

LAPORAN
PRAKTIKUM APLIKASI KOMPUTER
”WEBSITE”



LA ODE MUHAMMAD YUDHY PRAYITNO
E1E122064

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HALU OLEO
KENDARI
2022

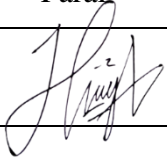




**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN RISET DAN
TEKNOLOGI
UNIVERSITAS HALU OLEO
FAKULTAS TEKNIK
JURUSAN TEKNIK INFORMATIKA**

Alamat : Jl. H.E.A Mokodompit Kampus Baru Tridarma Anduonohu, Kendari 92132
Tlp. (0401) 3195287, 3194347, 319083 Kendari Website : eng.uho.ac.id

LEMBAR ASISTENSI

NAMA : LA ODE MUHAMMAD YUDHY PRAYITNO
STAMBUK : E1E122064
MATA KULIAH : PRAKTIKUM APLIKASI KOMPUTER
JURUSAN : TEKNIK INFORMATIKA
JUDUL PRAKTIKUM: WEBSITE
KELOMPOK : I (SATU)

NO	Tanggal	Uraian	Paraf
1.	Sabtu, 10/12/2022	<ul style="list-style-type: none">Perbaiki indentansi paragraf	
2.	Rabu, 14/12/2022	<ul style="list-style-type: none"><i>Italic</i> kata asing pada daftar gambarUkuran spasi pada lembar asistensi tidak sesuai format yang diberi	
3.	Kamis, 15/12/2022	ACC	

Kendari, 15 Desember 2022

Asisten Dosen



Hadijah Nisa Ifayatin

E1E120008

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunianyalah saya dapat menyelesaikan laporan praktikum ini dengan judul “*Website*” ini tepat pada waktunya. Adapun tujuan dari penulisan laporan ini agar dapat memenuhi syarat dari tugas Praktikum Aplikasi Komputer. Selain itu, laporan ini bertujuan untuk menambah wawasan mengenai mengaplikasikan aplikasi komputer dalam kehidupan sehari-hari bagi para pembaca dan penulis.

Saya mengucapkan terima kasih kepada Bapak Rizal Adi Saputra, ST. M.Kom., selaku Dosen Praktikum Aplikasi Komputer atas bimbingannya dalam pembuatan laporan ini sehingga bisa meningkatkan wawasan pemikiran saya tentang bidang studi yang saya tekuni, dan saya juga berterima kasih kepada rekan-rekan yang telah membantu proses penyelesaian laporan yang tidak dapat disebutkan namanya satu persatu.

Dalam laporan ini saya merasa masih ada kekurangan baik dari segi penulisan dan materi yang saya paparkan maka dari itu saya sangat meminta kritik dan saran yang membangun dari para pembaca guna untuk memperbaiki pembuatan laporan selanjutnya.

Kendari, 15 Desember 2022

Penyusun

DAFTAR ISI

LEMBAR ASISTENSI.....	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	ix
BAB I PENDAHULUAN.....	1
1.1. Landasan Teori	1
1.1.1 Pengertian <i>Website</i>	1
1.1.2 Sejarah Perkembangan <i>Website</i>	2
1.1.3 Pengertian HTML dan CSS	4
1.2 Tujuan	5
1.3 Manfaat	5
BAB II METODOLOGI PRAKTIKUM.....	6
2.1 Waktu dan Tempat Praktikum	6
2.1.1 Waktu.....	6
2.1.1 Tempat	6
2.2 Alat dan Bahan	6
2.2.1 Alat	6
2.2.2 Bahan	7
2.3 Prosedur Praktikum	7
BAB III ANALISIS DAN PEMBAHASAN	8
3.1 Hasil Praktikum	8
3.2 Analisis Dan Pembahasan	23
3.2.1 Penggunaan HTML	23
3.2.2 Penggunaan CSS.....	29
3.2.3 Penggunaan <i>Hosting Website</i>	31
BAB IV PENUTUP.....	37
4.1 Kesimpulan	37
4.2 Saran	37
DAFTAR PUSTAKA	38

LAMPIRAN	39
----------------	----

DAFTAR GAMBAR

Gambar 3. 1 Tampilan membuat <i>New Folder</i>	8
Gambar 3. 2 <i>Software</i> Visual Studio Code.....	8
Gambar 3. 3 Open <i>Folder</i> pada Visual Studio Code.....	9
Gambar 3. 4 Open <i>Folder</i> pada Visual Studio Code.....	9
Gambar 3. 5 Mengaktifkan <i>sidebar</i> pada Visual Studio Code	9
Gambar 3. 6 Membuat <i>file</i> baru pada Visual Studio Code	10
Gambar 3. 7 Membuat <i>file index.html</i>	10
Gambar 3. 8 Tampilan <i>new file</i>	10
Gambar 3. 9 Membuat <i>file style.css</i>	11
Gambar 3. 10 Membuat <i>folder image</i>	11
Gambar 3. 11 Tampilan gambar yang berhasil dimasukan	11
Gambar 3. 12 <i>Snippet</i> HTML	12
Gambar 3. 13 Struktur standar <i>file</i> HTML	12
Gambar 3. 14 Mengubah <i>title</i> pada <i>website</i>	12
Gambar 3. 15 Menu <i>extension Live Server</i>	13
Gambar 3. 16 <i>Open with Live Server</i> pada Visual Studio Code.....	13
Gambar 3. 17 Membuat <i>navigation bar</i> pada HTML.....	14
Gambar 3. 18 Mengedit tampilan <i>navigation bar</i> melalui CSS	14
Gambar 3. 19 Membuat tampilan <i>home</i> di HTML	14
Gambar 3. 20 Mengedit tampilan <i>home</i> di CSS	15
Gambar 3. 21 Membuat tampilan <i>resume</i> di HTML	15
Gambar 3. 22 Mengedit tampilan <i>resume</i> di CSS	15
Gambar 3. 23 Mengedit tampilan <i>skill</i> kita di HTML	16
Gambar 3. 24 Mengedit tampilan <i>skill</i> di CSS	16
Gambar 3. 25 Membuat rangkuman materi Windows di HTML	16
Gambar 3. 26 Mengedit tampilan materi Sistem operasi Windows di CSS	17
Gambar 3. 27 Memasukan materi mengenai Word di HTML.....	17
Gambar 3. 28 Mengedit tampilan Word di CSS	17
Gambar 3. 29 Menambahkan tombol <i>button</i> pada materi Word di HTML.	18
Gambar 3. 30 Mengedit tampilan tombol <i>button</i> di CSS	18
Gambar 3. 31 Memasukan materi Excel dan Google Slides di HTML.....	18

Gambar 3. 32 Membuat judul Aplikasi Perkantoran di HTML.....	19
Gambar 3. 33 Membuat judul <i>digital art</i>	19
Gambar 3. 34 Memasukan nama <i>vector art</i> serta gambarnya di HTML.....	19
Gambar 3. 35 Mengatur tampilan dan ukuran <i>vector art</i> di CSS	20
Gambar 3. 36 Menambahkan tombol <i>button</i> di <i>border vector art</i>	20
Gambar 3. 37 Mengedit tampilan <i>button</i> di CSS.....	20
Gambar 3. 38 Menyalin <i>border vector art</i>	21
Gambar 3. 39 Menyesuaikan tampilan dengan jenis <i>digital art</i> lainnya	21
Gambar 3. 40 Menambahkan <i>contact</i> kita di bagian <i>footer</i> HTML	21
Gambar 3. 41 Mengatur tampilan <i>contact</i> di CSS.....	22
Gambar 3. 42 Tampilan <i>website</i> portfolio	22
Gambar 3. 43 Tampilan <i>website</i> portfolio	22
Gambar 3. 44 Tampilan <i>website</i> portfolio	23
Gambar 3. 45 . Struktur standar <i>file</i> HTML	23
Gambar 3. 46 Ubah <i>tag title</i> pada <i>website</i>	24
Gambar 3. 47 Hasil pengubahan <i>title</i> pada <i>website</i>	24
Gambar 3. 48 <i>Tag p</i> dan <i>br</i> pada HTML	25
Gambar 3. 49 Hasil tampilan pada <i>browser</i>	25
Gambar 3. 50 <i>Tag a</i> pada HTML.....	25
Gambar 3. 51 Tampilan pada <i>browser</i>	26
Gambar 3. 52 <i>Tag header</i>	26
Gambar 3. 53 Tampilan pada <i>browser</i>	26
Gambar 3. 54 <i>Tag img</i>	27
Gambar 3. 55 Tampilan pada <i>browser</i>	27
Gambar 3. 56 <i>Tag div</i>	27
Gambar 3. 57 Tampilan pada <i>browser</i>	28
Gambar 3. 58 <i>Tag span</i>	28
Gambar 3. 59 Tampilan pada <i>browser</i>	28
Gambar 3. 60 Komentar pada <i>file</i> HTML	28
Gambar 3. 61 Tampilan pada <i>browser</i>	29
Gambar 3. 62 <i>Tag formatting</i> pada HTML	29
Gambar 3. 63 Tampilan pada <i>browser</i>	29

Gambar 3. 64 Styling <i>file</i> HTML menggunakan CSS.....	30
Gambar 3. 65 Pemberian <i>style</i> menggunakan <i>file style.css</i>	30
Gambar 3. 66 Tampilan pada <i>browser</i>	31
Gambar 3. 67 Menghubungkan <i>file</i> CSS pada <i>file</i> HTML	31
Gambar 3. 68 Tampilan <i>file</i> yang dibutuhkan dalam <i>website</i>	31
Gambar 3. 69 Tampilan <i>home</i> Github	32
Gambar 3. 70 Tampilan menu <i>sign</i>	32
Gambar 3. 71 Membuat <i>repository</i>	32
Gambar 3. 72 Memasukan nama <i>repository</i>	33
Gambar 3. 73 Meng- <i>upload file</i>	33
Gambar 3. 74 Mengunggah <i>file</i>	34
Gambar 3. 75 Tampilan <i>file</i> yang berhasil diunggah.....	34
Gambar 3. 76 Mengatur tautan	34
Gambar 3. 77 Memilih <i>Branch</i> dan <i>Custom Domain</i>	35
Gambar 3. 78 Tampilan tautan <i>website</i> yang telah bisa diakses.....	35
Gambar 3. 79 Tampilan <i>Website</i> yang telah berhasil diakses	36

DAFTAR TABEL

Tabel 2. 1 Alat dan Fungsinya.....	6
Tabel 2. 2 Bahan dan Fungsinya.....	7

BAB I

PENDAHULUAN

1.1. Landasan Teori

1.1.1 Pengertian *Website*

Website adalah kumpulan halaman yang saling terhubung yang di dalamnya terdapat beberapa item seperti dokumen dan gambar yang tersimpan di dalam *web server*. *Web app* adalah sebuah aplikasi yang berada dalam *web server* yang bisa *user* akses melalui *browser*. *Web app* biasanya menampilkan data *user* dan informasi dari server.

Website adalah kumpulan halaman *web* yang saling terhubung dan seluruh *file* saling terkait. *Web* terdiri dari *page* atau halaman dan kumpulan halaman yang dinamakan homepage. *Homepage* berada pada posisi teratas dengan halaman-halaman terkait berada di bawahnya. Biasanya, setiap halaman di bawah *homepage* (*child page*) berisi *hyperlink* ke halaman lain dalam *web*.

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik antara *page* yang disimpan dalam server yang sama maupun server di seluruh dunia. Halaman dapat diakses dan dibaca melalui *browser* seperti Google Chrome, Mozilla Firefox, dan lainnya.

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk dikunjungi.

Website adalah kumpulan dari halaman-halaman situs yang terdapat dalam sebuah domain atau subdomain yang berada di dalam *World Wide Web* (WWW) di internet.

1.1.2 Sejarah Perkembangan Website

Dilihat dari sisi sejarahnya, penemu situs *web* yaitu Sir Timothy John "Tim" Berners-Lee, sedangkan situs *web* yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Tujuan dari Tim ketika merancang situs *web* adalah untuk memudahkan tukar menukar dan memperbarui informasi pada sesama peneliti di tempat ia bekerja. Pada tanggal 30 April 1993, CERN yaitu suatu tempat dimana tim bekerja mengumumkan bahwa WWW dapat digunakan secara gratis oleh publik. Sejarah *web service* berada pada perkembangan *web*. Dalam perkembangannya *web* dibagi menjadi 3 yaitu, *web 1.0*, *web 2.0* dan *web 3.0*.

A. *Web 1.0*

Web 1.0 merupakan generasi pertama dari layanan internet berbasis *web*. Pengguna hanya bisa membaca dan mencari, serta melihat-lihat informasi yang ada dalam sebuah *web* tanpa bisa melakukan interaksi lainnya. Pada masa *web 1.0* dimana *web* pertama kali berawal dan masi statis karena pembuat *web* dan pengguna *web* hanya terjadi komunikasi satu arah. Sehingga pada saat itu *web* tipe ini seolah-olah berperan sebagai papan pengumuman yang dihadirkan secara digital. Adapun ciri-ciri *web 1.0* sebagai berikut:

1. *Web pages* bersifat statik, di mana *website* hanya bersifat "*read-only*" sehingga terjadi komunikasi satu arah. Pembaca hanya dapat menerima informasi, tanpa berkontribusi dalam memberikan informasi.
2. Penggunaan *frameset*, yaitu tampilan beberapa *web page* dalam satu *window*.
3. Penggunaan *tag* HTML seperti *<blink>* dan *<marquee>*
4. Penggunaan *button* dengan *format* GIF, biasanya 88 x 31 untuk promosi produk.

