OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

Implementasi *Random Forest* untuk Klasifikasi Laporan Teknisi dan Administrasi Berbasis *Web* di PT Maxkomp

Harry Iskandar[1*], Erlangga Triantara Putra[1]

[1]Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia
Email: [1*]harryiskandar025@gmail.com, [2]dosen01881@unpam.ac.id
(* : coressponding author)

Abstrak−Pengelolaan laporan teknisi dan administrasi di PT Maxkomp masih menghadapi kendala berupa berkas laporan yang tersebar, format pencatatan yang beragam, data duplikat, nilai tanggal yang tidak seragam, serta proses pemeriksaan manual yang menyulitkan penentuan status penyelesaian pekerjaan. Penelitian ini bertujuan mengimplementasikan metode *Random Forest* dalam aplikasi berbasis *web* untuk mengolah dan mengklasifikasikan laporan teknisi serta administrasi ke dalam kelas SELESAI dan TINDAK_LANJUT sebagai rekomendasi pendukung pemeriksaan laporan. *Dataset* awal terdiri atas 3.000 *record* yang mencakup laporan nyata dan data sintetis. Setelah tahap pembersihan dan pengecualian *target* kosong, diperoleh 2.917 kandidat pemodelan yang dibagi pada eksperimen utama dengan rasio 75:25 menjadi 2.187 data latih dan 730 data uji. Metode *Random Forest* diterapkan melalui *pipeline* yang menggabungkan fitur numerik, kategori, teks berbasis *Term Frequency-Inverse Document Frequency* (*TF-IDF*), dan komponen tanggal. Hasil evaluasi pada data uji campuran menunjukkan *accuracy* sebesar 99,18%, *balanced accuracy* 98,79%, *macro F1-score* 99,03%, dan *recall* TINDAK_LANJUT 97,77%, sejalan dengan pembanding RapidMiner yang memperoleh *accuracy* 99,18% dan *recall* 97,32%. Namun, penguraian menurut asal data memperlihatkan bahwa *recall* TINDAK_LANJUT pada laporan nyata baru mencapai 16,67%, sedangkan data sintetis mencapai 100%. Aplikasi telah lulus 72 skenario pengujian *black box* dan 26 kasus *white box*, sehingga layak digunakan sebagai pemberi rekomendasi dengan keputusan akhir tetap memerlukan verifikasi manusia.

Kata Kunci: *Random Forest*, Klasifikasi Laporan, Aplikasi Berbasis *Web*, Laporan Teknisi, PT Maxkomp

Abstract−*Managing technician and administrative reports at PT Maxkomp faces challenges including scattered report files, diverse recording formats, duplicate records, inconsistent date values, and manual review processes that complicate determining work completion status. This study implements the Random Forest method within a web-based application to process and classify technician and administrative reports into COMPLETED (SELESAI) and FOLLOW_UP (TINDAK_LANJUT) classes as recommendations supporting report review. The initial dataset consists of 3,000 records comprising real reports and synthetic data. Following data cleaning and exclusion of missing targets, 2,917 modeling candidates were obtained and split at a 75:25 ratio into 2,187 training records and 730 testing records for the main experiment. Random Forest was applied through a pipeline combining numerical, categorical, Term Frequency-Inverse Document Frequency (TF-IDF) textual, and date component features. Evaluation on the mixed test set achieved 99.18% accuracy, 98.79% balanced accuracy, 99.03% macro F1-score, and 97.77% recall for the follow-up class, aligning with RapidMiner benchmarking at 99.18% accuracy and 97.32% recall. However, origin-based breakdown revealed that follow-up recall on real reports reached only 16.67%, compared to 100% on synthetic data. The application passed 72 black box scenarios and 26 white box test cases, demonstrating its suitability for presenting recommendations while final decisions remain under human verification.*

Keywords: *Random Forest, Report Classification, Web-Based System, Technician Reports, PT Maxkomp*

1. PENDAHULUAN

Kegiatan dukungan dan pemeliharaan teknologi informasi menghasilkan laporan pekerjaan yang memuat keluhan pelanggan, tindakan penanganan, serta hasil pekerjaan teknisi. Informasi tersebut diperlukan untuk menentukan apakah suatu laporan pekerjaan sudah dapat ditutup atau masih membutuhkan tindak lanjut teknis maupun administratif. PT Maxkomp, perusahaan layanan teknologi informasi dan teknisi perangkat jaringan serta komputer yang berdiri sejak tahun 1998, menghadapi tantangan pengelolaan laporan seiring bertambahnya jumlah pelanggan dan lokasi kerja lapangan. Pencatatan yang masih dilakukan secara manual pada berkas terpisah memicu keterlambatan pelaporan, ketidaksesuaian format, baris kosong, duplikasi data, ketidakkonsistenan tanggal, serta risiko *human error* dalam *monitoring* pekerjaan secara *real-time*. Selain masalah

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

kualitas data, catatan laporan memuat kombinasi atribut yang berbeda, yaitu teks bebas untuk keluhan dan solusi, data kategori untuk identitas pelanggan, serta data numerik dan tanggal, sehingga membutuhkan penyiapan data sebelum dapat dianalisis secara konsisten.

