

**LAPORAN TUGAS BESAR**  
**IF 2124 TEORI BAHASA FORMAL DAN OTOMATA**  
**HTML CHECKER**



**Disusun Oleh :**

Thea Josephine Halim	13522012
Debrina Veisha Rashika W.	13522025
Melati Anggraini	13522035

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

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>1</b>
<b>BAB I TEORI DASAR</b>	<b>2</b>
1.1 Push Down Automata	2
1.2 HTML	3
<b>BAB II APLIKASI PUSHDOWN AUTOMATA PADA SINTAKS HTML</b>	<b>5</b>
2.1 Push Down Automata pada Sintaks HTML	5
2.2. Contoh Penerapan	5
<b>BAB III HASIL PUSHDOWN AUTOMATA</b>	<b>7</b>
<b>BAB IV IMPLEMENTASI PROGRAM</b>	<b>29</b>
4.1. Regex.py	29
4.2. Openfile.py	29
4.3. Error.py	29
4.4. Main.py	30
<b>BAB V PENGUJIAN</b>	<b>31</b>
5.1. Test Case 1	31
5.2. Test Case 2	32
5.3. Test Case 3	33
5.4. Test Case 4	34
5.5. Test Case 5	35
5.6 Test Case 6	36
5.7. Test Case 7	37
5.8. Test Case 8	38
<b>BAB VI SIMPULAN DAN SARAN</b>	<b>40</b>
6.1 Simpulan	40
6.2 Saran	40
<b>LAMPIRAN</b>	<b>41</b>
A. Pembagian Tugas	41
B. Link Diagram State	41
C. Link Repository Github	41

## BAB I

### TEORI DASAR

#### 1.1 Push Down Automata

Push Down Automata adalah finite automata dengan stack untuk mengenali Context Free Languages (CFG). PDA terbentuk dari 7 tuple:

$$P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$$

Keterangan:

$Q$  = finite set of states

$\Sigma$  = set of input symbols

$\Gamma$  = set of pushdown symbols (untuk push dan pop di stack)

$\delta$  = fungsi transisi

$q_0$  = state awal

$z_0$  = stack awal, base stack

$F$  = Set of final states

Sedangkan CFG terdiri dari 4 tuples:

$$G = (V, T, P, S)$$

Keterangan:

$V$  = set of nonterminal symbols

$T$  = set of terminal symbols

$P$  = Set of production rules

$S$  = start symbol

PDA menggunakan stack untuk menyimpan informasi inputan sementara. Pada penerapannya, sebuah PDA akan membaca simbol inputan dan akan memasukkannya ke dalam stack (push) atau melakukan popping simbol teratas stack. Proses akan terjadi berkali-kali tergantung pada rules PDA yang kita buat. Penerimaan suatu bahasa dalam PDA bisa berdasarkan empty state (setelah semua input dibaca akan berada di salah satu keadaan akhir) ataupun final state (setelah semua input dibaca stack akan kosong karena pop berkali-kali). Contoh PDA adalah PDA untuk mengenali string berpola  $0^n 1^n \mid n \geq 0$ . Sebuah string yang akan menerima sejumlah 0 yang sama dengan jumlah 1, dengan catatan angka 0 berada sebelum angka 1 muncul. Dari string ini akan dibuat PDA dengan stack untuk

membandingkan jumlah 0 dan 1. Pada kasus ini, setiap kita menjumpai 0, kita perlu melakukan operasi push stack. Lalu ketika kita mendapat 1, kita dapat melakukan popping stack, sehingga setiap satu simbol 1 akan berpasangan dengan satu simbol 0. Dengan melakukan popping terus-menerus, stack akan menjadi kosong jika jumlah kedua simbol sama.

## 1.2 HTML

HTML (Hypertext Markup Language) adalah bahasa untuk membuat struktur sebuah website. HTML akan menjadi kerangka dasar seluruh konten website dan memiliki berbagai komponen tag dengan format yang berbeda-beda. Umumnya, sebuah komponen tag bisa memiliki tag buka dan tag tutup (nama tag diapit tanda kurung sudut <>), tetapi pada beberapa tag tidak perlu diberikan tag tutup. Contohnya sebuah komponen tag HTML: <p> konten </p>. Semua tag dapat memiliki atribut global dan atribut khusus dalam tag itu sendiri. Atribut global ada banyak, tetapi di tugas kali ini akan dibatasi menjadi class, style, dan id saja. Atribut class seolah-olah memberikan nama tag tersebut, atau lebih tepatnya untuk menghubungkan tag tersebut dengan CSS kita. Atribut style akan langsung mengubah gaya/tampilan pada elemen tersebut dengan properti CSS. Sedangkan atribut id akan memberikan identifikasi khusus elemen tersebut.

Struktur dasar sebuah HTML terdiri dari tag awal html, diikuti oleh tag head dan tag body yang keduanya berada di dalam tag html tadi. Tag head akan diisi oleh tautan-tautan ke file eksternal, title website, dan elemen lain yang mendefinisikan page website tersebut. Sedangkan tag body akan berisikan konten utama webpage, dapat berupa link, gambar, formulir, tombol, dan lainnya. Struktur dasar ini wajib ada, jika tidak HTML akan langsung dianggap salah. Berikut adalah komponen-komponen tag yang akan dibuat menjadi rules PDA pada tugas besar ini.

Tag	Attribute Tambahan	Void Element	Catatan Tambahan
html			Tag html wajib ada, Dokumen harus diawali tag html
head			Tag head wajib ada, Tag head harus berada di dalam tag html dan diatas tag body
body			Tag body wajib ada, Tag body harus berada di dalam tag html dan dibawah tag head. Semua elemen yang disebutkan setelah ini kecuali title harus berada di dalam body, namun tidak semua tag

			wajib ada di dalam body.
title			Hanya boleh berada dalam head
link	Rel, href	V	Atribut rel wajib ada, dapat berada dalam head
script	src		dapat berada dalam head
h1,h2,h3,h4,h5, h6			
p			
br		V	
em			
b			
abbr			
strong			
small			
hr		V	
div			
a	href		
img	src, alt	V	Atribut src wajib ada
button	type		Nilai attribute type dibatasi pada submit, reset, button
form	action, method		Nilai method harus dibatasi pada GET atau POST
input	type	V	Nilai attribute type harus dibatasi pada text, password, email, number, atau checkbox
table			Bentuk tabel tidak perlu diperhatikan (tidak apa-apa jika baris 1 terdiri dari 1 kolom sedangkan baris 2 terdiri dari 2 kolom)
tr, td, th			Harus berada dalam tag table

Pada tugas besar kali ini, kami akan memanfaatkan konsep PDA untuk membuat sebuah HTML checker dalam Python, untuk mengecek apakah sebuah file yang berformat HTML akan menimbulkan error atau tidak jika di-*compile*. PDA akan membaca berbagai jenis input, mulai dari jenis tag hingga setiap atribut tag dan akan melakukan pengecekan berdasar rules PDA yang telah dibuat.

