# Modul Workshop Pembuatan Portofolio Berbasis Web dengan Hosting Menggunakan GitHub di SMK Al-Amanah Tangerang Selatan



**Dibuat Oleh:**

**Andri Firman Saputra**

Daftar Isi

[Modul Workshop Pembuatan Portofolio Berbasis Web dengan Hosting Menggunakan GitHub di SMK Al-Amanah Tangerang Selatan 1](#_Toc114362832)

[A. Uraian Materi 3](#_Toc114362833)

[1. Git 3](#_Toc114362834)

[1) Definisi Git 3](#_Toc114362835)

[2) Perintah Dasar Git 3](#_Toc114362836)

[3) Cara kerja Git 4](#_Toc114362837)

[2. GitHub 5](#_Toc114362838)

[1) Fungsi GitHub 5](#_Toc114362839)

[3. HTML 5](#_Toc114362840)

[1) Definisi HTML 5](#_Toc114362841)

[2) Fungsi HTML 5](#_Toc114362842)

[3) Dasar tag-tag HTML 6](#_Toc114362843)

[4. CSS 6](#_Toc114362848)

[1) Definisi CSS 6](#_Toc114362849)

[2) Fungsi CSS 6](#_Toc114362850)

[3) Jenis-jenis CSS 6](#_Toc114362851)

[5. Figma 7](#_Toc114362854)

[1) Fungsi Figma: 7](#_Toc114362855)

[B. Langkah-langkah pembuatan portofolio berbasis website 8](#_Toc114362856)

# Uraian Materi

Sebelum kita membuat portofolio berbasis website, sebaiknya kita memahami terlebih dahulu materi yang akan kita gunakan.

## Git

### Definisi Git

Git adalah salah satu **sistem pengontrol versi** (Version Control System) pada project perangkat lunak yang diciptakan oleh Linus Torvalds.

Version Control System (VCS) adalah sistem yang **mengelola perubahan** dari sebuah dokumen, website, dan jenis file lainnya.

Git dikenal juga dengan **distributed revision control** (VCS terdistribusi), artinya **penyimpanan** Git tidak hanya berada dalam **satu tempat** saja. Bisa melalui **cloud storage** yang mempunyai **kemampuan VCS**. Contohnya adalah **GitHub.**

**Saat pertama kali** **menggunakan** Git, pada folder yang ingin kita **pasang** **Git**, kita harus **mengetikan** perintah **git init** di dalam folder, folder yang **memiliki Git** di dalamnya disebut dengan **repository.**

### Perintah Dasar Git

Untuk dapat mengetahui bagaimana cara menggunakan Git, berikut ini beberapa perintah dasar dari Git:

* **git init** : untuk membuat sebuah **repository** yang berada pada file **lokal.**
* **git status** : untuk mengetahui **status** dari sebuah **repository lokal.**
* **git add** : untuk **menambahkan file** **baru** pada **repository** yang **telah dipilih.**
* **git commit** : untuk **menyimpan perubahan** yang telah dilakukan.
* **git push** : digunakan untuk **mengirimkan perubahan** setelah di **commit** ke **remote repository.**
* **git branch** : untuk melihat seluruh branch yang ada dalam **repository.**
* **git checkout** : untuk **menukar branch** **yang aktif** dengan **branch** yang telah **dipilih** selain itu, dapat digunakan untuk **berpindah commit.**
* **git merge** : untuk menggabungkan **branch yang aktif** dan yang **telah dipilih.**
* **git clone** : untuk membuat **salinan** **repository cloud** ke **repository lokal.**
* **git remote** : untuk me-remot **repository cloud** yang biasanya **disimpan** di **server lain**.

### Cara kerja Git

Git akan **memantau semua perubahan yang terjadi** pada file project. Lalu **menyimpannya** ke dalam **commit**, **commit** adalah **perubahan** yang dilakukan oleh seorang individu pada **repository** dan **menyimpannya** dengan kode **hash** sebagai **penanda commit** tersebut. **Penanda commit** berbentuk **hash (acak)** yang digunakan jika seorang user ingin **berpindah-pindah** commit.

Contoh kasus dalam skripsi:



Gambar **lingkaran biru** adalah **commit**, teks di sebelah kanan adalah **message** atau **pesan commit** tersebut. **Kode** yang disebelah **kiri** lingkaran biru adalah **penanda commit**. Pengguna bebas **berpindah-pindah commit** dengan menggunakan **penanda commit**.

Perintah yang digunakan untuk **berpindah** commit:

**$ git checkout** ‘6-digit penanda commit’

Contoh saya ingin **kembali** ke commit **perbaiki analisis**, maka:

**$ git checkout** tu5e...

Jika saya berpindah commit dari **menghapus teori yang tidak relevan** **(commit terakhir)** ke commit **memperbaiki analisis (commit sebelumnya)** maka, perubahan file pada commit yang **menghapus teori yang tidak relevan (commit terakhir)** hilang.

## GitHub

GitHub adalah **layanan host web bersama** untuk **project** pengembangan perangkat lunak yang menggunakan **sistem kendali versi** **Git** dan **layanan hosting internet**. (sumber: wikipedia.org)

GitHub adalah **situs web** juga sebuah **layanan cloud** yang bisa membantu para pengguna untuk **menyimpan**, **mengelola** dan **mengembangkan project**. Di dalam GitHub kita bisa **mengupload file**, **membuat file** yang mana filenya bisa kita **kelola** dengan **version control system**. Github juga memiliki versi **desktop** dengan nama **GitHub Desktop.**

### Fungsi GitHub

* Memudahkan **kolaborasi** pengerjaan **project**
* Mencegah **perubahan kode** yang bisa **merusak** kode asli
* Sebagai **portofolio** bagi developer

## HTML

### Definisi HTML

HTML (Hypertext Markup Language) adalah **bahasa markup** yang digunakan untuk membuat **halaman website**. Isinya terdiri dari berbagai **kode** yang dapat **menyusun struktur** suatu website.

