

## **Tugas Pemrograman**

**IF 2124 Teori Bahasa Formal dan Otomata**

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



**Kelompok TRUMPET SAKTI :**

- |                                   |                 |
|-----------------------------------|-----------------|
| <b>1. Keanu Amadius Gonza W.</b>  | <b>13522082</b> |
| <b>2. Fabian Radenta Bangun</b>   | <b>13522105</b> |
| <b>3. Dimas Bagoes Hendrianto</b> | <b>13522120</b> |

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

## DAFTAR ISI

DAFTAR ISI .....	1
BAB I Deskripsi Masalah .....	2
BAB II Teori Singkat .....	5
1. <i>Push Down Automata</i> (PDA) .....	5
2. <i>Hypertext Markup Language</i> (HTML) .....	6
3. Implementasi PDA pada sintaksis HTML .....	7
BAB III Analisa Pemecahan Masalah .....	8
BAB IV Spesifikasi Dan Pengujian .....	22
BAB V Penutup .....	32
Daftar Pustaka .....	33

## BAB I

### Deskripsi Masalah

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>
```

**Gambar 1.1.** Struktur HTML yang diharapkan

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			Harus di dalam head 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

## BAB II

### Teori Singkat

#### *Push Down Automata (PDA)*

PDA adalah model mesin abstrak dalam teori automata yang memiliki peran penting dalam pemodelan bahasa kontekstual. PDA merupakan nondeterministic finite state machine yang memiliki memori tambahan dalam bentuk stack. Push Down Automata (PDA) merupakan mesin otomata dari bahasa bebas konteks. PDA digambarkan sebagai tempat penyimpanan tidak terbatas berupa stack/tumpukan. Stack adalah kumpulan elemen sejenis yang disimpan atau disusun dengan cara ditumpuk. Prinsip stack adalah Last In First Out (LIFO) yaitu elemen yang masuk dahulu keluar akhir. PDA memiliki kemampuan lebih dari finite state machine, tetapi tidak lebih baik dari turing machine. PDA terdiri atas pasangan 7 buah tuple  $M = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, A)$ , dimana:

- $Q$  adalah himpunan dari states
- $\Sigma$  adalah alfabet input
- $\Gamma$  adalah alfabet stack
- $\delta$  adalah fungsi transisi
- $q_0$  adalah start state
- $Z$  adalah start symbol
- $F$  adalah himpunan final states

PDA digunakan untuk mengenali bahasa kontekstual, yaitu kelas bahasa formal yang lebih kuat daripada bahasa yang dapat dikenali oleh mesin state biasa. Kemampuan PDA untuk menggunakan stek memungkinkannya mengenali struktur kontekstual yang lebih kompleks. Operasi dasar yang dapat dilakukan pada stek oleh PDA melibatkan penyisipan (push) dan penghapusan (pop) simbol dari stek. PDA menggunakan stek untuk menyimpan informasi sementara selama proses pengenalan bahasa.

PDA memiliki beberapa varian, termasuk PDA non-deterministik (NPDA) yang memungkinkan beberapa pilihan transisi untuk keadaan yang sama. PDA non-deterministik lebih kuat daripada PDA deterministik dan dapat mengenali beberapa bahasa yang tidak dapat dikenali oleh PDA deterministik.

## ***Hypertext Markup Language (HTML)***

Bahasa markah standar untuk dokumen yang dirancang untuk ditampilkan di peramban internet. Ini dapat dibantu oleh teknologi seperti Cascading Style Sheets (CSS) dan bahasa skrip lainnya seperti JavaScript, VBScript, dan PHP. Peramban internet menerima dokumen HTML dari server web atau dari penyimpanan lokal dan membuat dokumen menjadi halaman web multimedia. HTML menggambarkan struktur halaman web secara semantik dan isyarat awal yang disertakan untuk penampilan dokumen.

Elemen HTML digambarkan oleh *tag*, ditulis menggunakan tanda kurung siku. Tag seperti `<img />` dan `<input />` langsung memperkenalkan konten ke dalam halaman. Tag lain seperti `<p>` mengelilingi dan memberikan informasi tentang teks dokumen dan mungkin menyertakan tag lain sebagai sub-elemen. Peramban tidak menampilkan tag HTML, tetapi menggunakannya untuk menafsirkan konten halaman.

HTML dapat menyematkan program yang ditulis dalam bahasa skrip seperti JavaScript, yang memengaruhi perilaku dan konten halaman web. Dimasukkannya CSS mendefinisikan tampilan dan tata letak konten. World Wide Web Consortium (W3C), mantan pengelola HTML dan pemelihara standar CSS saat ini, telah mendorong penggunaan CSS pada HTML presentasi eksplisit sejak 1997. Fungsi HTML adalah untuk mengatur tampilan halaman web. Terutama menyusun teks seperti paragraf dan heading, hingga menambahkan link dan gambar.

Berikut adalah beberapa fungsi HTML lainnya:

### **1. Membuat Struktur Halaman Web**

Fungsi utama HTML adalah untuk membangun struktur halaman web. Mulai dari membuat header, footer, navigasi, hingga kontennya. Peran HTML juga sebagai pondasi halaman web. Sebab, untuk menerapkan beberapa bahasa pemrograman maupun bahasa lainnya seperti CSS dan JavaScript, website membutuhkan struktur yang dibangun dari HTML. Jadi, tak berlebihan juga jika HTML dikatakan sebagai dasar untuk memahami bahasa pemrograman.

### **2. Menambahkan Konten Website**

Selain membuat struktur, Anda juga bisa memasukkan gambar di HTML dan menambah konten pada halaman web. Sebab, bahasa markup ini memungkinkan Anda

