Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN)

Nanang Husin
Universitas Negeri Surabaya
Jl. Lidah Wetan, Lidah Wetan, Kec. Lakarsantri, kota Surabaya, Jawa Timur, Indonesia
nangryo@gmail.com

Intisari— **Pertukaran informasi melalui media digital terjadi dalam waktu singkat dan jumlah yang besar. Salah satu dampaknya adalah kemudahan dalam mengakses artikel berita melalui media internet, seperti media Cable News Network (CNN). Artikel berita CNN dikelompokkan ke dalam beberapa kategori. Jumlah kategori yang beragam ini tidak dapat diselesaikan dengan binary classification apabila ingin dikelompokkan. Oleh karena itu, pengelompokkan dapat dilakukan menggunakan metode multi-class classification. Multi-class classification adalah klasifikasi dengan lebih dari dua kelas dengan masing-masing sampel ditugaskan untuk satu label. Algoritma yang digunakan dalam penelitian ini yaitu Random Forest, Naïve Bayes, dan BERT. Random Forest dan Naïve Bayes merupakan algoritma Machine Learning sedangkan BERT merupakan algoritma Deep Learning. Data yang digunakan pada penelitian ini berjumlah 37904 artikel berita CNN dengan 6 kategori, yakni news, business, health, entertainment, sports, dan politics. Penelitian ini bertujuan untuk membandingkan performa dari ketiga algoritma tersebut pada klasifikasi artikel berita CNN. Dari hasil penelitian, diketahui bahwa algoritma BERT memiliki performa lebih baik dari Random Forest dan Naïve Bayes dengan akurasi 0.92 dan macro average f1 score 0.92.**
Kata kunci— Artikel CNN, BERT, Multi-class Classification, Naïve Bayes, Random Forest

Abstract— **The exchange of information through digital media occurs rapidly and in large quantities. One of the impacts is the ease of accessing news articles via the internet, such as through Cable News Network (CNN) media. CNN news articles are categorized into several categories. The diverse number of categories cannot be solved using binary classification if we want to classify them. Therefore, classification can be done using the multi-class classification method. Multi-class classification is a classification with more than two classes, where each sample is assigned to one label. The algorithms used in this study are Random Forest, Naïve Bayes, and BERT. Random Forest and Naïve Bayes are Machine Learning algorithms, while BERT is a Deep Learning algorithm. The data used in this study consists of 37,904 CNN news articles, categorized into 6 categories: news, business, health, entertainment, sports, and politics. The aim of this study is to compare the performance of these three algorithms in classifying CNN news articles. From the research results, it is known that the BERT algorithm performs better than Random Forest and Naïve Bayes, with an accuracy of 0.92 and a macro average F1 score of 0.92.**
Keywords— Article CNN, BERT, Multi-class Classification, Naïve Bayes, Random Forest

I. PENDAHULUAN

Di zaman ini semuanya berkaitan dengan teknologi, kita dapat memperoleh artikel dengan mudah diakses melalui media internet dengan berbagai portal berita yang telah disajikan, salah satunya yaitu CNN (Cable News Network) . Untuk mencari berita yang ingin pembaca ketahui , dapat dengan mengetik sebuah kata kunci pada mesin pencarian website . Sehingga dapat menghasilkan artikel sesuai dengan kata kunci yang diketikkan.

Sebuah artikel adalah tulisan yang berisi laporan atau karangan faktual tentang suatu peristiwa, kejadian, gagasan, atau fakta yang menarik dan penting untuk disampaikan atau dipublikasikan melalui media massa agar dapat dijangkau oleh masyarakat, sehingga dapat memberikan informasi kepada khalayak. Artikel dapat diperoleh dengan berbagai bentuk, mulai dari media cetak, siaran televisi,serta internet.

Artikel tentunya memiliki banyak topik pembahasan yang berbeda, dimana biasanya artikel disajikan dalam beberapa kategori, seperti di bidang kesehatan, olahraga, teknologi, politik, pendidikan , dan masih banyak yang lainnya. Dengan jumlah topikyang beragam ini tentunya tidak dapat diselesaikan dengan binary classification. Oleh karena itu, pengelompokan dengan berbagai kelas ini hendaknya diselesaikan dengan metode multiclass classification.

Beberapa penelitian pernah dilakukan untuk menyelesaikan permasalahan *multi-class classification*. Salah satu nya pada klasifikasi citra cuaca dengan perbandingan berbagai model algoritme SVM , KNN , dan CNN yang mendapatkan hasil akurasi yang paling tepat pada model CNN.

Saat ini, jumlah penelitian yang menggunakan metode klasifikasi multi-kelas masih terbatas. Oleh karena itu, untuk mendapatkan kemajuan yang signifikan dalam proses klasifikasi, penting untuk melakukan pencarian terhadap

penelitian terkini yang berusaha menyelesaikan masalah klasifikasi dan identifikasi dengan melibatkan lebih banyak kelas.

Fokus pada penelitian ini yaitu dapat menggali lebih banyak lagi mengenai metode dalam pemodelan *task* Artificial Intelligence mengenai *multi-class classification*. Sehingga dengan hasil dari penelitian ini, dapat membandingkan berbagai model seperti *Random Forest* dan Naive Bayes yang merupakan model berbasis *Machine Learning* serta BERT yang berbasis *Deep Learning* terhadap dataset kumpulan artikel CNN.

