

**PENDETEKSI BERITA PALSU MENGGUNAKAN MODEL ARTIFICIAL NEURAL
NETWORK DENGAN PYTORCH**



OLEH

MADE WIRADARMA

(2005541029)

I WAYAN YOGA ADI PRANATA

(2005541075)

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS UDAYANA
TAHUN 2023**

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
BAB II TINJAUAN PUSTAKA	3
2.1 Berita	3
2.2 Pytorch	3
2.3 Natural Language Processing.....	3
2.4 Artificial Neural Network	4
2.5 Rectified Linear Unit.....	4
2.6 Adam Optimizer.....	4
2.7 F1 Score.....	5
2.8 Confusion Matrix	5
BAB III METODE PENELITIAN	6
3.1 Gambaran Umum	6
3.2 Situasi dan Kondisi Pembuatan Aplikasi	6
3.2.1 Spesifikasi Laptop	6
3.2.2 Bahasa Pemrograman yang Digunakan.....	7
3.2.3 Environment Untuk Membuat Aplikasi	7
3.3 Akuisisi Data	8
3.4 Persiapan Data.....	8
3.5 Pra-Pemrosesan Data.....	8
3.6 Pembuatan Model.....	9
3.7 Pelatihan dan Pengujian Model.....	9
3.8 Evaluasi Model.....	9

BAB IV PEMBAHASAN	10
4.1 Hasil Akuisisi Data.....	10
4.2 Hasil Persiapan Data	11
4.3 Hasil Pra-Pemrosesan Data	12
4.4 Hasil Pembuatan Model	17
4.5 Hasil Pelatihan dan Pengujian Model	18
4.6 Evaluasi Model.....	20
BAB V PENUTUP.....	22
5.1 Kesimpulan.....	22
5.2 Saran.....	22
DAFTAR PUSTAKA.....	23
LAMPIRAN.....	24

BAB I

PENDAHULUAN

1.1 Latar Belakang

Akses internet yang semakin mudah mendorong kemajuan teknologi di berbagai bidang, salah satunya berdampak pada akses informasi yang menjadi lebih cepat. Hal ini juga berdampak pada peningkatan penggunaan media internet sebagai sarana informasi khususnya berita online. Namun, dibalik cepatnya informasi yang bisa diakses, muncul fenomena berita palsu yang meresahkan masyarakat. Berita palsu bisa dengan mudah menyebar dengan memanfaatkan kecepatan dan keterbukaan internet.

Berita palsu merupakan sebuah informasi yang disajikan tidak sesuai dengan fakta yang terjadi di lapangan. Berita palsu disebarkan oleh pihak yang tidak bertanggung jawab dengan tujuan untuk mempengaruhi opini public atau mendapatkan keuntungan pribadi. Dalam beberapa kasus, berita palsu telah menyebabkan kerugian yang signifikan terhadap individu maupun masyarakat secara keseluruhan.

Menghadapi masalah ini, diperlukan upaya untuk mengidentifikasi dan mengatasi berita palsu dengan cepat dan efektif. Metode konvensional yang digunakan untuk memeriksa keaslian berita sering kali kurang efisien dan tidak dapat mengatasi skala masalah yang ada. Oleh karena itu, penggunaan machine learning menjadi solusi yang menjanjikan dalam mendeteksi berita palsu. Machine learning memungkinkan komputer untuk mempelajari data yang ada dan mengenali pola-pola kompleks. Dalam pendekatan pendeteksian berita palsu, model machine learning dilatih menggunakan dataset yang mengandung berita palsu dan berita asli untuk menemukan karakteristik dan pola yang membedakan keduanya.

Dalam penelitian ini, digunakan metode artificial neural network (ANN) dengan framework PyTorch. ANN dapat memproses informasi teks, gambar, dan metadata lainnya untuk memperoleh pemahaman yang lebih mendalam tentang keaslian suatu berita. Keunggulan utama dari pendekatan ini adalah kemampuan untuk memperbarui dan meningkatkan model dengan dataser yang lebih baru, sehingga dapat terus menghadapi strategi baru yang digunakan dalam menyebarkan berita palsu. Sedangkan, PyTorch digunakan karena memberikan keuntungan dalam hal fleksibilitas, skalabilitas, dan kinerja yang baik. PyTorch memungkinkan pengembang untuk membangun dan melatih model neural network dengan mudah, sembari memberikan fleksibilitas dalam mengatur arsitektur model dan memanipulasi data.

Dengan berbagai penjelasan di atas, pendeteksi berita palsu menggunakan machine learning dengan model artificial neural network dengan PyTorch menjadi topik yang menarik dan relevan untuk dibahas. Pengembangan pendekatan ini diharapkan mampu membantu memerangi penyebaran berita palsu, mempromosikan informasi yang akurat, dan meningkatkan kepercayaan terhadap sumber berita di era digital yang semakin kompleks ini.

1.2 Rumusan Masalah

Dari latar belakang di atas, didapatkan beberapa rumusan masalah sebagai berikut.

- 1.2.1 Bagaimana cara kerja pendeteksi berita palsu menggunakan model artificial neural network dengan PyTorch?
- 1.2.2 Bagaimana cara melakukan standarisasi dan pengolahan data dalam proses pendeteksian berita palsu menggunakan model artificial neural network?
- 1.2.3 Bagaimana cara melakukan training, test, dan validasi model artificial neural network dalam pendeteksian berita palsu menggunakan metode K-Fold Cross Validation?
- 1.2.4 Bagaimana hasil, keefektifan, dan evaluasi sistem pendeteksi berita palsu menggunakan metode artificial neural network?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

- 1.3.1 Mengetahui cara kerja pendeteksi berita palsu menggunakan model artificial neural network dengan PyTorch.
- 1.3.2 Mengetahui cara melakukan standarisasi dan pengolahan data dalam proses pendeteksian berita palsu menggunakan model artificial neural network.
- 1.3.3 Mengetahui cara melakukan training, test, dan validasi model artificial neural network dalam pendeteksian berita palsu menggunakan metode K-Fold Cross Validation.
- 1.3.4 Mengetahui hasil, keefektifan, dan evaluasi sistem pendeteksi berita palsu menggunakan metode artificial neural network.

