

**DETEKSI BERITA *HOAX* DENGAN PERBANDINGAN WEBSITE
MENGUNAKAN PENDEKATAN *DEEP LEARNING*
ALGORITMA BERT**

SKRIPSI



Oleh :
ASEP RIPA'I
2020503011

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO**

2024

HALAMAN JUDUL

DETEKSI BERITA *HOAX* DENGAN PERBANDINGAN WEBSITE MENGUNAKAN PENDEKATAN *DEEP LEARNING* ALGORITMA BERT

SKRIPSI

Diajukan untuk Memenuhi Salah Satu Persyaratan dalam Menyelesaikan Program Sarjana (S-1) pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Ibrahimy



Oleh :

ASEP RIPA'I
2020.503.011

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO**

2024

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : **Asep Ripa'i**
 NPM/NIRM : 2020503011
 Program Studi : S-1 Teknologi Informasi
 Fakultas : Fakultas Sains dan Teknologi

Menyatakan dengan sebenarnya, bahwa tugas akhir/skripsi ini secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali pada bagian-bagian yang dirujuk sebagai sumber referensi dan disebut dalam daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa tugas akhir/skripsi ini hasil plagiasi, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Situbondo, 01 September 2024

Saya Yang Menyatakan,



Asep Ripa'i

PERSETUJUAN PEMBIMBING

Nama : **ASEP RIPA'I**

NPM : **2020503011**

Judul : **DETEKSI BERITA *HOAX* DENGAN PERBANDINGAN
WEBSITE MENGGUNAKAN PENDEKATAN *DEEP
LEARNING* ALGORITMA BERT**

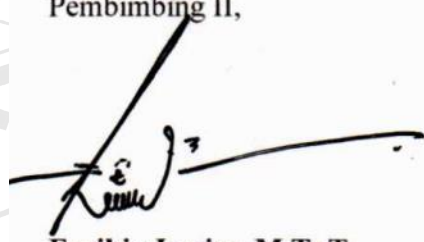
Telah disetujui oleh :

Pembimbing I,



Firman Santos, M.Kom
NIDN : 0722129201

Pembimbing II,



Farihin Lazim, M.Tr.T
NIDN : 0711099201

PENGESAHAN

SKRIPSI

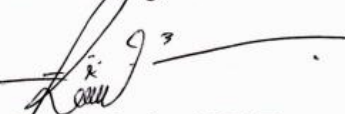
**DETEKSI BERITA *HOAX* DENGAN PERBANDINGAN WEBSITE
MENGUNAKAN PENDEKATAN *DEEP LEARNING*
ALGORITMA BERT**

ASEP RIPAI
2020503011

Telah dipertahankan di depan dewan penguji Sidang/Munaqasyah Skripsi pada hari Rabu Tanggal 07 Agustus 2024 sebagai salah satu syarat memperoleh gelar Sarjana (S.Kom) pada Fakultas Sains dan Teknologi Universitas Ibrahimiy.

Tim Penguji,

Ketua Sidang,


Parihini Lazim, M.Tr.T
NIDN: 0711099201

Sekretaris Sidang,


Abdul Hamid, S.Pi

Penguji I,


Abd. Ghofur, M.Kom
NIDN: 0711088303

Penguji II,

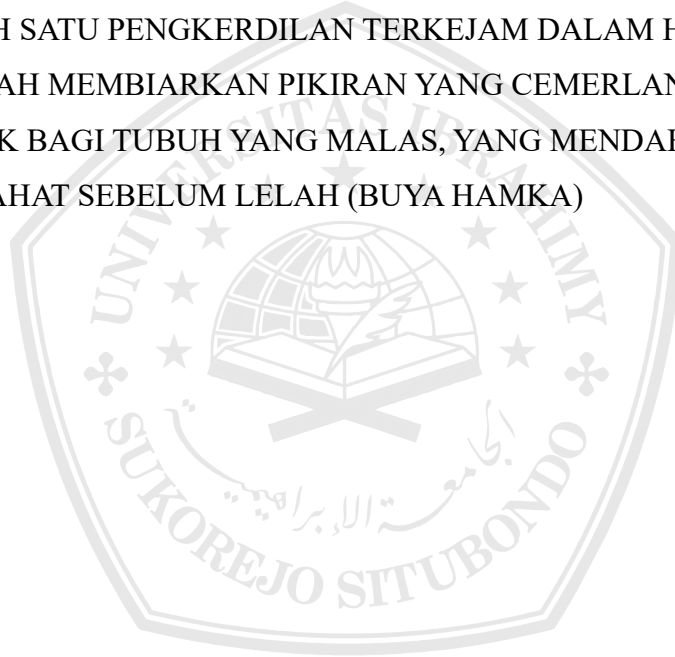

Adi Susanto, M.Kom
NIDN: 0708079104

Mengetahui
Dekan,


Abd. Ghofur, M.Kom
NIDN : 0711088303

MOTO

- ❖ KARENA SESUNGGUHNYA SESUDAH KESULITAN ITU ADA KEMUDAHAN. MAKA SESUNGGUHNYA KAMU TELAH SELESAI (DARI SESUATU URUSAN) KERJAKANLAH DENGAN SUNGGUH - SUNGGUH (URUSAN) YANG LAIN (QS. AL-INSYIRAH; 5-7)
- ❖ TERUSLAH BERJUANG SAMPAI LELAHMU LELAH MENGIKUTIMU
- ❖ SALAH SATU PENGKERDILAN TERKEJAM DALAM HIDUP ADALAH MEMBIARKAN PIKIRAN YANG CEMERLANG MENJADI BUDAK BAGI TUBUH YANG MALAS, YANG MENDAHULUKAN ISTIRAHAT SEBELUM LELAH (BUYA HAMKA)



PERSEMBAHAN

- ❖ Untuk sang kesatria yang gagah berani yang rela bekerja keras demi membuat kami bahagia yakni (Alm) Bapak Idi yang tercinta sebagai sumber inspirasiku yang kini telah berada dipangkuan sang ilahi rabbi. Semoga beliau senantiasa di limpahkan rahmat dan ampunan Allah Swt, dan kepada sang malaikat tak bersayap Ibu Nunung yang tercinta selalu melimpahkan kasih sayang, nasihat dan motivasi, semoga senantiasa dilimpahkan rezeki dan Kesehatan.
- ❖ Kepada kakak-kakakku Eulis Sa'adah, Umul Hasanah dan adikku Latifatulsupi dan juga tak lupa kepada keponakanku Ahmad Rofi'i, Azkiya Salsa Humaira, Elga Sahirameka, Wirda Uswatun Afifah dan seluruh keluarga besarku yang telah selalu memberika doa dan motivasi selama ini
- ❖ Dan Kepada Sahabat-sahabat terbaikku yang tak henti-hentinya memberikan support kepadaku
- ❖ Dan juga kepada guru-guruku yang telah mendidik dan membimbingku menjadikan orang yang bermanfaat.

KATA PENGANTAR

Segala puji syukur peneliti sampaikan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya, perencanaan, pelaksanaan dan penyelesaian tugas akhir/skripsi dengan judul “ Deteksi Berita *Hoax* Dengan Perbandingan Website Menggunakan *Deep Learning* Algoritma BERT ” sebagai salah satu syarat penyelesaian program diploma/sarjana dapat terselesaikan dengan baik dan lancar.

Kesuksesan ini dapat peneliti peroleh karena dukungan beberapa pihak. Peneliti menyampaikan terima kasih kepada :

1. KHR. Ach. Azaim Ibrahimy, S.Sy., M.H.I selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo.
2. KH. Ach Fadlail, S.H., M.H.I selaku Rektor Universitas Ibrahimy.
3. Firman Santoso, M.Kom selaku Ketua Program Studi Teknologi Informasi dan selaku Pembimbing I.
4. Farihin Lazim M.Tr.T selaku Pembimbing II.
5. Seluruh Dosen Fakultas Sains dan Teknologi yang telah memberikan kami ilmu sehingga sampai pada masa Tugas Akhir (Skripsi) saat ini.

Semoga semua amal baik yang telah diberikan oleh Bapak/Ibu kepada peneliti mendapat balasan yang sebaik mungkin dari Allah SWT, Aamiin.

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TULISAN.....	ii
PERSETUJUAN PEMBIMBING.....	iii
PENGESAHAN	iv
MOTO	v
PERSEMBAHAN	vi
KATA PENGNTAR.....	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
SEGMENT PROGRAM	xiii
ABSTRAK	xiv
<i>ABSTRACT</i>	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	4
1.3 Rumusan Masalah	5
1.4 Batasan Masalah.....	5
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian.....	6
1.7 Metode Penelitian.....	7
1.7.1 Jenis Penelitian	7
1.7.2 Teknik Pengumpulan Data	7
1.7.3 Metode Pengembangan Sistem.....	8
1.8 Sistematika Penulisan.....	10
BAB II TINJAUAN PUSTAKA	12

2.1 Penelitian Terdahulu	12
2.2 Landasan Teori	15
2.3 Perangkat Lunak Yang digunakan	24
2.4 Pemodelan	27
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	29
3.1 Kerangka Berpikir	29
3.2 Pengumpulan Data	29
3.2.1 Sumber Data	29
3.3 Metode Pengembangan Model <i>Machine Learning</i>	30
3.3.1 <i>Text Preprocessing</i>	30
3.3.2 <i>Feature Engineering</i>	32
3.4 Metode Pengembangan Sistem.....	36
3.4.1 <i>Planning</i>	36
3.4.2 <i>Analysis</i>	36
3.4.3 <i>Design</i>	37
3.4.4 <i>Implementation</i>	40
3.4.5 <i>Testing</i>	40
3.4.6 <i>Maintenance</i>	42
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....	43
4.1 Skenario Pengujian <i>Machine Learning</i>	43
4.1.1 <i>Load Datasets</i>	43
4.1.2 <i>Modelling</i>	44
4.1.3 Komparasi Model	54
4.2 Analisis Teks.....	55
4.3 Skenario Pengujian Sistem.....	56

4.4 Segmen Program	57
4.4.1 Segmen Program <i>Text Preprocessing</i>	57
4.4.2 Segmen Program <i>Feature Engineering</i>	59
4.4.3 Segmen Program <i>Training dan Evaluation</i>	60
4.4.4 Segmen Program Implementasi Model	63
BAB V PENUTUP	65
5.1 Kesimpulan.....	65
5.2 Saran.....	65
DAFTAR PUSTAKA.....	66
LAMPIRAN-LAMPIRAN.....	70
A. Hasil Implementasi Model	70
B. Hasil Prediksi Data Testing	71
C. Sumber Data.....	71
CURRICULUM VITAE	72

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur BERT	19
Gambar 2. 2 Ilustrasi Algoritma <i>Random Forest</i>	21
Gambar 2. 3 <i>Hyperlane</i> SVM	21
Gambar 2. 4 Simbol-Simbol <i>Flowchart</i>	27
Gambar 2. 5 Komponen <i>Data Flow Diagram</i>	28
Gambar 2. 4 Komponen <i>Data Flow Diagram</i>	28
Gambar 3. 1 Kerangka Berpikir	29
Gambar 3. 2 Persebaran Dataset (a) Sebelum <i>Balancing</i> (b) Sesudah <i>Balancing</i>	31
Gambar 3. 3 <i>Feature Engineering</i> BERT	32
Gambar 3. 4 Ilustrasi <i>Embedding</i> BERT	34
Gambar 3. 5 Contoh Hasil <i>Embedding</i> BERT	36
Gambar 3. 6 <i>Flowchart</i> Sistem	37
Gambar 3. 7 <i>Context Diagram</i> Sistem	38
Gambar 3. 8 <i>Data Flow Diagram</i> Level 0	39
Gambar 3. 9 Desain <i>Interface input</i>	39
Gambar 3. 10 Desain <i>Interface Output</i>	40
Gambar 3. 11 <i>Flowchart</i> Pengujian Sistem	41
Gambar 4. 1 Hasil <i>Load Datasets</i>	43
Gambar 4. 2 Hasil <i>Finetunning</i> BERT	46
Gambar 4. 3 Hasil <i>Confusion Matrix</i> BERT	47
Gambar 4. 4 Hasil <i>Training Model Random Forest</i>	49
Gambar 4. 5 Hasil <i>Confusion Matrix Random Forest</i>	50
Gambar 4. 6 Hasil <i>Confusion Matrix</i> SVM	53
Gambar 4. 7 Perbandingan Akurasi Semua Model	54
Gambar 4. 8 Hasil Perbandingan Kurva ROC-AUC	55
Gambar 4. 9 (a) <i>Word Cloud</i> Berita <i>Hoax</i> , (b) <i>Word Link</i> Berita <i>Hoax</i>	56

DAFTAR TABEL

Tabel 3. 1 Kumpulan Dataset Berita yang di Peroleh.....	30
Tabel 3. 2 Contoh Hasil <i>Tokenizing</i> BERT	33
Tabel 4. 1 <i>Hyperparameter</i> Model BERT.....	45
Tabel 4. 3 Parameter <i>Random Forest</i> yang di Gunakan	48
Tabel 4. 4 Parameter SVM yang di Gunakan.....	51
Tabel 4. 5 Hasil Pengujian Sistem.....	56



SEGMENT PROGRAM

Segmen Program 4. 1 <i>Text Peprocessing</i>	57
Segmen Program 4. 3 <i>Feature Engineering</i>	59
Segmen Program 4. 4 <i>Training dan Evaluation</i>	60
Segmen Program 4. 5 Implementasi Model	63



ABSTRAK

Asep Ripa'i. 2024. Deteksi Berita Hoax Dengan Perbandingan Website Menggunakan Pendekatan *Deep Learning*. Skripsi, Program Studi Teknologi Informasi, Universitas Ibrahmy. Pembimbing (I) Firman Santoso, M.Kom., (II) Farihin Lazim, M.Tr.T .

Berita *hoax* merupakan informasi yang salah dan menyesatkan yang dapat menyebabkan provokasi dan kebencian bagi pembaca. Dengan adanya akses internet yang mudah membuat persebaran berita *hoax* semakin masif. Oleh karena itu, perlu ada metode yang dapat melakukan deteksi berita *hoax*. Penelitian menggunakan metode *deep learning* dengan mengintegrasikan *text mining* untuk mencari informasi dan pola berita yang berhubungan dengan *hoax*. Dengan menggunakan dataset dari situs kaggle berjumlah sekitar 2700-an kemudian dilakukan *text preprocessing* agar data lebih terstruktur untuk diolah lanjut. Kemudian membuat *feature engineering* dari BERT agar data dapat diproses oleh *machine learning* dengan tiga metode klasifikasi yaitu BERT, SVM dan *random forest* kemudian dilakukan pengujian dan evaluasi. Pada penelitian ini model yang menghasilkan performa yang paling tinggi yaitu BERT dengan (akurasi = 0.99, ROC-AUC = 0.99) dibanding model *machine learning* tradisional.

Kata Kunci : Berita Hoax, *Text Mining*, *Deep Learning*, Algoritma BERT

ABSTRACT

Asep Ripa'i. 2024. *Hoax News Detection with Website Comparison Using Deep Learning Approach BERT Algorithm*. Thesis, Information Technology Study Program, Ibrahimi University. Supervisor (I) Firman Santoso, M.Kom., (II) Farihin lazim, M.Tr.T.

