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



**Dibuat Oleh:**

**Andri Firman Saputra**

Daftar Isi

[A. Uraian Materi 4](#_Toc121388995)

[1. Git 4](#_Toc121388996)

[1) Definisi Git 4](#_Toc121388997)

[2) Perintah Dasar Git 4](#_Toc121388998)

[3) Cara kerja Git 5](#_Toc121388999)

[2. GitHub 6](#_Toc121389000)

[1) Fungsi GitHub 6](#_Toc121389001)

[3. HTML 6](#_Toc121389002)

[1) Definisi HTML 6](#_Toc121389003)

[2) Fungsi HTML 6](#_Toc121389004)

[3) Dasar tag-tag HTML 7](#_Toc121389005)

[4. CSS 7](#_Toc121389010)

[1) Definisi CSS 7](#_Toc121389011)

[2) Fungsi CSS 7](#_Toc121389012)

[3) Jenis-jenis CSS 7](#_Toc121389013)

[5. Figma 8](#_Toc121389016)

[1) Fungsi Figma 8](#_Toc121389017)

[B. Langkah-langkah pembuatan portofolio berbasis website 9](#_Toc121389018)

[1. Menginstall Sublime Text 9](#_Toc121389019)

[2. Membuat akun GitHub 11](#_Toc121389020)

[3. Melihat Mockup Figma 18](#_Toc121389021)

[4. Membuat Portofolio 19](#_Toc121389022)

[1) Menyiapkan project 19](#_Toc121389023)

[2) Buat file index.html 19](#_Toc121389024)

[3) Buat file style.css 19](#_Toc121389025)

[4) Membuat portofolio 19](#_Toc121389026)

[5. Membuat Repository pada GitHub 29](#_Toc121389027)

[6. Memposting Portofolio ke GitHub 30](#_Toc121389028)

[7. Mereviews project portofolio 32](#_Toc121389029)

[8. Mengupdate perubahan pada portofolio 33](#_Toc121389030)

# 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-tag HTML dengan **atribut style=””**
* **Internal CSS** : CSS yang dituliskan di dalam tag **<head>** untuk menggunakan CSS-nya dengan di dalam **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.

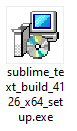
### Fungsi Figma

* 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**

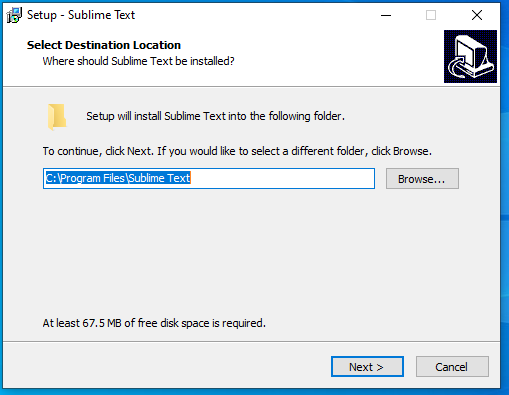
# Langkah-langkah pembuatan portofolio berbasis website

## Menginstall Sublime Text

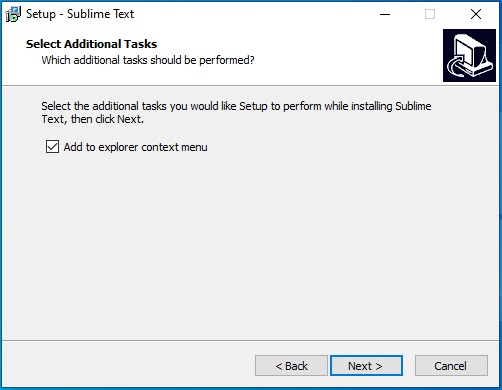
1. Buka software **sublime\_text\_build\_4126\_x64\_setup.exe**



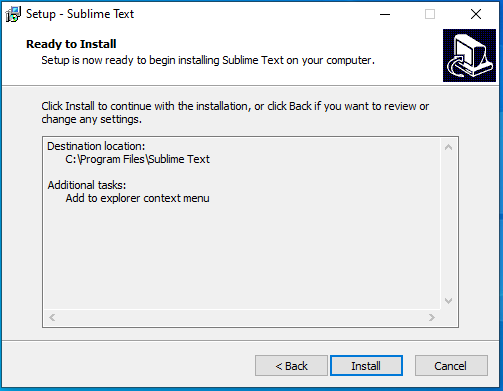
1. Pilih **lokasi** instalasi sublime text. Lalu, klik tombol **next**



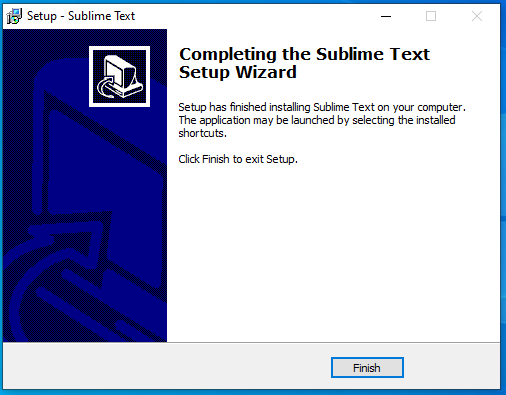
1. Centang kotak **add to explorer context menu**, supaya bisa di cari pada windows search. Kemudian, klik tombol **next**



1. Klik tombol **install**

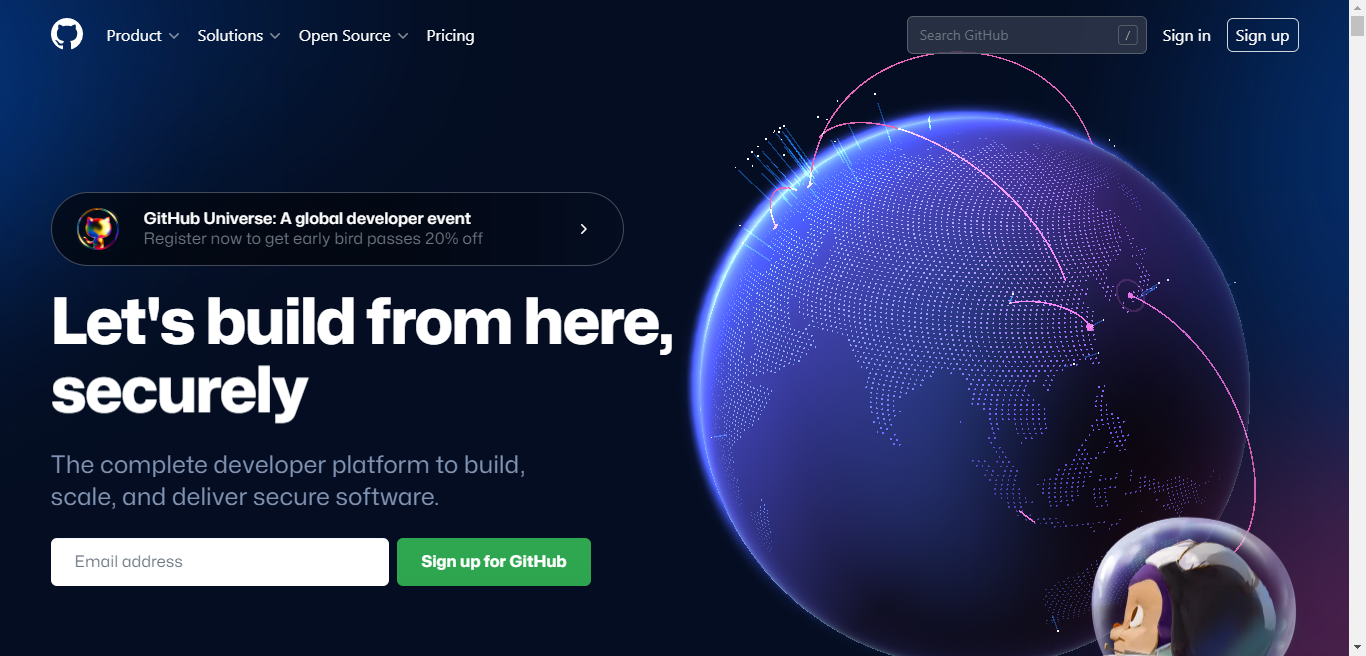


1. Jika instalasi **berhasil**, maka akan menampilkan dialog seperti ini. Kemudian, klik tombol **finish**

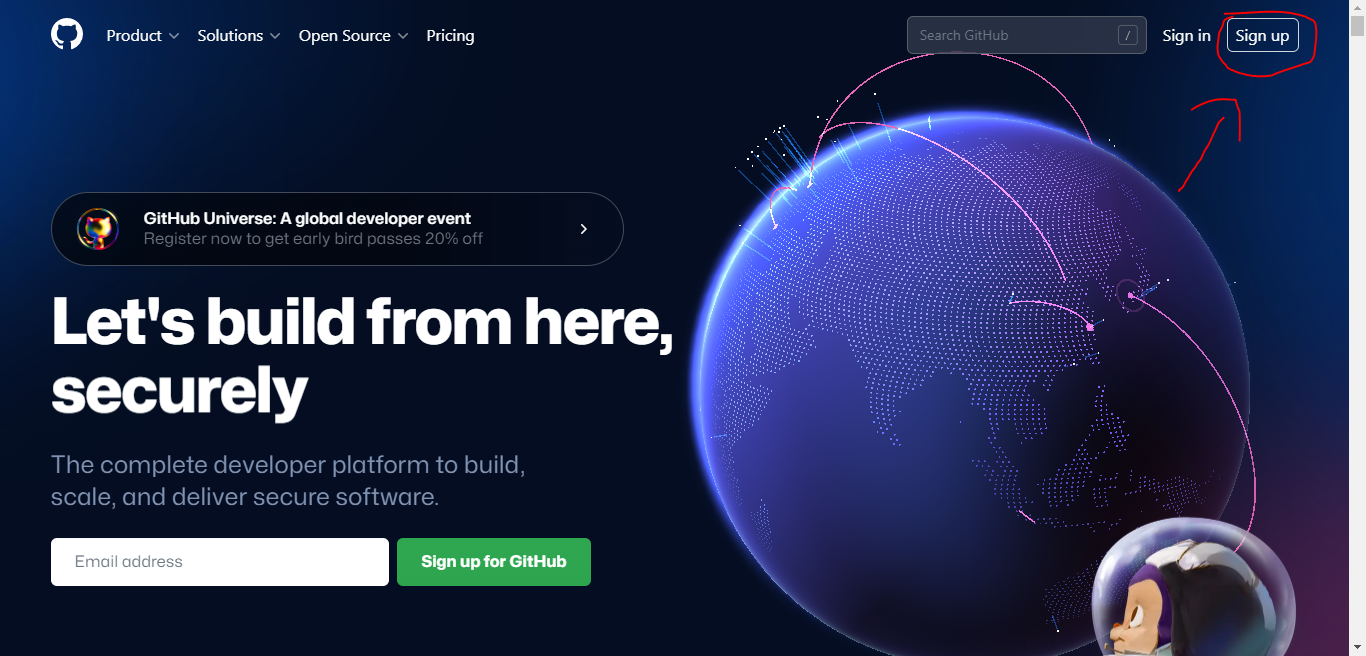


## Membuat akun GitHub

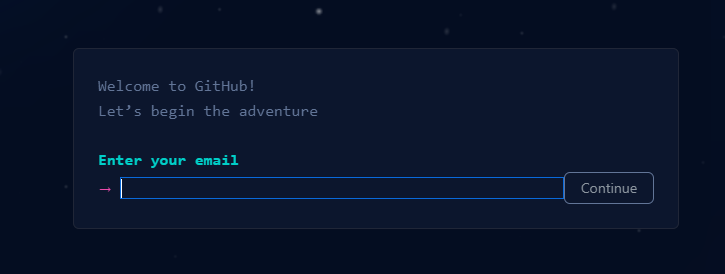
1. Buka website <https://github.com>, berikut adalah tampilan awal website



