

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Pelanggan

Pelanggan adalah orang-orang yang melakukan kegiatan membeli dan menggunakan suatu produk, baik berupa barang maupun jasa secara terus menerus. Pelanggan atau pengguna suatu produk adalah orang-orang yang berhubungan secara langsung maupun tidak langsung dengan perusahaan-perusahaan bisnis(Daryanto & Setyobudi, 2014).

2.1.2. Churn

Perputaran pelanggan atau *Customer churn* merupakan kondisi dimana pelanggan tidak lagi melanjutkan berlangganan pada suatu perusahaan yang dipilih dan berpindah kepada perusahaan pesaing(Richeldi & Perrucci, 2002). *Customer churn* merupakan istilah yang digunakan untuk mendefinisikan bahwa seorang *customer* (pelanggan) menghentikan hubungan bisnis dengan suatu perusahaan(Liao & Triantaphyllou, 2008). *Customer churn* merupakan istilah untuk mendefinisikan perputaran pelanggan, atau lebih khususnya disebut sebagai *customer churn management*. Manajemen *customer churn* merupakan sebuah konsep untuk mengidentifikasi para pelanggan yang berkeinginan untuk berpindah dari satu perusahaan yang telah dipilih ke perusahaan pesaing, sehingga sekali pelanggan tersebut teridentifikasi *churn* maka pelanggan tersebut menjadi target pemasaran proaktif sebagai upaya *retention*(Hadden dkk., 2007).

Masalah perputaran pelanggan ini menjadi krusial, karena biaya yang harus dikeluarkan untuk mendapatkan pelanggan baru, iklan, pemasaram, komisi, dan lain-lain akan jauh lebih besar dibandingkan biaya yang harus dikeluarkan untuk menjaga pelanggan yang sudah ada. Ditambah lagi kebanyakan pelanggan baru cenderung tidak lebih menghasilkan keuntungan dibandingkan pelanggan yang sudah lama dan bertahan (Herawati dkk., 2016). Sehingga mempertahankan pelanggan yang sudah ada lebih diprioritaskan daripada mencoba memenangkan persaingan untuk mendapatkan pelanggan baru (Hadden dkk., 2007).

Analisis *churn* sering digunakan pada semua sektor industri, salah satunya adalah sektor telekomunikasi yang mempunyai potensi *churn* yang sangat tinggi, sehingga biaya *churn* yang sangat besar menyebabkan potensi kerugian yang besar (Şimşek Gürsoy, 2010). Ketika informasi telah dimiliki oleh perusahaan, maka manajer pemasaran dapat mengambil tindakan yang tepat dan strategis untuk meminimalkan ataupun mencegah terjadinya *churn* (Richeldi & Perrucci, 2002).

Churn pelanggan dapat disebabkan oleh banyak hal, menurut (Keaveney, 1995) pada penelitian (Parmawati, 2008) dalam sebuah studi akademis sekitar 500 insiden pengalihan layanan, terdapat 8 motif terbesar yang melatar belakangi keputusan beralih (berhenti berlangganan) yang telah teridentifikasi. Delapan motif terbesar tersebut yaitu:

1. Masalah Tarif/Harga (*Pricing*)

Alasan tarif yang terlalu mahal, tarif mengalami kenaikan, serta tarif yang tidak wajar.

2. Masalah Ketidaknyamanan dan Ketidaksenangan Pelayanan (*Inconvenience*)

Ditunjukkan dengan ketidaknyamanan pada tempat dan waktu sehingga terlalu lama dan harus menunggu.

3. Kegagalan Inti Layanan (*Core Service Failure*)

Adanya kekeliruan dan kesalahan dalam pelayanan yang menyebabkan kerugian/masalah bagi pelanggan.

4. Kegagalan dalam Pemberian Pelayanan (*Service Encounter Failure*)

Disebabkan pelayanan yang tidak memerdulikan pelanggan, tidak sopan dan tanggap, petugas lalai dalam tugasnya.

5. Tanggapan Terhadap Kegagalan Pelayanan (*Response to Service Failure*)

Muncul tanggapan negatif dari pelanggan tentang pelayanan, tidak ada tanggapan, dan petugas melayani dengan enggan.

6. Persaingan (*Competition*)

Pelanggan menemukan layanan lain yang lebih baik.

7. Faktor Etik (*Ethical Problems*)

Menggunakan teknik pelayanan dengan kecurangan dan memaksa.

8. Faktor Peralihan Diluar Inisiatif (*Involuntary Switching*)

Pelanggan pindah atau penyedia layanan tutup.