B. *Web 2.0*

Web 2.0 merupakan generasi lanjutan dari *web 1.0* yang dikenal dengan istilah Web Sosial. *Web 2.0* mengacu pada situs *web* yang menekankan konten yang dibuat pengguna, kemudahan penggunaan, budaya partisipatif,

dan interoperabilitas (yaitu, kompatibel dengan produk, sistem, dan perangkat) untuk pengguna akhir.

Pada masa *web 2.0* berbagai fitur-fitur dan inovasi baru bermunculan membawa suatu pandangan baru bagi dunia *web*. Istilah *web 2.0* disebut-sebut oleh Dale Dougherty dari O'Reilly Media yang melakukan *brainstorming* dengan Craig Cline dari *Media Live* untuk menghasilkan ide konferensi di mana mereka menjadi *host*. *Web 2.0* memiliki ciri-ciri sebagai berikut:

1. Pengguna *web* dapat mengubah isi *web page*
2. Penggunaan *web page* untuk berhubungan dengan user lain, seperti pada situs Facebook karena pengguna mudah untuk berkomunikasi dan menjalin silaturahmi dengan kerabat atau temannya.
3. *Share content* yang cepat dan efisien, seperti YouTube, dimana pengguna bisa mengunggah video dengan relatif cepat dan berbagi video pada pengguna yang lainnya.
4. Cara baru untuk mendapatkan informasi, yaitu *Really Simple Syndication* (RSS). Dengan menggunakan RSS, dimana *user* akan mendapatkan notifikasi jika sebuah *web - page* ter *update*.
5. Koneksi internet tidak hanya melalui perangkat komputer, tetapi bisa diakses melalui perangkat lain seperti *handphone*.

Berdasarkan ciri-ciri diatas pada masa *web 2.0* sudah mulai berkembangnya *web* lebih baik dimana *user* tidak hanya berinteraksi satu arah. Dimana pengguna dapat berkomunikasi dengan pengguna lainnya pada situs seperti Facebook, Twitter dan lain-lain, bahkan kita dapat meng-*upload* video dan *sharing* di YouTube pada situs *web* yang menjadi platformnya itu terjadi dikarenakan ada pelayanan pada sistem *web*.

C. *Web 3.0* (2001)

Pada *web 3.0* perkembangan web lebih berkembang setelah *web 2.0*. konsep *web 3.0* pertama kali diperkenalkan pada tahun 2001, yaitu saat penemu *world wide web* Berners-lee menulis sebuah artikel ilmiah yang menggambarkan *web 3.0* dimana sebagai sarana bagi mesin untuk membaca halaman – halaman *web*. Hal ini mesin akan memiliki kemampuan untuk membaca halaman *web* yang manusia lakukan seperti saat ini. Dan ciri dari

web 3.0 adalah adanya trend teknologi salah satunya *API(Application Programming Interface)* merupakan sebuah *interface* yang diimplementasikan dengan menggunakan software sehingga *softwer* dapat saling berinteraksi dengan *software* lain. Ketika kita memperhatikan di masa *web 3.0* kita dapat melihat bahwa *web service* mulai ada dan menjadi *trend* teknologi.

Web 3.0 menawarkan metode yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data *online*. *Web 3.0* juga memungkinkan fitur *Web* menjadi sebuah sarana penyimpanan data dengan kapasitas yang luar biasa besar. Jika berbicara tentang *web 3.0* , maka erat hubungannya dengan konsep *Web Semantik*. Dimana *web* dapat dinikmati tidak hanya dalam bahasa asli pengguna, tetapi juga dalam bentuk format yang bisa diakses oleh agen-agen *software*. Beberapa ahli bahkan menamai *Web 3.0* ini sebagai *Web Semantik* itu sendiri.

1.1.3 Pengertian HTML dan CSS

1. Pengetian HTML

HTML (HyperText Markup Language) adalah suatu bahasa yang menggunakan tanda-tanda tertentu (*tag*) untuk menyatakan kode-kode yang harus ditafsirkan oleh *browser* agar halaman tersebut dapat ditampilkan secara benar. *HTML* digunakan untuk membuat dokumen elektronik (disebut halaman) yang ditampilkan di *World Wide Web (www)*. Setiap halaman berisi serangkaian koneksi ke halaman lain yang disebut *hyperlink*. Setiap halaman *web* yang Anda lihat di Internet ditulis menggunakan satu versi kode *HTML* atau yang lain.

2. Pengertian CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. *CSS* bukan merupakan bahasa pemrograman. Sama halnya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada

umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

1.2 Tujuan

Adapun tujuan dari Praktikum Aplikasi Komputer materi “*Website*” adalah sebagai berikut:

1. Untuk mengetahui pengertian dari *Website*, HTML5, dan CSS3.
2. Untuk mengetahui penggunaan HTML5, dan CSS3 dalam pembuatan *Website*.
3. Untuk mengetahui langkah-langkah dalam pembuatan *Website*.

1.3 Manfaat

Adapun manfaat dari praktikum Aplikasi Komputer materi “*Website*” adalah sebagai berikut:

1. Dapat memahami pengertian dari *Website*, HTML5, dan CSS3.
2. Dapat memahami penggunaan HTML5, dan CSS3 dalam pembuatan *Website*.
3. Dapat memahami langkah-langkah dalam pembuatan *Website*.

BAB II

METODOLOGI PRAKTIKUM

2.1 Waktu dan Tempat Praktikum

2.1.1 Waktu

Adapun waktu pelaksanaan Praktikum Aplikasi Komputer materi “*Website*” dilaksanakan tanggal 2 Desember 2022 pada pukul 16:00 WITA–selesai.

2.1.1 Tempat

Adapun tempat pelaksanaan Praktikum Aplikasi Komputer materi “*Website*” dilaksanakan secara *virtual* melalui Zoom dan LMS (*Learning Management System*) the E-Green Spada.

2.2 Alat dan Bahan

2.2.1 Alat

Adapun alat yang digunakan pada saat Praktikum Aplikasi Komputer materi “*Website*” adalah sebagai berikut:

Tabel 2. 1 Alat dan Fungsinya

No	Alat	Fungsi
1.	Laptop	Sebagai tempat dijalankannya <i>software</i> Visual Studio Code dan <i>browser</i>
2.	Visual Studio Code	Sebagai <i>software</i> untuk melakukan penulisan Bahasa pemrograman
	<i>Browser</i>	Sebagai <i>software</i> untuk melihat hasil tampilan <i>website</i>

2.2.2 Bahan

Adapun bahan yang digunakan pada saat Praktikum Aplikasi Komputer materi “*Website*” adalah sebagai berikut:

Tabel 2. 2 Bahan dan Fungsinya

No	Bahan	Fungsi
1.	Gambar	Sebagai objek yang akan dimasukkan ke dalam halaman <i>website</i>

2.3 Prosedur Praktikum

Adapun prosedur praktikum yang dilakukan dalam Praktikum Aplikasi Komputer materi “*Website*” adalah sebagai berikut:

- a. Siapkan alat dan bahan terlebih dahulu.
- b. Nyalakan laptop lalu buka Visual Studio Code.
- c. Operasikan *software* Visual Studio Code sesuai dengan langkah-langkah yang diajarkan selama praktikum.

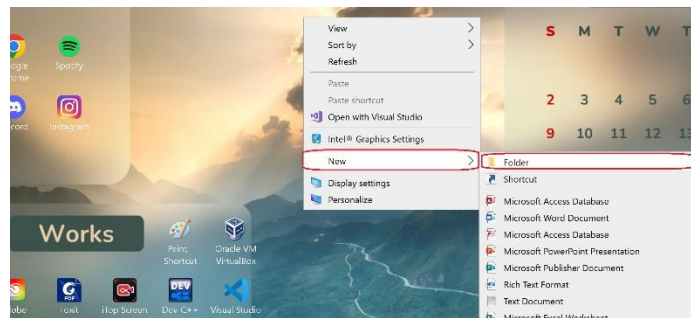
BAB III

ANALISIS DAN PEMBAHASAN

3.1 Hasil Praktikum

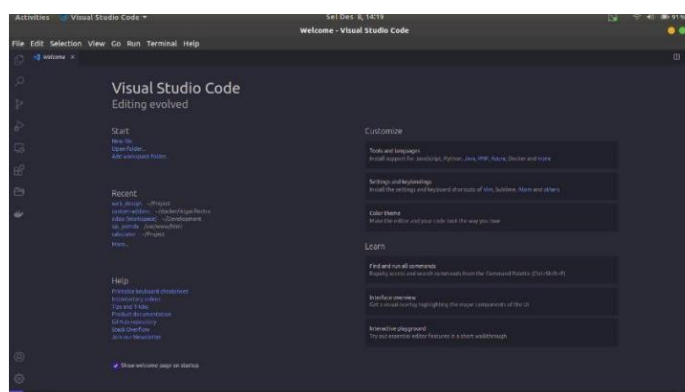
Adapun hasil praktikum yang dilakukan yaitu dapat mengetahui dan memahami tentang bagaimana cara mengoperasikan Visual Studio Code dengan mengembangkan *Website* berupa portfolio yang akan menampilkan informasi mengenai data pribadi kita serta menambahkan rangkuman materi Praktikum Aplikasi Komputer. Berikut cara mengembangkan *website*:

1. Pertama, membuat *new folder* di desktop kita beri nama *web design*, untuk menyimpan *file* gambar, kode HTML & CSS



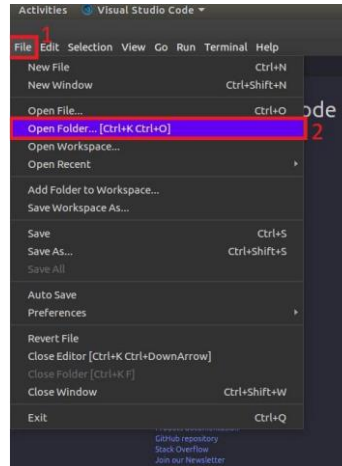
Gambar 3. 1 Tampilan membuat *New Folder*

2. Kemudian buka Visual Code Studio, yang sebelumnya telah menginstal *existence* terlebih dahulu.

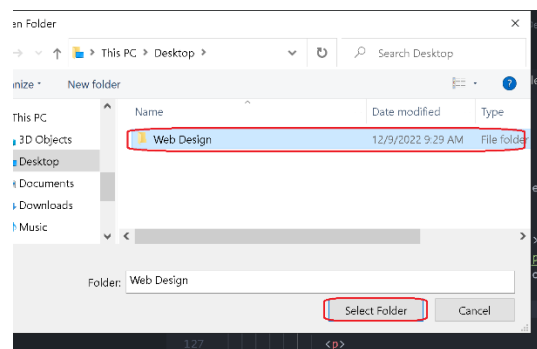


Gambar 3. 2 *Software* Visual Studio Code

3. Lalu pergi ke menu *file* untuk membuka *folder* yang kita buat tadi untuk menyimpan seluruh *file* HTML dan CSS.

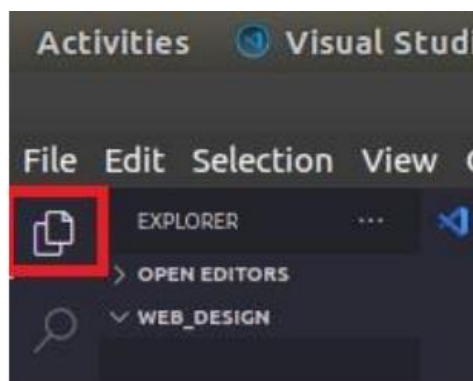


Gambar 3. 3 *Open Folder* pada Visual Studio Code



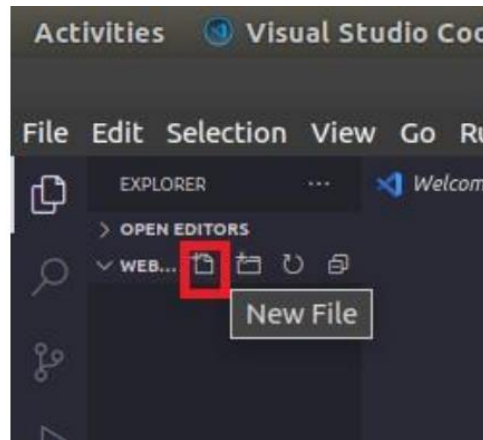
Gambar 3. 4 *Open Folder* pada Visual Studio Code

4. Kemudian aktifkan *sidebar* pada Visual Studio Code untuk melihat isi dari *folder* yang tadi kita buka.



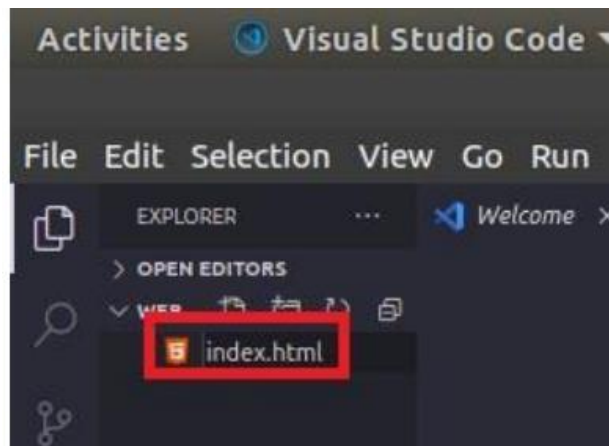
Gambar 3. 5 Mengaktifkan *sidebar* pada Visual Studio Code

5. Kemudian buat sebuah *file* HTML baru di dalam *folder* tersebut dengan cara menekan *icon new file* pada *sidebar folder*.