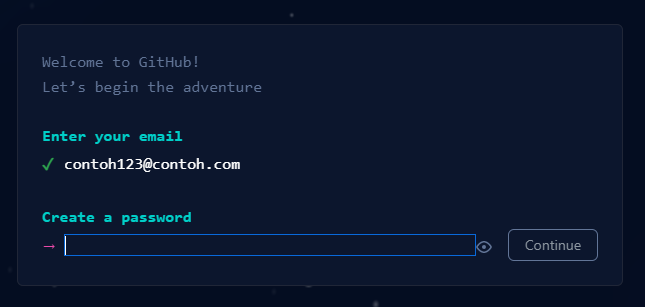
1. Klik tombol **Sign Up**, di kanan atas pada bagian nav bar



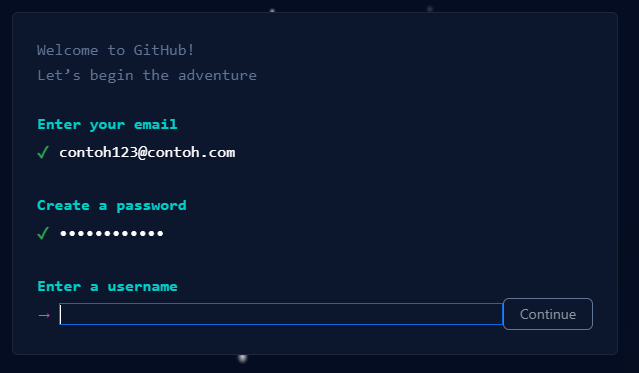
1. Masukkan **email** yang **aktif**. Kemudian, klik tombol **continue**



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



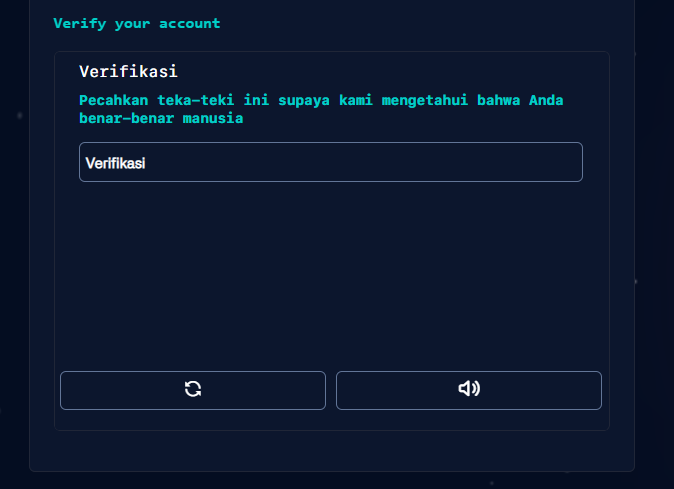
1. Masukkan **username baru** yang mudah **diingat**. Kemudian, klik tombol **continue**



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



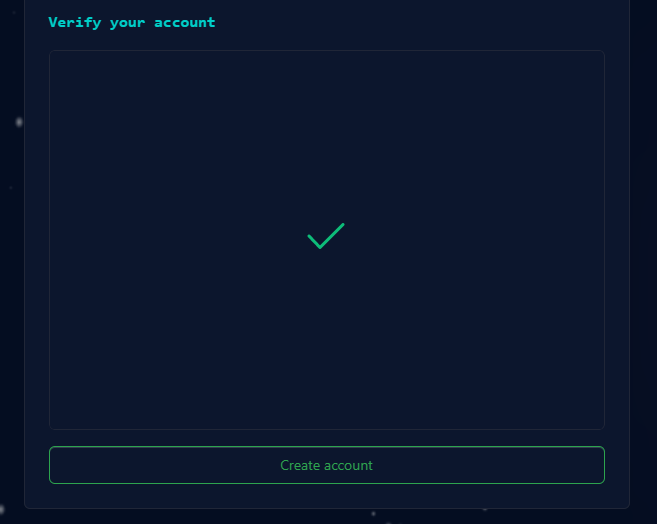
1. Untuk **verifikasi akun**, klik tombol **verifikasi**



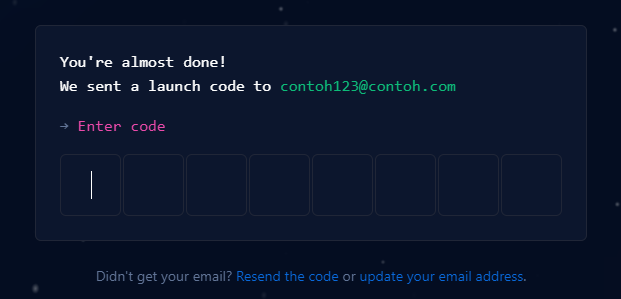
1. Kemudian, pilih galaxy yang **spiral**, sebanyak **2 kali** secara **bergantian**



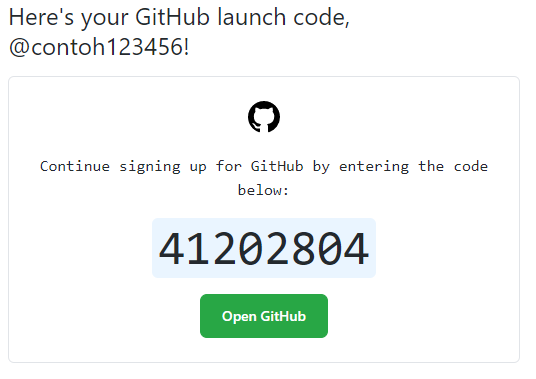
1. Jika sudah **berhasil**, klik tombol **create account**



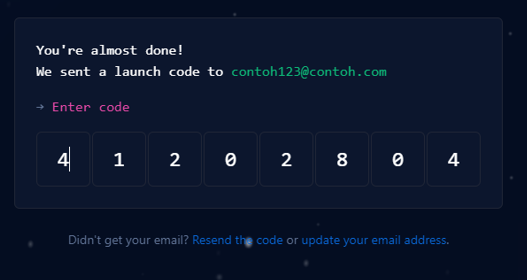
1. kemudian masukkan **kode** **verifikasi GitHub**, yang didapatkan dari **email**



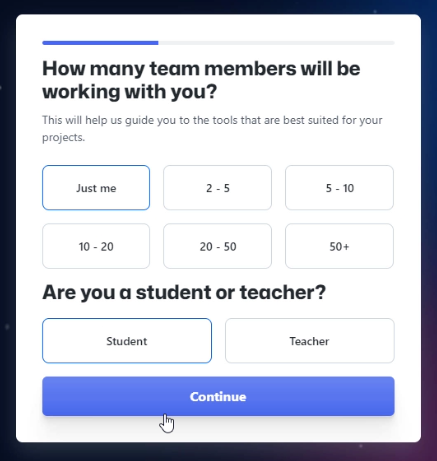
1. Berikut adalah tampilan **email**, yang didapatkan dari GitHub.



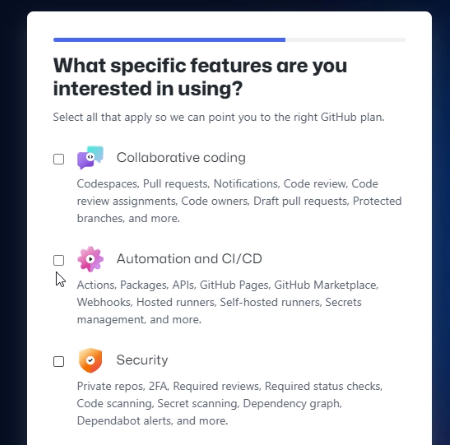
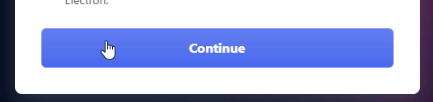
1. Masukkan **6-digit kode** tersebut ke **website GitHub**



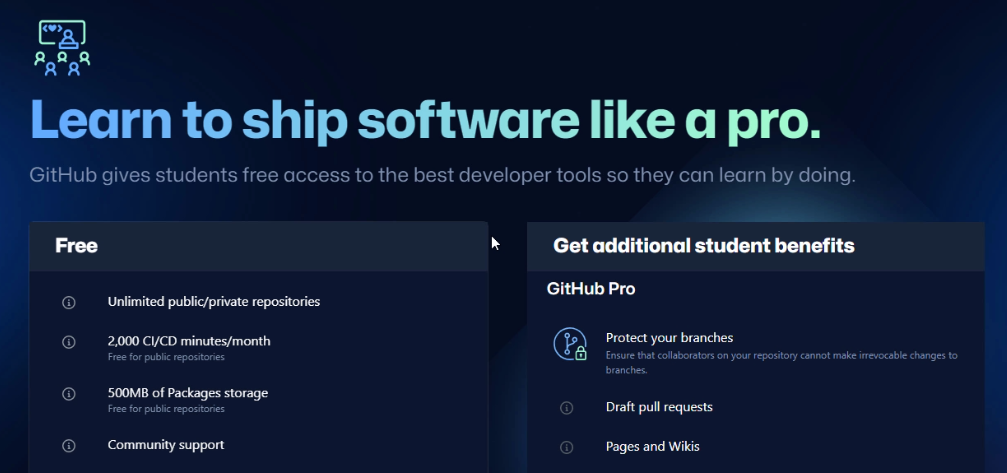
1. Isi **survey** sesuai dengan **kondisi** kamu. Lalu, klik tombol **continue**

