

Deteksi Berita Palsu Otomatis Berbahasa Indonesia Menggunakan BERT

Reza Fuad Rachmadi
Departemen Teknik Komputer
Fakultas Teknologi Elektro
dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
fuad@te.its.ac.id

Mauridhi Hery Purnomo
Departemen Teknik Komputer
Fakultas Teknologi Elektro
dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
hery@ee.its.ac.id

Aufa Nabil Amiri
Departemen Teknik Komputerr
Fakultas Teknologi Elektro
dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
aufa.17072@mhs.its.ac.id

Abstrak—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

I. 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 [1]. 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 [2]. 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 Kementerian 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 [3]. Masih dari sumber yang sama, pada bulan Juni 2020, hampir setiap harinya ditemukan puluhan berita palsu baru [4].

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 [5]. Survey lain yang dilakukan oleh Kunto dengan melibatkan 480 responden di Kota Jawa Barat menunjukkan bahwa sekitar 30% masyarakat Jawa Barat memiliki kecenderungan menengah sampai tinggi untuk menyebarkan berita palsu [6]. 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.

Sudah terdapat beberapa penelitian yang pernah dilakukan oleh orang lain mengenai pendeteksi berita hoaks ini. Aggarway et al. pernah melakukan penelitian untuk membandingkan antara BERT, XGBoost dan LSTM untuk melakukan klasifikasi berita palsu berbahasa Inggris. Dari penelitian tersebut didapatkan bahwa BERT memiliki tingkat akurasi yang lebih tinggi apabila dibandingkan dengan XGBoost dan LSTM [7]. Bahad et al. melakukan penelitian yang membandingkan antara CNN, RNN, *uni-directional* LSTM RNN dan *bi-directional* LSTM RNN. Hasil dari penelitian tersebut menunjukkan bahwa penggunaan LSTM ditambah dengan *attention* baik itu *uni-directional* maupun *bi-directional* memiliki tingkat akurasi yang lebih tinggi apabila dibandingkan dengan CNN atau RNN [8]. Dari kedua penelitian tersebut, dapat diambil kesimpulan bahwa algoritma yang 'mengingat' atau mengetahui suatu konteks dalam teks akan memiliki tingkat akurasi yang lebih tinggi dibanding algoritma dengan pendekatan yang lain.

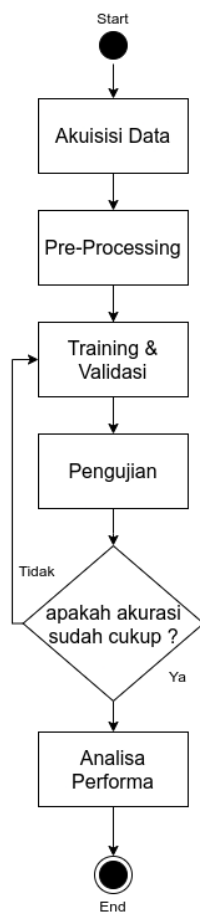
Untuk penelitian pendeteksi berita hoaks dengan berbahasa Indonesia, terdapat beberapa penelitian yang pernah dilakukan seperti oleh Prasetijo et al. yang meneliti penggunaan SVM dan SGD untuk mendeteksi berita hoaks berbahasa Indonesia. Penelitian tersebut berhasil membuat suatu model dengan

tingkat akurasi sebesar 85% [9]. Penelitian lain yang dilakukan oleh Rahutomo et al. dengan menggunakan algoritma *naive bayes* berhasil menghasilkan akurasi sebesar 80% [10].

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.

II. DESAIN DAN IMPLEMENTASI SISTEM

Paper ini membahas mengenai implementasi salah satu cabang ilmu dalam *deep learning*, yang bertujuan untuk mendeteksi suatu berita hoaks berbahasa indonesia secara otomatis menggunakan metode BERT. Pendeteksi ini dilatih menggunakan gabungan dataset dari <https://data.mendeley.com/datasets/p3hfgr5j3m/1> dan dataset yang kami buat sendiri menggunakan teknologi *web crawling*. Gambar 1 merupakan garis besar penelitian ini.



Gambar 1. Garis besar metodologi penelitian.