Hoax news is false and misleading information that can cause provocation and hatred for readers. With easy internet access, the spread of hoax news is getting more massive. Therefore, there needs to be a method that can detect hoax news. The research uses deep learning methods by integrating text mining to find information and news patterns related to hoaxes. By using a dataset from the kaggle site totaling around 27000 then text preprocessing is carried out so that the data is more structured for further processing. Then make feature engineering from BERT so that the data can be processed by machine learning with three classification methods namely BERT, SVM and random forest then testing and evaluation. In this study, the model that produces the highest performance is BERT with (accuracy = 0.99, ROC-AUC = 0.99) compared to traditional machine learning models.

Keyword : Hoax News, Text Mining, Deep Learning, BERT Algorithm

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah permasalahan di era digital ini adalah berita *hoax* (bohong) yang mudah tersebar di internet dan menjadi tidak terkontrol. Berita *hoax* atau berita palsu merupakan Informasi yang menyesatkan dan berbahaya karena dapat menyesatkan persepsi masyarakat dengan menyajikan informasi palsu sebagai kebenaran [1] dan menimbulkan provokasi dengan membuat pembaca terpengaruh akan informasi palsu sehingga membuat pembaca mengambil tindakan sesuai isi *hoax*.

menurut Selamatta berita *hoax* dapat dikenali dengan isi konten yang dapat menciptakan kecemasan, kebencian dan permusuhan, memberi penjurukan, judul dan pengantar provokatif dan tidak cocok dengan isinya, sumber media berita tidak jelas atau abal-abal cenderung menyudutkan pihak tertentu [2]. Menurut catatan Dewan Pers, di Indonesia terdapat sekitar 61.800 situs di Indonesia yang mengklaim sebagai portal berita. Dari jumlah tersebut, yang sudah terverifikasi sebagai situs berita resmi sekitar 1700-an [3]. Artinya terdapat setidaknya puluhan ribu situs yang berpotensi menyebarkan berita palsu di internet yang mesti diwaspadai.

Melalui teknologi *machine learning* merupakan suatu bidang keilmuan dalam *computer science* yang membuat suatu komputer

atau mesin dapat belajar seperti manusia agar mempunyai suatu kecerdasan dengan pembelajaran yang berasal dari data dan menghasilkan prediksi dimasa mendatang [4]. Dengan mengintegrasikan dengan *text mining* membuktikan kemampuannya dalam mengidentifikasi, mengekstrak, dan menganalisis informasi semantik dari berbagai sumber teks dengan tingkat kualitas yang lebih tinggi.

Melalui penggunaan algoritma *machine learning* dan metode analisis bahasa alami tingkat lanjut, *text mining* mampu melakukannya secara otomatis mengevaluasi pola dan tren yang muncul dalam teks yang dianalisis, termasuk literatur, dokumen, dan data teks lainnya. Dengan demikian, teknologi ini memungkinkan pengguna untuk memperoleh pemahaman tentang konten yang lebih mendalam dan komprehensif dari sekadar menganalisis jumlah kata dasar [5]. yang dapat diarahkan untuk menemukan suatu pola-pola dan ciri kusus konten hoax berdasarkan pemahaman kontek sosial dan historis pada suatu berita.

Untuk mendapatkan hasil yang optimal dalam *text mining* untuk mengidentifikasi berita *hoax* berdasarkan perbandingan website peneliti menggunakan metode *deep learning* yang berbasis *Natural Language Processing* (NLP) yang dapat mengolah bahasa manusia agar bisa dimengerti oleh komputer. *Deep learning* merupakan bagian dari *machine learning* yang berbasis jaringan saraf tiruan,

yang terinspirasi dari sistem otak manusia, sehingga dapat lebih optimal untuk menganalisis teks karena memiliki representasi yang lebih kompleks dari pada *machine learning* biasa.

Salah satu algoritma *deep learning* yang berbasis NLP adalah algoritma BERT (*Bidirectional Encoder Representations from Transformers*) merupakan algoritma yang yang diluncurkan google pada tahun 2018 oleh Jacob Devlin dan rekan-rekannya dari *Google Research*. BERT merupakan algoritma berdasarkan *transformer* pra-pelatihan yang menerapkan pelatihan dua arah kemudian menggabungkan konteks dari lapisan kiri dan kanan yang telah terbukti efektif meningkatkan banyak bahasa dan dapat memprediksi hubungan antar kalimat dengan menganalisisnya secara holistik [6].

Dalam penelitian [7] mengusulkan menggunakan metode *deep learning* yang berbasis BERT (*Bidirectional Encoder Representations from Transformers*) yang mampu melatih representasi dua arah teks sehingga dapat meningkatkan kinerja klasifikasi dengan kemampuan membedakan konteks dari kalimat lainnya. Hasil dari model BERT yang diusulkan dapat mengungguli dengan akurasi 98,90 % dari model dengan dengan algoritma *machine learning* tradisional seperti *decision tree*, *random forest*, KNN, *multinomial Naive Bayes* dan *deep learning* yang berbasis jaringan saraf seperti CNN, LSTM, dan *neural network*.

Dalam penelitian [8] melakukan percobaan dengan tiga sumber data dengan total data mencapai 2.216 dengan sebaran 1.055 data valid dan 1.161 data palsu. Perbandingan model BERT berbahasa Indonesia dengan algoritma *deep learning* seperti CNN, BiLST, dan Hybrid CNN-BiLSTM, dengan akurasi BERT 90%, CNN 74%, BiLSTM 85%, Hibrida CNN-BiLSTM 74%. Dari hasil penelitian yang telah dilakukan terlihat bahwa secara umum arsitektur BERT yang digunakan menghasilkan *classifer* yang lebih baik dibandingkan *classifer* lainnya.

Berdasarkan pemaparan diatas maka peneliti tertarik untuk membahas topik penelitian tentang “Deteksi Berita *Hoax* Dengan Perbandingan Website Menggunakan Pendekatan *Deep Learning* Algoritma BERT”, tujuan dari penelitian ini adalah untuk mengidentifikasi dan mencari pola berita berdasarkan perbandingan website dan mencari model terbaik dengan *feature engineering* dari BERT kemudian akan diproses klasifikasi dengan tiga metode yaitu BERT, SVM dan *Random Forest* untuk mengetahui tingkat performanya.

1.2 Identifikasi Masalah

Dari uraian latar belakang diatas dapat diidentifikasi beberapa masalah, berikut merupakan masalah yang telah teridentifikasi oleh peneliti:

- a. Sekitar 61.800 situs berita online hanya 1700-an yang telah terverifikasi berita resmi, berarti ada puluhan ribu situs yang berpotensi menyebarkan *hoax*.
- b. Menguji keefektifan *deep learning* pada *feature engineering* BERT dalam *text mining* untuk mengidentifikasi berita *hoax* dengan metode klasifikasi BERT, SVM dan *Random Forest*.

1.3 Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah, maka dapat dirumuskan sebagai berikut:

- a. Bagaimana membuat model *machine learning* menggunakan *feature engineering* BERT yang dapat membantu mengidentifikasi berita *hoax*?
- b. Bagaimana tingkat akurasi dan keefektifan *feature engineering* BERT dalam *text mining* untuk mengidentifikasi dengan metode klasifikasi BERT, SVM dan *Random Forest*?

1.4 Batasan Masalah

Agar pembahasan lebih terarah dan tidak melebar, maka dalam penelitian ini dibuat batasan masalah sebagai berikut:

- a. Penelitian ini menggunakan dataset dari Kaggle yang terdiri dari CNN, Kompas, Tempo untuk berita valid dan TurnBackHoax berisi berita *hoax*.
- b. Pada penelitian ini melakukan perbandingan website dari situs resmi CNN, Kompas, dan Tempo untuk berita valid dan TurnBackhoax sebagai data berita *hoax*.

- c. Untuk membuat model dengan algoritma BERT peneliti memakai model IndoBERT yang berbasis bahasa Indonesia.

1.5 Tujuan Penelitian

Dari rumusan masalah diatas maka, tujuan penelitian ini adalah:

- a. Implementasi *feature engineering* BERT dalam mengidentifikasi berita *hoax*.
- b. Mencari model terbaik dan menguji performa metode BERT, SVM dan *Random Forest* dengan *feature engineering* dari BERT (*Bidirectional Encoder Representations From Transformers*) dalam mengklasifikasi berita *hoax*.

1.6 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

- a. Bagi peneliti
Mengetahui implementasi *feature engineering* dari BERT (*Bidirectional Encoder From Representations Transformers*) dengan metode BERT, SVM dan *Random Forest* dalam mengidentifikasi berita *hoax*.

- b. Bagi Masyarakat umum dan peneliti yang lain

Semoga dengan adanya sistem cerdas dengan model BERT ini dapat membantu masyarakat lebih cermat dalam menanggapi berita politik dan untuk penelitian lanjutan bisa menjadi referensi dan dapat mengembangkan dengan metode lain misal dengan *OpenAI Generative*

Pre-trained Transformer (GPT), Embedding From Language Models (ELMO).

1.7 Metode Penelitian

1.7.1 Jenis Penelitian

a. Eksperimen Studi Kasus

Eksperimen merupakan metode penelitian ilmiah yang digunakan untuk menguji hipotesis atau mengidentifikasi hubungan sebab-akibat melalui pengumpulan data empiris [9]. Dalam hal ini peneliti akan menguji hasil dari algoritma tersebut dari data berita yang telah terverifikasi.

Studi kasus merupakan pendekatan yang digunakan untuk mengungkap karakteristik dan keunikan dari suatu peristiwa atau fenomena [10]. Dalam hal ini peneliti akan mencari pola kata kunci sering muncul dalam berita *hoax*.

1.7.2 Teknik Pengumpulan Data

a. Pengumpulan Data dan Analisis

Peneliti melakukan observasi terdahulu mengenai website yang akan digunakan untuk diambil data beritanya. Dalam penelitian ini memakai data dari kaggle yang terdiri dari berita valid yaitu Kompas sebanyak 4000, CNN 10.000, dan Tempo 6000 dan berita *hoax* Tumbackhoax 10.000 data, dengan total 20.000 data valid dan 10.000 data *hoax*.

b. Studi Pustaka

Untuk melengkapi data peneliti menambahkan metode studi pustaka yaitu dengan membaca karya tulis ilmiah yang berkaitan dengan teori, buku, jurnal dan skripsi penelitian terdahulu yang memiliki korelasi dengan topik Deteksi Berita *Hoax* Dengan Perbandingan Website Menggunakan Pendekatan *Deep Learning* Algoritma BERT.

1.7.3 Metode Pengembangan Sistem

Peneliti menggunakan *System Development Life Cycle* (SDLC) sebagai metode pengembangan sistem. SDLC adalah metodologi yang umum digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Dengan metode ini pengembangan sistem akan lebih cepat dicapai dengan peningkatan siklus hidup dan penggunaan peralatan pengembangan berbasis komputer, berikut tahapan-tahapan *System Development Life Cycle* [11]:

a. *Planning*

Planning merupakan tahap awal dalam pengembangan sistem. Pada tahap ini akan dilakukan indentifikasi sistem yang akan dibangun, berupa target pengguna, analisis kebutuhan bisnis dan rencana sumber daya proyek.

b. *Analysis*

Pada tahap ini dilakukan dilakukan studi kelayakan sistem yang bisa dipakai oleh khalayak umum dengan observasi melalui berbagai sumber, menganalisis fungsi dan fitur sesuai kebutuhan pengguna.

c. *Design*

Setelah selesai melakukan tahap analisis selanjutnya membuat *design* sistem berupa gambaran umum rancangan *blue print*. Dengan membuat pemodelan sistem, desain aplikasi berupa *mockup*, membuat deskripsi kebutuhan sistem dan pengguna dan membuat dokumentasi untuk panduan.

d. *Implementation*

Pada tahap ini sistem yang telah di rancang dan model BERT yang telah selesai dibuat akan di *convert* kebentuk pickle untuk implementasikan dalam aplikasi web yang bisa diakses melalui berbagai perangkat. Dalam hal ini peneliti menggunakan framework flask yang berbasis bahasa pemrograman python.

e. *Testing*

Untuk mendapatkan hasil yang optimal sistem yang telah dibuat agar sesuai dengan kebutuhan pengguna harus dilakukan testing dengan menguji setiap komponen dan fungsi-fungsi sistem dengan menggunakan metode *black box*.

f. *Maintenance*

Tahap ini dilakukan apabila di kemudian hari ada perbaikan atau pemeliharaan, sehingga sistem bisa tetap update dan berjalan sesuai kebutuhan pengguna.

1.8 Sistematika Penulisan

Pada penyusunan skripsi ini penulis menggunakan sistematika penulisan sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan, manfaat penulisan, metode penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA DAN KAJIAN TEORI

Pada bab ini akan membahas kajian-kajian terdahulu dan teori-teori yang relevan dengan topik penelitian yang berkaitan dan mendukung penelitian ini.

BAB III METODOLOGI PENELITIAN

Pada bab ini membahas langkah-langkah atau metode-metode yang digunakan dalam menyelesaikan masalah untuk mencapai tujuan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang hasil dari pembahasan setelah menyelesaikan permasalahan pada penelitian ini.

BAB V

PENUTUP

Pada bab ini berisi tentang kesimpulan dan saran dari hasil laporan penelitian.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

- a. “*FakeBERT: Fake news detection in social media with a BERT based deep learning approach*” [7]

Di era digital ini, ekosistem berita telah berubah dari media cetak tradisional menjadi media sosial. Platform media sosial dapat memberikan kemudahan dalam memberikan informasi yang cepat dan fitur berbagi tanpa batas, dari manfaat tersebut muncul juga dampak negatif yaitu persebaran berita *hoax* juga semakin banyak dan cepat.

Maka untuk mengatasi permasalahan tersebut peneliti membuat sebuah model FakeBERT yang berbasis algoritma BERT dengan menggabungkan beberapa kombinasi yaitu, *Convolutional layer* menggunakan tiga blok paralel ID-CNN, *Max-pooling layer* menggunakan lima lapisan *max-pooling* (tiga menggunakan blok paralel ID-CNN dan dua dengan lapisan konvolusional lurus ke depan) dengan mendapatkan akurasi 98,90% dari metode CNN dengan akurasi 91.50% dan LSTM 97.25%.

- b. “*Indonesia's Fake News Detection using Transformer Network*” [8]

Berita *hoax* atau palsu menjadi suatu permasalahan di era digital ini bagi Masyarakat yang bisa menimbulkan provokasi dan masalah. Indonesia adalah negara terbesar ke-4 lebih dari 30% penduduk desa dan perkotaan tertipu oleh masalah berita *hoax*. Dari permasalahan

tersebut penelitian ini dilakukan sebagai upaya untuk mencegah permasalahan diatas.

Peneliti disini menggunakan dataset dari portal berita mengidentifikasi berita palsu dari situs turnbackhoax.id dengan menggunakan *web scraping* untuk memperoleh sebanyak 1116 data yang terdiri dari berita valid dan berita palsu kemudian ditambah

dengan dataset lain yang pertama dari mendeley sebanyak 600 data, berisi 372 berita valid dan 228 berita bohong. dengan total data mencapai 2.216 dengan sebaran 1.055 data valid dan 1.161 data palsu. Dataset tersebut kemudian dibagi menjadi data dengan perbandingan 80% : 20% untuk data latih dan data uji secara berurutan. Sehingga total data latih yang digunakan sebanyak 1772 dan 444 data uji.

