

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Hypertext Markup Language* (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi didalam sebuah penjelajah web internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format *ASCII* normal sehingga menjadi halaman web dengan perintah-perintah HTML. HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (*W3C*). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa)[1]. Pada dasarnya Web dokumen ditulis menggunakan bahasa markup, seperti HTML, HTML 5 atau XHTML. Setiap mahasiswa yang tertarik dalam pengembangan website harus memiliki kemampuan dalam merancang bahasa *markup* HTML.

Akan tetapi bahasa *markup* HTML memiliki banyak struktur *code* dengan standar penulisan yang berbeda-beda dan walaupun penulisan *code* HTML tidak

sesuai dengan standar yang dikeluarkan W3C ini tidak menimbulkan adanya *error* dalam tampilan sehingga ketika seseorang mulai belajar HTML dia tidak mengetahui apakah code yang disusun telah sesuai dengan standar dan terdapat kesalahan atau tidak. Oleh sebab itu untuk membantu proses pembelajaran HTML diperlukan sebuah sistem yang dapat melakukan penilaian apakah kode yang di susun itu telah sesuai dengan standar dan tidak memiliki kesalahan.

Sistem penilaian kode HTML pada matakuliah pemrograman internet di Universitas Islam Negeri Sunan Gunung Djati Bandung masih memiliki sistem penilaian yang masih terbilang memakan banyak waktu, dimana dosen harus memeriksa satu persatu kode HTML setiap mahasiswa dan mengecek hasil atau *output* dari kode HTML tersebut selanjutnya dosen memberikan nilai kepada mahasiswa sesuai dengan penilaian dosen terhadap kode dan hasil kode HTML, hal ini menyebabkan konsumsi waktu yang banyak untuk proses penilaian kode HTML mahasiswa.

Sistem penilaian yang efektif dan efisien dalam pengecekan kode program yaitu menggunakan *autograder* hal ini dibuktikan dari penelitian sebelumnya yang menyebutkan bahwa *autograder* telah terbukti menjadi hakim yang efektif dan efisien dalam kontes pemrograman pada tingkat Internasional [3]. *Autograder* adalah suatu aplikasi atau modul program untuk melakukan penilaian atau *assessment* secara otomatis. Salah satu contoh sistem *autograder source code* untuk Java, C++ dan Python yang diintegrasikan dengan *CodeMirror* dapat diaplikasi dalam dunia pendidikan[4]. Dari penelitian international yang membahas mengenai *autograder* belum ada yang mengembangkan *autograder* untuk bahasa bahasa markup HTML [3] [4] [5] [6] [7] [8] [9] [10]. Sedangkan penelitian pada tingkat

internasional yang berjudul *Auto-Locating and Fix-Propagating for HTML Validation Errors to PHP Server-side Code* membahas mengenai sistem validator untuk HTML, akan tetapi sistem validator pada penelitian tersebut dianggap masih memiliki beberapa kekurangan diantaranya sistem penilaian tidak memberikan penilaian code yang telah di validasi [10]. Adapun aplikasi berbasis website yang sudah ada dalam validator HTML yang di keluarkan langsung oleh W3C validator masih ada beberapa kekurangan termasuk fitur sistem penilaian code yang tidak tersedia. Berdasarkan latar belakang yang telah dijelaskan diatas maka dibuatlah penelitian yang berjudul “Modul *Autograder* untuk HTML” dengan menggunakan sistem penilaian *code* HTML yang dilengkapi dengan sistem *real time feedback* atau sistem koreksi terhadap kesalahan pengetikan *grammar* HTML yang dilakukan oleh *user*, yang dapat memberikan informasi mengenai kesalahan yang dilakukan oleh *user* dalam penulisan kode HTML.

## 1.2 Rumusan Masalah

Penelitian ini mengkaji mengenai proses pengecekan dan penilaian sebuah *source code* HTML secara otomatis. Masalah - masalah yang akan dikaji dalam penelitian ini adalah :

1. Bagaimana menentukan sebuah skema penilaian *source code* HTML ?
2. Bagaimana memberikan *feedback source code* secara *real time* ?
3. Bagaimana membuat sebuah sistem penilaian *source code* HTML secara otomatis ?

