

HTTP

HTTP (Hypertext Transfer Protocol): Protokol untuk mentransmisikan dokumen hypermedia (HTML, CSS, JavaScript, dll.) antara client dan server.

Arsitektur Client-Server, Client (browser) mengirim HTTP Request ke Server (web server). Server merespons dengan HTTP Response. Stateless, Setiap request independen (tidak ada memori request sebelumnya).

Versi	Tahun	Status	HTTP Request: GET /home HTTP/1.1
HTTP/0.9	1991	Obsolete	Host: example.com
HTTP/1.0	1996	Obsolete	User-Agent: Chrome
HTTP/1.1	1997	Standar	HTTP Response: HTTP/1.1 200 OK
HTTP/2	2015	Standar	Content-Type: text/html
HTTP/3	2022	Standar	<html>...</html>

Perbedaan HTTP/1.1 vs HTTP/2: HTTP/2 menggunakan binary (lebih cepat) dan mendukung multiplexing (beberapa request dalam satu koneksi). HTTPS: HTTP dengan enkripsi SSL/TLS untuk keamanan. URL (Uniform Resource Locator): Alamat resource di web (contoh: <https://example.com/path?param=value>). DNS: Mengubah nama domain (contoh: example.com) ke alamat IP. TCP/IP: Protokol dasar untuk komunikasi jaringan. Web Server: Menyimpan data dan merespons request (contoh: Apache, Nginx).

Alur HTTP, Client (browser) mengirim HTTP Request ke Server. Server memproses request dan mengirim HTTP Response. Response berisi data yang diminta (HTML, JSON, dll.) atau pesan error.

Method	Kegunaan
GET	Meminta data dari server.
POST	Mengirim data ke server.
PUT	Mengganti seluruh data.
DELETE	Menghapus data.
PATCH	Mengubah sebagian data.

HTTP Status Code. 2xx (Sukses): 200 OK: Request berhasil. 3xx (Redirect): 301 Moved Permanently: Resource pindah permanen. 4xx (Client Error): 404 Not Found: Resource tidak ditemukan. 5xx (Server Error): 500 Internal Server Error: Kesalahan server.

HTTP Header, Content-Type: Menentukan tipe data (contoh: application/json). Authorization: Untuk autentikasi (contoh: token). Cache-Control: Mengatur caching di client.

HTTP Body, Request Body: Data dikirim client (contoh: JSON saat login). Response Body: Data dikirim server (contoh: HTML halaman web).

Cookie dan Session, Cookie: Data kecil yang disimpan di browser untuk melacak session (contoh: login). Contoh: Set-Cookie: session_id=abc123

Caching, Cache-Control: Header untuk menyimpan data sementara di client (contoh: Cache-Control: max-age=3600).

Teknologi Terkait, WebSocket: Komunikasi real-time dua arah. RESTful API: Desain API berbasis HTTP. OAuth: Protokol otorisasi.

HTML Dasar

Web vs Internet: Internet adalah jaringan komunikasi global. Web adalah kumpulan informasi yang diakses via internet menggunakan browser.

HTML (HyperText Markup Language): Struktur dasar halaman web. Menggunakan tag (contoh: <p>, <h1>) untuk mendefinisikan konten.

Struktur Dasar HTML:

```
<html><body>
```

```
<h1>Hello World</h1>
```

```
</body></html>
```

Tag Penting: <html>, <head>, <body>, <title>.

Elemen HTML, Heading (<h1> sampai <h6>): Untuk judul dan subjudul. Paragraf (<p>): Untuk teks panjang. Garis Horizontal (<hr>) dan Break Line (
). Format Teks: , <i>, , , , <ins>.

Atribut dan Style, Atribut Dasar: src (gambar), href (link), alt (teks alternatif). Inline CSS: <p style="color: red; font-size: 16px;">Teks merah</p>

List dan Tabel, List: Unordered (): Daftar dengan bullet. Ordered (): Daftar bernomor. Tabel (<table>): Tag: <tr>, <th>, <td>, <colgroup>. Atribut: colspan, rowspan.

