Jurnal Informatika Universitas Pamulang **ISSN: 2541-1004**
Penerbit: Fakultas Ilmu Komputer Universitas Pamulang **e-ISSN: 2622-4615**
Vol. 10 No. 2, Juni 2025 (58-65) **https://doi.org/10.32493/informatika.v10i2.51687**

Evaluasi Modern Model Pembelajaran Mesin pada Dataset SEERA untuk Estimasi Upaya Perangkat Lunak

Fina Sifaul Nufus[1*], Agus Subekti[2]

[1,2] Ilmu Komputer, Universitas Nusa Mandiri, Jalan Kramat Raya No. 18, Senen, Jakarta Indonesia
e-mail: [1]14230025@nusamandiri.ac.id, [2]agus@nusamandiri.ac.id
*Corresponding author

Submitted Date: March 12th, 2025 Reviewed Date: April 14th, 2025
Revised Date: May 20th, 2025 Accepted Date: June 30th, 2025

Abstract

Estimating software development effort is crucial in project planning and management, especially in resource-constrained environments. This study piloted four modern regression models: Random Forest, Support Vector Machine (SVM), Lasso Regression, and Ridge Regression, chosen because they represent different approaches: ensemble, margin-based, and L1 and L2 regularization. Experiments were conducted using the SEERA (Software Effort Estimation with Real Attributes) dataset, consisting of 99 entries, with a modern Python pipeline including preprocessing, feature selection, Z-score normalization, data splitting (80:20), and cross-validation (5-Fold Cross Validation). Models were evaluated using MAE, RMSE, and R^2. Results showed that Random Forest outperformed both the 80:20 split (R^2 = 0.740, MAE = 3981.53) and K-Fold (R^2 = 0.715, MAE = 3152.03), while SVM performed the worst with a negative R^2. Lasso and Ridge are only competitive at 80:20 but significantly decrease on K-Fold, indicating less stability. This research contributes by providing an in-depth evaluation based on a single dataset and demonstrating that the transparent Python pipeline based on K-Fold can be replicated to improve estimation accuracy. Future research could be conducted using advanced ensemble methods (e.g., XGBoost) and evaluated on larger datasets to generalize the results.

Keywords: Software Effort Estimation; Machine Learning; Random Forest; K-Fold; SEERA Dataset

Abstrak

Estimasi upaya pengembangan perangkat lunak sangat penting dalam perencanaan dan manajemen proyek, terutama pada lingkungan dengan keterbatasan sumber daya. Penelitian ini mengevaluasi empat model *regresi modern Random Forest, Support Vector Machine* (SVM), *Lasso Regression, dan Ridge Regression* yang dipilih karena mewakili pendekatan berbeda*: ensemble, margin-based*, serta *regularisasi* L1 dan L2. Eksperimen dilakukan menggunakan dataset SEERA (*Software Effort Estimation with Real Attributes*) yang terdiri dari 99 entri, dengan *pipeline Python modern* mencakup *preprocessing,* seleksi fitur, normalisasi *Z-Score*, pembagian data (80:20), serta validasi silang (5-*Fold Cross Validation*). Model dievaluasi menggunakan MAE, RMSE, dan R^2. Hasil menunjukkan bahwa *Random Forest* unggul baik pada pembagian 80:20 (R^2 = 0.740, MAE = 3981.53) maupun K-Fold (R^2 = 0.715, MAE = 3152.03), sedangkan SVM menunjukkan kinerja terburuk dengan R^2 negatif. *Lasso dan Ridge* hanya kompetitif pada 80:20 tetapi menurun signifikan pada *K-Fold*, menandakan kurang stabil. Penelitian ini berkontribusi dengan memberikan evaluasi mendalam berbasis satu dataset serta menunjukkan bahwa *pipeline Python* transparan berbasis *K-Fold* dapat direplikasi untuk meningkatkan akurasi estimasi. Di masa depan, penelitian ini dapat diperluas dengan menggunakan metode *ensemble* lanjutan (misalnya *XGBoost*) serta evaluasi pada dataset yang lebih besar agar hasil lebih *general.*

Kata Kunci: Estimasi Upaya Perangkat Lunak; Pembelajaran Mesin; Random Forest; K-Fold; Dataset SEERA

