Perbandingan Model Machine Learning dan Deep Learning dalam Deteksi Penipuan Transaksi Bank Digital dengan Optimasi Hyperparameter Tuning Menggunakan Gridsearch CV

Hendri Mardani

Fakultas Teknik Program Studi Sistem Informasi

19231232@bsi.ac.id

ABSTRAK

Kemajuan teknologi informasi yang sangat cepat dalam hal transaksi digital menjadi sangat mudah salah satunya penipuan dalam transaksi bank digital, sehingga memerlukan sistem yang melakukan deteksi transaksi yang tidak wajar untuk mengetahui kasus penipuan. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sebuah model mengklasifikasi penipuan mulai dari data preprocessing sampai feature extraction (pemilihan fitur) apa saja yang paling mempengaruhi fitur target. Penelitian ini menggunakan pendekatan kuantitatif dengan desain eksperimental. Desain ini dipilih untuk membandingkan beberapa model machine learning dalam tugas klasifikasi penipuan transaksi digital dengan metrik yang digunakan sebagai acuan performa model salah satunya akurasi. Dilakukan beberapa metode feature extraction seperti VarianceThreshold, SelectKBest dan Recursive Reature Elimination (RFE) dengan hasil yang didapat sama tetapi fitur yang dipilih berbeda. Porsi data training dan data testing sebesar 80:20 dengan menggunakan train_test_split. Data yang tidak seragam antar kelas positif dan negatif terhadap fitur target dilakukan teknik index slicing terhadap jumlah data yang besar. Percobaan pelatihan menggunakan metode Random Forest hasil akurasi didapat akurasi training 1.0 dan akurasi validasi 0.93, kemudian lakukan perbandingan dengan beberapa model, model yang paling unggul adalah metode Logstic Regression dan metode SVC dengan akurasi 0.941. Lalu melakukan hyperparameter tuning menggunakan gridsearch CV pada metode Random Forest hasil didapat ada peningkatan sebelum dan sesudah dilakukan hyperparameter tuning. Terakhir melakukan pelatihan metode lain yakni metode deep learning. Hasil evaluasi didapat metode deep learning

lebih unggul dengan akurasi 0.9412. Analisis ini diharapkan mampu menjadi masukan sekaligus insight untuk para peneliti dimasa yang akan datang.

Kata Kunci: penipuan, hyperparamter tuning, gridsearch CV, deep learning, machine learning

I. PENDAHULUAN

Bank Indonesia (BI) memiliki andil besar dalam evolusi sistem pembayaran digital Indonesia guna memenuhi tuntutan masyarakat akan transaksi yang efisien dan terjamin. Melalui implementasi Gerakan Nasional Non Tunai (GNNT) serta pembentukan kerangka regulasi yang mendukung inovasi fintech, BI berupaya menekan ketergantungan pada uang tunai seraya meningkatkan inklusi keuangan. Prioritas pada keamanan transaksi juga diwujudkan melalui pembangunan infrastruktur tangguh dan proteksi konsumen, di mana langkah-langkah ini secara kolektif telah menjadi kontributor utama transformasi sistem pembayaran Indonesia (Rangkuti *et al.*, 2024).

Kemajuan teknologi informasi yang sangat cepat dalam beberapa dekade terakhir telah membawa dampak besar pada berbagai bidang, termasuk sektor keuangan digital. Penggunaan sistem pembayaran digital seperti e-commerce, pinjaman peer-to-peer, serta layanan fintech lainnya kini semakin berkembang. Walaupun teknologi ini memberikan kemudahan dan efisiensi bagi para pengguna, di sisi lain juga membuka celah bagi tindakan penipuan yang kian kompleks (Pratama Adiwijaya and Sukma Maulana, 2023). Tindakan penipuan dalam transaksi digital dapat menyebabkan kerugian ekonomi yang signifikan, mencoreng reputasi, serta mengurangi tingkat kepercayaan konsumen terhadap sistem pembayaran digital. Oleh karena itu, upaya mendeteksi penipuan menjadi tantangan krusial yang harus ditangani oleh penyedia layanan maupun institusi keuangan.

Machine learning adalah konsep di mana komputer dapat belajar secara mandiri tanpa perlu diprogram secara eksplisit untuk setiap tugas. Kemampuan belajar mandiri inilah yang memungkinkan komputer menyelesaikan berbagai hal dengan mudah. Contohnya, sebuah komputer dianggap sedang "belajar" jika kemampuannya untuk menang dalam sebuah permainan terus meningkat seiring bertambahnya pengalaman bermain (Banerjee, 2024). Teknologi machine learning (ML) dan deteksi anomali menawarkan solusi andal untuk melawan penipuan. Secara khusus, ML mampu memproses data berskala besar untuk menemukan pola transaksi yang mencurigakan, sehingga deteksi

penipuan menjadi lebih akurat dan cepat dibandingkan metode tradisional yang berbasis aturan (Eldo *et al.*, 2024). Dalam kasus ini, algoritma deteksi anomali berfungsi untuk mengidentifikasi pola transaksi yang ganjil sebagai indikasi kemungkinan adanya penipuan (Japit *et al.*, 2024).

Random Forest merupakan teknik ensemble yang mengombinasikan sejumlah pohon keputusan untuk meningkatkan performa model, dan dikenal luas karena efektivitasnya dalam menyelesaikan tugas regresi maupun klasifikasi (Subagio and Utama, 2025). Selain itu metode Random Forest ini cukup popular dalam kasus klasifikasi dibandingkan metode lainnya seperti Logistik Regresi, Naive Bayes, Decision Tree, Support Vector Machine (SVM) dan sebagainya. Banyak perusahaan teknologi finansial (fintech) kini mengadopsi machine learning untuk membangun sistem anti-penipuan yang canggih demi meningkatkan keamanan pengguna. Di platform pinjaman *peer-to-peer* (P2P), misalnya, algoritma ini mampu menganalisis risiko kredit secara lebih akurat untuk menekan potensi penipuan saat pengajuan pinjaman (Pratama Adiwijaya and Sukma Maulana, 2023).

