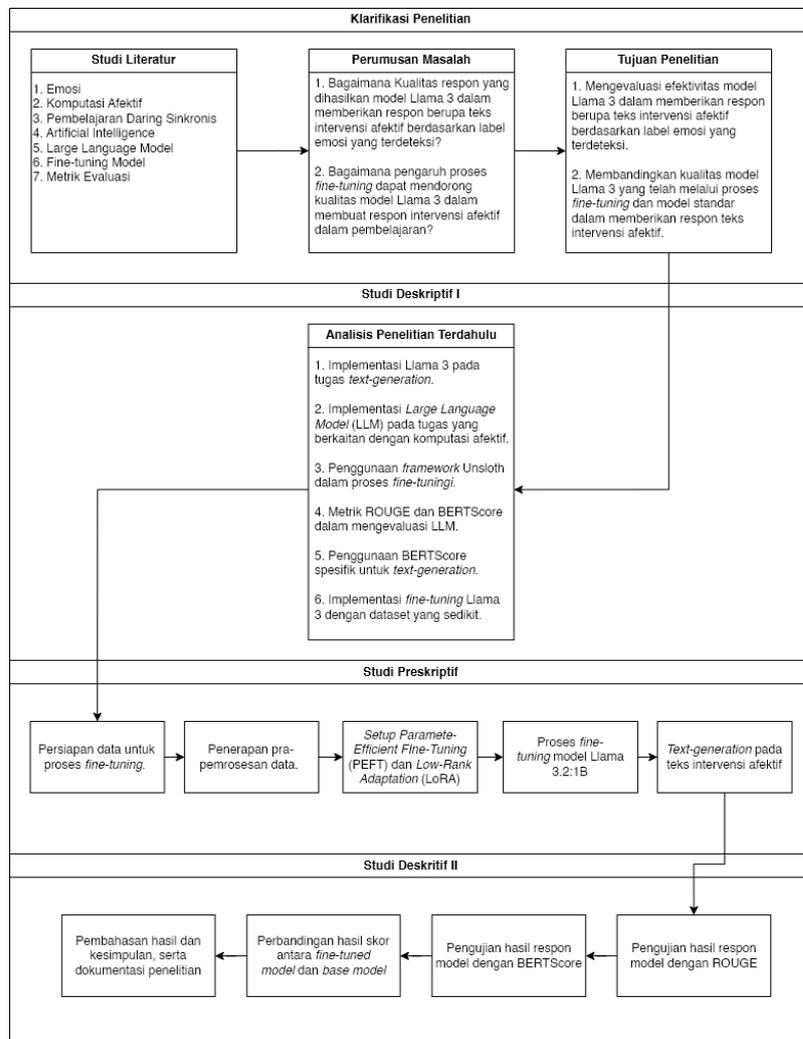


BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini mengadopsi model Design Research Methodology (DRM) yang berguna dalam menyediakan kerangka kerja yang terstruktur dan sistematis yang membimbing para peneliti melalui berbagai tahapan penelitian desain. Kerangka kerja ini membantu meningkatkan efektivitas dan efisiensi penelitian dengan memberikan dukungan untuk perencanaan, pelaksanaan, dan evaluasi secara metodis. sebagai kerangka yang memberikan langkah-langkah secara terperinci dan sistematis untuk mencapai hasil yang dibutuhkan (Ebneyamini, 2022). Gambar 3.1 memberikan gambaran penelitian yang akan dilakukan.



