PERANCANGAN DAN PEMBUATAN SISTEM VALIDASI XHTML 1.0

Yulia[1)], Sukanto Tedjokusuma[2)], Nicko Candra[3)]
Jurusan Teknik Informatika, Universitas Kristen Petra Surabaya
yulia@petra.ac.id [1)], sukanto@petra.ac.id[2)]

ABSTRACT
The use of internet has been increasing rapidly. The number of published web pages increases as well. On the other hand, web pages have tags and syntaxes inside. Those tags and syntaxes are going to be rendered into an interface by a web browser. The web page needs to be correctly setup structurally and grammatically before it is published. A web page can be claimed correct or valid after it is checked by a validation system. A validation system contains 3 main processes, lexical analysis, parsing, and token checking processes. Those which are going to be checked are element name, attribute name, and attribute content based on the DTD's standard. Checking process includes the checking of structure of a web page. This validation system can help the web developer in creating a correct web page. This validation system can detect an error inside an XHTML file, including grammatical error and structural error.
Keywords: *XHTML, DTD, Lexical Analysis, Parsing, Checking, Token*

1. Pendahuluan

Situs *web* pada dasarnya adalah sekumpulan halaman *file* yang memiliki *file extension* .html, .htm, .php, .xhtml, dan sebagainya. Halaman – halaman *file* tersebut berisi *tag* maupun *syntax* yang akan dibaca oleh *web browser* sebagai halaman *web*. *Tag* serta *syntax* yang digunakan seharusnya berpedoman pada standar *Document Type Definition* (DTD) yang telah ditetapkan oleh W3C (*World Wide Web Consortium*). Seiring dengan perkembangan dunia *internet* dan *web*, maka jumlah pembuatan *file* halaman *web* juga meningkat. Isi *file* halaman *web* yang ditulis juga diharapkan telah sesuai dengan standar DTD yang ditetapkan oleh W3C. Hal ini diperlukan agar *web browser* dapat menerjemahkan *tag dan syntax* dalam sebuah halaman *web* dengan benar dan tepat. Karena kebutuhan ini, maka sistem validasi terhadap halaman *web* diperlukan.

Pada situs resmi W3C, yaitu http://www.w3c.org, telah tersedia fasilitas untuk melakukan validasi terhadap sebuah halaman *web*. Fasilitas tersebut sudah sangat baik dan lengkap, karena menyediakan berbagai macam standar DTD sebagai dasar validasi. Di antaranya adalah, HTML 4.01 *Transitional* DTD, HTML 4.01 *Strict* DTD, XHTML 1.0 *Transitional* DTD, XHTML 1.1 *Transitional* DTD, dan sebagainya. Namun fasilitas tersebut mengharuskan pengguna untuk *online* atau terhubung dengan koneksi *internet* untuk menggunakannya. Maka, diperlukan adanya sebuah aplikasi alternatif untuk melakukan validasi halaman *web*, tanpa mengharuskan penggunanya terhubung dengan koneksi *internet*.

Tujuan dari penelitian ini adalah membuat sistem validasi untuk sebuah halaman *web* berdasarkan standar untuk DTD XHTML 1.0 *Transitional.* Sedangkan rumusan masalah dari penelitian ini adalah:

1. Bagaimana melakukan proses *lexical analysis file* XHTML?
2. Bagaimana melakukan pembuatan *syntax tree* pada proses *parsing*?
3. Bagaimana melakukan proses validasi terhadap *token* yang telah melalui proses *parsing*?
4. Bagaimana melakukan proses pemeriksaan dan validasi token berdasarkan standar XHTML 1.0 *Transitional Document Type Definiton*.
5. Bagaimana melakukan proses pemeriksaan struktural *syntax* terhadap halaman *web* XHTML.

