

**LAPORAN PRAKTIKUM
REKAYASA PERANGKAT LUNAK**

GIT LANJUT BAGIAN 2



**DISUSUN OLEH
LINDARWATI KUSUMAWARDANI M0517026**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET**

BAB I

PENDAHULUAN

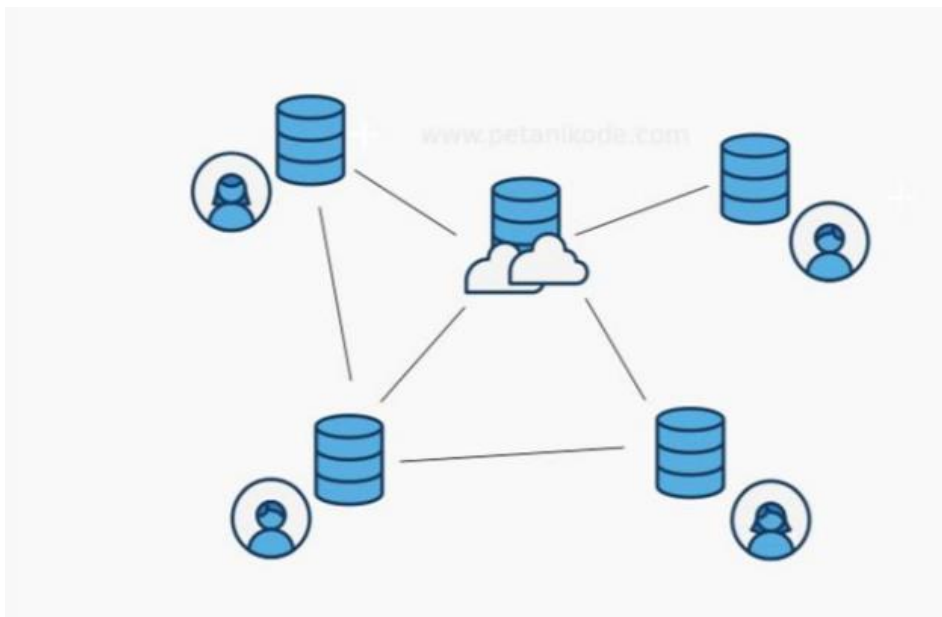
1.1 Tujuan Praktikum

1. Praktikan mampu membuat repository baru dalam proyek.
2. Praktikan mampu menyimpan perubahan revisi dengan git commit.
3. Praktikan mampu melihat catatan log revisi.
4. Praktikan mampu melihat perbandingan revisi file.

1.2 Dasar Teori

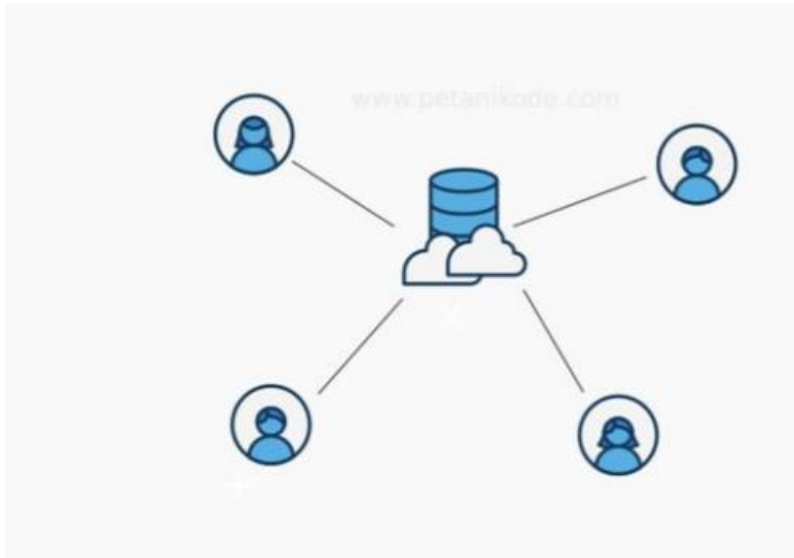
1. GIT

Git adalah salah satu *tool* yang sering digunakan dalam proyek pengembangan *software*. Git bahkan menjadi *tool* yang wajib dipahami oleh *programmer*, karena banyak digunakan di mana-mana. 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 orang yang terlibat dalam pengkodean proyek akan menyimpan *database* Git, sehingga akan memudahkan dalam mengelola proyek baik *online* maupun *offline*. Dalam Git terdapat *merge* untuk menyebut 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.

