Laporan Tugas Besar IF2124 Teori Bahasa Formal dan Otomata

HTML Checker dengan Pushdown Automata (PDA)

Semester I Tahun 2023/2024

oleh

1. Maximilian Sulistiyo 13522061
2. Marvel Pangondian 13522075
3. Andhika Tantyo Anugrah 13522094

Kelompok Ejaan Yang diFormalkan (EYF)

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

Daftar Isi

Daftar Isi 2
BAB 1
Latar Belakang 3
BAB 2
Teori Dasar 4
2.1 Teori Push Down Automaton 4
2.2 Pengaplikasiannya PDA pada Sintaksis HTML 5
2.3 Syntax HTML yang Perlu Diperhatikan Dalam Pembuatan PDA 6
BAB 3
Hasil Pushdown Automata (PDA) 8
3.1 Pushdown Automata (PDA) (not final version, final version can be seen in GitHub) 8
3.2 Stack Symbol & Transition Symbol 16
3.3 Tokenisasi 17
BAB 4
Implementasi 19
4.1 Program main.py 19
4.2 Program tag2_symbol.py 19
4.3 Program miscellaneous.py 19
BAB 5
Eksperimen 20
5.1 Accepted Files 20
5.2 Rejected Files 30
BAB 6
GitHub Repository 37
BAB 7
Diagram State 38
(diagram does not contain final version of PDA, final version can be seen in GitHub) 38
BAB 8
Pembagian Tugas 39
Daftar Pustaka 40

BAB 1
Latar Belakang

Parsing merupakan proses menganalisis teks dan menentukan apakah teks tersebut termasuk ke dalam language yang ditentukan atau tidak. Ada langkah-langkah parsing melakukan hal tersebut. Pertama dia memisahkan teks menjadi urutan token, token sendiri adalah sebuah unit paling sederhana dari sebuah sintaks contohnya adalah operator, identifier, atau keyword. Hal ini sama seperti bahasa natural dimana kita membagi-bagi kalimat menjadi subjek, predikat, objek, atau keterangan. Setelah itu sebuah parser akan struktur grammatik yang sesuai dengan kaidah bahasa formal yang dimiliki.

Dengan cara tersebut pun parser dapat mengetahui apakah sebuah teks dapat diterima oleh bahasa yang dimiliki dengan cara menganalisis setiap token dan menentukan apakah urutan token mengikut sintaks bahasa yang dipilih. Jika urutan token sesuai dengan aturan grammatik maka proses dilanjutkan. Sedangkan jika terdapat urutan yang salah maka parser akan mengidentifikasi masalah tersebut. Jika tidak ada masalah maka parser akan membuat struktur data yang kemudian akan digunakan oleh kompiler untuk membuat executable file.

Tentu hal ini sangat berguna pada dunia programming dimana sebuah compiler memerlukan sebuah teks yang sudah dipastikan benar agar dapat membuat executable file. Pada tugas besar ini kami ditugaskan untuk memeriksa apakah sebuah file html memiliki sintaks yang benar agar dapat digunakan di website.

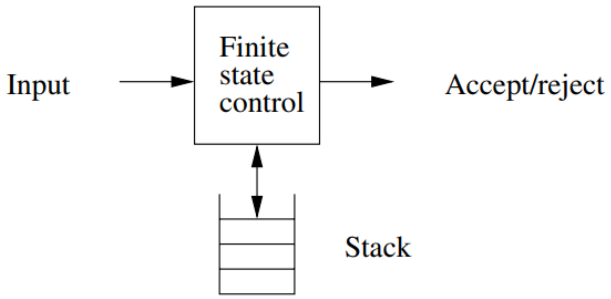
BAB 2
Teori Dasar

2.1 Teori Push Down Automaton

Pushdown automaton adalah perpanjangan dari *nondeterministic finite automaton* (NFA) dengan transisi epsilon (ϵ), yang merupakan salah satu cara untuk mendefinisikan *regular languages*. Secara sederhana, *pushdown automaton* adalah ϵ-NFA dengan tambahan *stack*. *Stack* ini sama seperti struktur data *stack*, hanya elemen paling atasnya saja (*top of the stack*) yang dapat dibaca (*read*), di-*push* (*pushed*), dan dikeluarkan (*popped*).

PDA deterministik, subkelas dari PDA, pertama kali dikemukakan oleh P. C. Fischer dan M. P. Schutzenberger. PDA ini nantinya menjadi penting karena digunakan sebagai model dari *parsers*. Terutama mengenai pengenalan "*LR(k) grammars,*" subkelas dari CFG yang menghasilkan bahasa deterministik PDA. *Grammar LR(k)* menjadi basis dari YACC, sebuah kakas untuk memproduksi *parser*.

PDA deterministik dapat menerima semua *regular languages*, namun hanya terbatas pada *proper subset* dari *context-free languages* (CFL). Karena PDA deterministik ini menyerupai cara kerja sebuah *parser* dalam sebuah *compiler*, pengamatan mengenai bahasa apa yang dapat dan tidak dapat dikenali oleh PDA deterministik menjadi penting.



Gambar 2.1.1 Contoh ilustrasi Pushdown Automaton
Sumber: Hopcroft, John E., Rajeev Motwani, Jeffrey D. Ullman. 2003. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley.

Dalam satu transisi, PDA melakukan hal berikut:

1. Mengkonsumsi sebuah simbol untuk melakukan transisi, jika simbol tersebut Ɛ maka tidak ada simbol yang dikonsumsi
2. Berpindah state, baik ke state berbeda maupun ke state yang sudah dilalui sebelumnya
3. Mengubah simbol stack yang berada di paling atas menjadi simbol stack apapun (tergantung pada fungsi transisi).

Notasi formal sebuah *Pushdown Automaton* (PDA) adalah sebagai berikut:

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

1. Q: *Finite states* (seperti q0, q1, dll)
2. $\sum$: *Input alphabet* (seperti a, b, c, dll)
3. Γ: *Stack alphabet* (seperti A, B, C, dll)
4. δ: *Transition function*, yakni sifat/aturan pada automaton
5. q_0: *Start state*

6. Z_0: *Start symbol*
7. *F*: *set of accepting states*

PDA memiliki 2 tipe, yakni PDA yang menerima *by final state*, dan PDA yang menerima *by empty stack*. PDA yang menerima berdasarkan final state adalah PDA yang menerima sebuah language ketika input sudah kosong dan state pada PDA adalah salah satu *final state*, berikut adalah contoh PDA yang menerima language dengan *final state*:
Informal notation (notasi yang akan digunakan pada PDA program penulis):

```
Q Q1 Q2         # total states
a               # input word symbols
K               # stack symbols
Q               # starting state
K               # starting stack
Q2              # accepting states
F               # E - accepts with empty stack or F - accepts with accepting state
Q e K Q1 K      #transition functions
Q1 a K Q2 K
Q2 a K Q1 K
```

Dengan notasi formal: P = ({Q,Q1,Q2},{a},{K},δ,Q,K,{Q2}). *Grammar* tersebut menerima bahasa dengan jumlah a yang ganjil seperti a, aaa, aaaaa, dll.

PDA tipe berikutnya adalah PDA yang menerima sebuah bahasa ketika stack dalam PDA tersebut kosong. PDA ini menerima sebuah language ketika input sudah kosong dan *stack* di PDA itu kosong (Z_0) sudah dikeluarkan (*popped*), berikut adalah contoh PDA yang menerima language dengan *empty state*:

$\delta(q,\varepsilon,S) = \{(q,0S1), (q,A)\}$
$\delta(q,\varepsilon,A) = \{(q,1A0), (q,S), (q,\varepsilon)\}$
$\delta(q,0,0) = \{(q,\varepsilon)\}$
$\delta(q,1,1) = \{(q,\varepsilon)\}$

Dengan notasi formal: P = ({q},{0,1},{0,1,A,S},δ,q,S). *Grammar* tersebut menerima bahasa dengan jumlah 0 dan 1 yang sama seperti 01, 0011, 1001, dll.