Dalam Penelitian ini menggunakan algoritma BERT sebagai baseline dengan memakai jenis model BERT yang ada yaitu BERT *Multilingual-Cased* dari huggingface. penelitian kemudian melakukan perbandingan dengan algoritma *deep learning* lainnya seperti CNN, BiLSTM, dan Hybrid CNN-BiLSTM. Hasil dari penelitian ini menunjukkan algoritma BERT memiliki akurasi tinggi sebesar 90% daripada algoritma CNN sebesar 74%, BiLSTM 86%, Hybrid CNNBiLSTM sebesar 74%.

“Dalam Penelitian ini menggunakan algoritma BERT sebagai baseline dengan memakai jenis model BERT yang ada yaitu BERT *Multilingual-Cased* dari huggingface. penelitian kemudian melakukan

perbandingan dengan algoritma deep learning lainnya seperti CNN, BiLSTM, dan Hybrid CNN-BiLSTM. Hasil dari penelitian ini menunjukkan algoritma BERT memiliki akurasi tinggi sebesar 90% dari pada algoritma CNN sebesar 74%, BiLSTM 86%, Hybrid CNNBiLSTM sebesar 74%.

c. *Machine Learning Untuk Deteksi Berita Hoax Menggunakan BERT [12]*”

Di zaman era digital ini penyebaran informasi yang mudah tersebar, berita *hoax* telah menjadi permasalahan yang serius karena jumlahnya yang semakin meningkat. Penelitian ini dibuat untuk mengatasi tantangan tersebut dengan menggunakan pendekatan algoritma BERT (*Bidirectional Encoder Representations from Transformers*) dan *Random Forest Classifier* dalam memodelkan dan mengklasifikasikan berita *hoax*).

Metode BERT digunakan untuk menghasilkan representasi yang kaya dari kalimat tersebut, sedangkan *Random Forest Classifier* digunakan sebagai pengklasifikasi yang efektif dan mampu mengatasi masalah *overfitting* pada data teks. Menggunakan *Google Collaboratory* sebagai alat eksplorasi kumpulan data, hasil evaluasi menunjukkan bahwa model klasifikasi BERT memiliki akurasi 67% pada data asli dalam mengklasifikasikan informasi phishing, yang menunjukkan kinerja model pembelajaran mesin dalam mengklasifikasikan informasi palsu. F1 - Skor untuk label positif adalah

0,67 dengan presisi 0,57 untuk label positif, *recall* untuk label positif adalah 0,80. Dapat disimpulkan bahwa model BERT memiliki kemampuan pemahaman dan analisis teks yang baik sehingga cocok untuk tugas klasifikasi *hoax*.

2.2 Landasan Teori

a. *Hoax*

Kata *hoax* dalam KBBI merupakan informasi yang sengaja direayasa untuk menutupi informasi sebenarnya. *Hoax* juga bermakna sebuah kebohongan atau informasi yang menyesatkan yang sengaja disamarkan agar terlihat benar [13]. *Hoax* juga dapat difahami juga sebagai cara untuk pemutarbalikan fakta menggunakan informasi yang seolah-olah meyakinkan tetapi tidak dapat diverifikasi kebenarannya [14].

Dalam konteks yang lebih luas, *hoax* bukan hanya sebatas berita palsu saja tapi mencakup suatu perbuatan yang bertujuan untuk menipu/membohongi orang lain dengan cara memanipulasi, mengakali, ataupun menutupi fakta yang sebenarnya [13].

b. Berita *Hoax*

Berita merupakan informasi yang melaporkan tentang peristiwa dan pendapat yang aktual, penting dan menarik yang disampaikan kepada publik lewat media internet, koran, radio, layar kaca atau media lainnya. Sebuah berita harus memenuhi unsur-unsur berikut, yaitu : 5W+1H (*what, who, when, where, why* dan *How*) [15].

Berita disebut *hoax* adalah berita kebohongan, fitnah dan tidak memiliki pola yang dapat di indentifikasi yang terlihat seperti berita faktual yang sengaja dibuat untuk menyebarkan propaganda fitnah atau pesan kebencian kepada seseorang atau intansi tertentu [13].

c. *Deep Learning*

Deep Learning merupakan salah satu dari bagian machine learning yang berbasis *artificial neural network* (ANN) yang terinspirasi dari sistem dan struktur otak manusia, Dimana neuron-neuron saling terkoneksi satu sama lain sehingga membentuk jaringan neuron [16].

Arsitektur *deep learning* memiliki beberapa layer layaknya jaringan neuron, yaitu *input layer*, *hidden layer*, dan *output layer*. Pada *hidden layer* dibuat berlapis-lapis dan saling terhubung dengan layer lainnya untuk menemukan komposisi algoritma yang tepat supaya meminimalisir error [17].

Deep learning memungkinkan pembelajaran otomatis representasi fitur yang kompleks. Selain itu *deep learning* juga dapat meningkatkan dan mengoptimalkan parameternya menggunakan algoritma pembelajaran yang disebut *backpropagation*. *Deep learning* memiliki beberapa algoritma yang umum digunakan seperti *recurrent neural network*, *self-organizing maps*, *convolutional neural network* dan *long short term memory network*. Dengan algoritma tersebut, mereka dapat menghasilkan suatu output yang tepat dan akurat [18].

d. *Machine Learning*

Machine Learning (Pembelajaran Mesin) merupakan suatu bagian dari kecerdasan buatan atau biasa disebut *Artificial Intelligence* (AI) yang dapat otomatis belajar dan meningkatkan kemampuannya berdasarkan pengalaman tanpa diprogram secara eksplisit [19] Dalam *machine learning* ada beberapa teknik pembelajaran untuk melatih mesin, yaitu sebagai berikut :

1) *Supervised Learning*

Supervised Learning adalah salah satu bagian teknik *machine learning* yang pembelajarannya berdasarkan data latih yang telah diberi label dengan tujuan dapat memberikan target *output* dengan membandingkan pembelajaran dimasa lalu.

2) *Unsupervised Learning*

Unsupervised Learning dalam pembelajaran mesin tidak membutuhkan data yang berlabel atau kategori. Teknik ini digunakan untuk mencari pola yang mungkin sebelumnya belum diketahui, teknik ini digunakan untuk *clustering* dan *association rule*.

3) *Semi- Supervised Learning*

Semi-Supervised Learning merupakan gabungan dari teknik *supervised learning* dan *unsupervised learning* yang dimana datanya sebagian memiliki label dan sebagiannya tidak memiliki label.

Teknik ini cocok digunakan untuk data dengan jumlah data yang sangat besar yang diberi label dan tidak diberi label.

4) *Reinforcement Learning*

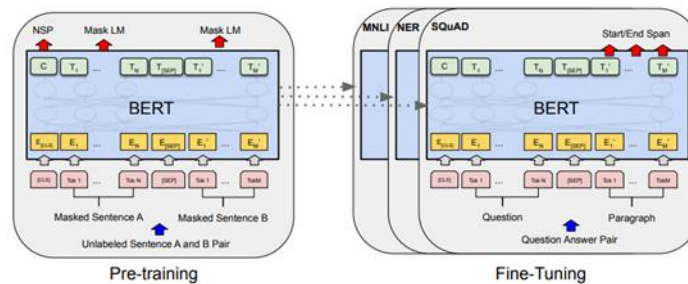
Reinforcement Learning merupakan teknik belajar menggunakan sistem *reward* dan penalti. Yang dapat mempelajari bagaimana membuat keputusan terbaik, secara berurutan dan memaksimalkan sukses kehidupan nyata.

e. BERT (*Bidirectional Encoder Representations From Transformers*)

BERT (*Bidirectional Encoder Representations From Transformers*) adalah satu algoritma *deep learning* yang untuk *pre-training* NLP (*Natural Language Processing*) berdasarkan transformer yang Dimana setiap elemen *input* dan bobot elemen dihitung secara dinamis berdasarkan hubungan antar elemen. BERT dibuat dan usulkan oleh Jacob dan rekan-rekannya dari Google Research pada tahun 2018, Google telah memanfaatkan BERT untuk mesin pencariannya sehingga dapat memahami penelusuran pengguna.

BERT dilatih dengan menjadi model bahasa dengan *encoder* representasi dua arah dari teks tidak berlabel dengan menyesuaikan konteks dari kedua arah setiap kelas kata. BERT dikelompokkan menjadi dua pendekatan pemodelan, yaitu *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Dalam pendekatan MLM, kata-kata yang ditutup dalam korpus dan diberi pelatihan, sedangkan

dalam pendekatan NSP BERT akan memprediksi kalimat selanjutnya yang konteksnya sesuai dengan kalimat sebelumnya [6].



Gambar 2. 1 Arsitektur BERT

(sumber: [6])

f. NLP (*Natural Language Processing*)

NLP (*Natural Language Processing*) merupakan bagian irisan dari *artificial intelligence* (AI), *computer science* dan *computational linguistics* yang bertujuan untuk pemrosesan bahasa alami dengan berfokus pada proses kemampuan komputer untuk menganalisis, memahami dan memperoleh makna dari bahasa manusia dengan cara mirip manusia [16].

Pemanfaatan NLP banyak digunakan pada kehidupan manusia yang sering kita jumpai seperti mesin penerjemah, *spam email filter* yang dapat mengklasifikasi email terdeteksi spam, *search engine* atau mesin telusur salah contohnya adalah *google search* dapat mencari kecocokan kalimat untuk mendapatkan berbagai informasi.

g. *Text mining*

Text mining adalah teknik untuk menemukan informasi baru dan tidak diketahui oleh komputer dengan mengekstraksi data yang tidak

terstruktur berupa teks untuk mencari pola dari kata-kata yang mewakili sebuah dokumen teks agar dapat dianalisa [20]. dokumen teks dapat berupa email, file html, dokument teks lengkap dll.

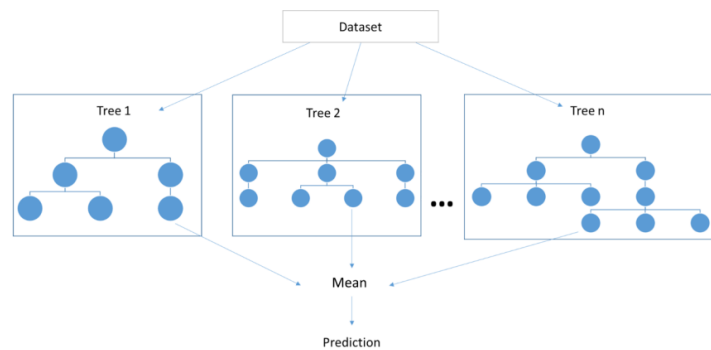
h. *Feature Engineering*

Feature engineering merupakan metode mengekstrasi dan mengonversi data mentah ke dalam format yang dapat digunakan oleh algoritma *machine learning*. Supaya model dapat bekerja lebih baik dan mudah memahami, dengan melakukan pengembangan fitur baru, pemilihan fitur dan modifikasi fitur yang ada [21].

i. *Random Forest*

Random forest merupakan algoritma pengembangan dari *decision tree* dengan berdasarkan konsep pembelajaran *ensemble* dengan teknik *bootstrap aggregating (bagging)* yaitu dengan menggabungkan beberapa model pembelajaran yang independen dalam bentuk pohon keputusan untuk membuat model yang lebih kuat dan keputusan akhir.

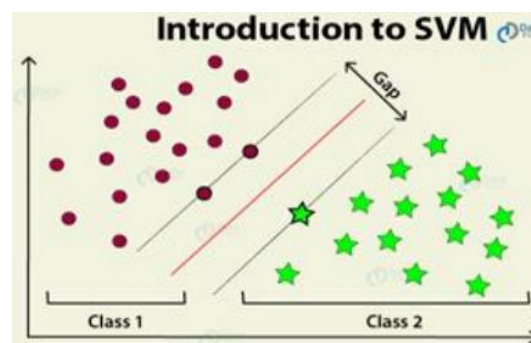
Berdasarkan namanya *random forest* (hutan acak) memiliki banyak pohon keputusan (*decision tree*) dari sekumpulan pohon tersebut terbentuk hutan (*forest*) semakin banyak pohon yang tumbuh maka hasilnya akan akurat dan tidak akan *overfitting*. Kemudian setiap keputusan pohon akan diambil berdasarkan *voting* terbanyak [22]. Dapat dilihat pada gambar 2.2.



Gambar 2. 2 Ilustrasi Algoritma *Random Forest*

j. *Support Vector Machine*

Support vector machine atau SVM merupakan algoritma *machine learning* dengan metode pembelajaran *supervised* terkait menganalisis data yang digunakan untuk klasifikasi dan regresi. SVM bekerja dengan mencari *hyperlane* terbaik yang memisahkan ruang input menjadi dua kelas. *Hyperlane* merupakan sebuah fungsi berdasarkan konsep matematika yang merujuk pada objek geometris dengan dimensi dua *hyperlane* berupa garis lurus, sedangkan dalam ruang tiga dimensi berupa bidang datar [23]. Dapat dilihat pada gambar 2.3.



Gambar 2. 3 *Hyperlane SVM*

(sumber : [23])

SVM pada dasarnya merupakan algoritma klasifikasi *linier*, namun dapat digunakan untuk klasifikasi *non-linier* dengan metode *trick kernel*, yaitu dengan memetakan ruang berdimensi rendah ke ruang berdimensi tinggi. SVM memiliki beberapa jenis fungsi *kernel*, berikut macam-macam *kernel* dan rumusnya.

1. *linear kernel* digunakan untuk klasifikasi data yang sudah terpisah secara linier. Fungsi kernel ini paling sederhana dan cocok ketika terdapat banyak fitur. Berikut rumus persamaannya.

$$K(x, x_i) = \sum(x, x_i)$$

2. *Polynomial kernel* digunakan untuk data yang tidak terpisah secara *linier*. Dimana parameter derajat (d) digunakan untuk mencari nilai optimal pada setiap dataset. Berikut rumus persamaannya.

$$K(x, x_i) = \sum(x, x_i)^d$$

3. *RBK Kernel* digunakan untuk klasifikasi data *non-linier* yang tidak dapat dipisahkan secara *linier*. Berikut rumus persamaannya.

$$K(x, x^i) = \exp(-\gamma ||x - x_i||^2)$$

k. *Confusion Matrix*

Confusion matrix merupakan teknik evaluasi *machine learning* untuk menghitung keakuratan model dengan menghitung prediksi klasifikasi benar dan salah. Perhitungan dapat dilakukan untuk menentukan *accuracy*, *precision*, *recall* dan *f1-score* dan ROC-AUC berikut [24].

1. *Accuracy* merupakan teknik evaluasi untuk mengukur tingkat akurat model dalam memprediksi nilai benar dan salah, berikut rumus *accuracy*.

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

2. *Precision* merupakan teknik evaluasi untuk mengukur tingkat ketepatan akurasi model dalam memprediksi kelas, berikut rumus *precision*.

$$Precision = \frac{TP}{TP+FP}$$

3. *Recall* merupakan teknik evaluasi untuk mengukur kemampuan model dalam menemukan nilai positif yang sesuai dengan data yang ada, berikut rumus *recall*.

$$Recall = \frac{TP}{TP+FN}$$

4. *F1-Score* merupakan teknik evaluasi untuk mengukur kinerja klasifikasi model dengan menggabungkan skor rata-rata *precision* dan *recall* yang dibobotkan, berikut rumus *f1-score*

$$F1-score = \frac{2 \times precision \times recall}{recall + precision}$$

5. *ROC-AUC* adalah metrik evaluasi untuk mengukur performa model klasifikasi pada semua batas klasifikasi yang menggambarkan antara hubungan *True Positif Rate* (TPR) dan *False Positif Rate* (FPR), berikut rumus TPR dan FPR.

$$TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{FP+TN}$$

Keterangan :

TP : *True Positive* adalah data yang benar-benar positif dan hasil klasifikasinya juga bernilai positif.

