

## BAB III

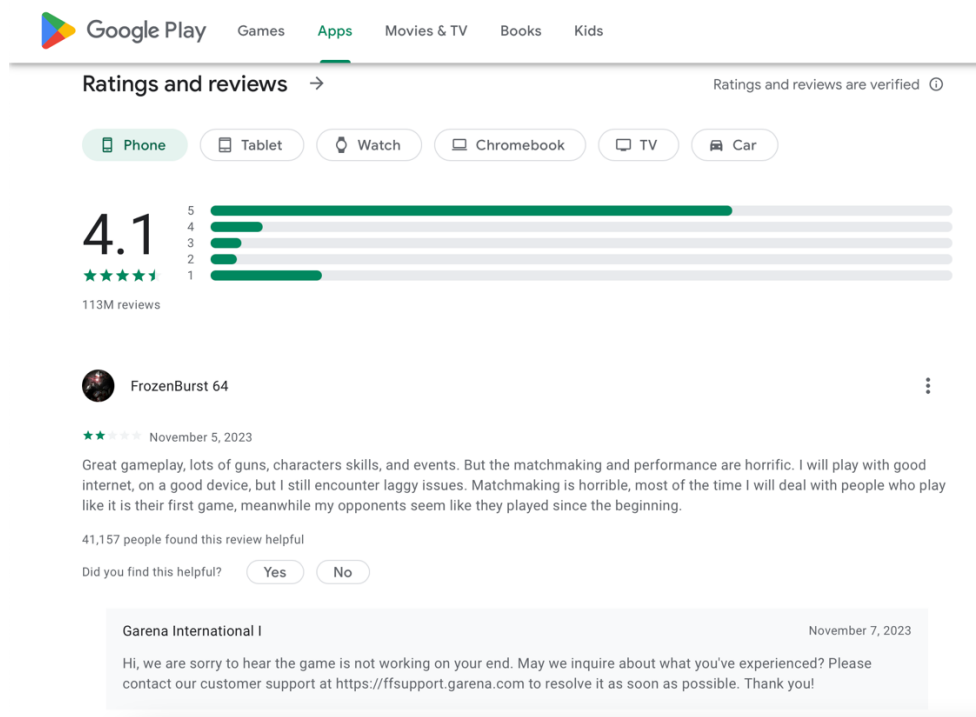
### METODOLOGI PENELITIAN

#### 3.1 Gambaran Umum Objek Penelitian

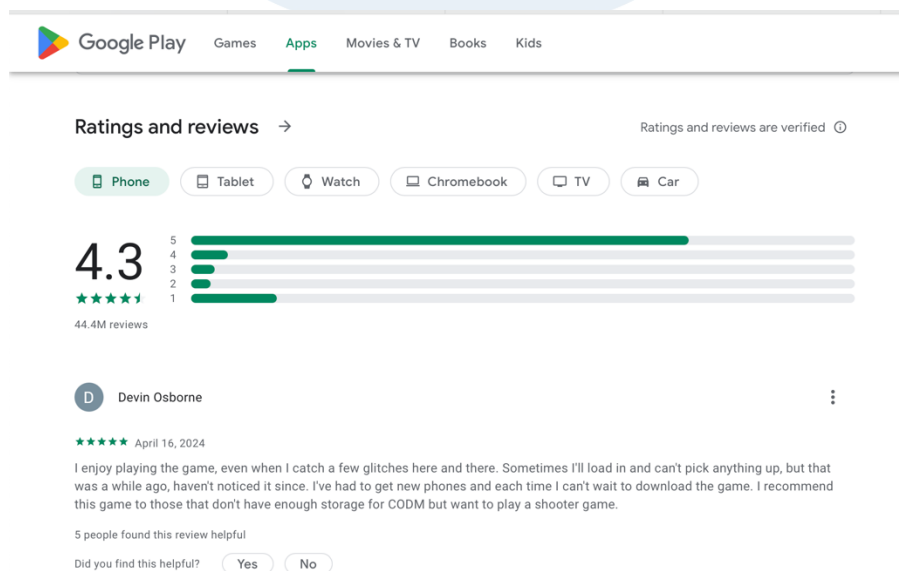
Gambaran umum objek penelitian ini berfokus kepada analisis sentimen terhadap *game* Free Fire dan juga PUBG Mobile. Penelitian dilakukan dengan menggunakan *K-Nearest Neighbor*. Penelitian ini dilakukan dengan tujuan memberikan hasil analisis sentimen yang memberikan pengetahuan mengenai sentimen pemain *game* tersebut. Melalui analisis ini juga dapat mengetahui perbedaan dari *game* Free Fire dan PUBG Mobile. Selain itu melalui penelitian dapat melihat perbedaan antara kedua *game* tersebut yang menyebabkan pemain hanya memainkan salah satu dari kedua *game* tersebut.