HTML terdiri dari kombinasi **teks dan simbol** yang disimpan dalam sebuah file. Dalam membuat file HTML, terdapat **standar** atau format khusus yang harus diikuti. Format tersebut telah tertuang dalam **standar kode internasional** atau **ASCII** (American Standard Code for Information Interchange).

### Fungsi HTML

* Membuat halaman web.
* Menampilkan berbagai informasi di dalam sebuah browser Internet.
* Membuat link menuju halaman web lain dengan kode tertentu (Hypertext).

### Dasar tag-tag HTML

* **<html>** Untuk memulai membuat halaman HTML yang **mencakup semua** konten dan elemen
* **<head>** Untuk membuat halaman HTML yang mencakup tampilan deskripsi di hasil pencarian (meta), style konten (CSS), dan lainnya
* **<title>** Untuk membuat **judul** website
* **<body>** Untuk membuat **bagian isi** website
* **<h1>** sampai **<h6>** Untuk membuat **heading**
* **<p>** Untuk membuat **paragraph**



## CSS

### Definisi CSS

CSS adalah singkatan dari **cascading style sheets**, yaitu bahasa yang digunakan untuk menentukan **tampilan** dan format halaman website. Dengan CSS, Anda bisa mengatur jenis font, warna tulisan, dan latar belakang halaman.

### Fungsi CSS

* Menawarkan lebih banyak variasi tampilan
* Membuat website tampilan rapi di semua ukuran layar

### Jenis-jenis CSS

* Inline CSS : CSS yang dituliskan di dalam tag HTML dengan **atribut style=””**
* Internal CSS : CSS yang dituliskan di bagian header file HTML dengan **tag** **<style>**
* External CSS : CSS yang dituliskan di luar file HTML dengan nama file diakhir ‘.css’. Contoh: ‘nama\_file.css’, kemudian file ‘.css’ dipanggil pada file HTML dengan tag **<link rel=”stylesheet” href=”nama\_file.css”>** di dalam tag **<head>** HTML.



## Figma

**Tool desain** berupa **website** yang terhubung dengan **cloud computing** sehingga bisa digunakan **kapanpun** dan **dimanapun** melalui **internet**. Tool ini berbasis **vector**, sehingga akan lebih cocok untuk mendesain **UI website** atau **mobile** dan aset **ilustrasi**. Tetapi, Figma juga bisa digunakan untuk **mengedit foto** namun hanya dengan pengaturan **dasar** saja.

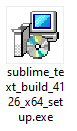
### Fungsi Figma:

* Untuk melakukan **prototyping** website baik di sisi **desktop** maupun **mobile**
* Membuat **tampilan UI** dan **wireframe** aplikasi mobile
* Desain untuk postingan di **media sosial** seperti Facebook, Twitter, Instagram, Dribble, dan LinkedIn
* Pembuatan **mockup desain** pada perangkat yang tersedia di template
* Pengeditan **gambar dasar**
* Membuat desain menjadi **video** (menggunakan plugin)
* Mendesain dalam **satu waktu** dan **realtime** bersama **tim** secara **online**

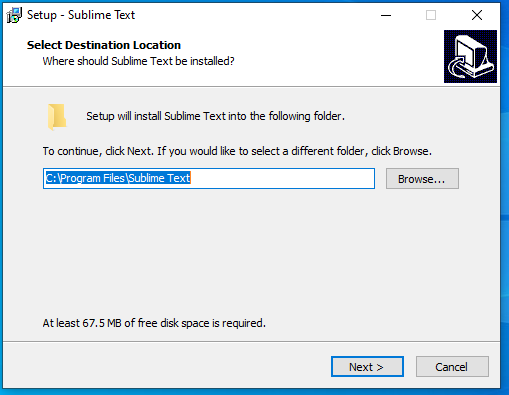
# Langkah-langkah pembuatan portofolio berbasis website

## Menginstall Sublime Text

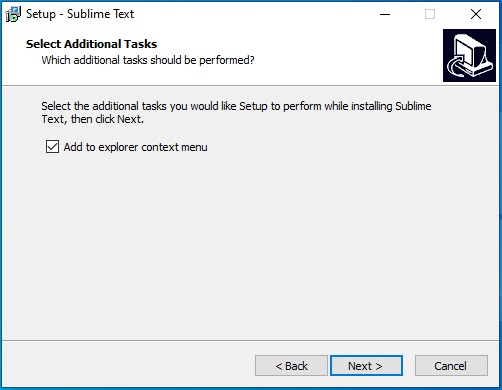
1. Buka software **sublime\_text\_build\_4126\_x64\_setup.exe**



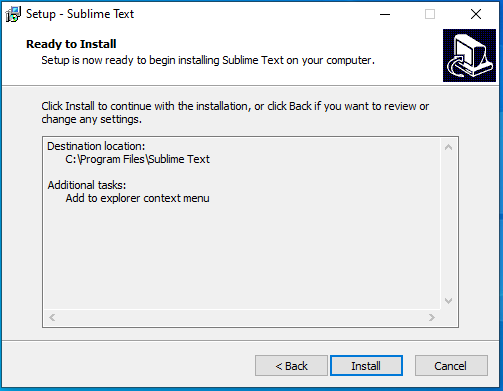
1. Pilih **lokasi** instalasi sublime text. Lalu, klik tombol **next**



