BAB 2
LANDASAN TEORI

2.1 Threads

Threads merupakan paltform media sosisal yang dikembangkan oleh Meta dan diluncurkan pada tahun 2023. Platform ini berhasil mencapai tingkat pertumbuhan yang sangat cepat, dengan mencapai total 100 juta pengguna dalam jangka waktu 5 hari setelah diluncurkan [14].

Berbeda dengan platform media sosial baru pada umumnya, Threads memiliki keunikan berupa fenomena migrasi induk ke anak (*parent to child migration*), di mana platform ini terikat langsung dengan platform induknya yakni Instagram. Pengguna diwajibkan memiliki akun Instagram untuk dapat mendaftar dan mengakses Threads. Syarat kepemilikan akun ini memungkinkan terjadinya pembentukan jaringan sosial dengan lebih cepat, karena pengguna dapat langsung terhubung dengan pengikut mereka yang sudah ada di Instagram [14].

Dari segi karakteristik konten, Threads memiliki dinamika yang berbeda dibandingkan platform Instagram. Teks pada unggahan Threads lebih banyak didominasi oleh opini dan diskusi terkait isu-isu terkini, seperti politik dan teknologi, berbanding terbalik dengan Instagram yang lebih berfokus pada konten visual terkait gaya hidup [14]. Karakteristik topik yang berfokus pada opini dan diskusi publik ini membuat teks unggahan pengguna di Threads menjadi sangat dinamis dan kaya akan berbagai ekspresi emosi, sehingga menjadikannya objek data yang sangat relevan untuk diekstraksi dalam tugas pemrosesan bahasa alami, khususnya klasifikasi emosi.

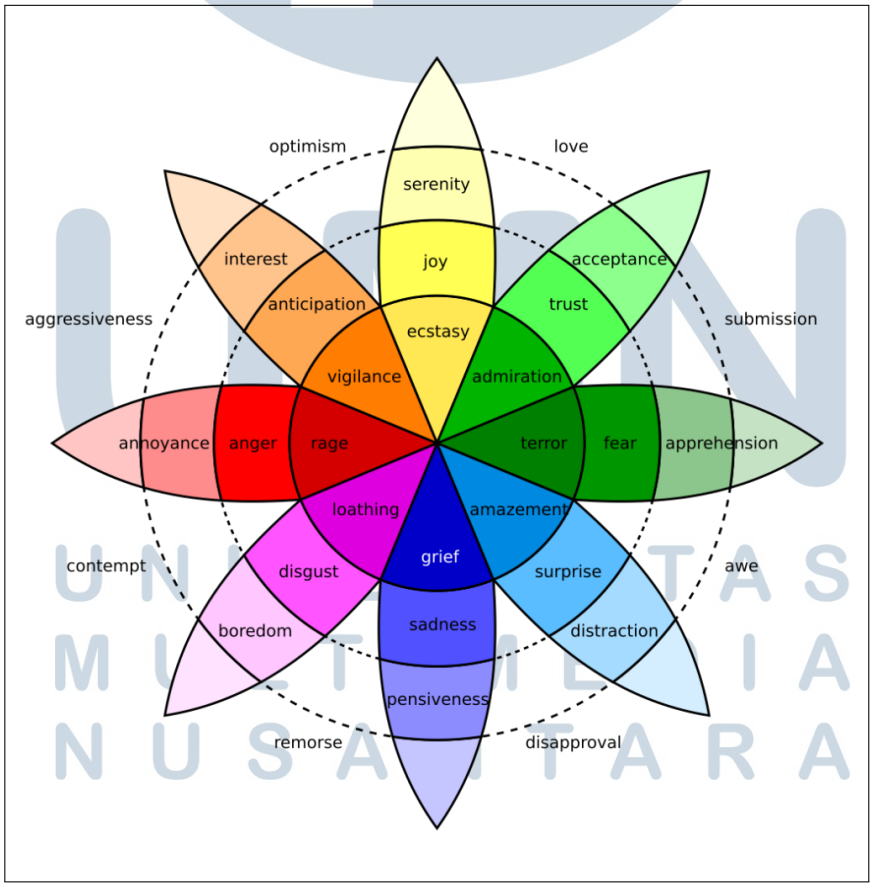
2.2 8 Emosi Dasar Robert Plutchik

Teori Psikoevolusioner Umum tentant Emosi (*General Psychoevolutionary Theory of Emotion*) yang dikembangkan oleh Robert Plutchik menyatakan bahwa konsep emosi berlaku di semua tingkat evolusi, mulai dari hewan hingga manusia [4]. Emosi tidak hanya sebatas perasaan subjektif, melainkan suatu rangkaian kompleks dari berbagai peristiwa yang saling berkaitan. Komponen ini mencakup pengalaman emosional, perubahan psikologis, dorongan untuk bertindak, hingga perilaku spesifik yang diarahkan pada tujuan tertentu [15].

Plutchik mengidentifikasi 8 emosi dasar (*basic/primary emotion*), yaitu:

1. Bahagia/Gembira (*Joy*)
2. Sedih (*Sadness*)
3. Marah (*Anger*)
4. Takut (*Fear*)
5. Percaya (*Trust*)
6. Muak (*Disgust*)
7. Antisipasi (*Anticipation*)
8. Terkejut (*Surprise*)

Keterkaitan antara kedelapan emosi tersebut divisualisasikan dalam bentuk Roda Emosi Plutchik (*Plutchik's Wheel of Emotions*), yang dapat dilihat pada gambar 2.1 [16].



Gambar 2.1. *Plutchik's Wheel of Emotions*

2.3 Natural Language Processing

Natural Language Processing (NLP) adalah sub disiplin dari ilmu komputer dan kecerdasan buatan (*Artifical Intelligence*) [1]. Studi dari NLP terdiri atas teori dan metode yang memunkinkan komunkasi efektif antara manusia dan komputer dalam bahasa alami. Tujuan utama dari NLP adalah menerjemahkan bahasa alami (bahasa manusia) menjadi serangkaian aturan atau perintah yang dapat dijalankan dan dipahami oleh komputer, sehingga menjadikan komunikasi antara manusia dan komputer lebih efektif [17].

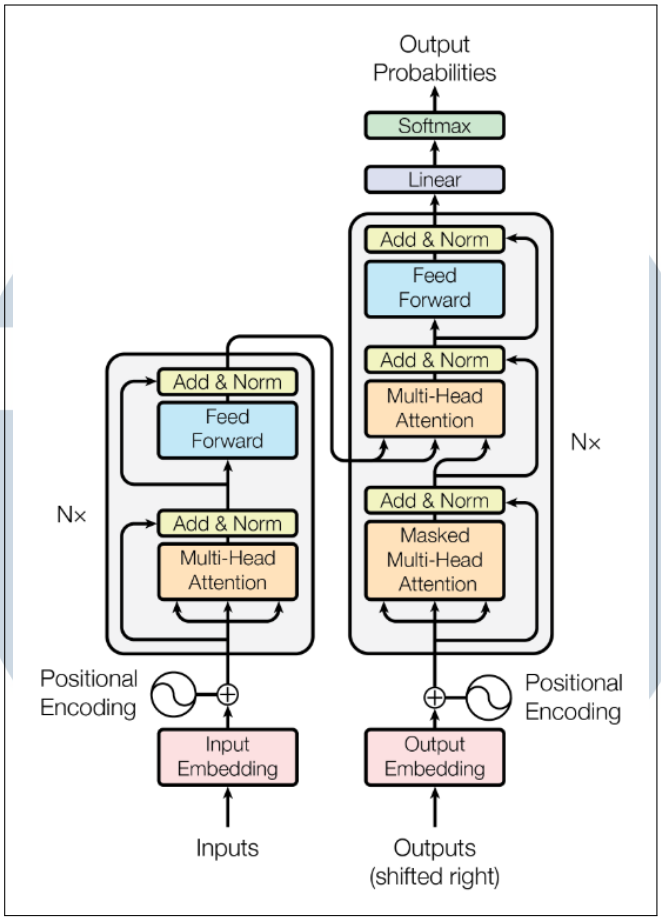
Natural Language Processing teridiri atas dua cabang arah penelitian, yaitu *Natural Language Understanding* (NLU) dan *Natural Language Generation* (NLG). Pada prinsipnya, tujuan dari NLU adalah untuk memahami bahasa alami (bahasa manusia) dengan cara menguraikan dokumen dan mengekstraksi informasi penting didalamnya. Kebalikan dari itu, NLG bertujuan menghasilkan teks berbahasa alami yang dapat dipahami oleh manusia berdasarkan input yang disediakan [17].

2.4 Transformer

Transformer merupakan sebuah arsitektur jaringan saraf tiruan (*neural network*) yang diperkenalkan oleh Vaswani et al. pada tahun 2017. Arsitektur ini sepenuhnya mengandalkan mekanisme *attention* (*attention mechanism*) dan meninggalkan penggunaan jaringan *recurrent* maupun *convolution* [6].

2.4.1 Arsitektur Transformer

Arsitektur Transformer dibangun atas tumpukan lapisan *self-attention* dan jaringan *feed-forward* yang terhubung dalam struktur *encoder-decoder* seperti yang terlihat pada Gambar 2.2. Keunggulan utama dari model ini adalah kemampuannya mengatasi keterbatasan model-model sebelumnya (seperti RNN) dengan membuang proses (*recurrence*) secara keseluruhan. Hal ini memungkinkan Transformer untuk menarik keterkaitan global antara *input* dan *output*, serta memproses seluruh *input* secara paralel tanpa adanya ketergantungan komputasi sekuensial [6].



