BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 di bawah ini merupakan 10 penelitian terdahulu yang berkaitan dengan *customer churn* dan menggunakan algoritma Decision Tree, Random Forest, dan XGBoost untuk dijadikan sebagai referensi dalam penelitian ini.

Tabel 2. 1 Penelitian Terdahulu

Artikel Jurnal	Nama Jurnal	Penulis/Tahun	Algoritma	Hasil
Customer churn prediction in telecom using machine learning in big data platform	Journal of Big Data	Abdelrahim Kasem Ahmad, Assef Jafar, Kadan Aljoumaa/2019	XGBoost, Decision Tree, Random Forest, GBM	Hasil yang diperoleh adalah, algoritma XGBoost dan Gradient Boosting Machine dapat memberikan performa terbaik tanpa harus ada teknik rebalancing. Untuk algoritma Decision Tree dan Random Forest, performa terbaik dapat dicapai dengan menggunakan teknik undersampling.
Perbandingan Metode Jaringan Syaraf Tiruan dan Pohon Keputusan Untuk Prediksi Churn	Jurnal Sistem Informasi	Helena Nurramdhani Irmanda, Ria Astriratma, Sarika Afrizal/2019	Jaringan saraf tiruan, pohon keputusan	Hasil akurasi, presisi, dan recall diperoleh setelah memilih beberapa atribut penting. Diperoleh akurasi, presisi, dan recall oleh algoritma jaringan syaraf tiruan secara berturut-turut 86%, 71%, dan 50%. Untuk algoritma decision tree, akurasi, presisi, dan recall secara berturut-turut adalah 81%, 52%, dan 58%.
Predicting Customer Retention using XGBoost and	International Journal of Advanced Computer	Atallah M. AL-Shatnwai, Mohammad Faris/2020	Random Forest, SVM, XGBoost, Logistic	Hasil yang diperoleh adalah akurasi sebesar 95,6%, presisi sebesar 92,4%, *recall* sebesar

Artikel Jurnal	Nama Jurnal	Penulis/Tahun	Algoritma	Hasil
Balancing Methods	Science and Applications		Regression, SGD	75,2%, dan F1 *measure* sebesar 82,9%. Hasil ini diperoleh dengan menggunakan algoritma XGBoost yang dinilai merupakan *classifier* terbaik setelah dikomparasikan dengan beberapa algoritma lain seperti SVM, Logistic Regression, dan SGD.
Analysis Implementation of the Ensemble Algorithm in Predicting Customer Churn in Telco Data: A Comparative Study	Informatica	Renny Puspita Sari, Ferdy Febriyanto, Ahmad Cahyono Adi/2023	AdaBoost, Gradient Boost, XGBoost, CatBoost, LightGBM	Hasil yang diperoleh adalah akurasi sebesar 81,2%, recall sebesar 91%, presisi sebesar 84%, dan F1 score sebesar 88%. Hasil tersebut diperoleh dengan menggunakan algoritma XGBoost yang memiliki performa terbaik dibandingkan algoritma lainnya seperti Adaboost, Gradient Boost, CatBoost, dan LightGBM.
Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms	International Journal of Engineering Research & Technology	V. Kavitha, G. Hemanth Kumar, S. V Mohan Kumar, M. Harish/2020	Random Forest, Logistic Regression, XGBoost	Random Forest memberikan hasil terbaik dibandingkan 2 classifier lainnya, yaitu XGBoost dan Logistic Regression. Akurasi yang diperoleh dengan menggunakan Random Forest adalah 80%.
Customer churn analysis in banking sector: Evidence from explainable machine learning models	Journal of Applied Microecono metrics	Hasraddin Guliyev, Ferda Yerdelen Tatoğlu/2021	Logistic Regression, Decision Tree, Random Forest, XGBoost	Di antara 4 algoritma machine learning yang digunakan (Logistic Regression, Decision Tree, Random Forest, dan XGBoost), XGBoost memberikan performa terbaik dengan akurasi sebesar 96,97%

