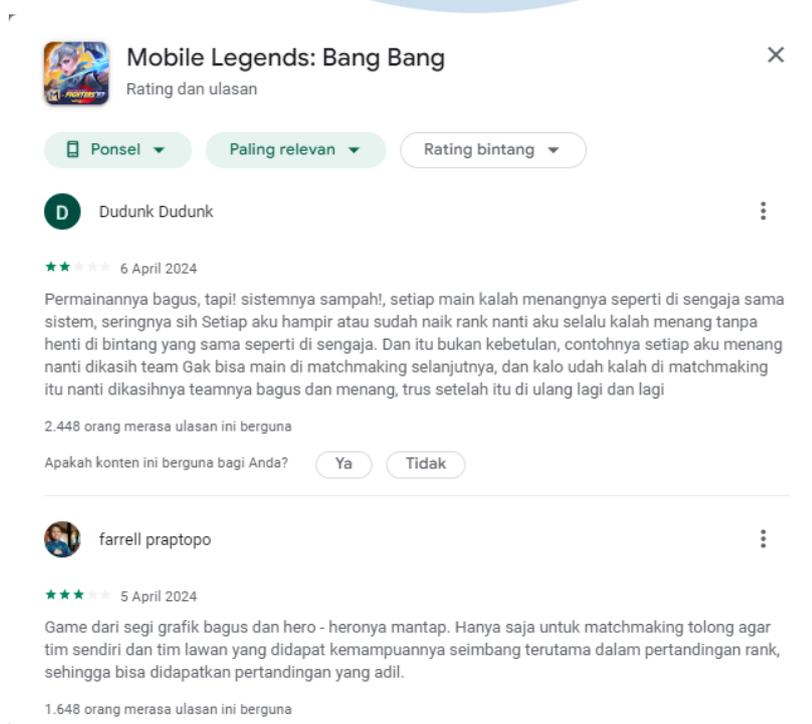


BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Fokus penelitian ini ialah melakukan sebuah analisis sentimen dari para pemain *game online* Mobile Legend: Bang Bang dan League of Legends dengan menggunakan algoritma Random Forest dan Decision Tree. Penelitian analisis sentimen ini didasari oleh tujuan dalam memahami bagaimana pemain dalam merespon dan merasakan pengalaman pada saat bermain untuk kedua *game* tersebut berdasarkan ulasan komentar yang dituliskan oleh pemain sehingga informasi tersebut dapat memberikan sebuah preferensi, kepuasan, kritikan ataupun sebuah potensi perbaikan kepada pengembang aplikasi *game online* Mobile Legend Bang Bang dan League of Legends. Data yang akan digunakan dalam proses sentimen analisis ini akan didapatkan dengan menggunakan metode *scraping* pada aplikasi Google Playstore.



Gambar 3. 1 Komentar-Komentar pada Aplikasi Google Playstore

Gambar 3.1 menampilkan ulasan atau opini dari para pengguna aplikasi yang telah mengunduh dan mencobanya. Informasi yang akan didapatkan dengan metode *scraping* tidak hanya komentar pengguna saja namun juga *username*, tanggal, dan *rating*. Hasil pengumpulan data tersebut akan menggunakan bahasa Python dan *dataset* yang terbentuk akan dalam format *.csv*.

3.2 Metode Penelitian

3.2.1 Metode Penyelesaian

Penelitian ini akan menggunakan metode CRISP-DM sebagai alur penelitian. Penggunaan metode ini dikarenakan lebih terstruktur pada tahapan dan kerangka kerjanya sehingga membuat peneliti lebih terarah dan mengetahui tahapan apa yang perlu dikerjakan dalam penelitian ini [45]. Berikut merupakan perbandingan antara CRISP-DM dengan metode *data mining* lainnya:

Tabel 3. 1 Perbandingan Metode *Data Mining*

Proses	CRISP-DM	KDD	SEMMA
	Bussines Understanding	Mengembangkan dan memahami aplikasi	-
	Data Understanding	Membuat target pada dataset	Sample
		Data akan di cleansing sample dan melakukan preprocessing	
	Data Preparation	Transformasi data	Explore
	Modeling	Memilih task data mining paling cocok	Modify
		Memilih algoritma data mining yang cocok	Model
		Mengimplementasikan algoritma data mining	
	Evaluation	Menafsirkan pola mining	Assesment
	Deployment	Menggunakan discovered knowledge	-
Total proses	6	9	5

Pada Tabel 3.1 menunjukan bahwa total proses tahapan metode *Knowledge Discovery in Databases* (KDD) menjadi metode dengan tahapan terbanyak dan terkompleks. Namun, banyak tahapan yang dapat dijadikan satu tahapan sehingga tidak membuang waktu. SEMMA menjadi metode dengan tahapan terdikit dibandingkan kedua metode lainnya meskipun hanya berbeda satu tahap

dari metode CRISP-DM tidak membuat metode ini menjadi metode terbaik karena masih ada hal yang perlu diproses. Metode CRISP-DM menjadi metode yang paling seimbang dibanding kedua metode lainnya. Tahapan pada metode ini lebih pendek ketimbang metode KDD namun tidak terlalu pendek seperti metode SEMMA. Pada penelitian ini akan menggunakan metode CRIPS-DM dikarenakan penelitian ini terdapat hasil *deployment* berupa situs yang dapat melakukan analisis sentimen untuk aplikasi yang ada di *platform* Google Playstore selain itu keseimbangan alur proses pada metode CRIPS-DM juga menjadi dasar pemilihan metode ini ketimbang metode lainnya seperti KDD atau SEMMA.

