

LAPORAN TUGAS BESAR
PENGUNAAN PUSHDOWN AUTOMATA UNTUK MEMERIKSA
SINTAKS BAHASA PEMROGRAMAN HTML
IF2124 TEORI BAHASA FORMAL DAN OTOMATA



Disusun oleh:

Mulia Pratama Clan

Aland Mulia Pratama (13522124)

Rizqika Mulia Pratama (13522126)

Ikhwan Al Hakim (13522147)

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023/2024

DAFTAR ISI

DAFTAR ISI.....	1
BAB I	
DESKRIPSI MASALAH.....	1
BAB II	
TEORI SINGKAT.....	3
I. Pushdown Automata (PDA).....	3
A. Pengertian PDA.....	3
B. Struktur PDA.....	4
C. Operasi Pada PDA.....	4
D. Kegunaan PDA.....	4
II. HyperText Markup Language (HTML).....	5
A. Elemen HTML.....	5
B. Atribut.....	6
C. Struktur Dasar Dokumen HTML.....	6
D. Heading dan Paragraf.....	6
E. List.....	6
F. Link.....	6
G. Gambar.....	6
H. Formulir.....	6
I. Semantik HTML5.....	7
BAB III	
IMPLEMENTASI PUSTAKA.....	7
A. Implementasi PDA.....	7
B. Implementasi Program.....	15
a. doc.....	15
b. src.....	15
BAB IV	
PROGRAM UTAMA.....	17
BAB V	
PENGUJIAN PROGRAM.....	18
BAB VI	
KESIMPULAN DAN SARAN.....	25
A. Kesimpulan.....	25
B. Saran.....	25
BAB VI	
PEMBAGIAN KERJA.....	26
DAFTAR PUSTAKA.....	27

BAB I DESKRIPSI MASALAH



Gambar 1 HTML

HTML, atau Hypertext Markup Language, merupakan suatu bahasa markup yang esensial dalam dunia pengembangan web. Fungsinya tak terbantahkan dalam merinci struktur dan presentasi konten web, termasuk teks, gambar, tautan, dan media lainnya, untuk ditampilkan secara optimal di berbagai browser web. Masing-masing dokumen HTML dimulai dengan elemen `<html>`, diikuti oleh `<head>` yang mengandung metadata dan tautan ke file eksternal, serta `<body>` yang memuat konten yang akan dihadirkan kepada pengguna.

Dalam proses konstruksi, HTML menggunakan elemen atau "tags" untuk mengelompokkan dan merapikan kontennya. Sebagai ilustrasi, `<p>` digunakan untuk membuat paragraf teks, `<h1>` hingga `<h6>` untuk judul dengan tingkat kepentingan berbeda, `<a>` untuk membuat tautan, `` untuk menyisipkan gambar, dan sebagainya. Pentingnya atribut juga tidak boleh diabaikan, karena mereka memberikan informasi tambahan tentang elemen tersebut. Misalnya, atribut `src` digunakan untuk menentukan sumber gambar, `href` untuk menentukan tujuan tautan, dan `class` untuk memberikan elemen kelas CSS tertentu.

Sebagaimana bahasa pemrograman lainnya, HTML memiliki sintaks yang harus diikuti dengan ketat. Kesalahan dalam sintaks bisa menyebabkan masalah tampilan atau fungsionalitas. Meskipun browser modern cenderung toleran terhadap kesalahan, memastikan keberlanjutan dan kebenaran struktur HTML tetap menjadi prioritas utama. Ini tidak hanya berkaitan dengan Search Engine Optimization (SEO), melainkan juga dengan aspek-aspek lainnya seperti aksesibilitas, pemeliharaan yang lebih baik, kecepatan tampilan, dan kesan profesional secara keseluruhan.

Untuk mengatasi potensi kesalahan, diperlukan alat deteksi kesalahan yang dapat secara spesifik memeriksa kevalidan HTML, termasuk penggunaan nama tag dan atributnya. Oleh karena itu, disarankan untuk mengimplementasikan program yang dapat melakukan pemeriksaan kevalidan HTML menggunakan konsep Pushdown Automata (PDA) dengan menggunakan bahasa pemrograman Python. Hal ini akan membantu memastikan bahwa dokumen HTML mematuhi standar yang ditetapkan dan dapat diinterpretasikan dengan benar oleh browser web.

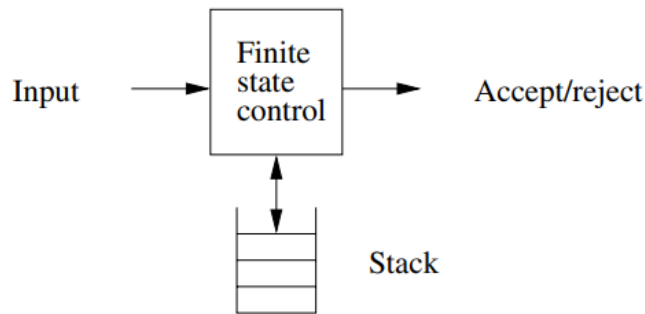
BAB II

TEORI SINGKAT

I. Pushdown Automata (PDA)

A. Pengertian PDA

Pushdown Automata (PDA) adalah jenis mesin otomata yang memiliki tumpukan (stack) sebagai memorinya. Tumpukan ini dapat digunakan untuk menyimpan informasi, seperti informasi tentang posisi dalam string input, informasi tentang hasil dari operasi-operasi sebelumnya, atau informasi lainnya yang diperlukan untuk memproses string input. Secara formal, PDA dapat dijelaskan sebagai tupel yang terdiri dari himpunan keadaan, alfabet input, alfabet tumpukan, fungsi transisi, keadaan awal, simbol tumpukan awal, dan himpunan keadaan akhir.



