

BAB II

KAJIAN LITERATUR

2.1 *Customer Churn*

Customer churn merupakan fenomena ketika pelanggan menghentikan hubungan bisnis dengan perusahaan dengan tidak lagi menggunakan layanan atau membeli produk yang ditawarkan. Dalam konteks *Customer Relationship Management (CRM)*, *churn* dipandang sebagai kebalikan dari loyalitas pelanggan dan menjadi perhatian utama dalam strategi retensi. Pada sektor *e-commerce*, *churn* umumnya diidentifikasi berdasarkan perilaku transaksi, yaitu ketika pelanggan tidak lagi melakukan pembelian dalam periode waktu tertentu sehingga kontribusinya terhadap pendapatan perusahaan berhenti [15].

Berdasarkan karakteristik hubungannya, *churn* dapat dibedakan menjadi *churn* kontraktual dan non-kontraktual. *Churn* kontraktual terjadi pada industri berbasis langganan, di mana pelanggan secara eksplisit membatalkan layanan. Sebaliknya, *churn* non-kontraktual, yang umum pada *e-commerce*, terjadi tanpa pemberitahuan resmi sehingga identifikasinya ditentukan melalui batas waktu sejak transaksi terakhir. Dalam konteks ini, kehilangan pelanggan dapat bersifat permanen maupun sementara, di mana sebagian pelanggan berpotensi kembali melakukan transaksi di masa mendatang [16].

Faktor-faktor yang memengaruhi *churn* umumnya mencakup aspek perilaku, demografi, dan persepsi pelanggan. Aspek perilaku meliputi pola transaksi seperti frekuensi pembelian, jumlah item, dan nilai transaksi. Faktor demografi dan lokasi juga dapat memengaruhi loyalitas pelanggan, sementara persepsi pelanggan terhadap layanan, yang tercermin melalui ulasan dan tingkat kepuasan, turut berkontribusi terhadap keputusan untuk tetap bertahan atau meninggalkan *platform* [16].

Tingginya tingkat *churn* berdampak signifikan terhadap keberlanjutan bisnis karena berkurangnya pelanggan aktif secara langsung menurunkan pendapatan dan meningkatkan tekanan persaingan. Oleh karena itu, kemampuan untuk mengidentifikasi dan memprediksi pelanggan yang berisiko *churn* menjadi langkah strategis dalam mendukung pengelolaan hubungan pelanggan serta menjaga stabilitas kinerja perusahaan *e-commerce* [17].

2.2 *Machine Learning*

Machine learning merupakan pendekatan dalam kecerdasan buatan yang menekankan pada peningkatan kinerja program komputer melalui pengalaman dalam

menyelesaikan suatu tugas berdasarkan ukuran kinerja tertentu. Pendekatan ini bertujuan mengotomatisasi pembangunan model analitis untuk menjalankan tugas-tugas kognitif, seperti klasifikasi, regresi, maupun klasterisasi, dengan memanfaatkan data sebagai sumber pembelajaran. Proses tersebut dilakukan melalui penerapan algoritma yang belajar secara iteratif dari data pelatihan yang spesifik terhadap permasalahan, sehingga sistem mampu mengidentifikasi pola dan keteraturan tanpa harus diprogram secara eksplisit untuk setiap aturan. Kemampuan ini menjadikan *machine learning* efektif dalam menangani data berdimensi tinggi serta dalam menghasilkan keputusan yang konsisten berdasarkan pengalaman komputasi sebelumnya [18].

Secara konseptual, teknik atau pendekatan dalam *machine learning* diklasifikasikan menjadi empat jenis, yaitu [19]:

a. *Supervised Learning*

Supervised learning bertujuan mempelajari suatu fungsi pemetaan dari input ke output berdasarkan data yang telah diberi label. Proses ini menggunakan *dataset* pelatihan yang berisi fitur sebagai *input* dan label (target) sebagai *output*, yang digunakan model untuk mempelajari pola hubungan di antara keduanya. Setelah proses pelatihan selesai, model dapat digunakan untuk memprediksi target pada data baru yang belum pernah diamati sebelumnya.

Dua tugas utama dalam *supervised learning* adalah klasifikasi dan regresi, di mana klasifikasi berfokus pada pengelompokan data ke dalam kategori tertentu, sedangkan regresi digunakan untuk memperkirakan nilai kontinu. Salah satu bentuk klasifikasi yang umum digunakan adalah klasifikasi *biner*, yaitu tugas prediksi dengan dua label kelas, seperti “ya” dan “tidak”. Prediksi *churn* pelanggan dalam penelitian ini termasuk dalam klasifikasi *biner* karena bertujuan mengelompokkan pelanggan ke dalam dua kategori, yaitu pelanggan *churn* (1) dan pelanggan tidak *churn* (0).

b. *Unsupervised Learning*

Berbeda dengan *supervised*, *unsupervised learning* bertujuan mengidentifikasi pola atau struktur dalam data tanpa menggunakan label. Pada metode ini, data pelatihan hanya terdiri dari variabel input, sehingga model diarahkan untuk menemukan keteraturan, seperti pengelompokan objek yang memiliki karakteristik serupa atau penyederhanaan representasi data melalui reduksi dimensi. Sebagai contoh, teknik ini dapat digunakan untuk mengelompokkan data transaksi berdasarkan kemiripan pola pembelian guna mengidentifikasi kelompok perilaku yang berbeda dalam suatu *dataset* [18].

c. *Semi-supervised Learning*

Semi-supervised learning merupakan teknik *machine learning* yang memanfaatkan kombinasi data berlabel dan tidak berlabel dalam proses pelatihannya. Metode ini relevan dalam kondisi ketika data berlabel terbatas, sementara data tanpa label tersedia dalam jumlah besar, sehingga informasi tambahan dari data tidak berlabel dapat membantu meningkatkan kinerja model. Tujuan utama pendekatan ini adalah menghasilkan prediksi yang lebih baik dibandingkan model yang hanya dilatih menggunakan data berlabel.

d. *Reinforcement Learning*

Reinforcement learning beroperasi melalui skema hadiah (*reward*) dan hukuman (*punishment*) untuk mengekstraksi wawasan *environment* sebagai landasan pengambilan keputusan yang memaksimalkan hasil sekaligus meminimalkan risiko. Meskipun banyak digunakan dalam optimalisasi sistem kompleks seperti robotika dan logistik, implementasi teknik ini tidak direkomendasikan untuk penyelesaian masalah yang sederhana.

