



Free Kelas WAPPSANDROID

Materi Pertemuan #02

Kamis, 22 Maret 2018

Pukul 20.00 – 23.00 WIB

- **Membuat akun Github dan Setting**
- **Membuat Repository baru**
- **Fitur social : Follow, Star, Watch, Fork**
- **Melakukan pencarian**
- **Fork/Copy, Download /Pull, Upload/Push**
- **Mencoba Aplikasi Desktop Gitkraken**

oleh :

Agus Muhajir

<http://github.com/hajirodeon>

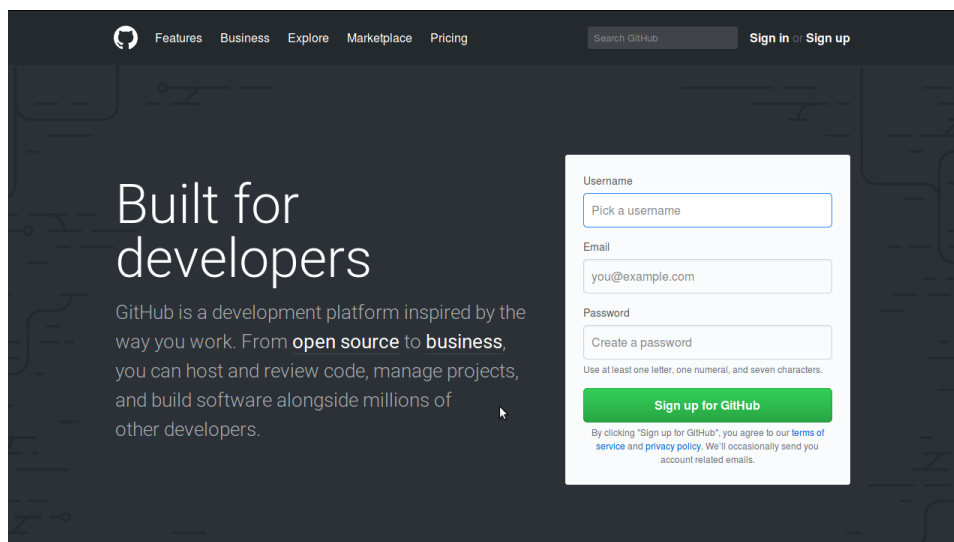
Apa itu Github?

Github adalah layanan web populer bersama untuk proyek pengembangan perangkat lunak yang menggunakan sistem pengontrol versi Git atau bisa dikatakan layanan hosting untuk proyek open source yang menggunakan Tool System revisi kontrol Git. Pengertian mudahnya, Github adalah jejaring sosial untuk para developer dalam mengembangkan suatu proyek.

Situs github ini menyediakan fungsionalitas jejaring sosial dan grafik jejaring sosial untuk menampilkan bagaimana para pengembang menggarap versi repositori mereka.

Github juga mengoperasikan layanan-layanan lain: situs bergaya pastebin bernama Gist yang menyajikan wiki untuk masing-masing repositori dan laman web yang bisa disunting melalui sebuah repositori Git, layanan penginangan slide bernama Speaker Deck, serta platform analisis web bernama Gauges.

Per Januari 2010, Github dioperasikan dengan nama Github, Inc. Perangkat lunak yang menjalankan Github ditulis dengan Ruby on Rails dan Erlang oleh para pengembang Github, Inc. (sebelumnya dikenal sebagai Logical Awesome), yakni Chris Wanstrath, PJ Hyett, dan Tom Preston-Werner.



Gambar 1. Tampilan Halaman Utama Situs Github.com

Apa itu Git?

GIT merupakan sebuah Version Control System (VCS) yang digunakan dalam tim pengembangan perangkat lunak untuk bekerja bersama. Version Control maksudnya sistem Git akan mencatat setiap perubahan yang terjadi pada source code kita sehingga memungkinkan untuk mengambil kembali source code lama jika suatu saat kita ingin kembali ke versi berapapun dari aplikasi yang pernah kita tulis.

Yang membedakan Github dari Project hosting yang lain bisa kita lihat di bawah ini:

1. Github menggunakan Git sebagai tool utama dalam merevisi code.
2. Github menyediakan free hosting untuk proyek open source, namun Github juga menawarkan hosting berbayar untuk proyek perusahaan atau pribadi yang sifatnya private.
3. Github ditargetkan untuk para developer yang bekerja secara tim dan tidak disatu tempat.
4. Github mempunyai fitur Social seperti Twitter seperti follow atau favorit (Star).
5. Github mendukung semua bahasa pemrograman, anda bebas memakai bahasa dan tool yang biasa anda pakai.

Kemudian untuk lebih memahami sedikit tentang github, dimisalkan kita sedang mengembangkan sebuah aplikasi web menggunakan PHP. Ketika kita sudah selesai melakukan koding maka kita menyimpannya ke dalam repository Git atau istilahnya commit. Pada langkah ini kita sudah membuat versi source code kita katakanlah versi 1. Besoknya kita melakukan perubahan pada versi 1 kita tadi dan seperti sebelumnya kita juga melakukan commit ke dalam repository, maka versi 2 akan tercipta. Lantas apakah versi 1 akan hilang? tidak. Setelah kita bekerja beberapa kali memperbaiki atau menambahkan fitur pada source code kita sampai 20 versi pun kita akan selalu bisa untuk kembali ke versi lama yang keberapapun begitu juga sebaliknya. Commit yang tadi kita lakukan hanya akan disimpan pada repository lokal yang ada pada komputer kita. Lantas bagaimana agar bisa diakses bersama-sama oleh anggota tim? maka kita membutuhkan sebuah repository central. Website yang menyediakan jasa repository central untuk Git adalah 'Github'.

Fitur social di github

Beberapa fitur Social di Github:

1. Github user profile.
2. Follow, Kita bisa memfollow profil programmer lain di github.
3. Star, fungsinya sama dengan Bookmark.
4. Watch, memonitor repository tertentu. Setiap ada perubahan di Repo, maka kita akan dikirim notifikasi.
5. Fork, mengcopy keseluruhan code repository ke repo kita sendiri.

Github Profile

Sama seperti Twitter, Github juga mempunyai profil. Profil berisi foto, jumlah repo yang dimiliki, jumlah follower, jumlah user yang kita follow dan Star (jumlah proyek yang kita beri tanda bintang). Biasanya makin banyak followernya, makin canggih skill user tersebut.

Fitur pencarian

Github mempunyai lebih dari 10 juta repo software. Nah banyak banget kan? Jika anda ingin mencari proyek open source tertentu, gunakan fitur pencarian, Hasil pencarian tampak seperti di gambar 2 diatas. Untuk memperkecil hasil pencarian bisa di filter berdasarkan Repo, code, issue atau nama user atau bahasa pemrograman. Hasil pencarian bisa diurutkan berdasarkan jumlah bintang tau jumlah “fork”. Biasanya makin banyak bintangnya, makin berkualitas source codenya.

Struktur halaman proyek di Github

Bagian Statistik

Bagian pertama adalah statistik, kita bisa melihat jumlah Watch, star dan fork dibagian atas halaman. Bootstrap adalah salahsatu proyek paling populer di github, maka jangan heran jika ada lebih dari 63 ribu user yang memberi tanda bintang.

Bagian Code

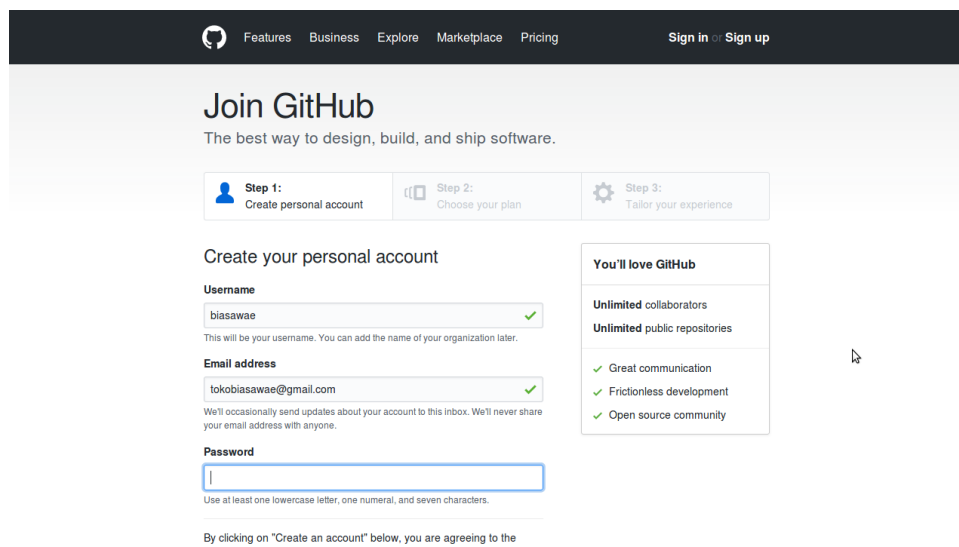
Bagian utama dari struktur proyek ini adalah code itu sendiri. Kode ini juga berisi penjelasan kapan dan apa file terakhir yang diedit berserta penjelasan perubahan apayang terjadi.

Bagian Clone dan Download

Clone in desktop di pakai untuk mengkopi source code proyek tersebut ke komputer kita , namun komputer kita harus sudah terinstall Github for Windows atau Mac. Bagian paling penting dan paling sering dipakai adalah menu **DOWNLOAD**. jika anda ingin langsung mencoba code di komputer , cukup klik download, maka source code otomatis akan dizip dan diunduh ke kompuer anda.

Membuat Akun Github

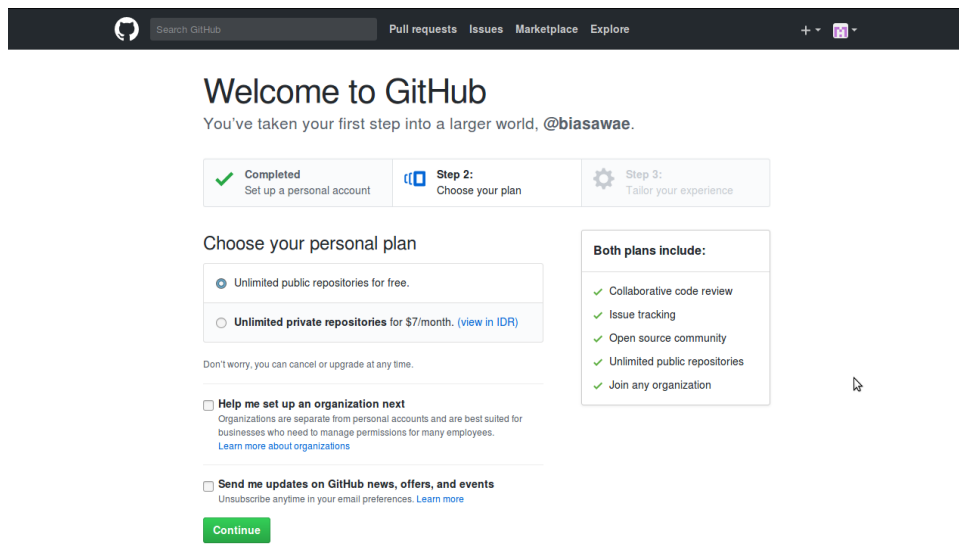
1. silahkan buka github.com, kemudian pilih Sign Up.



Gambar 2. Form SignUp Registrasi.

