

**IF2124 TEORI BAHASA FORMAL DAN OTOMATA  
LAPORAN TUGAS PEMROGRAMAN**

**HTML CHECKER DENGAN PUSHDOWN AUTOMATA**



Disusun oleh:

Zahira Dina	13522085
Shulha	13522087
Hayya Zuhaili	13522102

**PROGRAM STUDI INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2023**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>BAB I: TEORI DASAR.....</b>	<b>3</b>
<b>BAB II: HASIL PUSHDOWN AUTOMATA.....</b>	<b>5</b>
<b>BAB III: IMPLEMENTASI PROGRAM.....</b>	<b>9</b>
<b>BAB IV: EKSPERIMEN DAN PENGUJIAN.....</b>	<b>11</b>
IV. 1. Studi Kasus 1.....	11
IV. 2. Studi Kasus 2.....	11
IV. 3. Studi Kasus 3.....	12
IV. 4. Studi Kasus 4.....	13
IV. 5. Studi Kasus 5.....	14
IV. 6. Studi Kasus 6.....	15
<b>BAB V.....</b>	<b>16</b>
V. 1 Link Repository.....	16
V. 2 Link State Diagram.....	16
V. 3 Pembagian Tugas.....	16
V. 4 Referensi.....	16

# BAB I

## TEORI DASAR

Pushdown Automata (PDA) adalah model mesin otomata yang digunakan untuk mengenali bahasa bebas konteks. PDA direpresentasikan sebagai mesin yang dilengkapi dengan penyimpanan tambahan berupa tumpukan (stack). Tumpukan adalah struktur data yang memungkinkan penyimpanan dan pengambilan data sesuai dengan prinsip Last In First Out (LIFO), di mana elemen yang terakhir dimasukkan akan menjadi yang pertama dikeluarkan.

Perbedaan mendasar antara Pushdown Automata dan Finite Automata terletak pada jenis bahasa yang dapat diakui. PDA digunakan untuk bahasa bebas konteks, sementara Finite Automata digunakan untuk bahasa reguler. PDA memanfaatkan tumpukan untuk menyimpan informasi yang lebih kompleks, yang tidak dimiliki oleh Finite Automata.

Desain PDA, seperti Finite Automata, melibatkan penggunaan state awal dan state penerima. State awal ditandai dengan panah, dan state penerima ditandai dengan lingkaran ganda. Setiap transisi dalam PDA dijelaskan oleh sebuah triplet yang menentukan elemen alfabet input, elemen yang diambil dari tumpukan (top of stack), dan elemen yang akan didorong ke dalam tumpukan. Penting untuk dicatat bahwa transisi hanya dapat terjadi jika elemen yang diambil dari tumpukan berada di bagian atasnya. Secara formal, PDA dapat didefinisikan sebagai pasangan tujuh elemen, yaitu

- $Q$  : himpunan state (*set of states*)
- $\Sigma$  : alfabet input (*input symbols*)
- $\Gamma$  : alfabet/simbol stack (*stack symbols*)
- $q_0$  : state awal (*start state*)
- $Z$  : simbol awal stack (*start stack symbol*)
- $F$  : himpunan state penerima (*final states*)
- $\delta$  : fungsi transisi (*transition functions/rules*)

Hypertext Markup Language (HTML) adalah bahasa markup yang digunakan untuk mengatur dan menampilkan konten web. HTML adalah salah satu bahasa utama dalam pengembangan web, memungkinkan deskripsi elemen-elemen seperti teks, gambar, tautan, dan media dalam tampilan browser web.

Setiap dokumen HTML dimulai dengan elemen `<html>`, diikuti oleh `<head>` (untuk metadata dan tautan ke file eksternal) dan `<body>` (untuk konten yang ditampilkan). HTML

menggunakan elemen (tags) untuk mengelompokkan dan mengorganisir konten, seperti `<p>` untuk paragraf, `<h1>` hingga `<h6>` untuk judul, `<a>` untuk tautan, dan `<img>` untuk gambar. Elemen HTML sering memiliki atribut untuk memberikan informasi tambahan, seperti `src` untuk gambar, `href` untuk tautan, dan `class` untuk gaya CSS.

Sintaks HTML yang benar membutuhkan tag pembuka dan penutup yang sesuai, dan kesalahan dalam penulisan dapat menyebabkan masalah pada tampilan halaman web. Penanganan kasus pembukaan dan penutupan tag, analog dengan tanda kurung dalam ekspresi matematika, dapat dikelola dengan memanfaatkan prinsip Last In First Out (LIFO) dari struktur tumpukan.

