

# Memprediksi Keberhasilan Proyek ERP Menggunakan Indikator Kinerja Multi-Domain dan Machine Learning

Ahmad Khozi Waridi <sup>1\*</sup>, Muhammad Haikal Fikri <sup>2</sup>, Muhammad Ainul Yaqin <sup>3</sup>

<sup>1</sup> Universitas Islam Negeri Maulana Malik Ibrahim Malang; 230605110083@student.uin-malang.ac.id

<sup>2</sup> Universitas Islam Negeri Maulana Malik Ibrahim Malang; 230605110067@student.uin-malang.ac.id

<sup>3</sup> Universitas Islam Negeri Maulana Malik Ibrahim Malang; yaqinov@ti.uin-malang.ac.id

**Abstrak:** Penelitian ini bertujuan mengembangkan model prediksi keberhasilan implementasi Enterprise Resource Planning (ERP) menggunakan pendekatan Neural Network berbasis indikator kinerja multidomain. Tingginya tingkat kegagalan implementasi ERP dipengaruhi oleh kompleksitas proyek, ketidakseimbangan distribusi data, serta interaksi faktor teknis dan manajerial yang sulit dimodelkan menggunakan pendekatan konvensional. Penelitian ini memanfaatkan 4.000 data proyek ERP dengan 50 indikator kinerja yang dipetakan ke dalam delapan *Performance Domains* PMBOK Guide Edisi ke-7. Tahap *preprocessing* meliputi normalisasi data dan penerapan Synthetic Minority Over-sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan kelas. Model kemudian dioptimalkan melalui delapan skenario *hyperparameter* dan dievaluasi menggunakan sepuluh skenario pengujian untuk mengukur kemampuan generalisasi pada berbagai kondisi implementasi ERP. Hasil penelitian menunjukkan bahwa Neural Network memberikan performa terbaik dibandingkan algoritma pembandingan dan menghasilkan konfigurasi model yang mampu mempertahankan keseimbangan antara akurasi, kemampuan deteksi proyek berisiko tinggi, dan kompleksitas model. Pengujian lanjutan menunjukkan bahwa model memiliki performa yang stabil pada berbagai karakteristik proyek, meskipun sensitivitas terhadap kompleksitas proyek, metodologi pengembangan, dan karakteristik industri masih memengaruhi tingkat akurasi. Temuan ini menunjukkan bahwa integrasi indikator multidomain, optimasi *hyperparameter*, dan evaluasi berbasis berbagai skenario mampu menghasilkan model prediksi yang robust dan berpotensi diterapkan sebagai mekanisme *early warning system* untuk mendukung pengambilan keputusan pada implementasi ERP.

**Keywords:** Enterprise Resource Planning; Machine Learning; Neural Network; Performance Domains;

DOI: <https://doi.org/10.47134/jacis.v6i2.195>

\*Correspondensi: Ahmad Khozi Waridi

Email: 230605110083@student.uin-malang.ac.id

Receive: 29 Mei 2026

Accepted: 29 Juni 2026

Published: 3 Juli 2026



**Copyright:** © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

**Abstract:** This study aims to develop a predictive model for Enterprise Resource Planning (ERP) project success using a Neural Network approach based on multi-domain performance indicators. The high failure rate of ERP implementations is primarily attributed to project complexity, class imbalance, and the intricate interactions between technical and managerial factors, which are difficult to capture using conventional predictive approaches. The proposed model utilizes 4,000 ERP project records with 50 performance indicators mapped into the eight Performance Domains of the PMBOK Guide Seventh Edition. Data preprocessing included feature normalization and the application of the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance. The Neural Network model was optimized through eight hyperparameter optimization scenarios and subsequently evaluated using ten stress-testing scenarios to assess its generalization capability

under various ERP project conditions. The experimental results demonstrate that the proposed Neural Network outperformed the baseline machine learning algorithms and achieved a balanced trade-off between predictive accuracy, sensitivity to high-risk projects, and model complexity. Furthermore, the stress-testing evaluation confirmed that the optimized model maintained robust performance across diverse project characteristics, although prediction performance remained sensitive to project complexity, development methodology, and industry-specific characteristics. These findings indicate that integrating multi-domain performance indicators, hyperparameter optimization, and comprehensive scenario-based evaluation can produce a robust predictive model with strong potential to serve as an early warning system for supporting decision-making in ERP implementation projects

**Keywords:** Enterprise Resource Planning; Machine Learning; Neural Network; Performance Domains;

---

## PENDAHULUAN

*Enterprise Resource Planning* (ERP) telah menjadi tulang punggung operasi organisasi modern karena mampu mengintegrasikan berbagai fungsi bisnis, memperlancar aliran informasi antar departemen, serta meningkatkan efisiensi operasional dan kualitas pengambilan keputusan [1], [2], [3]. Meskipun demikian, implementasi ERP masih menghadapi tingkat kegagalan yang tinggi akibat kompleksitas proses, biaya implementasi yang besar, serta ketidaksesuaian antara kebutuhan organisasi dan karakteristik sistem ERP [2], [3], [5]. Faktor-faktor seperti kurangnya dukungan manajemen puncak, rendahnya keterlibatan pengguna, resistensi terhadap perubahan, serta lemahnya manajemen proyek menjadi penyebab dominan kegagalan implementasi ERP. Bahkan, hanya sebagian kecil proyek ERP yang berhasil diselesaikan tepat waktu dan sesuai anggaran [3].