Gambar 2 Pushdown Automaton dengan tumpukan data

Fungsi transisi PDA (*pushdown automaton*) memetakan keadaan, simbol input, dan simbol tumpukan ke keadaan berikutnya dan urutan simbol tumpukan yang baru. Operasi pada tumpukan, seperti push (mendorong) dan pop (mengeluarkan), memungkinkan PDA untuk melakukan manipulasi tumpukan selama prosesnya. PDA dapat menerima bahasa bebas konteks, yang dapat dihasilkan oleh tata bahasa bebas konteks.

PDA dapat menerima bahasa bebas konteks (*context-free language*). Bahasa bebas konteks adalah bahasa yang dapat digenerasikan oleh tata bahasa bebas konteks (*context-free grammar*). Tata bahasa bebas konteks (*context-free grammar*) sendiri terdiri dari himpunan simbol non-terminal, simbol terminal, aturan produksi, dan simbol awal. Aturan produksi menggambarkan cara simbol non-terminal dapat diubah menjadi urutan simbol terminal dan/atau non-terminal.

Setiap bahasa bebas konteks (*context-free language*) dapat diakui oleh sebuah PDA, dan sebaliknya, menunjukkan keterkaitan erat antara PDA dan bahasa bebas konteks. Aturan produksi dalam tata bahasa bebas konteks dapat diterapkan pada PDA untuk membentuk urutan transisi yang mengenali string dalam bahasa tersebut. Selain itu, tata bahasa bebas konteks dapat diubah ke dalam bentuk Chomsky Normal Form untuk memudahkan analisis formal. Melalui hubungan ini, PDA dan bahasa bebas konteks menjadi instrumen penting dalam memahami dan memodelkan keluasan bahasa formal.

B. Struktur PDA

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

PDA terdiri dari empat komponen utama, yaitu:

- State set (Q), yaitu himpunan keadaan yang dapat dimiliki oleh PDA.
- Input alphabet (Σ), yaitu himpunan simbol input yang dapat diterima oleh PDA.
- Stack alphabet (Γ), yaitu himpunan simbol yang dapat disimpan di tumpukan.
- Transition function (δ), yaitu fungsi transisi yang menentukan keadaan berikutnya dan operasi tumpukan yang dilakukan oleh PDA.
- Start state (q_0), PDA berada dalam state ini sebelum melakukan transisi apapun.
- Start symbol (z_0), Simbol inisial yang berada dalam PDA.
- Accepting States (F), Final states or empty stack.

C. Operasi Pada PDA

PDA dapat melakukan dua operasi tumpukan, yaitu:

- PUSH (menambahkan simbol ke tumpukan)

Operasi PUSH dilakukan untuk menambahkan simbol baru ke atas tumpukan. Saat PDA membaca simbol dari input atau sebagai hasil dari suatu transisi, simbol tersebut dapat ditambahkan ke tumpukan. Operasi PUSH memungkinkan PDA untuk menyimpan informasi tambahan yang diperlukan dalam pemrosesan string input. Sebagai contoh, jika PDA berusaha memahami struktur tertentu dalam string, simbol-simbol yang mewakili struktur tersebut dapat ditambahkan ke tumpukan agar dapat diakses dan digunakan dalam tahapan selanjutnya dari proses.

- POP (menghapus simbol dari tumpukan)

Operasi POP dilakukan untuk menghapus simbol teratas dari tumpukan. PDA dapat menggunakan operasi POP ketika perlu mengakses atau memanipulasi informasi yang disimpan dalam tumpukan. Sebagai contoh, jika PDA ingin mengenali pola tertentu dalam string input, operasi POP dapat digunakan untuk menghapus simbol-simbol yang tidak relevan dari tumpukan. Dengan demikian, PDA dapat fokus pada informasi yang lebih penting dan relevan untuk proses selanjutnya.

Dua operasi ini, PUSH dan POP, memberikan fleksibilitas pada PDA untuk menyimpan dan mengelola informasi secara dinamis selama prosesnya. Kombinasi dari operasi-operasi ini memungkinkan PDA untuk mengenali bahasa bebas konteks dan menjalankan proses pemrosesan yang kompleks tergantung pada struktur tata bahasa bebas konteks yang mendefinisikan bahasanya.