Dalam praktiknya, algoritma pada setiap teknik *machine learning* mengolah data untuk mempelajari pola yang berkaitan dengan individu, proses bisnis, transaksi, maupun peristiwa tertentu. Namun, pemilihan algoritma yang tepat menjadi sangat penting mengingat setiap algoritma memiliki tujuan dan karakteristik yang berbeda. Bahkan, algoritma yang berada dalam kategori pembelajaran yang sama dapat menghasilkan performa yang bervariasi tergantung pada karakteristik data yang digunakan. Oleh karena itu, kesesuaian antara algoritma dan sifat data menjadi faktor penting dalam keberhasilan penerapan *machine learning* pada suatu permasalahan [19].

2.3 Class Imbalance

Class imbalance merupakan permasalahan dalam pembelajaran mesin yang muncul ketika distribusi jumlah data tidak proporsional. Dalam konteks klasifikasi *biner*, kondisi ini terjadi ketika satu kelas memiliki jumlah sampel yang jauh lebih besar sebagai kelas mayoritas, sementara yang lainnya memiliki jumlah jauh lebih sedikit sebagai kelas minoritas. Permasalahan ini umum terjadi dalam industri *e-commerce*, di mana jumlah pelanggan yang melakukan *churn* jauh lebih sedikit dibandingkan pelanggan yang tetap aktif, sehingga menghasilkan distribusi data yang tidak seimbang. Kondisi tersebut menyebabkan *dataset* didominasi oleh kelas mayoritas, yaitu pelanggan setia [20,21].

Pada kondisi data yang sangat tidak seimbang, model klasifikasi dapat memperoleh nilai akurasi yang tinggi hanya dengan selalu memprediksi kelas mayoritas tanpa benar-

benar mempelajari perbedaan karakteristik antar kelas. Sebagai ilustrasi, apabila sebagian besar data terdiri dari orang yang memiliki jantung normal dan hanya sebagian kecil merupakan orang dengan gangguan jantung, model dapat mencapai tingkat akurasi yang tampak tinggi dengan mengklasifikasikan seluruh data ke dalam kelas dominan tersebut. Namun, pendekatan ini tidak mencerminkan kemampuan diskriminatif yang sesungguhnya, karena model tidak benar-benar mengidentifikasi pola yang membedakan kedua kelas. Akibatnya, penderita gangguan jantung gagal diamati [22].

Model pembelajaran mesin pada umumnya dikembangkan dengan asumsi distribusi kelas yang relatif seimbang, sehingga pada *dataset* tidak seimbang model cenderung menghasilkan bias terhadap kelas mayoritas, sehingga kinerja model dalam mengidentifikasi kelas minoritas menjadi kurang optimal meskipun nilai akurasi keseluruhan terlihat tinggi. Dengan demikian, evaluasi yang hanya berfokus pada akurasi berpotensi menyesatkan, karena tidak mampu menggambarkan performa model dalam mendeteksi kelas minoritas secara tepat. Oleh karena itu, evaluasi model seharusnya difokuskan pada kemampuan untuk mendeteksi kasus langka atau abnormal yang jarang terjadi, agar hasil prediksi lebih bermakna [21].

Ketidakseimbangan distribusi kelas tidak hanya memengaruhi hasil evaluasi model, tetapi juga memengaruhi proses pembelajaran setiap algoritma klasifikasi. Pada algoritma berbasis *bagging* seperti *Random Forest*, mayoritas pohon keputusan dibangun menggunakan data yang masih didominasi oleh kelas mayoritas sehingga proses *majority voting* cenderung menghasilkan prediksi ke arah kelas tersebut [8]. Sementara itu, pada algoritma *boosting* seperti *GBM* dan *XGBoost*, proses optimasi fungsi kerugian lebih banyak dipengaruhi oleh kesalahan pada kelas mayoritas karena jumlah observasinya lebih besar [7]. Akibatnya, model cenderung kurang sensitif dalam mempelajari karakteristik kelas minoritas sehingga meningkatkan kemungkinan terjadinya *false negative*, yaitu pelanggan *churn* yang diprediksi sebagai tidak *churn*. Kondisi tersebut menyebabkan nilai *recall* pada kelas minoritas menjadi rendah meskipun akurasi keseluruhan tampak tinggi. Oleh karena itu, penanganan *class imbalance* menjadi tahapan penting agar model mampu mempelajari pola dari kedua kelas secara lebih proporsional dan menghasilkan kemampuan generalisasi yang lebih baik.

2.4 Ensemble Learning

Ensemble learning merupakan pendekatan dalam *machine learning (supervised classification)* yang membangun beberapa model dasar (*base learner*) dan menggabungkan

hasil prediksinya untuk memperoleh kinerja yang lebih baik dibandingkan penggunaan satu model secara individual [23]. Pendekatan ini didasarkan pada pemahaman bahwa setiap model memiliki keterbatasan dan berpotensi menghasilkan kesalahan akibat bias maupun variansi. Dengan mengombinasikan beberapa model, *ensemble learning* berupaya memanfaatkan keunggulan masing-masing model sehingga kesalahan prediksi dapat diminimalkan. Secara teoretis, strategi ini mampu meningkatkan kemampuan generalisasi model terhadap data baru karena keputusan akhir tidak bergantung pada satu struktur pembelajaran saja. Dalam konteks klasifikasi, mekanisme penggabungan prediksi dapat dilakukan melalui teknik seperti *voting* [24,25].