BAB II

TINJAUAN PUSTAKA

2.1 Berita

Berita adalah informasi atau laporan terkini yang disebarakan melalui berbagai media seperti televisi, radio, internet, situs web, dan lainnya. Berita berisi informasi baru atau peristiwa yang sedang terjadi. Para pakar jurnalistik mendefinisikan berita sebagai apa yang ditulis di surat kabar, disiarkan di radio, dan ditayangkan di televisi. Setiap berita memiliki nilai berita (news value) yang menentukan apakah peristiwa tersebut layak diberitakan atau tidak. Terdapat 2 jenis berita secara umum, yaitu real news dan fake news. Real news atau berita asli adalah jenis berita yang didasarkan pada fakta-fakta yang dapat diverifikasi dan informasi yang benar-benar terjadi. Berita ini disusun dengan prinsip jurnalisme yang melibatkan pengumpulan informasi dari sumber yang dapat dipercaya, verifikasi data, dan pendekatan yang objektif. Berita real memberikan laporan tentang peristiwa aktual, informasi penting, dan perkembangan terkini yang terjadi di berbagai bidang, seperti politik, ekonomi, olahraga, hiburan, dan lainnya. Sedangkan Fake News adalah istilah yang digunakan untuk menggambarkan berita palsu atau disinformasi yang disebarakan melalui media massa atau platform online. Berita palsu ini tidak memiliki dasar fakta yang valid atau sudah terbukti tidak benar. Contohnya adalah ketika berita tentang Paus mendukung Donald Trump pada tahun 2016 menyebar melalui media online.

2.2 Pytorch

PyTorch merupakan sebuah library open source untuk deep learning yang dikembangkan oleh tim peneliti Facebook AI Research (FAIR). Framework ini pertama kali dirilis pada tahun 2016. PyTorch memudahkan para peneliti dan pengembang dalam membangun dan melatih jaringan saraf (neural networks) menggunakan bahasa pemrograman Python. Kelebihan PyTorch adalah fleksibilitas dan modularitasnya yang memudahkan dalam penelitian, namun tetap stabil dan dapat digunakan untuk produksi. Salah satu fitur yang membedakan PyTorch dari kerangka kerja machine learning lainnya adalah Dynamic Computational Graphs yang memungkinkan fleksibilitas yang lebih tinggi dalam pembuatan model.

2.3 Natural Language Processing

Natural Language Processing (NLP) adalah teknologi dalam bidang pembelajaran mesin yang memungkinkan komputer untuk memahami, memanipulasi, dan menganalisis bahasa manusia secara otomatis. NLP merupakan cabang kecerdasan buatan (AI) yang berfokus pada

interaksi antara komputer dan bahasa manusia. Tujuannya adalah membuat komputer mampu "memahami" isi dokumen dan konteks bahasa alami.

NLP digunakan dalam berbagai aplikasi dunia nyata, termasuk penelitian medis, mesin pencari, dan bisnis intelijen. Tantangan dalam NLP meliputi pengenalan ucapan, pemahaman bahasa alami, dan generasi bahasa alami. NLP juga berguna dalam pembuatan fake news detection, chatbot, penerjemahan bahasa, dan analisis sentimen.

2.4 Artificial Neural Network

Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan adalah teknik pengolahan informasi yang terinspirasi oleh sistem saraf biologis. Dalam ANN, neuron-neuron saling terhubung untuk menyelesaikan masalah klasifikasi atau prediksi. ANN terdiri dari lapisan-lapisan neuron, seperti input layer, hidden layer, dan output layer. Saat dilatih, ANN memperbarui bobotnya berdasarkan perbandingan antara keluaran yang dihasilkan dan yang diharapkan.

2.5 Rectified Linear Unit

Rectifier Linear Unit (ReLU) adalah salah satu jenis fungsi aktivasi yang digunakan dalam jaringan saraf tiruan (ANN) untuk mempercepat proses pembelajaran dan meningkatkan kinerja model. Fungsi ReLU menghasilkan keluaran nol jika masukan bernilai negatif, dan menghasilkan keluaran sama dengan masukan jika masukan bernilai positif. Beberapa penelitian dalam bidang deteksi berita palsu menggunakan fungsi ReLU dalam ANN untuk meningkatkan akurasi model. Sebagai contoh, sebuah penelitian yang dilakukan pada tahun 2022 menggunakan ReLU activation based multilayer perceptron (MLP) dalam pengolahan data untuk deteksi berita palsu. Dalam fake news detection, penggunaan ReLU dalam jaringan saraf tiruan dapat membantu meningkatkan akurasi model dan mempercepat proses pembelajaran. Beberapa penelitian telah menggunakan ReLU dalam ANN untuk deteksi berita palsu dan mencapai hasil yang cukup baik.

2.6 Adam Optimizer

Adam optimizer adalah sebuah algoritma optimasi yang sering digunakan dalam deep learning untuk memperbarui bobot jaringan saraf tiruan selama proses pelatihan. Algoritma ini menggabungkan estimasi momen orde pertama dan kedua untuk mengatur laju pembelajaran secara adaptif untuk setiap bobot dalam jaringan. Keunggulan Adam adalah kemampuannya

untuk mengatasi masalah gradien yang menghilang atau meledak, yang sering terjadi pada algoritma optimasi lainnya. Adam memiliki beberapa parameter yang dapat dikonfigurasi, seperti learning rate, beta1, beta2, dan epsilon. Algoritma ini telah diterapkan dalam berbagai aplikasi deep learning, termasuk dalam deteksi berita palsu.