## BAB II

### APLIKASI PUSHDOWN AUTOMATA PADA SINTAKS HTML

#### 2.1 Push Down Automata pada Sintaks HTML

Implementasi Pushdown Automata (PDA) pada konteks penguraian sintaks HTML dapat digunakan untuk memeriksa setiap karakter/symbol dalam dokumen secara berurutan, misalnya ketika karakter pembuka '<html>' terdeteksi, karakter tersebut secara otomatis dimasukkan ke dalam sebuah stack. Proses ini mencatat awal dari pembuka sebuah *tag* elemen HTML. Sebaliknya, ketika karakter penutup '</html>' muncul, PDA melakukan operasi *pop* pada tumpukan, menciptakan representasi penutup elemen HTML tersebut.

Aplikasi PDA ini sangat relevan dalam penanganan kasus-kasus kompleks seperti elemen bersarang (*nested*) dan penanganan atribut. PDA memungkinkan kita untuk mempertahankan konteks dari elemen-elemen yang bersarang. Sebagai contoh, pada saat menemui elemen <div>, PDA dapat melacak pembuka dan penutup yang sesuai, bahkan ketika terdapat elemen tag lain di dalamnya.

#### 2.2. Contoh Penerapan

Contoh sederhana untuk konsep *push* dan *pop* pada PDA dalam konteks penguraian sintaks HTML sebagai berikut:

1. Misalnya pengguna ingin memeriksa kebenaran kode berikut:

**<h1> ini adalah heading </h1>**

2. Push pada Karakter '<h1>':

Ketika PDA menemui karakter '<h1', PDA secara otomatis mem-*push* karakter tersebut ke dalam tumpukan.

- Pada awalnya, tumpukan kosong: []
- Ketika karakter '<h1' terdeteksi, tumpukan menjadi: ['<h1']

3. Push pada Karakter '>':

Selanjutnya, saat PDA menemui karakter '>', PDA melakukan operasi *push* untuk menangani penutup tag elemen HTML.

- Setelah karakter '<h1' dimasukkan ke dalam tumpukan: ['<h1']
- Ketika karakter '>' terdeteksi, tumpukan menjadi: ['<h1', '>']

4. Karakter di antara tag:

Selanjutnya, saat PDA menemui karakter diantara tag akan dianggap sebagai 'e' atau diabaikan sehingga tidak dilakukan *push* atau *pop* pada *stack*.

5. Pop pada Karakter '<':

Selanjutnya, saat PDA menemui karakter '<', PDA akan melakukan operasi pop.

- Setelah karakter '>' dimasukkan ke dalam tumpukan: ['<h1', '>']
- Ketika karakter '<' terdeteksi, PDA melakukan operasi pop, sehingga tumpukan menjadi: ['<h1']

6. Pop pada Karakter '/h1>'

Selanjutnya, saat PDA menemui karakter '/h1>', PDA melakukan operasi pop untuk menangani penutupan elemen HTML.

- Setelah karakter '<' terdeteksi, tumpukan menjadi: ['<h1']
- Ketika karakter '/h1>' terdeteksi, PDA melakukan operasi pop, sehingga tumpukan menjadi kosong: []

Dengan cara ini, PDA mengenali bahwa elemen HTML telah selesai, dan tumpukan kembali kosong, menunjukkan bahwa elemen tersebut telah ditutup secara benar.

Contoh tersebut sederhana dan hanya mencakup elemen pembuka dan penutup tunggal. Namun, dalam implementasi sebenarnya, tumpukan dapat mengandung informasi lebih kompleks, seperti mengelola elemen bersarang dan menangani atribut di dalamnya. Prinsip dasar *push* dan *pop* tetap sama, di mana tag pembuka ditambahkan ke dalam tumpukan dan tag penutup mengakibatkan penghapusan karakter terakhir dari tumpukan, dan juga tag pembuka dan penutup harus sama untuk bisa dilakukan operasi *pop*.

### BAB III

## HASIL PUSHDOWN AUTOMATA

Berikut Push Down Automata yang digunakan untuk memeriksa kebenaran pada file HTML. Kami menggunakan PDA yang menerima Final State untuk memeriksa sintaks dari file HTML. Untuk Diagram PDA terdapat pada lampiran di halaman terakhir.

```
# total states
Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Qdiv Qtemp
# input symbols
< > / html /html head/head body /body title /title h1 /h1 h2
/h2 h3 /h3 h4 /h4 h5 /h5 h6 /h6 p /p div /div table /table tr
/tr td /td th /th hr br small /small strong /strong form
/form button /button script /script b /b a /a em /em abbr
/abbr img input link class style id src href alt action rel
"get" "post" method1 method2 type1 type2 "checkbox" "number"
"text" "email" "password" "submit" "reset" "button" = - petik
petik2 < > e <!-- -->
# stack symbols
# Z I H dan R adalah tag/atribut sesuai dengan input symbol
yang
# digunakan untuk mempersingkat
< > / html /html head/head body /body title /title h1 /h1 h2
/h2 h3 /h3 h4 /h4 h5 /h5 h6 /h6 p /p div /div table /table tr
/tr td /td th /th hr br small /small strong /strong form
/form button /button script /script b /b a /a em /em abbr
/abbr img input link class style id src href alt action rel
"get" "post" method1 method2 type1 type2 "checkbox" "number"
"text" "email" "password" "submit" "reset" "button" = - petik
petik2 < > e <!-- -->
# starting state
Q0
# starting stack
Z
# accepting state
F
# list of productions
Q0 html Z Q0 HZ
Q0 > html Q1 >H
Q1 < > Q1 e
Q1 /html html F e
Q1 head > Q1 H>
Q1 > head Q2 >H
Q2 < > Q2 e
Q2 /head head Q2 e
Q1 id head Q1 IH
Q1 = id Q2 e
Q2 petik2 head Q2 "H
```



```

Q2 e petik2 Q2 "
Q2 petik2 petik2 Q1 e
Q2 petik head Q2 "H
Q2 e petik Q2 "
Q2 petik petik Q1 e
Q1 style head Q1 IH
Q1 = style Q2 e
Q1 class head Q1 IH
Q1 = class Q2 e
Q2 body > Q2 H>
Q2 > body Q3 >H
Q3 < > Q3 e
Q3 /body body Q1 e
Q2 id body Q2 IH
Q2 = id Q3 e
Q3 petik2 body Q3 "H
Q3 e petik2 Q3 "
Q3 petik2 petik2 Q2 e
Q3 petik body Q3 "H
Q3 e petik Q3 "
Q3 petik petik Q2 e
Q2 style body Q2 IH
Q2 = style Q3 e
Q2 class body Q2 IH
Q2 = class Q3 e
Q2 title > Q2 H>
Q2 > title Q3 >H
Q3 e > Q3 >
Q3 = > Q3 "
Q3 "GET" > Q3 "
Q3 'GET' > Q3 "
Q3 "POST" > Q3 "
Q3 'POST' > Q3 "
Q3 "get" > Q3 "
Q3 'get' > Q3 "
Q3 "checkbox" > Q3 "
Q3 'checkbox' > Q3 "
Q3 "number" > Q3 "
Q3 'number' > Q3 "
Q3 "text" > Q3 "
Q3 'text' > Q3 "
Q3 "email" > Q3 "
Q3 'email' > Q3 "
Q3 "password" > Q3 "
Q3 'password' > Q3 "
Q3 "submit" > Q3 "
Q3 'submit' > Q3 "
Q3 "reset" > Q3 "
Q3 'reset' > Q3 "
Q3 "button" > Q3 "

