BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Teknologi Finansial

Teknologi finansial atau *Financial Technology* merupakan suatu integrasi antara teknologi dengan jasa keuangan yang mengubah bentuk bisnis dari industri keuangan[21]. Transaksi keuangan yang dilakukan saat ini dapat lebih mudah dan cepat dibandingkan sebelumnya dengan berbagai kemudahan yang diberikan oleh bank maupun non bank yang membuat kegiatan ekonomi dapat dilakukan secara luas.

Selain dampak positif, perkembangan teknologi finansial juga memiliki dampak negatif. Beberapa dampak dari perkembangan teknologi ini berdampak pada kejahatan finansial dengan menggunakan instrumen keuangan. Proses kejahatan ini memanfaatkan sistem pembayaran secara elektronik dan memiliki prinsip kerahasiaan sehingga membuat pelaku kejahatan semakin mudah dalam melakukan tindak pidana[22]. Adapun beberapa kejahatan finansial yang dilakukan di antaranya adalah pencucian uang, penyebaran data pribadi, kecurangan, pendanaan terorisme serta penipuan yang dapat dilakukan dengan cepat[23].

Salah satu cara dalam mencegah terjadinya tindak kejahatan dalam finansial teknologi yaitu dengan melakukan verifikasi identitas. *Customer Due Diligence* (CDD) merupakan salah satu cara untuk mengurangi risiko keuangan melalui identitas dari pengguna layanan keuangan.

2.1.2 *Customer Due Diligence*

Customer Due Diligence atau CDD adalah sebuah kegiatan yang digunakan untuk identifikasi, verifikasi, dan analisa pelanggan untuk mencegah terjadinya pencucian uang (*money laundering*) atau pendanaan terorisme (*terrorism financing*)[24]. Regulasi Otoritas Jasa Keuangan (OJK) yang mengatur dalam proses *customer due diligence* pada POJK Nomor 23/POJK.01/2019 Tentang Perubahan Atas Peraturan Otoritas Jasa Keuangan Nomor 12/POJK.01/2017 Tentang Penerapan Program Anti Pencucian Uang dan Pencegahan Pendanaan Terorisme di Sektor Jasa Keuangan[25], [26]. Regulasi tersebut menyatakan bahwa setiap penyedia jasa keuangan wajib melakukan *customer due diligence* terhadap pemilik manfaat. Proses ini akan melakukan penilaian terhadap tingkat kriminalitas finansial dengan cara melakukan verifikasi identitas dari *customer* dan menilai potensi risiko sehingga perusahaan dapat menurunkan risiko dalam melakukan fasilitasi kejahatan pencucian uang atau pendanaan terorisme[24].

Customer due diligence membutuhkan data yang akurat untuk proses identifikasi risiko dalam finansial. Salah satu data yang akurat yaitu dengan menggunakan media massa sebagai sumber data dari *customer due diligence*. Hasil dari analisa data media massa akan menghasilkan penilaian risiko dari suatu perusahaan maupun individu.

2.1.3 Media Massa

Media massa adalah sebuah sarana dalam komunikasi dan informasi yang digunakan dalam menyebarkan informasi kepada banyak orang. Media massa juga digunakan sebagai alat untuk menyebarkan opini, berita, komentar, hiburan dan lainnya. Media massa digunakan untuk menyampaikan pesan dari sumber kepada masyarakat dalam bentuk alat komunikasi seperti televisi, radio, surat kabar serta film[27]. Perkembangan teknologi membuat perkembangan media massa menjadi pesat sehingga memudahkan masyarakat dalam mendapatkan informasi secara cepat dan

akurat dalam hitungan detik. Media massa juga digunakan sebagai pembangunan citra diri seseorang maupun perusahaan dalam pandangan publik. Contoh dari pembangunan citra diri perusahaan adalah konferensi pers yang digunakan untuk memberikan informasi mengenai suatu topik[28]. Selain itu media juga dapat memberikan pemberitaan negatif pada perusahaan yang dapat merugikan citra perusahaan sehingga mempengaruhi opini publik terhadap suatu perusahaan maupun individu[29].