Gambar 2.2. Arsitektur Transformer [6]

Mengacu pada diagram di Gambar 2.2, berikut adalah penjabaran operasional dari masing-masing komponen penyusun arsitektur Transformer beserta formulasi matematis yang melandasinya:

A Encoder

Bagian *encoder* tersusun atas tumpukan $N = 6$ lapisan yang identik. Setiap lapisan dikonstruksi oleh dua sub-lapisan utama, mekanisme *multi-head self-attention* dan *position-wise fully connected feed-forward network*. Setelahnya, diterapkan lapisan *residual connection* di sekitar kedua sub-lapisan tersebut, kemudian dilanjutkan dengan lapisan normalisasi [6].

B Decoder

Sama seperti encoder, decoder juga tersusun dari tumpukan lapisan yang identik ($N = 6$). Selain dua sub-lapisan yang biasa ditemukan pada encoder, decoder menambahkan sub-lapisan ketiga yang berfungsi menjalankan mekanisme *multi-head attention* terhadap hasil *output* dari tumpukan encoder. Serupa dengan encoder, *residual connection* diterapkan di sekeliling setiap sub-lapisannya, yang kemudian diikuti dengan lapisan normalisasi [6].

Di samping itu, terdapat modifikasi pada sub-lapisan *self-attention* di dalam tumpukan decoder untuk mencegah suatu urutan posisi memperhatikan posisi-posisi yang ada setelahnya. Penerapan *masking* ini dikombinasikan dengan fakta bahwa *output embeddings* digeser sebanyak satu posisi, memastikan bahwa prediksi untuk posisi ke-i hanya akan bergantung pada *output* yang sudah diketahui pada posisi-posisi sebelum i [6].

C Multi-Head Attention

Daripada hanya bergantung pada satu fungsi *attention*, Transformer memproyeksikan representasi kata ke dalam sub-ruang berdimensi lebih kecil sebanyak h kali (umumnya $h = 8$). Pendekatan ini memungkinkan model untuk memproses dan menggabungkan informasi dari berbagai posisi dan sub-ruang representasi secara bersamaan dan paralel [6].

D Position-Wise Feed-Forward Networks

Setiap lapisan pada *encoder* maupun *decoder* dilengkapi dengan jaringan *feed-forward* yang memproses masing-masing posisi secara independen. Proses ini mengubah vektor masukan x melalui dua proyeksi linear yang dipisahkan oleh fungsi aktivasi ReLU [6]:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.1}$$

Dalam notasi alternatif menggunakan vektor tersembunyi (*hidden vector*) h_i, operasi ini dapat dituliskan sebagai,

$$\mathrm{FFN}(h_i) = \mathrm{ReLU}(h_i W_1 + b_1)W_2 + b_2 \tag{2.2}$$

Pada implementasinya, transformasi ini mengekspansi dimensi masukan $d_{\text{model}} = 512$ menjadi $d_{\text{ff}} = 2048$ pada lapisan dalam (*inner-layer*), sebelum mengembalikannya ke dimensi awal [6].

E Residual Connection dan Layer Normalization

Setiap sub-lapisan pada arsitektur Transformer menggunakan *residual connection* (koneksi sisa) agar informasi gradien tidak hilang saat model dilatih [6]. Hasil dari koneksi sisa ini kemudian diproses melalui *Layer Normalization* untuk menstabilkan proses pelatihan model. Secara matematis, rumusan normalisasi untuk sebuah vektor masukan v menurut Ba er al. [18, 19] adalah sebagai berikut:

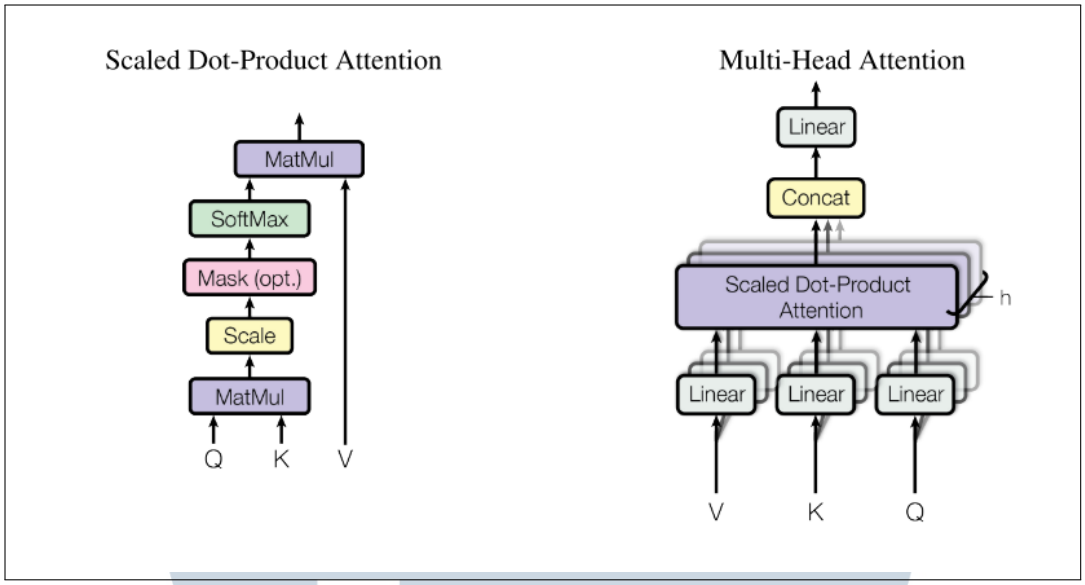
$$\text{LayerNorm}(v) = \gamma \frac{v - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{2.3}$$

di mana μ adalah parameter rata-rata (*mean*), σ^2 merepresentasikan varians, ε merupakan konstanta bernilai sangat kecil untuk stabilitas numerik guna mencegah pembagian dengan nol (*division by zero*), serta γ dan β merepresentasikan parameter skala (*scale*) dan bias yang dipelajari secara dinamis selama proses pelatihan [18, 19]. Maka dari itu, formula untuk *output* pada semua sub-lapisan dapat dituliskan sebagai [6],

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \tag{2.4}$$

2.4.2 Mekanisme *Self-Attention*

Konsep dasar dari fungsi *attention* memiliki kesamaan dengan sistem *information retrieval* (pengambilan kembali informasi). Berdasarkan Gambar 2.3, proses ini pada intinya menghitung hubungan antara matriks Kueri (Q) dengan Kunci (K) untuk mengekstrak informasi yang relevan dari matriks Nilai (V). Matriks Q, K, dan V itu sendiri dikonstruksi dari proyeksi linear representasi *word embedding* [6].



Gambar 2.3. Mekanisme *Scaled Dot-Product Attention* (kiri) dan *Multi-Head Attention* (kanan) [6]

Pada sisi kiri Gambar 2.3, operasi komputasi tersebut disebut sebagai *Scaled Dot-Product Attention*. Tahapan ini dimulai dari operasi perkalian titik (*dot product*) antara kueri Q dengan transposisi matriks kunci K^T. Hasilnya kemudian diskalakan dengan membaginya menggunakan skalar $\sqrt{d_k}$, yaitu akar dari dimensi kunci. Berdasarkan Vaswani et al. [6], penskalaan ini krusial untuk dilakukan. Tanpanya, hasil perkalian titik yang terlalu besar akan menyebabkan fungsi *softmax* berada pada area jenuh (*saturated regions*) dan memicu terjadinya masalah *vanishing gradients*. Langkah terakhir adalah mengaplikasikan fungsi *softmax* untuk memperoleh matriks bobot atensi yang kemudian dikalikan dengan nilai V:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

Berdasarkan sisi kanan Gambar 2.3, arsitektur Transformer tidak hanya bergantung pada satu fungsi *attention* tunggal. Sebaliknya, model ini menerapkan mekanisme *Multi-Head Attention*. Mekanisme ini menyatukan keluaran dari h buah sub-ruang *attention heads* melalui operasi penggabungan (*concatenation*). Hasil gabungan tersebut kemudian diproyeksikan kembali secara linear menggunakan matriks bobot W^O [6]:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O \tag{2.6}$$

di mana masing-masing *head* dikomputasi menggunakan formulasi berikut:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.7}$$

(dengan W_i^Q, W_i^K, dan W_i^V merepresentasikan matriks bobot proyeksi linear spesifik yang dipelajari untuk masing-masing kueri, kunci, dan nilai pada sub-ruang representasi ke-i).

2.4.3 Positional Encoding

Karena arsitektur Transformer tidak menggunakan mekanisme rekurensi (*recurrence*) maupun konvolusi (*convolution*), informasi mengenai posisi absolut dan relatif dari setiap token sangat diperlukan agar model dapat mengenali urutan *input*. [6].

Positional Encoding ditambahkan ke lapisan *input embedding* di posisi terbawah lapisan *encoder-decoder*. Komponen *positional encodings* memiliki ukuran dimensi d_{model} yang sama persis dengan vektor *embeddings*, sehingga keduanya dapat langsung dijumlahkan. Terdapat berbagai pilihan metode *positional encodings* yang dapat digunakan, baik yang parameternya dipelajari oleh model (*learned*) maupun yang bernilai tetap (*fixed*) [6].

