# Pendekatan Metode Transformers untuk Deteksi Bahasa Kasar dalam Komentar Berita Online Indonesia

Adriansyah Dwi Rendragraha[1], Moch. Arif Bijaksana[2], Ade Romadhony[3]

[1,2,3] Fakultas Informatika, Universitas Telkom, Bandung
[1]adriansyahdr@students.telkomuniversity.ac.id, [2]2arifbijaksana@telkomuniversity.ac.id,
[3]aderomadhony@telkomuniversity.ac.id

Abstrak

Penggunaan internet dalam keseharian dapat terlihat semakin meningkat dari tahun ke tahun. Aktifitas yang dilakukan pun beragam, dan salah satunya memberikan komentar terhadap suatu postingan. Komentar ini memiliki peranan yang cukup unik, dimana akan merepresentasikan pikiran seseorang dari postingan yang di baca-nya. Konten setiap komentar pun beragam, tetapi akan ada masalah ketika komentar tersebut bersifat kasar. Berkomentar dengan bahasa kasar ini dapat memberikan kesan buruk baik terhadap pembaca komentar ataupun bagi si pembuat postingan. Karena hal tersebut, banyak penelitian yang membuat deteksi bahasa kasar dengan berbagai macam metode, dengan metode machine learning hingga deep learning. Tetapi dalam komentar bahasa Indonesia, masih sedikit atau sulit untuk menemukan deteksi bahasa kasar menggunkan metode deep learning. Sehingga dalam penelitian ini, dikembangkan deteksi bahasa kasar dengan metode deep learning yaitu dengan Bidirectional Encoder Representational from Transformers (BERT). Model yang digunakan berupa model BERT dan model pre-train BERT Multilingual untuk menjadi baseline. Sistem akan mendapat masukan berupa teks komentar yang selanjutnya akan mengeluarkan label untuk mengklasifikasikan teks komentar tersebut, apakah termasuk Offensive, Normal, atau Non Offensive. Hasil dari Scratch model yang dilatih dengan dataset bahasa Indonesia mendapat Macro Average F1 Score sebesar 50% dibandingkan dengan BERT Multilingual sebesar 54%.

Kata kunci : BERT, bahasa kasar, berita, deteksi, komentar

Abstract

Daily use of the internet can be seen increasing from year to year. The activities carried out are also varied, and one of them is giving comments on a post. This comment has a unique role, which will represent someone's thoughts from the posts they comment on. The content of each comment varies, but there will be problems when the comments are abusive. Commenting in abusive language can give a bad impression both to the reader of the comment and to the creator of the post. Because of this, many studies have made the detection of offensive language using a variety of methods, from machine learning to deep learning. But in the Indonesian commentary, it is still little or difficult to find detection of offensive language using deep learning methods. So that in this study, a deep learning method was developed to detect abusive language, namely the Bidirectional Encoder Representational from Transformers (BERT). The model used is a self-designed BERT and a multilingual BERT pre-train model to become baseline. The system will receive input in the form of comment text which will then issue a label to classify the comment text, whether it includes Offensive, Normal, or Non Offensive. The results of the Scratch model trained with the Indonesian language dataset got a Macro Average F1 Score of 50% compared to the BERT Multilingual of 54%.

Keywords: abusive language, BERT, comment, detection, news

## 1. Pendahuluan

### Latar Belakang

Penggunaan internet dalam keseharian dapat terlihat semakin meningkat dari tahun ke tahun. Aktifitas yang dilakukan biasanya melakukan browsing, bertukar informasi, berkomunikasi, dan sebagai media iklan bagi beberapa usaha. Untuk membaca berita, saat ini sudah tidak dilakukan lagi dengan membaca koran ataupun majalah, tetapi dengan portal berita yang ada dalam internet. Pada portal berita ini memiliki fitur kolom komentar, dimana dengan adanya fitur tersebut pembaca dengan pembuat berita ataupun dengan pembaca lainnya bisa saling berinteraksi.

Komentar yang ada juga menjadi penting, karena dapat menjadi acuan apakah topik berita tersebut mendapat banyak respon atau tidak dan mungkin dapat membangun suatu diskusi online di topik berita tersebut. Dalam

