Deteksi *Fraud* pada Transaksi dalam Perbankan menggunakan *Random Forest*

Tugas Akhir

diajukan untuk memenuhi salah satu syarat

memperoleh gelar sarjana

dari Program Studi Informatika

Fakultas Informatika

Universitas Telkom

1301204210

Muhammad Naufal Zaidan

Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung

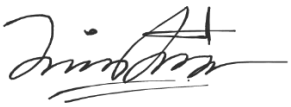
2024

LEMBAR PERNYATAAN

Dengan ini saya, Muhammad Naufal Zaidan, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul Deteksi *Fraud* pada Transaksi dalam Perbankan menggunakan *Random Forest* beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang belaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 8 Agustus 2024

Yang Menyatakan



Muhammad Naufal Zaidan

LEMBAR PENGESAHAN

Deteksi *Fraud* pada Transaksi dalam Perbankan menggunakan *Random Forest*

Fraud Detection in Banking Transactions using Random Forest

Muhammad Naufal Zaidan

NIM :1301204210

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh

gelar pada Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung, 8 Agustus 2024

Menyetujui

Pembimbing 1

Dr. DENI SAEPUDIN, S.Si., M. Si

99750013

Ketua Program Studi

Sarjana Teknik Informatika/Komputasi

Dr. Erwin Budi Setiawan, S.Si., M.T.

NIP: 00760045

ABSTRAK

Pada era digitalisasi, transaksi yang mencurigakan di rekening bank berdampak serius terhadap kepentingan pedagang dan nasabah bank. *Fraud detection* adalah salah satu upaya pencegahan terhadap kecurangan tersebut diatasi dengan deteksi pola-pola yang mencurigakan dari suatu transaksi. Pada penelitian ini dilakukan deteksi kecurangan(*fraud detection*) menggunakan algoritma *Random Forest* terhadap dataset transaksi bank yang telah diolah oleh aplikasi *Solve Ease Fraud Investigation System* (SEFIS). Algoritma *Random Forest* digunakan karena beberapa kasus yang dilakukan oleh peneliti sebelumnya menunjukkan algoritma *Random Forest* dapat mengelola permasalahan *fraud detection* dengan baik. Kontribusi utama pada penelitian ini adalah penerapan *machine learning* dengan model algoritma *Random Forest* pada data transaksi bank khususnya di indonesia untuk mendeteksi kecurangan yang terjadi dengan secara menyeluruh. Dataset yang digunakan adalah dataset transaksi bank yang telah diolah oleh aplikasi *Solve Ease Fraud Investigation System* (SEFIS) dengan karakteristik data tidak seimbang(*imbalance*). Untuk setiap *imbalance* data, kami melakukan penanganan menyeimbangkan data (*Balancing*) menggunakan *Random UnderSampling* (RUS). kami melakukan ujicoba dengan melakukan komparasi terhadap data yang dilakukan *balancing* menggunakan *RandomUndersampling* dengan hasil *Accuracy* = 0.9927, *Precision* = 0.9799, *Recall* = 0.9949, *Specificity* = 0.9999, *F1-Score* = 0.9873 dan data yang tidak dilakukan *balancing* dengan hasil *Accuracy* = 0.9986, *Precision* = 0.9947, *Recall* = 0.8732, *Specificity* = 0.9999, *F1-Score* = 0.9300.

Kata Kunci: *Fraud Detection*, Bank, *Random Forest*, *Random Under Sampling*, *Imbalance Data*

ABSTRACT

In the digital era, suspicious transactions in bank accounts have a serious impact on the interests of merchants and bank customers. Fraud detection is one of the efforts to prevent fraud that is overcome by detecting suspicious patterns from a transaction. In this study, a fraud detection was developed using the Random Forest algorithm on bank transaction datasets that have been processed by the Solve Ease Fraud Investigation System (SEFIS) application. The Random Forest algorithm is used because several cases conducted by previous researchers have shown that the Random Forest algorithm can manage fraud detection problems well. The main contribution to this study is the application of machine learning with the Random Forest algorithm model to bank transaction data, especially in Indonesia, to detect fraud that occurs comprehensively. The dataset used is a bank transaction dataset that has been processed by the Solve Ease Fraud Investigation System (SEFIS) application with unbalanced data characteristics. For each imbalanced data, we handle the data balancing using Random Undersampling (RUS). We conducted a trial by comparing the data that was balanced using RandomUndersampling with the results Accuracy = 0.9927, Precision = 0.9799, Recall = 0.9949, Specificity = 0.9999, F1-Score = 0.9873 and data that was not balanced with the results Accuracy = 0.9986, Precision = 0.9947, Recall = 0.8732, Specificity = 0.9999, F1-Score = 0.9300.

Keywords: *Fraud Detection, Bank, Random Forest, Random Under Sampling, Imbalanced Data*

KATA PENGANTAR

Dengan menyebut nama Allah yang maha pengasih lagi maha penyayang, puji syukur atas segala rahmat serta karunia-Nya sehingga kami dapat menyelesaikan tugas akhir ini.

Tugas akhir dengan judul "Deteksi *Fraud* pada Transaksi dalam Perbankan menggunakan *Random Forest*" merupakan hasil dari perjalanan masa studi yang telah kami tempuh serta untuk memenuhi salah satu syarat dalam menyelesaikan studi di Program Studi Sarjana Informatika, Fakultas Informatika, Telkom University. Kami ingin menyampaikan terima kasih yang sebesar-besarnya kepada dosen pembimbing kami, bapak Dr. Deni Saepudin, S.Si., M.Si., yang telah membimbing, memberi arahan, serta dukungan kepada kami selama pengerjaan tugas akhir ini. Kami juga ingin menyampaikan terima kasih kepada Chicco Eka Putra dan Nurhamdalah Kahfi dari pihak pengembang aplikasi *Solve Ease Fraud Investigation System* (SEFIS) yang telah memberikan dukungan serta izin untuk mengolah dataset yang kami gunakan untuk penelitian tugas akhir ini.

Tak lupa kami ucapkan terima kasih kepada keluarga kami yang selalu memberikan dukungan, kasih sayang, dan doa yang tak terhingga setiap saat. Keberhasilan ini merupakan buah dari pengorbanan serta kasih sayang mereka selama ini.

Semoga dengan tugas akhir yang kami lakukan dapat bermanfaat dan serta memberikan kontribusi kecil terhadap ranah pengembangan ilmu pengetahuan, khususnya bidang perbankan. Kami sadar bahwa dalam penyusunan tugas akhir ini masih terdapat kekurangan dan keterbatasan. Segala hal yang baik dalam tugas akhir ini tidak lepas dari bantuan serta dukungan berbagai pihak, serta mohon maaf atas kekurangan dan keterbatasan dalam penyusunan tugas akhir ini.

Terima kasih.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

1. PENDAHULUAN

1.1. Latar Belakang

Pada era digitalisasi, pengembangan Transaksi yang mencurigakan di rekening bank tidak hanya membawa risiko besar bagi industri perbankan, tetapi juga berdampak serius terhadap kepentingan pedagang dan nasabah bank, yang menyebabkan kerugian langsung maupun tidak langsung yang signifikan dengan tingkat yang bervariasi[1]. Upaya pencegahan terhadap kecurangan transaksi bank terus berkembang. Salah satu langkah pencegahan terhadap kecurangan tersebut dengan upaya pendeteksian pola-pola yang mencurigakan (*fraud*) dari suatu transaksi yang disebut juga *fraud detection*. *Fraud detection* sering kali melibatkan pendekatan *machine learning* untuk membangun model yang dapat mendeteksi antara transaksi normal(*non-fraud*) dan transaksi mencurigakan (*fraud*).