Pada penelitiannya, Vaswani et al. Menggunakan fungsi sinus dan kosinus dengan frekuensi yang berbeda:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \end{aligned} \tag{2.8}$$

Dalam formula ini, *pos* mewakili posisi token dan i adalah indeks dimensinya, yang berarti setiap dimensi dari *positional encoding* merujuk pada satu fungsi *sinusoid*. Gelombang-gelombang tersebut memiliki panjang yang membentuk sebuah deret geometri dari 2π hingga $10000 \cdot 2\pi$. Pemilihan fungsi

ini didasari oleh asumsi bahwa model akan lebih mudah mengenali *attention* berdasarkan jarak relatif antar-token. Secara matematis, hal ini dibuktikan dengan sifat fungsinya di mana untuk setiap jarak (*offset*) k yang bernilai tetap, PE_{pos+k} selalu dapat dinyatakan sebagai bentuk linear dari PE_{pos} [6].

2.5 BERT

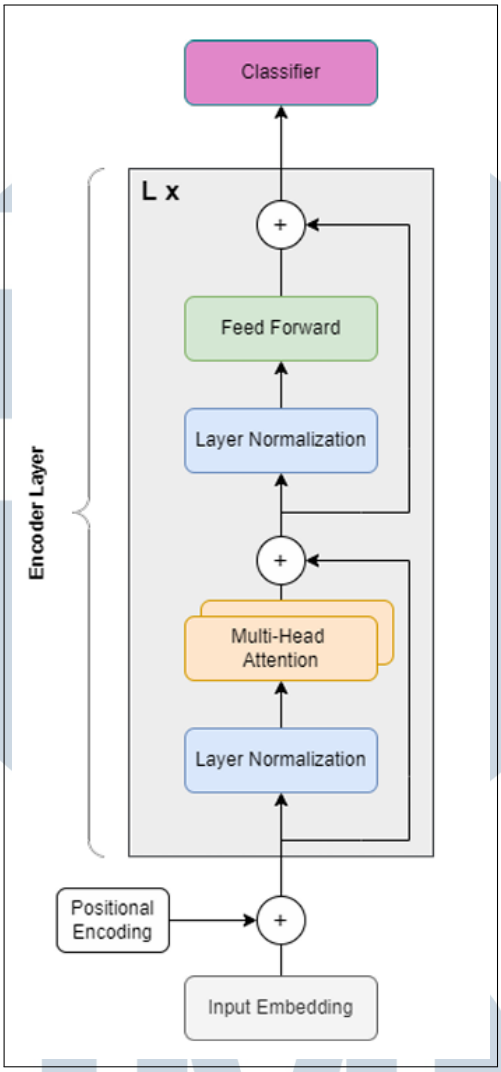
BERT (*Bidirectional Encoder Representation of Transformer*) adalah model *pre-trained* yang mendalam berdasarkan jaringan saraf arsitektur Transformer. Model ini dirancang untuk merepresentasikan teks dari dua arah (kiri ke kanan dan kanan ke kiri) secara bersamaan pada semua lapisan (*layers*) [20].

Model ini diperkenalkan pada akhir tahun 2018 oleh sekelompok peneliti dari laboratorium Google AI Language di bawah pimpinan J. Devlin [20]. Kemunculan BERT merepresentasikan sebuah pencapaian di bidang *Natural Language Processing* (NLP) dengan menggeser paradigma dari pendekatan tradisional menjadi pemanfaatan model *pre-trained* pada dataset teks berukuran masif [21].

BERT berperan sebagai model landasan (*foundation model*) universal yang dapat disesuaikan (*fine-tuned*) dengan hanya menambahkan satu lapisan keluaran (*output layer*) untuk mencapai kinerja *state-of-the-art* dalam berbagai tugas NLP, seperti analisis sentimen, pengenalan entitas bernama (NER), serta sistem tanya jawab [20].

2.5.1 Arsitektur BERT

Arsitektur BERT berpusat pada tumpukan *encoder* Transformer dua arah (*multi-layer bidirectional Transformer encoder*) yang mengadopsi struktur orisinal arsitektur Transformer rancangan Vaswani et al. (2017) [6]. Berbeda dengan Transformer standar yang terdiri atas konfigurasi utuh *encoder* dan *decoder*, BERT mengeliminasi komponen *decoder* dan hanya memanfaatkan blok *encoder*, sebagaimana diilustrasikan pada Gambar 2.4 [22]. Blok *encoder* ini mendayagunakan mekanisme *multi-head self-attention* dan jaringan *feed-forward* untuk mengkalkulasi bobot keterkaitan antara satu token dengan seluruh token lainnya di dalam sebuah sekuens masukan .



Gambar 2.4. Blok *Encoder* Transformer pada Arsitektur BERT [6, 20, 23]

Guna mengakomodasi skala komputasi yang beragam, arsitektur orisinal BERT dirilis dalam dua spesifikasi ukuran utama [20]:

A $\text{BERT}_{\text{BASE}}$

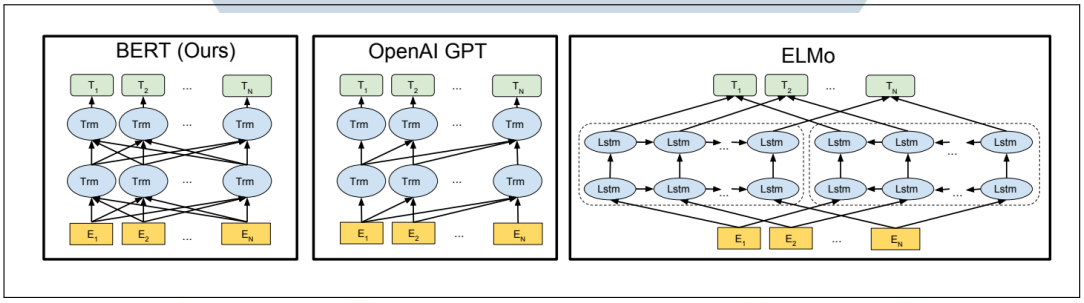
Varian ini dikonstruksi agar sepadan dengan spesifikasi model OpenAI GPT guna memfasilitasi komparasi performa yang adil [20]. Varian ini terdiri dari 12 tumpukan lapisan Transformer (L=12), dimensi vektor tersembunyi sebesar 768 (H=768), dan 12 mekanisme atensi (A=12), yang secara kumulatif merangkum sekitar 110 juta parameter terlatih [20, 21].

B $\text{BERT}_{\text{LARGE}}$

Varian berdimensi masif yang diekspansi menjadi 24 tumpukan lapisan Transformer (L=24), dimensi vektor tersembunyi sebesar 1024 (H=1024), dan 16 mekanisme atensi (A=16). Total kapasitas representasi parameter pada varian ini menyentuh angka 340 juta parameter [20, 21].

2.5.2 Bidirectional Learning

Karakteristik fundamental yang membedakan BERT dari arsitektur representasi bahasa lainnya adalah kapabilitas pembelajaran yang murni bersifat dua arah (*bidirectional*) secara mendalam. Dalam komputasinya, representasi fitur suatu kata dileburkan secara simultan dengan informasi kontekstual dari kata-kata yang mendahului dan mengikutinya pada setiap lapisan model.



Gambar 2.5. Komparasi Arsitektur Oleh Devlin et al. *Pre-Train* antara BERT (paling kiri), OpenAI GPT (tengah), dan ELMo (paling kanan) [20]

Berdasarkan komparasi visual pada Gambar 2.5, arsitektur BERT terbukti jauh lebih komprehensif dalam menyerap semantik kalimat. Model representasi bahasa pendahulunya, seperti OpenAI GPT, dibatasi oleh struktur searah (*unidirectional*) dari kiri ke kanan, sehingga proses penyerapan konteks token hanya terbatas pada kata-kata di sebelah kirinya [20]. Di sisi lain, model ELMo berupaya memecahkan batasan tersebut dengan menggabungkan arsitektur kiri-ke-kanan dan kanan-ke-kiri, namun penyatuan tersebut (*shallow concatenation*) hanya dilakukan secara dangkal di lapisan representasi akhir. BERT mengeliminasi kelemahan tersebut dengan memfasilitasi ekstraksi semantik kompleks secara dua arah murni sejak lapisan atensi pertama [20].

2.5.3 Pre-training pada BERT

Alih-alih dilatih menggunakan objektif pemodelan bahasa standar dari kiri ke kanan, arsitektur BERT membangun representasi bahasa fundamentalnya melalui dua objektif pelatihan tanpa pengawasan (*unsupervised tasks*) secara bersamaan, memanfaatkan korpus teks bervolume raksasa seperti English Wikipedia dan BooksCorpus [20]:

A Masked Language Modeling (MLM)

Objektif pembelajaran dua arah memiliki celah teknis di mana sebuah token berpotensi ”melihat dirinya sendiri” pada lapisan atensi multi-lapis. Untuk memitigasi anomali ini, BERT secara acak menyembunyikan (*masking*) 15% dari total sekuens *WordPiece token* pada setiap iterasi masukan, dan memaksa model untuk memprediksi identitas token orisinal yang disembunyikan tersebut [20]. Strategi probabilitas terhadap 15% token yang terpilih dijabarkan sebagai berikut:

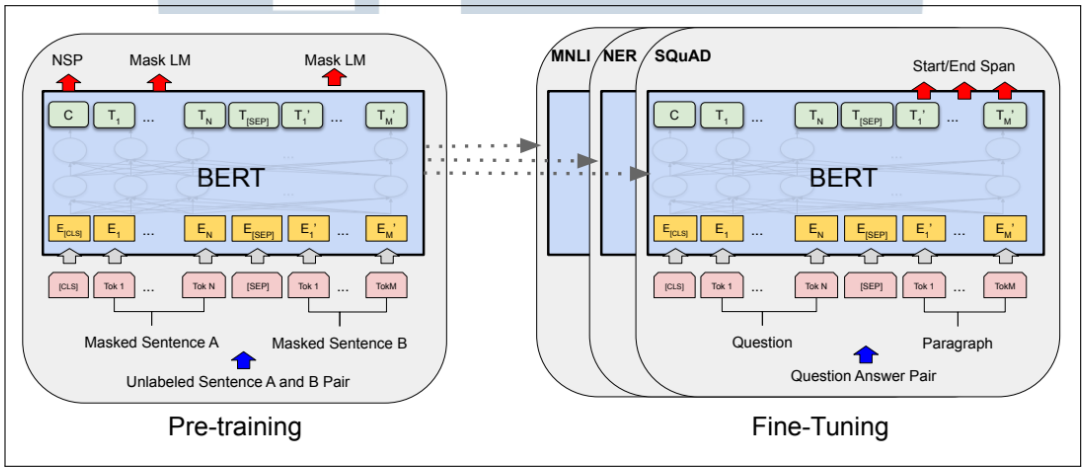
- 80% probalilitas token diganti dengan token spesial `[MASK]`.
- 10% probabilitas token diganti dengan token acak dari leksikon.
- 10% sisa probabilitas membiarkan token tersebut tidak berubah (sebagai referensi bias agar model dapat mengonfirmasi prediksi token yang benar) [20, 24].