Selain teknologi machine learning ada juga teknologi deep learning yang mana algoritma ini seluk beluk dari machine learning itu sendiri dan dibuat menggunakan matematika khususnya bidang statistika dan kalkulus dengan menggunakan rumus tersendiri. Menurut (Taye, 2023) Deep learning merupakan salah satu pendekatan utama dalam machine learning yang diterapkan di banyak area. Proses pembelajarannya dapat diibaratkan dengan perkembangan kognitif bayi, di mana koneksi antar neuron di otak dimodifikasi melalui pengalaman seperti saat belajar mengidentifikasi berbagai objek. Dalam analogi ini, jika kecerdasan buatan adalah otak, machine learning adalah proses akuisisi kemampuan kognitif, dan deep learning berfungsi sebagai sistem pelatihan mandiri yang paling efektif saat ini.

Dengan teknologi yang sangat pesat sampai saat ini banyak metode-metode yang baru bermunculan, pada penilitian ini akan menggunakan metode random forest karena metode ini paling populer disemua kalangan metode dalam kasus klasifikasi. Selain itu, dalam tahap evaluasi model akan dilakukan perbandingan dengan metode klasifikasi lain menggunakan cross validation sebagai percobaan training sebanyak 5 iterasi pada setiap modelnya, lalu dilakukan hyperparameter tuning menggunakan teknik gridsearch CV.

Menurut (Bui *et al.*, 2025) untuk meningkatkan performa model digunakan teknik gridsearch CV guna menemukan kombinasi hyperparameter terbaik dan cross validation untuk mencegah overfitting. Tetapi pada penelitian ini tidak melakukan hyperparameter tuning lebih lanjut pada metode deep learning, peneliti hanya berfokus pada machine learning klasik.

Pada penelitian ini dilakukan feature extraction (pemilihan fitur) dengan menggunakan beberapa metode seperti VarianceThreshold, SelectKBest dan Recursive Model Elimination (RFE). Menurut (Es-Sabery *et al.*, 2022) kinerja sebuah model klasifikasi sangat bergantung pada kualitas fitur yang diekstraksi, sehingga menjadikan ekstraksi fitur sebagai langkah paling krusial. Pemilihan fitur yang tepat, yang umumnya dilakukan melalui algoritma ekstraksi dan seleksi, sangat menentukan performa model. Oleh karena itu, diperlukan pemikiran dan konsepsi yang matang untuk mendefinisikan fitur terbaik untuk tugas klasifikasi. Dalam penelitian ini akan dibandingkan juga hasil metode-metode feature extraction dan akan digunakan metode yang terbaik sekaligus yang cukup popular dikalangan peneliti guna untuk meningkatkan kualitas performa model machine learning. Dataset yang digunakan memiliki data yang tidak seragam kelas negatif condong lebih banyak daripada kelas positif untuk itu peneliti melakukan penyeimbangan data dengan mengikuti jumlah data pada kelas positif dengan melakukan teknik *slicing index*.

1. **Tujuan Penelitian**

   Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sebuah model mengklasifikasi penipuan mulai dari data preprocessing sampai feature extraction (pemilihan fitur) apa saja yang paling mempengaruhi fitur target. Selain itu, dalam penelitian ini dilakukan perbandingan model klasifikasi dengan metode lain dan sebagai penutup supaya mendapatkan hasil yang lebih optimal dilakukan hyperparameter tuning menggunakan gridsearch CV.

II. STUDI LITERATUR

Penelitian sebelumnya telah banyak dilakukan untuk mengindentifikasi pnedekatan dalam mendeteksi penipuan pada sistem transaksi digital salah satunya dilakukan oleh (Reyhand Ardhitha, Revifal Anugerah and Tata Sutabri, 2025) dengan judul “Analisis

Penerapan Machine Learning dan Algoritma Anomali untuk Deteksi Penipuan pada Transaksi Digital" hasil penelitian menunjukan bahwa saat dibandingkan, SVM efektif menangani data tidak seimbang namun berisiko overfitting. Random Forest lebih unggul dalam stabilitas, akurasi, dan recall, tetapi lebih boros sumber daya komputasi. Sementara itu, Ensemble Learning mencapai performa deteksi penipuan tertinggi dengan menggabungkan model meskipun prosesnya paling memakan waktu dan menggunakan teknik resampling sebagai solusi untuk mengatasi data yang tidak seimbang.

Disisi lain juga terdapat peniliti lain yang terkait pendekatan yang sama pada penelitian ini yang dilakukan oleh (Malik Ibrahim, 2025) dengan judul "Analisis Kinerja Model Machine Learning untuk Mendeteksi Transaksi Fraud pada Sistem Pembayaran Online" hasil penelitian menunjukan bahwa machine Learning efektif untuk mendeteksi penipuan pembayaran online, di mana model XGBoost dan BiLSTM memberikan hasil terbaik berdasarkan metrik evaluasi yang komprehensif. Penggunaan teknik balancing data SMOTE terbukti krusial dalam meningkatkan recall, sehingga lebih banyak kasus penipuan berhasil diidentifikasi. Sebagai perbandingan, XGBoost unggul dalam kecepatan dan kemudahan interpretasi, sementara BiLSTM lebih baik dalam mengenali pola kompleks. Namun, penelitian ini memiliki keterbatasan terkait generalisasi dataset kaggle dan belum menganalisis biaya komputasi untuk implementasi di dunia nyata.

Penelitian ketiga dilakukan oleh (Homepage, Unogwu and Filali, 2023) dengan judul "Fraud Detection and Identification in Credit Card Based on Machine Learning Techniques" hasil penelitian menunjukan bahwa Metode Multi Layer Perceptron (MLP) menunjukkan performa sangat tinggi dalam mendeteksi penipuan kartu kredit, dengan raihan F1-Score 0,9998 dan akurasi 99,95% dibandingkan menggunakan metode lain seeperti Naïve Bayes dan Random Forest.

