BAB III
METODOLOGI PENELITIAN

3.1. Subjek dan Objek Penelitian

Subjek pada penelitian ini adalah ulasan pengguna aplikasi Brimo pada *platform google play store*. Sedangkan objek penelitiannya yaitu akurasi model *pretrained* IndoBERT dan BiLSTM untuk anotasi sentimen ulasan pengguna Brimo.

3.2. Alat dan Bahan Penelitian

Alat dan bahan yang digunakan pada pengerjaan penelitian ini yaitu:

3.2.1. Perangkat Keras / *Hardware*

Perangkat keras yang digunakan pada penelitian ini sebagai berikut:

1. Laptop Lenovo Yoga Slim 7
2. RAM 16GB
3. Processor AMD Ryzen 7 4800u
4. Storage 1TB SSD

3.2.2. Perangkat Lunak / *Software*

1. Python
2. Google Colaboratory
3. Numpy
4. Pandas
5. Matplotlib

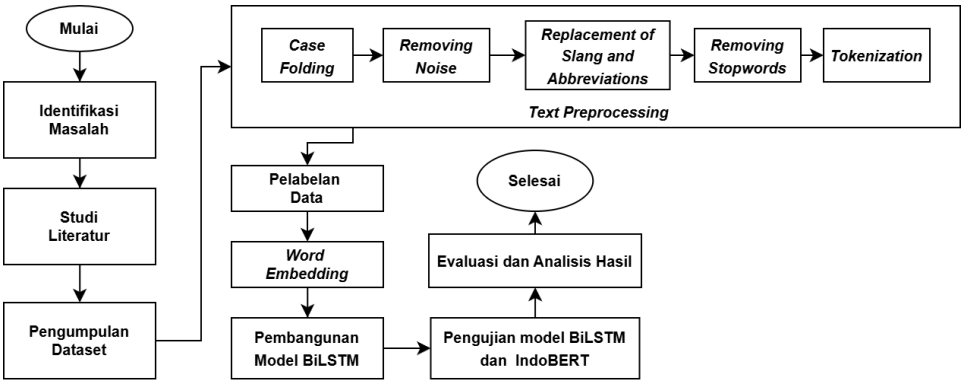
3.2.3. Bahan / Data

Sumber data pada penelitian ini diambil dari ulasan pengguna aplikasi Brimo dalam kurun waktu Maret 2024 - Mei 2024, sebanyak 20.000 baris, dengan rasio pengambilan data yang berlabel negatif dan positif sama yaitu 1:1 untuk menghindari *overfitting.*

3.3. Diagram Alir Penelitian

Anotasi sentimen memerlukan beberapa langkah dalam penelitian ini. Penulis memulai dengan mengidentifikasi masalah, melakukan studi

literatur untuk menjabarkan landasan teori penelitian, mengumpulkan dataset, *text preprocessing* supaya data menjadi bersih, pelabelan data, *word embedding, splitting dataset,* pemodelan BiLSTM. Setelah pemodelan BiLSTM selesai selanjutnya model BiLSTM terbaik dan juga *pretrained* model IndoBERT akan diuji menggunakan dataset validasi yang berasal dari labelling otomatis berdasarkan score. Setelah didapatkan hasil maka akan dievaluasi menggunakan confusion matrix untuk dibandingkan hasilnya. Adapun tahapan yang penulis lakukan seperti yang dijelaskan pada gambar 3.1 sebagai berikut.



Gambar 3. 1 Diagram alir penelitian

3.3.1 Identifikasi Masalah

Tahap pertama, penulis membuat rumusan masalah dengan melihat ulasan komentar aplikasi Brimo pada *platform google play store*. Selanjutnya peneliti menentukan tujuan dan manfaat dari penelitian ini.

3.3.2 Studi Literatur

Peneliti mengumpulkan informasi dari berbagai sumber seperti buku, jurnal, atau literatur yang relevan dengan topik penelitian. Sumber-sumber ini dijadikan referensi untuk mendukung penelitian, termasuk informasi terkait analisis sentimen, klasifikasi, scraping, serta metode *pretrained* IndoBERT dan BiLSTM yang akan diterapkan dalam penelitian ini.

