

Tugas Besar 2 IF2211 Strategi Algoritma
Semester II tahun 2025/2026

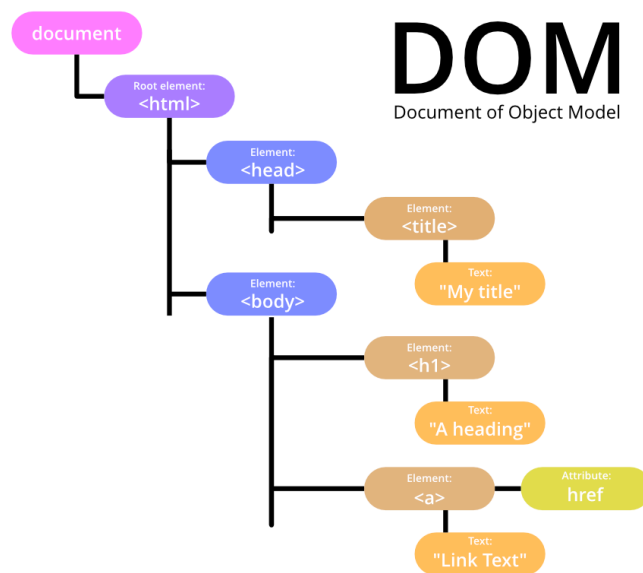
**Pemanfaatan Algoritma BFS dan DFS dalam Mekanisme Penelusuran CSS pada pohon
Document Object Model**

Batas pengumpulan : 22 April 2026, 23:59

Arsip pengumpulan :

- Source program yang dapat dijalankan disertai README
- Laporan (soft copy)

Deskripsi Tugas



Gambar 1. Struktur Pohon DOM

<https://www.geeksforgeeks.org/java/what-is-document-object-in-java-dom/>

Document Object Model (DOM) merupakan representasi terstruktur dari dokumen HTML dalam bentuk *tree* yang memungkinkan setiap elemen pada halaman web diakses, dimodifikasi, dan dimanipulasi oleh program. Struktur ini merepresentasikan hubungan antar elemen HTML seperti parent, child, dan sibling sehingga memudahkan pengolahan dokumen secara terprogram.

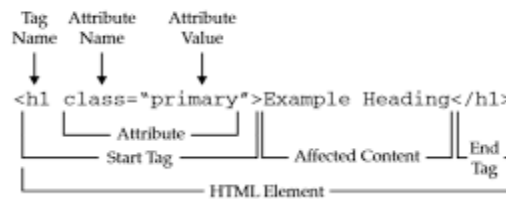
Struktur sebuah file HTML pada umumnya meliputi elemen root `<html>` yang berisi dua bagian utama yaitu `<head>` dan `<body>`. Bagian `<head>` berisi metadata seperti judul halaman, link stylesheet, dan script, sedangkan `<body>` berisi konten utama halaman seperti teks, gambar, tombol, dan elemen HTML lainnya yang ditampilkan kepada pengguna.

Sebuah website menampilkan sebuah DOM melalui sebuah file bernama Cascading Style Sheets (CSS) yang bertujuan mendeklarasikan visualisasi elemen-elemen pada pohon. Deklarasi visualisasi ditempelkan pada elemen-elemen yang berkaitan melalui CSS Selector.

Pada Tugas Besar 2 Strategi Algoritma ini, mahasiswa diminta untuk menelusuri dan mencari elemen pada struktur DOM berdasarkan CSS Selector tertentu dengan menggunakan algoritma penelusuran graf yaitu Breadth First Search (BFS) dan Depth First Search (DFS). Algoritma tersebut digunakan untuk menjelajahi struktur pohon DOM guna menemukan elemen yang sesuai dengan selector yang diberikan.

Komponen-komponen penting yang terdapat pada tugas besar ini antara lain:

1. Tags



Gambar 2. Struktur Syntax HTML

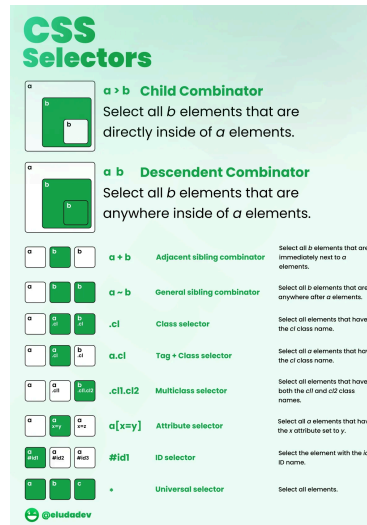
<https://tutorial.techaltum.com/htmlTags.html>

Tag pada HTML merupakan penanda (markup) yang digunakan untuk mendefinisikan struktur dan jenis elemen dalam sebuah dokumen HTML. Tag memberi tahu browser bagaimana suatu bagian konten harus ditafsirkan dan ditampilkan pada halaman web.

