

Mini Project Series Project #2

# Social Media Landing Page

Part 2 : dark mode, transitions, and open graph metadata



progate

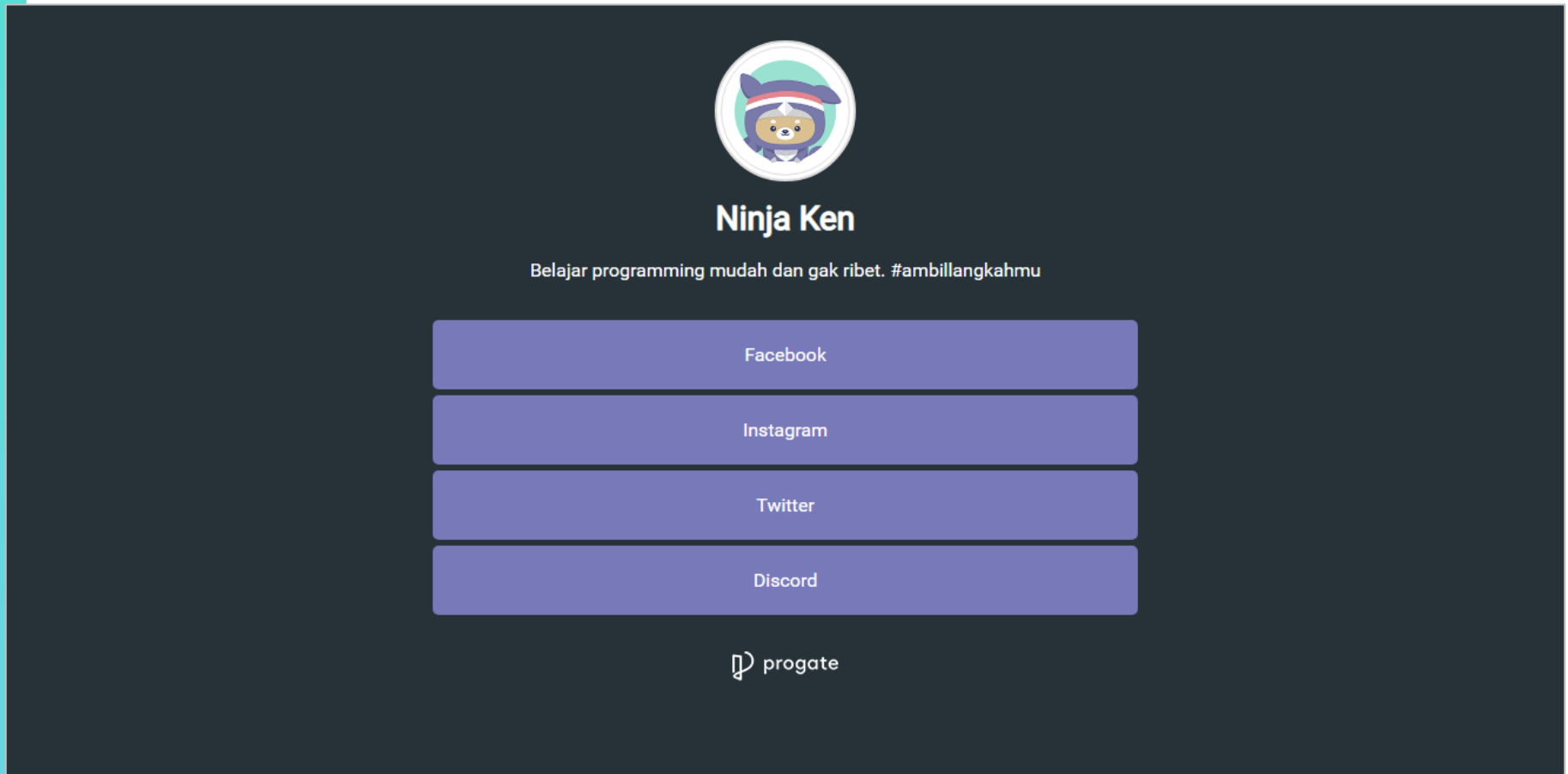
Mochammad Faizal, S.Ds., M.M., CSCU, ISM  
@moefc32 | [mf-chan.com](https://mf-chan.com)

## Yang telah kita buat sebelumnya



Jadi di mini project sebelumnya (part 1), kita sudah membuat sebuah social media landing page seperti yang terlihat pada gambar di atas.

