

Membuat Hyperlink: Tag `<a>`

Hyperlink adalah tulang punggung web, menghubungkan pengguna ke sumber daya di seluruh internet dan dalam satu website. Tag `<a>` (tag anchor) adalah elemen HTML yang membuat hyperlink ini. Memahami cara menggunakan tag `<a>` dan atributnya adalah fundamental untuk pengembangan web. Pelajaran ini akan memberikan panduan komprehensif untuk membuat hyperlink, mencakup konsep teoritis dan implementasi praktis.

Memahami Tag `<a>`

Tag `<a>` mendefinisikan hyperlink, yang merupakan tautan yang dapat diklik yang membawa Anda ke halaman atau lokasi lain. Struktur dasar dari tag `<a>` adalah sebagai berikut:

html

```
<a href="URL">Teks link</a>
```

- `<a>`: Ini adalah tag pembuka untuk elemen anchor.
- `href`: Atribut ini menentukan alamat tujuan atau URL (Uniform Resource Locator) yang dituju oleh link. Ini adalah atribut terpenting dari tag `<a>`.
- `"URL"`: Ini adalah alamat web aktual atau jalur ke sumber daya yang ditautkan.
- `"Teks link"`: Ini adalah teks yang terlihat yang diklik pengguna. Harus deskriptif dan memberi pengguna gambaran tentang kemana link akan membawa mereka.
- ``: Ini adalah tag penutup untuk elemen anchor.

Mari kita uraikan setiap komponen dengan contoh:

Contoh 1: Link Dasar

html

```
<a href="https://www.example.com">Kunjungi Website Example</a>
```

Dalam contoh ini, teks "Kunjungi Website Example" akan ditampilkan sebagai link yang dapat diklik. Mengklik akan membawa pengguna ke <https://www.example.com>.

Contoh 2: Link dengan atribut title

html

```
<a href="https://www.example.com" title="Pelajari lebih lanjut tentang example">Kunjungi Website Example</a>
```

Di sini, kita telah menambahkan atribut `title`. Teks dalam atribut `title` akan muncul sebagai tooltip ketika pengguna mengarahkan mouse mereka ke link. Ini dapat memberikan konteks tambahan untuk pengguna.

URL Absolut vs. Relatif

Atribut `href` menerima dua jenis URL: absolut dan relatif. Memahami perbedaannya sangat penting untuk pembuatan link yang efektif.

URL Absolut

URL absolut adalah alamat web lengkap, termasuk protokol (misalnya, `https://`), nama domain (misalnya, www.example.com), dan jalur ke sumber daya spesifik (misalnya, `/index.html`).

Contoh:

html

```
<a href="https://www.example.com/about.html">Tentang Kami</a>
...
```

Link ini menggunakan URL absolut, jadi akan selalu menunjuk ke halaman `about.html` di domain `www.example.com`, terlepas dari di mana link ditempatkan dalam website Anda. URL absolut biasanya digunakan untuk menautkan ke website eksternal.

URL Relatif

URL relatif menentukan jalur ke sumber daya relatif terhadap lokasi halaman saat ini. Mereka tidak termasuk protokol atau nama domain. URL relatif terutama digunakan untuk menautkan ke halaman atau sumber daya lain dalam website yang sama.

****Contoh:****

Asumsikan Anda memiliki struktur file berikut:

...

mywebsite/

└─ index.html

└─ about/

 └─ about.html

Di index.html, Anda dapat menautkan ke about.html menggunakan URL relatif:

html

```
<a href="about/about.html">Tentang Kami</a>
```

Berikut adalah rincian berbagai jenis URL relatif:

Direktori yang sama: Jika halaman yang ditautkan berada di direktori yang sama dengan halaman saat ini, Anda cukup menggunakan nama file:

html

```
<a href="contact.html">Kontak</a>
```

Subdirektori: Jika halaman yang ditautkan berada di subdirektori, tentukan jalur ke subdirektori diikuti oleh nama file:

html

```
<a href="blog/article.html">Baca Artikel</a>
```

Direktori induk: Untuk naik satu level direktori, gunakan `../`:

html

```
<a href="../index.html">Kembali ke Beranda</a>
```

...

Jika Anda perlu naik beberapa level, Anda dapat merangkai `../` bersama: `../../`.

****Keuntungan URL Relatif:****

- **Portabilitas:** Jika Anda memindahkan seluruh website ke domain baru, URL relatif akan tetap berfungsi dengan benar karena mereka relatif terhadap struktur website, bukan domain tertentu.
- **Kemudahan Pemeliharaan:** Mereka membuat kode Anda lebih mudah dipelihara. Anda tidak perlu memperbarui semua link jika nama domain Anda berubah.
- **Development/Staging:** URL relatif ideal ketika bekerja pada website secara lokal atau di server staging sebelum menerapkannya ke domain live.

Kapan Menggunakan Yang Mana:

- Gunakan URL absolut saat menautkan ke website eksternal (situs yang tidak Anda kontrol).
- Gunakan URL relatif saat menautkan ke halaman, gambar, atau sumber daya lain dalam website Anda sendiri.

Menautkan ke Halaman Internal dan Eksternal

Seperti yang ditunjukkan di bagian sebelumnya, tag `` dapat digunakan untuk menautkan ke halaman internal (dalam website Anda) dan eksternal (ke website lain). Perbedaan utama terletak pada jenis URL yang digunakan dalam atribut `href`.

Menautkan ke Halaman Internal

Menautkan ke halaman internal melibatkan penggunaan URL relatif untuk menavigasi antara bagian atau halaman berbeda dalam website Anda. Ini sangat penting untuk menciptakan pengalaman pengguna yang mulus dan website yang terstruktur dengan baik.

Contoh:

Pertimbangkan website dengan struktur berikut:

...

website/

|— index.html

|— products.html

|— services.html

|— contact.html

Untuk menautkan dari index.html ke products.html, Anda akan menggunakan kode berikut:

html

```
<a href="products.html">Lihat Produk</a>
```

Demikian pula, untuk menautkan dari products.html kembali ke index.html, Anda akan menggunakan:

html

```
<a href="index.html">Kembali ke Beranda</a>
```

Untuk menautkan ke halaman dalam subdirektori, ikuti struktur jalur relatif.

Menautkan ke Halaman Eksternal

Menautkan ke halaman eksternal memerlukan penggunaan URL absolut yang menunjuk ke sumber daya di website lain. Ini memungkinkan Anda mengarahkan pengguna ke informasi, sumber daya, atau layanan yang relevan yang dihosting di tempat lain di internet.

Contoh:

Untuk menautkan ke website Mozilla Developer Network (MDN), Anda akan menggunakan kode berikut:

html

```
<a href="https://developer.mozilla.org/">Kunjungi MDN</a>
```

Praktik Terbaik untuk Link Eksternal:

target="_blank": Pertimbangkan untuk menggunakan atribut `target="_blank"` untuk membuka link eksternal di tab atau jendela baru. Ini membuat pengguna tetap di website Anda sambil tetap memungkinkan mereka mengakses sumber daya eksternal.

html

```
<a href="https://developer.mozilla.org/" target="_blank" rel="noopener  
noreferrer">Kunjungi MDN (dibuka di tab baru)</a>
```

- **Penjelasan `target="_blank"`:** Atribut `target="_blank"` memberi tahu browser untuk membuka link di tab atau jendela baru.
- **Penjelasan `rel="noopener noreferrer"`:** Atribut ini sangat penting untuk alasan keamanan saat menggunakan `target="_blank"`. Ini mencegah halaman yang ditautkan dari kemungkinan mendapatkan kontrol atas halaman asli Anda. `noopener` mencegah halaman baru mengakses properti `window.opener`, dan `noreferrer` mencegah halaman baru mengetahui halaman mana yang menautkan ke sana.

Teks Link Deskriptif: Gunakan teks link yang jelas dan deskriptif untuk memberi tahu pengguna kemana link akan membawa mereka. Hindari frasa yang tidak jelas seperti "klik di sini."

Atribut Tag `<a>` Lainnya

Selain `href` dan `target`, tag `<a>` mendukung atribut berguna lainnya.

Atribut `title`

Seperti yang terlihat dalam contoh sebelumnya, atribut `title` memberikan informasi tambahan tentang link, ditampilkan sebagai tooltip saat hover. Ini meningkatkan aksesibilitas dan pengalaman pengguna.

html

```
<a href="https://www.example.com" title="Pelajari lebih lanjut tentang Example Corp">Kunjungi Example Corp</a>
```

Atribut `rel`

Atribut `rel` menentukan hubungan antara dokumen saat ini dan dokumen yang ditautkan. Ini terutama digunakan untuk tujuan SEO dan aksesibilitas. Kita telah melihat `rel="noopener noreferrer"` dalam konteks link eksternal, tetapi berikut adalah beberapa nilai `rel` berguna lainnya:

`nofollow`: Memberi tahu mesin pencari untuk tidak mengikuti link. Ini sering digunakan untuk link sponsor atau link ke konten yang tidak dipercaya.

html

```
<a href="https://www.example.com/sponsored" rel="nofollow">Link Sponsor</a>
```

noopener: Seperti yang dijelaskan sebelumnya, meningkatkan keamanan saat menggunakan `target="_blank"`.

noreferrer: Seperti yang dijelaskan sebelumnya, menyembunyikan informasi halaman rujukan dari website yang ditautkan.

alternate: Menentukan versi alternatif dari dokumen (misalnya, terjemahan atau jenis media berbeda).

html

```
<a href="https://www.example.com/fr/" rel="alternate" hreflang="fr">Versi Prancis</a>
```

Menautkan ke Alamat Email

Tag `<a>` juga dapat digunakan untuk membuat link "mailto" yang membuka klien email default pengguna dengan alamat penerima yang sudah diisi sebelumnya.

html

```
<a href="mailto:info@example.com">Email Kami</a>
```

Mengklik link ini akan membuka klien email pengguna dan membuat email baru yang ditujukan ke `info@example.com`.