2.2 Pengaplikasiannya PDA pada Sintaksis HTML

PDA dalam dunia *computer science* digunakan untuk memodelkan/memecahkan sintaks dari sebuah bahasa formal. Salah satu kegunaan PDA adalah menggunakan sebuah *syntax checker* pada sintaksis HTML. Alasan PDA dapat digunakan sebagai *syntax checker* HTML adalah sebagai berikut:

1. Dapat menangani struktur bersarang (*nested structure*) HTML.
   HTML memiliki struktur bersarang yang cukup kompleks, dengan menggunakan PDA, struktur tersebut dapat ditangani dengan menggunakan state dan stack.
2. Penanganan atribut
   Pada HTML, tags dapat memiliki atribut dengan value-value tersebut. PDA dapat menangani hal tersebut dengan memproses atribut tersebut ([nama_atribut, "=", value_atribut]) menjadi token yang dapat dicek kebenarannya menggunakan perpindahan state dan stack.
3. Design yang fleksibel
   PDA dapat dirancang untuk memenuhi persyaratan spesifik dari aturan sintaksis HTML. Aturan transisi pada PDA dapat disusun dengan cermat untuk memeriksa setiap aspek dari struktur HTML yang diinginkan.
4. Penangan kesalahan sintaks

PDA dapat mencari kesalah sintaks dari sebuah HTML berdasarkan dengan isi stack dan current state. Misalnya PDA menerima <input type = “eve”>, parser akan mentokenisasi sintaks tersebut menjadi [input,type,=,”eve’], kemudian PDA, dengan grammar yang ketatnya, akan mengkonsumsi token-token tersebut, ketika mencapai token “eve”, tidak ada aturan transisi yang sesuai dengan value tersebut sehingga PDA akan mendeteksi kesalahan sintaks.

Dalam pembuatan PDA yang dapat mengecek sintaks HTML, terdapat beberapa langkah-langkah yang perlu dilakukan yakni:

1. *Grammar*
   Sebelum mengimplementasi PDA dalam bentuk kode, perlu dibuat grammar PDA tersebut. Grammar PDA perlu mencakup aturan-aturan untuk elemen, atribut, dan struktur penanaman mereka. Grammar ini dapat digunakan untuk membuat aturan produksi untuk PDA.
2. Input dan *stack symbol*
   Dalam pembuatan grammar dan aturan produksi, perlu ditentukan input serta stack symbol apa saya yang akan digunakan dalam pembuatan PDA. Input PDA dapat berupa file html yang sudah diubah menjadi token (berupa *array of string*) yang dapat PDA konsumsi. Penamaan stack symbol dapat ditentukan sesuai dengan yang ingin pembuat, asalkan sesuai dengan aturan transisi yang akan dibuat.
3. Aturan transisi
   Setelah menentukan input dan stack symbol, perlu dibuat aturan transisi. Idealnya (pada umumnya), macam-macam input dan stack symbol ditentukan sejalan membuat aturan transisi. Aturan transisi harus sesuai dengan aturan sintaks HTML Pembuatan aturan transisi juga memperhatikan susunan tag dan *nested structure* HTML yang benar.
4. Implementasi
   Setelah membuat aturan transisi, maka perlu membuat program untuk mengimplementasikan PDA tersebut. Implementasi PDA harus mempertimbangkan bahasa yang akan digunakan. Implementasi PDA juga harus memperhatikan data struktur yang akan digunakan.
5. Uji coba
   Hasil implementasi PDA perlu diuji coba, PDA dapat diuji dengan beberapa tipe file test HTML yakni beberapa file dengan sintaks yang salah dan sintaks yang benar.

2.3 Syntax HTML yang Perlu Diperhatikan Dalam Pembuatan PDA

HTML merupakan tulang belakang dari sebuah Web. HTML berisi content dari sebuah web berupa teks, heading, paragraf, list, dll. HTML memiliki *nested structure* yang kompleks. Sintaks yang perlu diperhatikan dalam HTML adalah:

1. *White Space*
   Dalam pembentukan kode HTML, terdapat situasi dimana kode HTML tidak rapih (terdapat white space pada tag HTML) contoh : , walaupun tag tersebut tidak dalam bentuk formal, tetapi tag tersebut tetap valid sehingga perlu diatasi.
2. *Nested Structure*
   HTML memiliki struktur bersarang yang kompleks sehingga PDA harus dapat menangani hal tersebut, contoh *nested structure* adalah : hello , PDA harus dapat menangani hal tersebut.
3. *Attributes and values*
   Dalam tag HTML, terdapat attributes dan values tertentu yang harus diatasi. Sebagai contoh ada tag img harus memiliki atribut src, atau pada tag form, value pada atribut method harus dibatasi “GET” dan “POST” saja.

4. Restriksi
   Pada sintaks HTML terdapat beberapa restriksi yang perlu diikuti yakni lokasi teks yang yang tidak bisa di dalam tag-tag tertentu, jenis-jenis atribut yang memiliki value-value tertentu, tag-tag yang memiliki atribut tertentu, jenis tag berupa void tag atau tidak, dan lain-lain.

BAB 3
Hasil Pushdown Automata (PDA)

3.1 Pushdown Automata (PDA) (**not final version, final version can be seen in GitHub**)

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

A Finite set of states (Q) *= q0 q1 q2 q3 q4 qimg qlink*
***A Finite set of input symbols* (∑)** = html head body /html /head /body title /title link script /script h1 /h1 h2 /h2 h3 /h3 h4 /h4 h5 /h5 h6 /h6 p /p br em /em b /b abbr /abbr strong /strong small /small hr /hr div /div a /a img /img button /button form /form input /input table /table tr /tr td /td th /th
***Finite stack alphabets* (Γ)** = A B C D F G H I J K L M N O P Q R S T U V W X Y ! @ # 1 2 [ ] { } ? a b c d f g h i j k l m n o p q r s t u v 3 4 5 6 7 *
***Transition Functions* (*δ*)**:
δ(q0, html, Z) = (q1, aZ)
δ(q1, id, a) = (q1, 1a)
δ(q1, =, 1) = (q1, 2)
δ(q1, TEXT, 2) = (q1, e)
δ(q1, class, a) = (q1, 1a)
δ(q1, style, a) = (q1, 1a)
δ(q1, >, a) = (q1, A)
δ(q1, head, A) = (q2, bA)
δ(q2, id, b) = (q1, 1b)
δ(q2, =, 1) = (q1, 2)
δ(q2, TEXT, 2) = (q1, e)
δ(q2, class, b) = (q1, 1b)
δ(q2, style, b) = (q1, 1b)
δ(q2, >, b) = (q2, B)
δ(q2, title, B) = (q2, dB)
δ(q2, id, d) = (q2, 1d)
δ(q2, =, 1) = (q2, 2)
δ(q2, TEXT, 2) = (q2, e)
δ(q2, class, d) = (q2, 1d)
δ(q2, style, d) = (q2, 1d)
δ(q2, >, d) = (q2, D)
δ(q2, TEXT, D) = (q2, D)
δ(q2, /title, D) = (q2, e)
δ(q2, em, D) = (q2, 3D)
δ(q2, b, D) = (q2, 4D)
δ(q2, abbr, D) = (q2, 5D)
δ(q2, strong, D) = (q2, 6D)
δ(q2, small, D) = (q2, 7D)
δ(q2, link, B) = (qlink, w)
δ(qlink, id, w) = (qlink, 1w)
δ(qlink, =, 1) = (qlink, 2)
δ(qlink, TEXT, 2) = (qlink, e)
δ(qlink, class, w) = (qlink, 1w)

