BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada tahun 2021, Adriansyah Dwi Rendgraha, Moch. Arif Bijaksana, dan Ade Romadhony melakukan sebuah studi yang berjudul "Pendekatan Metode Transformer untuk Deteksi Bahasa Kasar dalam Komentar Berita Online Indonesia." Tujuan dari penelitian ini adalah untuk mengidentifikasi penggunaan bahasa kasar dalam komentar daring di Indonesia dengan memanfaatkan model Transformer, khususnya *Bidirectional Encoder Representations from Transformers* (BERT), serta model BERT Multilingual yang telah dilatih sebelumnya sebagai baseline. Sistem ini menerima input berupa teks komentar dan menghasilkan label untuk mengkategorikan komentar tersebut ke dalam kategori *Offensive*, Normal, atau *Non-Offensive*. (Rendragraha dkk., 2021). Penggunaan BERT memiliki kekurangan, yaitu model ini tidak menggunakan *preprocessing*, sehingga terjadi kesalahan dalam memahami teks. Berdasarkan hal tersebut, peneliti merancang sebuah metode yang menggabungkan *preprocessing* agar kata-kata dalam korpus teks dapat dipahami dengan lebih baik oleh model *Transformers*.

Pada tahun 2023, I Nyoman Purnama dan Ni Nengah Widya Utami melakukan penelitian yang berjudul "Implementasi Peringkas Dokumen Berbahasa Indonesia Menggunakan Metode Text to Text Transfer Transformer (T5)." Penelitian ini bertujuan untuk merangkum dokumen berita berbahasa Indonesia dengan menggunakan metode Transformer T5. Dalam proses pengerjaannya, penelitian ini dibagi menjadi tiga skenario, di mana perbedaan utama antara masing-masing skenario terletak pada tahap *preprocessing*. Pada skenario pertama, diterapkan proses *stemming*

dan penghilangan kata-kata yang tidak penting (*stopwords removal*). Skenario kedua hanya menerapkan *stemming* tanpa menghapus kata-kata yang tidak penting, sedangkan pada skenario ketiga, keduanya tidak diterapkan (Purnama & Utami, 2023). Penggunaan T5 memiliki kekurangan, yaitu memerlukan pre-trained berulang kali sehingga metode ini memerlukan waktu yang lama. Berdasarkan hal tersebut, sistem tersebut berbeda dengan sistem yang dirancang oleh penulis, di mana penulis merancang sistem yang berfokus pada penerjemahan Bahasa Indonesia ke Bahasa Papua daerah Kokas menggunakan Metode Arsitektur Transformer dengan pre-trained yang tidak memerlukan waktu lama.

Pada tahun 2017, Zaenal Abidin melakukan penelitian yang berjudul "Penerapan *Neural Machine Translation* untuk Eksperimen Penerjemahan Secara Otomatis pada Bahasa Lampung-Indonesia." Penelitian ini bertujuan untuk menghasilkan 3.000 kalimat paralel dalam Bahasa Lampung (dialek Api) dan Bahasa Indonesia, serta menetapkan parameter model NMT untuk proses pelatihan data. Tahapan selanjutnya mencakup pembangunan model NMT dan pengujian performanya. Neural Machine Translation (NMT) merupakan pencapaian inovatif dalam teknologi penerjemahan mesin dengan menggabungkan penggunaan encoder, yang terdiri dari komponen berupa *recurrent neural network* (RNN) (Abidin, 2017). Penggunaan RNN dalam model NMT memiliki kekurangan yaitu kurangnya penerapan konsep *out-of-vocabulary* (OOV) sehingga jika ada kalimat yang mengandung OOV penelitian akan gagal dalam menerjemahkan. Atas dasar itu sistem tersebut peneliti mengumpulkan korpus teks dengan menerapkan konsep OOV.