## Yang akan kita buat kali ini



Kali ini, kita akan memodifikasi halaman tersebut sehingga memiliki fitur deteksi dark mode secara otomatis, menambahkan sedikit animasi, dan implementasi dari Open Graph Metadata. Yuk kita mulai!

Sebelum memulai, Anda harus memiliki :

- Mini project yang akan kita lanjutkan
- IDE atau text editor (bebas, di sini penulis menggunakan VSCodium)
- Akun GitHub, sebagai hosting halaman website yang akan kita buat
- File persiapan yang dapat didownload bersama panduan ini

Sebelum memulai, pastikan Anda sudah membaca dan mempelajari artikel Progate yang berjudul “**Cara Mempublikasi Halaman Web dengan GitHub**”. Artikel tersebut bisa dibaca di sini :

<https://progate.com/docs/github-pages>

Karena kita hanya akan bermain pada client-side, Anda hanya membutuhkan web browser untuk melihat hasil akhirnya.

Disarankan untuk menggunakan web browser berbasis Chromium seperti Google Chrome dan Opera, atau bisa juga menggunakan Mozilla Firefox.

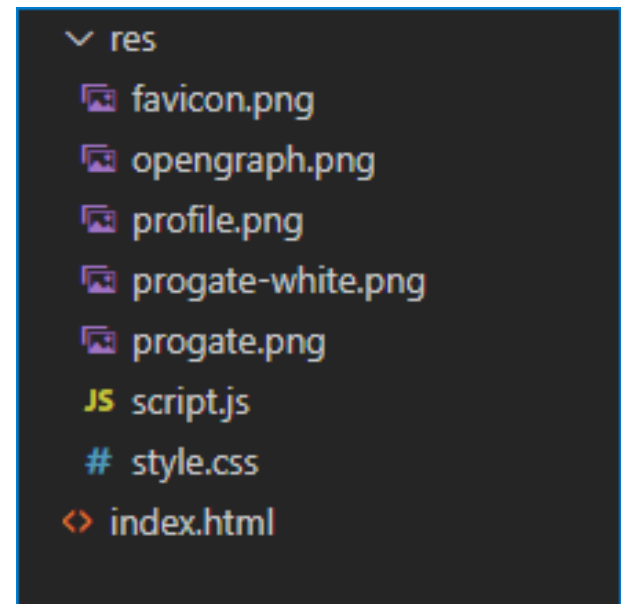
## Spesifikasi proyek

Halaman website yang akan kita buat nantinya akan memiliki fitur atau spesifikasi sebagai berikut :