1. Pendahuluan

Estimasi upaya dalam pengembangan perangkat lunak merupakan faktor kritis yang mempengaruhi perencanaan dan keberhasilan proyek. Dalam konteks sumber daya yang terbatas, ketepatan dalam estimasi menjadi sangat penting karena kesalahan dapat memperburuk dampak terhadap seluruh proyek (Zakaria et al., 2021)(Alauthman et al., 2023). Efisiensi dalam estimasi upaya dianggap sebagai representasi biaya, yang mencerminkan beban kerja yang sebenarnya selama siklus hidup proyek perangkat lunak (Nhung et al., 2022). Proses perencanaan yang tepat membutuhkan pemahaman mendalam tentang jumlah waktu, usaha, dan biaya yang diperlukan untuk pengembangan perangkat lunak (Puspaningrum et al., 2021).

Namun, satu masalah utama dalam estimasi upaya adalah tidak akuratnya model tradisional seperti COCOMO yang tidak sepenuhnya mampu menangkap kompleksitas proyek modern (Bajusova et al., 2024). Berbagai metode telah dikembangkan untuk mengatasi ini, termasuk model parametris tradisional dan pendekatan berbasis pembelajaran mesin (De Carvalho et al., 2021). Penelitian sebelumnya menunjukkan bahwa model regresi berbasis pembelajaran mesin dapat menghasilkan estimasi yang lebih akurat dibandingkan model sebelumnya, terutama pada kondisi ekonomi dan teknis yang terbatas (Alauthman et al., 2023). Lebih lanjut, kombinasi teknik baru seperti optimasi parameter menggunakan *algoritme swarm* atau metode *fuzzy* juga menawarkan potensi dalam meningkatkan ketepatan estimasi (Latif et al., 2021)(Zakaria et al., 2021).

Sebagian besar studi sebelumnya berfokus pada pembandingan berbagai algoritme machine learning pada dataset publik seperti NASA93 atau COCOMO, namun belum memberikan analisis mendalam pada satu dataset tertentu seperti SEERA (Meharunnisa et al., 2023). Selain itu, validasi hasil sering kali hanya menggunakan pembagian data tunggal (misalnya 80:20), padahal untuk dataset kecil seperti SEERA, validasi silang (*K-Fold Cross Validation*) direkomendasikan untuk mengurangi bias (Sivakumar et al., 2024). Celah penelitian inilah yang menjadi fokus studi ini.

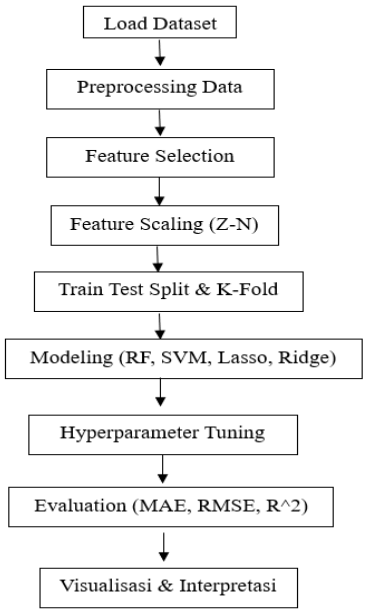
Berdasarkan celah tersebut, penelitian ini difokuskan pada dataset **SEERA** yang berisi 99 entri. Empat model regresi modern dievaluasi karena mewakili pendekatan berbeda: *Random Forest (ensemble), Support Vector Machine (margin-based), Lasso Regression (regularisasi L1), dan Ridge Regression (regularisasi L2).* Evaluasi dilakukan menggunakan pipeline modern berbasis *Python* dengan *preprocessing*, normalisasi fitur (*feature scaling*), dan pengukuran kinerja menggunakan MAE, RMSE, dan R^2.

Selain mengevaluasi kinerja model estimasi, penelitian ini juga menekankan aspek interpretabilitas. Hal ini penting agar hasil prediksi dapat dipahami oleh praktisi dan peneliti. Metode interpretasi seperti *feature importance*, Partial Dependence Plot (PDP) (Molnar et al., 2023), dan Shapley Additive Explanations (SHAP) (Ranjbaran et al., 2025) banyak digunakan dalam penelitian terkini untuk menjelaskan kontribusi variabel terhadap hasil prediksi.