Anda juga dapat mengisi subjek dan body email sebelumnya:

html

```
<a href="mailto:info@example.com?subject=Pertanyaan%20Website&body=Halo,%20Saya%20memiliki%20pertanyaan%20tentang...">Email Kami</a>
```

- `?subject=`: Menentukan subjek email.
- `&body=`: Menentukan body email.
- `%20`: Mewakili karakter spasi dalam URL.

Menautkan ke Bagian Spesifik dalam Halaman (Anchor)

Anda dapat menautkan ke bagian spesifik dalam halaman yang sama menggunakan anchor. Ini melibatkan dua langkah:

1. **Definisikan target:** Gunakan atribut `id` pada elemen yang ingin Anda tautkan.

html

```
<h2 id="section2">Bagian 2</h2>
```

2. **Buat link:** Gunakan atribut `href` dengan simbol `#` diikuti oleh id elemen target.

html

```
<a href="#section2">Pergi ke Bagian 2</a>
```

Ketika pengguna mengklik link, browser akan scroll ke elemen dengan id yang sesuai.

Contoh:

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Anchor</title>
</head>
<body>
  <h1>Halaman Saya</h1>
  <a href="#section1">Pergi ke Bagian 1</a><br>
  <a href="#section2">Pergi ke Bagian 2</a><br>
  <a href="#section3">Pergi ke Bagian 3</a>

  <h2 id="section1">Bagian 1</h2>
  <p>Ini adalah konten bagian 1.</p>

  <h2 id="section2">Bagian 2</h2>
  <p>Ini adalah konten bagian 2.</p>
```



```
<h2 id="section3">Bagian 3</h2>
<p>Ini adalah konten bagian 3.</p>
</body>
</html>
```

Dalam contoh ini, mengklik "Pergi ke Bagian 1" akan membawa Anda ke heading "Bagian 1". Ini sangat berguna untuk halaman panjang dengan beberapa bagian.

Latihan Praktis

1. **Buat menu navigasi:** Buat halaman HTML sederhana dengan menu navigasi yang menautkan ke tiga bagian berbeda dalam halaman yang sama menggunakan link anchor. Setiap bagian harus memiliki heading dan beberapa teks paragraf.
2. **Link eksternal dengan target="_blank":** Buat link ke website berita populer (misalnya, CNN, BBC) yang terbuka di tab atau jendela baru. Sertakan atribut `rel="noopener noreferrer"` untuk keamanan. Tambahkan atribut `title` yang menjelaskan kemana link pergi.
3. **Link email:** Buat link email yang mengisi subjek dengan "Pertanyaan Website" dan menyertakan pesan pengantar singkat di body.
4. **URL Relatif dan Absolut:** Buat dua halaman HTML, `index.html` dan `about.html`, dalam direktori yang sama. Di `index.html`, buat link relatif ke `about.html`. Juga di `index.html`, buat link absolut ke `about.html` menggunakan domain placeholder seperti www.example.com/about.html. Uji kedua link untuk melihat bagaimana mereka berperilaku.

Langkah Selanjutnya

Sekarang setelah Anda mempelajari cara membuat hyperlink menggunakan tag `<a>`, Anda siap untuk melanjutkan ke pelajaran berikutnya, yang mencakup penambahan gambar ke halaman web Anda menggunakan tag ``. Memahami cara menggabungkan link dan gambar akan memungkinkan Anda membuat pengalaman web yang lebih menarik dan interaktif. Kita juga akan mengeksplorasi atribut gambar seperti `src`, `alt`, `width`, dan `height` di pelajaran berikutnya.

Memahami URL Absolut dan Relatif

Memahami URL adalah fundamental untuk pengembangan web. URL adalah tulang punggung internet, memungkinkan kita untuk menavigasi antara sumber daya yang berbeda, baik itu di website yang sama atau tersebar di seluruh dunia. Pelajaran ini mendalami perbedaan penting antara URL absolut dan relatif, memberikan Anda pengetahuan untuk menautkan secara efektif dalam dokumen HTML Anda dan ke website eksternal. Memahami konsep ini tidak hanya akan membuat link Anda berfungsi dengan benar tetapi juga berkontribusi pada organisasi dan kemudahan pemeliharaan website yang lebih baik.

URL Absolut: Jalur Lengkap

URL absolut, seperti namanya, memberikan alamat lengkap dari sumber daya di web. Ini seperti memberikan alamat lengkap sebuah bangunan, termasuk kota, provinsi, dan kode pos. Ini berarti termasuk protokol (misalnya, `http`, `https`), nama domain (misalnya, www.example.com), dan jalur lengkap ke sumber daya (misalnya, `/images/logo.png` atau `/about.html`).

Anatomi URL Absolut

Mari kita uraikan contoh: `https://www.example.com/products/index.html`.

- `https://` adalah protokol, menunjukkan bagaimana browser harus mengakses sumber daya. `https` menandakan koneksi yang aman.
- `www.example.com` adalah nama domain, mengidentifikasi server web yang menghosting sumber daya.
- `/products/index.html` adalah jalur ke sumber daya spesifik di server.

Kapan Menggunakan URL Absolut

URL absolut biasanya digunakan ketika menautkan ke sumber daya di website atau domain berbeda. Jika Anda menautkan ke halaman di Wikipedia atau gambar yang dihosting di server lain, Anda perlu menggunakan URL absolut. Mereka juga berguna ketika lokasi file HTML Anda dalam struktur situs Anda sendiri mungkin berubah, dan Anda ingin memastikan link tetap valid.

Contoh URL Absolut dalam HTML

html

```
<a href="https://www.wikipedia.org/wiki/HTML">Pelajari lebih lanjut tentang HTML di
Wikipedia</a>

` ``
```

Dalam contoh pertama, kita menautkan ke halaman HTML di Wikipedia menggunakan URL absolut. Dalam yang kedua, kita menampilkan gambar yang dihosting di example.com, juga menggunakan URL absolut.

URL Relatif: Singkatan

URL relatif menentukan lokasi sumber daya relatif terhadap lokasi dokumen HTML saat ini. Ini seperti memberikan petunjuk arah ke kedai kopi terdekat dari lokasi Anda saat ini. Alih-alih memberikan alamat lengkap, Anda mungkin berkata, "Dua blok ke bawah jalan." Ini membuat URL relatif lebih pendek dan lebih nyaman untuk menautkan ke sumber daya dalam website yang sama.

Jenis URL Relatif

Ada beberapa jenis URL relatif, tergantung pada lokasi sumber daya:

****Direktori yang Sama:**** Jika sumber daya yang ditautkan berada di direktori yang sama dengan file HTML saat ini, Anda hanya perlu menentukan nama sumber daya. Misalnya, jika file HTML Anda berada di direktori `/about/` dan Anda ingin menautkan ke `contact.html` yang juga berada di `/about/`, URL relatifnya cukup `contact.html`.

****Subdirektori:**** Jika sumber daya berada di subdirektori dari direktori file HTML saat ini, Anda menentukan nama subdirektori diikuti dengan garis miring dan nama sumber daya. Misalnya, jika file HTML Anda berada di `/products/` dan Anda ingin menautkan ke gambar `image.jpg` di dalam `/products/images/`, URL relatifnya adalah `images/image.jpg`.

****Direktori Induk:**** Jika sumber daya berada di direktori induk dari direktori file HTML saat ini, Anda menggunakan `../` untuk naik satu level dalam struktur direktori. Misalnya,

jika file HTML Anda berada di `/products/electronics/` dan Anda ingin menautkan ke `products.html` di direktori `/products/`, URL relatifnya adalah `../products.html`. Anda dapat menggunakan beberapa segmen `../` untuk naik beberapa level. Misalnya, `../../` naik dua level.

Kapan Menggunakan URL Relatif

URL relatif umumnya lebih disukai untuk menautkan ke sumber daya dalam website Anda sendiri. Mereka menawarkan beberapa keuntungan:

****Portabilitas:**** Jika Anda memindahkan seluruh website ke domain baru, URL relatif akan tetap berfungsi dengan benar, karena mereka relatif terhadap struktur file, bukan domain.

****Kemudahan Pemeliharaan:**** Jika Anda mengubah nama domain, Anda tidak perlu memperbarui semua link internal Anda.

****Efisiensi Pengembangan:**** URL relatif lebih pendek dan lebih mudah diketik, terutama ketika berurusan dengan direktori yang sangat bersarang.

Contoh URL Relatif dalam HTML

Katakanlah Anda memiliki struktur direktori berikut:

...