## 1.3 Batasan Penelitian

Pada penelitian ini ditetapkan beberapa batasan masalah diantaranya :

1. *Source code* yang akan dinilai dalam sistem ini hanya *source code* yang menggunakan bahasa Markup HTML.
2. Sistem ini menghasilkan *output* penilaian *source code* HTML yang dikelompokkan menjadi berbagai kategori yaitu sukses (*match*) dan *didn't match*.

#### **1.4 Tujuan Penelitian**

Tujuan yang hendak dicapai dalam penelitian ini adalah :

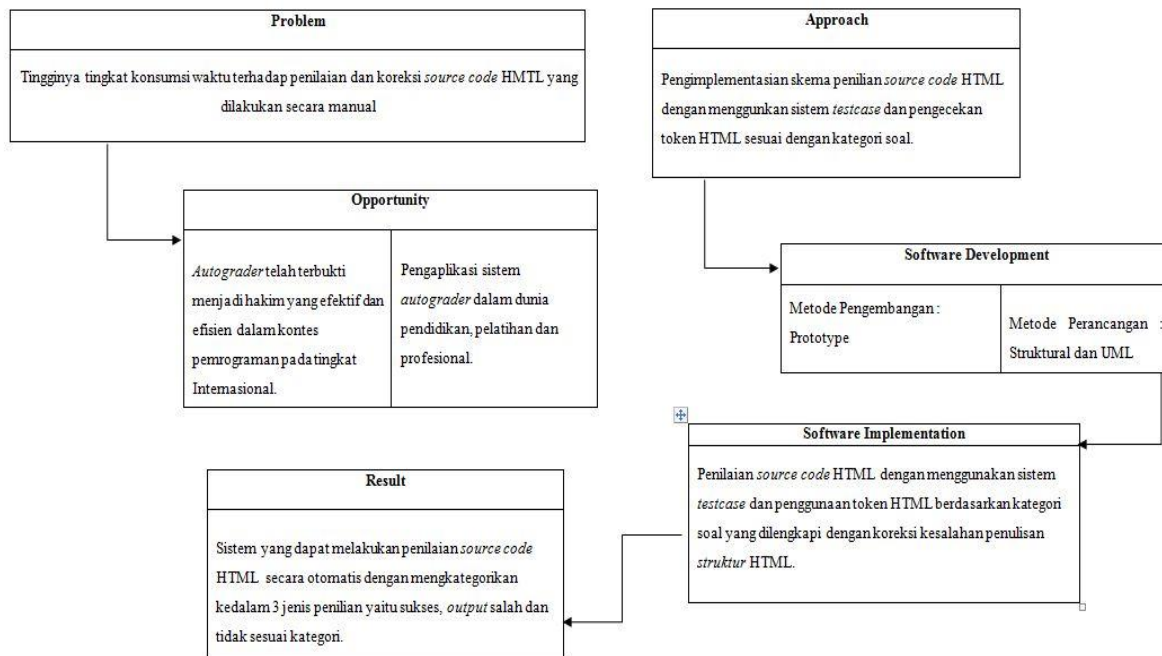
- 1 Menentukan skema penilaian terhadap *source code* HTML
- 2 Memberikan koreksi terhadap ketidak sesuaian *source code* yang dilakukan secara *real time*.
- 3 Membuat sebuah sistem penilaian *source code* HTML secara otomatis.

#### **1.5 Manfaat Penelitian**

Dengan adanya penelitian ini maka penilaian terhadap *source code* dan koreksi program dengan kode program yang menggunakan bahasa Markup HTML dapat lebih efektif dan efisien sehingga tingkat keterampilan pemrogramannya dapat terlatih untuk menyelesaikan sebuah masalah dalam membuat Markup HTML.