Data yang digunakan dalam analisis sentimen akan dikumpulkan terlebih dahulu dengan melakukan *crawling* ataupun menarik data komentar maupun ulasan yang diberikan pemain pada situs resmi ataupun forum terkait. Pada penelitian dilakukan pengumpulan data pada Google Playstore, dengan mengumpulkan ulasan dari para pengguna aplikasi. Data yang digunakan adalah data ulasan terbaru pada tahun 2024 dari bulan Maret sampai dengan bulan April.





Gambar 3. 1 Tampilan Halaman Review *Game* Free Fire pada Google Playstore



Gambar 3. 2 Tampilan Halaman Review *Game* PUBG Mobile pada Google Playstore

Pada gambar 3.1 adalah tampilan halaman *rating* dan *review game* Free Fire yang berisikan kumpulan dari ulasan yang diberikan oleh berbagai pemain Free Fire. Pada gambar 3.2 adalah tampilan halaman *rating* dan *review game* PUBG Mobile.

Pada penelitian, data atau informasi didapatkan melalui ulasan atau komentar yang diberikan oleh pemain kepada masing-masing *game*. Informasi dapat berisi pendapat maupun kepuasan dari pengguna maupun ungkapan kecewa dari penggunanya. Ulasan pengguna tersebut kemudian dikumpulkan, setelah itu data akan di proses menggunakan Google Colaboratory yang akan menghasilkan beberapa visualisasi data.

### 3.2 Metode Penelitian

#### 3.2.1 Metode Penyelesaian

Terdapat berbagai metode dalam melakukan *data mining*, diantaranya adalah metode KDD (*Knowledge Discovery in Databases*), SEMMA (*Sample, Explore, Model, Assess*), CRISP-DM. Metode tersebut memiliki tahapan yang berbeda. Proses maupun cara kerja dari setiap metode dapat memiliki kemiripan ataupun perbedaan. Berikut adalah berbagai tahapan dari ketiga metode tersebut:

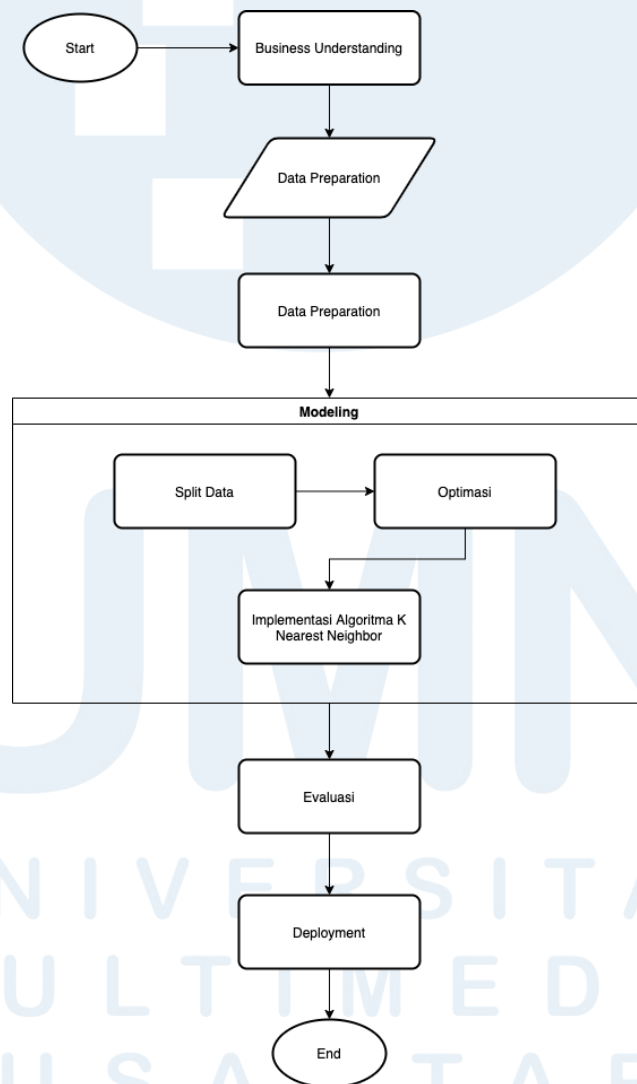
Tabel 3. 1 Tabel Perbandingan Tahapan Metode Data Mining