```

```

Q3 'button' > Q3 "
Q3 /title title Q2 e
Q2 id title Q2 IH
Q2 = id Q3 e
Q3 petik2 title Q3 "H
Q3 e petik2 Q3 "
Q3 petik2 petik2 Q2 e
Q3 petik title Q3 "H
Q3 e petik Q3 "
Q3 petik petik Q2 e
Q2 style title Q2 IH
Q2 = style Q3 e
Q2 class title Q2 IH
Q2 = class Q3 e
Q3 h1 > Q3 H>
Q3 > h1 Q4 >H
Q4 /h1 h1 Q3 e
Q3 id h1 Q3 IH
Q3 = id Q4 e
Q4 petik2 h1 Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik h1 Q4 "H
Q4 e petik Q4 "
Q4 = > Q4 "
Q4 "GET" > Q4 "
Q4 'GET' > Q4 "
Q4 "POST" > Q4 "
Q4 'POST' > Q4 "
Q4 "get" > Q4 "
Q4 'get' > Q4 "
Q4 "checkbox" > Q4 "
Q4 'checkbox' > Q4 "
Q4 "number" > Q4 "
Q4 'number' > Q4 "
Q4 "text" > Q4 "
Q4 'text' > Q4 "
Q4 "email" > Q4 "
Q4 'email' > Q4 "
Q4 "password" > Q4 "
Q4 'password' > Q4 "
Q4 "submit" > Q4 "
Q4 'submit' > Q4 "
Q4 "reset" > Q4 "
Q4 'reset' > Q4 "
Q4 "button" > Q4 "
Q4 'button' > Q4 "
Q4 petik petik Q3 e
Q3 style h1 Q3 IH
Q3 = style Q4 e

```

```

Q3 class h1 Q3 IH
Q3 = class Q4 e
Q3 h2 > Q3 H>
Q3 > h2 Q4 >H
Q4 /h2 h2 Q3 e
Q3 id h2 Q3 IH
Q3 = id Q4 e
Q4 petik2 h2 Q4 "H
Q4 petik2 petik2 Q3 e
Q4 petik h2 Q4 "H
Q4 petik petik Q3 e
Q3 style h2 Q3 IH
Q3 class h2 Q3 IH
Q3 h3 > Q3 H>
Q3 > h3 Q4 >H
Q4 /h3 h3 Q3 e
Q3 id h3 Q3 IH
Q3 = id Q4 e
Q4 petik2 h3 Q4 "H
Q4 petik2 petik2 Q3 e
Q4 petik h3 Q4 "H
Q4 petik petik Q3 e
Q3 style h3 Q3 IH
Q3 class h3 Q3 IH
Q3 h4 > Q3 H>
Q3 > h4 Q4 >H
Q4 < > Q4 e
Q4 e > Q4 >
Q4 /h4 h4 Q3 e
Q3 id h4 Q3 IH
Q3 = id Q4 e
Q4 petik2 h4 Q4 "H
Q4 petik2 petik2 Q3 e
Q4 petik h4 Q4 "H
Q4 petik petik Q3 e
Q3 style h4 Q3 IH
Q3 class h4 Q3 IH
Q3 h5 > Q3 H>
Q3 > h5 Q4 >H
Q4 /h5 h5 Q3 e
Q3 id h5 Q3 IH
Q3 = id Q4 e
Q4 petik2 h5 Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik h5 Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style h5 Q3 IH
Q3 class h5 Q3 IH

```

```

Q3 h6 > Q3 H>
Q3 > h6 Q4 >H
Q4 /h6 h6 Q3 e
Q3 id h6 Q3 IH
Q3 = id Q4 e
Q4 petik2 h6 Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik h6 Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style h6 Q3 IH
Q3 class h6 Q3 IH
Q3 a > Q3 H>
Q3 > a Q4 >H
Q4 /a a Q3 e
Q3 id a Q3 IH
Q3 = id Q4 e
Q4 petik2 a Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik a Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style a Q3 IH
Q3 class a Q3 IH
Q3 href a Q3 IH
Q3 = href Q4 e
Q3 button > Q3 H>
Q3 > button Q4 >H
Q4 /button button Q3 e
Q3 id button Q3 IH
Q3 = id Q4 e
Q4 petik2 button Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik button Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style button Q3 IH
Q3 class button Q3 IH
Q4 "submit" type1 Q3 e
Q4 'submit' type1 Q3 e
Q4 "reset" type1 Q3 e
Q4 'reset' type1 Q3 e
Q4 "button" type1 Q3 e
Q4 'button' type1 Q3 e
Q4 "submit" type2 Q3 e
Q4 'submit' type2 Q3 e
Q4 "reset" type2 Q3 e

```

```

Q4 'reset' type2 Q3 e
Q4 "button" type2 Q3 e
Q4 'button' type2 Q3 e
Q3 type1 button Q4 IH
Q3 type2 button Q4 IH
Q3 p > Q3 H>
Q3 > p Q4 >H
Q4 /p p Q3 e
Q3 id p Q3 IH
Q3 = id Q4 e
Q4 petik2 p Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik p Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style p Q3 IH
Q3 class p Q3 IH
Q3 div > Q3 H>
Q3 > div Q3 >H
Q3 e > Q3 >
Q3 /div div Q3 e
Q3 id div Q3 IH
Q3 = id Q4 e
Q4 petik2 div Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik div Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style div Q3 IH
Q3 class div Q3 IH
Q4 em > Q4 H>
Q4 > em Q5 >H
Q5 < > Q5 e
Q5 "GET" > Q5 "
Q5 'GET' > Q5 "
Q5 "POST" > Q5 "
Q5 'POST' > Q5 "
Q5 "get" > Q5 "
Q5 'get' > Q5 "
Q5 "checkbox" > Q5 "
Q5 'checkbox' > Q5 "
Q5 "number" > Q5 "
Q5 'number' > Q5 "
Q5 "text" > Q5 "
Q5 'text' > Q5 "
Q5 "email" > Q5 "
Q5 'email' > Q5 "
Q5 "password" > Q5 "