Tentukan Username Anda, agar bisa memiliki akses dengan alamat
github.com/usernameanda

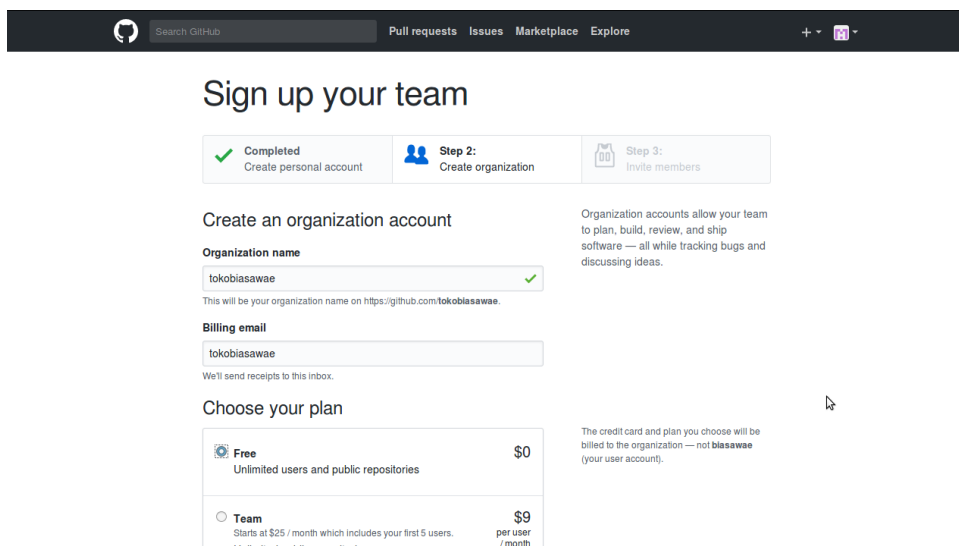
2. Pilih layanan personal yang Free.



The screenshot shows the GitHub 'Welcome to GitHub' page for a user named @biasawae. The page is divided into three steps: Step 1 (Completed: Set up a personal account), Step 2 (Choose your plan), and Step 3 (Tailor your experience). Under 'Choose your personal plan', there are two options: 'Unlimited public repositories for free.' (selected) and 'Unlimited private repositories for \$7/month. (view in IDR)'. A note states: 'Don't worry, you can cancel or upgrade at any time.' Below the options are checkboxes for 'Help me set up an organization next' and 'Send me updates on GitHub news, offers, and events'. A green 'Continue' button is at the bottom.

Gambar 3. Tampilan Layanan Personal Yang Akan Dipilih

3. Pilih layanan organisasi yang Free.

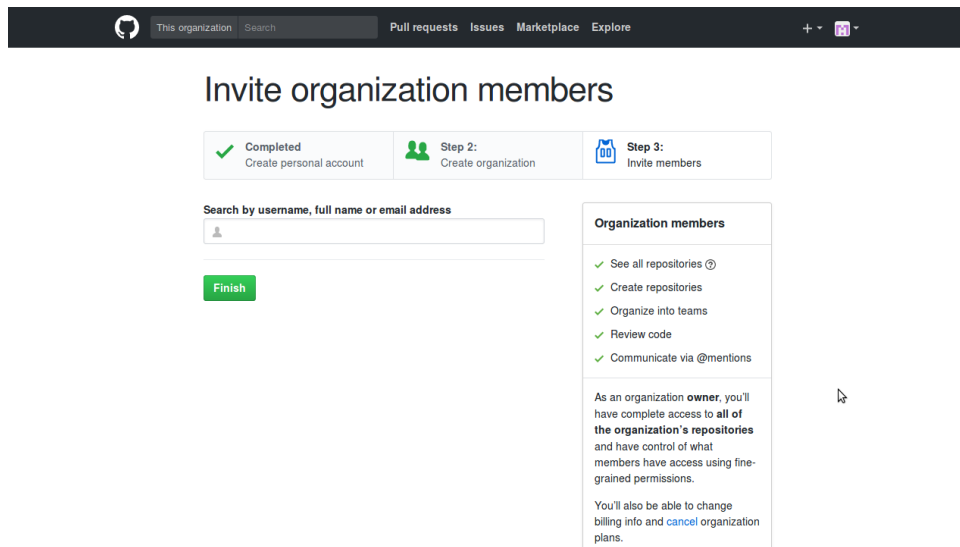


The screenshot shows the GitHub 'Sign up your team' page. The steps are: Step 1 (Completed: Create personal account), Step 2 (Create organization), and Step 3 (Invite members). Under 'Create an organization account', the 'Organization name' field is filled with 'tokobiasawae' and has a green checkmark. Below it, a note says: 'This will be your organization name on https://github.com/tokobiasawae.' The 'Billing email' field is also filled with 'tokobiasawae'. A note says: 'We'll send receipts to this inbox.' Below this is the 'Choose your plan' section with two options: 'Free' (Unlimited users and public repositories, \$0) and 'Team' (Starts at \$25 / month which includes your first 5 users. Unlimited public repositories, \$9 per user / month). A note on the right states: 'The credit card and plan you choose will be billed to the organization — not biasawae (your user account).'

Gambar 4. Tampilan Layanan Organisasi Yang Dipilih

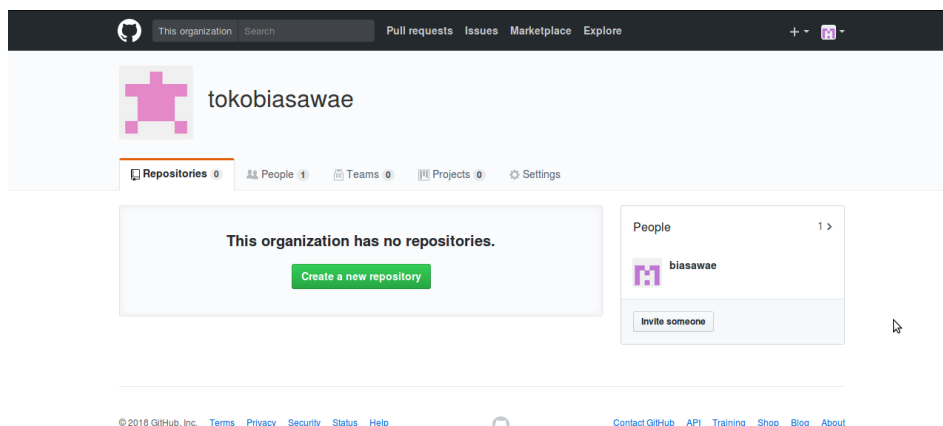
4. Menuju Finish, Invite Member

Selesaikan proses registrasi, dengan Invite Member bila tahu. Lalu klik Finish.



Gambar 5. Tampilan Invite Member untuk Finish

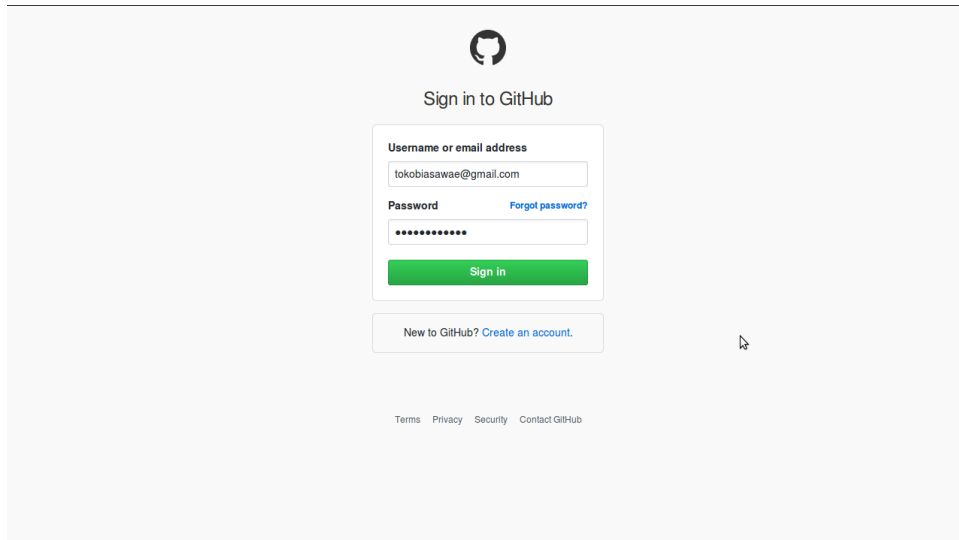
5. Pembuatan Akun Github Selesai.



Gambar 6. Tampilan Halaman Utama Akun Github Anda

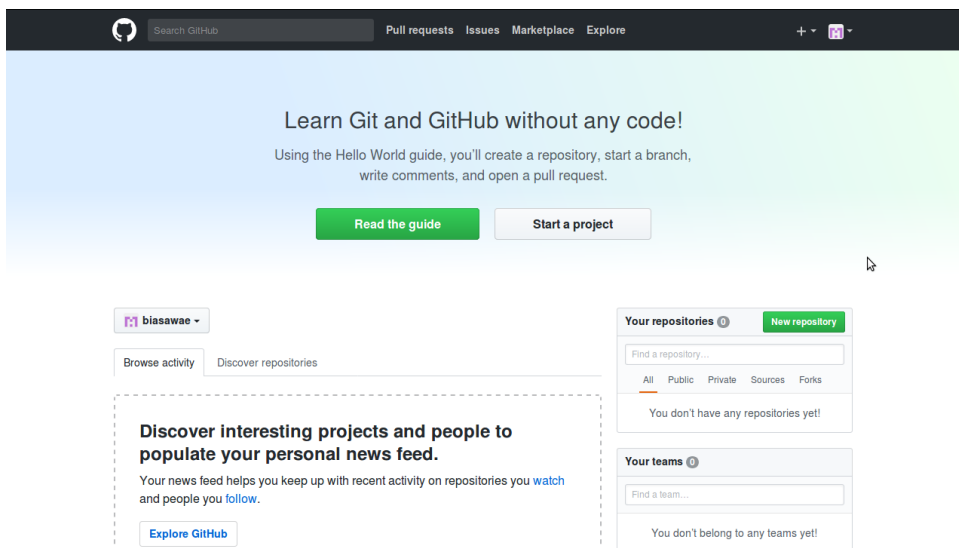
Login Akun Github

1. Silahkan klik Sign In



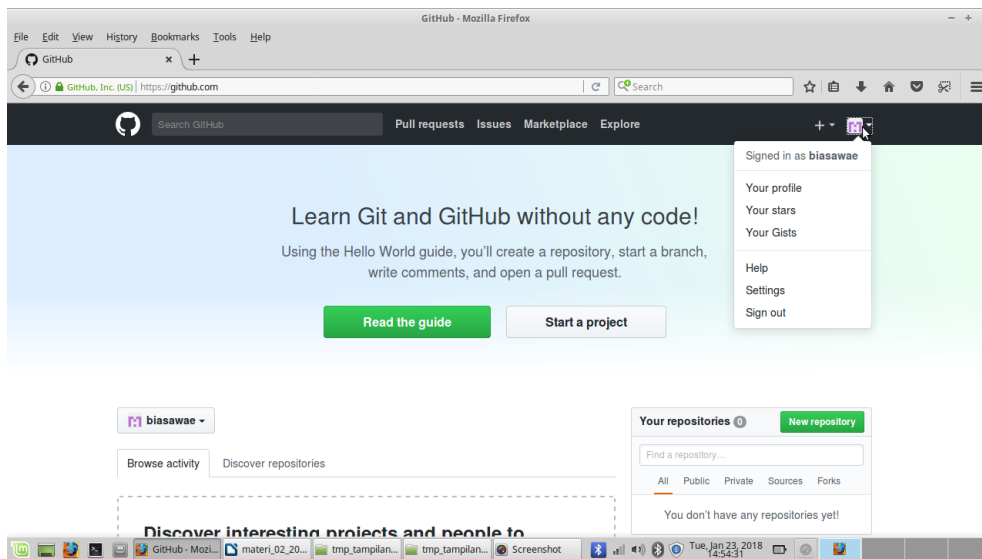
Gambar 7. Tampilan Halaman Login Akun Github.
Silahkan masukkan username dan password Anda ya..