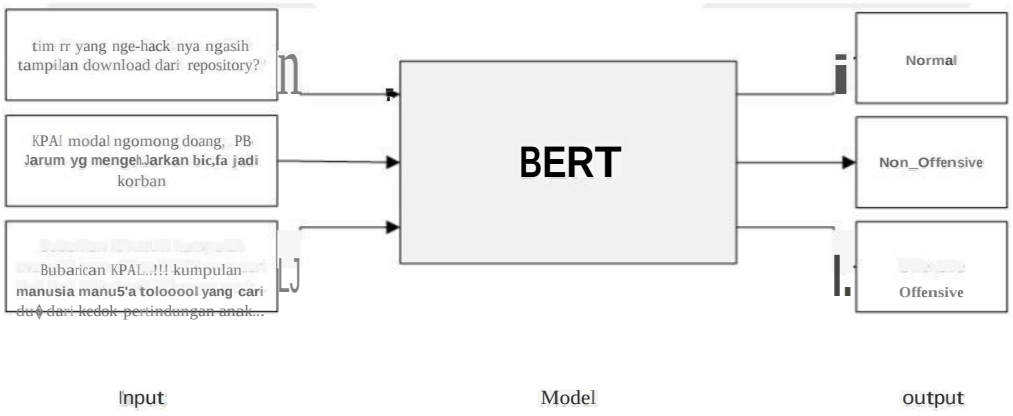
Kamus Besar Bahasa Indonesia (KBBI), komentar merupakan ulasan atau tanggapan atas berita, pidato, dan lain sebagainya. Dengan harapan komentar yang ada berisi komentar yang cerdas, informatif, dan relevan terhadap artikel yang berkaitan. Tetapi, pada kenyataannya isi dari kolom komentar tidak selama nya seperti itu. Konten dalam kolom komentar banyak terdapat bahasa kasar seperti kata intimidasi, kotor, ataupun ujaran kebencian. Target dari penggunaan bahasa kasar ini juga bisa ke berbagai kalangan, dari anak anak hingga dewasa. Remaja dan anak anak paling sering terkena dampak nya, karena mereka menangkap dari apa yang mereka baca. Bahkan beberapa dari mekeka juga ikut berkomentar dengan menggunakan bahasa kasar.

Karena berkomentar dengan bahasa kasar dapat memberikan pengaruh negatif, dibutuhkan suatu alat atau sistem yang dapat mendeteksi bahasa kasar dari suatu kalimat. Salah satu caranya adalah dengan menggunakan metode klasifikasi teks. Berdasarkan riset [4], metode klasifikasi terbaik untuk klasifikasi bahasa kasar dalam bahasa Indonesia di komentar portal berita menggunakan metode SVM dengan mengimplementasikan 'linear' kernel, dan pada riset [1], membangun model HateBERT untuk mendeteksi bahasa kasar dalam bahasa Inggris, hasilnya diperoleh untuk model HateBERT mendapat hasil terbaik dengan membandingkan pada model BERT base.

Berdasarkan hal tersebut, maka dilakukan deteksi bahasa kasar dalam komentar portal berita dengan menggunakan salah satu metode deep learning yang juga merupakan salah satu arsitektur dari transformers yaitu Bidirectional Encoder Representations from Transformers (BERT). Pemilihan penggunaan metode ini didasari karena metode BERT pernah memperoleh state-of-the-art baru pada sebelas permasalahan dalam task NLP [2], yang salah satunya adalah klasifikasi teks dan juga pengaruh dari penelitian [1]. Penelitian ini memiliki tiga opsi untuk penentuan label kelas yaitu Offensive, Normal, atau Non Offensive.

Topik dan Batasannya

Topik penelitian yaitu mendeteksi penggunaan kalimat kasar dengan masukan berupa teks bahasa Indonesia lalu diproses dengan model BERT yang mana hasil keluarannya berupa klasifikasi jenis kalimat. Terdapat 3 jenis label dalam klasifikasi hasil keluaran seperti terlihat pada Gambar 1.




Gambar 1. Contoh Input dan Output Teks

Batasan masalah dalam penelitian ini yaitu dataset yang digunakan terdapat imbalanced atau ketidakseimbangan antara jumlah data pada setiap labelnya. Label yang digunakan dalam klasifikasi teks hanya tiga label, yaitu Non Offensive, Normal, dan Offensive. Analisis performa didapat berdasarkan nilai Macro Average F1-Score dan nilai F1-score untuk masing masing kelas label.

Tujuan

Tujuan penelitian ini adalah untuk mengimplementasikan dan mengevaluasi metode transformer yaitu BERT dalam deteksi bahasa kasar pada teks bahasa Indonesia. Kemudian, sistem akan dievaluasi apakah sudah benar dalam melakukan klasifikasi pada kalimat tes. Metode evaluasi yang digunakan adalah menggunakan nilai Macro Average F1-Score dan nilai F1-score untuk masing masing kelas label.

Organisasi Tulisan

•Studi Terkait
Berisi teori atau studi literatur yang mendukung dengan permasalahan yang dikerjakan.

•Sistem yang Dibangun
Penjelasan mengenai rancangan dan sistem atau produk yang dihasilkan.

•Evaluasi
Berisi hasil pengujian dan analisis dari hasil pengujian. Pengujian dan analisis yang dilakukan selaras dengan tujuan yang telah dinyatakan sebelumnya.

•Kesimpulan
Memuat kesimpulan dan saran dengan mengambil dari hasil pengujian dana analisis hasil pengujian.

