JURNAL SAINS DAN TEKNOLOGI
KEILMUAN DAN APLIKASI TEKNOLOGI INDUSTRI
E-ISSN : 2615-2827 P-ISSN : 1412-5455
Volume 25, Nomor 2 Tahun 2025
https://ejurnalsttind.id/index.php/SainsdanTeknologi

Prediksi Pencapaian Target Kerja untuk Optimasi Manajemen Proyek Menggunakan Metode *Long Short Term Memory* dan *Random Forest*

Rizqatasyaa Achmad Zahra [1)*], Dhomas Hatta Fudholi [2)]

[1,2] Universitas Islam Indonesia, Yogyakarta, Indonesia

23917015@students.uii.ac.id*; hatta.fudholi@uii.ac.id

ABSTRAK

Dalam era digitalisasi, pengelolaan sumber daya manusia dalam proyek teknologi informasi menghadapi tantangan signifikan, khususnya dalam memperkirakan kebutuhan mandays secara akurat. Mandays merupakan satuan yang merepresentasikan total hari kerja yang dibutuhkan satu atau lebih tenaga kerja untuk menyelesaikan suatu tugas atau proyek. Ketidaktepatan estimasi mandays dapat menyebabkan risiko *under allocation* dan *over allocation*, yang berdampak pada pemborosan anggaran, ketidakseimbangan beban kerja, serta keterlambatan proyek. Permasalahan serupa masih dijumpai pada berbagai perusahaan berbasis proyek, baik di sektor teknologi informasi, konstruksi, maupun manufaktur, karena proses estimasi umumnya masih dilakukan secara manual dan belum didukung sistem prediktif berbasis data historis. Penelitian ini mengusulkan model prediktif berbasis kombinasi metode LSTM dan *Random Forest* dengan pendekatan *error reciprocal,* yaitu pemberian bobot lebih besar pada model dengan tingkat kesalahan prediksi (MAE) yang lebih rendah. Prediksi mandays dilakukan berdasarkan faktor-faktor proyek yang meliputi jenis proyek, status, durasi, jumlah SDM, dan nilai proyek. LSTM digunakan untuk mengenali pola temporal pada data historis proyek, sedangkan *Random Forest* untuk menangani hubungan *non-linear* dan menghasilkan prediksi yang lebih stabil. Hasil pengujian menunjukkan model tunggal LSTM memiliki performa yang lebih baik dibandingkan RF, dengan nilai MAE masing-masing sebesar 286.21 dan 369.57. Namun, model kombinasi LSTM-RF memberikan performa terbaik dengan nilai MAE sebesar 279.32 dan MAPE 17.65%. Evaluasi tambahan menggunakan metrik MMRE < 25% dan PRED(25) ≥ 75%, menunjukkan bahwa model kombinasi berada dalam kategori prediksi yang baik serta mampu meningkatkan akurasi prediksi sekitar 4% dibandingkan model tunggal. Dengan demikian, model LSTM-RF efektif dalam memprediksi pencapaian target kerja dan mendukung pengambilan keputusan dalam manajemen proyek TI.

Kata kunci: Prediksi Target Kerja, LSTM-RF, Error Reciprocal, Deep Learning, Manajemen Proyek

ABSTRACT

In the era of digitalization, human resource management in information technology projects faces significant challenges, particularly in accurately estimating manday requirements. A manday is a unit representing the total workdays required by one or more workers to complete a task or project. Inaccurate mandays estimations may lead to risks of under and over allocation, which can lead to budget wastage, workload imbalance, and project delays. Similar issues are still frequently encountered in various project-based industries, including IT, construction, and manufacturing, where estimation processes are generally conducted manually without the support of predictive systems based on historical data. This study proposes a predictive model based on a combination of the LSTM and Random Forest methods with a reciprocal error approach, which assigns greater weight to models with a lower prediction error rate (MAE). Manday predictions are based on project factors including project type, status, duration, number of human resources, and project value. LSTM is used to recognize temporal patterns in historical project data, while Random Forest handles non-linear relationships and produces stable predictions. The test results show that the single LSTM model has better performance than the RF model, with MAE of 286.21 and 369.57, respectively. However, the LSTM-RF model provides the best performance with MAE of 279.32 and MAPE of 17.65%. Additional evaluation using MMRE < 25% and PRED(25) ≥ 75% metrics, shows that the combined model is in the good prediction category and is able to increase prediction accuracy by about 4% compared to the single model. These findings indicate that the LSTM-RF model is effective in predicting the achievement of work targets and supporting decision making in IT project management.

Keywords: Work target prediction, LSTM-RF, Reciprocal error, Deep learning, Project management

Copyright (c) 2025 Rizqatasyaa Achmad Zahra, Dhomas Hatta Fudholi
DOI: https://doi.org/10.36275/8m9pg216

PENDAHULUAN

Di era digitalisasi, industri teknologi informasi dituntut untuk meningkatkan efektivitas dan efisiensi pelaksanaan proyek agar dapat beradaptasi dengan dinamika bisnis yang terus berkembang (Sudiantini, dkk., 2023; Indrayani, 2012). Salah satu permasalahan utama dalam pengelolaan proyek TI adalah ketidakakuratan dalam memprediksi pencapaian target kerja, terutama dalam estimasi kebutuhan mandays yang dibutuhkan untuk menyelesaikan setiap tugas proyek. Ketidakakuratan estimasi ini menyebabkan dua kondisi krusial dalam manajemen SDM yaitu *under allocation* dan *over allocation*, yang berdampak pada pemborosan anggaran, keterlambatan penyelesaian proyek, serta penurunan produktivitas proyek yang secara langsung berdampak pada pencapaian target kerja (Ramadhan, 2024).

Permasalahan serupa tidak hanya terjadi di sektor teknologi informasi, tetapi juga di berbagai sektor industri lain seperti konstruksi dan manufaktur yang mengandalkan estimasi mandays untuk perencanaan sumber daya manusia. Ketidaktepatan estimasi menyebabkan ketidakseimbangan distribusi beban kerja, di mana beberapa karyawan tidak menerima tugas selama satu siklus proyek (*under allocation*), sementara lainnya harus menangani beberapa tugas secara bersamaan (*over allocation*). Dalam kasus lain, kelebihan jumlah SDM pada proyek berskala kecil menyebabkan pemborosan anggaran tanpa kontribusi kerja yang proporsional terhadap *output* proyek. Di sisi lain, proses penentuan target kerja masih banyak dilakukan secara manual tanpa dukungan sistem prediktif berbasis data historis, sehingga tidak mampu mengantisipasi dinamika proyek yang kompleks dan berubah secara *real-time*.

