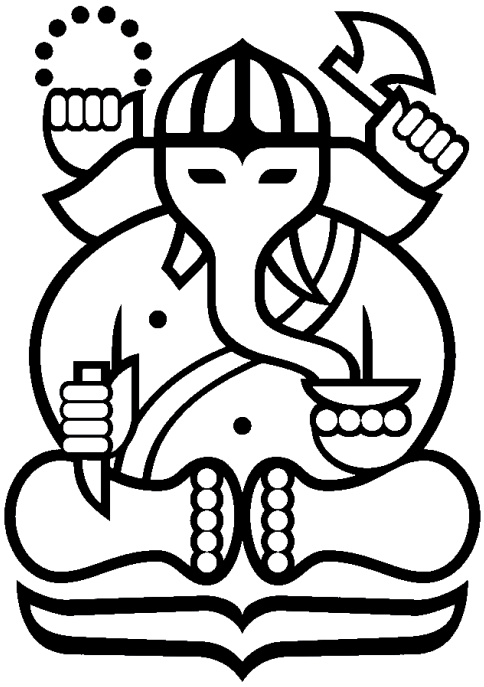
LAPORAN TUGAS BESAR
IF2124 TEORI BAHASA FORMAL DAN OTOMATA
HTML CHECKER



Disusun oleh:

Muhammad Naufal Aulia (13522074)

Muhammad Atpur Rafif (13522086)

Indraswara Galih Jayanegara (13522119)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2023

DAFTAR ISI

BAB I
DESKRIPSI MASALAH

HTML (Hypertext Markup Language) adalah bahasa markup yang digunakan untuk mengatur struktur dan tampilan konten pada situs web. HTML berfungsi untuk menentukan cara elemen-elemen konten seperti teks, gambar, tautan, dan media akan ditampilkan di dalam peramban web. Tiap dokumen HTML dimulai dengan elemen <html>, kemudian diikuti oleh <head> serta <body>.

Untuk mengelompokkan dan mengorganisir kontennya, HTML menggunakan elemen (tag). Sebagai contoh, <p> digunakan untuk paragraf teks, <h1> hingga <h6> digunakan untuk judul, <a> untuk tautan, <img> untuk gambar, dan sebagainya. Terdapat atribut yang memberikan informasi tambahan tentang elemen tersebut, seperti atribut src untuk gambar, href untuk tautan, dan class untuk memberikan kelas CSS pada elemen.

Mirip dengan bahasa lainnya, HTML memiliki sintaks khusus yang perlu dipatuhi agar tidak muncul error. Meskipun peramban web modern seperti Chrome dan Firefox biasanya mengabaikan kesalahan pada HTML, memastikan bahwa HTML benar dan terbentuk dengan baik tetap penting karena beberapa alasan seperti Search Engine Optimization (SEO), aksesibilitas, pemeliharaan yang lebih baik, kecepatan tampilan, dan tampilan yang profesional.

Oleh karena itu, diperlukan sebuah program pendeteksi kesalahan untuk HTML. Pada tugas pemrograman ini, akan digunakan konsep Pushdown Automata (PDA) dalam bahasa Python untuk mengimplementasikan program yang dapat memeriksa keakuratan html baik dari segi nama tag yang digunakan maupun atribut yang dimilikinya.

BAB II
TEORI DASAR

2.1 Pushdown Automata

Pushdown Automata (PDA) adalah suatu model mesin otomata yang ekivalen dengan Context Free Grammar (CFG). Dengan menggunakan pendekatan *stack*, memungkinkan PDA untuk memiliki tempat penyimpanan tidak terbatas untuk menyimpan query yang panjang.

PDA dideskripsikan dengan *tuple* yang berisikan 7 elemen, yakni sebagai berikut:

$$P = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$$

Dimana:

Q : himpunan *finite state*,

Σ : alfabet *input*,

Γ : alfabet *stack*,

$q_0 \in Q$: *start state*,

$Z_0 \in \Gamma$: *start stack symbol*,

δ : fungsi transisi, dan

$F \subseteq Q$: himpunan *final state.*

PDA terbagi menjadi 2 jenis, yaitu Deterministic Pushdown Automaton (DPDA) dan Non-deterministic Pushdown Automaton (NPDA). DPDA mempunyai aturan tertentu dan memiliki fungsi transisi yang deterministik. Artinya, untuk setiap konfigurasi, hanya ada satu aksi yang dapat diambil. Sedangkan NPDA memiliki pilihan aksi untuk setiap konfigurasi.