Secara umum, tag HTML ditulis menggunakan tanda kurung sudut (`<>`). Sebagian besar elemen HTML memiliki pasangan tag pembuka dan tag penutup. Tag pembuka menandai awal suatu elemen, sedangkan tag penutup menandai akhir elemen tersebut. Tag penutup ditulis dengan menambahkan tanda garis miring / sebelum nama tag. Terdapat beberapa jenis tag HTML yang *self-closing*.

Dalam struktur DOM, setiap tag HTML direpresentasikan sebagai node pada pohon DOM. Hubungan antar tag membentuk struktur hierarki seperti parent, child, dan sibling. Struktur inilah yang memungkinkan dokumen HTML diproses sebagai sebuah pohon ketika dilakukan penelusuran menggunakan algoritma seperti BFS atau DFS.

1. CSS Selector



Gambar 3. Visualisasi CSS Selector

https://www.reddit.com/r/webdev/comments/ur6v5m/css_selectors_visually_explained/

CSS Selector adalah mekanisme pada CSS untuk memilih elemen HTML tertentu pada struktur DOM agar dapat diberikan aturan style. Dengan selector, kita dapat menentukan elemen mana yang akan dikenai aturan visual seperti warna, ukuran, atau tata letak. Beberapa selector dasar yang umum digunakan adalah sebagai berikut.

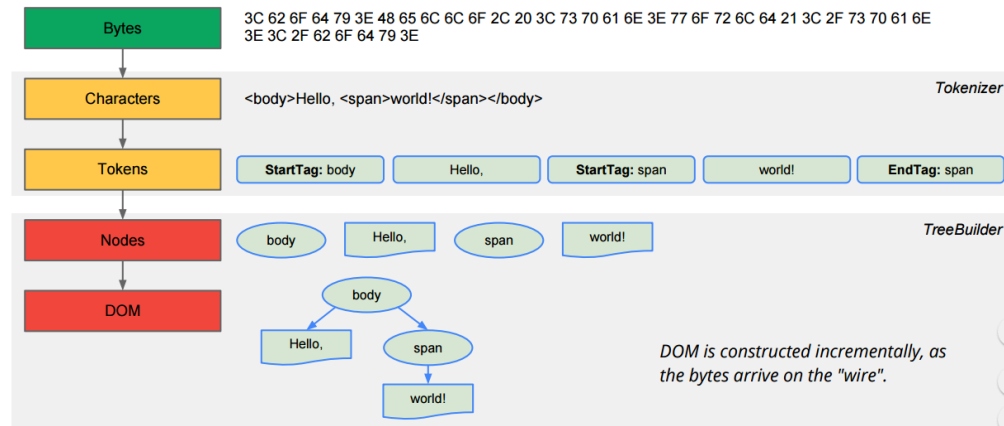
- Tag Selector
Memilih elemen berdasarkan nama tag HTML. Contoh: `p` memilih semua elemen `<p>`.
- Class Selector
Memilih elemen yang memiliki atribut class tertentu, ditulis dengan tanda titik (`.`). Contoh: `.box`
- ID Selector
Memilih elemen dengan atribut id tertentu, ditulis dengan tanda pagar (`#`).
Contoh: `#header`
- Universal Selector
Memilih semua elemen HTML.
Contoh: `*`

Combinator merupakan cara untuk menggabungkan beberapa jenis selector dalam penentuan elemen yang akan terpengaruh, dan terdiri dari:

- Child Combinator (`>`)

- Descendent Combinator (“ “)
- Adjacent Sibling Combinator (+)
- General Sibling Combinator (~)

2. HTML Parsing



Gambar 4. Proses Parsing Pohon HTML

<https://stackoverflow.com/questions/3150293/how-does-a-parser-for-example-html-work>

HTML Parsing adalah proses membaca dokumen HTML dan mengubahnya menjadi struktur data pohon (tree) yang merepresentasikan hubungan antar elemen HTML. Pada proses ini, parser memproses dokumen HTML secara berurutan, mengenali tag pembuka dan tag penutup, kemudian menyusunnya ke dalam struktur hierarki.