KDD	SEMMA	CRISP-DM
<i>Pre KDD</i>	-	<i>Business Understanding</i>
<i>Selection</i>	<i>Sample</i>	<i>Data Understanding</i>
<i>Pro Processing</i>	<i>Explore</i>	
<i>Transformation</i>	<i>Modify</i>	<i>Data Preparation</i>
<i>Data Mining</i>	<i>Model</i>	<i>Modeling</i>
<i>Interpretation / Evaluation</i>	<i>Assessment</i>	<i>Evaluation</i>
<i>Post KDD</i>	-	<i>Deployment</i>

Berdasarkan tabel 3.1 yang memberikan perbandingan tahapan pada berbagai metode *data mining*, terdapat beberapa perbedaan dari setiap metode. Metode KDD memiliki kemiripan pada tahapannya dengan CRISP-DM, namun pada metode CRISP-DM terdapat penggabungan tahapan *selection* dan *pro processing* dari KDD. Namun KDD lebih berfokus dalam penerapan *data mining* pada *database* yang membuat dalam tahapannya tidak bisa kembali pada tahap sebelumnya, sedangkan pada CRISP-DM memungkinkan melakukan tahapan sebelumnya kembali [36]. Melalui perbedaan tersebut menghasilkan penggunaan CRISP-DM yang lebih mudah dan dapat melakukan berbagai pembetulan pada tahapan sebelumnya tanpa harus menyelesaikan seluruh siklus tahapan.

Metode SEMMA memiliki tahapan yang lebih sedikit dibanding KDD dan CRISP-DM. Pada metode SEMMA hanya memiliki 5 tahap. Penggunaan metode SEMMA lebih spesifik digunakan pada *software* SAS. Metode CRISP-DM memiliki pendekatan *data mining* secara umum [37]. Berdasarkan perbedaan ketiga metode tersebut, penggunaan metode dapat disesuaikan dengan kebutuhan dari proyek ataupun penelitian yang akan dilakukan. Pada penelitian ini akan menggunakan metode CRISP-DM. Penelitian menggunakan metode CRISP-DM disebabkan tahapan yang terstruktur dan lebih fleksibel dengan dapat melakukan berbagai pembetulan pada tahap sebelumnya.

### 3.2.2 Alur Penelitian



Gambar 3. 3 Alur Penelitian

Pada penelitian akan menggunakan metode CRISP-DM, sehingga alur penelitian seperti pada gambar 3.3. Pada awal penelitian dilakukan *business understanding* terlebih dahulu. Penelitian dilakukan dengan melakukan pengumpulan data dengan melakukan *crawling* terlebih dahulu. Kemudian penelitian akan dilakukan dengan melakukan *data mining* berdasarkan data yang sudah dikumpulkan tersebut. Pada proses *data mining* akan menggunakan metode CRISP-DM yang memiliki beberapa tahapan. Setelah melalui proses *data mining*, maka akan dilakukan analisis terhadap hasil *data mining* yang dilakukan untuk mengetahui lebih jelas mengenai sentimen dari pemain Free Fire dan PUBG Mobile.

### 3.2.3 Metode *Data Mining*

Pada penelitian ini menggunakan *Cross Industry Standard Process - Data Mining (CRISP-DM)*, model ini menggambarkan siklus hidup *data mining* atau *data mining lifecycle*. Model ini membantu dalam mengolah proses penelitian menjadi lebih efisien dan efektif. *CRISP-DM* dibagi menjadi 6 tahap yaitu:

1. *Business understanding*
2. *Data understanding*
3. *Data preparation*
4. *Modeling*
5. *Evaluation*
6. *Deployment*

*Business Understanding*: Pada *game mobile* Free Fire dan PUBG Mobile saling bersaing bersamaan dengan *game* lainnya sebagai *game* dengan genre *battle royale*. Pada persaingan, dilakukan berbagai pengembangan yang bertujuan untuk membuat *game* lebih dinikmati oleh pemain, dan pemain memiliki tingkat kepuasan yang baik pada *game* tersebut. Oleh karena itu diperlukan pemahaman sentimen dari para pemain *game* genre *battle royale* tersebut.

Melalui analisis sentimen yang dilakukan, pembuat atau pengembang *game* dapat mengembangkan *game* sehingga *game* menjadi lebih unggul

dibanding pesaing lainnya. Selain itu dengan melakukan analisis sentimen akan memberikan informasi lebih terkait berbagai perbedaan pada *game* dengan *game* genre yang sama lainnya. Melalui hal tersebut, pengembang dapat mengembangkan *game* yang unik dan dapat menarik perhatian para pemain.