Salah satu contoh penggunaan PDA adalah dalam analisis sintaks pada pengembangan compiler, di mana PDA dapat digunakan untuk memeriksa apakah suatu string dapat dihasilkan oleh aturan sintaks dalam bahasa pemrograman tertentu. Dalam tugas besar ini konsep PDA digunakan untuk memeriksa kesalahan pada html.

2.2 Sintaks HTML

Secara umum, HTML memiliki sintaks dengan aturan atau struktur yang harus diikuti. Beberapa aspek kunci dari sintaks HTML adalah sebagai berikut.

a. Tag

Tag merupakan elemen dasar dalam HTML untuk menyusun dan mendefinisikan struktur halaman web. Dimulai dengan karakter "<", diikuti dengan nama tag, dan diakhiri dengan karakter ">". Berikut adalah contoh tag pada HTML :

```
html,
head,
body,
title,
head,
body,
title,
link,
script,
h1,h2,h3,h4,h5,h6,
p,
br,
em,
b,
abbr,
strong,
small,
hr,
div,
a,
img,
button,
form,
input,
table,
tr,
td,
th.
```

b. *Void element (singleton tag)*

Merupakan elemen yang tidak memerlukan tag penutup untuk menjadi valid. Berikut adalah contoh *void element* pada HTML :

`link,`

`br,`

`hr,`

`img,`

`input,`

c. Atribut

Berfungsi memberikan informasi tambahan pada elemen HTML dan mengonfigurasi atau menyesuaikan cara elemen tersebut ditampilkan. Berikut adalah contoh atribut pada HTML :

`rel,`

`href,`

`src,`

`alt,`

`type,`

`action,`

`method.`

d. Atribut global

Atribut global menjadi atribut yang dapat digunakan pada hampir semua elemen HTML, oleh karena itu atribut ini dapat ditempatkan pada sebagian besar elemen dalam HTML. Berikut adalah contoh atribut global pada HTML :

`id,`

`class,`

`style,`

2.3 Penjelasan Syntax HTML yang perlu diperhatikan dalam pembuatan PDA

Penggunaan tag pembuka dan penutup dalam HTML membentuk struktur hierarki yang dapat dimanfaatkan untuk membuat parser dengan menggunakan konsep PDA (Pushdown Automaton).

Konsep stack pada PDA sangat cocok karena memungkinkan untuk memantau dan melacak hierarki tag dalam dokumen HTML. Dalam HTML, setiap tag pembuka harus memiliki tag penutup yang sesuai. Misalnya, tag pembuka harus memiliki tag penutup , dan hal ini menciptakan struktur yang dapat dipetakan ke dalam konsep stack pada PDA. Dalam implementasi PDA untuk parser HTML, tag-tag tersebut digunakan untuk mengendalikan perubahan status stack.

Saat membaca tag pembuka, parser akan mendorong (push) tag tersebut ke dalam stack. Ketika menemukan tag penutup yang sesuai, parser akan memeriksa stack untuk memastikan bahwa tag tersebut cocok dengan tag teratas dalam stack sebelum kemudian mengeluarkan (pop) tag tersebut dari stack.

Dengan menggunakan struktur stack pada PDA, kita dapat melacak konsistensi dan kedalaman hierarki dari tag-tag HTML, memastikan bahwa setiap tag pembuka memiliki tag penutup yang sesuai dan bahwa strukturnya benar dan teratur.

2.3.1 Whitespace dan space

Beberapa tag di HTML perlu memperhatikan penggunaan Whitespace dan space, dimana setelah tag < > terdapat spasi contoh

```
< html   > </html>
```

pada PDA yang kami gunakan struktur HTML di atas kami terima. Jika ada spasi di yang memisahkan tag maka itu akan ditolak. Contoh

```
<ht ml> </html>
```

\struktur ini akan ditolak oleh PDA yang kami buat karena tidak sesuai dengan struktur file HTML. Kami juga menerima tag yang memiliki newline seperti

```
<html
>
</html>
```

struktur di atas akan diterima oleh PDA kami karena kami mengatasi hal tersebut.