Selain tantangan implementasi, sistem ERP konvensional juga menghadapi keterbatasan dalam mendukung kebutuhan bisnis yang semakin dinamis. ERP tradisional masih bergantung pada biaya lisensi dan infrastruktur yang tinggi, proses kustomisasi yang kompleks, serta analisis berbasis data historis yang kurang mampu menangkap pola non-linier dalam proses pengambilan keputusan [6], [7]. Oleh karena itu, integrasi teknologi seperti *Artificial Intelligence* (AI), *Machine Learning* (ML), dan *Big Data* menjadi kebutuhan penting untuk meningkatkan kemampuan prediksi, otomatisasi, dan optimalisasi proses bisnis [6], [8], [9].

Walaupun penerapan *machine learning* meningkatkan kapabilitas ERP, implementasinya juga menghadirkan tantangan baru, terutama berkaitan dengan kompleksitas algoritma, kualitas data, dimensi fitur yang tinggi, dan ketidakseimbangan kelas (*class imbalance*) pada data historis ERP [6], [7]. Kondisi ini menyebabkan model prediksi cenderung bias terhadap kelas mayoritas sehingga menurunkan kemampuan dalam mendeteksi kelas minoritas, seperti proyek ERP yang gagal [10]. Oleh karena itu, diperlukan pendekatan prediksi yang mampu mengatasi ketidakseimbangan data sekaligus memanfaatkan banyak indikator proyek secara efektif.

Berbagai penelitian telah memanfaatkan machine learning untuk memprediksi keberhasilan proyek ERP. *Support Vector Machine (SVM)*, *Neural Network (NN)*, *Random Forest*, dan *Long Short-Term Memory (LSTM)* telah menunjukkan performa yang baik pada berbagai studi [3], [13], [14], [15]. Selain itu, penelitian lain mengidentifikasi bahwa faktor teknis, proses bisnis, dan manajemen proyek merupakan penyebab utama keberhasilan maupun kegagalan implementasi ERP [16], [17], [18]. Namun, sebagian besar penelitian masih memiliki beberapa keterbatasan. Pertama, faktor teknis dan manajerial umumnya dianalisis secara terpisah sehingga belum mampu merepresentasikan kompleksitas proyek ERP secara menyeluruh. Kedua, jumlah fitur yang digunakan relatif sedikit sehingga belum menggambarkan interaksi multidimensi dalam implementasi ERP. Ketiga, sebagian besar penelitian belum secara khusus menangani permasalahan ketidakseimbangan data yang umum terjadi pada prediksi keberhasilan proyek.

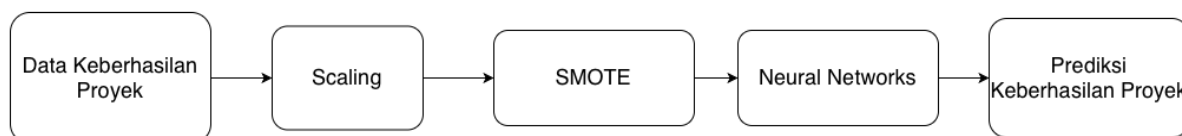
Berdasarkan hal tersebut, penelitian ini mengembangkan model prediksi keberhasilan proyek ERP berbasis *Neural Network* dengan mengintegrasikan 50 indikator kinerja yang dipetakan ke dalam delapan *Performance Domains* PMBOK *Guide* Edisi ke-7. Untuk meningkatkan kemampuan model dalam menangani data yang tidak seimbang, penelitian ini menerapkan teknik *Synthetic Minority Over-sampling Technique (SMOTE)* serta optimasi *hyperparameter*. Selain itu, model dievaluasi menggunakan 10 skenario *stress testing* untuk mengukur stabilitas, resiliensi, dan konsistensi performa pada berbagai kondisi data.

Kontribusi utama penelitian ini meliputi dua aspek. Dari sisi teoritis, penelitian ini menawarkan model prediksi berbasis sosio-teknis yang mengintegrasikan indikator teknis dan manajerial secara holistik dalam kerangka PMBOK *Guide* Edisi ke-7. Dari sisi praktis, penelitian ini menghasilkan mekanisme *early warning system* yang mampu mengidentifikasi potensi kegagalan proyek ERP secara lebih dini sehingga organisasi dapat melakukan mitigasi risiko sebelum proyek memasuki fase kritis.

## METODE

### Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode prediktif berbasis *machine learning* untuk membangun model klasifikasi tingkat keberhasilan implementasi proyek Enterprise Resource Planning (ERP). Model utama yang dikembangkan adalah *Neural Network (NN)* yang memanfaatkan 50 atribut proyek sebagai variabel prediktor. Tahapan penelitian meliputi pengumpulan dataset, *preprocessing*, pengembangan model, optimasi *hyperparameter*, evaluasi performa, serta pengujian lanjutan terhadap model terbaik. Alur penelitian secara keseluruhan ditunjukkan pada Gambar 1.