TN : *True Negative* adalah data yang benar-benar negatif dan hasil klasifikasinya juga bernilai negatif.

FP : *False Positive* merupakan data yang sebenarnya bernilai positif dan hasil klasifikasinya bernilai negatif adalah data yang benar-benar positif dan hasil klasifikasinya bernilai negatif.

FN : *False Negative* adalah data yang benar-benar negatif dan hasil klasifikasinya bernilai positif.

2.3 Perangkat Lunak Yang digunakan

a. Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer belakangan ini dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 dan juga merupakan bahasa pemrograman multifungsi dengan penulisan sintaksis yang mudah dibaca yang didukung dengan library yang lengkap dan memiliki komunitas yang besar.

Kemampuan Bahasa pemrograman python yang multifungsi yang jadi salah satu faktor Bahasa ini menjadi populer yang digunakan dalam berbagai bidang seperti *web development*, *data scient*, *software testing*, *machine learning* dll. Pada bidang *machine learning*, python menyediakan banyak library yang dapat digunakan seperti *Numpy*, *Scikit Learn*, *Tensorflow*, *Keras*, *Pytorch*, dan masih banyak library lain

yang dapat digunakan untuk mempermudah dan mempercepat proses pembuatan *machine learning* [25].

b. Google Colaboratory

Google colaboratory atau yang biasa dikenal *google colab* merupakan produk layanan gratis *cloud computing* dari *Google Internal Research* yang digunakan untuk *editor* dan *compiler* Bahasa pemrograman python secara online.

Dengan *google colab* pengguna dapat membuat model *machine learning* tanpa harus menginstal perangkat lunak tambahan dan juga menyediakan fitur layanan GPU untuk proses *backend* yang dapat dipakai secara gratis selama 12 jam [26]. Gpu sangat berguna untuk proses pelatihan *machine learning* dalam meningkatkan akurasi dan memungkinkan proses transfer data lebih cepat dan efisien [27].

c. *Flask*

Flask merupakan sebuah *web freamework* yang ditulis dengan bahasa pemrograman python dengan menyediakan *template script* dan dua *external libraries* yaitu WSGI toolkit dan Jinja2 *tempalte engine* yang dapat dipakai oleh *developer* untuk membuat aplikasi berbasis web.

Secara kategori, *flask* termasuk jenis *microframework* artinya *flask* merupakan *freamework* minimalis yang tidak memiliki library tertentu dan memiliki ukuran yang cukup ringan menjadikan kinerja *flask* menjadi cukup gesit [28].

d. Visual Studio Code

Visual studio code yang biasa disebut vs code merupakan sebuah *code editor* ringan dan handal yang dikembangkan oleh Microsoft yang tersedia untuk sistem operasi *multiplatform* untuk Linus, MacOS, dan Window. *Text editor* ini mendukung hampir semua bahasa pemrograman seperti JavaScript, Typescript, dan Node.js serta bahasa pemrograman lainnya dengan menggunakan plugin yang dapat dipasang melalui *extension* yang tersedia di vs code seperti Java, Python, Java, Php, C++, Go, dst.

Visual studio code memiliki banyak sekali fitur-fitur diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor-teks editor yang lain [29].

e. Drawio

Drawio merupakan platform visualisasi diagram online yang digunakan untuk membuat berbagai jenis diagram seperti, *flowchart diagram*, diagram organisasi, diagram jaringan dan banyak lagi. Drawio dapat diakses oleh secara gratis melalui link www.draw.io

dengan menggunakan akun google drive untuk penyimpanan, pengguna dapat membuat dan menyimpan diagram secara online tanpa perlu mengunduh atau menginstal perangkat lunak tambahan [30].

2.4 Pemodelan

a. *Flowchart*

Flowchart merupakan penyajian sistematis tentang langkah langkah dan urutan prosedur suatu program dengan penggambaran secara grafik. Biasanya mempengaruhi penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut.

Flowchart juga dapat digunakan untuk menyajikan kegiatan manual, pemrosesan atau keduanya [31]. Berikut simbol-simbol *flowchart* dapat dilihat pada gambar.

	Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	Off-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	Terminator Simbol yang menyatakan awal atau akhir suatu program.		Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	Process Simbol yang menyatakan suatu proses yang dilakukan komputer.		Display Simbol yang menyatakan peralatan output yang digunakan.
	Decision Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

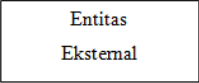
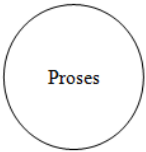
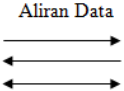
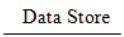
Gambar 2. 4 Simbol-Simbol *Flowchart*

b. *Data Flow Diagram (DFD)*

Data flow digram atau DFD merupakan suatu diagram yang menggambarkan arus data sistem dengan menggunakan notasi-notasi

yang menunjukkan hubungan antara data pada sistem dan proses pada sistem.

DFD adalah representasi grafik sistem dengan menunjukkan bagian sistem, aliran data di mana bagian tersebut berada, dan asal tujuan dan penyimpanan data [32]. Gambar berikut menunjukkan komponen DFD

Yourdon/De Marco	Keterangan
	Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi di luar sistem.
	Orang/unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
	Aliran data dengan arah khusus dari sumber ke tujuan
	Penyimpanan data atau tempat data dilihat oleh proses.

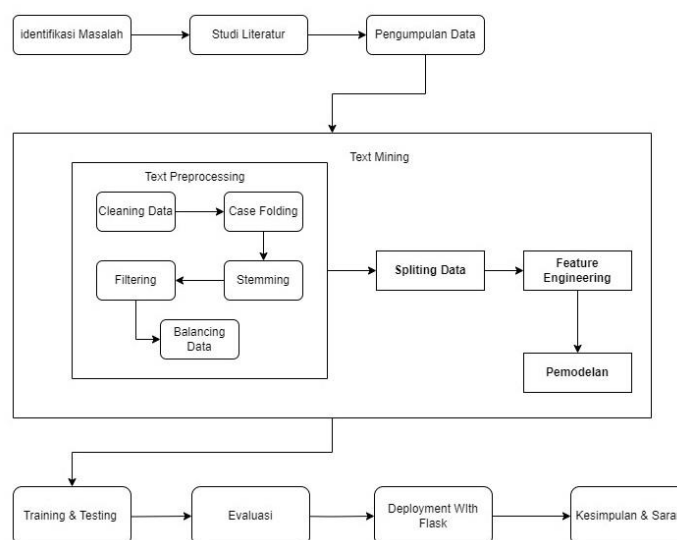
Gambar 2. 5 Komponen Data Flow Diagram

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Metodologi Penelitian merupakan tahapan-tahapan yang harus dilakukan penulis yang dijadikan sebagai panduan. Tahapan-tahapan meliputi sebagai berikut.

3.1 Kerangka Berpikir



Gambar 3. 1 Kerangka Berpikir

3.2 Pengumpulan Data

3.2.1 Sumber Data

Pada penelitian ini menggunakan data yang diambil dari situs *Kaggle* dengan *keyword* pencarian “*Indonesian Fact and Hoax Political News*” yang berisi berita berbahasa Indonesia yang telah di *scraping* dari situs CNN, Kompas, Tempo untuk dataset berita valid dan *Turnbackhoax* untuk dataset berita hoax, berikut informasi lengkap terlihat pada tabel 3.1.

Tabel 3. 1 Kumpulan Dataset Berita yang di Peroleh

Sumber	Tanggal		Total Dataset
	Awal	Akhir	
CNN	01 Januari 2021	21 Februari 2023	10001
Kompas	20 April 2017	21 Februari 2023	4749
Tempo	01 Januari 2021	04 Februari 2023	6591
Turbackhoax	08 September 2015	28 Februari 2023	10383

3.3 Metode Pengembangan Model *Machine Learning*

3.3.1 *Text Preprocessing*

Text preprocessing adalah proses yang terpenting dan juga tahapan pertama untuk pembuatan model *machine learning* pada proses ini berfungsi untuk menyeleksi data teks agar lebih terstruktur dan siap untuk diolah lanjut. Berikut penjelasan tahapan-tahapan yang dilakukan

a. *Cleaning Data*

Cleaning data merupakan proses menyeleksi, menemukan dan mengoreksi data yang tidak lengkap, hilang, duplikat, format yang tidak tepat dan kesalahan lainnya agar data lebih terstruktur dan siap untuk diolah lanjut.

b. *Case Folding*

Case folding merupakan proses mengubah format huruf kapital jadi huruf kecil (*lowercase*). Tujuannya untuk menyamaratakan teks agar kata-kata tidak terdeteksi berbeda karena perbedaan huruf kapital.

c. *Stemming*

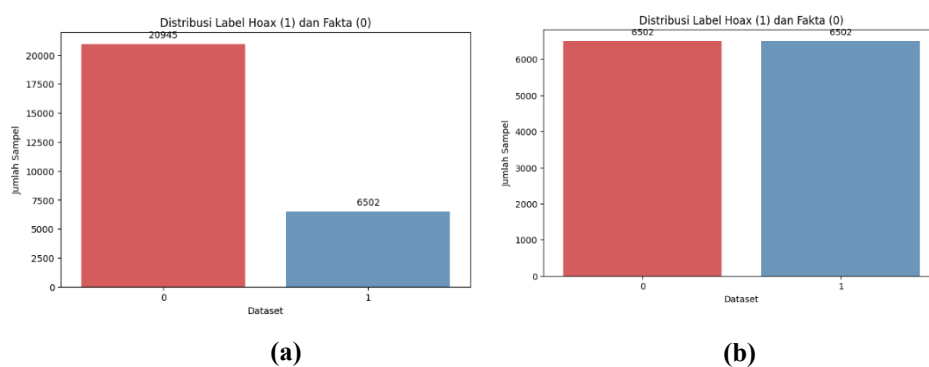
Stemming merupakan teknik untuk mengurangi kata-kata ke bentuk dasar atau akar kata. bertujuan untuk menghilangkan infleksi kata yang tidak diperlukan, seperti imbuhan dan konjugasi, sehingga kata-kata dengan akar yang sama dapat dikenali.

d. *Filtering*

Filtering merupakan proses untuk membersihkan dan menyaring kata-kata dari elemen yang tidak relevan atau tidak diinginkan. Seperti menghapus kata-kata umum yang tidak bearti, tanda baca, whitespace, angka, karakter spesial, url dll.

e. *Balancing Data*

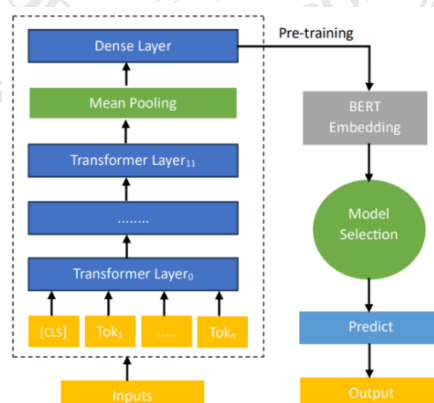
Setelah dilakukan *cleaning data*, *case folding*, *stemming*, dan *filtering* kemudian semua dataset digabungkan, dari dataset tersebut terdapat ketidakseimbangan data antara berita valid dan hoax yaitu 20945 data valid dan 6502 data hoax, oleh karena itu dilakukan balancing data dengan teknik under sampling. Data yang tidak seimbang bisa menyebabkan model *overfitting*. Dapat dilihat pada gambar 3.2 berikut.



Gambar 3. 2 Persebaran Dataset (a) Sebelum *Balancing* (b) Sesudah *Balancing*

3.3.2 Feature Engineering

Pada tahap ini dilakukan untuk mengekstraksi dan mengonversi data mentah ke dalam format yang dapat digunakan oleh algoritma *machine learning*. Supaya model dapat bekerja lebih baik dan mudah memahami, dengan melakukan pengembangan fitur baru, pemilihan fitur dan modifikasi fitur yang ada. Peneliti menggunakan model BERT sebagai *feature engineering* dengan memanfaatkan *embedding word* milik BERT yang dihasilkan melalui token-token yang di transformasi melalui 12 lapisan. setelah melewati transformer representasi dari setiap token teks digabungkan dengan teknik mean polling. Kemudian di akhir lapisan ada dense layer berfungsi untuk menggabungkan fitur-fitur telah dipelajari untuk melakukan tugas prediksi dan klasifikasi berikut tahapan untuk membuat token *embedding* BERT. Dapat dilihat pada gambar 3.3 untuk *feature engineering* BERT.



Gambar 3. 3 Feature Engineering BERT

a. Tokenizing BERT

Tokenizing merupakan tahap pemotongan teks input atau string menjadi kata yang berdiri sendiri berdasarkan kata yang

menyusunnya sehingga dapat dianalisa. Pada *tokenizing* BERT menggunakan *wordpiece tokenizer* untuk menghasilkan *dictionary*.

Token tersebut akan dipetakan ke indeks pada kosakata *tokenizer*. Dan juga ada proses *padding* dan *truncation*, yang tujuannya untuk menyesuaikan ukuran panjang kalimat dengan menambah atau mengurangi sesuai dengan kalimat yang diset. Serta juga terdapat *attention mask* yaitu *array* angka 0 dan 1 bertujuan untuk membedakan *padding* dan kalimat asli dengan memberitahu mekanisme *self-attention* pada BERT untuk tidak memasukan token *padding* ke dalam interpretasi kalimat. Berikut contoh hasil *tokenizing* BERT dapat dilihat pada tabel 3.2.

Tabel 3. 2 Contoh Hasil *Tokenizing* BERT

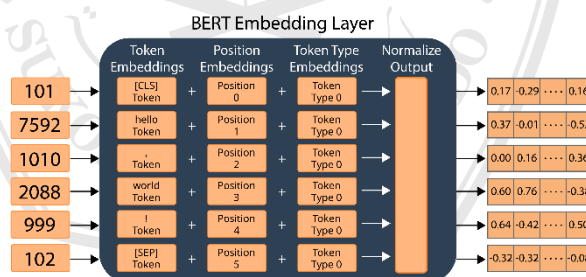
Teks		Token	Token IDS
Jokowi	Wanti-wanti	'jokowi', 'want', '##i', 'want', '##i',	5237, 16862, 30356,
Peluang	Kerja di	'peluang', 'kerja', 'di',	16862, 30356, 2882,
Masa Depan	Bisa Di	'masa', 'depan', 'bisa',	494, 26, 890, 1219, 166, 11134, 8490,
	gantikan Robot	'digantikan', 'robot'	
Sempat	Sindir	'sempat', 'sind', '##ir',	2740, 8244, 72, 4846, 6438, 30468,
Miskin	Gagasan,	'miskin', 'gagasan', ,', 'kini',	1562, 3443, 1049, 6052, 6392, 18006,
Kini	Joman Ngarep	'jo', '##man', 'nga',	30359, 22328, 11283,
Ganjar	Duet	'##rep', 'ganja', '##r', 'duet',	
Prabowo		'prabowo'	

Terlihat pada tabel diatas teks akan dipecah ke kebentuk yang lebih kecil biasanya kedalam kata atau sub-sub kata dan akan di konversi ke token ids numerik yang berdasarkan *vocabulary* (kosakata) yang telah dilatih.

Selain itu BERT juga menambahkan token [CLS] diawal setiap input, untuk merepresentasikan seluruh kalimat atau urutan teks. Token ini akan digunakan dalam tugas klasifikasi. Dan juga token [SEP] ditambahkan disetiap akhir input untuk memisahkan kalimat atau untuk menandai akhir dari urutan.

b. *Embedding* BERT

Embedding BERT merupakan representasi dari token-token yang telah di *tokenized* seperti terlihat pada gambar 3.4.



