

# Tugas Pemrograman

## IF2124 Teori Bahasa Formal dan Otomata

### HTML *Checker* dengan Pushdown Automata (PDA)

Tanggal Rilis : Jumat, 3 November 2023

Tanggal Pengumpulan : Minggu, 26 November 2023 pukul 23.59 WIB

Revisi 1 - Tag, void elemen, dan nesting - 9 November 2023

Revisi 2 - nesting pada form 23 November 2023

Revisi 3 - Pengunduran Deadline

Link QnA:

<https://docs.google.com/spreadsheets/d/1g3IBzFkH1edkMHGrsCAsjYFQD8APkmrhBAUQj7sXBk/edit?usp=sharing>



### Deskripsi Permasalahan

HTML (Hypertext Markup Language) adalah bahasa markup yang digunakan untuk membuat struktur dan tampilan konten web. HTML adalah salah satu bahasa utama yang digunakan dalam pengembangan web dan digunakan untuk menggambarkan bagaimana

elemen-elemen konten, seperti teks, gambar, tautan, dan media, akan ditampilkan di browser web. Setiap dokumen HTML dimulai dengan elemen `<html>`, lalu diikuti dengan `<head>` (untuk metadata dan tautan ke file eksternal) dan `<body>` (untuk konten yang akan ditampilkan)

HTML menggunakan elemen-elemen (*tags*) untuk mengelompokkan dan mengatur konten. Contohnya, `<p>` digunakan untuk paragraf teks, `<h1>` hingga `<h6>` digunakan untuk judul, `<a>` untuk tautan, `<img>` untuk gambar, dan sebagainya. Elemen HTML sering memiliki atribut yang memberikan informasi tambahan tentang elemen tersebut. Contohnya adalah atribut `src` untuk gambar, `href` untuk tautan, dan `class` untuk memberikan elemen kelas CSS.

Sama seperti bahasa pada umumnya, HTML juga memiliki sintaks tersendiri dalam penulisannya yang dapat menimbulkan error jika tidak dipenuhi. Meskipun web browser modern seperti Chrome dan Firefox cenderung tidak menghiraukan error pada HTML memastikan bahwa HTML benar dan terbentuk dengan baik masih penting untuk beberapa alasan seperti *Search Engine Optimization (SEO)*, aksesibilitas, *maintenance* yang lebih baik, kecepatan render, dan profesionalisme.

Dibutuhkan sebuah program pendeteksi *error* untuk HTML. Oleh sebab itu, implementasikan sebuah program yang dapat memeriksa kebenaran HTML dari segi nama *tag* yang digunakan serta *attribute* yang dimilikinya. Pada tugas pemrograman ini, gunakanlah konsep Pushdown Automata (PDA) dalam mencapai hal tersebut yang diimplementasikan dalam bahasa **Python**.

Struktur HTML yang diharapkan (Perhatikan struktur dan urutan pada `html`, `head`, `title`, `body`).

```
<html>
<head>
<title> </title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>
<!--... Elemen-elemen lain ... -->

</body>
</html>
```

Html selalu menjadi elemen terluar. Head selalu mendahului body. Title selalu di dalam head. Elemen-elemen lain berada di dalam body.

Mengingat banyaknya jenis *tags* beserta *attributes* yang tersedia pada HTML, *scope* tugas pemrograman akan dibatasi untuk memudahkan mahasiswa dalam pengerjaan. Batasan-batasan dari *tags* dan *attributes* yang diperiksa dapat dilihat di bawah ini.

Tag	Attribute Tambahan	Void Element	Catatan Tambahan
html			Tag html wajib ada, Dokumen harus diawali tag html
head			Tag head wajib ada, Tag head harus berada di dalam tag html dan diatas tag body
body			Tag body wajib ada, Tag body harus berada di dalam tag html dan dibawah tag head. Semua elemen yang disebutkan setelah ini kecuali title harus berada di dalam body, namun tidak semua tag wajib ada di dalam body.
title			<del>Harus di dalam head</del> hanya boleh berada dalam head
link	Rel, href	V	Atribut rel wajib ada, dapat berada dalam head
script	src		dapat berada dalam head
h1,h2, h3,h4, h5,h6			
p			
br		V	
em			
b			
abbr			
strong			
small			
hr		V	
div			
a	href		
img	src, alt	V	Atribut src wajib ada
button	type		Nilai attribute type dibatasi pada submit, reset, button

form	action, method		Nilai method harus dibatasi pada GET atau POST
input	type	V	Nilai attribute type harus dibatasi pada text, password, email, number, atau checkbox
table			Bentuk tabel tidak perlu diperhatikan (tidak apa-apa jika baris 1 terdiri dari 1 kolom sedangkan baris 2 terdiri dari 2 kolom)
tr			Tag tr harus berada dalam tag table
td			Tag td harus berada dalam tag table
th			Tag th harus berada dalam tag table

Catatan tambahan:

1. **Seluruh tag dapat memiliki *global attribute*.** *Global attribute* yang akan digunakan pada tugas kali ini dibatasi pada atribut **id, class, dan style**.
2. Selain dari yang dibatasi pada Catatan Tambahan, nilai atribut selalu valid selama di dalam kutip dua (""). Misal pada atribut **id**, meskipun *whitespace* sebenarnya tidak diperbolehkan, pada scope tubes ini tidak diperhatikan.
3. Void Element merupakan elemen / singleton tag adalah elemen yang tidak memerlukan terdapat tag penutup untuk menjadi valid. Misal:

****

4. Jangan lupa untuk **menghandle komentar** (`<!-- ini komentar -->`)
5. Terkait nesting pada tag, asisten hanya akan mengecek nesting pada:
  - a. Html, head, body
  - b. Tag div
  - c. ~~Html formatting element pada konten~~
  - d. Tabel-tabelan (table, tr, td, th)
  - e. Form
6. Selain tag-tag yang disebutkan di nomor 5, seluruh tag yang memiliki konten juga dapat melakukan nesting apabila isi nesting merupakan HTML formatting elemen. Berikut merupakan contoh yang dimaksud:

**<tag0>** merupakan tag yang disebutkan dapat melakukan nesting di spek tubes

**<tag1>** merupakan tag yang tidak disebutkan dapat melakukan nesting di spek tubes

**<tag2>** merupakan html formatting elemen

<tag3> bukan merupakan html formatting elemen

Contoh 1:

```
<tag1>Hello<tag2>World</tag2><tag1>
```

Contoh 2:

```
<tag1>Hello<tag3>World</tag3><tag1>
```

Contoh 1 akan dikeluarkan oleh asisten. Contoh 2 **tidak akan dikeluarkan oleh asisten** karena nesting dilakukan oleh tag yang tidak di specify di spek pada no 5.

Catatan tambahan:

```
<tag0><tag3>Hello</tag3><tag3>World</tag3></tag0>
```

Merupakan nesting yang valid

## Pengerjaan

Pengerjaan dilakukan secara berkelompok, setiap kelompok terdiri dari **3 mahasiswa dari kelas yang sama**, mahasiswa diperkenankan membentuk kelompok masing masing dan mengisi data kelompok di sheets berikut :

<https://docs.google.com/spreadsheets/d/10BsoEnc5gNOecG3WAjynOeTsL86Llrup34aQoojYOhU/edit?usp=sharing>

Deadline pengisian data kelompok adalah **Kamis, 9 November 2023, 23:59 WIB**

## Deliverables

1. Pengerjaan dalam Github dengan repository private berisi kode program. Buatlah sebuah rilis dengan **nama rilis "v-x"** pada repositori sebelum mengumpulkan, dengan x adalah nomor rilis **dimulai dari 1**. Apabila ada revisi, silakan buat rilis baru dengan meng-*increment* nilai x. Program yang akan dinilai berasal dari **rilis terakhir sebelum deadline**.
2. Gambar state-state pada PDA. Gunakan tools diagram maker seperti [diagrams.net](https://diagrams.net). Tools yang dipakai bebas asalkan online dan asisten dapat mengaksesnya.
3. Laporan Tugas, isi laporan akan dijelaskan di bawah.

## Spesifikasi Program

1. Tugas dikerjakan menggunakan bahasa pemrograman **Python**.
2. Program menggunakan antarmuka berbasis CLI (Command Line Interface).
3. Mahasiswa akan menggunakan konsep PDA.
4. Program akan menerima suatu teks file yang merupakan kode HTML, lalu program akan memanfaatkan PDA untuk mengevaluasi kebenaran syntax dari kode yang dijadikan input tadi.
5. Definisi PDA menggunakan sebuah text file eksternal (tidak dihardcode di program). File tersebut berisi hal-hal yang dibutuhkan untuk mendefinisikan PDA seperti *states*, *productions*, *starting state*, dll. Berikut adalah contoh definisi PDA

```
Q P F # total states
a # input word symbols
Z Y # stack symbols
Q # starting state
Z # starting stack
F # accepting states
F # E - accepts with empty stack or F - accepts with accepting
state
Q a Z Q YZ # list of productions (current state, read from word,
take from stack, next state, add to stack)
Q a Y Q YY
Q e Z P Z
Q e Y P Y
P a Z P e
P a Y P e
P e Z F e
```

### Catatan:

- Pada contoh di atas, “e” adalah epsilon.
  - Format yang kalian implementasikan / gunakan tidak harus mengikuti contoh di atas.
6. Dengan begitu, program harus dapat mengkonversikan definisi PDA ke dalam program.
  7. Alur umum dari program yang akan dibangun adalah sebagai berikut.
    - a. Program python menerima dua buah argumen pada saat eksekusi dengan ketentuan sebagai berikut:
      - i. Argumen pertama adalah path menuju file definisi PDA
      - ii. Argumen kedua adalah path menuju file HTML yang ingin diperiksa
    - b. Melakukan evaluasi sintaks dengan PDA.
    - c. Memberikan keluaran hasil evaluasi program antara “Accepted” jika input diterima atau “Syntax Error” jika input tidak diterima