Gambar 1. Diagram desain penelitian

## Dataset dan Preprocessing Data

Penelitian menggunakan dataset publik yang diperoleh dari Kaggle dengan jumlah 4.000 data proyek ERP yang terdiri atas 49 fitur prediktor dan 1 label target. Seluruh data digunakan sebagai sampel penelitian (*total sampling*). Lima puluh atribut tersebut dipetakan ke dalam delapan Performance Domains pada PMBOK Guide Edisi ke-7, yaitu *Stakeholder, Team, Development Approach & Lifecycle, Planning, Project Work, Delivery, Measurement, dan Uncertainty*. Pemetaan ini bertujuan agar fitur-fitur yang digunakan mampu merepresentasikan faktor teknis maupun manajerial yang memengaruhi keberhasilan implementasi ERP. Tahap *preprocessing* dilakukan untuk meningkatkan kualitas data sebelum proses pelatihan model. Variabel kategorikal dikonversi menggunakan teknik *encoding*, sedangkan variabel numerik dinormalisasi menggunakan StandardScaler (Z-Score Normalization) sehingga seluruh fitur memiliki skala yang seragam. Analisis distribusi data menunjukkan adanya ketidakseimbangan kelas, terutama pada kategori Critical, sehingga diterapkan Synthetic Minority Over-sampling Technique (SMOTE) untuk menghasilkan distribusi kelas yang lebih seimbang. Setelah proses SMOTE, jumlah data meningkat dari 4.000 menjadi 5.584 sampel.

## Pengembangan Model Neural Network

Model prediksi dibangun menggunakan arsitektur Neural Network dengan fungsi aktivasi ReLU, optimizer Adam, *batch size* 32, maksimum 200 *epoch*, serta mekanisme *early stopping* untuk mencegah *overfitting*. Seluruh konfigurasi dasar tersebut dipertahankan pada setiap eksperimen sehingga perbedaan performa model hanya dipengaruhi oleh variasi *hyperparameter* yang diuji. Optimasi model dilakukan melalui delapan skenario eksperimen dengan memvariasikan learning rate, jumlah hidden layer, penggunaan SMOTE, serta rasio pembagian data latih dan data uji. Rincian konfigurasi masing-masing eksperimen disajikan pada Tabel 1.

**Tabel 1.** Konfigurasi Skenario Optimasi Hyperparameter Neural Network

Ekperimen	Tujuan	Learning Rate	Hidden Layer	Dropout	Data Latih : Data Uji	SMOTE
E1	Baseline	0,001	3	0,3 : 0,2	80 : 20	√
E2	Learning rate tinggi	0,010	3	0,3 : 0,2	80 : 20	√
E3	Learning rate rendah	0,0001	3	0,3 : 0,2	80 : 20	√
E4	Shallow Network	0,001	1	0,3 : 0,2	80 : 20	√
E5	Split 90 : 10	0,001	3	0,3 : 0,2	90 : 10	√
E6	Split 70 : 30	0,001	3	0,3 : 0,2	70 : 30	√
E7	Tanpa SMOTE	0,001	3	0,3 : 0,2	80 : 20	x
E8	Deep Network	0,0005	5	0,2; 0,15; 0,1	80 : 20	√

## Evaluasi Model

Performa setiap model dievaluasi menggunakan metrik Accuracy, Precision, Recall, dan F1-score yang dihitung berdasarkan *confusion matrix*. Konfigurasi terbaik kemudian dibandingkan dengan tiga algoritma pembandingan, yaitu Random Forest (RF), Support Vector Machine (SVM), dan Extreme Gradient Boosting (XGBoost)

menggunakan skema fixed holdout partition. Pendekatan ini dipilih agar seluruh algoritma diuji pada pembagian data yang sama sehingga hasil perbandingan dapat dilakukan secara objektif.

### Pengujian Lanjutan

Setelah konfigurasi terbaik diperoleh, model dievaluasi lebih lanjut melalui sepuluh skenario pengujian untuk mengukur kemampuan generalisasi pada berbagai kondisi implementasi ERP, meliputi: (1) proyek berisiko rendah, (2) proyek berisiko kritis, (3) proyek ideal, (4) proyek dengan kompleksitas tinggi, (5) pengaruh metodologi Agile dan Waterfall, (6) tekanan jadwal, (7) ketergantungan terhadap vendor, (8) proyek dengan riwayat risiko tinggi, (9) ketahanan terhadap data yang tidak lengkap (*data resiliency*), dan (10) konsistensi performa pada berbagai sektor industri. Pengujian ini bertujuan untuk memastikan bahwa model tidak hanya memiliki akurasi yang tinggi, tetapi juga mampu mempertahankan performa pada berbagai karakteristik proyek ERP.

## HASIL DAN PEMBAHASAN

### Perbandingan Model Baseline

Sebelum dilakukan optimasi *hyperparameter*, penelitian ini terlebih dahulu mengevaluasi performa model Neural Network (NN) sebagai model dasar dan membandingkannya dengan tiga algoritma klasifikasi yang banyak digunakan pada penelitian *machine learning*, yaitu Random Forest (RF), Support Vector Machine (SVM), dan Extreme Gradient Boosting (XGBoost). Seluruh algoritma dilatih menggunakan dataset hasil *preprocessing* dengan konfigurasi evaluasi yang sama sehingga perbedaan performa yang diperoleh mencerminkan kemampuan masing-masing algoritma dalam mempelajari pola hubungan antarvariabel pada proyek ERP. Perbandingan ini bertujuan untuk menentukan model dasar yang paling sesuai sebelum dilakukan optimasi *hyperparameter*.

