KLASIFIKASI *RENEWAL STATUS* PADA PENGGUNA PLATFORM BIMBINGAN BELAJAR DARING DENGAN METODE *RANDOM FOREST*

SKRIPSI

Oleh :
JENNY ANGELS AYU INTANIA
NIM. 220605110156

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026

KLASIFIKASI *RENEWAL STATUS* PADA PENGGUNA PLATFORM BIMBINGAN BELAJAR DARING DENGAN METODE *RANDOM FOREST*

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
JENNY ANGELS AYU INTANIA
NIM. 220605110156

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

HALAMAN PERSETUJUAN

KLASIFIKASI *RENEWAL STATUS* PADA PENGGUNA PLATFORM BIMBINGAN BELAJAR DARING DENGAN METODE *RANDOM FOREST*

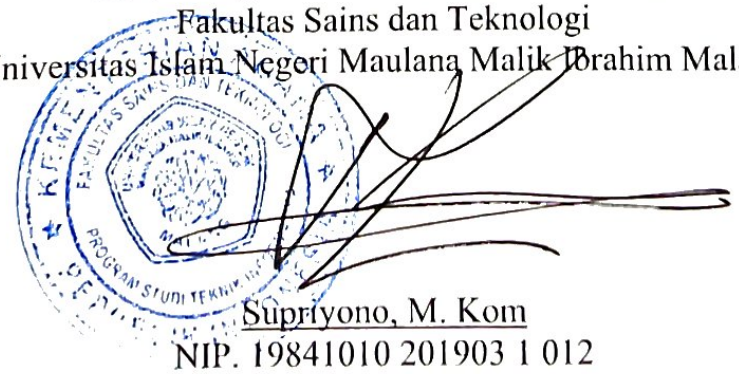
SKRIPSI

Oleh :
JENNY ANGELS AYU INTANIA
NIM. 220605110156

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 2 Juni 2026

Pembimbing I,	Pembimbing II,
Prof. Dr. Suhartono, S.Si M.Kom	Dr. Ir. Yunifa Miftachul Arif, S.ST., MT., SMIEEE
NIP. 19680519 200312 1 001	NIP. 19830616 201101 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M. Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

KLASIFIKASI *RENEWAL STATUS* PADA PENGGUNA PLATFORM BIMBINGAN BELAJAR DARING DENGAN METODE *RANDOM FOREST*

SKRIPSI

Oleh :
JENNY ANGELS AYU INTANIA
NIM. 220605110156

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 12 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: Dr. Irwan Budi Santoso, M.Kom NIP. 19770103 201101 1 004	( )
Anggota Penguji I	: Tri Mukti Lestari, M.Kom NIP. 19911108 202012 2 005	( )
Anggota Penguji II	: Prof. Dr. Suhartono, S.Si M.Kom NIP. 19680519 200312 1 001	( )
Anggota Penguji III	: Dr. Ir. Yunifa Miftachul Arif, S.ST., MT., SMIEE NIP. 19830616 201101 1 004	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Jenny Angels Ayu Intania
NIM	: 220605110156
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Klasifikasi *Renewal Status* Pada Platform Bimbingan Belajar Daring Dengan Metode *Random Forest*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 22 Juni 2026
Yang membuat pernyataan,

Jenny Angels Ayu Intania
NIM.220605110156

MOTTO

"Keberadaan kita di dunia ini adalah sebuah keajaiban, meskipun kehidupan tidak selalu berjalan dengan sempurna."

- Dr. Gia -

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT atas segala rahmat, kasih sayang, dan pertolongan-Nya, karya sederhana ini penulis persembahkan kepada:

Ayah Ibu Tercinta,

Terima kasih atas cinta, doa, pengorbanan, dan kasih sayang yang tidak pernah berhenti mengiringi setiap langkah penulis. Terima kasih telah menjadi alasan terbesar penulis untuk terus kuat. Terima kasih karena selalu percaya, mendukung, dan mendoakan penulis dalam setiap langkah kehidupan. Meskipun kini Ibu telah berada di sisi Allah SWT dan tidak lagi menemani penulis secara langsung, penulis yakin kasih sayang dan doa Ibu tidak pernah benar-benar pergi. Semoga Ibu bangga melihat penulis dari tempat terbaik di sisi-Nya.

Keluarga Tercinta,

Terima kasih karena selalu memberikan doa, perhatian, dan semangat. Kehadiran kalian menjadi kekuatan terbesar yang membuat penulis mampu bertahan hingga menyelesaikan perjalanan ini.

Dan untuk diri penulis sendiri,

Terima kasih telah memilih untuk tetap bertahan. Terima kasih karena tidak menyerah meskipun perjalanan ini dipenuhi air mata, rasa lelah, dan kehilangan. Hari ini menjadi bukti bahwa setiap doa, usaha, dan kesabaran tidak pernah sia-sia.

Semoga karya sederhana ini menjadi awal dari langkah yang lebih bermakna, membawa manfaat bagi banyak orang, serta menjadi bagian dari langkah menuju masa depan yang lebih baik.

KATA PENGANTAR

Alhamdulillahirabbil ‘alamin, segala puji hanya milik Allah Subḥānahu wa Ta‘ālā atas segala nikmat dan kasih sayang-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Klasifikasi Renewal Status Pada Platform Bimbingan Belajar Daring Dengan Metode Random Forest” dengan baik. Penulis menyadari bahwa dalam proses penyusunan skripsi ini terdapat berbagai keterbatasan dan tantangan, namun berkat bantuan, bimbingan, dukungan, serta motivasi dari berbagai pihak, skripsi ini dapat diselesaikan dengan baik. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. Suhartono, S.Si, M.Kom., selaku Dosen Pembimbing I, yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan arahan, bimbingan, serta masukan yang sangat berharga selama proses penyusunan skripsi ini.
5. Dr. Ir. Yunifa Miftachul Arif, S.ST., MT., SMIEEE, selaku Dosen Pembimbing II, yang telah memberikan bimbingan, arahan, serta berbagai saran yang membangun dalam penyusunan skripsi ini.

6. Dr. Irwan Budi Santoso, M.Kom., selaku Ketua Penguji, yang telah memberikan saran, dan masukan yang konstruktif sehingga skripsi ini menjadi lebih baik.
7. Tri Mukti Lestari, M.Kom., selaku Anggota Penguji I, yang telah memberikan berbagai saran, masukan, dan evaluasi yang sangat bermanfaat dalam penyempurnaan skripsi ini.
8. Seluruh Dosen, Admin, dan Staf Program Studi Teknik Informatika, yang telah berbagi ilmu, tenaga, dan kebaikan selama masa studi ini, baik secara langsung maupun tidak langsung.
9. Kedua orang tua tercinta, Ayah Anang Hartadi dan Ibu Mei Supriyanti yang sangat penulis sayangi dan cintai. Terima kasih atas segala doa, kasih sayang, dan pengorbanan yang telah diberikan kepada penulis. Teruntuk Ayah, terima kasih telah menjadi sosok yang selalu menguatkan penulis untuk tetap melanjutkan hidup dan terus berjuang menyelesaikan pendidikan ini. Teruntuk Ibu, terima kasih atas cinta, doa, dan segala kebaikan yang telah Ibu tanamkan semasa hidup. Semoga Allah SWT senantiasa melimpahkan rahmat dan menempatkan Ibu di tempat terbaik di sisi-Nya. Semoga pencapaian ini menjadi salah satu bentuk bakti yang dapat penulis persembahkan kepada Ayah dan Ibu.
10. Keluarga tercinta, terutama Mamak, Mbahkong, serta kedua adik penulis, Lana dan Malva. Terima kasih atas doa, kasih sayang, perhatian, dan dukungan yang selalu diberikan kepada penulis. Terima kasih telah menjadi

penguat dan tempat penulis kembali bangkit dalam setiap proses perjuangan hingga mampu menyelesaikan skripsi ini.

11. Sahabat-sahabat terbaik penulis, Ilmi, Leny, Annisa, Runa, Irfana, Jihan, Hafizah, Safa, dan Sella. Terima kasih atas kebersamaan, doa, dukungan, serta semangat yang telah diberikan kepada penulis selama menjalani perkuliahan hingga proses penyusunan skripsi ini. Terima kasih telah menjadi tempat berbagi cerita, tawa, dan saling menguatkan dalam setiap perjalanan yang telah dilalui.
12. Seluruh teman-teman Angkatan 2022 Teknik Informatika yang telah banyak membantu, mendukung, dan memotivasi dalam menyelesaikan penyusunan skripsi ini.
13. Terakhir, kepada diri penulis sendiri. Terima kasih telah bertahan, bangkit, dan terus melangkah di tengah berbagai ujian kehidupan. Terima kasih karena tidak menyerah, meskipun perjalanan ini tidak selalu mudah. Semoga setiap proses yang telah dilalui menjadi pelajaran berharga untuk menjadi pribadi yang lebih kuat, lebih ikhlas, dan lebih baik di masa yang akan datang.

Malang, 19 Juni 2026

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Intania, Jenny Angels Ayu. 2026. **Klasifikasi *Renewal Status* pada Pengguna Platform Bimbingan Belajar Daring dengan Metode *Random Forest.*** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, S.Si M.Kom, (II) Dr. Ir. Yunifa Miftachul Arif, S.ST., M.T., SMIEEE.

Kata Kunci: *Churn Prediction,* Klasifikasi, *Random Forest*, Renewal Status, SMOTE.

Fenomena *churn* atau tidak diperpanjangnya langganan pengguna merupakan salah satu permasalahan yang dapat memengaruhi retensi pengguna pada platform bimbingan belajar daring. Penelitian ini bertujuan untuk menganalisis performa algoritma Random Forest dalam mengklasifikasikan *Renewal Status* pengguna menjadi kelas *churn* dan *non-churn*. Dataset yang digunakan berasal dari Kaggle dengan total 14.843 data. Tahapan penelitian meliputi *preprocessing* data, *feature selection*, normalisasi menggunakan MinMax Scaling, penyeimbangan data menggunakan SMOTE, serta optimasi parameter menggunakan GridSearchCV. Evaluasi model dilakukan menggunakan Accuracy, Precision, Recall, F1-Score, ROC AUC, Confusion Matrix, dan Stratified K-Fold Cross Validation. Hasil penelitian menunjukkan bahwa model Random Forest mampu melakukan klasifikasi *Renewal Status*, namun performanya masih belum optimal. Skenario terbaik diperoleh pada penggunaan 1.000 data dengan nilai Accuracy 71%, Precision 23%, Recall 25%, F1-Score 24%, dan ROC AUC 54%. Hasil tersebut menunjukkan bahwa kemampuan model dalam membedakan pengguna *churn* dan *non-churn* masih terbatas, sehingga diperlukan pengembangan lebih lanjut untuk meningkatkan performa klasifikasi.

ABSTRACT

Intania, Jenny Angels Ayu. 2026. **Classification of Renewal Status on Online Tutoring Platform Users Using the Random Forest Method.** Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Advisors: (I) Prof. Dr. Suhartono, S.Si M.Kom, (II) Dr. Ir. Yunifa Miftachul Arif, S.ST., M.T., SMIEEE.

The phenomenon of user churn, or the discontinuation of subscription renewal, is one of the challenges that can affect user retention in online tutoring platforms. This study aims to analyze the performance of the Random Forest algorithm in classifying users' Renewal Status into churn and non-churn categories. The dataset used in this study was obtained from Kaggle and consists of 14,843 records. The research process includes data preprocessing, feature selection, MinMax Scaling normalization, data balancing using SMOTE, and hyperparameter optimization using GridSearchCV. Model evaluation was conducted using Accuracy, Precision, Recall, F1-Score, ROC AUC, Confusion Matrix, and Stratified K-Fold Cross Validation. The results indicate that the Random Forest model is capable of performing Renewal Status classification; however, its performance is still not optimal. The best scenario was obtained using 1,000 data samples, achieving an Accuracy of 71%, Precision of 23%, Recall of 25%, F1-Score of 24%, and ROC AUC of 54%. These findings suggest that the model's ability to distinguish between churn and non-churn users remains limited. Therefore, further improvements are needed to enhance the classification performance.

Keywords: Churn Prediction, Classification, Random Forest, , Renewal Status, SMOTE.

مستخلص البحث

إنتانيا، جيني أنجلز أيو. ٢٠٢٦ .تصنيف حالة تجديد الاشتراك لدى مستخدمي منصات التعليم الإلكتروني باستخدام خوارزمية الغابة العشوائية .(Random Forest) أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانغ .المشرفان :(١) الأستاذ الدكتور سوهارتونو، S.Si، .M.Kom(٢) الدكتور المهندس يونيفا مفتاح العارف، .S.ST، .M.T، .SMIEEE

ال كلمات المفتاحية: التصنيف ،SMOTE ،الغابة العشوائية، التنبؤ بالتسرب، حالة تجديد الاشتراك

أو عدم تجديد الاشتراك من المشكلات التي تؤثر على معدل الاحتفاظ (Churn) تُعد ظاهرة تسرب المستخدمين (Random Forest) بالمستخدمين في منصات التعليم الإلكتروني .تهدف هذه الدراسة إلى تحليل أداء خوارزمية الغابة العشوائية في تصنيف حالة تجديد الاشتراك إلى فئتي المستخدمين المتسربين وغير المتسربين .اعتمدت الدراسة على مجموعة بيانات مأخوذة من تضم 14,843 سجلًا .شملت مراحل البحث معالجة البيانات الأولية، واختيار الخصائص، وتطبيع البيانات باستخدام Kaggle منصة بالإضافة إلى تحسين معلمات النموذج باستخدام ،SMOTE وموازنة الفئات باستخدام تقنية ،MinMax Scaling والدقة النوعية ،(Recall) والاستدعاء ،(Accuracy) وتم تقييم النموذج باستخدام مقاييس الدقة .GridSearchCV Stratified K-Fold وطريقة ،(Confusion Matrix) ومصفوفة الالتباس ،ROC AUCو ،F1 ودرجة ،(Precision) أظهرت النتائج أن نموذج الغابة العشوائية قادر على تصنيف حالة تجديد الاشتراك، إلا أن أداءه ما زال غير .Cross Validation مثالي .وقد حقق أفضل أداء عند استخدام 1000 سجل بيانات، حيث بلغت الدقة 71%، والدقة النوعية 23%، والاستدعاء بنسبة 54% .وتشير هذه النتائج إلى أن قدرة النموذج على التمييز بين ROC AUC بنسبة 24%، وقيمة F1 ودرجة ،25% .المستخدمين المتسربين وغير المتسربين ما زالت محدودة، مما يستدعي إجراء مزيد من التحسينات لرفع كفاءة التصنيف

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital telah membawa perubahan besar dalam bidang pendidikan, salah satunya melalui hadirnya platform bimbingan belajar daring. Platform ini memungkinkan proses belajar mengajar berlangsung tanpa batas ruang dan waktu, sehingga siswa dapat mengakses materi, mengerjakan latihan, serta berinteraksi dengan tutor secara fleksibel melalui perangkat digital. Di Indonesia, platform seperti Ruangguru, Zenius, Quipper, dan Pahamify telah menjadi bagian penting dalam mendukung proses pembelajaran modern (Aisyah et al., 2025).

Namun, salah satu tantangan utama yang dihadapi oleh platform bimbingan belajar daring adalah fenomena *churn*, yaitu kondisi ketika pengguna berhenti menggunakan layanan atau tidak memperpanjang masa berlangganan (Bhoite et al., 2023). Fenomena ini sering kali terjadi karena berbagai faktor, seperti penurunan motivasi belajar, ketidakpuasan terhadap konten, kendala teknis, atau pergeseran kebutuhan siswa setelah mencapai tujuan tertentu (Flood, 2020). Dalam konteks bisnis berbasis langganan (*subscription-based model*), *churn* berdampak langsung pada penurunan pendapatan dan meningkatkan biaya operasional, karena biaya memperoleh pengguna baru umumnya lebih tinggi dibanding mempertahankan pengguna lama (Maan & Maan, 2023).

Permasalahan umum yang dihadapi platform bimbingan belajar daring adalah belum optimalnya kemampuan untuk mengidentifikasi pengguna yang

berpotensi *churn* secara dini. Tanpa informasi tersebut, perusahaan sulit melakukan intervensi retensi, seperti personalisasi materi, pemberian promosi, atau peningkatan dukungan belajar (Jang et al., 2021). Oleh karena itu, kemampuan memprediksi status pengguna menjadi penting untuk menjaga stabilitas bisnis dan efektivitas layanan yang diberikan.

Pendekatan berbasis *machine learning* menawarkan solusi efektif untuk masalah ini, karena mampu menganalisis pola perilaku pengguna dari data historis dan melakukan prediksi secara otomatis serta akurat (Verbeke et al., 2011). Melalui data aktivitas pengguna, seperti tingkat penyelesaian pembelajaran, poin yang diperoleh, durasi penggunaan platform, maupun keterlibatan pengguna terhadap layanan tertentu, model machine learning dapat membantu mengidentifikasi pengguna yang berpotensi berhenti menggunakan layanan.

Salah satu algoritma yang terbukti efektif dalam prediksi *churn* adalah *Random Forest*, metode *ensemble learning* yang menggabungkan banyak pohon keputusan untuk menghasilkan prediksi yang stabil dan akurat (Glandorf et al., 2024). *Random Forest* memiliki kemampuan yang baik dalam menangani data berukuran besar, tahan terhadap noise, serta mampu memberikan nilai feature importance untuk mengetahui tingkat pengaruh masing-masing fitur terhadap hasil klasifikasi (Orji & Vassileva, 2022).

Penelitian ini menggunakan dataset *Tutoring Platform User Dataset* untuk melakukan klasifikasi *Renewal_Status* menggunakan algoritma *Random Forest Classifier*. Fitur yang digunakan difokuskan pada seluruh fitur numerik serta dua fitur kategorikal yang relevan terhadap perilaku pengguna, yaitu Tutoring dan

Subscription_Tier. Selanjutnya dilakukan *feature selection* menggunakan *feature importance Random Forest* untuk memperoleh fitur yang paling berpengaruh terhadap klasifikasi *Renewal_Status*. Selain itu, penelitian ini menggunakan beberapa skenario jumlah data, yaitu 1000 data, 2000 data, 5000 data, dan full dataset untuk melihat pengaruh ukuran data terhadap performa model.

Selain permasalahan performa model, penelitian ini juga memperhatikan kondisi ketidakseimbangan kelas *(imbalanced data)*, di mana jumlah data *non-churn* lebih besar dibandingkan *churn*. Kondisi tersebut dapat menyebabkan model cenderung bias terhadap kelas mayoritas sehingga kemampuan mendeteksi *churn* menjadi kurang optimal. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) pada data latih setelah proses pembagian data *(train-test split)*. Penerapan SMOTE bertujuan untuk menyeimbangkan distribusi kelas pada data latih tanpa mengubah distribusi data uji sehingga dapat membantu model mengenali kedua kelas secara lebih optimal (Orji & Vassileva, 2022).

Fenomena pengguna yang berhenti belajar di platform daring dapat pula dipandang dari perspektif Islam sebagai bentuk tantangan dalam menjaga komitmen terhadap ilmu. Al-Qur'an menegaskan pentingnya menuntut ilmu dan konsistensi dalam mengamalkannya. Hal ini sesuai dengan firman Allah SWT dalam QS. Az-Zumar ayat 9:

اَمَّنْ هُوَ قَانِتٌ اٰنَاۤءَ الَّيْلِ سَاجِدًا وَّقَاۤىِٕمًا يَّحْذَرُ الْاٰخِرَةَ وَيَرْجُوْا رَحْمَةَ رَبِّهٖۗ قُلْ هَلْ يَسْتَوِى

الَّذِيْنَ يَعْلَمُوْنَ وَالَّذِيْنَ لَا يَعْلَمُوْنَۗ اِنَّمَا يَتَذَكَّرُ اُولُوا الْاَلْبَابِ ۝٩

"(Apakah sama orang yang beribadah di waktu malam dengan sujud dan berdiri karena takut akan akhirat dan mengharapkan rahmat Tuhannya) dengan orang yang tidak berbuat demikian? Katakanlah, "Apakah sama orang-orang yang mengetahui dengan orang-orang yang tidak mengetahui?" Sesungguhnya hanya orang-orang yang berakal sehat yang dapat mengambil pelajaran." (QS. Az-Zumar: 9)

Ayat ini menegaskan bahwa ilmu harus digunakan untuk menciptakan kemaslahatan dan kemajuan, bukan sekadar untuk pengetahuan tanpa nilai moral. Dalam konteks penelitian ini, fenomena *churn* pada platform bimbingan belajar daring bukan hanya sekadar kegiatan teknis dalam penerapan algoritma machine learning, tetapi juga merupakan bentuk penerapan akal dan ilmu untuk kemaslahatan umat. Upaya memahami perilaku pengguna dan meningkatkan retensi siswa mencerminkan tanggung jawab ilmiah yang sejalan dengan pesan ayat tersebut bahwa ilmu harus digunakan untuk menghasilkan manfaat yang nyata, bukan sekadar pengetahuan teoretis.

Hadis Nabi Muhammad ﷺ juga menegaskan keutamaan menuntut ilmu. Dalam riwayat Muslim (no. 2699) disebutkan:

وَمَنْ سَلَكَ طَرِيقًا يَلْتَمِسُ فِيهِ عِلْمًا سَهَّلَ اللَّهُ لَهُ بِهِ طَرِيقًا إِلَى الْجَنَّةِ

"Siapa yang menempuh jalan untuk mencari ilmu, maka Allah akan mudahkan baginya jalan menuju surga." (HR. Muslim, no. 2699)

Dalam konteks hadis ini, Allah menjanjikan kemudahan menuju surga bagi orang yang bersungguh-sungguh menempuh jalan ilmu karena ilmu akan menuntun seseorang pada amal saleh, menjauhkannya dari kebodohan, dan menumbuhkan keimanan yang benar. Imam Nawawi dalam Syarh Shahih Muslim menjelaskan

bahwa “jalan menuju surga” tidak hanya bermakna fisik, tetapi juga kemudahan spiritual dan intelektual untuk memahami kebenaran dan mengamalkannya.

Dengan demikian, penelitian ini tidak hanya bernilai akademik, tetapi juga mengandung nilai moral dan spiritual dalam upaya mendorong keberlanjutan proses belajar sebagai bagian dari amanah ilmu.

1.2 Pernyataan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah dalam penelitian ini dapat dijabarkan sebagai berikut:

1. Bagaimana mengklasifikasikan status pengguna pada platform bimbingan belajar daring menggunakan algoritma *Random Forest*?
2. Bagaimana pengaruh variasi jumlah data terhadap performa model *Random Forest* berdasarkan metrik evaluasi accuracy, precision, recall, F1-score, dan ROC-AUC?

1.3 Batasan Masalah

1. Data yang digunakan dalam penelitian ini berasal dari *Tutoring Platform User Dataset* yang diunduh melalui situs Kaggle. Dataset ini berisi informasi perilaku dan interaksi pengguna terhadap layanan platform bimbingan belajar daring.
2. Penelitian ini hanya menggunakan algoritma *Random Forest Classifier* untuk proses klasifikasi status pengguna tanpa melakukan pembandingan dengan algoritma lain.

3. Fitur yang digunakan difokuskan pada seluruh fitur numerik serta dua fitur kategorikal yang relevan terhadap perilaku pengguna, yaitu Tutoring dan Subscription_Tier.

1.4 Tujuan Penelitian

1. Membangun model klasifikasi untuk menentukan status pengguna pada platform bimbingan belajar daring menggunakan algoritma *Random Forest*.
2. Menganalisis pengaruh variasi jumlah data terhadap performa model *Random Forest* berdasarkan metrik accuracy, precision, recall, F1-score, dan ROC-AUC.

1.5 Manfaat Penelitian

1. Manfaat Akademik:

   Menambah khazanah penelitian di bidang *machine learning*, khususnya penerapan algoritma *Random Forest* dalam klasifikasi status pengguna pada sektor pendidikan digital.
2. Manfaat Praktis:

   Menjadi referensi bagi penyedia platform bimbingan belajar daring untuk memahami perilaku pengguna dan merancang strategi retensi berbasis data.
3. Manfaat Spiritual dan Sosial:

   Mendorong kesadaran pentingnya konsistensi dalam menuntut ilmu sebagaimana dianjurkan dalam Islam, serta mendukung penerapan teknologi sebagai sarana meningkatkan mutu pendidikan.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian yang dilakukan Yuliani & Kartikasari (2025) mengenai algoritma *Random Forest* untuk klasifikasi pelanggan *churn* pada kegiatan servis dan penjualan sparepart di industri otomotif. Data yang digunakan merupakan data transaksi pelanggan yang menggambarkan frekuensi servis, pembelian suku cadang, serta durasi keterlibatan pelanggan dengan bengkel. Tujuan penelitian ini adalah untuk membantu pihak perusahaan dalam mengidentifikasi pelanggan yang berpotensi tidak kembali melakukan servis *(churner)*, sehingga dapat dirancang strategi retensi yang lebih efektif. Model dikembangkan dengan pendekatan *supervised learning* dan dievaluasi menggunakan *confusion matrix* serta nilai AUC. Secara keseluruhan, penelitian ini menunjukkan efektivitas *Random Forest* dalam konteks data pelanggan layanan purna jual, yang umumnya memiliki pola perilaku kompleks dan ketidakseimbangan kelas (Yuliani & Kartikasari, 2025).

