

Estimasi Waktu Pengembangan Dengan Simple Function Point Menggunakan Metode Berbasis Ensemble

Eri Hutama¹, Ferry Noorman Setiawan², Luhur Bayuaji³

^{1,2,3}Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, DKI Jakarta, Indonesia

e-mail: eri.hutama@yahoo.com¹, ferry_ns@yahoo.com², luhur.bayuaji@budiluhur.ac.id³

Received : Agustus, 2024

Accepted : September, 2024

Published : December, 2024

Abstract

Software project development heavily relies on time and cost estimation planning to help developers reduce late software delivery, increase customer satisfaction, allow organizations to allocate resources efficiently, reduce costs, and optimize the development process. To address this issue, a quick and efficient prediction of software development time is needed to provide the best prediction results. This study compares the machine learning methods Ada Boost Regressor, Support Vector Regression, Random Forest Regression and Ensemble Method with data taken from a scrum project at an institute. The result of using machine learning based on ensemble methods with the addition of the simple function point (SiFP) feature can provide the best results, with an RMSE value of 0.0714 for the dataset with SiFP and 0.0819 without SiFP, an MSE value of 0.0051 for the dataset with SiFP and 0.0067 without SiFP, an MAE value of 0.0589 for the dataset with SiFP and 0.0674 without SiFP, and an R2 value of 0.7698 for the dataset with SiFP and 0.6964 without SiFP. From these results, it can be concluded that the ensemble machine learning method model can be used to measure development time estimates on scrum projects.

Keywords: Data Mining, Ensemble, Machine Learning, Scrum, Software Estimation

Abstrak

Pengembangan proyek perangkat lunak sangat bergantung pada perencanaan estimasi waktu dan biaya agar membantu pengembang mengurangi terlambatnya delivery software, meningkatkan kepuasan pelanggan, memungkinkan organisasi untuk mengalokasikan sumber daya secara efisien, mengurangi biaya dan mengoptimalkan proses pengembangan. Untuk mengatasi permasalahan tersebut diperlukan prediksi waktu pengembangan perangkat lunak yang cepat dan efisien serta memberikan hasil prediksi terbaik. Penelitian ini membandingkan metode machine learning Ada Boost Regressor, Support Vector Regression, Random Forest Regression dan Ensemble Method dengan data yang diambil dari proyek scrum pada salah satu instansi. Hasil penggunaan machine learning berbasis metode ensemble dengan penambahan fitur simple function point (SiFP) memberikan hasil terbaik dengan nilai RMSE dataset dengan SiFP 0.0714 dan tanpa SiFP 0.0819, nilai MSE dataset dengan SiFP 0.0051 dan tanpa SiFP 0.0067, nilai MAE dataset dengan SiFP 0.0589 dan tanpa SiFP 0.0674 sedangkan nilai R2 dataset dengan SiFP 0.7698 dan tanpa SiFP 0.6964. Dari hasil tersebut disimpulkan model machine learning ensemble methods dengan penambahan fitur SiFP meningkatkan estimasi waktu pengembangan pada proyek scrum.

Kata Kunci: Data Mining, Ensemble, Machine Learning, Scrum, Software Estimation

1. PENDAHULUAN

Pengembangan perangkat lunak merupakan bagian penting yang berkelanjutan di dalam era digitalisasi saat ini. Perangkat lunak modern sering kali melibatkan tim yang terdiri dari berbagai anggota dengan peran dan tanggung jawab yang berbeda, serta memerlukan manajemen proyek yang efektif dan terukur. Rata-rata pengembangan proyek perangkat lunak mengalami *overtime* sekitar 30% [1]. *Software effort estimation* merupakan cara untuk memprediksi sumber daya manusia, waktu dan biaya yang dibutuhkan untuk mengembangkan atau memelihara perangkat lunak. Tujuannya adalah untuk memberikan estimasi yang akurat terkait upaya yang diperlukan agar proyek perangkat lunak berhasil. Dalam beberapa kasus, *software estimation effort* melibatkan penggunaan *machine learning*. Terdapat dua teknik estimasi *machine learning* dalam penerapannya: *solo* dan *ensemble*. Hal yang penting dalam mengendalikan, dan menyelesaikan proyek perangkat lunak yang sukses adalah dalam batas anggaran dan jadwal [2]. Perkiraan yang berlebihan dan perkiraan yang terlalu rendah keduanya merupakan tantangan utama untuk pengembangan perangkat lunak di masa depan, oleh karena itu, ada kebutuhan terus-menerus akan akurasi dalam *software estimation effort*.

