**MODUL**

**PRAKTIKUM KUALITAS PERANGKAT LUNAK**

**MODUL KATALON**



Oleh

TEAM TI 4B

|  |  |  |
| --- | --- | --- |
| Dosen Pembimbing | : | Musta’inul Abdi, SST., M.Kom. |

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER**

**TAHUN 2022**

# KATA PENGANTAR

Puji serta syukur penulis ucapkan kepada Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan hasil Magang Industri ini dilakukan dalam rangka memenuhi salah satu syarat untuk mendapatkan nilai Mata Kuliah Praktikum Kualitas Perangkat Lunak pada semester VII (Tujuh) Jurusan Teknologi Informasi dan Komputer Program Studi D-IV Teknik Informatika.

Katalon adalah software testing yang digunakan untuk menguji kualitas dan fungsi dari aplikasi yang telah diproduksi. Diluncurkan pada September 2016, Katalon sukses menembus hingga 9% penetrasi pasar untuk UI (user interface) test automation hanya dalam jangka waktu dua tahun.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Bapak Musta’inul Abdi, SST., M.Kom. selaku pembimbing Mata Kuliah Praktikum Kualitas Perangkat Lunak.
2. Bapak Muhammad Arhami, S.Si, M.Kom selaku Ketua Jurusan Teknologi Informasi dan Komputer.
3. Bapak Salahuddin, ST, M.Cs selaku Ketua Program Studi D-IV Teknik Informatika.
4. Bapak Ir. Rizal Syahyadi, ST., M.Eng. Sc selaku Direktur Politeknik Negeri Lhokseumawe.

Penulis menyadari masih banyak kekurangan dan kelemahan dalam pelaksanaan dan penyusunan Modul ini. Namun, penulis berharap semoga Modul ini dapat bermanfaat bagi pembaca. Dengan demikian, segala kritik dan saran yang membangun dari para pembaca akan penulis terima sehingga dapat menjadi sebuah pelajaran agar dapat membuat dengan lebih baik lagi.

Lhokseumawe, 2022

Penulis

TEAM TI 4B

# DAFTAR ISI

HAL

[KATA PENGANTAR i](#_Toc117676247)

[DAFTAR ISI iii](#_Toc117676248)

[DAFTAR GAMBAR iv](#_Toc117676249)

[DAFTAR TABEL v](#_Toc117676250)

[BAB 1 1](#_Toc117676251)

[1.1 Tujuan 1](#_Toc117676252)

[1.2 Dasar Teori 1](#_Toc117676253)

[1.3 Percobaan 1](#_Toc117676254)

[BAB 2 2](#_Toc117676255)

[2.1 Tujuan 2](#_Toc117676256)

[2.2 Dasar Teori 2](#_Toc117676257)

[2.3 Percobaan 2](#_Toc117676258)

[BAB 3 3](#_Toc117676259)

[3.1. Kesimpulan 3](#_Toc117676260)

[3.2. Dasar Teori 3](#_Toc117676261)

[3.3. Percobaan 3](#_Toc117676262)

[BAB 4 4](#_Toc117676263)

[4.1 Tujuan 4](#_Toc117676264)

[4.2 Dasar Teori 4](#_Toc117676265)

[4.3 Percobaan 4](#_Toc117676266)

[BAB 5 INTEGRATION WITH GIT AND JENKINS 5](#_Toc117676267)

[5.1 Tujuan 5](#_Toc117676268)

[5.2 Dasar Teori 5](#_Toc117676269)

[5.3 Percobaan Integrasi Dengan GIT 5](#_Toc117676270)

[5.4 Percobaan Integrasi Dengan Jenkins 14](#_Toc117676271)

[BAB 6 20](#_Toc117676272)

[6.1 Pendahuluan 20](#_Toc117676273)

[6.2 Dasar Teori 20](#_Toc117676274)

[6.3 Percobaan 20](#_Toc117676275)

# DAFTAR GAMBAR

**No table of figures entries found.**

# DAFTAR TABEL

**No table of figures entries found.**

# 

## Tujuan

## Dasar Teori

## Percobaan

# 

## Tujuan

## Dasar Teori

## Percobaan

# 

## Kesimpulan

## Dasar Teori

## Percobaan

# 

**MENJALANKAN TEST MENGGUNAKAN COMMANDLINE DAN PLUGIN JENKINS PADA KATALON STUDIO**

## Tujuan

Bagian ini akan membahas dasar – dasar bagaimana cara menjalankan test melalui commandline katalon studio.

Pada akhir pembahasan, diharapkan pembaca dapat :

1. Mengetahui kenapa menjalankan test di commandline itu penting.
2. Mempelajari cara menjalankan test melalui command line.
3. Mengetahui kelebihan menjalankan test melalui commandline.
4. Mempelajari cara menginstall Jenkins & menjalankan Jenkins.
5. Mengetahui cara membuat sebuah project di Jenkins.
6. Mengetahui cara menambahkan command pada aplikasi katalon studio.
7. Mempelajari cara menjalankan test.

## Dasar Teori

Katalon Studio merupakan salah satu software yang digunakan untuk automation testing untuk aplikasi berbasis web dan mobile. Katalon studio dapat dijalankan di semua sistem operasi yaitu windows, linux, dan mac os.

Jenkins adalah server otomatisasi open source mandiri yang dapat digunakan untuk mengotomatiskan semua jenis tugas yang terkait dengan pembuatan, pengujian, dan pengiriman atau penerapan perangkat lunak. Jenkins dapat diinstal melalui paket sistem asli, Docker, atau bahkan dijalankan secara mandiri oleh mesin apa pun dengan Java Runtime Environment (JRE) yang diinstal.

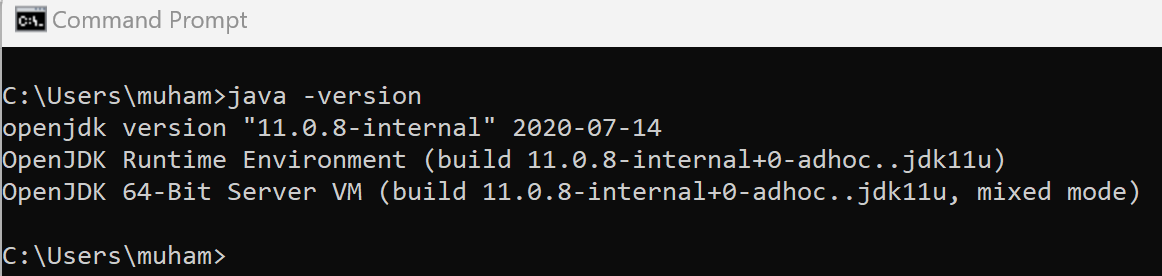
Beberapa kemungkinan langkah yang dapat dilakukan menggunakan Jenkins adalah:

1. Build software menggunakan sistem build seperti Gradle, Maven, dan lainnya.
2. Pengujian otomasi menggunakan kerangka kerja pengujian seperti Nose2, PyTest, Robot, Selenium, dan banyak lagi.
3. Jalankan skrip pengujian (menggunakan terminal Windows, shell Linux, dll.
4. Dapatkan hasil pengujian dan lakukan tindakan posting seperti mencetak laporan pengujian, dan banyak lagi.
5. Jalankan skenario pengujian terhadap kombinasi input yang berbeda untuk mendapatkan cakupan pengujian yang lebih baik.
6. Continuous Integration (CI) tempat artefak dibuat dan diuji secara otomatis. Ini membantu dalam mengidentifikasi masalah dalam produk pada tahap awal pengembangan.

Alasan mengapa menggunakan Jenkins adalah untuk membuat dan menguji produk software secara bekerlanjutan, sehingga pengembang dapat terus mengintegrasikan perubahan ke dalam build. Jenkins adalah alat CI / CD open source yang paling populer di pasaran saat ini dan digunakan untuk mendukung DevOps, bersama dengan alat native cloud lainnya.Dalam hampir semua diskusi tentang integrasi berkelanjutan sumber terbuka atau alat pengiriman berkelanjutan (CI / CD), Jenkins pasti akan diangkat.Otomatisasi (termasuk otomatisasi pengujian ) adalah salah satu praktik utama yang memungkinkan tim DevOps memberikan solusi teknologi yang "lebih cepat, lebih baik, lebih murah".Jenkins telah menjadi teknologi pendukung utama yang semakin membantu praktik DevOps mendapatkan adopsi yang luas di banyak organisasi di seluruh dunia.

## Percobaan

Syarat yang perlu diperhatikan pada tahapan utama adalah menginstall java, atau jika sudah terinstall di perangkat bisa dilihat dengan menggunakan perintah sebagai berikut:

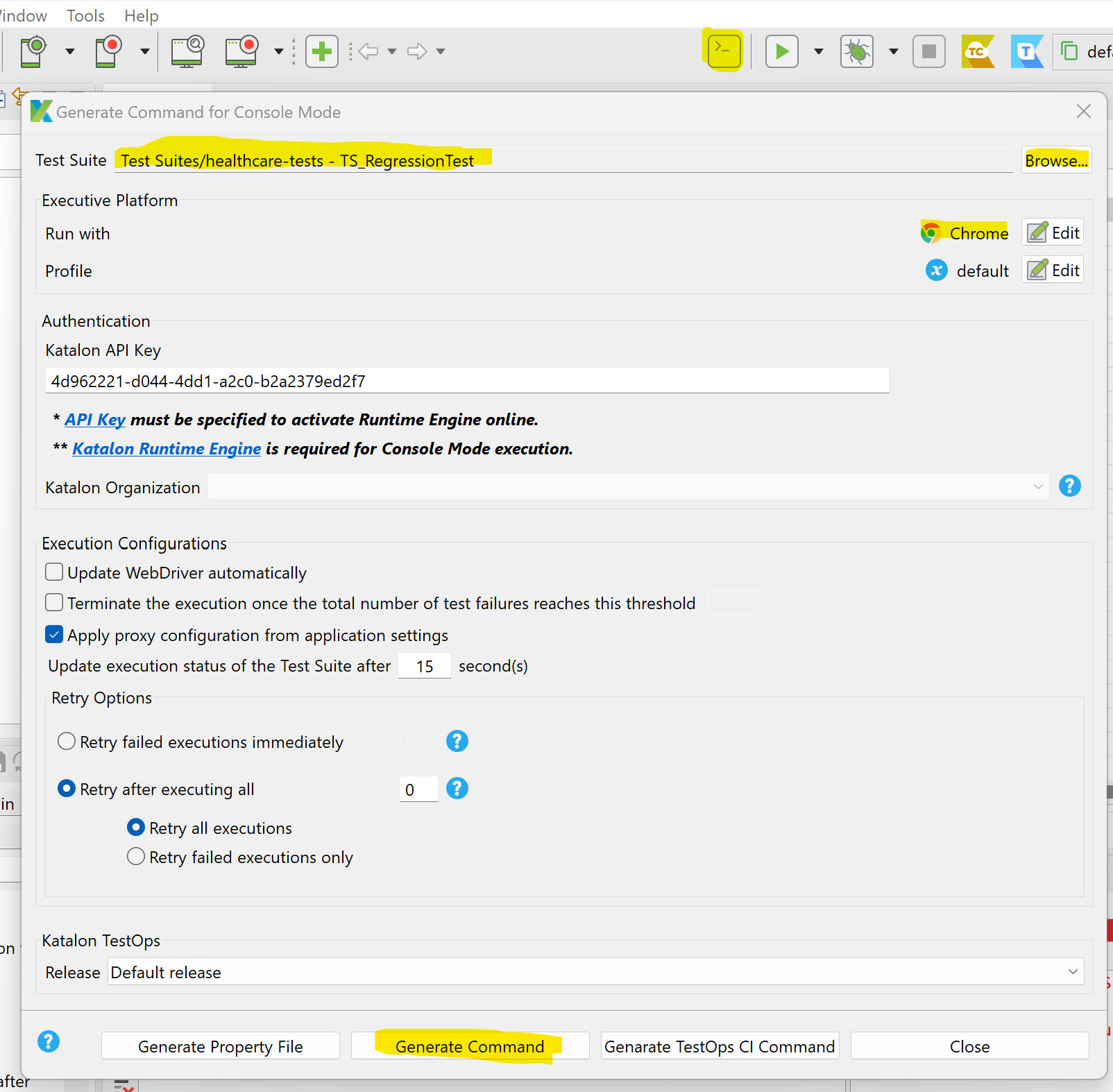


Gambar 4. 1. Melihat Versi Java Pada CMD

Dapat dilihat pada gambar diatas terdapat list java yang sudah terinstall setelah menggunakan perintah java -version pada commandline. Jika list java atau java belum terinstall di perangkat, download dan install java melalui link berikut ini: <https://www.oracle.com/java/technologies/downloads/>