```
website/
├── index.html
├── about/
│   └── about.html
├── css/
│   └── styles.css
└── images/
    └── logo.png
```

Berikut adalah beberapa contoh URL relatif yang mungkin Anda gunakan di `index.html`:

html

```
<a href="about/about.html">Tentang Kami</a> <!-- Menautkan ke about.html di subdirektori
"about" -->
<link rel="stylesheet" href="css/styles.css"> <!-- Menautkan ke styles.css di subdirektori
"css" -->
 <!-- Menautkan ke logo.png di subdirektori
"images" -->
```

Dan berikut adalah contoh bagaimana Anda mungkin menggunakan URL relatif di `about/about.html` untuk menautkan kembali ke halaman beranda:

html

```
<a href="../index.html">Kembali ke Beranda</a> <!-- Menautkan kembali ke index.html di
direktori induk -->
```

URL Absolut vs. Relatif: Perbandingan Berdampingan

Fitur	URL Absolut	URL Relatif
Definisi	Alamat lengkap sumber daya di web	Alamat relatif terhadap dokumen HTML saat ini
Contoh	<code>https://www.example.com/images/logo.png</code>	<code>images/logo.png</code> atau <code>../index.html</code>
Penggunaan	Menautkan ke website eksternal, sumber daya di CDN	Menautkan ke sumber daya internal dalam situs yang sama
Portabilitas	Kurang portabel	Lebih portabel
Kemudahan Pemeliharaan	Memerlukan pembaruan jika nama domain berubah	Tidak perlu pembaruan jika nama domain berubah
Panjang	Lebih panjang	Lebih pendek

Praktik Terbaik

Gunakan URL relatif untuk link internal: Ini membuat website Anda lebih portabel dan lebih mudah dipelihara.

Gunakan URL absolut untuk link eksternal: Ini memastikan bahwa Anda selalu menautkan ke sumber daya yang benar, bahkan jika struktur website eksternal berubah.

Konsisten: Pilih gaya (relatif atau absolut) untuk link internal Anda dan pertahankan di seluruh website Anda. Ini akan membuat kode Anda lebih mudah dibaca dan dipelihara.

Uji link Anda: Selalu uji link Anda untuk memastikan mereka berfungsi dengan benar. Link yang rusak dapat membuat frustrasi pengguna dan dapat merusak reputasi website Anda.

Latihan

1. **Buat website sederhana dengan setidaknya tiga halaman HTML:** `index.html`, `about.html`, dan `contact.html`. Tautkan halaman-halaman ini bersama-sama menggunakan URL relatif.
2. **Tambahkan gambar ke website Anda:** Tempatkan gambar dalam folder images dan tautkan ke sana dari setiap halaman HTML Anda menggunakan URL relatif.
3. **Tautkan ke website eksternal dari halaman `index.html` Anda:** Gunakan URL absolut untuk menautkan ke website yang relevan, seperti Mozilla Developer Network (MDN) atau Wikipedia.
4. **Latihan menggunakan link direktori induk:** Buat subdirektori dalam website Anda (misalnya, blog). Tempatkan file HTML di subdirektori ini dan tautkan kembali ke halaman `index.html` Anda menggunakan `../index.html`.
5. **Skenario:** Bayangkan perusahaan Anda, "AwesomeTech," telah mendesain ulang websitenya dan mengubah domainnya dari `awesometech.com` menjadi `awesome-tech.net`. Perubahan apa yang perlu Anda lakukan pada file HTML Anda jika Anda telah menggunakan URL absolut untuk semua link internal? Bagaimana jika Anda telah menggunakan URL relatif?

Dengan menyelesaikan latihan ini, Anda akan mendapatkan pemahaman yang solid tentang cara menggunakan URL absolut dan relatif secara efektif dalam proyek pengembangan web Anda. Anda juga akan siap untuk menangani skenario linking yang lebih kompleks di masa depan.

Ringkasan

Pelajaran ini telah membahas konsep fundamental dari URL absolut dan relatif. Memahami kapan dan bagaimana menggunakan setiap jenis URL sangat penting untuk membangun website yang terstruktur dengan baik dan mudah dipelihara. Di pelajaran berikutnya, kita akan mengeksplorasi cara membuat hyperlink menggunakan tag `<a>`, membangun pengetahuan Anda tentang URL untuk membuat pengalaman web yang dapat dinavigasi. Kita akan membahas penautan ke bagian spesifik dalam halaman (anchor) dan teknik linking lanjutan lainnya.

Menautkan ke Halaman Internal dan Eksternal

Membuat hyperlink adalah salah satu aspek paling fundamental dari HTML, memungkinkan Anda menghubungkan sumber daya yang berbeda dan membuat pengalaman web interaktif. Pelajaran ini akan mengeksplorasi cara menggunakan tag `<a>` untuk menautkan ke halaman internal dalam website Anda dan halaman eksternal di website lain, membangun fondasi yang solid untuk membuat konten web yang dapat dinavigasi dan saling terhubung. Memahami perbedaan antara URL absolut dan relatif sangat penting untuk menjaga integritas link Anda saat website Anda berkembang.

Memahami URL Absolut dan Relatif

Sebelum mendalami pembuatan link, penting untuk memahami perbedaan antara URL absolut dan relatif. URL (Uniform Resource Locator) adalah alamat untuk sumber daya di internet.

URL Absolut

URL absolut berisi semua informasi yang diperlukan untuk menemukan sumber daya, termasuk protokol (misalnya, `https://`), nama domain (misalnya, www.example.com), dan jalur lengkap ke sumber daya (misalnya, `/images/logo.png`).

Contoh: `https://www.example.com/about-us.html`

Dalam contoh ini:

- `https://` adalah protokol (HTTP aman)

- www.example.com adalah nama domain
- </about-us.html> adalah jalur ke file HTML spesifik

Kapan Menggunakan:

- Menautkan ke website eksternal. Karena Anda tidak mengontrol struktur direktori situs eksternal, Anda harus menggunakan URL absolut untuk memastikan link berfungsi dengan benar.
- Ketika Anda memerlukan link yang dijamin dan tidak ambigu, terlepas dari lokasi halaman saat ini dalam situs Anda sendiri.
- Ketika menautkan ke sumber daya yang dihosting di domain atau subdomain berbeda.

Skenario Hipotetis:

Bayangkan Anda membuat website untuk perpustakaan lokal. Anda ingin menautkan ke website resmi Library of Congress. Karena website Library of Congress sepenuhnya terpisah dari website perpustakaan Anda, Anda akan menggunakan URL absolut: <https://www.loc.gov/>.

URL Relatif

URL relatif menentukan lokasi sumber daya relatif terhadap lokasi dokumen HTML saat ini. Ini tidak termasuk protokol atau nama domain.

Contoh: [images/my-image.jpg](#)

URL ini menunjukkan bahwa file gambar [my-image.jpg](#) terletak di direktori [images](#), yang diasumsikan berada di direktori yang sama dengan file HTML saat ini atau subdirektornya.

Kapan Menggunakan:

- Menautkan ke halaman lain dalam website Anda sendiri. URL relatif membuat website Anda lebih portabel. Jika Anda memindahkan seluruh website ke domain baru, link internal akan tetap berfungsi.
- Menautkan ke gambar, stylesheet, dan aset lain dalam website Anda.
- Ketika sumber daya target terletak di domain yang sama dengan halaman saat ini.

Skenario Hipotetis:

Pertimbangkan website dengan file `index.html` dan file `about.html`, keduanya terletak di direktori root. Dari `index.html`, Anda dapat menautkan ke `about.html` menggunakan URL relatif `about.html`.

URL Absolut vs. Relatif: Perbandingan Cepat

Fitur	URL Absolut	URL Relatif
Protokol	Termasuk (misalnya, <code>https://</code>)	Tidak termasuk
Nama Domain	Termasuk (misalnya, <code>www.example.com</code>)	Tidak termasuk
Jalur	Jalur lengkap dari root	Jalur relatif terhadap dokumen saat ini
Kasus Penggunaan	Link eksternal, link yang dijamin	Link internal, portabilitas

Menautkan ke Halaman Eksternal

Untuk membuat link ke halaman eksternal, Anda akan menggunakan tag `<a>` (anchor) bersama dengan atribut `href`, mengatur atribut `href` ke URL absolut dari halaman eksternal.

Contoh:

html

```
<a href="https://www.example.com">Kunjungi Example.com</a>
```

Dalam contoh ini:

- `<a>` adalah tag anchor, yang membuat hyperlink.
- `href="https://www.example.com"` menentukan URL tujuan.
- `Kunjungi Example.com` adalah teks yang akan ditampilkan sebagai link.

Ketika pengguna mengklik "Kunjungi Example.com," browser mereka akan menavigasi ke website `https://www.example.com`.

Membuka Link di Tab/Jendela Baru

Terkadang, Anda mungkin ingin halaman yang ditautkan terbuka di tab atau jendela baru, sehingga pengguna tidak menavigasi pergi dari situs Anda. Anda dapat mencapai ini menggunakan atribut `target`, mengaturnya ke `_blank`.

Contoh:

html