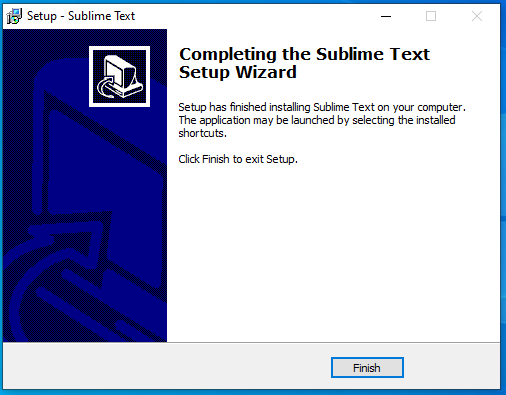
1. Centang kotak **add to explorer context menu**, supaya bisa di cari pada windows search. Kemudian, klik tombol **next**



1. Klik tombol **install**

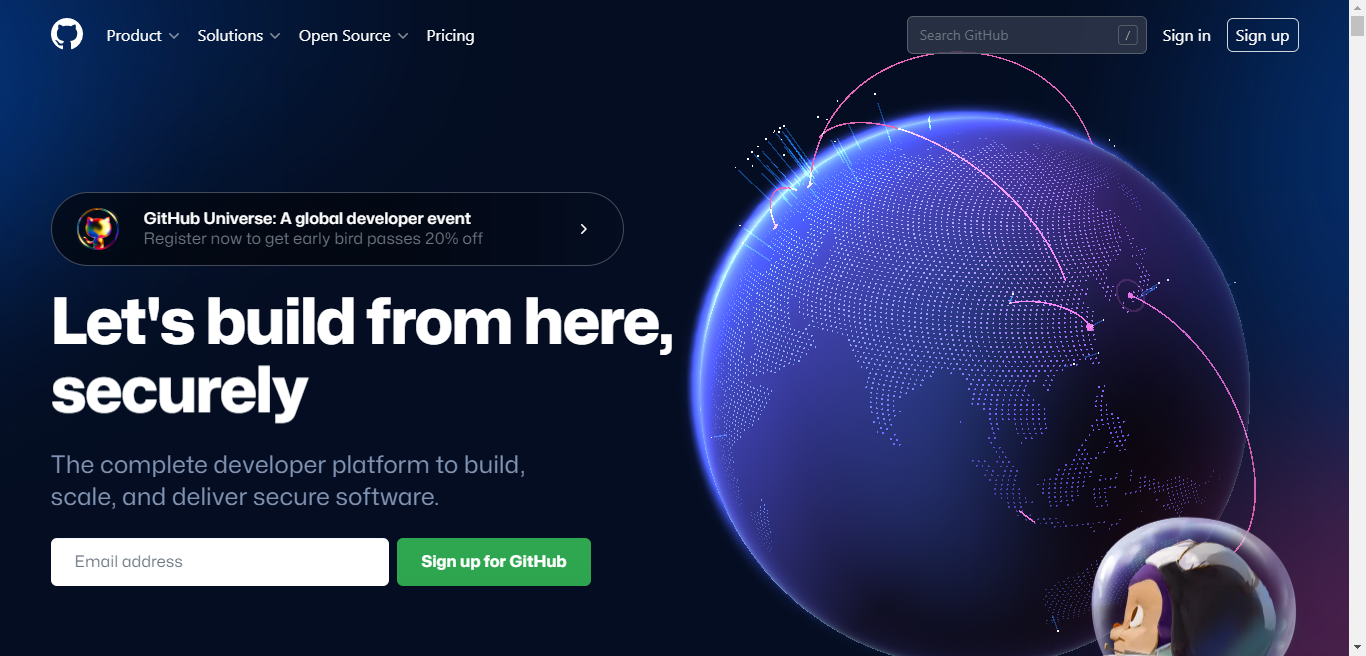


1. Jika instalasi **berhasil**, maka akan menampilkan dialog seperti ini. Kemudian, klik tombol **finish**

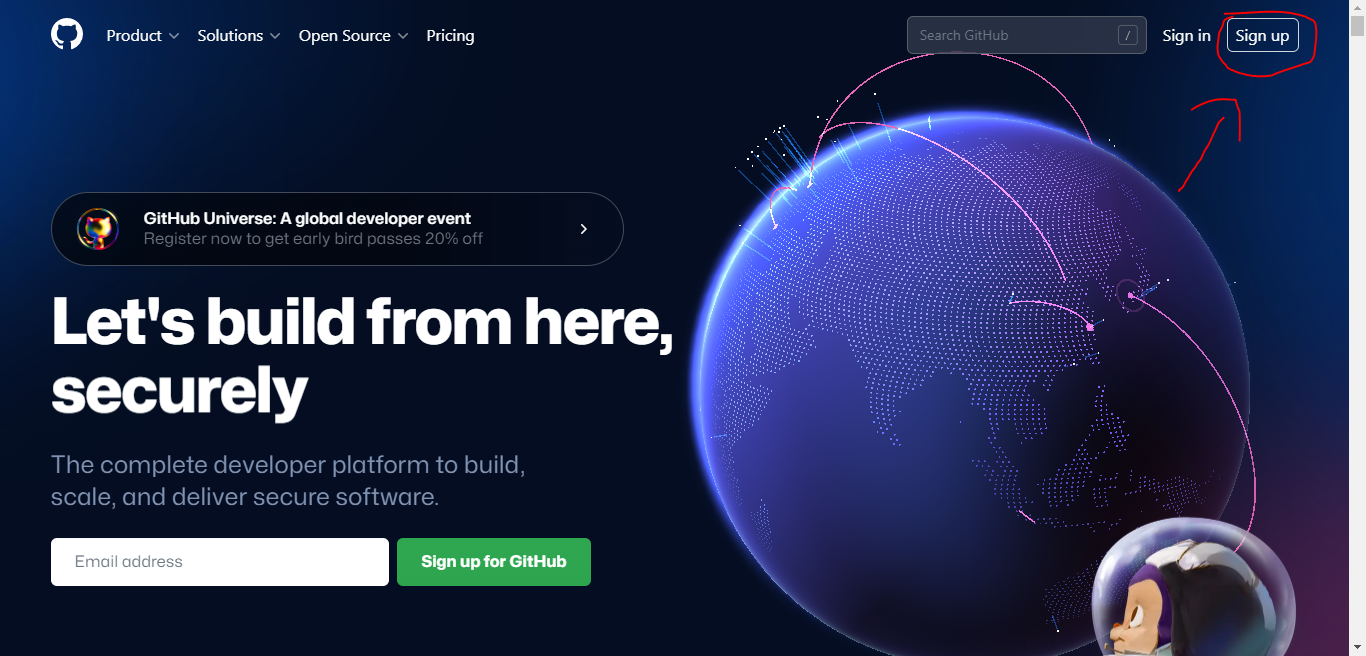


## Membuat akun GitHub

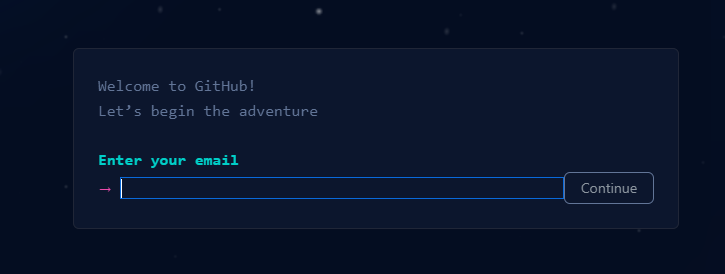
1. Buka website <https://github.com>, berikut adalah tampilan awal website



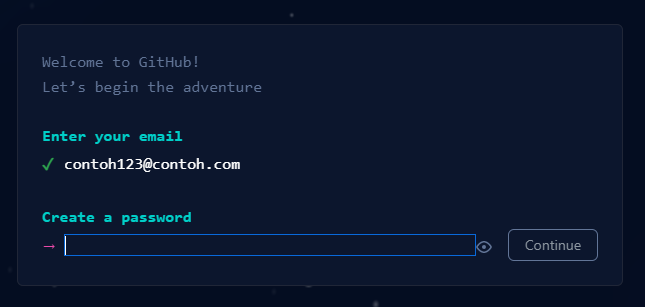
1. Klik tombol **Sign Up**, di kanan atas pada bagian nav bar