```

```

Q5 'password' > Q5 "
Q5 "submit" > Q5 "
Q5 'submit' > Q5 "
Q5 "reset" > Q5 "
Q5 'reset' > Q5 "
Q5 "button" > Q5 "
Q5 'button' > Q5 "
Q5 = > Q5 "
Q5 e > Q5 >
Q5 /em em Q4 e
Q4 b > Q4 H>
Q4 > b Q5 >H
Q5 < > Q5 e
Q5 e > Q5 >
Q5 /b b Q4 e
Q4 abbr > Q4 H>
Q4 > abbr Q5 >H
Q5 < > Q5 e
Q5 e > Q5 >
Q5 /abbr abbr Q4 e
Q4 strong > Q4 H>
Q4 > strong Q5 >H
Q5 < > Q5 e
Q5 e > Q5 >
Q5 /strong strong Q4 e
Q4 small > Q4 H>
Q4 > small Q5 >H
Q5 < > Q5 e
Q5 e > Q5 >
Q5 /small small Q4 e
Q3 br > Q3 H>
Q3 > br Q3 e
Q3 hr > Q3 H>
Q3 > hr Q3 e
Q3 table > Q3 H>
Q3 > table Q12 >H
Q12 /table table Q3 e
Q12 < > Q12 e
Q3 id table Q3 IH
Q3 = id Q4 e
Q4 petik2 table Q4 "H
Q4 e petik2 Q4 "
Q4 petik2 petik2 Q3 e
Q4 petik table Q4 "H
Q4 e petik Q4 "
Q4 petik petik Q3 e
Q3 style table Q3 IH
Q3 = style Q4 e
Q3 class table Q3 IH
Q3 = class Q4 e

```

```

Q12 tr > Q12 H>
Q12 > tr Q5 >H
Q5 < > Q5 e
Q5 /tr tr Q12 e
Q12 id tr Q12 IH
Q12 = id Q5 e
Q5 petik2 tr Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik tr Q5 "H
Q5 e petik Q5 "
Q5 petik petik Q12 e
Q12 style tr Q12 IH
Q12 = style Q5 e
Q12 class tr Q12 IH
Q12 = class Q5 e
Q5 td > Q5 H>
Q5 > td Q6 >H
Q6 < > Q6 e
Q6 e > Q6 >
Q6 "GET" > Q6 "
Q6 'GET' > Q6 "
Q6 "POST" > Q6 "
Q6 'POST' > Q6 "
Q6 "get" > Q6 "
Q6 'get' > Q6 "
Q6 "checkbox" > Q6 "
Q6 'checkbox' > Q6 "
Q6 "number" > Q6 "
Q6 'number' > Q6 "
Q6 "text" > Q6 "
Q6 'text' > Q6 "
Q6 "email" > Q6 "
Q6 'email' > Q6 "
Q6 "password" > Q6 "
Q6 'password' > Q6 "
Q6 "submit" > Q6 "
Q6 'submit' > Q6 "
Q6 "reset" > Q6 "
Q6 'reset' > Q6 "
Q6 "button" > Q6 "
Q6 'button' > Q6 "
Q6 = > Q6 "
Q6 /td td Q5 e
Q5 id tr Q5 IH
Q5 = id Q6 e
Q6 petik2 tr Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e
Q6 petik tr Q6 "H

```

```

Q6 e petik Q6 "
Q6 petik petik Q5 e
Q5 style tr Q5 IH
Q5 = style Q6 e
Q5 class tr Q5 IH
Q5 = class Q6 e
Q5 th > Q5 H>
Q5 > th Q6 >H
Q6 e > Q6 >
Q6 /th th Q5 e
Q5 id th Q5 IH
Q5 = id Q6 e
Q6 petik2 th Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e
Q6 petik th Q6 "H
Q6 e petik Q6 "
Q5 style th Q5 IH
Q5 = style Q6 e
Q5 class th Q5 IH
Q5 = class Q6 e
Q3 form > Q3 H>
Q3 action form Q12 IH
Q12 = action Q12 e
Q12 petik2 form Q12 "H
Q12 e petik2 Q12 "
Q12 petik2 petik2 Q3 e
Q12 petik petik Q3 e
Q3 > form Q3 >H
Q3 /form form Q3 e
Q3 method form Q4 IH
Q4 = method Q4 e
Q4 > form Q3 >H
Q4 petik form Q4 'H
Q4 e petik Q4 '
Q4 petik petik Q3
Q3 method1 form Q4 IH
Q3 method2 form Q4 IH
Q4 "GET" method2 Q3 e
Q4 'GET' method2 Q3 e
Q4 "POST" method2 Q3 e
Q4 'POST' method2 Q3 e
Q4 "GET" method1 Q3 e
Q4 'GET' method1 Q3 e
Q4 "POST" method1 Q3 e
Q4 'POST' method1 Q3 e
Q3 > form Q3 >H
Q3 id form Q3 IH
Q3 = id Q4 e
Q3 class form Q3 IH

```



```

Q3 = class Q4 e
Q3 style form Q3 IH
Q3 = style Q4 e
Q2 script > Qtemp H>
Qtemp src script Qtemp IH
Qtemp = src Qtemp e
Qtemp petik2 script Qtemp "H
Qtemp e petik2 Qtemp "
Qtemp petik2 petik2 Qtemp e
Qtemp petik script Qtemp 'H
Qtemp e petik Qtemp '
Qtemp petik petik Qtemp e
Qtemp > script Qtemp >H
Qtemp < > Qtemp e
Qtemp e > Qtemp >
Qtemp /script script Q2 e
Qtemp petik2 > Qtemp >
Qtemp petik > Qtemp >
Qtemp = > Qtemp >
Qtemp id script Qtemp IH
Qtemp = id Qtemp e
Qtemp class script Qtemp IH
Qtemp = class Qtemp e
Qtemp style script Qtemp IH
Qtemp = style Qtemp e
Q3 script > Qdiv H>
Qdiv src script Qdiv IH
Qdiv = src Qdiv e
Qdiv petik2 script Qdiv "H
Qdiv e petik2 Qdiv "
Qdiv petik2 petik2 Qdiv e
Qdiv petik script Qdiv 'H
Qdiv e petik Qdiv '
Qdiv petik petik Qdiv e
Qdiv > script Qdiv >H
Q3 script > Q4 H>
Q4 src script Q4 IH
Q4 = src Q4 e
Q4 petik2 script Q4 "H
Q4 petik2 petik2 Q4 e
Q4 > script Qdiv >H
Qdiv petik2 > Qdiv >
Qdiv = > Qdiv >
Qdiv < > Qdiv e
Qdiv petik2 > Qdiv >
Qdiv petik > Qdiv >
Qdiv = > Qdiv >
Qdiv < > Qdiv e
Qdiv e > Qdiv >
Qdiv /script script Q3 e