Tabel .... Perbandingan Performa Model Baseline

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)	F1 Critical	F1 High	F1 Medium	F1 Low
MLP	0,83	0,83	0,83	0,83	0,92	0,76	0,75	0,89
Random Forest	0,69	0,68	0,68	0,68	0,82	0,55	0,55	0,80
SVM	0,82	0,82	0,82	0,82	0,91	0,75	0,74	0,88
XGBoost	0,72	0,73	0,72	0,72	0,86	0,60	0,63	0,80

Berdasarkan Tabel ...., model Neural Network menunjukkan performa terbaik dibandingkan ketiga algoritma pembanding. Model ini memperoleh accuracy sebesar 83%, lebih tinggi dibandingkan Support Vector Machine yang mencapai 82%, serta secara signifikan melampaui XGBoost (72%) dan Random Forest (69%). Selain menghasilkan akurasi tertinggi, Neural Network juga memperoleh nilai precision, recall, dan F1-score macro sebesar 0,83, yang menunjukkan kemampuan klasifikasi yang konsisten pada seluruh kategori risiko proyek ERP.

Evaluasi pada masing-masing kelas risiko menunjukkan bahwa Neural Network mempertahankan performa yang relatif seimbang. Nilai F1-score tertinggi diperoleh pada kelas Critical sebesar 0,92, diikuti kelas Low sebesar 0,89. Sementara itu, kelas High dan Medium masing-masing memperoleh F1-score sebesar 0,76 dan 0,75, sedikit lebih tinggi dibandingkan SVM serta jauh lebih baik dibandingkan Random Forest dan XGBoost. Hasil ini menunjukkan bahwa Neural Network memiliki kemampuan yang lebih baik dalam membedakan kelas-kelas yang memiliki karakteristik saling beririsan, yang umumnya menjadi tantangan pada permasalahan klasifikasi multikelas.

Keunggulan tersebut mengindikasikan bahwa hubungan antarvariabel pada delapan Performance Domains PMBOK Guide Edisi ke-7 bersifat kompleks dan tidak sepenuhnya linier. Neural Network mampu memodelkan hubungan nonlinier antarfitur secara lebih efektif sehingga menghasilkan performa klasifikasi yang lebih stabil dibandingkan algoritma pembanding. **Temuan ini sejalan dengan berbagai penelitian sebelumnya yang menyatakan bahwa arsitektur jaringan saraf memiliki kemampuan yang baik dalam mempelajari hubungan antarvariabel yang kompleks, terutama pada data berdimensi tinggi yang mengandung interaksi antarfitur.**

Meskipun selisih akurasi antara Neural Network dan SVM relatif kecil, Neural Network menunjukkan konsistensi yang lebih baik pada seluruh metrik evaluasi, khususnya dalam mengidentifikasi kelas Critical yang menjadi fokus utama penelitian. Dalam konteks prediksi keberhasilan implementasi ERP, kemampuan mengenali proyek dengan tingkat risiko tinggi lebih penting dibandingkan sekadar memperoleh akurasi keseluruhan yang tinggi, karena kesalahan klasifikasi pada kategori tersebut dapat menyebabkan keterlambatan tindakan mitigasi terhadap potensi kegagalan proyek.

Berdasarkan hasil tersebut, Neural Network dipilih sebagai model utama untuk tahap optimasi *hyperparameter*. Pemilihan ini tidak hanya didasarkan pada nilai akurasi tertinggi, tetapi juga mempertimbangkan keseimbangan performa pada seluruh metrik evaluasi serta kemampuannya dalam mengenali proyek dengan tingkat risiko tinggi. Selanjutnya, model ini dioptimalkan melalui delapan skenario konfigurasi *hyperparameter*

### Optimasi Hyperparameter Neural Network

Setelah Neural Network ditetapkan sebagai model utama berdasarkan hasil perbandingan pada Subbab 4.1, tahap selanjutnya adalah melakukan optimasi *hyperparameter* untuk memperoleh konfigurasi model yang memberikan keseimbangan terbaik antara akurasi, kemampuan generalisasi, dan sensitivitas terhadap proyek berisiko tinggi. Proses optimasi dilakukan melalui delapan skenario eksperimen sebagaimana dijelaskan pada Tabel 3, dengan memvariasikan *learning rate*, jumlah *hidden layer*, rasio pembagian data latih dan data uji, serta penggunaan Synthetic Minority Over-sampling Technique (SMOTE). Hasil evaluasi masing-masing konfigurasi disajikan pada Tabel ...

Tabel .... Perbandingan Performa Setiap Konfigurasi Hyperparameter

Eks Perimen	Lear-ning rate	Hidden layer	Split data	SMOTE	Accuracy	Precision (macro)	Recall (macro)	F1 Score (macro)	Recall Critical
E1	0,001	3	80:20	√	0,83	0,82	0,82	0,82	0,93
E2	0,010	3	80:20	√	0,81	0,81	0,81	0,81	0,91
E3	0,0001	3	80:20	√	0,83	0,83	0,83	0,83	0,90

E4	0,001	1	80:20	√	0,84	0,83	0,83	0,83	0,90
E5	0,001	3	90:10	√	0,82	0,82	0,82	0,82	0,88
E6	0,001	3	70:30	√	0,81	0,82	0,81	0,81	0,85
E7	0,001	3	80:20	x	0,73	0,74	0,73	0,74	0,76
E8	0,0005	5	80:20	√	0,84	0,83	0,84	0,83	0,91

Berdasarkan Tabel ..., maka perubahan *learning rate* memberikan pengaruh terhadap proses pembelajaran model. Konfigurasi dengan *learning rate* lebih besar (E2) menghasilkan accuracy sebesar 81%, sedangkan penggunaan *learning rate* yang lebih kecil (E3) meningkatkan akurasi menjadi 83%. Hasil ini menunjukkan bahwa nilai *learning rate* yang terlalu besar menyebabkan proses pembaruan bobot berlangsung terlalu agresif sehingga model kurang mampu mencapai titik konvergensi yang optimal. Sebaliknya, *learning rate* yang lebih kecil menghasilkan proses pembelajaran yang lebih stabil sehingga representasi pola pada dataset ERP dapat dipelajari secara lebih baik.