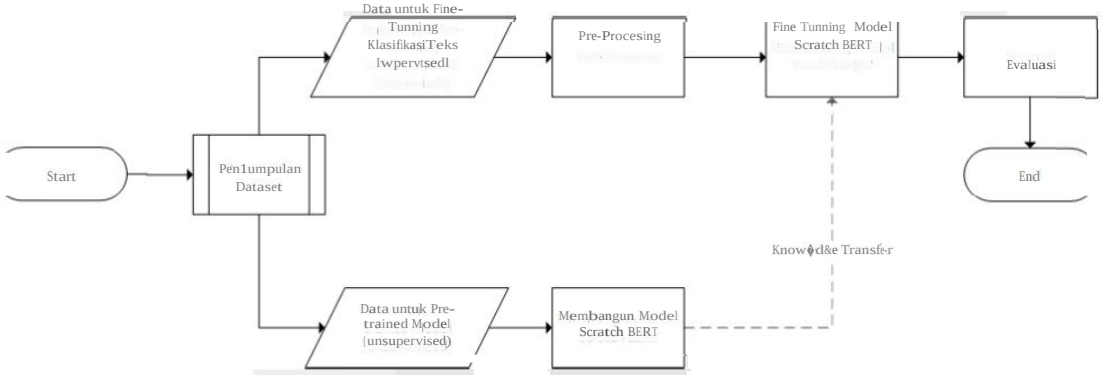
## 2. Studi Terkait

Deteksi bahasa kasar merupakan suatu alat (tools) yang digunakan untuk mendeteksi apakah kalimat tersebut termasuk kedalam kategori kasar atau tidak. Penelitian ini dibangun berdasarkan beberapa referensi dari penelitian yang sudah dilakukan sebelumnya. Penelitian [1], penelitian ini membangun model HateBERT, BERT yang dilatih ulang untuk deteksi bahasa yang kasar dalam bahasa Inggris. Model ini dilatih dengan dataset RAL-E, yang merupakan kumpulan data skala besar dari komentar Reddit dalam bahasa Inggris dari komunitas yang dilarang karena menyinggung, kasar, atau penuh kebencian. Dengan data uji menggunakan tiga dataset bahasa kasar dari OffensEval 2019, AbusEval, dan HatEval, HateBERT mengalahkan performansi dari BERT base ke semua data ujinya. Dengan nilai macro average f1-score sebesar 90.9% untuk dataset OffensEval 2019, 76.5% untuk dataset AbusEval, dan 51.6% untuk dataset HatEval. Selanjutnya pada penelitian [3], memiliki fokus untuk mendeteksi bahasa kasar di twitter dalam bahasa Indonesia dengan pendekatan deep learning. Penelitian ini menggunakan LSTM dengan word embedding untuk mendeteksi bahasa kasar. Hasilnya menunjukkan f1-score yang didapat sebesar 83.68%.

Penelitian [8], membahas mengenai deteksi bahasa kasar di twitter dengan menggunakan skema ansembel RNN dan menggabungkan berbagai fitur yang terkait dengan informasi pengguna, seperti kecenderungan pengguna terhadap rasisme atau seksisme. Skema ini berhasil membedakan pesan rasisme dan seksisme dari teks normal dengan hasil terbaik f1 score pada rasisme sebesar 70.84%, seksisme sebesar 99.86%, dan normal sebesar 95.17%. Pada penilitian [4], mengenai deteksi bahasa kasar dalam bahasa Indonesia dilakukan dengan menggunakan metode Naïve Bayes, SVM, dan KNN. penilitian ini juga menggunakan feature selection berdasarkan nilai Mutual Information antar kata. Dimana dari hasil yang didapat, metode SVM yang mendapatkan hasil terbaik diantara dua metode lainnya dengan nilai akurasi sebesar 90.19% dan penggunaan Mutual Information dapat meningkatkan akurasi sebesar 1.63%.

## 3. Sistem yang Dibangun

Sistem yang dibangun merupakan sistem yang dapat mendeteksi penggunaan bahasa kasar pada kalimat atau teks bahasa Indonesia. Gambar 2 merupakan gambaran alur sistem yang dibangun pada penelitian ini.




Gambar 2. Diagram Alir Sistem

### 3.1 Dataset

Dataset yang digunakan terdiri dari dua buah datset, dimana dataset pertama digunakan untuk membangun pre-trained model dan dataset kedua digunakan untuk melakukan fine-tunning untuk klasifikasi teks. Dataset untuk membangun pre-trained model merupakan dataset yang berisi banyak kalimat atau teks tanpa label, datanya didapat dari crawling data pada twitter dengan keyword kata kasar, berita kompas dan tempo, corpus frog storytelling[7], dan kalimat mix wikipedia yang digabungkan, dengan total dataset sebesar 95,876 KB. Contoh untuk dataset pre-trained model dapat dilihat pada Tabel 1.

Tabel 1. Contoh Dataset pre-trained model

| Teks |
|---|
| Duh gua sendiri cape anjing scroll tl war wor wur wir semua hiks |
| ”Zoro setuju untuk bergabung asalkan ia dijadikan sebagai pimpinan Baroque Works, namun Mr. 7 menolak sehingga terjadi pertempuran antara keduanya yang kemudian dimenangkan oleh Zoro.” |
| Saat ini jumlah napi di sana sebanyak 2500 orang padahal kapasitas hanya 1800 orang. Seorang penjaga, rata-rata, harus mengurus 40 narapida. |

Sedangkan untuk dataset fine-tunning untuk klasifikasi teks didapat dari komentar komentar yang ada pada portal berita yang telah diberi label [4]. Komentar dipilih berdasarkan berita yang sedang trend pada bulan maret 2019 hingga september 2019.Total data yang didapatkan sebanyak 3184 komentar. Data terdiri dari tiga label dengan jumlahnya dapat terlihat pada Tabel 2 dan Contoh untuk dataset fine-tunning untuk klasifikasi teks dapat dilihat pada Tabel 3.