Pada tahun 2022, Yustiana Fauziyah, Ridwan Ilyas, Fatan Kasyidi. Melakukan penelitian berjudul "Mesin Penerjemah Bahasa Indonesia-Bahasa Sunda Menggunakan *Recurrent Neural Network*". Tujuan dari studi ini adalah untuk mengembangkan sebuah perangkat penerjemah yang mengalihkan bahasa Indonesia ke bahasa Sunda dengan menerapkan pendekatan NMT. Dalam penelitian ini, digunakan arsitektur *Encoder-Decoder* yang memanfaatkan teknik RNN LSTM untuk menerjemahkan teks dari bahasa Indonesia ke bahasa Sunda (Fauziyah dkk., 2022). Penggunaan RNN memiliki kekurangan yaitu mempunyai akurasi yang kurang jika metode *attention* tidak diterapkan pada RNN. Atas dasar itu sistem tersebut berbeda dengan sistem yang dirancang oleh penulis, dimana penulis hanya menggunakan metode Arsitektur untuk melakukan suatu penerjemahan otomatis, yang di dalamnya sudah terdapat *attention.*

Pada tahun 2023, Alda Dwi Meilinda, Herry Sujaini, Novi Safriadi melakukan suatu penelitian yang berjudul "*Pivot Language* Bahasa Melayu Pontianak ke Bahasa Bugis Menggunakan *Neural Machine Translation*". Penelitian ini bertujuan membuat penerjemahan bahasa Melayu Pontianak ke bahasa Bugis dengan memanfaatkan *Neural Machine Translation*. Penelitian ini mengaplikasikan metode Arsitektur *Transformers* serta menambahkan pendekatan *Pivot Language* berfungsi sebagai bahasa penghubung untuk menerjemahkan dari bahasa sumber menuju bahasa sasaran. Dengan menggabungkan beberapa metode yang digunakan hasil dari tingkat akurasi penerjemahan meningkat dari hasil sebelumnya yang tidak menggunakan *Pivot Language*(Meilinda dkk., 2023). Penggunaan *pivot language* memiliki kekurangan yaitu metode ini digunakan sebagai penengah dari bahasa daerah ke bahasa daerah

lainnya, penggunaan metode ini diharuskan membuat korpus teks tambahan (bahasa terjemahan) agar hasil evaluasi yang dihasilkan meningkat. Perbedaan dengan sistem yang akan dilakukan, penulis merancang suatu terjemahan menggunakan metode *Transformers* tanpa menggunakan *Pivot Language* sehingga penerjemahan yang dilakukan tidak memerlukan suatu perantara Bahasa.

Berdasarkan Review jurnal diatas, research gap pada penelitian ini

1. Dalam penelitian ini, dilakukan pengujian terhadap jumlah data dan jumlah *epoch* yang bertujuan untuk mengetahui skor *bleu* selama proses *training*
2. Pada saat proses *training*, model yang digunakan adalah *Arsitektur Transformers* yang dikembangkan menggunakan *Framework Marian,*

2.2 Teori Terkait

2.2.1 *Machine Learning*

Pembelajaran mesin, yang menjadi bagian dari kecerdasan buatan, adalah ilmu yang berkaitan dengan pengembangan algoritma dan model statistic yang memungkinkan sistem komputer untuk belajar dari pola dan inferensi tanpa memerlukan instruksi secara eksplisit. Dengan menggunakan algoritma *machine learning*, sistem komputer dapat memproses data historis dalam jumlah besar dan mengidentifikasi pola-pola tertentu (Ott dkk., 2018). Hal ini memungkinkan sistem untuk memprediksi hasil dengan lebih akurat berdasarkan serangkaian *input* tertentu. Selain itu, *machine learning* banyak digunakan untuk menyelesaikan berbagai kategori utama, yaitu pembelajaran terarah, pembelajaran tidak terarah, dan pembelajaran penguatan (Roihan dkk., 2020).

2.2.2 *Teks Mining*

Text mining adalah ekstraksi menarik dari sebuah teks bebas atau teks tidak terstruktur dan mencakup penggalian informasi, pengklasifikasi teks, pengelompokan teks, penggalian entitas, relasi, dan ekstraksi peristiwa lainnya. Sebelum melakukan *text mining*, langkah awal yang harus dilakukan yaitu mempersiapkan data yang akan dianalisis. Tahap *text preprocessing* merupakan tahap pengolahan data yang sebelumnya tidak berstruktur menjadi data yang berstruktur (Adi dkk., 2018).