2. Tampilan Halaman Utama Akses Akun Github Kita



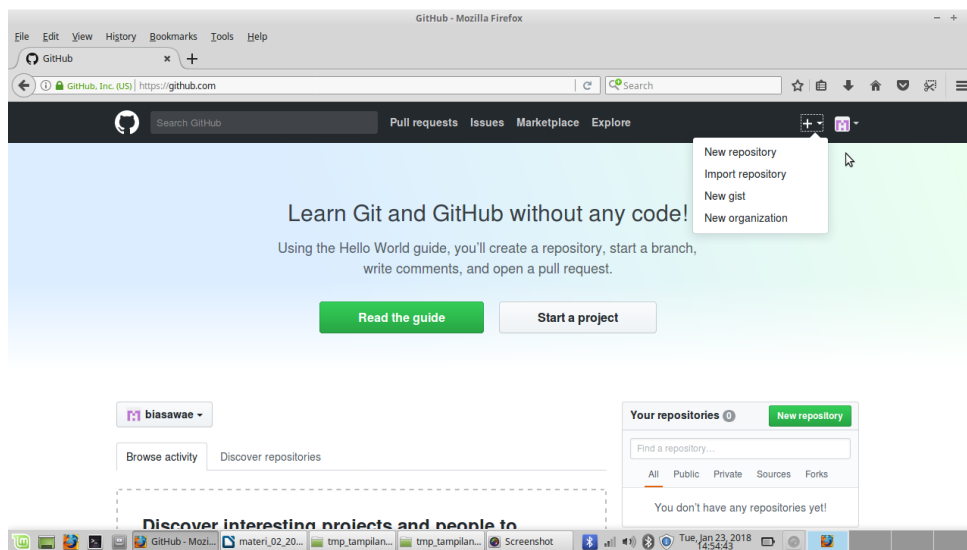
Gambar 8. Tampilan Halaman Utama Akses Github Kita

3. Menu Profil



Gambar 9. Tampilan Menu Profil

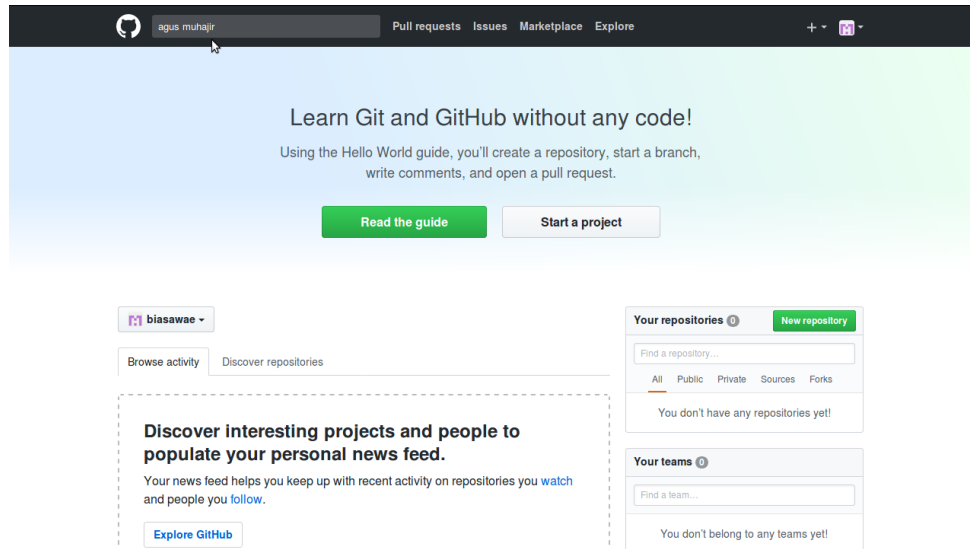
4. Menu Repository



Gambar 10. Tampilan Menu Repository

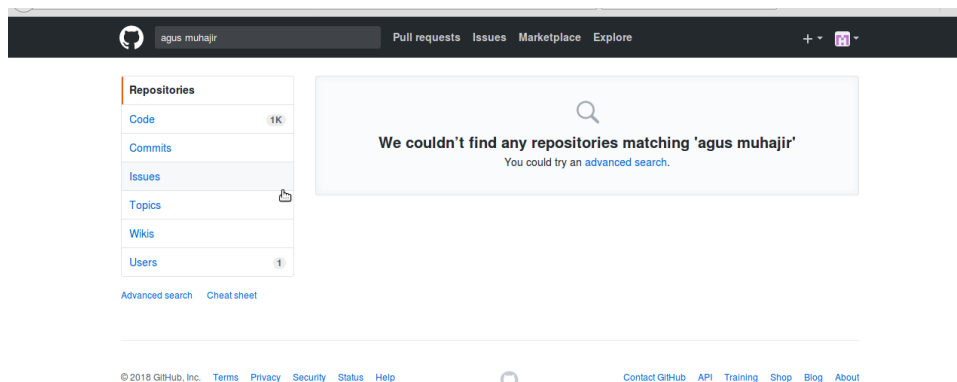
Mencari User dan Proyek

1. Misalkan mencari dengan memasukkan kata kunci “agus muhajir”



Gambar 11. Tampilan Kata Kunci Pencarian
Masukkan kata kunci, kemudian tekan enter.

2. Hasil Pencarian

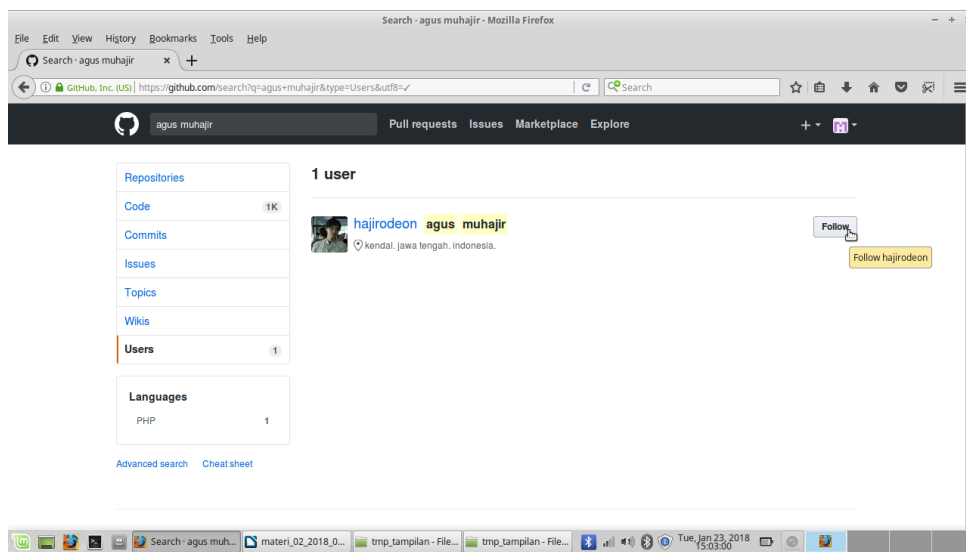


<https://github.com/search?q=agus+muhajir&type=issues&utf8=%E2%9C%93>

Gambar 12. Tampilan Hasil Pencarian.

Terlihat disini, ditemukan pada “code” dan “users” untuk kata kunci tersebut.
Silahkan klik.

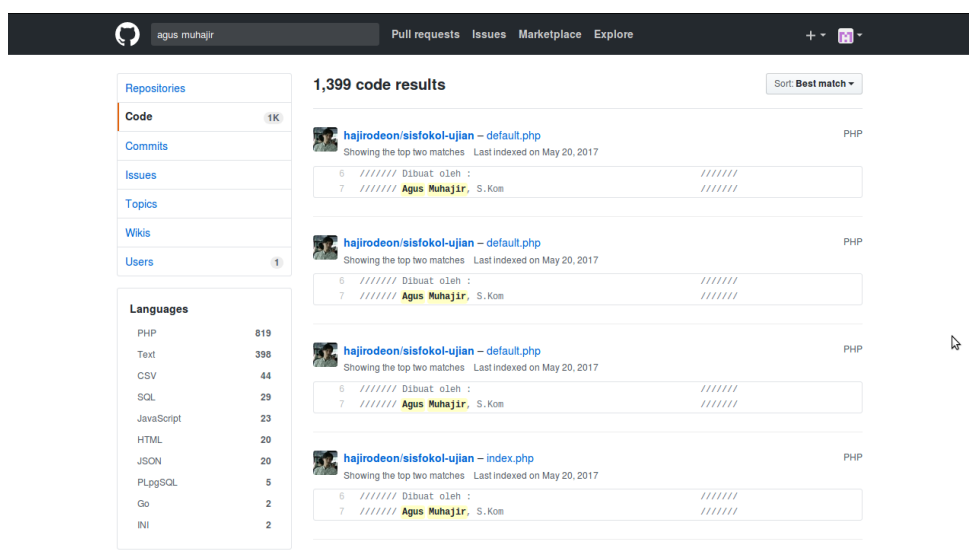
3. Follow Users



Gambar 13. Tampilan Follow User

Dengan klik follow, maka kita akan mengikuti terus informasi terbaru dari user tersebut.

4. Hasil Pencarian Code



Gambar 14. Tampilan Hasil Pencarian Code

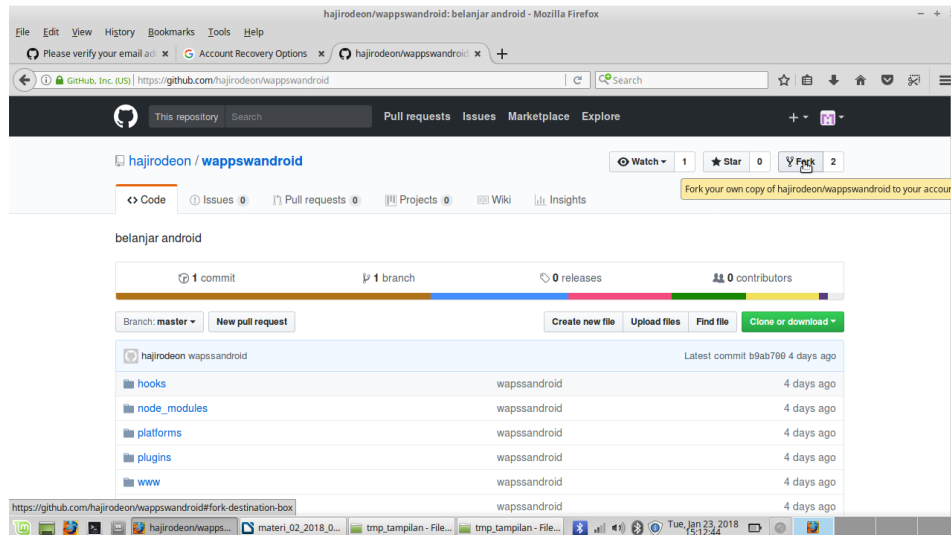
Terlihat diatas, beragam baris code program, yang isinya sesuai dengan kata kunci pencarian.

Salin Repository

1. Buka alamat suatu user github dan proyek repository nya. Misalkan :

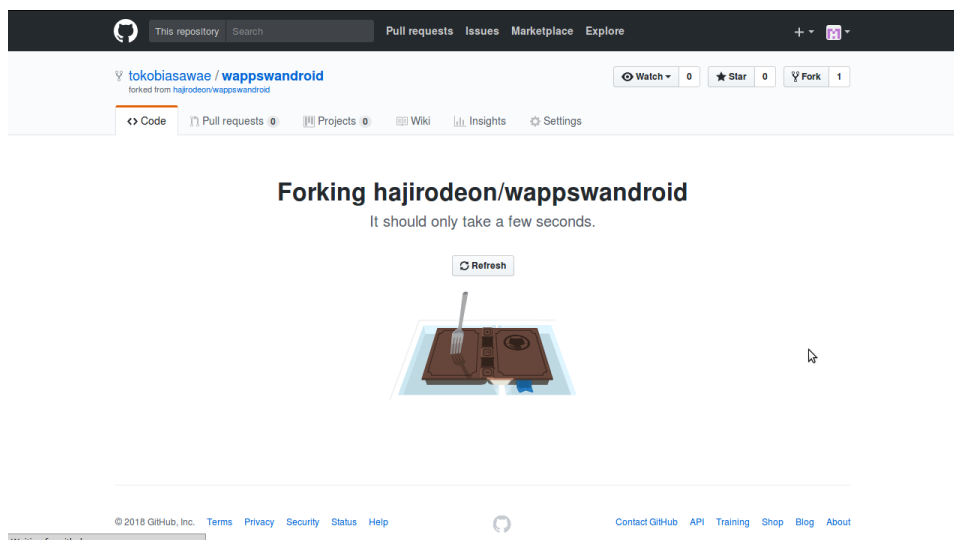
<https://github.com/hajirodeon/wappswandroid>

2. Tekan FORK, untuk salin



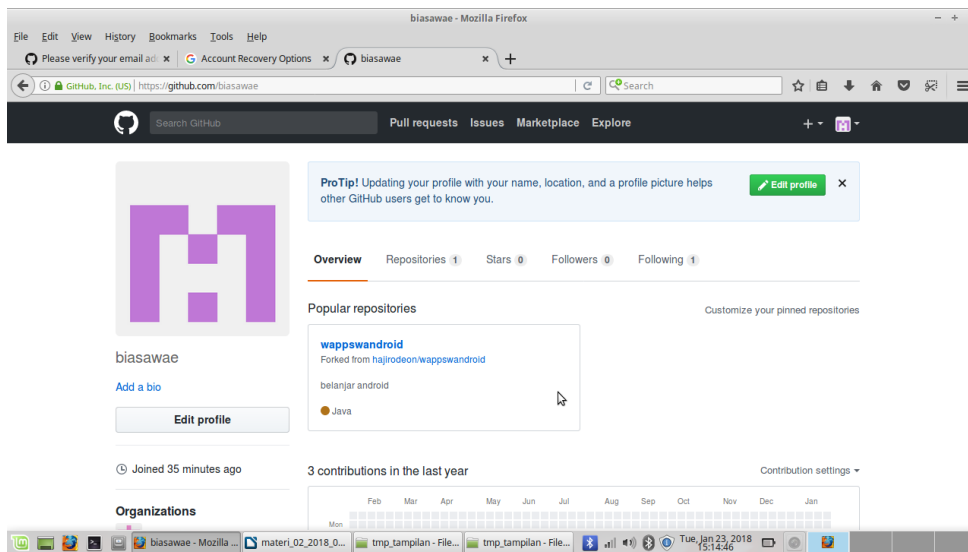
Gambar 15. Tampilan Halaman Suatu Proyek Repository yang akan disalin.

