**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERBASIS WEB**

**“Pengenalan XHTML dan GITHUB ”**



Disusun Oleh :

Nama : I Kadek Ari Surya

Nim : 1808561026

Kelas : B

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**BADUNG**

**2020**

# **KATA PENGANTAR**

Puji dan syukur ke hadirat Tuhan Yang Maha Kuasa atas segala rahmat yang diberikan-Nya sehingga tugas Laporan Praktikum Pemrograman Berbasis Web yang berjudul ”Pengenalan XHTML dan GITHUB” ini dapat saya selesaikan. Laporan ini saya buat sebagai kewajiban untuk memenuhi tugas. Dalam kesempatan ini, penulis menghaturkan terimakasih yang dalam kepada semua pihak yang telah membantu menyumbangkan ide dan pikiran mereka demi terwujudnya makalah ini. Akhirnya saran dan kritik pembaca yang dimaksud untuk mewujudkan kesempurnaan makalah ini penulis sangat hargai.

Badung, 25 September 2020

Penyusun

# **DAFTAR ISI**

**COVER**

[**KATA PENGANTAR** ii](#_Toc52233768)

[**DAFTAR ISI** iii](#_Toc52233769)

[**BAB I PENDAHULUAN** 1](#_Toc52233770)

[**1.1** **Latar Belakang** 1](#_Toc52233771)

[**1.2** **Tujuan** 1](#_Toc52233772)

[**1.3** **Manfaat** 1](#_Toc52233773)

[**BAB II TINJAUAN PUSTAKA** 2](#_Toc52233774)

[**2.1** **XHMTL** 2](#_Toc52233775)

[**2.2** **GIT** 3](#_Toc52233776)

[**2.3** **GITHUB** 4](#_Toc52233777)

[**2.4** **Visual Studio Code** 6](#_Toc52233778)

[**BAB III PEMBAHASAN** 7](#_Toc52233779)

[**3.1** **Perintah Dasar Pada XHTML** 7](#_Toc52233780)

[**3.2** **Cara Membuat *Directory* di *Github*** 8](#_Toc52233781)

[**3.3** **Cara *Upload File* ke *Directory* di *Github*** 12](#_Toc52233782)

[**BAB IV PENUTUP** 18](#_Toc52233783)

[**4.1. Simpulan** 18](#_Toc52233784)

[**4.2. Saran** 18](#_Toc52233785)

[**DAFTAR PUSTAKA** 20](#_Toc52233786)

# **BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Sekarang ini *website* telah menjadi kebutuhan yang pokok dalam dunia bisnis dan internet. Macam dan jenis nya beragam menjadikan web ini menjadi teknologi interaktif sekaligus menjadi bumerang saat tidak ada proses pengaturan. Seiring dengan berkembang teknologi web yang semakin kreatif, maka harus ada standarisasi web yang baik. W3C merupakan organisasi yang mengatur standarisasi web yang menjadi pokok bahasan sekarang ini. W3C menawarkan konsep teknologi XHTML yang merupakan gabungan HTML dan XML, dimana dengan pengaturan yang baik dan benar pada XHTML dokumen akan memungkinkan proses otomatis yang akan dilakukan dengan menggunakana alat-alat standar XML, tidak seperti HTML yang membutuhkan yang relatif rumit, sabar, dan umumnya *custom parser*. Disamping itu dalam membangun *website* secara kolektif dibutuhkan suatu media yang bisa mewadahi kelompok tim dalam membangun kolaborasi kode program.

Untuk itulah dalam praktikum ini akan dipelajari perintah-perintah dasar XHMTL dalam membuat *website* terutama bagi pemula. Selain itu juga dikenalan GITHUB sebagai media yang bisa digunakan oleh *developer* dalam membangun *website* secara kolektif dengan mudah dan terstruktur.

* 1. **Tujuan**

1. Mengimplementasikan perintah dasar XHTML.
2. Mengimplementasikan *upload directory* pada akun GITHUB.
   1. **Manfaat**
3. Mengetahui perintah dasar yang ada pada XHTML.
4. Mengetahui cara untuk *upload directory* pada akun GITHUB.

# **BAB II**

**KAJIAN PUSTAKA**

1. **XHMTL**

XHTML adalah sebuah keluarga HTML, merupakan masa depan tipe dokumen dan modul yang hasil perkembangan, subset, dan regenerasi HTML 4. XHTML adalah tipe dokumen HTML berbasis XML, dan pada akhirnya dirancang untuk bekerja bersama-sama dengan pengguna berbasis client XML.

XHTML 1.0 adalah satu jenis dokumen dalam keluarga XHTML. Ini adalah satu dari tiga reformulasi HTML 4 bertipe dokumen aplikasi XML 1.0. Hal ini dimaksudkan untuk digunakan sebagai bahasa untuk konten yang baik dan penyesuaian XML, jika sederhana beberapa pedoman yang diikuti, beroperasi dalam HTML 4 penyesuian agen pengguna. Pengembang konten mereka yang bermigrasi ke XHTML 1.0 akan menyadari manfaat sebagai berikut:

1. Mereka mudah dilihat, diedit, dan validasi dengan alat standar XML.
2. XHTML dokumen dapat ditulis untuk beroperasi sebagai baik atau lebih baik daripada yang mereka lakukan sebelumnya yang ada di HTML 4 penyesuaian pengguna agen maupun baru, XHTML 1.0 penyesuaian agen pengguna.
3. XHTML dokumen dapat memanfaatkan aplikasi (misalnya skrip dan applet) yang bergantung pada salah satu HTML Document Object Model atau XML Document Object Model.
4. Sebagai keluarga perkembangan XHTML, dokumen penyesuaian ke XHTML 1.0 akan lebih cenderung interoperate di dalam dan di antara berbagai XHTML lingkungan.

Keluarga XHTML merupakan langkah berikutnya dalam evolusi Internet. Oleh migrasi ke XHTML hari ini, pengembang konten dapat memasukkan XML dunia dengan semua yang hadir manfaat, sementara sisanya masih yakin konten mereka dari masa depan dan kompatibilitas ke belakang.

1. **GIT**

GIT adalah *software* (perangkat lunak) untuk mengontrol versi atau untuk memanajemen sebuah *project* aplikasi yang diciptakan oleh Linus Torvalds. Desain tampilan pada GIT sendiri, terinspirasi dari BitKeeper dan Monotone. GIT sebenarnya dirancang sebagai alat yang dapat digunakan oleh tampilan muka (front end) lain seperti CoGITo atau StGIT. Dalam pengoperasian GIT bisa melalui GUI (aplikasi) maupun CLI (perintah menggunakan text), dan banyak juga aplikasi yang mendukung GIT dan salah satunya Android Studio. GIT juga bisa di jalankan di beberapa aplikasi lain dan beberapa aplikasi tersebut membutuhkan plugin untuk menjalankannya sebagai contoh aplikasi Notepad++ yang membutuhkan plugin bantuan agar mempermudah programer dalam mengakses GIT. Area yang dikenali pada git adalah :

* *Working Tree*

Folder tempat file yang dikerjakan pada local komputer atau folder biasa yang berisikan file-file.

* *Staging Area*

Area ini memberi tahu GIT bahwa telah terjadi perubahan pada file di dalam folder yang sudah kita kerjakan.

* *History*

Area ini akan menyimpan hasil *commit*  dari perubahan file yang sudah masuk pada staging area. Staging Area dan History akan disimpan pada folder .git yang mana lokasi tersembunyi dalam folder yang sudah diinisialisasi yang nantinya dikelola secara otomatis oleh git.

Berikut ini beberapa perintah dasar yang sering digunakan pada git :

* git config : digunakan untuk mengatur konfigurasi tertentu sesuai keinginan pengguna, seperti email, algoritma untuk diff, username, format file, dll.
* git init : Perintah ini digunakan untuk membuat repositori baru.
* git add : Perintah git add bisa digunakan untuk menambahkan file ke index.
* git clone : Perintah git clone digunakan untuk mengambil repositori.
* git commit : Perintah git commit digunakan untuk melakukan commit pada perubahan ke head.
* git status : Perintah git status menampilkan daftar file yang berubah bersama dengan file yang ingin di tambahkan atau di-commit.
* git push : Push akan mengirimkan perubahan ke master branch dari remote repository yang berhubungan dengan direktori kerja.
* git remote : Perintah git remote akan membuat user terhubung ke remote repository.
* git branch : Perintah git branch bisa digunakan untuk me-list, membuat atau menghapus branch.
* git pull : Untuk menggabungkan semua perubahan yang ada di remote repository ke direktori local.
* git merge : Perintah merge digunakan untuk menggabungkan sebuah branch ke branch aktif.
* git fetch : Perintah ini digunakan untuk menampilkan semua object dari remote repository yang tidak berada di direktori kerja lokal.

1. **GITHUB**

GITHUB atau github.com adalah Hosting untuk proyek open source yang menggunakan Tool System revisi kontrol yaitu GIT. Jadi GIT adalah tool untuk melakukan revisi code, sedangkan GITHUB adalah webhostingnya. Jadi GITHUB adalah Webhosting untuk mengerjakan dan menyimpan proyekproyek software yang sama seperti sourceforge.net, GITHUB bisa dikatakan sebagai jejaring sosial untuk software developer (Putra, 2014). Dengan adanya GITHUB mempermudah programer dalam bekerjasama membangun sebuah aplikasi dengan jarak yang berjauhan, dengan menggunakan koneksi internet. Programer tingal perlu mengatur sambungan antara GIT dan respository yang ada di GITHUB agar dapat mengatur file yang ada di GITHUB.

Berikut istilah-istilah yang sering digunakan pada github:

* Repository : Database yang menyimpan history/ riwayat perubahan.
* Snapshoot : Potret kondisi file dan folder pada saat tertentu.
* Commit : Snapshot yang disimpan di repository.
* Branch : Serangkaian commit yang berkaitan sehingga kalau digambar seperti garis lurus berisi banyak commit. Satu repository bisa berisi banyak branch.
* Master : Nama branch default yang diberikan git pada waktu kita membuat repository. Branch master ini tidak istimewa. Dia bisa dihapus dan direname sesuka hati.
* Pull Request : Pull request adalah ketika Anda menginformasikan user bahwa Anda sudah push perubahan yang dilakukan di branch ke master repositori. Collaborator repositori akan menerima atau menolak pull request. Segera setelah pull request diterima, Anda bisa mendiskusikan dan mengulas proyek bersama dengan collaborator.
* Hash : Hash adalah suatu penanda unik pada sebuah commit.
* Merge : Merge adalah suatu cara untuk menggabungkan 2 buah branch.
* Remote : Remote adalah sumber yang memiliki repository.
* Clone : Clone adalah mengambil repository dari commit.
* Push : Push adalah mengirimkan commit ke repository. Push merupakan suatu proses mengirim file dari laptop ke server github.
* Pull : Pull adalah suatu proses mengambil commit repository atau suatu proses mengirim file dari server github ke laptop.

