Eksplorasi Teknik *Pre-Processing* Berbasis *eXtreme Gradient Boosting* (XGBoost) pada Klasifikasi Kredit Default

Abdul Khahar[1], Yuni Yamasari[2]

[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

[1]abdul.20026@mhs.unesa.ac.id

[2]yuniyamasari@unesa.ac.id

***Abstrak*— Perilaku konsumtif dari masyarakat berdampak pada peningkatan aktivitas kredit. Tentu saja, peningkatan ini dapat membawa risiko yang semakin besar terhadap risiko default atau gagal bayar. Permasalahan ini krusial dan perlu diselesaikan. Di sisi lain, disiplin ilmu *machine learning* dapat menangani berbagai permasalahan kehidupan, termasuk di dalamnya masalah perekonomian. Untuk itu, penelitian ini memfokuskan pada pemecahan masalah kredit default dengan *eXtreme Gradient Boosting* (XGBoost). Lebih dari itu, penelitian meningkatkan kinerja dari model klasifikasi kredit default dengan eksplorasi penggunaan teknik-teknik pra-pemrosesan. Adapun *dataset* yang digunakan adalah *"Credit Card Default Risk"* yang memiliki sampel sebanyak 45.528 dengan 18 fitur sebagai data latih dan sebanyak 11.383 sampel sebagai data uji. Model yang dibangun menggunakan teknik terbaik yaitu *Random UnderSampler, feature selection* berdasarkan angka *correlation* dengan *treshold* 0.001, dan rasio *test-size* dari *train-test-split* sebagai evaluasi pada ukuran 0.2, Hasil evaluasi kinerja saat pelatihan model XGBoost menunjukkan skor AUC 0.9730, *F1-Score* 0.7807, *Recall* 0.9946, *Precision* 0.6426, dan *Accuracy* 0.9550. Setelah dilakukan *re-training* diperoleh skor AUC 0.9910, *F1-Score* 0.9806, *Recall* 0.9940, *Precision* 0.9676, dan *Accuracy* 0.9896. Hasil penelitian ini mengindikasikan bahwa pemilihan teknik pra-pemrosesan yang tepat pada XGBoost dan distribusi *dataset* berpengaruh terhadap kinerja model klasifikasi.**

***Kata Kunci*— *Pre-Processing*, XGBoost, Klasifikasi, Kredit Default**

I. PENDAHULUAN

Kebutuhan masyarakat dapat meningkat apabila perekonomian suatu negara semakin berkembang. Namun secara materi, masyarakat masih banyak yang tergolong rendah kondisi perekonomiannya. Untuk memenuhi kebutuhan masing-masing, aktivitas kredit semakin sering dilakukan oleh masyarakat. Tujuannya pun berbeda-beda dengan sifat konsumtif yang beragam. [1].

Dengan persepsi yang sama, [2] menyebutkan bahwa ekosistem digital saat ini berpotensi membuat masyarakat berperilaku lebih konsumtif tanpa kesadaran atas kondisi ekonomi mereka. Sehingga berpotensi terjadinya masalah perekonomian yang lebih besar secara jangka panjang. Sebagaimana menurut [3], perkembangan ekonomi tentu membawa risiko yang semakin besar, khususnya risiko kredit. Risiko kredit tergolong sebagai masalah yang dapat dikatakan berdampak besar dalam bidang keuangan. Oleh karena itu, perlu adanya perhatian khusus pada risiko kredit tersebut. Biaya penyisihan dalam laporan laba atau rugi dapat ditimbulkan karena adanya nominal yang gagal terbayar atau tidak tertagih dimana kemudian menjadi kredit macet.

Risiko kredit timbul karena adanya debitur yang tidak mampu membayar kembali atau memenuhi tagihan hutangnya sesuai dengan yang disepakati atau adanya penurunan ekonomi yang dialami oleh debitur sehingga risiko kegagalan bayar dapat meningkat. Atas hal itu perlu dilakukan pengelolaan yang baik dalam hal risiko kredit pada debitur oleh pihak lembaga keuangan terkait agar tidak menimbulkan dampak buruk pada kondisi lembaga keuangan tersebut. [3].

Menurut penjelasan oleh [4], Home Credit sebagai mitra finansial penyedia layanan pinjaman berupaya memperluas inklusi keuangan bagi masyarakat dalam mengatasi masalah kebutuhan peminjaman dengan memberikan pengalaman pinjaman yang aman. Namun atas adanya risiko kredit yang mungkin terjadi, untuk menghindari potensi dampak kerugian pada pihak lembaga Home Credit diperlukan prediksi risiko default pada pengajuan kredit oleh para calon debitur.

Metode *machine learning* dapat digunakan sebagai solusi permasalahan yang telah disebutkan. Seperti yang dijelaskan oleh [5], bahwa akurasi pendekatan menggunakan *machine learning* dalam menangani masalah perekonomian lebih unggul dibandingkan dengan menggunakan metode statistik secara konvensional.

Pada penelitian pada studi yang relevan oleh [1], terkait prediksi risiko kredit, digunakan beberapa metode yaitu *Logistic Regression*, *Random Forest*, *Gaussian Naïvee Bayes*, dan *Decision Tree*. Dimana dari model-model tersebut, *Random Forest* menunjukkan hasil skor evaluasi yang paling baik dengan metrik evaluasi ROC AUC *train score* sebesar 1,00 dan ROC AUC *test score* sebesar 0,72. Namun dikarenakan kelima algoritma yang digunakan tersebut merupakan algoritma klasifikasi dasar, peneliti menyarankan untuk menggunakan *eXtreme Gradient Boosting* (XGBoost) dalam penelitian selanjutnya sebagai perbandingan dengan algoritma *Random Forest*.

Selain penelitian tersebut, terdapat penelitian oleh [6] yang membandingkan antara algoritma *Support Vector Machine* (SVM), *Random Forest*, dan *eXtreme Gradient Boosting* (XGBoost). Dimana mendapatkan skor metrik tertinggi pada model XGBoost dengan nilai akurasi sebesar 82%, *recall* 70%, dan *precision* 92%. Pada penelitian ini digunakan pula teknik *sampling* SMOTE sebelum membagi dataset, karena data yang digunakan bersifat *imbalance*.

Selain teknik optimalisasi dalam hal akurasi, berdasarkan *dataset* yang berukuran besar dibutuhkan proses klasifikasi yang terbilang cukup lama. Maka diperlukan optimalisasi dalam hal kecepatan waktu klasifikasi. Dibuktikan pada penelitian oleh [7], dalam pengujian dengan perbedaan persentase pembagian data *training* dan *testing*. Hasil menunjukkan waktu proses pendeteksian dapat dipersingkat menggunakan teknik *feature selection* dalam penerapan metode *Random Forest Classifier*, walaupun terjadi 1% angka penurunan pada skor akurasi.

Sebagaimana dengan beberapa literatur yang ada, pada penelitian ini akan dilakukan penggunaan algoritma *eXtreme Gradient Boosting* (XGBoost) untuk mengklasifikasi *dataset* "*Credit Card Default Risk*" yang bersumber dari *open-source platform*, Kaggle, dengan beberapa teknik pra-pemrosesan *sampling*, dan *feature selection*. Selanjutnya akan dilakukan perbandingan penggunaan teknik yang telah disebutkan sehingga didapatkan skor metrik klasifikasi yang stabil dan optimal. Dengan ini diharapkan dapat terbentuk suatu model yang optimal serta dapat membantu dalam pengembangan sistem klasifikasi kedepannya terutama untuk lembaga keuangan dalam klasifikasi status default kredit pada calon debitur agar lembaga-lembaga keuangan secara luas dapat mengurangi adanya risiko timbulnya kerugian yang berdampak buruk secara berkepanjangan.

II. METODOLOGI PENELITIAN

A. Identifikasi Masalah

Permasalahan pada penelitian ini adalah adanya risiko default pada calon debitur. Lembaga keuangan layanan pemimjaman perlu mencegah adanya kerugian yang dapat ditimbulkan dari kemungkinan gagal bayar kembali pinjaman oleh calon debitur. Maka dari itu diperlukan klasifikasi kemampuan bayar kembali kredit agar pihak kreditur dapat terhindar dari kerugian nantinya. Namun saat ini penelitian yang membahas topik tersebut masih terbatas pada beberapa metode saja dan belum optimal untuk algoritma klasifikasi dalam pemodelannya. Sehingga penelitian ini ingin memberikan solusi dengan mengklasifikasi status default kredit menggunakan metode XGBoost dan dilakukan percobaan eksplorasi dengan beberapa teknik pendukung dimana menghasilkan performa terbaik dari seluruh percobaan.