A. Material dan Spesifikasi Alat

Pada penelitian ini, dataset yang digunakan adalah dataset yang berasal dari <https://data.mendeley.com/datasets/p3hfgr5j3m/1> yang digabungkan dengan dataset yang kami buat sendiri menggunakan teknologi *web crawling*. Total dataset yang berhasil kami kumpulkan adalah 1621 data dengan

rincian tertera pada tabel II, sedangkan pada tabel I merupakan dataset awal yang kami dapatkan dari data.mendeley.com.

Dataset tersebut berisi isi berita disertai dengan label Valid atau Hoaks. Sumber berita dataset yang kami gunakan berasal dari berbagai sumber yang sudah terakreditasi untuk berita yang valid, dan dari berbagai sumber yang kurang terakreditasi untuk berita yang hoaks. Tabel III merupakan contoh sebagian data yang kami gunakan dalam penelitian ini.

Tabel I
JUMLAH DATASET DARI DATA.MENDELEY.COM

Label	Jumlah Data
Hoaks	228
Valid	372
Total	600

Tabel II
JUMLAH DATASET YANG DIGUNAKAN

Label	Jumlah Data
Hoaks	885
Valid	736
Total	1621

Tabel III
CONTOH DATASET

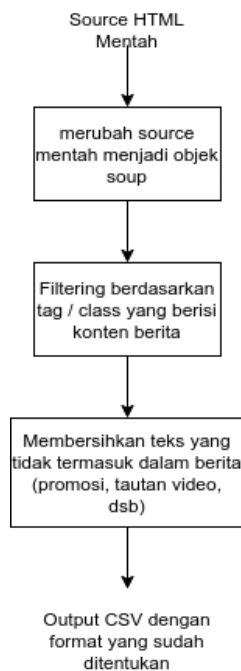
berita	tagging
Wakil Gubernur DKI Jakarta Sandiaga Uno menargetkan pengerjaan tahap awal Stadion BMW dilakukan pada Oktober. Stadion ini diperuntukkan bagi klub Persija....	Valid
"Komisi II bersama KPU dan Bawaslu masih membahas ketentuan wajib cuti bagi petahana presiden yang maju Pilpres 2019. Mekanisme pengambilan.....	Valid
Jaksa penuntut Ulumum (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 ditujukan buat islam. Kl mau jd komunis pake aja istilah kadrun buat umat islam. Auto lsg Komunis"	Hoax

B. Akuisisi Data

Berhubung dataset dari <https://data.mendeley.com/datasets/p3hfgr5j3m/1> dirasa sangat kurang karena hanya berjumlah 600 data saja. Maka dari itu kami membuat sebuah program *web crawling* yang digunakan untuk mengambil teks berita dari beberapa situs berita yang sudah terakreditasi seperti liputan6.com, detik.com, tempo.com. Sedangkan untuk teks berita hoaks, sumber yang kami gunakan adalah turnbackhoax.id. Alasan kami menggunakan situs turnbackhoax.id adalah

karena dibalik situs tersebut terdapat suatu forum yang cukup aktif, setiap berita hoaks yang ditemukan akan diunggah ke forum tersebut terlebih dahulu sebelum akhirnya akan diunggah ke situs utama, diharapkan dengan proses semacam itu, maka suatu unggahan sudah dilihat oleh orang lain selama beberapa kali sehingga menurunkan kemungkinan terjadi bias. Selain itu, situs turnbackhoax.id juga menuliskan teks asli dari berita hoaks yang diunggah dan memiliki suatu format dalam setiap unggahan di situs tersebut sehingga memudahkan untuk melakukan *webcrawling*. Kami melakukan *webcrawling* dengan mengambil tanggal yang bervariasi antara April 2018 sampai dengan April 2021.

Gambar 2 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.



Gambar 2. Garis besar alur program *web crawl*.

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.

```

...
<div class="detail__body_itp_bodycontent_wrapper">
<div class="detail__body-text_itp_bodycontent">

<strong>Jakarta</strong> - Koalisi <a href="https://
detik.com/tag/jokowi" target="_blank">Jokowi</a>

```

```

sedang menyusun visi-misi jagoannya. Setelah
menerima masukan dari <a href="https://detik.com/
tag/muhammadiyah" target="_blank"> Muhammadiyah</a>,
...
Dan kita pun membuka diri untuk menerima
masukan untuk penyempurnaan," imbuhnya.<br><br><!--
s:parallaxindetail--><div class="clearfix"></div><style>
...