1. **Visual Studio Code**

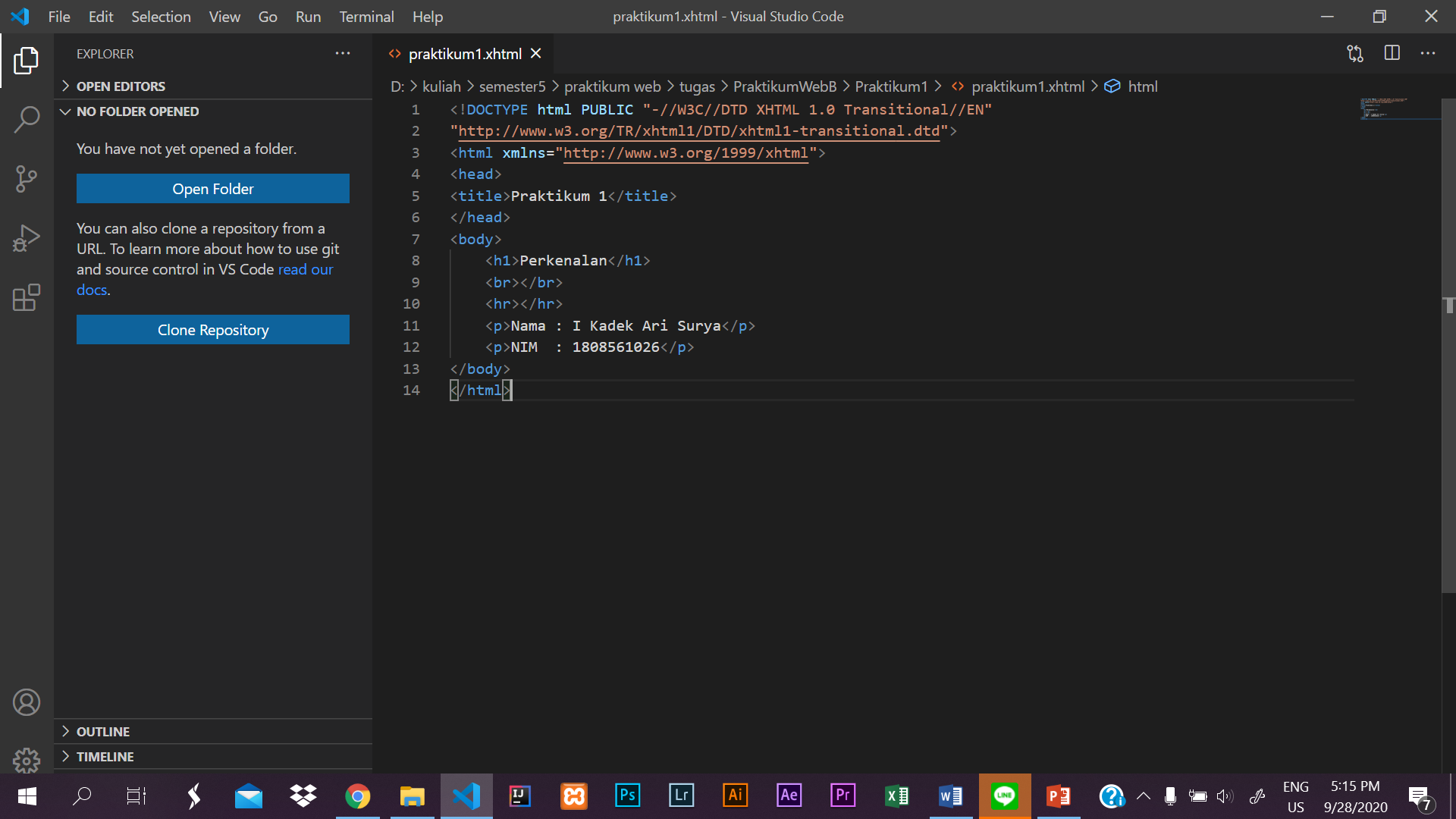
Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Controlyang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi.Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary. Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blinklayout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online.

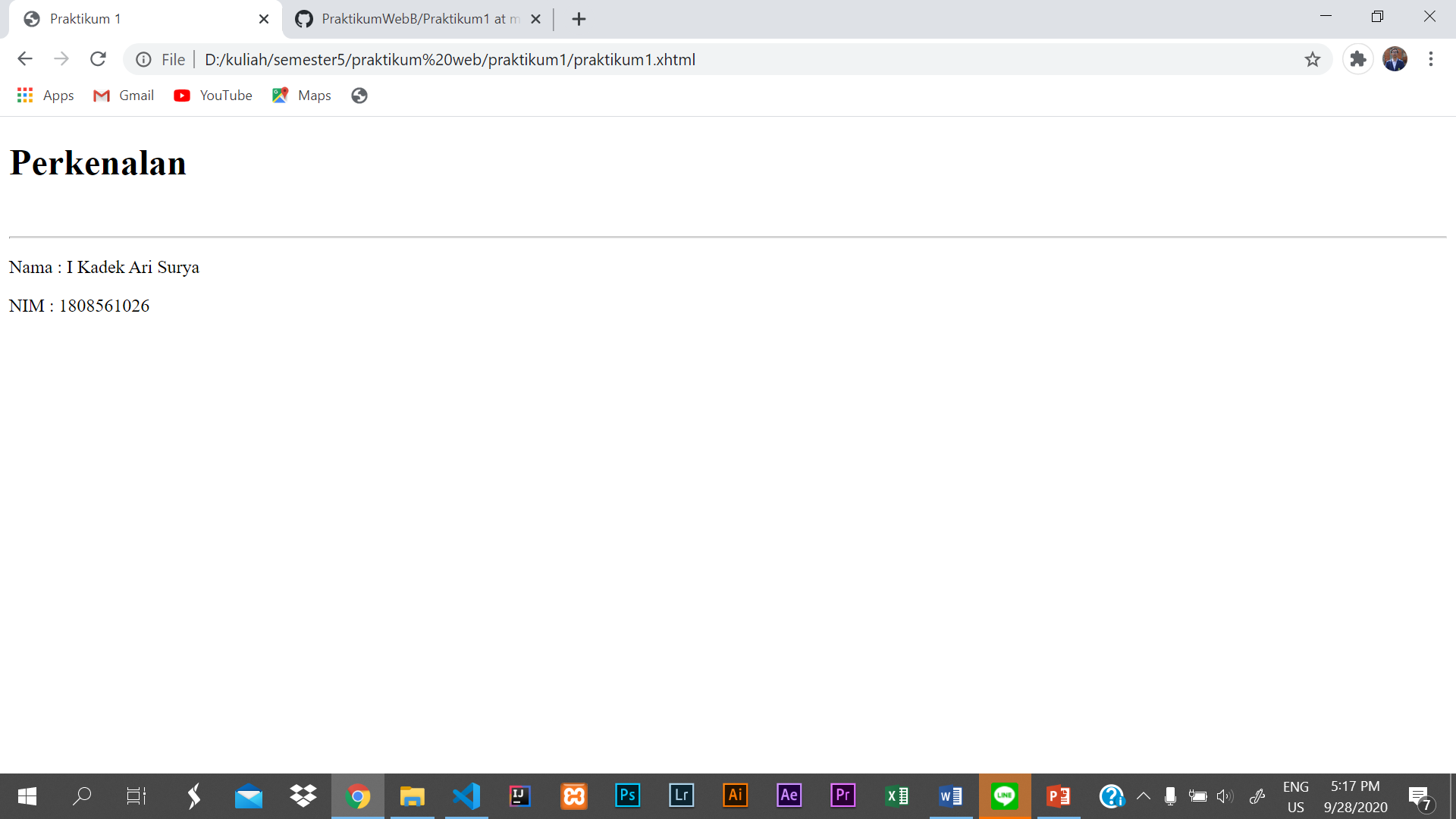
# **BAB III**

**PEMBAHASAN**

1. **Perintah Dasar Pada XHTML**

Pada pembahasan ini, diberikan soal untuk membuat file dengan format .xhtml yang berfungsi untuk menampilkan nama dan nim. Untuk membuat file tersebut, digunakan text editor *Visual Studio Code*. Berikut adalah codingan dan hasil *output* dari program.



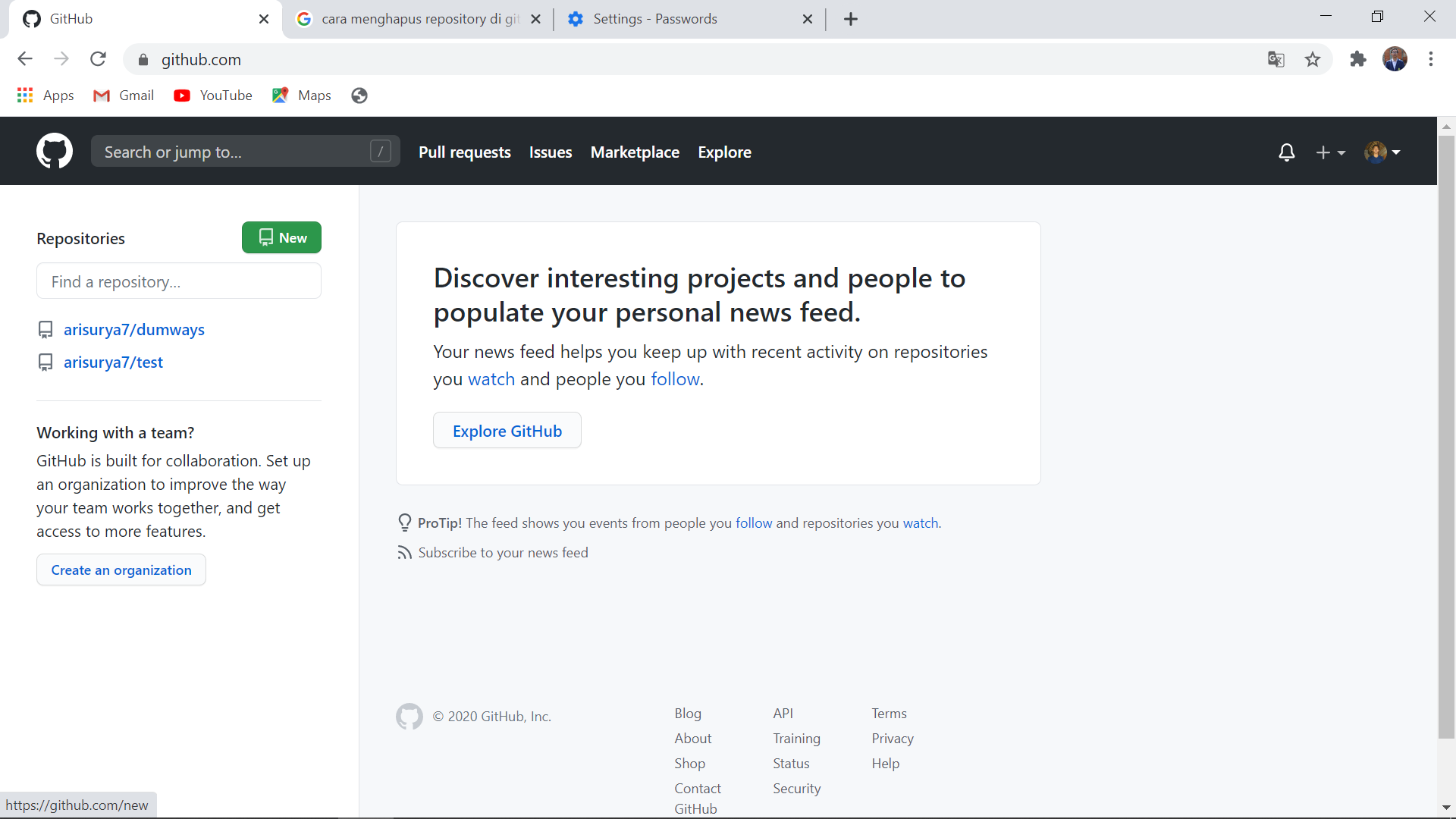


Penjelasan :

* Pertama kita harus membuat kerangka file dari .xhtml. Kita deklarasikan tipe dari dokumen yang berupa .xhtml dengan menautkan html PUBLIC yang dihubungkan dengan *XHTML 1.0 Transitional*. Dilanjutkan dengan tag *html* yang didalamnya berisi tautan xmlns. Kemudian dibuat kerangka *head* dan *body*.
* *Title* pada *head* yang digunakan adalah “Praktikum 1”
* Pada bagian *body* <h1> </h1> digunakankan untuk menampilkan judul dengan *type Heading 1* atau ukuran tilisan judul yang terbesar.
* Kemudian untuk memberikan spasi antar komponen judul dan *paragraph* digunakan perintah <br></br> atau disebut *break.*
* Setelah itu dilanjutkan dengan pembuatan garis horizontal dengan perintah <hr></hr> atau disebut *horixzontal rule.*
* Terakhir kita membuat konten nama dan nim kita menggunakan tag <p></p> atau disebut *paragraph*.