Artikel Jurnal	Nama Jurnal	Penulis/Tahun	Algoritma	Hasil
				disusul dengan Random Forest pada posisi kedua dengan akurasi sebesar 96,75%, Decision Tree pada posisi ketiga dengan akurasi sebesar 96,32% dan terakhir Logistic Regression dengan akurasi sebesar 94,10%.
Predictive Analysis of Customer Churn in Telecom Industry using Supervised Learning	ICTACT Journal on Soft Computing	Shreyas Rajesh Labhsetwar/2020	Logistic Regression, Gaussian Naïve Bayes, AdaBoost Classifier, XGB Classifier, SGD Classifier, Extra Trees Classifier, SVM	Berdasarkan 7 algoritma machine learning yang digunakan, algoritma dengan performa terbaik diperoleh Extra Trees Classifier dengan akurasi 93,74%, disusul oleh XGBoost Classifier dengan akurasi 92,60%.
Churn Prediction for Savings Bank Customers: A Machine Learning Approach	Journal of Statistics Applications & Probability	Prashant Verma/2020	Random Forest, XGBoost, ANN, GLM	Algoritma Random Forest berhasil menjadi algoritma paling efisien dalam melakukan prediksi dengan akurasi di angka 78%. XGBoost adalah algoritma kedua yang memiliki performa cukup baik dengan akurasi 72% namun akurasinya menurun saat melakukan validasi.
Predicting customers churning in banking industry: A machine learning approach	Indonesian Journal of Electrical Engineering and Computer Science	Amgad Muneer, Rao Faizan Ali, Amal Alghamdi, Shakirah Mohd Taib, Ahmed Almaghthawi, Ebrahim Abdulwasea Abdullah Ghaleb/2022	Random Forest, AdaBoost, SVM	Dari 3 algoritma yang digunakan (Random Forest, AdaBoost, SVM), Random Forest merupakan algoritma dengan performa terbaik dengan akurasi sebesar 88,7%.
An Effective Classifier for Predicting Churn in	Journal of Advanced Research in Dynamical	J. Pamina, J. Beschi Raja, S. Sathya Bama, S. Soundarya, M.S. Sruthi, S.	KNN, Random Forest, XGBoost	Digunakan 3 algoritma, yaitu K-Nearest Neighbour, Random Forest, dan XGBoost. XGBoost

Artikel Jurnal	Nama Jurnal	Penulis/Tahun	Algoritma	Hasil
Telecommunication	& Control Systems	Kiruthika, V.J. Aiswaryadevi, G. Priyanka/2019		memiliki performa yang lebih baik dibandingkan kedua algoritma lainnya dengan akurasi sebesar 79,8%.

Berdasarkan 10 penelitian terdahulu mengenai *customer churn* pada berbagai sektor, didapatkan bahwa 8 dari 10 penelitian menggunakan algoritma XGBoost, 7 dari 10 penelitian menggunakan algoritma Random Forest, dan 3 dari 10 penelitian menggunakan algoritma Decision Tree.

Penelitian yang dilakukan oleh [11], [12], [3], [4], dan [13] memberikan hasil bahwa XGBoost merupakan algoritma dengan performa terbaik dibandingkan dengan algoritma lain yang mereka gunakan. Pada sisi lain, Random Forest menjadi algoritma dengan performa terbaik pada penelitian yang dilakukan oleh [14], [15], dan [16]. Walaupun Decision Tree tidak menjadi algoritma terbaik dalam 10 penelitian tersebut, algoritma ini masih sering digunakan untuk dijadikan komparasi seperti yang dapat ditemukan pada [11], [17], dan [4].

Algoritma Decision Tree, Random Forest, dan XGBoost digunakan karena dapat menyelesaikan permasalahan klasifikasi dan memiliki kemiripan dalam mekanisme kerjanya. XGBoost dan Random Forest merupakan algoritma yang berdasar pada Decision Tree. Perbedaannya adalah XGBoost menerapkan teknik *boosting* [3] dan Random Forest menerapkan teknik *bagging* terhadap Decision Tree [18] dengan tujuan untuk mendapatkan performa yang lebih baik dibandingkan Decision Tree.

2.2 Customer Churn

Customer churn secara sederhana dapat didefinisikan sebagai pelanggan dari suatu perusahaan atau bisnis yang memutuskan untuk tidak lagi menjadi seorang pelanggan dari perusahaan atau bisnis tersebut [1]. Hal ini dapat ditunjukkan oleh pelanggan dengan tidak lagi melakukan pembelian atau berhenti

berlangganan pada suatu perusahaan/bisnis dan berpaling melakukan pembelian atau berlangganan ke perusahaan/bisnis lain [13].