2. Landasan Teori

2.1 Extensible Hyper Text Markup Language (XHTML)

XHTML adalah sebuah aplikasi *Standard Generalized Markup Language* (SGML). Berbeda dengan *Hyper Text Markup Language* (HTML) yang merupakan aplikasi SGML yang sangat fleksibel, XHTML dibuat sebagai aplikasi SGML yang lebih ketat syarat penggunaannya. XHTML diharuskan untuk *well-formed*. XHTML dapat juga disebut sebagai penggabungan antara HTML dengan *Extensible Markup Language* (XML)[3]. Perbedaan antara XHTML 1.0 *Strict* dengan XHTML 1.1 adalah:

- Atribut lang digantikan atrribut xml:lang
- Atribut name pada elemen a dan map diganti dengan atribut id
- Ditambahkan koleksi elemen ruby

Dalam banyak hal, *grammar* HTML dengan XHTML tidak banyak berbeda. Namun terdapat beberapa perbedaan penting pada XHTML, antara lain:

- Nama elemen harus ditulis menggunakan huruf kecil

  Contoh :

  Salah:

  ```
  <FONT size="14">coba</font>
  ```

  Benar:

  ```
  <font size="14">coba</font>
  ```

- Nama atribut harus ditulis menggunakan huruf kecil
- Nilai atribut harus diawali dan diakhiri oleh tanda petik ganda (“) atau tanda petik tunggal (‘)
- Minimalisasi nilai atribut tidak diperbolehkan
- Penggunaan atribut id menggantikan atribut name

2.2 Document Type Definition (DTD)

DTD merupakan *grammar* dari sebuah SGML. DTD memuat tentang hal – hal yang diperbolehkan ada dalam sebuah SGML. Untuk menggunakan DTD dalam sebuah SGML, perlu dilakukan deklarasi dalam dokumen. Deklarasi itu disebut Document Type Declaration, yang juga biasa disebut DOCTYPE. Deklarasi dilakukan pada bagian awal dokumen. Contoh deklarasi DTD adalah: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR html4/strict.dtd">. Deklarasi DOCTYPE di atas adalah deklarasi penggunaan DTD HTML 4.01 Stric[6]. Terdapat tiga macam DTD untuk XHTML 1.0. yaitu DTD strict, DTD *Transitional*, DTD frameset[6].

2.3 Parsing

Parsing merupakan proses menganalisa sebuah atau beberapa urutan *token* untuk menentukan struktur *grammatical* dari sebuah bahasa pemrograman. Dalam sebuah *compiler*, terdapat bagian yang disebut dengan nama *parser*, yang bertugas melakukan *parsing*. Proses *parsing* mengubah sebuah *input text* ke dalam struktur data, biasanya sebuah *tree*. Struktur data ini nantinya digunakan untuk proses lebih lanjut serta dapat mendeteksi hirarki dari sebuah *input*. Proses dari *parsing* melewati beberapa tahap. Yang pertama adalah *token generation*, atau *lexical analysis*, di mana karakter *input* dipisah menjadi simbol – simbol (*token*) yang bermakna dalam sebuah *grammar*. Tahap berikutnya adalah *parsing* atau *syntactic analysis*, di mana terjadi proses pemeriksaan *token* terhadap ekspresi standar. Tahap akhir adalah *semantic parsing* atau analisa *semantic*. Dalam tahap ini, kode akan dievaluasi/di-*generate* menjadi sebuah hasil[2].

2.3 Non-deterministic Finite Automata (NFA)

Non-deterministic Finite Automata (NFA) adalah salah satu jenis dari *Finite State Machine*. *Finite State Machine* biasa digunakan untuk menggambarkan alur dari sebuah bahasa maupun untuk otomatisasi. NFA merupakan sebuah diagram yang menggambarkan hubungan antara satu kondisi dengan kondisi lain. NFA juga dapat menjelaskan faktor apakah yang menyebabkan perubahan dari sebuah kondisi ke kondisi lainnya[2].