3.3.3 Pengumpulan Dataset

Pengumpulan data dilakukan dengan cara *scraping* menggunakan *library* yang ada pada python yaitu google-play-scraper. Data yang diambil merupakan teks yang berbentuk ulasan mengenai aplikasi Brimo, jumlah data 20.000 baris dengan 10.000 data untuk *training,* 9.000 data untuk testing dan 1.000 data untuk validation. Serta untuk menguji keakuratan model BiLSTM dan *pretrained* IndoBERT, maka diambil 2.000 data baru dari rentang waktu Februari 2024 - Mei 2024. Berikut contoh dataset pada Tabel 3.1.

Tabel 3. 1 Dataset

No	ulasan	score	date
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	5	2023-03-05 23:43:27
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	2	2023-03-05 21:52:49
3	Tp ceria ga bisa di pakai	4	2023-03-05 17:33:52

3.3.4 *Text Preprocessing*

Format data teks yang sudah diambil gaya penulisannya masih tidak terstruktur, maka diperlukan proses *text preprocessing* supaya data yang diolah menjadi lebih terstruktur. Pada penelitian ini terdapat 5 tahapan yang dilakukan, yaitu *case folding*, penghapusan noise, penggantian slang dan singkatan, tokenisasi, stemming, serta penghapusan *stopwords.*

a). *Case Folding*

Case folding merupakan tahapan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil (lowercase).

Tabel 3. 2 *Case folding*

No	Data	Case Folding
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	cepat tepat bersama brimo bri ya lancar semuanya
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	apk nya bagus tpi kok akun saya keblokir ya!!!
3	Tp ceria ga bisa di pakai	Tp ceria ga bisa di pakai

b). *Removing Noise*

Proses *removing noise* merupakan tahapan untuk menghilangkan karakter yang tidak penting seperti url, email, html, tags, tanda seru (!), tanda titik (.), dan tanda baca lainnya yang dapat dilihat pada tabel 3.3.

Tabel 3. 3 *Removing noise*

No	Data	Removing Noise
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	cepat tepat bersama brimo bri ya lancar semuanya
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	apk nya bagus tpi kok akun saya keblokir ya
3	Tp ceria ga bisa di pakai	tp ceria ga bisa di pakai

c). *Replacement of Slang and Abbreviations.*

Proses penggantian slang dan singkatan dalam teks *preprocessing* melibatkan pengubahan istilah informal atau singkatan ke bentuk yang lebih formal atau standar. Contohnya kata ‘ga’ menjadi ‘tidak’, ‘tp’ menjadi ‘tapi’, ‘apk’ menjadi ‘aplikasi’. Hal ini dapat membantu meningkatkan pemahaman dan analisis teks seperti pada tabel 3.4.

Tabel 3. 4 *Replacement of slang and abbreviations*

No	Data	Hasil Penggantian Slang dan Singaktan
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	cepat tepat bersama brimo bri ya lancar semuanya
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	aplikasi bagus tapi kok akun saya keblokir ya
3	Tp ceria ga bisa di pakai	tapi ceria tidak bisa di pakai

d). *Removing Stopwords*

Removing stopwords adalah langkah preprocessing teks yang melibatkan penghapusan kata-kata umum yang tidak memberikan banyak informasi secara khusus dalam suatu konteks. *Stopword* adalah kata-kata umum seperti "dan", "atau", "yang", "di", dan sebagainya, yang sering muncul dalam suatu teks tetapi tidak membawa makna khusus. Berikut contohnya pada tabel 3.5.

Tabel 3. 5 *Removing stopwords*

No	Data	Removing Stopwords
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	cepat, tepat, bersama, brimo, bri, lancar, semua
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	aplikasi, bagus, tapi, kok, akun, blokir
3	Tp ceria ga bisa di pakai	tapi, ceria, tidak, bisa, pakai

e). *Tokenization*

Tokenisasi merupakan langkah *text preprocessing* untuk mengolah teks dengan cara membagi teks menjadi bagian yang lebih kecil yang disebut token, yang dapat berbentuk kata, frasa, atau

karakter sesuai dengan kebutuhan analisis. Seperti contoh pada tabel 3.6 berikut.