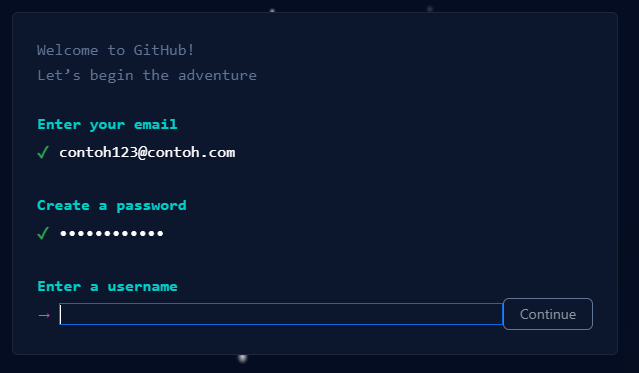
1. Masukkan **email** yang **aktif**. Kemudian, klik tombol **continue**



1. Masukkan **password baru** untuk akun **GitHub**. Lalu, klik tombol **continue**



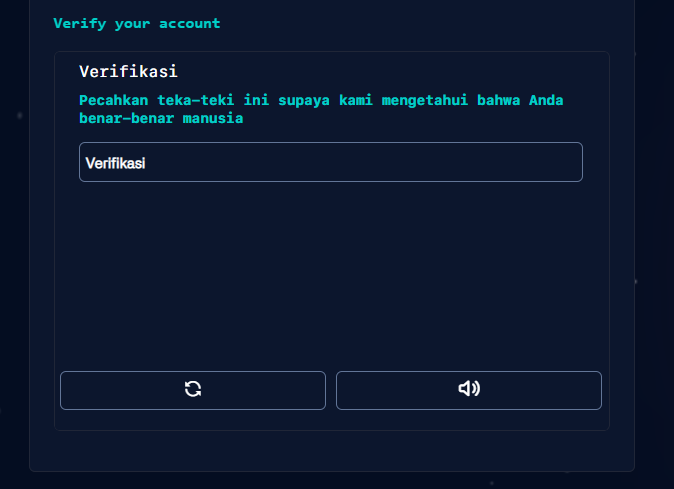
1. Masukkan **username baru** yang mudah **diingat**. Kemudian, klik tombol **continue**



1. Untuk mendapatkan **informasi terupdate** tentang **GitHub** ketik **y** yang nantinya dikirim lewat email, jika **tidak** ingin ketik **n**. Lalu, klik tombol **continue**



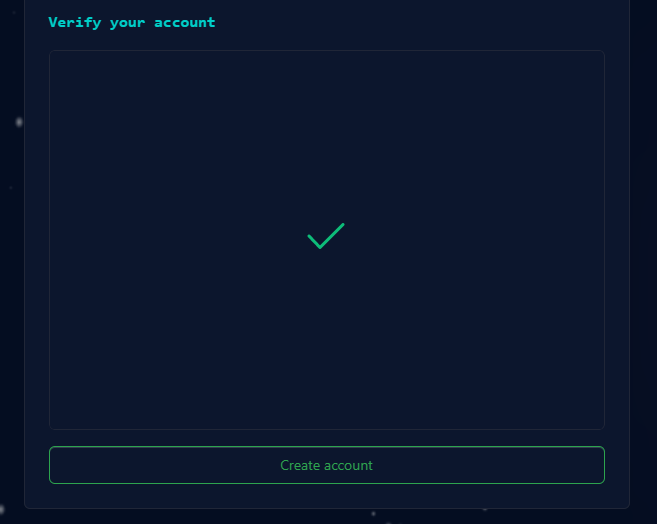
1. Untuk **verifikasi akun**, klik tombol **verifikasi**



