**PRAKTIKUM MATA KULIAH**

**Pemrograman Web 1**

**MODUL 8**

**GIT & GITHUB**

**DOSEN PENGAMPU**

Dr. Harja Santanapurba, M.Kom

Novan Alkaf B. S., S.Kom., M.T

Ihdalhubbi Maulida, M.Kom

**DISUSUN OLEH:**

Rizqa Oktavia Ramadhani / 2410131120010 / A2



**PROGRAM STUDI PENDIDIKAN KOMPUTER**

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

UNIVERSITAS LAMBUNG MANGKURAT

|  |
| --- |
| Tujuan Praktikum |

1. Mahasiswa mampu menginstall Git dan menjalankan perintah command pada Git
2. Mahasiswa mampu membuat akun github
3. Mahasiswa mampu upload, push and pull file

|  |
| --- |
| Kebutuhan Perangkat Keras, Infrastruktur & Perangkat Lunak |

1. Perangkat Keras
2. Komputer atau Laptop
3. Jaringan Internet
4. Perangkat Lunak
5. Visual Studio Code
6. Peramban (*browser*)
7. Git dan akun GitHub

|  |
| --- |
| Materi Pokok |

1. Git

Git merupakan software berbasis Version Control System (VCS) yang bertugas untuk mencatat perubahan seluruh file atau repository suatu project. Developer software biasa menggunakan Git untuk distributed revision (VCS terdistribusi), hal ini bertujuan untuk menyimpan database tidak hanya ke satu tempat. Namun semua orang yang terlibat dalam penyusunan kode dapat menyimpan database ini.

Prosedur yang diterapkan ini dapat membantu antar divisi project untuk memantau dan menghubungkan (merge) antar ekstensi yang berbeda dengan mudah. Sehingga aplikasi yang dibuat oleh sebuah tim project dapat berfungsi tanpa menghubungkan secara manual.

Terdapat istilah commit pada Git yang berfungsi untuk menyimpan riwayat perubahan data pada file. Melalui commit, developer dapat kembali ke source code sebelumnya dengan istilah checkout.

Untuk mengoperasikan Git, kamu perlu menginstall software terlebih dahulu sehingga pekerjaan ini dapat dilakukan secara offline (tidak terkoneksi internet). Software ini juga tersedia secara gratis melalui web unduhan resminya di Git Downloading.

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

* **Git init** ini untuk membuat sebuah repository yang berada pada file lokal dimana nantinya file tersebut bernama .git.
* **Git status** ini untuk mengetahui status dari sebuah repository lokal.
* **Git add** ini untuk menambahkan file baru yang ada pada repository yang telah dipilih.
* **Git commit** ini untuk menyimpan perubahan yang telah dilakukan.
* **Git push** ini digunakan untuk mengirimkan perubahan file setelah di commit ke remote repository.
* **Git branch** ini melihat seluruh branch yang ada dalam repository.
* **Git checkout** ini dapat menukar branch yang aktif dengan branch yang telah dipilih.
* **Git merge** ini untuk menggabungkan branch yang aktif dan yang telah dipilih.
* **Git clone** ini dapat membuat salinan repository lokal.

1. Github

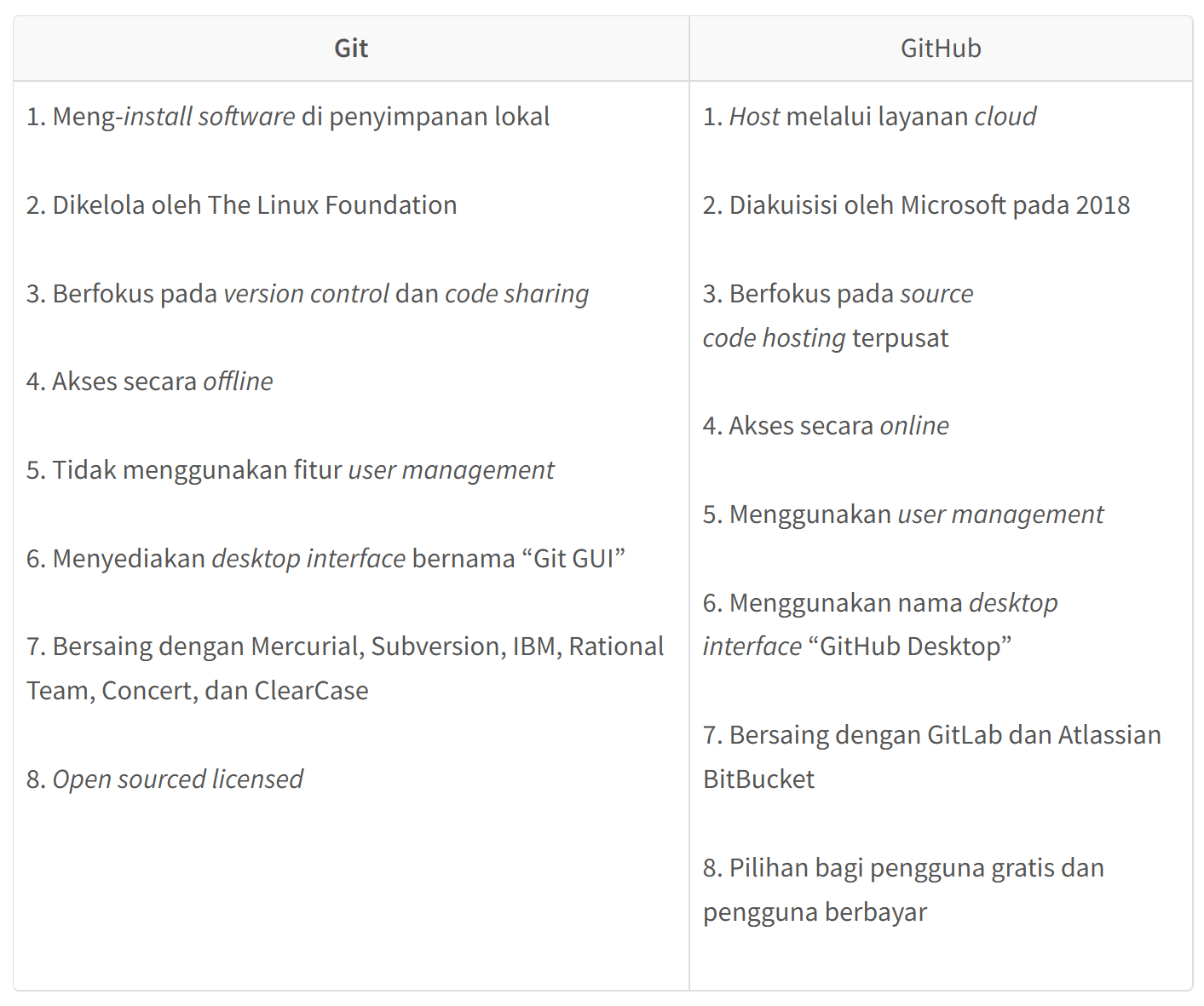
Melalui homepage pada [GitHub](https://github.com/), disebutkan bahwa GitHub adalah platform khusus developer yang dibuat karena terinspirasi dari cara bekerja para programmer. Dari open source ke bisnis, kita bisa meng-host dan meninjau sebuah kode, mengelola project, dan membangun perangkat lunak bersama 56 juta lebih pengguna di seluruh dunia.

Gampangnya, GitHub merupakan manajemen project, sistem versioning code, sekaligus platform jaringan sosial bagi para developer seluruh dunia. Banyak sekali fungsi yang bisa digunakan oleh para developer, sehingga melalui platform ini memudahkan developer dalam mengembangkan sebuah karya.

Konsep kerja GitHub pada dasarnya sama dengan Git yaitu dapat menulis source code secara individu atau tim. User interface yang tersedia pada GitHub lebih menarik dan mudah dipahami oleh pengguna awal. Pekerjaan secara tim, pengguna juga bisa melihat siapa penulis kode dan tanggal berapa kode tersebut dibuat.

1. Perbedaan Git dan Github

Perbedaan mendasar antara Git dan GitHub adalah fungsinya. Git adalah alat yang mengelola perubahan kode di komputer kita secara lokal, sedangkan GitHub adalah layanan online yang digunakan untuk menyimpan kode secara cloud. Git itu seperti aplikasi, sementara GitHub adalah website yang meng-host kode kita agar lebih mudah diakses dan dikelola secara kolaboratif. Berikut ini perbedaan dari kedua platform tersebut.



1. Cara Mengenal Command

Developer yang menggunakan Git dapat menggunakan command-line tool yaitu pengubah kode dan dapat digabungkan menuju perangkat lokal. Sedangkan, GitHub menyediakan interface grafis berbasis cloud sebagai tempat untuk melakukan seluruh tugas. Interface Githubini juga menawarkan kontrol akses developer dan fitur kolaborasi, serta tools lainnya sebagai sarana manajemen pekerjaan yang sedang dilakukan.

Sedangkan Git cenderung fokus pada tugas-tugas SCM yang eksklusif seperti push and pull, reset, commit, fetch, dan merge*.* GitHub berfungsi sebagai tuan rumah terhadap repositori Git dalam penyimpanan kode di lokasi yang terpusat. Selain itu, Git menjadi alat yang digunakan untuk mengelola beberapa versi editing dari sumber kode yang ditransfer menuju repository. GitHub menjadi lokasi akhir dari unggahan salinan repository Git.

1. Fitur yang Tersedia

Dengan fungsi dan tujuan yang saling mendukung di tengah perbedaan, ada beragam fitur yang bisa kamu peroleh dari masing-masing tools. Seluruh fitur dapat dioperasikan untuk memudahkan pembangunan pemrograman dan coding dalam peran kamu sebagai developer. Berikut fitur-fitur yang ditawarkan oleh Git:

* Kompatibilitas: Git cenderung kompatibel dengan seluruh sistem operasi. Fitur ini juga sudah mencakup repository jarak jauh yang dibuat oleh Subversion (SVN).
* Branching: Fitur Branching dapat membantu membuat modifikasi pada proyek tanpa memengaruhi versi aslinya.
* Reliabilitas: Data yang disimpan pada server pusat akan selalu dicadangkan. Bahkan ketika server pusat rusak, data-data tidak akan hilang dan dapat diperoleh kembali dengan mudah.
* Kecepatan: Git mampu menangani proyek berskala besar secara efisien. Lebih cepat dan terukur bila dibandingkan dengan kontrol versi (VCS) yang lain.
* Ekonomis: Git dapat dioperasikan secara gratis dan berhasil dirilis melalui General Public License (GPL) yang mampu menghemat biaya ketika harus menghabiskannya demi server mahal.

Sedangkan, GitHub juga punya fitur-fitur yang tidak kalah bermanfaat untuk mendukung kinerja developer, yaitu:

* Drag and drop Gist: Gist merupakan sebuah widget GitHub yang menerima hosting dari kode di dalam file. Terdapat sejumlah besar cuplikan kode dalam berbagai bahasa
* File finder: Fitur ini digunakan untuk menavigasi file yang ada di repository proyek secara cepat. Bila ingin mengaktifkan fitur ini, kamu perlu menekan tombol T atau ketikkan nama file yang dimaksud.
* Emoji GitHub: Merupakan sekumpulan emoticon yang mampu mengekspresikan sesuatu sesuai keinginan kamu. Emoji ini dapat ditambahkan menuju Wiki dan README.md
* Git.io: Berfungsi untuk mempersingkat URL dari GitHub dalam memudahkan pembagian repositori ke tujuan yang berbeda-beda.
* Linking lines: Fitur ini bermanfaat untuk membagi garis tertentu pada sebuah file dengan menambahkan #L dan diikuti nomor baris. Salin garis awal dan akhir kemudian tempelkan setelah parameter tertentu

Salah satu fitur terbaik yang jadi populer dari Git adalah kemampuannya untuk membuat repository yaitu tempat penyimpanan yang berisi code source dan riwayat perubahannya. Dalam repository ini setiap ada perubahan akan disimpan secara terpisah dalam commit yang memungkinkan developer untuk kembali ke versi sebelumnya jika terjadi kesalahan atau perlu memulihkan kode yang hilang. Selain itu, Git juga populer karena bisa membuat branch yang memungkinkan kamu untuk bekerja di fitur baru tanpa harus mengganggu kode yang sudah ada.

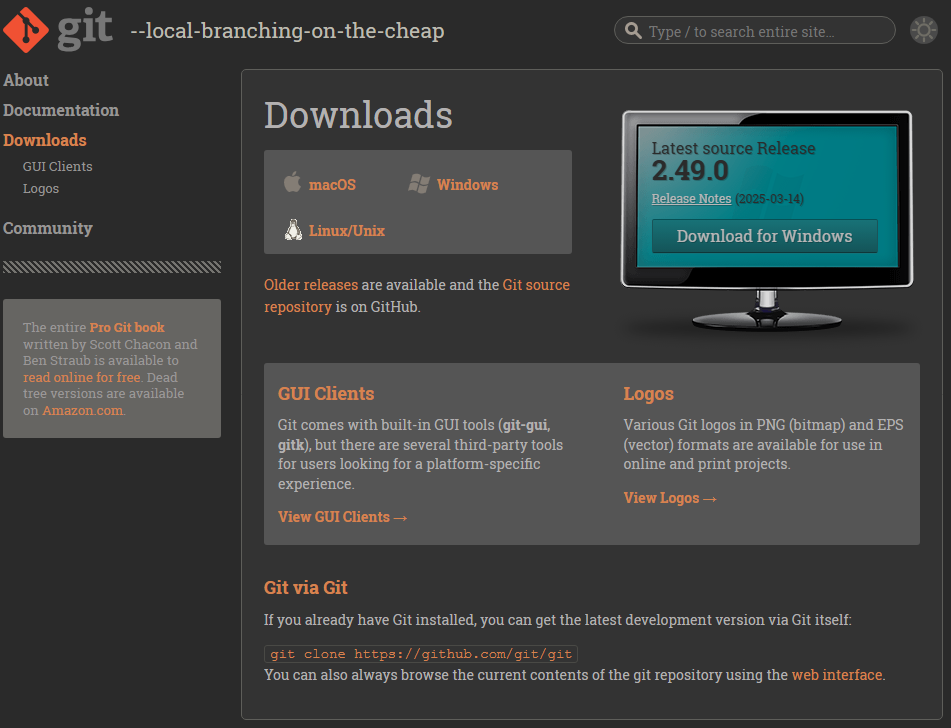
Sedangkan di GitHub ada fitur populer yaitu “Continuous Integration” atau biasa dikenal dengan istilah CI. Fitur ini bisa membantu pekerjaan kamu lebih mudah karena bisa otomatisasi pengujian kode dan memastikan semua perubahan masih bisa berfungsi sebelumnya dimasukan ke versi utama. Fitur ini juga bisa membantu agar kualitas perangkat lunak yang kamu kembangkan semakin optimal.

|  |
| --- |
| Lembar Kerja / Langkah Praktikum |

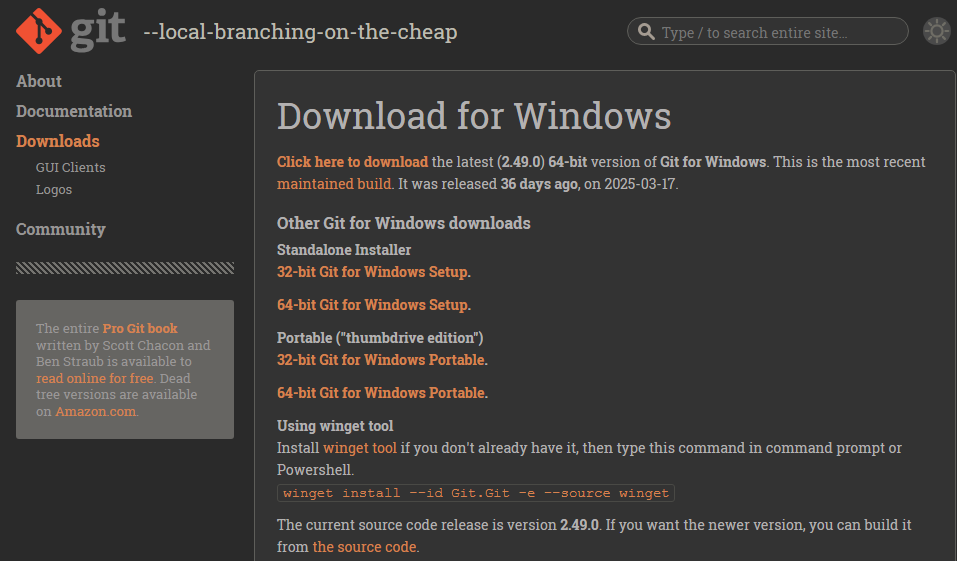
1. **Instalasi Git**
   1. **Cara download Git** melalui link berikut.