Selanjutnya Penelitian Azmi & Voutama (2024) berfokus pada prediksi *churn* nasabah bank menggunakan algoritma *Random Forest* dan *Decision tree* dengan evaluasi berbasis *confusion matrix*. Data yang digunakan bersumber dari *Bank Customer Churn Dataset* di Kaggle, yang mencakup variabel perilaku seperti saldo, lama menjadi nasabah, jumlah produk bank yang digunakan, dan status keanggotaan aktif. Penelitian ini bertujuan untuk membandingkan performa kedua algoritma dalam mengidentifikasi nasabah yang berpotensi meninggalkan layanan perbankan. Hasil evaluasi menunjukkan bahwa *Random Forest* memiliki performa

lebih unggul dibandingkan *Decision Tree*. Temuan ini mengindikasikan bahwa pendekatan ansambel yang digunakan pada *Random Forest* dapat menangkap hubungan *non-linear* antar fitur dengan lebih baik dan memberikan hasil klasifikasi yang lebih stabil. Penelitian ini menegaskan bahwa *Random Forest* dapat menjadi model yang andal dalam memprediksi *churn* di sektor keuangan yang memiliki variabel perilaku kompleks (Azmi & Voutama, 2024).

Penelitian yang dilakukan Wakhidah, Zyen, Wahono (2025) mengkaji klasifikasi *churn* pelanggan telekomunikasi dengan menerapkan teknik SMOTE (*Synthetic Minority Oversampling Technique*) untuk mengatasi ketidakseimbangan data. Dua algoritma utama yang diuji adalah *Random Forest* dan XGBoost, dengan tujuan membandingkan kinerja keduanya dalam mengenali pelanggan *churn* dan *non-churn*. Dataset yang digunakan terdiri atas atribut seperti jenis kontrak, total tagihan, dan lama berlangganan. Hasil eksperimen menunjukkan bahwa penerapan SMOTE secara signifikan meningkatkan kemampuan model dalam mengenali kelas minoritas. Algoritma *Random Forest* menghasilkan akurasi sedikit lebih rendah daripada algoritma *XGBoost*. Kedua model menunjukkan stabilitas yang baik dan mampu memberikan interpretasi fitur penting, seperti *Contract, TotalCharges*, dan *tenure* sebagai variabel paling berpengaruh terhadap *churn*. Penelitian ini menegaskan pentingnya teknik penyeimbangan data dalam meningkatkan performa model klasifikasi serta memperkuat peran *Random Forest* sebagai algoritma yang efisien dalam konteks prediksi *churn* pelanggan telekomunikasi (Wakhidah et al., 2025).

Dalam penelitian Marcellina & Mukhlason (2024) membahas analisis prediktif *churn* untuk meningkatkan tingkat retensi pelanggan pada perusahaan *Software as a Service* (SaaS) dengan menerapkan metode SMOTE dan algoritma *machine learning* berbasis ansambel, yaitu *XGBoost* dan *Random Forest*. Data yang digunakan mencakup aktivitas pelanggan dalam sistem SaaS, seperti frekuensi tiket bantuan, interaksi dengan sistem ERP atau POS, serta jenis bisnis pengguna. Penelitian ini bertujuan mengembangkan model yang mampu mendeteksi pelanggan yang berpotensi *churn*, sekaligus mengidentifikasi faktor-faktor utama yang memengaruhi perilaku tersebut. Hasil pengujian menunjukkan bahwa model *XGBoost* dengan data yang telah diseimbangkan melalui SMOTE memberikan hasil terbaik. Sementara itu, *Random Forest* memberikan hasil yang kompetitif meskipun sedikit lebih rendah. Faktor-faktor dominan yang berkontribusi terhadap *churn* meliputi frekuensi keluhan, durasi penggunaan layanan, serta jenis industri pengguna. Kesimpulan dari penelitian ini adalah bahwa penerapan SMOTE dan algoritma ansambel dapat secara signifikan meningkatkan sensitivitas model terhadap kelas *churn* tanpa menurunkan akurasi secara keseluruhan (Marcellina & Mukhlason, 2024).

Pada penelitian Pamina et al., (2019) ini bertujuan membangun model klasifikasi *churn* pelanggan telekomunikasi yang efektif dengan membandingkan beberapa algoritma, termasuk *K-Nearest Neighbors* (KNN), *Random Forest*, dan XGBoost, menggunakan data dari IBM Watson. Studi ini menekankan pada analisis karakteristik pelanggan untuk memprediksi kemungkinan *churn* serta mengevaluasi kinerja model menggunakan metrik akurasi, *recall*, dan AUC. Hasil

eksperimen menunjukkan bahwa model *XGBoost* memiliki kinerja terbaik dengan nilai akurasi dan AUC tertinggi, diikuti oleh *Random Forest* sebagai algoritma yang paling stabil dalam menangani variasi data. Penelitian ini juga menemukan bahwa pelanggan dengan layanan fiber optic dan biaya bulanan tinggi memiliki kecenderungan lebih besar untuk *churn*, menjadikan variabel tersebut sebagai indikator penting dalam proses klasifikasi. Kesimpulannya, penelitian ini memperkuat peran algoritma ansambel seperti *Random Forest* dan *XGBoost* dalam memecahkan permasalahan *churn* pada industri telekomunikasi yang memiliki data berskala besar dan multivariat (Pamina et al., 2019).

Terakhir, Penelitian yang dilakukan oleh Holilurrahman dan Imron (2025) mengenai implementasi model prediksi *churn* pelanggan menggunakan algoritma *Random Forest* pada website industri telekomunikasi, peneliti memanfaatkan dataset pelanggan *Internet Service Provider* (ISP) yang berjumlah 72.274 data dengan 11 variabel utama. Tujuan penelitian ini adalah untuk membantu perusahaan telekomunikasi mengidentifikasi pelanggan yang berpotensi melakukan *churn* sehingga dapat dilakukan strategi retensi yang tepat. Algoritma yang digunakan adalah *Random Forest* yang dibandingkan dengan *Decision Tree*. Hasil penelitian menunjukkan bahwa *Random Forest* mampu mencapai akurasi sebesar 95%, sedangkan *Decision tree* hanya mencapai 92%. Perbedaan performa ini menegaskan bahwa *Random Forest* lebih unggul dalam mengatasi kompleksitas data dan menghasilkan prediksi yang lebih stabil. Selain itu, penelitian ini juga memperlihatkan bagaimana model dapat diintegrasikan ke dalam website industri telekomunikasi, sehingga tidak hanya sebatas eksperimen akademik tetapi juga

memiliki penerapan praktis. Dengan demikian, penelitian ini memberikan bukti empiris bahwa *Random Forest* efektif diterapkan dalam konteks bisnis telekomunikasi dan mampu memberikan kontribusi nyata bagi perusahaan dalam menghadapi tantangan *churn* pelanggan (Imron, 2025).

Dari beberapa studi yang telah diuraikan sebelumnya, penulis menyusun sebuah tabel untuk membandingkan setiap studi tersebut, yang dapat ditemukan pada Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No.	Judul & Nama Peneliti	Metode	Hasil Penelitian	Perbedaan
1.	Klasifikasi Pelanggan *Churn* pada Kegiatan Servis dan Penjualan Sparepart dengan Metode *Random Forest* Yuliani & Kartikasari (2025)	*Random Forest*	Akurasi 67,71%; presisi 72,93%; *recall* 85,16%; F1 78,6%; AUC 0,6798; RF efektif untuk identifikasi pelanggan berisiko pada konteks ritel otomotif.	Penelitian terdahulu menggunakan data pelanggan ritel otomotif, sedangkan penelitian ini menggunakan data pengguna platform bimbingan belajar daring dengan penerapan *SMOTE* dan variasi jumlah data.
2.	Prediksi *Churn* Nasabah Bank Menggunakan Klasifikasi *Random Forest* dan *Decision tree* dengan Evaluasi *Confusion matrix* Azmi & Voutama (2024)	*Random Forest* & *Decision Tree*	*Random Forest* unggul dengan akurasi 78%, presisi 79%, *recall* 78%, dan *F1-score* 78%	Penelitian terdahulu menggunakan data telekomunikasi dan membandingkan dua algoritma, sedangkan penelitian ini berfokus pada analisis *feature importance* pada platform bimbingan belajar daring.
3.	*Evaluation of Telecommunication Customer Churn Classification with SMOTE Using Random Forest and XGBoost Algorithms* Wakhidah, Zyen, & Wahono (2025)	*Random Forest* & *XGBoost*	*XGBoost* unggul dengan akurasi 82%, *recall* 84%, F1 83%, AUC 0,91 dan fitur penting yang didapat yaitu Contract, TotalCharges, tenure.	Penelitian terdahulu menggunakan data telekomunikasi dan membandingkan dua algoritma, sedangkan penelitian ini berfokus pada analisis *feature importance* pada platform bimbingan belajar daring.

4.	Analisis Prediktif *Churn* untuk Meningkatkan Tingkat Retensi Pelanggan pada Perusahaan SaaS Menggunakan *Machine learning* Marcellina & Mukhlason (2024)	*Random Forest* & *XGBoost*	Model terbaik *XGBoost* (setelah SMOTE). Akurasi 0,71, presisi 0,27, *recall* 0,86, F1 0,41, AUC 0,596; faktor dominan: frekuensi keluhan/tiket, isu ERP/POS, jenis usaha.	Penelitian terdahulu menggunakan data SaaS dan faktor layanan pelanggan, sedangkan penelitian ini menggunakan data aktivitas pembelajaran pengguna pada platform pendidikan digital.
5.	*An Effective Classifier for Predicting Churn in Telecommunication* Pamina et al. (2019)	KNN, *Random Forest*, *XGBoost*	*XGBoost* terbaik, pelanggan fiber optic dengan biaya bulanan tinggi cenderung *churn* (indikasi fitur paling berpengaruh).	Penelitian terdahulu menggunakan beberapa algoritma klasifikasi pada data pelanggan telekomunikasi untuk membandingkan performa model. Sementara itu, penelitian ini hanya menggunakan *Random Forest Classifier* dengan penerapan *SMOTE*, *feature selection*, dan variasi jumlah data.
6.	Implementasi Model Prediksi *Churn* Pelanggan Menggunakan Algoritma *Random Forest* Pada Website Industri Telekomunikasi Holilurrahman & Imron (2025)	*Random Forest* vs *Decision Tree*	*Random Forest*: 95% akurasi, *Decision Tree:* 92% akurasi. Signifikan *improvement*	Penelitian terdahulu berfokus pada perbandingan algoritma di bidang telekomunikasi, sedangkan penelitian ini menganalisis pengaruh jumlah data terhadap performa *Random Forest* pada platform pendidikan daring.

Fenomena *churn* atau berhentinya pengguna dalam memanfaatkan suatu layanan telah menjadi perhatian besar di berbagai sektor industri berbasis digital. Pada perusahaan telekomunikasi, perbankan, hingga *e-commerce, churn prediction* dipandang sangat penting karena biaya untuk memperoleh pelanggan baru jauh lebih besar dibandingkan mempertahankan pelanggan lama (Imani et al., 2025).

Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa algoritma *Random Forest* terbukti efektif dalam berbagai kasus prediksi *churn* pada berbagai sektor. Namun, sebagian besar penelitian masih menggunakan banyak fitur dan belum secara spesifik menguji performa model dengan jumlah fitur yang terbatas. Oleh karena itu, Penelitian ini menggunakan fitur-fitur hasil feature selection berdasarkan nilai *feature importance Random Forest* untuk memperoleh atribut yang paling berpengaruh terhadap klasifikasi *Renewal_Status*.

2.2 Platform Bimbingan Belajar Daring dalam Konteks *Churn* Pengguna

Platform bimbingan belajar daring merupakan transformasi digital dari layanan pendidikan *non-formal* yang memanfaatkan aplikasi berbasis internet untuk menyediakan materi pembelajaran, latihan soal, ujian simulasi, serta konsultasi dengan tutor secara daring (Assidiqi & Sumarni, 2020). Karakteristik utamanya adalah fleksibilitas dan aksesibilitas, di mana siswa dapat belajar kapan saja dan di mana saja. Banyak platform dilengkapi fitur video pembelajaran, bank soal, sistem poin, serta gamifikasi untuk meningkatkan motivasi belajar.

Interaktivitas dan personalisasi juga menjadi aspek penting dalam meningkatkan keterlibatan pengguna. Beberapa platform menyediakan kelas langsung, forum diskusi, dan rekomendasi materi yang disesuaikan dengan kemampuan siswa, sementara analitik pembelajaran digunakan untuk memantau progres dan aktivitas pengguna (Akpen et al., 2024). Sebagian besar platform menggunakan model bisnis *subscription-based*, di mana pengguna membayar biaya langganan untuk mengakses layanan penuh. Model ini menjadikan retensi

pengguna sebagai faktor penting dalam menjaga keberlangsungan bisnis (Maan & Maan, 2023).

Salah satu tantangan utama yang dihadapi adalah fenomena *churn*, yaitu kondisi ketika pengguna berhenti berlangganan atau tidak lagi aktif menggunakan aplikasi. Faktor penyebab *churn* dapat berasal dari pengguna, seperti motivasi belajar dan tujuan akademik, maupun dari platform, seperti kualitas konten dan pengalaman belajar (Aufa & Qoiriah, 2023).

Melihat kompleksitas penyebab *churn*, kemampuan untuk memprediksi *churn* secara akurat menjadi sangat penting. Dengan prediksi yang tepat, platform dapat merancang intervensi seperti promosi, konten tambahan, atau pendampingan khusus bagi pengguna berisiko tinggi (Aufa & Qoiriah, 2023). Pendekatan ini tidak hanya membantu menjaga retensi pelanggan, tetapi juga mendukung keberhasilan belajar siswa (Aisyah et al., 2025). Oleh karena itu, penelitian tentang prediksi *churn* pada platform bimbingan belajar daring memiliki nilai strategis baik dalam aspek bisnis maupun pendidikan digital.

2.3 Klasifikasi Status Pengguna

Klasifikasi merupakan salah satu teknik fundamental dalam *machine learning* yang termasuk dalam kategori *supervised learning*, di mana algoritma dilatih menggunakan data berlabel untuk memprediksi label dari data baru (Yan et al., 2017). Berbeda dengan *unsupervised learning*, pembelajaran terawasi memiliki *ground truth* sebagai acuan, sehingga algoritma dapat belajar melalui mekanisme umpan balik (Yan et al., 2017). Dalam penelitian ini, klasifikasi digunakan untuk

membedakan pengguna yang melakukan *churn* dan yang tetap bertahan pada platform bimbingan belajar daring.

Secara formal, klasifikasi bertujuan mempelajari fungsi pemetaan $f : X \rightarrow Y$, di mana X adalah ruang fitur dan Y adalah label kelas (*churn* dan *non-churn*). Model dilatih untuk menemukan *decision boundary* yang mampu memisahkan kelas secara optimal (Loog, 2017). Proses ini meliputi dua tahap, yaitu pelatihan model dengan data latih dan prediksi label pada data baru.

Dalam prediksi *churn*, klasifikasi biner menjadi formulasi paling relevan karena hanya terdapat dua kelas, *churn* (positif) dan *non-churn* (negatif). Algoritma tidak hanya menghasilkan label, tetapi juga probabilitas kontinu $P(Y=1|X)$ yang ditentukan melalui ambang batas tertentu. Tantangan utama dalam klasifikasi *churn* adalah ketidakseimbangan kelas, di mana jumlah *non-churner* jauh lebih besar dibanding *churner* (Ramadhon et al., 2024). Kondisi ini menyebabkan model cenderung bias terhadap kelas mayoritas, sehingga evaluasi model sebaiknya menggunakan metrik seperti *precision*, *recall*, *F1-score*, dan AUC yang lebih representatif daripada sekadar akurasi.

Evaluasi kinerja model dilakukan menggunakan *confusion matrix*, yang menggambarkan jumlah prediksi benar dan salah pada setiap kelas. Empat komponennya adalah *True positive* (TP), *True negative* (TN), *False positive* (FP), dan *False negative* (FN). Dalam konteks bisnis, keseimbangan antara FP dan FN penting karena FP menyebabkan pemborosan biaya retensi, sedangkan FN menyebabkan kehilangan pelanggan potensial. Metrik evaluasi lain seperti *precision* mengukur ketepatan prediksi *churn*, *recall* mengukur kemampuan model

mendeteksi *churner*, dan *F1-score* menyeimbangkan keduanya. Sementara itu, *Area Under Curve* (AUC) dari kurva ROC memberikan ukuran yang lebih stabil untuk membandingkan performa antar model (Maan & Maan, 2023).

Tahap penting lain dalam klasifikasi adalah data *preprocessing*, karena data riil sering mengandung *missing values*, duplikasi, dan *outlier*. Penanganan dapat dilakukan dengan penghapusan data, imputasi sederhana, atau metode lanjutan seperti *multiple imputation*. Selain itu, *feature engineering* dan *feature selection* berperan penting untuk meningkatkan performa model dan mencegah *overfitting* (Loog, 2017). Langkah *preprocessing* juga mencakup transformasi dan normalisasi data, seperti *min–max scaling* atau *z-score*, tergantung algoritma yang digunakan. Dengan kualitas data yang baik, model klasifikasi dapat menghasilkan prediksi *churn* yang lebih akurat dan bermanfaat bagi pengelola platform bimbingan belajar daring.

2.4 Decision Tree

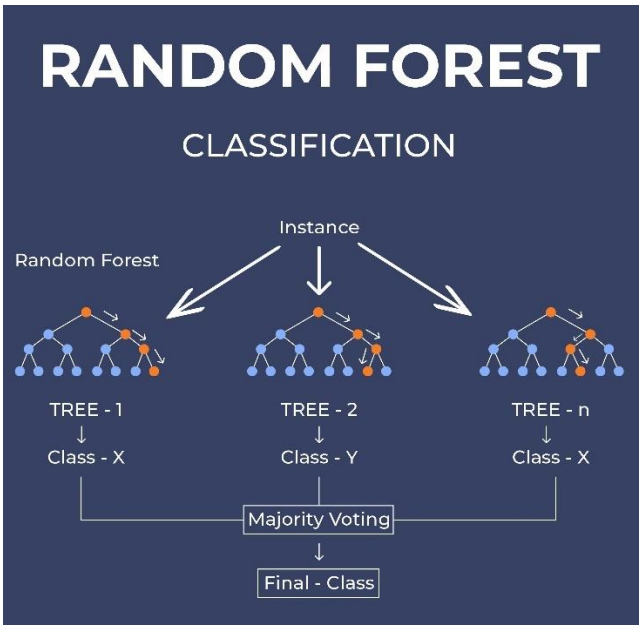
Decision tree merupakan suatu model prediktif yang memiliki struktur menyerupai pohon, di mana setiap node merepresentasikan suatu keputusan, dan setiap cabang menunjukkan kemungkinan konsekuensi dari keputusan tersebut. Model ini tergolong ke dalam metode pembelajaran terawasi *(supervised learning)*, yang berarti membutuhkan data latih *(training dataset)* untuk menggantikan peran pengalaman manusia sebelumnya dalam proses pengambilan keputusan (Ramadhon et al., 2024). Struktur pada pohon keputusan terdiri atas simpul akar *(root node),* simpul internal *(internal node),* dan simpul daun *(leaf node).* Secara visual, pohon keputusan menyerupai diagram alir *(flowchart)*, di mana setiap

simpul internal merepresentasikan kondisi atau kriteria pengujian, sedangkan simpul daun merepresentasikan hasil klasifikasi atau kelas dari suatu data. Penyusunan pohon keputusan dilakukan dengan pendekatan *divide and conquer*, yaitu membagi permasalahan menjadi sub-permasalahan yang lebih kecil secara berulang. Setiap jalur dari akar hingga daun dalam pohon tersebut menggambarkan suatu aturan klasifikasi yang digunakan untuk menentukan kelas suatu data (Kusumarini et al., 2021). Pembuatan model *Decision tree* dilakukan dengan cara membagi data ke dalam kelompok-kelompok yang lebih kecil berdasarkan atribut yang dimiliki oleh data tersebut. Proses pemisahan ini dilakukan secara berulang hingga seluruh data yang berasal dari kelas yang sama terkelompok dalam satu bagian yang seragam. Dalam struktur pohon keputusan, terdapat beberapa istilah penting, antara lain *decision node*, yaitu simpul yang merepresentasikan fitur atau atribut yang digunakan untuk mengambil keputusan. Simpul keputusan yang berada di posisi paling atas disebut sebagai *root node*. Selain itu, terdapat *leaf node* yang menunjukkan hasil akhir atau kelas dari suatu keputusan, serta *subtree* yang merupakan bagian atau cabang dari keseluruhan pohon keputusan (Ramadhon et al., 2024).

2.5 *Random Forest*

Random Forest merupakan pengembangan dari algoritma *Decision tree* yang termasuk dalam kategori *ensemble learning*, yaitu teknik pembelajaran yang menggabungkan beberapa model dasar untuk menghasilkan performa yang lebih stabil dan akurat. *Random Forest* membangun banyak pohon keputusan secara acak

dan menggabungkan hasilnya melalui mekanisme *voting* (untuk klasifikasi) atau rata-rata (untuk regresi).



Gambar 2. 1 Prinsip Kerja *Random Forest* (Sumber: www.miro.medium.com)

Gambar 2.2 memperlihatkan ilustrasi prinsip kerja *Random Forest*, di mana setiap pohon dibangun dari subset data dan subset fitur yang berbeda menggunakan metode *bootstrap sampling* dan random *feature selection*. Proses ini menghasilkan sekumpulan pohon yang independen satu sama lain, sehingga model akhir lebih tahan terhadap *noise* dan variasi data (Loog, 2017).

Keunggulan utama *Random Forest* adalah kemampuannya mengurangi *overfitting* dan meningkatkan stabilitas hasil prediksi. Model ini juga tahan terhadap *missing values* dan *outlier*, karena keputusan akhir dihasilkan dari gabungan banyak pohon yang saling menyeimbangkan (Dewi et al., 2019). Selain itu, *Random Forest* mampu menangani berbagai tipe data, baik numerik maupun kategorikal, serta mendeteksi hubungan *non-linear* antar variabel yang sering muncul pada perilaku

pengguna digital. Setiap pohon dalam *Random Forest* melakukan prediksi secara independen, kemudian seluruh hasil digabungkan untuk menentukan keputusan akhir melalui mekanisme *majority voting* (Aini et al., 2024).

Selain meningkatkan akurasi, *Random Forest* juga menyediakan *feature importance*, yaitu ukuran kontribusi relatif setiap fitur terhadap hasil klasifikasi, yang dihitung berdasarkan penurunan impurity (misalnya Gini Index) di seluruh pohon. Semakin besar pengaruh suatu fitur terhadap penurunan impurity, semakin tinggi kepentingannya dalam menentukan keputusan model. Informasi ini bermanfaat untuk memahami variabel mana yang paling berpengaruh terhadap keputusan *churn* (Aguilar-Ruiz, 2024).

Sebagai model ensemble, keputusan akhir pada *Random Forest* ditentukan melalui mekanisme *majority voting*, yaitu pemilihan kelas yang paling banyak diprediksi oleh seluruh pohon dalam model. Secara matematis, hasil prediksi akhir dapat dirumuskan sebagai berikut (Perdana & Farhana, 2022):

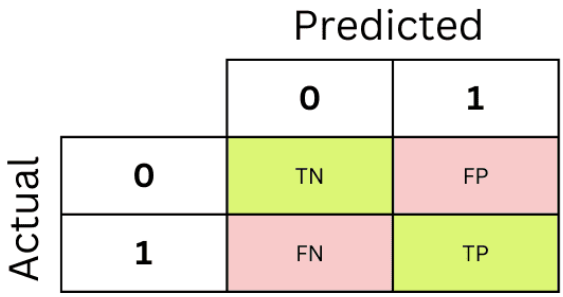
$$\hat{y} = mode(h_1(x), h_2(x), \dots, h_B(x)) \quad (2.\ 1)$$

Keterangan:
$\hat{y}_{RF}(x)$: hasil prediksi akhir *Random Forest* untuk data x.
$h_b(x)$: hasil prediksi dari pohon ke-b.
B: jumlah total pohon dalam model.

Rumus ini menunjukkan bahwa setiap pohon memberikan prediksi *independen*, kemudian model memilih kelas yang paling dominan sebagai keputusan akhir, sehingga menghasilkan prediksi yang lebih stabil dan akurat dibandingkan satu pohon keputusan tunggal.

2.6 *Confusion Matrix*

Confusion matrix adalah alat yang sering digunakan untuk mengevaluasi performa model klasifikasi, termasuk dalam penelitian ini yang menggunakan algoritma *Random Forest* untuk memprediksi status pengguna (*churn* dan *non-churn*) (Istiqamah & Rijal, 2024). Matriks ini menggambarkan hasil prediksi model dalam bentuk tabel, di mana setiap baris mewakili kelas sebenarnya, dan setiap kolom mewakili prediksi model.



Gambar 2. 2 *Confusion Matrix* (Sumber: www.kdnuggets.com)

Ada empat komponen utama dalam *confusion matrix*:

1. *True Negative* (TN): Jumlah data yang diprediksi sebagai *non-churn* dan memang benar *non-churn*.
2. *False Positive* (FP): Jumlah data yang diprediksi sebagai *churn*, tetapi sebenarnya bukan *churn (non-churn).*
3. *False Negative* (FN): Jumlah data yang diprediksi sebagai *non-churn*, tetapi sebenarnya *churn.*
4. *True Positive* (TP): Jumlah data yang diprediksi sebagai *churn* dan benar-benar merupakan *churn* (Istiqamah & Rijal, 2024).