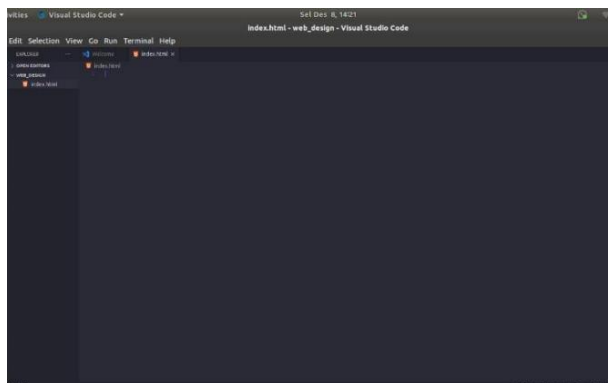
Gambar 3. 6 Membuat *file* baru pada Visual Studio Code

6. Kemudian ketikkan nama *file* sesuai yang kita inginkan (disini dibuat *file* index.html)



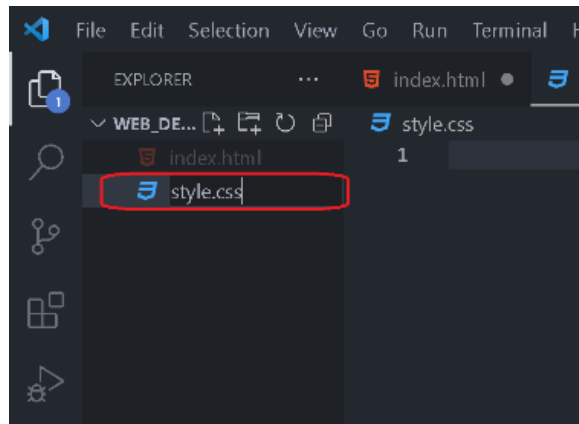
Gambar 3. 7 Membuat *file* index.html

7. Kemudian akan memunculkan tampilan sebagai berikut



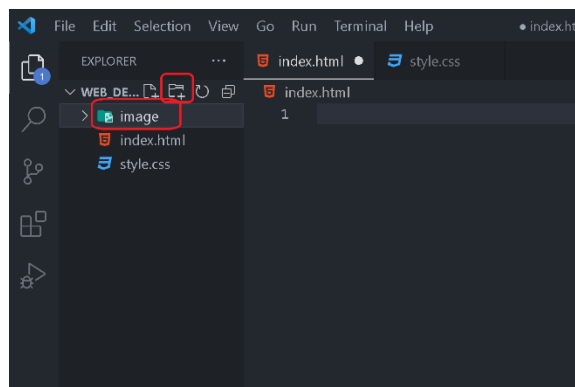
Gambar 3. 8 Tampilan *new file*

8. Kemudian untuk *file* CSS, tekan *new file* lagi dan masukkan nama *style.css*

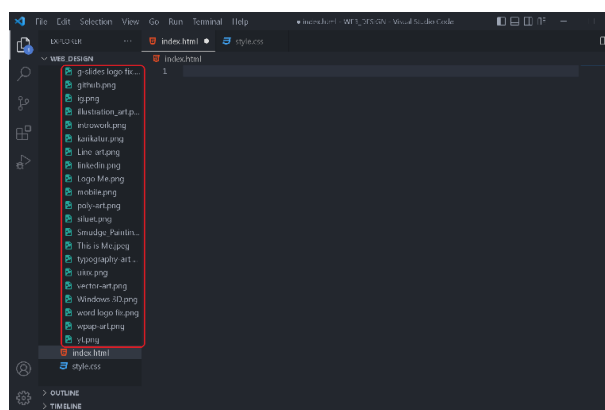


Gambar 3. 9 Membuat *file* *style.css*

9. Lalu membuat *folder image* untuk menyimpan gambar berupa logo untuk mempermudah menyalin *link relative path* gambar tersebut guna di-paste di *file* HTML. Sehingga bisa tertampil gambar di *website* kita.

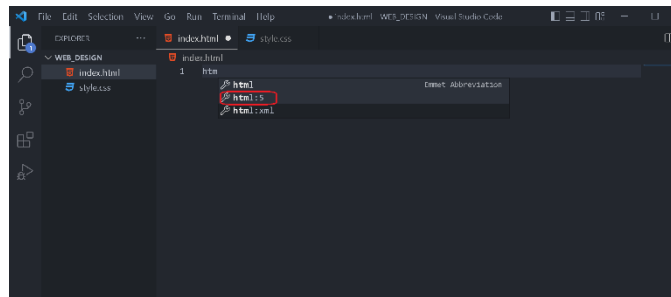


Gambar 3. 10 Membuat *folder image*

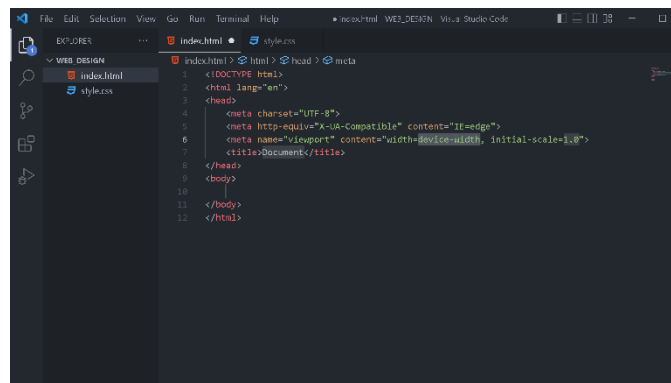


Gambar 3. 11 Tampilan gambar yang berhasil dimasukkan

10. Lalu menulis struktur standar dalam membuat sebuah *website* terlebih dahulu pada sebuah *file* HTML, dengan mengetikkan “html” (tanpa tanda kutip “”), lalu pilih “html:5” pada *snippet* yang muncul.

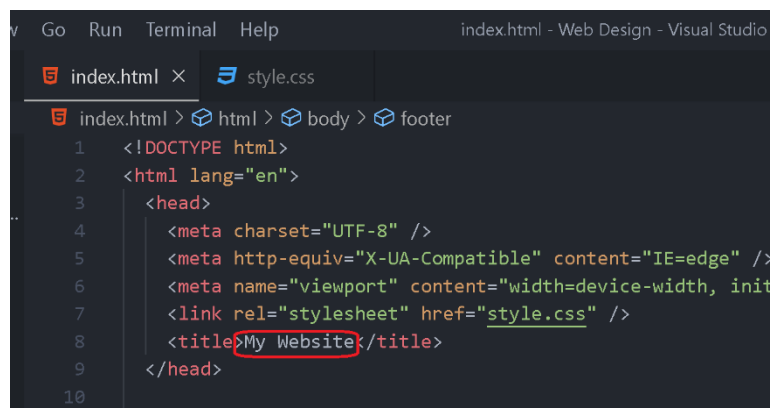


Gambar 3. 12 *Snippet* HTML



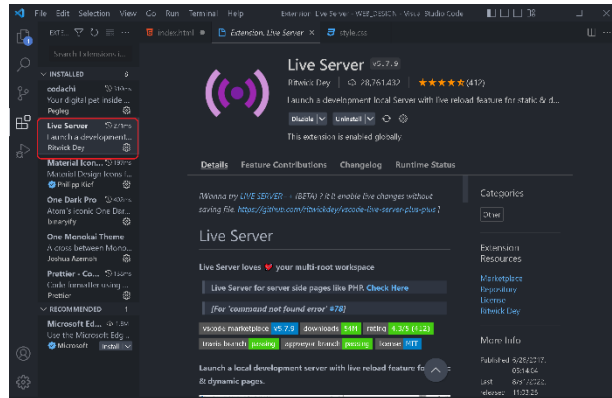
Gambar 3. 13 Struktur standar *file* HTML

11. Lalu, untuk mengedit *title* dari *website* yang akan kita buat, ubah *tag title* pada *file* *index.html*, disini kita mengubah *title*-nya menjadi “*My Website*”.



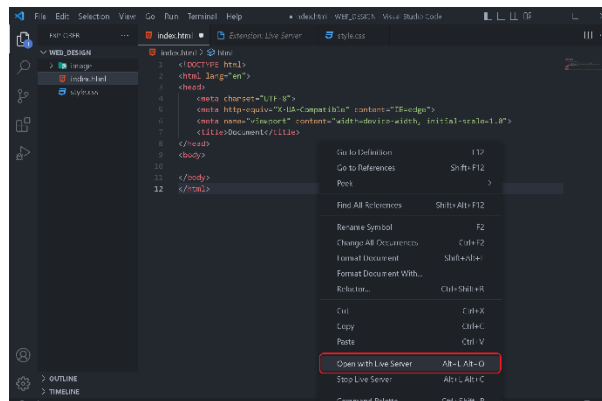
Gambar 3. 14 Mengubah *title* pada *website*

12. Kemudian untuk melihat hasil tampilan dari *file* HTML di atas yaitu menggunakan fitur *live server* yang disediakan oleh Visual Studio Code. Caranya pergi di menu *Extensions* dan cari dikolom pencarian dengan mengetikkan *Live Server*. Ketika sudah ditemukan tinggal *install* dan jalankan pada peranti Visual Studio Code.



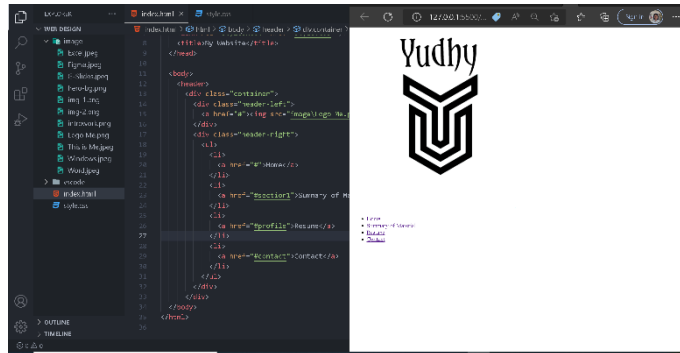
Gambar 3. 15 Menu *extension Live server*

13. Lalu, untuk yang menggunakan *Live Server*, klik kanan pada *file* *index.html*, lalu pilih “*Open with Live Server [Alt+L Alt+O]*” atau menggunakan *shortcut* *Alt+L Alt+O*.



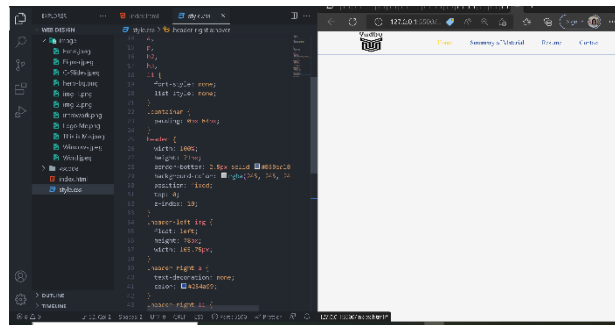
Gambar 3. 16 *Open with Live Server* pada Visual Studio Code

14. Kemudian mulailah membuat *website* dengan diawali dengan Membuat *navigation bar* sesuai yang diinginkan dan menambahkan logo kita pada *file* HTML.



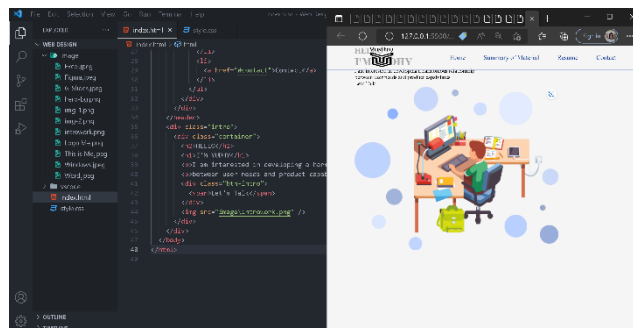
Gambar 3. 17 Membuat *navigation bar* pada HTML

15. Kemudian pergi ke CSS untuk mengatur tampilannya , baik itu jenis *font*, *margin*, dan lain hal. Lalu disini bisa ditambahkan efek ketika menekan salah satu *navigation bar* akan berwarna kuning.



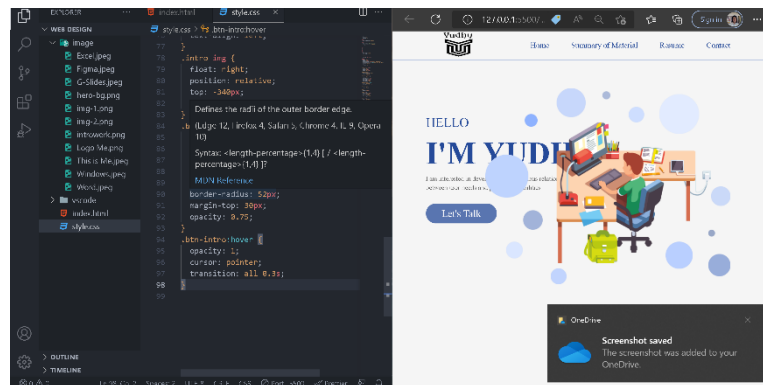
Gambar 3. 18 Mengedit tampilan *navigation bar* melalui CSS

16. Kemudian lanjut isi tampilan *home* dengan mendiskripsikan tentang diri kita dan menambahkan gambar animasi sesuai keinginan di HTML. Serta menambahkan tombol *button* untuk memperindah.