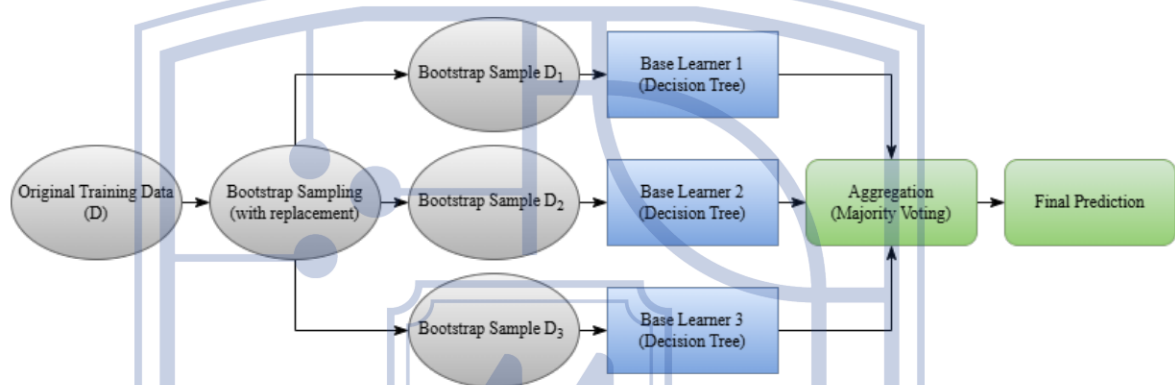
Secara umum, metode *ensemble* dapat diklasifikasikan ke dalam pendekatan paralel dan sekuensial berdasarkan cara pembentukan model dasarnya. Pada pendekatan paralel, beberapa model dilatih secara independen menggunakan data yang berbeda, kemudian hasil prediksinya digabungkan melalui mekanisme agregasi tertentu. Strategi ini bertujuan mendorong keberagaman antar model sehingga kesalahan prediksi dapat saling dikompensasi, sebagaimana diterapkan dalam metode *bagging*. Sebaliknya, pendekatan sekuensial membangun model secara bertahap, di mana setiap model baru dilatih untuk memperbaiki kesalahan model sebelumnya, seperti pada algoritma *boosting*. Selain berdasarkan mekanisme pelatihan, *ensemble* juga dapat dibedakan menjadi homogen dan heterogen, tergantung pada apakah model dasar yang digunakan berasal dari algoritma yang sama atau berbeda. Keberhasilan suatu *ensemble* sangat dipengaruhi oleh dua faktor utama, yaitu akurasi dan keberagaman model dasar. Akurasi berkaitan dengan kemampuan generalisasi model terhadap data yang belum pernah dilihat, sedangkan keberagaman mengacu pada perbedaan pola kesalahan antar model sehingga tidak menghasilkan prediksi yang identik [25].

Dalam literatur *machine learning*, pendekatan *ensemble learning* umumnya diklasifikasikan ke dalam tiga strategi utama, yaitu *bagging*, *boosting*, dan *stacking*, yang masing-masing memiliki mekanisme berbeda dalam membangun dan menggabungkan model untuk meningkatkan kinerja prediksi, seperti berikut ini [24,26]:

a. *Bagging*

Bagging (bootstrap aggregating) merupakan metode *ensemble* yang membangun beberapa model dasar menggunakan variasi sampel data pelatihan yang dihasilkan melalui teknik *bootstrap sampling*. Setiap model dilatih secara independen pada replika data yang berbeda, sehingga tercipta keberagaman antar model dalam *ensemble*. Prediksi akhir kemudian diperoleh melalui mekanisme agregasi, seperti *majority voting* pada

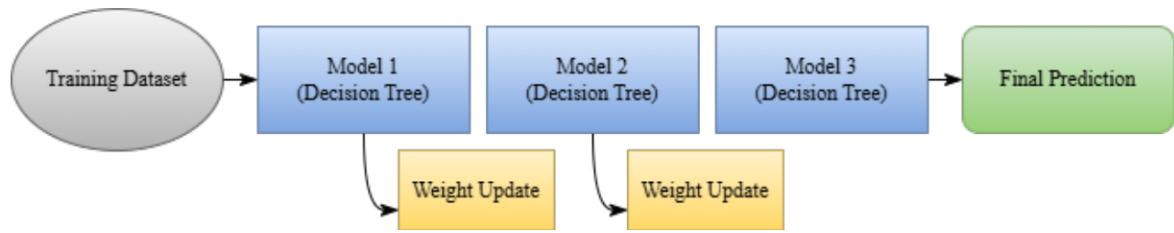
tugas klasifikasi atau perataan nilai pada regresi. Pendekatan ini bertujuan untuk mengurangi variansi model dan meningkatkan stabilitas prediksi tanpa meningkatkan bias secara signifikan. Salah satu algoritma yang paling representatif dari metode *bagging* adalah *Random Forest*, yang mengombinasikan *bootstrap sampling* dengan pemilihan fitur secara acak. Selain itu, variasi lain seperti *Extra Trees* dan beberapa *meta-estimator* juga dikembangkan dalam kerangka *bagging*. Secara umum, *bagging* efektif diterapkan pada model yang memiliki variansi tinggi, seperti *decision tree*. Berikut merupakan ilustrasi mekanisme *bagging* dalam metode *ensemble*.



Gambar 2.1 Mekanisme *Bagging* pada *Ensemble Learning*

b. *Boosting*

Boosting merupakan metode *ensemble* yang membangun model secara bertahap dan berurutan, di mana setiap model baru dilatih untuk memperbaiki kesalahan prediksi yang dihasilkan oleh model sebelumnya. Proses ini dilakukan dengan memberikan perhatian lebih besar pada data pelatihan yang sebelumnya salah diklasifikasikan, sehingga pembelajaran menjadi lebih adaptif terhadap kesalahan. Hasil akhir diperoleh melalui kombinasi prediksi seluruh model dasar dengan pembobotan tertentu sesuai dengan kontribusi kinerjanya. Berbeda dengan *bagging* yang berfokus pada pengurangan variansi, *boosting* lebih diarahkan untuk mengurangi bias model. Beberapa algoritma yang termasuk dalam pendekatan ini antara lain *AdaBoost*, *Gradient Boosting (GBM)*, *XGBoost*, *LightGBM*, dan *CatBoost*. Metode *boosting* dikenal memiliki kemampuan prediktif yang tinggi, namun memerlukan pengaturan parameter yang cermat untuk menghindari *overfitting*. Dalam konteks klasifikasi seperti prediksi *churn*, *boosting* sering digunakan karena kemampuannya menangkap pola kompleks dan non-linear. Berikut merupakan ilustrasi mekanisme *boosting* dalam metode *ensemble*.