Kelemahan terbesar pada metode machine learning yaitu dataset yang cukup besar tidak terlalu menghasilkan model yang sangat baik, sedangkan metode *deep learning* akan memberikan hasil yang baik apabila menggunakan dataset yang besar, oleh karena itu membutuhkan *hardware* komputasi yang mumpuni.

Kontribusi penulis dalam penelitian ini adalah:

1. Proses pengenalan otomatis yang menggunakan pendekatan kecerdasan buatan
2. Pengimplementasian model *Random Forest* dan Naive Bayes pada *multi-class classification.*
3. Pengimplementasian model BERT Fine Tuning pada *multi-class classification*

II. LATAR BELAKANG

2.1 Random Forest

Pada tahun 2001, Breiman dikenal sebagai orang pertama yang memperkenalkan random forest sebagai metode bagging. Dalam penelitiannya, Breiman mencatat beberapa keunggulan dari random forest, seperti kemampuannya untuk menghasilkan klasifikasi akhir yang baik dengan tingkat kesalahan yang rendah, kemampuan mengatasi jumlah data yang besar dengan baik, serta menjadi algoritma yang efektif dalam penanganan data yang hilang [11]. Kelas klasifikasi dalam random forest ditentukan melalui penggunaan sistem voting dari beberapa decision tree yang telah terbentuk. Keputusan kelas klasifikasi ditentukan oleh decision tree yang mendapatkan suara terbanyak [12].

Random forest merupakan suatu teknik klasifikasi yang menggabungkan beberapa pohon keputusan dengan cara memilih data dan variabel secara acak. Dalam setiap pohon yang terbentuk, random forest menghasilkan sejumlah variabel dependen. Teknik klasifikasi lain seperti decision tree, k-nearest neighbor, artificial neural network, naïve bayes, support vector machine, serta teknik ensemble seperti bagging, adaboosting, random forest, ensemble filtering, dan voting algorithm telah terbukti dapat meningkatkan akurasi pengukuran pada algoritma klasifikasi dan prediksi [13].

2.2 Naive Bayes Classifier (NBC)

Naive Bayes. Algoritma ini membangun tabel probabilitas yang digunakan untuk memperkirakan kemungkinan milik berbagai kelas. Probabilitas dihitung menggunakan rumus dikenal sebagai teorema Bayes, yang menentukan bagaimana peristiwa terkait. Meskipun teorema Bayes bisa mahal secara komputasi, versi yang disederhanakan itu membuat apa yang disebut asumsi "naif" tentang independensi fitur mampu menangani kumpulan data yang sangat besar. Pengklasifikasi Bayesian paling baik diterapkan pada masalah dimana informasi dari banyak atribut harus dipertimbangkan secara bersamaan untuk memperkirakan probabilitas keseluruhan dari suatu

hasil.[4]

Meskipun naïve bayes mengasumsikan independensi antara atribut (tanpa adanya kaitan antara atribut), algoritma ini tetap memiliki performa klasifikasi yang kompetitif.

Asumsi independensi atribut ini jarang terjadi dalam data sebenarnya, namun meskipun asumsi ini dilanggar, naïve bayes masih mam

pu memberikan tingkat performa klasifikasi yang tinggi [4].

Penggunaan Naive Bayes Classifier

a Deteksi intrusi atau anomali dalam jaringan komputer
b Klasifikasi teks, seperti pemfilteran email sampah (spam)
c Mendiagnosis kondisi medis berdasarkan serangkaian gejala yang diamati.[4]

Kelebihan Naïve Bayes

a Sederhana, cepat, dan sangat efektif
b Membutuhkan relatif sedikit contoh untuk pelatihan, tetapi juga bekerja dengan baik dengan jumlah contoh yang sangat banyak
c Mudah untuk mendapatkan perkiraan probabilitas untuk prediksi.[4]

Kekurangan Naïve Bayes

a. Bergantung pada asumsi yang sering salah sama pentingnya dan fitur independen
b. Tidak ideal untuk kumpulan data dengan banyak fitur numerik
c. Kemungkinan yang diperkirakan lebih kecil dapat diandalkan daripada kelas yang diprediksi.[4]

Pemodelan dengan Naive Bayes

Model Naive Bayes dikelompokan berdasarkan tipe dan fungsinya[4]

Gaussian Naive Bayes

Ini adalah bentuk klasifikasi Naive Bayes yang paling dasar, dengan asumsi bahwa data dari setiap label diambil dari distribusi Gaussian yang sederhana [4].

Multinomial Naive Bayes

Salah satu jenis pengklasifikasi Naive Bayes yang berguna adalah Multinomial Naive Bayes, di mana fitur-fiturnya diasumsikan diambil dari distribusi Multinomial yang sederhana. Naive Bayes seperti ini sangat cocok untuk fitur-fitur

yang mewakili jumlah diskrit, seperti klasifikasi kategori dokumen. Dengan menggunakan frekuensi kata-kata yang muncul dalam dokumen, sebuah dokumen dapat dikategorikan ke dalam tema olahraga, politik, teknologi, dan sebagainya [4].

Bernoulli Naive Bayes

Model penting lainnya adalah Bernoulli Naive Bayes dimana fitur diasumsikan biner (0s dan 1s). Klasifikasi teks dengan model 'bag of words' dapat menjadi aplikasi dari Bernoulli Naive Bayes [4].