```

Listing 1. Penggalan Kode Sumber HTML detik.com.

Yang pertama kali harus kami lakukan adalah menentukan *tag* atau *class* HTML yang akan kami gunakan untuk melakukan penyaringan terlebih dahulu. Apabila merujuk pada listing 1 *class* yang berisi teks seluruh berita adalah *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 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* 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 filter tersebut.

C. 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. Langkah terakhir adalah menambahkan token [CLS] di awal kalimat. Untuk lebih jelasnya, bisa melihat pada Gambar 3.

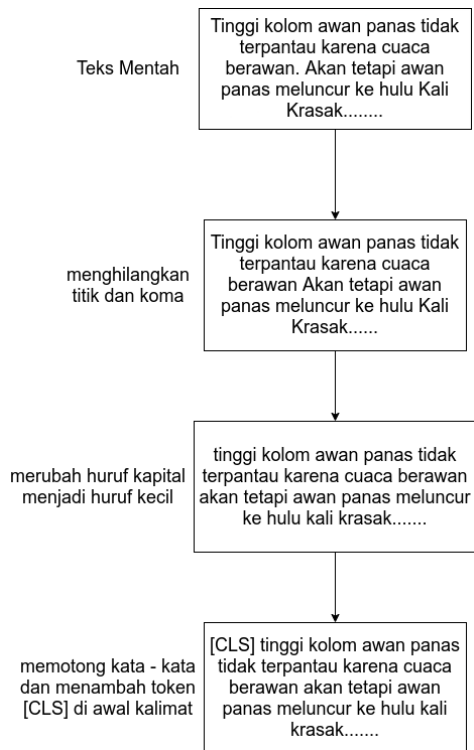
Selain itu, juga akan dilakukan pembagian dataset yang awalnya berjumlah 1621 akan dibagi menjadi 3 bagian dengan ketentuan :

- 70% *Training*, 10% *Validasi*, 20% *Pengujian*

1) *Training*

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

2) *Validasi*



Gambar 3. Metode Preprocessing

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 dari pengujian inilah yang akan digunakan sebagai hasil dari model.

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

Tabel IV
RINCIAN PEMBAGIAN DATASET

Bagian	Hoaks	Valid	Total Data
Training	647	519	1166
Validasi	85	78	163
Pegujian	153	139	292
Total			1621

D. Perancangan Arsitektur BERT

BERT adalah salah satu implementasi pembelajaran mesin yang paling mutakhir saat ini. Merupakan pengembangan

lebih lanjut dari implementasi Transformer, BERT memiliki tingkat akurasi yang lebih tinggi dalam memahami konteks pada suatu teks mentah apabila dibandingkan dengan beberapa implementasi pengembangan transformer lain.

Salah satu hal yang membedakan antara BERT dengan model - model yang lain adalah cara BERT di *training*. Terdapat 2 langkah yang dilakukan untuk melakukan *training* pada BERT. Yang pertama adalah dengan menutup sebagian kata - kata yang dijadikan sebagai masukan BERT, atau yang biasa disebut sebagai MLM (*Masked Language Model*). Model akan diminta untuk melihat apa hubungan antara kata yang ditutup tersebut dengan kata lainnya. Langkah selanjutnya adalah *Next Sentence Prediction* dimana BERT akan digunakan untuk menentukan apakah suatu kalimat B merupakan kelanjutan dari kalimat A. Kedua metode *training* tersebut adalah *unsupervised task* sehingga hanya diperlukan dataset saja.

Dalam penelitian ini, kami menggunakan pendekatan *fine-tuning*, dimana model BERT yang sudah di *pre-training* sebelumnya, akan kami gabungkan dengan *layer* klasifikasi. Dalam hal ini, kami menggunakan *Linear Regression* sebagai *layer* klasifikasi. Gambar 4 merupakan arsitektur BERT untuk penelitian ini.