Customer churn pada sektor pendidikan dikenal sebagai *student attrition*. Hal ini tentunya dapat mempengaruhi kualitas, reputasi, bahkan kesuksesan Universitas Multimedia Nusantara. Mahasiswa dapat dikategorikan *churn* saat mereka tidak lagi memiliki hubungan apapun dengan Universitas Multimedia Nusantara.

2.3 Framework dan Algoritma

2.3.1 CRISP-DM

Gambar 2. 1 *CRISP-DM Methodology*

Sumber: Medium (2023)

Gambar 2.1 merupakan flow dari *Cross-Industry Standard Process for Data Mining* (CRISP-DM). CRISP-DM merupakan standar dilakukannya data *mining* yang paling populer. CRISP-DM memiliki 6 tahapan di dalamnya, yaitu *business understanding*, data *understanding*, data *preparation*, *modeling*, *evaluation*, dan *deployment* [19].

2.3.1.1 Business Understanding

Sebagai tahap pertama dalam *framework* CRISP-DM, tahap *business understanding* berfokus pada pemahaman dan tujuan bisnis serta tujuan dilakukannya data *mining* terhadap bisnis tersebut. Pada tahap ini, perlu juga untuk mengetahui bagaimana nantinya data akan didapatkan untuk proses data *mining*. Dilakukannya tahap *business understanding* akan menghasilkan model *machine learning* terbaik, sesuai dengan tujuan bisnis yang telah ditentukan.

2.3.1.2 Data Understanding

Data *understanding* berfokus pada memahami data yang akan digunakan untuk melakukan penelitian. Pada tahap ini, pekerjaan utama yang akan dilakukan adalah melakukan *explore* terhadap data serta mendeskripsikan data. Pengecekan terhadap kualitas data juga merupakan hal yang penting untuk dilakukan pada tahap ini. Atribut dalam data juga akan ditentukan dan dideskripsikan.

2.3.1.3 Data Preparation

Data *preparation* atau disebut juga sebagai data *preprocessing* berfokus pada menyiapkan data untuk nantinya digunakan. Data yang telah diperiksa kualitasnya pada tahap sebelumnya dan ternyata memiliki kualitas yang buruk/*bad quality* akan diperbaiki pada tahap ini. Kualitas data diperbaiki dengan cara membersihkan data dari *null*, *missing value*, *noisy* data, dan *outlier*.

2.3.1.4 Modeling

Modeling merupakan tahapan dalam CRISP-DM yang berfokus untuk membentuk model berdasarkan algoritma yang telah ditentukan untuk digunakan. Teknik data *mining* yang ingin

digunakan dapat disesuaikan dengan masalah yang ada pada bisnis serta data yang dimiliki.

2.3.1.5 Evaluation

Evaluation berfokus pada melakukan evaluasi terhadap model yang sudah dibuat pada tahap *modeling*. Evaluasi dapat dilakukan dengan cara melihat akurasi model, presisi model, *recall*, dan *f-measure* atau *f1-score* dari model yang telah dibuat.

2.3.1.6 Deployment

Deployment merupakan tahapan terakhir dalam CRISP-DM yang berfokus pada menerapkan model yang telah dibuat ke dalam data.

2.3.2 Klasifikasi

Klasifikasi dalam *machine learning* merupakan sebuah pendekatan dari *supervised learning*. Klasifikasi dapat dikatakan juga sebagai sebuah proses pengkategorisasian data ke dalam beberapa kelas. Proses klasifikasi dimulai dengan cara memprediksi kelas dari data yang diberikan. Kelas dalam klasifikasi sering disebut sebagai target, label, atau kategori. Tujuan dari dilakukannya klasifikasi tentunya adalah mengidentifikasi kelas apa yang cocok dengan data yang diberikan [20].