D. Kegunaan PDA

PDA digunakan dalam berbagai bidang ilmu komputer, seperti:

a. Teori bahasa (Language Theory)

PDA memiliki peran penting dalam teori bahasa, terutama dalam memahami dan menganalisis bahasa bebas konteks. Dengan menggunakan PDA, peneliti dapat membuktikan hubungan antara bahasa formal dan mesin otomata, membantu dalam membuktikan teorema seperti teorema Pumping Lemma, dan memahami sifat-sifat bahasa formal.

b. Kompilasi (Compiler)

Dalam kompilasi, PDA digunakan untuk menerjemahkan bahasa pemrograman tingkat tinggi ke dalam bentuk bahasa mesin atau bentuk yang dapat dieksekusi oleh komputer. PDA membantu dalam analisis sintaksis (parsing) kode sumber dan memastikan bahwa kode tersebut sesuai dengan struktur tata bahasa yang benar.

c. Parsing

PDA digunakan dalam proses parsing untuk memecahkan string input menjadi komponen-komponennya atau untuk memverifikasi apakah string tersebut termasuk dalam bahasa yang diterima oleh tata bahasa bebas konteks. Algoritma parsing yang berdasarkan pada konsep PDA digunakan dalam kompilasi, penguraian ekspresi matematika, dan berbagai aplikasi lain yang melibatkan analisis struktural string.

d. Verifikasi program (program verification)

Dalam verifikasi perangkat lunak, PDA dapat digunakan untuk memeriksa kebenaran suatu program. PDA membantu dalam mengidentifikasi kesalahan sintaksis atau kesalahan struktural pada tingkat bahasa formal. Selain itu, PDA dapat digunakan dalam model-model formal untuk memverifikasi properti tertentu dari program, seperti keamanan atau ketidakberubahannya.

II. HyperText Markup Language (HTML)

Sintaks HTML merupakan aturan atau struktur dasar dalam penulisan dokumen HTML. Sintaks HTML bisa dibagi menjadi beberapa struktur,

A. Elemen HTML

HTML dibangun menggunakan elemen-elemen. Elemen dimulai dengan tag dan diakhiri dengan tag yang serupa. Contoh: `<tag>konten</tag>`. Tag juga terdiri dari nama tag yang ditulis di dalam kurung sudut. Contoh: `<tag>` untuk paragraf dan `<tag>` untuk heading level satu.

B. Atribut

Sintaks HTML dapat memiliki atribut yang memberikan informasi tambahan tentang elemen tersebut. Atribut ditempatkan di dalam tag pembuka dan memiliki nilai tertentu. Contoh: ``.

C. Struktur Dasar Dokumen HTML

Sebuah dokumen HTML biasanya dimulai dengan tag `<html>` sebagai elemen induk. Bagian `<head>` mengandung informasi meta, judul halaman, dan referensi ke stylesheet atau skrip JavaScript. Bagian `<body>` berisi konten utama dokumen, seperti teks, gambar, dan elemen-elemen lainnya.

D. Heading dan Paragraf

Tag-heading `<h1>` hingga `<h6>` digunakan untuk memberikan judul dengan tingkat kepentingan berbeda. Sedangkan tag paragraf `<p>` digunakan untuk menulis teks paragraf.

E. List

Terdapat dua jenis list pada HTML, yaitu *ordered* (dengan nomor) menggunakan tag `` dan *unordered* (tanpa nomor) dengan menggunakan tag ``.

F. Link

Untuk membuat tautan atau *hyperlink*, digunakan tag `<a>` dengan atribut 'href' yang berisi URL tujuan.

Contoh: `Tautan`

G. Gambar

Tag `` digunakan untuk menyisipkan gambar. Atribut 'src' berisi URL gambar, dan 'alt' memberikan deskripsi alternatif jika gambar tidak dapat ditampilkan.

H. Formulir

Tag `<form>` digunakan untuk membuat formulir interaktif. Berisi elemen-elemen seperti `<input>`, `<textarea>`, dan lainnya.

I. Semantik HTML5

HTML5 memperkenalkan elemen-elemen semantik seperti <header>, <nav>, <article>, <section>, <footer>, yang memberikan arti khusus pada bagian bagian dokumen.

BAB III IMPLEMENTASI PUSTAKA

A. Implementasi PDA

PDA digunakan untuk melakukan pembangunan syntax yang sesuai dengan Javascript.

Berikut adalah implementasi dari PDA pada tugas ini:

```
< > / = " html head body title link rel href script src h1 h2 h3 h4 h5 h6 p br
em b abbr strong small hr div a img alt button type form action method input
table tr td th id class style ! -
Z < > = div G / A html head title link script body S h1 h2 h3 h4 h5 h6 p em b
abbr strong small img a form input button table rel link href id class style
src alt type method tr th td
Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE AF AG AH AI
AJ AK AL AM AN AO AP AQ AR AS AT AU AV AW AX AY AZ BA BB BC BD BE BF BG
A
A ! < C1 !,<
C1 - ! C1 -,!
C1 - - C1 C
C1 any-'-' C C1 C
C1 - C C2 e
C2 - ! C2 e
C2 > < A e
B ! < C3 !,<
C3 - ! C3 -,!
C3 - - C3 C
C3 any-'-' C C3 C
C3 - C C4 e
C4 - ! C4 e
C4 > < B e
C ! > C5 !,CC,>
C ! < C5 !,<
C5 - ! C5 -,!
C5 - - C5 C
C5 any-'-' C C5 C
C5 - C C6 e
C6 - ! C6 e
C6 > CC C G
C6 > < C e
D ! head C7 !,<,head
C7 - ! C7 -,!
C7 - - C7 C
C7 any-'-' C C7 C
C7 - C C8 e
C8 - ! C8 e
```



```

C8 > < D <
E ! < C9 !,<
C9 - ! C9 -,!
C9 - - C9 C
C9 any-'-' C C9 C
C9 - C C10 e
C10 - ! C10 e
C10 > CC E G
C10 > < E e
G ! > C11 !,CC,>
F ! < C11 !,<
C11 - ! C11 -,!
C11 - - C11 C
C11 any-'-' C C11 C
C11 - C C12 e
C12 - ! C12 e
C12 > CC F G
C12 > < F e
C12 > CC G G
AQ ! < C13 !,<
C13 - ! C13 -,!
C13 - - C13 C
C13 any-'-' C C13 C
C13 - C C14 e
C14 - ! C14 e
C14 > CC F G
C14 > < AQ e
AS ! < C15 !,<
C15 - ! C15 -,!
C15 - - C15 C
C15 any-'-' C C15 C
C15 - C C16 e
C16 - ! C16 e
C16 > CC F G
C16 > < AS e
AU ! < C17 !,<
C17 - ! C17 -,!
C17 - - C17 C
C17 any-'-' C C17 C
C17 - C C18 e
C18 - ! C18 e
C18 > CC F G
C18 > < AU e
AW ! < C19 !,<
AW ! > C19 !,CC,>
AW ! tr C19 !,CC,tr
C19 - ! C19 -,!
C19 - - C19 C
C19 any-'-' C C19 C
C19 - C C20 e
C20 - ! C20 e
C20 > < AW e
C20 > CC AW G
AX ! table C21 !,<,table
C21 - ! C21 -,!
C21 - - C21 C