[Git - Downloads](https://git-scm.com/downloads)

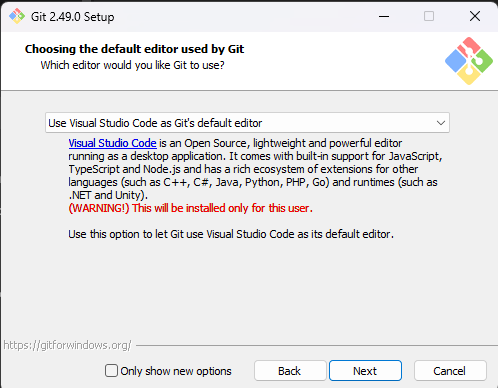
* 1. Klik ***Download for Windows***



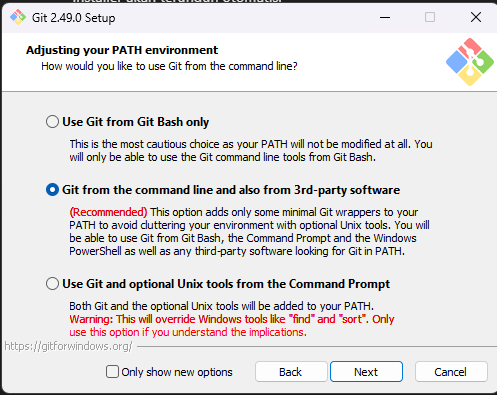
* 1. Pilih ***Click here to download***



* 1. Setelah terunduh, jalankan installer.
  2. Pengaturan saat Instalasi (biarkan default kecuali):
     1. **Choosing the default editor:** Pilih “Use Visual Studio Code as Git’s default editor” jika menggunakan VS Code.



* + 1. **Adjusting your PATH environment:** Pilih “Git from the command line and also from 3rd-party software”.



* + 1. Klik ***Next*** terus hingga tombol *Install* aktif, lalu klik ***Install***.
  1. Jika proses telah selesai, klik ***Finish***. Sekarang Git sudah terpasang.

1. **Kolaborasi Proyek Web dalam Kelompok dengan GitHub**
   1. **Membuat Proyek Portofolio Pribadi**
      1. Buat folder dengan nama **PortofolioNama.**
      2. Isikan folder tersebut dengan file **index.html** dan **style.css.**
      3. Tulis kode berikut pada file **index.html.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<h1>Siti Kania Nur Alya Putri</h1>

<p>Mahasiswa Pendidikan Komputer</p>

</header>

<nav>

<a href="#tentang">Tentang</a>

<a href="#proyek">Proyek</a>

<a href="#kontak">Kontak</a>

</nav>

<section id="tentang">

<h2>Tentang Saya</h2>

<p>Saya adalah mahasiswa yang sedang belajar

membuat website.

Saya menyukai dunia teknologi dan ingin menjadi

web developer.</p>

</section>

<section id="proyek">

<h2>Proyek Saya</h2>

<ul>

<li>Website Daftar Buku</li>

<li>Aplikasi Pemesanan Online</li>

<li>To-Do List Sederhana</li>

</ul>

</section>

<section id="kontak">

<h2>Kontak</h2>

<p>Email: stkania@email.com</p>

<p>Instagram: @stkania</p>

</section>

<footer>

<p>&copy; 2025 Siti Kania Nur Alya Putri</p>

</footer>

</body>

</html>

* + 1. Tulis kode berikut pada file **style.css**

**body {**

**font-family: 'Trebuchet MS', Tahoma, Verdana, sans-serif;**

**margin: 0;**

**padding: 0;**

**background-color: #f0f2f5;**

**color: #333;**

**}**

**header {**

**background-color: #4a90e2;**

**color: white;**

**text-align: center;**

**padding: 2em 1em;**

**box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);**

**}**

**nav {**

**background-color: #3b74c5;**

**text-align: center;**

**padding: 1em 0;**

**}**

**nav a {**

**color: white;**

**text-decoration: none;**

**margin: 0 15px;**

**font-weight: bold;**

**}**

**nav a:hover {**

**text-decoration: underline;**

**}**

**section {**

**max-width: 800px;**

**margin: 2em auto;**

**padding: 1.5em;**

**background-color: white;**

**border-radius: 10px;**

**box-shadow: 0 4px 10px rgba(0, 0, 0, 0.05);**

**}**

**section h2 {**

**border-bottom: 2px solid #ddd;**

**padding-bottom: 0.5em;**

**margin-bottom: 1em;**

**}**

**ul {**

**padding-left: 20px;**

**}**

**footer {**

**background-color: #4a90e2;**

**color: white;**

**text-align: center;**

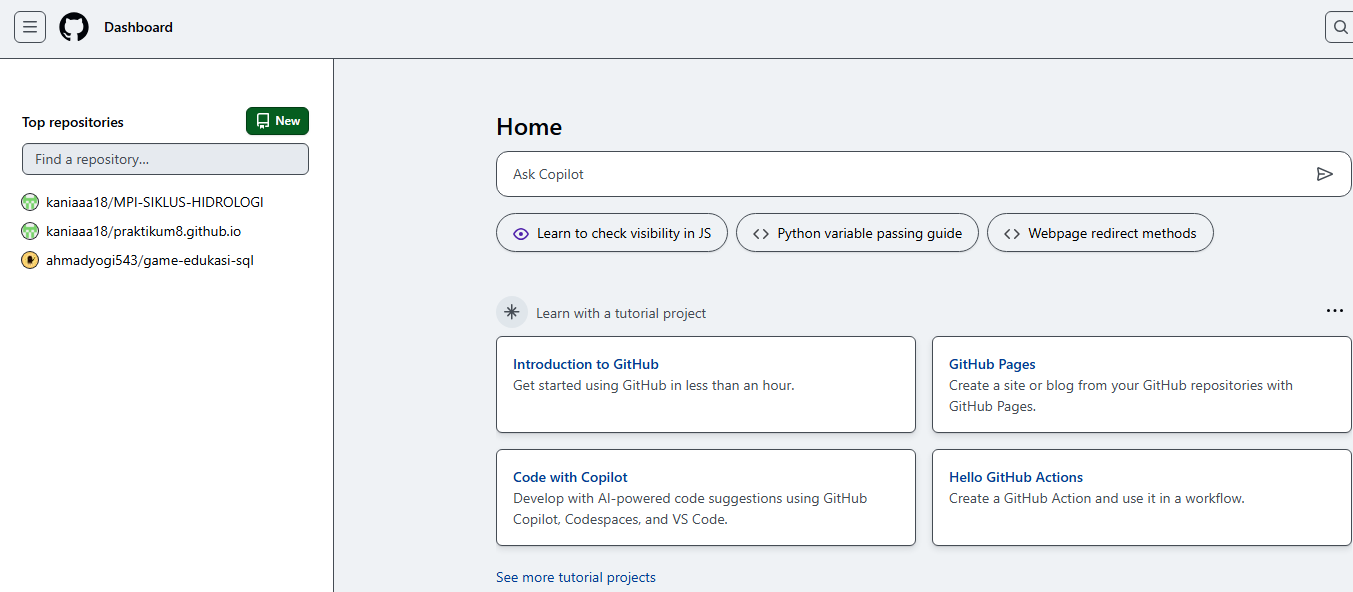
**padding: 1em;**

**margin-top: 3em;**

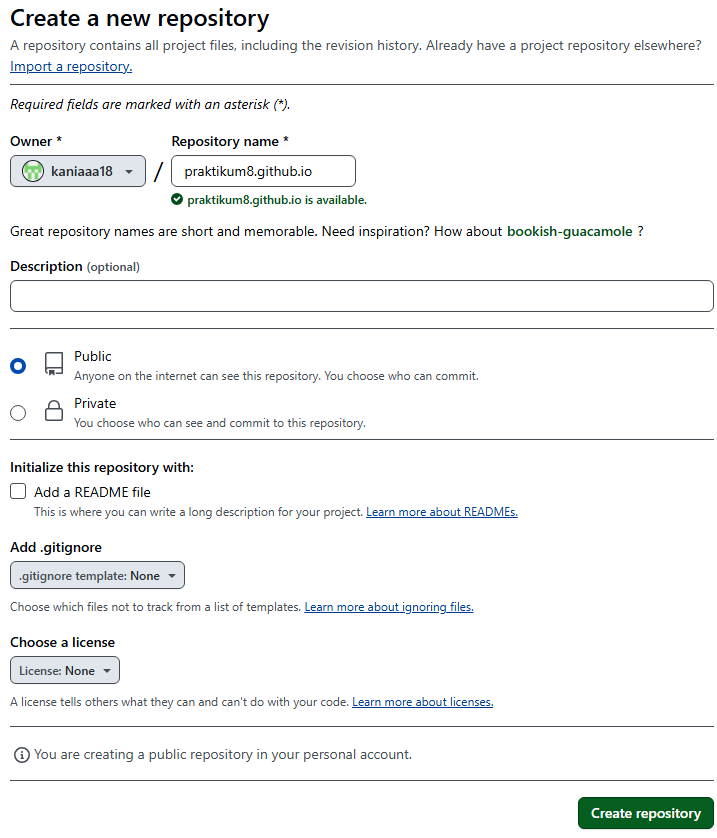
**font-size: 0.9em;**

**}**

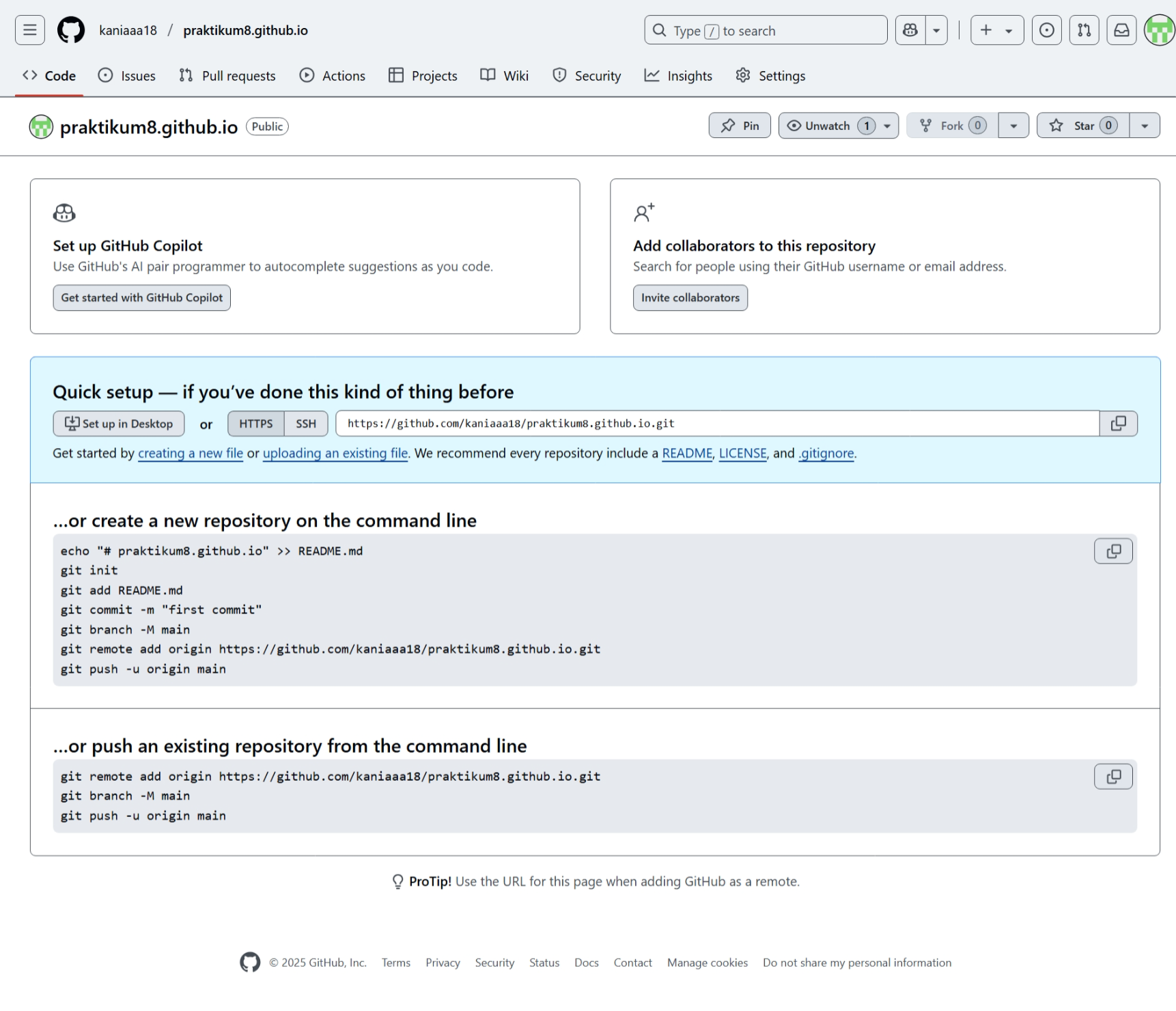
* 1. **Cara Penggunaan GitHub**
     1. Masuk ke halaman GitHub [GitHub](https://github.com/) dan lakukan registrasi/login.
     2. Klik **New** untuk membuat repository baru.