Tahapan selanjutnya adalah mengikuti tatacara mengeksekusi test melalui commandline pada halaman dokumentasi katalon studio sebagai pedoman, menggunakan link berikut ini: <https://docs.katalon.com/docs/legacy/katalon-runtime-engine/command-syntax-command-lineconsole-mode-execution>

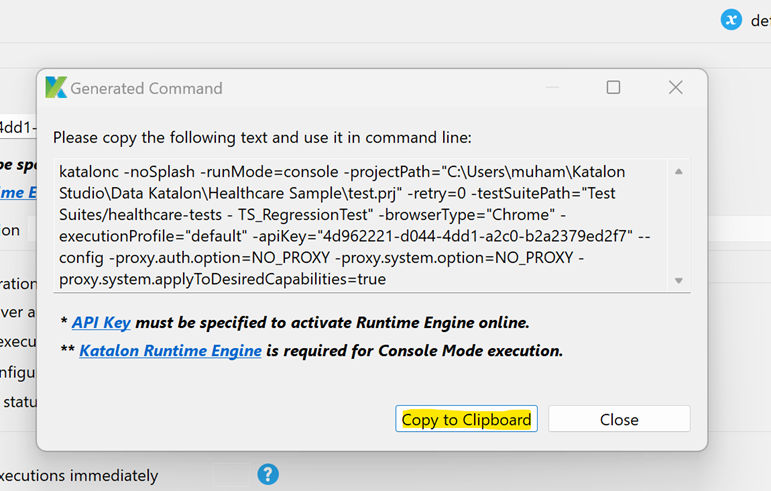
Langkah selanjutnya adalah membuka aplikasi testing catalon studio untuk mempersiapkan command dengan cara mengenerate command seperti gambar di bawah ini:



Gambar 4. 2. Mengenerate Command Pada Aplikasi Katalon

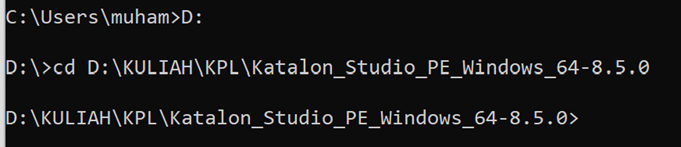
Pada gambar diatas dapat dilihat tahapan pertama untuk mengenerate command adalah dengan cara:

1. Klik pada icon terminal.
2. Kemudian muncul modal / popup menu, pilih Test suite dengan cara klik browse.
3. Selanjutnya pilih platform untuk menjalankan test, pada contoh diatas adalah dengan menggunakan google chrome.
4. Kemudian klik tombol generate command dan copy hasil generate tersebut seperti gambar di bawah ini:



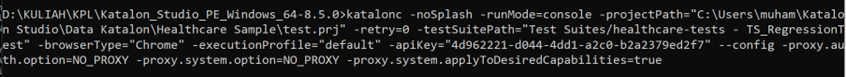
Gambar 4. 3. Copy to Clipboard Generated Command

1. Setelah mengenerate command, tahapan selanjutnya adalah buka terminal pada laptop dan arahkan path kedalam folder instalasi atau folder aplikasi katalon tersimpan, seperti gambar di bawah ini:



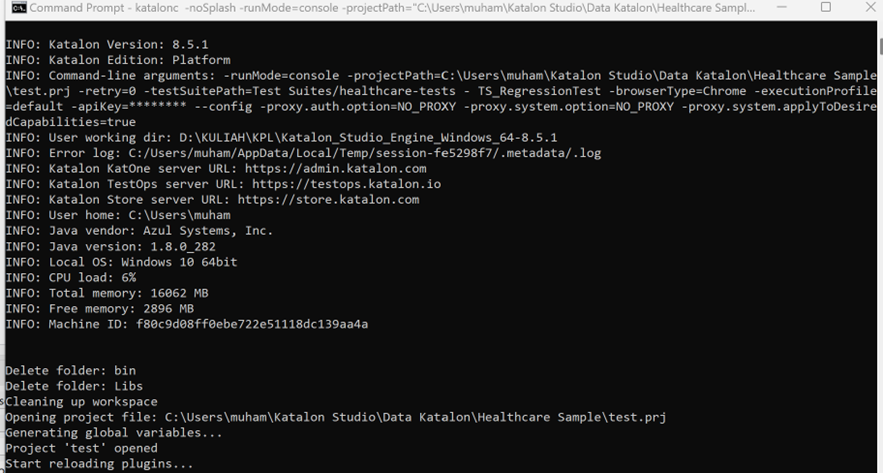
Gambar 4. 4. Path Instalasi Folder Pada CMD

1. Tahapan selanjutnya adalah copy command dan paste di dalam terminal laptop yang sudah diarahkan ke path instalasi aplikasi katalon, kemudian klik ENTER seperti gambar di bawah ini:



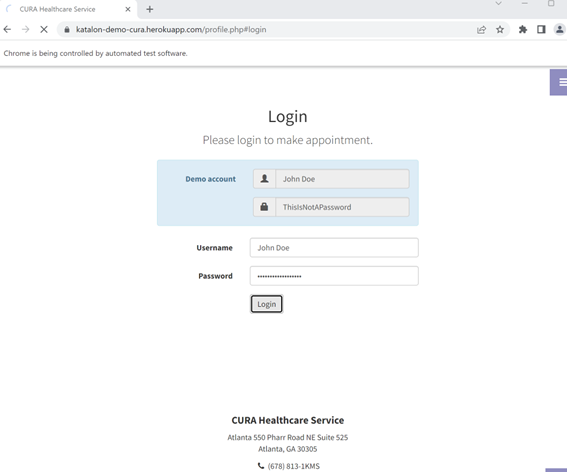
Gambar 4. 5. Menempel Perintah Generated Command

1. Kemudian akan muncul error seperti di bawah ini, penyebabnya adalah katalonc tidak bisa di jalankan di terminal windows, solusinya adalah dengan menggunakan aplikasi tambahan yaitu Katalon Runtime Engine, yang bisa di download menggunakan tautan berikut ini <https://testops.katalon.io/api/v1/katalon/download?platform=win_64&type=re>
2. Tahapan berikutnya adalah ganti path di terminal windows kearah dimana runtime engine, copy command yang sudah tergenerate dan kemudian jalankan seperti gambar di bawah ini:



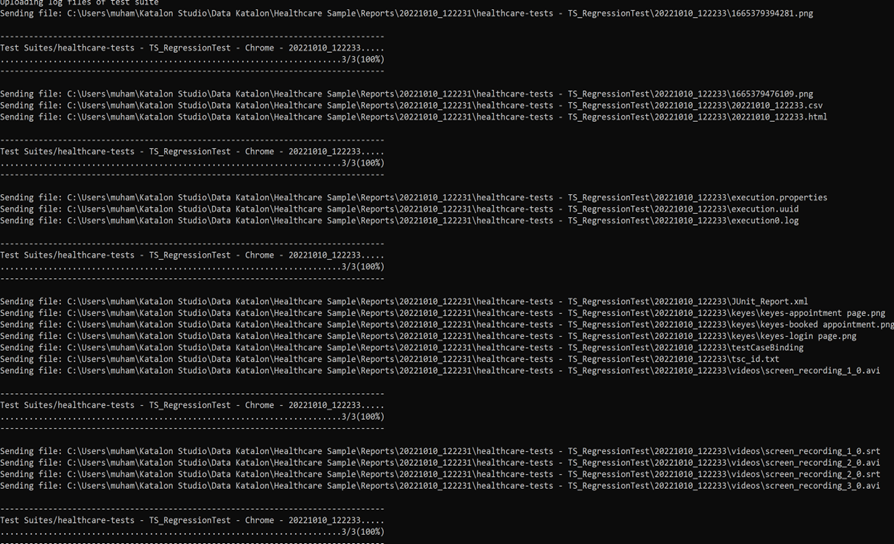
Gambar 4. 6. Mengganti Path Ke File Runtime Engine

1. Selanjutnya akan otomatis terbuka browser chrome yang menjalankan website untuk testing secara otomatis seperti gambar di bawah ini:



Gambar 4. 7. Testing Website Otomatis Pada Browser Chrome

1. Berikut adalah beberapa hasil atau output dari testing menggunakan commandline:



Gambar 4. 8. Output Testing Pada CMD

# INTEGRATION WITH GIT AND JENKINS

## Tujuan

1. Mengetahui cara menambahkan projek ke dalam GIT
2. Mengetahui cara meng-Clone Projek
3. Mengetahui cara commit, pull, dan push

## Dasar Teori

1. **GIT**

Git adalah sistem kontrol versi perangkat lunak yang gratis. Git dapat digunakan untuk menyimpan dan mengelola proyek TestComplete, git juga dapat bekerja dengan repositori lokal (terletak di mesin Anda), serta dengan repositori jarak jauh (terletak di jaringan).

1. **Integration with Git**

Integration with Gitberarti dapat mengintegrasikan TestComplete dengan Git dan bekerja dengan repositori langsung dari antarmuka pengguna TestComplete. Tanpa meninggalkan IDE TestComplete, Anda dapat menambahkan file proyek ke repositori, melakukan perubahan, membatalkan perubahan dan memasukkan ke repositori, dan lainnya. Tindakan ini dapat dilakukan untuk masing-masing item proyek dan elemen turunannya, tidak hanya untuk file project dan project suite.TestComplete menyertakan dialog bawaan khusus yang menyederhanakan pelaksanaan berbagai perintah pada file dan repositori, seperti melihat riwayat file, membuat dan menggabungkan cabang, dan sebagainya.

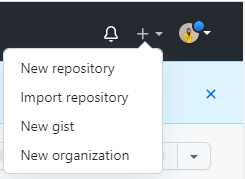
Disini kita akan membuat remote repository di github, membuat project di katalon studio kemudian meng-clone repository di katalon studio.