Gambar 3. 4 Ilustrasi *Embedding* BERT

Untuk melakukan *embedding* BERT mempunyai tiga proses *embedding* yaitu *token embedding*, *segmen embedding* dan *positional embedding* kemudian semua token embedding disatukan dan akan di *encoding* kemudin dimasukan kedalam layer *transformer* sebanyak 12 *layer* untuk BERT-*base* dan 16 *layer* untuk BERT-*large*.

1. *Token Embedding*

Merupakan representasi dari token ids yang dihasilkan dari *tokenize* BERT dari pemecahan kata yang dimana token ids akan di ubah menjadi vektor *embedding* dengan dimensi tetap.

2. *Segment Embedding*

Apabila input terdiri dari dua kalimat, setiap token akan dipisah dengan ditandai angka 0 untuk kalimat pertama dan 1 untuk kalimat kedua, sebagai pembeda antara kalimat pertama dan kedua.

3. *Positional Embedding*

BERT menggunakan *positional embedding* untuk meng-*encode* posisi setiap token dalam urutan input bertujuan untuk memberikan informasi tentang urutan token, karena BERT tidak memiliki mekanisme bawaan untuk memahami urutan token.

Dari tiga proses *embedding* tersebut akan digabungkan agar informasi pada teks tetap terjaga, berikut contoh hasil *embedding* dengan kata Jokowi presiden Indonesia, terlihat pada gambar 3.5.

[illegible]

Gambar 3. 5 Contoh Hasil *Embedding* BERT

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *System Development Life Cycle* (SDLC). SDLC adalah metodologi yang umum digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Adapun tahapan-tahapan sebagai berikut.

3.4.1 Planning

Pada tahap ini merupakan tahap awal untuk pembuatan sistem dengan mengidentifikasi dan mempelajari konsep sistem dari permasalahan yang ada.

3.4.2 Analysis

Pada tahap ini dilakukan analisa kelayakan sistem dengan menentukan kebutuhan sistem yang telah teridentifikasi dari permasalahan berdasarkan target pengguna.

- a. Kebutuhan Fungsional

1. Sistem Deteksi Berita hoax dibangun berdasarkan *model feature engineering* BERT yang telah dilatih dan ambil model yang memiliki performa baik.
2. Model yang telah dibuat di implementasikan kedalam web dengan *freamwork* flask

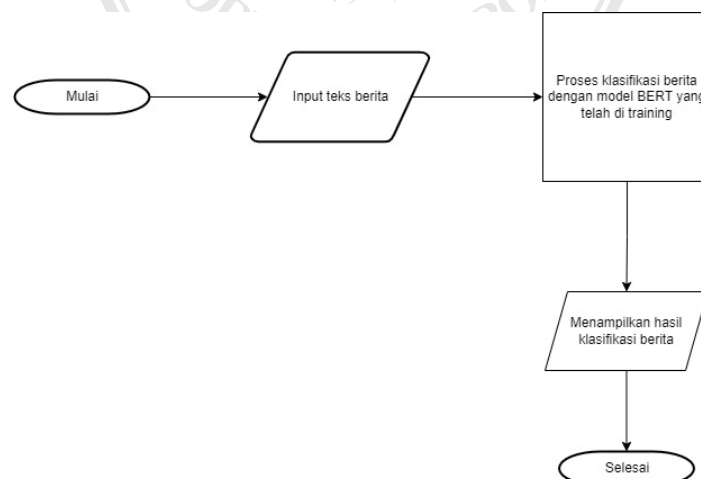
b. Kebutuhan Non-Fungsional

1. Sistem dapat berjalan di berbagai perangkat web
2. Waktu respon sistem kurang dari satu menit dalam melakukan prediksi
3. Sistem dapat diakses 24 jam

3.4.3 Design

a. *Flowchart* Sistem

Gambaran umum sistem deteksi berita *hoax* disajikan dalam *flowchart* dapat dilihat pada gambar 3.6 berikut.



Gambar 3. 6 *Flowchart* Sistem

b. *Data Flow Diagram* (DFD)

1. *Context Diagram*

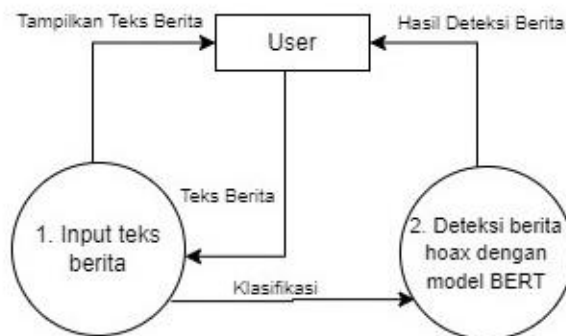
Context diagram pada sistem ini hanya memiliki satu entitas yaitu *user* yang dapat mengakses dan menjalankan sistem deteksi berita *hoax* dengan alur data berupa *input* teks berita dari *user* dan diproses oleh sistem dengan algoritma BERT, dan alur data berupa teks berita dari sistem menuju *user* dapat dilihat pada gambar 3.7 berikut.



Gambar 3. 7 Context Diagram Sistem

2. Data Flow Diagram (DFD) Level 0

DFD level 0 dapat diketahui bahwa terdapat dua proses, yaitu input teks berita dan proses deteksi atau klasifikasi berita *hoax*. Pada proses input teks berita terdapat alur data berupa teks berita dari *user* ke sistem, dan tampilan teks berita yang telah diinputkan dari sistem ke *user*. Dan untuk proses deteksi berita dilakukan dengan algoritma BERT terdapat alur data berupa proses input teks berita dari *user* menuju proses kedua yaitu proses deteksi berita dengan klasifikasi algoritma BERT ke sistem, dan alur data hasil deteksi berita dari sistem ke *user* dapat dilihat pada gambar 3.8 berikut.



Gambar 3. 8 Data Flow Diagram Level 0

c. *Interface Design*

1. *Interface Input*

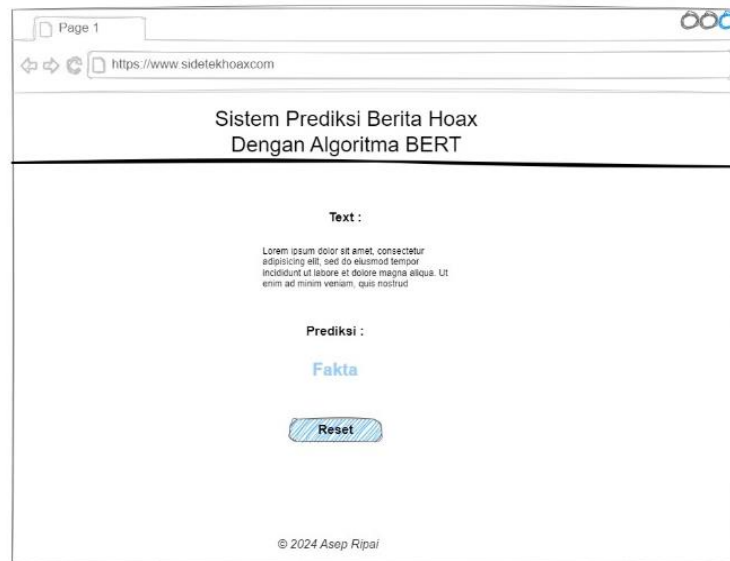
Desain *interface input* sistem deteksi berita *hoax* dapat dilihat pada gambar 3.9 berikut.



Gambar 3. 9 Desain Interface input

2. *Interface Output*

Desain *interface output* sistem deteksi berita *hoax* dapat dilihat pada gambar 3.10 berikut.



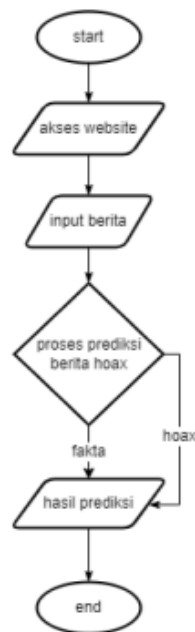
Gambar 3. 10 Desain *Interface Output*

3.4.4 *Implementation*

Pada tahap ini setelah membuat model *machine learning* akan di simpan kedalam bentuk file ekstensi bin dan akan dilakukan pembuatan code dengan freamwor flask untuk mengimplentasikan model *machine learning* kedalam aplikasi berbasis website yang dapat diakses dimana saja.

3.4.5 *Testing*

Setelah selesai membangun sistem harus dilakukan pengujian agar sistem berjalan sesuai dengan harapan, dalam hal ini peneliti melakukan pengujian dengan metode *black box*, yaitu dengan menguji semua fungsionalitas pada sistem, berikut *flowchart* skenario pengujian sistem dapat dilihat pada gambar 3.11.



Gambar 3. 11 Flowchart Pengujian Sistem

a. Mengakses Website

Langkah pertama yang dilakukan adalah pengguna mengakses website *url* apabila telah dihosting, untuk saat ini peneliti menggunakan localhost pengguna membuka website melalui `http://127.0.0.1:5000` setelah memberika perintah `python -m flask run` pada CMD

b. *Input* Berita

Selanjutnya pengguna dapat menginputkan teks berita yang akan diprediksi pada *form* yang telah disediakan, inputan harus berupa teks valid dan tidak boleh kosong.

c. Menunggu Prediksi

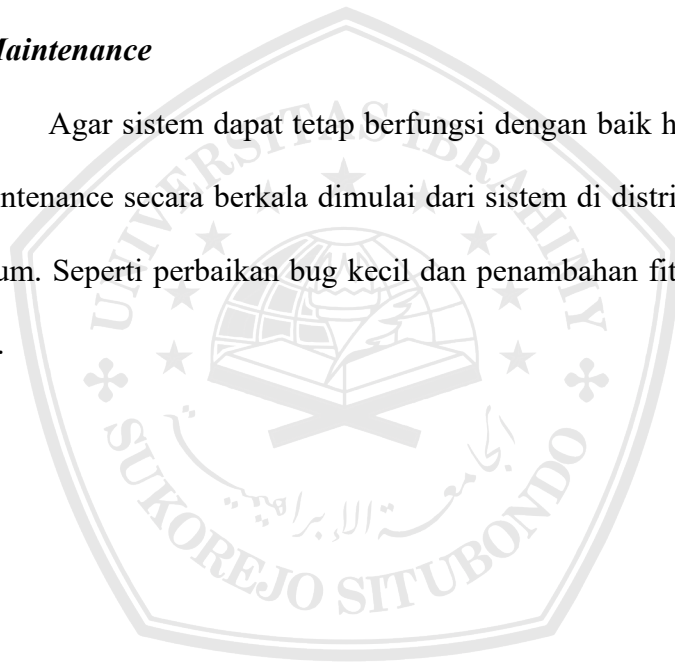
Pada tahap ini setelah pengguna memasukkan teks dan menekan tombol prediksi, sistem akan memproses teks untuk diprediksi.

d. Melihat Hasil

Tahap terakhir setelah pengguna menunggu prediksi dari sistem akan menampilkan hasil berupa prediksi dan pengembalian teks *input*.

3.4.6 Maintenance

Agar sistem dapat tetap berfungsi dengan baik harus dilakukan maintenance secara berkala dimulai dari sistem di distribusikan secara umum. Seperti perbaikan bug kecil dan penambahan fitur yang belum ada.



BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Skenario Pengujian *Machine Learning*

Untuk mendapatkan model *machine learning* yang memiliki performa baik maka akan dilakukan pengujian dan evaluasi terhadap model yang dibuat. Adapun model yang dibuat menggunakan metode *feature engineering* dari arsitektur BERT untuk menghasilkan *text embedding* yang akan diproses dengan tiga algoritma klasifikasi yaitu BERT, *Random Forest* dan *Support Vektor Machine*.

4.1.1 *Load Datasets*

Setelah dataset selesai melewati tahap *text preprocessing*, maka dataset siap digunakan untuk di pelatihan dan pengujian model dengan membagi dataset untuk data *training* 80% dan data *testing* 20% dengan total dataset sebanyak 13.408 data berita. Berikut hasil dataset yang di *load* yang telah di *text prepoessing* dapat dilihat pada gambar 4.1.

	text	label
0	psi banding pemecatan viani limardi ditolak pe...	0
1	paslon tunggal pilkada bawaslu parpol introspe...	0
2	dpr pemerintah kpu sepakat honor petugas tps k...	0
3	presiden jokowi bahas perang rusia ukraina men...	0
4	koalisi pkb pks dinilai redam politik pecah be...	0
...	...	...
13043	raja salman arab saudi membawa orang orang men...	1
13044	hehe tersenyum tingkah laku jokowi terimakasih...	1
13045	jokowi walikota periode solo salah s d mari bu...	1
13046	rabu nilai tukar rupiah terpuruk melewati rp d...	1
13047	mari tolak desember acara pastur monas majlis ...	1
13048 rows x 2 columns		

Gambar 4. 1 Hasil *Load Datasets*

4.1.2 Modelling

a. BERT

Proses *modelling* BERT yang digunakan BERT berbahasa Indonesia dari hugingface yaitu, “indobenchmark/indobert-base-pl” dan menggunakan fungsi dari library transformer untuk *me-load* model BERT, yaitu “*from transformers import AutoTokenizer, AutoModelForSequenceClasification, AutoModel, Trainer, TrainingArguments*”.

Fungsi *AutoModelForSequenceClasification* merupakan kelas yang dipakai untuk tugas klasifikasi urutan berdasarkan arsitektur BERT, dan berikut proses *training* dan pengujian.

1. *Fine Tuninng* dan *Training*

BERT merupakan model *pre-trained* sehingga tidak dapat dipakai langsung untuk proses klasifikasi berita. Oleh karena itu diperlukan proses *finetunning* yaitu, proses memperbaharui semua parameter dalam untuk tugas tertentu, dan *training* merupakan proses belajar sebuah model dari sebuah data untuk mendapatkan pengetahuan.

Finetunning pada model BERT yaitu dengan mengatur *hyperparameter* berupa *batch size*, *learning rate*, dan jumlah *epoch*, berikut *hyperparameter* yang digunakan disajikan pada tabel 4.1.

