

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu sub keahlian dalam bidang ilmu komputer adalah pemrograman. Pemrograman berkata dasar “program”, yang menurut KBBI memiliki arti “program yang diciptakan secara khusus sehingga memungkinkan komputer melakukan fungsi tertentu”. Pemrograman adalah kegiatan yang menciptakan suatu program komputer. Dalam rekayasa perangkat lunak (RPL), pemrogram, yang lebih akrab dipanggil *programmer*, harus memiliki keahlian yang diperlukan dalam pembuatan program. Keahlian tersebut melingkupi namun tidak dibatasi algoritma dan dokumentasi program. Ilmu algoritma dan pemrograman dalam program studi ilmu komputer, teknik informatika, dan sejenisnya sangat penting untuk mendidik *programmer*. Orang-orang yang mempelajarinya juga harus berlatih terus-menerus agar menguasai sepenuhnya dan bisa mengaplikasikan ilmu tersebut dengan baik (Austin dkk., 2018; Wang, 2019).

Berhubungan dengan dokumentasi, sebuah keahlian dari pemrograman yang perlu dipahami, dikuasai, dan diaplikasikan dengan baik adalah penulisan komentar dalam *source code* perangkat lunak yang dibuat. Komentar sering dimanfaatkan untuk dokumentasi dan membantu pengembang perangkat lunak untuk memahami *source code* tersebut. Program dengan *source code* yang dikomentari dengan jelas dapat dipahami dengan jauh lebih mudah dibanding tanpa komentar (Tenny, 1988; Woodfield dkk., 1981). Dokumentasi yang kurang baik dapat menyebabkan kesalahpahaman (Buse & Weimer, 2010) dan dapat menyulitkan perawatan dan perkembangan lanjut perangkat lunak (Lientz, 1983). Walau begitu, memberi komentar pada *source code* sering diabaikan karena *deadline* atau batasan waktu lainnya (Steidl dkk., 2013). Ditemukan juga bahwa proyek *open source* yang sukses memiliki jumlah komentar yang padat dan konsisten dalam *source code* mereka (Arafat & Riehle, 2009; Arafati & Riehle, 2009).

Dalam perkuliahan Algoritma dan Pemrograman di Prodi Ilmu Komputer Universitas Pendidikan Indonesia (UPI), dosen pengampu memanfaatkan sebuah Sistem *Online Judge* bernama *Computer Science Programming Contest (CSPC)*

(Sukanto, 2011). *Online Judge* adalah sebuah program penilai *source code* program otomatis. Sistem tersebut sangat penting untuk membantu proses latihan dan tahap evaluasi pembelajaran pemrograman para mahasiswa. CSPC juga sangat membantu dosen dan asisten dosen untuk menilai program hasil latihan mahasiswa. Namun, selama ini, CSPC hanya menilai kebenaran *output* dari program yang dibuat. Dalam menilai komentar *source code*, dosen dan asisten dosen membaca komentar setiap *source code* dan menilai secara manual. Hal ini tidak efisien serta juga rawan hal-hal seperti bias dan ketidakkonsistenan dari setiap penilai (Kurnia dkk., 2001). Selain itu, terdapat masalah jumlah komentar yang harus dinilai bila dilakukan secara manual. Menurut dosen pengampu, dalam mata kuliah Algoritma dan Pemrograman I, evaluasi dilakukan lima kali di mana masing-masing evaluasi memiliki dua soal. Bila mahasiswa yang mengontrak ada 90 orang, maka penilai harus memeriksa 900 komentar *source code*. Penting bagi *programmer* untuk membiasakan diri menulis komentar dalam *source code* yang jelas dan deskriptif untuk menjelaskan program yang dibuat. Maka, akan sangat membantu apabila *Online Judge* yang ada juga dapat menilai komentar *source code*.

Sistem penilaian komentar *source code* bukanlah teknologi baru. Sudah terdapat sejumlah aplikasi *text mining* untuk menilai kualitas *source code* dalam satu dekade ke belakang (Khamis dkk., 2010; Steidl dkk., 2013; Tan dkk., 2012). Namun, kebanyakan aplikasi ini dirancang untuk lingkup profesional, di mana penilaian komentar dilakukan berdasarkan kualitasnya sebagai sebuah bentuk dokumentasi. Kenyataannya, komentar yang dibuat seseorang yang baru mulai mempelajari pemrograman tidak bisa disamakan dengan komentar yang dibuat *programmer* yang lebih berpengalaman. Menurut dosen yang mengampu mata kuliah Algoritma dan Pemrograman di Prodi Ilmu Komputer UPI, seorang pelajar yang baru memulai pemrograman belum memahami pentingnya komentar *source code* sehingga hanya membuat sesuai yang mereka pahami, sedangkan seorang *programmer* profesional memang dituntut untuk membuat dokumentasi dari program yang ia tulis dengan format dan standar tertentu.