2.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT adalah sebuah model Deep Learning yang digunakan untuk merepresentasikan kata-kata secara kontekstual dalam prapelatihan pemrosesan bahasa alami (NLP). Model ini dikembangkan oleh Google dan diterbitkan pada tahun 2018. Dalam pelatihan, kata-kata disesuaikan dengan menggunakan Masked Language Model (MLM) dan Transformers dua arah (bidirectional). Sesuai dengan namanya, BERT fokus pada proses encoding dan menghasilkan sebuah model bahasa. Model BERT terdiri dari struktur encoder transformator dua arah dengan beberapa lapisan. BERT hanya menggunakan tumpukan encoder dalam Transformers dan tidak menggunakan tumpukan decoder [1].

BERT menyediakan beberapa model pre-trained, salah satunya adalah model BERT Multilingual. Model ini telah dilatih menggunakan data Wikipedia dalam 104 bahasa yang berbeda. Namun, model BERT Multilingual memiliki beberapa keterbatasan ketika digunakan untuk tugas yang hanya melibatkan satu bahasa. Model ini tidak memiliki deteksi bahasa atau pemilihan bahasa, sehingga tokenizer dapat mencampuradukkan kata-kata dari berbagai bahasa. Selain itu, model BERT English memiliki kamus (vocabulary) dengan ukuran 28.996 token, sedangkan model multilingual yang mencakup 100 bahasa hanya memiliki 119.547 token untuk seluruh bahasa [4]. BERT memiliki keunggulan dibandingkan dengan metode lain, yang meliputi hal-hal berikut.

1. BERT memanfaatkan encoder, prinsip attention, dan melibatkan pemahaman keseluruhan teks sebagai input, bukan hanya mengikuti urutan sekuensial. Hal ini memungkinkan BERT untuk memahami hubungan kontekstual setiap token dengan baik. Selain itu, BERT juga dapat mengatasi teks yang panjang dengan lebih baik daripada RNN karena menggunakan positional encoding saat membaca input [8].
2. BERT baik digunakan untuk dataset berukuran kecil karena telah dipre-train sehingga hanya memerlukan fine-tuning. [8]
3. BERT menggunakan struktur Transformer dan memiliki jumlah parameter yang lebih sedikit dibandingkan dengan model CNN, sehingga dapat mencapai kinerja yang lebih baik dalam waktu yang lebih efisien [8].

Ada dua tahapan framework pada BERT yaitu *PreTraining* dan *Fine Tuning*. Kita dapat melakukan *fine-tune* pada *pretrained* model. Model Bert ini akan di fine-tune ketika di train oleh pihak ketiga dan di upload ke *Hugging Face*. Dimana Hugging face merupakan library python yang menyediakan berbagai model yang telah dilatih sebelumnya dan dapat digunakan untuk berbagai jenis tugas dalam Pemahaman Bahasa Alami (NLU) dan Pembangkitan Bahasa Alami (NLG). Kemudian, model pre-trained dapat di fine-tuned pada berbagai jenis tugas Natural language Processing (NLP) [1].

1. Masked Language Modeling (MLM)

Metode Masked Language Model (MLM) digunakan untuk mengisi kekosongan dalam kalimat, di mana model menggunakan kata-kata konteks di sekitar token yang tersembunyi untuk memprediksi kata yang seharusnya ada di tempat tersebut. Sementara itu, metode Next Sentence Prediction (NSP) digunakan untuk memprediksi kalimat berikutnya dengan menggunakan dua model yang diberikan. Pada BERT Pre-Training model, bagian MLM ini menjadi metode yang perlu diperhatikan. Dimana pada tahapan ini terdapat token yang ditutupi (mask) dari kata-kata input secara acak, kemudian model akan memprediksi kata-kata yang ditutupi tersebut. Masked Language modelling ini tidak memperhatikan pelatihan model dengan urutan kata yang nyata (terlihat) yang diikuti oleh urutan masked dalam memprediksinya namun akan diproses secara acak oleh mask tersebut [1].

2. BERT Fine Tuning

Setelah proses pre-training data, BERT melanjutkan dengan tahap fine-tuning. Fine-tuning merupakan langkah di mana model yang sudah dilatih sebelumnya digunakan untuk melatih model pada tugas yang berbeda. Proses ini juga dikenal sebagai Transfer Learning. Transfer Learning memanfaatkan pengetahuan yang telah ada dalam model yang sudah dilatih untuk memecahkan masalah yang serupa dengan melakukan modifikasi dan memperbarui parameter sesuai dengan dataset baru [1]. Pada tahap fine-tuning, model BERT diinisialisasi dengan parameter pretrain dan kemudian disesuaikan dengan menggunakan data yang memiliki label [6].

3. BERT Tokenization

BERT menggunakan WordPiece tokenizer untuk menghasilkan dictionary hingga 30000 token. Kata asli akan dipecah menjadi sub kata dan karakter yang lebih kecil atau menjadi token. Kemudian token tersebut harus dipetakan ke indeks pada kosakata tokenizer.

4. Required Formatting

BERT memiliki beberapa format tertentu yang harus diikuti, yaitu Menambahkan special token di awal dan akhir setiap kalimat, menambahkan padding dan memotong semua kalimat sehingga memiliki panjang yang konstan, serta membedakan token padding dengan attention mask. Dimana Attention mask merupakan array 1 dan 0 untuk dapat membedakan padding dengan memberi tahu mekanisme self

attention di BERT untuk tidak memasukkan token [PAD] ke dalam interpretasi kalimat [8].