2.1.3. Telekomunikasi

Telekomunikasi adalah setiap pemancaran, pengiriman, dan atau penerimaan dari setiap informasi dalam bentuk tanda-tanda, isyarat, tulisan,

gambar, suara, dan bunyi melalui sistem kawat, optik, radio atau sistem elektromagnetik lainnya (Indonesia. Pemerintah Pusat, 1999).

2.1.4. Performa

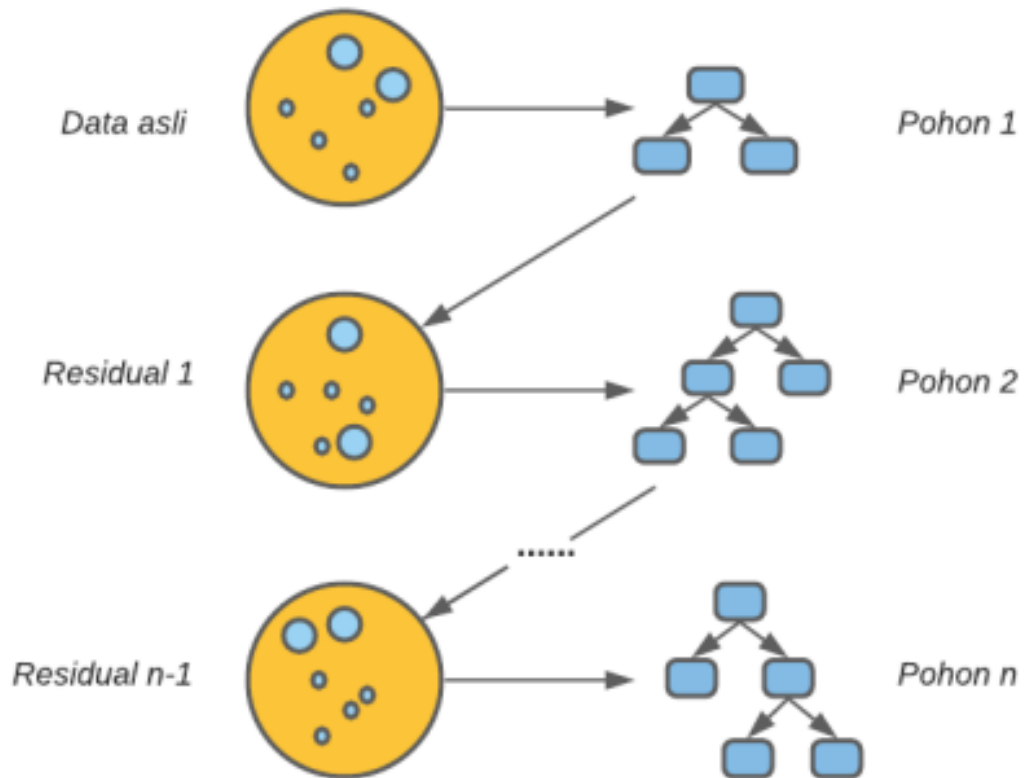
Performa dalam bidang algoritma dapat didefinisikan sebagai efektivitas dan efisiensi algoritma dalam menyelesaikan tugas tertentu (Sedgewick & Wayne, 2011).

2.1.5. Algoritma

Dalam bidang matematika dan informatika, algoritma adalah serangkaian prosedur yang saling berinteraksi untuk mencapai suatu tujuan dalam memecahkan masalah tertentu. Sehingga algoritma dapat digunakan untuk penghitungan, pemrosesan data, pencarian, penalaran, optimasi, pembelajaran dan sejenisnya untuk menyelesaikan suatu masalah (Santoso dkk., 2020).

2.1.6. *Boosting*

Boosting pertama kali diusulkan oleh Robert E. Schapire pada tahun 1999 (Schapire, 1999). Sesuai dengan namanya, teknik *boosting* bekerja dengan cara memperkuat (*boost*) sebuah model klasifikasi awal yang lemah. Teknik ini melibatkan penggabungan *weak learners* atau umumnya juga disebut *base learners*, untuk membentuk *strong learners* sehingga menghasilkan model yang lebih baik (Shakya, 2018). Berbeda dengan teknik *bagging* yang prosesnya berjalan secara paralel, teknik *boosting* melatih *weak learners* secara sekuensial lalu menggabungkannya. Contoh algoritma dengan *boosting* yaitu *AdaBoost*, *Gradient Boost*, dan lain-lain. Ilustrasi dari *boosting* tertera pada Gambar 2.1.

