

Report Website HTML



Disusun oleh:

Ramadhan Naufal Rahmantio

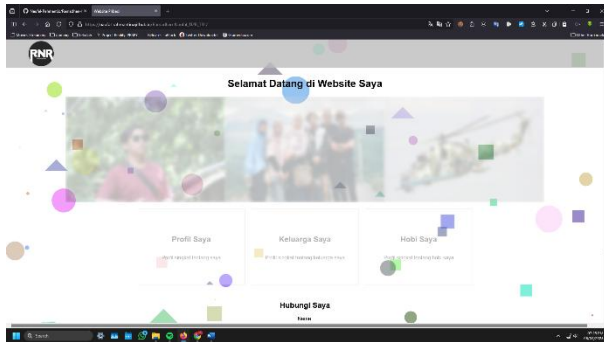
(2307411042)

Politeknik Negeri Jakarta

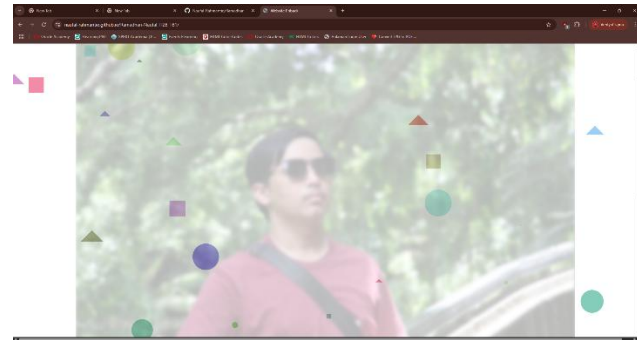
2025

1. Index.html

Firefox



Chrome



Jika page index.html di buka dalam browser selain Firefox elemen photo tidak bersusun sampingan tetapi menjadi full size dan mendorong photo lainnya keluar frame.

<pre><!DOCTYPE html> <html lang="id"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Beranda</title> <link rel="stylesheet" href="Index.css"> </head></pre>	<p>Bagian <code><head></code> berisi informasi metadata tentang halaman. <code>charset="UTF-8"</code> digunakan agar halaman bisa menampilkan berbagai karakter dengan benar. <code>viewport</code> membuat halaman responsif di berbagai ukuran layar, terutama mobile. Tag <code><title></code> menampilkan teks "Beranda" di tab browser. Terakhir, file CSS <code>Index.css</code> dihubungkan untuk memberikan gaya visual pada halaman.</p>
<pre><body> <canvas id="backgroundCanvas"></canvas> <div class="navbar"> <div class="nav-links"> Beranda </div> </div></pre>	<p>Bagian <code><body></code> dimulai dengan elemen <code><canvas></code> yang akan digunakan untuk menampilkan animasi latar. Elemen ini diidentifikasi dengan <code>id="backgroundCanvas"</code> dan akan dikendalikan melalui JavaScript di file <code>script.js</code>. Bagian ini membuat navigasi atas (navbar) dengan dua elemen utama: logo di kiri dan link ke halaman "Beranda" di kanan. Ini memungkinkan pengguna untuk mengakses halaman utama dengan mudah, dan logo memberikan identitas visual.</p>
<pre><div class="container"> <div class="grid"> <div class="box">Profil</div> <div class="box">Keluarga</div> <div class="box">Hobi</div> </div> </div></pre>	<p>Bagian ini menampilkan tiga kotak utama di halaman, masing-masing berisi tautan ke halaman yang relevan. Struktur ini menggunakan sistem grid, yang memungkinkan tampilan responsif dan rapi di berbagai ukuran layar.</p>
<pre><script src="script.js"></script> </body> </html></pre>	<p>Di akhir halaman, file <code>script.js</code> dimuat untuk menjalankan animasi pada elemen canvas di latar belakang. Penempatan script di bagian akhir body memastikan semua elemen HTML telah dimuat sebelum JavaScript dijalankan.</p>