*Data Understanding*: Pada tahap ini, dilakukan pengumpulan data dan pemahaman atas data serta mencari berbagai faktor yang dapat mempengaruhi penelitian sentimen. Data yang digunakan berasal dari hasil *crawling* atau menarik data komentar, opini atau ulasan dari pemain dalam bentuk teks yang diberikan pada berbagai situs, ataupun forum resmi terkait *game* tersebut. Proses pengambilan data digunakan dengan menggunakan *library google\_play\_scraper* yang tersedia pada Google Colaboratory untuk melakukan *scraping* pada Google Playstore. Data yang sudah terkumpul kemudian dikumpulkan dan dirancang menjadi sebuah *dataset*. Pada pembuatan *dataset*, hanya memasukkan beberapa informasi yang relevan dengan penelitian seperti 'userName', 'score', 'at', dan 'content'. *Dataset* mengandung informasi berisi ekspresi, maupun perasaan pemain ketika memainkan *game* Free Fire ataupun PUBG Mobile.

*Data Preparation*: Pada tahapan ini, data yang dikumpulkan akan dipersiapkan sehingga sesuai dengan pemodelan. Data pertama-tama dapat dibersihkan terlebih dahulu, seperti data komentar atau ulasan yang diberikan tidak memiliki keterkaitan dengan *game* tersebut. Tidak hanya itu juga dilakukan pembersihan berbagai karakter pada kalimat ulasan, terutama karakter atau huruf yang berulang lebih dari 2 kali.

Proses akan dilakukan dengan urutan *cleansing*, *case folding*, *tokenization*, filter *stop words*, *stemming*, proses ini disesuaikan dengan penelitian terlebih dahulu yang menjadi acuan penelitian yaitu penelitian [15], [13], dan [14]. Proses *cleansing* adalah proses yang menghilangkan duplikat data, ataupun data yang tidak memiliki informasi yang dibutuhkan pada teks. Proses *case folding* adalah proses yang merubah kalimat menjadi huruf kapital ataupun huruf kecil. Tokenisasi adalah proses pemecahan kalimat yang dibagi menjadi beberapa bagian atau kata yang disebut token. Filter *stopwords* adalah penghapusan atau

proses eliminasi kata yang tidak dibutuhkan. *Stemming* adalah proses penghilangan imbuhan pada awal maupun akhir kata menjadi kata dasar [15].

Pembersihan atau *cleansing* dilakukan dengan melakukan proses *text-processing* dengan menggunakan *tools* NLTK (*Natural Language Tools Kit*). Proses ini menggunakan berbagai *library* seperti *re*, *string*, *emoji*, dan *BeautifulSoup*. Proses ini dilakukan dengan pembersihan kalimat dengan menghilangkan berbagai karakter spesial, *url*, *emote*, tanda baca, ataupun simbol maupun kalimat yang tidak memiliki nilai atau informasi. Dilanjutkan dengan proses pencarian huruf lebih dari 2 kali, dan digantikan menjadi 1 huruf, dengan tujuan memudahkan dan memastikan translasi yang dihasilkan nantinya akan sesuai dengan informasi yang disampaikan pada *review*. Proses *case folding* kemudian dilakukan dengan melakukan atau mengubah seluruh kalimat menjadi kalimat dengan huruf kecil tanpa huruf kapital. Proses selanjutnya adalah proses tokenisasi untuk memisahkan kata dari berbagai kalimat sehingga menjadi beberapa kata yang dapat identifikasi lebih lanjut. Proses ini dilakukan sehingga memudahkan data untuk diolah pada proses selanjutnya [38].

Teks yang sudah ditokenisasi kemudian dapat diolah dengan menghilangkan *stop words* atau kata umum ataupun kata penghubung. Kata tersebut dihapus dengan tujuan pembersihan teks dari berbagai kata yang tidak memiliki informasi ataupun tidak relevan dengan penelitian [38]. Proses filter ini menggunakan *tools* NLTK, pada *library stopwords*. Proses selanjutnya adalah proses *stemming*, pada proses ini berbagai kata dikembalikan ke bentuk kata dasarnya, berbagai kata yang memiliki imbuhan akan dikembalikan menjadi sebuah kata dasar [38]. Proses *stemming* tersebut memanfaatkan *library* Sastrawi. Data yang sudah melewati proses *stemming* kemudian diolah dengan melakukan translasi dari bahasa Indonesia menjadi bahasa Inggris menggunakan *deepL* pada *library* GoogleTranslator. Keperluan translasi ini dilakukan disebabkan terbatasnya *tools* dalam melakukan *automatic labeling* pada Bahasa Indonesia.