Penelitian-penelitian terdahulu telah mengeksplorasi berbagai pendekatan prediktif berbasis *machine learning* dan *deep learning* dalam hal manajemen proyek, terutama untuk prediksi durasi, alokasi sumber daya, dan efisiensi biaya. Beberapa studi menunjukkan bahwa metode seperti LSTM dan *Random Forest* mampu meningkatkan akurasi prediksi kinerja proyek. Namun demikian, sebagian besar penelitian tersebut masih berfokus pada penggunaan model tunggal dan optimalisasi aspek tertentu dan belum secara spesifik mengkaji estimasi kebutuhan mandays sebagai indikator utama pencapaian target kerja proyek TI.

Beberapa penelitian juga telah mengembangkan pendekatan prediktif yang lebih kompleks, seperti penggabungan LSTM dengan metode optimasi atau model prediksi lanjutan. Meskipun menghasilkan peningkatan performa, pendekatan-pendekatan tersebut umumnya belum mempertimbangkan karakteristik temporal data proyek secara mendalam maupun megintegrasikan model *ensemble* untuk meningkatkan stabilitas prediksi pada data proyek yang bersifat kompleks dan tidak seimbang, serta belum mempertimbangkan pembobotan kesalahan model dalam proses pengambilan keputusan. Untuk memperjelas penelitian ini terhadap penelitian terdahulu, ringkasan kajian pustaka disajikan pada Tabel 1.

Tabel 1. Kajian Pustaka

No	Peneliti	Metode	Objek Penelitian	Hasil	Kelemahan
1	Sanjaya & Budi (2020)	LSTM + DEA	Pencapaian target kerja proyek TI	Prediksi pencapaian target kerja dengan tingkat akurasi sebesar 71,2%	Akurasi masih terbatas, belum memanfaatkan model *ensemble*
2	Pratama et al. (2023)	*Random Forest*	Alokasi SDM dan Alokasi tugas proyek	Peningkatan efisiensi yang signifikan, antara lain pengurangan biaya operasional sebesar 14%, percepatan fase	Tidak fokus pada estimasi mandays

				perencanaan hingga 88,7%, serta peningkatan waktu penyelesaian proyek sebesar 50,8%	
3	Farhanuddin et al. (2024)	*Multiple Linear Regression* & *Random Forest*	Anggaran proyek Sistem Informasi	RF unggul (R^2 = 81,5%), dibandingkan dengan *Multiple Linear Regression*	Belum mempertimbangkan aspek temporal atau pola sekuensial data proyek
4	Park & Song (2019)	LSTM + G*raf bipartit minimum cost and maximum flow*	Optimasi alokasi sumber daya dalam proses bisnis	Menurunkan *weighted completion time* yang signifikan dibandingkan pendekatan *baseline* tradisional	Belum mengombinasikan dengan model *ensemble* untuk meningkatkan stabilitas dan akurasi prediksi dalam data proyek yang bersifat kompleks dan tidak seimbang
5	Kumari et al. (2025)	LSTM PRO	Prediksi durasi tugas, perencanaan sumber daya, dan risiko keterlambatan	Akurasi tinggi (R^2 = 0,96) dan MSE sebesar 0,0025	Belum mempertimbangkan struktur temporal data proyek dan belum menggabungkannya dengan metode pembobotan *error* dan model *ensemble*

Berdasarkan kajian terhadap penelitian-penelitian terdahulu, telah ditunjukkan bahwa penggunaan LSTM maupun *Random Forest* efektif dalam prediksi durasi proyek, alokasi sumber daya, dan efisiensi proyek. Namun, masih terdapat celah penelitian dalam hal penggabungan kedua metode untuk prediksi jumlah mandays pada proyek TI. Sebagian besar penelitian tersebut masih berfokus pada penerapan masing-masing metode secara terpisah, tanpa memanfaatkan keunggulan temporal dari LSTM dan kemampuan generalisasi serta kestabilan prediksi dari *Random Forest* secara bersamaan (Hoseinzade & Haratizadeh, 2019; Dachi, 2023). Penerapan kombinasi model ini menjadi lebih menguntungkan karena mampu mengatasi keterbatasan masing-masing metode, seperti yang diungkapkan oleh Vuong et al., (2022). Oleh karena itu, penelitian ini mengusulkan pengembangan model prediktif kombinasi LSTM dan *Random Forest* dengan pendekatan *error reciprocal method*, yaitu pemberian bobot lebih besar pada model dengan tingkat kesalahan prediksi yang lebih rendah berdasarkan nilai *Mean Absolute Error* (MAE).

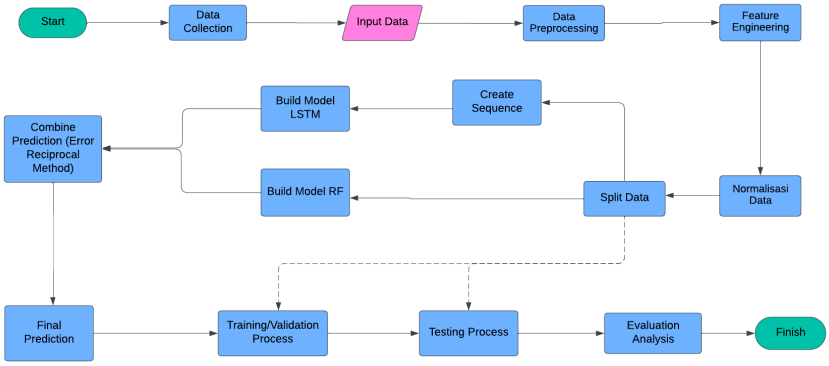
Untuk membangun model tersebut, digunakan sejumlah variabel *input* yang mencerminkan karakteristik proyek, yaitu jenis proyek, status proyek, durasi, jumlah SDM, dan nilai proyek. Sementara itu, variabel target yang diprediksi adalah jumlah mandays aktual yang dibutuhkan. Data kemudian disusun dalam bentuk sekuensial agar pola temporal dapat dikenali oleh LSTM, sedangkan *Random Forest* berperan menjaga kestabilan hasil melalui kemampuannya dalam mengolah variasi data yang kompleks. Dengan pendekatan ini, keunggulan kedua model dapat saling melengkapi sehingga menghasilkan estimasi kebutuhan mandays yang lebih akurat dan stabil untuk mendukung perencanaan target kerja proyek.

Kebaruan dari penelitian ini terletak pada pengembangan model prediktif kebutuhan mandays proyek teknologi informasi yang mengintegrasikan LSTM dan *Random Forest* melalui pendekatan *error reciprocal method*. Berbeda dengan penelitian sebelumnya yang umumnya menerapkan LSTM atau *Random Forest* secara terpisah, penelitian ini mengombinasikan keunggulan LSTM dalam menangkap pola temporal data proyek dengan kemampuan *Random Forest* dalam menjaga kestabilan prediksi pada data yang kompleks dan

tidak seimbang. Selain itu, penggunaan pembobotan adaptif berbasis nilai *Mean Absolute Error* (MAE) memungkinkan kontribusi masing-masing model disesuaikan secara objektif berdasarkan performanya. Pendekatan ini memberikan kontribusi metodologis yang belum banyak dikaji dalam prediksi mandays proyek TI dan menawarkan solusi yang lebih akurat dan stabil untuk mendukung perencanaan target kerja serta pengambilan keputusan manajerial berbasis data.