Berdasarkan hasil penelitian yang dilakukan oleh para peneliti, dapat diidentifikasi beberapa celah. Pertama sebagaian peneliti menggunakan teknik resampling atau SMOTE sebagai solusi untuk menyeimbangkan data antar kelas padahal teknik SMOTE ini tergantung dari banyaknya jumlah data antar kelas yang digunakan dan ini akan sangat kurang efektif jika teknik SMOTE ini diterapkan semua pada kasus dataset tanpa memperhatikan jumlah data antar kelasnya. Kasus kedua para peneliti tidak melakukan feature extraction (pemilihan fitur) sebelum dilakukan pelatihan pada model fungsi feature

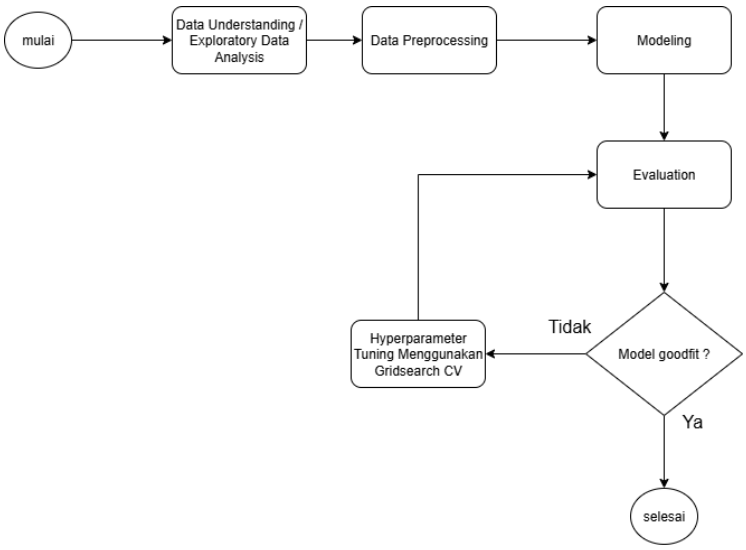
extraction berfungsi sebagai pemilahan fitur aja yang paling banyak berpengaruh terhadap fitur target atau label. Terakhir, pada kasus sistem pendeteksi transaksi para peneliti belum melakukan banyak eksplorasi pada teknik hyperparameter tuning menggunakan gridsearch CV, teknik ini sangat amat penting bagi performa model untuk menghindari terjadinya overfitting dan menyesuaikan optimize parameter model yang dipilih dengan tujuan supaya meningkatkan akurasi model.

III. METODOLOGI

Penelitian ini menggunakan pendekatan kuantitatif dengan desain eksperimental. Desain ini dipilih untuk membandingkan beberapa model machine learning dalam tugas klasifikasi penipuan transaksi digital dengan metrik yang digunakan sebagai acuan performa model salah satunya akurasi.

Dataset yang digunakan dalam penelitian ini adalah dataset sekunder (dataset publik) dengan nama "Bank Account Fraud Dataset Suite (NeurIPS 2022)" yang diperoleh dari situs kaggle. Dataset ini terdiri dari 1 juta baris data transaksi dengan 32 fitur diataranya fitur kategori seperti payment_type, employment_status, housing_status, source, device_os, dan fraud_bool sisanya adalah bentuk fitur numerik, tetapi pada fitur income ini sudah dilakukan binning. Proses ini bertujuan untuk menyederhanakan dan menyeragamkan fitur data dengan cara menangani pola yang tidak teratur dan kompleks. Hal ini dilakukan dengan mengubah input numerik menjadi kategori, mengelompokkan nilai-nilai yang serupa, dan mengatur semua fitur agar bertipe kategorikal. Tujuannya adalah untuk memastikan data kompatibel dengan model encoder dan standar untuk diolah lebih lanjut (Lee *et al.,* 2024). Karena memiliki fitur yang cukup banyak maka pada penelitian ini juga dilakukan feature extraction (pemilihan fitur) pada variabel independent berdasarkan jumlah n fitur teratas yang mana jumlah fitur yang diambil sebanyak 5 fitur, dan didapat hasilnya adalah fitur velocity_24h, velocity_4w, credit_risk_score, housing_status, month dan satu fitur untuk variabel target yaitu fitur fraud_bool

Penelitian ini dilakukan melalui lima tahapan utama sebagai berikut:



Gambar 1. Flowchart Pengolahan Data sampai Evaluasi Model

1. Data Understanding: dalam metodologi CRISP-DM, tahap pemahaman data sangat krusial, terutama pada proyek yang kompleks dan rentan terhadap kesalahan. Fase ini melibatkan pemeriksaan mendalam terhadap data yang tersedia untuk mencegah masalah tak terduga pada tahap persiapan data, yang seringkali merupakan bagian terpanjang dari proyek. Aktivitasnya mencakup mengakses dan mengeksplorasi data menggunakan alat bantu seperti tabel dan grafik (Hnatiienko, 2023).
2. Data Preprocessing: tahap penting dalam pelatihan machine learning di mana data mentah diubah atau diolah menjadi format yang terstruktur. Tujuannya adalah agar fitur-fitur dalam data tersebut dapat dengan mudah dipahami dan diinterpretasikan oleh algoritma (Patil *et al.*, 2021).
3. Modeling: pendekatan yang memanfaatkan statistika dan machine learning untuk membangun model. Dengan menganalisis data historis untuk menemukan pola, model ini bertujuan untuk membuat prediksi yang akurat tentang hasil di masa depan. Prosesnya melibatkan pemilihan algoritma yang tepat, seperti regresi linier, pohon keputusan, atau jaringan syaraf (Aiswarya A S and Rajeev, 2024).
4. Evaluation: kinerja setiap algoritma pemodelan diukur dan dibandingkan menggunakan data uji untuk menemukan model dengan performa terbaik. Penilaian ini didasarkan pada metrik seperti accuracy, precision, recall, dan f-1 score (Maulidah, 2023).

5. Hyperparameter tuning: menurut (J. Wong, T. Manderson, M. Abrahamowicz, D. L. Buckeridge, and R. Tamblyn, 2019, dalam Alinda Rahmi and Defit, 2024) Penyetelan hyperparameter adalah proses untuk menemukan setelan nilai paling optimal yang dapat meningkatkan performa sebuah model machine learning.

