BAB III
METODOLOGI PENELITIAN

3.1 Objek dan Subjek Penelitian

Pada bab 1 telah dijelaskan adanya latar belakang masalah yang akan diteliti yaitu *aspect based sentiment analysis* komentar mahasiswa pada data EDOM ITTP. Subjek dari penelitian ini adalah pengisi survei EDOM (mahasiswa), sementara objek dari penelitian ini adalah deteksi opini mahasiswa menggunakan *aspect based sentiment* pada EDOM ITTP.

3.2 Alat dan Bahan

Dalam penelitian ini diperlukan beberapa spesifikasi minimum dari peralatan dan bahan yang akan digunakan, seperti data, perangkat lunak dan perangkat keras, yaitu sebagai berikut.

3.2.1 Data

Dataset yang digunakan dalam penelitian ini adalah dataset komentar mahasiswa terhadap proses pembelajaran yang didapatkan dari data EDOM Institut Teknologi Telkom Purwokerto antara tahun akademik tahun 21/22 dan tahun 22/23.

3.2.2 Spesifikasi Kebutuhan Perangkat Keras

Berikut merupakan spesifikasi perangkat keras yang dibutuhkan pada penelitian ini yang terdapat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Kebutuhan Perangkat keras

No.	Komponen	Spesifikasi
1	*CPU*	Intel i5-1135G7
2	*RAM*	16GB Ram DDR4

No.	Komponen	Spesifikasi
3	*GPU*	Intel TigerLake-LP GT2 [Iris X]
4	*Storage*	512 GB

3.2.3 Spesifikasi Kebutuhan Perangkat Lunak

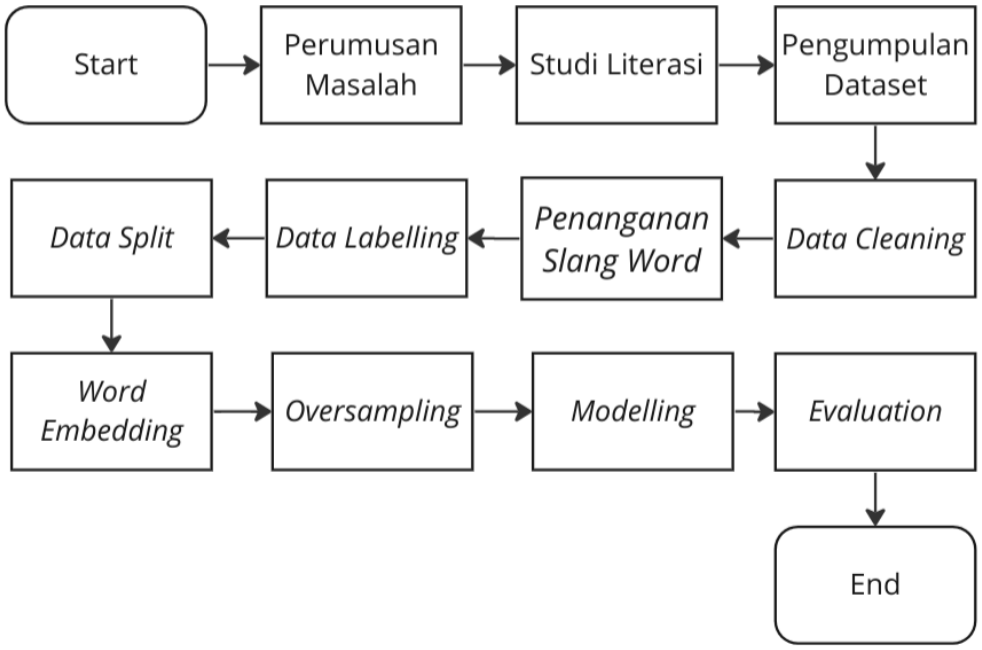
Berikut merupakan kebutuhan perangkat lunak yang akan dibutuhkan dalam penelitian, terdapat pada Tabel 3.2.