Tabel 4. 1 *Hyperparameter* Model BERT

No	<i>Hyperparameter</i>	Ukuran
1	<i>Batch Size</i>	8 dan 16
2	<i>Learning Rate</i>	2e-5
3	<i>Epsilon</i>	1e-8
4	<i>Epoch</i>	5

Berikut penjelasan *hyperparameter* yang digunakan oleh peneliti.

a) *Batch Size*

Batch size adalah ukuran jumlah sampel data yang diberikan pada model dalam satu perulangan dalam proses *training* atau *testing* pada model *machine learning*. Pada tabel diatas dapat dilihat peneliti memakai *batch size* 8 untuk proses *training* dan 16 untuk proses *testing*.

b) *Learning Rate*

Learning rate merupakan salah parameter penting dalam model *machine learning* untuk melakukan optimasi. *Learning rate* menentukan seberapa besar langkah yang diambil untuk memperbarui bobot model selama proses *training*. Pada tabel diatas memakai ukuran *learning rate* sebesar 2e-5 yang mewakili 2 kali 10 pangkat – 5 atau setara dengan 0,00002

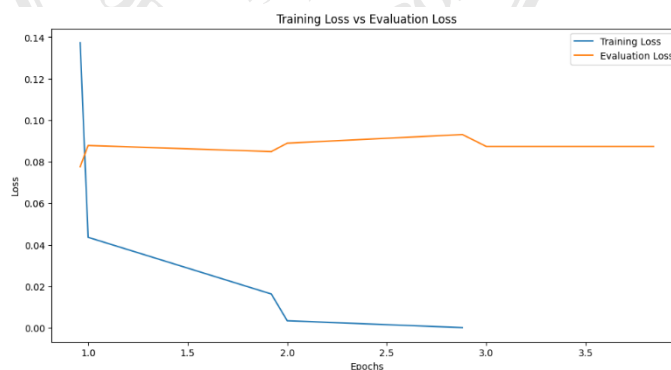
c) *Epsilon*

Epsilon merupakan parameter untuk optimasi yang penting bagi model *machine learning* untuk menghindari masalah pembagian dengan nol ketika menghitung *gradient*. Seperti pada tabel diatas peneliti menggunakan *epsilon* sebesar $1e-8$ yang merupakan representasi dari bilangan desimal yang mewakili 1 kali 10 pangkat -8.

d) *Epoch*

Epoch adalah siklus ketika seluruh dataset sudah selesai melewati proses *training* dalam sekali *iteration* untuk memperbarui bobot model. Sebagaimana pada tabel diatas menggunakan 5 *epoch* berarti akan dilakukan 5 kali *iteration* dari seluruh dataset yang telah di *batch*.

Dari percobaan *hyperparameter* yang digunakan untuk *training* dapat dilihat hasilnya pada gambar 4.2 berikut.

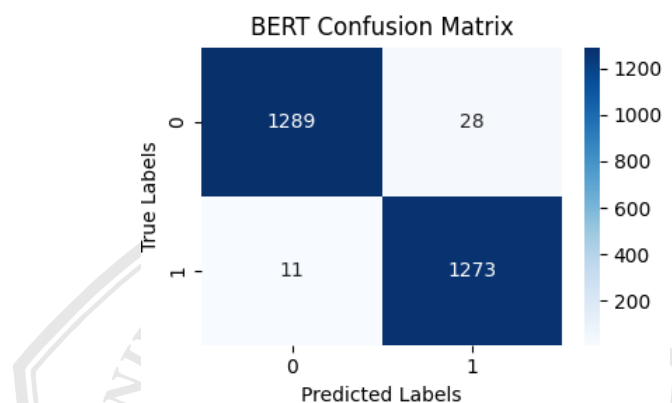


Gambar 4. 2 Hasil *Finetuning* BERT

2. *Testing dan evaluation model*

Agar model dapat berjalan dengan baik sebelum masuk proses *deployment* model harus dilakukan pengujian dan

validasi terhadap data yang belum dikenali. Salah satu metode untuk pengujian dan evaluasi adalah metode *confusion matrix* peneliti menggunakan data *testing* dari data sebelumnya yang telah di bagi yaitu 20% dari seluruh dataset. Berikut hasil uji *confusion matrix* dapat dilihat pada gambar 4.3.



Gambar 4. 3 Hasil *Confusion Matrix* BERT

Berdasarkan gambar diatas dapat diketahui pada baris pertama yang menunjukkan data berita valid yang di prediksi benar ada 1289 dan 28 untuk data valid di prediksi salah (*hoax*).

Pada baris kedua menunjukan data yang salah (*hoax*) yang di prediksi salah (*hoax*) ada 1273 dan 11 data salah diprediksi berita valid. Berdasarkan hasil *confusion matrix* dapat diketahui performa model dengan menghitung menggunakan metode *matrix* yaitu, *accuracy*, *precission*, *recall*, dan *f1-score*, berikut perhitungannya.

$$\text{Persamaan } accuracy = \frac{1289+1273}{1289+1273+11+28} = 98\%$$

$$\text{Persamaan } precision = \frac{1289}{1289+28} = 97\%$$

$$\text{Persamaan recall} = \frac{1289}{1289+11} = 99\%$$

$$\text{Persamaan fl-score} = \frac{2 \times 97 \times 99}{99 + 97} = 97\%$$

b. *Random Forest*

Proses *modelling* pada algoritma *random forest* menggunakan *library* dari *scikit learn* yaitu, '*from sklearn.ensemble import RandomForestClassifier*', merupakan fungsi yang digunakan untuk me-load model *random forest* untuk proses klasifikasi.

1. *Training*

Random forest memiliki beberapa parameter yang dapat digunakan untuk mencari performa yang terbaik, berikut parameter yang digunakan untuk *training*. dapat dilihat pada tabel 4.2.

Tabel 4. 2 Parameter *Random Forest* yang di Gunakan

No	Parameter	Ukuran
1	<i>N_estimator</i>	100
2	<i>Random_state</i>	42

Berikut penjelasan parameter yang digunakan peneliti.

a) *N_estimator*

N_estimator pada *random forest* mengacu pada jumlah pohon keputusan yang akan dibuat dan digabungkan dengan model berdasarkan subset acak dari data *training* dan fitur, dan menghasilkan rata-rata atau dari setiap pohon. Seperti tabel diatas peneliti menggunakan nilai 100 untuk

n_estimators yang berarti akan dibuat 100 pohon keputusan yang digunakan dalam model ini.

b) *Random_state*

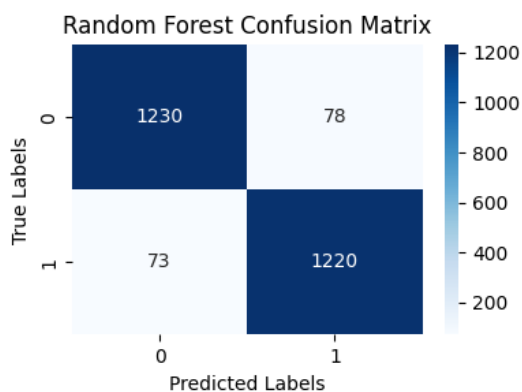
Random_state merupakan parameter yang mengacu pada replikasi hasil eksperimen yang digunakan untuk memastikan hasil yang tetap sama setiap kali menjalankan model dengan parameter yang sama. Seperti pada tabel diatas menggunakan nilai 42 untuk *random_state* jadi ketika akan mengulang model untuk mendapat nilai yang tetap maka memakai nilai tersebut. Berikut hasil *training* dapat dilihat pada gambar 4.4.

RANDOM FOREST				
	precision	recall	f1-score	support
0	0.94	0.94	0.94	1308
1	0.94	0.94	0.94	1293
accuracy			0.94	2601
macro avg	0.94	0.94	0.94	2601
weighted avg	0.94	0.94	0.94	2601

Gambar 4. 4 Hasil *Training Model Random Forest*

2. *Testing dan evaluation model*

Untuk memastikan model dapat berjalan dengan baik maka harus dilakukan pengujian dan evaluasi. Peneliti melakukan pengujian menggunakan dataset yang telah dibagi yaitu mengambil 20% dari total dataset dan untuk evaluasi menggunakan metode *confusion matrix*. Berikut hasil *confusion matrix* dapat dilihat pada gambar 4.5.



Gambar 4. 5 Hasil *Confusion Matrix Random Forest*

Berdasarkan gambar diatas dapat diketahui hasil uji dengan *confusion matrix*, pada baris pertama menunjukkan hasil prediksi pada data berita valid yang di prediksi benar adalah 1230 dan di prediski salah (*hoax*) ada 78.

Pada baris kedua yang menunjukkan data hasil prediksi pada berita salah (*hoax*) yang diprediksi salah adalah 1220 dan yang di prediksi benar ada 73. Berdasarkan hasil uji *confusion matrix* dapat diketahui performa model dengan menggukan metode *matrix* yaitu, *accuracy*, *precission*, *recall*, dan *f1-score*, berikut perhitungannya.

$$\text{Persamaan } accuracy = \frac{1230+1220}{1230+1220+78+73} = 94\%$$

$$\text{Persamaan } precision = \frac{1230}{1230+78} = 94\%$$

$$\text{Persamaan } recall = \frac{1289}{1289+73} = 94\%$$

$$\text{Persamaan } f1\text{-score} = \frac{2 \times 94 \times 94}{94 + 94} = 94\%$$

c. *Support Vektor Machine*

Modelling pada algoritma SVM memiliki dua jenis yaitu regresi dan klasifikasi dengan memakai *library* dari *sicikit learn* yaitu, '*from sklearn.svm import SVC*' merupakan sebuah kelas yang berfungsi untuk melakukan tugas klasifikasi.

1. *Training*

Untuk membuat model yang memiliki performa yang baik perlu dilakukan pengaturan paramater, berikut parameter yang digunakan untuk *training* pada algoritma SVM dapat dilihat pada tabel 4.3.

Tabel 4. 3 Parameter SVM yang di Gunakan

No	Parameter	Ukuran
1	<i>Probability</i>	<i>True</i>
2	<i>C</i>	1
3	<i>Gamma</i>	0.1
4	<i>Kernel</i>	<i>Linier</i>

Berikut penjelasan parameter yang digunakan peneliti.

a) *Probability*

Probability pada SVM mengacu pada opsi yang memungkinkan model SVM mengeluarkan probabilitas prediksi pada setiap kelas yang dapat memberikan informasi tambahan. Peneliti mengisi parameter *probability*

dengan *true* yang berarti, mengaktifkan probabilitas untuk model yang digunakan

b) *C*

Parameter *c* pada algoritma SVM berfungsi untuk meminimalkan kesalahan klasifikasi pada margin dengan mengontrol *trade off* untuk menghindari penalti pada margin. Ukuran parameter yang digunakan peneliti adalah 1 seperti pada tabel diatas, pemilihan ukuran yang tepat merupakan kunci keseimbangan antara generalisasi dan fit pada data pelatihan.

c) *Gamma*

Gamma pada merupakan parameter SVM yang mengacu pada penggunaan parameter *kernel* yang menentukan seberapa jauh pengaruh satu contoh pelatihan yang dapat dijangkau terhadap batas margin. Dalam hal ini peneliti memakai nilai *gamma* yaitu 0.1.

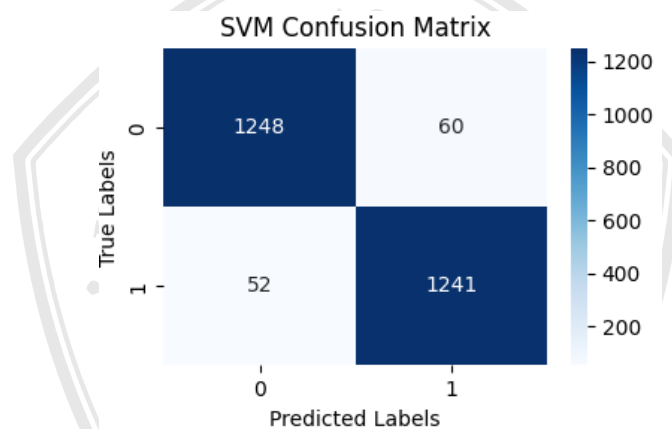
d) *Kernel*

Parameter *kernel* pada SVM digunakan untuk mengubah data *input* ke ruang yang lebih tinggi bertujuan untuk membuat data yang tidak bisa dipisahkan secara *linier* menjadi dapat dipisahkan secara linier dengan memanfaatkan fitur ruang yang lebih tinggi. *Kernel* yang digunakan peneliti adalah jenis *linier* karena data yang di

miliki sudah terpisah secara *linier* dan juga merupakan *kernel* yang sederhana.

2. *Testing dan evaluation model*

Agar model dapat berjalan baik ketika implementasi tentu harus dilakukan pengujian dan evaluasi. Dalam pengujian peneliti mamakai dari dataset yang telah di bagi sebanyak 20% untuk data *testing* kemudian dilakukan evaluasi dengan metode *confusion matrix*. Berikut hasilnya dapat dilihat pada gambar 4.6.



Gambar 4. 6 Hasil *Confusion Matrix* SVM

Berdasarkan hasil pengujian dari *confusion matrix* dapat diketahui pada baris pertama menunjukkan data berita yang valid diprediksi benar ada 1248 dan data yang diprediksi salah ada 60.

Pada baris kedua menunjukkan data berita salah (*hoax*) yang diprediksi salah ada 1241 dan data yang diprediksi benar ada 52. Berdasarkan hasil uji *confusion matrix* dapat diketahui performa model dengan menggunakan metode *matrix* yaitu, *accuracy*, *preciission*, *recall*, dan *f1-score*, berikut perhitungannya.

$$\text{Persamaan accuracy} = \frac{1248+1241}{1248+1241+60+52} = 96\%$$

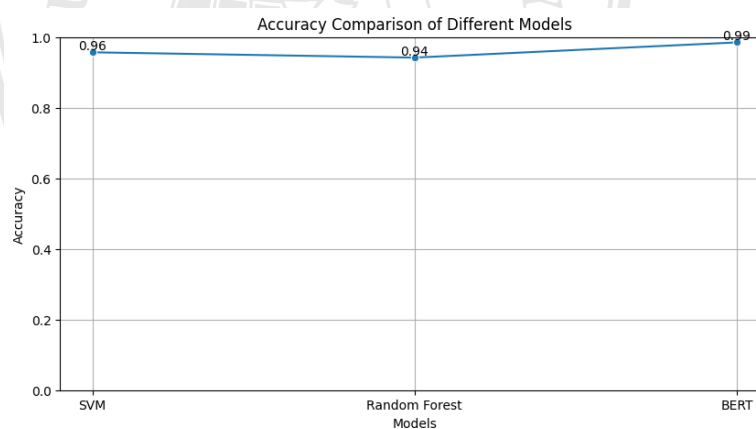
$$\text{Persamaan precision} = \frac{1248}{1248+60} = 95\%$$

$$\text{Persamaan recall} = \frac{1248}{1248+52} = 95\%$$

$$\text{Persamaan f1-score} = \frac{2 \times 95 \times 95}{95 + 95} = 95\%$$

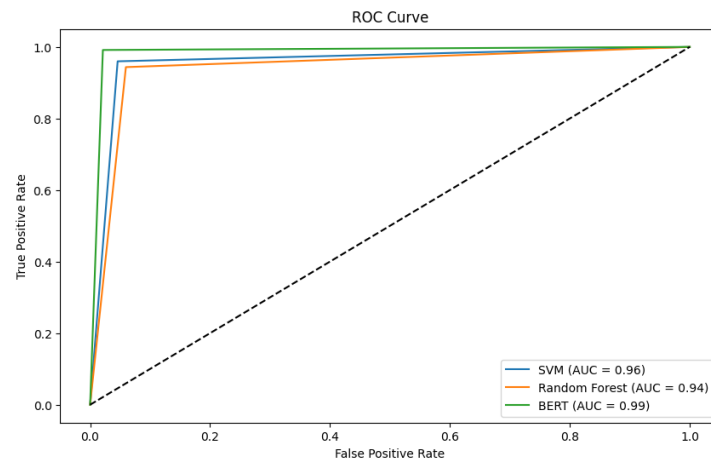
4.1.3 Komparasi Model

Setelah selesai tahapan pelatihan dan pengujian kemudian dilakukan perbandingan performa semua model guna untuk memilih model yang terbaik. Dengan membandingkan hasil akurasi dan menguji dengan metode ROC-AUC. Berikut gambar 4.7 hasil perbandingan akurasi.



Gambar 4. 7 Perbandingan Akurasi Semua Model

Dapat diketahui dari gambar diatas hasil akurasi yang paling tinggi adalah model dengan algoritma BERT sebesar 99% dan diikuti algoritma SVM 96, algoritma *random forest* 94%. Berikut hasil pengujian metode ROC-AUC untuk menentukan performa klasifikasi terbaik dapat dilihat pada gambar 4.8.



