

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Pengajaran mata kuliah pemrograman biasanya diikuti oleh banyak mahasiswa. Dengan semakin besarnya jumlah mahasiswa dan banyaknya tugas yang harus diperiksa, pemeriksaan tugas *source code* manual menjadi sangat menghabiskan waktu dan tenaga, kadang melebihi yang dihabiskan untuk interaksi langsung. Otomasi penilaian atau *assessment* terhadap tugas *source code* telah menjadi suatu kebutuhan penting. Oleh karena itu pembuatan aplikasi *autograder* dibutuhkan untuk meringankan tugas pemeriksaan *source code* secara manual sehingga bisa membuat waktu yang terpakai menjadi efektif dan tenaga yang dipakai pun tidak terlalu besar. *Autograder* adalah suatu aplikasi atau modul program untuk melakukan penilaian secara otomatis.

Aplikasi *autograder* yang akan dibuat memiliki keunikan dibandingkan aplikasi-aplikasi *autograder* yang sudah ada misalnya UVA online judge dan aplikasi *autograder* milik Olympia. Karena hampir semua *autograder* tidak bisa mentolerir kesalahan dan hanya memberikan *output* berupa *true* atau *false*. Selain itu banyak yang melakukan kecurangan jika *autograder* tersebut berupa web, misalnya duplikasi kode. Untuk mengatasi masalah-masalah di atas maka perlu dibangun *autograder* yang lebih efektif. Aplikasi yang akan dibuat dapat mentolerir kesalahan *output* yang keluar, misalnya pada suatu *output* terdapat kelebihan *space*, *enter* atau *whitespace* maka sebenarnya program tersebut tidak sepenuhnya salah, atau mungkin hanya terdapat salah ejaan pada satu hurufnya dan sistem penilaiannya pun akan bervariasi tidak hanya memberikan keluaran *true* atau *false* saja tetapi bisa memberikan nilai yang dinamis. Contohnya jika terdapat kelebihan spasi maka program tersebut tidak sepenuhnya salah, tetapi jika kelebihan sebuah angka maka itu merupakan kesalahan yang fatal dan tentu saja nilainya akan jauh berbeda dari yang hanya kelebihan tanda spasi.

Aplikasi yang nantinya akan dibangun akan berbentuk desktop. Dan akan dibedakan menjadi 2 yaitu aplikasi untuk admin dan untuk peserta didik. Hal ini dibuat agar *autograder* dapat digunakan secara langsung di kelas. Peserta juga dapat mengumpulkan *file* yang mereka kerjakan dengan aplikasi tersebut. Aplikasi pada admin dapat menerima *file* tersebut dan langsung melakukan penilaian secara otomatis. Jadi pekerjaan pengajar dalam mengumpulkan tugas dan memberikan nilai dapat terselesaikan secara efektif dan pastinya lebih cepat. Penilaian dilakukan dengan membandingkan kesamaan *output* dari *input* yang diberikan. Perbandingan dilakukan dengan algoritma *string matching*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, maka dapat diambil kesimpulan beberapa rumusan masalah sebagai berikut:

1. Bagaimana cara mendeteksi plagiarisme dari kode sumber yang dikumpulkan dan menghindari peserta melakukan tindak plagiarisme?
2. Bagaimana cara memberikan penilaian yang objektif dan optimal terhadap kode sumber peserta didik?
3. Hasil dari sebuah jawaban ada memang harus memiliki jawaban yang 100% tepat, tetapi bagaimana cara agar dapat memberikan penilaian yang lebih tepat selain benar dan salah?

1.3 Tujuan Pembahasan

Berdasarkan rumusan masalah yang telah terurai di atas, maka dapat dijelaskan tujuan pembuatan aplikasi ini sebagai berikut :

1. Menggunakan Algoritma *Rabin-Karp* untuk mendeteksi kesamaan dari beberapa kode sumber. Aplikasi yang dibuat juga berbentuk desktop sehingga kode sumber yang diterima hanya akan berasal dari satu jaringan yang sama. Kode sumber akan dibentuk kembali menjadi bahasa yang lebih rendah, Java akan dirubah menjadi bentuk *bytecode*,

sedangkan C dan Python akan dirubah menjadi bentuk *assembly* sehingga dapat mengatasi masalah semantic.

2. Cara yang dipakai untuk melakukan penilaian adalah dengan menggunakan algoritma *Bruteforce* untuk melakukan pencocokan *string* dari *output* yang dikeluarkan program dengan *output* dari sumber kode jawaban.
3. Dengan cara memberikan *multiple testcase* sehingga jika ada testcase yang benar kode sumber tersebut masih bisa diberikan nilai yang sesuai.

1.4 Ruang Lingkup Kajian

Berikut ini akan dijabarkan ruang lingkup kajian yang akan dibahas dalam pembuatan aplikasi ini :

1. Aplikasi akan berbentuk software desktop.
2. Aplikasi hanya dapat menerima dan memberikan nilai kepada user yang ada dalam 1 jaringan saja.
3. Aplikasi akan menggunakan algoritma *Bruteforce* untuk melakukan *String matching* dari *outputnya*.
4. Penilaian hanya dapat diberikan melalui 2 jenis penilaian saja yaitu penilaian yang dapat memberikan toleransi kesalahan dan penilaian yang tidak dapat mentoleransi kesalahan
5. Aplikasi hanya dapat menerima kode sumber yang berupa kode sumber dari bahasa pemrograman Java, C, dan Python.
6. Kode sumber harus berada dalam 1 file
7. Untuk melakukan perbandingan plagiarisme semua *file* ingin diperiksa diharuskan berada di dalam 1 direktori folder yang sama.
8. Untuk melakukan perbandingan plagiarisme semua *file* ingin diperiksa diharuskan memiliki nama awalan yang sama, contohnya : "T_1.java, T_2.java, P1.py, P2.py.

1.5 Sumber Data

Sumber data yang digunakan dalam pembuatan laporan tugas akhir ini adalah sebagai berikut :

1. Sumber data primer diperoleh dengan cara observasi, serta percobaan langsung dengan pengguna.
2. Sumber data sekunder diperoleh dengan berupa buku teks, situs-situs di Internet, serta referensi ilmiah lain yang memuat materi yang berhubungan dengan penelitian.

1.6 Sistematika Penyajian

Sistematika pembahasan dari penyusunan laporan tugas akhir ini direncanakan sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat tentang latar belakang, rumusan masalah, tujuan, ruang lingkup kajian, serta sistematika pembahasan dari tugas akhir ini.

BAB II LANDASAN TEORI

Bab ini membahas mengenai teori-teori yang berkaitan dalam penyelesaian proyek tugas akhir.

BAB III ANALISIS DAN DESAIN SISTEM

Bab ini membahas analisis dan desain sistem yang akan dibangun pada proyek tugas akhir.

BAB IV PENGEMBANGAN PERANGKAT LUNAK

Bab ini berisi kumpulan screenshot dari proyek yang dibuat beserta deskripsi dari tiap fitur yang dibuat.

BAB V TESTING DAN EVALUASI SISTEM

Bab ini berisi hasil pengujian dan evaluasi dari sistem yang telah dibangun.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran dari pembahasan pada perancangan serta analisa pengujian aplikasi yang dibuat untuk perkembangan aplikasi.

