BAB 2
LANDASAN TEORI

Penelitian analisis sentimen terhadap penanggulangan bencana banjir bandang di Pulau Sumatra ini disusun berdasarkan berbagai teori yang mendukung penelitian. Berbagai teori yang digunakan dalam penelitian ini diuraikan sebagai berikut.

2.1 Banjir Bandang

Banjir Bandang merupakan salah satu jenis bencana hidrometeorologi yang terjadi ketika aliran air yang deras membawa berbagai material, seperti batu, lumpur, pohon, dan puing-puing bangunan [21]. Bencana ini umumnya dipicu oleh berbagai faktor, antara lain curah hujan ekstrem, kondisi geografis berupa lereng yang curam, serta aktivitas penggundulan hutan dan alih fungsi area resapan air yang menyebabkan berkurangnya kapasitas infiltrasi tanah [1, 21]. Banjir bandang tidak hanya menyebabkan kerusakan fisik pada permukiman dan infrastruktur, tetapi juga mengganggu aktivitas sosial, menurunkan kualitas lingkungan, serta menimbulkan risiko terhadap keselamatan masyarakat.

2.2 Siklus Penanggulangan Bencana

Siklus Penanggulangan Bencana (*Disaster Management Cycle*) merupakan kerangka yang menggambarkan tahapan pengelolaan bencana [12]. Pada penelitian ini, tahapan dalam siklus tersebut digunakan sebagai dasar penentuan kategori aspek yang terdiri atas mitigasi, tanggap darurat, dan pemulihan. Definisi dan cakupan masing-masing aspek dijelaskan pada bagian berikut.

1. Mitigasi

   Mitigasi merupakan aspek yang membahas upaya pencegahan dan pengurangan risiko bencana, termasuk faktor-faktor yang dapat menyebabkan terjadinya bencana serta pengelolaan lingkungan yang berkaitan dengan risiko kebencanaan [12]. Aspek ini mencakup pembahasan mengenai penyebab bencana, kerusakan lingkungan, pembalakan liar, alih fungsi lahan, sedimentasi sungai, konservasi, reboisasi, kebijakan perizinan yang berkaitan

dengan lingkungan, serta berbagai upaya pencegahan dan pengurangan risiko bencana.

2. Tanggap Darurat

   Tanggap darurat merupakan aspek yang membahas berbagai respons yang dilakukan selama bencana berlangsung oleh pemerintah, lembaga penanggulangan bencana, maupun masyarakat [12]. Aspek ini mencakup kegiatan evakuasi korban, operasi pencarian dan penyelamatan (SAR), bantuan logistik, distribusi makanan dan air bersih, bantuan kesehatan darurat, pengungsian sementara, penanganan korban bencana, serta berbagai bentuk respons terhadap kejadian bencana.

3. Pemulihan

   Pemulihan merupakan aspek yang membahas upaya mengembalikan kondisi sosial, ekonomi, pendidikan, infrastruktur, dan kehidupan masyarakat setelah bencana terjadi [12]. Aspek ini mencakup relokasi korban, pembangunan hunian tetap, perbaikan rumah dan fasilitas umum, rehabilitasi dan rekonstruksi pascabencana, pemulihan layanan publik, pemulihan pendidikan, pemulihan ekonomi masyarakat, serta upaya pemulihan kondisi wilayah terdampak bencana.

2.3 X

X atau yang sebelumnya bernama Twitter merupakan platform sosial media berbasis pesan singkat (*microblogging*) yang diluncurkan pada tahun 2006 oleh Jack Dorsey [22]. X menyediakan berbagai fitur utama seperti unggahan (*post*), *repost*, tanda pagar (*hashtag*), balasan (*reply*), serta fitur topik tren yang membantu pengguna mengikuti isu yang sedang banyak dibicarakan.

2.4 Youtube

YouTube merupakan platform media sosial yang diperkenalkan pada tahun 2005 oleh Chad Hurley, Steve Chen, dan Jawed Karim [23]. Melalui platform ini, pengguna dapat mengakses berbagai jenis konten video sekaligus mengunggah karya mereka sendiri untuk ditonton oleh pengguna lain. Pengguna juga dapat memberikan tanggapan terhadap video, tanda menyukai video (*like*), serta membagikan video kepada pengguna lain.

2.5 Analisis Sentimen

Analisis sentimen merupakan salah satu metode komputasional yang dilakukan untuk mengenali dan menentukan kecenderungan sentimen yang terdapat dalam suatu teks berdasarkan polaritasnya [8, 10]. Dalam bidang kebencanaan, analisis sentimen dapat dimanfaatkan untuk mengkaji respons masyarakat terhadap berbagai upaya penanganan bencana yang telah dilakukan. Analisis ini dapat digunakan untuk memahami bagaimana masyarakat menilai upaya penanganan bencana, sehingga masukan yang tercermin dalam opini publik dapat dimanfaatkan dalam penyusunan strategi penanggulangan bencana pada masa mendatang [11].