Jadi selain untuk mengontrol versi, git juga digunakan untuk kolaborasi. Saat ini Git menjadi salah satu tool terpopuler yang digunakan pada pengembangan software open source maupun closed source. Google, Microsoft, Facebook dan berbagai perusahaan raksasa lainnya menggunakan Git. Selain itu, berikut ini ada beberapa manfaat yang akan anda rasakan setelah bisa menggunakan Git:

- a. Bisa menyimpan seluruh versi source code;
- b. Bisa paham cara kolaborasi dalam proyek;
- c. Bisa ikut berkontribusi ke proyek open-source;
- d. Lebih aman digunakan untuk kolaborasi, karena kita bisa tahu apa yang diubah dan siapa yang mengubahnya.
- e. Bisa memahami cara deploy aplikasi modern;
- f. Bisa membuat blog dengan SSG.

BAB II

PEMBAHASAN

2.1 Percobaan

1. MEMBUAT REPOSITORI

Pembuatan repository dapat dilakukan dengan git init nama-dir dalam contoh ini namanya adalah proyek-01 dilihat pada baris berikutnya disana terdapat alokasi dimana repo disimpan, di dalam repo itu nanti semua catatan perubahan akan tersimpan.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads (master)
$ git init proyek-01
Initialized empty Git repository in C:/Users/acer/Downloads/proyek-01/.git/

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads (master)
$
```

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git init
Reinitialized existing Git repository in C:/Users/acer/Downloads/proyek-01/.git/

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git init .
Reinitialized existing Git repository in C:/Users/acer/Downloads/proyek-01/.git/

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git init home
Initialized empty Git repository in C:/Users/acer/Downloads/proyek-01/home/.git/

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git ini contact
git: 'ini' is not a git command. See 'git --help'.

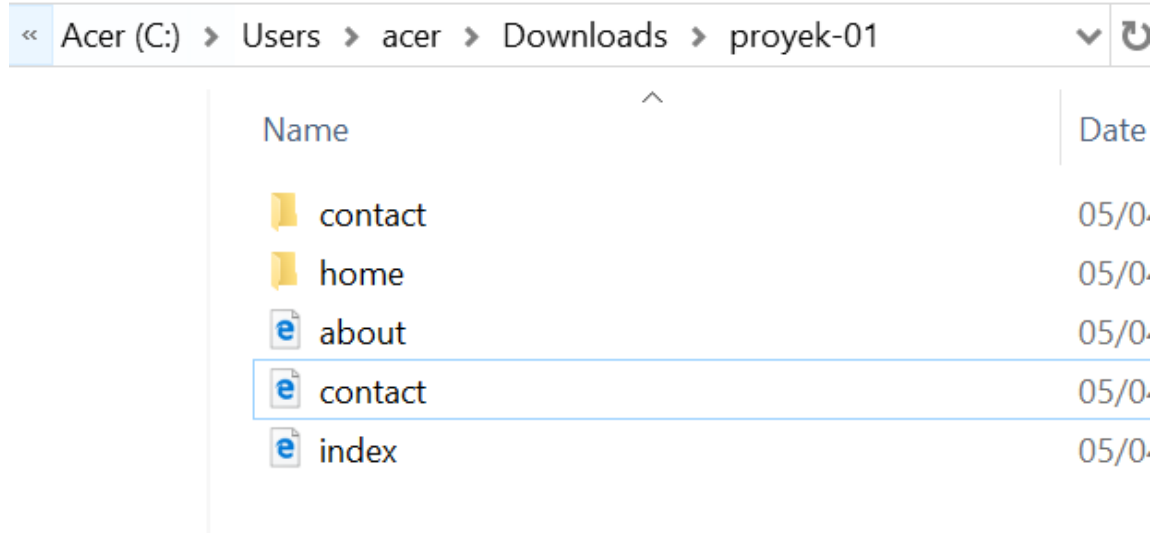
The most similar commands are
  init
  gui
  init-db

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git init contact
Initialized empty Git repository in C:/Users/acer/Downloads/proyek-01/contact/.git/

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git init index.html
Initialized empty Git repository in C:/Users/acer/Downloads/proyek-01/index.html/.git/
```

2. SIMPAN PERUBAHAN REVISI DENGAN GIT COMMIT

Pada tutorial sebelumnya kita sudah belajar cara membuat repo kosong, sekarang kita akan mencoba menambahkan tiga *file* HTML kosong,



Kemudian setelah ditambahkan coba cek perintah dengan `git status` untuk melihat status repo nya, tampilan menjadi seperti dibawah: dapat dilihat posisi kita sekarang yaitu berada di *branch master* dan terlihat bahwa belum ada *commits*. Dalam GIT terdapat tiga kelompok kondisi *file* yaitu:

1. Modified

Modified adalah kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di version control. Contohnya pada gambar di atas, ada tiga file HTML yang dalam kondisi modified.