2. Index.css

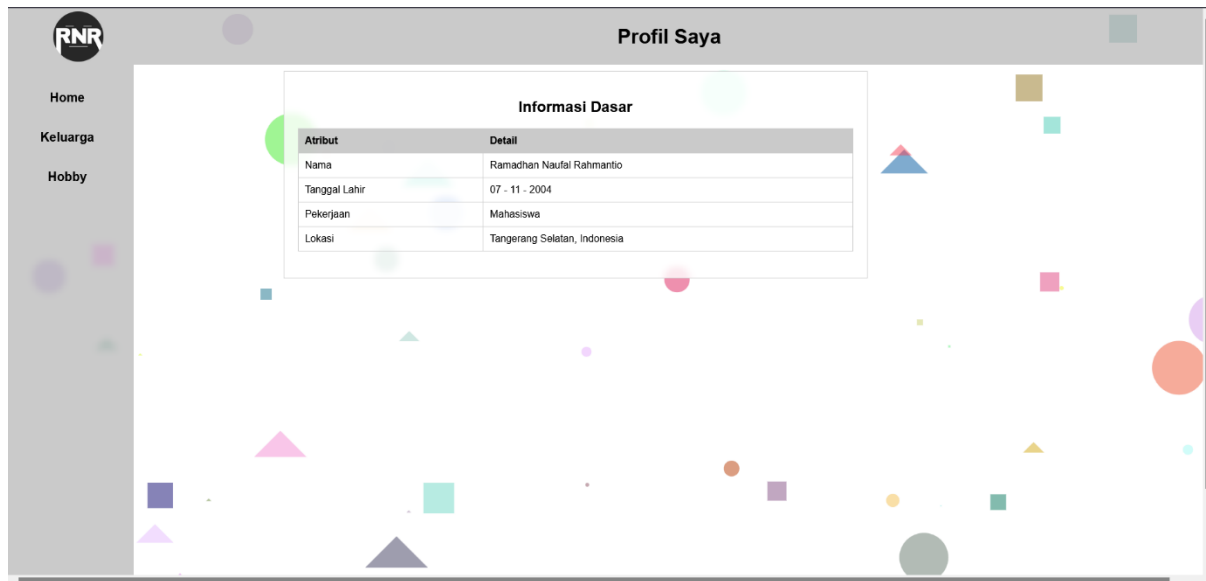
<pre>body { margin: 0; font-family: Arial, sans-serif; background-color: #111; color: white; }</pre>	<p>Blok ini mengatur tampilan dasar seluruh halaman. <code>margin: 0</code> menghilangkan jarak default di tepi halaman, sementara <code>font-family</code> mengatur jenis huruf agar tampak modern dan mudah dibaca. Warna latar belakang diatur menjadi hitam keabu-abuan (<code>#111</code>), dengan teks berwarna putih untuk kontras yang kuat.</p>
<pre>.navbar { display: flex; justify-content: space-between; align-items: center; background-color: #222; padding: 10px 20px; position: fixed; width: 100%; top: 0; z-index: 1000; }</pre>	<p>Bagian ini mengatur tampilan navbar. Menggunakan <code>flex</code> agar elemen logo dan link sejajar secara horizontal. <code>justify-content: space-between</code> membuat jarak antara logo dan link maksimal. <code>position: fixed</code> memastikan navbar tetap berada di atas saat halaman digulir, dan <code>z-index: 1000</code> menempatkannya di lapisan paling atas.</p>
<pre>.logo { width: 50px; } .nav-links a { color: white; text-decoration: none; margin-left: 15px; }</pre>	<p>Mengatur ukuran gambar logo menjadi 50 piksel agar tidak terlalu besar dan menjaga keseimbangan visual dalam navbar.</p> <p>Menghilangkan garis bawah default pada link (<code>text-decoration: none</code>) dan memberi jarak antar link agar tampak rapi.</p>
<pre>.box { background-color: #444; padding: 40px; text-align: center; border-radius: 10px; transition: background-color 0.3s; } .box:hover { background-color: #555; }</pre>	<p>Kotak-kotak konten diberi warna latar abu gelap, padding yang cukup, dan sudut melengkung (<code>border-radius</code>). Saat kursor diarahkan (<code>hover</code>), akan terjadi transisi halus pada warna latar. Efek hover ini mengubah warna kotak jadi sedikit lebih terang saat disentuh, memberikan kesan interaktif.</p>

