



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

Deteksi Berita Palsu Otomatis Berbahasa Indonesia Menggunakan BERT

Aufa Nabil Amiri
NRP 0721 17 4000 0029

Dosen Pembimbing
Reza Fuad Rachmadi ST., MT., Ph.D
Prof. Dr. Ir. Mauridhi Hery Purnomo, M. Eng

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

Deteksi Berita Palsu Otomatis Berbahasa Indonesia Menggunakan BERT

Aufa Nabil Amiri
NRP 0721 17 4000 0029

Dosen Pembimbing
Reza Fuad Rachmadi ST., MT., Ph.D
Prof. Dr. Ir. Mauridhi Hery Purnomo, M. Eng

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

[Halaman ini sengaja dikosongkan]



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC184801

Automatic Indonesian Hoax News Detection Using BERT

Aufa Nabil Amiri
NRP 0721 17 4000 0029

Advisor
Reza Fuad Rachmadi ST., MT., Ph.D
Prof. Dr. Ir. Mauridhi Hery Purnomo, M. Eng

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2021

[Halaman ini sengaja dikosongkan]

ABSTRAK

Nama Mahasiswa : Aufa Nabil Amiri
Judul Tugas Akhir : Deteksi Berita Palsu Otomatis Berbahasa Indonesia Menggunakan BERT
Pembimbing : 1. Reza Fuad Rachmadi ST., MT., Ph.D
2. Prof. Dr. Ir. Mauridhi Hery Purnomo, M. Eng

Berita palsu atau yang biasa disebut hoaks adalah suatu yang hal yang sering melanda Indonesia. Dengan adanya sosial media, suatu berita palsu dapat memiliki tingkat penyebaran yang sangat luas. Selain itu, masyarakat Indonesia memiliki tingkat kecenderungan untuk menyebarkan berita palsu yang cukup tinggi. Sehingga, suatu metode pendeteksi berita palsu harus ada. Penelitian ini memanfaatkan algoritma BERT yang digunakan untuk mendeteksi apakah suatu berita adalah berita hoaks atau tidak secara otomatis. Dari suatu teks yang mentah, akan dilakukan tokenisasi sebelum akhirnya dimasukkan ke dalam algoritma BERT. Selanjutnya, keluaran dari BERT akan dijadikan sebagai inputan dari algoritma klasifikasi Linear Regression. Barulah pada saat ini, kita bisa mendapatkan klasifikasi apakah suatu teks itu berupa berita hoaks atau tidak. Tujuan dari penelitian ini adalah untuk membuat sebuah model yang dapat digunakan untuk melakukan klasifikasi suatu teks apakah termasuk ke dalam berita hoaks atau tidak.

Kata Kunci: BERT, Hoaks, Klasifikasi, Linear Regression.

[Halaman ini sengaja dikosongkan]

ABSTRACT

Name : Aufa Nabil Amiri
Title : *Automatic Indonesian Hoax News Detection Using BERT*
Advisors : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D
2. Prof. Dr. Ir. Mauridhi Hery Purnomo, M. Eng

Fake news or usually called hoax, is one of things that often plaguing Indonesia. With a social media, a fake news can spread wider and faster than ever before. On another note, Indonesian people have quite a high tendencies to share fake news. Based on that note, we are in dire need of a method to detect fake news. This research is using BERT method to automatically detect whether a news can be considered as hoax or not. From a raw text, we are doing a tokenization process before we feed the text to the BERT method. Next, the pooled output of the BERT is being used as the input for Linear Regression, a tested-and-true method for classifying task. Now that we have pass-through all those steps, we can determine whether a text is a hoax or not. The purpose of this research is to create a machine learning model to help the people to determine whether a text can be considered as hoax or not.

Keywords: BERT, Hoax, Fake News Classification, Linear Regression.

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|---|------------|
| ABSTRAK | i |
| ABSTRACT | iii |
| DAFTAR ISI | v |
| DAFTAR GAMBAR | ix |
| DAFTAR TABEL | xi |
| 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Permasalahan | 2 |
| 1.3 Tujuan | 2 |
| 1.4 Batasan Masalah | 2 |
| 1.5 Sistematika Penulisan | 3 |
| 2 TINJAUAN PUSTAKA | 5 |
| 2.1 Berita Palsu | 5 |
| 2.2 <i>Machine Learning</i> | 5 |
| 2.2.1 <i>Supervised Learning</i> | 6 |
| 2.2.2 <i>Unsupervised Learning</i> | 6 |
| 2.2.3 <i>Reinforcement Learning</i> | 7 |
| 2.3 <i>Deep Learning</i> | 7 |
| 2.4 BERT | 8 |
| 2.5 Metode Analisa Performa | 10 |

| | | |
|----------|--|-----------|
| 2.5.1 | <i>Recall</i> | 10 |
| 2.5.2 | <i>Precision</i> | 10 |
| 2.5.3 | <i>F1-Score</i> | 11 |
| 2.5.4 | <i>Confusion Matrix</i> | 12 |
| 3 | DESAIN DAN IMPLEMENTASI | 15 |
| 3.1 | Desain Sistem | 15 |
| 3.2 | Alur Kerja | 16 |
| 3.3 | Akuisisi Data | 16 |
| 3.3.1 | Sumber Akuisisi Data | 17 |
| 3.3.2 | Proses Akuisisi Data | 17 |
| 3.4 | <i>Preprocessing</i> | 21 |
| 3.5 | Proses <i>Training</i> | 24 |
| 3.6 | Proses Pengujian | 25 |
| 3.7 | Analisa Performa | 26 |
| 4 | PENGUJIAN DAN ANALISIS | 27 |
| 4.1 | Pengujian Performa berdasar pada Penggalan Kata yang Diambil | 28 |
| 4.1.1 | Pengujian dengan Mengambil Bagian Awal Teks | 28 |
| 4.1.2 | Pengujian dengan Mengambil Bagian Akhir Teks | 29 |
| 4.1.3 | Pengujian dengan Mengambil Bagian Awal dan Akhir Teks | 30 |
| 4.2 | Pengujian Performa berdasarkan model BERT yang digunakan | 31 |
| 4.2.1 | Pengujian pada model khusus bahasa Melayu | 33 |
| 4.2.2 | Pengujian pada model dengan kemampuan multibahasa | 34 |

| | | |
|----------|--|-----------|
| 4.2.3 | Pengujian pada model <i>bert-base-indonesian-522M</i> | 35 |
| 4.2.4 | Pengujian pada model <i>bert-base-indonesian-1.5G</i> | 36 |
| 4.2.5 | Pengujian pada model <i>indobert-base-p1</i> . . . | 37 |
| 4.3 | Pengujian Performa berdasarkan pada metode <i>transformer</i> yang digunakan | 39 |
| 4.3.1 | Pengujian dengan menggunakan ROBERTA . | 40 |
| 4.3.2 | Pengujian dengan menggunakan GPT-2 . . . | 41 |
| 4.3.3 | Pengujian dengan menggunakan BERT . . . | 42 |
| 4.4 | Pengujian Performa berdasarkan pada pendekatan cara <i>training</i> | 43 |
| 4.4.1 | <i>Baseline</i> | 43 |
| 4.4.2 | Parameter <i>Freeze</i> | 44 |
| 4.4.3 | Parameter <i>Dropout</i> | 45 |
| 5 | PENUTUP | 57 |
| 5.1 | Kesimpulan | 57 |
| 5.2 | Saran | 57 |
| | DAFTAR PUSTAKA | 59 |

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | | |
|------|--|----|
| 2.1 | Contoh <i>Deep Learning</i> dengan 4 layer [1] | 8 |
| 2.2 | pendekatan dua arah BERT | 8 |
| 2.3 | Token dalam BERT [2] | 9 |
| 3.1 | bagan umum metodologi penelitian | 15 |
| 3.2 | contoh situs sumber berita terverifikasi | 18 |
| 3.3 | contoh situs sumber berita palsu | 19 |
| 3.4 | Garis besar alur program <i>web crawl</i> | 20 |
| 3.5 | Metode Preprocessing | 21 |
| 3.6 | Metode Training | 24 |
| 4.1 | Nilai <i>Loss</i> saat Pengujian dengan Mengambil Bagian Awal Teks | 31 |
| 4.2 | Nilai <i>Loss</i> saat Pengujian dengan Mengambil Bagian Akhir Teks | 33 |
| 4.3 | Nilai <i>Loss</i> saat Pengujian dengan Mengambil Bagian Tengah Teks | 35 |
| 4.4 | Nilai <i>Loss</i> saat Pengujian dengan model <i>bert-bahasa</i> | 37 |
| 4.5 | Nilai <i>Loss</i> saat Pengujian dengan model <i>bert-base</i> | 39 |
| 4.6 | Nilai <i>Loss</i> saat Pengujian dengan model <i>cahya-522M</i> | 41 |
| 4.7 | Nilai <i>Loss</i> saat Pengujian dengan model <i>cahya-1.5G</i> | 43 |
| 4.8 | Nilai <i>Loss</i> saat Pengujian dengan model <i>indobert</i> | 45 |
| 4.9 | Nilai <i>Loss</i> pada model ROBERTA | 47 |
| 4.10 | Nilai <i>Loss</i> pada model GPT-2 | 47 |
| 4.11 | Nilai <i>Loss</i> pada model BERT | 50 |

| | |
|---|----|
| 4.12 Nilai <i>loss</i> Acuan | 50 |
| 4.13 Nilai <i>loss</i> saat melakukan <i>training</i> dengan parameter <i>freeze</i> | 53 |
| 4.14 Nilai <i>loss</i> saat melakukan <i>training</i> dengan merubah <i>dropout</i> | 55 |

DAFTAR TABEL

| | | |
|------|---|----|
| 2.1 | Contoh <i>Confusion Matrix</i> | 12 |
| 3.1 | Contoh Dataset | 22 |
| 3.2 | Jumlah Dataset yang digunakan | 23 |
| 3.3 | Rincian Pembagian Dataset | 24 |
| 3.4 | Konfigurasi pada BERT | 25 |
| 4.1 | spesifikasi PC yang digunakan | 27 |
| 4.2 | Konfigurasi Parameter Untuk Pengujian berdasarkan Pemotongan Kata | 29 |
| 4.3 | Hasil Pengujian dengan Mengambil Awal Teks | 30 |
| 4.4 | Hasil Pengujian dengan Mengambil Akhir Teks | 32 |
| 4.5 | Hasil Pengujian dengan Mengambil Tengah Teks | 34 |
| 4.6 | Konfigurasi yang digunakan oleh model BERT yang digunakan | 36 |
| 4.7 | Hasil Pengujian dengan model <i>bert-bahasa</i> | 38 |
| 4.8 | Hasil Pengujian dengan model <i>bert-base</i> | 40 |
| 4.9 | Hasil Pengujian dengan model <i>cahya-522M</i> | 42 |
| 4.10 | Hasil Pengujian dengan model <i>cahya-1.5G</i> | 44 |
| 4.11 | Hasil Pengujian dengan model <i>indobert</i> | 46 |
| 4.12 | Konfigurasi yang dipakai oleh model <i>transformer</i> yang digunakan | 46 |
| 4.13 | Hasil Pengujian pada model ROBERTA | 48 |
| 4.14 | Hasil Pengujian pada model GPT-2 | 49 |
| 4.15 | Hasil Pengujian pada model BERT | 51 |

| | | |
|------|--|----|
| 4.16 | Konfigurasi yang digunakan pada saat melakukan pendekatan training | 51 |
| 4.17 | Hasil Pengujian Acuan | 52 |
| 4.18 | Hasil pengujian saat <i>training</i> dengan parameter <i>freeze</i> | 54 |
| 4.19 | Hasil pengujian saat <i>training</i> dengan merubah <i>dropout</i> | 56 |

BAB 1

PENDAHULUAN

Penelitian ini dilatar belakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar Belakang

Berita adalah laporan atau cerita faktual yang disajikan paling cepat, memiliki pemaparan masalah yang baik, serta berlaku adil kepada seluruh masalah yang disajikan [3]. Berita memiliki peran yang sangat penting dalam masyarakat karena sebagai media yang dapat digunakan untuk mengetahui peristiwa paling baru, juga dapat digunakan sebagai media untuk menambah wawasan.

Hoaks atau berita palsu adalah sebuah cara atau usaha yang berusaha untuk menipu orang sehingga mempercayai sesuai yang salah sebagai hal benar dan seringnya hal yang salah tersebut sama sekali tidak masuk akal [4]. Selain kerugian dalam hal pengetahuan, berita palsu memiliki efek yang beragam, seperti kerugian dalam bentuk reputasi, harta benda, sampai ancaman pembunuhan.

Berdasarkan data yang diperoleh dari Kementrian Komunikasi dan Informatika total jumlah berita palsu yang ditemukan pada tahun Agustus 2018 sampai dengan Maret 2020 berjumlah 5156. pada bulan Januari 2020 sampai Maret 2020, sudah terdapat 959 berita palsu yang ditemukan [5]. Masih dari sumber yang sama, pada bulan Juni 2020, hampir setiap harinya ditemukan puluhan berita palsu baru [6].