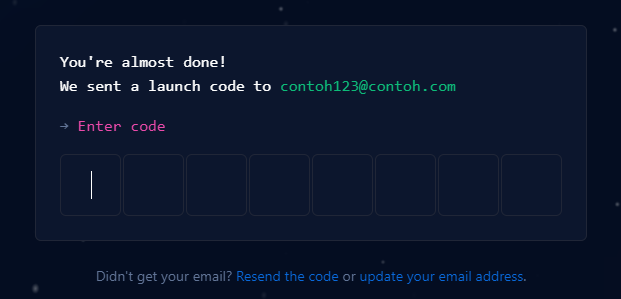
1. Kemudian, pilih galaxy yang **spiral**, sebanyak **2 kali** secara **bergantian**



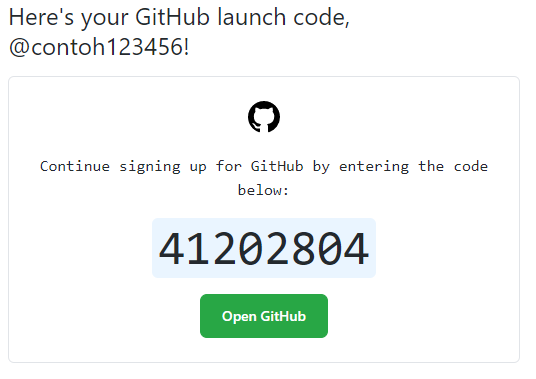
1. Jika sudah **berhasil**, klik tombol **create account**



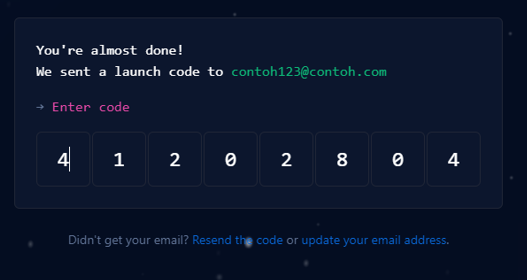
1. kemudian masukkan **kode** **verifikasi GitHub**, yang didapatkan dari **email**



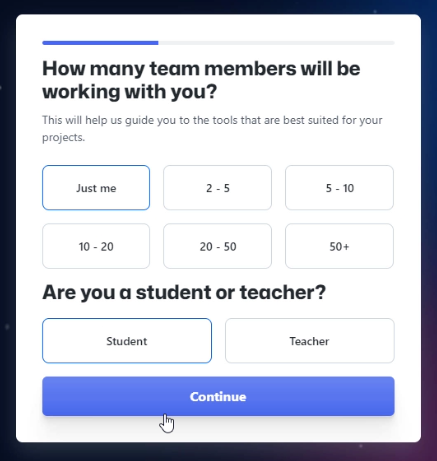
1. Berikut adalah tampilan **email**, yang didapatkan dari GitHub.



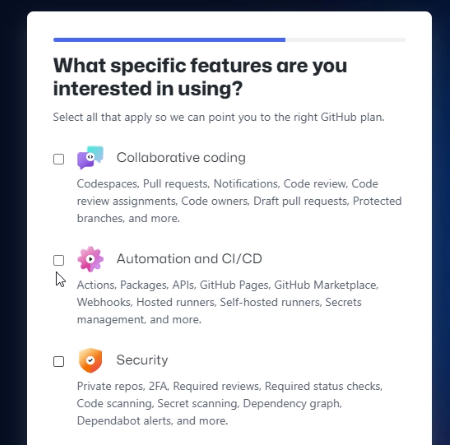
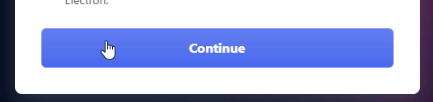
1. Masukkan **6-digit kode** tersebut ke **website GitHub**



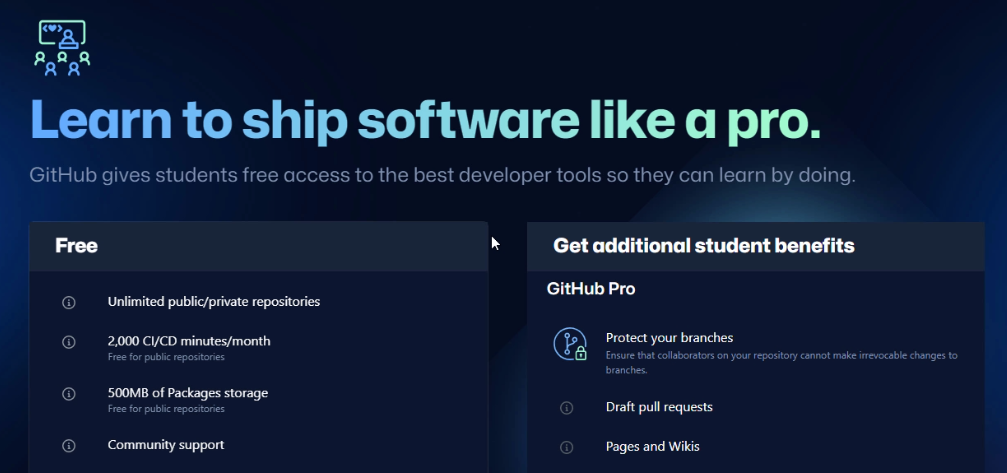
1. Isi **survey** sesuai dengan **kondisi** kamu. Lalu, klik tombol **continue**