```

```

Qdiv id script Qdiv IH
Qdiv = id Qdiv e
Qdiv class script Qdiv IH
Qdiv = class Qdiv e
Qdiv style script Qdiv IH
Qdiv = style Qdiv e
Q3 input > Q3 H>
Q3 type1 input Q4 IH
Q3 type2 input Q4 IH
Q4 "text" type1 Q3 e
Q4 'text' type1 Q3 e
Q4 "email" type1 Q3 e
Q4 'email' type1 Q3 e
Q4 "password" type1 Q3 e
Q4 'password' type1 Q3 e
Q4 "checkbox" type1 Q3 e
Q4 'checkbox' type1 Q3 e
Q4 "number" type1 Q3 e
Q4 'number' type1 Q3 e
Q4 "text" type2 Q3 e
Q4 'text' type2 Q3 e
Q4 "email" type2 Q3 e
Q4 'email' type2 Q3 e
Q4 "password" type2 Q3 e
Q4 'password' type2 Q3 e
Q4 "checkbox" type2 Q3 e
Q4 'checkbox' type2 Q3 e
Q4 "number" type2 Q3 e
Q4 'number' type2 Q3 e
Q3 > input Q3 e
Q3 id input Q3 IH
Q3 = id Q4 e
Q4 petik input Q4 'H
Q4 e petik Q4 '
Q4 petik2 input Q4 "H
Q4 e petik2 Q4 "
Q3 class input Q3 IH
Q3 = class Q4 e
Q3 style input Q3 IH
Q3 = style Q4 e
Q6 href link Q6 RL
Q6 = href Q6 e
Q6 petik2 link Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik link Q7 "<
Q7 e petik Q7 "
Q7 petik petik Q6 e
Q8 href link Q8 RL
Q8 = href Q8 e

```

```

Q8 petik2 link Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik link Q9 "<
Q9 e petik Q9 "
Q6 id link Q6 RL
Q6 = id Q6 e
Q6 petik2 link Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik link Q6 "<
Q7 e petik Q7 "
Q7 petik petik Q7 e
Q8 id link Q8 RL
Q8 = id Q8 e
Q8 petik2 link Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik link Q9 "<
Q9 e petik Q9 "
Q6 style link Q6 RL
Q6 = style Q6 e
Q6 petik2 link Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik link Q7 "<
Q7 e petik Q7 "
Q7 petik petik Q6 e
Q8 style link Q8 RL
Q8 = style Q8 e
Q8 petik2 link Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik link Q9 "<
Q9 e petik Q9 "
Q6 class link Q4 RL
Q6 = class Q6 e
Q6 petik2 link Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik link Q7 "<
Q7 e petik Q7 "
Q7 petik petik Q6 e
Q8 class link Q8 RL
Q8 = class Q8 e
Q8 petik2 link Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik link Q9 "<
Q9 e petik Q9 "

```

```

Q9 petik2 petik2 Q8 e
Q3 link > Q6 L>
Q6 rel link Q6 RL
Q6 = rel Q7 e
Q7 petik2 link Q8 "H
Q8 e petik2 Q8 "
Q8 petik2 petik2 Q8 e
Q7 petik link Q8 "<
Q8 e petik Q8 "
Q8 petik petik Q8 e
Q8 > link Q3 e
Q9 href link Q9 RL
Q9 = href Q9 e
Q9 petik2 link Q10 "<
Q10 e petik2 Q10 "
Q9 petik2 petik2 Q9 e
Q9 petik link Q10 "<
Q10 e petik Q10 "
Q10 petik petik Q9 e
Q10 href link Q10 RL
Q10 = href Q10 e
Q10 petik2 link Q11 "<
Q11 e petik2 Q11 "
Q11 petik2 petik2 Q10 e
Q10 petik link Q11 "<
Q11 e petik Q11 "
Q11 petik petik Q10 e
Q9 id link Q9 RL
Q9 = id Q9 e
Q9 petik2 link Q10 "<
Q10 e petik2 Q10 "
Q10 petik2 petik2 Q9 e
Q9 petik link Q10 "<
Q10 e petik Q10 "
Q10 petik petik Q9 e
Q10 id link Q10 RL
Q10 = id Q10 e
Q10 petik2 link Q11 "<
Q11 e petik2 Q11 "
Q11 petik2 petik2 Q10 e
Q10 petik link Q11 "<
Q11 e petik Q10 "
Q9 style link Q9 RL
Q9 = style Q9 e
Q9 petik2 link Q10 "<
Q10 e petik2 Q9 "
Q10 petik2 petik2 Q9 e
Q9 petik link Q10 "<
Q10 e petik Q10 "
Q10 petik petik Q9 e

```

```

Q10 style link Q10 RL
Q10 = style Q10 e
Q10 petik2 link Q11 "<
Q11 e petik2 Q11 "
Q11 petik2 petik2 Q10 e
Q10 petik link Q11 "<
Q11 e petik Q11 "
Q9 class link Q9 RL
Q9 = class Q9 e
Q9 petik2 link Q10 "<
Q10 e petik2 Q10 "
Q10 petik2 petik2 Q9 e
Q9 petik link Q10 "<
Q10 e petik Q10 "
Q10 petik petik Q9 e
Q10 class link Q10 RL
Q10 = class Q10 e
Q10 petik2 link Q11 "<
Q11 e petik2 Q11 "
Q11 petik2 petik2 Q10 e
Q10 petik link Q11 "<
Q11 e petik Q11 "
Q11 petik2 petik2 Q10 e
Q2 link > Q9 L>
Q9 rel link Q9 RL
Q9 = rel Q10 e
Q10 petik2 link Q11 "H
Q11 e petik2 Q10 "
Q10 petik2 petik2 10 e
Q9 petik link Q9 "<
Q10 e petik Q10 "
Q10 petik petik Q10 e
Q10 > link Q2 e
Q2 <!-- > Q3 L>
Q3 - <!-- Q4 -<
Q4 e - Q4 -
Q4 - - Q3 e
Q3 -> <!-- Q2 e
Q3 <!-- > Q4 L>
Q4 - <!-- Q5 -<
Q5 e - Q5 -
Q5 - - Q4 e
Q4 -> <!-- Q3 e
Q4 <!-- > Q5 L>
Q5 - <!-- Q6 -<
Q6 e - Q6 -
Q6 - - Q5 e
Q5 -> <!-- Q4 e
Q5 <!-- > Q6 L>
Q6 - <!-- Q7 -<

```

```

Q7 e - Q7 -
Q7 - - Q6 e
Q6 -> <!-- Q5 e
Q1 <!-- > Q2 L>
Q2 - <!-- Q3 -<
Q3 e - Q3 -
Q3 - - Q2 e
Q2 -> <!-- Q1 e
Q0 <!-- Z Q1 L>
Q1 - <!-- Q2 -<
Q2 e - Q2 -
Q2 - - Q1 e
Q1 -> <!-- Q0 e
Q12 <!-- > Q13 L>
Q13 - <!-- Q14 -<
Q14 e - Q14 -
Q14 - - Q13 e
Q13 -> <!-- Q12 e
Q6 alt img Q6 RL
Q6 = alt Q6 e
Q6 petik2 img Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik img Q7 "<
Q7 e petik Q7 "
Q7 petik petik Q6 e
Q8 alt img Q8 RL
Q8 = alt Q8 e
Q8 petik2 img Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik img Q9 "<
Q9 e petik Q9 "
Q6 id img Q6 RL
Q6 = id Q6 e
Q6 petik2 img Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik img Q6 "<
Q7 e petik Q7 "
Q7 petik petik Q7 e
Q8 id img Q8 RL
Q8 = id Q8 e
Q8 petik2 img Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik img Q9 "<
Q9 e petik Q9 "
Q6 style img Q6 RL
Q6 = style Q6 e