METODE

Penelitian ini menggunakan pendekatan kuantitatif dengan rancangan eksploratif-prediktif untuk membangun dan mengevaluasi model prediksi kebutuhan mandays berbasis *machine learning* dan *deep learning* pada proyek TI di PT. XYZ, sebuah perusahaan ICT yang bergerak dalam pengolahan data besar. Penggunaan *dataset* yang berasal dari satu perusahaan menyebabkan hasil penelitian ini memiliki keterbatasan dalam hal generalisasi ke perusahaan lain dengan karakteristik yang berbeda. Adapun tahapan penelitian secara menyeluruh disajikan pada Gambar 1.



Gambar 1. Tahapan Penelitian

1. Data *Collection*

Data yang digunakan adalah data sekunder berupa *dataset* historis proyek yang diperoleh langsung dari arsip dan sistem internal Perusahaan. *Dataset* penelitian ini terdiri dari 386 proyek yang terbagi ke dalam tiga kategori utama, yaitu *project*, *internal*, dan *maintenance*. Kategori *project* umumnya mencakup aktivitas pengembangan aplikasi baru, implementasi sistem informasi. Kategori *internal* berkaitan dengan pengembangan produk aplikasi atau *tools* yang dimiliki perusahaan serta kegiatan riset terhadap produk internal. Sementara itu, kategori *maintenance* berfokus pada monitoring aplikasi yang telah berjalan, perbaikan bug, pembaruan sistem, serta penyesuaian sesuai kebutuhan klien.

Data yang diperoleh menyimpan informasi terkait faktor-faktor proyek, dan juga data mandays aktual. Selain itu, perusahaan juga mencatat mandays target hasil perhitungan manual ($durasi \times jumlah\ SDM$). Selisih antara mandays aktual dan target digunakan untuk melihat deviasi estimasi pada tiap kategori. Distribusi jumlah proyek beserta total selisih mandays dari tiap proyek ditunjukkan pada Tabel 2.

Tabel 2. Distribusi Data Proyek

Kategori	Jumlah Proyek	Total Selisih Mandays
Project	152	93.433
Maintenance	139	14.210
Internal	95	35.204

Berdasarkan Tabel 2, dapat dilihat bahwa kategori *project* mendominasi baik dari sisi jumlah maupun total selisih mandays, diikuti oleh *internal* dan *maintenance*. Namun, analisis deviasi saja belum cukup untuk menggambarkan karakteristik distribusi data secara keseluruhan (Ritonga, dkk. 2025). Oleh karena itu, dilakukan pula analisis statistik deskriptif terhadap mandays aktual pada data uji yang mencakup 116 proyek. Ringkasan statistik deskriptif tersebut ditampilkan pada Tabel 3, yang memberikan gambaran umum mengenai sebaran nilai mandays aktual, termasuk rata-rata mandays yang kemudian dijadikan acuan dalam perhitungan *error* relatif pada tahap evaluasi model.

Tabel 3. Statistik Deskriptif

Statistik	Nilai
Jumlah sampel (count)	116
Rata-rata (mean)	1.601,42
Simpangan baku (std)	1.396,69
Minimum	53
Kuartil 1 (25%)	547,25
Median (50%)	1.161,50
Kuartil 3 (75%)	2.439,25
Maksimum	6.798

2. Data *Preprocessing*

Tahap *pre-processing* meliputi *data cleaning*, *feature engineering*, normalisasi dan *split* data.

a. Data *Cleaning*

Salah satu teknik dalam *pre-processing* yang bertujuan untuk mengidentifikasi dan memperbaiki kesalahan serta inkonsistensi pada data, sehingga data menjadi lebih akurat dan dapat diandalkan. (Dessiaming et al., 2022).

b. *Feature Engineering*

c. Proses merekayasa fitur-fitur data yang diperlukan untuk membentuk representasi data yang optimal bagi model pembelajaran. Penelitian ini menggunakan *Label Encoding* untuk mengonversi data kategorikal yang menjadi numerik. Proses ini memberikan nilai angka yang unik pada setiap kategori atau label dalam variabel kategorikal (Larose & Coyle, 2020).

d. Normalisasi Data

Proses normalisasi dilakukan agar seluruh variabel berada pada rentang nilai yang sama, yakni antara 0 hingga 1, sehingga tidak ada variabel yang terlalu besar maupun terlalu kecil dan analisis statistik menjadi lebih mudah (Cahyani, dkk. 2023). Normalisasi pada penelitian ini menggunakan metode *Min-Max*, di mana nilai data hasil normalisasi (x_{norm}) dihitung dari selisih antara nilai data ke-i (x_i) dengan nilai minimum (x_{min}), kemudian dibagi dengan rentang antara nilai maksimum (x_{max}) dan nilai minimum (x_{min}) sebagaimana ditunjukkan pada Persamaan (1).

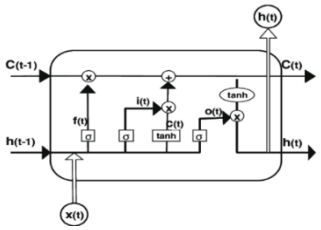
$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

e. *Split* Data

Pembagian data dilakukan untuk membagi data menjadi dua bagian yaitu data latih (*training*) dan data uji (*testing*). Model dilatih menggunakan data latih dan performanya diuji menggunakan data uji (Ramadhan & Ashari, 2024). Penelitian ini membagi data menjadi 70% *training* dan 30% *testing*, berdasarkan penelitian Sanjaya & Budi (2020).

3. *Long Short Term Memory* (LSTM)

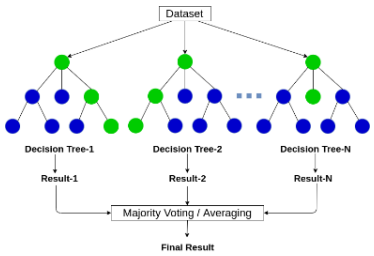
LSTM merupakan varian dari RNN yang dirancang untuk mengatasi masalah *vanishing gradient*. Masalah ini dapat menyebabkan hilangnya informasi penting dari data sebelumnya, sehingga LSTM digunakan untuk menjaga akumulasi nilai *gradien* secara optimal (Cholissodin, dkk., 2019). Dengan memanfaatkan *cell state* yang diperbarui sebagai memori jangka panjang seperti ditunjukkan pada Gambar 2, masukan berupa data saat ini (x_t), *hidden state* sebelumnya (h_{t-1}) dan *cell state* sebelumnya (C_{t-1}) diproses melalui tiga gerbang utama. *Forget gate* (f_t). mengatur informasi lama yang harus dilupakan, *Input gate* (i_t) bersama *candidate cell state* $(\tilde{C}_t)$ menambahkan informasi baru ke dalam memori. Sedangkan, *output gate* (o_t) menghasilkan *hidden state* terbaru (h_t). Mekanisme ini membuat LSTM mampu menyimpan informasi penting sekaligus menyesuaikan dengan data baru (Khalifatul & Prihanto, 2023).



