THESIS – ES235401

DEBIASING PEMBERITAAN *ONLINE* TERHADAP BERITA PEMILU PRESIDEN DI INDONESIA

LIDIYA YUNIARTI
NRP. 6026231027

DOSEN PEMBIMBING:
Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.
NIP. 198201202005012001

PROGRAM MAGISTER
DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024

THESIS - ES235401

DEBIASING ONLINE NEWS ON PRESIDENTIAL ELECTION COVERAGE IN INDONESIA

LIDIYA YUNIARTI
NRP. 6026231027

SUPERVISOR:
Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.
NIP. 198201202005012001

MASTER PROGRAM
DEPARTMENT OF INFORMATION SYSTEMS
FACULTY OF INTELLIGENT ELCTRICAL AND INFORMATICS TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024

(halaman ini sengaja di kosongkan)

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sistem Informasi (M.Kom.)
di
Institut Teknologi Sepuluh Nopember

Oleh:
Lidiya Yuniarti
NRP: 6026231027

Tanggal Ujian: 19 Juli 2024
Periode Wisuda ITS: 130

Disetujui oleh:
Pembimbing:

Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D
NIP: 198201202005012001

Penguji:

Dr. Ir. Aris Tjahyanto, M.Kom
NIP: 196503101991021001

Dr. Rarasmaya Indraswari, S.Kom
NIP: 1995202012057

Surabaya, 02 Agustus 2024
Kepala Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas

LP/P/24/92
DSI-130-F-23-267/9

Dr. Mudjahidin, ST, MT
NIP: 197010102003121001

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP	:	Lidiya Yuniarti / 6026231027
Program studi	:	S2 Sistem Informasi
Dosen Pembimbing / NIP	:	Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D / 198201202005012001

dengan ini menyatakan bahwa Tesis dengan judul "DEBIASING PEMBERITAAN ONLINE TERHADAP BERITA PEMILU PRESIDEN DI INDONESIA" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 02 Agustus 2024

Mengetahui

Dosen Pembimbing

Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D
NIP. 198201202005012001

Mahasiswa

Lidiya Yuniarti

NRP. 6026231027

Debiasing Pemberitaan *Online* terhadap Berita Pemilu Presiden di Indonesia

Nama Mahasiswa : Lidiya Yuniarti
NRP : 6026231027
Dosen Pembimbing 1 : Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

ABSTRAK

Pemberitaan politik mengenai presiden dalam media massa termasuk berita *online* yang memiliki peran penting dalam proses pemilu dan dalam membentuk opini publik. Namun, pemberitaan tersebut cenderung rentang terhadap bias (seperti nama kandidiat, partai politik dan organisasi yang terlibat dalam pemilu) sehingga dapat memengaruhi persepsi dan partisipasi politik dari masyarakat. oleh karena itu, penting untuk mengindentifikasi dan mengatasi bias dalam pemberitaan politik khususnya terkait pemilu presiden di Indonesia. Adapun metodologi yang digunakan dalam penelitian ini menggunakan *word embedding* yaitu model Word2Vec dan IndoBERT dalam menganalisis resepsentasi dalam pemberitaaan pada berita politik khususnya pemilu presiden di Indonesia dan menggunakan sentimen analisis berbasis leksikon untuk memastikan proses *debiasing* yang diterapkan berhasil mengurangi bias. Adapun hasil dari penelitian ini representasi bias menggunakan *word embedding* berhasil dilakukan, namun metode sentimen analisis berbasis leksikon yang digunakan tidak efektif dalam melakukan melakukan evaluasi bias pada berita pemilu presiden 2024 di Indonesia dibuktikan dengan naiknya nilai sentimen yang menunjukkan kecenederungan sentimen dalam analisis seperti naiknya sentimen negatif sebesar 0,18 dan sentimen positif sebesar 2,05 serta terjadi penurunan nilai sentimen netral sebesar 2,24 setelah dilakukan proses *debiasing.*

Kata kunci: *Debiasing*, isu calon presiden, *word embedding*, dan sentimen analisis.

Debiasing Online Reporting of Presidential Election News in Indonesia

Student Name	: Lidiya Yuniarti
NRP	: 6026231027
Supervisor 1	: Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

ABSTRACK

Political news coverage regarding the president in mass media, including online news, plays an important role in the election process and in shaping public opinion. However, such coverage tends to be susceptible to bias (such as the names of candidates, political parties, and organizations involved in the election), which can influence public perception and political participation. Therefore, it is important to identify and address bias in political news coverage, especially related to the presidential election in Indonesia. The methodology used in this study involves word embeddings, specifically the Word2Vec and IndoBERT models, to analyze representations in political news, particularly the Indonesian presidential election. Additionally, a lexicon-based sentiment analysis is used to ensure that the debiasing process effectively reduces bias. The findings of this study indicate that bias representation using word embeddings was successfully conducted. However, the lexicon-based sentiment analysis method used was not effective in evaluating bias in the news of the 2024 Indonesian presidential election. This is evidenced by the increase in sentiment values, indicating a tendency in sentiment analysis, with a rise in negative sentiment by 0.18, positive sentiment by 2.05, and a decrease in neutral sentiment by 2.24 after the debiasing process.

***Keyword**: Debiasing, presidential candidate issues, word embedding, and sentiment analysis .*

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah SWT, karena dengan rahmat dan hidayah-Nya penulis dapat menyelesaikan Tesis yang berjudul "**Debiasing Pemberitaan Online Terhadap Berita Pemilu Presiden Di Indonesia**" dengan baik dan tepat waktu. Shalawat beserta salam semoga tetap terlimpahkan kepada junjungan kita Nabi Muhammad SAW, sang pembawa kabar gembira dan tauladan bagi umatnya. Penulisan Tesis ini dilakukan dalam rangka memenuhi salah satu syarat kelulusan Program Magister Departemen Sistem Informasi – FTEIC ITS Surabaya.

Dalam proses penulisan Tesis ini, penulis banyak menerima bantuan, bimbingan, masukan dan do'a dari banyak pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Mudjahidin, ST., MT. selaku Kepala Departemen Sistem Informasi Institut Teknologi Sepuluh Nopember.
2. Bapak Ahmad Mukhlason, S.Kom., M.Sc., Ph.D. selaku Kepala Program Studi Magister Sistem Informasi Informasi Institut Teknologi Sepuluh Nopember
3. Ibu Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D. selaku dosen pembimbing sekaligus dosen wali yang selalu memberikan bimbingan, arahan serta motivasi dalam kesuksesan pengerjaan Tesis ini.
4. Bapak Dr. Ir. Aris Tjahyanto, M.Kom. dan Ibu Dr. Rarasmaya Indraswari S.Kom. selaku dosen penguji yang memberikan saran dan masukan pada Tesis ini
5. Orang tua penulis, Sumarni dan Capeng, yang selalu memberikan dukungan moril, materil, nasehat serta do'a yang luar biasa dalam setiap langkah hidup penulis, yang merupakan anugrah terbesar dalam hidup penulis.

6. Semua pihak yang tidak bisa disebutkan satu-persatu. Terimakasih atas do'a, dukungan, bantuan dan semangat sehingga dapat menyelesaikan Tesis ini
7. Lidiya Yuniarti, diri saya sendiri. Apresiasi yang sebesar-besarnya karena sudah berhasil menyelesaikan apa yang sudah dimulai dengan versi terbaikmu. Terimakasih telah bertahan dan berusahan untuk tidak menyerah sejauh ini. Terimakasih telah menikmati setiap proses yang sudah dilewati hingga sejauh ini.

Penulis menyadari penyusunan Tesis ini jauh dari kata sempurna. Oleh karenanya, penulis bersedia menerima kritikan dan saran yang membangun. Terakhir, harapan penulis semoga Tesis ini dapat memberikan manfaat bagi siapa saja yang membacanya.

Surabaya, 02 Agustus 2024

Penyusun,

Lidiya Yuniarti

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Di era perkembangan teknologi informasi yang sangat pesat pada berbagai bidang termasuk dalam dunia jurnalistik. Pertumbuhan portal berita *online* semakin bertambah dan semakin pesat, dibuktikan dengan sebanyak 902 media digital yang mendominasi dari 1.711 perusahaan media yang telah terverifikasi hingga Januari 2023 yang ada di Indonesia sehingga menimbulkan persaingan yang ketat dalam dunia industri tersebut (Muliawanti, 2018) (Indonesiabaik.id, 2023). Pertumbuhan portal berita *online* yang pesat ini dikarenakan sejalan dengan tingginya penetresi internet dan perkembangan kebutuhan pembaca yang semakin membutuhkan informasi dari sumber media tersebut, sehingga media *online* berusaha untuk menyediakan dan menyajikan berita yang dapat memenuhi kebutuhan pembaca (Intyaswati, 2021) (Indonesiabaik.id, 2023). Kemajuan dari perkembangan berita *online* ini dapak berdampak dalam kehidupan sehari-hari, baik dampak positif maupun negatif. Adapun dampak positif yang dirasakan langsung oleh pembaca adalah kemudahan dalam mengakses informasi yang diinginkan secara mudah, terbaru dan cepat (Arini, 2020). Sehingga kualitas informasi yang disajikan dari berita dari berita harus diperhatikan agar informasi yang disajikan tersebut sesuai dengan keinginan pembaca dan menarik perhatian pembaca (Syarifudin, 2019).

Berita *online* adalah salah satu bentuk dari media massa yang memiliki peran penting sebagai saah satu pelopor perubahan dalam kehiduapan masyarakat melalui informasi, hiburan maupun pesan-pesan yang yang disajikan (Khatimah, 2018). Media massa juga memberikan peran penting dalam proses pemelihan umum (pemilu) sebagai pusat informasi untuk mengetahui informasi mengenai politik yang dibutuhkan seperti informasi tentang kandidat, isu-isu politik dan pandangan mengenai partai politik serta memainkan peran kunci dalam membentuk pandangan dan pengetahuan politik masyarakat secara keseluruhan (Adinugroho *et al.*, 2019). Dalam proses pemilu media massa tidak hanya sebagai pusat informasi saja, tetapi juga sebagai pembentuk narasi mengenai politik yang

sedang diberitakan sehingga informasi yang di sajikan dapat memengaruhi persepsi atau minat pemilih dan pemahaman publik mengenai isu-isu politik tersebut (Kleinnijenhuis, van Hoof and van Atteveldt, 2019). Dengan demikian, media massa memainkan peran penting dalam memfasilitasi pertukaran informasi yang penting antara pihak-pihak yang terlibat dalam proses politik, sehingga kualitas informasi politik yang disajikan oleh media berita akan berpengaruh terhadap arah kampanye dan hasil pemilu tersebut (Farid, 2023).

Bias merupakan kecenderungan atau keberpihakan tertentu yang dapat memengaruhi pemikiran atau keputusan (Chen *et al.*, 2020). Dalam pemberitaan politik khususnya mengenai pemilu presiden bias merujuk pada kecenderungan atau keberpihakan dari media tertentu terhadap kandidat calon presiden dan calon wakil presiden (capres-cawapres) atau partai politik tertentu dalam penyajian berita sehingga menyebabkan isi berita dapat mengandung bias atau keberpihakan, misalnya media massa A yang cenderung memberitakan terkait pasangan calon capres-cawapres 1 saja tanpa memberitakan capres-cawapres lainnya. Tidak hanya itu, keberadaan bias gender (nama baik capres-cawapres ataupun partai politik, jenis kelamin, kandidat partai politik atau istilah gender lainnya) dalam berita politik juga juga termasuk dalam kategori bias, sehingga hal menjadi suatu perhatian penting yang harus diperhatikan. Akibat dari bias sendiri dapat menyebabkan distorsi atau penyimpangan makna dari informasi yang dihasilkan sehingga dapat memengaruhi persepsi publik secara tidak objektif dan tidak adil (Kleinnijenhuis, van Hoof and van Atteveldt, 2019). Penyimpangan makna ini bisa terjadi dalam berbagai bentuk seperti ketidakseimbangan representasi, penyimpangan fakta atau framing dari penyajian berita yang tidak objektif sehingga dapat mengubah pandangan publik terhdap calon atau partai politik serta dapat memengaruhi hasil pemilu secara keseluruhan (Suryo and Aji, 2020). Dengan demikian, perlu adanya pengurangan bias dalam pemberitaan politik dalam menyajikan informasi secara objektif dan adil, sehingga dapat memastikan bahwa opini atau keputusan publik didasarkan pada informasi yang akurat dan seimbang.

Adapun penelitian terdahulu yang dilakukan oleh Xiao *et al.*, (2022) Penelitian ini berfokus pada identifikasi polarisasi politik dalam teks media sosial

twitter menggunakan metode Polarity-aware Embedding Multi-task learning (PEM). Metode ini dapat membuat represetasi kata yang dapat memahami polarittas politik serta mengklasifikasikan teks tersebut. Metode tersebut berhasil dalam mempelajari embedding yang memahami dan merepresentasikan polaritas atau orientasi sentimen dari teks yang diberikan. Model tersebut dapat mengenali dan menggambarkan arah atau orientasi opini, pandangan, atau sentimen yang mendasari teks tersebut.