menyisipkan ilustrasi dan video, membuat tabel HTML, dan memasukkan media lainnya. Misalnya, dengan menggunakan tag `<p>` untuk menambah paragraf, tag `<img>` untuk memasukkan gambar.

### 3. Mengatur Format dan Tata Letak

Berikutnya, HTML memungkinkan Anda mengatur format dan mengatur tata letak konten dalam halaman web. Seperti memilih ukuran dan jenis huruf, menentukan warna huruf dengan berbagai kode warna HTML, serta mengatur tata letak konten pada website. Semisal untuk menandai kalimat cetak tebal, gunakan kode HTML `<bold>`. Sedangkan untuk memiringkan tulisan, gunakan tag `<italic>`. Dengan kode HTML yang tepat, Anda dapat membuat tampilan konten menarik serta mudah dibaca.

### 4. Mengarahkan Pengguna ke Halaman Website Lain

HTML juga berfungsi mengarahkan pengguna ke halaman atau website lain menggunakan link tertentu. Link ini bisa disematkan ke dalam teks tertentu, alias sebagai anchor text. Cara membuat hyperlink dengan kode HTML yaitu menggunakan tag `<a>`. Dengan hyperlink, pengunjung bisa dengan mudah mendapatkan informasi yang saling berhubungan dengan halaman yang sedang dibaca. Anda juga bisa mengarahkan mereka ke landing page tertentu.

## ***Implementasi PDA pada sintaksis HTML***

Struktur program HTML pada dasarnya adalah kumpulan tag biasa dan void. Setiap tag pada umumnya pasti memiliki open tag dan close tag, tapi yang lebih utamanya lagi pasti memiliki “<”, “>”, “/”, dan nama dari tag itu sendiri yang bisa digunakan dalam penerapan push dan stack. Kelompok kami memakai pendekatan PDA dengan *empty stack* yang menurut kami lebih mudah dan lebih cocok digunakan dalam menyelesaikan masalah tugas pemrograman ini.

Sebagai contoh pada stack `<html> </html>`, anggap suatu program HTML yang hanya tersusun dari tag tersebut valid. Pertama kali akan menerima “<” diatas  $Z_0$ , dilanjutkan dengan “html” yang melalui validasi terlebih dahulu kalau nama tag benar sebagai “html” jika tidak maka simbol dan stack akan tidak bisa di *pop* nantinya. Lanjut menerima “>” dan diikuti dengan “<” dan “/” yang menandakan perpindahan state yang menuju ke tahapan *pop*. Lanjut akan memeriksa nama dari tag penutup yang harus sama dengan yang ada dalam stack yaitu “html”, karena sama maka “html” dalam stack bisa di *pop*. Terakhir akan menerima “>” dan menjadi penanda untuk *pop*  $Z_0$ . Didapatilah stack kosong yang berarti teks HTML valid.



### BAB III

#### Analisa Pemecahan Masalah

Q0 HTML HEAD PREBODY BODY TITLE SCRIPT LINK HTMLF F CEKa CEKa CEKb  
CEKc CEKc ACCEPT P H1 H2 H3 H4 H5 H6 DIV a BUTTON FORM IMG INPUT TYPE  
METHOD HR BR TABLE TR TD TH EM B SMALL STRONG ABBR GLOBAL

A a B b C c d D F f G g H h I i J j K k L l M m N n O o P p Q q R r S s T t U u V v W w X x  
Y y 1 2 3 4 5 6 7 8 9 0 ( ) @ # \$ % ^ = " < > ! - / ~

A a B b C c d D F f G g H h I i J j K k L l M m N n O o P p Q q R r S s T t U u V v W w X x  
Y y 1 2 3 4 5 6 7 8 9 0 ( ) @ # \$ % ^ = " < > ! - / ~ Z

Q0

Z

ACCEPT

F

Q0 < Z Q0 <Z	HEAD ( < SCRIPT (	PREBODY C < BODY C
Q0 A < HTML A	HEAD 1 < LINK 1	BODY > C BODY >C
HTML > A HTML >A	TITLE > D TITLE >D	BODY < > BODY <
HTML < > HTML <	TITLE ~ > TITLE >	BODY ~ > BODY >
HTML B < HEAD B	TITLE < > TITLE e	BODY c < HTMLF c
HEAD > B HEAD >B	TITLE d D HEAD e	HTMLF > c HTMLF >c
HEAD < > HEAD <	PREBODY > b PREBODY	HTMLF a < HTMLF a
HEAD b < PREBODY b	>b	HTMLF < > HTMLF <
HEAD D < TITLE D	PREBODY < > PREBODY	HTMLF > a F a
	<	

F e a CEKa e	BODY 9 < INPUT 9	H3 v V H3 e
CEKa e c CEKc e	BODY 1 < LINK 1	H3 > C BODY >C
CEKc e C CEKC e	P > F P >F	H4 > W H4 >W
CEKC e b CEKb e	P ~ > P >	H4 ~ > H4 >
CEKb e B CEKB e	P < > P e	H4 < > H4 e
CEKB e A CEKA e	P f F P e	H4 w W H4 e
CEKA e Z ACCEPT e	P > C BODY >C	H4 > C BODY >C
BODY F < P F	H1 > T H1 >T	H5 > X H5 >X
BODY T < H1 T	H1 ~ > H1 >	H5 ~ > H5 >
BODY U < H2 U	H1 < > H1 e	H5 < > H5 e
BODY V < H3 V	H1 t T H1 e	H5 x X H5 e
BODY W < H4 W	H1 > C BODY >C	H5 > C BODY >C
BODY X < H5 X	H2 > U H2 >U	H6 > Y H6 >Y
BODY Y < H6 Y	H2 ~ > H2 >	H6 ~ > H6 >
BODY L < DIV L	H2 < > H2 e	H6 < > H6 e
BODY M < a M	H2 u U H2 e	H6 y Y H6 e
BODY ( < SCRIPT (	H2 > C BODY >C	H6 > C BODY >C
BODY N < BUTTON N	H3 > V H3 >V	DIV > L DIV >L
BODY O < FORM O	H3 ~ > H3 >	DIV ~ > DIV >
BODY 7 < IMG 7	H3 < > H3 e	DIV < > DIV e

DIV 1 L DIV e	SCRIPT ) ( SCRIPT e	FORM # 0 METHOD #0
DIV > C BODY >C	SCRIPT > C BODY >C	FORM o 0 FORM e
a > M a >M	SCRIPT > B HEAD >B	FORM > C BODY >C
a ~ > a >	BUTTON > N BUTTON >N	METHOD = # METHOD =
a < > a e	BUTTON ~ > BUTTON >	METHOD " = METHOD "
a 3 M a 3M	BUTTON < > BUTTON e	METHOD + " METHOD "
a = 3 a =	BUTTON 0 N TYPE 0N	METHOD " " FORM e
a " = a "	BUTTON n N BUTTON e	BODY 6 < HR 6
a * " a "	BUTTON > C BODY >C	HR > 6 HR e
a " " a e	TYPE = 0 TYPE =	HR < C BODY <C
a m M a e	TYPE " = TYPE "	BODY 5 < BR 5
a > C BODY >C	TYPE ` " TYPE "	BR > 5 BR e
SCRIPT > ( SCRIPT >(	TYPE " " BUTTON e	BR < C BODY <C
SCRIPT ~ > SCRIPT >	FORM > 0 FORM >0	IMG 4 7 IMG 4
SCRIPT < > SCRIPT e	FORM ~ > FORM >	IMG = 4 IMG =4
SCRIPT 4 ( SCRIPT 4(	FORM < > FORM e	IMG " = IMG "