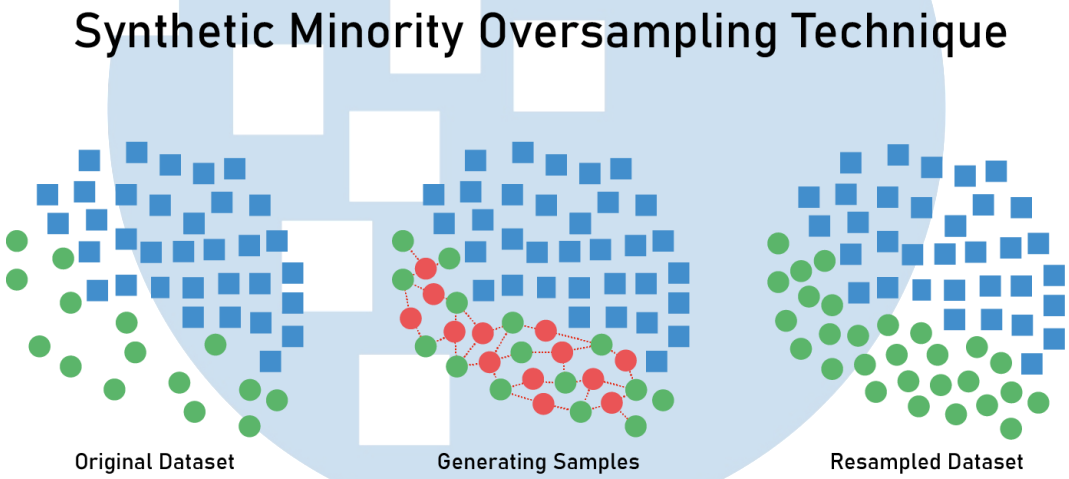
Pengimplementasian klasifikasi dapat menggunakan beberapa algoritma. 3 algoritma klasifikasi yang digunakan pada penelitian ini antara lain adalah Decision Tree, Random Forest, dan XGBoost.

2.3.3 Oversampling

Pada masalah klasifikasi, sering ditemukan adanya kelas minoritas dan kelas mayoritas di dalam sebuah dataset. Hal ini disebut juga sebagai *imbalanced dataset*. *Imbalanced dataset* dapat menyebabkan terjadinya

bias terhadap kelas mayoritas yang mengakibatkan performa yang buruk pada klasifikasi kelas minoritas [4].

Oversampling memiliki kegunaan dalam menyeimbangkan jumlah data pada kelas minoritas dan kelas mayoritas. Oversampling dapat dilakukan salah satunya dengan menggunakan *Synthetic Minority Oversampling Technique* (SMOTE).



Gambar 2. 2 SMOTE

Sumber: [21]

SMOTE berfokus dalam meningkatkan jumlah data pada kelas minoritas dengan membuat data sintetik yang sangat mirip dengan data asli pada kelas minoritas tersebut [3]. Gambar 2.2 di atas merupakan bagaimana cara SMOTE membuat data baru, yaitu dengan cara interpolasi [21].

