

## **BAB 2**

### **Tinjauan Pustaka**

*Autograder* atau sering disebut juga *auto grading system*, memungkinkan siswa untuk mengontrol semua aspek dalam penyerahan tugas mereka. Mereka dapat memilih kapan untuk menyerahkan pekerjaan mereka, seberapa sering untuk mengirimkannya, kapan harus berhenti mengirimkan, dan seketika menerima feedback dari tugas tersebut. Dengan metode tradisional panduan pengumpulan tugas dan prosedur penilaian, hanya waktu pengumpulan tugas yang berada di bawah kontrol siswa. *Autograder* membantu mengajarkan mahasiswa untuk memastikan kesesuaian tugas dengan persyaratan soal. Teknik ini memiliki efek samping yaitu ambiguitas persyaratan tugas mengakibatkan kesalahan jawaban. Untuk memastikan *Autograder* bekerja dengan baik, instruktur harus memastikan solusi untuk tugas sudah sesuai dengan soal yang diberikan. Penulis telah menemukan lebih dari satu kesalahan dalam solusi tugas yang ditulis sendiri oleh instruktur. Pada akhirnya, sebagian besar karir seorang programmer akan dihabiskan membaca kode orang lain, ini merupakan keterampilan yang tidak mudah untuk dipelajari. Ketika siswa menggunakan *autograder*, mereka akan sering memeriksa kode mereka sendiri untuk menentukan bagaimana membenarkan kode mereka. Mereka sangat termotivasi untuk membaca, memahami, dan bahkan memodifikasi kode program mereka (Nordquist, 2007).

Selama ini sudah terdapat beberapa penelitian yang melakukan pembuatan *automated grading system* untuk tugas pemrograman. Nordquist (2007) pernah melakukan penelitian serupa untuk melakukan penilaian secara

otomatis terhadap tugas pemrograman yang menggunakan bahasa pemrograman java. Hanya saja arsitektur sistem ini kurang praktis, user harus mengunduh aplikasi *grader* dari website tersebut, lalu melakukan penilaian menggunakan aplikasi tersebut. Setelah menilai, sistem akan mengirim data ke server. Hal ini tentu merepotkan pengguna karena harus mengunduh dahulu, selain itu juga lebih memakan waktu apalagi jika koneksi internet terbatas. Sistem penilaian yang ingin penulis kembangkan adalah dengan menyediakan sebuah program yang akan menilai secara otomatis dari pihak server. Dengan demikian *user*, tidak perlu lagi mengunduh aplikasi *grader* dari website, sehingga pengguna dapat berfokus pada pengerjaan soal, dan tidak terganggu dengan proses yang panjang.

Colton et al., (2006) membangun *Autograder* berbasis web dan mampu menilai beberapa bahasa pemrograman seperti C, C++, Java, Perl, MIPS. Hanya saja, fokus pembuatan website ini hanya pada fungsionalitasnya saja, tampilan website yang dibuat masih terlalu sederhana dan kurang interaktif. Pembuatan Website pembelajaran pemrograman tentu membuat pengguna akan jenuh. Oleh karena itu, sisi tampilan website juga harus diperhatikan. Dengan menggunakan tampilan yang lebih menarik akan membuat pengguna lebih bersemangat untuk mengerjakan soal pemrograman.

Cheang et al (2003) membangun website belajar pemrograman dikhususkan untuk belajar bahasa C++, penilaian sudah otomatis, tetapi tidak memiliki pembagian kategori soal yang jelas.

Harris et al (2004) membangun website untuk pengumpulan tugas pemrograman, dengan memberikan aturan-aturan seperti pemberian penalti ketika siswa salah ketika mengumpul kode programnya. Website ini dirancang untuk pengumpulan tugas, sehingga tidak ada pembagian soal untuk belajar para siswa.

Alhami & Alsmadi (2011) membuat tool penilaian yang berjalan di desktop dengan menerapkan penilaian otomatis, namun belum ada fitur tambahan seperti *achievement* dan *level*. Tampilan pun masih sederhana dan mendukung bahasa pemrograman C++.

Zanden et al. (2012) membangun aplikasi untuk mengumpulkan tugas pemrograman, dengan bahasa pemrograman C++ saja, dan dikhususkan untuk pengumpulan tugas, dan tidak ada bundel soal, atau pembagian soalnya. Sudah berjalan di website dengan kurang memperhatikan faktor GUI atau tampilan websitenya.

Roberts & Verbyla (2003) membuat website untuk soal pemrograman. Website dapat menguji bahasa pemrograman java. selain itu website ini juga terdapat soal pilihan ganda. Di website yang penulis kembangkan lebih ke arah latihan pemrograman, sehingga tidak ada model soal pilihan ganda.

Rosanti,Vita (2008) pernah membuat website untuk kompetisi pemrograman. Website ini khusus untuk kompetisi, mampu melakukan kompilasi bahasa pemrograman c,java,dan pascal. Akan tetapi, karena khusus untuk kompetisi, website ini tentu saja tidak dapat digunakan untuk berlatih bagi programmer awal.

