



LATAR BELAKANG





Penggunaan mobil sebagai alat transportasi masih menjadi pilihan utama bagi sebagian besar warga Indonesia. Didukung dengan penggunaan internet di Indonesia yang terus meningkat. Menurut Survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) tahun 2024, tingkat penetrasi internet di Indonesia mencapai 79,5%. Hal ini berarti sebanyak lebih dari 221 juta jiwa penduduk di Indonesia sudah terhubung ke internet. media sosial sebagai tempat masyarakat menyampaikan opini dapat menjadi sumber evaluasi berharga bagi para produsen mobil, salah satu caranya dengan melakukan analisis sentimen.

Pada penelitian ini menggunakan data lima merek mobil terlaris di Indonesia pada tahun 2023:











Data media sosial yang digunakan adalah:













Data diambil dengan interval waktu mulai bulan Januari 2023 sampai April 2024.





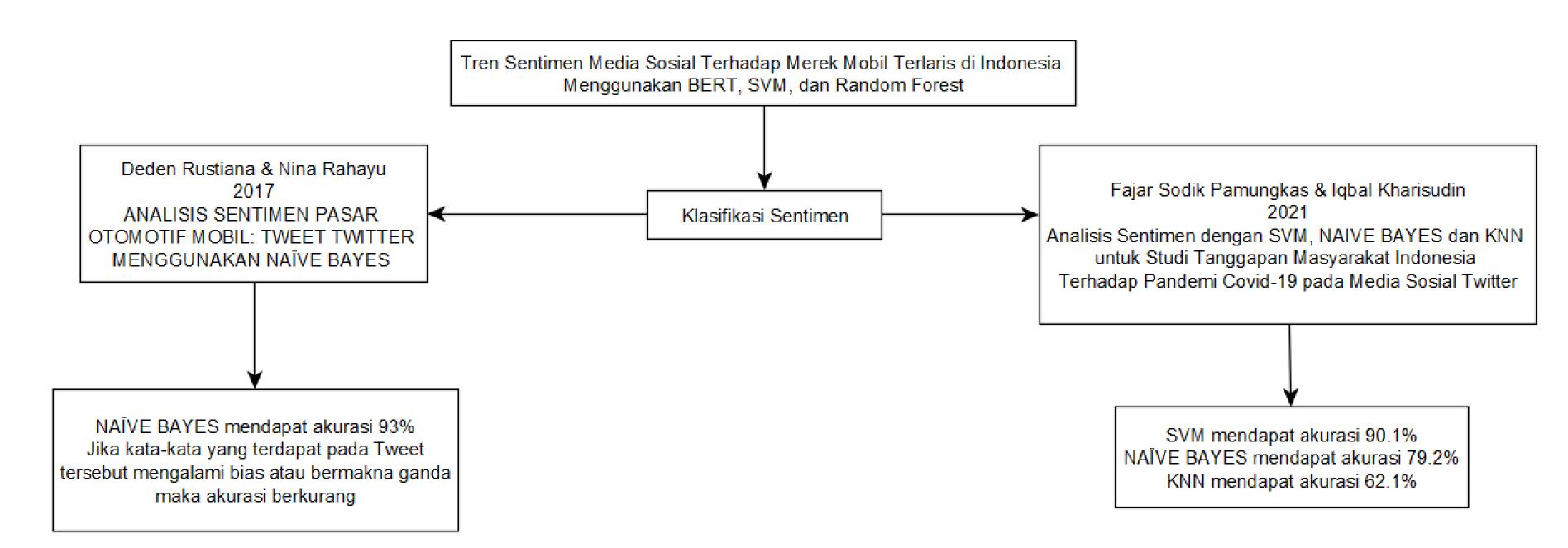






STUDI PUSTAKA





Namun pada penelitian diatas hanya menggunakan data dari twiter serta terdapat kekurangan metode Naive Bayes yang memiliki keterbatasan dalam memahami konteks dalam bahasa alami karena asumsi independensinya, oleh karena itu saya memilih BERT sebagai metode utama karena mampu menangkap makna yang lebih kompleks dan akurat dalam analisis sentimen melalui pemahaman konteks yang mendalam.







TUJUAN PENELITIAN



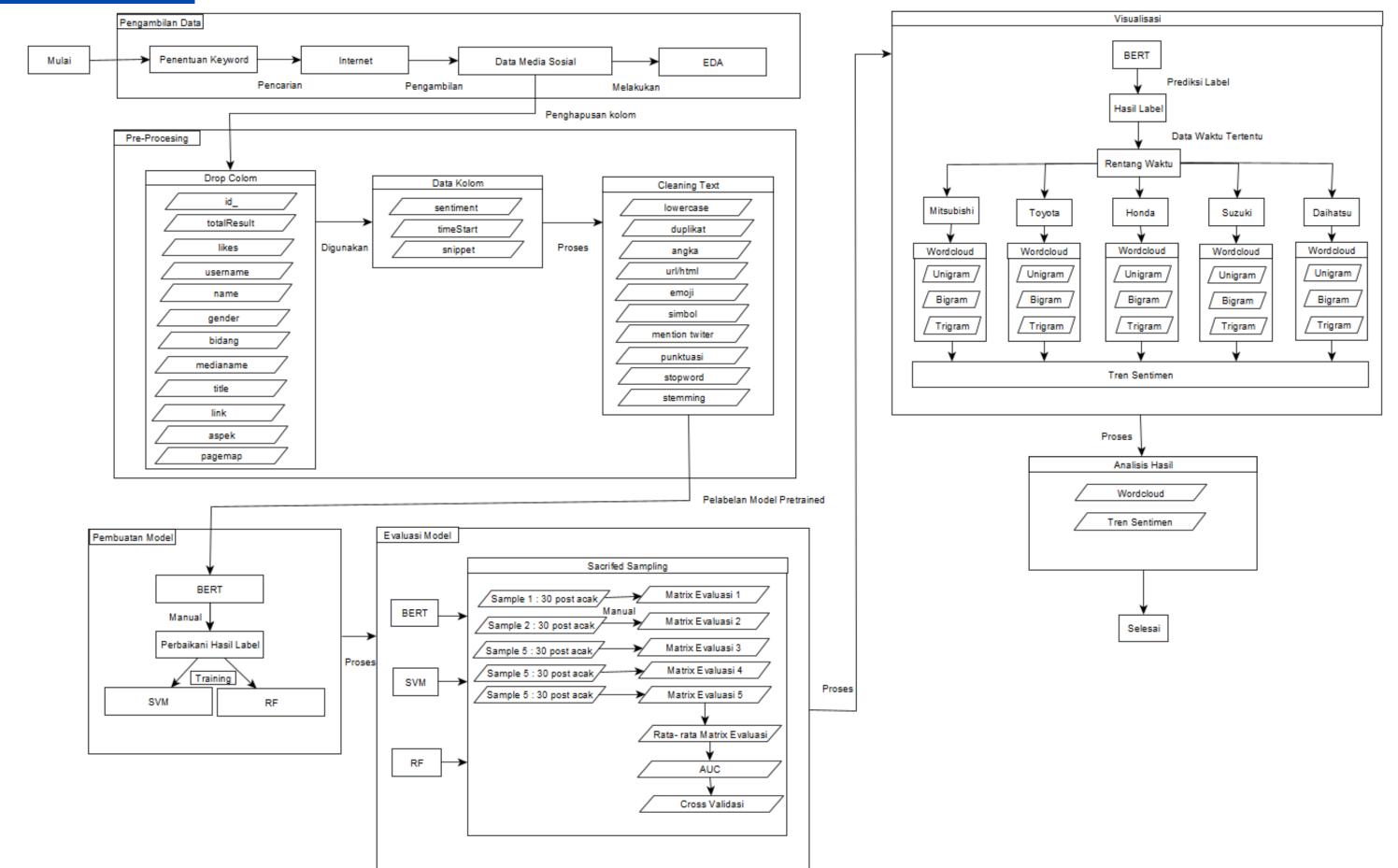
- Mengetahui model yang paling layak digunakan untuk kasus klasifikasi sentimen.
- Mengetahui kata yang mewakili sentimen positif dan negatif untuk setiap merek mobil
- Mengetahui tren sentimen positif dan negatif dari setiap merek mobil





ALUR PENELITIAN







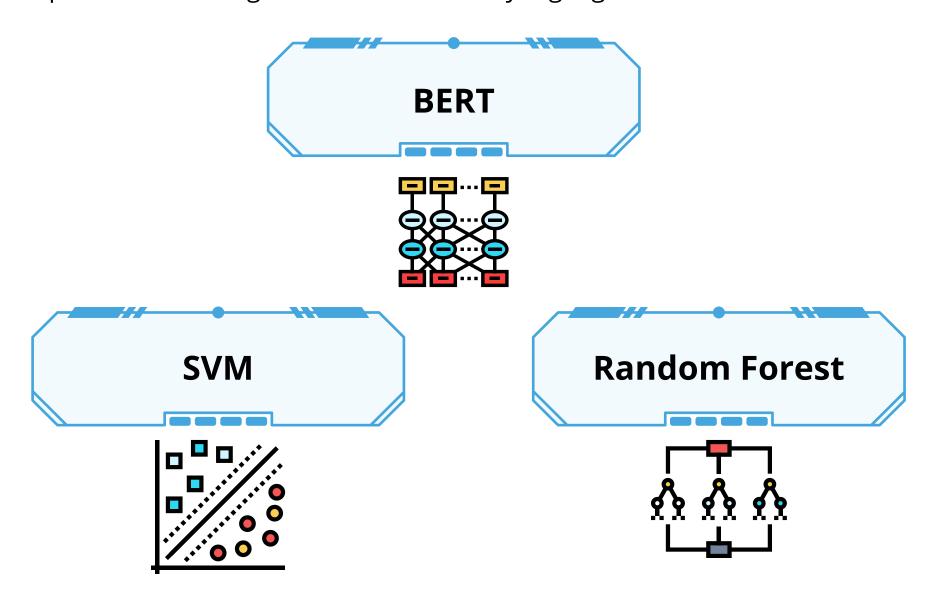


METODE





Metode utama yang digunakan dalam penelitian ini adalah model BERT khusus yang sebelumnya sudah dilatih untuk klasifikasi sentimen teks berbahasa Indonesia yaitu "mdhugol/indonesia-bert-sentiment-classification". Selain itu sebagai perbandingan digunakan juga metode SVM dan Random Forest untuk memberikan pandangan yang lebih komprehensif tentang keefektifan metode yang digunakan.