```
<a href="https://www.example.com" target="_blank">Kunjungi Example.com di tab baru</a>
```

Atribut `target="_blank"` memberi tahu browser untuk membuka link di tab atau jendela baru (tergantung pada pengaturan browser).

Praktik Terbaik untuk Link Eksternal

Gunakan HTTPS: Selalu gunakan `https://` untuk link eksternal bila memungkinkan, karena ini memastikan koneksi yang aman.

Teks Link Deskriptif: Gunakan teks link yang jelas dan deskriptif yang memberi tahu pengguna kemana link akan membawa mereka. Hindari teks generik seperti "Klik di sini."

Pertimbangkan Pengalaman Pengguna: Pikirkan apakah membuka link di tab baru sesuai untuk konteks pengguna.

Periksa Link Anda Secara Berkala: Website eksternal dapat berubah atau hilang, jadi secara berkala periksa link Anda untuk memastikan mereka masih berfungsi.

Menautkan ke Halaman Internal

Menautkan ke halaman internal dalam website Anda sendiri biasanya dilakukan menggunakan URL relatif. Pendekatan ini membuat website Anda lebih portabel dan lebih mudah dipelihara.

Contoh:

Misalkan Anda memiliki dua file HTML di direktori yang sama: `index.html` dan `about.html`. Untuk menautkan dari `index.html` ke `about.html`, Anda akan menggunakan kode berikut di `index.html`:

html

```
<a href="about.html">Pelajari lebih lanjut tentang kami</a>  
...
```

Menautkan ke Halaman di Subdirektori

Jika halaman target terletak di subdirektori, Anda perlu menyertakan nama subdirektori dalam URL relatif.

****Contoh:****

Misalkan Anda memiliki struktur direktori berikut:

...

```
mywebsite/  
├── index.html  
└── pages/  
    └── contact.html
```

Untuk menautkan dari `index.html` ke `contact.html`, Anda akan menggunakan kode berikut di `index.html`:

html

```
<a href="pages/contact.html">Hubungi kami</a>
```

Menautkan Kembali ke Direktori Induk

Terkadang, Anda perlu menautkan dari halaman di subdirektori kembali ke halaman di direktori induk. Anda dapat melakukan ini menggunakan notasi `../`, yang mewakili direktori induk.

Contoh:

Menggunakan struktur direktori yang sama seperti di atas, untuk menautkan dari `contact.html` kembali ke `index.html`, Anda akan menggunakan kode berikut di `contact.html`:

html

```
<a href="../index.html">Kembali ke beranda</a>
```

Menautkan ke Bagian Spesifik dalam Halaman (Anchor Internal)

Anda juga dapat menautkan ke bagian spesifik dalam halaman yang sama atau halaman lain menggunakan anchor internal. Ini dilakukan dengan menetapkan atribut `id` ke elemen HTML dan kemudian menautkan ke `id` tersebut menggunakan simbol `#` dalam URL.

Contoh:

Di `about.html`:

html

```
<h2 id="our-mission">Misi Kami</h2>
```

```
<p>Kami berusaha untuk...</p>
```

Di `index.html`:

html

```
<a href="about.html#our-mission">Pelajari tentang misi kami</a>
```

Ketika pengguna mengklik "Pelajari tentang misi kami," browser akan menavigasi ke halaman `about.html` dan scroll ke bagian dengan `id="our-mission"`.

Contoh: Menautkan ke bagian dalam halaman yang sama

html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Contoh Anchor Internal</title>
```

```
</head>
```

```
<body>
```

```
  <nav>
```

```
    <ul>
```

```
      <li><a href="#introduction">Pengenalan</a></li>
```

```
<li><a href="#section1">Bagian 1</a></li>
<li><a href="#section2">Bagian 2</a></li>
<li><a href="#conclusion">Kesimpulan</a></li>
</ul>
</nav>
```

```
<h2 id="introduction">Pengenalan</h2>
```

```
<p>Ini adalah bagian pengenalan. Tambahkan konten pengenalan Anda di sini.</p>
```

```
<h2 id="section1">Bagian 1</h2>
```

```
<p>Ini adalah bagian pertama. Tambahkan informasi detail dan konten yang relevan dengan bagian ini.</p>
```

```
<h2 id="section2">Bagian 2</h2>
```

```
<p>Ini adalah bagian kedua. Anda dapat menyertakan detail lebih spesifik atau contoh di bagian ini.</p>
```

```
<h2 id="conclusion">Kesimpulan</h2>
```

```
<p>Ini adalah kesimpulan. Ringkas poin Anda atau tambahkan pernyataan penutup di sini.</p>
```

```
</body>
```

```
</html>
```

Dalam contoh ini:

- Menu navigasi di atas berisi link yang menunjuk ke bagian berbeda dalam halaman yang sama menggunakan anchor internal (misalnya, `Pengenalan`).
- Setiap bagian ditandai dengan atribut `id` yang sesuai dengan anchor yang digunakan dalam link navigasi (misalnya, `<h2 id="introduction">Pengenalan</h2>`).

Praktik Terbaik untuk Link Internal

Gunakan URL Relatif: Lebih suka URL relatif untuk link internal untuk memastikan portabilitas.

Pertahankan Struktur Direktori yang Konsisten: Atur file website Anda dalam struktur direktori yang logis dan konsisten.

Gunakan ID yang Bermakna: Pilih atribut `id` yang deskriptif dan bermakna untuk anchor internal.

Uji Link Anda: Selalu uji link Anda untuk memastikan mereka berfungsi dengan benar.

Latihan Praktis

1. **Link Eksternal:** Buat halaman HTML dengan link ke mesin pencari favorit Anda. Buka link di tab baru.
2. **Link Internal:** Buat dua halaman HTML: `home.html` dan `services.html`. Tautkan dari `home.html` ke `services.html` dan kembali.
3. **Link Subdirektori:** Buat direktori bernama "blog" dan tempatkan file HTML bernama `article.html` di dalamnya. Tautkan dari halaman `home.html` Anda ke `article.html`.
4. **Anchor Internal:** Buat halaman HTML panjang dengan beberapa bagian. Tambahkan menu navigasi di atas yang menautkan ke setiap bagian menggunakan anchor internal.

Ringkasan dan Langkah Selanjutnya

Dalam pelajaran ini, Anda telah belajar cara membuat hyperlink ke halaman internal dan eksternal menggunakan tag `<a>`. Anda juga telah mempelajari perbedaan penting antara URL absolut dan relatif dan kapan menggunakan setiap jenis. Selain itu, Anda mendapatkan pengalaman praktis menggunakan anchor internal untuk menautkan ke bagian spesifik dalam halaman.

Di pelajaran berikutnya, Anda akan belajar cara menyematkan gambar ke dalam halaman HTML Anda menggunakan tag ``, lebih meningkatkan daya tarik visual dan konten informasi dari website Anda.

Menambahkan Gambar: Tag ``

Gambar adalah bagian penting dari web. Mereka meningkatkan pengalaman pengguna, membuat website lebih menarik dan menarik secara visual. Tag `` dalam HTML

memungkinkan Anda menyematkan gambar ke dalam halaman web Anda. Memahami cara menggunakan tag ini dan atributnya adalah fundamental untuk setiap pengembang web. Pelajaran ini akan membahas hal-hal penting dalam menambahkan gambar, termasuk menentukan sumber gambar, memberikan teks alternatif, dan mengontrol dimensi gambar. Kami akan fokus pada praktik terbaik saat ini untuk penanganan gambar di web, mempersiapkan Anda untuk teknik optimasi gambar yang dibahas di pelajaran berikutnya.

Tag ``: Menyematkan Gambar

Tag `` adalah elemen kosong, artinya tidak memerlukan tag penutup. Ini menggunakan atribut untuk menentukan sumber gambar dan properti lainnya. Atribut terpenting adalah `src`, yang mendefinisikan URL gambar.

Sintaks Dasar:

html

```

```

- **src:** Menentukan URL (lokasi) file gambar. Ini bisa berupa URL absolut (menunjuk ke gambar di website lain) atau URL relatif (menunjuk ke gambar dalam struktur file proyek Anda).
- **alt:** Memberikan teks alternatif untuk gambar. Teks ini ditampilkan jika gambar tidak dapat dimuat (misalnya, karena link rusak atau koneksi lambat) dan sangat penting untuk aksesibilitas.

Contoh:

Katakanlah Anda memiliki gambar bernama "logo.png" di direktori yang sama dengan file HTML Anda. Kode untuk menampilkan gambar ini adalah:

html

```

```

Jika gambar terletak di folder "images" dalam proyek Anda, atribut `src` akan menjadi:

html

```

```

Penjelasan:

- Browser membaca tag ``.
- Mengambil gambar dari URL yang ditentukan dalam atribut `src`.
- Menampilkan gambar pada halaman di mana tag berada.
- Jika gambar tidak dapat dimuat, teks dalam atribut `alt` akan ditampilkan sebagai gantinya.

Atribut Gambar Esensial: src dan alt

Atribut src: Menentukan Sumber Gambar

Atribut `src` (source) wajib untuk tag ``. Ini memberi tahu browser di mana menemukan file gambar. Anda dapat menggunakan URL absolut atau relatif.

URL Absolut: Alamat web lengkap yang termasuk protokol (misalnya, `http://` atau `https://`) dan nama domain.

html

```

```

Gunakan URL absolut ketika menautkan ke gambar yang dihosting di website lain. Namun, ketahuilah bahwa ketersediaan gambar bergantung pada website eksternal.