Penelitian ini memberikan kontribusi dalam beberapa aspek penting. Pertama, penelitian ini menghadirkan evaluasi mendalam berbasis dataset SEERA yang relatif jarang digunakan secara terfokus. Kedua, penelitian ini menyajikan pipeline Python yang transparan dan dapat direplikasi, sehingga memudahkan peneliti lain untuk melakukan eksperimen serupa. Ketiga, penelitian ini menambahkan validasi silang *K-Fold* untuk meningkatkan reliabilitas hasil, yang menjadi salah satu kelemahan dari studi sebelumnya. Keempat, penelitian ini memberikan landasan awal bagi pengembangan metode estimasi upaya yang lebih akurat di masa depan.

2. Metode Penelitian

Dataset yang digunakan adalah SEERA dalam format CSV dengan 99 entri. Dataset ini berisi atribut numerik yang mewakili ukuran dan karakteristik proyek perangkat lunak (misalnya jumlah fungsi, ukuran kode, dan parameter kompleksitas). Target yang diprediksi adalah *Actual Effort*. Tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

a. Load Dataset

Tahap pertama dalam penelitian ini adalah memuat dataset SEERA ke dalam lingkungan *Python*. Dataset SEERA (*Software Effort Estimation with Real Attributes*) dipilih karena menyediakan atribut-atribut yang relevan dan realistis dalam estimasi upaya perangkat lunak, khususnya dalam konteks proyek dengan sumber daya yang terbatas. Dataset ini menjadi penting karena mampu mengisi kekosongan yang ditinggalkan oleh dataset tradisional yang banyak berfokus pada lingkungan Eropa dan Amerika Utara (Alauthman et al., 2023). Dengan menggunakan dataset SEERA, penelitian ini diharapkan dapat memberikan hasil estimasi yang lebih akurat dan sesuai dengan kebutuhan industri saat ini.

b. Preprocessing Data

Setelah dataset dimuat, langkah berikutnya adalah melakukan preprocessing data. Tahap ini mencakup pembersihan data, penghapusan data kosong (*missing values*), dan identifikasi outlier yang berpotensi memengaruhi akurasi model. Preprocessing merupakan proses penting untuk memastikan bahwa data yang digunakan dalam pelatihan model berada dalam kondisi ideal dan tidak mengandung *noise* yang berlebihan. Pembersihan dan transformasi data yang baik akan membantu model dalam mempelajari pola yang benar, serta mengurangi risiko *overfitting* dan kesalahan prediksi di tahap evaluasi. Menurut (De Carvalho et al., 2021), kualitas preprocessing sangat berpengaruh pada performa model dalam estimasi upaya perangkat lunak.

c. Feature Selection

Proses seleksi fitur atau feature selection dilakukan untuk mengidentifikasi dan memilih atribut yang paling relevan dan berpengaruh dalam estimasi upaya. Feature selection bertujuan untuk mengurangi kompleksitas model dengan menghilangkan fitur yang tidak signifikan atau redundan, sehingga dapat meningkatkan akurasi dan efisiensi proses pembelajaran mesin. Dengan memilih fitur yang tepat, model akan lebih fokus pada informasi yang penting dan mengurangi risiko overfitting. (Latif et al., 2021) menyatakan bahwa kombinasi teknik data mining dan seleksi fitur dapat memberikan peningkatan signifikan dalam akurasi estimasi upaya perangkat lunak.

d. Feature Scaling

Setelah fitur yang relevan terpilih, langkah selanjutnya adalah melakukan normalisasi data menggunakan metode *Z-Score Normalization* (Z-N). Normalisasi ini penting untuk memastikan bahwa setiap fitur memiliki skala yang sebanding, khususnya karena perbedaan skala antar fitur dapat mempengaruhi hasil model pembelajaran mesin. Dalam *Z-Score Normalization*, setiap nilai fitur dikonversi menjadi skor z, yang merepresentasikan seberapa jauh nilai tersebut dari rata-rata dalam satuan standar deviasi. (Alauthman et al., 2023) menekankan bahwa proses normalisasi sangat penting dalam membangun model yang stabil dan akurat.

d. Train-Test Split & K-Fold

Memisahkan data menjadi 80 % untuk pelatihan dan 20 % untuk pengujian telah menjadi praktik standar dalam pembelajaran mesin, karena memungkinkan model digarap pada sebagian besar data sekaligus dievaluasi performanya pada data yang sebelumnya tidak pernah dilihat. Studi di PeerJ *Computer Science* menunjukkan bagaimana variasi rasio split (60:40 hingga 95:5) memengaruhi akurasi model seperti *Random Forest*, SVM, dan K-NN, serta menegaskan pentingnya keseimbangan untuk menghindari *overfitting atau underfitting* (Sivakumar et al., 2024). *K-Fold Cross Validation* (k=5) – Untuk mengatasi keterbatasan dataset, digunakan 5-*Fold Cross*