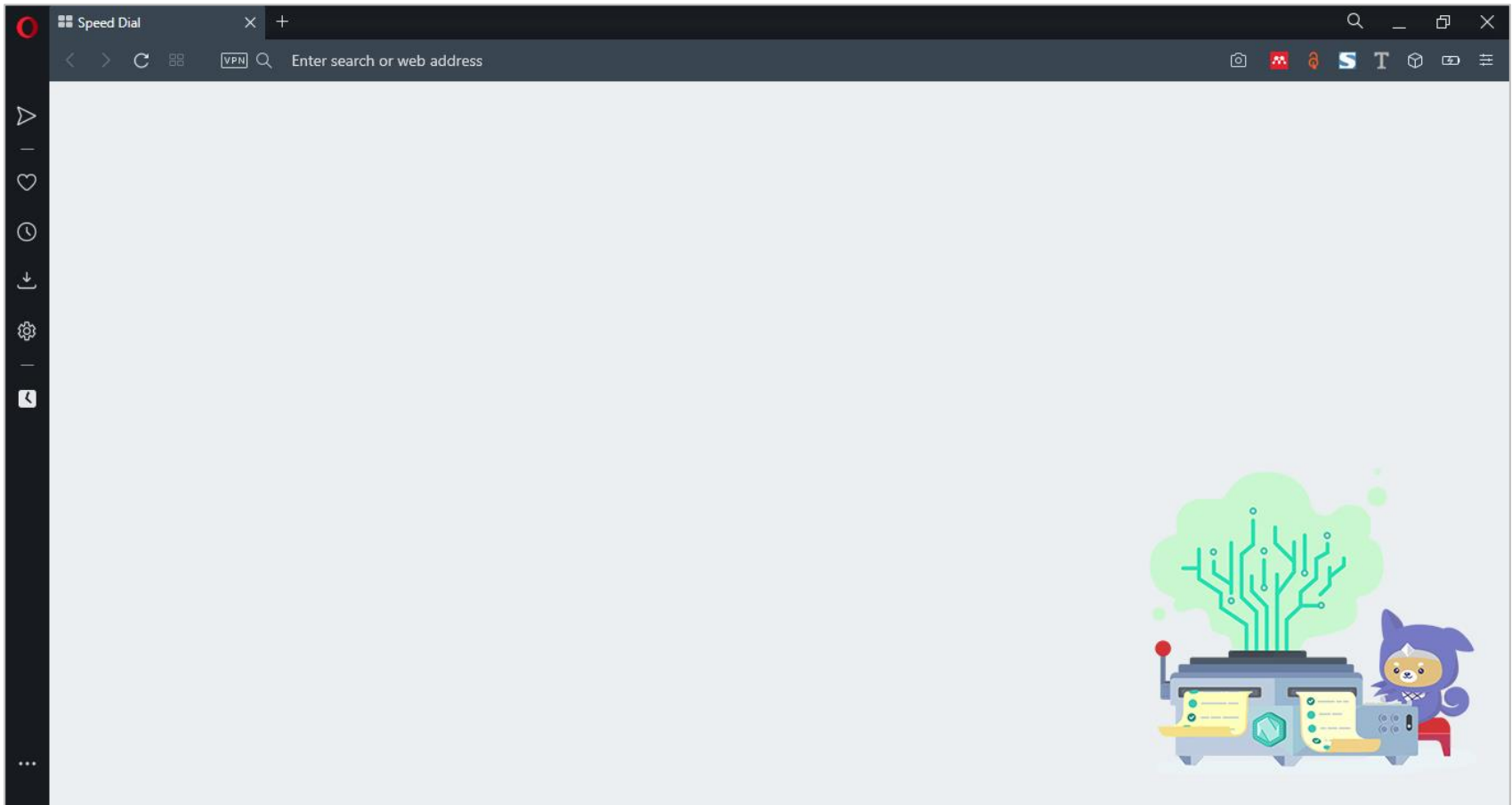
- Mengubah tampilan menjadi dark mode secara otomatis
- Mengimplementasikan CSS transitions sederhana
- Implementasi Open Graph Metadata

Adapun struktur direktori pada proyek ini sangatlah sederhana, Anda dapat melihatnya seperti pada gambar di samping. Silakan download dan ekstrak file persiapan ke dalam komputer Anda untuk melanjutkan proyek ini.

Jika Anda perhatikan, akan ada dua file baru pada proyek part 2 ini, yakni opengraph.png dan progate-white.png



## Langkah 1 : Fitur dark mode



Sesuai namanya, dark mode adalah mode di mana web browser yang kita gunakan akan menggunakan skema “gelap”, dan biasanya halaman yang ditampilkan akan menggunakan skema warna yang gelap juga (misal, hitam).

## Langkah 1 : Fitur dark mode

Fitur dark mode ini juga mungkin akan familier bagi Anda yang menggunakan perangkat mobile.

Dengan adanya fitur dark mode, antarmuka yang ditampilkan akan lebih “friendly” saat diakses dari lingkungan yang gelap (misal di saat malam hari). Halaman yang gelap biasanya akan lebih nyaman dilihat, dibanding halaman terang yang akan menyilaukan mata.

Pada gambar di samping, adalah contoh aplikasi yang menggunakan skema dark mode. Visualisasi halaman akan lebih dominan warna gelap (dalam contoh ini, warna hitam), dan teks akan menggunakan suatu warna yang kontras terhadap warna latarnya (warna terang, biasanya putih).



## Langkah 2 : Kode CSS

Pada proyek ini, kita akan mendesain halaman untuk mendeteksi secara otomatis, apakah web browser yang sedang digunakan sedang menggunakan skema ini atau tidak. Apabila ya, tampilan halaman akan berubah menjadi gelap.

Silakan buka style.css dan tambahkan kode berikut ini :

```
128 }
129
130 @media (prefers-color-scheme: dark) {
131
132     body,
133     aside {
134         background: #263238;
135         color: #fff;
136     }
137
138     img {
139         filter: grayscale(25%);
140     }
141
142     footer a {
143         background: url(progate-white.png) center center no-repeat;
144     }
145 }
```