Berita hoaks juga memiliki tingkat penyebaran yang cepat seiring dengan semakin tingginya penggunaan media sosial oleh masyarakat. Berdasarkan survey yang dilakukan oleh Khan dan Idris, lebih dari 50% masyarakat Indonesia memiliki tingkat kecenderungan untuk melakukan share suatu tautan berita tanpa melakukan

validasi terlebih dahulu [7]. Survey lain yang dilakukan oleh Kunto yang melibatkan 480 responden di Kota Jawa Barat menunjukkan bahwa sekitar 30% masyarakat Jawa Barat memiliki kecenderungan menengah sampai tinggi untuk menyebarkan berita palsu [8]. Dari sampel tersebut, dapat disimpulkan bahwa Indonesia memiliki kecenderungan tinggi untuk menyebarkan berita palsu.

Neural Networks adalah salah satu cabang dalam pembelajaran mesin yang menerapkan *neurons* layaknya struktur otak manusia untuk memproses data dan menghasilkan keluaran. Salah satu metode *neural network* yang cukup baru adalah *Bi-directional Encoder Representations from Transformers* atau disingkat sebagai BERT. BERT adalah metode yang digunakan untuk mendapatkan suatu konteks dalam suatu teks yang dimasukkan.

1.2 Permasalahan

Berdasarkan data yang telah dipaparkan di latar belakang, dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Cara pendeteksi berita palsu berbahasa Indonesia yang masih menggunakan cara manual.
2. Masih belum ada model pendeteksi berita palsu berbahasa Indonesia berbasis *deep learning* terkhusus BERT dengan akurasi tinggi.

1.3 Tujuan

Tujuan dari penelitian ini adalah pengembangan pendeteksi berita palsu berbahasa Indonesia dengan menggunakan BERT yang diharapkan dapat membantu meningkatkan tingkat efisiensi dan akurasi pendeteksi berita palsu berbahasa Indonesia.

1.4 Batasan Masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Batasan - batasan masalah tersebut diantaranya adalah :

1. Data input yang digunakan adalah data yang diambil dari <https://data.mendeley.com/datasets/p3hfgr5j3m/1> yang ditambahkan data dari *web crawling* beberapa situs berita.
2. Berita yang akan digunakan dalam penelitian ini adalah berita berbahasa Indonesia.
3. Bahasa Indonesia yang digunakan hanya menggunakan bahasa baku dan tidak memperhitungkan gaya bahasa seperti satir, sarkasme, ironi, hiperbola dan sebagainya.
4. Hasil deteksi berupa label apakah suatu teks termasuk dalam berita hoaks atau tidak.
5. Hasil keluaran dari penelitian ini adalah model pembelajaran mesin yang dapat digunakan sebagai model untuk mendeteksi berita hoaks berbahasa Indonesia.

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.

2. BAB II Tinjauan Pustaka

Bab ini berisi tentang uraian secara sistematis teori - teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori - teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *Deep Learning*, *Transformer*, *Bidirectional Encode Representations from Transformers (BERT)* dan teori - teori penunjang lainnya

3. BAB III Desain dan Implementasi Sistem

Bab ini berisi tentang penjelasan - penjelasan terkait eksperimen yang akan dilakukan, langkah - langkah pengambilan dataset dan proses deteksi berita hoaks. Untuk mendukung hal

tersebut, maka ditampilkan pula *workflow* agar model yang akan dibuat dapat terlihat dan mudah dibaca untuk proses pembuatan pada pelaksanaan tugas akhir.

4. **BAB IV Pengujian dan Analisa**

Bab ini menjelaskan tentang hasil serta analisis yang didapatkan dari pengujian yang dilakukan mulai dari hasil pengujian *f1-score*, *recall*, *Confusion Matrix* serta rekomendasi penggunaan model.

5. **BAB V Penutup**

Bab ini berisi penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bagian ini.

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Berita Palsu

Berita palsu atau biasa dikenal dengan berita hoaks adalah sebuah informasi yang sesungguhnya tidak benar, tetapi dibuat seolah-olah benar adanya [4]. Di Indonesia sendiri, hoaks menjadi sebuah masalah tersendiri, hal ini karena masih banyak masyarakat yang langsung mempercayai apapun yang mereka temui di internet tanpa melakukan cek fakta terlebih dahulu.

Ada banyak sekali efek dari berita palsu ini, mulai dari hilangnya reputasi sampai nyawa yang terancam. Salah satu contoh kasus yang cukup parah adalah kerusuhan yang terjadi di Papua, dimana kerusuhan tersebut disebabkan karena adanya hoaks soal ucapan rasialis dari seorang guru SMP kepada muridnya [9].

2.2 *Machine Learning*

Machine Learning atau Pembelajaran Mesin adalah salah satu cabang dalam kecerdasan buatan dan ilmu komputer yang menggunakan data dan algoritma untuk meniru manusia dalam mempelajari sesuatu [10]. Salah satu hal yang membuat pembelajaran mesin sangat diminati adalah kemampuannya untuk menyelesaikan suatu tugas dengan sedikit intervensi dari manusia.

Sekarang ini, pembelajaran mesin adalah salah satu fokus yang cukup diminati pada bidang *data science*. Dimana dengan menggunakan pembelajaran mesin, diharapkan suatu kecerdasan buatan dapat menyelesaikan beberapa tugas yang bagi komputer cukup rumit seperti misalnya, memberikan prediksi yang akurat berdasarkan

an data, melakukan klasifikasi pada teks maupun pada gambar, melakukan pemrosesan citra guna mengenali objek di dalam citra tersebut, dan masih banyak lagi.

Untuk prosesnya sendiri, awalnya kita harus mengumpulkan data, data ini dapat kita ambil dari berbagai sumber atau bisa juga menggunakan data yang berasal dari instansi atau pribadi (data yang kita buat sendiri). Selanjutnya adalah proses *training* dimana data akan dimasukkan ke dalam model pembelajaran mesin yang sudah dipilih. Kita dapat merubah beberapa parameter dari model tersebut untuk meningkatkan akurasi dari suatu model pembelajaran mesin. Terakhir adalah melakukan proses *testing*, dimana model akan melakukan prediksi pada set data yang berbeda dari yang digunakan pada saat proses *training*. Apabila ternyata tingkat akurasi dirasa kurang memadai, dapat dilakukan proses *re-training* sampai tingkat akurasi nya dirasa cukup. Hasil akhir dari proses ini adalah sebuah model pembelajaran mesin yang dapat digunakan walaupun menggunakan data yang berbeda [11].

2.2.1 Supervised Learning

Salah satu cabang dalam bidang pembelajaran mesin. Disini data yang dijadikan masukan ke model sudah diberikan label atau struktur tertentu [12]. Berdasarkan dari data berlabel tersebut, sebuah model akan merubah parameter internalnya agar mendekati atau sesuai dengan label yang diberikan [10]. Salah satu contoh model pembelajaran mesin dengan metode pembelajaran seperti ini adalah *Linear Regression*, *Random Forest*, dan sebagainya.

2.2.2 Unsupervised Learning

Salah satu cabang dalam bidang pembelajaran mesin. Disini data yang dijadikan masukan ke model tidak diberikan label sama sekali. Nantinya model akan membuat pengelompokan (*clusters*) dan hubungan berdasarkan dari data yang diberikan [11]. Contoh model yang menggunakan metode pembelajaran ini adalah *BERT*, *GPT-2/3* dan sebagainya.

2.2.3 Reinforcement Learning

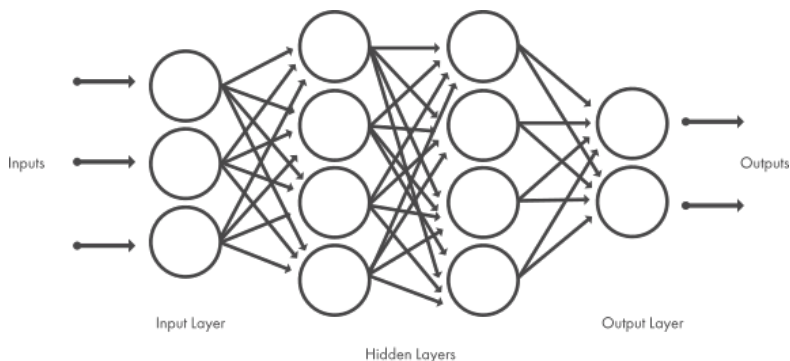
Salah satu cabang dalam bidang pembelajaran mesin. Disini model tidak diberikan data awal sama sekali, namun, model dibiarkan melakukan proses percobaan secara mandiri terus-menerus sampai tercapai hasil atau respon yang diinginkan. Apabila terdapat parameter yang menghasilkan respon positif, maka parameter tersebut disimpan dan digunakan sebagai masukan untuk iterasi *training* berikutnya [11].

2.3 Deep Learning

Mirip seperti pembelajaran mesin, *Deep Learning* juga merupakan salah satu bidang dalam bidang kecerdasan buatan. Yang membedakan antara pembelajaran mesin biasa dengan *deep learning* adalah penggunaan *layer* yang sangat banyak dibandingkan dengan pembelajaran mesin yang hanya memiliki 3 *layers*. Keuntungan dari model jenis ini adalah model ini dapat memproses masukan yang paling abstrak sekalipun, sehingga menghilangkan proses ekstraksi fitur secara manual [1]. Namun, karena *deep learning* memiliki *layers* yang sangat banyak, maka diperlukan jumlah data yang jauh lebih banyak pula, karena itu pulalah, sebuah model *deep learning* memerlukan daya komputasi yang jauh lebih besar dibandingkan dengan model pembelajaran mesin biasa [11].

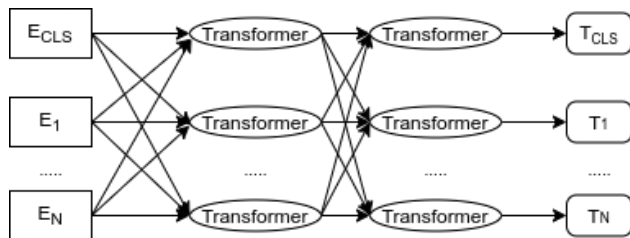
Gambar 2.1 merupakan contoh bentuk *layer* dalam suatu model *deep learning* yang menggunakan 4 *layers* didalamnya. Setiap *layer* dapat memiliki fungsi dan tanggung jawabnya masing - masing [11], seperti misal apabila kita menggunakan *deep learning* untuk mendeteksi angka plat nomor di kendaraan bermotor, bisa saja beberapa layer pertama berfungsi untuk mendeteksi letak plat nomor dalam suatu citra, kemudian beberapa layer selanjutnya berfungsi untuk mengambil bentuk dari setiap objek dalam plat nomor tersebut, beberapa layer terakhir berfungsi untuk mengenali bentuk - bentuk dari objek menjadi tulisan teks. Semakin banyak layer yang digunakan, maka semakin tinggi pula kemungkinan kita melakukan sesuatu yang lebih kompleks [11].

2.4 BERT



Gambar 2.1: Contoh *Deep Learning* dengan 4 layer [1]

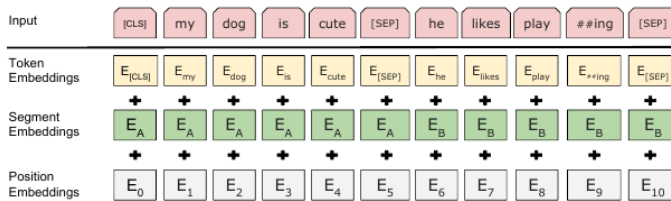
BERT merupakan suatu model yang cukup baru dan merupakan singkatan dari *Bidirectional Encode Representations from Transformers*, adalah sebuah model bahasa yang sudah dilakukan proses *pretrained* dengan menggunakan pendekatan *fine-tuning*. BERT merupakan hasil penggabungan antara *bi-directionality* dan *transformer encoder*. [2]



Gambar 2.2: pendekatan dua arah BERT

Masukan dari BERT dapat berupa teks mentah dengan sedikit melakukan *preprocessing* sebelumnya. Awalnya, BERT akan melakukan *word embedding*, yaitu suatu metode untuk mendapatkan token dari kata yang dimasukkan berdasar pada kamus. Karena proses ini pulalah, terdapat beberapa versi BERT yang khusus un-

tuk bahasa tertentu dan ada juga versi BERT untuk beberapa bahasa sekaligus. Setelah melakukan *word embedding*, BERT akan memasukkan token [CLS]. Token tersebut digunakan sebagai representasi keseluruhan teks dan dapat digunakan sebagai basis dari proses klasifikasi. Selanjutnya, BERT akan membagi kalimat menggunakan token [SEP]. Dari token tersebut akan didapat *segment embeddings* yang berfungsi untuk membedakan antara kalimat A dan kalimat B. *Embedding* yang terakhir adalah berupa *position embedding* yang digunakan untuk menunjukkan posisi kata dalam suatu kalimat. Gambar 2.3 merupakan gambaran secara garis besar bagaimana BERT memahami suatu input.



Gambar 2.3: Token dalam BERT [2]

Keluaran dari BERT adalah berupa token - token yang merepresentasikan kalimat tersebut. Nantinya, token - token tersebut dapat digunakan sebagai masukan dari algoritma lain seperti misalnya CNN, maupun menggunakan isi token [CLS] sebagai masukan algoritma klasifikasi seperti *Logistic Regression*.