Pendekatan *data mining* dan *machine learning* berjenis *supervised learning* banyak dimanfaatkan untuk mengolah data layanan menjadi model klasifikasi otomatis. (Anjani et al., 2023) menerapkan *Random Forest* disertai prapemrosesan teks, pembobotan *Term Frequency-Inverse Document Frequency* (*TF-IDF*), dan *Synthetic Minority Oversampling Technique* (*SMOTE*) pada klasifikasi ulasan aplikasi dengan capaian *accuracy* hingga 98,9%. Pada domain layanan teknologi informasi, (Oliveira et al., 2022) membandingkan beberapa algoritma termasuk *LinearSVC* dan *Random Forest* untuk mengklasifikasikan tiket insiden berbahasa Portugis, sedangkan (Crosley & Wasito, 2023) memadukan *IndoBERT* dan *Random Forest* guna menentukan prioritas tiket dukungan teknologi informasi. Selain itu, (Aini et al., 2024), (Utomo & Zakaria, 2026), serta (Yuda et al., 2026) menunjukkan bahwa alur pengolahan data terstruktur menggunakan *Random Forest* mampu memetakan beragam atribut kategorikal dan numerik untuk klasifikasi serta prediksi keberhasilan pekerjaan maupun keterlambatan proyek pada sistem *work order* dan aplikasi berbasis *web*.

Meskipun berbagai penelitian terdahulu telah menunjukkan kemampuan *Random Forest* dalam klasifikasi teks dan tiket layanan, pengelolaan laporan teknisi pada PT Maxkomp memiliki karakteristik tersendiri. Laporan perusahaan menggabungkan empat jenis fitur sekaligus dalam satu *pipeline*, yaitu teks bebas berbahasa Indonesia yang menyatukan keluhan dan solusi, kategori pelanggan, tahun sumber, serta komponen tanggal yang memerlukan penguraian kalender. Di samping itu, *dataset* penelitian mencakup laporan nyata dan data sintetis hasil augmentasi dengan label acuan yang dibentuk melalui kebijakan pelabelan berbantuan aturan otomatis, belum berupa pengesahan keputusan manusia secara independen. Kondisi ini menuntut evaluasi model tidak hanya membaca *accuracy* gabungan, melainkan turut mengukur *balanced accuracy*, *recall* kelas minoritas, serta memisahkan capaian pada laporan nyata dan data sintetis agar penafsiran hasil tetap objektif.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan merancang dan mengimplementasikan metode *Random Forest* ke dalam aplikasi berbasis *web* lokal untuk mengolah serta mengklasifikasikan laporan teknisi dan administrasi PT Maxkomp ke dalam dua kelas, yaitu SELESAI dan TINDAK_LANJUT. Pemanfaatan antarmuka *web* dipilih karena mampu menyajikan alur pengolahan data, pelatihan model, evaluasi, dan rekomendasi prediksi secara interaktif melalui peramban (Saia et al., 2022). Seluruh *dataset*, konfigurasi, model terlatih, hasil evaluasi, dan riwayat prediksi dikelola menggunakan penyimpanan berbasis berkas agar asal-usul data dan setiap tahapan pemrosesan dapat ditelusuri serta direproduksi (Wagner et al., 2022). Keluaran aplikasi diposisikan sebagai rekomendasi pendukung pemeriksaan laporan, sementara keputusan akhir tetap berada di tangan pihak administrasi atau manajer berdasarkan kondisi lapangan yang sebenarnya.