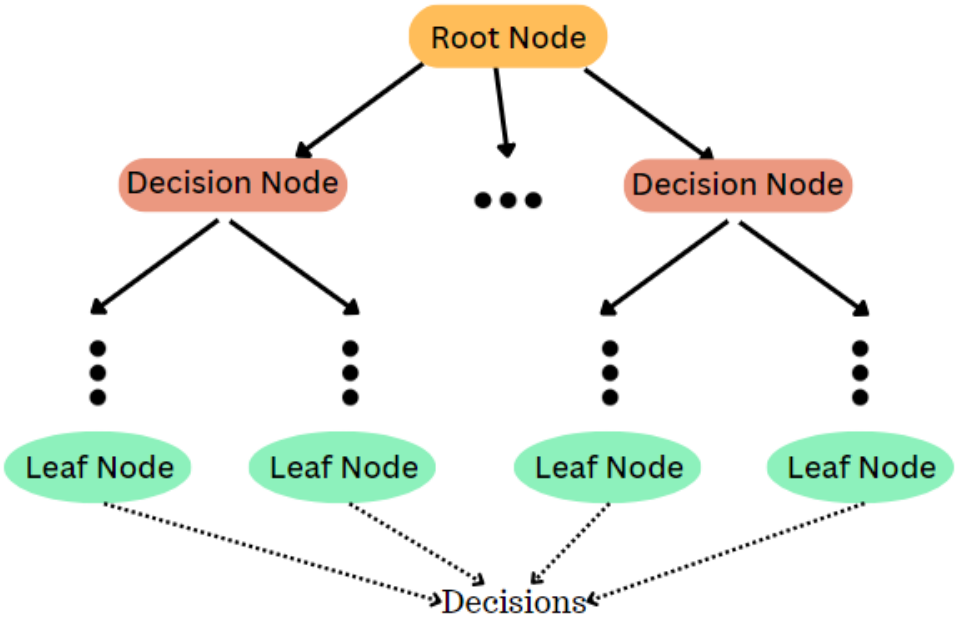
2.3.4 Algoritma

Decision Tree, XGBoost, dan Random Forest merupakan tiga algoritma yang digunakan dalam penelitian ini. Ketiga algoritma tersebut mampu menyelesaikan permasalahan klasifikasi dan memiliki mekanisme kerja yang serupa. Decision Tree merupakan algoritma dasar dari kedua algoritma lainnya, yaitu XGBoost dan Random Forest. XGBoost menerapkan teknik *boosting* [3], sedangkan Random Forest di sisi lain menerapkan teknik *bagging* terhadap Decision Tree [18]. Tujuan

diterapkannya teknik *boosting* dan *bagging* adalah untuk memperoleh performa yang lebih baik dibandingkan algoritma dasarnya, yaitu Decision Tree.

2.3.4.1 Decision Tree

Decision Tree merupakan sebuah algoritma *supervised learning* yang dapat menyelesaikan masalah klasifikasi dan regresi. Dapat dikatakan bahwa Decision Tree merupakan algoritma yang cukup mudah untuk digunakan bagi para pemula di bidang *machine learning*. Algoritma ini memiliki struktur seperti pohon hierarkis yang di dalamnya terdapat sebuah *root node*, cabang, beberapa *decision node*, dan beberapa *leaf node*. Prediksi yang dihasilkan oleh model yang dibuat menggunakan Decision Tree berasal dari fitur-fitur yang ada di dalam data yang diberikan [22].



Gambar 2. 3 Decision Tree *Classifier*

Sumber: Medium (2023)

Gambar 2.3 di atas menunjukkan cara kerja dari algoritma Decision Tree. Cara kerja algoritma ini diawali dari *root node* yang melambangkan keseluruhan data yang diberikan. Berdasarkan data tersebut, fitur terpenting yang memecah data ke dalam beberapa kelompok akan dicari dan akan digunakan untuk membuat pertanyaan yang menyerupai *if else* hingga hasil prediksi didapatkan pada *leaf node* [22]. Pencarian fitur sebagai pemisah tersebut dicari dengan menggunakan perhitungan Gini *index*.

$$Gini\ Index = 1 - \sum_{i=1}^{n} \left(P_i\right)^2$$

$$= 1 - [(P_+)^2 + (P_-)^2]$$

Rumus 2. 1 Gini *Index*

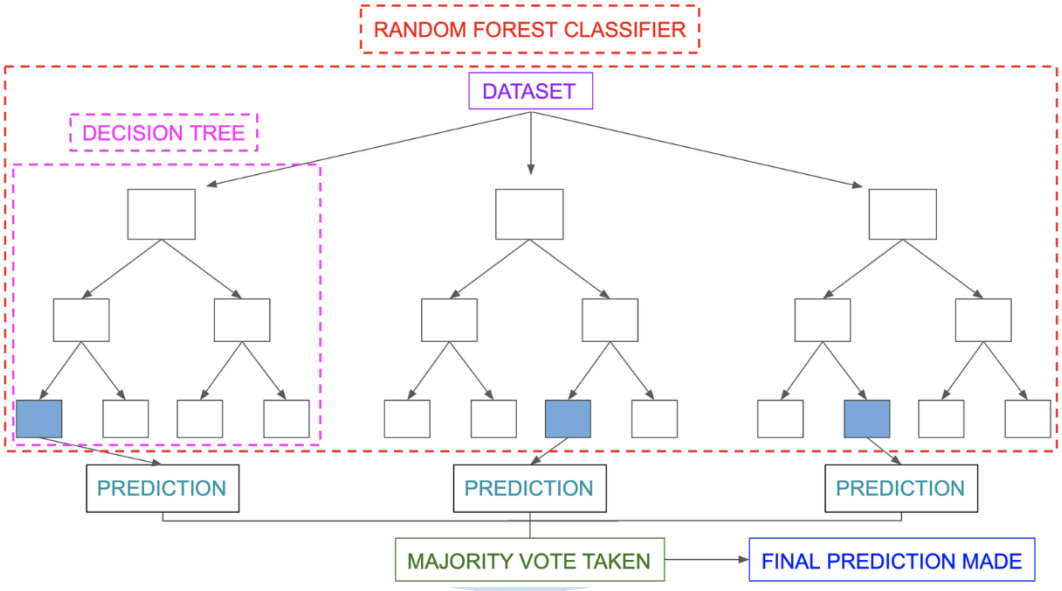
Sumber: [23]