2.7 F1 Score

F1 score adalah sebuah metrik evaluasi dalam machine learning yang digunakan untuk mengukur kinerja model klasifikasi biner. F1 score menggabungkan precision dan recall dari model menjadi satu metrik, sehingga memberikan gambaran yang lebih baik tentang kinerja model. F1 score memberikan nilai antara 0 dan 1, di mana nilai 1 menunjukkan kinerja model yang sempurna dan nilai 0 menunjukkan kinerja model yang buruk. F1 score sering digunakan dalam deteksi berita palsu untuk mengukur kinerja model dalam mengklasifikasikan berita sebagai palsu atau benar.

2.8 Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam machine learning. Confusion matrix menampilkan jumlah data yang diklasifikasikan dengan benar dan salah oleh model, serta jenis kesalahan yang dilakukan oleh model. Terdiri dari empat sel, yaitu true positive (TP), true negative (TN), false positive (FP), dan false negative (FN). TP adalah jumlah data positif yang diklasifikasikan dengan benar, TN adalah jumlah data negatif yang diklasifikasikan dengan benar, FP adalah jumlah data negatif yang salah diklasifikasikan sebagai positif, dan FN adalah jumlah data positif yang salah diklasifikasikan sebagai negatif. Confusion matrix dapat digunakan untuk menghitung metrik evaluasi seperti precision, recall, dan F1 score. Selain itu, confusion matrix membantu memahami kinerja model dan mengidentifikasi area yang perlu diperbaiki.

BAB III

METODE PENELITIAN

3.1 Gambaran Umum

Fake News adalah istilah yang digunakan untuk menggambarkan berita palsu atau disinformasi yang disebarluaskan melalui media massa atau platform online. Berita palsu ini tidak memiliki dasar fakta yang valid atau sudah terbukti tidak benar. Contohnya adalah ketika berita tentang Paus mendukung Donald Trump pada tahun 2016 menyebar melalui media online.

Penelitian yang dilakukan oleh Massachusetts Institute of Technology (MIT) menyimpulkan bahwa informasi palsu menyebar lebih cepat daripada informasi yang benar dan akurat. Ini bisa disebabkan oleh berbagai faktor, seperti sensasi atau kontroversi yang terkait dengan berita palsu, kecenderungan orang untuk membagikan berita yang sesuai dengan keyakinan mereka, atau algoritma media sosial yang mendorong penyebaran berita viral tanpa memeriksa kebenarannya.

Penting untuk selalu memverifikasi dan memastikan kebenaran informasi sebelum mempercayainya atau membagikannya kepada orang lain. Mengandalkan sumber berita yang terpercaya dan kritis terhadap informasi yang diterima adalah langkah penting untuk mengatasi masalah berita palsu dan menjaga keakuratan informasi yang diterima.

Penelitian ini bertujuan untuk mengembangkan pendeteksi berita palsu menggunakan artificial neural network (ANN) dengan framework PyTorch. Pendeteksi ini akan dilatih menggunakan dataset yang terdiri dari dataset berita asli dan berita palsu untuk mengenali pola dan karakteristik yang membedakan keduanya. Sebelum diolah, data di dalam dataset akan diberi label 0 dan 1 untuk membedakan berita palsu dengan berita asli. Setelah itu, dilanjutkan dengan pra-pemrosesan data yang terdiri dari beberapa tahapan. Kemudian, data yang telah melewati pra-pemrosesan data akan dilatih dan diuji menggunakan model artificial neural network. Selanjutnya, hasil pelatihan dan pengujian tadi akan dievaluasi dan dinilai keefektifannya.

3.2 Situasi dan Kondisi Pembuatan Aplikasi

3.2.1 Spesifikasi Laptop

Penelitian ini menggunakan dua buah laptop dengan spesifikasi yang berbeda. Adapun spesifikasi laptop pertama yang digunakan untuk penelitian Pendeteksi Berita Palsu Menggunakan Model Artificial Neural Network dengan Pytorch adalah sebagai berikut.

Proscessor	: Intel(R) Core (TM) i7-9750H
Operating System	: Windows 11 64 bit
RAM	: 16 GB
Graphics Card	: NVIDIA GeForce GTX 1050 TI Max-Q Design
HDD	: 1TB
SSD	: 512GB

Sedangkan, spesifikasi laptop kedua yang digunakan untuk penelitian Pendeteksi Berita Palsu Menggunakan Model Artificial Neural Network dengan Pytorch adalah sebagai berikut.

Proscessor	: Intel(R) Core (TM) i7-9750H
Operating System	: Windows 10 Home Single Language 64-bit
RAM	: 16 GB
Graphics Card	: NVIDIA GeForce GTX 1660 Ti with Max-Q Design
HDD	: 1TB
SSD	: 250GB

3.2.2 Bahasa Pemrograman yang Digunakan

Bahasa pemrograman yang digunakan dalam pembuatan aplikasi ini adalah bahasa pemrograman Python. Python sendiri merupakan salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax. Sebagian lain mengartikan Python sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Walaupun Python tergolong bahasa pemrograman dengan level tinggi, nyatanya Python dirancang sedemikian rupa agar mudah dipelajari dan dipahami.