Teknik *software estimation expert based* pada pengembangan *software* dengan metode *agile* yang paling banyak digunakan adalah *Story Points*. Pada posisi kedua penggunaan *Function Point* digunakan juga sebagai pengukuran pengembangan perangkat lunak [3]. Evolusi metode *Function Points* telah mengarah pada berbagai metode ukuran berbasis *function point* yang telah di standarisasi ke dalam beberapa standar ISO, yakni *International Function Point Users Group* (IFPUG), *Mark II Function Point Analysis* (MKII), *The Netherlands Software Metrics Association* (NESMA), dan *COmmon Software Measurement International Consortium* (COSMIC). Kemajuan ini memberikan kontribusi yang besar terhadap penggunaan metode ukuran perangkat lunak secara luas. Namun, biaya dan konsumsi waktu dari metode-metode ini masih menjadi

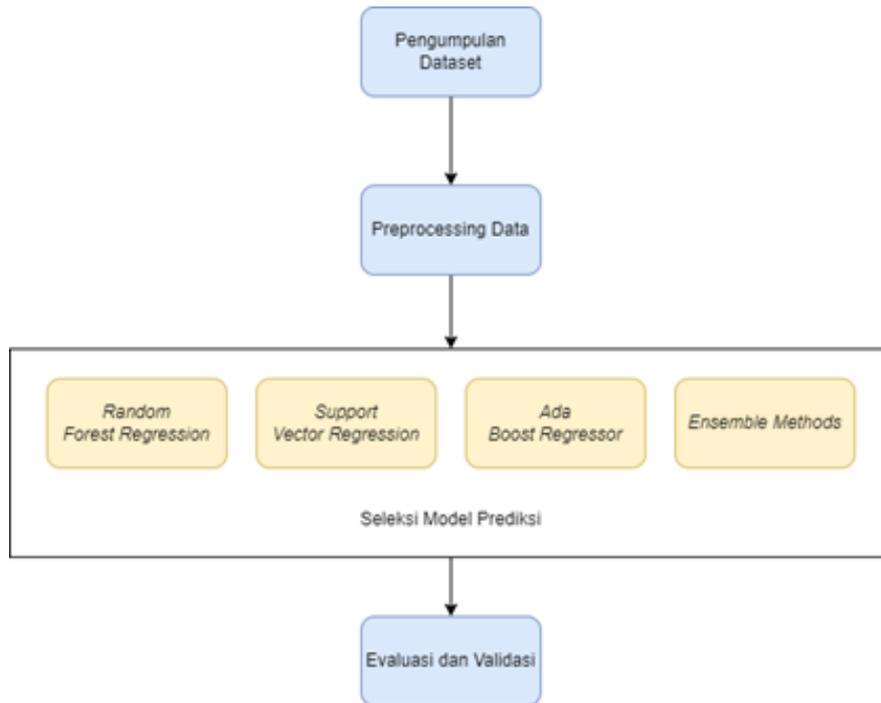
hambatan yang signifikan untuk diterima oleh industri saat ini [4]. Untuk mengatasi masalah ini, *Function Point* yang disederhanakan yang disebut *Simple Function Point* (SiFP) diusulkan pada tahun 2011, yang membutuhkan lebih sedikit waktu dan upaya daripada *International Function Point Users Group Function Point Analysis* (IFPUG FPA) dan sepenuhnya kompatibel [5].

Penggunaan teknik *software estimation expert based* memiliki kekurangan, sehingga digunakan teknik *machine learning* untuk mencapai prediksi yang lebih baik [6]. Teknik *machine learning* yang digunakan untuk memprediksi upaya peningkatan perangkat lunak (*software enhancement effort*) seperti *Machine Learning Regression Methods* (MLRM): *Ada Boost Regressor* (ABR), *Gradient Boosting Regressor* (GBR), *Linear Support Vector Regression* (SVR), dan *Random Forest Regression* (RFR) [7]. Penggunaan teknik *Ensemble Effort Estimation* (EEE) menghasilkan akurasi estimasi yang lebih baik dibandingkan dengan satu teknik *machine learning* [8].

Berdasarkan uraian penjelasan di atas, penelitian ini diharapkan membantu menentukan metode terbaik dari beberapa metode tunggal dan *ensemble machine learning* yang diperbandingkan dengan menambahkan fitur *simple function point* untuk mengukur estimasi waktu pengembangan .

2. METODE PENELITIAN

Jenis penelitian ini adalah analisis deskriptif dan kuantitatif dimana dalam penelitian ini memanfaatkan data kuantitatif dan di jabarkan secara deskriptif. Penelitian ini dilakukan dengan pengumpulan data, dianalisis kemudian diinterpretasikan menggunakan bahasa pemrograman *python 3* dan divisualisasikan dalam bentuk statistik grafik. Tahapan metode penelitian dibagi menjadi tahapan pengumpulan dataset, *preprocessing data*, seleksi model prediksi, serta evaluasi dan validasi [9]



Gambar 1 Tahapan Penelitian [9]

2.1 Pengumpulan Dataset

Penelitian ini menggunakan *dataset* proses pengembangan perangkat lunak yang menggunakan metode *scrum* pada instansi X. Dataset dari instansi X dibagi dengan perbandingan 80:20. Delapan puluh persen digunakan untuk membuat model dengan beberapa *machine learning* dan dua puluh persennya digunakan untuk pengujian model. Penggunaan perbandingan 80:20 didasarkan pada kasus yang membutuhkan data *training* lebih banyak untuk mendapatkan akurasi yang lebih baik [10]. Pada tahap pembuatan dataset dilakukan estimasi pengukuran ukuran fungsional dengan menggunakan perhitungan *Simple Function Point* yang akan digunakan sebagai salah satu atribut independen.