IV. HASIL ANALISIS

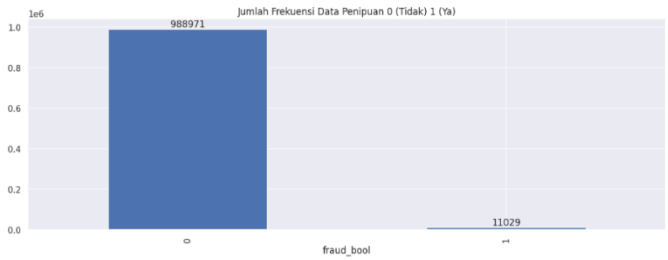
Pada pertama ini dilakukan eksplorasi pada masing-masing fitur baik fitur kategori maupun numerik dengan tujuan untuk mengetahui fitur apa saja yang saling berkorelasi atau yang mempengaruhi variabel target yakni fitur fraud_bool. Selain itu dalam tahap ini juga dilakukan pengecekan apakah terdapat data kosong, data tidak konsisten ataupun data yang duplicate atau tidak. Dengan menggunakan metode info() bawaan dari pandas bisa melakukan pengecekan apakah terdapat data kosong atau tidak. Hasil menunjukan dalam dataset ini tidak terdapat data kosong dibuktikan dengan jumlah data secara keseluruhan sudah sesuai yaitu 1 juta baris data.