2.1.4 *Adverse Media*

Adverse Media adalah informasi negatif atau informasi yang tidak menguntungkan bagi bisnis maupun individu[30]. Informasi ini biasanya berada di berbagai jenis media massa. *Adverse Media Screening* adalah proses penyaringan pengguna berdasarkan data yang didapatkan dari *adverse media*. *Adverse Media Screening* juga digunakan dalam proses *customer due diligence* untuk mengurangi risiko pengguna dalam penyalahgunaan teknologi finansial serta kepatuhan terhadap peraturan[30]. Analisa risiko dari *adverse media* dapat dilakukan dengan melakukan pencarian media negatif terhadap suatu perusahaan maupun individu berkaitan dengan aktivitas kejahatan keuangan seperti tindak pidana korupsi, pencucian uang dan pendanaan terorisme[31]. Contoh dari *adverse media* yaitu berita dengan dengan judul "AJI: Pelaku Kekerasan terhadap Jurnalis Paling Banyak Polisi"[32].

2.1.5 *Classification*

Classification atau klasifikasi adalah jenis dari *supervised machine learning* yang digunakan untuk melakukan prediksi kelas maupun label[33]. *Machine learning* ini biasanya digunakan untuk memecahkan masalah prediksi label atau kelas berdasarkan data yang diberikan. Contoh dalam penyelesaian masalah dengan menggunakan *machine learning* dengan jenis klasifikasi adalah pendeteksi spam, klasifikasi hewan dan tumbuhan, serta analisa sentimen[33]. Berikut adalah beberapa jenis dari *classification* pada machine learning.

2.1.5.1 *Binary Classification*

Binary Classification adalah klasifikasi yang digunakan untuk membedakan dua kelas atau label[34]. Contoh dari klasifikasi ini adalah spam dan tidak spam. Terdapat algoritma yang dapat menangani klasifikasi jenis ini diantaranya yaitu k-nearest neighbors, decision trees, random forest, Bayesian networks, neural networks, dan support vector machine.

2.1.5.2 *Multi-class Classification*

Multi-class Classification adalah klasifikasi yang digunakan untuk membedakan lebih dari dua kelas atau label[34]. Contoh dari klasifikasi ini seperti klasifikasi berita berdasarkan kategori, klasifikasi dokumen berdasarkan jenisnya dan lainnya. Terdapat beberapa algoritma yang dapat digunakan untuk melakukan klasifikasi jenis ini diantaranya yaitu decision tree, naïve bayes, random forest, gradient boosting, k-nearest neighbors.

2.1.5.3 *Multi-label Classification*

Multi-label Classification adalah klasifikasi dengan kelas atau label lebih dari satu[34]. Contoh dari klasifikasi ini adalah artikel yang memiliki lebih dari satu topik seperti teknologi dan bisnis. Terdapat beberapa algoritma yang dapat digunakan dalam proses klasifikasi ini. Algoritma yang digunakan adalah dengan menggunakan algoritma dengan versi *multi-label* yaitu multi-label decision trees, multi-label boosting, dan multi-label random forest.

2.1.5.4 *Imbalanced Classification*

Imbalanced Classification adalah klasifikasi dengan label atau kelas tidak seimbang secara proporsional[34]. Hal ini menjadi salah satu tantangan pada *machine learning* yang pada umumnya bertujuan untuk mendapatkan akurasi. Klasifikasi ini sebagian besar memiliki data minor yang dibutuhkan namun sering membuat

algoritma *machine learning* melakukan prediksi pada data utama. Contoh dari klasifikasi ini adalah tes diagnosa penyakit, deteksi penipuan, dan deteksi *outlier*.