2.6 Analisis Sentimen Berbasis Aspek

Analisis sentimen berbasis aspek (ABSA) merupakan pengembangan dari analisis sentimen yang berfokus pada identifikasi sentimen untuk setiap aspek yang muncul dalam teks. Melalui pendekatan ini, suatu opini dapat dianalisis berdasarkan aspek tertentu sehingga hasil analisis dapat menunjukkan kecenderungan sentimen pada setiap aspek yang terdapat dalam teks [13].

Dalam literatur, ABSA mencakup beberapa elemen utama, yaitu *aspect term*, *aspect category*, *opinion term*, dan *sentiment polarity*. *Aspect term* merujuk pada istilah aspek yang muncul secara eksplisit dalam teks, sedangkan *aspect category* merupakan kategori aspek yang telah ditentukan sebelumnya. Sementara itu, *opinion term* menunjukkan kata atau frasa yang mengandung opini, sedangkan *sentiment polarity* menunjukkan orientasi sentimen seperti positif, negatif, atau netral [13].

Salah satu bentuk penerapan ABSA adalah *Aspect Category Sentiment Analysis* (ACSA), yaitu pendekatan yang berfokus pada identifikasi kategori aspek dan penentuan sentimen pada kategori tersebut [14]. Cai et al. [15] menjelaskan bahwa ACSA melibatkan dua proses utama, yaitu *aspect category detection* (deteksi kategori aspek) dan *category-oriented sentiment classification* (klasifikasi sentimen berdasarkan kategori aspek).

2.7 Data Preprocessing

Data preprocessing dapat didefinisikan sebagai langkah yang diterapkan untuk menyesuaikan dan membersihkan data teks sebelum digunakan pada pelatihan model atau analisis. Data teks yang berasal dari berbagai sumber sering

kali masih mengandung karakter khusus, kesalahan penulisan, kata tidak baku, maupun informasi yang tidak berkaitan dengan topik pembahasan [24]. Oleh karena itu, *data preprocessing* digunakan untuk mengurangi unsur-unsur tersebut sehingga data menjadi lebih sesuai untuk proses analisis dan pemodelan. Beberapa tahapan pra-premrosesan data yang dilakukan dijelaskan sebagai berikut.

1. Cleaning

   Cleaning merupakan tahapan yang dilakukan untuk mengurangi berbagai unsur dalam teks yang tidak memberikan informasi penting bagi proses analisis. Unsur yang dimaksud dapat berupa alamat situs *web*, simbol tertentu, karakter nonalfabet, maupun tanda baca yang tidak memiliki keterkaitan dengan informasi utama dalam teks [25].

2. Case Folding

   Case folding merupakan teknik untuk menormalisasi setiap kata pada suatu teks dengan mengubah seluruh teks ke dalam format huruf kecil. Penerapan *case folding* bertujuan untuk mencegah adanya perbedaan arti dan makna kata yang muncul akibat penggunaan huruf kapital pada suatu kata [25].

3. Normalisasi

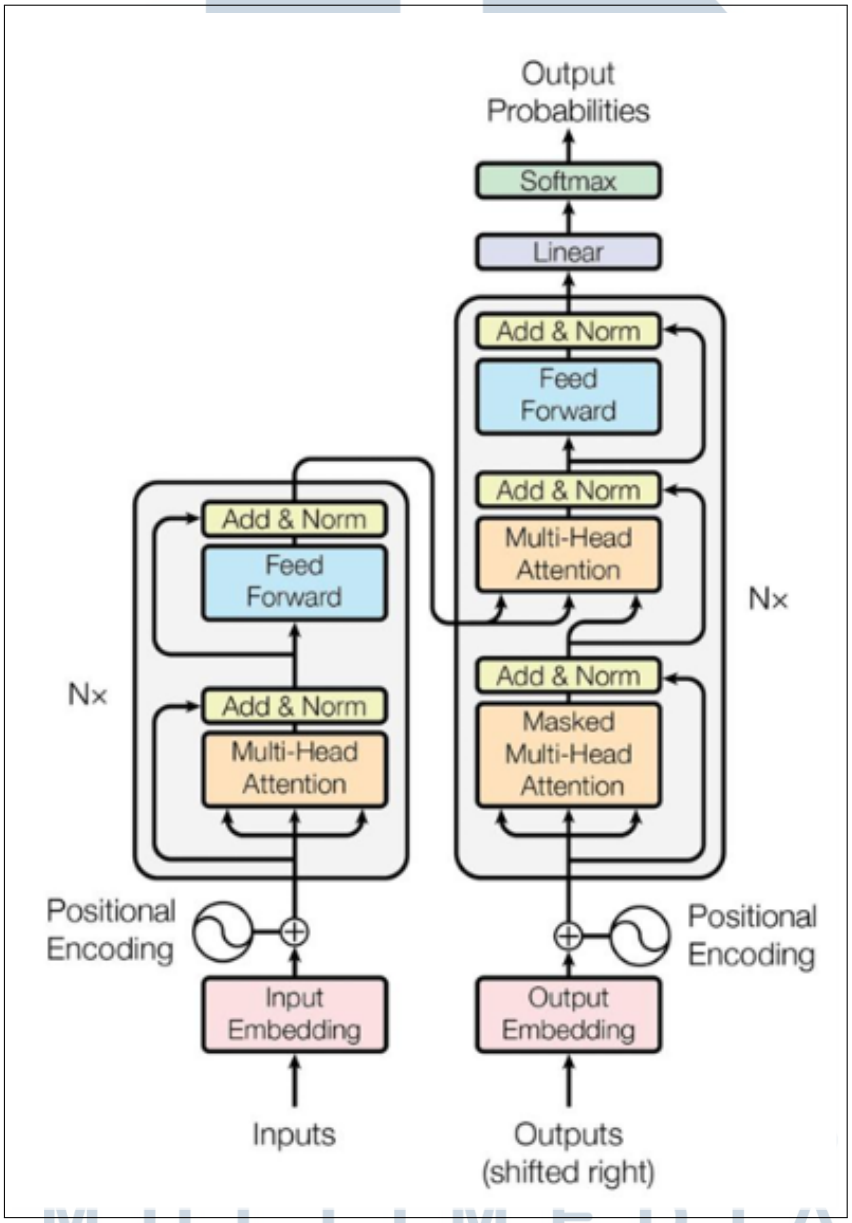
   Normalisasi merupakan tahapan yang berfokus pada penyeragaman bentuk kata dalam teks. Proses ini dilakukan dengan menyesuaikan bentuk penulisan yang beragam ke bentuk yang sesuai dengan kaidah Bahasa Indonesia [26].