3. script.js

<pre>const canvas = document.getElementById("backgroundCanvas"); const ctx = canvas.getContext("2d");</pre>	<p>Kode ini memilih elemen canvas dengan <code>id="backgroundCanvas"</code> dan mengambil konteks 2D-nya melalui <code>getContext("2d")</code>. Konteks ini digunakan untuk menggambar objek grafis seperti partikel di dalam canvas.</p>
<pre>canvas.width = window.innerWidth; canvas.height = window.innerHeight;</pre>	<p>Di sini, ukuran canvas disesuaikan agar selalu sebesar ukuran jendela browser pengguna. Hal ini memastikan bahwa animasi menutupi seluruh latar belakang halaman.</p>
<pre>window.addEventListener("resize", function () { canvas.width = window.innerWidth; canvas.height = window.innerHeight; });</pre>	<p>Event listener ini menjamin bahwa jika pengguna mengubah ukuran jendela (resize), ukuran canvas akan otomatis ikut menyesuaikan, sehingga animasi tetap proporsional dan tidak terpotong.</p>
<pre>class Particle { constructor() { this.x = Math.random() * canvas.width; this.y = Math.random() * canvas.height; this.size = Math.random() * 2 + 1; this.speedX = Math.random() * 0.5 - 0.25; this.speedY = Math.random() * 0.5 - 0.25; } update() { this.x += this.speedX; this.y += this.speedY; // Kembalikan ke dalam layar jika keluar if (this.x < 0 this.x > canvas.width) this.speedX *= -1; if (this.y < 0 this.y > canvas.height) this.speedY *= -1; } draw() { ctx.fillStyle = "white"; ctx.beginPath(); ctx.arc(this.x, this.y, this.size, 0, Math.PI * 2); ctx.fill(); } }</pre>	<p>Kelas <code>Particle</code> ini mendefinisikan setiap titik animasi. Masing-masing partikel memiliki posisi acak, ukuran acak, serta kecepatan acak untuk bergerak di sumbu X dan Y. Fungsi <code>update()</code> membuat partikel bergerak dan memantul jika menyentuh tepi layar. Fungsi <code>draw()</code> menggambarkan titik sebagai lingkaran kecil berwarna putih.</p>
<pre>let particlesArray = []; for (let i = 0; i < 100; i++) { particlesArray.push(new Particle()); }</pre>	<p>Kode ini membuat 100 objek partikel dan menyimpannya dalam array <code>particlesArray</code>. Setiap partikel memiliki posisi dan kecepatan unik.</p>
<pre>function animate() { ctx.clearRect(0, 0, canvas.width, canvas.height);</pre>	<p>Fungsi <code>animate()</code> menjalankan animasi terus-menerus menggunakan</p>

<pre> for (let i = 0; i < particlesArray.length; i++) { particlesArray[i].update(); particlesArray[i].draw(); } requestAnimationFrame(animate); } </pre>	<p><code>requestAnimationFrame</code>, yang memberi efek halus dan efisien. Setiap frame, canvas dibersihkan (<code>clearRect</code>) dan seluruh partikel diperbarui dan digambar ulang.</p>
<pre> animate(); </pre>	<p>Baris ini menjalankan fungsi animasi saat halaman dimuat.</p>

