

Pertemuan 7

Membuat Halaman yang SEO-Friendly dengan Next.js

1. Tujuan Pembelajaran

2. Memahami pentingnya SEO (Search Engine Optimization) dalam pengembangan website.
3. Menerapkan teknik SEO dasar pada halaman Next.js.
4. Menambahkan metadata, tag Open Graph, dan favicon untuk meningkatkan SEO.
5. Menggunakan `getStaticProps` untuk mengelola metadata secara dinamis.

2. Alat dan Bahan

1. Komputer dengan sistem operasi Windows, macOS, atau Linux.
2. Node.js dan npm (Node Package Manager) terinstall.
3. Text editor seperti Visual Studio Code.
4. Browser modern (Chrome, Firefox, Edge, dll.).
5. Proyek Next.js yang sudah dibuat (referensi dari lampiran pertama).

3. Konsep Dasar SEO (Search Engine Optimization)

SEO adalah serangkaian teknik untuk meningkatkan visibilitas website di mesin pencari (Google, Bing, dll.). Tujuannya:

- Meningkatkan peringkat website di hasil pencarian organik.
- Meningkatkan traffic pengunjung.
- Memastikan konten mudah diindeks oleh mesin pencari.

Elemen Penting SEO:

1. Metadata:

- **Title Tag:** Judul halaman yang muncul di SERP (Search Engine Results Page).
- **Meta Description:** Deskripsi singkat yang muncul di bawah judul di hasil pencarian.
- **Open Graph (OG) Tags:** Metadata untuk tampilan konten di media sosial (Facebook, Twitter, dll.).
- **Structured Data (Schema Markup):** Data terstruktur untuk membantu mesin pencari memahami konten.

2. Struktur Konten:

- **Heading Tags (H1, H2, H3):** Memberikan hierarki konten.
- **URL yang Deskriptif:** Contoh: /about lebih baik daripada /page?id=1.
- **Konten Berkualitas:** Relevan, original, dan bermanfaat.

3. Optimasi Teknis:

- **Kecepatan Loading:** Menggunakan optimasi gambar, lazy loading, dan code splitting.
- **Mobile-Friendly:** Desain responsif dengan Tailwind CSS.
- **Canonical Tags:** Mencegah duplikasi konten.

4. Next.js dan App Router untuk SEO

Next.js menyediakan fitur bawaan untuk optimasi SEO, terutama dengan **App Router** (diperkenalkan di Next.js 13+).

Perbedaan Pages Router vs. App Router:

Fitur	Pages Router (pages/)	App Router (app/)
Data Fetching	getStaticProps, getServerSideProps	fetch() dengan caching
Metadata	Manual (next/head)	Metadata API (otomatis)
Routing	File-based (contoh: pages/about.js)	Folder-based (app/about/page.js)
SEO	Harus konfigurasi manual	Lebih terintegrasi

Keunggulan App Router untuk SEO:

1. Metadata API:

- Next.js menyediakan metadata object yang secara otomatis menghasilkan <title>, <meta>, dan <link> tags.
- Contoh:

```
export const metadata = {  
  title: "Tentang Saya",  
  description: "Halaman profil pribadi",  
};
```

- **Open Graph** bisa ditambahkan langsung:

```
openGraph: {  
  images: [{ url: "/og-image.jpg" }],
```

```
}
```

2. Generate Metadata Dinamis:

- o Untuk halaman dinamis (misal: blog post), dapat menggunakan generateMetadata:

```
export async function generateMetadata({ params }) {  
  const post = await getPost(params.slug);  
  return { title: post.title };  
}
```

3. Optimasi Gambar:

- o Komponen <Image> Next.js otomatis melakukan:
 - **Lazy Loading**
 - **Format Optimization** (WebP)
 - **Responsive Sizing**

5. Best Practices SEO di Next.js

1. Gunakan Metadata API

- Selalu definisikan metadata di setiap halaman.
- Untuk halaman dinamis, gunakan generateMetadata.

2. Optimasi Gambar

- Gunakan komponen <Image> Next.js:

```
import Image from 'next/image';  
<Image src="/profile.jpg" alt="Profil" width={500} height={300} />
```

3. Gunakan Semantic HTML

- Gunakan tag <header>, <main>, <footer>, <article>, dll.
- Contoh:

```
<header>  
  <h1>Judul Halaman</h1>  
</header>  
<main>  
  <article>Konten utama</article>
```

```
</main>  
Run HTML
```

4. Tambahkan Structured Data (Schema.org)

- Berguna untuk **rich snippets** di hasil pencarian.
- Contoh schema untuk "Person":

```
<script type="application/ld+json">  
{  
  "@context": "https://schema.org",  
  "@type": "Person",  
  "name": "Nama Anda",  
  "url": "https://example.com"  
}  
</script>
```

6. Langkah-langkah Praktikum

6.1. Memahami Pentingnya SEO

SEO (Search Engine Optimization) adalah praktik mengoptimalkan website agar mudah ditemukan oleh mesin pencari seperti Google. Dengan SEO yang baik, website Anda akan lebih mudah muncul di hasil pencarian, sehingga meningkatkan traffic pengunjung.