Tabel 3. 6 *Tokenization*

No	Data	Hasil Tokensisasi
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	[cepat, tepat, bersama, brimo, bri, lancar, semua]
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	[aplikasi, bagus, tapi, kok, akun, blokir]
3	Tp ceria ga bisa di pakai	[tapi, ceria, tidak, bisa, pakai]

3.3.5 Pelabelan Data

Pelabelan data dilakukan secara otomatis dengan menetapkan label negatif untuk rating 1-2 dan label positif untuk rating 4-5. Rating 3 tidak disertakan untuk menghindari bias terhadap salah satu label. Serta dilakukan *manual labelling* oleh manusia karena terdapat beberapa data yang memiliki rating tinggi tapi sentimennya negatif ataupun sebaliknya.

Tabel 3. 7 Pelabelan Data

No	ulasan	score	sentimen
1	Cepat tepat bersama Brimo BRI ya lancar semuanya	5	positif
2	APK NYA BAGUS TPI KOK AKUN SAYA KEBLOKIR YA!!!	2	negatif
3	Tp ceria ga bisa di pakai	4	positif

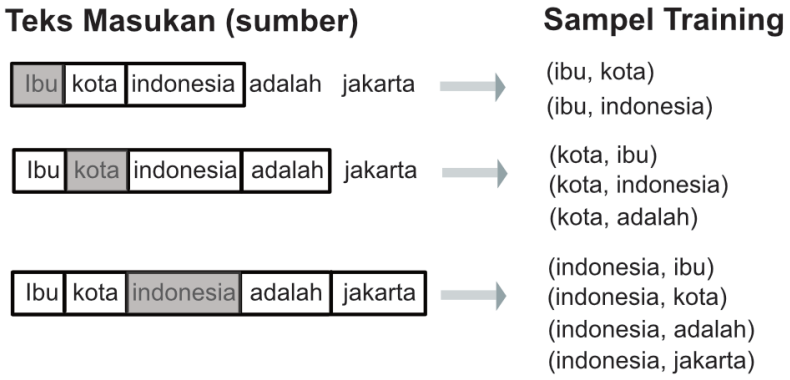
3.3.6 *Word Embedding*

Menampilkan fitur kata dengan tingkat kemiripan semantik yang tinggi, Word2Vec menggunakan metode kerja jaringan saraf tiruan (JST) [37]. CBOW dan *skip-gram* adalah dua arsitektur yang digunakan. Model *skip-gram* digunakan dalam penelitian ini untuk memprediksi kata-kata target dan kata konteks. Posisi jumlah kata sesudah dan sebelum kata

konteks dapat dilihat pada window. Ukuran *window* 2 dapat dilihat pada gambar 3.2

Contoh kalimat: ibu kota indonesia adalah jakarta

Jika contex kata "ibu" diberikan, kata target di sebelah kiri kosong, sedangkan kata "kota" dan "indonesia" di sebelah kanan akan menjadi pasangan karena berada pada 2 kata selanjutnya dari kata “ibu” . Oleh karena itu, pasangan {konteks kata, kata target} yang dihasilkan adalah {(ibu, kota), (ibu, indonesia)}.



Gambar 3. 2 Skip-gram dua *window*

Input dan *output* diambil dari pasangan {kata konteks, kata target}. Berikut adalah langkah kerja proses *skip gram*.

