

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata
HTML *Checker* dengan Pushdown Automata (PDA)

TUGAS PEMROGRAMAN IF 2124 TEORI BAHASA FORMAL DAN OTOMATA HTML CHECKER dengan PDA

DISUSUN OLEH:

13522029 - Ignatius Jhon Hezkiel Chan

13522043 - Daniel Mulia Putra Manurung

13522054 - Benjamin Sihombing



**Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung**

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, `` 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.

BAB II

Landasan Teori

2.1. HTML

HyperText Markup Language, atau HTML, adalah komponen fundamental dari World Wide Web. Ini menetapkan struktur dan makna konten web. Presentasi dan fungsionalitas halaman web biasanya dijelaskan oleh teknologi lain selain HTML. (CSS, JavaScript).

Tautan yang menghubungkan halaman web bersama-sama, baik dalam satu situs web atau di antara situs web, disebut sebagai "hipertekst". Salah satu komponen penting dari web adalah link. Anda dapat berpartisipasi aktif di World Wide Web dengan menerbitkan konten online dan membuat tautan ke halaman orang lain.

"Tags" yang mengelilingi nama elemen HTML dengan "<" dan ">" berfungsi untuk membedakannya dari teks lain di halaman. Nama elemen dalam tag tidak terpengaruh oleh kapitalisasi. Artinya, dapat ditulis dalam kombinasi huruf rendah dan huruf besar. Tag <title>, misalnya, dapat dinyatakan sebagai <Title>, <TITLE>, atau kombinasi karakter lainnya. Namun demikian, menulis tag dalam lowercase adalah prosedur standar dan disarankan.

2.2. Push Down Automata (PDA)

Pekerjaan teoritis pada komputasi mesin menggunakan pushdown automata. Mereka tidak begitu kuat seperti mesin Turing, tetapi mereka masih lebih kuat daripada komputer-komputer yang terbatas. Sementara non-deterministic pushdown automata dapat mengenali setiap bahasa bebas konteks, otomatis deterministik sering digunakan dalam desain parser karena mereka bisa mengenali bahasa bebas kontekstual.

PDA secara formal didefinisikan sebagai 7-tuple dengan aturan sebagai berikut:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z, F) \text{ where}$$

- Q is a finite set of *states*
- Σ is a finite set which is called the *input alphabet*
- Γ is a finite set which is called the *stack alphabet*
- δ is a finite subset of $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$, the *transition relation*
- $q_0 \in Q$ is the *start state*
- $Z \in \Gamma$ is the *initial stack symbol*
- $F \subseteq Q$ is the set of *accepting states*

δ adalah sebuah fungsi atau rule yang digunakan untuk menentukan perpindahan state dan flow dari PDA sesuai dengan parameter input.

2.3. Pengaplikasian PDA pada syntax HTML

Push Down Automata dapat diaplikasikan untuk membaca syntax HTML dan menyatakan apakah syntax yang sedang dibaca tersebut benar atau tidak. Dalam hal ini tupel PDA akan memiliki komponen sebagai berikut:

- Q berisi seluruh state yang dimiliki berdasarkan rules PDA.

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML *Checker* dengan Pushdown Automata (PDA)

- Σ berisi seluruh kemungkinan input, yang pada kasus syntax HTML berisi pembuka tag HTML seperti <head, <body, dan penutup seperti >.
- Γ berisi seluruh elemen stack
- δ berisi seluruh rules PDA yang ditetapkan.
- q_0 adalah state awal

Dengan PDA tersebut dapat dihasilkan sebuah

BAB III

Hasil PDA

3.1. Total States

ATTRIBUTE HEAD BODY CONTENT HTML HTMLH HTMLHB START FINISH TABLE
ERROR

3.2. Input Word Symbols

```
id="" class="" style="" href="" rel="" src="" alt=""  
type="submit" type="reset" type="button" type="text"  
type="password" type="email" type="number" type="checkbox"  
action="" method="GET" method="POST" <!--> X <title </title>  
<script </script> <link <br <em </em> <b </b> <abbr </abbr>  
<strong </strong> <small </small> <hr <div </div> <a </a> <h1  
</h1> <h2 </h2> <h3 </h3> <h4 </h4> <h5 </h5> <h6 </h6> <p </p>  
<img <button </button> <form </form> <input <table </table> <tr  
</tr> <th </th> <td
```

3.3. Stack Symbols

```
<X> </linkH> </linkB> </validLinkH> </validLinkB> </a> </img>  
</scriptH> </scriptB> </validImg> </button> </input> </form>  
</comment> </body> </head> </table> </tr> </title> </brB> </emB>  
</bB> </abbrB> </strongB> </smallB> </hrB> </div> </h1> </h2>  
</h3> </h4> </h5> </h6> </p> </typeButton> </methodForm>  
</typeInput> </brC> </emC> </bC> </abbrC> </strongC> </smallC>  
</smallC> </hrC> </th> </td> <Z> </html>
```

3.4. Starting State

START

3.5. Starting Stack

<Z>

3.6. Accepting States

FINISH

3.7. List of Productions

```
d(ATRIBUT, id="", <X>) = (ATRIBUT, <X>)  
d(ATRIBUT, class="", <X>) = (ATRIBUT, <X>)  
d(ATRIBUT, style="", <X>) = (ATRIBUT, <X>)  
d(ATRIBUT, href="", </linkH>) = (ATRIBUT, </linkH>)
```

```

d(ATRIBUT, href="", </linkB>) = (ATRIBUT, </linkB>)
d(ATRIBUT, href="", </validLinkH>) = (ATRIBUT, </validLinkH>)
d(ATRIBUT, href="", </validLinkB>) = (ATRIBUT, </validLinkB>)
d(ATRIBUT, href="", </a>) = (ATRIBUT, </a>)
d(ATRIBUT, rel="", </linkH>) = (ATRIBUT, </validLinkH>)
d(ATRIBUT, rel="", </linkB>) = (ATRIBUT, </validLinkB>)
d(ATRIBUT, src="", </img>) = (ATRIBUT, </validImg>)
d(ATRIBUT, src="", </scriptH>) = (ATRIBUT, </scriptH>)
d(ATRIBUT, src="", </scriptB>) = (ATRIBUT, </scriptB>)
d(ATRIBUT, alt="", </img>) = (ATRIBUT, </img>)
d(ATRIBUT, alt="", </validImg>) = (ATRIBUT, </validImg>)
d(ATRIBUT, type="submit", </button>) = (ATRIBUT, </typeButton>)
d(ATRIBUT, type="reset", </button>) = (ATRIBUT, </typeButton>)
d(ATRIBUT, type="button", </button>) = (ATRIBUT, </typeButton>)
d(ATRIBUT, type="text", </input>) = (ATRIBUT, </typeInput>)
d(ATRIBUT, type="password", </input>) = (ATRIBUT, </typeInput>)
d(ATRIBUT, type="email", </input>) = (ATRIBUT, </typeInput>)
d(ATRIBUT, type="number", </input>) = (ATRIBUT, </typeInput>)
d(ATRIBUT, type="checkbox", </input>) = (ATRIBUT, </typeInput>)
d(ATRIBUT, action="", </form>) = (ATRIBUT, </form>)
d(ATRIBUT, method="GET", </form>) = (ATRIBUT, </methodForm>)
d(ATRIBUT, method="POST", </form>) = (ATRIBUT, </methodForm>)
d(HEAD, <!--, <X>) = (HEAD, </comment><X>)
d(HEAD, >, </comment>) = (HEAD, e)
d(BODY, <!--, <X>) = (BODY, </comment><X>)
d(BODY, >, </comment>) = (BODY, e)
d(CONTENT, <!--, <X>) = (CONTENT, </comment><X>)
d(CONTENT, >, </comment>) = (CONTENT, e)
d(HTML, <!--, <X>) = (HTML, </comment><X>)
d(HTML, >, </comment>) = (HTML, e)
d(HTMLH, <!--, <X>) = (HTMLH, </comment><X>)
d(HTMLH, >, </comment>) = (HTMLH, e)
d(HTMLHB, <!--, <X>) = (HTMLHB, </comment><X>)
d(HTMLHB, >, </comment>) = (HTMLHB, e)
d(START, <!--, <X>) = (START, </comment><X>)
d(START, >, </comment>) = (START, e)
d(FINISH, <!--, <X>) = (FINISH, </comment><X>)
d(FINISH, >, </comment>) = (FINISH, e)
d(TABLE, <!--, <X>) = (TABLE, </comment><X>)
d(TABLE, >, </comment>) = (TABLE, e)
d(HEAD, X, <X>) = (HEAD, <X>)
d(BODY, X, <X>) = (BODY, <X>)
d(TABLE, X, <X>) = (TABLE, <X>)
d(CONTENT, X, <X>) = (CONTENT, <X>)
d(BODY, X, </body>) = (ERROR, e)
d(HEAD, X, </head>) = (ERROR, e)