Beberapa elemen penting dalam SEO:

- **Title Tag:** Judul halaman yang muncul di hasil pencarian.
- **Meta Description:** Deskripsi singkat tentang halaman.
- **Heading Tags (H1, H2, dll.):** Struktur konten yang membantu mesin pencari memahami isi halaman.
- **URL Structure:** URL yang deskriptif dan mudah dibaca.
- **Open Graph Tags:** Metadata untuk media sosial seperti Facebook dan Twitter.
- **Favicon:** Ikon kecil yang muncul di tab browser.

6.2. Menambahkan Favicon

Favicon adalah ikon kecil yang muncul di tab browser dan bookmark. Favicon membantu meningkatkan identitas brand website Anda.

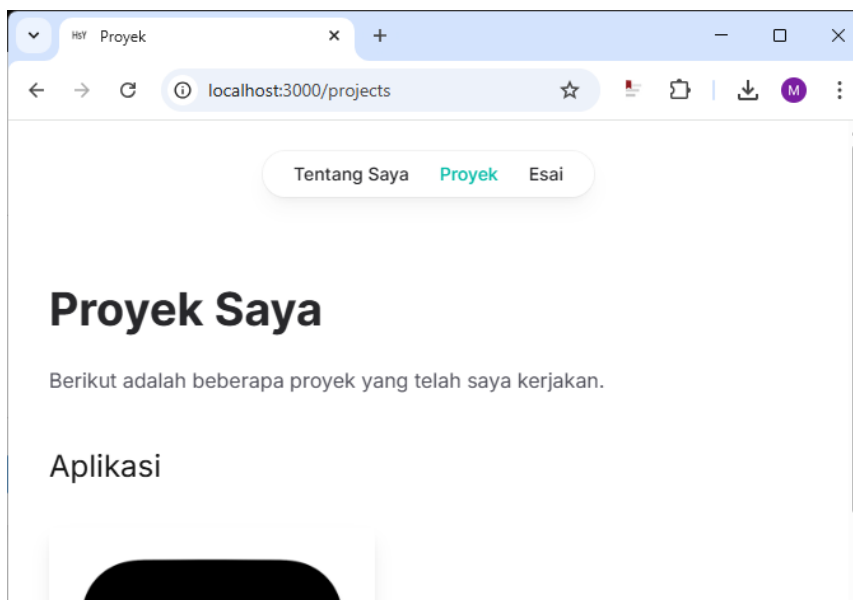
Langkah-langkah:

1. Siapkan file favicon (format .ico) dan letakkan di folder **public/**.

2. Hapus faveicon.ico yang ada di folder **src/app**
3. Update file **src/components/layout.tsx** untuk menambahkan favicon:

```
9   export default function Layout({ children }: LayoutProps) {
10     return (
11       <>
12         <Head>
13           <link rel="icon" href="/favicon.ico" />
14         </Head>
15         <Navbar />
16         <main>{children}</main>
17         <Footer />
18       </>
19     );
20   }
```

3. Simpan dan jalankan proyek. Favicon akan muncul di tab browser.



6.3. Menambahkan Metadata untuk SEO

Metadata seperti title dan description sangat penting untuk SEO. Metadata ini akan muncul di hasil pencarian dan membantu mesin pencari memahami konten halaman.

Langkah-langkah:

1. Buat file **src/utlis/metadata.ts** untuk menyimpan metadata default:

src > utils > TS metadata.ts > defaultMetadata > openGraph > images

```
1 // utils/metadata.ts
2 export const defaultMetadata = {
3   title: "Website Pribadi",
4   description: "Website pribadi untuk menampilkan proyek dan esai.",
5   openGraph: {
6     images: "https://example.com/default-image.jpg",
7     url: "https://example.com",
8   },
9 };
```

2. Update file **src/components/layout.tsx** untuk menggunakan metadata default jika tidak ada metadata yang di-pass:

src > components > Layout.tsx > Layout

```
1 import Head from 'next/head';
2 import Navbar from './Navbar';
3 import Footer from './Footer';
4 import { defaultMetadata } from '../utils/metadata';
5
6 type LayoutProps = {
7   children: React.ReactNode;
8 };
9
10 export default function Layout({ children }: any) {
11   const metadata = children.props.metadata || defaultMetadata;
12
13   return (
14     <>
15       <Head>
16         <link rel="icon" href="/favicon.ico" />
17         <title>{metadata.title}</title>
18         <meta name="description" content={metadata.description} />
19         { /* Open Graph Tags */ }
20         <meta property="og:title" content={metadata.title} />
21         <meta property="og:description" content={metadata.description} />
22         <meta property="og:image" content={metadata.openGraph.image} />
23         <meta property="og:url" content={metadata.openGraph.url} />
24       </Head>
25       <Navbar />
26       <main>{children}</main>
27       <Footer />
28     </>
29   );
30 }
```