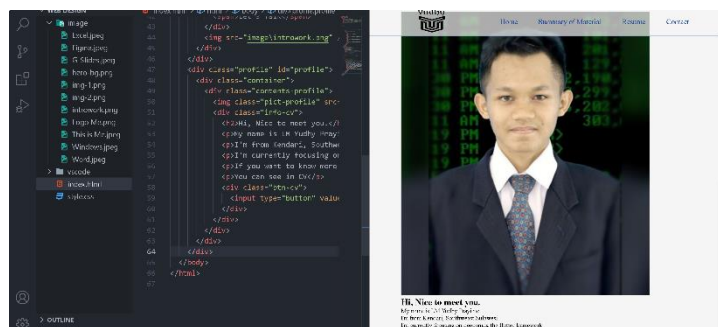
Gambar 3. 19 Membuat tampilan *home* di HTML

17. Kemudian pergi ke CSS untuk mengubah tampilan agar memindahkannya. Dengan cara menambahkan *margin*, merubah *color*.



Gambar 3. 20 Mengedit tampilan *home* di CSS

18. Lalu, menambahkan data dan gambar profil kita, selain itu tambah tombol *button* untuk melampirkan *resume* atau CV kita



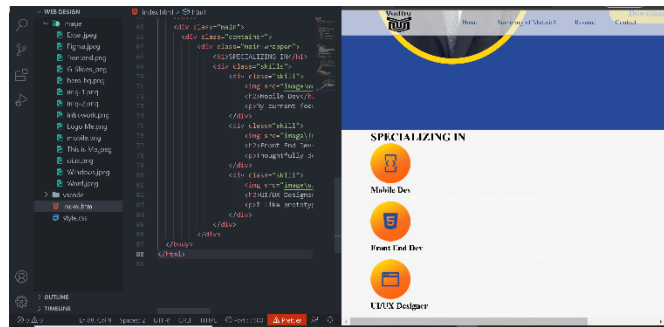
Gambar 3. 21 Membuat tampilan *resume* di HTML

19. Kemudian pergi ke CSS untuk mengaturnya, mulai dari gambar profil kita atur *shape*-nya jadi poligon, dan *margin* teksnya ada di tengah, dan mengatur *color*-nya.



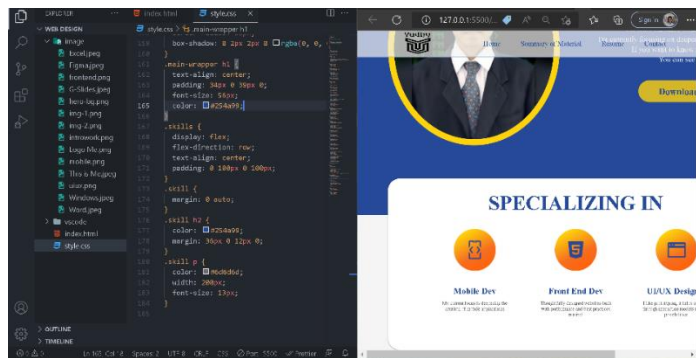
Gambar 3. 22 Mengedit tampilan *resume* di CSS

20. Kemudian memasukkan *skill* kita, serta menambahkan *image* agar dapat memperjelas tentang apa *skill* kita di HTML.



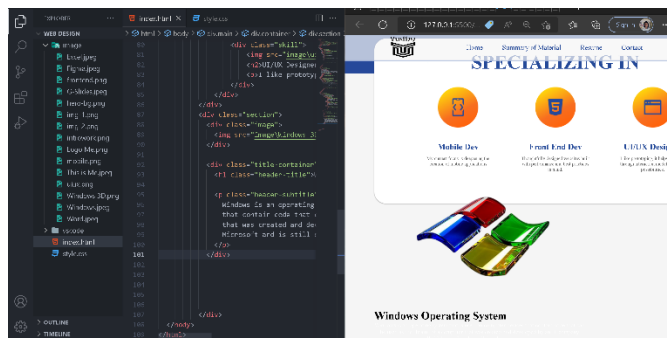
Gambar 3. 23 Mengedit tampilan *skill* kita di HTML

21. Kemudian karena saya ingin menempatkan di tengah halaman dan membuat border agar lebih menarik maka di tambahkan beberapa kode di CSS.



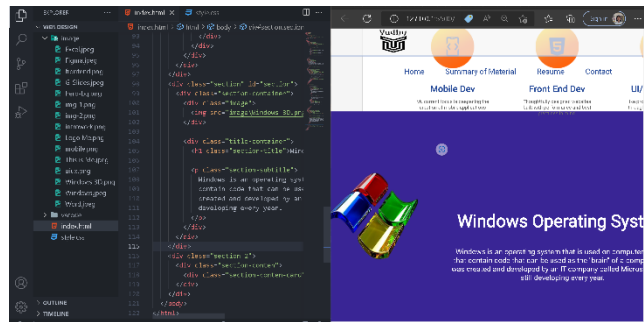
Gambar 3. 24 Mengedit tampilan *skill* di CSS

22. Setelah itu, masuk kembali lagi ke HTML untuk menambahkan tampilan materi sistem operasi Windows. Dengan memasukkan beberapa kode dan menambahkan logo Windows.



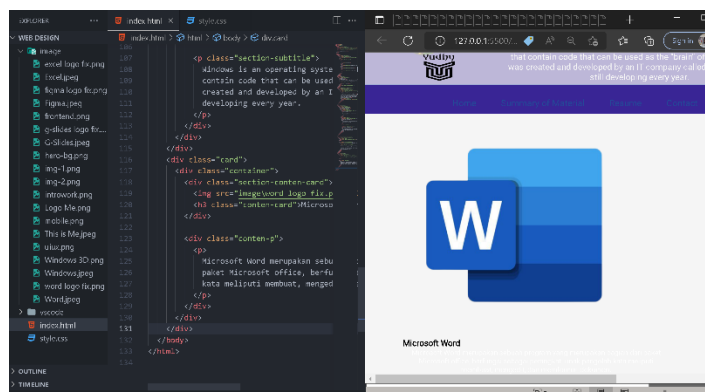
Gambar 3. 25 Membuat rangkuman materi Windows di HTML

23. Kemudian masuk ke CSS untuk mengatur tampilannya sesuai keinginan kita.



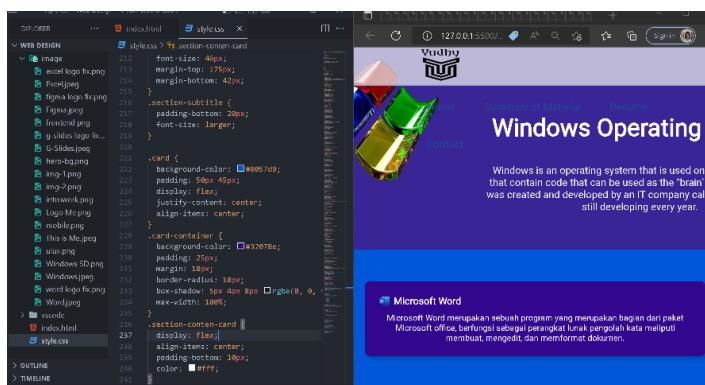
Gambar 3. 26 Mengedit tampilan materi Sistem operasi Windows di CSS

24. Kemudian memasukan materi mengenai Word, di HTML dengan memasukan *code-code* yang diberi nama *card-continer* agar mempermudah untuk melihat *code* tersebut di HTML, serta menambahkan logo word.



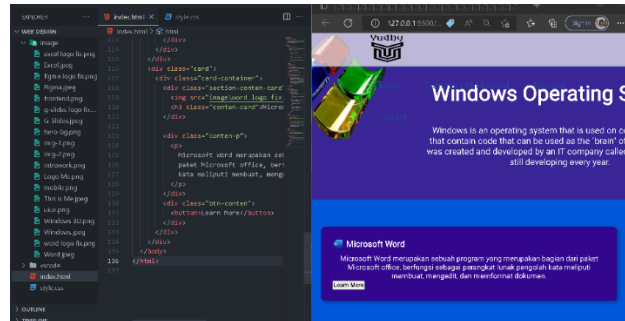
Gambar 3. 27 Memasukan materi mengenai Word di HTML

25. Kemudian masuk pada tampilan CSS untuk mengatur letak dan warna teks.



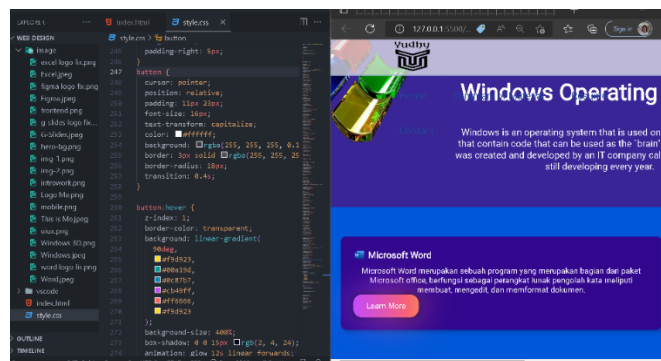
Gambar 3. 28 Mengedit tampilan Word di CSS

26. Kemudian disini, memasukkan tombol button laern more agar menampilkan halaman lain yang menjelaskan lebih rinci mengenai Word, yaitu dengan cara ke HTML tambahkan *code button*.



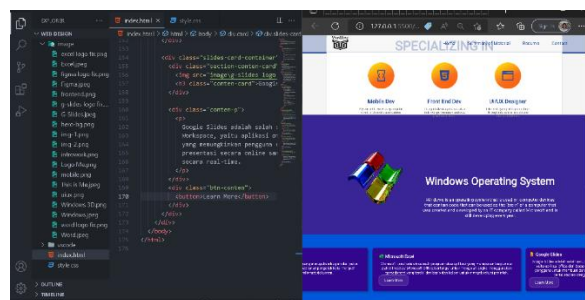
Gambar 3. 29 Menambahkan tombol *button* pada materi Word di HTML

27. Setelah mengisi kode *button* di HTML kemudian beralih ke CSS untuk mengatur tampilannya, dengan menambahkan efek transisi yang sesuai dengan keinginan kita.



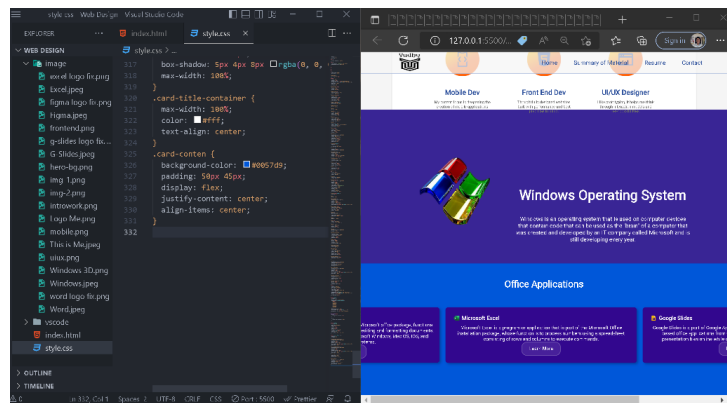
Gambar 3. 30 Mengedit tampilan tombol *button* di CSS

28. Kemudian memasukan materi mengenai Excel,Google Slides, di HTML dengan cara menyalin kode *card-container* pada Word tadi, sehingga layar menampilkan 3 sekaligus materi. Setelah itu tinggal mengganti logo dan ringkasan materinya.



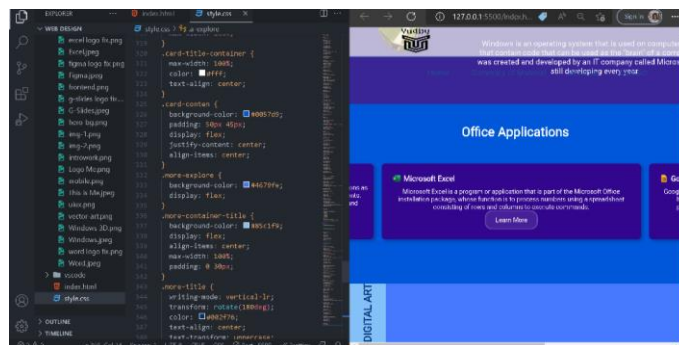
Gambar 3. 31 Memasukan materi Excel dan Google Slides di HTML

29. Menambahkan judul Aplikasi Perkantoran di HTML dan mengedit tampilan berada di tengah atas melalui CSS.



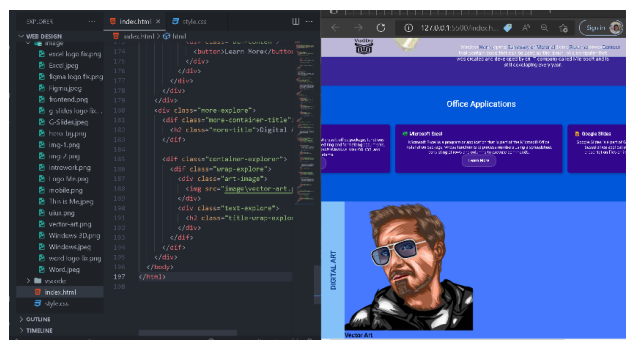
Gambar 3. 32 Membuat judul Aplikasi Perkantoran di HTML

30. Setelah itu, menambahkan materi desain grafis dengan cara buat *section* baru di HTML. Pertama membuat judul *digital art*, lalu mengatur tampilannya di CSS menempatkan di sudut kiri. Serta mengatur *background* sesuai keinginan.