## Checkpoint I : Halaman peringatan



### Peringatan

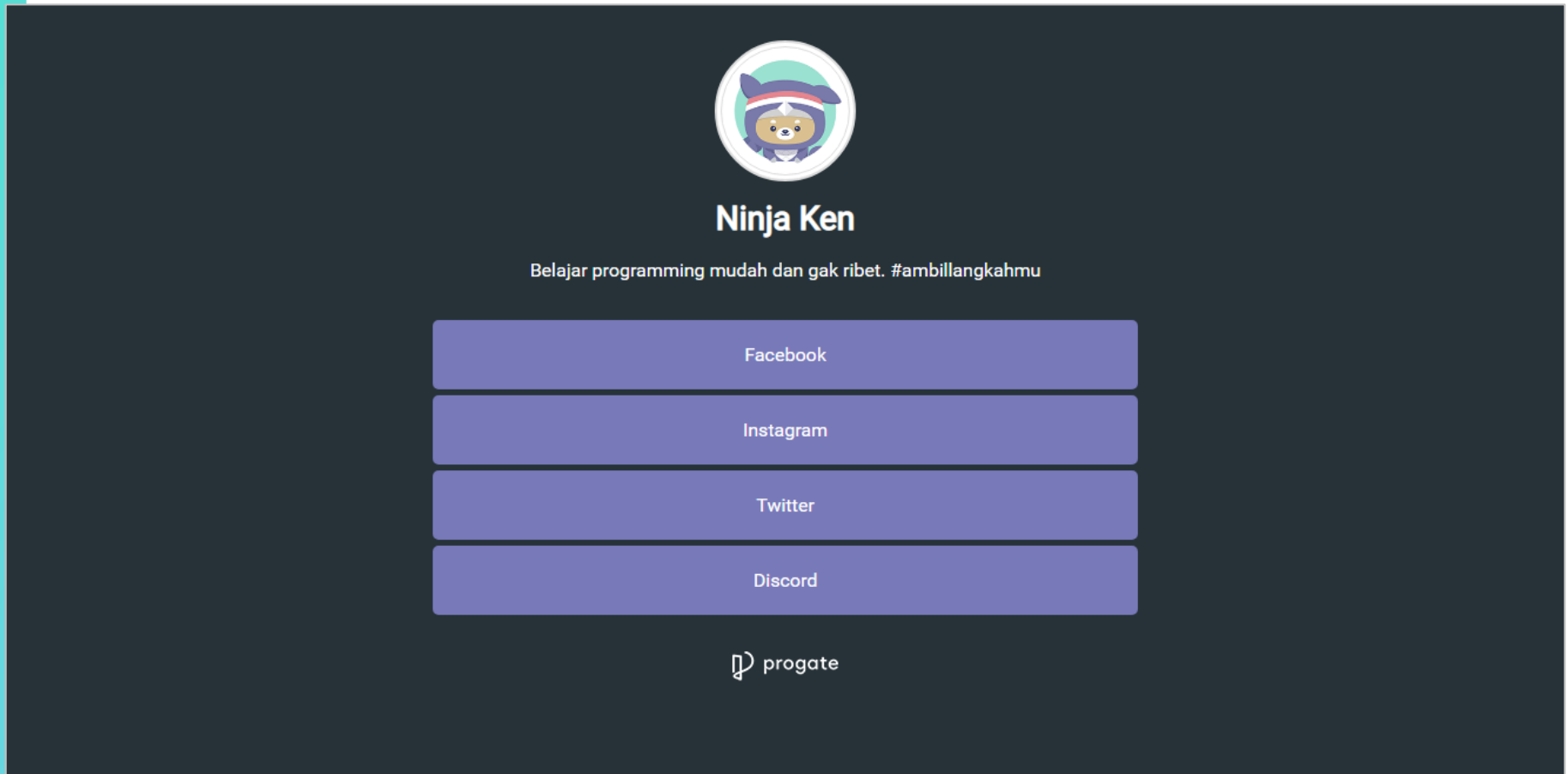
Belajar di Progate sangat mudah dan menyenangkan!

[Kembali](#)

[Oke, lanjutkan!](#)

Silakan Anda buka index.html melalui web browser yang sudah diset ke dark mode. Apakah tampilan halaman sudah ikut berubah seperti yang terlihat pada gambar di atas?

## Checkpoint I : Halaman utama



Silakan Anda buka index.html melalui web browser yang sudah diset ke dark mode. Apakah tampilan halaman sudah ikut berubah seperti yang terlihat pada gambar di atas?

### Langkah 3 : Memperbaiki tampilan

Tunggu sebentar, kenapa pada halaman peringatan muncul foto Ninja Ken?