Gambar 1. Pengecekan data yang kosong

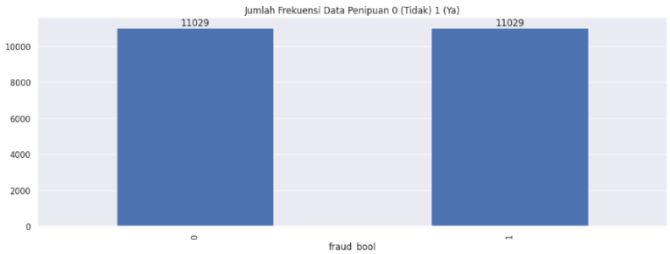
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 32 columns):
 #   Column                            Non-Null Count    Dtype
---  ------                            --------------    -----
 0   fraud_bool                        1000000 non-null  int64
 1   income                            1000000 non-null  float64
 2   name_email_similarity             1000000 non-null  float64
 3   prev_address_months_count         1000000 non-null  int64
 4   current_address_months_count      1000000 non-null  int64
 5   customer_age                      1000000 non-null  int64
 6   days_since_request                1000000 non-null  float64
 7   intended_balcon_amount            1000000 non-null  float64
 8   payment_type                      1000000 non-null  object
 9   zip_count_4w                      1000000 non-null  int64
 10  velocity_6h                       1000000 non-null  float64
 11  velocity_24h                      1000000 non-null  float64
 12  velocity_4w                       1000000 non-null  float64
 13  bank_branch_count_8w              1000000 non-null  int64
 14  date_of_birth_distinct_emails_4w  1000000 non-null  int64
 15  employment_status                 1000000 non-null  object
 16  credit_risk_score                 1000000 non-null  int64
 17  email_is_free                     1000000 non-null  int64
 18  housing_status                    1000000 non-null  object
 19  phone_home_valid                  1000000 non-null  int64
 20  phone_mobile_valid                1000000 non-null  int64
 21  bank_months_count                 1000000 non-null  int64
 22  has_other_cards                   1000000 non-null  int64
 23  proposed_credit_limit             1000000 non-null  float64
 24  foreign_request                   1000000 non-null  int64
 25  source                            1000000 non-null  object
 26  session_length_in_minutes         1000000 non-null  float64
 27  device_os                         1000000 non-null  object
 28  keep_alive_session                1000000 non-null  int64
 29  device_distinct_emails_8w         1000000 non-null  int64
 30  device_fraud_count                1000000 non-null  int64
 31  month                             1000000 non-null  int64
dtypes: float64(9), int64(18), object(5)
memory usage: 244.1+ MB
```

Pada tahap kedua melakukan penghapusan data duplicate, data kosong, dan merubah data yang tidak konsisten, lalu melakukan encoding pada fitur kategori. Untuk melakukan pengecekan data duplicate bisa menggunakan metode duplicated().sum() yang berasal dari bawaan library pandas. Setelah itu melakukan penghapusan sebanyak n jumlah data pada fitur fraud_bool yang berlabel positif, hal ini dikarenakan jumlah data berlabel positif dan negatif tidak seimbang untuk itu dilakukan penghapusan pada yang berlabel negatif mengingat jumlah label negatif lebih banyak daripada positif. Setelah itu proses encoding dilakukan pada kolom label menggunakan label encoder untuk mentransformasi label klasifikasi ke dalam bentuk numerik. Hasil transformasi ini dialokasikan ke dalam kolom baru, encoded_label, agar dapat digunakan sebagai input dalam model machine learning (Liu *et al.*, 2024). Fitur-fitur kategori itu diantaranya: payment_type, employment_status, housing_status, source, dan device_os.

Gambar 2. Sebelum Dilakukan Penyeimbangan Data



Gambar 3. Setelah Dilakukan Penyeimbangan Data



Pada tahap ketiga melakukan teknik feature extraction pada variabel independen dengan menggunakan beberapa metode seperti VarianceThreshold, SelectKBest dan Recursive Model Elimination (RFE). Metode VarianceThreshold ini bekerja dengan cara menetapkan nilai ambang batas untuk menghilangkan fitur-fitur yang memiliki varians rendah. Secara default, ia akan menghapus semua fitur bervarians nol, yaitu fitur yang

nilainya konstan di semua data. Karena hanya menganalisis hubungan antar fitur itu sendiri—bukan dengan variabel target-teknik ini dapat digunakan untuk pembelajaran terawasi maupun tidak terawasi. Meskipun didasarkan pada asumsi bahwa varians yang lebih besar berarti informasi yang lebih penting, kelemahan utamanya adalah ia mengabaikan relasi antara fitur dan target (Beshay, 2022).

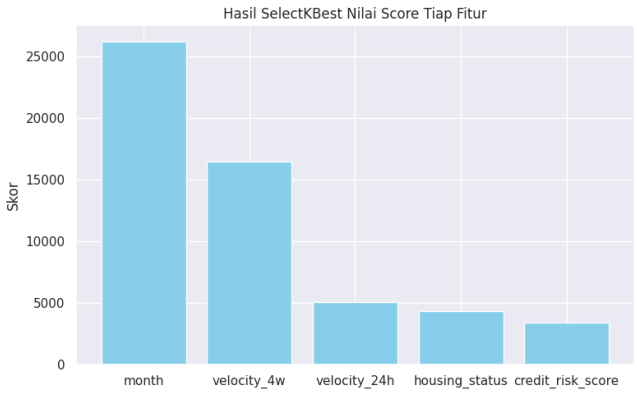
Metode SelectKBest ini bekerja dengan cara menilai setiap fitur secara individual menggunakan uji statistik univariat, lalu hanya mempertahankan fitur-fitur dengan skor tertinggi (Huseynov and Ozdenizci Kose, 2024).

Metode Recursive Feature Elimination (RFE) ini bekerja melakukan teknik seleksi fitur yang secara berulang (iteratif) melatih model dan membuang fitur yang paling tidak penting di setiap langkahnya. Proses ini terus berlanjut hingga semua fitur dievaluasi, di mana pada akhirnya fitur-fitur tersebut diberi peringkat berdasarkan urutan eliminasinya (ASIM, 2022).

Selanjutnya pada tahap ini dilakukan penghapusan outlier. Outlier dapat mengganggu parameter model, menyebabkan estimasi efek menjadi tidak akurat baik terlalu tinggi maupun terlalu rendah. Oleh karena itu, sangat penting untuk mengidentifikasi outlier dalam data (Mramba *et al.*, 2023).

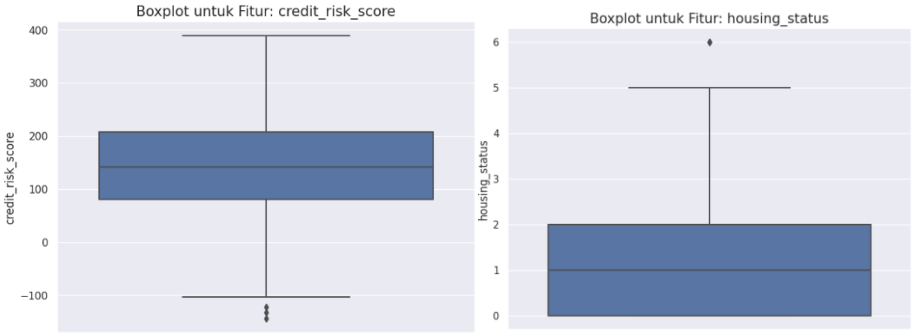
Hasil akhir dari beberapa metode feature extraction didapat semua metode mendapatkan hasil akurasi yang sempurna yaitu 1.0, akan tetapi untuk hasil pemilihan fiturnya sedikit berbeda, Pada metode VarianceThreshold didapat pemilihan seleksi terbaik sebanyak 19 fitur, hasilnya banyak karena kita perlu mengatur kembali pada parameter threshold, semakin tinggi threshold maka batas ambang variance akan sangat tinggi dan fitur-fitur yang dipilih juga akan semakin sedikit. Sedangkan pada metode SelectKBest dan metode Recursive Feature Elimination (RFE) didapat 5 fitur terbaik, metode SelectKBest mendapatkan fitur month, velocity_4w, velocity_24h, housing_status, credit_risk_score dan metode RFE didapat fitur velocity_24h, velocity_4w, velocity_6h, housing_status, month. Sehingga pada beberapa metode ini peneliti akan memilih feature extraction SelectKBest karena metode ini cukup populer disemua kalangan.

Gambar 4. Hasil Feature Extraction SelectKBest

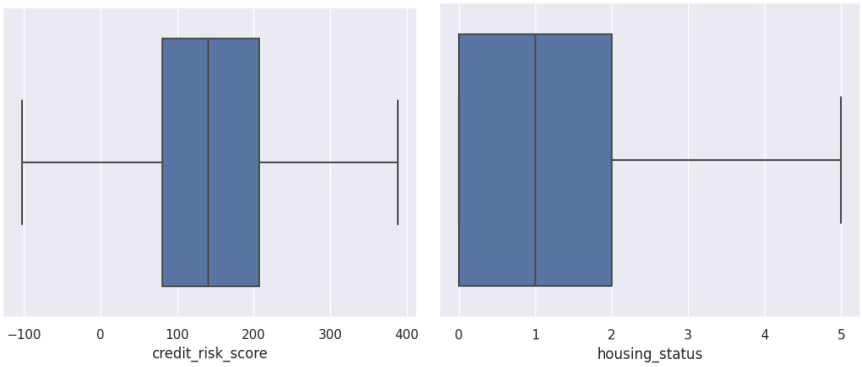


Lalu melakukan penghapusan outlier itu menggunakan metode Interquartile Range (IQR) yakni menggunakan statistika kuartil batas atas dan batas bawah. Hasil didapat fitur credit_risk_score dan fitur housing_status terdapat outlier. Maka dari itu bisa melakukan penghapusan outlier pada kedua fitur tersebut.

Gambar 5. Visualisasi Sebelum Penghapusan Outlier



Gambar 6. Visualisasi Setelah Penghapusan Outlier



Lalu setelah penghapusan outlier dilakukan pembagian data training dan data testing menggunakan metode train_test_split dari library scikit learn dengan pembagian 80% untuk data

training dan untuk data testing 20%. Kemudian dilakukan normalisasi atau standarisasi menggunakan StandardScaler. Standardisasi adalah teknik penskalaan data numerik yang mengubah setiap fitur secara independen agar memiliki rata-rata nol dan standar deviasi satu. Proses ini, yang dilakukan setelah ekstraksi fitur menggunakan StandardScaler, dicapai dengan mengurangi setiap nilai dengan rata-rata fiturnya lalu membaginya dengan standar deviasi (Tekieh, 2021). Lalu, melakukan training pada model menggunakan metode Random Forest, karena metode ini paling sering banyak digunakan dan cukup baik dalam menangani klasifikasi sekaligus mengurangi terjadinya overfitting model.

Perlu diperhatikan yang dilakukan training normalisasi hanya data training saja (data yang sudah di split), hal ini dilakukan supaya mencegah terjadinya data leakage (kebocoran data training pada saat proses pelatihan model).

Gambar 7. Penggunaan Normalisasi Supaya Terhindari dari Data Leakage

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
X_train_scaled
```