Gambar 2.2 Mekanisme *Boosting* pada *Ensemble Learning*

c. *Stacking*

Stacking atau *stacked generalization* merupakan metode *ensemble* yang menggabungkan berbagai jenis model dasar yang dibangun menggunakan algoritma yang berbeda. Berbeda dengan *bagging* dan *boosting*, *stacking* tidak hanya mengandalkan satu jenis algoritma, melainkan memanfaatkan keberagaman struktur model untuk menghasilkan kesalahan prediksi yang tidak berkorelasi. Dalam pendekatan ini, model-model tingkat pertama (*base learner*) dilatih menggunakan *dataset* yang sama, kemudian prediksinya dijadikan *input* bagi model tingkat kedua yang disebut *meta-learner*. *Meta-learner* bertugas mempelajari cara terbaik dalam mengombinasikan prediksi dari model tingkat pertama untuk menghasilkan *output* akhir. Strategi ini memungkinkan integrasi keunggulan berbagai algoritma dalam satu sistem prediksi. Namun demikian, *stacking* memiliki kompleksitas implementasi yang lebih tinggi dibandingkan *bagging* dan *boosting*. Oleh karena itu, metode ini umumnya digunakan ketika diperlukan peningkatan performa yang lebih lanjut melalui kombinasi model yang heterogen.

2.5 *Random Forest*

Random Forest merupakan salah satu algoritma dalam pendekatan *bagging* pada *ensemble learning* yang dibangun melalui penggabungan sejumlah *decision tree* untuk meningkatkan kinerja klasifikasi. Metode ini dikembangkan untuk mengatasi keterbatasan penggunaan satu pohon keputusan dalam memperoleh solusi yang optimal. Dalam *random forest*, setiap pohon dilatih menggunakan *subset* data pelatihan yang dihasilkan melalui teknik *bootstrap* serta hanya memanfaatkan sebagian variabel yang dipilih secara acak. Meskipun pembatasan pada data dan variabel dapat menurunkan kinerja masing-masing pohon secara individual, pendekatan ini menghasilkan korelasi yang lebih rendah antar pohon sehingga meningkatkan keandalan prediksi secara keseluruhan. Prediksi akhir ditentukan melalui mekanisme *majority voting* pada tugas klasifikasi. Dengan banyaknya pohon yang dibangun dalam model, proses pemangkasan individu tidak lagi diperlukan [23].

Pendekatan ini menjadikan *random forest* sebagai algoritma yang mampu meningkatkan akurasi sekaligus mengurangi varians model [27].

Berbagai studi empiris menunjukkan bahwa *random forest* memiliki tingkat ketahanan dan fleksibilitas yang tinggi dalam berbagai domain aplikasi. Algoritma ini dikenal efektif dalam menangani data berdimensi tinggi serta tetap relatif tahan terhadap *overfitting*. Selain itu, *random forest* mampu memproses variabel kategorikal maupun kontinu dalam satu kerangka model yang sama. Pengembangan lebih lanjut dalam konstruksi pohon dan metode pemilihan fitur juga dilakukan untuk mengoptimalkan performa serta mengurangi kompleksitas komputasi. Karakteristik tersebut memperkuat posisi *random forest* sebagai salah satu metode *ensemble* yang kuat dalam menghadapi permasalahan klasifikasi yang kompleks. Dalam konteks prediksi *churn*, kemampuan ini relevan mengingat data pelanggan umumnya berskala besar dan memiliki banyak variabel [28].

Meskipun memiliki berbagai keunggulan, *random forest* juga memiliki beberapa keterbatasan. Model ini dapat mengalami perubahan hasil prediksi apabila terjadi perubahan kecil pada data pelatihan. Selain itu, semakin banyak pohon yang dibangun, semakin tinggi pula kompleksitas perhitungan yang diperlukan. Interpretasi model juga cenderung lebih sulit dibandingkan dengan satu pohon keputusan tunggal. Dari sisi implementasi, algoritma ini memproses data dalam bentuk numerik, sehingga variabel tertentu perlu disesuaikan sebelum digunakan. Oleh karena itu, pemilihan parameter dan pengelolaan data menjadi aspek penting dalam penerapan *random forest* [23]. Formulasi matematis prediksi pada model *random forest* berbasis *majority voting* pada kasus klasifikasi dapat dilihat pada persamaan berikut [29].

Prediksi *Random Forest* (Klasifikasi):

$$\hat{Y} = \text{mode}\{T_1(x), T_2(x), \dots, T_B(x)\} \dots\dots\dots (1)$$

Dimana:

$\hat{Y}$ = prediksi akhir model

$T_B(x)$ = prediksi dari pohon ke- b terhadap input x

B = jumlah total pohon dalam forest

mode = kelas yang paling sering muncul dari seluruh prediksi pohon

2.6 GBM dan XGBoost

Gradient Boosting Machine (GBM) dan *Extreme Gradient Boosting (XGBoost)* merupakan jenis algoritma dalam pendekatan *boosting*, yang banyak digunakan dalam tugas klasifikasi [30,31]. Kedua algoritma ini dikembangkan berdasarkan prinsip pembelajaran sekuensial, di mana model dibangun secara bertahap untuk memperbaiki kesalahan prediksi sebelumnya. Meskipun memiliki mekanisme dasar yang serupa, masing-masing algoritma memiliki karakteristik dan pengembangan teknis yang berbeda. Oleh karena itu, penjelasan mengenai *GBM* dan *XGBoost* disajikan secara terpisah untuk memberikan pemahaman yang lebih sistematis.

2.6.1 GBM

Gradient Boosting Machine (GBM) merupakan algoritma yang berbasis pada pohon keputusan dan membangun model secara bertahap melalui proses aditif dan sekuensial, dengan menggabungkan sejumlah *decision tree* dalam bentuk kombinasi linear untuk menghasilkan prediksi akhir. Pada setiap iterasi, model baru dibentuk untuk mempelajari dan memperbaiki kesalahan prediksi yang dihasilkan oleh model sebelumnya. Proses perbaikan kesalahan dilakukan dengan memanfaatkan informasi gradien dari fungsi kerugian sehingga pembelajaran diarahkan pada observasi yang sulit diprediksi. Setiap pohon berperan sebagai pembelajar lemah yang secara kolektif menghasilkan model prediksi yang lebih kuat. Hasil akhir diperoleh melalui penggabungan prediksi seluruh pohon dengan pembobotan tertentu. Pendekatan ini memungkinkan *GBM* menghasilkan performa yang baik pada berbagai tugas klasifikasi maupun regresi. Selain itu, algoritma ini dapat digunakan pada data dengan variabel numerik maupun kategorikal [32].