Keuntungan dari penggunaan BERT adalah apabila dibandingkan dengan *word2vec* yang juga sama - sama merubah kata - kata menjadi vektor atau token adalah, BERT tidak hanya merubah kata - kata tersebut menjadi token saja, namun juga melakukan relasi dan konteks *learning* sehingga token dapat lebih menunjukkan konteks dari kalimat. Sebagai contoh, dalam kalimat "hadiah untuk ibuku sudah dikemas" dan kalimat "acara tersebut dikemas dengan rapi". Kata - kata "dikemas" disini memiliki 2 arti yang berbeda, yang pertama adalah berarti dibungkus, sedangkan yang kedua adalah ditampilkan. Apabila kita menggunakan *word2vec* biasa, kata

- kata "dikemas" akan memiliki token yang sama, sedangkan apabila menggunakan BERT, kata tersebut akan memiliki token yang berbeda. [13]

Di Indonesia sendiri sudah terdapat beberapa model BERT yang khusus untuk bahasa Indonesia seperti model yang dibuat oleh Cahya dengan menggunakan gabungan dari 522 MB wikipedia Indonesia dan 1GB surat kabar Indonesia [14], dan IndoBERT dengan 12 *layer* dan dilatih menggunakan 31,923 kata dalam bahasa Indonesia [15]. Hal ini kurang lebih sama dengan ukuran BERT-*Base* yang juga memiliki 12 *layer* dan 30,000 kata dalam bahasa Inggris [2].

2.5 Metode Analisa Performa

Terdapat beberapa metode yang bisa dilakukan untuk mengetahui apakah suatu model memiliki akurasi yang cukup. Penelitian ini menggunakan beberapa formula yang sudah ditentukan seperti *recall*, *precision*, *f1-score* dan *confusion matrix*.

2.5.1 *Recall*

Recall adalah formula yang harus digunakan ketika kita memiliki data yang tidak seimbang. Berbeda dengan akurasi yang hanya menghitung persentase model memprediksi hasil yang sesuai dengan label secara keseluruhan, *recall* akan menghitung rasio nilai yang diprediksi positif dengan total keseluruhan nilai yang positif [16]. Rumus 2.1 merupakan rumus untuk menghitung *recall*.

$$Recall = \frac{TP}{TP + FN} \quad (2.1)$$

2.5.2 *Precision*

Seringnya, kita tidak hanya melihat tingkat akurasi suatu model hanya dengan besaran *recall* maupun tingkat akurasi nya. *Precision* adalah formula untuk menghitung rasio dari prediksi TP (*True Positive*) yang benar dengan keseluruhan prediksi. Apabila prediksi yang dilakukan oleh model kita ternyata memiliki tingkat presisi yang tinggi namun memiliki tingkat *recall* yang rendah, ada ke-

mungkinan model tidak dapat melakukan prediksi pada data yang bersifat negatif [16]. Rumus 2.2 merupakan rumus untuk menghitung nilai dari *precision* suatu model.

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

Baik *recall* maupun *precision* merupakan nilai yang cukup penting terutama pada data yang tidak seimbang. Terdapat 3 kondisi yang umum terjadi pada saat membandingkan antara *precision* dengan *recall*.

- *Recall* tinggi, *Precision* rendah

Sebagian besar data positif dapat diprediksi dengan benar (*False Negative* yang Rendah), namun hanya ada sebagian kecil data negatif yang diprediksi dengan benar (*True Negative* rendah).

- *Recall* rendah, *Precision* tinggi

Hasil prediksi model memiliki banyak sekali prediksi negatif (*False Negative* tinggi), namun apabila digunakan untuk memprediksi data positif, maka hasil prediksi sebagian besarnya adalah benar (*False Positive* rendah).

- *Recall* tinggi, *Precision* tinggi

Merupakan hasil yang ideal dalam pembuatan model pembelajaran mesin. Disini didapatkan bahwa baik hasil prediksi untuk data positif maupun hasil prediksi untuk data negatif sebagian besarnya adalah benar (*True Positive* dan *True Negative* tinggi).

2.5.3 *F1-Score*

F1-Score adalah besaran yang berasal dari rata - rata harmonik dari *recall* dan *precision*. Rata - rata harmonik dipilih karena akan menghasilkan nilai rata - rata yang lebih rendah dalam kondisi tidak seimbang apabila dibandingkan dengan rata - rata aritmatik biasa. Dengan rata - rata seperti itu, suatu model dapat menjadi lebih

Tabel 2.1: Contoh *Confusion Matrix*

| | | Aktual | |
|----------|---------|---------|---------|
| | | Positif | Negatif |
| Prediksi | Positif | TP | FP |
| | Negatif | FN | TN |

rentan terhadap bias dan memudahkan pada saat pembuatan model [16]. Rumus 2.3 merupakan rumus untuk menghitung f1-score.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.3)$$

2.5.4 *Confusion Matrix*

Confusion Matrix adalah tabel kesimpulan yang berisi jumlah prediksi baik yang benar maupun yang salah dan nilai label baik yang benar maupun salah. Biasanya tabel jenis ini digunakan untuk tugas yang bersifat klasifikasi dan berfungsi untuk memvisualisasi bagaimana suatu performa model dalam suatu dataset.

Tabel 2.1 adalah contoh tabel *confussion matrix* untuk prediksi dengan 2 label. Apabila melihat pada tabel tersebut, dapat terlihat bahwa jumlah prediksi dipecah menjadi masing - masing kelas. Di-harapkan dengan dipecah menjadi beberapa kelas seperti itu, akan membuat proses pengujian lebih mudah karena akan lebih mudah melihat pada saat model memprediksi jenis data apa yang masih memiliki tingkat akurasi yang kurang bagus. Terdapat beberapa hal yang harus diketahui untuk dapat memahami sebuah tabel *confussion matrix*, yaitu :

- Positif (P)
Berisi data yang bernilai positif, baik data tersebut berasal dari hasil prediksi maupun data aktual yang didapat dari dataset.
- Negatif (N)
Berisi data yang bernilai negatif, baik data tersebut berasal dari hasil prediksi maupun data aktual yang didapat dari da-

taset.

- *True Positive* (TP)

Suatu kondisi dimana baik hasil prediksi maupun data aktual sama - sama bernilai positif. Semakin tinggi nilai dari TP, semakin akurat pulalah model yang sudah dibuat.

- *False Positive* (FP)

Suatu kondisi dimana hasil prediksi adalah positif, namun pada data aktual bernilai negatif. Biasanya, semakin tinggi nilai dari FP ini, maka model semakin memiliki kecenderungan untuk mengeluarkan nilai positif dibanding negatif atau terjadinya bias pada model.

- *True Negative* (TN)

Suatu kondisi dimana hasil prediksi dan data aktual bernilai negatif. Semakin tinggi nilai FN berarti semakin akurat model yang sudah dibuat.

- *False Negative* (FN)

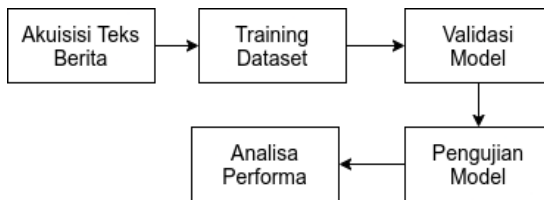
Suatu kondisi dimana hasil prediksi adalah negatif, namun pada data aktual bernilai positif. Biasanya, semakin tinggi nilai dari TN ini, maka model semakin memiliki kecenderungan untuk mengeluarkan nilai negatif dibanding positif.

[Halaman ini sengaja dikosongkan]

BAB 3

DESAIN DAN IMPLEMENTASI

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Sedangkan untuk bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.



Gambar 3.1: bagan umum metodologi penelitian

Gambar 3.1 merupakan bagan yang menunjukkan secara garis besar bagaimana proses implementasi model pada penelitian ini. Proses *training* akan dilakukan beberapa kali dengan merubah parameter dari model untuk meningkatkan tingkat akurasi pada model akhir.

3.1 Desain Sistem

Tugas akhir ini adalah penelitian dalam bidang pemrosesan teks yang bertujuan untuk mendeteksi suatu berita hoaks berbahasa Indonesia secara otomatis berbasis *Deep Learning*. Dalam proses *training*-nya model ini akan menggunakan data yang dikumpulkan dari situs - situs berita yang sudah terverifikasi dan situs berita yang memang berisi berita palsu yang sudah ditemukan.

3.2 Alur Kerja

Terdapat beberapa langkah dalam melakukan implementasi dalam penelitian ini. Tahapan - tahapan ini sesuai berdasarkan dengan metodologi penelitian, yaitu :

1. Akuisisi Data
2. *Pre-Processing*
3. Proses *Training*
4. Proses Pengujian
5. Analisa Performa

Berdasarkan bagan pada gambar 3.1, proses yang pertama kali dilakukan adalah proses akuisisi data. Data yang diambil adalah data berupa berita yang berasal dari situs yang sudah terverifikasi seperti *liputan6.com*, *detik.com*, dan sebagainya. Selain itu, kami juga mengambil data berupa berita palsu yang sudah dipastikan sebagai palsu yang berasal dari situs seperti *turnbackhoax.com*. Tujuan dari proses akuisisi data ini adalah karena hampir tidak ada set data yang cukup banyak yang dapat digunakan sebagai data untuk proses klasifikasi berita palsu. Pada proses akuisisi ini juga dilakukan proses memberi label kepada kumpulan data dengan melihat tautan sumber berita.

Setelah set data berhasil didapatkan, proses selanjutnya adalah *training* dimana data akan dimasukkan ke dalam model yang menggunakan metode berbasis *Transformer* yaitu *Bidirectional Encode Representations from Transformers*(BERT).

3.3 Akuisisi Data

Pada tahap akuisisi data, data diambil dari situs - situs berita berbahasa Indonesia yang beredar di internet. Data - data tersebut akan diambil menggunakan metode *webcrawling* sehingga diharapkan, apabila suatu saat diperlukan data dengan jumlah yang lebih banyak dapat dengan mudah memanggil program *webcrawling* yang sudah dibuat. Untuk saat ini, keseluruhan kode program dapat diakses dan diunduh pada tautan <https://github.com/chillytaka/berita-crawler>.

3.3.1 Sumber Akuisisi Data

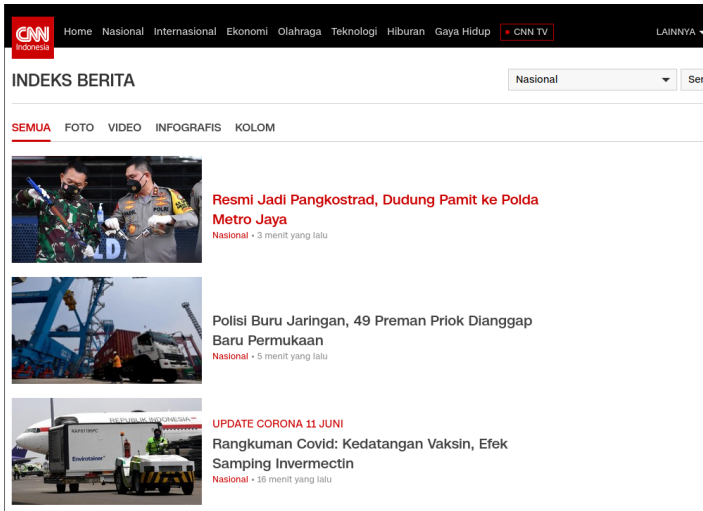
Sebagai dasar awal dari dataset, kami menggunakan dataset yang bersumber dari situs <https://data.mendeley.com/datasets/p3hfgr5j3m/1>. Namun, karena jumlah data dari situs tersebut dinilai kurang memadai untuk penelitian ini (data dari sumber tersebut hanya berjumlah 600 data), kami memutuskan untuk menambah dataset lagi menggunakan teknologi *webcrawling*.

Sumber data yang pertama diambil berasal dari beberapa situs berita yang sudah terverifikasi seperti *liputan6.com*, *detik.com*, *kompas.com*, *cnnindonesia.com*. Sumber - sumber berita terverifikasi ini dipilih untuk menghilangkan proses pemberian label setelah dataset terkumpul. Diharapkan dengan mengambil teks berita dari situs yang sudah terpercaya dapat membuat model mengetahui bagaimana teks suatu berita yang berasal dari sumber terverifikasi. Selain itu, untuk lebih mengerucutkan lagi, kami hanya mengambil berita yang membahas isu nasional dan tidak mengambil berita yang membahas olahraga, opini, maupun jenis tipe berita lainnya. Gambar 3.2 menunjukkan salah satu situs berita yang digunakan dalam pengambilan dataset penelitian ini.

Sedangkan sumber kedua yang digunakan untuk proses pengambilan data pada penelitian ini adalah situs *turnbackhoax.id*. Situs tersebut adalah situs khusus yang mengumpulkan berita - berita yang sudah dipastikan palsu yang berasal dari berbagai sumber. Selain itu, situs ini juga mengambil data dari hasil unggahan masyarakat Indonesia sendiri melalui sosial media dan grup resmi *turnbackhoax.id*. Keuntungan dari metode seperti ini adalah sebuah unggahan akan dicek berkali - kali oleh anggota grup sehingga mengurangi kemungkinan terdapat berita yang salah klasifikasi. Alasan lain situs ini dipilih sebagai sumber adalah adanya format yang kurang lebih sama antara setiap unggahan sehingga akan memudahkan dalam proses pengambilan teks berita.