Dalam ranah *text mining*, pengukuran kemiripan teks atau *text similarity measures* adalah fungsi yang menghitung seberapa mirip dua buah teks. Kemiripan dapat diukur dari sisi leksikal (bentuk) dan semantik (makna) (Wael H. Gomaa &

Fahmy, 2013). Dalam bidang pendidikan, *text similarity measures* sering digunakan sebagai pendekatan untuk berbagai proses otomatis. Salah satu proses ini yang populer adalah penilaian esai otomatis atau *automatic essay scoring* (AES). AES umumnya dilakukan dengan membandingkan kemiripan jawaban peserta dengan soal atau teks yang diberikan dengan berbagai macam metode. Salah satu metode yang dapat digunakan adalah *word embedding* (Cozma dkk., 2018; Wael Hassan Gomaa & Fahmy, 2020; Jin dkk., 2017). *Word embedding* dalam kasus ini digunakan sebagai basis perbandingan antara jawaban peserta dan soal.

Berangkat dari premis itu, penelitian dalam skripsi ini bertujuan untuk merancang proses penilaian komentar *source code* secara otomatis dalam sistem *online judge* dengan menggunakan *word embedding* untuk melakukan perhitungan *text similarity* atau kemiripan teks. Penelitian ini merupakan kelanjutan dari penelitian serupa oleh Sukamto dkk. (2019). Penelitian tersebut bertujuan merancang fitur penilaian komentar otomatis untuk sistem *online judge*. Nilai didapatkan dengan menghitung kepadatan komentar serta kemiripan komentar dengan soal. Kepadatan komentar dihitung dengan membandingkan jumlah baris komentar dengan jumlah baris *source code*. Kemiripan teks dihitung menggunakan pendekatan *string-based similarity* dengan dua metode statistik serta metode *Longest Common Subsequence* (LCS). Hasil penelitian tersebut menyatakan bahwa fitur yang dirancang masih membutuhkan kriteria tambahan, khususnya kemampuan untuk mengukur koherensi antara komentar dan soal (Sukamto dkk., 2019). Penelitian ini bertujuan untuk memenuhi kriteria tersebut dengan menggunakan pendekatan *text similarity* dengan *word embedding*. Pendekatan ini diharapkan dapat menangkap makna semantik dengan lebih baik untuk membantu perhitungan kemiripan komentar dan soal.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah penelitian ini adalah: “Bagaimana kinerja dan hasil metode *word embedding* dengan perhitungan *Word Mover’s Distance* (WMD) untuk menghitung *text similarity* dalam kasus penilaian komentar otomatis dalam *online judge*?”

1.3 Tujuan

Penelitian yang diusulkan bertujuan menguji coba performa *text similarity* dengan *word embedding* dan *Word Mover's Distance* untuk fitur penilaian komentar otomatis untuk sistem *online judge*. Penelitian ini melakukan perhitungan *text similarity* dengan *word embedding* untuk menghitung relevansi antara komentar yang dibuat para peserta dengan soal.

1.4 Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut.

- 1) *Source code* yang diproses berbahasa pemrograman C.
- 2) Penilaian komentar hanya dilakukan pada *source code* program prosedural satu berkas (hanya *source code* “*main.c*”).
- 3) Komentar yang dinilai hanya komentar berbahasa Indonesia.
- 4) Metode diimplementasikan menggunakan bahasa pemrograman Python.

1.5 Struktur Penulisan

Untuk memahami lebih jelas penelitian ini, maka materi-materi pada skripsi ini dikelompokkan ke dalam beberapa sub bab dengan sistematika penyampaian sebagai berikut.

Bab I Pendahuluan

Bab ini menguraikan latar belakang beserta maksud dari penelitian ini, rumusan masalah yang ingin diselesaikan, tujuan dari penelitian ini, serta batasan-batasan masalah. Penguraian latar belakang membahas pentingnya bagi *programmer* untuk terbiasa membuat dokumentasi *inline* berupa komentar dalam *source code*, bagaimana program studi Ilmu Komputer UPI mengevaluasi komentar yang dibuat mahasiswa-mahasiswi, hingga mengapa diperlukannya proses penilaian otomatis untuk komentar tersebut.

Bab II Kajian Teori

Bab ini berisi definisi tentang materi-materi atau teori-teori—dikutip dari buku dan jurnal nasional maupun internasional—yang berhubungan dengan penelitian ini. Materi dan teori yang dibahas di bab ini adalah komentar dalam *source code*, sistem *online judge*, AES, *word embeddings*, *text similarity*, *Word Mover Distance* (WMD), koefisien korelasi, serta penelitian terkait.

Bab III Metodologi Penelitian

Bab ini menjelaskan tahap-tahap yang dilakukan dalam penelitian ini. Tahap-tahap tersebut diawali dari studi literatur dan diakhiri penarikan kesimpulan serta dokumentasi hasil. Bab ini juga menjelaskan rencana metode yang akan diimplementasikan untuk memecahkan masalah penelitian ini.

Bab IV Implementasi dan Pembahasan

Bab ini akan memaparkan hasil implementasi metode beserta hasil eksperimen yang dilakukan.

Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan dari penelitian yang telah dilaksanakan beserta saran untuk penelitian-penelitian terkait berikutnya.