5. Sentence length dan Special Tokens

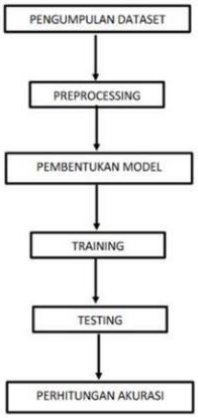
Kalimat pada dataset tentunya memiliki panjang kalimat yang bervariasi. Dengan demikian, BERT memiliki 2 batasan yaitu Semua kalimat harus memiliki panjang yang sama. Oleh karena itu perlu dilakukan padding atau truncate pada setiap kalimat. Kemudian, panjang kalimat maksimalnya yaitu 512 token .

Pada setiap akhir kalimat perlu ditambahkan dengan token [SEP]. Kemudian, pada setiap awal kalimat perlu ditambahkan token [CLS] untuk tugas klasifikasi teks. Serta menambahkan panjang kalimat dapat dilakukan dengan menambahkan token [PAD].

III. METODOLOGI PENELITIAN

Dalam saddsadasdasd

Metodologi yang dilakukan terdiri dari pengumpulan dataset, preprocessing data, pembentukan model klasifikasi, training model klasifikasi, testing dan evaluasi model, Gambar 1. menunjukkan alur metodologi.



Gambar 1. Kerangka Alur Berfikir

3.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian komparasi model ini adalah data artikel berita *Cable News Network* (CNN) dari tahun 2011 sampai 2022, dataset ini memiliki 37904 baris artikel yang dikumpulkan menggunakan web crawler, yang berisikan data Author, Publication date, Category, Article Section, Url source, Headline, Description, Keywords, Second headline dan Article text yang ditunjukkan pada Gambar 2.

```
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Index            37949 non-null  int64
 1   Author           37949 non-null  object
 2   Date published   37949 non-null  object
 3   Category         37949 non-null  object
 4   Section          37949 non-null  object
 5   Url              37949 non-null  object
 6   Headline         37949 non-null  object
 7   Description      37949 non-null  object
 8   Keywords         37949 non-null  object
 9   Second headline  37949 non-null  object
 10  Article text     37940 non-null  object
dtypes: int64(1), object(10)
memory usage: 3.2+ MB
```

Gambar 2. Data column dataset CNN

Pada gambar diatas terdapat baris *column* yang berisikan judul dari data tersebut dan juga terdapat keterangan tipe data, index bertipe data Integer yang dapat menampung data angka dan author sampai article text bertipe data object yang dapat menampung data string atau text.

3.2 Preprocessing Data

Tahap Preprocessing data adalah proses *cleaning* dataset dari duplikasi data, pengisian data yang kosong, memeriksa *inkonsisten* data dan memperbaiki kesalahan pada data, biasa data yang kosong disebabkan oleh data baru yang belum ada informasinya.Selain itu pada proses ini juga dilakukan case folding, menghapus simbol, hingga stemming [18].

3.2.1 Labelling Data

Proses pelabelan data diperlukan untuk menentukan kelas daripada jenis artikel dalam kumpulan artikel CNN yaitu apakah deskripsi artikel tersebut termasuk kedalam kelas berlabel berita, olahraga, bisnis,kesehatan,politik, dan hiburan.

3.2.2 Split data testing dan training

Split data testing dan training adalah teknik mengevaluasi kinerja dari algoritma machine learning yang biasa digunakan untuk permasalahan klasifikasi atau regresi dan semua algoritma supervised machine learning.

Prosedur ini adalah proses pembagian dataset subset menjadi dua. Subset pertama digunakan agar sesuai

dengan model dan disebut sebagai training data. Subset kedua tidak digunakan untuk melatih model sebagai gantinya, elemen input dari kumpulan data diberikan ke model, kemudian prediksi dibuat dan dibandingkan dengan nilai yang diharapkan.

Kumpulan data kedua ini disebut sebagai testing data

3.2.3 Term Frequency Inverse Document Frequency (TF-IDF)

Metode TF-IDF adalah sebuah metode yang digunakan untuk menghitung bobot setiap kata yang umum digunakan dalam proses information retrieval. Metode ini terkenal karena efisiensinya, kemudahan implementasinya, dan hasil yang akurat [17].

TF-IDF merupakan metode yang digunakan untuk memberikan bobot pada hubungan antara kata-kata (term) dengan dokumen. Metode ini menggunakan ukuran statistik untuk mengevaluasi seberapa penting suatu kata dalam sebuah dokumen atau dalam sekelompok kata. Dalam kasus dokumen tunggal, setiap kalimat dianggap sebagai dokumen terpisah. Frekuensi kemunculan kata dalam dokumen menunjukkan tingkat kepentingan kata tersebut dalam dokumen tersebut, sementara frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut dalam seluruh dokumen. Bobot kata akan semakin tinggi jika kata tersebut sering muncul dalam suatu dokumen dan semakin rendah jika kata tersebut muncul dalam banyak dokumen [17].