3.3.2 Proses Akuisisi Data

Setelah menentukan situs - situs untuk digunakan sebagai sumber berita, langkah selanjutnya adalah memulai proses akuisisi data. Disini, hal yang paling pertama kali dilakukan adalah mengambil

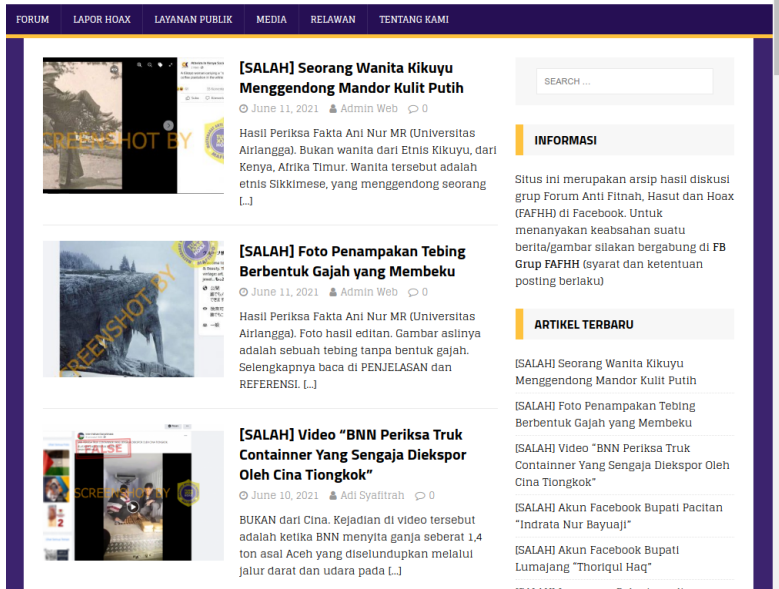


Gambar 3.2: contoh situs sumber berita terverifikasi

kode sumber dari situs - situs tersebut. Hal ini dilakukan guna mempermudah saat melakukan penyaringan untuk mendapatkan teks berita. Gambar 3.4 adalah gambaran garis besar yang kami lakukan dalam program *web crawl* kami. Dimulai dengan memasukkan kode HTML mentah, kemudian merubah kode mentah tersebut menjadi objek yang lebih mudah untuk dilakukan pemrosesan dalam python, mengambil teks berita dan melakukan pembersihan terhadap teks tersebut, terakhir menghasilkan keluaran berupa file .CSV dengan format yang sesuai.

Library yang kami gunakan untuk melakukan *crawling* adalah *BeautifulSoup*, sebuah *library* yang akan secara otomatis merubah dari suatu teks HTML menjadi objek *soup* yang lebih mudah untuk dilakukan pemrosesan di dalam python.

Yang pertama kali harus kami lakukan adalah menentukan *tag* atau *class* HTML yang akan digunakan sebagai masukan pada program guna melakukan penyaringan terlebih dahulu. Apabila merujuk pada listing 3.1 *class* yang berisi teks seluruh berita adalah

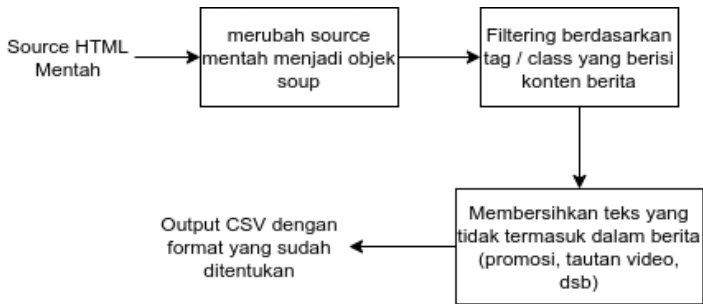


Gambar 3.3: contoh situs sumber berita palsu

`detail_bodytext` sehingga kami melakukan penyaringan dengan memasukkan *class* tersebut ke dalam parameter.

Namun, walaupun sudah melakukan penyaringan, masih terdapat beberapa teks yang tidak diperlukan. Biasanya, teks - teks tersebut masuk ke dalam kategori seperti catatan dari penulis, iklan, dan tautan untuk menuju ke berita yang masih berhubungan. Sehingga, setelah melakukan penyaringan, masih diperlukan lagi pembersihan isi berita dari teks - teks yang tidak diperlukan.

Terakhir, adalah melakukan keluaran berupa *file* .CSV. Alasan menggunakan format CSV sebagai keluaran adalah karena format tersebut bersifat 'terbuka' dan dapat dibuka oleh berbagai *software*



Gambar 3.4: Garis besar alur program *web crawl*.

Listing 3.1: Penggalan Kode Sumber HTML detik.com.

```

1
2 ...
3 <div class="detail__body itp_bodycontent_wrapper">
4 <div class="detail__body-text itp_bodycontent">
5
6 <strong>Jakarta</strong> - Koalisi <a href="https://
7 detik.com/tag/jokowi" target="_blank">Jokowi</a>
8 sedang menyusun visi-misi jagoannya. Setelah
9 menerima masukan dari <a href="https://detik.com/
10 tag/muhammadiyah" target="_blank"> Muhammadiyah</a>,
11 ...
12 Dan kita pun membuka diri untuk menerima
13 masukan untuk penyempurnaan," imbuhnya.<br><br><!--
14 s:parallaxindetail--><div class="clearfix"></div><style↵
15 ...
  
```

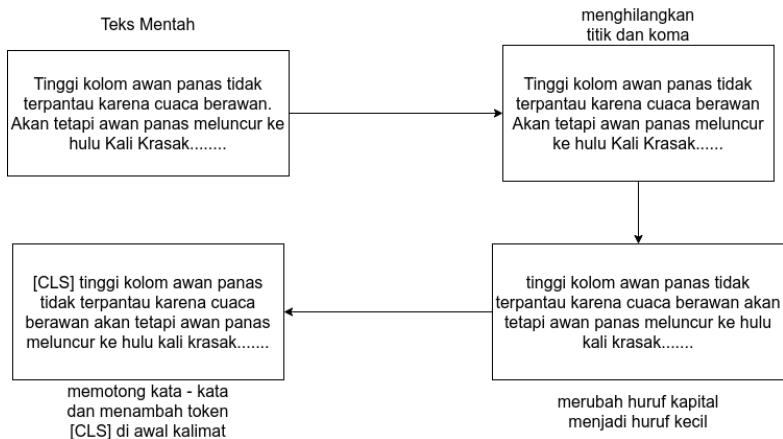
spreadsheet pada umumnya, selain itu akan lebih mudah memproses data dalam bentuk .CSV di python dibandingkan dengan format lainnya.

Untuk memudahkan penggunaan perangkat lunak *webcrawler* yang kami buat, kami menggunakan berkas dengan format .json untuk mengatur sumber, banyak berita dan filter tanggal yang nantinya akan dibaca oleh program dan mengambil berita dengan parameter tersebut. Tabel 3.1 merupakan contoh hasil keluaran dari

program *webcrawling* yang digunakan.

Setelah melakukan penggabungan antara data yang berasal dari <https://data.mendeley.com/datasets/p3hfgr5j3m/1> dengan dataset hasil dari proses *webcrawling*, maka didapatkan total data sebesar 1621 data dengan pola penyebaran data yang kurang lebih seimbang sehingga mengurangi kemungkinan terjadinya bias pada saat proses melatih model. Tabel 3.2 menunjukkan secara lebih detail berapa banyak distribusi data yang berada di dalam dataset yang digunakan dalam penelitian ini.

3.4 Preprocessing



Gambar 3.5: Metode Preprocessing

Pada proses ini, data akan disiapkan terlebih dahulu agar dapat diproses oleh BERT. Proses penyiapan data meliputi menghilangkan titik dan koma, dan merubah huruf kapital yang ada menjadi huruf kecil seluruhnya. Dan karena BERT memiliki maksimal kata - kata yang dapat diproses dalam sekali waktu sejumlah 512 kata atau token, maka harus dilakukan penyingkatan teks, dapat dengan cara melakukan pengambilan 512 karakter pertama, terakhir maupun gabungan dari kedua bentuk. Chi Sun et al. menemukan bahwa

Tabel 3.1: Contoh Dataset

| berita | <i>tagging</i> |
|---|-----------------------|
| Wakil Gubernur DKI Jakarta Sandiaga Uno menargetkan pengerjaan tahap awal Stadion BMW dilakukan pada Oktober. Stadion ini diperuntukkan bagi klub Persija.... | Valid |
| "Komisi II bersama KPU dan Bawaslu masih membahas ketentuan wajib cuti bagi petahana presiden yang maju Pilpres 2019. Mekanisme pengambilan..... | Valid |
| Jaksa penuntut Umum (JPU) pada Komisi Pemberantasan Korupsi (KPK) mencecar Pejabat Pembuat Komitmen (PPK) reguler pada Direktorat Perlindungan Sosial Korban Bencana Sosial Kemensos Victorious Saut Hamonangan Siahaan soal... | Valid |
| "Halo Kak! Aku Winda Dari Team Giveaway BAIM WONG Anda Memenangkan Hadiah Uang 100Jt dari kami info klik: https://wa.me/+6285796306857 " | Hoax |
| "Apa yang terjadi dengan hewan dalam penelitian? Teknologi ini telah dicoba pada hewan, dan pada hewan penelitian yang dilakukan, semua hewan mati , tidak langsung dari suntikan... | Hoax |
| "Kadrun istilah dr PKI alias KOMUNIS ditunjukkan buat islam. Kl mau jd komunis pake aja istilah kadrun buat umat islam. Auto lsg Komunis" | Hoax |

Tabel 3.2: Jumlah Dataset yang digunakan

| Label | Jumlah Data |
|--------------|--------------------|
| <i>Hoaks</i> | 885 |
| <i>Valid</i> | 736 |
| Total | 1621 |

mengambil teks dengan cara mengambil bagian tengah sebanyak 128 kata pertama dan mengambil sebanyak 382 kata pada bagian akhir menghasilkan hasil akurasi yang lebih baik dalam beberapa tugas [17]. Langkah terakhir adalah menambahkan token [CLS] di awal kalimat. Untuk lebih jelasnya, bisa melihat pada Gambar 3.5.

Selain itu, juga akan dilakukan pembagian dataset yang awalnya berjumlah 1621 akan dibagi menjadi 3 bagian dengan ketentuan 70% akan digunakan pada saat proses *training*, 10% akan digunakan untuk proses validasi, dan sisanya sebesar 20% akan digunakan pada saat pengujian.

1. *Training*

Set ini digunakan oleh algoritma BERT sebagai masukan saat melakukan proses *training* sehingga akan didapat model yang sesuai.

2. Validasi

Set ini digunakan pada saat selesai melakukan validasi model setelah melakukan *training*. Digunakan untuk menentukan apakah suatu model sudah memiliki *weight* yang sesuai ataukah masih perlu melakukan *training* lagi. Selain itu, set ini juga digunakan untuk menghindari kemungkinan *overfitting* maupun *underfitting* dalam model.

3. Pengujian

Set yang digunakan untuk melakukan pengujian akurasi model setelah proses *training* dan validasi selesai. Hasil akurasi

Tabel 3.3: Rincian Pembagian Dataset

| Bagian | Hoaks | Valid | Total Data |
|------------------|-------|-------|-------------|
| <i>Training</i> | 647 | 519 | 1166 |
| <i>Validasi</i> | 85 | 78 | 163 |
| <i>Peguajian</i> | 153 | 139 | 292 |
| Total | | | 1621 |

dari pengujian inilah yang akan digunakan sebagai hasil dari model.

Untuk lebih jelasnya, silahkan lihat tabel 3.3. Dari tabel tersebut dapat dilihat bahwa pembagian dan total dari dataset sudah sesuai.

3.5 Proses *Training*



Gambar 3.6: Metode Training

Pada tahap ini, teks yang sudah melewati proses *preprocessing* akan dilakukan proses Tokenizer. Tokenizer adalah proses untuk merubah kata - kata dalam teks menjadi token sesuai dengan *word embedding* yang sudah ada pada *pretrained* BERT. Barulah pada saat itu, BERT dapat melakukan *training* berdasarkan data dari *dataset*.

Keluaran dari BERT akan diambil isi token [CLS]-nya dan dimasukkan kedalam algoritma klasifikasi, dalam penelitian ini kami memilih untuk menggunakan metode *Linear Regression*. *Linear Regression* digunakan sebagai algoritma klasifikasi yang cukup mudah namun memiliki tingkat akurasi yang cukup. Gambar 3.6 dapat

Tabel 3.4: Konfigurasi pada BERT

| Jenis Konfigurasi | Keterangan |
|-----------------------|------------|
| <i>batch</i> | 8 |
| <i>learning rates</i> | 2e-6 |
| <i>epoch</i> | 5 |

digunakan sebagai penjas.

Pada tahap ini juga dilakukan pengaturan ukuran *batch*, *learning rate* dan juga *epoch*. *Batch* adalah banyaknya teks yang diproses untuk setiap iterasi, semakin tinggi nilai *batch* yang dikonfigurasi, maka proses *training* akan semakin cepat namun memakan memori yang lebih banyak. Berhubung algoritma BERT adalah algoritma yang cukup berat karena memiliki *layer* yang cukup banyak, dan karena terdapat keterbatasan *resource* maka dalam penelitian ini kami menggunakan *batch* dengan nilai 8.

Epoch adalah berapa banyak suatu algoritma melakukan proses *training* dan validasi sebelum dianggap final. Disini *epoch* harus diperhatikan agar jumlah *loss* yang terjadi pada saat proses *training* tidak terlalu tinggi karena merupakan ciri - ciri *underfitting* namun juga tidak terlalu rendah selama beberapa *epoch* untuk menghindari kemungkinan *overfitting*. Berhubung kami hanya menggunakan BERT untuk memproses teks yang relatif lebih mudah, kami hanya menggunakan *epoch* sebesar 10.

learning rate adalah seberapa banyak *hyperparameter* yang di rubah selama proses *training*. *hyperparameter* digunakan untuk merubah *weight* selama proses *training* berdasarkan *feedback* saat proses validasi. Disini kami menggunakan nilai yang direkomendasikan oleh pembuat model BERT yang kami gunakan, yaitu 0.000002 [15].

