

# LAPORAN

MATA KULIAH PEMROGRAMAN WEB 1

DOSEN PENGAMPU:

M. Miftakhul Syaikhudin, S.Kom. M.Kom.



OLEH:

Muhammad Genio Brilliant

4123009

JURUSAN SISTEM INFORMASI

UNIVERSITAS PESANTREN TINGGI DARUL 'ULUM

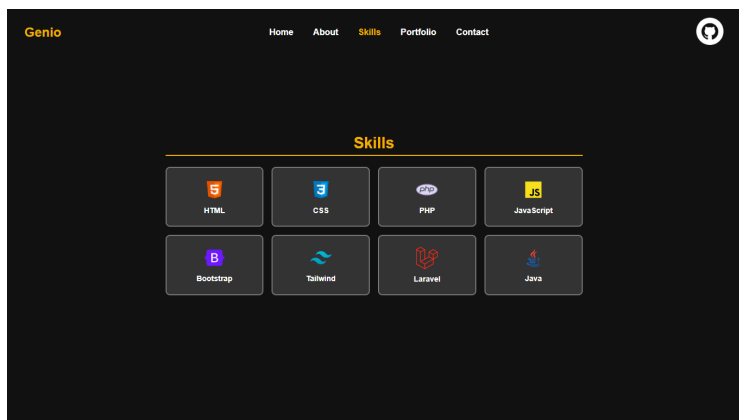
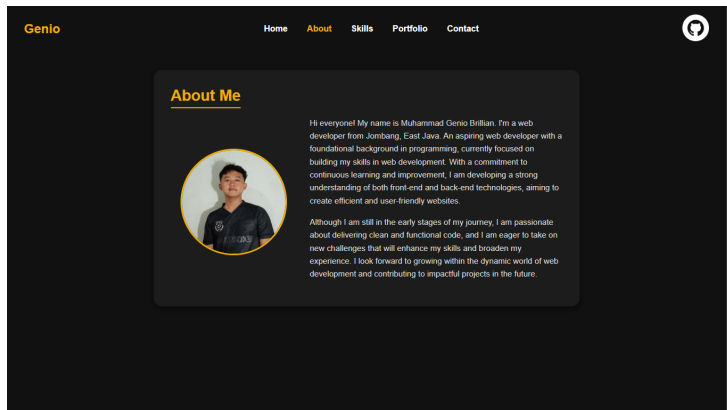
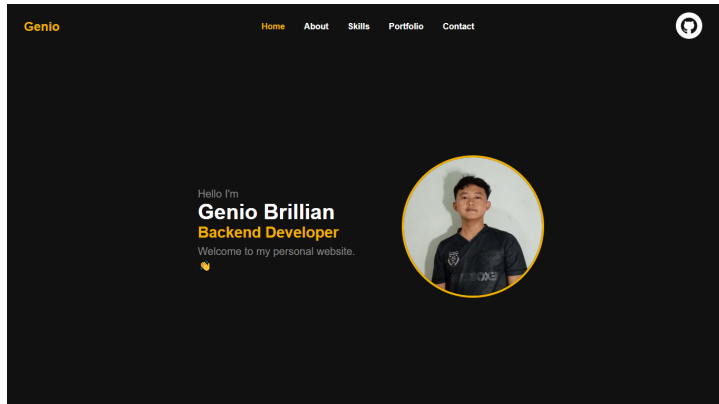
JOMBANG

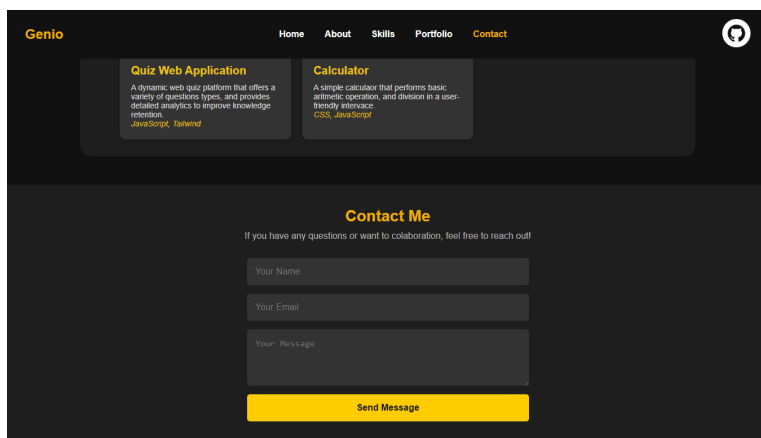
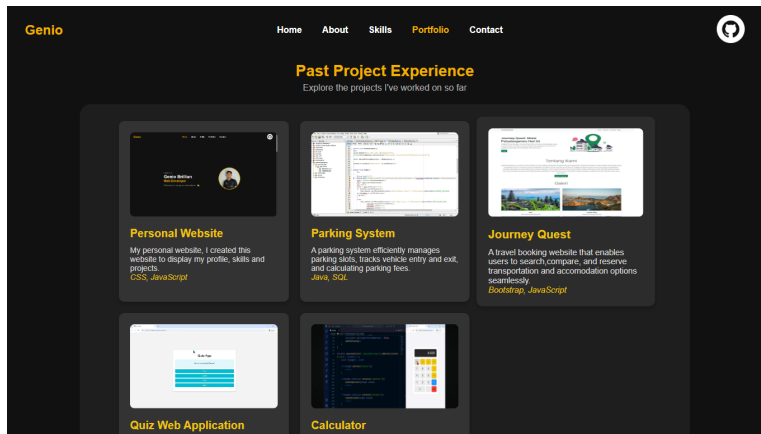
2024

