

Manual Pengguna

Penggunaan GitHub Desktop (GUI)



Oleh Zulkifli Mohamed

Dokumen ini disediakan pada waktu dan ketika versi GitHub Desktop semasa. Oleh itu, proses pemasangan dan cara penggunaannya adalah bergantung kepada versi semasa pada masa-masa akan datang. Mungkin ada kandungan yang tidak sah pada masa anda lakukan pemasangan.



Langkah permulaan bagi menggunakan GitHub Desktop

Dapatkan GitHub Desktop bagi memudahkan anda untuk menguruskan kerja-kerja projek anda.

Apa itu Git ?

Git adalah *tools* yang berfungsi sebagai *Version Control System (VCS)* dan kalau diertikan ke bahasa kita ertinya sebuah sistem pelacak perubahan pada file. Git sendiri dibuat oleh orang yang menciptakan *Kernel Linux*, yup... tidak salah lagi dia adalah Linus Torvalds.

Pada awalnya Git digunakan untuk membantu pengembangan Kernel Linux sebelum akhirnya dipakai lebih luas di berbagai pengembangan software lainnya dan dibuat multi-platform sehingga tersedia di hampir semua *OS*. Selain Git ada beberapa alternatif lain dari sistem VCS, contohnya: Subversion, Mercurial, CVS, dan lain-lain.











Kenapa Git penting ?

Kenapa Git penting untuk tim developer ? Dengan menggunakan Git, setiap orang dalam sebuah tim dapat melakukan perubahan pada *source-code* tanpa harus takut terjadi perselisihan ataupun kesulitan dalam menggabungkan hasil perubahan yang mereka lakukan. Dengan menggunakan Git, setiap perubahan pada *source-code* akan direkodkan perubahannya, apa saja yang diubah, siapa yang mengubah dan bila waktunya.

Ok cukup untuk pengenalan tentang Git, sekarang kita masuk ke fungsi dan cara penggunaan dasar dari Git itu sendiri.

Perintah arahan asas Git ?

Perintah-perintah asas dari Git yang paling sering digunakan dan wajib kita kuasai iaitu: **git init**, **git status**, **git add**, **git commit**, **git pull**, **git push**, **git checkout**, **git diff**, **git merge**, **git remote** dan **git reset**. Sebetulnya masih banyak fungsi lain dari Git untuk kegunaan tertentu tapi mungkin perintah yang telah disebutkan sebelumnya yang akan paling sering kita gunakan, berikut ini adalah penjelasan singkat mengenai arahan tersebut:

-  **git init** - untuk membuat repo local baru pada perintah ini akan dibuat sebuah folder baru yang bernama ".git"
-  **git status** - untuk melihat status dari repo local
-  **git add** - untuk menambahkan file ke dalam repo yang sebelumnya sudah dibuat/ubah
-  **git commit** - untuk menyimpan seluruh perubahan yang terjadi
-  **git pull / push** - untuk menyimpan dan mengambil data dari *remote repo*
-  **git checkout** - untuk pindah *branch*
-  **git diff** - untuk membandingkan perubahan *file*
-  **git merge** - untuk melakukan penggabungan antara *branch*
-  **git remote** - untuk menambahkan/mengubahsuaikan *remote* repo baru
-  **git reset** - untuk membatalkan perubahan pada repo local



ISI KANDUNGAN

Setting GitHub Desktop

Pemasangan GitHub Desktop.....	4
Akses masuk ke akaun GitHub & Konfigurasi Git.....	5

Kolaborasi Projek dengan GitHub Desktop

Menambah dan kaedah *cloning* repositori GitHub

Menambah repositori sediada melalui GitHub Desktop.....	7
<i>Cloning</i> repositori melalui GitHub Desktop.....	9
<i>Cloning</i> repositori dari laman GitHub.....	10

Melakukan perubahan pada *branch* projek

Wujudkan <i>branch</i> untuk kerja anda.....	11
Mengenai <i>commits</i>	12
Melihat perubahan dan <i>committing</i> yang telah dilakukan pada projek.....	13
Pembatalan <i>commit</i>	16

Melihat sejarah projek anda

Melihat mengenai perbezaan graf.....	17
--------------------------------------	----

Bekerja dengan repositori *remote* anda di GitHub

Mengemaskini <i>branch</i> anda.....	18
Menghantar <i>pull request</i>	19
Menukar URL remote melalui GitHub Desktop.....	21

References dan Sumber Rujukan.....	22
------------------------------------	----



Setting GitHub Desktop

Pemasangan GitHub Desktop

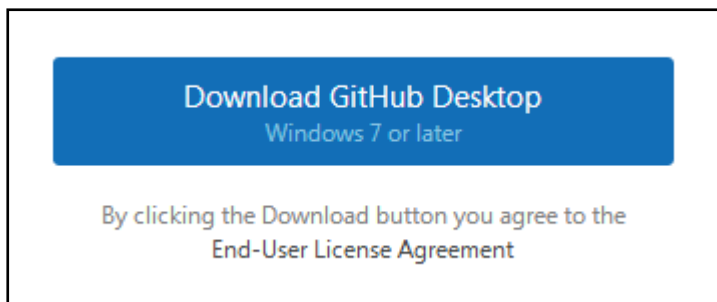
Anda boleh memasang GitHub Desktop di atas platform Microsoft Windows atau sistem operasi OS X.

Sebelum anda membuat pemasangan GitHub Desktop, pastikan anda telah mempunyai akaun di GitHub.

Muat Turun Perisian & Pemasangan

Anda boleh memasang GitHub Desktop pada sistem operasi Windows 7 ke atas.