B. Studi Literatur

Studi literatur ini mempelajari tentang *machine learning,* XGBoost, *sampling,* dan *feature selection*. Beberapa teori didapat dari jurnal-jurnal yang menjadi referensi dari penulisan penelitian ini sehingga dapat bermanfaat dalam menjadi referensi atau rujukan untuk penyelesaian masalah pada penelitian yang diusulkan.

C. Analisis Kebutuhan

Penelitian ini menggunakan algoritma XGBoost yang dijalankan dengan bahasa pemrograman Python, dimana nantinya digunakan juga untuk menguji *dataset*. Oleh karena itu, dibutuhkan perangkat yang dapat mendukung penelitian ini agar dapat berjalan sesuai tujuan. Beberapa kebutuhan perangkat seperti berikut:

- *Browser* Google Chrome versi 123.0.6312.122 (*Official Build*) (64-bit).
- IDE Jupyter Notebook *via* Microsoft Visual Studio Code versi 1.88.1 dan Google Colab dengan *update notes* 2024/02/21.
- Python 3.8.15.

D. Dataset

Dataset yang digunakan dalam penelitian ini adalah *dataset* "*Credit Card Default Risk"* unggahan sumber terbuka Kaggle, dari *AmExpert CodeLab 2021* oleh *American Express*. [8]. *Dataset* ini memiliki dua *file* utama yaitu '*train.*csv' dengan sampel sebanyak 45.528 dengan 18 fitur dan 1 label '*credit card default*' sebagai kelas target yang berisi klasifikasi status bayar kembali pinjaman oleh debitur. Adapun perincian fitur dan target pada *train* data terdapat pada Tabel I.

TABEL I
PERINCIAN FITUR PADA *DATASET*

No.	Nama Fitur	Tipe Data
1	*customer_id*	*object*
2	*name*	*object*
3	*age*	*int64*
4	*gender*	*object*
5	*owns_car*	*object*
6	*owns_house*	*object*
7	*no_of_children*	*float64*
8	*net_yearly_income*	*float64*
9	*no_of_days_employed*	*float64*
10	*occupation_type*	*object*
11	*total_family_members*	*float64*
12	*migrant_worker*	*float64*
13	*yearly_debt_payments*	*float64*
14	*credit_limit*	*float64*
15	*credit_limit_used(%)*	*int64*
16	*credit_score*	*float64*
17	*prev_defaults*	*int64*
18	*default_in_last_6months*	*int64*
19	*credit_card_default*	*int64*

Pada pelabelan '*credit card default'* nilai 0 menunjukkan bahwa debitur berstatus kredit lancar yang berarti mampu membayar kembali pinjaman. Sedangkan nilai 1 menunjukkan informasi sebaliknya, debitur berstatus kredit macet atau mengalami kendala dalam membayar kembali pinjaman. Kemudian *file* '*test*.csv' dimana merupakan kumpulan data yang akan diklasifikasi dengan model yang telah dirancang nantinya. Data ini memiliki sampel sebanyak 11.383 dengan 18 fitur.

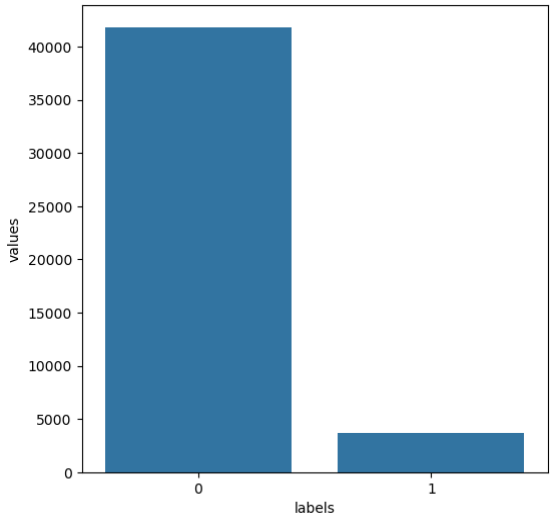
E. Exploratory Data Analysis (EDA)

Tahap setelah *input data* adalah *Exploratory Data Analysis* (EDA). Dalam setiap komposisi data dilakukan analisis. Berdasarkan hasil EDA, dari 45.528 sampel pada '*train.csv*', kolom target '*credit_card_default*', didapatkan terdapat sebanyak 41.831 sampel (91,9%) bernilai 0 (kredit lancar), dan 3.697 sampel (8,1%) sisanya bernilai 1 (kredit macet).

Sehingga dapat dikatakan *dataset* bersifat *imbalance* atau tidak seimbang. Maka dari itu, dibutuhkan teknik-teknik tertentu untuk mengatasi hal tersebut sehingga diharapkan model tidak menjadi *overfitting* atau bias terhadap kelas yang berifat mayoritas.

Dengan adanya sampel yang besar pada *dataset*, maka diperlukan model maupun teknik tertentu agar eksekusi model tidak memakan waktu yang terlalu lama, misalnya dengan teknik pemodelan *boosting* mampu atau dapat mengatasi waktu eksekusi pada *dataset* yang berukuran besar. Sehingga selama penelitian dilakukan dengan menggunakan algoritma pemodelan yang berbasis pada *eXtreme Gradient Boosting* (XGBoost).

Tahap setelah *input data* adalah *Exploratory Data Analysis* (EDA). Dalam setiap komposisi data dilakukan analisis. Berdasarkan hasil EDA, dari 45.528 sampel pada '*train.csv*', kolom target '*credit card default*', didapatkan terdapat sebanyak 41.831 sampel (91,9%) bernilai 0 (kredit lancar), dan 3.697 sampel (8,1%) sisanya bernilai 1 (kredit macet). Sehingga dapat dikatakan *dataset* bersifat *imbalance* atau tidak seimbang seperti pada Gbr. 1. Maka dari itu, dibutuhkan teknik-teknik tertentu untuk mengatasi hal tersebut sehingga diharapkan model tidak menjadi *overfitting* atau bisa terhadap kelas yang berifat mayoritas.



Gbr. 1 Distribusi Sampel pada Kelas Target

F. Data Preprocessing

Langkah selanjutnya setelah EDA yaitu pra-pemrosesan data. Berikut perincian dari tahap *data pre-processing*:

1) Drop unnecessary features: Beberapa fitur kurang memiliki manfaat terhadap perancangan maupun pelatihan model. Misalnya fitur *'customer_id'* dan *'name'*.

2) Missing values handling: Dari fitur yang ada, terdapat missing values dengan persentase yang lumayan besar. Kemudian dilakukan imputasi pada fitur yang memiliki kekosongan dengan nilai *mean* atau *median* pada fitur yang bersifat numerik dan imputasi dengan nilai *mode* pada fitur yang bersifat kategorik.

3) Outliers handling: Adapun pereduksian *outlier* apabila terdapat nilai pada fitur yang dianggap sebagai anomali atau jauh dari *range* nilai normal pada umumnya maupun berdasarkan statistik pada persebaran data masing-masing fitur.

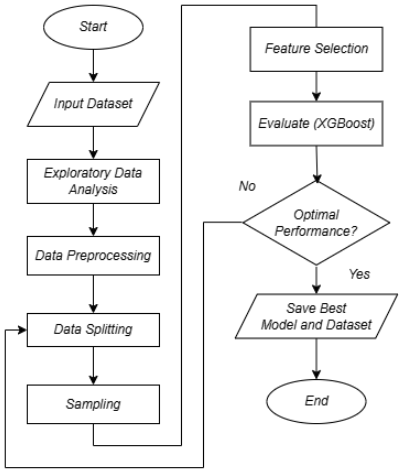
4) Categorical features encoding: Mengingat adanya fitur kategorik maka perlu dilakukan *encoding* dalam *dataset*. Teknik ini dapat dilakukan menggunakan library Python yaitu *label-encoder* ataupun *one-hot-encoding*.

5) Features generating: Setelah pengecekan skor korelasi fitur terhadap label, skor terbagi cukup rata dan dapat dikatakan beberapa fitur memiliki korelasi yang agak lemah, sehingga dapat dilakukan *feature-engineering* dengan membagi antar fitur sebagai rasio dari fitur-fitur tertentu dan mendapatkan korelasi yang lebih kuat terhadap kelas target. Fitur yang dihasilkan terdapat pada Tabel II.