Link Github:

<https://github.com/geniobrillian/Pemrograman-Web1/tree/main/Minggu-ke6/web-portofolio>

Output:





## penjelasan HTML:

### 1. Bagian Head

Judul Situs: `<title>Genio Portfolio</title>` menampilkan judul "Genio Portfolio" di tab browser.

Tautan ke CSS: `<link rel="stylesheet" href="css/style.css">` menyambungkan file CSS eksternal untuk mengatur tampilan situs.

### 2. Bagian Header dan Navigasi

Header: Menggunakan elemen `<header>` untuk menyimpan navbar atau menu navigasi.

Navbar: Menu ini berisi logo di kiri, tautan ke beberapa bagian halaman (Home, About, Skills, Portfolio, Contact), serta ikon GitHub di kanan atas yang mengarah langsung ke profil GitHub Genio. Setiap tautan di dalam `<nav>` mengarahkan pengguna ke bagian tertentu dalam satu halaman ini.

### 3. Bagian Home

Perkenalan Singkat: Bagian ini perkenalan dengan teks "Hello I'm Genio Brilliant" serta animasi teks di `<span id="typing-text">Web</span>`. Selain itu, terdapat gambar profil di sebelah teks perkenalan untuk memberikan kesan lebih personal. Bagian ini menggunakan `<section id="home">`, yang membuat tampilan terfokus dan menarik saat pertama kali pengunjung membuka situs.

#### 4. Bagian About

Deskripsi Diri: Bagian ini memperkenalkan diri dengan detail. Deskripsi ini disertai dengan gambar dan teks yang menjelaskan latar belakang, minat, dan tujuan Genio sebagai pengembang web.

Struktur Tampilan: Dibagi menjadi dua bagian utama—gambar di sebelah kiri dan teks di sebelah kanan—untuk menjaga keseimbangan visual. Teks yang panjang dipecah menjadi paragraf, membuat informasi lebih mudah dibaca.

#### 5. Bagian Skills

Keterampilan yang Dimiliki: Bagian ini menampilkan keterampilan dalam bentuk kartu-kartu yang berisi logo dan nama teknologi yang dikuasai Genio, seperti HTML, CSS, PHP, JavaScript, dan lainnya.

Desain Kartu: Setiap keterampilan dikemas dalam `<div class="skill-card">` yang memudahkan pengguna untuk melihat teknologi yang dikuasai Genio dengan sekilas, sehingga memberi kesan profesional dan teratur.

#### 6. Bagian Portfolio

Pengalaman Proyek: Bagian ini menunjukkan proyek-proyek yang pernah dikerjakan Genio, seperti Personal Website, Parking System, dan Journey Quest. Setiap proyek diberi judul, deskripsi singkat, gambar, dan teknologi yang digunakan.

Desain Grid: Proyek-proyek ini ditampilkan dalam format grid yang rapi, memberikan kesan bahwa Genio memiliki pengalaman yang beragam di berbagai teknologi. Desain grid ini memudahkan pengunjung menjelajahi proyek dengan cepat.

#### 7. Bagian Contact

Formulir Kontak: Bagian ini menyediakan formulir yang memungkinkan pengunjung mengirimkan pesan langsung ke Genio. Terdapat kolom untuk mengisi nama, email, dan pesan, serta tombol "Send Message" untuk mengirim.

Desain Formulir: Formulir ini dirancang dengan elemen input sederhana yang memudahkan pengguna untuk mengisi data. Bagian ini memungkinkan calon klien atau kolega menghubungi Genio dengan mudah.

## 8. Script JavaScript Eksternal

Tautan JavaScript: `<script src="js/script.js"></script>` menghubungkan file JavaScript eksternal

## Penjelasan CSS:

### 1. (Global Style)

Selector \*: Mengatur margin dan padding semua elemen menjadi nol dan menerapkan box-sizing: border-box; agar batas elemen dihitung dalam ukuran elemen itu sendiri. Ini membantu menjaga konsistensi dan mencegah tampilan yang tidak diinginkan.

body: Menggunakan font Arial dengan warna teks putih dan latar belakang hitam. Ini menciptakan tampilan kontras yang nyaman di mata.

### 2. Header dan Navigasi

.navbar: Mengatur posisi navigasi di bagian atas halaman dengan gaya fixed agar selalu terlihat saat halaman di-scroll. Navbar diatur dengan latar belakang hitam dan elemen di dalamnya disejajarkan menggunakan display: flex.