Selain *learning rate*, variasi jumlah hidden layer juga memberikan pengaruh terhadap performa model. Konfigurasi satu hidden layer (E4) maupun lima hidden layer (E8) menghasilkan akurasi tertinggi sebesar 84%, sedangkan konfigurasi tiga hidden layer (E1) memperoleh akurasi 83%. Meskipun demikian, selisih performa tersebut relatif kecil sehingga peningkatan kompleksitas arsitektur tidak memberikan peningkatan akurasi yang signifikan. Temuan ini menunjukkan bahwa pada dataset ERP yang digunakan, hubungan antarfitur telah dapat dipelajari secara efektif menggunakan arsitektur yang lebih sederhana tanpa memerlukan jaringan yang lebih dalam.

Untuk pengaruh penggunaan SMOTE terlihat yang paling signifikan dibandingkan variasi *hyperparameter* lainnya. Hal ini ditunjukkan oleh perbandingan antara E1 dan E7, yang menggunakan konfigurasi serupa namun berbeda pada penerapan SMOTE. Ketika SMOTE tidak digunakan, akurasi model turun dari 83% menjadi 73%, sedangkan F1-score macro menurun dari 0,82 menjadi 0,74. Penurunan tersebut menunjukkan bahwa ketidakseimbangan distribusi kelas menyebabkan model lebih cenderung mempelajari kelas mayoritas sehingga kemampuan mengenali proyek pada kategori risiko tinggi menjadi berkurang. Dengan demikian, penerapan SMOTE terbukti berkontribusi dalam meningkatkan keseimbangan performa model pada seluruh kelas. Sedangkan perubahan rasio pembagian data latih dan data uji pada E5 (90:10) dan E6 (70:30) memberikan pengaruh terhadap performa model, meskipun tidak sebesar variasi *learning rate* maupun penggunaan SMOTE. Perbedaan nilai akurasi yang relatif kecil menunjukkan bahwa model memiliki tingkat stabilitas yang baik terhadap perubahan proporsi data pelatihan. Temuan ini mengindikasikan bahwa kualitas representasi fitur yang dipelajari model lebih dipengaruhi oleh konfigurasi *hyperparameter* dan distribusi data dibandingkan oleh perubahan rasio pembagian data.

Meskipun E4 dan E8 memperoleh accuracy tertinggi sebesar 84%, penelitian ini menetapkan E1 sebagai konfigurasi terbaik. Keputusan tersebut tidak hanya mempertimbangkan akurasi keseluruhan, tetapi juga kemampuan model dalam mendeteksi proyek yang berpotensi mengalami kegagalan. Sebagaimana ditunjukkan pada Tabel ..., E1 menghasilkan Recall Critical sebesar 0,93, lebih tinggi dibandingkan konfigurasi lainnya. Nilai *recall* yang tinggi pada kelas Critical menunjukkan bahwa model mampu mengenali sebagian besar proyek dengan tingkat risiko tertinggi sehingga meminimalkan kemungkinan proyek kritis diklasifikasikan sebagai proyek dengan risiko yang lebih rendah.

Dalam konteks implementasi ERP, kesalahan mendeteksi proyek berisiko tinggi memiliki konsekuensi yang lebih besar dibandingkan sedikit penurunan akurasi secara keseluruhan. Oleh karena itu, pemilihan model terbaik pada penelitian ini didasarkan pada kemampuan menghasilkan keseimbangan antara akurasi, kemampuan generalisasi, sensitivitas terhadap kelas Critical, serta kompleksitas arsitektur yang tetap efisien. Berdasarkan pertimbangan tersebut, E1 dipilih sebagai konfigurasi akhir yang selanjutnya digunakan pada tahap pengujian lanjutan terhadap berbagai skenario implementasi proyek ERP.

### Evaluasi Model Terbaik pada Berbagai Skenario Proyek ERP

Setelah konfigurasi E1 ditetapkan sebagai model terbaik melalui proses optimasi *hyperparameter*, penelitian dilanjutkan dengan evaluasi pada sepuluh skenario yang dirancang untuk menguji kemampuan generalisasi model pada berbagai kondisi implementasi ERP. Pengujian ini bertujuan untuk memastikan bahwa model tidak hanya memiliki performa yang baik pada data evaluasi, tetapi juga mampu mempertahankan konsistensi prediksi ketika dihadapkan pada karakteristik proyek yang berbeda. Ringkasan hasil pengujian disajikan pada Tabel ....