```

```

C21 any-'-' C C21 C
C21 - C C22 e
C22 - ! C22 e
C22 > < AX >
H ! B C23 !,<,<,<B
H < Z H <,<Z
H ! < C23 !,<
C23 - ! C23 -,!
C23 - - C23 C
C23 any-'-' C C23 C
C23 - C C24 e
C24 - ! C24 e
C24 > < H e
A < Z A <,<Z
A html < A html,<
A > html B html
A id html P A,html
A class html P A,html
A style html P A,html
B < html B <,<html
B head < B head,<
B > head C head
B id head R A,head
B class head R A,head
B style head R A,head
C < head C <,<head
C title < C >,<title,<
C > > C G,>
C any-< G C G
C < G C e
C / > C e
C title title C e
C script script C e
C link < L link,<
C id > T A,>
C class > T A,>
C style > T A,>
C script < X >,<script,<
C / < K e
C > < D <
D < < D e
D / head D e
D head < D e
D title head C >,<title,<,<head
D link head L link,<,<head
D > html E B,html
D script head X >,<script,<,<head
E < B E <,<B
E body < E body,<
E > body F S,body
E id body AZ A,body
E class body AZ A,body
E style body AZ A,body
F < S F <,<S
F div < F >,<div,<
F h1 < F >,<h1,<

```

```

F h2 < F >,h2,<
F h3 < F >,h3,<
F h4 < F >,h4,<
F h5 < F >,h5,<
F h6 < F >,h6,<
F p < F >,p,<
F em < F >,em,<
F b < F >,b,<
F abbr < F >,abbr,<
F strong < F >,strong,<
F small < F >,small,<
F > > F G,>
F any-< G F G
F script < AY >,script,<
F br < N <
F hr < N <
F < G G e
F / < H e
F link < V link,<
F id > I A,>
F class > I A,>
F style > I A,>
F img < Z img,<
F a < Y >,a,<
F form < AG >,form,<
F input < AL input,<
F button < AB >,button,<
F table < AQ >,table,<
G / > G e
G div div G e
G h1 h1 G e
G h2 h2 G e
G h3 h3 G e
G h4 h4 G e
G h5 h5 G e
G h6 h6 G e
G p p G e
G em em G e
G b b G e
G a a G e
G abbr abbr G e
G strong strong G e
G small small G e
G button button G e
G form form G e
G script script G e
G > < G e
G < G G e
G br > N >
G hr > N >
G < S F <,S
G div > F >,div,<,G,>
G h1 > F >,h1,<,G,>
G h2 > F >,h2,<,G,>
G h3 > F >,h3,<,G,>
G h4 > F >,h4,<,G,>

```

```

G h5 > F >,h5,<,G,>
G h6 > F >,h6,<,G,>
G p > F >,p,<,G,>
G em > F >,em,<,G,>
G b > F >,b,<,G,>
G abbr > F >,abbr,<,G,>
G strong > F >,strong,<,G,>
G small > F >,small,<,G,>
G a > Y >,a,<,G,>
G img > Z img,<,G,>
G link > V link,<,G,>
G button > AB >,button,<,G,>
G form > AG >,form,<,G,>
G input > AL input,<,G,>
G table > AQ >,table,<,G,>
G script > AY >,script,<,G,>
H body S H e
H > body H e
H < < H e
H / B H e
H html html H e
H > < H e
I = A I =,A
I " = J e
J any-" A J A
J " A F e
K head head K e
K > < E B
L rel link L rel,link
L = rel M e
L = href M e
L = id M e
L = class M e
L = style M e
L id link BB A,link
L class link BB A,link
L style link BB A,link
M " link M A
M any-" A M A
M " A M e
M href < L href,link,<
M id < L id,link,<
M class < L class,link,<
M style < L style,link,<
M > < D <
N id < N A,<
N class < N A,<
N style < N A,<
N id > N A,>
N class > N A,>
N style > N A,>
N = A N =,A
N " = O e
N > < F e
N > > F G,>
O any-" A O A