2.8 Transformer

Transformer merupakan arsitektur pemrosesan urutan yang membentuk representasi token dengan menggunakan mekanisme *attention*. Melalui mekanisme tersebut, representasi pada suatu posisi dapat dibentuk berdasarkan hubungan posisi tersebut dengan posisi-posisi lain yang terdapat dalam urutan yang sama. Proses ini berbeda dengan model rekuren yang membawa informasi secara bertahap dari satu posisi menuju posisi berikutnya. Ketergantungan antartoken pada Transformer tidak dihitung berdasarkan perpindahan keadaan tersembunyi secara berurutan, sehingga perhitungan pada beberapa posisi dapat dilakukan secara paralel [27].

Arsitektur Transformer pada bentuk awalnya terdiri atas *encoder* dan *decoder*. *Encoder* menerima urutan masukan dan menghasilkan representasi kontekstual, sedangkan *decoder* menggunakan representasi tersebut untuk

membentuk urutan keluaran. Kedua bagian tersebut tersusun atas beberapa lapisan yang mempunyai komponen pemrosesan masing-masing. BERT menggunakan tumpukan *encoder* tanpa menyertakan bagian *decoder* [28]. Susunan umum dari arsitektur Transformer diperlihatkan pada Gambar 2.1.



Gambar 2.1. Susunan umum arsitektur Transformer

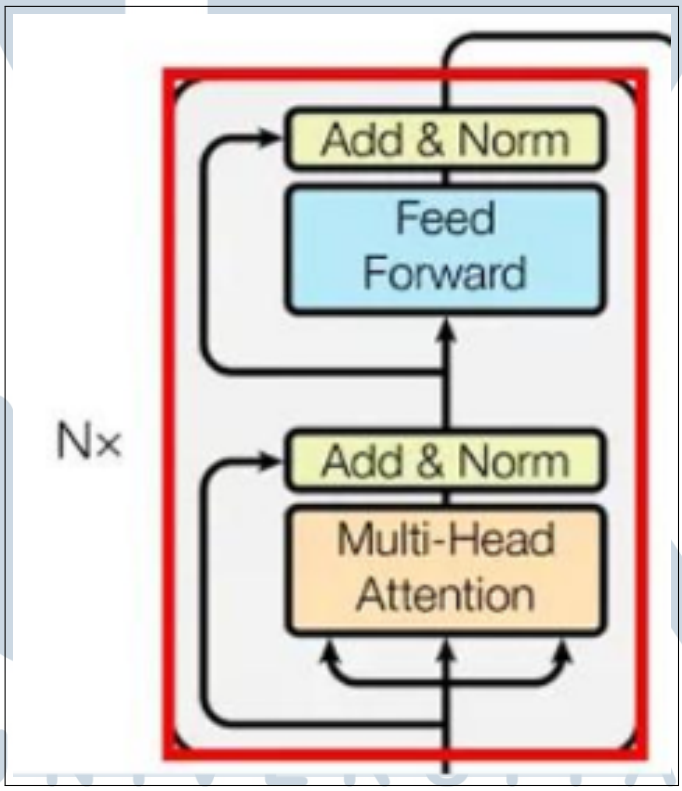
Sumber: [27]

Pada Gambar 2.1, bagian *encoder* dan *decoder* dibentuk melalui lapisan-lapisan yang disusun secara bertingkat. Keluaran dari suatu lapisan menjadi masukan bagi lapisan berikutnya. Representasi yang melewati susunan tersebut

mengalami pembaruan secara berulang, sehingga informasi yang dihasilkan tidak hanya berasal dari token pada satu posisi, tetapi juga telah melibatkan konteks dari posisi lain.