3.2.3 Environment Untuk Membuat Aplikasi

Dalam pembuatan aplikasi ini menggunakan dua environment, yaitu Jupyter Notebook. Jupyter notebook adalah sebuah aplikasi web yang bersifat open-source yang diperuntukan dalam data cleaning dan transformasinya, simulasi angka, visualisasi data, pemodelan statistic, machine learning dll. Jupyter Notebook merupakan singkatan dari tiga bahasa pemrograman, yakni Julia (Ju), Python (Py), dan R. Ia adalah sebuah aplikasi web

gratis yang paling banyak dipakai oleh data scientist. Aplikasi ini dipakai untuk membuat dan membagikan dokumen yang memiliki kode, hasil hitungan, visualisasi, dan teks. Ketiga bahasa pemrograman pada Jupyter Notebook sendiri adalah sesuatu yang penting bagi seorang data scientist.

3.3 Akuisisi Data

Akuisisi data merupakan tahap awal dalam pembuatan pendeteksi berita palsu. Pada tahap ini, data yang diperlukan untuk melatih dan menguji model artificial neural network dikumpulkan dan dimuat ke dalam environment. Data yang digunakan dalam tahap ini sudah disediakan melalui Google Drive. Data terdiri dari dua dataset yang berisikan berita palsu dan berita asli. Di dalam dataset terdapat berbagai macam berita dari berbagai sumber.

3.4 Persiapan Data

Setelah data berhasil dimuat ke dalam environment, selanjutnya data tersebut akan melalui proses persiapan data. Persiapan data merupakan tahap penting dalam pembuatan pendeteksi berita palsu. Pada tahap ini, data yang telah dimuat akan diberi label 0 dan 1. Label 0 menandakan bahwa berita tersebut palsu dan label 1 menandakan bahwa berita tersebut asli. Setelah kedua dataset diberi label, langkah selanjutnya adalah menggabungkan kedua dataset menjadi satu. Setelah itu, dataset akan dilihat keseimbangan datanya serta berapa banyak kelas subjeknya.

3.5 Pra-Pemrosesan Data

Pra-Pemrosesan data merupakan proses di mana data yang telah disiapkan akan diolah agar bisa digunakan untuk pelatihan dan pengujian. Proses ini terdiri dari beberapa tahapan, dimulai dari penghapusan kolom yang tidak diperlukan, pembersihan data, normalisasi data, stemming, lemmatization, dan pembagian data. Hal tersebut dilakukan agar data yang akan digunakan dalam pelatihan dan pengujian sudah terstruktur, relevan, dan siap untuk membangun model yang akurat dalam mendeteksi berita palsu.

3.6 Pembuatan Model

Pembuatan model merupakan langkah yang dilakukan untuk pengembangan model untuk pendeteksi berita palsu. Tahapan ini bertujuan untuk membangun model yang dapat mempelajari pola-pola kompleks dalam data dan dapat mengklasifikasikan berita palsu dengan akurat. Pada tahap ini diharapkan model dapat memberikan kontribusi yang signifikan.

3.7 Pelatihan dan Pengujian Model

Setelah membuat model, langkah selanjutnya yang dilakukan adalah melatih dan menguji model menggunakan dataset yang sudah melalui tahap pra-pemrosesan data. Pada tahap ini, data yang sudah diolah akan dijadikan input untuk melatih model agar bisa mengklasifikasikan berita menjadi berita palsu atau berita asli.

3.8 Evaluasi Model

Evaluasi model merupakan tahapan setelah model selesai dilatih. Pada tahap ini dilakukan evaluasi untuk mengukur performa dan akurasi model dalam mendeteksi berita palsu. Metric evaluasi seperti confusion matrix, akurasi, presisi, recall, dan F1-score akan digunakan untuk mengevaluasi performa model.

BAB IV

PEMBAHASAN

4.1 Hasil Akuisisi Data

Pada tahap ini apabila sudah mendapatkan dua dataset yaitu berita palsu dan berita asli, selanjutnya dataset tersebut diunggah ke environment yang digunakan, yaitu Jupyter Notebook. Kemudian import dataset yang digunakan dan lihat informasi tentang struktur dan tipe data yang digunakan pada dataset tersebut.

```
In [2]: #Membaca Dataset

asli = pd.read_csv("True.csv")
palsu = pd.read_csv("Fake.csv")
```

Script yang tersebut menggunakan library Pandas untuk membaca dua file CSV yang disebut "True.csv" dan "Fake.csv" ke dalam dua objek DataFrame yang disebut "asli" dan "palsu".

```
In [3]: #Mendapatkan informasi tentang struktur dan tipe data dalam Dataset 'asli'

asli.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   title       21417 non-null  object
1   text        21417 non-null  object
2   subject     21417 non-null  object
3   date        21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB
```

```
In [4]: ##Mendapatkan informasi tentang struktur dan tipe data dalam Dataset 'palsu'

palsu.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   title       23481 non-null  object
1   text        23481 non-null  object
2   subject     23481 non-null  object
3   date        23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

Dari informasi di atas, terlihat bahwa dataset berita asli dan palsu memiliki sebuah DataFrame menggunakan library Pandas. DataFrame adalah struktur data tabular yang digunakan untuk menyimpan dan mengelola data dalam bentuk kolom dan baris.

4.2 Hasil Persiapan Data

Jika sudah berhasil diimport, langkah selanjutnya adalah memberikan label pada setiap dataset. Label digunakan untuk membedakan dan memberi nilai antara berita palsu dengan berita asli. Berita palsu diberi nilai 0 dan berita asli diberi nilai 1. Nilai pada label menggunakan type integer.