Penelitian terkait *Fraud Detection* sudah banyak dilakukan, salah satunya oleh Muhammad Sopiyan, dkk. yang menunjukkan *Random Forest Classifier* (RFC) memiliki performansi terbaik dibanding dengan algoritma lain dengan nilai akurasi pada data *training* adalah 100% dan data *testing* 99.99% [2]. Hasil serupa juga didapat oleh K. Deepika, dkk. dengan hasil pada nilai *accuracy* 100%, *recall* 78%, *precision* 91%, dan *F1-Score* 84% [3].

Berdasarkan beberapa penelitian yang telah dilakukan, akan dilakukan deteksi kecurangan menggunakan algoritma *Random Forest* terhadap dataset transaksi bank yang telah diolah oleh aplikasi *Solve Ease Fraud Investigation System* (SEFIS) dalam jangka waktu 1 November 2023 hingga 19 November 2023. Penelitian ini menggunakan algoritma *Random Forest* karena dari beberapa kasus yang telah dilakukan oleh peneliti sebelumnya dengan melakukan komparasi terhadap beberapa algoritma, hasil menunjukkan algoritma *Random Forest* dapat mengelola permasalahan *fraud detection* dengan baik.

Pada dataset transaksi yang akan kami gunakan memiliki karakteristik data yang tidak seimbang (*imbalance*). Ketidakseimbangan(*imbalance*) data terlihat dengan data transaksi normal(*non-fraud*) memiliki nilai mayoritas dan data transaksi

mencurigakan (*fraud*) memiliki nilai minoritas. Perlu dilakukan penyeimbangan data(*Balancing Data*) dengan menerapkan *undersampling* terhadap data mayoritas. Untuk mengetahui performa yang baik, kami melakukan ujicoba dengan melakukan komparasi terhadap data yang dilakukan *undersampling* dan data yang tidak dilakukan *undersampling* dengan menganalisa dengan beberapa metriks performansi yang meliputi: a. *Accuracy*, b. *Precision*, c. *Recall*, d. *Specificity*, dan e. *F1-Score*.

kontribusi utama pada penelitian ini adalah penerapan *machine learning* dengan model algoritma *Random Forest* pada data transaksi bank khususnya di indonesia untuk mendeteksi kecurangan yang terjadi dengan secara menyeluruh. Dengan melakukan perbandingan data asli tanpa *Balancing* dan data yang dilakukan *Balancing* menggunakan *RandomUndersampling*(RUS), penelitian ini memberikan wawasan terhadap pendeteksian pada kecurangan transaksi bank menggunakan machine learning, khususnya pada algoritma *Random Forest*, serta dapat dilakukan implementasi pada aplikasi *Solve Ease Fraud Investigation System* (SEFIS) untuk pengembangan sistem deteksi kecurangan menggunakan *machine learning* dengan model algoritma *Random Forest* di masa depan.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, maka dirumuskan permasalahan yang diangkat sebagai berikut:

1. Bagaimana membangun *fraud detection system* menggunakan algoritma *random forest*?
2. Bagaimana kinerja algoritma *random forest* pada *fraud detection system* dalam memprediksi transaksi yang termasuk *fraud* atau *non-fraud?*

1.3. Batasan Masalah

Batasan permasalahan pada penelitian tugas akhir ini mencakup:

1. *Fraud* pada kasus ini merupakan data yang dikategorikan sebagai *suspected fraud* yang dilakukan oleh aplikasi *Solve Ease Fraud Investigation System* (SEFIS)

1.4. Tujuan

Tujuan yang di capai pada penelitian tugas akhir ini, yaitu:

1. Membangun *Fraud Detection System* menggunakan metode *Random Forest.*
2. Mengukur kinerja algoritma *Random Forest* untuk *Fraud Detection System* dengan metriks performansi *accuracy*, *precision*, *recall*, *specificity* dan *F1-Score*

1.5. Sistematika Penulisan

Dalam laporan ini, sistematika penulisan dibagi menjadi lima bab yang dijabarkan sebagai berikut:

1. BAB I Pendahuluan

   Pada BAB I dijelaskan terkait permasalahan *fraud detection*, metode *Random Forest*, dan dataset secara umum terkait kasus yang akan dibahas.

2. BAB II Kajian Pustaka

   Pada BAB II dijelaskan terkait dasar-dasar teori mengenai *Fraud Detection*, *Random Forest*, dan *Confusion Matrix*.

3. BAB III Perancangan Sistem

   Pada BAB III dijelaskan mengenai alur sistem yang dirancang terdiri dari *Preprocessing data*, *Splitting Dataset*, dan *Hyperparameter Tuning.*

4. BAB IV Pengujian dan Analisis

   Pada BAB IV dilakukan pemaparan hasil dan analisa dari pengujian yang telah dilakukan.

5. BAB V Kesimpulan dan Saran

   Pada BAB V diambil Kesimpulan dari hasil Analisa serta pengujian yang telah dilakukan serta saran untuk penelitian selanjutnya.

2. KAJIAN PUSTAKA

2.1. Studi Terkait

Studi terkait *Fraud Detection* sudah banyak dilakukan, salah satunya yang dilakukan oleh Muhammad Sopiyan, dkk. terkait *fraud detection* terhadap dataset kartu kredit dengan melakukan komparasi algoritma antara *Logistic Regression* (LGR), *Gradient Boosting Classifier* (GBC), dan *Random Forest Classifier* (RFC). Hasil dari pengujian menunjukkan *Random Forest Classifier* (RFC) memiliki performansi lebih baik dibandingkan algoritma lainnya dilihat dari nilai akurasi data *training* adalah 100% dan data *testing* 99.99% [2].

Penelitian serupa juga dilakukan K. Deepika, dkk. terkait *fraud detection* terhadap kartu kredit dengan melakukan komparasi algoritma antara *Naïve Bayes*, *Support Vector Machine*, dan *Random Forest*. Hasil dari penelitian menunjukkan bahwa *Random Forest* lebih baik dibanding dengan algoritma lainnya. Hal ini ditunjukan dengan metriks performansi memiliki nilai *accuracy* 100%, *recall* 78%, *precision* 91%, dan *F1-Score* 84% pada algoritma *Random Forest* [3].

Penelitian lain juga dilakukan oleh Lakshmi S V S S, dan Selvani Deepthi Kavila terkait *fraud detection system* pada dataset kartu kredit menggunakan *machine learning*. Pendekatan *machine learning* dilakukan dengan komparasi antara algoritma *Logistic Regression, Decision Tree,* dan *Random Forest.* metriks performansi yang digunakan adalah akurasi dengan masing-masing nilai akurasi adalah *Logistic Regression* dengan 90,0%, *Decision Tree* dengan 94,3%, dan *Random Forest* dengan 95,5%. Dari hasil evaluasi metriks performansi akurasi yang dilakukan, algoritma *Random Forest* lebih unggul dibanding *Decision Tree,* dan *Logistic Regression*[4].