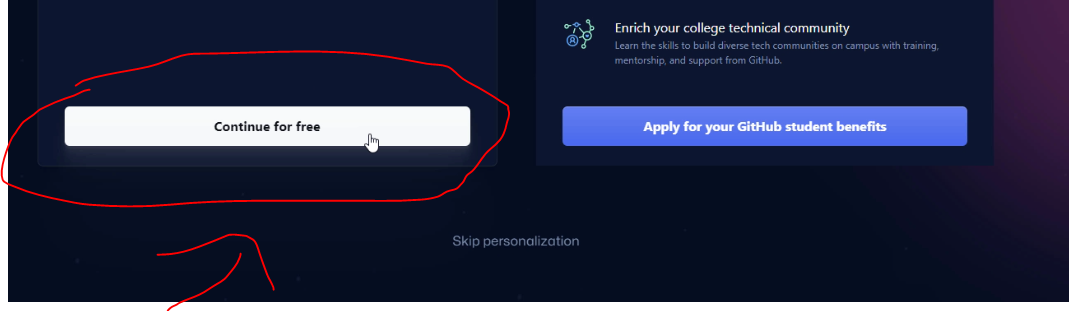


1. Pilih apa yang kamu **gunakan** **fitur** di **GitHub** ini. **Tidak perlu** memilih, langsung saja **scroll ke paling bawah** kemudian, klik tombol **continue**

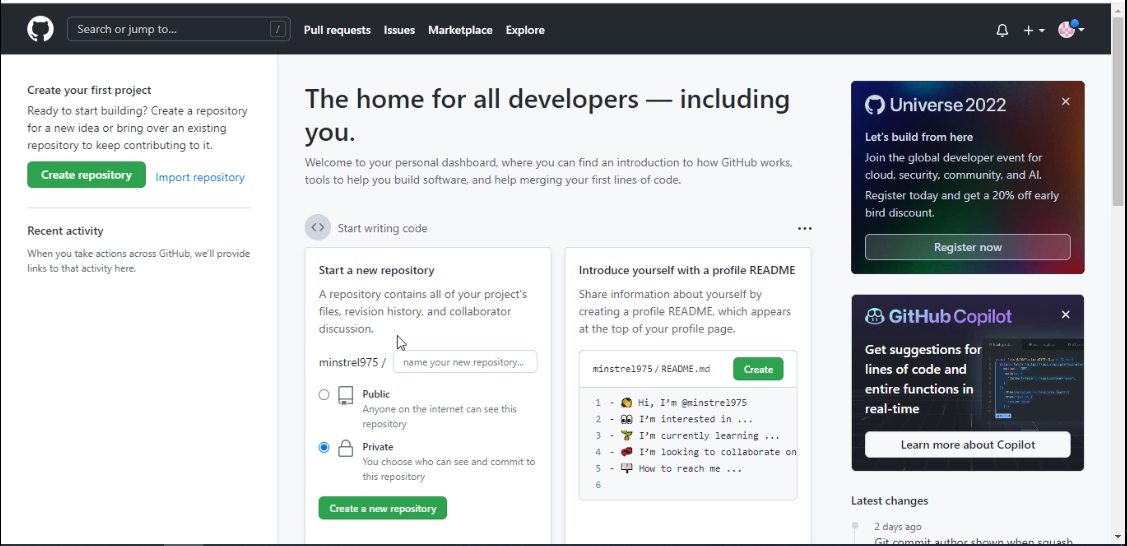
 

1. Pada halaman ini, langsung saja **scroll ke paling bawah**. Kemudian klik tombol **continue for free**





1. Selamat **akun** kamu **sudah siap digunakan**

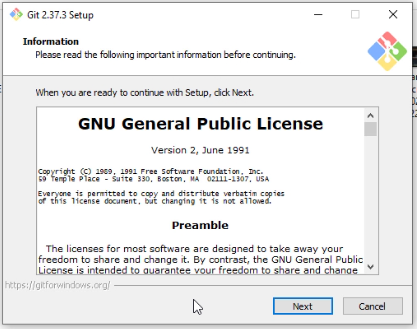


## Menginstall Git

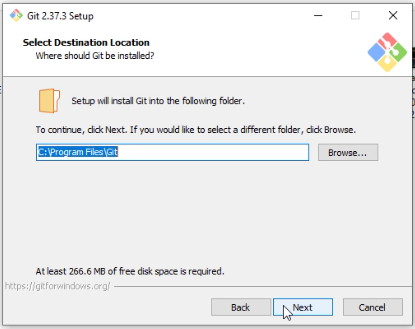
1. Buka software **Git-2.37.3-64-bit.exe**



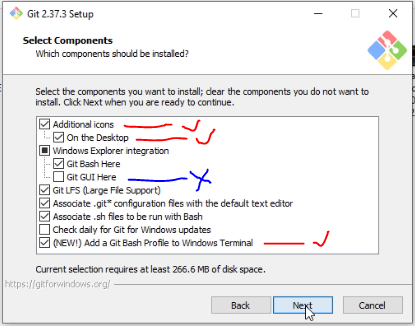
1. Setelah membaca **lisensi**, lalu klik tombol **next**



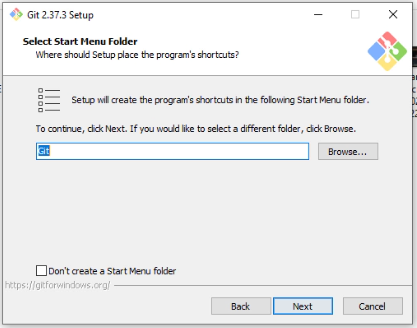
1. Pilih **lokasi** instalasi Git, kemudian klik tombol **next**



