

# LAPORAN TUGAS BESAR

IF2124 TEORI BAHASA FORMAL DAN AUTOMATA

Kelompok JOJIK



Disusun Oleh :

Jonathan Emmanuel Saragih (13522121)

Satriadhikara Panji Yudisthira (13522125)

Mohammad Andhika Fadillah (13522128)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA - KOMPUTASI

INSTITUT TEKNOLOGI BANDUNG

## **DAFTAR ISI**

<b>DAFTAR ISI</b>	<b>2</b>
<b>BAB 1 TEORI DASAR</b>	<b>3</b>
<b>BAB 2 HASIL PDA</b>	<b>4</b>
<b>BAB 3 IMPLEMENTASI &amp; PENGUJIAN</b>	<b>5</b>
<b>PEMBAGIAN TUGAS</b>	<b>10</b>
<b>DAFTAR PUSTAKA</b>	<b>11</b>

# **BAB 1**

## **TEORI DASAR**

Dalam dunia pemrograman web, HTML (HyperText Markup Language) memegang peranan penting sebagai bahasa dasar yang membentuk struktur halaman web. HTML terdiri dari berbagai elemen yang ditandai dengan tag, di mana setiap tag memiliki fungsi spesifik dalam menentukan struktur dan tampilan konten. Ketepatan sintaks dalam HTML tidak hanya vital untuk menampilkan konten dengan benar, tetapi juga berpengaruh terhadap aksesibilitas dan efisiensi SEO (Search Engine Optimization) suatu website. Oleh karena itu, memeriksa kebenaran struktur HTML menjadi aspek krusial dalam pengembangan web.

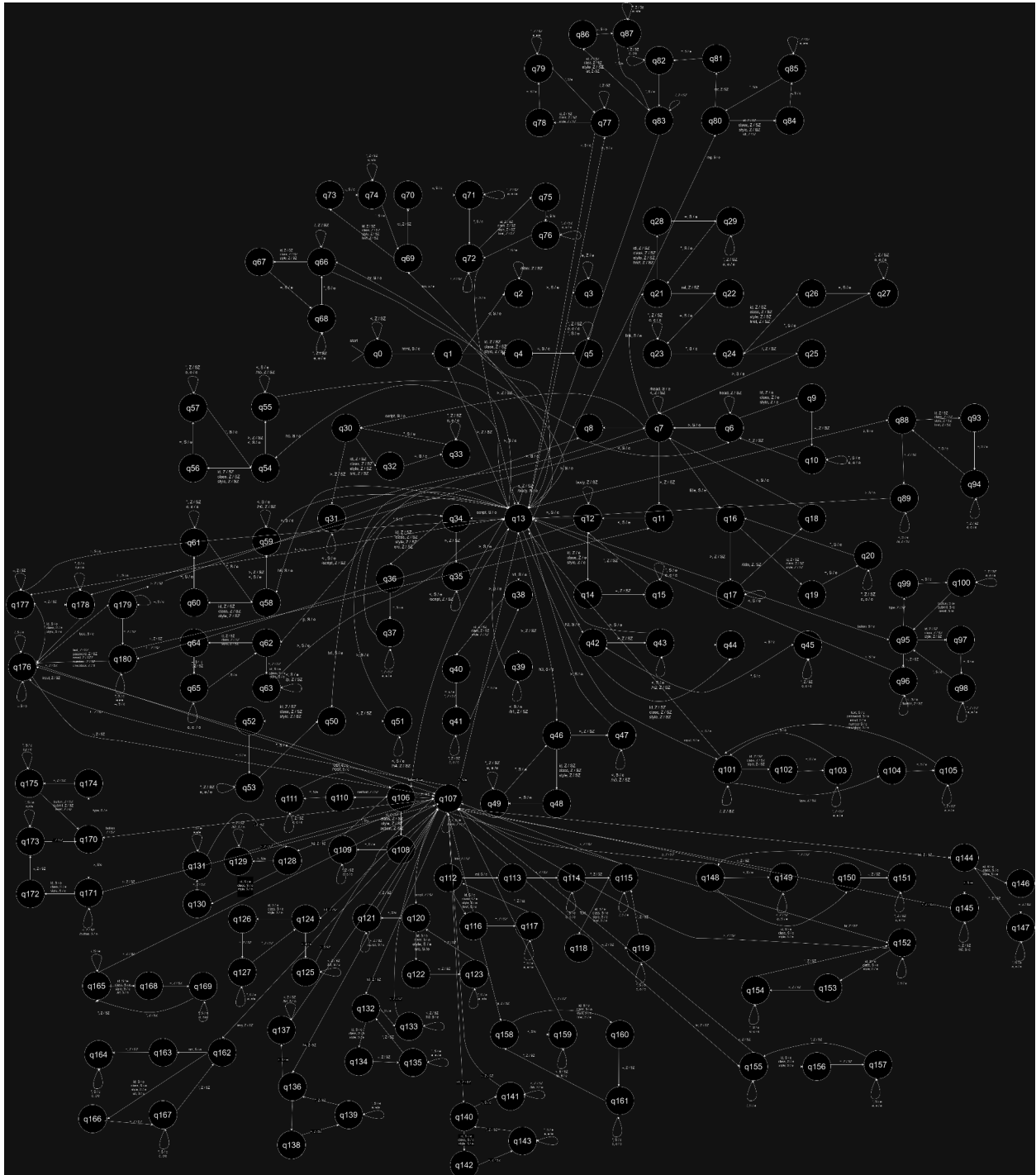
Dalam konteks ini, Automata Pushdown (PDA) menjadi alat yang berguna. PDA adalah model komputasi teoretis yang mirip dengan automata berhingga, namun dilengkapi dengan sebuah "stack" yang memungkinkan penyimpanan sejumlah informasi tidak terbatas. Konsep PDA sangat relevan dalam menangani struktur beraneka ragam seperti HTML. Komponen utama dari PDA meliputi sekumpulan state, alfabet input, alfabet stack, fungsi transisi, state awal, dan state penerima. Dalam aplikasinya untuk pemeriksaan HTML, PDA menggunakan stack untuk melacak pembukaan dan penutupan tag, memastikan nesting tag yang benar serta urutan elemen-elemen HTML sesuai aturan.