Tabel 3.2 Spesifikasi Kebutuhan Perangkat Lunak

No	Kebutuhan	Keterangan	Fungsi
1.	Sistem Operasi	Ubuntu 22.04 LTS	Untuk menjalankan aplikasi
2.	Aplikasi	Google Collaboratory	Untuk membangun model CNN
		Miro.com	Untuk membuat diagram

3.3 Diagram Alir Penelitian

Pada penelitian ini penulis menggunakan diagram alir dalam proses pengembangannya. Di bawah ini adalah ilustrasi diagram alir yang penulis gunakan.



Gambar 3.1 Diagram Alir Penelitian

Gambar 3.1 menjelaskan tahapan penelitian yang akan dilakukan dimulai dari tahap :

3.3.1 Perumusan Masalah

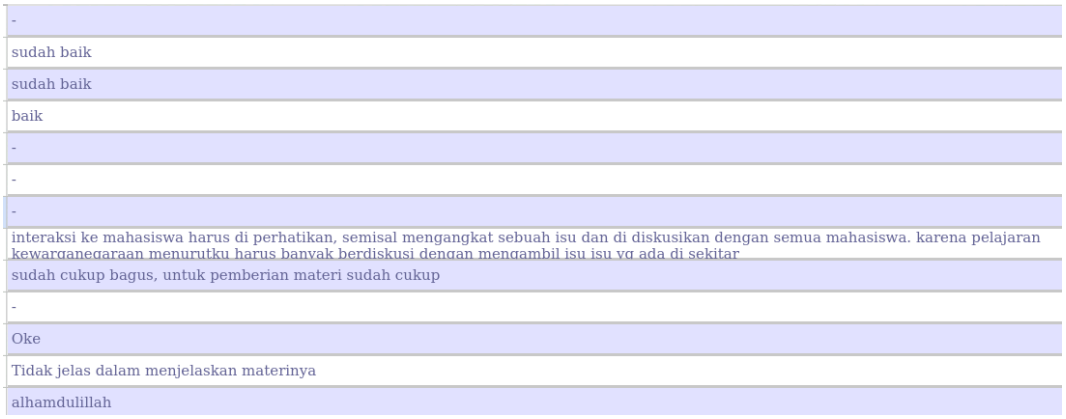
Pada tahapan ini, dalam rangka meningkatkan pemahaman dan efektivitas analisis sentimen berbasis aspek pada data EDOM di kampus ITTP, akan dilakukan identifikasi permasalahan yang mungkin muncul. Identifikasi ini bertujuan untuk mengidentifikasi kendala dan tantangan yang perlu diatasi agar analisis sentimen dapat dilakukan dengan lebih akurat dan informatif.

3.3.2 Studi Literasi

Pada tahap ini akan dilakukan riset kepustakaan. Penelitian yang akan dilakukan adalah dengan mengumpulkan beberapa data terkait dengan topik analisis sentimen berbasis aspek, dan penelitian ini menggunakan metode CNN. Data dari permasalahan tersebut dapat diperoleh dari jurnal, artikel, buku maupun survei.

3.3.3 Pengumpulan Dataset

Pada tahap ini dilakukan untuk mencari dataset yang akan digunakan pada penelitian ini. Data yang dikumpulkan berasal dari data EDOM Institut Teknologi Telkom Purwokerto, dengan *range* data dari tahun ajaran 21/22 dan 22/23. Aspek yang akan digunakan adalah cara mengajar dan kelengkapan materi, untuk data yang masih mentah terdapat sebanyak 5116. Gambar 3.2 adalah contoh dari dataset EDOM di ITTP yang masih mentah.



Gambar 3.2 Contoh *Dataset* EDOM ITTP.

3.3.4 *Data Cleaning*

Setelah mendapatkan *dataset,* maka diperlukan *data cleaning* sebelum data digunakan untuk tahap selanjutnya, perlunya *data cleaning* karena digunakan untuk melakukan penghapusan data yang tidak diperlukan yang bisa mempengaruhi performa dari model. Pada proses *data cleaning* ini akan dilakukan penyaringan kata untuk menghilangkan simbol-simbol, kolom yang

kosong, karakter khusus dan emotikon. Proses ini bertujuan agar saat pengolahan data tidak terjadi error yang tinggi dikarenakan data yang tidak penting.