Gambar 2. Struktur *Long-Short Term Memory*

4. *Random Forest* (RF)

Random Forest merupakan algoritma *ensemble learning* yang diperkenalkan oleh Leo Breiman (2001) dan banyak digunakan untuk klasifikasi maupun regresi. Algoritma ini membangun sejumlah *decision tree* dari sampel data hasil *bootstrap*, kemudian menggabungkan prediksi melalui *averaging* (regresi) atau *majority voting* (klasifikasi), sehingga meningkatkan akurasi, mengurangi risiko *overfitting*, dan mampu menangani data kompleks serta non-linear (Wardhana, dkk., 2022; Amaliah, dkk., 2022). Ilustrasi mekanisme tersebut ditunjukkan pada Gambar 3.



Gambar 3. Ilustrasi *Random Forest*

5. Kombinasi LSTM-RF

LSTM memiliki keterbatasan seperti kompleksitas tinggi akibat banyaknya parameter yang harus diproses terutama pada urutan yang panjang, serta rentan terhadap *overfitting* pada data kecil. Sedangkan, RF unggul dalam kemampuan prediktif, lebih stabil, dan tidak mudah mengalami *overfitting*. Untuk mengatasi kelemahan LSTM sekaligus memanfaatkan keunggulan RF, kedua model dikombinasikan menjadi model LSTM-RF dengan pendekatan *error reciprocal* (Guasing & Yihao, 2023). Pada metode ini, bobot ditentukan berdasarkan nilai *error* (MAE) dari masing-masing model, sebagaimana ditunjukkan pada Persamaan (2) dan (3), di mana (ε_l) dan (ε_r) merepresentasikan *error* LSTM dan *Random Forest*, sedangkan (w_l) dan (w_r) adalah bobot yang diberikan secara proporsional sehingga model dengan *error* lebih rendah memperoleh bobot lebih besar (Cao dkk., 2023). Selanjutnya, prediksi akhir $(y^i{}_l)$ untuk data ke-i dihitung melalui Persamaan (4), yaitu kombinasi linier antara prediksi LSTM $(y^i{}_l)$

dan *Random Forest* ($y^i{}_r$) yang telah diberi bobot (w_l) dan (w_r). Dengan mekanisme ini, kontribusi model yang lebih akurat menjadi dominan sehingga meningkatkan performa prediksi pada data berurutan yang kompleks (Abumoen dkk., 2024).

$$w_l = 1 - (\frac{\varepsilon_l}{\varepsilon_l + \varepsilon_r}) \quad (2)$$

$$w_r = 1 - (\frac{\varepsilon_r}{\varepsilon_l + \varepsilon_r}) \quad (3)$$

$$y^i = w_l y^i{}_l + w_r y^i{}_r \quad (4)$$

6. Evaluasi Model

Hasil prediksi yang telah didapatkan selanjutnya memasuki tahapan pengujian untuk memastikan keakurasiannya. Matrix evaluasi yang digunakan MAE, RMSE, dan MAPE.

1. *Mean Absolute Error* (MAE), merupakan nilai rata-rata kesalahan (*error*) *absolute* antara *output* prediksi dengan nilai data sebenarnya (sautomo & pardede, 2021). Sebagaimana ditunjukkan pada persamaan (5).

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - \hat{y}_t\right| \quad (5)$$

2. *Root Mean Square Error* (RMSE), merupakan akar kuadrat dari rata-rata perbedaan kuadrat antara prediksi dan observasi aktual dari sebuah data (Ashari & Sadiki, 2020). Sebagaimana ditunjukkan pada persamaan (6).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}{n}} \quad (6)$$

3. *Mean Ansolute Percentage Error* (MAPE), merupakan rata-rata selisih persentase antara data aktual dan hasil prediksi (Fachid & Triayudi, 2022). Sebagaimana ditunjukkan persamaan (7).

$$MAPE = \frac{100\%}{n}\left(\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|\right) \quad (7)$$

dengan :

n = Banyaknya data observasi

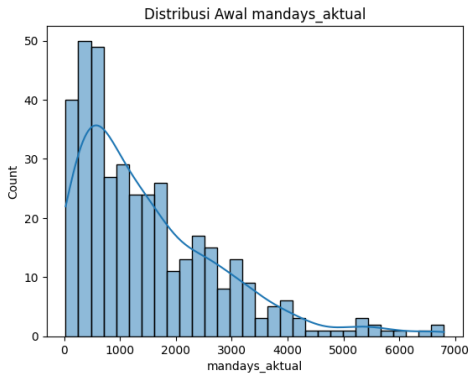
$\hat{y}_t$ = Nilai prediksi dari y

y_t = Nilai aktual atau data asli

HASIL DAN PEMBAHASAN

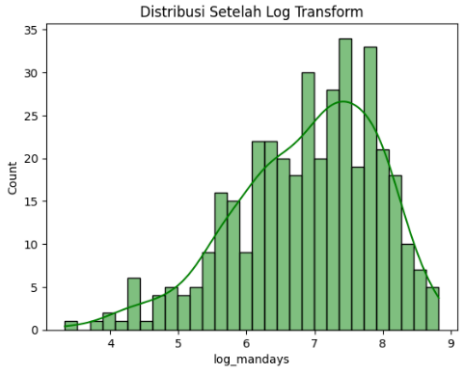
1. Eksplorasi dan *Pre-processing* Data

Eksplorasi data dilakukan untuk memahami karakteristik dan distribusi variabel target, mandays aktual menunjukkan pola *right-skewed* dengan *outlier* pada proyek-proyek tertentu, sebagaimana ditunjukkan pada Gambar 4. Sebagian besar proyek memiliki nilai mandays yang relatif rendah, namun terdapat beberapa proyek dengan nilai sangat tinggi yang menyebabkan penyebaran data tidak simetris. Distribusi ini berpotensi memengaruhi kinerja model (Ritonga, dkk. 2025). Khususnya LSTM yang sensitif terhadap skala data.



Gambar 4. Distribusi Awal Mandays Aktual

Untuk mengatasinya, dilakukan transformasi logaritmik untuk menstabilkan varians, memperkecil rentang nilai, serta mengurangi pengaruh *outlier* pada model menggunakan fungsi *np.log1p*, yang merupakan bentuk log-transformasi dengan penambahan konstanta 1 untuk menghindari nilai log dari nol. Hasil transformasi ini ditunjukkan pada Gambar 5, di mana distribusi mandays tampak lebih simetris dan mendekati bentuk distribusi normal. Dan hasil dari transformasi ini disimpan di kolom *log_mandays*.