3. Metodologi

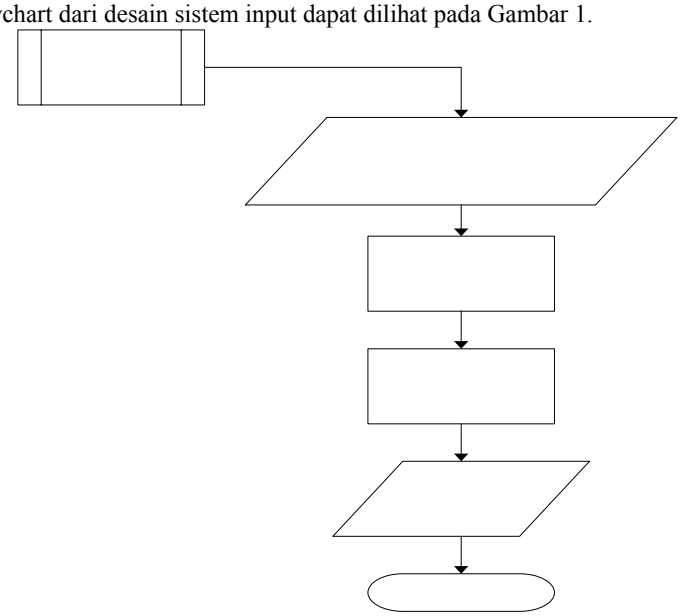
Metodologi dalam penelitian ini adalah:

1) Sistem *Input* adalah sistem pertama yang dilakukan dalam aplikasi validasi sebuah *file* XHTML.
2) Sistem Validasi XHTML (proses) merupakan sistem proses inti dari aplikasi ini.
3) Sistem *Output* adalah sistem terakhir yang dilakukan oleh sistem validasi XHTML ini.

4. Desain Sistem

4.1 Desain Sistem Input

Sistem *Input* diperlukan untuk memberi *input* pada sistem validasi ini. *Input* untuk sistem validasi ini berasal dari tiga sumber yaitu dari *file* yang berasal dari komputer lokal, URL *internet* dan dari *text* yang dapat diketikkan langsung dalam sebuah *rich edit text*. Flowchart dari desain sistem input dapat dilihat pada Gambar 1.



Gambar 1. *Flowchart* Sistem *Input*

4.2 Desain Sistem Validasi

Input dari sistem ini adalah *text* yang berasal dari *rich edit text* dari sistem sebelumnya. *Output* dari sistem ini berupa sebuah *state* yang menunjukkan kondisi dari *text* yang di-*input*-kan. *State* tersebut menjadi *input* untuk sistem selanjutnya. Sistem ini akan berjalan terus menerus hingga ditemukan *state* salah atau hingga *text* telah habis diperiksa semuanya. Dalam sistem validasi XHTML ini, terdapat tiga proses penting. Proses tersebut adalah *Lexical Analysis* XHTML, *Parsing* XHTML dan Pemeriksaan XHTML.

4.2.1 Proses Lexical Analysis XHTML

Sistem *lexical analysis* diperlukan untuk membaca *text* dalam *file* XHTML yang telah dibuka, sehingga dapat diambil *token-token*-nya. *Token-token* inilah nantinya yang diproses untuk sistem validasi.

4.2.2 Proses Parsing XHTML

Proses *parsing* diperlukan untuk membaca *token-token* yang dihasilkan oleh proses *lexical analysis* sebelumnya. Proses *parsing* juga menterjemahkan *token-token* tersebut menjadi suatu *state*. *State* tersebutlah yang menentukan *state* proses *parsing* selanjutnya.

4.2.3 Proses Pemeriksaan XHTML

Proses ini dijalankan tepat setelah proses *parsing* menghasilkan *token* elemen dan atribut yang disimpan dalam *pointer of string*. Proses pemeriksaan dilakukan tepat setelah proses *parsing token* dari *file* XHTML selesai dijalankan secara elemen per elemen (berdasarkan hasil proses *lexical analysis*). Gambar 2. merupakan NFA state sistem validasi.

4.3 Desain Sistem Output

Sistem ini merupakan hasil *output* dari keseluruhan sistem validasi ini. Sistem mengembalikan hasil validasi sesuai dengan kondisi hasil yang didapatkan pada sistem validasi sebelumnya. Pada sistem *Output*, diperiksa kondisi apakah yang diterima dari sistem validasi sebelumnya. Ada beberapa kondisi yang bernilai sebagai hasil validasi yang salah. Hanya ada sebuah kondisi yang bernilai sebagai hasil validasi yang benar. Flowchart sistem output dapat dilihat pada Gambar 3.