Gambar 3.1 Design Research Methodology (DRM)

3.2 Klarifikasi Penelitian

Pada tahap klarifikasi penelitian, dilakukan analisis literatur secara sistematis melalui berbagai referensi terkait emosi dalam pembelajaran, dilanjutkan dengan pembahasan mengenai komputasi afektif melalui model *Attention, Relevance, Confidence*, dan *Satisfaction* (ARCS), dan penguatan positif negatif. Studi selanjutnya menganalisis terkait cakupan *Artificial Intelligence* yang mencakup *Machine Learning* (ML), *Natural Language Processing* (NLP). Selanjutnya, analisis mendalami pembahasan *Large Language Model* (LLM), dengan model Llama 3 sebagai model dasar yang digunakan dalam penelitian untuk memberikan respon berupa teks intervensi afektif pada pembelajaran daring secara sinkronis, aplikasi rekognisi emosi Emodu yang telah dikembangkan pada penelitian sebelumnya (Utama, 2024). Terakhir, pembahasan menganalisa terkait metrik evaluasi yang digunakan dalam penelitian ini, berupa ROUGE dan BERTScore. Kajian literatur dari berbagai referensi untuk penelitian dibahas dalam BAB II. Adapun perumusan masalah dan tujuan penelitian dirancang berdasarkan analisis literatur dibahas pada BAB I.

3.3 Studi Deskriptif I

Tahap ini menjelaskan permasalahan serta arah penelitian yang dijabarkan pada studi literatur di tahapan sebelumnya. Tahapan ini bertujuan untuk merumuskan state-of-the-art penelitian ini dari penelitian-penelitian terdahulu, sebagai acuan kebaruan dari metode, proses, hingga hasil yang digagas untuk dilaksanakan pada penelitian ini. Adapun pembahasan meliputi implementasi model Llama 3 pada tugas *text-generation*, pengimplementasian *Large Language Model* (LLM) pada tugas yang berkaitan dengan komputasi afektif, penggunaan Unsloth sebagai *framework* pada proses *fine-tuning*, penggunaan metrik ROUGE dan BERTScore dalam mengevaluasi hasil respon LLM, penggunaan BERTScore yang spesifik untuk tugas *text-generation*, dan implementasi *fine-tuning* pada model Llama 3 dengan jumlah data yang sedikit.

3.4 Studi Preskriptif

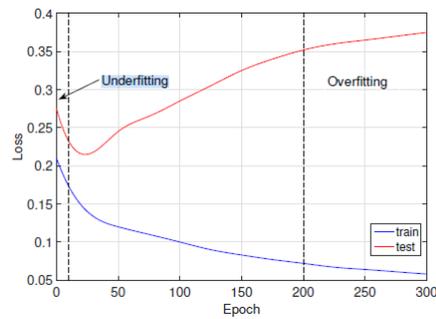
Pada tahap ini, metode-metode yang akan digunakan untuk menyelesaikan permasalahan dalam penelitian akan dilakukan. Tahap-tahap dalam studi preskriptif bertujuan untuk mengimplementasikan sistem respon berbasis teks intervensi melalui proses *fine-tuning* model Llama 3 sebagai model respon teks intervensi. Tahapan dimulai dari persiapan data untuk proses *fine-tuning*, penerapan pra-pemrosesan data untuk meningkatkan kualitas model, *setup* terhadap parameter dari *Low-Rank Adaptation* (LoRA) dan pemanfaatan *Parameter-Efficient Fine-Tuning* (LoRA), hingga proses *fine-tuning* model dilakukan.

3.4.1 Pengembangan Model Llama 3

Pengembangan model dilakukan dalam beberapa tahap, dimulai dari persiapan data. Data dipersiapkan dengan mengunggah ke dalam platform HuggingFace sebagai layanan *repository* untuk menampung data pada *cloud*. Selanjutnya, pra-pemrosesan data dilakukan untuk meningkatkan kualitas data, dan menyesuaikan data agar dapat dipahami oleh model dalam proses *fine-tuning*. *Large Language Model* (LLM) GPT-4o digunakan sebagai untuk melakukan proses *data augmentation*. Penelitian Dai dkk. (2023) menunjukkan bahwa LLM memiliki kemampuan yang baik dalam membuat data sintesis menggunakan data asli sebagai referensi. Data akan melalui proses augmentasi hingga jumlah sampel mencapai minimal 1000 data untuk mencapai hasil performa yang baik (Tomanek, Cai, dan Venugopalan, 2023).