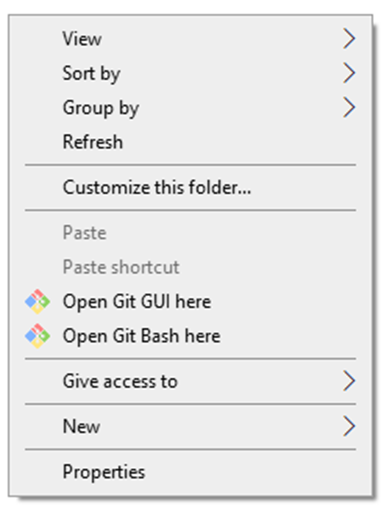
* + 1. Maka tampilannya akan seperti berikut. Isikan bagian ***Repository name*** dengan ***namafile.github.io*** lalu klik ***Create repository.***



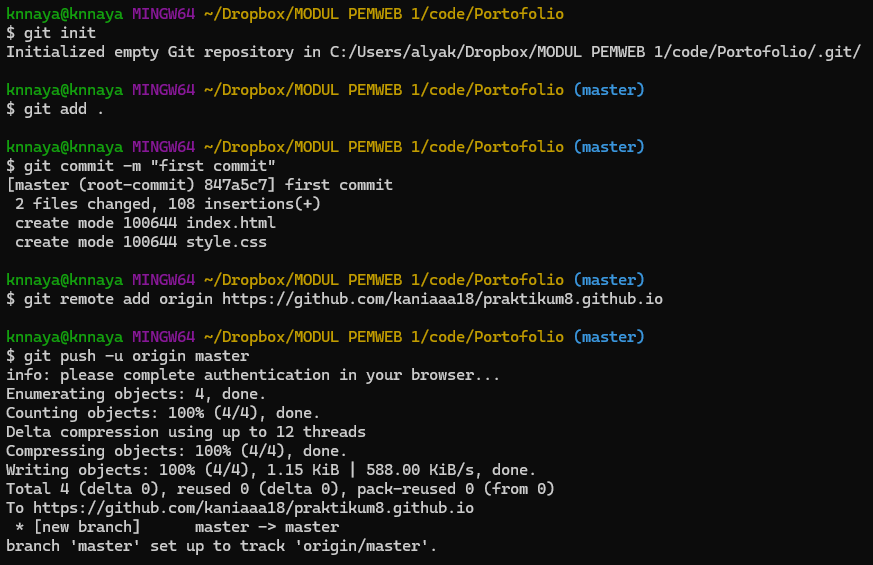
* + 1. Maka hasil tampilannya akan seperti berikut.



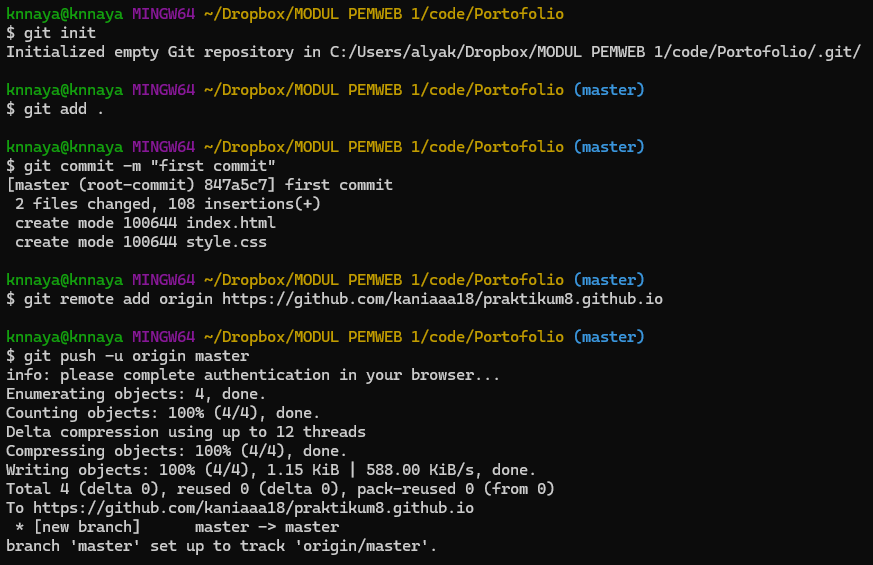
* + 1. Setelah itu, buka **File Explorer**kalian dan cari folder yang telah di buat. Klik kanan pada folder, lalu pilih ***open git bash here*** untuk kita push di GitHub.



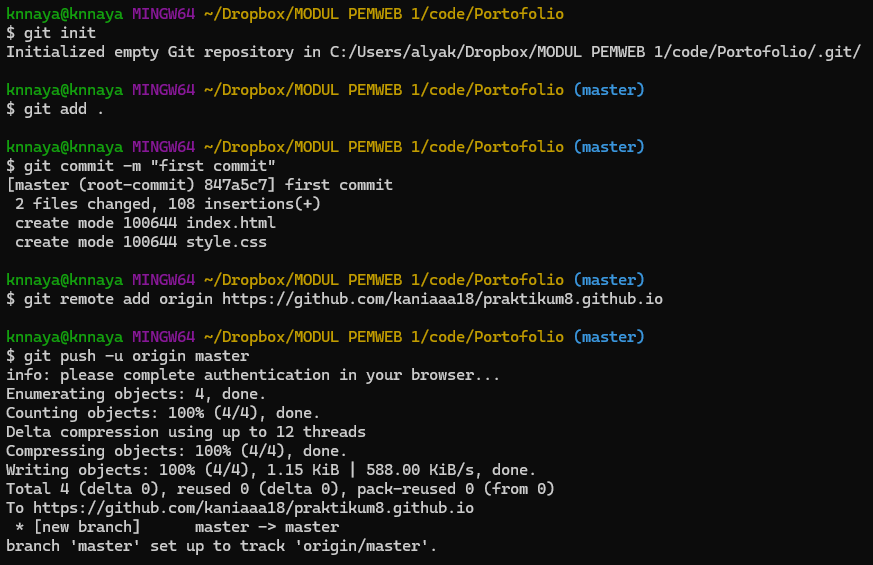
* + 1. Maka akan terbuka terminal seperti ini. Ketikkan ***git init*** lalu enter.



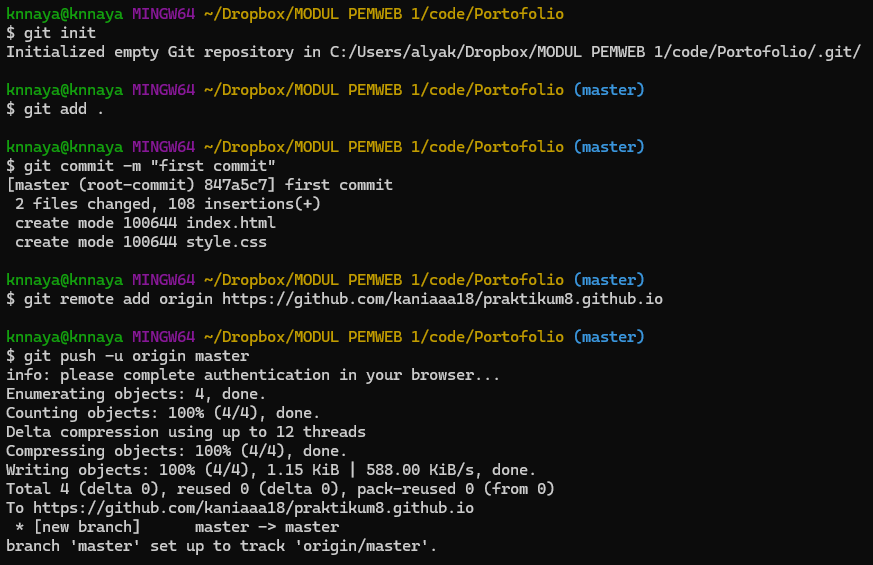
* + 1. Ketikkan ***git add .*** lalu enter.



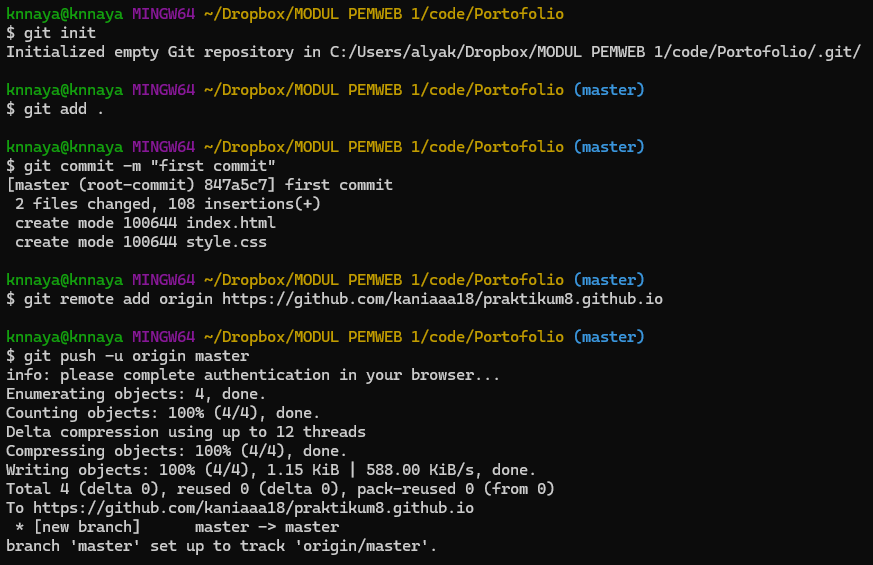
* + 1. Ketikkan ***git commit -m “first commit”.***

******

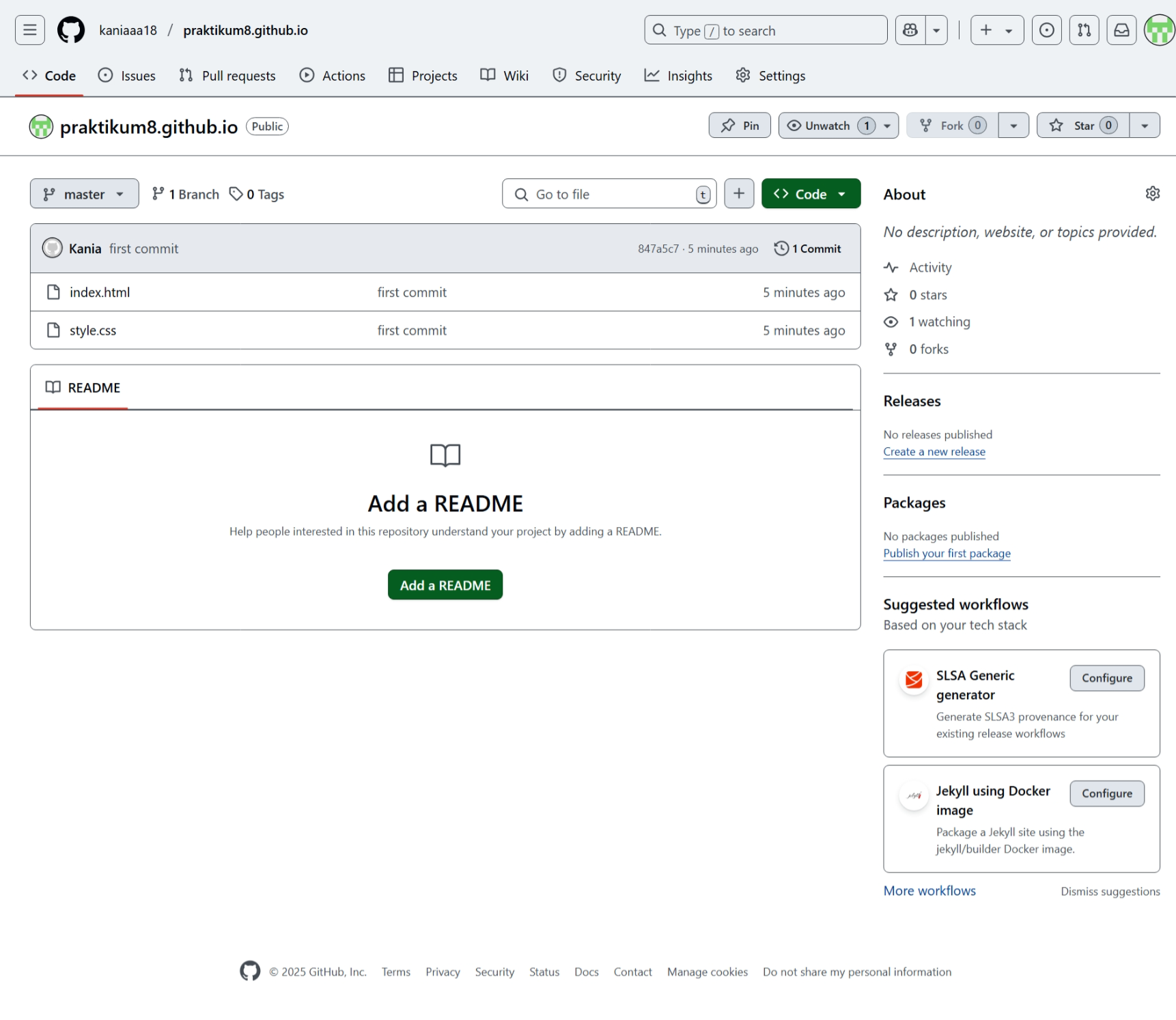
* + 1. Kemudian ketikkan ***git remote add origin link\_GitHub\_kalian.***

******

* + 1. Ketikkan ***git push -u origin master***. Tunggu hingga proses berjalan dan selesai.

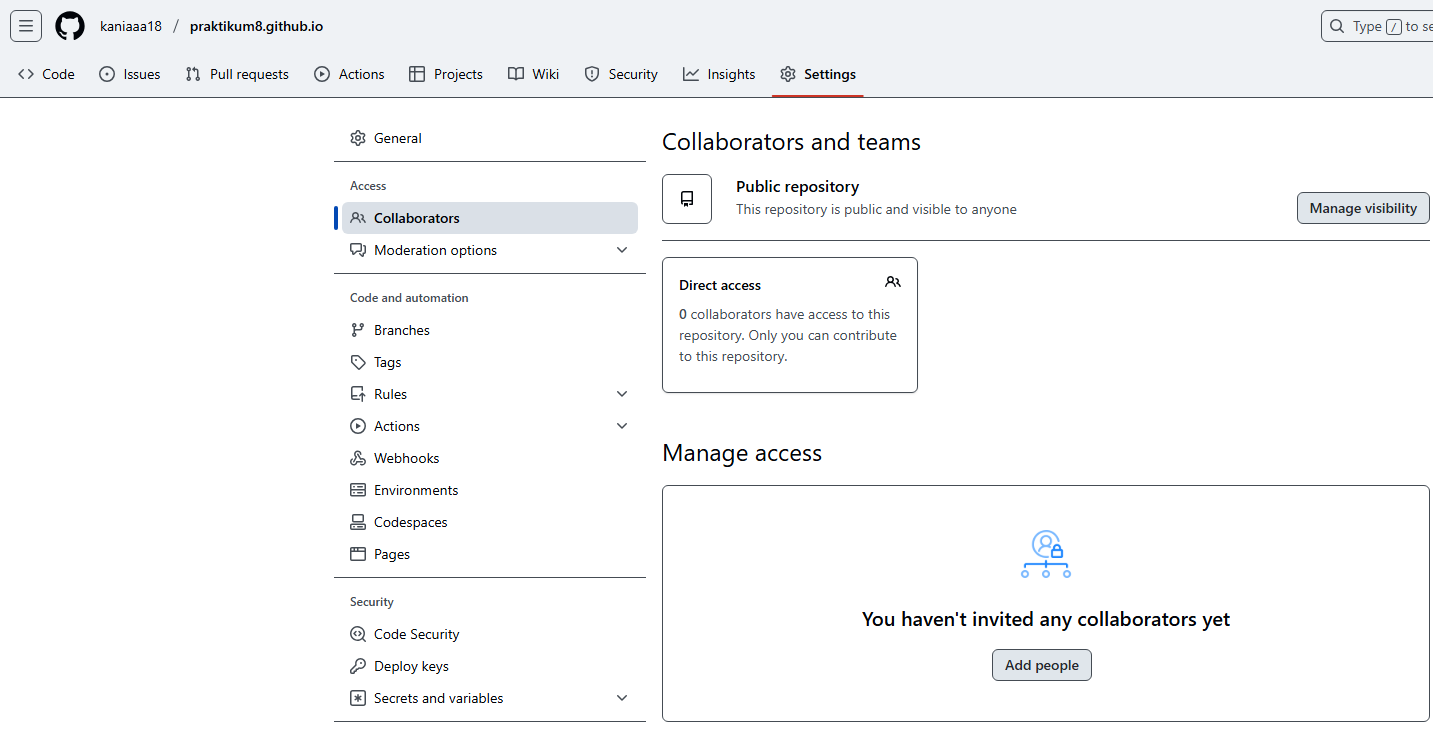


* + 1. Refresh halaman repository pada GitHub kalian. Maka semua file dalam folder sudah ter-upload.