Meskipun memiliki kemampuan prediktif yang tinggi, *GBM* juga memiliki beberapa keterbatasan. Proses pembelajaran yang dilakukan secara berurutan menyebabkan waktu komputasi dan kebutuhan memori relatif lebih besar dibandingkan metode yang lebih sederhana. Selain itu, karena model terus berupaya meminimalkan kesalahan pada data pelatihan, *GBM* berpotensi mengalami *overfitting*, terutama ketika data mengandung *noise* yang tinggi. Kompleksitas struktur model juga membuat interpretasi hasil menjadi kurang intuitif. Oleh karena itu, penerapan *GBM* memerlukan pengaturan parameter yang cermat agar diperoleh keseimbangan antara kemampuan prediksi dan generalisasi model [23]. Secara matematis, model pada metode *GBM* dibangun secara iteratif dengan menambahkan fungsi baru pada setiap tahap sebagai berikut [33].

Rumus Model Iteratif GBM

$$f_m(x) = f_{m-1}(x) + g_m(x) \dots\dots\dots(2)$$

Dimana:

$f_m(x)$ = model pada iterasi ke- m
 $f_{m-1}(x)$ = model pada iterasi sebelumnya
 $g_m(x)$ = fungsi pembelajar baru (*weak learner*)
 x = variabel *input*/data

2.6.2 XGBoost

Extreme Gradient Boosting (XGBoost) merupakan pengembangan dari metode *gradient tree boosting* yang dirancang untuk meningkatkan kinerja dan efisiensi komputasi. Algoritma ini termasuk dalam pendekatan *ensemble learning*, di mana proses pembentukan model dilakukan secara bertahap dengan menghasilkan model baru yang berfokus pada perbaikan kesalahan atau residual yang dihasilkan oleh model sebelumnya. Selanjutnya, seluruh model yang terbentuk digabungkan untuk menghasilkan prediksi akhir yang lebih akurat. Secara umum, algoritma ini digunakan pada berbagai tugas seperti klasifikasi maupun regresi [34].

Meskipun memiliki efisiensi dan kemampuan prediktif yang tinggi, *XGBoost* tetap memiliki beberapa keterbatasan. Kompleksitas model yang dihasilkan membuat interpretasi menjadi kurang intuitif dan visualisasi model relatif lebih sulit dibandingkan algoritma seperti *random forest*. Proses penyetelan parameter juga dapat menjadi lebih menantang karena banyaknya komponen yang harus dikonfigurasi. Selain itu, algoritma ini tidak secara langsung menangani variabel kategorikal, sehingga data perlu direpresentasikan dalam bentuk numerik sebelum diproses. Dengan demikian, meskipun *XGBoost* menawarkan keunggulan dalam hal kecepatan dan kontrol terhadap kompleksitas model, penerapannya tetap memerlukan perhatian terhadap aspek interpretabilitas dan persiapan data [23]. Secara matematis, fungsi objektif *XGBoost* dapat dilihat pada persamaan berikut [35].

Rumus Fungsi Objektif *XGBoost*:

$$L = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \dots\dots\dots(3)$$

Dengan:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \sum_{jj=1}^T w_{jj}^2 \dots\dots\dots(4)$$

Dimana:

L = fungsi objektif (*objective function*)

$l(y_i, \hat{y}_i)$	= fungsi loss antara nilai actual (y_i) dan prediksi ($\hat{y}_i$)
n	= jumlah data
K	= jumlah pohon (<i>trees</i>)
$\Omega(f_k)$	= fungsi regularisasi pada pohon ke- k
γ	= parameter penalty kompleksitas model
T	= jumlah daun (<i>leaf nodes</i>) pada pohon
w_{jj}	= bobot pada daun ke- jj
λ	= parameter regularisasi untuk mengontrol <i>overfitting</i>

2.7 SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) merupakan salah satu metode *oversampling* yang banyak digunakan untuk menangani permasalahan ketidakseimbangan kelas [36]. Tujuan utama metode ini adalah memperbaiki distribusi kelas dengan meningkatkan jumlah sampel pada kelas minoritas tanpa mengurangi data pada kelas mayoritas. Dengan distribusi data yang lebih seimbang, model klasifikasi diharapkan mampu mempelajari karakteristik kelas minoritas secara lebih baik sehingga bias terhadap kelas mayoritas dapat dikurangi.

Berbagai pendekatan telah dikembangkan untuk menangani permasalahan *class imbalance*, antara lain *Random Oversampling*, *Random Undersampling*, *Adaptive Synthetic Sampling (ADASYN)*, *Borderline-SMOTE*, serta *cost-sensitive learning*. *Random Oversampling* berpotensi menyebabkan *overfitting* akibat duplikasi data, sedangkan *Random Undersampling* dapat menghilangkan informasi penting karena mengurangi jumlah data kelas mayoritas. *ADASYN* menghasilkan sampel sintetis secara adaptif pada area yang sulit dipelajari, namun lebih sensitif terhadap *noise* dan memerlukan pengaturan parameter yang lebih kompleks [37]. Dibandingkan metode-metode tersebut, *SMOTE* dipilih pada penelitian ini karena memiliki implementasi yang relatif sederhana, mampu menghasilkan distribusi kelas yang lebih seimbang tanpa menghilangkan informasi pada kelas mayoritas, serta telah banyak digunakan dan terbukti efektif meningkatkan performa model klasifikasi pada berbagai permasalahan data tidak seimbang [38].

Secara teknis, *SMOTE* bekerja dengan membentuk sampel sintetis melalui proses interpolasi linier antara suatu data kelas minoritas dan salah satu dari K tetangga terdekatnya yang juga berasal dari kelas minoritas. Proses ini menghasilkan titik data baru yang berada di antara dua sampel minoritas dalam ruang fitur. Dengan mekanisme tersebut, data yang

dihasilkan bukan merupakan duplikasi langsung, melainkan kombinasi yang merepresentasikan karakteristik kelas minoritas secara lebih luas. Jumlah sampel sintetis yang dihasilkan disesuaikan dengan tingkat ketidakseimbangan yang ingin diperbaiki. Dengan menambahkan contoh buatan pada kelas minoritas, rasio distribusi kelas menjadi lebih proporsional. Perbaikan distribusi ini bertujuan untuk meningkatkan performa model klasifikasi pada *dataset* yang sebelumnya didominasi oleh kelas mayoritas [37,38].