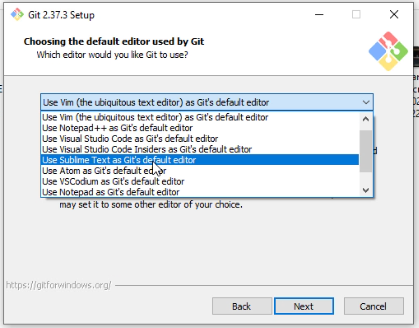
1. Centang **Additional Icons**, **On the Desktop**, dan **Add a Git Bash Profile.  
   Hapus** centang **Git GUI Here**, karena kita **tidak menggunakannya.** Kemudian, klik tombol **next**



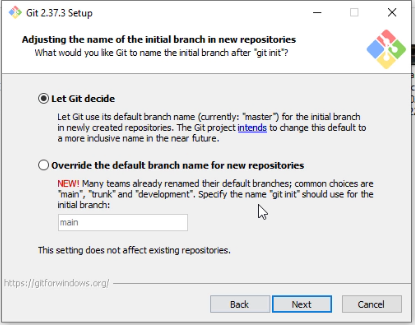
1. Klik tombol **next**



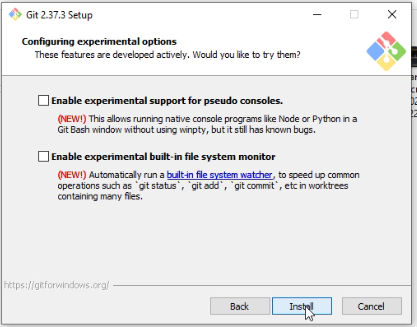
1. Klik **dropdown** **Use Vim**, ganti menjadi **use Sublime Text**, pastikan sublime textnya **sudah terinstal** terlebih dahulu. Kemudian, klik tombol **next**



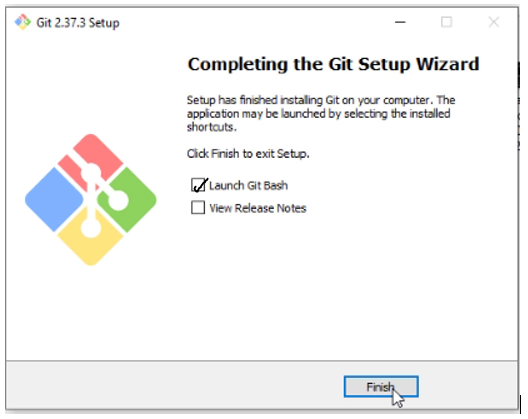
1. Jika ingin mengganti nama **branch** utama, pilih **override**, jika **tidak** pilih **let git decide**. Kemudian, klik tombol **next**



1. **Next saja terus** sampai menemukan tombol **install**, kemudian klik tombol **install**



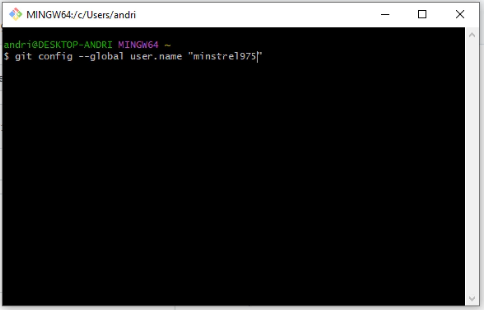
1. **Tunggu** proses instalasi
2. Kemudian, **centang** launch Git Bash. Lalu, **hapus** centang **view release notes**. Lalu, klik tombol **finish**



## Meng-konfigurasi Git

1. Mengatur **username** pada **Git** dengan cara ketik:

$ git config –-global user.name “username-github”



1. Mengatur **Email** pada **Git** dengan cara ketik:

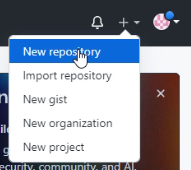
$ git config –-global user.email “email-github”



Jika kalian **lupa** email GitHub-nya, pada GitHub kalian ke **settings** > pada public email klik **email settings**.

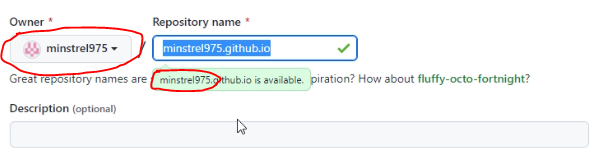
## Membuat Repository pada GitHub

1. Pada **halaman utama** GitHub, klik tombol **tambah** di **nav bar** bagian **kanan atas**. Kemudian, klik tombol **new repository**

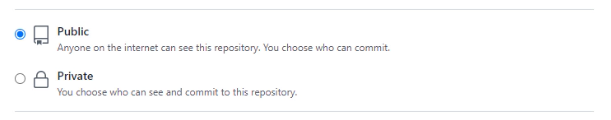


1. Untuk membuat **GitHub Pages**, **buat baru repository** dengan **nama**:

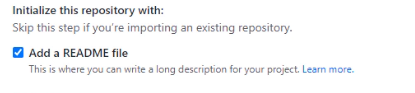
**username**.github.io, lihat seperti gambar di bawah ini



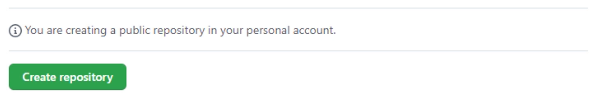
1. Untuk repository-nya wajib **public**



1. Kemudian, **Centang kotak Add a README file**. Tujuannya agar kita **tidak** perlu **meng-inisialisasi** pada **lokal**