2.1.6 Model Evaluation

Evaluasi model dilakukan dengan melihat nilai dari *precision, recall, f1-score,* dan *accuracy* [35]. Nilai ini dapat menentukan kualitas dari model yang dihasilkan serta bagaimana model dapat melakukan prediksi dengan tepat. Proses evaluasi dilakukan dengan melihat *confusion matrix* dengan menghitung nilai dari *true positive, true negative, false positive* dan *false negative*[36]. Hasil *confusion matrix* akan digunakan dalam proses kalkulasi *precision, recall, f1-score* dan *accuracy*.

2.1.6.1 *Accuracy*

Accuracy adalah rasio dari prediksi benar pada data positif maupun data negatif dari keseluruhan data [36]. Akurasi sering digunakan untuk menilai kualitas dari model yang dihasilkan. Rumus dari *accuracy* adalah sebagai berikut.

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

2.1.6.2 *Precision*

Precision adalah rasio dari prediksi benar pada data positif terhadap total hasil yang diprediksi positif [36]. Hal ini untuk mengetahui seberapa besar model dapat menghindari data *false positive*. Rumus dari *precision* adalah sebagai berikut.

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

2.1.6.3 *Recall*

Recall adalah rasio dari prediksi benar dibandingkan dengan data aktual positif[36]. Hal ini digunakan untuk mengetahui seberapa besar model dapat menghindari data *false negative*. Rumus dari *recall* adalah sebagai berikut.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

2.1.6.4 *F1-Score*

F1-Score adalah nilai dari rata-rata harmonik dari *precision* maupun *recall* [37]. Ukuran ini dapat digunakan untuk data yang tidak seimbang dan tingkat akurasi yang tidak akurat. Rumus dari *F1-score* adalah sebagai berikut.

$$\text{F1} - \text{Score} = \frac{2(\frac{\text{TP}}{\text{TP} + \text{FP}}\frac{\text{TP}}{\text{TP} + \text{FN}})}{(\frac{\text{TP}}{\text{TP} + \text{FP}} + \frac{\text{TP}}{\text{TP} + \text{FN}})} \tag{4}$$

2.1.7 TF-IDF

TF-IDF merupakan kombinasi dari dua kata yaitu *Term Frequency* dan *Inverse Document Frequency*[38]. *Term Frequency* digunakan dalam mengukur seberapa sering sebuah kata pada sebuah dokumen teks. Jika sebuah kata muncul lebih sering pada sebuah dokumen maka dapat melakukan penghitungan frekuensi dengan membagi frekuensi kata dengan keseluruhan kata yang ada di dokumen teks. Selanjutnya yaitu *Inverse Document Frequency* akan memberikan bobot rendah pada kata yang sering muncul namun akan memberikan bobot tinggi pada kata yang jarang muncul. Terakhir yaitu TF-IDF adalah melakukan perkalian antara data dari *Term Frequency* dengan *Inverse Document Frequency* dan hasil perkalian tersebut menghasilkan nilai dari TF-IDF.

2.1.8 CountVectorizer

CountVectorizer adalah alat yang digunakan dalam machine learning untuk memproses teks dalam Pemrosesan Bahasa Alami (*Natural Language Processing*) dan mengambil fitur-fitur dari teks tersebut[39]. CountVectorizer digunakan untuk membuat kamus dari data teks yang diberikan. CountVectorizer akan mengubah teks menjadi sebuah matriks yang berisi jumlah kemunculan setiap kata dalam teks tersebut.

2.1.9 Tokenizer

Tokenizer adalah sebuah *class* dari *library* Tensorflow yang memungkinkan pengvektoran korpus teks dengan mengubah setiap teks menjadi urutan bilangan bulat[40]. Setiap bilangan bulat merupakan indeks dari token yang ada di kamus. Semua tanda baca akan dihapus dan mengubah teks menjadi urutan kata yang dipisahkan oleh spasi lalu dipecah menjadi daftar token dan diberikan representasi vektor.