1. Muat Turun perisian GitHub Desktop di sini <https://desktop.github.com>
2. Pilih **“Download GitHub Desktop”**



3. Selepas selesai muat turun, dwi-klik **“GitHubSetup.exe”**

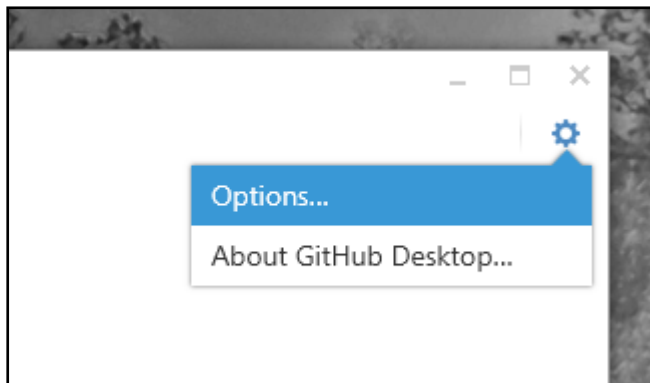


Setting GitHub Desktop

Akses masuk ke akaun GitHub dan Konfigurasi Git

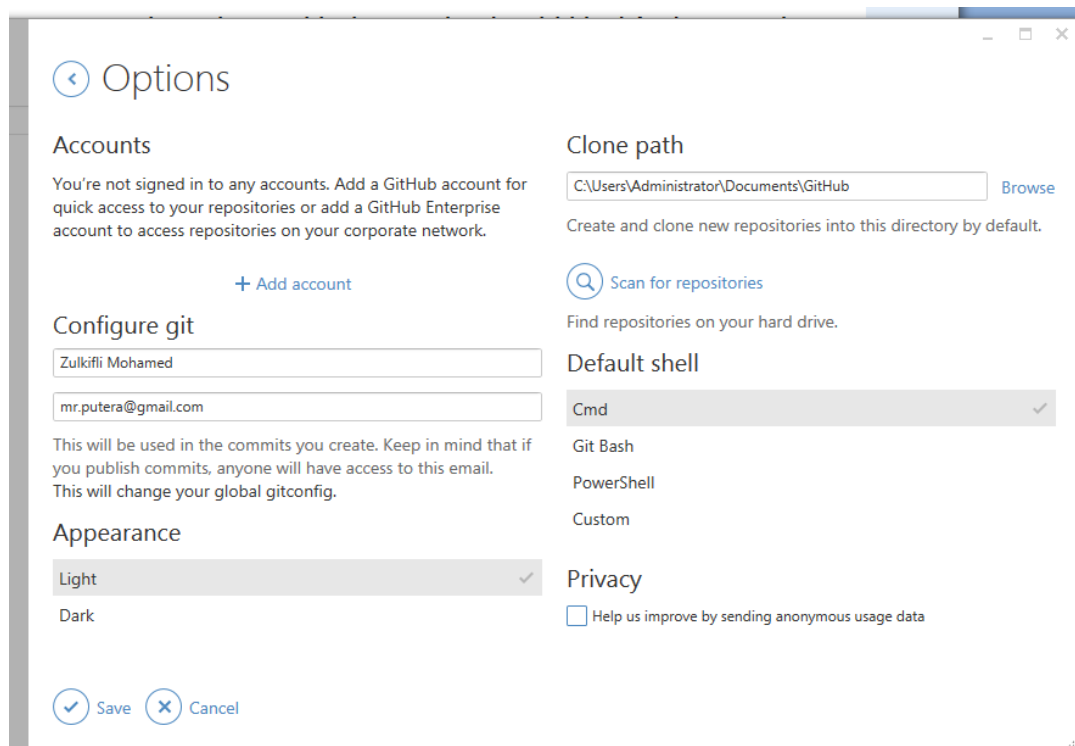
Untuk mengakses masuk ke akaun GitHub, anda perlulah berdaftar dengan GitHub.com. Sila daftarkan diri anda di www.github.com (jika masih belum mendaftar)

Buka GitHub Desktop



Klik pada ikon **“Gear”** dan pilih **“Options...”**

Tetingkap Options akan muncul seperti dibawah.



Bahagian **“Accounts”**, Tekan **“+ Add account”**



Log in

GitHub GitHub Enterprise

mr.putera@gmail.com

.....

The best way to build and ship software. Go to github.com to sign up for an account

Log in Cancel

Masukkan alamat e-mel anda (akaun GitHub) dan kata laluan dan tekan “**Log in**”

Bahagian “**Configure git**”, masukkan nama dan alamat e-mel anda (akaun GitHub) seperti dibawah

Configure git

Zulkifli Mohamed

mr.putera@gmail.com

This will be used in the commits you create. Keep in mind that if you publish commits, anyone will have access to this email. This will change your global gitconfig.

Bahagian “**Clone path**”, anda boleh memilih lokasi *default* apabila anda membuat *clone* repositori pada masa akan datang.

Clone path

C:\Users\Administrator\Documents\GitHub Browse

Create and clone new repositories into this directory by default.



Kolaborasi Projek dengan GitHub Desktop

Menambah dan kaedah *cloning* repositori GitHub

Menambah repositori sediaada melalui GitHub Desktop

Terdapat **2 (dua)** cara untuk menambah repositori projek anda menggunakan GitHub Desktop.

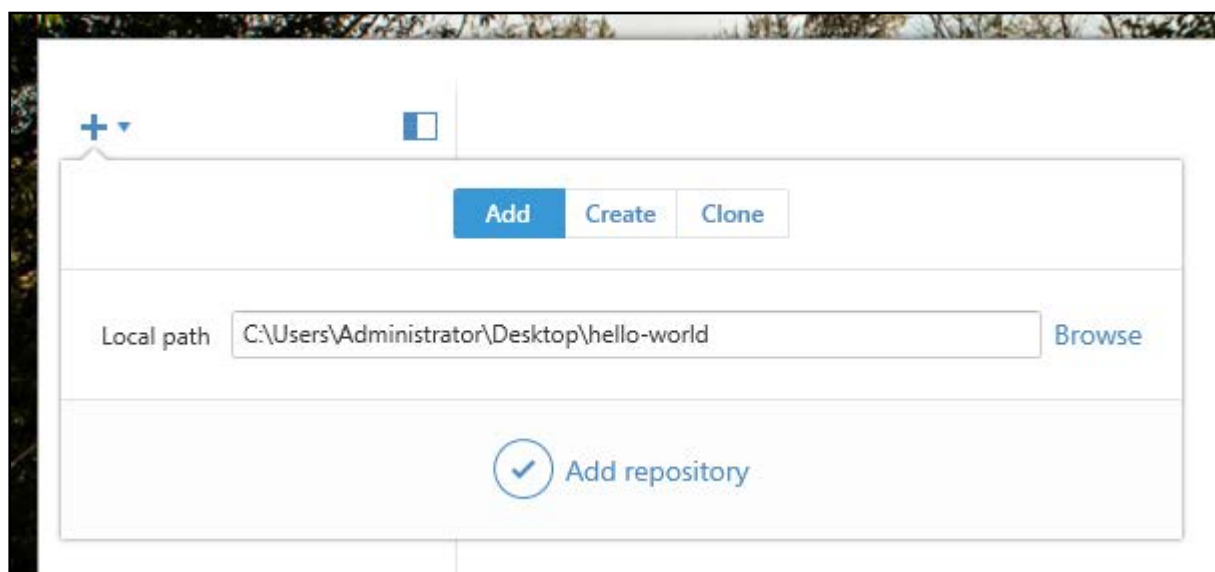
Kaedah #1 (Projek Sediada)

Sekiranya anda telah mempunyai repositori projek sebelum ini di komputer anda, anda boleh menguruskan projek tersebut melalui GitHub Desktop GUI.