URL Relatif: Menentukan jalur ke file gambar relatif terhadap file HTML saat ini.

html

```

```

```
<!-- jika gambar ada di folder "images" -->
```

html

```

```

```
<!-- jika gambar ada di folder "images" satu level di atas file HTML saat ini. ../ berarti "naik satu direktori" -->
```


URL relatif umumnya lebih disukai untuk gambar dalam website Anda sendiri karena lebih portabel. Jika Anda memindahkan website ke domain berbeda, link akan tetap berfungsi selama struktur direktori tetap sama.

Memilih antara URL Absolut dan Relatif:

- Gunakan URL relatif untuk gambar yang merupakan bagian dari konten website Anda. Ini membuat website Anda lebih portabel.
- Gunakan URL absolut hanya ketika diperlukan untuk menautkan ke gambar yang dihosting di website eksternal. Ketahuilah bahwa Anda bergantung pada website eksternal untuk menjaga ketersediaan gambar.

Atribut alt: Memberikan Teks Alternatif

Atribut `alt` (alternative text) sangat direkomendasikan dan penting untuk aksesibilitas dan SEO (Search Engine Optimization). Ini memberikan deskripsi teks dari gambar, yang ditampilkan dalam situasi berikut:

- Gambar gagal dimuat (misalnya, link rusak, koneksi lambat).
- Pengguna menggunakan pembaca layar (aplikasi perangkat lunak yang memungkinkan pengguna tunanetra mengakses konten digital).
- Mesin pencari menggunakan teks alt untuk memahami konten gambar.

Praktik Terbaik untuk Teks alt:

Deskriptif: Teks alt harus secara akurat menggambarkan konten gambar.

- Baik: ``
- Buruk: `` (terlalu umum)
- Buruk: `` (teks alt kosong padahal seharusnya deskriptif - lihat pengecualian di bawah)

Ringkas: Buat teks alt singkat dan to the point (idealnya di bawah 125 karakter).

Sertakan kata kunci yang relevan: Jika gambar relevan dengan konten halaman, sertakan kata kunci yang relevan dalam teks alt. Ini dapat membantu meningkatkan SEO website Anda.

Hindari redundansi: Jangan ulangi teks di sekitarnya dalam teks alt.

Gunakan atribut alt kosong untuk gambar yang murni dekoratif: Jika gambar murni dekoratif dan tidak menyampaikan konten yang berarti, gunakan atribut alt kosong (`alt=""`). Ini memberi tahu pembaca layar untuk mengabaikan gambar. Ini penting karena pembaca layar sering kali akan membaca nama file jika tidak ada teks alt yang bisa membingungkan.

Contoh:

Gambar Informatif:

html

```

```

Gambar Dekoratif:

html

```

```

Gambar sebagai Link:

html

```
<a href="https://www.example.com">
  
</a>
```

Mengontrol Dimensi Gambar: Atribut width dan height

Atribut `width` dan `height` memungkinkan Anda menentukan dimensi gambar dalam piksel.

html

```

```

Mengapa menggunakan atribut width dan height?

Peningkatan Kecepatan Muat Halaman: Ketika browser mengetahui dimensi gambar sebelumnya, ia dapat mengalokasikan ruang yang tepat untuk gambar di halaman sebelum gambar sepenuhnya diunduh. Ini mencegah layout halaman bergeser saat gambar dimuat, meningkatkan pengalaman pengguna.

Penskalaan Gambar: Anda dapat menggunakan width dan height untuk menskalakan gambar. Misalnya, jika gambar asli Anda adalah 1000x600 piksel, Anda dapat menampilkannya pada 500x300 piksel menggunakan atribut ini. Namun, umumnya lebih baik mengubah ukuran gambar menggunakan program pengeditan gambar sebelum mengunggahnya ke website Anda (kita akan membahas optimasi gambar di pelajaran berikutnya). Menggunakan width dan height untuk menskalakan gambar besar masih dapat memperlambat waktu muat halaman Anda.

Pendekatan Modern (CSS):

Meskipun atribut width dan height adalah HTML yang valid, pendekatan modern adalah mengontrol dimensi gambar menggunakan CSS (Cascading Style Sheets). CSS memberikan lebih banyak fleksibilitas dan kontrol atas styling gambar. Kita akan membahas CSS di modul mendatang, tetapi berikut adalah pratinjau singkat:

html

```

```

atau, lebih baik lagi, menggunakan kelas CSS:

html

```

```

dan di file CSS Anda (yang akan kita bahas di modul mendatang):

css

```
.responsive-image {  
  width: 100%; /* Gambar akan mengambil 100% dari lebar containernya */  
  height: auto; /* Tinggi akan menyesuaikan secara otomatis untuk mempertahankan rasio aspek */  
}
```

Mempertahankan Rasio Aspek:

Ketika hanya menentukan satu dimensi (width atau height), browser akan secara otomatis menyesuaikan dimensi lainnya untuk mempertahankan rasio aspek gambar.

html

```

```

<!-- Tinggi akan disesuaikan secara otomatis -->

Pertimbangan Penting:

Distorsi: Jika Anda menentukan atribut width dan height dan rasio aspeknya tidak cocok dengan gambar asli, gambar akan terdistorsi (direntang atau diperas).

Kualitas Gambar: Penskalaan gambar menggunakan atribut width dan height (atau CSS) terkadang dapat mengurangi kualitas gambar, terutama jika Anda secara signifikan meningkatkan ukuran gambar. Hampir selalu lebih baik menggunakan editor gambar untuk mengubah ukuran gambar ke dimensi yang diinginkan sebelum mengunggahnya ke website Anda.

Contoh Praktis dan Demonstrasi

Mari kita asumsikan Anda sedang membangun website pribadi sederhana. Anda ingin menyertakan foto profil dan beberapa gambar yang menampilkan hobi Anda.

Contoh 1: Foto Profil

html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Website Pribadi Saya</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Tentang Saya</h1>
```

```
  
```

```
  <p>Selamat datang di website saya! Saya seorang web developer...</p>
```

```
</body>
```

```
</html>
```

Dalam contoh ini:

- Kita asumsikan Anda memiliki file bernama "profile.jpg" di folder "images" dalam proyek Anda.
- Teks alt memberikan deskripsi gambar untuk aksesibilitas.
- Kita mengatur width dan height ke 200 piksel untuk memastikan gambar ditampilkan pada ukuran yang diinginkan.

Contoh 2: Menampilkan Hobi

html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Hobi Saya</title>
```

```
</head>
```

```
<body>
```

```
  <h2>Hobi Saya</h2>
```

```
  <div>
```

```
    
```

```
    <p>Saya suka hiking di pegunungan. Udara segar dan pemandangan indah sungguh luar biasa.</p>
```

```
  </div>
```

```
  <div>
```

```
    
```

```
    <p>Saya juga suka coding dan membangun website.</p>
```

```
  </div>
```

```
</body>
```

```
</html>
```

Dalam contoh ini:

- Kita menampilkan dua gambar yang mewakili hobi.
- Setiap gambar memiliki teks alt yang deskriptif.

- Kita mengatur width ke 300 piksel untuk setiap gambar, memungkinkan browser untuk secara otomatis menyesuaikan tinggi untuk mempertahankan rasio aspek.

Contoh 3: Menangani Gambar yang Hilang

Katakanlah Anda secara tidak sengaja salah mengeja nama file salah satu gambar Anda:

html

```

```

Karena file "hikin.jpg" tidak ada, browser akan menampilkan teks alt ("Hiking di pegunungan") sebagai pengganti gambar. Inilah mengapa atribut alt sangat penting!

Latihan

1. **Galeri Gambar:** Buat halaman HTML sederhana dengan galeri gambar. Sertakan setidaknya tiga gambar dengan atribut src dan alt yang sesuai. Gunakan URL relatif.
2. **Gambar Rusak:** Secara sengaja buat link gambar yang rusak (misalnya, dengan salah mengeja nama file). Verifikasi bahwa teks alt ditampilkan dengan benar.
3. **Mengubah Ukuran Gambar:** Gunakan atribut width dan height untuk mengubah ukuran gambar. Eksperimen dengan nilai yang berbeda untuk melihat bagaimana pengaruhnya terhadap tampilan gambar. Apa yang terjadi jika Anda hanya mengatur width atau hanya height? Apa yang terjadi jika Anda mengatur width dan height ke nilai yang tidak cocok dengan rasio aspek asli?
4. **Gambar Dekoratif:** Tambahkan gambar dekoratif ke halaman Anda (misalnya, border atau gambar latar belakang). Gunakan atribut alt kosong (`alt=""`) untuk gambar dekoratif. Jelaskan mengapa ini adalah pendekatan yang benar dalam kasus ini.

Ringkasan dan Langkah Selanjutnya

Dalam pelajaran ini, Anda belajar cara menyematkan gambar di halaman HTML Anda menggunakan tag ``. Anda mengeksplorasi atribut esensial `src` dan `alt`, dan cara menggunakannya secara efektif. Anda juga belajar cara mengontrol dimensi gambar menggunakan atribut `width` dan `height`, dan diperkenalkan pada pendekatan modern menggunakan CSS (yang akan kita bahas di modul mendatang).