Gambar 2.1 Ilustrasi *Boosting*

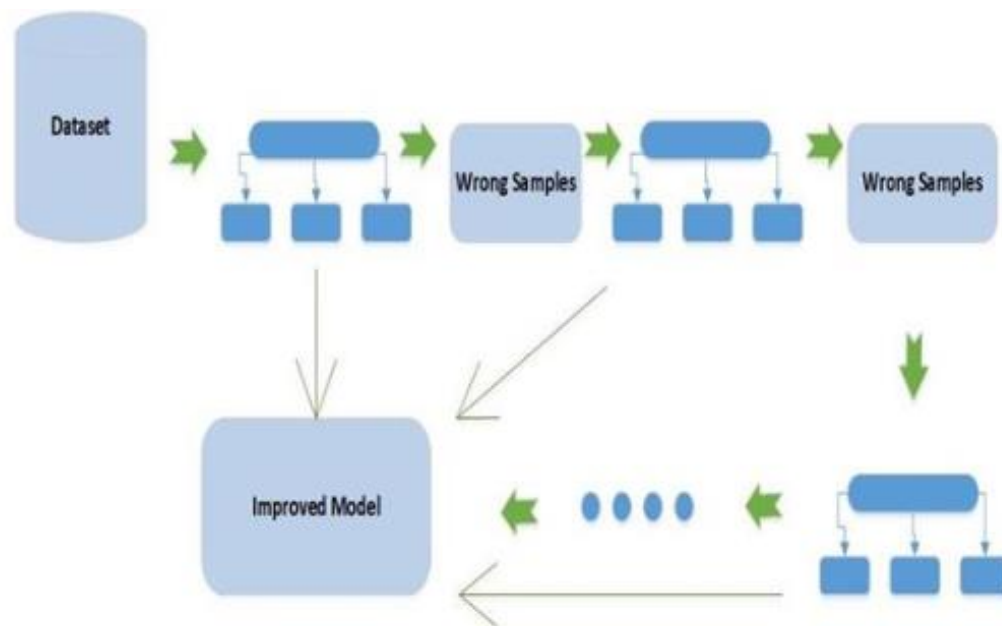
Sumber: (Shakya, 2018)

2.1.7. *Gradient Boost*

Gradient Boost adalah algoritma pengembangan dari algoritma *Decision Tree*. *Gradient boost* menjadi lebih populer karena kemampuannya untuk mengkategorikan data yang sulit (Friedman, 2001). Rumus umum untuk *Gradient Boost* dapat dinyatakan seperti Persamaan 2.1 (Son dkk., 2015):

$$Z_{p+1}(a) = Z_p(a) + d(a) = b \quad (2.1)$$

Dimana p adalah setiap tahap, Z_p adalah model yang tidak sempurna, b adalah variabel *output*, dan d adalah *estimator* yang ditambahkan untuk memberikan model yang lebih baik. Ilustrasi dari *Gradient Boost* tertera pada Gambar 2.2.



Gambar 2.2 Ilustrasi *Gradient Boost*

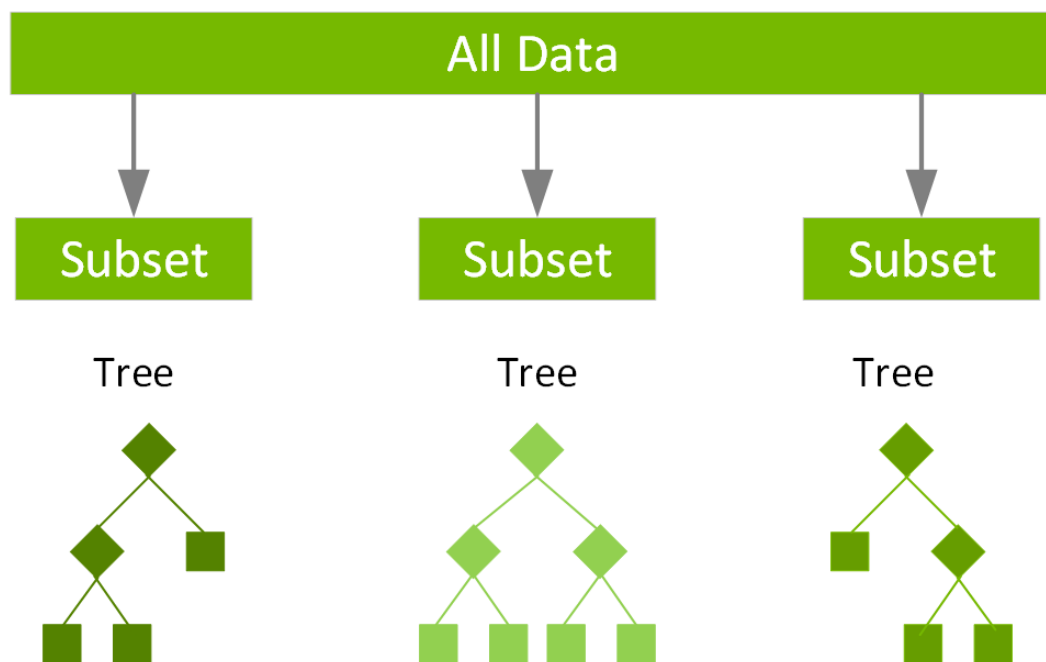
Sumber: (Friedman, 2001)