Jelas kita tidak menginginkan hal ini, dan ini diakibatkan posisi foto tersebut berada “di atas” halaman peringatan. Untuk memastikan bahwa halaman peringatan selalu menutupi bagian halaman lainnya, kita harus memodifikasi dengan menambahkan satu baris CSS pada bagian aside. Silakan Anda selipkan kode z-index berikut ini (baris 20) pada style.css :

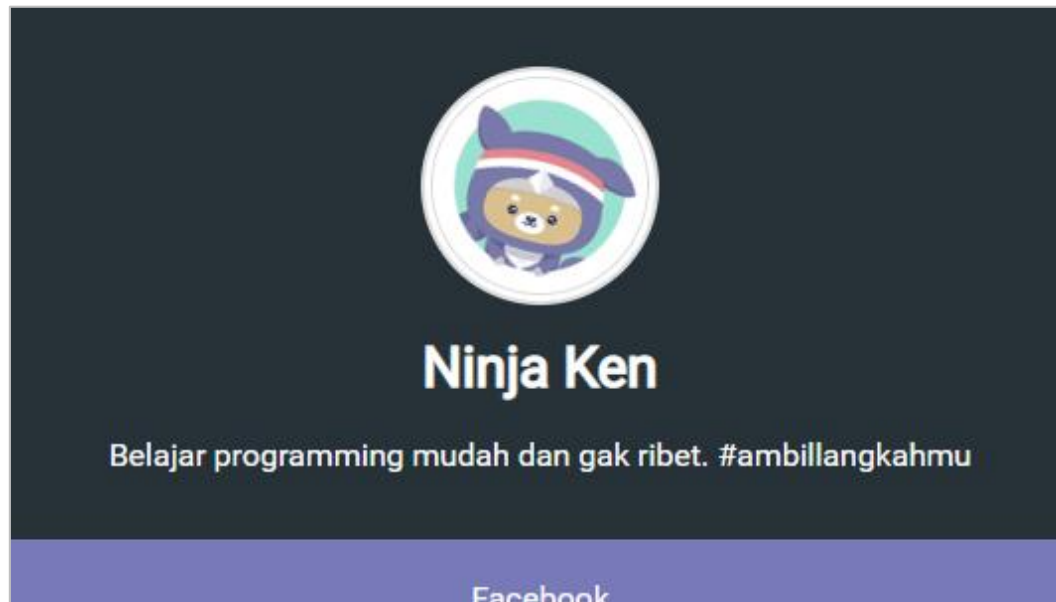
```
17
18  aside {
19      position: fixed;
20      z-index: 100;
21      top: 0;
22      left: 0;
23      display: flex;
24      align-items: center;
25      justify-content: center;
```

Silakan Anda cek kembali apakah halaman sudah tertampil dengan sesuai.

## Langkah 4 : Implementasi CSS transitions

Selanjutnya kita akan menambahkan animasi sederhana pada halaman yang sedang kita rancang. Sebenarnya, pada halaman tersebut sudah menampilkan animasi ketika mouse pointer diarahkan pada tombol, akan terjadi animasi yang membuat latar belakang tombol tersebut berubah warna (menjadi transparan).

Kali ini, kita akan menambahkan animasi yang membuat foto Ninja Ken menjadi Sedikit miring ketika kita mengarahkan mouse pointer ke foto tersebut.



## Langkah 4 : Implementasi CSS transitions

Selanjutnya, buka juga script.js, lalu tambahkan kode berikut :

```
1  function goBack() {  
2    |    history.back();  
3  }  
4  
5  function closeWarning() {  
6    |    document.getElementsByTagName('aside')[0].style.display = 'none';  
7  }
```

Fungsi dari script di atas adalah ketika tombol “Kembali” ditekan, maka web browser akan kembali ke halaman sebelumnya. Sementara ketika tombol “Oke, lanjutkan!” ditekan, maka halaman peringatan akan hilang dan halaman utama akan ditampilkan pada layar.

## Checkpoint II

Silakan Anda buka index.html dan arahkan mouse pointer pada foto Ninja Ken. Apakah foto tersebut bergerak? Apabila ya, kode Anda berfungsi dengan baik.