Pada algoritma TF-IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus yaitu :

Wdt = tfdt * Idft

Dimana: Wdt= bobot dokumen ked terhadap kata ket tfdt = banyaknya kata yang dicari pada sebuah dokumen Idft = Inverse Document Frequency (log (N/df)) N = total dokumen df = banyak dokumen yang mengandung kata yang

dicari.[17]

3.2.4 Balancing data

Data Imbalance/data tidak seimbang merupakan kondisi dimana suatu kelompok kelas memiliki jumlah data yang jauh berbeda dibandingkan dengan kelas lainnya [19]. Dataset artikel berita CNN tergolong tidak seimbang karena jumlah artikel pada tiap kategori memiliki perbedaan yang terlampau jauh. Penelitian ini menggunakan *library* "imblearn" dengan metode RandomUnderSampler untuk menyeimbangkan data pada tiap kategori berdasarkan jumlah kateori terkecil. Gambar 3 menunjukkan Jumlah data pada setiap kategori sebelum dan setelah dilakukan *balancing data.*

Gambar 3. Balancing data

3.2.5 Pre-processing BERT

1.Case Folding

Tahapan ini dilakukan dengan *lower case*, *remove white spaces*, *remove url*, hingga melakukan *stemming* terhadap dataset yang digunakan. Selain itu juga dilakukan *filtering* data yang digunakan untuk menghilangkan atau menghapus data duplikat dan memilih attribute apa saja yang akan digunakan, guna merapikan dataset.

2.Tokenization & Input Formatting

Proses tokenisasi dilakukan agar setiap record di dalam data tersebut, diubah menjadi setiap kata yang berdiri sendiri atau dengan kata lain untuk memisah-misahkan kata. Setiap potongan kata tersebut dinamakan dengan token [21]. BERT menggunakan *wordPiece* tokenizer untuk menghasilkan dictionary. Token akan dipetakan ke indeks pada kosakata tokenizer. Selain itu,terdapat proses padding dan truncation, yang bertujuan memotong kata pada kalimat yang ukurannya lebih panjang daripada kalimat yang diset. Serta terdapat attention mask yaitu array 1 dan 0 untuk membedakan padding dengan memberitahu mekanisme self-attention pada BERT untuk tidak memasukkan token [PAD] ke dalam interpretasi kalimat [21].

3.Split data Testing dan Training

Dalam pengerjaan model BERT ini,data dibagi data training dan data validation, dimana 85% digunakan sebagai data training dan 15% sebagai data testing.

4.Balancing data

Dataset yang didapatkan memiliki jumlah kelas target yang berbeda / imbalanced. Dengan demikian perlu dilakukan sampling terhadap dataset yang digunakan. Dalam hal ini, dilakukan tahapan *random under sampling*. Teknik ini menggunakan pendekatan yang mempertimbangkan perbedaan antara jumlah kelas mayoritas dan minoritas untuk memilih dataset. Jika terdapat perbedaan dalam jumlah kelas, maka dataset dari kelas mayoritas akan dihapus secara acak sampai jumlahnya sejajar dengan kelas minoritas [20].

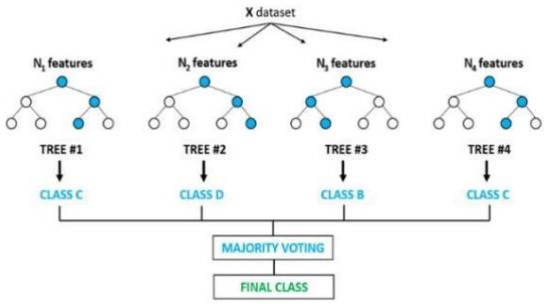
3.3 Pembentukan Model

Pada penelitian ini menggunakan tiga model atau algoritma, dengan dua algoritma machine learning Random Forest dan Naive Bayes Classifier, dan satu algoritma deep learning yaitu Bidirectional Encoder Representations from Transformers (BERT).

3.3.1 Random Forest

Random Forest merupakan pengembangan dari metode Decision Tree yang menggunakan beberapa Decision Tree. Setiap Decision Tree dalam Random Forest dilatih menggunakan sampel individu dan setiap atribut dipisahkan pada pohon yang dipilih secara acak dari subset atribut yang ada. Random Forest memiliki beberapa kelebihan, seperti meningkatkan akurasi ketika terdapat data yang hilang, menghasilkan tingkat error yang rendah, mampu mengatasi data pelatihan dalam jumlah besar secara efisien, serta tahan terhadap outliers. Selain itu, Random Forest juga efisien dalam penyimpanan data dan melakukan seleksi fitur untuk meningkatkan performa model klasifikasi [10].

Dalam prosesnya, Random Forest menggunakan metode pemisahan biner rekursif untuk mencapai simpul akhir dalam struktur pohon. Pohon-pohon klasifikasi dan regresi dihasilkan secara independen dalam Random Forest, dengan subset yang dipilih secara acak melalui bootstrap dari sampel pelatihan dan variabel input di setiap simpul [10].



Gambar 4. Representasi algoritma random forest

3.3.2 Naive Bayes Classifier (NBC)

Metode tambahan yang diterapkan dalam penelitian ini adalah Naive Bayes Classifier (NBC). NBC adalah suatu metode yang sesuai digunakan untuk melakukan klasifikasi dalam bentuk biner maupun multikelas. Metode ini, juga dikenal sebagai Naive Bayes Classifier, menerapkan pendekatan klasifikasi berbasis pengawasan (supervised) dengan memberikan label kelas kepada instance menggunakan probabilitas bersyarat. Penelitian ini menggunakan *software* Jupyter Notebook dengan GPU i3 530 @ 2.93 GHz, dengan library Scikit-Learn, Pandas dan Numpy serta Metode Multinomial Naive Bayes.

Metode Multinomial Naive Bayes merupakan variasi lain dari Naive Bayes. Metode ini mengasumsikan bahwa semua atribut saling bergantung satu sama lain mengingat konteks kelas, dan mengabaikan semua dependensi antar atribut[5].