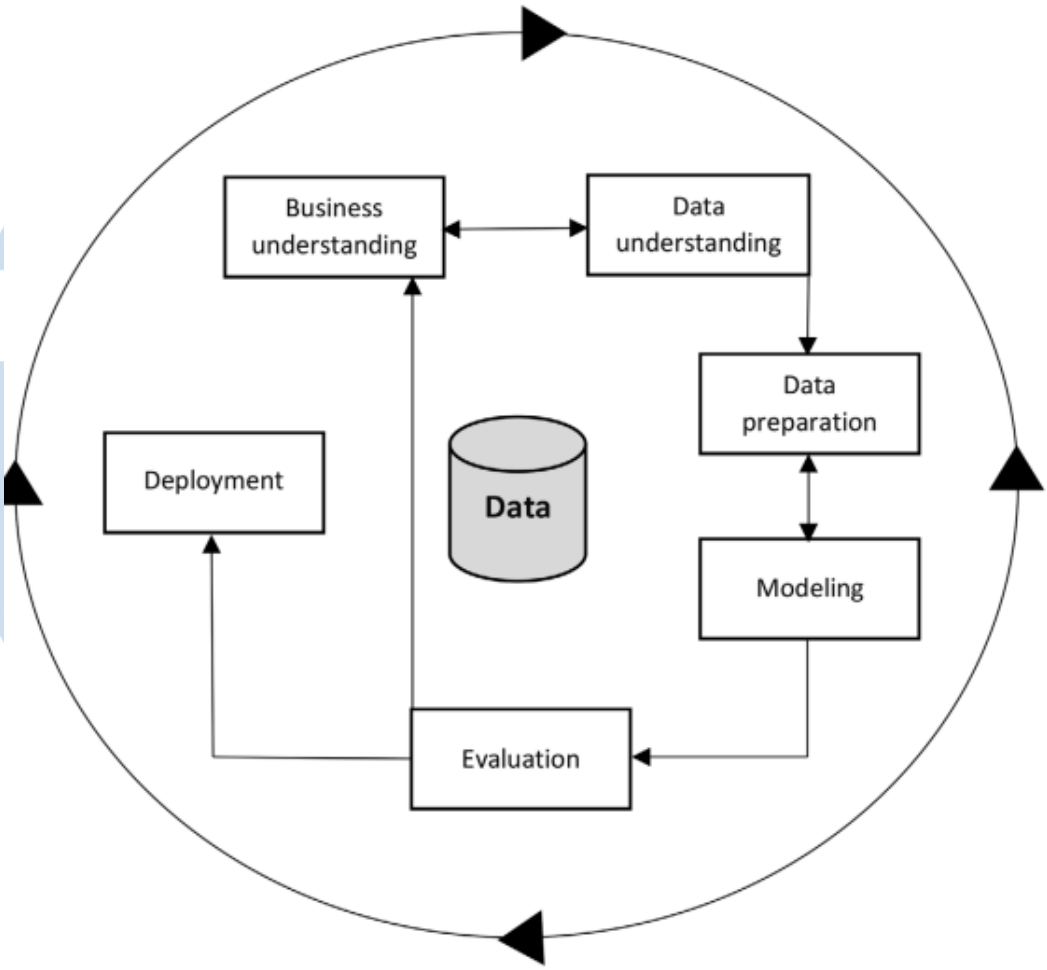
2.1.10 Grid Search

Grid search adalah metode yang digunakan untuk menemukan *hyperparameter* terbaik pada saat pembuatan model *machine learning* [41]. *Grid search* akan melakukan *brute-force* dengan kombinasi yang sudah ditentukan serta mengevaluasi hasil yang didapatkan pada masing-masing percobaan. *Grid search* akan melatih *model* dan membandingkan setiap model yang dilatih dan memilih *model* terbaik.