Sebelum memulai proses *fine-tuning*, *parameter* LoRA yang disediakan oleh Unsloth diaplikasikan untuk mengurangi penggunaan memori dan mempercepat proses *fine-tuning* menggunakan PEFT dengan hanya melatih lapisan teratas pada model (Sepulveda dkk., 2024). Pengaplikasian LoRA dan PEFT menyesuaikan dengan *parameter default* yang disediakan pada Unsloth, tanpa mengubah nilai *parameter*. Beberapa bagian yang diubah berfokus pada *parameter trainer* untuk memvisualisasikan proses *fine-tuning*. Setelah *parameter* dipersiapkan, model akan melalui proses *fine-tuning* dengan jumlah epoch sebanyak satu kali, yang disesuaikan dengan jumlah data yang sedikit agar model tidak mengalami indikasi *overfit*, ketika model belajar dengan baik dari data *training* namun tidak dengan

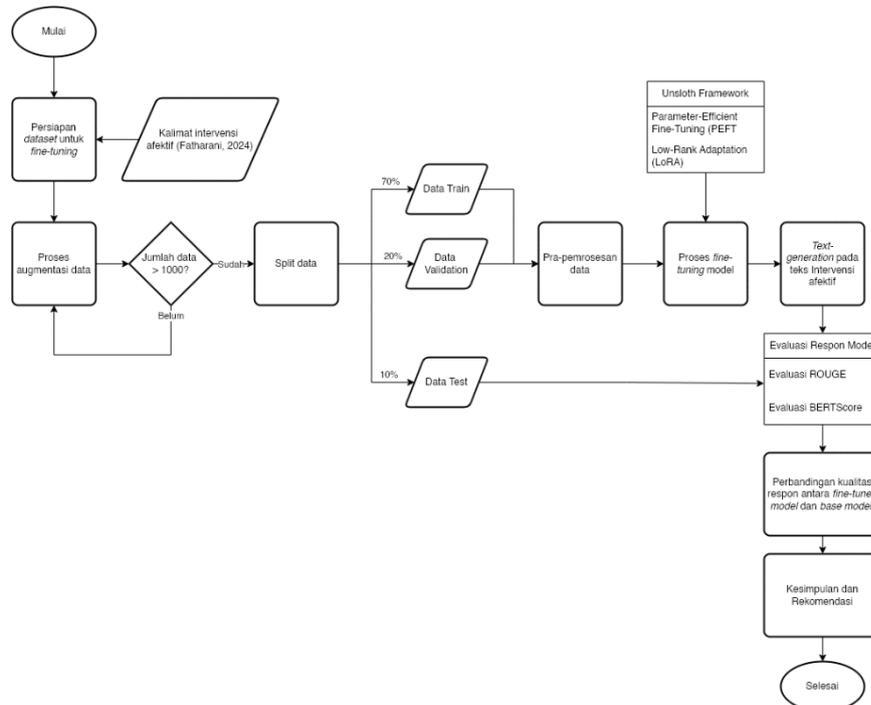
validation (Cogswell, Ahmed, Girshick, Zitnick, dan Batra, 2015), serta gejala *underfit*, ketika tidak terdapat penurunan *training loss* tidak diiringi dengan penurunan *validation loss* (Zhang, Zhang dan Jiang, 2019). Gambar 3.2 memberikan gambaran mengenai *overfitting* dan *underfitting*.



Gambar 3.2 Indikasi *Overfitting* dan *Underfitting*

Sumber: Zhang, Zhang, dan Jiang (2019)

Model kemudian dilatih dengan memperhatikan nilai *loss* untuk mengamati terkait indikasi *overfit*, atau *underfit* (Parthasarathy, dkk., 2024). Gambar 3.3 memberikan rincian terkait seluruh proses ini dalam sebuah diagram alir.