```

```

Q6 petik2 img Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik img Q7 "<
Q7 e petik Q7 "
Q7 petik petik Q6 e
Q8 style img Q8 RL
Q8 = style Q8 e
Q8 petik2 img Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik img Q9 "<
Q9 e petik Q9 "
Q6 class img Q4 RL
Q6 = class Q6 e
Q6 petik2 img Q7 "<
Q7 e petik2 Q7 "
Q7 petik2 petik2 Q6 e
Q6 petik img Q7 "<
Q7 e petik Q7 "
Q7 petik petik Q6 e
Q8 class img Q8 RL
Q8 = class Q8 e
Q8 petik2 img Q9 "<
Q9 e petik2 Q9 "
Q9 petik2 petik2 Q8 e
Q8 petik img Q9 "<
Q9 e petik Q9 "
Q9 petik2 petik2 Q8 e
Q3 img > Q6 L>
Q6 src img Q6 RL
Q6 = src Q7 e
Q7 petik2 img Q8 "H
Q8 e petik2 Q8 "
Q8 petik2 petik2 Q8 e
Q7 petik img Q8 "<
Q8 e petik Q8 "
Q8 petik petik Q8 e
Q8 > img Q3 e
Q12 h1 > Q12 H>
Q12 > h1 Q5 >H
Q5 /h1 h1 Q12 e
Q12 id h1 Q12 IH
Q12 = id Q5 e
Q5 petik2 h1 Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik h1 Q5 "H
Q5 e petik Q5 "
Q5 "GET" > Q5 "

```

```

Q5 'GET' > Q5 "
Q5 "POST" > Q5 "
Q5 'POST' > Q5 "
Q5 "get" > Q5 "
Q5 'get' > Q5 "
Q5 "checkbox" > Q5 "
Q5 'checkbox' > Q5 "
Q5 "number" > Q5 "
Q5 'number' > Q5 "
Q5 "text" > Q5 "
Q5 'text' > Q5 "
Q5 "email" > Q5 "
Q5 'email' > Q5 "
Q5 "password" > Q5 "
Q5 'password' > Q5 "
Q5 "submit" > Q5 "
Q5 'submit' > Q5 "
Q5 "reset" > Q5 "
Q5 'reset' > Q5 "
Q5 "button" > Q5 "
Q5 'button' > Q5 "
Q5 = > Q5 "
Q5 petik petik Q12 e
Q12 style h1 Q12 IH
Q12 = style Q5 e
Q12 class h1 Q12 IH
Q12 = class Q5 e
Q12 h2 > Q12 H>
Q12 > h2 Q5 >H
Q5 /h2 h2 Q12 e
Q12 id h2 Q12 IH
Q12 = id Q5 e
Q5 petik2 h2 Q5 "H
Q5 petik2 petik2 Q12 e
Q5 petik h2 Q5 "H
Q5 petik petik Q12 e
Q12 style h2 Q12 IH
Q12 class h2 Q12 IH
Q12 h3 > Q12 H>
Q12 > h3 Q5 >H
Q5 /h3 h3 Q12 e
Q12 id h3 Q12 IH
Q12 = id Q5 e
Q5 petik2 h3 Q5 "H
Q5 petik2 petik2 Q12 e
Q5 petik h3 Q5 "H
Q5 petik petik Q12 e
Q12 style h3 Q12 IH
Q12 class h3 Q12 IH
Q12 h4 > Q12 H>

```



```

Q12 > h4 Q5 >H
Q5 < > Q5 e
Q5 e > Q5 >
Q5 /h4 h4 Q12 e
Q12 id h4 Q12 IH
Q12 = id Q5 e
Q5 petik2 h4 Q5 "H
Q5 petik2 petik2 Q12 e
Q5 petik h4 Q5 "H
Q5 petik petik Q12 e
Q12 style h4 Q12 IH
Q12 class h4 Q12 IH
Q12 h5 > Q12 H>
Q12 > h5 Q5 >H
Q5 /h5 h5 Q12 e
Q12 id h5 Q12 IH
Q12 = id Q5 e
Q5 petik2 h5 Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik h5 Q5 "H
Q5 e petik Q5 "
Q5 petik petik Q12 e
Q12 style h5 Q12 IH
Q12 class h5 Q12 IH
Q12 h6 > Q12 H>
Q12 > h6 Q5 >H
Q5 /h6 h6 Q12 e
Q12 id h6 Q12 IH
Q12 = id Q5 e
Q5 petik2 h6 Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik h6 Q5 "H
Q5 e petik Q5 "
Q5 petik petik Q12 e
Q12 style h6 Q12 IH
Q12 class h6 Q12 IH
Q12 a > Q12 H>
Q12 > a Q5 >H
Q5 /a a Q12 e
Q12 id a Q12 IH
Q12 = id Q5 e
Q5 petik2 a Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik a Q5 "H
Q5 e petik Q5 "
Q5 petik petik Q12 e
Q12 style a Q12 IH

```

```

Q12 class a Q12 IH
Q12 href a Q12 IH
Q12 = href Q5 e
Q12 button > Q12 H>
Q12 > button Q5 >H
Q5 /button button Q12 e
Q12 id button Q12 IH
Q12 = id Q5 e
Q5 petik2 button Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik button Q5 "H
Q5 e petik Q5 "
Q5 petik petik Q12 e
Q12 style button Q12 IH
Q12 class button Q12 IH
Q5 "submit" type1 Q12 e
Q5 'submit' type1 Q12 e
Q5 "reset" type1 Q12 e
Q5 'reset' type1 Q12 e
Q5 "button" type1 Q12 e
Q5 'button' type1 Q12 e
Q5 "submit" type2 Q12 e
Q5 'submit' type2 Q12 e
Q5 "reset" type2 Q12 e
Q5 'reset' type2 Q12 e
Q5 "button" type2 Q12 e
Q5 'button' type2 Q12 e
Q12 type1 button Q5 IH
Q12 type2 button Q5 IH
Q12 p > Q12 H>
Q12 > p Q5 >H
Q5 /p p Q12 e
Q12 id p Q12 IH
Q12 = id Q5 e
Q5 petik2 p Q5 "H
Q5 e petik2 Q5 "
Q5 petik2 petik2 Q12 e
Q5 petik p Q5 "H
Q5 e petik Q5 "
Q5 petik petik Q12 e
Q12 style p Q12 IH
Q12 class p Q12 IH
Q5 h1 > Q5 H>
Q5 > h1 Q6 >H
Q6 /h1 h1 Q5 e
Q5 id h1 Q5 IH
Q5 = id Q6 e
Q6 petik2 h1 Q6 "H
Q6 e petik2 Q6 "