**Tabel ....** Ringkasan Hasil Pengujian Model pada Berbagai Skenario Proyek ERP

Skenario	Fokus pengujian	Hasil utama	Implikasi
Risiko Rendah	Keandalan pada proyek Low Risk	Akurasi sangat tinggi	Model stabil pada proyek homogen
Risiko Kritis	Deteksi proyek Critical	Recall kelas Critical tetap tinggi	Mendukung fungsi early warning
Kondisi Ideal	Tim senior, stakeholder tinggi	Performa paling konsisten	Model efektif pada kondisi optimal
Kompleksitas Tinggi	Kompleksitas tinggi + tim junior	Performa menurun	Kompleksitas meningkatkan kesulitan prediksi
Agile vs Waterfall	Bias metodologi	Agile lebih stabil dibanding Waterfall	Karakteristik metodologi memengaruhi prediksi
Tekanan Jadwal	Deadline ketat	Sedikit penurunan akurasi	Jadwal menjadi faktor risiko
Ketergantungan Vendor	Vendor tidak andal	Prediksi lebih bervariasi	Faktor eksternal memengaruhi hasil
Riwayat Risiko	Legacy risk	Model tetap mengenali pola kegagalan	Riwayat proyek berkontribusi terhadap prediksi
Data Tidak Lengkap	Missing value	Akurasi menurun namun tetap stabil	Model cukup resilien terhadap data tidak lengkap
Lintas Industri	Generalisasi antar sektor	Performa berbeda antar industri	Diperlukan adaptasi pada domain tertentu

Hasil pengujian menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan proyek berdasarkan tingkat risiko. Pada proyek berisiko rendah maupun kondisi ideal, model menghasilkan prediksi yang konsisten karena karakteristik data relatif seragam dengan pola yang dipelajari selama proses pelatihan. Sementara itu, pada proyek dengan kategori Critical, model tetap mampu mempertahankan kemampuan deteksi terhadap proyek berpotensi gagal. Temuan ini menunjukkan bahwa konfigurasi hasil

optimasi tidak hanya meningkatkan performa secara keseluruhan, tetapi juga mempertahankan sensitivitas terhadap kelas risiko yang menjadi fokus utama penelitian.

Selain itu performa model mulai mengalami penurunan ketika diterapkan pada proyek dengan kompleksitas tinggi, tekanan jadwal yang ketat, maupun ketergantungan yang besar terhadap vendor eksternal. Kondisi tersebut menunjukkan bahwa semakin kompleks karakteristik proyek, semakin sulit model membedakan batas antar kelas risiko. Meskipun demikian, penurunan performa masih berada pada tingkat yang dapat diterima, sehingga mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik terhadap variasi kondisi operasional implementasi ERP.

Pengujian terhadap metodologi pengembangan menunjukkan bahwa model tetap mampu menghasilkan prediksi yang konsisten pada proyek berbasis Agile maupun Waterfall, meskipun terdapat sedikit perbedaan performa akibat karakteristik masing-masing pendekatan. Selain itu, model masih mampu mempertahankan akurasi yang relatif stabil ketika dihadapkan pada data yang tidak lengkap setelah melalui tahapan *preprocessing* dan penerapan SMOTE. Variasi performa pada berbagai sektor industri menunjukkan bahwa karakteristik domain bisnis tetap memengaruhi hasil prediksi, sehingga pengembangan model yang lebih spesifik terhadap sektor industri tertentu masih menjadi peluang penelitian selanjutnya.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa model tidak hanya memiliki performa klasifikasi yang baik, tetapi juga mampu mempertahankan konsistensi prediksi pada berbagai karakteristik implementasi ERP. Pengujian terhadap variasi tingkat risiko, kompleksitas proyek, metodologi pengembangan, kualitas data, serta sektor industri membuktikan bahwa model memiliki tingkat robustness dan kemampuan generalisasi yang memadai. Hasil tersebut memperkuat bahwa model yang dikembangkan layak digunakan sebagai dasar prediksi pada berbagai kondisi implementasi ERP sebelum diterapkan pada lingkungan operasional yang sebenarnya.

### **Perbandingan dengan Penelitian Terdahulu dan Implikasi Penelitian**

Hasil penelitian ini memperkuat temuan berbagai penelitian sebelumnya yang menunjukkan bahwa pendekatan *machine learning* mampu meningkatkan akurasi prediksi keberhasilan proyek. Penelitian oleh Setiawan dan Rahman [13] menunjukkan bahwa Support Vector Machine (SVM) dan Neural Network memberikan performa yang baik pada prediksi keberhasilan proyek perangkat lunak, sedangkan Moravej dkk. [14] melaporkan bahwa Random Forest menghasilkan nilai *precision* dan *recall* yang lebih baik dibandingkan beberapa algoritma pembanding pada prediksi keberhasilan proyek. Pada penelitian ini, Neural Network juga menunjukkan performa terbaik dibandingkan Random Forest, SVM, dan XGBoost pada tahap evaluasi model *baseline*. Hasil tersebut mengindikasikan bahwa Neural Network memiliki kemampuan yang lebih baik dalam mempelajari hubungan nonlinier antarvariabel pada implementasi proyek ERP yang melibatkan interaksi kompleks antara aspek teknis dan manajerial.