δ(qlink, style, w) = (qlink, 1w)
δ(qlink, href, w) = (qlink, 1w)
δ(qlink, rel, w) = (qlink, 1w)
δ(qlink, TEXT, 2) = (q2, e)
δ(q2, id, w) = (q2, 1w)
δ(q2, class, w) = (q2, 1w)
δ(q2, style, w) = (q2, 1w)
δ(q2, href, w) = (q2, 1w)
δ(q2, >, w) = (q2, e)
δ(q2, script, B) = (q2, xB)
δ(q2, id, x) = (q2, 1x)
δ(q2, class, x) = (q2, 1x)
δ(q2, style, x) = (q2, 1x)
δ(q2, src, x) = (q2, 1x)
δ(q2, >, x) = (q2, X)
δ(q2, TEXT, X) = (q2, X)
δ(q2, /script, X) = (q2, e)
δ(q2, em, X) = (q2, 3D)
δ(q2, b, X) = (q2, 4D)
δ(q2, abbr, X) = (q2, 5D)
δ(q2, strong, X) = (q2, 6D)
δ(q2, small, X) = (q2, 7D)
δ(q2, /head, B) = (q3, e)
δ(q3, body A) = (q3, cA)
δ(q3, id, c) = (q3, 1c)
δ(q3, =, 1) = (q3, 2)
δ(q3, TEXT, 2) = (q3, e)
δ(q3, class, c) = (q3, 1c)
δ(q3, style, c) = (q3, 1c)
δ(q3, >, c) = (q3, C)
δ(q3, link, C) = (qlink, w)
δ(qlink, class, 1) = (qlink, 1w)
δ(qlink, TEXT, 2) = (q3, e)
δ(q3, id, w) = (q3, 1w)
δ(q3, class, w) = (q3, 1w)
δ(q3, style, w) = (q3, 1w)
δ(q3, href, w) = (q3, 1w)
δ(q3, >, w) = (q3, e)
δ(q3, link, N) = (q3, wN)
δ(q3, link, U) = (q3, wN)
δ(q3, script, C) = (q3, x)
δ(q3, id, x) = (q3, 1x)
δ(q3, class, x) = (q3, 1x)
δ(q3, style, x) = (q3, 1x)
δ(q3, src, x) = (q3, 1x)
δ(q3, >, x) = (q3, X)
δ(q3, TEXT, X) = (q3, X)
δ(q3, /script, X) = (q3, e)

δ(q3, em, X) = (q3, 3X)
δ(q3, b, X) = (q3, 4X)
δ(q3, abbr, X) = (q3, 5X)
δ(q3, strong, X) = (q3, 6X)
δ(q3, small, X) = (q3, 7X)
δ(q3, script, N) = (q3, xN)
δ(q3, script, U) = (q3, xU)
δ(q3, h1, C) = (q3, fC)
δ(q3, id, f) = (q3, 1f)
δ(q3, class, f) = (q3, 1f)
δ(q3, style, f) = (q3, 1f)
δ(q3, >, f) = (q3, F)
δ(q3, TEXT, F) = (q3, F)
δ(q3, /h1, F) = (q3, e)
δ(q3, em, F) = (q3, 3F)
δ(q3, b, F) = (q3, 4F)
δ(q3, abbr, F) = (q3, 5F)
δ(q3, strong, F) = (q3, 6F)
δ(q3, small, F) = (q3, 7F)
δ(q3, h1, N) = (q3, fN)
δ(q3, h1, U) = (q3, fU)
δ(q3, h2, C) = (q3, gC)
δ(q3, id, g) = (q3, 1g)
δ(q3, class, g) = (q3, 1g)
δ(q3, style, g) = (q3, 1g)
δ(q3, >, g) = (q3, G)
δ(q3, TEXT, G) = (q3, G)
δ(q3, /h2, G) = (q3, e)
δ(q3, em, G) = (q3, 3G)
δ(q3, b, G) = (q3, 4G)
δ(q3, abbr, G) = (q3, 5G)
δ(q3, strong, G) = (q3, 6G)
δ(q3, small, G) = (q3, 7G)
δ(q3, h2, N) = (q3, gN)
δ(q3, h2, U) = (q3, gU)
δ(q3, h3, C) = (q3, hC)
δ(q3, id, h) = (q3, 1h)
δ(q3, class, h) = (q3, 1h)
δ(q3, style, h) = (q3, 1h)
δ(q3, >, h) = (q3, H)
δ(q3, TEXT, H) = (q3, H)
δ(q3, /h3, H) = (q3, e)
δ(q3, em, H) = (q3, 3H)
δ(q3, b, H) = (q3, 4H)
δ(q3, abbr, H) = (q3, 5H)
δ(q3, strong, H) = (q3, 6H)
δ(q3, small, H) = (q3, 7H)
δ(q3, h3, N) = (q3, hN)

δ(q3, h3, U) = (q3, hU)
δ(q3, h4, C) = (q3, iC)
δ(q3, id, i) = (q3, 1i)
δ(q3, class, i) = (q3, 1i)
δ(q3, style, i) = (q3, 1i)
δ(q3, >, i) = (q3, I)
δ(q3, TEXT, I) = (q3, I)
δ(q3, /h4, I) = (q3, e)
δ(q3, em, I) = (q3, 3I)
δ(q3, b, I) = (q3, 4I)
δ(q3, abbr, I) = (q3, 5I)
δ(q3, strong, I) = (q3, 6I)
δ(q3, small, I) = (q3, 7I)
δ(q3, h4, N) = (q3, iN)
δ(q3, h4, U) = (q3, iU)
δ(q3, h5, C) = (q3, jC)
δ(q3, id, j) = (q3, 1j)
δ(q3, class, j) = (q3, 1j)
δ(q3, style, j) = (q3, 1j)
δ(q3, >, j) = (q3, J)
δ(q3, TEXT, J) = (q3, J)
δ(q3, /h5, J) = (q3, e)
δ(q3, em, J) = (q3, 3J)
δ(q3, b, J) = (q3, 4J)
δ(q3, abbr, J) = (q3, 5J)
δ(q3, strong, J) = (q3, 6J)
δ(q3, small, J) = (q3, 7J)
δ(q3, h5, N) = (q3, jN)
δ(q3, h5, U) = (q3, jU)
δ(q3, h6, C) = (q3, kC)
δ(q3, id, k) = (q3, 1k)
δ(q3, class, k) = (q3, 1k)
δ(q3, style, k) = (q3, 1k)
δ(q3, >, k) = (q3, K)
δ(q3, TEXT, K) = (q3, K)
δ(q3, /h6, K) = (q3, e)
δ(q3, em, K) = (q3, 3K)
δ(q3, b, K) = (q3, 4K)
δ(q3, abbr, K) = (q3, 5K)
δ(q3, strong, K) = (q3, 6K)
δ(q3, small, K) = (q3, 7K)
δ(q3, h6, N) = (q3, kN)
δ(q3, h6, U) = (q3, kU)
δ(q3, p, C) = (q3, lC)
δ(q3, id, l) = (q3, 1l)
δ(q3, class, l) = (q3, 1l)
δ(q3, style, l) = (q3, 1l)
δ(q3, >, l) = (q3, L)

δ(q3, TEXT, L) = (q3, L)
δ(q3, /p, L) = (q3, e)
δ(q3, em, L) = (q3, 3L)
δ(q3, b, L) = (q3, 4L)
δ(q3, abbr, L) = (q3, 5L)
δ(q3, strong, L) = (q3, 6L)
δ(q3, small, L) = (q3, 7L)
δ(q3, p, N) = (q3, lN)
δ(q3, p, U) = (q3, lU)
δ(q3, a, C) = (q3, mC)
δ(q3, id, m) = (q3, 1m)
δ(q3, class, m) = (q3, 1m)
δ(q3, style, m) = (q3, 1m)
δ(q3, href, m) = (q3, 1m)
δ(q3, >, m) = (q3, M)
δ(q3, TEXT, M) = (q3, M)
δ(q3, /a, M) = (q3, e)
δ(q3, em, M) = (q3, 3M)
δ(q3, b, M) = (q3, 4M)
δ(q3, abbr, M) = (q3, 5M)
δ(q3, strong, M) = (q3, 6M)
δ(q3, small, M) = (q3, 7M)
δ(q3, a, N) = (q3, mN)
δ(q3, a, U) = (q3, mU)
δ(q3, img, C) = (qimg, @C)
δ(qimg, id, @) = (qimg, 1@)
δ(qimg, =, 1) = (qimg, 2)
δ(qimg, TEXT, 2) = (qimg, e)
δ(qimg, class, @) = (qimg, 1@)
δ(qimg, style, @) = (qimg, 1@)
δ(qimg, alt, @) = (qimg, 1@)
δ(qimg, src, @) = (q3, 1@)
δ(q3, id, @) = (q3, 1@)
δ(q3, class, @) = (q3, 1@)
δ(q3, style, @) = (q3, 1@)
δ(q3, alt, @) = (q3, 1@)
δ(q3, >, @) = (q3, e)
δ(q3, img, N) = (q3, @N)
δ(q3, img, U) = (q3, @U)
δ(q3, div, C) = (q3, nC)
δ(q3, id, n) = (q3, 1n)
δ(q3, class, n) = (q3, 1n)
δ(q3, style, n) = (q3, 1n)
δ(q3, >, n) = (q3, N)
δ(q3, TEXT, N) = (q3, N)
δ(q3, /div, N) = (q3, e)
δ(q3, em, N) = (q3, 3N)
δ(q3, b, N) = (q3, 4N)