Tabel 2. Jumlah Label pada dataset

| Label | Jumlah |
|---|---|
| Non Offensive | 110 |
| Normal | 2789 |
| Offensive | 285 |

Tabel 3. Contoh Dataset fine-tunning untuk Klasifikasi Teks

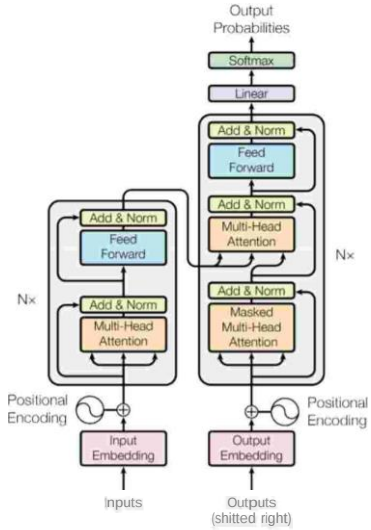
| Teks | Label |
|---|---|
| Beginilah tipe org yg sdg cr pencitraan tdk berfikir pjg...bs nya ngebodohin dan membohongi rakyat sj | Non Offensive |
| tim IT yang nge-hack nya ngasih tampilan download dari repository? | Normal |
| Bubarkan KPAI...!!! kumpulan manusia manusia tolooool yang cari duit dari kedok perlindungan anak... | Offensive |

Kemudian, pada dataset klasifikasi dilakukan pembagian antara data latih dan data uji, dengan perbandingan 70:30. Untuk eksperimen, data latih klasifikasi akan dicoba untuk diuji dengan metode random undersampling dan oversampling [5]. Pada random undersampling label Normal akan dikurang sebanyak 50% dan 45%, sedangkan pada oversampling label Non Offensive dan Offensive akan ditambah sebanyak 5 dan 9 kali lipat.

### 3.2 Pembangunan Model

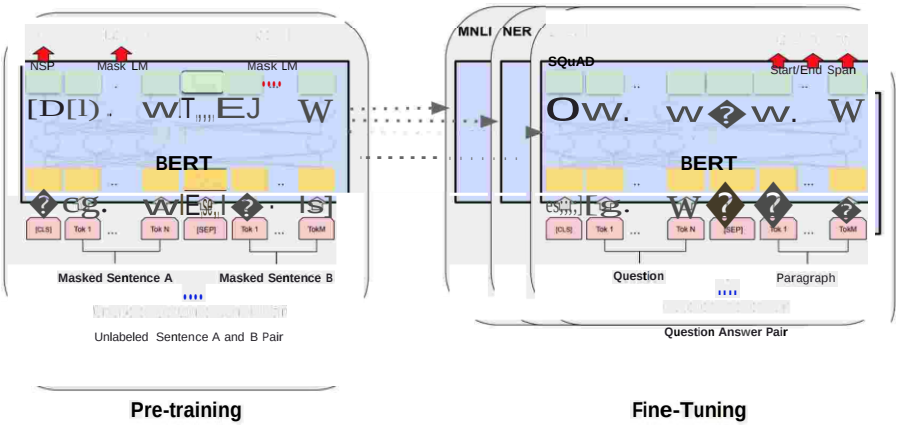
Transformer adalah model transduksi pertama yang mengandalkan sepenuhnya pada self-attention sendiri untuk menghitung representasi input dan outputnya [10]. Keseluruhan arsitektur transformer ini menggunakan tumpukan self-attention dan point-wise, fully connected layers yang terhubung untuk encoder dan decoder, ditunjukkan di bagian kiri dan kanan pada Gambar 3.

Encoder terdiri dari tumpukan N layer identik, dimana tiap layer memiliki dua sub-layer, yang pertama adalah multi-head self-attention mechanism dan yang kedua adalah positionwise fully connected feed-forward network. Pada decoder juga terdiri dari tumpukan N layer identik, dengan tiap layer memiliki tiga sub-layer, yaitu masked multi-head self-attention mechanism, multi-head self-attention mechanism, dan positionwise fully connected feed-forward network.




Gambar 3. Arsitektur Transformers

BERT model arsitektur merupakan multi-layer bidirectional Transformer encoder berdasarkan dari implementasi aslinya yang di jelaskan oleh Vasmani, dkk(2017) dan dikeluarkan di tensor2tensor library [10]. Pada pengerjaannya, mereka menginisialkan banyak layer dengan L, hidden size dengan H, dan banyak self-attention head dengan A. Dengan melaporkan hasil pada dua ukuran model : BERTbase (L=12, H=768, A=12, Total Parameter = 110M) dan BERTlarge (L=24, H=1024, A=16, Total Parameter=340M) [2]. Gambar 4 merupakan contoh gambar jaringan pada pre-train dan fine-tunning BERT.