2. METODOLOGI PENELITIAN

2.1 Metode *Random Forest*

Random Forest merupakan metode *ensemble learning* yang menggabungkan sejumlah *decision tree* untuk menghasilkan prediksi klasifikasi yang lebih stabil (Utomo & Zakaria, 2026). Setiap pohon keputusan dibentuk menggunakan sampel *bootstrap* dengan pengembalian (*bagging*) serta pemilihan sebagian fitur secara acak pada setiap simpul pemisahan, kemudian keputusan akhir ditentukan melalui mekanisme *majority voting* dari seluruh pohon (Mahmuda, 2024), (Kencana et al., 2022). Pemilihan pemisahan terbaik pada tiap simpul didasarkan pada penurunan ketidakmurnian kelas menggunakan indeks *Gini impurity* seperti pada Persamaan (1). Penggunaan banyak pohon yang beragam membantu menekan risiko *overfitting* sekaligus menghasilkan informasi *feature importance* untuk melihat kontribusi variabel dominan, meskipun jumlah pohon

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

perlu ditentukan secara proporsional agar beban komputasi tetap terkendali dan hasilnya tetap bergantung pada kecukupan data riil (Siregar et al., 2023), (Yuda et al., 2026).

Sebelum dilatih oleh *Random Forest*, data laporan yang beragam diubah ke bentuk numerik di dalam satu *pipeline* terpadu. Fitur teks keluhan dan solusi dibobotkan menggunakan *Term Frequency-Inverse Document Frequency* (*TF-IDF*) berdasarkan frekuensi kemunculan kata dan tingkat kelangkaannya pada seluruh dokumen sebagaimana Persamaan (2) (Anjani et al., 2023). Atribut kategori pelanggan diubah menjadi vektor biner melalui *One-Hot Encoding* (Utomo & Zakaria, 2026), nilai kosong pada fitur numerik dan hasil uraian tanggal (tahun, bulan, serta hari dalam minggu) ditangani dengan imputasi median (Siregar et al., 2023). Kinerja model pada data uji dievaluasi melalui *confusion matrix* untuk menghitung *accuracy*, *precision*, *recall*, *F1-score*, dan *balanced accuracy* sesuai Persamaan (3) hingga (5), sehingga ketepatan prediksi terhadap kelas mayoritas maupun minoritas dapat dibaca secara seimbang (Hicks et al., 2022), (Opitz, 2024).

$$G(D) = 1 - \sum_{k=1}^{K} p_k^2$$

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \log\left(\frac{N}{\text{df}(t)}\right)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.2 Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui wawancara, observasi berkas laporan, dan studi pustaka. Wawancara dilaksanakan pada 8 November 2026 bersama dua teknisi dan seorang manajer PT Maxkomp untuk memperoleh keterangan mengenai alur pencatatan keluhan pelanggan, tindakan perbaikan, pemeriksaan administrasi, serta kriteria awal penentuan laporan yang telah selesai dan yang masih memerlukan tindak lanjut. Studi pustaka dilakukan terhadap jurnal ilmiah dan dokumentasi teknis terkait *data mining*, pemrosesan teks, klasifikasi *Random Forest*, penyimpanan berbasis berkas dengan format tabular dan metadata terstruktur (Alpatov & Bogatireva, 2024), tantangan pemuatan berkas tabular (Vitagliano et al., 2023), serta standar pengujian perangkat lunak (ISTQB, 2024).

Data sekunder penelitian berupa berkas *workbook* konsolidasi laporan PT Maxkomp yang memuat 3.000 *record* dan 21 kolom. Sebanyak 1.541 *record* merupakan laporan nyata yang dihimpun dari 31 berkas sumber dan 11 kelompok pelanggan pada periode 2023 sampai 2025, sedangkan 1.459 *record* lainnya merupakan data sintetis hasil augmentasi berbasis contoh berbobot dan variasi tanggal untuk keperluan eksperimen. Empat kolom ditetapkan sebagai fitur masukan, yakni tahun sumber, identitas pelanggan, catatan keluhan dan solusi, serta tanggal laporan, dengan satu kolom *target* keputusan laporan bernilai SELESAI atau TINDAK_LANJUT. Kolom status lama, identitas sumber, penanda asal data, serta metadata pelabelan tidak dipakai sebagai fitur prediktor, melainkan disimpan terpisah guna menjaga penelusuran asal-usul data dan reproduksibilitas eksperimen (Wagner et al., 2022).

2.3 Metode Pengembangan Sistem

Pengembangan aplikasi berbasis *web* dalam penelitian ini menerapkan *System Development Life Cycle* (*SDLC*) model *Waterfall* sebagai kerangka kerja utama. Menurut (Ningsih & Nurfauziah, 2023), model *Waterfall* menyusun tahapan pembangunan perangkat lunak secara sistematis dan berurutan sehingga keluaran satu tahap menjadi dasar pelaksanaan tahap berikutnya, sangat sesuai bagi sistem yang ruang lingkup dan kebutuhannya dapat dirumuskan dengan jelas sejak awal. Rangkaian tahapan tersebut meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan sebagaimana diterapkan pada pengembangan sistem manajemen berbasis *web*

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

(Ludiansyah et al., 2026). Analisis kebutuhan mengidentifikasi kondisi berkas dan fungsi pengolahan data, perancangan menerjemahkan kebutuhan menjadi struktur data dan alur antarmuka, implementasi membangun layanan *frontend* dan *backend*, pengujian memeriksa kesesuaian fungsi serta alur kendali program, dan pemeliharaan menyiapkan penelusuran berkas pascaimplementasi.