Dalam prediksi *churn* pelanggan *e-commerce*, data umumnya didominasi oleh pelanggan yang tidak *churn*, sedangkan pelanggan *churn* merupakan kelas minoritas. Ketidakseimbangan ini berpotensi menyebabkan model klasifikasi cenderung memprioritaskan kelas mayoritas dan mengabaikan pola penting dari kelas minoritas. Akibatnya, model dapat menghasilkan akurasi tinggi tetapi kurang sensitif dalam mendeteksi pelanggan yang benar-benar berisiko *churn*. Penerapan *SMOTE* dilakukan untuk mengurangi bias tersebut dengan memperkaya representasi kelas *churn* dalam data pelatihan [39,40]. Dengan distribusi yang lebih seimbang, model *ensemble* yang digunakan dalam penelitian ini diharapkan mampu mempelajari karakteristik *churn* secara lebih optimal.

Meskipun efektif dalam memperbaiki distribusi kelas, *SMOTE* memiliki keterbatasan karena tidak secara eksplisit mengontrol posisi kemunculan sampel sintetis dalam ruang fitur. Dalam kondisi tertentu, sampel baru dapat terbentuk pada wilayah yang mengandung *noise* atau berada pada area tumpang tindih antar kelas. Hal tersebut berpotensi memengaruhi kemampuan generalisasi model apabila tidak dikombinasikan dengan pengaturan parameter yang tepat. Oleh karena itu, penerapan *SMOTE* perlu disesuaikan dengan karakteristik *dataset* dan tujuan analisis yang dilakukan [37,38]. Secara matematis, proses pembentukan data sintetis pada *SMOTE* dapat dirumuskan sebagai berikut [42].

Rumus Pembentukan Data Sintetis *SMOTE*:

$$x_{nenn} = x_i + \delta(x_{zi} - x_i) \dots\dots\dots (5)$$

Dimana:

x_{nenn}	= data sintetis yang dihasilkan
x_i	= sampel dari kelas minoritas
x_{zi}	= sampel tetangga terdekat dari kelas minoritas
δ	= bilangan acak pada rentang 0 hingga 1
$x_{zi} - x_i$	= selisih vektor antara dua sampel minoritas
$x_i + \delta(x_{zi} - x_i)$	= proses interpolasi linear antar sampel minoritas

2.8 Evaluasi Model

Evaluasi model merupakan tahapan penting dalam pengembangan model *machine learning* untuk menilai sejauh mana model mampu melakukan prediksi secara akurat dan konsisten. Dalam *supervised learning*, proses ini umumnya dilakukan dengan membagi data menjadi data pelatihan dan data pengujian, di mana model dilatih menggunakan data pelatihan dan kemudian diuji menggunakan data yang belum pernah dilihat sebelumnya. Hasil prediksi pada data pengujian kemudian dibandingkan dengan nilai aktual (*ground truth*) untuk mengukur tingkat kinerja model. Evaluasi tidak hanya berfungsi untuk menentukan apakah suatu model layak digunakan, tetapi juga untuk membandingkan performa beberapa model serta mengidentifikasi metode yang paling optimal. Model dengan kinerja rendah dapat dieliminasi, sedangkan model yang menjanjikan dapat dioptimalkan lebih lanjut. Oleh karena itu, pemilihan metrik evaluasi yang tepat menjadi aspek krusial agar hasil pengukuran benar-benar mencerminkan kemampuan model secara objektif [42,43].

Dalam penelitian ini, evaluasi model dilakukan menggunakan *Confusion Matrix* sebagai dasar perhitungan kinerja klasifikasi. Berdasarkan matriks tersebut, digunakan metrik *Recall* untuk mengukur kemampuan model dalam mendeteksi pelanggan *churn* serta *Area Under the Curve (AUC)* untuk menilai kemampuan diskriminatif model secara keseluruhan. Metrik-metrik tersebut diuraikan sebagai berikut.

1. *Confusion Matrix*

Confusion matrix merupakan tabel evaluasi berbentuk matriks 2×2 yang digunakan untuk menilai kinerja model klasifikasi *biner* dengan membandingkan hasil prediksi model terhadap label aktual (*ground truth*). Matriks ini terdiri atas empat komponen utama, yaitu *true positive (TP)*, *true negative (TN)*, *false positive (FP)*, dan *false negative (FN)*, yang merepresentasikan prediksi benar dan salah yang dilakukan oleh model. Dalam *confusion matrix*, kinerja model dikatakan semakin baik apabila proporsi prediksi benar yang direpresentasikan oleh nilai *true positive (TP)* dan *true negative (TN)* meningkat terhadap keseluruhan data, karena hal ini menunjukkan kemampuan model dalam mengklasifikasikan data secara tepat. Sebaliknya, penurunan nilai *false positive (FP)* dan *false negative (FN)* mengindikasikan berkurangnya kesalahan klasifikasi, yang berdampak pada peningkatan nilai *recall* serta kemampuan diskriminatif model yang tercermin melalui *AUC* [39,44].

Tabel 2.1 *Confusion Matrix*

Aktual/Prediksi	Positif	Negatif
Positif	<i>TP</i>	<i>FN</i>
Negatif	<i>FP</i>	<i>TN</i>

2. *Recall*

Recall merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam mengidentifikasi *instance* positif secara benar dari seluruh *instance* positif yang sebenarnya ada dalam *dataset*. Metrik ini juga dikenal sebagai sensitivitas atau *true positive rate* karena berfokus pada keberhasilan model dalam mendeteksi kelas positif. Nilai *recall* menunjukkan proporsi prediksi positif yang benar dibandingkan dengan total kasus positif aktual. Nilai *recall* berkisar antara 0 dan 1, di mana nilai yang semakin mendekati 1 menunjukkan kinerja model yang semakin baik dalam mengidentifikasi kelas positif [38]. Secara matematis, *recall* dirumuskan sebagai berikut.