3. Proses Salin sedang berlangsung...



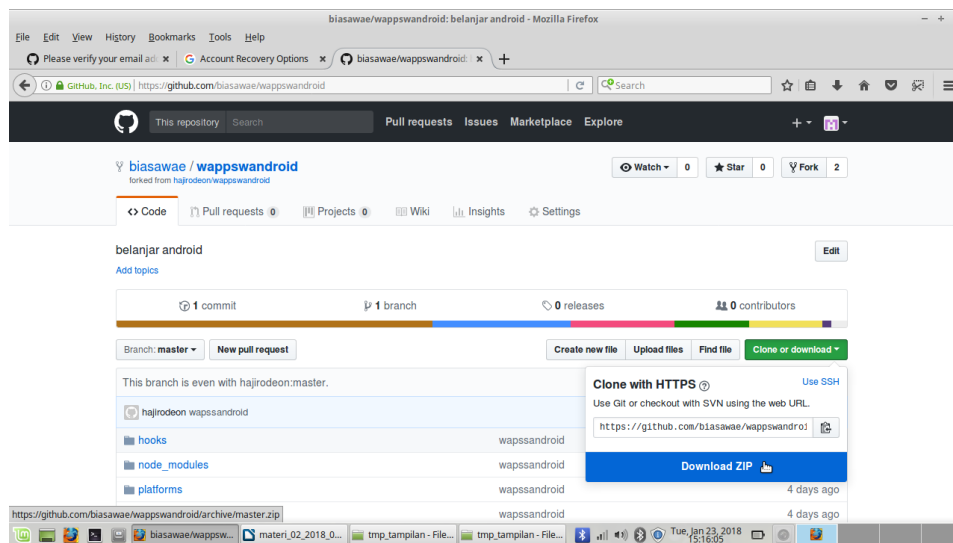
Gambar 16. Tampilan Proses salin

4. Repository yang berhasil kita ambil



Gambar 17. Tampilan Hasil Repository yang telah kita salin

5. Unduh Repository



Gambar 18. Cara Unduh Repository suatu Proyek

buat repositori baru

buat lah direktori baru, buka dan jalankan

```
# git init
```

untuk membuat repositori git baru.

periksa repositori

buat lah salinan kerja dari repositori lokal dengan menjalankan perintah

```
# git clone /jalur/ke/repositori
```

saat menggunakan server jarak-jauh, perintahnya menjadi

```
# git clone namapengguna@host:/jalur/ke/repositori
```

alur-kerja

repositori lokal kamu terdiri dari tiga bagian pokok yang disebut "trees" dikelola oleh git. yang pertama adalah Direktori Kerja yang menyimpan berkas aktual. kedua adalah Indeks yang berperan sebagai pengolah data dan terakhir HEAD yang mengarah pada komit terakhir.

tambah & komit

kamu bisa melakukan perubahan (penambahan ke **Indeks**) menggunakan

```
# git add <namaberkas>
```

```
# git add *
```

Ini merupakan langkah awal alur-kerja dasar git. Untuk komit sepenuhnya gunakan

```
# git commit -m "Pesan komit"
```

Sekarang berkas telah berkomit di **HEAD**, tapi belum di repositori jarak-jauh.

mengirim perubahan

Saat ini perubahan telah tersimpan di **HEAD** salinan kerja lokal kamu. Untuk mengirimkannya ke repositori jarak-jauh, lakukan

```
# git push origin master
```

Ubah *master* sesuai cabang yang kamu inginkan.

Jika repositori yang ada belum dikloning dan ingin dihubungkan ke server jarak-jauh, kamu perlu menambahkan

```
# git remote add origin <server>
```

Sekarang kamu bisa mengirimkan perubahan ke server jarak-jauh yang dituju

percabangan

percabangan atau *branching* digunakan untuk mengembangkan fitur-fitur secara terisolasi. Cabang utama atau *master* merupakan cabang bawaan ketika kamu membuat repositori. Gunakan cabang lain untuk pengembangan, setelah selesai, gabungkan kembali ke cabang utama.

buat cabang baru dengan nama "fitur_x" dan beralih kedalamnya menggunakan

```
# git checkout -b fitur_x
```

beralih kembali ke *master*

```
# git checkout master
```

dan hapus cabang yang tadi dibuat

```
# git branch -d fitur_x
```

suatu cabang *tidak terbuka untuk yang lainnya* kecuali jika kamu mengirimkannya ke repositori jarak-jauh.

```
# git push origin <cabang>
```

perbaru & gabung

untuk memperbarui repositori lokal ke komit terkini, lakukan

```
# git pull
```

dari direktori kerja kamu untuk *mengambil* dan *menggabungkan* perubahan jarak-jauh.

untuk menggabungkan cabang lain ke cabang aktif (misal *master*), gunakan

```
# git merge <cabang>
```

pada kasus diatas, git mencoba menggabungkan perubahan secara otomatis. Sayangnya hal ini tak selalu berjalan mulus dan bisa menyebabkan *konflik*. Kamu lah yang bertanggung jawab menggabungkan *konflik* tersebut secara manual dengan menyunting berkas yang ditunjukkan git. Setelah itu, kamu perlu memarkahnya dengan

```
# git add <namaberkas>
```

sebelum penggabungan berlaku, kamu bisa melakukan pratinjau menggunakan

```
# git diff <cabang_asal> <cabang_tujuan>
```

menandai

sangat dianjurkan membuat penanda atau *tags* untuk perangkat lunak yang dirilis. Hal ini amat lah lazim, yang juga terjadi di SVN. Kamu bisa membuat penanda baru dengan nama *1.0.0* dengan menjalankan

```
# git tag 1.0.0 1b2e1d63ff
```

1b2e1d63ff adalah 10 karakter pertama dari identitas komit yang ingin kamu referensikan ke penanda. Kamu bisa mendapatkan identitas komit dengan melihat...

log

dalam bentuknya yang paling sederhana, kamu bisa mempelajari riwayat repositori menggunakan..

```
# git log
```

kamu bisa menambahkan banyak parameter untuk menampilkan log sesuai keinginan. Untuk

melihat komit penulis tertentu:

```
# git log --author=bob
```

Untuk melihat log yang dimampatkan, satu baris per komit:

```
# git log --pretty=oneline
```

Atau mungkin kamu ingin melihat pohon *ASCII art* seluruh percabangan disertai nama dan penandanya:

```
# git log --graph --oneline --decorate --all
```

Sekedar melihat berkas yang berubah:

```
# git log --name-status
```

Ini baru sedikit saja dari sekian banyak parameter yang bisa kamu gunakan. Lebih jauh lagi, lihat

```
# git log --help
```

mengembalikan perubahan lokal

Seandainya kamu melakukan kesalahan (yang tentunya tak pernah terjadi ;) kamu bisa mengembalikannya menggunakan perintah

```
# git checkout -- <namaberkas>
```

perintah di atas mengembalikan perubahan di dalam pokok kerja kamu dengan konten terakhir dari *HEAD*. Perubahan dan berkas baru yang telah ditambahkan ke indeks akan tetap tersimpan.

Jika kamu ingin menggugurkan perubahan dan komit lokal seutuhnya, ambil riwayat terakhir dari server dan arahkan ke cabang *master* lokal seperti ini

```
# git fetch origin
```

```
# git reset --hard origin/master
```

GUI git bawaan

```
# gitk
```

menggunakan output git penuh warna

```
# git config color.ui true
```

menunjukkan log satu baris per komit

```
# git config format.pretty oneline
```

menggunakan penambahan interaktif

```
# git add -i
```

MENGGUNAKAN GITKRAKEN UNTUK MEMUDAHKAN KELOLA GIT

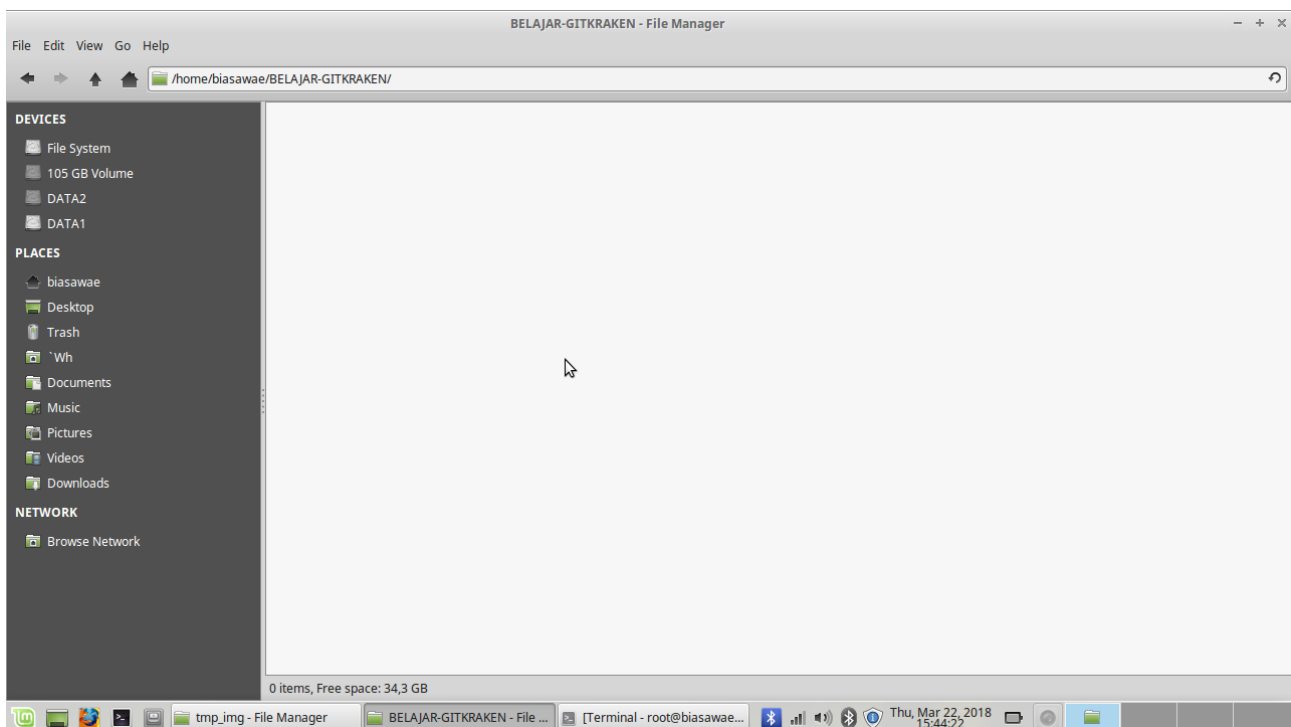
Selain menggunakan terminat, mengelola GIT juga bisa menggunakan aplikasi desktop. Diantara banyak pilihan, GITKRAKEN sangat bagus juga untuk memudahkan melakukan upload (push) dan download (pull).