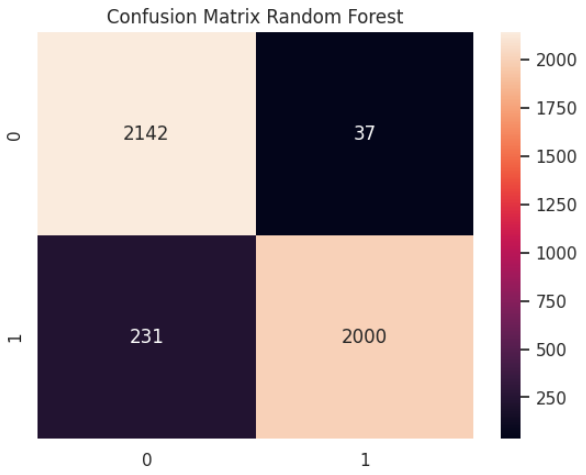
Pada tahap keempat melakukan evaluasi pada model yang sudah terlatih metrik yang diukur seperti akurasi, precision, recall, dan f1-score. Hasil akhir didapat evaluasi training akurasi Random Forest didapat 0.99 dan evaluasi validasi akurasi didapat 0.93 secara garis besar hasilnya cukup baik dan sedikit mengalami overfitting karena terdapat gap yang sedikit antara akurasi training dan akurasi validasi

Gambar 8. Hasil Evaluasi Metrik Metode Random Forest

	precision	recall	f1-score	support
0	0.90	0.98	0.94	2179
1	0.98	0.90	0.94	2231
accuracy			0.94	4410
macro avg	0.94	0.94	0.94	4410
weighted avg	0.94	0.94	0.94	4410

Pada hasil confusion matrix, model mampu melakukan prediksi dengan benar sesuai dengan kenyataan di dunia nyata. Hal ini jumlah True Positive (TP) dan True Negative (TN) cukup besar sekitar 90% dari dataset yang berhasil di prediksi oleh model. TN sebanyak 2142 data danTP 2000 data.

Gambar 9. Hasil Confusion Matrix Metode Random Forest



Setelah dilakukan training pada beberapa model klasifikasi seperti Random Forest, Logistc Regression, SVC, Naive Bayes, KNeighbors, dan Decision Tree. Hasil akhir menunjukan metode Logstic Regression lebih unggul diantara yang lainnya dengan akurasi 0.94. Itu berarti dalam dataset yang digunakan ini lebih cocok metode Logistic Regression daripada metode Random Forest. Tetapi pada penelitian ini peniliti akan lebih focus terhadap metode Random Forest.

Gambar 10. Hasil Training dari Beberapa Model

	accuracy	precision	recall	f1_score
LogisticRegression()	0.941058	0.946151	0.941058	0.940226
SVC()	0.941058	0.946151	0.941058	0.940226
GaussianNB()	0.940828	0.945875	0.940828	0.940000
RandomForestClassifier()	0.938423	0.940917	0.938423	0.937850
KNeighborsClassifier()	0.924467	0.926101	0.924467	0.923986
DecisionTreeClassifier()	0.904896	0.904967	0.904896	0.904923

Langkah terakhir supaya model lebih optimal dilakukan optimasi hyperparameter tuning menggunakan gridsearch CV dengan tujuan untuk menyesuaikan atau mengatur parameter yang paling baik (*optimized*) dalam hal ini karena peneliti menggunakan metode Random Forest maka

parameter yang diatur seperti n_estimators, max_depth, dan bootstrap. Dalam kasus ini metrik yang digunakan adalah mean_test_score bawaan dari gridsearch CV itu sendiri, semakin mendekati nol maka parameter itulah yang dipilih. Dalam hal ini pengaturan parameter terbaik ada pada pengaturan:

Tabel 1. Parameter Terbaik Hasil dari Gridsearch CV

Nama Parameter	Tipe Data	Nilai
bootstrap	boolean	False
max_depth	integer	4
n_estimators	integer	6

Gambar 9. Hasil Gridsearch CV Metode Random Forest

```
0.2586376307428623 {'max_depth': 2, 'n_estimators': 10}
0.25287462947336675 {'max_depth': 2, 'n_estimators': 20}
0.24559613253766907 {'max_depth': 2, 'n_estimators': 30}
0.24420726485634447 {'max_depth': 2, 'n_estimators': 40}
0.24467109673647863 {'max_depth': 2, 'n_estimators': 50}
0.24651769724847208 {'max_depth': 4, 'n_estimators': 10}
0.24420726485634447 {'max_depth': 4, 'n_estimators': 20}
0.24374255032513767 {'max_depth': 4, 'n_estimators': 30}
0.24374255032513767 {'max_depth': 4, 'n_estimators': 40}
0.24467109673647863 {'max_depth': 4, 'n_estimators': 50}
0.24880667576405963 {'max_depth': 6, 'n_estimators': 10}
0.24651769724847208 {'max_depth': 6, 'n_estimators': 20}
0.24327694808466288 {'max_depth': 6, 'n_estimators': 30}
0.24467109673647863 {'max_depth': 6, 'n_estimators': 40}
0.24513405097588098 {'max_depth': 6, 'n_estimators': 50}
0.2501700102202609 {'max_depth': 8, 'n_estimators': 10}
0.24971639242146468 {'max_depth': 8, 'n_estimators': 20}
0.24651769724847208 {'max_depth': 8, 'n_estimators': 30}
0.24467109673647863 {'max_depth': 8, 'n_estimators': 40}
0.24513405097588098 {'max_depth': 8, 'n_estimators': 50}
0.24605734633835835 {'max_depth': 10, 'n_estimators': 10}
0.24605734633835835 {'max_depth': 10, 'n_estimators': 20}
0.24697719009314 17 {'max_depth': 10, 'n_estimators': 30}
0.24835056781724377 {'max_depth': 10, 'n_estimators': 40}
0.24880667576405965 {'max_depth': 10, 'n_estimators': 50}
0.3011693009684171 {'bootstrap': False, 'max_depth': 2, 'n_estimators': 2}
0.2664114424895117 {'bootstrap': False, 'max_depth': 2, 'n_estimators': 4}
0.24651769724847208 {'bootstrap': False, 'max_depth': 2, 'n_estimators': 6}
0.25510657592340713 {'bootstrap': False, 'max_depth': 2, 'n_estimators': 8}
0.2501700102202609 {'bootstrap': False, 'max_depth': 4, 'n_estimators': 2}
0.24513405097588098 {'bootstrap': False, 'max_depth': 4, 'n_estimators': 4}
0.24281045302822785 {'bootstrap': False, 'max_depth': 4, 'n_estimators': 6}
0.24513405097588098 {'bootstrap': False, 'max_depth': 4, 'n_estimators': 8}
0.253769763704271 {'bootstrap': False, 'max_depth': 6, 'n_estimators': 2}
0.24697719009314 17 {'bootstrap': False, 'max_depth': 6, 'n_estimators': 4}
0.24513405097588098 {'bootstrap': False, 'max_depth': 6, 'n_estimators': 6}
0.24559613253766907 {'bootstrap': False, 'max_depth': 6, 'n_estimators': 8}
```

