

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam konteks pengembangan perangkat lunak, estimasi usaha merupakan suatu aktivitas guna memperkirakan nilai usaha yang diperlukan oleh para pekerja untuk menyelesaikan suatu proyek pengembangan perangkat lunak [1]. Proses estimasi sangat penting karena dapat digunakan sebagai acuan perhitungan dan pengukuran perangkat lunak yang dihasilkan [2]. Salah satu metode yang biasa digunakan untuk menghitung estimasi usaha pengembangan perangkat lunak adalah *use case point* (UCP).

UCP merupakan metode estimasi usaha yang digunakan dalam analisis berorientasi objek yang dapat digunakan untuk menangkap dan menggambarkan persyaratan fungsional dari suatu sistem [3]. Dalam UCP, *productivity factor* (PF) menjadi salah satu penggerak utama pada saat proses perhitungan estimasi usaha [4], terutama ketika *dataset* historis tidak tersedia [5]. UCP mengestimasi usaha dengan menghitung ukuran proyek perangkat lunak dengan PF, dimana PF mencerminkan tingkatan produktivitas sebuah tim untuk menyelesaikan sebuah proyek.

Menurut [4], perhitungan nilai PF dapat dilakukan dengan beberapa cara, diantaranya: nilai tetap, tiga tingkatan PF, regresi linear, dan *machine learning*. Perhitungan UCP menggunakan nilai tetap atau tiga tingkatan PF dilakukan ketika tidak adanya *dataset* historis yang tersedia. Sebaliknya, ketika *dataset* historis tersedia maka metode regresi dan *machine learning* dapat digunakan untuk membuat model prediksi untuk memprediksi *productivity factor* [4].

Algoritma *Support Vector Machine* (SVM) merupakan salah satu algoritma *machine learning* yang dikenal dan dapat digunakan terkait permasalahan estimasi dan klasifikasi. SVM cocok digunakan dalam berbagai permasalahan dan implementasi, seperti pemrosesan gambar, diagnosis penyakit, hingga klasifikasi gelombang. Hanya saja performa SVM sangat bergantung kepada kualitas pengaturan dari nilai *hyperparameter*, *penalty parameter*, dan fungsi kernel. Pengaturan *hyperparameter* yang tepat akan meningkatkan performa prediksi dan estimasi dikarenakan pengaturan yang benar akan menghasilkan *hyperplane* yang optimal [4].

Untuk mengatasi permasalahan *hyperparameter* yang kurang optimal, maka diperlukan algoritma metaheuristik yang terbukti dapat digunakan untuk menyelesaikan hampir semua permasalahan yang berkaitan dengan optimasi dan hanya memerlukan sedikit modifikasi untuk penyesuaian kasus yang diselesaikan [6]. Terdapat berbagai jenis algoritma metaheuristik yang dapat digunakan untuk memecahkan permasalahan kasus optimasi *hyperparameter* SVM, diantaranya seperti *Particle Swarm Optimization* (PSO), *Evolutionary Competitive Swarm Optimization* (ECSO), *Grey Wolf Optimizer* (GWO), *Moth Optimization Algorithm* (MOA), *Reptile Search Algorithms* (RSA), *Genetic Algorithm* (GA), dan masih banyak lagi.

Dalam penelitian ini, algoritma yang digunakan adalah algoritma *Particle Swarm Optimization* (PSO). PSO adalah algoritma metaheuristik yang diperkenalkan oleh Kennedy dan Eberhart pertama kali pada tahun 1995. Meskipun relatif baru, algoritma ini efektif dalam melakukan optimasi terhadap sebuah variabel terkait [7]. Algoritma PSO terinspirasi dari konsep *swarm intelligent* (SI) yang meniru perilaku koloni hewan, seperti burung dan serangga, dalam cara kerjanya [6]. Menurut [8] [9], PSO cocok digunakan karena performa

yang konsisten, baik, efektif untuk permasalahan optimasi, terlebih lagi kemudahan pada penerapan kode program beserta daya komputasi yang relatif murah dan cepat.

Pada penelitian [4] dilakukan optimasi pada nilai *hyperparameter* SVM menggunakan beberapa algoritma, algoritma yang digunakan adalah algoritma berjenis *predatory swarm* yang terdiri dari Komodo mlipir algorithm, Grey wolf optimization, dan Reptile search algorithm untuk memprediksi *productivity factor*. Hasilnya memberikan kesimpulan bahwa proses optimasi pada *hyperparameter* SVM menggunakan algoritma *predatory swarm* dapat meningkatkan nilai prediksi estimasi effort pada use case points.

Dengan begitu berdasarkan pemaparan di atas, penelitian ini bertujuan untuk mencari *hyperparameter Support Vector Machine* dengan bantuan algoritma *Particle Swarm Optimization* untuk memprediksi nilai *productivity factor* pada estimasi *use case point* yang optimal.

1.2 Rumusan Masalah

Berdasarkan penulisan latar belakang di atas, dapat diketahui bahwa SVM cocok digunakan dalam berbagai jenis permasalahan estimasi dan klasifikasi. Hanya saja, performa SVM sangat bergantung kepada kualitas pengaturan nilai *hyperparameter*, *penalty parameter*, dan fungsi kernel. Sehingga rumusan masalahnya adalah bagaimana pemilihan *hyperparameter* SVM dengan bantuan PSO dapat meningkatkan akurasi prediksi *productivity factor* pada estimasi *use case point*.

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah di atas, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menentukan setting parameter *Support Vector Machine* (SVM) yang tepat dengan menggunakan algoritma *Particle Swarm Optimization* (PSO).
2. Mengimplementasikan algoritma *Particle Swarm Optimization* (PSO) pada pencarian *hyperparameter Support Vector Machine* (SVM) untuk memprediksi nilai *productivity factor* pada estimasi *use case point*.
3. Mengevaluasi performa *Support Vector Machine* (SVM) dengan bantuan algoritma *Particle Swarm Optimization* (PSO) pada prediksi nilai *productivity factor* untuk estimasi *use case point*.

1.4 Manfaat Penelitian

Adapun manfaat yang dapat diambil dari penelitian ini diantaranya adalah sebagai berikut:

1. Meningkatkan akurasi estimasi usaha pada fase pengembangan perangkat lunak atau biasa dikenal software development life cycle (SDLC).
2. Mengoptimalkan hyperparameter support vector machine (SVM) menggunakan algoritma particle swarm optimization (PSO).
3. Memberikan panduan perhitungan productivity factor yang lebih akurat bagi pengembang perangkat lunak.
4. Memberikan acuan untuk penelitian selanjutnya mengenai prediksi *productivity factor* pada estimasi *use case point*.