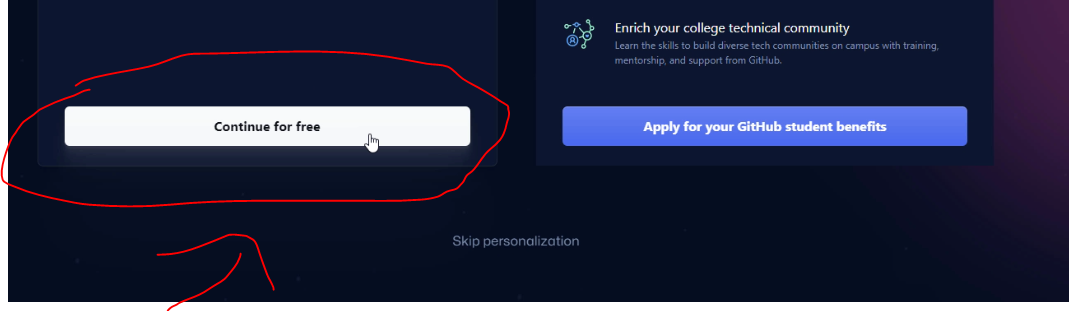


1. Pilih apa yang kamu **gunakan** **fitur** di **GitHub** ini. **Tidak perlu** memilih, langsung saja **scroll ke paling bawah** kemudian, klik tombol **continue**

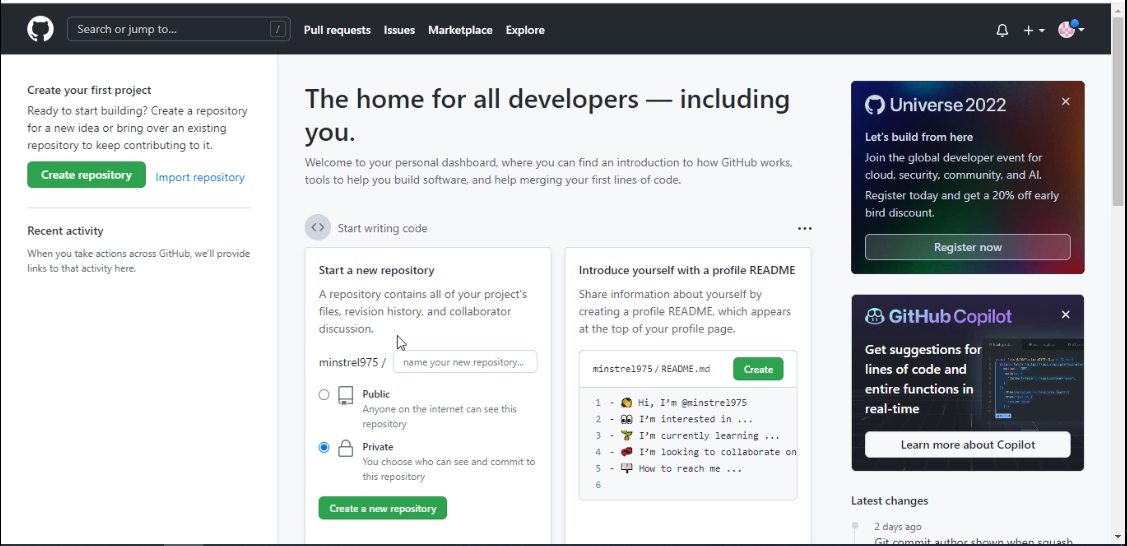
 

1. Pada halaman ini, langsung saja **scroll ke paling bawah**. Kemudian klik tombol **continue for free**



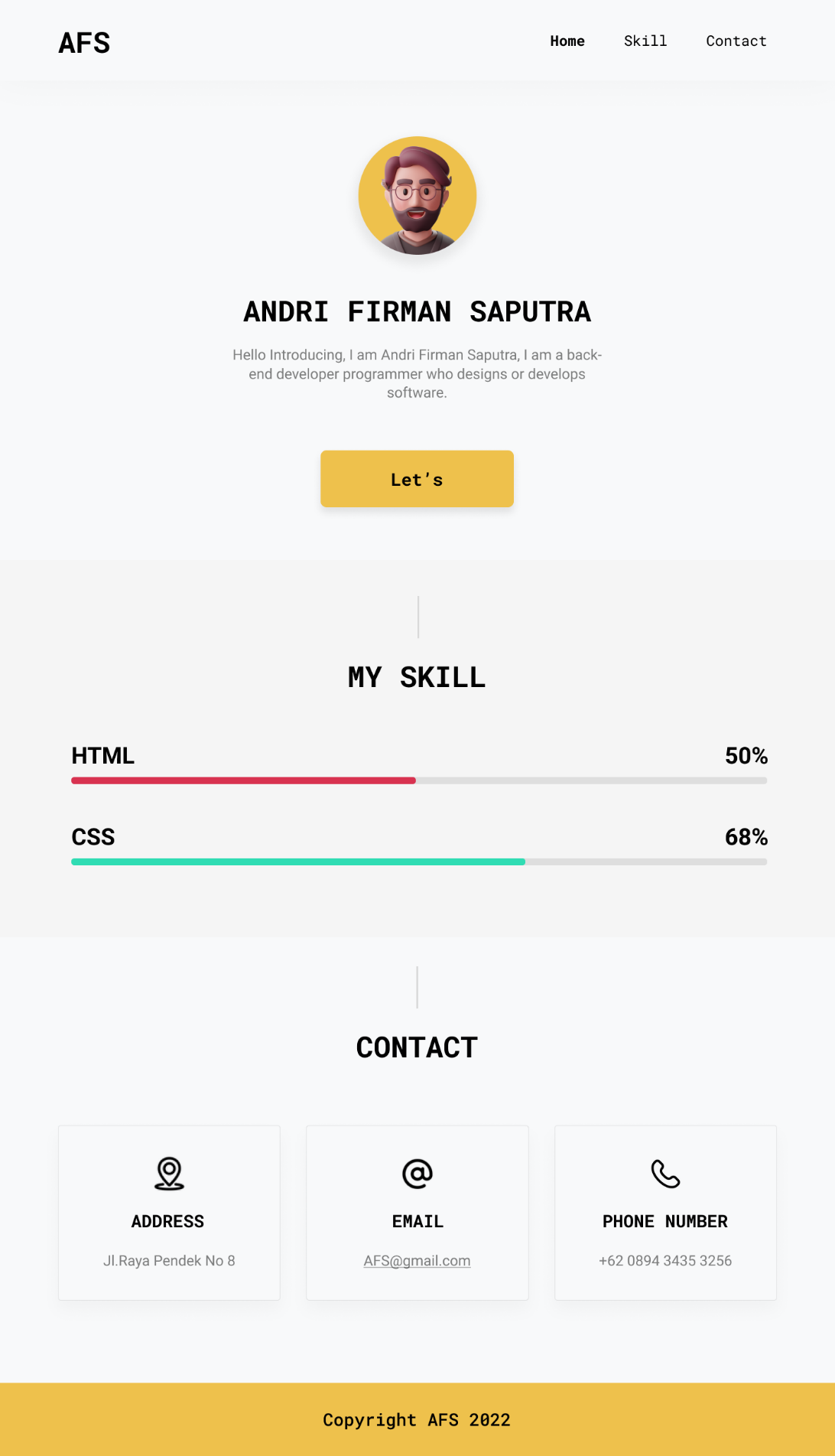


1. Selamat **akun** kamu **sudah siap digunakan**



## Melihat Mockup Figma

**Sebelum** kita **membuat** suatu **aplikasi**, **wajib** kita mengetahui **desainnya** terlebih dahulu.



## Membuat Portofolio

### Menyiapkan project

1. **Buat folder** dengan**nama kalian portofolio** di **Desktop**
2. **Buka** **Sublime Text,** lalu pilih **file > open folder**
3. Setelah itu, buka **Desktop** pilih **folder** **nama kalian portofolio,** lalu **select folder**

### Buat file index.html

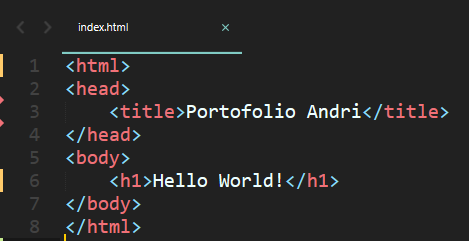
1. **Buat** file **baru** dengan cara **ctrl + n**
2. **Simpan** file dengan cara **ctrl + s**
3. **Simpan** dengan nama **index.html**
4. **Simpan** di **Desktop** **>** **folder nama kalian portofolio**

### Buat file style.css

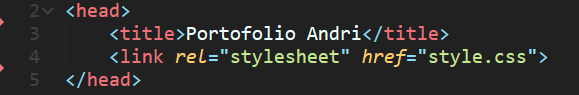
1. **Buat** file **baru** dengan cara **ctrl + n**
2. **Simpan** file dengan cara **ctrl + s**
3. **Simpan** dengan nama **style.css**
4. **Simpan** di **Desktop** **>** **folder nama kalian portofolio**

### Membuat portofolio

1. Pertama kita membuat **kode** script **HTML** yang paling **sederhana**



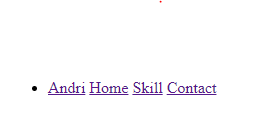
1. **Hubungkan** file **CSS** metode **External CSS**, dengan cara **ketik kode** berikut:



1. Membuat **nav bar** sederhana, ketikan **kode** di dalam tag **<body>**



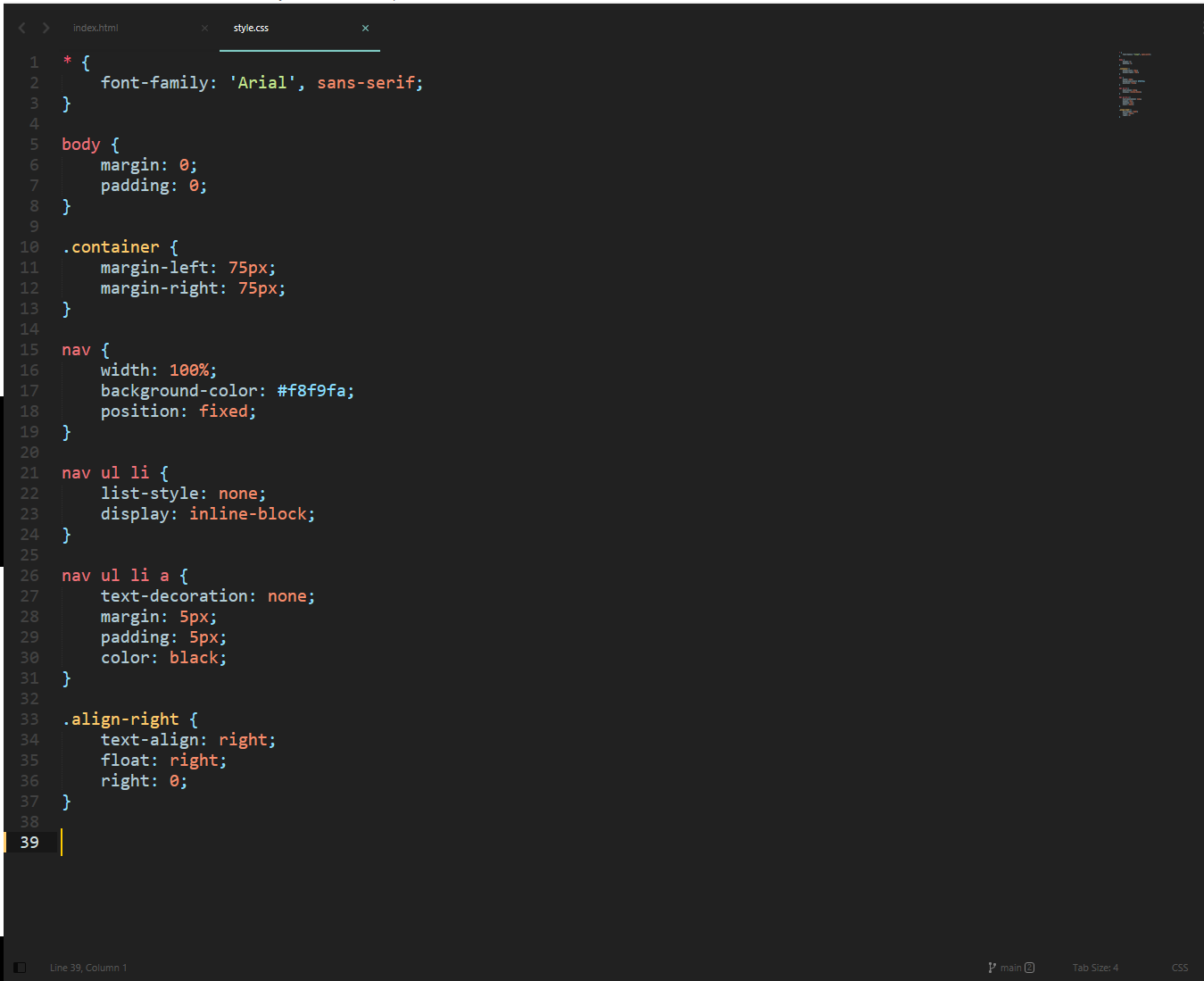
**Simpan** file dengan cara **ctrl + s**. Kemudian, jika kita **buka** file **index.html** nya melalui **browser**, maka akan **tampil** seperti ini:



1. Sekarang kita **ubah sedikit** **kode** pada bagian **nav**, mulai dari menambahkan **<div>** **class container**, sampai **memisahkan** **sebagian** **menu** ke sebelah **kanan**.Dengan cara membuat **<li>** **baru** kemudian, diberi class **align-right**. **Class-class** ini akan kita **konfigurasi** melalui file **style.css**



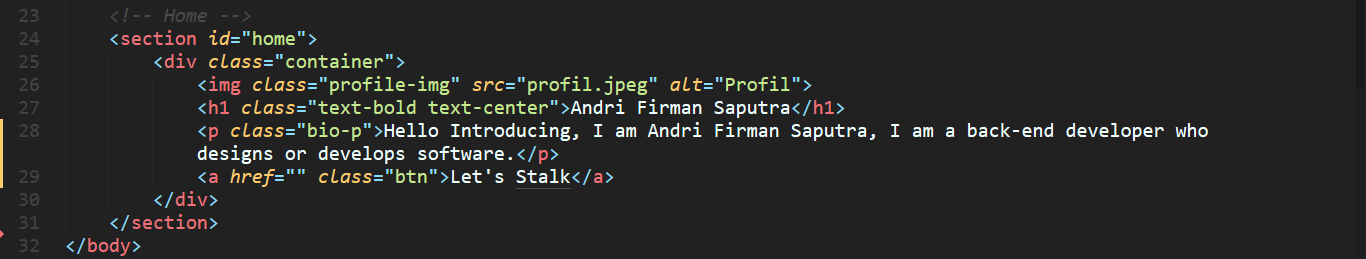
1. Sekarang kita menghias **nav bar** nya, buka file **style.css** ketikan **kode** berikut:



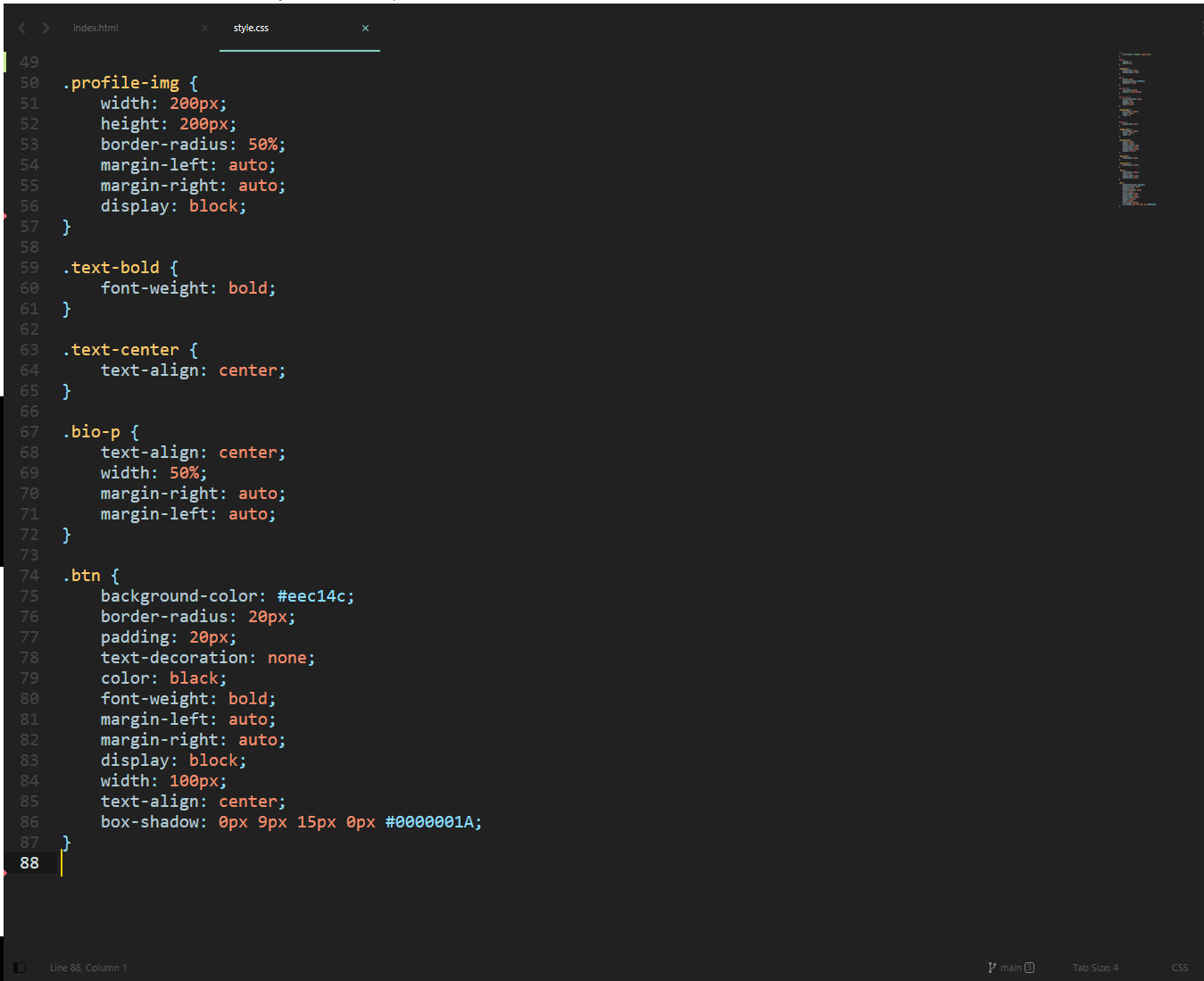
**Simpan** file dengan cara **ctrl + s**. Kemudian, jika kita **buka** file **index.html** nya melalui **browser**, maka akan **tampil** seperti ini:



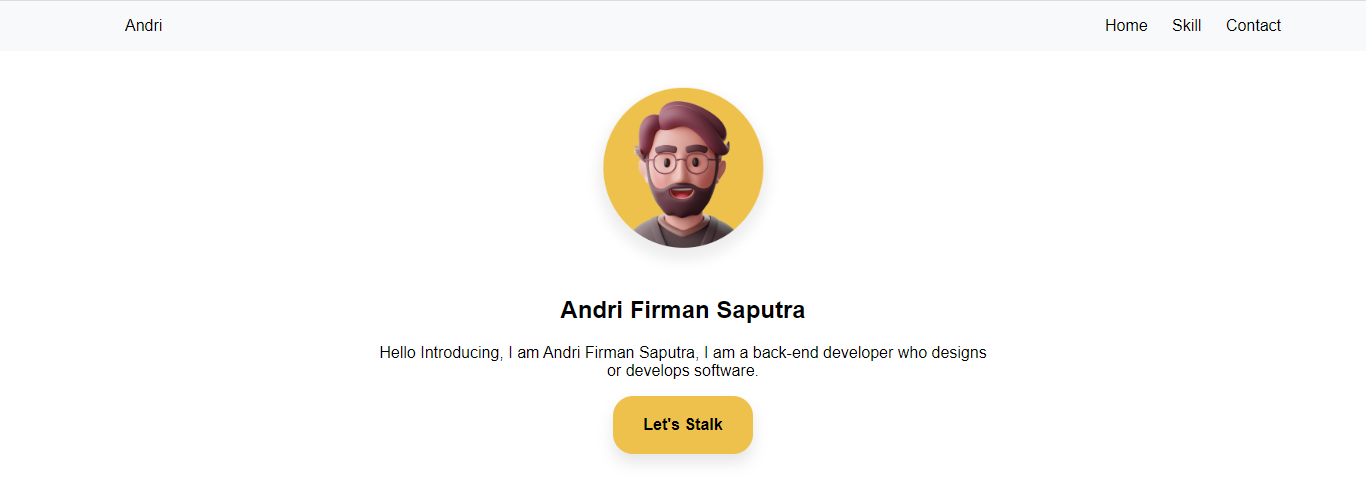
1. Selanjutnya, kita **membuat** bagian **home**.Ketikan **kode** di bawah tag **</nav>**