2.1.8. *Extreme Gradient Boost*

Extreme Gradient Boost (XGBoost) dikembangkan oleh (T. Chen & Guestrin, 2016). *XGBoost* merupakan salah satu metode *boosting* yaitu kumpulan *Decision Tree* yang pembangunan pohon berikutnya akan bergantung pada pohon sebelumnya. *XGBoost* adalah salah satu algoritma *machine learning* yang banyak digunakan. Algoritma ini dapat digunakan untuk tugas-tugas pembelajaran yang diawasi seperti regresi, klasifikasi, dan pemeringkatan, yang dibuat berdasarkan standar sistem *Gradient Boost* dan dimaksudkan untuk mendorong jangkauan terjauh perhitungan mesin yang luar biasa untuk memberikan hasil yang serbaguna, ringkas, dan tepat. Untuk dataset yang diberikan dengan n model dan k fitur $F=(a_i, b_i) (|F|=n, a_i \in Z^k, b_i \in Z)$ model *ensemble* pohon, menggunakan fungsi aditif S untuk meramalkan hasil seperti Persamaan 2.2 (Zhang dkk., 2018):

$$b_i = \phi(a_i) = \sum_{s=1}^S m_s(a_i), m_s \in M \quad (2.2)$$

Dimana $M = \{m(a) = \omega_l(a)\} (l : Z^k \rightarrow N, \omega \in Z^N)$ dimaksudkan sebagai ruang pohon regresi. Selain itu, l menunjukkan struktur setiap pohon yang memetakan panduan untuk membandingkan catatan daun, N adalah jumlah daun, m_s menandakan struktur pohon bebas l dan bobot daun ω . Setiap pohon regresi berisi skor persisten pada setiap daun, di mana ω_i digunakan untuk menunjukkan skor pada daun ke- i . Ilustrasi dari *Extreme Gradient Boost* tertera pada Gambar 2.3.



Gambar 2.3 Ilustrasi *Extreme Gradient Boost*

Sumber: (T. Chen & Guestrin, 2016)

2.1.9. Adaptive Boost

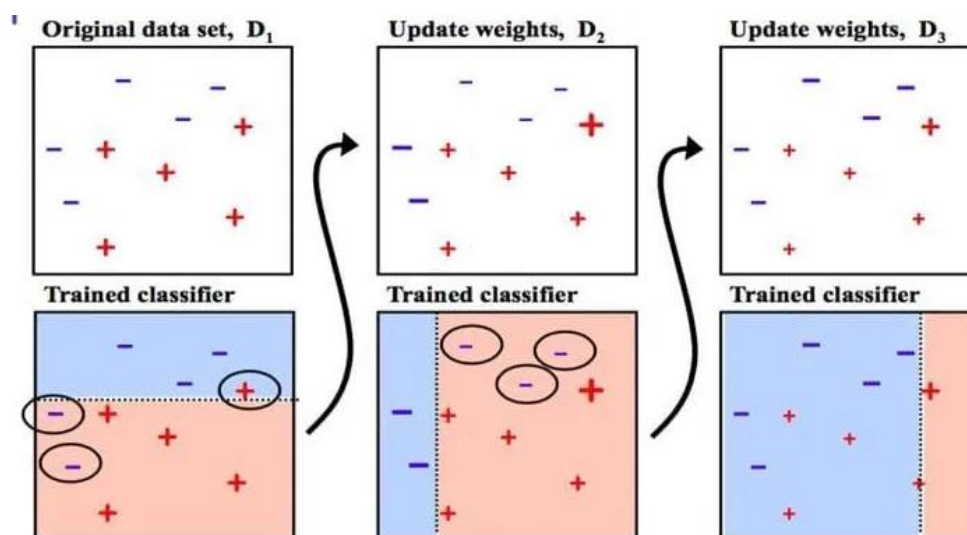
Adaptive Boost (AdaBoost) adalah algoritma *boosting meta-estimator* yang dimulai dengan menyesuaikan pengklasifikasi pada dataset asli, kemudian menyesuaikan salinan pengklasifikasi lebih lanjut pada dataset yang sama, tetapi