B Next Sentence Prediction (NSP)

Sebagian besar tugas NLP di ranah hilir (*downstream tasks*), seperti Sistem Tanya Jawab (QA) maupun Inferensi Bahasa Alami (NLI), menuntut pemodelan untuk mengkorelasikan keterikatan logis antar-kalimat. Objektif NSP secara khusus melatih model untuk mengekstraksi relasi tersebut. Dalam teknisnya, model diumpankan dua buah kalimat berpasangan (A dan B). Pada 50% sampel, kalimat B adalah kelanjutan yang sebenarnya dari kalimat A (dilabeli sebagai *IsNext*) [20]. Sementara 50% sisanya menggunakan kalimat B yang diambil secara acak dari korpus (dilabeli sebagai *NotNext*). Model dilatih untuk membedakan kesinambungan relasi kedua kalimat tersebut secara presisi [20].

2.5.4 Fine Tuning BERT

Arsitektur BERT mengadopsi paradigma transfer pembelajaran (*transfer learning*), sebuah kerangka di mana model dasar pratalatih dapat direstrukturisasi ke dalam tugas spesifik (*downstream tasks*) tanpa memerlukan modifikasi arsitektural yang masif. Prosedur *fine-tuning* ini divisualisasikan pada Gambar 2.6. Fleksibilitas mekanisme *self-attention* memungkinkan model mengakomodasi tugas spesifik hanya melalui substitusi pasangan masukan dan luaran, dilanjutkan dengan tahap modifikasi komprehensif atas seluruh parameternya secara *end-to-end* menggunakan kumpulan data berlabel [20].



Gambar 2.6. *Pre-Training* dan *Fine-Tuning* pada Arsitektur BERT [20]

Secara spesifik untuk tugas klasifikasi teks, sekuens masukan (baik berupa kalimat tunggal maupun paragraf) disisipkan satu token penanda khusus, yakni `[CLS]`, pada indeks terdepan. Arsitektur model kemudian mengesktraksi vektor keluaran status tersembunyi (*hidden state*) terakhir dari token `[CLS]` tersebut, yang dinotasikan sebagai vektor $C \in \mathbb{R}^H$. Vektor ini difungsikan secara komputasional sebagai representasi agregat komprehensif (*aggregate sequence representation*) dari seluruh sekuens. Sebuah lapisan klasifikasi terhubung-penuh (*fully-connected classification layer*) kemudian diintegrasikan untuk memetakan ruang vektor tersebut ke dalam probabilitas label yang ditargetkan [20].

Distribusi probabilitas untuk kelas prediksi dihitung menggunakan fungsi *Softmax* terhadap hasil proyeksi representasi token agregat C yang dikalikan dengan parameter matriks bobot (W) dari lapisan spesifik klasifikasi tersebut:

$$P = \text{softmax}(CW^T) \tag{2.9}$$

di mana P merepresentasikan distribusi probabilitas kelas prediksi, C merupakan vektor fitur representasi luaran akhir dari token `[CLS]`, dan W adalah matriks parameter bobot dari lapisan klasifikasi yang dikalibrasi selama iterasi *fine-tuning* [20, 25].

Guna mengoptimisasi kalibrasi bobot selama proses pelatihan, fungsi perhitungan kerugian klasifikasi umumnya mendayagunakan *Categorical Cross-Entropy*. Formulasi matematis kerugian (J) terhadap parameter model (θ) didefinisikan sebagai berikut:

$$J(\theta) = -\sum_{c=1}^{M} y_c \log(p_c) \tag{2.10}$$

di mana y_c merupakan label target aktual yang berformat penandaan kategori tunggal (*one-hot encoded*), p_c adalah luaran prediksi probabilitas yang dihasilkan oleh fungsi *Softmax* untuk kelas ke-c, dan M mendeskripsikan total kuantitas ruang label kelas klasifikasi yang digunakan pada tugas spesifik [26].

2.6 IndoBERT

Pengertian IndoBERT IndoBERT adalah model bahasa prapelatihan (pre-trained language model) dengan representasi kontekstual mutakhir yang merupakan varian langsung dari arsitektur BERT (Bidirectional Encoder Representations from Transformers) [27]. IndoBERT dilatih secara eksklusif menggunakan kumpulan data (korpus) berskala sangat besar khusus berbahasa Indonesia. Model ini dibuat untuk memahami dan memetakan struktur semantik serta konteks dari teks bahasa Indonesia secara dua arah [27].

Pengembangan khusus Bahasa Indonesia Meskipun bahasa Indonesia merupakan salah satu bahasa dengan penutur terbanyak di dunia, penelitian NLP berbahasa Indonesia pada awalnya sangat terhambat karena kurangnya ketersediaan dataset dan belum adanya standardisasi sumber daya (under-represented) [7]. Berbeda dengan Multilingual BERT (mBERT) yang kapabilitasnya dibagi untuk memahami lebih dari 100 bahasa sekaligus, IndoBERT difokuskan secara eksklusif untuk mendalami linguistik dan budaya bahasa Indonesia [3]. Pengembangan khusus ini mencakup pembuatan kosakata modul WordPiece mandiri yang berisi 31.923 tipe token khusus berbahasa Indonesia untuk mengakomodasi penandaan

dan ejaan khas Indonesia [3, 7].

2.6.1 Dataset IndoBERT

IndoBERT (khususnya varian dari IndoLEM) dilatih menggunakan lebih dari 220 juta kata yang diagregasi ke dalam ukuran sekitar 23 GB teks bahasa Indonesia [7, 23]. Kumpulan data ini bersumber dari tiga domain utama:

A Indonesian Wikipedia

Terdiri dari 74 juta kata. Korpus ini memberikan model kemampuan untuk memahami struktur kalimat yang mewakili pengetahuan umum dalam berbagai bidang topik di ensiklopedia [3].

B Indonesian Web Corpus

Terdiri dari 90 juta kata yang dikumpulkan secara luas dari sumber-sumber web. Teks ini mengenalkan model pada gaya ekspresi kasual dan bahasa tidak formal sehari-hari yang sering ditemukan di internet serta User-Generated Content (UGC) [3].

C Indonesian News Corpus

Terdiri dari 55 juta kata yang diekstraksi dari artikel surat kabar ternama seperti Kompas, Tempo, dan Liputan6. Korpus berita ini membekali model dengan kepekaan terhadap gaya bahasa formal dan jurnalistik [3].

2.6.2 Arsitektur IndoBERT

Secara fundamental, arsitektur IndoBERT dibangun di atas fondasi *Multi-layer Bidirectional Transformer Encoder* [20, 27]. Komponen inti pembentuk arsitektur ini mencakup mekanisme *Multi-Head Attention* dan jaringan saraf *Feed-Forward* untuk proyeksi linear [23]. Kapabilitas utama dari arsitektur ini terletak pada mekanisme *self-attention* yang secara komputasional mampu memetakan relasi semantik antara satu token dengan seluruh token lain di dalam sekuens, baik yang mendahului maupun yang mengikutinya secara serentak [23].

Komputasi nilai atensi tersebut dikalkulasi menggunakan *Scaled Dot-Product Attention* sebagaimana yang telah dijabarkan sebelumnya pada Persamaan 2.5. Dalam mekanisme ini, matriks Kueri (Q), Kunci (K), dan Nilai (V) diekstraksi dari representasi tersembunyi (*hidden representation*) pada lapisan sebelumnya, dengan pembagian skalar $\sqrt{d_k}$ yang difungsikan untuk menstabilkan rentang gradien [19]. Selain itu, fase pratalatih model ini sepenuhnya mengadopsi objektif orisinal BERT, yakni *Masked Language Modeling* (MLM) dengan rasio penopengan 15% dan *Next Sentence Prediction* (NSP) untuk mengevaluasi koherensi logis antar-kalimat [27]. Mengingat IndoBERT mewarisi kerangka struktural yang identik dengan arsitektur BERT orisinal, ilustrasi komprehensif mengenai alur pemrosesannya dapat merujuk kembali pada Gambar 2.4 di sub-bab sebelumnya. Alur pemrosesan tersebut bergerak dari bawah ke atas, diawali dengan lapisan *Input Embeddings* yang merupakan hasil agregasi dari *Token*, *Segment*, dan *Position Embeddings*. Representasi tersebut kemudian diekstraksi melewati tumpukan lapisan blok *Encoder* Transformer yang memuat mekanisme *Self-Attention* dan *Normalization*, sebelum diakhiri pada lapisan klasifikasi luaran (*output layer*) yang mengambil representasi vektor dari token agregat `[CLS]` [27].