****** Ianya sangat membantu, kerana anda tidak lagi perlu menghafal *command* arahan git push, git pull dan sebagainya untuk lakukan semua proses perubahan pada projek anda.

Cara-cara nya adalah seperti berikut :-

1. Contoh: *Folder “hello-world”* di *Desktop* saya adalah projek saya sebelum ini dan ianya menggunakan git untuk proses penambahbaikkkan. Saya ingin menggunakan GitHub Desktop untuk menguruskan semua projek *hello-world* ini.
2. Buka **GitHub Desktop**
3. Klik pada butang “+” dan “Add”. Klik “Browse” untuk cari *folder* projek anda. Dan tekan “Add repository” untuk tambah. Selesai !



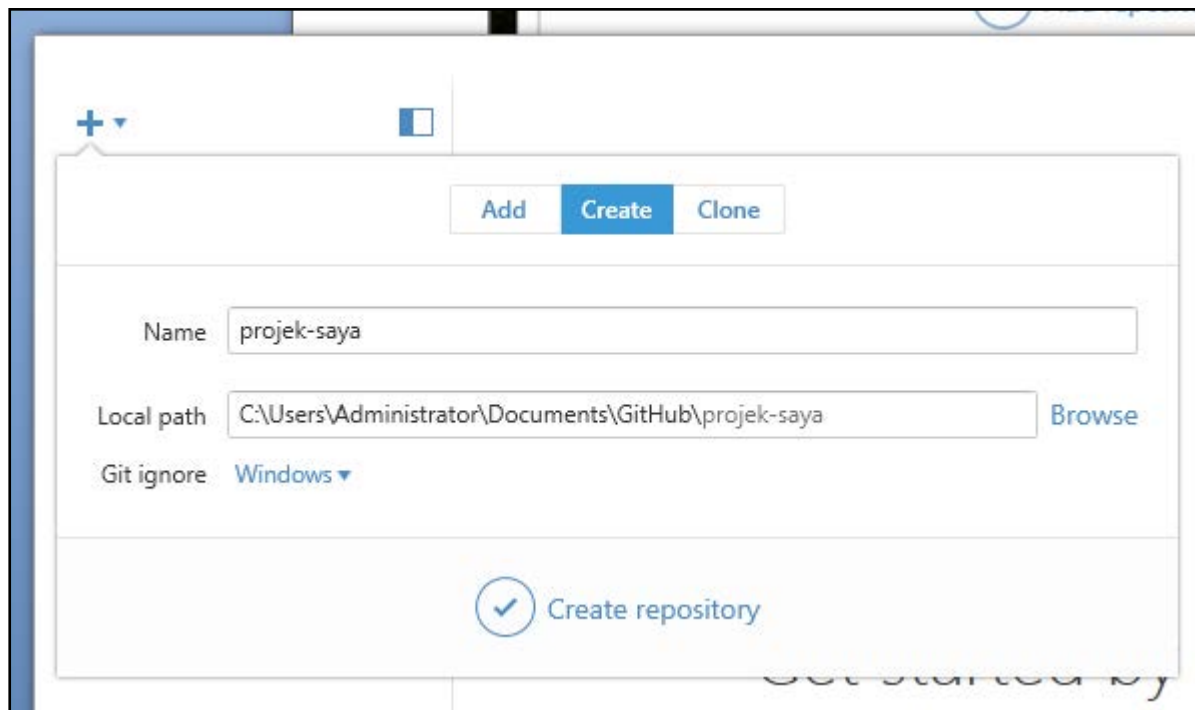


Kaedah #2 (Projek Baru)

Anda boleh membuat repositori baru untuk projek baru anda melalui GitHub Desktop.

Cara-cara nya adalah seperti berikut :-

Klik pada butang “+” dan “**Create**”. Masukkan nama projek anda diruangan “**Name**”. Klik “**Browse**” untuk set dimana lokasi *folder* projek anda dan tekan “**Create repository**”.
Selesai !