1. **Cara Membuat *Directory* di *Github***

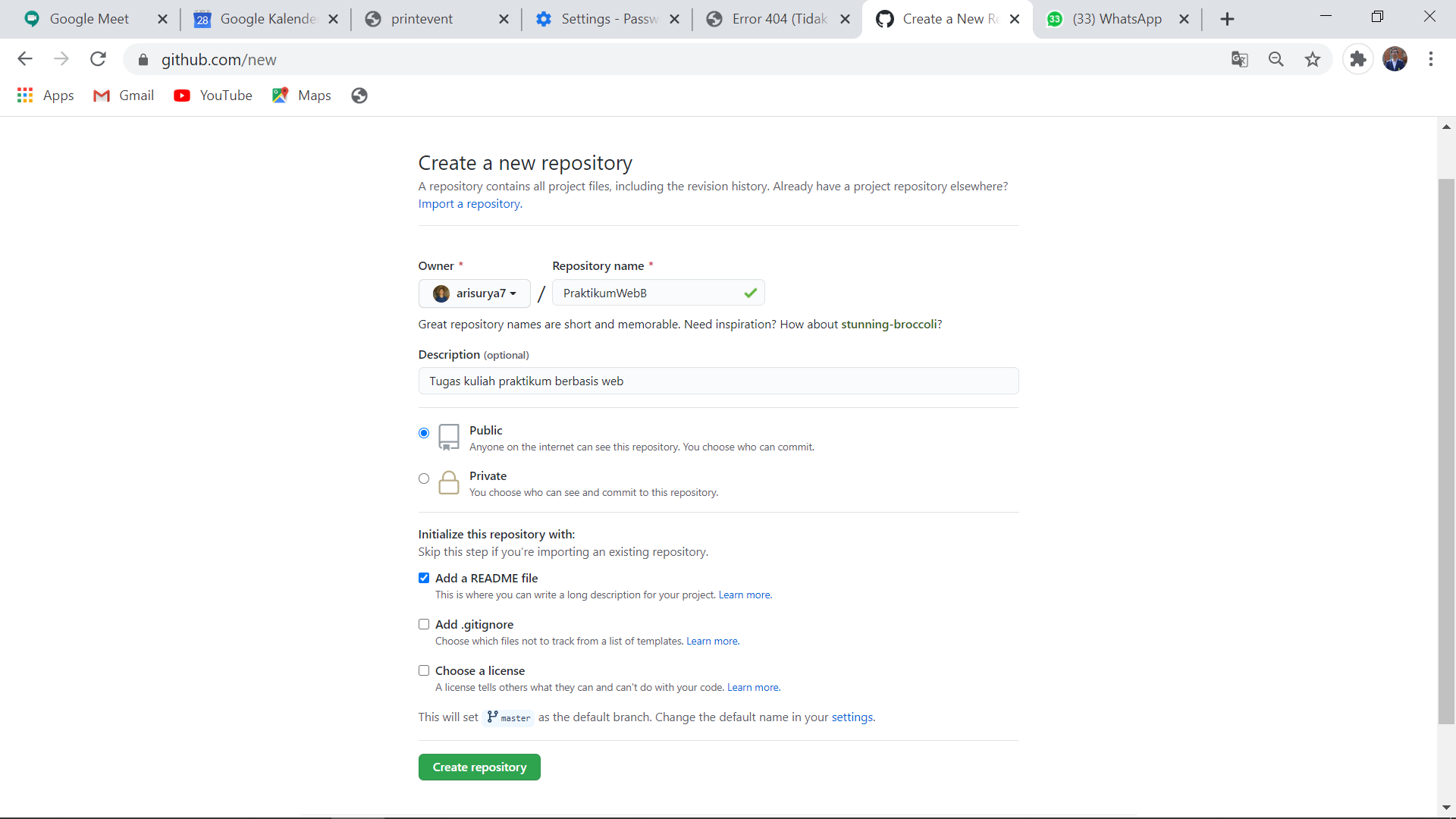
* Pada Github, *directory* disebut sebagai *repositories* atau folder. Untuk membuat *repositories* tentu terlebih dahulu harus mempunyai akun github. Nama akun github yang sudah saya buat adalah arisurya7.
* Setelah berhasil login pada github akan ditampilkan halaman utama seperti gambar dibawah. Untuk membuat *repositories*, tepat pada pojok kiri dibawah navbar terdapat *new* (tombol berwarna hijau) yang bisa kita klik untuk membuat *repositories* baru.



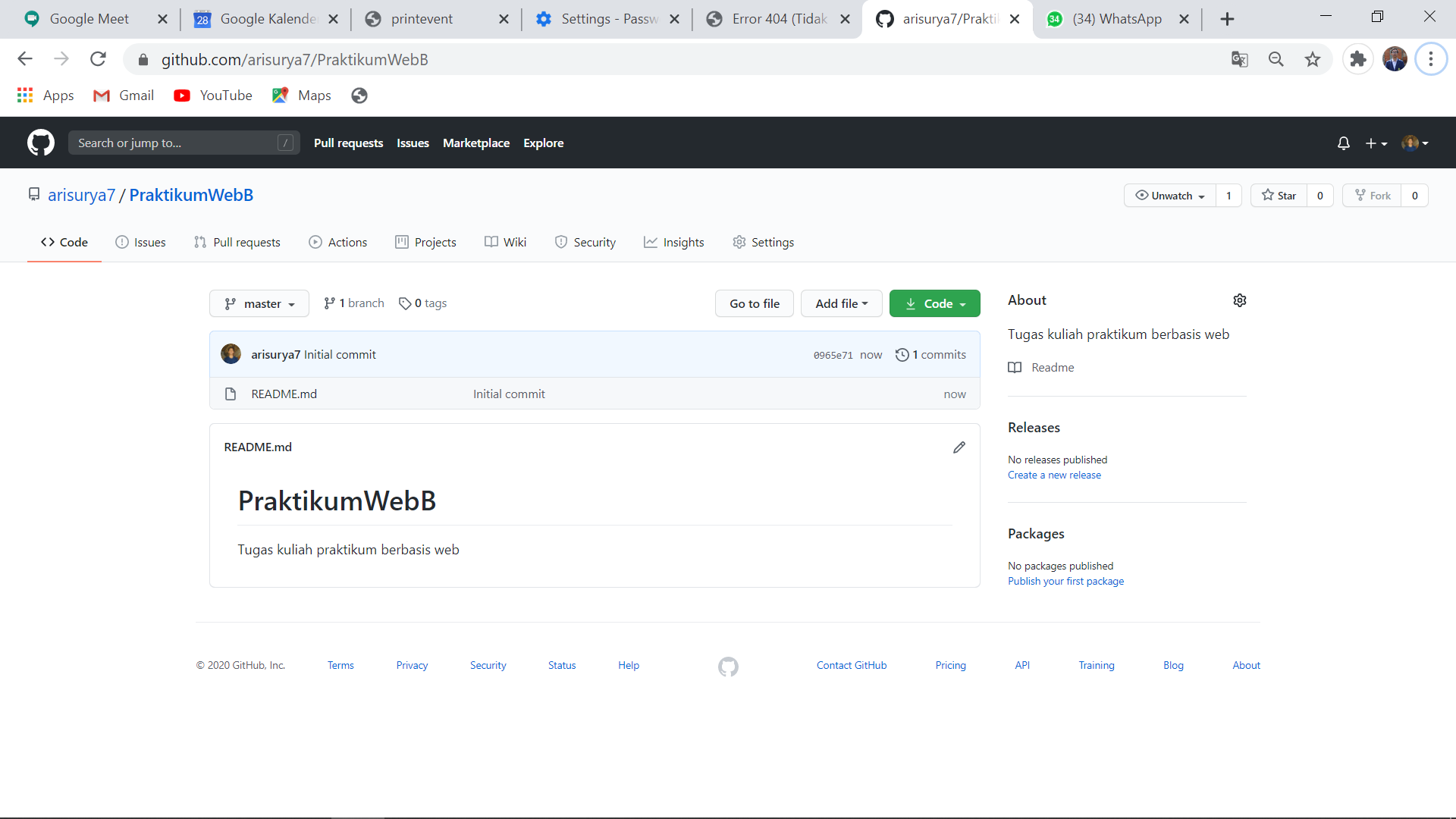
* Setelah klik tombol *new*, maka akan ditampilkan halaman baru sepeti pada gambar dibawah.
* Pada halaman tersebut terdapat beberapa kolom *owner* yang merupakan pemilik dari *account*. Kemudian disampingnya, disediakan kolom untuk mengisikan nama *repository* yang akan kita buat. Nama *repository* tidak boleh mengandung spasi. Jika menggunakan spasi makan secara default nanti akan diganti dengan tanda dash (-).
* Kemudian terdapat kolom deskripsi yang bisa kita isikan untuk memperjelas *repository* yang kita pergunakan untuk apa.
* Setelah itu kita akan memilih tipe dari *repository*, *public* atau *private*. Kerena nanti *repository* PraktikumWebB dibuat agar bisa diakses oleh semua orang maka kita pilih *public*. Namum jika sebaliknya, kita bisa memilih private agar repository tidak bisa diakses oleh semua orang.
* Kemudian pada *initilized*, kita disediakan pilhan check box untuk memilih *Add a README file*, *Add .gitignore*, dan *Choose a license*.
* *README* berfungsi untuk memberikan keterangan terhadap program atau file yang akan kita buat nanti.
* *Add .gitignore* adalah salah satu standar pengabaian atau *ignoring* perangkat lunak Git. Dan biasanya ketika kita mengambil suatu file dari *repository* baik dengan cara *clone* ataupun mengunduh langsung berbentuk ZIP, file .*gitignore* ini selalu diikutsertakan.
* *Add license* merupakan jenis lisensi dari projek kita. Bisa cek macam-macam lisensi yang bisa dipilih misalnya seperti GNU, MIT dan lain-lain.

Namun pada kasus ini, kita hanya memilih *README* karena *repository* yang kita buat belum terlalu kompleks.

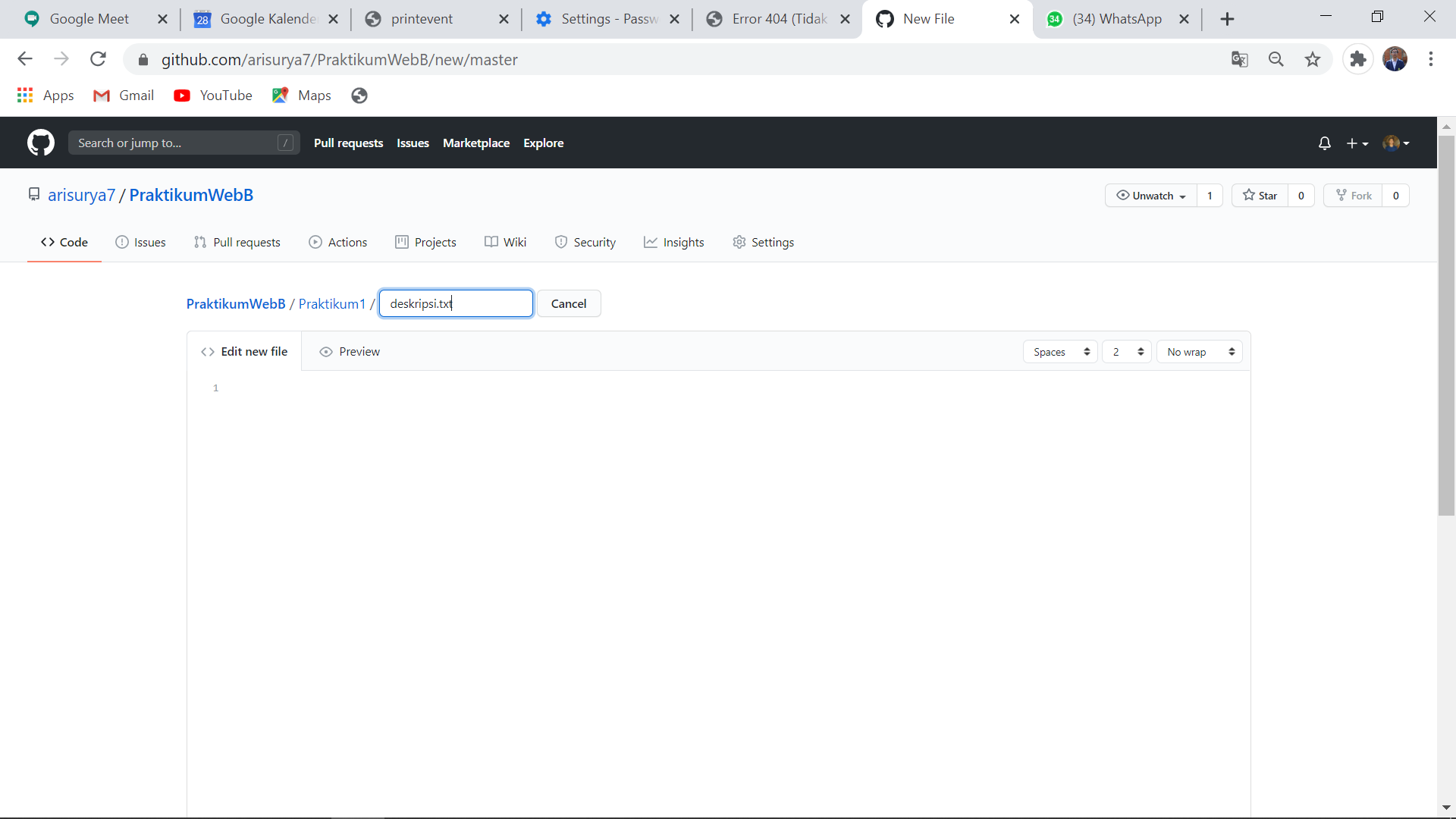
* Setelah itu kita klik *Create Repository.*