E. Training dan Validasi

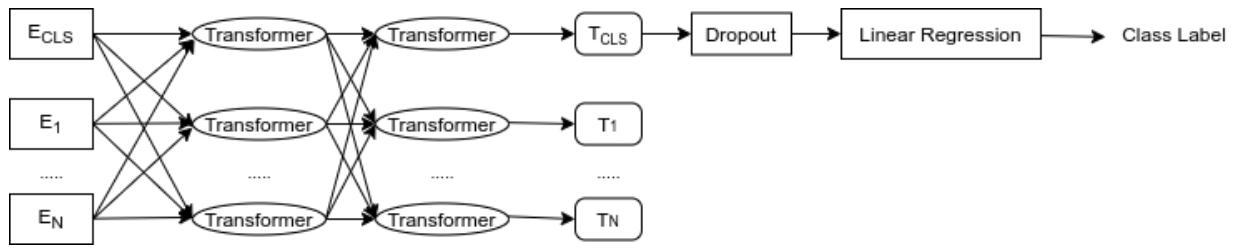
Pada tahap ini, teks yang sudah melewati proses *preprocessing* akan dilakukan proses Tokenizer. Tokenizer adalah proses untuk merubah kata - kata dalam teks menjadi token sesuai dengan *word embedding* yang sudah ada pada *pretrained* BERT. Barulah pada saat itu, BERT dapat melakukan *training* berdasarkan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

Keluaran dari BERT akan diambil isi token [CLS]-nya dan dimasukkan kedalam algoritma klasifikasi, seperti *Linear Regression*. *Linear Regression* digunakan sebagai algoritma klasifikasi yang cukup mudah namun memiliki tingkat akurasi yang cukup. Gambar 5 dapat digunakan sebagai penjelas.

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, 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 *hiperparameter* yang dirubah selama proses *training*. *hiperparameter* digunakan untuk merubah *weight* selama proses *training* berdasarkan



Gambar 4. Arsitektur BERT dalam penelitian ini



Gambar 5. Metode Training

feedback saat proses validasi. Disini kami menggunakan nilai yang direkomendasikan oleh pembuat model BERT yang kami gunakan, yaitu 0.00002 [11].

Selanjutnya, dari model yang sudah dibentuk pada saat proses *training*, akan dilakukan pengujian terhadap data yang sama sekali baru dan tidak digunakan selama proses *training*. Hal ini untuk menghindari bias yang mungkin terjadi apabila model diuji pada data yang sama yang digunakan pada saat proses *training*. Proses inilah yang disebut dengan proses validasi.

Hasil validasi model akan diambil nilai *loss*-nya dan dibandingkan dengan nilai *loss* pada saat proses *training*. Apabila nilai *loss* validasi semakin tinggi, namun nilai *loss training* semakin rendah, maka model sudah mengalami *overfitting*, sebaliknya, apabila baik nilai *loss training* dan nilai *loss* validasi masih tinggi, kemungkinan model mengalami *underfitting*. Kedua kondisi tersebut merupakan tanda bahwa model masih dapat dioptimisasi lagi, sehingga harus dilakukan *re-training* sambil menyesuaikan parameter yang diberikan.

F. Pengujian

Setelah melakukan proses validasi dan *training*, maka yang harus dilakukan adalah melakukan pengujian terhadap model yang sudah dibuat. Dari proses ini dapat diambil kesimpulan apakah model tersebut sudah cukup baik, ataukah masih dapat dilakukan perbaikan atau optimisasi lagi salah satunya dengan cara mengatur ulang konfigurasi - konfigurasi yang sudah diatur pada saat proses *training*.

G. Analisa Performa

Setelah melakukan proses pengujian, langkah berikutnya adalah melakukan analisa performma 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 mendapat pengelompokkan berdasarkan hasil prediksi dan dataset acuan sehingga 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*.

III. HASIL DAN PEMBAHASAN

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 :

nolistsep

- 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 V

Tabel V
SPESIFIKASI PC YANG DIGUNAKAN

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

A. 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, dengan rincian sebagai berikut :

1) Mengambil bagian awal teks

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

2) 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.

3) Mengambil 129 token dari bagian awal teks dan 383 token dari bagian 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 [12]. Alasan dari penyebab lebih tingginya akurasi adalah karena dengan mengambil sebagian di awal maka sebagian dari ringkasan berita akan didapatkan, sedangkan 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.

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 VI 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 sebelumnya.