Dalam skrip HTML, ketika tag pembuka dijumpai, simbol tumpukan akan dimasukkan ke dalam tumpukan, dan simbol tersebut akan dihapus ketika tag penutup yang sesuai ditemukan. Transisi antar state dilakukan untuk memastikan bahwa semua tag yang diperlukan dalam skrip HTML dapat dibaca, menuju ke state penerima. Pada state penerima, periksa bahwa tumpukan kosong menjamin bahwa semua tag pembuka telah tertutup dengan tag penutupnya.

## BAB II

### HASIL PUSHDOWN AUTOMATA

Pada *Pushdown Automata* terdapat simbol-simbol yang kami gunakan untuk merepresentasikan suatu kondisi pada sebuah *state* dan *stack*. Selain itu terdapat pula macam-macam *string* yang mungkin akan diterima sebagai masukan dalam program alias *input word*. Kami menggunakan jenis program yang diterima dengan sampainya kondisi pada kondisi final yakni *accepting state*.

Aturan pada *Pushdown Automata* (PDA) mengikuti format sebagai berikut,

[current state] [input word] [top of stack] [next state] [element to push]

Tabel 2.1 Simbol yang dipakai pada *Pushdown Automata*

SYMBOLS	
State symbols	f q0 q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 q12 q13 q14 q15 q16 q17 q18 q19 q20 q21 q22 p0 p1 p2 r0 r1 r2 s2 s3 s4 s5 ss6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21
Input word symbols	< > html /html head /head body /body any id= class= style= title /title link script /script rel= "a" h1 /h1 h2 /h2 h3 /h3 h4 /h4 h5 /h5 h6 /h6 p /p br hr div /div a /a img button form input table /table td /td tr /tr th /th href= src= alt= action= em /em b /b abbr /abbr strong /strong small /small type= "text" "password" "email" "number" "checkbox" "submit" "reset" "button" method= "GET" "POST" -- !--
Stack symbols	Z X Y A B C D E F G H I J K L M N O P Q R S T 1 2 3 4 5
Starting state	q0
Starting stack	Z
Accepting state	f
Accept with final state	F

Tabel 2.2 Aturan yang dipakai pada *Pushdown Automata*

RULES		
q0 < Z q0 AZ q0 html A q0 B q0 > B q1 X q0 id= B p0 1B q0 class= B p0 1B q0 style= B p0 1B q0 !-- A s8 SA q1 < X q1 AX q1 head A q1 B q1 > B q2 X q1 id= B p0 2B q1 class= B p0 2B q1 style= B p0 2B q1 !-- A s9 SA q2 !-- A s10 SA q2 < X q2 AX q2 /head A q2 eps q2 > X q3 eps q2 id= B p0 32 q2 class= B p0 3B q2 style= B p0 3B q2 title A q6 B q2 link A q6 C q2 script A q6 D q3 < X q3 AX q3 body A q3 B q3 > B q4 X q3 !-- A s11 SA q4 < X q4 AX q4 /body A q4 eps q4 > X q5 eps q4 link A q11 B q4 script A q11 C q4 h1 A q11 D q4 h2 A q11 E q4 h3 A q11 F q4 h4 A q11 G q4 h5 A q11 H q4 h6 A q11 I q4 p A q11 J q4 br A q11 K q4 hr A q11 K q4 div A q11 L q4 a A q11 M q4 img A q11 N	q11 id= P p1 1P q11 class= P p1 1P q11 style= P p1 1P q11 action= P p1 1P q11 id= Q p1 1Q q11 class= Q p1 1Q q11 style= Q p1 1Q q11 id= R p1 1R q11 class= R p1 1R q11 method= P p2 2P q11 style= R p1 1R q11 rel= B q12 1B q11 src= N q13 1N q11 > K q4 eps q11 > Q q4 eps q11 > C q15 1 q11 > D q16 D q11 > E q16 E q11 > F q16 F q11 > G q16 G q11 > H q16 H q11 > I q16 I q11 > J q16 J q11 > M q16 M q11 type= O p2 3O q11 > O q16 O q11 type= Q p2 1Q q11 > R q16 R q11 > L q4 L q11 method= P p2 2P q11 > P q4 P q12 any 1 q13 eps q13 id= B p1 2B q13 class= B p1 2B q13 style= B p1 2B q13 href= B p1 2B q13 > B q4 eps q13 any 1 q14 eps q14 id= N p1 3N q14 class= N p1 3N q14 style= N p1 3N q14 alt= N p1 3N q14 > N q4 eps q15 !-- 2 s2 S2 q15 any 1 q15 1	q18 < B q19 C q19 > D q16 eps q19 th C q19 1C q19 td C q19 2C q19 /tr C q19 D q19 > 1 q20 1 q19 > 2 q20 2 q19 !-- C s14 SC q20 any 1 q20 1 q20 any 2 q20 2 q20 < 1 q21 1 q20 < 2 q21 2 q21 /th 1 q18 eps q21 /td 2 q18 eps q21 em 1 r0 11 q21 em 2 r0 12 q21 b 1 r0 21 q21 b 2 r0 22 q21 abbr 1 r0 31 q21 abbr 2 r0 32 q21 strong 1 r0 41 q21 strong 2 r0 42 q21 small 1 r0 51 q21 small 2 r0 52 q21 !-- 1 s6 S1 q21 !-- 2 s6 s2 q22 > L q4 eps r0 > 1 r1 1 r0 > 2 r1 2 r0 > 3 r1 3 r0 > 4 r1 4 r0 > 5 r1 5 r1 any 1 r1 1 r1 any 2 r1 2 r1 any 3 r1 3 r1 any 4 r1 4 r1 any 5 r1 5 r1 < 1 r2 A r1 < 2 r2 B r1 < 3 r2 C r1 < 4 r2 D r1 < 5 r2 E r2 /em A r2 eps r2 /b B r2 eps r2 /abbr C r2 eps