3.3.5 Penanganan *Slang Word*

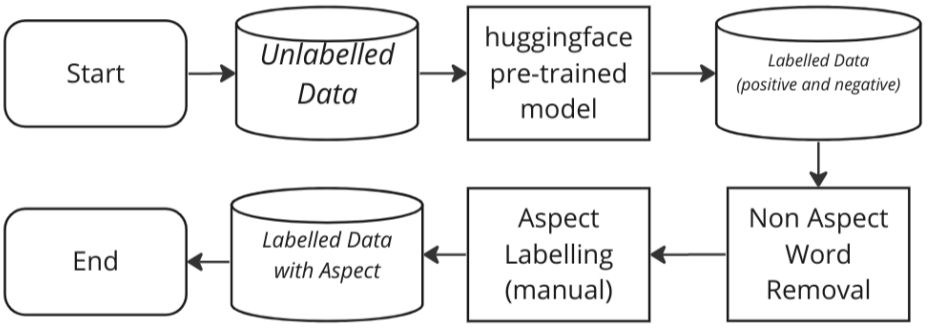
Setelah melewati data *cleaning*, maka tahap selanjutnya adalah melakukan penanganan *slang word*, *slang word* merupakan kata-kata yang tidak formal, penanganan *slang word* sendiri merupakan cara agar meningkatkan kualitas dari data sebelum memasuki tahap selanjutnya. Ilustrasi Penanganan *Slang Word* ada pada Tabel 3.3

Tabel 3.3 Ilustrasi Penanganan Slang Word

Teks mengandung slang	Teks tidak mengandung slang
Kurangi marah ya ibu, biar awet muda dan anak mahasiswa nya senang sama budos ny	Kurangi marah ya ibu, biar awet muda dan anak mahasiswa nya senang sama ibu dosen nya

3.3.6 *Data Labelling*

Data yang sudah melewati penanganan *slang word* akan melewati pelabelan data terlebih dahulu. Hal ini dilakukan agar sistem dapat memahami makna dari setiap kata yang akan diujikan. Berikut merupakan gambaran dari data *Labelling*.



Gambar 3.3 Flowchart Data Labelling.

Pada tahap pertama adalah mencari apakah kalimat tersebut apakah merupakan kalimat positif atau negatif yang akan dilakukan secara otomatis, cara otomatis ini akan menggunakan *pre-trained* model dari *huggingface* yaitu *bert-base-indonesian-1.5G-sentiment-analysis-smsa*, kemudian tahap kedua adalah melakukan penghapusan kata-kata yang tidak bermakna secara aspek (cara mengajar dan kelengkapan materi), tujuan dari penghapusan kata-kata yang tidak mengandung aspek adalah menghapus kolom yang hanya mengandung kata yang tidak bermakna, sebagai contoh terdapat kolom yang hanya berisi dengan teks "sudah baik", maka kolom itu akan dihapus dari dataset dan dilanjutkan dengan penggantian label sentimen netral ke label yang sentimen yang lebih cocok (label positif atau negatif), pada tahap ketiga akan dilanjut ke dalam tahap pelabelan aspek, pada tahap ini penulis akan menggunakan label cara mengajar dan kelengkapan materi.