Dalam konteks bisnis, khususnya pada platform bimbingan belajar daring, keseimbangan antara kesalahan prediksi FP dan FN menjadi penting. Kesalahan FP dapat menyebabkan pemborosan sumber daya karena sistem melakukan intervensi pada pengguna yang sebenarnya tidak berisiko *churn*, sedangkan kesalahan FN lebih berdampak serius karena pengguna yang sebenarnya berpotensi *churn* tidak terdeteksi dan berpotensi meninggalkan layanan. Selain digunakan untuk menggambarkan distribusi hasil prediksi, *confusion matrix* juga menjadi dasar dalam perhitungan berbagai metrik evaluasi model. Metrik-metrik tersebut dihitung berdasarkan nilai TP, TN, FP, dan FN, antara lain sebagai berikut (Ullah et al., 2019):

1. Akurasi *(Accuracy)*

   Seberapa banyak prediksi yang benar secara keseluruhan.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.\ 2)$$

2. Recall *(Sensitivity / True Positive Rate)*

   Seberapa banyak data positif yang berhasil ditemukan.

$$Recall = \frac{TP}{TP+FN} \quad (2.\ 3)$$

3. *Precision*

   Seberapa tepat prediksi positif yang dihasilkan model.

$$Precission = \frac{TP}{TP+FP} \quad (2.\ 4)$$

4. *F1-Score*

   keseimbangan antara *Precision* dan *Recall*.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.5)$$

Dengan menggunakan *confusion matrix*, performa model tidak hanya dinilai berdasarkan jumlah prediksi yang benar, tetapi juga kemampuan model dalam mengidentifikasi kelas *churn* secara tepat, yang sangat penting dalam pengambilan keputusan strategi retensi pengguna.

2.7 *K-Fold Cross Validation*

K-*Fold* Cross Validation merupakan salah satu metode evaluasi model yang digunakan untuk mengukur performa model terhadap data yang belum pernah dilihat selama proses pelatihan. Teknik ini bekerja dengan membagi dataset menjadi K bagian atau *Fold*. Dalam setiap iterasi, satu bagian akan digunakan sebagai data uji, sedangkan K-1 bagian lainnya digunakan sebagai data latih. Proses ini diulang sebanyak K kali hingga setiap bagian dataset pernah menjadi data uji satu kali (Hafid, 2023).

Keunggulan utama dari metode *K-Fold Cross Validation* adalah kemampuannya dalam meminimalkan variasi hasil evaluasi dan menghasilkan estimasi performa model yang lebih stabil dan akurat pada data baru. Pendekatan ini juga memastikan bahwa model tidak hanya memiliki performa baik pada subset tertentu, tetapi juga mampu melakukan generalisasi pada seluruh dataset (Azis et al., 2020). Secara umum, tahapan dalam *K-Fold Cross Validation* meliputi:

1. Membagi dataset menjadi K subset berukuran sama.
2. Melatih model sebanyak K kali, menggunakan K-1 subset sebagai data latih dan satu subset sebagai data uji.

3. Menghitung rata-rata hasil evaluasi dari seluruh iterasi untuk memperoleh performa keseluruhan model.

Hasil dari proses ini biasanya digunakan untuk menilai metrik seperti akurasi dan *F1-score*, serta menjadi dasar untuk membandingkan performa antar model klasifikasi (Azis et al., 2020).

Untuk memperoleh performa keseluruhan model, nilai evaluasi dari setiap *Fold* dihitung rata-ratanya. Secara matematis, nilai rata-rata *K-Fold* dapat dinyatakan sebagai berikut:

$$Accuracy_{avg} = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \quad (2.\ 6)$$

Keterangan:
$Accuracy_i$= nilai akurasi yang diperoleh pada *Fold* ke–i
K= jumlah total *Fold* dalam proses *K-Fold Cross Validation*
$Accuracy_{avg}$= nilai akurasi rata-rata dari seluruh *Fold*

Rumus tersebut menunjukkan bahwa performa akhir model merupakan rata-rata dari seluruh hasil evaluasi pada tiap iterasi, sehingga memberikan estimasi yang lebih stabil dan tidak bergantung pada satu pembagian data tertentu.

2.8 *Receiver Operating Characteristic* (ROC) dan *Area Under Curve* (AUC)

Receiver Operating Characteristic (ROC) merupakan kurva evaluasi kinerja model klasifikasi biner yang menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai nilai ambang batas (threshold) (Fawcett, 2006). ROC digunakan untuk menilai kemampuan model dalam membedakan dua kelas, dalam konteks ini antara *churn* dan *non-churn*. Setiap titik pada kurva ROC merepresentasikan pasangan nilai TPR dan FPR

untuk suatu nilai *threshold* tertentu yang digunakan untuk mengubah probabilitas prediksi menjadi label kelas.

True Positive Rate (juga disebut *Recall* atau *Sensitivity*) mengukur kemampuan model dalam mendeteksi kelas positif, sedangkan *False Positive Rate* mengukur proporsi kelas negatif yang salah diprediksi sebagai positif. Secara matematis dapat dirumuskan sebagai berikut (Han et al., 2011):

$$TPR = \frac{TP}{TP+FN} \quad (2.\ 7)$$

$$FPR = \frac{FP}{FP+TN} \quad (2.\ 8)$$

Kurva ROC memberikan gambaran visual mengenai *trade-off* antara TPR dan FPR. Model yang baik akan menghasilkan kurva yang mendekati sudut kiri atas grafik, yang berarti TPR tinggi dan FPR rendah.

Area Under Curve (AUC) merupakan ukuran numerik dari luas area di bawah kurva ROC. Nilai AUC berada pada rentang 0 hingga 1 dan digunakan untuk menilai kualitas model secara umum tanpa bergantung pada *threshold* tertentu. Semakin besar nilai AUC, semakin baik kemampuan model dalam membedakan kedua kelas (Bradley, 1997). Interpretasi umum nilai AUC adalah:

AUC = 0,90–1,00 → sangat baik

AUC = 0,80–0,89 → baik

AUC = 0,70–0,79 → cukup

AUC = 0,50–0,69 → lemah (mendekati tebakan acak)

AUC = 0,50 menunjukkan bahwa model tidak memiliki kemampuan klasifikasi dan hanya setara dengan prediksi acak. Sebaliknya, AUC mendekati 1 menunjukkan

bahwa model mampu membedakan *churn* dan *non-churn* secara konsisten pada berbagai nilai threshold.

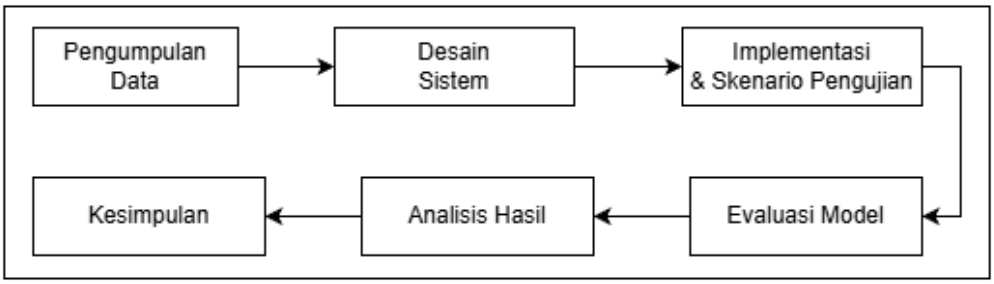
Dalam penelitian ini, ROC dan AUC digunakan untuk mengevaluasi performa model *Random Forest* pada kasus *imbalanced classification*, karena AUC lebih stabil dan tidak terpengaruh distribusi kelas dibandingkan metrik seperti akurasi. Oleh karena itu, ROC–AUC menjadi indikator penting untuk menilai kemampuan generalisasi model dalam memprediksi risiko *churn* pada platform bimbingan belajar daring.

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan menerapkan teknik *machine learning* untuk membangun model klasifikasi status pada pengguna platform bimbingan belajar daring. Proses penelitian disusun secara sistematis mengikuti tahapan umum proyek *machine learning*, yang meliputi pengumpulan data, *preprocessing* data, implementasi model, evaluasi, dan analisis hasil.



Gambar 3. 1 Desain Penelitian

Pada gambar 3.1 tahapan penelitian dimulai dengan pengumpulan data menggunakan dataset publik *Tutoring Website Data* yang diperoleh melalui situs Kaggle (Keats, n.d.). Dataset tersebut merepresentasikan perilaku pengguna, performa pembelajaran, serta status keanggotaan pada platform bimbingan belajar

daring. Variabel target yang digunakan adalah *Renewal_Status* yang terdiri atas kelas *churn* dan *non-churn*.

Tahap selanjutnya adalah *preprocessing* data untuk mempersiapkan data sebelum proses pemodelan. Pada tahap ini dilakukan pengecekan struktur data,

transformasi tipe data, serta *encoding* pada fitur kategorikal yang digunakan. Penelitian ini memfokuskan penggunaan fitur pada seluruh fitur numerik serta dua fitur kategorikal yang relevan terhadap perilaku pengguna, yaitu Tutoring dan Subscription_Tier. Selanjutnya dilakukan *feature selection* menggunakan *feature importance Random Forest* untuk memperoleh fitur yang paling berpengaruh terhadap klasifikasi *Renewal_Status*.

Setelah *preprocessing* selesai, data dibagi menjadi data latih dan data uji menggunakan rasio 80:20 dengan metode *stratified sampling* agar distribusi kelas tetap terjaga. Untuk mengatasi ketidakseimbangan kelas antara *churn* dan *non-churn*, penelitian ini menerapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) pada data latih setelah proses pembagian data dilakukan. Penerapan SMOTE bertujuan agar model dapat mempelajari kedua kelas secara lebih seimbang tanpa menyebabkan data *leakage* pada data uji.

Tahap berikutnya adalah pembentukan model klasifikasi menggunakan algoritma *Random Forest Classifier* dari pustaka *scikit-learn*. Penelitian ini menggunakan beberapa skenario jumlah data, yaitu 1000 data, 2000 data, 5000 data, dan *full dataset* untuk menganalisis pengaruh ukuran data terhadap performa model. Selain itu, dilakukan *hyperparameter* tuning menggunakan *GridSearchCV*

untuk memperoleh kombinasi parameter terbaik yang mampu meningkatkan performa model klasifikasi.

Setelah model terbentuk, dilakukan evaluasi performa menggunakan *confusion matrix* dan beberapa metrik evaluasi, yaitu *accuracy, precision, recall, F1-score,* dan ROC-AUC. Penelitian ini juga menerapkan *Stratified K-Fold Cross Validation* untuk mengukur stabilitas model dan mengurangi risiko *overfitting*. Selain itu, dilakukan analisis *feature importance* dan visualisasi *decision tree* untuk mengetahui fitur yang paling berpengaruh terhadap hasil klasifikasi serta memahami proses pengambilan keputusan model *Random Forest*.

Tahap terakhir adalah analisis hasil pengujian untuk menafsirkan performa model pada setiap skenario yang digunakan. Hasil penelitian diharapkan dapat membantu pengelola platform bimbingan belajar daring dalam memahami perilaku pengguna serta merancang strategi retensi yang lebih efektif berdasarkan pola penggunaan layanan.

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang berasal dari *Tutoring Website Data*, sebuah dataset publik yang disediakan oleh Jesse Keats melalui platform Kaggle (Keats, n.d.). Dataset ini berisi log aktivitas pengguna yang disimulasikan pada sebuah platform bimbingan belajar daring, mencakup atribut perilaku seperti frekuensi login, tingkat penyelesaian kursus, nilai rata-rata, durasi langganan, serta status pembaruan langganan. Dataset ini dipilih karena struktur datanya lengkap dan relevan untuk analisis perilaku dalam konteks prediksi *churn*.

Dataset terdiri dari 14.843 data pengguna dengan 23 atribut *dependen* dan 1 atribut *independen* yang merepresentasikan aktivitas pengguna selama menggunakan platform bimbingan belajar daring. Variabel target utama dalam penelitian ini adalah *Renewal_Status*, yaitu status pembaruan langganan pengguna yang terdiri atas dua kelas, yaitu *churn* (1) dan *non-churn* (0). Berdasarkan distribusi data, terdapat 11.874 data *non-churn* dan 2.969 data *churn*, sehingga dataset menunjukkan kondisi ketidakseimbangan kelas *(imbalanced data)*. Penjelasan lengkap mengenai atribut dataset ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Penjelasan Atribut Dataset

No.	Atribut	Keterangan	Tipe Data
1.	User_ID	Identitas unik pengguna	Integer/String
2.	Age_in_Months	Usia pengguna dalam bulan	Integer
3.	Gender	Jenis kelamin pengguna	String (Male/Female)
4.	Location	Lokasi atau wilayah tempat pengguna berasal	String
5.	Grade	Tingkat pendidikan atau kelas pengguna	String
6.	Logins_per_Month	Jumlah *login* pengguna dalam satu bulan	Integer
7.	Days_Completed_Activity	Jumlah hari aktif pengguna dalam kegiatan belajar	Integer
8.	Exercises_Started	Jumlah latihan yang dimulai pengguna	Integer
9.	Exercises_Completed	Jumlah latihan yang berhasil diselesaikan	Integer
10.	Total_Time_Spent_in_Minutes	Total waktu belajar pengguna dalam menit	Float
11.	Completion_Rate	Persentase penyelesaian materi pembelajaran	Float
12.	Average_Score	Nilai rata-rata hasil latihan atau ujian	Float
13.	Course_Rating	Penilaian pengguna terhadap kursus yang diikuti	Float
14.	Subscription_Tier	Jenis langganan pengguna (*Basic, Premium*, dsb.)	String
15.	Subscription_Cost	Biaya langganan bulanan	Float
16.	Subscription_Length_in_Months	Lama masa berlangganan (bulan)	Integer
17.	Tutoring	Status penggunaan layanan tutor pribadi	Boolean/String
18.	Referrals	Jumlah pengguna yang direferensikan	Integer

19.	Recommendation_Likelihood	Probabilitas pengguna merekomendasikan platform	Float
20.	Points_Earned	Jumlah poin yang diperoleh dari aktivitas belajar	Integer
21.	Academic_Grade	Nilai akademik pengguna secara keseluruhan	Float
22.	Engagement_Score	Skor gabungan keterlibatan pengguna	Float
23.	Activity_Index	Indeks aktivitas pengguna secara keseluruhan	Float
24.	Renewal_Status	Status *churn* pengguna (1 = *churn*, 0 = *non-churn*)	Integer

Pada tahap awal penelitian, fitur yang digunakan difokuskan pada seluruh fitur numerik serta dua fitur kategorikal yang relevan terhadap perilaku pengguna, yaitu Tutoring dan Subscription_Tier. Pemilihan fitur awal dilakukan berdasarkan relevansi atribut terhadap aktivitas belajar, keterlibatan pengguna, dan status layanan pengguna pada platform bimbingan belajar daring. Daftar fitur awal yang digunakan dalam proses *feature selection* ditunjukkan pada Tabel 3.2.

Tabel 3. 2 Fitur Penelitian untuk Pemodelan *Random Forest*

No.	**Nama Fitur**	**Jenis Data**	**Kegunaan dalam Prediksi *Churn***
1.	Age_in_Month	Numerik	Menggambarkan loyalitas pengguna terhadap platform.
2.	Login_per_Month	Numerik	Mengukur frekuensi akses pengguna ke platform.
3.	Days_Completed_Activity	Numerik	Menunjukkan tingkat keaktifan pengguna.
4.	Exercises_Started	Numerik	Menggambarkan partisipasi pengguna dalam belajar.
5.	Total_Time_Spent_in_Minutes	Numerik	Menunjukkan tingkat aktivitas pengguna pada platform.
6.	Completion_Rate	Numerik	Mengukur tingkat penyelesaian aktivitas belajar.
7.	Average_Score	Numerik	Merepresentasikan performa akademik pengguna.
8.	Course_Rating	Numerik	Menggambarkan kepuasan pengguna terhadap layanan.
9.	Recommendation_Likelihood	Numerik	Menunjukkan tingkat kepuasan dan loyalitas pengguna.
10.	Exercises_Completed	Numerik	Mengukur konsistensi pengguna dalam belajar.
11.	Points_Earned	Numerik	Merepresentasikan tingkat keterlibatan pengguna.
12.	Subscription_Cost	Numerik	Menggambarkan nilai layanan yang digunakan pengguna.

13.	Subscription_Length_in_Month	Numerik	Menunjukkan lama pengguna berlangganan.
14.	Referrals	Numerik	Menggambarkan loyalitas dan rekomendasi pengguna.
15.	Tutoring	Kategori	Menunjukkan penggunaan layanan tutor oleh pengguna.
16.	Subscription_Tier	Kategori	Menunjukkan jenis layanan langganan pengguna.

Random Forest untuk memperoleh fitur yang paling berpengaruh terhadap klasifikasi Renewal_Status. Fitur dengan nilai *importance* tertinggi kemudian digunakan pada proses implementasi algoritma *Random Forest* dan pengujian beberapa skenario jumlah data, yaitu 1000 data, 2000 data, 5000 data, dan full dataset. Daftar fitur hasil *feature selection* yang digunakan pada model akhir ditunjukkan pada Tabel 3.3.

Tabel 3. 3 fitur hasil *feature selection*

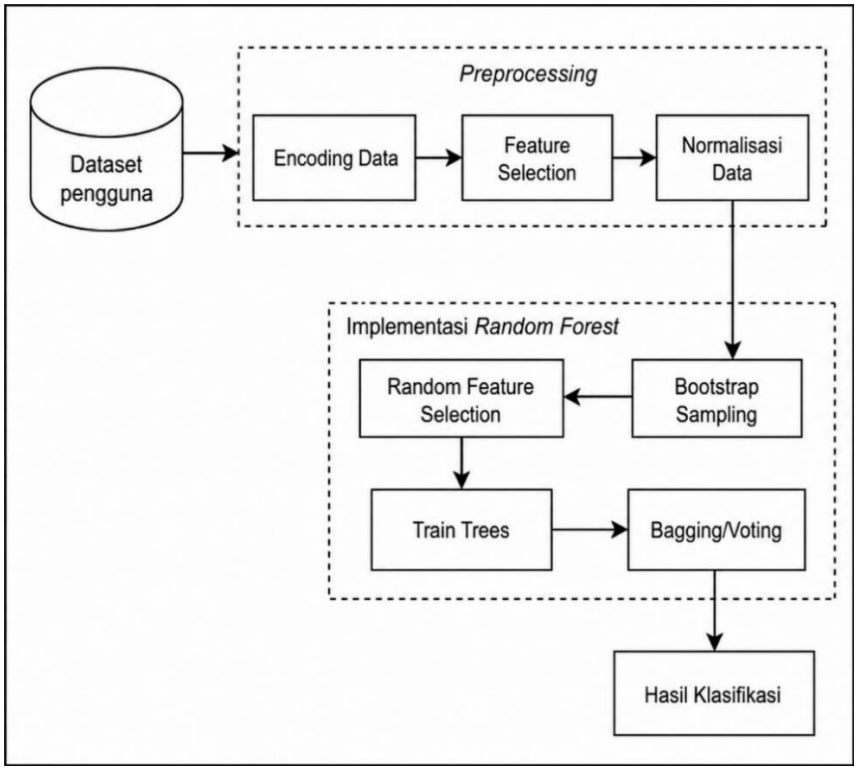
No.	**Nama Fitur**	**Jenis Data**	**Kegunaan dalam Prediksi *Churn***
1.	Average_Score	Numerik	Merepresentasikan performa akademik pengguna.
2.	Points_Earned	Numerik	Menggambarkan tingkat keterlibatan pengguna.
3.	Completion_Rate	Numerik	Mengukur tingkat penyelesaian aktivitas belajar.
4.	Exercises_Started	Numerik	Menunjukkan partisipasi pengguna dalam latihan belajar.
5.	Total_Time_Spent_in_Minutes	Numerik	Menggambarkan tingkat aktivitas pengguna pada platform.
6.	Age_in_Months	Numerik	Menunjukkan lama pengguna menggunakan platform.
7.	Exercises_Completed	Numerik	Mengukur konsistensi pengguna dalam menyelesaikan latihan.

Fitur-fitur yang digunakan dalam penelitian ini merepresentasikan berbagai aspek perilaku pengguna, tingkat keterlibatan dalam proses pembelajaran, serta status layanan pengguna pada platform bimbingan belajar daring. Kombinasi fitur

tersebut diharapkan mampu membantu model *Random Forest* dalam mengenali pola pengguna yang berpotensi melakukan *churn* maupun tetap aktif menggunakan layanan. Dengan pemilihan fitur yang relevan melalui *feature selection*, proses klasifikasi diharapkan menjadi lebih efektif, stabil, dan mampu menghasilkan performa prediksi yang lebih optimal.

3.3 Desain Sistem

Desain sistem penelitian ini menggambarkan alur proses klasifikasi *churn* pada pengguna platform bimbingan belajar daring menggunakan algoritma *Random Forest Classifier*. Tujuan dari desain ini adalah menjelaskan hubungan antar komponen mulai dari tahap persiapan data hingga analisis hasil akhir model. Secara umum, proses penelitian terdiri atas tahapan *preprocessing* data, *feature selection*, pembagian data, penyeimbangan data, pemodelan *Random Forest*, *hyperparameter* tuning, evaluasi model, dan analisis hasil penelitian sebagaimana ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Desain Sistem

Pada gambar 3.2 tahapan penelitian diawali dengan penggunaan dataset yang berisi data perilaku pengguna pada platform bimbingan belajar daring. Selanjutnya dilakukan *preprocessing* data yang meliputi pengecekan struktur data, transformasi tipe data, serta *encoding* pada fitur kategorikal yang digunakan dalam penelitian. Pada tahap awal, penelitian menggunakan seluruh fitur numerik serta dua fitur kategorikal, yaitu Tutoring dan Subscription_Tier sebagai fitur awal penelitian.

Setelah *preprocessing* selesai, dilakukan *feature selection* menggunakan *feature importance Random Forest* untuk memperoleh fitur yang paling berpengaruh terhadap klasifikasi Renewal_Status. Fitur hasil *feature selection* kemudian digunakan pada proses implementasi model dan pengujian skenario penelitian.

Tahap berikutnya adalah pembagian data menjadi data latih dan data uji menggunakan metode *stratified sampling* dengan rasio 80:20 agar distribusi kelas *churn* dan *non-churn* tetap seimbang pada kedua dataset. Selanjutnya diterapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) pada data latih untuk mengatasi ketidakseimbangan kelas. Penerapan SMOTE bertujuan agar model dapat mempelajari kedua kelas secara lebih optimal tanpa menyebabkan data leakage pada data uji.

Data latih dan data uji kemudian dinormalisasi menggunakan metode *MinMax Scaling* untuk menyamakan skala antar fitur sehingga proses pembelajaran model menjadi lebih optimal. Setelah itu dilakukan proses pemodelan menggunakan algoritma *Random Forest Classifier* yang membangun beberapa *decision tree* melalui mekanisme *bootstrap sampling* dan *random feature selection*. Hasil prediksi dari setiap pohon keputusan kemudian digabungkan menggunakan metode *voting (bagging)* untuk menghasilkan prediksi akhir yang lebih stabil.

Untuk meningkatkan performa model, dilakukan *hyperparameter tuning* menggunakan metode *GridSearchCV* guna memperoleh kombinasi parameter terbaik. Parameter optimal yang diperoleh kemudian digunakan pada model akhir untuk melakukan proses klasifikasi terhadap data uji.

Model yang telah terbentuk selanjutnya dievaluasi menggunakan *confusion matrix* dan beberapa metrik evaluasi, yaitu *accuracy, precision, recall, F1-score,* dan ROC-AUC untuk mengukur performa model secara menyeluruh. Selain itu, penelitian ini juga menerapkan *Stratified K-Fold Cross Validation* untuk mengukur stabilitas model dan mengurangi risiko *overfitting*.

Tahap terakhir adalah analisis hasil penelitian untuk menafsirkan performa model pada setiap skenario jumlah data yang digunakan. Selain itu, dilakukan visualisasi *decision tree* dan analisis *feature importance* untuk memahami fitur yang paling berpengaruh dalam proses prediksi *churn* pengguna pada platform bimbingan belajar daring.

3.4 *Preprocessing Data*

Tahap *preprocessing data* dilakukan untuk mempersiapkan data sebelum proses pemodelan menggunakan algoritma *Random Forest Classifier*. Proses ini bertujuan agar data yang digunakan memiliki format yang sesuai, lebih terstruktur, serta dapat diproses dengan optimal oleh model machine learning. Pada penelitian ini, tahapan *preprocessing* meliputi *encoding data*, *feature selection*, dan normalisasi data menggunakan *MinMax Scaling*.

3.4.1 *Encoding data*

Tahap *encoding data* dilakukan pada fitur kategorikal yang digunakan dalam penelitian, yaitu Tutoring dan Subscription_Tier. Proses ini bertujuan untuk mengubah data kategorikal menjadi format numerik agar dapat diproses oleh algoritma *Random Forest*. Algoritma *machine learning* umumnya tidak dapat mengolah data berbentuk teks secara langsung sehingga diperlukan proses transformasi data kategorikal menjadi numerik.

Pada fitur Tutoring, proses *encoding* dilakukan dengan mengubah nilai *"Yes"* menjadi 1 dan *"No"* menjadi 0. Sementara itu, pada fitur Subscription_Tier, kategori *"Free"* diubah menjadi 0, *"Basic"* menjadi 1, dan *"Premium"* menjadi 2.

Proses *encoding* dilakukan sebelum tahap pemodelan agar seluruh fitur yang digunakan berada dalam format numerik dan dapat diproses secara optimal pada tahap klasifikasi menggunakan algoritma *Random Forest Classifier*.