5. Pengujian

5.1 Pemeriksaan Elemen dan Atribut

Pada *source code file* XHTML yang diuji, nama elemen *table* diganti dengan *tablfe*. Nama elemen *tablfe* bukan merupakan nama elemen yang valid berdasarkan DTD XHTML 1.0 *Transitional*. Kemudian dilakukan proses validasi. Perangkat lunak sistem validasi XHTML 1.0 menunjukkan terjadi kesalahan elemen *tablfe* dalam elemen *body* pada baris ke-53. Baris di mana terjadi kesalahan dicetak dengan warna merah. Gambar 4 ini adalah *screenshot* dari pengujian.

5.2 Pemeriksaan Struktur *File* XHTML

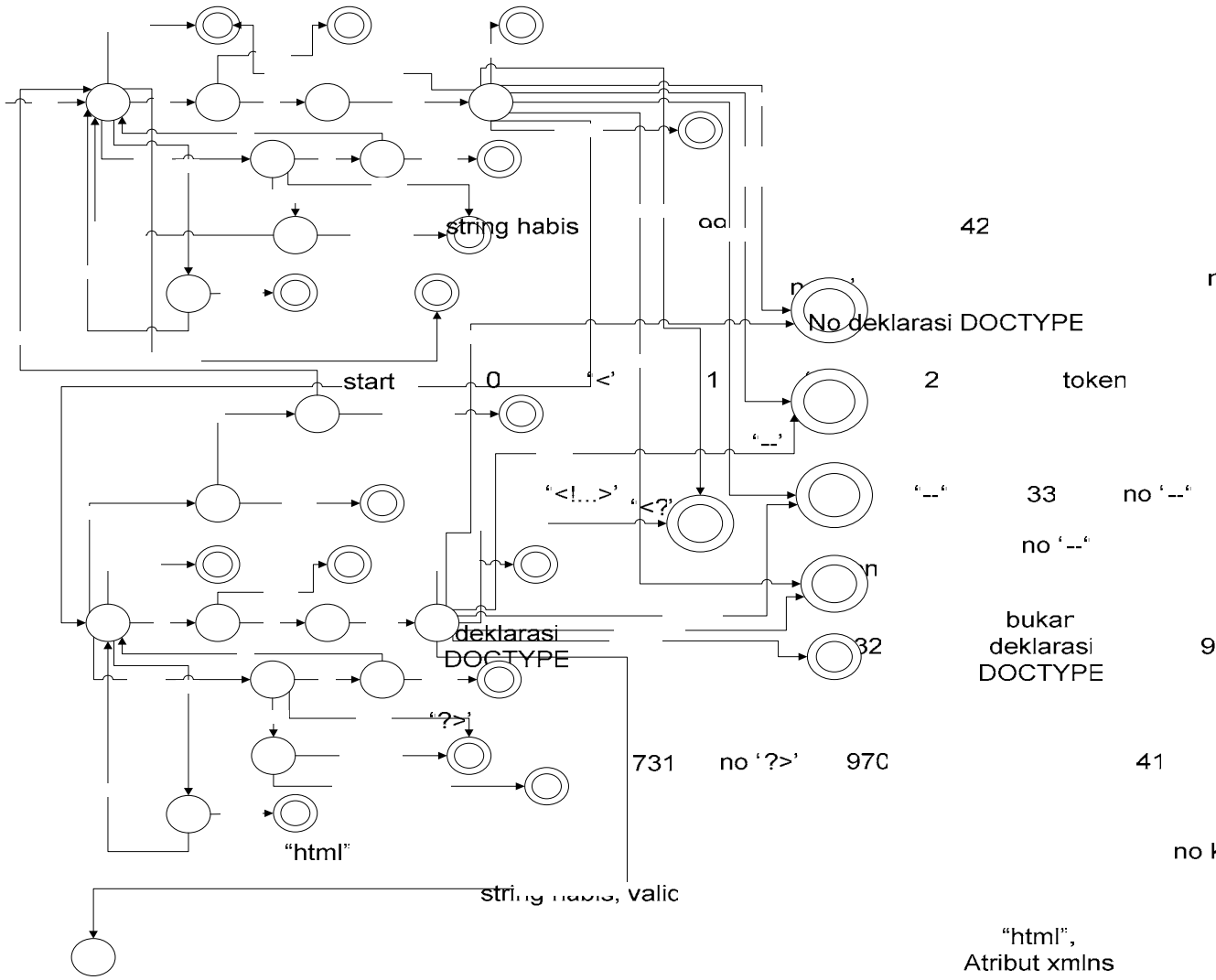
Pada *source code file* XHTML yang diuji, deklarasi DOCTYPE dihapus. Struktur XHTML 1.0 yang valid harus memiliki satu deklarasi DOCTYPE di awal *file*. Kemudian dilakukan proses validasi. Perangkat lunak sistem validasi XHTML 1.0 menunjukkan terjadi kesalahan yakni tidak ada deklarasi DOCTYPE pada baris ke-1. Baris di mana terjadi kesalahan dicetak dengan warna merah. Gambar 5 adalah *screenshot* dari pengujian.