.nav-link: Teks pada tautan navigasi berwarna putih dan berubah menjadi warna emas (#f0b000) saat di-hover atau dalam keadaan aktif, memberikan penekanan pada tautan yang dipilih.

### 3. Bagian Home

#home: Bagian ini didesain untuk menampilkan teks pengantar dan foto profil secara seimbang dengan tata letak flexbox. Tinggi 100vh memastikan seluruh bagian menutupi layar penuh, memberikan kesan tampilan awal yang menarik.

.profile-pic img: Foto profil ditampilkan dalam bentuk lingkaran dengan border emas, membuatnya terlihat lebih menonjol dan profesional.

### 4. Bagian About

.about-section: Latar belakang abu-abu gelap dengan bayangan (shadow) ringan membuat bagian ini terasa terpisah dari latar belakang utama. Judul dibuat lebih besar dan diberi garis bawah berwarna emas untuk menonjolkan bagian.

.about-image dan .about-text: Menggunakan flexbox untuk menyusun gambar dan teks berdampingan. Gambar profil di sini juga dibuat melingkar dengan border yang konsisten.

### 5. Bagian Skills

.skills-section: Diatur dalam bentuk grid untuk menampilkan ikon keterampilan dalam empat

kolom. Hal ini memberikan tampilan yang teratur dan mudah dibaca.

.skill-card: Masing-masing kartu keterampilan menggunakan border abu-abu dengan efek zoom ringan saat di-hover, menambah kesan interaktif.

## 6. Bagian Portfolio

.projects-section: Bagian ini menampilkan proyek dalam tiga kolom grid, memungkinkan beberapa proyek tampil rapi dalam satu baris. Latar belakang abu-abu gelap memberikan kesan serius dan elegan.

.project-card: Setiap kartu proyek memiliki efek transformasi dan perubahan warna saat di-hover, sehingga menarik perhatian pengguna saat menjelajah proyek.

## 7. Bagian Contact

.contact-section: Bagian ini diatur dengan latar belakang abu-abu tua dan teks putih. Judul dan teks tambahan diberi warna emas untuk menyesuaikan tema warna keseluruhan.

.contact-form: Formulir kontak untuk kemudahan pengguna, dengan input dan textarea yang memiliki gaya minimalis dan efek perubahan warna saat fokus.

# Penjelasan Javascript:

## 1. Efek Mengetik pada Subtitle

Bagian ini mengatur efek mengetik pada elemen teks dengan id="typing-text" di halaman, di mana teks akan menampilkan kata yang berbeda secara bergantian ("Web Developer" dan "Backend Developer").

Variabel Awal:

words menyimpan daftar kata yang akan ditampilkan.

wordIndex dan letterIndex mengatur indeks kata dan huruf yang sedang ditampilkan.

currentText untuk menyimpan kata saat ini yang sedang ditampilkan.

isDeleting digunakan untuk menentukan apakah karakter sedang dihapus (untuk efek mengetik ulang).

Fungsi type():

Jika isDeleting bernilai true, kode akan mengurangi jumlah huruf pada currentText. Jika false, kode menambah huruf pada currentText.

Setelah semua huruf pada kata tertentu selesai ditampilkan, `isDeleting` menjadi `true`, dan setelah semua huruf terhapus, indeks kata (`wordIndex`) diubah untuk menampilkan kata berikutnya.

Fungsi ini menggunakan `setTimeout()` dengan berbagai jeda waktu untuk memberikan efek mengetik dan menghapus yang halus.

Event Listener untuk `DOMContentLoaded`:

Kode ini memastikan bahwa fungsi `type()` mulai berjalan setelah konten halaman selesai dimuat.

## 2. Pergantian Link Aktif di Navbar

Bagian ini mengatur agar tautan navigasi (`.nav-link`) aktif saat diklik, memberikan indikator visual untuk link yang sedang dipilih pengguna.

Query Selector dan Event Listener:

`navLinks` memilih semua elemen dengan kelas `.nav-link`.

Untuk setiap link, event listener ditambahkan sehingga ketika sebuah tautan diklik, kelas `active` akan diberikan pada tautan tersebut dan menghapus kelas `active` dari tautan lainnya.

Ini memungkinkan pengguna mengetahui bagian mana yang sedang aktif berdasarkan gaya yang diterapkan pada tautan yang dipilih.

## 3. Efek Scroll Halus (Smooth Scroll)

Bagian ini mengaktifkan scroll halus saat pengguna mengklik tautan navigasi untuk berpindah ke bagian halaman yang berbeda.

Event Listener pada `.nav-link`:

Setiap anchor elemen dengan kelas `.nav-link` dipasang event listener `click`.

Saat diklik, `preventDefault()` mencegah perilaku standar, yaitu langsung loncat ke bagian target.

Dengan `scrollIntoView()`, halaman akan menggulir ke bagian target (`targetSection`) dengan efek scroll yang halus (`behavior: 'smooth'`), menciptakan transisi yang lebih nyaman bagi pengguna.