2.2 Framework dan Algoritma

2.2.1 CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) adalah sebuah *framework* yang dibuat pada tahun 1990-an oleh grup dari lima perusahaan yaitu: SPSS, Daimler AG, NCR, OHRA, dan TeraData. CRISP-DM terdiri dari enam tahapan yaitu *business understanding, data understanding, data preparation, modeling, evaluation,* dan *deployment*[42]. Alur kerja dari CRISP-DM seperti pada Gambar 2.1.



Gambar 2. 1 CRISP-DM *Lifecycle*

Sumber: [42]

1) *Business Understanding*

Business Understanding adalah tahapan pertama dalam CRISP-DM untuk memahami tujuan dari proyek dalam sudut pandang bisnis. Proses Business Understanding sangat penting untuk menentukan langkah yang akan dilakukan pada proses berikutnya [42]

2) *Data Understanding*

Data Understanding adalah proses yang melibatkan dari pengumpulan data, mengidentifikasi permasalahan yang ada pada

data, kualitas dari data yang digunakan, serta identifikasi *insight* yang tersembunyi dari data [42].

3) *Data Preparation*

Tahapan ini adalah tahapan dari penyiapan data yang akan digunakan pada tahapan berikutnya. Tahapan ini meliputi dari pembersihan data, transformasi data untuk digunakan dalam proses analisis dan pengujian berikutnya [42].

4) *Modeling*

Tahapan ini adalah tahapan dari pembuatan model dan pengembangan teknik dari model yang akan dibuat. Tahapan ini menggunakan algoritma yang dipilih dalam pembuatan model. Tahapan ini juga biasanya menggunakan pembagian data untuk proses *training* dan *testing* untuk mempermudah dalam mengetahui kualitas dari model yang dihasilkan [42].

5) *Evaluation*

Tahapan evaluasi adalah tahapan dalam peninjauan model yang dibuat dan interpretasi hasil analisis dalam konteks tujuan bisnis. Tahapan ini akan melihat berbagai *metric* yang digunakan untuk mengetahui kualitas dari model. Jika terdapat ketidaksesuaian maka perlu adanya peninjauan lebih lanjut dalam proses pembuatan model [42].

6) *Deployment*

Tahapan *deployment* adalah tahapan akhir dari CRISP-MD. Tahapan ini adalah penerapan model pada proyek yang berupa sistem maupun dashboard. Tahapan ini akan menggunakan model yang dipilih pada tahapan evaluasi pada penerapan proyek [42].

2.2.2 Gradient Boosting

Friedman menciptakan metode *multiple additive trees* (MAT) atau dikenal sebagai Gradient Boosting [43]. Strategi umum yang dikenal sebagai "boosting" yaitu digunakan untuk meningkatkan akurasi dari algoritma pembelajaran dengan memiliki tingkat kesalahan yang rendah dan menggabungkannya menjadi sebuah *ensemble* yang memiliki kinerja lebih baik. Algoritma Gradient Boosting dapat digunakan dalam klasifikasi maupun regresi[17].

Proses Gradient Boosting dimulai dari melakukan *fitting* data dengan menggunakan model pohon keputusan yang sederhana. Selanjutnya yaitu melakukan pengurangan kesalahan yang dihasilkan dari pohon sebelumnya. Hal ini dilakukan secara terus menerus hingga mendapatkan model yang sesuai dengan data dengan akurasi yang tinggi [43]. Semisal terdapat sampel pelatihan $\{(x_i, y_i)\}_{i=1}^{n}$ dan fungsi loss adalah $\Omega(y, \mathcal{F}(x))$ dengan iterasi ke I. Model dengan rumus berikut akan mendapatkan dengan menerapkan prinsip minimalisasi risiko empiris.