5.2 Pengujian Perbandingan Hasil dengan Sistem Validasi

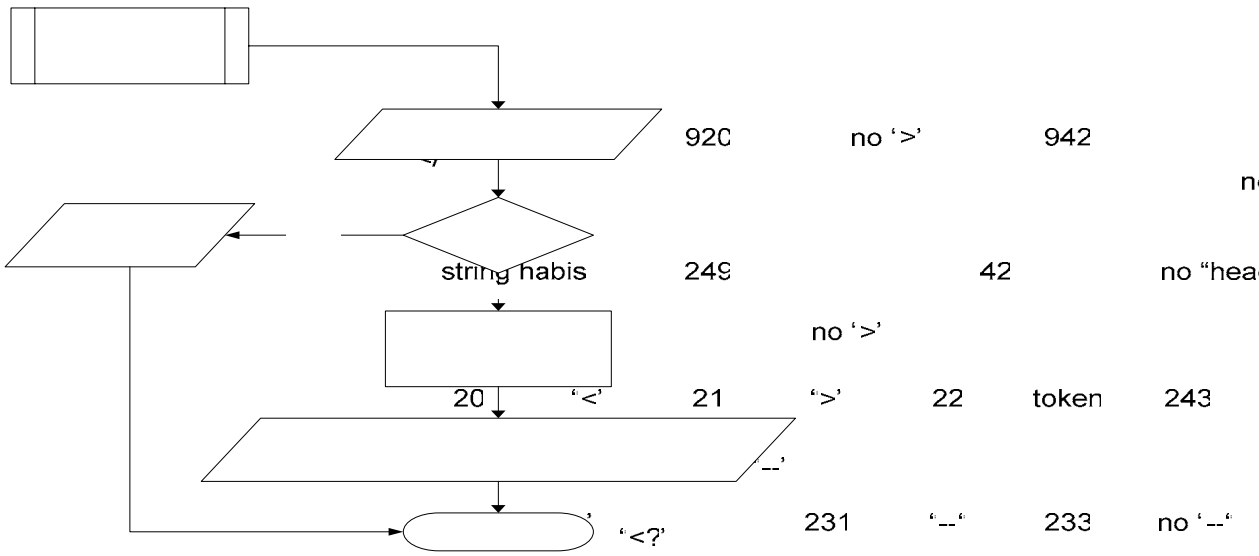
Pada pengujian perbandingan hasil dengan sistem validasi http://w3.org, akan diuji perbandingan *output* dari kedua sistem. Pada *source code file* XHTML yang diuji tidak dilakukan perubahan apa-apa. Ketika dilakukan proses validasi menggunakan sistem validasi pada http://w3.org, dinyatakan bahwa *file* tersebut telah valid sesuai DTD XHTML 1.0 *Transitional*. *Screenshot* pengujian dapat dilihat pada Gambar 6. Ketika dilakukan proses validasi menggunakan perangkat lunak yang dibuat penulis, muncul pesan bahwa *file* tersebut valid sesuai DTD XHTML 1.0 *Transitional*. Gambar 7. berikut ini adalah *screenshot* dari pengujian menggunakan perangkat lunak.