Pada tahap ini teks disusun mejadi teks terstruktur dan bagian – bagian yang tidak diperlukan dihilangkan dari teks hingga siap untuk diproses lebih lanjut. Berikut tahapan preprocessing teks yang digunakan pada penelitian ini:

1. *Case Folding:* proses mengubah semua huruf besar menjadi huruf kecil untuk menyamakan kata yang sejenis tetapi memiliki perbedaan kapitalisasi (Wahyuni dkk., 2017).
2. *remove punctuation*: menghilangkan tanda baca seperti titik, koma, tanda tanya, dan simbol lainnya yang tidak diperlukan.
3. *Remove whitespace*: Menghapus spasi yang berlebihan dalam teks
4. *Remove Stopword*: Menghilangkan kata – kata yang tidak bermakna atau tidak relevan, seperti ‘dan’, ‘atau’, ‘adalah’, yang tidak memberikan kontribusi berarti pada analisis teks.
5. *Lemmatization*: Menghilangkan kata ke bentuk dasarnya dengan mempertimbangkan struktur morfologis dan aturan tata bahasa. Sebagai contoh, kata “melihat”, “melihatkan”, dan “terlihat” akan dikembalikan ke bentuk dasar “lihat” (Supriyati & Iqbal, 2018).

6. *Tokenizing*: Proses pemecehan teks menjadi bagian – bagian kecil yang disebut token, berdasarkan spasi atau tanda penghubung. Tokenisasi ini membantu dalam mempermudah pemrosesan teks selanjutnya (Supriyati & Iqbal, 2018).

Pada penelitian ini, tahapan *text preprocessing* yang digunakan meliputi:

Case folding yaitu mengubah semua huruf menjadi huruf kecil untuk konsistensi data, *remove punctuation* yang berfungsi untuk menghilangkan tanda baca yang tidak diperlukan, *remove whitespace* yang menghapus spasi berlebihan dalam teks, serta *tokenizing* yaitu memecah teks menjadi bagian-bagian kecil yang disebut token berdasarkan spasi atau tanda penghubung agar lebih mudah diproses lebih lanjut