q4 button A q11 O	q15 < 1 q15 2	r2 /strong D r2 eps
q4 form A q11 P	q15 /script 2 q15 3	r2 /small E r2 eps
q4 input A q11 Q	q15 > 3 q4 eps	r2 > D q16 D
q4 table A q11 R	q16 any D q16 D	r2 > E q16 E
q4 /div A q22 eps	q16 any E q16 E	r2 > F q16 F
q4 < L q4 AL	q16 any F q16 F	r2 > G q16 G
q4 any L q4 L	q16 any G q16 G	r2 > H q16 H
q4 div A q11 L	q16 any H q16 H	r2 > I q16 I
q4 em A r0 1A	q16 any I q16 I	r2 > J q16 J
q4 b A r0 2A	q16 any J q16 J	r2 > M q16 M
q4 abbr A r0 3A	q16 any M q16 M	p0 any 1 q0 eps
q4 strong A r0 4A	q16 < D q17 D	p0 any 2 q1 eps
q4 small A r0 5A	q16 < E q17 E	p0 any 3 q2 eps
q4 !-- A s12 SA	q16 < F q17 F	p0 any 4 q6 eps
q4 /form A q17 eps	q16 < G q17 G	p1 any 1 q11 eps
q4 any P q4 P	q16 < H q17 H	p1 any 2 q13 eps
q4 < P q4 AP	q16 < I q17 I	p1 any 3 q14 eps
q5 < X q5 AX	q16 < J q17 J	p2 "text" 1 q11 eps
q5 /html A q5 eps	q16 < M q17 M	p2 "password" 1 q11 eps
q5 > X f eps	q16 < O q17 O	p2 "email" 1 q11 eps
q5 !-- A s13 SA	q16 any O q16 O	p2 "number" 1 q11 eps
q6 id= B p0 4B	q16 < R q17 R	p2 "checkbox" 1 q11 eps
q6 class= B p0 4B	q17 /h1 D q17 A	p2 "submit" 3 q11 eps
q6 style= B p0 4B	q17 /h2 E q17 A	p2 "reset" 3 q11 eps
q6 id= C p0 4C	q17 /h3 F q17 A	p2 "button" 3 q11 eps
q6 class= C p0 4C	q17 /h4 G q17 A	p2 "GET" 2 q11 eps
q6 style= C p0 4C	q17 /h5 H q17 A	p2 "POST" 2 q11 eps
q6 id= D p0 4D	q17 /h6 I q17 A	p2 "GET" 2 q11 eps
q6 class= D p0 4D	q17 /p J q17 A	p2 "POST" 2 q11 eps
q6 style= D p0 4D	q17 /a M q17 A	r2 > O q16 O
q6 rel= C q8 4E	q17 > A q4 eps	r2 > 1 q20 1
q6 > B q7 X	q17 em D r0 1D	r2 > 2 q20 2
q6 > D q7 Y	q17 b D r0 2D	r2 > A r3 eps
q7 < X q10 A	q17 abbr D r0 3D	r2 !-- A s7 SA
q7 < Y q10 B	q17 strong D r0 4D	r2 !-- B s7 SB
q7 any X q7 X	q17 small D r0 5D	r2 !-- C s7 SC
q7 any Y q7 Y	q17 em E r0 1E	r2 !-- D s7 SD
q8 any 4 q9 eps	q17 b E r0 2E	r2 !-- E s7 SE
q9 id= E p0 5E	q17 abbr E r0 3E	r2 > A r3 eps
q9 class= E p0 5E	q17 strong E r0 4E	r3 any L q4 L
q9 style= E p0 5E	q17 small E r0 5E	r3 < L q4 AL
q9 any 5 q9 eps	q17 em F r0 1F	r3 any P q4 P
q9 > E q2 eps	q17 b F r0 2F	s2 any S s2 S
q9 > X q2 X	q17 abbr F r0 3F	s2 -- S s2 eps
q9 > Y q2 Y	q17 strong F r0 4F	s2 > A q7 X
q10 /title A q9 eps	q17 small F r0 5F	s2 > B q7 Y
q10 /script B q9 eps	q17 em G r0 1G	s2 any S s2 S
q10 !-- A s2 SA	q17 b G r0 2G	s2 -- S s2 eps
q10 !-- B s0 SB	q17 abbr G r0 3G	s2 > 2 q15 1
q11 id= B p1 1B	q17 strong G r0 4G	s3 any S s3 S
q11 class= B p1 1B	q17 small G r0 5G	s3 -- S s3 T

