

1. Pendahuluan

1.1 Latar Belakang

Salah satu tahap dalam proses seleksi karyawan adalah wawancara kerja yang harus dilalui oleh pelamar kerja sebelum mendapatkan pekerjaan. Wawancara kerja merupakan tahap yang sangat penting karena dalam tahap ini pewawancara dapat menilai kepriadian pelamar kerja. Tujuan wawancara kerja adalah untuk menilai aspek psikologis, perilaku, kepemimpinan, komitmen, dan berbagai aspek lain yang masuk dalam penilaian perusahaan. Sehingga pelamar kerja yang lolos tahap wawancara diharapkan memiliki nilai-nilai yang sejalan dengan budaya kerja perusahaan[1].

PT. Telkom merupakan salah satu perusahaan telekomunikasi terbesar di Indonesia yang rutin mengadakan rekrutmen karyawan setiap tahun. Setiap kali PT. Telkom melakukan rekrutmen karyawan, pelamar kerja yang mendaftar mencapai puluhan ribu orang. Untuk mendapatkan karyawan terbaik yang memiliki nilai-nilai yang sesuai dengan budaya kerja PT. Telkom, maka harus dilakukan seleksi berdasarkan kesesuaian karakter pelamar dengan budaya kerja PT. Telkom. Namun, untuk dapat melakukan seleksi terhadap puluhan ribu orang diperlukan banyak sekali tenaga ahli. Sehingga mengakibatkan biaya yang sangat banyak dan durasi waktu seleksi yang sangat lama.

Untuk mengatasi besarnya biaya yang harus dikeluarkan untuk menyewa tenaga ahli dan mempersingkat waktu seleksi, diperlukan sebuah sistem yang dapat memberikan rekomendasi kelulusan pelamar kerja. Dengan menganalisis esai yang diinput oleh pelamar, sistem kemudian dapat mengklasifikasi kesesuaian karakter pelamar dengan budaya kerja perusahaan. Dari hasil klasifikasi tersebut, sistem dapat memberikan rekomendasi apakah pelamar tersebut lolos atau tidak.

Sejak beberapa tahun terakhir, klasifikasi teks telah dipelajari secara luas dan terlihat kemajuan pesat di bidang ini, termasuk pendekatan *Machine Learning* seperti pengklasifikasi Bayesian, *Decision Tree* dan *Random Forest*, *k-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), Jaringan Saraf Tiruan, dan Rocchio[2]. Algoritma-algoritma tersebut merupakan algoritma yang sangat populer karena menghasilkan akurasi yang sangat baik. Sehingga penelitian ini akan difokuskan pada algoritma lain yang lebih jarang digunakan.

Algoritma klasifikasi berbasis *tree*, seperti *Decision Tree*, merupakan algoritma yang memiliki kelebihan dalam hal kesederhanaan, interpretabilitas, dan kemampuan untuk menangani interaksi fitur[3]. Namun, algoritma ini sangat sensitif terhadap gangguan kecil dalam data, dan dapat dengan mudah *overfit*[4]. Model ini juga memiliki masalah dengan prediksi *out-of-sample*[5]. Menurut Suyanto[6], untuk mengatasi kelemahan dari algoritma *Decision Tree*, teknik *Swarm Intelligence* (SI) dapat digunakan untuk mengoptimasi algoritma tersebut. Oleh karena itu, penelitian ini berfokus pada pengujian dan perbandingan algoritma *Decision Tree* tanpa *Swarm Intelligence* dan algoritma *Decision Tree* yang dioptimasi dengan *Swarm Intelligence*. Algoritma *Swarm Intelligence* yang digunakan pada penelitian ini adalah *Particle Swarm Optimization* karena konsep yang sederhana dan implementasi yang mudah[7].

1.2 Topik dan Batasannya

Berdasarkan masalah yang telah diuraikan di atas, didapatkan beberapa rumusan masalah, yaitu:

1. Bagaimana performansi sistem yang dihasilkan dengan algoritma *Decision Tree* yang dioptimasi menggunakan *Particle Swarm Optimization*?
2. Bagaimana perbandingan akurasi algoritma *Decision Tree* tanpa *Particle Swarm Optimization* dengan algoritma *Decision Tree* yang dioptimasi menggunakan *Particle Swarm Optimization*?

Beberapa batasan masalah dari penelitian ini adalah:

1. Tahapan seleksi yang dibahas hanya tahap wawancara.
2. Algoritma yang dibahas pada penelitian ini hanya *Decision Tree* dan *Particle Swarm Optimization*.

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk menganalisis performansi sistem yang dihasilkan dengan algoritma *Decision Tree* yang dioptimasi menggunakan *Particle Swarm Optimization*;
2. Untuk memperoleh perbandingan akurasi algoritma *Decision Tree* tanpa *Particle Swarm Optimization* dengan algoritma *Decision Tree* yang dioptimasi menggunakan *Particle Swarm Optimization*;