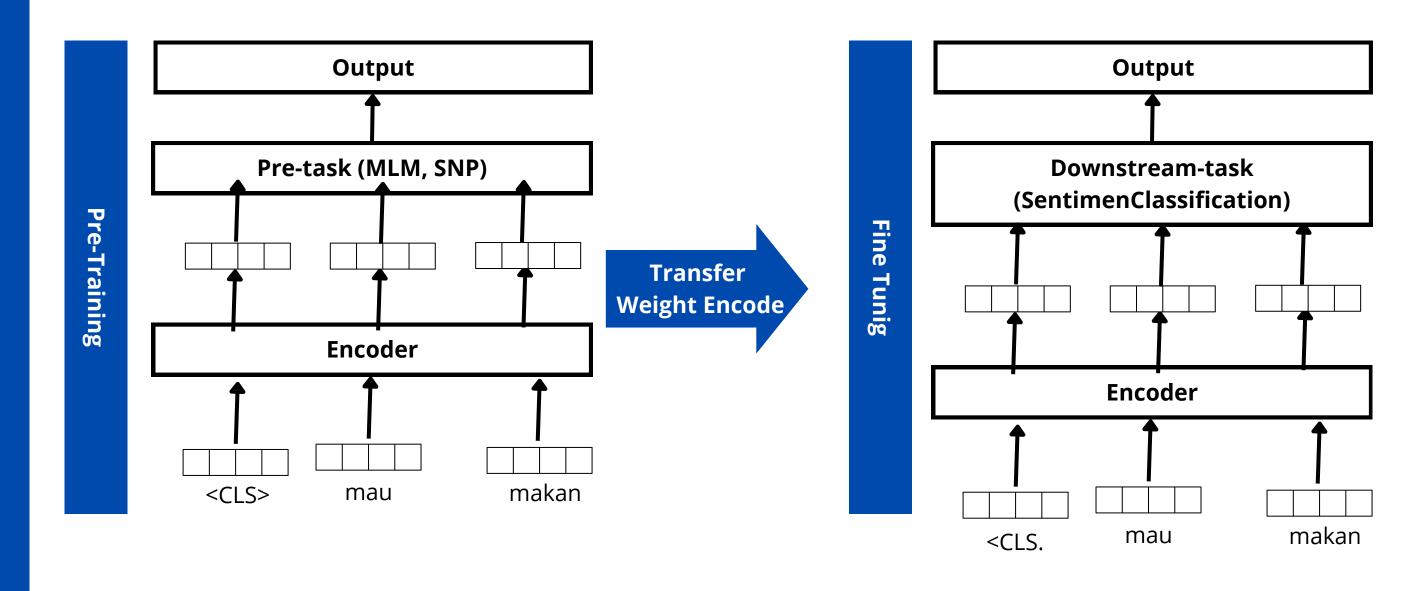




METODE



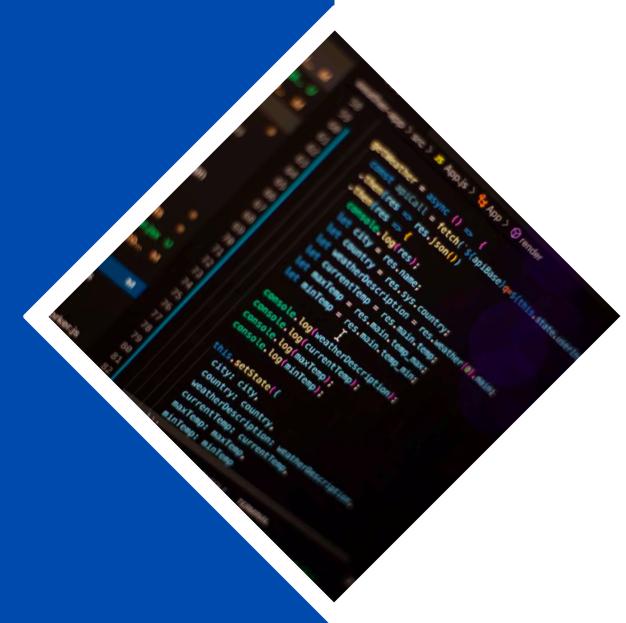






METODE

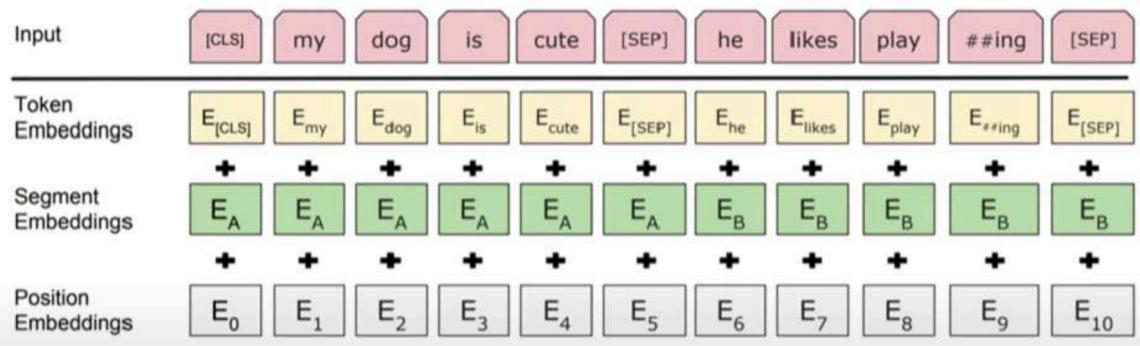




Pada model BERT pretrained mdhugol/indonesia-bert-sentiment-classification menggunakan dua pre-task utama yang sering digunakan dalam pretraining model BERT secara umum yaitu:

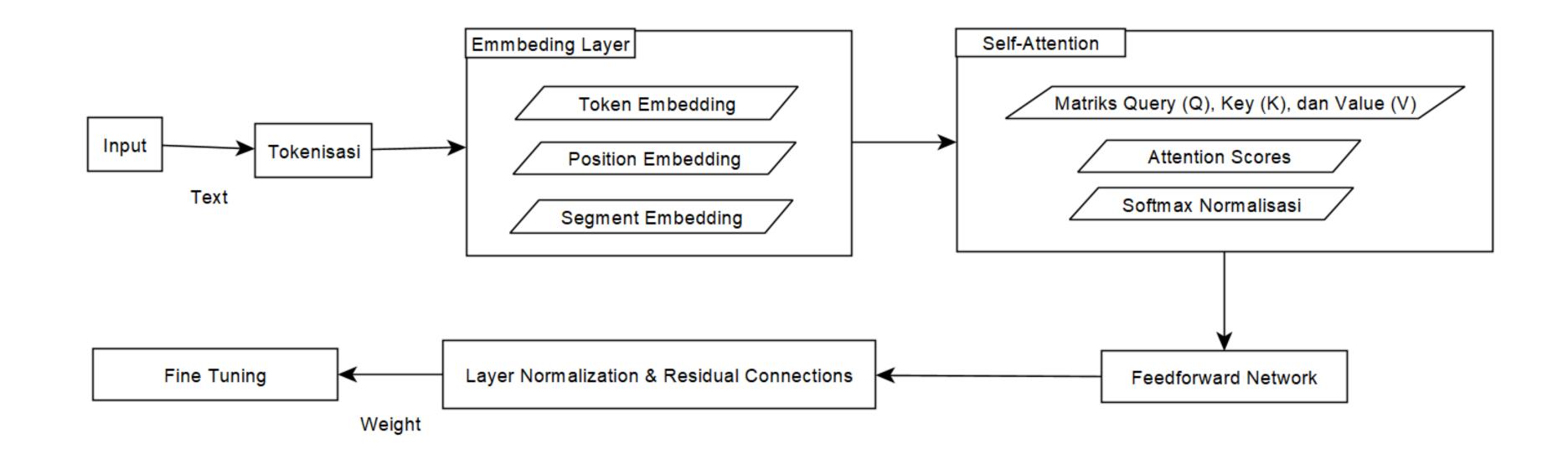
- Masked Language Modeling (MLM) untuk mempelajari representasi konteks kata dengan mengganti beberapa kata dalam sebuah kalimat dengan token khusus [MASK].
- Next Sentence Prediction (NSP) untuk mempelajari hubungan antar kalimat dalam teks dengan token pemisah [SEP].

Downstream-task yang digunakan untuk sentiment analysis



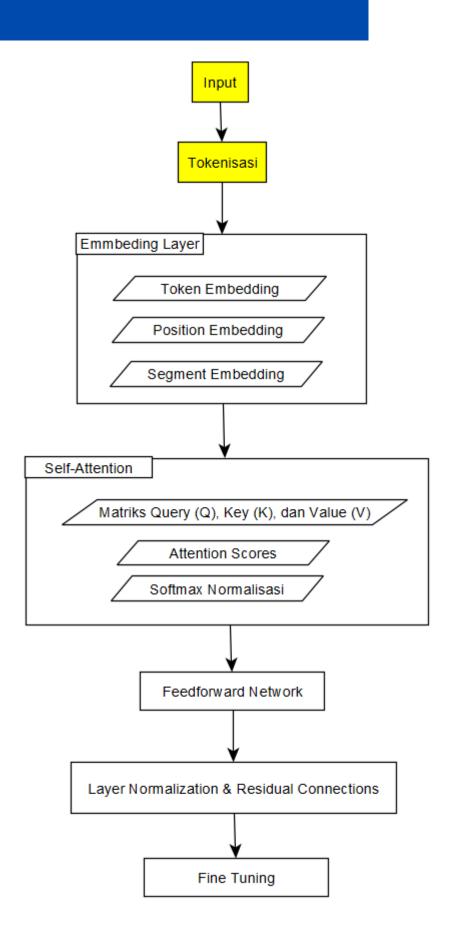












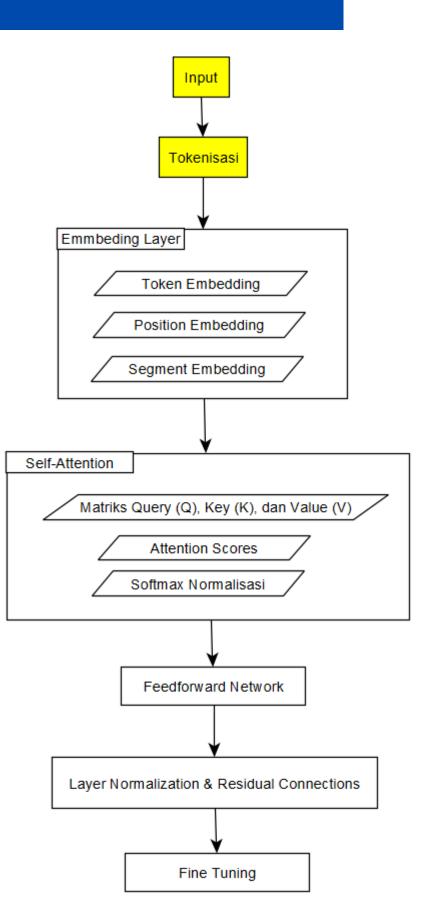
Operasi Matematika model BERT dengan kalimat aku mau makan:

Tahap pertama adalah memecah kalimat menjadi token dan mengonversinya ke indeks angka menggunakan tabel kosakata.

- Tambahkan token [CLS] (penanda awal) dan [SEP] (penanda akhir). Token: [CLS], Saya, mau, makan, [SEP]
- Gunakan tabel kosakata (lookup table) untuk mengonversi token menjadi angka. Misal: [CLS] = 101, Saya = 4567, mau = 7890, makan = 12345, [SEP] = 102
- Hasil Tokenisasi: [101, 4567, 7890, 12345, 102]







Operasi Matematika model BERT dengan kalimat aku mau makan:

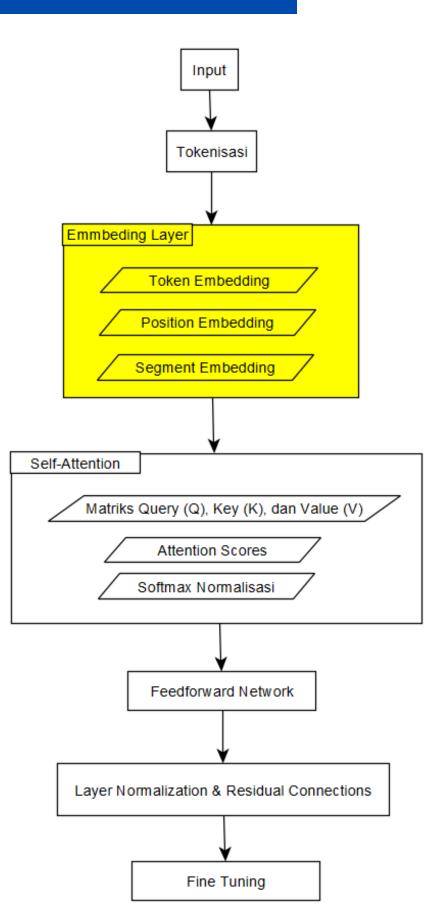
Tahap pertama adalah memecah kalimat menjadi token dan mengonversinya ke indeks angka menggunakan tabel kosakata.