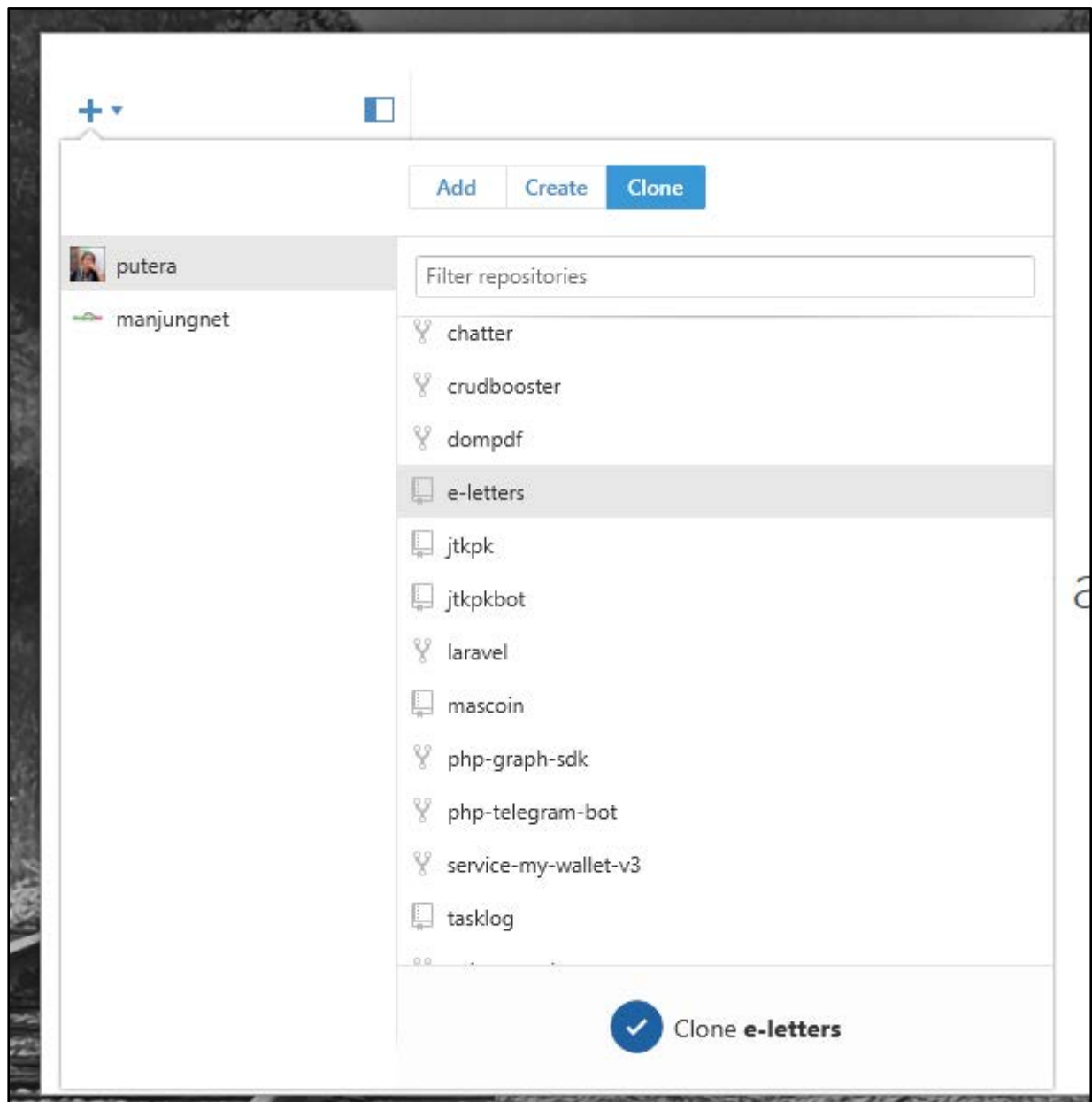
Menambah dan kaedah *cloning* repositori GitHub

Cloning repositori melalui GitHub Desktop

Sekiranya projek anda berada di GitHub.com (atau anda telah *fork*) daripada repositori orang lain. Anda boleh membuat salinan (*clone*) ke komputer anda untuk melakukan perubahan pada projek anda.

Cara-cara untuk *cloning* adalah seperti berikut :-

1. Untuk *clone* projek anda, Tekan “+” dan klik “**Clone**” dan pilih projek anda. Kemudian, tekan “**Clone <nama-projek>**”.





Menambah dan kaedah *cloning* repositori GitHub

Cloning repositori dari laman GitHub

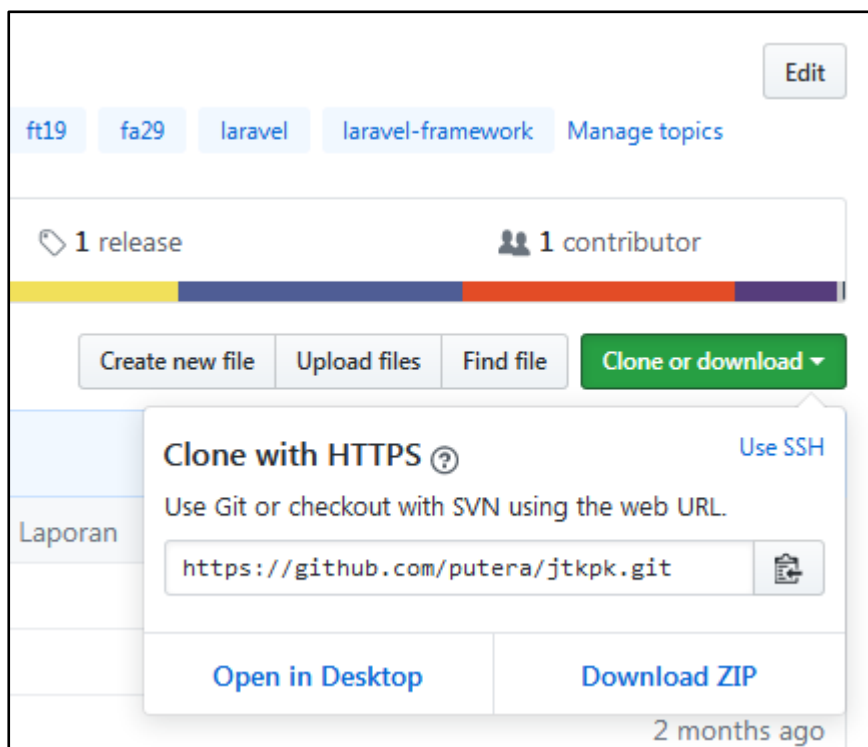
Untuk membuat *cloning* melalui laman web GitHub, sila ikuti langkah berikut. Sebagai contoh, saya ingin membuat salinan (*cloning*) projek daripada URL ini

Contoh:

URL Projek : <https://github.com/putera/jtkpk>

Langkah-langkahnya adalah seperti berikut :-

1. Buka *browser* dan terus ke URL Projek diatas

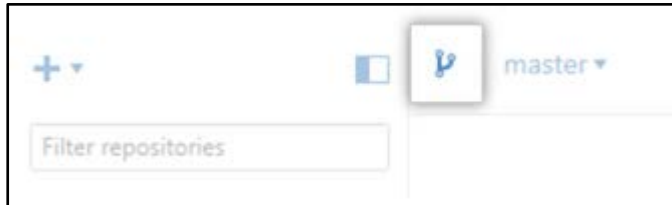


2. Klik pada **“Open in Desktop”**. Perisian GitHub Desktop akan muncul dan pilih lokasi di mana projek anda hendak disimpan.