Untuk konfigurasi varian dasarnya (*Base*), IndoBERT mengadopsi spesifikasi yang identik dengan arsitektur $\text{BERT}_{\text{BASE}}$ (*uncased*), yang terdiri atas 12 tumpukan lapisan *encoder* ($L = 12$), dimensi vektor tersembunyi sebesar 768 ($H = 768$), dan 12 mekanisme atensi ($A = 12$) [3]. Guna memfasilitasi komputasi korpus berdimensi masif, dikembangkan pula varian turunan $\text{IndoBERT}_{\text{LARGE}}$ yang diekspansi menjadi 24 lapisan *encoder*, dimensi vektor 1024, dan 16 mekanisme atensi [23]. Perbedaan fundamental yang membedakan IndoBERT dari arsitektur BERT orisinal berbahasa Inggris terletak pada proses tokenisasi dan leksikonnya. *Encoder* IndoBERT dikalibrasi secara eksklusif menggunakan modul *WordPiece* yang memuat 31.923 sub-kata spesifik bahasa Indonesia. Adaptasi ruang kosakata ini memastikan bahwa prosedur tokenisasi teks kalimat bahasa Indonesia—termasuk penanganan struktur imbuhan yang kompleks—dapat dikonstruksi secara jauh lebih optimal dan presisi [3, 7].

2.6.3 Kelebihan dan Kekurangan IndoBERT

Mengadopsi arsitektur Transformer yang kompleks dan dilatih pada korpus berskala masif, IndoBERT menawarkan keunggulan komputasional yang revolusioner untuk pemrosesan bahasa Indonesia. Meski demikian, model ini

juga menghadirkan sejumlah tantangan teknis dalam implementasinya. Secara terstruktur, kelebihan dan kekurangan arsitektur IndoBERT dijabarkan sebagai berikut:

A Kelebihan IndoBERT

- Pemahaman Semantik yang Superior: Berkat implementasi mekanisme atensi dua arah (*bidirectional attention*), IndoBERT mampu menghasilkan penanaman kata yang sangat peka terhadap konteks (*context-aware embeddings*). Kapabilitas ini secara efektif memecahkan persoalan ambiguitas makna kata (polisemi) dengan mengevaluasi struktur sintaksis kalimat secara utuh [3, 27].
- Kinerja *State-of-the-Art*: Pada berbagai tolok ukur (*benchmarks*) evaluasi dataset bahasa Indonesia, IndoBERT secara konsisten mencatatkan performa yang melampaui algoritma konvensional, model *Deep Learning* rekuren (seperti CNN/LSTM), hingga model *Multilingual BERT* (mBERT). Keunggulan ini terbukti pada berbagai tugas hilir seperti peringkasan ekstraktif, Pengenalan Entitas Bernama (NER), dan klasifikasi teks [23].
- Fleksibilitas *Fine-Tuning*: Melalui penambahan lapisan luaran (*output layer*), parameter pratalatih IndoBERT dapat dikalibrasi ulang secara efisien untuk beradaptasi dengan analisis teks pada ranah industri atau domain spesifik, seperti deteksi emosi pada diskursus publik [3, 28].

B Kekurangan IndoBERT

- Beban Komputasi dan Memori yang Masif: Konfigurasi parameter yang raksasa (mencapai 110–124 juta pada varian dasar dan lebih dari 335 juta pada varian besar) sangat menuntut kapasitas ruang penyimpanan dan daya komputasi Unit Pemroses Grafis (GPU) yang tinggi. Hal ini menjadikan IndoBERT tidak efisien jika diimplementasikan secara langsung pada perangkat bersumber daya terbatas (*edge devices*) [23, 29].
- Limitasi Dimensi Sekuens Teks: Arsitektur dasar model ini menetapkan batasan masukan maksimal sebesar 512 token [7, 29]. Konsekuensinya, dokumen teks berskala panjang (seperti naskah jurnalistik atau riset) tidak

dapat diproses secara utuh tanpa melalui teknik pemotongan (*chunking*), yang berisiko memutus relasi konteks antar-paragraf [3, 29].

- Tantangan Ekstrem pada Korpus Media Sosial: Mengingat sumber data pratalatih utamanya didominasi oleh literatur formal (seperti artikel Wikipedia dan portal berita), IndoBERT kerap mengalami degradasi performa saat dihadapkan pada korpus media sosial yang informal. Model ini memiliki keterbatasan inheren dalam memetakan bahasa gaul(*slang*), singkatan non-baku, dan struktur tata bahasa tidak standar pada platform seperti Threads atau Twitter. Untuk menangani skenario ekstrem ini, sering kali dibutuhkan intervensi prapemrosesan yang sangat ketat atau adaptasi domain lanjutan (seperti varian IndoBERTweet) [28].

2.7 XLM - RoBERTa

XLM-RoBERTa (XLM-R) merupakan model representasi bahasa multibahasa berskala masif yang dibangun berdasarkan arsitektur *Transformer Masked Language Model* (MLM). Diperkenalkan oleh Conneau et al. (2020) [8], model ini telah melalui fase pratalatih (*pre-training*) menggunakan korpus teks yang mencakup 100 bahasa berbeda. Tujuan fundamental dari pengembangan arsitektur ini adalah untuk mendongkrak metrik kinerja pada berbagai tugas pemahaman bahasa lintas-bahasa (*Cross-lingual Language Understanding* / XLU). Hal ini dicapai melalui proses ekstraksi representasi fitur lintas-bahasa secara murni tanpa pengawasan (*unsupervised learning*) memanfaatkan volume data berskala raksasa [8, 30].

Sebagaimana yang tersirat dari penamaannya, XLM-R secara langsung mengadopsi dan mengintegrasikan prosedur pemodelan serta skenario pelatihan dari arsitektur RoBERTa (*Robustly Optimized BERT Approach*) [8,31]. Terinspirasi dari kesuksesan model monolingual tersebut, para peneliti mendemonstrasikan premis bahwa model representasi multibahasa pendahulunya, seperti mBERT dan XLM orisinal, pada dasarnya masih kurang dilatih secara optimal (*undertuned*) [8]. Mengadopsi filosofi optimalisasi RoBERTa, XLM-R merekonstruksi metodologi pratalatihnya melalui serangkaian peningkatan strategis. Peningkatan tersebut mencakup penggunaan ukuran tumpukan (*batch size*) komputasi yang jauh lebih masif, penyerapan volume data teks tanpa label (*unlabeled data*) yang diekspansi secara ekstrem, serta perpanjangan durasi iterasi pelatihan. Selain itu, modifikasi paling krusial yang diwarisi dari RoBERTa adalah eliminasi total terhadap tugas

Next Sentence Prediction (NSP) [31]. Penghapusan objektif komputasi antar-kalimat ini memungkinkan XLM-R untuk mengalokasikan seluruh kapasitas atensinya murni pada objektif *Masked Language Modeling* (MLM), yang terbukti secara empiris memberikan representasi semantik tingkat kata yang jauh lebih superior [8, 32].

2.7.1 Pembelajaran Lintas Bahasa (Cross-Lingual Learning)

Fase pembelajaran multibahasa pada arsitektur XLM-R dieksekusi secara gabungan (*jointly*) dengan mendayagunakan objektif *multilingual Masked Language Model* (MLM) yang dilatih murni di atas korpus data monolingual [8]. Arsitektur ini menyerap aliran teks dari beragam bahasa secara serentak dan dilatih untuk memprediksi identitas token yang disembunyikan (*masked tokens*) di dalam sekuens masukan [8]. Guna mengatasi ketimpangan volume data antar-bahasa, proses pengambilan sampel bahasa dikalibrasi menggunakan teknik penghalusan eksponensial (*exponential smoothing*). Probabilitas distribusi sampel tersebut dikontrol secara matematis oleh parameter $\alpha = 0,3$ [8]. Nilai parameter ini berperan krusial dalam meregulasi frekuensi kemunculan sampel, sehingga model dapat memproses tumpukan data (*batches*) dari bahasa berjejaring tinggi (*high-resource languages*, seperti bahasa Inggris) maupun bahasa berjejaring rendah (*low-resource languages*) secara lebih proporsional dan berimbang [8].

Skema pelatihan lintas bahasa berskala raksasa ini memfasilitasi terjadinya fenomena transfer lintas bahasa (*cross-lingual transfer*). Melalui mekanisme ini, model yang telah disesuaikan (*fine-tuned*) pada satu bahasa sumber dapat diaplikasikan untuk melakukan inferensi pada bahasa target lainnya secara sangat efektif, tanpa membutuhkan pengawasan data paralel tambahan untuk pelatihan silang [8]. Sebagai model yang mengelola 100 bahasa secara serentak, XLM-R dikonstruksi menggunakan ruang kosakata bersama (*shared vocabulary*) berdimensi sangat masif, yang mencakup 250.000 tipe token [8]. Berbeda dengan model pendahulunya, konstruksi kosakata pada XLM-R tidak bergantung pada prosedur prapemrosesan yang spesifik untuk setiap bahasa (*language-specific preprocessing*) maupun metode *Byte-Pair Encoding* (BPE) konvensional. Sebagai gantinya, arsitektur ini mengimplementasikan algoritma *SentencePiece Model* (SPM) [33] berbasis pemodelan bahasa *unigram* yang dieksekusi secara langsung pada teks mentah.