δ(q3, abbr, N) = (q3, 5N)
δ(q3, strong, N) = (q3, 6N)
δ(q3, small, N) = (q3, 7N)
δ(q3, div, N) = (q3, nN)
δ(q3, div, U) = (q3, nU)
δ(q3, br, C) = (q3, yC)
δ(q3, id, y) = (q3, 1y)
δ(q3, class, y) = (q3, 1y)
δ(q3, style, y) = (q3, 1y)
δ(q3, >, y) = (q3, e)
δ(q3, br, N) = (q3, yN)
δ(q3, br, U) = (q3, yU)
δ(q3, hr, C) = (q3, !C)
δ(q3, id, !) = (q3, 1!)
δ(q3, class, !) = (q3, 1!)
δ(q3, style, !) = (q3, 1!)
δ(q3, >, !) = (q3, e)
δ(q3, hr, N) = (q3, !N)
δ(q3, hr, U) = (q3, !U)
δ(q3, input, C) = (q3, *)
δ(q3, id, *) = (q3, 1*)
δ(q3, class, *) = (q3, 1*)
δ(q3, style, *) = (q3, 1*)
δ(q3, type, *) = (q3, 1*)
δ(q3, "text", 2) = (q3, e)
δ(q3, "password", 2) = (q3, e)
δ(q3, "email", 2) = (q3, e)
δ(q3, "number", 2) = (q3, e)
δ(q3, "checkbox", 2) = (q3, e)
δ(q3, >, *) = (q3, e)
δ(q3, input, N) = (q3, *N)
δ(q3, input, U) = (q3, *U)
δ(q3, form, C) = (q3, uC)
δ(q3, id, u) = (q3, 1u)
δ(q3, class, u) = (q3, 1u)
δ(q3, style, u) = (q3, 1u)
δ(q3, action, u) = (q3, 1u)
δ(q3, method, u) = (q3, 1u)
δ(q3, "POST", 2) = (q3, e)
δ(q3, "GET", 2) = (q3, e)
δ(q3, >, u) = (q3, U)
δ(q3, /form, U) = (q3, e)
δ(q3, TEXT, U) = (q3, U)
δ(q3, em, U) = (q3, 3U)
δ(q3, b, U) = (q3, 4U)
δ(q3, abbr, U) = (q3, 5U)
δ(q3, strong, U) = (q3, 6U)
δ(q3, small, U) = (q3, 7U)

δ(q3, form, N) = (q3, sN)
δ(q3, table, C) = (q3, o)
δ(q3, id, o) = (q3, 1u)
δ(q3, class, o) = (q3, 1o)
δ(q3, style, o) = (q3, 1o)
δ(q3, >, o) = (q3, O)
δ(q3, /table, O) = (q3, e)
δ(q3, tr, U) = (q3, p)
δ(q3, id, p) = (q3, 1p)
δ(q3, class, p) = (q3, 1p)
δ(q3, style, p) = (q3, 1p)
δ(q3, >, p) = (q3, P)
δ(q3, th, U) = (q3, r)
δ(q3, id, r) = (q3, 1r)
δ(q3, class, r) = (q3, 1r)
δ(q3, style, r) = (q3, 1r)
δ(q3, >, r) = (q3, R)
δ(q3, em, R) = (q3, 3R)
δ(q3, b, R) = (q3, 4R)
δ(q3, abbr, R) = (q3, 5R)
δ(q3, strong, R) = (q3, 6R)
δ(q3, small, R) = (q3, 7R)
δ(q3, td, P) = (q3, q)
δ(q3, id, q) = (q3, 1q)
δ(q3, class, q) = (q3, 1q)
δ(q3, style, q) = (q3, 1q)
δ(q3, >, q) = (q3, Q)
δ(q3, em, Q) = (q3, 3Q)
δ(q3, b, Q) = (q3, 4Q)
δ(q3, abbr, Q) = (q3, 5Q)
δ(q3, strong, Q) = (q3, 6Q)
δ(q3, small, Q) = (q3, 7Q)
δ(q3, id, 3) = (q3, 13)
δ(q3, class, 3) = (q3, 13)
δ(q3, style, 3) = (q3, 13)
δ(q3, >, 3) = (q3, [)
δ(q3, TEXT, [) = (q3, [)
δ(q3, button, C) = (q3, sC)
δ(q3, id, s) = (q3, 1s)
δ(q3, class, s) = (q3, 1s)
δ(q3, style, s) = (q3, 1s)
δ(q3, action, s) = (q3, 1s)
δ(q3, type, s) = (q3, 1s)
δ(q3, "submit", 2) = (q3, e)
δ(q3, "reset", 2) = (q3, e)
δ(q3, "button", 2) = (q3, e)
δ(q3, >, s) = (q3, S)
δ(q3, /button, S) = (q3, e)

δ(q3, TEXT, S) = (q3, S)
δ(q3, em, S) = (q3, 3S)
δ(q3, b, S) = (q3, 4S)
δ(q3, abbr, S) = (q3, 5S)
δ(q3, strong, S) = (q3, 6S)
δ(q3, small, S) = (q3, 7S)
δ(q3, button, N) = (q3, sN)
δ(q3, button, U) = (q3, sU)
δ(q3, id, 4) = (q3, 14)
δ(q3, class, 4) = (q3, 14)
δ(q3, style, 4) = (q3, 14)
δ(q3, >, 4) = (q3,])
δ(q3, TEXT,]) = (q3,])
δ(q3, id, 5) = (q3, 15)
δ(q3, class, 5) = (q3, 15)
δ(q3, style, 5) = (q3, 15)
δ(q3, >, ?) = (q3, {)
δ(q3, TEXT, {) = (q3, {)
δ(q3, id, 6) = (q3, 16)
δ(q3, class, 6) = (q3, 16)
δ(q3, style, 6) = (q3, 16)
δ(q3, >, 6) = (q3, })
δ(q3, TEXT, }) = (q3, })
δ(q3, id, 7) = (q3, 17)
δ(q3, class, 7) = (q3, 17)
δ(q3, style, 7) = (q3, 17)
δ(q3, >, 7) = (q3, ?)
δ(q3, TEXT, ?) = (q3, ?)
δ(q1, CMT, A) = (q1, A)
δ(q2, CMT, B) = (q2, B)
δ(q3, CMT, C) = (q3, C)
δ(q3, CMT, D) = (q3, D)
δ(q3, CMT, F) = (q3, F)
δ(q3, CMT, G) = (q3, G)
δ(q3, CMT, H) = (q3, H)
δ(q3, CMT, I) = (q3, I)
δ(q3, CMT, J) = (q3, J)
δ(q3, CMT, K) = (q3, K)
δ(q3, CMT, L) = (q3, L)
δ(q3, CMT, M) = (q3, M)
δ(q3, CMT, N) = (q3, N)
δ(q3, CMT, O) = (q3, O)
δ(q3, CMT, P) = (q3, P)
δ(q3, CMT, Q) = (q3, Q)
δ(q3, CMT, R) = (q3, R)
δ(q3, CMT, S) = (q3, S)
δ(q3, CMT, T) = (q3, T)
δ(q3, CMT, U) = (q3, U)