4. Profile.html

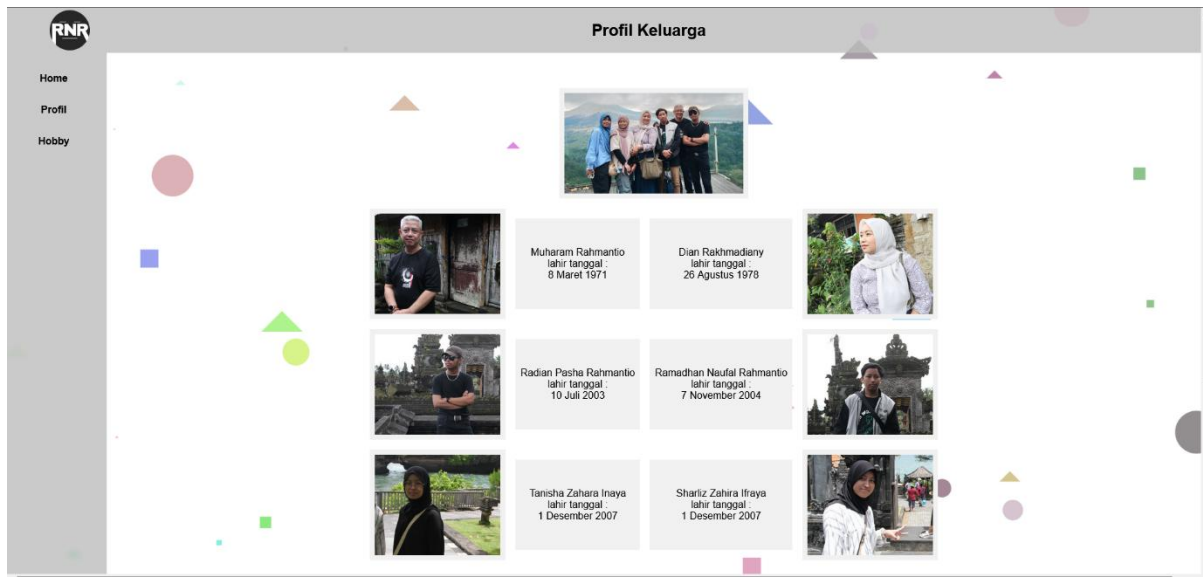


```
<div class="container profile-container">
  <div class="grid">
    <div class="probox">
      <h2>Informasi Dasar</h2>
      <table class="info-table">
        <tr>
          <th>Atribut</th>
          <th>Detail</th>
        </tr>
        <tr>
          <td>Nama</td>
          <td>Ramadhan Naufal
Rahmanto</td>
        </tr>
        <tr>
          <td>Tanggal Lahir</td>
          <td>07 - 11 - 2004</td>
        </tr>
        <tr>
          <td>Pekerjaan</td>
          <td>Mahasiswa</td>
        </tr>
        <tr>
          <td>Lokasi</td>
          <td>Tangerang Selatan,
Indonesia</td>
        </tr>
      </table>
    </div>
  </div>
</div>
```

Pada bagian ini, kita memiliki struktur HTML yang rapi dan informatif. Tag `<div class="container profile-container">` adalah pembungkus utama yang memastikan seluruh isi bagian ini tetap berada dalam batas tata letak utama halaman. Di dalamnya ada elemen `<div class="grid">` yang kemungkinan besar mengambil gaya dari sistem grid di `Index.css`, yang sebelumnya sudah dijelaskan. Elemen `<div class="probox">` berfungsi sebagai kotak atau blok untuk satu bagian informasi.

Judul "Informasi Dasar" ditampilkan menggunakan tag `<h2>`. Di bawahnya, terdapat tabel HTML (`<table class="info-table">`) yang digunakan untuk menyajikan data dalam dua kolom: "Atribut" dan "Detail". Baris pertama adalah header (`<th>`) yang memberi label kolom. Baris-baris selanjutnya (`<tr>`) memuat masing-masing informasi, seperti nama lengkap, tanggal lahir, pekerjaan, dan lokasi tempat tinggal. Penggunaan tabel di sini cocok karena menampilkan data yang bersifat pasangan label-nilai dengan struktur yang rapi dan mudah dibaca.