3.4.2 *Feature Selection*

Tahap *feature selection* dilakukan untuk memilih fitur yang paling berpengaruh terhadap klasifikasi *Renewal_Status*. Pada tahap awal penelitian, digunakan seluruh fitur numerik serta dua fitur kategorikal, yaitu Tutoring dan Subscription_Tier sebagai fitur awal penelitian. Pemilihan fitur awal dilakukan berdasarkan relevansi atribut terhadap aktivitas belajar, tingkat keterlibatan pengguna, dan status layanan pengguna pada platform bimbingan belajar daring.

Selanjutnya dilakukan proses *feature selection* menggunakan *feature importance Random Forest*. Metode ini digunakan untuk mengukur tingkat pengaruh masing-masing fitur terhadap hasil klasifikasi yang dihasilkan model. Fitur dengan nilai *importance* tertinggi dianggap memiliki kontribusi paling besar dalam membantu model membedakan pengguna *churn* dan *non-churn*.

Pada penelitian ini, hasil *feature importance* digunakan sebagai dasar dalam menentukan fitur yang akan digunakan pada proses implementasi model *Random Forest* dan pengujian skenario penelitian. Dengan penerapan *feature selection*, model diharapkan dapat bekerja lebih optimal karena hanya menggunakan fitur-fitur yang memiliki kontribusi penting terhadap proses klasifikasi.

3.4.3 Normalisasi Data

Tahap normalisasi data dilakukan menggunakan metode *MinMax Scaling* untuk menyamakan rentang nilai antar fitur numerik sehingga proses pembelajaran model menjadi lebih optimal. Normalisasi diperlukan karena setiap fitur memiliki rentang nilai yang berbeda, sehingga perbedaan skala dapat mempengaruhi proses pembelajaran model dan penerapan teknik SMOTE yang berbasis perhitungan jarak.

Metode *MinMax Scaling* bekerja dengan mentransformasikan nilai data ke dalam rentang [0,1]. Proses normalisasi dilakukan menggunakan *MinMaxScaler()* dari pustaka *scikit-learn*.

Rumus normalisasi:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.\ 1)$$

Misalkan perhitungan normalisasi pada Logins_per_Month:

x = 12
min = 0
max = 30

$$x' = \frac{12}{30} = 0.40$$

Jika x = 25:

$$x' = \frac{25}{30} = 0.83$$

Proses normalisasi dilakukan pada data latih setelah tahap pemisahan data. Parameter hasil normalisasi kemudian diterapkan pada data uji untuk menjaga konsistensi skala dan mencegah kebocoran data (data *leakage*).

3.5 Pemisahan Data

Tahap pemisahan data dilakukan untuk membagi dataset menjadi data latih (*training set*) dan data uji (*testing set*) sebelum proses pemodelan menggunakan algoritma *Random Forest Classifier*. Pembagian data bertujuan agar model dapat dilatih menggunakan sebagian data dan kemudian diuji menggunakan data lain yang belum pernah dilihat sebelumnya sehingga performa model dapat dievaluasi secara objektif.

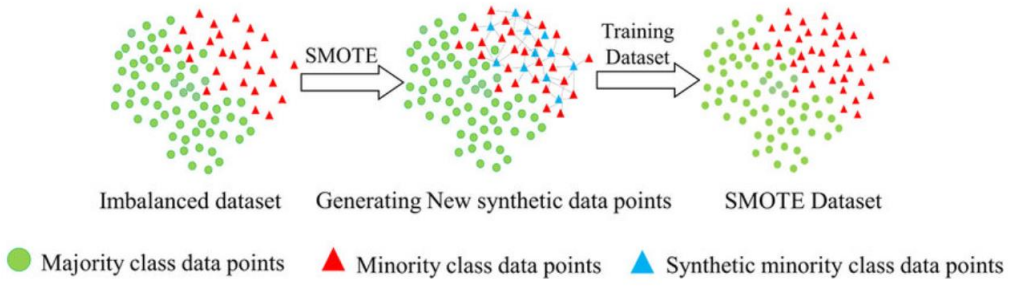
Pada penelitian ini, pembagian data dilakukan menggunakan metode *stratified sampling* dengan rasio 80:20. Metode *stratified sampling* digunakan untuk menjaga proporsi kelas *churn* dan *non-churn* tetap seimbang pada data latih maupun data uji. Penggunaan metode ini penting karena dataset memiliki distribusi kelas yang tidak seimbang (*imbalanced data*).

Dengan total 14.843 data, diperoleh sekitar 11.874 data sebagai data latih dan 2.969 data sebagai data uji. Proses pembagian data dilakukan menggunakan fungsi *train_test_split()* dari pustaka *scikit-learn* dengan parameter *stratify* berdasarkan variabel target Renewal_Status.

Pemisahan data dilakukan sebelum proses normalisasi dan penerapan *SMOTE* untuk mencegah terjadinya *data leakage*. Setelah data dibagi, proses normalisasi dan penyeimbangan data hanya diterapkan pada data latih, sedangkan data uji tetap digunakan dalam kondisi asli untuk memastikan proses evaluasi model dilakukan secara objektif dan merepresentasikan kondisi data sebenarnya.

3.6 *Synthetic Minority Oversampling Technique* (SMOTE)

Teknik *Synthetic Minority Oversampling Technique* (SMOTE) digunakan untuk mengatasi ketidakseimbangan data *(imbalanced data)* antara pengguna *churn* dan *non-churn* pada dataset. Pada penelitian ini, proporsi data menunjukkan bahwa pengguna *non-churn* jauh lebih banyak dibandingkan pengguna *churn*. Ketidakseimbangan ini dapat menurunkan kemampuan model dalam mengenali kelas minoritas *(churner)*, karena model cenderung mengikuti pola kelas mayoritas.



Gambar 3. 3 Ilustrasi Proses SMOTE pada Data Tidak Seimbang (Sumber: www.researchget.net)

Gambar 3.3 menunjukkan ilustrasi proses kerja *Synthetic Minority Over-sampling Technique* (SMOTE). Pada kondisi awal (*imbalanced dataset*), jumlah data pada kelas mayoritas (ditunjukkan oleh lingkaran berwarna hijau) lebih banyak dibandingkan kelas minoritas (ditunjukkan oleh segitiga berwarna merah). Ketidakseimbangan ini dapat menyebabkan model cenderung lebih mudah mempelajari pola dari kelas mayoritas dibandingkan kelas minoritas.

Melalui metode SMOTE, data sintetis baru dibangkitkan pada kelas minoritas berdasarkan interpolasi antara data minoritas yang berdekatan. Data sintetis tersebut ditunjukkan oleh simbol segitiga berwarna biru pada gambar.

Setelah proses pembangkitan data sintetis dilakukan, jumlah data pada kelas minoritas meningkat sehingga distribusi kedua kelas menjadi lebih seimbang. Dataset hasil SMOTE kemudian digunakan sebagai data pelatihan (*training dataset*) untuk membangun model klasifikasi yang lebih mampu mengenali pola pada kelas minoritas. Langkah-langkah penerapan metode SMOTE dalam penelitian ini adalah sebagai berikut:

a. Memilih secara acak satu sampel dari kelas minoritas (*churn*).
b. Menentukan beberapa tetangga terdekat (k) dari sampel minoritas yang dipilih.
c. Membentuk data sintetis baru dengan menggabungkan sampel minoritas dan salah satu tetangga terdekatnya menggunakan persamaan berikut (Handoko & Aditya, 2025):

$$x_{new} = x_i + \alpha\,(x_j - x_i) \qquad (3.\ 2)$$

Keterangan:
x_baru : sampel sintetis baru yang dihasilkan.
x_i : sampel asli dari kelas minoritas (*churn*).
x_j : salah satu tetangga terdekat dari x_i yang juga termasuk kelas minoritas.
α : bilangan acak antara 0 dan 1.

Misalkan distribusi kelas pada data latih sebagai berikut:

- *Non-churn*: 9.000
- *Churn*: 2.874

Jumlah sampel sintetis yang perlu ditambahkan:

$$9.000 - 2.874 = 6.126 \text{ data } \textit{churn} \text{ sintetis}$$

SMOTE menghasilkan data sintetis berdasarkan interpolasi titik-titik minoritas.

Contoh perhitungan satu sampel sintetis:

- xi = [0.2, 0.5, 0.7]
- xj = [0.3, 0.6, 0.8]

- α = 0.4

Kemudian dimasukkan rumus:

$$x_{\text{new}} = x_i + \alpha(x_j - x_i)$$

Hitungan:

- *Feature* 1: 0.24
- *Feature* 2: 0.54
- *Feature* 3: 0.74

Hasil sampel sintetis: [0.24, 0.54, 0.74]

Dengan pendekatan ini, SMOTE menghasilkan titik data baru yang berada di antara sampel minoritas yang sudah ada, sehingga model dapat belajar pola kelas *churn* dengan lebih baik. Penerapan SMOTE diterapkan secara manual hanya pada data latih setelah proses pemisahan data. SMOTE digunakan untuk menyeimbangkan distribusi kelas *churn* dan *non-churn* sebelum proses pelatihan model *Random Forest*. Data uji tidak dikenai proses SMOTE agar tetap merepresentasikan kondisi distribusi data asli dan menghindari terjadinya data *leakage*.

3.7 Algoritma *Random Forest*

Algoritma *Random Forest* merupakan metode *ensemble learning* yang membangun sejumlah pohon keputusan *(decision tree)* secara acak dan menggabungkan hasil prediksi masing-masing pohon untuk menghasilkan keputusan akhir melalui mekanisme *voting* mayoritas (Imron, 2025).

3.7.1 Tahapan Algoritma *Random Forest*

Algoritma *Random Forest* bekerja melalui tiga mekanisme utama, yaitu *bootstrap sampling*, random *feature selection*, dan *bagging (bootstrap aggregating)*

(Aufa & Qoiriah, 2023). Ketiga proses tersebut dilakukan secara otomatis oleh pustaka *scikit-learn* tanpa memerlukan pengaturan manual tambahan. Selain itu, pada setiap pohon yang terbentuk dilakukan proses pembentukan *decision tree* untuk menghasilkan aturan klasifikasi berdasarkan fitur-fitur yang digunakan dalam penelitian.

- ***Bootstrap sampling***

  Setiap pohon dilatih menggunakan subset data hasil pengambilan acak dengan pengembalian (*sampling with replacement*). Ukuran subset sama dengan jumlah data latih, tetapi beberapa sampel bisa terambil lebih dari satu kali. Probabilitas sebuah data tidak terambil sama sekali dalam proses pembentukan subset adalah:

$$P(\text{tidak terambil}) = (1 - \frac{1}{N})^N \approx e^{-1} \approx 0{,}368 \quad (3.\ 3)$$

  Keterangan:
  N: jumlah data pada dataset pelatihan.
  e: bilangan eksponensial (≈ 2,71828).

  Artinya, sekitar 36,8% data menjadi *out-of-bag* (OOB), sedangkan sisanya digunakan untuk melatih pohon. Data OOB berfungsi sebagai *validation set internal* yang berguna untuk memperkirakan kesalahan generalisasi model tanpa memerlukan data validasi tambahan.

- ***Random Feature selection***

Pada setiap pembentukan *node*, tidak semua fitur digunakan untuk mencari pemisahan terbaik. Sebagian kecil fitur dipilih secara acak untuk meningkatkan keragaman antar pohon. Jumlah fitur acak yang dipilih dihitung dengan rumus:

$$m = \sqrt{M} \tag{3. 4}$$

Keterangan:

- M: jumlah total fitur.
- m: jumlah fitur acak yang digunakan pada setiap split.

Dengan $M = 7$, maka $m = \sqrt{7} \approx 2{,}64$, sehingga setiap *node* mempertimbangkan sekitar 2–3 fitur acak pada setiap proses pemisahan data (*split*). Strategi ini membantu setiap pohon mempelajari pola data dari sudut pandang berbeda, menghasilkan model yang lebih kuat dan tidak bias terhadap satu fitur tertentu.

- **Pembentukan *Decision Tree***

Perhitungan ini dilakukan ntuk memperjelas cara kerja algoritma *Random Forest* dalam membentuk pohon keputusan, dilakukan ilustrasi perhitungan *impurity* pada salah satu *node* menggunakan *Gini Index*, dilakukan ilustrasi perhitungan manual pada salah satu *node* pohon keputusan. Nilai *impurity* dihitung dengan rumus :

$$Gini(S) = 1 - \sum_{k=1}^{K} p_k^2 \tag{3. 5}$$

Keterangan:

$Gini(S)$: nilai *impurity Node* induk.

p_k: proporsi data pada kelas ke-k.

K: jumlah kelas (dua pada kasus ini: *churn* dan *non-churn*).

Sebagai ilustrasi, misalkan pada suatu node terdapat 100 data yang terdiri dari 80 data *non-churn* dan 20 data *churn*. Dengan demikian:

$$p_1 = \frac{20}{100} = 0{,}2 \ , \ p_0 = \frac{80}{100} = 0{,}8$$

Sehingga:

$$Gini(S) = 1 - (0{,}8^2 + 0{,}2^2) = 0{,}32$$

Nilai *Gini* sebesar 0,32 menunjukkan bahwa *node* tersebut masih mengandung campuran dua kelas sehingga diperlukan proses pemisahan (*split*) untuk mengurangi *impurity* dan meningkatkan kemurnian data pada *node* berikutnya.

Pada algoritma *Random Forest*, proses perhitungan *impurity* dan penentuan split terbaik dilakukan secara otomatis pada setiap node pohon keputusan. Perhitungan manual pada penelitian ini hanya digunakan sebagai ilustrasi untuk menjelaskan mekanisme pembentukan *decision tree* yang menjadi dasar kerja *Random Forest*.

Setelah nilai *impurity* pada *node* induk diperoleh, algoritma kemudian mencari pemisahan (*split*) terbaik berdasarkan nilai penurunan *impurity* (*ΔGini*). Split yang menghasilkan nilai *ΔGini* terbesar akan dipilih untuk membentuk cabang baru pada *decision tree*.

$$\Delta Gini = Gini(S) - (\frac{n_L}{n} Gini(S_L) + \frac{n_R}{n} Gini(S_R)) \quad (3.\ 6)$$

Keterangan:

n_L, n_R: jumlah data pada cabang kiri dan kanan hasil pemisahan.
$Gini(S_L)$, $Gini(S_R)$: impurity masing-masing cabang.

Split yang menghasilkan nilai ΔGini terbesar dipilih karena mampu memberikan penurunan *impurity* paling tinggi dibandingkan kandidat *split* lainnya. Pada penelitian ini, perhitungan *ΔGini* tidak dilakukan secara manual untuk seluruh *node*, karena proses tersebut dilakukan secara otomatis oleh algoritma *Random Forest* pada pustaka *scikit-learn*. Perhitungan *Gini Index* yang ditampilkan hanya digunakan sebagai ilustrasi untuk menjelaskan mekanisme pembentukan *decision tree*.

- ***Bagging (bootstrap aggregating)***

  Setelah setiap pohon selesai dilatih, hasil prediksi digabungkan menggunakan prinsip *bagging*. Dalam kasus klasifikasi seperti penelitian ini, penggabungan dilakukan dengan *voting* mayoritas, di mana kelas yang paling banyak dipilih oleh seluruh pohon akan menjadi hasil akhir model. Proses ini secara matematis ditulis sebagai:

$$\hat{y}_{RF}(x) = \text{mode}\{h_b(x)\}, b = 1,2, \dots, B \qquad (3.\ 7)$$

Keterangan:
$\hat{y}_{RF}(x)$: prediksi akhir model *Random Forest* untuk data x.
$h_b(x)$: hasil prediksi pohon ke-b.
B: jumlah total pohon.

Selain hasil klasifikasi, model juga dapat menghitung probabilitas *churn* untuk tiap pengguna:

$$P(\ churn \mid x\ ) = \frac{\text{jumlah pohon yang memprediksi } churn}{B} \qquad (3.\ 8)$$

Keterangan:

$P(churn \mid x)$: probabilitas bahwa pengguna x tergolong *churn*.

Pembilang: jumlah pohon yang memprediksi kelas 1 (*churn*).

Penyebut: jumlah total pohon ($B = 300$).

3.7.2 Implementasi Algoritma *Random Forest*

Implementasi algoritma *Random Forest* dilakukan menggunakan bahasa pemrograman *Python* dengan pustaka *scikit-learn*. Dataset hasil *preprocessing* terdiri atas 14.843 data yang dibagi menjadi 80% data latih dan 20% data uji.

Pada tahap awal penelitian, digunakan seluruh fitur numerik serta dua fitur kategorikal, yaitu Tutoring dan Subscription_Tier sebagai fitur awal penelitian. Selanjutnya dilakukan *feature selection* menggunakan *feature importance Random Forest* untuk memperoleh fitur yang paling berpengaruh terhadap klasifikasi Renewal_Status. Fitur hasil *feature selection* kemudian digunakan pada proses implementasi model dan pengujian skenario penelitian.

Model *Random Forest Classifier* dibentuk menggunakan pustaka *scikit-learn* dengan proses pelatihan yang melibatkan pembentukan sejumlah *decision tree* melalui mekanisme *bootstrap sampling* dan *random feature selection*. Setiap pohon keputusan dilatih secara independen menggunakan data latih hasil penerapan SMOTE. Selanjutnya, hasil prediksi dari seluruh pohon digabungkan menggunakan metode *majority voting* untuk menghasilkan keputusan akhir model.

Algoritma *Random Forest Classifier* pada *scikit-learn* digunakan untuk membentuk model dengan parameter awal: *n_estimators* = [300, 500], *max_depth* = [5, 10, 15], *min_samples_split* = [2, 5], *min_samples_leaf* = [1, 2], *max_features* = 'sqrt', dan *random_state* = 42. Parameter-parameter tersebut memiliki arti sebagai berikut:

- *n_estimators*: jumlah pohon yang dibangun dalam hutan. Semakin banyak pohon, semakin stabil hasil prediksi.
- *max_depth*: kedalaman maksimum pohon; nilai 10 dipilih agar model tidak terlalu kompleks (*overfitting*).
- *min_samples_split* dan *min_samples_leaf*: mengontrol jumlah minimum sampel yang diperlukan untuk membentuk cabang baru, menjaga agar model tetap general.
- *max_features*='sqrt': menentukan jumlah fitur acak yang digunakan di setiap pemisahan node.
- *random_state*: memastikan hasil pelatihan dapat direproduksi.

Secara matematis, setiap pohon dalam *Random Forest* dilatih pada subset data hasil *bootstrap sampling*, lalu hasil prediksi seluruh pohon digabung menggunakan prinsip *bagging* (*bootstrap aggregating*) dengan mekanisme *voting mayoritas* (Perdana & Farhana, 2022):

$$\hat{y}_{RF}(x) = \text{mode}\{h_b(x)\}, b = 1,2, \dots, B \quad (3.\ 9)$$

Keterangan:
$\hat{y}_{RF}(x)$: hasil prediksi akhir *Random Forest* untuk data x.
$h_b(x)$: hasil prediksi dari pohon ke-b.
B: jumlah total pohon dalam model.

Dengan $h_b(x)$adalah hasil prediksi dari pohon ke-b, dan B jumlah total pohon. Selain menghasilkan label, model juga menghitung probabilitas *churn*:

$$P(churn \mid x) = \frac{\text{Jumlah pohon yang memprediksi } churn}{B} \quad (3.\ 10)$$

Rumus ini menunjukkan proporsi jumlah pohon yang memprediksi bahwa pengguna akan *churn* terhadap total jumlah pohon yang dibentuk.

3.8 *Tuning Parameter*

Tahap *tuning parameter* dilakukan untuk memperoleh kombinasi parameter terbaik (*optimal hyperparameter*) pada algoritma *Random Forest Classifier* sehingga model dapat menghasilkan performa klasifikasi yang lebih optimal dan stabil. Proses *tuning* dilakukan karena parameter bawaan (*default parameter*) belum tentu menghasilkan performa terbaik pada setiap dataset.

Optimasi dilakukan menggunakan metode *Grid Search Cross Validation (GridSearchCV)* dari pustaka *scikit-learn.* Metode ini menguji beberapa kombinasi parameter secara menyeluruh *(exhaustive search)* dan menilai setiap kombinasi dengan metrik akurasi rata-rata hasil validasi silang *(5-Fold cross validation).* Langkah-langkah *tuning* dijelaskan sebagai berikut:

1. **Menentukan ruang pencarian parameter**

   Proses *tuning* dilakukan menggunakan pendekatan *Grid Search CV* untuk menentukan kombinasi parameter terbaik berdasarkan hasil akurasi tertinggi. Parameter dan rentang nilai yang diuji disajikan pada Tabel 3.3.

Tabel 3. 4 Parameter dan Nilai Pengujian *Random Forest*

Parameter	**Nilai yang Diuji**	**Keterangan**
n_estimators	[300, 500]	Jumlah pohon pada hutan
max_depth	[5, 10, 15]	Kedalaman maksimum tiap pohon
min_samples_split	[2, 5]	Minimum sampel untuk membuat cabang baru
min_samples_leaf	[1, 2]	Minimum sampel pada daun
max_features	['sqrt']	Jumlah fitur acak tiap split
criterion	['gini']	Ukuran *impurity* yang digunakan

Pada Tabel 3.3 parameter *n_estimators* digunakan untuk menentukan jumlah pohon keputusan yang dibangun dalam model *Random Forest*. Parameter *max_depth* berfungsi mengontrol kedalaman pohon agar model tidak mengalami *overfitting*. Selanjutnya, *min_samples_split* dan *min_samples_leaf*

digunakan untuk mengatur jumlah minimum data pada proses pembentukan cabang dan daun pohon keputusan. Parameter *max_features* menentukan jumlah fitur acak yang digunakan pada setiap proses pemisahan *node*, sedangkan *criterion* digunakan untuk mengukur kualitas pemisahan berdasarkan nilai *Gini impurity*. Seluruh kombinasi nilai tersebut diuji menggunakan metode *Grid Search Cross Validation* untuk memperoleh konfigurasi parameter paling optimal yang memberikan hasil klasifikasi terbaik.

2. Evaluasi kombinasi Parameter

Setiap kombinasi parameter diuji melalui *GridSearchCV* dengan validasi silang 3 *fold*. Nilai skor rata-rata akurasi dari setiap kombinasi dihitung menggunakan persamaan (Handoko & Aditya, 2025):

$$Accuracy_{avg}(\theta) = \frac{1}{k} \sum_{i=1}^{k} Accuracy_i(\theta) \quad (3.\ 11)$$

Persamaan tersebut digunakan untuk menghitung nilai rata-rata akurasi hasil validasi silang *(cross validation)* sebanyak k kali. Dalam penelitian ini digunakan *3-Fold cross validation*, artinya data latih dibagi menjadi lima bagian *(Fold)*. Setiap *fold* secara bergantian digunakan sebagai data validasi, sementara empat *fold* lainnya digunakan untuk pelatihan model. Nilai akurasi dari kelima pengujian tersebut kemudian dirata-ratakan untuk mendapatkan performa keseluruhan model terhadap kombinasi parameter tertentu (θ).

3. Menentukan parameter terbaik

Kombinasi parameter dengan nilai akurasi rata-rata tertinggi dipilih sebagai konfigurasi optimal:

$$\theta^* = \text{argmax}\theta \in \Theta \; Accuracy_{avg}(\theta) \quad (3.\ 12)$$

Keterangan:

Θ = himpunan seluruh kombinasi parameter yang diuji.
θ^*= kombinasi terbaik hasil pencarian.

Persamaan tersebut digunakan untuk menentukan kombinasi parameter terbaik (θ*), yaitu kombinasi yang menghasilkan nilai rata-rata akurasi tertinggi dari seluruh pengujian. Dalam konteks *GridSearchCV*, setiap kombinasi parameter dalam himpunan Θ dievaluasi, kemudian sistem secara otomatis memilih konfigurasi dengan performa paling optimal untuk digunakan pada model final *Random Forest*.

3.9 Skenario Pengujian

Skenario pengujian pada penelitian ini dilakukan untuk menganalisis pengaruh jumlah data terhadap performa algoritma *Random Forest Classifier* dalam mengklasifikasikan Renewal_Status. Pengujian dilakukan menggunakan beberapa variasi jumlah data yang diambil dari dataset utama. Setiap skenario menggunakan tahapan *preprocessing*, pemodelan, dan evaluasi yang sama sehingga perbedaan hasil yang diperoleh dapat dianalisis berdasarkan jumlah data yang digunakan.

Tabel 3. 5 Tabel Skenario Pengujian

No	**Skenario**	**Jumlah Data**	**Teknik *Sampling***	***Preprocessing***	**Model**	**Tujuan Pengujian**
1	Skenario 1	1000	*Random Sampling*	*Encoding, Feature Selection, MinMax Scaling,* SMOTE	*Random Forest*	Menguji performa model pada jumlah data kecil

2	Skenario 2	2000	*Random Sampling*	*Encoding, Feature Selection, MinMax Scaling,* SMOTE	*Random Forest*	Melihat perubahan performa saat jumlah data ditingkatkan
3	Skenario 3	5000	*Random Sampling*	*Encoding, Feature Selection, MinMax Scaling,* SMOTE	*Random Forest*	Menguji stabilitas model pada jumlah data lebih besar
4	Skenario 4	*Full Data* (14.843)	Tanpa *sampling*	*Encoding, Feature Selection, MinMax Scaling,* SMOTE	*Random Forest*	Menguji performa model pada seluruh data

Pada tabel 3.4 dilakukan pemilihan variasi jumlah data pada setiap skenario untuk menganalisis pengaruh ukuran dataset terhadap performa model *Random Forest*. Jumlah data 1000, 2000, dan 5000 dipilih sebagai representasi dataset kecil, menengah, dan besar, sedangkan penggunaan *full dataset* dilakukan untuk melihat performa model pada keseluruhan data penelitian. Pendekatan ini digunakan untuk mengevaluasi apakah peningkatan jumlah data dapat membantu model menghasilkan performa klasifikasi yang lebih stabil dan optimal.