## Percobaan Integrasi Dengan GIT

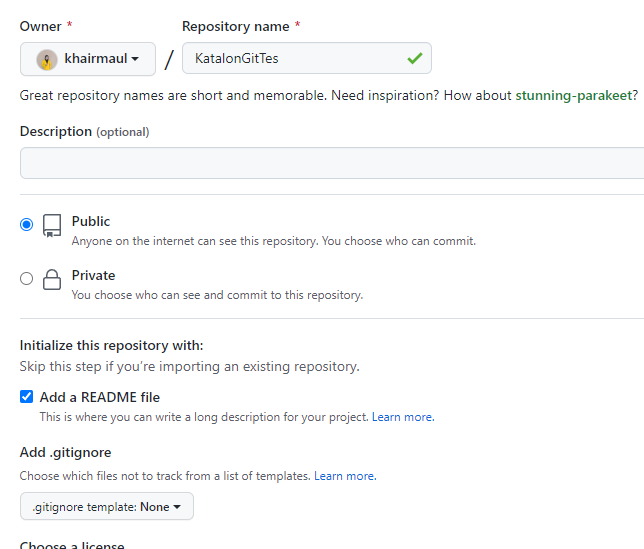
1. Pastikan di katalon studio telah terhubung GIT 
2. Buatlah sebuah repository baru di GitHub,

Caranya sebagai berikut:

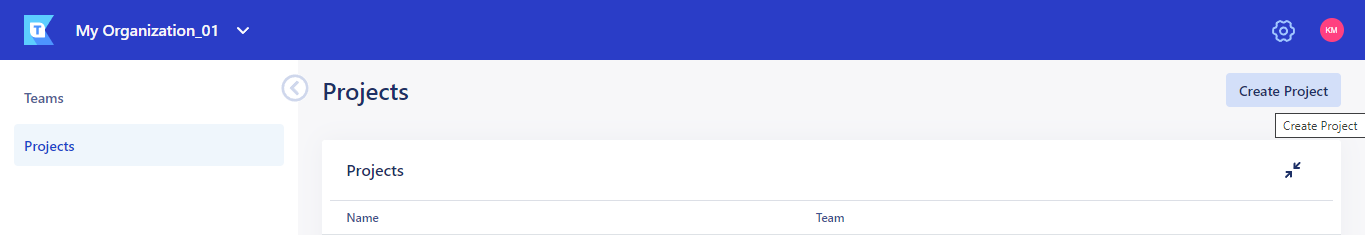
1. Buka situs github di <https://github.com>, buat akun github kemudian login dengan akun yang telah dibuat.
2. Klik tanda plus(+) pada pojok kanan atas kemudian pilih *new repository*

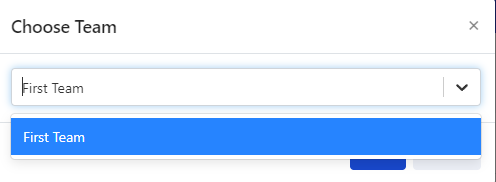


1. Buatlah nama repository yang anda inginkan, kemudian pilih public agar semua orang dapat melihat repository anda dan beri centang di add a readme file. Lalu klik create repository

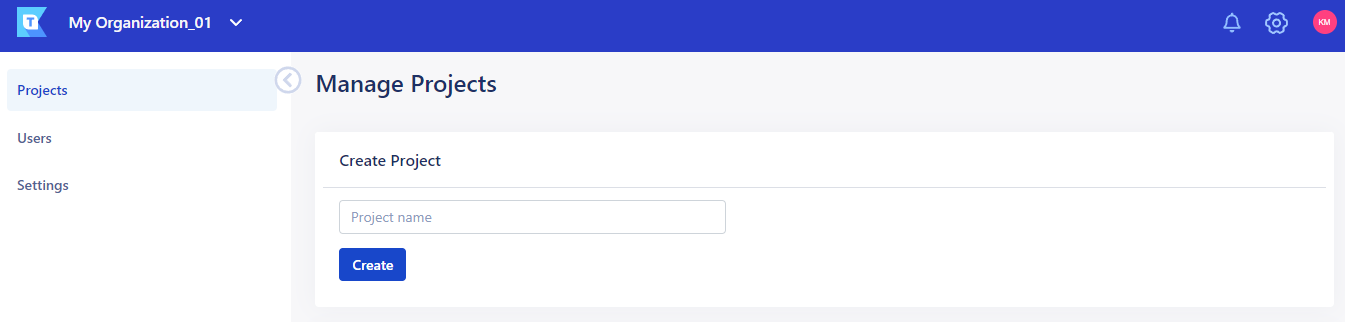


1. Buatlah sebuah projek baru di katalon studio  
   Caranya sebagai berikut:
2. Buka situs katalon di <https://katalon.com/> , daftarkan akun di web tersebut.
3. Kemudian masuk dengan akun yang telah didaftar lalu akan diarahkan pada dashboard web katalon studio.
4. Klik create project, lalu pilih team yang diinginkan atau membuat team yang baru.

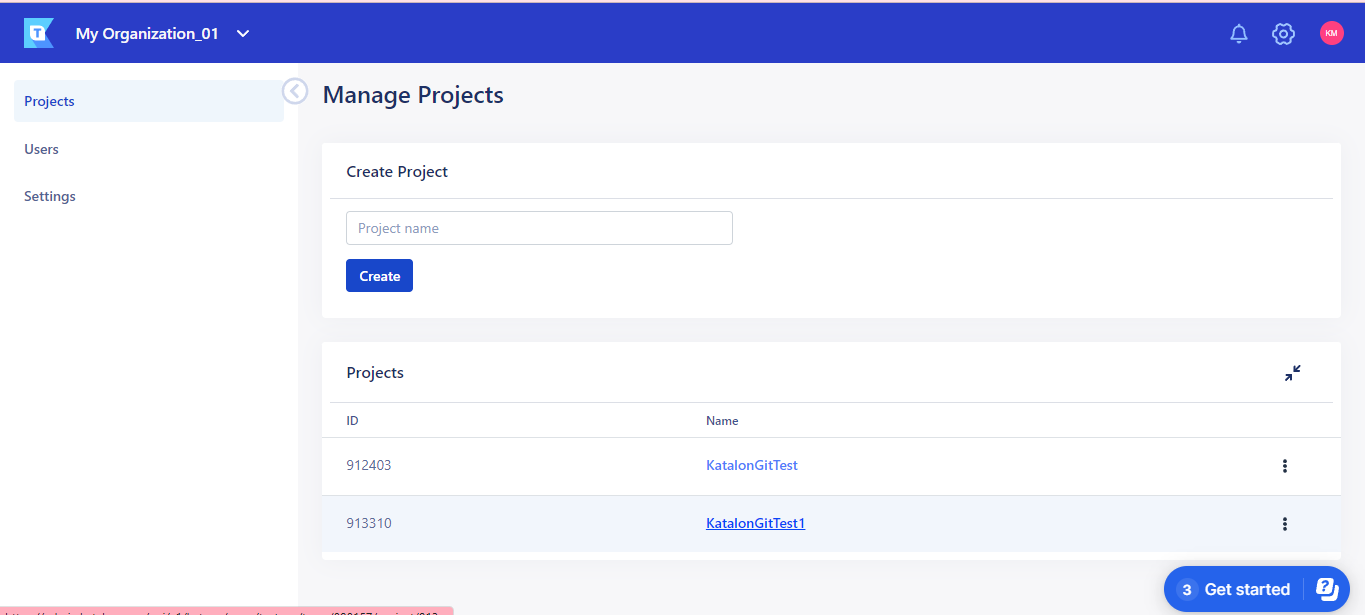




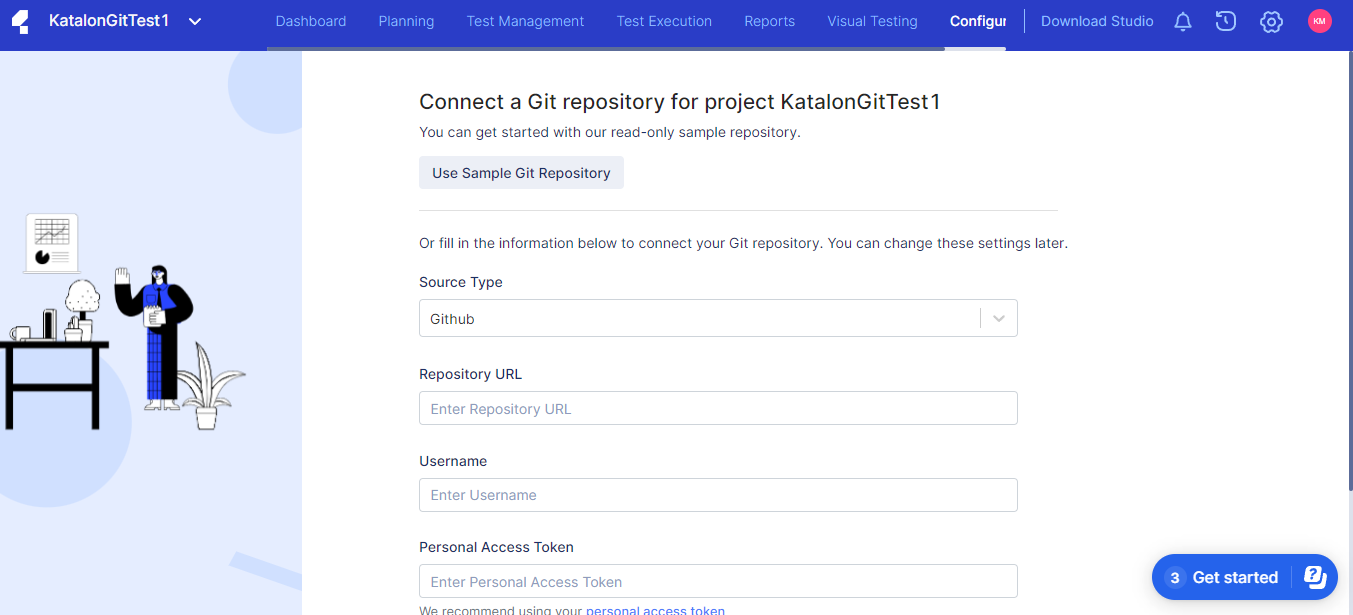
1. Masukkan nama project yang diinginkan



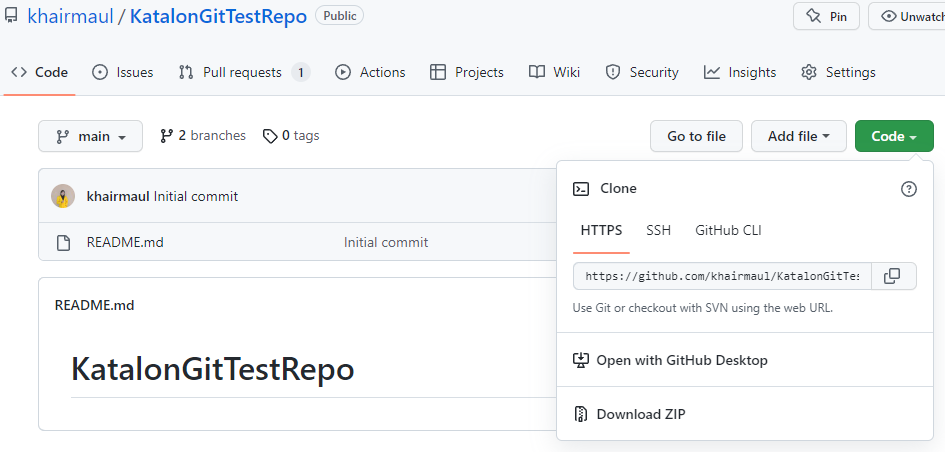
Maka projek yang telah dibuat akan tampil seperti pada gambar berikut



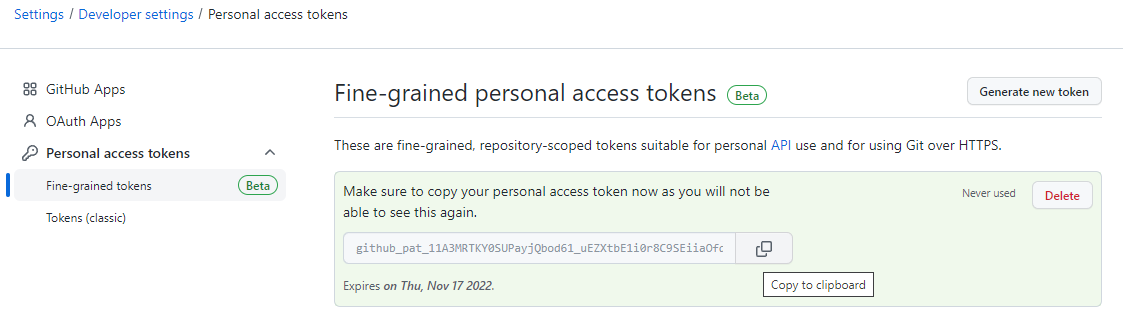
1. Klik pada nama project tersebut, kemudian akan diarahkan untuk mengkoneksi Git dengan project yang telah dibuat.