Untuk memperjelas kebutuhan dan bentuk tampilan antarmuka sebelum implementasi final diselesaikan, penelitian ini memanfaatkan pendekatan *prototyping* secara iteratif sebagai pendukung. (Ningsih & Nurfauziah, 2023) menjelaskan bahwa *Prototype* memungkinkan penyusunan rancangan awal yang dievaluasi dan diperbaiki lebih dini sehingga ketidaksesuaian alur dapat diketahui sejak tahap perancangan. Pemodelan perilaku dan struktur perangkat lunak didokumentasikan menggunakan *Unified Modeling Language* (*UML*) yang terdiri atas *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* (Wayahdi & Ruziq, 2023). Sementara itu, hubungan antarartefak data pada penyimpanan berkas dimodelkan secara logis melalui *Entity Relationship Diagram* (*ERD*) dan *Logical Record Structure* (*LRS*) (Anna & Lisnawanty, 2024), serta diverifikasi menggunakan metode pengujian *black box* dan *white box* (ISTQB, 2024).

3. ANALISA DAN PEMBAHASAN

3.1 Analisis Sistem

Hasil wawancara teknisi dan manajer serta pemeriksaan terhadap 1.541 *record* laporan nyata dari 31 berkas sumber dan 11 kelompok pelanggan menunjukkan bahwa sistem pelaporan berjalan masih bertumpu pada berkas terpisah tanpa penyeragaman struktur. Catatan keluhan, pemeriksaan, tindakan, dan hasil pekerjaan tergabung dalam satu kolom teks bebas, penulisan nama pelanggan dan uraian sering mengandung spasi berlebih, serta ditemukan sejumlah baris kosong dan duplikasi kombinasi fitur maupun *target*. Pada aspek waktu, tercatat 77 nilai tanggal kosong atau tidak dapat digunakan serta 472 *record* laporan nyata yang menunjukkan perbedaan antara tahun berkas sumber dan tahun kalender pada tanggal laporan. Kondisi permasalahan data beserta implikasi rancangannya dirangkum pada Tabel 1.

Untuk menjawab kendala tersebut, sistem usulan dirancang sebagai aplikasi *web* lokal untuk satu operator berarsitektur *modular monolith* yang memisahkan antarmuka *frontend* (React, TypeScript, dan Vite) dengan layanan *backend* (FastAPI dan Python dengan pustaka Pandas, OpenPyXL, Scikit-learn, serta Joblib). Seluruh *dataset* mentah, berkas hasil pembersihan, data siap latih, konfigurasi, *manifest* pembagian data, *pipeline* model terlatih, laporan evaluasi, hingga riwayat prediksi disimpan pada *filesystem* lokal berdasarkan identitas ruang kerja terpisah agar berkas sumber tidak pernah tertimpa. Alur fungsional sistem mencakup tujuh tahap berkesinambungan mulai dari unggah dan konfigurasi *dataset*, pemahaman data, pembersihan dan audit, prapemrosesan, pembagian data dan pelatihan model, evaluasi dan aktivasi manual model, sampai pengajuan rekomendasi prediksi dan pencatatan keputusan aktual oleh pemeriksa.

Tabel 1. Analisis Kondisi Data dan Kebutuhan Rancangan Sistem

Kondisi Data Berjalan	Dampak Pengolahan	Solusi Rancangan Sistem
Berkas tersebar lintas pelanggan dan tahun	Sulit melacak asal *record*	Simpan berkas mentah dan identitas sumber
Tipe dan penulisan nilai tidak seragam	Fitur tidak siap langsung dimodelkan	Validasi tipe dan normalisasi terukur
Kombinasi fitur dan *target* berulang	*Record* dihitung lebih dari satu kali	Deduplikasi persis disertai laporan audit
Sebagian nilai *target* tidak tersedia	Tidak dapat dilatih atau dievaluasi	Pengecualian *target* kosong dengan catatan
Label dibentuk aturan otomatis	Belum berupa pengesahan manusia	Pelaporan asal data dan verifikasi manual