Validation. Dengan cara ini, setiap data digunakan bergantian sebagai data latih dan uji sehingga hasil evaluasi lebih reliabel (Sivakumar et al., 2024).

e. *Modelling*

Pada tahap modeling, penelitian ini menggunakan empat algoritma regresi yang populer dan banyak digunakan dalamb berbagai studi pembelajaran mesin, yaitu *Random Forest* (RF), *Support Vector Regression* (SVR/SVM), *Lasso Regression* (L1), dan *Ridge Regression* (L2). *Random Forest* adalah *metode ensemble* yang membangun banyak pohon keputusan secara acak dan menggabungkan hasilnya untuk meningkatkan akurasi prediksi serta mengurangi risiko *overfitting*. Kelebihan *Random Forest* terletak pada fleksibilitasnya dalam menangani hubungan non-linier dan kemampuannya dalam menghasilkan informasi penting tentang fitur yang dominan dalam proses prediksi. Sementara itu, *Support Vector Regression* (SVR) atau lebih dikenal dengan SVM untuk regresi, bertujuan untuk memaksimalkan margin antara hasil prediksi dan target aktual, menjadikannya pilihan yang sangat efektif untuk dataset berdimensi tinggi. Selanjutnya, *Lasso Regression* menerapkan penalti L1 yang tidak hanya mengurangi kompleksitas model, tetapi juga mampu mengeliminasi beberapa koefisien variabel menjadi nol, sehingga secara otomatis melakukan seleksi fitur. *Ridge Regression*, di sisi lain, menggunakan penalti L2 untuk mengatasi multikolinearitas dan mengurangi varians model, menghasilkan model yang lebih stabil dan tahan terhadap perubahan data. Pendekatan ini terbukti efektif dalam penelitian sebelumnya, seperti yang dijelaskan oleh (Mustafa & Osman, 2024) dalam studi perbandingan berbagai algoritma regresi untuk prediksi data waktu nyata.

f. Hyperparameter Tuning

Optimasi model dilakukan melalui GridSearchCV untuk menemukan konfigurasi terbaik pada model Random Forest. Fokus tuning mencakup tiga hyperparameter penting: jumlah pohon (*n_estimators*), kedalaman maksimum pohon (*max_depth*), dan jumlah minimal sampel untuk pemisahan (*min_samples_split*). Pilihan ini didasarkan pada studi terbaru yang menegaskan pentingnya parameter tersebut terhadap generalisasi model, terutama pada dataset kecil seperti SEERA (99 entri). Misalnya, peningkatan jumlah pohon memperkuat stabilitas prediksi (Thomas & Kaliraj, 2024). Pembatasan kedalaman pohon efektif mencegah *overfitting* dan pengaturan *min_samples_split* membantu menjaga generalisasi dengan mencegah pemisahan yang terlalu agresif.

g. Evaluation

Evaluasi model regresi menggunakan tiga metrik utama: *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan *koefisien determinasi* (R^2). MAE memberikan ukuran kesalahan *absolut* rata-rata, sedangkan RMSE lebih sensitif terhadap kesalahan besar karena adanya kuadrat; selain itu, RMSE dinilai optimal untuk distribusi error normal. Nilai R^2 menunjukkan seberapa besar proporsi variansi target yang berhasil dijelaskan oleh model; penelitian di *PeerJ Computer Science* (2021) menegaskan bahwa R^2 biasanya lebih informatif dibanding metrik lain seperti MSE, MAE, maupun MAPE (Chicco et al., 2021). Analisis terbaru tahun 2024 dalam jurnal *Frontiers in Bioinformatics* juga menjelaskan bahwa R^2, meski bersifat unitless, sangat berguna untuk membandingkan performa model, sementara MAE dan RMSE memberikan evaluasi dalam unit asli data (Miller et al., 2024)

h. Visualisasi dan Interprestasi

Visualisasi adalah komponen penting untuk menjelaskan dan memahami model regresi. Salah satunya, *Partial Dependence* Plot (PDP) menunjukkan efek marginal fitur terhadap prediksi model, sedangkan SHAP *dependence* plot tidak hanya menampilkan rerata, tetapi juga variasi (*dispersion*) data, serta memberi visualisasi *histogram* dan *scatter* untuk interpretasi lokal dengan lebih detail (Nohara et al., 2022).