Multimedia, Gambar (): . Video (<video>): <video controls> <source src="video.mp4" type="video/mp4"> </video>. Audio (<audio>): Mirip dengan video.

Link dan Navigasi, Link (<a>): Kunjungi Contoh. Bookmark: Lompat ke Section 1 <h2 id="section1">Section 1</h2>

Form dan Input, Form Dasar: <form action="/submit" method="post">

```
<input type="text" name="nama" placeholder="Nama"> <button type="submit">Kirim</button> </form>
```

Semantic HTML, Tag Semantic: <header>, <footer>, <article>, <section>, <nav>. Memudahkan SEO dan aksesibilitas.

Responsive Design, Viewport: <meta name="viewport" content="width=device-width, initial-scale=1.0">. Unit Responsive: %, vw, vh.

HTML Form

Digunakan untuk menerima input dari pengguna (contoh: login, registrasi). Tag dasar: <form>.

Atribut Form, action: URL tujuan pengiriman data. method: GET (data di URL) atau POST (data di request body). enctype: multipart/form-data untuk upload file.

Input Element, Text: <input type="text">. Password: <input type="password">. Checkbox/Radiobutton:

```
<input type="checkbox" name="hobby" value="coding"> Coding
```

```
<input type="radio" name="gender" value="male"> Male. File Upload: <input type="file">. Date/Time: <input type="date">, <input type="time">. Dropdown: <select> dengan <option>.
```

Label dan Button, Label: <label for="id_input"> untuk aksesibilitas. Button: <input type="submit"> untuk kirim form. <button type="button"> untuk aksi custom.

Grup Input, Fieldset: <fieldset> untuk grouping input. Legend: <legend> sebagai judul grup.

Validasi Input, Required: <input required>. Pattern: <input pattern="[0-9]{5}"> untuk validasi regex.

CSS Dasar

CSS (Cascading Style Sheets) digunakan untuk mengatur tampilan dan tata letak elemen HTML. Fungsi utama: mengubah warna, font, ukuran, layout, dan lainnya. Memastikan konsistensi tampilan di berbagai browser.

Syntax: h1 {color: font-size: 20px;}

Internal CSS: Ditulis dalam tag <style> di dalam HTML. External CSS: File terpisah (.css) dan dihubungkan dengan <link>. Inline CSS: Langsung di elemen HTML (tidak disarankan).

Simple Selector: Type (tag), Class (.), ID (#). (.red { color: red; } #judul { font-size: 24px; }

Combinator Selector: Descendant (div p), Child (div > p), Adjacent Sibling (div + p), General Sibling (div ~ p).

Attribute Selector: Memilih elemen berdasarkan atribut HTML. (a[target="_blank"] { color: blue; })

Styling Teks dan Font, Text: text-align, text-decoration, text-transform, text-shadow, Font: font-family, font-size, font-weight, font-style. Google Fonts: Import font eksternal.

Box Model, Komponen: Content, Padding, Border, Margin. Properti: width, height, padding, margin, border, border-radius.

Background dan Warna, Background: background-color, background-image, gradient. Warna: Bisa menggunakan nama, HEX, RGB, HSL.

Pseudo-class dan Pseudo-element, Pseudo-class: :hover, :focus. Pseudo-element: ::before, ::after.

Konsep Cascade dan !important, Prioritas aturan CSS: User Agent < Author < User. !important digunakan untuk memaksa prioritas tertinggi.