q11 style= B p1 1B	q17 em H r0 1H	s3 > T q16 eps
q11 href= B p1 1B	q17 b H r0 2H	s4 !-- A s5 SA
q11 id= C p1 1C	q17 abbr H r0 3H	s5 any S s5 S
q11 class= C p1 1C	q17 strong H r0 4H	s5 -- S s5 eps
q11 style= C p1 1C	q17 small H r0 5H	s5 > A f eps
q11 src= C p1 1C	q17 em I r0 1I	s6 any S s6 S
q11 id= D p1 1D	q17 b I r0 2I	s6 -- S s6 T
q11 class= D p1 1D	q17 abbr I r0 3I	s6 > T q20 eps
q11 style= D p1 1D	q17 strong I r0 4I	s7 any S s7 S
q11 id= E p1 1E	q17 small I r0 5I	s7 -- S s7 eps
q11 class= E p1 1E	q17 em J r0 1J	s7 > A r1 1
q11 style= E p1 1E	q17 b J r0 2J	s7 > B r1 2
q11 id= F p1 1F	q17 abbr J r0 3J	s7 > C r1 3
q11 class= F p1 1F	q17 strong J r0 4J	s7 > D r1 4
q11 style= F p1 1F	q17 small J r0 5J	s7 > E r1 5
q11 id= G p1 1G	q17 em M r0 1M	s8 any S s8 S
q11 class= G p1 1G	q17 b M r0 2M	s8 -- S s8 T
q11 style= G p1 1G	q17 abbr M r0 3M	s8 > T s15 T
q11 id= H p1 1H	q17 strong M r0 4M	s9 any S s9 S
q11 class= H p1 1H	q17 small M r0 5M	s9 -- S s9 T
q11 style= H p1 1H	q17 > A q4 eps	s9 > T s16 T
q11 id= I p1 1I	q17 em O r0 1O	s10 any S s10 S
q11 class= I p1 1I	q17 b O r0 2O	s10 -- S s10 T
q11 style= I p1 1I	q17 abbr O r0 3O	s10 > T s17 T
q11 id= J p1 1J	q17 strong O r0 4O	s11 any S s11 S
q11 class= J p1 1J	q17 small O r0 5O	s11 -- S s11 T
q11 style= J p1 1J	q17 /button O q17 A	s11 > T s18 T
q11 id= K p1 1K	q17 > A q4 eps	s12 any S s12 S
q11 class= K p1 1K	q17 /table R q17 A	s12 -- S s12 T
q11 style= K p1 1K	q17 tr R q18 AR	s12 > T s19 T
q11 id= L p1 1L	q17 !-- D s3 SD	s13 any S s13 S
q11 class= L p1 1L	q17 !-- E s3 SE	s13 -- S s13 T
q11 style= L p1 1L	q17 !-- F s3 SF	s13 > T s20 T
q11 id= M p1 1M	q17 !-- G s3 SG	s14 any S s14 S
q11 class= M p1 1M	q17 !-- H s3 SH	s14 -- S s14 T
q11 style= M p1 1M	q17 !-- I s3 SI	s14 > T s21 T
q11 href= M p1 1M	q17 !-- J s3 SJ	s15 < T q0 eps
q11 id= N p1 1N	q17 !-- M s3 SM	s16 < T q1 eps
q11 class= N p1 1N	q17 !-- P s3 SP	s17 < T q2 eps
q11 style= N p1 1N	q17 !-- R s3 SR	s18 < T q3 eps
q11 alt= N p1 1N	q17 !-- O s3 SO	s19 < T q4 eps
q11 id= O p1 1O	q17 > P q4 eps	s20 < T q5 eps
q11 class= O p1 1O	q18 > A q18 B	s21 < T q19 eps
q11 style= O p1 1O	q18 > C q18 B	f < Z s4 AZ