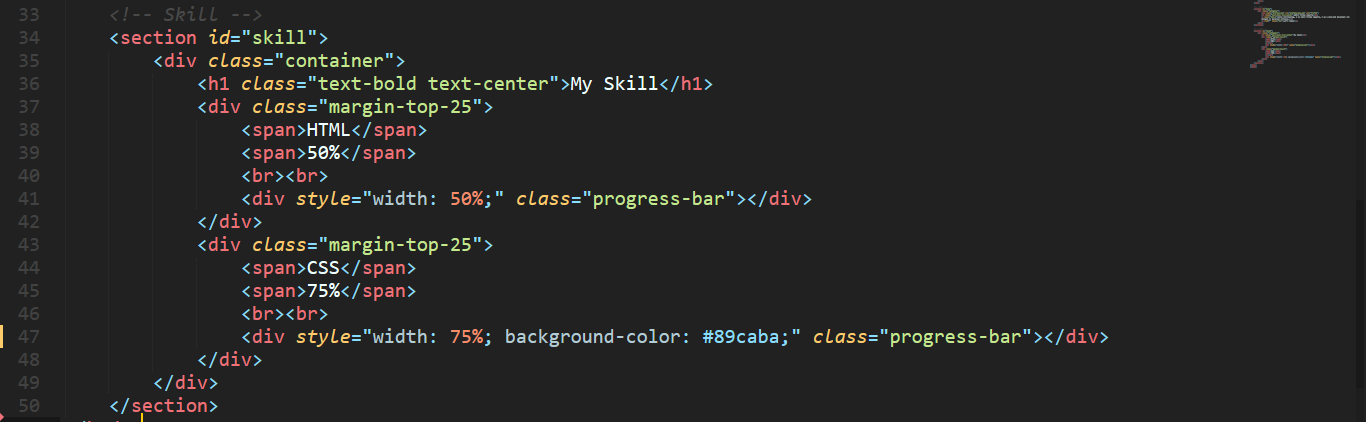
1. Kemudian, **menambahkan** **class** **baru** pada file **style.css**



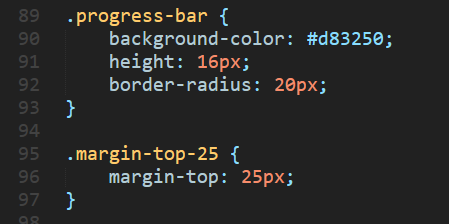
**Simpan** file dengan cara **ctrl + s**. Kemudian, jika kita **buka** file **index.html** nya melalui **browser**, maka akan **tampil** seperti ini:



1. Setelah itu, kita membuat menu **skill**. ketikan **kode** setelah menu **Home** di bawah tag **</section>**



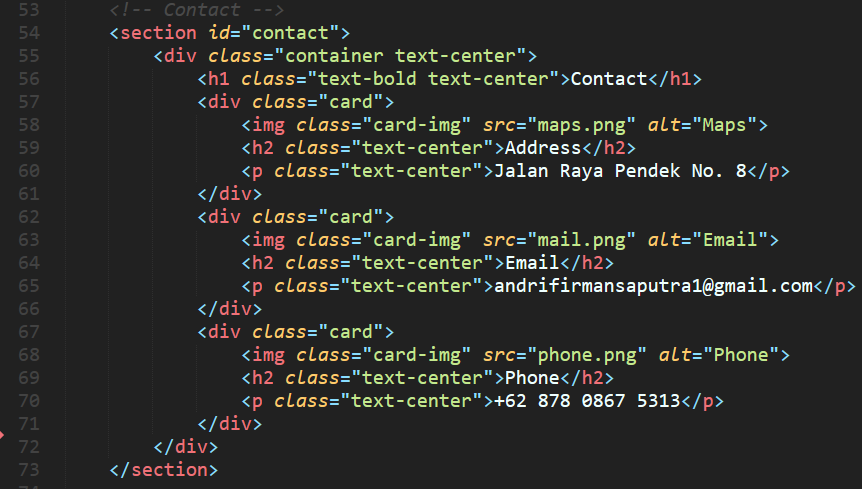
1. Kemudian, **menambahkan** **class** **baru** pada file **style.css**



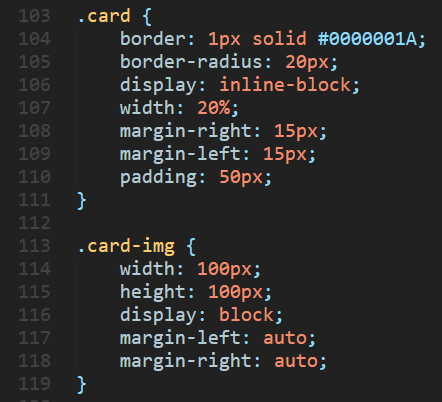
**Simpan** file dengan cara **ctrl + s**. Kemudian, jika kita **buka** file **index.html** nya melalui **browser**, maka akan **tampil** seperti ini:



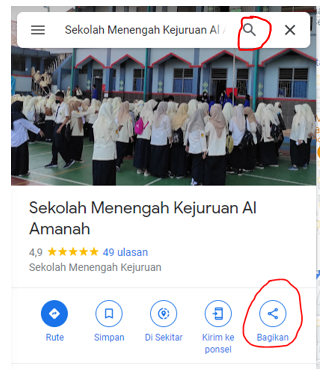
1. Setelah itu, kita membuat menu **contact**. ketikan **kode** setelah menu **skill** di bawah tag **</section>**



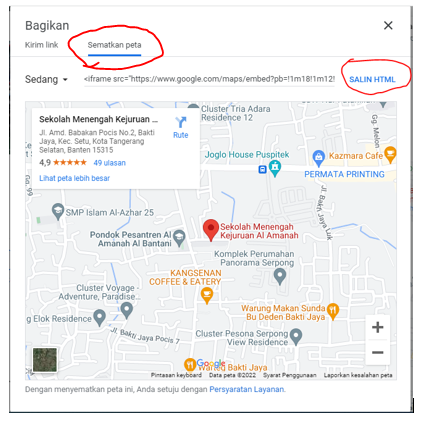
1. Kemudian, **menambahkan** **class** **baru** pada file **style.css**



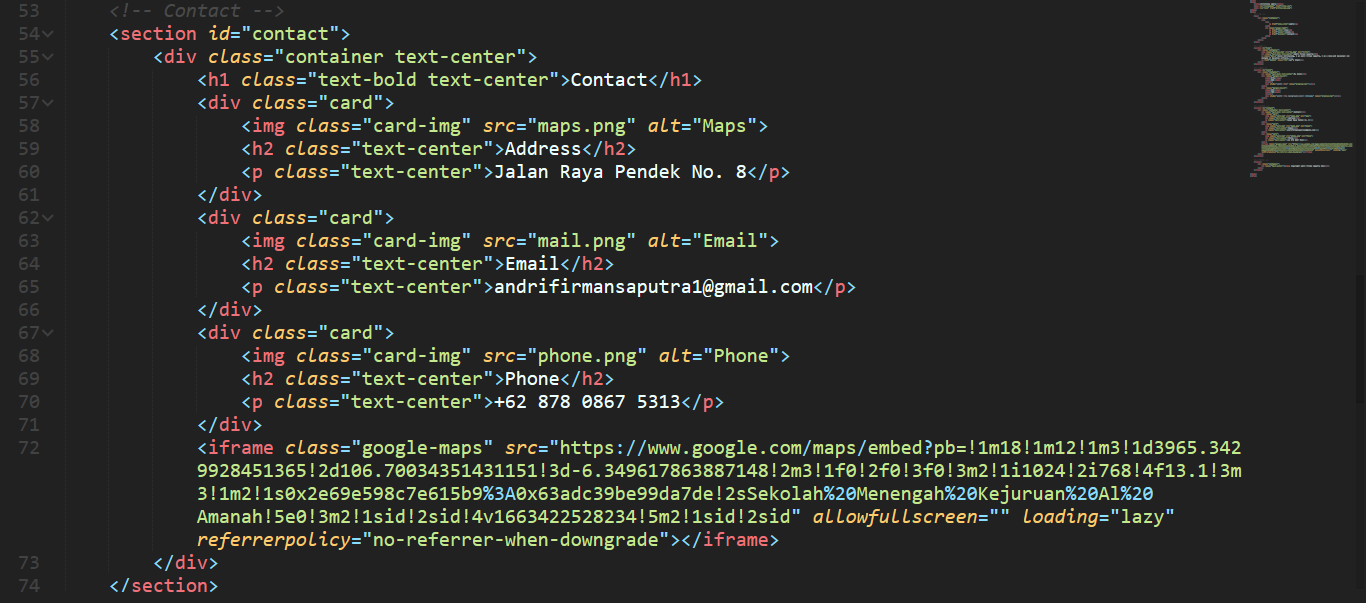
1. Selanjutnya, kita akan menambahkan **maps**.
   1. Pertama, kita membuka Google Maps. <https://maps.google.com/>
   2. Kedua, ketikan **lokasi** yang ingin kamu **cari**, kemudian klik tombol **cari**
   3. Ketiga, klik tombol **bagikan**



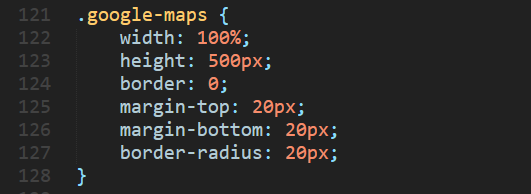
* 1. Keempat, klik tab **Sematkan peta**, lalu klik **SALIN HTML**



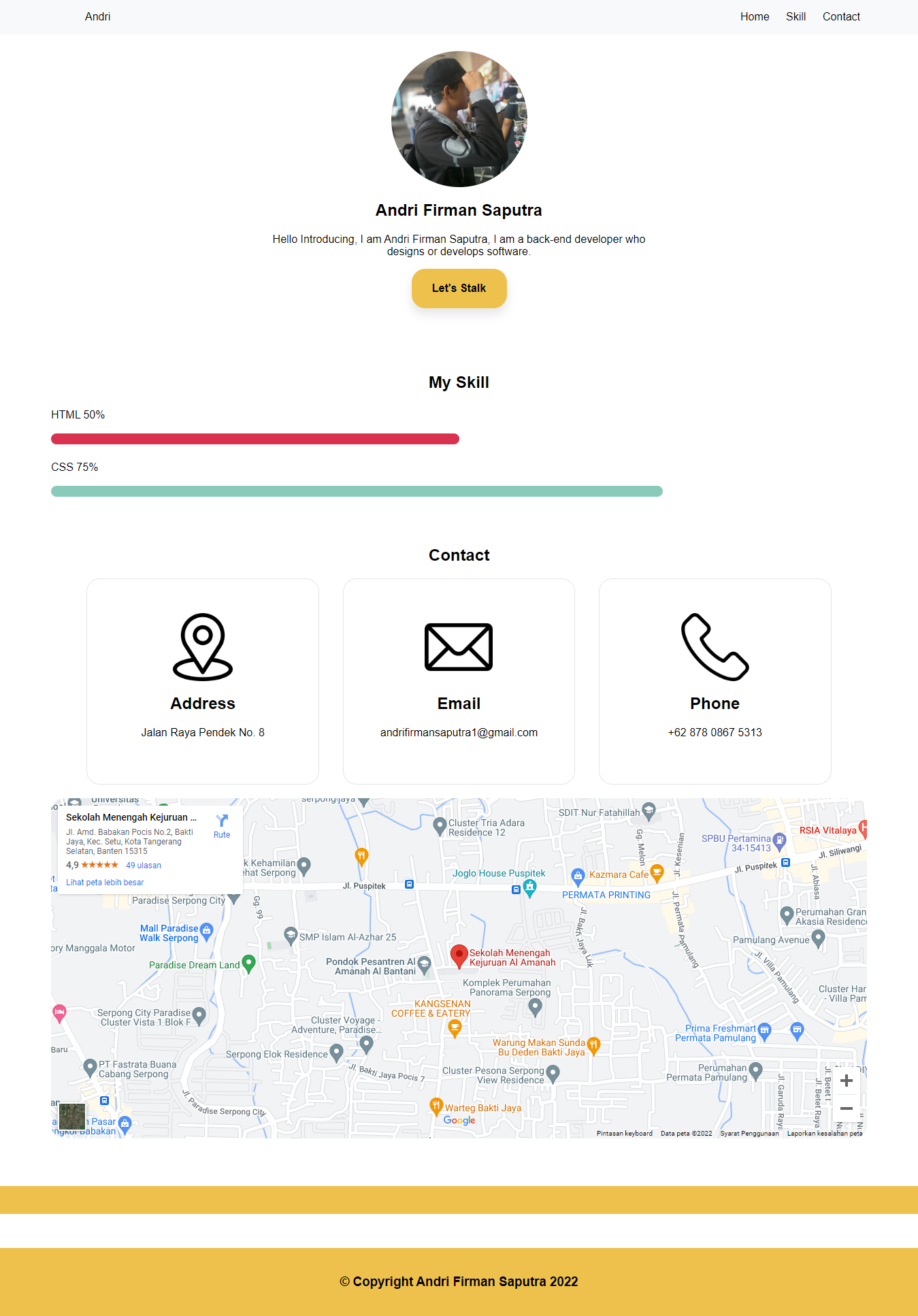
* 1. Kelima, **paste-kan** kode **<iframe>** seperti di bawah ini:



* 1. Kemudian, **menambahkan** **class** **baru** pada file **style.css**



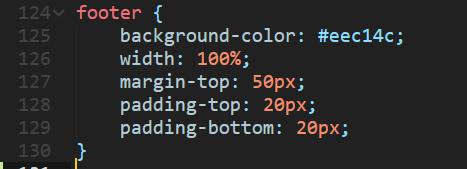
**Simpan** file dengan cara **ctrl + s**. Kemudian, jika kita **buka** file **index.html** nya melalui **browser**, maka akan **tampil** seperti ini:



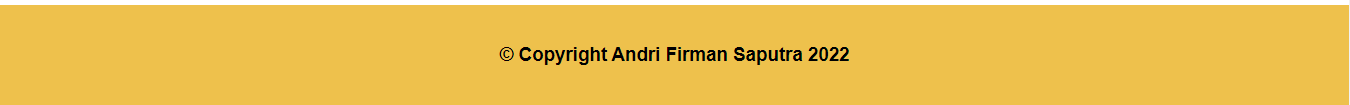
1. Setelah itu, kita membuat menu **footer**. ketikan **kode** setelah menu **contact** di bawah tag **</section>**



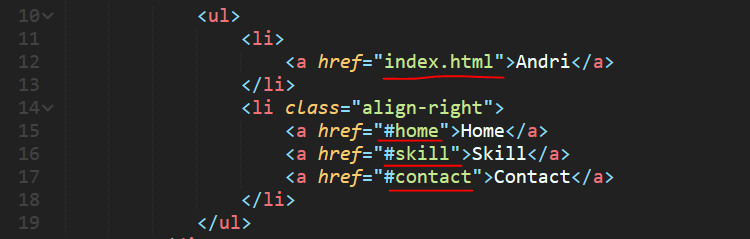
1. Kemudian, **menambahkan** **kode baru** pada file **style.css**



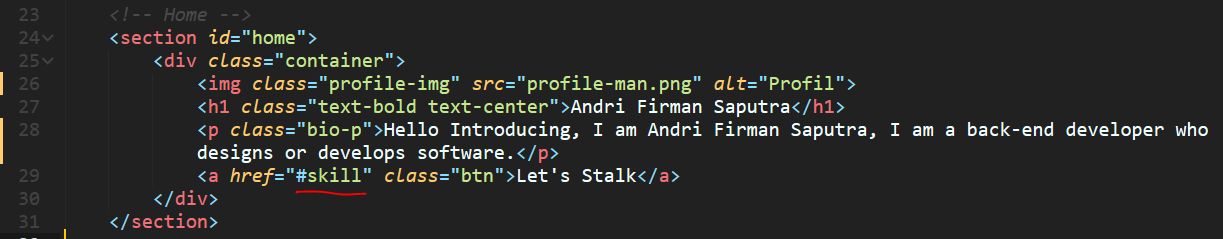
**Simpan** file dengan cara **ctrl + s**. Kemudian, jika kita **buka** file **index.html** nya melalui **browser**, maka akan **tampil** seperti ini:



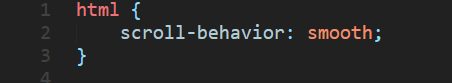
1. Setelah itu, kita **link** semua tag **<a href=””>**, di dalam tag **<nav>**



1. Di dalam menu **home**, tambahkan **<a href=”#skill”>**



1. Untuk m**emperhalus** **scroll** saat menekan **tombol**, tambahkan **kode** pada file **style.css** di **paling atas.**



1. Mengganti **ikon website** dengan cara menambahkan **tag** **<link>** dengan **value rel-nya icon**



**Sebelum** di ganti:

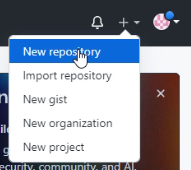


**Setelah** di ganti menjadi seperti ini:

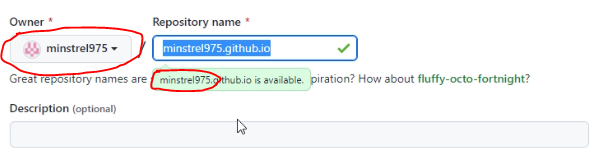


## Membuat Repository pada GitHub

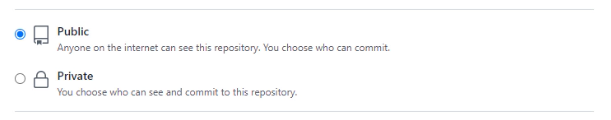
1. Pada **halaman utama** GitHub, klik tombol **tambah** di **nav bar** bagian **kanan atas**. Kemudian, klik tombol **new repository**



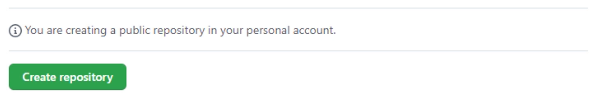
1. Untuk membuat **GitHub Pages**, **buat baru repository** dengan **nama** **username**.github.io, lihat seperti gambar di bawah ini:



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

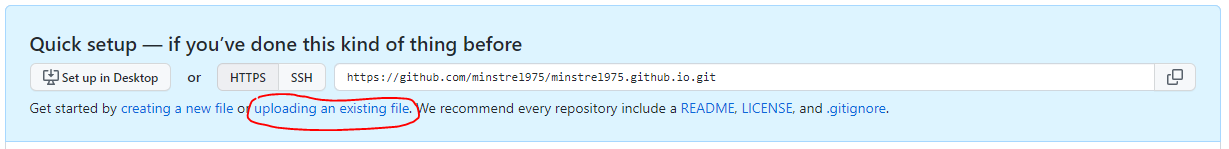


1. Lalu, klik tombol **create repository**

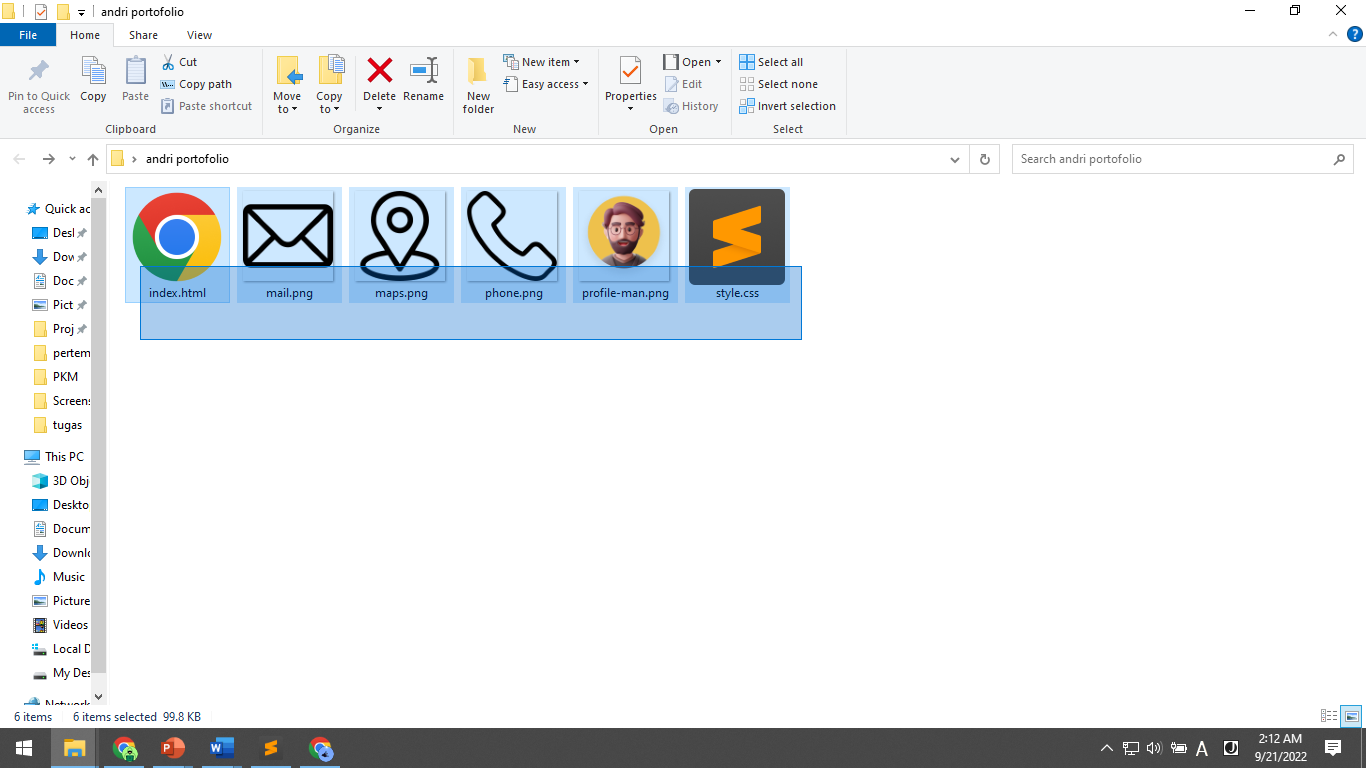


## Memposting Portofolio ke GitHub

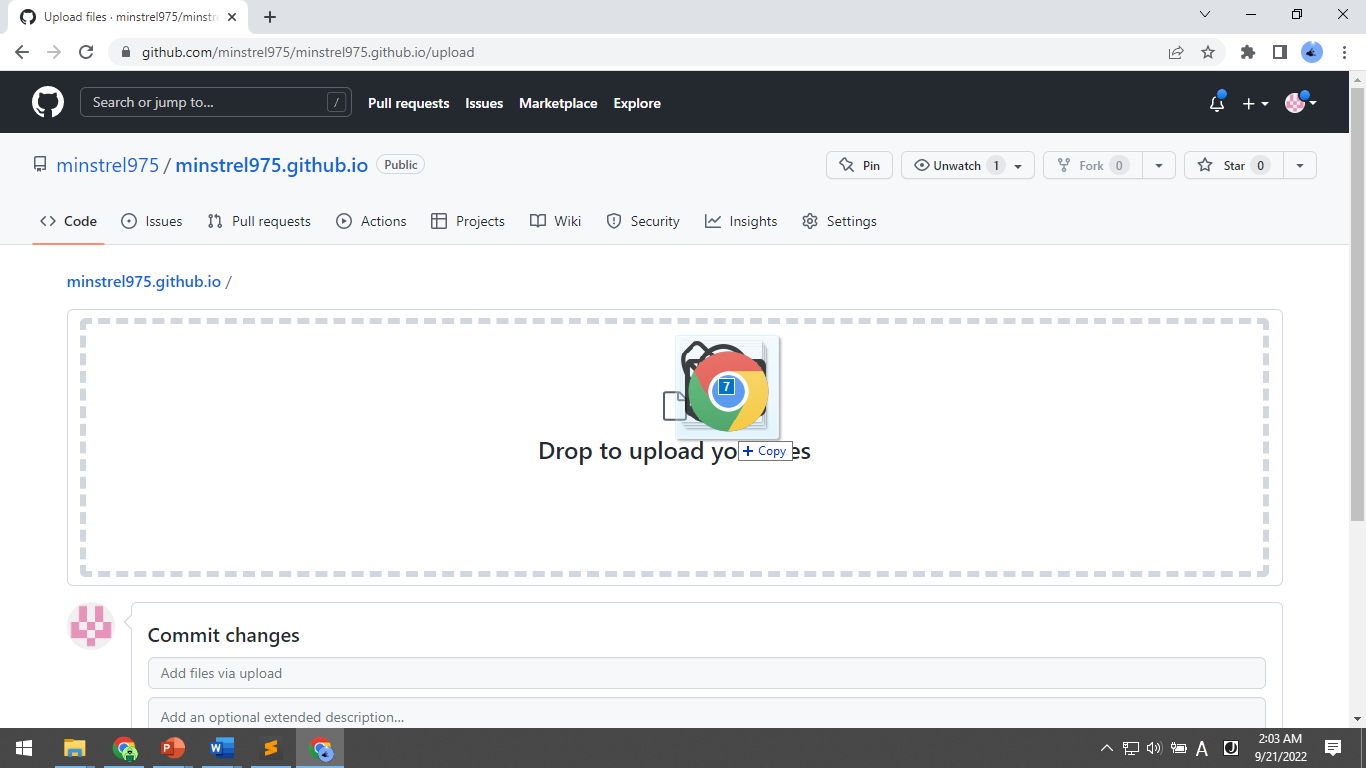
1. Setelah membuat **repository**, klik link **uploading an existing file**



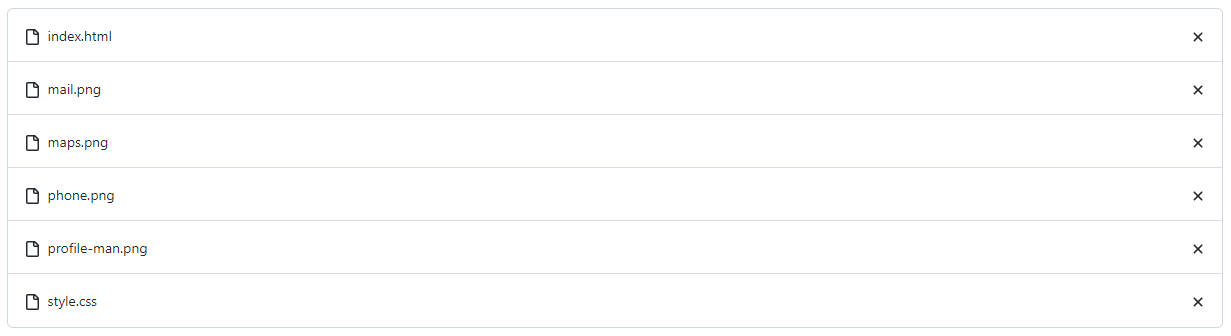
1. Buka folder **nama kalian portofolio,** pilih **semua file** lalu, **drag** ke **browser**



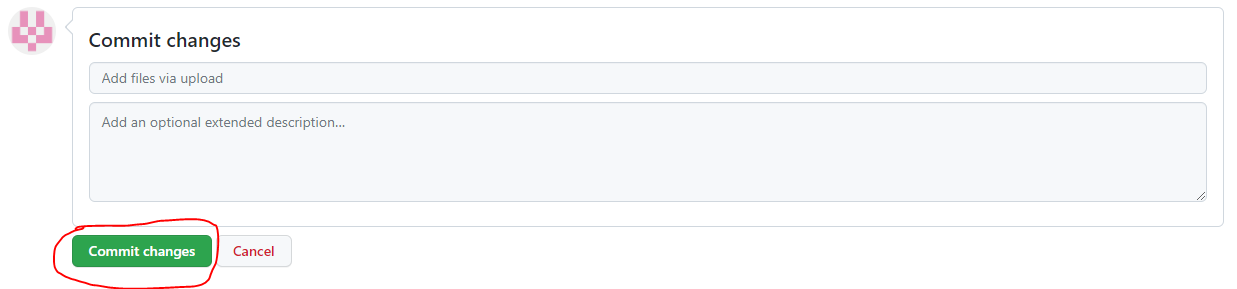
1. Kemudian, **drop** ke **GitHub**



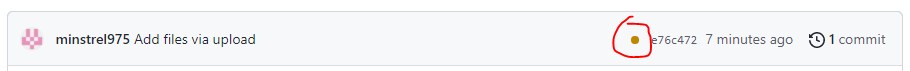
1. Pastikan **semua file telah terupload**



1. Setelah itu, kalian boleh **menambahkan komentar** sebelum mengupload **(opsional)**. Jika sudah, klik tombol **commit changes** untuk **mengupload.**



1. Jika terdapat **indicator** berwarna **orange** artinya **GitHub** sedang **memproses** website kamu.

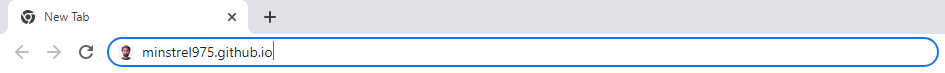


1. Jika **indicator** berubah menjadi **centang hijau**, artinya **website** kamu **siap digunakan.**

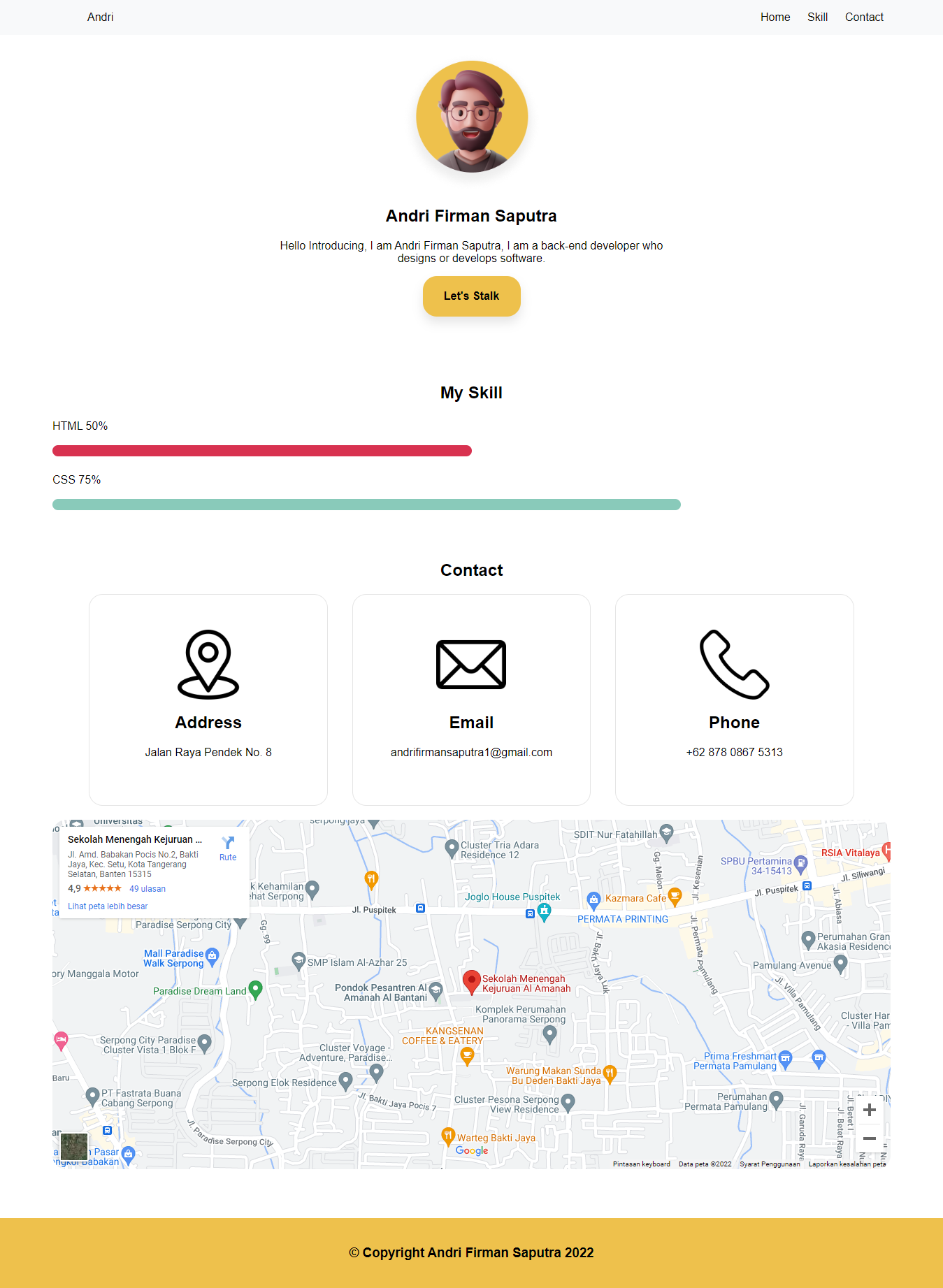


## Mereviews project portofolio

1. Ketik **username.github.io** pada **URL** di **browser**



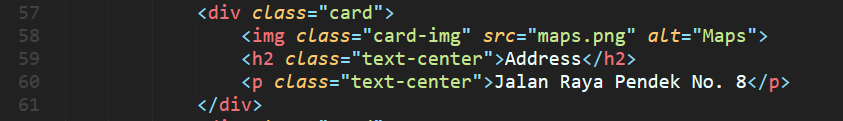
1. Berikut **hasil** website **project portofolio**