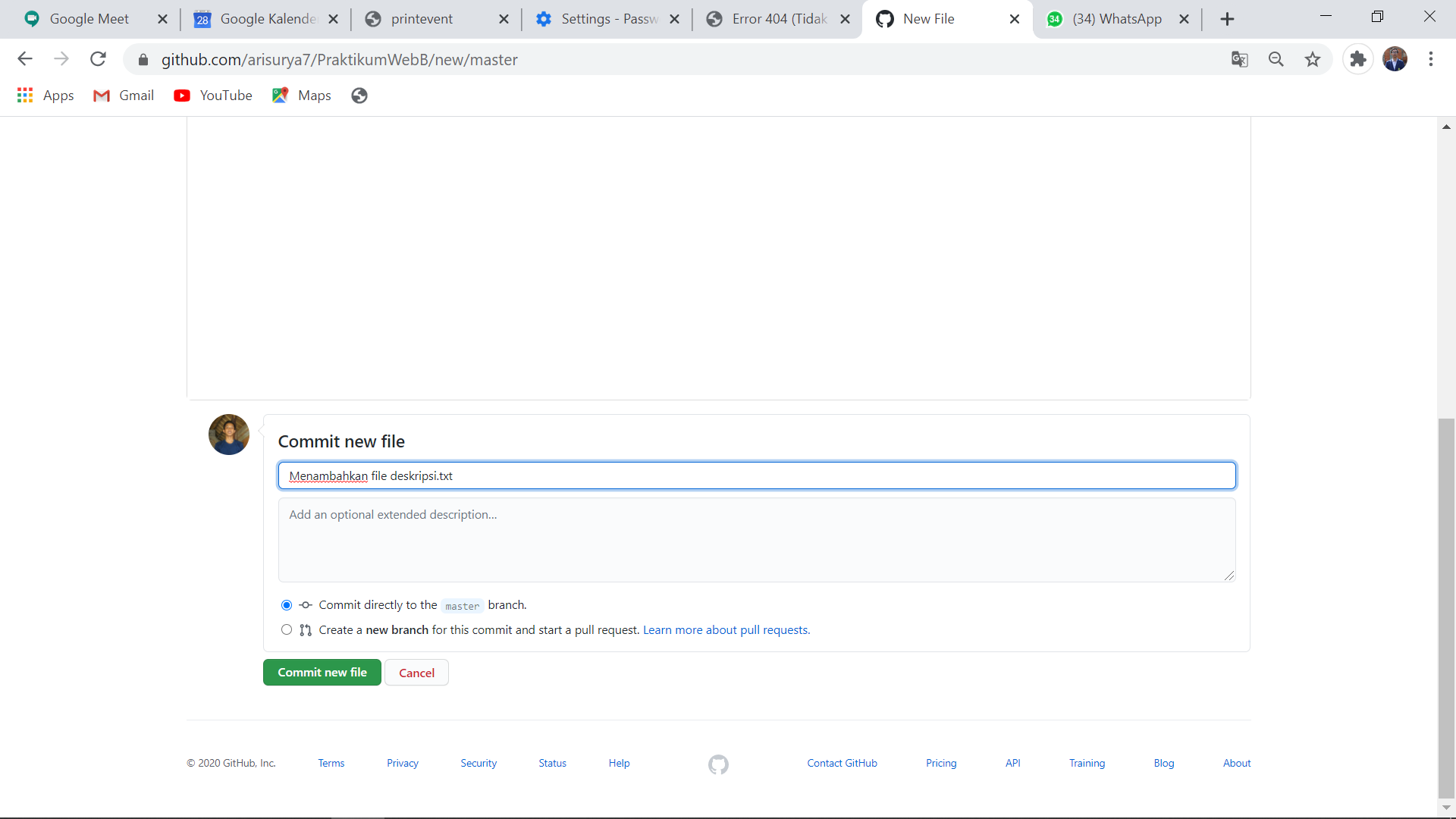
* *Repository* berhasil dibuat. Berikut ini adalah tampilan *widows* setelah kita berhasil membuat *repository* baru.



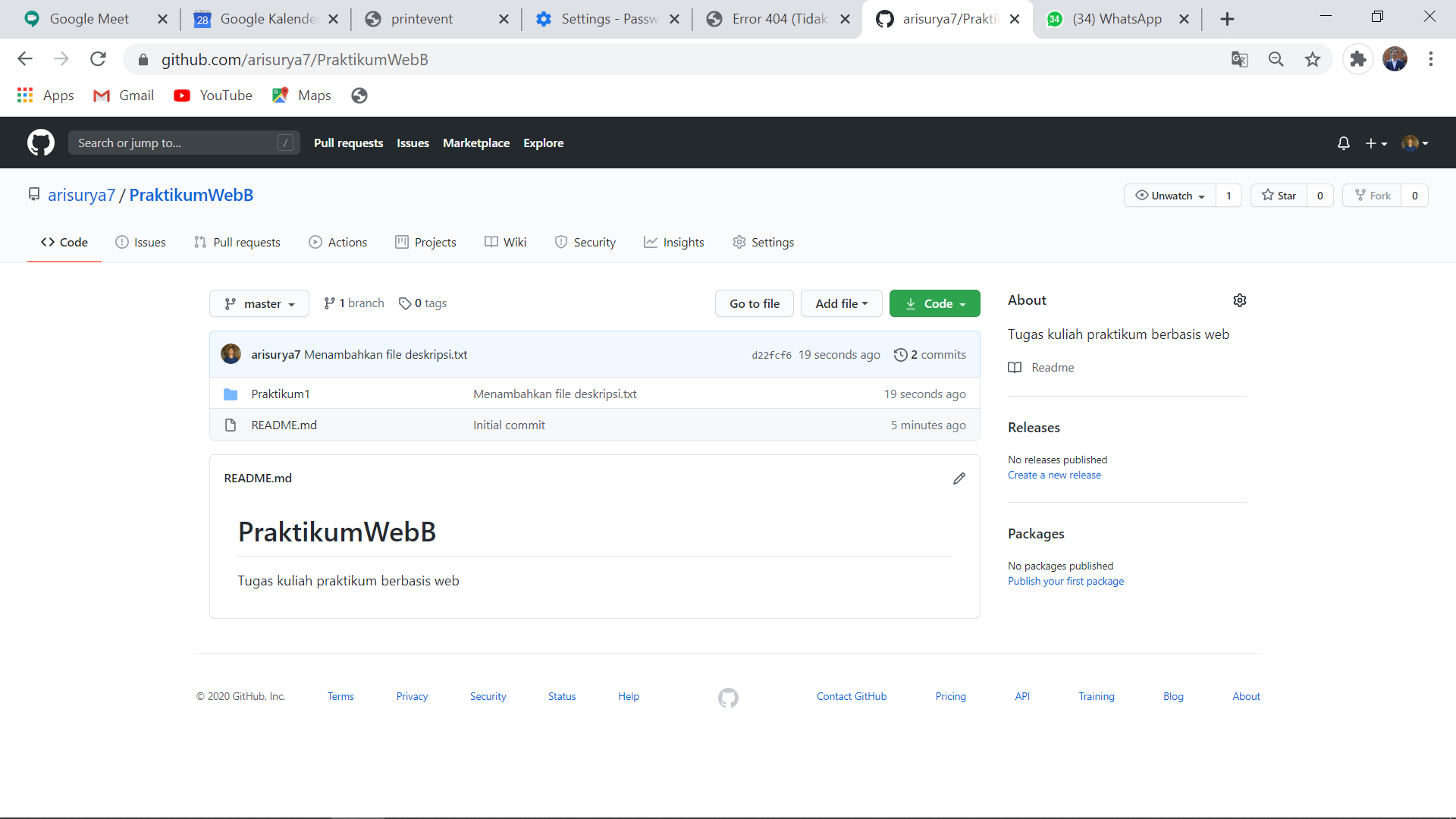
* Setelah *repository* PraktikumWebB berhasil dibuat, langkah selanjutnya dalah membuat folder Pertemuan1. Caranya kita menuju pada *Add File 🡪 Create new file*. Kemudian akan ditampilkan *box* untuk menuliskan nama file beserta formatnya. Karena kita akan membuat folder maka kita tuliskan Praktikum1/deskripsi.txt. Artinya kita membuat file deskripsi.txt yang terletak pada folder Praktikum1 dalam *repository* PraktikumWebB. Ketika kita membuat folder, minimal harus ada 1 file yang terdapat di dalamnya sehingga sebagai pengisi sementara kita membuat file deskripsi.txt yang bisa kita gunakan untuk memperjelas isi folder Praktikum1.



* Ketika sudah selesai membuat file dalam folder, langkah selanjutnya adalah melakukan *commit* untuk menyimpan perubahan yang terjadi. Terlebih dahulu kita juga bisa memberikan pesan perubahan apa yang terjadi. Kemudian klik *Commit new* *file*.

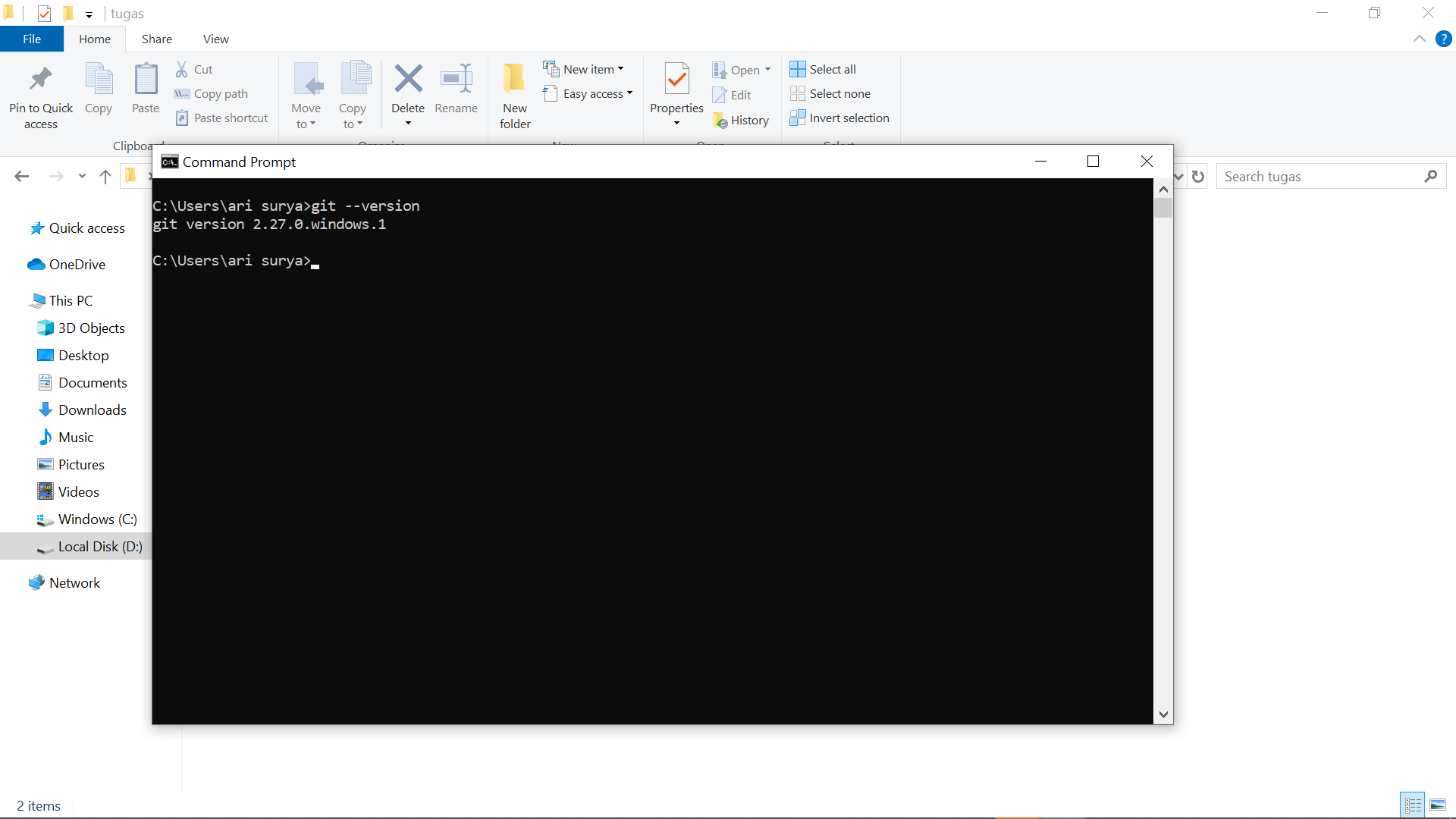


* Setelah melakukan *commit*, lakukan refresh pada *repository* PraktikumWebB dan folder Praktikum1 sudah berhasil dibuat.