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

3.2 Analisa Penerapan Metode *Random Forest*

Penerapan *Random Forest* diawali dengan proses *cleaning* dan prapemrosesan dua lapis untuk mencegah kebocoran informasi dari data uji. Dari 3.000 *record* awal pada berkas kerja, tahap *cleaning* menormalisasi spasi teks, menyeragamkan format tanggal ke standar ISO, memvalidasi tipe numerik, serta mengeluarkan 71 duplikat persis (termasuk 69 *record* bertarget kosong) sehingga menyisakan 2.929 *record*. Pada tahap prapemrosesan, 12 *record* bertarget kosong yang tersisa dikecualikan sehingga terbentuk 2.917 kandidat pemodelan yang terdiri atas 2.022 kelas SELESAI dan 895 kelas TINDAK_LANJUT dengan rasio ketidakseimbangan sekitar 2,26:1. Lapisan prapemrosesan pertama menguraikan fitur tanggal menjadi tiga komponen kalender (tahun, bulan, dan hari dalam minggu), sedangkan lapisan kedua dijalankan melalui *pipeline* hanya pada data latih untuk mempelajari imputasi median (tahun sumber 2024, tahun tanggal 2024, bulan 7, dan hari ke-3), kategori *One-Hot Encoding*, serta kosakata *TF-IDF unigram* huruf kecil hingga batas 5.000 kata yang menghasilkan 1.453 kosakata dan total dimensi matriks 1.468 fitur tanpa nilai kosong.

Pembagian data menerapkan skema *fixed holdout* berbasis pengelompokan (*StratifiedGroupKFold* untuk rasio 75:25 dan 80:20, serta *GroupShuffleSplit* untuk 70:30) dengan *random state* 42 agar tidak terjadi tumpang tindih identitas *record* maupun kelompok fitur serupa antara data latih dan data uji. Eksperimen utama ditetapkan sejak awal pada rasio 75:25 yang menghasilkan 2.187 data latih dan 730 data uji, seperti dirangkum pada Tabel 2, sementara rasio 80:20 dan 70:30 dijadikan eksperimen pendamping. Model *Random Forest* dilatih menggunakan parameter *baseline* tetap berupa 100 pohon keputusan, kriteria *Gini impurity*, tanpa batas kedalaman maksimum, serta pembobotan kelas seimbang (*balanced*) yang dihitung otomatis dari distribusi data latih. Pelatihan tidak menggunakan pencarian *hyperparameter* maupun penambahan *oversampling* baru, kemudian *pipeline* terlatih disimpan bersama *transformer* agar proses prediksi laporan baru menerapkan transformasi yang persis sama.

Tabel 2. Rekonsiliasi Jumlah *Record* pada Tahapan Pengolahan Utama

Tahap Pengolahan	Jumlah *Record*	Keterangan
Dataset mentah awal	3.000	1.541 laporan nyata dan 1.459 data sintetis
Setelah *cleaning* dan audit	2.929	71 duplikat persis dikeluarkan
Kandidat pemodelan final	2.917	12 *target* kosong tersisa dikecualikan
Data latih utama (75%)	2.187	Dimensi matriks 2.187 x 1.468 fitur
Data uji utama (25%)	730	364 laporan nyata dan 366 data sintetis