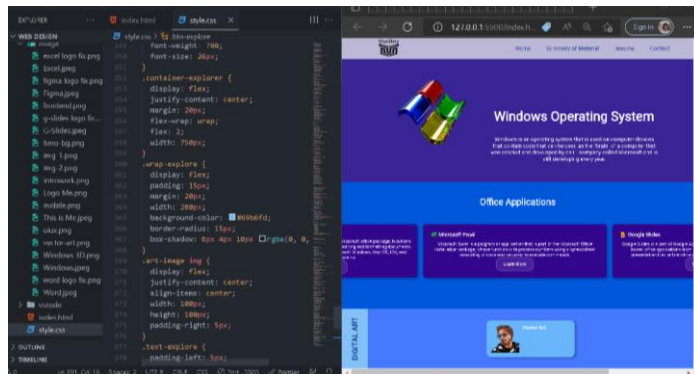
Gambar 3. 33 Membuat judul *digital art*

31. Setelah itu, memasukan materi mengenai desain grafis yaitu jenis-jenis *digital art*. Disini kita masukan *vector art*, dan menambahkan gambarnya di HTML.



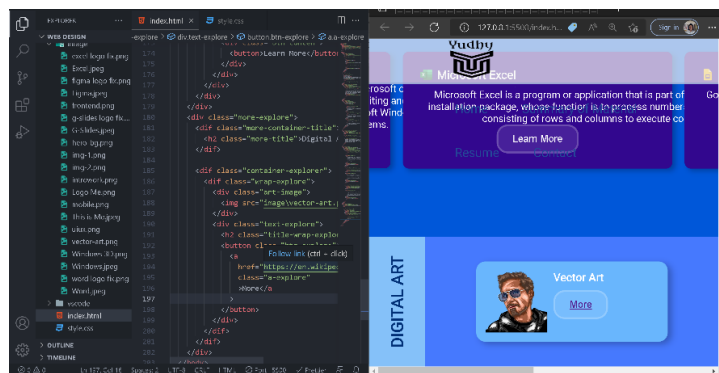
Gambar 3. 34 Memasukan nama *vector art* serta gambarnya di HTML

32. Kemudian lanjut pada CSS untuk mengatur tampilan dan ukurannya sesuai keinginan.



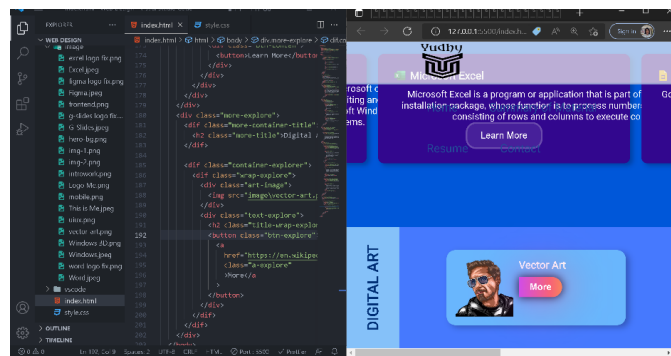
Gambar 3. 35 Mengatur tampilan dan ukuran *vector art* di CSS

33. Kemudian untuk menambahkan penjelasan mengenai *vector art*, bisa menggunakan tombol *button more* yang akan mengarahkan kita ke halaman yang menjelaskan *vector art*. Dengan cara menambahkan kode *button* di HTML.



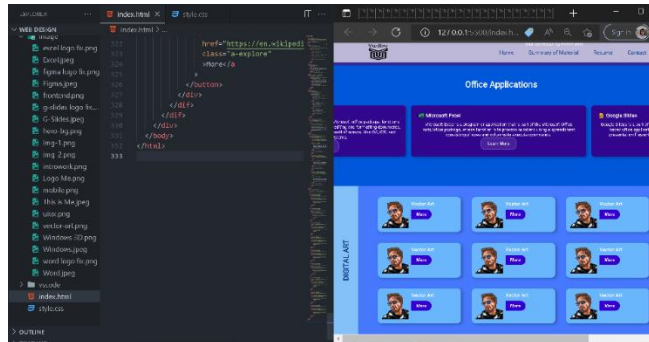
Gambar 3. 36 Menambahkan tombol *button* di border *vector art*

34. Setelah itu mengatur bentuk tampilannya dan menambahkan efek sesuai keinginan kita di CSS untuk tampilan *button*.



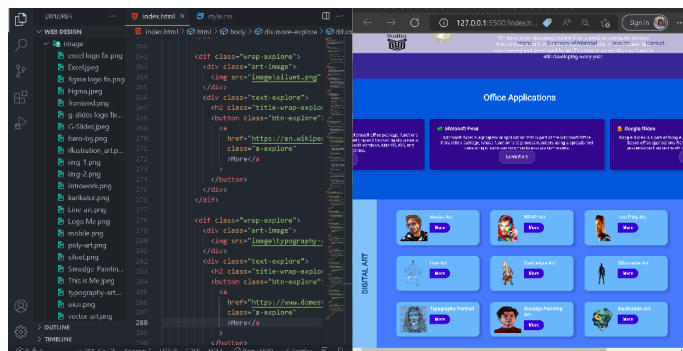
Gambar 3. 37 Mengedit tampilan *button* di CSS

35. Setelah terbentuk, kembali pada tampilan HTML untuk menambahkan jenis-jenis *digital art* lainnya, sengan cara *copy* dan *paste code vector art* tadi sebanyak jenis *digital art* yang diinginkan.



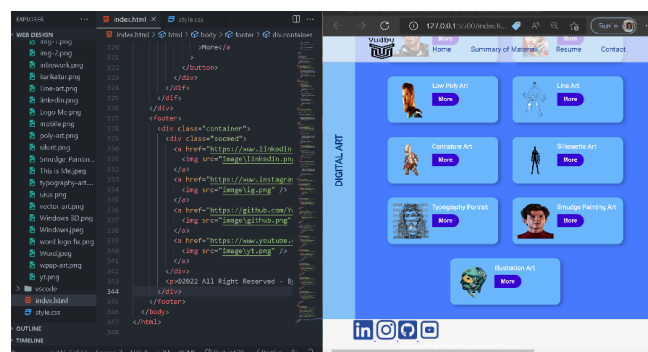
Gambar 3. 38 Menyalin border *vector art*

36. Kemudian tinggal mengganti nama *digital art*-nya serta gambarnya dan *link button more*-nya disesuaikan di HTML.



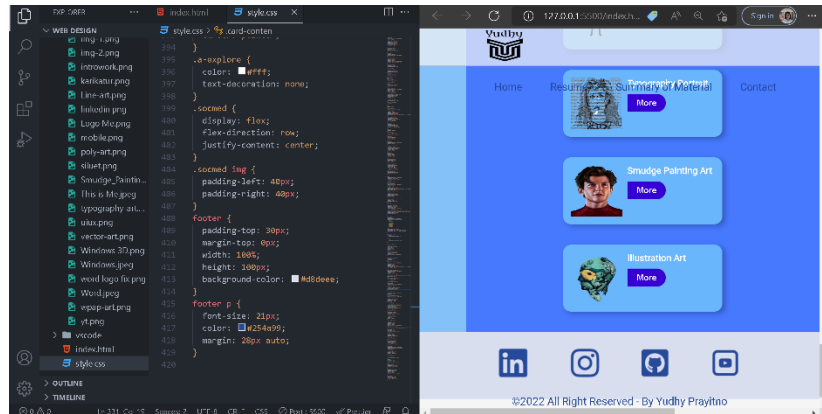
Gambar 3. 39 Menyesuaikan tampilan dengan jenis *digital art* lainnya

37. Kemudian yang terakhir yaitu bagian *footer* dengan menambahkan mengenai sosial media kita yaitu berupa gambar logo aplikasi dan disertai *link*-nya agar mengarahkan pengguna ke profil sosial media kita ketika mengkliknya.



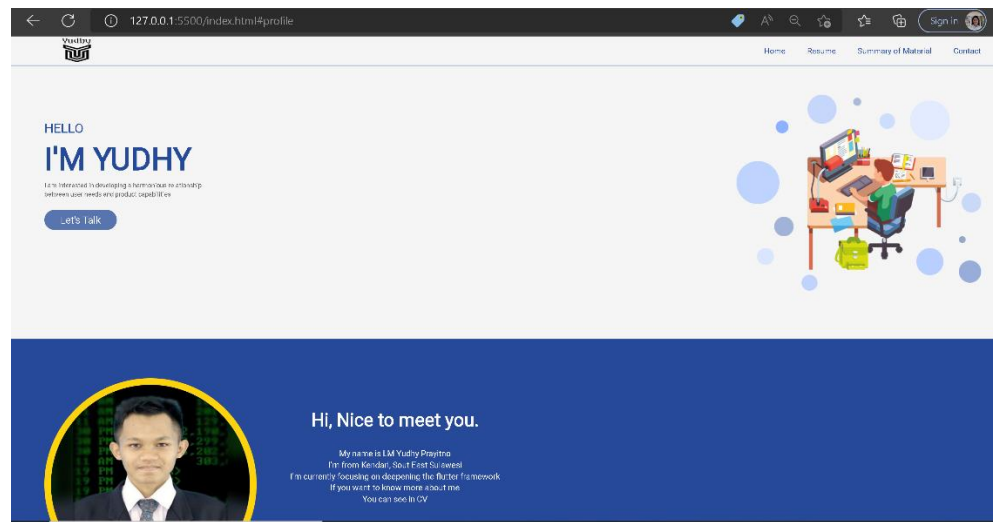
Gambar 3. 40 Menambahkan *contact* kita di bagian *footer* HTML

38. Lalu beralih pada css untuk mengedit tampilan *footer*-nya.

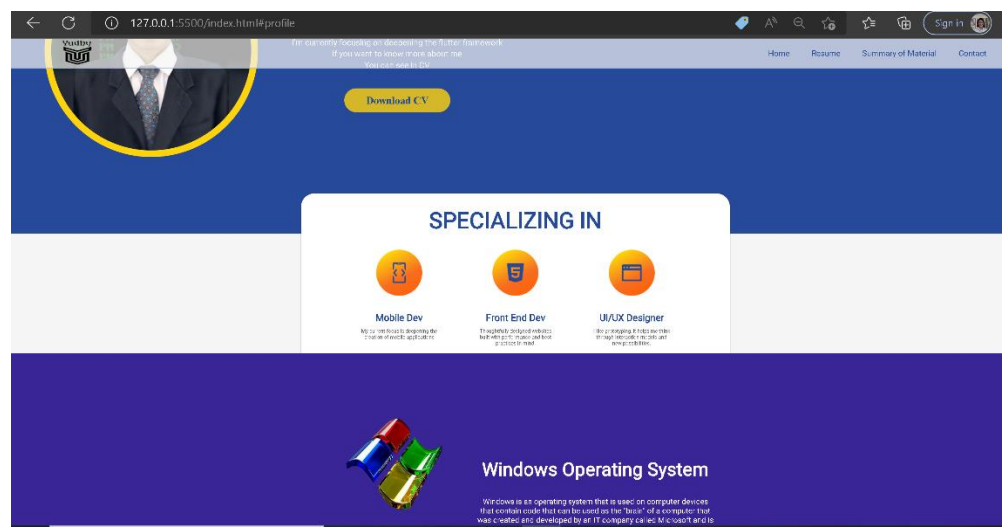


Gambar 3. 41 Mengatur tampilan *contact* di CSS

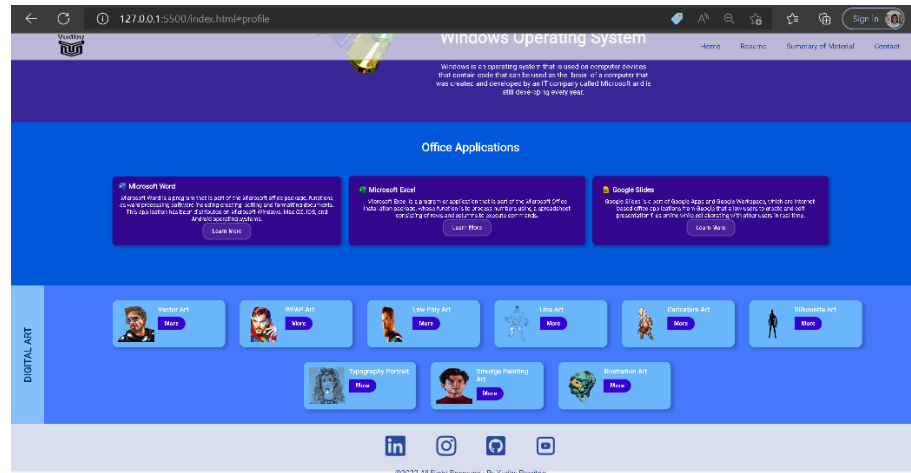
39. Berikut adalah tampilan *website* yang telah dibuat.



Gambar 3. 42 Tampilan *website* portfolio



Gambar 3. 43 Tampilan *website* portfolio



Gambar 3. 44 Tampilan *website* portfolio

3.2 Analisis Dan Pembahasan

3.2.1 Penggunaan HTML

Penggunaan HTML tidak luput dari *tag-tag* HTML itu sendiri. *Tag* html merupakan sebuah kode yang digunakan untuk mendefinisikan sebuah elemen pada html, apakah elemen itu nantinya akan ditampilkan dalam bentuk judul, paragraf, *quote* dan lain sebagainya. Ada banyak sekali *tag-tag* pada HTML, fungsinya pun bermacam. Yang pada intinya semua *tag* tersebut digunakan untuk mendefinisikan sebuah elemen pada HTML, apakah dia berupa paragraf, tabel dan yang lainnya.