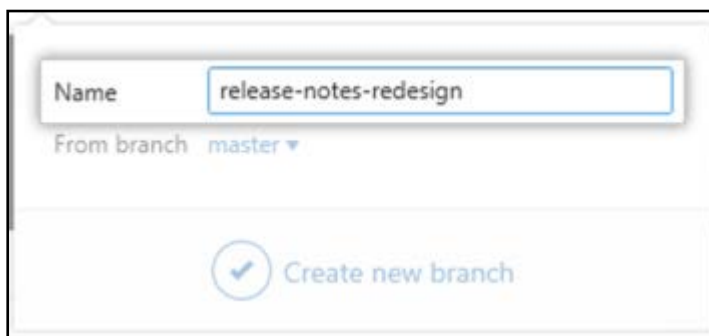


Melakukan perubahan pada *branch* proyek

Wujudkan *branch* untuk kerja anda

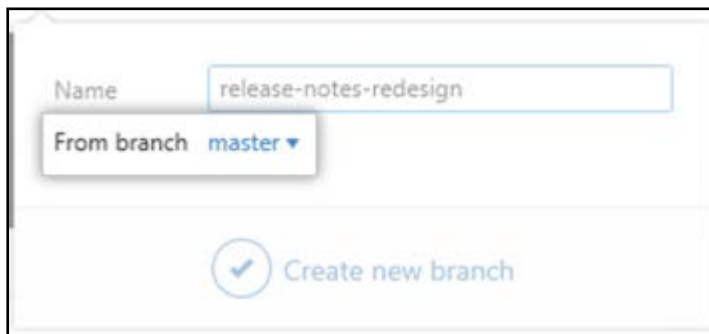


Untuk membuat *branch* baru, klik pada ikon seperti di atas

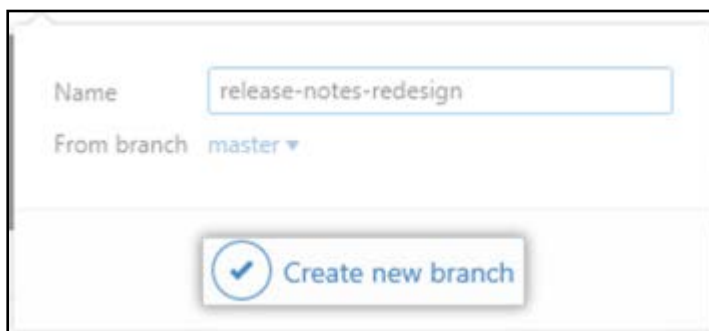


Masukkan nama *branch* dibagian “**Name**”

****Pemilihan nama untuk branch juga penting seperti contoh, anda boleh gunakan seperti “Versi 1”, atau “Versi 2.0 - Beta” dan sebagainya.**



Kemudian, pilih “**master**” dari *From branch*.



Klik “**Create new branch**” untuk wujudkan *branch*.



Melakukan perubahan pada *branch* projek

Mengenai *commits*

Selepas anda melakukan *cloning* repositori, anda boleh melakukan sebarang perubahan kod, atau apa sahaja perubahan terhadap projek anda dengan menggunakan fungsi *commits*.

Ianya sama maksud seperti “*saving*” fail. Atau simpan fail yang telah diubah. Setiap *commit* akan diwujudkan satu “**SHA**” atau “**hash**” ID untuk kita *track* setiap perubahan yang telah dilakukan. Ianya akan merekodkan maklumat seperti :-

- Perubahan apa yang telah dilakukan dan fail apa yang terlibat.
- Bila perubahan ini dilakukan
- Dan Siapa yang membuat perubahan tersebut.

Apabila anda melakukan sebarang *commit*, anda boleh meletakkan nota atau mesej mengenai apa yang telah dilakukan oleh perubahan tersebut.

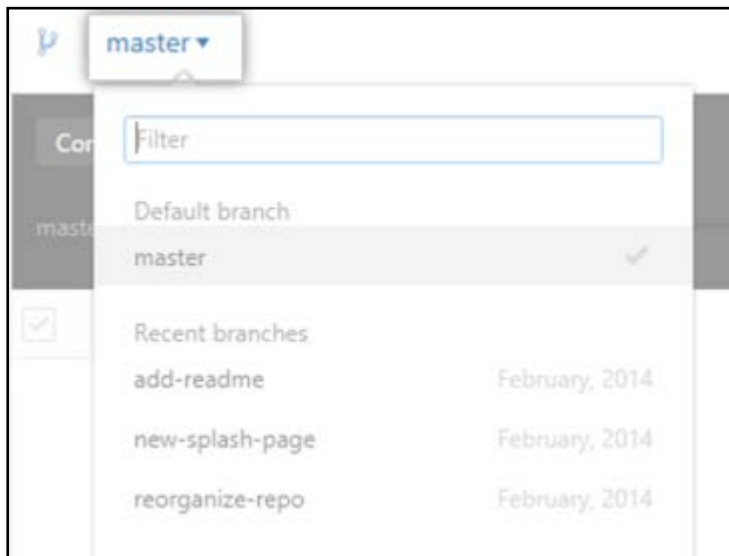


Melakukan perubahan pada *branch* proyek

Melihat perubahan dan *committing* yang telah dilakukan pada proyek

GitHub Desktop *track* setiap perubahan dan semua fail yang terlibat apabila anda mengubahnya. Anda boleh menguruskan fail-fail yang terlibat bagi setiap perubahan yang ingin dilakukan.

1. Pilih “*branch*” dan lakukan perubahan



Pilih “**master**” untuk melakukan perubahan.

**** Biasanya kita akan menggunakan branch “*master*” untuk membuat perubahan kod**

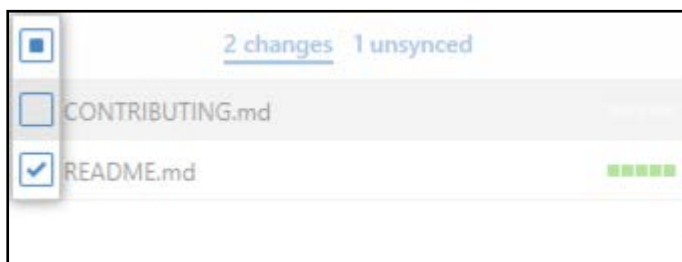
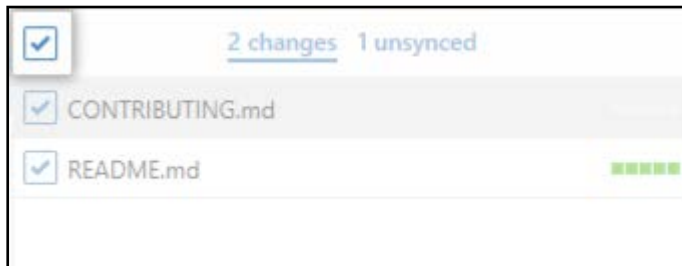
**** Anda boleh melakukan sebarang perubahan pada projek anda dengan menggunakan aplikasi text editor seperti Notepad, Dreamweaver, Sublime dan sebagainya untuk mengubah *coding* anda.**



2. Pilih fail yang berkaitan untuk lakukan proses *commit*

Setelah anda membuat perubahan pada projek anda, samada ubah kod, update gambar, tambah fail dan sebagainya. Anda akan dapat melihat sebarang perubahan yang berlaku di GitHub Desktop.