Gambar 3.3 Alur Kerja Implementasi Llama 3

3.4.2 Pengumpulan Data

Pada tahap ini, *dataset* utama yang digunakan bersumber pada penelitian Fatharani (2024), dengan total 160 baris data berisikan kategori emosi, label model sesuai teori ARCS, kategori penguatan, serta kalimat penguatan yang telah divalidasi oleh ahli. Tabel 3.1 menampilkan deskripsi data dari *file csv dataset* yang digunakan pada penelitian ini.

Tabel 3.1 Deskripsi *Dataset*

Kolom	Deskripsi
<i>emotion</i>	Label emosi yang terbagi kedalam 4 kategori emosi negatif berdasarkan Ekman (1997), yaitu <i>angry</i> , <i>sad</i> , <i>disgust</i> , dan <i>fear</i> .
<i>arcs_model</i>	Model pendekatan ARCS yang terdiri dari komponen <i>Attention</i> , <i>Relevance</i> , <i>Confidence</i> , dan <i>Satisfaction</i> yang dikemukakan Keller (1987).
<i>reinforcement_type</i>	Jenis kategorisasi kalimat penguatan yang terbagi kedalam dua kategori, positif dan negatif.
<i>sentence</i>	Kalimat intervensi afektif yang disusun berdasarkan kategori emosi, model ARCS, dan jenis penguatan, disertai dengan emoji di akhir kalimat.

3.4.3 Pra-pemrosesan Data

Pada tahap ini, data melalui tahap augmentasi untuk meningkatkan jumlah, dan pemrosesan menjadi sebuah format yang dapat dipahami oleh model Llama 3. Augmentasi dilakukan menggunakan *Large Language Model* (LLM) dengan kemampuan yang dapat membuat data sintesis dengan kualitas yang hampir sama melalui *prompt* atau perintah yang sesuai (Woźniak dan Kocoń, 2023; Whitehouse, Choudhury, dan Aji, 2023). Proses augmentasi dilakukan menggunakan metode

random insertion, *random deletion*, *synonym replacement* (Wei dan Zou, 2019), dan *synthetic generation* (Dhruva, Bhat, Rangayyan, dan Preethi, 2024) menggunakan LLM dengan menggunakan teknik persona, yaitu memberikan instruksi dengan membuat sebuah identitas pada LLM tersebut (White dkk., 2023). Perincian struktur *prompt* berbasis teknik persona terdapat pada Tabel 3.2.

Tabel 3.2 Struktur *Prompt*

Struktur	Contoh Pemahaman Kontekstual
instruksi	“Asumsikan diri kamu sebagai profesi X. Berikan <i>output</i> yang akan diberikan oleh seorang X.”
<i>input</i>	“Contoh <i>input</i> oleh X: X memberikan Y terhadap Z”

Setelah melalui proses augmentasi, *dataset* dibagi menjadi tiga bagian, yaitu data *train*, data *validation*, dan data *test*, dengan proporsi pembagian 70%, 20%, dan 10%. Merujuk pada penelitian Vrigazova (2021), proporsi ideal untuk proses *training* adalah 70% untuk data *train* dan 30% untuk data *validation*. Namun, dalam penelitian ini, proporsi data *validation* diperkecil menjadi 20%, sementara 10% sisanya digunakan sebagai data *test* untuk menguji kualitas model menggunakan metrik ROUGE dan BERTScore. Pembagian ini dilakukan untuk memberikan porsi data *train* yang lebih besar sehingga model memiliki peluang lebih baik dalam memahami pola data (Xu dan Goodacre, 2018).