Dataset mencakup data *mandays* yakni yotal waktu yang dibutuhkan untuk menyelesaikan pekerjaan oleh satu orang dalam satu hari kerja, *story points* yakni *Estimasi effort* pada setiap *user story*, *Simple Function Point* yakni ukuran fungsional, prioritas yakni, status prioritas pengembangan, *Unadjusted Use Case Points* (UUCP), jumlah pengembang yakni hitungan pengembang yang bekerja dalam tugas tersebut, pengalaman pengembang yakni Pengalaman pengembang selama bekerja sebagai programmer (tahun) dan waktu aktual pengembangan.

2.2 Preprocessing Data

Pada tahap *preprocessing* data dilakukan penanganan *missing value* dengan pemeriksaan baris data yang duplikat atau memiliki nilai nol dalam kolomnya, lalu data dinormalisasi dengan tujuan mengurangi redundansi data dan memastikan dependensi data [11]. Pada *preprocessing* dilakukan perbaikan dengan mengisi data yang kosong dengan nilai rata-rata sedangkan atribut kategorikal diisi dengan nilai yang sering muncul. Pada atribut-atribut kategorikal dikonversi menjadi atribut nominal [12]. Pada tahap ini juga dilakukan pengecekan data terhadap *outlier* untuk memastikan kualitas data yang baik. Normalisasi data (*feature scaling*) dilakukan dengan metode *MinMaxScaler* atau *StandardScaler* [13].

2.3 Seleksi Model

Pada tahap seleksi model prediksi untuk pelatihan *split data training* dan *testing* menggunakan *k-fold cross validation*, dimana melibatkan pemisahan dataset menjadi sepuluh bagian yang sama ukurannya secara acak [14]. Model dilatih menggunakan 9 subset sebagai data training dan 1 subset yang tersisa digunakan untuk data pengujian kinerja model. Penggunaan *k-fold cross validation* digunakan untuk mengukur kemampuan model untuk

sesuai dengan dataset. Metode ini juga digunakan pada dataset yang jumlahnya terbatas [15]. *Tuning hyperparameter* yang digunakan untuk mendapatkan parameter-parameter terbaik pada model adalah *GridSearch*. Metode *machine learning* yang digunakan adalah algoritma *Ada Boost Regressor* (ABR) *Support Vector Regression* (LinearSVR), *Random Forest Regression* dan *Ensemble Methods*. Prediksi dilakukan dengan membandingkan model yang terbaik untuk digunakan pada dataset yang menggunakan atribut SiFP. Model terbaik disimpan untuk digunakan sebagai pengujian pada dataset proyek *scrum* yang berbeda.

2.4 Evaluasi dan Validasi

Evaluasi yang digunakan untuk mengukur kinerja model yang diusulkan dengan *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), *Mean Squared Error* (MSE) dan *R-squared* (R2) [16]. MAE untuk mengukur rata-rata selisih absolut antara nilai prediksi dan nilai sebenarnya. RMSE untuk mengevaluasi model regresi, merupakan akar kuadrat dari MSE. MSE untuk mengukur rata-rata dari kuadrat selisih antara nilai prediksi yang dihasilkan oleh model dan nilai sebenarnya dari data. R2 untuk mengukur sejauh mana model regresi cocok dengan data. Hasil pengukuran kinerja model akan dibandingkan dengan model lainnya untuk melihat model mana dengan evaluasi metrik dengan nilai terkecil. Validasi dilakukan dengan melakukan pengujian model terbaik dari eksperimen dengan dataset proyek *scrum* yang berbeda.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Dataset

Pada tahapan pengumpulan set data, akan dilakukan perhitungan ukuran fungsional setiap *user story* dengan menghitung nilai dari poin fungsional dalam sub proses. Metode yang digunakan untuk mengukur ukuran fungsional pada set data proyek *scrum* adalah *simple function point*. Pada tahap pengukuran proses pengukuran *simple function point*, setiap *user story* dalam set data proyek *scrum* didetailkan berdasarkan sub proses yang ada. SiFP difokuskan pada elemen-elemen dasar yang dihasilkan pengguna dan kelompok data yang dihasilkan pengguna, dengan tujuan untuk memberikan estimasi yang cepat dan mudah mengenai ukuran sistem perangkat lunak yang kemudian dihitung dengan menjumlahkan elemen fungsi pengguna dan elemen kelompok data yang dihasilkan pengguna. Pada tabel 1 menunjukkan bagaimana cara mendapatkan nilai *simple function point* (SiFP) pada salah satu *user story* berdasarkan deskripsi sub proses yang ada pada *user story* tersebut.

Ukuran aplikasi perangkat lunak – yang dinyatakan sebagai jumlah *Simple Function Point* disesuaikan berdasarkan referensi manual adalah:

$$\text{SiFP} = M (\text{UGEP}) + M (\text{UGDG}) \quad (1)$$

dimana M adalah konstanta untuk nilai proses: 4,6 dan untuk nilai data group: 7, *UGEP* adalah jumlah proses yang ada pada sistem yang diukur, dan *UGDG* adalah jumlah data group yang digunakan pada proses yang diukur [17].