```

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata
HTML *Checker* dengan Pushdown Automata (PDA)

```
d(TABLE, X, </table>) = (ERROR, e)
d(TABLE, X, </tr>) = (ERROR, e)
d(HEAD, <title, </head>) = (ATRIBUT, </title></head>)
d(ATRIBUT, >, </title>) = (HEAD, </title>)
d(HEAD, </title>, </title>) = (HEAD, e)
d(HEAD, <script, </head>) = (ATRIBUT, </scriptH></head>)
d(ATRIBUT, >, </scriptH>) = (HEAD, </scriptH>)
d(HEAD, </script>, </scriptH>) = (HEAD, e)
d(HEAD, <link, </head>) = (ATRIBUT, </linkH></head>)
d(ATRIBUT, >, </validLinkH>) = (HEAD, e)
d(BODY, <br, <X>) = (ATRIBUT, </brB><X>)
d(ATRIBUT, >, </brB>) = (BODY, e)
d(BODY, <em, <X>) = (ATRIBUT, </emB><X>)
d(ATRIBUT, >, </emB>) = (CONTENT, </emB>)
d(CONTENT, </em>, </emB>) = (BODY, e)
d(BODY, <b, <X>) = (ATRIBUT, </bB><X>)
d(ATRIBUT, >, </bB>) = (CONTENT, </bB>)
d(CONTENT, </b>, </bB>) = (BODY, e)
d(BODY, <abbr, <X>) = (ATRIBUT, </abbrB><X>)
d(ATRIBUT, >, </abbrB>) = (CONTENT, </abbrB>)
d(CONTENT, </abbr>, </abbrB>) = (BODY, e)
d(BODY, <strong, <X>) = (ATRIBUT, </strongB><X>)
d(ATRIBUT, >, </strongB>) = (CONTENT, </strongB>)
d(CONTENT, </strong>, </strongB>) = (BODY, e)
d(BODY, <small, <X>) = (ATRIBUT, </smallB><X>)
d(ATRIBUT, >, </smallB>) = (CONTENT, </smallB>)
d(CONTENT, </small>, </smallB>) = (BODY, e)
d(BODY, <hr, <X>) = (ATRIBUT, </hrB><X>)
d(ATRIBUT, >, </hrB>) = (BODY, e)
d(BODY, <div, <X>) = (ATRIBUT, </div><X>)
d(ATRIBUT, >, </div>) = (BODY, </div>)
d(BODY, </div>, </div>) = (BODY, e)
d(BODY, <a, <X>) = (ATRIBUT, </a><X>)
d(ATRIBUT, >, </a>) = (CONTENT, </a>)
d(CONTENT, </a>, </a>) = (BODY, e)
d(BODY, <h1, <X>) = (ATRIBUT, </h1><X>)
d(ATRIBUT, >, </h1>) = (CONTENT, </h1>)
d(CONTENT, </h1>, </h1>) = (BODY, e)
d(BODY, <h2, <X>) = (ATRIBUT, </h2><X>)
d(ATRIBUT, >, </h2>) = (CONTENT, </h2>)
d(CONTENT, </h2>, </h2>) = (BODY, e)
d(BODY, <h3, <X>) = (ATRIBUT, </h3><X>)
d(ATRIBUT, >, </h3>) = (CONTENT, </h3>)
d(CONTENT, </h3>, </h3>) = (BODY, e)
d(BODY, <h4, <X>) = (ATRIBUT, </h4><X>)
d(ATRIBUT, >, </h4>) = (CONTENT, </h4>)
```

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata
HTML *Checker* dengan Pushdown Automata (PDA)

```
d(CONTENT, </h4>, </h4>) = (BODY, e)
d(BODY, <h5, <X>) = (ATRIBUT, </h5><X>)
d(ATRIBUT, >, </h5>) = (CONTENT, </h5>)
d(CONTENT, </h5>, </h5>) = (BODY, e)
d(BODY, <h6, <X>) = (ATRIBUT, </h6><X>)
d(ATRIBUT, >, </h6>) = (CONTENT, </h6>)
d(CONTENT, </h6>, </h6>) = (BODY, e)
d(BODY, <p, <X>) = (ATRIBUT, </p><X>)
d(ATRIBUT, >, </p>) = (CONTENT, </p>)
d(CONTENT, </p>, </p>) = (BODY, e)
d(BODY, <link, <X>) = (ATRIBUT, </linkB><X>)
d(ATRIBUT, >, </validLinkB>) = (BODY, e)
d(BODY, <script, <X>) = (ATRIBUT, </scriptB><X>)
d(ATRIBUT, >, </scriptB>) = (CONTENT, </scriptB>)
d(CONTENT, </script>, </scriptB>) = (BODY, e)
d(BODY, <img, <X>) = (ATRIBUT, </img><X>)
d(ATRIBUT, >, </validImg>) = (BODY, e)
d(BODY, <button, <X>) = (ATRIBUT, </button><X>)
d(ATRIBUT, >, </typeButton>) = (CONTENT, </typeButton>)
d(CONTENT, </button>, </typeButton>) = (BODY, e)
d(BODY, <form, </form>) = (ERROR, e)
d(BODY, <form, <X>) = (ATRIBUT, </form><X>)
d(ATRIBUT, >, </form>) = (BODY, </form>)
d(ATRIBUT, >, </methodForm>) = (BODY, </methodForm>)
d(BODY, </form>, </methodForm>) = (BODY, e)
d(BODY, <input, <X>) = (ATRIBUT, </input><X>)
d(ATRIBUT, >, </typeInput>) = (BODY, e)
d(BODY, <table, <X>) = (ATRIBUT, </table><X>)
d(ATRIBUT, >, </table>) = (TABLE, </table>)
d(TABLE, </table>, </table>) = (BODY, e)
d(CONTENT, <br, <X>) = (ATRIBUT, </brC><X>)
d(ATRIBUT, >, </brC>) = (CONTENT, e)
d(CONTENT, <em, <X>) = (ATRIBUT, </emC><X>)
d(ATRIBUT, >, </emC>) = (CONTENT, </emC>)
d(CONTENT, </em>, </emC>) = (CONTENT, e)
d(CONTENT, <b, <X>) = (ATRIBUT, </bC><X>)
d(ATRIBUT, >, </bC>) = (CONTENT, </bC>)
d(CONTENT, </b>, </bC>) = (CONTENT, e)
d(CONTENT, <abbr, <X>) = (ATRIBUT, </abbrC><X>)
d(ATRIBUT, >, </abbrC>) = (CONTENT, </abbrC>)
d(CONTENT, </abbr>, </abbrC>) = (CONTENT, e)
d(CONTENT, <strong, <X>) = (, ATRIBU)
d(ATRIBUT, >, </strongC>) = (CONTENT, </strongC>)
d(CONTENT, </strong>, </strongC>) = (CONTENT, e)
d(CONTENT, <small, <X>) = (ATRIBUT, </smallC><X>)
d(ATRIBUT, >, </smallc>) = (CONTENT, </smallC>)
```