$P(C) = count\ (c)+K/N+K\ .\ |classes|$ dimana:

Count : jumlah kemunculan atribut pada kelas tertentu
K : nilai parameter
Count (c) : jumlah kemunculan kelas pada sampel c

N : jumlah total kejadian
| | : jumlah atribut pada sampel

Pada tahapan preprocessing, peneliti melakukan split data dengan module Scikit-Learn train_test_split, yaitu module yang dapat membagi dataset menjadi data train dan test dengan pembagian 80% untuk data training dan 20% untuk testing, serta melakukan nilai frekuensi dengan TFIDF.

3.3.3 Bidirectional Encoder Representations from Transformers (BERT)

Salah satu metode yang digunakan dalam penelitian ini yaitu menggunakan metode BERT (Bidirectional Encoder Transformers for Language Understanding). BERT adalah model representasi bahasa yang menghasilkan model pre-train representasi bidirectional dari teks yang tidak berlabel dengan mengkondisikan dari kedua konteks di semua layer.

Dalam hal ini penelitian menggunakan BERT Fine Tuning. BERT Fine Tuning atau Transfer memanfaatkan / mentransfer pengetahuan model yang sudah dilatih untuk dapat menyelesaikan permasalahan lain yang serupa dengan memodifikasi serta mengupdate parameternya sesuai dengan dataset yang baru. Dengan kata lain pre-trained model yang telah dilatih dengan data besar dapat di *fine tuned* sesuai dengan dataset kita. Dalam step *pre-train*, dilakukan training model terhadap data yang tidak berlabel, sedangkan dalam step *fine-tuning* model BERT diinisialisasi dengan parameter *pre-train*, dan semua parameter tersebut disesuaikan dengan menggunakan data yang berlabel [1].

Penelitian ini menggunakan transformers dari *Hugging face library* yang mana akan menyediakan pytorch yang akan bekerja pada tugas BERT ini.

Kemudian *hugging face* menyediakan model yang sudah dilatih / pretrained model, pada penelitian ini digunakan *Bert For Sequence Classification*. Dimana Bert For Sequence Classification adalah model bert dengan tambahan layer on top untuk klasifikasi yang digunakan pada pengklasifikasian kalimat multilabel serta multiclass. Layer BERT-base yang digunakan dalam penelitian ini memiliki 12-layer, 768-hidden, 12-heads, 110M parameters [1]. Penelitian ini dilakukan menggunakan google Colab dengan GPU Tesla T4 dengan library pytorch serta transformers.

Penelitian ini menggunakan tokenizer WordPiece dengan *bert-base-uncased.* Proses tokenisasi dilakukan agar setiap *record* di dalam data tersebut, diubah menjadi setiap kata yang berdiri sendiri atau dengan kata lain untuk memisah-misahkan kata. Setiap potongan kata tersebut dinamakan dengan token [3]. Pada proses tokenizer ini dihasilkan token ids yaitu daftar id numerik untuk teks tokenized. Selain itu, terdapat special token dalam pengerjaan model BERT. Dimana token CLS bertugas dalam tugas klasifikasi teks. Berikut tabel 3.1. menjelaskan tokenized dan token ids pada kalimat original dari dataset.

Tabel 1.1 Tokenized Data

Original	Tokenized	Token IDs
The e commerce boom has exacerbated a global truck driver shortage, but could autonomous trucks help fix the problem	‘the', 'e', 'commerce', 'boom', 'has', 'ex', '##ace', '##rba', '##ted', 'a', 'global', 'truck', 'driver', 'shortage', ',', 'but', 'could', 'autonomous', 'trucks', 'help', 'fix', 'the', 'problem',	1996, 1041, 6236, 8797, 2038, 4654, 10732, 28483, 3064, 1037, 3795, 4744, 4062, 15843, 1010, 2021, 2071, 8392, 9322, 2393, 8081, 1996, 3291

3.4 Evaluasi Model

Confusion matrix digunakan untuk menghitung berbagai *performance metrics* untuk mengukur kinerja model yang telah dibuat. Pada penelitian ini *performance metrics* yang digunakan adalah *accuracy*. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, *accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai *accuracy* dapat diperoleh dengan persamaan dibawah ini.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 5 . Persamaan untuk mencari nilai accuracy permasalahan.

III. HASIL DAN PEMBAHASAN

Bagian ini akan dijelaskan mengenai data, metode, serta hasil yang didapatkan berdasarkan eksperimen yang telah dilakukan. Peneliti melakukan eksperimen dengan 3 model dengan berbeda, yaitu Random Forest, Naïve Bayes, dan BERT.

4.1 Random Forest

Metode random forest yang digunakan diatur dengan parameter random_state=5. Berikut adalah matrik evaluasi hasil pelatihan model menggunakan algoritma random forest ditunjukkan oleh Gambar 6

```
Accuracy score on train:  1.0

Accuracy score on test:  0.7903225806451613

Classification report:
              precision    recall  f1-score   support

           0       0.78      0.76      0.77        62
           1       0.69      0.84      0.76        62
           2       0.77      0.81      0.79        62
           3       0.73      0.56      0.64        62
           4       0.85      0.84      0.85        62
           5       0.92      0.94      0.93        62

    accuracy                           0.79       372
   macro avg       0.79      0.79      0.79       372
weighted avg       0.79      0.79      0.79       372
```

Gambar 6. Matrik evaluasi hasil pelatihan model menggunakan metode random forest

Pada gambar diatas terlihat bahwa akurasi pada *training* bernilai sempurna, yakni 1, sedangkan akurasi pada *testing* bernilai 0,79. Dapat disimpulkan bahwa terjadi *overfitting* pada model. Hal tersebut terjadi pada saat *training*, dimana algoritma mampu memahami data dengan baik, tetapi pada saat *testing* algoritma tidak sanggup untuk melakukan klasifikasi.