Rumus 2.1 merupakan persamaan matematika dalam mencari Gini *index*. Pada persamaan tersebut, P_+ merupakan probabilitas kelas positif dan P_- merupakan probabilitas kelas negatif. Perhitungan Gini *index* akan dilakukan terhadap seluruh kemungkinan pemisahan. Hasil Gini *index* terendah atau disebut juga sebagai *low impurity* akan dipilih [23].

2.3.4.2 Random Forest

Algoritma Random Forest merupakan salah satu algoritma *supervised learning*. Random Forest juga merupakan algoritma yang berbasis Decision Tree [24]. Algoritma ini dapat digunakan untuk menyelesaikan masalah klasifikasi dan regresi.

Di dalam sebuah algoritma Random Forest, terdapat banyak pohon keputusan di dalamnya (*forest*). *Forest* yang dihasilkan oleh algoritma Random Forest ini dilatih melalui *bootstrap aggregating* atau *bagging*. *Bagging* merupakan sebuah algoritma *ensemble* yang meningkatkan akurasi dari algoritma *machine learning* [18].



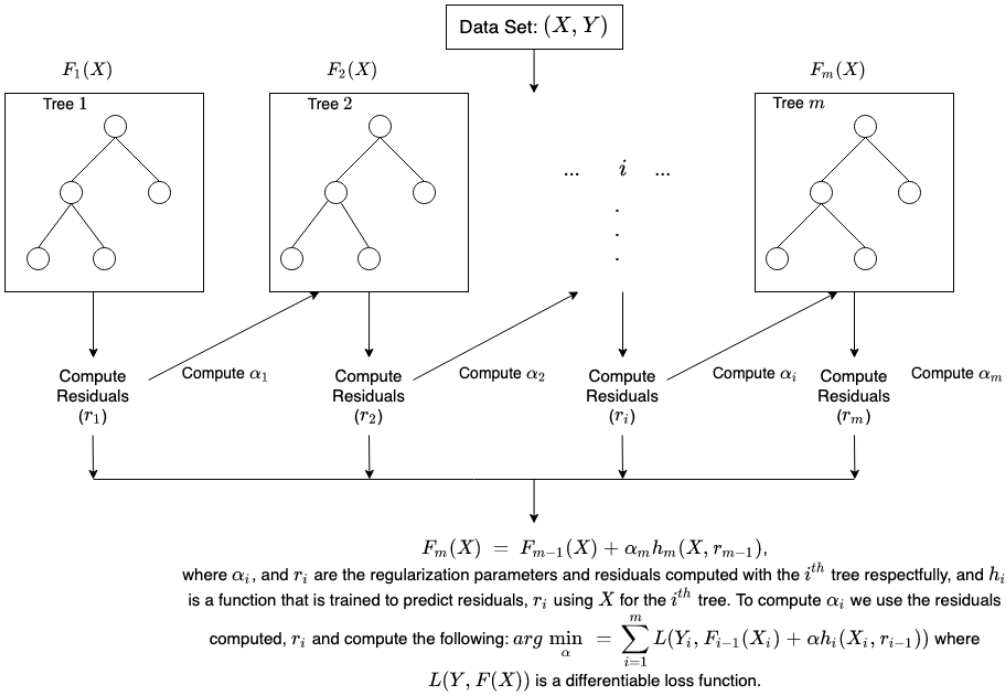
Gambar 2. 4 Random Forest *Classifier*

Sumber: [18]