## BAB III

### IMPLEMENTASI PROGRAM

Program *HTML Checker* yang kami buat dengan menggunakan algoritma *Pushdown Automata* disusun pada suatu file `main.py`. File ini nantinya akan membaca *rules* untuk PDA yang digunakan dari suatu file bernama `pda.txt`. File `main.py` ini akan meminta untuk menuliskan sebuah file HTML yang hendak diperiksa. File `main.py` ini akan membuka file masukan dan memeriksanya dengan *rules* yang telah dibaca sebelumnya.

Pada `main.py` telah dibuat beberapa variabel, fungsi, dan prosedur untuk mendukung pengecekan file html. Berikut struktur data pada file `main.py`:

#### 1. Variabel:

- ***rules***: Sebuah *dictionary* yang menyimpan aturan transisi dari satu state ke state lainnya.
- ***start\_state***: Sebuah *string* yang menyimpan nama state awal.
- ***start\_stack***: Sebuah *string* yang menyimpan simbol awal pada stack.
- ***current\_state***: Sebuah *string* yang menyimpan state saat ini.
- ***accept\_state***: Sebuah *string* yang menyimpan state akhir yang diterima.
- ***stack***: Sebuah *list* yang menyimpan isi stack pada suatu waktu.
- ***inp***: Sebuah *list* yang menyimpan input HTML yang akan diolah.
- ***line\_counter***: Sebuah *integer* yang menyimpan nomor baris saat terjadi syntax error.

#### 2. Fungsi dan Prosedur

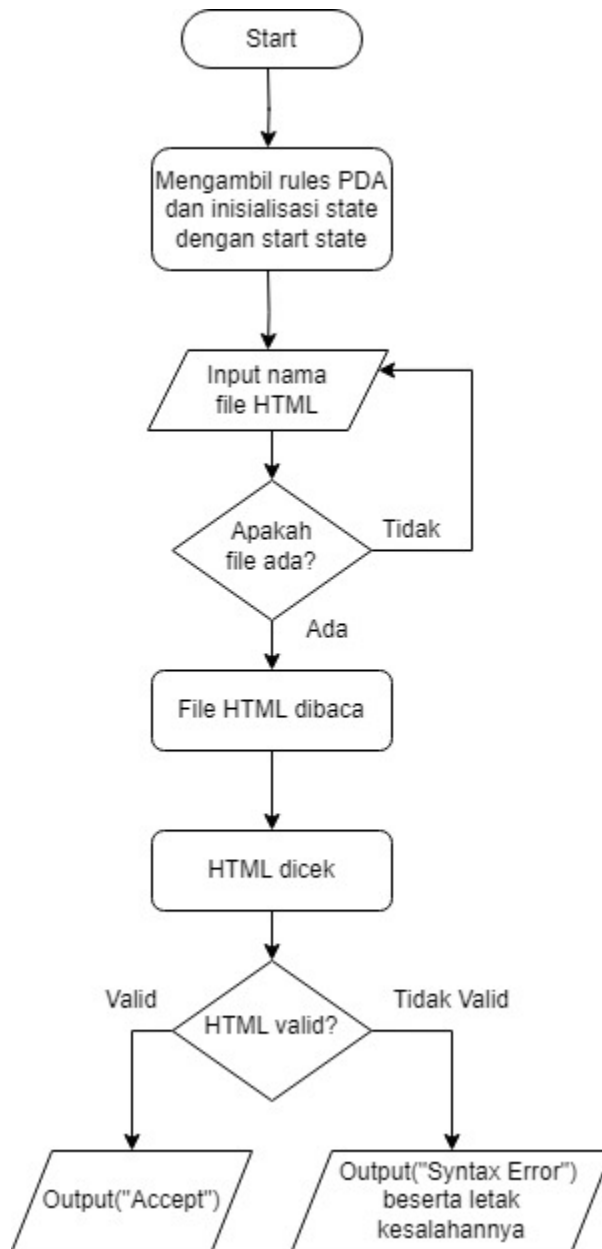
- ***check\_html()***: Memeriksa token pada input HTML, melaksanakan transisi sesuai aturan PDA, dan memeriksa ketersediaan aturan yang sesuai dengan kondisi stack dan input.
- ***get\_pda()***: Membaca file "`pda.txt`" yang berisi definisi PDA (Pushdown Automaton) dan menginisialisasi variabel-variabel seperti `start_state`, `start_stack`, `accept_state`, dan `rules`.
- ***get\_input(string)***: Mengolah string input HTML dan menghasilkan list `inp` yang berisi token-token.