Seperti bisa dilihat pada tabel VII, metode pemotongan teks

Tabel VI
KONFIGURASI PARAMETER UNTUK PENGUJIAN BERDASARKAN PEMOTONGAN KATA

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

Tabel VII
PERFORMA PADA PENGUJIAN BERDASAR PADA LOKASI PEMOTONGAN KATA

lokasi pemotong-an	<i>recall</i>	<i>precision</i>	<i>f1-score</i>	<i>accuracy</i>
awal	89%	90%	89%	89%
akhir	88%	85%	86%	86%
gabungan (129 awal + 383 akhir)	88%	88%	88%	87%

dengan hanya mengambil bagian awal saja memiliki tingkat akurasi yang paling tinggi dan memiliki nilai *recall* dan *precision* yang seimbang. Hal ini berbeda dengan apabila dilakukan pemotongan pada bagian akhir teks yang menunjukkan ada kemungkinan lebih besar bagi model untuk mengklasifikasi suatu teks berita termasuk ke dalam berita palsu. Metode pemotongan yang menggabungkan 129 token dari awal dan 383 token dari bagian akhir teks memiliki rasio nilai *recall* dan *precision* yang bagus juga sama seperti rasio nilai pada pemotongan pada bagian awal teks, namun, secara isi nilai masih kalah.

B. 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 digunakan 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 menggunakan beberapa model dengan rincian sebagai berikut :

1) bert-base-bahasa-standard-case (*bert-bahasa*)

Merupakan model BERT yang dibuat oleh huzeinzol05. Model ini hanya mendukung bahasa Melayu, namun pembuat model ini mengklaim bahwa model yang telah dibuat dapat digunakan untuk bahasa melayu maupun bahasa Indonesia karena kesamaan tata bahasa dan arti. Model dilatih pada dataset yang diambil dari Wikipedia berbahasa melayu, Wattpad, dan sosial media [13].

2) bert-base-multilingual-uncased (*bert-base*)

Model BERT ini adalah model dasar yang dibuat langsung oleh tim di Google. Model inilah yang merupakan hasil dari penelitian yang sudah dibuat oleh devlin et al dalam penelitian awal mengenai metode BERT ini. Dilatih menggunakan seluruh data yang dimiliki oleh Wikipedia, sehingga salah satu kelebihan dari model ini adalah kemampuannya dalam mendukung 104 bahasa sekaligus [14].

3) indobert-base-pl (*indobert*)

Model BERT ini adalah salah satu model BERT yang spesifik mendukung bahasa Indonesia. Model ini dibuat oleh 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 berbagai sumber seperti Wikipedia, Twitter, OpenSubtitle, dan banyak lagi [15].

4) bert-base-indonesian-522M (*cahya-522M*)

Model BERT yang spesifik mendukung hanya bahasa Indonesia yang dibuat oleh Cahya Wirawan. Model ini dilatih menggunakan 522M data yang semuanya berasal dari Wikipedia. Merupakan model dengan jumlah *pre-trained* dataset yang paling sedikit dibandingkan model - model yang lainnya.

5) bert-base-indonesian-1.5G (*cahya-1.5G*)

Sama seperti model sebelumnya, model ini juga dibuat oleh Cahya Wirawan. Yang menjadi pembeda dengan model sebelumnya adalah adanya penambahan dataset sebesar 1G dari situs - situs berita berbahasa Indonesia yang membuat total dataset yang digunakan pada waktu *pre-trained* adalah sebesar 1.5G

Tabel VIII

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-pl	3	0.1	2e-5
bert-base-indonesian-522M	3	0.1	2e-5
bert-base-indonesian-1.5G	3	0.2	2e-5

Untuk melakukan *training*, sebelumnya kami mengatur konfigurasi yang akan digunakan oleh model BERT yang sudah disiapkan. Tabel VIII adalah rincian konfigurasi yang kami gunakan pada setiap model. Terdapat beberapa perbedaan pada konfigurasi seperti jumlah *epoch* dan jumlah *dropout*. Hal ini karena apabila menggunakan parameter yang sama untuk setiap model, terdapat model yang mengalami *overfitting* dan ada juga model yang dirasa belum optimal.