1). Langkah pertama yaitu mengubah kata menjadi angka pada vektor menggunakan *one-hot-encoding,* sehingga didapatkan nilai dimensi vektor [1, |v|], yang terlihat pada tabel 3.8 berikut

Tabel 3. 8 *Encoding* hasil kata

Id-term	Term Ibu	Term kota	Term Indonesia	Term adalah	Term jakarta
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0

3	0	0	0	1	0
4	0	0	0	0	1

Dari tabel 3.8 memperlihatkan vektor kata dihasilkan dari *encoder*. Contoh kata "ibu" setelah proses encoding menjadi vektor dengan nilai [1 0 0 0 0]. Urutan term pada matrik hasil *encoding* akan diacak secara random sehingga urutan termnya tidak selalu urut sesuai dengan urutan kata pada kalimat *input*, kemudian vektor w(t) hasil *encoding* menjadi vektor *input* dan target *ouput*. W(t) manakah yang akan menjadi *input*-nya dan mana yang menjadi *output*-nya jika dilihat dari pasangan {konteks kata, target kata}

2). Vektor *input* dari melewati *hidden layer* dari sejumlah |v| *neurons*, |v| adalah ukuran fitur yang digunakan. *Hidden layer* ini merupakan perkalian antara bobot vektor W[|v|, N] dan *input* vector w(t). Matrik bobot W merupakan N adalah jumlah context atau *window*, dan W[|v|, N] akan menjadi *output* (H[1, N]), dimana H adalah *hidden layer.*

3). Tidak adanya fungsi aktivasi menyebabkan H[1,k] akan melewati *layer output* secara langsung.

4). *Output* layer dihitung dengan hasil perkalian antara H[1, N] dan W'[N, |v|] sehingga didapatkan hasil vektor U.

5). Menghitung probabilitas untuk setiap vektor dengan fungsi *softmax* menggunakan persamaan (1). Satu nilai *one hot encoding* dihasilkan dari tiap iterasi . Term atau *word* dengan probabilitas tertinggi adalah target kata pada konteks kata yang diberikan. Seperti yang ditampilkan pada persamaan 3.1 berikut.

$$p(w_{c,j} = w_{O,c} \mid w_I) = \frac{\exp u_{c,j}}{\sum_{j'=1}^{V} \exp u_{j'}} \quad \text{.....(3.1)}$$

Dimana :

1) w (c, j) merupakan kata ke-j yang akan diprediksi pada saat posisi context ke-c;
2) w (O, c) adalah kata sebenarnya yang ada pada posisi context ke-c;
3) w (I) merupakan kata input;
4) u (c, j) merupakan nilai ke-j dalam vektor U saat memprediksi kata untuk posisi konteks ke-c.

Berdasarkan parameter kita bisa membuat window = 2, ukuran fitur atau dimensi vektor = 4, minimum frekuensi kata = 1 (proses training akan melibatkan tiap kata yang muncul minimal sekali). Skip-gram akan membuat ukuran fitur bernilai 100 secara *default* jika tidak dilakukan penentuan parameter sebelumnya.

Tabel 3. 9 Nilai vektor serta output kata

Id term	Term Ibu	Term kota	Term Indonesia	Term adalah	Term jakarta
0	0.18279415	0.12675655	0.16894233	0.01907164	0.18279415
1	0.07191449	0.02479684	-0.20713037	-0.23622045	0.07191449
2	-0.11341533	0.16385129	-0.12150401	-0.04540044	-0.11341533
3	-0.1253857	-0.09408429	0.18451262	-0.03833678	-0.1253857
4	-0.23257375	-0.17792022	0.16147181	0.2243247	-0.23257375

Pada tabel 3.9 kolom v0, v1,v2,v3 merupakan fitur atau dimensi yang dibentuk sesuai ukuran fitur yang ditentutkan sebelumnya. Matrik vektor hasil word2vec merepresentasikan seluruh vektor term dalam korpus yang digunakan.

6). Model klasifikasi akan memilih vektor *output* untuk digunakan sebagai *input* pemodelan klasifikasi. Metode yang digunakan adalah *average base*, yaitu mencari nilai rata-rata dari vektor-vektor kata penyusun kalimat yang akan diprediksi jenis sentimennya, seperti contoh berikut:

Kalimat x: ibu kota indonesia

Kalimat y: ibu kota jakarta

Dengan melihat tabel 3.9, nilai rata-ratauntuk kalimat x pada tabel 3.10 dan kalimat y pada tabel 3.11. Nilai rata-rata setiap fitur inilah yang akan digunakan sebagai atribut untuk membangun model klasifikasi.