```
In [5]: #menambahkan kolom "Label" pada Data asli dan palsu dengan nilai yang menandakan  
asli["label"] = np.ones(len(asli),dtype=int)  
palsu["label"] = np.zeros(len(palsu),dtype=int)
```

Dalam baris pertama, `pd.read_csv("True.csv")`, file CSV dengan nama "True.csv" dibaca menggunakan fungsi `read_csv` dari Pandas. Data dari file CSV tersebut dimuat ke dalam objek DataFrame yang disimpan dalam variabel "asli". Begitu pula dengan dataset Fake.csv, dimana data dari file CSV tersebut dimuat ke dalam objek DataFrame yang disimpan dalam variabel "palsu".

```
In [6]: #menampilkan beberapa baris pertama dari Data 'asli'  
asli.head()
```

Out[6]:

	title	text	subject	date	label
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1

```
In [7]: #menampilkan beberapa baris pertama dari Data 'fake'  
palsu.head()
```

Out[7]:

	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn't wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0

Metode `head()` secara default akan menampilkan 5 baris teratas dari DataFrame. Jika ingin menampilkan jumlah baris yang berbeda, hal itu bisa ditentukan melalui metode `head()`. Misalnya, `asli.head(10)` akan menampilkan 10 baris teratas dari DataFrame "asli".

4.3 Hasil Pra-Pemrosesan Data

Proses ini dimulai dengan mengubah kolom `subject` menjadi variable dummy agar dapat digunakan dalam pemodelan lebih lanjut.

```
In [12]: #data = pd.get_dummies(data, columns=["subject"]): Baris ini digunakan untuk m
#get_dummies() digunakan untuk menghasilkan setiap nilai dalam fitur "subject"

data = pd.get_dummies(data, columns=["subject"])
data.head()
```

Out[12]:

	title	text	date	label	subject_Government News	subject_Middle- east
10588	TRUMP'S LAWYER DESTROYS #ChrisWallace In Heate...	Let s just get this out of the way Chris Walla...	Jun 18, 2017	0	0	0
16330	FANTASTIC! TRUMP'S 7 POINT PLAN To Reform Heal...	Since March of 2010, the American people have ...	Nov 1, 2016	0	1	0
191	Democrat wins U.S. Senate seat in Alabama in s...	(Reuters) - Democrat Doug Jones won Alabama's ...	December 13, 2017	1	0	0

Script di atas menggunakan library `pandas (pd)` dan terdapat fungsi `pd.get_dummies()` yang berfungsi untuk mengubah kolom tertentu dalam dataframe menjadi kolom-kolom baru yang berisi nilai-nilai biner (1 atau 0). Misalkan pada script diatas kolom yang diubah adalah kolom 'subject'. Jadi setiap nilai unik di kolom tersebut akan dipecah menjadi kolom-kolom baru dengan nama kolom yang sesuai dengan nilai unik tersebut. Setiap baris dalam dataframe akan memiliki nilai 1 di kolom yang sesuai dengan nilai unik di kolom "subject" dan nilai 0 di kolom-kolom yang lain.

Hapus Kolom Date

Selanjutnya hapus kolom date karena kolom tersebut tidak diperlukan dalam pemrosesan selanjutnya dan tidak memberikan informasi yang relevan.

```
In [13]: #data = data.drop("date", axis=1): Baris ini digunakan untuk menghapus kolom "
#axis=1 menunjukkan menghapus kolom (axis 1) bukan baris
#data = data.drop("date", axis=1) adalah untuk menghapus kolom "date" dari Dat

data = data.drop("date", axis=1)
data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 44898 entries, 10588 to 22035
Data columns (total 11 columns):
#   column                                Non-Null Count  Dtype
---  -
0   title                                44898 non-null  object
1   text                                44898 non-null  object
2   label                                44898 non-null  int32
3   subject_Government News             44898 non-null  uint8
4   subject_Middle-east                 44898 non-null  uint8
5   subject_News                        44898 non-null  uint8
6   subject_US_News                     44898 non-null  uint8
7   subject_left-news                   44898 non-null  uint8
8   subject_politics                    44898 non-null  uint8
9   subject_politicsNews                 44898 non-null  uint8
10  subject_worldnews                    44898 non-null  uint8
dtypes: int32(1), object(2), uint8(8)
memory usage: 1.5+ MB
```

Langkah selanjutnya, yaitu menghapus kolom 'date' karena kolom tersebut tidak diperlukan dalam pemrosesan data selanjutnya dan tidak memberikan informasi yang relevan. Script di atas digunakan untuk menghapus kolom "date" dari dataframe data dan kemudian menampilkan informasi tentang dataframe tersebut menggunakan fungsi data.info().

Natural Language Processing

Pemrosesan bahasa alami. Pada langkah ini akan memproses fitur teks dan label.

Import nltk dan download modul punkt yang berguna untuk tokenisasi data.

```
In [14]: #import nltk digunakan untuk mengimpor pustaka Natural Language Toolkit (NLTK)
#nltk.download('punkt') digunakan untuk mengunduh data yang diperlukan oleh NLTK
import nltk
nltk.download('punkt')
print("Unduhan Selesai")

Unduhan Selesai

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\MSI\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
```

Script di atas menggunakan library NLTK (Natural Language Toolkit). NLTK merupakan kumpulan library dan program untuk pemrosesan NLP. Pada script di atas terdapat fungsi 'nltk.download('punkt') untuk mengunduh dataset berisikan model yang diperlukan oleh NLTK dalam tokenization (pemisahan kata).


```
In [15]: #nltk.download('wordnet') digunakan untuk mengunduh data yang diperlukan oleh
import nltk
nltk.download('wordnet')
print("Unduhan Selesai")
```

Unduhan Selesai

```
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\MSI\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
```

Kemudian terdapat fungsi `nltk.download('wordnet')` untuk mengunduh dataset yang berisi WordNet, yaitu kamus bahasa inggris yang digunakan dalam berbagai tugas Natural Language Processing seperti melakukan lemmatisasi kata serta mencari sinonim dan antonim.