## 1.6 Kerangka Pemikiran

Berdasarkan tinjauan pustaka yang telah dijelaskan maka kerangka pemikiran dapat diuraikan sebagai berikut yang terdapat dalam gambar 1.1 di bawah ini:



Gambar 1.1 Kerangka Pemikiran Penelitaian

## 1.7 Metodologi Penelitian

Metodelogi yang digunakan dalam penelitian kali ini adalah sebagai berikut :

### 1.1.1 Teknik Pengumpulan Data

Teknik yang digunakan pada pengumpulan data yang dilakukan kali ini terdiri dari 2 tahapan, yaitu :

1. Observasi

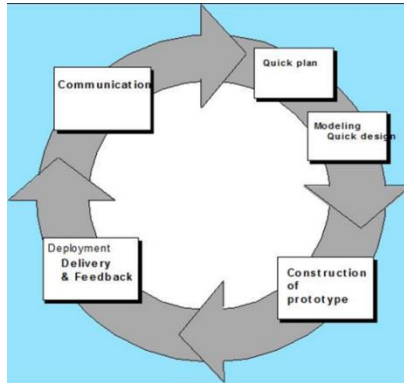
Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

## 2. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul tugas akhir.

### 1.1.2 Metode Pengembangan

Metode pengembangan perangkat lunak yang digunakan yaitu *model prototype*. Model pengembangan ini dipilih karena perangkat lunak yang di kembangkan memiliki kebutuhan yang belum jelas secara keseluruhan sehingga metode yang cocok di gunakan untuk pengembangan jenis ini adalah *prototype*. Proses model dalam pembangunan sistem model *prototype* terdiri dari *communication, quick plan, modelling quick design, construction of prototype, deployment delivery & feedback* [12]. Dengan menggunakan metode *prototype* ini pengembang dan pelanggan dapat saling berinteraksi selama pembuatan sistem. Metode *prototype* ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali, oleh karena itu dalam membangun “*Modul HTML auto grader*” ini di terapkan Metode *prototype*.



**Gambar 1.2** Prototype model [12]

Dari skema model *prototype* dalam pengembangan perangkat lunak di atas dapat dijelaskan bahwa setiap langkah proses memiliki arti tersendiri berikut ini deskripsi dari masing masing proses yang ada dalam prototype model.

1. *Communication*
2. *Quick Plan*
3. *Modelling Quick Design*
4. *Construction of Prototype*
5. *Deployment Delivery & Feedback*

### **1.8 Sistematika Penulisan**

Untuk memudahkan dalam melakukan penyusunan skripsi ini maka dibuatlah sebuah sistematika penulisan yang terdiri dari bagian bagian berikut ini :

### **BAB I PENDAHULUAN**

Bab ini menguraikan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, *state of the art*, kerangka pemikiran, metodologi penelitian dan sistematika penulisan.

## **BAB II LANDASAN TEORI**

Bab ini membahas mengenai teori-teori yang digunakan untuk memperkuat pernyataan-pernyataan yang berkaitan dengan judul skripsi yang berkorelasi dengan teori yang telah ada. Selain itu dalam bab ini menjelaskan berbagai teori – teori pendukung untuk digunakan penyusunan skripsi ini.

## **BAB III ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan mengenai analisis dan perancangan sistem yang akan dibuat yang diawali dengan penjelesan mengenai sistem yang berjalan dan kemudian melakukan analisis terhadap sistem tersebut. Selanjutnya membuat tentang desain sistem, basis data maupun mengenai *interfacenya*. Pada bab ini analisis dan perancangan yang diuraikan adalah analisis sistem operasi, arsitektur sistem, perancangan sistem, perancangan basis data, dan perancangan aplikasi.

## **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini akan menguraikan mengenai implementasi dari perancangan yang telah dibuat serta pengujian aplikasi yang telah dianalisa dan dirancang sebelumnya. Bab ini membahas mengenai perangkat lunak pembangun, perangkat keras pembangun, implementasi basis data, implementasi antar muka, pengujian perangkat lunak.

## **BAB V PENUTUP**



Bab ini menjelaskan mengenai kesimpulan, usulan, solusi dan saran terhadap aplikasi yang telah dibangun.