Untuk melakukan proses “***commit***” iaitu dalam erti kata lainnya “***save***”, Pilih “***tick***” *checkbox* tersebut untuk memilih fail-fail yang terlibat.

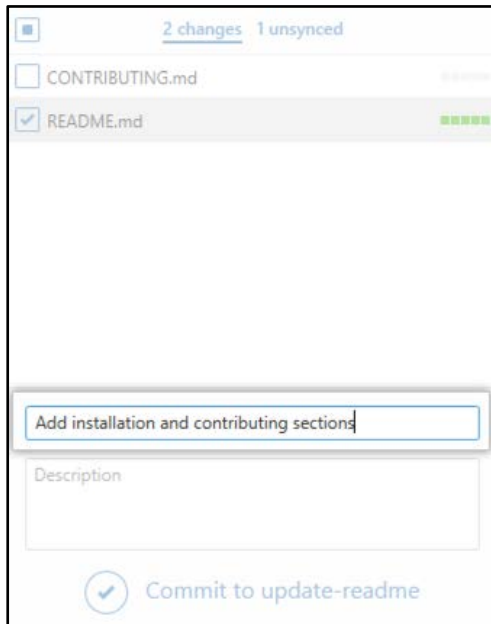


Anda juga boleh memilih fail-fail tertentu sahaja jika anda mahu untuk dimasukkan dalam proses *commit*.

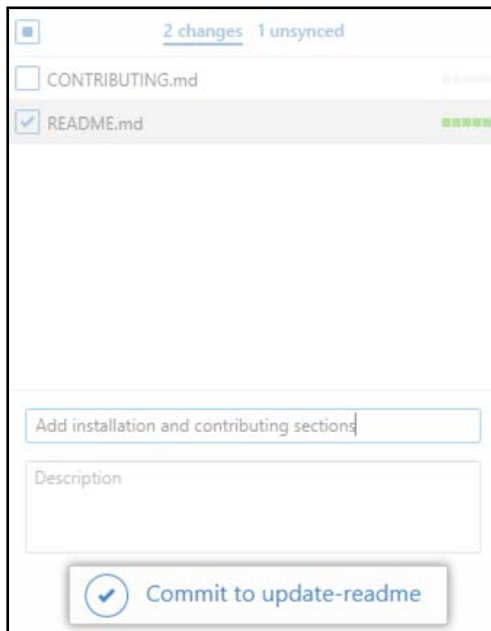


3. Tulis mesej/komen bagi *commit* anda dan lakukan *commit*

Setelah anda berpuas hati dengan semua perubahan yang dilakukan, dan telah memilih fail-fail yang terlibat untuk melakukan *commit*, anda boleh menulis mesej atau komen berkenaan dengan *commit* yang dilakukan.



Dibahagian bawah, terdapat ruangan untuk mengisi mesej/komen bagi *commit* yang akan dilakukan.



Tekan **“Commit to <nama_branch>”**

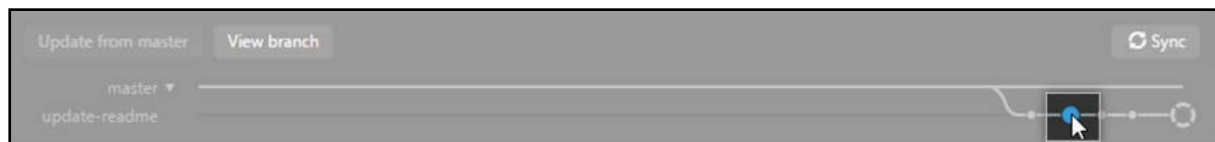
Kemudian, klik butang **“Sync”** untuk *update* dan kemaskini repositori anda di GitHub.com



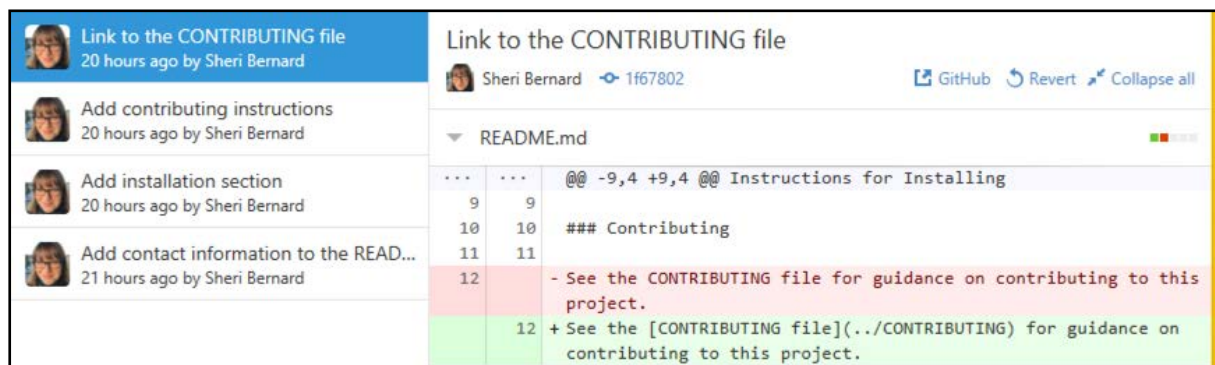
Melakukan perubahan pada *branch* proyek