Tabel 3. 10 *Word output* dan vektor kalimat x

Kata	**V0**	**V1**	**V2**	**V3**
ibu	0.18279415	0.12675655	0.16894233	0.01907164
kota	0.07191449	0.02479684	-0.20713037	-0.23622045
indonesia	-0.11341533	0.16385129	-0.12150401	-0.04540044
Rata-rata	**0.047098**	**0.105135**	**-0.05323**	**-0.08752**

Tabel 3. 11 *Word output* dan vektor kalimat y

Kata	**V0**	**V1**	**V2**	**V3**
ibu	0.18279415	0.12675655	0.16894233	0.01907164
kota	0.07191449	0.02479684	-0.20713037	-0.23622045
jakarta	-0.23257375	-0.17792022	0.16147181	0.2243247
Rata-rata	**0.007378**	**-0.00879**	**-0.041095**	**-0.002392**

3.3.7 Pembangunan Model BiLSTM

Tahap pertama pemodelan diawali dengan membagi dataset sejumlah 20.000, dengan rincian 10.000 untuk *training* dan 9.000 untuk *testing* dan 1.000 untuk *validation.* Sejumlah 10.000 data akan digunakan untuk *training* pemodelan sentimen dengan BiLSTM, 9.000 baris data lainnya untuk *testing* dengan BiLSTM serta 1.000 lainnya untuk validasi.

Langkah selanjutnya yaitu melabeli dataset pemodelan sejumlah 20.000 ulasan dengan pelabelan secara otomatis untuk setiap ulasan yang memiliki rating 1-2 dengan sentimen negatif dan rating 4-5 untuk sentimen positif, pelabelan tersebut juga akan divalidasi ulang manual oleh manusia, karena manusia lebih mampu untuk mengetahui konteks secara lengkap. Setelah data terlabeli kemudian data di-*embedding* ke dalam bentuk vektor,

kemudian dilakukan pemodelan dengan arsitektur BiLSTM dengan beberapa parameter *tuning* pada tabel 3.12 sebagai berikut.

Tabel 3. 12 *Parameter tuning* model BiLSTM

No	Parameter	Detail
1	Dropout Rate	Mengontrol seberapa banyak neuron akan di-"drop out" selama pelatihan untuk mencegah overfitting. Dropout adalah teknik regularisasi yang dapat membantu mencegah model mempelajari detail-detail kecil dari data pelatihan yang mungkin tidak umum. Nilai dropout yang akan dilakukan yaitu 0.1, 0.2, 0.3
2	Epochs	Jumlah kali keseluruhan dataset pelatihan dilewati maju dan mundur melalui jaringan saraf. Penting untuk menemukan nilai yang optimal untuk menghindari *underfitting* atau *overfitting*. Epoch yang digunakan 20, 30, 50.
3	Embedding Layer	Lapisan embedding sebelum layer BiLSTM, parameter seperti embedding layer yang akan digunakan yaitu *Word2Vec (CBOW dan Skip-Gram)*

Setelah pemodelan selesai, dilakukan evaluasi untuk mencari model klasifikasi BiLSTM terbaik. Evaluasi model dilakukan dengan melihat manakah arsitektur model BiLSTM terbaik yang menghasilkan nilai akurasi yang tinggi.

3.3.8 Pengujian model BiLSTM dan IndoBERT

Pengujian dilakukan dengan dataset berjumlah 2.000 data baru yang telah dilabeli secara otomatis dan pelabelan ulang manual oleh manusia. Data yang telah terlabeli tersebut akan diujikan pada model BiLSTM dan *pretrained* model IndoBERT.

3.3.9 Evaluasi dan Analisis Hasil

Evaluasi model dilakukan untuk mengetahui kinerja model. Proses evaluasi model dilakukan dengan melihat tingkat akurasi metode melalui confusion matrix untuk tiap model. Setelah data test yang baru telah selesai diujikan terhadap model BiLSTM dan IndoBERT, maka akan dihasilkan

klasifikasi nilai akurasi yang didapatkan dan dapat ditarik kesimpulan dari penelitian yang telah dilakukan.