Gambar 4. Pre-training dan Fine-tunning BERT

Dataset yang digunakan untuk membuat pre-trained model, awalnya dilakukan pelatihan tokenizer. Dimana nantinya hasil dari pelatihan ini akan menjadi vocab tokenisasi yang berisi dataset. pelatihan tokenizer menggunakan library BertWordPieceTokenizer yang disediakan oleh transformers [2], dengan menambahkan empat special token, yaitu: ”[UNK]”, ”[SEP]”, ”[PAD]”, ”[CLS]”, ”[MASK]”. Untuk konfigurasi model, digunakan konifgurasi yang hampir sama dengan konfigurasi bert-based-uncased [2]. Perbedaanya hanya pada vocab_size, karena menyesuaikan vocab_size pada hasil pelatihan tokenizer sebelumnya. Dimana pada model HateBERT[1] juga menggunakan arsitektur dari bert-based-uncased [2]. Berikut pada Gambar 5 adalah konfigurasi model bert yang dibangun.

```
Bertconfig {
  "attention_probs_dropout_prob": 0.1,
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "type_vocab_size": 2,
  "vocab_size": 52000
```


Gambar 5. Konfigurasi Model

Selanjutnya inisialisasi model dengan menggunakan library BertForPreTraining [2] dengan menggunakan konfigurasi yang telah didefinisikan sebelumnya. Library BertForPreTraining sudah memberikan dua heads untuk melakukan pre-training: MLM Head dan NSP Head. Data latih untuk model dibentuk dari dataset yang sudah dipersiapkan dengan menerapkan tokenizer dan menggunakan library TextDatasetForNextSentencePrediction [2]. Pada Data collator didefinisikan untuk membantu mengumpulkan sampel dari kumpulan data menjadi objek yang PyTorch tahu cara melakukan backprop, baik untuk permasalahan MLM atau NSP nya. Probailitas MLM nya adalah 15% dan probabilitas NSP nya adalah 50%. Sebelum melakukan pelatihan, dilakukan pendefinisian argumen untuk pelatihan model, berikut pada Gambar 6 adalah argumen yang digunakan untuk melakukan pelatihan model.

```
training_args = TrainingArguments(
    output_dir="./Data Baru/Model-4",
    overwrite_output_dir=True,
    num_train_epochs=1,
    per_gpu_train_batch_size=S,
    save_steps=2_000,
    save_total_limit=2,
    logging_dir='./Data Baru/Model-4/logs'
```

Gambar 6. Argumen Pelatihan

Pada pelatihannya, dilakukan sebanyak dua belas kali run dengan total step nya adalah 399,780 step, dengan global loss sebesar 5.073524031967582.

## 3.3 Pre-Processing

Pre-processing dilakukan dengan tiga tahapan untuk dataset klasifikasi teks, pertama penentuan max length untuk token, kedua melakukan penambahan token spesial ([CLS], [SEP], [UNK], [PAD]), dan ketiga membuat array attention . Berikut pada Gambar 7 merupakan pre-processing pada satu kalimat dengan max length token = 10.

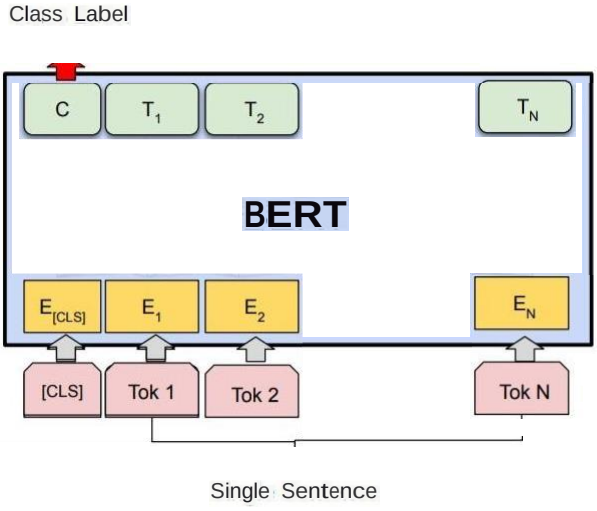
```
     Teks: sang presiden udah mulai ngaco
   Tokens: ['[CLS)', 'sang', 'presiden', 'udah', 'mulai', 'ngaco', '[SEP)', '[PAD)', '[PAD)', '[PAD)')
Input ids: tensor([    3,  2139,  1956,  4022,  2468, 47816,     1,     2,     2,     2))
Attention: tensor([1, 1, 1, 1, 1, 1, 1, 0, 0, 0))
```

Gambar 7. Contoh Pre-Processing

Teks memiliki banyak kata sebanyak 5 kata, pada proses tokenisasi nya terlihat ada penambahan token spesial, yaitu token [CLS], [SEP], [PAD]. Untuk penjelasannya, token [CLS] menandakan awal dari suatu kalimat, token [SEP] menandakan akhir dari kalimat tersebut, dan token [PAD] merupakan penambahan token untuk memaksimalkan length token yang sudah diinisialisasikan. Input ids memiliki informasi id dari tiap token sesuai dengan urutannya. Attention merupakan array yang berisi informasi susunan dari token, dengan membedakan antara token asli dengan angka 1 dan token [PAD] dengan angka 0.