3.6 Proses Pengujian

Proses pengujian dalam penelitian ini dibagi menjadi 2 bagian, bagian yang pertama dilakukan pada saat model selesai melakuk-

an proses *training* namun masih memiliki iterasi *epoch* yang belum selesai. Proses ini bernama validasi. Proses ini sangat vital karena dengan validasi kita dapat mengetahui apakah model kita mengalami *overfitting* maupun *underfitting*. Salah satu ciri yang paling mudah yang menandakan kemungkinan terjadinya *overfitting* adalah ketika besaran *training loss* suatu model menjadi semakin kecil, namun besaran *validation loss*-nya malah semakin besar di setiap iterasi. Sedangkan *underfitting* terjadi ketika baik *training loss* maupun *validation loss* memiliki nilai yang terlalu besar. Selain itu pada proses validasi ini, data yang digunakan adalah data yang sama sekali baru dan tidak digunakan selama proses *training* guna menghindari kemungkinan terjadinya bias yang biasa terjadi apabila suatu model diuji pada data yang sama yang digunakan pada saat proses *training*. Berdasarkan data pada proses validasi, algoritma *optimizer* akan memutuskan untuk merubah *weight* dalam *hidden node* BERT sehingga semakin mendekati titik akurasi tertingginya.

Bagian selanjutnya dari proses pengujian adalah melakukan *test*. Sama seperti pada waktu proses validasi, dataset yang digunakan juga sama sekali berbeda dengan data yang digunakan pada waktu *training* maupun pada waktu *validasi*. Dari proses ini, dapat diambil kesimpulan apakah suatu model tersebut dapat dilakukan perbaikan lagi dengan cara *re-training* dan merubah beberapa parameter maupun konfigurasi yang sudah diatur pada saat proses *training*, ataukah model tersebut dirasa sudah cukup baik dan akan melanjutkan ke proses berikutnya.

3.7 Analisa Performa

Setelah melakukan proses pengujian, langkah berikutnya adalah melakukan analisa performa pada model yang sudah dibuat. Hal ini untuk mengetahui bagaimana kira - kira performa model pada saat sudah diimplementasi. Untuk analisa performa ini akan digunakan beberapa metode seperti *confusion matrix* agar mendapatkan pengelompokan berdasarkan data menjadi *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Selain itu, penelitian ini juga akan menggunakan rumus - rumus seperti *Recall*, *Precision*, *F1-score*.

BAB 4

PENGUJIAN DAN ANALISIS

Pada penelitian ini dipaparkan hasil pengujian serta analisis yang dilakukan sesuai dengan desain sistem yang sudah dirancang pada bab sebelumnya. Dataset yang digunakan berasal dari **data.mendeley.com** ditambah dengan dataset yang berasal dari proses *webcrawling* sendiri. Pengujian dilakukan dengan beberapa bagian sebagai berikut :

1. Pengujian Performa berdasar pada Penggalan Kata yang Diambil
2. Pengujian Performa berdasarkan model BERT yang digunakan
3. Pengujian Performa berdasarkan pada metode *transformer* yang digunakan.
4. Pengujian Performa berdasarkan Pendekatan Cara *Training*.

Pada pengujian, masing - masing model menggunakan Google Collab dengan spesifikasi *hardware* seperti yang dilampirkan pada tabel 4.1

Tabel 4.1: spesifikasi PC yang digunakan

| | |
|-----------------------|---|
| Prosesor | 2 v-core Intel(R) Xeon(R) CPU @ 2.20GHz |
| RAM | Virtual Memory : 12GB |
| <i>Storage</i> | SSD : 69GB |
| GPU | Nvidia Tesla T4 16GB |
| | Nvidia K80 12GB |
| Sistem Operasi | Ubuntu 18.04.5 LTS (Bionic Beaver) 64-bit |

4.1 Pengujian Performa berdasar pada Penggalan

Kata yang Diambil

Untuk saat ini, BERT hanya dapat memproses sebanyak 512 token sekaligus. Sehingga, untuk melakukan pemrosesan pada data dengan teks yang panjang, diperlukan pemotongan teks agar panjang teks menjadi sesuai.

Pengujian performa berdasar pada penggalan kata yang diambil ini bertujuan untuk mengetahui tingkat akurasi model BERT pada teks dengan cara pemotongan yang berbeda. Perbedaan ini dilakukan berdasarkan pada adanya berita yang menuliskan kesimpulan di awal, atau bisa juga menuliskan kesimpulan di akhir. Alternatif lain adalah mengambil sebagian teks di bagian awal dan mengambil sisanya di bagian akhir. Maka dari itu, dalam pengujian performa ini dilakukan dengan membagi teks pada beberapa cara memenggal kata, antara lain :

1. Mengambil bagian awal teks.
2. Mengambil bagian akhir teks.
3. Mengambil 129 token dari bagian awal teks dan 383 token dari bagian akhir teks.

Dari total data yang berjumlah 1621 data, akan diambil 18% nya sebagai dataset pengujian, sehingga berjumlah 292 dataset sebagai pengujian. Parameter pada pengujian untuk *training* di atur agar sama untuk setiap pengujian, *epoch* sebesar 7, *learning rate* sebesar $2e-5$, dan *epsilon* sebesar $1e-8$, hal yang sama juga dilakukan pada model, pengujian ini menggunakan model BERT yang telah di-*pre-trained* oleh Indobert. Untuk lebih jelasnya, silahkan lihat Tabel 4.2 yang berisi rincian parameter dan model yang digunakan untuk proses *training*.

Keluaran dari model akan dibandingkan dengan label pada dataset, yang kemudian akan dihitung untuk menghasilkan *confusion matrix*, *recall*, *precision*, *accuracy* dan *f1-score* sesuai dengan rumus yang telah dijelaskan pada bab tinjauan pustaka.

4.1.1 Pengujian dengan Mengambil Bagian Awal Teks

Terdapat beberapa ciri - ciri yang terdapat pada kebanyakan teks berita berbahasa Indonesia, salah satu dari ciri - ciri terse-

Tabel 4.2: Konfigurasi Parameter Untuk Pengujian berdasarkan Pemotongan Kata

| | |
|-----------------------|--------------------------------|
| <i>epoch</i> | 3 |
| <i>learning rates</i> | 2e-5 |
| <i>epsilon</i> | 1e-4 |
| model | indobenchmark/indobert-base-p1 |

but adalah menuliskan ringkasan berita pada paragraf awal kalimat. Format seperti ini biasanya cukup sering ditemui terutama pada berita yang memanfaatkan fitur halaman pada teks beritanya. Karenanya, pada jenis - jenis berita seperti ini, orang hanya perlu melihat beberapa kalimat awal untuk mengetahui apakah bahwa berita tersebut valid dan dapat dipercaya.

Seperti bisa dilihat pada tabel 4.3, dengan memotong teks berita pada bagian awal, didapatkan tingkat akurasi sebesar 89% dengan nilai *recall*, *precision*, *f1-score* kurang lebih sama. Selain itu, dapat dilihat juga pada kolom *confusion matrix*, model yang sudah dibuat memiliki nilai FP dan FN yang sudah cukup sedikit. Namun, apabila melihat pada gambar 4.1, terlihat bahwa model agak sedikit *overfit*, sehingga terdapat kemungkinan akan memiliki tingkat akurasi yang lebih rendah pada saat implementasi.

4.1.2 Pengujian dengan Mengambil Bagian Akhir Teks

Mirip seperti pengujian dengan mengambil bagian awal teks, terdapat ciri - ciri lain yang biasanya terdapat pada teks berita berbahasa Indonesia adalah adanya kesimpulan pada bagian akhir teks berita. Sehingga, setelah isi berita yang biasanya dibahas cukup dalam, pembaca dapat mengetahui bagaimana dan apa hubungan setiap informasi yang disajikan dengan peristiwa yang sedang dibahas dalam berita.

Dapat dilihat pada tabel 4.4, model dengan cara memotong teks seperti ini berhasil mendapatkan tingkat akurasi sebesar 86%.

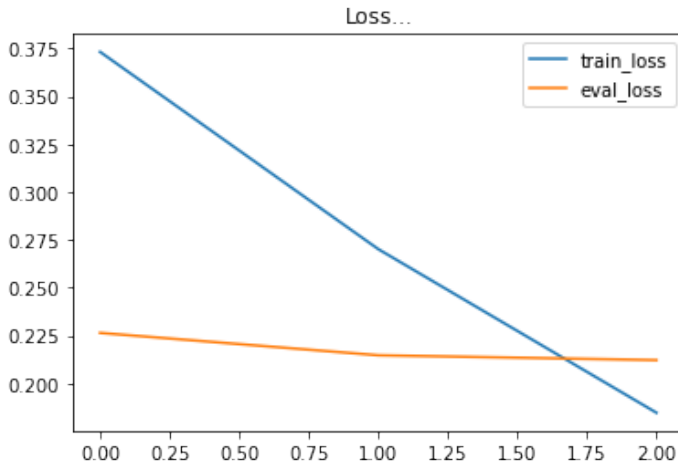
Tabel 4.3: Hasil Pengujian dengan Mengambil Awal Teks

| Hasil Model | | Nilai |
|-------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 122 |
| | FP | 17 |
| | TN | 137 |
| | FN | 16 |
| <i>Recall</i> | Hoax | 89% |
| | Valid | 88% |
| <i>Precision</i> | Hoax | 90% |
| | Valid | 88% |
| <i>F1-Score</i> | Hoax | 89% |
| | Valid | 88% |
| <i>Accuracy</i> | | 89% |

Namun, apabila melihat nilai *recall* dan *precision*-nya, ada indikasi bias dimana model lebih memiliki kecenderungan untuk mengeluarkan hasil hoaks bahkan pada berita valid sekalipun. Sedangkan, apabila melihat pada gambar 4.2, dapat dilihat bahwa baik nilai *train loss* dan *validation loss* sudah berada di titik optimal dan apabila dilakukan *training* tambahan, akan membuat model *overfit*.

4.1.3 Pengujian dengan Mengambil Bagian Awal dan Akhir Teks

Pengujian ini berdasarkan pada penelitian Chi Sun et al. yang menemukan bahwa dengan strategi pengambilan teks yang dibagi dua seperti ini akan dapat memberikan nilai akurasi yang lebih baik apabila dibandingkan dengan mengambil hanya di bagian awal maupun di bagian akhir saja [17]. Alasan dari penyebab lebih tingginya akurasi adalah karena dengan mengambil sebagian di awal maka sebagian dari ringkasan berita akan didapatkan, sedangkan



Gambar 4.1: Nilai *Loss* saat Pengujian dengan Mengambil Bagian Awal Teks

mengambil sebagian di akhir adalah agar kesimpulan berita juga masuk ke dalam proses *training*. Namun, pengujian tersebut dilakukan pada dataset teks berita berbahasa Inggris sehingga masih harus dilakukan pengujian lagi pada dataset teks berita berbahasa Indonesia.

4.2 Pengujian Performa berdasarkan model BERT yang digunakan

Terdapat banyak sekali model BERT yang sudah dibuat oleh berbagai orang di internet, ada model yang memiliki kemampuan *multilanguage* sehingga bisa digunakan di berbagai bahasa sekaligus, namun kebanyakan model yang beredar adalah model yang menggunakan bahasa yang spesifik. Hal ini karena waktu *pre-training* yang lebih singkat karena dataset yang lebih sedikit apabila dibandingkan model dengan kemampuan *multilanguage* dan karena waktu *pre-training* lebih sedikit, maka sumber daya yang digunak-

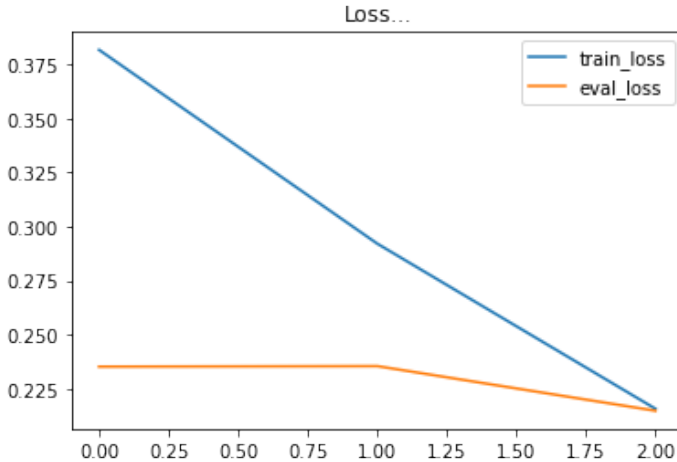
Tabel 4.4: Hasil Pengujian dengan Mengambil Akhir Teks

| Hasil Model | | Nilai |
|-------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 121 |
| | FP | 18 |
| | TN | 130 |
| | FN | 23 |
| <i>Recall</i> | Hoax | 88% |
| | Valid | 84% |
| <i>Precision</i> | Hoax | 85% |
| | Valid | 87% |
| <i>F1-Score</i> | Hoax | 86% |
| | Valid | 86% |
| <i>Accuracy</i> | | 86% |

an juga menjadi lebih sedikit. Selain itu, dan hal ini adalah yang paling penting, hasil akurasi dari model yang hanya menggunakan 1 bahasa memiliki tingkat akurasi yang lebih tinggi apabila dibandingkan dengan model dengan banyak bahasa sekaligus. Maka dari itu, kami menguji pada beberapa model BERT sekaligus dengan rincian nama model sebagai berikut :

1. Model khusus bahasa Melayu, diwakili oleh : *bert-base-bahasa-standard-case*
2. Model dengan multibahasa, diwakili oleh : *bert-multilingual-uncased*
3. Model khusus bahasa Indonesia, *indobert-base-p1*
4. Model khusus bahasa Indonesia, *bert-base-indonesian-522M*
5. Model khusus bahasa Indonesia, *bert-base-indonesian-1.5G*

Untuk melakukan *training*, sebelumnya kami mengatur konfigurasi yang akan digunakan oleh model BERT yang sudah disiapkan



Gambar 4.2: Nilai *Loss* saat Pengujian dengan Mengambil Bagian Akhir Teks

an. Terdapat beberapa perbedaan pada konfigurasi seperti jumlah *epoch* dan jumlah *dropout*. Hal ini agar membuat model dapat memiliki konfigurasi yang lebih optimal dan tidak mengalami *overfit* maupun *underfit*.