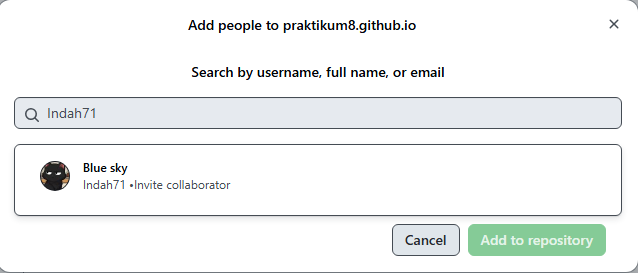


* 1. **Kolaborasi Proyek Kelompok**
     1. Tambahkan teman/collabolators pada repository GitHub kalian.

**Setting > Collabolators > Add People**



* + 1. Klik ***Add People*** dan tambahkan username/email teman kalian.



* + 1. Pilih salah satu repository dari teman kelompok untuk menjadi repository utama.
    2. Untuk melakukan **Pull,** teman yang lain lakukan kembali langkah **vi - ix.** Pada bagian link GitHub gunakan link repository utama tadi.
    3. Lalu pada terminal teman yang lain, ketikkan ***git pull origin master***.



* + 1. Semua dilakukan bergantian setelah push, tunggu teman yang lain pull, lalu ulangi hal itu kembali

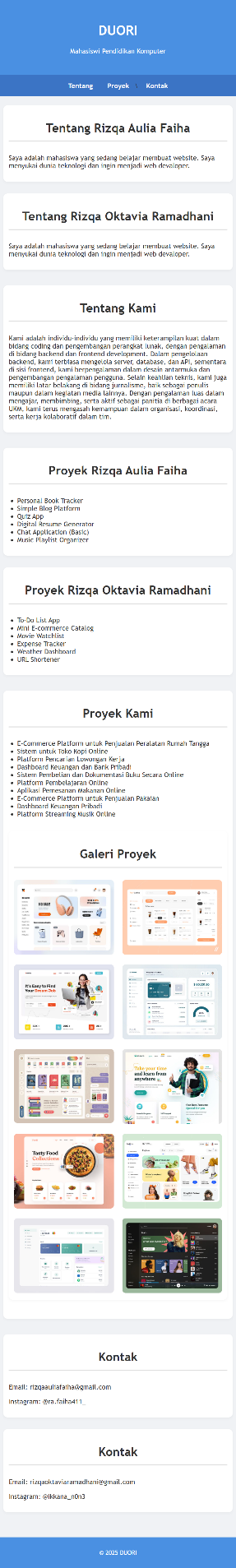
|  |
| --- |
| Tugas Praktikum |

1. Buat **tim** dengan beranggotakan 2 atau 3 orang. Kemudian kerjakan Lembar Kerja dengan melakukan pembagian tugas yang sesuai.
2. Buat sebuah repository di Github, kemudian lakukan **push** projek Lembar Kerja menggunakan Git.
3. Lakukan **dokumentasi** pada langkah yang sudah dilakukan. Lalu jelaskan langkah-langkah tersebut.
4. Sertakan **link** repositori atau Github pada akhir penjelasan.

**Catatan:** Kode HTML boleh berisi portofolio tim. Kembangkan isi web sekreatif kalian!

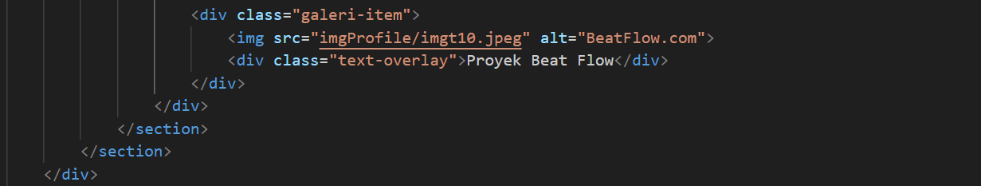
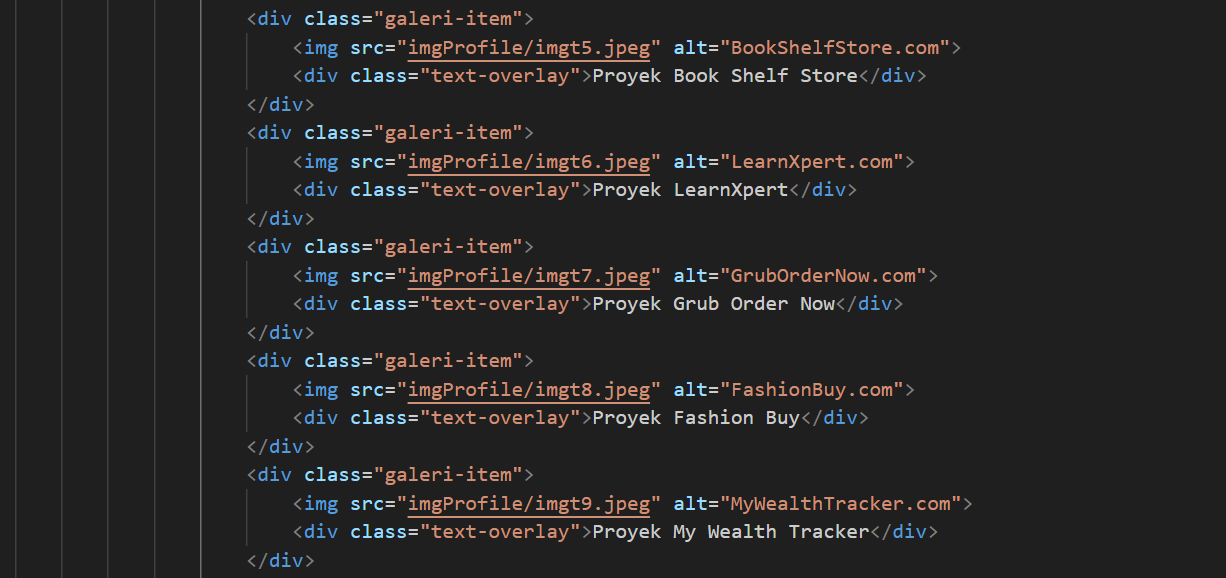
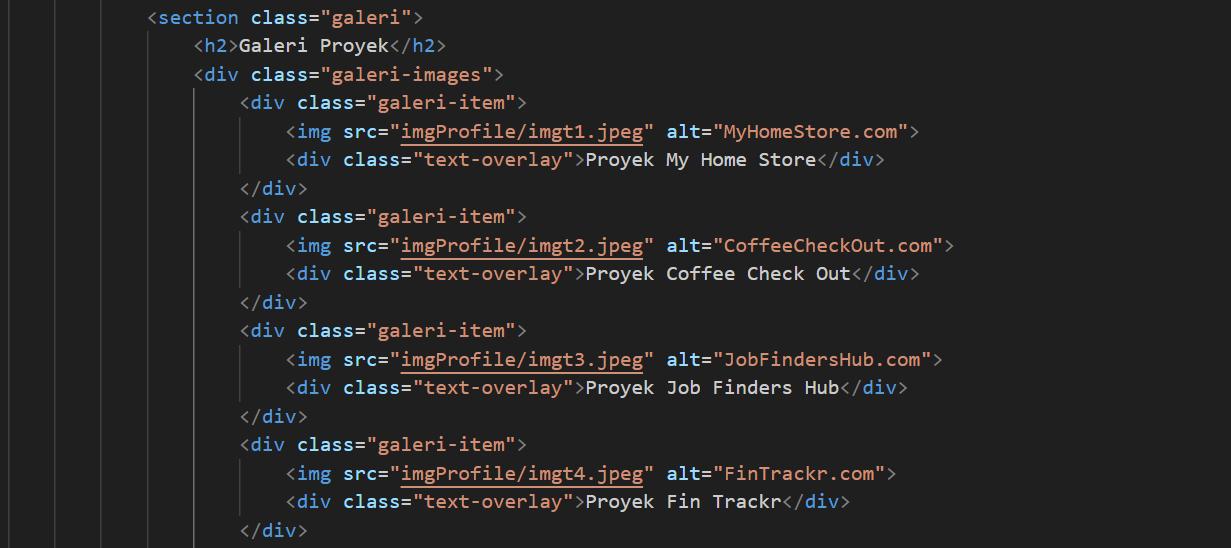
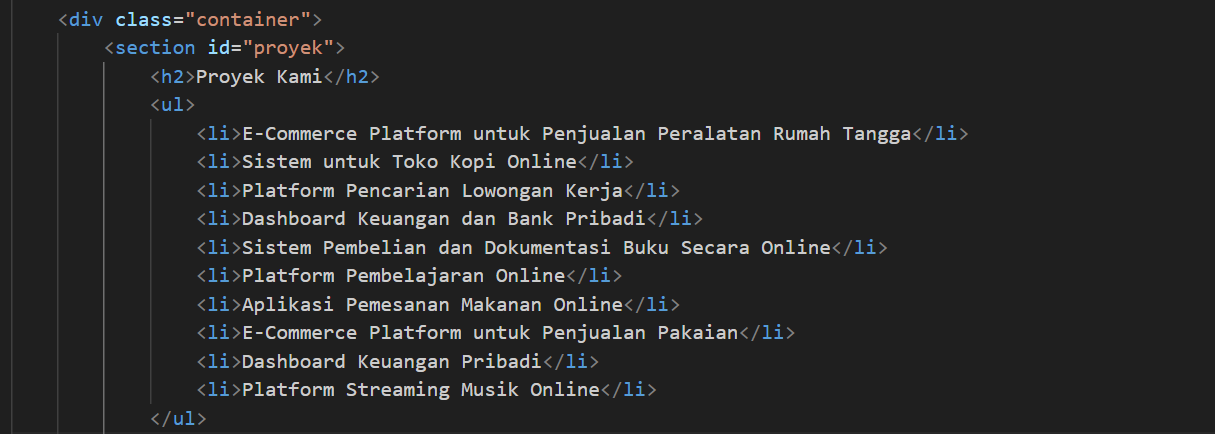
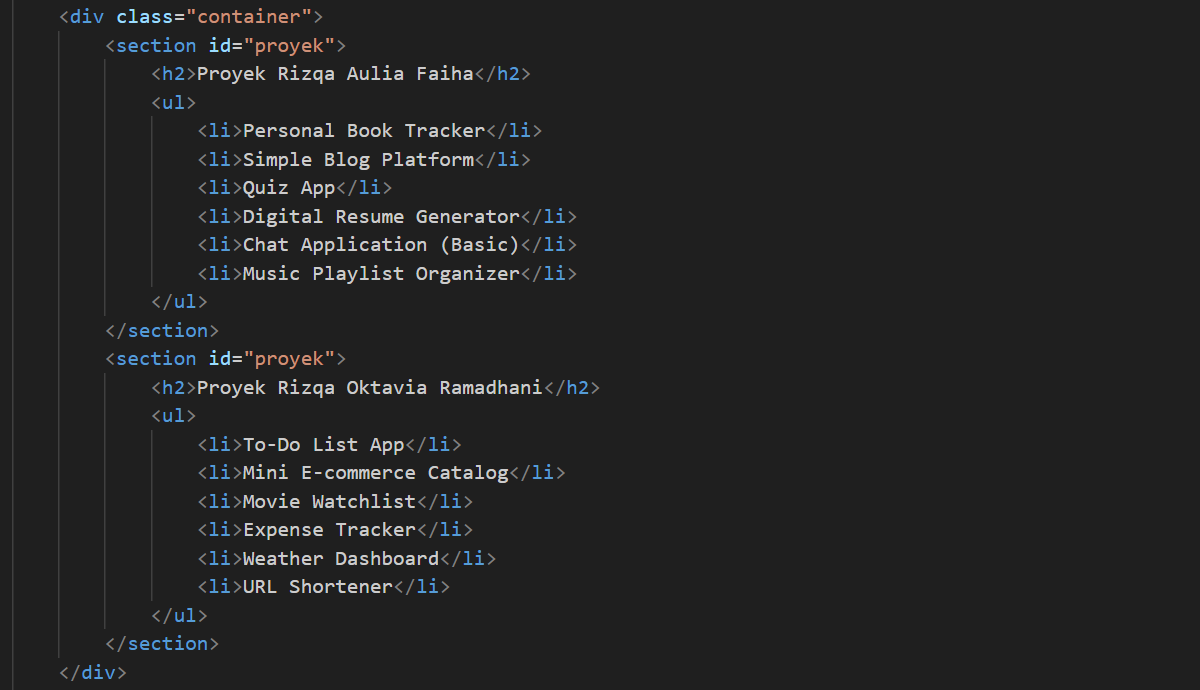
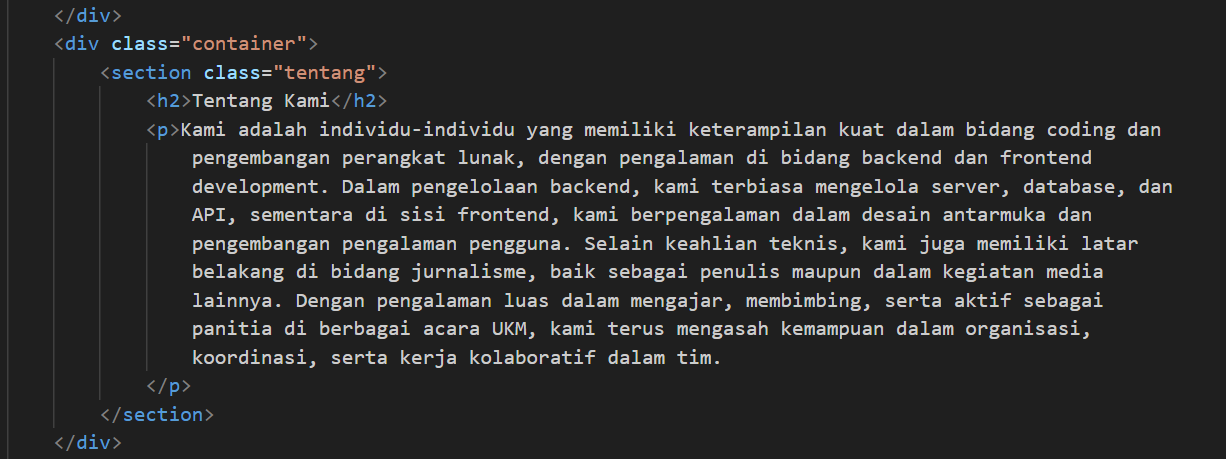
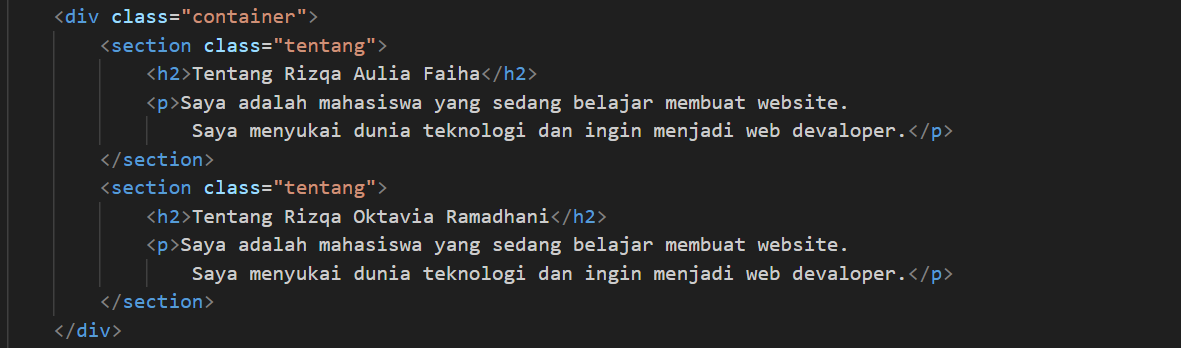
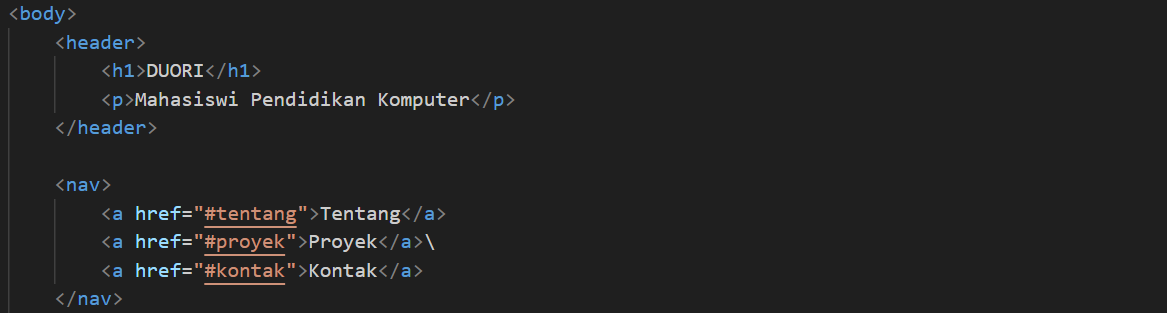
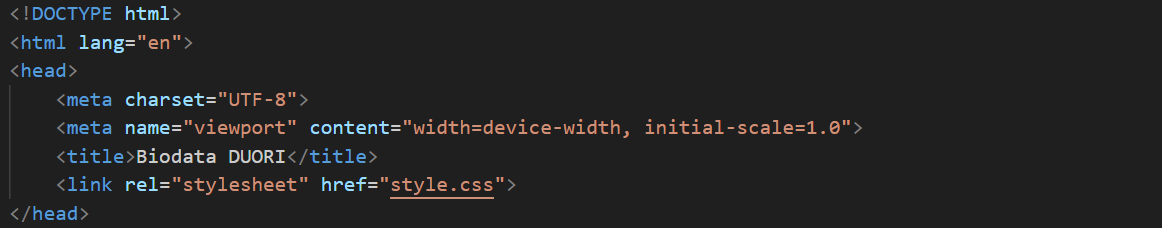
**Hasil Pengerjaan Tugas Praktikum**