Di pelajaran berikutnya, "Optimasi Gambar untuk Web," Anda akan belajar cara mengoptimalkan gambar Anda untuk web untuk meningkatkan kecepatan muat halaman dan pengalaman pengguna. Ini akan melibatkan teknik seperti memilih format gambar yang tepat, mengompresi gambar, dan menggunakan gambar responsif. Anda juga akan mendalami kapan dan bagaimana menggunakan atribut `width` dan `height` versus mengontrol dimensi gambar menggunakan CSS.

Atribut Gambar: `src`, `alt`, `width`, `height`

Gambar adalah bagian vital dari web, menambahkan daya tarik visual dan konteks pada konten Anda. Meskipun tag `` itu sendiri memberi tahu browser untuk menampilkan gambar, atribut di dalam tag itulah yang memberikan informasi penting tentang lokasi, deskripsi, dan ukuran gambar. Atribut `src`, `alt`, `width`, dan `height` adalah fundamental untuk mengontrol bagaimana gambar ditampilkan dan memastikan aksesibilitas. Memahami cara menggunakan atribut ini dengan benar sangat penting untuk membangun halaman web yang terstruktur dengan baik dan ramah pengguna.

Memahami Atribut `src`

Atribut `src`, singkatan dari "source," adalah atribut terpenting dari tag ``. Ini menentukan URL gambar yang ingin Anda tampilkan di halaman web Anda. URL memberi tahu browser di mana menemukan file gambar.

URL Absolut vs. Relatif (Ditinjau Kembali)

Seperti yang kita bahas dalam pelajaran tentang "Memahami URL Absolut dan Relatif," URL dapat berupa absolut atau relatif. Prinsip yang sama berlaku untuk atribut `src` gambar.

URL Absolut: URL absolut memberikan alamat lengkap gambar, termasuk protokol (misalnya, `https://`) dan nama domain. Misalnya:

html

```

```

Menggunakan URL absolut berguna ketika Anda ingin menampilkan gambar yang dihosting di website berbeda atau CDN (Content Delivery Network).

URL Relatif: URL relatif menentukan lokasi gambar relatif terhadap dokumen HTML saat ini. Jika gambar Anda berada di direktori yang sama dengan file HTML Anda, Anda cukup menggunakan nama file gambar:

html

```

```

Jika gambar berada di subdirektori bernama "images," Anda akan menggunakan:

html

```

```

URL relatif umumnya lebih disukai untuk gambar dalam website Anda sendiri karena lebih portabel. Jika Anda memindahkan website ke domain berbeda, URL relatif akan tetap berfungsi selama struktur direktori tetap sama.

Menangani Jalur Gambar dengan Benar

Kesalahan umum adalah menggunakan jalur gambar yang salah, yang mengakibatkan gambar rusak di halaman web Anda. Berikut beberapa tips untuk menghindari ini:

Periksa ulang ejaan Anda: Pastikan nama file dan nama direktori dalam atribut `src` Anda cocok dengan nama sebenarnya dari file dan direktori gambar Anda. Nama file case-sensitive di banyak server web, jadi `Logo.png` berbeda dari `logo.png`.

Gunakan jalur relatif yang benar: Jika gambar Anda berada di subdirektori, pastikan Anda menyertakan jalur yang benar ke direktori tersebut. Kesalahan umum adalah melupakan `/` di awal untuk jalur relatif terhadap direktori root.

Uji jalur gambar Anda: Setelah menambahkan gambar ke halaman web Anda, selalu uji di browser Anda untuk memastikan ditampilkan dengan benar. Jika gambar tidak dimuat, periksa konsol developer browser Anda untuk pesan error.

Pentingnya Atribut alt

Atribut `alt` memberikan teks alternatif untuk gambar. Teks ini ditampilkan jika gambar tidak dapat dimuat (misalnya, karena link rusak atau error jaringan). Lebih penting lagi, ini sangat penting untuk aksesibilitas. Pembaca layar menggunakan teks alt untuk menjelaskan gambar kepada pengguna tunanetra.

Menulis Teks alt yang Efektif

Menulis teks alt yang baik sangat penting untuk membuat website Anda dapat diakses dan ramah pengguna. Berikut beberapa panduan:

Deskriptif: Teks alt harus secara akurat menggambarkan konten dan fungsi gambar. Bayangkan Anda menjelaskan gambar kepada seseorang yang tidak dapat melihatnya.

Ringkas: Buat teks alt relatif singkat, idealnya di bawah 125 karakter. Pembaca layar mungkin memotong deskripsi yang lebih panjang.

Sertakan kata kunci yang relevan: Jika gambar relevan dengan konten halaman Anda, sertakan kata kunci yang relevan dalam teks alt. Ini dapat meningkatkan SEO (Search Engine Optimization) website Anda.

Hindari redundansi: Jangan ulangi informasi yang sama yang sudah ada dalam teks di sekitarnya. Teks alt harus memberikan konteks tambahan atau menjelaskan gambar lebih detail.

Gunakan atribut alt kosong untuk gambar yang murni dekoratif: Jika gambar murni dekoratif dan tidak menyampaikan informasi penting apa pun, gunakan atribut alt kosong (`alt=""`). Ini memberi tahu pembaca layar untuk mengabaikan gambar. Misalnya:

html

```

```

Contoh Teks alt yang Baik dan Buruk

Mari kita pertimbangkan beberapa contoh untuk mengilustrasikan perbedaan antara teks alt yang baik dan buruk:

Gambar: Foto anak anjing Golden Retriever bermain dengan bola.

- **Teks alt buruk:** `alt="gambar"` atau `alt="foto"` (Tidak cukup deskriptif)

- **Teks alt lebih baik:** `alt="Anak anjing Golden Retriever bermain dengan bola di rumput"` (Deskriptif dan ringkas)

Gambar: Logo perusahaan.

- **Teks alt buruk:** `alt="logo"` (Tidak cukup spesifik)
- **Teks alt lebih baik:** `alt="Logo Acme Corp"` (Mengidentifikasi perusahaan)

Gambar: Tombol yang bertuliskan "Kirim."

- **Teks alt buruk:** `alt="tombol"` (Redundan)
- **Teks alt lebih baik:** `alt="Kirim formulir"` (Menjelaskan fungsi tombol)

Pentingnya Aksesibilitas

Memberikan teks alt untuk semua gambar Anda bukan hanya praktik terbaik; ini adalah bagian penting dari aksesibilitas web. Dengan memberikan teks alt yang deskriptif, Anda memastikan bahwa pengguna tunanetra dapat memahami konten dan fungsi gambar Anda, membuat website Anda lebih inklusif dan ramah pengguna. Mengabaikan atribut alt dapat menyebabkan pengalaman yang membuat frustrasi dan tidak dapat diakses bagi pengguna dengan disabilitas.

Mengontrol Ukuran Gambar dengan Atribut width dan height

Atribut `width` dan `height` menentukan dimensi gambar dalam piksel. Meskipun Anda dapat menggunakan CSS untuk mengontrol ukuran gambar (yang akan kita pelajari di modul mendatang), mengatur atribut ini langsung di tag `` masih dianggap praktik yang baik untuk beberapa alasan.

Mengapa Menggunakan Atribut width dan height?

Mencegah Pergeseran Layout: Ketika browser memuat halaman web, ia perlu menghitung layout halaman, termasuk ruang yang akan ditempati gambar. Jika atribut width dan height tidak ditentukan, browser mungkin tidak tahu berapa banyak ruang yang harus dialokasikan untuk gambar sampai ia sepenuhnya memuat file gambar. Ini dapat menyebabkan pergeseran layout, di mana elemen di halaman melompat-lompat saat gambar dimuat. Menentukan atribut ini memungkinkan browser untuk mereservasi jumlah ruang yang tepat

untuk gambar dari awal, mencegah pergeseran layout dan meningkatkan pengalaman pengguna.

Rendering Lebih Cepat: Dengan memberikan atribut `width` dan `height`, Anda memberi browser petunjuk tentang dimensi gambar. Ini memungkinkan browser untuk mulai me-render halaman lebih cepat, bahkan sebelum file gambar sepenuhnya diunduh.

Fallback untuk CSS: Dalam kasus langka di mana CSS tidak dimuat atau tidak didukung, atribut `width` dan `height` akan memastikan bahwa gambar ditampilkan pada ukuran yang benar.

Mengatur Atribut `width` dan `height`

Atribut `width` dan `height` menerima nilai dalam piksel. Misalnya:

html

```

```

Kode ini memberi tahu browser untuk menampilkan gambar pada lebar 600 piksel dan tinggi 400 piksel.

Mempertahankan Rasio Aspek

Ketika Anda menentukan atribut `width` dan `height`, browser akan meregangkan atau menyusutkan gambar agar sesuai dengan dimensi yang ditentukan. Jika Anda hanya menentukan salah satu dari atribut ini, browser akan secara otomatis menyesuaikan atribut lainnya untuk mempertahankan rasio aspek gambar.

Misalnya, jika gambar asli Anda adalah 1200x800 piksel (rasio aspek 3:2) dan Anda hanya menentukan `width="300"`, browser akan secara otomatis menghitung tinggi menjadi 200 piksel, mempertahankan rasio aspek 3:2.

Jika Anda ingin mengubah rasio aspek gambar, Anda harus menentukan atribut `width` dan `height`. Namun, ini dapat mendistorsi gambar jika rasio aspek yang ditentukan sangat berbeda dari yang asli.