Tabel 1: Menghitung Nilai Simple Function Point (SiFP)

User Story	Deskripsi Proses	User	Deskripsi Subproses	Tipe BFC		SiFP
				UGEP	UGDG	
Fitur <i>Tagging</i> Perjanjian kinerja pada Admin Perangkat Daerah (PD) /Unit Kerja Perangkat Daerah (UKPD)	Admin Perangkat Daerah /Unit Kerja Perangkat Daerah dapat melakukan <i>Tagging</i> indikator Perjanjian Kinerja pada sistem	<i>Admin System</i>	Admin memiliki fitur grup <i>tagging</i> perkin	1 Proses	2 Data	18.6 Poin
		<i>Admin PD</i>	Admin melakukan perencanaan <i>tagging</i> perkin (read, add, edit)	3 Proses	3 Data	34.8 Poin
		<i>Admin PD</i>	Admin dapat melihat <i>tagging</i> yang sudah ditag secara keseluruhan	1 Proses	1 Data	11.6 Poin

Setelah semua deskripsi fungsional proses didetailkan ke dalam sub proses, maka dapat diketahui tipe *Basic Function Component* dan nilai SiFP. Selanjutnya setelah didapatkan nilai SiFP, dapat diketahui jumlah atribut yang digunakan pada dataset. Pada penelitian ini menggunakan 1 jenis dataset yang akan dibagi menjadi data *training* dan data *testing* dengan perbandingan 80:20 dengan detail data *training* sebanyak 110 *sample* dan data *testing* sebanyak 28 *sample*. Berikut adalah dimensi data yang digunakan pada penelitian ini.

Pada tabel 2 menunjukkan jumlah instance dataset instansi X sebanyak 138 dengan jumlah atribut 8 serta target kelasnya aktual waktu pengembangan.

Tabel 2: Dimensi Data

Dataset	Jumlah Instance	Jumlah Atribut	Target
Instansi X	138	8	Actual Dev Time

	UGEP	UGDG	SiFP	Story_Points	Mandays	Priority	Number_of_developers	Experience_of_Developers(Years)	UUCP	Actual_Dev_Time(Days)
count	138	138	138	138	138	138	138	138	138	138
mean	1.8696	1.4058	18.4406	7.7101	1.6783	2.3986	1.029	4.3478	9.4058	1.7062
std	1.5275	1.5122	15.839	6.2505	1.8005	0.8331	0.1684	2.0634	4.5315	1.6738
min	0	0	4.6	1	0.1	1	1	2	0	0.1
25%	1	0	9.2	3	0.5	2	1	2	6	0.5
50%	1	1	13.8	5	1	3	1	6	7	1
75%	2	2	22.6	11.75	2	3	1	6	12	2
max	10	8	88.2	34	10	3	2	8	18	8.5

Gambar 2 Ringkasan Statistik Dataset

Tabel 3: Atribut Data

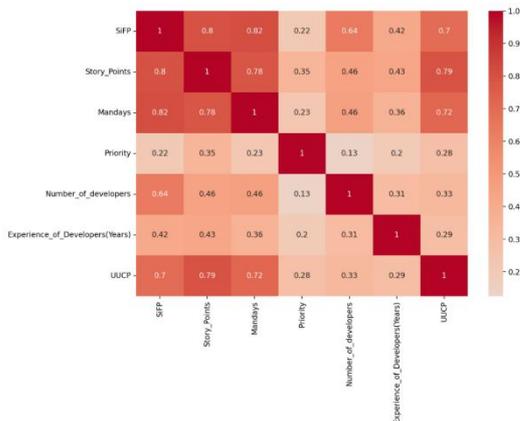
No	Atribut	Deskripsi
1	Mandays	Total waktu yang dibutuhkan untuk menyelesaikan pekerjaan oleh satu orang dalam satu hari kerja
2	Story points	Estimasi effort pada setiap user story
3	SiFP	Ukuran fungsional menggunakan <i>simple function point</i>
4	Prioritas	Prioritas pengembangan
5	UUCP	<i>Unadjusted Use Case Point</i>
6	Jumlah Pengembang	Hitungan pengembang yang bekerja dalam tugas tersebut
7	Pengalaman Pengembang	Pengalaman pengembang selama bekerja sebagai programmer (tahun)
8	Actual Dev Time	waktu aktual (sebenarnya) yang dibutuhkan untuk menyelesaikan sebuah <i>task/user story</i> dalam satuan hari

Pada tabel 3 dijelaskan atribut yang digunakan pada dataset instansi X yang terdiri dari 8 atribut berupa data numerik dan kategorikal. Atribut data numerik yaitu *mandays*, *story points*, SiFP, *Unadjusted Use Case Point (UUCP)*, jumlah pengembang, pengalaman pengembang, *actual dev time*, sedangkan atribut data kategorikal yaitu Prioritas. Untuk mengetahui statistik deskriptif dari delapan fitur dataset proyek *scrum* instansi X dijelaskan pada Gambar 2.

Pada gambar 2 menunjukkan uraian ringkasan statistik untuk *count* yaitu jumlah data *not null* dalam setiap kolom, *mean* untuk rata-rata dalam setiap kolom, *std* yaitu standar deviasi untuk mengukur seberapa tersebar data dari rata-ratanya, *min* yaitu nilai minimum pada setiap kolom, 25% yaitu nilai data di bawah 25%, 50% yaitu nilai median atau nilai tengah dari data, 75% nilai data di bawah 75%, dan *max* nilai maksimum pada setiap kolom.