SCRIPT = 4 SCRIPT =	FORM @ 0 FORM @0	IMG * " IMG "
SCRIPT " = SCRIPT "	FORM = @ FORM =	IMG " " IMG e
SCRIPT * " SCRIPT "	FORM " = FORM "	IMG 8 4 IMG 84
SCRIPT " " SCRIPT e	FORM " " FORM e	IMG 8 7 IMG 87

IMG = 8 IMG =	TR < > TR e	FORM V 0 H3 V0
IMG > 4 IMG e	TR q Q TABLE e	H3 > 0 FORM >0
IMG < C BODY <C	TR R Q TD RQ	FORM W 0 H4 W0
LINK 2 1 LINK 2	TD > R TD >R	H4 > 0 FORM >0
LINK = 2 LINK =2	TD ~ > TD >	FORM X 0 H5 X0
LINK " = LINK "	TD < > TD e	H5 > 0 FORM >0
LINK * " LINK "	TD r R TR e	FORM Y 0 H6 Y0
LINK " " LINK e	TR S Q TH SQ	H6 > 0 FORM >0
LINK 3 2 LINK 32	TH > S TH >S	FORM L 0 DIV LO
LINK 3 1 LINK 31	TH ~ > TH >	DIV > 0 FORM >0
LINK = 3 LINK =	TH < > TH e	FORM L 0 DIV LO
LINK > 2 LINK e	TH s S TR e	DIV > 0 FORM >0
LINK < C BODY <C	FORM F 0 P FO	FORM N 0 BUTTON NO
LINK > 2 HEAD >	P > 0 FORM >0	BUTTON > 0 FORM >0
BODY P < TABLE P	FORM M 0 a MO	FORM ( 0 SCRIPT (0
TABLE > P TABLE >P	a > 0 FORM >0	SCRIPT > 0 FORM >0
TABLE < > TABLE e	FORM T 0 H1 TO	FORM 9 0 INPUT 90
TABLE p P BODY e	H1 > 0 FORM >0	INPUT < 0 FORM 0
TABLE Q P TR QP	FORM U 0 H2 UO	FORM 7 0 IMG 70
TR > Q TR >Q	H2 > 0 FORM >0	IMG < 0 FORM 0

FORM 1 0 LINK 10	ABBR < > ABBR e	H1 I T ABBR IT
LINK < 0 FORM 0	ABBR ~ > ABBR >	ABBR > T H1 >T
FORM 5 0 BR 50	ABBR i I ABBR e	H1 J T STRONG JT
BR < 0 FORM 0	ABBR > F P >F	STRONG > T H1 >T
FORM 6 0 HR 60	P J F STRONG JF	H1 K T SMALL KT
HR < 0 FORM 0	STRONG > J STRONG >J	SMALL > T H1 >T
P G F EM GF	STRONG < > STRONG e	H2 G U EM GU
EM > G EM >G	STRONG ~ > STRONG >	EM > U H2 >U
EM < > EM e	STRONG j J STRONG e	H2 H U B HU
EM ~ > EM >	STRONG > F P >F	B > U H2 >U
EM g G EM e	P K F SMALL KF	H2 I U ABBR IU
EM > F P >F	SMALL > K SMALL >K	ABBR > U H2 >U
P H F B HF	SMALL < > SMALL e	H2 J U STRONG JU
B > H B >H	SMALL ~ > SMALL >	STRONG > U H2 >U
B < > B e	SMALL k K SMALL e	H2 K U SMALL KU
B ~ > B >	SMALL > F P >F	SMALL > U H2 >U
B h H B e	H1 G T EM GT	H3 G V EM GV
B > F P >F	EM > T H1 >T	EM > V H3 >V
P I F ABBR IF	H1 H T B HT	H3 H V B HV
ABBR > I ABBR >I	B > T H1 >T	B > V H3 >V

H3 I V ABBR IV	H5 I X ABBR IX	a I M ABBR IM
ABBR > V H3 >V	ABBR > X H5 >X	ABBR > M a >M
H3 J V STRONG JV	H5 J X STRONG JX	a J M STRONG JM
STRONG > V H3 >V	STRONG > X H5 >X	STRONG > M a >M
H3 K V SMALL KV	H5 K X SMALL KX	a K M SMALL KM
SMALL > V H3 >V	SMALL > X H5 >X	SMALL > M a >M
H4 G W EM GW	H6 G Y EM GY	TITLE G D EM GD
EM > W H4 >W	EM > Y H6 >Y	EM > D TITLE >D
H4 H W B HW	H6 H Y B HY	TITLE H D B HD
B > W H4 >W	B > Y H6 >Y	B > D TITLE >D
H4 I W ABBR IW	H6 I Y ABBR IY	TITLE I D ABBR ID
ABBR > W H4 >W	ABBR > Y H6 >Y	ABBR > D TITLE >D
H4 J W STRONG JW	H6 J Y STRONG JY	TITLE J D STRONG JD
STRONG > W H4 >W	STRONG > Y H6 >Y	STRONG > D TITLE >D
H4 K W SMALL KW	H6 K Y SMALL KY	TITLE K D SMALL KD
SMALL > W H4 >W	SMALL > Y H6 >Y	SMALL > D TITLE >D
H5 G X EM GX	a G M EM GM	BUTTON G N EM GN
EM > X H5 >X	EM > M a >M	EM > N BUTTON >N
H5 H X B HX	a H M B HM	BUTTON H N B HN
B > X H5 >X	B > M a >M	B > N BUTTON >N

BUTTON I N ABBR IN	DIV I L ABBR IL	DIV T L H1 TL
ABBR > N BUTTON >N	ABBR > L DIV >L	H1 > L DIV >L
BUTTON J N STRONG JN	DIV J L STRONG JL	DIV U L H2 UL
STRONG > N BUTTON >N	STRONG > L DIV >L	H2 > L DIV >L
BUTTON K N SMALL KN	DIV K L SMALL KL	DIV V L H3 VL
SMALL > N BUTTON >N	SMALL > L DIV >L	H3 > L DIV >L
SCRIPT G ( EM G(	BODY G < EM G	DIV W L H4 WL
EM > ( SCRIPT >(	EM > C BODY >C	H4 > L DIV >L
SCRIPT H ( B H(	BODY H < B H	DIV X L H5 XL
B > ( SCRIPT >(	B > C BODY >C	H5 > L DIV >L
SCRIPT I ( ABBR I(	BODY I < ABBR I	DIV Y L H6 YL
ABBR > ( SCRIPT >(	ABBR > C BODY >C	H6 > L DIV >L
SCRIPT J ( STRONG J(	BODY J < STRONG J	DIV O L FORM OL
STRONG > ( SCRIPT >(	STRONG > C BODY >C	FORM > L DIV >L
SCRIPT K ( SMALL K(	BODY K < SMALL K	DIV 9 L INPUT 9L
SMALL > ( SCRIPT >(	SMALL > C BODY >C	INPUT > L DIV >L
DIV G L EM GL	DIV F L P FL	DIV N L BUTTON NL
EM > L DIV >L	P > L DIV >L	BUTTON > L DIV >L
DIV H L B HL	DIV M L a ML	DIV ( L SCRIPT (L
B > L DIV >L	a > L DIV >L	SCRIPT > L DIV >L

DIV 7 L IMG 7L	PREBODY > ! PREBODY	FORM - - FORM e
	>	
IMG < L DIV L		FORM > ! FORM >
	BODY ! < BODY !	
DIV 1 L LINK 1L		H1 ! T H1 !T
	BODY - ! BODY -!	
LINK < L DIV L		H1 - ! H1 -!
	BODY / - BODY -	
HTML ! < HTML !		H1 / - H1 -
	BODY - - BODY e	
HTML - ! HTML -!		H1 - - H1 e
	BODY > ! BODY >	
HTML / - HTML -		H1 > ! H1 >
	HTMLF ! < HTMLF !	
HTML - - HTML e		H2 ! U H2 !U
	HTMLF - ! HTMLF -!	
HTML > ! HTML >		H2 - ! H2 -!
	HTMLF / - HTMLF -	
HEAD ! < HEAD !		H2 / - H2 -
	HTMLF - - HTMLF e	
HEAD - ! HEAD -!		H2 - - H2 e
	HTMLF > ! HTMLF >	
HEAD / - HEAD -		H2 > ! H2 >
	DIV ! L DIV !L	
HEAD - - HEAD e		H3 ! V H3 !V
	DIV - ! DIV -!	
HEAD > ! HEAD >		H3 - ! H3 -!
	DIV / - DIV -	
PREBODY ! < PREBODY		H3 / - H3 -
!	DIV - - DIV e	
		H3 - - H3 e
PREBODY - ! PREBODY	DIV > ! DIV >	
-!		H3 > ! H3 >
	FORM ! O FORM !O	
PREBODY / - PREBODY		H4 ! W H4 !W
-	FORM - ! FORM -!	
		H4 - ! H4 -!
PREBODY - - PREBODY	FORM / - FORM -	
e		H4 / - H4 -



H4 - - H4 e	a - - a e	TD / - TD -
H4 > ! H4 >	a > ! a >	TD - - TD e
H5 ! X H5 !X	BUTTON ! N BUTTON !N	TD > ! TD >
H5 - ! H5 -!	BUTTON - ! BUTTON -	TH ! S TH !S