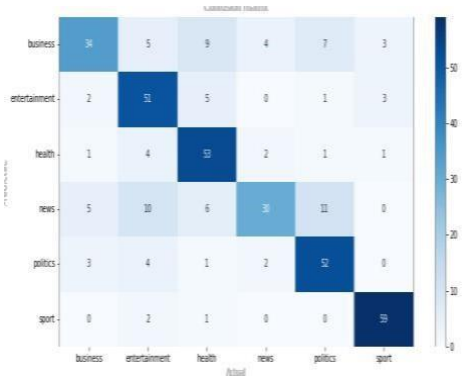
Random forest memang cocok diterapkan pada data dalam jumlah besar. Namun, tanpa penanganan yang tepat maka sering terjadi *overfitting* maupun *underfitting*. Oleh karena itu, diperlukan penanganan yang tepat pada algoritma random forest, salah satunya dengan menggunakan kombinasi parameter yang tepat. Penelitian ini akan mencari kombinasi parameter terbaik untuk mengoptimalkan hyperparameter dalam algoritma random forest. Metode yang digunakan untuk pengoptimalan hyperparameter adalah *search crossvalidation* (searchCV). SearchCV adalah metode pemilihan kombinasi model dan hyperparameter dengan cara menguji coba satu persatu kombinasi dan melakukan validasi untuk setiap kombinasi. Tujuannya adalah menentukan kombinasi yang menghasilkan performa model terbaik yang dapat dipilih untuk dijadikan model untuk klasifikasi. Algoritma yang digunakan pad searchCV adalah randomizedsearchcv yang berguna untuk mencari kombinasi parameter terbaik dari rangkaian parameter yang diberikan secara acak.

Pencarian kombinasi parameter memerlukan waktu yang lama. Penelitian ini membutuhkan waktu 200 menit untuk *fitting* 100 *folds* pada setiap 10 kandidat, dengan total 1000 *fits*. Kemudian diperoleh parameter terbaik, yakni {'n_estimators': 576, 'min_samples_split': 2, 'min_samples_leaf': 3, 'max_features': 'sqrt', 'max_depth': 5, 'bootstrap': True}. Parameter tersebut kemudian di *fit*-kan kedalam algoritma random forest. Berikut adalah matrik evaluasi hasil pelatihan model menggunakan algoritma random forest ditunjukkan oleh Gambar 6.

```
Accuracy score on train:  0.8100664767331434

Accuracy score on test:  0.7446236559139785
Classification report:
              precision    recall  f1-score   support

           0       0.73      0.52      0.60        62
           1       0.68      0.82      0.74        62
           2       0.71      0.87      0.78        62
           3       0.76      0.45      0.57        62
           4       0.73      0.84      0.78        62
           5       0.87      0.97      0.92        62

    accuracy                           0.74       372
   macro avg       0.75      0.74      0.73       372
weighted avg       0.75      0.74      0.73       372
```

Gambar 7. Matrik evaluasi hasil pelatihan model menggunakan metode random forest dan pengoptimalan hyperparameter

Pada tabel diatas terlihat bahwa jarak antara akurasi pada training dan testing tidak terlalu jauh sehingga dapat disimpulkan bahwa dalam model tidak terjadi overfitting maupun underfitting. Berikut Gambar 8 adalah confusion matrix dari hasil klasifikasi.



Gambar 8. Confusion matrix hasil klasifikasi ke dalam enam kategori dengan metode random forest

Dari gambar diatas dapat terlihat bahwa klasifikasi ke dalam enam kategori berhasil dilakukan dengan nilai akurasi pada *testing* 0,74. Kategori *sport* memiliki akurasi tertinggi dan kategori *news* memiliki akurasi terendah.

4.2 Naive Bayes Classifier (NBC)

Untuk model NBC digunakan model MultinomialNB, berikut adalah matriks evaluasi hasil pelatihan model menggunakan algoritma NBC ditunjukan oleh Gambar 9.

```
Accuracy score on train:  0.7872744539411206
Accuracy score on test:  0.7876344086021505
Classification report:
              precision    recall  f1-score   support

           0       0.80      0.73      0.76        62
           1       0.70      0.82      0.76        62
           2       0.75      0.82      0.78        62
           3       0.78      0.61      0.68        62
           4       0.81      0.89      0.85        62
           5       0.91      0.85      0.88        62

    accuracy                           0.79       372
   macro avg       0.79      0.79      0.79       372
weighted avg       0.79      0.79      0.79       372
```

Gambar 9. Matrik evaluasi hasil pelatihan model menggunakan metode random forest dan pengoptimalan hyperparameter

Pada gambar di atas terlihat bahwa jarak antara akurasi pada *training* dan *testing* tidak terlalu jauh sehingga dapat disimpulkan bahwa dalam model tidak terjadi *overfitting* maupun *underfitting*.