Pada setiap skenario, data dipilih sesuai jumlah data yang telah ditentukan kemudian dibagi menjadi data latih dan data uji menggunakan metode *stratified sampling* dengan rasio 80:20. Setelah proses pembagian data dilakukan, tahap normalisasi menggunakan *MinMax Scaling* diterapkan pada data latih dan data uji untuk menyamakan rentang nilai antar fitur.

Selanjutnya dilakukan penyeimbangan data menggunakan teknik *Synthetic Minority Over-sampling Technique (SMOTE)* pada data latih untuk mengatasi ketidakseimbangan kelas. Setelah itu, model *Random Forest Classifier* dilatih menggunakan parameter terbaik hasil proses *GridSearchCV*.

Untuk memastikan stabilitas model dan mengurangi risiko *overfitting*, penelitian ini menerapkan *Stratified K-Fold Cross Validation* sebanyak 3 *fold*. Model kemudian dievaluasi menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, *F1-score*, dan *ROC-AUC*.

Melalui skenario pengujian tersebut, penelitian ini dapat menganalisis pengaruh jumlah data terhadap kemampuan model *Random Forest* dalam melakukan klasifikasi pengguna *churn* dan *non-churn* pada platform bimbingan belajar daring.

3.10 Evaluasi Model

Evaluasi model bertujuan untuk menilai sejauh mana algoritma *Random Forest* mampu memprediksi status *churn* pengguna secara akurat. Melalui evaluasi ini dapat diketahui tingkat keberhasilan model dalam mengenali pola perilaku pengguna yang berhenti berlangganan (*churner*) maupun yang tetap menggunakan layanan (*non-churner*). Hasil evaluasi digunakan sebagai dasar untuk menilai efektivitas model serta menentukan aspek yang perlu diperbaiki pada tahap pengembangan berikutnya.

3.10.1 *Confusion matrix*

Evaluasi model dilakukan berdasarkan *confusion matrix*, yang menggambarkan perbandingan antara hasil prediksi model dengan kondisi aktual data uji. *Confusion matrix* memiliki empat komponen utama seperti yaitu:

- *True Positive* (TP) : Data positif yang diprediksi benar sebagai positif
- *True Negative* (TN) : Data negatif yang diprediksi benar sebagai negatif

- *False Positive* (FP) : Data negatif yang salah diprediksi sebagai positif
- *False Negative* (FN) : Data positif yang salah diprediksi sebagai negatif

Dengan menggunakan *confusion matrix*, beberapa metrik evaluasi seperti *Accuracy, Recall, Precision, dan F1-Score* dapat dihitung untuk mengukur kinerja model secara menyeluruh.. Pada penelitian ini, proporsi kelas pada data uji mempertahankan distribusi awal *(stratified split)*, sehingga tetap mencerminkan ketidakseimbangan kelas antara *churn* dan *non-churn*. Setelah penerapan SMOTE pada data latih, model menunjukkan peningkatan kemampuan mendeteksi kelas *churn* (kenaikan TP) disertai konsekuensi meningkatnya FP. Pola ini umum terjadi pada masalah *imbalanced classification*. *Recall* kelas minoritas meningkat, sementara *precision* dapat menurun.

Contoh perhitungan manual *confusion matrix*, misalkan setelah model dijalankan, diperoleh hasil sebagai berikut untuk 200 data uji, dengan hasil sebagai berikut:

1. TP = Sistem memprediksi *churn* dan data aktualnya juga *churn* sebanyak 30.
2. TN = Sistem memprediksi *non-churn* dan data aktualnya juga *non-churn* sebanyak 140.
3. FN = Sistem memprediksi *non-churn*, tetapi data aktualnya adalah *churn* sebanyak 10.
4. FP = Sistem memprediksi *churn*, tetapi data aktualnya adalah *non-churn* sebanyak 20.

Sehingga, *Confusion matrix* yang terbentuk adalah:

Tabel 3. 6 Contoh *Confusion Matrix*

Nilai Aktual / Nilai Prediksi	**Prediksi *Churn* (1)**	**Prediksi *Non-churn* (0)**

Aktual *Churn* (1)	TN = 140	FP = 20
Aktual *Non-churn* (0)	FN = 10	TP = 30

Dari tabel 3.5 dapat dihitung metrik evaluasi sebagai berikut (Ullah et al., 2019):

1. Akurasi *(Accuracy)*

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3.\ 13)$$

$$= \frac{(30 + 140)}{(30 + 140 + 20 + 10)}$$

$$= \frac{170}{200}$$

$$= 0{,}85\ (85\%)$$

2. *Recall (Sensitivity)*

$$Recall = \frac{TP}{(TP + FN)} \quad (3.\ 14)$$

$$= \frac{30}{(30 + 10)}$$

$$= \frac{30}{40}$$

$$= 0{,}75\ (75\%)$$

3. *Precission*

$$Precission = \frac{TP}{TP+FN} \quad (3.\ 15)$$

$$= \frac{30}{30+20}$$

$$= \frac{30}{50}$$

$$= 0{,}60 = 60\%$$

4. *F1-score*

$$F1\text{-}score = 2\ x\ \frac{(Precision\ x\ Recall)}{(Precision + Recall)} \quad (3.\ 16)$$

$$= 2\ x\ \frac{(0{,}60\ \text{x}\ 0{,}75)}{(0{,}60 + 0{,}75)}$$

$$= 2\ x\ \frac{0{,}45}{1{,}35}$$

$$= \frac{0{,}90}{1{,}35}$$

$$= 0{,}6667\ (66{,}67\%)$$

3.10.2 *K-Fold Cross Validation*

Untuk memastikan bahwa peningkatan performa model tidak hanya terjadi pada satu pembagian data, digunakan metode *K-Fold Cross Validation* pada data latih yang telah di-SMOTE. Dalam penelitian ini, jumlah *fold* (K) disesuaikan dengan skenario pengujian, yaitu menggunakan nilai K = 3. Pada setiap iterasi, model dilatih menggunakan K−1 subset sebagai data latih dan satu subset sebagai data validasi, hingga seluruh subset digunakan sebagai data validasi secara bergantian.

Rata-rata nilai akurasi dari seluruh iterasi dihitung dengan rumus (Dewi et al., 2019):

$$Accuracy_{avg} = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \quad (3.\ 17)$$

Keterangan:
$Accuracy_i$= nilai akurasi yang diperoleh pada *Fold* ke–i
K= jumlah total Fold dalam proses *K-Fold Cross Validation*
$Accuracy_{avg}$= nilai akurasi rata-rata dari seluruh *Fold*

Contoh perhitungan manual *K-Fold Cross Validation* (K=3). dataset dibagi menjadi tiga bagian dengan ukuran yang relatif sama. Pada setiap iterasi, dua bagian digunakan sebagai data latih dan satu bagian sebagai data validasi.

Tabel 3. 7 Contoh *k-fold cross validation*

Fold	Data Latih (80 data)	Data Uji (20 Data)	Akurasi
Fold 1	*Fold* 2, 3	*Fold* 1	80%
Fold 2	*Fold* 1, 3	*Fold* 2	82%
Fold 3	*Fold* 1, 2	*Fold* 3	78%

Berdasarkan Tabel 3.6, Rata-rata akurasi dihitung sebagai berikut:

$$\text{Akurasi rata-rata} = \frac{80\%+82\%+78\%}{3}$$

$$= 80\%$$

Dengan demikian, nilai rata-rata akurasi dari proses *K-Fold Cross Validation* digunakan sebagai indikator performa model yang lebih stabil dibandingkan hanya menggunakan satu pembagian data.

3.10.3 ROC & AUC *(Receiver Operating Characteristic & Area Under Curve)*

Sebagai evaluasi tambahan, digunakan kurva ROC *(Receiver Operating Characteristic)* dan nilai AUC *(Area Under Curve)* untuk menilai kemampuan model dalam membedakan kelas *churn* dan *non-churn*. ROC menggambarkan hubungan antara *True positive rate* (TPR) dan *False positive rate* (FPR) pada berbagai nilai ambang batas (*threshold*) (Kurniawan et al., 2023).

$$TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{FP+TN} \quad (3.\ 18)$$

$$\text{TPR} = \frac{30}{30+10} = 0.75$$

Artinya, model berhasil mendeteksi 75% dari seluruh pengguna yang benar-benar *churn*.

$$\text{FPR} = \frac{20}{32+140} = 0.125$$

Artinya, 12,5% pengguna yang sebetulnya *non-churn* diprediksi *churn* oleh model.

Nilai AUC dihitung dari luas area di bawah kurva ROC dan berkisar antara 0,5 hingga 1:

- AUC mendekati 1 menunjukkan model sangat baik dalam membedakan kelas.
- AUC mendekati 0,5 menunjukkan model tidak memiliki kemampuan klasifikasi (setara tebakan acak).

Perhitungan AUC dapat dilakukan dengan pendekatan trapesium menggunakan tiga titik ROC, yaitu:

1. (0, 0)
2. (FPR = 0,125 , TPR = 0,75)
3. (1, 1)

Luas area trapesium pertama:

$$AUC_1 = \frac{(0+0.75)}{2} \times (0.125 - 0) = 0.375 \times 0.125 = 0.046875$$

Luas area trapesium kedua:

$$AUC_2 = \frac{(0.75+1)}{2} \times (1 - 0.125) = 0.875 \times 0.875 = 0.765625$$

Total AUC:

$$AUC = 0.046875 + 0.765625 = 0.8125$$

Dengan demikian nilai AUC model adalah sekitar 0,81, yang menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan pengguna *churn*

dan *non-churn*. Kurva ROC yang dihasilkan menggambarkan keseimbangan antara tingkat deteksi *churn* (TPR) dan tingkat kesalahan prediksi (FPR), dan nilai AUC memberikan bukti kuantitatif bahwa model bekerja secara efektif dalam melakukan klasifikasi.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Data Penelitian

Dataset yang digunakan dalam penelitian ini merupakan data aktivitas pengguna pada platform bimbingan belajar online yang diperoleh dari Kaggle. Dataset ini digunakan untuk memprediksi kemungkinan pengguna melakukan perpanjangan langganan (*renewal*) atau tidak. Variabel target dalam penelitian ini adalah Renewal_Status, yang menunjukkan status perpanjangan langganan pengguna. Variabel ini terdiri dari dua kelas, yaitu:

0 (Tidak *Churn*) → pengguna melakukan perpanjangan langganan

1 (*Churn*) → pengguna tidak melakukan perpanjangan langganan

Dengan demikian, penelitian ini termasuk dalam klasifikasi biner (*binary classification*). Dataset terdiri atas 14.843 data pengguna dengan 24 atribut yang merepresentasikan aktivitas pengguna, performa pembelajaran, serta status layanan pengguna pada platform bimbingan belajar daring. Pada tahap awal penelitian digunakan seluruh fitur numerik serta dua fitur kategorikal, yaitu Tutoring dan Subscription_Tier sebagai fitur awal penelitian.

Selanjutnya dilakukan proses *feature selection* menggunakan *feature importance Random Forest* untuk memperoleh fitur yang paling berpengaruh terhadap klasifikasi Renewal_Status.

```
top_features = feature_importance.head(7)['Feature'].tolist()

print("\nTop Features:")
print(top_features)
```

Gambar 4. 1 Source Code Top Features

Source code pada Gambar 4.1 digunakan untuk memilih tujuh fitur dengan nilai *feature importance* tertinggi. Fungsi *head(7)* mengambil tujuh fitur teratas berdasarkan tingkat kepentingannya terhadap proses klasifikasi, kemudian disimpan ke dalam *variabel top_features* untuk digunakan pada tahap pemodelan selanjutnya. Berdasarkan hasil *feature selection*, diperoleh beberapa fitur utama yang digunakan pada proses pemodelan, yaitu:

Tabel 4. 1 Fitur Hasil *Feature Selection*

No.	Nama Fitur
1.	Average_Score
2.	Points_Earned
3.	Completion_Rate
4.	Exercises_Started
5.	Total_Time_Spent_in_Minutes
6.	Age_in_Months
7.	Exercises_Completed

Berdasarkan Tabel 4.1 hasil *feature selection*, fitur-fitur yang terpilih merepresentasikan tingkat aktivitas, keterlibatan, dan performa pengguna selama menggunakan platform pembelajaran daring. Fitur-fitur tersebut digunakan sebagai variabel input pada proses klasifikasi menggunakan algoritma *Random Forest Classifier*.

4.2 Hasil *Preprocessing* Data

Tahap *preprocessing data* dilakukan untuk mempersiapkan data sebelum digunakan pada proses pemodelan menggunakan algoritma *Random Forest Classifier*. Tahapan ini bertujuan agar data memiliki format yang sesuai, lebih terstruktur, dan dapat diproses secara optimal oleh model machine learning. Pada penelitian ini, tahapan *preprocessing* meliputi *encoding data*, *feature selection*, dan normalisasi data menggunakan *MinMax Scaling*.

4.2.1 *Encoding Data*

Tahap *encoding data* dilakukan pada fitur kategorikal yang digunakan dalam penelitian, yaitu *Tutoring* dan *Subscription_Tier*. Proses ini bertujuan untuk mengubah data kategorikal menjadi format numerik agar dapat diproses oleh algoritma *Random Forest*.

```
df_raw['Tutoring'] = df_raw['Tutoring'].map({
    'Yes': 1,
    'No': 0
})
```

Gambar 4. 2 Source Code Encoding Tutoring

```
df_raw['Subscription_Tier'] =
df_raw['Subscription_Tier'].map({
    'Free': 0,
    'Basic': 1,
    'Premium': 2
})
```

Gambar 4. 3 Source Code Encoding Subscription_Tier

Pada Gambar 4.2 dan Gambar 4.3 ditunjukkan proses *encoding* pada variabel Tutoring dan Subscription_Tier. Variabel Tutoring dikonversi dari

kategori *"Yes"* dan *"No"* menjadi nilai numerik 1 dan 0, sedangkan variabel Subscription_Tier dikonversi dari kategori *"Free", "Basic",* dan *"Premium"* menjadi nilai 0, 1, dan 2. Proses *encoding* dilakukan untuk mengubah data kategorikal menjadi data numerik sehingga dapat diproses oleh algoritma *Random Forest*. Hasil *encoding* menunjukkan bahwa seluruh fitur kategorikal berhasil diubah ke dalam bentuk numerik sehingga dapat digunakan pada proses pemodelan menggunakan algoritma *Random Forest*.

4.2.2 Normalisasi Data

Tahap normalisasi data dilakukan menggunakan metode *MinMax Scaling* untuk menyamakan rentang nilai antar fitur numerik sehingga proses pembelajaran model menjadi lebih optimal.

Metode *MinMax Scaling* mengubah rentang nilai fitur menjadi skala [0,1]. Proses normalisasi dilakukan pada data latih, kemudian parameter hasil normalisasi diterapkan pada data uji untuk menjaga konsistensi skala data dan mencegah terjadinya *data leakage*.

Hasil normalisasi menunjukkan bahwa seluruh fitur numerik berhasil ditransformasikan ke dalam rentang nilai yang seragam sehingga data menjadi lebih optimal untuk digunakan pada proses klasifikasi menggunakan algoritma *Random Forest Classifier*.

4.3 Penyeimbang Data (SMOTE)

Dataset yang digunakan dalam penelitian ini memiliki distribusi kelas yang tidak seimbang (*imbalanced data*), di mana jumlah data *churn* lebih besar

dibandingkan *non-churn*. Kondisi tersebut dapat menyebabkan model *Random Forest* menjadi bias terhadap kelas mayoritas sehingga kemampuan model dalam mengenali kelas minoritas menjadi kurang optimal.

Untuk mengatasi permasalahan tersebut, digunakan metode *Synthetic Minority Over-sampling Technique (SMOTE)*. Metode *SMOTE* bekerja dengan menghasilkan data sintetis pada kelas minoritas berdasarkan kedekatan antar data sehingga distribusi data menjadi lebih seimbang tanpa hanya melakukan duplikasi data.

Penerapan *SMOTE* dilakukan setelah tahap normalisasi data dan hanya diterapkan pada data latih (*training data*) pada setiap skenario pengujian. Hal ini dilakukan untuk mencegah terjadinya *data leakage* yang dapat memengaruhi hasil evaluasi model.

```
smote = SMOTE(random_state=42)

X_train, y_train = smote.fit_resample(X_train, y_train)
```

Gambar 4. 4 Source Code SMOTE

Pada Gambar 4.4 ditunjukkan proses penerapan *SMOTE* pada data latih (*training data*) menggunakan fungsi *fit_resample()*. Proses ini bertujuan untuk menyeimbangkan distribusi kelas dengan menghasilkan data sintetis pada kelas minoritas sebelum model *Random Forest* dibangun. Hasil distribusi data sebelum dan sesudah penerapan *SMOTE* pada setiap skenario ditunjukkan pada Tabel 4.2.

Tabel 4. 2 Distribusi Data Sebelum dan Sesudah SMOTE

Skenario	Sebelum SMOTE	Sesudah SMOTE
Skenario 1	Churn = 654, Non-Churn = 146	Churn = 654, Non-Churn = 654
Skenario 2	Churn = 1290, Non-Churn = 310	Churn = 1290, Non-Churn = 1290
Skenario 3	Churn = 3211, Non-Churn = 789	Churn = 3211, Non-Churn = 3211
Skenario 4	Churn = 9510, Non-Churn = 2364	Churn = 9510, Non-Churn = 9510

Berdasarkan hasil penerapan *SMOTE*, jumlah data pada kelas minoritas berhasil ditingkatkan sehingga distribusi kelas menjadi lebih seimbang pada setiap skenario pengujian. Dengan distribusi data yang lebih seimbang, model *Random Forest* diharapkan dapat mempelajari pola dari kedua kelas secara lebih optimal dan menghasilkan performa klasifikasi yang lebih baik.

4.4 Pembentukan Model *Random Forest*

Model *Random Forest* dilatih menggunakan data latih yang telah melalui proses transformasi sebelumnya. Pada tahap ini, model mempelajari pola hubungan antara fitur input dan variabel target (Renewal_Status). Setiap pohon dalam *Random Forest* dibangun menggunakan subset data yang berbeda melalui proses *bootstrap sampling*. Selain itu, pemilihan fitur secara acak pada setiap *node* dilakukan untuk meningkatkan variasi antar pohon dan mengurangi risiko *overfitting*. Setelah seluruh pohon terbentuk, hasil prediksi dari masing-masing pohon digabungkan menggunakan mekanisme *majority voting* untuk menghasilkan prediksi akhir.

Implementasi model *Random Forest* pada penelitian ini dilakukan menggunakan pustaka *scikit-learn*. Model dilatih menggunakan data latih yang diperoleh dari skenario pengujian, dengan memanfaatkan fitur-fitur yang telah ditentukan pada tahap sebelumnya. Untuk meningkatkan performa model, dilakukan proses tuning *hyperparameter* sehingga diperoleh kombinasi parameter terbaik yang digunakan dalam pembentukan model. Parameter tersebut meliputi jumlah pohon (*n_estimators*), kedalaman maksimum pohon (*max_depth*), jumlah

minimum sampel untuk melakukan split (*min_samples_split*), serta jumlah minimum sampel pada node akhir (*min_samples_leaf*).

4.4.1 *Boostrap Sampling*

Pada tahap implementasi *Random Forest*, proses *bootstrap sampling* dilakukan secara otomatis oleh algoritma untuk membentuk subset data latih pada setiap pohon keputusan. Metode bootstrap sampling merupakan teknik pengambilan sampel dengan pengembalian *(sampling with replacement)*, sehingga memungkinkan suatu data terpilih lebih dari satu kali dalam satu subset.

Dalam penelitian ini, proses bootstrap sampling tidak diatur secara manual, melainkan menggunakan pengaturan default dari algoritma *Random Forest* pada pustaka *Scikit-learn*, di mana setiap pohon dibangun dari sampel acak yang diambil dari data latih. Hal ini menyebabkan setiap pohon memiliki variasi data yang berbeda.

Variasi data latih pada setiap pohon ini bertujuan untuk meningkatkan keberagaman model *(diversity)*, sehingga ketika dilakukan proses agregasi *(ensemble)*, model *Random Forest* dapat menghasilkan prediksi yang lebih stabil dan memiliki kemampuan generalisasi yang lebih baik.

4.4.2 *Random Feature selection*

Pada tahap selanjutnya dalam pembentukan model *Random Forest*, dilakukan proses Random *Feature selection*. Setelah subset data latih terbentuk melalui *bootstrap sampling*, setiap pohon keputusan tidak menggunakan seluruh fitur yang tersedia untuk melakukan pemisahan *node*.

Sebaliknya, pada setiap node, algoritma secara acak memilih sebagian fitur dari keseluruhan fitur yang tersedia untuk dipertimbangkan dalam proses pembentukan split. Mekanisme ini bertujuan untuk mengurangi korelasi antar pohon keputusan yang terbentuk dalam *Random Forest*.

Dalam penelitian ini, proses *Random Feature selection* dilakukan secara otomatis oleh algoritma *Random Forest* yang diimplementasikan menggunakan pustaka *Scikit-learn*. Parameter *max_features* yang digunakan mengikuti pengaturan default, sehingga jumlah fitur yang dipilih secara acak pada setiap node menyesuaikan dengan karakteristik data yang digunakan.

Dengan adanya pemilihan fitur secara acak ini, setiap pohon dalam *Random Forest* akan memiliki struktur yang berbeda karena dibangun berdasarkan kombinasi fitur yang tidak sama. Hal ini meningkatkan keberagaman model *(diversity)* dan membantu model dalam menangkap berbagai pola dalam data. Keberagaman antar pohon tersebut berkontribusi terhadap peningkatan kemampuan generalisasi model secara keseluruhan, sehingga *Random Forest* menjadi lebih robust dan tidak bergantung pada satu fitur tertentu.

4.4.3 Pembentukan Decision Tree

Pada algoritma *Random Forest*, model dibangun dari kumpulan beberapa *decision tree* yang dilatih menggunakan subset data berbeda melalui mekanisme *bootstrap sampling*. Setiap *decision tree* bekerja dengan membagi data berdasarkan fitur tertentu hingga menghasilkan klasifikasi akhir pada *leaf node*.

Pada penelitian ini, visualisasi *decision tree* dilakukan untuk melihat proses pembentukan aturan klasifikasi pada salah satu pohon keputusan yang terbentuk

dalam model *Random Forest*. Visualisasi dilakukan menggunakan fungsi *plot_tree()* dari pustaka *scikit-learn*.

```
tree = rf.estimators_[0]
plt.figure(figsize=(20,10))
plot_tree(
    tree,
    filled=True,
    feature_names=fitur,
    class_names=['Non-Churn', 'Churn'],
    fontsize=8
)
plt.title(f"Decision Tree - {name}")
plt.show()
```

Gambar 4. 5 Source Code Visualisasi *Decision Tree*

Fungsi pada Gambar 4.5 tersebut digunakan untuk menampilkan salah satu *decision tree* yang terbentuk dalam model *Random Forest*. Visualisasi *decision tree* untuk setiap skenario pengujian disajikan pada subbab hasil skenario uji coba sesuai dengan jumlah data yang digunakan pada masing-masing skenario.

4.4.4 Agregasi Hasil Prediksi (Bagging)

Pada tahap akhir dalam pembentukan model *Random Forest*, dilakukan proses agregasi hasil prediksi yang dikenal sebagai *Bootstrap Aggregating* atau *Bagging*. Pada tahap ini, setiap pohon keputusan yang telah dibangun sebelumnya akan menghasilkan prediksi secara independen terhadap data uji.

Dalam penelitian ini, karena permasalahan yang diangkat merupakan klasifikasi *churn*, maka metode agregasi yang digunakan adalah *majority voting*. Setiap pohon

keputusan memberikan prediksi berupa kelas (0 atau 1), kemudian hasil akhir ditentukan berdasarkan kelas yang paling banyak diprediksi oleh seluruh pohon dalam *Random Forest*.

Mekanisme *voting* mayoritas ini bertujuan untuk meningkatkan stabilitas dan akurasi model, karena keputusan akhir tidak bergantung pada satu pohon saja, melainkan merupakan hasil kombinasi dari banyak pohon yang memiliki variasi struktur dan pola pembelajaran yang berbeda.

Dengan adanya proses bagging, *Random Forest* mampu mengurangi variansi model serta meminimalkan risiko *overfitting* yang sering terjadi pada model *decision tree* tunggal. Hal ini membuat model menjadi lebih *robust* dalam melakukan prediksi terhadap data baru.

4.5 Hasil *Tuning Hyperparameter*

Pada tahap ini dilakukan proses *tuning hyperparameter* untuk memperoleh kombinasi parameter terbaik yang dapat meningkatkan performa model *Random Forest*. Proses tuning bertujuan untuk mengoptimalkan kinerja model sehingga mampu menghasilkan prediksi yang lebih akurat dan stabil.