Implementasi PDA dalam pemrograman, khususnya menggunakan bahasa Python, melibatkan penciptaan model PDA yang dapat membaca dan memproses file HTML untuk mengecek kebenaran sintaksnya. Python, dengan sintaksnya yang mudah dibaca dan dukungan luas terhadap operasi berorientasi teks, menjadi pilihan yang tepat untuk mengimplementasikan konsep PDA ini. Dalam implementasinya, kode Python akan memanfaatkan definisi eksternal PDA dan memproses file HTML untuk memeriksa kesesuaian tag dan atributnya sesuai dengan aturan sintaks HTML.

Pemahaman mendalam tentang struktur HTML, prinsip kerja PDA, dan kemampuan implementasi menggunakan Python, menjadi kunci dalam menyelesaikan tugas ini dengan efektif. Melalui penerapan konsep PDA dalam pemrograman Python, kita dapat menciptakan sebuah alat yang mampu memeriksa keakuratan sintaks HTML, yang tidak hanya meningkatkan kualitas kode, tetapi juga memastikan kepatuhan terhadap standar web yang telah ditetapkan.

## BAB 2

### HASIL PDA



## BAB 3

### IMPLEMENTASI & PENGUJIAN

#### 3.1 Struktur data

1. **PDA**: *Class* yang mewakili Pushdown Automaton (PDA). Ini berisi atribut sebagai berikut:

- *states*: Daftar *states* di PDA.
- *input\_symbols*: Daftar simbol input yang dapat diproses oleh PDA.
- *stack\_symbols*: Daftar simbol yang ada di *stack*.
- *initial\_state*: *States* awal PDA.
- *initial\_stack\_symbol*: Simbol awal pada *stack*.
- *final\_states*: Daftar *states* akhir di PDA.
- *transition\_function*: Daftar fungsi transisi PDA.

#### 3.2 Fungsi dan Prosedur

1. `PDA.__init__(self, States, input_symbols, stack_symbols, Initial_state, Initial_stack_symbol, Final_states, Transition_function)`: Menginisialisasi instance baru dari *class* PDA.
2. `PDA.process(self, string)`: Memproses string menggunakan PDA dan mengembalikan apakah string tersebut diterima atau tidak.
3. `print_fancy(text, font='slant', color='green')`: Mencetak teks tertentu dengan cara yang menarik menggunakan font dan warna yang ditentukan.

### 3.3 *Interface*

Program berinteraksi dengan pengguna melalui antarmuka baris perintah. Ia menerima dua argumen baris perintah yang mewakili jalur ke file PDA dan HTML. Ia membaca file-file ini, memproses file HTML yang sudah di tokenisasi dengan regular expression dan manual tokenisasi kemudian diproses menggunakan PDA, dan mencetak apakah file HTML diterima atau tidak.

### 3.4 **Komponen Penting Lainnya**

1. *colorama*: Pustaka pihak ketiga yang digunakan untuk mewarnai teks keluaran.
2. *pyfiglet*: Pustaka pihak ketiga yang digunakan untuk menghasilkan seni ASCII dari teks.
3. *re*: Modul Python bawaan yang digunakan untuk operasi ekspresi reguler.
4. *sys*: Modul Python bawaan yang digunakan untuk berinteraksi dengan argumen baris perintah.

Program ini juga menggunakan beberapa fungsi bawaan Python seperti *open*, *read*, *split*, *replace*, *findall*, *append*, dan *pop*.

### 3.5 **Penjelasan Tambahan**

Pada program yang kami buat, void element hanya menerima input yang memiliki akhiran “/>”.