Meskipun demikian, kontribusi penelitian ini tidak hanya terletak pada pemilihan algoritma klasifikasi. Berbeda dengan penelitian Setiawan dan Rahman [13] yang menggunakan 28 fitur pada proyek perangkat lunak B2C, maupun penelitian Krvavac dan Durmić [3] yang menggunakan jumlah atribut relatif terbatas pada prediksi kegagalan proyek ERP, penelitian

ini mengintegrasikan 50 indikator kinerja yang dipetakan ke dalam delapan Performance Domains PMBOK Guide Edisi ke-7. Pendekatan tersebut memungkinkan model merepresentasikan hubungan antara faktor teknis, organisasi, tim proyek, pemangku kepentingan, hingga ketidakpastian proyek secara lebih komprehensif dibandingkan pendekatan yang hanya berfokus pada sebagian indikator keberhasilan.

Perbedaan lain juga terlihat pada strategi optimasi model. Sebagian besar penelitian terdahulu berfokus pada perbandingan algoritma atau peningkatan akurasi menggunakan satu konfigurasi model [13], [14], sedangkan penelitian ini menerapkan optimasi *hyperparameter* yang dipadukan dengan SMOTE untuk mengatasi ketidakseimbangan kelas pada data historis ERP. Hasil penelitian menunjukkan bahwa penerapan SMOTE memberikan kontribusi nyata terhadap peningkatan kemampuan model dalam mengenali proyek berisiko tinggi. Temuan ini sejalan dengan penelitian Gao dkk. [7] yang menyatakan bahwa penanganan *class imbalance* merupakan faktor penting dalam meningkatkan performa model pada dataset berdimensi tinggi.

Dari sisi evaluasi, penelitian ini juga menawarkan pendekatan yang lebih komprehensif dibandingkan penelitian sebelumnya. Nishat dkk. [15] memanfaatkan pendekatan *deep learning* berbasis Long Short-Term Memory (LSTM) untuk prediksi *variation orders* pada proyek konstruksi, sedangkan penelitian ini mengevaluasi model melalui sepuluh skenario pengujian yang mencakup variasi tingkat risiko, kompleksitas proyek, metodologi pengembangan, kualitas data, ketergantungan terhadap vendor, serta karakteristik sektor industri. Pendekatan tersebut memungkinkan evaluasi tidak hanya berfokus pada akurasi global, tetapi juga pada kemampuan generalisasi model ketika diterapkan pada berbagai kondisi implementasi ERP. Dengan demikian, model yang dihasilkan lebih representatif terhadap kondisi operasional yang dihadapi organisasi.

Selain memberikan kontribusi metodologis, penelitian ini juga memiliki implikasi praktis bagi organisasi yang mengimplementasikan ERP. Model yang dikembangkan berpotensi dimanfaatkan sebagai mekanisme *early warning system* untuk mengidentifikasi proyek yang berpotensi mengalami kegagalan sejak tahap awal implementasi. Informasi tersebut memungkinkan manajer proyek melakukan tindakan mitigasi terhadap faktor-faktor risiko sebelum proyek memasuki fase kritis, sehingga potensi keterlambatan, pembengkakan biaya, maupun kegagalan implementasi dapat diminimalkan. Pendekatan ini sejalan dengan kebutuhan transformasi ERP modern yang tidak hanya berfungsi sebagai sistem operasional, tetapi juga sebagai platform pendukung pengambilan keputusan berbasis data.

Meskipun memberikan hasil yang menjanjikan, penelitian ini masih memiliki beberapa keterbatasan. Dataset yang digunakan berasal dari sumber publik sehingga belum sepenuhnya merepresentasikan karakteristik implementasi ERP pada sektor industri tertentu. Selain itu, penelitian ini masih berfokus pada satu arsitektur Neural Network, sehingga pengembangan model menggunakan arsitektur *deep learning* lainnya atau pengujian pada dataset organisasi nyata masih menjadi peluang penelitian selanjutnya. Pengembangan tersebut diharapkan dapat meningkatkan kemampuan generalisasi model sekaligus memperluas penerapannya pada berbagai konteks implementasi ERP.

## SIMPULAN

Penelitian ini berhasil mengembangkan model prediksi keberhasilan implementasi Enterprise Resource Planning (ERP) berbasis Neural Network dengan memanfaatkan 50 indikator kinerja yang dipetakan ke dalam delapan Performance Domains PMBOK Guide Edisi ke-7. Hasil penelitian menunjukkan bahwa Neural Network memberikan performa terbaik dibandingkan Random Forest, Support Vector Machine (SVM), dan XGBoost. Melalui proses optimasi *hyperparameter*, konfigurasi E1 dipilih sebagai model terbaik karena memberikan keseimbangan antara akurasi, kemampuan generalisasi, dan sensitivitas dalam mendeteksi proyek pada kategori Critical, sehingga lebih sesuai untuk mendukung prediksi risiko implementasi ERP.

Pengujian lanjutan pada sepuluh skenario proyek menunjukkan bahwa model mampu mempertahankan performa yang stabil pada berbagai kondisi implementasi, termasuk variasi tingkat risiko, kompleksitas proyek, metodologi pengembangan, kualitas data, dan karakteristik industri. Temuan ini menunjukkan bahwa kombinasi Neural Network, SMOTE, dan pemetaan indikator ke dalam Performance Domains PMBOK Guide Edisi ke-7 mampu menghasilkan model prediksi yang tidak hanya akurat, tetapi juga memiliki kemampuan generalisasi yang baik sehingga berpotensi diterapkan sebagai mekanisme *early warning system* untuk mendukung pengambilan keputusan pada implementasi ERP.