Silahkan bisa unduh di :

<https://www.gitkraken.com/>

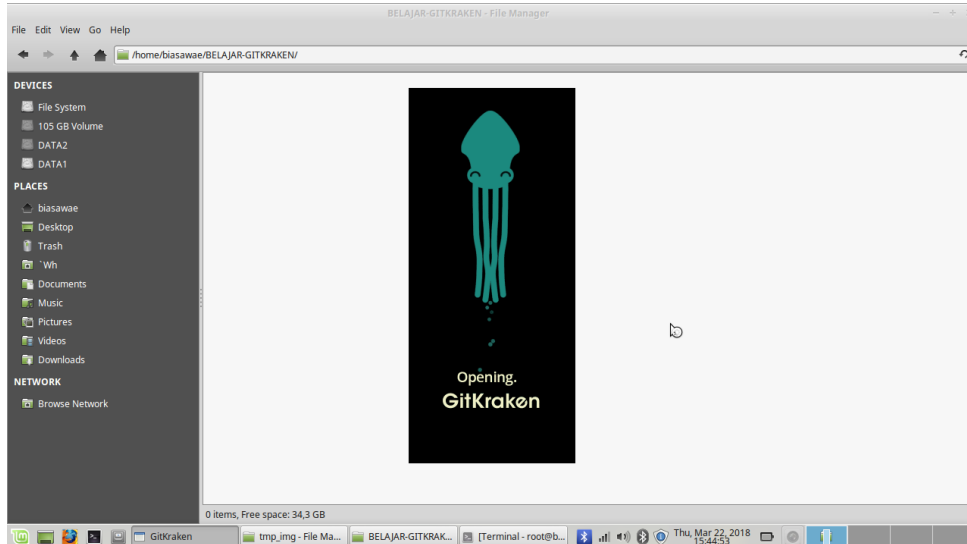
Setelah melakukan instalasi, langkah berikutnya adalah melakukan setting GITKRAKEN terhadap akun github kita yang telah kita buat sebelumnya.

1. Buatlah sebuah folder proyek yang masih kosong. Misalkan “BELAJAR-GITKRAKEN”



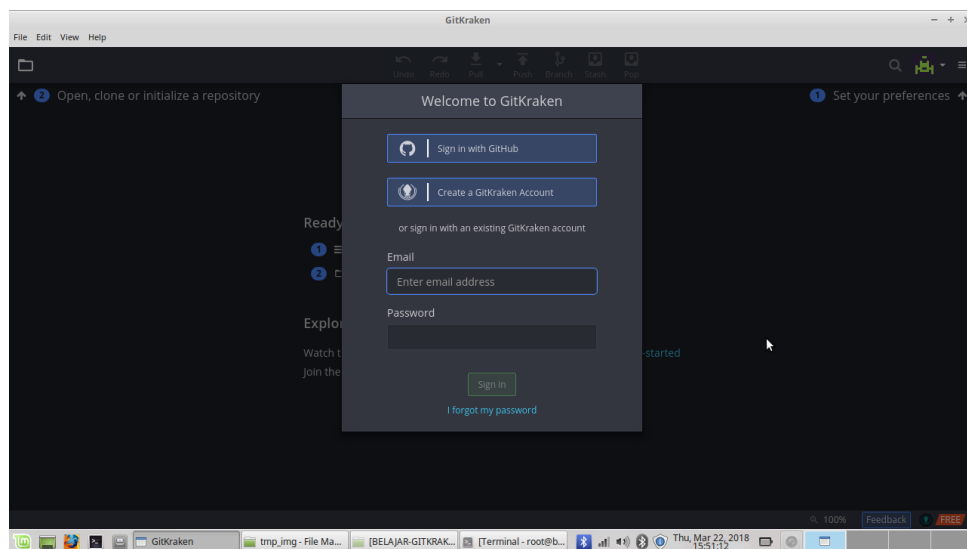
Tampilan Folder Proyek yang masih kosong.

2. Jalankan Aplikasi GITKRAKEN



Tampilan Splash Screen dari GITKRAKEN

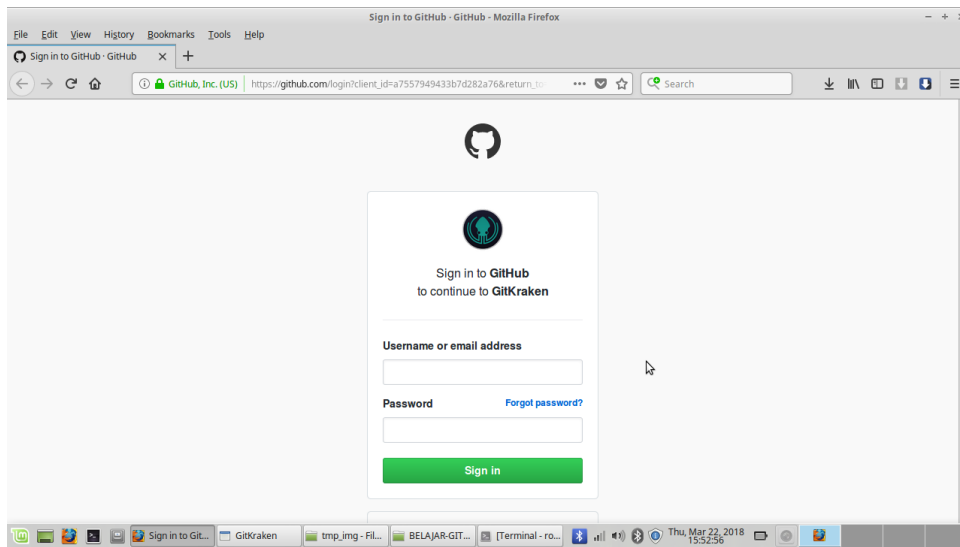
3. Tampilan Halaman Awal GITKRAKEN yang belum setting.



Tampilan Halaman Awal GITKRAKEN.

Silahkan pilih “Sign In with Github”

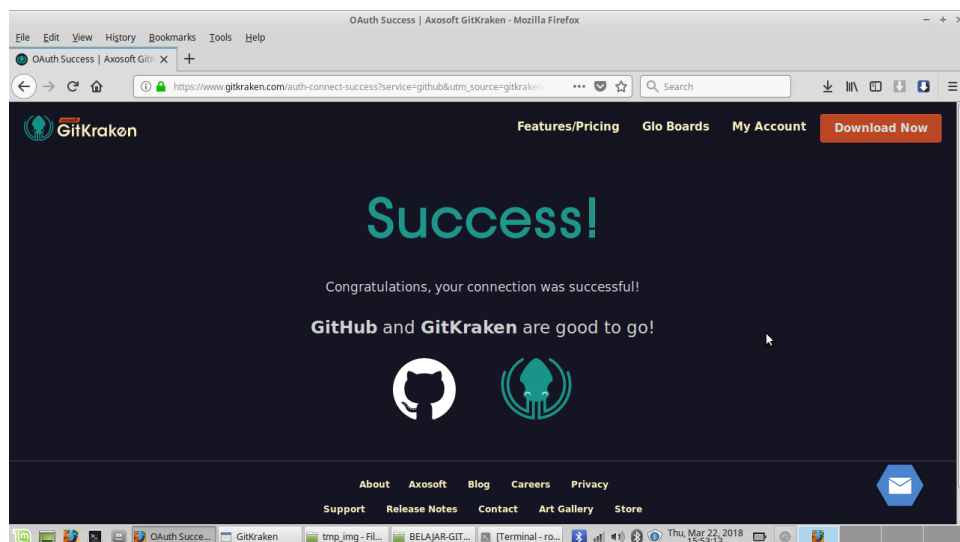
4. Tampilan Halaman Login Github



Tampilan Halaman Login Github.

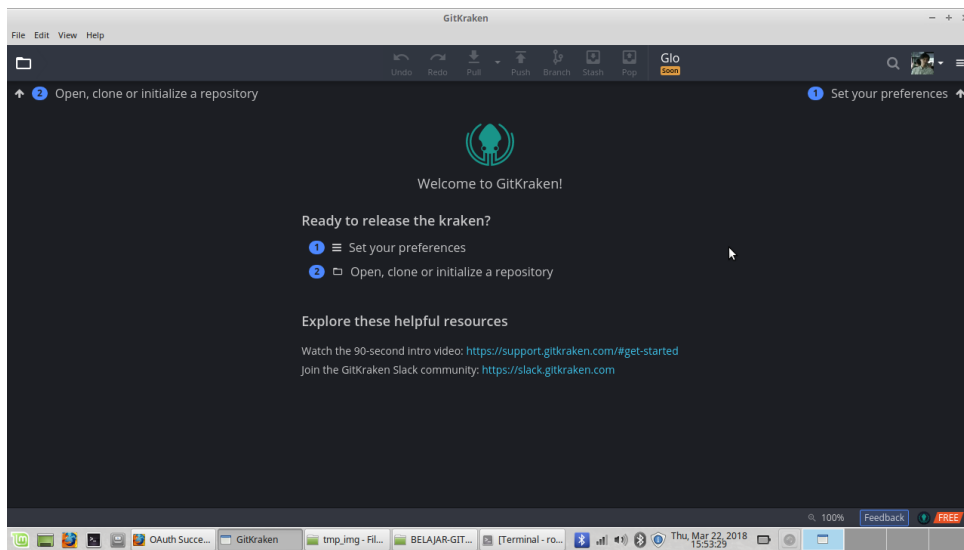
Silahkan isi dengan username dan password yang telah kita buat.

5. Login Berhasil



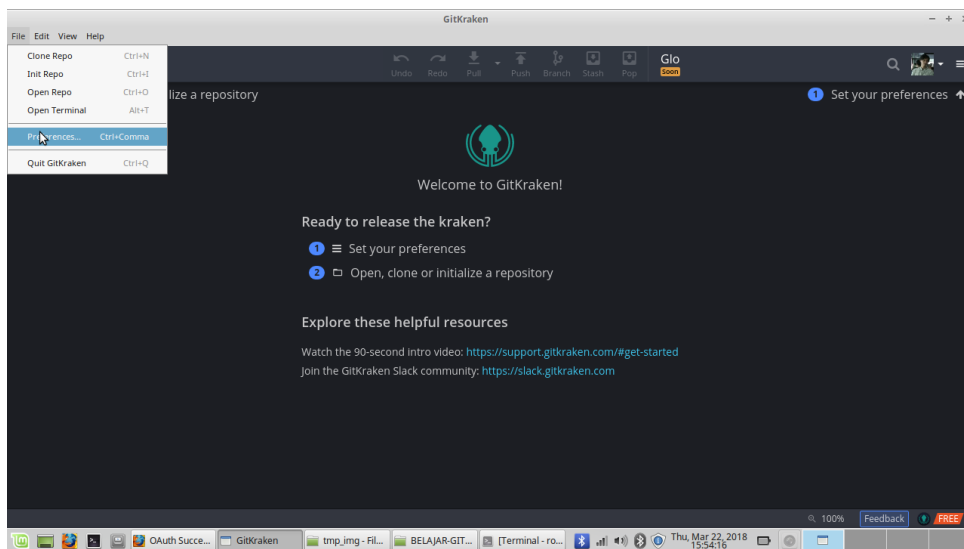
Tampilan Halaman Login Github
yang berhasil integrasi dengan GITKRAKEN

6. Halaman Utama GITKRAKEN



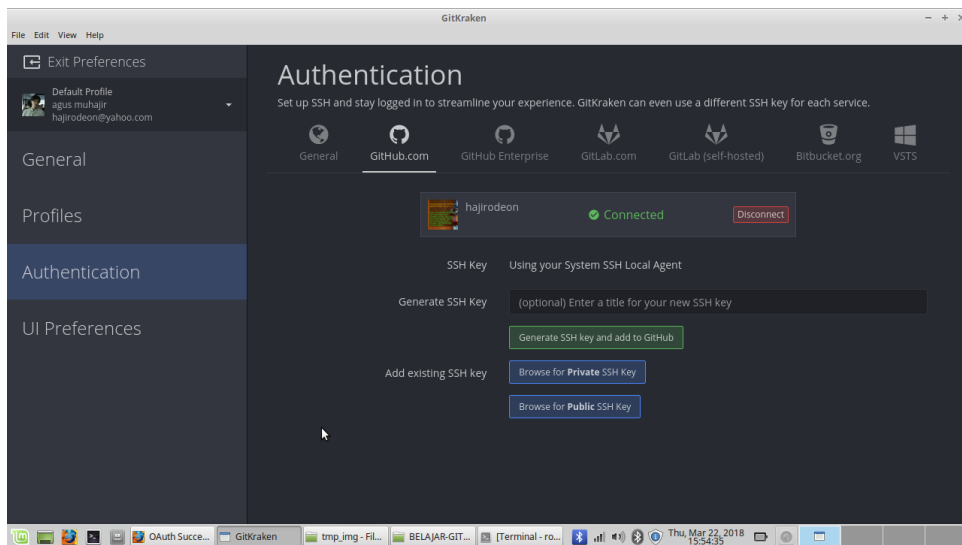
Tampilan Halaman Utama GITKRAKEN

7. Silahkan setting dengan masuk ke menu Preferences.



Tampilan Menu Preferences.