Screanshoots hasil program:

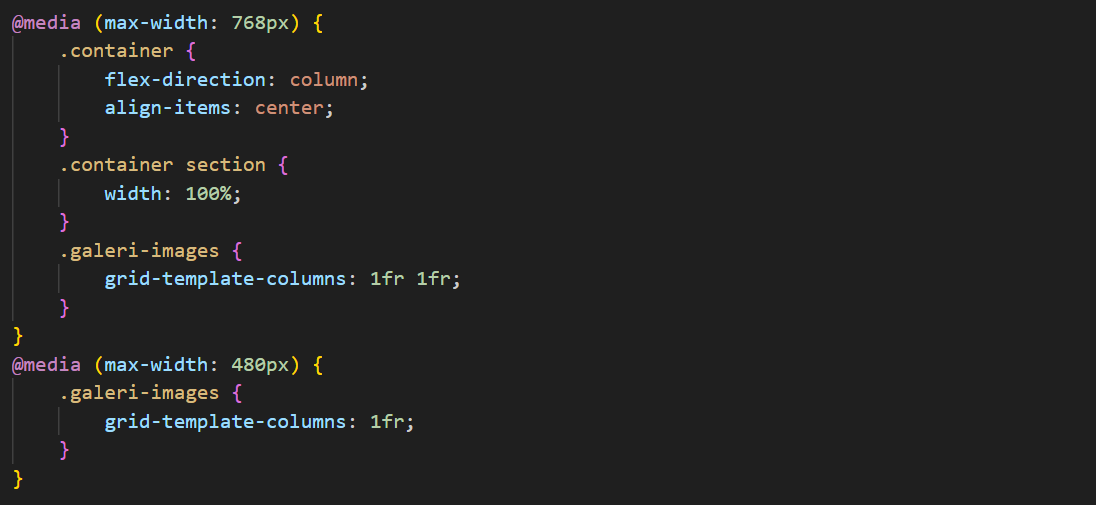
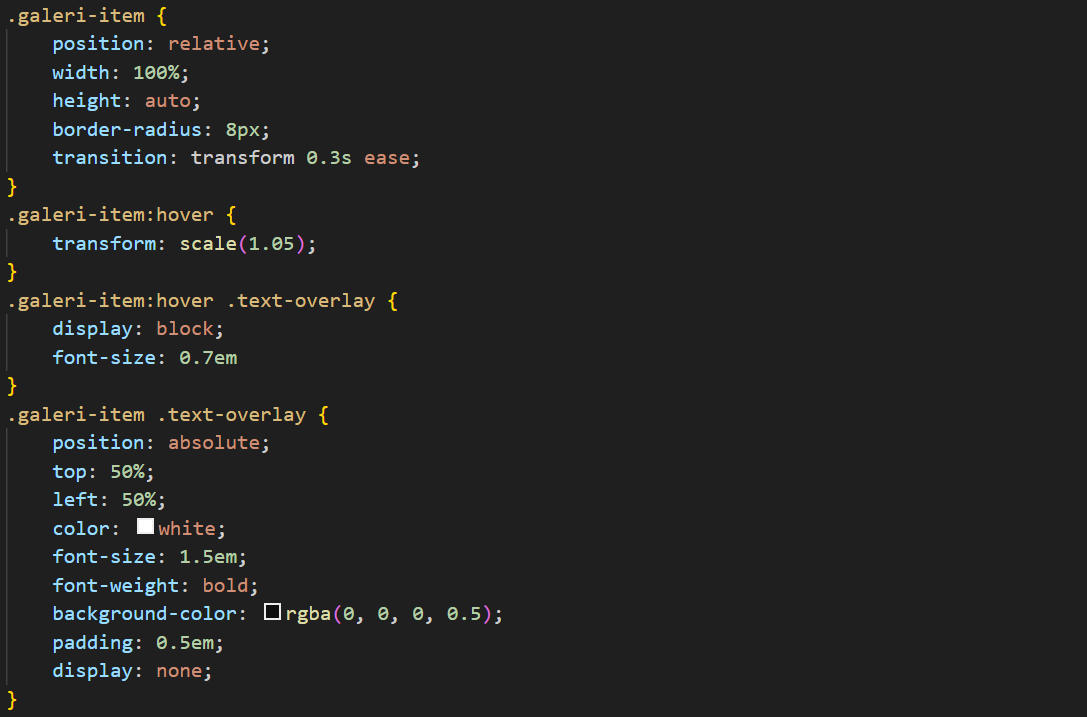
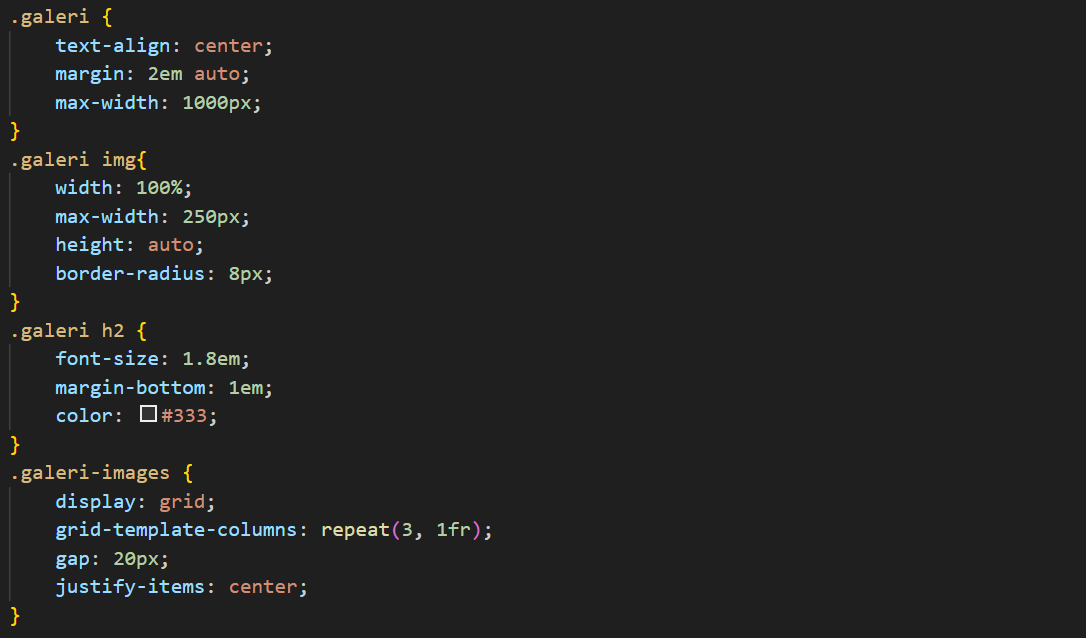
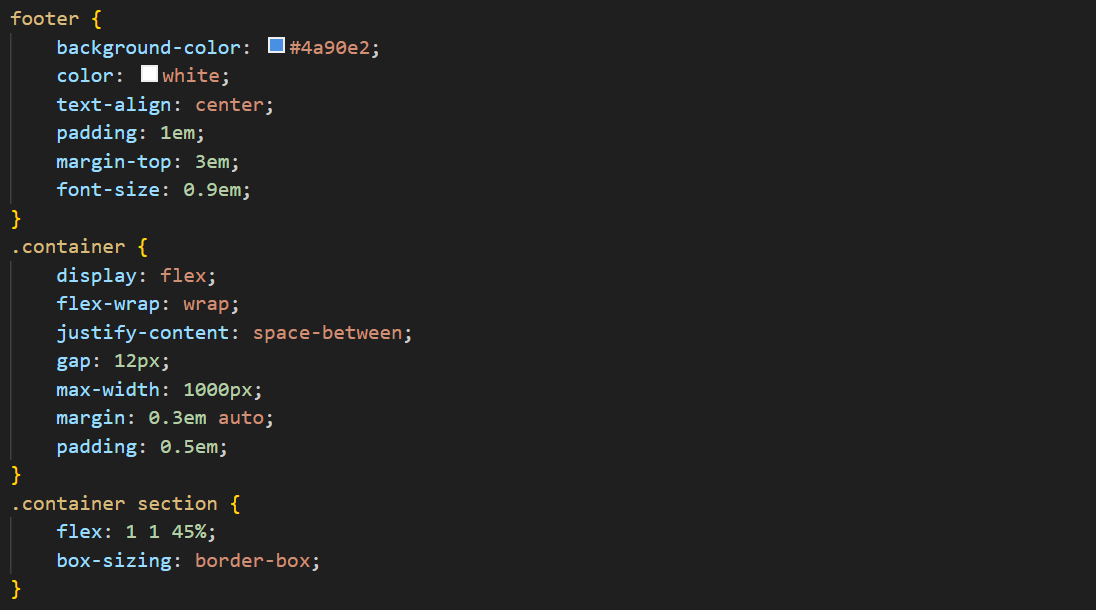
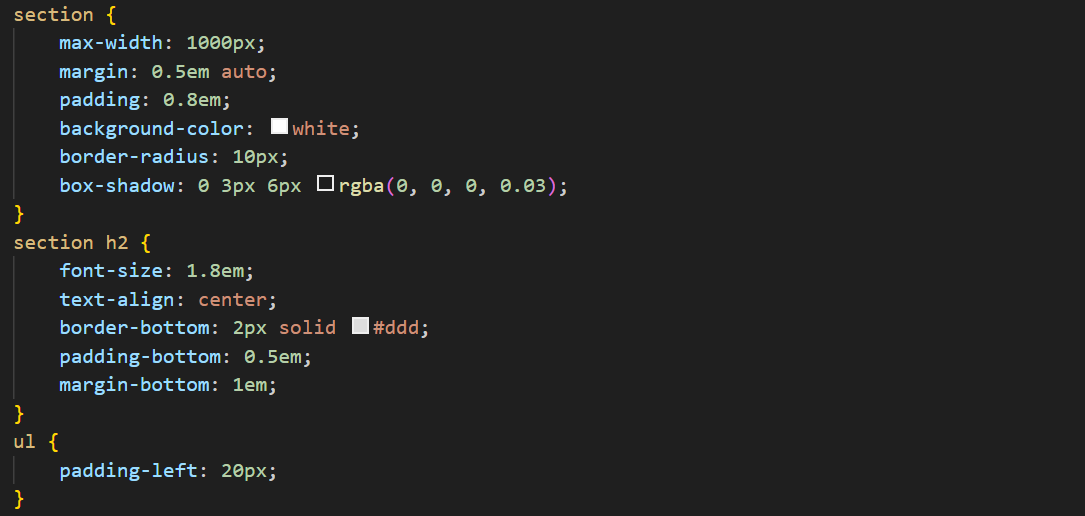
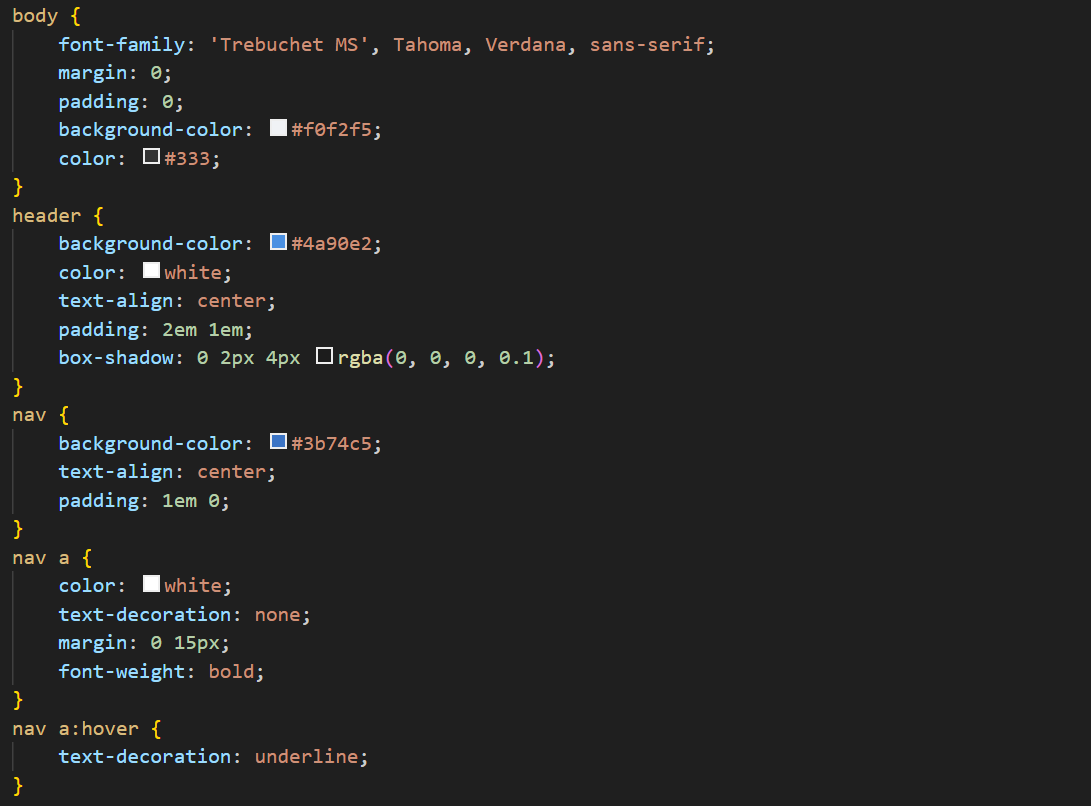