Pembatalan *commit*

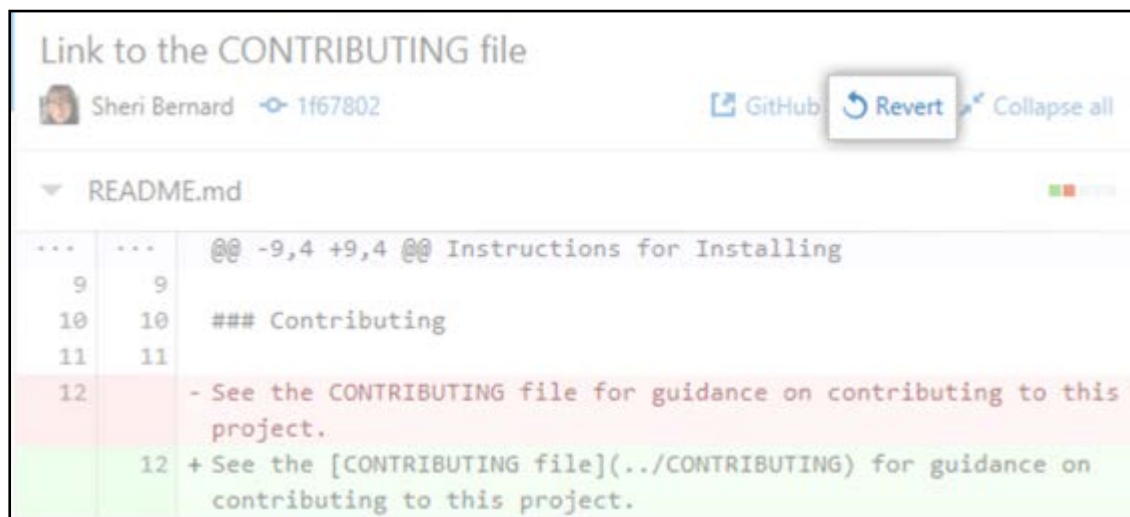
Sekiranya anda ingin membuat pembatalan kepada *commit* yang telah dilakukan, anda sentiasa boleh “*revert*” atau membuat pembatalan.



Di bahagian “*comparison graph*”, klik pada “*commit*” yang telah dilakukan



Anda boleh melihat apa sahaja yang telah dilakukan di dalam *commit* yang dilakukan.



Untuk membuat pembatalan, tekan “*Revert*”.






Melihat sejarah projek anda

Melihat mengenai perbezaan graf

Anda boleh melihat keseluruhan perubahan kepada projek yang anda lakukan. Termasuk gabungan commit "***merge commit***".

Dibahagian "*comparison graph*" terdapat beberapa simbol yang anda perlu tahu. Antaranya adalah seperti berikut :-



Ikun	Keterangan
	Ini bermaksud, <i>commit</i> anda telah dikemaskini di <i>remote</i> GitHub.com. Biasanya ia dipanggil " <i>origin</i> "
	Ini adalah gabungan <i>commit</i> " <i>merge commit</i> ". Bermaksud, repositori utama (<i>upstream</i>) telah digabungkan dengan branch semasa.
	Ini bermaksud <i>commit</i> masih belum dikemaskini di <i>remote</i> GitHub.com.

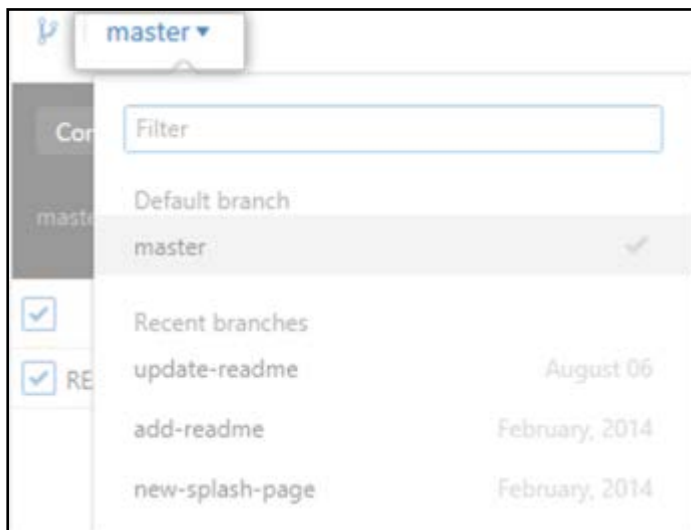


Bekerja dengan repositori *remote* anda di GitHub

Mengemaskini *branch* anda

Setelah anda membuat perubahan kepada fail-fail projek anda, anda boleh sentiasa mengemaskini semua perubahan tersebut di repositori *remote* anda di GitHub.com

Apabila anda melakukan sebarang *commit*, hanya hanya tersimpan di komputer *local* anda sahaja dan bukan di repositori *remote*. Oleh itu, anda perlu mengemaskini repositori *remote* anda untuk sentiasa *up-to-date*.



Pilih branch “**master**”



Tekan “**Sync**”



Bekerja dengan repositori *remote* anda di GitHub

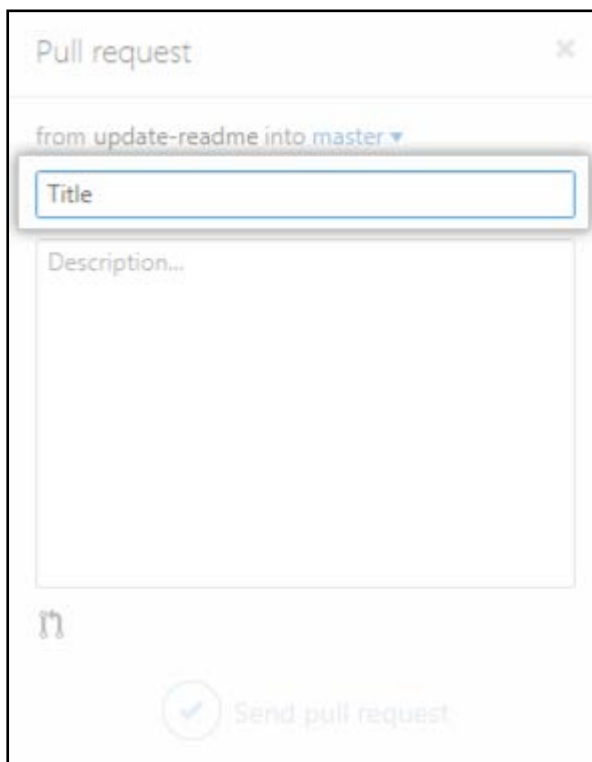
Menghantar *pull request*

Apabila anda selesai melakukan perubahan kepada projek anda dan telah dikemaskini di GitHub.com, anda boleh menghantar “*pull request*” kepada **pemunya projek** atau **pengendali projek asal** bagi beserta perubahan yang anda telah lakukan untuk dikemaskini.