```

```

Q6 petik2 petik2 Q5 e
Q6 petik h1 Q6 "H
Q6 e petik Q6 "
Q6 "GET" > Q6 "
Q6 'GET' > Q6 "
Q6 "POST" > Q6 "
Q6 'POST' > Q6 "
Q6 "get" > Q6 "
Q6 'get' > Q6 "
Q6 "checkbox" > Q6 "
Q6 'checkbox' > Q6 "
Q6 "number" > Q6 "
Q6 'number' > Q6 "
Q6 "text" > Q6 "
Q6 'text' > Q6 "
Q6 "email" > Q6 "
Q6 'email' > Q6 "
Q6 "password" > Q6 "
Q6 'password' > Q6 "
Q6 "submit" > Q6 "
Q6 'submit' > Q6 "
Q6 "reset" > Q6 "
Q6 'reset' > Q6 "
Q6 "button" > Q6 "
Q6 'button' > Q6 "
Q6 = > Q6 "
Q6 petik petik Q5 e
Q5 style h1 Q5 IH
Q5 = style Q6 e
Q5 class h1 Q5 IH
Q5 = class Q6 e
Q5 h2 > Q5 H>
Q5 > h2 Q6 >H
Q6 /h2 h2 Q5 e
Q5 id h2 Q5 IH
Q5 = id Q6 e
Q6 petik2 h2 Q6 "H
Q6 petik2 petik2 Q5 e
Q6 petik h2 Q6 "H
Q6 petik petik Q5 e
Q5 style h2 Q5 IH
Q5 class h2 Q5 IH
Q5 h3 > Q5 H>
Q5 > h3 Q6 >H
Q6 /h3 h3 Q5 e
Q5 id h3 Q5 IH
Q5 = id Q6 e
Q6 petik2 h3 Q6 "H
Q6 petik2 petik2 Q5 e
Q6 petik h3 Q6 "H

```

```

Q6 petik petik Q5 e
Q5 style h3 Q5 IH
Q5 class h3 Q5 IH
Q5 h4 > Q5 H>
Q5 > h4 Q6 >H
Q6 < > Q6 e
Q6 e > Q6 >
Q6 /h4 h4 Q5 e
Q5 id h4 Q5 IH
Q5 = id Q6 e
Q6 petik2 h4 Q6 "H
Q6 petik2 petik2 Q5 e
Q6 petik h4 Q6 "H
Q6 petik petik Q5 e
Q5 style h4 Q5 IH
Q5 class h4 Q5 IH
Q5 h5 > Q5 H>
Q5 > h5 Q6 >H
Q6 /h5 h5 Q5 e
Q5 id h5 Q5 IH
Q5 = id Q6 e
Q6 petik2 h5 Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e
Q6 petik h5 Q6 "H
Q6 e petik Q6 "
Q6 petik petik Q5 e
Q5 style h5 Q5 IH
Q5 class h5 Q5 IH
Q5 h6 > Q5 H>
Q5 > h6 Q6 >H
Q6 /h6 h6 Q5 e
Q5 id h6 Q5 IH
Q5 = id Q6 e
Q6 petik2 h6 Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e
Q6 petik h6 Q6 "H
Q6 e petik Q6 "
Q6 petik petik Q5 e
Q5 style h6 Q5 IH
Q5 class h6 Q5 IH
Q5 a > Q5 H>
Q5 > a Q6 >H
Q6 /a a Q5 e
Q5 id a Q5 IH
Q5 = id Q6 e
Q6 petik2 a Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e

```

```

Q6 petik a Q6 "H
Q6 e petik Q6 "
Q6 petik petik Q5 e
Q5 style a Q5 IH
Q5 class a Q5 IH
Q5 href a Q5 IH
Q5 = href Q6 e
Q5 button > Q5 H>
Q5 > button Q6 >H
Q6 /button button Q5 e
Q5 id button Q5 IH
Q5 = id Q6 e
Q6 petik2 button Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e
Q6 petik button Q6 "H
Q6 e petik Q6 "
Q6 petik petik Q5 e
Q5 style button Q5 IH
Q5 class button Q5 IH
Q6 "submit" type1 Q5 e
Q6 'submit' type1 Q5 e
Q6 "reset" type1 Q5 e
Q6 'reset' type1 Q5 e
Q6 "button" type1 Q5 e
Q6 'button' type1 Q5 e
Q6 "submit" type2 Q5 e
Q6 'submit' type2 Q5 e
Q6 "reset" type2 Q5 e
Q6 'reset' type2 Q5 e
Q6 "button" type2 Q5 e
Q6 'button' type2 Q5 e
Q5 type1 button Q6 IH
Q5 type2 button Q6 IH
Q5 p > Q5 H>
Q5 > p Q6 >H
Q6 /p p Q5 e
Q5 id p Q5 IH
Q5 = id Q6 e
Q6 petik2 p Q6 "H
Q6 e petik2 Q6 "
Q6 petik2 petik2 Q5 e
Q6 petik p Q6 "H
Q6 e petik Q6 "
Q6 petik petik Q5 e
Q5 style p Q5 IH
Q5 class p Q5 IH

```

## BAB IV

### IMPLEMENTASI PROGRAM

#### 4.1. **Regex.py**

Modul ini dibuat untuk membaca file program HTML yang akan diuji. Modul ini akan melakukan pembacaan tiap tag, atribut, kata, ataupun simbol dan mengubahnya menjadi sebuah token.

No.	Fungsi / Prosedur	Tujuan
1.	<code>comparing</code>	Mencocokkan tiap tag, atribut, kata atau simbol pada file HTML yang dibaca dengan aturan token yang dibuat dan menyimpannya pada sebuah list.
2.	<code>createToken</code>	Membaca file HTML dan memproses file HTML menjadi sebuah list token menggunakan fungsi <code>comparing</code> .

#### 4.2. **Openfile.py**

Modul ini digunakan untuk membaca file `pda.txt` yang berisi *states*, *productions*, *starting state*, dll. Setelah membaca file, bagian dari PDA akan dibagi menjadi 8 bagian PDA yang terdiri dari *states*, *input symbols*, *stack symbols*, *starting state*, *starting stack*, *accepting state*, dan *list of productions*.

No.	Fungsi / Prosedur	Tujuan
1.	<code>readFile</code>	Membaca file <code>pda.txt</code> per baris dan menyimpannya pada sebuah list yang selanjutnya akan di- <i>parse</i> per bagian.
2.	<code>parseFile</code>	Membagi list hasil <code>readFile</code> menjadi bagian-bagian pada DFA yaitu, <i>states</i> , <i>input symbols</i> , <i>stack symbols</i> , <i>starting state</i> , <i>starting stack</i> , <i>accepting state</i> , dan <i>list of productions</i> .

#### 4.3. **Error.py**

Modul ini digunakan untuk mencari lokasi error jika file html yang diperiksa menghasilkan syntax error atau hasil dari PDA tidak mencapai final state. Modul ini akan mencari letak dari kesalahan dan mencetak baris yang menyebabkan error pada file tersebut.