## 3.4 Fine-Tunning

Pada tahap fine-tunning model, untuk melakukan klasifikasi teks dilakukan dengan menggunakan library ktrain [6]. ktrain merupakan wrapper yang digunakan untuk melakukan model-building, model-inspection, dan model-application pada machine learning ataupun deep learning [6]. Pada kasus ini, ktrain dapat mendukung untuk mempermudah melakukan klasifikasi teks pada model BERT. Parameter yang digunakan untuk klasifikasi teks adalah max length = 250, batch size = 6, learning rate = 4e-5, dan epoch = 10. Gambar 8 merupakan contoh jaringan BERT untuk melakukan klasifikasi teks.




Gambar 8. Fine-Tunning

## 3.5 Evaluasi

Evaluasi dilakukan untuk mengukur performansi dari sistem ataupun model yang telah dibangun. Evaluasi dapat dilakukan dengan berbagai macam cara, salah satunya dengan confussion matrix dapat dilihat pada Tabel 1. Hasil prediksi yang telah dilakukan oleh model, dapat di representasikan dengan confussion matrix. Confussion matrix memiliki empat struktur : True Positives adalah data yang hasil prediksinya positif dan sesuai dengan data aktual , True Negatives adalah data yang hasil prediksinya negatif dan sesuai dengan data aktual, False Positives adalah data yang hasil prediksinya positif tetapi pada data aktual negatif, dan False Negatives adalah data yang hasil prediksinya negatif tetapi pada data aktual positif. Dengan adanya confusion matrix ini, selanjutnya dapat menghitung precision, recall, F1-score, dan Macro Average F1-Score. Formula dari parameter-parameter tersebut diberikan pada persamaan (1-4).

Tabel 4. Confusion Matrix Multiclass

| | actual Non Offensive | actual Normal | actual Offensive |
|---|---|---|---|
| predicted Non Offensive | T Non Offensive | F | F |
| predicted Normal | F | T Normal | F |
| predicted Offensive | F | F | T Offensive |

Precision didefinisikan sebagai rasio jumlah dokumen yang relevan yang diambil ke jumlah dokumen yang relevan. Rumus recall adalah sebagai berikut [9] :

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Recall didefinisikan sebagai jumlah dokumen yang relevan yang diambil dibagi dengan jumlah dokumen yang diambil. Rumus precision adalah sebagai berikut [9]

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Setelah mendapat nilai precision dan recall, selanjutnya dapat dihitung nilai F1-Score dengan rumus sebagai berikut :

$$F1Score = \frac{2(Precision * Recall)}{Precision + Recall} \quad (3)$$

Untuk mendapatkan evaluasi keseluruhan pada sistem untuk tiap label, setelah mendapat nilai F1-Score untuk tiap label, maka dapat dilakukan perhitungan Macro Average F1-Score dengan rumus sebagai berikut :

$$Macro - Average - F1Score = \frac{1}{M} \sum_{c=1}^{M} \left( \frac{2(Precision_c * Recall_c)}{Precision_c + Recall_c} \right) \quad (4)$$

Penghitungan nilai F1-score masing masing kelas label dilakukan untuk melihat performasi masing masing label pada model yang dibangun, sedangkan untuk nilai Macro Average F1-Score agar mendapatkan nilai performansi secara keseluruhan untuk model, karena pada macro average tiap label akan dianggap setara kepentingannya, jadi nilainya akan rendah jika model tersebut performansi nya bagus pada label yang dominan sementara performansinya rendah untuk label yang sedikit. buruk pada kelas langka.

## 4. Evaluasi

### 4.1 Hasil Pengujian

Tabel 5. Hasil Pengujian

| Model - Dataset | Macro Average-F1 | F1 Non Offensive | F1 Normal | F1 Offensive |
|---|---|---|---|---|
| scratch - Langsung | 49% | 23% | 93% | 32% |
| scratch - R. Undersampling 50% | 50% | 15% | 92% | 43% |
| scratch - R. Undersampling 45% | 48% | 15% | 91% | 36% |
| scratch - Oversampling (nlpaug) | 45% | 7% | 92% | 35% |
| BertMulti - Langsung | 54% | 11% | 95% | 57% |
| BertMulti - R. Undersampling 50% | 53% | 11% | 93% | 55% |
| BertMulti - R. Undersampling 45% | 51% | 14% | 93% | 47% |
| BertMulti - Oversampling (nlpaug) | 44% | 17% | 94% | 20% |
| Random Forest - Langsung Random Forest | 32% | 0% | 92% | 5% |
| - R. Undersampling 50% Random Forest - | 38% | 15% | 90% | 8% |
| R. Undersampling 45% Random Forest - | 31% | 0% | 91% | 3% |
| Oversampling (nlpaug) | 30% | 3% | 70% | 15% |

### 4.2 Analisis Hasil Pengujian

Dari hasil pengujian didapat bahwa untuk model non-neural seperti random forest, hasil evaluasinya masih kurang baik jika dibandingkan dengan model BERT, serta untuk scratch model yang merupakan model rancangan sendiri terlihat masih belum lebih baik hasilnya dibandingkan dengan model Bert Multilanguage. Kedua model tersebut sudah dapat memperdiksi tiga label kelas yang ada, walau hasilnya masih belum maksimal. Label dengan performansi paling tinggi didapat oleh label Normal dengan performansi di atas 90%, dilanjut dengan label Offensive, dan label dengan performansi terendah adalah label Non Offensive. Untuk bagian per label pada model, prediksi label Non Offensive scratch model lebih baik dibandingkan dengan BERT Multilanguage, tetapi untuk prediksi label normal dan Offensive BERT Multilanguage memiliki hasil yang lebih baik dibandingkan dengan scratch model. Pada Tabel 6 memaparkan hasil confusion matrix untuk scratch - R. Undersampling 50%.