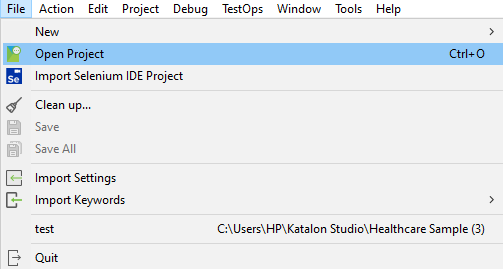
* Untuk mendapatkan repository URL, kembali ke github klik code pada repository yang telah dibuat, dan copi url clone httpsnya, tampak seperti pada gambar dibawah ini



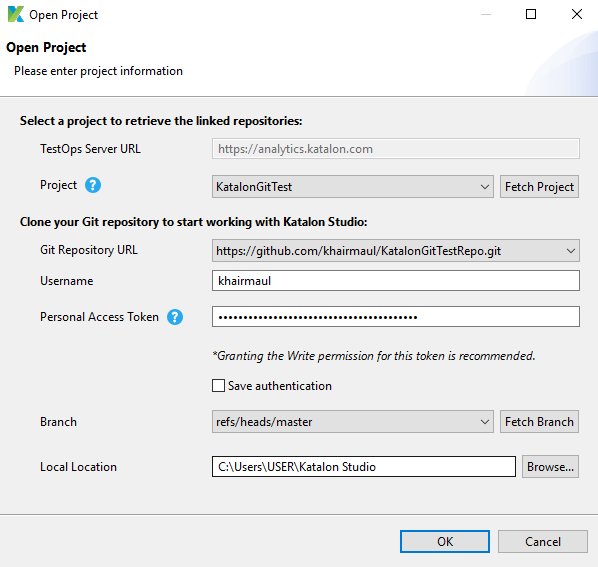
* Untuk mendapatkan personal acces token, klik segitiga yang ada pada pojok kanan atas pilih settings🡪 🡪 🡪 🡪 dimintakan untuk memasukkan password akun github🡪 masukkan nama token🡪 lalu klik generate, maka akan diperoleh token seperti pada gambar dibawah ini



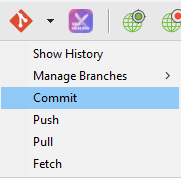
1. Open dan clone clone project



Pilih project yang ingin di open, masukkan username dan acces token dan pilih branch master, seperti dibawah ini

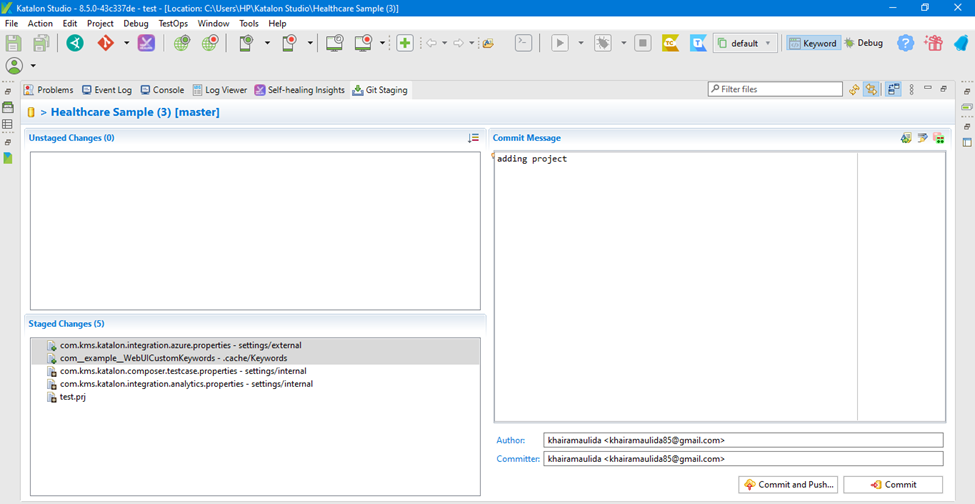


1. Commit project

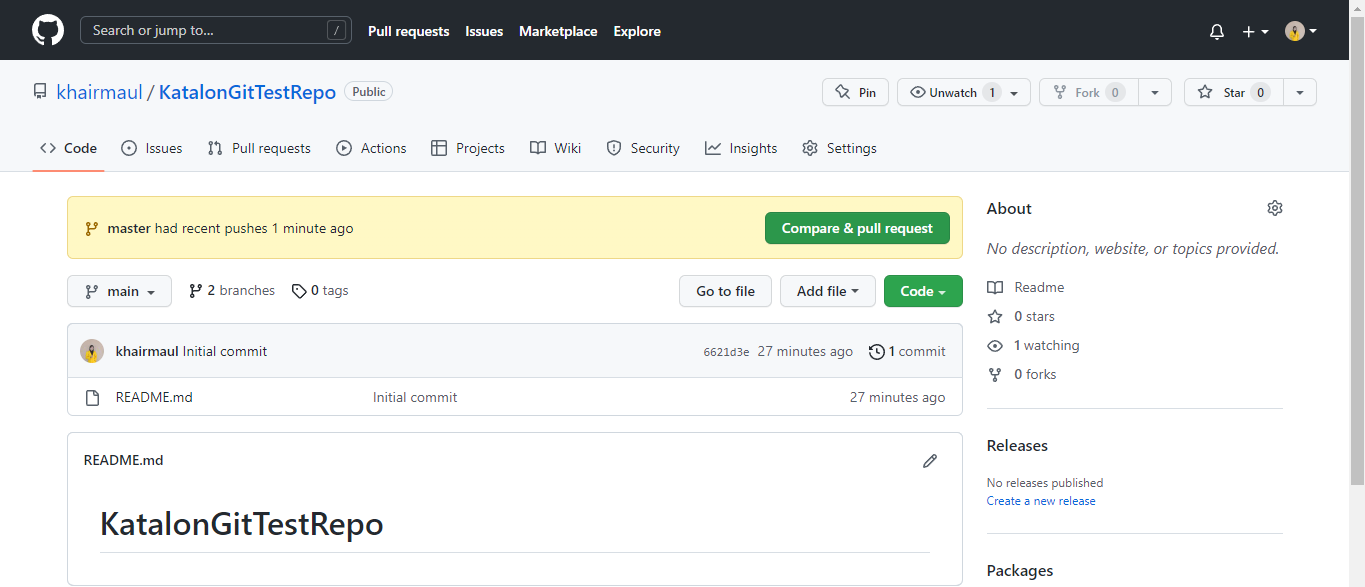


Kemudian lakukan langkah-langkah dibawah ini:

1. Pindahkan isi yang ada di unstaged changes ke dalam staged changes
2. Isikan text dalam jendela commit massage kemudian klik klik commit and push.



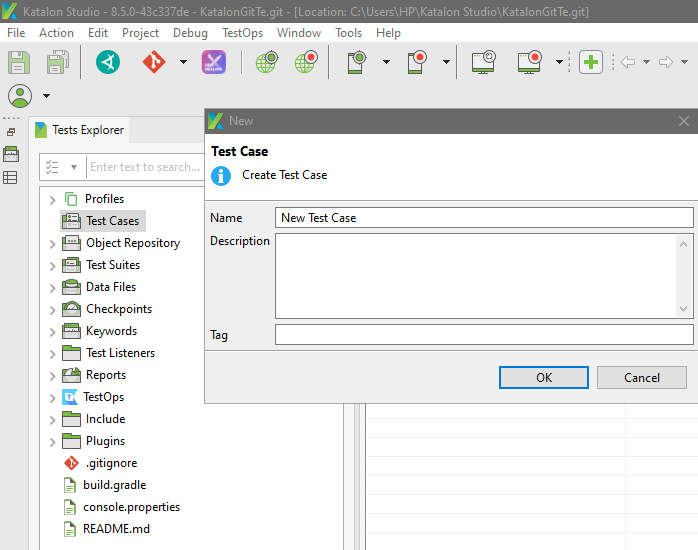
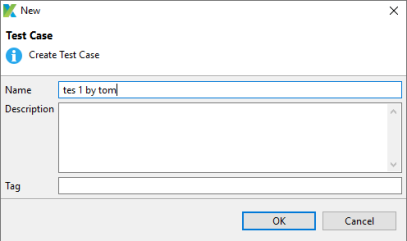
Hasilnya dapat dilihat di akun github, seperti pada gambar dibawah ini, kemudian refresh terlebih dahulu.



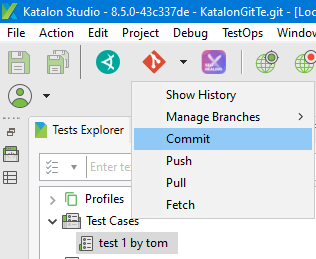
Dalam projek ini dapat dibuat perubahan apa pun dan kapan pun yang diinginkan, juga dapat melakukan tes literasi lengkap dan push di test case baru.

**Contoh 1**:

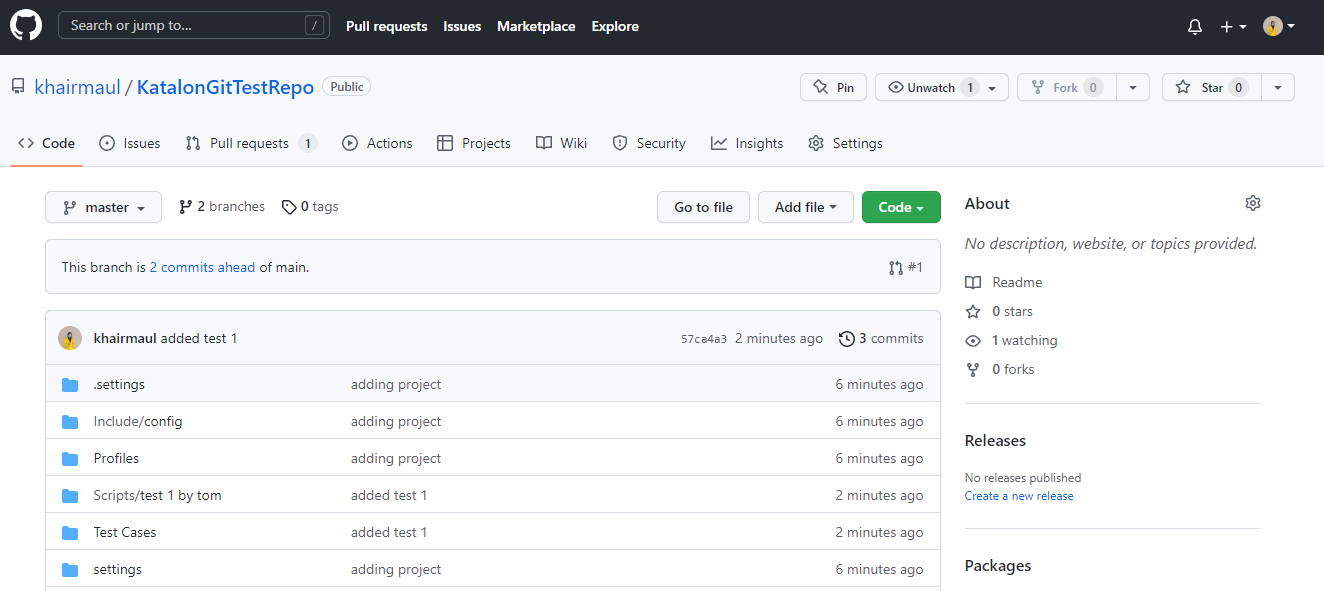
* Saya akan menambahkan test case dan memberi nama test 1 by tom  
  Dapat dilihat seperti pada gambar dibawah ini