Dhwanir Shah, dan Lokesh Kumar Sharma juga melakukan penelitian terhadap *fraud detection* pada dataset kartu kredit yang mengalami *imbalance* data menggunakan *Random Forest* dan *Decision Tree*. Untuk mengatasi *imbalance data*, mereka melakukan penanganan *imbalance* data menggunakan

RandomUndersampling(RUS) untuk *undersampling* dan *SMOTE* untuk *oversampling*, serta melakukan *parameter tuning* untuk meningkatkan performa. Penerapan *imbalance data* dan *parameter tuning* dilakukan pada data *training* dan *testing*[5].

Berdasarkan penelitian terkait, model algoritma *Random Forest* terbukti efektif pada kasus *fraud detection system* dengan nilai akurasi yang tinggi dibanding dengan algoritma *Logistic Regression* (LGR), dan *Gradient Boosting Classifier* (GBC) juga nilai *precision, recall,* dan *F1-score* yang baik dibandingkan algoritma *Naïve Bayes*, dan *Support Vector Machine* (SVM). Melihat dari penelitian sebelumnya [5], penanganan *imbalance data* dengan *undersampling* relevan pada penelitian ini. Oleh karena itu, algoritma *Random Forest* dengan *balancing data* menggunakan *RandomUndersampling*(RUS) layak dilakukan implementasi pada *Fraud Detection System* dengan dataset transaksi bank dari pengolahan yang dilakukan aplikasi *Solve Ease Fraud Investigation System* (SEFIS).

2.2. *Fraud Detection*

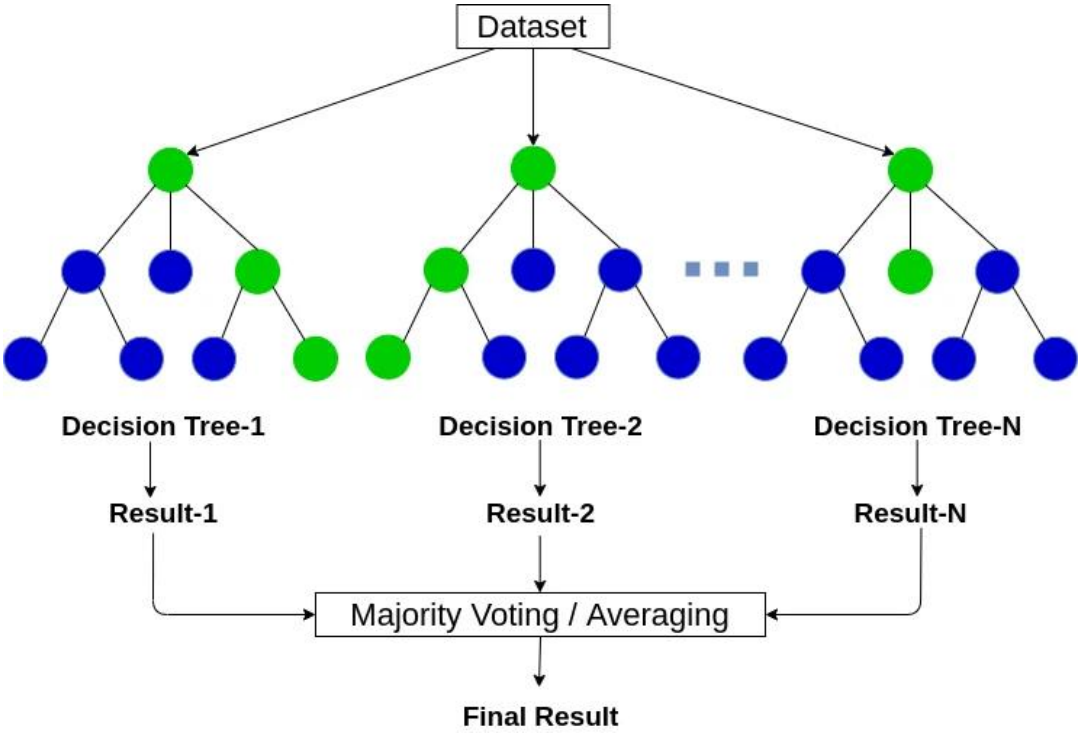
Fraud detection adalah upaya pencegahan kecurangan transaksi dengan melakukan deteksi pola-pola yang mencurigakan dari suatu transaksi. Dalam transaksi pada perbankan, tindakan pengambilan keputusan cepat dilakukan untuk investigasi lebih lanjut terhadap transaksi yang dicurigai melakukan kecurangan. *Fraud detection* dengan *machine learning* dilakukan untuk mengatasi ancaman tersembunyi [6].

Pengembangan sistem menggunakan *machine learning* yang mendeteksi antara transaksi normal (*non-fraud*) dan transaksi mencurigakan (*fraud*) disebut juga *fraud detection system* (FDS). *Fraud detection system* (FDS) dilakukan untuk mengendalikan seperduabelas dari satu persen kemungkinan pada keseluruhan transaksi yang diproses masih menghasilkan kerugian miliaran dolar[4]. Dalam penelitian ini, akan dilakukan deteksi mengurangi kasus-kasus kecurangan (*fraud*) yang terjadi pada transaksi di bank.

2.3. Random Forest

Random Forest adalah algoritma yang tangguh, terukur, dan banyak digunakan dalam berbagai bidang termasuk klasifikasi, regresi, dan deteksi anomali. Karena akurasi yang tinggi, ketahanan terhadap *overfitting*, dan kapasitas untuk menangani kumpulan data yang besar dan rumit menjadikan *Random Forest* adalah algoritma *machine learning* yang cocok untuk *fraud detection* [6].

Random Forest adalah salah satu teknik *machine learning* yang membangun banyak *decision tree* [3]. *Decision tree* dibuat dengan menentukan node akar dan berakhir dengan beberapa node daun untuk mendapatkan hasil akhir [7]. Keputusan akhir dibuat berdasarkan hasil mayoritas *decision tree* [3]. Saat membagi node untuk setiap *decision tree*, *random forest* mencari fitur terbaik dari subset data *training* secara acak yang menghasilkan keberagaman fitur berdampak pada model memprediksi lebih baik [8].