2.3.2 Comment

Kami menghandle kasus untuk memberikan comment di luar tag html. Jadi

```
< !  - - comment  - - >
<html>
</html>
```

hal di atas akan diterima oleh PDA yang kami buat. Akan tetapi, kami juga menghandle kasus comment ada di dalam tag html sesuai spek yang diminta. Jadi, Struktur di bawah ini akan diterima oleh PDA.

```
<html>
<! - - comment  - - >
</html>
```

2.3.3 Ignore Case

Ada beberapa kasus dimana attribute di dalam sebuah tag menerima sentence seperti dibawah ini. Semua kasus akan diterima baik yang hanya menggunakan uppercase, lowercase, maupun gabungan dari keduanya.

```
<input type = “POST” >
<input type = “Post”>
<input type = “post”>
```

2.3.4 Klasifikasi Transition Functions

A. Definisi

Merupakan sebuah fungsi transisi yang mendefinisikan sebuah bentuk secara garis besar. Sebuah bentuk dapat terdiri dari kumpulan macro, fungsi, atapun karakter. Berikut adalah contohnya:

```
(BODY \eps BODY) = BODY | WS < BODY_TAG GLOBAL_ATTRIB
WS > BODY_CHILD < / BODY_TAG WS > WS
```

B. Macro

Merupakan fungsi transisi yang memungkinkan adanya pengelompokan sebuah karakter. Selain itu hal ini dapat membuat sebuah percabangan. Macro disini tidak melakukan perpindahan state, dan hanya dapat melakukan ekspansi dari sebuah stack.

```
(BODY_CHILD_BASE \eps BODY_CHILD_BASE) =
BODY_CHILD_BASE | DIV
(BODY_CHILD_BASE \eps BODY_CHILD_BASE) =
BODY_CHILD_BASE | TABLE
```

C. Fungsi

1. Fungsi Tidak Berparameter

Pada organisasi dan arsitektur komputer, terdapat sebuah cara yang dilakukan oleh komputer untuk melakukan transfer control dari sebuah

fungsi menjadi fungsi yang lain menggunakan stack. Ide tersebut dapat diterapkan pada PDA agar dapat melakukan isolasi pada setiap state. Sebelum masuk ke dalam state lain memiliki arti masuk kedalam fungsi lain. Pada saat pemanggilan fungsi lain, pada stack diisi dengan nilai yang relevan pada state yang dipanggil dan state pemanggil agar kontrol dapat dikembalikan.

Misalkan dibawah ini terdapat fungsi yang dinamakan dengan `WS` yang nantinya akan dipanggil oleh `HTML`. Pada fungsi transisi `#1,` hal ini disebut dengan definisi. Didalamnya akan dipanggil fungsi `WS.` Pada fungsi `#2` merupakan setup yang dilakukan oleh pemanggil sebelum masuk kedapam fungsi yang dipanggil. Perhatikan bahwa pada stack terdapat nilai `HTML` , yang nantinya setelah fungsi yang dipanggil selesai, state akan menuju nilai tersebut. Hal ini tertulis pada fungsi `#3` , yaitu kembalinya state ke `HTML.` Fungsi juga bisa memanggil fungsi lainnya untuk membentuk sebuah grammar utuh.

```
(WS \eps WS) = WS |
(WS \eps WS) = WS | WHITESPACE WS
(WS \eps WHITESPACE) = WHITESPACE | WHITESPACE
WS
(WHITESPACE \eps WS) = WS |

(HTML \eps HTML) = HTML | ... WS ... #1
(HTML \eps WS) = WS | WS HTML #2
(WS \eps HTML) = HTML |#3
```

2. Fungsi Berparameter

Pada kasus khusus, misalkan ketika kita ingin tidak mempedulikan kapital tidaknya dari kumpulan karakter, kita bisa menggunakan fungsi yang berparameter. Sama seperti sebelumnya, namun kali ini kita

memberikan nilai dari parameter langsung pada stack. Berikut merupakan contohnya:

```
(HTML \eps HTML_TAG) = IGNORE_CASE | h t m l
HTML
(BODY \eps BODY_TAG) = IGNORE_CASE | b o d y
BODY
(DIV \eps DIV_TAG) = IGNORE_CASE | d i v DIV
(TH_BASE \eps TH_TAG) = IGNORE_CASE | t h
TH_BASE
(TR_BASE \eps TR_TAG) = IGNORE_CASE | t r
TR_BASE
```

2.3.5 Rekursif

Pada PDA kami menggunakan rekursif pada transition function yang kami gunakan untuk mendeteksi nested tag pada file HTML yang akan dicek.