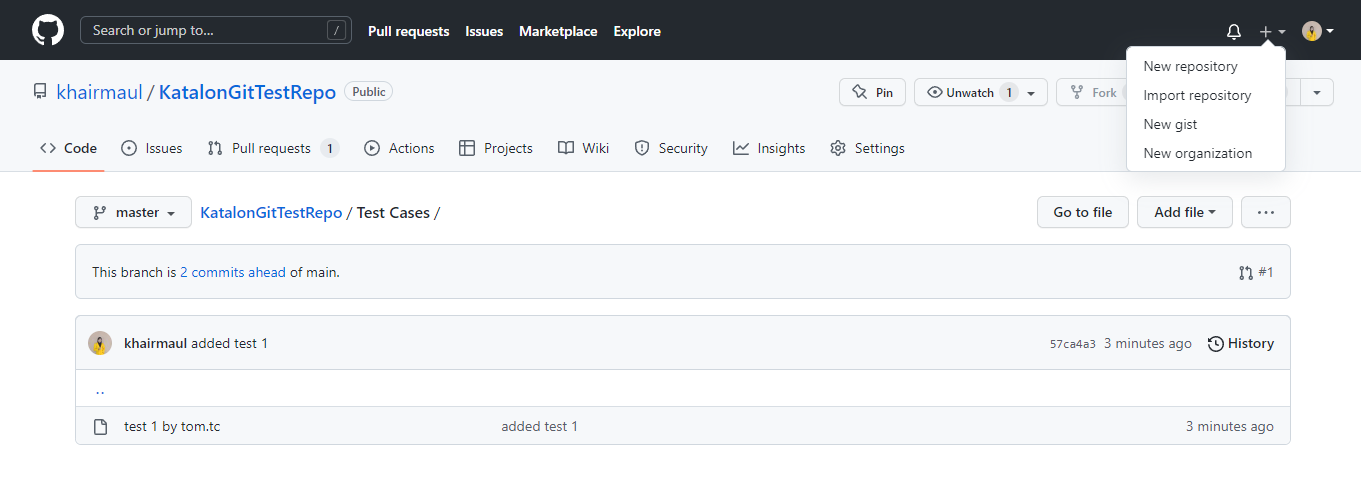
Kemudian klik icon github dan pilih commit



Ulangi langkah-langkah seperti pada saat commit project sebelumnya, dan hasilnya dapat dilihat di akun github setelah direfresh seperti pada gambar di bawah ini. Test case “test 1 by tom telah masuk ke dalam repo github”



Ketika folder dibuka maka akan ditampilkan file testcase tersebut.

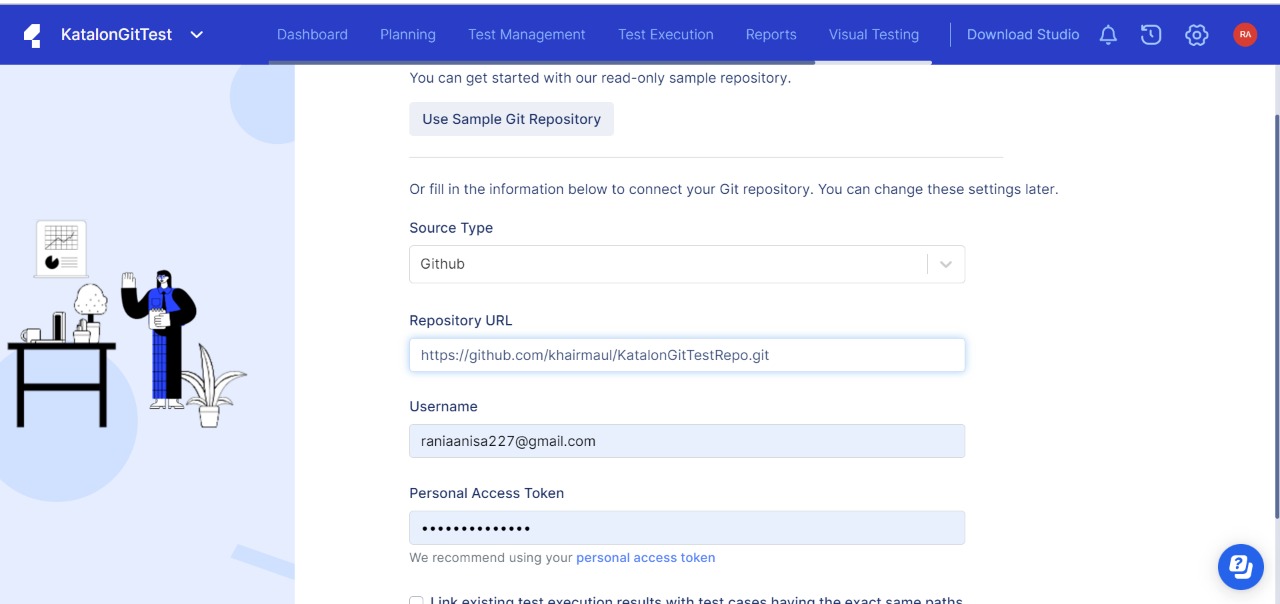


**Contoh 2**:

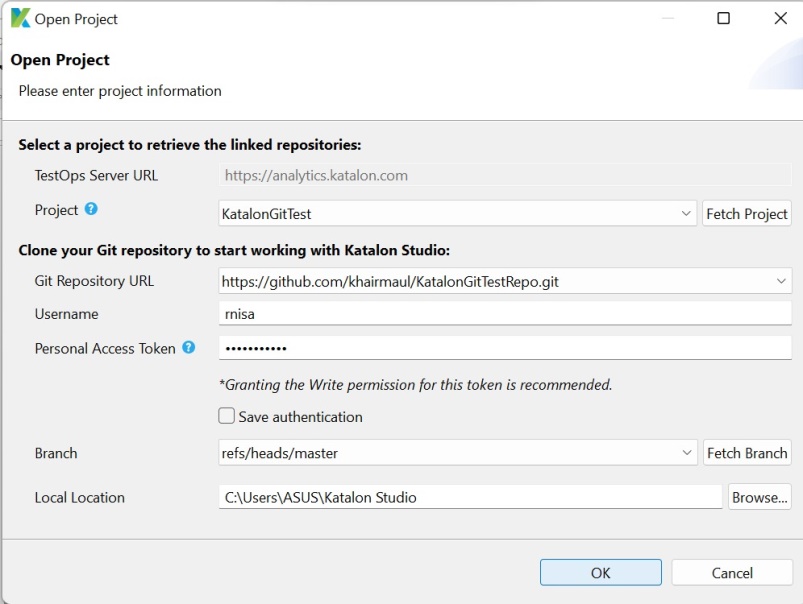
* Ran akan mengkloning proyek yang dibuat oleh tom ke dalam katalon studio miliknya.

Berikut langkah-langkahnya:

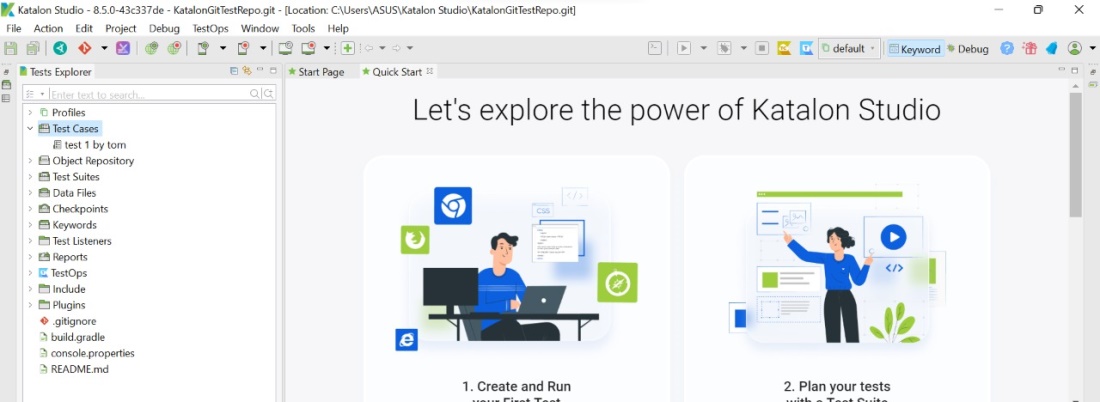
1. Buat sebuah project di katalon studio dengan nama yang sama dengan tom kemudian koneksikan dengan git.



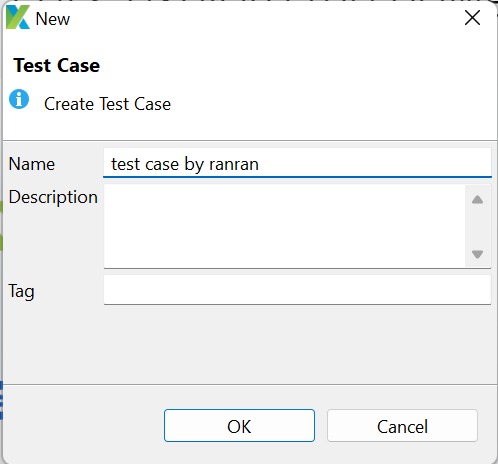
1. Open project dan clone repository tom



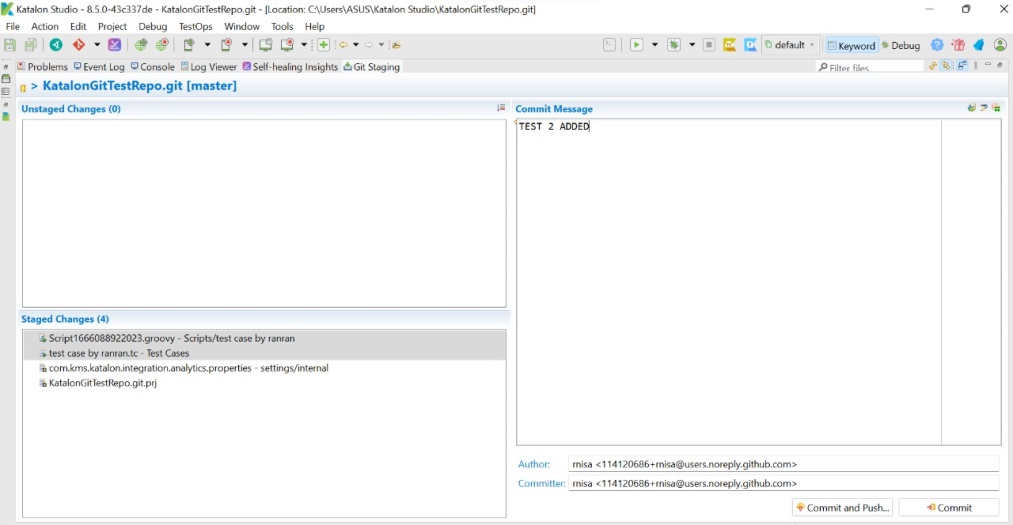
Maka akan terlihat test case yang telah dibuat oleh tom seperti pada gambar dibawah.