4.3 Bidirectional Encoder Representations from Transformers (BERT)

Pada uji coba BERT, peneliti melakukan *training* dan *evaluasi* pada 3 *epoch*, dimana *epoch* ini berfungsi untuk menentukan beberapa kali model dapat melihat dataset secara keseluruhan. Rata-rata Waktu eksekusi pada tiap *epoch* yang dibutuhkan adalah 816 detik dan menghasilkan rata-rata akurasi data 0.93 pada tiap *epoch.* Ratarata *Training loss* pada *epoch* pertama adalah 0.33 dan *epoch* kedua mendapatkan rata-rata *training loss* 0.17, serta di *epoch* ketiga sebesar 0.10 seperti pada gambar 10 BERT membutuhkan waktu yang lebih lama dalam training dibandingkan dengan Random Forest dan NBC. Namun, BERT memiliki akurasi yang paling baik dan performa yang cukup signifikan dibandingkan dengan Random Forest dan NBC.

Gambar 10. Training Loss tiap epoch BERT

Meskipun model ini menghasilkan performa yang jauh lebih baik daripada model Random Forest dan NBC, namun model ini juga memiliki beberapa kekurangan. Diantaranya yaitu masa *training* yang memakan waktu cukup lama.

Berikut grafik perbandingan performa dari ketiga model yang di uji coba dalam penelitian ini.

Tabel 2. Perbandingan Performa

Algoritma	Accurasi		Macro AVG F1 Score
	Training	Testing	
Random Forest	81%	74%	73%
Naïve Bayes	78%	78%	79%
BERT	93%	92%	92%

IV. KESIMPULAN

Berdasarkan uji coba yang dilakukan dalam penelitian ini, kesimpulan yang didapat, algoritma/model BERT memiliki performa yang terbaik dalam melakukan klasifikasi pada dataset artikel berita CNN dari tahun 2011 hingga 2022. Performa yang didapatkan oleh algoritma BERT adalah akurasi training sebesar 93% dan akurasi testing sebesar 92% serta marco avg dari f1 score 92%, jadi algoritma BERT cukup baik dalam melakukan klasifikasi teks dan artikel berita yang memiliki data yang cukup banyak.

Saran untuk penelitian selanjutnya adalah mencari dataset yang sudah balance dan menggunakan perangkat dengan spesifikasi yang cukup sehingga dalam proses lebih cepat dalam menghitung akurasinya.

REFERENSI

[1] Rothman, D. (2021). Transformers for Natural Language Processing. Birmingham,Mumbai: Packt Publishing Ltd.

[2] MCMahan, D. r. (2019). Natural Language Processing with Pytorch. Gravenstein Highway North, Sebastopol: O'Reilly Media, Inc.

[3] Putra, Jan Wira. (2019). Pengenalan Konsep Pembelajaran Mesin dan Deep Learning. Tokyo:Jepang

[4] R. Bali, D. Sarkar, B. Lantz, C. Leismeter. (2016).R: Unleash Machine Learning Techniques. Packt Publishing Ltd.

[5] Bunga, Meilani T. H. (2018).Multinomial Naive Bayes Untuk Klasifikasi Status Kredit Mitra Binaan Di Pt. Angkasa Pura I Program Kemitraan. Nusa Tenggara Timur

[6] Cindy Alifia Putri, A. A. (2020). Analisis Sentimen Review Film Berbahasa Inggris Dengan Pendekatan Bidirectional Encoder Representations from Transformers. Vol. 6, No. 2, Maret 2020, 181-192.

[7] Alan Tusa Bagus W, D. H. (2021).Klasifikasi emosi pada teks dengan menggunakan metode Deep Learning. Vol. 6, Special Issue No. 1, November 2021, 547-553.

[8] Febrian Adhi Pratama, A. R. (2020). Identifikasi Komentar Toksik dengan BERT. *Vol.7,No.2 Agustus 2020*, 7941 - 7949.

[9] Chollet, F. (2018). Deep Learning with Python. Shelter Island, NY: Manning Publications Co.

[10] S. Devella, Y. Yohannes dan F. N. Rahmawati, "Implementasi Random Forest Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT," JATISI (Jurnal Tek. Inform. dan Sist. Informasi), vol. 7, no. 2, pp. 310-320, 2020.

[11] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

[12] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995.

[13] Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Educational Data to Predict Student's Academic Performance using Ensemble Methods.

International Journal of Database Theory and Application , 9 119- 136.

[14] Kumari, P., Jain, P. K., & Pamula, R. (2018). An Efficient use of Ensemble Methods to Predict Students Academic Performance. 4th International Conference on Recent Advances in Information Technology (RAIT) .

[15] Pan, Q., Zhang, Y., Zuo, M., Xiang, L., & Chen, D. (2016). Improved Ensemble Classification Method of Thyroid Disease Based on Random Forest. 2016 8th International Conference on Information

Technology in Medicine and Education (ITME).

[16] Rahman, M. H., & Islam, M. R. (2017). Predict Student's Academic Performance and Evaluate the Impact of Different Attributes on the Performance Using Data Mining Techniques. 2nd International Conference on Electrical & Electronic Engineering (ICEEE) .

[17] Ria, Victor, Hendra, Taslimun. (2018). "Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web" (Jurnal Teknik Informatika Vol 11 No. 2)

[18] Kusrini. and E. Luthfi. (2019). "Algoritma Data Mining" . Yogyakarta: ANDI

[19] Akbar, Khafid & Hayati, Mardhiya. (2020). Data Balancing untuk Mengatasi Imbalance Dataset pada Prediksi Produksi Padi. Jurnal Ilmiah Intech : Information Technology Journal of UMUS. 2. 10.46772/intech.v2i02.283.

[20] Erna Irawan, R. S. (2017). Penggunaan Random Under Sampling untuk Penanganan Ketidakseimbangan Kelas pada prediksi Cacat Software berbasis Neural Network. Vol. 1, No. 2, 92-100.