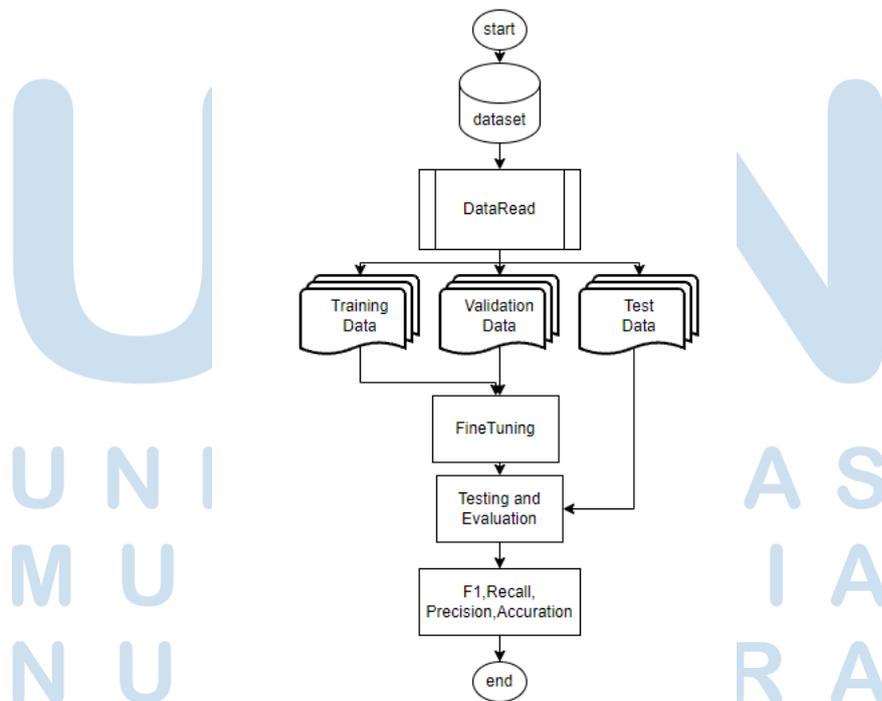


## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Metodologi Penelitian

Pada penelitian ini, metode penelitian yang digunakan adalah studi literatur, *Data preparation*, *Model Development*, *Fine Tuning*, *Testing*, *Model Evaluation*. Pada tahapan studi literatur, peneliti mengumpulkan landasan teori dari masalah yang akan diteliti. *Data preparation* adalah proses untuk pengumpulan data dan persiapan untuk diterapkan pada penelitian. *Model development* adalah proses pemilihan dan pengembangan model yang sesuai dengan masalah yang diteliti. *Fine Tuning* adalah proses pelatihan dari model yang sudah ada agar bisa mendapatkan hasil yang lebih baik. *Testing* adalah proses pengujian dari model yang sudah ada dengan *sample test* yang sudah dipersiapkan sebelumnya. *Model evaluation* adalah proses evaluasi dari perbandingan model yang diuji untuk mengetahui model yang memiliki nilai *F1*, *accuracy*, *recall*, dan *precision* paling tinggi. Tahapan yang dilakukan pada penelitian ini dapat dilihat pada gambar 3.1



Gambar 3. 1 Gambaran alur implementasi

## 3.2. Persiapan Data

### 3.2.1. Pengumpulan Data

Dataset yang digunakan merupakan data ulasan pelanggan dari Tokopedia dan Shopee TokoTeknoPro yang aktif hingga April 2024. Data didapatkan dengan menggunakan metode *webscraping* dan didapatkan 11.397 *review* dari Shopee dan 14.343 *review* dari Tokopedia. Setelah dilakukan *pre-processing* data didapatkan 22.897 data *review* yang bersifat *positive* dan 2121 data bersifat *negative*. Dikarenakan data yang ada tidak seimbang, dan penelitian berfokus hanya pada satu toko yaitu TokoTeknoPro maka dilakukan proses *undersampling* pada data. *Undersampling* dilakukan secara manual dengan aplikasi Libre Calc untuk memfilter data yang memiliki *review* terlalu pendek. Setelah melalui proses *undersampling* didapatkan 2929 data *positive* dan 2121 data *negative*.

Dari keseluruhan data yang ada, untuk masing-masing model data akan dibagi menjadi tiga kebutuhan yaitu data *training*, data *validation* dan data *testing* dengan ratio 70% untuk data *training*, 10% untuk *validation*, dan 20% untuk data *testing*. Data untuk masing-masing model bersumber dari *dataset* yang sama, Namun data untuk setiap kategori baik *training*, *validation*, ataupun *testing* mungkin berbeda antara satu model dengan model yang lain. Hal ini dikarenakan proses pembagian data dilakukan bersamaan dengan proses penyesuaian format data untuk masing-masing model.

### 3.2.2. Pre-processing

Pada *pre-processing* terdapat beberapa prosedur yang dilakukan seperti menghapus data yang ganda, menghapus data yang bernilai nil, menghapus unicode dan emoji, dan mengisi nilai data yang kosong dengan *default value*.