2.8.1 Encoder Transformer

Encoder mengubah urutan masukan menjadi sekumpulan representasi kontekstual. Proses tersebut dilakukan melalui beberapa lapisan dengan susunan komponen yang sama. Setiap lapisan menerima representasi dari lapisan sebelumnya, kemudian melakukan pembentukan hubungan antartoken dan transformasi pada masing-masing posisi. Aliran pemrosesan dalam satu lapisan *encoder* ditunjukkan pada Gambar 2.2.



Gambar 2.2. Komponen dalam satu lapisan encoder Transformer
Sumber: [27]

Satu lapisan *encoder* memiliki dua sublapisan utama, yaitu *multi-head self-attention* dan *feed-forward network* (FFN). Sublapisan *multi-head self-attention* digunakan untuk membentuk representasi berdasarkan hubungan di antara token-token dalam urutan. Keluaran dari proses tersebut kemudian diteruskan menuju

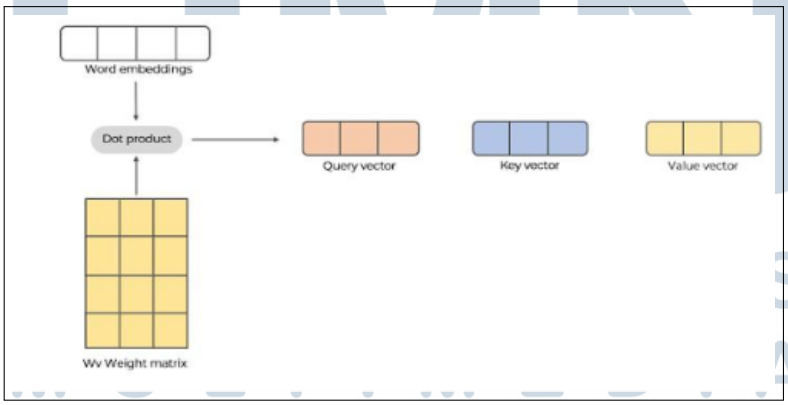
FFN untuk mengalami transformasi pada setiap posisi.

Keluaran pada setiap sublapisan digabungkan kembali dengan masukannya melalui koneksi residual. Jika x merupakan masukan suatu sublapisan, maka hasil transformasi dari sublapisan tersebut tidak langsung menggantikan x, tetapi ditambahkan kembali terhadap representasi tersebut. Hasil penambahan kemudian diproses menggunakan normalisasi lapisan. Susunan koneksi residual dan normalisasi ini diterapkan pada sublapisan *attention* maupun FFN [27]. Melalui susunan tersebut, representasi sebelumnya tetap memiliki jalur untuk diteruskan meskipun telah mengalami beberapa proses transformasi.

A *Self-Attention*

Self-attention melakukan perhitungan hubungan di antara posisi-posisi yang berasal dari urutan masukan yang sama. Representasi masukan diproyeksikan menjadi tiga matriks, yaitu *query* (Q), *key* (K), dan *value* (V). Ketiga matriks tersebut mempunyai sumber masukan yang sama, tetapi digunakan untuk fungsi yang berbeda dalam proses perhitungan.

Matriks *query* digunakan untuk merepresentasikan posisi yang sedang melakukan pencocokan [29]. Matriks *key* memuat representasi yang digunakan sebagai dasar dalam menentukan tingkat hubungan, sedangkan matriks *value* memuat informasi yang akan digabungkan untuk menghasilkan representasi baru. Hubungan ketiga matriks tersebut diperlihatkan pada Gambar 2.3.



Gambar 2.3. Pengolahan query, key, dan value pada mekanisme attention
Sumber: [29]

Skor hubungan antartoken diperoleh melalui perkalian antara matriks Q dan transpose dari matriks K. Perkalian tersebut menghasilkan skor bagi setiap

pasangan posisi di dalam urutan. Nilai skor kemudian dibagi dengan akar dari dimensi vektor *key*, yaitu $\sqrt{d_k}$. Penskalaan dilakukan untuk mengendalikan besarnya hasil perkalian ketika dimensi vektor *key* semakin besar. Setelah itu, fungsi *softmax* diterapkan untuk menghasilkan bobot hubungan yang digunakan terhadap matriks V. Proses tersebut dinyatakan dalam Persamaan 2.1 [27].

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

Keterangan:

1. Q : Matriks *query* yang digunakan untuk melakukan pencocokan terhadap posisi lain dalam urutan.
2. K : Matriks *key* yang digunakan sebagai dasar dalam menentukan tingkat hubungan antartoken.
3. V : Matriks *value* yang memuat informasi untuk membentuk representasi keluaran.
4. K^T : Hasil transpose dari matriks *key*.
5. d_k : Dimensi dari vektor *key* yang digunakan dalam proses penskalaan.
6. softmax$(\cdot)$: Fungsi normalisasi yang mengubah skor hubungan menjadi bobot perhatian.
7. Attention(Q,K,V) : Representasi keluaran yang diperoleh dari penggabungan matriks *value* berdasarkan bobot perhatian.

Bagian QK^T menghasilkan skor kecocokan antara setiap *query* dan *key*. Skor tersebut dibagi dengan $\sqrt{d_k}$ sebelum fungsi *softmax* diterapkan. Proses penskalaan dilakukan untuk mengurangi kemungkinan nilai hasil perkalian menjadi terlalu besar ketika dimensi vektor *key* bertambah. Bobot yang dihasilkan selanjutnya dikalikan dengan matriks V.