```

```

O " A N e
P = A P =,A
P " = Q e
Q any-" A Q A
Q " A A e
R = A R =,A
R " = S e
S any-" A S A
S " A B e
T = A T =,A
T " = U e
U any-" A U A
U " A C e
V rel link V rel,link
V = rel W e
V = href W e
V = id W e
V = class W e
V = style W e
V id link BD A,link
V class link BD A,link
V style link BD A,link
W " link W A
W any-" A W A
W " A W e
W href < V href,link,<
W id < V id,link,<
W class < V class,link,<
W style < V style,link,<
W > < F e
X src > T A,>
X > > C G,>
Y href > I A,>
Y id > I A,>
Y class > I A,>
Y style > I A,>
Z src img Z src,img
Z = src AA e
Z = alt AA e
Z = id AA e
Z = class AA e
Z = style AA e
Z id img BF A,img
Z class img BF A,img
Z style img BF A,img
AA " img AA A
AA any-" A AA A
AA " A AA e
AA alt < Z alt,img,<
AA id < Z id,img,<
AA class < Z class,img,<
AA style < Z style,img,<
AA > < F e
AB id > AB A,>
AB class > AB A,>
AB style > AB A,>

```

```

AB = A AB =,A
AB " = AC e
AB type > AD type,>
AB > > F G,>
AC any-" A AC A
AC " A AB e
AD = type AD =,type
AD " = AE e
AE submit type AF type
AE reset type AF type
AE button type AF type
AF " type AB e
AG id > AG A,>
AG class > AG A,>
AG style > AG A,>
AG action > AG A,>
AG = A AG =,A
AG method > AH method,>
AG " = AK e
AG > > F G,>
AH = method AH =,method
AH " = AI e
AI GET method AJ method
AI POST method AJ method
AJ " method AG e
AK any-" A AK A
AK " A AG e
AL id input AL id,input
AL class input AL class,input
AL style input AL style,input
AL type input AN type,input
AL = id AM e
AL = class AM e
AL = style AM e
AM " input AM A
AM any-" A AM A
AM " A AM e
AM id < AL id,input,<
AM class < AL class,input,<
AM style < AL style,input,<
AM type < AN type,input,<
AM > < F e
AN = type AO e
AO " input AO e
AO text < AP <
AO password < AP <
AO email < AP <
AO number < AP <
AO checkbox < AP <
AP " < AM <
AN = type AO e
AO " input AO e
AO text < AP <
AO password < AP <
AO email < AP <
AO number < AP <

```

```

AO checkbox < AP <
AP " < AM <
AQ id > AQ A,>
AQ class > AQ A,>
AQ style > AQ A,>
AQ = A AQ =,A
AQ " = AR e
AQ > > AS >
AR any-" A AR A
AR " A AQ e
AS < > AS <,>
AS tr < AS tr,<
AS id tr AS A,tr
AS class tr AS A,tr
AS style tr AS A,tr
AS = A AS =,A
AS " = AT e
AS > tr AU >,tr
AS / < BH e
AT any-" A AT A
AT " A AS e
AU < > AU <,>
AU th < AU th,<
AU id th AU A,th
AU class th AU A,th
AU style th AU A,th
AU = A AU =,A
AU " = AV e
AU > th AW G,>,th
AU td < AU td,<
AU id td AU A,td
AU class td AU A,td
AU style td AU A,td
AU = A AU =,A
AU " = AV e
AU > th AW G,>,th
AU > td AW G,>,td
AU / < BK e
AV any-" A AV A
AV " A AU e
AW any-< G AW G
AW < G AW e
AW / > AW e
AW th th AW e
AW td td AW e
AW > < AW e
AW < > AW e
AW / tr AW e
AW tr < AW e
AW th tr AU th,<,>,tr
AW td tr AU td,<,>,tr
AW > > AX >
AX < > AX e
AX / table AX e
AX table < AX e
AX > S F S

```

```

AX tr table AS tr,<,>,table
AX < G G e
AY src > I A,>
AY id > I A,>
AY class > I A,>
AY style > I A,>
AY > > F G,>
AZ = A AZ =,A
AZ " = BA e
BA any-" A BA A
BA " A E e
BB = A BC e
BC " link BC A,link
BC any-" A BC A
BC " A L e
BD = A BE e
BE " link BE A,link
BE any-" A BE A
BE " A V e
BF = A BG e
BG " img BG A,img
BG any-" A BG A
BG " A Z e
BH table > BH e
BH > table BI e
BI e < BJ e
BJ < G G e
BK tr > BK e
BK > tr BL e
BL e < BM e
BM < > AX e

```

B. Implementasi Program

Terdapat dua folder dalam program kami, yaitu:

a. doc

Folder `doc` berisi laporan Tugas Besar Teori Bahasa Formal Otomata mengenai *HTML Language Checker*.

b. src

Folder `src` berisi source code dari program kami. Di dalam folder ini juga terdapat file `main.py` yang berfungsi sebagai “gerbang utama” program kami. File ini berisi algoritma yang dapat mengolah PDA yang dimasukkan pengguna dan menerapkannya untuk membaca sintaks dari file html yang juga dimasukkan pengguna. Selain itu, juga terdapat tiga folder lainnya dalam folder `src` yaitu:

1. functions

Terdapat dua kode dalam folder ini, yaitu `load.py` dan `readpda.py`. Di dalam `load.py` terdapat algoritma untuk membaca input file konfigurasi PDA dan juga file html dengan menggunakan library `argparse`. Sementara di dalam `readpda.py` terdapat algoritma untuk

mengkonversi file konfigurasi PDA yang dibaca oleh `'load.py'` kedalam class buatan sehingga dapat digunakan di file `'main.py'`.