menyesuaikan bobot contoh yang diklasifikasikan dengan lemah sehingga pengklasifikasi di masa depan lebih fokus pada kasus-kasus yang sulit (Freund & Schapire, 1997).

AdaBoost adalah adaptif dalam arti bahwa ia menyesuaikan *weak learners* yang berurutan untuk memilih contoh-contoh yang salah diklasifikasikan oleh pengklasifikasi sebelumnya. Masing-masing *learners* bisa jadi tidak efektif, selama kinerja *learners* tidak lebih efisien daripada menebak secara acak, keseluruhan model akhir akan bergabung dengan *learners* yang kuat, persamaan akhir untuk *AdaBoost* direpresentasikan seperti Persamaan 2.3 (Li dkk., 2008):

$$Z(a) = \text{sign} \left(\sum_{p=1}^P \Omega_p z_p(a) \right) \quad (2.3)$$

Dimana Z_p mewakili pengklasifikasi ke- p^{th} dan Ω_p adalah bobot pembandingan. Ini sebenarnya adalah campuran tertimbang dari P pengklasifikasi lemah. Ilustrasi dari *Adaptive Boost* tertera pada Gambar 2.4.



Gambar 2.4 Ilustrasi *Adaptive Boost*

Sumber: (Freund & Schapire, 1997)

2.1.10. *Imbalanced Data*

Data yang tidak seimbang atau lebih sering disebut *Imbalanced Data*, adalah suatu kondisi dimana data memiliki rasio yang tidak seimbang antara satu kelas dengan kelas yang lain, sehingga terdapat kelas mayoritas dan kelas minoritas. Kelas yang tidak seimbang adalah masalah umum dalam klasifikasi pembelajaran mesin. *Imbalance class* merupakan suatu kondisi dimana distribusi antar kelas yang tidak proporsional pada sebuah dataset, dimana terdapat salah satu kelas yang memiliki jumlah data sangat besar (kelas mayoritas) dibandingkan dengan kelas lainnya (kelas minoritas)(Fernández dkk., 2018). Perbedaan jumlah data yang sangat besar antar kelas dapat mengakibatkan model klasifikasi sering tidak mampu memprediksikan kelas minoritas dengan tepat sehingga banyak data pengujian yang seharusnya berada pada kelas minoritas diprediksikan salah oleh model klasifikasi(AlMahadin dkk., 2022).

2.1.11. *Prediksi*

Prediksi nilai masa depan dari deret waktu menggunakan beberapa langkah manipulasi data untuk mendapatkan tren kumulatif agar sesuai dengan tren kumpulan data original di masa lalu(Parbat & Chakraborty, 2020).

2.1.12. *Confusion Matrix*

Confusion Matrix adalah suatu proses yang digunakan dalam mengevaluasi keakuratan model klasifikasi yang telah dibuat untuk mengidentifikasi data dalam kelas yang berbeda. Dengan mengukur tingkat keakuratan, maka dapat digunakan untuk menilai performa model klasifikasi tersebut. Akurasi dari model klasifikasi diperoleh dari data pelatihan yang diukur sebagai persentase dari data pelatihan

yang berhasil diklasifikasikan dengan benar. Sedangkan, evaluasi hasil uji dari model klasifikasi menggunakan tabel yang dikenal sebagai *confusion matrix*, yang menyajikan jumlah data uji yang diklasifikasikan dengan benar dan yang salah (Townsend, 1971). Representasi *Confusion Matrix* tertera pada Gambar 2.5.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 2.5 *Confusion Matrix*

Sumber: (Xu dkk., 2020)

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP) merupakan data positif yang diprediksi benar, *True Negative* (TN) merupakan data negatif yang diprediksi benar, *False Positive* (FP) *Type I Error* merupakan data negatif namun diprediksi sebagai data positif, dan *False Negative* (FN) *Type II Error* merupakan data positif namun diprediksi sebagai data negatif (Xu dkk., 2020).