H5 / - H5 -	!	TH - ! TH -!
H5 - - H5 e	BUTTON / - BUTTON -	TH / - TH -
H5 > ! H5 >	BUTTON - - BUTTON e	TH - - TH e
H6 ! Y H6 !Y	BUTTON > ! BUTTON >	TH > ! TH >
H6 - ! H6 -!	TABLE ! P TABLE !P	HTML " = HTML "
H6 / - H6 -	TABLE - ! TABLE -!	HTML * " HTML "
H6 - - H6 e	TABLE / - TABLE -	HTML " " HTML e
H6 > ! H6 >	TABLE - - TABLE e	HTML \$ A HTML \$A
P ! F P !F	TABLE > ! TABLE >	HTML = \$ HTML =
P - ! P -!	TR ! Q TR !Q	HTML % A HTML %A
P / - P -	TR - ! TR -!	HTML = % HTML =
P - - P e	TR / - TR -	HTML ^ A HTML ^A
P > ! P >	TR - - TR e	HTML = ^ HTML =
a ! M a !M	TR > ! TR >	HEAD " = HEAD "
a - ! a -!	TD ! R TD !R	HEAD * " HEAD "
a / - a -	TD - ! TD -!	HEAD " " HEAD e

HEAD \$ B HEAD \$B	TITLE % D TITLE %D	H1 ^ T H1 ^T
HEAD = \$ HEAD =	TITLE = % TITLE =	H1 = ^ H1 =
HEAD % B HEAD %B	TITLE ^ D TITLE ^D	H2 " = H2 "
HEAD = % HEAD =	TITLE = ^ TITLE =	H2 * " H2 "
HEAD ^ B HEAD ^B	SCRIPT " = SCRIPT "	H2 " " H2 e
HEAD = ^ HEAD =	SCRIPT * " SCRIPT "	H2 \$ U H2 \$U
BODY " = BODY "	SCRIPT " " SCRIPT e	H2 = \$ H2 =
BODY * " BODY "	SCRIPT \$ ( SCRIPT \$(	H2 % U H2 %U
BODY " " BODY e	SCRIPT = \$ SCRIPT =	H2 = % H2 =
BODY \$ C BODY \$C	SCRIPT % ( SCRIPT %(	H2 ^ U H2 ^U
BODY = \$ BODY =	SCRIPT = % SCRIPT =	H2 = ^ H2 =
BODY % C BODY %C	SCRIPT ^ ( SCRIPT ^C	H3 " = H3 "
BODY = % BODY =	SCRIPT = ^ SCRIPT =	H3 * " H3 "
BODY ^ C BODY ^C	H1 " = H1 "	H3 " " H3 e
BODY = ^ BODY =	H1 * " H1 "	H3 \$ V H3 \$V
TITLE " = TITLE "	H1 " " H1 e	H3 = \$ H3 =
TITLE * " TITLE "	H1 \$ T H1 \$T	H3 % V H3 %V
TITLE " " TITLE e	H1 = \$ H1 =	H3 = % H3 =
TITLE \$ D TITLE \$D	H1 % T H1 %T	H3 ^ V H3 ^V
TITLE = \$ TITLE =	H1 = % H1 =	H3 = ^ H3 =

H4 " = H4 "

H6 " " H6 e

EM = \$ EM =

H4 \* " H4 "

H6 \$ Y H6 \$Y

EM % G EM %G

H4 " " H4 e

H6 = \$ H6 =

EM = % EM =

H4 \$ W H4 \$W

H6 % Y H6 %Y

EM ^ G EM ^G

H4 = \$ H4 =

H6 = % H6 =

EM = ^ EM =

H4 % W H4 %W

H6 ^ Y H6 ^Y

B " = B "

H4 = % H4 =

H6 = ^ H6 =

B \* " B "

H4 ^ W H4 ^W

P " = P "

B " " B e

H4 = ^ H4 =

P \* " P "

B \$ H B \$H

H5 " = H5 "

P " " P e

B = \$ B =

H5 \* " H5 "

P \$ F P \$F

B % H B %H

H5 " " H5 e

P = \$ P =

B = % B =

H5 \$ X H5 \$X

P % F P %F

B ^ H B ^H

H5 = \$ H5 =

P = % P =

B = ^ B =

H5 % X H5 %X

P ^ F P ^F

ABBR " = ABBR "

H5 = % H5 =

P = ^ P =

ABBR \* " ABBR "

H5 ^ X H5 ^X

EM " = EM "

ABBR " " ABBR e

H5 = ^ H5 =

EM \* " EM "

ABBR \$ I ABBR \$I

H6 " = H6 "

EM " " EM e

ABBR = \$ ABBR =

H6 \* " H6 "

EM \$ G EM \$G

ABBR % I ABBR %I

ABBR = % ABBR =	SMALL = ^ SMALL =	BUTTON * " BUTTON "
ABBR ^ I ABBR ^I	DIV " = DIV "	BUTTON " " BUTTON e
ABBR = ^ ABBR =	DIV * " DIV "	BUTTON \$ N BUTTON \$N
STRONG " = STRONG "	DIV " " DIV e	BUTTON = \$ BUTTON =
STRONG * " STRONG "	DIV \$ L DIV \$L	BUTTON % N BUTTON %N
STRONG " " STRONG e	DIV = \$ DIV =	BUTTON = % BUTTON =
STRONG \$ J STRONG \$J	DIV % L DIV %L	BUTTON ^ N BUTTON ^N
STRONG = \$ STRONG =	DIV = % DIV =	BUTTON = ^ BUTTON =
STRONG % J STRONG %J	DIV ^ L DIV ^L	FORM " = FORM "
STRONG = % STRONG =	DIV = ^ DIV =	FORM * " FORM "
STRONG ^ J STRONG ^J	a " = a "	FORM " " FORM e
STRONG = ^ STRONG =	a * " a "	FORM \$ O FORM \$O
SMALL " = SMALL "	a " " a e	FORM = \$ FORM =
SMALL * " SMALL "	a \$ M a \$M	FORM % O FORM %O
SMALL " " SMALL e	a = \$ a =	FORM = % FORM =
SMALL \$ K SMALL \$K	a % M a %M	FORM ^ O FORM ^O
SMALL = \$ SMALL =	a = % a =	FORM = ^ FORM =
SMALL % K SMALL %K	a ^ M a ^M	TABLE " = TABLE "
SMALL = % SMALL =	a = ^ a =	TABLE * " TABLE "
SMALL ^ K SMALL ^K	BUTTON " = BUTTON "	TABLE " " TABLE e

TABLE \$ P TABLE \$P	TD % R TD %R	BR ^ 5 BR ^5
TABLE = \$ TABLE =	TD = % TD =	BR = ^ BR =
TABLE % P TABLE %P	TD ^ R TD ^R	HR " = HR "
TABLE = % TABLE =	TD = ^ TD =	HR * " HR "
TABLE ^ P TABLE ^P	TH " = TH "	HR " " HR e
TABLE = ^ TABLE =	TH * " TH "	HR \$ 6 HR \$6
TR " = TR "	TH " " TH e	HR = \$ HR =
TR * " TR "	TH \$ R TH \$R	HR % 6 HR %6
TR " " TR e	TH = \$ TH =	HR = % HR =
TR \$ Q TR \$Q	TH % R TH %R	HR ^ 6 HR ^6
TR = \$ TR =	TH = % TH =	HR = ^ HR =
TR % Q TR %Q	TH ^ R TH ^R	IMG " = IMG "
TR = % TR =	TH = ^ TH =	IMG * " IMG "
TR ^ Q TR ^Q	BR " = BR "	IMG " " IMG e
TR = ^ TR =	BR * " BR "	IMG \$ 7 IMG \$7
TD " = TD "	BR " " BR e	IMG \$ 4 IMG \$4
TD * " TD "	BR \$ 5 BR \$5	IMG = \$ IMG =
TD " " TD e	BR = \$ BR =	IMG % 7 IMG %7
TD \$ R TD \$R	BR % 5 BR %5	IMG % 4 IMG %4
TD = \$ TD =	BR = % BR =	IMG = % IMG =

IMG ^ 7 IMG ^7	LINK ^ 1 LINK ^1	INPUT % 9 GLOBAL %9
IMG ^ 4 IMG ^4	LINK ^ 2 LINK ^2	INPUT % 0 GLOBAL %0
IMG = ^ IMG =	LINK = ^ LINK =	INPUT ^ 9 GLOBAL ^9
LINK " = LINK "	INPUT 0 9 INPUT 0	INPUT ^ 0 GLOBAL ^0
LINK * " LINK "	INPUT = 0 INPUT =0	GLOBAL = \$ GLOBAL =
LINK " " LINK e	INPUT " = INPUT "	GLOBAL = % GLOBAL =
LINK \$ 1 LINK \$1	INPUT ; " INPUT "	GLOBAL = ^ GLOBAL =
LINK \$ 2 LINK \$2	INPUT " " INPUT e	GLOBAL " = GLOBAL "
LINK = \$ LINK =	INPUT > 0 INPUT e	GLOBAL * " GLOBAL "
LINK % 1 LINK %1	INPUT < C BODY <C	GLOBAL " " INPUT e
LINK % 2 LINK %2	INPUT \$ 9 GLOBAL \$9	
LINK = % LINK =	INPUT \$ 0 GLOBAL \$0	

## BAB IV

### SPEKIFIKASI DAN PENGUJIAN

#### *SPEKIFIKASI*

`main.py`

program utama kami berisi pemanggilan fungsi dari file lainnya.