Tuning dilakukan menggunakan metode *Grid Search Cross Validation (GridSearchCV)*, yang bekerja dengan menguji seluruh kombinasi parameter yang telah ditentukan untuk menemukan parameter dengan performa terbaik. Metode ini memastikan bahwa kombinasi parameter yang dipilih merupakan hasil evaluasi menyeluruh dari ruang parameter yang diuji.

```
grid = GridSearchCV(
       RandomForestClassifier(random_state=42),
       param_grid,
       cv=3,
       scoring='f1',
       n_jobs=-1
   )
```

Gambar 4. 6 Source Code *Hyperparameter*

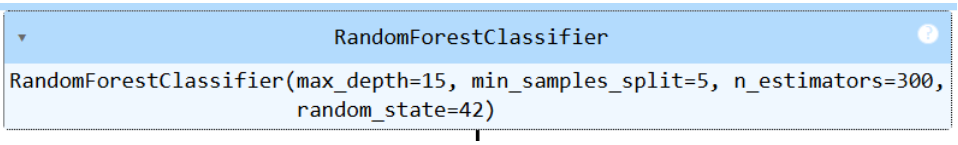
Pada gambar 4. 6 menunjukkan implementasi metode *GridSearchCV* yang digunakan untuk melakukan tuning *hyperparameter* pada model *Random Forest*. Pada kode tersebut, objek *GridSearchCV* dibentuk dengan menggunakan model dasar *RandomForestClassifier* dengan parameter *random_state = 42* untuk menjaga konsistensi hasil.

Parameter *param_grid* berisi kombinasi nilai *hyperparameter* yang akan diuji, seperti *n_estimators*, *max_depth*, dan *min_samples_split*. *GridSearchCV* kemudian akan menguji seluruh kombinasi parameter tersebut untuk menemukan konfigurasi yang memberikan performa terbaik.

Proses evaluasi dilakukan menggunakan teknik *cross-validation* dengan nilai *cv* = 3, di mana data latih dibagi menjadi tiga bagian untuk dilakukan pelatihan dan pengujian secara bergantian. Selain itu, digunakan parameter *scoring = 'f1'* sebagai acuan dalam menilai performa model pada setiap kombinasi parameter yang diuji.

Parameter *n_jobs = -1* digunakan untuk memanfaatkan seluruh inti prosesor yang tersedia sehingga proses pencarian parameter dapat berjalan lebih cepat. Hasil dari proses ini berupa kombinasi parameter terbaik yang kemudian digunakan dalam pembentukan model *Random Forest* pada tahap selanjutnya. Berdasarkan

hasil tuning yang telah dilakukan, diperoleh kombinasi parameter terbaik sebagai berikut:



Gambar 4. 7 Nilai *Hyperparamater*

Berdasarkan Gambar 4.7 hasil proses tuning menggunakan metode GridSearchCV diperoleh kombinasi parameter terbaik pada model *Random Forest*, yaitu *n_estimators = 500*, *max_depth = 15*, *min_samples_split = 5, min_samples_leaf = 2*.

Parameter *n_estimators* menunjukkan jumlah pohon yang digunakan dalam model, di mana penggunaan 500 pohon bertujuan untuk meningkatkan stabilitas dan akurasi prediksi. Parameter *max_depth = 15* membatasi kedalaman pohon sehingga model tidak terlalu kompleks dan dapat mengurangi risiko *overfitting*.

Dengan kombinasi parameter tersebut, model *Random Forest* diharapkan mampu menghasilkan performa yang lebih baik dalam melakukan klasifikasi dibandingkan dengan penggunaan parameter default.

4.6 Evaluasi Model dengan *Cross Validation*

Untuk memastikan bahwa performa model tidak bergantung pada satu pembagian data tertentu, pada penelitian ini digunakan metode *Cross Validation* pada data latih yang telah melalui proses *preprocessing* dan penyeimbangan menggunakan SMOTE. Metode yang digunakan adalah *Stratified K-Fold Cross Validation*, yang bertujuan untuk menjaga proporsi kelas *churn* dan *non-churn* tetap

seimbang pada setiap *fold*. Hal ini penting karena dataset yang digunakan memiliki karakteristik ketidakseimbangan kelas. Pada setiap iterasi, data latih dibagi menjadi beberapa subset *(fold),* di mana satu subset digunakan sebagai data validasi dan sisanya digunakan sebagai data latih. Proses ini diulang hingga seluruh subset digunakan sebagai data validasi.

```
    skf = StratifiedKFold(
        n_splits=scenario["cv"],
        shuffle=True,
        random_state=scenario["rs"]
    )

    cv_score = cross_val_score(rf, X_train, y_train, cv=skf,
scoring='f1')
    print("CV F1 Mean:", cv_score.mean())
```

Gambar 4. 8 Source Code *Cross Validation*

Pada Gambar 4.8 menunjukkan proses validasi model menggunakan metode *Stratified K-Fold Cross Validation*. Parameter *n_splits=scenario["cv"]* digunakan untuk menentukan jumlah *fold* yang digunakan pada setiap skenario, yaitu sebanyak 3 *fold*. Parameter *shuffle=True* berfungsi untuk mengacak data sebelum proses pembagian *fold* dilakukan sehingga distribusi data menjadi lebih merata. Sementara itu, *random_state=scenario["rs"]* digunakan untuk menjaga konsistensi hasil pembagian data sehingga eksperimen dapat direproduksi pada pengujian berikutnya. Setelah pembagian *fold* dilakukan, fungsi *cross_val_score()* digunakan untuk menghitung nilai *F1-Score* pada setiap *fold* dengan model *Random Forest (rf)*. Nilai evaluasi yang digunakan adalah *F1-Score (scoring='f1')* karena metrik ini mampu mengukur keseimbangan antara *precision* dan *recall* pada kasus klasifikasi dengan distribusi kelas yang tidak seimbang. Hasil evaluasi dari

seluruh *fol*d kemudian dirata-ratakan menggunakan *cv_score.mean()* untuk memperoleh nilai performa *cross validation* pada masing-masing skenario.

Sebagai contoh implementasi *Stratified K-Fold Cross Validation*, Tabel 4.3 menunjukkan komposisi pembagian data pada Skenario 1. Proses yang sama diterapkan pada seluruh skenario pengujian dengan jumlah data yang berbeda.

Tabel 4. 3 Contoh Komposisi *Fold* pada Skenario 1

Fold	**Data *Training***	**Data Validasi**
Fold 1	533	267
Fold 2	533	267
Fold 3	534	266

Berdasarkan Tabel 4.3, data training pada Skenario 1 yang berjumlah 800 data dibagi menjadi tiga *fold* menggunakan metode *Stratified K-Fold Cross Validation*. Pada setiap iterasi, dua *fold* digunakan sebagai data *training* dan satu *fold* digunakan sebagai data validasi. Pembagian data dilakukan secara proporsional sehingga setiap *fold* memiliki jumlah data yang relatif seimbang.

Tabel 4. 4 Contoh Distribusi Kelas pada Setiap *Fold*

Fold	***Non-Churn Train***	***Churn Train***	***Non-Churn* Validasi**	***Churn* Validasi**
Fold 1	436	97	218	49
Fold 2	436	97	218	49
Fold 3	436	98	218	48

Berdasarkan Tabel 4.4, distribusi kelas *churn* dan *non-churn* pada setiap *fold* relatif konsisten. Hal ini menunjukkan bahwa metode *Stratified K-Fold* berhasil mempertahankan proporsi kelas pada data *training* maupun data validasi. Dengan demikian, proses evaluasi model dapat dilakukan secara lebih representatif karena setiap *fold* memiliki karakteristik distribusi data yang serupa dengan dataset awal.

Tabel 4. 5 Hasil Evaluasi Setiap *Fold* pada Skenario 1

Fold	***F1-Score***
Fold 1	0,8080
Fold 2	0,7974
Fold 3	0,8115

Berdasarkan Tabel 4.5, nilai *F1-Score* yang diperoleh pada setiap *fold* menunjukkan hasil yang relatif stabil dengan perbedaan yang tidak terlalu besar. *Fold* 3 menghasilkan nilai *F1-Score* tertinggi sebesar 0,8115, sedangkan *Fold* 2 menghasilkan nilai terendah sebesar 0,7974. Hasil ini menunjukkan bahwa model *Random Forest* memiliki performa yang cukup konsisten pada setiap proses validasi sehingga model dapat dikatakan stabil dalam melakukan klasifikasi *Renewal Status*.

Hasil *cross validation* menghasilkan nilai rata-rata *F1-score* pada setiap skenario pengujian. Hasil *cross validation* ditunjukkan sebagai berikut:

Tabel 4. 6 Hasil *Cross Validation*

Skenario	Hasil C*ross Validation*
Skenario 1	CV F1 Mean: 0.805
Skenario 2	CV F1 Mean: 0.795
Skenario 3	CV F1 Mean: 0.811
Skenario 4	CV F1 Mean: 0.773

Berdasarkan Tabel 4.6, nilai rata-rata *F1-Score* hasil *cross validation* menunjukkan bahwa skenario 3 memperoleh performa terbaik dengan nilai sebesar 0,8116. Sementara itu, skenario 4 memperoleh nilai terendah sebesar 0,7737. Perbedaan nilai *F1-Score* antar skenario menunjukkan adanya variasi performa model pada jumlah data yang berbeda.

Pada penelitian ini, digunakan *3-fold cross validation* (K = 3) yang berarti data latih dibagi menjadi tiga bagian. Pada setiap iterasi, dua bagian digunakan sebagai data latih dan satu bagian sebagai data validasi. Penggunaan *Stratified K-*

Fold memastikan bahwa distribusi kelas tetap seimbang pada setiap *fold*, sehingga evaluasi model menjadi lebih representatif, terutama pada kasus *imbalanced data*. Nilai evaluasi yang digunakan pada proses *cross validation* adalah F1-score, karena metrik ini mempertimbangkan keseimbangan antara *precision* dan *recall*, sehingga lebih sesuai untuk kasus klasifikasi *churn* yang memiliki distribusi kelas tidak seimbang.

4.7 Hasil Skenario Uji Coba

Pada tahap ini dilakukan pengujian model *Random Forest Classifier* untuk mengetahui performa model dalam mengklasifikasikan Renewal_Status berdasarkan variasi jumlah data yang digunakan. Pengujian dilakukan menggunakan empat skenario, yaitu 1000 data, 2000 data, 5000 data, dan seluruh data (*full dataset*).

Setiap skenario menggunakan tahapan *preprocessing* dan proses pemodelan yang sama sehingga perbedaan hasil yang diperoleh dapat dianalisis berdasarkan jumlah data yang digunakan pada proses pelatihan dan pengujian model.

4.7.1 Alur Pengujian Model

Pengujian dilakukan menggunakan empat skenario jumlah data yang berbeda, yaitu 1000 data, 2000 data, 5000 data, dan seluruh dataset. Pada setiap skenario, data dibagi menjadi data training dan data testing menggunakan rasio 80:20 dengan metode stratified sampling untuk mempertahankan proporsi kelas churn dan non-churn.

Tabel 4. 7 Pembagian Data Training dan Testing

Skenario	**Total Data**	***Training***	***Testing***	***Training Non-Churn***	***Training Churn***	***Testing Non-Churn***	***Testing Churn***
Skenario 1	1000	800	200	654	146	164	36
Skenario 2	2000	1600	400	1290	310	322	78
Skenario 3	5000	4000	1000	3211	789	803	197
Skenario 4	14843	11874	2969	9510	2364	2378	591

Berdasarkan Tabel 4.7, proses pembagian data dilakukan menggunakan rasio 80:20 dengan metode *stratified sampling*. Metode ini mempertahankan proporsi kelas *churn* dan *non-churn* pada data *training* maupun *testing* di setiap skenario. Hal tersebut terlihat dari distribusi kelas yang relatif konsisten setelah proses pembagian data. Data *training* digunakan untuk proses pelatihan model *Random Forest*, sedangkan data *testing* digunakan untuk mengevaluasi kemampuan model dalam mengklasifikasikan data yang belum pernah digunakan pada proses pelatihan.

Pada setiap skenario, proses pengujian dilakukan melalui beberapa tahapan yang sama, yaitu sebagai berikut:

1. Pengambilan Sampel Data

   Dataset diambil dari data hasil preprocessing sesuai dengan jumlah data pada masing-masing skenario.

2. Pembagian Data (*Split Data*)

   Dataset dibagi menjadi data latih dan data uji dengan perbandingan 80:20 menggunakan metode *stratified sampling*.

3. Normalisasi Data

Data latih dinormalisasi menggunakan metode *MinMax Scaling*, kemudian digunakan untuk mentransformasikan data uji.

4. Penyeimbangan Data (SMOTE)

   Teknik SMOTE diterapkan pada data latih untuk mengatasi ketidakseimbangan data.

5. Pelatihan Model

   Model *Random Forest* dilatih menggunakan parameter yang telah ditentukan.

6. Evaluasi Model

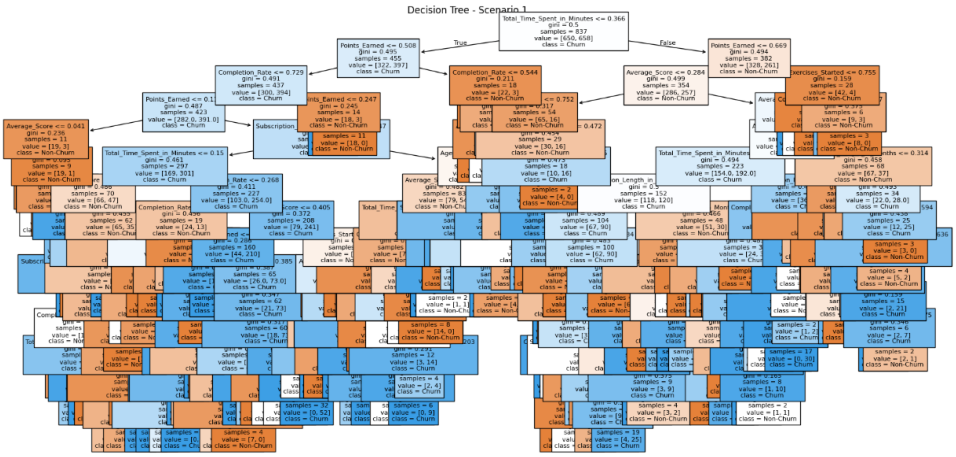
   Model dievaluasi menggunakan *Stratified K-Fold Cross Validation* dan juga metrik *Accuracy, Recall, Precision, F1-Score*, dan ROC AUC.

4.7.2 Skenario 1

Pada skenario pertama, digunakan sebanyak 1000 data yang diambil secara acak (*random sampling*) dari dataset utama. Data kemudian dibagi menjadi data latih dan data uji menggunakan rasio 80:20 dengan metode *stratified sampling*. Setelah proses pembagian data, dilakukan normalisasi menggunakan *MinMax Scaling* serta penyeimbangan data menggunakan metode *SMOTE* pada data latih. Model *Random Forest Classifier* kemudian dilatih menggunakan parameter terbaik hasil *GridSearchCV* dan dievaluasi menggunakan metode *Stratified K-Fold Cross Validation* sebanyak 3 *fold*.

Sebelum dilakukan evaluasi model, ditampilkan terlebih dahulu visualisasi salah satu *decision tree* yang terbentuk pada model *Random Forest* untuk

memberikan gambaran mengenai proses pembentukan aturan klasifikasi pada Skenario 1. Visualisasi tersebut ditunjukkan pada Gambar 4.9.



Gambar 4. 9 Visualisasi Decision Tree Skenario 1

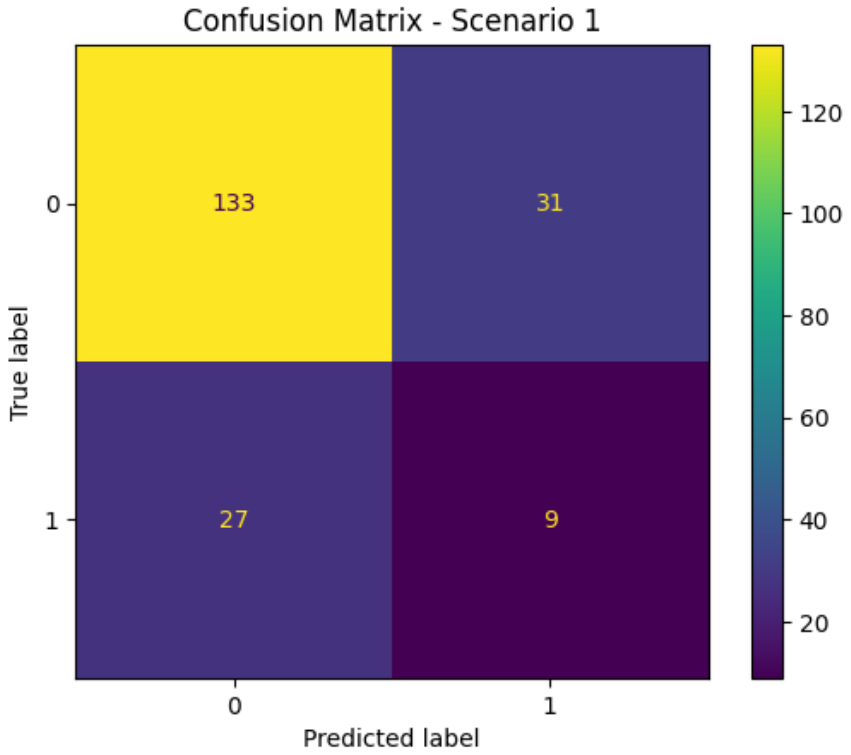
Berdasarkan visualisasi *decision tree* pada Skenario 1, terlihat bahwa *node* akar (*root node*) menggunakan fitur *Total_Time_Spent_in_Minutes* sebagai pemisah awal data. Selanjutnya, fitur seperti *Points_Earned*, *Completion_Rate*, dan *Average_Score* digunakan pada node-node berikutnya untuk membentuk aturan klasifikasi. Struktur pohon yang terbentuk menunjukkan bahwa model memanfaatkan kombinasi beberapa fitur aktivitas belajar pengguna untuk membedakan kelas *churn* dan *non-churn*.

Hasil evaluasi model pada skenario pertama ditunjukkan pada Tabel 4. 4.

Tabel 4. 8 Evaluasi Model Skenario 1

No.	Parameter	Nilai
1.	*Accuracy*	71%
2.	*Recall*	25%
3.	*Precision*	23%
4.	*F1-Score*	24%
5.	ROC AUC	54%

Berdasarkan Tabel 4.4 Hasil evaluasi model menunjukkan bahwa nilai *Accuracy* sebesar 71%, *Recall* sebesar 25%, *Precision* sebesar 23%, *F1-Score* sebesar 24%, dan ROC AUC sebesar 54%.



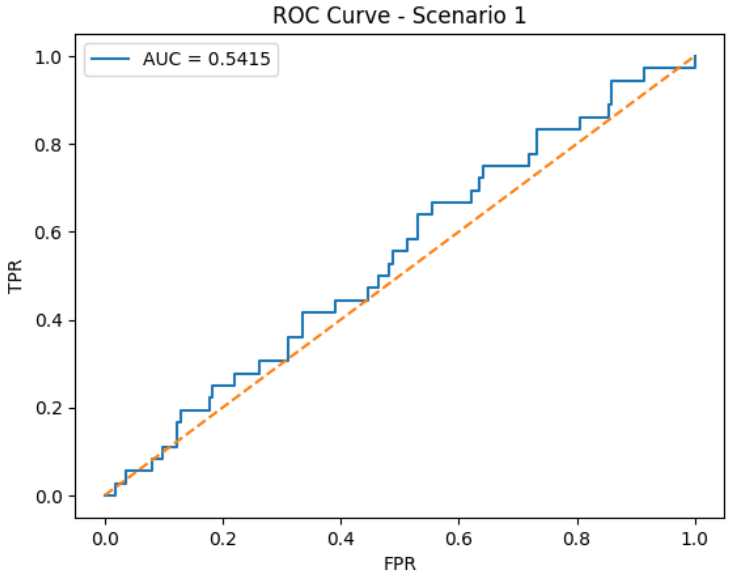
Gambar 4. 10 *Confusion matrix* Skenario 1

Berdasarkan *confusion matrix* pada Gambar 4.10, terlihat bahwa model memiliki kecenderungan yang sangat kuat dalam memprediksi kelas tidak *churn* (kelas 0). Hal ini ditunjukkan oleh tingginya nilai *True Negative* dibandingkan *True Positive.* Sebaliknya, jumlah True Positive sangat rendah, yang berarti model gagal mengenali sebagian besar pelanggan yang benar-benar *churn*.

Selain itu, nilai *False Negative* yang cukup tinggi menunjukkan bahwa banyak data *churn* yang salah diklasifikasikan sebagai tidak *churn*. Kondisi ini

sangat krusial dalam konteks bisnis, karena pelanggan yang seharusnya terdeteksi berpotensi *churn* justru terlewatkan oleh model.

Rendahnya nilai *recall* (0.23) semakin memperkuat bahwa model belum mampu menangkap pola dari kelas *churn* dengan baik. Hal ini mengindikasikan bahwa distribusi fitur antara kelas *churn* dan tidak *churn* kemungkinan memiliki kemiripan yang tinggi sehingga sulit dipisahkan oleh model.



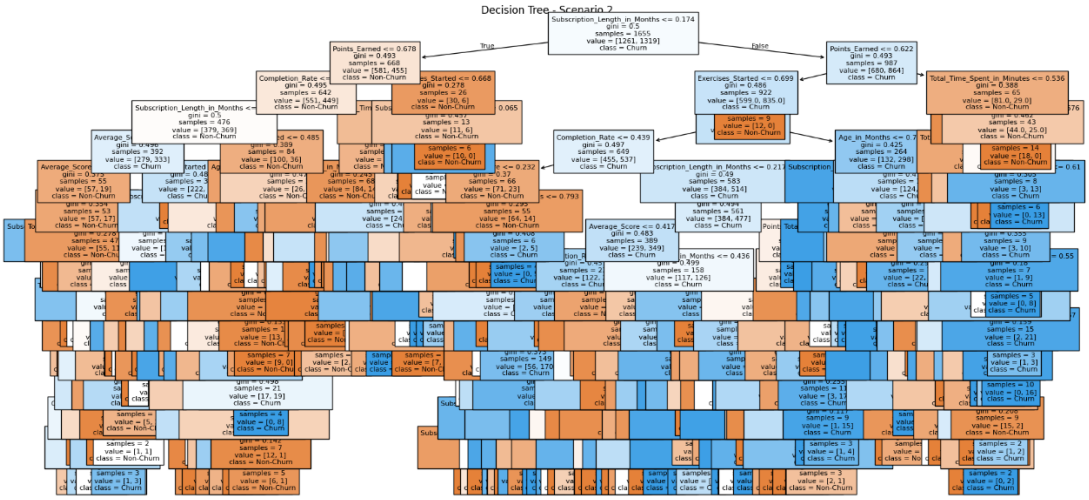
Gambar 4. 11 Kurva ROC Skenario 1

Berdasarkan Gambar 4.8, nilai ROC AUC pada Skenario 1 sebesar 54.15% menunjukkan bahwa model masih memiliki kemampuan klasifikasi yang rendah dalam membedakan pengguna *churn* dan *non-churn*. Kurva ROC yang mendekati garis diagonal mengindikasikan bahwa performa model masih mendekati prediksi acak (*random guessing*). Hal ini menunjukkan bahwa pola data pada Skenario 1 masih cukup sulit dipelajari oleh model karena jumlah data yang digunakan relatif kecil.

4.7.3 Skenario 2

Pada skenario kedua, digunakan sebanyak 2000 data yang diambil secara acak (*random sampling*) dari dataset utama. Data kemudian dibagi menjadi data latih dan data uji menggunakan rasio 80:20 dengan metode *stratified sampling*. Selanjutnya dilakukan normalisasi menggunakan *MinMax Scaling* dan penyeimbangan data menggunakan metode SMOTE pada data latih.

Visualisasi salah satu *decision tree* yang terbentuk dari model *Random Forest* ditampilkan untuk memberikan gambaran mengenai proses pembentukan aturan klasifikasi berdasarkan data yang digunakan. Visualisasi tersebut ditunjukkan pada Gambar 4.12.



Gambar 4. 12 Visualisasi *Decision Tree* Skenario 2

Berdasarkan visualisasi *decision tree* pada Gambar 4.12, terlihat bahwa node akar (*root node*) menggunakan fitur *Subscription_Length_in_Months* sebagai pemisah awal data. Hal ini menunjukkan bahwa lama berlangganan pengguna menjadi faktor utama yang digunakan model dalam melakukan klasifikasi pada

pohon keputusan tersebut. Setelah proses pemisahan awal, fitur Points_Earned, Completion_Rate, Exercises_Started, dan Average_Score digunakan pada *node-node* berikutnya untuk membentuk aturan klasifikasi yang lebih spesifik.