2. Html

Folder ini berisi input file html yang sintaksnya akan diperiksa oleh program yang kami buat. Kami menggunakan 4 (empat) tipe file HTML yang akan diperiksa seperti `'input.html'`, `'inputAcc.html'`, `'inputReject.html'`, dan `'testing.html'`. `'input.html'` merupakan sebuah file HTML yang digunakan untuk mengecek apakah seluruh *tags* dan *attributes* telah ditangani oleh program. `'inputAcc.html'` berisi file HTML dengan hasil yang diharapkan keluar dari program adalah *accepted*.

3. PDA

Folder ini berisi file konfigurasi PDA yang akan digunakan. Di dalam folder ini terdapat 4 (empat) file seperti `'pda.txt'`, `'pda.txt guide.md'`, `'pda.drawio'`, `'pda.png'`, dan `'pda.txt'`. `'pda.txt guide.md'` merupakan file yang memberikan panduan mengenai file `'pda.txt'` yang telah dirancang. `'pda.txt'` berisi logika utama dari program *HTML Language Checker* ini. `'pda.drawio'` dan `'pda.png'` merupakan file yang menunjukkan diagram alur kerja PDA program ini. Hal yang membedakan antara `'pda.drawio'` dan `'pda.png'` adalah format filenya.

BAB IV PROGRAM UTAMA

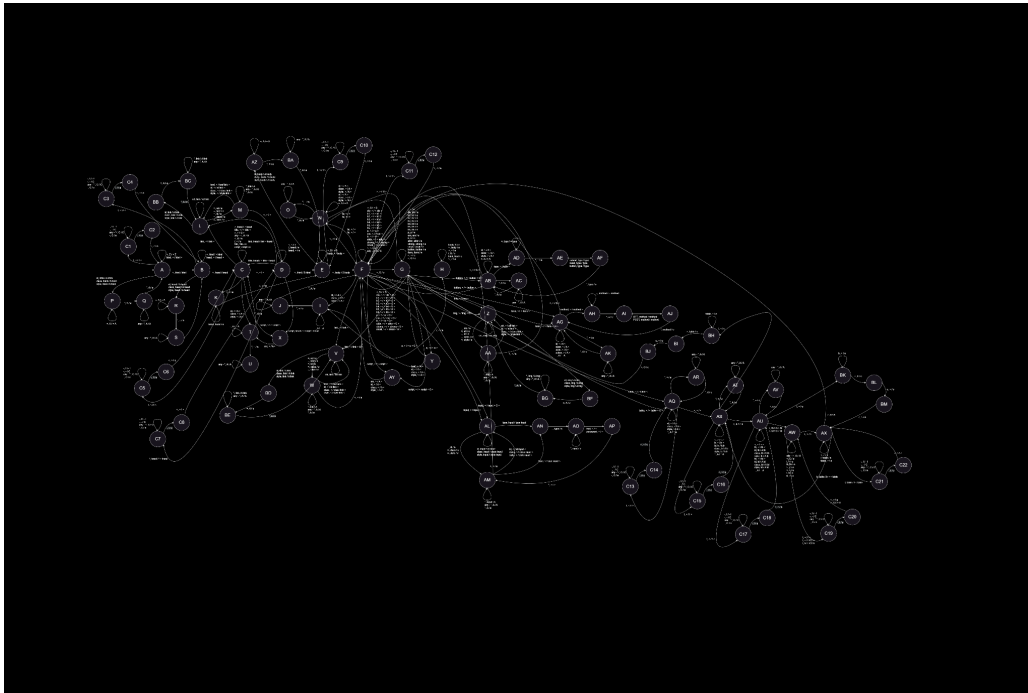
Dalam penanganan tugas besar ini, kami memutuskan untuk menggunakan Pushdown Automaton (PDA) dengan tipe Acceptance by Empty Stack (AES). Keputusan ini didasarkan pada pertimbangan bahwa implementasi PDA AES jauh lebih mudah dibandingkan dengan PDA tipe Acceptance by Final State (AFS).

Dalam PDA AFS, setiap state harus didefinisikan dengan keadaan yang sedang berlangsung, yang bisa mengakibatkan kompleksitas yang tinggi karena memerlukan banyak state. Sebaliknya, PDA AES memungkinkan pendekatan implementasi yang lebih sederhana. Kami memilih untuk tidak menggunakan PDA AFS karena dapat membutuhkan pengelolaan state yang berlebihan, yang dapat menyulitkan pemahaman dan pemeliharaan program.

Dengan menggunakan PDA AES, kami dapat mengadopsi pendekatan brute force atau coba-coba dalam proses implementasinya. Meskipun terkesan lebih sederhana, namun tetap memastikan akurasi dalam pengecekan sintaksis. Keuntungan utama dari penggunaan PDA AES adalah jumlah state yang dibutuhkan jauh lebih sedikit dibandingkan dengan PDA AFS. Hal ini dapat signifikan mempermudah proses implementasi dan pemeliharaan, serta mengurangi kompleksitas program secara keseluruhan.