Dalam *confusion matrix*, kita dapat menghitung nilai akurasi, presisi, *recall*, dan *F1-score*, yang memberikan gambaran lebih detail tentang performa model klasifikasi tersebut(Goutte & Gaussier, 2005).

1. *Accuracy*

Akurasi (*Accuracy*) adalah metrik evaluasi yang mengukur seberapa baik model membuat prediksi yang benar dari total prediksi yang dilakukan(Goutte & Gaussier, 2005). Nilai akurasi dapat dihitung dengan Persamaan 2.4 berikut:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (2.4)$$

2. *Precision*

Presisi (*Precision*) adalah metrik evaluasi yang mengukur seberapa baik model membuat prediksi yang benar untuk kelas positif dari total prediksi positif yang dilakukan(Goutte & Gaussier, 2005). Nilai presisi dapat dihitung dengan Persamaan 2.5 berikut:

$$Precision = \frac{TP}{(TP+FP)} \quad (2.5)$$

3. *Recall*

Sensitivitas (*Recall*) adalah metrik evaluasi yang menggambarkan seberapa baik suatu model dalam mengidentifikasi kelas positif dengan benar(Goutte & Gaussier, 2005). Nilai sensitivitas dapat dihitung dengan Persamaan 2.6 berikut:

$$Recall = \frac{TP}{(TP+FN)} \quad (2.6)$$

4. *F1-Score*

F1-Score merupakan metrik evaluasi yang mencerminkan keseimbangan antara Presisi (*Precision*) dan Sensitivitas (*Recall*)(Goutte & Gaussier, 2005). Nilai *F1-Score* dapat dihitung dengan Persamaan 2.7 berikut:

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.7)$$

2.1.13. *Log Loss*

Log Loss adalah metrik evaluasi yang umum digunakan dalam masalah klasifikasi untuk mengevaluasi kualitas prediksi dari model probabilitas. Secara matematis, *log loss* mengukur seberapa baik model mengestimasi probabilitas target yang benar(Bishop, 2006). *Log Loss* ini mengukur kinerja model klasifikasi, di mana outputnya adalah probabilitas dengan nilai antara 0 dan 1. Model yang sempurna akan memiliki *Log Loss* 0 atau bisa dikatakan semakin nilainya mendekati 0 semakin baik model tersebut(Hastie dkk., 2001). Rumus untuk menghitung *log loss* adalah Persamaan 2.8 berikut:

$$H(P, Q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(P(y_i)) + (1 - y_i) \cdot \log(1 - P(y_i)) \quad (2.8)$$

N adalah jumlah total sampel, y_i adalah label aktual dari sampel ke-i (0 untuk kelas negatif, 1 untuk kelas positif), p_i adalah probabilitas prediksi bahwa sampel ke-i termasuk dalam kelas positif menurut model.

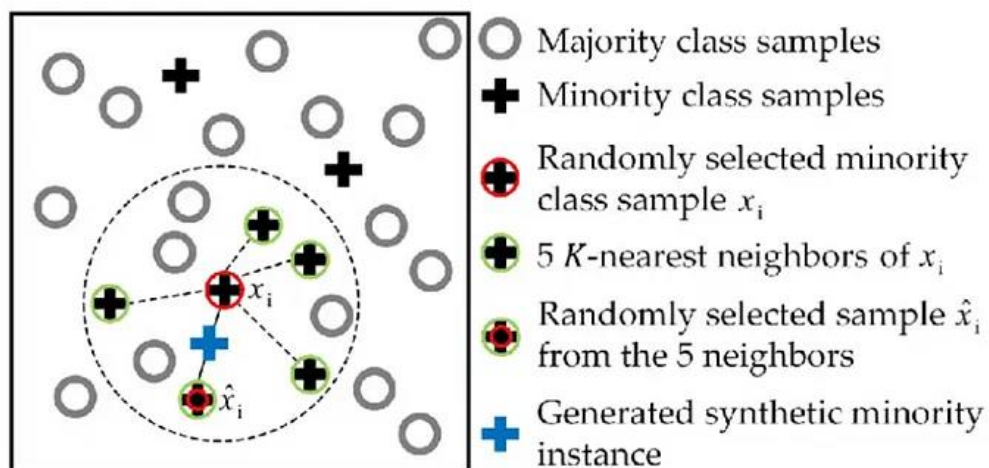
2.1.14. *Synthetic Minority Over-sampling Technique*