Data yang sudah dibersihkan kemudian diolah dengan melakukan proses *automatic labeling* data. Pada proses *labeling*, dilakukan proses penyesuaian data sehingga dapat dilakukan *labeling* sesuai dengan metode yang digunakan.

Proses *labeling* dapat menggunakan berbagai metode salah satunya adalah metode pendekatan *Lexicon*. Pada penelitian menggunakan pendekatan *Lexicon* yaitu *Vader Lexicon*, pendekatan ini memberikan *labeling* secara otomatis berdasarkan nilai *polarity* sentimen [19]. Nilai *polarity* sentimen dibagi menjadi beberapa label seperti positif, negatif, maupun netral. Kondisi untuk nilai *polarity* positif adalah lebih besar dari 0. Label negatif untuk nilai *polarity* lebih kecil dari 0, dan sisanya label netral [39]. Data yang sudah diberikan *labeling* kemudian digunakan untuk berbagai visualisasi sehingga mencapai tujuan penelitian. Visualisasi yang diberikan berupa visualisasi distribusi sentimen, serta kumpulan kata awan pada setiap kelas label dengan menggunakan *library matplotlib*. Visualisasi komentar atau ulasan tersebut memiliki pemahaman kepada *gamer* sehingga dapat memutuskan *game* yang cocok untuk dimainkan, selain itu analisis sentimen juga dapat membantu pemain sehingga dapat menyesuaikan ekspektasi terhadap suatu *game* [40]. Pengembang *game* juga dapat diberikan pemahaman dan wawasan atas pengembangan yang dilakukan terhadap *game* tersebut. Melalui pemahaman tersebut pengembang *game* dapat membuat keputusan pengembangan maupun desain *game* yang lebih baik untuk masa depan [40].

*Modeling*: Pada tahap ini akan dilakukan pembuatan model yang sesuai dengan data yang sudah dipersiapkan. Pada proses pemodelan ini akan menggunakan *library sklearn*. Data yang sudah disiapkan diproses dan diubah menjadi bentuk vektor melalui proses vektorisasi TF-IDF (*Term Frequency-Inverse Document Frequency*). Proses ini digunakan untuk memberikan bobot pada sebuah kata dengan menentukan nilai frekuensi dari kata tersebut [15]. Proses ini akan mengubah data menjadi vektor, dimana setiap fiturnya merepresentasikan sebuah nilai dan kepentingannya. Penggunaan metode TF-IDF ini disesuaikan dengan penelitian sebelumnya yang menjadi acuan yaitu penelitian [15] dan [14].

Proses yang dilakukan selanjutnya adalah proses pengurangan dimensi menggunakan *TruncatedSVD (Truncated Singular Value Decomposition)*. Data yang dihasilkan proses vektorisasi berbentuk TF-IDF matriks dan umumnya berukuran besar. Proses ini akan mengurangi dimensi matriks dan



mempertahankan fitur yang penting pada data [41]. Data yang sudah diproses kemudian akan dibagi menjadi 2 bagian yaitu data *training* dan *testing*. Pada penelitian sebelumnya [13] menggunakan 1900 komen sebagai data *training* dan 100 komen sebagai data *testing*, berdasarkan jumlah tersebut data dibagi menjadi 95% data *training* dan 5% data *testing*. Penelitian akan dilakukan dengan pembagian data yang berbeda dari penelitian sebelumnya dengan pembagian 80% data *training* dan 20% data *testing*. Pembagian dilakukan dengan tujuan mengurangi kemungkinan pembelajaran yang dilakukan menggunakan *machine learning* menjadi bias terhadap data *training* [42].

Data yang sudah dibagi menjadi 2 bagian kemudian dilakukan pengecekan jumlah pada setiap labelnya. Data yang tidak seimbang pada setiap label atau kelas akan mengakibatkan kesalahan klasifikasi pada kelas minor [43]. Oleh karena itu ketika ditemukan data yang tidak seimbang akan diproses menggunakan teknik *oversampling*. Proses *oversampling* ini akan dilakukan dengan menggunakan *library* imblearn.