δ(q3, CMT, V) = (q3, V)
δ(q3, CMT, W) = (q3, W)
δ(q3, CMT, X) = (q3, X)
δ(q3, CMT, Y) = (q3, Y)
δ(q3, CMT, !) = (q3, !)
δ(q3, CMT, @) = (q3, @)
δ(q3, CMT, #) = (q3, #)
δ(q3, CMT, 1) = (q3, 1)
δ(q3, CMT, 2) = (q3, 2)
δ(q3, CMT, []) = (q3, [])
δ(q3, CMT,]) = (q3,])
δ(q3, CMT, {) = (q3, {)
δ(q3, CMT, }) = (q3, })
δ(q3, CMT, ?) = (q3, ?)
δ(q3, /em, []) = (q3, e)
δ(q3, /b,]) = (q3, e)
δ(q3, /abbr, {) = (q3, e)
δ(q3, /strong, }) = (q3, e)
δ(q3, /small, ?) = (q3, e)
δ(q3, /body, C) = (q3, e)
δ(q3, /html, A) = (q4, e)
δ(q4, e, Z) = (q4, e)
Start State (q_0) = q_0
Start Symbol (Z_0) = Z
Final State (F) = q4
(https://docs.google.com/spreadsheets/d/15O7XWvdYpQuFBia7w2uVTHNjthilNHSGM8zhh7tzSo4/edit?usp=sharing)
(https://www.figma.com/file/4PwujSGJVT1hRkg9xEyco1/PDA-Diagram-HTML-EYF?type=whiteboard&node-id=0%3A1&t=XUFgDX8i28xRLZXx-1)

3.2 Stack Symbol & Transition Symbol

	Stack symbol	Transition Symbol
<html	A	a
<head	B	b
<body	C	c
<title	D	d
<h1	F	f
<h2	G	g
<h3	H	h
<h4	I	i
<h5	J	j
<h6	K	k
<p	L	l

<a	M	m
<div	N	n
<table	O	o
<tr	P	p
<td	Q	q
<th	R	r
<button	S	s
<script	T	t
<form	U	u
TEXT	V	v
link	W	
script	X	
br	Y	
hr	!	
img	@	
input	#	*
id	1	
class	1	
style	1	
=	2	
em	[	3
b	]	4
abbr	{	5
strong	}	6
small	?	7

3.3 Tokenisasi

	Token
<html	html
<head	head
<body	body
<title	title
<h1	h1
<h2	h2
<h3	h3
<h4	h4
<h5	h5

<h6	h6
<p	p
<a	a
<div	div
<table	table
<tr	tr
<td	td
<th	th
<button	button
<script	script
<form	form
Any text	TEXT
<link	link
<script	script
<br	br
<hr	hr
<img	img
<input	input
id	id
class	class
style	style
=	=
<em	em
<b	b
<abbr	abbr
<strong	strong
<small	small
>	>

BAB 4
Implementasi

4.1 Program main.py

Program utama dari *HTML Checker* ini diimplementasikan pada `main.py`. Program ini menggunakan fungsi-fungsi yang telah dibuat. Program ini dibuat menggunakan paradigma pemrograman berbasis objek. Program dimulai dengan melakukan *parsing* terhadap argumen kedua (*path* dari file HTML yang ingin dicek). Rencana awal penulis adalah menghapus semua *new line* (`\n` atau `<br>`) untuk mempermudah pemeriksaan terhadap file HTML. Untuk memenuhi bonus, rencana awal tersebut akhirnya batal, dan penulis membuat matriks token untuk melacak lokasi kesalahan pada file HTML tadi. Program kemudian membaca file teks yang berisi PDA (PDA.txt). File HTML lalu di-tokenize. Kemudian token dan PDA akan dimasukkan ke program PDA yang berbentuk OOP tadi. Jika PDA berakhir di *final state*, maka bahasa pada file HTML valid, sebaliknya PDA akan tersangkut di *state* yang bukan final jika file HTML tidak valid.

4.2 Program tag2_symbol.py

Program pembantu dari main adalah tag2_symbol.py. Program ini fokus dalam mengubah HTML content menjadi token yang dibaca oleh PDA pada main.py. Program ini menggunakan modul *built-in* re python untuk mengubah HTML content menjadi token yang sudah dijelaskan pada bab 3 sub-bab 3.3. Data struktur yang digunakan adalah matriks, yakni *array of array of tokens* yang elemen setiap matriksnya merupakan token-token dari sebuah line, hal ini diimplementasikan agar kami dapat mengetahui posisi sintaks error pada sebuah file html. Ini juga membantu kita dalam men-*debug* PDA.txt yang kami telah buat.

4.3 Program miscellaneous.py

Program ini berisi fungsi-fungsi yang membantu penulis dalam pembentukan PDA. Awalnya, program ini berisi fungsi-fungsi yang digunakan pada *development* awal program tag2_symbol.py, tetapi sejalan berubahnya spesifikasi tugas besar dan bertambahnya/berubahnya spesifikasi berdasarkan qna, fungsi-fungsi di program miscellaneous mulai berkurang sehingga tersisa satu fungsi saja yakni fungsi print_with_color() untuk mengubah warna string. Kami memutuskan untuk tidak menghapus program miscellaneous sebagai pengingat perjalanan kami dalam mengembangkan program kami.

BAB 5
Eksperimen

Terdapat n eksperimen yang akan dicoba pada bagian ini, berikut hasilnya:

5.1 Accepted Files

5.1.1 acc_1.html

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

5.1.2 Output acc_1.html

```
PS D:\SMS 3\Tubes TBFO 1> py main.py PDA.txt "acc_1.html"

  EEEE Y   Y  FFFFF
  E     Y Y   F
  EEE    Y    FFFF
  E      Y    F
  EEEE   Y    F

HTML SYNTAX CHECKER


PDA tokens :
['html', '>', 'head', '>', 'title', '>', 'TEXT', '/title', '/head', 'body', '>', 'h1', '>', 'TEXT', '/h1', 'p', '>', 'TEXT', '/
p', '/body', '/html']

Processing.......
Accepted

Your HTML file:

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

5.1.3 acc_2.html

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
```

```
    <img src=”./welcome.jpeg”>
    <p>This is a <em>simple</em> webpage.</p>


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

5.1.4 Output acc_2.html

```
HTML SYNTAX CHECKER


PDA tokens :
['html', '>', 'head', '>', 'title', '>', 'TEXT', '/title', '/head', 'body', '>', 'h1', '>', 'TEXT', '/h1', 'h2', '>', 'TEXT',
/h2', 'img', 'src', '=', 'TEXT', '>', 'p', '>', 'TEXT', 'em', '>', 'TEXT', '/em', 'TEXT', '/p', 'div', 'id', '=', 'TEXT', 'cla
s', '=', 'TEXT', '>', 'TEXT', '/div', '/body', '/html']

Processing.......
Accepted

Your HTML file:

<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <img src=”./welcome.jpeg”>
    <p>This is a <em>simple</em> webpage.</p>

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

5.1.5 acc_3.txt

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <hr>
    <img src="./welcome.jpeg" 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>
```

5.1.6 Output acc_3.txt

```
Processing.......
Accepted

Your HTML file:

<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <hr>
    <img src="./welcome.jpeg" 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>
```

5.1.7 acc_4.txt

```
<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>
```

5.1.8 Output acc_4.txt

```
Processing.......
Accepted

Your HTML file:

<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>
```

5.1.9 acc_5.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>
```

5.1.10 Output acc_5.html

```
HTML SYNTAX CHECKER


PDA tokens :
['html', '>', 'head', '>', 'title', '>', 'TEXT', '/title', 'script', '>', 'TEXT', '/
TEXT', '/h1', 'a', '>', 'TEXT', '/a', 'br', '>', 'a', 'href', '=', 'TEXT', '>', 'TEX
, '/body', '/html']

Processing.......
Accepted

Your HTML file:

<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>
```