Penelitian ini masih memiliki keterbatasan karena menggunakan dataset publik dan hanya mengevaluasi satu arsitektur Neural Network. Oleh karena itu, penelitian selanjutnya disarankan menggunakan dataset implementasi ERP dari organisasi nyata, mengeksplorasi arsitektur *deep learning* lainnya, serta melakukan validasi pada berbagai sektor industri untuk meningkatkan kemampuan generalisasi dan penerapan model dalam lingkungan operasional yang lebih luas.

## DAFTAR PUSTAKA

- [1] K. Al-Fawaz, Z. Al-Salti, dan T. Eldabi, "Critical Success Factors in Erp Implementation: A Review," 2008.
- [2] Niel Agrisman Barus, Iskandar Muda, dan Sambas Ade Kesuma, "A Systematic Review of the DeLone & McLean Model in Enterprise Resource Planning (ERP) Systems Success," *JMAA*, vol. 21, no. 3, Jun 2025, doi: 10.17265/1548-6583/2025.03.002.
- [3] E. Krvavac dan N. Durmić, "Erp Project Failure Prediction using Machine Learning Algorithms," *International Journal of Innovative Science and Research Technology (IJISRT)*, hlm. 2247–2257, Mei 2025, doi: 10.38124/ijisrt/25may1435.
- [4] R. Basu, P. Upadhyay, M. C. Das, dan P. K. Dan, "An approach to identify issues affecting ERP implementation in Indian SMEs," *JIEM*, vol. 5, no. 1, hlm. 133–154, Jun 2012, doi: 10.3926/jiem.416.
- [5] M.-I. Mahraz, "Success Factors for ERP Implementation: a Systematic Literature Review," 2019.
- [6] Z. N. Jawad dan V. Balázs, "Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review," *Beni-Suef Univ J Basic Appl Sci*, vol. 13, no. 1, hlm. 4, Jan 2024, doi: 10.1186/s43088-023-00460-y.

- [7] R. Gao, S. Cui, Y. Wang, dan W. Xu, "Predicting financial distress in high-dimensional imbalanced datasets: a multi-heterogeneous self-paced ensemble learning framework," *Financ Innov*, vol. 11, no. 1, hlm. 50, Jan 2025, doi: 10.1186/s40854-024-00745-w.
- [8] M. Kukulagunta, "Critical Success Factors in Enterprise Resource Planning (ERP) Implementation: A Comprehensive Review," *IJSET*, vol. 12, no. 6, hlm. 1–15, Nov 2024, doi: 10.61463/ijset.vol.12.issue6.335.
- [9] Md. Al-Amin, Md. T. Hossain, Md. J. Islam, dan S. Kumar Biwas, "History, Features, Challenges, and Critical Success Factors of Enterprise Resource Planning (ERP) in The Era of Industry 4.0," *ESJ*, vol. 19, no. 6, hlm. 31, Feb 2023, doi: 10.19044/esj.2023.v19n6p31.
- [10] A. Luz dan J. Oloyede, "Impact of Dataset Imbalance on Machine Learning Models for Diabetes Mellitus Prediction," 23 Januari 2025, *Public Health and Healthcare*. doi: 10.20944/preprints202501.1684.v1.
- [11] Fangfang Lee, "What Is a Neural Network? | IBM," What is a neural network? Diakses: 1 April 2026. [Daring]. Tersedia pada: <https://www.ibm.com/think/topics/neural-networks>
- [12] Meegle, "Neural Network Vs Biological Neural Networks." Diakses: 1 April 2026. [Daring]. Tersedia pada: [https://www.meegle.com/en\\_us/topics/neural-networks/neural-network-vs-biological-neural-networks](https://www.meegle.com/en_us/topics/neural-networks/neural-network-vs-biological-neural-networks)
- [13] R. Setiawan dan T. K. A. Rahman, "Machine learning-based B2C software project success prediction model in Indonesia," *Int. J. Adv. Intell. Informatics*, vol. 11, no. 3, hlm. 480, Agu 2025, doi: 10.26555/ijain.v11i3.2123.
- [14] K. Moravej, S. L. Laroche, dan S. Musick, "Machine Learning Model Prediction of Project Success," dipresentasikan pada 2024 European Conference on Computing in Construction, Jul 2024. doi: 10.35490/EC3.2024.196.
- [15] M. M. Nishat, A. Ahsan, dan N. O. E. Olsson, "Applying Machine Learning for Predictive Analysis in Project-Based Data: Insights into Variation Orders," *ITcon*, vol. 30, hlm. 807–825, Mei 2025, doi: 10.36680/j.itcon.2025.033.
- [16] Sunil Kumar Mishra, "Improving ERP Adoption Through Predictive Modeling: A Data-Driven Recommendation System," *jisem*, vol. 10, no. 23s, hlm. 852–877, Mar 2025, doi: 10.52783/jisem.v10i23s.3786.
- [17] M. Z. Alvaro, M. A. Darunnaja, dan M. A. Yaqin, "Black Box Testing on a Mini ERP System: A Strategy for Bug Detection and Minimisation," *Journal of Computing and Data Science (JOC DAS)*, vol. 2, no. 1, hlm. 19–31, 2024, <https://doi.org/10.18860/jocdas.v2i1.28059>.
- [18] Larasati, S. D., Eitiveni, I., & Mahardhika, P, "Analysis of ERP Critical Failure Factors: A Case Study in an Indonesian Mining Company," *Jurnal Sistem Informasi*, 19(2), 34–47. <https://doi.org/10.21609/jsi.v19i2.1291>.