TABEL II
HASIL *FEATURES GENERATING*

No.	Nama Fitur
1	*in_hand_balance*
2	*in_profit*
3	*total_income_received*
4	*employment_years*
5	*log_net_yearly_income*
6	*log_yearly_debt_payments*
7	*log_credit_limit*

G. Perancangan Model



Gbr. 2 *Flowchart* Perancangan Model Klasifikasi Kredit Default Berbasis pada XGBoost

Sebagaimana pada Gbr.2, penelitian ini akan dilakukan dengan menggunakan metode atau algoritma *eXtreme Gradient Boosting* (XGBoost) dalam pemodelan klasifikasinya. Metode ini akan dieksplorasi dalam penggunaan beberapa teknik pra-pemrosesan. Hasil eksplorasi akan dilakukan perbandingan pada hasilnya berdasarkan metrik yang digunakan sebagai evaluasi.

Setelah *dataset* siap. *Dataset* akan dibagi menjadi data latih dan data uji dengan persentase tertentu. Data latih sendiri merupakan data *historical* yang berisi sisi positif dari

informasi-informasi yang ada dan digunakan untuk memperoleh kelas yang sesuai pada data uji nantinya. Sedangkan data uji merupakan informasi baru yang akan diklasifikasikan oleh pemodelan mendatang. [9]. Umumnya dengan *dataset* yang berdimensi besar dilakukan pembagian data dengan perbandingan 80:20, seperti pada penelitian oleh [10]. *Dataset* dibagi secara linier dan dilakukan dalam satu kali pemanggilan atau eksekusi, sehingga tidak diperlukan waktu yang relatif lama dalam tahap pembagian *dataset* berukuran besar.

Dengan *dataset* yang bersifat tidak seimbang atau *imbalance*, diperlukan teknik *sampling* untuk penanganannya. Dengan *dataset* yang berukuran besar, lebih banyak sampel yang terhitung penting sebagai data latih, data yang hilang akan memengaruhi performa model yang akan dibangun. Oleh karena itu, dapat dilakukan *over-sampling* untuk menambah kelas minoritas, atau *under-sampling* untuk mengurangi kelas mayoritas. Metode yang dapat dicoba adalah *Random Over Sampling* ataupun *Random Under Sampling.* Dimana metode tersebut merupakan metode *sampling* yang paling sederhana dan umum untuk dilakukan, algoritma *non-heuristic* yang secara acak mencoba menyeimbangkan data mayoritas ke minoritas, atau sebaliknya. [11].

Untuk mengoptimalkan waktu eksekusi pemodelan, dapat dilakukan reduksi dimensi atau fitur. Dalam hal ini akan dipilih fitur penting dan relevan dari data input dan menghapus fitur yang tidak berguna serta tidak relevan. Fase algoritma seleksi fitur dibagi menjadi dua fase seperti (i) *subset generation*; dan (ii) *subset evaluation*. Dalam *subset generation*, perlu untuk menghasilkan *subset* dari input *dataset*, dan untuk *subset evaluation* yang diperlukan adalah pemeriksaan apakah *subset* yang dihasilkan optimal atau tidak. [12]. Setelah didapatkan fitur yang penting berdasarkan angka korelasi fitur terhadap target, kemudian diterapkan pada *data* yang ada dengan cara memilih fitur tertentu sebagai data yang dipakai dalam pelatihan model kedepannya.

Setelah pemilihan fitur, maka dapat dilakukan pembuatan model *eXtreme Gradient Boosting* (XGBoost). XGBoost dipilih karena telah diakui secara luas dalam kompetisi *machine learning* Kaggle atas keunggulannya dalam hal efisiensi yang tinggi dan fleksibilitas yang memadai. [13]. XGBoost sendiri lebih teratur dalam membangun struktur pohon regresi, sehingga dapat memberikan performa yang lebih baik dan dapat mengurangi kompleksitas model dalam tujuan menghindari terjadinya *overfitting*. [14]. Penggunaan algoritma *Gradient Boosting* pada XGBoost diharapkan dapat menemukan solusi optimal untuk klasifikasi, dimana konsep dasarnya adalah dengan penyesuaian parameter pembelajaran iteratif untuk mengurangi *loss function*. [9].

Setelah dilakukan pelatihan model, hasil latih akan dilakukan evaluasi pada performanya sesuai metrik evaluasi yang telah ditentukan. Evaluasi dilakukan dengan tujuan mengetahui kelayakan penggunaan suatu algoritma apabila diimplementasikan pada sistem. [15]. Metrik-metrik tersebut adalah *accuracy, recall, precision, F1-Score,* dan AUC. Skor metrik utama dalam pemilihan teknik terbaik adalah skor AUC karena data awal bersifat *imbalanced* atau tidak seimbang. Metrik-metrik tersebut dapat dijabarkan dan dijelaskan seperti berikut:

1) Accuracy merupakan metrik ukur persentase keakuratan klasifikasi secara benar yang dicapai dalam implementasi pemodelan. Penghitungan dilakukan dengan cara menjumlahkan *True Negative (TN)* dan *True Positive (TP)* kemudian dibagi dengan total sampel yang ada dalam *dataset.* [16].

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

2) Precision merupakan metrik ukur yang menunjukkan seberapa akurat identifikasi sebagai persentase klasifikasi optimis benar dari seluruh klasifikasi positif yang ada. Nilai ini dihitung dengan cara pembagian dari *True Positive (TP)* dengan jumlah seluruh klasifikasi positif yaitu *True Positive (TP)* dan *False Positive (FP)*. [16].

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

3) Recall merupakan metrik ukur penghitung proporsi *True Positive (TP)* yang secara benar dideteksi, dimana menunjukkan rasio TP terhadap kejadian yang seharusnya dideteksi sebagai positif secara keseluruhan. [16].

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

4) F1-score merupakan metrik ukur dimana menimbang antara *recall* dan *precision*. Skor ini memiliki rentang nilai antara 0 dan 1. Jika bernilai mendekati atau sama dengan 1, kinerja algoritma atau metode pemodelan dapat dikatakan mampu melakukan klasifikasi dengan sangat baik. Namun sebaliknya jika bernilai mendekati atau sama dengan 0, kinerja algoritma atau metode pemodelan melakukan klasifikasi yang dapat dikatakan buruk. [17].

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{(Precision \times Recall)} \quad (4)$$

5) Area Under The Curve (AUC) merupakan metrik ukur dalam melakukan evaluasi kinerja algoritma atau model klasifikasi. Kemampuan model untuk membedakan kelas positif dan negatif diukur dengan melakukan *plot* dari *True Positive Rate (TPR)* terhadap *False Positive Rate (FPR)*. Model klasifikasi yang dapat dikatakan sempurna menghasilkan skor *AUC* dengan nilai 1.0, sedangkan model kasifikasi yang acak memiliki skor *AUC* dengan nilai 0.5 atau mendekati. Nilai *AUC* didapat dari pengintegrasian kurva *TPR-FPR* pada rentang *FPR* secara keseluruhan dari nilai 0 hingga 1. [16].

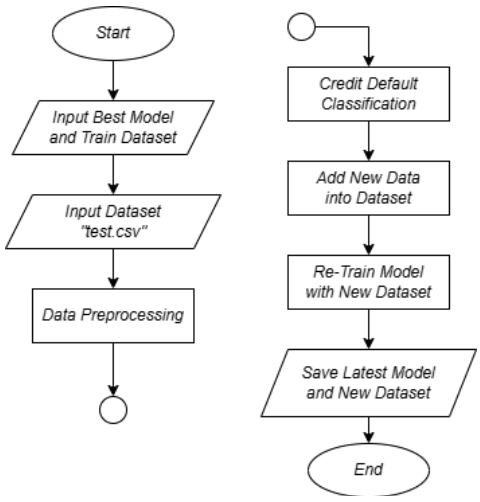
$$TPR = \frac{TP}{TP+FN} \quad (5)$$

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

$$\int_0^1 TPR(FPR), dFPR \quad (7)$$

Apabila model mencapai hasil yang optimal dalam penggunaan *sampling* dan juga *feature selection*, dapat dilakukan tahapan selanjutnya yaitu pengujian model. Model terbaik yang dihasilkan akan digunakan dalam pengujian dimana mengklasifikasi data uji yang telah tersedia di awal dan telah dilakukan teknik *preprocessing* yang sesuai.

H. Pengujian Model



Gbr. 3 *Flowchart* Pengujian pada Rancangan Model Terbaik untuk Klasifikasi Kredit Default

Pengujian dilakukan dengan cara mengklasifikasikan data uji yang terdapat sebelumnya pada *file* '*test*.csv' seperti pada Gbr. 3. Data masih dengan kondisi perlu dilakukan pra-pemrosesan dimana sesuai berdasarkan pada proses *training* atau perancangan model klasifikasi sebelumnya. Jika dari klasifikasi tersebut mendapatkan hasil klasifikasi yang akurat, maka *data input* serta hasil klasifikasi dimasukkan ke dalam *dataset* dan menjadi *dataset* baru. Lalu dilakukan latih model kembali dengan *dataset* baru tersebut untuk mendapatkan model baru. Model dan *dataset* baru disimpan ke sistem lokal dengan format *file* '.csv' pada *dataset*, dan '.sav' untuk model.