Posisi yang memperoleh bobot lebih besar memberikan kontribusi yang lebih besar terhadap representasi keluaran. Sebaliknya, informasi dari posisi dengan bobot yang lebih kecil digunakan dalam jumlah yang lebih terbatas. Hasil akhirnya berupa representasi baru pada setiap posisi yang telah memuat informasi dari token-token lain dalam urutan yang sama.

B *Multi-Head Attention*

Satu proses *self-attention* menghasilkan hubungan antartoken pada satu ruang proyeksi. Transformer menggunakan beberapa *attention head* agar perhitungan dapat dilakukan pada beberapa ruang proyeksi yang berbeda. Setiap *head* menghasilkan keluaran tersendiri, kemudian seluruh keluaran tersebut digabungkan dan diproyeksikan kembali menjadi satu representasi. Bentuk umum penggabungan tersebut diberikan pada Persamaan 2.2 [27].

$$\text{MultiHead}(Q,K,V) = \text{Concat}(head_1,\ldots,head_h)W^O \tag{2.2}$$

Keterangan:

1. $\text{MultiHead}(Q,K,V)$: Representasi keluaran yang dihasilkan oleh mekanisme *multi-head attention*.
2. Q : Matriks *query* yang digunakan sebagai masukan bagi seluruh *attention head*.
3. K : Matriks *key* yang digunakan sebagai masukan bagi seluruh *attention head*.
4. V : Matriks *value* yang digunakan sebagai masukan bagi seluruh *attention head*.
5. $head_1,\ldots,head_h$: Keluaran yang dihasilkan oleh setiap *attention head*.
6. h : Jumlah *attention head* yang digunakan.
7. $\text{Concat}(\cdot)$: Operasi penggabungan keluaran dari seluruh *attention head*.
8. W^O : Matriks bobot yang digunakan untuk memproyeksikan hasil penggabungan seluruh *head*.

Keluaran dari $head_1$ sampai dengan $head_h$ terlebih dahulu disatukan melalui operasi *Concat*. Hasil penggabungan tersebut kemudian diproyeksikan menggunakan matriks W^O. Proses ini menghasilkan satu representasi keluaran yang memuat hasil pemrosesan dari seluruh *attention head* dan mempunyai dimensi yang sesuai untuk diteruskan menuju komponen berikutnya pada lapisan *encoder*.

Setiap *head* memiliki matriks proyeksi yang berbeda untuk membentuk *query*, *key*, dan *value*. Matriks Q, K, dan V terlebih dahulu dikalikan dengan

parameter proyeksi yang dimiliki oleh *head* terkait. Perhitungan keluaran pada *head* ke-i dirumuskan pada Persamaan 2.3 [27].

$$head_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2.3}$$

Keterangan:

1. $head_i$: Representasi keluaran yang dihasilkan oleh *attention head* ke-i.
2. i : Indeks yang menunjukkan urutan suatu *attention head*.
3. Q : Matriks *query* sebelum melalui proses proyeksi pada *head* ke-i.
4. K : Matriks *key* sebelum melalui proses proyeksi pada *head* ke-i.
5. V : Matriks *value* sebelum melalui proses proyeksi pada *head* ke-i.
6. W_i^Q : Matriks bobot yang digunakan untuk memproyeksikan matriks *query* pada *head* ke-i.
7. W_i^K : Matriks bobot yang digunakan untuk memproyeksikan matriks *key* pada *head* ke-i.
8. W_i^V : Matriks bobot yang digunakan untuk memproyeksikan matriks *value* pada *head* ke-i.
9. Attention$(\cdot)$: Operasi *scaled dot-product attention* yang diterapkan terhadap hasil proyeksi Q, K, dan V.

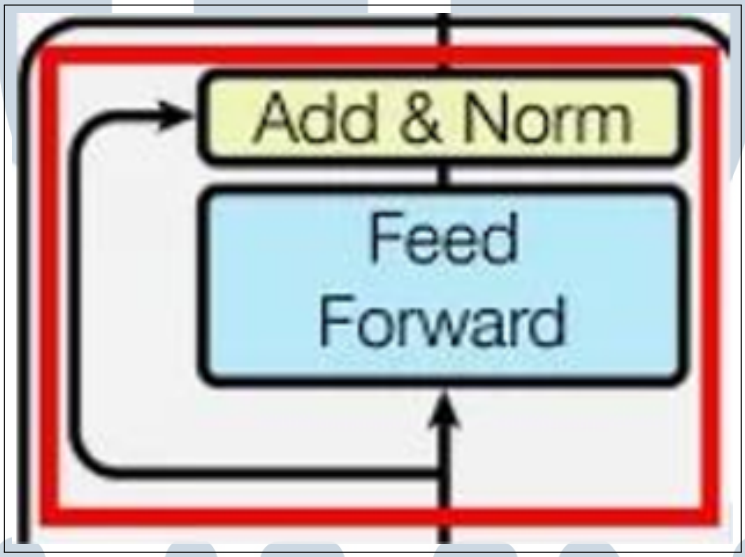
Ketiga matriks proyeksi tersebut merupakan parameter yang dipelajari selama proses pelatihan model. Setiap *head* mempunyai parameter proyeksi yang terpisah, sehingga hasil proyeksi yang digunakan dalam perhitungan *attention* tidak selalu sama antara satu *head* dan *head* lainnya. Perbedaan tersebut memungkinkan setiap *head* menghasilkan pola bobot perhatian yang berbeda walaupun representasi masukan yang diterima berasal dari urutan yang sama.