Dalam implementasi PDA AES, fokus utama akan diberikan pada mekanisme pengecekan menggunakan tumpukan dan bagaimana kondisi stack kosong menunjukkan penerimaan. Pendekatan ini dirancang untuk memberikan keseimbangan antara efisiensi dan kemudahan implementasi, sehingga proyek dapat berkembang dengan baik dan mudah dikelola oleh tim pengembang.

Berikut adalah PDA yang kami buat.



Gambar 3 PDA untuk memeriksa sintaks HTML

BAB V

PENGUJIAN PROGRAM

Kami menggunakan PDA tersebut untuk memeriksa sintaks pada contoh input file html yang ada di spesifikasi yaitu `inputAcc.html`.

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

Hasil keluarannya adalah sebagai berikut.

```
PS C:\Users\ASUS\Documents\Project\Tubes-TBFO-2023\src> python main.py pda.txt inputAcc.html
```

File berhasil dimuat!

Accepted

Gambar 4 Hasil pemeriksaan sintaks inputAcc.html

Setelah itu kami membuat file yang akan ditolak oleh PDA kami yang kami beri nama `inputReject.html`. Berikut adalah isi dari file tersebut.

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

Dapat dilihat pada line 7 terdapat kesalahan sintaks dimana tag <p tidak memiliki penutup di kanannya. Hasil keluarannya adalah sebagai berikut.

```
PS C:\Users\ASUS\Documents\Project\Tubes-TBFO-2023\src> python main.py pda.txt inputReject.html
```

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 7:

```
<pThis is a simple webpage.</p>
```

Gambar 5 Hasil pemeriksaan sintaks inputReject.html

Setelah melakukan masing-masing satu contoh untuk input *accepted* dan input *rejected*, kami akan melakukan *test case* tambahan. Berikut adalah variasi test case yang kami gunakan:

1. **Rejected:** Head muncul setelah body

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

PS C:\Users\aland\Desktop\Tubes-TBF0-2023\src> python main.py pda.txt variasi1.html

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 2:

```
<body>
```

Gambar 6 Hasil pemeriksaan sintaks variasi1.html

2. **Rejected:** tag pertama bukan html

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

PS C:\Users\aland\Desktop\Tubes-TBF0-2023\src> python main.py pda.txt variasi2.html

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 1:

```
hmif>
```

Gambar 7 Hasil pemeriksaan sintaks variasi2.html

3. **Rejected:** Tidak punya head

```
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
</html>
```

```
PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi3.html
```

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 2:

```
<body>
```

Gambar 8 Hasil pemeriksaan sintaks variasi3.html

4. Accepted

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    
    <p>This is a <em>simple</em> webpage.</p>

    <div id="footer" class="footer"> This is the end of the page </div>
  </body>
</html>
```

```
PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi4.html
```

File berhasil dimuat!

Accepted

Gambar 9 Hasil pemeriksaan sintaks variasi4.html

5. Accepted: Penanganan komentar pada sintaks

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <hr>
    
    <p>This is a <em>simple</em> webpage.</p>

    <!-- Custom element -->
    <div id="footer" class="footer"> This is the end of the page </div>
  </body>
</html>
```

```
PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi5.html
```

File berhasil dimuat!

Accepted

Gambar 10 Hasil pemeriksaan sintaks variasi5.html

6. Rejected: Atribut wajib src pada img tidak ada

```
<html>
<head>
  <title>Simple Webpage</title>
</head>
<body>
  <!-- Bagian utama web -->
  <h1>Hello, World!</h1>
  <h2>Welcome to my page</h2>
  <img alt="Welcome Banner">
  <p>This is a <em>simple</em> webpage.</p>

  <!-- Custom element -->
  <div id="footer" class="footer"> This is the end of the page </div>
</body>
</html>
```

```
PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi6.html
```

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 9:

```
<img alt="Welcome Banner">
```

Gambar 11 Hasil pemeriksaan sintaks variasi6.html

7. Accepted

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

<h2>HTML Forms</h2>

<form action="/action_page.php" method="POST">
  <h5 class="label">First name:</h5><br>
  <input type="text" id="fname"><br>
  <h5 class="label">Last name:</h5><br>
  <input type="text" id="lname"><br><br>
  <button type="submit">Submit</button>
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>
```

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

PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi7.html

File berhasil dimuat!

Accepted

Gambar 11 Hasil pemeriksaan sintaks variasi7.html

8. **Rejected:** tidak termasuk value yang diperbolehkan (POST, GET)

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

</head>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php" method="TEBAK">
  <div id="label">First name:</div><br>
  <input type="text" id="fname"><br>
  <div id="label">Last name:</div><br>
  <input type="text" id="lname"><br><br>
  <button type="submit">Submit</button>
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>

</body>
</html>
```

PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi8.html

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 10:

```
<form action="/action_page.php" method="TEBAK">
```

Gambar 12 Hasil pemeriksaan sintaks variasi8.html

9. **Accepted**

```
<html>
<head>
  <title>Simple Webpage</title>
  <script>
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
  </script>
</head>
<body>
```

```

<h1>The script element</h1>

<p id="demo"></p>

</body>
</html>

```

PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi9.html

File berhasil dimuat!