III. Hasil dan Pembahasan

A. Skenario Teknik Pre-Processing

Pada percobaan perancangan model klasifikasi, penggunaan beberapa teknik *sampling* dan *feature selection* akan dibandingkan menggunakan rasio *train-test split* sebagai hasil pemisahan data awal sebagai evaluasi. Perincian skenario terdapat pada Tabel III.

Tabel III
Skenario Perbandingan Teknik *Sampling*

Scenario	*Sampling*	*Feature Selection*	*Test Size*
a1	-	-	0.1
a2	-	-	0.2
a3	-	-	0.3
a4	-	-	0.4
a5	-	*Correlation*	0.1
a6	-	*Correlation*	0.2
a7	-	*Correlation*	0.3
a8	-	*Correlation*	0.4
a9	-	*Information Gain*	0.1
a10	-	*Information Gain*	0.2
a11	-	*Information Gain*	0.3
a12	-	*Information Gain*	0.4
b1	*RandomOverSampler*	-	0.1
b2	*RandomOverSampler*	-	0.2
b3	*RandomOverSampler*	-	0.3
b4	*RandomOverSampler*	-	0.4
b5	*RandomOverSampler*	*Correlation*	0.1
b6	*RandomOverSampler*	*Correlation*	0.2
b7	*RandomOverSampler*	*Correlation*	0.3
b8	*RandomOverSampler*	*Correlation*	0.4
b9	*RandomOverSampler*	*Information Gain*	0.1
b10	*RandomOverSampler*	*Information Gain*	0.2
b11	*RandomOverSampler*	*Information Gain*	0.3
b12	*RandomOverSampler*	*Information Gain*	0.4
c1	*SMOTE*	-	0.1
c2	*SMOTE*	-	0.2
c3	*SMOTE*	-	0.3
c4	*SMOTE*	-	0.4
c5	*SMOTE*	*Correlation*	0.1
c6	*SMOTE*	*Correlation*	0.2
c7	*SMOTE*	*Correlation*	0.3
c8	*SMOTE*	*Correlation*	0.4
c9	*SMOTE*	*Information Gain*	0.1
c10	*SMOTE*	*Information Gain*	0.2
c11	*SMOTE*	*Information Gain*	0.3
c12	*SMOTE*	*Information Gain*	0.4
d1	*RandomUnderSampler*	-	0.1
d2	*RandomUnderSampler*	-	0.2
d3	*RandomUnderSampler*	-	0.3
d4	*RandomUnderSampler*	-	0.4
d5	*RandomUnderSampler*	*Correlation*	0.1
d6	*RandomUnderSampler*	*Correlation*	0.2
d7	*RandomUnderSampler*	*Correlation*	0.3
d8	*RandomUnderSampler*	*Correlation*	0.4
d9	*RandomUnderSampler*	*Information Gain*	0.1
d10	*RandomUnderSampler*	*Information Gain*	0.2
d11	*RandomUnderSampler*	*Information Gain*	0.3
d12	*RandomUnderSampler*	*Information Gain*	0.4
e1	*NearMiss*	-	0.1
e2	*NearMiss*	-	0.2
e3	*NearMiss*	-	0.3
e4	*NearMiss*	-	0.4
e5	*NearMiss*	*Correlation*	0.1
e6	*NearMiss*	*Correlation*	0.2
e7	*NearMiss*	*Correlation*	0.3
e8	*NearMiss*	*Correlation*	0.4
e9	*NearMiss*	*Information Gain*	0.1
e10	*NearMiss*	*Information Gain*	0.2
e11	*NearMiss*	*Information Gain*	0.3
e12	*NearMiss*	*Information Gain*	0.4

Tabel IV
Treshold pada *Feature Selection*

Teknik *Feature Selection*	*Correlation*	*Information Gain*
Treshold	0.001	0.0001

Adapun perincian nilai batasan *(treshold)* pada Tabel IV, serta nilai keterhubungan fitur dengan target berdasarkan nilai koefisien masing-masing teknik *feature selection* sebagai berikut di dalam Tabel V.

Tabel V
Seleksi berdasarkan Nilai Keterhubungan Fitur pada Target

Nama Fitur	***Correlation***	***Information Gain***
age	-0.000983	0.000923
owns_car	-0.017104	0.001864
owns_house	-0.002693	0.009010
no_of_children	0.022876	0.000419
net_yearly_income	-0.022623	0.000265
no_of_days_employed	-0.069960	0.005183

Nama Fitur	*Correlation*	*Information Gain*
total_family_members	0.010421	0.003929
migrant_worker	0.034018	0.000221
yearly_debt_payments	-0.012829	0.000000
credit_limit	-0.013233	0.001083
credit_limit_used(%)	0.326641	0.090178
credit_score	-0.543091	0.220491
prev_defaults	0.771704	0.159668
default_in_last_6months	0.776078	0.146050
occupation_type_encoded	0.002612	0.004603
gender_encoded	0.057588	0.007336
in_hand_balance	-0.021972	0.000000
in_profit	-0.000552	0.013761
total_income_received	-0.059382	0.003187
employment_years	-0.069960	0.004636
log_net_yearly_income	-0.020157	0.000254
log_yearly_debt_payments	0.001881	0.000000
log_credit_limit	-0.013227	0.001127

Berdasarkan nilai keterhubungan fitur dengan target pada Tabel V, fitur dengan angka keterhubungan rendah diberikan *highlight* berwarna merah muda, berdasarkan *treshold* pada Tabel IV. Fitur dengan pewarnaan merah muda tersebut tidak digunakan pada percobaan dengan skenario yang sesuai.

B. Analisis Kinerja Tanpa Sampling Tanpa Feature Selection

Pertama dilakukan pengukuran performa awal tanpa penggunaan teknik *sampling* dan *feature selection* dengan beberapa rasio pembagian data yang didefinisikan saat memanggil fungsi *train_test_split*. Pada uji coba ini digunakan variasi rasio *data testing* sebesar 0.1, 0.2, 0.3, dan 0.4. Pewarnaan *table rows* dengan biru muda menandakan kinerja terbaik yang didapatkan saat percobaan dilakukan.

TABEL VI
PERBANDINGAN KINERJA TANPA *SAMPLING* DAN *FEATURE SELECTION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
a1	0.8891	0.8453	0.7841	0.9169	0.9778
a2	0.9005	0.8630	0.8065	0.9279	0.9794
a3	0.9017	0.8567	0.8104	0.9087	0.9785
a4	0.9028	0.8627	0.8119	0.9203	0.9788

Berdasarkan hasil perbandingan percobaan tanpa penggunaan *sampling* maupun menggunakan *feature selection*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.4. Skor tersebut sebesar 0.9028 dimana terdapat pada Tabel VI. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, semakin tinggi angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall*. Namun di sisi lain, pada skor selain kedua sebelumnya seperti *F1 Score, Precision,* dan *Accuracy,* tidak dapat disimpulkan bagaimana kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola.

C. Analisis Kinerja Tanpa Sampling dengan Correlation

Pada percobaan ini dibandingkan kinerja klasifikasi tanpa menggunakan teknik *sampling* dengan *feature selection* berdasarkan *correlation*. Berdasarkan hasil perbandingan percobaan tanpa penggunaan *sampling* dengan *feature selection* berdasarkan *correlation*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.4. Skor tersebut sebesar 0.9054. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, semakin tinggi angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall*. Tabel VII merupakan hasil perbandingan pada percobaan ini.

TABEL VII
PERBANDINGAN KINERJA TANPA *SAMPLING* DENGAN *CORRELATION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
a5	0.8965	0.8443	0.8011	0.8924	0.9772
a6	0.9042	0.8635	0.8147	0.9186	0.9792
a7	0.9029	0.8612	0.8122	0.9165	0.9793
a8	0.9054	0.8624	0.8179	0.9119	0.9786

D. Analisis Kinerja Tanpa Sampling dengan Info Gain

Pada percobaan ini dibandingkan kinerja klasifikasi tanpa menggunakan teknik *sampling* dengan *feature selection* berdasarkan *information gain*. Dari hasil perbandingan pada Tabel VIII dimana percobaan tanpa penggunaan *sampling* dengan *feature selection* berdasarkan *information gain*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.3. Skor tersebut sebesar 0.9065. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, semakin besar angka kinerja klasifikasi dalam skor AUC.