CSS Layout

CSS Layout digunakan untuk mengatur tata letak komponen HTML. Jangan gunakan tabel untuk layout karena sulit dikelola

Normal Flow: <h1>Judul</h1> <p>Paragraf pertama</p> <!-- Akan muncul di bawah judul -->

Display Property: block: mengambil lebar penuh (contoh: <div>). inline: hanya mengambil space yang diperlukan (contoh:). inline-block: gabungan keduanya. none: menyembunyikan elemen

Untuk layout satu dimensi (horizontal/vertikal): .container {display: flex; flex-direction: row; /* atau column */ justify-content: center; /* penataan horizontal */ align-items: center; /* penataan vertikal */ gap: 10px; /* jarak antar item */}

Untuk layout dua dimensi (baris & kolom): .container { display: grid; grid-template-columns: 1fr 2fr 1fr; /* 3 kolom */ grid-template-rows: 100px auto; gap: 15px;}

Jenis Positioning: relative: relatif terhadap posisi normal, absolute: relatif terhadap elemen parent terdekat, fixed: tetap di posisi viewport, sticky: gabungan relative dan fixed, contoh: .popup { position: fixed; top: 20px; right: 20px; z-index: 100; /* menentukan tumpukan */}

Media Queries, Untuk menyesuaikan layout di berbagai ukuran layar:

```
/* Mobile first */ .container {padding: 10px;}
```

```
/* Tablet */ @media (min-width: 768px) { .container {padding: 20px;}}
```

```
/* Desktop */ @media (min-width: 1024px) { .container {display: grid; grid-template-columns: 1fr 3fr;}}
```

JavaScript

Dibuat tahun 1995 oleh Brendan Eich di Netscape. Awalnya bernama Mocha, lalu LiveScript, akhirnya JavaScript. Bahasa pemrograman untuk membuat web lebih interaktif. Sekarang bisa digunakan di server (Node.js) dan client (browser)

Tipe Data: Number: Bilangan bulat/desimal (100, 100.100). Boolean: true/false. String: Teks ("Hello", 'World'). Array: Kumpulan data (["Eko", "Kurniawan"]). Object: Data dengan properti ({name: "Eko", age: 30}). undefined: Variabel belum diisi nilai. null: Data kosong yang disengaja. Variabel: let name = "Eko"; // bisa diubah. const age = 30; // tidak bisa diubah

Operator: Matematika: +, -, *, /, %, ** Perbandingan: ==, ===, !=, !==, >, < Logika: && (dan), || (atau), ! (bukan) Ternary: kondisi ? nilai_true : nilai_false

Tips: Gunakan const dan let bukan var, Gunakan strict mode ('use strict'), Hindari global variable, Gunakan arrow function untuk menjaga konteks this, Lakukan validasi input, Gunakan template literals untuk string kompleks, Manfaatkan destructuring untuk kode yang lebih bersih

- Percabangan: if (nilai > 80) {

```
  console.log("Lulus");
```

```
} else if (nilai > 60) {
```

```
  console.log("Remedial");
```

```
} else {
```

```
  console.log("Tidak Lulus");
```

```
} // Switch case
```

```

switch(grade) {
  case "A": console.log("Excellent"); break;
  case "B": console.log("Good"); break;
  default: console.log("Try again");}

  - Perulangan: // For loop
for(let i = 0; i < 10; i++) {
  console.log(i);
} // While loop

let i = 0;
while(i < 10) {
  console.log(i);
  i++;
} // Do-while

do {
  console.log(i);
  i++;
} while(i < 10); // For-of (array)

for(const item of array) {
  console.log(item);
} // For-in (object)

for(const key in object) {
  console.log(key, object[key]);}

  - Fungsi: // Deklarasi fungsi

function greet(name) {
  return `Hello ${name}`;
} // Arrow function

const greet = (name) => `Hello ${name}`; // Parameter default

function greet(name = "Guest") {
  return `Hello ${name}`;
} // Rest parameter

function sum(...numbers) {
  return numbers.reduce((total, num) => total + num, 0);}

Error Handling: try { // Kode yang mungkin error
} catch(error) {
  console.error(error);
} finally { // Selalu dijalankan}

```