5.1.11 Andhika's TestCase (TC).html

```
<html><!--  faffaf -->
<!--  faffaf -->
<!--  faffaf --><head>
<!--  faffaf -->    <title>HAHAHA-094Andhika(laughing)</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,20
0;0,300;0,400;0,500;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,700&d
isplay=swap" rel="stylesheet">
<!--  faffaf -->    <link rel="LOL" href="../static/style.css">
<!--  faffaf -->    <script src="LOLOLOLOL"></script> <!--  faffaf --><!--
faffaf -->
<!--  faffaf --></head>

<!--  faffaf --><body>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.7.1/jszip.min.js"></scr
ipt>
    <script src="../static/resultScript.js"></script>
    <div id="loading" style="display: none">
        <img src="../static/assets/kuru2.gif" alt="Loading">
    </div>
    <div class="title">
        <h1>ALGEO LENS</h1>
    </div>
<!--  faffaf -->    <div id="title">     <!--  faffaf -->
 <!--  faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf --><!--
faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf -->
<div class        =                  "Graveyard" id =
"aslf;ksflfj"><!--  faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf
```

```
--><!--  faffaf --><!--  faffaf -->
    <h2>RIP</h2>
  <!--  faffaf --></div>
 <!--  faffaf --><h1>How To Use</h1><br>
 <!--  faffaf --><br><h2>How To Use</h2><!--  faffaf -->
 <!--  faffaf --><h3>How To Use</h3><br><!--  faffaf -->
 <!--  faffaf --><h4>How To Use</h4><br><!--  faffaf -->
 <!--  faffaf --><h5>How To Use</h5><br><!--  faffaf -->
 <!--  faffaf --><h6>How To Use</h6> <br><!--  faffaf -->
    </div>
<!--  faffaf -->    <div class="box"> <!--  faffaf --> <!--  faffaf -->
 <!--  faffaf -->       <div class="text"> <!--  faffaf --> <!--  faffaf
-->
 <!--  faffaf -->           <h2>1. Clone the repo</h2><br> <!--  faffaf
--><!--  faffaf --><h1>The Main Languages of the Web</h1>

<p>HTML is the standard markup language for creating Web pages. HTML
describes the structure of a Web page, and consists of a series of
elements. HTML elements tell the browser how to display the content.</p>
<!--  faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf --><!--
faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf --><hr><!--
faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf --><!--  faffaf --><!--
faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf --><p>CSS is a language that
describes how HTML elements are to be displayed on screen, paper, or in
other media. CSS saves a lot of work, because <!--  faffaf --><!--  faffaf
--><!--  faffaf -->it can control the layout of multiple web pages all at
once.<!--  faffaf --><!--  faffaf --><!--  faffaf --></p><!--  faffaf
--><!--  faffaf --><!--  faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf --><hr><!--  faffaf --><!--
faffaf --><!--  faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf -->
<!--  faffaf --><!--  faffaf --><!--  faffaf --><p>JavaScript is the
programming language of HTML and the Web. <!--  faffaf --><!--  faffaf
--><!--  faffaf -->JavaScript can change HTML content and attribute values.
JavaScript can cha<!--  faffaf --><!--  faffaf -->nge CSS. JavaScript can
hide and show HTML elements, and more.</p><!--  faffaf --><!--  faffaf
--><!--  faffaf -->
  <!--  faffaf -->          <p>This can be done by opening up the terminal,
change directory to the desired loc<!--  faffaf --><!--  faffaf -->ation,
and type: git clone git@github.com:riyora<!--  faffaf
-->x/Algeo02-22061.git <!--  faffaf --></p> <br><!--  faffaf -->
     <!--  faffaf -->   </div><!--  faffaf -->
<!--  faffaf -->    </div><!--  faffaf -->
 <!--  faffaf -->   <div class="box"><!--  faffaf -->
  <!--  faffaf -->      <div class="text"><!--  faffaf -->
    <input type="checkbox" class="toggle-input">
      <!--  faffaf -->     <h2>2. Navigate to the repo's source file
directory</h2><!--  faffaf -->
        <!--  faffaf -->    <p>Change directory to the cloned git repo by
typing: cd ./Algeo02-22061/src/</p><!--  faffaf -->
  <!--  faffaf -->      </div><!--  faffaf -->
  <!--  faffaf -->  </div><!--  faffaf -->
   <!--  faffaf --> <div class="box"><!--  faffaf -->
    <!--  faffaf -->    <div class="text"><!--  faffaf -->
       <!--  faffaf -->     <h2>3. Install the dependencies</h2><!--
faffaf -->
```

```
     <!--  faffaf -->       <p>Change directory to the cloned git repo and
type: pip install -r requirements.txt or python3 -m pip install -r
requirements.txt</p>
        </div><!--  faffaf -->
<!--  faffaf -->    </div>
    <div class="box"><!--  faffaf -->
        <div class="text">
            <h2>4. Run the website locally</h2><!--  faffaf -->
            <p>This is done by typing: py app.py or python3 app.py</p>
  <!--  faffaf -->      </div><!--  faffaf -->
  <!--  faffaf -->  </div><!--  faffaf -->
    <div class="box"><!--  faffaf -->
        <div class="text">
            <h2>5. <strong>the localhost website</strong> </h2>
            <p>Look at the<!--  faffaf --> terminal output, see where the
web is hosted at and Crtl + click<!--  faffaf --></p><!--  faffaf -->
            <abbr>WHO</abbr><!--  faffaf -->
            <input type="checkbox" class="toggle-input">
        </div>
    </div><!--  faffaf -->
    <div class="box">
        <div class="text">
            <h2>6.<!--  faffaf --> <b>Click the toggle to switch between
color or texture mode</b></h2>
            <p><small>Just click it, as simple as that</small><!--  faffaf
--></p>
        </div>
    </div><!--  faffaf -->
    <div class="box">
        <div class="text">
            <h2>7. <!--  faffaf --><em>Upload Image</em> section</h2>
            <p>J<!--  faffaf -->ust click <em>Choose File</em>, select
<em>one</em> image</p>
        </div>
    </div>
    <div class="box"><!--  faffaf -->
        <div class="text">
            <h2>8. <em>Upload<!--  faffaf --> Dataset</em> section</h2>
    <!--  faffaf -->        <p>Just click <em>Choose Files</em>, select one
or more images or even folders</p>
    <button type="button">Click Me!</button>
        </div>
    </div><!--  faffaf -->
    <div class="box">
        <div class="text">
            <h2>9. Run it</h2><!--  faffaf -->
            <p>click <em>Upload</em>, wait a few minutes or hopefully
seconds <em>(depending on your CPU multicore performance)</em><!--  faffaf
--></p><!--  faffaf -->
            <p><strong>pray that your potato PDA can handle this
file</strong></p>
        </div>
        <table>
          <tr>
            <td><b>Function Name</b></td>
            <td><b>Progress</b></td>
          </tr>
          <tr>
            <td>Inisialisi</td>
            <td>Done</td>
```

```
          </tr>
          <tr>
            <td>Perintah</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Pengguna: DAFTAR</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Pengguna: MASUK</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Pengguna: KELUAR</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Pengguna: TUTUP_PROGRAM</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Profil: GANTI_PROFIL</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Profil: LIHAT_PROFIL [NAMA]</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Profil: ATUR_JENIS_AKUN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Profil: UBAH_FOTO_PROFIL</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Teman: DAFTAR_TEMAN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Teman: HAPUS_TEMAN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Permintaan Pertemanan: TAMBAH_TEMAN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Permintaan Pertemanan: DAFTAR_PERMINTAAN_PERTEMANAN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Permintaan Pertemanan: SETUJUI_PERTEMANAN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Kicauan: KICAU</td>
            <td>Done</td>
```

```
          </tr>
          <tr>
            <td>Kicauan: KICAUAN</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Kicauan: SUKA_KICAUAN [IDKicau]</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Kicauan: UBAH_KICAUAN [IDKicau]</td>
            <td>Done</td>
          </tr>
          <tr>
   <!--  faffaf -->         <td>Balasan: BALAS [IDKicau] [IDBalasan]</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Balasan: BALASAN [IDKicau]</td>
            <td>Done</td>
          </tr>
          <tr><!--  faffaf -->
            <td>Balasan: HAPUS_BALASAN [IDKicau] [IDBalasan]</td>
            <td>Done</td>
   <!--  faffaf -->       </tr>
          <tr><!--  faffaf -->
            <td>Draf Kicauan: BUAT_DRAF</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Draf Kicauan: LIHAT_DRAF</td>
            <td>Done</td>
    <!--  faffaf -->     </tr>
          <tr>
            <td>Utas: UTAS [IDKicau]</td>
            <td>Done</td>
          </tr>
          <tr><!--  faffaf -->
            <td>Utas: SAMBUNG_UTAS [IDUtas] [index]</td>
            <td>Done</td>
          </tr>
          <tr>
            <td>Utas: HAPUS_UTAS [IDUtas] [index]</td>
 <!--  faffaf -->           <td>Done</td>
          </tr>
          <tr>
            <td>Utas: CETAK_UTAS [IDUtas]</td>
            <td>Done</td>
          </tr>
          <tr><!--  faffaf -->
            <td>Simpan dan Muat: SIMPAN</td>
            <td>Not Done</td>
          </tr>
          <tr>
            <td>Simpan dan Muat: MUAT</td>
            <td>Done</td>
          </tr>
      </table>
<!--  faffaf -->    </div><!--  faffaf -->
    <div class="item-image"><!--  faffaf -->
```

```
   <!--  faffaf -->   <img src="/static/image_upload/{{ single_file_name
}}" alt="Image">
   <!--  faffaf -->   <!--  faffaf -->
      <form action="/" method="POST">
 <!--  faffaf -->         <div class="side-div-top">
  <!--  faffaf -->        </div>
         <h3>Upload Image</h3><!--  faffaf -->
      </form>
<!--  faffaf -->
      <form id="uploadDatasetForm">
         <h3>Upload Dataset</h3><!--  faffaf -->
    <!--  faffaf -->      <br>
      </form>

      <form action="{{ url_for('download_pdf') }}" method="POST">
      </form><!--  faffaf -->
  </div>
</body><!--  faffaf -->
<!--  faffaf -->
<!--  faffaf --></html>

         <h3>Upload Image</h3><!--  faffaf -->
      </form>
<!--  faffaf -->
      <form id="uploadDatasetForm">
         <h3>Upload Dataset</h3><!--  faffaf -->
    <!--  faffaf -->      <br>
      </form>

      <form action="{{ url_for('download_pdf') }}" method="POST">
      </form><!--  faffaf -->
  </div>
</body><!--  faffaf -->
<!--  faffaf -->
<!--  faffaf --></html>
```