```
import..
```

berfungsi untuk multi file maksudnya yaitu kita dapat memanggil file lain di dalam satu module yang berbeda.

```
if __name__ == "__main__"
```

call fungsi dari file lain.

`pda.py`

program yang memproses text yang berisi rule PDA.

```
def start(filepda)
```

fungsi memulai membaca file menjadi rule – rule PDA menetapkan variabel variabel global

```
def process(arr):
```

fungsi yang akan memproses Grammar berdasarkan aturan yang telah dibuat.

```
def done()
```

fungsi yang akan menampilkan hasil pemrosesan file.

```
def readPDA(fileName)
```

fungsi yang membaca file baris per baris dan menyimpannya dalam variabel yang sudah dibuat diatas.

```
def generate(state, input, stack, config)
```

fungsi yang membuat semua konfigurasi yang mungkin dari masukan

```
def next(state, input, stack, config)
```

fungsi yang memeriksa apakah simbol yang diterima terminal atau non-terminal

```
def check(state, input, stack)
```

fungsi memeriksa hasil akhir yang berupa empty stack atau final state sudah benar

OpenFile.py

Program yang memproses file .html

```
def getDirectory()
```

fungsi yang membaca letak file .html mengembalikan pathnya.

```
def readFile>NamaFile)
```

fungsi yang membaca file .html baris per barisnya mengembalikan array berisi string (kalimat).

```
def splitFile(rawFile)
```

fungsi yang memisah kalimat dalam array pada fungsi sebelumnya berdasarkan beberapa *separator* yang telah kami tentukan dan mengembalikan array of string berupa nama tag dan simbol.

```
def handleString(arr)
```

fungsi yang menggantikan keberadaan string yang tidak perlu yaitu kata – kata diantara tag tertentu menjadi suatu variabel dan dan mengembalikan array of string berupa nama tag dan simbol.



```
def handlePetik(arr)
```

fungsi yang menggantikan keberadaan petik yang tidak perlu menjadi suatu variabel dan dan mengembalikan array of string berupa nama tag dan simbol.

```
def handleComment(arr)
```

fungsi yang menggantikan keberadaan komentar yang tidak perlu menjadi suatu variabel dan dan mengembalikan array of string berupa nama tag dan simbol.

```
def handleComment(arr)
```

fungsi yang menggantikan keberadaan string yang tidak perlu menjadi suatu variabel dan dan mengembalikan array of string berupa nama tag dan simbol.

```
def getGrammar(processedSyntax)
```

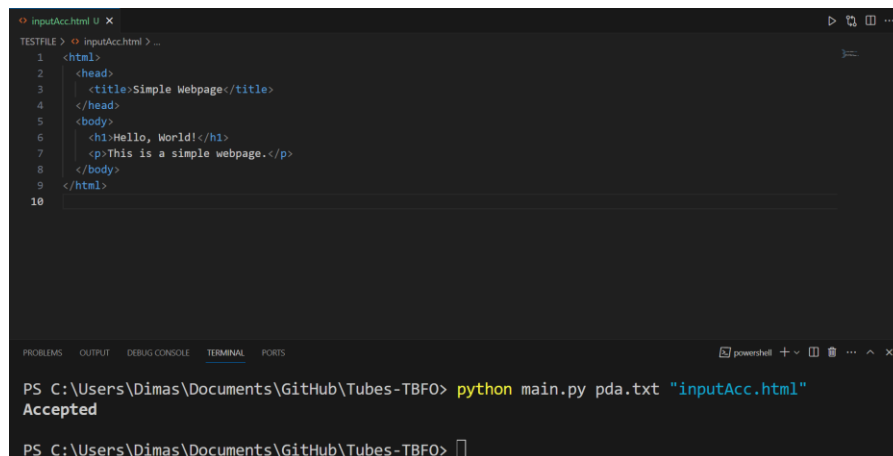
fungsi yang membaca array of string berupa nama tag dan simbol dan kembaliannya adalah suatu string of symbol yang menjadi Grammar yang akan diproses sesuai rule.



**Gambar 4.1. Antarmuka Program**

Gambar diatas merupakan antarmuka di terminal ketika menjalankan program kami. Sesuai pada spesifikasi maka ketika menjalankan program pastinya dimulai dengan “python”, lalu program utama yaitu “main.py” diikuti file pda “pda.txt” dan diakhiri nama dari file html yang akan dicek diberi tanda petik “inputAcc.html”. Pada program kami tidak handle salah input jadi pastikan ketika akan memulai program nama dari tiap file harus benar.

## PENGUJIAN

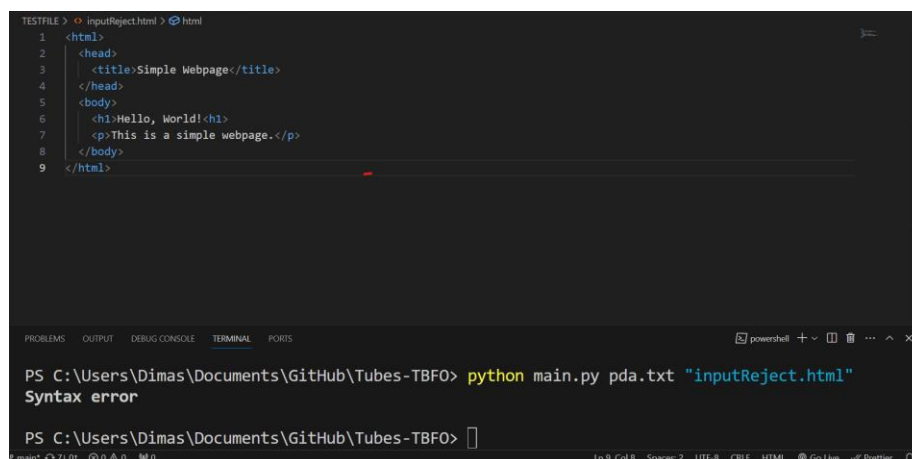


```
inputAcc.html U X
TESTFILE > inputAcc.html > ...
1 <html>
2 <head>
3 <title>Simple Webpage</title>
4 </head>
5 <body>
6 <h1>Hello, World!</h1>
7 <p>This is a simple webpage.</p>
8 </body>
9 </html>
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "inputAcc.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

Gambar 4.2. Pengujian 1 “inputAcc.html”

Ketika file “inputAcc.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid yaitu tidak memiliki kesalahan baik urutan struktur file maupun penulisan *syntax* didalamnya.

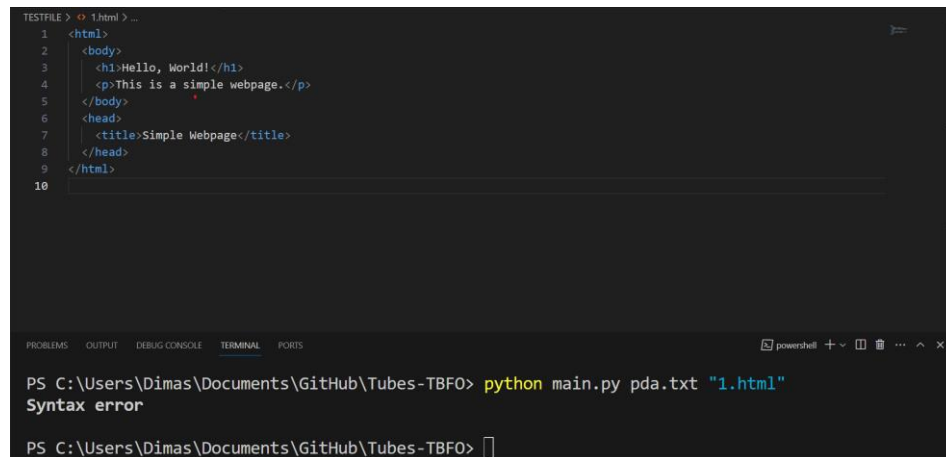


```
inputReject.html > html
1 <html>
2 <head>
3 <title>Simple Webpage</title>
4 </head>
5 <body>
6 <h1>Hello, World!<h1>
7 <p>This is a simple webpage.</p>
8 </body>
9 </html>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "inputReject.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

Gambar 4.3. Pengujian 2 “inputReject.html”

Ketika file “inputReject.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan penulisan *syntax* didalamnya. Kesalahan file tersebut terdapat pada *line* nomor 7 pada colse tag <h1> yang tidak memiliki “/” yang berarti pada stacknya “h1” dan dibawahnya tidak akan pernah di *pop* sehingga tidak pernah menuju keadaan *empty stack*.

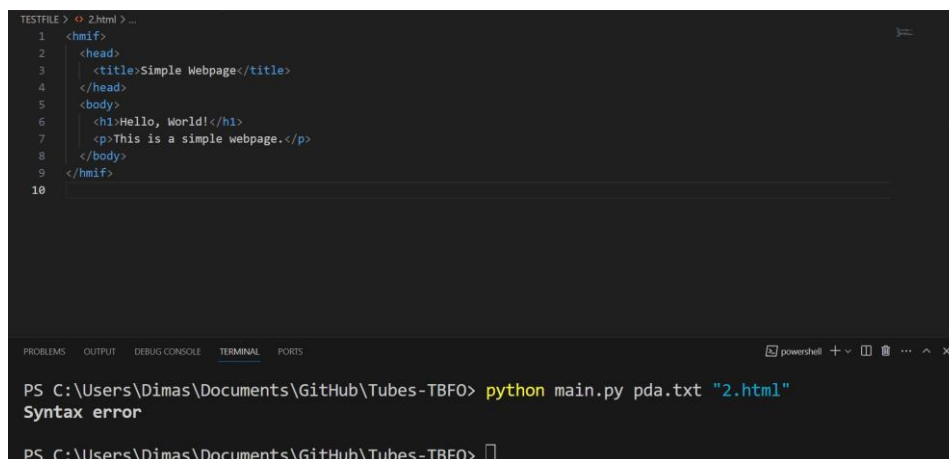


```
TESTFILE > 1.html > ...
1 <html>
2 <body>
3   <h1>Hello, World!</h1>
4   <p>This is a simple webpage.</p>
5 </body>
6 <head>
7   <title>Simple Webpage</title>
8 </head>
9 </html>
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "1.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.4. Pengujian 3 “1.html”**

Ketika file “1.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan penulisan *syntax* didalamnya. Kesalahan file tersebut terdapat pada urutan penulisan tag, terletak pada *line* nomor 2 - 8 dimana tidak seharusnya tag `body` berada terlebih dahulu atau diatas dari tag `head`. Hal ini menyalahi aturan PDA yang kami buat dimana seharusnya setelah tag `<html>` hanya boleh diikuti oleh tag `<body>` atau komentar.