Dalam HTML, terdapat struktur standar dalam membuat sebuah *website*, berikut struktur standar pada sebuah *file* HTML.

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Web Design</title>
8    </head>
9    <body></body>
10 </html>
11

```

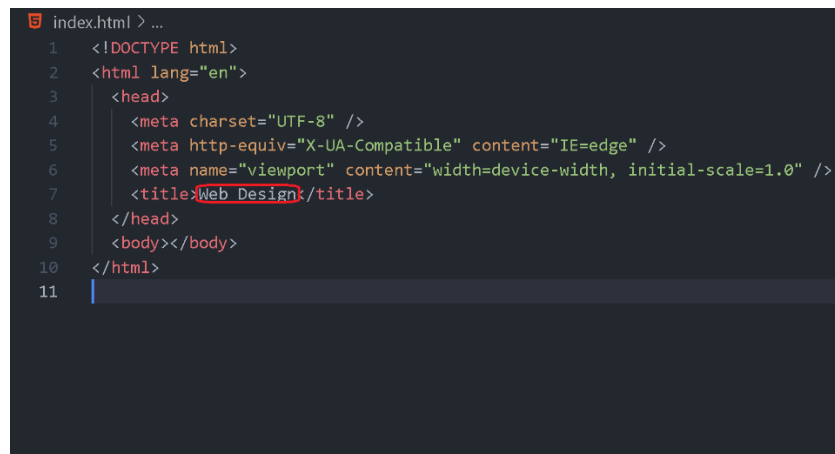
Gambar 3. 45 . Struktur standar *file* HTML

Pada struktur standar tersebut, baris 1 merupakan *tag* untuk informasi pada *browser* bahwa file HTML ini adalah *file* HTML *version* 5. Untuk baris 2 merupakan *tag* pembuka HTML dan baris 10 (baris terakhir) adalah *tag* penutup HTML. Baris 3 adalah *tag* pembuka *head* dan baris 6 adalah *tag*

penutupnya. *Tag head* memiliki fungsi untuk mendefinisikan *file* html yang kita buat. Pada *tag* ini, dapat berisi informasi tentang *author*, *file* CSS atau *style* yang digunakan, *title website*, dan lain sebagainya. Baris 5 adalah *tag title* yang berisi informasi *title* untuk hasil *website* dari *file* tersebut. Baris 7 adalah *tag* pembuka *body* dan baris 9 adalah penutupnya. Pada *tag body* inilah kita akan menuliskan *tag-tag* HTML yang nantinya akan membuat tampilan *website* kita.

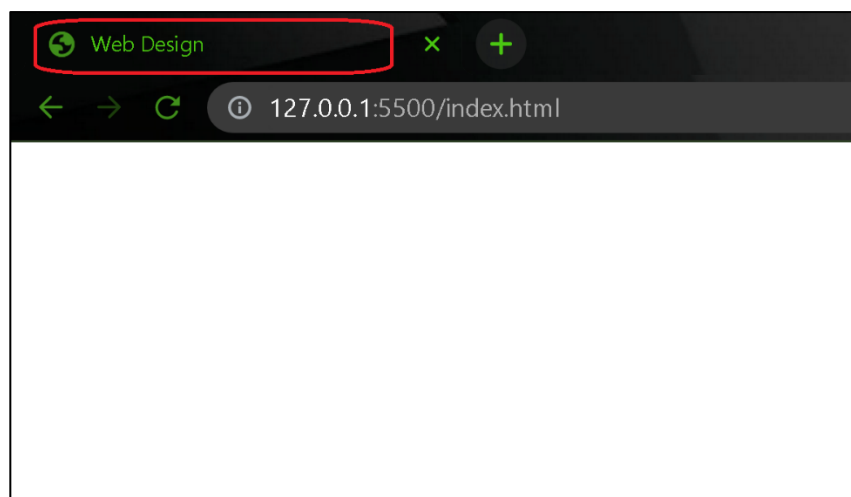
Tag HTML umumnya terdapat *tag* pembuka () dan *tag* penutup (). Tetapi ada beberapa juga *tag* HTML yang tidak memiliki *tag* penutup.

Untuk mengedit *title* dari *website* yang akan kita buat, ubah *tag title* pada *file* index.html, disini kita mengubah *title*-nya menjadi “Web Design”.



```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Web Design</title>
8   </head>
9   <body></body>
10 </html>
11
```

Gambar 3. 46 Ubah *tag title* pada *website*



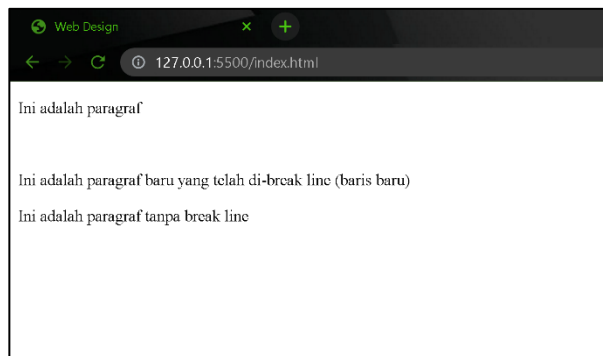
Gambar 3. 47 Hasil perubahan *title* pada *website*

Kemudian untuk membuat paragraf pada *website*, kita menggunakan *tag* p, dan untuk membuat baris baru (*break line*) kita menggunakan *tag* br (*tag* br adalah salah satu contoh *tag* yang tidak memiliki *tag* penutup).



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Web Design</title>
8   </head>
9   <body>
10    <p>Ini adalah paragraf</p>
11    <br>
12    <p>Ini adalah paragraf baru yang telah di-break line (baris baru)</p>
13    <p>Ini adalah paragraf tanpa break line</p>
14  </body>
15 </html>
```

Gambar 3. 48 *Tag* p dan br pada HTML



Gambar 3. 49 Hasil tampilan pada *browser*

Biasanya dalam sebuah *website* kita menjumpai adanya *link* yang bila kita klik akan mengarahkan kita ke halaman yang di arahkan *link* tersebut. Untuk membuat sebuah *link*, kita menggunakan *tag* a, kemudian mengisi alamat halaman atau *website* yang kita ingin tuju ketika *link* tersebut di klik(dalam contoh ketika diklik akan mengarah ke halaman www.facebook.com).



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Web Design</title>
8   </head>
9   <body>
10    <a href="http://www.facebook.com">Ini link untuk menuju ke halaman www.facebook.com</a>
11  </body>
12 </html>
```

Gambar 3. 50 *Tag* a pada HTML



Gambar 3. 51 Tampilan pada *browser*

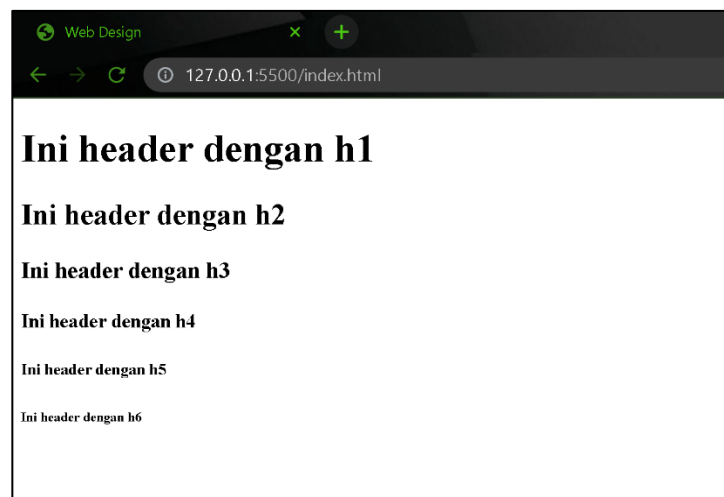
Untuk membuat sebuah *header*, kita menggunakan *tag* h1, h2, h3, h4, h5, atau h6 (h1 memiliki ukuran *font* terbesar dan h6 yang terkecil).

```

index.html • style.css
index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Web Design</title>
8    </head>
9    <body>
10     <h1>Ini header dengan h1</h1>
11     <h2>Ini header dengan h2</h2>
12     <h3>Ini header dengan h3</h3>
13     <h4>Ini header dengan h4</h4>
14     <h5>Ini header dengan h5</h5>
15     <h6>Ini header dengan h6</h6>
16   </body>
17 </html>
18

```

Gambar 3. 52 *Tag header*

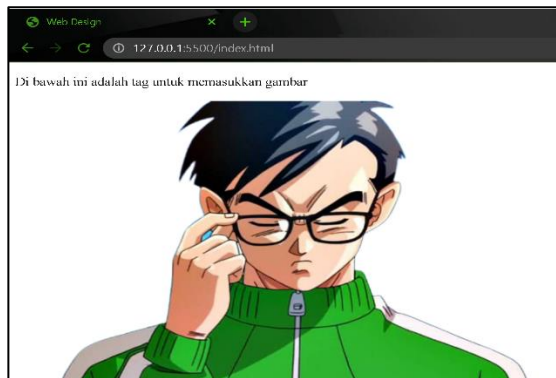


Gambar 3. 53 Tampilan pada *browser*

Untuk memasukkan gambar dalam sebuah *website*, kita menggunakan *tag* `img`, kemudian menuliskan nama gambar yang akan kita masukkan ke dalam *website* tersebut pada `src`.

```
index.html > html > body > img
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Web Design</title>
8   </head>
9   <body>
10    <p>Di bawah ini adalah tag untuk memasukkan gambar</p>
11    
12  </body>
13 </html>
```

Gambar 3. 54 *Tag* `img`

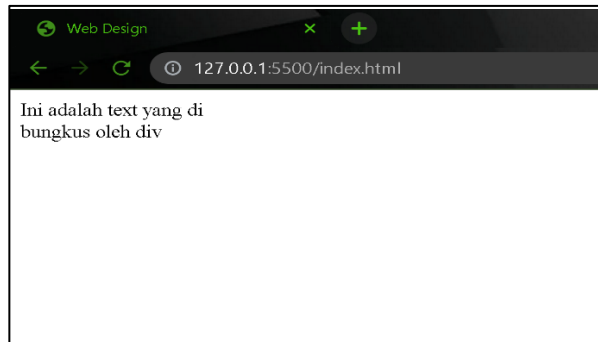


Gambar 3. 55 Tampilan pada *browser*

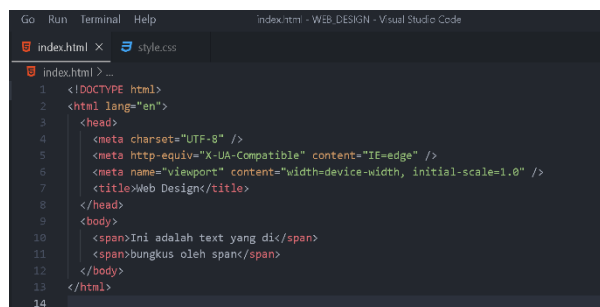
Pada HTML, kita dapat mengelompokkan *tag-tag* yang telah kita buat, ada dua cara untuk mengelompokkan *tag-tag* tersebut, yaitu menggunakan *tag* `div` atau *tag* `span`. Ketika kita menggunakan *tag* `div`, maka antara kelompok tersebut akan ditampilkan secara *vertical* (atas ke bawah), sedangkan *tag* `span` akan ditampilkan *horizontal* (kiri ke kanan).

```
Go Run Terminal Help index.html - WEB DESIGN - Visual Studio Code
index.html x style.css
index.html > html > body > div
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Web Design</title>
8   </head>
9   <body>
10    <div>Ini adalah text yang di</div>
11    <div>bungkus oleh div</div>
12  </body>
13 </html>
```

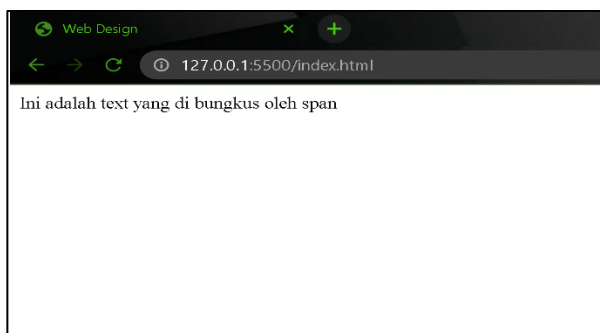
Gambar 3. 56 *Tag* `div`



Gambar 3. 57 Tampilan pada *browser*

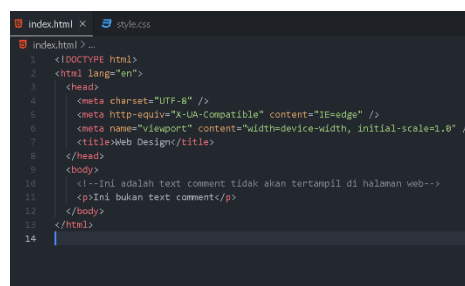


Gambar 3. 58 Tag *span*

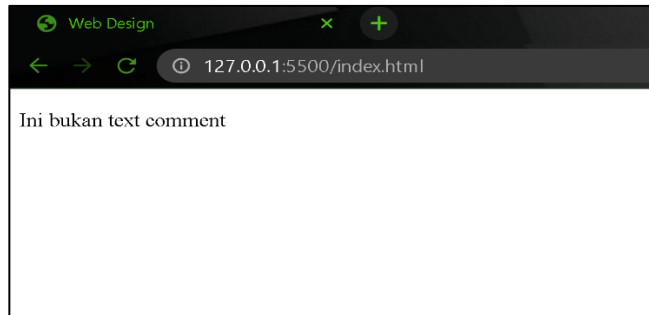


Gambar 3. 59 Tampilan pada *browser*

Kita dapat menuliskan komentar pada *file* HTML, dan komentar tersebut tidak akan tertampil pada tampilan di *browser*. Kegunaan komentar ini adalah dapat memudahkan kita untuk meninggalkan komentar atau sebagai penjelasan mengenai apa yang telah kita buat pada *file* HTML.



Gambar 3. 60 Komentar pada *file* HTML



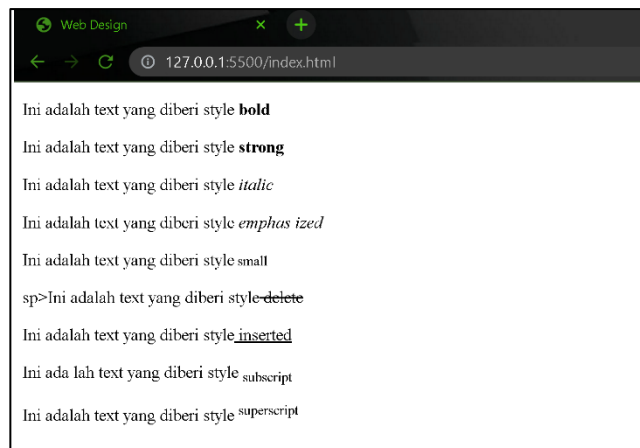
Gambar 3. 61 Tampilan pada *browser*

Pada HTML terdapat juga pemberian *style* berupa *italic*(miring) menggunakan *tag i*, *bold*(tebal) menggunakan *tag b* dan lain-lain.