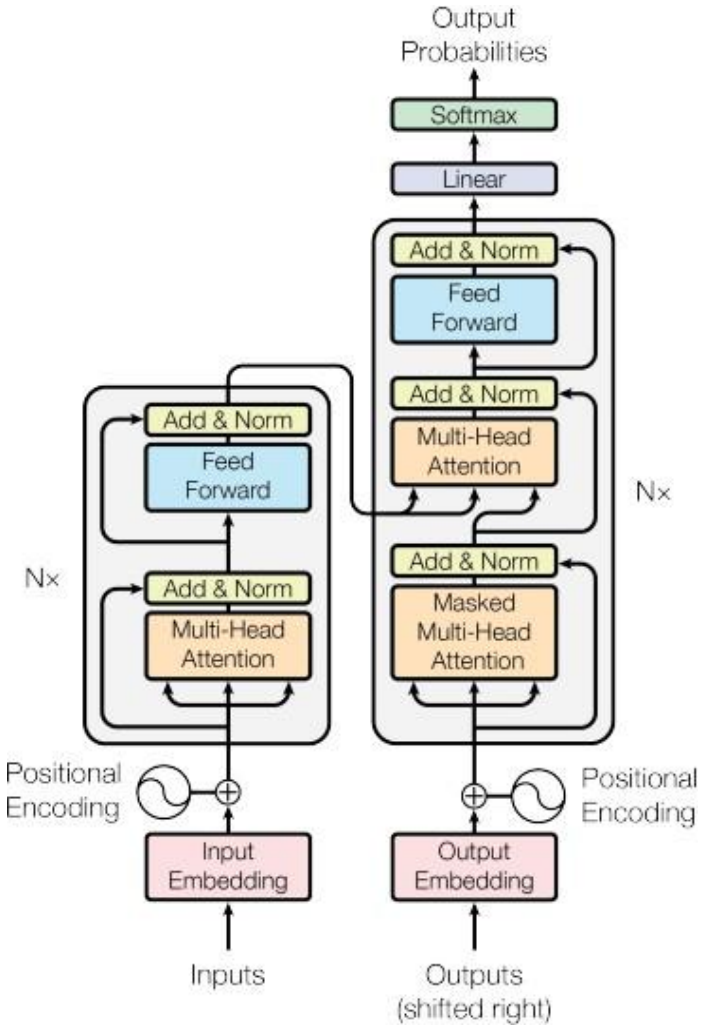
2.2.3 Arsitektur Transformers

Arsitektur Transformers adalah salah satu inovasi yang sangat penting dalam pengembangan pemrosesan bahasa alami dan tugas-tugas pemrosesan berbasis urutan lainnya. Metode ini memungkinkan mesin untuk mengatasi pemahaman bahasa alami, terjemahan mesin, pengenalan entitas berbasis teks, dan berbagai tugas lainnya dengan sangat baik. Transformers juga merupakan model pertama yang hanya mengandalkan perhatian diri untuk memperoleh representasi kalimat masukan dan keluaran selama proses pelatihan (Vaswani dkk., 2017). Transformer mengikuti keseluruhan arsitektur ini dengan menggunakan lapisan self-attention yang ditumpuk dan lapisan point-wise yang terhubung sepenuhnya untuk encoder dan decoder, yang ditunjukkan pada Gambar 2.1 Arsitektur Transformers

2.2.3.1 Encoder dan Decoder

Transformers terdiri dari dua komponen utama, yaitu *encoder* dan *decoder*.

Encoder memiliki dua *sub*-layer, pertama adalah *multi-head attention*, kedua adalah *fully-connected feed-forward network* (Yin, 2020). *Multi-head attention layer* membantu *encoder* menfokuskan perhatian pada kata–kata tertentu dan memahami konteks keseluruhan dari masukan. *Feed-forward network* kemudian menghasilkan vector yang diteruskan ke *decoder*.



Gambar 2.1 Arsitektur Transformers

Arsitektur **Decoder** mirip dengan *encoder,* namun ditambahkan *sub-layer* ketiga yang disebut *masked multi-head attention*. Lapisan ini berfungsi untuk menyembunyikan beberapa vektor yang dihasilkan oleh *encoder*, sehingga tidak hanya mencegah kebocoran informasi tetapi juga memeriksa output yang dihasilkan. *Input* awal dari *decoder* yaitu *token start*, yang menghindari kekosongan dalam pergeseran *output*. Selain itu, *token end* digunakan untuk menandai akhir dari kalimat, dan *decoder* tidak akan memproses *token end* tersebut

2.2.3.2 Self-Attention

Self-attention merupakan mekanisme penting dalam arsitektur *transformers* yang memungkinkan model untuk memperhatikan bagian tertentu dari input teks selama proses penerjemahan. Mekanisme ini membantu model fokus pada kata-kata tertentu yang relevan, berdasarkan konteksnya.

Proses *self-attention* melibatkan tiga komponen utama yaitu *query*, *key*, dan *value* dimana setiap kata dalam kalimat direpresentasikan dalam bentuk vektor. Mekanisme ini memungkinan model menghitung bobot atau perhatian pada setiap kata dalam kalimat dan menentukan seberapa relevan suatu kata dalam hubungannya dengan kata-kata lainnya di seluruh kalimat.

2.2.3.2.1 Scarled Dot-Product Attention

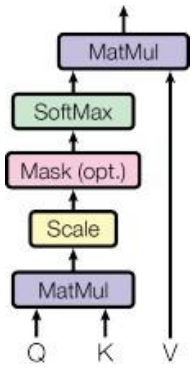
Fungsi utama dari *scarled dot-product attention* adalah memetakan *query* dan *key* ke dalam *output* yang dihitung dari nilai *weigthed sum* atau penjumlahan berbobot dari *values*. Proses dimulai dengan mengalikan *query* dengan *key*, kemudian hasil perkaliannya dibagi dengan akar dari dimensi *query* $\sqrt{d_k}$, sebelum melalui proses

sofmax untuk mendapatkan bobot atensinya. Setelah itu, bobot-bobot ini dikalikan dengan *value* untuk menghasilkan *output* akhir.