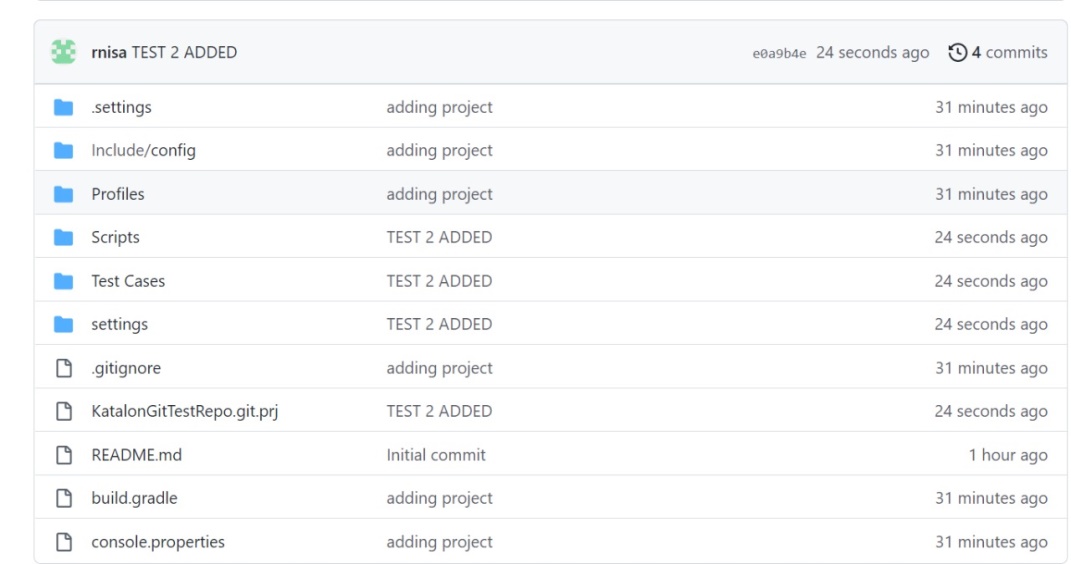
1. Kemudian ran akan membuat perubahan pada repository tom yaitu membuat testcase baru



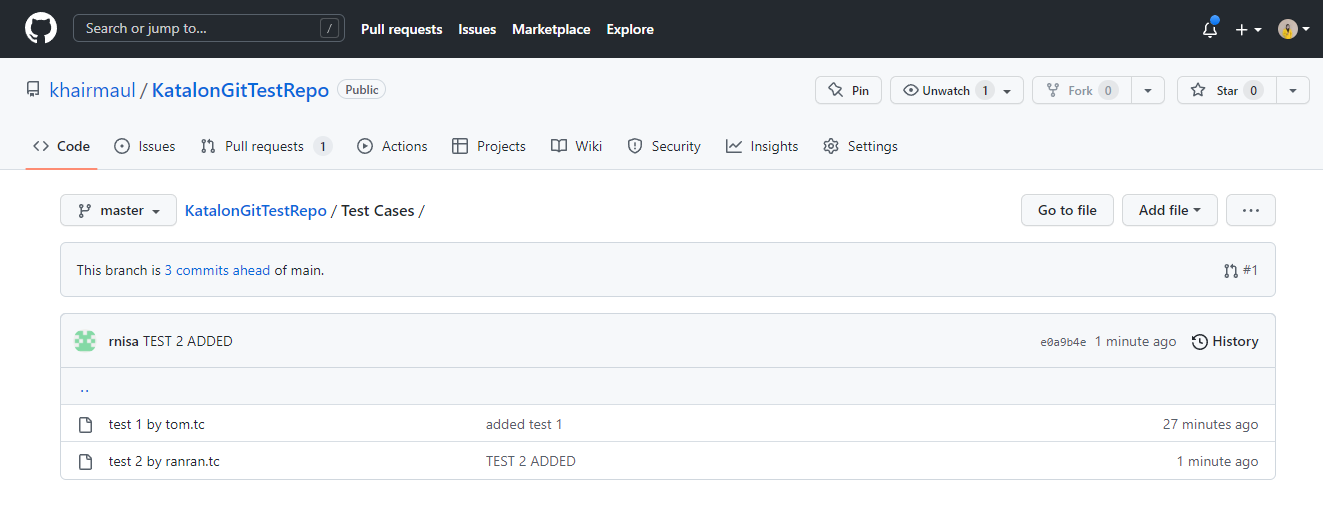
1. Kemudian meng-commit testcase yang telah dibuat. Caranya sama seperti yang telah dilakukan diatas sebelumnya.



1. Sekarang dapat dilihat pada repository github terdapat folder Test dan Cases commit yaitu MANAGE TEST 2 ADDED

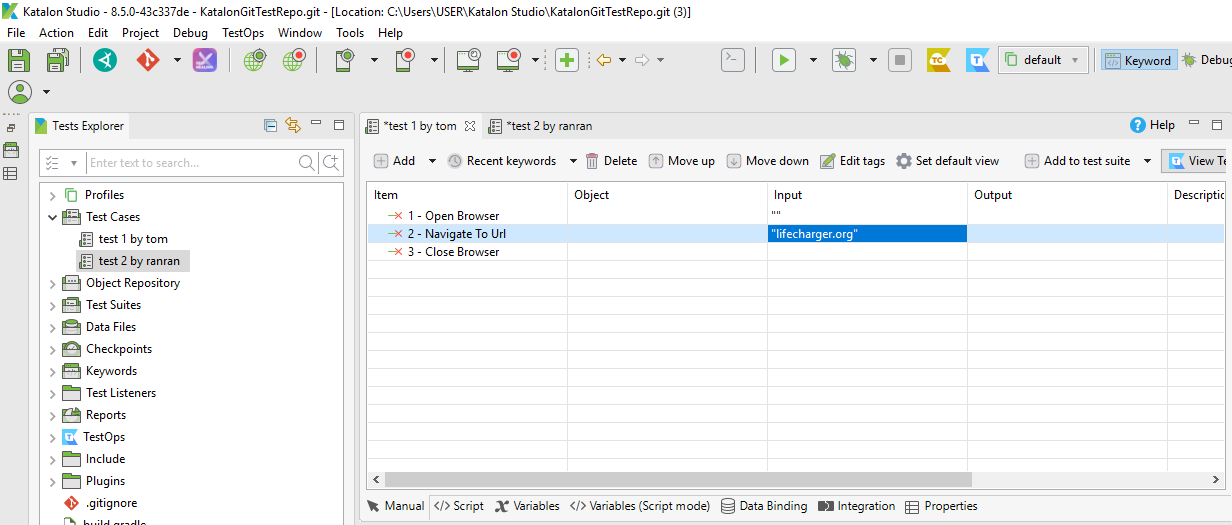


Ketika folder tersebut di klik maka akan menampilkan test case yang telah dibuat oleh ran

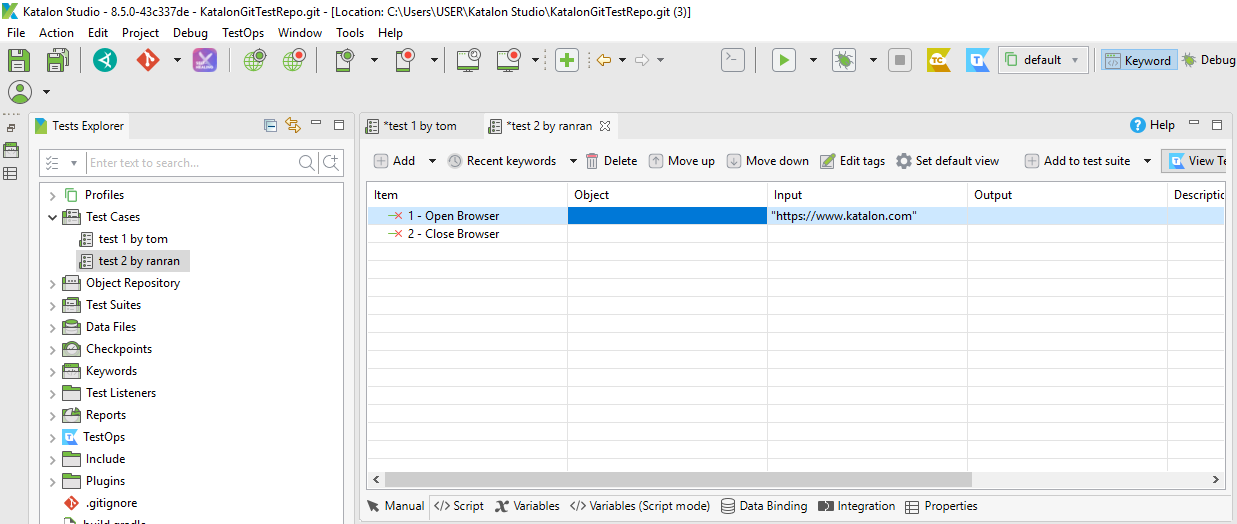


## Percobaan Integrasi Dengan Jenkins

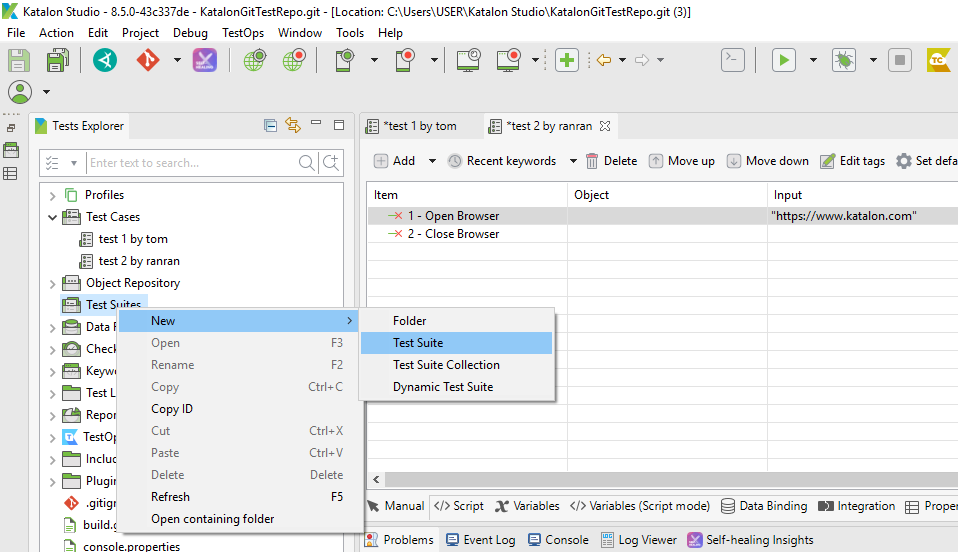
1. Buka katalon studio, pilih *test cases ‘test 1 by tom’* kemudian klik *add* dan pilih *item-item* seperti gambar dibawah.



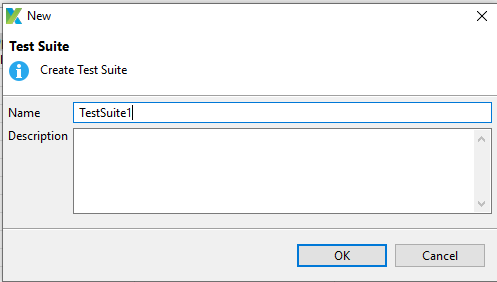
1. Kemudian pada *tes 2 by ranran* klik *Add* pilih *item-item* seperti gambar dibawah.