Algoritma yang akan digunakan untuk melakukan analisis sentimen adalah *K-Nearest Neighbor*. Pembuatan model dilakukan dengan melakukan berbagai optimasi. Optimasi dapat dilakukan dengan berbagai cara, salah satunya adalah mencari nilai K terbaik yang akan digunakan pada algoritma KNN. Pada proses pembuatan model untuk menguji optimasi lainnya yang akan dilakukan, dilakukan pemodelan dengan algoritma *K-Nearest Neighbor* dengan *hyperparameter tuning* nilai K yang nantinya akan digunakan sebagai tolak ukur hasil optimasi.

Optimasi dilakukan dengan berbagai cara salah satunya adalah optimasi dengan *feature scaling*. Algoritma KNN melakukan klasifikasi berdasarkan jarak, oleh karena itu dengan menggunakan *feature scaling* dapat membantu meningkatkan performa algoritma. Fitur dengan magnitudo yang lebih besar dapat mempengaruhi jarak secara tidak proporsional, menyebabkan hasil yang bias. Pada optimasi *feature scaling* ini tidak semua data sesuai dan cocok menggunakan *feature scaling* [44]. *Feature scaling* dilakukan dengan menggunakan *package* preprocessing pada *library* sklearn.

Optimasi lainnya yang dilakukan adalah optimasi dengan melakukan seleksi fitur, menyesuaikan dengan penelitian sebelumnya yang menjadi acuan yaitu penelitian [16] dan penelitian [17] yang menggunakan seleksi fitur dengan teknik RFE. Teknik RFE ini dapat dilakukan dengan menggunakan berbagai metode lainnya, namun diantaranya seleksi fitur menggunakan *Random Forest Classifier* memiliki hasil yang terbaik dan logistik regression menjadi yang terbaik kedua [45]. Penggunaan teknik RFE ini akan digabungkan dengan *density feature*. Penggunaan kedua teknik ini bertujuan untuk mengatasi masalah yang dialami pada penelitian [15], fitur kepadatan ini dapat dikalkulasi menggunakan *NearestNeighbor* model sehingga model dapat memiliki informasi tambahan terkait kepadatan data. Informasi tersebut dapat menjadi suatu hal penting terutama ketika data distribusi yang tidak merata [46].

Pada pemodelan juga dapat dilakukan optimasi *hyperparameter tuning* pada algoritma KNN. Optimasi ini menggunakan *package* model\_selection pada *library* sklearn. *Hyperparameter tuning* ini berguna untuk mencari parameter terbaik yang dapat digunakan pada model, parameter yang dapat dicari seperti nilai K, metrik, maupun *leaf size*. Pemodelan akan menghasilkan tingkat akurasi yang kemudian akan ditentukan algoritma KNN dengan model yang memiliki akurasi paling tinggi. Proses yang dilakukan akan dilakukan ke 2 *dataset* yaitu *dataset game* Free Fire dan *game* PUBG Mobile.

*Evaluation*: Pada tahap ini akan dilakukan evaluasi terhadap model yang sudah dibuat. Evaluasi dilakukan dengan memperhatikan tingkat akurasi maupun kesalahan dari algoritma yang dibentuk. Melalui tingkat akurasi yang tinggi serta tingkat kesalahan yang kecil tentunya akan menghasilkan analisis sentimen yang tepat. Hasil analisis sentimen tersebut yang nantinya akan dibandingkan dengan penelitian terdahulu, serta menjawab rumusan masalah maupun permasalahan dan mencapai tujuan yang sudah ditentukan.

*Deployment*: Pada tahap ini penelitian akan dipublikasikan sehingga dapat menjadi referensi untuk para pengembang *game* maupun masyarakat. Publikasi dilakukan melalui pembuatan laporan, serta pembuatan *prototype website* yang dapat membantu melakukan analisis sentimen. Pada *prototype* yang dikembangkan akan memiliki fungsi sehingga orang yang mengakses *website*

tersebut dapat memilih dan memasukkan *id* sebuah *game* dari Google Playstore. Pengguna kemudian dapat melakukan proses analisis sentimen melalui tombol yang disediakan. Melalui tahapan ini akan mencapai manfaat yang diinginkan sehingga para pengembang *game* dapat memiliki pemahaman dan pengetahuan lebih jelas atas sentimen para pemain, sehingga pengembang *game* dapat terus mengembangkan *game* yang dimiliki. Selain itu masyarakat juga dapat terbantu dengan menjadikan penelitian sebagai referensi dalam memilih untuk bermain *game* Free Fire ataupun PUBG Mobile.