Hal yang sama terjadi pada skor *Recall* dan *F1 Score*. Pada *test-size* yang lebih besar skor menurun. Hal tersebut dapat terjadi karena kurangnya rasio data latih. Alasan lainnya adalah telah tereduksinya fitur sebagai bahan pemrediksi kelas target sehingga model klasifikasi kurang mampu mengenali kondisi-kondisi yang terdapat dalam data. Seperti sebelumnya, namun pada percobaan ini, skor lebih awal terjadi penurunan setelah dilakukan percobaan dengan *test-size* yang lebih besar. Pada sisi skor *Accuracy* tertinggi dalam percobaan ini masih dalam rasio *test-size* yang sama yaitu pada ukuran 0.3 dengan skor sebesar 0.9788.

TABEL VIII
PERBANDINGAN KINERJA TANPA *SAMPLING* DENGAN *INFO GAIN*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
a9	0.8960	0.8393	0.8011	0.8813	0.9763
a10	0.8988	0.8516	0.8052	0.9037	0.9774
a11	0.9065	0.8599	0.8205	0.9033	0.9788
a12	0.9009	0.8518	0.8099	0.8983	0.9769

E. Analisis Kinerja dengan RandomOverSampler Tanpa Feature Selection

Berdasarkan hasil perbandingan percobaan dengan penggunaan *RandomOverSampler* dan tanpa *feature selection*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.1. Skor tersebut sebesar 0.9355. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, semakin rendah angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall*. Namun di sisi lain, pada skor selain kedua sebelumnya seperti

F1 Score, Precision, dan *Accuracy,* tidak dapat ditarik kesimpulan dari kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola seperti yang disebutkan dalam Tabel IX.

TABEL IX
PERBANDINGAN KINERJA DENGAN *RANDOMOVERSAMPLER* TANPA *FEATURE SELECTION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
b1	0.9355	0.8103	0.8977	0.7383	0.9675
b2	0.9333	0.8316	0.8883	0.7818	0.9710
b3	0.9325	0.8237	0.8881	0.7680	0.9699
b4	0.9308	0.8406	0.8809	0.8039	0.9726

F. Analisis Kinerja dengan RandomOverSampler dan Correlation

Berdasarkan hasil perbandingan percobaan pada Tabel X dengan penggunaan *RandomOverSampler* dan dengan *feature selection* berdasarkan *correlation*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.2. Skor tersebut bernilai sebesar 0.9364. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, maka semakin rendah angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall*. Namun di sisi lain, pada skor selain kedua sebelumnya seperti *F1 Score, Precision,* dan *Accuracy,* tidak dapat ditarik kesimpulan dari kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola. Sama seperti percobaan sebelumnya tanpa *feature selection.*

TABEL X
PERBANDINGAN KINERJA DENGAN *RANDOMOVERSAMPLER* DAN *CORRELATION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
b5	0.9357	0.8113	0.8977	0.7400	0.9677
b6	0.9364	0.8222	0.8978	0.7583	0.9687
b7	0.9314	0.8209	0.8862	0.7646	0.9694
b8	0.9280	0.8386	0.8748	0.8053	0.9724

G. Analisis Kinerja dengan RandomOverSampler dan Info Gain

Berdasarkan hasil perbandingan percobaan pada Tabel XI dengan penggunaan *RandomOverSampler* dan dengan *feature selection* berdasarkan *information gain*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.1. Skor tersebut sebesar 0.9433. Dari percobaan tersebut tidak memiliki kecenderungan terhadap angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall*. Namun di sisi lain, pada skor *F1 Score,* dapat ditarik kesimpulan dari kecenderungannya yaitu semakin besar ukuran *test-size* semakin tinggi *F1 Score*, berbeda dari percobaan sebelumnya.

TABEL XI
PERBANDINGAN KINERJA DENGAN *RANDOMOVERSAMPLER* DAN *INFO GAIN*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
b9	0.9433	0.8131	0.9148	0.7318	0.9675
b10	0.9296	0.8148	0.8842	0.7555	0.9676
b11	0.9312	0.8162	0.8871	0.7557	0.9684
b12	0.9262	0.8181	0.8762	0.7673	0.9680

H. Analisis Kinerja dengan SMOTE Tanpa Feature Selection

Pada percobaan ini dibandingkan kinerja klasifikasi menggunakan SMOTE dan tanpa *feature selection*. Berdasarkan hasil perbandingan percobaan pada Tabel XII dengan SMOTE dan tanpa *feature selection*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.4 yaitu sebesar 0.9209. Dari percobaan tersebut memiliki kecenderungan dimana semakin besar ukuran *test-size* pada pembagian data, semakin tinggi angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall, F1 Score,* dan *Accuracy*. Namun di sisi lain, pada skor *Precision* tidak dapat ditarik kesimpulan dari kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola.

TABEL XII
PERBANDINGAN KINERJA DENGAN *SMOTE* TANPA *FEATURE SELECTION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
c1	0.9127	0.8303	0.8409	0.8199	0.9734
c2	0.9173	0.8442	0.8488	0.8396	0.9747
c3	0.9193	0.8451	0.8529	0.8374	0.9753
c4	0.9209	0.8523	0.8554	0.8492	0.9757

I. Analisis Kinerja dengan SMOTE dan Correlation

Berdasarkan hasil perbandingan percobaan dengan penggunaan SMOTE dan dengan *feature selection* berdasarkan *correlation*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.4 seperti sebelumnya. Skor tersebut bernilai sebesar 0.9208. Apabila dilihat dari Tabel XIII, pada percobaan memiliki kecenderungan terhadap skor AUC, *Recall*, *F1 Score, Precision,* dan *Accuracy,* yaitu semakin besar rasio *test-size* semakin tinggi pula angka kinerja klasifikasinya.

TABEL XIII
PERBANDINGAN KINERJA DENGAN *SMOTE* DAN *CORRELATION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
c5	0.9092	0.8324	0.8324	0.8324	0.9741
c6	0.9136	0.8374	0.8420	0.8329	0.9736
c7	0.9206	0.8459	0.8557	0.8363	0.9753
c8	0.9208	0.8511	0.8554	0.8469	0.9755

J. Analisis Kinerja dengan SMOTE dan Info Gain

Berdasarkan hasil perbandingan percobaan pada Tabel XIV dengan penggunaan SMOTE dan dengan *feature selection* berdasarkan *information gain*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.3. Skor tersebut sebesar 0.9183. Dari percobaan tersebut tidak memiliki kecenderungan terhadap skor AUC, begitupun pada skor *Recall*, *F1 Score, Precision,* dan *Accuracy,* tidak dapat ditarik kesimpulan dari kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola, sama seperti pada percobaan sebelumnya.

TABEL XIV
PERBANDINGAN KINERJA DENGAN *SMOTE* DAN *INFO GAIN*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
c9	0.9079	0.8319	0.8295	0.8343	0.9741
c10	0.9136	0.8429	0.8406	0.8452	0.9747
c11	0.9183	0.8425	0.8511	0.8341	0.9748
c12	0.9168	0.8438	0.8481	0.8396	0.9742

K. Analisis Kinerja dengan RandomUnderSampler Tanpa Feature Selection

Berdasarkan hasil perbandingan percobaan dengan penggunaan *RandomUnderSampler* dan tanpa *feature selection*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.1. Skor tersebut yaitu sebesar 0.9728. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, semakin rendah angka kinerja klasifikasi dalam skor AUC. Namun pada *test-size* dengan angka 0.4, skor menjadi tidak teratur dan acak. Hal yang sama terjadi pada skor *Recall*. Namun di sisi lain, pada skor selain kedua sebelumnya seperti *F1 Score, Precision,* dan *Accuracy,* tidak dapat ditarik kesimpulan dari kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola, seperti pada Tabel XV.

TABEL XV
PERBANDINGAN KINERJA DENGAN *RANDOMUNDERSAMPLER* TANPA *FEATURE SELECTION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
d1	0.9728	0.7630	0.9972	0.6180	0.9521
d2	0.9707	0.7767	0.9905	0.6388	0.9541
d3	0.9686	0.7719	0.9861	0.6341	0.9539
d4	0.9710	0.7915	0.9873	0.6605	0.9573

L. Analisis Kinerja dengan RandomUnderSampler dan Correlation

Berdasarkan hasil perbandingan percobaan pada Tabel XVI dengan penggunaan *RandomUnderSampler* dan dengan *feature selection* berdasarkan *correlation*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.2 berbeda dari sebelumnya. Skor tersebut bernilai sebesar 0.9730. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, maka semakin rendah angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor *Recall*. Kemudian pada skor selain kedua sebelumnya tidak dapat ditarik kesimpulan dari kecenderungannya karena skor yang dihasilkan bersifat acak atau tidak berpola.