Perbedaan parameter menyebabkan setiap *head* dapat menghasilkan pola bobot perhatian yang tidak sama walaupun menerima representasi masukan yang sama. Keluaran dari beberapa pola perhatian tersebut kemudian digabungkan sebagaimana dinyatakan dalam Persamaan 2.2. Hasil penggabungannya tetap berupa satu representasi bagi setiap posisi token.

C *Feed-Forward Network*

Keluaran dari mekanisme *multi-head attention* diteruskan menuju *feed-forward network*. Komponen ini tidak menghitung kembali hubungan antartoken. Transformasi dilakukan terhadap setiap posisi secara terpisah, sedangkan parameter yang digunakan tetap sama bagi seluruh posisi dalam satu lapisan *encoder*. Oleh sebab itu, perubahan representasi pada FFN terjadi pada dimensi fitur setiap token dan bukan melalui pertukaran informasi langsung dengan posisi lain.

Susunan umum FFN pada Transformer diperlihatkan pada Gambar 2.4.



Gambar 2.4. Susunan feed-forward network pada Transformer
Sumber: [27]

FFN pada Transformer asli terdiri atas dua transformasi linear dengan fungsi aktivasi ReLU yang ditempatkan di antara keduanya [27]. Transformasi linear pertama memetakan representasi masukan dari dimensi model menuju dimensi tersembunyi FFN. Hasil pemetaan tersebut kemudian melewati fungsi aktivasi sebelum diproses oleh transformasi linear kedua. Transformasi kedua mengembalikan hasil pemrosesan menuju dimensi yang digunakan oleh lapisan *encoder*.

BERT mempertahankan susunan dua transformasi linear tersebut, tetapi menggunakan fungsi aktivasi GELU sebagai pengganti ReLU [28, 29]. Keluaran FFN selanjutnya digabungkan kembali dengan masukan sublapisan melalui koneksi residual dan diproses menggunakan normalisasi lapisan. Hasil dari proses ini menjadi keluaran satu lapisan *encoder* dan dapat diteruskan menuju lapisan

berikutnya.

Mekanisme *self-attention* dan FFN menjalankan fungsi yang berbeda dalam satu lapisan *encoder*. *Self-attention* membentuk representasi dengan melibatkan hubungan antartoken, sedangkan FFN melakukan transformasi lanjutan terhadap setiap posisi. Susunan kedua komponen tersebut digunakan pada lapisan *encoder* BERT dan tetap dipertahankan pada IndoBERT sebagai model yang dikembangkan berdasarkan arsitektur BERT [28, 30].

2.8.2 IndoBERT

IndoBERT adalah model bahasa yang dibangun berdasarkan arsitektur BERT yang dirancang untuk menangani berbagai bentuk teks dalam bahasa Indonesia [31]. Model ini dikembangkan oleh IndoNLU pada tahun 2020 dan dilatih menggunakan kumpulan teks berbahasa Indonesia yang berasal dari berbagai sumber, yang mencakup beragam jenis teks berbahasa Indonesia, mulai dari pemberitaan daring hingga konten yang berasal dari platform digital dan ensiklopedia daring [32, 30]. Penggunaan data pelatihan yang berasal dari lingkungan bahasa Indonesia memungkinkan model membentuk representasi bahasa yang lebih sesuai dengan karakteristik penulisan dalam bahasa tersebut. Oleh karena itu, IndoBERT mampu menangani berbagai bentuk penggunaan bahasa Indonesia, termasuk variasi penggunaan imbuhan, kata ulang, struktur kalimat pasif, serta perbedaan gaya bahasa formal dan informal [31, 30].

2.9 Hyperparameter Tuning

Hyperparameter tuning adalah tahapan untuk mencari konfigurasi *hyperparameter* yang digunakan untuk menentukan pengaturan model yang paling sesuai. Dalam penelitian ini, penyesuaian konfigurasi pada model *IndoBERT* dilakukan pada tahap *fine-tuning* dengan bantuan Optuna. Optuna merupakan *framework* yang mendukung proses eksplorasi berbagai konfigurasi *hyperparameter* secara otomatis. Adapun *hyperparameter* yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Learning Rate*

   Learning rate adalah *hyperparameter* yang menentukan besarnya langkah yang diambil model saat memperbarui parameter selama proses pelatihan. Nilai *Learning rate* yang terlalu rendah dapat memperlambat proses

pembelajaran, sementara nilai yang terlalu tinggi dapat menyebabkan perubahan parameter yang berlebihan [33].