```

Go Run Terminal Help index.html - Web Design - Visual Studio Code
index.html x style.css
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>Web Design</title>
8 </head>
9 <body>
10  <p>Ini adalah text yang diberi style<b> bold</b></p>
11  <p>Ini adalah text yang diberi style<strong> strong</strong></p>
12  <p>Ini adalah text yang diberi style<i> italic</i></p>
13  <p>Ini adalah text yang diberi style<em> emphas ized</em></p>
14  <p>Ini adalah text yang diberi style<small> small</small></p>
15  <p>Ini adalah text yang diberi style<del> delete</del></p>
16  <p>Ini adalah text yang diberi style<ins> inserted</ins></p>
17  <p>Ini ada lah text yang diberi style <sub>subscript</sub></p>
18  <p>Ini adalah text yang diberi style <sup>superscript</sup></p>
19  <script src="script.js"></script>
20 </body>
21 </html>
22
  
```

Gambar 3. 62 *Tag formatting* pada HTML



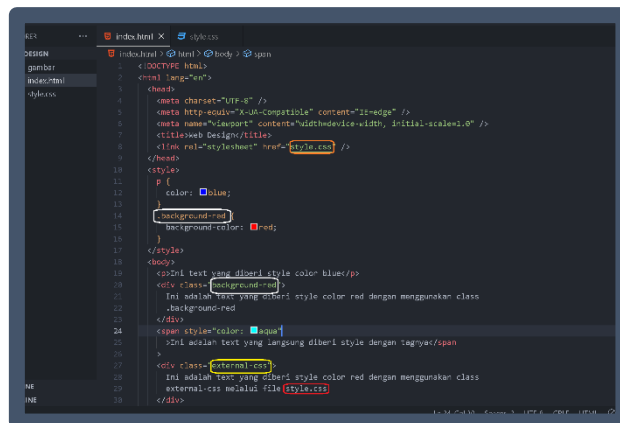
Gambar 3. 63 Tampilan pada *browser*

3.2.2 Penggunaan CSS

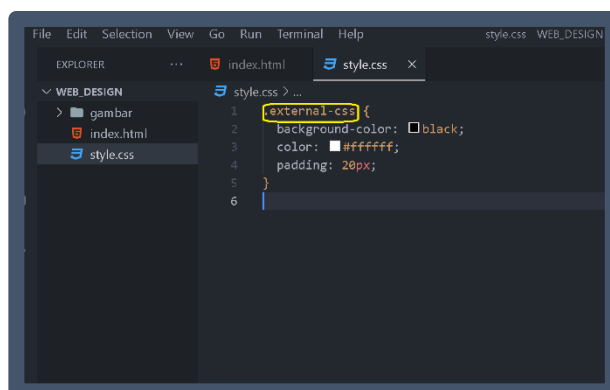
Dalam penggunaannya CSS digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa *markup*, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. Pada gambar 3.62

di HTML kita juga dapat memberi *style* pada *website* menggunakan *file* CSS. Ada 3 cara untuk memberi *style* menggunakan CSS, yaitu:

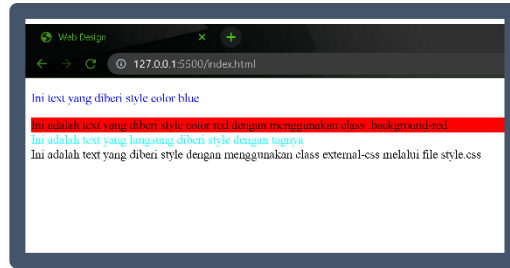
1. Memberi *style* langsung pada tag HTML dengan menggunakan atribut *style*. Pada gambar 3. 64, pemberian *style* dengan cara ini dilakukan pada *tag span*(baris 24).
2. Memberi *style* menggunakan *atribut class* yang *styling*-nya didefinisikan dalam *file* HTML. Pada gambar 3. 61 dicontohkan dengan *tag div* yang dikotakkan dengan kotak putih. Pemberian nama *class*-nya adalah “*background-red*” pada HTML, dan ketika mendefinisikan *style*-nya pada CSS, nama *class*-nya ditambahkan “.” didepan sehingga menjadi “*.background-red*”.
3. Memberi *style* menggunakan *atribut class* yang *styling*-nya didefinisikan terpisah pada *file* CSS, yang kemudian dihubungkan ke *file* HTML yang diinginkan. Pada gambar 3. 64, dicontohkan dengan *class* “*external-css*” yang kemudian didefinisikan pada *file* *style.css*.



Gambar 3. 64 *Styling file* HTML menggunakan CSS

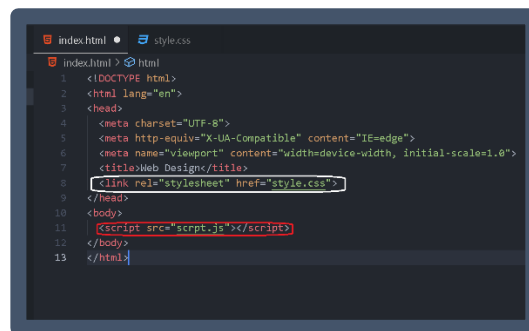


Gambar 3. 65 Pemberian *style* menggunakan *file* *style.css*



Gambar 3. 66 Tampilan pada *browser*

Untuk menghubungkan *file* HTML dengan CSS menggunakan *tag link* yang dituliskan antara *tag head* (pada gambar 3. 67, seperti kotak putih), dengan atribut *href* diisi nama *file* CSS yang akan dihubungkan.

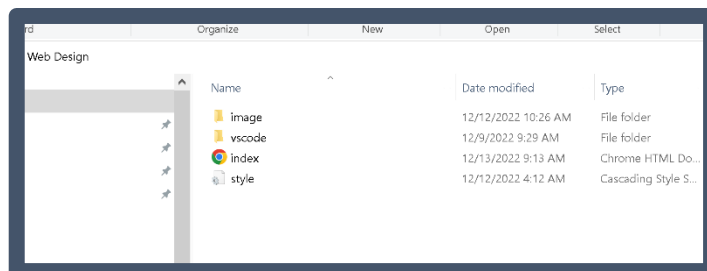


Gambar 3. 67 Menghubungkan *file* CSS pada *file* HTML

3.2.3 Penggunaan *Hosting Website*

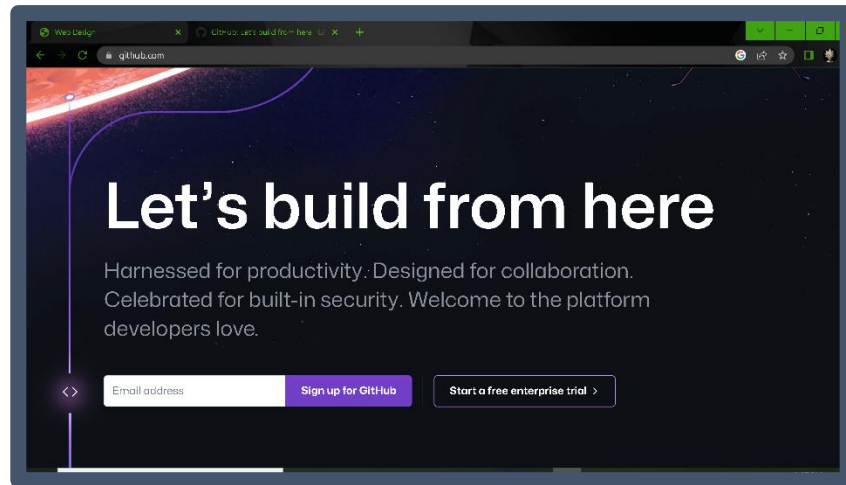
Penggunaan *hosting website* digunakan untuk menampung data-data yang diperlukan oleh sebuah *website* dan sehingga dapat diakses lewat Internet. Data disini dapat berupa *file*, gambar, *email*, aplikasi/program/*script* dan *database*. Adapun langkah-langkah untuk meng-*hosting web*, adalah sebagai berikut :

1. Siapkan *file web* dan *file media* yang ingin di-*upload*. Untuk nama *file web*, disarankan untuk menggunakan nama “*index.html*”. Agar pada saat pengaksesan, langsung menuju ke *file index*.



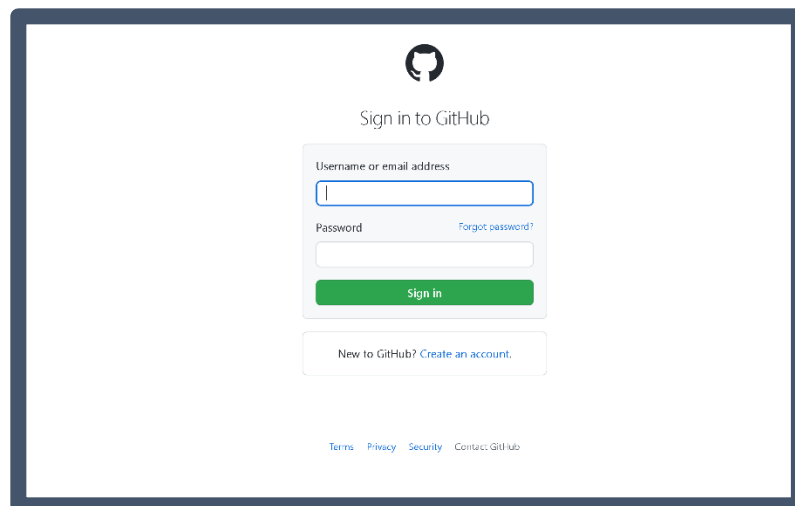
Gambar 3. 68 Tampilan *file* yang dibutuhkan dalam *website*

2. Mengakses tautan Github



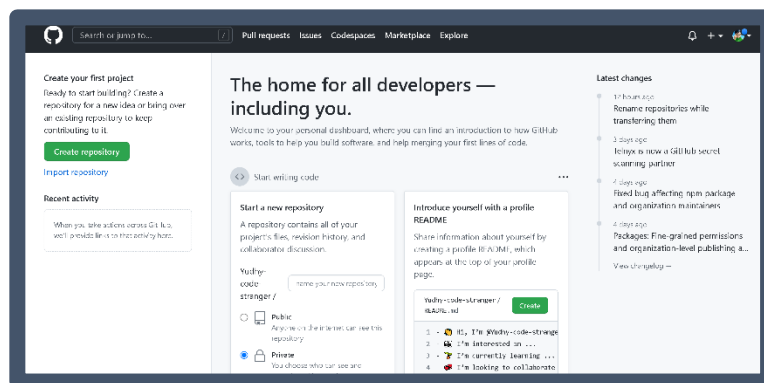
Gambar 3. 69 Tampilan *home* Github

3. Klik Sign In untuk masuk ke akun, atau klik Sign up untuk membuat akun jika belum memiliki akun



Gambar 3. 70 Tampilan *menu sign*

4. Klik tombol *Create repository* untuk membuat *repository* baru



Gambar 3. 71 Membuat *repository*

5. Lengkapi nama *repository* kemudian simpan

to...

Pull requests

Issues

Codespaces

Marketplace

Explore

🔔

+

👤

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

Repository name *

👤

Yudhy-code-stranger ▾

/

TugasWebsiteAplikom 22

✓

Great repository names are short and **your new repository will be created as TugasWebsiteAplikom-22**. [Learn more.](#)

Description (optional)

☒

🔓

Public