Gambar 2-1 : Model Algoritma *Random Forest*

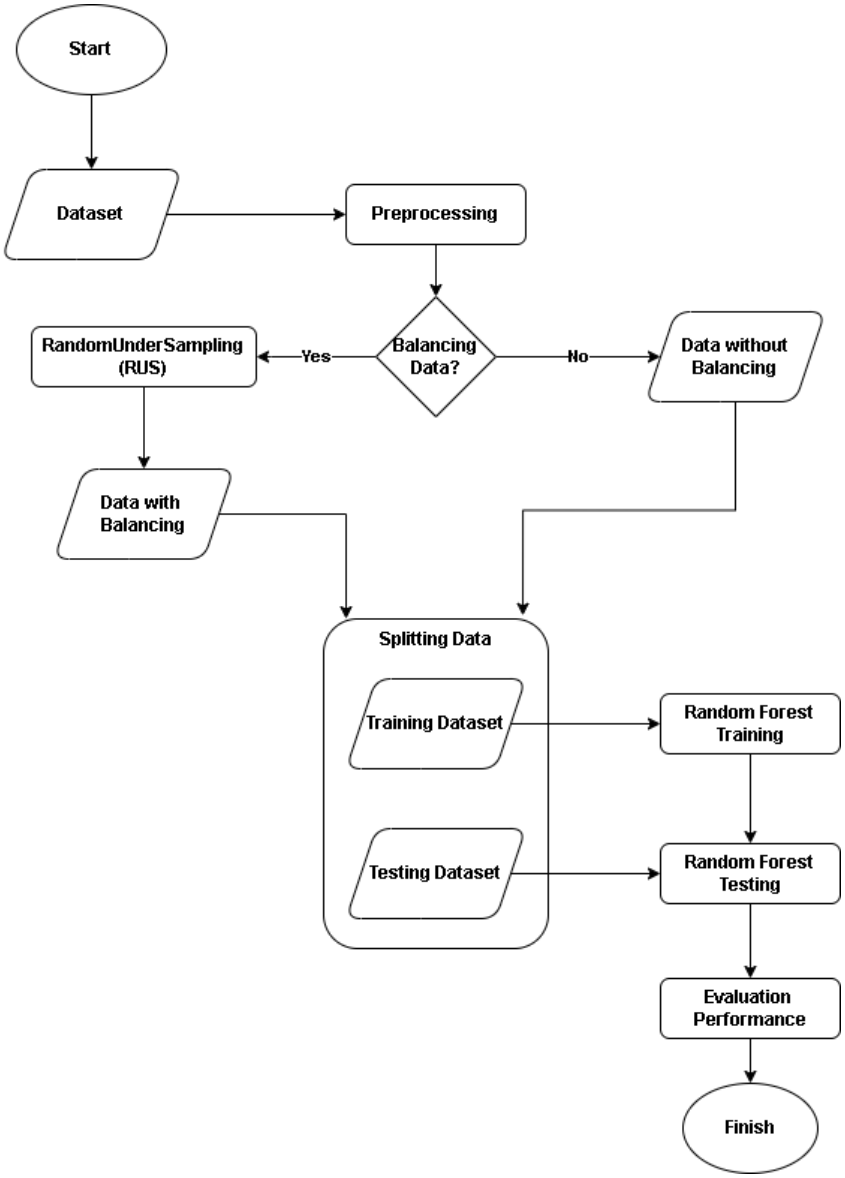
Pada (Gambar 2-1) menunjukkan model alur model algoritma pada deteksi kecurangan dalam transaksi bank. Dalam kasus ini *Random Forest* merupakan

gabungan dari setiap pohon terhadap penipuan transaksi yang baik yang kemudian digabungkan menjadi satu model[9]. Tahapan awal dilakukan pada dataset yang berisi data transaksi bank dibagi menjadi beberapa subset acak. Subset ini digunakan untuk melatih masing-masing *decision tree* secara independen. Pada setiap *decision tree* dilatih menggunakan subset yang berbeda, sehingga *decision tree* tersebut belajar untuk mengenali pola-pola spesifik yang mungkin menunjukkan indikasi kecurangan(*fraud*). Setelah dilakukan pelatihan terhadap setiap *decision tree*, model akan menghasilkan prediksi apakah sebuah transaksi merupakan kecurangan(*fraud*) atau bukan(*non-fraud*), berdasarkan subset data yang dilatih. Setelah semua *decision tree* memberikan prediksi, hasil prediksi digabungkan melalui proses *majority voting*, dimana prediksi yang paling sering muncul di antara semua *decision tree* akan menjadi prediksi akhir.

3. PERANCANGAN SISTEM

3.1. Desain Sistem

Pada penelitian ini akan dilakukan deteksi data transaksi yang dicurigai sebagai *fraud* dengan data yang telah diolah aplikasi *Solve Ease Fraud Investigation System* (SEFIS). Tahapan sistem deteksi kecurangan(*fraud*) meliputi: a. *Preprocessing*, b. *Balancing Data*, c. *Splitting dataset*, d. *Hyperparameter tuning*, e. *Confusion Matrix*



Gambar 3-1: *Flowchart* Desain Sistem

3.2. Dataset

Dataset yang digunakan penelitian adalah data sekunder yang dikembangkan oleh aplikasi *Solve Ease Fraud Investigation System* (SEFIS). Dataset yang digunakan merupakan data transaksi yang diperoleh dari pengolahan aplikasi yang diambil dari bank tertentu dalam jangka waktu 01 November 2023 hingga 19 November 2023. Dataset memiliki data berjumlah 213011 dan fitur berjumlah 33 fitur dengan gambaran data terdapat pada (Tabel 3-1) berikut.

Tabel 3-1: *Sample Dataset Transaction*

Uttrno	Acct1	Acct2	Amount	Is_alerted	Trans_type	Trans_type_desc	Sysdate
1	12341251231	2512342423	10000	N	101	Transfer	2023-11-01 22:47:02.668
2	363453457	556464643		N	107	QRIS Payment	2023-11-02 11:20:32.542
3	5345345		10000000	N			2023-11-03 08:20:51.222
4	10205674566		1400000	Y		Transfer	2023-11-04 13:38:02.861
5	6756756	2435262	32000	N	106	BIFAST Transfer	2023-11-05 17:39:32.336

Pada (Tabel 3-1) terdapat contoh fitur dan data yang ada pada dataset transaksi bank. Dalam (Tabel 3-1) terdapat fitur “uttrno” sebagai kode unik dari suatu transaksi, “acct1” sebagai nomor rekening pengirim, “acct2” sebagai nomor rekening tujuan, “amount” sebagai nominal transaksi, “is_alerted” sebagai penanda bahwa data tersebut dicurigai sebagai *fraud* atau tidak, dan “sysdate” sebagai tanggal dan waktu dilakukan transaksi.

3.3. *Preprocessing Data*

Pada penelitian ini, tahapan *preprocessing data* dilakukan menjadi beberapa bagian yang terdiri dari a. *Load Dataset*, b. Reduksi Dimensi Fitur, c. Penyesuaian Tipe Data, d. *Handling Missing Value*

3.3.1. *Load Dataset*

Tahapan awal dalam preprocessing data dengan melakukan load dataset menggunakan perintah read_csv yang ada pada *library* pandas. Pandas melakukan load data terhadap dataset dengan format '.csv' yang telah disimpan di google drive.

3.3.2. Reduksi Dimensi Fitur

Setelah melakukan proses load dataset, kami melakukan analisis untuk menentukan fitur-fitur yang perlu dihapus berdasarkan beberapa kriteria. Dari 33 fitur yang tersedia, kami memutuskan untuk menggunakan 21 fitur, yang dianggap lebih relevan untuk analisis dan model yang akan dibangun.

Beberapa fitur dihapus karena termasuk dalam kategori fitur informasi tambahan yang tidak memberikan kontribusi signifikan terhadap prediksi. Contohnya adalah fitur '#', yang berfungsi sebagai penanda atau identifikasi tetapi tidak memiliki nilai analitis yang penting.

Selanjutnya, beberapa fitur dihapus karena seluruh nilainya kosong. Fitur seperti 'pos_data_code', 'prc_code', dan 'acct_balance' dihilangkan karena tidak memberikan informasi apapun yang dapat digunakan dalam proses pembelajaran model.