Gambar 5. Distribusi Setelah *Log-Transform*

Selanjutnya dilakukan *pre-processing*. Pertama, data *cleaning* dengan pengecekan *missing value* pada seluruh kolom yang menunjukkan tidak adanya nilai kosong. Kemudian, menghapus kolom id karena hanya berfungsi sebagai identifikasi unik, sedangkan mandays aktual dihapus karena telah digantikan oleh log mandays.

Kedua, dilakukan proses konversi data kategorikal menjadi numerik menggunakan *label encoding*. Hasil konversi menunjukkan setiap kategori diurutkan berdasarkan abjad dan diberi indeks mulai dari nol ditunjukkan pada Gambar 6.

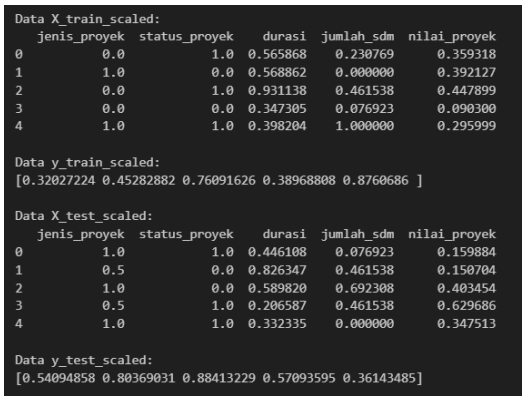
```
jenis_proyek:
  Nilai asli: ['Internal' 'Project' 'Maintenance']
  Nilai encoded: [0 2 1]

status_proyek:
  Nilai asli: ['DONE' 'CLOSED']
  Nilai encoded: [1 0]
```

Gambar 6. Hasil Konversi Variabel Kategori

Ketiga, dilakukan pembagian data menjadi 70% *training* dan 30% *testing* tanpa pengacakan *(shuffle=False)* untuk menjaga urutan waktu data. Selanjutnya proses normalisasi

pada seluruh fitur numerik, hasil dari proses normalisasi yang telah diterapkan pada data *training* dan *testing*, baik untuk variabel input *X_train*, *X_test* maupun target *y_train*, *y_test*. Seluruh nilai pada masing-masing fitur telah berhasil dikonversi ke dalam rentang [0,1] ditunjukkan pada Gambar 7.



```
Data X_train_scaled:
   jenis_proyek  status_proyek    durasi  jumlah_sdm  nilai_proyek
0           0.0            1.0  0.565868    0.230769      0.359318
1           1.0            0.0  0.568862    0.000000      0.392127
2           0.0            1.0  0.931138    0.461538      0.447899
3           0.0            0.0  0.347305    0.076923      0.090300
4           1.0            1.0  0.398204    1.000000      0.295999

Data y_train_scaled:
[0.32027224 0.45282882 0.76091626 0.38968808 0.8760686 ]

Data X_test_scaled:
   jenis_proyek  status_proyek    durasi  jumlah_sdm  nilai_proyek
0           1.0            1.0  0.446108    0.076923      0.159884
1           0.5            0.0  0.826347    0.461538      0.150704
2           1.0            0.0  0.589820    0.692308      0.403454
3           0.5            1.0  0.206587    0.461538      0.629686
4           1.0            1.0  0.332335    0.000000      0.347513

Data y_test_scaled:
[0.54094858 0.80369031 0.88413229 0.57093595 0.36143485]
```

Gambar 7. Hasil Normalisasi

2. Pemodelan

Pada penelitian ini untuk memprediksi pencapaian target kerja dengan indikator mandays dalam proyek TI menggunakan algoritma LSTM, *Random Forest* dan Kombinasi LSTM-RF.

2.1 *Long Short Term Memory*

Langkah awal dalam pembangunan model LSTM adalah mengubah format data dari 2D menjadi 3D. Format awal data hasil *pre-processing* adalah 2D, yaitu (jumlah sampel, jumlah fitur). Namun, model LSTM membutuhkan input data 3D dengan format (jumlah sampel, *timestep*, jumlah fitur), karena model ini dirancang untuk menangkap pola sekuensial atau temporal dalam data historis. Proses konversi ini dilakukan melalui fungsi *create_dataset()* yang menambahkan data *padding* nol di awal sebanyak *timesteps* - 1 untuk menjaga panjang data agar tetap konstan. *Padding* ini penting agar prediksi tetap bisa dilakukan dari titik awal data, dan pola temporal bisa dikenali sejak awal.

Setelah data dikonversi ke bentuk 3D, selanjutnya adalah menyusun arsitektur model dan melakukan *tuning* parameter. Model LSTM yang dibangun terdiri dari dua layer LSTM berturut-turut, masing-masing diikuti oleh layer *dropout* untuk mencegah *overfitting*, dan diakhiri dengan satu *layer Dense* untuk menghasilkan *output* berupa prediksi kebutuhan mandays.

Proses *tuning* dilakukan untuk mencari kombinasi parameter terbaik yang memberikan performa prediksi paling optimal. Pada model LSTM melakukan percobaan tiga kali *tuning* pada parameter *timestep* = [1, 3, 5]. Parameter *tuning* dapat dilihat pada Tabel 4.

Tabel 4. Hyperparameter Tuning LSTM

Hyperparameter	**Tuning**
units	18, 32, 64
timesteps	1, 3, 5
activation	linear, relu, tanh
learning_rates	0.00075, 0.001, 0.01
epochs	25, 50, 100

Peneliti menggunakan Pustaka dari *scikitLearn* yaitu *GridSearchCV*. Yang merupakan sebuah fungsi proses pencarian parameter terbaik menggunakan *cross-validation* untuk menemukan kombinasi parameter yang menghasilkan kinerja terbaik. *Tuning* terbaik hasil dari *GridSearchCV* untuk masing-masing kombinasi *timestep*, dirangkum dalam Tabel 5.

Tabel 5. Perbandingan Hasil *Tuning* Setiap *Timestep*

Timestep	MSE	Params
1	0.0022	'units': 64, 'activation': 'relu', 'learning_rate': 0.01, 'epochs': 50
3	0.0034	‘units': 18, 'activation': 'linear', 'learning_rate': 0.01, 'epochs': 100
5	0.0038	‘units': 64, 'activation': 'tanh', 'learning_rate': 0.01, 'epochs': 25

Berdasarkan Tabel 5, nilai MSE terendah diperoleh pada *timestep* = 1 dengan MSE sebesar 0,0022. Konfigurasi optimal pada *timestep* ini adalah *units* = 64, *activation* = 'relu', *learning_rate* = 0,01, dan *epochs* = 50. Nilai MSE yang lebih rendah menunjukkan kesalahan prediksi yang lebih kecil, sehingga *timestep* = 1 dipilih sebagai pengaturan terbaik dibandingkan *timestep* 3 maupun 5. Hal ini mengindikasikan bahwa, penggunaan *timestep* yang lebih pendek mampu menangkap pola data secara lebih efektif tanpa menambah kompleksitas yang berlebihan.