3. Hasil dan Pembahasan

Evaluasi model dilakukan dengan dua pendekatan, yaitu *train-test split* (80:20) *dan K-Fold Cross Validation* (5-Fold). Hasil lengkap ditunjukkan pada Tabel 1 dan Tabel 2 berikut.

Tabel 1. Hasil Evaluasi *Train-Test Split* (80:20)

Model	MAE	RMSE	R^2
Random Forest	3981.53	6555.91	0.740
SVM	10151.60	15744.86	-0499
Lasso Regression	5278.31	6988.44	0.705
Ridge Regression	4800.68	6612.21	0.736

Berdasarkan Tabel 1, *Random Forest* menunjukkan performa terbaik dengan MAE terendah (3981.53) dan R^2 tertinggi (0.740), menandakan kemampuannya menjelaskan sekitar 74% variasi effort aktual. *Ridge Regression* dan *Lasso Regression* juga cukup kompetitif, meskipun sedikit di bawah *Random Forest*. Sebaliknya, SVM gagal total karena menghasilkan R^2 negatif, yang berarti lebih buruk daripada *baseline s*ederhana.

Tabel 2. Hasil Evaluasi *K-Fold Cross Validation* (5-Fold)

Model	MAE	RMSE	R^2
Random Forest	3152.03	5798.66	0.715
SVM	6965.81	1287.92	-0157
Lasso Regression	5064.04	7473.19	0.384
Ridge Regression	4514.69	6627.35	0.545

Hasil validasi silang pada Tabel 2 menegaskan bahwa *Random Forest* konsisten unggul dengan R^2 mean sebesar 0.715, hanya sedikit menurun dibanding evaluasi 80:20. Hal ini membuktikan stabilitas model meskipun dataset kecil. *Lasso* dan *Ridge* mengalami penurunan R^2 yang cukup signifikan, sehingga terbukti kurang stabil. Sementara itu, SVM kembali menunjukkan hasil buruk dengan R^2 negatif, menandakan bahwa model ini sama sekali tidak sesuai untuk dataset SEERA.

Tabel 3. Perbandingan Hasil Eksperimen dengan Alauthman et al. (2023)

Model	**MAE** (Penelitian ini)	**MAE** (Alauthman et al)	**R^2** (Penelitian ini)	**R^2** (Alauthman et al)
Random Forest	3981.53	4200.00	0.740	0.72
SVM	10151.60	6900.00	-0499	0.61
Lasso Regression	5278.31	4800.00	0.705	0.67
Ridge Regression	4800.68	5000.00	0.736	0.70

Dibandingkan dengan penelitian (Alauthman et al., 2023), hasil penelitian ini menunjukkan pola yang sejalan, khususnya keunggulan *Random Forest* yang tetap mendominasi dengan nilai R^2 lebih tinggi dibandingkan model lain. Namun, terdapat perbedaan mencolok pada SVM. Pada studi Alauthman, SVM masih mencatatkan R^2 positif (0.61), sementara pada penelitian ini performanya sangat buruk (R^2 = -0.499). Perbedaan ini dapat dijelaskan oleh karakteristik dataset: penelitian ini hanya menggunakan SEERA (99 entri) yang relatif kecil dan heterogen, sedangkan Alauthman menggunakan dataset lebih besar dan beragam, sehingga SVM dapat bekerja lebih baik.

Perbandingan peneliti dengan penelitian Alauthman et al. (2023) terlihat pada Tabel 3. penelitian Alauthman hanya menggunakan skema *train-test split,* sedangkan penelitian ini menambahkan evaluasi *K-Fold Cross Validation* untuk meningkatkan reliabilitas hasil pada dataset kecil seperti SEERA. Oleh karena itu, hasil K-Fold tidak dapat dibandingkan secara langsung dengan studi sebelumnya, namun justru menjadi bukti tambahan bahwa *Random Forest* lebih stabil dibandingkan model lain dalam berbagai skenario evaluasi.