Ada juga fitur-fitur yang dihapus karena informasinya telah direpresentasikan oleh fitur lain yang lebih efisien. Fitur 'ttime', 'udate', dan 'capt_date' dihapus karena seluruh informasi yang terkandung dalam ketiga fitur ini sudah tergabung dalam satu fitur, yaitu 'sysdate', yang menyederhanakan struktur data tanpa kehilangan informasi penting. Fitur 'cvt_amount' juga dihapus karena seluruh nilainya identik dengan fitur 'amount', sehingga dianggap tidak memberikan informasi tambahan. Selain itu, fitur 'resp_code_desc' dihapus karena sudah direpresentasikan oleh fitur 'resp_code', dan fitur 'trans_type_desc' dihapus karena informasinya sudah tercakup dalam fitur 'trans_type'.

Beberapa fitur dihapus karena seluruh nilai di dalamnya sama, sehingga tidak memberikan variasi data yang dapat digunakan untuk analisis. Fitur 'currency' memiliki nilai seragam '360' dan 'ID', yang keduanya merujuk pada mata uang Indonesia (Rupiah), dan fitur 'fraud_flags' memiliki seluruh nilai 0. Kedua fitur ini tidak memberikan informasi tambahan yang berguna untuk model prediksi, sehingga dihapus.

Penghapusan fitur-fitur ini merupakan bagian dari upaya menyederhanakan dataset, serta meningkatkan efisiensi komputasi untuk mendapatkan kualitas prediksi yang lebih baik pada model algoritma *Random Forest*.

3.3.3. Penyesuaian Tipe Data

Langkah selanjutnya adalah penyesuaian tipe data dengan menggunakan *Label Encoding*, yang berfungsi untuk mengubah fitur-fitur yang berisi nilai teks menjadi angka menggunakan *library LabelEncoder* dari sklearn. Hal ini penting dilakukan karena algoritma *Random Forest* hanya dapat bekerja dengan fitur-fitur yang memiliki tipe data numerik. Teks tidak bisa langsung digunakan dalam algoritma ini karena tidak memiliki nilai numerik yang dapat dibandingkan dalam proses pembentukan pohon keputusan (*decision trees*).

Dalam *Label Encoding*, setiap kategori dalam fitur diubah menjadi angka unik. Sebagai contoh, jika fitur 'is_alerted' memiliki kategori 'Y' untuk transaksi yang diduga fraud dan 'N' untuk transaksi normal, maka kategori-kategori tersebut akan diubah menjadi angka 1 untuk transaksi yang diduga fraud dan 0 untuk transaksi normal. Pada penelitian ini, fitur-fitur yang dikenai label encoding meliputi 'cif_id', 'is_alerted', 'terminal_id', 'terminal_address', 'merchant_type', 'stan', 'ref_num', dan 'sysdate'. Proses ini memastikan bahwa model dapat memahami dan memproses informasi dengan tepat sehingga dapat menghasilkan prediksi yang akurat.

3.3.4. *Handling Missing Value*

Dengan fitur-fitur yang sudah disesuaikan, dilakukan *handling missing value* dengan mengkonversi data string kosong menjadi *Not a Number* (NaN) guna

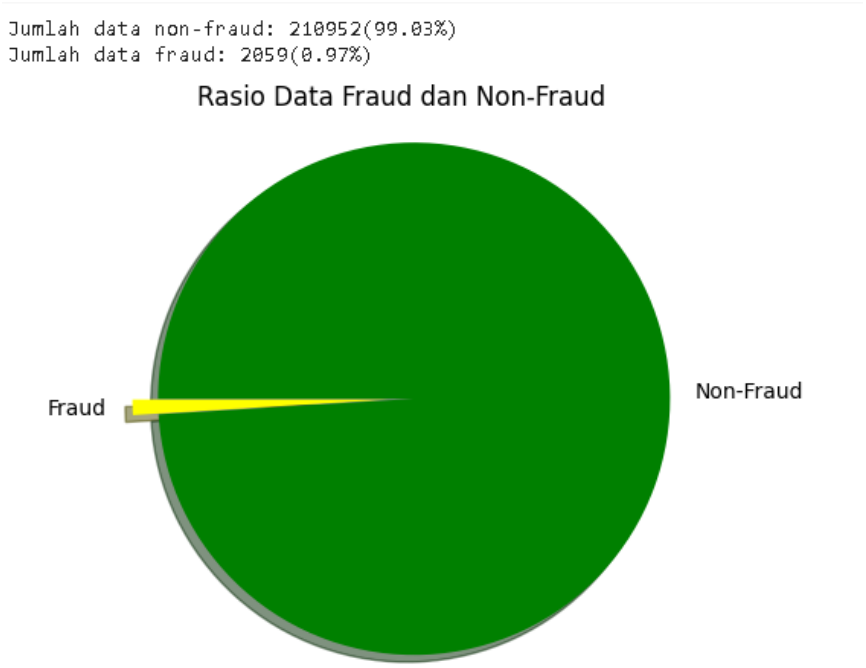
memudahkan dalam eksplorasi data yang kosong. Setelah dikonversi menjadi *Not a Number* (NAN), fitur yang memiliki nilai kosong diisi dengan *dummy value* yang berbeda dari nilai biasanya. *Dummy value* diisi dengan nilai -10.000.000 guna model akan mendeteksi sebagai pencilan data. Hal ini dilakukan untuk mempertahankan data yang ada, karena bisa jadi dari data kosong tersebut merupakan salah satu informasi yang penting. Contoh data yang telah diolah dapat dilihat pada (Tabel 3-2) berikut.

Tabel 3-2 : *Sample Dataset Transaction* yang telah dilakukan *preprocessing*

Uttrno	Acct1	Acct2	Amount	Is_alerted	Trans_type	Sysdate
1	12341251231	2512342423	10000	0	101	20231101224702668
2	363453457	556464643	-10000000	0	107	20231102112032542
3	5345345	-10000000	10000000	0	-10000000	20231103082051222
4	10205674566	-10000000	1400000	1	-10000000	20231104133802861
5	6756756	2435262	32000	0	106	20231105173932336

3.4. *Balancing Data*

Sebagai fitur target yang merupakan acuan untuk fitur lainnya, “is_alerted” perlu dilakukan pengecekan nilai yang ada didalamnya. Nilai yang ada di dalam “is_alerted” seteleh dilakukan *preprocessing* terdiri dari data transaksi yang dicurigai sebagai *fraud* yang diberi nilai 1 dan data transaksi normal (*non-fraud*) yang diberi nilai 0. Kami melakukan *Exploration Data Analysis* (EDA) terhadap fitur “is_alerted” untuk mengetahui rasio atau perbandingan antara data transaksi yang dicurigai sebagai *fraud* dan data transaksi normal (*non-fraud*).