4.2.1 Pengujian pada model khusus bahasa Melayu

Model yang diberi nama *bert-base-bahasa-standard-case* adalah model yang di-*pretrained* menggunakan bahasa Melayu saja dengan mengambil berbagai sumber dataset seperti media sosial, Wikipedia, sampai Wattpad [18]. Model ini dilatih oleh huzeinzol05 dan menurut pembuat model, model ini seharusnya dapat digunakan baik untuk tugas - tugas berbahasa Melayu maupun tugas - tugas berbahasa Indonesia dikarenakan kesamaan pada tata bahasa dan arti suatu kata. Dalam buku ini model ini disebutkan dengan *bert-bahasa*

Pada gambar 4.4 terlihat bahwa model memiliki nilai *valida-*

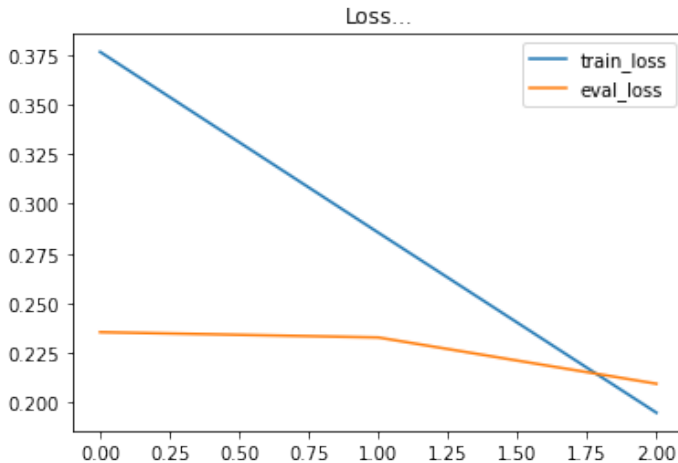
Tabel 4.5: Hasil Pengujian dengan Mengambil Tengah Teks

| Hasil Model | | Nilai |
|-------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 120 |
| | FP | 19 |
| | TN | 134 |
| | FN | 19 |
| <i>Recall</i> | Hoax | 88% |
| | Valid | 86% |
| <i>Precision</i> | Hoax | 88% |
| | Valid | 86% |
| <i>F1-Score</i> | Hoax | 88% |
| | Valid | 86% |
| <i>Accuracy</i> | | 87% |

tion loss dan *training loss* yang cukup bagus. Namun, apabila diuji dengan menambahkan epoch, hasil pengujian tersebut menghasilkan model yang *overfitting* sehingga tidak kami gunakan sebagai nilai pembandingan. Berdasarkan tabel 4.7, terlihat bahwa *recall* lebih tinggi dibanding *precision* saat mendeteksi berita hoaks, sehingga dapat disimpulkan bahwa model mengalami kesulitan saat digunakan untuk mendeteksi berita yang valid.

4.2.2 Pengujian pada model dengan kemampuan multibahasa

Model BERT yang digunakan pada pengujian ini adalah model dasar yang dibuat langsung oleh tim di Google. Selain itu, model ini jugalah yang merupakan hasil dari penelitian yang sudah dibuat oleh devlin et al. dalam penelitian awal mengenai BERT. Model ini dilatih pada seluruh bahasa yang ada pada Wikipedia sehingga untuk saat ini, model tersebut dapat mendukung sebanyak 104 bahasa



Gambar 4.3: Nilai *Loss* saat Pengujian dengan Mengambil Bagian Tengah Teks

sekaligus termasuk didalamnya adalah bahasa Indonesia. Oleh tim dari Google, model ini diberi nama *bert-base-multilingual-uncased* dan dalam buku ini disingkat sebagai *bert-base*.

Mirip seperti pada model *bert-bahasa*, model *bert-base* ini adalah model dengan grafik yang cukup bagus, terlihat pada gambar 4.5, nilai dari *training loss* memiliki *trend* yang kurang lebih sama apabila dibandingkan dengan nilai *validation loss*. Namun, apabila melihat dari tabel 4.8 nilai dari *recall* dan *precision*-nya menunjukkan model kesulitan dalam mendeteksi berita valid.

4.2.3 Pengujian pada model *bert-base-indonesian-522M*

Merupakan model yang dibuat oleh Cahya Wirawan dengan memanfaatkan dataset yang berasal dari Wikipedia berbahasa Indonesia sebesar 522M. Model ini merupakan model dengan ukuran dataset yang paling kecil apabila dibandingkan dengan model - model yang digunakan dalam pengujian ini. Dalam buku ini model ini

Tabel 4.6: Konfigurasi yang digunakan oleh model BERT yang digunakan

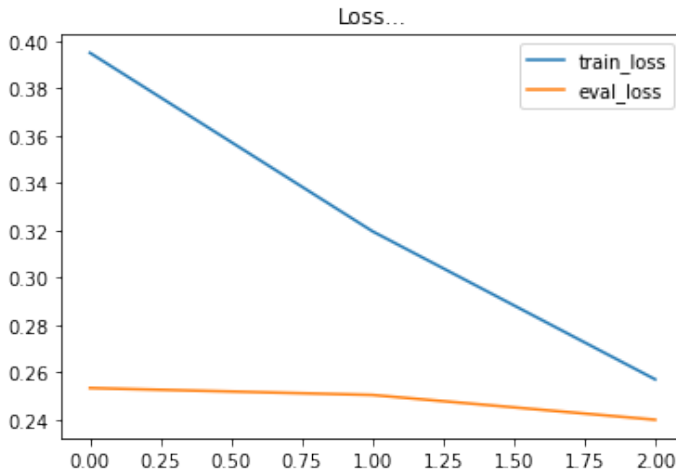
| Model | epoch | dropout | learning rates |
|--------------------------------|-------|---------|----------------|
| bert-base-bahasa-standard-case | 4 | 0.2 | 2e-5 |
| bert-base-multilingual-uncased | 4 | 0.2 | 2e-5 |
| indobert-base-p1 | 3 | 0.1 | 2e-5 |
| bert-base-indonesian-522M | 3 | 0.1 | 2e-5 |
| bert-base-indonesian-1.5G | 3 | 0.2 | 2e-5 |

disebut sebagai *cahya-522M*

Dari tabel 4.9 dapat dilihat bahwa model memiliki tingkat akurasi yang paling sedikit apabila dibandingkan dengan model - model yang lain. Hal ini karena BERT sebagai model yang cukup kompleks sangat bergantung pada data yang digunakan saat *pre-trained*. Namun, model ini juga memiliki waktu *training* yang paling sedikit dibanding model - model lain yang digunakan dalam pengujian ini. Selain itu, berdasarkan pada nilai *recall* dan *precision*-nya, dapat diambil kesimpulan bahwa model kurang bagus dalam mendeteksi berita valid. Gambar 4.6 menunjukkan nilai *validation loss* yang menunjukkan tanda - tanda akan terjadi *overfit*.

4.2.4 Pengujian pada model *bert-base-indonesian-1.5G*

Sama seperti model sebelumnya, model ini juga dibuat oleh Cahya Wirawan. Yang menjadi pembeda antara model ini dengan model sebelumnya adalah penambahan dataset menjadi tidak hanya berasal dari Wikipedia bahasa Indonesia saja, namun juga menambahkan dari situs - situs berita berbahasa Indonesia sebesar 1G, sehingga membuat total dataset yang digunakan pada waktu *pre-trained* adalah 1.5G. Pada laporan ini, model ini disebut dengan *cahya-1.5G*.



Gambar 4.4: Nilai *Loss* saat Pengujian dengan model *bert-bahasa*

Hampir mirip seperti pada model sebelumnya, pada tabel 4.10 terlihat bahwa model ini agak kesulitan untuk mendeteksi berita valid, sehingga nilai FP-nya juga cukup tinggi. Namun, apabila dibandingkan dengan model sebelumnya yang di-*pretrained* hanya dengan 522M data, terlihat peningkatan tingkat akurasi. Berdasarkan pada gambar 4.7, nilai *validation loss* yang lebih rendah apabila dibandingkan dengan model sebelumnya. Walaupun juga terlihat kemungkinan adanya *overfit*.

4.2.5 Pengujian pada model *indobert-base-p1*

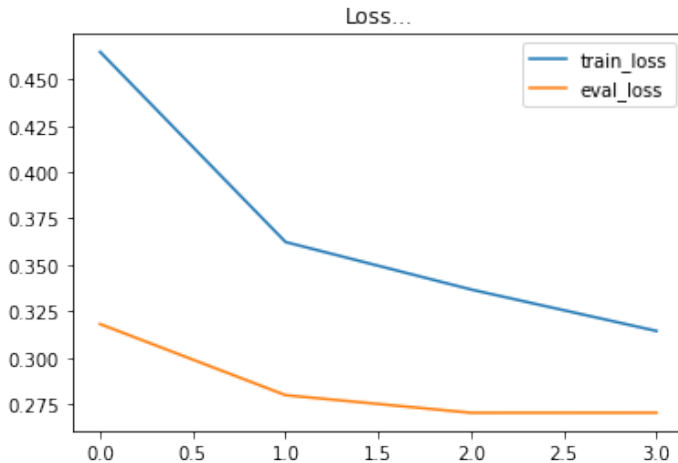
Model BERT ini adalah salah satu dari 3 model khusus berbahasa Indonesia yang diuji dalam penelitian ini. Dibuat oleh tim indobenchmark sebagai bagian dari penelitian untuk uji *benchmarking* pada *Natural Language Understanding* (NLU) berbahasa Indonesia. Diantara model - model BERT khusus bahasa Indonesia yang lain, model ini adalah model yang dilatih pada dataset terbanyak, yaitu sebesar 23 GB lebih data yang berasal dari banyak sumber seperti Wikipedia, sosial media, OpenSubtitle, dan banyak lagi [15].

Tabel 4.7: Hasil Pengujian dengan model *bert-bahasa*

| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 124 |
| | FP | 28 |
| | TN | 125 |
| | FN | 15 |
| <i>Recall</i> | Hoax | 89% |
| | Valid | 82% |
| <i>Precision</i> | Hoax | 82% |
| | Valid | 89% |
| <i>F1-Score</i> | Hoax | 85% |
| | Valid | 85% |
| <i>Accuracy</i> | | 85% |
| <i>Avg. Training Time (mnt)</i> | | 03:43 |

Sebagai model yang dilatih pada dataset khusus berbahasa Indonesia yang paling banyak, model ini memiliki tingkat akurasi yang paling tinggi apabila dibanding model - model yang lain. Gambar 4.8 menunjukkan grafik *loss* dari model ini, terlihat pada grafik tersebut, nilai *validation loss* nya sudah lebih tinggi dari *training loss*-nya sehingga model sudah sedikit *overfit*. Berdasarkan pada tabel 4.11, dapat dilihat bahwa model ini memiliki tingkat akurasi yang paling tinggi apabila dibandingkan dengan model - model yang lain.