```
new_text = []
pattern = "[^a-zA-Z]"
lemma = nltk.WordNetLemmatizer()

for txt in data.text:

    txt = re.sub(pattern, " ",txt) # Cleaning
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt) # Tokenizing
    txt = [lemma.lemmatize(word) for word in txt] # Lemmatizing
    txt = " ".join(txt)
    new_text.append(txt)

new_text[0]
```

Selanjutnya dilakukan pemrosesan teks dalam kolom "text" dari DataFrame data dengan langkah-langkah pembersihan (cleaning), konversi ke huruf kecil (lowercasing), tokenisasi kata (word tokenization), lemmatisasi, dan penggabungan kembali teks yang telah diproses. Hasilnya disimpan dalam list `new_text`, dan `new_text[0]` digunakan untuk mencetak teks yang telah diproses pertama dari list tersebut.

```
# kode ini adalah untuk memproses judul (title) dalam kolom "title"
new_title = []
for txt in data.title:

    txt = re.sub(pattern, " ",txt) # Cleaning
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt) # Tokenizing
    txt = [lemma.lemmatize(word) for word in txt] # Lemmatizing
    txt = " ".join(txt)
    new_title.append(txt)

new_title[0]
```

```
Out[17]: 'trump s lawyer destroys chriswallace in heated exchange no chris i deal with
fact and law video'
```

Selanjutnya dilakukan pemrosesan teks dalam kolom title dari data mulai dari pembersihan (cleaning), konversi ke huruf kecil (lowercasing), tokenisasi kata (word tokenization), lemmatisasi, dan penggabungan kembali teks yang telah diproses. Kemudian hasilnya disimpan dalam list new_title.

```
#countVectorizer untuk mengubah teks judul (title) dan teks utama (text) menjadi vektor
#kode ini adalah untuk menghasilkan matriks representasi vektor untuk teks judul
vectorizer_title = CountVectorizer(stop_words="english",max_features=1000)
vectorizer_text = CountVectorizer(stop_words="english",max_features=4000)

title_matrix = vectorizer_title.fit_transform(new_title).toarray()
text_matrix = vectorizer_text.fit_transform(new_text).toarray()

print("Selesai")
```

Kode ini digunakan untuk menghasilkan matriks representasi vektor untuk teks judul (title) dan teks utama (text) berdasarkan frekuensi kemunculan kata-kata yang telah diproses sebelumnya menggunakan CountVectorizer.

```
data.drop(["title","text"],axis=1,inplace=True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44898 entries, 10588 to 22035
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   label                                44898 non-null  int32
1   subject_Government News              44898 non-null  uint8
2   subject_Middle-east                  44898 non-null  uint8
3   subject_News                         44898 non-null  uint8
4   subject_US_News                      44898 non-null  uint8
5   subject_left-news                    44898 non-null  uint8
6   subject_politics                     44898 non-null  uint8
7   subject_politicsNews                 44898 non-null  uint8
8   subject_worldnews                    44898 non-null  uint8
dtypes: int32(1), uint8(8)
memory usage: 876.9 KB
```

Script di atas digunakan untuk menghapus kolom "title" dan "text" dari DataFrame serta mencetak informasi ringkas tentang DataFrame setelah penghapusan.

```
In [20]: #melihat dimensi dari DataFrame 'data', t
print(data.shape)
print(title_matrix.shape)
print(text_matrix.shape)

(44898, 9)
(44898, 1000)
(44898, 4000)
```

Setelah itu, pastikan pemrosesan dan transformasi teks telah dilakukan dengan benar, serta untuk memahami ukuran data yang akan digunakan dalam analisis atau pemodelan selanjutnya dengan cara melihat dimensi dari DataFrame data, title_matrix, dan text_matrix setelah dilakukan pemrosesan dan transformasi teks menggunakan CountVectorizer.

```
#membuat target variabel y berdasarkan kolom "label" dalam DataFrame 'data', dan membuat m
# Creating Y
y = data.label
# Creating X
x = np.concatenate((np.array(data.drop("label",axis=1)),title_matrix,text_matrix),axis=1)
```

Langkah selanjutnya adalah membuat target variabel y berdasarkan kolom "label" dalam DataFrame data, dan membuat matriks fitur x dengan menggabungkan beberapa komponen seperti kolom-kolom lain dari DataFrame data, title_matrix, dan text_matrix.

```
#untuk melihat dimensi dari matriks fitur x dan target variabel y
print(x.shape)
print(y.shape)

(44898, 5008)
(44898,)
```

Setelah itu, pastikan data telah disiapkan dengan benar dengan cara melihat dimensi dari matriks fitur dan target variabel.

```

#membagi dataset menjadi subset pelatihan dan pengujian menggunakan fungsi tra
from sklearn.model_selection import train_test_split

# Train Test Split
X_train,X_test,Y_train,Y_test = train_test_split(x,np.array(y),test_size=0.25,
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)

```

(33673, 5008)
(11225, 5008)
(33673,)
(11225,)

Jika data sudah diolah, langkah selanjutnya dilakukan pembagian dataset menjadi subset training dan testing, di mana subset testing sebesar 25% dari jumlah keseluruhan dataset.

4.4 Hasil Pembuatan Model

Pada bagian ini dilakukan pembuatan model ANN dengan menggunakan PyTorch serta menggunakan ReLU sebagai fungsi aktivasi dan Adam sebagai Optimizer serta cross entropy sebagai loss. ReLU digunakan karena hanya melibatkan operasi matematika yang sederhana dan tanpa memerlukan fungsi eksponensial atau logaritma. Sedangkan, Adam digunakan sebagai optimizer karena memiliki kemampuan untuk menghasilkan konvergensi yang lebih cepat dibandingkan optimizer lainnya. Pada tahap ini juga dilakukan pengaturan terhadap hyperparameter yang terdapat pada masing-masing layer ANN. Pemilihan nilai hyperparameter berdasarkan beberapa percobaan yang telah dilakukan.