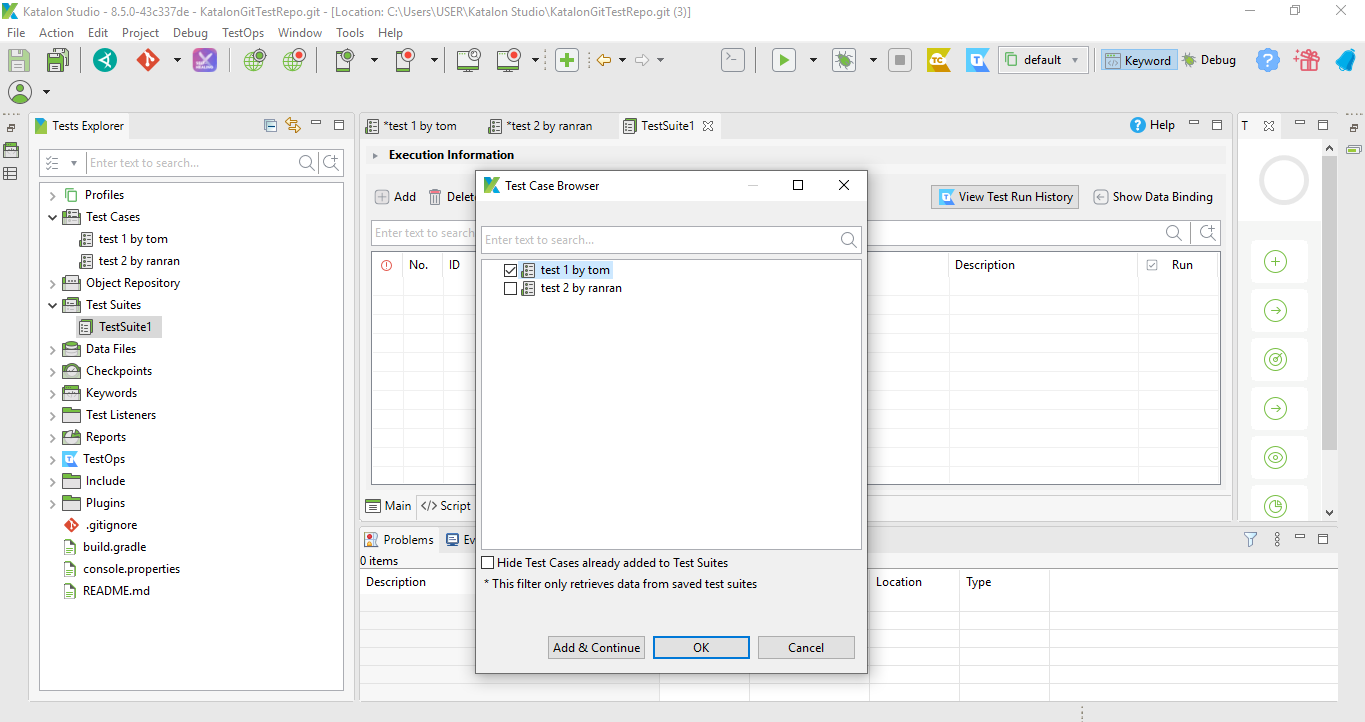
1. Klik kanan pada *test suite* pilih *New* kemudian pilih *Test Suite*



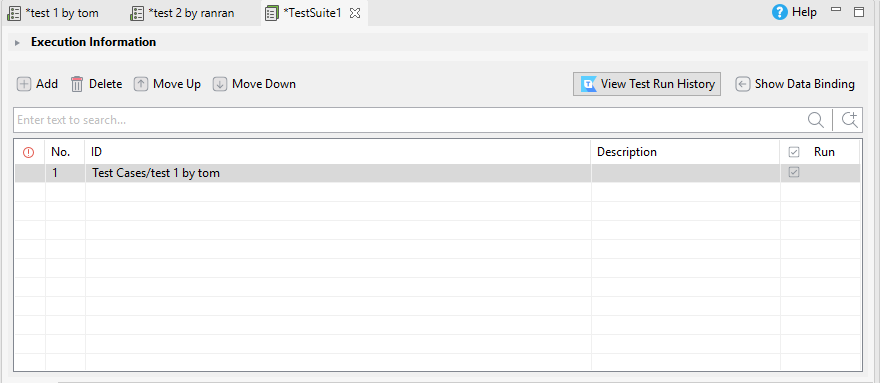
1. Membuat test suite baru dengan memasukkan nama TestSuite yang diinginkan.



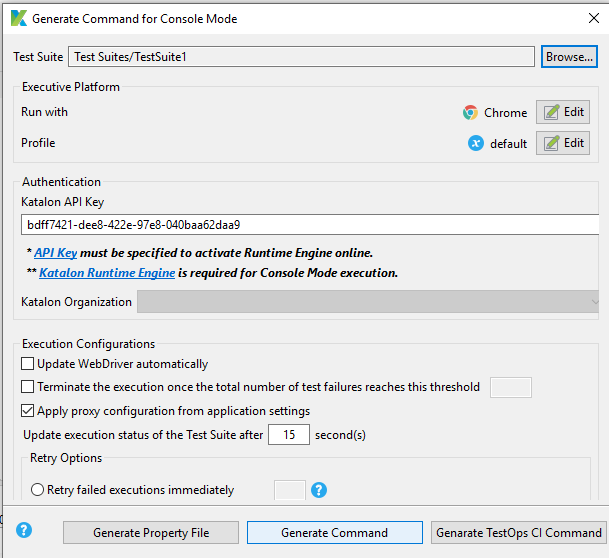
1. Kemudian klik add dan pilih test 1 by tom