Gambar 3-2 : rasio nilai *fraud* dan *non-fraud* pada *Imbalance Dataset*

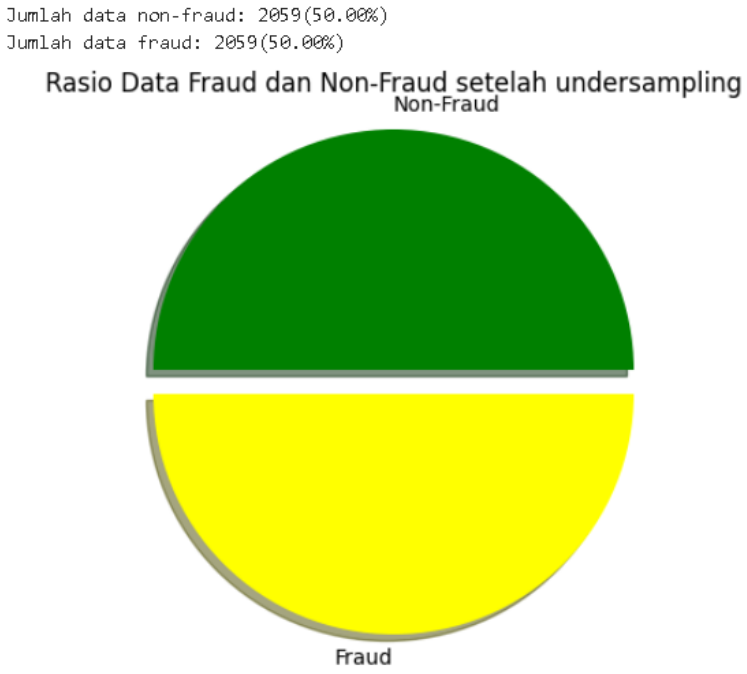
Pada (Gambar 3-2) diatas, dapat dilihat hasil EDA yang menunjukkan data transaksi normal (*non-fraud*) dan data transaksi yang dicurigai sebagai *fraud* tidak seimbang dengan perbandingan 210952 untuk data transaksi normal (*non-fraud*), dan 2059 untuk data yang dicurigai sebagai *fraud*. Pada penelitian ini, kami melakukan penerapan *Balancing* data serta melakukan komparasi guna mengetahui performa model algoritma terhadap data yang tidak seimbang. *Balancing* data akan dilakukan dengan menggunakan metode *Random Undersampling* (RUS).

3.4.1. *Random Undersampling* (RUS)

RandomUndersampling (RUS) adalah metode pengambilan sample data secara acak dengan memilih sebagian besar sample data dan menghilangkan hingga distribusi sample data yang diinginkan tercapai [10]. Pada penelitian ini, dataset memiliki karakteristik *imbalance data*. Meski *Random Forest* memiliki ketahanan terhadap *overfitting*, perlu dilakukan *balancing data* mellihat data minor atau data transaksi yang dicurigai sebagai *fraud* sangat minim dengan persentasi kurang dari satu persen. Dengan data yang sangat minim, kami berusaha untuk

mempertahankan data yang dicurigai sebagai *fraud* karena dirasa data-data tersebut merupakan informasi yang sangat penting untuk mendeteksi *fraud*. Namun kami akan melakukan ujicoba pada model algoritma *Random Forest* menggunakan dataset yang dilakukan *balancing* menggunakan *RandomUndersampling* (RUS) dan dataset tanpa *balancing*. Hal ini dilakukan untuk mengetahui pengaruh *balancing* terhadap performansi dari model algoritma *Random Forest.*

Tahapan *Balancing* data dilakukan dengan perintah *RandomUnderSampler* menggunakan *library* imblearn untuk menangani *Imbalance Data*. Parameter yang digunakan pada *RandomUnderSampler* adalah sampling_strategy, dimana diisi dengan data dari fitur target yaitu "is_alerted" yang terdiri dari data transaksi yang dicurigai sebagai *fraud* yang bernilai 1 dan data transaksi normal (*non-fraud*) yang bernilai 0. Hasil dari *RandomUndersampling* (RUS) dapat dilihat pada (Gambar 3-3) berikut.



Gambar 3-3 : rasio nilai *fraud* dan *non-fraud* pada dataset yang telah dilakukan *RandomUndersampling* (RUS)

3.5. *Splitting Data*

Tahapan *Splitting Data* adalah membagi dataset menjadi dua bagian yaitu *training data* yang digunakan untuk melatih model algoritma yang akan digunakan dan *testing data* yang digunakan untuk uji coba model algoritma. Distribusi data yang dilakukan pada tahapan *splitting data* adalah 70% untuk *training data* dan 30% untuk *testing data* yang dapat dilihat pada (Gambar 3-4) berikut.

```
Training Set:
Jumlah data di Training Set: 149107 (70.00%)

Test Set:
Jumlah data di Testing Set: 63904 (30.00%)
```

Gambar 3-4 : *Splitting Dataset*

3.6. Hyperparameter Tuning

Tahapan *Hyperparameter Tuning* dilakukan untuk menentukan nilai parameter terbaik pada setiap parameter yang digunakan dalam model algoritma *Random Forest*. Dalam penelitian ini, kami menggunakan teknik *RandomSearch* menggunakan bantuan *library* RandomizedSearchCV dari *Scikit-Learn* yang akan mencari parameter secara acak dalam rentang parameter yang kami tentukan. Parameter yang dimaksud merupakan parameter yang ada di dalam algoritma

Random Forest meliputi n_estimators dan max_depth. Parameter n_estimator adalah banyak *desicion tree* yang dibuat pada model algoritma *Random Forest*, dimana jika nilai parameter menentukan banyak *decision tree* yang dibuat guna meningkatkan performansi pada algoritma, namun memiliki dampak terhadap biaya komputasi yang tinggi. Sedangkan untuk max_dept yaitu nilai maksimum untuk kedalaman dari suatu *decision tree* pada algoritma *Random Forest,* dimana jika nilai pada max_dept yang digunakan tinggi dapat menangani *overfitting* sedangkan jika nilai rendah maka dapat menangani *underfitting*.

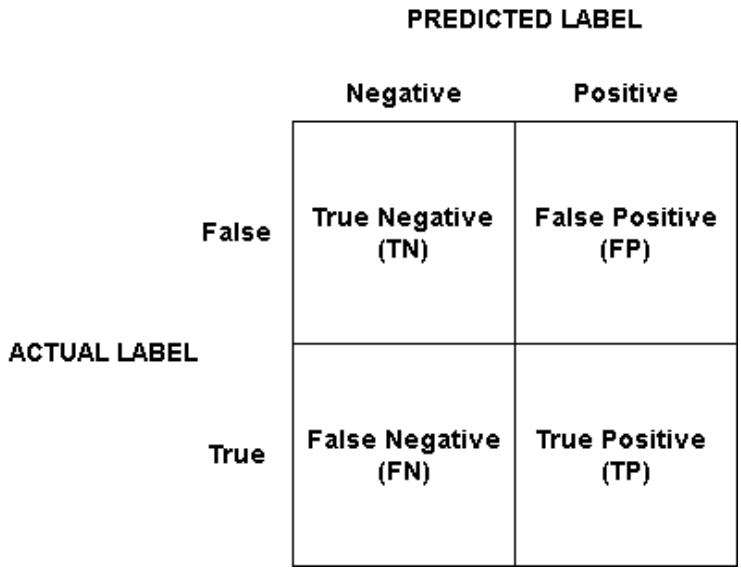
Tabel 3-3 : Hasil *Hyperparamter Tuning* menggunakan *RandomSearch*

Parameter	**Nilai**
n_estimator	164
max_depth	17

Pada (Tabel 3-3) hasil dari *hyperparameter tuning* dengan *RandomSearch*, menunjukkan hasil nilai n_estimator adalah 164 dari rentang nilai 50 hingga 500 yang dilakukan pengambilan nilai secara acak, dan max_dept adalah 17 dari rengant nilai 1 hingga 20 yang dilakukan pengambilan nilai secara acak. Nilai parameter ini akan kami gunakan pada model algoritma *Random Forest* untuk kasus deteksi kecurangan(*fraud*) pada transaksi bank.