- Tambahkan token [CLS] (penanda awal) dan [SEP] (penanda akhir). Token: [CLS], Saya, mau, makan, [SEP]
- Gunakan tabel kosakata (lookup table) untuk mengonversi token menjadi angka. Misal: [CLS] = 101, Saya = 4567, mau = 7890, makan = 12345, [SEP] = 102
- Hasil Tokenisasi: [101, 4567, 7890, 12345, 102]









Representasi Input (Embedding)

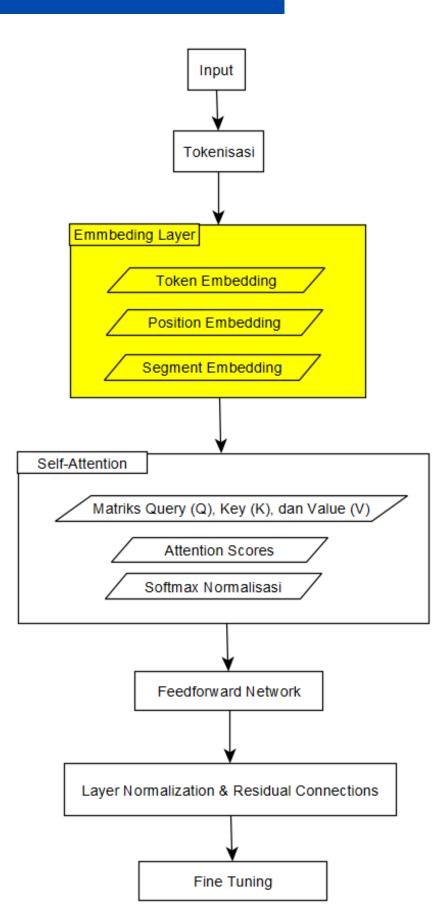
Kalimat: "aku mau makan" terdiri dari 3 token: ["aku", "mau", "makan"]. Dalam BERT, token-token ini akan diubah menjadi vektor embedding. Misalnya, kita menggunakan representasi embedding 2 dimensi untuk menyederhanakan.

- Token Embeddings:
 - Token("aku") = [0.5, 0.1]
 - Token("mau") = [0.2, 0.3]
 - Token("makan") = [0.4, 0.6]
- Segment Embeddings: Misalnya kita hanya memiliki satu kalimat, jadi semua token mendapatkan embedding yang sama:
 - Segment = [0, 0, 0]
- Position Embeddings: Berikan posisi token di urutan kalimat (misalnya posisi pertama, kedua, dan ketiga):
 - Position("aku") = [0.1, 0.2]
 - Position("mau") = [0.3, 0.4]
 - Position("makan") = [0.5, 0.6]



SENADA Seminar Nasional Sains Data





Representasi input untuk token pertama (aku) akan menjadi:

$$E_{\rm "aku"} = {\rm Token\ Embedding_{\rm "aku"}} + {\rm Segment\ Embedding} + {\rm Position\ Embedding_{\rm "aku"}}$$

$$= [0.5, 0.1] + [0, 0] + [0.1, 0.2] = [0.6, 0.3]$$

Begitu juga untuk token lainnya:

$$E_{\text{"mau"}} = [0.2, 0.3] + [0, 0] + [0.3, 0.4] = [0.5, 0.7]$$

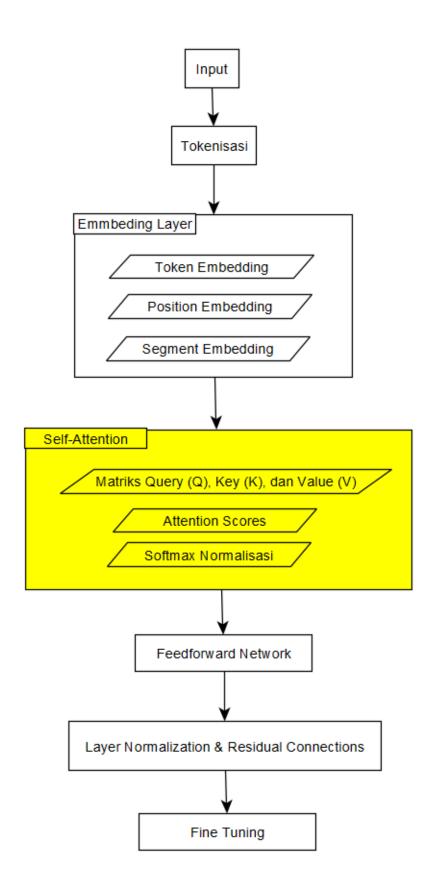
$$E_{
m `makan"} = [0.4, 0.6] + [0, 0] + [0.5, 0.6] = [0.9, 1.2]$$

Jadi, representasi input keseluruhan (X) adalah:

$$X = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix}$$







Self-Attention

Sekarang, kita akan menghitung self-attention menggunakan matriks **Query** (**Q**), **Key** (**K**), dan **Value** (**V**). Untuk kesederhanaan, kita asumsikan bahwa matriks bobot W_Q , W_K , dan W_V adalah matriks identitas 2x2.

Menghitung Q, K, dan V:

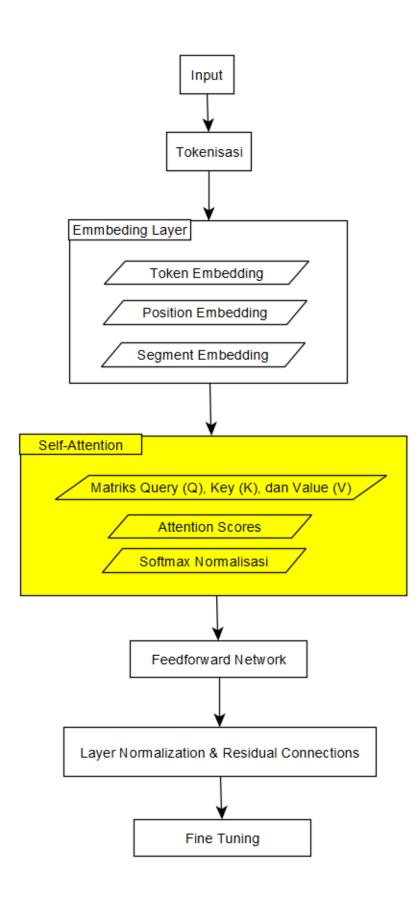
$$Q = XW_Q = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix} imes egin{bmatrix} 1 & 0 \ 0 & 1 \end{bmatrix} = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix}$$

$$K = XW_K = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix} imes egin{bmatrix} 1 & 0 \ 0 & 1 \end{bmatrix} = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix}$$

$$V = XW_V = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix} imes egin{bmatrix} 1 & 0 \ 0 & 1 \end{bmatrix} = egin{bmatrix} 0.6 & 0.3 \ 0.5 & 0.7 \ 0.9 & 1.2 \end{bmatrix}$$







Self-Attention

Menghitung Attention Scores:

Sekarang kita akan menghitung perhatian antar-token. Skor perhatian antara token i dan token j dihitung dengan rumus:

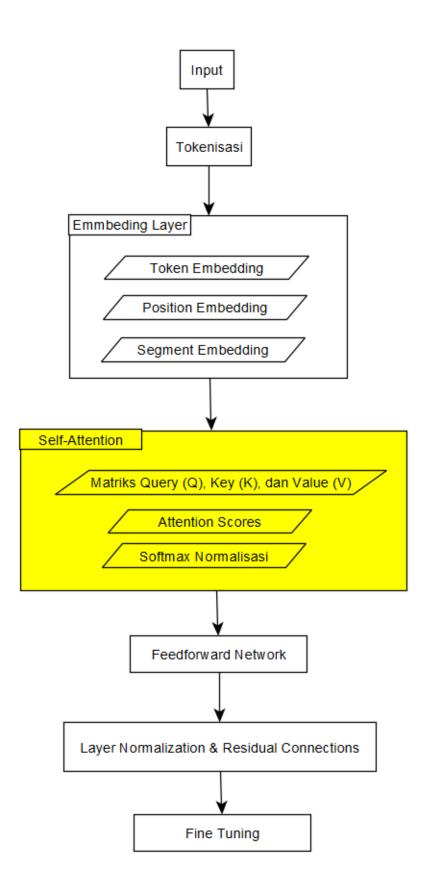
$$ext{Attention Score}_{ij} = rac{Q_i K_j^ op}{\sqrt{d_k}}$$

Dengan $d_k=2$ (dimensi embedding). Untuk token "aku" (i=1) dan "mau" (j=2):

$$\text{Attention Score}_{12} = \frac{[0.6, 0.3] \cdot [0.5, 0.7]}{\sqrt{2}} = \frac{0.6 \times 0.5 + 0.3 \times 0.7}{\sqrt{2}} = \frac{0.3 + 0.21}{1.414} \approx \frac{0.51}{1.414} \approx 0.36$$

Begitu juga dengan perhatian antara token lainnya.





Self-Attention

Misalnya, kita hitung softmax untuk token "aku":

- 1. Skor perhatian antara "aku" dan token lainnya:
 - Attention $Score_{12} = 0.36$ (antara "aku" dan "mau")
 - Attention $Score_{13} = 0.42$ (antara "aku" dan "makan")
 - Attention $Score_{11} = 0.25$ (antara "aku" dan "aku")
- 2. Eksponensial dari skor perhatian:

$$e^{0.36} pprox 1.433$$

$$e^{0.42}pprox 1.526$$

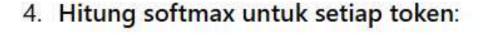
$$e^{0.25} \approx 1.284$$

3. Jumlahkan hasil eksponensial:

$$\sum e^{x_j} = 1.433 + 1.526 + 1.284 = 4.243$$







Menghitung Softmax: Softmax digunakan untuk mengubah skor perhatian menjadi distribusi probabilitas. Softmax untuk setiap token dilakukan sebagai berikut:

$$ext{Softmax}(x_i) = rac{e^{x_i}}{\sum e^{x_j}}$$

Dimana x_i adalah skor perhatian antara token i dan semua token lainnya, dan $\sum e^{x_j}$ adalah jumlah eksponensial dari semua skor perhatian untuk semua pasangan token.