#### 3. Program Utama

Program memulai validasi pada file HTML dengan meminta pengguna memasukkan nama file HTML. Setelah itu, program memeriksa keberadaan file tersebut, dan jika tidak ditemukan, pengguna diminta untuk memasukkan nama file kembali. Setelah file ditemukan, program membuka dan memproses isi HTML dengan memanggil fungsi `get_input()` untuk mendapatkan token. Kemudian, menggunakan definisi PDA dari file "`pda.txt`", program memvalidasi struktur HTML dengan memanggil prosedur ***check\_html()***. Antarmuka pada program ialah saat program meminta nama file HTML yang akan diperiksa. Selain itu, program akan mengeluarkan "Accepted" apabila HTML valid dan "Syntax Error" beserta letak kesalahannya apabila terdapat kesalahan sintaks

pada file HTML. akan memberikan output "Accepted" jika struktur HTML valid. Sebaliknya, jika terdapat kesalahan sintaks, program memberikan output "Syntax Error" dan menampilkan nomor baris tempat kesalahan terjadi pada file HTML.

Flowchart program:



Gambar 3.1 *Flowchart HTML Checker* dengan PDA

## BAB IV

### EKSPERIMEN DAN PENGUJIAN

#### IV. 1. Studi Kasus 1

Kasus berikut seharusnya akan menampilkan “Syntax Error” dan letak kesalahannya ialah pada baris pertama dikarenakan tag pertama bukan html.

File HTML yang diuji
<pre>&lt;&gt; hmif.html U x test &gt; &lt;&gt; hmif.html &gt; ... 1  &lt;hmif&gt; 2      &lt;head&gt; 3          &lt;title&gt;Simple Webpage&lt;/title&gt; 4      &lt;/head&gt; 5      &lt;body&gt; 6          &lt;h1&gt;Hello, World!&lt;/h1&gt; 7          &lt;p&gt;This is a simple webpage.&lt;/p&gt; 8      &lt;/body&gt; 9  &lt;/hmif&gt;</pre>
Hasil Pengujian
<pre>zahir@LAPTOP-V9MSHBAK MINGW64 /d/hirs/Tubes TBFO/HTMLCheckerUsingPDA/src (main) \$ python -u "d:\hirs\Tubes TBFO\HTMLCheckerUsingPDA\src\main.py"  Tuliskan nama file yang akan di cek: hmif.html Syntax Error There is fault in line:  1 &lt;hmif&gt;</pre>

#### IV. 2. Studi Kasus 2

Kasus berikut seharusnya akan menampilkan “Accepted”.

File HTML yang diuji
<pre>&lt;&gt; inputAcc.html x test &gt; &lt;&gt; inputAcc.html &gt; html 1  &lt;html&gt; 2      &lt;head&gt; 3          &lt;title&gt;Simple Webpage&lt;/title&gt; 4      &lt;/head&gt; 5      &lt;body&gt; 6          &lt;h1&gt;Hello, World!&lt;/h1&gt; 7          &lt;p&gt;This is a simple webpage.&lt;/p&gt; 8      &lt;/body&gt; 9  &lt;/html&gt;</pre>

Hasil Pengujian

```
$ python -u "d:\hirs\Tubes TBFO\HTMLCheckerUsingPDA\src\main.py"

Tuliskan nama file yang akan di cek: inputAcc.html
Accepted
```

### IV. 3. Studi Kasus 3

Kasus berikut seharusnya akan menampilkan “Accepted”.

File HTML yang diuji

```
<> acceptTable.html •
test > <> acceptTable.html > ...
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          <table>
10             <tr>
11                 <th>Company</th>
12                 <th>Contact</th>
13                 <th>Country</th>
14             </tr>
15             <tr>
16                 <td>Alfreds Futterkiste</td>
17                 <td>Maria Anders</td>
18                 <td>Germany</td>
19             </tr>
20             <tr>
21                 <td>Centro comercial Moctezuma</td>
22                 <td>Francisco Chang</td>
23                 <td>Mexico</td>
24             </tr>
25         </table>
26     </body>
27 </html>
28 |
```