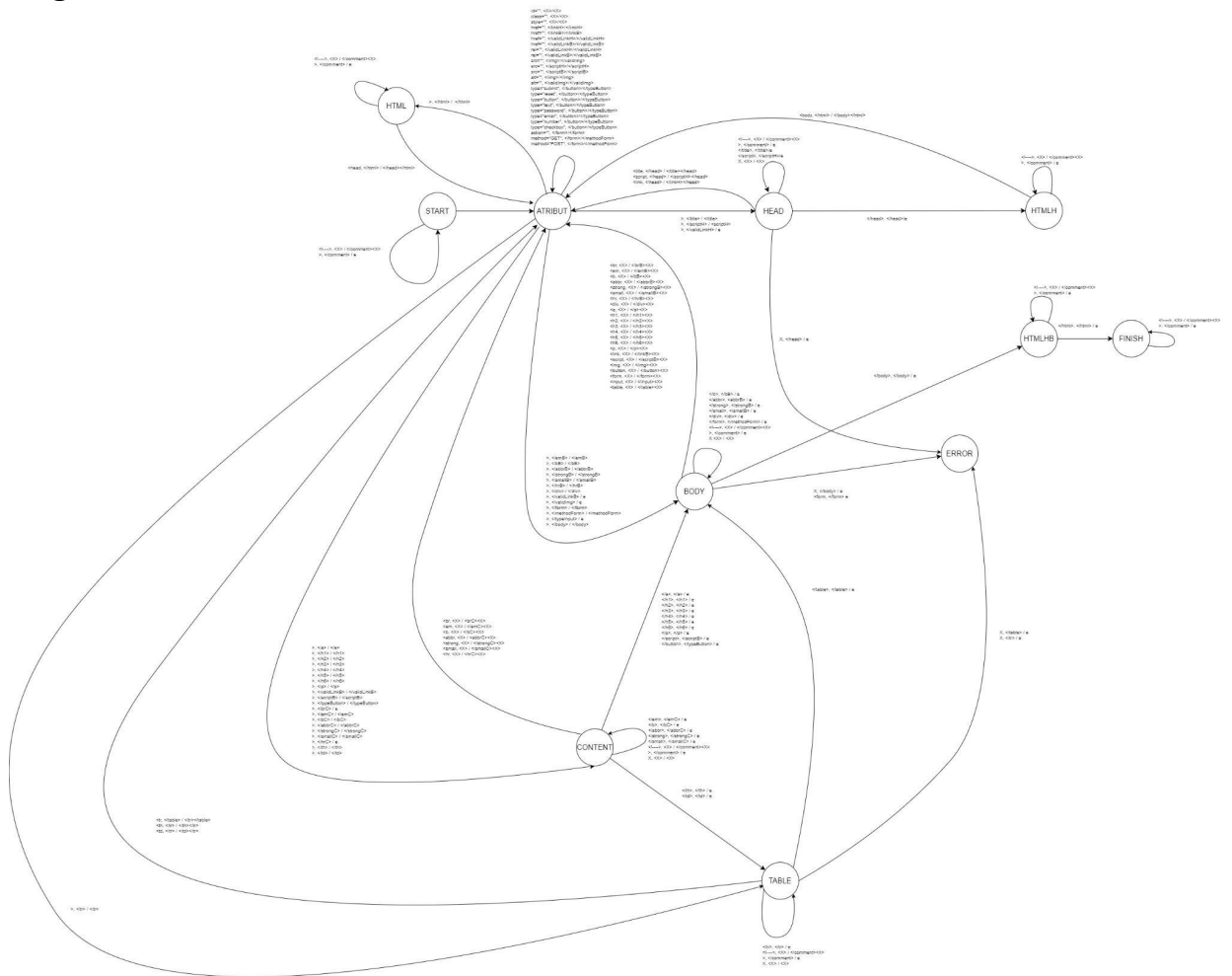


```
d(CONTENT, </small>, </smallC>) = (CONTENT, e)
d(CONTENT, <hr, <X>) = (ATRIBUT, </hrC><X>)
d(ATRIBUT, >, </hrC>) = (CONTENT, e)
d(TABLE, <tr, </table>) = (ATRIBUT, </tr></table>)
d(ATRIBUT, >, </tr>) = (TABLE, </tr>)
d(TABLE, </tr>, </tr>) = (TABLE, e)
d(TABLE, <th, </tr>) = (ATRIBUT, </th></tr>)
d(ATRIBUT, >, </th>) = (CONTENT, </th>)
d(CONTENT, </th>, </th>) = (TABLE, e)
d(TABLE, <td, </tr>) = (ATRIBUT, </td></tr>)
d(ATRIBUT, >, </td>) = (CONTENT, </td>)
d(CONTENT, </td>, </td>) = (TABLE, e)
d(START, <html, <Z>) = (ATRIBUT, </html><Z>)
d(ATRIBUT, >, </html>) = (HTML, </html>)
d(HTML, <head, </html>) = (ATRIBUT, </head></html>)
d(ATRIBUT, >, </head>) = (HEAD, </head>)
d(HEAD, </head>, </head>) = (HTMLH, e)
d(HTMLH, <body, </html>) = (ATRIBUT, </body></html>)
d(ATRIBUT, >, </body>) = (BODY, </body>)
d(BODY, </body>, </body>) = (HTMLHB, e)
d(HTMLHB, </html>, </html>) = (FINISH)
```