Adapun penelitian lain yang dilakukan oleh Liang *et al.*, (2020) pentingnya untuk menganalisis bias sosial seperti gender, ras, agama dan sebagainya dalam representasi kalimat dalam pengolahan bahasa alami. Penelitian ini menggunakan metode SENTDEBIAS adapun hasil dari penelitian ini menunjukkan bahwa metode SENTDEBIAS yang digunakan mampu menghilangkan bias sosial tersebut dari representasi kalimat dalam bahasa alami. Dari penelitian yang telah dilakukan ini, metode SENTDEBIAS yang digunakan hanya menganalisis bias dalam *word embedding* sehingga belum bisa menyelesaikan permasalahan sejauh mana representasi kata dalam pemberitaan *online* mengenai politik isu politik bersifat adil dan seimbang khususnya calon presiden yang ada di Indonesia.

Selajutnya penelitian lain oleh Spinde *et al.*, (, 2020) menganalisis bias dalam liputan berita di media Jerman dimana cakupan berita tersebut lebih condong ke satu sisi (bias media) sehingga dapat berpengaruh signifikan terhadap bagaimana konsumen berita memahami dan bereaksi terhadap beritat. Penelitian ini menggunakan metode Komponen berbasis IDF (*Inverse Document Frequency*). Dari hasil penelitian tersebut menunjukkan bahwa menunjukkan bahwa kombinasi komponen terbaik menghasilkan skor F1 sebesar 0,31 yang menunjukkan kinerja yang belum optimal dalam mendeteksi bias dan Hasil survei ditemukan bahwa ada perbedaan antara apa yang dianggap sebagai bias oleh sistem dengan apa yang dianggap oleh responden (anotasi manusia).

Berdasarkan penelitian yang dilakukan oleh Xiao *et al.*, (2022) berhasil mengidentifikasi polarisasi politik di Twitter dengan metode PEM. Walaupun metode SENDEBIAS yang dilakukan oleh Liang *et al.*, (2020) terbukti efektif dalam mengatasi bias sosial dalam representasi kalimat, namun metode tersebut belum sepenuhnya mengatasi isu representasi kata dalam pemberitaan politik di

Indonesia. Selain itu, penelitian yang dilakukan oleh Spinde *et al.,* (2020) menunjukkan keterbatasan dalam mendeteksi bias dalam liputan berita media Jerman menggunakan metode IDF. Keterbatasan ditemukan bahwa terdapat perbedaan persepsi antara sistem deteksi dan anotasi manusia mengenai adanya bias dalam berita tersebut.

Dari beberapa penelitian yang telah dilakukan tersebut belum ada yang meneliti tentang bagaimana berita *online* terkait pemilu presiden di Indonesia memberitakan berita serta menilai sejauh mana berita tersebut dapat adil dan simbang. Sehingga pada penelitian ini mencoba menganalisis bagaimana berita *online* memberitakan berita terkait isu politik khususnya mengenai pemilu presiden di Indonesia menggunakan model Word2vec dan IndoBERT untuk menganalisis bias pada isu pemilihan presiden dan dilakukan analisis sentimen untuk mengevaluasi proses *debiasing* yang dilakukan pada pemberitaan *online* tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, rumusan masalah yang akan diselesaikan pada penelitian ini adalah:

1. Bagaimana representasi bias baik sebelum dan sesudah *debiasing* dalam pemberitaaan pada berita *online* terkait pemilu presiden di Indonesia menggunakan *word embedding*?
2. Bagaimana menggunakan analisis sentimen untuk mengevaluasi bias sebelum dan sesudah *debiasing* pada berita politik terkait pemilu presiden 2024 di Indonesia?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang diuraikan sebelumnya, adapun tujuan penelitian yang ingin di capai adalah:

1. Menginvestigasi representasi bias baik seebelum dan sesudah *debiasing* dalam pemberitaaan pada berita *online* terkait pemilu presiden di Indonesia menggunakan *word embedding*.
2. Menggunakan analisis sentimen untuk mengevaluasi bias sebelum dan sesudah *debiasing* pada berita politik terkait pemilu presiden 2024 di Indonesia.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian yang diuraikan, terdapat dua manfaat penelitian yaitu manfaat teoritis dan manfaat praktis. Adapun kedua manfaat tersebut sebagai berikut:

1. Manfaat bagi akademis

   Penelitian ini diharapkan mampu memberikan kontribusi terhadap pengembangan ilmu pengetahuan dalam Pengolahan Bahasa Alami (PBA) utamanya dalam representasi teks terkait analisis mengenai isu politik pada berita *online* dan sentimen pada berita politik. Melalui penyediaan dataset yang sudah tidak mengandung bias, penelitian ini diharapkan dapat digunakan sebagai acuan atau perbandingan untuk penelitian serupa, menyumbangkan pemahaman, ide atau metode analisis yang dapat menjadi dasar untuk penelitian selanjutnya.

2. Manfaat bagi masyarakat

   Penelitian ini diharapkan dapat memberikan manfaat kepada masyarakat langsung berupa pemahaman yang lebih dalam mengenai pengaruh teknologi utamanya pada pemberitaan media dalam penyajian informasi seperti potensi bias dalam berita, kualitas berita itu sendiri serta bagaimana dampak dari informasi yang disajikan tersebut. Selain itu, penelitian ini dapat digunakan sebagai bahan analisis sentimen dalam berbagai aplikasi praktis seperti analisis media sosia, survei opini publik dan lain-lain terkait berita politik di Indonesia.

1.5 Kontribusi Penelitian

Penelitian ini diharapkan mampu memberikan kontribusi secara teori maupun praktis. Secara teori penelitian ini memberikan kontribusi sebuah metode *debiasing* dan analisis sentimen dalam representasi kata dalam konteks politik khususnya pemilu di Indonesia yang dapat dijadikan referensi bagi penelitian yang lain. Secara praktis penelitian ini dapat memberikan panduan praktis bagi media agar dapat meningkatan representasi kata yang lebih adil dan akurat dalam pemberitaan dalam konteks politik khususnya pemilu presiden di Indonesia.

1.6 Batasan Penelitan

Adapun batasan penelitian yang menjadi ruang lingkup penelitian ini adalah sebagai berikut:

1. Data yang digunakan dalam penelitian ini adalah teks berita *online* yang ada di Indonesia terkait pemilu presiden.
2. Teks berita *online* dari situs web Detik.com yang diterbitkan pada periode 29 Juli 2023 hingga 28 November 2023.
3. Analisis *word embedding* dan sentimen analisis akan dilakukan pada teks berita *online* dalam Bahasa Indonesia.
4. Penelitian ini fokus pada aspek tertentu seperti kriteria bias yang tidak memperhatikan konteks dari isi berita atau konteks spesifik lainnya.

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

Dalam bab ini dibahas mengenai kajian pustaka dan dasar teori yang mendukung pengerjaan penelitian. Kajian pustaka terdiri dari penelitian terdahulu yang menjadi acuan dalam penelitian ini. Dasar teori terdiri dari konsep dasar penelitian yang digunakan sebagai acauan teori pada penelitian.

2.1 Kajian Pustaka

Adapun berbagai penelitian terdahulu mengenai bias khususnya bias politik atau sosial yang telah dilakukan sebelumnya dapat dilihat pada Tabel 1 berikut ini.

Tabel 1 Penelitian Terdahulu

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
1.	Analyzing Political Bias and Unfairness in News Articles at Different Levels of Granularity	(Chen *et al.*, 2020)	6.964 artikel dari adfontesmedia.com	• *Recurrent Neural Networks* (RNN) • *Linguistic Inquiry and Word Count* (LIWC)	Politik	Jerman	Hasil dari metode tersebut menyatakan bahwa penggunaan kata-kata atau gaya tertentu seperti emosi negatif, penekanan kata tertentu, cara pandang, dan pengamatan tertentu cenderung memiliki korelasi yang kuat dalam bias politik dalam berita tersebut sehingga memengaruhi bagaimana pemberitaan tersebut mengandung bias politik dan ketidakadilan.
2.	Detecting Political Biases of Named Entities and Hashtags on Twitter	(Xiao *et al.*, 2022)	3.200 data Twitter	*Polarity-aware Embedding Multi-task learning* (PEM)	Politik	Amerika Serikat	Hasil eksperimen menunjukkan bahwa model *Polarity-aware Embedding Multi-task learning* (PEM) berhasil dalam mempelajari embedding yang

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
							memahami dan merepresentasikan polaritas atau orientasi sentimen dari teks yang diberikan. Model tersebut dapat mengenali dan menggambarkan arah atau orientasi opini, pandangan, atau sentimen yang mendasari teks tersebut.
3.	CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models	(Nangia *et al.*, 2020)	1.508 data *Crowdsourced Stereotype Pairs* atau *CrowS-Pairs*	• BERTBase • RoBERTaLarge • ALBERTXXL-v2	Sosial	Amerika Serikat	Hasilnya menunjukkan bahwa ketiga metode yang digunakan secara signifikan cenderung kalimat-kalimat yang menggambarkan stereotip yang berkaitan dengan berbagai jenis bias seperti ras tertentu dan agama-agama tertentu dalam setiap kategori yang ada di CrowS-Pair.

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
4.	Media Bias in German News Articles: A Combined Approach	(Spinde, Hamborg and Gipp, 2020)	Empat sumber berita Jerman: • Süddeutsche Zeitung: 65.000 artikel. • TAZ: 500.000 artikel. • Südkurier: 286.700 artikel. • BILD: Dengan jumlah 2.000 artikel,	*Inverse Document Frequency* (IDF)	Politik	Jerman	Hasil yang diperoleh menunjukkan bahwa ketika menggunakan kombinasi komponen menghasilkan skor F1 sebesar 0.3 menunjukkan bahwa kinerja sistem dalam mendeteksi bias tidak optimal atau belum sempurna. Hasil survei ditemukan bahwa ada perbedaan antara apa yang dianggap sebagai bias oleh sistem dengan apa yang dianggap oleh responden (anotasi manusia).

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
5.	Political Depolarization of News Articles Using Attribute-aware *Word embedding* s	(Liu *et al.*, 2021)	360.000 artikel berita	*Text Annealing Depolarization Algorithm* (TADA)	Politik	Amerika Serikat	Metode TADA peneliti berhasil mendepolarisasi teks dengan mempertahankan makna asli dan kejelasan bacaan. Evaluasi yang dilakukan, baik dari pendekatan empiris, kualitatif, maupun melalui evaluasi oleh manusia, secara konsisten menunjukkan kinerja yang positif dari kerangka kerja yang dikembangkan.
6.	On Measuring Social Biases in Sentence Encoders	(May *et al.*, 2019)	-	• *Word embedding Association Test* (WEAT) • *Sentence Embedding Association*	Sosial	Eropa/Afrika-Amerika	Hasil pengujian menunjukkan variasi bukti adanya bias di antara encoder kalimat tersebut. Beberapa pengujian menunjukkan hubungan yang signifikan atau relevan antara konsep tertentu dan atribut yang mungkin memiliki bias.

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
				Test (SEAT)			
7.	Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems	(Kiritchenko and Mohammad, 2018)	8.640 kalimat dalam bahasa Inggris.	*Equity Evaluation Corpus* (EEC)	Gender dan ras	Afrika-Amerika Eropa-Amerika	Hasilnya menunjukkan bahwa beberapa sistem menunjukkan bias yang signifikan secara statistik atau secara konsisten memberikan prediksi intensitas sentimen yang sedikit lebih tinggi untuk satu ras atau satu gender tertentu.
8.	Studying Political Bias via *Word embedding*s	(Gordon, Babaeianjelodar and Matthews, 2020)	Twitter API dari 576 akun terkait kandidat presiden dan anggota kongres di Amerika Serikat	*Neural Networks*	Politik	Amerika Serikat	Ditemukan tren menarik yang menunjukkan bahwa tweet dari kandidat presiden, baik dari Partai Republik maupun Demokrat, menunjukkan lebih banyak bias politik dibandingkan dengan tweet dari politisi lain di partai yang sama.

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
9.	Assessing Social and Intersectional Biases in Contextualized Word Representations	(Tan and Celis, 2019)	250.000 WebText dataset	*Word embedding Association Test* (WEAT)	Sosial	Afrika Amerika	Hasil penelitian menunjukkan adanya bias dalam tingkat korpus, variasi bukti bias dalam tes asosiasi embedding, dan khususnya bias rasial sangat kuat terenkripsi dalam model kata kontekstual.
10.	Towards *Debiasing* Sentence Representations	(Liang *et al.*, 2020b)	1.080 WikiText-2 dataset	SENT-DEBIAS	Sosial	Multiple	Metode yang digunakan berhasil menunjukkan efektivitas dalam menghilangkan bias sosial pada representasi kalimat dan tetap mempertahankan kinerja pada tugas-tugas pemrosesan bahasa alami di level kalimat.
11.	Gender bias recognition in political news articles	(Davis, Worsnop and Hand, 2022)	5000 artikel media *online* Amerika Serikat	• Regular Expression (Regex) • Named	Politik	Amerika Serikat	Hasil dari penelitian ini menunjukkan bahwa meskipun sudah dilakukan penghilangan informasi berbasis gender (nama,

No	Judul	Penulis	Dataset	Metode Evaluasi Bias	Domain	Bahasa/Negara	Hasil
				Entity Rcognation (NER)			jenis kelamin atau istilah gender lainnya) dan informasi pribadi secara efektif namun masih terdapat tingkat bias yang cukup besar terdeteksi dalam berita politik tersebut.