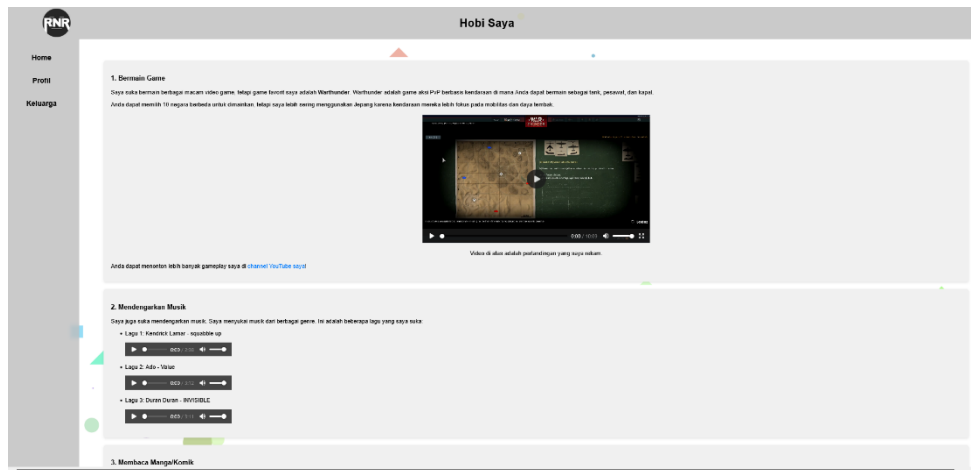
6. Keluarga.html



<pre><style> .content-keluarga { display: grid; grid-template-columns: auto auto auto auto; gap: 20px; position: fixed; top: 150px; padding: 10px; left: 26%; }</pre>	<p>Bagian CSS di atas mendefinisikan class <code>.content-keluarga</code>, yang menggunakan layout CSS Grid dengan 4 kolom otomatis (<code>auto auto auto auto</code>). Elemen ini juga memiliki gap sebesar 20px untuk memberikan jarak antar elemen, dan menggunakan <code>position: fixed</code> agar konten tetap menempel di posisi tertentu (tidak ikut scroll). Nilai <code>top: 150px</code> dan <code>left: 26%</code> mengatur posisinya dari atas dan kiri layar.</p>
<pre>.content-keluarga img { height: 200px; width: auto; background-color: #f1f1f1; text-align: center; padding: 10px; }</pre>	<p>Bagian ini mengatur tampilan semua gambar di dalam <code>.content-keluarga</code>. Tinggi gambar diatur menjadi 200px, sementara lebarnya otomatis agar proporsinya tidak rusak. Background abu-abu muda (<code>#f1f1f1</code>) dan padding 10px memberi jarak agar gambar tidak mepet ke tepi. <code>text-align: center</code> meskipun agak tidak relevan untuk gambar, mungkin dimaksudkan untuk keselarasan secara umum.</p>

<pre>.content-keluarga p { display: flex; align-items: center; justify-content: center; background-color: #f1f1f1; padding: 10px; font-size: 20px; }</pre>	<p>Paragraf yang muncul di grid ini menggunakan display: flex untuk menengahkan teks secara vertikal dan horizontal. Ini membuat setiap nama anggota keluarga dan tanggal lahirnya berada di tengah blok paragraf. Ukuran teks diperbesar menjadi 20px untuk keterbacaan, dan background-nya juga konsisten dengan elemen gambar.</p>
<pre>.family-image { grid-column: span 4; margin: 0 auto; }</pre>	<p>Class .family-image digunakan pada gambar utama keluarga (image2.jpg) agar menempati seluruh baris grid (4 kolom sekaligus). Ini dilakukan dengan grid-column: span 4. margin: 0 auto memastikan gambar berada di tengah grid.</p>
<pre><div class="content-keluarga"> <p align="center">Muharam Rahmantio
lahir tanggal :
8 Maret 1971</p> <p align="center">Dian Rakhmadiany
lahir tanggal :
26 Agustus 1978</p> <p align="center">Radian Pasha Rahmantio
lahir tanggal :
10 Juli 2003</p> <p align="center">Ramadhan Naufal Rahmantio
lahir tanggal :
7 November 2004</p> <p align="center">Tanisha Zahara Inaya
lahir tanggal :
1 Desember 2007</p> <p align="center">Sharliz Zahira Ifraya
lahir tanggal :
1 Desember 2007</p> </div></pre>	<p>Bagian ini adalah inti dari halaman keluarga. Elemen div.content-keluarga menampung susunan gambar dan teks tiap anggota keluarga. Kontennya sudah diposisikan dengan CSS Grid, jadi elemen-elemen akan tersusun rapi sesuai baris dan kolom.</p> <p>Masing-masing anggota keluarga diwakili oleh foto dan paragraf teks yang berisi nama dan tanggal lahir. Susunannya cukup unik karena teks dan gambar diselingi — misalnya, gambar ayah ditampilkan dulu, lalu diikuti teks ayah dan ibu, lalu gambar ibu, dan seterusnya. Ini menunjukkan struktur keluarga secara visual dan naratif sekaligus.</p>