Praktik Terbaik untuk `width` dan `height`

Gunakan dimensi gambar asli: Idealnya, atribut `width` dan `height` harus cocok dengan dimensi asli gambar. Ini memastikan bahwa gambar ditampilkan pada resolusi optimalnya dan menghindari penskalaan yang tidak perlu.

Pertimbangkan gambar responsif: Untuk website responsif, Anda mungkin ingin menggunakan CSS untuk mengontrol ukuran gambar berdasarkan ukuran layar. Dalam kasus ini, Anda masih dapat menggunakan atribut `width` dan `height` untuk mencegah pergeseran layout, tetapi atur nilainya ke dimensi intrinsik gambar. Kemudian, gunakan CSS untuk membuat gambar menskalakan secara responsif (lebih lanjut tentang ini di modul selanjutnya).

Hindari upscaling gambar: Upscaling gambar (menampilkannya pada ukuran yang lebih besar dari dimensi aslinya) dapat menghasilkan gambar yang buram atau pixelated. Umumnya lebih baik menggunakan gambar yang cukup besar untuk kebutuhan Anda dan kemudian menskalakannya ke bawah jika perlu.

Contoh Praktis dan Demonstrasi

Mari kita pertimbangkan contoh praktis tentang cara menggunakan atribut `src`, `alt`, `width`, dan `height` dalam dokumen HTML. Misalkan kita sedang membangun halaman web sederhana untuk blog perjalanan. Kita ingin menyertakan foto Menara Eiffel di Paris.

```
html
<!DOCTYPE html>
<html>
<head>
  <title>Blog Perjalanan</title>
</head>
<body>
  <h1>Perjalanan Saya ke Paris</h1>
  <p>Berikut beberapa foto dari perjalanan saya ke Paris baru-baru ini.</p>
  
  <p>Menara Eiffel adalah menara kisi besi tempa di Champ de Mars di Paris, Prancis.
Dinamai setelah insinyur Gustave Eiffel, yang perusahaannya merancang dan membangun
menara tersebut.</p>
</body>
```

`</html>`

Dalam contoh ini:

- Atribut `src` menunjuk ke file gambar `eiffel-tower.jpg`, yang terletak di subdirektori `images`.
- Atribut `alt` memberikan teks deskriptif untuk gambar, yang membantu untuk aksesibilitas dan SEO.
- Atribut `width` dan `height` menentukan dimensi gambar dalam piksel, mencegah pergeseran layout dan memastikan bahwa gambar ditampilkan pada ukuran yang benar.

Demonstrasi:

1. Buat file HTML baru bernama `travel-blog.html`.
2. Buat subdirektori bernama `images`.
3. Unduh gambar Menara Eiffel dan simpan sebagai `eiffel-tower.jpg` di subdirektori `images`.
4. Salin kode HTML di atas ke file `travel-blog.html` Anda.
5. Buka file `travel-blog.html` di browser Anda. Anda seharusnya melihat foto Menara Eiffel ditampilkan di halaman, bersama dengan teks di sekitarnya.

Eksperimen:

- Coba ubah atribut `width` dan `height` untuk melihat bagaimana pengaruhnya terhadap ukuran gambar.
- Coba hapus atribut `alt` dan lihat apa yang terjadi ketika gambar tidak dapat dimuat.
- Coba gunakan jalur gambar yang salah dalam atribut `src` dan lihat bagaimana browser menangani error.

Latihan

1. **Website Pribadi:** Buat website pribadi sederhana dengan foto diri Anda. Gunakan atribut `src`, `alt`, `width`, dan `height` untuk menampilkan gambar dengan benar.
2. **Halaman Produk:** Rancang halaman produk untuk website e-commerce. Sertakan gambar produk dengan teks `alt` dan dimensi yang sesuai.

3. **Posting Blog:** Tulis posting blog tentang topik pilihan Anda. Sertakan gambar yang relevan dengan teks alt yang deskriptif dan dimensi yang sesuai.

Ringkasan dan Langkah Selanjutnya

Dalam pelajaran ini, kita belajar tentang atribut gambar fundamental: `src`, `alt`, `width`, dan `height`. Atribut `src` menentukan URL gambar, sementara atribut `alt` memberikan teks alternatif untuk aksesibilitas dan SEO. Atribut `width` dan `height` mengontrol dimensi gambar dan mencegah pergeseran layout.

Memahami cara menggunakan atribut ini dengan benar sangat penting untuk membangun halaman web yang terstruktur dengan baik, ramah pengguna, dan dapat diakses.

Di pelajaran berikutnya, kita akan mengeksplorasi teknik untuk mengoptimalkan gambar untuk web, termasuk kompresi gambar dan gambar responsif. Kita juga akan membahas format gambar yang berbeda dan cara memilih format terbaik untuk kebutuhan Anda.

Baik, saya akan merapikan dan menerjemahkan dokumen keenam belas ini ke Bahasa Indonesia:

Optimasi Gambar untuk Web

Optimasi gambar sangat penting untuk website yang cepat dan efisien. File gambar yang besar dapat secara signifikan memperlambat waktu muat halaman, menyebabkan pengalaman pengguna yang buruk dan berpotensi berdampak pada peringkat mesin pencari website Anda. Dengan mengoptimalkan gambar Anda, Anda dapat mengurangi ukuran file tanpa mengorbankan kualitas, menghasilkan waktu muat yang lebih cepat dan pengalaman keseluruhan yang lebih baik untuk pengunjung Anda. Pelajaran ini akan mendalami teknik dan praktik terbaik untuk optimasi gambar di web.

Memahami Format File Gambar

Memilih format gambar yang tepat adalah langkah pertama dalam optimasi. Format yang berbeda cocok untuk jenis gambar yang berbeda dan menawarkan tingkat kompresi yang bervariasi.

JPEG/JPG

Penggunaan Terbaik: Foto dan gambar dengan banyak warna dan gradien.

Kompresi: Menggunakan kompresi lossy, yang berarti beberapa data gambar dibuang untuk mengurangi ukuran file. Ini dapat menghasilkan kehilangan kualitas, terutama dengan tingkat kompresi tinggi.

Kelebihan: Ukuran file kecil, didukung secara luas.

Kekurangan: Kompresi lossy dapat menurunkan kualitas gambar.

Contoh: Foto pemandangan yang indah akan cocok untuk format JPEG.

PNG

Penggunaan Terbaik: Gambar dengan garis tajam, teks, logo, dan grafik dengan transparansi.

Kompresi: Menggunakan kompresi lossless, yang berarti tidak ada data gambar yang hilang selama kompresi.

Kelebihan: Mendukung transparansi, kompresi lossless mempertahankan kualitas gambar.

Kekurangan: Umumnya ukuran file lebih besar dari JPEG untuk foto.

Contoh: Logo perusahaan dengan latar belakang transparan harus disimpan sebagai PNG.

GIF

Penggunaan Terbaik: Animasi sederhana dan gambar dengan warna terbatas (256 warna atau kurang).

Kompresi: Menggunakan kompresi lossless, tetapi palet warna terbatas.

Kelebihan: Mendukung animasi, kompresi lossless.

Kekurangan: Palet warna terbatas, tidak cocok untuk foto.

Contoh: Animasi pendek yang berulang, seperti bola memantul, bisa menjadi GIF.

WebP

Penggunaan Terbaik: Format modern yang menawarkan kompresi lossy dan lossless, sering kali menghasilkan ukuran file yang lebih kecil dari JPEG dan PNG dengan kualitas yang sebanding.

Kompresi: Mendukung kompresi lossy dan lossless, serta transparansi dan animasi.

Kelebihan: Kompresi dan kualitas superior dibandingkan JPEG dan PNG, mendukung animasi dan transparansi.

Kekurangan: Tidak didukung oleh semua browser lama (tetapi dukungan berkembang pesat).

Contoh: Hampir semua gambar, dari foto hingga grafik, dapat mengambil manfaat dari disimpan sebagai WebP.

AVIF

Penggunaan Terbaik: Format gambar modern berbasis codec video AV1, menawarkan kompresi yang lebih baik dari WebP dalam banyak kasus.

Kompresi: Mendukung kompresi lossy dan lossless, serta transparansi dan animasi.

Kelebihan: Kompresi sangat baik, berpotensi ukuran file lebih kecil dari WebP.

Kekurangan: Dukungan browser terbatas dibandingkan WebP (tetapi berkembang), encoding dapat lebih intensif CPU.

Contoh: Foto beresolusi tinggi di mana ukuran file sangat penting.

Skenario Contoh: Bayangkan Anda membuat website untuk portofolio fotografi. Anda kemungkinan akan menggunakan JPEG untuk sebagian besar foto Anda karena kemampuannya menangani warna dan gradien kompleks dengan ukuran file yang relatif kecil. Namun, untuk logo Anda, yang menampilkan garis tajam dan transparansi, Anda akan memilih PNG. Untuk gambar di blog Anda yang menampilkan karya Anda, Anda mungkin

mempertimbangkan menggunakan format WebP untuk mengurangi waktu muat blog Anda tanpa kehilangan kualitas gambar.

Teknik Optimasi Gambar

Selain memilih format yang tepat, beberapa teknik dapat lebih mengoptimalkan gambar Anda untuk web.

Mengubah Ukuran Gambar

Konsep: Menyajikan gambar pada dimensi yang tepat saat ditampilkan di website Anda.