Berdasarkan dari penelitian terdahulu yang telah dilakukan, langkah selanjutnya adalah untuk mengetahui gap pada penelitian yang dilakukan. Dari hasil penelitian terdahulu pada tabel diatas hanya berfokus pada mendeteksi bias dan melakukan *debiasing* saja menggunakan *word embedding* , sedangkan pada penelitian yang akan dilakukan tidak hanya berfokus pada mendeteksi bias dan *debiasing* atau menghilangkan bias saja tetapi juga dilakukan perhitungan keadilan (*fairness measurement*) untuk memastikan bahwa algoritma, model, atau sistem NLP tidak memperkuat atau memperluas bias pada berita politik terkait pemilu presiden yang ada dalam berita tersebut serta menilai sejauh mana mana representasi bias adil dan seimbang pada pemberitaan *online* khususnya mengenai pemilu presiden Indonesia. Berikut merupakan Tabel 1 mengenai gap penelitian terdahulu dengan penelitian saat ini.

Tabel 2 Gap Penelitian Terdahulu

No	**Judul**	**Referensi**	***Debiasing***	***Pengukuran Evalusasi Bias***	**Penelitian yang akan dilakukan**
1.	Analyzing Political Bias and Unfairness in News Articles at Different Levels of Granularity	(Chen *et al.*, 2020)		☑	Pada penelitian yang akan dilakukan akan melakukan dua *task* atau pekerjaan yaitu *debiasing* sekaligus *fairness measurement* pada pemberitaan politik terkait berita pemilu presiden di Indonesia.
2.	Detecting Political Biases of Named Entities and Hashtags on Twitter	(Xiao *et al.*, 2022)		☑	
3.	CrowS-Pairs: A Challenge Dataset for	(Nangia *et al.*, 2020)		☑	

No	Judul	Referensi	*Debiasing*	*Pengukuran Evalusasi Bias*	Penelitian yang akan dilakukan
	Measuring Social Biases in Masked Language Models				
4.	Media Bias in German News Articles: A Combined Approach	(Spinde, Hamborg and Gipp, 2020)		☑	
5.	Political Depolarization of News Articles Using Attribute-aware *Word embedding*s	(Liu *et al.*, 2021)		☑	
6.	On Measuring Social Biases in Sentence Encoders	(May *et al.*, 2019)		☑	
7.	Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems	(Kiritchenko and Mohammad, 2018)		☑	
8.	Studying Political Bias via *Word embedding*s	(Gordon, Babaeianjelodar and Matthews, 2020)		☑	

No	Judul	Referensi	*Debiasing*	*Pengukuran Evalusasi Bias*	Penelitian yang akan dilakukan
9.	Assessing Social and Intersectional Biases in Contextualized Word Representations	(Tan and Celis, 2019)		☑	
10.	Towards *Debiasing* Sentence Representations	(Liang *et al.*, 2020b)	☑		
11.	Gender bias recognition in political news articles	(Davis, Worsnop and Hand, 2022)	☑		
Penelitian yang akan dilakukan			☑	☑	

2.2 Dasar Teori

Konsep dasar yang dijadikan dasar teori dalam penelitian ini, bersumber dari buku, jurnal, artikel, dan lain-lain untuk memahami konsep atau teori dalam menyelesaikan permasalahan *debiasing* dan sentimen analisis pada isu politik khususnya pemilu presiden di Indonesia. Subbagian perikut ini menjelaskan mengenai dasar teori yang digunakan dalam penelitian ini.

2.2.1 Pemilihan Umum (Pemilu)

Pemilihan umum atau yang biasa dikenal dengan pemilu adalah sebuah proses yang dilakukan pada suatu negara untuk memilih wakil-wakil dalam pemerintahan atau lembaga legislatif (Solihah, 2018). Proses pemilihan tersebut melibatkan pemilihan individu atau partai politik yang akan menduduki posisi politik tertentu, seperti presiden, anggota parlemen, gubernur atau jabatan lokal yang ada pada negara tersebut (Subiyanto, 2020). Pemilu sendiri dapat dilakukan

dengan beberapa sistem, seperti sistem pemilihan langsung pemilihan langsung dan sistem pemilihan tidak langsung. Pemilu secara langsung adalah warga negara yang sudah berhak memilih memberikan suara secara langsung kepada calon yang ada sesuai dengan hari nurani dan tanpa perantara apapun. Sedangkan pemilu secara tidak langsung adalah pemilihan yang dilakukan oleh warga yang memilih calon dengan memberikan suara kepada partai politik yang menjadi perwakilan (Nugraha and Mulyandari, 2016).

Adapun tujuan dilakukannya pemilu adalah untuk memastikan jalan dalam pergantian pemerintahan dengan tertib dan aman, serta sebagai rangka melaksanakan hak dasar warga negara untuk ikut andil dalam proses demokratis. Dikutip pada data KPU mengenai calon presiden pada pemilu tahun 2024 terdapat tiga pasangan calon (paslon) presiden dan wakil presiden, yakni Ganjar Pranowo dan Mahfud MD, Prabowo Subianto dan Rakabumin Raka serta Anies Rasyid Baswedan (Anies Baswedan) dan Muhaimin Iskandar (Komisi Pemilihan Umum Republik Indonesia, 2023).

2.2.2 Bias

Bias adalah kecenderungan atau preferensi yang tidak seimbang, kecenderungan memihak atau memberikan perlakuan yang tidak adil terhadap suatu pihak atau kelompok tertentu (Swinger *et al.*, 2018). Hal ini dapat terjadi dalam berbagai bentuk, seperti pengambilan keputusan, sikap atau tindakan yang dapat berpengaruh terhadap cara pandang, asumsi atau bertindak terhadap seseorang atau kelompok tersebut. Bias dapat terbagi dalam beberapa jenis, seperti bias gender, sosial, ras, suku dan lain sebagainya (Petreski and Hashim, 2023). Kemunculan bias dapat terjadi dalam berbagai bidang kehidupan, seperti pemikiran manusia, media, teknologi, penelitian ilmiah yang menghasilkan kecenderungan sifat positif atau negatif yang terjadi tanpa disadari.

Adapun bias dalam politik khususnya mengenai pemilu presiden sendiri merujuk pada kecenderungan atau keberpihakan tertentu terhadap kandidat (capres-cawapres) atau partai politik tertentu. Selain dari pada itu, keberadaan bias dalam pemberitaan *online* mengenai politik, adanya bias gender (nama baik capres-cawapres ataupun partai politik, jenis kelamin, kandidat partai politik atau istilah gender lainnya) termasuk dalam kategori bias, sehingga hal menjadi suatu

perhatian penting yang harus diperhatikan. Akibat dari adanya bias pada berita politik bagi pemilu sendiri dapat menyebabkan framing (cara penyajian) yang cenderung memihak sehingga dapat menyebabkan persepsi tertentu terhadap kandidat atau partai politik serta pandangan masyarakat terhadap isu pemilu tersebut (Eriyanto, 2011).

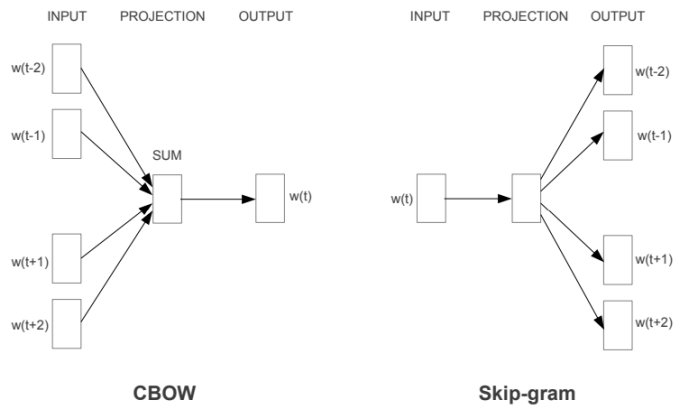
2.2.3 Word embedding

Word embedding merupakan sebuah teknik yang ada dalam permrosesan bahasa alami atau *neural language processing* (NLP) dimana kata-kata atau frasa direpresentasikan sebagai vektor nilai dalam suatu ruang vektor yang telah ditentukan sebelumnya (Yu *et al.*, 2018). Representasi vektor kata dalam *word embedding* dapat membantu menggambarkan pesebaran makna seperti hubungan antar kata yang berkaitan atau muncul (Tan and Celis, 2019). Ada beberapa jenis *word embedding* yang umumnya dapat digunakan dalam pemrosesan bahasa alami, diantaranya: Word2vec, GloVe (Global Vectors for Word Representatition), FastText, IndoBERT (Indonesian Bidirectional Encoder Representation from Transformers), ELMo (Embeddings from Language Models) (Tan and Celis, 2019) (Romanyshyn, Chaplynskyi and Zakharov, 2023). Pada penelitian ini *word embedding* yang digunakan adalah model Word2Vec dan IndoBERT.

2.2.3.1 Word2Vec

Model Word2Vec digunakan untuk memprediksi kata dalam bentuk vektor (Styawati *et al.*, 2022). Model ini adalah aplikasi dari teks yang tanpa label (*unsupervised learning*) yang menggunakan jaringan saraf dengan struktur berisi lapisan tersembunyi dan lapisan terhubung penuh (Zhu and Ren, 2023). Ukuran matriks bobot di setiap lapisan dihitung dengan mengalikan jumlah kata dalam korpus dengan jumlah neuron tersembunyi dilapisan tersembunyi. Matriks bobot di lapisan tersembunyi dari model yang sudah dilatih berfungsi untuk mengonversi kata menjadi vektor. Secara konseptual, matriks bobot ini bekerja seperti tabel pencarian, di mana setiap baris merepresentasikan satu kata dan setiap kolom merepresentasikan vektor dari kata tersebut. Ada dua arsitektur yang dapat digunakan dalam word2vec, yaitu *Continuous Bagof-Word* (CBOW) yang digunakan untuk memprediksi target kata berdasarkan frekuensi kata dan skip-

gram yang digunakan untuk memprediksi target konteks dan akan memprediksikan kata-kata yang anggap jarang atau tidak biasa (Nurdin *et al.*, 2020). Berikut merupakan gambar dari arsitektur CBOW dan skip-gram.



Gambar 1 Model COBOW dan skip-gram (The MathWork, 2023)

Pada arsitektur COBOW mempunyai tiga lapisan yang terdiri dari *input, projection dan output*. Berikut merupakan penjelasan untuk lapisan tersebut (Mikolov, Yih and Zweig, 2013):

- Kata saat ini *(current word)* atau "w(t)" merupakan *output* untuk memprediksi konteks (sebagai target) pada waktu "t".
- "w(t-2)", "w(t-1)" merupakan kata *(word)* yang berjarak dua atau satu posisi sebelum kata target "w(t)" merupakan *input*.
- "w(t+2)", "w(t+1)" merupakan kata *(word)* yang berjarak dua atau satu posisi setelah kata target "w(t)" merupakan *input*.
- Sum sebagai *hidden layer* pada *projection* yang menghitung titik dari vektor tersebut.

Sedangkan pada arsitektur skip-gram sama seperti COBOW yang mempunyai tiga lapisan yang terdiri dari *input, projection dan output*. Tetapi target kata yang menjadi *output*nya merupakan kebalikan dari arsitektur COBOW, berikut merupakan penjelasan dari arsitektur skip-gram sebagai berikut (Suleiman and Awajan, 2018):

- Kata saat ini *(current word)* atau "w(t)" merupakan *input* untuk memprediksi konteks (sebagai target) pada waktu "t".

- "w(t-2)", "w(t-1)" merupakan kata *(word)* yang berjarak dua atau satu posisi sebelum kata target "w(t)" sebagai *output*.
- "w(t+2)", "w(t+1)" merupakan kata *(word)* yang berjarak dua atau satu posisi setelah kata target "w(t)" sebagai *output*.