3.4.4 Setup PEFT dan LoRA

Pada tahap ini, penerapan *Parameter-Efficient Fine-Tuning* (PEFT) dan *Low-Rank Adaptation* (LoRA) dilakukan sebelum memulai proses *fine-tuning* model. Penerapan PEFT dilakukan untuk mempercepat proses *fine-tuning* dengan penggunaan memori yang lebih sedikit, namun tetap mencapai proses dengan hasil yang mirip pada proses *full fine-tuning* (Xu, Xie, Qin, Tao, dan Wang, 2023). Pada *framework* Unsloth, setup PEFT telah disediakan secara *default* sehingga tidak perlu penyesuaian lebih lanjut.

Selanjutnya, penerapan LoRA digunakan untuk mempersingkat proses *training* dengan hanya melatih sekitar 10% dari total *parameter* pada model, dan juga mengurangi kebutuhan memori pada *Graphic Processing Unit* (GPU) hingga 3 kali lipat (Hu dkk., 2021). Unsloth telah mengimplementasikan beberapa *parameter* LoRA secara *default* sehingga tidak perlu ada penyesuaian lebih lanjut, terkecuali pada penelitian yang berfokus terhadap pengaruh LoRA *hyperparameter* terhadap hasil *fine-tuning*.

3.4.5 Proses Fine-tuning Model

Proses *fine-tuning* akan dilakukan menggunakan model *pre-trained* berbasis Llama 3.2 varian 1B *parameter* dengan data yang telah melalui proses pra-pemrosesan. Model yang digunakan merupakan turunan dari model sebelumnya, Llama 3.1 dengan pemangkasan *parameter* yang ada sehingga sesuai dengan fokus penelitian yang dilakukan, proses *fine-tuning* dilakukan untuk menyesuaikan model tersebut dengan data yang dimiliki. Proses *fine-tuning* dilakukan menggunakan Unsloth sebagai *framework* (Kumar, 2024; Sepulveda dkk., 2024) yang telah menyediakan *pipeline* pada proses *fine-tuning*.

3.4.6 Respon Teks Intervensi Afektif

Setelah model melalui tahap *fine-tuning*, tahap terakhir yaitu melakukan proses *text-generation* menggunakan model yang telah di *fine-tune* dan model standar untuk dilakukan proses perbandingan antara kualitas respon yang dihasilkan dengan data referensi yang telah disiapkan. Proses ini akan menggunakan *library* dari Ollama, sebuah layanan *open-source* yang dapat digunakan untuk melakukan proses *deployment* dari *Large Language Model* (LLM) (Liu dkk., 2024). Ollama memberikan fasilitas yang memudahkan akses terhadap LLM yang telah beredar secara terbuka, maupun LLM hasil *fine-tuning* terhadap suatu domain spesifik yang dilakukan oleh pengguna (Gruber dan Weber, 2024). Penerapan Ollama akan diiringi oleh Langchain, sebuah *framework* yang dapat mengintegrasikan penggunaan LLM dengan *local environment* pengguna (Pandya dan Holia, 2023). Langchain berfungsi sebagai alat untuk berkomunikasi dengan model LLM melalui proses *information retrieval* menggunakan algoritma *Natural Language Processing*

(NLP) (Priya dkk., 2024). Menggunakan kedua *framework* tersebut, respon akan dibuat berdasarkan label emosi, kategori model ARCS, dan jenis penguatan pada data referensi agar teks intervensi afektif yang dihasilkan selaras dengan teks orisinal sehingga dapat dibandingkan dengan sesuai.