Anyone on the internet can see this repository. You choose who can commit.

☐

🔒

Private

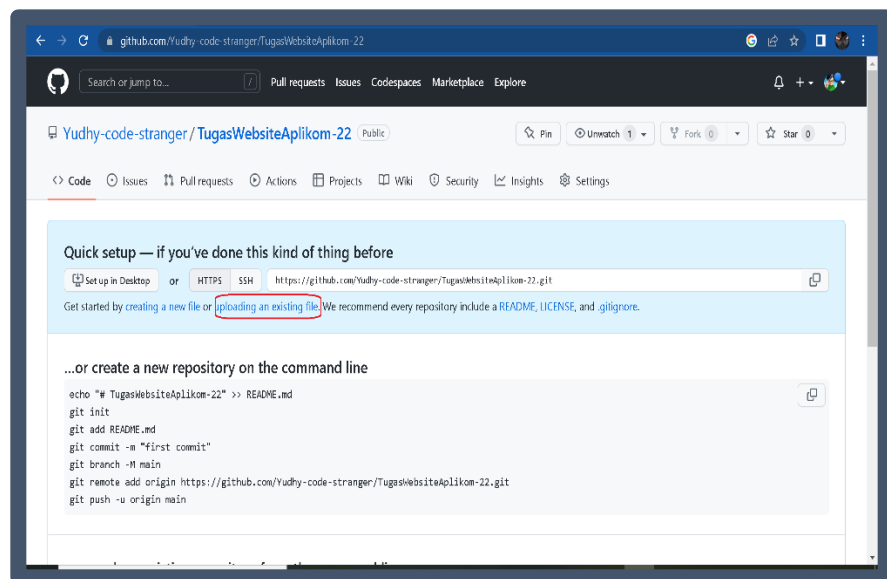
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ Add a README file

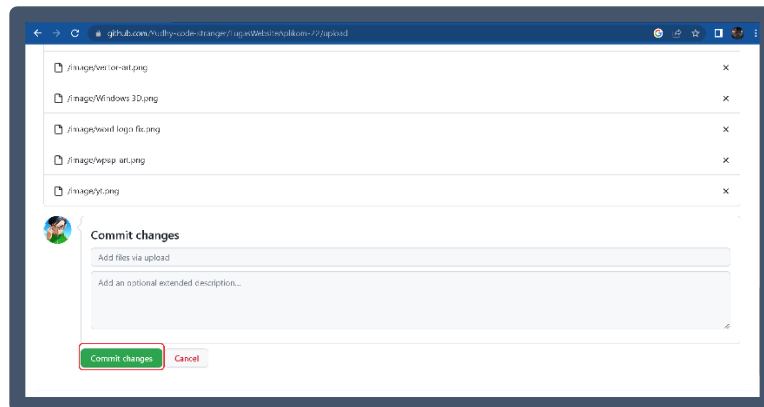
Gambar 3. 72 Memasukan nama *repository*

6. Jika *repository* telah dibuat, unggah *file* yang telah disiapkan dengan klik pada tulisan *upload an existing file*



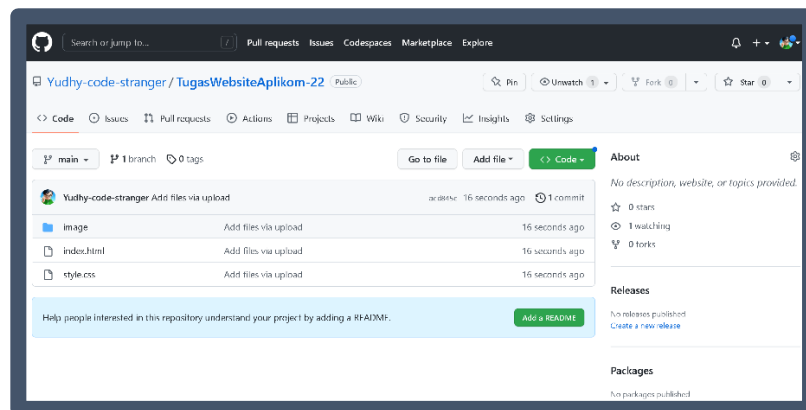
Gambar 3. 73 Meng-*upload file*

7. Pilih *file* yang telah disiapkan kemudian unggah dengan mengklik tombol *Commit changes*



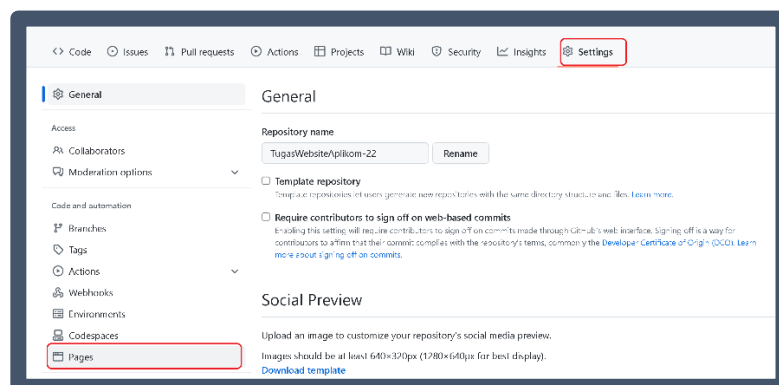
Gambar 3. 74 Mengunggah *file*

8. *File* yang diunggah berhasil tersimpan di *repository*.



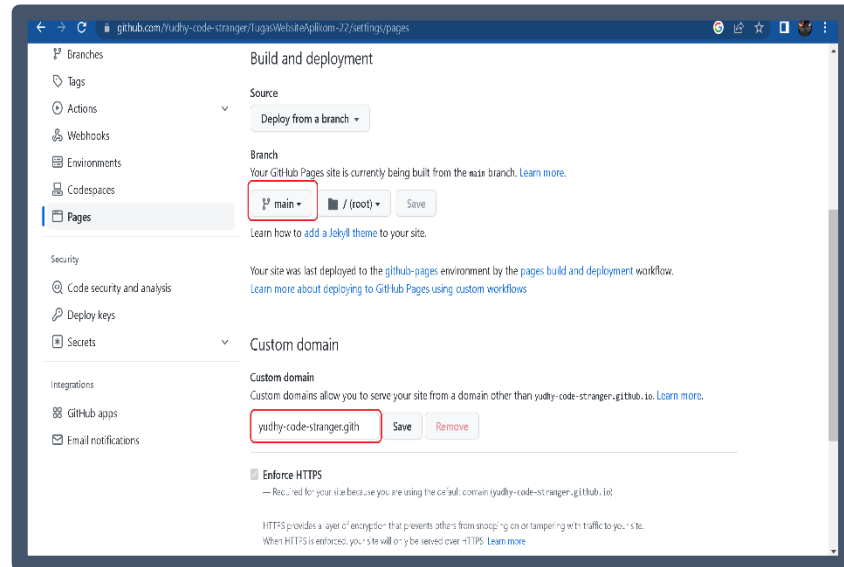
Gambar 3. 75 Tampilan *file* yang berhasil diunggah

9. Kemudian untuk mengatur tautan dan sumber *website*, klik *Settings* > *Pages*.



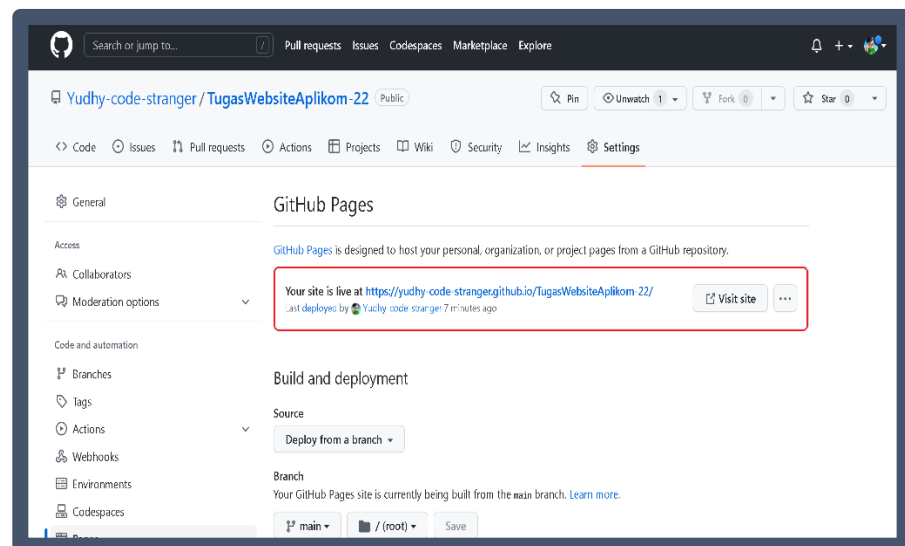
Gambar 3. 76 Mengatur tautan

10. Pilih *Branch* untuk sumber *Website*, yaitu *branch main*, kemudian klik tombol *save*. Selanjutnya, masukkan nama profil Github dan tautan *website*, pada kolom masukan *Custom Domain*, kemudian klik *Save*.



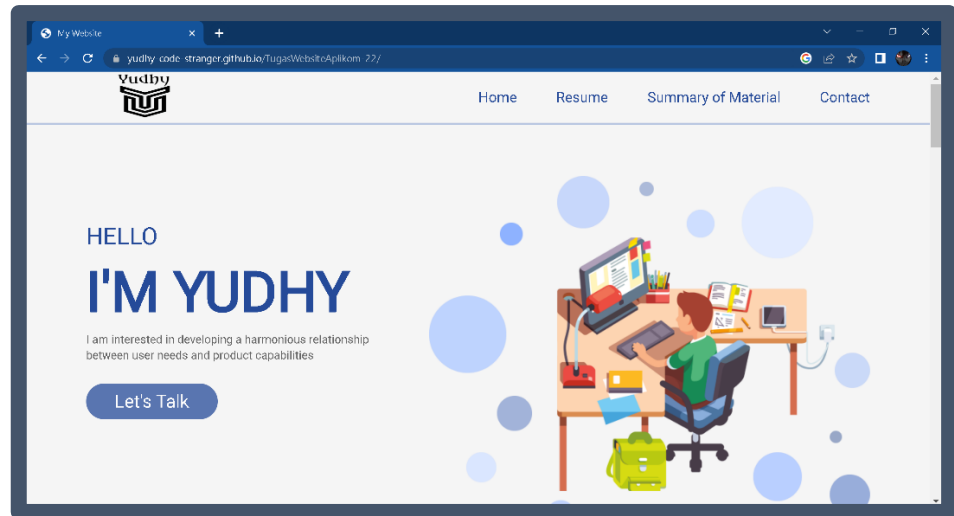
Gambar 3. 77 Memilih *Branch* dan *Custom Domain*

11. Setelah mengisikan sumber dan tautan *website*, tunggu beberapa saat dan muat ulang halaman. Dan akan tampil pemberitahuan bahwa *website* telah bisa diakses melalui Github *Pages* beserta tautannya. Klik tautan untuk melihat hasil *hosting website*.



Gambar 3. 78 Tampilan tautan *website* yang telah bisa diakses

12. *Website* telah berhasil diakses melalui *Github Pages*



Gambar 3. 79 Tampilan *Website* yang telah berhasil diakses

BAB IV

PENUTUP

4.1 Kesimpulan

Dari uraian yang telah dibahas di atas, dapat penulis simpulkan bahwa *World Wide Web* atau sering di kenal sebagai *website* adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet). Keistimewaan inilah yang telah menjadikan *website* sebagai *service* yang paling cepat pertumbuhannya. *Website* mengijinkan pemberian *highlight* (penyorotan atau penggaris bawahan) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, *frase*, *movie clip*, atau *file* suara.

Untuk membuat sebuah *website*, kita menggunakan HTML dan CSS sebagai bahasa *markup* dan *style*. HTML adalah bahasa penulisan yang digunakan untuk membuat dokumen halaman *website*, sedangkan CSS adalah bahasa yang digunakan untuk memberi *style* pada dokumen halaman yang telah dibuat menggunakan HTML

Penggunaan *hosting website* digunakan untuk menampung data-data yang diperlukan oleh sebuah *website* dan sehingga dapat diakses lewat internet. Data disini dapat berupa *file*, gambar, *email*, aplikasi/program/script dan *database*

4.2 Saran

Mengingat banyaknya manfaat yang dapat diperoleh dari penggunaan aplikasi Visual Studio Code ini, penulis menyarankan kepada seluruh pembaca agar selalu memperdalam pengetahuan tentang tata cara penggunaan dari sintaksis HTML dan CSS yang ter-*install* di Visual Studio Code, agar jika terdapat kebutuhan dalam pembuatan *website* dapat dilakukan secara efektif dan efisien guna meringankan beban pekerjaan dan mengefisienkan waktu.

DAFTAR PUSTAKA

- Evi Nurhayati. Rahmi.S. 2022. Sejarah *Web Service* Dalam Perkembangan Teknologi Informasi.
https://www.researchgate.net/profile/EviNurhayati/publication/339677960_SEJARAH_WEB_SERVICE_DALAM_PERKEMBANGAN_TEKNOLOGI_INFORMASI/links/5e5f9a8d4585152ce8059378/SEJARAH-WEB-SERVICE-DALAM-PERKEMBANGAN-TEKNOLOGI-INFORMASI.pdf. Diakses pada 8 Desember 2022.
- Iftitah Nurul. 2022. Pengertian *Website*, Beserta Jenis dan Fungsinya.
<https://katadata.co.id/safrezi/berita/6200a2a9697ec/pengertian-website-menurut-para-ahli-beserta-jenis-dan-fungsinya>. Di akses pada 7 Desember 2022.
- Kurnia Azizah. 2022. Pengertian HTML Lengkap dengan Fungsi dan Sejarah Kemunculannya.
<https://www.merdeka.com/trending/pengertian-html-lengkap-dengan-fungsi-dan-sejarah-kemunculannya-kln.html>. Diakses pada 8 Desember 2022.
- Nangi, Jumadil, Rizal Adi Saputra, dan Muhammad Nur Iksan. 2022. Modul Praktikum VII, Praktikum Aplikasi Komputer 2022, *Design Web - HTML,CSS*. Diakses pada 8 Desember 2022.

LAMPIRAN