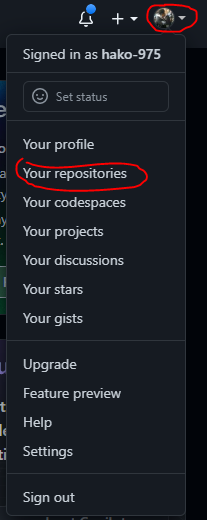


1. Lalu, klik tombol **create repository**



## Meng-clone Repository ke Lokal

1. Pada GitHub, pilih **repository** yang ingin **di-clone**. Untuk **mencari** repository, klik **logo profile** kemudian, klik tombol **Your repositories**



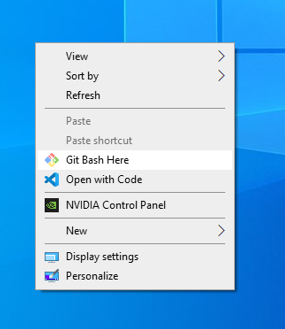
1. Lalu, untuk meng-clone **repository** **cloud** di **GitHub**, klik tombol **code**



1. Kemudian **salin URL** **HTTPS**



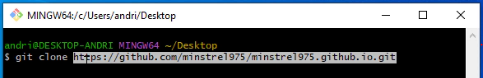
1. Kembali ke **Desktop**, kemudian klik **kanan** Git Bash Here



1. Kemudian untuk **meng-clone repository cloud**, ketik:

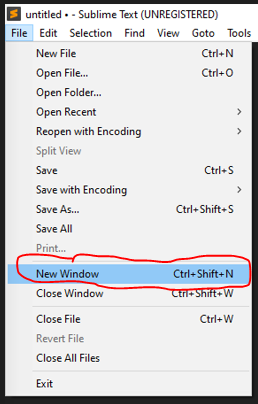
$ git clone **paste URL yang telah di copy**

Untuk **paste** di Git, klik **kanan** paste, atau tekan **shift + insert** setelah itu, tekan enter



## Membuka Project Repository dengan Sublime Text

1. Buka **Sublime Text**
2. Klik **file** > **new window**



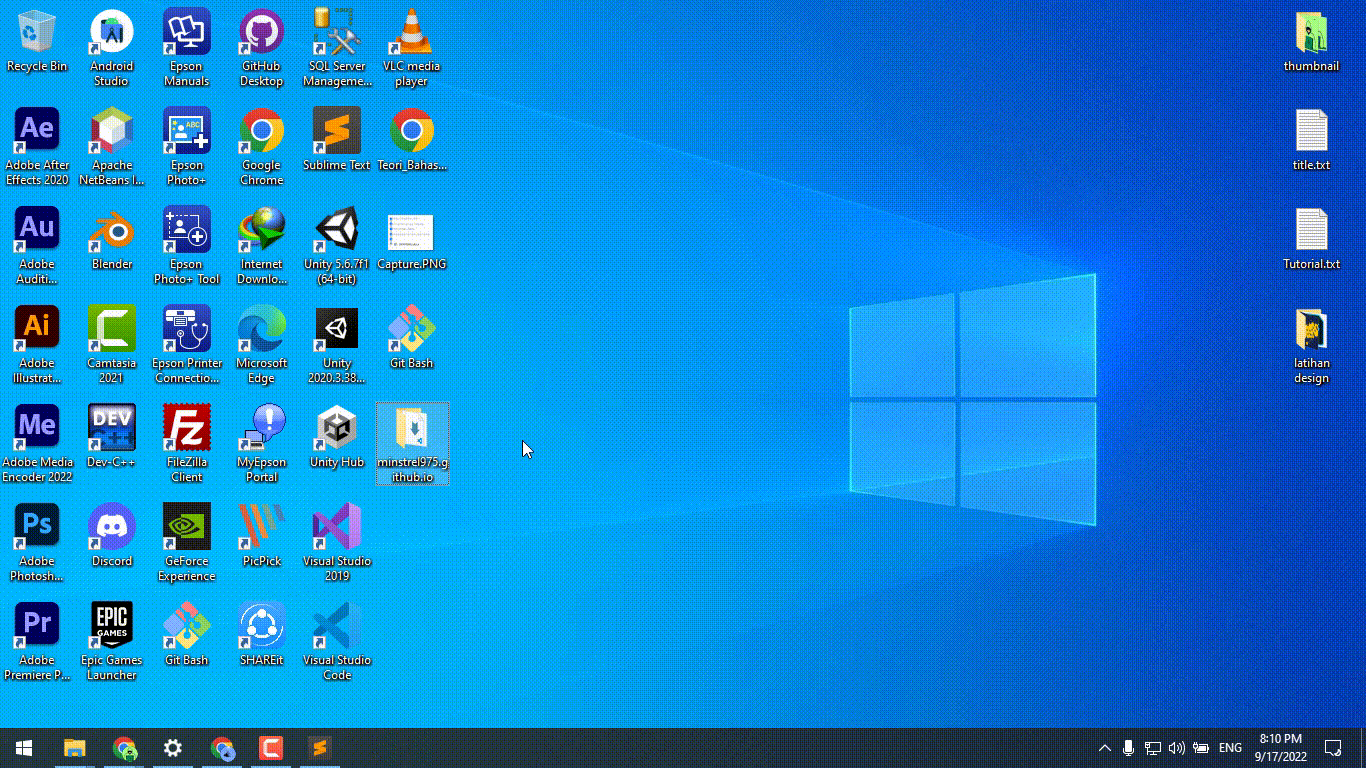
1. **Minimize** window



1. **Close** window



1. **Drag** folder project ke **Sublime Text**



## Membuat file .html

## Membuat file .css

## Memposting project Repository ke GitHub

## Mereviews project portofolio

## Mengupdate perubahan pada project portofolio di lokal ke cloud