TABEL XVI
PERBANDINGAN KINERJA DENGAN *RANDOMUNDERSAMPLER* DAN *CORRELATION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
d5	0.9648	0.7591	0.9801	0.6194	0.9519
d6	0.9730	0.7807	0.9946	0.6426	0.9550
d7	0.9691	0.7697	0.9880	0.6305	0.9532
d8	0.9697	0.7886	0.9853	0.6574	0.9567

M. Analisis Kinerja dengan RandomUnderSampler dan Info Gain

Berdasarkan hasil perbandingan percobaan dengan penggunaan *RandomUnderSampler* dan dengan *feature selection* berdasarkan *information gain*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.4. Skor tersebut sebesar 0.9694. Dari percobaan tersebut memiliki kecenderungan semakin besar ukuran *test-size* pada pembagian data, semakin tinggi angka kinerja klasifikasi dalam skor AUC. Hal yang sama terjadi pada skor metrik lain yaitu *Accuracy.* Skor *Accuracy* tertinggi berada pada posisi yang sama yaitu pada *test-size* sebesar 0.4, seperti pada Tabel XVII.

TABEL XVII
PERBANDINGAN KINERJA DENGAN *RANDOMUNDERSAMPLER* DAN *INFO GAIN*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
d9	0.9660	0.7588	0.9830	0.6179	0.9517
d10	0.9689	0.7764	0.9864	0.6401	0.9542
d11	0.9686	0.7716	0.9861	0.6338	0.9538
d12	0.9694	0.7863	0.9853	0.6542	0.9561

N. Analisis Kinerja dengan NearMiss Tanpa Feature Selection

Berdasarkan hasil perbandingan percobaan dengan penggunaan *NearMiss* dan tanpa *feature selection*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.1. Skor tersebut yaitu sebesar 0.9668. Dari percobaan tersebut tidak memiliki kecenderungan terhadap skor AUC dan pada skor metrik lainnya kecuali *Recall* dan *Accuracy*. Namun pada skor *Recall* didapatkan kecenderungan bahwa semakin besar ukuran *test-size* maka semakin rendah angka pada skor *Recall.* Adapun pada skor *Accuracy* terdapat kecenderungan meningkat pada angka ukuran *test-size* yang semakin besar, seperti pada Tabel XVIII.

TABEL XVIII
PERBANDINGAN KINERJA DENGAN *NEARMISS* TANPA *FEATURE SELECTION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
e1	0.9668	0.7646	0.9830	0.6257	0.9532
e2	0.9664	0.7754	0.9809	0.6411	0.9542
e3	0.9665	0.7737	0.9806	0.6389	0.9546
e4	0.9643	0.7830	0.9746	0.6544	0.9557

O. Analisis Kinerja dengan NearMiss dan Correlation

Berdasarkan hasil perbandingan percobaan pada Tabel XIX dengan penggunaan *NearMiss* dan dengan *feature selection* berdasarkan *correlation*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.3 berbeda dengan sebelumnya. Skor tersebut bernilai sebesar 0.9685. Dari percobaan tersebut memiliki kecenderungan semakin besar angka *test-size* semakin tinggi kinerja dalam setiap skor metriknya, namun pada angka 0.4, skor metrik AUC mengalami penurunan akibat adanya penurunan pada skor *Recall* dimana lebih berdampak daripada *Precision* karena memuat hasil klasifikasi kelas negatif, sehingga skor AUC yang didapatkan tidak lebih baik daripada angka *test-size* sebesar 0.3.

TABEL XIX
PERBANDINGAN KINERJA DENGAN *NEARMISS* DAN *CORRELATION*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
e5	0.9635	0.7585	0.9773	0.6198	0.9519
e6	0.9678	0.7772	0.9837	0.6423	0.9545
e7	0.9685	0.7769	0.9843	0.6417	0.9553
e8	0.9669	0.7854	0.9799	0.6553	0.9561

P. Analisis Kinerja dengan NearMiss dan Info Gain

Berdasarkan hasil perbandingan percobaan dengan penggunaan *NearMiss* dan dengan *feature selection*

berdasarkan *information gain*, skor AUC tertinggi terdapat pada *test-size* dengan ukuran rasio 0.1. Skor tersebut sebesar 0.9659. Dari percobaan tersebut tidak memiliki kecenderungan dalam skor AUC. Begitupun pada skor lainnya terjadi hal yang sama, seperti yang disebutkan dalam Tabel XX.

TABEL XX
PERBANDINGAN KINERJA DENGAN *NEARMISS* DAN *INFO GAIN*

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
e9	0.9659	0.7667	0.9801	0.6296	0.9539
e10	0.9651	0.7702	0.9796	0.6346	0.9529
e11	0.9655	0.7725	0.9787	0.6381	0.9544
e12	0.9648	0.7826	0.9759	0.6532	0.9555

Q. Analisis Kinerja Seluruh Skenario

Berdasarkan pendefinisian nama skenario sebelumnya, berikut Tabel XXI merupakan perbandingan dari keseluruhan penggunaan teknik *sampling* dan *feature selection* dengan skor metrik yang dihasilkan pada kinerja klasifikasi pemodelan.

TABEL XXI
PERBANDINGAN KINERJA SELURUH SKENARIO

Scenario	*AUC*	*F1*	*Recall*	*Precision*	*Accuracy*
a1	0.8891	0.8453	0.7841	0.9169	0.9778
a2	0.9005	0.8630	0.8065	0.9279	0.9794
a3	0.9017	0.8567	0.8104	0.9087	0.9785
a4	0.9028	0.8627	0.8119	0.9203	0.9788
a5	0.8965	0.8443	0.8011	0.8924	0.9772
a6	0.9042	0.8635	0.8147	0.9186	0.9792
a7	0.9029	0.8612	0.8122	0.9165	0.9793
a8	0.9054	0.8624	0.8179	0.9119	0.9786
a9	0.8960	0.8393	0.8011	0.8813	0.9763
a10	0.8988	0.8516	0.8052	0.9037	0.9774
a11	0.9065	0.8599	0.8205	0.9033	0.9788
a12	0.9009	0.8518	0.8099	0.8983	0.9769
b1	0.9355	0.8103	0.8977	0.7383	0.9675
b2	0.9333	0.8316	0.8883	0.7818	0.9710
b3	0.9325	0.8237	0.8881	0.7680	0.9699
b4	0.9308	0.8406	0.8809	0.8039	0.9726
b5	0.9357	0.8113	0.8977	0.7400	0.9677
b6	0.9364	0.8222	0.8978	0.7583	0.9687
b7	0.9314	0.8209	0.8862	0.7646	0.9694
b8	0.9280	0.8386	0.8748	0.8053	0.9724
b9	0.9433	0.8131	0.9148	0.7318	0.9675
b10	0.9296	0.8148	0.8842	0.7555	0.9676
b11	0.9312	0.8162	0.8871	0.7557	0.9684
b12	0.9262	0.8181	0.8762	0.7673	0.9680
c1	0.9127	0.8303	0.8409	0.8199	0.9734
c2	0.9173	0.8442	0.8488	0.8396	0.9747
c3	0.9193	0.8451	0.8529	0.8374	0.9753
c4	0.9209	0.8523	0.8554	0.8492	0.9757
c5	0.9092	0.8324	0.8324	0.8324	0.9741
c6	0.9136	0.8374	0.8420	0.8329	0.9736
c7	0.9206	0.8459	0.8557	0.8363	0.9753
c8	0.9208	0.8511	0.8554	0.8469	0.9755
c9	0.9079	0.8319	0.8295	0.8343	0.9741
c10	0.9136	0.8429	0.8406	0.8452	0.9747
c11	0.9183	0.8425	0.8511	0.8341	0.9748
c12	0.9168	0.8438	0.8481	0.8396	0.9742
d1	0.9728	0.7630	0.9972	0.6180	0.9521
d2	0.9707	0.7767	0.9905	0.6388	0.9541
d3	0.9686	0.7719	0.9861	0.6341	0.9539
d4	0.9710	0.7915	0.9873	0.6605	0.9573
d5	0.9648	0.7591	0.9801	0.6194	0.9519
d6	0.9730	0.7807	0.9946	0.6426	0.9550
d7	0.9691	0.7697	0.9880	0.6305	0.9532
d8	0.9697	0.7886	0.9853	0.6574	0.9567
d9	0.9660	0.7588	0.9830	0.6179	0.9517
d10	0.9689	0.7764	0.9864	0.6401	0.9542
d11	0.9686	0.7716	0.9861	0.6338	0.9538
d12	0.9694	0.7863	0.9853	0.6542	0.9561
e1	0.9668	0.7646	0.9830	0.6257	0.9532
e2	0.9664	0.7754	0.9809	0.6411	0.9542
e3	0.9665	0.7737	0.9806	0.6389	0.9546
e4	0.9643	0.7830	0.9746	0.6544	0.9557
e5	0.9635	0.7585	0.9773	0.6198	0.9519
e6	0.9678	0.7772	0.9837	0.6423	0.9545
e7	0.9685	0.7769	0.9843	0.6417	0.9553
e8	0.9669	0.7854	0.9799	0.6553	0.9561
e9	0.9659	0.7667	0.9801	0.6296	0.9539
e10	0.9651	0.7702	0.9796	0.6346	0.9529
e11	0.9655	0.7725	0.9787	0.6381	0.9544
e12	0.9648	0.7826	0.9759	0.6532	0.9555