Softmax untuk "aku" dengan "aku", "mau", "makan":

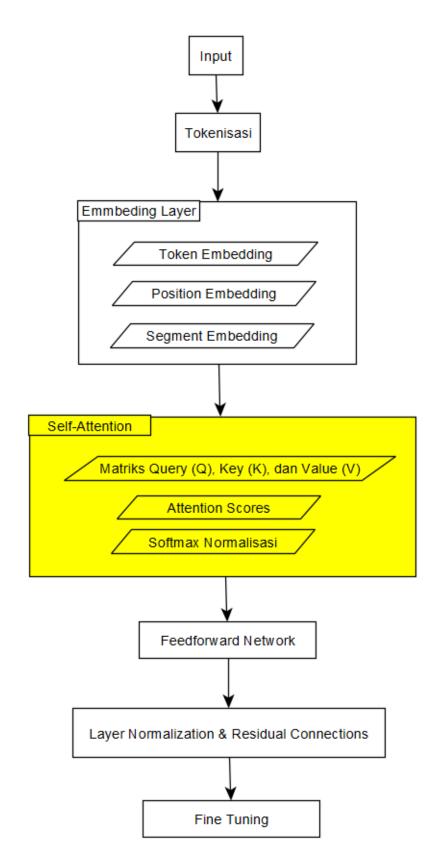
Softmax(aku) =
$$\frac{e^{0.25}}{4.243} = \frac{1.284}{4.243} \approx 0.303$$

Softmax untuk "aku" dengan "mau":

Softmax(mau) =
$$\frac{e^{0.36}}{4.243} = \frac{1.433}{4.243} \approx 0.337$$

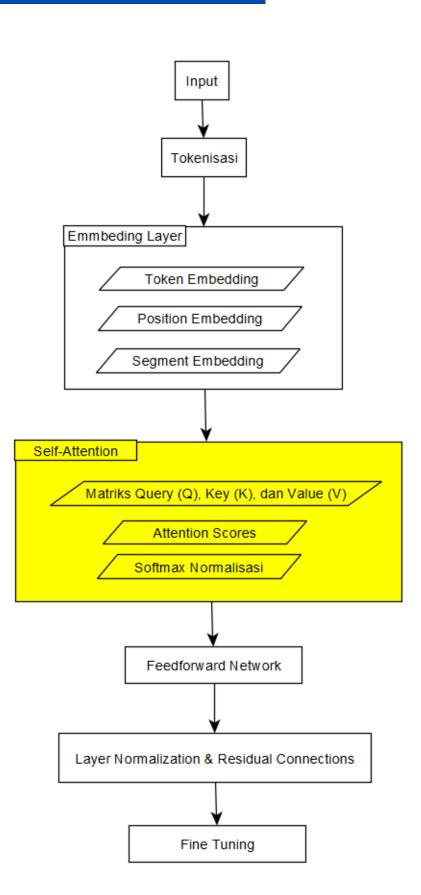
Softmax untuk "aku" dengan "makan":

Softmax(makan) =
$$\frac{e^{0.42}}{4.243} = \frac{1.526}{4.243} \approx 0.360$$







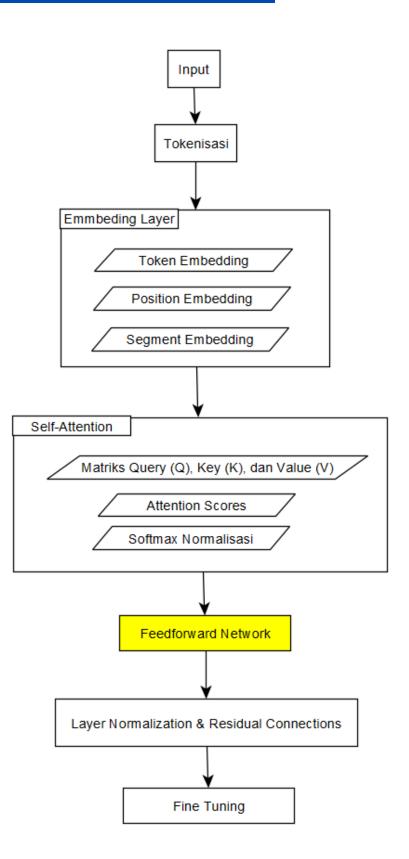


Hasil Softmax untuk token "aku" terhadap token lainnya: Softmax(aku)=[0.303,0.337,0.360]

Artinya, token "aku" akan memberikan bobot perhatian (attention weight) terhadap token "aku", "mau", dan "makan" masing-masing sekitar 30.3%, 33.7%, dan 36.0%.







Feedforward Network

Setelah mekanisme perhatian selesai, output dari self-attention diteruskan ke feedforward network. Feedforward network pada umumnya menggunakan fungsi aktivasi ReLU (Rectified Linear Unit) dan diikuti oleh transformasi linear:

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

Dimana W_1 , b_1 , W_2 , dan b_2 adalah bobot dan bias yang dilatih selama proses pelatihan.

Misal:

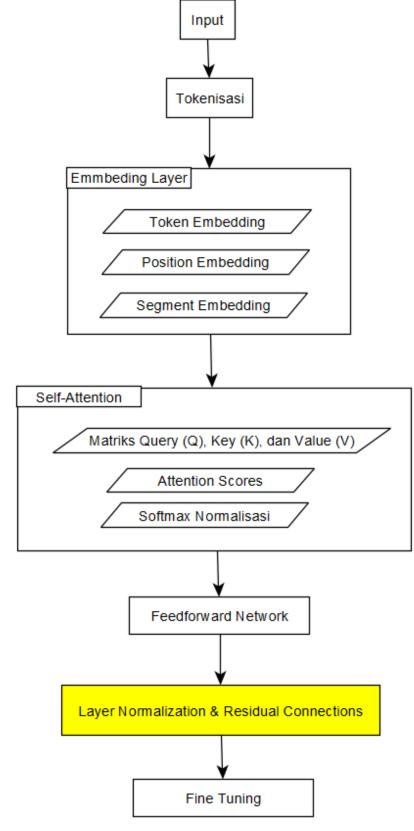
$$W_1 = egin{bmatrix} 0.1 & 0.2 \ 0.3 & 0.4 \end{bmatrix}, \quad b_1 = [0.1, 0.1]$$

Hitung untuk setiap token.

Setelah proses ini, kita akan mendapatkan representasi yang lebih kaya dan bisa diproses untuk tugas selanjutnya (misalnya, klasifikasi sentimen).









Layer Normalization & Residual Connections

Residual Connections

Pada langkah ini, residual connections bertujuan untuk menambahkan input lapisan sebelumnya ke output lapisan saat ini. Secara matematis, langkah ini dituliskan sebagai:

Output Residual = X + Output dari Layer (misalnya Self-Attention atau Feedforward)

Dimana:

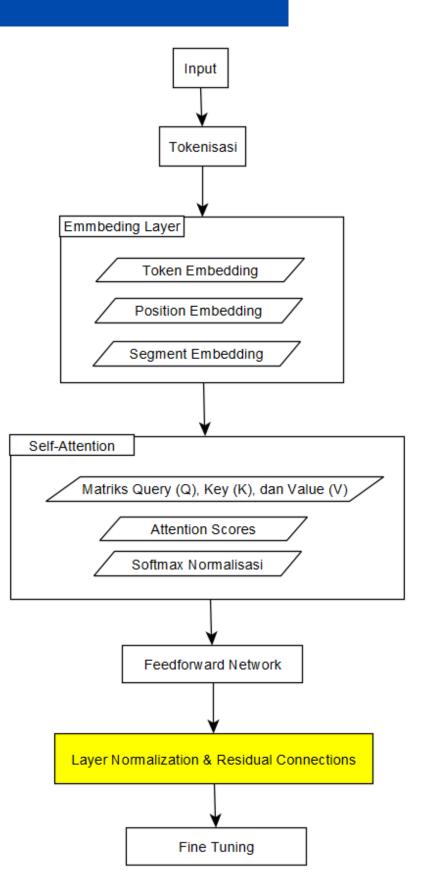
- \bullet X adalah input asli dari lapisan (input ke lapisan perhatian atau feedforward).
- Output dari Layer adalah hasil output setelah operasi perhatian atau feedforward.

Contoh: Misalnya, jika output dari layer self-attention adalah $\mathrm{Output_{SA}} = [0.8, 0.7]$, dan input ke layer adalah X=[0.5,0.4], maka:

$${\rm Output\,Residual} = X + {\rm Output_{SA}} = [0.5, 0.4] + [0.8, 0.7] = [1.3, 1.1]$$







Layer Normalization & Residual Connections

Layer Normalization

Setelah residual connection, **layer normalization** diterapkan pada hasil penjumlahan ini untuk menstabilkan distribusi output. Layer normalization mengurangi perbedaan distribusi antar-lapisan yang bisa menghambat pelatihan, serta mencegah masalah eksploding atau vanishing gradient. Layer normalization dilakukan dengan rumus berikut:

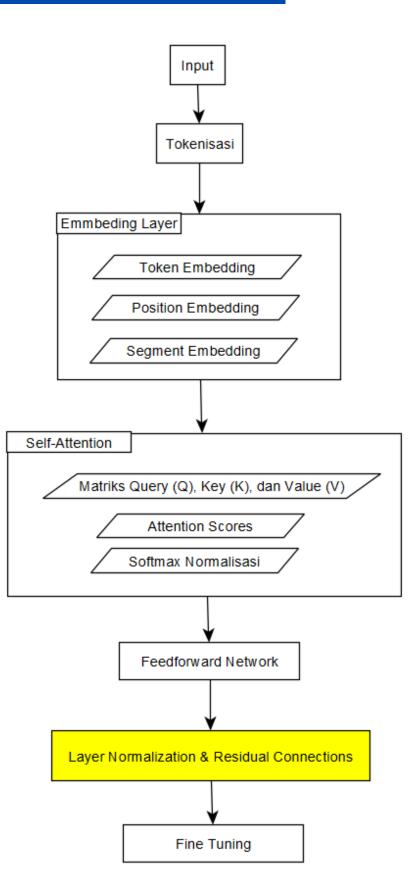
$$\hat{x} = \frac{x - \mu}{\sigma}$$

Dimana:

- ullet adalah input ke layer normalization (yaitu hasil penjumlahan residual connection).
- ullet μ adalah rata-rata dari elemen-elemen dalam vektor x.
- σ adalah deviasi standar dari vektor x.







Layer Normalization & Residual Connections

Contoh: Jika hasil penjumlahan residual adalah Output Residual = [1.3, 1.1], kita hitung rata-rata dan deviasi standar dari vektor ini:

- Rata-rata (μ) = $\frac{1.3+1.1}{2} = 1.2$
- Deviasi standar (σ) = $\sqrt{\frac{(1.3-1.2)^2+(1.1-1.2)^2}{2}}=\sqrt{\frac{0.01+0.01}{2}}=\sqrt{0.01}=0.1$

Sekarang kita normalisasi elemen-elemen dalam vektor [1.3, 1.1]:

$$\hat{x}_1 = rac{1.3 - 1.2}{0.1} = 1.0$$

$$\hat{x}_2 = rac{1.1 - 1.2}{0.1} = -1.0$$

Jadi, setelah normalisasi, hasil outputnya menjadi:

$$\hat{X} = [1.0, -1.0]$$







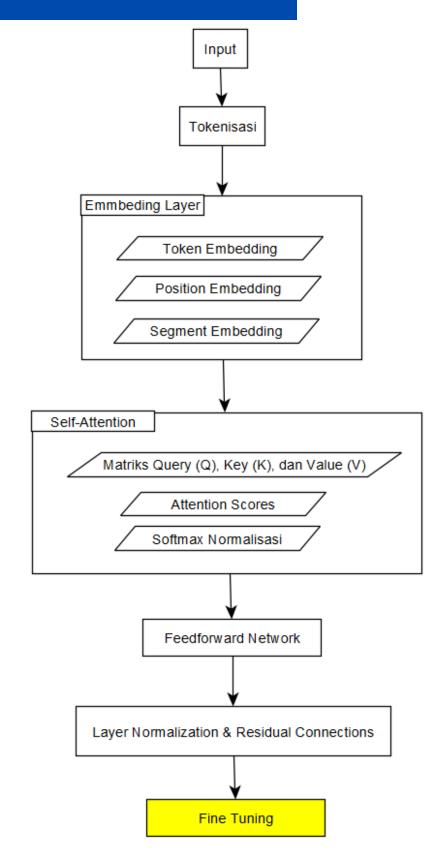


Setelah tahap pre-training, BERT dapat di-*fine-tune* untuk tugas spesifik seperti klasifikasi sentimen. Di sini, representasi terakhir dari token atau agregasi dari token (misalnya, representasi token pertama) digunakan sebagai input untuk lapisan classifier untuk menghasilkan prediksi.