Adapun persamaan yang digunakan dalam word2vec untuk merepresentasi kata menggunakan persamaan *cosine similarity,* yang menghitung kemiripan dua vektor (Jatnika, Bijaksana and Suryani, 2019).

$$Similarity = \cos\ \theta\ (x.y)/(|(|x|)|\ .||y||) \quad (2.1)$$

Dimana:

- x.y adalah hasil perkalian vektor
- ||x|| adalah panjang vektor x
- ||y|| adalah panjang vektor y

2.2.3.2 IndoBERT

IndoBERT (Indonesian BERT), BERT sendiri merupakan singkatan dari *Bidirectional Encoder Representations from Transformers* adalah salah satu dari beberapa model Bahasa Indonesia yang digunakan dalam *word embedding* pada pemrosesan bahasa alami yang dikembangkan dari hasil kolaborasi Institut Teknologi Bandung (ITB) dengan Pusat Teknologi Informasi dan Bahasa (PTIB) dengan perusahaan teknologi Google. Model ini dirancang untuk merepresentasikan kata dalam teks kalimat yang tidak memiliki label dengan mempertimbangkan konteks kata-kata dari dua arah yaitu kanan dan kiri (dibireksional) atau gabungan keduanya dalam lapisan (Devlin *et al.*, 2018). Adapun arsitektur *transformer* pada model IndoBERT menggunakan arsitektur jaringan saraf tiruan yang memiliki beberapa tugas khusus untuk setiap jaringan, dimana arsitektur ini sendiri memiliki dua bagian utama, yaitu *encoder* yang hanya berfokus pada pengubahan urutan input kata menjadi representasi vektor dan *decoder* yang bertugas untuk menghasilkan urutan output berdsarkan representasi vektor input yang diberikan. Untuk model ini hanya berada pada bagian encoder saja yang bertugas untuk membuat model bahasa yang siap digunakan untuk pemrosesan lainnya (Putra, Arif Bijaksana and Romadhony, 2021).

Adapun tahapan kerja pada IndoBERT sendiri ada dua, yaitu tahap *pre-traing* dan *fine-tuning* tahapan ini sebenarnya sama saja dengan BERT, hanya saja IndoBERT dilakukan untuk melatih teks dalam Bahasa Indonesia. Tahap *pre-training* adalah tahap untuk melatih model menggunakan data teks yang tidak berlabel dengan melakukan tugas pemrosesan teks yang lebih besar dan umum. Ada dua teknik yang dilakukan oleh IndoBERT pada tahap *pre-training* ini, yaitu teknik *Masked Language Model* (MLM) yang memprediksi kata dalam kalimat tersebut dengan tujuan untuk melatih model memahami kata disekitarnya. Sedangkan teknik *Next Sentence Prediction* (NSP) adalah untuk memrediksi kata apakah kalimat setelahnya merupakan kelanjutan dari kalimat sebelumnya dengan tujuan untuk melatih model memahami hubungan antar kalimat dalam konteks yang lebih luas. Setelah tahapan pre-training selesai, model IndoBERT dapat melakukan tahapan *fine-tuning* pada IndoBERT digunakan untuk melakukan tugas pemrosesan yang lebih khusus dan tertentu saja seperti klasifikasi teks atau tugas lainnya dengan menggunakan dataset yang lebih kecil untuk mendapakan hasil yang lebih akurat berdasarkan model yang sudah dilatih sebelumnya pada proses *pre-train* (Devlin *et al.*, 2018).

2.2.4 Web Scraping

Web scraping adalah suatu teknik yang dilakukan untuk mengambil data atau informasi yang banyak untuk beberapa keperluan seperti riset, analisis dan lain sebagainya (Flores, Permatasari and Jasa, 2020). Tujuan utama dilakukannya *web scraping* adalah untuk mendapatkan data yang terstruktur dari sumber data yang diambil dan menjadi format yang dapat dianalisis (A. Yani, Pratiwi and Muhardi, 2019). Secara umum ada beberapa langkah yang dapat dilakukan untuk melakukan scraping seperti mengidentifikasi website yang akan di scraping, *inspect element* pada website untuk menemukan struktur HTML yang digunakan, membuat skrip atau kode untuk mengekstrak data dan menjalankan kode yang sudah dibuat untuk mendapatkan data (Rahmatulloh and Gunawan, 2020).

2.2.5 Sentimen Analisis

Analisis sentimen adalah teknik untuk menganalisis klasifikasi sentimen untuk memahami opini atau emosi dalam teks dalam bentuk klasifikasi positif, negatif dan netral. Analisis sentimen dapat dilakukan dengan beberapa

pendekatan, seperti laxicon-based (berbasis lexicon) atau berbasis pembelajaran mesin. Lexicon adalah kumpulan kata yang telah diberi label nilai yang menentukan sentimen tertentu, baik negatif ataupun positif (Najib *et al.*, 2019). Adapun metode yang digunakan pada tahapan pengukuran bias ini adalah metode sentimen analisis berbasis *lexicon-based*. Adapun polaritas skor sentimen untuk tiap kata yang bernilai positif, negatif dan netral dalam teks *(compound)* dilakukan proses normalisasi, sehingga hasil akhirnya berada dalam rentang -1 hingga 1, dengan penjelasan sebagai berikut (Bessa, 2023):

- Nilai -1 menunjukkan sentimen yang paling negatif
- Nilai 0 menunjukkan sentimen netral
- Nilai 1 menunjukkan sentimen paling positif

Selain itu, terdapat beberapa skor sentimen khusus untuk kategori negatif, positif dan netral dalam teks yang dinyatakan dengan rentang nilai 0 hingga 1 dengan penjelasan sebagai berikut :

- Skor negatif: untuk menunjukkan sentimen negatif dalam teks, nilai 0 menunjukkan tidak adanya sentimen negatif dalam teks sedangkan 1 menunjukkan adanya sentimen negatif dalam teks
- Skor positif: untuk menunjukkan sentimen positif dalam teks, nilai 0 menunjukkan tidak adanya sentimen positif dalam teks sedangkan 1 menunjukkan adanya sentimen positif dalam teks
- Skor netral: untuk menunjukkan sentimen netral dalam teks, nilai 0 menunjukkan tidak adanya sentimen netral dalam teks sedangkan 1 menunjukkan adanya sentimen netral dalam teks

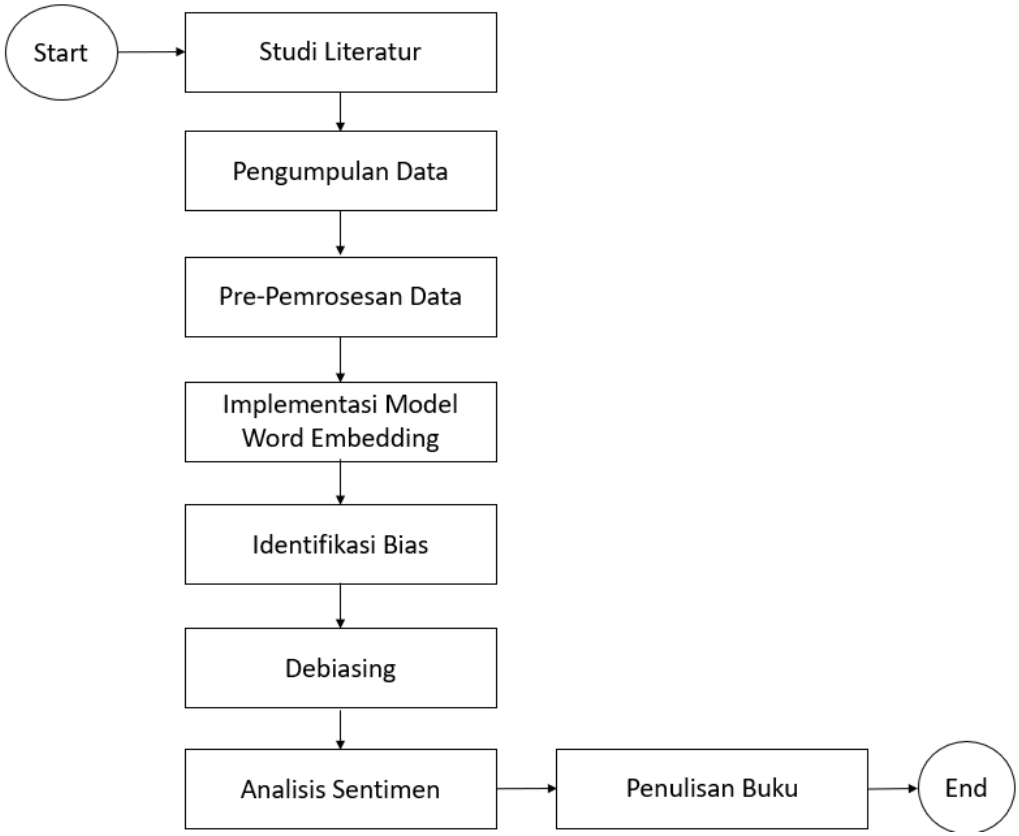
Halaman ini sengaja dikosongkan

BAB 3
METODOLOGI PENELITIAN

Dalam bab ini akan dibahas mengenai metodologi penelitian yang akan dilakukan. Metode penelitian ini berisi cara sistematis yang menyusun kerangka kerja secara terstruktur untuk mencapai tujuan penelitian yang mencangkup langkah-langkah atau prosedur untuk menyelesaikan penelitian.

3.1 Diagram Metodologi Penelitian

Berikut merupakan diagram penelitian yang menggambarkan secara umum dari langkah-langkah yang akan dilakukan pada penelitian. Secara umum diagram penelitian ini dimulai dari studi literatur, dilanjutkan dengan pengumpulan data, pre-pemrosesan data, implementasi model *word embedding*, identifikasi bias, *debiasing*, analisis sentimen dan penulisan buku dan penelitian selesai untuk lebih jelasnya dapat dilihat pada Gambar 2



Gambar 2 Diagram Metodologi Penelitian

3.2 Uraian Metodologi Penelitian

Pada sub bab bagian uraian penelitian akan menjelaskan mengenai detail dari langkah umum yang akan dilakukan. Berikut merupakan penjelasan dari tiap langkah tersebut.

3.2.1 Studi Literatur

Langkah pertama adalah studi literatur mengenai topik yang akan dilakukan penelitian. Dalam Studi literatur terdapat proses penting yang harus dilakukan yaitu menganalisis atau mengulas karya tulis yang sudah ada yang relevan dengan topik penelitian yang ingin dilakukan. Sumber-sumber mengenai studi literatur dapat berasal dari berbagai media seperti jurnal ilmiah, buku, *conference* atau pusat informasi lainnya yang memuat mengenai informasi yang dibutuhkan sesuai dengan topik yang akan dilakukan. Tujuan dari studi literatur ini adalah untuk memahami dan mengevaluasi penelitian sebelumnya untuk

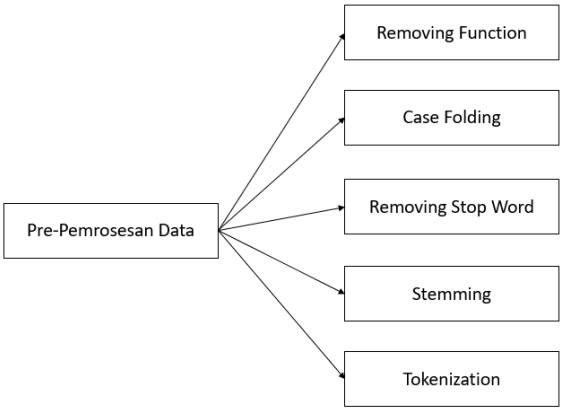
mengidentifikasi celah pengetahuan dan memberikan dasar teoritis untuk penelitian yang akan dilakukan.

3.2.2 Pengumpulan Data

Langkah kedua dari metodologi penelitian ini adalah pengumpulan data yang merupakan tahap kunci dalam mengumpulkan informasi atau fakta yang diperlukan. Dalam pengumpulan data penelitian dapat dilakukan dengan beberapa cara tergantung dengan metode yang digunakan, baik cara kualitatif maupun kuantitatif. Adapun metode pengumpulan data kualtitif seperti wawancara, fokus grup diskusi, studi kasus, observasi dan lain sebagaimana. Sedangkan metode pengumpulan data kuantitatif seperti survei, eksperimen, pengukuran dengan alat ukur dan lain sebagainya. Pada penelitian yang akan dilakukan ini pengumpulan data dilakukan dengan cara scraping dengan library newspaper3k pada website berita *online* detik.com menggunakan kata kunci terkait calon presiden dan data berita yang diambil pada periode data dari 29 Juli 2023 hingga 28 November 2023.

3.2.3 Pre-pemrosesan Data

Tahap ketiga adalah pre-pemrosesan data atau *data preprocessing* yang merupakan proses pengolahan data mentah sebelum data tersebut dapat digunakan untuk analisis lebih lanjut. Proses dalam pre-pemrosesan data dapat dilakukan sesuai dengan kebutuhan analisis yang akan diterapkan. Adapun tujuan dari pre-pemrosesan data adalah untuk mempersiapkan data sehingga dapat menghasilkan analisis yang lebih baik. Berikut merupakan Gambar 1 serta penjelasan dari pre-pemrosesan data yang akan dilakukan pada penelitian ini:



Gambar 3 Tahapan Pra-pemrosesan Data

3.2.3.1 Removing function

Pre-pemrosesan yang pertama adalah *removal function* atau penghapusan yang menghilangkan tanda baca, karakter khusus, atau elemen teks lainnya yang tidak penting. Penghapusan tanda baca ini meliputi koma (,), titik (.), tanda seru (!), tanda tanya (?), kurung (), petik tunggal atau dua (‘/”) dan tanda lain sebagainya. Tujuan dari penghapusan tanda baca adalah untuk menyederhanakan teks dan memastikan bahwa tanda baca tidak menggangu analisis atau pemrosesan selanjutnya.