$$\mathcal{F}_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{5}$$

2.2.3 XGBoost

XGBoost atau *Extreme Gradient Boosting* adalah algoritma *supervised learning* yang fleksibel dan memiliki skalabilitas yang tinggi. XGBoost merupakan implementasi dari *gradient boosting machine* (*gbm*) dengan kinerja terbaik dalam *supervised learning* [20]. Algoritma ini dirancang untuk memaksimalkan sumber daya dan mengatasi keterbatasan *gradient boosting* sebelumnya. Perbedaan antara XGBoost dengan gradient boosting lain yaitu penggunaan teknik regularisasi baru untuk mengendalikan *overfitting*. Teknik ini membuat XGBoost lebih cepat dan stabil dalam proses *model tuning*[44].

XGBoost memiliki cara kerja sebagai berikut: Apabila memiliki sekumpulan data dengan ***m*** fitur dan ***n*** jumlah sampel $DS = (x_i, y_i) : i = 1 \cdots .n, x_i \in R^{m, y_i \in R}$. $\hat{y}_i$ menjadi hasil prediksi dari model pohon yang dihasilkan dari persamaan pada rumus berikut:

$$\text{Å}_i^i = \boldsymbol{\phi}(\mathbf{x_i}) = \sum_{k=1}^{K} \mathbf{f_k}(\mathbf{x_i}), \mathbf{f_k} \in \boldsymbol{\mathcal{F}} \quad (6)$$

K melambangkan jumlah dari pohon di model seperti pada rumus tersebut, $4, f_k$ merepresentasikan (pohon ke-k). Proses penyelesaian persamaan tersebut, maka perlu menemukan himpunan fungsi terbaik untuk meminimalkan *loss* dan *regularization* objective seperti pada rumus berikut.

$$\boldsymbol{\mathcal{L}}(\boldsymbol{\phi}) = \sum_{i} \mathbf{l}(\mathbf{y_i}, \text{Å}. \cdot \mathbf{i}) + \sum_{k} \boldsymbol{\Omega}(\mathbf{f_k}) \quad (7)$$

l melambangkan dari *loss function* dengan perbedaan antara hasil keluaran prediksi $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ dengan hasil keluaran sesungguhnya $\mathbf{y_i}$. Ketika **Ω** adalah ukuran kompleksitas model, hal ini membantu dalam menghindari *over-fitting* model dan dihitung dengan menggunakan rumus berikut:

$$\boldsymbol{\Omega}(\mathbf{f_k}) = \boldsymbol{\gamma}\mathbf{T} + \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\lambda} \parallel \mathbf{w} \parallel^2 \quad (8)$$

T pada rumus tersebut menggambarkan jumlah daun dari pohon dan ***w*** adalah bobot dari masing-masing daun. Pada *decision tree*, untuk meminimalkan fungsi tujuan, *boosting* digunakan di dalam proses pelatihan model yang berfungsi dengan menambahkan fungsi baru ***f*** saat model melatih model. Iterasi ke-***t*** ditambahkan fungsi baru (pohon) seperti rumus berikut.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \text{Å}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t)$$

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \quad (9)$$

$$g_i = \partial_{\tilde{A}^{(t-1)}} l\left(y_i, \text{Å} \cdot^{(t-1)}\right)$$

$$h_i = \partial^2_{\dot{A}^{(t-1)}} l\left(y_i, \dot{\text{Å}}^{(t-1)}\right)$$

XGBoost memiliki beberapa parameter yang bisa digunakan untuk mengoptimalkan model dalam proses *training*. Beberapa parameter tersebut diantaranya adalah sebagai berikut.

2.2.3.1 Early Stopping Rounds

Parameter early_stopping_rounds digunakan untuk menghentikan proses *training* pada saat tidak ada peningkatan nilai dalam proses *training* . Hal ini digunakan untuk menurunkan tingkat *overfitting* pada model. Proses *early stopping* dapat dilakukan dengan menambahkan parameter *evals* dengan data validasi untuk membandingkan model yang dibuat dengan hasil yang diharapkan [45].