Secara matematis, output dari BERT (h) kemudian dilewatkan ke lapisan classifier untuk menghasilkan skor kelas:

$$\hat{y} = \operatorname{softmax}(W_c h + b_c)$$

Di mana W_c adalah matriks bobot dari classifier dan b_c adalah bias, dan \hat{y} adalah prediksi kelas dari model.







Tokenisasi Emmbeding Layer Token Embedding Position Embedding Segment Embedding Self-Attention Matriks Query (Q), Key (K), dan Value (V) Attention Scores Softmax Normalisasi

Feedforward Network

Layer Normalization & Residual Connections

Fine Tuning





Fine-Tuning untuk Tugas Spesifik

1. Input Token

"aku mau makan" dengan token yang sudah diproses melalui BERT menghasilkan output embedding untuk token [CLS]. Misalkan embedding [CLS] memiliki dimensi 768.

2. Representasi Output dari BERT

Output dari model BERT adalah vektor fitur dari token [CLS], misalnya:

$$h_{[CLS]} = [h_1, h_2, ..., h_{768}]$$

Contoh nilai (dimensi 768):

$$h_{[CLS]} = [0.5, -0.2, 0.3, ..., 0.1] \\$$





Tokenisasi Emmbeding Layer Token Embedding Position Embedding Segment Embedding Self-Attention Matriks Query (Q), Key (K), dan Value (V) Attention Scores Softmax Normalisasi Feedforward Network Layer Normalization & Residual Connections

Fine Tuning

BERT



Fine-Tuning untuk Tugas Spesifik

3. Fully Connected Layer

Pada fine-tuning, kita tambahkan lapisan fully connected (dense layer) di atas output [CLS] untuk memproyeksikan ke jumlah kelas (K=3). Rumus:

$$z = h_{[CLS]} \cdot W + b$$

- $h_{[CLS]}$: Vektor keluaran token <code>[CLS]</code> dari BERT (shape = [1,768]).
- W: Matriks bobot dense layer (shape = [768, 3]).
- b: Vektor bias (shape = [1, 3]).
- z: Logit untuk setiap kelas (shape = [1, 3]).

Misalkan:

$$W = egin{bmatrix} 0.1 & -0.2 & 0.3 \ ... & ... & ... \ 0.05 & 0.1 & -0.05 \end{bmatrix}, \quad b = [0.01, -0.02, 0.03]$$

Hasil logit z adalah:

$$z=\left[z_1,z_2,z_3\right]$$





Tokenisasi Emmbeding Layer Token Embedding Position Embedding Segment Embedding Self-Attention Matriks Query (Q), Key (K), dan Value (V) Attention Scores Softmax Normalisasi

Feedforward Network

Layer Normalization & Residual Connections

Fine Tuning

BERT



Fine-Tuning untuk Tugas Spesifik

4. Softmax untuk Probabilitas Softmax digunakan untuk mengubah logit zzz menjadi probabilitas:

$$\operatorname{softmax}(z_i) = rac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

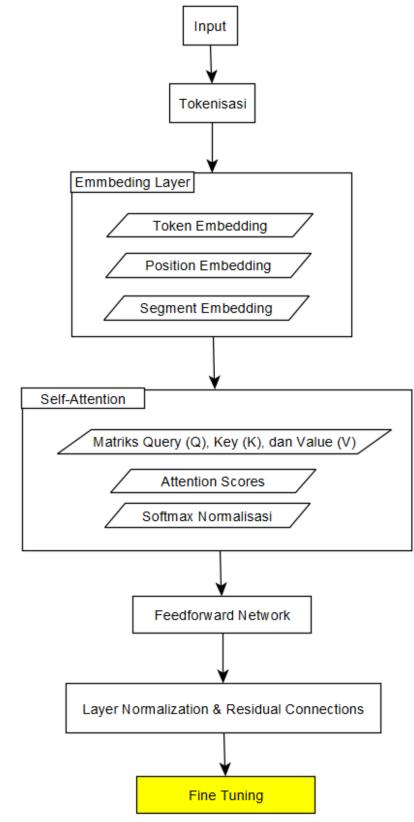
Contoh: Misalkan z = [1.0, 0.5, -0.5]:

$$ext{softmax}(z) = \left[rac{e^{1.0}}{e^{1.0} + e^{0.5} + e^{-0.5}}, rac{e^{0.5}}{e^{1.0} + e^{0.5} + e^{-0.5}}, rac{e^{-0.5}}{e^{1.0} + e^{0.5} + e^{-0.5}}
ight] \ ext{softmax}(z) pprox [0.58, 0.34, 0.08]$$

Hasil akhir adalah distribusi probabilitas untuk setiap kelas.









Fine-Tuning untuk Tugas Spesifik

5. Loss Function (Cross-Entropy)

Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah Integer Label.

1. Rumus Loss untuk Integer Label

Untuk label integer, Cross-Entropy Loss didefinisikan sebagai:

$$Loss = -\log(\hat{y}_{target})$$

di mana:

• \hat{y}_{target} : Probabilitas prediksi untuk kelas target (y).

2. Ambil Probabilitas Kelas Target

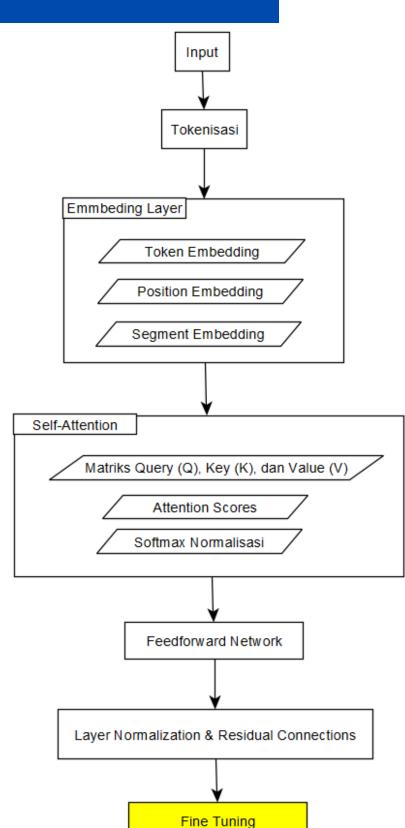
Dari data yang diberikan:

- Target adalah kelas ke-1 (y = 1).
- Probabilitas prediksi untuk kelas ke-1 adalah:

$$\hat{y}_{\mathrm{target}} = \hat{y}[1] = 0.34$$









Fine-Tuning untuk Tugas Spesifik

5. Loss Function (Cross-Entropy)

Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah Integer Label.

3. Hitung Logaritma Probabilitas

Gunakan logaritma natural (ln):

$$\log(0.34) pprox -1.0792$$

4. Hitung Nilai Loss

Masukkan hasil logaritma ke dalam rumus:

$$Loss = -(-1.0792) = 1.0792$$

Hasil Akhir

Nilai Cross-Entropy Loss untuk prediksi $\hat{y} = [0.58, 0.34, 0.08]$ dan label target y = 1 adalah:

$$Loss = 1.0792$$

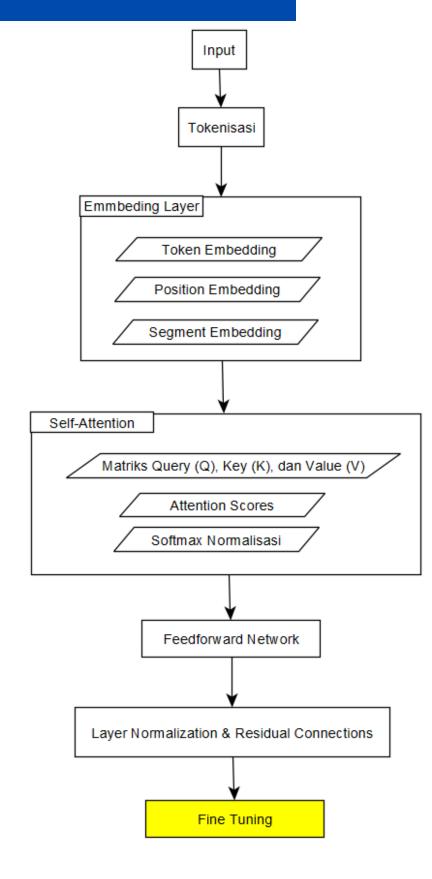
Jika Ada Batch Data

Untuk batch dengan beberapa sampel, loss dihitung dengan rata-rata nilai loss dari setiap sampel:

$$ext{Loss}_{ ext{batch}} = rac{1}{N} \sum_{i=1}^{N} -\log(\hat{y}_{i, ext{target}})$$



SENADA Seminar Nasional Sains Data



BERT



Fine-Tuning untuk Tugas Spesifik

5. Loss Function (Cross-Entropy)

Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah One Hot Encoding.

Rumus:

$$ext{Loss} = -\sum_{i=1}^K y_i \cdot \log(\hat{y}_i)$$

- y_i : Label sebenarnya (0 atau 1 untuk setiap kelas).
- \hat{y}_i : Probabilitas prediksi untuk kelas ke-i (hasil softmax).

Contoh:

- Label y = [0, 1, 0] (kelas 2 adalah target).
- Prediksi $\hat{y} = [0.58, 0.34, 0.08]$.

Loss:

$$ext{Loss} = -\left(0 \cdot \log(0.58) + 1 \cdot \log(0.34) + 0 \cdot \log(0.08)\right)$$
 $ext{Loss} = -\log(0.34) \approx 1.08$





Input Tokenisasi Emmbeding Layer Token Embedding Position Embedding Segment Embedding Self-Attention Matriks Query (Q), Key (K), dan Value (V) Attention Scores Softmax Normalisasi Feedforward Network Layer Normalization & Residual Connections

Fine Tuning

BERT



Fine-Tuning untuk Tugas Spesifik

- 5. Loss Function (Cross-Entropy)
 Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah One Hot Encoding.
- 1. Menghitung Gradien dari Loss terhadap Output Softmax Kita mulai dengan menghitung gradien dari loss terhadap output softmax (\hat{y}) .
 - Cross-Entropy Loss untuk label integer y dan probabilitas prediksi \hat{y} adalah:

$$\mathrm{Loss} = -\log(\hat{y}_y)$$

Di mana \hat{y}_y adalah probabilitas untuk kelas target y.

• Gradien dari loss terhadap \hat{y}_y adalah:

$$\frac{\partial \text{Loss}}{\partial \hat{y}_y} = \frac{-1}{\hat{y}_y}$$

Ini berasal dari turunan logaritma.