Sebagaimana telah dikatakan sebelumnya, algoritma Random Forest terdiri dari banyak pohon keputusan sehingga disebut sebagai *forest*. Data yang diberikan akan dibagi ke dalam beberapa bagian dan setiap bagian data akan menghasilkan sebuah pohon keputusan. Pemilihan fitur sebagai pemisah pada setiap pohon menggunakan perhitungan yang sama seperti yang tertera pada rumus 2.1. Pada akhirnya, setiap pohon keputusan akan menghasilkan sebuah *output*. *Output* mayoritas akan menjadi *output final* yang sesungguhnya seperti yang dapat dilihat pada gambar 2.4 [18].

2.3.4.3 XGBoost

XGBoost merupakan sebuah versi algoritma yang lebih baik dari algoritma gradient boosting. Algoritma XGBoost dapat digunakan untuk menyelesaikan masalah klasifikasi maupun regresi. XGBoost bekerja dengan cara mendorong (*boost*) *weak learners* sehingga menjadi *stronger learners* dengan menggunakan mekanisme Decision Tree [3].



Gambar 2. 5 XGBoost *Classifier*

Sumber: [25]

Cara kerja XGBoost digambarkan seperti algoritma *tree* yang mengambil fitur yang ada di dalam data sebagai *conditional node*. *Conditional node* ini nantinya akan terbagi menjadi berbagai cabang dan terpecah hingga menjadi sebuah *leaf node* yang merepresentasikan deteksi masalah yang dipilih [3]. XGBoost juga bekerja dengan cara menghitung residual saat proses *training* model seperti yang dapat dilihat pada gambar 2.5. Residual akan dihitung

dari setiap pohon yang terbentuk. Dalam proses pembentukan pohon yang berjalan secara iteratif, penjumlahan akan dilakukan secara berlanjutan untuk setiap residual dari tiap pohon hingga nantinya mencapai angka yang berada di atas atau di bawah *threshold*. Angka ini akan menjadi hasil akhir dalam penentuan suatu data poin untuk masuk ke dalam suatu kelas [25]. Penggunaan XGBoost dapat memberikan beberapa keuntungan seperti proses pembuatan model yang cepat, menghasilkan model dengan performa yang baik, dan mengurangi terjadinya *error* pada model [26].

2.3.5 Confusion Matrix

Confusion matrix merupakan sebuah metode dalam data *mining* yang berfungsi untuk melakukan pengukuran terhadap akurasi yang dihasilkan suatu model terhadap data. Dalam *confusion matrix*, terdapat 4 nilai yang digunakan yang mendeskripsikan performa dari klasifikasi yang telah dilakukan oleh suatu model. 4 nilai tersebut antara lain adalah *true negative* (TN), *true positive* (TP), *false negative* (FN), dan *false positive* (FP) [27].

		True Values	
		True	False
Prediction	True	TP Correst result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

Gambar 2. 6 *Confusion Matrix*

Sumber: [27]

Gambar 2.6 di atas memperlihatkan 4 values yang terdapat dalam sebuah confusion matrix, yaitu:

- *True negative* (TN) merupakan sebuah *value* yang menandakan bahwa prediksi negatif terhadap data adalah benar.

- *True positive* (TP) merupakan sebuah *value* yang menandakan bahwa prediksi positif terhadap data adalah benar.
- *False negative* (FN) merupakan sebuah *value* yang menandakan bahwa data positif diprediksi sebagai negatif.
- *False positive* (FP) merupakan sebuah *value* yang menandakan bahwa data negatif diprediksi sebagai positif.

Performance pada *confusion matrix* dapat diukur melalui presisi, *recall*, dan akurasi [27].

2.3.6 Precision

Precision/*confidence* menunjukkan jumlah dari prediksi positif yang memang sesungguhnya adalah positif [27], [28]. Rumus 2.2 di bawah merupakan rumus untuk menghitung nilai *precision*. Nilai *precision* dapat diperoleh dengan membagi jumlah *true positive* dengan total dari *true positive* ditambah dengan *false positive*.

$$Precision = \frac{TP}{TP + FP}$$

Rumus 2. 2 Rumus *Precision*

Sumber: [27]

2.3.7 Recall

Recall/*sensitivity* menunjukkan jumlah dari data positif yang diprediksi positif atau dapat dikatakan juga sebagai alat pengukuran terhadap seberapa baik suatu model dalam melakukan prediksi positif terhadap data positif. Perlu diperhatikan bahwa recall hanya fokus terhadap pengklasifikasian data positif dan sama sekali tidak memperhatikan data negatif [27], [28]. Rumus 2.3 di bawah merupakan cara mendapatkan nilai *recall*. *Recall* dapat diperoleh dengan membagi *true positive* dengan total dari *true positive* ditambah dengan *false negative*.