3.2.3.2 Case folding

Kemudian dilakukan *case folding* untuk mengubah semua huruf menjadi huruf kecil *(lower case)* untuk mempermudah proses analisis. *Case folding* ini dilakukan untuk meningkatkan konsistensi dalam teks dan mengurangi dimensi atau kompleksitas dalam analisis data sehingga analisis yang dilakukan lebih mudah dan efisien.

3.2.3.3 Removing stop word

Pre-pemrosesan ketiga adalah *removing stop word* atau menghilangkan kata umum yang tidak memberikan banyak informasi kontekstual seperti “di”, “dan”, “ke”, ”dari” dan lain sebagainya. Tujuan dari menghilangkan kata umum ini adalah untuk meningkatkan efisiensi emrosesan dengan mengurangi ukuran data, memfokuskan pemrosesan pada kata kunci, meningkatkan akurasi model analisis yang dilakukan.

3.2.3.4 Stemming

Setelah beberapa informasi yang tidak penting pada tahap sebelumnya dilakukan, selanjutnya dilakukan *stemming* untuk mengubah kata-kata menjadi bentuk kata dasarnya, seperti “lari”, “berlari” akan menjadi lari. Tujuan dari langkah ini adalah untuk menyederhakan teks sehingga komputer dapat dengan mudah untuk memahami kata tersebut. Pada tahap ini kata dalam teks akan diubah kedalam kamus Bahasa Indonesia.

3.2.3.5 Tokenization

Pre-pemrosesan terakhir yang dilakukan adalah *tokenization* atau tokenisasi yang memisahkan teks menjadi unit berupa kata yang lebih kecil atau yang disebut dengan token. Adapun tujuan dari tokenisasi ini adalah untuk memudahkan analisis teks yang diproses secara terpisah, selain itu dapat memudahkan pengukuran frekuensi kata dalam teks serta dapat memungkinkan pembentukan n-gram untuk menganalisis konteks atau hubungan antar kata yang dapat membuat teks lebih terstruktus dan siap untuk diolah dan dianalisis lebih lanjut.

3.4 Implementasi *Word embedding*

Langkah keempat adalam mengimplementasikan *word embedding* . *Word embedding* sendiri merupakan representasi vektor dalam ruang numerik yang memperhitungkan hubungan antar kata. Dalam mengimplementasikan *word embedding* pada penelitian ini menggunakan Word2Vec dan IndoBERT untuk menghasilkan representasi vektor kata-kata dalam teks berita. Berikut merupakan langkah yang dilakukan:

1. Merepresentasikan kata pada berita menjadi vektor menggunakan Word2Vec dan IndoBERT.
2. Membuat visualisasi kata menggunakan Principal Component Analysis (PCA) dan t-distributed Stochastic Neighbor Embedding (t-SNE) menjadi vektor 2 dimensi (2D) pada Word2Vec dan IndoBERT.
3. Membandingkan hasil visualisasi kata pada PCA dan TSNE yang dilakukan pada Word2Vec dan IndoBERT untuk mengidentifikasi perbedaan pola atau distribusi kata pada saat representasi.
4. Pengambilan vektor kata pada Word2Vec dan IndoBERT

Adapun tujuan dari implementasi *word embedding* ini adalah untuk melihat bagaimana kata-kata di representasikan dalam ruang vektor, serta dengan visualisasi yang dilakukan dapat mempermudah memahami kata dalam ruang vektor.

3.5 Identifikasi Bias

Langkah ke lima adalah mengidentifikasi bias. Beberapa langkah yang dapat dilakukan dengan menentukan kriteria untuk mengidentifikasi kata-kata yang mengandung bias. Misalnya, kata-kata terkait dengan nama orang, nama kandidat partai politik, kandidat capres, partai politik atau organisasi tertentu yang sering muncul dalam konteks berita pemilu presiden.

Tahapan ini dilakukan untuk membuat dasar yang objektif untuk *debiasing* teks pada proses selanjutnya. Identifikasi bias ini juga sangat penting dilakukan, karena dengan identifikasi bias dapat memahami informasi terkait bias dan memastikan analisis sentimen pada tahap akhir lebih objektif .

3.6 Debiasing

Setelah mengidentifikasi bias pada proses sebelumnya, langkah selanjutnya akan dilakukan *debiasing* atau proses untuk menghilangkan bias. Adapun beberapa langkah yang dilakukan pada saat *debiasing*:

1. Pelabelan.

   Pada proses pelabelan ini akan dilakukan pelabelan kata yang bias berdasarkan kriteria pada proses sebelumnya menggunakan teknik NER (Named Entity Recognition).

2. Penggantian kata (*debiasing*)

   Setelah dilakukan pelabelan maka akan dilakukan penggantian kata atau *debiasing* berdasarkan pelabelan kata dengan NER sebelumnya dan di gantikan dengan kata yang telah ditentukan agar lebih netral.

3. Memastikan proses *debiasing*

   Memastikan proses *debiasing* yang dilakakukan apakah *debiasing* sudah dilakukan berhasil. Proses ini dilakukan pengecekan kemunculan kata yang bias baik sebelum dan sesudah *debiasing,* kemudian dibandingkan hasilnya

Tujuan dari *debiasing* ini adalah untuk membuat kata menjadi netral sehingga dapat mengurangi bias atau menghilangkan distorsi yang tidak diinginkan dalam informasi, khususnya terkait nama kandidat capres-cawapres, partai politik dan organiasi tertentu dalam berita pemilu presiden di Indonesia.

3.7 Analisis Sentimen

Pada tahap ini dilakukan pengkuruan evaluasi bias sebelum dan sesudah proses *debiasing* dengan menganalisis sentimen dengan metode Lxicon-Based pada data yang belum di *debiasing* dan yang sudah di *debiasing*, berikut merupakan langkah yang dilakukan adalah:

1. Menghitung nilai sentimen untuk setiap berita dengan menjumlahkan nilai sentimen dari kata yang ada dalam kamus sentimen. Perhitungan nilai sentimen ini dilakukan pada saart sebelum proses *debiasing* dan setelah proses *debiasing*
2. Mengklasifikasikan nilai sentimen yang sudah dihitung sebelumnya dan diklasifikasikan ke dalam kategori sentimen positif, negatif dan netral pada tiap text berita. Perhitungan nilai sentimen ini juga dilakukan pada saart sebelum proses *debiasing* dan proses *didebiasing.*
3. Membandingkan nilai sentimen yang sudah dihitung tersebut pada saat belum dilakukan *debiasing* ataupun yang sudah di *debiasing.*

Analisis sentimen ini digunakan untuk mengevaluasi apakah proses *debiasing* yang digunakan berhasil mengurangi bias.

Halaman ini sengaja dikosongkan

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dan pembahasan yang didapatkan dari penelitian yang dilakukan. Hasil pada bab ini berisikan dan menjelaskan temua-temuan yang didapatkan serta akan dibahas dari hasil tersebut. Berikut merupakan merupakan hasil dan pembahasan pada penelitian ini.

4.1 Pengumpulan Data

Pada penelitian ini, sumber data yang digunakan dalam mengumpulkan data adalah website berita detik.com (www.detik.com). Pengumpulan data dilakukan dengan cara scraping dengan library newspaper3k pada website berita *online* detik.com menggunakan kata kunci terkait calon presiden dan data berita yang diambil pada periode data dari 29 Juli 2023 hingga 28 November 2023. Kata kunci yang digunakan dalam pengumpulan link berita adalah terkait tag mengenai kandidat presiden pada pemilu presiden 2024 di Indonesia, kata kunci tersebut meliputi berita mengenai Anies Baswedan, Muhaimin Iskandar, Prabowo Subianto, Ganjar pranowo dan Mahfud MD. Kemudian dilakukan pengumpulan data berdasarkan link berita sesuai kata kunci yang sudah dikumpulkan, kemudian dilakukan penghapusan berita yang duplikat dan terakhir dilakukan scraping berita yang sudah tidak duplikat. Hasil scraping yang di dapatkan akan disimpan dalam format Comma Separated Value (CSV) dengan total 7 kolom informasi. Adapun 7 kolom yang memuat informasi tersebut seperti nomor, *title, writer, publish_date, article_text, url,* dan *main_image*.

Berikut merupakan potongan kode pada saat scraping link berita dapat dilihat pada Kode 1 berikut

Kode 1 Scraping Link Berita

```
# Memproses scraping untuk setiap artikel
df_article = pd.DataFrame()
for index, response in tqdm(grequests.imap_enumerated(rs, size=5)):
    full_url = urljoin(url, article_links[index])
    article = Article(full_url)
    article.download()
```

```
    article.parse()
    # Menggunakan XPath untuk informasi tambahan
    tree = html.fromstring(response.content)
    #XPath untuk informasi tambahan
    xpath_title = '//h1[@class="detail__title"]/text()'
    xpath_author = '//div[@class="detail__author"]/span/text()'
    xpath_publish_date = '//div[@class="detail__date"]/text()'
    xpath_main_image = '//div[@class="detail__media"]/img/@src'
    # Ekstraksi dengan XPath
    title = tree.xpath(xpath_title)
    author = tree.xpath(xpath_author)
    publish_date = tree.xpath(xpath_publish_date)
    main_image = tree.xpath(xpath_main_image)
     data = {'Judul': title, 'Penulis': author, 'Tanggal publikasi': publish_date, 'Teks berita':
article.text, 'URL': full_url}
    df = pd.DataFrame(data)
    df_article = pd.concat([df_article, df])
```

Pada saat scraping link berita dilakukan menggunakan ‘grequest’ dan ‘newspaper3k’ untuk mengunduh dan membaca artikel, kemudian menggunakan metode tambahan ‘Xpath’ dengan ‘lxml’ untuk mengambil informasi seperti judul, penulis, tanggal publikasi dan gambar utama. Kemudian disimpan dalam variabel yang sesuai dalam DataFrame pandas. Setelah pengumpulan link selesai kemudian dilakukan penghapusan link yang duplikat dengan Kode 2berikut

Kode 2 Penghapusan Link Duplikat

```
print(df_links.shape)
df_links = df_links.drop_duplicates()
print(df_links.shape)
```

Setelah penghapusan link berita yang duplikat dilakukan scraping data akhir, berikut potongan kodenya

Kode 3 Scraping Berita Akhir

```
article_links = list(df_links['url'])
start_pos = 24820
```

```
rs = (grequests.get(u) for u in article_links[start_pos:25000])

# Setting user agent
user_agent = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101 Firefox/78.0'

config = Config()
config.browser_user_agent = user_agent
config.request_timeout = 60

# Memproses scraping untuk setiap artikel
# df_article = pd.DataFrame()
with open('hasil_scraping.csv', 'a', encoding="utf-8", newline='') as csvfile:
    writer = csv.writer(csvfile)
      writer.writerow(['id','title',  'writer',  'publish_date',  'article_text',  'url',  'main_image', 'tag'])
    for index, response in tqdm(grequests.imap_enumerated(rs, size=5)):
        full_url = article_links[index]
        article = Article(full_url, config=config)
        article.download()
        article.parse()

        if response is None:
            print('Error: {}'.format(full_url))
            continue

        # Menggunakan XPath untuk informasi tambahan
        tree = html.fromstring(response.content)

        #XPath untuk informasi tambahan
        xpath_title = '//h1[@class="detail__title"]/text()'
        xpath_author = '//div[@class="detail__author"]/span/text()'
        xpath_publish_date = '//div[@class="detail__date"]/text()'
        xpath_main_image = '//div[@class="detail__media"]/figure/img/@src'
```

```
        xpath_tag = '//div[@class="detail__body-tag"]/a/text()'

        # Ekstraksi dengan XPath
        title = tree.xpath(xpath_title)
        author = tree.xpath(xpath_author)
        publish_date = tree.xpath(xpath_publish_date)
        main_image = tree.xpath(xpath_main_image)
        tag = tree.xpath(xpath_tag)
          # data = {'Judul': title, 'Penulis': author, 'Tanggal publikasi': publish_date, 'Teks berita': article.text, 'URL': full_url}
        # df = pd.DataFrame(data)

         writer.writerow([index + start_pos, title, author, publish_date, article.text, full_url, main_image, tag])
        # df_article = pd.concat([df_article, df])
```