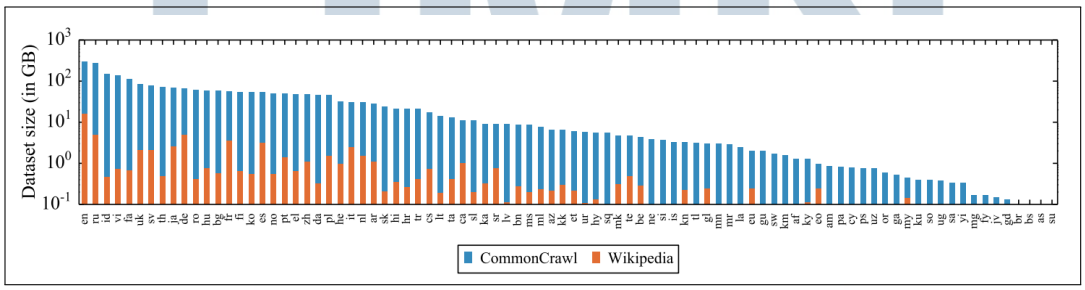
Dimensi leksikon yang sangat besar ini memegang peranan struktural yang

esensial. Ekspansi kapasitas kosakata secara langsung mengalokasikan persentase parameter yang jauh lebih besar ke dalam lapisan penanaman (*embedding layer*). Alokasi parameter masif pada lapisan awal ini memfasilitasi model multibahasa untuk menyerap kapasitas representasi semantik yang lebih kaya, yang pada akhirnya mendongkrak performa model secara signifikan pada berbagai tugas hilir (*downstream tasks*) [8].

2.7.2 Karakteristik dan Korpus Pre-Training XLM-R

Secara arsitektural, XLM-R dirancang untuk mengakomodasi pemrosesan 100 bahasa di dunia secara serentak. Cakupan ini merepresentasikan beragam rumpun bahasa utama dan mengikutsertakan pembaruan linguistik praktis secara ekstensif. Sebagai contoh, model ini memiliki dukungan yang dioptimalkan untuk memproses teks bahasa Hindi yang diromanisasi (*romanized Hindi*) serta karakter aksara Mandarin tradisional [8].

Terobosan paling signifikan dari arsitektur XLM-R terletak pada skala data pratalatihnya. Model ini dipelopori oleh pemanfaatan korpus teks berskala masif yang dikenal sebagai dataset CC-100. Dataset ini memiliki volume mencapai lebih dari 2,5 Terabita, yang dikompilasi dari ekstraksi *CommonCrawl* secara ekstensif pada 100 bahasa target [8]. Sebagaimana diilustrasikan pada Gambar 2.7, volume korpus *CommonCrawl* ini secara drastis mendominasi dan mengungguli skala dataset Wikipedia yang secara konvensional sering dijadikan standar pratalatih pada arsitektur model sebelumnya, seperti mBERT maupun XLM orisinal [8].



Gambar 2.7. Komparasi Volume Dataset (GiB) dalam Skala Logaritmik antara *CommonCrawl* dan Wikipedia pada Berbagai Bahasa di XLM-R [8]

Transisi korpus menuju dataset CC-100 memberikan eskalasi ketersediaan data pelatihan hingga rata-rata dua orde magnitudo, khususnya bagi bahasa-bahasa bersumber daya rendah (*low-resource languages*). Lebih dari sekadar peningkatan kuantitas data, pemodelan multibahasa gabungan pada XLM-R memicu terjadinya

fenomena transfer positif (*positive transfer*) [8]. Fenomena ini mendeskripsikan kondisi di mana bahasa bersumber daya rendah (seperti bahasa Swahili, Urdu, maupun variasi dialek lokal) memperoleh lonjakan performa yang signifikan akibat ”bantuan” representasi fitur dari bahasa bersumber daya tinggi (*high-resource languages*, seperti bahasa Inggris) selama fase pratalatih. Secara empiris, implementasi XLM-R mampu mendongkrak tingkat akurasi hingga 23% lebih tinggi dibandingkan dengan arsitektur mBERT pada skenario klasifikasi lintas bahasa untuk kategori *low-resource* [8].

2.7.3 Kelebihan dan Kekurangan XLM-RoBERTa

Sebagai salah satu model representasi bahasa lintas-bahasa yang paling komprehensif, implementasi XLM-RoBERTa (XLM-R) menawarkan terobosan performa sekaligus memunculkan konsekuensi arsitektural tertentu. Rincian mengenai kelebihan dan kelemahan operasional dari model ini dijabarkan sebagai berikut:

A Kelebihan XLM-RoBERTa

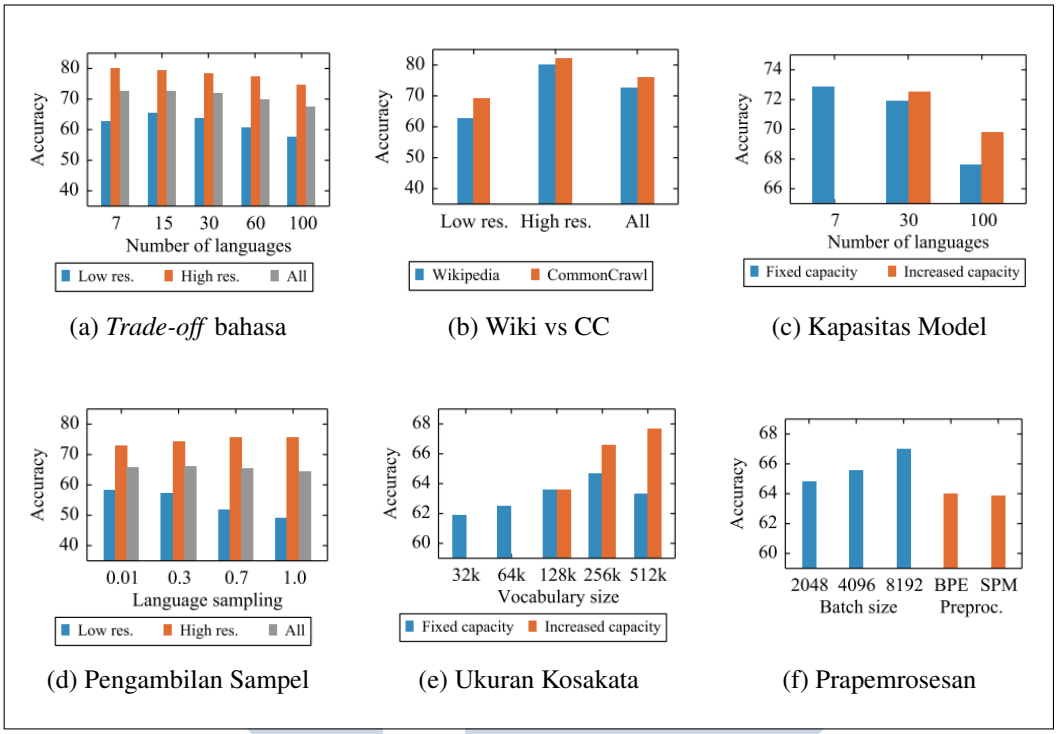
- Kinerja *State-of-the-Art* Multibahasa: XLM-R memecahkan batasan yang selama ini melekat pada model bahasa lintas-bahasa pendahulunya. Arsitektur ini sukses mencatatkan performa tertinggi pada berbagai tolok ukur pengujian (*benchmarks*), termasuk Inferensi Bahasa Alami Lintas-Bahasa (*Cross-lingual Natural Language Inference* / XNLI), Pengenalan Entitas Bernama (NER), hingga analisis sentimen dan Sistem Tanya Jawab (QA) [8].
- Kompetitif Terhadap Arsitektur Monolingual: Secara historis, model multibahasa sering kali mengorbankan performa untuk bahasa spesifik. Namun, XLM-R menjadi model berdimensi raksasa pertama yang terbukti mampu mengakomodasi 100 bahasa secara serentak, sekaligus tampil sangat kompetitif saat diadu dengan arsitektur monolingual terkuat sekelas RoBERTa pada uji spesifik bahasa seperti tolok ukur GLUE [8].
- Ketangguhan Terhadap Fenomena Campur Kode (*Code-Switching*): Dengan ditiadakannya *language embeddings* (injeksi representasi bahasa buatan ke dalam model), XLM-R dipaksa untuk memahami semantik secara universal.

Karakteristik ini membuat model secara natural jauh lebih tangguh dalam mengekstraksi konteks pada teks yang memuat fenomena campur kode (*code-switching* atau *code-mixing*), di mana pengguna mencampuradukkan dua bahasa atau lebih di dalam satu kalimat.

B Kekurangan XLM-RoBERTa

- Kutukan Multilingualitas (*Curse of Multilinguality*): Ini merupakan konsekuensi *trade-off* terhadap dilusi kapasitas model. Di bawah batasan kapasitas jumlah parameter yang konstan, upaya menambahkan bahasa baru ke dalam korpus pelatihan akan menggeser ruang representasi model, yang berujung pada menurunnya ”kapasitas per-bahasa”. Apabila ekspansi bahasa dilanjutkan melewati ambang batas maksimalnya, efek transfer positif (*positive transfer*) akan terhenti dan performa model pada keseluruhan bahasa justru akan mengalami degradasi akibat interferensi (*interference*). Solusi utama untuk memitigasi hal ini hanyalah dengan melakukan eskalasi dimensi vektor tersembunyi (*hidden size*) dan ukuran parameter model secara keseluruhan [8].