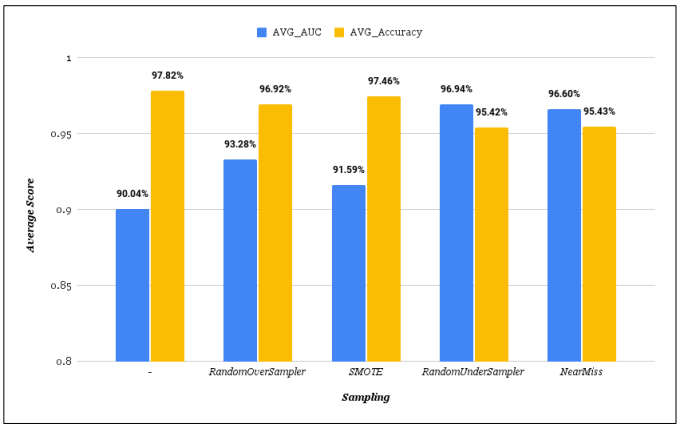
Dari hasil rekap seluruh skenario pada Tabel XXI dengan teknik *sampling* dan *feature selection*yang berbeda didapatkan skor AUC tertinggi pada skenario (d6) dimana dengan penggunaan *RandomUnderSampler* dengan *test-size* 0.2 dan dengan *feature selection* berdasarkan *correlation* yaitu sebesar 0.9730. Sedangkan untuk *Accuracy* tertinggi dengan angka 0.9794 pada skenario (a2) yaitu tanpa penggunaan teknik *sampling,* tanpa penggunaan teknik *feature selection,* dan *test-size* di ukuran 0.2.

Adapun perbandingan rerata skor dari setiap penggunaan teknik *sampling* dan *feature selection*. Gbr. 4 dan Gbr. 5 adalah bagan perbandingan dari rerata skor pada setiap teknik *sampling* dan *feature selection* baik berdasarkan skor AUC maupun skor *Accuracy*.

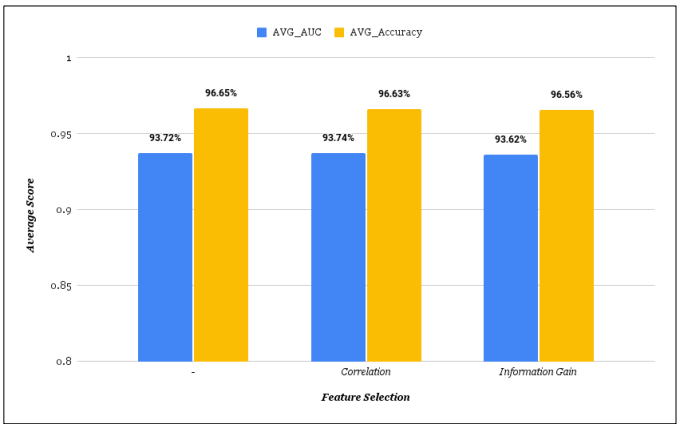
Pada perbandingan dalam bagan Gbr. 4 didapatkan bahwa rerata skor AUC tertinggi yaitu pada penggunaan teknik *sampling* dengan *RandomUnderSampler,* namun rerata skor *Accuracy* tertinggi tanpa menggunakan teknik *sampling* apapun. Pada Gbr. 5 perbandingan pada teknik *feature selection* rerata tertinggi AUC yaitu menggunakan *correlation*, sedangkan rerata tertinggi *Accuracy* pada skenario tanpa *feature selection*.

Dari perbandingan kinerja secara menyeluruh pada seluruh skenario baik secara metrik individu maupun rerata, didapatkan kinerja terbaik pada penggunaan teknik *sampling Random UnderSampler* dengan *test-size* 0.2 dan dengan *feature selection* berdasarkan *correlation* yaitu AUC sebesar 0.9730.

Sehingga pada pemodelan akan dilakukan pengujian menggunakan teknik pra-pemrosesan terbaik tersebut.



Gbr. 4 Perbandingan Rerata AUC dan *Accuracy* pada Seluruh Skenario *Sampling*



Gbr. 5 Perbandingan Rerata AUC dan *Accuracy* pada Seluruh Skenario *Feature Selection*

R. *Pengujian Model*

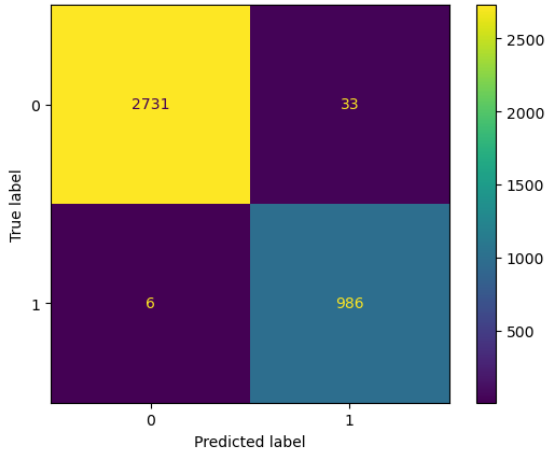
Pada perancangan model telah didapatkan model dengan kinerja tertinggi yaitu model XGBoost dimana menggunakan teknik *RandomUnderSampler, feature selection by correlation,* dan *test size 0.2*. Model tersebut dilakukan pengujian pada tahap ini. Untuk data yang diujikan adalah data yang telah ada namun terpisah dari awal sehingga tidak melalui tahap *training* sebelumnya yaitu *'test.csv'*. Data tersebut dilakukan *handling* atau penanganan terlebih dahulu karena masih terdapat data yang memiliki nilai kosong dan juga data dengan jenis kategorikal. Selanjutnya model mengklasifikasikan '*credit_card_default*' sebagai variabel *target*.

Hasil klasifikasi yaitu label 0 atau 1 seperti pada Gbr. 6. Dimana label 0 berarti klasifikasi mengatakan status sebagai non-default atau debitur tidak akan mengalami kegagalan bayar. Sedangkan label 1 sebaliknya, debitur diklasifikasikan mengalami status default atau kendala dalam pembayaran kembali kredit.

Berdasarkan pengujian, model yang telah dilakukan perbandingan kinerja atau performa mampu untuk melakukan klasifikasi status default kredit dari data uji dengan sangat baik. Oleh karena itu, tahap selanjutnya yaitu melakukan pengujian performa dan ketahanan model terhadap data baru. Data tersebut didapatkan dari data hasil klasifikasi sebelumnya sebanyak 11.383 sampel data yang telah dijadikan satu ke dalam *dataset* lama dan berformat '*.csv'*. Kemudian model dilakukan *re-training* menggunakan data yang telah diperbarui tersebut dan didapatkan *confusion matrix* seperti pada Gbr. 7. Setelah itu model disimpan menjadi format '*.sav*' untuk digunakan pada klasifikasi selanjutnya.

```
✓ ...
... 0        0
    1        0
    2        1
    3        0
    4        1
            ..
    11378    0
    11379    0
    11380    0
    11381    0
    11382    0
    Name: credit_card_default, Length: 11383, dtype: int64
```

Gbr. 6 Hasil Klasifikasi Kredit Default pada Data Uji "*test.csv"* Berbasis pada XGBoost dengan *RandomUnderSampler* dan *feature selection by correlation*



Gbr. 7 *Confusion Matrix* Klasifikasi Data Uji "*test.csv"* Berbasis pada XGBoost dengan *RandomUnderSampler, feature selection by correlation, dan test_size 0.2*

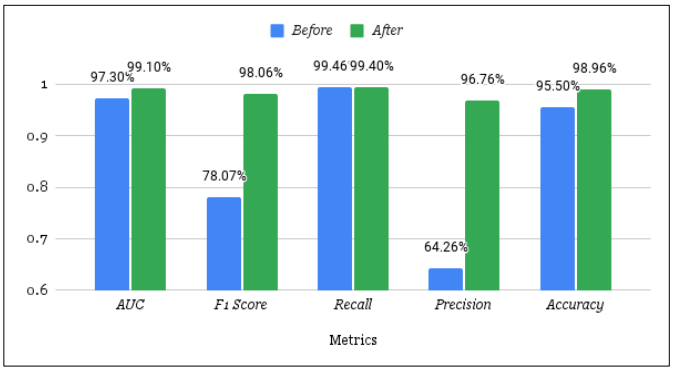
Hasil kinerja model yang telah dilakukan *re-training* dengan *dataset* baru menunjukkan hasil yang sangat baik seperti pada Gbr. 7 dengan skor yang lebih tinggi daripada kinerja model saat *training* di awal. Hal ini menandakan bahwa model memiliki kemampuan sangat baik pada saat melakukan klasifikasi pada data yang baru.