3.5 Studi Deskriptif II

Tahap ini berfokus pada berbagai proses pengujian yang bertujuan untuk mengukur performa model dalam membuat respon. Setelah melalui seluruh tahapan *fine-tuning* dan implementasi model, hasil respon model kemudian diuji dengan menggunakan metrik ROUGE dan BERTScore dengan membandingkan hasil respon dengan data referensi yang telah dipersiapkan. Pengujian pertama dilakukan dengan menguji kualitas respon model Llama 3 menggunakan metrik evaluasi ROUGE. Pengujian kedua dilakukan untuk menguji performa model secara lebih lanjut menggunakan instrumen BERTScore. Model yang telah melalui proses *fine-tuning* kemudian dibandingkan dengan model dasar untuk melihat perbedaan pada kualitas respon yang dihasilkan. Pada bagian ini juga dipaparkan kesimpulan serta rekomendasi yang didapat dari penelitian, serta disajikan kesimpulan yang bertujuan untuk menjawab pertanyaan dalam rumusan masalah. Selain itu, terdapat saran bagi peneliti berikutnya guna meningkatkan kualitas penelitian ini di masa mendatang. Penyusunan dokumen skripsi juga menjadi tahap akhir yang disajikan dalam bagian ini.

3.5.1 Evaluasi Respon Model dengan ROUGE

Metrik ROUGE yang dipilih untuk mengevaluasi performa Llama 3 adalah ROUGE-N, dan ROUGE-L. ROUGE-N digunakan untuk menghitung tingkat kecocokan '*n-gram*' antara teks yang dihasilkan oleh model dan teks referensi yang telah disiapkan. *N-gram* merujuk pada serangkaian kata yang terdapat dalam sebuah kalimat. Evaluasi ROUGE pertama akan menggunakan ROUGE-N dimana *N* mewakili jumlah *gram* atau *n-gram* yang digunakan. Sebagai contoh, *unigram* (1-*gram*) terdiri dari satu kata, dan *bigram* (2-*gram*) terdiri dari 2 kata yang berurutan. Perhitungan ROUGE-N (Lin, 2004) dirincikan pada persamaan berikut.

$$= \left[\frac{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)} \right] \quad (1)$$

Dimana n merepresentasikan panjang dari n -gram, gram_n , dan $\text{countmatch}(\text{gram}_n)$ adalah jumlah maksimum n -gram yang muncul bersamaan dalam ringkasan kandidat dan kumpulan ringkasan referensi. Selanjutnya, perhitungan ROUGE-L akan menangkap *Longest Common Subsequence* (LCS), yaitu urutan elemen terpanjang yang muncul dalam satu teks meskipun tidak berurutan secara langsung (Lin, 2004). Sebagai contoh, terdapat dua kalimat yang akan digunakan pada perhitungan LCS, kalimat pertama sebagai referensi yaitu “*the cat is on the mat*”, dan kalimat kedua sebagai kandidat yaitu “*the cat sat on the mat*”. Berdasarkan kedua kalimat tersebut, panjang LCS yang ditemukan sejumlah 4 yaitu “*the cat on the mat*”, yang kemudian dihitung secara *precision* dan *recall* dengan perhitungan sebagai berikut.

$$P = \frac{\text{Panjang LCS}}{\text{Panjang Kandidat}} \quad (2)$$

$$R = \frac{\text{Panjang LCS}}{\text{Panjang Referensi}} \quad (3)$$

Pada penelitian ini, ROUGE-1 dan ROUGE-2 akan digunakan dengan fungsi mengukur tingkat kecocokan *unigram* dan *bigram* antara hasil *output* model dan referensi yang dibuat. Selanjutnya, metrik ROUGE-L akan digunakan untuk membandingkan LCS antara *output* model dan referensi. Metrik ROUGE yang digunakan menggunakan *library* akan menghasilkan nilai ROUGE-1, ROUGE-2, serta ROUGE-L.

3.5.2 Evaluasi Respon Model dengan BERTScore

BERTscore digunakan untuk menguji kemampuan model dalam menghasilkan teks dengan melakukan perbandingan kemiripan secara semantik. BERTscore bekerja dengan menghitung skor terkait kemiripan setiap token dalam teks *output* dengan token dalam teks referensi, dimana perhitungan kemiripan token

menggunakan konsep *semantic similarity* (Tang, Yousuf, dan Jin, 2024). Luaran dari pengukuran BERTscore berupa *precision*, *recall*, dan *F1-score* (Zhang, dkk., 2019).