Screanshoot kode program:

Kode HTML:



Kode CSS:



Hasil pembahasan:

File index.html:

Kita buat file dengan nama index.html agar memuat tampilan isi web nanti:

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Biodata DUORI</title>*

*<link rel="stylesheet" href="style.css">*

*</head>*

Kode bagian awal dari sebuah halaman web berbasis HTML. Baris pertama <!DOCTYPE html> berfungsi untuk mendeklarasikan bahwa dokumen ini menggunakan HTML5 sebagai standarnya. Tag <html lang="en"> membuka elemen utama HTML dan menentukan bahwa bahasa yang digunakan di dalam dokumen ini adalah bahasa Inggris (lang="en"). Pada bagian <head>, terdapat beberapa elemen penting: <meta charset="UTF-8"> digunakan untuk mendefinisikan set karakter yang dipakai, yaitu UTF-8, agar semua karakter (termasuk simbol internasional) dapat ditampilkan dengan benar. Selanjutnya, <meta name="viewport" content="width=device-width, initial-scale=1.0"> memastikan bahwa tampilan halaman akan responsif, menyesuaikan lebar layar perangkat pengguna, terutama pada perangkat mobile. Tag <title>Biodata DUORI</title> menentukan judul halaman yang akan ditampilkan di tab browser. Terakhir, <link rel="stylesheet" href="style.css"> berfungsi untuk menghubungkan halaman HTML ini dengan file CSS eksternal bernama style.css, yang berisi aturan-aturan styling untuk mempercantik tampilan halaman. Bagian ini merupakan struktur dasar yang penting untuk membangun halaman web yang terorganisir dan responsif.

*<body>*

*<header>*

*<h1>DUORI</h1>*

*<p>Mahasiswi Pendidikan Komputer</p>*

*</header>*

Kode mulai membangun isi utama halaman web dengan membuka tag <body>, yang berisi semua konten yang akan ditampilkan kepada pengguna. Di dalam <body>, terdapat elemen <header>, yang umumnya digunakan untuk menampilkan bagian atas (kepala) dari halaman, seperti judul atau informasi pengantar. Di dalam <header> tersebut, terdapat tag <h1>DUORI</h1>, yang menunjukkan heading utama dengan teks "DUORI", biasanya ditampilkan dengan ukuran font terbesar untuk menarik perhatian. Setelah itu, ada elemen <p>Mahasiswi Pendidikan Komputer</p>, yaitu sebuah paragraf pendek yang berfungsi untuk memberikan deskripsi singkat tentang "DUORI",

*<nav>*

*<a href="#tentang">Tentang</a>*

*<a href="#proyek">Proyek</a>\*

*<a href="#kontak">Kontak</a>*

*</nav>*

Kode membuat sebuah navigasi menggunakan elemen <nav>, yang berfungsi untuk menampung tautan-tautan (link) yang akan membantu pengguna berpindah ke bagian-bagian tertentu di halaman web. Di dalam <nav>, terdapat tiga buah elemen <a>, masing-masing berisi atribut href yang menunjuk ke id bagian tertentu di halaman: #tentang, #proyek, dan #kontak. Ini berarti ketika pengguna mengklik salah satu tautan tersebut, halaman akan secara otomatis menggulir ke bagian yang sesuai dengan id yang dituju. Misalnya, klik pada "Tentang" akan membawa pengguna ke elemen dengan id="tentang".

*<div class="container">*

*<section class="tentang">*

*<h2>Tentang Rizqa Aulia Faiha</h2>*

*<p>Saya adalah mahasiswa yang sedang belajar membuat website.*

*Saya menyukai dunia teknologi dan ingin menjadi web devaloper.</p>*

*</section>*

*<section class="tentang">*

*<h2>Tentang Rizqa Oktavia Ramadhani</h2>*

*<p>Saya adalah mahasiswa yang sedang belajar membuat website.*

*Saya menyukai dunia teknologi dan ingin menjadi web devaloper.</p>*

*</section>*

*</div>*

Kode membuat sebuah <div> dengan kelas container, yang berfungsi sebagai pembungkus untuk mengelompokkan beberapa elemen sekaligus agar lebih mudah diatur tata letaknya, terutama dengan bantuan CSS. Di dalam <div> tersebut terdapat dua buah <section> yang masing-masing juga memiliki kelas tentang. Setiap <section> berisi informasi tentang dua orang yang berbeda.

Section pertama memuat heading <h2> dengan teks "Tentang Rizqa Aulia Faiha", diikuti dengan paragraf <p> yang berisi deskripsi singkat: menyatakan bahwa Rizqa Aulia Faiha adalah seorang mahasiswa yang sedang belajar membuat website, menyukai dunia teknologi, dan bercita-cita menjadi web developer. Begitu pula section kedua, yang memuat informasi serupa tentang "Rizqa Oktavia Ramadhani".

*</div>*

*<div class="container">*

*<section class="tentang">*

*<h2>Tentang Kami</h2>*

*<p>Kami adalah individu-individu yang memiliki keterampilan kuat dalam bidang coding dan*

*pengembangan perangkat lunak, dengan pengalaman di bidang backend dan frontend*

*development. Dalam pengelolaan backend, kami terbiasa mengelola server, database, dan*

*API, sementara di sisi frontend, kami berpengalaman dalam desain antarmuka dan*

*pengembangan pengalaman pengguna. Selain keahlian teknis, kami juga memiliki latar*

*belakang di bidang jurnalisme, baik sebagai penulis maupun dalam kegiatan media*

*lainnya. Dengan pengalaman luas dalam mengajar, membimbing, serta aktif sebagai*

*panitia di berbagai acara UKM, kami terus mengasah kemampuan dalam organisasi,*

*koordinasi, serta kerja kolaboratif dalam tim.*

*</p>*

*</section>*

*</div>*

Kode kembali membuat sebuah <div> dengan kelas container, yang berfungsi untuk mengelompokkan konten dalam struktur yang rapi dan konsisten dengan bagian sebelumnya. Di dalam <div> ini, terdapat satu <section> dengan kelas tentang, yang berisi heading <h2> bertuliskan "Tentang Kami". Ini menandakan bahwa isi dari bagian ini adalah deskripsi umum tentang kelompok atau tim yang bersangkutan. Dan tag p beisu deskripsi tentang DUORI.

*<div class="container">*

*<section id="proyek">*

*<h2>Proyek Rizqa Aulia Faiha</h2>*

*<ul>*

*<li>Personal Book Tracker</li>*

*<li>Simple Blog Platform</li>*

*<li>Quiz App</li>*

*<li>Digital Resume Generator</li>*

*<li>Chat Application (Basic)</li>*

*<li>Music Playlist Organizer</li>*

*</ul>*

*</section>*

Kode menggunakan <div> dengan kelas container untuk membungkus konten, menjaga konsistensi struktur tata letak halaman. Di dalamnya terdapat sebuah <section> dengan atribut id="proyek", yang berarti bagian ini akan menjadi target navigasi jika pengguna mengklik tautan "Proyek" di menu navigasi atas. Ini memudahkan perpindahan cepat ke bagian proyek.

Di dalam <section>, terdapat heading <h2> bertuliskan "Proyek Rizqa Aulia Faiha", yang memperkenalkan daftar proyek yang pernah atau sedang dikerjakan oleh individu tersebut. Selanjutnya, digunakan elemen <ul> (unordered list) untuk membuat daftar tidak berurutan. Daftar ini berisi beberapa <li> (list item) yang masing-masing mendeskripsikan proyek

*<section id="proyek">*

*<h2>Proyek Rizqa Oktavia Ramadhani</h2>*

*<ul>*

*<li>To-Do List App</li>*

*<li>Mini E-commerce Catalog</li>*

*<li>Movie Watchlist</li>*

*<li>Expense Tracker</li>*

*<li>Weather Dashboard</li>*

*<li>URL Shortener</li>*

*</ul>*

*</section>*

*</div>*

Kode menggunakan <div> dengan kelas container untuk membungkus konten, menjaga konsistensi struktur tata letak halaman. Di dalamnya terdapat sebuah <section> dengan atribut id="proyek", yang berarti bagian ini akan menjadi target navigasi jika pengguna mengklik tautan "Proyek" di menu navigasi atas. Ini memudahkan perpindahan cepat ke bagian proyek.

Di dalam <section>, terdapat heading <h2> bertuliskan "Proyek Rizqa Oktavia Ramadhani", yang memperkenalkan daftar proyek yang pernah atau sedang dikerjakan oleh individu tersebut. Selanjutnya, digunakan elemen <ul> (unordered list) untuk membuat daftar tidak berurutan. Daftar ini berisi beberapa <li> (list item) yang masing-masing mendeskripsikan proyek

*<div class="container">*

*<section id="proyek">*

*<h2>Proyek Kami</h2>*

*<ul>*

*<li>E-Commerce Platform untuk Penjualan Peralatan Rumah Tangga</li>*

*<li>Sistem untuk Toko Kopi Online</li>*

*<li>Platform Pencarian Lowongan Kerja</li>*

*<li>Dashboard Keuangan dan Bank Pribadi</li>*

*<li>Sistem Pembelian dan Dokumentasi Buku Secara Online</li>*

*<li>Platform Pembelajaran Online</li>*

*<li>Aplikasi Pemesanan Makanan Online</li>*

*<li>E-Commerce Platform untuk Penjualan Pakaian</li>*

*<li>Dashboard Keuangan Pribadi</li>*

*<li>Platform Streaming Musik Online</li>*

*</ul>*

Kode kembali membuat <div> dengan kelas container untuk mengelompokkan konten ke dalam struktur yang rapi. Di dalam <div> ini terdapat <section> baru dengan id="proyek", sama seperti section sebelumnya. Namun, penggunaan id yang sama sebaiknya dihindari, karena dalam HTML, setiap id harus unik untuk menghindari kebingungan saat navigasi atau styling dengan CSS.

Di dalam <section> ini, terdapat heading <h2> dengan teks "Proyek Kami", yang memperkenalkan daftar proyek yang dikerjakan secara tim. Daftar proyek ditulis menggunakan elemen <ul> (unordered list), di mana masing-masing item proyek berada di dalam elemen <li>.

*<section class="galeri">*

*<h2>Galeri Proyek</h2>*

*<div class="galeri-images">*

*<div class="galeri-item">*

*<img src="imgProfile/imgt1.jpeg" alt="MyHomeStore.com">*

*<div class="text-overlay">Proyek My Home Store</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt2.jpeg" alt="CoffeeCheckOut.com">*

*<div class="text-overlay">Proyek Coffee Check Out</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt3.jpeg" alt="JobFindersHub.com">*

*<div class="text-overlay">Proyek Job Finders Hub</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt4.jpeg" alt="FinTrackr.com">*

*<div class="text-overlay">Proyek Fin Trackr</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt5.jpeg" alt="BookShelfStore.com">*

*<div class="text-overlay">Proyek Book Shelf Store</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt6.jpeg" alt="LearnXpert.com">*

*<div class="text-overlay">Proyek LearnXpert</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt7.jpeg" alt="GrubOrderNow.com">*

*<div class="text-overlay">Proyek Grub Order Now</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt8.jpeg" alt="FashionBuy.com">*

*<div class="text-overlay">Proyek Fashion Buy</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt9.jpeg" alt="MyWealthTracker.com">*

*<div class="text-overlay">Proyek My Wealth Tracker</div>*

*</div>*

*<div class="galeri-item">*

*<img src="imgProfile/imgt10.jpeg" alt="BeatFlow.com">*

*<div class="text-overlay">Proyek Beat Flow</div>*

*</div>*

*</div>*

*</section>*

*</section>*

*</div>*

Kode melanjutkan galeri proyek dengan menambahkan lebih banyak gambar yang mewakili berbagai proyek yang telah dikerjakan. Di dalam elemen <section class="galeri">, terdapat heading <h2> yang memberikan penjelasan bahwa ini adalah bagian dari "Galeri Proyek".

Di dalam <div class="galeri-images">, setiap proyek yang ditampilkan diwakili oleh sebuah <div class="galeri-item">. Setiap item berisi elemen <img> yang menampilkan gambar proyek, dengan atribut alt untuk memberikan deskripsi singkat mengenai proyek tersebut (seperti "MyHomeStore.com", "CoffeeCheckOut.com", dan lain-lain). Setiap gambar juga dilengkapi dengan elemen <div class="text-overlay">, yang berfungsi untuk menambahkan teks yang menjelaskan nama proyek di atas gambar, seperti "Proyek My Home Store", "Proyek Coffee Check Out", dan seterusnya.

Ada juga satu item galeri tambahan, yaitu proyek "Beat Flow", yang ditampilkan dengan gambar dari file imgt10.jpeg, menambah variasi lebih lanjut pada galeri proyek. Secara keseluruhan, galeri ini memberikan visual yang sangat informatif bagi pengunjung situs tentang proyek-proyek yang telah dilakukan, dilengkapi dengan gambar yang memperjelas tampilan setiap aplikasi atau sistem yang telah dikembangkan.

*<div class="container">*

*<section id="kontak">*

*<h2>Kontak</h2>*

*<p>Email: rizqaauliafaiha@gmail.com</p>*

*<p>Instagram: @ra.faiha411\_</p>*

*</section>*

*<section id="kontak">*

*<h2>Kontak</h2>*

*<p>Email: rizqaoktaviaramadhani@gmail.com</p>*

*<p>Instagram: @ikkana\_n0n3</p>*

*</section>*

*</div>*

Kode membuat dua elemen <section> yang memiliki id="kontak", yang berfungsi untuk menampilkan informasi kontak dari dua individu. Meskipun kedua bagian ini memiliki fungsi yang sama, yaitu menampilkan informasi kontak, terdapat kesalahan dalam penggunaan atribut id. Dalam HTML, setiap id harus unik, namun di sini kedua section menggunakan id="kontak", yang bisa menyebabkan konflik atau kebingunguan saat melakukan navigasi atau styling. Sebaiknya, setiap section memiliki id yang berbeda, misalnya id="kontak1" untuk yang pertama dan id="kontak2" untuk yang kedua, atau lebih baik lagi menggunakan kelas (class) jika tidak perlu ada perbedaan pengelolaan antara keduanya.

Setiap <section> dimulai dengan sebuah <h2> yang berjudul "Kontak", menunjukkan bahwa bagian ini menyediakan informasi kontak. Selanjutnya, ada dua buah paragraf <p>, yang masing-masing menyebutkan alamat email dan akun Instagram dari dua individu yang berbeda. Informasi ini memungkinkan pengunjung situs untuk menghubungi mereka melalui email atau media sosial.