Tabel 6. Confusion Matrix Multiclass scratch - R. Undersampling 50%

| | actual Non Offensive | actual Normal | actual Offensive |
|---|---|---|---|
| predicted Non Offensive | 7 | 17 | 11 |
| predicted Normal | 35 | 749 | 48 |
| predicted Offensive | 14 | 34 | 41 |

Tabel 6 menunjukan untuk kalimat Normal dapat di prediksi dengan baik, dari total 832 kalimat, total diprediksi benar sebanyak 749 kalimat dan prediksi salah sebanyak 83 kalimat, dimana kesalahan prediksi cenderung ke label Offensive sebanyak 48 kalimat. Pada kalimat Offensive, dari total 89 kalimat , total di prediksi benar sebanyak 41 kalimat dan prediksi salah sebanyak 48 kalimat, dimana kesalahan prediksi terbesarnya ke label Normal sebanyak 34 kalimat. Terakhir untuk kalimat Non Offensive dari total 35 kalimat, yang diprediksi benar hanya sebanyak 7 kalimat dan prediksi salah sebesar 28 kalimat, dengan kesalahan prediksi terbesarnya ke label normal sebanyak 17 kalimat. Pada tabel 7 merupakan sampel kecenderungan kata yang muncul pada tiap label untuk mengklasifikasikan kalimat.

Tabel 7. Kata Pada Prediksi - Scratch Model

| Label | Kata | Sekuens dua kata |
|---|---|---|
| Normal | kpai, anak, dan djarum | (pb, djarum), (anak, anak), dan (bulu, tangkis) |
| Non Offensive | onta, ngabalin, dan kencing | (kencing, onta), (zon, coba), dan (youtube, kuncinya) |
| Offensive | kpai, goblok, dan tolol | (tolol, sih), (pb, djarum), dan (kpai, goblok) |

Dilihat dari Tabel 7, untuk prediksi label Normal dan Offensive memiliki irisan pada kata ’kpai’ dan sekuens dua kata pada (pb, djarum). Sehingga ada kemungkinan untuk dua label tersebut akan menjadi salah memprediksi jika ada kalimat yang memiliki antara dua kata atau sekuens tersebut. Pada label Normal, kata nya lebih sering muncul mengenai kata kpai, anak, dan djarum. Pada label Offensive, kata yang mempengaruhi kalimat tersebut menjadi offensive adalah kata kpai, tolol, dan goblok. Untuk label Non Offensive kata seperti onta, ngabalin, dan kencing ini akan dapat terklasifikasikan cenderung sebagai kalimat yang Non Offensive.

Berdasarkan pengujian terhadap hasil dari prediksi data tes, dilakukan analisis terhadap kalimat-kalimat yang belum sesuai diklasifikasikan atau belum sesuai dideteksi oleh model dan didapatkan beberapa alasan sebagai berikut:

1. Terdapat beberapa kalimat yang penulisan kata kasarnya dengan menggabungkan huruf dan simbol, sehingga pada saat dilakukan tokenisasi teks maka penulisan dari kata kasar tersebut berubah. Contoh kata ‘b*kep’ akan dianggap menjadi tiga token [b], [*], [kep], atau ’bl**n’ akan dianggap menjadi empat token [bl], [*], [*], [n].
2. Terdapat beberapa penulisan kata kasar dengan mengganti beberapa hurufnya. Contoh kata ‘bangcatt’, ’geblek’, ’bangsad’.
3. Terdapat beberapa penulisan kata kasar dengan menambahkan beberapa hurufnya. Contoh kata ’goblook’, ’tolooool’, ’guoblog’.
4. Terdapat penggunaan kata kasar yang disingkat atau dihilangkan huruf vokalnya sehingga tidak dapat dikenali kesamaan makna kata tersebut dengan kata kasar yang serupa. Contoh kata ’gblg’.
5. Penggunaan kalimat atau kata dalam bahasa daerah. Contoh kata kimpoi yang memiliki arti hubungan suami istri, atau kalimat ’sakarepmu wae....ramelumelu.....’.

Untuk hasil performansi terbaik dari BERT Multilanguage, didapat pada dataset langsung dengan nilai macro average F1 nya sebesar 54%. Pada Tabel 8 memaparkan confussion matrix dari BertMulti - Langsung.

Tabel 8. Confusion Matrix Multiclass BertMulti - Langsung

| | actual Non Offensive | actual Normal | actual Offensive |
|---|---|---|---|
| predicted Non Offensive | 3 | 20 | 12 |
| predicted Normal | 9 | 803 | 20 |
| predicted Offensive | 6 | 35 | 48 |

Jika dibandingkan hasil confusion matrix antara scratch model dan BERT Multilanguage, pada BERT Multilanguage untuk memprediksi label Normal hasilnya lebih baik 54 kalimat dan untuk label Offensive lebih baik sebanyak 7 kalimat, tetapi untuk prediksi label Non Offensive masih lebih baik scratch model dibanding dengan BERT Multilanguage dengan perbedaan 4 kalimat. Selanjutnya pada tabel 9 merupakan sampel kecenderungan kata yang muncul pada tiap label untuk mengklasifikasikan kalimat di model BERT Multilanguage dari hasil prediksi.