Accepted

Gambar 13 Hasil pemeriksaan sintaks variasi9.html

10. **Rejected:** <p> bukan void element, wajib ditutup

```

<html>
<head>
  <title>Simple Webpage</title>
  <script>
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
  </script>
</head>
<body>

<h1>The script element</h1>

<p id="demo">

</body>
</html>

```

PS C:\Users\aland\Desktop\Tubes-TBFO-2023\src> python main.py pda.txt variasi10.html

File berhasil dimuat!

Syntax Error

Terdapat kesalahan sintaks pada line 14:

```
</body>
```

Gambar 14 Hasil pemeriksaan sintaks variasi10.html

11. **Accepted**

```

<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>
```

PS C:\Users\aland\Desktop\Tubes-TBF0-2023\src> python main.py pda.txt variasi11.html

File berhasil dimuat!

Accepted

Gambar 15 Hasil pemeriksaan sintaks variasi11.html

BAB VI

KESIMPULAN DAN SARAN

A. Kesimpulan

PDA adalah salah satu alat komputasi yang sangat kuat dan efektif dalam melakukan parsing suatu bahasa. Dengan kemampuan untuk menyimpan informasi dalam bentuk tumpukan, PDA mampu mengenali dan menganalisis struktur sintaksis dari bahasa formal dengan presisi tinggi. Keunggulan utama PDA terletak pada kemampuannya dalam mengenali tata bahasa kontekstual, sehingga sering digunakan dalam bidang pemrosesan bahasa alami, kompilasi bahasa pemrograman, dan berbagai aplikasi lain yang memerlukan analisis sintaksis yang cermat dan efisien.

Selain itu, PDA juga menunjukkan fleksibilitas yang tinggi dalam menangani berbagai jenis bahasa formal, termasuk bahasa yang lebih kompleks. Dengan konsep tumpukan (stack) yang memungkinkan penyimpanan sementara informasi, PDA dapat melibatkan diri dalam analisis bahasa yang lebih rumit, seperti bahasa yang memerlukan pengecekan kondisi atau pembacaan input secara iteratif.

Keberagaman penggunaan PDA mencakup aplikasi di dalam dunia pengembangan perangkat lunak, terutama dalam fase kompilasi bahasa pemrograman. Proses parsing yang dilakukan oleh PDA membantu dalam memastikan bahwa kode sumber program sesuai dengan struktur sintaksis yang benar, sehingga meminimalkan kesalahan selama proses kompilasi.

Secara keseluruhan, PDA adalah alat komputasi yang berdaya tinggi, dapat diandalkan, dan serbaguna dalam menangani analisis sintaksis bahasa formal. Keberhasilannya dalam berbagai aplikasi mencerminkan kontribusi pentingnya dalam dunia komputasi dan pengembangan perangkat lunak.

B. Saran

Saran untuk tugas besar ini adalah:

- a. Tahapan yang terencana secara sistematis dalam pengerjaan tugas besar dapat memberikan keuntungan signifikan dalam memudahkan proses kerja. Untuk meningkatkan efisiensi ke depannya, disarankan untuk memberikan prioritas pada tugas yang melibatkan penggambaran diagram. Langkah ini bertujuan untuk menghindari kompleksitas yang mungkin muncul dalam implementasi program, sehingga memfasilitasi pemahaman dan kerja sama tim secara lebih efektif.
- b. Langkah-langkah yang terencana dengan baik dalam pengembangan proyek dapat membawa manfaat yang signifikan. Oleh karena itu, disarankan untuk menghindari penambahan atau perubahan secara mendadak dalam spesifikasi program. Tindakan ini perlu dihindari karena dapat menyebabkan perubahan besar dalam struktur kode yang sudah ada.

BAB VI PEMBAGIAN KERJA

NIM	Nama	Pekerjaan
13522124	Aland Mulia Pratama	<ul style="list-style-type: none"> • Membuat readme • Membuat PDA • Converter dari pda.txt ke class buatan
13522126	Rizqika Mulia Pratama	<ul style="list-style-type: none"> • Membuat readme • Membuat PDA • Membuat algoritma pengolahan PDA
13522147	Ikhwan Al Hakim	<ul style="list-style-type: none"> • Membuat argparse untuk menerima input file • Membuat PDA • Membuat diagram PDA

DAFTAR PUSTAKA

Tautan Repository Github: <https://github.com/Nerggg/Tubes-TBFO-2023>

Tautan Diagram State:

<https://drive.google.com/file/d/1Z1RdXXst4m55tQXxSkzxrRiNIXlTy8he/view?usp=sharing>

Hasan, R. (2013, March 6). *Pengertian HTML - Rahman Hasan - Universitas negeri gorontalo*. mahasiswa.ung.ac.id.

https://mahasiswa.ung.ac.id/281412091/home/2013/3/6/pengertian_html.html

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2022). *Introduction to automata theory, languages, and computation*. Pearson Education.

Sudiadi, & Teguh, R. (2017). *Diktat Teori Bahasa Dan Otomata STMIK-MDP*.

Guntoro. (2022, March 20). *√ 65 Tag HTML dilengkapi Fungsi dan Contoh Penggunaannya*.

Badoy Studio. <https://badoystudio.com/tag-html/>