8. Tampilan Halaman Preferences.

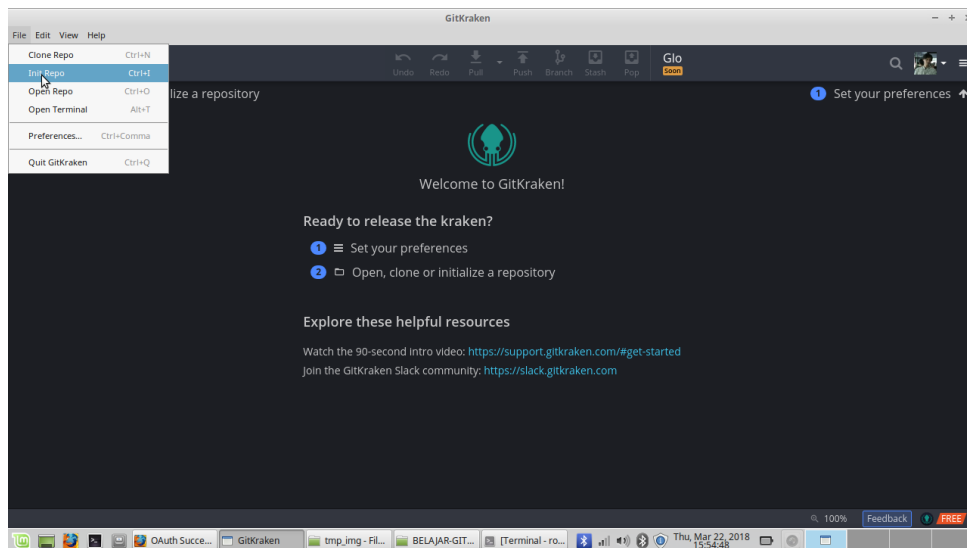


Tampilan Halaman Preferences.

Silahkan masuk pada menu Authentication.

Pastikan Connected dengan github.com

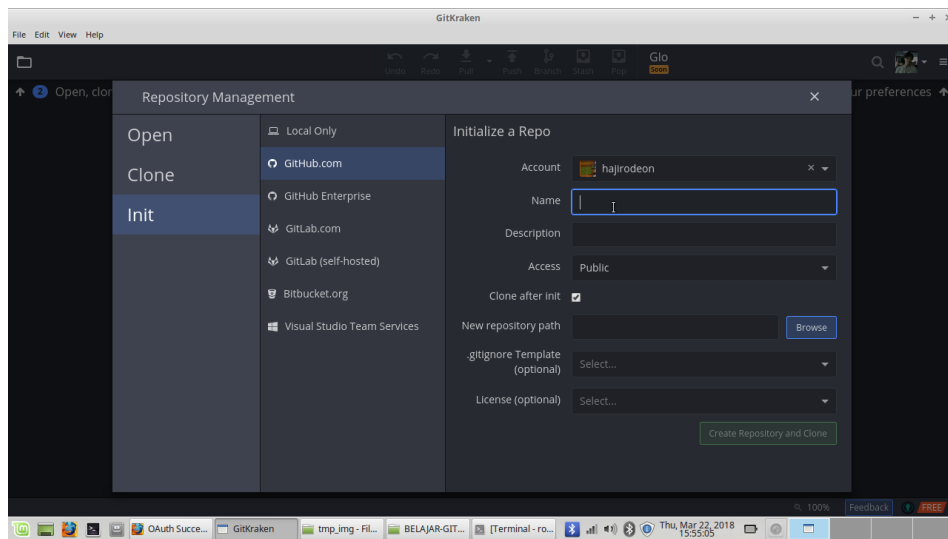
9. Membuat Proyek GIT



Tampilan Halaman Menu Pilih Init Repo.

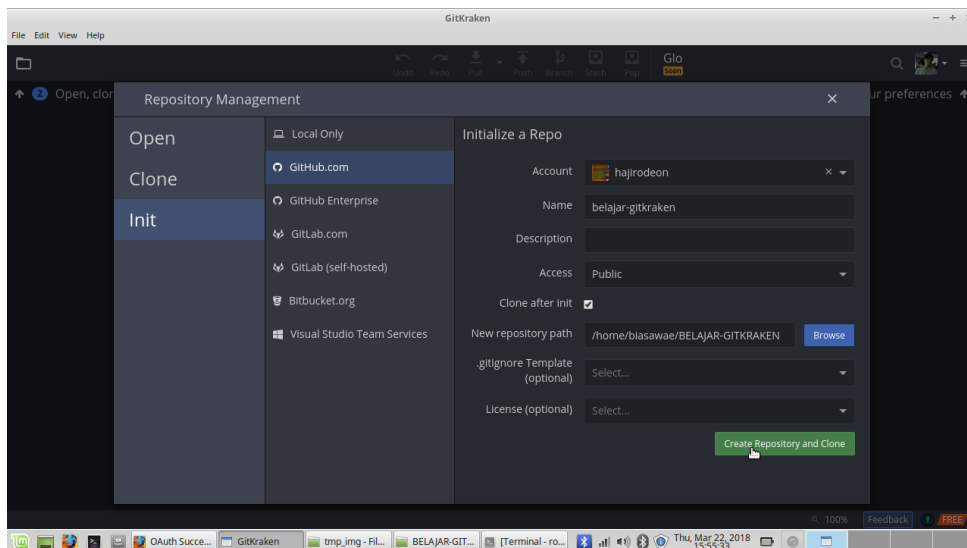
Untuk membuat proyek GIT, silahkan klik menu Init Repo.

10. Detail Pembuatan Proyek GIT pada Github.com



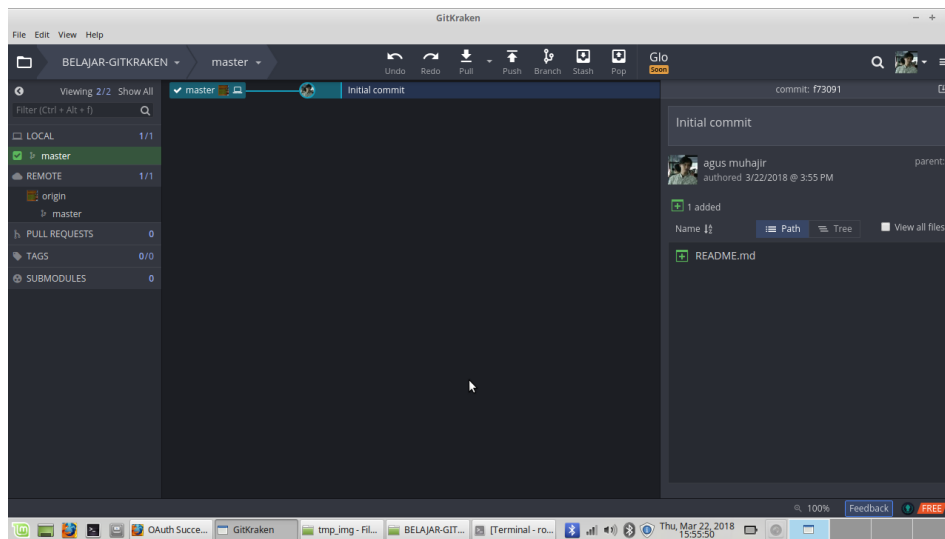
Tampilan Halaman Detail Pembuatan Proyek GIT

Silahkan pilih “Account”, kemudian “name” Proyek Repository,
Kemudian “Browse” untuk “New Repository Path” Lokasi folder kita di lokal. Yakni
“/home/biasawae/BELAJAR-GITKRAKEN/”



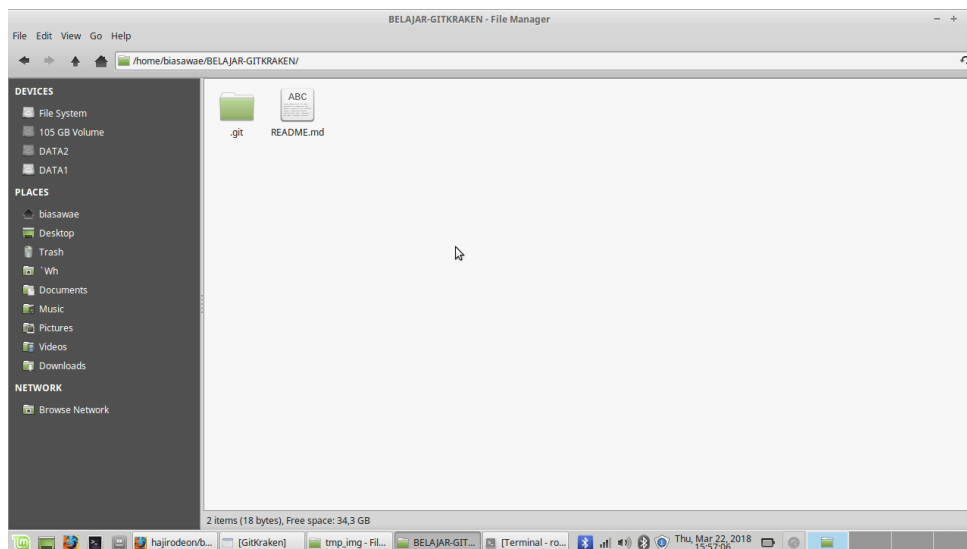
Kemudian klik “Create Repository and Clone”

11. Tampilan Halaman Detail Repository

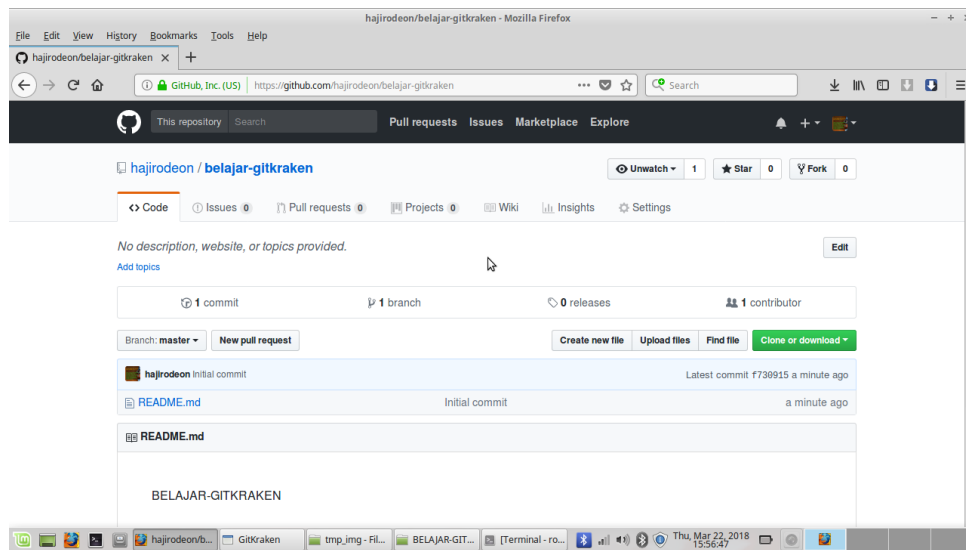


Tampilan Halaman Detail Repository.