2.2.3.2 Nthread

Parameter nthread digunakan untuk menjalankan XGBoost secara paralel. Proses ini akan mempercepat dalam proses *training* model XGBoost karena dilakukan bersamaan sesuai dengan jumlah *thread* yang ditentukan[46].

2.2.3.3 Eval Metric

Parameter eval_metric digunakan untuk memilih jenis *evaluation validation* pada algoritma XGBoost. Terdapat beberapa *evaluation metric* yang bisa digunakan pada XGBoost diantaranya adalah rmse, rmsle, mae, mape, logloss, error, merror, mlogloss, auc,aucpr, ndcg, map, poisson-nloglik, gamma-nloglik, cox-nloglik,

Penelitian ini menggunakan *evaluation metric* aucpr, error dan logloss [46].

2.3 Tools

2.3.1 Python

Python adalah sebuah bahasa pemrograman yang menggunakan interpreter dengan melakukan penerjemahan kode program ke bentuk biner secara *real-time*. Python memiliki banyak pustaka pendukung yang bisa digunakan dalam proses komputasi numerik. Python memiliki dukungan yang luas mulai dari pengembangan website, akses basis data, GUI desktop, perhitungan ilmiah dan pengembangan game. Python juga memiliki pustaka untuk melakukan penulisan kode secara interaktif dengan menggunakan Jupyter Notebook khususnya dalam pekerjaan konteks analisis data[47]. Alternatif lain yaitu dengan menggunakan Google Colab dengan layanan yang menyediakan ketersediaan GPU dan TPU dalam mempermudah proses pembelajaran mesin.

Gambar 2. 2 Logo Python

2.3.2 Google Colaboratory

Google Colaboratory memiliki fitur yang sama dengan Jupyter Notebook namun dengan beberapa fitur tambahan dengan komputasi awan yang dimiliki oleh Google. Google Colaboratory memungkinkan pengguna untuk menjalankan kode program secara daring dengan arsitektur yang dimiliki oleh Google. Selain itu, pengguna dapat melakukan eksekusi kode program dengan CPU, GPU maupun TPU sesuai dengan kebutuhan. Selain

itu, pengguna dapat melakukan *code sharing* kepada pengguna lain untuk melakukan kolaborasi.

Gambar 2. 3 Logo Google Colaboratory

2.3.3 Google Drive

Google Drive adalah penyimpanan berbasis *cloud* yang disediakan oleh Google[48]. Pengguna dapat menyimpan, mengubah maupun menghapus data dari Google Drive dari manapun selama terkoneksi dengan internet. Google Drive juga mendukung banyak perangkat seperti perangkat *mobile, tablet,* dan komputer. Google Drive memiliki tingkat keamanan yang tinggi sehingga berkas yang disimpan akan dienkripsi.

Gambar 2. 4 Logo Google Drive

2.3.4 Pentaho Data Integration

Pentaho Data Integration adalah sebuah aplikasi yang digunakan dalam proses integrasi data dan transformasi data. Transformasi data adalah sebuah aliran data dari sumber data, *transforming* data, dan memuat data pada lokasi target[49]. Penggunaan Pentaho Data Integration akan memudahkan dalam proses filter data, pemilihan kolom yang dibutuhkan

dan menghasilkan *output* dalam bentuk file maupun *query* ke dalam basis data.