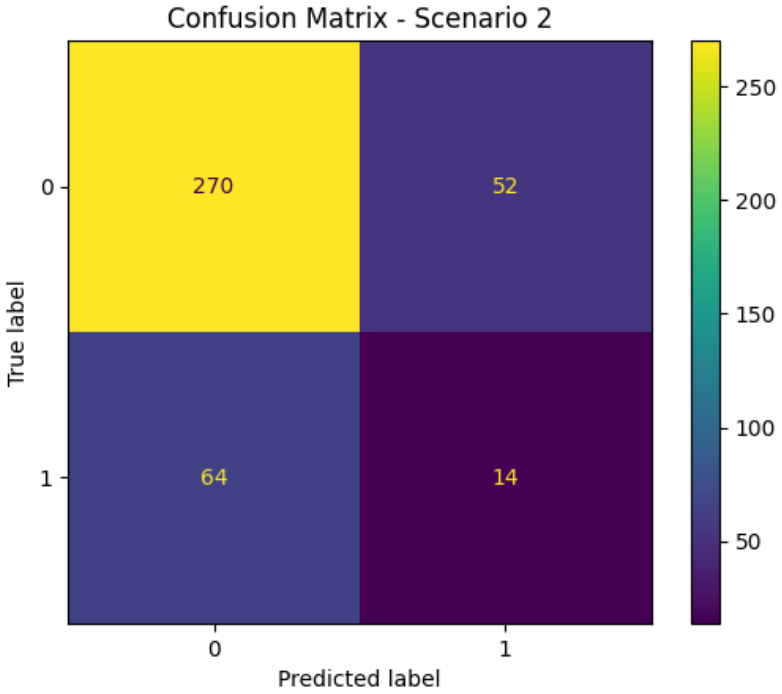
Struktur pohon yang terbentuk menunjukkan bahwa model memanfaatkan kombinasi berbagai fitur aktivitas dan karakteristik pengguna dalam membedakan kelas *churn* dan *non-churn*. Banyaknya cabang yang terbentuk mengindikasikan bahwa model membangun aturan klasifikasi yang cukup kompleks untuk mengenali pola pada data pengguna.

Hasil evaluasi model pada skenario kedua ditunjukkan pada Tabel 4. 5.

Tabel 4. 9 Evaluasi Model Skenario 2

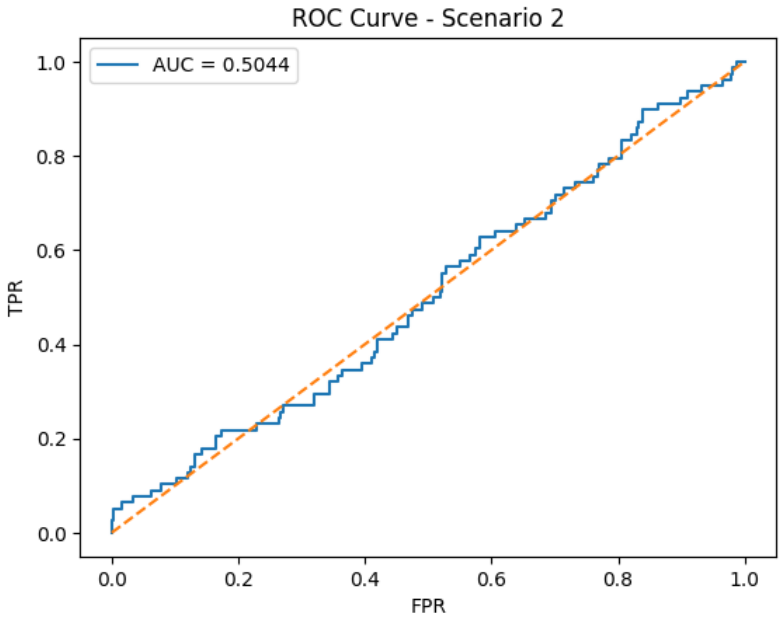
No.	Parameter	Nilai
1.	*Accuracy*	71%
2.	*Recall*	18%
3.	*Precision*	21%
4.	*F1-Score*	29%
5.	ROC AUC	50%

Berdasarkan hasil evaluasi pada Tabel 4.9, model menghasilkan nilai *Accuracy* sebesar 71.00%, namun nilai *Recall* sebesar 17.95% menunjukkan bahwa model masih kesulitan mendeteksi data *churn*. Nilai *Precision* dan *F1-Score* yang masih rendah juga menunjukkan bahwa kemampuan model dalam mengklasifikasikan pengguna *churn* belum optimal.



Gambar 4. 13 *Confusion matrix* Skenario 2

Dari confusion matrix Gambar 4.13, terlihat adanya peningkatan jumlah *True Positive* dibandingkan skenario pertama. Hal ini menunjukkan bahwa model mulai mampu mengenali sebagian data *churn* dengan lebih baik. Namun demikian, jumlah *False Negative* masih cukup tinggi, sehingga masih banyak data *churn* yang tidak terdeteksi.



Gambar 4. 14 Kurva ROC Skenario 2

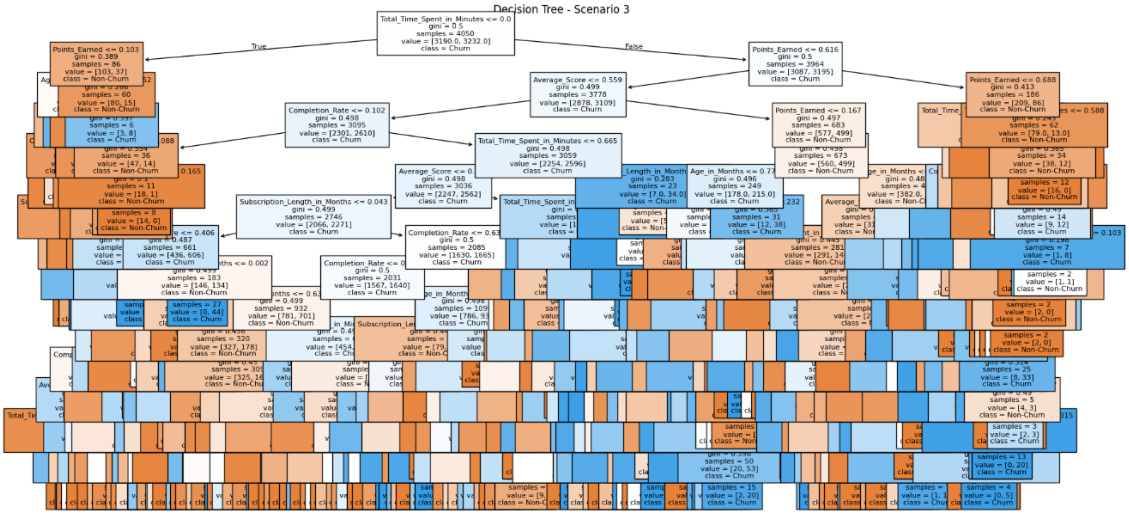
Pada Gambar 4.14 Nilai ROC AUC sebesar 50.44% menunjukkan bahwa kemampuan model dalam membedakan kelas *churn* dan *non-churn* masih rendah. Kurva ROC yang mendekati garis diagonal mengindikasikan bahwa performa model masih mendekati prediksi acak (*random guessing*).

4.7.4 Skenario 3

Pada skenario ketiga, digunakan sebanyak 5000 data yang diambil secara acak (*random sampling*) dari dataset utama. Data kemudian dibagi menjadi data latih dan data uji menggunakan rasio 80:20 dengan metode *stratified sampling*. Selanjutnya dilakukan normalisasi menggunakan *MinMax Scaling* dan penyeimbangan data menggunakan metode SMOTE pada data latih.

Visualisasi salah satu *decision tree* yang terbentuk dari model *Random Forest* ditampilkan untuk memberikan gambaran mengenai proses pembentukan

aturan klasifikasi berdasarkan data yang digunakan. Visualisasi tersebut ditunjukkan pada Gambar 4.15.



Gambar 4. 15 Visualisasi *Decision Tree* Skenario 3

Berdasarkan visualisasi *decision tree* pada Gambar 4.15, terlihat bahwa node akar (*root node*) menggunakan fitur Total_Time_Spent_in_Minutes sebagai pemisah awal data. Hal ini menunjukkan bahwa total waktu yang dihabiskan pengguna pada platform menjadi salah satu faktor utama yang digunakan model dalam membedakan pengguna *churn* dan *non-churn*. Setelah proses pemisahan awal, fitur Points_Earned, Completion_Rate, Average_Score, dan Subscription_Length_in_Months digunakan pada *node-node* berikutnya untuk membentuk aturan klasifikasi yang lebih rinci.

Dibandingkan dengan skenario sebelumnya, struktur pohon yang terbentuk terlihat lebih kompleks karena jumlah data yang digunakan lebih besar. Hal ini menunjukkan bahwa model memiliki lebih banyak informasi untuk mempelajari pola perilaku pengguna dan membentuk aturan klasifikasi yang lebih beragam.

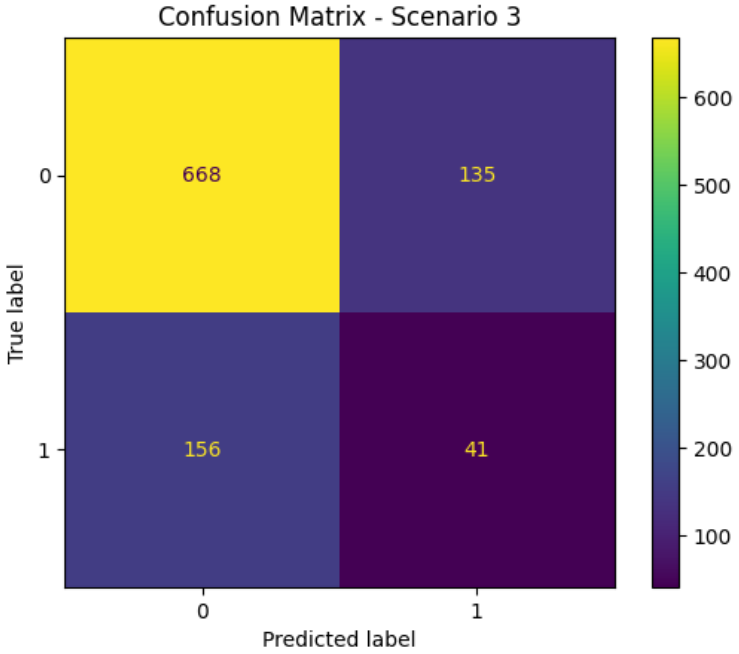
Meskipun demikian, kompleksitas pohon tidak selalu menghasilkan peningkatan performa yang signifikan, sebagaimana terlihat pada hasil evaluasi model.

Hasil evaluasi model pada skenario ketiga ditunjukkan pada Tabel 4. 6.

Tabel 4. 10 Evaluasi Model Skenario 3

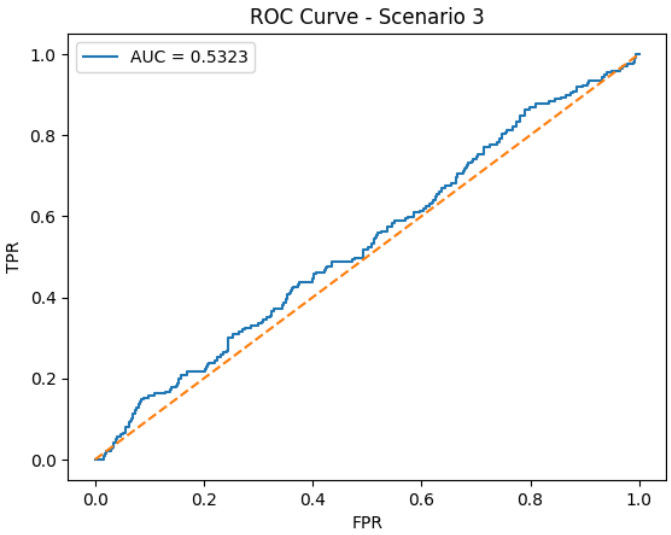
No.	Parameter	Nilai
1.	*Accuracy*	70.90%
2.	*Recall*	20.81%
3.	*Precision*	23.30%
4.	*F1-Score*	21.98%
5.	ROC AUC	53.23%

Berdasarkan hasil evaluasi pada Tabel 4.10, model menghasilkan nilai *Accuracy* sebesar 70.90%. Nilai *Recall* sebesar 20.81% menunjukkan bahwa model masih mengalami kesulitan dalam mendeteksi pengguna *churn*, meskipun terjadi sedikit peningkatan dibandingkan skenario sebelumnya. Nilai *Precision* dan *F1-Score* juga menunjukkan bahwa kemampuan klasifikasi model masih belum optimal.



Gambar 4. 16 *Confusion matrix* Skenario 3

Berdasarkan *confusion matrix* pada Gambar 4.16, model masih lebih dominan memprediksi kelas *non-churn* dibandingkan *churn*. Hal ini terlihat dari jumlah *False Negative* yang masih cukup tinggi, sehingga masih banyak data *churn* yang salah diklasifikasikan sebagai *non-churn*.



Gambar 4. 17 Kurva ROC Skenario 3

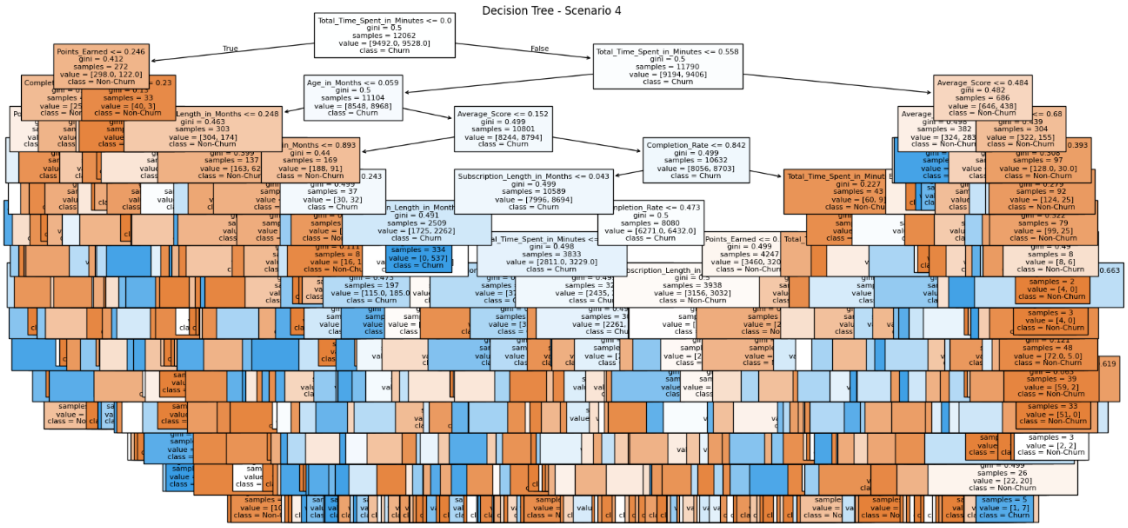
Pada Gambar 4.17 nilai ROC AUC sebesar 53.23% menunjukkan bahwa kemampuan model dalam membedakan kelas *churn* dan *non-churn* mulai mengalami peningkatan dibandingkan skenario sebelumnya. Kurva ROC juga terlihat sedikit menjauhi garis diagonal, yang menunjukkan bahwa model mulai memiliki kemampuan diskriminasi yang lebih baik meskipun peningkatannya masih terbatas.

4.7.5 Skenario 4

Pada skenario keempat, digunakan seluruh data (*full dataset*) tanpa dilakukan proses pengambilan sampel. Data kemudian dibagi menjadi data latih dan data uji menggunakan rasio 80:20 dengan metode *stratified sampling*.

Selanjutnya dilakukan normalisasi menggunakan *MinMax Scaling* dan penyeimbangan data menggunakan metode *SMOTE* pada data latih.

Visualisasi salah satu *decision tree* yang terbentuk dari model *Random Forest* ditampilkan untuk memberikan gambaran mengenai proses pembentukan aturan klasifikasi menggunakan seluruh data penelitian (*full dataset*). Visualisasi tersebut ditunjukkan pada Gambar 4.18.



Gambar 4. 18 Visualisasi *Decision Tree* Skenario 4

Berdasarkan visualisasi *decision tree* pada Skenario 4, terlihat bahwa node akar (*root node*) menggunakan fitur Total_Time_Spent_in_Minutes sebagai pemisah awal data. Hal ini menunjukkan bahwa total waktu yang dihabiskan pengguna dalam menggunakan platform menjadi salah satu faktor penting dalam proses klasifikasi churn. Selanjutnya, fitur Age_in_Months, Completion_Rate, Average_Score, Subscription_Length_in_Months, dan Points_Earned digunakan pada node-node berikutnya untuk membentuk aturan klasifikasi yang lebih spesifik.

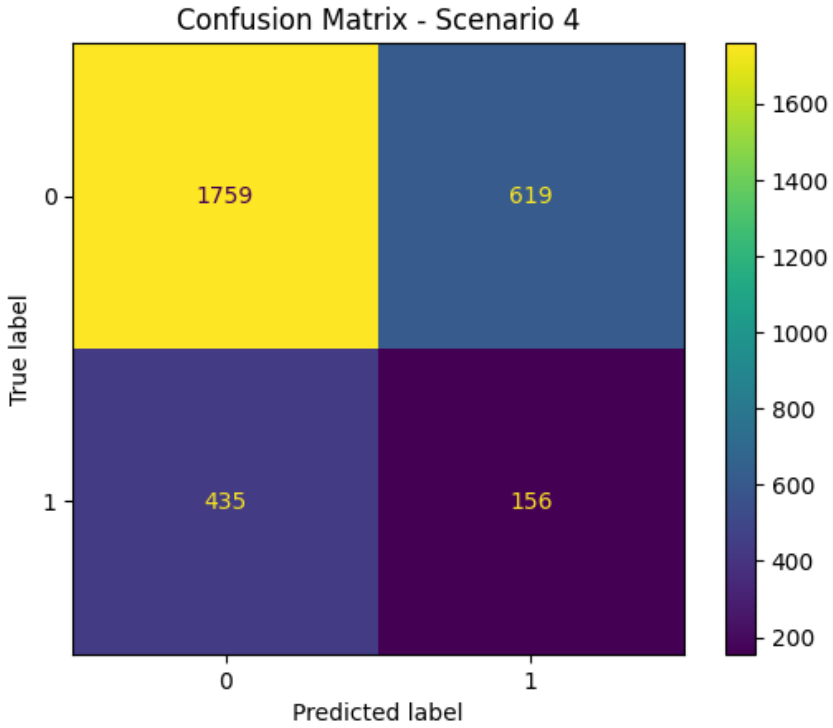
Dibandingkan dengan skenario sebelumnya, struktur pohon pada Skenario 4 terlihat lebih kompleks karena dibangun menggunakan seluruh data yang tersedia. Kompleksitas tersebut menunjukkan bahwa model memanfaatkan lebih banyak variasi pola data untuk membedakan pengguna *churn* dan *non-churn*. Namun, struktur pohon yang lebih kompleks tidak selalu menghasilkan peningkatan performa yang signifikan, sehingga evaluasi model tetap diperlukan untuk menilai kemampuan klasifikasinya secara keseluruhan.

Hasil evaluasi model pada skenario keempat ditunjukkan pada Tabel 4. 7.

Tabel 4. 11 Evaluasi Model Skenario 4

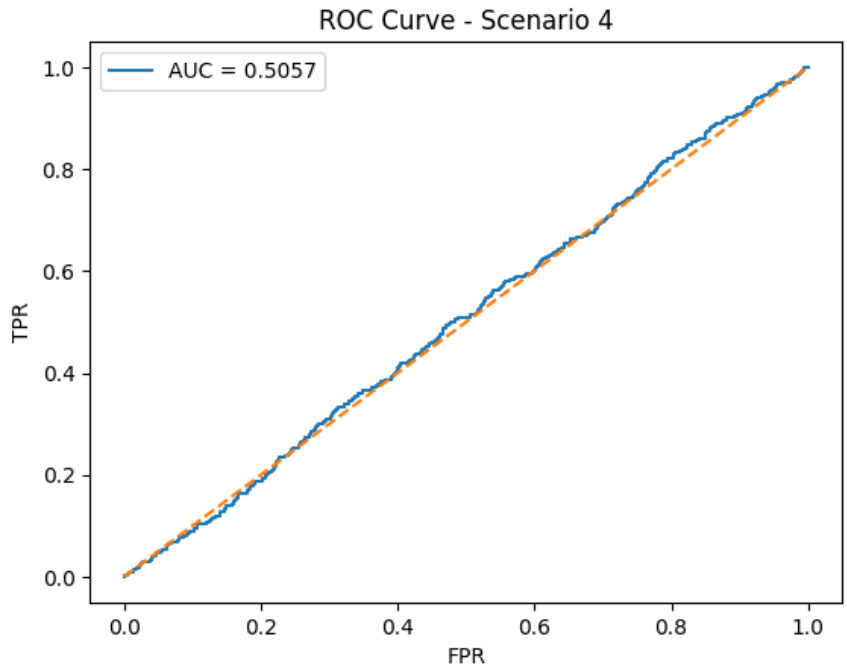
No.	Parameter	Nilai
1.	*Accuracy*	64.50%
2.	*Recall*	26.40%
3.	*Precision*	20.13%
4.	*F1-Score*	22.84%
5.	ROC AUC	50.57%

Berdasarkan hasil evaluasi pada Tabel 4.11, model menghasilkan nilai *Accuracy* sebesar 64.50%. Nilai *Recall* sebesar 26.40% menunjukkan bahwa kemampuan model dalam mendeteksi pengguna *churn* mengalami peningkatan dibandingkan skenario sebelumnya. Namun, nilai *Precision* dan *F1-Score* masih relatif rendah, sehingga performa model dalam mengklasifikasikan pengguna *churn* belum optimal.



Gambar 4. 19 *Confusion matrix* Skenario 4

Berdasarkan *confusion matrix* pada Gambar 4.19, model masih lebih dominan memprediksi kelas *non-churn* dibandingkan kelas *churn*. Hal ini terlihat dari jumlah *False Negative* yang masih cukup tinggi, sehingga masih terdapat banyak data *churn* yang salah diklasifikasikan sebagai *non-churn*.



Gambar 4. 20 Kurva ROC Skenario 4

Pada Gambar 4.20 Nilai *ROC AUC* sebesar 50.57% menunjukkan bahwa kemampuan model dalam membedakan kelas *churn* dan *non-churn* masih tergolong rendah. Kurva ROC yang mendekati garis diagonal menunjukkan bahwa performa model masih mendekati prediksi acak (*random guessing*), meskipun menggunakan jumlah data yang lebih besar.

4.8 Pembahasan

Pada penelitian ini, model *Random Forest Classifier* diuji menggunakan data yang telah melalui beberapa tahapan *preprocessing*, yaitu *encoding data*, normalisasi menggunakan metode *MinMax Scaling*, pembagian data latih dan data uji menggunakan rasio 80:20 dengan metode *stratified sampling*.

Selain itu, untuk mengatasi ketidakseimbangan data, dilakukan penyeimbangan data pada data latih menggunakan teknik SMOTE. Setelah melalui

tahapan tersebut, model *Random Forest* kemudian dilatih dan diuji pada beberapa skenario dengan jumlah data yang berbeda.

4.8.1 Analisis Performa Model

Berdasarkan hasil pengujian pada seluruh skenario, dapat disimpulkan bahwa performa model *Random Forest Classifier* masih belum optimal dalam memprediksi Renewal_Status. Hal ini ditunjukkan oleh nilai *Recall*, *Precision*, *F1-Score*, dan *ROC AUC* yang relatif rendah pada setiap skenario pengujian.

Pada Skenario 1 dengan jumlah data sebanyak 1000, model menghasilkan nilai *Accuracy* sebesar 71.00%, *Recall* sebesar 25.00%, *Precision* sebesar 22.50%, dan *F1-Score* sebesar 23.68%. Kondisi ini menunjukkan bahwa meskipun model memiliki tingkat akurasi yang cukup baik, model masih belum mampu mendeteksi sebagian besar data *churn*. Hal ini terjadi karena nilai *Accuracy* masih dipengaruhi oleh dominasi kelas mayoritas sehingga belum sepenuhnya mencerminkan performa model secara keseluruhan.

Pada Skenario 2 dengan jumlah data sebanyak 2000, model menghasilkan nilai *Accuracy* sebesar 71.00%, *Recall* sebesar 17.95%, *Precision* sebesar 21.21%, dan *F1-Score* sebesar 19.44%. Dibandingkan skenario sebelumnya, performa model belum menunjukkan peningkatan yang signifikan. Nilai *Recall* dan *Precision* yang masih rendah menunjukkan bahwa model masih kesulitan dalam mengenali pola pengguna *churn*.

Pada Skenario 3 dengan jumlah data sebanyak 5000, model menghasilkan nilai *Accuracy* sebesar 70.90%, *Recall* sebesar 20.81%, *Precision* sebesar 23.30%, dan *F1-Score* sebesar 21.98%. Pada skenario ini terlihat adanya sedikit peningkatan

pada nilai *Recall*, *Precision*, *F1-Score*, dan *ROC AUC* yang mencapai 53.23%. Hal ini menunjukkan bahwa penambahan jumlah data mulai membantu model dalam mempelajari pola klasifikasi, meskipun peningkatannya masih terbatas.

Pada Skenario 4 dengan penggunaan seluruh data (*full dataset*), model menghasilkan nilai *Accuracy* sebesar 64.50%, *Recall* sebesar 26.40%, *Precision* sebesar 20.13%, dan *F1-Score* sebesar 22.84%. Nilai *Recall* yang meningkat menunjukkan bahwa model menjadi lebih sensitif dalam mendeteksi kelas *churn*. Akan tetapi, nilai *ROC AUC* yang hanya mencapai 50.57% menunjukkan bahwa kemampuan model dalam membedakan kelas *churn* dan *non-churn* masih tergolong rendah.

Dengan demikian, dapat disimpulkan bahwa penambahan jumlah data belum memberikan peningkatan performa yang signifikan terhadap model *Random Forest Classifier* dalam penelitian ini.

4.8.2 Analisis Hasil Evaluasi

1) *Confusion Matrix*

Confusion matrix digunakan untuk memberikan gambaran yang lebih rinci mengenai distribusi hasil prediksi model, khususnya dalam mengidentifikasi kesalahan klasifikasi pada masing-masing kelas.