Dari tabel IX, dapat disimpulkan bahwa apabila suatu model BERT menggunakan dataset yang sedikit pada waktu *pre-training*, maka waktu yang diperlukan untuk melakukan *fine-*

Tabel IX

TINGKAT AKURASI DARI SELURUH MODEL BERT YANG DIGUNAKAN

model	<i>recall</i>	<i>precision</i>	<i>f1-score</i>	<i>accuracy</i>	avg. training time
bert-bahasa	89%	82%	85%	85%	03:43
bert-base	97%	75%	85%	86%	02:07
indobert	89%	90%	89%	89%	02:05
cahya-522M	88%	80%	84%	84%	02:03
cahya-1.5G	93%	80%	86%	87%	02:08

tuning juga semakin sedikit, namun hal ini dibayar dengan tingkat akurasi yang lebih rendah apabila dibandingkan dengan model yang lain. Hal lain menunjukkan bahwa model BERT berbahasa Melayu kurang cocok digunakan sebagai pendeteksi berita palsu berbahasa Indonesia. Model yang dibuat oleh indobenchmark memiliki tingkat akurasi yang paling bagus dan disertai dengan nilai *precision* dan *recall* yang seimbang sehingga lebih dapat diandalkan dalam mendeteksi berita palsu.

C. Pengujian performa berdasarkan pada metode transformer yang digunakan.

BERT adalah metode *state-of-the-art* dan pengembangan lebih lanjut dari *transformer*. Namun, selain BERT, ada beberapa pengembangan lebih lanjut lain dari *transformer* dimana masing - masing memiliki keuntungan dan kekurangannya masing - masing. Maka dari itu, kami menguji bagaimana performa metode BERT dalam mendeteksi berita hoaks berbahasa Indoensia. Sebagai penyamaan dan sebagai variabel kontrol, kami menggunakan model yang dibuat oleh orang yang sama, yaitu Cahya Wirawan dan di-*pretrained* dengan sumber dan jumlah dataset yang sama, yaitu sebesar 522M. Berikut adalah rincian dari metode yang kami gunakan dalam pengujian ini :

1) ROBERTA

Merupakan pengembangan lebih lanjut dari BERT. Dalam jurnal awalnya, ROBERTA dilatih menggunakan *dynamically masking*, berbeda dengan *static masking* yang digunakan oleh BERT. Selain itu, ROBERTA dilatih dengan dataset yang lebih besar apabila dibandingkan dengan BERT yang pada akhirnya, membuat waktu yang dibutuhkan ROBERTA untuk melakukan *pre-training* menjadi lebih lama [16].

2) GPT-2

Adalah model yang terkenal karena kemampuannya dalam melakukan pembuatan teks secara otomatis namun masih dapat dimengerti oleh manusia dan masih memiliki konteks dalam suatu kalimat. Salah satu yang membedakan antara BERT dengan GPT-2 adalah bagaimana *attention head*-nya di konfigurasi, dimana dalam BERT *attention head* akan melihat kanan dan kiri suatu kata, sedangkan GPT-2 hanya melihat bagian kanan saja.

3) BERT

Model BERT ini merupakan model BERT yang sama seperti sebelum - sebelumnya, dimana model yang di-

gunakan adalah model *bert-base-indonesian-522M* oleh Cahya Wirawan. Walaupun pada pengujian sebelumnya model ini memiliki tingkat akurasi yang paling kecil, namun, disaat yang bersamaan model ini adalah model dengan waktu tercepat dalam melakukan *training*.

Tabel X
KONFIGURASI YANG DIGUNAKAN OLEH MODEL TRANSFORMER YANG DIGUNAKAN

Model	epoch	dropout	learning rates
Roberta	3	0.2	2e-6
GPT-2	2	-	2e-5
BERT	3	0.1	2e-5

Sedangkan pada tabel X merupakan konfigurasi yang digunakan pada pengujian ini. Terdapat perbedaan dalam pengaturan agar model mencapai hasil yang optimal, GPT-2 menggunakan *epoch* yang paling rendah dikarenakan tidak adanya parameter *dropout* pada bagian *classifier*.

Tabel XI
TINGKAT AKURASI DARI MODEL TRANSFORMER YANG DIGUNAKAN

model	recall	precision	f1-score	accuracy	avg. training time
ROBERTA	90%	73%	80%	82%	02:12
GPT-2	86%	81%	83%	83%	02:15
BERT	88%	80%	84%	84%	02:03

D. Pengujian performa berdasarkan pendekatan 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 model yang lebih optimal dibandingkan sebelumnya. Untuk pengujian ini, kami menggunakan model *indobert-base-pl* karena model tersebut memiliki tingkat akurasi yang paling tinggi dibandingkan model - model yang lain pada pengujian - pengujian sebelumnya. Dibawah ini adalah rincian dari pendekatan yang kami lakukan :

1) parameter *freeze*

Yang dimaksud dengan parameter *freeze* adalah ketika suatu model yang sudah di-*pretrained* sebelumnya tidak akan dirubah lagi *weight*-nya dan masih menggunakan nilai *weight* yang didapat pada waktu *pretrained*.

