

BAB I

Pendahuluan

1.1. Latar Belakang

Ketika mahasiswa baru Ilmu Komputer diberi sebuah tugas pemrograman yang harus dikumpulkan setiap minggunya, hasil yang didapat tidak begitu baik. Mereka mengalami kesulitan dalam mempelajari pemrograman, bahkan banyak diantaranya yang sampai menyerah dengan frustrasi, atau pada akhirnya mereka terlalu mendapat banyak bantuan, yang mengakibatkan gagalnya dalam memahami materi pemrograman (Colton et al., 2006).

Kegagalan mahasiswa baru dalam membuat program mingguan dikarenakan banyaknya materi baru yang diberikan dalam sebuah program tiap minggunya. Seharusnya soal pemrograman yang diberikan lebih banyak dan masing-masing memiliki konsep atau materi baru di dalamnya. Akan tetapi materi yang kecil dan terdapat dalam banyak program tersebut akan memerlukan umpan balik yang diperlukan mahasiswa agar dapat mengetahui apakah jawaban sudah benar atau belum (Colton et al., 2006). Dalam mempelajari materi yang terbagi menjadi jauh lebih banyak, tentu diperlukan juga soal yang jauh lebih banyak yang harus disediakan untuk mahasiswa. Untuk hal itu diperlukan pengelolaan soal, atau gudang soal untuk mengelola soal ini sesuai kategori dan tingkat kesulitannya, sehingga mahasiswa dapat mengerjakan soal pemrograman sesuai kategori dan tingkatan yang jelas. Akan tetapi umpan balik terhadap mahasiswa menjadi masalah yang cukup

berat. Dengan meningkatnya tugas yang harus dikerjakan berarti lebih banyak kode program yang harus diperiksa atau diuji. Oleh karena itu, sebuah program *autograder* perlu dibuat. Program ini akan meng-kompile dan mengeksekusi kode program yang dikumpulkan, kemudian melakukan tes menggunakan masukan yang sudah disiapkan instruktur, lalu hasil keluaran dibandingkan dengan hasil keluaran yang diharapkan. Kemudian secara langsung mahasiswa mendapatkan umpan balik mengenai program yang telah dibuatnya (Nordquist, 2007).

Melakukan penilaian menggunakan *autograder* memberikan keuntungan yaitu penilaian yang dilakukan konsisten, teliti, dan efisien. Setiap program akan dinilai dengan menggunakan standar penilaian yang sama. Selain itu juga akan memberikan umpan balik terhadap mahasiswa mengenai kode program yang dikumpul (Isong, 2001).

Hasil dari penelitian kali ini adalah sebuah sistem pembelajaran pemrograman yang berjalan di lingkungan web. Terdapat dua pengguna utama yaitu *admin* atau pengelola soal dan pengguna atau mahasiswa. Pengelola soal bertugas membuat soal, serta memasukan *testcase* dan jawaban yang diharapkan. Mahasiswa dapat melakukan pemilihan soal, mengerjakan soal, mengumpulkan soal, dan mendapat umpan balik dari jawaban yang dikumpulkannya. Kemudian berdasarkan jawaban yang benar pengguna akan mendapatkan *achievement* berupa *level* yang dapat meningkat seiring dengan bertambahnya jumlah jawaban yang benar.

Pembuatan Web menggunakan bahasa pemrograman PHP, dan menggunakan framework CodeIgniter, basis data yang digunakan adalah MySQL. Selain itu digunakan Fossil grader, program yang akan melakukan eksekusi terhadap program sesuai *queue*, dan beroperasi di sistem operasi Windows.

1.2. Rumusan Masalah

Dari latar belakang tersebut dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana membangun website pembelajaran pemrograman dengan menggunakan *automated grading system*.
2. Bagaimana membangun website pembelajaran pemrograman menggunakan fitur *level* bagi pengguna website.
3. Bagaimana membangun website pembelajaran pemrograman dengan pengelolaan soal berdasarkan kategori dan tingkat kesulitan soal.

1.3. Batasan Masalah

Berikut beberapa batasan dalam pembangunan website ini:

1. Sistem yang akan dikembangkan dapat melakukan penilaian terhadap bahasa C.
2. Soal-soal yang akan digunakan adalah soal pengenalan pemrograman, dan soal competitive programming.

1.4. Tujuan Penelitian

Tujuan pembangunan website ini adalah sebagai berikut:

1. Membangun website pembelajaran pemrograman dengan menggunakan *automated grading system*.

2. membangun website pembelajaran pemrograman menggunakan fitur *level* bagi pengguna website.
3. membangun website pembelajaran pemrograman dengan pengelolaan soal berdasarkan kategori dan tingkat kesulitan soal.



1.5 Metodologi Penelitian

1.5.1 Perangkat Keras dan Perangkat Lunak

Perangkat Keras yang dibutuhkan untuk pembangunan website ini adalah sebuah komputer server dengan menggunakan sistem operasi Windows yang terkoneksi dengan *Internet* dan menggunakan *web server* apache serta basis data yang digunakan adalah *MySql*.

1.5.2 Langkah Penelitian

1. Studi Literatur

Studi Literatur dilakukan dengan melakukan pencarian Journal yang pernah membangun aplikasi sejenis dan website yang menerapkan *automatic grading system*. Journal dan website ini kemudian digunakan sebagai acuan dan pertimbangan pembangunan website nantinya. Selain itu juga dilakukan studi terhadap aplikasi yang diperlukan selama proses pembangunan website, seperti bahasa pemrograman PHP, Fossil Grader, dan Framework CodeIgniter.

2. Pembangunan Perangkat Lunak

A. Analisis

Analisis dilakukan dengan menganalisis data dan informasi yang diperoleh sehingga dapat dijadikan bahan pengembangan perangkat lunak. Hasil analisis adalah berupa model website yang dituliskan dalam dokumen teknis Spesifikasi Kebutuhan Perangkat Lunak (SKPL).

B. Perancangan Perangkat Lunak

Perancangan dilakukan untuk mendapatkan deskripsi arsitektural dari website

pembelajaran pemrograman yang hendak dibangun, deskripsi antarmuka, deskripsi data, dan deskripsi prosedural. Hasil perancangan berupa dokumen Deskripsi Perancangan Perangkat Lunak (DPPL).

C. Implementasi Perangkat Lunak

Implementasi dilakukan perangkat lunak dilakukan berdasarkan SKPL dan DPPL yang telah dibuat. Hasil dari pembangunan perangkat lunak berupa website pembelajaran pemrograman yang menerapkan *automated grading system*.

D. Pengujian Perangkat Lunak

Pengujian akan dilakukan kepada anggota Kelompok Studi Pemrograman Universitas Atma Jaya Yogyakarta, dan juga Mahasiswa baru yang ingin belajar pemrograman. Hasil dari pengujian ini berupa data kuisisioner yang telah diisi. Selain itu juga dilakukan pengujian fungsionalitas yang perangkat lunak yang menghasilkan dokumen Perencanaan Deskripsi dan Hasil Uji Perangkat Lunak (PDHUPL).

1.6 Sistem Penulisan Tugas Akhir

Adapun Sistematika penulisan laporan tugas akhir adalah sebagai berikut :

BAB 1 PENDAHULUAN

BAB 2 TINJAUAN PUSTAKA

BAB 3 LANDASAN TEORI

BAB 4 ANALISIS DAN PERANCANGAN SISTEM

BAB 5 IMPLEMENTASI DAN PENGUJIAN SISTEM

BAB 6 KESIMPULAN DAN SARAN