2. Staged

Staged adalah kondisi dimana revisi sudah ditandai, tetapi belum disimpan di version control. Untuk mengubah kondisi file dari modified ke staged gunakan perintah `git add nama_file`.

3. Committed

Committed adalah kondisi dimana revisi sudah disimpan di version control. perintah untuk mengubah kondisi file dari staged ke committed adalah git commit.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   about.html
        new file:   contact.html
        new file:   index.html
```

3. MEMBUAT REVISI PERTAMA

Pada langkah awal perlu diubah ketiga files tadi menjadi *staged* dengan perintah-perintah dibawah ini : bisa dilakukan dalam satu kali dengan *git add index.html about.html contact.html* atau dengan *git add *.html* atau dengan *git add .* Jika proses sudah pernah dilakukan akan muncul tulisan fatal dibagian bawahnya.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add index.html

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add about.html
fatal: pathspec 'about.html' did not match any files

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add contact.html
fatal: pathspec 'contact.html' did not match any files

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add index.html

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add index.html about.html contact.html
fatal: pathspec 'about.html' did not match any files

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add index.html about.html contact.html
```

Setelah dirasa sudah melakukan perintah *staged* coba ketikkan *git status* kondisi file akan menjadi staged. Kemudian ubah kondisi file tersebut ke committed agar perubahan disimpan oleh git. Dengan *git commit -m "isi"* setelah itu ketik *git status* untuk menampilkan, jika muncul tulisan seperti digambar paling bawah berarti revisi pertama berhasil dibuat.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)


        new file:   about.html
        new file:   contact.html
        new file:   index.html

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git commit -m "iya ini diisi"
[master (root-commit) 1fb7581] iya ini diisi
 3 files changed, 3 insertions(+)
 create mode 100644 about.html
 create mode 100644 contact.html
 create mode 100644 index.html

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git status
On branch master
nothing to commit, working tree clean
```


4. MEMBUAT REVISI KEDUA

Lakukan perubahan pada isi text html dengan judul index seperti yang terdapat di modul sehingga menjadi seperti berikut, kemudian cek dengan git status untuk melihat apakah modifikasi file berhasil dan dapat dilihat dari tulisan **modified : index.html**. Kemudian lakukan git add index.html untuk melakukan commit pada file index

 index - Notepad

```
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Belajar Git - Project 01</title>
</head>
<body>
<p>Hello Dunia!, Saya sedang belajar Git</p>
</body>
</html>
```

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git index.html
git: 'index.html' is not a git command. See 'git --help'.

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git add index.html

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git commit -m "ditambahkan isi"
[master b4ef327] ditambahkan isi
1 file changed, 10 insertions(+), 1 deletion(-)
```

5. MELIHAT CATATAN LOG REVISI

Dengan fitur **git log** maka kita dapat melihat perubahan yang baru saja dilakukan dan sudah dilakukan oleh siapa pada hari serta pukul berapa. Contohnya seperti dibawah ini.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log
commit b4ef32788f69ae62e84629ae725c215faf546d55 (HEAD -> master)
Author: lklindar <lindarkusuma@student.uns.ac.id>
Date:   Fri Apr 5 14:02:02 2019 +0700

    ditambahkan isi

commit 1fb7581dd18e083f2ba7a7fe4ff7a3bf508fb6d6
Author: lklindar <lindarkusuma@student.uns.ac.id>
Date:   Fri Apr 5 13:58:53 2019 +0700

    iya ini diisi

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log --oneline
b4ef327 (HEAD -> master) ditambahkan isi
1fb7581 iya ini diisi
```

6. LOG YANG LEBIH PENDEK

Coba masukan **git log --oneline** maka akan menampilkan kode serta perubahan apa yang dilakukan seperti yang dapat dilihat dari atas.

7. LOG PADA NOMER REVISI/COMMIT

Jika kita sudah melihat nomer atau ode commit dari **git log --online** dan ingin melihat lebih detail mengenai perubahan yang dilakukakn dari si kode tadi dapat memasukan git log #nomer kemudian akan menampilkan isi yang mirip dengan git log hanya saja ini hanya menampilkan isi dari pemilk kode tidak semuanya.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log 1fb7581
commit 1fb7581dd18e083f2ba7a7fe4ff7a3bf508fb6d6
Author: lklindar <lindarkusuma@student.uns.ac.id>
Date:   Fri Apr 5 13:58:53 2019 +0700

    iya ini diisi
```