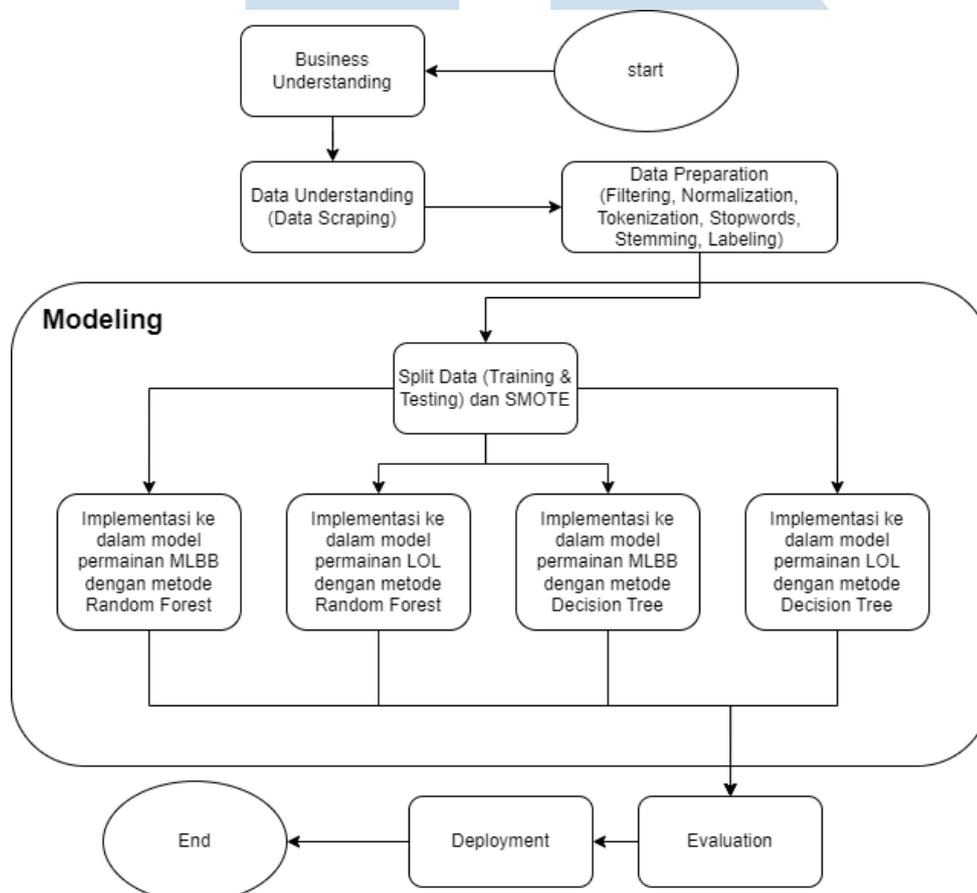
Tabel 3. 2 Perbandingan Algoritma

Algoritma	Kelebihan	Kekurangan
Random Forest	Memiliki kemampuan terbaik dalam mengolah data <i>outlier</i> dan data <i>non-linear</i> , kemungkinan terjadinya <i>overfitting</i> rendah, dan lebih efisien pada data yang besar.	Cenderung bias pada variabel yang kategorikal, waktu yang diperlukan dalam mengolah data besar cukup lambat, dan kurang cocok pada metode <i>linear</i> yang punya fitur <i>sparse</i> .
Decision Tree	Mudah untuk dipakai dan dipahami, berguna dalam memecah masalah keputusan, memberikan semua kemungkinan hasil pada suatu masalah, dan mempunyai data <i>cleaning</i> yang lebih kecil.	<i>Layer</i> yang ditampilkan cukup banyak sehingga rumit, memungkinkan terjadinya <i>overfitting</i> , label yang banyak dapat membuat komputasi meningkat.

Pada tabel 3.2 menunjukkan perbandingan antara algoritma yang akan digunakan pada penelitian ini yaitu, Random Forest dan Decision Tree. Jika dilihat pada perbandingan diatas algoritma Random Forest mempunyai kelebihan dalam mengolah data yang jumlah besar dan algoritma Decision Tree mempunyai kelebihan penimplementasian yang cukup mudah. Pemilihan algoritma ini didasari oleh keunggulan pada kedua algoritma tersebut dan pemilihan algoritma ini juga didasari oleh penelitian terdahulu [10], [11], [12], [13], dan [14] serta juga merupakan saran dari penelitian terdahulu [16]. Random Forest juga merupakan gabungan dari berbagai pohon yang ada di algoritma Decision Tree sehingga diharapkan bahwa kinerja model Random Forest dapat lebih baik dibandingkan model Decision Tree [41]. Apabila mengacu pada jurnal terdahulu [14], model

Decision Tree mempunyai kinerja lebih baik dibandingkan model Random Forest, oleh karena itu pada penelitian ini akan menguji performa mengenai kedua algoritma tersebut dengan memadukan teknik *oversampling*.

3.2.2 Alur Penelitian



Gambar 3. 2 Alur Penelitian

Penelitian ini akan menggunakan CRISP-DM akan menjadi alur pada penelitian sentimen analisis ini, seperti pada gambar 3.2. Penggunaan metode CRISP-DM pada penelitian ini akan menggambarkan mengenai langkah yang sistematis dalam melakukan analisis sentimen kepada para pemain *game online* MLBB dan LOL dengan fokus kepada tingkat kepuasan dan preferensi para pemain. Metodologi CRISP-DM mempunyai 6 tahapan dalam siklus prosesnya, yakni *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*. Seluruh tahapan siklus proses ini akan

menjadi dasar dalam alur penelitian dan akan disesuaikan oleh kebutuhan penelitian.

3.2.3 Metode Data Mining

Berdasarkan alur penelitian dengan metode CRISP-DM dengan 6 tahapan siklus, berikut merupakan proses dari CRISP-DM yang dapat diterapkan ke dalam bentuk penelitian ini, sebagai berikut:

1. *Business Understanding*

Melakukan analisis sentimen kepada para pemain atau pengguna kedua aplikasi *game online* tersebut, yakni League of Legends dan Mobile Legends Bang-Bang bertujuan untuk mencari tahu kelebihan dan kekurangan yang ada sehingga diharapkan dapat bermanfaat bagi pengembang aplikasi untuk perkembangan *game* tersebut kedepannya.