2. *Batch Size*

   Batch size menentukan banyaknya sampel yang diproses model sebelum dilakukan pembaruan parameter. Perbedaan nilai *batch size* dapat memengaruhi kebutuhan komputasi serta pola pembelajaran model selama proses pelatihan [34].

3. *Epoch*

   Epoch merupakan satuan yang digunakan untuk menunjukkan jumlah siklus pelatihan terhadap seluruh data latih. Nilai *epoch* yang tinggi akan membuat model semakin sering mempelajari data yang sama selama proses pelatihan. Namun, penggunaan jumlah *epoch* yang terlalu tinggi juga dapat membuat model lebih banyak menghafal pola pada data latih dibandingkan mempelajari pola yang dapat digeneralisasikan ke data baru [34].

4. *Weight Decay*

   Weight decay adalah teknik yang digunakan untuk membatasi pertumbuhan nilai parameter model selama proses pelatihan [35]. Pembatasan tersebut dilakukan agar model tidak terlalu menyesuaikan diri terhadap data yang digunakan pada saat pelatihan.

5. *Dropout Rate*

   Dropout rate adalah parameter yang menentukan proporsi neuron yang dinonaktifkan sementara selama proses pelatihan [36]. Penonaktifan ini dilakukan secara acak pada setiap iterasi sehingga model tidak terlalu bergantung pada neuron tertentu ketika mempelajari pola dari data.

6. *Warmup Ratio*

   Warmup ratio adalah parameter yang menentukan proporsi tahap awal pelatihan yang digunakan untuk menaikkan nilai *learning rate* secara bertahap [37]. Penyesuaian ini dilakukan agar perubahan parameter model pada awal proses pembelajaran tidak terjadi secara terlalu agresif.

7. *Label smoothing*

   Label smoothing merupakan teknik yang diterapkan untuk mengurangi dominasi satu label selama proses pelatihan [38]. Dengan pendekatan ini, model tidak diarahkan untuk memberikan tingkat keyakinan yang terlalu

tinggi pada satu kelas tertentu sehingga hasil prediksi yang dihasilkan menjadi lebih seimbang.

2.10 Confusion Matrix

Confusion Matrix merupakan tabel yang digunakan untuk mengidentifikasi kesesuaian antara hasil prediksi model dan label sebenarnya pada data evaluasi [39]. Penyajian tersebut bertujuan untuk mengidentifikasi jumlah prediksi yang benar maupun kesalahan klasifikasi yang terjadi pada setiap kelas. Struktur dasar *confusion matrix* ditunjukkan pada Tabel 2.1.

Tabel 2.1. Hubungan Antara Label Aktual dan Hasil Prediksi Model

Label Aktual	**Positif**	**Negatif**
Positif	TP (Positif Benar)	FN (Negatif Salah)
Negatif	FP (Positif Salah)	TN (Negatif Benar)

Keterangan setiap komponen dalam *confusion matrix* dijelaskan sebagai berikut:

- TP (*True Positive*) menunjukkan jumlah data yang berhasil dikenali model sebagai kelas positif dan sesuai dengan label sebenarnya.
- TN (*True Negative*) menunjukkan jumlah data yang berhasil dikenali model sebagai kelas negatif dan sesuai dengan label sebenarnya.
- FP (*False Positive*) menunjukkan jumlah data yang diklasifikasikan ke dalam kelas positif meskipun label sebenarnya termasuk kelas negatif.
- FN (*False Negative*) menunjukkan jumlah data yang diklasifikasikan ke dalam kelas negatif meskipun label sebenarnya termasuk kelas positif.

Nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang diperoleh dari *confusion matrix* dapat digunakan untuk menghitung berbagai metrik evaluasi pada tugas klasifikasi, yaitu *accuracy*, *precision*, *recall*, dan *F1-Score*.

- *Accuracy*

  Accuracy merupakan metrik yang digunakan untuk menggambarkan tingkat kesesuaian hasil klasifikasi model terhadap data evaluasi secara keseluruhan.

Nilai *accuracy* diperoleh dari perbandingan antara jumlah prediksi yang sesuai dan jumlah seluruh data yang dievaluasi. Rumus perhitungan *accuracy* ditunjukkan pada Persamaan 2.4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

- *Precision*

  Precision digunakan untuk mengukur tingkat ketepatan model dalam memprediksi data yang termasuk ke dalam kelas positif. Rumus perhitungan *precision* ditunjukkan pada persamaan 2.5.

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

- *Recall*

  Recall digunakan untuk mengukur kemampuan model dalam mengidentifikasi seluruh data yang benar-benar termasuk dalam kelas positif. Rumus perhitungan *recall* ditunjukkan pada persamaan 2.6.

$$Recall = \frac{TP}{TP + FN} \tag{2.6}$$

- *F1-Score*

  F1-Score merupakan nilai rata-rata harmonik dari *precision* dan *recall* yang digunakan untuk menyeimbangkan kedua metrik tersebut. Rumus perhitungan *F1-Score* ditunjukkan pada persamaan 2.7.

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.7}$$

Rumus di atas digunakan secara langsung pada evaluasi model sentimen yang terdiri atas dua kelas. Pada model aspek yang terdiri atas kelas Mitigasi,

Tanggap Darurat, Pemulihan, dan Lainnya, evaluasi dilakukan untuk setiap kelas [40].