```

#mendefinisikan model ANN dengan beberapa layer dan inisialisasi optimizer serta
class ANN(nn.Module):

    def __init__(self):

        super(ANN,self).__init__() # Inheritance

        self.linear1 = nn.Linear(5008,2000) # IN 5008 OUT 2000
        self.relu1 = nn.ReLU() # Actfunc 1

        self.linear2 = nn.Linear(2000,500) # IN 2000 OUT 500
        self.relu2 = nn.ReLU()

        self.linear3 = nn.Linear(500,100) # IN 500 OUT 100
        self.relu3 = nn.ReLU()

        self.linear4 = nn.Linear(100,20) # IN 100 OUT 20
        self.relu4 = nn.ReLU()

        self.linear5 = nn.Linear(20,2) # IN 20 OUT 2 | OUTPUT

```

```

def forward(self,x):

    out = self.linear1(x) # Input Layer
    out = self.relu1(out)

    out = self.linear2(out) # Hidden Layer 1
    out = self.relu2(out)

    out = self.linear3(out) # Hidden Layer 2
    out = self.relu3(out)

    out = self.linear4(out) # Hidden Layer 3
    out = self.relu4(out)

    out = self.linear5(out) # Output Layer

    return out

model = ANN()
optimizer = torch.optim.Adam(model.parameters(),lr=0.01)
error = nn.CrossEntropyLoss()

```

4.5 Hasil Pelatihan dan Pengujian Model

Setelah model ANN dibuat, selanjutnya dilakukan training, testing, dan evaluasi terhadap model yang telah dibuat.

```

#melatih model ANN menggunakan data pelatihan, menghitung loss, melakukan backpropagation

#melakukan konversi array numpy menjadi tensor Pytorch
X_train = torch.Tensor(X_train)

#menentukan jumlah epoch (iterasi) pelatihan
Y_train = torch.Tensor(Y_train).type(torch.LongTensor)

X_test = torch.Tensor(X_test)
Y_test = torch.Tensor(Y_test)

EPOCHS = 10

for epoch in range(EPOCHS):

    #menghapus gradien
    optimizer.zero_grad()

    #forward Propagation
    outs = model(X_train)

    #menghitung Loss
    loss = error(outs,Y_train)

    #backward propagation
    loss.backward()

    #memperbarui parameters
    optimizer.step()

    #mencetak Loss
    print(f"Loss after iteration {epoch} is {loss}")

```

```
Loss after iteration 0 is 0.7028209567070007
Loss after iteration 1 is 5.01705265045166
Loss after iteration 2 is 0.9272289276123047
Loss after iteration 3 is 0.800531804561615
Loss after iteration 4 is 0.7735241651535034
Loss after iteration 5 is 0.6166948080062866
Loss after iteration 6 is 0.5827904939651489
Loss after iteration 7 is 0.4194452166557312
Loss after iteration 8 is 0.39133065938949585
Loss after iteration 9 is 0.25175124406814575
```

Pelatihan model dilakukan agar model terlatih untuk menghasilkan model yang dapat belajar dari data yang diberikan dan dapat melakukan prediksi berita palsu. Proses ini diulang sebanyak jumlah epoch yang telah ditentukan, dan setiap kali dilakukan iterasi, nilai loss dicetak. Tujuan dari pelatihan adalah untuk mengoptimalkan model sehingga loss semakin kecil dan model dapat memberikan prediksi yang lebih akurat pada data yang belum pernah dilihat sebelumnya.

```
# Import metrics
from sklearn.metrics import accuracy_score, confusion_matrix

# Prediksi
y_head = model(X_test)
print(y_head[0])
# Konversi prediksi menjadi label
y_pred = torch.max(y_head,1)[1]
print(y_pred[0])

# Menghitung skor akurasi
print("Skor Model Akurasi adalah ", accuracy_score(y_pred, Y_test))

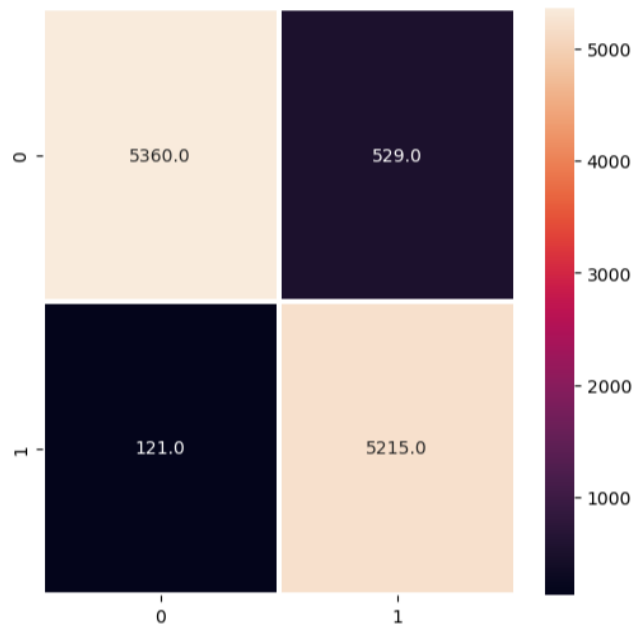
tensor([-0.3321, -0.7762], grad_fn=<SelectBackward0>)
tensor(0)
Skor Model Akurasi adalah 0.9420935412026726
```

Script di atas melakukan prediksi menggunakan model yang telah dilatih pada data pengujian, dan kemudian mengukur seberapa baik model tersebut melakukan prediksi dengan membandingkan prediksi dengan label asli. Skor akurasi memberikan gambaran tentang sejauh mana model dapat memprediksi dengan benar kelas-kelas pada data yang belum pernah dilihat sebelumnya.