- Beban Komputasional Tingkat Tinggi: Mengelola model multibahasa dengan skala leksikon raksasa menuntut biaya komputasi yang ekstrem. Arsitektur ini dibebani oleh komputasi lapisan matriks probabilitas *softmax* atas ruang kosakata bersama (*shared vocabulary*) yang berukuran 250.000 token. Kondisi ini secara tidak langsung menguras porsi masif dari kapasitas parameter model yang seharusnya dapat dimaksimalkan untuk memperdalam tumpukan lapisan Transformer [8].



Gambar 2.8. Kumpulan Evaluasi Karakteristik Arsitektur XLM-RoBERTa: (a) *Trade-off* Transfer-Interferensi, (b) Komparasi Wikipedia dan *CommonCrawl*, (c) Peningkatan Kapasitas Model, (d) Parameter Pengambilan Sampel Bahasa, (e) Skala Ukuran Kosakata, dan (f) Dampak Penyederhanaan Prapemrosesan [8]

Visualisasi empiris dari fenomena ”Kutukan Multilingualitas” tersebut diilustrasikan secara komprehensif pada Gambar 2.8. Grafik tersebut menampilkan proyeksi penurunan tingkat akurasi pada bahasa-bahasa bersumber daya tinggi (*high-resource*) secara proporsional seiring dengan bertambahnya jumlah bahasa (mulai dari 7, 15, 30, 60, hingga 100 bahasa) yang direpresentasikan dalam satu arsitektur model bervolume parameter tetap [8].

2.8 Pustaka Leksikon NRCLex dan EMOLEX

2.8.1 Pendekatan Berbasis Leksikon (Lexicon-based)

Pendekatan berbasis leksikon (*lexicon-based* atau *key-word spotting*) adalah metode fundamental yang menganggap bahwa suatu teks terdiri dari deretan kata, di mana sentimen atau kandungan emosinya dapat ditentukan secara langsung berdasarkan muatan emosi yang terungkap pada kata-kata tersebut [34]. Berbeda dengan metode statistik atau *machine learning* yang membutuhkan pelatihan model menggunakan dataset berlabel besar, pendekatan leksikon beroperasi dengan

mendayagunakan daftar kamus (korpus kata) yang telah dipetakan dengan berbagai emosi terkait. Sistem bekerja dengan cara memecah teks masukan menjadi token-token individu, kemudian mencocokkannya dengan basis data leksikon yang memuat skor sentimen [34]. Pendekatan ini memungkinkan otomasi klasifikasi emosi berjalan selama dokumen yang diuji memiliki minimal satu fitur kata yang dikenali dalam leksikon, sehingga penentuan kelas emosi dapat dilakukan tanpa memerlukan proses pelatihan (*training*) awal yang masif [34].

2.8.2 Struktur Kamus EMOLEX

Dalam konteks klasifikasi emosi teks, salah satu sumber daya pustaka leksikon yang paling diandalkan dan komprehensif adalah *NRC Word-Emotion Association Lexicon* (EMOLEX), yang diciptakan oleh Saif M. Mohammad dan Peter D. Turney pada tahun 2013 [35]. Pembangunan EMOLEX dikonstruksi secara empiris melalui proyek urun daya (*crowdsourcing*) berskala besar menggunakan platform *Amazon Mechanical Turk* [35]. Sistem ini memetakan puluhan ribu kata ke dalam dua kutub sentimen (positif dan negatif) serta delapan kelas emosi dasar (*basic emotions*) yang bersandar pada landasan teoretis Plutchik: *Anger* (marah), *Anticipation* (antisipasi), *Disgust* (jijik/muak), *Fear* (takut), *Joy* (gembira), *Sadness* (sedih), *Surprise* (terkejut), dan *Trust* (percaya/yakin) [34]. Pada perkembangannya, struktur basis data EMOLEX yang awalnya berbasis bahasa Inggris telah difasilitasi dengan translasi leksikal lintas bahasa ke dalam 105 bahasa, yang memuat total 14.182 kata representasi emosional untuk kosakata bahasa Indonesia. Hal ini menjadikannya sangat mumpuni dalam membedah dimensi afektif teks-teks berbahasa non-Inggris.

2.8.3 Implementasi Anotasi Leksikon

Otomatisasi pelabelan data menggunakan EMOLEX dieksekusi melalui teknik ekstraksi fitur kata, di mana algoritma akan mendeteksi irisan semantik token dalam teks dan memberikan probabilitas skor biner (bernilai 0 atau 1) untuk masing-masing label emosi [34]. Nilai 1 menandakan bahwa fitur kata tersebut berasosiasi erat dengan jenis emosi yang bersangkutan, sementara nilai 0 menandakan ketiadaan asosiasi. Sebagai representasi implementatif, sebuah kata mungkin saja tidak terikat secara eksklusif pada satu dimensi emosi. Misalnya, jika sebuah korpus teks memuat kata leksikon ”bencana”, sistem secara otomatis

dapat mengasosiasikannya sebagai pemicu multikelas emosi, yang secara dominan diwakili oleh kelas *Fear* (Ketakutan) dan *Sadness* (Kesedihan). Berdasarkan data empiris praktis di dalam kamus, sebuah term seperti ”marah” memiliki representasi unik yang memicu skor 1 hanya pada kelas *Anger*, sebaliknya term seperti ”pecah” berasosiasi dengan dua jenis emosi, memicu probabilitas skor pada kelas *Sadness* dan *Surprise* sekaligus [34]. Agregasi atau pembobotan skor keseluruhan dari kemunculan kata-kata berbobot emosi di dalam suatu kalimat ini kemudian diakumulasikan (menggunakan metode *pooling* untuk mencari nilai skor probabilitas tertinggi), yang pada akhirnya menjadi landasan empiris untuk menetapkan label emosi (*ground truth*) otomatis pada level kalimat atau dokumen [34].

2.9 Explainable AI

Explainable Artificial Intelligence (XAI) dirancang secara khusus untuk membuka mekanisme ”kotak hitam” (*black-box*) dari model kecerdasan buatan yang kompleks, guna memastikan bahwa proses komputasional dapat dipahami secara transparan oleh manusia [**?**]. Seiring dengan ekskalasi jumlah parameter pada arsitektur pemrosesan bahasa alami, transparansi menjadi instrumen krusial untuk membangun kepercayaan sekaligus mendiagnosis potensi bias [**?**]. Pada implementasi model yang berisiko tinggi, interpretabilitas merupakan prasyarat mutlak agar setiap keputusan prediksi dapat dipertanggungjawabkan secara logis dan selaras dengan standar etika komputasi.

Sebagai instrumen analitik untuk mewujudkan transparansi tersebut, metode SHapley Additive exPlanations (SHAP) memformulasikan kerangka kerja terpadu untuk mengevaluasi besaran kontribusi setiap fitur terhadap sebuah luaran (output) prediksi [11]. Pemahaman terhadap nilai SHAP ini memfasilitasi developer atau data scientist dalam melakukan proses debugging model secara presisi. Mekanisme evaluasi ini krusial untuk meninjau apakah model telah mempelajari representasi linguistik yang benar (misalnya tidak salah mengenali tanda baca atau konteks kalimat), serta memvalidasi bagaimana setiap token atau kata memberikan signifikansi terbesar terhadap probabilitas klasifikasi emosi yang dihasilkan.

2.9.1 SHAP

SHAP (*SHapley Additive exPlanations*) adalah sebuah kerangka kerja yang bersifat *model-agnostic* (dapat diimplementasikan pada arsitektur *machine learning* apa pun) yang difungsikan untuk menginterpretasikan luaran prediksi model. Pendekatan ini bekerja dengan memberikan penjelasan secara lokal melalui kalkulasi nilai kontribusi (skor) dari setiap fitur (*input*) terhadap suatu prediksi instans spesifik. Nilai-nilai lokal ini juga dapat diagregasi untuk memproyeksikan penjelasan perilaku model secara global [11, 36].

Secara teoretis, kalkulasi SHAP dilandasi oleh teori permainan kooperatif (*cooperative game theory*) yang diformulasikan oleh Lloyd Shapley pada tahun 1953 [37, 38]. Dalam analogi matematis ini, fitur-fitur yang merepresentasikan suatu data diasumsikan sebagai ”pemain” (*players*) yang berpartisipasi dalam sebuah permainan kooperatif, sementara hasil prediksi dari model direpresentasikan sebagai ”pembayaran” atau hasil akhir (*payout/outcome*) [36]. SHAP mengekstraksi nilai *Shapley* (*Shapley values*) untuk mendistribusikan hasil prediksi tersebut secara proporsional dan adil kepada setiap fitur, berdasarkan kalkulasi bobot pengaruh marginal yang disumbangkan oleh masing-masing pemain [38].

2.9.2 Shapley Value

Nilai *Shapley* merupakan mekanisme komputasi matematis yang menjamin distribusi kontribusi secara adil (memenuhi properti efisiensi, simetri, dan aditivitas). Nilai ini diekstraksi dengan mengevaluasi performa model pada seluruh kemungkinan kombinasi subset fitur yang ada. Formulasi matematis dari perhitungan ini diekspresikan pada Persamaan 2.11:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{2.11}$$

dengan definisi variabel sebagai berikut:

- N merepresentasikan himpunan seluruh fitur (pemain) yang dilibatkan dalam pemodelan.
- i merepresentasikan fitur spesifik yang sedang dikalkulasi nilai kontribusinya.
- S merepresentasikan subset dari fitur-fitur yang tidak mengandung fitur i.

- $|N|$ dan $|S|$ merepresentasikan kardinalitas (jumlah fitur) di dalam himpunan N dan S.
- $f(S \cup \{i\}) - f(S)$ melambangkan kontribusi marginal (*marginal contribution*) yang dihasilkan dari penambahan fitur i ke dalam subset S [11]. Koefisien faktorial yang mendahuluinya berfungsi sebagai penyeimbang (normalisasi) probabilitas berdasarkan jumlah permutasi kombinasi dari subset S [37, 38].