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2. 3 Rumus *Recall*

Sumber: [27]

2.3.8 F-measure

F-measure atau disebut juga sebagai *f1-score* merupakan sebuah nilai rata-rata harmonik yang berasal dari *precision* (P) dan *recall* (R). Keterkaitan antara nilai *f-measure*, *precision*, dan *recall* akan memberikan sebuah kesimpulan bahwa saat nilai *f-measure* yang diperoleh tinggi, maka nilai *precision* dan *recall* pada suatu model juga baik [29]. Rumus 2.4 di bawah merupakan cara untuk mendapatkan nilai *f-measure*. Nilai *f-measure* dapat diperoleh dengan cara mengalikan dengan 2 nilai *precision* dan *recall* yang sebelumnya telah dikali dan dibagi dengan nilai *precision* yang telah dijumlah dengan *recall*.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

Rumus 2. 4 Rumus *f-measure*

Sumber: Towards Data Science (2020)

2.3.9 Accuracy

Accuracy merujuk pada persentase prediksi benar dari total data yang ada. Tingkat akurasi yang tinggi menunjukkan bahwa data dari hasil prediksi memiliki tingkat kedekatan yang tinggi dengan data sebenarnya [27]. Rumus 2.5 di atas merupakan cara untuk mendapatkan nilai akurasi. Nilai akurasi dapat diperoleh dengan membagi hasil penjumlahan *true positive* dan *true negative* dengan hasil penjumlahan *true positive*, *true*

negative, *false positive*, dan *false negative*. Nilai tersebut kemudian akan dikalikan dengan 100%.

$$Accuration = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \times 100\%$$

Rumus 2. 5 Rumus *Accuracy*

Sumber: [27]

2.4 Tools

2.4.1 Microsoft Excel

Microsoft Excel merupakan sebuah *software* buatan Microsoft pada tahun 1985. Excel digunakan oleh banyak bisnis karena kemampuannya dalam menyimpan data dalam *spreadsheet*, membuat *chart*, grafik, serta melakukan komputasi statistikal terhadap data. Excel merupakan ciptaan Microsoft, maka akan sangat mudah untuk menghubungkan Excel dengan *software* Microsoft lainnya, seperti misalnya Power BI. Hal ini membuat Excel masih menjadi pilihan banyak orang termasuk data *analyst* [30].

2.4.2 Visual Studio Code

Visual Studio Code (VS Code) merupakan sebuah *platform* yang dikembangkan oleh Microsoft serta dinominasikan sebagai *Most Popular Development Environment* pada *Stack Overflow Developer Survey* di tahun 2019. Visual Studio Code dapat digunakan secara gratis, bersifat open source dan *cross-platform*. VS Code memiliki fitur yang melimpah dan mendapatkan pembaruan setiap bulannya disertai dengan *bug fixes*. Pemrograman berbasis Python juga dapat dilakukan dengan menggunakan VS Code [31].

2.4.3 Python

Python merupakan sebuah bahasa pemrograman yang sifatnya *object-oriented*. Python juga digolongkan sebagai bahasa pemrograman

tingkat tinggi. *Syntax* yang ada dalam bahasa pemrograman ini juga cenderung sederhana dan mudah untuk dipelajari. *Library* pada Python juga dapat dengan bebas digunakan dan tidak dikenakan biaya apapun. Dengan menggunakan Python, dikatakan bahwa pengguna merasakan peningkatan produktivitas yang dapat dirasakan karena tidak ada tahapan kompilasi serta proses *edit*, *test*, dan *debug* yang sangat cepat. Selain itu, saat *programmer* melakukan input yang salah Python hanya akan memunculkan sebuah *exception* tanpa adanya *segmentation fault* [32].