- d. **(BONUS)** Memberi tahu letak dan detail kesalahan syntax jika ada
8. Buatlah Laporan terkait pengerjaan tugas kalian, laporan wajib memiliki :
  - a. Teori dasar PDA dan pengaplikasiannya pada sintaksis HTML.
  - b. Hasil PDA yang dibuat
  - c. Implementasi dan Pengujian
    - i. Spesifikasi teknis program, termasuk di dalamnya struktur data, fungsi dan prosedur (header fungsi dan prosedur saja, tidak perlu source code), antarmuka, dan lain-lain yang dianggap perlu.
    - ii. Screenshot yang memperlihatkan contoh dari berbagai kasus yang muncul (buatlah minimal 3 kemungkinan tipe kasus uji yang mungkin) dan analisis hasilnya
  - d. Link ke repository github.
  - e. Link ke diagram state.
  - f. Pembagian Tugas.

## Pengumpulan

- Waktu pengerjaan tugas adalah **23 Hari**, batas akhir pengumpulan ialah tanggal **26 November** pukul **23.59 WIB**.
- Pengumpulan dilakukan dengan melalui form <https://forms.gle/N3gV74TEaoPyeyhG7>, buatlah repository github yang PRIVATE untuk pengerjaan kalian, lalu set ke PUBLIC paling lambat tepat 24 jam setelah deadline.
- Kecurangan dan plagiarisme akan mengakibatkan sanksi ke semua peserta terkait.
- Demo akan dilakukan setelah pengumpulan. Metode dan peraturan demo akan diberitahukan nanti.

## Contoh Input dan Output Program

Diberikan 2 file HTML

### 1. inputAcc.html

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
```

```
</html>
```

## 2. inputReject.html

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!<h1>
    <p>This is a simple webpage.</p>
  </body>
</html>
```

## 3. Contoh interaksi program python untuk file "inputAcc.html" pada nomor 1

```
> python main.py pda.txt "inputAcc.html"
Accepted
```

## 4. Contoh interaksi program python untuk file "inputReject.html" pada nomor 2

```
> python main.py pda.txt "inputReject.html"
Syntax Error

(Output tambahan jika mengerjakan bonus)
Terjadi kesalahan ekspresi pada line 6 : "<h1>Hello, World!<h1>"
```

# Contoh Test Case

## 1. Rejected: head muncul setelah body

```
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
  <head>
    <title>Simple Webpage</title>
  </head>
</html>
```

## 2. Rejected: tag pertama bukan html

```
<h1>f>
  <head>
    <title>Simple Webpage</title>
```



```
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is a simple webpage.</p>
</body>
</html>
```

### 3. Rejected: tidak punya head

```
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
</html>
```

### 4. Accepted

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    
    <p>This is a <em>simple</em> webpage.</p>

    <div id="footer" class="footer"> This is the end of the page </div>
  </body>
</html>
```

### 5. Accepted

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <hr>
    
    <p>This is a <em>simple</em> webpage.</p>

    <!-- Custom element -->
    <div id="footer" class="footer"> This is the end of the page </div>
  </body>
</html>
```

## 6. Rejected: Atribut wajib src pada img tidak ada

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <img alt="Welcome Banner">
    <p>This is a <em>simple</em> webpage.</p>

    <!-- Custom element -->
    <div id="footer" class="footer"> This is the end of the page </div>
  </body>
</html>
```

## 7. Accepted

```
<html>
<head>
  <title>Simple Webpage</title>

</head>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php" method="POST">
  <h5 class="label">First name:</h5><br>
  <input type="text" id="fname"><br>
  <h5 class="label">Last name:</h5><br>
  <input type="text" id="lname"><br><br>
  <button type="submit">Submit</button>
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>

</body>
</html>
```

## 8. Rejected: tidak termasuk value yang diperbolehkan (POST, GET)

```
<html>
<head>
  <title>Simple Webpage</title>

</head>
<body>
```

```

<h2>HTML Forms</h2>

<form action="/action_page.php" method="TEBAK">
  <div id="label">First name:</div><br>
  <input type="text" id="fname"><br>
  <div id="label">Last name:</div><br>
  <input type="text" id="lname"><br><br>
  <button type="submit">Submit</button>
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>

</body>
</html>

```

## 9. Accepted

```

<html>
<head>
  <title>Simple Webpage</title>
  <script>
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
  </script>
</head>
<body>

<h1>The script element</h1>

<p id="demo"></p>

</body>
</html>

```

## 10. Rejected: <p> bukan void element, wajib ditutup

```

<html>
<head>
  <title>Simple Webpage</title>
  <script>
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
  </script>
</head>
<body>

<h1>The script element</h1>

<p id="demo">

</body>
</html>

```

## 11. Accepted

```
<html>
  <head>
    <title>Simple Webpage</title>
    <script>
      document.getElementById("demo").innerHTML = "Hello JavaScript!";
    </script>
  </head>
  <body>
    <h1>The script element</h1>
    <a>Not going anywhere</a><br>
    <a href="https://www.google.co.id/">Might send you somewhere</a>

    <p id="demo"></p>
  </body>
</html>
```