2.2 *Random Forest*

Sebagai metode pembanding terhadap model LSTM, peneliti juga membangun model *Random Forest*. Langkah pertama dalam pembangunan model *Random Forest* adalah inisialisasi model dengan parameter dasar atau *default*. Parameter awal yang digunakan dalam model ditampilkan pada Tabel 6.

Table 6. Parameter *Default Random Forest*

Params	Value
min_samples_split	2
min_samples_leaf	1
max_features	‘log2’
criterion	‘squared_error’
random_state	42

Model kemudian dibentuk menggunakan fungsi *RandomForestRegressor* dari *scikit-learn* dengan parameter *default*. Langkah berikutnya melakukan *tuning* parameter yang dapat dilihat pada pada Tabel 7.

Table 7. *Hyperparameter Tuning* RF

Hyperparameter	Tuning
n_estimators	50, 75, 100
Max_depth	None, 10, 15, 20

Tuning dilakukan pada dua parameter utama yaitu *n_estimators* untuk menentukan jumlah pohon pada *ensemble*, serta *max_depth* untuk mengatur kedalaman maksimum pohon. Hasil tuning menunjukkan parameter terbaik diperoleh pada *n_estimators* = 100 dan *max_depth* = 10, yang memberikan keseimbangan antara akurasi prediksi dan kemampuan generalisasi model.

3. Kombinasi Model LSTM-RF

Setelah masing-masing model tunggal dilatih dan dievaluasi, diperoleh hasil MAE dari masing-masing model yang ditunjukan pada Tabel 8.

Tabel 8. Nilai MAE Model Tunggal

Model	MAE
LSTM	286.21
Random Forest	369.57

Berdasarkan nilai MAE pada Tabel 8, MAE digunakan untuk melakukan kombinasi prediksi dari kedua model tersebut untuk memperoleh performa yang lebih optimal. Pendekatan kombinasi model ini dilakukan dengan mempertimbangkan kelemahan dan kekuatan masing-masing algoritma. Metode yang digunakan untuk menggabungkan kedua model adalah pendekatan *Error Reciprocal*, yaitu metode pembobotan berdasarkan kebalikan dari nilai *error* MAE masing-masing model. Model yang memiliki *error* lebih kecil akan diberikan bobot prediksi lebih besar karena dianggap lebih andal. Bobot masing-masing model dihitung menggunakan formula pada persamaan (1) dan (2). Hasil pembobotan ditunjukkan pada Tabel 9.

Tabel 9. Hasil Bobot *Error Reciprocal Method*

Model	Bobot
LSTM	0.5636
Random Forest	0.4364

Hasil perhitungan bobot pada Tabel 9, menunjukkan bahwa LSTM mendapat porsi kontribusi prediksi lebih besar karena memiliki tingkat *error* yang lebih rendah dibandingkan *Random Forest*. Hasil prediksi akhir akan lebih banyak dipengaruhi oleh LSTM, sementara RF tetap berkontribusi sebagai penyeimbang untuk mengatasi potensi kelemahan LSTM. pembobotan ini memungkinkan model kombinasi menghasilkan prediksi yang lebih stabil dan akurat dibandingkan penggunaan model tunggal. Setelah dihitung bobot, dilakukan prediksi menggunakan persamaan (4).

4. Evaluasi Model

Setelah seluruh model dibangun dan diuji, tahap berikutnya adalah evaluasi menggunakan tiga metrik utama, yaitu MAE, RMSE, dan MAPE. Ketiga metrik ini digunakan untuk menilai tingkat *error* baik secara *absolut* maupun proporsional. Hasil evaluasi ditampilkan pada Tabel 10, yang menjadi dasar dalam menentukan model terbaik untuk memprediksi pencapaian target kerja berdasarkan estimasi mandays.

Tabel 10. Evaluasi Perbandingan Model

Model	Training			Testing		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
LSTM	255.36	403.49	24.08%	286.21	453.86	21.50%
Random Forest	139.63	253.21	10.29%	369.57	639.62	21.49%
LSTM-RF	162.15	283.10	13.43%	279.32	499.98	17.65%

Berdasarkan Tabel 10, terlihat bahwa model kombinasi LSTM-RF memberikan performa prediksi yang paling optimal dibandingkan model tunggal. Pada data pengujian, model kombinasi mencatat nilai MAE sebesar 279.32 dan MAPE sebesar 17.65%, lebih rendah dibandingkan LSTM maupun *Random Forest*. Penurunan nilai MAPE dari sekitar 21.5% menjadi 17.65% menegaskan bahwa model kombinasi lebih akurat dalam memberikan estimasi dibandingkan model individu dengan akurasi sebesar 82,35%. Peningkatan akurasi ini sekitar 4%, yang dapat dianggap signifikan pada estimasi mandays.

Dengan rata-rata mandays aktual sebesar 1601.42 mandays yang ditunjukkan pada Tabel 3, nilai *error* relatif yang dihasilkan model kombinasi lebih kecil, sehingga estimasi lebih mendekati kondisi riil. Perhitungan *error* relatif ini diperoleh dengan membagi nilai MAE dengan rata-rata mandays aktual, sehingga memberikan gambaran proporsional mengenai besar *error* terhadap nilai sesungguhnya. Ringkasan *error* relatif ditunjukkan pada Tabel 11.

Tabel 11. *Relative Error* Terhadap Rata-rata Mandays Aktual

Model	MAE	% terhadap rata-rata mandays
LSTM	286.21	17.87%
Random Forest	369.57	23.07%
LSTM-RF	279.32	17.44%

Dari Tabel 11 dapat dilihat bahwa kombinasi LSTM-RF memiliki *error* relatif kombinasi terendah sebesar 17.44%, lebih kecil daripada LSTM maupun *Random Forest*. Nilai ini menunjukkan bahwa model kombinasi mampu menghasilkan estimasi mandays yang lebih proporsional terhadap kondisi sebenarnya. Dan dapat dikategorikan wajar untuk estimasi *effort* proyek TI yang memiliki karakteristik kompleks.

Namun demikian, perlu ditekankan bahwa MAE bersifat *absolut* sehingga tidak terdapat ambang baku universal untuk menilai kelayakannya. Oleh karena itu, evaluasi dilengkapi dengan metrik relatif seperti MRE, MMRE, dan PRED(25) yang umum digunakan dalam penelitian estimasi *effort* perangkat lunak. Metrik-metrik ini memberikan acuan kelayakan yang lebih obyektif (Ardiansyah dkk., 2018). Selain itu, mandays sendiri merupakan salah satu satuan *effort* yang dipakai dalam penelitian estimasi *effort* perangkat lunak (Popovic et al., 2015), sehingga relevan digunakan dalam penelitian ini.