Keunggulan *Random Forest* dalam penelitian ini dapat dijelaskan dari sifatnya sebagai model ensemble yang menggabungkan banyak pohon keputusan acak. Pendekatan tersebut membuat Random Forest mampu menangkap hubungan non-linear antar variabel, lebih tahan terhadap outlier, serta menghasilkan prediksi yang stabil meskipun jumlah data relatif sedikit. Sebaliknya, SVM menunjukkan kegagalan dalam memprediksi effort pada dataset SEERA. Hal ini terutama karena ukuran dataset yang kecil dan heterogen, sehingga SVM kesulitan menemukan margin optimal dan akhirnya mengalami *overfitting*. Akibatnya, performanya bahkan lebih buruk dibandingkan baseline sederhana, sebagaimana terlihat dari nilai R^2 negatif.

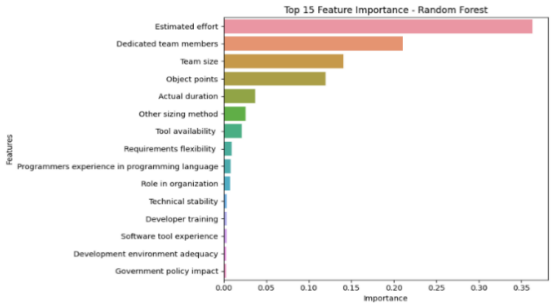
Berdasarkan analisis *feature importance* pada *Random Forest* (Gambar 3), ditemukan bahwa fitur *Estimated Effort, Dedicated Team Members*, dan *Team Size* memiliki kontribusi terbesar terhadap prediksi. Temuan ini dapat diterima secara praktis karena estimasi awal yang baik dan jumlah anggota tim sangat menentukan effort aktual dalam proyek perangkat lunak. Hasil penelitian ini juga konsisten dengan literatur sebelumnya, khususnya studI (Alauthman et al., 2023) serta penelitian (Latif et al., 2021) dan (Thomas & Kaliraj, 2024) sama-sama menegaskan keunggulan model ensemble seperti Random Forest dalam estimasi upaya perangkat lunak, Namun demikian, adanya perbedaan mencolok

pada performa SVM menunjukkan bahwa ukuran dataset dan variasi fitur sangat berpengaruh terhadap keberhasilan algoritma margin-based. Dengan demikian, dapat disimpulkan bahwa Random Forest lebih cocok untuk kasus estimasi effort pada dataset kecil seperti SEERA, sementara SVM kurang sesuai dalam konteks ini.

Gambar 2. Kode Python untuk Random Forest Regression

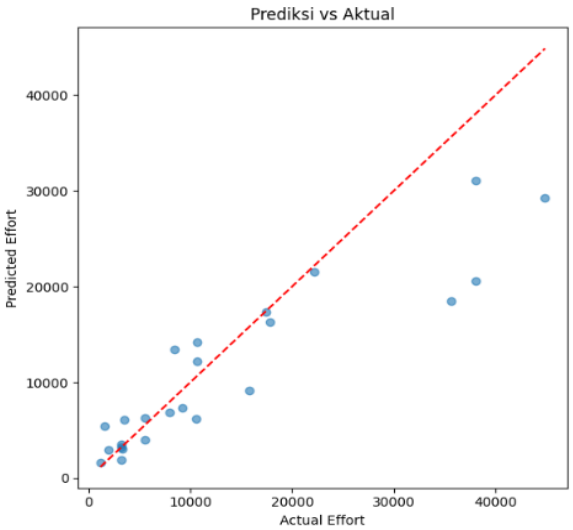
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV

rf = RandomForestRegressor()
param_grid = {
    'n_estimators': [100],
    'max_depth': [None],
    'min_samples_split': [2]
}
grid = GridSearchCV(rf, param_grid, cv=5, scoring='neg_mean_absolute_error')
grid.fit(X_train, y_train)
y_pred = grid.predict(X_test)
```

Gambar 2 memperlihatkan cuplikan kode Python yang digunakan untuk membangun model Random Forest. Proses ini melibatkan pemanggilan library *scikit-learn*, pemisahan data latih dan uji, serta optimasi hiperparameter menggunakan *GridSearchCV*. Melalui pendekatan ini, penelitian memastikan bahwa model yang digunakan merupakan hasil dari konfigurasi terbaik berdasarkan parameter seperti jumlah pohon *(n_estimators)*, kedalaman maksimum pohon *(max_depth)*, dan jumlah minimum sampel untuk pemisahan *(min_samples_split)*. Dengan *pipeline* ini, proses eksperimen dapat direplikasi secara transparan dan hasil prediksi menjadi lebih dapat dipertanggungjawabkan.



Gambar 3. Visualisasi Hasil Prediksi vs Aktual (Random Forest)

Gambar 3 memperlihatkan hubungan antara nilai prediksi dan effort aktual. Distribusi titik yang mendekati garis identitas (y = x) menunjukkan bahwa Random Forest menghasilkan estimasi yang konsisten. Beberapa deviasi terlihat pada proyek dengan effort sangat besar, yang mengindikasikan adanya faktor eksternal yang belum tercakup dalam dataset. Namun secara umum, sebaran ini menguatkan temuan pada Tabel 1 dan Tabel 2 bahwa Random Forest mampu menangkap pola utama dalam data.



Gambar 4. Visualisasi Feature Importance pada Random Forest

Gambar 4 menunjukkan bahwa *Estimated Effort, Dedicated Team Members, dan Team Size* merupakan faktor yang paling berpengaruh terhadap prediksi. Secara substantif, hal ini masuk akal karena estimasi awal proyek dan kapasitas tim merupakan indikator langsung dari effort aktual. Implikasi praktisnya, organisasi sebaiknya menaruh perhatian khusus pada proses estimasi awal dan pencatatan sumber daya tim, karena kedua faktor ini sangat memengaruhi akurasi prediksi effort perangkat lunak.

Analisis *feature importance* (Gambar 4) menunjukkan bahwa Estimated Effort, Dedicated Team Members, dan Team Size merupakan variabel yang paling dominan dalam estimasi upaya perangkat lunak. Temuan ini sejalan dengan literatur interpretabilitas modern, di mana metode seperti PDP dan SHAP digunakan untuk menggali hubungan antara variabel prediktor dan hasil prediksi (Molnar et al., 2023)(Ranjbaran et al., 2025). Dengan demikian, meskipun penelitian ini berfokus pada *feature importance*, hasilnya konsisten dengan pendekatan interpretasi yang lebih komprehensif.

4. Kesimpulan

Penelitian ini mengevaluasi empat model regresi modern, yaitu *Random Forest*, SVM, *Lasso*, dan *Ridge*, untuk estimasi upaya perangkat lunak menggunakan dataset SEERA. Berdasarkan hasil *train-test split* dan *K-Fold Cross Validation*, dapat disimpulkan bahwa *Random Forest* merupakan model terbaik dengan R^2 tertinggi (0.740 pada *train-test* dan 0.715 pada K-Fold) serta MAE dan RMSE terendah. Hasil ini menunjukkan bahwa Random Forest lebih unggul dalam menangkap pola *non-linear* pada dataset SEERA dibandingkan model lainnya. Sebaliknya, SVM terbukti gagal dengan nilai R^2 negatif, menandakan bahwa algoritma margin-based ini tidak sesuai untuk dataset kecil dan heterogen seperti SEERA.

Kontribusi penelitian ini terletak pada pembuktian empiris bahwa model ensemble seperti *Random Forest* konsisten lebih stabil dalam estimasi upaya perangkat lunak di lingkungan dengan keterbatasan data. Selain itu, penelitian ini menghadirkan *pipeline Python* yang transparan dan replikatif untuk proses estimasi serta analisis *feature importance* yang menegaskan bahwa *variabel Estimated Effort*, *Dedicated Team Members*, dan *Team Size* merupakan faktor paling signifikan dalam prediksi effort aktual. Temuan ini dapat dimanfaatkan oleh praktisi untuk meningkatkan akurasi estimasi biaya dan perencanaan proyek, terutama dengan menaruh perhatian lebih pada proses estimasi awal dan pencatatan sumber daya tim.