2. *Data Understanding*

Sumber data akan berasal dari ulasan-ulasan dan komentar para pemain di halaman Google Playstore dari kedua *game online* tersebut. Menganalisis pola struktur dan konten-konten dari ulasan pemain, menentukan variabel sentimen, dan juga mengidentifikasi pola umum atau tren. Penelitian ini akan menggunakan teknik *scraping* dalam mengumpulkan ulasan atau komentar para pemain dari halaman Google Playstore. Variabel yang akan diambil melibatkan “Username”, “Rating”, “Date”, dan “Comment”. Proses *scraping* akan dilakukan dengan bahasa Python menggunakan *library* yang telah disediakan bernama Google-play-scraper di aplikasi Google Colaboratory. Data yang dikumpulkan akan menggunakan bahasa Indonesia dengan jumlah sebanyak 1000 data untuk *game* Mobile Legends Bang-Bang dan 1000 data lagi untuk *game* League of Legends.

3. *Data Preparation*

Proses selanjutnya ialah dengan membersihkan data atau *filtering* dari beberapa karakter khusus, tautan, emotikon, data duplikat, dan elemen

noise lainnya. Selain itu pada tahap ini juga akan melakukan proses normalisasi teks bertujuan untuk membuat semua teks menjadi huruf kecil. Selanjutnya melakukan proses *stopwords* menggunakan *library* NLTK.corpus yang bertujuan untuk mengurangi kata-kata yang tidak mempunyai arti khusus [10]. Proses selanjutnya ialah tokenisasi pada data teks yang berfungsi untuk memisah-misahkan teks menjadi per kata. Proses terakhir ialah *stemming* dengan menggunakan *library* Sastrawi yang berguna dalam mengubah kata yang berimbuhan menjadi kata dasar. Selanjutnya ialah proses *labeling* yang bertujuan untuk menilai data tersebut merupakan kategori positif, negatif, atau netral. Namun, sebelum masuk ke dalam proses *labeling* diperlukannya proses penerjemahan teks dikarenakan data yang digunakan merupakan bahasa Indonesia. Penerjemahan teks akan menggunakan *library* GoogleTranslator. Proses *labeling* ini akan menggunakan metode yaitu dengan metode *labeling* NLTK Vader Lexicon. Penggunaan teknik *labeling* NLTK Vader Lexicon dikarenakan kemampuannya dalam menganalisis data yang berbahasa Inggris dengan mempunyai 7.500 kata baik sinonim ataupun akronim. Metode *labeling* ini akan mendeteksi sentimen berdasarkan *compound scores* dengan ketentuan apabila *score* < 0 merupakan sentimen negatif, *score* > 0 merupakan sentimen positif, dan *score* $= 0$ merupakan sentimen netral [46]. Penentuan sentimen pada metode NLTK ini akan ditentukan berdasarkan *compound score* yang dimana setiap teks akan dipecah menjadi sebuah token kemudian akan diberikan skor sentimen berdasarkan kamus VADER yang berbahasa Inggris. Misalnya kata “Good” 2.5, “Bad” -1.5, dan “okey” 0.5. Setiap token yang telah diberikan *score* akan ditambahkan untuk menghasilkan sebuah *compound score* [47].

4. Modeling

Pada tahap ini data yang telah melewati fase *data preparation* akan dilakukan pemisahan *dataset* menjadi data *train* dan data *test* dengan rasio 80:20. Rasio ini dipilih sesuai dengan penelitian terdahulu yang