5.1.12 Output Andhika's TestCase (TC).html

5.2 Rejected Files

5.2.1 reject_1.html

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

5.2.2 output reject_1.html

```
HTML SYNTAX CHECKER


PDA tokens :
['html', '>', 'head', '>', 'title', '>', 'TEXT', '/title', '/head', 'body', '>', 'h1', '>', 'TEXT', 'h1', '>', 'p', '>', 'TEXT'
, '/p', '/body', '/html']

Processing.......
Failed at State: q3 with Input: ['h1', '>'] and Stack: FCAZ

Syntax Error
Failed at line 6 : <h1>Hello, World!<h1>

Possible transitions:
[('TEXT', 'F', 'q3', 'F'), ('/h1', 'F', 'q3', ''), ('em', 'F', 'q3', '3F'), ('b', 'F', 'q3', '4F'), ('abbr', 'F', 'q3', '5F'),
('strong', 'F', 'q3', '6F'), ('small', 'F', 'q3', '7F'), ('CMT', 'F', 'q3', 'F')]

Expected next inputs :
['TEXT', '/h1', 'em', 'b', 'abbr', 'strong', 'small', 'CMT']

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

5.2.3 reject_2.html

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

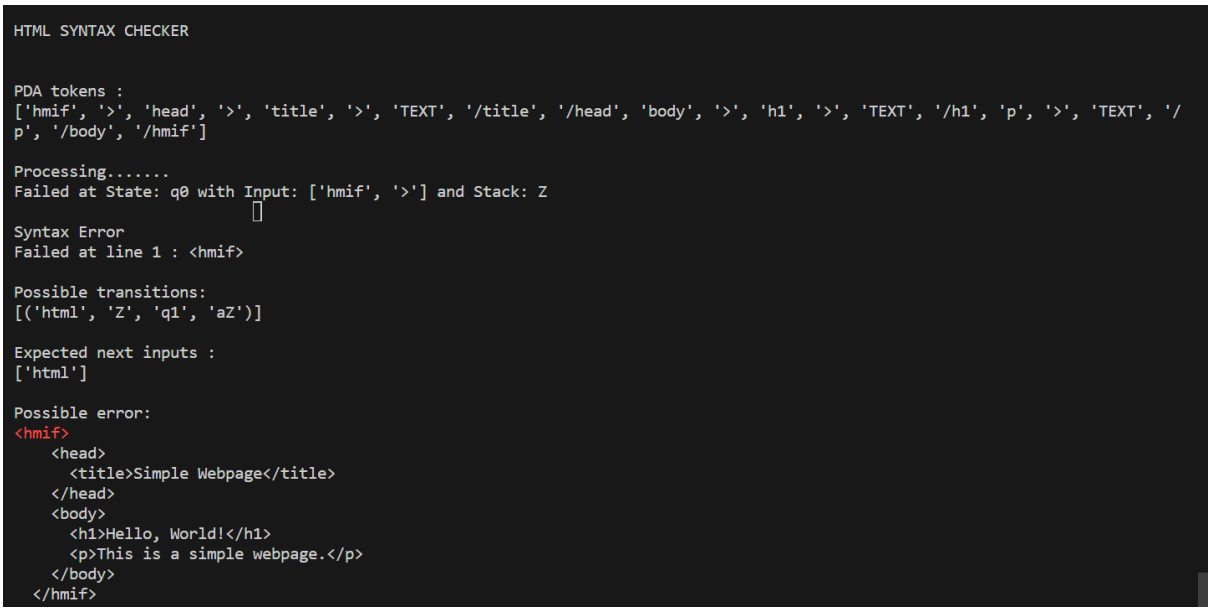
5.2.4 Output reject_2.html

```
HTML SYNTAX CHECKER


PDA tokens :
['html', '>', 'body', '>', 'h1', '>', 'TEXT', '/h1', 'p', '>', 'TEXT', '/p', '/body', 'head', '>', 'title', '>', 'TEXT', '/titl
e', '/head', '/html']

Processing.......
Failed at State: q1 with Input: ['body', '>'] and Stack: AZ

Syntax Error
Failed at line 2 : <body>

Possible transitions:
[('head', 'A', 'q2', 'bA'), ('CMT', 'A', 'q1', 'A')]

Expected next inputs :
['head', 'CMT']

Possible error:
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
  <head>
    <title>Simple Webpage</title>
  </head>
</html>
PS D:\SMS 3\Tubes TBFO 1>
```

5.2.4 reject_3.html

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

5.2.5 Output reject_3.html



5.2.6 reject_4.html

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

5.2.7 Output reject_4.html

```
PS D:\SMS 3\Tubes TBFO 1> py main.py PDA.txt "reject_4.html"

  EEEE Y   Y  FFFFF
  E     Y Y   F
  EEE    Y    FFFF
  E      Y    F
  EEEE   Y    F

HTML SYNTAX CHECKER


PDA tokens :
['html', '>', 'body', '>', 'h1', '>', 'TEXT', '/h1', 'p', '>', 'TEXT', '/p', '/body', '/html']