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

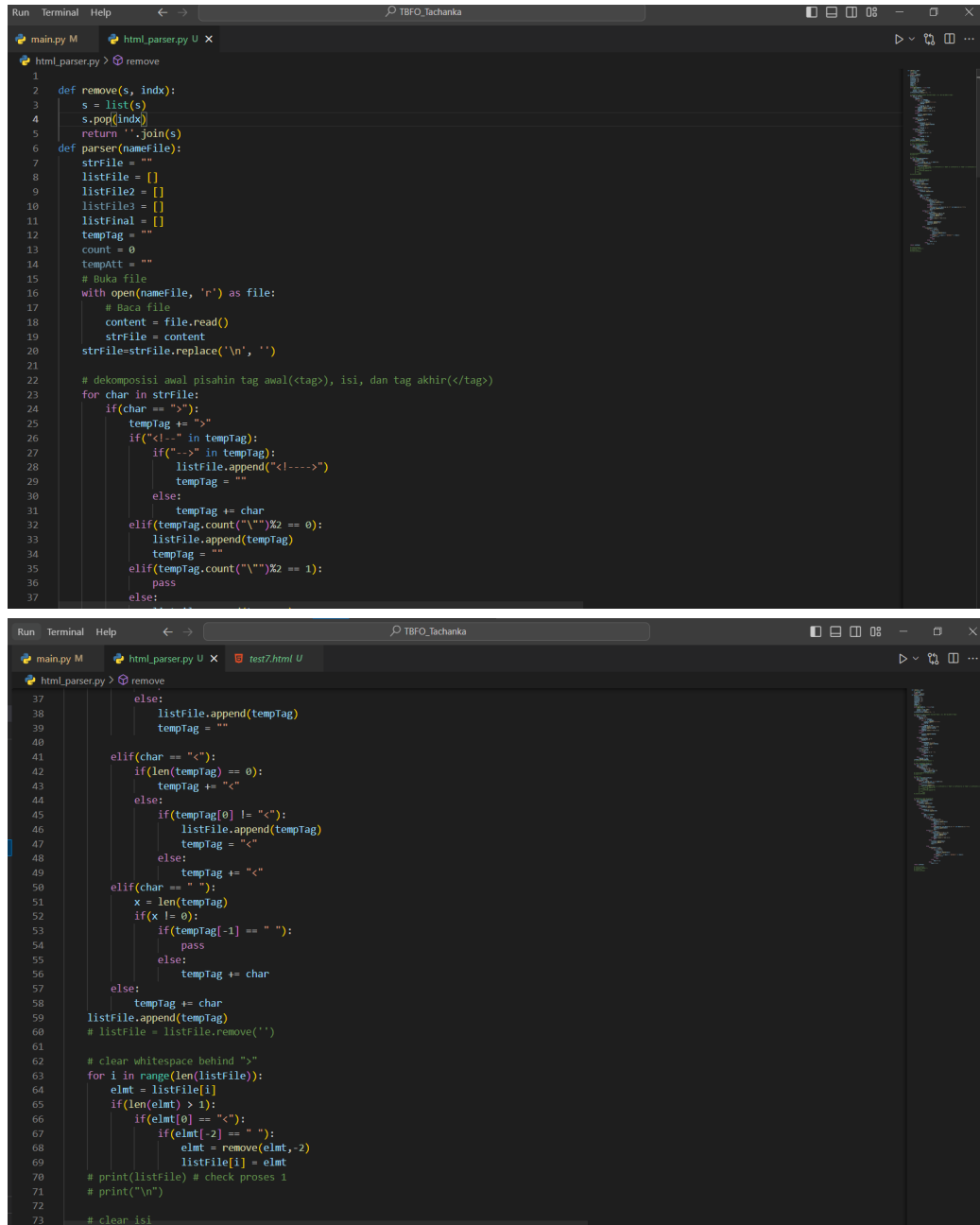
3.8. Diagram PDA



BAB IV

Implementasi dan Pengujian

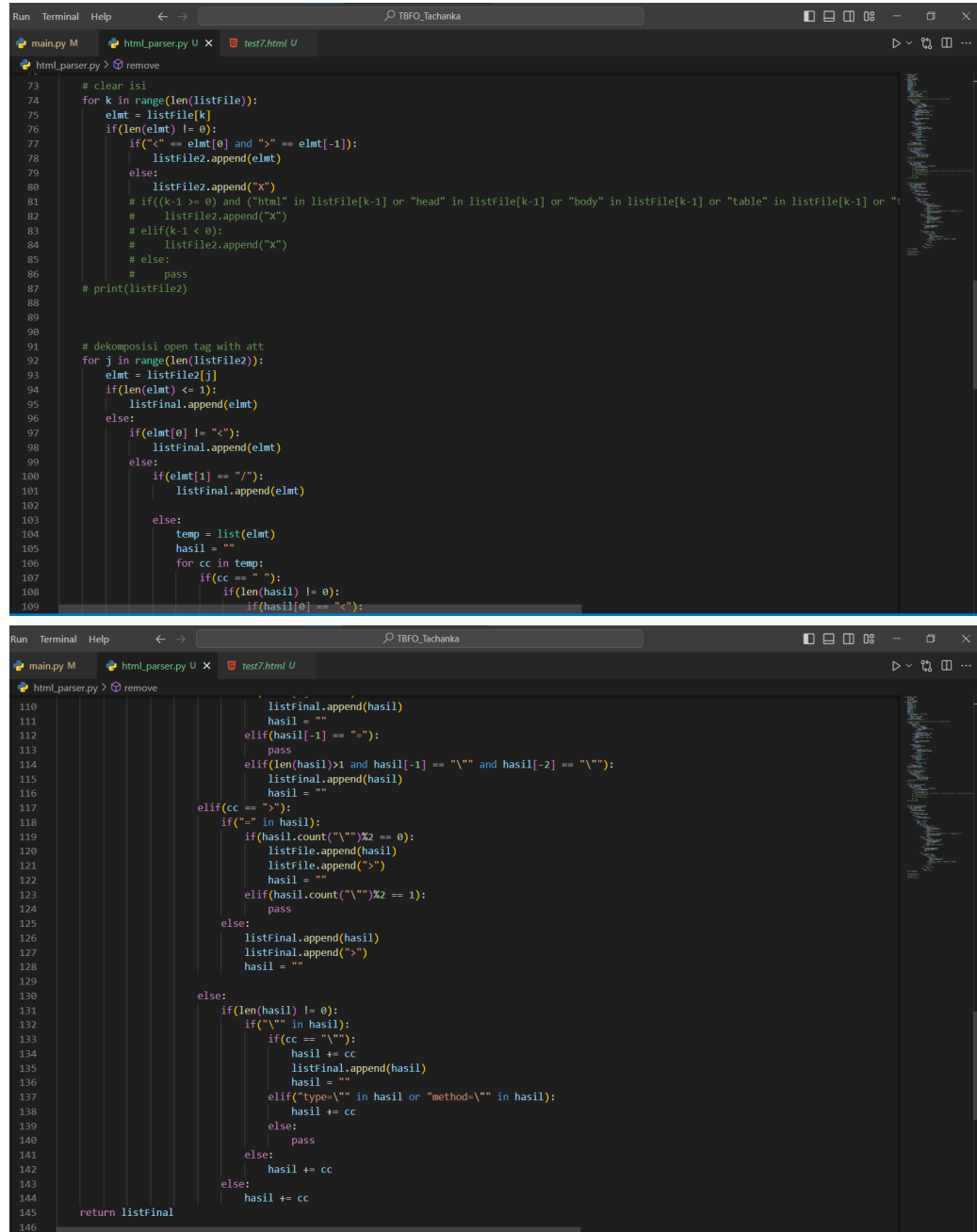
4.1. HTML Parser



```
1 def remove(s, indx):
2     s = list(s)
3     s.pop(indx)
4     return ''.join(s)
5
6 def parser(namefile):
7     strFile = ""
8     listFile = []
9     listFile2 = []
10    listFile3 = []
11    listFinal = []
12    tempTag = ""
13    count = 0
14    tempAtt = ""
15    # Buka file
16    with open(namefile, 'r') as file:
17        # Baca file
18        content = file.read()
19        strFile = content
20        strFile=strFile.replace('\n', '')
21
22    # dekomposisi awal pisahin tag awal(<tag>), isi, dan tag akhir(</tag>)
23    for char in strFile:
24        if(char == ">"):
25            tempTag += ">"
26            if("<!--" in tempTag):
27                if("-->" in tempTag):
28                    listFile.append("<!-->")
29                    tempTag = ""
30                else:
31                    tempTag += char
32            elif(tempTag.count("\")%2 == 0):
33                listFile.append(tempTag)
34                tempTag = ""
35            elif(tempTag.count("\")%2 == 1):
36                pass
37            else:
38                listFile.append(tempTag)
39                tempTag = ""
40
41        elif(char == "<"):
42            if(len(tempTag) == 0):
43                tempTag += "<"
44            else:
45                if(tempTag[0] != "<"):
46                    listFile.append(tempTag)
47                    tempTag = "<"
48                else:
49                    tempTag += "<"
50        elif(char == " "):
51            x = len(tempTag)
52            if(x != 0):
53                if(tempTag[-1] == " "):
54                    pass
55                else:
56                    tempTag += char
57        else:
58            tempTag += char
59    listFile.append(tempTag)
60    # listFile = listFile.remove('')
61
62    # clear whitespace behind ">"
63    for i in range(len(listFile)):
64        elmt = listFile[i]
65        if(len(elmt) > 1):
66            if(elmt[0] == "<"):
67                if(elmt[-2] == " "):
68                    elmt = remove(elmt,-2)
69                    listFile[i] = elmt
70    # print(listFile) # check proses 1
71    # print("\n")
72
73    # clear isi
```