Notasi yang digunakan untuk fungsi ini adalah:

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

Proses ini memastikan bahwa setiap kata dalam *input* dapat menyesuaikan atensinya terhadap kata-kata lain dengan bobot yang sesuai, berdasarkan relevansinya dalam kalimat.

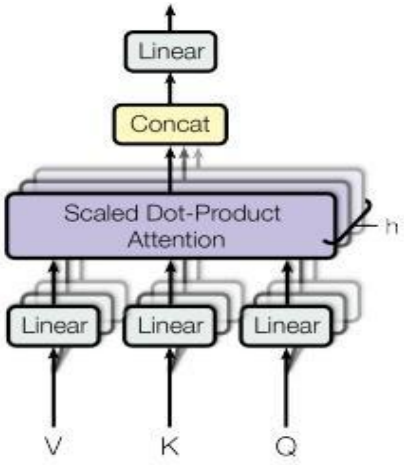


Gambar 2.2 Scarlet dot-product attention

2.2.3.2.2 Multi-Head Attention

Multi-head attention merupakan pengembangan dari mekanisme *self-attention* yang memungkinkan model untuk memberikan perhatian pada berbagai bagian input secara simultan. Alih-alih hanya menghasilkan satu set *query*, *key*, dan *value*, *multi-head attention* membuat beberapa set secara parallel. Setiap set menghasilkan *attention* yang berbeda, memungkinkan model untuk memproses berbagai aspek dari

kalimat *input* secara bersamaan. Setelah itu, hasil dari setiap *head* digabungkan dan diteruskan ke lapisan *feed-forward* (Vaswani dkk., 2017). Vaswani menggunakan 8 *head* dalam eksperimennya, sehingga menghasilkan delapan *matrik attention* yang berbeda, yang kemudian digabungkan sebelum diteruskan ke *feed-forward network*



Gambar 2.3 Multi-head Attention

2.2.3.3 Position-wise Feed-Forward Network

Pada arsitektur *transformers*, setiap lapisan *encoder* dan *decoder* memiliki jaringan *feed-forward* yang beroperasi dengan kondisi independent pada setiap posisi dalam urutan *input*. *Feed-forward nerwork* ini terdiri dari dua lapisan transformasi linier yang dipisahkan oleh fungsi aktivasi *rectified linier unit* (ReLU). *Input* dari lapisan *self-attention* diproses oleh *feed-forward network* untuk mengekstrak informasi lebih lanjut (Vaswani dkk., 2017).

Notasi umum *feed-forward network* yaitu:

$$FFN(x) = (0, xW_1 + b_1)\, W_2 + b_2$$

Di sini, x adalah *input vektor*, sementara $W1$ dan $W2$ adalah bobot dari dua lapisan linier, dan $b1$ serta $b2$ adalah bias masing-masing lapisan. *Output* dari *feed-forward network* ini kemudian diteruskan ke lapisan berikutnya dalam bentuk model.

2.2.3.4 Embedding dan Softmax

Embedding digunakan sebagai konversi kata-kata dari teks *input* menjadi *vektor* berdimensi tetap, yang dapat diproses oleh model. Setiap kata dalam teks diubah menjadi *vektor* dengan dimensi d_{model}, yang menunjukkan representasi numerik dari kata tersebut. Setelah itu, output dari *decode* pada *Trasnformers* diubah kembali menjadi probalitasi melalui lapisan *softmax*. Fungsi *softmax* mengonversi *output* menjadi distribusi probabilitas untuk memprediksi token berikutnya dalam urutan terjemahan (Vaswani dkk., 2017).

Selama proses pelatihan, *embedding* digunakan untuk menangani *input* dan *output*, dengan bobot *embedding input* dan *output* sering kali dibagi untuk menghemat parameter. *Transformers* linier dan fungsi *softmax* bersama-sama membantu model memprediksi token yang paling mungkin berdasarkan *input* yang telah diproses oleh lapisan sebelumnya.