1. MRE (*Mean Relative Error*) per data, mengukur kesalahan relatif antara nilai prediksi dan nilai aktual, yang dirumuskan pada persamaan (9).
   $$MMRE = \frac{|y_t - \hat{y}_t|}{y_t} \quad (9)$$
2. MMRE **(***Mean Magnitude of Relative Error***),** rata-rata kesalahan relatif yang dirumuskan pada persamaan (10). Menurut Khatibi & Bardsiri (2015), validasi model estimasi dengan batas MMRE ≤ 0.25 sebagai tolak ukur keberhasilan model.
   $$MMRE = \frac{1}{n}\sum_{t=1}^{n} MRE_i \quad (10)$$
3. PRED(25), dengan $MRE_i \leq 0.25\ or\ 25$, Menunjukkan persentase estimasi dengan kesalahan relatif kurang dari atau sama dengan 25%, yang berfungsi sebagai ambang batas yang diakui industri untuk akurasi praktis. Standar yang sering dipakai yaitu $PRED(25) \geq 75\%$, Dimana nilai tersebut sudah dianggap baik sebagai tolak ukur keberhasilan model (Jeklin, dkk., 2025; Gora & Sinha, 2023).
   $$PRED(25) = \left(\frac{1}{n}\sum_{t=1}^{n}(MRE_i \leq 0{,}25\ or\ 25)\right) \times 100\% \quad (11)$$

dengan :

n = Banyaknya data observasi
$\hat{y}_t$ = Nilai prediksi dari y
y_t = Nilai aktual atau data asli

Tabel 12. Evaluasi *Error* Metrik Relatif

Model	Total MRE	MMRE	PRED(25)
LSTM	24.93	21.50%	71.55%
Random Forest	24.92	21.49%	62.93%
LSTM-RF	20.48	17.65%	78.45%

Tabel 12 memperkuat hasil sebelumnya, di mana model kombinasi LSTM-RF konsisten lebih unggul dengan nilai MMRE paling rendah sebesar 17.65% berarti rata-rata kesalahan sekitar 17% dari nilai aktual mandays, sejalan dengan *error* relatif pada Tabel 10. Selain itu, PRED(25) sebesar 78.45%, menunjukkan bahwa hampir 8 dari 10 prediksi masuk kategori

error kecil $\leq 25\%$, melampaui standar kelayakan MMRE $< 25\%$ dan PRED(25) $\geq 75\%$ yang diakui dalam penelitian (Khatibi & Bardsiri, 2015; Gora & Sinha, 2023). Dengan demikian, meskipun MAE *Absolute* terlihat besar, secara relatif model kombinasi dapat dianggap layak dan praktis digunakan dalam estimasi *effort* proyek TI.

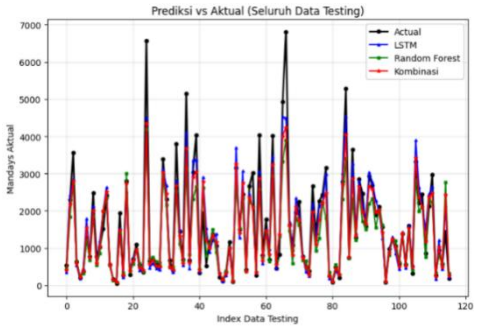
Selanjutnya, dilakukan analisis deviasi terhadap target mandays hasil perhitungan manual, dengan membagi total selisih mandays dan jumlah proyek tiap kategori seperti yang ditunjukkan pada Tabel 2. Hasil rata-rata selisih tiap kategori ditunjukkan pada Tabel 13.

Tabel 13. Rata-rata Selisih Mandays

Kategori	Rata-rata Selisih
Project	615
Maintenance	102
Internal	371

Berdasarkan Tabel 13, rata-rata selisih mandays terbesar terdapat pada kategori *project* sebesar 615 mandays. Meskipun nilainya cukup tinggi, hal ini dapat dimaklumi karena proyek biasanya melibatkan lingkup pekerjaan yang kompleks, variabel yang beragam, serta ketidakpastian yang lebih besar dibandingkan kategori lainnya. Pada kategori *internal*, selisih rata-rata tercatat sebesar 371 mandays, yang relatif lebih kecil dibandingkan *project*. Hal ini menunjukkan bahwa prediksi model lebih akurat dan stabil untuk pekerjaan *internal* karena ruang lingkupnya cenderung lebih terstruktur dan terstandarisasi. Sementara itu, kategori *maintenance* memiliki selisih terkecil yaitu 102 mandays. Namun, meskipun deviasinya kecil secara *absolut*, proporsi kesalahan terhadap total *effort* justru lebih signifikan, yang umumnya terjadi pada pekerjaan rutin, sehingga prediksi pada kategori ini masih memerlukan perbaikan. Dengan demikian, dapat disimpulkan bahwa model LSTM-RF sudah cukup baik dalam memprediksi kategori *project* dan *internal*, sementara akurasi pada kategori *maintenance* masih perlu ditingkatkan untuk menghasilkan estimasi yang lebih optimal.

Untuk memberikan Gambaran visual terhadap performa prediksi. Gambar 8 menyajikan perbandingan antara nilai aktual (garis hitam) dengan hasil prediksi tiga model, yaitu LSTM (biru), *Random Forest* (hijau), dan kombinasi LSTM-RF (merah). Terlihat bahwa LSTM dan *Random Forest* masih menunjukkan deviasi cukup besar pada beberapa titik, khususnya ketika terjadi lonjakan mandays yang tinggi. Sebaliknya, model kombinasi LSTM-RF mampu mengikuti pola aktual dengan lebih konsisten, baik pada puncak maupun lembah data. Sumbu horizontal merepresentasikan *index data testing* yang menunjukkan urutan data uji secara kronologis, sehingga grafik ini sekaligus memperlihatkan kemampuan model dalam menangkap dinamika data dari waktu ke waktu. Dengan demikian, kedekatan visual antara kurva kombinasi dan data aktual memperkuat temuan kuantitatif sebelumnya bahwa model kombinasi lebih stabil dan akurat dibandingkan model tunggal.



Gambar 8. Plot Prediksi Vs Aktual

SIMPULAN

Berdasarkan hasil penelitian, model tunggal LSTM memiliki keunggulan relatif dibandingkan *Random Forest* dengan nilai MAE lebih rendah sebesar 286.21 dan 369.57. sementara, pendekatan kombinasi LSTM-RF dengan metode *Error Reciprocal* terbukti menghasilkan performa terbaik secara keseluruhan. Model kombinasi ini mencatat MAE sebesar 279.32 dan MAPE 17.65%, yang berarti lebih akurat dibandingkan model tunggal, dengan peningkatan akurasi sekitar 4%. Evaluasi menggunakan MMRE ($< 25\%$) dan PRED(25) $\geq 75\%$ mengonfirmasi bahwa model ini memenuhi standar kelayakan dalam estimasi *effort* perangkat lunak.