menjadi acuan pada analisis yaitu jurnal [11] dan [12]. Namun sebelum melakukan proses *split data* diperlukannya melakukan teknik *oversampling SMOTE* sehingga data yang digunakan dapat lebih seimbang [10]. Pemilihan metode *Oversampling SMOTE* didasari oleh kemampuannya dalam menyeimbangkan data yang tidak seimbang dan juga didasari oleh penelitian terdahulu yang telah memakai metode SMOTE [10] dan [13] serta juga saran dari penelitian terdahulu [14] dan [16]. Berdasarkan hasil proses *labeling* jumlah keseluruhan setiap sentimen dalam persentase sebesar 51,33% untuk label positif, 29,71% untuk label negatif, dan 18,96% untuk label netral. Hal ini menunjukkan bahwa mayoritas sentimen merupakan positif. Apabila dirasioikan maka sentimen positif dengan sentimen negatif mempunyai perbedaan sebesar 1.73 kali lipat dibandingkan sentimen negatif, sedangkan sentimen positif dengan sentimen netral mempunyai perbedaan sebesar 2.71 kali lipat dibandingkan sentimen netral. Hal tersebut memerlukan penyeimbangan pada data dikarenakan perbedaan tingkat jumlah data [48]. Penyeimbangan data diperlukan supaya menghindari adanya *bias* pada kelas mayoritas tertentu sehingga mengurangi terjadinya *overfitting* pada model dengan menggunakan metode SMOTE [48]. *Overfitting* merupakan keadaan dimana suatu model mempelajari data pelatihan yang terlalu baik sehingga menyebabkan kinerja yang buruk. *Dataset* yang telah dipisah dapat dilakukan implementasi ke dalam *Machine Learning* dengan menggunakan algoritma Random Forest dan Decision Tree untuk kedua *game* Mobile Legends Bang-Bang dan League of Legend. Hasil proses tersebut akan menghasilkan nilai akurasi, *precision*, *recall* dan *F1-Scores*.

5. Evaluation

Pada tahap ini peneliti akan melakukan pengukuran pada akurasi dan kinerja model yang didapatkan dari kedua *game online* tersebut dalam mengklasifikasi sentimen pada data pelatihan dan akan dibandingkan dengan penelitian terdahulu yang menggunakan algoritma serupa.

6. Deployment

Tahap terakhir ialah dengan melakukan pembuatan laporan atau mengimplementasikan hasil *modeling* proses *data mining* kepada para pengembang aplikasi. Model yang telah diuji dan dievaluasi dapat dibuatkan laporan dan hasil laporan tersebut dapat menjadi informasi pengembangan aplikasi mereka serta pembuatan sebuah *prototype* dalam bentuk situs pengolahan data sentimen. *Prototype* yang akan dikembangkan dapat berfungsi untuk melakukan proses sentimen analisis untuk aplikasi yang ada di Google Playstore dengan menggunakan *framework* Streamlit. Situs tersebut dapat menampilkan visualisasi dari hasil sentimen yang telah diproses untuk aplikasi yang di *input*.

3.3 Teknik Pengumpulan Data

Data akan dikumpulkan dengan menggunakan bahasa pemrograman Python melalui aplikasi Google Colaboratory, data tersebut akan diambil dari beberapa ulasan dan komentar yang terdapat di halaman Google Playstore proses ini biasanya disebut juga dengan istilah data *scraping*. Aplikasi Google Colaboratory menyediakan fitur *library* bernama Google-play-scraper yang berfungsi untuk menarik data ulasan sesuai dengan ID aplikasi yang di *input* pada *script*. Data ulasan yang akan diambil sekitar 2000 ulasan untuk kedua *game* tersebut, 1000 data untuk *game* Mobile Legends: Bang-Bang dan 1000 data *game* Leagues of Legends yang akan diambil dari Google Playstore dengan menggunakan teknik *scraping*. Variabel yang akan ditarik berupa “userName”, “score”, “at”, dan “content”.