2.2.3.5 Positional Encoding

Model ini tidak mengandalkan RNN (*Recurrent Neural Network*) atau CNN (*Convolution Neural Network*). Untuk menangani urutan kata dalam kalimat, vektor *positional encoding* memiliki dimensi yang sebanding dengan vektor dari *embedding layer*. Selanjutnya, vektor ini digabungkan dengan vektor yang dihasilkan dari *embedding layer* (Vaswani dkk., 2017).

Karena arsitektur *Transformers* tidak memiliki elemen berurutan seperti pada RNN, *positional encoding* ditambahkan untuk memberikan model informasi tentang urutan kata dalam kalimat. *Positional encoding* menggunakan fungsi sinus dan cosinus dengan frekuensi berbeda untuk menghasilkan vektor posisi yang kemudian ditambahkan ke *embedding*, memungkinkan model untuk mempertahankan informasi tentang posisi relatif setiap kata dalam kalimat (Vaswani dkk., 2017).

Model ini menggunakan fungsi sin dan cos dari frekuensi yang berbeda

$$PE_{(pos,2i)} = sin\ sin\ (pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = sin\ sin\ (pos/10000^{2i/d_{model}})$$

dimana pos adalah posisi kata dalam urutan, dan i menunjukkan dimensi vektor. Dengan adanya *positional encoding*, *transformers* dapat mempertimbangkan urutan kata, meskipun tidak menggunakan mekanisme berulang seperti pada RNN.

2.2.4 MarianMT

MarianMT adalah sebuah framework yang dikembangkan untuk melatih model Neural Machine Translation yang efisien dan skalabel (Aulamo & Boggia, 2023). MarianMT dikembangkan oleh tim Microsoft dan menggunakan arsitektur berbasis Transformers, yang telah terbukti sangat efektif dalam berbagai tugas terjemahan mesin (Abidin, 2017). Arsitektur MarianMT memanfaatkan metode self-attention dan multi-head attention yang memungkinkan model untuk fokus pada bagian-bagian tertentu dari input ketika menghasilkan output (Aulamo & Boggia, 2023). MarianMT mendukung pemrosesan batch dan dapat berjalan dengan baik pada GPU, menjadikannya pilihan yang kuat untuk tugas-tugas penerjemahan yang membutuhkan kinerja tinggi (Purnama & Utami, 2023).

Keunggulan utama MarianMT terletak pada kemampuannya untuk menangani berbagai bahasa dengan sumber daya terbatas dan kemampuannya untuk mengakomodasi model-model besar tanpa memerlukan sumber daya komputasi yang berlebihan (Aulamo & Boggia, 2023). Framework ini juga mendukung berbagai teknik optimasi seperti back-translation dan fine-tuning, yang membantu meningkatkan kualitas terjemahan terutama pada pasangan bahasa dengan data terbatas (Abidin, 2017).

2.2.5 BLEU

BLEU (*bilingual evaluation understudy*) merupakan salah satu metrik evaluasi yang paling sering digunakan untuk mengukur kualitas hasil terjemahan mesin. Algoritma ini menghitung seberapa mirip hasil terjemahan otomatis dengan terjemahan referensi. Skor BLEU mengukur akurasi statistic berbasis *n-gram*, dengan mempertimbangkan panjang kalimat hasil terjemahan dan kalimat referensi. BLEU juga menggunakan *penalti brevity* (BP) untuk mengurangi skor jika terjemahan terlalu pendek dibandingkan dengan referensi, sehingga menjaga keseimbangan antara akurasi dan panjang terjemahan (Abidin, 2017).

Proses perhitungan BLEU dimulai dengan menghitung *precision* dari *n-gram* yang cocok antara hasil terjemahan dan referensi. Formula umum untuk menghitung BLEU adalah sebagai berikut:

$$BP_{BLEU} = \{1, e\,(1 - \frac{r}{c}), if\ if\ c > r\ c\ \leq r\}$$

$$p_n = \frac{C \in \sum\{Candidates\}\ \sum n - gram \in C\ \ Count_{dip}(n - gram)}{C' \in \sum\{Candidates\}\ \sum n - gram' \in C'\ \ Count\ (n - gram')}$$

$$BLEU = BP_{BLEU} \cdot exp\left(\sum_{n=1}^{N} w_n \, log \, log \, p_n\right)$$

Keterangan:

c = jumlah kata dalam hasil terjemahan otomatis.