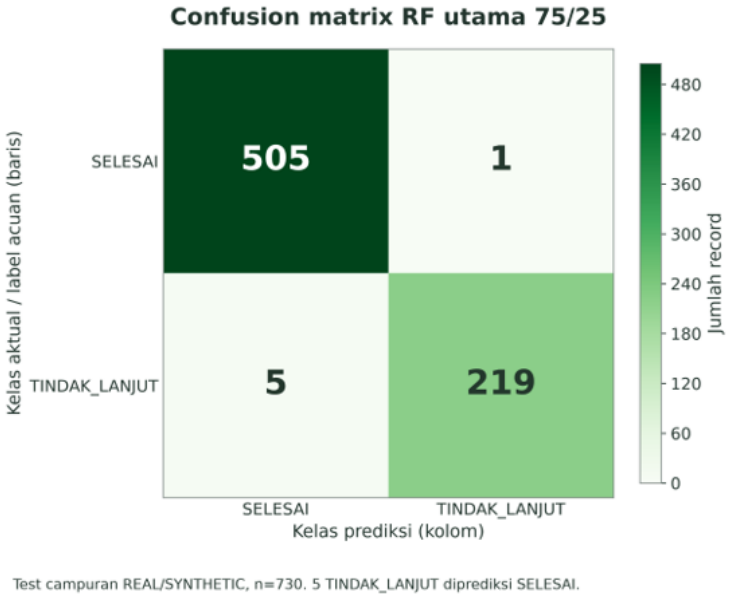
3.3 Pembahasan Hasil

Pada pengujian utama rasio 75:25 terhadap 730 data uji, model *Random Forest* pada aplikasi menghasilkan 724 prediksi sesuai label acuan, yaitu 505 kelas SELESAI dan 219 kelas TINDAK_LANJUT, dengan enam kesalahan yang terdiri atas satu SELESAI diprediksi TINDAK_LANJUT dan lima TINDAK_LANJUT diprediksi SELESAI seperti terlihat pada Gambar 1. Capaian tersebut menghasilkan *accuracy* sebesar 99,18%, *balanced accuracy* 98,79%, *macro F1-score* 99,03%, *weighted F1-score* 99,18%, *precision* TINDAK_LANJUT 99,55%, dan *recall* TINDAK_LANJUT 97,77%. Pengujian pendamping pada rasio 80:20 dan 70:30 memperlihatkan tren agregat yang konsisten dengan rentang *accuracy* 98,80% hingga 99,18% serta *recall* TINDAK_LANJUT sebesar 96,09% (7 dari 179 terlewat) pada 80:20 dan 96,64% (9 dari 268 terlewat) pada 70:30. Perbandingan terhadap RapidMiner (Altair AI Studio Free 2026.1.1) pada 1.468 matriks fitur dan partisi 75:25 yang sama menghasilkan *accuracy* identik sebesar 99,18% dengan 728 prediksi sama dan dua *record* berbeda; aplikasi unggul tipis pada *recall* TINDAK_LANJUT (97,77% berbanding 97,32%), sedangkan RapidMiner memperoleh *precision* TINDAK_LANJUT lebih tinggi (100% berbanding 99,55%) karena tidak menghasilkan kesalahan positif semu pada kelas SELESAI, sebagaimana dirangkum pada Tabel 3.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

Tabel 3. Perbandingan Hasil Evaluasi Random Forest pada Data Uji Utama (75:25)

Metrik Evaluasi	Aplikasi	RapidMiner	Keterangan Pengujian
Accuracy keseluruhan	99,18%	99,18%	724 benar dari 730 data uji
Precision TINDAK_LANJUT	99,55%	100,00%	Kesalahan FP: 1 (Aplikasi), 0 (RapidMiner)
Recall TINDAK_LANJUT (Gabungan)	97,77%	97,32%	219/224 (Aplikasi); 218/224 (RapidMiner)
Recall TINDAK_LANJUT (Sintetis)	100,00%	100,00%	218 dari 218 kasus terdeteksi
Recall TINDAK_LANJUT (Laporan Nyata)	16,67%	0,00%	1/6 (Aplikasi); 0/6 (RapidMiner)



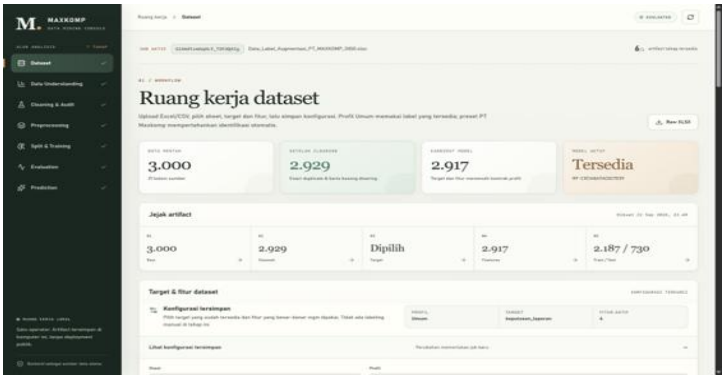
Gambar 1. *Confusion Matrix* Pengujian Utama Rasio 75:25