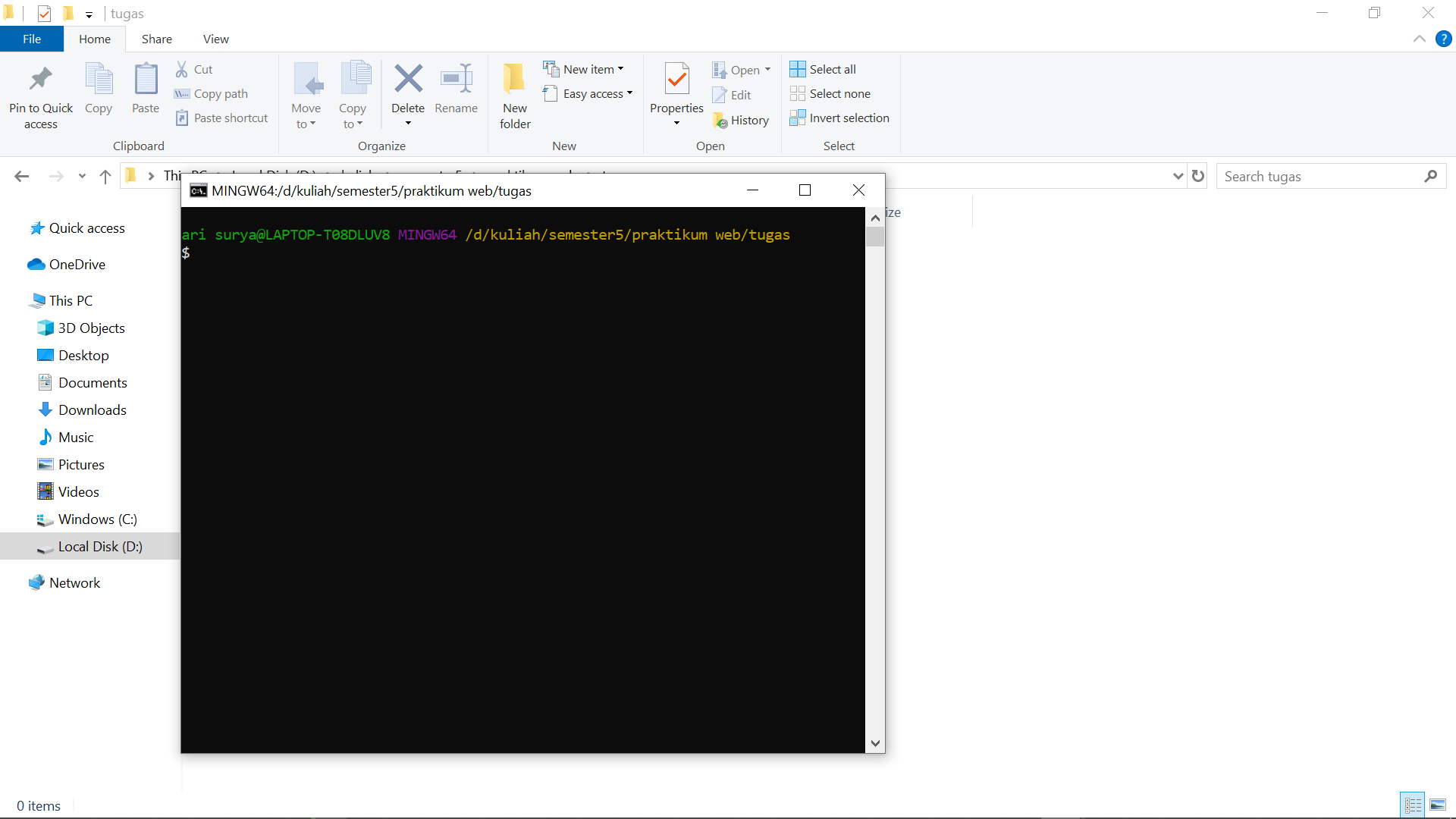
1. **Cara *Upload File* ke *Directory* di *Github***

* Untuk bisa mengupload file dari komputer kita ke *Github* terlebih dahulu kita harus menginstall *Git*. Kita bisa mendownloadnya di <https://git-scm.com/downloads> dan sesuaikan dengan sistem operasi komputer kita. Setelah selesai mendownload instalasinya, silahkan install file instalasi yang sudah di download hingga selesai. Karena sebelumnya saya sudah menginstal Git di komputer saya, untuk mengecek apakah sudah berhasil diinstal kita bisa buka command promt dan menuliskan perintah *git --version .*Versi *git* yang sudah saya install adalah *git version 2.27.0.windows.1*

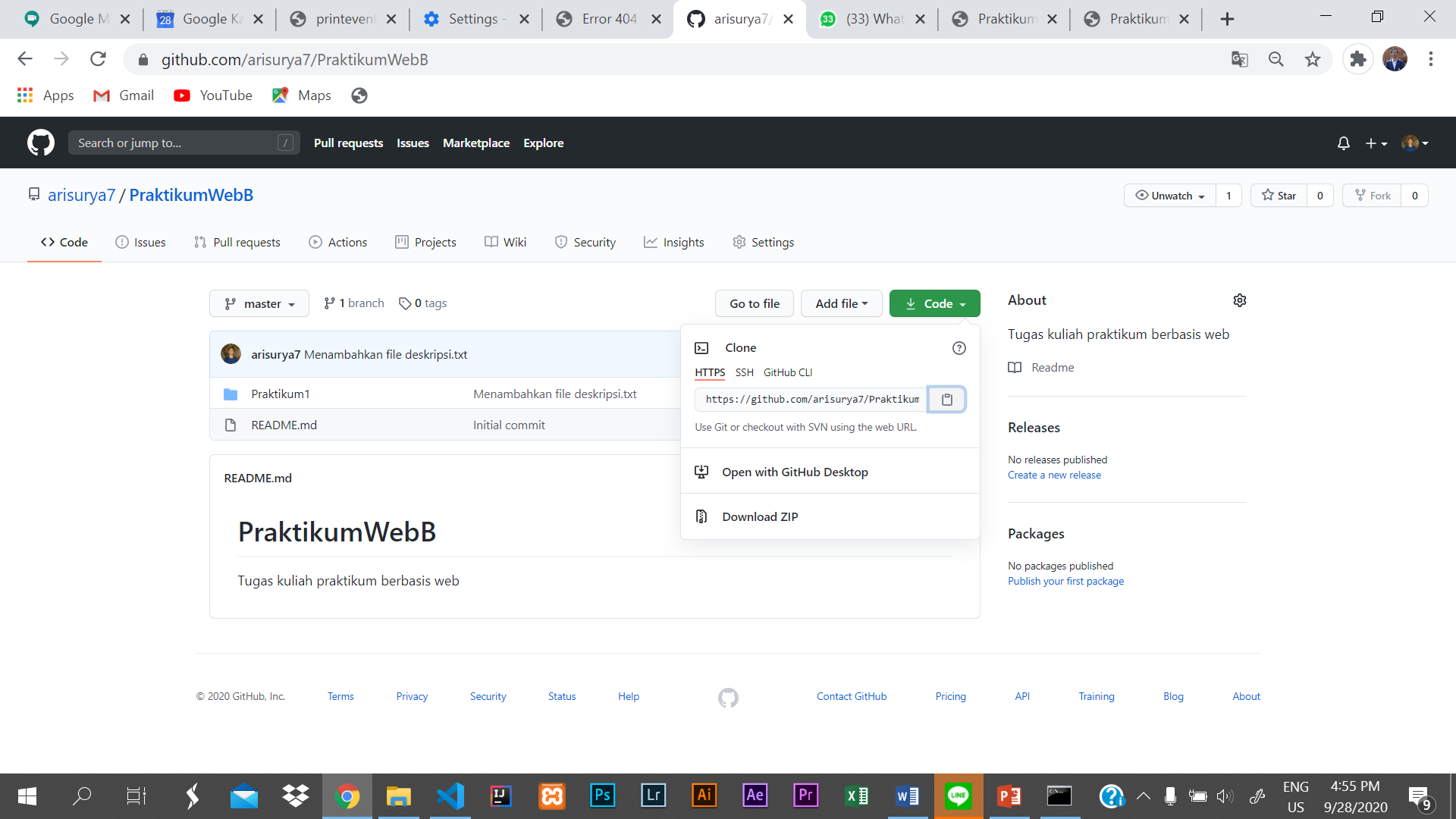


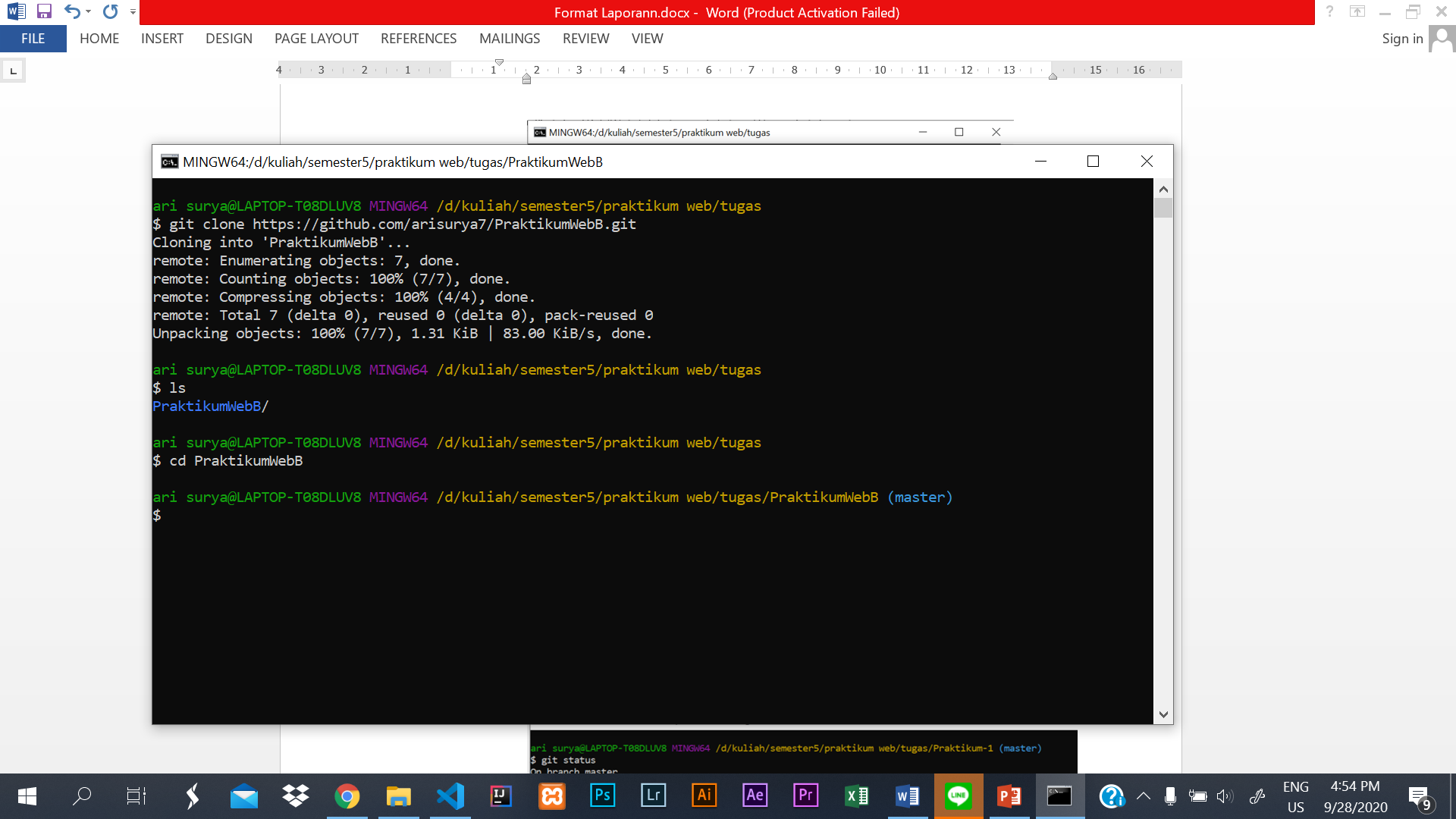
* Langkah berikutnya kita akan melakukan *clone* terhadap repository “PraktikumWebB” yang sebelumnya sudah kita buat untuk disalin dan dihubungkan dengan lokasi local dari folder yang kita simpan di komputer kita.