BP = brevity penalty (penalti singkat).

Pn = jumlah *n-gram* dalam hasil terjemahan yang cocok dengan referensi dibagi dengan jumlah *n-gram* dalam hasil terjemahan.

Pn = *modified precision score* (skor presisi yang di modifikasi).

Wn = 1/N (standar nilai N untuk BLEU yaitu 4).

Semakin tinggi skor BLEU, semakin mendekati hasil terjemahan mesin dengan terjemahan referensi, yang berarti terjemahan tersebut dianggap berkualitas lebih baik. Skor ini berkisar dari 0 hingga 1, dimana 1 adalah hasil terjemahan yang identik dengan referensi (Fauziyah dkk., 2022). Skor BLEU pada umumnya dianggap baik jika berada di atas 30, sementara skor di atas 50 menunjukkan bahwa hasil terjemahan sudah sangat mendekati kualitas manusia (Lavie, 2010). Namun, perlu diingat bahwa meskipun BLEU memberikan indikasi kualitas, skor ini tidak selalu mencerminkan aspek linguistik yang lebih kompleks, seperti keakuratan konteks dan kealamian bahasa.

BLEU dirancang untuk mengukur akurasi dalam berbagai tugas penerjemahan, dan penggunaanya yang luas menjadikannya standar dalam menilai performa model terjemahan otomatis (Vaswani dkk., 2017). Penggunaan beberapa terjemahan referensi akan meningkatkan skor BLEU karena lebih banyak variasi terjemahan yang dianggap benar, sehingga memberikan penilaian yang lebih akurat terhadap model (Shaw et al., 2018).

2.2.6 Django

Django, *framework web* tingkat tinggi berbasis *Python*, dirancang untuk mempercepat pengembangan aplikasi web yang aman dan skalabel. Django memungkinkan pengembangan untuk membangun aplikasi web dengan cepat tanpa mengorbankan kinerja dan keamanan, dengan mengadopsi pola arsitektur Model-template-view (MTV) yang mirip dengan model-view-controller (MVC) pada framework lainnya.

Sistem manajemen url yang fleksibel dan fitur Django Admin untuk antarmuka administrative membuat Django sangat efisien dan *user-friendly*. Django unggul dalam kinerja jumlah permintaan dan efisiensi penggunaan dan sumber daya, menjadikannya pilihan yang ideal untuk aplikasi web yang memerlukan skalabilitas tinggi (Amarulloh dkk., 2023). Selain itu Django memiliki waktu respon cepat dan penggunaan CPU serta memori yang rendah, membuatnya sangat cocok untuk pengembangan aplikasi web berbasis mobile dan server (Saputra, 2018).

2.2.7 Struktur Bahasa

Struktur bahasa Indonesia mencakup berbagai unsur yang membentuk kalimat. Berikut adalah penjelasan lebih rinci tentang struktur bahasa Indonesia:

1. **Kalimat dasar:**
    - Setiap kalimat dasar terdiri dari subjek, predikat, dan objek.
    - *Contoh:* "Ani (subjek) makan (predikat) nasi (objek)."
2. **Subjek (Pelaku):**
    - Subjek merupakan orang, hewan, atau benda yang melakukan tindakan dalam kalimat.
    - *Contoh:* "Mereka (subjek) bermain (predikat) di taman."
3. **Predikat (Pekerjaan atau Tindakan):**

- Predikat adalah kata kerja yang menyatakan tindakan atau keadaan.
- *Contoh:* "Dia (subjek) tidur (predikat)."