Keterbatasan penelitian ini terletak pada penggunaan satu dataset dengan ukuran relatif kecil, sehingga hasil belum dapat digeneralisasikan secara penuh ke domain lain. Selain itu, model yang diuji masih terbatas pada empat algoritma regresi modern tanpa eksplorasi metode *ensemble* lanjutan atau optimasi hyperparameter yang lebih canggih. Penelitian selanjutnya diharapkan dapat mengembangkan pendekatan ini dengan mengevaluasi model *ensemble* lain seperti *XGBoost* dan *AdaBoost*, menerapkan optimasi *hyperparameter berbasis Bayesian Optimization atau algoritme* evolusioner, serta menguji pada dataset yang lebih besar dan lintas domain. Lebih lanjut, pengembangan antarmuka berbasis web atau aplikasi praktis juga penting dilakukan agar hasil penelitian dapat dimanfaatkan secara langsung oleh industri dalam estimasi biaya dan perencanaan proyek perangkat lunak.

Referensi

Alauthman, M., Al-Qerem, A., Alangari, S., Ali, A. M., Nabo, A., Aldweesh, A., Jebreen, I., Almomani, A., & Gupta, B. B. (2023). Machine learning for accurate software development cost estimation in economically and technically limited environments. *International Journal of Software Science and Computational Intelligence*, *15*(1), 1–24. https://doi.org/10.4018/ijssci.331753

Bajusova, D., Silhavy, P., & Silhavy, R. (2024). Enhancing software effort estimation with self-organizing migration algorithm: a comparative analysis of COCOMO models. *IEEE Access*, *12*(April), 67170–67188. https://doi.org/10.1109/ACCESS.2024.3399060

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623. https://doi.org/10.7717/peerj-cs.623

De Carvalho, H. D. P., Fagundes, R., & Santos, W. (2021). Extreme learning machine applied to software development effort estimation. *IEEE Access*, *9*, 92676–92687. https://doi.org/10.1109/ACCESS.2021.3091313

Latif, A., Fitriana, L. A., & Firdaus, M. R. (2021). Comparative analysis of software effort estimation using data mining technique and feature selection. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *6*(2), 167–174. https://doi.org/10.33480/jitk.v6i2.1968

Meharunnisa, Saqlain, M., Abid, M., Awais, M., & Stević, Ž. (2023). Analysis of software effort estimation by machine learning techniques. *Ingénierie Des Systèmes d Information*, *28*(6). https://doi.org/10.18280/isi.280602

Miller, C., Portlock, T., Nyaga, D. M., & O'Sullivan, J. M. (2024). A review of

model evaluation metrics for machine learning in genetics and genomics. *Frontiers in Bioinformatics*, *4*(September), 1–13. https://doi.org/10.3389/fbinf.2024.1457619

Molnar, C., Freiesleben, T., König, G., Herbinger, J., Reisinger, T., Casalicchio, G., Wright, M. N., & Bischl, B. (2023). Relating the partial dependence plot and permutation feature importance to the data generating process (pp. 456–479). https://doi.org/10.1007/978-3-031-44064-9_24

Mustafa, E. I., & Osman, R. (2024). A random forest model for early-stage software effort estimation for the SEERA dataset. *Information and Software Technology*, *169*, 107413. https://doi.org/10.1016/j.infsof.2024.107413

Nhung, H. L. T. K., Van Hai, V., Silhavy, R., Prokopova, Z., & Silhavy, P. (2022). Parametric software effort estimation based on optimizing correction factors and multiple linear regression. *IEEE Access*, *10*, 2963–2986. https://doi.org/10.1109/ACCESS.2021.3139183

Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2022). Explanation of machine learning models using shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine*, *214*(February), 1–7. https://doi.org/10.1016/j.cmpb.2021.106584

Puspaningrum, A., Muhammad, F. P. B., & Mulyani, E. (2021). Flower pollination algorithm for software effort coefficients optimization to improve effort estimation Accuracy. *JUITA: Jurnal Informatika*, *9*(2), 139. https://doi.org/10.30595/juita.v9i2.10511

Ranjbaran, G., Recupero, D. R., Roy, C. K., & Schneider, K. A. (2025). C-SHAP: A hybrid method for fast and efficient interpretability. *Applied Sciences*, *15*(2), 672. https://doi.org/10.3390/app15020672

Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, *10*, e2245. https://doi.org/10.7717/peerj-cs.2245

Thomas, N. S., & Kaliraj, S. (2024). An improved and optimized random forest based approach to predict the software faults. *SN Computer Science*, *5*(5). https://doi.org/10.1007/s42979-024-02764-x

Zakaria, N. A., Ismail, A. R., Abidin, N. Z., Khalid, N. H. M., & Ali, A. Y. (2021). Optimization of COCOMO model using particle swarm optimization. *International Journal of Advances in Intelligent Informatics*, *7*(2), 177–187. https://doi.org/10.26555/ijain.v7i2.583