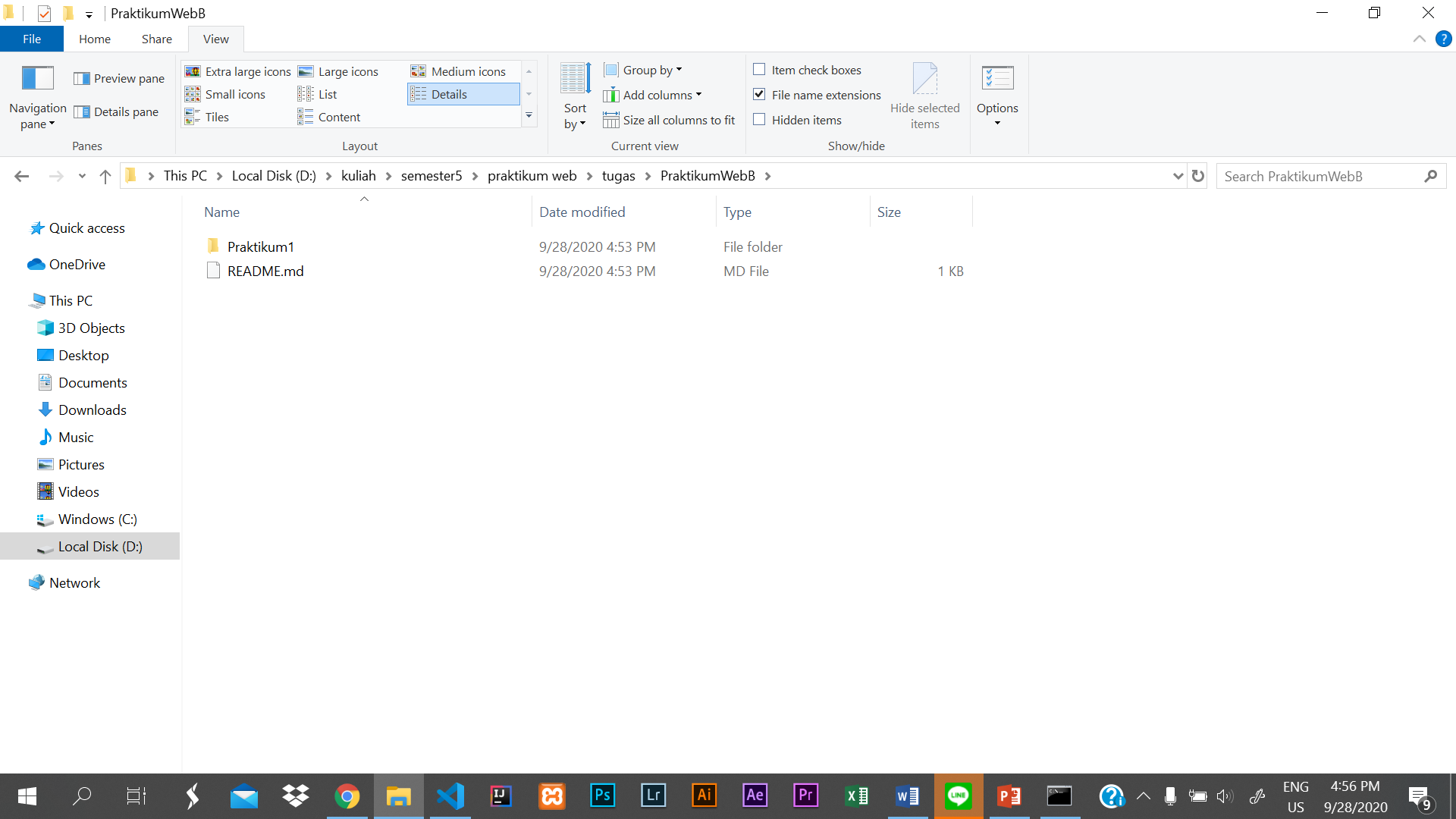
Setelah kita menuju direktori yang dituju untuk melakukan *clone* lalu klik kanan pilih *Git Bash Here*. Maka akan muncul tampilan seperti berikut.



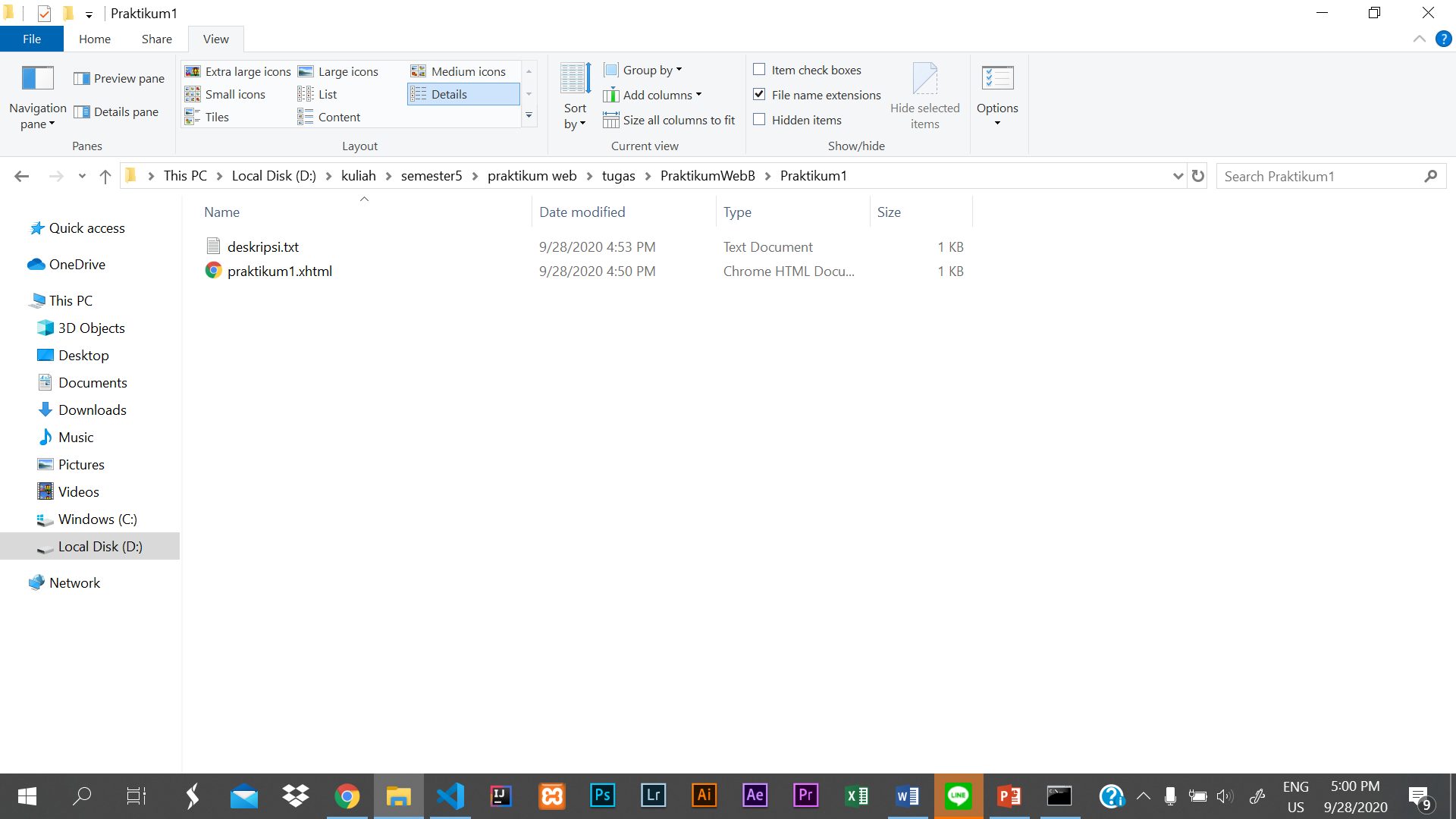
* Setelah muncul terminal bash, kita akan melakukan *clone* dengan perintah *git clone <url repository>*. Untuk *url repository* bisa kita *copy* pada *Code 🡪 HTTPS 🡪 Copy link*. Sehingga perintah pada terminal akan menjadi *git clone* [*https://github.com/arisurya7/PraktikumWebB.git*](https://github.com/arisurya7/PraktikumWebB.git)
* Setelah menggu beberapa detik maka *repository* yang ada di *Github* akan di *download* dan dihubungkan (*clone)* ke dalam lokasi *directory* pada lokal komputer yang kita sudah tentukan. Untuk mengecek apakah kita berhasil melakukan *clone*, kita bisa mengecek dengan perintah *ls* pada terminal atau mengecek direktorynya langsung pada file explorer. Dan hasilnya *repository* “PraktikumWebB” sudah berhasil di-*clone*  serta isinya juga sama dengan yang ada di github. Ketika kita melakukan c*lone,* secara tidak langsung kita juga telah menginisalisasi folder tersebut dan menghubungan *Github* dengan *Git* pada *repository* yang telah ditentukan. Setelah itu pastikan kita pada masuk pada *direktory* PraktikumWebB dengan menggunakan perintah *cd ParktikumWebB*.