Data yang sudah mengalami proses *pre-processing* kemudian *ditranspose* ke dalam bentuk format masing-masing model. Untuk format data masing-masing model adalah sebagai berikut :

- IndoBERT:

```
1 {sentence};label1,label2,label3,..
```

Gambar 3.2 Struktur data IndoBERT

- Gemini

```
1 {  
2   "messages": [  
3     {  
4       "role": "system",  
5       "content": "instruction"  
6     },  
7     {  
8       "role": "user",  
9       "content": "stringreview"  
10    },  
11    {  
12      "role": "model",  
13      "content": "LABEL"  
14    }  
15  ]  
16 }
```

Gambar 3.3 Struktur data Gemini

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

- GPT
  - *Training and validation*

```
1 {
2   "messages": [
3     {
4       "role": "system",
5       "content": "instruction"
6     },
7     {
8       "role": "user",
9       "content": "stringreview"
10    },
11    {
12      "role": "assistant",
13      "content": "label"
14    }
15  ]
16 }
```

Gambar 3.4 Struktur data *training* dan *validation* gpt

- *Testing*

```
1 {
2   "custom_id": "id",
3   "method": "POST",
4   "url": "/v1/chat/completions",
5   "body": {
6     "model": "modelname",
7     "messages": [
8       {
9         "role": "system",
10        "content": "instruction"
11      },
12      {
13        "role": "user",
14        "content": "stringreview"
15      },
16      {
17        "role": "assistant",
18        "content": "label"
19      }
20    ]
21  }
22 }
```

Gambar 3.5 Struktur data *testing* GPT

### 3.3. Implementasi Model

#### 3.3.1. IndoBERT

Pada pengembangan model LLM *Open Source* yang digunakan adalah IndoBERT. Proses pengembangan model dilakukan melalui Jupyter Notebook berbasis bahasa Python. *Benchmark* yang digunakan adalah model *Pre-Trained* dari “indobert base p1”. Pada konfigurasi model IndoBERT, untuk kebutuhan *sentiment analysis task* yang dipilih adalah “*BertMultiLabelClassification*”. Untuk pengaturan *hyperparameter* yang digunakan adalah nilai *default* dari model yang sudah ada. Nilai *default* dari *hyperparameter* model IndoBERT adalah sebagai berikut:

Tabel 3.1 Default Hyperparameter IndoBERT

Parameter	Value
Learning Rate	2e-05
Train Batch Size	64
Eval Batch Size	64
Optimizer	Adam
Adam Betas	(0.9, 0.999)
Adam Epsilon	1e-08
Learning Rate Scheduler	Linear
Number of Epochs	5
Seed	42

#### 3.3.2. GPT oleh OpenAI

Pada pengembangan GPT, pengembangan model dilakukan melalui platform OpenAI. Peneliti memilih model yang digunakan melalui *Graphical User Interface* yang disediakan. *Base model* yang digunakan adalah “gpt-3.6-turbo-0125”. Konfigurasi *task* untuk kebutuhan *Multi Aspect Based* pada model ini dilakukan dengan menyesuaikan format data yang akan diunggah. Untuk *default hyperparameter* dari model

GPT tidak bisa diakses oleh *public*, namun *developer* bisa mengatur *epoch*, *batch size*, dan *learning rate multiplier* jika dibutuhkan.

### 3.3.3. Gemini oleh Google

Pada pengembangan Gemini, pengembangan model dilakukan melalui platform VertexAI. Peneliti memilih model yang digunakan melalui *Graphical User Interface* yang disediakan. *Base model* yang digunakan adalah “gemini-1.0-pro-002” dengan *region* “us-central1(lowa)”. Konfigurasi *task* untuk kebutuhan *Multi Aspect Based* pada model ini dilakukan dengan menyesuaikan format data yang akan diunggah. Untuk default *hyperparameter* dari model Gemini, tidak dapat diakses oleh *public*.

## 3.4. Pelatihan Model

### 3.4.1. IndoBERT

Proses *training* IndoBERT dilakukan melalui Jupyter Notebook berbasis bahasa Python dengan melakukan penyesuaian *weight* secara berulang berdasarkan dari *loss* dari *batch training* data dan melakukan monitoring *performance matrix*. Pada proses *training*, nilai *epoch* yang digunakan oleh seluruh model adalah 5.

### 3.4.2. GPT oleh OpenAI

Proses *training* GPT dilakukan melalui platform OpenAI. *Fine-tuning* dilakukan dengan *Graphical User Interface*. Pada tampilan ini juga peneliti memilih *base model*, mengunggah data JSONL yang akan di *train*, dan mengunggah data JSONL yang akan digunakan untuk *validation*. Pada proses *training*, nilai *epoch* yang digunakan oleh seluruh model adalah 5.

### 3.4.3. Gemini oleh Google

Proses *training* Gemini dilakukan melalui platform VertexAI. *Fine-tuning* dilakukan dengan *Graphical User Interface*. Metode yang digunakan pada model ini adalah *supervised tuning*. Pada tampilan ini juga peneliti memilih *base model*, mengunggah data JSONL yang akan

di *train*, dan menggugah data JSONL yang akan digunakan untuk *validation*. Pada proses *training*, nilai *epoch* yang digunakan oleh seluruh model adalah 5.

### 3.5. *Testing and Evaluation*

Proses *testing* pada ketiga model dilakukan melalui Jupyter Notebook berbasis bahasa Python. *Testing* dilakukan dengan melakukan iterasi pada *batch data test*, mengakumulasi prediksi label, dan menghitung performa dari matriks.

Untuk GPT dan Gemini, *batch testing* dilakukan dengan mengakses *Library* dan API token dari masing-masing platform model tersebut.

Dalam proses evaluasi ini data prediksi hasil dari *test* dan label akan dikirimkan ke *function matrix* yang akan melakukan perhitungan sesuai dengan rumus untuk masing-masing penilaian.