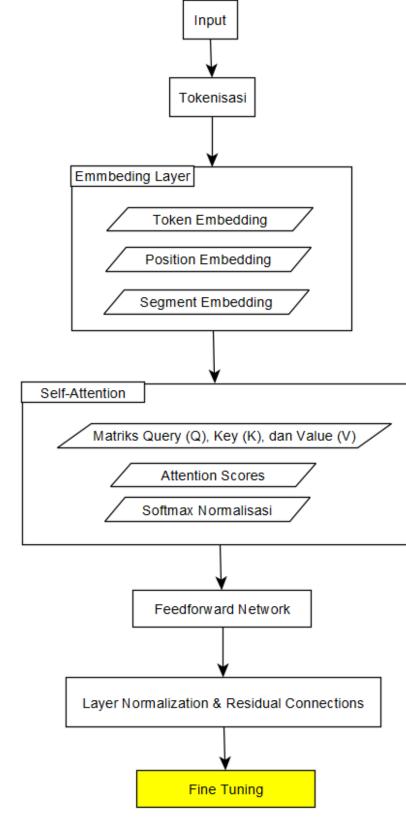
Untuk kasus kita:

- $\hat{y}_1 = 0.34$ (probabilitas untuk kelas target).
- Maka:

$$rac{\partial \mathrm{Loss}}{\partial \hat{y}_1} = rac{-1}{0.34} pprox -2.9412$$









Fine-Tuning untuk Tugas Spesifik

5. Loss Function (Cross-Entropy) Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah One Hot Encoding.

2. Menghitung Gradien untuk Kelas Lainnya

ullet Untuk kelas lain (misalnya $\hat{y}_0=0.58$ dan $\hat{y}_2=0.08$), gradiennya dihitung dengan rumus:

$$rac{\partial ext{Loss}}{\partial \hat{y}_i} = \hat{y}_i \quad ext{untuk} \quad i
eq y$$

Di sini y=1, sehingga:

Untuk kelas 0 (non-target):

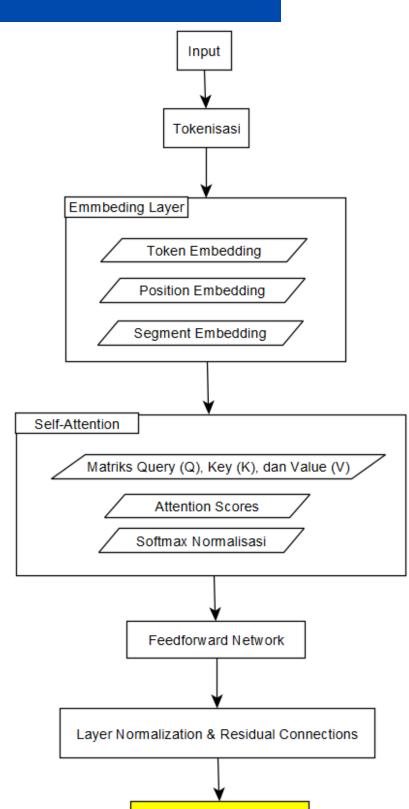
$$\frac{\partial \text{Loss}}{\partial \hat{y}_0} = 0.58$$

• Untuk kelas 2 (non-target):

$$\frac{\partial \text{Loss}}{\partial \hat{y}_2} = 0.08$$







Fine Tuning

BERT



Fine-Tuning untuk Tugas Spesifik

5. Loss Function (Cross-Entropy)
Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah One Hot Encoding.

3. Menghitung Gradien terhadap Parameter (Bobot dan Bias)

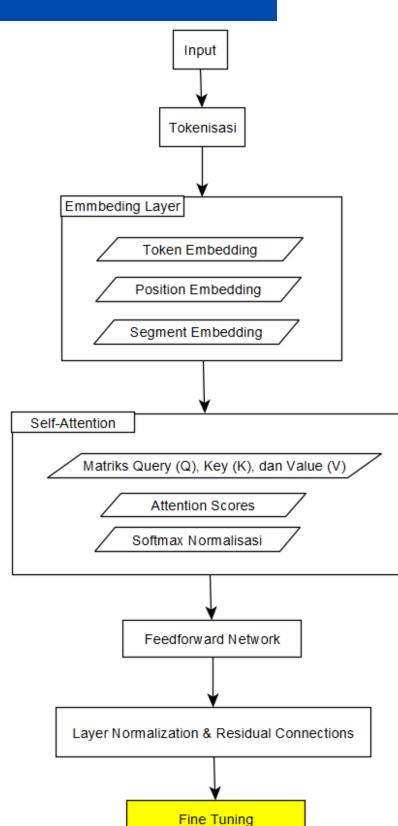
- Setelah menghitung gradien terhadap output \hat{y} , kita akan menghitung gradien terhadap parameter model. Dalam hal ini, kita perlu menghitung gradien loss terhadap bobot dan bias di setiap lapisan model.
- Gradien ini dihitung dengan menggunakan chain rule dari kalkulus:

$$\frac{\partial \text{Loss}}{\partial W} = \frac{\partial \text{Loss}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W}$$

Ini menghitung bagaimana perubahan pada bobot W akan mempengaruhi loss.









Fine-Tuning untuk Tugas Spesifik

- 5. Loss Function (Cross-Entropy)
 Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah One Hot Encoding.
- 4. Update Parameter dengan Optimizer
 - Setelah kita mendapatkan gradien untuk setiap parameter, kita akan memperbarui parameter model menggunakan algoritma optimisasi seperti Adam atau SGD.
 - Pada umumnya, update parameter menggunakan rumus:

$$W_{
m new} = W_{
m old} - \eta \cdot rac{\partial {
m Loss}}{\partial W}$$

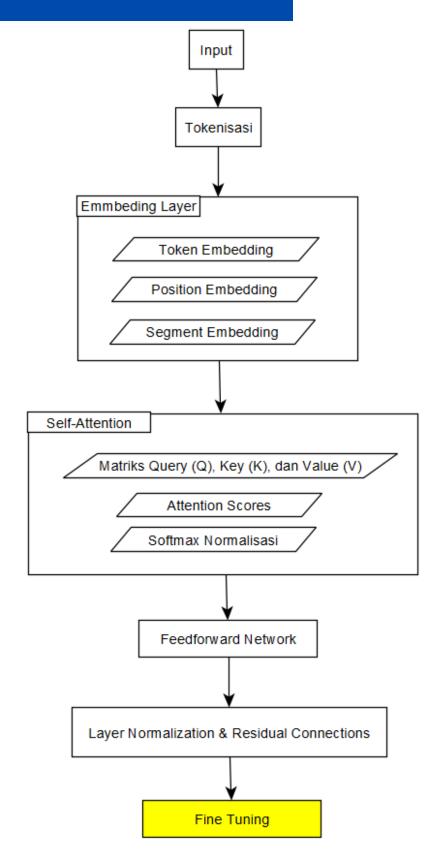
Di sini:

- η adalah **learning rate** (misalnya, 0.001).
- ullet $W_{
 m new}$ adalah bobot yang diperbarui.
- $\frac{\partial \text{Loss}}{\partial W}$ adalah gradien loss terhadap bobot.

Algoritma **Adam** (Adaptive Moment Estimation) adalah salah satu optimizers yang banyak digunakan, yang menggabungkan gradien pertama (momentum) dan gradien kedua (skala).









Fine-Tuning untuk Tugas Spesifik

5. Loss Function (Cross-Entropy)

Pada fine-tuning, model dihitung loss-nya menggunakan cross-entropy jika label target adalah One Hot Encoding.

Kita ambil contoh dengan update bobot menggunakan algoritma SGD dengan learning rate $\eta=0.01$.

1. Gradien untuk Output: Seperti yang dihitung sebelumnya:

$$rac{\partial ext{Loss}}{\partial \hat{y}_1} pprox -2.9412, \quad rac{\partial ext{Loss}}{\partial \hat{y}_0} = 0.58, \quad rac{\partial ext{Loss}}{\partial \hat{y}_2} = 0.08$$

2. Update Bobot: Misalkan kita memiliki bobot $W_{\rm old}=0.5$. Dengan gradien yang dihitung, kita dapat mengupdate bobot untuk masing-masing kelas (output softmax) menggunakan rumus:

$$W_{
m new} = W_{
m old} - \eta \cdot rac{\partial {
m Loss}}{\partial W}$$

• Misalkan gradien untuk bobot tertentu $rac{\partial \mathrm{Loss}}{\partial W} = -0.3$, maka:

$$W_{\text{new}} = 0.5 - 0.01 \cdot (-0.3) = 0.5 + 0.003 = 0.503$$

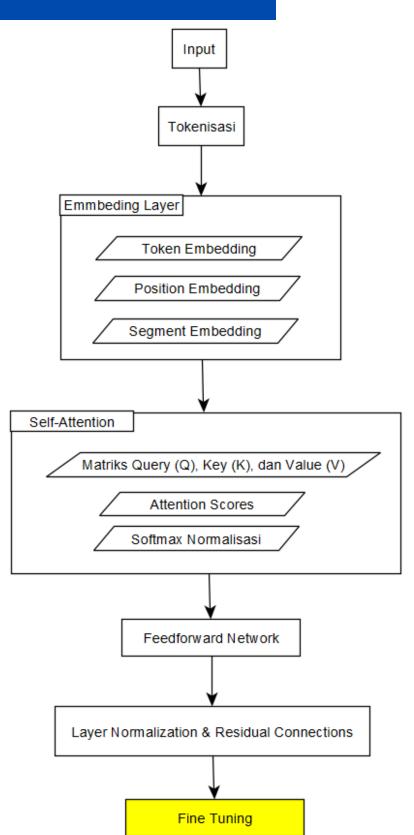
 Perbarui Semua Bobot dan Bias: Proses ini dilakukan untuk semua bobot dan bias dalam model, dan proses ini diulang di setiap epoch hingga model mencapai konvergensi.





BERT





Fine-Tuning untuk Tugas Spesifik

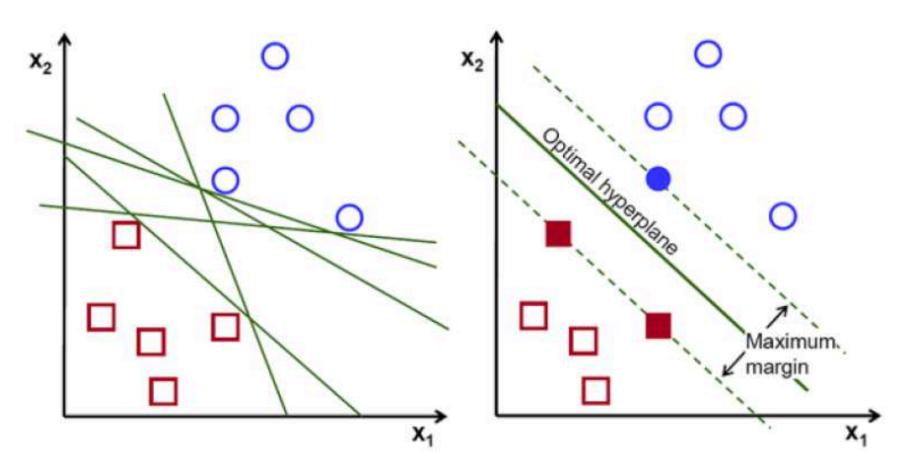
Kesimpulan

Pada bagian Backpropagation, kita menghitung gradien loss terhadap output softmax dan kemudian mengaplikasikan chain rule untuk menghitung gradien terhadap parameter model. Setelah itu, optimizer digunakan untuk memperbarui parameter (bobot dan bias) agar model dapat belajar dan meminimalkan loss. Dengan langkah ini, model melakukan fine-tuning untuk meningkatkan prediksi pada data baru.