BAB III
HASIL PDA

3.1 Hasil PDA

Kurang lebih, seperti ini hasil PDA kami untuk state HTML saja. Dikarenakan terlalu banyak untuk diperlihatkan dalam laporan maka hanya HTML saja yang ditampilkan. Disini kami menggunakan fungsi ini untuk menyamakan string dan PDA

```
matcher = re.match(r"\((\\?\$?\w+)\s+([^\s]+)\s+(\\?\$?\w+)\)\s+=\s+(\\?\$?\w+)\s+\|\s?(.+)?", line)
```

```
(Q \eps Z0) = HTML | HTML

(HTML \eps HTML) = HTML | WS OPTIONAL_COMMENT < HTML_TAG
GLOBAL_ATTRIB WS > WS OPTIONAL_COMMENT HEAD WS OPTIONAL_COMMENT
BODY WS OPTIONAL_COMMENT < / HTML_TAG WS > WS OPTIONAL_COMMENT
(HTML \eps OPTIONAL_COMMENT) = HTML | COMMENT WS
(HTML \eps OPTIONAL_COMMENT) = HTML |
(HTML \eps HTML_TAG) = IGNORE_CASE | h t m l HTML
(IGNORE_CASE \eps HTML) = HTML |
(HTML \eps WS) = WS | WS HTML
(WS \eps HTML) = HTML |
(HTML \eps HEAD) = HEAD | HEAD HTML
(HEAD \eps HTML) = HTML |
(HTML \eps BODY) = BODY | BODY HTML
(BODY \eps HTML) = HTML |
(HTML \eps COMMENT) = COMMENT | COMMENT HTML
(COMMENT \eps HTML) = HTML |
(HTML \eps GLOBAL_ATTRIB) = GLOBAL_ATTRIB | GLOBAL_ATTRIB HTML
(GLOBAL_ATTRIB \eps HTML) = HTML |
(HTML < <) = HTML |
```

```
(HTML > >) = HTML |
(HTML / /) = HTML |
```

BAB IV
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

4.1.1 PDA.py

Nama Fungsi (parameter)	Deskripsi
`__init__(self, states: list[str], input: list[str], stack: list[str], startState: str, startStack: str, transition: Transition`	memanggil seluruh state yang ada di pda.txt dan diolah untuk menjalankan algoritma PDA
`pushJob(self, id: ID)`	melakukan push pada PDA
`start(self, input: InputPDA)`	memulai eksekusi PDA

4.1.2 ConfigParser.py

Nama Fungsi (parameter)	Deskripsi
`parserPDAConfig(raw: str)`	Menerima file txt untuk diubah ke dalam rule production

4.1.3 Main.py

Nama Fungsi (parameter)	Deskripsi
-	Menjalankan program secara keseluruhan

4.2 Uji Coba Program

4.2.1 Kasus 1

File: stressTest1.html

```
<html >
    <head             >
    <tiTlE> ini title random tcuy </titLe>
    </HeAd>
    <BodY>
        <H1 class = "ini Checker yang kami buat"
          > ini judul H1 # </h1>
        <p> masukkan inputmu disini atau Rating Kami</P>
        <input type = "password" id = "harus" class = "harus kamu
isi kalo gak aku hantuin!!!!!!" />
        <form method = "GET" class = "ini adalah method"> </form>
        <p id = "momen"><em> ini momen bjir</Em> </P>
        <p id = "$%^&*#@"> <strong id = "@#$%^&"> Ini #$%</strong>
</P>
          <div id = "momen">
            <Div id = "showoff">
                <dIv id = "brutal">
                    <dIv id = "psikopat">
                        <DiV id = "Manayangnulis"
                            >
                            <DIv id = "HTML">
                                <div class = "Begini BJIR BJIR
BJIR BJIR bjir">
                                    <p>
                                        <Em> this is FORMATED
```

```
TEXT</eM>
                                    </P>
                                </div>
                                <diV>
                                <DIV>
                                    <p> Ini stress tress kami
paling brutal silakan dicoba </P>
                                    <H5 id = "Yakinkamumas"> Cepet
mas kerJain!!!! </h5>
                                </dIV>
                                    <p> </P>
                                    <tabLe>
                                        <tr>
                                        <td>
                                            <b><ABBR> Ini SH 0W
0Ff!!!!!!%$#^&*@ </AbbR></B>
                                        </td>
                                        </tr>
                                        <th>
                                        <td>
                                        </td>
                                        </th>
                                    </TaBlE>
                                </DIv>
                            </DIv>
                        </DIv>
                    </div>
                </diV>
            </diV>
          </diV>
    </BOdY>
</html>
```