Penjelasan: Menampilkan gambar besar yang diperkecil oleh browser membuang bandwidth.

Cara Implementasi: Gunakan perangkat lunak pengeditan gambar (seperti Adobe Photoshop, GIMP, atau alat online) untuk mengubah ukuran gambar ke dimensi yang diperlukan sebelum mengunggahnya ke website Anda.

Contoh: Jika gambar akan ditampilkan pada 300x200 piksel, ubah ukurannya ke dimensi tersebut sebelum mengunggah. Jangan unggah gambar 1200x800 piksel dan bergantung pada browser untuk memperkecilnya.

Kompresi

Konsep: Mengurangi ukuran file gambar dengan menghapus data yang tidak perlu.

Penjelasan: Algoritma kompresi lossy dan lossless dapat digunakan untuk mengurangi ukuran file.

Cara Implementasi:

Kompresi Lossy: Mengurangi ukuran file dengan membuang beberapa data gambar. Sesuaikan tingkat kompresi dengan hati-hati untuk menemukan keseimbangan antara ukuran file dan kualitas gambar. Gunakan ini terutama untuk gambar JPEG.

Kompresi Lossless: Mengurangi ukuran file tanpa kehilangan data gambar apa pun. Gunakan ini untuk gambar PNG dan GIF, atau ketika mempertahankan kualitas gambar adalah yang terpenting.

Alat: Perangkat lunak pengeditan gambar, pengoptimal gambar online (seperti TinyPNG atau ImageOptim), dan alat command-line (seperti ImageMagick) dapat digunakan untuk kompresi.

Contoh: Mengurangi pengaturan kualitas JPEG dari 90% menjadi 70% dapat secara signifikan mengurangi ukuran file dengan perbedaan visual minimal, terutama pada layar yang lebih kecil.

Mengoptimalkan untuk Layar Retina

Konsep: Layar retina (layar kepadatan tinggi) memerlukan gambar dengan kepadatan piksel dua kali lipat agar tampak tajam.

Penjelasan: Menyajikan gambar resolusi standar pada layar retina menghasilkan tampilan yang buram.

Cara Implementasi:

- Buat gambar pada 2x ukuran tampilan yang dimaksudkan. Misalnya, jika gambar ditampilkan pada 300x200 piksel, buat versi 600x400 piksel.
- Gunakan atribut `srcset` dalam tag `` untuk menyediakan sumber gambar yang berbeda untuk kepadatan layar yang berbeda.

html

```

```

Penjelasan: Browser akan secara otomatis memilih gambar yang sesuai berdasarkan kepadatan piksel perangkat. 1x mengacu pada resolusi standar (kepadatan piksel 1x), dan 2x mengacu pada resolusi retina (kepadatan piksel 2x).

Contoh: Logo yang ditampilkan pada 100x50 piksel harus memiliki versi 200x100 piksel untuk layar retina.

Menggunakan CSS Sprites

Konsep: Menggabungkan beberapa gambar kecil menjadi satu gambar yang lebih besar untuk mengurangi jumlah permintaan HTTP.

Penjelasan: Setiap gambar di website memerlukan permintaan HTTP terpisah, yang dapat menambah overhead dan memperlambat waktu muat halaman.

Cara Implementasi:

- Gabungkan semua gambar kecil menjadi satu file gambar (sprite).
- Gunakan properti CSS `background-image`, `background-position`, dan `width/height` untuk menampilkan gambar individual dalam sprite.

Contoh: Menggunakan sprite untuk ikon, seperti ikon media sosial atau panah navigasi, dapat secara signifikan mengurangi jumlah permintaan HTTP.

Lazy Loading

Konsep: Memuat gambar hanya ketika mereka terlihat di viewport (area yang terlihat dari jendela browser).

Penjelasan: Gambar di bawah fold (tidak langsung terlihat) tidak dimuat sampai pengguna menggulir ke bawah, meningkatkan waktu muat halaman awal.

Cara Implementasi:

Gunakan atribut `loading="lazy"` dalam tag ``.

html

```

```

Atau, gunakan library JavaScript untuk teknik lazy loading yang lebih canggih dan kompatibilitas browser.

Contoh: Pada posting blog panjang dengan banyak gambar, lazy loading dapat secara signifikan meningkatkan waktu muat awal dengan hanya memuat gambar yang awalnya terlihat.

Menggunakan Content Delivery Network (CDN)

Konsep: Mendistribusikan konten website Anda (termasuk gambar) di beberapa server yang terletak di lokasi geografis yang berbeda.

Penjelasan: CDN memastikan bahwa pengguna menerima konten dari server yang paling dekat dengan mereka, mengurangi latensi dan meningkatkan waktu muat.

Cara Implementasi: Daftar untuk layanan CDN (seperti Cloudflare, AWS CloudFront, atau Akamai) dan konfigurasi website Anda untuk menggunakan CDN.

Contoh: Jika website Anda memiliki pengunjung dari seluruh dunia, menggunakan CDN memastikan bahwa pengguna di Eropa menerima gambar dari server di Eropa, bukan dari server di Amerika Serikat, yang akan lebih lambat.

Skenario Hipotetis: Bayangkan Anda membangun website e-commerce. Anda memiliki ratusan gambar produk. Dengan mengubah ukuran gambar ke ukuran tampilan yang sesuai, menggunakan format WebP di mana didukung, menerapkan lazy loading, dan menggunakan CDN, Anda dapat secara drastis meningkatkan kinerja dan pengalaman pengguna website. Jika Anda tidak mengoptimalkan gambar, pengguna harus menunggu lebih lama untuk setiap halaman dimuat, terutama mereka yang memiliki koneksi internet lebih lambat, dan mungkin meninggalkan situs.

Latihan Praktis: Mengoptimalkan Gambar untuk Website

Mari kita tinjau kembali contoh halaman resep dari pelajaran sebelumnya. Misalkan Anda telah menambahkan gambar ke halaman resep, dan halaman memuat dengan lambat. Berikut cara Anda dapat mengoptimalkan gambar:

1. **Identifikasi Gambar Besar:** Gunakan alat developer browser Anda (tab Network) untuk mengidentifikasi gambar yang paling berkontribusi pada waktu muat halaman.
2. **Pilih Format yang Tepat:** Tentukan format yang sesuai untuk setiap gambar. Untuk foto hidangan, gunakan JPEG atau WebP. Untuk ikon atau grafik, gunakan PNG atau SVG (Scalable Vector Graphics - format berbasis vektor yang ideal untuk logo dan ikon).
3. **Ubah Ukuran Gambar:** Ubah ukuran gambar ke dimensi yang tepat saat ditampilkan di halaman.
4. **Kompres Gambar:** Gunakan alat optimasi gambar untuk mengompres gambar. Eksperimen dengan tingkat kompresi yang berbeda untuk menemukan keseimbangan antara ukuran file dan kualitas.
5. **Implementasikan Lazy Loading:** Tambahkan atribut `loading="lazy"` ke tag ``.

6. **Uji dan Ukur:** Gunakan alat developer browser Anda untuk mengukur waktu muat halaman sebelum dan sesudah optimasi. Anda seharusnya melihat peningkatan yang signifikan.

Misalnya, halaman resep asli mungkin berisi gambar JPEG 2MB dari hidangan jadi yang ditampilkan pada 600x400 piksel. Dengan mengubah ukuran gambar ke 600x400 piksel dan mengompresnya, Anda dapat mengurangi ukuran file menjadi 200KB atau kurang. Menerapkan lazy loading akan lebih meningkatkan waktu muat awal dengan mencegah gambar dimuat sampai terlihat di viewport.

Aplikasi Dunia Nyata

Banyak website sekarang menggunakan optimasi gambar otomatis sebagai bagian dari proses build mereka. Alat seperti gulp-imagemin atau plugin webpack dapat secara otomatis mengoptimalkan gambar ketika website dibangun, memastikan bahwa semua gambar dioptimalkan dengan benar sebelum di-deploy. Layanan seperti Cloudinary juga menyediakan solusi manajemen gambar komprehensif, termasuk optimasi otomatis, perubahan ukuran, dan konversi format. Alat-alat ini dapat secara signifikan menyederhanakan proses optimasi gambar dan memastikan optimasi yang konsisten di seluruh website. Misalnya, situs e-commerce dengan ribuan gambar produk akan mendapat manfaat dari mengotomatisasi proses optimasi untuk memastikan gambar dimuat dengan cepat untuk semua pengguna.

Ringkasan

Dalam pelajaran ini, kita mengeksplorasi teknik esensial untuk mengoptimalkan gambar untuk web, termasuk memilih format file yang tepat, mengubah ukuran gambar, kompresi, mengoptimalkan untuk layar retina, menggunakan CSS sprites, lazy loading, dan menggunakan CDN. Dengan menerapkan teknik ini, Anda dapat secara signifikan mengurangi ukuran file gambar, meningkatkan waktu muat halaman, dan meningkatkan pengalaman pengguna di website Anda. Selanjutnya, kita akan mengeksplorasi HTML semantik, di mana Anda akan belajar cara menyusun konten Anda dengan cara yang bermakna. Memahami pentingnya optimasi gambar mempersiapkan Anda untuk langkah selanjutnya, yaitu meningkatkan struktur keseluruhan website Anda.