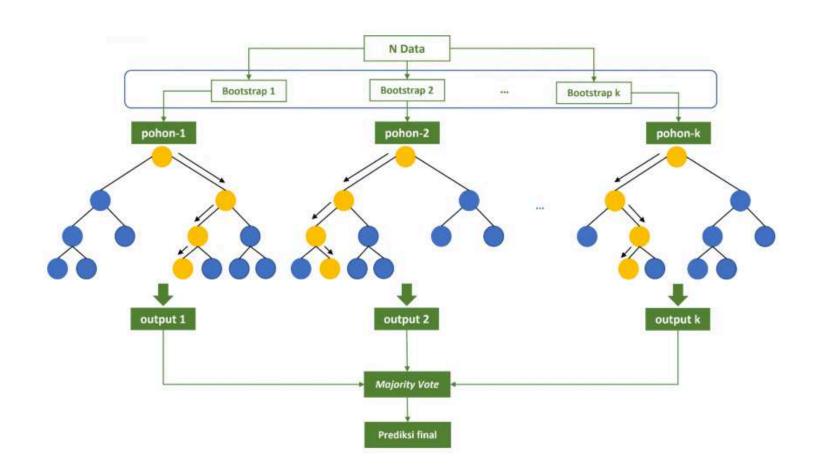
https://codingstudio.id/blog/support-vector-machine/

SVM untuk klasifikasi sentimen positif, negatif, dan netral bekerja dengan memproses teks menjadi fitur numerik, mencari hyperplane optimal untuk memisahkan kelas dengan margin maksimal, dan memprediksi sentimen berdasarkan lokasi data dalam ruang fitur.



SVM





https://sainsdata.id/machine-learning/893/random-forestuntuk-model-klasifikasi-menggunakan-scikitlearn-python/vector-machine/ Random Forest untuk klasifikasi sentimen positif, negatif, dan netral bekerja dengan membangun banyak pohon keputusan dari subset data dan fitur secara acak, menggabungkan hasil prediksi tiap pohon melalui voting mayoritas untuk menghasilkan prediksi akhir, dengan keunggulan ketahanan terhadap overfitting dan kemampuan menangani multiclass secara alami.







HASIL PREPROCESSING





Tabel 2. Tabel Hasil Pra Pemrosesan Data

Data Awal

a~ Free Aksesoris (Kaca Film, Karpet Dasar, Karpet Ori, Dompet Honda, Gantungan Kunci, Sensor Parkir, APAR, dll). *syarat & ketentuan berlaku (MELAYANI Mobil Honda Payakumbuh added a... - Mobil Honda Payakumbuh Mobil Honda Payakumbuh added a new photo.

Data Bersih

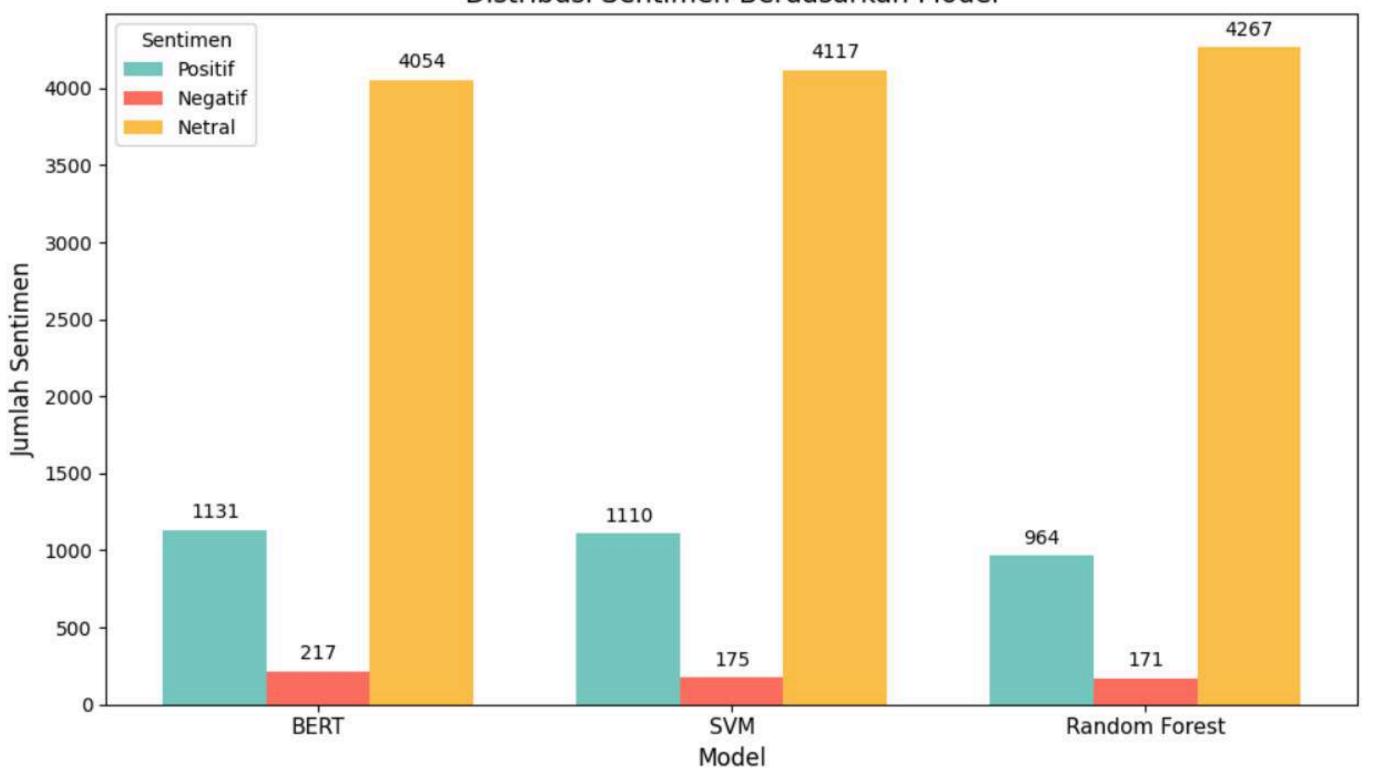
free aksesoris kaca film karpet dasar karpet ori dompet honda gantungan kunci sensor parkir apar dll syarat ketentuan berlaku melayani mobil honda payakumbuh added a mobil honda payakumbuh mobil honda payakumbuh added a new photo



HASIL PELABELAN



Distribusi Sentimen Berdasarkan Model

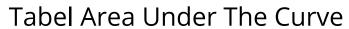








Model	Accuracy	recall	presicion	F1-Score
BERT	85%	84,86%	84,79%	85,07%
SVM	75%	75,74%	77,33%	75,67%
Random Forest	70%	71,11%	76,19	70,79%



Model	Positif	Negatif	Netral
BERT	86,96 %	96,25 %	82,94 %
SVM	75,24%	92,50%	76,98%
Random Forest	71,05%	90%	73,81%

Tabel Cross-Validation

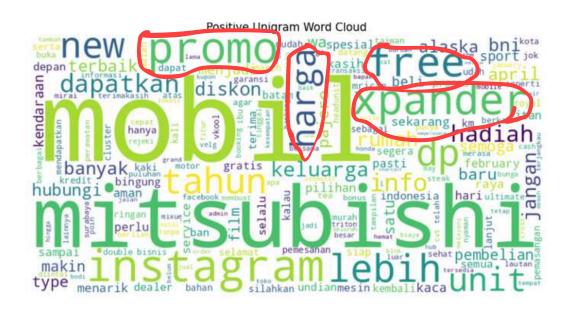
Fold	BERT	SVM	Random Forest
1	91,67%	41,67%	58,33%
2	100%	66,67%	33,33%
3	100%	58,33%	58,33%
4	66,67%	75%	58,33%
5	66,67%	66,67%	41,67%
Average	85%	61,67%	50%



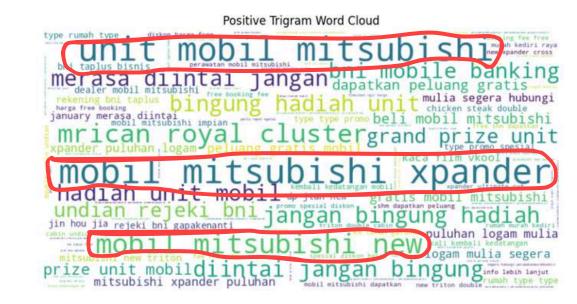




Wordcloud Mitsubishi



```
with the property of the prope
```

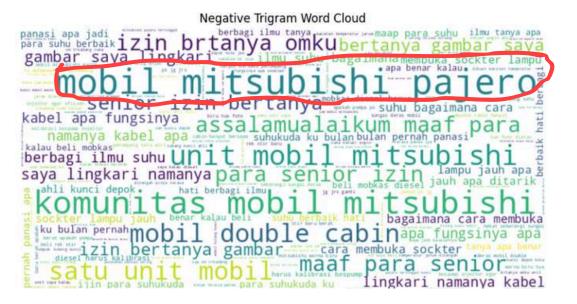


```
Negative Unigram Word Cloud

The property of t
```

```
Negative Bigram Word Cloud

Cara membuka maap para sukabumi benar kalau korban pelecehan namanya kabel apa ditarik kota sukabumi benar kalau korban pelecehan namanya kabel musockter lampu benar ya musockter lampu benar senior mananya lampu jauh gambar saya gunda senior izinpolda lampung suhukuda ku kunci mobil asuhu berbaik injelior apar mitsubishi worna gunda suhukuda ku kunci mobil senior izinpolda lampung suhukuda ku kunci mobil suhu berbaik injelior apar mitsubishi worna gunda suhukuda ku kunci mobil suhu berbaik injelior apar mitsubishi worna gunda suhukuda ku kunci mobil suhu berbaik injelior apar mitsubishi worna gunda suhukuda ku kunci mobil suhu berbaik injelior apar mitsubishi worna gunda suhukuda ku kunci mobil suhu berbaik injelior apar membuka sokterbeli mobkas lama nongoli suhu baranga di mobil suhu baranga suhukuda ku kunci mobil maar para di mobil suhu baranga di mobil suhu
```

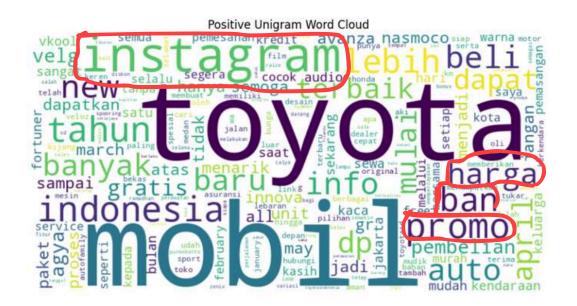


Mitsubishi pada sentimen positif menunjukkan keberhasilan promosi dan untuk model xpander mendapat respon positif dari masyarakat, sementara sentimen negatif mengindikasikan masalah bagian kunci dan untuk model pajero mendapat respon negatif.