Terlihat disini ada sebuah file bernama “README.md”



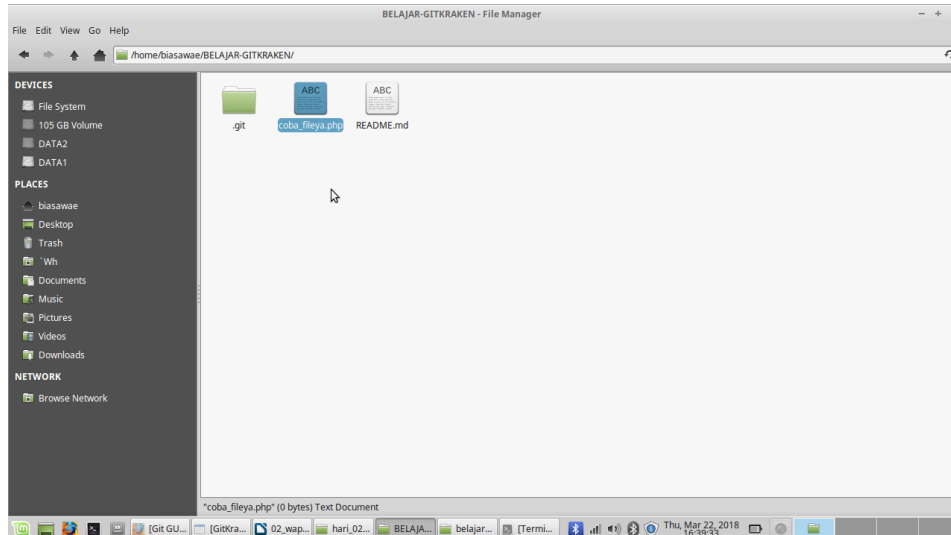
Tampilan Folder Detail Repository di lokal.



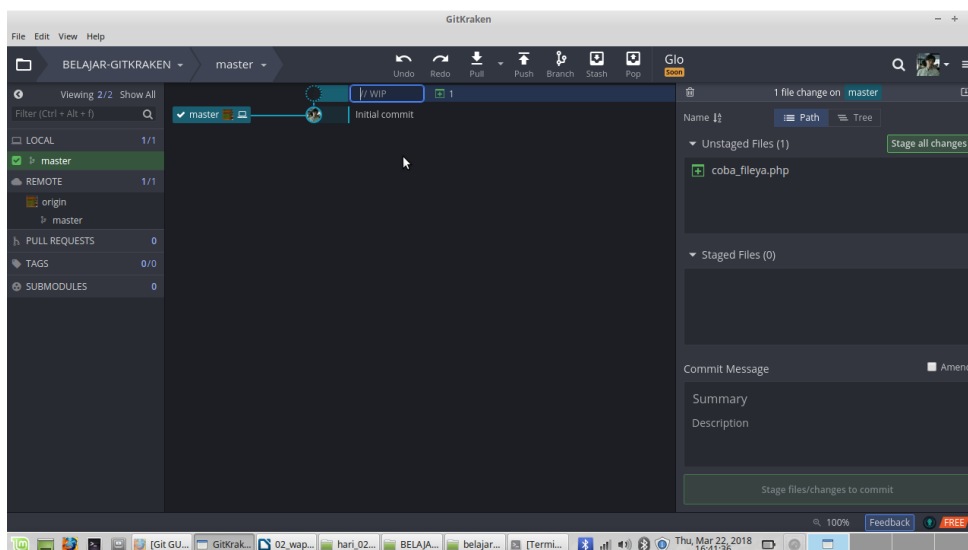
Tampilan Detail Repository di Github

UPLOAD atau PUSH

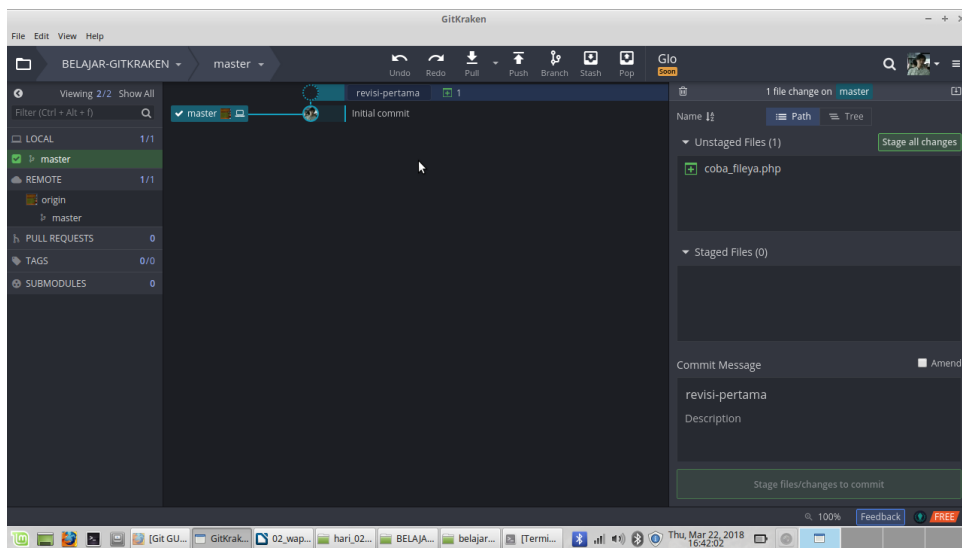
Buatlah sebuah file text pada folder proyek. Misalkan “coba_fileya.php”



Sebuah file yang kita letakkan pada folder “BELAJAR-GITKRAKEN”



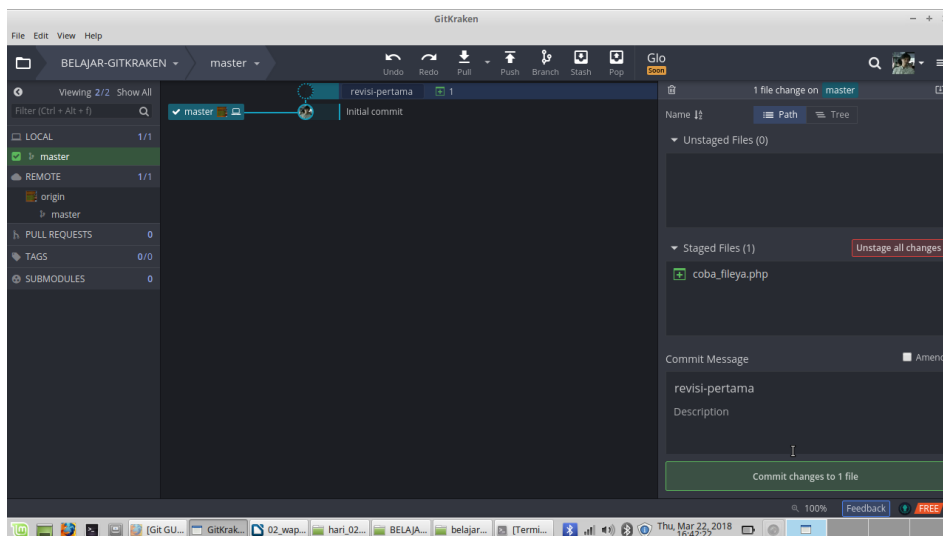
Kita cek di GITKRAKEN, diatas “initial commit”
ada update baru yang perlu kita beri nama commit nya.



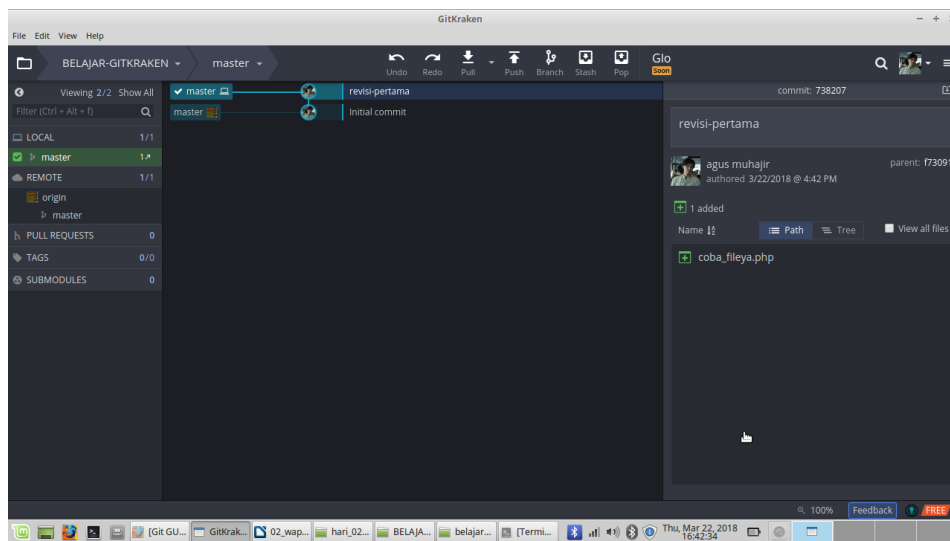
Kita beri nama commit “revisi-pertama”.

Disini terlihat “stage”, file apa saja yang berubah.

Silahkan klik “stage all change”, untuk masuk dalam daftar target yang akan kita upload / push.



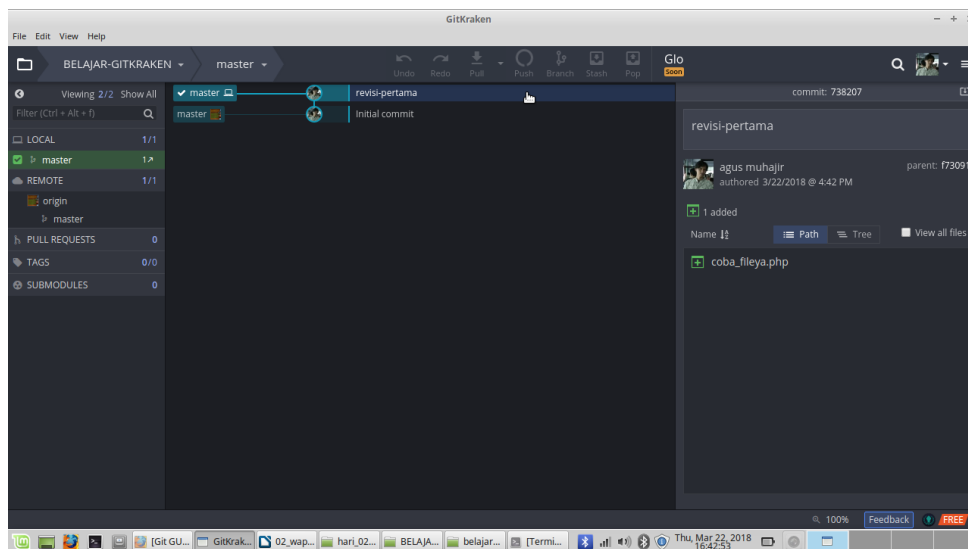
Setelah masuk daftar, silahkan commit changes.



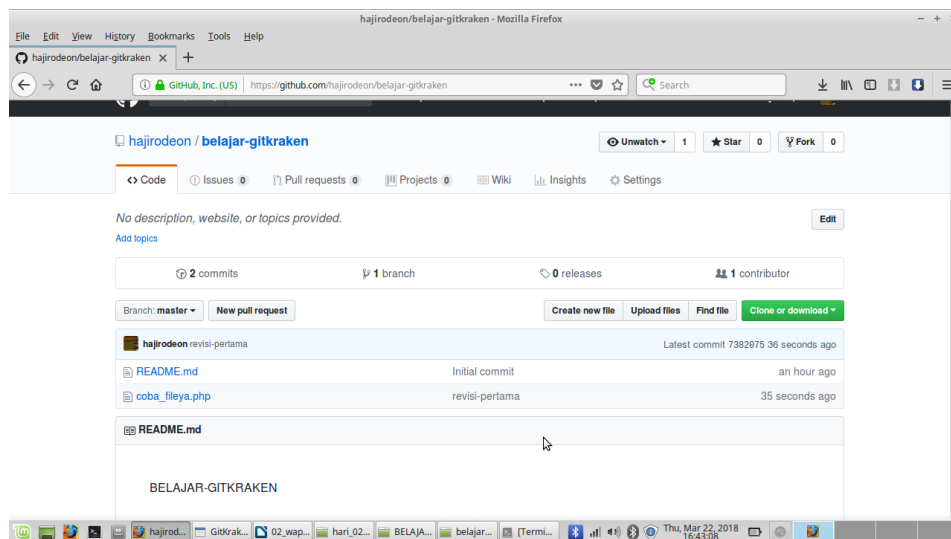
Setelah berhasil commit, saatnya kita upload / push.