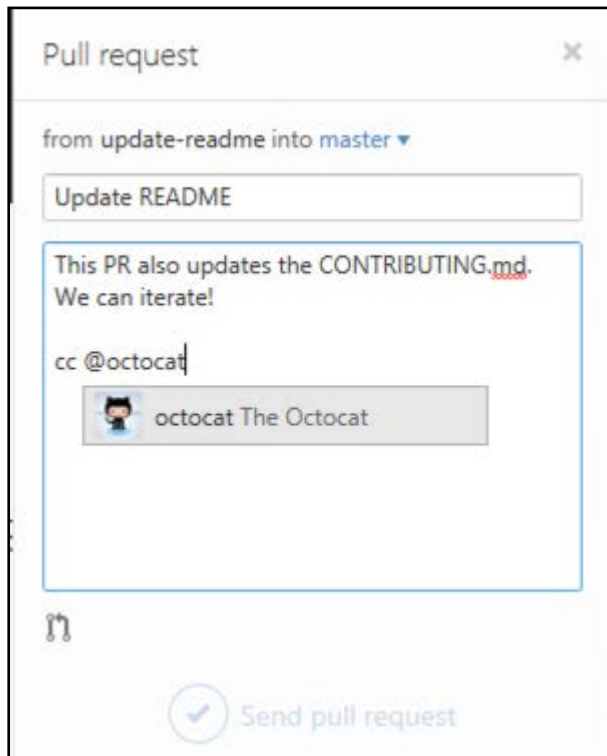
“*Pull Request*” adalah salah satu cara untuk mengemaskini projek asal supaya perubahan yang anda telah lakukan tadi akan dimasukkan ke dalam projek asal.



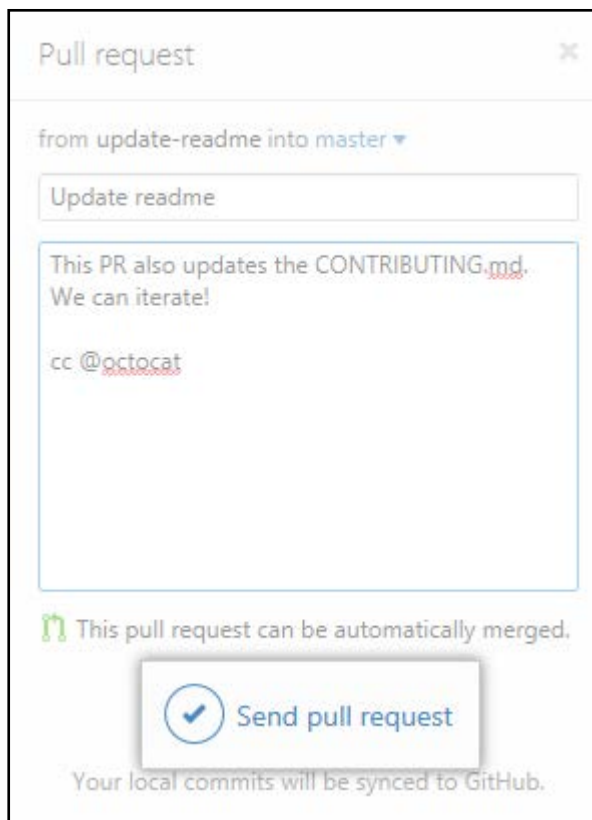
Klik pada “*Pull request*”



Masukkan tajuk dan keterangan bagi perubahan yang anda lakukan.



Contoh ayat bagi *pull request*



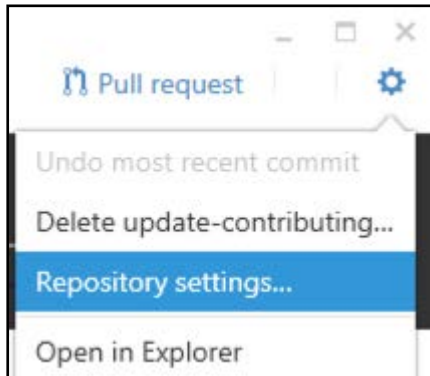
Tekan “*Send pull request*”



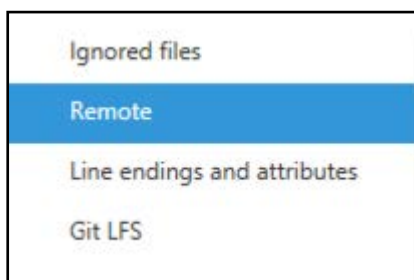
Bekerja dengan repositori *remote* anda di GitHub

Menukar URL remote melalui GitHub Desktop

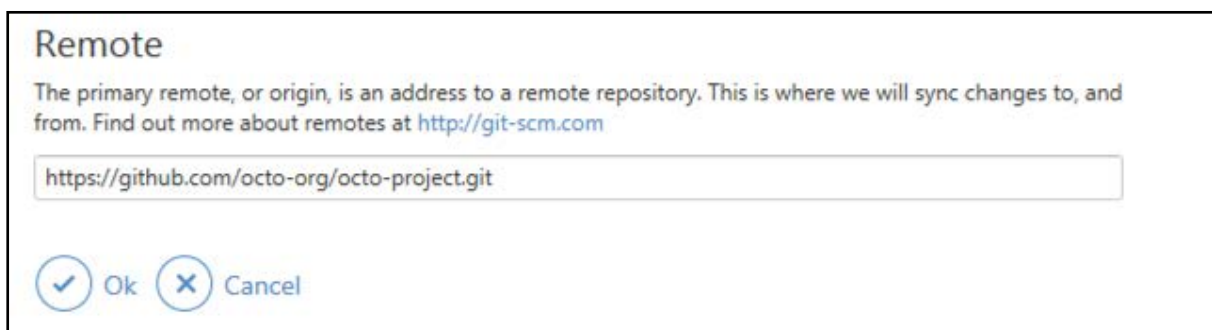
Anda boleh mengubah *remote URL* untuk repositori anda. Anda hanya perlu lakukan ini sekiranya anda **rename** repositori anda, **username** atau **organisasi**.



Klik ikon “**Gear**” dan pilih “**Repository settings...**”



Pilih “**Remote**”



Tukarkan **URL remote** dan tekan “**Ok**”



References dan Sumber Rujukan

- | | |
|------------------------------|---|
| GitHub Desktop | - https://desktop.github.com |
| GitHub Desktop Documentation | - https://help.github.com/desktop |
| Git SCM | - https://git-scm.com |
| GitHub Help | - https://help.github.com |