Hasil:

```
PDA file     :  pda.txt
HTML file    :  test/stresstest1.html
Checking test/stresstest1.html, please wait...
Character count                 : 2187
Iteration count                 : 110855
Itaration/Character Ratio       : 5068%
Itaration/Character Ratio       : 5068%
Maximum ID count                : 29
Execution time                  : 0.3537ms
```

File: stressTest2.html

```
<html>
<head>
    <TItLe id = "stress">
        Ini Stress Test ketiga, good luck
    </TItLe>
</head>
<body>
    <div>
        <p>
            makin sayang sama tubes, uwooghhhhhh
        </P>
        <DiV>
            <DIV class = "iniclassTest">
                <P>
                    Coba terus ini ketemu atau enggak
                </p>
                <H2 id = "text">
                    Ini bagian H2
                </h2>
            </dIV>
            <dIV>
                <div>
                    <div>
                        <H1 id="bruh">
                            Ini kayaknya Heading 1????
                        </H1       >
                        <P>
```

```
                    uh ini cuma buat testcase
                </p>
            </div>
            <img src="if.momen" alt="HMIF">
            <img alt="ASSIST" src="sti.momen" >
            <DIv>
                <dIv>
                    <H1>
                       Ini di H1, cek lagi ayok
                    </h1>
                    <p>
                        H3h3, apakah punyamu accept (?)
                    </P   >
                </dIv  >
            </DIv   >
            <img src="momen.IF" alt="akucintatubes">
            <img alt="gambarTubes" src="sayangsamaTubes" >
        </div    >
    </dIV>
    <diV>
        <H3   >
            Ini "stress banget"
        </h3>
    </div>
    <dIv class = "ini div keberapa"  >
        <tAblE   >
            <th>
                <tD  >
                    123
                </td>
                <Td>
                    <smaLl>123  123  %^&</sMALl>
                </td   >
                <Td>
                    <b>654</b>
                </tD    >
            </th>
            <tr>
                <td  >
```

```
                                $#%
                            </td  >
                            <td    >
                                <strong   >^&*</strong     >
                            </td>
                            <td     >
                                <em>$#%</em>
                            </td>
                        </tr>
                    </TablE>
                </div>
            </DiV>
        </div>

</body>
</html>
```

Hasil:

```
PDA file     :  pda.txt
HTML file    :  test/stressTest2.html
Checking test/stressTest2.html, please wait...
Character count                 : 9504
Iteration count                 : 483792
Itaration/Character Ratio       : 5090%
Maximum ID count                : 29
Execution time                  : 1.5433ms
```

4.2.2 Kasus 2

```
<html>
    <head>
        <script id = "script">
            document.getElementById("script")
        </script>
        <title>
            Ini test ke-1
        </title>
    </head>
    <body>
        <h1>
            ini judul di body
        </h1>
        <h2>
            ini ya heading2
        </h2>
        <p>
            ini paragraph
        </p>
    </body>
</html>
```

Hasil:

```
PDA file     :  pda.txt
HTML file    :  test/test1.html
Checking test/test1.html, please wait...
Character count                : 377
Iteration count                : 12984
Itaration/Character Ratio      : 3444%
Maximum ID count               : 29
Execution time                 : 0.0380ms
```

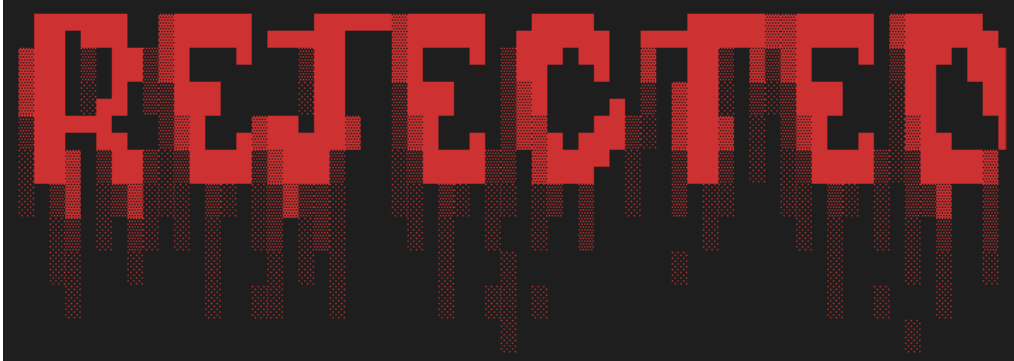
File: test2.html

```
<hmif>
    <head>
```