Berdasarkan gambar 3, *feature correlation* dapat dilihat berdasarkan fitur yang berpengaruh dengan *class*. Melihat dari statistik tersebut gelapnya warna pada fitur menandakan bahwa korelasi fitur tersebut tergolong baik [18]. Pada Gambar 3, terlihat bahwa kolom SiFP, *Mandays* dan UUCP memiliki

korelasi yang kuat, ditandai dengan warna merah tua pada *heatmap*. Ini menunjukkan hubungan erat antara kolom-kolom tersebut, sehingga perubahan nilai pada satu kolom cenderung mempengaruhi kolom lainnya. Sebaliknya, kolom Prioritas tidak menunjukkan korelasi signifikan dengan kolom lainnya.



Gambar 3 Feature Correlation Heatmap

3.2 Preprocessing Data

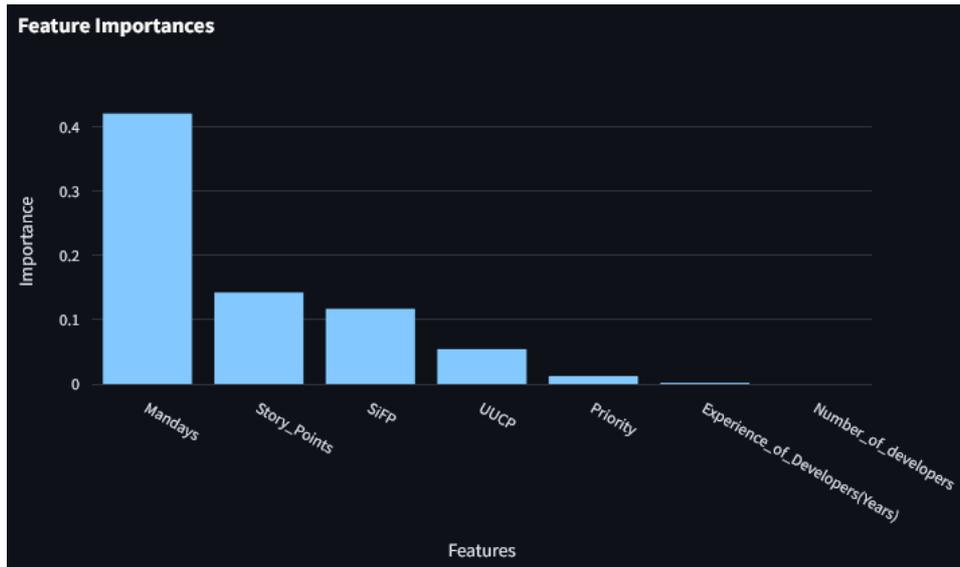
Pada tahap preprocessing dilakukan penanganan *missing value* dengan melakukan pengecekan baris data yang terduplikasi, memiliki nilai nol dalam kolomnya, dan melakukan normalisasi dengan tujuan mengurangi redundansi data dan memastikan dependensi data. *Preprocessing* pertama yang dilakukan yaitu menghapus data yang terduplikasi, memperbaiki data yang bernilai 0 dengan melakukan perbaikan data untuk mengisi data yang kosong dengan nilai rata-rata dari data yang ada dan memperbaiki atribut kategorikal yang kosong dengan nilai atribut yang sering muncul pada atribut kategorikal.

Pada dataset awal memiliki fitur atribut sebanyak 12, yakni: *Story_ID*, *Deskripsi_story*, *UGEP*, *UGDG*, *SiFP*, *Story Points*, *Mandays*, *Priority*, *Number of developers*, *Experience of Developers(Years)*, *UUCP*, *Actual Dev Time(Days)*. Terdapat 4 fitur yang tidak dipergunakan sebagai train dan test model machine learning yakni, *Story_ID*, *Deskripsi_story* di-drop dikarenakan bersifat nominal dan tidak dapat diperhitungkan dalam model regresi yang digunakan pada penelitian ini. Sedangkan untuk 2 fitur lainnya yakni, *UGEP* dan *UGDG* juga tidak digunakan pada dataset ini dikarenakan 2 fitur tersebut adalah data pembentuk fitur *SiFP* yang memiliki persamaan hitung yang sama.

Pada *preprocessing* kedua dilakukan perubahan data kategorikal menjadi nominal. *Preprocessing* dilakukan dengan menggunakan *LabelEncoder* dari *scikit learn*. Modul ini mengubah atribut kategorikal dengan tingkatan atau urutan tertentu menjadi atribut nominal. Kemudian dilakukan pengecekan terhadap isi data set setelah dilakukan perubahan pada atribut prioritas untuk memastikan kesesuaian perubahan tersebut. Kemudian dilakukan pengecekan pada tipe data yang akan diolah pada setiap fitur agar dapat sesuai dengan metode yang akan digunakan pada tahap seleksi model regresi. Berdasarkan hasil identifikasi deskripsi data set, Pada *preprocessing* yang ketiga dilakukan identifikasi untuk melihat apakah data merupakan *outlier* atau tidak, karena data yang terlalu ekstrem atau tidak biasa dapat mempengaruhi hasil analisis.

Pada *preprocessing* yang keempat dilakukan normalisasi data (*feature scaling*) agar variabel yang digunakan memiliki rentang nilai yang sama, tidak ada yang terlalu besar dan terlalu kecil. Metode normalisasi data yang digunakan ada *MinMaxScaler* sehingga mendapatkan nilai baru dengan skala 0 dan 1. Pemilihan *MinMaxScaler* dikarenakan mampu mempertahankan distribusi asli dari atribut yang dinormalisasi sehingga dapat membantu mempertahankan informasi asli dari data, atribut yang dinormalisasi pada rentang yang telah ditentukan, biasanya 0 dan 1.