Setiap tag HTML direpresentasikan sebagai node pada pohon. Tag yang berada di dalam tag lain akan menjadi child node, sedangkan tag yang membungkusnya menjadi parent node. Node paling atas pada struktur ini disebut root, yang umumnya merupakan elemen <html>. Dari node root tersebut, elemen-elemen lain seperti <head> dan <body> akan menjadi cabang yang membentuk struktur pohon DOM.

Hasil dari proses parsing adalah DOM Tree, yaitu representasi pohon dari dokumen HTML yang memungkinkan elemen-elemen pada halaman web ditelusuri, diakses, dan dimanipulasi secara terstruktur. Struktur pohon ini kemudian dapat digunakan dalam proses penelusuran elemen, misalnya dengan menggunakan algoritma Breadth First Search (BFS) atau Depth First Search (DFS).

Spesifikasi Wajib

Pada tugas ini, mahasiswa diminta untuk membuat aplikasi traversal pohon HTML (DOM Tree) yang menggunakan algoritma Breadth First Search (BFS) dan Depth First Search (DFS) untuk melakukan pencarian elemen berdasarkan CSS selector pada struktur DOM.

- Buatlah sebuah aplikasi berbasis web yang terdiri dari frontend bebas dan backend dalam bahasa **Go, C#, Rust, atau Typescript** yang mengimplementasikan **algoritma BFS dan DFS** untuk melakukan penelusuran CSS pada pohon DOM.
- Aplikasi dapat melakukan scraping HTML dari sebuah website melalui URL yang diberikan oleh pengguna
- Secara umum, berikut adalah input yang diharapkan
 - URL website atau memasuki teks HTML sendiri
 - Pilihan algoritma traversal (BFS / DFS)
 - CSS selector
 - Jumlah hasil yang ingin ditampilkan:
 - Top n kemunculan
 - Semua kemunculan
- HTML yang diperoleh kemudian diparsing menjadi struktur data pohon (DOM Tree).
- Output yang diharapkan adalah sebagai berikut:
 - Aplikasi dapat menampilkan visualisasi struktur pohon DOM, disertai dengan keterangan kedalaman maksimum tree
 - Aplikasi dapat menampilkan highlight pada jalur yang di-traversal oleh algoritma penelusuran yang dipilih untuk menandakan elemen-elemen yang terpengaruh.
 - Aplikasi dapat menampilkan waktu pencarian serta banyak node yang dikunjungi.
 - Setelah melakukan penelusuran, aplikasi menyimpan sebuah *traversal log* yang memiliki informasi tahapan penelusuran yang dilakukan.
- Repository bagian frontend dan backend boleh digabung/dipisah.
- Setiap kelompok harap mengisi nama kelompok dan anggotanya pada link [berikut](#), dengan maksimal tanggal **Minggu, 5 April 2026 23.59**. **Nama kelompok dilarang mengandung unsur SARA, penghinaan, provokasi, atau hal-hal lain yang bertentangan dengan norma hukum.**
- Perlu diperhatikan bahwa kelompok tubes untuk mata kuliah ini tidak boleh sama. Jadi jika berkelompok dengan X pada tubes ini, maka pada tubes selanjutnya anda tidak diperkenankan berkelompok dengan X.

Spesifikasi Bonus (Maks 20 poin)