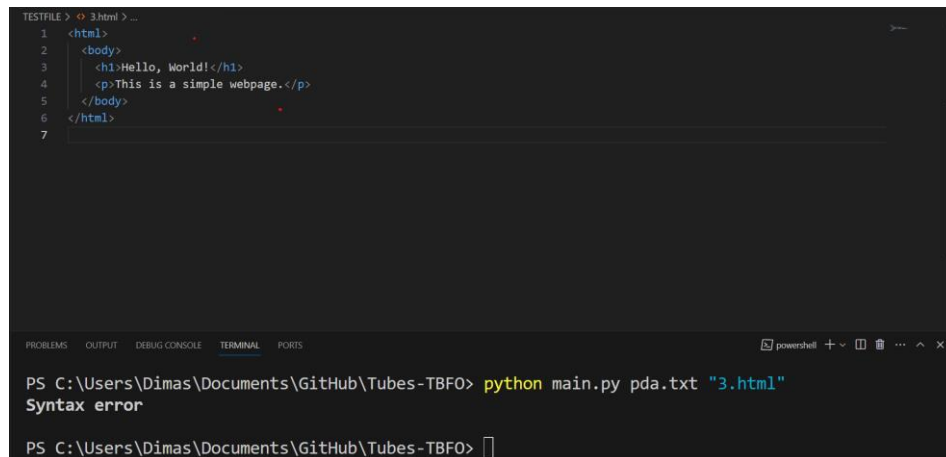


```
TESTFILE > 2.html > ...
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4 </head>
5 <body>
6   <h1>Hello, World!</h1>
7   <p>This is a simple webpage.</p>
8 </body>
9 </html>
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "2.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.5. Pengujian 4 “2.html”**

Ketika file “2.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan penulisan *syntax* didalamnya. Kesalahan file tersebut terdapat pada tag `<html>` dan `</html>`, kedua tag tersebut bukan merupakan tag yang valid yang bisa diterima oleh PDA tugas pemrograman ini. Hal lainnya yaitu pada PDA kami file hanya boleh dan harus diawali dengan tag `<html>` atau komentar.

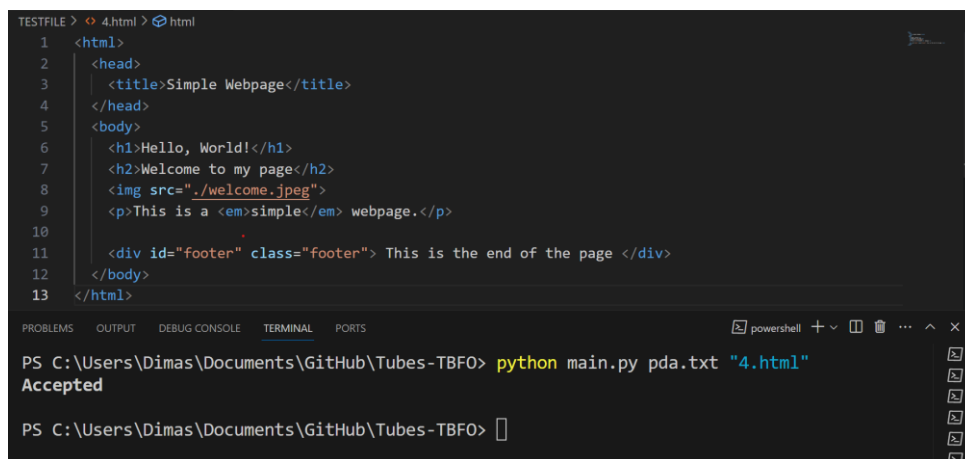


```
TESTFILE > 3.html > ...
1 <html>
2 <body>
3   <h1>Hello, World!</h1>
4   <p>This is a simple webpage.</p>
5 </body>
6 </html>
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "3.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.6. Pengujian 5 “3.html”**

Ketika file “3.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan penulisan *syntax* didalamnya. Kesalahan file tersebut terdapat pada bentuk file itu sendiri dimana file tersebut tidak memiliki `<body> </body>` setelah tag `</head>` yang “body” sendiri merupakan bagian suatu file HTML. Pada PDA kamipun setelah `</head>` hanya boleh dan harus diikuti dengan `<body>` atau komentar.

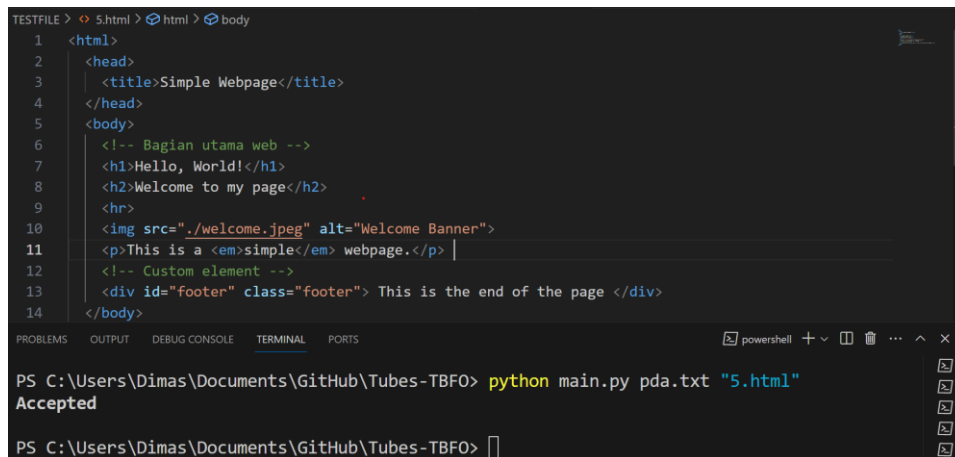


```
TESTFILE > 4.html > html
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4 </head>
5 <body>
6   <h1>Hello, World!</h1>
7   <h2>Welcome to my page</h2>
8   <img src= \"./welcome.jpeg\" >
9   <p>This is a <em>simple</em> webpage.</p>
10
11   <div id= \"footer\" class= \"footer\" > This is the end of the page </div>
12 </body>
13 </html>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "4.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.7. Pengujian 6 “4.html”**

Ketika file “4.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid baik dari segi penulisan *syntax* maupun urutannya. Sedikit catatan adalah dalam spesifikasi untuk file “4.html” ini petik yang digunakan adalah “ ” “ ” yang berbeda dari seharusnya sehingga dalam pengujiannya kami mengganti petik tersebut menjadi yang seharusnya yaitu “ ” “ ”.

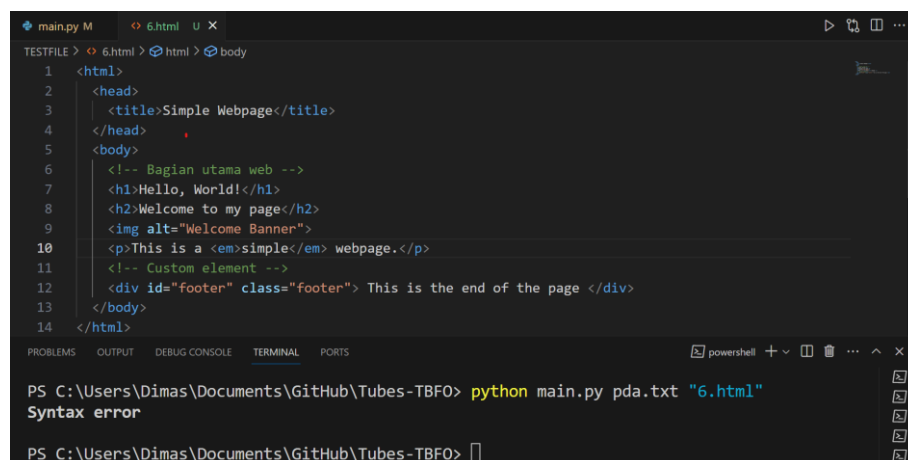