Processing.......
Failed at State: q1 with Input: ['body', '>'] and Stack: AZ

Syntax Error
Failed at line 2 : <body>

Possible transitions:
[('head', 'A', 'q2', 'bA'), ('CMT', 'A', 'q1', 'A')]

Expected next inputs :
['head', 'CMT']

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

5.2.8 reject_5.html

```
<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>
```

5.2.9 Output reject_5.html

```
Processing.......
Failed at State: qimg with Input: ['>'] and Stack: @CAZ

Syntax Error
Failed at line 9 : <img alt="Welcome Banner">

Possible transitions:
[('id', '@', 'qimg', '1@'), ('class', '@', 'qimg', '1@'), ('style', '@', 'qimg', '1@'), ('alt', '@', 'qimg', '1@'), ('src', '@'
, 'qimg', '2@')]

Expected next inputs :
['id', 'class', 'style', 'alt', 'src']

Possible error:
<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 -->
......
```

5.2.10 rejecet_6.html

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

</head>
<body>

<h2>HTML Forms</h2>

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

5.2.11 Output reject_6.html

```
Processing.......
Failed at State: q3 with Input: ['"TEMBAK"', '>'] and Stack: 2uCAZ

Syntax Error
Failed at line 10 : <form action="/action_page.php" method="TEMBAK">

Possible transitions:
[('TEXT', '2', 'q3', ''), ('"text"', '2', 'q3', ''), ('"password"', '2', 'q3', ''), ('"email"', '2', 'q3', ''), ('"number"', '2
', 'q3', ''), ('"checkbox"', '2', 'q3', ''), ('"POST"', '2', 'q3', ''), ('"GET"', '2', 'q3', ''), ('"submit"', '2', 'q3', ''),
('"reset"', '2', 'q3', ''), ('"button"', '2', 'q3', ''), ('CMT', '2', 'q3', '2'), ('TEXT', '2', 'q3', ''), ('TEXT', '2', 'q3',
''), ('TEXT', '2', 'q3', ''), ('TEXT', '2', 'q3', '')]

Expected next inputs :
['TEXT', '"text"', '"password"', '"email"', '"number"', '"checkbox"', '"POST"', '"GET"', '"submit"', '"reset"', '"button"', 'CM
T', 'TEXT', 'TEXT', 'TEXT', 'TEXT']

Possible error:
<html>
<head>
  <title>Simple Webpage</title>

</head>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php" method="TEMBAK">
  <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>
......

PS D:\SMS 3\Tubes TBFO 1>
```

5.2.12 reject_7.html

```
<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>
```

5.2.13 Output reject_8.html

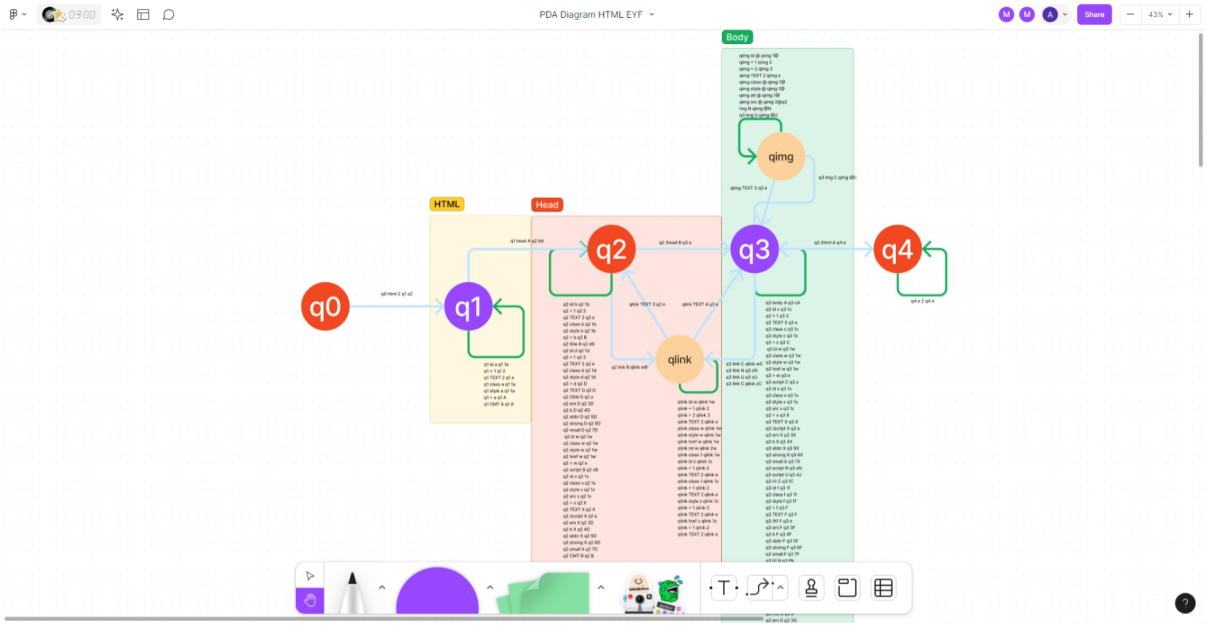
```
PDA tokens :
['html', '>', 'head', '>', 'title', '>', 'TEXT', '/title', 'script', '>', 'TEXT', '/script', '/head', 'body', '>', 'h1', '>', '
TEXT', '/h1', 'p', 'id', '=', 'TEXT', '>', '/body', '/html']

Processing.......
Failed at State: q3 with Input: ['/body'] and Stack: LCAZ

Syntax Error
Failed at line 14 : </body>

Possible transitions:
[('TEXT', 'L', 'q3', 'L'), ('/p', 'L', 'q3', ''), ('em', 'L', 'q3', '3L'), ('b', 'L', 'q3', '4L'), ('abbr', 'L', 'q3', '5L'), (
'strong', 'L', 'q3', '6L'), ('small', 'L', 'q3', '7L'), ('CMT', 'L', 'q3', 'L')]

Expected next inputs :
['TEXT', '/p', 'em', 'b', 'abbr', 'strong', 'small', 'CMT']

Possible error:
<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>
```

(p bukan void element, sehingga setelah harus ada , tidak boleh </body> . “Expected next input ‘/p’ “)

BAB 6
GitHub Repository

https://github.com/CrystalNoob/Tugas-Besar-TBFO.git

BAB 7
Diagram State



Gambar 1. *Preview Diagram State* di Figma

(diagram does not contain final version of PDA, final version can be seen in GitHub)

https://www.figma.com/file/4PwujSGJVT1hRkg9xEyco1/PDA-Diagram-HTML-EYF?type=whiteboard&node-id=0%3A1&t=XUFgDX8i28xRLZXx-1

BAB 8
Pembagian Tugas

NIM	Nama	Tugas
13522061	Maximilian Sulistiyo	Design PDA, laporan, design diagram PDA, design PDA.txt
13522075	Marvel Pangondian	Laporan, implementasi PDA, parser, dan bonus
13522094	Andhika Tantyo Anugrah	Tester (memberikan testcase yang merusak PDA kelompok sendiri dan kelompok lain sehingga memaksa mereka dan kami untuk debugging), laporan.

Tabel 1. Tabel Pembagian Tugas

Daftar Pustaka

1. Hopcroft, John E., Rajeev Motwani, Jeffrey D. Ullman. 2003. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley. Halaman 225-260.
2. https://www.w3schools.com/python/python_regex.asp
3. https://www.w3schools.com/python/python_classes.asp
4. https://www.w3schools.com/html/
5. https://www.freecodecamp.org/news/html-crash-course/
6. https://docs.google.com/spreadsheets/d/1g3IBzFkH1edkMHGrsCAsjYFQD8APkmrxhBAUQj7sXBk/edit#gid=0
7. https://html.spec.whatwg.org/multipage/
8. https://developer.mozilla.org/en-US/docs/Glossary/Void_element
9. https://docs.google.com/document/d/1W5QSSHVrXvArj3Aonw4FhbfctBK6J2YGefXpWsLW43Y/edit
10. riyorax/Algeo02-22061: Tugas Besar Algeo 2 2023-2024 (github.com)