No.	Fungsi / Prosedur	Tujuan
1.	<code>searchposition</code>	Mencari lokasi kesalahan dari file html dengan menerima baris dan input simbol yang error. Fungsi akan mengembalikan posisi indeks kesalahan pada file html.
2.	<code>printerror</code>	Mencetak baris yang menyebabkan error pada program html tersebut.

#### 4.4. Main.py

Modul ini digunakan sebagai program utama untuk menjalankan aplikasi. Program utama akan menerima input file PDA dan program html yang akan dibaca menggunakan modul-modul di atas. Selanjutnya, list token yang berisi program html akan dicocokkan dengan pda dan dilakukan proses *push* dan *pop* pada stack sesuai dengan proses produksinya (jika ada). Jika tidak ada, maka pencocokan akan selesai dan program akan dinilai error. Di akhir akan dilihat apakah program berhasil mencapai final state atau tidak.

No.	Fungsi / Prosedur	Tujuan
1.	<code>printopening</code>	Fungsi untuk mencetak pembuka saat pengguna memulai menjalankan program.
2.	<code>printtrue</code>	Fungsi untuk mencetak pesan jika program html yang dibuat benar / accepted.
3.	<code>printfalse</code>	Fungsi untuk mencetak pesan jika program html yang dibuat error / masih terdapat kesalahan.
4.	<code>compute</code>	Mencocokkan token dari file html dengan list produksi yang ada pada DFA. Jika ada state atau top stack yang sesuai, maka akan dilakukan push/pop. Jika tidak ada proses produksi yang sesuai, program akan berhenti dan mencatat baris serta simbol yang menyebabkan error. Program dinilai benar jika mencapai final states setelah membaca seluruh simbol pada file html.
5.	<code>main</code>	Program utama yang menerima input file pda dan html yang akan diperiksa lalu keduanya akan diolah dengan modul dan fungsi yang ada di atas. Selanjutnya, jika benar program akan mencetak pesan accepted dan jika salah akan ada pesan error beserta lokasi errornya.

## BAB V

### PENGUJIAN

#### 5.1. Test Case 1

```
<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>
```





## 5.2. Test Case 2

```
<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>

    <div id="hs" class="dd" style="color:antiquewhite">
      
      <!-- hahah -->
      <link rel="ass" href="dd">
      <link rel="ass">
      <link href="dd" rel="ass" id ="ss" class="ss"
style="color:antiquewhite" >
      <button type="reset" id="hs" class="dd"
style="color:antiquewhite">sss</button>
      <button type="submit">ini submit</button>
      <button type="button">ini button</button>
      <hr>
      <br>
      <div id="hs" class="dd" style="color:antiquewhite">
        <h1>hshs<b>ss</b></h1>
        <h1>hshs<em>ss</em></h1>
      </div>
    </div>
    <!-- Custom element -->
    <div id="footer" class="footer"> This is the end of the
page </div>
  </body>
</html>
```



### 5.3. Test Case 3

```
<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>
    <form action="/action_page.php" method="POST">
      <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>
  </body>
</html>
```



#### 5.4. Test Case 4

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>ini bukan tombol "submit"</h1>
    <h2>juga bukan "GET"</h2>
    <table id="hs" class="dd" style="ddd:hh">
      <tr id="hs" class="dd" style="ddd:hh">
        <!-- hahah -->
        <th id="hs" class="dd" style="ddd:hh">Month</th>
        <th>Savings</th>
      </tr>
      <tr>
        <td>January</td>
        <td>$100</td>
      </tr>
    </table>
    <div id="footer" class="footer"> This is the end of the
page </div>
  </body>
</html>
```



### 5.5. Test Case 5

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



Parser menganggap program salah karena urutan tag body dan head terbalik dari yang seharusnya. Oleh karena itu, kode pada kode di atas tidak terdapat list produksi PDA yang sesuai sehingga ditolak.

## 5.6 Test Case 6

```
<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>
```



Parser menganggap program salah karena pada tag img tidak memiliki atribut src yang mana hal tersebut wajib ada untuk tag img. Oleh karena itu, kode pada kode di atas tidak terdapat list produksi PDA yang sesuai sehingga ditolak.

### 5.7. Test Case 7

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

</head>
<body>

<h2>HTML Forms</h2>

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



Parser menganggap program salah karena tag form nilai method TEMBAK tidak valid sebagai nilai atribut. Oleh karena itu, kode pada kode di atas tidak terdapat list produksi PDA yang sesuai sehingga ditolak.

### 5.8. Test Case 8

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

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



Parser menganggap program salah karena program html seharusnya hanya memiliki satu struktur saja untuk tag html, head, dan body. Oleh karena itu, kode pada kode di atas tidak terdapat list produksi PDA yang sesuai sehingga ditolak.



## **BAB VI**

### **SIMPULAN DAN SARAN**

#### **6.1 Simpulan**

Dalam tugas besar ini, kami berhasil mengimplementasikan konsep Teori Bahasa Formal dan Automata terkait konsep Push Down Automata (PDA) pada pengecekan sintaks HTML. PDA sendiri digunakan untuk melakukan pembacaan dan pengecekan setiap karakter HTML dalam sebuah file yang keseluruhan implementasinya menggunakan bahasa Python.

Struktur HTML terdiri dari elemen-elemen yang bersarang. Kami membuat aturan-aturan PDA dengan *final state* untuk memeriksa tag dan atribut HTML dan juga mengelola stack. Sehingga penerapan PDA untuk pengecekan sintaks HTML dapat bekerja secara otomatis sesuai sintaks HTML.

#### **6.2 Saran**

Berikut adalah saran-saran kami untuk pengembangan aplikasi PDA untuk HTML Checker lebih lanjut kedepannya.

1. Mempelajari konsep PDA secara matang sebelum merumuskan aturan-aturan PDA serta melakukan trial and error terhadap rancangan PDA yang telah dibuat untuk memastikan kebenaran dalam pengecekan nantinya.
2. Penanganan error lebih lanjut, dengan perluasan penanganan error akan sangat membantu pengguna memahami kesalahan yang terjadi dalam file HTML mereka.
3. Penanganan elemen lainya karena pembuatan checker ini kami prioritaskan untuk tag-tag elemen yang sifatnya wajib terlebih dahulu.

## LAMPIRAN

### A. Pembagian Tugas

Nama	Pembagian Tugas
Thea Josephine Halim	PDA, Laporan, Diagram State
Debrina Veisha Rashika W.	PDA, Program utama, Laporan
Melati Anggraini	PDA, Laporan, Diagram State

### B. Link Diagram State

<https://www.figma.com/file/0sjCvdBxRxxLX2iIrYKKsV/PDA-TBFO?type=design&node-id=0%3A1&mode=design&t=cz6YrBprBLo25k0Q-1>

### C. Link Repository Github

<https://github.com/pandaandsushi/Dongeng-Jawa-Petualangan-Thea-dan-Melati-di-Negeri-Automata-Milik-Penguasa-Jahat-Shika.git>