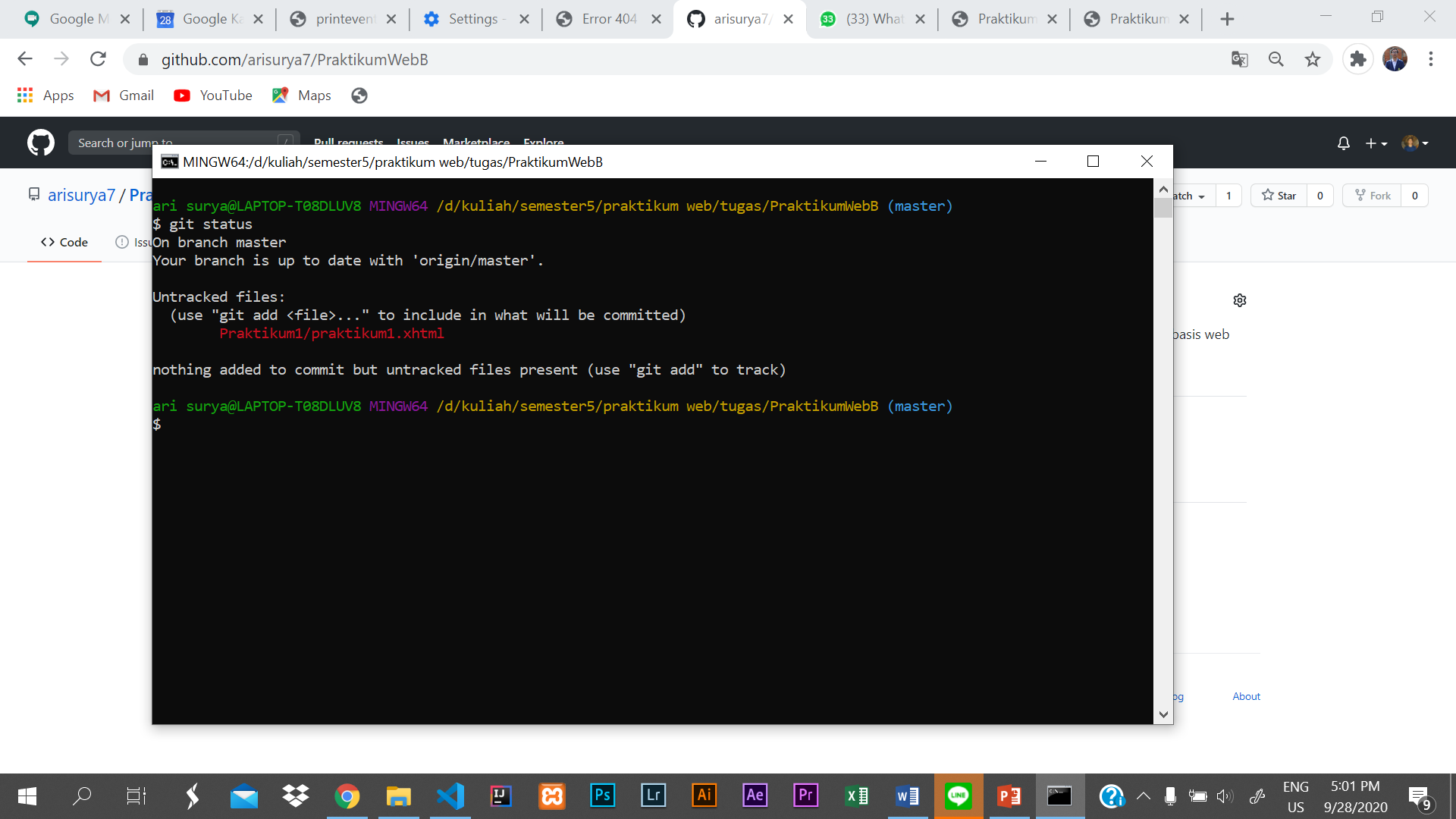




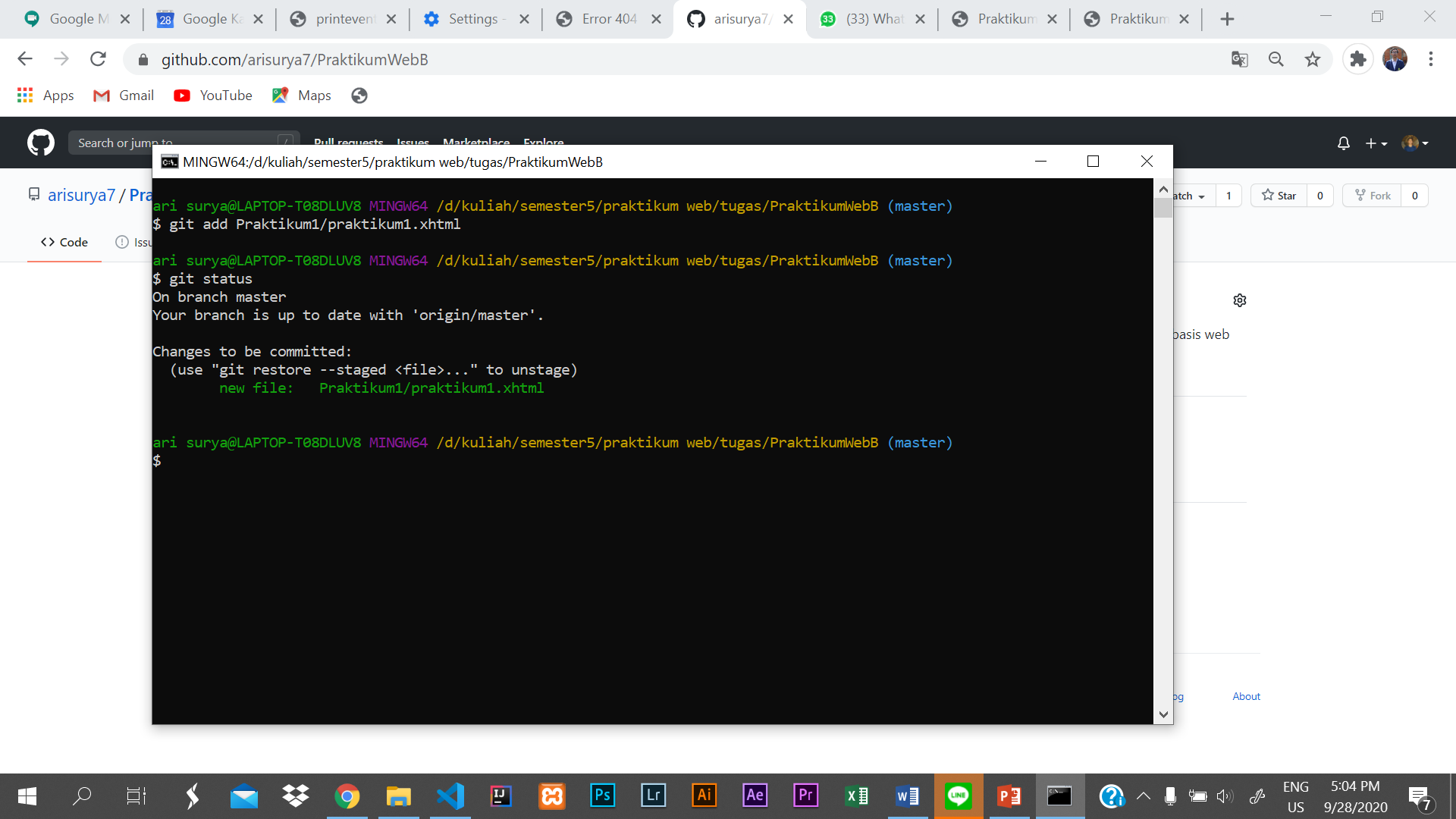


* Selanjutnya kita akan manambahkan file dengan nama *praktikum1.xhtml* yang sebelumnya sudah kita buat ke pada folder “Praktikum1”. Setelah itu kita menuju lagi pada terminal bash. Karena menambahkan file baru kedalam folder tersebut maka akan terjadi perubahan. Kita bisa mengeceknya dengan perintah *git status* untuk memastikan perubahan apa yang terjadi dalam folder tersebut.

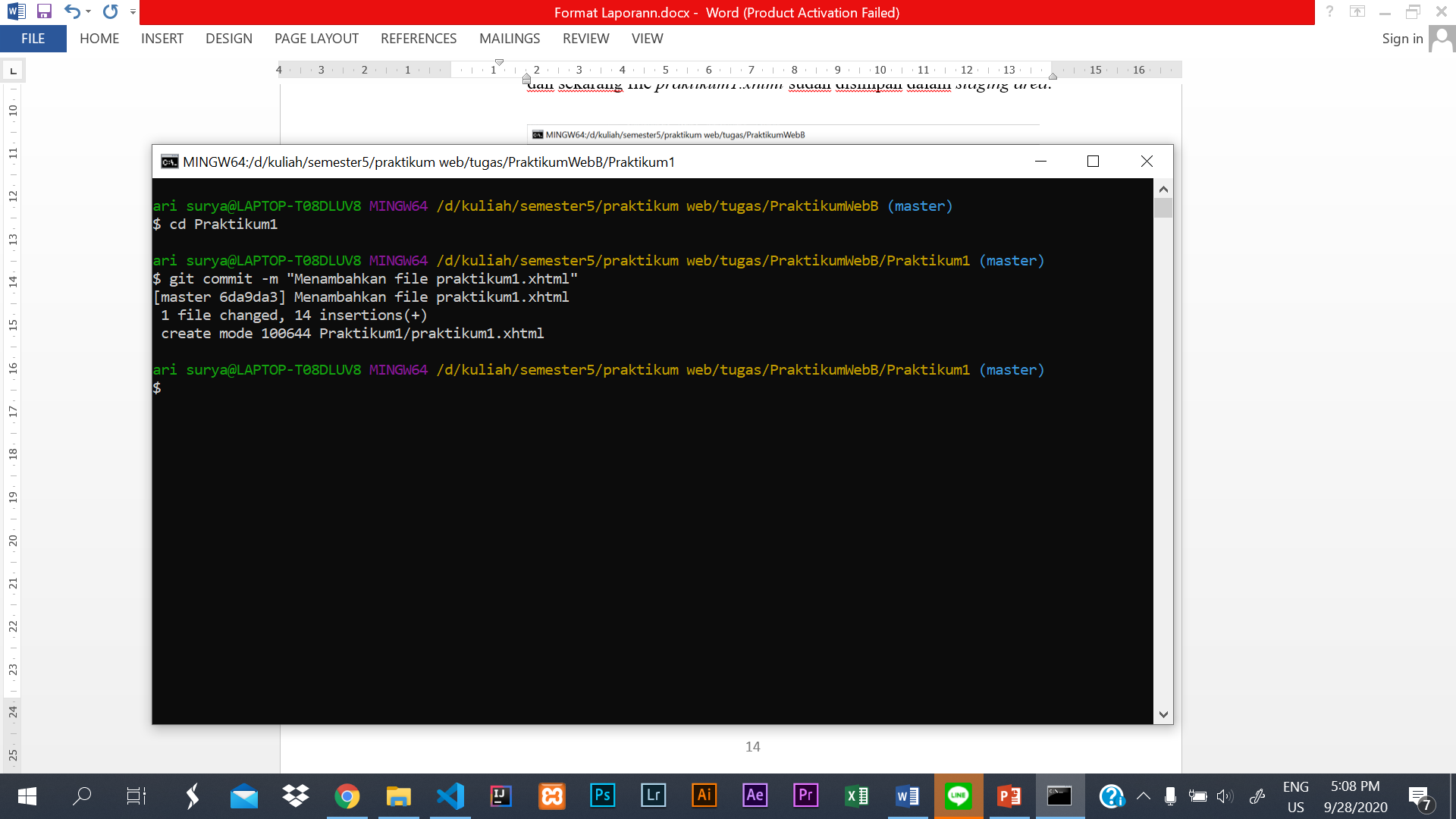




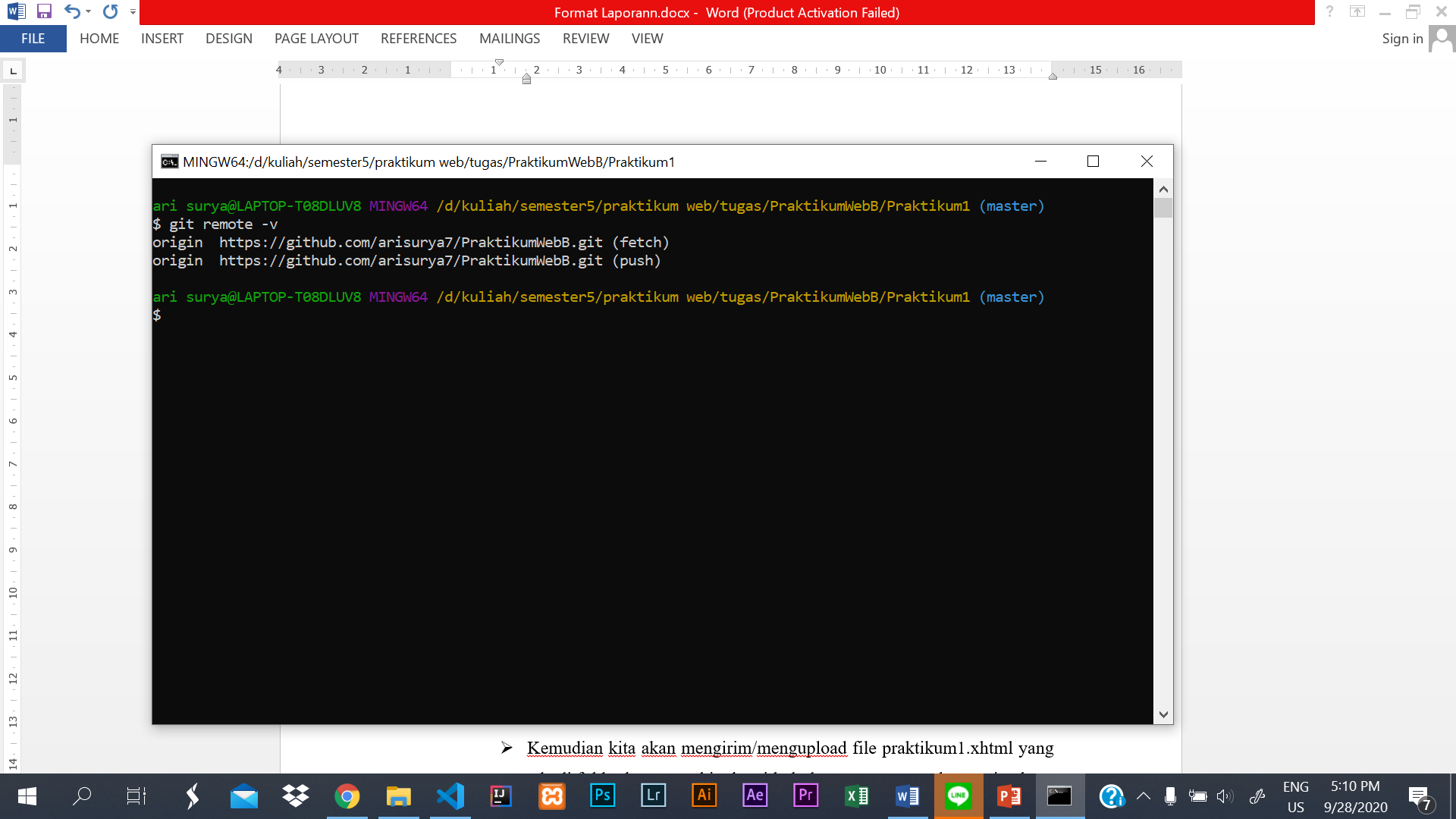
* Setelah mengetahui perubahan yang terjadi dan yang diketahui bahwa terdapat file baru dalam folder Praktikum1 dengan nama file *praktikum1.xhtml*. Untuk menyimpan perubahan tersebut kita perlu membawanya pada *Staging Area* dengan menggunakan perintah *git add Praktikum1/praktikum1.xhtml* atau bisa juga digunakan  *git add .* (untuk semua file baru). Kemudian kita akan cek lagi menggunakan *git status* dan sekarang file *praktikum1.xhtml* sudah disimpan dalam *staging area*.