Pada penelitian ini diperlukan juga untuk mengidentifikasi *feature importance* dari data set, seberapa signifikan setiap fitur (variabel) dalam memprediksi variabel target dalam model. *Feature importance* juga memberikan informasi tentang seberapa besar pengaruh masing-masing fitur (variabel independen) terhadap prediksi yang dihasilkan oleh model [19]. *Feature importance* pada gambar 4 menunjukkan, *Mandays* merupakan fitur yang paling penting dengan tingkat kepentingan tertinggi, diikuti oleh *Story Points* dan *SiFP* yang memiliki tingkat kepentingan hampir sama. Fitur *UUCP* juga cukup penting meskipun tidak sebesar tiga fitur sebelumnya. *Priority*, *Experience of Developers(years)*, dan *Number of developers* memiliki tingkat kepentingan yang relatif rendah dalam model ini.



Gambar 4 Feature Importance

3.3 Seleksi Model

Dalam penelitian ini, dataset dibagi menjadi dua bagian utama, yaitu data *training* dan data *testing*. Metode yang digunakan dalam melakukan evaluasi model yaitu *k-folds cross validation* dengan nilai $k=10$, di mana melibatkan pemisahan data set menjadi sepuluh bagian yang sama ukurannya secara acak [14]. Model dilatih menggunakan 9 sub set sebagai data *training* dan 1 sub set yang tersisa digunakan untuk data pengujian kinerja model. Proses pelatihan dan pengujian diulang sebanyak 10 kali dengan menggunakan sub set yang berbeda setiap kali, sehingga setiap sub set digunakan untuk pengujian satu kali. Setelah metode *k-folds cross validation* diterapkan, maka data dilatih dengan metode *Ensemble Methods* dan dibandingkan dengan beberapa metode seperti, *Support Vector Regression*, *Random Forest Regression*, dan *AdaBoost Regression*.

Pada penelitian ini dilakukan penerapan *tuning hyperparameter* dengan metode *GridSearch* untuk mendapatkan parameter yang optimal dengan data training. Teknik *Gridsearch* yaitu dengan membuat *grid* lalu mengevaluasi model pada setiap model *grid*. *Gridsearch* dilakukan dengan menentukan rentang nilai yang diinginkan [20]. Berikut adalah hasil parameter yang digunakan pada setiap model yang diperbandingkan menggunakan *library scikit-learn* dengan bahasa pemrograman *Python3*.

3.3.1 Random Forest Regression

Pada metode *Random Forest Regression*, proses pencarian parameter dilakukan dengan mendefinisikan range *hyperparameter* yang digunakan. Kemudian masukan model *Random Forest Regression* dan range *hyperparameter*. Setelah itu lakukan *k-fold cross validation* pada data *training*. Berikut nilai parameter terbaik metode *Random Forest Regression* yakni, $max_depth = 5$, $min_samples_leaf = 1$, $min_samples_split = 2$, $max_features = 6$ dan $n_estimators = 2$. Kemudian ditemukan bahwa *k-fold score* dengan model random forest regression yakni 0.9970497485141662.

3.3.2 Support Vector Regression

Pada metode *Support Vector Regression*, proses pencarian parameter dilakukan dengan mendefinisikan range *hyperparameter* yang digunakan. Kemudian buat dalam satu *dictionary* untuk dapat dimasukkan kedalam model *support vector regression*. Proses pencarian dilanjutkan dengan melakukan *k-fold cross validation* terhadap model menggunakan data training. Berikut nilai parameter yang didapatkan pada model *support vector regression* yakni, $kernel = linear$, $C = 1$ dan $Gamma = auto$. Kemudian ditemukan bahwa *k-fold score* dengan model support vector regression yakni 0.9823384107354534.

3.3.3 Ada Boost Regression

Pada metode *Ada Boost Regression*, proses pencarian sama seperti pada dua model sebelumnya dengan mendefinisikan *range*

hyperparameter dan dimasukkan pada *object* model *Ada Boost Regression* dengan data training yang telah di split sebelumnya. Berikut adalah nilai parameter terbaik untuk metode *AdaBoost Regression* yang didapat pada parameter *learning_rate* = 0.001, *n_estimators* = 500 dan *random_state* = 1. Kemudian ditemukan bahwa *k-fold score* dengan model *ada boost regression* yakni 0.9656481601266267.

3.3.4 Ensemble Methods

Ensemble methods yang digunakan pada penelitian ini merupakan pengambilan model dengan *hyperparameter* yang telah ditentukan diambil pengukuran *k-fold score* tertinggi diantara ketiga model lainnya, kemudian dilakukan proses *Bagging* untuk menambah tingkat performa yang optimal. Pada pengujian ini didapatkan *Random Forest Regression* memiliki tingkat *k-fold score* yang paling tinggi dibandingkan dengan model lainnya.. Kemudian dilakukan proses *bagging* untuk meningkatkan performa model. Proses *bagging* dilakukan dengan *setting default* pada *library scikit-learn* yakni *BaggingRegressor* (*estimator* = None, *n_estimators* = 10, *max_samples* = 1.0, *max_features* = 1.0, *bootstrap* = True, *bootstrap_features* = False, *oob_score* = False, *warm_start* = False, *n_jobs* = None, *random_state* = None, *verbose* = 0). Ditemukan bahwa *k-fold score* dengan model *ensemble* meningkatkan *k-fold score* menjadi 0.9982746175296983.