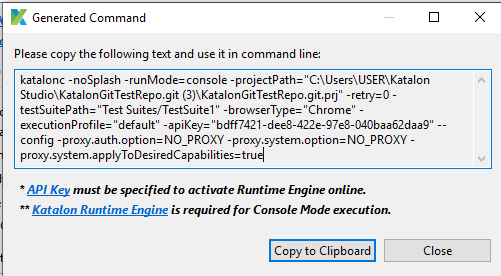
1. Maka hasilnya seperti berikut



1. Kemudian klik build CMD , pada *Browser* pilih Test suites lalu klik generate command

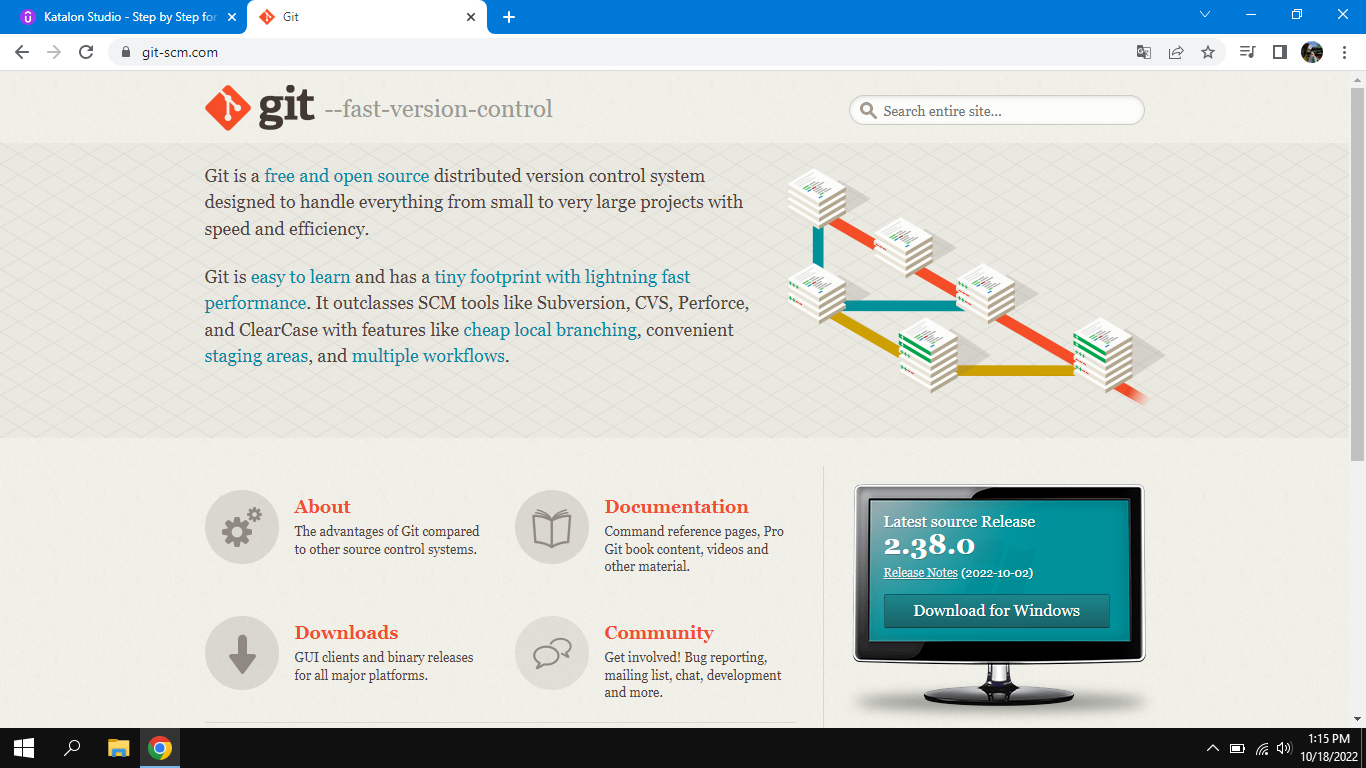


1. Lalu klik pada tombol Copy to Clipboard

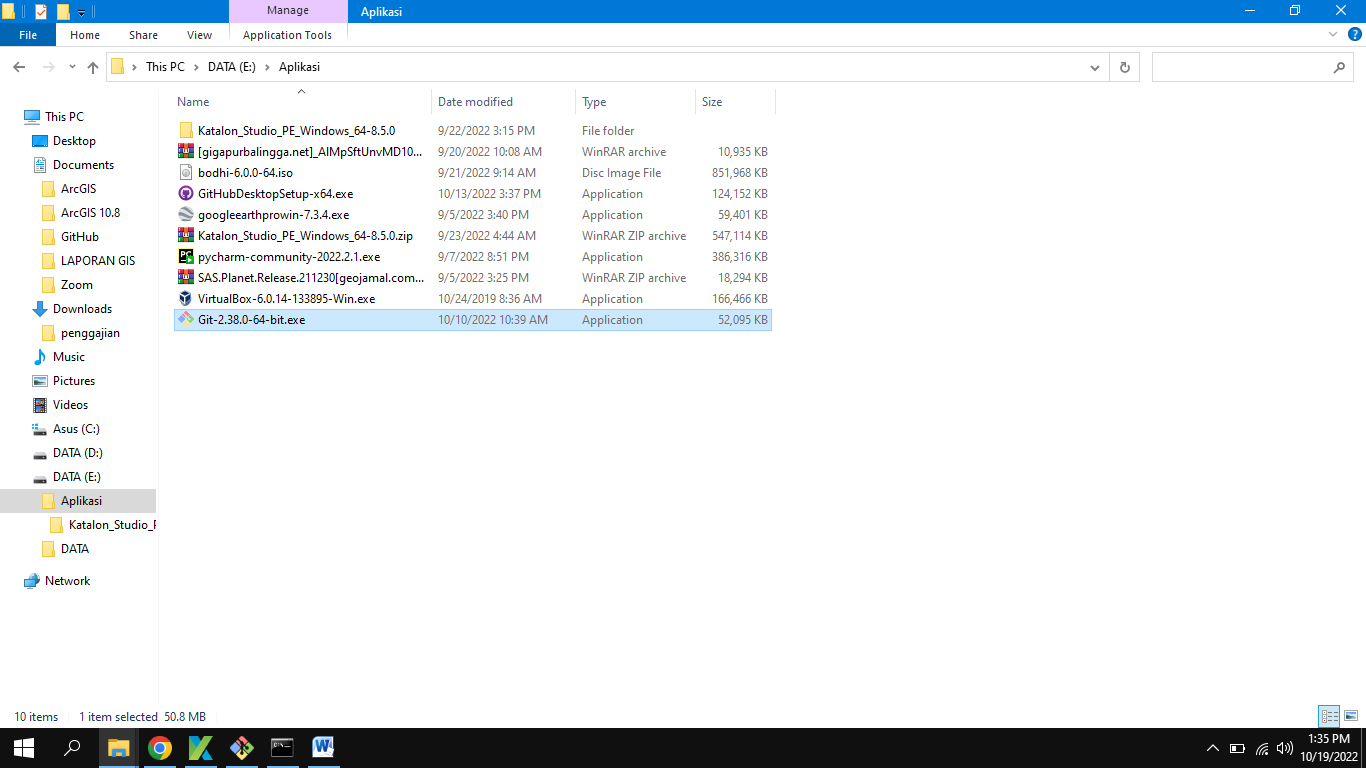


Tahap 2 Install Git

1. Buka git melalui link <https://git-scm.com/>
2. Download git sesuai operasi anda

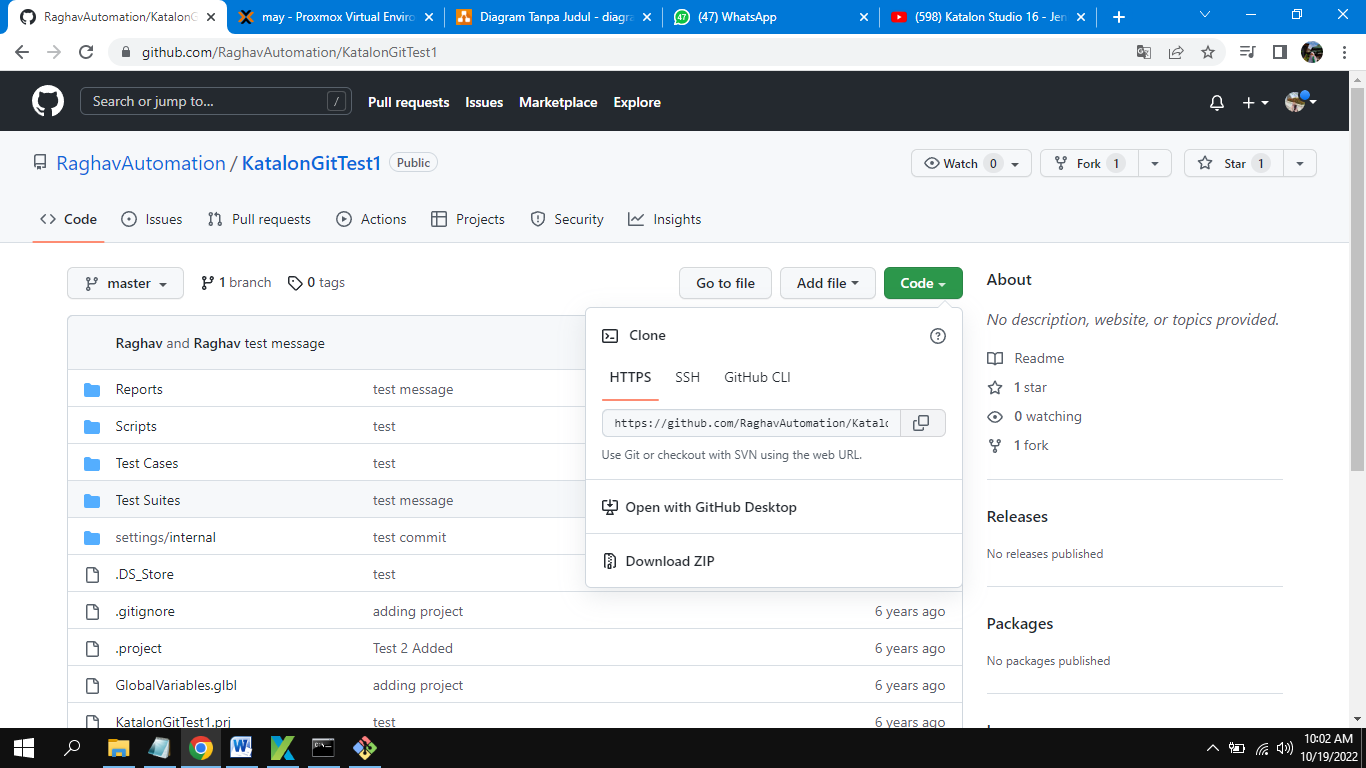


1. Kemudian install

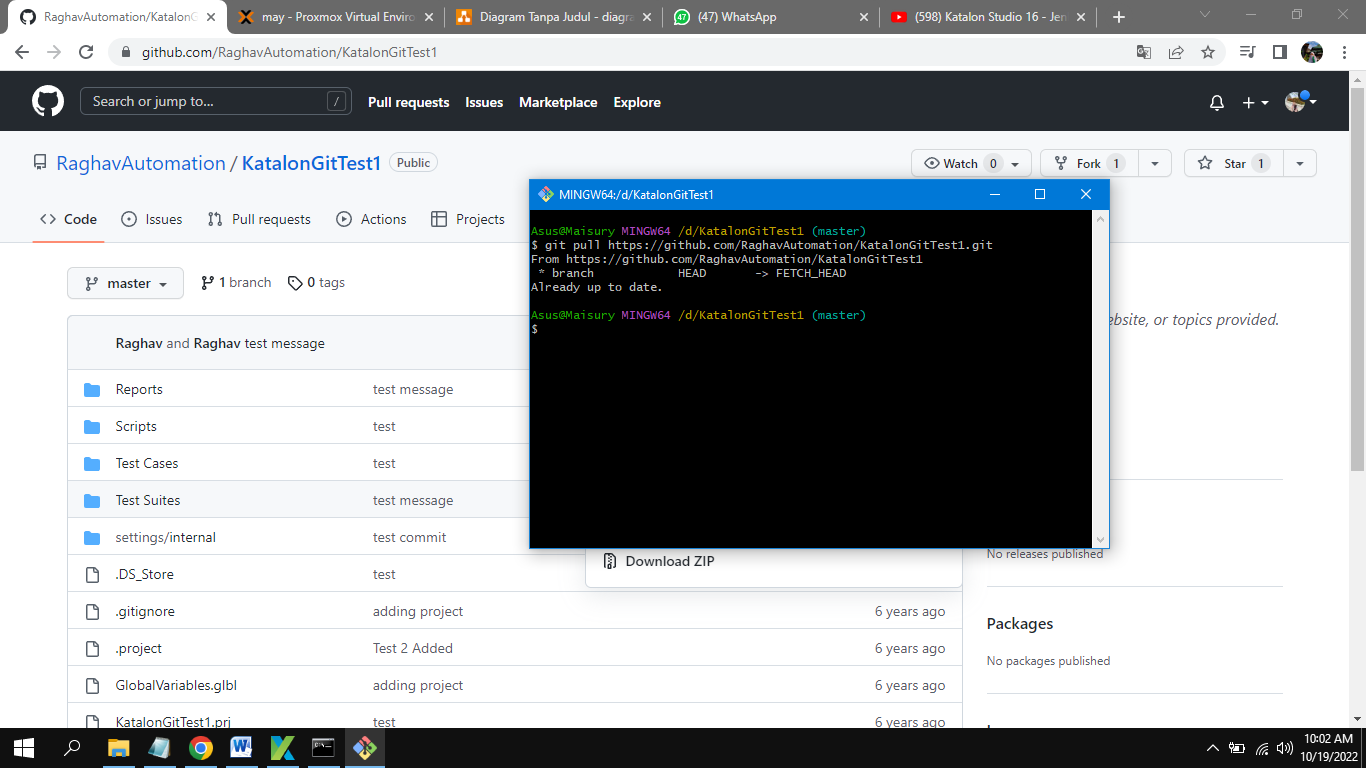


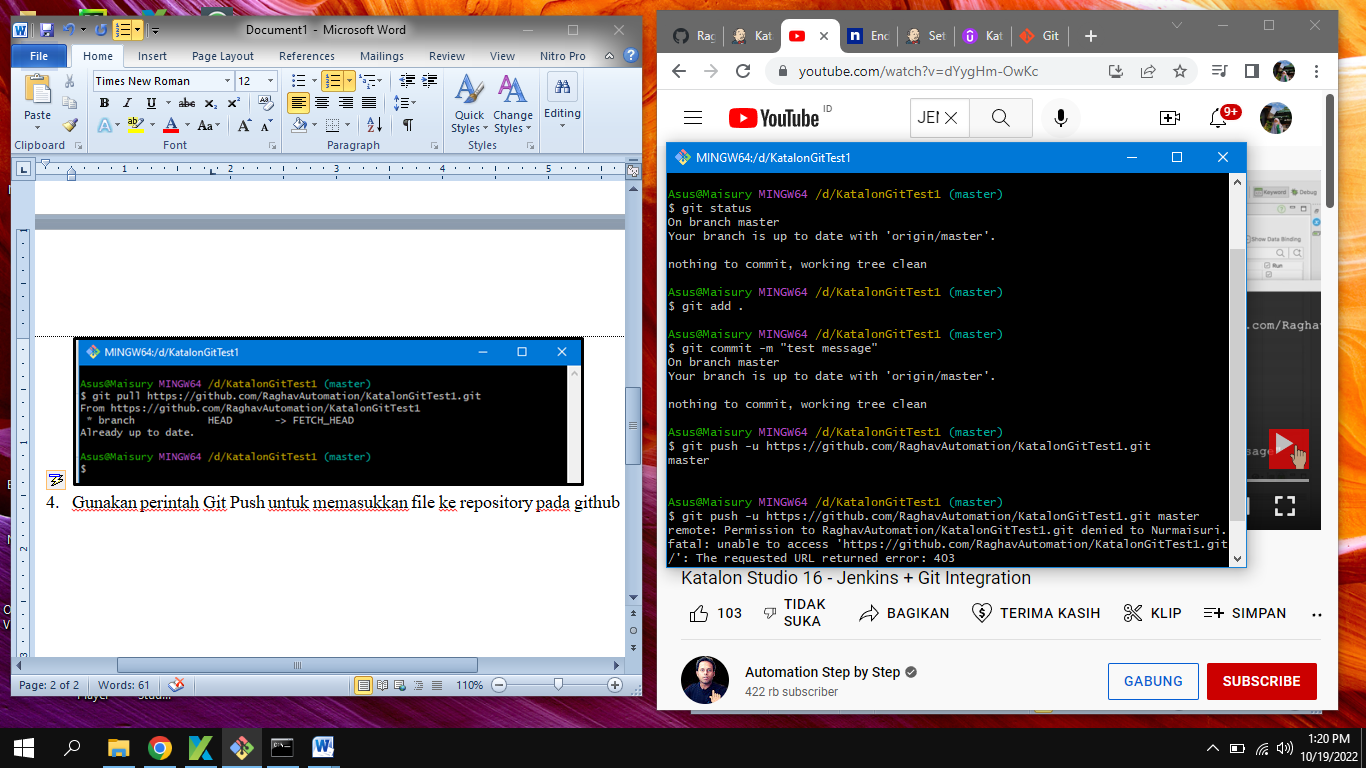
Tahap 3 Uji perintah Git

1. Buka terminal cmd
2. Masuk ke github salin link



1. Setelah salin link pada github, paste kan pada git bash



1. Gunakan perintah Git Push untuk memasukkan file ke repository pada github.

# BAB 6

## Pendahuluan

## Dasar Teori

## Percobaan