Persamaan *Recall*:

$$\text{Recall} = \frac{TP}{TP+FN} \dots\dots\dots(6)$$

Dimana:

TP = jumlah *instance* positif yang berhasil diprediksi sebagai positif oleh model

FN = jumlah *instance* yang salah diprediksi sebagai negatif oleh model

3. *AUC*

Area Under the Curve (AUC) merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model klasifikasi dalam membedakan dua kelas, dalam konteks penelitian ini yaitu pelanggan *churn* dan tidak *churn*. Nilai *AUC* merepresentasikan probabilitas bahwa model memberikan skor prediksi yang lebih tinggi kepada pelanggan yang benar-benar *churn* dibandingkan pelanggan yang tidak *churn*. Rentang nilai *AUC* berada antara 0,5 hingga 1, di mana nilai 0,5 menunjukkan kinerja setara dengan tebakan acak, sedangkan nilai yang semakin mendekati 1 menunjukkan kemampuan diskriminasi yang semakin baik. Oleh karena itu, *AUC* digunakan sebagai indikator komprehensif untuk menilai kualitas model dalam membedakan pelanggan berisiko *churn* tanpa bergantung pada satu ambang klasifikasi tertentu. Untuk memberikan gambaran yang

lebih jelas mengenai tingkat kinerja model berdasarkan nilai *AUC* yang diperoleh, berikut disajikan tabel interpretasi nilai *AUC* [46].

Tabel 2.2 Nilai *AUC*

Nilai <i>AUC</i>	Interpretasi Kemampuan Model
0,50 – 0,60	Sangat Kurang
0,60 – 0,70	Kurang
0,70 – 0,80	Cukup
0,80 – 0,90	Baik
0,90 – 1,00	Sangat Baik

2.9 OSEM N

OSEMN merupakan salah satu kerangka kerja dalam *data science* yang digunakan untuk mengelola alur kerja analisis data secara sistematis dan terstruktur. *OSEMN* adalah akronim dari *Obtain*, *Scrub*, *Explore*, *Model*, dan *iNterpret*, yang merepresentasikan tahapan-tahapan utama dalam mengolah data hingga menghasilkan informasi yang bermakna. Kerangka kerja ini membantu peneliti dalam memastikan bahwa setiap proses, mulai dari pengumpulan data hingga interpretasi hasil model, dilakukan secara terorganisir dan logis [46,47]. Adapun penjelasan masing-masing tahapan dalam *framework OSEM N* diuraikan sebagai berikut.

1. *Obtain*

Obtain merupakan proses pengumpulan data yang dapat diperoleh dari berbagai sumber dan relevan dengan tujuan penelitian. Pada tahap ini, peneliti memastikan bahwa data yang dikumpulkan sesuai dengan kebutuhan analisis, mencakup variabel-variabel yang merepresentasikan karakteristik pelanggan serta status *churn*. Kualitas dan kelengkapan data yang diperoleh pada tahap ini sangat menentukan keberhasilan proses analisis pada tahapan berikutnya [49].

2. *Scrub*

Tahap *scrub* berfokus pada proses pembersihan dan prapemrosesan data untuk meningkatkan kualitas *dataset* [49]. Kegiatan yang dilakukan meliputi penanganan data hilang (*missing values*), penghapusan duplikasi, koreksi inkonsistensi format, serta transformasi variabel agar sesuai dengan kebutuhan pemodelan. Selain itu, tahap ini juga dapat mencakup proses *encoding* data kategorikal dan normalisasi data numerik.

Prapemrosesan yang tepat bertujuan untuk meminimalkan bias dan kesalahan yang dapat memengaruhi kinerja model prediksi [47].

3. *Explore*

Tahap *explore* bertujuan untuk memahami karakteristik dan pola yang terdapat dalam data melalui analisis eksploratif. Proses ini dilakukan dengan menggunakan statistik deskriptif dan visualisasi data untuk mengidentifikasi distribusi variabel, hubungan antar fitur, serta potensi ketidakseimbangan kelas. Analisis eksploratif membantu peneliti dalam menentukan strategi pemodelan yang tepat dan mengidentifikasi fitur-fitur yang berpotensi berkontribusi signifikan terhadap prediksi [49].

4. *Model*

Tahap *model* merupakan proses pembangunan model prediktif berdasarkan data yang telah diproses. Proses ini mencakup pembagian data latih dan data uji, pelatihan model, serta optimasi parameter jika diperlukan. Tujuan utama tahap ini adalah menghasilkan model dengan kemampuan klasifikasi yang optimal [49].

5. *iNterpret*

Tahap *iNterpret* merupakan proses evaluasi dan penafsiran hasil model untuk memperoleh pemahaman yang komprehensif terhadap kinerjanya. Evaluasi dapat dilakukan menggunakan berbagai metrik guna menilai kemampuan model. Selain menilai performa, tahap ini juga bertujuan untuk menarik implikasi praktis dari hasil prediksi, sehingga dapat mendukung pengambilan keputusan strategis dalam bisnis [48,49].

2.10 Studi Literatur

Tabel 2.3 Studi Literatur

No	Judul	Domain	Model	Kontribusi	Teknik Imbalance	Metrik	Hasil	Analisis Gap
1	<i>Customer churn prediction system: a machine learning approach [51]</i>	Telekomunikasi	Beberapa algoritma ML dan <i>boosting</i> (RF dan <i>XGBoost</i>)	Menggunakan pendekatan <i>ensemble learning</i> untuk mencapai hasil yang lebih baik	Tidak disebutkan	<i>Confusion matrix</i> dan <i>AUC curve</i>	<i>XGBoost</i> memperoleh akurasi 80,8% dan <i>AUC</i> 84%, menunjukkan performa terbaik dibandingkan model lainnya berdasarkan <i>confusion matrix</i> dan kurva <i>AUC</i>	Penelitian ini menggunakan <i>GSA</i> untuk seleksi fitur untuk prediksi <i>churn</i> , namun belum menerapkan teknik <i>SMOTE</i> seperti yang digunakan dalam penelitian ini
2	<i>BC2 E-Commerce Customer Churn Prediction Based on K-Means and SVM [52]</i>	<i>E-Commerce</i>	<i>SVM</i> dan <i>Logistic Regression</i>	Mengusulkan model prediksi <i>churn</i> pelanggan <i>B2C e-commerce</i> yang mengintegrasikan segmentasi <i>K-Means</i> dengan klasifikasi <i>SVM</i>	<i>SMOTE</i>	<i>Accuration</i> , <i>recall</i> , <i>precision</i> , dan <i>AUC</i>	Segmentasi <i>K-Means</i> meningkatkan performa prediksi, dengan <i>SVM</i> menghasilkan akurasi lebih tinggi	Penelitian ini berfokus pada integrasi <i>K-Means</i> dengan <i>SVM</i> dan <i>LR</i> untuk memprediksi <i>churn</i> , namun belum mengeksplorasi