Tabel 9. Kata Pada Prediksi - BERT Multilanguage

| Label | Kata | Sekuens dua kata |
|---|---|---|
| Normal | kpai, anak, dan djarum | (pb, djarum), (anak, anak), dan (bulu, tangkis) |
| Non Offensive | xxx, fuck, dan sadis | (xxx, fuck), (tom, and), dan (jerry, sadis) |
| Offensive | kpai, goblok, dan tolol | (tolol, sih), (pb, djarum), dan (kpai, goblok) |

Pada tabel 9, terlihat perbedaan pada kata dengan bahasa Inggris disini muncul ('fuck'), sedangkan pada scratch model tidak. Kemungkinan ini muncul karena BERT Multilanguage dilatih dengan multi bahasa [2], juga karena prediksi untuk label Non Offensive hanya didapat 3 kalimat dari total 35 kalimat. Untuk label Normal dan Offensive kemunculan kata dan sekuens yang mendominasi kedua label sama seperti kemunculan pada scratch model, hanya frekuensi kemunculannya saja yang sedikit berbeda. Dalam perbandingan modelnya, dilakukan analisis terhadap scratch model dan BERT Multilanguage, berikut ini hasil yang didapatkan mengenai hasil dari BERT Multilanguage lebih baik dibandingkan dengan scratch model.

1.BERT Multilanguage memiliki vocab size sebesar 105,879, sedangkan scratch model memiliki vocab size sebesar 52,000.

2.BERT Multilanguage melakukan train selama 40 epoch. sedangkan scrartch model melakukan train selama 1 epoch dengan duabelas kali run.

3.BERT Multilanguage menggunakan weight decay dalam proses pelatihannya, sedangkan scratch model tidak.

4.Pelatihan BERT Multilanguage menggunakan total 16 TPU chip.

## 5. Kesimpulan

Secara keseluruhan scratch model performansi nya masih kurang baik jika dibandingkan pada model BERT Multilanguage dengan nilai performansi Macro Average F1 untuk Scratch model adalah 50% pada dataset random undersampling 50% dan untuk BERT Multilanguage adalah 54% pada dataset langsung. Kesalahan klasifikasi pada prediksi ditemukan karena adanya penuisan kata atau kalimat yang tidak dipahami oleh model, seperti kata atau kalimat dengan penggabungan huruf dan simbol, pengubahan huruf pada kata, penambahan atau pengurangan huruf pada kata, dan penggunaan bahasa daerah.

Saran atau Future Work nya adalah untuk dataset pada model scratch bisa ditambah lagi sehingga kalimat ataupun teks pada data latih lebih beragam dan pemahaman model dapat lebih baik lagi. Beberapa caranya bisa dengan menerjemahkan corpus yang digunakan oleh HateBERT [1] atau bisa crawling data dari beberapa komentar media sosial lain dengan keyword kata kasar. Pada pelatihan juga bisa lebih lama lagi untuk epochnya dan dapat menggunakan weight decay agar dapat menjaga loss tetap rendah. Pada pengklasifikasian teks untuk data fine-tunning bisa digunakan sistem augmentasi yang lebih baik seperti penggunaan gpt-2 atau penambahan data secara manual untuk label yang masih sedikit agar hasil performansi sistem meningkat.

## Referensi

[1]T. Caselli, V. Basile, J. Mitrovi ć, and M. Granitzer. Hatebert: Retraining bert for abusive language detection in english, 2021.

[2]J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019.

[3]M. O. Ibrohim, E. Sazany, and I. Budi. Identify abusive and offensive language in indonesian twitter using deep learning approach. Journal of Physics: Conference Series, 1196:012041, mar 2019.

[4]D. R. Kiasati Desrul and A. Romadhony. Abusive language detection on indonesian online news comments. In 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pages 320–325, 2019.

[5]E. Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

[6]A. S. Maiya. ktrain: A low-code library for augmented machine learning. arXiv, arXiv:2004.10703 [cs.LG], 2020.

[7]D. Moeljadi. Usage of indonesian possessive verbal predicates: a statistical analysis based on questionnaire and storytelling surveys. APLL-5 conference, 2012.

[8]G. Pitsilis, H. Ramampiaro, and H. Langseth. Detecting offensive language in tweets using deep learning. 01 2018.

[9]V. Raghavan, G. Jung, and P. Bollmann. A critical investigation of recall and precision as measures of retrieval system performance. ACM Transactions on Information Systems, 7:205–229, 07 1989.

[10]A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. CoRR, abs/1706.03762, 2017.

## Lampiran

Keywords kata kasar untuk crawling data pada twitter

Tabel 10. Keywords

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| anjing | babi | monyet | kunyuk | bajingan | asu | bangsat | kontol |
| peler | memek | ngentot | ngewe | perek | pecun | jablay | banci |
| bencong | bego | goblok | idiot | geblek | orang gila | gila | sinting |