Tabel 3. 2 Tabel Perbandingan Algoritma

Algoritma	Kelebihan	Limitasi
<i>K-Nearest Neighbor</i>	<ol style="list-style-type: none"> <li>1. Tidak membutuhkan asumsi parameter</li> <li>2. Dapat mempertimbangkan lebih banyak karakteristik dari teks</li> </ol>	<ol style="list-style-type: none"> <li>1. Membutuhkan waktu dalam menentukan nilai optimal K</li> <li>2. Kendala dalam melakukan pencarian <i>Neighbor</i> terdekat</li> </ol>
<i>Naïve Bayes Classifier</i>	<ol style="list-style-type: none"> <li>1. Mudah untuk digunakan</li> <li>2. Daya komputasi yang digunakan optimal</li> </ol>	<ol style="list-style-type: none"> <li>1. Membutuhkan data dalam jumlah besar untuk menghasilkan performa baik</li> <li>2. Memiliki asumsi yang besar tentang bentuk distribusi data</li> </ol>
<i>Support Vector Machine</i>	<ol style="list-style-type: none"> <li>1. Dapat memodelkan berbagai batasan <i>non-linear decision</i></li> <li>2. Memiliki tingkat ketahanan terhadap <i>overfitting</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Kompleksitas tinggi dan membutuhkan memori besar dalam klasifikasi</li> <li>2. Kurangnya transparansi dalam hasil yang disebabkan jumlah dimensi yang tinggi khususnya data teks</li> </ol>

Berdasarkan tabel perbandingan 3.2 menjelaskan berbagai kekurangan dan kelebihan dari berbagai algoritma. Algoritma KNN membutuhkan daya komputasi dan waktu yang besar, namun menghasilkan hasil dan performa yang baik [47]. Algoritma *Naïve Bayes Classifier* adalah salah satu algoritma yang mudah untuk diimplementasi dan digunakan, namun terdapat berbagai kekurangan diantaranya model ini adalah model yang simpel, memiliki asumsi besar dan membutuhkan data dalam jumlah besar [48]. Algoritma SVM memiliki kemampuan yang bagus dalam melakukan klasifikasi dan memiliki akurasi tinggi, namun memiliki kekurangan

yang membutuhkan memori besar serta memiliki limitasi waktu dan besar pada fase *training* dan *testing* [49].

Berdasarkan kekurangan maupun kelebihan dari berbagai algoritma, pada penelitian ini menggunakan algoritma KNN yang memiliki keunggulan dengan dapat berfungsi secara efektif dalam mengolah data berbentuk teks. Selain itu pada penelitian tidak memiliki batasan masalah terkait waktu, sehingga dalam penelitian dapat dilakukan dan mencari berbagai nilai yang dibutuhkan pada algoritma KNN secara optimal. Melalui optimasi yang dilakukan pada penelitian dapat mengatasi berbagai kelemahan atau limitasi yang dimiliki pada algoritma KNN menyebabkan algoritma KNN akan menghasilkan performa yang lebih unggul dibandingkan algoritma lainnya. Oleh karena itu penelitian berfokus dalam melakukan analisis data menggunakan algoritma KNN.

### 3.3 Teknik Pengumpulan Data

Pengumpulan data adalah salah satu tahap penting pada penelitian, dikarenakan hasil penelitian akan terpengaruh oleh data yang digunakan. Melalui teknik pengumpulan data yang tepat mampu memberikan penelitian informasi yang tepat dan akurat serta relevan untuk penelitian. Pada penelitian ini menggunakan teknik pengumpulan data primer dengan mengumpulkan seluruh komentar, maupun ulasan dari pemain atas *game* Free Fire dan PUBG Mobile pada Google Playstore. Periode data yang dikumpulkan adalah data ulasan terbaru pada tahun 2024 dari bulan Maret sampai dengan bulan April. Pengumpulan data dilakukan pada beberapa situs resmi maupun forum aktif atas kedua *game* tersebut, dengan tujuan mendapatkan data asli yang berkualitas serta berisi sentimen dari pemain ketika memainkan kedua *game* tersebut.