3. Update Setiap Halaman untuk dapat mem-pass definisi metadata

Contoh Halaman "Tentang Saya"

```

src > app > page.tsx > metadata > openGraph
1  import type { Metadata } from 'next';
2  import { defaultMetadata } from '../utils/metadata';
3
4  export const metadata: Metadata = {
5    title: 'Tentang Saya',
6    description: 'Halaman tentang saya.',
7    openGraph: {
8      images: "https://example.com/default-image.jpg",
9      url: "https://example.com",
10    },
11  };
12
13  export default function Home({ metadata }: any) {
14    return (
15      <div className="mt-16 px-8">
16        <header>
17          <h1 className="font-bold text-4xl text-zinc-800">Tentang Saya</h1>
18        </header>
19      </div>
20    );
21  }

```

4. Lanjutkan untuk halaman **Proyek** dan **Esai**

6.4. Open Graph Tags untuk Media Sosial

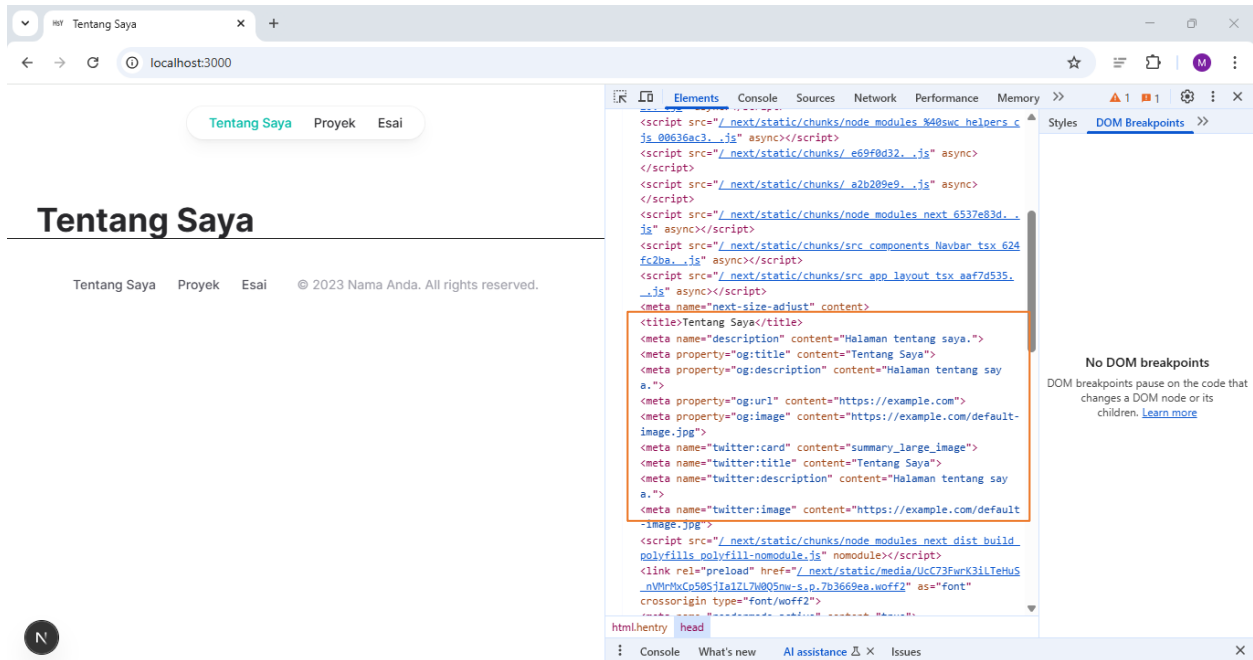
Open Graph Tags digunakan untuk mengontrol bagaimana konten Anda ditampilkan saat dibagikan di media sosial seperti Facebook dan Twitter.

Langkah-langkah:

1. Pastikan Anda telah menambahkan Open Graph Tags di layout.tsx seperti yang telah dijelaskan sebelumnya.
2. Update metadata di setiap halaman untuk menyertakan Open Graph Tags.

6.5. Menguji SEO

Setelah menambahkan metadata, favicon, dan Open Graph Tags, Anda dapat menguji SEO dengan Membuka halaman di browser dan memeriksa elemen HTML menggunakan Developer Tools (Ctrl + Shift + I atau F12).



7. Tugas

Lakukan pengujian SEO dengan

1. Gunakan tools seperti Google Search Console untuk memeriksa performa SEO dari suatu website.

8. Kesimpulan

Dalam praktikum ini, Anda telah mempelajari cara membuat halaman Next.js yang SEO-friendly dengan menambahkan metadata, favicon, dan Open Graph Tags. Dengan teknik ini, website Anda akan lebih mudah ditemukan oleh mesin pencari dan menarik lebih banyak pengunjung.