* Selanjutnya kita akan membawa file *praktikum1.xhtml* ke dalam area *History* dengan menggunakan perintah *git commit -m (message).* Untuk lebih memudahkah kita akan masuk terlebih dahulu ke directory Praktikum1 dengan perintah *cd* Praktikum1. Pada terminal kita lanjutkan dengan menggunakan perintah *git commit –m* “*Menambahkan file praktikum1.xhmtl*”

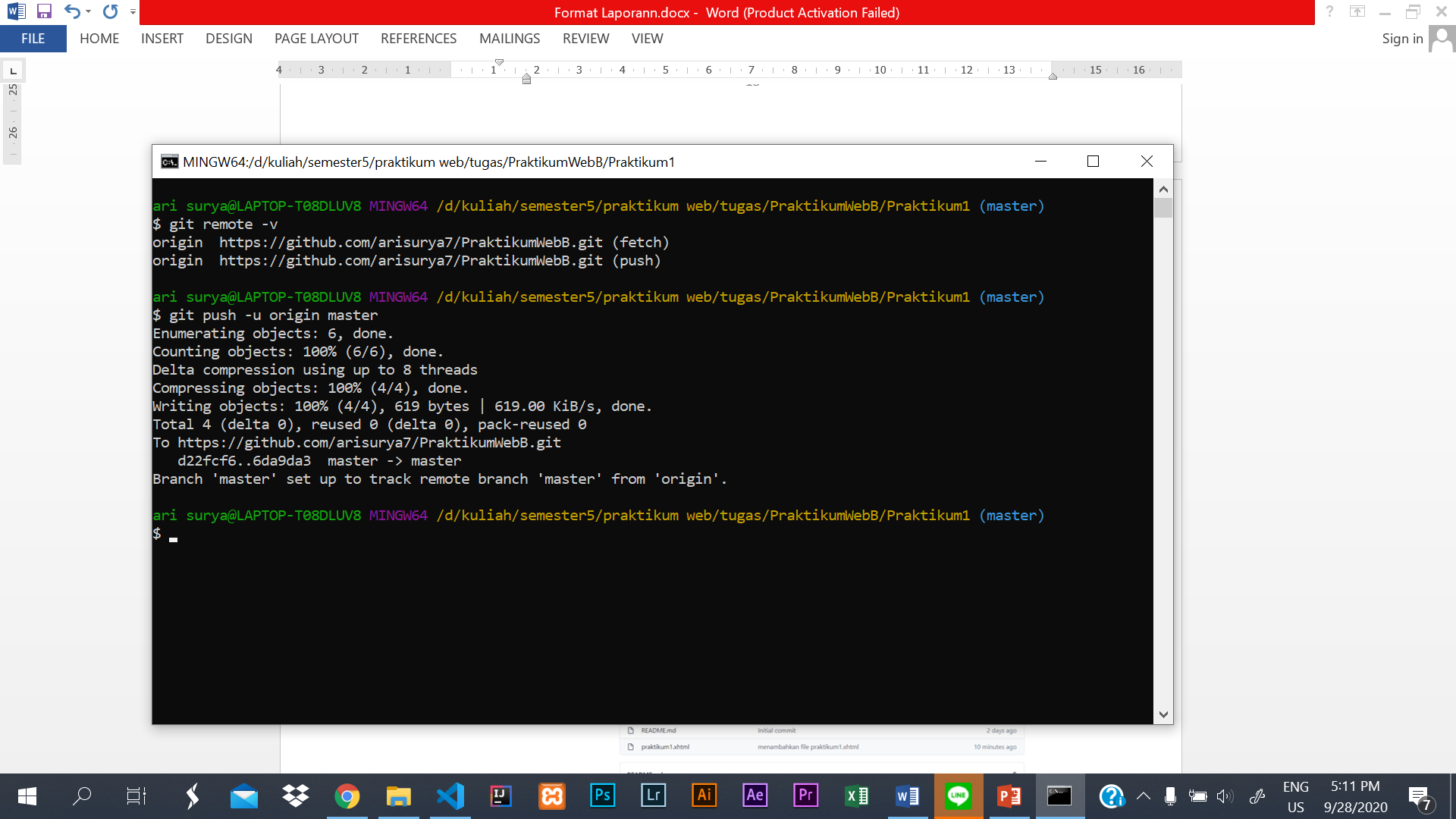


* Setelah berhasil melakukan *commit*, kita akan membuat remote dari git terhadapat *Github* yang merujuk pada *repository* “PraktikumWebB” pada *Github*. Karena kita sudah melakan clone pada tahan awal seharusnya remote dari folder ini sudah terbentuk. Untuk memasikannya lagi kita bisa menggunakan perintah *git remote –v* . Terlihat bahwa sudah benar remote yang dibuat dan merujuk pada alamat [*https://github.com/arisurya7/PraktikumWebB*](https://github.com/arisurya7/PraktikumWebB) . Namun jika belum terbentuk, kita bisa menggunakan perintah *git remote origin <url repository>*

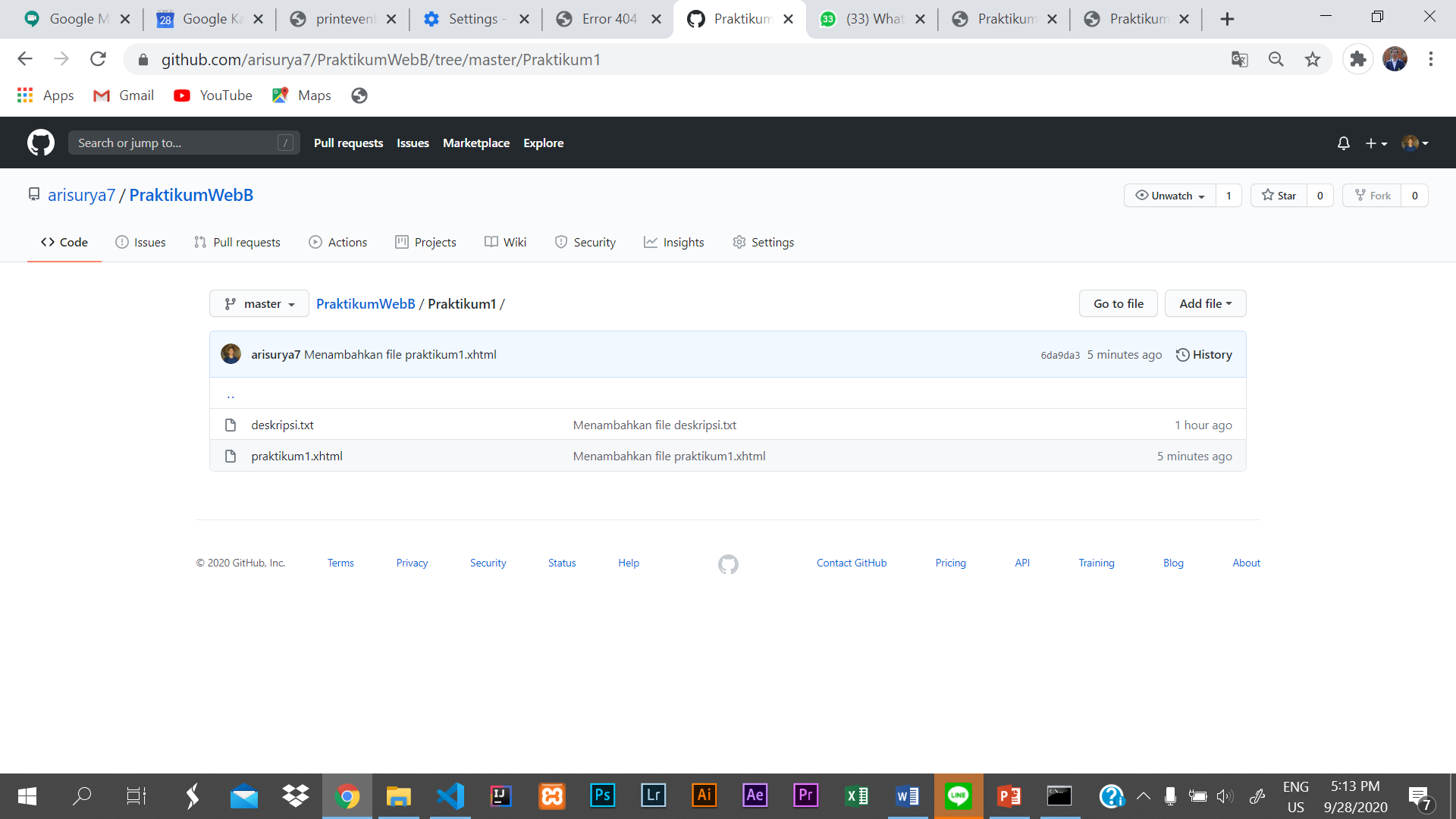


* Kemudian kita akan mengirim/mengupload file *praktikum1.xhtml* yang ada di folder komputer kita ke *Github* dengan menggunakan perintah *git push –u origin master* . Ketika nanti kita ada mengirimkan file baru ke github dari lokal komputer, kita cukup kita gunakan perintah *git push*. Hal ini dikarenakan ketika pertama kali kita melakukan push sudah dilakukan inisialisasi *remote* yang dituju dengan *–u origin master*.

Biasanya ketika kita hendak melakukan *push*, tidak jarang juga akan diminta untuk memasukan *username* dan *password* dari *Github* kita. Kebetulan karena sebelumnya saya sudah pernah menginputkan *username* dan *password* jadi tidak ditampilkan lagi untuk melakukan *action* serupa.



* Setelah selesai melakukan push kita bisa melihat hasilkan di *repository* *Github* yang sudah kita buat. File *praktikum1.xhtml* berhasil diupload ke repository PraktikumWebB dalam folder Praktikum1.



* Lakukan langkah yang sama dengan file laporan 1808561026\_I Kadek Ari Surya\_pertemuan1.docx yang akan di upload juga.

# **BAB IV**

**PENUTUP**

## **4.1. Simpulan**

4.1.1Antara XHTML dengan HTML memiliki kesamaan yang tidak jauh berbeda. Karena memang XHTML adalah bagian dari HTML. XHTML merupakan kombinasi antara HTML dan XML yang telah direkomendasikan oleh W3C sejak 2003 sebagai standar dalam penulisan website. Oleh karena itu perintah dasar yang ada di HTML dengan XHTML tidak jauh berbeda.

4.1.2*Direktory* pada Github dikenali sebagai *repository*. Dalam membuat *repository* pada *Github* tentu kita mempunyai akun Github terlebih dahulu. Kemudian kita bisa menentukan nama, memilih lisensi hingga membuat deskripsi untuk penjelasan lebih detail. Setelah *repository* berhasil dibuat maka kita bisa membuat/menambahkan file atau folder baru didalamnya.

4.1.3Untuk melakukan *upload file* dari lokal komputer kita ke akun *Github* ada serangkaian proses yang harus dilakukan. Pertama kita harus menginstal *Git* pada komputer dilanjutkan dengan melakukan *setting* perintah pada terminal bash *Git*. Kita harus menginisialisasi lokasi file yang akan kita upload telebih dahulu. Setelah itu kita akan membawa file tersebut ke *Staging Area* agar perubahan bisa dikenali hingga dilakukan *commit* untuk menyimpan perubahan. Selanjutnya kita membuat *remote* ke tautan *repository Github* yang akan dituju. Terakhir kita lakukan *push* terhadap file tersebut untuk mengupload. (Ardhianto, 2019)

## **4.2. Saran**

4.2.1Ketika kita membuat file .xhtml pastikan jika struktur kode program yang dibuat sudah benar dan lengkap. Hal ini karena kesalahan kecil akan berpengaruh pada running program yang gagal.

4.1.2Dalam melakukan *upload* file ke *Github* pastikan jika sudah dibuat *repository* pada lokal komputer dan *remote* yang dibuat sudah merujuk pada alamat *repository Github* yang akan dikirimkan.

# **DAFTAR PUSTAKA**

Daru Setiawan, dkk, 2011. *Analisis Dan Perbandingan Xhtml (Extensible Hypertext Markup Language) 1.0 Strict Dan Xhtml 1.0 Transitional ( Studi Kasus Web Toko Buku [Web 2.0] )*. Universitas Telkom.

Edison S., 2019. Implementasi Git Dan Github Untuk Membangun Aplikasi Menggunakan Android Studio.*Jurnal Pradita*, I(1), pp. 26-36.