Peneliti menambahkan metode deep learning untuk melakukan perbanding kualitas model. Hasil menunjukan metode deep learning lebih unggul daripada metode machine learning klasik, karena memang secara umum metode deep learning sering digunakan dalam kasus yang kompleks

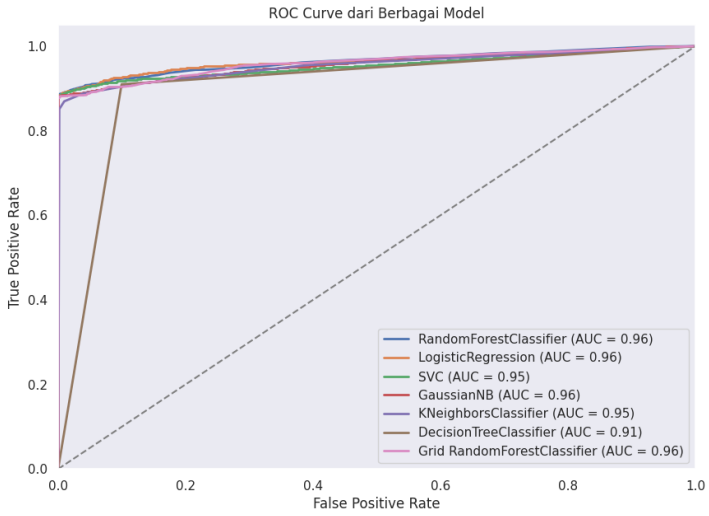
mulai dari data yang struktur hingga tidak terstruktur (teks, gambar, dan audio). Sehingga hasil evaluasi secara keseluruhan menjadi seperti berikut:

Gambar 11. Hasil Training Beberapa Model Beserta Gridsearch CV dan Metode Deep learning

	accuracy
DeepLearning	0.941270
LogisticRegression	0.941058
SVC	0.941058
GaussianNB	0.940828
Grid RandomForestClassifier	0.940136
RandomForestClassifier	0.938423
KNeighborsClassifier	0.924467
DecisionTreeClassifier	0.904896

Dalam tahap ini dilakukan juga penggunaan evaluasi pada ROC Curve. Kurva ROC memplot Tingkat True Positive (TP) pada sumbu vertikal melawan Tingkat False Positive (FP) pada sumbu horizontal. Tingkat TP adalah rasio prediksi positif yang benar terhadap total data positif, sedangkan Tingkat FP adalah rasio prediksi positif yang salah terhadap total data negatif. Estimasi dianggap lebih akurat ketika Tingkat TP mendekati 1 dan Tingkat FP mendekati 0 (Heo *et al.*, 2024). Pada metode Random Forest dan Logistic Regression memiliki peringkat yang unggul diantara yang lainnya, ini berarti pada kasus dataset ini kedua metode ini sangat cocok untuk digunakan sekaligus cukup akurat dalam melakukan prediksi pada model.

Gambar 12. Hasil ROC Curve dari Berbagai Model



V. KESIMPULAN DAN REKOMENDASI KEBIJAKAN

Berdasarkan analisis dan evaluasi model yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

- Dalam dataset yang digunakan sebanyak 1 juta baris data memiliki jumlah data antar kelas positif dan negatif yang berbeda pada variabel target, kelas negatif lebih banyak daripada kelas positif perbedaan ini memiliki rasio 90:10 untuk itu dilakukan teknik slicing pada kelas negatif supaya jumlah data seimbang. Peneliti tidak melakukan teknik SMOTE dikarenakan rasio yang memiliki perbedaan yang cukup jauh. Karena konsep dari teknik SMOTE ini melakukan replica pada kelas tertentu yang berati melakukan duplikasi kembali terhadap data dan ini akan rentan terhadap model menjadi overfitting.
- Feature extraction yang digunakan adalah teknik SelectKBest dengan 5 fitur yang terbaik diantaranya fitur month, velocity_4w, velocity_24h, housing_status dan credit_risk_score. Dan terdapat outlier pada credit_risk_score dan housing status sehingga dilakukan penghapusan outlier pada fitur tersebut.
- Metode deep learning lebih unggul dari pada machine learning klasik hal ini karena metode deep learning umunnya sering digunakan pada hal yang kompleks mulai dari data terstruktur dan data yang tidak terstruktur.

Untuk rekomendasi dari peneliti disarankan untuk melakukan penyeimbangan jumlah data yang seragam pada masing-masing kelas variabel target yakni fitur fraud_bool untuk mendapatkan hasil yang optimal dan mengurangi terjadinya overfitting model. Karena terkait cakupan dan batasan dalam penilitian, disarankan untuk melakukan hyperparameter tuning kembali pada metode deep learning supaya hasilnya lebih optimal, hyperparameter tuning tersebut bisa dilakukan pada jumlah layer, neuron, dan pengaturan fungsi aktifasi seperti ReLu dan sebagainya.

DAFTAR PUTAKA

Aiswarya A S and Rajeev, H. (2024) 'Youtube Comment Sentimental Analysis', *Indian Journal of Data Mining*, 4(1), pp. 5–8. Available at: https://doi.org/10.54105/ijdm.A1633.04010524.