Synthetic Minority Over-sampling Technique (SMOTE) adalah metode *over-sampling* dimana data pada kelas minoritas diperbanyak dengan menggunakan data sintetik yang berasal dari replikasi data pada kelas minoritas. *Over-sampling*

pada *SMOTE* mengambil contoh dari kelas minoritas lalu mencari *k-nearest neighbors* dari setiap contoh, kemudian menghasilkan contoh sintetik daripada mereplikasi contoh kelas minoritas (Chawla dkk., 2002). Cara kerja *SMOTE* :

1. Mengidentifikasi kelas minoritas
2. Secara acak memilih 1 titik minoritas data
3. Memilih *K-nearest neighbors* dari kelas yang sama (*default* $k = 5$)
4. Memilih 1 secara acak tetangganya lalu menghitung jarak perbedaannya kemudian dikalikan dengan bilangan antara 0–1. Hasil dari ini menjadi *synthetic sample*
5. Mengulang langkah 1–4 sampai jumlah data seimbang.

Ilustrasi dari proses *SMOTE* tertera pada Gambar 2.6.



Gambar 2.6 Ilustrasi *SMOTE*

Sumber: (Chawla dkk., 2002)

2.1.15. Korelasi Variabel

Korelasi variabel adalah ukuran statistik yang menggambarkan sejauh mana dua variabel berkaitan satu sama lain. Korelasi mengukur kekuatan dan arah hubungan linier antara dua variabel (Lu, 2017). Nilai korelasi dapat berkisar dari -1 hingga 1, dimana:

1. Nilai 1 menunjukkan korelasi positif sempurna: Ketika satu variabel meningkat, variabel lainnya juga meningkat dengan proporsi yang sama.
2. Nilai -1 menunjukkan korelasi negatif sempurna: Ketika satu variabel meningkat, variabel lainnya menurun dengan proporsi yang sama.
3. Nilai 0 menunjukkan tidak ada korelasi: Tidak ada hubungan linier antara dua variabel tersebut.

Korelasi sering diukur menggunakan koefisien korelasi *Pearson*, yang mengukur kekuatan dan arah hubungan linear antara dua variabel. Rumus menghitung korelasi variabel adalah Persamaan 2.9 berikut:

$$r_{XY} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \quad (2.9)$$

Dimana:

1. X_i dan Y_i adalah nilai individu dari variabel X dan Y.
2. \bar{X} dan \bar{Y} adalah rata-rata dari variabel X dan Y.

2.2. Penelitian Terkait

Penelitian yang dilakukan menggunakan beberapa penelitian terkait yaitu tertera pada tabel 2.1 berikut:

Tabel 2.1 Penelitian Terkait

No	Peneliti, Tahun	Ringkasan	Hasil
1	(Sutoyo & Fadlurrahman, 2020)	Pada penelitian ini, diidentifikasi bahwa dataset performansi <i>rating</i> iklan TV yang digunakan memiliki permasalahan <i>imbalance class</i> yang sangat besar dimana <i>instance</i> yang memiliki nilai <i>rating</i> tinggi, jauh lebih sedikit dibandingkan <i>instance</i> yang memiliki nilai <i>rating</i> kecil dan menengah. Sehingga diperlukan metode <i>over-sampling</i> untuk mengatasi permasalahan <i>imbalance class</i> tersebut. Metode digunakan adalah <i>Synthetic Minority Oversampling Technique (SMOTE)</i> . Untuk memvalidasi keefektifan model yang diusulkan, dilakukan dua skenario eksperimental yaitu:	Hasil eksperimen menunjukkan bahwa performansi <i>ANN+SMOTE</i> mencapai akurasi sebesar 87.06% dibandingkan <i>ANN</i> yang hanya sebesar 86.35%. Penerapan Teknik <i>SMOTE</i> terbukti dapat mengatasi masalah ketidakseimbangan data dan mendapatkan hasil klasifikasi yang lebih baik.

Tabel 2.1 Penelitian Terkait (lanjutan 1)

No	Peneliti, Tahun	Ringkasan	Hasil
		<p>pertama algoritma <i>ANN</i> langsung digunakan untuk pemodelan tanpa mempertimbangkan ketidakseimbangan kelas, dan kedua dilakukan <i>oversampling SMOTE</i> untuk meningkatkan jumlah dataset agar mencapai dataset yang seimbang.</p>	
2	(Lalwani dkk., 2022)	<p>Pada penelitian ini dilakukan perbandingan performa algoritma <i>Logistic Regression, Naïve Bayes, Support Vector Machines, Decision Trees, Random Forest, XGBoost, CatBoost, AdaBoost</i> dan <i>Extra tree Classifier</i> dalam memprediksi <i>churn</i> pelanggan suatu industri telekomunikasi.</p>	<p>Algoritma <i>boosting (XGBoost, CatBoost dan AdaBoost)</i> memiliki performa yang lebih baik dibandingkan algoritma lainnya.</p>
3	(Iqbal dkk., 2022)	<p>Pada penelitian ini bertujuan untuk meningkatkan akurasi dalam mendeteksi kanker payudara dengan tiga algoritma yakni <i>Gradient Boost, Adaptive Boost,</i> dan <i>Extreme Gradient Boost.</i></p>	<p>Algoritma <i>Gradient Boost</i> mengungguli algoritma lainnya dengan akurasi 98,83%</p>