4. IMPLEMENTASI

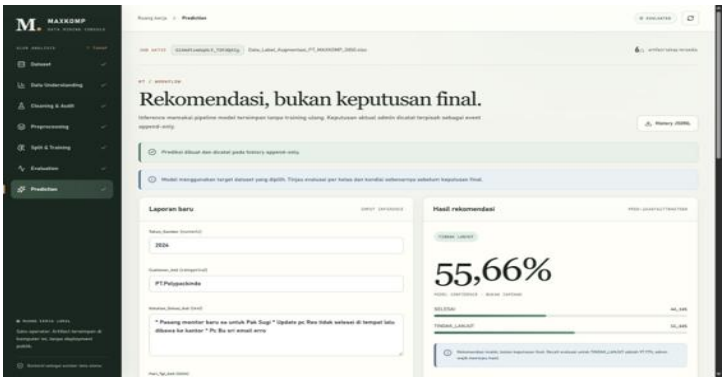
4.1 Implementasi Sistem

Aplikasi dijalankan pada lingkungan lokal dengan layanan *frontend* di alamat 127.0.0.1:5174 dan *backend* di 127.0.0.1:8001 menggunakan penyimpanan berbasis berkas tanpa basis data relasional. Antarmuka sistem disusun mengikuti tujuh tahapan utama yang saling terhubung melalui navigasi di sisi kiri layar. Halaman *Dataset* pada Gambar 2 menjadi titik awal bagi operator untuk mengunggah berkas *workbook* atau CSV, membuka ruang kerja lokal yang sudah tersimpan, memilih *sheet* dan profil pengolahan, serta menetapkan kolom *target* dan empat fitur masukan beserta jenis datanya. Selanjutnya, halaman *Data Understanding* menampilkan ringkasan jumlah baris, kolom, nilai kosong, dan distribusi *target*; halaman *Cleaning* dan Audit menjalankan normalisasi serta penghapusan duplikat ke berkas terpisah; sedangkan halaman *Preprocessing* memisahkan 12 *record* bertarget kosong dan menetapkan kontrak transformasi fitur sebelum memasuki tahap pemodelan.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003



Gambar 2. Implementasi Halaman *Dataset*



Gambar 3. Implementasi Halaman *Prediction*

4.2 Implementasi dan Pengujian Sistem

Pengujian perangkat lunak dilaksanakan secara terpisah dari penyimpanan eksperimen utama untuk menjaga agar data simulasi tidak bercampur dengan artefak penelitian. Pengujian *black box* memeriksa kesesuaian masukan valid dan tidak valid, validasi prasyarat urutan proses, serta keluaran respons layanan. Sebanyak 72 skenario pengujian melalui *Application Programming Interface* (API) yang mencakup kesehatan layanan, unggah dan konfigurasi *dataset*, pemahaman data, *cleaning*, prapemrosesan, pembagian data dan pelatihan, evaluasi, aktivasi model, prediksi, riwayat, serta unduhan artefak seluruhnya dinyatakan lulus (72 lulus, 0 gagal). Skenario negatif memastikan aplikasi menolak format berkas tidak didukung, *workbook* rusak, penggunaan *target* sebagai fitur, serta eksekusi tahap sebelum prasyarat terpenuhi, sedangkan pengujian empat nilai fitur kosong berhasil diterima sesuai kebijakan imputasi. Tiga observasi pelengkap pada antarmuka *web* untuk koneksi ruang kerja, penolakan berkas teks, dan penolakan *workbook* rusak juga seluruhnya lulus.

Tabel 4. Rekapitulasi Hasil Pengujian *Black Box* dan *White Box*

Jenis Pengujian	Objek Pengujian	Jumlah Kasus / Kompleksitas	Hasil Pengujian
Black Box (API)	Seluruh layanan dan validasi alur	72 skenario uji	72 Lulus (100%)
Black Box (Antarmuka)	Koneksi ruang kerja dan tolak berkas	3 observasi langsung	3 Lulus (100%)
White Box	Verifikasi integritas eksperimen	V(G) = 6 (14 simpul, 18 sisi)	Lulus (cakupan 100%)
White Box	Aktivasi model terlatih	V(G) = 4 (12 simpul, 14 sisi)	Lulus (cakupan 100%)
White Box	Pencatatan keputusan aktual	V(G) = 5 (14 simpul, 17 sisi)	Lulus (cakupan 100%)

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 5, No. 11, November 2026
ISSN 2828-2442 (media online)
Hal 995-1003

5. KESIMPULAN

Aplikasi klasifikasi laporan teknisi dan administrasi PT Maxkomp berbasis *web* telah berhasil dibangun menggunakan penyimpanan berbasis berkas yang menghubungkan proses unggah *dataset*, pemahaman data, *cleaning*, prapemrosesan, pembagian data dan pelatihan, evaluasi, hingga prediksi dan riwayat keputusan, serta lulus 72 skenario pengujian *black box* API, 3 observasi antarmuka, dan 26 kasus *white box* dengan cakupan baris dan cabang sebesar 100%. Metode *Random Forest* dengan 100 pohon keputusan berhasil diterapkan melalui *pipeline* yang memadukan fitur numerik, kategori, teks *TF-IDF*, dan komponen tanggal terhadap 2.917 kandidat hasil pembersihan dari 3.000 *record* awal (2.187 data latih, 730 data uji, dan 1.468 fitur hasil transformasi) tanpa kebocoran data uji. Pada data uji campuran rasio 75:25, model mencapai *accuracy* 99,18%, *balanced accuracy* 98,79%, *macro F1-score* 99,03%, dan *recall* TINDAK_LANJUT 97,77%, sejalan dengan hasil RapidMiner yang memperoleh *accuracy* 99,18% dan *recall* 97,32%. Namun, karena *recall* TINDAK_LANJUT pada enam laporan nyata baru mencapai 16,67% pada aplikasi dan 0% pada RapidMiner dibanding 100% pada data sintetis, hasil klasifikasi perlu diposisikan sebagai rekomendasi pendukung pemeriksaan laporan dengan keputusan akhir tetap melalui verifikasi manusia; penelitian lanjutan disarankan menambah jumlah laporan nyata berlabel terverifikasi manusia, menerapkan validasi temporal dengan beberapa *seed*, serta menguji kalibrasi probabilitas untuk penggunaan operasional.