4.3 Pengujian Performa berdasarkan pada metode



Gambar 4.5: Nilai *Loss* saat Pengujian dengan model *bert-base*

***transformer* yang digunakan**

BERT adalah model yang berdasarkan pada metode pendekatan *transformer*, yaitu pendekatan yang hanya menggunakan *attention head* saja dibandingkan dengan pengembangan sebelumnya yang menggunakan LSTM + *attention head*. Terdapat beberapa metode lain yang merupakan turunan dari metode *transformer* ini, seperti misalnya BERT yang digunakan dalam penelitian ini, kemudian terdapat GPT-2, XLMNet, Roberta dan masih banyak lagi. Maka dari itu, kami menguji bagaimana performa model BERT yang digunakan dengan beberapa metode pendekatan transformer lain tersebut. Beberapa model yang kami gunakan dalam penelitian ini adalah :

1. Roberta
2. BERT
3. GPT-2

Sebagai penyamaan dan memastikan bahwa yang diuji disini adalah benar - benar performa dari setiap cara pendekatan mo-

Tabel 4.8: Hasil Pengujian dengan model *bert-base*

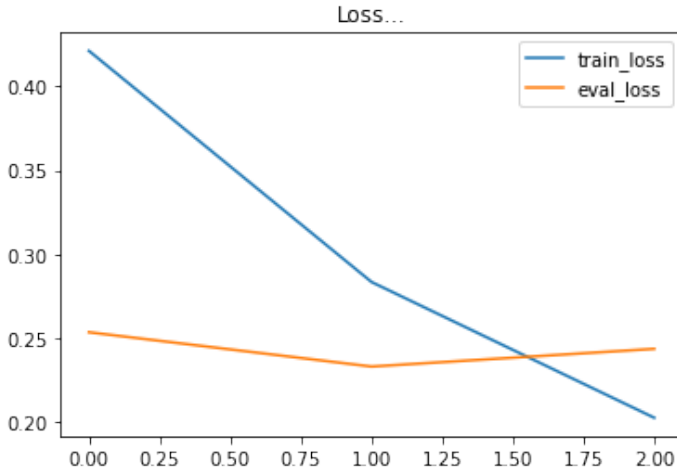
| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 135 |
| | FP | 4 |
| | TN | 115 |
| | FN | 38 |
| <i>Recall</i> | Hoax | 97% |
| | Valid | 78% |
| <i>Precision</i> | Hoax | 75% |
| | Valid | 97% |
| <i>F1-Score</i> | Hoax | 85% |
| | Valid | 87% |
| <i>Accuracy</i> | | 86% |
| <i>Avg. Training Time (mnt)</i> | | 02:07 |

del, ketiga metode yang diuji menggunakan model yang dibuat oleh orang yang sama, yaitu Cahya Wirawan dan di-*pretrained* dengan sumber dan jumlah dataset yang sama. Tabel 4.12 merupakan konfigurasi yang digunakan pada waktu melakukan *training* pada pengujian ini.

4.3.1 Pengujian dengan menggunakan ROBERTA

ROBERTA merupakan pengembangan lebih lanjut dari BERT yang dibuat oleh tim Facebook AI. Salah satu pengembangan yang dilakukan adalah merubah bagaimana teks di *mask* pada saat melakukan *pretrain*.

Dikarenakan model ROBERTA ini adalah model hasil pengembangan lebih lanjut dari BERT, terlihat dari grafik 4.9 bahwa nilai *loss* dari model ini sangat bagus. Sedangkan pada tabel 4.13, terli-



Gambar 4.6: Nilai *Loss* saat Pengujian dengan model *cahya-522M*

hat bahwa nilai akurasi hanya terpaut 1% dari pada BERT.

4.3.2 Pengujian dengan menggunakan GPT-2

Sebagai pembandingan lebih lanjut, kami menggunakan GPT-2. Sebuah model yang cukup terkenal karena kemampuan generator teks-nya yang menghasilkan teks yang bagus dan masih dapat dimengerti oleh manusia.

Model menggunakan nilai *epoch* yang paling sedikit apabila dibandingkan model - model yang lain. Hal ini karena tidak adanya *node dropout* pada *head*. Walaupun begitu, seperti bisa dilihat pada grafik 4.10, nilai *loss*-nya juga cukup bagus. Namun, apabila merubah nilai *epoch* menjadi 3, akan langsung terjadi *overfit*. Tingkat akurasi yang dijelaskan pada tabel 4.14 juga lebih baik dari ROBERTA, namun masih kalah apabila dibandingkan dengan BERT.

Tabel 4.9: Hasil Pengujian dengan model *cahya-522M*

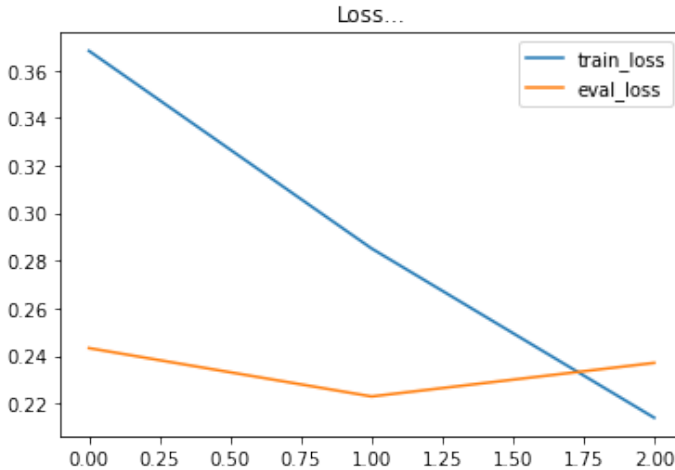
| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 122 |
| | FP | 17 |
| | TN | 122 |
| | FN | 31 |
| <i>Recall</i> | Hoax | 88% |
| | Valid | 80% |
| <i>Precision</i> | Hoax | 80% |
| | Valid | 88% |
| <i>F1-Score</i> | Hoax | 84% |
| | Valid | 84% |
| <i>Accuracy</i> | | 84% |
| <i>Avg. Training Time (mnt)</i> | | 02:03 |

4.3.3 Pengujian dengan menggunakan BERT

Model BERT yang digunakan disini sama seperti pada model yang digunakan pada pengujian berdasarkan model BERT yang digunakan. Berhubung model yang digunakan sama, maka tidak banyak terjadi perbedaan.

Apabila dibandingkan dengan model - model transformer yang sudah diuji sebelumnya, model BERT ini memiliki tingkat performa yang lebih bagus, tampak pada tabel 4.15. Namun, apabila melihat pada gambar 4.11, tampak model BERT ini memiliki nilai *loss* yang lebih jelek dan mendekati kondisi *overfit*.

4.4 Pengujian Performa berdasarkan pada pende-



Gambar 4.7: Nilai *Loss* saat Pengujian dengan model *cahya-1.5G*

katan cara *training*

Selain membandingkan antara model - model yang sudah dibuat sebelumnya oleh beberapa orang di internet, kami juga mencoba untuk memperbaiki model kami sehingga menjadi lebih optimal dibandingkan sebelumnya. Untuk pengujian ini, kami menggunakan model *indobert-base-p1* karena memiliki tingkat akurasi yang paling tinggi pada pengujian - pengujian sebelumnya. Sebagai *baseline* atau dasar, kami menggunakan nilai yang didapat pada pengujian sebelumnya. Untuk pengujian ini kami mencoba beberapa hal dengan rincian dapat dilihat pada tabel 4.16

4.4.1 *Baseline*

Diperlukan adanya nilai *baseline* yang digunakan sebagai acuan akan bagaimana suatu performa model setelah diberikan pendekatan cara *training* yang berbeda. Tabel 4.17 adalah performa yang didapatkan dengan konfigurasi seperti yang disebutkan diatas.

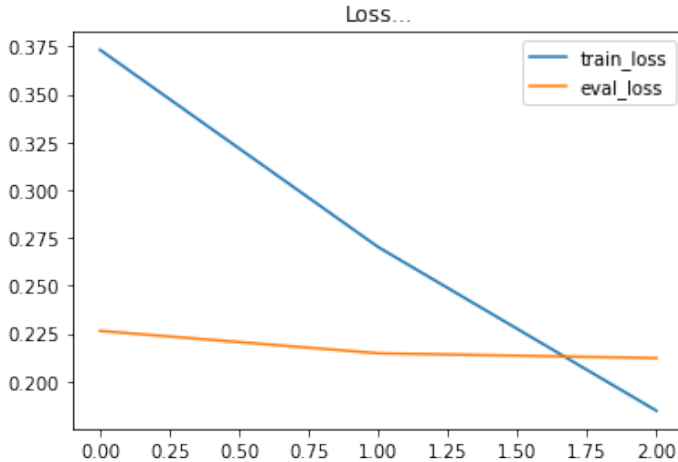
Tabel 4.10: Hasil Pengujian dengan model *cahya-1.5G*

| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 130 |
| | FP | 9 |
| | TN | 123 |
| | FN | 30 |
| <i>Recall</i> | Hoax | 93% |
| | Valid | 81% |
| <i>Precision</i> | Hoax | 80% |
| | Valid | 94% |
| <i>F1-Score</i> | Hoax | 86% |
| | Valid | 87% |
| <i>Accuracy</i> | | 87% |
| <i>Avg. Training Time (mnt)</i> | | 02:08 |

4.4.2 Parameter *Freeze*

Pada pengujian ini kami mencoba untuk tidak merubah *weight* yang sudah didapatkan pada proses *pre-trained* pada waktu model dibuat. Sehingga, *weight* yang kami rubah adalah *weight* pada bagian *classifier*-nya saja.

Berdasarkan pada tabel 4.18, dapat dilihat bahwa waktu yang dibutuhkan model untuk melakukan *training* menjadi jauh lebih singkat dan dengan tingkat akurasi tidak terpaud jauh apabila dibandingkan tingkat akurasi *baseline*. Sayangnya, apabila melihat gambar 4.13, terlihat bahwa parameter yang di-*freeze* tersebut membuat model tidak dapat mencapai tingkat akurasi optimal bahkan setelah 50 epoch walaupun tampak grafik *loss* model sangat bagus.



Gambar 4.8: Nilai *Loss* saat Pengujian dengan model *indobert*

4.4.3 Parameter *Dropout*

Pada pengujian ini kami ingin melihat pengaruh perubahan parameter *dropout* pada performa suatu model. Perubahan nilai *dropout* ini dapat digunakan agar model tidak mengalami *overfit* dan menjadi lebih umum untuk digunakan.

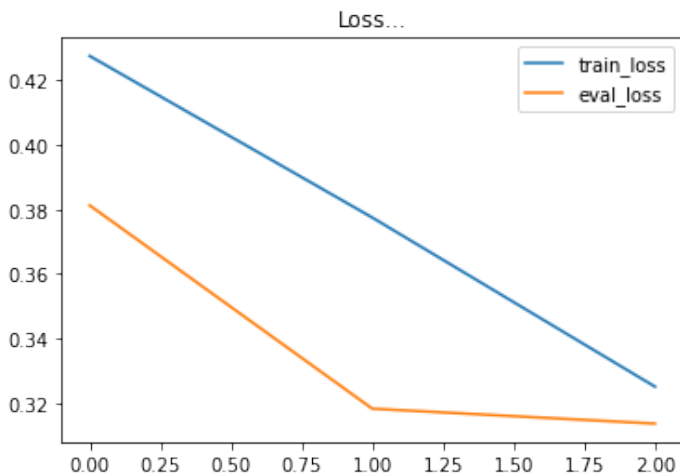
Apabila melihat pada tabel 4.19, dapat dilihat bahwa akurasi model menjadi lebih buruk apabila dibandingkan dengan pengujian acuan. Namun, di saat bersamaan, nilai *loss* yang terdapat pada gambar 4.14 lebih bagus apabila dibandingkan dengan hasil pengujian acuan.

Tabel 4.11: Hasil Pengujian dengan model *indobert*

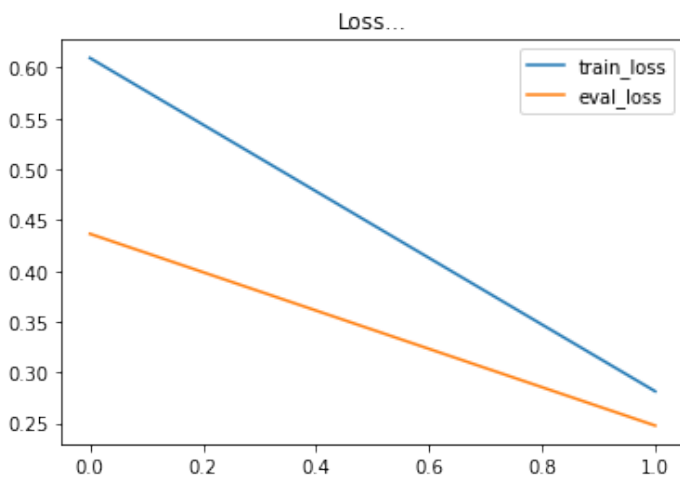
| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 122 |
| | FP | 17 |
| | TN | 137 |
| | FN | 16 |
| <i>Recall</i> | Hoax | 89% |
| | Valid | 88% |
| <i>Precision</i> | Hoax | 90% |
| | Valid | 88% |
| <i>F1-Score</i> | Hoax | 89% |
| | Valid | 88% |
| <i>Accuracy</i> | | 89% |
| <i>Avg. Training Time (mnt)</i> | | 02:05 |

Tabel 4.12: Konfigurasi yang dipakai oleh model *transformer* yang digunakan

| Model | <i>epoch</i> | <i>dropout</i> | <i>learning rates</i> |
|---------|--------------|----------------|-----------------------|
| Roberta | 3 | 0.1 | 2e-5 |
| GPT-2 | 2 | - | 2e-5 |
| BERT | 3 | 0.1 | 2e-5 |



Gambar 4.9: Nilai *Loss* pada model ROBERTA



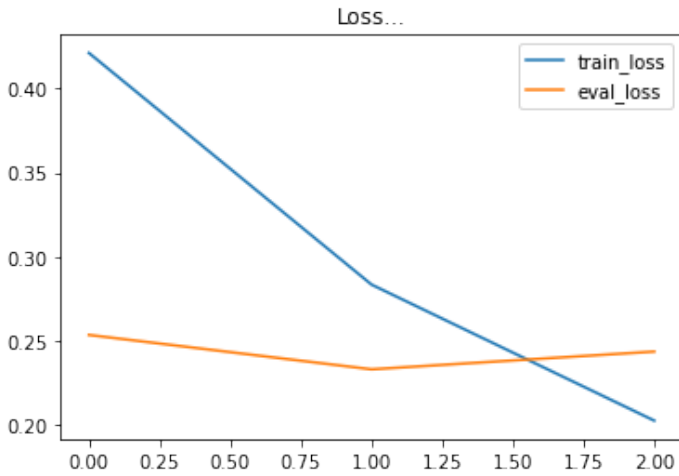
Gambar 4.10: Nilai *Loss* pada model GPT-2