Pada potongan Kode 3 menggunakan ‘user agen’ dan ‘config’ untuk mensimulasikan permintaan dari browser dan mengatur waktu tunggu permintaan HTTP. User agen yang digunakan adalah Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101 Firefox/78.0 dan timeout: 60. Kemudian menggunakan ‘Xpath’ untuk mengekstraksi informasi tambahan seperti judul, penulis, tanggal publikasi, gambar utama dan tag. Dari scraping data tersebut didapatkan total keseluruhan data berita sebesar 5.217 berita pada saat scraping. Namun, ditemukan 5 data yang kosong tetapi masih tersimpan pada file CSV. Sehingga data yang valid hanya 5212 data, sehingga 5 data yang hilang tidak di proses pada tahapan berikutnya. Data yang akan digunakan pada proses selanjutnya adalah data yang ada pada bagian *article_text.* Berikut merupakan Gambar 4 dari data hasil scraping data dan Gambar 5 dari satu teks berita hasil scraping yang didapatkan:

	title	**writer**	**Publish_date**	**Article_text**	**url**	**Main_image**
0	Timnas AMIN Ungkap	detikNews	Selasa, 28 Nov 2023 09.32 WIB	Jakarta - Calon Presiden	https://news.detik.com/p	-

	Alasan Tanah Merah Jakut Jadi Lokasi Pertama Kampanye Anies			nomor urut 1, Anies Baswedan me...	emilu/d-7060111/timnas-amin-ungkap-alasan-tanah-merah-jakut-jadi-lokasi-pertama-kampanye-anies	

Gambar 4 Hasil Scraping Data

Gambar 4 diatas merupakan gambar data yang telah dilakukan scraping dan disimpan dalam file SCV, dimana pada gambar tersebut terdapat informasi seperti *title* mengenai judul dari artikel yang di scrape*, writer* mengenai penulis atau kontributor artikel*, pubish_date* merupakan tanggal artikel tersebut dipublikasikan*, article_text* yang berisi teks lengkap dari artikel yang di scrape*, url* berisi tautan atau URL dari artikel *dan main_image* yang merupakan URL dari gambar utama terkait dengan artikel tersebut. Adapun teks yang diambil untuk pengolahan ada pada bagian article_text, berikut merupakan isi dari articel_text nya

Jakarta - Calon Presiden nomor urut 1, Anies Baswedan memulai kampanye hari pertama dengan mengunjungi kawasan Tanah Merah di Jakarta Utara. Kapten Tim Pemenangan AMIN (Timnas AMIN, M Syaugi menjelaskan Tanah Merah merupakan tempat bersejarah bagi Anies. "Ya betul (Tanah Merah) ini merupakan sejarah dari beliau yaitu mudah-mudahan ini menjadi pertanda baik seperti pada zaman Gubernur masa lalu,"

kata Syaugi di Pendopo Anies Baswedan di Jakarta Selatan, Selasa (28/11/2023).
Syaugi menyebut melalui tempat ini, Anies bisa meraih kesuksesan saat Pemilihan Gubernur DKI Jakarta periode 2017-2022. Sebabnya dia berharap Anies dapat mengulang sukses kemenangan ditempat ini.

Gambar 5 Potongan Teks Berita Hasil Scraping

Gambar 5 diatas merupakan contoh potongan hasil dari berita yang ada pada baris 0 dalam kolom article_text yang dilakukan scraping sebelumnya. Data yang sudah didapatkan selanjutnya akan dilakukan pra-pemrosesan data pada tahap berikutnya.

4.2 Pre-pemrosesan Data

Pada proses ini dilakukan untuk mempersiakan data yang belum diolah sebelumnya agar dapat digunakan secara efektif untuk dianalisis dan dimodelkan lebih lanjut serta merupakan langkah awal yang sangat penting untuk memastikan kesiapan data teks untuk diolah selanjutnya. Pada penelitian yang dilakukan ini ada 5 pre-pemrosesan yang dilakukan. Berikut 5 pre-pemrosesan data tersebut:

4.2.1 Removing function

Removal function atau penghapusan adalah menghilangkan tanda baca, karakter khusus, atau elemen teks lainnya yang tidak penting. Penghapusan tanda baca ini meliputi koma (,), titik (.), tanda seru (!), tanda tanya (?), kurung (), petik tunggal atau dua (‘/”) dan tanda lain sebagainya. Tujuan dari penghapusan tanda baca ini adalah untuk fokus pada kata-kata yang lebih relevan dan bermakna. Selain itu juga untuk menyederhanakan teks dan memastikan bahwa tanda baca tidak terlalu memberikan kontribusi yang sangat penting dalam penelitian ini. sehingga dilakukan penghapusan tanda baca agar analisis yang dilakukan pada tahap selanjutnya lebih konsisten makna dalam mengekstraksi informasi dalam representasi teks dan penghapusan tanda baca dalam analisis sentimen juga dapat membantu perhitungan sentimen lebih akurat. Berikut merupakan potongan kode pada saat *removing function* dapat dilihat pada Kode 4 berikut

Kode 4 Potongan Kode Removing Function

```
def remove_unused_char(texts):
  data = texts.map(lambda x: str(x).casefold())

  data = data.map(lambda x: re.sub(r'[-~\"\n]', r' ', str(x))) # Split word with dash
   data = data.map(lambda x: re.sub(r'[^a-zA-Z0-9 ]', r'', str(x))) # Remove unused character
   data = data.map(lambda x: re.sub('[!”#$%&’()*+,./:;<=>?@[\]^_`{|}]', '', str(x))) # Remove punctuation
  data = data.map(lambda x: re.sub('[0123456789]', '', str(x))) #Remove number
  data = data.map(lambda x: re.sub(r'@\S+', '', x)) # Remove mention
  data = data.map(lambda x: re.sub(r'#\S+', '', x)) # Remove hashtag
  data = data.map(lambda x: re.sub(r'https\S+', '', x)) # Remove URL
  data = data.map(lambda x: x.lstrip()) # lowercase

  return data
```

Pada saat *removing function* menggunakan fungsi ‘remove_unused_char’ yang bertujuan untukmenghapus karakter-karakter yang tidak diperlukan seperti tanda hubung (-), tilde (~), tanda kutip ganda ("), dan newline (\n) diganti dengan spasi, tanda baca, angka, mention, simbol, hastag, spasi diawal teks ‘lstrip()’ dan URL. Berikut merupakan cuplikan Gambar 6 hasil proses *removing function* yang dilakukan

Jakarta Calon Presiden nomor urut Anies Baswedan memulai kampanye hari pertama mengunjungi kawasan Tanah Merah Jakarta Utara Kapten Tim Pemenangan AMIN Timnas AMIN M Syaugi menjelaskan Tanah Merah merupakan tempat bersejarah Anies betul Tanah Merah merupakan sejarah beliau mudah-mudahan menjadi pertanda baik zaman Gubernur masa lalu kata Syaugi Pendopo Anies Baswedan Jakarta Selatan Selasa

Syaugi menyebut tempat Anies bisa meraih kesuksesan Pemilihan Gubernur DKI Jakarta periode Sebabnya berharap Anies mengulang sukses kemenangan tempat ini

Gambar 6 Hasil *Removing function*

Gambar diatas merupakan cuplikan potongan dari hasil *removing function* seperti penghapusan tanda baca, karakter khusus, mention hastag dan lainnya, dimana dapat dilihat hasil dari potongan teks berita tersebut lebih berfokus pada kata-kata penting yang mencerminkan isi utama artikel sehingga lebih mudah untuk diolah dan dianalisis pada tahap selanjutnya.

4.2.2 Case folding

Kemudian dilakukan *case folding* untuk mengubah semua huruf menjadi huruf kecil *(lower case)* untuk mempermudah proses analisis. *Case folding* ini dilakukan untuk meningkatkan konsistensi dalam teks dan mengurangi dimensi atau kompleksitas dalam analisis data sehingga analisis yang dilakukan lebih mudah dan efisien. Pengubahan semua huruf menjadi huruf kecil juga dapat memastikan bahwa semua huruf dapat bermkana sama baik huruf yang mengandung kapitalisasi ataupun huruf kecil (*lower case*) serta dapat menghindari kemungkinan duplikat makna atau entitas yang sama namun dengan kapitalisasi berbeda. Contohnya seperti kata “Presiden” dan “presiden” yang memiliki makna sama namun penulisan huruf yang berbeda. Dalam implementasi *word embedding* pada penelitian ini, *case folding* sangat membantu model untuk mengenali banyak model kata, sehingga dapat merepresentasikan vektor kata yang konsisten. Adapun untuk analisis sentimen yang dilakukan, *case folding* dapat memastikan bahwa konsistensi dalam identifikasi kata, memastikan analisis frequensi serta memastikan bahwa analisis semua kata dapat dikenal dan diberikan skor sentimennya dengan lebih akurat. Berikut merupakan potongan kode pada saat dilakukan *case folding*

Kode 5 Potongan Kode *Case folding*

```
def remove_unused_char(texts):
  data = texts.map(lambda x: str(x).casefold())


  data = data.map(lambda x: re.sub(r'[-~\"\n]', r' ', str(x))) # Split word with dash
   data = data.map(lambda x: re.sub(r'[^a-zA-Z0-9 ]', r'', str(x))) # Remove unused
```

```
character
  data = data.map(lambda x: re.sub('[!”#$%&’()*+,./:;<=>?@[\]^_`{|}]', '', str(x))) # Remove punctuation
  data = data.map(lambda x: re.sub('[0123456789]', '', str(x))) #Remove number
  data = data.map(lambda x: re.sub(r'@\S+', '', x)) # Remove mention
  data = data.map(lambda x: re.sub(r'#\S+', '', x)) # Remove hashtag
  data = data.map(lambda x: re.sub(r'https\S+', '', x)) # Remove URL
  data = data.map(lambda x: x.lstrip()) # lowercase


  return data
```

Potongan Kode 5 diatas merupakan potongan kode untuk *case folding* setelah dilakukan *removing function* sebelumnya semua teks diubah menjadi huruf kecil dengan menggunakan metode ‘casefold()’. Berikut merupakan hasil cuplikan Gambar 7 dari proses *case folding* yang dilakukan.

```
jakarta calon presiden nomor urut anies baswedan memulai kampanye hari pertama mengunjungi kawasan tanah merah jakarta utara kapten tim pemenangan amin timnas amin m syaugi menjelaskan tanah merah merupakan tempat bersejarah anies betul tanah merah merupakan sejarah beliau mudah-mudahan menjadi pertanda baik zaman gubernur masa lalu kata syaugi pendopo anies baswedan jakarta selatan selasa syaugi menyebut tempat anies bisa meraih kesuksesan pemilihan gubernur dki jakarta periode sebabnya berharap anies mengulang sukses kemenangan tempat ini
```

Gambar 7 Hasil *Case folding*

Gambar diatas merupakan contoh potongan hasil dari teks yang sudah dilakukan *case folding,* dimana semua huruf dalam teks diubah menjadi huruf kecil dengan tujuan untuk menyamakan format huruf dan menghindari perbedaan yang disebabkan oleh huruf kapital sehingga memudahkan untuk analisis selanjutnya.

4.2.3 Removing stop word

Removing stop word atau menghilangkan kata umum yang tidak memberikan banyak nilai informasi seperti “di”, “dan”, “ke”, ”dari”, “yang” dan lain sebagainya. Tujuan dari menghilangkan kata umum ini adalah untuk meningkatkan efisiensi pemrosesan dengan mengurangi ukuran data, memfokuskan pemrosesan pada kata kunci, meningkatkan akurasi model analisis

yang dilakukan. Pemrosesan *removing stop word* dalam *word embedding* pada penelitian ini dapat memungkinkan model untuk fokus pada kata-kata yanng memiliki makna subtantif. Dalam analisis sentimen, pemrosesan ini sangat membantu dalam pengekstraksian kata yang lebih bermakna serta menghindari kata-kata umum yang sering muncul sehingga dapat meningkatkan kualitas sentimen yang dihasilkan. Berikut merupakan potongan kode untuk *removing stop word*

Kode 6 Potongan Kode *Removing stop word*

```
stopword_factory = StopWordRemoverFactory()
stopword_remover = stopword_factory.create_stop_word_remover()

# print average number of word
print(df_data['prep_text'].map(lambda x: len(x.split())).mean())
print(df_data['prep_text'][0])

initial_avg_word = 0
while True:
  avg_word = df_data['prep_text'].map(lambda x: len(x.split())).mean()
  if avg_word == initial_avg_word:
    print(avg_word)
    break
  else:
    print(avg_word)
    initial_avg_word = avg_word
                df_data['prep_text'] = df_data['prep_text'].map(lambda x:
stopword_remover.remove(x))
print(df_data['prep_text'][0])
```

Pada saat proses *stop word* dilakukan dengan ‘StopWordRemoverFactory’ untuk membuat objek ‘stopword_remover’ dari teks, kemudian dihitung rata-rata jumlah kata dalam setiap teks dari ‘df_data’ dengan iterasi loop dan data di print dalam df_data ‘prep_text’. Berikut merupakan cuplikan Gambar 8 hasil dari *removing stop word* yang dilakukan

```
208.58769407705577 calon presiden nomor urut anies baswedan mulai kampanye...
208.58769407705577
204.98447383553767
204.9081847805252
204.9078014184397
204.9078014184397 calon presiden nomor urut anies baswedan mulai kampanye...
```

Gambar 8 Hasil *Removing stop word*

Gambar diatas merupakan potongan hasil dari *removing stop word* yang dilakukan, dimana tujuan dari proses ini adalah untuk memfokuskan teks pada kata-kata yang penting dan bermakna dalam analisis.