3.7. Confusion Matrix

Confusion Matrix merupakan tahapan yang digunakan untuk mengukur performa algoritma dalam bentuk tabel matriks [3]. Matriks ini dilakukan terhadap algoritma klasifikasi untuk menunjukan sejauh mana model algoritma dapat mengklasifikasikan data dengan benar.



Gambar 3-5 : Tabel *Confusion Matrix*

Pada (Gambar 2-2) dengan nilai dan label pada *confusion matrix* terdapat kombinasi nilai pada label aktual dan prediksi. Nilai *positive* merujuk pada data transaksi yang dicurigai sebagai *fraud* dan nilai *negative* merujuk pada data transaksi normal(*non-fraud*). Pada penelitian ini, kami membentuk *confusion matrix* berdasarkan kasus deteksi kecurangan pada transaksi bank yang terdiri atas empat kombinasi nilai dengan kondisi meliputi:

1. *True Positive* (TP): model memprediksi data transaksi dicurigai sebagai *fraud*, dan nilai asli (*actual*) data transaksi merupakan *fraud.*
2. *False Positive* (FP): model memprediksi data transaksi dicurigai sebagai *fraud*, namun nilai asli (*actual*) data transaksi normal. Kombinasi ini disebut juga dengan *Type-I Error*.
3. *False Negative* (FN): model memprediksi data transaksi normal, namun nilai asli (*actual*) data transaksi merupakan *fraud.* Kombinasi ini disebut juga dengan *Type-II Error.*
4. *True Negative* (TN): model memprediksi data transaksi normal, dan nilai asli (*actual*) data transaksi normal.

Berdasarkan kombinasi nilai dari *confusion matrix* yang telah kita ketahui, kita dapat hitung dan mencari tahu beberapa nilai performansi metriks untuk tolak ukur evaluasi model adalah *Accuracy, Precision, Recall, Specificity,* dan *F1-score.*

3.7.1. *Accuracy*

Accuracy merupakan jumlah total dari berapa sering model memuat klasifikasi dengan benar. Nilai *Accuracy* dihitung oleh kedekatan nilai prediksi (*predicted*) dengan nilai asli (*actual*) yang dibandingkan dengan keseluruhan nilai kombinasi. *Accuracy* yang baik dicapai saat nilai *accuracy* adalah 1, kondisi dimana data transaksi normal(*non-fraud*), dan data transaksi yang dicurigai sebagai *fraud* diklasifikasikan dengan benar. Persamaan *Accuracy* dapat dilihat pada persamaan (1) berikut:

$$Accuracy = \frac{TP+TN}{TP+FP++TN+FN} \quad \textbf{(}1\text{)}$$

3.7.2. *Precision*

Precision merupakan jumlah total dari prediksi *positive* dengan nilai asli (*actual*) benar atau *True Positive*(TP). Nilai *Precision* dihitung dengan nilai *True Positive* (TP) relatif pada prediksi *positive* (TP + FP). *Precision* yang baik dicapai saat nilai *precision* adalah 1, kondisi dimana semua prediksi *positive* merupakan nilai asli (*actual*) *positive*. Persamaan *Precision* dapat dilihat pada persamaan (2) berikut:

$$Precision = \frac{TP}{TP+FP} \quad \textbf{(}2\text{)}$$

3.7.3. *Recall*

Recall merupakan gambaran performansi dari keberhasilan klasifikasi mendapatkan nilai *True Positive* (TP). Nilai *Recall* dihitung dengan nilai *True Positive* (TP) dibagi dengan nilai asli (*actual*) *positive* (TP + FN). *Recall* yang baik dicapai saat nilai *recall* adalah 1, kondisi dimana semua nilai asli (*actual*) *positive* dan diprediksi *positive*. Persamaan *Recall* dapat dilihat pada persamaan (3) berikut:

$$Recall = \frac{TP}{TP+FN} \quad \textbf{(}3\text{)}$$

3.7.4. *Specificity*

Specificity merupakan kebalikan dari *Recall*, dimana gambaran performansi dari keberhasilan klasifikasi mendapatkan nilai *True Negative* (TN). Nilai *Specificity* dihitung dengan nilai asli (*actual*) *True Negative* (*TN*) dibagi dengan nilai asli (*actual*) *negative* (TN + FP). *Specificity* yang baik dicapai saat nilai *specificity* adalah 1, kondisi dimana semua nilai asli (*actual*) *negative* diprediksi sebagai *negative*. Persamaan *Specificity* dapat dilihat pada persamaan berikut:

$$Specificity = \frac{TN}{TN+FP} \quad \textbf{(}4\text{)}$$

3.7.5. *F1-Score*

F1-Score merupakan gambaran performansi dari cerminan keseimbangan antara *Precision* dan *Recall*. Nilai *F1-Score* dihitung dengan *harmonic mean* antara *Precision* dan *Recall*. Persamaan *F1-Score* dapat dilihat pada persamaan (5) berikut:

$$F1\_score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad \textbf{(}5\text{)}$$

4. PENGUJIAN DAN ANALISIS

4.1. Skenario Uji Coba

Dalam penelitian ini, dilakukan pengujian analisis performansi. Analisis performansi dilakukan untuk mengukur kinerja algoritma *Random Forest* terhadap dataset yang dilakukan *Balancing* menggunakan *RandomUndersampling*(RUS) dengan dataset yang dibiarkan data utuh tanpa dilakukan *Balancing* data.

4.2.Analisis Performansi Matriks

Dalam penelitian ini, kami membandingkan nilai *accuracy*, *precision*, *recall*, *specificity*, dan *F1-Score* dengan dua skenario pada dataset *training* dan *testing*. Skenario pertama melibatkan klasifikasi data asli menggunakan model algoritma *Random Forest* tanpa metode *Balancing* Dataset. Skenario kedua melibatkan klasifikasi data yang telah diolah dengan metode *Balancing* Dataset menggunakan *RandomUndersampling*(RUS). Pembandingan antara dua skenario dilakukan guna mengetahui peningkatan performansi terhadap model *Random Forest* dalam mendeteksi kecurangan(*fraud*).

4.2.1. Analisis Performansi Matriks untuk setiap skenario pada *training dataset*

Tabel 4-1 : Nilai Performansi Matriks pada *Training Dataset*

Sampling	Performansi	Hasil
Tanpa Balancing method (Skenario pertama)	Accuracy	0.9998
	Precision	0.9978
	Recall	0.9886
	Specificity	0.9999
	F1-Score	0.9932
Menggunakan Balancing method	Accuracy	1.0
	Precision	1.0

(Skenario Kedua)	Recall	1.0
	Specificity	1.0
	F1-Score	1.0

Dari hasil eksperimen diatas, menunjukkan bahwa pada (Tabel 4-1) performansi pada skenario kedua lebih unggul dengan seluruh nilai performansi matriks berada pada nilai terbaik dengan seluruh nilai adalah 1, yang menandakan bahwa skenario pertama lebih unggul terhadap dataset *training* dibanding dengan skenario pertama. Meskipun demikian, skenario pertama juga memiliki nilai performansi matriks yang sangat baik dengan seluruh nilai performansi matriks mendekati 1. Hal ini menunjukkan bahwa model algoritma *Random Forest* cocok terhadap kasus deteksi kecurangan(*fraud*). Dengan dilakukan *Balancing* data menggunakan *RandomUnderSampling*(RUS), model algoritma *Random Forest* lebih fokus dan efektif untuk melatih data transaksi dalam mendeteksi data dicurigai sebagai kecurangan(*fraud*) yang jarang terjadi tanpa perlu mengorbankan kemampuan model dalam mendeteksi data transaksi normal(*non-fraud*).