				untuk meningkatkan akurasi prediksi dan mendukung strategi retensi pelanggan			dibandingkan regresi logistik	penggunaan <i>EL</i> seperti <i>Random Forest</i> dan <i>XGBoost</i>
3	<i>Research on telecom customer churn prediction based on ensemble learning [53]</i>	Telekomunikasi	Diantaranya yaitu: <i>random forests</i> dan <i>gradient boosting decision tree</i>	Mengombinasikan algoritma <i>machine learning</i> guna meningkatkan efisiensi alokasi sumber daya dan menekan tingkat <i>churn</i> pelanggan	<i>SMOTE</i>	<i>Precision, Recall, dan F1</i>	<i>Ensemble learning</i> yang dipadukan dengan <i>preprocessing, K-means</i> , dan <i>SMOTE</i> mampu meningkatkan akurasi prediksi <i>churn</i> hingga 6,43%	Penelitian ini berfokus pada <i>churn</i> pelanggan kontrak di telekomunikasi, sehingga belum merepresentasikan perilaku pelanggan non-kontrak yang lebih dinamis pada <i>e-commerce</i>
4	<i>Telecom Churn Prediction System Based on Ensemble Learning Using Feature Grouping [54]</i>	Telekomunikasi	<i>XGBoost, Logistic regression, decision tree, naïve bayes classifier, dan soft voting</i>	Mengusulkan sistem prediksi <i>churn</i> berbasis <i>ensemble learning</i> dengan konstruksi fitur baru untuk meningkatkan akurasi model	Tidak disebutkan	<i>Accuracy, Precision, Recall, dan F1score</i>	Model yang diusulkan mencapai akurasi tertinggi sebesar 98,09%, setelah dilakukan konstruksi fitur	Penelitian ini menggunakan <i>EL</i> pada domain telekomunikasi, namun belum menguji pada <i>e-commerce</i> serta tidak menerapkan teknik penanganan <i>class imbalance</i>

5	<i>Proactive customer retention management in a non-contractual B2B setting based on churn predictionn with random forests [55]</i>	<i>E-commerce</i>	<i>Random forest</i>	Merancang kerangka retensi pelanggan proaktif berbasis prediksi <i>churn</i> yang terintegrasi pada proses pengambilan keputusan bisnis pada konteks <i>B2B</i> non-kontraktual	Tidak disebutkan	<i>AUC</i>	Model <i>random forest</i> mencapai <i>AUC</i> 0.76 pada data uji, menunjukkan kemampuan cukup baik dalam membedakan pelanggan berisiko <i>churn</i>	Penelitian ini belum mempertimbangkan pelanggan yang tidak jadi <i>churn</i> akibat intervensi retensi, sehingga belum sepenuhnya mencerminkan perilaku aktual pelanggan
6	<i>Research on customer churn prediction and model interpretability analysis [56]</i>	Perbankan	<i>XGBoost, LightGBM, dan GA-XGBoost</i>	Mengembangkan model <i>XGBoost</i> yang dioptimalkan melalui algoritma genetika untuk penyetelan hiperparameter pasca-penyeimbangan data		<i>Precision, recall, F1-score, accuracy, dan AUC</i>	<i>GA-XGBoost</i> memperoleh performa tertinggi pada seluruh metrik evaluasi melampaui model pembandingan	Penelitian ini menunjukkan efektivitas metode <i>boosting</i> , namun masih menghadapi potensi redundansi dan kovarians fitur, sehingga membuka peluang pengembangan melalui pendekatan fitur perilaku
7	<i>Customer Personality Analysis Using Hybrid Ensemble Models and</i>	Pemasaran	<i>Hybrid HSLR (RF, XGBoost, LGBM, AdaBoost + LR</i> sebagai	Menerapkan teknik <i>CTGAN</i> dan <i>SMOTE</i> untuk mengatasi ketidakseimbangan data, sehingga	<i>CTGAN, dan SMOTE</i>	<i>Accuracy, precision, recall, F1, MCC, dan skor ROC</i>	<i>Hybrid HSLR</i> dengan <i>SMOTE</i> mencapai performa tertinggi	Penelitian ini mencapai performa tinggi, namun masih terbatas pada satu <i>dataset</i> sehingga belum

	<i>Class Balancing Techniques [57]</i>		<i>meta classifier)</i>	meningkatkan keandalan prediksi <i>churn</i> .			dengan <i>AUC</i> 0,984	menguji generalisasi model pada domain berbeda seperti <i>e-commerce</i>
8	<i>Utilizing data sampling technique on algorithmic fairness for customer churn prediction with data imbalance [58]</i>	Telekomunikasi	<i>Random forest, decision tree, LightGBM, gradient boosting, logistic regression, dan XGBoost</i>	Menganalisis pengaruh teknik penyeimbangan data terhadap kinerja serta keadilan algoritma yang digunakan dalam prediksi <i>churn</i>	<i>SMOTE</i> dan <i>ADSYN</i>	<i>Precision, recall, accuracy, F-I, dan AUC-ROC</i>	Penerapan teknik <i>oversampling</i> khususnya pada <i>XGBoost</i> mampu meningkatkan performa klasifikasi sekaligus memperbaiki metrik <i>fairness</i> dibandingkan model tanpa penanganan <i>imbalance</i>	Penelitian ini masih menghadapi potensi bias pada data latih meskipun telah menerapkan teknik penanganan ketidakseimbangan, serta belum secara komprehensif mengakomodasi aspek <i>fairness</i> dan bias atribut dalam pemodelan <i>churn</i> sehingga membuka peluang pengembangan melalui penelitian ini