Gambar 2. 5 Logo Pentaho Data Integration

2.4 Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
Advance Sustainable Science, Engineering and Technology[13]	*Automatic Complaints Categorization Using Random Forest and Gradient Boosting*	Muchamad Taufiq Anwar, Anggy Eka Pratiwi, Khadijah Febriana Rukhmanti Udhayana	Gradient Boosting dan Random Forest	Gradient Boosting memiliki tingkat akurasi lebih tinggi dibandingkan dengan Random Forest dengan akurasi sebesar 73%
Jurnal Nasional Teknologi dan Sistem Informasi [14]	Perbandingan Metode Decision Tree dan XGBoost untuk Klasifikasi Sentimen Vaksin Covid-19 di Twitter	Habib Hakim Sinaga, Surya Agustian	Decision Tree dan XGBoost	XGBoost keunggulan dalam proses klasifikasi sebesar 66% dan nilai *f1-score* sebesar 57%
IOP Conference Series:	*Fake news classification for Indonesian*	J P Haumahu, S D H Permana and	XGBoost	Model yang dibuat pada penelitian ini

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
Materials Science and Engineering [15]	*news using Extreme Gradient Boosting (XGBoost)*	Y Yaddarabullah		memiliki tingkat akurasi sebesar 92% dengan perbandingan *dataset* sebesar 80:20
Ultimatics : Jurnal Teknik Informatika [18].	Implementasi Algoritma *Complement* dan *Multinomial* Naïve Bayes *Classifier* Pada Klasifikasi Kategori Berita Media Online	Julio Christian Young, Alethea Suryadibrata, Hadian Mandala	*complement* dan *multinomial naïve bayes classifier*	Complement dan Multinomial Naïve Bayes dapat melakukan klasifikasi berita dengan baik dengan tingkat *f1-score* mencapai 95%
SN Applied Sciences [17]	*Prediction and diagnosis of future diabetes risk: a machine learning approach*	Roshan Birjais, Ashish Kumar Mourya, Ritu Chauhan & Harleen Kaur	Gradient Boosting, Logistic Regression, Naïve Bayes	Gradient Boosting memiliki akurasi tertinggi sebesar 85%, Naïve Bayes sebesar 77%, Logistic Regression sebesar 79%
International Journal of Intelligent Engineering & System[16]	*Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to*	Dimas Aryo Anggoro, Salsa Sasmita Mukti	XGBoost	Hasil dari penelitian yaitu XGBoost memiliki tingkat akurasi mencapai 99.29% dan *f-measure* sebesar 99%

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
	Predict Chronic Kidney Failure			

Tabel 2.1 menunjukkan penelitian terdahulu dalam melakukan klasifikasi teks dan penggunaan algoritma XGBoost serta Gradient Boosting. Penelitian ini menggunakan metode yang dilakukan pada penelitian sebelumnya namun dengan beberapa perbedaan diantaranya adalah penggunaan Grid Search serta *back translation*. Penelitian ini menggunakan judul berita dengan melakukan klasifikasi apakah judul tersebut termasuk *adverse media* atau bukan *adverse media*. *Feature extraction* yang dilakukan pada penelitian ini menggunakan TF-IDF seperti yang digunakan pada penelitian [18]. Selain TF-IDF, penelitian ini menggunakan *feature extraction* yang umum digunakan dalam mengolah data teks yaitu CountVectorizer dan Tokenizer. Penelitian ini menggunakan algoritma Gradient Boosting seperti pada penelitian [13], [17] yang memberikan hasil akurasi terbaik dibandingkan dengan algoritma pembandingnya yaitu Random Forest, Logistic Regression, dan Naïve Bayes. Penelitian ini juga menggunakan algoritma XGBoost seperti pada penelitian [14] yang memberikan hasil terbaik dibandingkan dengan algoritma Decision Tree. Penggunaan XGBoost juga berdasarkan pada penelitian [15] yang menghasilkan tingkat akurasi sebesar 92% dengan *parameter tuning*. Pada penelitian [16] penggunaan Grid Search sangat membantu dalam mendapatkan akurasi sebesar 99.29%. Kebaruan dari penelitian ini adalah melakukan klasifikasi *adverse media* atau bukan *adverse media* yang belum ada penelitian terkait hal tersebut. Penelitian ini akan menggunakan XGBoost dan Gradient Boosting serta Grid Search untuk mencari *hyperparameter* terbaik seperti pada penelitian [16]