Wordcloud Toyota











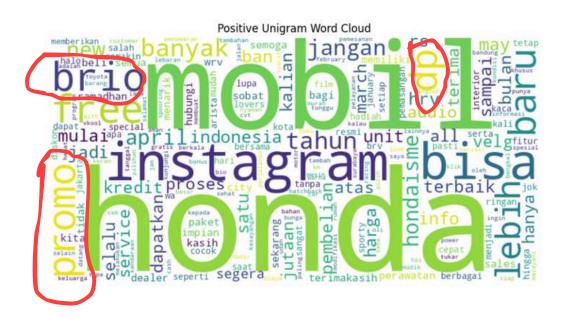


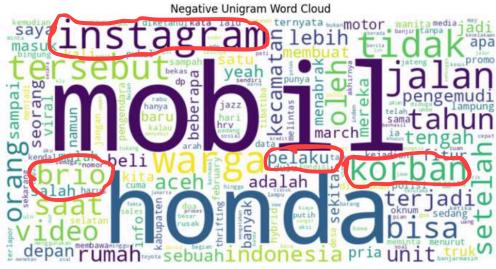
Toyota mendapat banyak respon positif di Instagram dan untuk model avanza, fortuner dan agya mendapat respon positif, kemudian terdapat isu pelaku, korban, dan balap liar yang mendominasi untuk sentimen negatif dan untuk model alpart, rush, dan kijang inova mendapat respon negatif.





Wordcloud Honda









Honda mendapat respon positif untuk model brio dan hrv serta untuk bagian velg ban dan fitur, tetapi berdasarkan sentimen negatifnya perlu evaluasi pada media sosial Instagram, kemudian untuk model jazz, brio dan hrv juga mendapat respon negatif, isu korban juga banyak di perbincangkan.





Wordcloud Suzuki











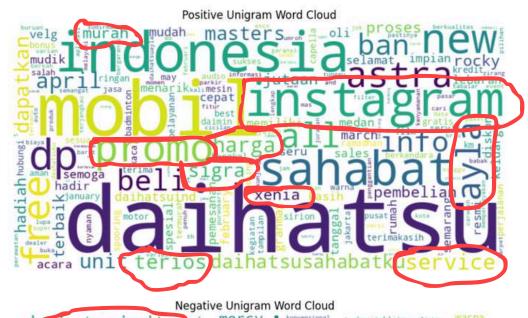


Suzuki mendapat respon positif untuk slogan "suzuki teman berkendara" begitu juga untuk model spreso,new carry, dan ertiga, namun berdasarkan sentimen negatifnya perlu evaluasi di Instagram dan isu balap liar yang mendominasi.

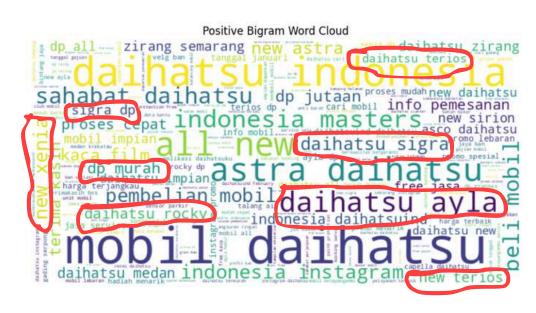


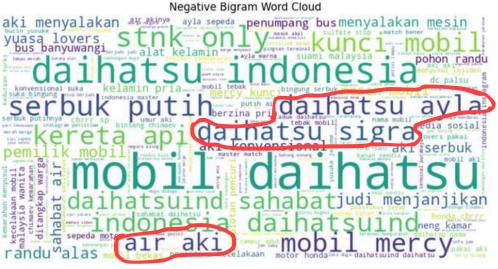


Wordcloud Daihatsu









```
Positive Trigram Word Cloud

adihatsu all new dp all new velg ban variasi
daihatsu indonesia daihatsuind adihatsu cari mobil

all new xenia daihatsu zirang semarang
taint propriation free daihatsu indonesia daihatsu zirang semarang
taint propriation free daihatsu indonesia mobil lebaran ter
taint propriation free daihatsu indonesia mobil daihatsu per
taint propriation free daihatsu indonesia mobil daihatsu per
taint propriation free daihatsu indonesia mobil daihatsu per
taint propriation free daihatsu sigralinto mobil daihatsu per
taint propriation free daihatsu sigralinto mobil daihatsu per
taint propriation daihatsu free daihatsu sigralinto mobil daihatsu per
taint propriation daihatsu per
taint propri
```

```
Regative ingram word cloud

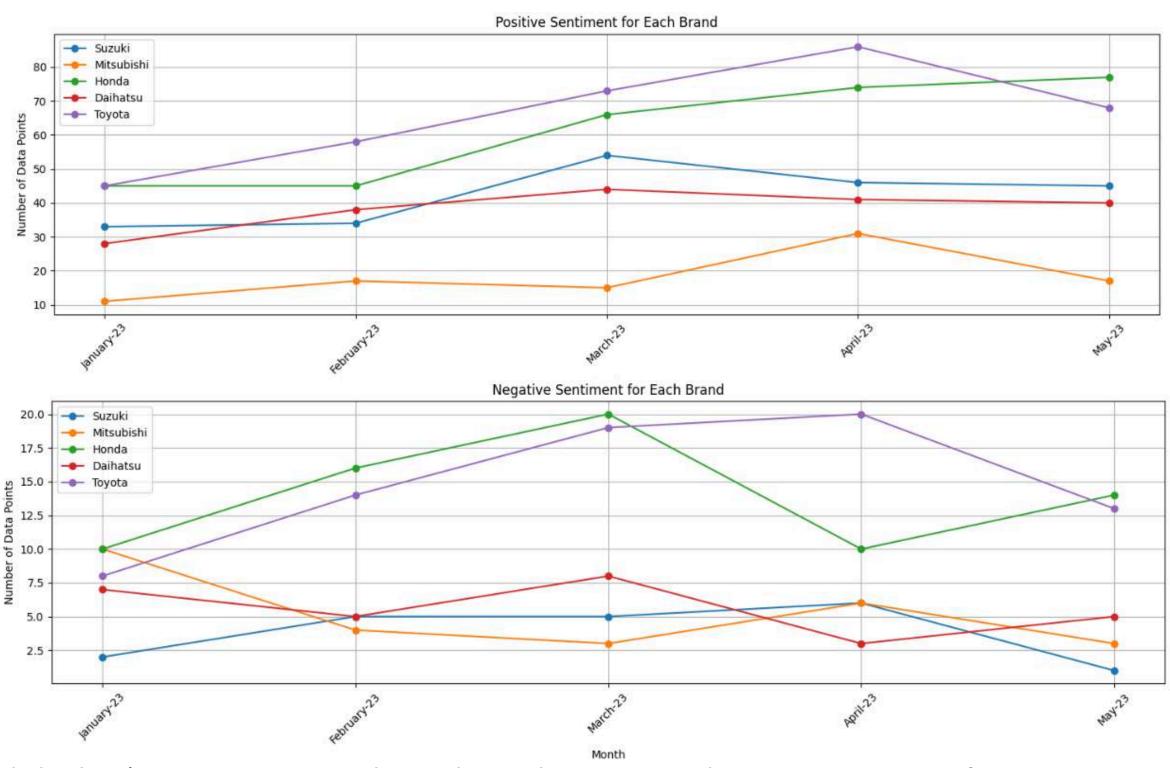
adjusted in the part of th
```

Daihatsu mendapat respon positif dominan di media sosial Instagram dan untuk model ayla, xenia, sigra dan terios mendapat respon yang positif, sedangkan untuk sentimen negatifnya isu aki dan kecelakaan mendominasi.





Tren Sentimen



Tren sentimen menunjukkan keberhasilan promosi Toyota dan Daihatsu dengan peningkatan sentimen positif, sementara Honda dan Suzuki menunjukkan fluktuasi dengan puncak sentimen negatif pada April 2023. Mitsubishi juga menunjukkan peningkatan kepuasan sejak Januari, sementara fluktuasi sentimen negatif menunjukkan perlu adanya evaluasi.



KESIMPULAN



Berdasarkan analisis dan pengujian yang dilakukan pada penelitian ini, maka didapatkan kesimpulan model BERT menunjukkan kinerja terbaik dengan rata-rata akurasi sebesar 85%, recall 84,86%, presisi 84,79%, dan F1-Score 85,07%.

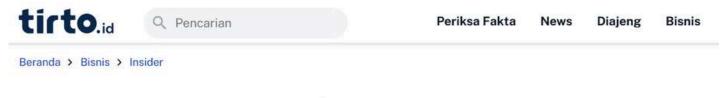
Secara keseluruhan pada sentimen positif setiap merek mobil cukup baik dalam melakukan promosi untuk menarik minat pembeli, sedangkan pada sentimen negatifnya setiap merek mobil didominasi dengan isu kecelakaan dan korban yang memungkinkan hal ini berkaitan dengan isu skandal uji keselamatan pada industri otomotif yang terjadi pada tahun 2023. Namun, masing-masing merek juga memiliki aspek yang perlu diperbaiki. Mitsubishi harus menangani masalah pada kunci dan model Pajero, sementara Toyota perlu mengatasi sentimen negatif terhadap model Alphard, Rush, dan Kijang Innova. Honda mendapatkan respon positif untuk model Brio dan HRV, namun perlu dilakukan evaluasi di media sosial serta sentimen negatif pada model Jazz, Brio, dan HRV. Suzuki berhasil mempertahankan respon positif untuk model S-Presso, New Carry, dan Ertiga, namun berdasarkan sentimen negatifnya perlu evaluasi di Instagram dan isu balap liar yang mendominasi. Daihatsu mendapat respon positif dominan di media sosial Instagram serta untuk model Ayla, Xenia, Terios, dan Sigra, namun perlu evaluasi pada isu aki yang mendominasi.

Tren sentimen menunjukkan bahwa Honda dan Toyota mengalami pertumbuhan signifikan dalam sentimen positif, dengan Honda menunjukkan konsistensi tertinggi, sementara Suzuki dan Daihatsu memiliki tren positif yang stabil. Sentimen negatif cenderung fluktuatif pada semua merek, dengan peningkatan signifikan pada Toyota hingga April 2023, sedangkan Suzuki dan Mitsubishi menunjukkan penurunan yang konsisten pada sentimen negatif. Merek-merek dengan sentimen positif yang tinggi, seperti Honda dan Suzuki, perlu mempertahankan strategi mereka, sementara Toyota harus memprioritaskan pengelolaan isu untuk mengurangi peningkatan sentimen negatif yang mencolok.

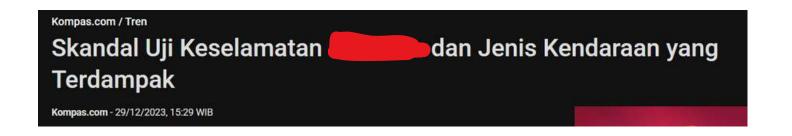


KESIMPULAN





Menguak Tabir Skandal Uji **Keselamatan Industri Otomotif**











- https://tirto.id/skandal-daihatsu-gUgR
- https://www.kompas.com/tren/read/2023/12/29/152925265/skandal-uji-keselamatan-daihatsu-dan-jenis-kendaraan-yang-terdampak? page=all
- https://oto.detik.com/mobil/d-7158607/skandal-uji-keselamatan-terbongkar-daihatsu-recall-ratusan-ribu-mobil
- https://www.tempo.co/otomotif/daihatsu-terlibat-skandal-keselamatan-mobil-begini-detailnya-106152
- https://www.kompas.id/baca/internasional/2024/06/04/skandal-uji-keselamatan-mobil-jepang-kantor-pusat-toyota-diperiksa-aparat