Tabel 2.1 Penelitian Terkait (lanjutan 2)

No	Peneliti, Tahun	Ringkasan	Hasil
4	(Christanto dkk., 2023)	Pada penelitian ini dilakukan analisis sentimen <i>review</i> hotel pada <i>platform</i> trip advisor dengan menggunakan algoritma <i>Decision Tree</i> , <i>Support Vector Machine</i> , dan <i>XGBoost</i>	Pada penelitian ini algoritma terbaik yang dapat menentukan sentimen <i>analysis review</i> hotel adalah algoritma <i>XGBoost</i> dengan akurasi sebesar 99%
5	(Sabita & Trisnawati, 2023)	Fokus penelitian ini adalah untuk membuat sebuah model <i>machine learning</i> yang nantinya akan dapat memprediksi waktu kelulusan mahasiswa dengan menggunakan dua metode yang berbeda yaitu <i>Support Vector Machine (SVM)</i> dan <i>AdaBoost</i> .	Model algoritma <i>AdaBoost</i> memiliki nilai akurasi yang lebih baik dari pada <i>SVM</i> , dengan selisih 0,04 (0,69 dan 0,73)

2.3. Kebaruan Penelitian

Kebaruan pada penelitian ini berdasarkan pada penelitian-penelitian sebelumnya dengan masalah atau tujuan yang sama adalah penggunaan algoritma yang berbeda tertera pada tabel 2.2 berikut:

Tabel 2.2 Penelitian dengan tujuan yang sama

No	Peneliti, Tahun	Algoritma yang digunakan
1	(Kaharudin dkk., 2019)	<i>Naive Bayes</i> dan <i>K-Nearest Neighbor</i>
2	(Latief dkk., 2021)	<i>Adaptive Boost, Extreme Gradient Boost</i> dan <i>Random Forest</i>
3	(Lalwani dkk., 2022)	<i>Adaptive Boost, Extreme Gradient Boost</i> dan <i>Categorical Boost</i>
4	(Adhitya dkk., 2023)	<i>Classification And Regression Tree</i> dan <i>Naive Bayes</i>
5	(Fariz Sidki, 2024)	<i>Gradient Boost, Extreme Gradient Boost,</i> dan <i>Adaptive Boost</i>

Pada penelitian ini menggunakan algoritma *Gradient Boost, Extreme Gradient Boost, dan Adaptive Boost* yang belum digunakan pada penelitian sebelumnya dengan kasus yang sama yakni prediksi *churn* pelanggan telekomunikasi.

Adapun kebaruan pada penelitian ini berdasarkan penelitian-penelitian serupa yang membandingkan algoritma *Gradient Boost, Extreme Gradient Boost, dan Adaptive Boost* adalah algoritma *boosting* tersebut akan diimplementasikan pada dua karakteristik data *imbalanced* dan *balanced* dengan *SMOTE* dan juga

atribut atau variabel data yang digunakan lebih banyak sehingga data pada penelitian ini menjadi lebih kompleks tertera pada tabel 2.3.

Tabel 2.3 Penelitian dengan algoritma yang sama

No	Peneliti, Tahun	Tujuan / Masalah Penelitian	Karakteristik Data	Variabel Data
1	(Gohil & Meniya, 2021)	Deteksi Penipuan Iklan Klik	<i>Imbalanced</i>	8
2	(Pham & Ho, 2021)	Prediksi Kebrangskutan Bank	<i>Balanced</i>	21
3	(Siswoyo dkk., 2021)	Deteksi tingkat kontrol bank	<i>Imbalanced</i>	5
4	(Iqbal dkk., 2022)	Deteksi Tumor Payudara	<i>Imbalanced</i>	11
5	(Fariz Sidki, 2024)	Prediksi Churn Pelanggan Telekomunikasi	<i>Imbalanced dan Balanced</i>	45