Gambar 4. 8 Hasil Perbandingan Kurva ROC-AUC

Berdasarkan gambar diatas dapat diketahui bahwa semua model berada diatas ambang batas dengan garis diagonal yang menunjukan tingkat klasifikasi model berjalan sangat baik. Dari ketiga model, BERT memiliki performa lebih baik dengan nilai ROC-AUC 0.99. BERT dapat memiliki performa yang lebih dikarenakan BERT merupakan merupakan algoritma berbasis deep learning yang memiliki banyak *layer* (*deep neural network*) akan tetapi memiliki kekurangan yaitu membutuhkan biaya komputasi lebih besar dan waktu yang lama.

4.2 Analisis Teks

Setelah mendapatkan model terbaik dari hasil pengujian dan evaluasi yaitu model BERT selanjutnya analisis teks pada hasil prediksi untuk melihat pola dengan memvisualisasikan dengan *wordcloud*, dan *wordlink*.

Wordcloud berguna untuk melihat keterkaitan kata yang sering muncul, *wordlink* berguna untuk melihat keterhubungan antar kata pada dataset yang digunakan.



Gambar 4. 9 (a) *Word Cloud Berita Hoax* , (b) *Word Link Berita Hoax*

Berdasarkan gambar 4.9 diatas dapat diketahui pada kata yang sering muncul pada *word cloud* di representasikan dengan kata-kata yang berukuran lebih besar yaitu Jokowi, Indonesia, foto, covid, video. Dan berdasarkan *word link* terdapat kata yang menonjol yaitu video dan foto yang mengindikasikan berita *hoax* sering menggunakan foto atau video yang tidak sesuai dengan isi berita untuk membuat isu dan memprovokasi pembaca agar percaya.

4.3 Skenario Pengujian Sistem

Pada tahap pengujian sistem peneliti menggunakan metode *black box* yaitu pengujian yang fokus pada fungsionalitas sistem, bagaimana sistem berinteraksi dengan pengguna. Berikut hasil pengujian sistem dapat dilihat pada tabel 4.4.

Tabel 4. 4 Hasil Pengujian Sistem

No	Skenario Pengujian	Hasil Harapan	Hasil Pengujian
1	Pengguna mengakses aplikasi website	Sistem menampilkan antarmuka website	Baik

Tabel Lanjutan 4.4

2	Menginputkan form dengan text kosong	Sistem menolak dan memberikan peringatan teks wajib diisi	Baik
3	Menginputkan teks berita valid	Sistem memberikan <i>feedback</i> “ASLI”	Baik
4	Menginputkan teks berita <i>hoax</i>	Sistem memeberikan <i>feedback</i> ”Hoax”	Baik
5	Menekan tombol prediksi dan Menunggu Prediksi	Sistem memberikan prediksi kurang dari satu menit	Baik
6	Menekan tombol reset	Sistem mereset hasil prediksi ke semula	Baik

4.4 Segmen Program

4.4.1 Segmen Program *Text Preprocessing*

Segmen 4.1 ini merupakan *source code* untuk tahapan *text preprocessing* dengan bahasa pemrograman python melalui google collab.

Segmen Program 4. 1 *Text Peprocessing*

```
//load dataset
def excel_to_df(excel):
    df = pd.read_excel(excel, index_col='index')
    # df = df.drop(columns=df.columns=[0])
    return df

df_cnn =
excel_to_df('data/Summarized/dataset_cnn_summarized.xlsx')
```

Segmen Program 4.1 (Lanjutan)

```
df_kompas =
excel_to_df('data/Summarized/dataset_kompas_summarized.xlsx')
df_tempo =
excel_to_df('data/Summarized/dataset_tempo_summarized.xlsx')
df_turnbackhoax=excel_to_df('data/Summarized/dataset_turnbackhoax_summarized.xlsx')
//proses perbaikan dataset
df_cnn= df_cnn[df_cnn['cleaned'].isna() == False]
df_kompas = df_kompas[df_kompas['cleaned'].isna() == False]
df_tempo = df_tempo[df_tempo['cleaned'].isna() == False]
df_turnbackhoax =
df_turnbackhoax[df_turnbackhoax['cleaned'].isna() == False]

df_cnn['cleaned'] = df_cnn['cleaned'].astype(str)
df_kompas['cleaned'] = df_kompas['cleaned'].astype(str)
df_tempo['cleaned'] = df_tempo['cleaned'].astype(str)
df_turnbackhoax['cleaned'] =
df_turnbackhoax['cleaned'].astype(str)
from mpstemmer import MPStemmer
import Levenshtein
import json
stemmers = MPStemmer()

def wordopt(text):
    # Menghapus karakter non-ASCII
    text = unicodedata.normalize('NFKD', text).encode('ascii',
'ignore').decode('utf-8', 'ignore')

    # Menghapus URL
    text = re.sub(r'(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-9.\-
]+[.])a-
z){2,4}/)(?:[^\s()<>+|\\((([^\s()<>+|\\([^\s()<>+|\\)))*\\))+(?:\\(
([^\s()<>+|\\([^\s()<>+|\\)))*\\)|[^\s`!()\\[\]{};:~\".,<>?«»"'"']
))', '', text)

    # Menghapus tanda baca
    text = re.sub(r'^\w|_|_', ' ', text)

    # Menghapus kata yang mengandung angka
    text = re.sub("\S*\d\S*", "", text).strip()

    # Menghapus angka yang berdiri sendiri
    text = re.sub(r"\b\d+\b", " ", text)

    # Mengubah huruf menjadi huruf kecil
    text = text.lower()
    text = re.sub('%s'% re.escape(string.punctuation), ' ',
text)
    # Menghapus spasi tambahan
    text = re.sub('[\s]+', ' ', text)

    # Menghapus karakter newline dan kata 'advertisement'
    text = re.sub('\n', '', text)
```

Segmen Program 4.1 (Lanjutan)

```
text = re.sub(r"//t",r"\t", text)
text = re.sub(r"( )\1+",r"\1", text)
text = re.sub(r"(\n)\1+",r"\1", text)
text = re.sub(r"(\r)\1+",r"\1", text)
text = re.sub(r"(\t)\1+",r"\1", text)
text = re.sub('advertisement', '', text)
# Menghapus stopwords
stopword = set(stopwords.words('indonesian'))
words = text.split()
words = [word for word in words if word not in stopword]

# Proses stemming
text = ' '.join([stemmers.stem(word) for word in words])

return text
```

4.4.2 Segmen Program *Feature Engineering*

Segmen program 4.2 merupakan *source code* untuk proses *feature engineering* dengan bahasa pemrograman python melalui google collab.

Segmen Program 4.2 *Feature Engineering*

```
class BERTEmbeddingsDataset(Dataset):
    def __init__(self, texts, tokenizer, max_len):
        self.texts = texts
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        text = self.texts[idx]
        encoding = self.tokenizer.encode_plus(
            text,
            add_special_tokens=True,
            max_length=self.max_len,
            return_token_type_ids=False,
            padding='max_length',
            return_attention_mask=True,
            return_tensors='pt',
            truncation=True,
        )
        return {
            'input_ids': encoding['input_ids'].flatten(),
            'attention_mask':
                encoding['attention_mask'].flatten()
```

Segmen Program 4.2 (Lanjutan)

```

    }
def create_data_loader(df, tokenizer, max_len, batch_size):
    ds = BERTEmbeddingsDataset(
        texts=df.cleaned.to_numpy(),
        tokenizer=tokenizer,
        max_len=max_len
    )
    return DataLoader(ds, batch_size=batch_size, num_workers=4)

data_loader = create_data_loader(df_new, tokenizer, max_len=128,
batch_size=16)
bert_model = AutoModel.from_pretrained("indobenchmark/indobert-
base-p1")
bert_model = bert_model.to(device)
def extract_embeddings(data_loader, model, device):
    model.eval()
    embeddings = []
    labels = []
    with torch.no_grad():
for d in tqdm(data_loader):
        input_ids = d['input_ids'].to(device)
        attention_mask = d['attention_mask'].to(device)

        outputs = model(input_ids=input_ids,
attention_mask=attention_mask)
        embeddings.append(outputs.last_hidden_state[:, 0,
:].cpu().numpy())

    embeddings = np.concatenate(embeddings, axis=0)
    return embeddings

```

4.4.3 Segmen Program *Training dan Evaluation*

Segmen 4.3 merupakan *source code* proses pelatihan dan evaluasi model *machine learning* yang dibuat dengan bahasa pemrograman python menggunakan google collab.

Segmen Program 4.3 *Training dan Evaluation*

```

# Training BERT
# pembagian dataset train/test
df_train, df_test = train_test_split(df_new, test_size=0.2,
random_state=1)
df_train, df_eval = train_test_split(df_train, test_size=0.2,
random_state=1)
# Convert to HuggingFace Datasets
hf_train = HFDataset.from_pandas(df_train)
hf_eval = HFDataset.from_pandas(df_eval)
hf_test = HFDataset.from_pandas(df_test)

```

Segmen Program 4.3 (Lanjutan)

```
dataset_full = DatasetDict({
    'train': hf_train,
    'eval': hf_eval,
    'test': hf_test
})

# Tokenize data
tokenizer =
AutoTokenizer.from_pretrained("indobenchmark/indobert-base-p1")

def tokenize_dataset(data):
    return tokenizer(data["text"], padding=True,
truncation=True, max_length=512)

dataset_full = dataset_full.map(tokenize_dataset, batched=True)
dataset_full = dataset_full.remove_columns(["text"])
# reload model BERT
model =
AutoModelForSequenceClassification.from_pretrained("indobenchmark/indobert-base-p1", num_labels=2)
model = model.to(device)
# Training arguments for BERT model
training_args = TrainingArguments(
    output_dir="results",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=16,
    num_train_epochs=5,
    weight_decay=0.01,
    logging_dir="./logs",
    load_best_model_at_end=True,
    metric_for_best_model="f1",
    push_to_hub=False,
    report_to="none"
)
# Pembuatan fungsi evaluasi untuk BERT model
def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    predictions = np.argmax(predictions, axis=1)
    precision, recall, f1, _ =
precision_recall_fscore_support(labels, predictions,
average='binary')
    accuracy = accuracy_score(labels, predictions)
    roc_auc = roc_auc_score(labels, predictions)
    return {
        'accuracy': accuracy,
        'precision': precision,
        'recall': recall,
        'f1': f1,
        'roc_auc': roc_auc
    }
# Train BERT model
```

Segmen Program 4.3 (Lanjutan)

```

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=dataset_full["train"],
    eval_dataset=dataset_full["eval"],
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)
# pemanggilan fungsi train untuk melatih model
trainer.train()
# pemanggilan fungsi evaluate untuk melakukan evaluasi model
pred_output
=trainer.evaluate(eval_dataset=dataset_full["test"])
# proses evaluasi matrix
pred_label_eval = np.argmax(pred_output_eval.predictions,
axis=1)
gt_label_eval = dataset_full["eval"]['label']
pred_output_eval.metrics

#proses training algoritma SVM dan Random forest
X_bert = extract_embeddings(data_loader, bert_model, device)

# Train SVM and Random Forest on BERT embeddings
X_train, X_test, y_train, y_test = train_test_split(X_bert,
y_new, test_size=0.2, random_state=42)
svm_model = SVC(probability=True,C=1,gamma= 0.1,
kernel='linear')
rf_model = RandomForestClassifier(n_estimators=100,
random_state=42)

svm_model.fit(X_train, y_train)
rf_model.fit(X_train, y_train)
svm_pred = svm_model.predict(X_test)
rf_pred = rf_model.predict(X_test)

# Evaluate models
def evaluate_model(y_true, y_pred, model_name):
    precision = precision_score(y_true, y_pred)
    recall = recall_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)
    accuracy = accuracy_score(y_true, y_pred)
    print(f"{model_name} Model - Precision: {precision}, Recall:
{recall}, F1: {f1}, Accuracy: {accuracy}")
    return precision, recall, f1, accuracy

svm_accuracy = accuracy_score(y_test, svm_pred)
rf_accuracy = accuracy_score(y_test, rf_pred)

vm_metrics = evaluate_model(y_test, svm_pred, "SVM")
rf_metrics = evaluate_model(y_test, rf_pred, "Random Forest")

# Confusion Matrices
bert_conf_matrix = confusion_matrix(gt_label, pred_label)
print(f'confusion matrix bert = {bert_conf_matrix}')

```

Segmen Program 4.3 (Lanjutan)

```
svm_conf_matrix = confusion_matrix(y_test, svm_pred)
print(f'confusion_matrix svm = {svm_conf_matrix}')
rf_conf_matrix = confusion_matrix(y_test, rf_pred)
print(f'confusion_matrix rf = {rf_conf_matrix}')
```

4.4.4 Segmen Program Implementasi Model

Segmen Program 4.4 merupakan *source code* implementasi model *machine learning* dengan *freamwork flask* berbasis bahasa pemrograman python.

Segmen Program 4. 4 Implementasi Model

```
from flask import Flask, render_template, request, jsonify,
redirect, url_for
from transformers import AutoTokenizer,
AutoModelForSequenceClassification
import torch
app = Flask(__name__)
# python -m flask run
# Load tokenizer and model
tokenizer =
AutoTokenizer.from_pretrained("indobenchmark/indobert-base-pl")
model =
AutoModelForSequenceClassification.from_pretrained("model")
model.eval()

def predict_text(text):
    inputs = tokenizer(text, return_tensors="pt", padding=True,
truncation=True)
    with torch.no_grad():
        outputs = model(**inputs)
    predicted_class = torch.argmax(outputs.logits, dim=1).item()

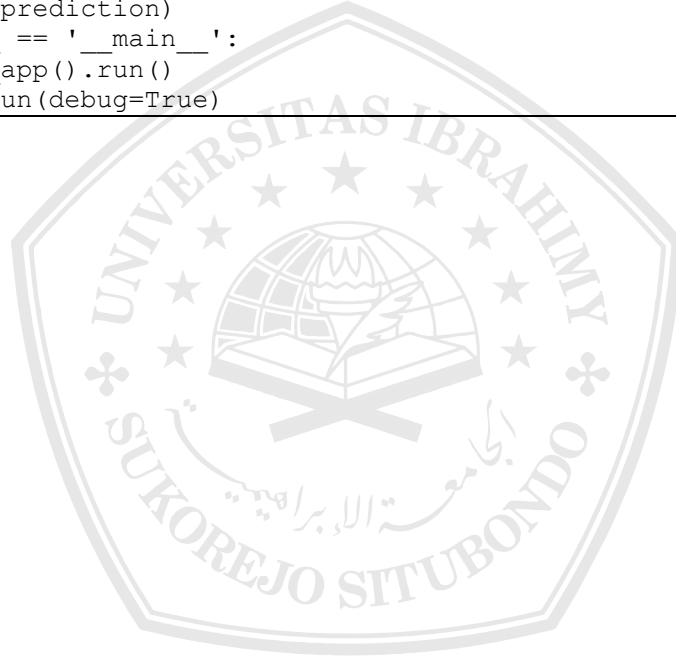
    # Mengubah hasil prediksi dari angka menjadi kategori
    if predicted_class == 0:
        prediction_result = "asli"
    else:
        prediction_result = "hoax"

    return prediction_result
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
    data = request.json
    text = data['text']
    predicted_class = predict_text(text)
```

Segmen Lanjutan 4.4

```
# Mengirim kembali hasil prediksi dalam format yang
diinginkan
if predicted_class == "asli":
    result = "asli"
else:
    result = "hoax"

return jsonify({'predicted_class': result, 'input_text':
text})
@app.route('/result', methods=['GET'])
def result():
    text = request.args.get('text')
    prediction = request.args.get('prediction')
    return render_template('result.html', text=text,
prediction=prediction)
if __name__ == '__main__':
    create_app().run()
    # app.run(debug=True)
```



BAB V

PENUTUP

5.1 Kesimpulan

Dapat disimpulkan hasil dari penelitian tentang deteksi berita *hoax* dengan melakukan pengujian menggunakan tiga metode BERT, SVM dan *Random Forest* untuk klasifikasi berita *hoax* menunjukkan performa yang sangat baik dan metode yang paling unggul adalah BERT dengan performa (akurasi = 0.99, ROC-AUC = 0.99) dikarenakan BERT merupakan algoritma berbasis *deep learning* yang mempunyai banyak *layer* sehingga dapat belajar dan menganalisa lebih dalam. Pada analisa teks dapat diketahui bahwa berita *hoax* sering menggunakan foto dan video yang dimanipulasi tidak sesuai dengan isi konten.