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

Hasil:

```
PDA file     :  pda.txt
HTML file    :  ./test/test2.html
Checking ./test/test2.html, please wait...
Character count                 : 173
Iteration count                 : 16
Itaration/Character Ratio       : 9%
Maximum ID count                : 3
Execution time                  : 0.0010ms
```




```
<hmif>
Error at line 1
```

Penjelasan kesalahan:

Tag <html> di awal berisi sintaks yang tidak valid

4.2.3 Kasus 3

File: test11.html

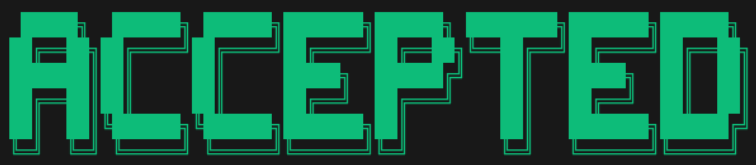
```
<html>
    <head>
        <title>Simple Webpage</title>
```

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

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

Hasil:

```
PDA file     :  pda.txt
HTML file    :  test/test11.html
Checking test/test11.html, please wait...
Character count                 : 384
Iteration count                 : 13235
Itaration/Character Ratio       : 3446%
Maximum ID count                : 29
Execution time                  : 0.0390ms
```



File: test12.html

```
<html>
    <head    >
        <title    >
            simple table
        </title>
    </head   >
    <body     >
        <h1   >show table</h1   >
        <table id = "testing-table">
            <th>
                <td>
                    data 1
```

```
                </td   >
                <td   >
                    data 2
                </td>
                <td  >
                    data 3
                </td>
            </th   >
            <tr   >
                <td>
                    data1
                </td   >
                <td>
                    data2
                </td  >
                <td>
                    data3
                </td   >
            </tr   >
        </table  >
    </body  >
</html>
```

Hasil:

```
PDA file     :  pda.txt
HTML file    :  test/test12.html
Checking test/test12.html, please wait...
Character count                : 739
Iteration count                : 21746
Itaration/Character Ratio      : 2942%
Maximum ID count               : 29
Execution time                 : 0.0550ms
```

BAB V
PENUTUP

5.1 Kesimpulan

Dalam tugas besar Teori Bahasa Formal dan Automata yang menitikberatkan pada pembuatan HTML Checker dengan memanfaatkan konsep Pushdown Automata (PDA), kami berhasil mengembangkan sebuah program yang dapat memvalidasi kebenaran sintaks HTML. Hal ini dicapai melalui implementasi PDA sebagai model komputasi yang dapat menangani struktur hierarkis yang umumnya ditemui dalam sintaksis HTML.

Melalui Tugas kuliah ini, kami mendapatkan pemahaman terhadap cara kerja PDA secara lebih dalam dan implementasinya dalam pemrograman untuk mengecek HTML. Tidak hanya itu, tugas ini pun turut membiasakan kami terhadap struktur dan sintaks HTML, mendorong kemampuan dalam analisa teoritikal dan praktikal, serta keterampilan *debugging*.

5.4 Link Repository

Link Github *repository* Tugas Besar:

Link Repository

5.5 Link Diagram State

Link *Google Drive* diagram:

Link Google Drive

Link diagram.net:

Link diagram.net

5.6 Pembagian Tugas

Muhammad Naufal Aulia	**Rules, Diagram State, Laporan**
Muhammad Atpur Rafif	**PDA, Rules, ReadConfig, Laporan**
Indraswara Galih Jayanegara	**Rules, Diagram state, Laporan**

REFERENSI

https://medium.com/@zahma.abi/push-down-automata-a004841adde1 diakses pada 21 November 2022

https://www.w3schools.com/html/ diakses pada 21 November 2022

HOPCROFT, J. E. (n.d.). *Introduction to Automata Theory, Languages, and Computation 3rd Edition*.