Alinda Rahmi, N. and Defit, S. (2024) *The Use of Hyperparameter Tuning in Model Classification: A Scientific Work Area Identification*. Available at: www.joiv.org/index.php/joiv.

ASIM, N.M. (2022) *AN EFFICIENT AUTOMATED MACHINE LEARNING FRAMEWORK FOR GENOMICS AND PROTEOMICS SEQUENCE ANALYSIS From DNA to RNA and Protein Sequence Analysis*.

Banerjee, [1]sudipta (2024) *Basic Concepts of Machine Learning*. Bhopal, Madhya Pradesh, India.

Beshay, N. (2022) *A DEEP LEARNING BASED MULTILINGUAL HATE SPEECH DETECTION FOR RESOURCE SCARCE LANGUAGES*.

Bui, Q.A.T. *et al.* (2025) 'Prediction of Shear Bond Strength of Asphalt Concrete Pavement Using Machine Learning Models and Grid Search Optimization Technique', *CMES - Computer Modeling in Engineering and Sciences*, 142(1), pp. 691–712. Available at: https://doi.org/10.32604/cmes.2024.054766.

Eldo, H. *et al.* (2024) 'Penggunaan Algoritma Support Vector Machine (SVM) Untuk Deteksi Penipuan pada Transaksi Online', *Jurnal Minfo Polgan*, 13(2), pp. 1627–1632. Available at: https://doi.org/10.33395/jmp.v13i2.14186.

Es-Sabery, F. *et al.* (2022) ‘Evaluation of different extractors of features at the level of sentiment analysis’, *Infocommunications Journal*, 14(2), pp. 85–96. Available at: https://doi.org/10.36244/ICJ.2022.2.9.

Heo, W. *et al.* (2024) ‘Identifying Hidden Factors Associated with Household Emergency Fund Holdings: A Machine Learning Application’, *Mathematics*, 12(2). Available at: https://doi.org/10.3390/math12020182.

Hnatiienko, H. (2023) ‘Information Technology and Implementation (Satellite): Conference Proceedings’, pp. 2–343.

Homepage, J., Unogwu, O.J. and Filali, Y. (2023) ‘Wasit Journal of Computer and Mathematics Science Fraud Detection and Identification in Credit Card Based on Machine Learning Techniques’, *WJCMS*, pp. 16–22. Available at: https://doi.org/10.31185/wjcm.185.

Huseynov, F. and Ozdenizci Kose, B. (2024) ‘Using machine learning algorithms to predict individuals’ tendency to be victim of social engineering attacks’, *Information Development*, 40(2), pp. 298–318. Available at: https://doi.org/10.1177/02666669221116336.

Japit, S. *et al.* (2024) ‘Deteksi Anomali Transaksi E-Commerce Menggunakan Support Vector Machine Berbasis Data Mining’, *Jurnal Minfo Polgan*, 13(2), pp. 1976–1980. Available at: https://doi.org/10.33395/jmp.v13i2.14325.

Lee, K. *et al.* (2024) ‘Binning as a Pretext Task: Improving Self-Supervised Learning in Tabular Domains’, pp. 1–8. Available at: http://arxiv.org/abs/2405.07414.

Liu, T. *et al.* (2024) ‘Spam Detection and Classification Based on DistilBERT Deep Learning Algorithm’, *Applied Science & Engineering Journal for Advanced Research Peer Reviewed and Refereed Journal ISSN*, pp. 6–10. Available at: https://doi.org/10.5281/zenodo.11180575.

Malik Ibrahim, M. (2025) ‘ANALISIS KINERJA MODEL MACHINE LEARNING UNTUK MENDETEKSI TRANSAKSI FRAUD PADA SISTEM PEMBAYARAN ONLINE’, *JINU)*, 2(3), pp. 35–49. Available at: https://doi.org/10.61722/jinu.v2i3.4276.

Maulidah, S.S. (2023) *ANALISIS SENTIMEN TERHADAP BRAND REPUTATION SUPER APPS GOJEK DAN GRAB DI INDONESIA MENGGUNAKAN ALGORITMA MACHINE LEARNING*. Jakarta.

Mramba, L.K. *et al.* (2023) 'Detecting Potential Outliers in Longitudinal Data with Time-Dependent Covariates'. Available at: https://doi.org/10.20944/preprints202305.0390.v1.

Patil, A.N. *et al.* (2021) *A Survey Paper On Data Preprocessing In Digital Card Fraud Detection*, *International Journal of All Research Education and Scientific Methods (IJARESM)*.

Pratama Adiwijaya, A. and Sukma Maulana, W. (2023) 'ANALISIS PEMBUATAN SISTEM ANTIFRAUD PADA STARTUP FINTECH, KHUSUSNYA PEER-TO-PEER LENDING', *JUIT*, 2(3), pp. 69–76.

Rangkuti, N.R. *et al.* (2024) 'PERAN BANK INDONESIA DALAM PENGEMBANGAN TRANSAKSI UANG DIGITAL', *Jurnal Akademi kEkonomi dan Manajemen*, 1(4), pp. 9–17. Available at: https://doi.org/10.61722/jaem.v1i4.3108.

Reyhand Ardhitha, Revifal Anugerah and Tata Sutabri (2025) 'Analisis Penerapan Machine Learning dan Algoritma Anomali untuk Deteksi Penipuan pada Transaksi Digital', *Repeater : Publikasi Teknik Informatika dan Jaringan*, 3(1), pp. 80–90. Available at: https://doi.org/10.62951/repeater.v3i1.345.

Subagio, A. and Utama, D.N. (2025) 'Fraud credit card transaction detection using hybrid multilayer perceptron-random forest method', *Edelweiss Applied Science and Technology*, 9(3), pp. 2482–2494. Available at: https://doi.org/10.55214/25768484.v9i3.5823.

Taye, M.M. (2023) 'Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions', *Computers*. MDPI, pp. 1–26. Available at: https://doi.org/10.3390/computers12050091.

Tekieh, R. (2021) *Understanding How Developers Reuse Stack Overflow Code in Their GitHub Projects*.