4.2.4 Stemming

Setelah beberapa informasi yang tidak penting pada tahap sebelumnya dihilangkan, selanjutnya dilakukan *stemming* untuk mengubah kata-kata menjadi bentuk kata dasarnya, seperti “lari”, “berlari” akan menjadi lari. Pada tahap ini kata dalam teks akan diubah kedalam kamus Bahasa Indonesia dengan *library open-source* yang sangat populer digunakan, yaitu Sastrawi. Tujuan dari langkah ini adalah untuk menyederhakan teks dan mengurangi variasi kata yang berbeda tetapi masih memiliki makna yang sama. Dalam model dalam *word embedding* pemrosesan *stemming* sangat dibutuhkan untuk dapat dengan mudah untuk memahami kata yang akan di representasikan. *Stemming* pada analisis sentimen juga sangat membantu dalam meningkatkan akurasi deteksi kata pada kamus sentimen sehingga perhitungan skor sentimen menjadi lebih akurat, dikarenakan variasi kata akan dipertimbankan sebagai entitas yang berbeda. Adapun potongan kode ketika dilakukan *stemming* sebagai berikut

Kode 7 Potongan kode *Stemming*

```
stemmer_factory = StemmerFactory()
stemmer = stemmer_factory.create_stemmer()


print(df_data['prep_text'][0])
stemmed_text = []
for text in tqdm(df_data['prep_text']):
  stemmed_text.append(stemmer.stem(text))
```

```
df_data['prep_text'] = stemmed_text
print(df_data['prep_text'][0])
```

Kode Kode 7 diatas merupakan kode yang dilakukan untuk proses *stemming*, dimana proses tersebut menggunakan 'StemmerFactory' pada objek data pada kolom 'prep_text' yang dilakukan dalam loop untuk semua teks, kemudia data disimpan dalam daftar 'stemmed_text'. Berikut merupakan Gambar 9 hasil dari *stemming* yang dilakukan

```
calon presiden nomor urut anies baswedan...
100%|██████████| 5217/5217 [1:52:53<00:00,  1.30s/it]
calon presiden nomor urut anies baswedan...
```

Gambar 9 Hasil *Stemming*

Gambar berikut merupakan hasil dari proses *stemming* yang dilakukan untuk mengubah kata menjadi kata dasarnya berdasarkan bahasa Indonesia berdasarkan kamus Sastrawi. Pada gambar tersebut proses *stemming* diproses untuk 5217 teks dalam waktu 1 jam 52 menit 53 detik.

4.2.5 Tokenization

Pre-pemrosesan terakhir yang dilakukan adalah *tokenization* atau tokenisasi yang memisahkan teks menjadi unit berupa kata yang lebih kecil atau yang disebut dengan token. Adapun tujuan dari tokenisasi ini adalah untuk memudahkan analisis teks yang diproses secara terpisah, selain itu dapat memudahkan pengukuran frekuensi kata dalam teks serta dapat memungkinkan pembentukan n-gram untuk menganalisis konteks atau hubungan antar kata yang dapat membuat teks lebih terstruktus dan siap untuk diolah dan dianalisis lebih lanjut. Pada penelitian ini sendiri dilakukan tokenisasi dengan cara unigram (kata tunggal). Dalam pengimplementasian pada *word embedding* tokenisasi merupakan langkah dasar yang menjadi input untuk melakukan representasi vektor kata, sehingga pada penelitian ini dilakukan pada pre-pemrosesan. Untuk melakukan analisis sentimen, tokenisasi dapat memungkinkan perhitungan frekuensi kata pada kamus sentimen yang digunakan sehingga dapat membantu penilaian pada sentimennya. Berikut merupakan kode yang dilakukan untuk proses tokenisasi kata

Kode 8 Potongan Kode Tokenisasi

```
tokenized_text = [word_tokenize(x) for x in df_data['prep_text']]
print(len(tokenized_text))
print(tokenized_text[:10])
```

Pada proses ini menggunakan ‘word_tokenize” dari NLTK untuk membagi setiap teks dalam kolom ‘prep_text’ menjadi token-token kata yang lebih kecil. Adapun contoh dari tokenisasi yang dilakukan pada teks berita pemilu pada penelitian ini seperti pada Tabel 3 dibawah ini:

Tabel 3 Contoh Tokenisasi

Sebelum tokenisasi	Setelah tokenisasi
calon presiden nomor urut anies baswedan mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang amin timnas	'calon', 'presiden', 'nomor', 'urut', 'anies', 'baswedan', 'mulai', 'kampanye', 'hari', 'pertama', 'unjung', 'kawasan', 'tanah', 'merah', 'jakarta', 'utara', 'kapten', 'tim', 'menang', 'amin', 'timnas',

Berikut merupakan contoh kalimat sebelum pre-pemrosesan data dan setelah pre-pemrosesan data

sebelum	**sesudah**
Jakarta - Calon Presiden nomor urut 1, Anies Baswedan memulai kampanye hari pertama dengan mengunjungi kawasan Tanah Merah di Jakarta Utara. Kapten Tim Pemenangan AMIN (Timnas AMIN, M Syaugi menjelaskan Tanah Merah	calon presiden nomor urut anies baswedan mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang amin timnas amin m syaugi jelas tanah merah

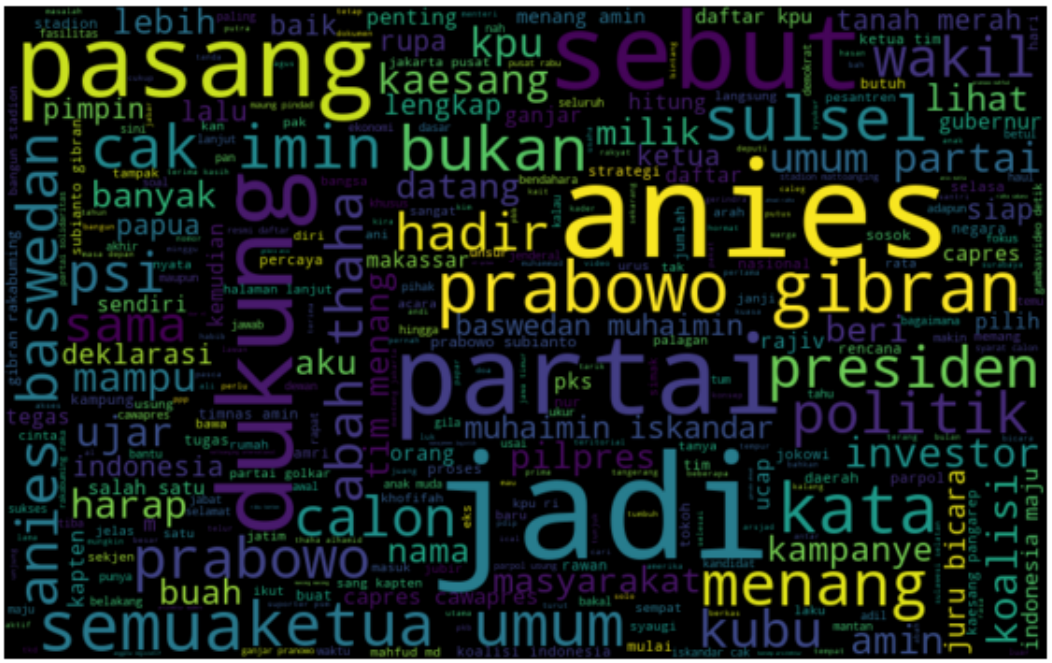
Setelah dilakukan dari ke-5 pra-pemrosesan data, selanjutnya dilakukan representasi visual kata teks dengan *word cloud* (awan kata) yang ditampilkan dengan ukuran yang lebih besar atau menonjol. Visualisasi ini dilakukan untuk melihat frequensi kata-kata yang sering muncul dalam teks dan mengidentifikasikan kata-kata, tema atau topik utama yang paling sering muncul dalam kumpulan teks atau dokumen kata dengan tujuan untuk memudahkan

pembaca untuk mengetahui topik yang dibahas tanpa membaca keseluruhan teks yang ada. Berikut merupakan kode ketika melakukan *word cloud*

Kode 9 Potongan Kode Word Cloud

```
from wordcloud import WordCloud
# Convert all elements in 'cleaned_text' to strings explicitly
all_words = ' '.join([str(text) for text in df_data['prep_text']])

# Generate the word cloud
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110,
max_words=len(tokenized_text)).generate(all_words)

print(wordcloud)

# Plot the WordCloud image
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

Kode Kode 9 merupakan kode yang dilakukan pada saat melakukan *word cloud* setelah semua proses pre-pemrosesan data dilakukan. Data yang dipakai adalah data pada kolom ‘prept_text’ dengan menggunakan parameter ‘width=800’ dan ‘height=500’ untuk menentukan ukuran gambar. ‘random_state=21’ untuk memastikan hasil konsisten pada setiap proses, ‘max_fonth_size=110’ untuk mengatur ukuran font dan ‘max_word’ untuk menentukan jumlah kata maksimum yang ditampilkan dalam *word cloud* kemudian dilakukan ‘generate’ untuk menggabungkan hasil kemduian ditampilkan dengan matplotib kode ‘plt.figure(figsize=(10, 27))’. Berikut merupakan Gambar 10 hasil dari representasi dari *word cloud* yang dilakukan pada teks berita mengenai pemilu presiden di Indonesia.



Gambar 10 Word Cloud

Dapat dilihat dari Gambar 10 diatas bahwa terdapat beberapa kata yang menonjol dan ditandai dengan beberapa warna tersendiri dalam menggambarkan katanya. Adapun beberapa kata yang menonjol seperti kata “anies”, “prabowo” dan “gibran” yang ditunjukkan dengan warna kuning dan dapat diasumsikan bahwa topik utama yang sering dibahas didalam kumpulan teks berita tersebut adalam tiga kata tersebut. Kata lain seperti “partai” yang digambarkan dengan warna ungu juga menonjol dan dapat diasumsikan juga menjadi kata yang sering dibahas, selain itu masih banyak lagi kata yang tergambar dalam visualiasasi tersebut yang menunjukkan tema atau topik yang sedang dibahas.

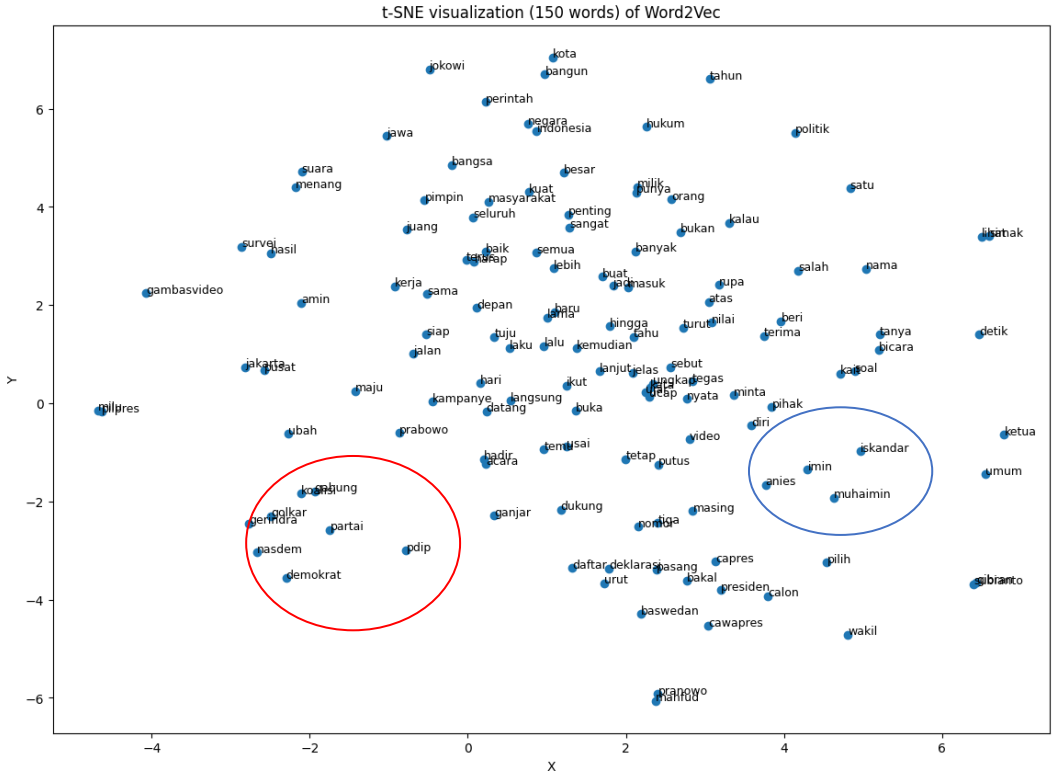
4.3 Implementasi Word embedding

Setelah pre-prosesing data dilakukan, selanjutnya data yang sudah bersih akan diimplementasikan pada *word embedding. Word embedding* ini dilakukan untuk mengubah kata menjadi vektor, sehingga kata yang sudah siap untuk dilakukan representasi dan akan dilakukan perhitungan hubungan antar kata dalam ruang vektor. Pada penelitian ini, pengimplementasian *word embedding* untuk melakukan representasi kata terkait berita pemilu presiden di Indonesia ini menggunakan Word2Vec dan IndoBERT yang masing-masing visualisasi menggunakan 150 kata. Berikut merupakan potongan Kode 10 nya

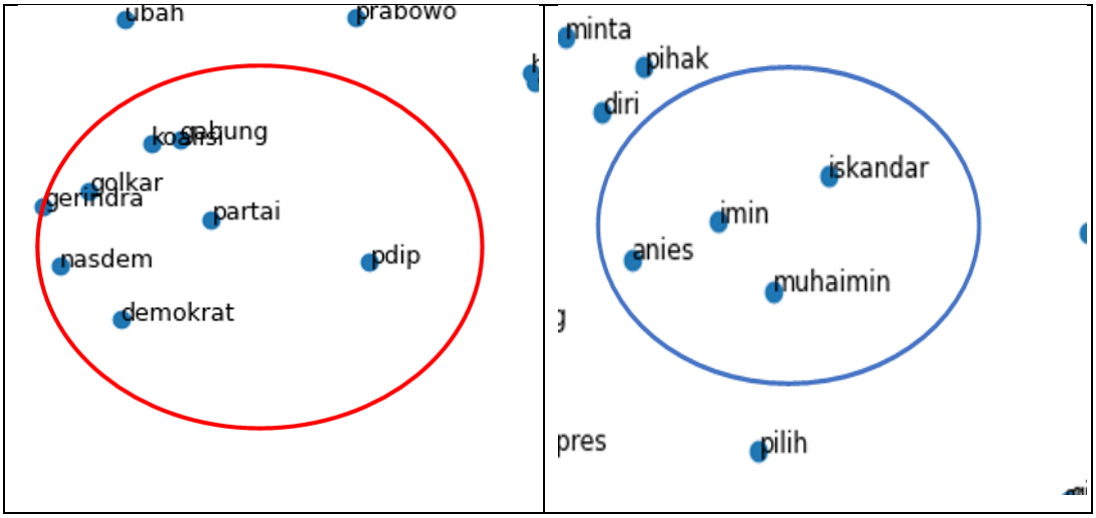
Kode 10 Potongan Kode Visualisasi t-SNE Word2Vec Sebelum *Debiasing*