7. Hobby.html



<pre>.content-hobby { margin-left: 250px; padding: 20px; padding-top: 150px; display: grid; grid-template-columns: 1fr; gap: 20px; }</pre>	<p>Class <code>.content-hobby</code> ini digunakan untuk mengatur tata letak utama dari konten hobi. Ia menggunakan sistem CSS Grid dengan satu kolom (<code>1fr</code>) dan memberi jarak antar elemen sebesar <code>20px</code> (<code>gap</code>). <code>margin-left: 250px</code> dipakai untuk mengimbangi sidebar yang tetap di kiri. Kemudian <code>padding-top: 150px</code> menggeser konten ke bawah agar tidak tertutup oleh navbar tetap.</p>
<pre>.content-hobby > div { background-color: #f0f0f0; padding: 20px; border-radius: 8px; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); }</pre>	<p>Setiap blok hobi (dibungkus oleh <code><div></code>) diberi warna latar abu terang, sudut membulat (<code>border-radius</code>), dan bayangan lembut (<code>box-shadow</code>) agar terlihat seperti kartu atau panel yang rapi dan modern.</p>
<pre>.content-hobby h2 { text-align: center; }</pre>	<p>Judul utama di setiap bagian konten hobi ditata agar berada di tengah. Meskipun dalam file ini sebenarnya yang digunakan adalah <code><h3></code>, aturan ini akan tetap berlaku karena <code>h2</code> juga mungkin digunakan di masa depan.</p>
<pre>.content-hobby video { display: block; margin: 20px auto; }</pre>	<p>Video ditampilkan sebagai blok dan ditempatkan di tengah dengan <code>margin: auto</code>, serta diberi jarak ke atas dan bawah sebesar <code>20px</code> agar tidak menempel dengan elemen lain.</p>
<pre>.content-hobby a { color: #007BFF; text-decoration: none; } .content-hobby a:hover { text-decoration: underline; }</pre>	<p>Tautan yang ada di dalam konten <code>.content-hobby</code> diberi warna biru cerah dan efek underline saat disorot (<code>hover</code>), memberikan umpan balik visual agar pengguna tahu bahwa itu bisa diklik.</p>

<h3>1. Bermain Game</h3></h3> <p>... Warthunder ...</p></p> <p><video width="640" height="360" controls> <source src="hobby.mp4" type="video/mp4"> </video></p> <p >video="" adalah="" align="center" atas="" di="" p="" p><="" pertandingan="" rekam.<="" saya="" yang=""> <p>... channel YouTube saya!</p></p> </p>	<p>Bagian ini menjelaskan hobi pertama: bermain game, khususnya Warthunder. Deskripsi diberikan dalam dua paragraf, lalu disusul dengan elemen <video> yang menampilkan video permainan pribadi pengguna (hobby.mp4). Ada teks penjelas di bawah video, dan juga tautan ke channel YouTube eksternal (target="_blank" agar terbuka di tab baru).</p>
<h3>2. Mendengarkan Musik</h3></h3> <p>... berbagai genre ...</p></p> <ul style="list-style-type: none"> <p>Lagu 1: Kendrick Lamar - squabble up</p> <audio controls> <source src="song1.mp3" type="audio/mpeg"> </audio> ... 	<p>Hobi kedua adalah mendengarkan musik. Setiap lagu dimasukkan ke dalam yang berisi <p> nama lagu dan <audio> player. Elemen <audio> memungkinkan pengguna memutar lagu langsung dari halaman. Total ada tiga lagu: dari Kendrick Lamar, Ado, dan Duran Duran.</p>
<h3>3. Membaca Manga/Komik</h3></h3> <p>Saya suka menjelajahi berbagai cerita dan gaya seni dalam manga dan komik.</p></p>	<p>Bagian terakhir cukup ringkas, hanya satu paragraf yang menjelaskan minat membaca manga dan komik.</p>