TABEL XXII
PERBANDINGAN KINERJA KLASIFIKASI SEBELUM DAN SESUDAH *RE-TRAINING* BERBASIS PADA MODEL XGBOOST DENGAN *RANDOMUNDERSAMPLER, FEATURE SELECTION BY CORRELATION, TEST_SIZE 0.2*

Metrics	***Before***	***After***	***Δ***
AUC	0.9730	0.9910	0.0180
F1 Score	0.7807	0.9806	0.1999
Recall	0.9946	0.9940	-0.0006

Precision	0.6426	0.9676	0.3250
Accuracy	0.9550	0.9896	0.0346

Kinerja pada model sesudah dilakukan *re-training* menggunakan hasil klasifikasi data uji meningkat seperti pada Tabel XXII dimana skor AUC dari 0.9730 menjadi 0.9910, selanjutnya yaitu *Recall* sebesar 0.9940. Kemudian pada skor *Accuracy* 0.9896, skor *Precision* 0.9676, dan skor *F1* 0.9806. Pada perubahan skor terdapat peningkatan meskipun tidak terlalu signifikan seperti pada Gbr. 8.



Gbr. 8 Perbandingan Kinerja Klasifikasi Sebelum dan Sesudah *Re-Training* Berbasis pada XGBoost dengan *RandomUnderSampler, feature selection by correlation, dan test_size 0.2*

IV. KESIMPULAN

Hasil kinerja yang baik dari model diperoleh dengan pemilihan teknik *sampling* dan teknik *feature selection* yang tepat. Dalam penelitian ini, penggunaan teknik *sampling* mampu meningkatkan kinerja model dari klasifikasi kredit default. Kinerja tertinggi dicapai ketika teknik *sampling RandomUnderSampler* diterapkan. Hal ini dikarenakan teknik ini sangat diperlukan ketika data yang digunakan memiliki sampel kelas yang tidak seimbang (*imbalanced class*) agar mendapatkan hasil *training* yang baik dan model dapat mempelajari data secara lebih luas serta tidak bias. Lebih lanjut, performa tertinggi dicapai pada penerapan teknik evaluasi *train-test-split* rasio 0.2 dan *feature selection* berdasarkan *correlation* dengan *treshold* 0.001. Adapun level AUC yang tertinggi dicapai sebesar 0.9730. Setelah, *re-training* dilakukan pada data uji, level AUC meningkat hingga menjadi 0.9910. Hal ini menandakan bahwa pemilihan teknik pra-pemrosesan yang tepat dapat berpengaruh pada kinerja model klasifikasi kredit default berbasis XGBoost.

V. SARAN

Penelitian pada topik klasifikasi kredit default ini belum dapat digunakan oleh masyarakat secara luas. Karena belum adanya pengembangan suatu aplikasi yang dapat digunakan secara langsung. Besar harapan penelitian ini dapat dijadikan sebagai rujukan pada penelitian selanjutnya dengan topik yang relevan. Adapun pada penelitian ini hanya mengeksplorasi teknik *sampling* dan *feature selection* berbasis model XGBoost. Maka peneliti memiliki harapan besar pada penelitian selanjutnya dapat dilakukan eksplorasi ataupun implementasi dengan teknik lain.

UCAPAN TERIMA KASIH

Puji Syukur terhadap Tuhan Yang Maha Esa atas berkat dan rahmat-Nya dan atas kekuatan, kesehatan, serta kesabaran yang telah diberikan sehingga penelitian ini dapat terselesaikan. Terima kasih sebesar-besarnya kepada kedua Orang Tua dan Keluarga yang senantiasa mendoakan dan memberikan dukungan, kepada Ibu Yuni Yamasari yang telah memberikan perhatian terbaiknya selama penelitian. Terima kasih banyak juga kepada teman-teman yang telah membantu dan mendukung sebaik-baiknya hingga penelitian ini terselesaikan.

REFERENSI

[1] Afifudin, M., & Rizki, A. M. (2023). Analisis Perbandingan Penggunaan Model *Machine Learning* Pada Kasus Deteksi Kemampuan Calon Klien Dalam Membayar Kembali Pinjaman.

[2] Ihsan, A., & Mutahir, A. (2023). Seduction Dan Simulakra Pada Layanan Spaylater (Vol. 12, Issue 1).

[3] Sari, I. M., Siregar, S., & Harahap, I. (2020). Manajemen Risiko Kredit Bagi Bank Umum. Seminar Nasional Teknologi Komputer & Sains (SAINTEKS), 1(1), 553–557.

[4] Montoya, A., Odintsov, K., & Kotek, M. (2018). *Home Credit Default Risk*. Kaggle. https://kaggle.com/competitions/home-credit-default-risk

[5] Bhattacharya, A., Biswas, S. Kr., & Mandal, A. (2023). *Credit risk evaluation: a comprehensive study. Multimedia Tools and Applications*, 82(12), 18217–18267. https://doi.org/10.1007/s11042-022-13952-3

[6] Givari, M. R., Mochamad, R., & Sulaeman2, Y. U. (2022). Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit. 16(1). https://journal.uniku.ac.id/index.php/ilkom

[7] Budiman, S., Sunyoto, A., & Nasiri, A. (2021). Analisa Performa Penggunaan *Feature Selection* untuk Mendeteksi In*trusion Detection Systems* dengan Algoritma *Random Forest Classifier*. SISTEMASI: Jurnal Sistem Informasi, 10(3), 754–760.

[8] Basak, P. (2021). *AmExpert CodeLab 2021*. Kaggle. https://www.kaggle.com/datasets/pradip11/amexpert-codelab-2021/data

[9] Yulianti, S. E. H., Soesanto, O., & Sukmawaty, Y. (2022). Penerapan Metode *Extreme Gradient Boosting* (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics: Theory and Applications*, 21–26.

[10] Primandari, A. H. (2020). Implementasi Metode *Random Forest* Dan Xgboost Pada Klasifikasi *Customer Churn*.

[11] Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). *Machine learning with oversampling and undersampling techniques: overview study and experimental results. 2020 11th International Conference on Information and Communication Systems (ICICS),* 243–248.

[12] Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). *A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. Journal of Applied Science and Technology Trends*, 1(2), 56–70. https://doi.org/10.38094/jastt1224

[13] Zhang, W., Wu, C., Zhong, H., Li, Y., & Wang, L. (2021). *Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. Geoscience Frontiers,* 12(1), 469–477. https://doi.org/10.1016/j.gsf.2020.03.007

[14] Sunata, H. (2020). Komparasi Tujuh Algoritma Identifikasi *Fraud* ATM Pada PT. Bank Central Asia Tbk. JATISI (Jurnal Teknik Informatika Dan Sistem Informasi), 7(3), 441–450.

[15] Fadillah Hermawan, F., & Yamasari, Y. (2022). Implementasi *K-Nearest Neighbor* dengan Pemilihan Fitur pada Aplikasi Prediksi Kelayakan Pengajuan Pinjaman. *Journal of Informatics and Computer Science*, 03.

[16] Chowdhury, M. M., Ayon, R. S., & Hossain, M. S. (2023*). Diabetes Diagnosis through Machine Learning: Investigating Algorithms and Data Augmentation for Class Imbalanced BRFSS Dataset. MedRxiv*, 2010–2023.

[17] Ghosh, S., & Islam, M. A. (2023). *Performance Evaluation and Comparison of Heart Disease Prediction Using Machine Learning*

Methods with Elastic Net Feature Selection. American Journal of Applied Mathematics and Statistics, 11(2), 35–49.