2) *dropout*

dropout adalah parameter yang digunakan oleh *optimizer* untuk melakukan penghapusan nilai *weight* sehingga kembali menjadi kondisi awal. Keuntungan dari penghapusan *weight* ini adalah untuk menghindari kemungkinan *overfit* dan membuat model lebih tahan terhadap perubahan data sebagai masukan.

Tabel XII adalah rincian parameter yang kami gunakan dalam pengujian ini. Pada parameter *freeze* kami mengatur *epoch* sebanyak 50 dan masih belum mengalami *overfit* sehingga dapat diteruskan sampai mencapai titik optimal. Namun,

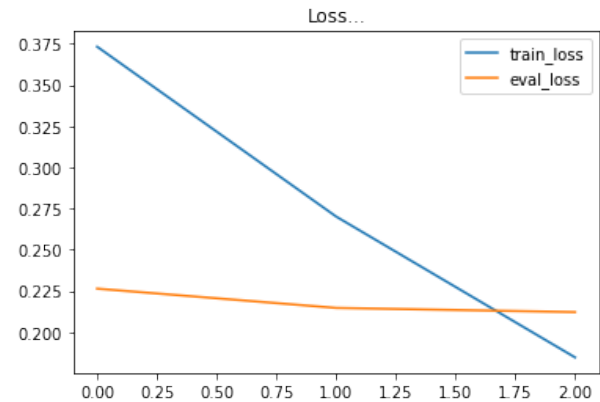
Tabel XII
KONFIGURASI YANG DIGUNAKAN SAAT MELAKUKAN PENGUJIAN PENDEKATAN TRAINING

Model	epoch	dropout	learning rates
baseline	3	0.1	2e-5
parameter freeze	50	0.1	2e-5
dropout	3	0.2	2e-5

karena perubahan nilai *loss* yang terjadi sudah sangat kecil, kami memutuskan untuk menghentikan pengujian pada 50 *epoch*.

Tabel XIII
TINGKAT AKURASI PADA PENGUJIAN PENDEKATAN CARA TRAINING

model	recall	precision	f1-score	accuracy	avg. training time
baseline	89%	90%	89%	89%	02:05
parameter freeze	90%	73%	81%	82%	00:44
dropout	83%	88%	85%	84%	03:44



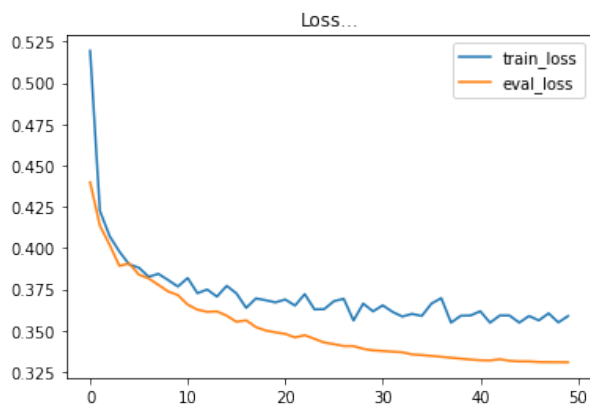
Gambar 6. Nilai *loss* Acuan

Berdasarkan tabel XIII, dapat disimpulkan bahwa pada saat melakukan pendekatan berupa parameter *freeze*, waktu *training* menjadi jauh lebih singkat, hal ini karena *optimizer* tidak perlu menghitung ulang *weight* pada setiap *node* dari BERT. Namun, apabila melihat grafik acuan pada gambar 6, terlihat model acuan mengalami sedikit *overfit*, berbeda dengan saat melakukan pendekatan *dropout* dimana grafik *loss* seperti bisa dilihat pada gambar 8, masih menunjukkan grafik yang bagus dan model tidak *overfit* dan memiliki tingkat akurasi yang tidak terpaut jauh dibanding model acuan.