*<footer>*

*<p>&copy; 2025 DUORI</p>*

*</footer>*

*</body>*

*</html>*

Kode menambahkan elemen <footer> yang berfungsi untuk menampilkan informasi footer di bagian bawah halaman. Di dalam <footer>, terdapat sebuah paragraf <p> yang berisi teks "© 2025 DUORI", yang menunjukkan hak cipta dan tahun pembuatan situs. Teks ini menandakan bahwa konten di situs ini adalah milik DUORI, dan tahun 2025 menunjukkan tahun pembaruan atau pembuatan situs tersebut.

Bagian ini diakhiri dengan penutupan tag </body> dan </html>, yang menandakan bahwa HTML ini telah selesai. Tag </body> menutup seluruh konten halaman, sementara </html> menandakan akhir dari dokumen HTML itu sendiri.

Dengan demikian, halaman ini selesai dengan struktur yang jelas dan rapi, dimulai dari header, konten utama, galeri proyek, hingga informasi kontak dan hak cipta di bagian footer.

File style.css:

Kita buat file dengan nama style.css agar memuat kode untuk memperbagus tampilan isi web nanti:

*body {*

*font-family: 'Trebuchet MS', Tahoma, Verdana, sans-serif;*

*margin: 0;*

*padding: 0;*

*background-color: #f0f2f5;*

*color: #333;*

*}*

Kode menetapkan beberapa aturan dasar untuk elemen <body> pada halaman web. Pertama, Anda memilih jenis font yang digunakan dengan properti font-family, yang mencantumkan 'Trebuchet MS' sebagai font utama, diikuti dengan font fallback seperti Tahoma, Verdana, dan akhirnya font default sans-serif jika font sebelumnya tidak tersedia. Selanjutnya, properti margin: 0; digunakan untuk menghapus margin default yang diterapkan oleh browser di sekitar elemen <body>, memberikan kontrol penuh atas jarak antara elemen halaman. Selain itu, properti padding: 0; menghilangkan padding default yang ada di dalam elemen <body>, yang biasanya memberikan ruang di dalam elemen, agar Anda bisa menentukan jarak antar elemen secara lebih presisi. Terakhir, dengan menetapkan background-color: #f0f2f5;, Anda memberikan warna latar belakang yang cerah, membuat halaman lebih bersih dan nyaman dibaca, sementara color: #333; mengatur warna teks menjadi abu-abu gelap agar kontras dengan latar belakang yang lebih terang.

*header {*

*background-color: #4a90e2;*

*color: white;*

*text-align: center;*

*padding: 2em 1em;*

*box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);*

*}*

Kode mendefinisikan gaya untuk elemen <header>. Pertama, properti background-color: #4a90e2; memberikan warna latar belakang biru terang pada header, menciptakan tampilan yang menarik dan profesional. Warna teks diatur dengan color: white;, yang memberikan kontras yang baik terhadap latar belakang biru, memastikan teks tetap mudah dibaca. Dengan menggunakan text-align: center;, teks dalam header, seperti judul atau deskripsi, akan diposisikan di tengah secara horizontal, memberikan kesan simetris dan rapi. Properti padding: 2em 1em; memberi ruang di dalam header, dengan jarak vertikal lebih besar (2em) dan jarak horizontal lebih kecil (1em), menciptakan keseimbangan visual antara teks dan tepi header. Terakhir, box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); menambahkan bayangan lembut di bawah header.

*nav {*

*background-color: #3b74c5;*

*text-align: center;*

*padding: 1em 0;*

*}*

Kode mendefinisikan gaya untuk elemen <nav>. Properti background-color: #3b74c5; memberikan warna latar belakang biru yang sedikit lebih gelap dibandingkan dengan warna header, menciptakan kesan konsistensi desain yang rapi di seluruh halaman. Dengan text-align: center;, Anda memastikan bahwa teks atau tautan navigasi di dalam elemen <nav> akan terpusat secara horizontal, memberikan tampilan yang lebih teratur dan mudah dinavigasi. Sementara itu, padding: 1em 0; memberikan ruang di dalam elemen navigasi, dengan jarak vertikal 1em di atas dan bawah, menciptakan ruang yang cukup untuk elemen navigasi tanpa terasa terlalu sesak.

*nav a {*

*color: white;*

*text-decoration: none;*

*margin: 0 15px;*

*font-weight: bold;*

*}*

Kode mendefinisikan gaya untuk elemen <a> (tautan) yang ada di dalam elemen <nav>. Properti color: white; mengatur warna teks tautan menjadi putih, memberikan kontras yang baik dengan latar belakang biru dari elemen navigasi, sehingga tautan mudah terlihat. Dengan text-decoration: none;, Anda menghapus garis bawah default yang biasanya muncul pada tautan, memberi kesan lebih bersih dan modern. Sementara itu, margin: 0 15px; memberikan jarak horizontal sebesar 15px di antara tautan-tautan navigasi, menciptakan ruang yang cukup antara satu tautan dengan tautan lainnya, membuatnya terlihat lebih terpisah dan tidak terlalu rapat. Terakhir, font-weight: bold; memberikan ketebalan pada teks tautan.

*nav a:hover {*

*text-decoration: underline;*

*}*

Kode menetapkan gaya untuk elemen tautan (<a>) di dalam elemen navigasi (<nav>) ketika kursor berada di atas tautan, menggunakan selector nav a:hover. Properti text-decoration: underline; memberikan efek garis bawah pada tautan saat pengguna mengarahkan kursor ke atasnya.

*section {*

*max-width: 1000px;*

*margin: 0.5em auto;*

*padding: 0.8em;*

*background-color: white;*

*border-radius: 10px;*

*box-shadow: 0 3px 6px rgba(0, 0, 0, 0.03);*

*}*

Kode Pada bagian CSS ini, Anda memberikan gaya untuk elemen <section>, yang umumnya digunakan untuk membagi konten dalam bagian-bagian yang terpisah di halaman. Properti max-width: 1000px; membatasi lebar maksimum elemen <section> menjadi 1000px, sehingga elemen ini tidak akan melebihi ukuran tersebut, meskipun berada di layar yang lebih besar. Dengan margin: 0.5em auto;, Anda memberi jarak vertikal sebesar 0.5em di atas dan bawah elemen, serta mengatur margin kiri dan kanan ke auto untuk memusatkan elemen di tengah halaman, menciptakan tampilan yang simetris dan rapi.

padding: 0.8em; memberikan ruang internal di sekitar konten dalam elemen <section>, sehingga teks atau elemen lain tidak terlalu menempel pada tepi elemen, membuat konten lebih mudah dibaca. Warna latar belakang diatur dengan background-color: white;, memberi kesan bersih dan kontras yang baik terhadap warna latar belakang halaman. Properti border-radius: 10px; memberikan sudut yang melengkung pada elemen, memberi kesan lebih lembut dan modern pada desain. Terakhir, box-shadow: 0 3px 6px rgba(0, 0, 0, 0.03); menambahkan bayangan halus di sekitar elemen <section>, memberikan efek kedalaman yang subtil dan membuat elemen tersebut tampak terangkat dari halaman.

*section h2 {*

*font-size: 1.8em;*

*text-align: center;*

*border-bottom: 2px solid #ddd;*

*padding-bottom: 0.5em;*

*margin-bottom: 1em;*

*}*

Kode memberikan gaya untuk elemen <h2> yang ada di dalam elemen <section>. Properti font-size: 1.8em; mengatur ukuran font judul menjadi lebih besar dari ukuran font default, memberikan penekanan pada heading agar lebih menonjol dan mudah dibaca. Dengan text-align: center;, teks judul akan diposisikan di tengah, menciptakan kesan simetris dan seimbang dalam desain.

Properti border-bottom: 2px solid #ddd; menambahkan garis bawah dengan ketebalan 2px dan warna abu-abu terang (#ddd) di bawah judul, yang memberikan pemisah visual antara judul dan konten di bawahnya, serta memberi elemen desain yang lebih rapi dan terstruktur. padding-bottom: 0.5em; memberikan ruang di bawah teks judul agar garis bawah tidak terlalu dekat dengan teks, menciptakan jarak yang nyaman secara visual. Terakhir, margin-bottom: 1em; memberikan jarak tambahan di bawah elemen <h2>, memastikan bahwa ada ruang yang cukup antara judul dan konten berikutnya.

ul {

    padding-left: 20px;

}

Kode menetapkan gaya untuk elemen <ul> (daftar tidak terurut). Properti padding-left: 20px; memberikan ruang internal sebesar 20px di sisi kiri elemen <ul>. Ini berarti bahwa setiap item dalam daftar akan memiliki sedikit jarak dari tepi kiri elemen induknya, memberikan ruang agar teks tidak terlalu dekat dengan margin kiri dan membuat tampilan daftar lebih rapi.

*footer {*

*background-color: #4a90e2;*

*color: white;*

*text-align: center;*

*padding: 1em;*

*margin-top: 3em;*

*font-size: 0.9em;*

*}*

Kode Pada bagian CSS ini, Anda memberikan gaya untuk elemen <footer>. Properti background-color: #4a90e2; memberikan warna latar belakang biru yang konsisten dengan elemen-elemen lainnya di halaman, menciptakan desain yang harmonis. Warna teks diatur dengan color: white;, memastikan teks di dalam footer terlihat jelas dan kontras dengan latar belakang biru.

Dengan text-align: center;, teks dalam footer akan terpusat secara horizontal, memberikan tampilan yang rapi dan simetris. padding: 1em; menambahkan ruang di dalam footer, memberikan jarak antara teks dan tepi footer sehingga terlihat lebih lega dan nyaman dibaca. Sementara itu, margin-top: 3em; memberikan jarak yang cukup di atas footer, memisahkannya dengan konten sebelumnya agar tampilan halaman lebih teratur.

Terakhir, font-size: 0.9em; mengatur ukuran font di footer sedikit lebih kecil dari ukuran default, menciptakan perbedaan visual yang jelas antara konten utama dan informasi footer.

*.container {*

*display: flex;*

*flex-wrap: wrap;*

*justify-content: space-between;*

*gap: 12px;*

*max-width: 1000px;*

*margin: 0.3em auto;*

*padding: 0.5em;*

*}*

Kode memberikan gaya untuk elemen dengan kelas .container, yang menggunakan model layout flexbox. Properti display: flex; mengubah elemen .container menjadi kontainer flex, yang memungkinkan Anda untuk mengatur posisi dan aliran elemen di dalamnya dengan lebih fleksibel. Dengan flex-wrap: wrap;, Anda memastikan bahwa jika elemen-elemen di dalam kontainer melebihi lebar yang tersedia, mereka akan membungkus ke baris berikutnya, menciptakan tata letak yang responsif.

justify-content: space-between; mengatur agar elemen-elemen di dalam kontainer didistribusikan secara merata dengan ruang yang sama di antara mereka, tetapi tanpa ruang di ujung kontainer, memberikan tampilan yang terorganisir dan efisien. Properti gap: 12px; menambahkan jarak sebesar 12px antara elemen-elemen dalam kontainer, memastikan bahwa mereka tidak terlalu rapat dan ada ruang yang cukup antar elemen.

Dengan max-width: 1000px;, Anda membatasi lebar maksimal kontainer menjadi 1000px, sehingga kontainer tidak akan melebar lebih dari itu, memberikan kontrol terhadap tampilan pada layar besar. Sementara itu, margin: 0.3em auto; memusatkan kontainer di halaman dengan jarak vertikal kecil sebesar 0.3em di atas dan bawah, dan margin kiri-kanan otomatis untuk menjaga simetri. Terakhir, padding: 0.5em; memberikan ruang di dalam kontainer agar elemen-elemen di dalamnya tidak terlalu menempel pada tepi kontainer, menciptakan kesan yang lebih rapi dan terorganisir.

*.container section {*

*flex: 1 1 45%;*

*box-sizing: border-box;*

*}*

Kode mendefinisikan gaya untuk elemen <section> di dalam kontainer dengan kelas .container. Dengan flex: 1 1 45%;, Anda mengatur elemen <section> untuk menggunakan flexbox, yang memungkinkan elemen tersebut tumbuh dan menyusut sesuai dengan ukuran ruang yang tersedia. Angka pertama (1) mengindikasikan bahwa elemen dapat tumbuh jika ada ruang tambahan, sementara angka kedua (1) menunjukkan bahwa elemen juga dapat menyusut jika ruang terbatas. Nilai 45% pada bagian ketiga menetapkan lebar dasar elemen menjadi 45% dari lebar kontainer, sehingga memungkinkan dua elemen <section> untuk berada di samping satu sama lain pada baris yang sama, dengan sedikit ruang antar keduanya.

Properti box-sizing: border-box; memastikan bahwa padding dan border elemen dihitung dalam lebar total elemen tersebut. Artinya, lebar 45% yang ditetapkan untuk elemen sudah mencakup padding dan border, bukan hanya konten elemen saja. Ini membantu dalam menjaga tata letak yang konsisten dan mencegah elemen melampaui lebar kontainer yang telah ditetapkan.

*.galeri {*

*text-align: center;*

*margin: 2em auto;*

*max-width: 1000px;*

*}*

Kode memberikan gaya untuk elemen dengan kelas .galeri. Properti text-align: center; memastikan bahwa konten di dalam elemen .galeri, seperti gambar atau teks, akan terpusat secara horizontal, menciptakan tampilan yang lebih rapi dan simetris. Dengan margin: 2em auto;, Anda menambahkan jarak vertikal sebesar 2em di atas dan bawah elemen, serta memusatkan elemen di halaman dengan margin kiri dan kanan otomatis, menciptakan keseimbangan visual di seluruh halaman.

Selain itu, max-width: 1000px; membatasi lebar maksimal elemen .galeri menjadi 1000px, yang berarti galeri tidak akan meluas melebihi ukuran tersebut meskipun berada di layar yang lebih besar. Ini memberikan kontrol terhadap tampilan galeri, memastikan bahwa elemen-elemen di dalamnya tetap terorganisir dan tidak terdistorsi.

*.galeri img{*

*width: 100%;*

*max-width: 250px;*

*height: auto;*

*border-radius: 8px;*

*}*

Kode Pada bagian CSS ini, Anda memberikan gaya untuk elemen gambar (<img>) di dalam kelas .galeri. Properti width: 100%; memastikan bahwa gambar akan mengambil seluruh lebar elemen induknya, membuat gambar responsif dan menyesuaikan dengan lebar kontainer tempat gambar tersebut berada. Namun, dengan max-width: 250px;, Anda membatasi lebar maksimum gambar menjadi 250px, memastikan gambar tidak akan melebihi ukuran tersebut meskipun ruang yang tersedia lebih besar. Hal ini berguna untuk menjaga proporsi dan memastikan gambar tidak terlalu besar dan mengganggu tata letak.

height: auto; menjaga proporsi gambar agar tetap seimbang, mengatur tinggi gambar secara otomatis berdasarkan lebar yang ditentukan, sehingga gambar tidak akan terdistorsi. Terakhir, border-radius: 8px; memberikan sudut melengkung pada gambar, menciptakan tampilan yang lebih lembut dan modern, yang meningkatkan estetika desain keseluruhan.

*.galeri h2 {*

*font-size: 1.8em;*

*margin-bottom: 1em;*

*color: #333;*

*}*

Kode memberikan gaya untuk elemen <h2> di dalam kelas .galeri. Properti font-size: 1.8em; meningkatkan ukuran font dari judul menjadi lebih besar, memberikan penekanan dan membuatnya lebih mencolok di halaman. Ukuran ini memastikan judul tetap jelas terlihat dan memberikan hierarki visual yang lebih baik dalam desain.