3.4 Evaluasi dan Validasi Model

Evaluasi model dilakukan dengan menggunakan beberapa metode *machine learning*, yaitu *Random Forest Regression*, *Support Vector Regression*, *AdaBoost Regression*, dan *Ensemble Method*. Evaluasi model dilakukan menggunakan dataset yang telah melalui preprocessing seperti dijelaskan pada tahap sebelumnya. Pemodelan dilakukan dengan menggunakan *library scikit-learn* dalam bahasa pemrograman Python3. Pada tahap evaluasi ini, metrik yang akan digunakan meliputi *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Square Error* (MSE) dan *R-Square* (R2). Tujuannya adalah untuk

menentukan metode yang lebih unggul dalam memprediksi waktu pengembangan dengan mempertimbangkan ketepatan dan akurasi prediksi.

Pada tabel 4 dijelaskan bahwa prediksi pada keempat algoritma tersebut sangat dekat nilainya. Penilaian kinerja dengan menggunakan SiFP dan algoritma *Random Forest Regression*, *AdaBoost Regression*, *Support Vector Regression* serta *Ensemble Method* dilatih dan diuji menggunakan atribut-atribut yang dipilih dalam *scrum project*, maka dapat menghasilkan prediksi waktu pengembangan yang lebih akurat ketika *Simple Function Point* sebagai salah satu atribut independen. Hasil perbandingan nilai evaluasi R2 pada model *Ensemble Methods* dataset menggunakan SiFP mendapatkan hasil yang lebih baik dibandingkan dengan dataset tanpa SiFP, nilai RMSE dataset menggunakan SiFP mendapatkan hasil yang lebih baik dibandingkan dengan dataset tanpa menggunakan SiFP, nilai MSE dataset menggunakan SiFP mendapatkan hasil yang lebih baik dibandingkan dengan dataset tanpa menggunakan SiFP. Hasil perbandingan nilai evaluasi R2, RMSE, MSE, dan MAE pada model *Random Forest Regresion*, *Ada Boost Regression* dan *Support Vector Regression* menunjukkan hasil perbandingan yang sebaliknya jika menambahkan nilai independen SiFP. Nilai evaluasi menunjukkan lebih baik tanpa menggunakan SiFP.

Metode *ensemble* dengan SiFP adalah pilihan terbaik untuk meminimalkan kesalahan prediksi. Pada tabel 4, model *ensemble* dengan penggunaan fitur SiFP memiliki nilai evaluasi yang lebih baik disetiap matriks pengujian. Hal ini disebabkan dikarenakan faktor *feature importance* dari SiFP merupakan 3 fitur utama sebagai peningkatan model yang diuji. SiFP memberikan informasi tambahan yang relevan dan signifikan untuk masalah terkait dengan dataset yang diuji. Data SiFP menjadi salah satu fitur penting dalam peningkatan kinerja yang lebih tinggi jika dibandingkan tanpa menggunakan fitur SiFP.

Tabel 4: Hasil Perbandingan Evaluasi Kinerja Model

No	MODEL	R2		RMSE		MSE		MAE	
		dengan SiFP	tanpa SiFP						
1	<i>Random Forest Regression</i>	0.6498	0.6588	0.088	0.0869	0.0077	0.0075	0.0732	0.072
2	<i>AdaBoost Regression</i>	0.5932	0.621	0.0949	0.0916	0.009	0.0084	0.0764	0.0717
3	<i>Support Vector Regression</i>	0.5936	0.6296	0.0948	0.0905	0.009	0.0082	0.0672	0.0679
4	<i>Ensemble Methods</i>	0.7698	0.6964	0.0714	0.0819	0.0051	0.0067	0.0589	0.0674

4. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan oleh penulis, dapat disimpulkan bahwa penggunaan *simple function point* sebagai salah satu atribut independen pada prediksi waktu pengembangan memiliki tingkat kesalahan evaluasi yang lebih rendah dibandingkan tanpa menggunakan *simple function point*. Performa model *ensemble method* dalam memprediksi estimasi waktu pengembangan juga menunjukkan tingkat kesalahan yang lebih rendah dibandingkan dengan metode tunggal. Sehingga hasil prediksi waktu pengembangan pada dataset *scrum project* dengan penambahan fitur SiFP menggunakan model *bagging ensemble method* memberikan tingkat kesalahan yang lebih rendah dengan nilai evaluasi kinerja R2 dataset dengan SiFP 0.7698 dan tanpa SiFP 0.6964, nilai RMSE dengan SiFP 0.0714 dan RMSE tanpa SiFP 0.0819, nilai MSE dengan SiFP 0.0051 dan MSE tanpa SiFP 0.0067, sedangkan nilai MSE dengan SiFP 0.0589 dan tanpa SiFP - 0.0674. Hasil dari eksperimen membuktikan bahwa *Ensemble Method* bisa digunakan untuk memprediksi waktu pengembangan secara cepat dan efisien dibandingkan dengan model *machine learning* tunggal. . Saran dalam penelitian ini yaitu memperluas pengujian dengan menggunakan metode *machine learning* yang lain serta menggunakan dataset *scrum project* yang lebih banyak dalam penelitian.