1. (6 poin) [**Video**] Membuat video tentang aplikasi BFS dan DFS pada penelusuran pohon DOM kemudian mengunggahnya di Youtube. Video dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Untuk contoh video tubes stima tahun-tahun sebelumnya dapat dilihat di Youtube dengan kata kunci “Tubes Stima”, “strategi algoritma”, “Tugas besar stima”, dll. Semakin menarik video, maka semakin banyak poin yang diberikan.
2. (4 poin) [**Deploy**] Aplikasi web yang telah dibuat perlu di deploy menggunakan [Microsoft Azure](#) Virtual Machine (VM) sehingga aplikasi dapat diakses secara publik melalui internet. Sebagai mahasiswa, terdapat kredit gratis sebesar \$100 dari Microsoft untuk dapat digunakan hanya dengan menggunakan email kampus ITB. Karena layanan ini menggunakan sistem pembayaran berbasis penggunaan (pay-as-you-go), penggunaan Virtual Machine akan mengurangi kredit yang dimiliki. Oleh karena itu, disarankan untuk mematikan (stop/deallocate) Virtual Machine ketika tidak digunakan agar kredit Azure tidak cepat habis.
3. (2 Poin) [**Docker**] Aplikasi dijalankan menggunakan Docker baik untuk frontend maupun backend.
4. (6 Poin) [**Animasi**] Menampilkan animasi penelusuran pohon DOM untuk segala pencarian.
5. (3 Poin) [**Multithreading**] Mengimplementasikan multithreading pada algoritma BFS dan/atau DFS untuk melakukan traversal pohon DOM secara paralel. Proses traversal dibagi ke beberapa thread/worker sehingga beberapa node dapat diproses secara bersamaan.
6. (3 Poin) [**LCA Binary Lifting**]
Mengimplementasikan algoritma Lowest Common Ancestor (LCA) dengan teknik Binary Lifting pada pohon DOM yang dihasilkan dari proses parsing HTML. LCA digunakan untuk menentukan node leluhur terendah yang sama dari dua node dalam sebuah pohon. Dalam konteks DOM, fitur ini dapat digunakan untuk mengetahui elemen HTML terdekat yang menjadi parent bersama dari dua elemen tertentu. Penjelasan lebih lanjut dapat dilihat pada link [berikut](#).

Petunjuk Pengerjaan

- Program disimpan dalam repository dengan nama **Tubes2_NamaKelompok** (**Tubes2_NamaKelompok_FE/BE** jika repository dipisah) dengan nama kelompok sesuai pada sheet kelompok tertera.
- Buatalah README untuk tata cara penggunaan yang minimal berisi:
 - ❖ Penjelasan singkat algoritma BFS dan DFS yang diimplementasikan
 - ❖ Requirement program dan instalasi tertentu bila ada
 - ❖ Command atau langkah-langkah dalam meng-compile atau build program

- ❖ Author (identitas pembuat)
- Sangat disarankan untuk menggunakan [semantic commit](#). Buatlah release dengan format v1.x dengan x adalah nomor versi dimulai dari revisi ke 0. Contoh : v1.0 untuk release pertama, v1.1 release kedua dan selanjutnya.
- Pastikan untuk membuat repository bersifat Public paling lambat H+1 deadline (1 hari setelah deadline). Sebelum deadline repository harus bersifat Private.
- Laporan dibuat dengan format nama Tubes2_[Nama kelompok].pdf
- Laporan dan tautan rilis repository dikumpulkan 22 April 2026 pada alamat Google Form [berikut](#) paling lambat pukul 23.59 WIB:
PERINGATAN: Keterlambatan akan mengurangi nilai sebanyak 1 poin untuk setiap menit keterlambatan.
- Adapun pertanyaan terkait tugas besar ini bisa disampaikan melalui QnA [berikut](#).

Ketentuan Laporan

- **Cover:** Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini)
- **Bab 2:** Landasan Teori.
- **Bab 3:** Aplikasi BFS dan DFS
- **Bab 4:** Implementasi dan pengujian.
- **Bab 5:** Kesimpulan dan saran.
- **Lampiran:** Tautan repository GitHub dan video (jika membuat)
- **Daftar Pustaka**

Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan mengikuti format pada section “Isi laporan” dengan baik dan benar.
3. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).
4. Pastikan memasukkan pernyataan tersebut pada laporan, yaitu **Pernyataan tidak melakukan kecurangan yang di tandatangi** dengan format sebagai berikut:

Tugas ini disusun sepenuhnya tanpa bantuan kecerdasan buatan (*Generative AI*), melainkan hasil pemikiran dan analisis mandiri.