3.4 Teknik Analisis Data

Tabel 3. 3 Perbandingan *Tools*

Tools	Kelebihan	Kekurangan
Google Colaboratory	<i>Built in library</i> lengkap, berbasis <i>Cloud</i> , mudah diakses dan <i>edit</i> , mempunyai fitur GPU dan CPU gratis.	Penggunaan Google Colab secara <i>continuous</i> hanya 12 jam, saat lewat dari itu maka pengguna akan dipindahkan ke <i>instance</i> lain yang

		dimana harus <i>install</i> kembali <i>library</i> tambahan.
Anaconda	Mudah, cepat saat mengunduh <i>package</i> , mengatur <i>library</i> depedensi dengan mudah, gratis dan <i>opensource</i> .	Pengguna harus sangat paham dengan bahasa Python, kurang ideal untuk mengerjakan tugas yang <i>memory intensive</i> , dan program akan sulit diakses dengan perangkat lain karena bersifat <i>local host</i> .

Penelitian ini akan menggunakan *tools* Google Colaboratory yang mempunyai *library* sangat lengkap. Google Colaboratory juga menyediakan fitur GPU dan CPU serta berbasis *Cloud* artinya peneliti tidak perlu takut akan kehilangan memori dan *file* akan ditampung pada *drive* yang telah dibuat. Program yang telah dibuat juga akan lebih mudah diakses dengan perangkat apapun. Perbandingan *tools* yang digunakan dengan *tools* lain disajikan pada tabel 3.3.

Python juga akan menjadi bahasa pemrograman yang dipakai dalam penelitian ini. Bahasa ini digunakan dikarenakan sederhana, memiliki *library* yang luas, ringkas, dan juga sintuitif [49]. Berikut merupakan perbandingan antara bahasa Python dengan bahasa R:

Tabel 3. 4 Perbandingan Bahasa Python dan R

Bahasa Pemrograman	Kelebihan	Kekurangan
Python	Mudah dipelajari dan digunakan, mendukung IoT, <i>opensource</i> , fleksibel.	Menggunakan memori yang cukup tinggi, memiliki banyak batasan desain.
R	<i>Opensource</i> , banyak <i>package</i> .	Sulit untuk dipahami bagi pemula, terlalu kompleks, eksekusi terbilang lama

Pada tabel 3.4 menunjukkan perbandingan bahasa program pengolahan data antara bahasa Python dengan bahasa R. Perbandingan tersebut membuat peneliti memutuskan untuk menggunakan bahasa Python sebagai bahasa pengolah data peneliti ini. Hal ini disebabkan karena bahasa Python lebih mudah untuk dipelajari serta digunakan dan juga lebih fleksibel ketimbang bahasa R. Meskipun Python

akan memakan banyak memori dalam mengolah data yang memory *intensive task* tidak akan menjadi masalah bagi penelitian ini, dikarenakan peneliti menggunakan Google Colaboratory sebagai *toolsnya* yang dimana *program file* akan secara otomatis tersimpan pada Google Drive.

Tabel 3. 5 *Library* yang Digunakan pada Penelitian

<i>Library</i>	Kegunaan
google-play-scraper	Untuk melakukan teknik <i>scraping</i> pada aplikasi Google Playstore.
Pandas	Untuk membaca <i>file</i> dengan format .csv dan membuat <i>dataframe</i> .
stopwords	Untuk melakukan proses <i>Stopwords</i> yang berfungsi untuk mengurangi kata-kata yang tidak mempunyai arti khusus.
StemmerFactory	Untuk melakukan proses <i>Stemming</i> yang bertujuan untuk mengurangi kata-kata imbuhan menjadi kata dasar.
GoogleTranslator	Untuk melakukan proses penerjemahan teks pada data dari bahasa Indonesia ke Inggris.
SentimentIntensityAnalyzer	Untuk melakukan proses sentimen pada data yang telah melakukan proses <i>labeling</i> .
word_tokenize	Untuk melakukan tokenisasi data atau memisah-misahkan teks.

Pada tabel 3.5 merupakan *library* yang akan digunakan dalam penelitian ini. *Library* tersebut dapat di *import* secara langsung pada Python namun, terdapat beberapa *package* yang diperlukan proses pengunduhan. *Package* yang telah di unduh dapat di *import* untuk digunakan *library* nya.