Online judge adalah sebuah sistem online untuk menguji program dalam kontes pemrograman. Online Judge

juga digunakan untuk berlatih kontes pemrograman. Banyak sistem Online Judge yang menyelenggarakan kontes mereka sendiri.

Sistem ini dapat mengkompilasi dan mengeksekusi kode, dan menguji mereka dengan data pra-dibangun. Kode yang dikumpul dapat dijalankan dengan pembatasan, termasuk batas waktu, batas memori, pembatasan keamanan dan sebagainya. Output dari kode akan ditangkap oleh sistem, dan dibandingkan dengan output yang diharapkan. Sistem kemudian akan mengembalikan hasil. Ketika kesalahan yang ditemukan dalam output, penilaian dilakukan menggunakan metode yang harus dibuat.

Beberapa Online Judge yang sering digunakan antara lain UVA online Judge dan Sphere Online Judge. Kedua website ini merupakan website Online judge yang paling populer digunakan untuk latihan pemrograman. UVA Online judge memiliki beberapa fitur antara lain: soal yang relatif banyak, fitur cari soal berdasar kode soal atau nama soal, statistik 50 pengumpul terakhir, statistik penjawab soal terbanyak, dan statistik tahunan (UVA, 2014). UVA Online Judge sangat unggul dalam fungsi utamanya yaitu Online Judge, dengan jumlah soal yang sangat banyak. Hanya saja kelemahan dari website ini adalah tidak ada pembagian soal berdasarkan kategori, sehingga sangat sulit menentukan soal mana yang harus dikerjakan terlebih dahulu, atau yang harus dipelajari lebih dahulu. UVA sering digunakan sebagai situs untuk belajar competitive programming, akan tetapi untuk menentukan soal mana yang dikerjakan biasanya pengguna menggunakan acuan lain seperti buku atau situs lain yang telah memilah-milah soal dalam UVA Online Judge.

Spehere Online Judge sama seperti UVA online judge, hanya saja tampilan sedikit lebih rapi. Untuk konten soal termasuk lengkap dan variatif, akan tetapi kelemahannya sama yaitu pembagian kategori soal. Terdapat pembagian soal, akan tetapi tidak dibagi sesuai bab-bab tertentu. SPOJ membagi soal ke dalam kategori classical dan challenges. Pada kategori challenges belum dibagi sesuai kategori soal yang lebih spesifik, seperti graph, dynamic programming, dan lain-lain. sampai saat ini SPOJ memiliki jumlah submission atau pengumpulan jawaban soal pemrograman hingga 11 juta lebih, ini termasuk angka yang cukup besar untuk website online judge. SPOJ menyediakan 48 bahasa pemrograman yang dapat ia kompilasi.

Dari beberapa projek yang sudah pernah dibuat tersebut, permasalahan yang sama dari semua projek tersebut adalah pengelolaan soal belum dibagi sesuai kategori. Hal ini akan menyulitkan siswa yang ingin belajar pemrograman, khususnya bagi siswa yang masih pemula di bidang pemrograman. Mereka tidak tahu kemana harus mulai mengerjakan soal, mereka membaca soal, lalu apabila mereka belum sanggup mengerjakan, atau tidak paham mengenai subtopik pemrograman yang diperlukan, mereka terpaksa mencari soal lain yang dapat mereka kerjakan. Bagi mereka yang aktif dalam website *online judge*, mereka menggunakan pihak ketiga seperti buku atau website lain, untuk menentukan soal mana yang akan dikerjakan. Masalah ini dapat diselesaikan yaitu apabila, web latihan pemrograman tersebut sudah membagi soalnya ke dalam beberapa kategori. Dengan demikian kejadian membaca soal berkali-kali, dan penggunaan

pihak ketiga sebagai panduan pengerjaan soal pemrograman menjadi tidak diperlukan.

Masalah lain yang umum adalah masalah GUI (*Graphical User Interface*) atau tampilan website itu sendiri. Beberapa website pemrograman yang sudah dikembangkan kurang menarik. Membuat pengguna website tersebut jenuh ketika melihat tampilan yang kurang teratur, bahkan beberapa *link* ada yang mati atau tidak dapat diakses. Selain itu, website yang dibangun juga murni untuk mengerjakan pemrograman, belum terdapat *reward* atau penghargaan bagi mereka yang berhasil menuntaskan soal pemrograman. Padahal dengan adanya sistem penghargaan seperti ini, dapat membuat seseorang lebih tertarik mengerjakan soal pemrograman. Karena setelah puas memecahkan soal, mereka juga mendapat penghargaan atau *reward*, yang dapat mereka *share*, atau mereka bagikan pada teman-teman mereka untuk memotivasi. Sehingga harapannya dapat meningkatkan semangat untuk bersaing dalam mengerjakan atau memecahkan soal pemrograman di website tersebut.

Perbandingan Website yang ingin penulis kembangkan dengan website sejenis yang pernah dibuat sebelumnya :

Fitur	Penulis	Nordquist	Colton	Cheang	Harris	Alhami	Zanden	Roberts	UVA	SPOJ	VITA
Web-based	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
automatic grading system	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
penilaian di sisi server	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
pembagian jenis soal	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
pemberian level	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