6. Kesimpulan

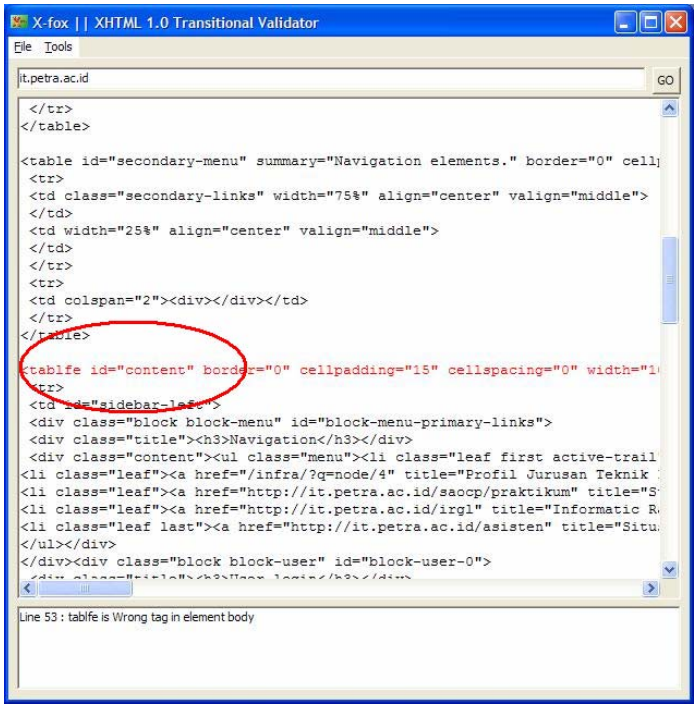
Sistem yang dikembangkan telah terbukti dapat melakukan validasi terhadap dokumen XHML baik dari segi elemen dan atribut yang terdapat di dalam dokumen, maupun dari segi struktur filenya. Salah satu kelebihan sistem ini dibandingkan dengan sistem lainnya adalah kemampuannya untuk digunakan secara offline, sehingga pengguna dapat tetap menggunakan sistem validasi walaupun dia tidak terkoneksi ke Internet.