REFERENCES

Aini, N., Arif, M., Agustin, I. T., & Toyibah, Z. B. (2024). Implementasi algoritma Random Forest untuk klasifikasi bidang MSIB di Prodi Pendidikan Informatika. Jurnal Informatika, 11(1), 11–15. https://doi.org/10.31294/inf.v11i1.20637

Alpatov, A. N., & Bogatireva, A. A. (2024). Data storage format for analytical systems based on metadata and dependency graphs between CSV and JSON. Software Systems and Computational Methods, (2), 1–14. https://doi.org/10.7256/2454-0714.2024.2.70229

Anjani, A. F., Anggraeni, D., & Tirta, I. M. (2023). Implementasi Random Forest menggunakan SMOTE untuk analisis sentimen ulasan aplikasi Sister for Students UNEJ. Jurnal Nasional Teknologi Dan Sistem Informasi, 9(2), 163–172. https://doi.org/10.25077/TEKNOSI.v9i2.2023.163-172

Anna, & Lisnawanty. (2024). Sistem informasi pengelolan laporan keuangan berbasis web pada CV. Damar Abadi. JUSTIAN: Jurnal Sistem Informasi Akuntansi, 5(1), 67–78. https://jurnal.bsi.ac.id/index.php/justian/article/view/5057

Crosley, N., & Wasito, I. (2023). Improving IT support efficiency using AI-driven ticket random forest classification technique. Sinkron: Jurnal Dan Penelitian Teknik Informatika, 7(4), 2283–2293. https://doi.org/10.33395/sinkron.v8i4.12925

Hicks, S. A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M. A., Halvorsen, P., & Parasa, S. (2022). On evaluation metrics for medical applications of artificial intelligence. Scientific Reports, 12. https://doi.org/10.1038/s41598-022-09954-8

ISTQB. (2024). Certified Tester Foundation Level Syllabus (Version 4.0.1).

Kencana, A. K., Ananda, F. D., Hartanto, A. D., & Hartatik. (2022). Implementasi metode Random Forest klasifikasi untuk phishing link detection. Information Technology Journal, 4(2), 55–59.

Ludiansyah, R. S., Lisnawanty, & Handayani, K. (2026). Pengembangan sistem manajemen apotek berbasis web dengan fitur prediksi pengadaan obat menggunakan Random Forest. JUKTISI, 5(1), 815–824. https://doi.org/10.62712/juktisi.v5i1.1163

Mahmuda, S. (2024). Implementasi metode Random Forest pada kategori konten kanal Youtube. Jurnal Jendela Matematika, 2(1), 21–31.

Ningsih, W., & Nurfauziah, H. (2023). Perbandingan model Waterfall dan metode Prototype untuk pengembangan aplikasi pada sistem informasi. Jurnal Ilmiah METADATA, 5(1), 83–95. https://doi.org/10.47652/metadata.v5i1.311

Oliveira, D. F., Nogueira, A. S., & Brito, M. A. (2022). Performance comparison of machine learning algorithms in classifying information technologies incident tickets. AI, 3(3), 601–622. https://doi.org/10.3390/ai3030035

Opitz, J. (2024). A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. Transactions of the Association for Computational Linguistics, 12, 820–836. https://doi.org/10.1162/tacl_a_00675

Saia, S. M., Nelson, N. G., Young, S. N., Parham, S., & Vandegrift, M. (2022). Ten simple rules for researchers who want to develop web apps. PLOS Computational Biology, 18(1). https://doi.org/10.1371/journal.pcbi.1009663

Siregar, A. P., Purba, D. P., Pasaribu, J. P., & Bakara, K. R. (2023). Implementasi algoritma Random Forest dalam klasifikasi diagnosis penyakit stroke. Jurnal Penelitian Rumpun Ilmu Teknik (JUPRIT), 2(4), 155–164. https://doi.org/10.55606/juprit.v2i4.3039