1. *Precision*

Precision dalam BERTScore mengukur sejauh mana kata-kata dalam kalimat kandidat memiliki makna yang sesuai atau relevan dengan kalimat referensi. Berikut penggambaran persamaan matematika dari *precision*:

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^T \hat{x}_j \quad (4)$$

Keterangan:

P_{BERT} = *Precision BERTscore*

x = Token referensi

\hat{x} = Token *output*

x_i = Urutan vektor x

\hat{x}_j = Urutan vektor \hat{x}

$\sum_{\hat{x}_j \in \hat{x}}$ = Jumlah \hat{x}_j dalam \hat{x}

$\max_{x_i \in x}$ = Nilai maksimum x_i dalam x

$x_i^T \hat{x}_j$ = *cosine similarity* dari x dan \hat{x}

2. *Recall*

Recall dalam BERTScore mengukur sejauh mana kata-kata dalam kalimat referensi dapat ditemukan dalam kalimat kandidat dengan makna yang sesuai. Berikut penggambaran persamaan matematika dari *recall*:

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j \quad (5)$$

Keterangan:

R_{BERT} = *Recall BERTscore*

x = Token referensi

\hat{x} = Token *output*

x_i = Urutan vektor x

\hat{x}_j = Urutan vektor \hat{x}

$\sum_{x_i \in x}$ = Jumlah x_i dalam x

$\max_{\hat{x}_j \in \hat{x}}$ = Nilai maksimum \hat{x}_j dalam \hat{x}

$x_i^\top \hat{x}_j$ = *cosine similarity* dari x dan \hat{x}

3. *F1 score*

F1 score mengambil rata-rata dari gabungan nilai *precision* dan *recall*. Metrik ini berguna untuk membandingkan kinerja model dengan melihat nilai rata-rata dari kedua metrik akurasi sebelumnya sehingga diahilkan sebuah skor yang dapat memberikan hasil lebih jelas. Penggambaran persamaan dari perhitungan *F1 score* dapat dilihat pada rumus matematika berikut.

$$F_{BERT} = 2 * \frac{PBERT * RBERT}{PBERT + RBERT} \quad (6)$$

Keterangan:

F_{BERT} = *F1-score BERT*

3.5.3 Perbandingan *Fine-tuned Model* dan *Base Model*

Dalam melakukan perbandingan kualitas model, pengujian hasil *output* akan dilakukan pada *fine-tuned model* dan *base model* secara terpisah. Model dasar yang digunakan yaitu Llama 3.2 varian 1B parameter. Perbandingan akan dilakukan dengan melihat perbedaan skor rata-rata atau *mean* yang dihasilkan pada kedua metrik evaluasi, ROUGE-1, ROUGE-2, dan ROUGE-L pada ROUGE, dan *precision*, *recall*, dan *F1 score* pada BERTScore, dengan selisih nilai dari kedua metrik evaluasi tersebut berfungsi sebagai pengambilan keputusan terkait mana model yang dapat memberikan hasil terbaik dalam melakukan tugas pembuatan teks intervensi afektif.

3.6 Instrumen Penelitian

Instrumen penelitian dapat diartikan sebagai sarana yang digunakan untuk menganalisis data secara sistematis dan objektif (Nasution, 2016). Dalam konteks penelitian ini, instrumen yang digunakan yaitu ROUGE dan BERTScore, bertujuan untuk mengukur kesuksesan model dalam membuat teks intervensi afektif yang memiliki kemiripan secara kata maupun semantik dengan data uji yang telah dipersiapkan.