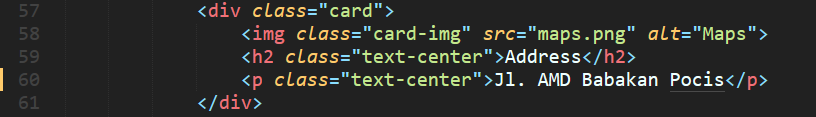
## Mengupdate perubahan pada portofolio

1. Misalkan saya **merubah** lokasi **alamat** saya

**Sebelum:**



**Sesudah:**

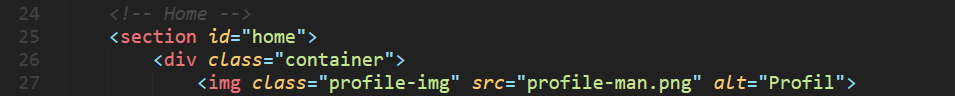


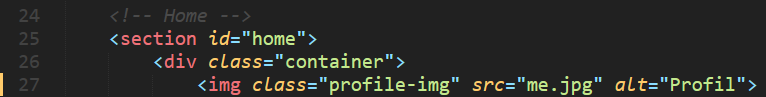
1. Lalu, saya juga ingin **mengganti** **foto** profil. Pertama saya **masukkan** terlebih dahulu **foto** saya ke dalam **folder.**



1. Kemudian, mengubah **kode** file **index.html**

**Sebelum:**

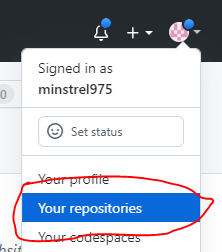
**Sesudah:**



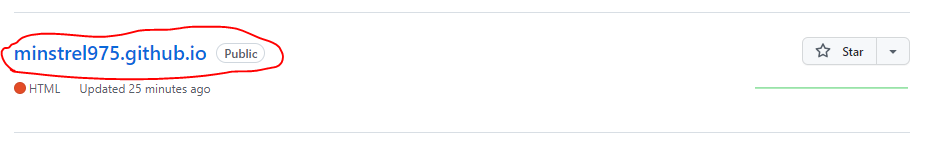
1. Ketikkan **github.com/username/username.github.io** pada **URL** di **browser** untuk membuka **repository.**



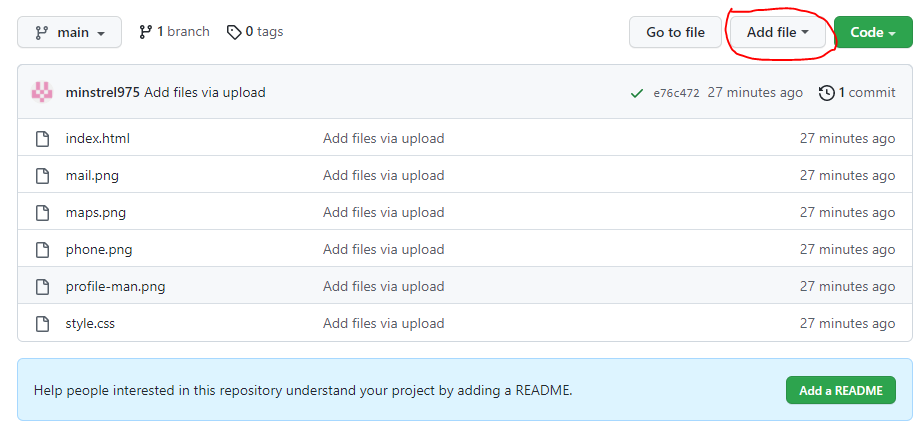
1. Bisa juga dengan cara lain, klik **logo** **profile** di **kanan atas** bagian **nav bar**. Kemudian, klik **Your repositories.**



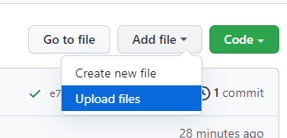
1. Setelah itu, untuk **membuka** **repository** klik **nama repository-nya.**



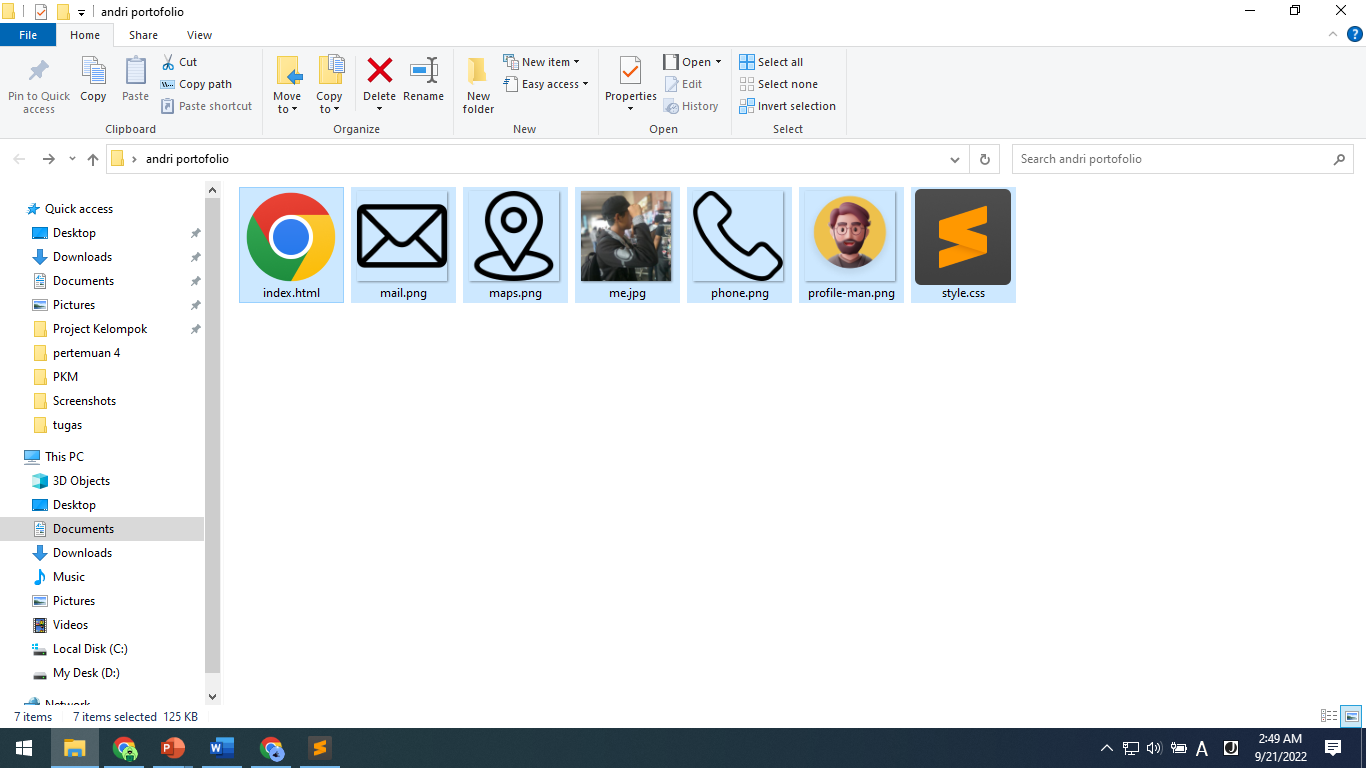
1. Kemudian, klik tombol **Add file**



1. Lalu, klik **Upload files**



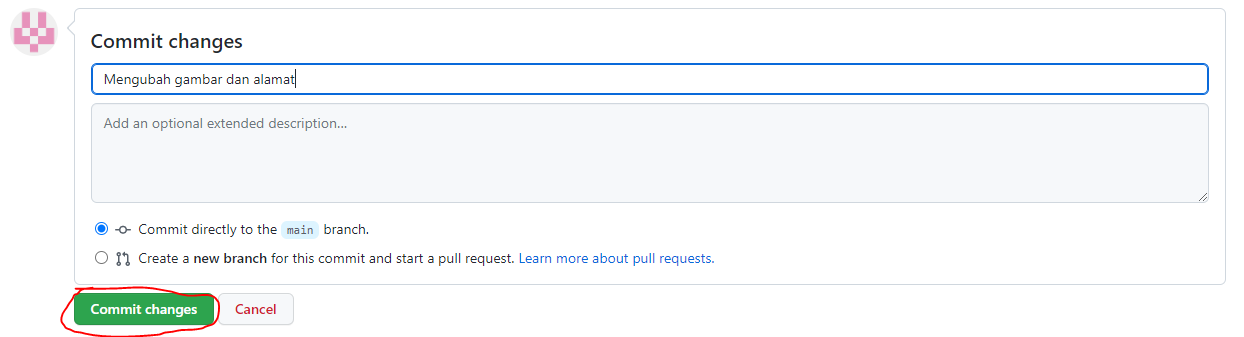
1. Pilih **semua file**, sebenarnya hanya **file** yang **terubah** saja yang **dipilih**, tetapi untuk **menimalisir** kesalahan sebaiknya pilih **semua** **file.** Kemudian, **drag** file ke **browser.**



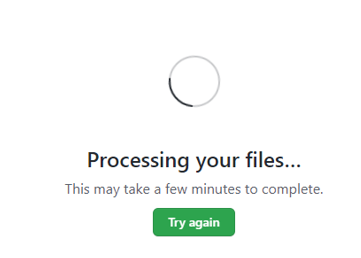
1. **Drop** file ke **GitHub**



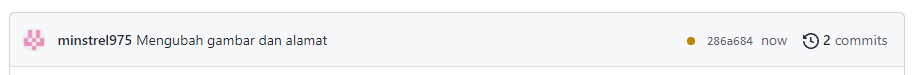
1. Kemudian, isi **komentar** (opsional). Lalu, klik tombol **commit changes.**



1. Tunggu **proses** upload file



1. Setelah itu, akan tampil lagi **indicator orange**



1. Tunggu sampai **proses** **selesai**



1. Review Kembali **Portofolio,** ketikkan **username.github.io** pada **URL** di **browser.**