```
TESTFILE > 5.html > html > body
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4 </head>
5 <body>
6   <!-- Bagian utama web -->
7   <h1>Hello, World!</h1>
8   <h2>Welcome to my page</h2>
9   <hr>
10  
11  <p>This is a <em>simple</em> webpage.</p>
12  <!-- Custom element -->
13  <div id="footer" class="footer"> This is the end of the page </div>
14 </body>
15 </html>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "5.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.8. Pengujian 7 “5.html”**

Ketika file “5.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid baik dari segi penulisan *syntax* maupun urutannya. Sedikit catatan adalah dalam spesifikasi untuk file “4.html” ini petik yang digunakan adalah “ ” yang berbeda dari seharusnya sehingga dalam pengujiannya kami mengganti petik tersebut menjadi yang seharusnya yaitu “ ”.

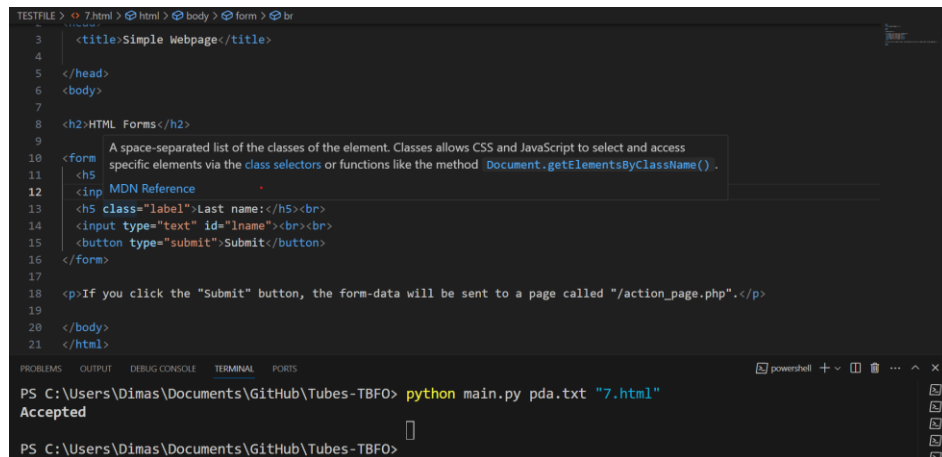


```
main.py M 6.html
TESTFILE > 6.html > html > body
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4 </head>
5 <body>
6   <!-- Bagian utama web -->
7   <h1>Hello, World!</h1>
8   <h2>Welcome to my page</h2>
9   <img alt="Welcome Banner">
10  <p>This is a <em>simple</em> webpage.</p>
11  <!-- Custom element -->
12  <div id="footer" class="footer"> This is the end of the page </div>
13 </body>
14 </html>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "6.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.9. Pengujian 8 “6.html”**

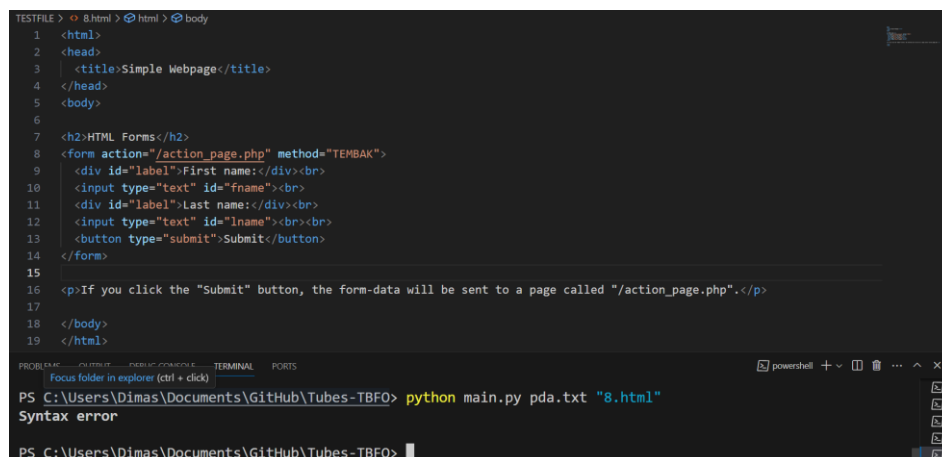
Ketika file “6.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan penulisan *syntax* didalamnya. Kesalahan file tersebut terdapat pada *line* 9 penulisan tag <img> yang seharusnya memiliki atribut src tetapi tidak dimiliki. Pada PDA kami pun tag <img> harus memiliki src yang valid.



```
TESTFILE > 7.html > html > body > form > br
1 <html>
2 <head>
3 <title>Simple Webpage</title>
4 </head>
5 <body>
6
7 <h2>HTML Forms</h2>
8
9 A space-separated list of the classes of the element. Classes allows CSS and JavaScript to select and access
10 specific elements via the class selectors or functions like the method Document.getElementsByClassName().
11 <h5>
12 <input type="text" id="lname"><br><br>
13 <input type="text" id="lname"><br><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 </body>
20 </html>
21
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "7.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

Gambar 4.10. Pengujian 9 “7.html”

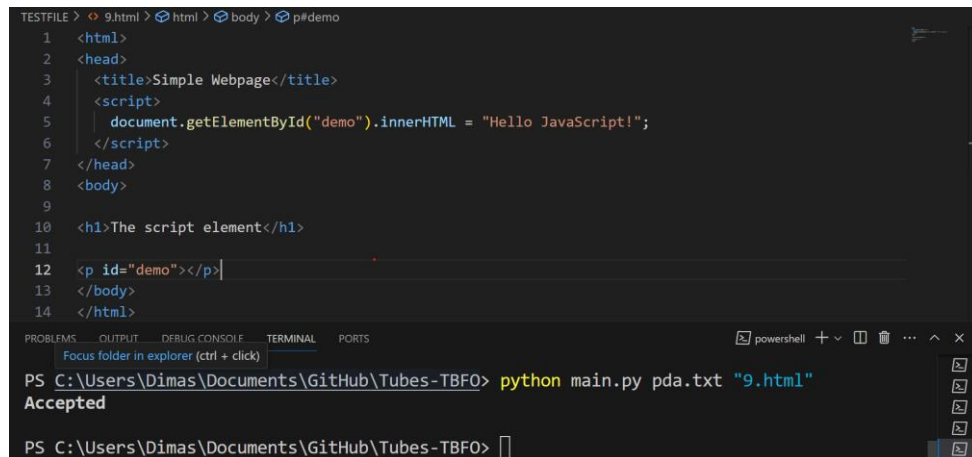
Ketika file “7.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid yaitu tidak memiliki kesalahan baik urutan struktur file maupun penulisan *syntax* didalamnya.



```
TESTFILE > 8.html > html > body
1 <html>
2 <head>
3 <title>Simple Webpage</title>
4 </head>
5 <body>
6
7 <h2>HTML Forms</h2>
8 <form action="/action_page.php" method="TEMBAK">
9 <div id="label">First name:</div><br>
10 <input type="text" id="fname"><br>
11 <div id="label">Last name:</div><br>
12 <input type="text" id="lname"><br><br>
13 <button type="submit">Submit</button>
14 </form>
15
16 <p>If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".</p>
17
18 </body>
19 </html>
20
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "8.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

Gambar 4.11. Pengujian 10 “8.html”

Ketika file “8.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan tipe dari *attribute* didalamnya. Kesalahan file tersebut terdapat pada *line* 8 penulisan tipe pada atribut *method* yaitu “TEMLAK” yang bukan merupakan salah satu tipe atribut *method* yang membuat file tidak valid . Pada PDA kamipun tag <form> harus memiliki tipe *method* yang valid.

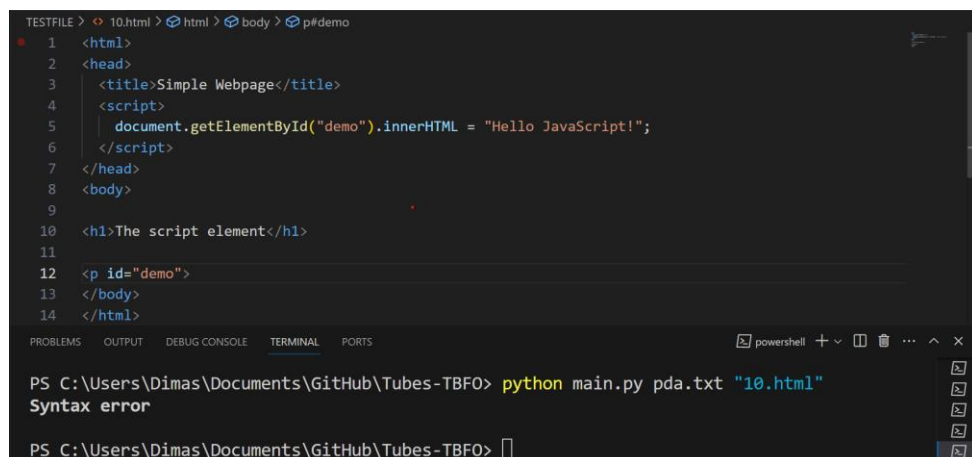