Tabel 4.13: Hasil Pengujian pada model ROBERTA

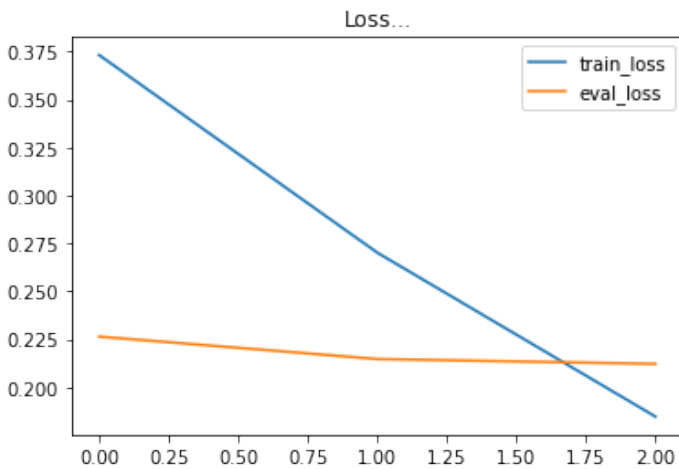
| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 127 |
| | FP | 12 |
| | TN | 111 |
| | FN | 42 |
| <i>Recall</i> | Hoax | 90% |
| | Valid | 75% |
| <i>Precision</i> | Hoax | 73% |
| | Valid | 91% |
| <i>F1-Score</i> | Hoax | 80% |
| | Valid | 82% |
| <i>Accuracy</i> | | 82% |
| <i>Avg. Training Time (mnt)</i> | | 02:12 |

Tabel 4.14: Hasil Pengujian pada model GPT-2

| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 118 |
| | FP | 21 |
| | TN | 124 |
| | FN | 29 |
| <i>Recall</i> | Hoax | 86% |
| | Valid | 80% |
| <i>Precision</i> | Hoax | 81% |
| | Valid | 85% |
| <i>F1-Score</i> | Hoax | 83% |
| | Valid | 83% |
| <i>Accuracy</i> | | 83% |
| <i>Avg. Training Time (mnt)</i> | | 02:15 |



Gambar 4.11: Nilai *Loss* pada model BERT



Gambar 4.12: Nilai *loss* Acuan

Tabel 4.15: Hasil Pengujian pada model BERT

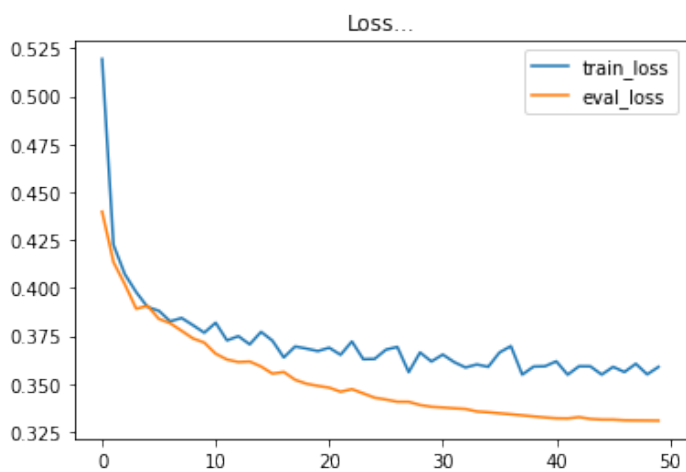
| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 122 |
| | FP | 17 |
| | TN | 122 |
| | FN | 31 |
| <i>Recall</i> | Hoax | 88% |
| | Valid | 80% |
| <i>Precision</i> | Hoax | 80% |
| | Valid | 88% |
| <i>F1-Score</i> | Hoax | 84% |
| | Valid | 84% |
| <i>Accuracy</i> | | 84% |
| <i>Avg. Training Time (mnt)</i> | | 02:03 |

Tabel 4.16: Konfigurasi yang digunakan pada saat melakukan pendekatan training

| pendekatan | <i>epoch</i> | <i>dropout</i> | <i>learning rates</i> |
|-------------------------|--------------|----------------|-----------------------|
| <i>baseline</i> | 3 | 0.1 | 2e-5 |
| <i>parameter freeze</i> | 50 | 0.1 | 2e-5 |
| <i>dropout</i> | 3 | 0.2 | 2e-5 |

Tabel 4.17: Hasil Pengujian Acuan

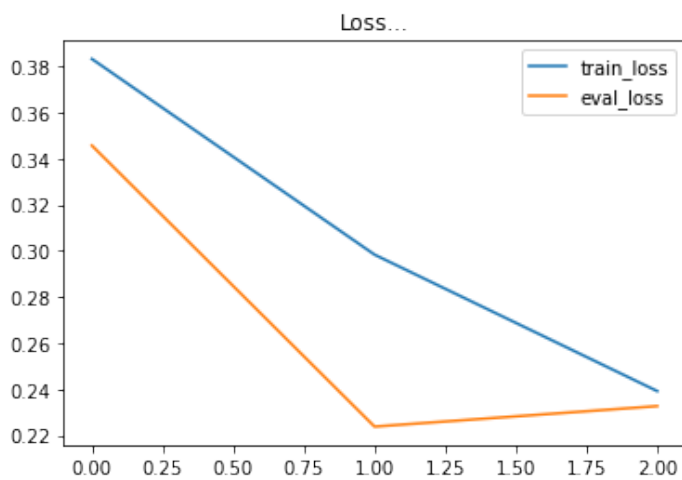
| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 122 |
| | FP | 17 |
| | TN | 137 |
| | FN | 16 |
| <i>Recall</i> | Hoax | 89% |
| | Valid | 88% |
| <i>Precision</i> | Hoax | 90% |
| | Valid | 88% |
| <i>F1-Score</i> | Hoax | 89% |
| | Valid | 88% |
| <i>Accuracy</i> | | 89% |
| <i>Avg. Training Time (mnt)</i> | | 02:05 |



Gambar 4.13: Nilai *loss* saat melakukan *training* dengan parameter *freeze*

Tabel 4.18: Hasil pengujian saat *training* dengan parameter *freeze*

| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 127 |
| | FP | 12 |
| | TN | 112 |
| | FN | 41 |
| <i>Recall</i> | Hoax | 90% |
| | Valid | 76% |
| <i>Precision</i> | Hoax | 73% |
| | Valid | 91% |
| <i>F1-Score</i> | Hoax | 81% |
| | Valid | 83% |
| <i>Accuracy</i> | | 82% |
| <i>Avg. Training Time (mnt)</i> | | 00:44 |



Gambar 4.14: Nilai *loss* saat melakukan *training* dengan merubah *dropout*

Tabel 4.19: Hasil pengujian saat *training* dengan merubah *dropout*

| Hasil Model | | Nilai |
|---------------------------------|-------|-------|
| <i>Confusion Matrix</i> | TP | 111 |
| | FP | 28 |
| | TN | 134 |
| | FN | 19 |
| <i>Recall</i> | Hoax | 83% |
| | Valid | 85% |
| <i>Precision</i> | Hoax | 88% |
| | Valid | 80% |
| <i>F1-Score</i> | Hoax | 85% |
| | Valid | 83% |
| <i>Accuracy</i> | | 84% |
| <i>Avg. Training Time (mnt)</i> | | 03:44 |

BAB 5

PENUTUP

5.1 Kesimpulan

Dari keseluruhan penelitian yang telah dilakukan, maka dapat dimabil beberapa kesimpulan sebagai berikut :

1. Semakin banyak dataset yang digunakan pada waktu *pretrain* BERT, maka semakin akurat juga suatu model tersebut, hal ini terbukti oleh model *indobert-base-p1* yang dilatih dengan 23 GB lebih data.
2. Metode pemenggalan kata dengan hanya mengambil bagian awal dari suatu teks berhasil memperoleh tingkat akurasi yang lebih baik apabila dibandingkan dengan metode - metode pemenggalan teks lainnya.
3. Penggunaan model BERT yang spesifik untuk bahasa Indonesia secara umum memiliki nilai akurasi yang lebih baik apabila dibandingkan dengan model multibahasa maupun bahasa Melayu dalam mendeteksi berita palsu berbahasa Indonesia. Bahkan nilai akurasi yang lebih tinggi juga didapatkan dengan model yang di-*pretrain* dengan data yang lebih sedikit.
4. Walaupun sudah terdapat beberapa model lain yang merupakan turunan dari BERT itu sendiri, namun BERT masih merupakan model yang cukup bagus untuk klasifikasi teks.
5. BERT adalah model yang sangat mudah terjadi *overfit*, sehingga pengaturan parameter seperti dengan *dropout* dan parameter *freeze* akan membuat model lebih *general* namun memiliki tingkat akurasi beberapa persen lebih rendah.

5.2 Saran

Adapun dari penelitian ini terdapat beberapa saran dari penulis yang sekiranya dapat membantu untuk meningkatkan hasil dari penelitian ini, saran - saran tersebut adalah :

1. Dataset yang digunakan dalam penelitian ini masih dirasa kurang dan dapat diperbanyak lagi. Semakin banyak dataset yang digunakan semakin bagus pula tingkat akurasi modelnya.
2. Dataset yang digunakan dalam penelitian ini adalah dataset yang diambil langsung dari situs - situs di internet. Walaupun situs - situs tersebut adalah resmi, namun keabsahan datanya masih harus diverifikasi lagi, lebih spesifik lagi, data dengan label hoaks masih harus diverifikasi lagi.
3. Salah satu kelemahan BERT adalah jumlah token yang dapat diprosesnya dalam sekali waktu. Sudah terdapat penelitian lain hasil pengembangan dari BERT namun tidak memiliki limitasi jumlah token seperti BERT.
4. Membuat sistem yang sudah terintegrasi sehingga memudahkan masyarakat dalam mendeteksi berita palsu.

DAFTAR PUSTAKA

- [1] Mathwork Team. What is deep learning ?, . URL <https://www.mathworks.com/discovery/deep-learning.html>. diakses 10 Juli 2021.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.
- [3] Rani and Ni Luh Ratih Maha. Persepsi jurnalis dan praktisi humas terhadap nilai berita. 2013. doi: 10.24002/jik.v10i1.155.
- [4] Wikipedia. Berita bohong. URL https://id.wikipedia.org/wiki/Berita_bohong. diakses 27 November 2020.
- [5] Kementrian Kominfo. Temuan isu hoaks, 03 2020. URL https://eppid.kominfo.go.id/storage/uploads/2_12_Data_Statistik_Hoax_Agustus_2018_-_31_Maret_2020.pdf.
- [6] Kementrian Kominfo. Laporan isu hoax juni 2020, 07 2020. URL https://eppid.kominfo.go.id/storage/uploads/2_31_Laporan_Isu_Hoaks_Bulan_Juni_2020.pdf.
- [7] M. Laeeq Khan and Ika Idris. Recognize misinformation and verify before sharing: A reasoned action and information literacy perspective. *Behaviour and Information Technology*, 01 2019. doi: 10.1080/0144929X.2019.1578828.
- [8] Kunto Wibowo, Detta Rahmawan, and Eni Maryani. Penelitian di indonesia: umur tidak mempengaruhi kecenderungan orang menyebarkan hoaks, 2019. URL <https://theconversation.com/penelitian-di-indonesia-umur-tidak-mempengaruhi-kecenderungan-orang-menyebarkan-hoaks-110621>. diakses 27 November 2020.

- [9] Tempo. Kerusuhan di waena dan wamena, 33 jiwa tewas akibat terpicu hoaks, 10 2019. URL <https://grafis.tempo.co/read/1831/kerusuhan-di-waena-dan-wamena-33-jiwa-tewas-akibat-terpicu-hoaks>. diakses 02 Desember 2020.
- [10] IBM Cloud Education. Machine learning, 06 2020. URL <https://www.ibm.com/cloud/learn/machine-learning>. diakses 10 Juli 2021.
- [11] Sara Brown. Machine learning, explained, 04 2021. URL <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>. diakses 10 Juli 2021.
- [12] Microsoft Azure Cloud Team. What is machine learning ?, . URL <https://azure.microsoft.com/en-us/overview/what-is-machine-learning-platform/>. diakses 10 Juli 2021.
- [13] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.
- [14] Cahya Wirawan. bert-base-indonesian-1.5g, 12 2020. URL <https://huggingface.co/cahya/bert-base-indonesian-1.5G>. diakses 11 Juli 2021.
- [15] Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp. *arXiv preprint arXiv:2011.00677*, 2020.
- [16] banerjeer2611. Understanding accuracy, recall, precision, f1 scores, and confusion matrices, 01 2021. URL <https://www.realpythonproject.com/understanding-accuracy-recall-precision-f1-scores-and-confusion-matrices/>. diakses 12 Juli 2021.
- [17] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National*

Conference on Chinese Computational Linguistics, pages 194–206. Springer, 2019.

- [18] Zolkepli Husein. Natural-language-toolkit library for baha-sa malaysia, powered by deep learning tensorflow. <https://github.com/huseinzol105/malaya>, 2018.

[Halaman ini sengaja dikosongkan]