Hasil Pengujian

```
$ python -u "d:\hirs\Tubes TBFO\HTMLCheckerUsingPDA\src\main.py"

Tuliskan nama file yang akan di cek: acceptTable.html
Accepted
```

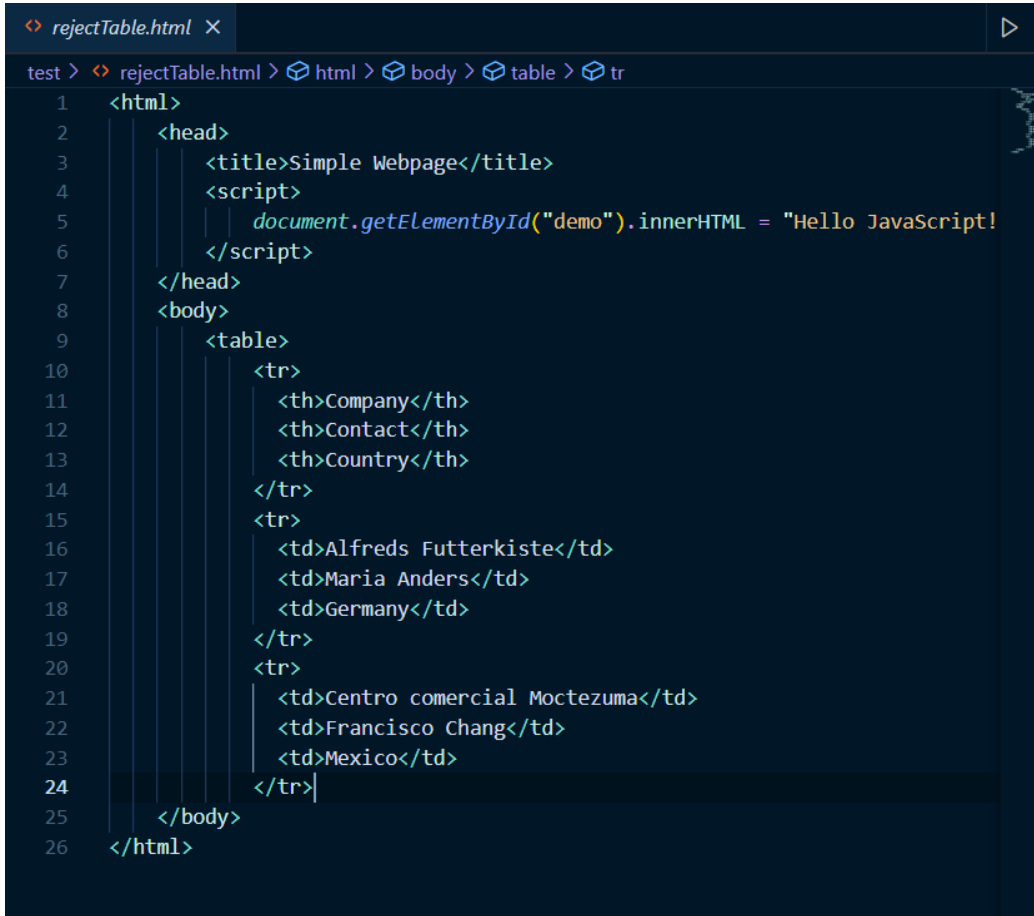
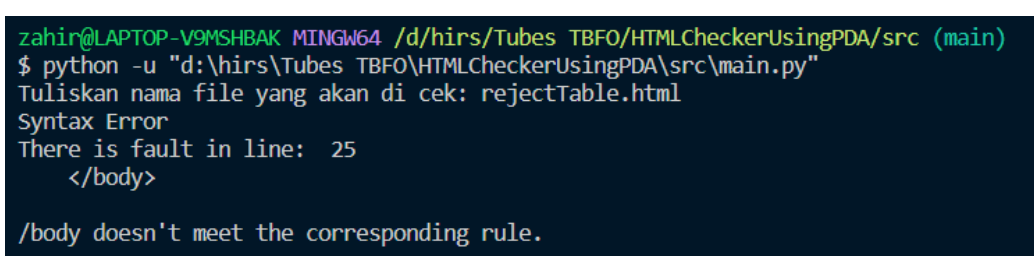
#### IV. 4. Studi Kasus 4

Kasus berikut seharusnya akan menampilkan “Syntax Error” dan letak kesalahannya ialah pada baris kesembilan karena tag img tidak dilengkapi dengan attribute src.