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)



```
73 # clear isi
74 for k in range(len(listFile)):
75     elmt = listFile[k]
76     if(len(elmt) != 0):
77         if("<" == elmt[0] and ">" == elmt[-1]):
78             listFile2.append(elmt)
79         else:
80             listFile2.append("X")
81         # if((k-1 >= 0) and ("html" in listFile[k-1] or "head" in listFile[k-1] or "body" in listFile[k-1] or "table" in listFile[k-1] or "
82         # listFile2.append("X")
83         # elif(k-1 < 0):
84         # listFile2.append("X")
85         # else:
86         # pass
87 # print(listFile2)
88
89
90
91 # dekomposisi open tag with att
92 for j in range(len(listFile2)):
93     elmt = listFile2[j]
94     if(len(elmt) <= 1):
95         listFinal.append(elmt)
96     else:
97         if(elmt[0] != "<"):
98             listFinal.append(elmt)
99         else:
100             if(elmt[1] == "/"):
101                 listFinal.append(elmt)
102             else:
103                 temp = list(elmt)
104                 hasil = ""
105                 for cc in temp:
106                     if(cc == " "):
107                         if(len(hasil) != 0):
108                             if(hasil[0] == "<"):
109
110                     listFinal.append(hasil)
111                     hasil = ""
112                     elif(hasil[-1] == "-"):
113                         pass
114                     elif(len(hasil)>1 and hasil[-1] == "\"" and hasil[-2] == "\""):
115                         listFinal.append(hasil)
116                         hasil = ""
117                     elif(cc == ">"):
118                         if("=" in hasil):
119                             if(hasil.count("\\"">2 == 0):
120                                 listFile.append(hasil)
121                                 listFile.append(">")
122                                 hasil = ""
123                             elif(hasil.count("\\"">2 == 1):
124                                 pass
125                         else:
126                             listFinal.append(hasil)
127                             listFinal.append(">")
128                             hasil = ""
129                     else:
130                         if(len(hasil) != 0):
131                             if("\\"" in hasil):
132                                 if(cc == "\\""):
133                                     hasil += cc
134                                     listFinal.append(hasil)
135                                     hasil = ""
136                                 elif("type=\\"" in hasil or "method=\\"" in hasil):
137                                     hasil += cc
138                                 else:
139                                     pass
140                             else:
141                                 hasil += cc
142                         else:
143                             hasil += cc
144                     return listFinal
145
146
```

Fungsi remove(a, b)

- Berfungsi untuk menghapus karakter ke-b dari string a

Fungsi parser(a)

- Merubah file HTML menjadi token-token, dengan proses:
 1. Mendekomposisi HTML menjadi opentag (<tag>), isi, dan closetag (</tag>).
 2. Membersihkan white space.
 3. Mengubah string di dalam tag menjadi token X.
 4. Mendekomposisi opentag menjadi jenis opentag, atribut, dan >.

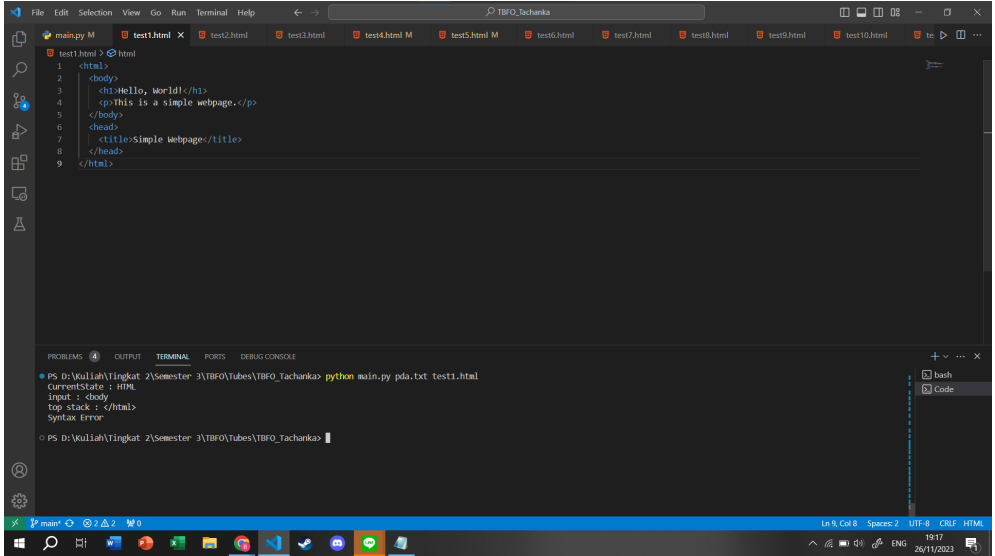
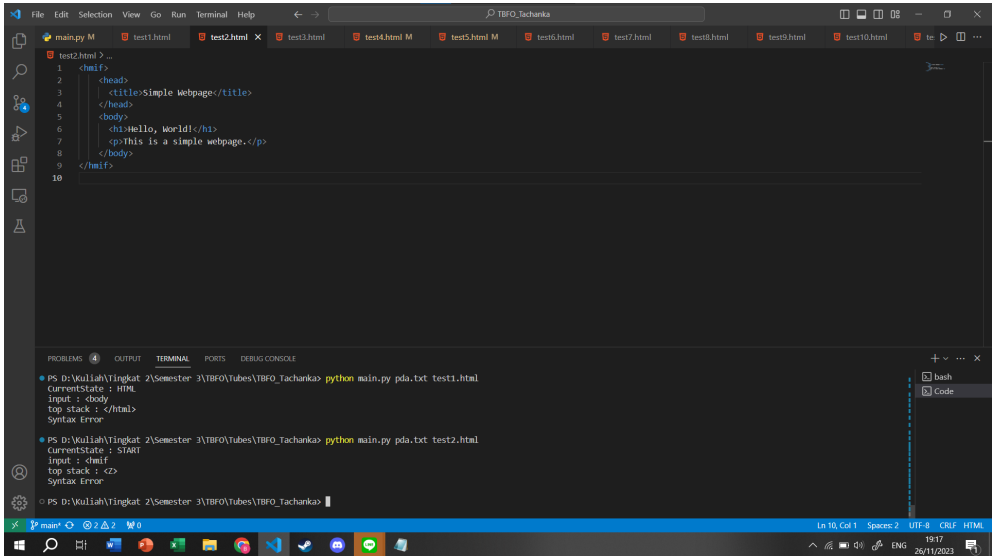
Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

