git clone**  
   Perintah git clone digunakan untuk checkout repositori. Jia repositori berada di remove server, gunakan:
   1. git clone alex@93.188.160.58:/path/to/repository
   2. Jika salinan repositori lokal ingin dibuat, gunakan:
   3. git clone /path/to/repository
5. **git commit**  
   Perintah git commit digunakan untuk melakukan commit pada perubahan ke head. Ingat bahwa perubahan apapun yang di-commit tidak akan langsung ke remote repository. Gunakan:
   1. git commit –m “Isi dengan keterangan untuk commit”
6. **git status**  
   Perintah git status menampilkan daftar file yang berubah bersama dengan file yang ingin di tambahkan atau di-commit. Gunakan:
   1. git status
7. **git push**  
   git push adalah perintah git dasar lainnya. Push akan mengirimkan perubahan ke master branch dari remote repository yang berhubungan dengan direktori kerja Anda. Misalnya:
   1. git push origin master
8. **git checkout**  
   Perintah git checkout bisa digunakan untuk membuat branch atau untuk berpindah diantaranya. Misalnya, perintah berikut ini akan membuat branch baru dan berpindah ke dalamnya:
   1. command git checkout -b <nama-branch>
   2. Untuk berpindah dari branch satu ke lainnya, gunakan:
   3. git checkout <branch-name>
9. **git remote**  
   Perintah git remote akan membuat user terhubung ke remote repository. Perintah berikut ini akan menampilkan repository yang sedang dikonfigurasi:
   1. git rmote -v
   2. Perintah ini membuat user bisa menghubungkan repository lokal ke remote server:
   3. git remote add origin <93.188.160.58>
10. **git branch**  
    Perintah git branch bisa digunakan untuk me-list, membuat atau menghapus branch. Untuk menampilkan semua branch yang ada di repository, gunakan:
    1. git branch
    2. Untuk menghapus branch:
    3. git branch -d <branch-name>
11. **git pull**  
    Untuk menggabungkan semua perubahan yang ada di remote repository ke direktori lokal, gunakan perintah pull:
    1. git pull
12. **git merge**  
    Perintah merge digunakan untuk menggabungkan sebuah branch ke branch aktif. Gunakan:
    1. git merge <nama-branch>
13. **git diff**  
    Perintah git diff digunakan untuk menampilkan conflicts. Untuk melihat conflicts dengan file dasar, gunakan:
    1. git diff --base <nama-file>
    2. Perintah berikut digunakan untuk menampilkan conflicts diantara branch yang akan di-merge:
    3. git diff <source-branch> <target-branch>
    4. Untuk menampilkan semau conflict yang ada, gunakan:
    5. git diff
14. **git tag**  
    Tagging digunakan untuk menandai commits tertentu. Contohnya:
    1. git tag 1.1.0 <insert-commitID-here>
15. **git log**  
    Dengan menjalankan peritah ini akan menampilkan daftar commits yang ada di branch beserta detail-nya. Contoh outputnya adalah:

commit 15f4b6c44b3c8344caasdac9e4be13246e21sadw

Author: Alex Hunter <alexh@gmail.com>

* 1. Date: Mon Oct 1 12:56:29 2016 -0600

1. **git reset**  
   Untuk me-reset index dan bekerja dengan kondisi commit paling baru, gunakan perintah git reset:
   1. git reset --hard HEAD
2. **git rm**  
   Gunakan perintah ini untuk menghapus file dari index dan direktori kerja. Contohnya:
   1. git rm filename.txt
3. **git stash**  
   Mungkin inilah salah satu perintah dasar git yang jarang digunakan orang, yang bisa membantu menyimpan perubahan yang tidak langsung di-commit, namun hanya sementara. Contoh:
   1. git stash
4. **git show**  
   Untuk menampilkan informasi tentang object pada git, gunakan git show:
   1. git show
5. **git fetch**  
   Perintah ini digunakan untuk menampilkan semua object dari remote repository yang tidak berada di direktori kerja lokal. Contohnya:
   1. git fetch origin
6. **git ls-tree**  
   Untuk menampilkan susunan object berdasarkan nama dan mode setiap item, dan nilai blob SHA-1, gunakan perintah git ls-tree. Contohnya:
   1. git ls-tree HEAD
7. **git cat-file**  
   Menggunakan nilai SHA-1, menampilkan tipe object dengan menggunakan perintah git cat-file. Contohnya:
   1. git cat-file –p d670460b4b4aece5915caf5c68d12f560a9fe3e4
8. **git prep**  
   git prep mengizinkan pengguna mencari frase dan/atau kata yang berada di dalam direktori. Contohnya, untuk mencari www.hostinger.co.id di dalam semua file, gunakan:
   1. git grep "www.hostinger.co.id"
9. **gitk**  
   gitk adalah tampilan grafis dari repository lokal yang bisa dipanggil dengan menjalankan perintah:
   1. gitk
10. **git instaweb**  
    Dengan perintah git instaweb, web server bisa dijalan berdampingan dengan repository lokal. Nantinya web browser akan langsung diarahkan ke server tersebut. Contohnya:
    1. git instaweb –httpd=webrick
11. **git gc**  
    Untuk mengoptimasi repository melalui garbage collection, yang akan membersihkan file yang tidak dibutuhkan dan mengoptimasinya, gunakan:
    1. git gc
12. **git archive**  
    Perintah git archive memungkinkan user membuat file zip atau tar yang mengandung susunan repository. Contohnya:
    1. git archive --format=tar master
13. **git prune**  
    Melalui perintah git prune, object yang tidak memiliki incoming pointers akan dihapus. Gunakan:
    1. git prune
14. **git fsck**  
    Untuk membuat pengecekan keseluruhan dari file system git, gunakan perintah git fsck. Object yang corrupt akan dikenali:
    1. git fsck
15. **git rebase**  
    Perintah ini digunakan untuk menerapkan ulang commit di branch yang lain. Contohnya:
    1. git rebase master