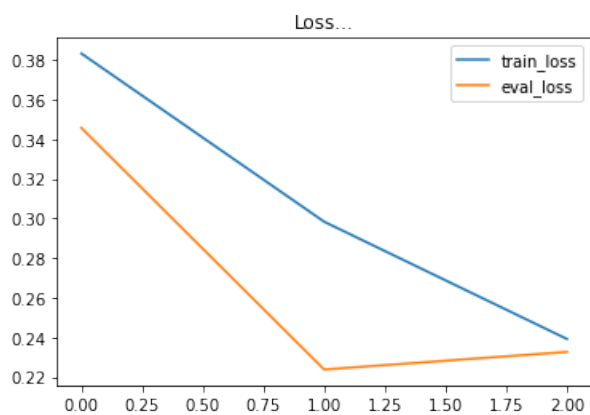
IV. KESIMPULAN DAN SARAN

Dari keseluruhan penelitian yang dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut :

- 1) Semakin banyak dataset yang digunakan pada waktu *pretrain* BERT, maka semakin akurat juga suatu model



Gambar 7. Nilai *loss* saat melakukan *training* dengan parameter *freeze*



Gambar 8. nilai *loss* saat melakukan *training* dengan merubah *dropout*

tersebut, hal ini terbukti oleh model *indobert-base-pl* 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.

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.

PUSTAKA

- [1] Rani and N. L. R. Maha, "Persepsi jurnalis dan praktisi humas terhadap nilai berita," 2013.
- [2] Wikipedia, "Berita bohong," diakses 27 November 2020. [Online]. Available: https://id.wikipedia.org/wiki/Berita_bohong
- [3] K. Kominfo, "Temuan isu hoaks," 03 2020. [Online]. Available: https://eppid.kominfo.go.id/storage/uploads/2_12_Data_Statistik_Hoax_Agustus_2018_-_31_Maret_2020.pdf
- [4] —, "Laporan isu hoax juni 2020," 07 2020. [Online]. Available: https://eppid.kominfo.go.id/storage/uploads/2_31_Laporan_Isu_Hoaks_Bulan_Juni_2020.pdf
- [5] M. L. Khan and I. Idris, "Recognize misinformation and verify before sharing: A reasoned action and information literacy perspective," *Behaviour and Information Technology*, 01 2019.
- [6] K. Wibowo, D. Rahmawan, and E. Maryani, "Penelitian di indonesia: umur tidak mempengaruhi kecenderungan orang menyebarkan hoaks," 2019, diakses 27 November 2020. [Online]. Available: <https://theconversation.com/penelitian-di-indonesia-umur-tidak-mempengaruhi-kecenderungan-orang-menyebarkan-hoaks-110621>
- [7] A. Aggarwal, A. Chauhan, D. Kumar, M. Mittal, and S. Verma, "Classification of fake news by fine-tuning deep bidirectional transformers based language model," p. 163973, 04 2020.
- [8] P. Bahad, P. Saxena, and R. Kamal, "Fake news detection using bi-directional lstm-recurrent neural network," *Procedia Computer Science*, vol. 165, pp. 74–82, 02 2020.
- [9] A. B. Prasetyo, R. R. Isnanto, D. Eridani, Y. A. A. Soetrisno, M. Arfan, and A. Sofwan, "Hoax detection system on indonesian news sites based on text classification using svm and sgd," in *2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 10 2017.
- [10] F. Rahutomo, I. Pratiwi, and D. Ramadhani, "Eksperimen naïve bayes pada deteksi berita hoax berbahasa indonesia," *JURNAL PENELITIAN KOMUNIKASI DAN OPINI PUBLIK*, vol. 23, 07 2019.
- [11] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp," *arXiv preprint arXiv:2011.00677*, 2020.
- [12] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *China National Conference on Chinese Computational Linguistics*. Springer, 2019, pp. 194–206.
- [13] Z. Husein, "Natural-language-toolkit library for bahasa malaysia, powered by deep learning tensorflow," <https://github.com/huseinzol05/malaya>, 2018.
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.
- [15] B. Willie, K. Vincentio, G. I. Winata, S. Cahyawijaya, X. Li, Z. Y. Lim, S. Soleman, R. Mahendra, P. Fung, S. Bahar, and A. Purwarianti,

“Indonlu: Benchmark and resources for evaluating indonesian natural language understanding,” in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized BERT pretraining approach,” *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: <http://arxiv.org/abs/1907.11692>