```
TESTFILE > 9.html > html > body > p#demo
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4   <script>
5     document.getElementById("demo").innerHTML = "Hello JavaScript!";
6   </script>
7 </head>
8 <body>
9
10  <h1>The script element</h1>
11
12  <p id="demo"></p>
13 </body>
14 </html>
```

```
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBF0> python main.py pda.txt "9.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBF0>
```

**Gambar 4.12. Pengujian 11 “9.html”**

Ketika file “9.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid yaitu tidak memiliki kesalahan baik urutan struktur file maupun penulisan *syntax* didalamnya.

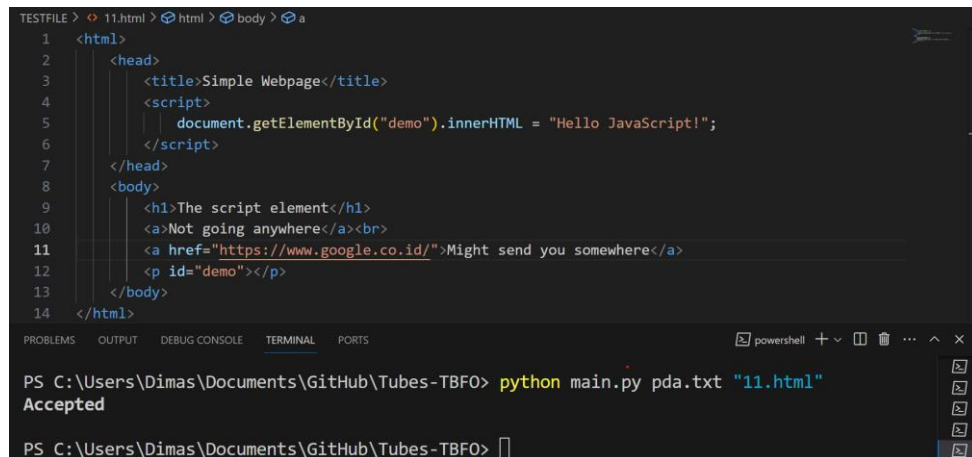


```
TESTFILE > 10.html > html > body > p#demo
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4   <script>
5     document.getElementById("demo").innerHTML = "Hello JavaScript!";
6   </script>
7 </head>
8 <body>
9
10  <h1>The script element</h1>
11
12  <p id="demo">
13 </body>
14 </html>
```

```
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBF0> python main.py pda.txt "10.html"
Syntax error
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBF0>
```

**Gambar 4.13. Pengujian 12 “10.html”**

Ketika file “10.html” dites didapati hasil yang sesuai yaitu **Syntax error** karena memang file HTML tersebut tidak valid yaitu memiliki kesalahan penulisan *syntax* didalamnya. Kesalahan file tersebut terdapat pada *line* 12 penulisan tag <p> yang yang tidak diikuti dengan closing tag </p>, hal ini membuat keadaan tidak akan pernah menuju empty stack. Pada PDA kamipun tag <p> harus memiliki closing tag </p>.



```
TESTFILE > 11.html > html > body > a
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    <p id="demo"></p>
13  </body>
14 </html>
```

```
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "11.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.14. Pengujian 13 “11.html”**

Ketika file “11.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid yaitu tidak memiliki kesalahan baik urutan struktur file maupun penulisan *syntax* didalamnya.



```
TESTFILE > test.html > html > body > img#ijsif
1 <html>
2   <head>
3     <title>Simple Webpage</title>
4   </head>
5   <body>
6     <h1>Hello, World!</h1>
7     <h2>Welcome to my page</h2>
8     <p><strong></strong> <em></em> a webpage.</p>
9     <p><b></b></p>
10    <p><strong></strong></p>
11    <strong></strong>
12    <em> </em>
13    <small></small>
14    <form method="GET" id="ijsif">
```

```
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO> python main.py pda.txt "test.html"
Accepted
PS C:\Users\Dimas\Documents\GitHub\Tubes-TBFO>
```

**Gambar 4.15. Pengujian 14 “test.html”**

Ketika file “test.html” dites didapati hasil yang sesuai yaitu **Accepted** karena memang file HTML tersebut valid yaitu tidak memiliki kesalahan baik urutan struktur file maupun penulisan *syntax* didalamnya.



## **BAB V**

### **PENUTUP**

#### ***KESIMPULAN***

Tugas pembuatan HTML *Checker* dengan Pushdown Automata (PDA) merupakan suatu tantangan yang menarik dan relevan dalam mata kuliah Teori Bahasa Formal dan Otomata. Penerapan PDA dengan *empty stack* memberikan kontribusi signifikan terhadap peningkatan akurasi dan keefektifan sistem dalam memeriksa ketepatan suatu file HTML. Penggunaan aturan PDA yang tepat dan terstruktur sangat signifikan terhadap hasil yang optimal.

#### ***SARAN***

Untuk selanjutnya manajemen waktu harus dapat dijaga dengan baik karena pengerjaan masih terkesan terburu-buru sehingga kurang tereksekusi maksimal. Berikutnya dalam memahami spesifikasi kami bisa lebih baik dalam memahami spesifikasi tugas terlebih lagi catatan tambahan sehingga pada proses pembuatannya kami bisa lebih efektif dan efisien. Selain itu pembagian kerja juga harus lebih merata lagi karena beban kerja yang sekarang masih tidak seimbang sehingga tidak tereksekusi secara efisien. Dalam hal komunikasi juga masih kurang optimal sehingga terdapat miskomunikasi yang menghambat pengerjaan.

## DAFTAR PUSTAKA

“Introduction of Pushdown Automata”, geeksforgeeks.com. 24 November 2023.

<https://www.geeksforgeeks.org/introduction-of-pushdown-automata/>

“Push Down Automata”, medium.com. 23 November 2023.

<https://medium.com/@zahma.abi/push-down-automata-a004841adde1>

“Pushdown Automata Introduction”, tutorialspoint.com. 20 November 2023.

[https://www.tutorialspoint.com/automata\\_theory/pushdown\\_automata\\_introduction.htm](https://www.tutorialspoint.com/automata_theory/pushdown_automata_introduction.htm)

“Pushdown Automata(PDA)”, javapoint.com. 20 November 2023.

<https://www.javatpoint.com/pushdown-automata>

“HTML Tutorial”, w3schools.com. 17 November 2023. <https://www.w3schools.com/html/>

“HTML Tags”, javapoint.com. 16 November 2023. <https://www.javatpoint.com/html-tags>

*Github* <https://github.com/keanugonza/Tubes-TBFO.git>

*Diagram*

<https://drive.google.com/file/d/16NI0Vj91VI0BAqI6511A6w3t6t67EmwU/view?usp=sharing>

[g](#)