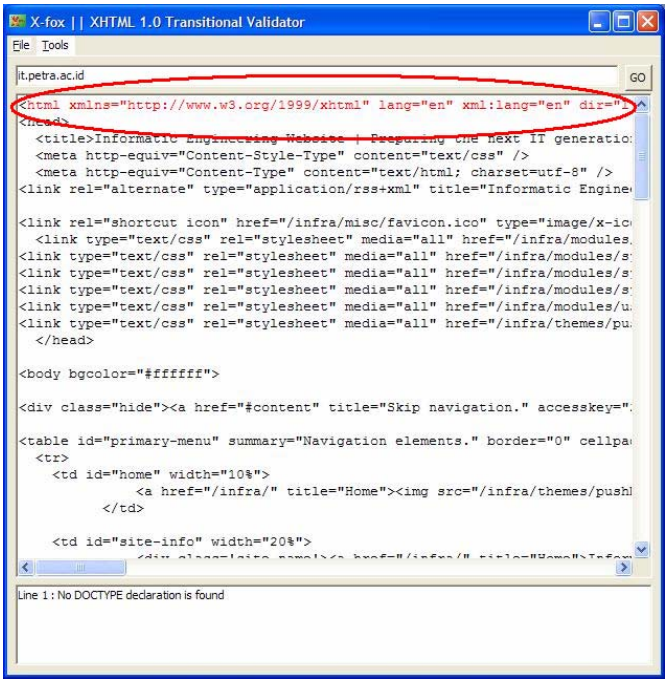
Gambar 2. NFA *State* Sistem Validasi



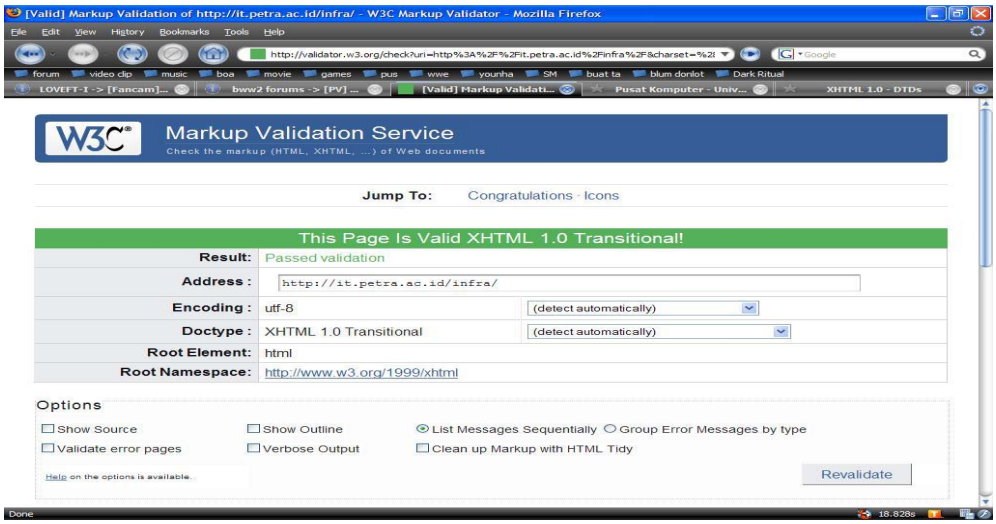
Gambar 3. *Flowchart* Sistem *Output*



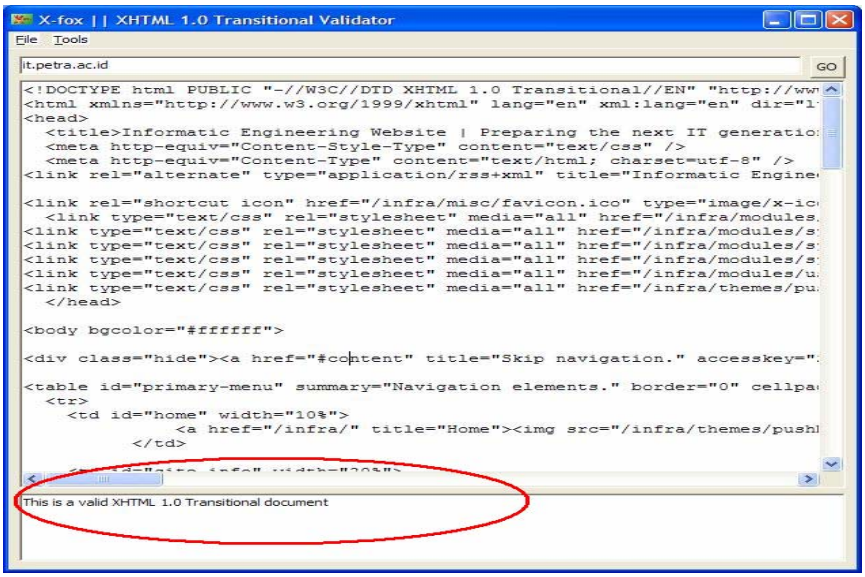
Gambar 4. Pengujian Kesalahan Nama Elemen



Gambar 5. Pengujian Kesalahan Tidak Ada DOCTYPE



Gambar 6. Halaman validasi *file* XHTML http://it.petra.ac.id dengan http://w3.org



Gambar 7. Halaman validasi *file* XHTML http://it.petra.ac.id dengan Perangkat Lunak

Daftar Pustaka

[1] *DATASTEAD Software* (2007). ThttpScan, http://www.datastead.com/httpscan/index.htm, diakses terakhir tanggal 20 April 2008.

[2] Hariyanto, Bambang (2004). *Teori Bahasa, Otomata, dan Komputasi Serta Terapannya*. Informatika, Bandung.

[3] *W3Schools Online Web Tutorials*. (2007). XHTML vs HTML, http://www.w3schools.com/xhtml/xhtml_html.asp, diakses terakhir tanggal 20 November 2007.

[4] *W3Schools Online Web Tutorials*. (2007). XHTML Syntax, http://www.w3schools.com/xhtml/xhtml_*syntax*.asp, diakses terakhir tanggal 20 November 2007.

[5] *World Wide Web Consortium - Web Standards*. (2007). Validator, http://validator.w3.org, diakses terakhir tanggal 1 November 2007.

[6] *World Wide Web Consortium - Web Standards*. (2002). XHTML 1.0 – DTDs, http://www.w3.org/TR/ xhtml1/dtds.html, diakses terakhir tanggal 4 November 2007.