### 3.4 Teknik Analisis Data

Tabel 3. 3 Tabel Perbandingan Bahasa Pemrograman

Bahasa Pemrograman	Kelebihan	Kekurangan
R	Beragam analisis analitik, serta berbagai ragam visualisasi data.	Kompleks dan memiliki banyak <i>command</i> sehingga sulit dipelajari.
Python	Lebih mudah dipelajari, mudah diaplikasikan, dan memiliki fungsi yang fleksibel	Memiliki keterbatasan desain, pekerjaan yang dilakukan

		pada Python menggunakan banyak memori.
--	--	--

Pada tabel 3.3 memberikan penjelasan mengenai perbandingan antara bahasa pemrograman R dan Python. Melalui perbandingan tersebut, penelitian memutuskan menggunakan Python disebabkan fungsi fleksibel dari Python. Python dapat digunakan dalam menyelesaikan penelitian dengan melakukan pengolahan data, melakukan *machine learning*, dan juga pembuatan *prototype*. Bahasa pemrograman Python memiliki berbagai kekurangan, namun kelebihan yang dimiliki lebih penting dan sangat berguna pada penelitian.

Tabel 3. 4 Tabel Perbandingan *Tools* Jupyter Notebook dan Google Colab

<b>Tools</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
Jupyter Notebook	Mempunyai <i>interface</i> yang mudah digunakan dan dimengerti oleh pengguna. Penulisan kode dapat terbagi dalam bentuk <i>block</i> , serta memiliki fitur <i>markdown</i> maupun <i>heading</i> yang membantu pengguna untuk menjelaskan kode. Selain itu memiliki <i>library</i> maupun <i>packages</i> yang cukup lengkap.	Perlu menginstall <i>software</i> jupyter notebook. Diperlukan <i>import</i> serta <i>install library</i> maupun <i>packages</i> ketika ingin digunakan, bahasa pemrograman terbatas.
Google Colab	Berbasis Jupyter Notebook, sehingga memiliki <i>interface</i> yang juga mudah dipahami dan digunakan. Kode yang ditulis juga dalam berbentuk <i>block</i> , namun kode tersebut dapat diakses secara <i>online</i> dikarenakan kode akan disimpan di <i>cloud</i> berbeda dari Jupyter Notebook yang menyimpan <i>file</i> di perangkat anda. Terdapat fitur <i>share</i> maupun <i>collaboration</i> .	Dapat mengakses Google Colab secara <i>online</i> namun perlu menginstall dan <i>import library</i> ataupun <i>packages</i> ketika ingin menjalankan kode. Membutuhkan komputasi yang lebih memberatkan <i>GPU</i> ketimbang Jupyter Notebook.

*Tools* yang digunakan pada penelitian ini adalah Google Colab yang mempunyai *user interface* yang baik serta memiliki *block* dalam menulis kode. Fitur ini memudahkan pengguna ketika mendeteksi *error*. Berdasarkan tabel 3.4 Google Colab juga menawarkan kemudahan dan tidak adanya limitasi ukuran data yang digunakan untuk melakukan *testing* maupun *training*. Fitur *collaboration* dan *share* memudahkan penulis dalam mencapai manfaat yang diinginkan, dengan kode yang dapat disebar dan dijadikan pembelajaran. Melalui fitur tersebut penelitian

ini dapat dijadikan referensi dalam melakukan sentimen analisis atas *game* Free Fire dan PUBG Mobile.

Tabel 3. 5 Library yang Digunakan

<b>Library</b>	<b>Fungsi</b>
google-play-scraper	Untuk melakukan <i>scraping</i> data pada aplikasi Google Playstore
pandas	Untuk mengolah <i>dataframe</i> , seperti membaca <i>file</i> ataupun membuat <i>file</i> atau <i>dataframe</i>
numpy	Untuk mengolah dan memanipulasi data
googletrans	Untuk melakukan translasi data dari suatu bahasa ke bahasa lainnya, pada penelitian digunakan untuk translasi Bahasa Indonesia ke Bahasa Inggris
nltk	Untuk melakukan pemrosesan bahasa alami
BeautifulSoup	Untuk melakukan menghilangkan teks yang memiliki elemen html
re	Untuk memanipulasi teks dengan pola tertentu
Sastrawi	Untuk melakukan <i>stemming</i> pada teks dengan Bahasa Indonesia
vaderSentiment	Untuk melakukan <i>automatic labeling</i>
matplotlib	Untuk memberikan visualisasi data
sklearn	Untuk mengakses berbagai algoritma <i>machine learning</i> maupun alat dalam proses <i>data science</i>
imblearn	Untuk implementasi teknik <i>oversampling</i>
seaborn	Untuk visualisasi data dengan grafik statistik informatif

Pada tabel 3.5 menampilkan seluruh *library* yang digunakan pada penelitian. *Library* yang digunakan mencakup untuk seluruh proses yang akan dijalankan sesuai dengan alur penelitian. Setiap *library* memiliki fungsinya masing-masing dan tentunya dibutuhkan selama proses penelitian.