Analisis per jenis proyek menunjukkan akurasi yang bervariasi, kategori *internal* paling akurat, *project* cukup baik, dan *maintenance* masih perlu ditingkatkan meski selisih *absolut* relatif kecil. Secara praktis, kondisi ini mengindikasikan bahwa proyek *maintenance* memiliki karakteristik pekerjaan yang lebih dinamis dan tidak sepenuhnya terencana, seperti perbaikan insidental, perubahan kebutuhan klien, serta tingkat ketergantungan yang tinggi terhadap kondisi sistem yang sedang berjalan. Oleh karena itu, penerapan model prediktif pada kategori *maintenance* memerlukan penyesuaian model atau penambahan variabel operasional agar mampu meningkatkan akurasi prediksi.

Dengan demikian, model LSTM-RF terbukti mampu mengatasi keterbatasan model tunggal dengan mengombinasikan kekuatan LSTM dan *Random Forest*. Dengan tingkat kesalahan relatif yang masih dalam batas wajar, model LSTM-RF dapat dianggap andal untuk memprediksi pencapaian target kerja khususnya mandays pada proyek TI, sehingga berpotensi mendukung pengambilan keputusan dalam perencanaan sumber daya dan manajemen proyek.

DAFTAR PUSTAKA

Abumohsen, M., Owda, A. Y., Owda, M., & Abumihsan, A. (2024). Hybrid machine learning model combining of CNN-LSTM-RF for time series forecasting of Solar Power Generation. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, *9*, 100636.

Amaliah, S., Nusrang, M., & Aswi, A. (2022). Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng. *VARIANSI: Journal of Statistics and Its application on Teaching and Research*, 4(3), 121-127.

Ardiansyah., Murein, M.M., & Handayaningsih, S. (2018). Analogy-based model for software project effort estimation. *International Journal of Advances in Intelligent Informatics*. 4(3), 251-260.

Ashari, M. L., & Sadikin, M. (2020). Prediksi Data Transaksi Penjualan Time Series Menggunakan Regresi LSTM. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, *9*(1), 1-10.

Cahyani, J., Mujahidin, S., & Fiqar, T. P. (2023). Implementasi Metode Long Short Term Memory (LSTM) untuk Memprediksi Harga Bahan Pokok Nasional. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 11(2), 346-357.

Cao, M., Mao, K., Bateni, S. M., Jun, C., Shi, J., Du, Y., & Du, G. (2023). Granulation-Based LSTM-RF Combination Model For Hourly Sea Surface Temperature Prediction. *International Journal of Digital Earth*, 16(1), 3838-3859.

Chen, J., Song, H., Wachowicz, M., & He, X. (2018). Deep Learning For Job Demand Forecasting. *Procedia Computer Science*, 130, 724-731.

Cholissodin, I., Sutrisno., Hasanah, U., Febiola, Y.I., & Soebroto, A. (2019). Buku Ajar AI, Machine Learning & Deep Learning. Malang : FILKOM.

Dachi, J. M. A. S., & Sitompul, P. (2023). Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam*, 2(2), 87-103.

Dessiaming, T. Z., Anraeni, S., & Pomalingo, S. (2022). College Academic Data Analysis Using Data Visualization. *Jurnal Teknik Informatika (Jutif)*, 3(5), 1203-1212.

Fachid, S., & Triayudi, A. (2022). Perbandingan Algoritma Regresi Linier dan Regresi Random Forest dalam Memprediksi Kasus Positif Covid-19. *Jurnal Media Informatika Budidarma*, 6(1), 68-73.

Farhanuddin., Sihombing, S. E. K., & Yahfizham (2024). Komparasi Multiple Linear Regression dan Random Forest Regression Dalam Memprediksi Anggaran Biaya Manajemen Proyek Sistem Informasi. *Journal of Computers and Digital Business*, *3*(2), 86-97.

Gora, R.K., & Sinha, R.R. (2023). A Studyof Evaluation Measures for Software Effort Estimation Using Machine Learning. *International Journal of Intelligent Systems and Aplications in Engineering (IJISAE)*, 11(6), 267-275.

Guangxing, L & Yihao, MA. (2024). Fault Diagnosis Of Electric Drill Winch Gearbox Based On LSTM-RF. *Journal of Vibration and Shock*, 43(21): 156-162

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-Based Stock Market Prediction Using A Diverse Set Of Variables. *Expert Systems with Applications*, 129, 273-285.

Indrayani, H. (2012). Penerapan Teknologi Informasi Dalam Peningkatan Efektivitas, Efisiensi Dan Produktivitas Perusahaan. *Jurnal El-Riyasah*, 3(1), 48-56.

Jeklin, U., Saad, M.I., & Ekawati, H. (2025). Evaluation Of COCOMO Model Accuracy In Software Effort Estimation. *Bulletin of Information Technology (BIT)*, 6(2), 126-135.

Khatibi, E., & Khatibi Bardsiri, V. (2015), Model To Estimate The Software Development Effort Based On In-Depth Analysis Of Project Attributes. *IET Softw*, 9, 109-118.

Kholifatul, B.A.H & Prihanto, A. (2023). Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech. *Journal of Informatics and Computer Science*, 4(3), 292-297.

LaRose, R., & Coyle, B. (2020). Robust Data Encodings For Quantum Classifiers. *Physical Review A*, 102(3), 032420.

Popovic, J., Bojic, D. & Korolija, N. (2015), Analysis Of Task Effort Estimation Accuracy Based On Use Case Point Size. *IET Softw*, 9, 166-173.

Ramadhan, R. F., & Ashari, W. M. (2024). Performance Comparison of Random Forest and Decision Tree Algorithms for Anomaly Detection in Networks. *Journal of Applied Informatics and Computing*, *8*(2), 367-375.

Ritonga, A., Saragih, E. L., Purba, G. A., Pandiangan, P. P. S., Damanik, R. N., & Al Azmi, F. (2025). Penerapan Distribusi Normal Dalam Pengukuran Tinggi Badan Mahasiswa FMIPA Universitas Negeri Medan 2024. *Bilangan: Jurnal Ilmiah Matematika, Kebumian dan Angkasa*, *3*(2), 39-53.

Sautomo, S., & Pardede, H. F. (2021). Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short-Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *5*(1), 99-106.

Steinki, O., & Mohammad, Z. (2015). Introduction To Ensemble Learning. *SSRN*.

Sudiantini, D., Naiwasha, A., Izzati, A., & Rindiani, C. (2023). Penggunaan Teknologi Pada Manajemen Sumber Daya Manusia Di Dalam Era Digital Sekarang. Digital Bisnis: *Jurnal Publikasi Ilmu Manajemen dan E-Commerce*, 2(2), 262-269.

Suryanto, Agus, dan Anan Nugroho. (2020). *Manajemen Proyek Teknologi Informasi*. Yogyakarta : Deepublish.

Vuong, P. H., Dat, T. T., Mai, T. K., & Uyen, P. H. (2022). Stock-price forecasting based on XGBoost and LSTM. *Computer Systems Science & Engineering*, 40(1).

Wardhana, I., Ariawijaya, M., Isnaini, V. A., & Wirman, R. P. (2022). Gradient Boosting Machine, Random Forest dan Light GBM untuk Klasifikasi Kacang Kering. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *6*(1), 92-99.