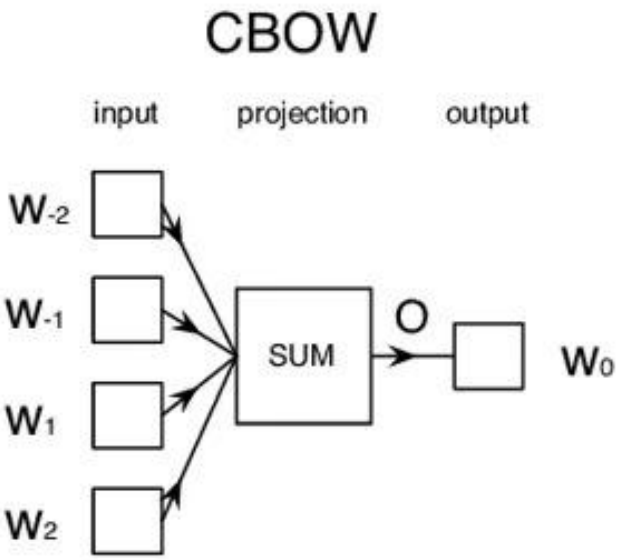
3.3.7 *Data Splitting*

Data *splitting* pada klasifikasi berbasis teks untuk model analisis sentimen dan pembuatan model *aspect* melibatkan pembagian dataset teks menjadi dua subset yang berbeda, yaitu untuk tujuan pelatihan (*training*), untuk tujuan pengujian (*testing*), pembagian data akan dilakukan sebanyak 80:20, 80 persen untuk *training* dan 20 persen untuk *testing*. Tujuan utama dari pembagian data ini adalah untuk menguji kinerja model pada data yang belum pernah dilihat sebelumnya dan memastikan bahwa model mampu menggeneralisasi informasi dengan baik pada data teks baru. Pada tahap ini juga akan dilakukan label mapping, label *mapping* akan mengubah label yang ada di dalam dataset menjadi label angka.

3.3.8 *Word Embedding*

Word2Vec adalah salah satu metode *word embedding* yang berguna untuk menjadikan kata menjadi sebuah vektor. Arsitektur Word2vec hanya terdiri dari 3 layer yaitu *Input*, *Projection* (*Hidden Layer*) dan *Output.* Pada penelitian ini akan menggunakan arsitektur CBOW , CBOW adalah salah satu metode dalam model

Word2Vec, yang merupakan suatu teknik dalam pengolahan bahasa alami (*Natural Language Processing*/NLP) untuk menghasilkan representasi vektor kata (*word embeddings*) dari teks. CBOW bertujuan untuk memprediksi kata target berdasarkan konteks kata-kata di sekitarnya. untuk ilustrasi dari CBOW dapat dilihat pada gambar 3.5.



Gambar 3.4 Contoh Implementasi CBOW

Kemudian pada tahap ini juga akan dilakukan penggunaan *text to sequences,* `*texts_to_sequences*` adalah fungsi kunci dalam pemrosesan teks menggunakan *library* Keras. Dalam konteks tokenisasi, fungsi ini memungkinkan konversi teks menjadi urutan angka berdasarkan token yang telah ditentukan sebelumnya. Misalnya, jika kita memiliki dua kalimat, "Ibu mengajar matkul dengan baik semoga kedepannya bisa lebih baik lagi.". Berikut merupakan contoh dari penggunaan *text to sequences*.

Tabel 3.4 Contoh *Text to Sequences*

Sebelum Text to Sequences	**Sesudah Text to Sequences**
Ibu mengajar matkul dengan baik semoga kedepannya bisa lebih baik lagi	[3, 4, 5, 6, 2, 7, 8, 9, 10, 2, 11]
Terima kasih pak	[12, 13, 14]

Dengan menggunakan metode `*texts_to_sequences*`, teks-teks tersebut diubah menjadi urutan angka yang merepresentasikan token-token yang sesuai dengan kamus token yang telah dibangun. Hasilnya adalah urutan numerik yang dapat digunakan sebagai input untuk model *machine learning*, memungkinkan model untuk mengolah dan memahami teks secara lebih efektif.

Setelah itu, tahap selanjutnya adalah melakukan *padding* pada teks yang sudah di *tokenize sequences. Padding* adalah proses menyesuaikan panjang sekuen data dengan cara menambahkan atau memotong nilai-nilai tertentu pada akhir atau awal sekuen. Dalam konteks pemrosesan teks, *padding* sering digunakan untuk membuat panjang kalimat atau dokumen seragam, memastikan bahwa setiap teks memiliki dimensi yang sama. Berikut merupakan contoh dari penggunaan *padding* menggunakan *pad_sequences*.