4. **Objek (Sasaran atau Penerima Tindakan):**
    - Objek adalah orang, hewan, atau benda yang menjadi sasaran atau penerima tindakan.
    - *Contoh:* "Saya (subjek) menyapu (predikat) lantai (objek)."
5. **Pelengkap:**
    - Pelengkap memberikan informasi tambahan tentang subjek atau objek.
    - *Contoh:* "Bunga (subjek) itu wangi (pelengkap)."
6. **Keterangan:**
    - Keterangan memberikan informasi tambahan tentang waktu, tempat, cara, alasan, atau kondisi.
    - *Contoh:* "Mereka (subjek) bermain (predikat) di pantai (keterangan)."
7. **Penghubung:**
    - Penghubung digunakan untuk menghubungkan kata, frasa, atau klausa dalam kalimat.
    - *Contoh:* "Saya suka makan es krim, tetapi adik saya lebih suka cokelat."
8. **Klausa:**
    - Klausa adalah bagian dari kalimat yang memiliki subjek dan predikat sendiri.
    - *Contoh:* "Ketika hujan turun, kami tinggal di dalam rumah."
9. **Kalimat Majemuk:**
    - Kalimat majemuk terdiri dari dua atau lebih klausa yang saling terkait.
    - *Contoh:* "Saya suka membaca buku, dan teman saya suka menulis cerita."
10. **Tanda Baca:**
    - Tanda baca digunakan untuk memisahkan, menghubungkan, atau memberikan penekanan dalam kalimat.
    - *Contoh:* "Maaf, saya tidak bisa datang besok!"

Penting untuk memahami dan mengaplikasikan struktur bahasa Indonesia agar komunikasi lebih jelas dan efektif (Sulistiyo, 2016). Dengan memahami struktur ini, Anda dapat merangkai kalimat dengan jelas sehingga mudah dipahami oleh orang lain.

Berikut adalah contoh struktur bahasa indonesia dan bahasa kokas

Saya akan pergi ke kantor sekarang

Saya (subjek) akan pergi (predikat) ke kantor (objek) sekarang (keterangan)

Yai e mau eti kantor madige

Yai (subjek) e mau (predikat) kantor (objek) madige (keterangan)

2.2.8 Linguistik dan Penerjemahan

Linguistik adalah ilmu yang mempelajari bahasa secara ilmiah dan sistematis, mencakup berbagai aspek mulai dari fonologi (kajian tentang bunyi bahasa), morfologi (struktur dan pembentukan kata), sintaksis (struktur kalimat), semantik (kajian makna), hingga pragmatik (konteks dan situasi penggunaan bahasa) (Sukmana dkk., 2003). Setiap cabang ini memiliki fungsi yang berbeda namun saling melengkapi dalam memahami bahasa secara utuh.

Linguistik korpus berperan penting dalam kajian penerjemahan karena memungkinkan studi penerjemahan sebagai fenomena empiris. Korpus linguistik mendukung analisis penerjemahan dengan menyediakan data yang besar dan autentik, seperti frekuensi kata dan pola gramatikal, sehingga peneliti dapat mengidentifikasi ciri khas teks terjemahan (Ummah, 2019). Metodologi korpus memungkinkan penerapan analisis statistik dan kualitatif untuk mengungkap pola dalam bahasa terjemahan, seperti simplifikasi atau pergeseran makna.

Terdapat beberapa jenis korpus yang digunakan, termasuk korpus paralel, yang mencocokkan teks sumber dengan terjemahannya, dan korpus sebanding, yang berisi teks dalam bahasa yang berbeda namun memiliki kesamaan topik (Ummah, 2019). Penggunaan korpus paralel dan sebanding memungkinkan perbandingan lintas bahasa dan gaya dalam penerjemahan (Abidin & Permata, 2021). Selain itu, korpus umum dan khusus bisa digunakan sesuai dengan fokus penelitian, misalnya untuk mengkaji terminologi atau variasi gaya.

Dalam kajian penerjemahan, metode seperti konkordansi dan analisis frekuensi sering digunakan untuk melihat kecenderungan stilistik dan norma dalam teks terjemahan, yang dikenal sebagai "kode ketiga" atau bentuk bahasa yang unik dalam terjemahan. Hal ini menciptakan pemahaman bahwa terjemahan adalah tindakan komunikatif yang dipengaruhi oleh konteks sosial-budaya, dan bukan hanya upaya untuk mencapai kesepadanan dengan teks sumber(Abidin & Permata, 2021).