File HTML yang diuji
<pre>&lt;? reject4.html X test &gt; &lt;? reject4.html &gt; html 1 &lt;html&gt; 2 &lt;head&gt; 3   &lt;title&gt;Simple Webpage&lt;/title&gt; 4 &lt;/head&gt; 5 &lt;body&gt; 6   &lt;!-- Bagian utama web --&gt; 7   &lt;h1&gt;Hello, World!&lt;/h1&gt; 8   &lt;h2&gt;Welcome to my page&lt;/h2&gt; 9   &lt;img alt="Welcome Banner"&gt; 10   &lt;p&gt;This is a &lt;em&gt;simple&lt;/em&gt; webpage.&lt;/p&gt; 11   12   13   &lt;!-- Custom element --&gt; 14   &lt;div id="footer" class="footer"&gt; This is the end of the page &lt;/div&gt; 15 &lt;/body&gt; 16 &lt;/html&gt;</pre>
Hasil Pengujian
<pre>\$ python -u "d:\hirs\Tubes TBFO\HTMLCheckerUsingPDA\src\main.py"  Tuliskan nama file yang akan di cek: reject4.html Syntax Error There is fault in line: 9     &lt;img alt="Welcome Banner"&gt;</pre>

#### IV. 5. Studi Kasus 5

Kasus berikut seharusnya akan menampilkan “Syntax Error” dan letak kesalahannya ialah pada baris kedupuluhempat tag table tidak dilengkapi closing tag nya.

File HTML yang diuji
 <pre>1 &lt;html&gt; 2   &lt;head&gt; 3     &lt;title&gt;Simple Webpage&lt;/title&gt; 4     &lt;script&gt; 5       document.getElementById("demo").innerHTML = "Hello JavaScript! 6     &lt;/script&gt; 7   &lt;/head&gt; 8   &lt;body&gt; 9     &lt;table&gt; 10      &lt;tr&gt; 11        &lt;th&gt;Company&lt;/th&gt; 12        &lt;th&gt;Contact&lt;/th&gt; 13        &lt;th&gt;Country&lt;/th&gt; 14      &lt;/tr&gt; 15      &lt;tr&gt; 16        &lt;td&gt;Alfreds Futterkiste&lt;/td&gt; 17        &lt;td&gt;Maria Anders&lt;/td&gt; 18        &lt;td&gt;Germany&lt;/td&gt; 19      &lt;/tr&gt; 20      &lt;tr&gt; 21        &lt;td&gt;Centro comercial Moctezuma&lt;/td&gt; 22        &lt;td&gt;Francisco Chang&lt;/td&gt; 23        &lt;td&gt;Mexico&lt;/td&gt; 24      &lt;/tr&gt; 25    &lt;/body&gt; 26  &lt;/html&gt;</pre>
Hasil Pengujian
 <pre>zahir@LAPTOP-V9MSHBK MINGW64 /d/hirs/Tubes TBFO/HTMLCheckerUsingPDA/src (main) \$ python -u "d:\hirs\Tubes TBFO\HTMLCheckerUsingPDA\src\main.py" Tuliskan nama file yang akan di cek: rejectTable.html Syntax Error There is fault in line: 25     &lt;/body&gt;  /body doesn't meet the corresponding rule.</pre>

#### IV. 6. Studi Kasus 6

Kasus berikut seharusnya akan menampilkan “Syntax Error” dan letak kesalahannya ialah pada baris kesepuluh karena attribute action kosong.

File HTML yang diuji
Hasil Pengujian

## BAB V

### LAMPIRAN

#### V. 1 Link Repository

Berikut link repository kami:

<https://github.com/shulhajws/HTMLCheckerUsingPDA.git>

#### V. 2 Link State Diagram

Berikut link state diagram kami:

<https://drive.google.com/file/d/1gq2MWJwzamalVcAtSkjLfzD2uBFNJPBi/view?usp=ssharing>

#### V. 3 Pembagian Tugas

No	NIM	Nama	Tugas
1.	13522085	Zahira Dina Amalia	<ul style="list-style-type: none"><li>- Menyusun laporan bab 2-5</li><li>- Menyusun PDA</li><li>- Testing</li></ul>
2.	13522087	Shulha	<ul style="list-style-type: none"><li>- Menyusun laporan bab 1</li><li>- Membantu dalam merancang dan menyusun PDA</li><li>- Testing</li></ul>
3.	13522102	Hayya Zuhailii Kinasih	<ul style="list-style-type: none"><li>- Merancang diagram PDA</li><li>- Merancang program utama</li><li>- Testing</li></ul>

#### V. 4 Referensi

<https://www.oreilly.com/library/view/understanding-computation/9781449330071/ch04.html>

<https://meral.edu.mm/records/3683>

<https://github.com/theodoregold/pushdown-automata>