Tabel 3.5 Contoh penggunaan *pad_sequences*

Sebelum *Padding*	**Sesudah *Padding***
[3, 4, 5, 6, 2, 7, 8, 9, 10, 2, 11]	[3 4 5 6 2 7 8 9 10 2 11]
[12, 13, 14]	[0 0 0 0 0 0 0 0 12 13 14]

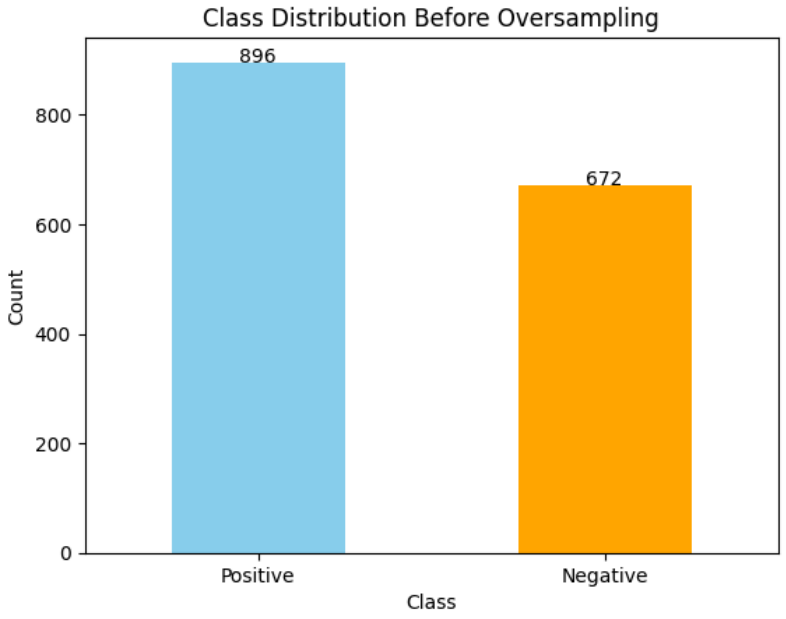
Tabel 3.7 memberikan contoh penggunaan pad_sequences pada data urutan token numerik sebelum dan setelah proses *padding*. Pada contoh pertama, urutan token [3, 4, 5, 6, 2, 7, 8, 9, 10, 2, 11] di-pad dengan nilai nol di awalnya sehingga memiliki panjang yang sama dengan urutan terpanjang, yaitu [3 4 5 6 2 7 8 9 10 2 11]. Ini membantu menciptakan data dengan dimensi seragam, yang sangat berguna saat menggunakan model CNN. Sementara itu, contoh kedua, [12, 13, 14], yang lebih pendek, di-pad sehingga panjangnya menjadi sama dengan urutan terpanjang, yaitu [0 0 0 0 0 0 0 0 12 13 14].

Tahap selanjutnya adalah masuk ke dalam tahap *embedding matrix,* pada tahap ini akan dilakukan untuk menyimpan representasi vektor. *Embedding matrix*

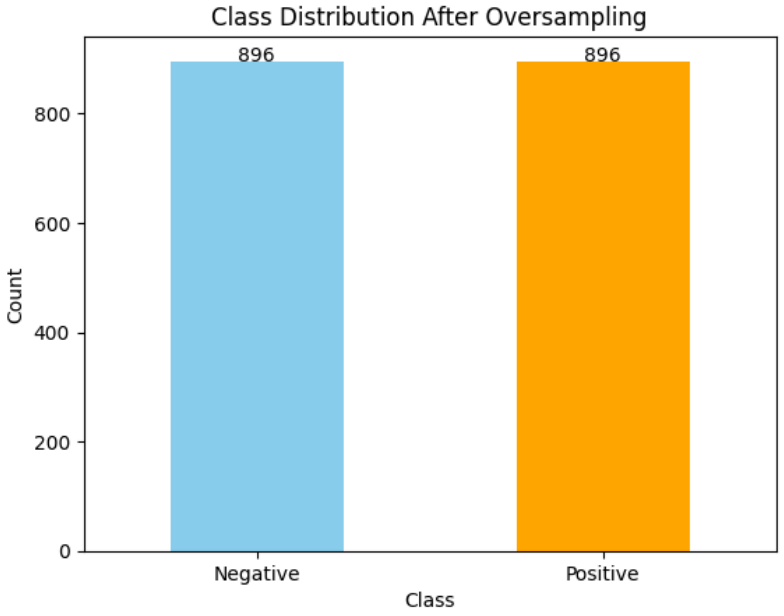
ini akan digunakan sebagai *layer embedding* pada model neural *network*, proses ini melibatkan ekstraksi vektor kata-kata dari model word2vec untuk kemudian disimpan dan digunakan dalam proses tahap *modelling*.