2.9.3 Mekanisme Kerja SHAP

Secara operasional, pendekatan SHAP memecah interpretasi model ke dalam tiga dimensi analisis utama:

A Kontribusi Fitur

Mekanisme ini mengukur magnitudo dampak sebuah fitur terhadap deviasi nilai prediksi jika dibandingkan dengan nilai ekspektasi dasar (*base value* atau rata-rata prediksi tanpa kehadiran fitur) [11, 39]. Fitur yang mendorong hasil prediksi naik akan menghasilkan skor SHAP bernilai positif, sementara fitur yang menekan hasil prediksi akan menghasilkan skor negatif [37].

B Penjelasan Lokal (Local Explanation)

Pada dimensi analitik tingkat instans, SHAP memetakan alokasi nilai kepentingan fitur untuk satu baris data spesifik secara individual [36]. Analisis ini berfokus pada dinamika kausalitas mengapa model menghasilkan suatu keputusan tertentu untuk satu kasus observasi (misalnya, prediksi kelas emosi pada satu dokumen teks) dengan mengevaluasi interaksi dorongan dan tarikan skor fitur dari *base value* [37, 38].

C Penjelasan Global (Global Explanation)

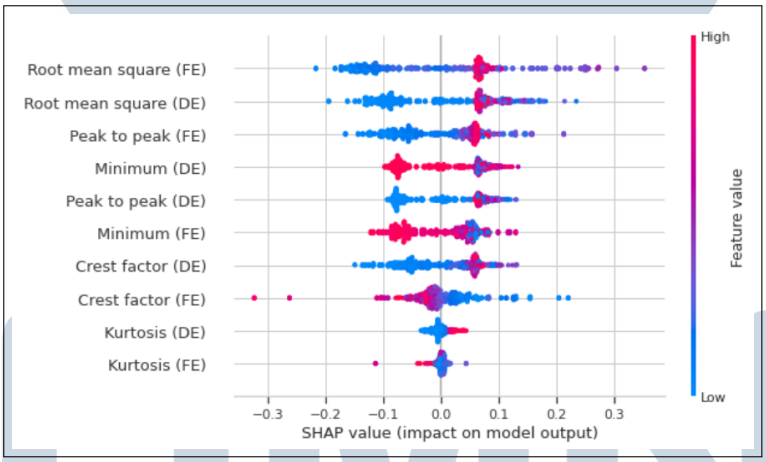
Pemahaman makro terhadap perilaku model diperoleh melalui agregasi dan rata-rata nilai penjelasan lokal dari seluruh instans di dalam dataset [36]. Melalui sintesis ini, pengguna dapat mengidentifikasi variabel mana yang secara global memegang peranan paling esensial, serta meninjau arah pengaruh rata-ratanya terhadap kapabilitas prediksi model [36, 40].

2.9.4 Spektrum Visualisasi SHAP

Untuk memfasilitasi proses interpretasi hasil komputasi matematis, pustaka SHAP menyediakan beragam instrumen visualisasi analitik, baik untuk ruang lingkup lokal maupun global:

A Summary Plot

Visualisasi ini memproyeksikan ringkasan komprehensif efek seluruh fitur secara agregat (global). Fitur-fitur diurutkan pada sumbu vertikal berdasarkan tingkat kepentingannya. Titik-titik persebaran merepresentasikan instans data secara horizontal berdasarkan skor SHAP yang dihasilkan, sementara spektrum gradasi warna (dari biru ke merah) mengindikasikan nilai aktual fitur dari rendah ke tinggi [39, 40].



Gambar 2.9. Ilustrasi *Summary Plot*

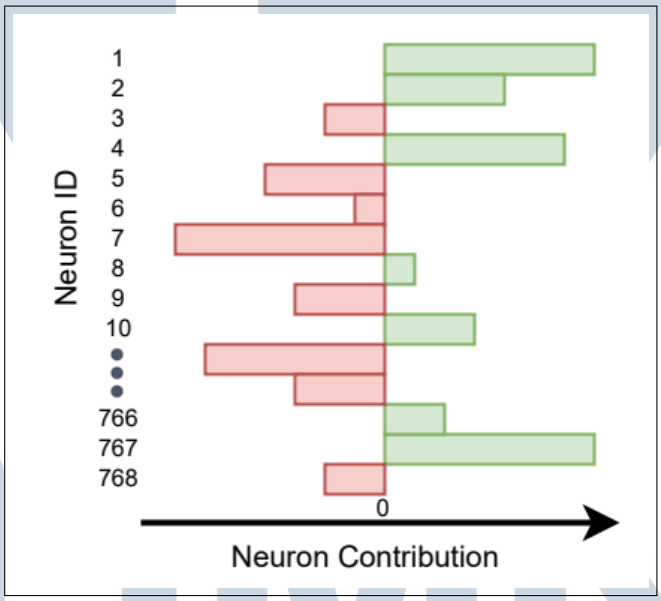
B Force Plot

Visualisasi yang ditujukan untuk penjelasan lokal ini merepresentasikan vektor dorongan kontribusi setiap fitur, bertolak dari *base value* menuju prediksi *output* aktual. Fitur bervalidasi positif yang mendorong prediksi divisualisasikan dengan blok panah berwarna merah, sedangkan dorongan fitur negatif direpresentasikan dengan warna biru [37, 41].

Gambar 2.10. Ilustrasi *Force Plot*

C Waterfall Plot

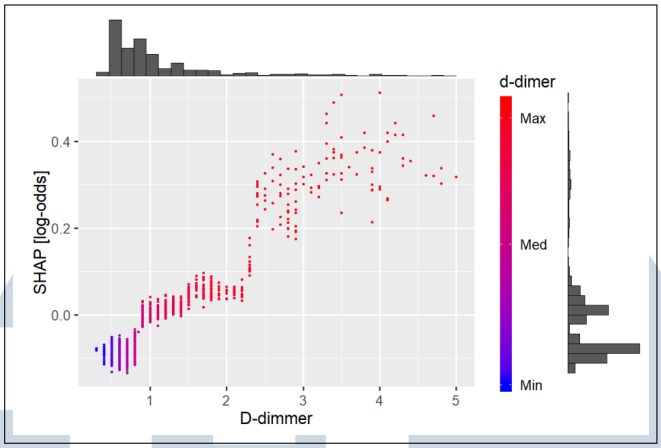
Serupa dengan *Force Plot*, visualisasi lokal ini mendekonstruksi kontribusi prediksi menjadi barisan anak tangga vertikal secara sekuensial. Setiap langkah anak tangga memvisualisasikan margin penambahan atau pengurangan secara individual dari *base value* hingga mencapai ekuilibrium prediksi akhir [36, 37].



Gambar 2.11. Ilustrasi *Waterfall Plot*

D Dependence Plot

Plot sebaran ini mendeskripsikan korelasi interaktif antara variasi nilai sebuah fitur (sumbu horizontal) dengan efek skor SHAP-nya (sumbu vertikal). Instrumen ini sangat krusial untuk menangkap pola non-linear dari pengaruh fitur terhadap *output* serta mengidentifikasi interaksi tersembunyi antar-fitur [39].



Gambar 2.12. Ilustrasi *Dependence Plot*

2.9.5 Penerapan SHAP pada Arsitektur Transformer

Arsitektur model pemrosesan bahasa berbasis Transformer (seperti BERT, IndoBERT, dan XLM-RoBERTa) mengeksekusi teks dengan memecahnya ke dalam token-token sub-kata (*subword tokens*). Mengingat kalkulasi SHAP secara inheren dirancang untuk memproses data berdimensi tabular, implementasinya pada domain NLP menuntut sebuah adaptasi komputasional [41]. Pada praktiknya, representasi teks di perturbasi (dilakukan penopengan atau *masking* pada sejumlah token) sebelum diumpankan kembali ke dalam model. Kalkulasi deviasi probabilitas hasil prediksi ini memungkinkan SHAP untuk mengekstraksi nilai *Shapley* secara independen untuk setiap unit fragmen teks (token) di dalam sekuens, alih-alih pada kolom numerik statis [41].

Dalam konteks klasifikasi teks, seperti deteksi sentimen maupun pelabelan spektrum emosi pada cuitan media sosial, SHAP berfungsi untuk menyingkap token spesifik mana yang memberikan bobot dorongan tertinggi (baik memihak pada suatu probabilitas kelas emosi maupun berlawanan terhadapnya) [41]. Sebagai studi kasus, pada sebuah dokumen teks yang diklasifikasikan ke dalam kelas emosi *Anger* (Kemarahan), mekanisme SHAP berkemampuan untuk mendeteksi bahwa keberadaan token spesifik (seperti kata makian atau leksikon afektif) mendominasi distribusi probabilitas dengan skor SHAP positif yang tinggi. Sebaliknya, token sintaktis pendukung (seperti tanda baca atau pronomina) umumnya memproyeksikan skor SHAP yang asimtotik terhadap nol [41].

Guna menyederhanakan analisis visual pada teks berurutan, luaran komputasi SHAP diadaptasi ke dalam wujud penyorotan teks (*text highlighting*).

Dalam antarmuka visualisasi ini, setiap token diberikan intensitas warna yang proporsional sesuai dengan kedudukannya pada kalimat asli (umumnya merah untuk kontribusi positif yang menaikkan probabilitas, dan biru untuk kontribusi berlawanan), sehingga memvalidasi alur nalar model dalam merumuskan kesimpulan prediksi kelas emosinya [41].