Terkait CSS transitions, Anda bisa mencari contoh lainnya di search engine dan berkreasi sesuai dengan yang Anda inginkan. Sebagai contoh referensi, Anda bisa mempelajari dari dua website berikut :

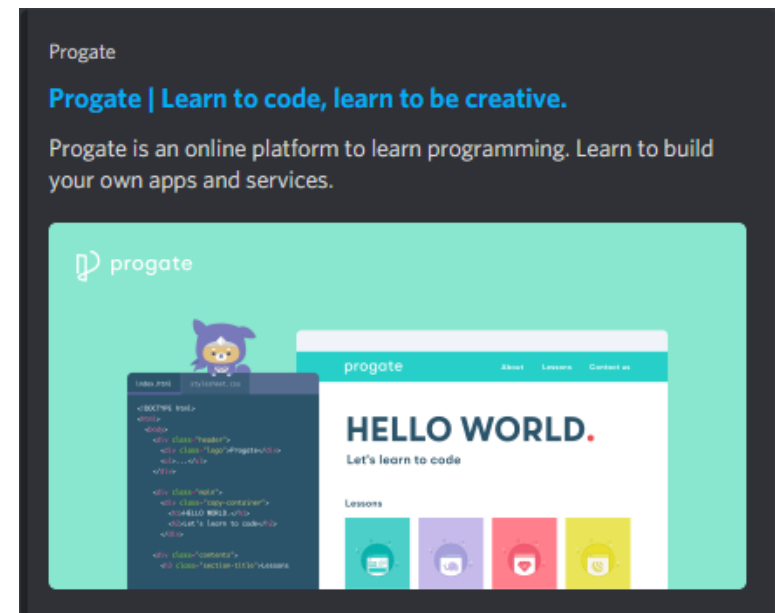
- [https://www.w3schools.com/css/css3\\_transitions.asp](https://www.w3schools.com/css/css3_transitions.asp)
- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Transitions/Using\\_CSS\\_transitions](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions)

## Langkah 5 : Open Graph Metadata

Oke, terakhir kita akan sedikit bermain dengan Open Graph Metadata. Tapi apa lagi itu? Apa fungsinya? Dan kenapa kita harus menambahkannya di dalam website yang sedang kita kembangkan?

Sederhananya, dengan adanya Open Graph Metadata ini, halaman website akan memiliki informasi-informasi yang dibutuhkan atau dapat disesuaikan ketika kita akan memposting halaman tersebut pada media sosial maupun aplikasi chat. Sebagai contoh, ketika kita share link <https://progate.com>, akan muncul seperti :

Informasi-informasi halaman seperti judul halaman, deskripsi, serta thumbnail akan ditampilkan dengan benar, lain halnya ketika kita tidak menyertakan metadata tersebut. Inilah keunggulan yang akan kita dapatkan ketika menerapkan Open Graph Metadata.



## Langkah 6 : Kode HTML

Untuk melakukan ini sangatlah mudah, silakan tambahkan kode di bawah ini pada index.html :

```
10 <link rel="stylesheet" href="res/style.css" >
11
12 <meta property="og:title" content="Ninja Ken">
13 <meta property="og:site_name" content="Social Link">
14 <meta property="og:type" content="website">
15 <meta property="og:description" content="Social Link oleh Ninja Ken">
16 <meta property="og:image" content="res/opengraph.png">
17
18 <!-- Global site tag (gtag.js) - Google Analytics -->
```

Untuk dokumentasi lengkap terkait Open Graph Metadata, silakan Anda akses pada website <https://ogp.me>. Anda dapat menemukan berbagai metadata yang lain yang dapat Anda sesuaikan dengan jenis website yang sedang Anda kembangkan.



## Checkpoint III

Setelah Anda menambahkan kode untuk Open Graph Metadata, secara visual tidak akan terlihat suatu perubahan pada halaman website.

Namun, hasilnya akan terlihat ketika Anda mengupload seluruh file website Anda ke hosting (seperti GitHub Pages), dan Anda bagikan URL dari halaman website tersebut pada media sosial (Facebook, Twitter, dlsb.) maupun aplikasi chat (Telegram, WhatsApp, Discord, dlsb).

## Wrap up

Dengan demikian, kita sudah berhasil membuat sebuah social media landing page yang fungsional, yang telah mendukung dark mode dan diperkaya dengan beberapa penambahan animasi, serta didukung Open Graph Metadata untuk indeks halaman yang lebih baik.

Langkah selanjutnya adalah push semua dokumen yang telah kita update ke GitHub repository, agar teman-teman yang lain juga dapat melihat perubahan-perubahan atau kreasi yang sudah kita lakukan. Untuk Anda yang masih belum mengetahui caranya, bisa dipelajari di pelajaran Progate berikut ini :

<https://progate.com/languages/git>

Sampai jumpa di mini project berikutnya!