Berdasarkan hasil yang diperoleh pada seluruh skenario, terlihat bahwa model memiliki kecenderungan yang konsisten, yaitu lebih baik dalam memprediksi kelas tidak *churn* dibandingkan kelas *churn*. Hal ini ditunjukkan oleh nilai *True Negative* yang relatif tinggi, sedangkan nilai *True Positive* masih rendah.

Pada skenario pertama, nilai *False Negative* cukup tinggi, yang menunjukkan bahwa banyak data *churn* tidak berhasil dikenali oleh model. Kesalahan ini sangat penting dalam konteks *churn*, karena pelanggan yang seharusnya terdeteksi berpotensi *churn* justru tidak teridentifikasi oleh model.

Pada skenario kedua dan ketiga, meskipun terjadi peningkatan jumlah *True Positive*, namun nilai False Negative masih cukup besar. Hal ini menunjukkan bahwa model masih belum mampu mengenali pola *churn* secara konsisten.

Pada skenario keempat, meskipun jumlah data yang digunakan jauh lebih besar, pola kesalahan yang dihasilkan masih serupa. Model masih menghasilkan jumlah *False Negative* yang signifikan, yang menunjukkan bahwa peningkatan jumlah data tidak secara langsung memperbaiki kemampuan model dalam mendeteksi *churn*.

Dominasi kesalahan pada *False Negative* menunjukkan bahwa model cenderung lebih berhati-hati dalam memprediksi kelas *churn*, sehingga lebih sering mengklasifikasikan data sebagai tidak *churn*. Kemungkinan karena pola antar kelas tidak terlalu berbeda, sehingga model tidak cukup yakin untuk memprediksi *churn*. Walaupun sudah dibantu dengan SMOTE, ternyata kalau pola dasarnya memang mirip, hasilnya tetap tidak banyak berubah.

2) ROC AUC

ROC AUC digunakan sebagai metrik untuk mengukur kemampuan model dalam membedakan antara kelas *churn* dan tidak *churn* secara keseluruhan. Berdasarkan hasil pengujian, nilai ROC AUC pada seluruh skenario berada pada kisaran 50% hingga 54%. Nilai ini menunjukkan bahwa kemampuan model dalam melakukan klasifikasi masih sangat rendah dan cenderung mendekati prediksi acak.

Pada skenario pertama, nilai ROC AUC sebesar 54% menunjukkan bahwa model belum mampu menangkap pola yang relevan dalam data. Pada skenario kedua, ketiga, dan keempat, nilai ROC AUC hanya mengalami sedikit peningkatan yang menunjukkan bahwa kemampuan model dalam membedakan kedua kelas masih sangat terbatas.

Kurva ROC yang dihasilkan cenderung mendekati garis diagonal, yang menunjukkan bahwa model tidak memiliki batas pemisah yang jelas antara kelas *churn* dan tidak *churn*.

Kondisi ini menjadi indikasi kuat bahwa model memang tidak menemukan pola yang benar-benar bisa membedakan *churn* dan tidak *churn*. Jadi walaupun model sudah dilatih dengan berbagai cara, hasilnya tetap terbatas karena informasi yang dipelajari memang tidak cukup kuat untuk memisahkan kedua kelas.

4.8.3 Analisis Pengaruh Jumlah Data

Pada penelitian ini dilakukan pengujian menggunakan empat skenario jumlah data yang berbeda, yaitu 1000 data, 2000 data, 5000 data, dan seluruh dataset. Tujuan pengujian ini adalah untuk menganalisis pengaruh variasi jumlah data terhadap performa model *Random Forest* berdasarkan metrik *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *ROC AUC*. Hasil pengujian menunjukkan bahwa peningkatan jumlah data tidak selalu diikuti oleh peningkatan performa model secara konsisten pada seluruh metrik evaluasi. Beberapa metrik mengalami peningkatan pada skenario tertentu, namun mengalami penurunan pada skenario lainnya.

Untuk mengetahui apakah terdapat perbedaan performa yang signifikan antar skenario, dilakukan uji Friedman menggunakan nilai evaluasi yang diperoleh pada setiap *fold* hasil *Stratified K-Fold Cross Validation*. Nilai evaluasi tersebut terdiri atas *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *ROC AUC* pada masing-masing skenario pengujian. Uji Friedman dipilih karena digunakan untuk membandingkan performa lebih dari dua kelompok yang saling berhubungan, yaitu empat skenario pengujian dengan jumlah data yang berbeda. Hipotesis yang digunakan dalam pengujian ini adalah sebagai berikut:

- H0 : Tidak terdapat perbedaan performa yang signifikan antara Skenario 1, Skenario 2, Skenario 3, dan Skenario 4.
- H1 : Terdapat perbedaan performa yang signifikan antara Skenario 1, Skenario 2, Skenario 3, dan Skenario 4.

Hasil uji Friedman pada setiap metrik evaluasi ditunjukkan pada Tabel 4.12.

Tabel 4. 12 Tabel Uji Friedman

Metrik	**P-value**	**Keputusan**
Accuracy	0,0858	H0 gagal ditolak
Precision	0,0858	H0 gagal ditolak
Recall	0,3340	H0 gagal ditolak
F1-Score	0,2407	H0 gagal ditolak
ROC AUC	0,1218	H0 gagal ditolak

Berdasarkan Tabel 4.12, seluruh metrik evaluasi menghasilkan nilai *p-value* lebih besar dari tingkat signifikansi 0,05. Nilai *p-value* yang diperoleh berturut-turut sebesar 0,0858 untuk *Accuracy*, 0,0858 untuk *Precision*, 0,3340 untuk *Recall*, 0,2407 untuk *F1-Score*, dan 0,1218 untuk ROC AUC. Karena seluruh nilai *p-value* lebih besar dari 0,05, maka H0 gagal ditolak pada seluruh metrik evaluasi.

Hasil tersebut menunjukkan bahwa tidak terdapat perbedaan performa yang signifikan secara statistik antara penggunaan 1000 data, 2000 data, 5000 data, dan seluruh *dataset*. Dengan demikian, variasi jumlah data pada penelitian ini tidak memberikan pengaruh yang signifikan terhadap performa model *Random Forest* dalam mengklasifikasikan Renewal_Status pengguna platform bimbingan belajar daring. Hasil ini mengindikasikan bahwa peningkatan jumlah data saja belum cukup untuk meningkatkan performa model secara signifikan, sehingga kualitas dan relevansi fitur yang digunakan menjadi faktor yang perlu diperhatikan dalam proses klasifikasi.

4.9 Perbandingan Kinerja Model

Hasil pengujian menunjukkan bahwa peningkatan jumlah data dari 1000 menjadi 2000 tidak memberikan perubahan performa yang terlalu signifikan terhadap model. Meskipun terdapat sedikit peningkatan pada beberapa metrik seperti recall dan F1-score, namun peningkatan tersebut tidak konsisten dan tidak diikuti oleh peningkatan ROC AUC yang berarti.

Hal ini menunjukkan bahwa selain jumlah data, faktor lain seperti kualitas fitur dan karakteristik data juga memiliki peran penting dalam menentukan performa model. Dengan kata lain, penambahan data saja belum tentu mampu meningkatkan performa model secara optimal jika pola antar kelas masih sulit dibedakan.

Tabel 4. 13 Perbandingan Kinerja Model

Skenario	Jumlah Data	Accuracy	Recall	Precision	F1-Score	ROC AUC
Skenario 1	1000	71.00%	25.00%	22.50%	23.68%	54.15%
Skenario 2	2000	71.00%	17.95%	21.21%	19.44%	50.44%
Skenario 3	5000	70.90%	20.81%	23.30%	21.98%	53.23%
Skenario 4	Full Data	64.50%	26.40%	20.13%	22.84%	50.57%

Berdasarkan tabel hasil pengujian pada Tabel 4.13, terlihat bahwa perubahan jumlah data pada setiap skenario belum memberikan peningkatan performa yang signifikan terhadap model *Random Forest Classifier*. Meskipun terdapat perbedaan nilai pada beberapa metrik evaluasi, performa model secara keseluruhan masih cenderung berada pada kisaran yang sama.

Pada Skenario 1, model menghasilkan *Accuracy* sebesar 71.00% dengan *Recall* sebesar 25.00% dan *F1-Score* sebesar 23.68%. Hasil ini menunjukkan bahwa meskipun model memiliki tingkat akurasi yang cukup baik, kemampuan model dalam mendeteksi pengguna *churn* masih tergolong rendah.

Pada Skenario 2, nilai *Recall* dan *F1-Score* mengalami penurunan menjadi 17.95% dan 19.44%. Selain itu, nilai *ROC AUC* sebesar 50.44% menunjukkan bahwa kemampuan model dalam membedakan kelas *churn* dan *non-churn* masih terbatas.

Pada Skenario 3, nilai *ROC AUC* meningkat menjadi 53.23%, yang menunjukkan adanya sedikit peningkatan kemampuan model dalam membedakan kedua kelas. Namun demikian, nilai *Recall* dan *F1-Score* masih relatif rendah sehingga performa model belum stabil.

Pada Skenario 4 dengan penggunaan seluruh data (*full dataset*), nilai *Recall* meningkat menjadi 26.40%, yang menunjukkan bahwa model menjadi lebih sensitif dalam mendeteksi data *churn*. Akan tetapi, nilai *Accuracy* menurun menjadi 64.50% dan nilai *ROC AUC* tetap berada di sekitar 50%, sehingga kemampuan model secara keseluruhan masih belum mengalami peningkatan yang signifikan.

Secara keseluruhan, hasil pengujian menunjukkan bahwa penambahan jumlah data tidak secara langsung meningkatkan performa model. Hal ini mengindikasikan bahwa karakteristik data dan kualitas fitur yang digunakan memiliki pengaruh yang lebih besar terhadap kemampuan model dalam melakukan klasifikasi.

4.10 Integrasi Islam

Dalam penelitian ini, penerapan algoritma *Random Forest* digunakan untuk memprediksi *Renewal_Status*, yaitu kondisi apakah pelanggan akan melanjutkan (*renewal*) atau menghentikan layanan (*churn*). Proses prediksi ini tidak hanya berkaitan dengan aspek teknis, tetapi juga berdampak pada pengambilan keputusan terhadap pelanggan.

Oleh karena itu, penggunaan teknologi berbasis data seperti machine learning perlu dikaitkan dengan nilai-nilai Islam, khususnya dalam hal keadilan, kejujuran, kehati-hatian, serta tanggung jawab dalam pengelolaan informasi. Integrasi nilai Islam dalam penelitian ini dikaji melalui tiga dimensi utama, yaitu *Muamalah Ma'a Allah*, *Muamalah Ma'a An-Nas*, dan *Muamalah Ma'al Bi'ah*.

- ***Muamalah Ma'a Allah***

Muamalah Ma'a Allah merupakan hubungan manusia dengan Allah SWT yang menjadi landasan dalam setiap aktivitas, termasuk dalam penggunaan ilmu pengetahuan dan teknologi. Dalam penelitian ini, proses analisis data dan pemodelan untuk memprediksi *Renewal_Status* harus dilakukan dengan penuh kehati-hatian dan berdasarkan kebenaran data.

Allah SWT berfirman dalam Al-Qur'an Surah Al-Hujurat ayat 6:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ ۝٦

"Wahai orang-orang yang beriman, jika seorang fasik datang kepadamu membawa berita penting, maka telitilah kebenarannya agar kamu tidak mencelakakan suatu kaum karena ketidaktahuan(-mu) yang berakibat kamu menyesali perbuatanmu itu." (QS. Al-Hujurat ayat 6).

Menurut M. Quraish Shihab dalam *Tafsir Al-Misbah*, ayat ini mengajarkan prinsip *tabayyun*, yaitu melakukan klarifikasi dan verifikasi terhadap setiap informasi sebelum dijadikan dasar pengambilan keputusan. Sikap tersebut bertujuan untuk menghindari kesalahan yang dapat merugikan pihak lain akibat informasi yang tidak benar.

Nilai *tabayyun* tersebut sejalan dengan penelitian ini, di mana sebelum proses pemodelan dilakukan, data terlebih dahulu *preprocessing data*. Tahapan tersebut bertujuan memastikan bahwa data yang digunakan telah melalui proses verifikasi sehingga hasil prediksi Renewal Status dapat dipertanggungjawabkan secara ilmiah maupun moral.

- ***Muamalah Ma'a An-Nas***

Muamalah Ma'a An-Nas berkaitan dengan hubungan manusia dengan sesama, khususnya dalam konteks pengambilan keputusan yang berdampak pada pihak lain. Dalam penelitian ini, hasil prediksi *Renewal_Status* berkaitan langsung dengan kemungkinan pelanggan untuk tetap menggunakan layanan atau berhenti (*churn*), sehingga keputusan yang diambil berdasarkan hasil tersebut harus memperhatikan prinsip keadilan.

Allah SWT berfirman dalam Al-Qur'an Surah An-Nahl ayat 90:

اِنَّ اللّٰهَ يَأْمُرُ بِالْعَدْلِ وَالْاِحْسَانِ وَاِيْتَآئِ ذِى الْقُرْبٰى وَيَنْهٰى عَنِ الْفَحْشَآءِ وَالْمُنْكَرِ وَالْبَغْيِ يَعِظُكُمْ لَعَلَّكُمْ تَذَكَّرُوْنَ ﴿٩٠﴾

"Sesungguhnya Allah menyuruh berlaku adil, berbuat kebajikan, dan memberikan bantuan kepada kerabat. Dia (juga) melarang perbuatan keji, kemungkaran, dan permusuhan. Dia memberi pelajaran kepadamu agar kamu selalu ingat." (QS. An-Nahl ayat 90).

Menurut M. Quraish Shihab dalam *Tafsir Al-Misbah*, perintah berlaku adil pada ayat ini tidak hanya berlaku dalam kehidupan sosial, tetapi juga dalam setiap bentuk pengambilan keputusan yang berkaitan dengan hak orang lain. Keadilan menuntut agar setiap keputusan didasarkan pada pertimbangan yang objektif dan tidak merugikan pihak tertentu.

Dalam penelitian ini, hasil prediksi Renewal Status tidak digunakan untuk mendiskriminasi pelanggan yang diprediksi *churn*, melainkan sebagai dasar bagi perusahaan untuk memberikan pelayanan yang lebih baik melalui strategi retensi pelanggan. Dengan demikian, penerapan model *Random Forest* tetap mencerminkan nilai keadilan dan kemaslahatan bagi pengguna layanan.

- ***Muamalah Ma'al Bi'ah***

Muamalah Ma'al Bi'ah berkaitan dengan tanggung jawab manusia dalam menjaga lingkungan, termasuk lingkungan digital dan sistem informasi. Dalam penelitian ini, data pelanggan yang digunakan merupakan amanah yang harus dikelola secara benar dan bertanggung jawab.

Allah SWT berfirman dalam Al-Qur'an Surah Al-Isra ayat 36:

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهٖ عِلْمٌۗ اِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ اُولٰۤىِٕكَ كَانَ عَنْهُ مَسْـُٔوْلًا ﴿٣٦﴾

"Janganlah engkau mengikuti sesuatu yang tidak kauketahui. Sesungguhnya pendengaran, penglihatan, dan hati nurani, semua itu akan diminta pertanggungjawabannya." (QS. Al-Isra ayat 36).

Menurut M. Quraish Shihab dalam *Tafsir Al-Misbah*, ayat ini menegaskan bahwa seseorang tidak boleh mengambil keputusan tanpa didasarkan pada pengetahuan yang benar karena setiap bentuk pengetahuan dan tindakan akan dimintai pertanggungjawaban.

Prinsip tersebut diterapkan dalam penelitian ini melalui pengelolaan data secara bertanggung jawab, mulai dari proses pengumpulan, *preprocessing*, hingga pembangunan model *Random Forest*. Dengan demikian, hasil prediksi yang dihasilkan diharapkan memiliki tingkat keakuratan yang baik sehingga dapat digunakan sebagai dasar pengambilan keputusan secara bertanggung jawab.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, algoritma *Random Forest Classifier* dapat digunakan untuk melakukan klasifikasi status pengguna (*churn* dan *non-churn*) pada platform bimbingan belajar daring. Hasil pengujian menunjukkan bahwa performa model masih belum optimal, yang ditunjukkan oleh nilai *Recall*, *Precision*, *F1-Score*, dan *ROC AUC* yang relatif rendah pada seluruh skenario pengujian. Skenario terbaik diperoleh pada penggunaan 1000 data dengan nilai *Accuracy* sebesar 71.00%, *Recall* sebesar 25.00%, *Precision* sebesar 22.50%, *F1-Score* sebesar 23.68%, dan *ROC AUC* sebesar 54.15% menunjukkan bahwa kemampuan model dalam membedakan kelas *churn* dan *non-churn* mampu melakukan klasifikasi namun belum maksimal.

Berdasarkan analisis pengaruh variasi jumlah data terhadap performa model, diperoleh hasil bahwa perbedaan jumlah data tidak memberikan pengaruh yang signifikan terhadap performa *Random Forest*. Hal ini dibuktikan melalui uji Friedman yang menghasilkan nilai statistik sebesar 4,20 dengan *p-value* sebesar 0,2407 (> 0,05), sehingga tidak terdapat perbedaan performa yang signifikan secara statistik antara penggunaan 1000 data, 2000 data, 5000 data, dan seluruh *dataset*. Meskipun telah dilakukan *preprocessing*, normalisasi, penyeimbangan data menggunakan SMOTE, serta *tuning hyperparameter*, model masih mengalami kesulitan dalam mengenali pola pengguna *churn*. Hasil tersebut mengindikasikan bahwa fitur yang digunakan dalam penelitian ini belum mampu memberikan

pemisahan kelas yang cukup baik sehingga performa klasifikasi model masih belum optimal.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian selanjutnya.

Pertama, penelitian selanjutnya dapat melakukan pengembangan fitur (feature engineering) agar diperoleh fitur yang lebih representatif dalam membedakan antara kelas *churn* dan tidak *churn*. Dengan fitur yang lebih informatif, diharapkan model dapat menangkap pola data dengan lebih baik. Kedua, penelitian selanjutnya dapat mencoba menggunakan algoritma lain sebagai pembanding untuk melihat kemungkinan peningkatan performa model. Hal ini dapat memberikan gambaran metode yang paling sesuai untuk kasus yang diteliti. Ketiga, dapat dilakukan eksplorasi data yang lebih mendalam untuk memahami karakteristik data secara lebih baik, sehingga proses pemodelan dapat dilakukan dengan lebih optimal.

DAFTAR PUSTAKA

Aguilar-Ruiz, J. S. (2024). *Class-specific feature selection for classification explainability* (arXiv:2411.01204). arXiv. https://doi.org/10.48550/arXiv.2411.01204

Aini, N., Arif, M., Agustin, I. T., & Toyibah, Z. B. (2024). Implementasi Algoritma Random Forest untuk Klasifikasi Bidang MSIB di Prodi Pendidikan Informatika. *Jurnal Informatika*, *11*(1), 11–16. https://doi.org/10.31294/inf.v11i1.20637

Aisyah, A., Setiawan, A., & Rozak, R. W. A. (2025). *Development of Information Technology Application in Education in Indonesia*. *23*(1).

Akpen, C. N., Asaolu, S., Atobatele, S., Okagbue, H., & Sampson, S. (2024). Impact of online learning on student's performance and engagement: A systematic review. *Discover Education*, *3*(1), 205. https://doi.org/10.1007/s44217-024-00253-0

Assidiqi, M. H., & Sumarni, W. (2020). *Pemanfaatan Platform Digital di Masa Pandemi Covid-19*.

Aufa, M. J., & Qoiriah, A. (2023). Analisis Sentimen Pengguna Platform Belajar Online Coursera menggunakan Random Forest dengan Metode Ekstraksi Fitur Word2vec. *Journal of Informatics and Computer Science (JINACS)*, 244–255. https://doi.org/10.26740/jinacs.v4n02.p244-255

Azis, H., Purnawansyah, P., Fattah, F., & Putri, I. P. (2020). Performa Klasifikasi K-NN dan Cross Validation Pada Data Pasien Pengidap Penyakit Jantung. *ILKOM Jurnal Ilmiah*, *12*(2), 81–86. https://doi.org/10.33096/ilkom.v12i2.507.81-86

Azmi, A. F., & Voutama, A. (2024). *Prediksi Churn Nasabah Bank Menggunakan Klasifikasi Random Forest dan Decision Tree Dengan Evaluasi Confusion Matrix. 13*(1).

Bhoite, S. N., Gadekar, V. D., Kapadnis, S. V., & Ghuge, P. R. (2023). Churn Prediction using Machine Learning Models. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 2233–2236. https://doi.org/10.22214/ijraset.2023.52101

Dewi, N. K., Syafitri, U. D., & Mulyadi, S. Y. (2019). *Penerapan Metode Random Forest Dalam Driver Analysis*.

Flood, J. (2020). *Read all about it: Online learning facing 80% attrition rates*.

Glandorf, D., Lee, H. R., Orona, G. A., Pumptow, M., Yu, R., & Fischer, C. (2024). Temporal and Between-Group Variability in College Dropout Prediction. *Proceedings of the 14th Learning Analytics and Knowledge Conference*, 486–497. https://doi.org/10.1145/3636555.3636906

Hafid, H. (2023). *Penerapan K-Fold Cross Validation untuk Menganalisis Kinerja Algoritma K-Nearest Neighbor pada Data Kasus Covid-19 di Indonesia*.

Handoko, C. B., & Aditya, C. S. K. (2025). Penerapan Teknik SMOTE Dalam Mengatasi Imbalance Data Penyakit Diabetes Menggunakan Algoritma ANN. *Smart Comp: Jurnalnya Orang Pintar Komputer*, *14*(1). https://doi.org/10.30591/smartcomp.v14i1.7045

Imani, M., Joudaki, M., Beikmohammadi, A., & Arabnia, H. (2025). Customer Churn Prediction: A Systematic Review of Recent Advances, Trends, and Challenges in Machine Learning and Deep Learning. *Machine Learning and Knowledge Extraction*, *7*(3), 105. https://doi.org/10.3390/make7030105

Imron, M. (2025). *Implementasi Model Prediksi Churn Pelanggan Menggunakan Algoritma Random Forest Pada Website Industri Telekomunikasi*. *1*(1).

Istiqamah, N., & Rijal, M. (2024). *Klasifikasi Ulasan Konsumen Menggunakan Random Forest dan SMOTE*. *5*(1).

Jang, K., Kim, J., & Yu, B. (2021). On Analyzing Churn Prediction in Mobile Games. *2021 6th International Conference on Machine Learning Technologies*, 20–25. https://doi.org/10.1145/3468891.3468895

Keats, J. (n.d.). *Tutoring Website Data* [Dataset]. Kaggle. https://www.kaggle.com/datasets/jessekeats/tutoring-website-data

Kurniawan, M. R., Sabrina, P. N., & Ilyas, R. (2023). *Prediksi Customer Churn pada PErusahaan telekomunikasi Menggunakan Algoritma C4.5 Berbasis Particle Swarm Optimization*. *7*(5).

Loog, M. (2017). *Supervised Classification: Quite a Brief Overview* (arXiv:1710.09230). arXiv. https://doi.org/10.48550/arXiv.1710.09230

Maan, J., & Maan, H. (2023). *Customer Churn Prediction Model using Explainable Machine learning*. *11*(1).

Marcellina, M., & Mukhlason, A. (2024). Analisis Prediktif Churn untuk Meningkatkan Tingkat Retensi Pelanggan pada Perusahaan SaaS Menggunakan Machine Learning. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, *6*(1), 21–32. https://doi.org/10.28926/ilkomnika.v6i1.598

Orji, F. A., & Vassileva, J. (2022). *Machine Learning Approach for Predicting Students' Academic Performance and Study Strategies based on their Motivation.*

Pamina, J., Raja, J. B., Bama, S. S., Soundarya, S., Sruthi, M. S., Kiruthika, S., Aiswaryadevi, V. J., & Priyanka, G. (2019). An Effective Classifier for Predicting Churn in Telecommunication. *Control Systems*, *11*.

Perdana, A., & Farhana, N. A. (2022). *Penerapan Majority Vote Method Dalam Penentuan Pembobotan Pada Metode Weighted Product (WP) Pada Pemeringkatan Kampus di Kota Medan*.

Ramadhon, R. N., Ogi, A., Agung, A. P., Putra, R., Febrihartina, S. S., & Firdaus, U. (2024). Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank. *Karimah Tauhid*, *3*(2), 1860–1874. https://doi.org/10.30997/karimahtauhid.v3i2.11952

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. *IEEE Access*, *7*, 60134–60149. https://doi.org/10.1109/ACCESS.2019.2914999

Verbeke, W., Martens, D., Mues, C., & Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, *38*(3), 2354–2364. https://doi.org/10.1016/j.eswa.2010.08.023

Wakhidah, L. N., Zyen, A. K., & Wahono, B. B. (2025). Evaluation of Telecommunication Customer Churn Classification with SMOTE Using Random Forest and XGBoost Algorithms. *Journal of Applied Informatics and Computing*, *9*(1), 89–95. https://doi.org/10.30871/jaic.v9i1.8740

Yan, A., Lee, M. J., & Ko, A. J. (2017). Predicting Abandonment in Online Coding Tutorials. *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 191–199. https://doi.org/10.1109/VLHCC.2017.8103467

Yuliani, S. D., & Kartikasari, P. (2025). *Klasifikasi Pelanggan Churn Pada Kegiatan Servis dan PEeN-ISJSUN:A30L47A-65N69 Sparepart Dengan Metode Random Forest*. *2*(2).