Dengan margin-bottom: 1em;, Anda menambahkan jarak vertikal antara judul dan konten di bawahnya, menciptakan ruang yang cukup agar tampilan tidak terasa terlalu padat dan lebih teratur. Terakhir, color: #333; mengatur warna teks menjadi abu-abu gelap, yang memberikan kontras yang baik dengan latar belakang.

*.galeri-images {*

*display: grid;*

*grid-template-columns: repeat(3, 1fr);*

*gap: 20px;*

*justify-items: center;*

*}*

Kode memberikan gaya untuk elemen dengan kelas .galeri-images yang menggunakan model layout grid. Properti display: grid; mengubah elemen menjadi kontainer grid, yang memungkinkan Anda mengatur tata letak elemen anak (dalam hal ini gambar-gambar dalam galeri) menggunakan grid.

Dengan grid-template-columns: repeat(3, 1fr);, Anda menentukan bahwa kontainer grid akan memiliki tiga kolom yang lebar masing-masing kolomnya sama (1fr), di mana "fr" adalah satuan yang mewakili bagian dari ruang yang tersedia. Artinya, setiap kolom akan mendapatkan proporsi yang sama dari ruang yang tersedia, dan dalam hal ini ada tiga kolom yang ditentukan.

gap: 20px; menambahkan jarak sebesar 20px antara setiap elemen grid, baik secara horizontal maupun vertikal, menciptakan ruang yang cukup agar gambar-gambar dalam galeri tidak saling tumpang tindih dan tampak terpisah dengan baik.

Terakhir, justify-items: center; memastikan bahwa setiap item (gambar) dalam grid akan diposisikan di tengah kolomnya secara horizontal, memberikan tampilan yang simetris dan rapi.

*.galeri-item {*

*position: relative;*

*width: 100%;*

*height: auto;*

*border-radius: 8px;*

*transition: transform 0.3s ease;*

*}*

Kode memberikan gaya untuk elemen dengan kelas .galeri-item, yang biasanya digunakan untuk setiap item dalam galeri gambar. Properti position: relative; mengatur posisi elemen secara relatif terhadap posisi defaultnya. Ini memungkinkan Anda untuk menempatkan elemen anak (seperti teks atau overlay) di dalam .galeri-item menggunakan properti posisi lain, seperti position: absolute;, tanpa mengubah posisi dasar dari elemen tersebut.

width: 100%; memastikan bahwa setiap item galeri akan mengisi seluruh lebar kontainer yang ada, memberikan fleksibilitas tata letak yang responsif. Sementara itu, height: auto; menjaga proporsi gambar atau konten dalam elemen .galeri-item sehingga tidak terdistorsi.

Dengan border-radius: 8px;, Anda memberikan sudut melengkung pada setiap item galeri, menciptakan tampilan yang lebih lembut dan modern. Terakhir, transition: transform 0.3s ease; menambahkan efek transisi halus pada properti transformasi, yang memungkinkan perubahan seperti perbesaran atau pergeseran elemen dengan animasi yang mulus selama 0.3 detik, memberikan efek interaktif yang menyenangkan saat pengguna berinteraksi dengan elemen tersebut (misalnya, saat hover).

.galeri-item:hover {

    transform: scale(1.05);

}

Kode memberikan gaya untuk elemen dengan kelas .galeri-item saat berada dalam keadaan hover (ketika pengguna mengarahkan kursor mouse ke atas elemen). Properti transform: scale(1.05); memperbesar ukuran elemen sebesar 5% (dengan nilai 1.05), memberikan efek zoom halus pada gambar atau item galeri saat pengguna berinteraksi dengan elemen tersebut.

Efek ini memberikan umpan balik visual yang menarik, menyoroti elemen yang sedang di-hover. Efek ini bekerja dengan lancar karena sudah ada transisi yang ditetapkan sebelumnya (transition: transform 0.3s ease;), yang memungkinkan perubahan ukuran terjadi secara bertahap dalam 0.3 detik, menciptakan pengalaman pengguna yang lebih halus dan dinamis.

*.galeri-item:hover .text-overlay {*

*display: block;*

*font-size: 0.7em*

*}*

Kode memberikan gaya untuk elemen dengan kelas .text-overlay yang berada di dalam elemen .galeri-item ketika pengguna mengarahkan kursor (hover) pada elemen tersebut.

display: block; mengubah tampilan elemen .text-overlay dari keadaan tersembunyi (biasanya display: none; dalam kondisi normal) menjadi elemen yang tampil dalam aliran dokumen, memaksa teks overlay untuk muncul saat pengguna melakukan hover pada item galeri. Hal ini memberikan efek interaktif di mana informasi tambahan, seperti nama proyek atau deskripsi, muncul di atas gambar.

font-size: 0.7em; mengatur ukuran font dari teks overlay menjadi lebih kecil, yaitu 70% dari ukuran font induknya. Ini memberikan tampilan yang lebih halus dan tidak mengganggu visual gambar, sambil tetap menjaga teks tetap terbaca.

*.galeri-item .text-overlay {*

*position: absolute;*

*top: 50%;*

*left: 50%;*

*color: white;*

*font-size: 1.5em;*

*font-weight: bold;*

*background-color: rgba(0, 0, 0, 0.5);*

*padding: 0.5em;*

*display: none;*

*}*

Kodedengan kelas .text-overlay di dalam .galeri-item diatur untuk tampil sebagai overlay teks di atas gambar dalam galeri. Dengan properti position: absolute;, overlay diposisikan secara absolut relatif terhadap elemen induknya yang memiliki position: relative;, memungkinkan penempatan overlay yang tepat di dalam setiap item galeri. top: 50%; dan left: 50%; memposisikan overlay di tengah gambar baik secara vertikal maupun horizontal. Namun, pengaturan posisi ini masih memerlukan penyesuaian untuk memastikan elemen benar-benar berada di tengah.

color: white; mengubah warna teks menjadi putih agar dapat terlihat jelas di atas gambar dengan latar belakang yang bervariasi. Ukuran font overlay diatur dengan font-size: 1.5em; untuk memastikan teks terlihat besar dan jelas, sementara font-weight: bold; memberikan ketebalan pada teks untuk menambah visibilitas. Untuk latar belakang teks, background-color: rgba(0, 0, 0, 0.5); memberikan warna hitam dengan tingkat transparansi 50%, sehingga teks tetap terlihat jelas tanpa menghalangi gambar sepenuhnya.

padding: 0.5em; memberikan ruang di sekitar teks, menciptakan efek estetis yang lebih rapi. Secara default, display: none; menyembunyikan overlay, dan hanya akan muncul saat elemen .galeri-item di-hover, mengubah propertinya menjadi display: block; pada keadaan hover.

*@media (max-width: 768px) {*

*.container {*

*flex-direction: column;*

*align-items: center;*

*}*

*.container section {*

*width: 100%;*

*}*

*.galeri-images {*

*grid-template-columns: 1fr 1fr;*

*}*

*}*

Kode menggunakan aturan media query untuk menyesuaikan tampilan desain agar responsif pada perangkat dengan lebar layar maksimum 768 piksel, seperti tablet atau smartphone. Aturan @media (max-width: 768px) memastikan bahwa gaya ini hanya diterapkan pada perangkat dengan lebar layar yang lebih kecil. Di dalam media query ini, perubahan pertama adalah mengatur .container untuk menggunakan flex-direction: column;, yang mengubah arah susunan elemen dari baris menjadi kolom, membuat elemen-elemen dalam kontainer disusun secara vertikal saat tampilan di layar kecil. Selain itu, align-items: center; memastikan elemen-elemen tersebut diratakan di tengah secara horizontal, menciptakan tampilan yang lebih simetris dan rapi. Untuk setiap elemen di dalam kontainer, width: 100%; memastikan bahwa setiap bagian akan mengisi seluruh lebar kontainer, menghindari masalah seperti elemen yang terpotong atau tumpang tindih. Pada bagian .galeri-images, aturan grid-template-columns: 1fr 1fr; mengubah layout grid dari tiga kolom menjadi dua kolom, yang memberikan ruang lebih banyak untuk gambar dan membuatnya lebih mudah dilihat pada perangkat dengan layar kecil. Semua perubahan ini dirancang untuk meningkatkan pengalaman pengguna di perangkat mobile dan tablet.

*@media (max-width: 480px) {*

*.galeri-images {*

*grid-template-columns: 1fr;*

*}*

*}*

Kode menggunakan media query dengan aturan @media (max-width: 480px) yang diterapkan pada perangkat dengan lebar layar maksimum 480 piksel, seperti ponsel pintar. Di dalam media query ini, terdapat perubahan pada elemen .galeri-images, di mana grid-template-columns: 1fr; mengubah layout grid dari dua kolom (yang sebelumnya diatur pada media query dengan lebar layar maksimum 768px) menjadi satu kolom saja. Hal ini memastikan bahwa gambar-gambar dalam galeri akan ditampilkan secara vertikal, satu per satu, mengisi seluruh lebar kontainer. Pendekatan ini memberikan pengalaman visual yang lebih optimal pada layar yang lebih kecil dengan menyesuaikan jumlah kolom, sehingga gambar-gambar tampil lebih besar dan lebih mudah dilihat pada perangkat dengan layar kecil seperti ponsel.

Cara membuat sebuah respository di github dan menambahkan teman:

1. Buka browser dan kunjungi laman GitHub.
2. Login ke akun GitHub dengan cara menakan tombol sign in di sudut kanan atas halaman Github lalu masukkan username dan password dan daftar
3. Tekan tombol + di sudut kanan atas dan pilih new respository.
4. Lalu klik create respository untuk membuar repository baru.
5. Masuk ke pengaturan repository dan klik tab settings yang ada di bagain atas.
6. Pilih manage access dan di sidebar kiri
7. Lalu klik invite a a collaborator.
8. Tuliskan nama pengguna teman pada kolom pencarian yang muncul setelah menemukannya klik nama teman dan tekan tombol add.
9. Jika teman sudah menerima undangan maka bisa lanjut ke tahap untuk saling berbagi file kode dan saling merevisi.

Analisis pembahasan:

HTML (HyperText Markup Language) dan CSS (Cascading Style Sheets) adalah dua teknologi inti yang digunakan dalam pengembangan situs web. Meskipun keduanya berfungsi untuk tujuan yang berbeda, mereka bekerja sama untuk membangun dan mendesain halaman web yang fungsional dan estetis. HTML adalah bahasa markup yang digunakan untuk struktur dasar halaman web. HTML menyusun konten dengan menggunakan elemen-elemen berbentuk tag, seperti <html>, <body>, <header>, <footer>, <section>, dan sebagainya. Elemen-elemen ini membentuk struktur konten dan memberikan makna atau fungsi tertentu, misalnya teks, gambar, tautan, tabel, dan lain-lain. HTML tidak menentukan bagaimana konten tersebut ditampilkan atau diposisikan di layar—itu adalah tugas CSS. HTML bekerja seperti kerangka atau struktur halaman, di mana semua elemen seperti teks, gambar, dan video diletakkan.

CSS adalah bahasa yang digunakan untuk mendesain dan menyusun tata letak halaman web, termasuk mengubah warna, ukuran, font, margin, posisi, dan banyak lagi. CSS bekerja dengan memberikan gaya pada elemen-elemen yang didefinisikan dalam HTML, memungkinkan Anda untuk menata halaman web agar lebih menarik dan mudah digunakan. CSS dapat diterapkan secara langsung dalam tag HTML (inline), di dalam tag <style> di bagian <head>, atau melalui file eksternal yang di-link ke halaman HTML. CSS menggunakan selektor untuk menargetkan elemen-elemen HTML tertentu dan mendefinisikan gaya yang diinginkan.

List kode tag html:

1. <html>: Menandai elemen root dari dokumen HTML.
2. <head>: Berisi informasi metadata tentang dokumen, seperti pengaturan karakter encoding dan judul halaman.
3. <meta>: Menyediakan metadata tentang halaman, seperti pengaturan karakter atau pengaturan viewport.
4. <title>: Menentukan judul halaman yang muncul di tab browser.
5. <body>: Berisi semua konten yang akan ditampilkan di halaman web.
6. <header>: Digunakan untuk mendefinisikan bagian header dari halaman web.
7. <nav>: Menandai bagian dari halaman yang berisi navigasi.
8. <section>: Digunakan untuk mendefinisikan bagian dari dokumen yang terkait dengan suatu tema.
9. <h1>, <h2>, <h3>, ...: Digunakan untuk membuat judul atau subjudul dengan ukuran yang berbeda.
10. <p>: Digunakan untuk menampilkan teks dalam bentuk paragraf.
11. <ul>: Digunakan untuk membuat daftar tak terurut (bullet list).
12. <ol>: Digunakan untuk membuat daftar terurut (angka).
13. <li>: Digunakan untuk mendefinisikan elemen dalam daftar (<ul> atau <ol>).
14. <a>: Digunakan untuk membuat hyperlink.
15. <img>: Digunakan untuk menampilkan gambar di halaman.
16. <footer>: Digunakan untuk mendefinisikan bagian bawah halaman.
17. <div>: Digunakan untuk mengelompokkan elemen-elemen lain dan sering digunakan dengan CSS.
18. <span>: Digunakan untuk menandai teks kecil di dalam elemen lain.
19. <form>: Digunakan untuk membuat formulir input pengguna.
20. <input>: Digunakan untuk membuat elemen input di dalam formulir.
21. <button>: Digunakan untuk membuat tombol interaktif.

List kode tag css:

1. border-radius: Digunakan untuk membuat sisi-sisi elemen menjadi melengkung (rounded corners).
2. font-family: Digunakan untuk menentukan jenis font yang digunakan dalam elemen.
3. background-color: Digunakan untuk mengatur warna latar belakang dari elemen.
4. color: Digunakan untuk mengatur warna teks dalam elemen.
5. margin: Digunakan untuk memberikan jarak di luar elemen.
6. padding: Digunakan untuk memberikan jarak di dalam elemen, antara konten dan batas elemen.
7. text-align: Digunakan untuk mengatur perataan teks di dalam elemen.
8. box-shadow: Digunakan untuk memberikan efek bayangan di sekitar elemen.
9. display: Digunakan untuk menentukan model tata letak elemen (misalnya flex, grid, block).
10. flex: Digunakan dalam tata letak flexbox untuk menentukan fleksibilitas elemen.
11. justify-content: Digunakan dalam flexbox untuk mengatur distribusi ruang antara elemen.
12. align-items: Digunakan dalam flexbox untuk mengatur perataan elemen di sepanjang sumbu vertikal.
13. gap: Digunakan untuk menentukan jarak antar elemen dalam tata letak flexbox atau grid.
14. width: Digunakan untuk mengatur lebar elemen.
15. height: Digunakan untuk mengatur tinggi elemen.
16. max-width: Digunakan untuk mengatur lebar maksimal elemen.
17. font-size: Digunakan untuk mengatur ukuran teks.
18. font-weight: Digunakan untuk mengatur ketebalan teks (misalnya bold).
19. text-decoration: Digunakan untuk menambahkan efek dekorasi pada teks (misalnya underline).
20. transition: Digunakan untuk memberi efek transisi pada perubahan properti elemen.
21. position: Digunakan untuk mengatur posisi elemen di halaman (misalnya relative, absolute).
22. top, left, right, bottom: Digunakan untuk mengatur posisi elemen yang diposisikan secara absolut atau relatif.
23. grid-template-columns: Digunakan dalam tata letak grid untuk menentukan jumlah dan ukuran kolom.
24. @media: Digunakan untuk menentukan aturan CSS yang berlaku hanya pada kondisi tertentu, seperti lebar layar tertentu.

|  |
| --- |
| Kesimpulan |