DAFTAR PUSTAKA

- [1] T. Halkjelsvik and M. Jørgensen, "From Origami to software development: A review of studies on judgment-based predictions of performance time," *Psychol Bull*, vol. 138, no. 2, pp. 238–271, Mar. 2012, doi: 10.1037/a0025996.
- [2] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation," *Softw Pract Exp*, vol. 52, no. 1, pp. 39–65, Jan. 2022, doi: 10.1002/spe.3009.
- [3] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão, and E. Insfran, "An update on effort estimation in agile software development: A systematic literature review," 2020, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2020.3021664.
- [4] K. Zhang, X. Wang, J. Ren, and C. Liu, "Efficiency Improvement of Function Point-Based Software Size Estimation with Deep Learning Model," *IEEE Access*, vol. 9, pp. 107124–107136, 2021, doi: 10.1109/ACCESS.2020.2998581.
- [5] L. Lavazza and R. Meli, "An evaluation of simple function point as a replacement of IFPUG function point," in *Proceedings - 2014 Joint Conference of the International Workshop on Software Measurement, IWSM 2014 and the International Conference on Software Process and Product Measurement, Mensura 2014*, Institute of Electrical and Electronics Engineers Inc., Dec. 2014, pp. 196–206. doi: 10.1109/IWSM.Mensura.2014.28.
- [6] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski, "An effective

- approach for software project effort and duration estimation with machine learning algorithms," *Journal of Systems and Software*, vol. 137, pp. 184–196, Mar. 2018, doi: 10.1016/j.jss.2017.11.066.
- [7] Z. Sakhravi, A. Sellami, and N. Bouassida, "Software Enhancement Effort Estimation using Machine Learning Regression Methods," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 12, pp. 412–423, 2020.
- [8] J. T. H. Jose Thiago and A. L. I. Oliveira, "Ensemble Effort Estimation using dynamic selection," *Journal of Systems and Software*, vol. 175, May 2021, doi: 10.1016/j.jss.2021.110904.
- [9] O. H. Alhazmi and M. Z. Khan, "Software Effort Prediction Using Ensemble Learning Methods," *Journal of Software Engineering and Applications*, vol. 13, no. 07, pp. 143–160, 2020, doi: 10.4236/jsea.2020.137010.
- [10] E. Haerani, F. Syafria, F. Lestari, and I. Marzuki, "CLASSIFICATION ACADEMIC DATA USING MACHINE LEARNING FOR DECISION MAKING PROCESS," *Journal of Applied Engineering and Technological Science*, vol. 4, no. 2, pp. 955–968, May 2023.
- [11] I. Sembiring, S. N. Wahyuni, and E. Sedyono, "LSTM algorithm optimization for COVID-19 prediction model," *Heliyon*, vol. 10, no. 4, Feb. 2024, doi: 10.1016/j.heliyon.2024.e26158.
- [12] S. Clara *et al.*, "Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learning Untuk Prediksi Penghasilan Pada Adult Income Dataset," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, pp. 743–744, Apr. 2021.
- [13] V. R. Prasetyo, M. Mercifia, A. Averina, L. Sunyoto, and Budiarjo, "PREDIKSI RATING FILM PADA WEBSITE IMDB MENGGUNAKAN METODE NEURAL NETWORK," *Jurnal Ilmiah NERO*, vol. 7, no. 1, Apr. 2022.
- [14] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?," *Comput Stat*, vol. 36, no. 3, pp. 2009–2031, Sep. 2021, doi: 10.1007/s00180-020-00999-9.
- [15] A. Vabalasid, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," 2019, doi: 10.1371/journal.pone.0224365.
- [16] B. Alsaadi and K. Saeedi, "Ensemble effort estimation for novice agile teams," *Inf Softw Technol*, vol. 170, p. 107447, 2024, doi: <https://doi.org/10.1016/j.infsof.2024.107447>.
- [17] L. Lavazza, A. Locoro, and R. Meli, "Software Development Effort Estimation Using Function Points and Simpler Functional Measures: a Comparison under Creative Commons License Attribution 4.0 International (CC BY 4.0)," vol. 23, 2023, [Online]. Available: <http://ceur-ws.org>
- [18] E. N. R. Khakim, A. Hermawan, and D. Avianto, "IMPLEMENTASI CORRELATION MATRIX PADA KLASIFIKASI DATASET WINE," *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 1, p. 158, Feb. 2023, doi: 10.26798/jiko.v7i1.771.
- [19] R. Rodríguez-Pérez and J. Bajorath, "Feature importance correlation from machine learning indicates functional relationships between proteins and similar compound binding characteristics," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-93771-y.
- [20] T. Aulia, E. Putri, T. Widiharih, and R. Santoso, "PENERAPAN TUNING HYPERPARAMETER RANDOMSEARCHCV PADA ADAPTIVE BOOSTING UNTUK PREDIKSI KELANGSUNGAN HIDUP PASIEN GAGAL JANTUNG," *Jurnal Gaussian*, vol. 11, no. 3, pp. 397–406, Jan. 2023, doi: 10.14710/J.GAUSS.11.3.397-406.