5.2 Saran

Untuk peneliti yang akan datang bisa melakukan *fine-tuning* untuk algoritma *machine learning* tradisional untuk meningkatkan performa dan juga berbagai metode yang lain.

DAFTAR PUSTAKA

- [1] N. N. A. Sri Andrian, "Literasi Digital Dalam Tindak Pidana Penyebaran Berita Bohong dan Menyesatkan Berdasarkan Undang-Undang Nomor 19 Tahun 2016 Tentang Informasi dan Transaksi Elektronik," *AMEENA JOURNAL*, pp. 340-350, 2023.
- [2] "BIRO KESEJAHTERAAN RAKYAT Provinsi Kepulauan Bangka Belitung," 01 November 2019. [Online]. Available: <https://birokesra.babelprov.go.id/content/ciri-informasi-atau-berita-hoax-yang-mudah-dikenali>. [Diakses 18 Mei 2024].
- [3] B. Mufarida, "SindoNews," 01 Maret 2024. [Online]. Available: <https://nasional.sindonews.com/read/1331869/15/dewan-pers-baru-1700-media-yang-sudah-terverifikasi-1709283799>. [Diakses 19 05 2024].
- [4] A. Zuhairah, "Penerapan Algoritma Random Forest, Support Vector Machines (SVM) dan Gradient Boosted Tree (GBT) Untuk Deteksi Penipuan (Fraud Detection) Pada Transaksi Kartu Kredit," Universitas Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, 2022.
- [5] P. K. Tipawan Silwattananusarn, "A text mining and topic modeling based bibliometric," *IAES International Journal of Artificial Intelligence (IJ-AI)*, pp. 1057-1065, 2022.
- [6] M.-W. C. K. L. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Language Understanding," dalam *HLT 2019-2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies - Proceedings of Conference*, 2019.
- [7] A. G. P. N. Rohit Kumar Kaliyar, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimedia Tools and Applications*, pp. 11765-11788, 2021.

- [8] J. F. R. Y. K. N. Y. Aisyah Awalina, "Indonesia's Fake News Detection using Transformer Network," dalam *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, Malang, 2021.
- [9] A. Jaedun, "METODOLOGI PENELITIAN," Fakultas Teknik UNY, DIY Yogyakarta, 2011.
- [10] M. Dr. Ubaid Ridlo, METODE PENELITIAN STUDI KASUS: TEORI DAN PRAKTIK, Jakarta: Publica Indonesia Utama, 2023.
- [11] A. WAHYUDI, "Perancangan Sistem Menggunakan Metode SDLDC," *Jurnal Dinamika Informatika*, pp. 1-11, 2018.
- [12] A. T. J. H. K. L. Isnaeni Imroatus Sholikhah, "Machine Learning Untuk Deteksi Berita Hoax Menggunakan BERT," Universitas PGRI Semarang, Semarang, 2023.
- [13] R. W. Edwi Arief Sosiawan, "Kontestasi Berita Hoax Pemilu Presiden Tahun 2019 di Media Daring dan Media Sosial," *Jurnal Ilmu Komunikasi*, pp. 133-142, 2019.
- [14] A. P. P. A. P. Rio Yunanto, "Survey Literatur Deteksi Berita Palsu Menggunakan Pendekatan Deep Learning," *Journal Manajemen Informatika*, pp. 188-130, 2021.
- [15] A. A. M. Hery Mustofa, "Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes," *Walisongo Journal of Information Technologi*, pp. 1-12, 2019.
- [16] J. K. Marc Moreno Lopez, "Deep Learning applied to NLP," 2017.
- [17] R. Saputra, "Implementasi Bidirectional Encoder Representations from Transformers (BERT) Untuk Mendeteksi Hatespeech dan Abusive Language Pada Twitter Bahasa Indonesia Jurusan Teknik Informatika UIN Suska Riau," UIN Suska Riau, Pekanbaru, 2022.

- [18] "School of Information System - Binus University," 18 Juli 2023. [Online]. Available: <https://sis.binus.ac.id/2023/07/18/mengenal-deep-learning-beserta-contoh-penerapannya/>. [Diakses 2024 Februari 27].
- [19] P. D. Kusuma, *Machine Learning Teori, Program, dan Studi Kasus*, Sleman: Deepublish Publisher, 2020.
- [20] L. F. I. M. F. A. Amal Mahmoud Yehia, "Text Mining and Knowledge Discovery from Big Data: Challenges and Promise," *IJCSI International Journal of Computer Science*, pp. 54-61, 2016.
- [21] d. Imrus Salehin, "AutoML: A systematic review on automated machine learning with neural architecture search," *Journal of Information and Intelligence*, pp. 52-81, 2024.
- [22] d. Matthias Schonlau, "The random forest algorithm for statistical learning," *The Stata Journal*, pp. 3-29, 2020.
- [23] J. A. M. L. Daniel Valero-Carreras, "Comparing two SVM models through different metrics based on the confusion matrix," *Computer & Operations Research*, pp. 1-12, 2023.
- [24] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, pp. 168-192, 2021.
- [25] d. M. Riziq sirfatullah Alfarizi, "Penggunaan Python Sebagai Bahasa Pemrograman Untuk Machine Learning Dan Deep Learning," *Karimah Tauhid*, vol. 2, no. 1, pp. 1-6, 2023.
- [26] R. G. Guntara, "Pelatihan Sains Data Bagi Pelaku UMKM di Kota Tasikmalaya Menggunakan Google Colab," *JOONG-KI : Jurnal Pengabdian Masyarakat*, vol. 2, no. 2, pp. 245-251, 2023.
- [27] T. Pramandyo, "Apa itu GPU? Akselerasi Kinerja AI Hingga Super Komputer," *Litasarta*, 10 Juni 2024. [Online]. Available: <https://www.lintasarta.net/blog/cloud-services/gpu/apa-itu-gpu-akselerasi-kinerja-ai-hingga-super-komputer>. [Diakses 26 Juni 2024].

- [28] P. O. N. S. Graciela Fausten Novindri, "Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang Di Distribution Center Pada Pt Xyz Berbasis Web," *Jurnal MNEMONIC*, vol. 5, no. 2, pp. 80-85, 2022.
- [29] "Implementasi Pemrograman Python Menggunakan Visual Studio Code," *Jurnal Ilmu Komputer*, vol. XI, no. 1, pp. 1-9, 2020.
- [30] "Membuat Diagram dengan Draw.io: Alat Online Gratis untuk Visualisasi Ide dan Konsep," Diginews, 5 April 2024. [Online]. Available: <https://diginews.id/membuat-diagram-dengan-draw-io-alat-online-gratis-untuk-visualisasi-ide-dan-konsep/>. [Diakses 3 Juli 2024].
- [31] ,. S. S. R. N. A. ,. F. M. Y. P. Ilham Budiman, "Analisis Pengendalian Mutu Di Bidang Industri Makanan," *Jurnal Inovasi Penelitian*, vol. 01, no. 10, pp. 2185-2190, 2021.
- [32] D. B. P. Daniel B. Paillin, "Rancangan Aplikasi Monitoring Online Untuk Meningkatkan Pemeliharaan Prediktif Pada PLTD," *Jurnal Sistem Informasi Bisnis*, pp. 9-16, 2020.

LAMPIRAN-LAMPIRAN

A. Hasil Implementasi Model

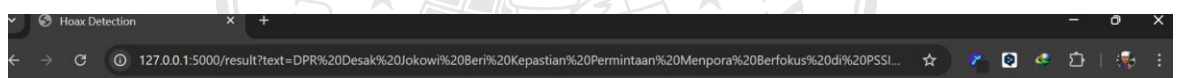


Sistem Prediksi Berita Hoax

Masukan Teks Berita

Prediksi Teks

© 2024 Asep Ripai



Sistem Prediksi Berita Hoax

Hasil Prediksi

Teks:

DPR Desak Jokowi Beri Kepastian Permintaan Menpora Berkokus di PSSI

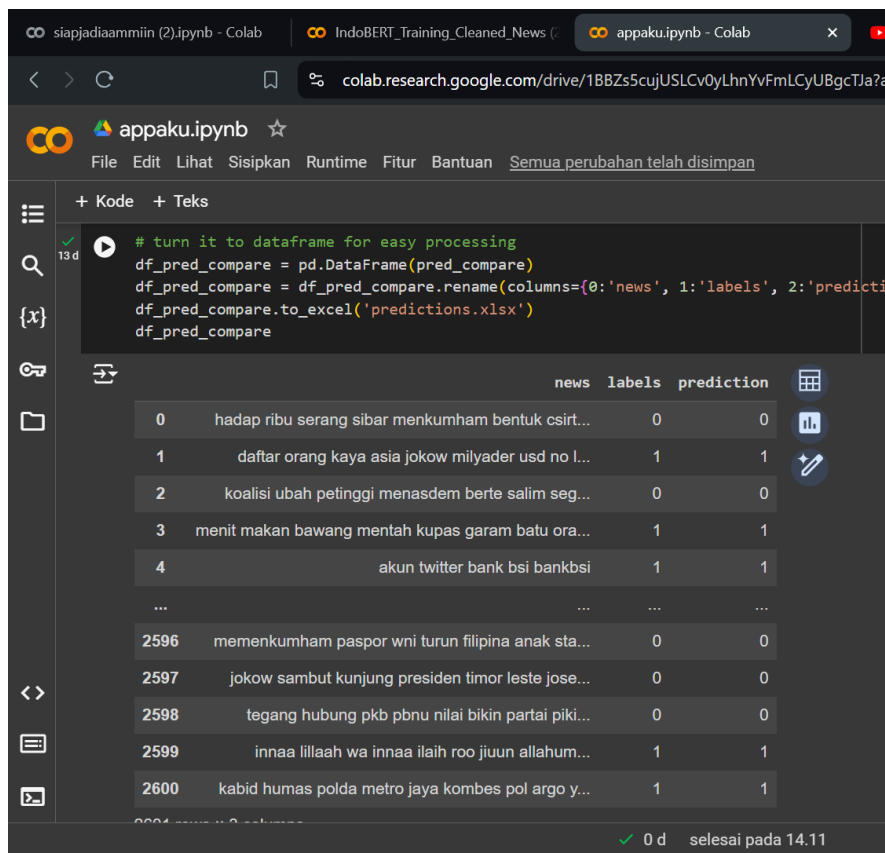
Prediksi:

asli

Reset

© 2024 Asep Ripai

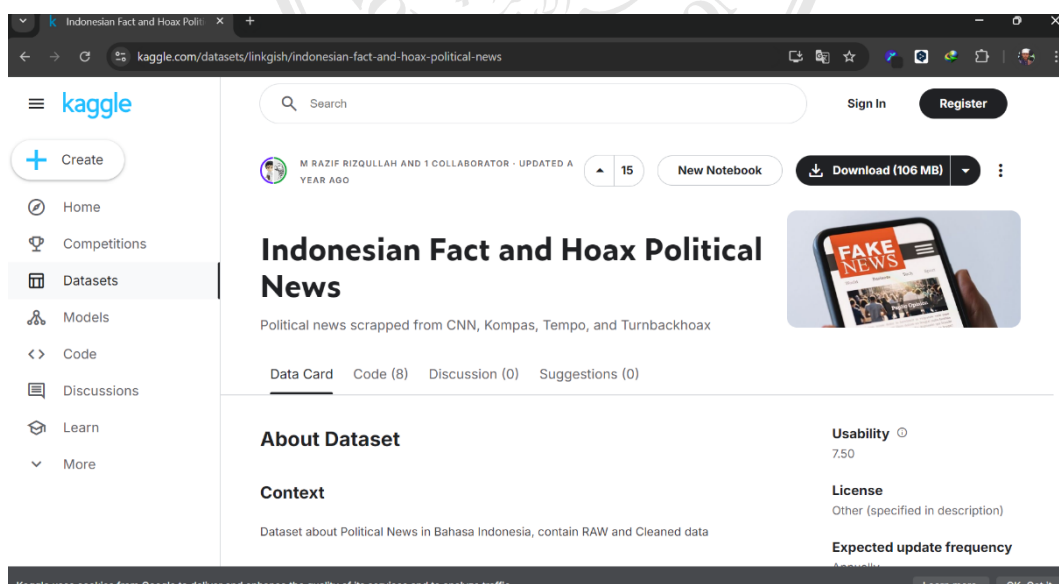
B. Hasil Prediksi Data Testing



```
# turn it to dataframe for easy processing
df_pred_compare = pd.DataFrame(pred_compare)
df_pred_compare = df_pred_compare.rename(columns={0:'news', 1:'labels', 2:'prediction'})
df_pred_compare.to_excel('predictions.xlsx')
df_pred_compare
```

	news	labels	prediction
0	hadap ribu serang sibar menkumham bentuk csirt...	0	0
1	daftar orang kaya asia jokow milyader usd no l...	1	1
2	koalisi ubah petinggi menasdem berte salim seg...	0	0
3	menit makan bawang mentah kupas garam batu ora...	1	1
4	akun twitter bank bsi bankbsi	1	1
...	...	...	...
2596	memenkumham paspor wni turun filipina anak sta...	0	0
2597	jokow sambut kunjung presiden timor leste jose...	0	0
2598	tegang hubung pkb pbnu nilai bikin partai piki...	0	0
2599	innaa lillaah wa innaa ilaih roo jiuun allahum...	1	1
2600	kabid humas polda metro jaya kombes pol argo y...	1	1

C. Sumber Data



Indonesian Fact and Hoax Political News

Political news scrapped from CNN, Kompas, Tempo, and Turnbackhoax

About Dataset

Context

Dataset about Political News in Bahasa Indonesia, contain RAW and Cleaned data

Usability

7.50

License

Other (specified in description)

Expected update frequency

Annually

CURRICULUM VITAE



Nama : Asep Ripa'i
NPM : 2020503011
Tempat, Tanggal Lahir : Subang, 29 April 2001
Program Studi : Teknologi Informasi
Orang Tua

Ayah : Idi

Ibu : Nunung

Latar Belakang Pendidikan

SD/MI : MI Al-Insyroh

SLTP : Mts Al-Mubarak

SLTA : Ma Al-Husna

Latar Organisasi :-

Alamat Rumah : Kampung Mayang Rt 02/Rw 01 Desa
Mayang Kec. Cisalak Kab
Subang, Jawa Barat

No Telpon : 087763135407

E-mail : Asepkhabril20@gmail.com