4.2. Main

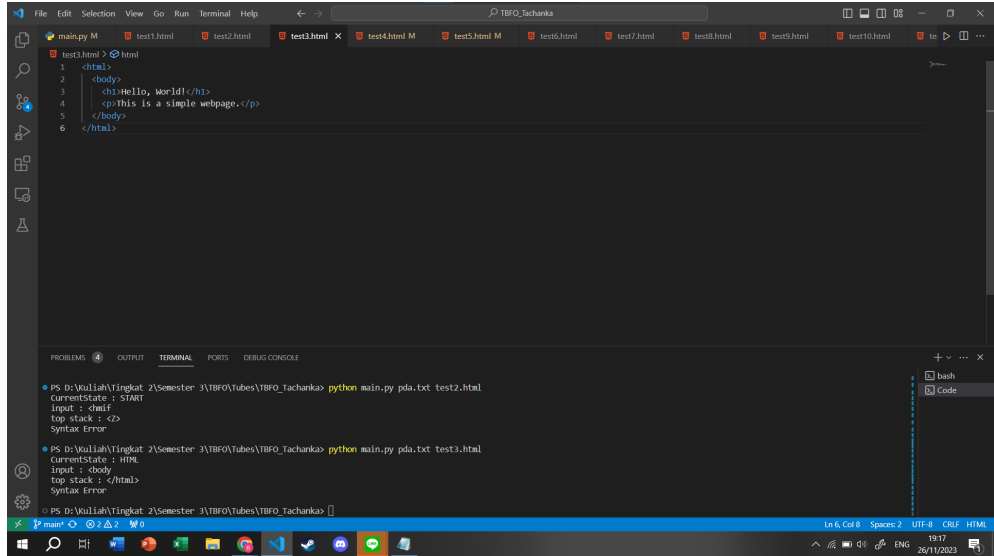
File main bertujuan untuk menerima nama file txt PDA, menerima nama file HTML yang ingin diakusisi, memanggil fungsi parser, dan melakukan proses validasi HTML berdasarkan PDA yang telah dipanggil.

4.3. Pengujian

Testcase	Hasil
1	 <p>Rejected</p>
2	 <p>Rejected</p>

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata
HTML Checker dengan Pushdown Automata (PDA)

3



```
test3.html > <html>
1
2   <body>
3     <h1>Hello, World!</h1>
4     <p>This is a simple webpage.</p>
5   </body>
6 </html>
```

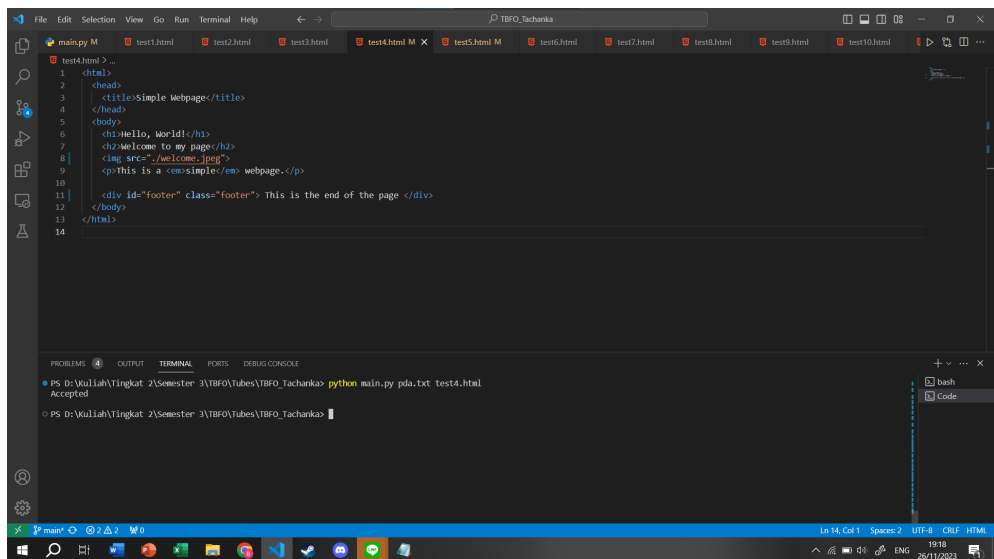
```
PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test2.html
CurrentState : START
input : dnull
top stack : <z>
Syntax Error

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test3.html
CurrentState : HTML
input : <dbody
top stack : </html>
Syntax Error

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Rejected

4



```
test4.html > <html>
1
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.jpg\\
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>
14
```

```
PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test4.html
Accepted

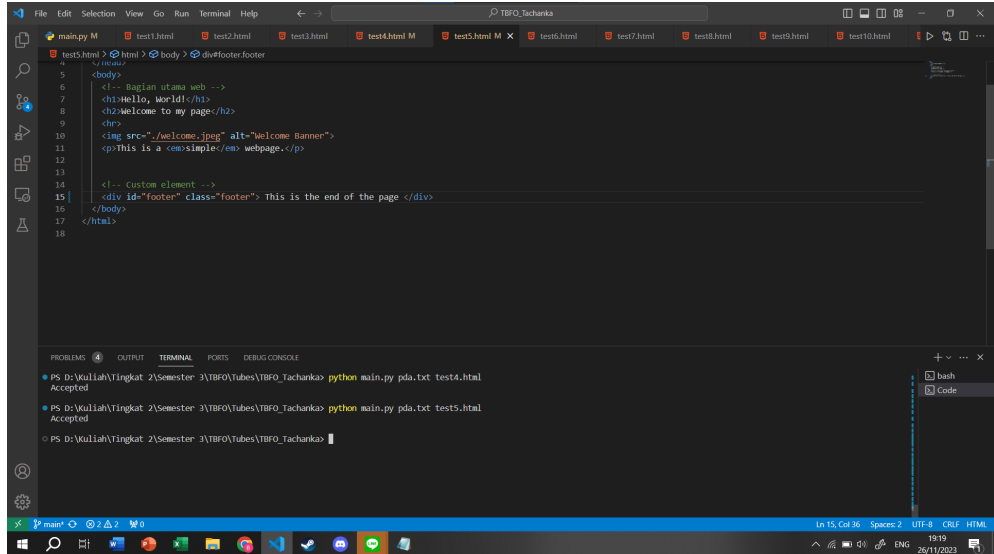
PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Accepted