```
confusion_matrix = confusion_matrix(y_pred=y_pred, y_true=Y_test)
fig, ax = plt.subplots(figsize=(6,6))
sns.heatmap(confusion_matrix, annot=True, fmt="0.1f", linewidths=1.5)
plt.show()
```

Selanjutnya, diberikan visualisasi yang lebih intuitif dan informatif. Pada bagian ini, menggunakan confusion matrix yang telah dihitung sebelumnya untuk membuat visualisasi

heatmap menggunakan library seaborn dan matplotlib. Heatmap dengan warna dan angka di dalam setiap selnya membantu dalam melihat pola kesalahan prediksi model, serta distribusi kelas yang diprediksi dengan benar. Sumbu x pada confusion matrix menunjukkan label yang diprediksi oleh model, sedangkan sumbu y pada confusion matrix menunjukkan label yang sebenarnya dari data. Adapun hasil dari confusion matrix penelitian ini adalah sebagai berikut.



Berdasarkan gambar di atas, dapat terlihat bahwa pemodelan memiliki akurasi yang baik, di mana hasil true negative dan true positifnya mencapai angka di atas 5000, sedangkan false negative dan false positifnya tidak sampai menyentuh angka 1000.

4.6 Evaluasi Model

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
result = confusion_matrix(y_pred=y_pred,y_true=Y_test)
print("Confusion Matrix:")
print(result)
result1 = classification_report(y_pred=y_pred,y_true=Y_test)
print("Laporan Klasifikasi:",)
print (result1)
result2 = accuracy_score(y_pred=y_pred,y_true=Y_test)
print("Akurasi:",result2)
```

```

Confusion Matrix:
[[5360  529]
 [ 121 5215]]
Laporan Klasifikasi:

```

	precision	recall	f1-score	support
0.0	0.98	0.91	0.94	5889
1.0	0.91	0.98	0.94	5336
accuracy			0.94	11225
macro avg	0.94	0.94	0.94	11225
weighted avg	0.94	0.94	0.94	11225

```

Akurasi: 0.9420935412026726

```

Pada bagian ini kinerja model diukur dalam melakukan prediksi dan mendapatkan pemahaman yang lebih baik tentang keandalan dan keakuratan prediksi yang diberikan oleh model. Script tersebut menggunakan beberapa metrik evaluasi dari modul sklearn.metrics untuk melakukan evaluasi hasil prediksi model. Dari hasil prediksi di atas, dapat disimpulkan bahwa model memiliki prediksi yang sangat baik karena nilai akurasinya menyentuh 0,94.

BAB V

PENUTUP

5.1 Kesimpulan

Dalam penelitian ini, dilakukan pendekatan menggunakan model Artificial Neural Network (ANN) dengan menggunakan framework PyTorch untuk mendeteksi berita palsu. Pendekatan ini diambil mengingat meningkatnya penyebaran berita palsu yang dapat mempengaruhi opini publik dan menciptakan kerugian bagi individu maupun masyarakat.

Dapat disimpulkan bahwa pendekatan menggunakan model ANN dengan menggunakan framework PyTorch memiliki potensi untuk menjadi solusi yang efektif dalam mendeteksi berita palsu. Dengan pengembangan dan penelitian lebih lanjut, pendekatan ini dapat membantu meningkatkan kepercayaan terhadap sumber berita, mempromosikan informasi yang akurat, dan melindungi masyarakat dari penyebaran berita palsu di era digital yang semakin kompleks ini.

5.2 Saran

Untuk penelitian berikutnya, disarankan untuk membandingkan model ANN yang digunakan pada penelitian ini dengan model lainnya agar mendapatkan hasil yang lebih baik. Selain itu, disarankan juga agar menggunakan dataset yang berbeda supaya mengetahui apakah model pada penelitian ini sudah bekerja dengan optimal atau belum. Meskipun model pada penelitian ini menghasilkan akurasi yang baik, tetapi perlu ditingkatkan lagi keandalannya dengan cara mengatur hyperparameter dan menambah jumlah epoch.

DAFTAR PUSTAKA

- Jason Brownie. (2017). Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. Diakses pada 29 Juni 2023, dari <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- Widiputra, H. D. (2016). Artificial Neural Network. Diakses pada 29 Juni 2023, dari <https://dosen.perbanas.id/artificial-neural-network/>
- Setiawan, S. (2020). Membicarakan Precision, Recall, dan F1-Score. Diakses pada 29 Juni 2023, dari <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354>
- Yazar, K. Definition Pytorch. Diakses pada 29 Juni 2023, dari <https://www.techtarget.com/searchenterpriseai/definition/PyTorch>
- What is Natural Language Processing (NLP)? Diakses pada 29 Juni 2023, dari <https://aws.amazon.com/what-is/nlp/>
- Dika. (2003). Framework Pembelajaran Mesin Python. Diakses pada 29 Juni 2023, dari https://appkey.id/pembuatan-website/teknologi-web/pytorch-adalah/#PyTorch_Aadalah

LAMPIRAN

1. Akses Dataset

<https://drive.google.com/drive/folders/1tIZm-VEEn7QBbXpyVngs28at6VBboWtHa?usp=sharing>

2. Akses GitHub

<https://github.com/yogaadipranata/Berita-Palsu-ANN-PyTorch>