4.2.2. Analisis Performansi Matriks untuk setiap skenario pada Pengujian model dengan *testing dataset*

Setelah melihat nilai performansi matriks pada pelatihan model sebelumnya, perlu dilakukan pengujian model dengan dataset yang digunakan adalah dataset *testing.* Model algoritma *Random Forest* dilakukan pengujian guna mengetahui hasil dari setiap skenario pengujian dalam melakukan deteksi kecurangan(*fraud*). Pada (Tabel 4-2) berikut adalah hasil analisis performansi metriks terhadap model algoritma *Random Forest* dalam mendeteksi *fraud* pada dataset *testing* yang dilakukan *Balancing* menggunakan *RandomUnderSampling*(RUS) dan tanpa *Balancing*.

Tabel 4-2: Nilai Performansi Matriks pada *Testing Dataset*

Sampling	Performansi	Hasil
Tanpa Balancing method (Skenario pertama)	Accuracy	0.9986
	Precision	0.9947
	Recall	0.8732
	Specificity	0.9999
	F1-Score	0.9300
Menggunakan Balancing method (Skenario Kedua)	Accuracy	0.9927
	Precision	0.9799
	Recall	0.9949
	Specificity	0.9999
	F1-Score	0.9873

Dari hasil eksperimen diatas, menunjukan bahwa pada (Tabel 4-2) skenario pertama memiliki keunggulan pada nilai *accuracy*, dan *precision*, sedangkan pada skenario kedua menunjukkan nilai *recall*, dan *F1-score*. Hal ini membuktikan bahwa metode tanpa *Balancing* data model algoritma *Random Forest* memiliki performa yang sangat baik dalam mendeteksi transaksi normal(*non-fraud*) dengan nilai *precision*, dan *accuracy* yang sangat tinggi. Namun, nilai *Recall* yang lebih rendah menunjukkan bahwa beberapa transaksi yang dicurigai sebagai *fraud* tidak terdeteksi dengan baik, yang menyebabkan resiko kecurangan yang dilakukan masih belum ditangani dengan baik. Sedangkan dengan dilakukan *Balancing* menggunakan *RandomUnderSampler* (RUS) mengalami peningkatan pada nilai *Recall*, yang membuat model algoritma *Random Forest* lebih ketat dalam mendeteksi transaksi yang dicurigai sebagai *fraud*. Meskipun nilai *Precision* sedikit menurun, namun hal ini meningkatkan nilai *F1-score* secara keseluruhan, yang menunjukkan dengan dilakukan *Balancing* data menggunakan

RandomUnderSampler (RUS) lebih efektif untuk mendeteksi data transaksi yang dicurigai sebagai *fraud* dengan keseimbangan nilai *Precision*, dan *Recall*.

Dalam hasil pengujian pada dataset *testing*, penerapan dengan metode *Balancing* data menggunakan *RandomUnderSampling* (RUS) lebih efektif dibanding tanpa menggunakan *Balancing* data, sebab dalam penelitian ini, fokus utama kami adalah mendeteksi transaksi yang dicurigai sebagai kecurangan(*fraud*). Hal ini menunjukkan bahwa dengan fokus utama terhadap nilai yang diindikasi sebagai kecurangan(*fraud*) menjadikan nilai *F1-Score* menjadi acuan utama dibanding dengan nilai *Accuracy*. Hasil dari model algoritma *Random Forest* dengan *Balancing* data lebih sensitif dan ketat dalam mendeteksi kecurangan yang terjadi pada data transaksi, meski dalam deteksi perlu mengorbankan sedikit *Precision*, dan *accuracy*.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil dari ujicoba dan analisa yang telah dilakukan pada bagian sebelumnya, maka dapat disimpulkan:

1. Penerapan model algoritma *Random Forest* pada kasus deteksi kecurangan(*fraud*) data transaksi bank sangat cocok.
2. Hasil pengujian performansi yang diterapkan pada kedua skenario memiliki hasil yang sangat baik dengan seluruh nilai dari kedua skenario memiliki nilai yang mendekati 1.
3. Penerapan metode *Balancing* data dengan *RandomUndersampling*(RUS) dengan fokus utama terhadap nilai yang diindikasi sebagai kecurangan(*fraud*) menjadikan nilai *F1-Score* menjadi acuan utama dibanding dengan nilai *Accuracy* dengan nilai *Recall* yang meningkat dari 0.8732 menjadi 0.9949, dan nilai *F1-Score* dari 0.9300 menjadi 0.9873 yang memiliki dampak bagus bagi deteksi kecurangan(*fraud*).

5.2. Saran

Berdasarkan analisa serta hasil yang telah dilakukan, kami memiliki saran sebagai berikut:

1. Perlu dilakukan ujicoba untuk penganganan *imbalance* data dengan metode atau algoritma lain, seperti *Anomaly Detection*, *Ensemble Methods*, dan algoritma lainnya.
2. Perlu dilakukan ujicoba untuk *hyperparameter tuning* dengan metode lain seperti *Grid Search*, *Bayesian Optimization*, dan metode lainnya

DAFTAR PUSTAKA

[1] Y. Liu, Z. Tang, and W. Zheng, *Suspicious Bank Card Transaction Recognition Based on K-means Clustering and Random Forest Algorithm*. IEEE, 2019.

[2] M. Sopiyan, Fauziah, and Y. F. Wijaya, "Fraud Detection Using Random Forest Classifier, Logistic Regression, and Gradient Boosting Classifier Algorithms on Credit Cards," 2022.

[3] K. Deepika, M. P. S. Nagenddra, M. V. Ganesh, and N. Naresh, "Implementation of Credit Card Fraud Detection Using Random Forest Algorithm," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 3, pp. 797–804, Mar. 2022, doi: 10.22214/ijraset.2022.40702.

[4] L. S. V S S and S. Deepthi Kavila, "Machine Learning For Credit Card Fraud Detection System," 2018. [Online]. Available: http://www.ripublication.com

[5] D. Shah and L. Kumar Sharma, "Credit Card Fraud Detection using Decision Tree and Random Forest," *ITM Web of Conferences*, vol. 53, p. 02012, 2023, doi: 10.1051/itmconf/20235302012.

[6] O. J. Unogwu and Y. Filali, "Fraud Detection and Identification in Credit Card Based on Machine Learning Techniques," 2023, doi: 10.31185/wjcm.185.

[7] D. Alita and A. Rahman, "Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier," 2020.

[8] G. Niveditha, K. Abarna, and G. V. Akshaya, "Credit Card Fraud Detection Using Random Forest Algorithm," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 301–306, Mar. 2019, doi: 10.32628/cseit195261.

[9] A. Saputra and Suharjito, "Fraud Detection using Machine Learning in e-Commerce," 2019. [Online]. Available: www.ijacsa.thesai.org

[10] D. Trisanto, N. Rismawati, M. F. Mulya, and F. I. Kurniadi, "Effectiveness Undersampling Method and Feature Reduction in Credit Card Fraud Detection," *International Journal of Intelligent Engineering and Systems (IJIES)*, 2020.