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

5

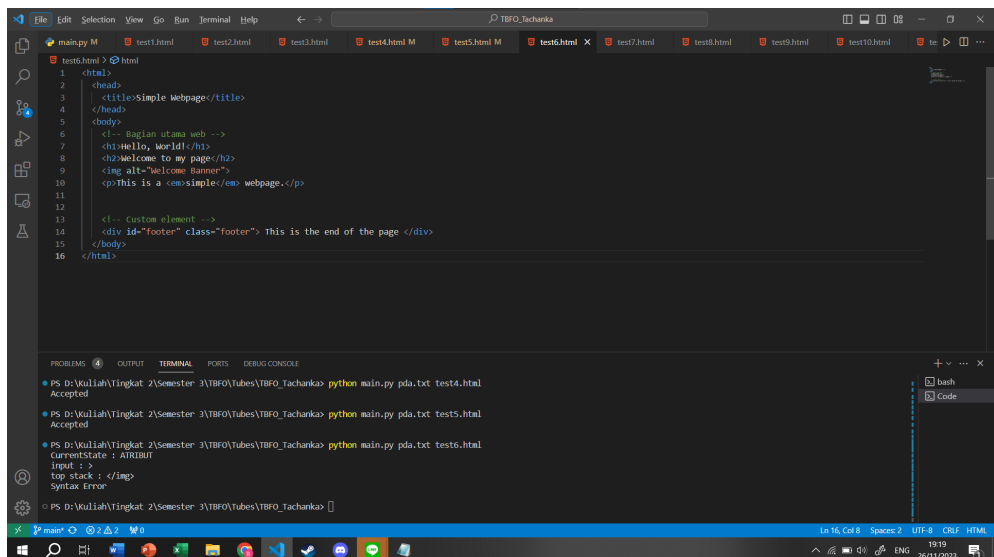


```
test5.html > <html> <body> <div footer.footer>
1
2
3
4
5
6 <!-- Bagian utama web -->
7 <h1>Hello, world!</h1>
8 <h2>Welcome to my page</h2>
9 <div>
10 
11 <p>This is a <em>simple</em> webpage.</p>
12
13
14 <!-- Custom element -->
15 <div id="footer" class="footer"> This is the end of the page </div>
16 </body>
17 </html>
18
```

```
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test4.html
Accepted
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test5.html
Accepted
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Accepted

6



```
test6.html > <html>
1
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
12
13 <!-- Custom element -->
14 <div id="footer" class="footer"> This is the end of the page </div>
15 </body>
16 </html>
```

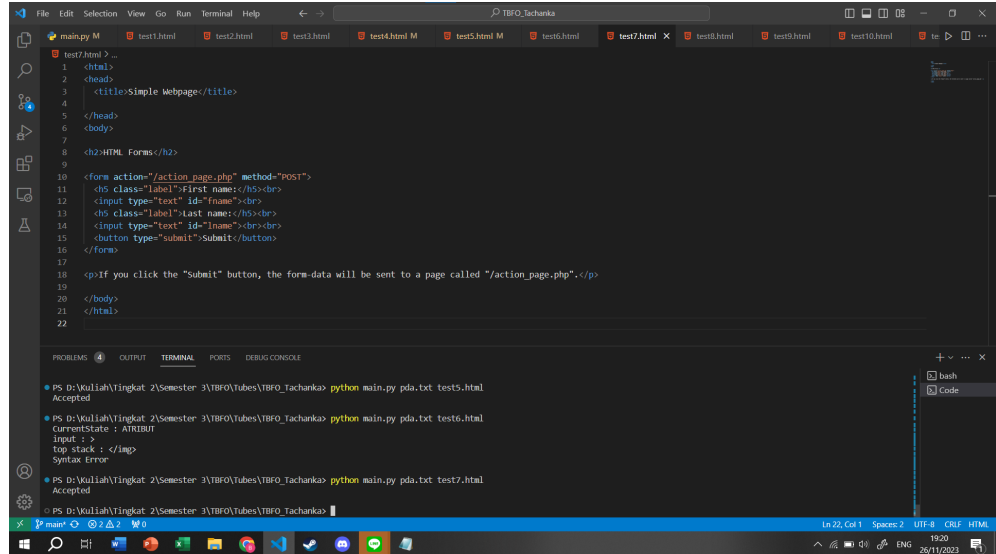
```
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test4.html
Accepted
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test5.html
Accepted
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test6.html
CurrentState : AIRIBUT
input : >
top stack : </img>
Syntax Error
PS D:\Vullah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Rejected

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

7

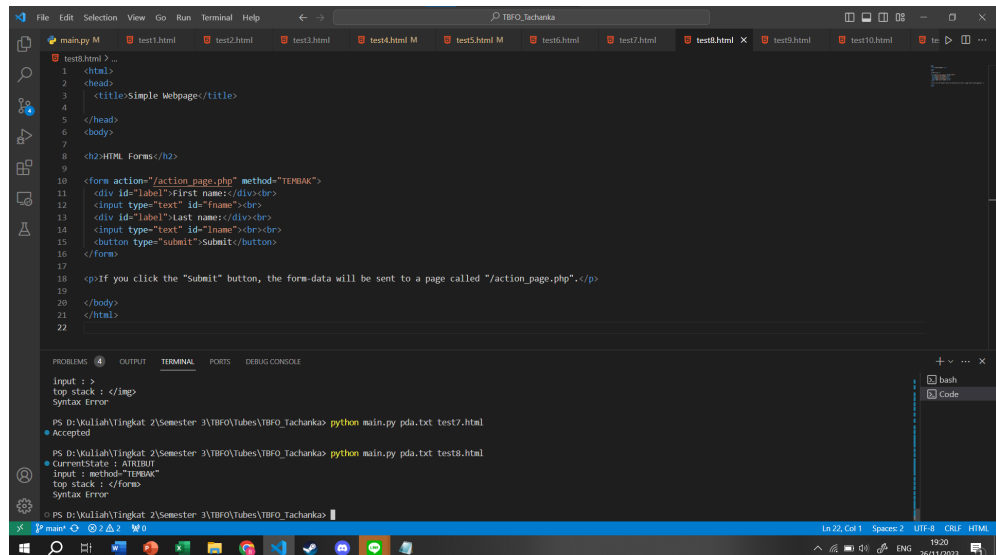


```
test7.html > ...
1 <html>
2 <head>
3   <title>Simple Webpage</title>
4
5 </head>
6 <body>
7
8 <h2>HTML Forms</h2>
9
10 <form action="/action_page.php" method="POST">
11   <div class="label">First name:</div><br>
12   <input type="text" id="fname"><br>
13   <div class="label">Last name:</div><br>
14   <input type="text" id="lname"><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>
22
```

```
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka> python main.py pda.txt test5.html
Accepted
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka> python main.py pda.txt test6.html
CurrentState : AIRIBUT
Input : >
top stack : </img>
Syntax Error
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka> python main.py pda.txt test7.html
Accepted
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka>
```

Accepted

8



```
test8.html > ...
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="TEMPBAK">
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>
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>
22
```