Pada klasifikasi dengan lebih dari dua kelas, *accuracy* dihitung dengan membagi jumlah seluruh prediksi yang benar dari setiap kelas dengan jumlah data evaluasi. Perhitungannya ditunjukkan pada Persamaan 2.8 [40].

$$Accuracy_{multikelas} = \frac{\sum_{k=1}^{K} TP_k}{N} \tag{2.8}$$

Keterangan:

1. $Accuracy_{multikelas}$ merupakan nilai akurasi untuk seluruh kelas.
2. TP_k merupakan jumlah data pada kelas ke-k yang diprediksi dengan benar.
3. k merupakan indeks kelas yang sedang dihitung.
4. K merupakan jumlah seluruh kelas. Pada model aspek penelitian ini, nilai K adalah 4.
5. N merupakan jumlah seluruh data evaluasi.
6. $\sum_{k=1}^{K}$ merupakan penjumlahan dari kelas pertama sampai kelas ke-K.

Nilai *precision*, *recall*, dan *F1-score* kemudian dihitung pada setiap kelas melalui Persamaan 2.9 [40].

$$P_k = \frac{TP_k}{TP_k + FP_k}, \qquad R_k = \frac{TP_k}{TP_k + FN_k}, \qquad F1_k = \frac{2P_kR_k}{P_k + R_k} \tag{2.9}$$

Keterangan:

1. P_k merupakan nilai *precision* pada kelas ke-k.
2. R_k merupakan nilai *recall* pada kelas ke-k.
3. $F1_k$ merupakan nilai *F1-score* pada kelas ke-k.
4. TP_k merupakan jumlah data pada kelas ke-k yang diprediksi dengan benar.
5. FP_k merupakan jumlah data dari kelas lain yang keliru diprediksi sebagai kelas ke-k.

6. FN_k merupakan jumlah data kelas ke-k yang keliru diprediksi sebagai kelas lain.

7. k merupakan indeks kelas yang sedang dihitung.

Agar hasil dari seluruh kelas dapat dirangkum, *macro average* menghitung rata-rata dengan memberikan bobot yang sama kepada setiap kelas. Rumusnya ditunjukkan pada Persamaan 2.10 [40].

$$P_{macro} = \frac{1}{K}\sum_{k=1}^{K} P_k, \qquad R_{macro} = \frac{1}{K}\sum_{k=1}^{K} R_k, \qquad F1_{macro} = \frac{1}{K}\sum_{k=1}^{K} F1_k \tag{2.10}$$

Keterangan:

1. P_{macro} merupakan rata-rata *precision* dari seluruh kelas.

2. R_{macro} merupakan rata-rata *recall* dari seluruh kelas.

3. $F1_{macro}$ merupakan rata-rata *F1-score* dari seluruh kelas.

4. P_k, R_k, dan $F1_k$ masing-masing merupakan nilai *precision*, *recall*, dan *F1-score* pada kelas ke-k.

5. k merupakan indeks kelas yang sedang dihitung.

6. K merupakan jumlah seluruh kelas.

7. $\frac{1}{K}$ menunjukkan bahwa setiap kelas memperoleh bobot yang sama.

Selain itu, *weighted F1-score* memberikan bobot berdasarkan jumlah data aktual pada setiap kelas. Perhitungannya ditunjukkan pada Persamaan 2.11 [40].

$$F1_{weighted} = \sum_{k=1}^{K} \frac{n_k}{N} F1_k \tag{2.11}$$

Keterangan:

1. $F1_{weighted}$ merupakan nilai *F1-score* yang mempertimbangkan jumlah data pada setiap kelas.

2. $F1_k$ merupakan nilai *F1-score* pada kelas ke-k.

3. n_k merupakan jumlah data aktual pada kelas ke-k.
4. N merupakan jumlah seluruh data evaluasi.
5. $\frac{n_k}{N}$ merupakan bobot kelas ke-k berdasarkan proporsi jumlah datanya.
6. k merupakan indeks kelas yang sedang dihitung.
7. K merupakan jumlah seluruh kelas.

2.11 Cohen's Kappa

Cohen's Kappa digunakan untuk mengukur tingkat kesepakatan antara dua penilai pada data kategorikal setelah memperhitungkan kesepakatan yang mungkin terjadi secara kebetulan [41]. Perhitungannya ditunjukkan pada Persamaan 2.12.

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{2.12}$$

Keterangan:

1. κ merupakan nilai Cohen's Kappa.
2. p_o merupakan proporsi kesepakatan yang diamati antara kedua penilai.
3. p_e merupakan proporsi kesepakatan yang diperkirakan terjadi secara kebetulan berdasarkan distribusi label kedua penilai.

Nilai Kappa menunjukkan reliabilitas atau konsistensi antar-penilai di atas tingkat kebetulan. Metrik ini tidak membuktikan bahwa label yang disepakati merupakan kebenaran absolut; validitas label tetap bergantung pada definisi operasional, pedoman anotasi, kompetensi penilai, dan prosedur penyelesaian perbedaan label.