8. LOG PADA FILE TERTENTU

Cara lain untuk melihat revisi adalah dengan **git log #namafile** cara ini relative lebih mudah karena kita tidak perlu mencari tahu kodenya terlebih dahulu.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log index.html
commit b4ef32788f69ae62e84629ae725c215faf546d55 (HEAD -> master)
Author: lklindar <lindarkusuma@student.uns.ac.id>
Date:   Fri Apr 5 14:02:02 2019 +0700

    ditambahkan isi

commit 1fb7581dd18e083f2ba7a7fe4ff7a3bf508fb6d6
Author: lklindar <lindarkusuma@student.uns.ac.id>
Date:   Fri Apr 5 13:58:53 2019 +0700

    iya ini diisi

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log --author="lindarrrr"
```

9. LOG REVISI YANG DILAKUKAN OLEH AUTHOR TERTENTU

Dengan menggunakan **git diff #nomer** maka kita dapat melihat isi perubahan yang dilakukan, disana akan muncul tampilan text sebelum diubah dan isinya yang baru. Hal inilah yang menjadikan kegunaan git sangat bermanfaat karena ia hanya menyimpan satu file yang baru namun jika kita memerlukan untuk melihat file lama kita masih dapat melakukannya sehingga lebih menghemat penyimpanan di PC atau dapat juga menggunakan **git diff #nama file** tanda tambah adalah text baru yang ditambahkan yang warna merah data lamanya.

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git diff 1fb7581
diff --git a/index.html b/index.html
index 7af5537..6e1e061 100644
--- a/index.html
+++ b/index.html
@@ -1,10 @@
- nnnn
\ No newline at end of file
+ <!DOCTYPE html>
+<html>
+<head>
+<meta charset="utf-8">
+<title>Belajar Git - Project 01</title>
+</head>
+<body>
+<p>Hello Semua, Saya sedang belajar Git</p>
+</body>
+</html>
\ No newline at end of file
```

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git diff 1fb7581
diff --git a/index.html b/index.html
index 7af5537..2cd720e 100644
--- a/index.html
+++ b/index.html
@@ -1,10 @@
- nnnn
\ No newline at end of file
+ <!DOCTYPE html>
+<html>
+<head>
+<meta charset="utf-8">
+<title>Belajar Git - Project 01</title>
+</head>
+<body>
+<p>Hello Dunia!, Saya sedang belajar Git</p>
+</body>
+</html>
\ No newline at end of file

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log --oneline
b4ef327 (HEAD -> master) ditambahkan isi
1fb7581 iya ini diisi
```

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git diff index.html
diff --git a/index.html b/index.html
index 6e1e061..2cd720e 100644
--- a/index.html
+++ b/index.html
@@ -5,6 +5,6 @@
<title>Belajar Git - Project 01</title>
</head>
<body>
- <p>Hello Semua, Saya sedang belajar Git</p>
+ <p>Hello Dunia!, Saya sedang belajar Git</p>
</body>
- </html>
\ No newline at end of file
+ </html>
\ No newline at end of file

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git diff 6e1e0612cd720e
fatal: ambiguous argument '6e1e0612cd720e': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git log --oneline
b4ef327 (HEAD -> master) ditambahkan isi
1fb7581 iya ini diisi
```

```
acer@DESKTOP-GB08TFL MINGW64 ~/Downloads/proyek-01 (master)
$ git diff b4ef327 1fb7581
diff --git a/index.html b/index.html
index 6e1e061..7af5537 100644
--- a/index.html
+++ b/index.html
@@ -1,10 +1 @@
- <!DOCTYPE html>
- <html>
- <head>
- <meta charset="utf-8">
- <title>Belajar Git - Project 01</title>
- </head>
- <body>
- <p>Hello Semua, Saya sedang belajar Git</p>
- </body>
- </html>
\ No newline at end of file
+ nnnn
\ No newline at end of file
```