Pada menu atas tengah, ada panah keatas, push.

Silahkan klik.



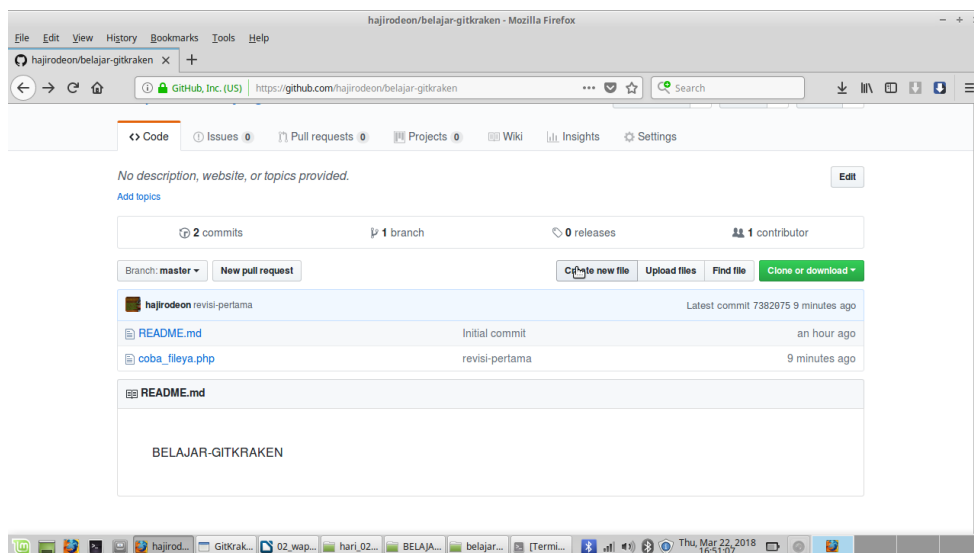
Proses Push sedang berlangsung

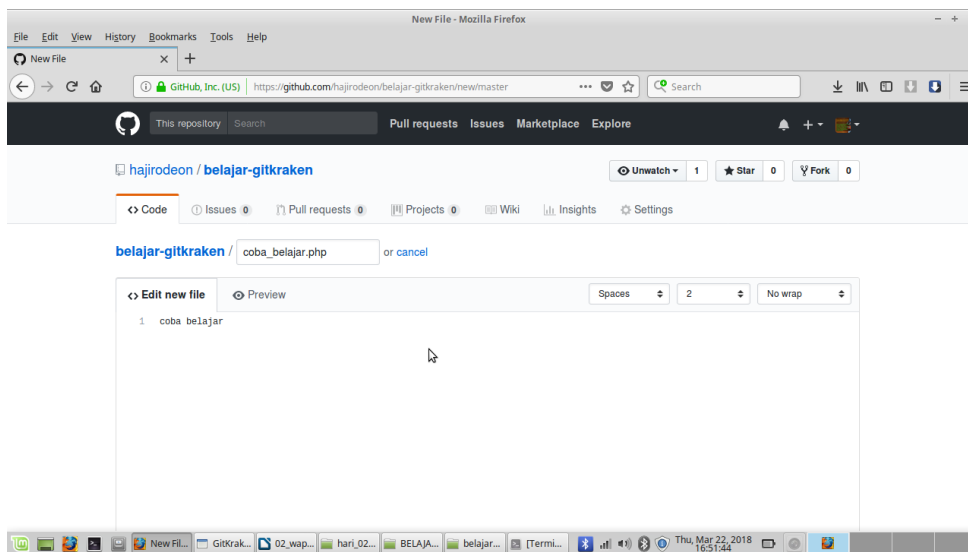


Kita cek di halaman github, upload telah berhasil.

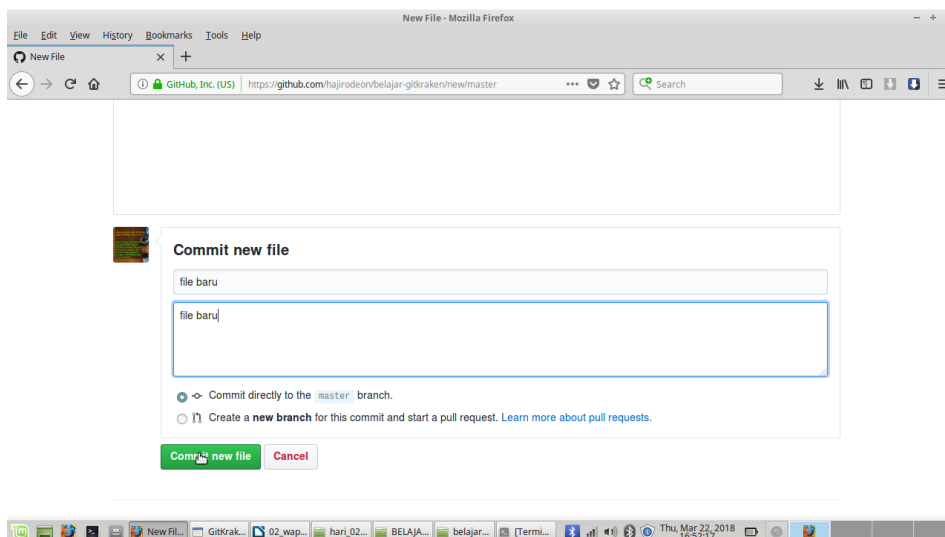
DOWNLOAD atau PULL

Misalkan kita membuat sebuah file baru pada github kita.



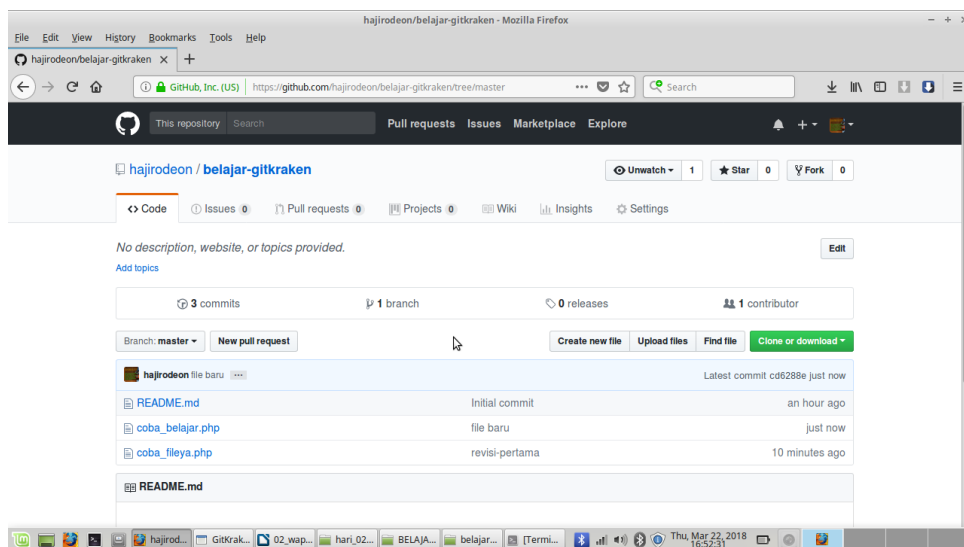


Silahkan beri nama “coba_belajar.php”

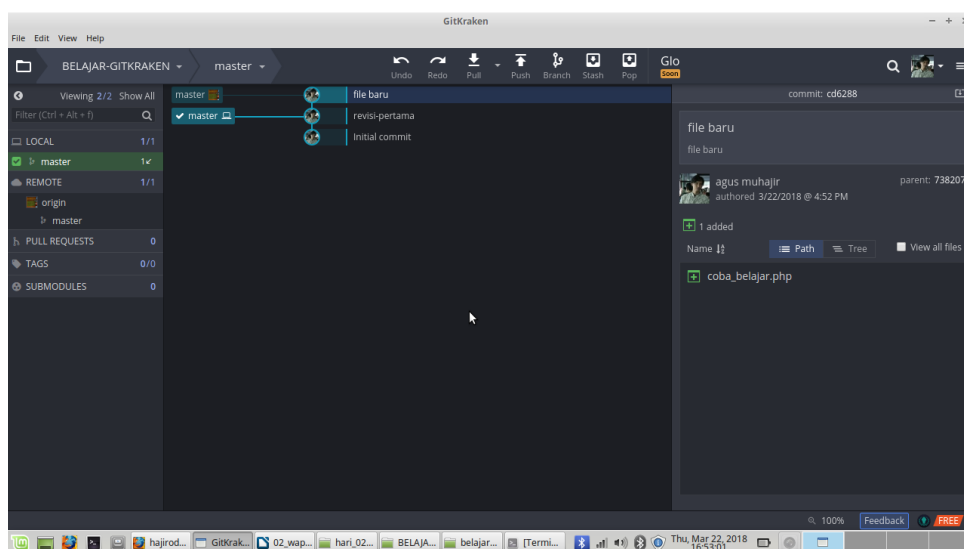


Beri judul dan deskripsi dari commit.

Disini kita beri judul dan deskripsi “file baru”.

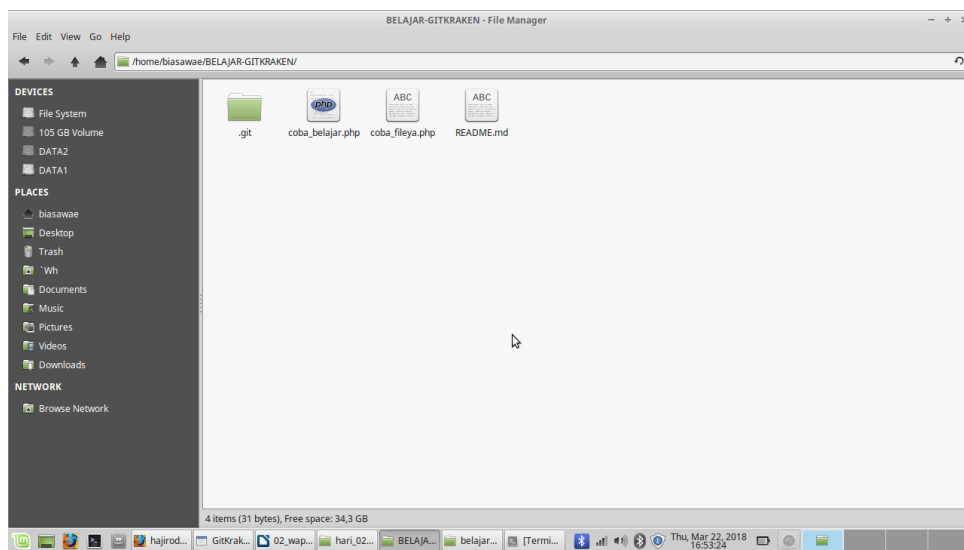


Pada daftar file di Proyek GIT kita sudah ada “coba_belajar.php”



Kita lihat di GITKRAKEN, commit “file baru”, telah muncul seketika.

Selanjutnya kita download / pull apa yang berubah dari github.



Pada folder “BELAJAR-GITKRAKEN”, akan muncul file “coba_belajar.php”

TUGAS.

Silahkan membuat repository, yang isinya beberapa file dan folder. Yang nantinya akan upload / push ke github.

Begitu juga bila melakukan perubahan secara online di situs github, download / pull, agar perubahan yang ada, masuk juga ke repository lokal.

DAFTAR PUSTAKA :

<http://github.com>

<http://d3f4lt.github.io/pengetahuan/post/2014/06/10/kenalan-dengan-github/>

<https://www.candra.web.id/pengenalan-github-untuk-pemula/>

<http://rogerdudler.github.io/git-guide/index.id.html>

<http://gitkraken.com/>

<http://github.com/hajirodeon>