3.3.9 *Oversampling*

Pada tahap ini akan dilakukan penyeimbangan data pada kelas di dalam *dataset* sebelum masuk ke tahap *modelling*. Hal ini dapat mencegah model CNN cenderung memihak satu kelas dan akan membantu meningkatkan akurasi pada model CNN. Berikut merupakan contoh dari dataset sebelum *oversampling* dan sesudah *oversampling.*



Gambar 3.5 Sebelum *Oversampling*



Gambar 3.6 Setelah *Oversampling*

Pada data sebelum *oversampling* yaitu gambar 3.5 terdapat data sebanyak 1568 dataset, yaitu data kelas negatif berjumlah 672 dan kelas positif berjumlah 896. Kemudian setelah memasuki tahap *oversampling* dataset memiliki jumlah total data sebanyak 1792, yaitu kelas positif berjumlah 896 dan kelas negatif berjumlah 896. Pada tahap ini akan dilakukan *experiment* untuk mencari metode atau algoritma *oversampling* yang terbaik untuk model *CNN* nanti, berikut merupakan tabel *experiment* yang akan digunakan untuk mencari algoritma *oversampling* terbaik:

Tabel 3.6 Algoritma *Oversampling*

No.	**Algoritma *Oversampling***
1.	SMOTE
2.	*Random Oversampling*
3.	ADASYN
4.	SMOTE-NC
5.	*Borderline* SMOTE

3.3.10 Modelling

Proses analisis data pada penelitian ini dilakukan dengan menggunakan *python* dengan *tools Google Colab*. Data yang sudah melewati proses *word embedding* dan *oversampling* selanjutnya akan masuk ke *modelling* CNN. Tingkat akurasi yang dihasilkan dari proses ini sangat dipengaruhi oleh proses *text preprocessing* (*data cleaning, penanganan slang word, sop word removing*) dan data *labelling*. apabila proses tersebut tidak dilakukan secara benar maka tingkat akurasi akan terpengaruh. Berikut merupakan contoh konfigurasi arsitektur CNN yang akan digunakan ada pada Tabel 3.7.

Tabel 3.7 Konfigurasi CNN.

Hyper Parameter	Konfigurasi
Jumlah Kernel	64
Panjang Kernel	3x3
Pooling	*Maximum*
Dense	64
Fungsi Aktivasi	ReLu
Neuron Output Layer	1
Optimasi MPL	Adam
dropout	0.5
batch size	32
epoch	40

Pada model CNN dengan konfigurasi pada Tabel 3.7 akan menggunakan kernel 3x3 dan menggunakan *layer pooling* dengan *Max Pooling*. Jumlah *Dense layer* yang digunakan adalah sebanyak 64 *layer*, dan setiap *layer* tersebut akan menggunakan fungsi aktivasi ReLU. Pada akhirnya, *output layer* akan menghasilkan satu *neuron* untuk tugas klasifikasi biner. Model CNN yang dikembangkan oleh penulis juga akan menggunakan fungsi optimisasi Adam dan

dropout sebesar 0.5 untuk menghindari *overfitting*. Selama proses pelatihan model, *batch size* yang digunakan adalah sebanyak 32, dan proses pelatihan akan dilakukan sebanyak 40 *epoch*.

3.3.11 *Evaluation*

Setelah proses pemodelan selesai dilakukan, proses akan dilanjut dengan proses testing dengan data testing yang sudah dipisah dengan *dataset* utama dan evaluasi untuk mendapatkan nilai performa model terbaik. Validasi model digunakan untuk menampilkan nilai akurasi dari model yang sudah dilakukan. Proses validasi dan evaluasi pada model ini menggunakan *Confusion Matrix*. *Library* yang digunakan untuk mendukung penggunaan *Confusion Matrix* adalah *sklearn*. *Library* yang dirujuk akan dipanggil dengan menggunakan *syntax sklearn.metrics.confusion matrix, Confusion Matrix* akan menghasilkan nilai *accuracy, precision, recall dan f1 score.*