### 3.6 Uji Kasus

## 1. Kasus 1

```

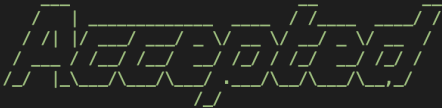
1  <html>
2    <head>
3      <title>Simple Webpage</title>
4      <script>
5        document.getElementById("demo").innerHTML = "Hello JavaScript!";
6      </script>
7    </head>
8    <body>
9      <h1>The script element</h1>
10     <a>Not going anywhere</a><br/>
11     <a href="https://www.google.co.id/">Might send you somewhere</a>
12
13     <p id="demo"></p>
14   </body>
15 </html>
16

```

```

(base) satriadhikara@Satriadhikaras-MacBook-Pro TUBES-TBF0-JOJIKA % python main.py pda.txt "kasus1.html"

```



Dalam kasus ini HTML berhasil diterima. Dalam html tersebut sudah mempunyai komponen wajib seperti tag hmtl, head, dan body. Ditambah juga peraturan-peraturan lainnya seperti tag title dalam head.

## 2. Kasus 2

```
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4
5 </head>
6 <body>
7
8 <h2>HTML Forms</h2>
9
10 <form action="/action_page.php" method="TEBAK">
11   <div id="label">First name:</div><br/>
12   <input type="text" id="fname"/><br/>
13   <div id="label">Last name:</div><br/>
14   <input type="text" id="lname"/><br/><br/>
15   <button type="submit">Submit</button>
16 </form>
17
18 <p>If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".</p>
19
20 </body>
21 </html>
```

```
• (base) satriadhikara@Satriadhikaras-MacBook-Pro TUBES-TBF0-JOJIIKA % python main.py pda.txt "kasus2.html"
```

Syntax Error

Dalam kasus ini HTML tidak diterima. Hal ini dikarenakan karena adanya suatu peraturan/*syntax* yang tidak sesuai yaitu pada tag form, tag form hanya bisa menerima *attribute* method POST atau GET.



### 3. Kasus 3

```

1  <html>
2  <head>
3    <title>HTML Contoh Tabel</title>
4  </head>
5  <body>
6    <!-- ini adalah tabel -->
7    <h2>Sample HTML Table</h2>
8
9    <table>
10      <tr>
11        <th>Name</th>
12        <th>Age</th>
13        <th>City</th>
14      </tr>
15      <tr>
16        <td>John Doe</td>
17        <td>25</td>
18        <td>New York</td>
19      </tr>
20      <tr>
21        <td>Jane Smith</td>
22        <td>30</td>
23        <td>San Francisco</td>
24      </tr>
25      <tr>
26        <td>Bob Johnson</td>
27        <td>22</td>
28        <td>Los Angeles</td>
29      </tr>
30    </table>
31
32  </body>
33 </html>
34

```

```
● (base) satriadhikara@Satriadhikaras-MacBook-Pro TUBES-TBF0-J0JIKA % python main.py pda.txt "kasus3.html"
```

Dalam kasus ini HTML diterima. Hal ini dikarenakan HTML tersebut sudah sesuai dengan *syntax*. Di sini bisa dilihat kalau HTML tersebut memakai banyak komponen tag tabel.

## PEMBAGIAN TUGAS

Nim	Nama	Tugas
13522121	Jonathan Emmanuel Saragih	Membuat program python yang membaca pda, dan membuat isi file txt, dan membuat laporan
13522125	Satriadhikara Panji Yudhistira	Membuat program python yang membaca pda, dan membuat isi file txt, dan membuat laporan
13522128	Mohammad Andhika Fadillah	Membuat program python yang membaca pda, dan membuat isi file txt, dan membuat laporan

## LAMPIRAN

1. [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/HTML basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)
  2. [https://eng.libretexts.org/Bookshelves/Computer Science/Book%3A Foundations of Computation \(Critchlow and Eck\)/04%3A Grammars/4.04%3A Pushdown Automata](https://eng.libretexts.org/Bookshelves/Computer_Science/Book%3A_Foundations_of_Computation_(Critchlow_and_Eck)/04%3A_Grammars/4.04%3A_Pushdown_Automata)
  3. [https://www.w3schools.com/html/html\\_formatting.asp](https://www.w3schools.com/html/html_formatting.asp)
  4. [https://www.researchgate.net/figure/Site-through-PDA-using-HTML-tables fig1 253889823](https://www.researchgate.net/figure/Site-through-PDA-using-HTML-tables_fig1_253889823)
  5. [https://www.w3schools.com/python/python\\_classes.asp](https://www.w3schools.com/python/python_classes.asp)
- Link ke repository github :  
<https://github.com/Andhikafdh/TUBES-TBFO-JOJIKA>
  - Link ke diagram state :  
<https://app.diagrams.net/#G1ai1FF30WAdYnWN7atccjY3VxOVO1tJ0n>