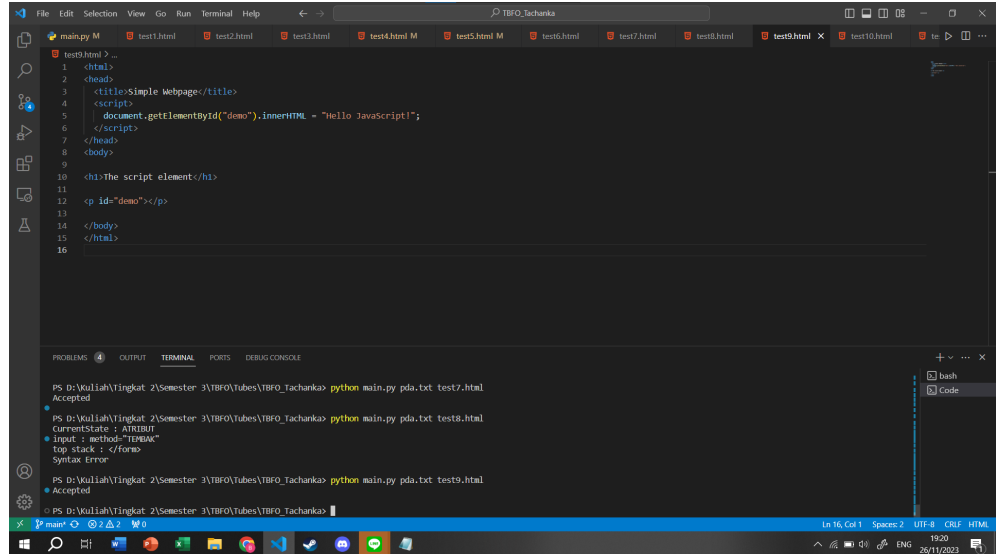
```
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka> python main.py pda.txt test7.html
Accepted
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka> python main.py pda.txt test8.html
CurrentState : AIRIBUT
Input : method="TEMPBAK"
top stack : </form>
Syntax Error
PS D:\Vulliah\Tingkat 2\Semester 3\TBO\Tubes\TBO_Tachanka>
```

Rejected

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

9



```
test9.html 2 ...
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
14 </body>
15 </html>
16

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test7.html
Accepted

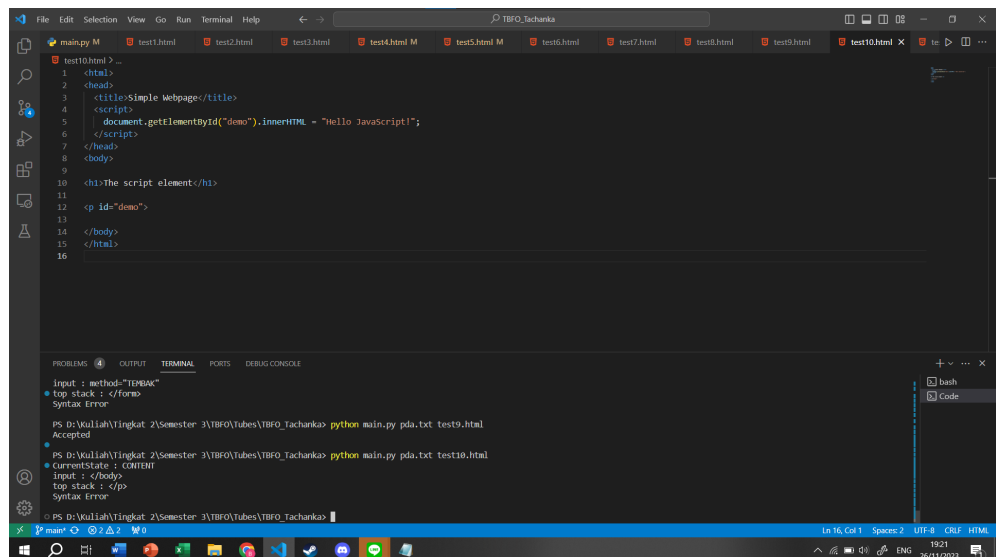
PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test8.html
CurrentState : ACCEPT
input : method="TEPBAC"
top_stack : </form>
Syntax Error

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test9.html
Accepted

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Accepted

10



```
test10.html 2 ...
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
14 </body>
15 </html>
16

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

input : method="TEPBAC"
top_stack : </form>
Syntax Error

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test9.html
Accepted

PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test10.html
CurrentState : CORRUPT
input : </body>
top_stack : </p>
Syntax Error

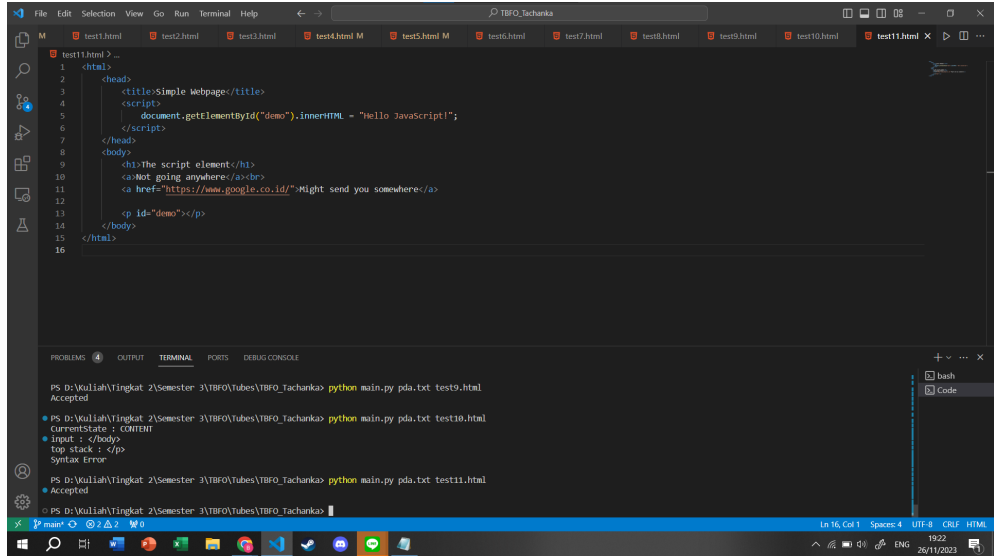
PS D:\Vulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Rejected

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

11



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

```
PS D:\Mulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test9.html
Accepted

PS D:\Mulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test10.html
CurrentState : CATCH
input : </body>
top stack : </p>
Syntax Error

PS D:\Mulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka> python main.py pda.txt test11.html
Accepted

PS D:\Mulliah\Tingkat 2\Semester 3\TBFO\Tubes\TBFO_Tachanka>
```

Accepted

Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata
HTML *Checker* dengan Pushdown Automata (PDA)

BAB V

Deliverables

Link repository dari Tugas Pemrograman IF 2124 Teori Bahasa Formal dan Otomata
HTML *Checker* dengan Pushdown Automata (PDA) kelompok Tachanka:

https://github.com/chankiel/TBFO_Tachanka

Link diagram PDA:

<https://app.diagrams.net/#G1iPfSGZLnOWw1GGVdnSUGrLHBHzWpk7YH>

BAB VI

Pembagian Tugas

NIM	Nama	Tugas
13522029	Ignatius Jhon Hezkiel Chan	Membuat PDA, Main, Laporan
13522043	Daniel Mulia Putra Manurung	Membuat PDA, Diagram, Laporan
13522054	Benjamin Sihombing	Membuat Parser, Diagram, Laporan

BAB VII

Daftar Pustaka

- “HTML: HyperText Markup Language | MDN.” *MDN Web Docs*, 17 July 2023,
<https://developer.mozilla.org/en-US/docs/Web/HTML>. Accessed 25 November 2023.
- “Pushdown automaton.” *Wikipedia*, https://en.wikipedia.org/wiki/Pushdown_automaton.
Accessed 25 November 2023.