3.7 Alat dan Bahan Penelitian

Penelitian yang diusung akan mengimplementasikan *Large Language Model* (LLM) dengan basis model Llama 3.2 varian 1B yang merupakan LLM yang bersifat *open-source* yang dikembangkan oleh Facebook (Dubey dkk., 2024), serta *environment code editor* pada lingkup *local* yang akan digunakan yaitu Visual Studio Code, dan proses *fine-tuning* akan dilakukan menggunakan *environment* berbasis *cloud*, Google Colab. Spesifikasi perangkat yang akan digunakan dalam pengembangan program pada penelitian ini, dijabarkan pada Tabel 3.3.

Tabel 3.3 Spesifikasi Perangkat

Jenis Perangkat	Tipe
Perangkat Keras Penulisan Kode	
<i>Processor</i>	10 th Gen Intel Core I5-1035G1
RAM	12 GB DDR4
<i>Storage</i>	SSD 512GB
GPU	NVIDIA GeForce MX330
Perangkat Keras <i>Fine-tuning</i> Model	
GPU	NVIDIA Tesla T4
Perangkat Lunak	
Sistem Operasi	Windows 11
Lingkup Pengembangan	Google Colab, Visual Studio Code

3.7.1 Library Python

Adapun bahan yang akan digunakan, seperti *library* pada bahasa pemrograman Python dan komponen pada *framework* Unsloth, beserta *module* yang akan digunakan dalam pengimplementasian model Llama 3 dijabarkan pada Tabel 3.4.

Tabel 3.4 Daftar *Library Python*

Nama Library	Versi	Kegunaan
<i>unsloth</i>	2025.1.5	<i>Library</i> yang disediakan melalui <i>framework</i> Unsloth untuk proses <i>fine-tuning</i> yang efektif dalam menghemat memori serta mempercepat proses.
<i>torch</i>	2.5.1+cu121	<i>Library</i> yang disediakan oleh HuggingFace yang menyediakan <i>pre-tuned</i> model seperti Llama 3.2 varian 1B.
<i>trl</i>	0.13.0	<i>trl</i> yang merupakan singkatan dari <i>transformer reinforcement learning</i> , bertujuan untuk melakukan proses <i>training</i> melalui konsep <i>supervised fine-tuning trainer</i> .
<i>transformers</i>	4.47.1	<i>Library</i> yang berisi model <i>pre-tuned</i> dan modul untuk memproses <i>hyperparameter</i> pada proses <i>training</i> .
<i>ollama</i>	0.5.4	<i>Library</i> yang bertujuan sebagai <i>platform</i> untuk menampung model yang telah melalui proses <i>fine-tuning</i> , sekaligus memberikan akses terhadap model tersebut.
<i>evaluate</i>	0.4.2	<i>Library</i> yang disediakan oleh HuggingFace, dengan modul yang bertujuan untuk melakukan evaluasi terhadap model, seperti ROUGE dan BERTScore.

<i>pandas</i>	2.2.1	<i>Library</i> yang berguna untuk memproses data, termasuk memanipulasi data dalam jumlah besar.
<i>langchain-core</i>	0.3.29	<i>Library</i> ini bertujuan untuk membantu proses pembuatan aplikasi yang mengintegrasikan <i>Large Language Model</i> (LLM) dalam suatu <i>environment</i> .
<i>langchain-ollama</i>	0.2.2	Ekstensi dari LangChain untuk mendukung penggunaan Ollama pada <i>environment</i> .
<i>scikit-learn</i>	1.4.0	Mendukung proses <i>splitting data</i> melalui salah satu modulnya, yaitu <i>train_test_split</i> .

3.8 Analisis Data

Pada penelitian ini, analisis data dilakukan berdasarkan hasil nilai ROUGE-1, ROUGE-2, ROUGE-L, dan BERTScore pada *fine-tuned model* dan *base model* Llama 3.2:1B dalam memberikan respon berupa teks intervensi afektif yang sesuai. Keberhasilan diukur berdasarkan nilai ROUGE-1, ROUGE-2, dan ROUGE-L dari metrik ROUGE, *precision*, *recall*, dan *F1-score* dari metrik BERTScore.