	[Tanda tangan mahasiswa] [Nama mahasiswa]
	[Tanda tangan mahasiswa] [Nama mahasiswa]
	[Tanda tangan mahasiswa] [Nama mahasiswa]

Penilaian

1. Bagian 1: Desain dan analisis laporan (45%)

- Pemahaman tugas besar (5%)
- Analisis web scraping dan pemodelan pohon DOM (5%)
- Perancangan algoritma traversal BFS dan DFS (25%)
- Analisis efisiensi dan perbandingan BFS vs DFS (10%)

2. Bagian 2: Implementasi program dan demo (55%)

- Kesesuaian strategi BFS dan DFS yang dituliskan dengan implementasi dan saat demo (15%)
- Fungsionalitas keseluruhan pada aplikasi berjalan sesuai spesifikasi (20%)
- Demo dan pemahaman program (20%)

3. Bagian 3: Komponen bonus

- Membuat video (6 poin)
- Deploy aplikasi menggunakan VM Azure (4 poin)
- Menggunakan Docker (2 poin)
- Implementasi animasi penelusuran (6 poin)
- Analisis dan Implementasi multithreading (3 poin)
- Analisis dan Implementasi LCA Binary Lifting (3 poin)

Perhatian

- Dilarang keras copy paste program dari internet, AI, repository lain, ataupun program milik teman. Program yang dikerjakan menggunakan agent (Opus, Sonnet, dan lain-lain) akan sangat terlihat dan akan memiliki kemiripan dengan mahasiswa yang menggunakan hal serupa. Segala bentuk kecurangan akan diberikan sanksi berat yaitu nilai tugas menjadi nol. Kami tidak ingin hal itu terjadi dan kami ingin Anda untuk dapat memikirkan solusi algoritma dari hasil pemikiran Anda sendiri.
- Pastikan program **dapat dikompilasi setidaknya** pada windows dan linux.
- Apabila program **tidak dapat dijalankan** maka tidak akan dinilai oleh asisten.

- Tambahkan tabel berikut yang diisi checklist (✓) pada bagian lampiran laporan dan readme Anda.

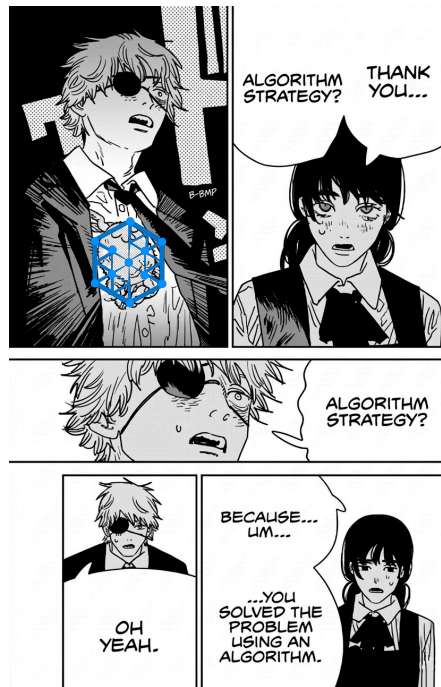
No	Poin	Ya	Tidak
1	Aplikasi berhasil di kompilasi tanpa kesalahan		
2	Aplikasi berhasil dijalankan		
3	Aplikasi dapat menerima input URL web, pilihan algoritma, CSS selector, dan jumlah hasil		
4	Aplikasi dapat melakukan scraping terhadap web pada input		
5	Aplikasi dapat menampilkan visualisasi pohon DOM		
6	Aplikasi dapat menelusuri pohon DOM dan menampilkan hasil penelusuran		
7	Aplikasi dapat menandai jalur tempuh oleh algoritma		
8	Aplikasi dapat menyimpan jalur yang ditempuh algoritma dalam traversal log		
9	[Bonus] Membuat video		
10	[Bonus] Deploy aplikasi		
11	[Bonus] Implementasi animasi pada penelusuran pohon		
12	[Bonus] Implementasi multithreading		
13	[Bonus] Implementasi LCA Binary Lifting		

Referensi

- **CSS Selector Reference**
<https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Selectors>
- **Client-Server Architecture**
https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Client-Server_overview
- **Rust Documentation**
[Rust Documentation](https://rust-lang.org/)
- **C# Documentation**

- **TypeScript Documentation**
- **Go Documentation**

“CP-Algorithms mentioned!!!???? hehe i like cp”



--- Fariz---

“tung tung tung tubes”

--- Hakim---

“bonus nya seru seru yah”

--- Radhi---

--- Lukas---

“Selamat sudah menempuh ½ perjalanan Tugas Stima ;b”

--- Carlo---

--- Orvin---

--- Rizal---

“”

--- Theo---