```
# Create a DataFrame with t-SNE results
df = pd.DataFrame(tsne_result, index=words, columns=['x', 'y'])

# Filter out words with less than 4 characters
filtered_df = df[df.index.str.len() >= 4]

# Pilih 150 kata pertama setelah difilter
subset_df = filtered_df.head(150)

# Plot the t-SNE results
plt.figure(figsize=(14, 10))
plt.scatter(df['x'], df['y'])

# Annotate the points with words
for word, pos in df.iterrows():
    plt.annotate(word, pos, fontsize=9)

plt.title('t-SNE visualization (150 word) of Word2Vec')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

dimana dilakukan pembuatan data frame dengan ‘df_tsne = pd.DataFrame(tsne_result,index=unique_tokens,columns=['x','y']).reset_index().rname(columns={'index': 'word'}). ‘tsne_result’ adalah hasil dari algoritma tsne yang mengubah data embedding kata menjadi dua dimensi. unique_tokens adalah daftar kata yang sesuai dengan hasil tsne tersebut. Kemudian memilih subset dari data frame dengan mengambil 150 kata pertama dari data frame ‘subset_df_tsne = filtered_df_tsne.head(150)’ terkahir hasil tsne tersebut di plot dengan ‘plt.scatter’. Adapun hasil representasi berita menggunakan WordeVec dengan dengan t-SNE dapat dilihat pada Gambar 11



Gambar 11 Visualisasi t-SNE Word2Vec



Gambar 12 Contoh Detail Kumpulan Kata t-SNE Word2Vec

Visualisasi Gambar 11 pada t-SNE dari Word2Vec diatas merupakan hasil representasi dari vektor dalam ruang semantik. Pada visualisasi diatas dapat dilihat bahwa t-SNE sangat menjaga jarak antar kata yang emiliki hubungan semantik dekat, sehingga kumpulan-kumpulan kata yang memiliki konteks yang

sama cenderung lebih jelas terlihat. Gambar 12 merupakan gambar ketika dilihat lebih detail dari t-SNE Word2Vec pada kumpulan kata yang dilingkari dengan warna merah yang menunjukkan mengenai kumpulan kata terkait partai politik yang ada di Indonesia dan kumpulan kata warna biru yang merupakan kumpulan kata mengenai kandidat presiden. Kedua kumpulan kata ini mencerminkan dari teks berita yang dianalisis, yaitu berita terkait pemilu presiden di Indonesia. Berdasarkan visualisasi Gambar 12 diatas yang dilingkari dengan warna merah mengenai partai politik, seperti pada kata "gerindra", "nasdem", "golkar", "pdip", pkb", "pks", "golkar" yang jaraknya berdekatan satu sama lain. Kata-kata tersebut berkaitan eratdengan konteks mengenai partai politik yang ada di Indonesia. adapun pada gambar yang dilingkari dengan warna biru menunjukkan kandidat capre-cawapres pada pemilu presiden 2024 yang dibuat dalam kata, seperti "anies", "imin", "iskandar", "muahimin". Selain dilakukan implementasi dalam

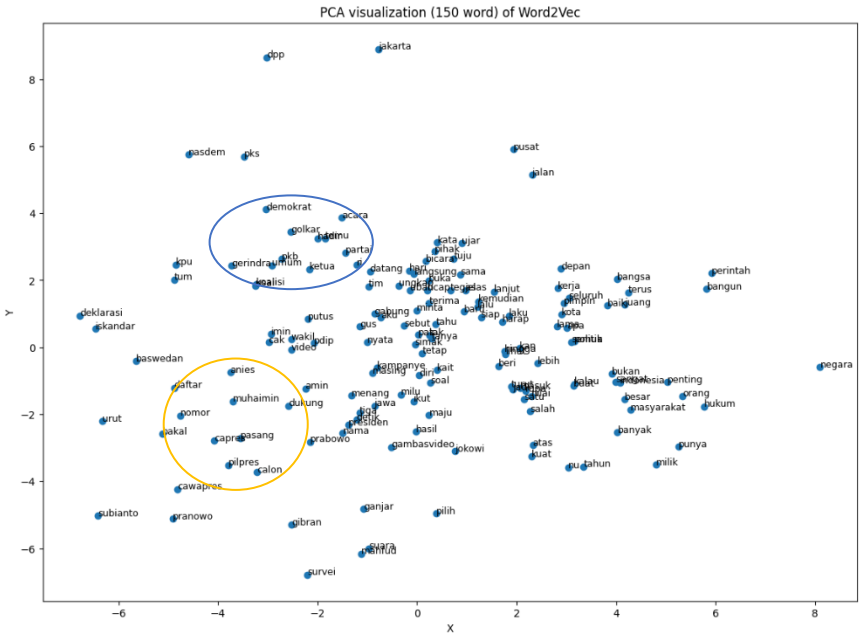
Kemudian potongan Kode 11 untuk PCA sebagai berikut

Kode 11 Potongan Kode Visualisasi PCA Word2Vec Sebelum *Debiasing*

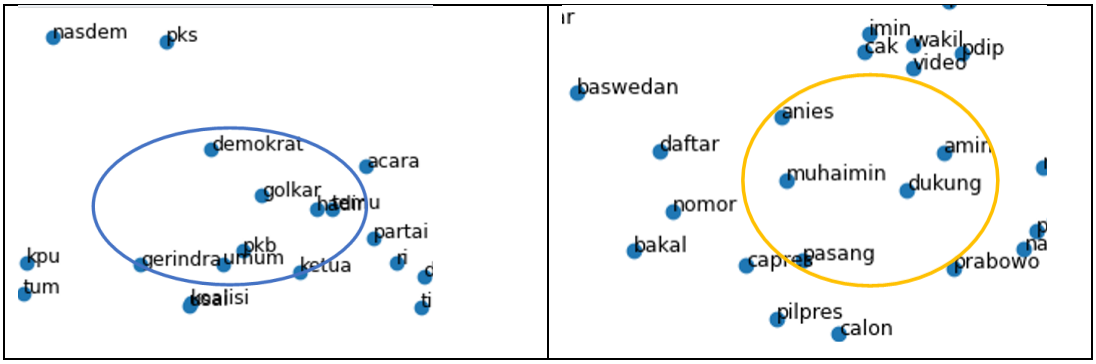
```
# Filter out words with less than 4 characters
filtered_df = df[df.index.str.len() >= 4]
# Pilih 150 kata pertama setelah difilter
subset_df = filtered_df.head(150)

# Plot the PCA results
plt.figure(figsize=(14, 10))
plt.scatter(pca_result[:150, 0], pca_result[:150, 1])  # Limit to top 150 words

# Annotate the points with words
for i, word in enumerate(words):
    plt.annotate(word, (pca_result[i, 0], pca_result[i, 1]), fontsize=9)

plt.title('PCA visualization (150 word) of Word2Vec')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

dimana dilakukan pembuatan data frame dengan ‘df_pca = pd.DataFrame(pca_result,index=unique_tokens,columns=['x','y']).reset_index().rname(columns={'index': 'word'}). ‘pca_result’ adalah hasil dari algoritma pca yang mengubah data embedding kata menjadi dua dimensi. unique_tokens adalah daftar kata yang sesuai dengan hasil pca tersebut. Kemudian memilih subset dari data frame dengan mengambil 150 kata pertama dari data frame ‘subset_df_pca = filtered_df_pca.head(150)’ terkahir hasil pca tersebut di plot dengan ‘plt.scatter’

Potongan Kode 11 diatas merupakan potongan kode untuk visualisasi PCA Word2Vec. Adapun hasil representasi berita menggunakan PCA pada Word2Vec dapat dilihat pada Gambar 13



Gambar 13 Visualisasi PCA Word2Vec



Gambar 14 Contoh Detail PCA Word2Vec

Visualisasi Gambar 13 diatas merupakan hasil visualisasi PCA dari Word2Vec yang direpresentasikan dalam bentuk dua dimensi (2D). Pada visualisasi diatas dapat dilihat bahwa kata-kata cenderung padat atau berkumpul lebih rapat di sekitar pusat grafik (pusat X (0) dan pusat Y (0), selain itu terdapat kata-kata yang mencerminkan konteks yang sama dengan jarak yang berdekatan. Gambar 14 diatas dapat dilihat lebih detail terkait kata yang berada pada lingkaran biru atas terdapat kata "demokrat", "golkar", "pkb", "gerindra" yang mencerminkan mengenai partai politik yang ada di Indonesia. Kata yang lain, seperti kata yang berada pada lingkaran kuning terdapat kata "muhaimin", "anies", "amin" yang memiliki jarak yang berdekatan dan secara keseluruhan kata tersebut masuk dalam konteks berita politik. Kata lain yang tersebar lebih jauh satu sama lain dalam visualisasi tersebut menunjukkan konteks yang masih sama, seperti kata "baswedan" memiliki konteks mengenai partai politik.

Setelah dilakukan visualisasi Word2Vec dilakukan penyimpanan nilai vektor dan dilakukan ektraksi model pada Word2Vec diatas digunakan untuk melatih model dengan ukuran dimensi vektor 100, ukuran jendela konteks di sekitar kata target 5, minimal kemunculan kata 1, workers untuk mempercepat proses pelatihan sbesar 4, sg yang digunakan sebagai algoritma pelatihan dengan CBOW dan penetapan nilai dengan dengan deafult seed yaitu 42. Kemudian model yang telah disimpan dengan dungsi 'model.save' dipanggil kembali dengan menggunakan fungsi 'model.load' dan simpan dalam bentuk .txt

Berikut merupakan Kode 13 Potongan Kode Load Hasil Word2Vec Kode 12 yang digunakan untuk menyimpan vektor nya

Kode 12 Potongan Kode Menyimpan Model Word2Vec

```
# Train the Word2Vec model
model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4, sg=0, seed=42)
# Menyimpan model ke file
model.save("word2vec_model.model")
```

Kode 13 Potongan Kode Load Hasil Word2Vec

```
model = Word2Vec.load("word2vec.model")
word_embeddings = np.array([word_vectors[word] for word in words])
word_vectors = model.wv
words = list(word_vectors.index_to_key)[:150]
word_embeddings = np.array([word_vectors[word] for word in words])
with open('embeddings_sebelum_word2vec.txt', 'w') as f:
    for word, vector in zip(words, word_embeddings):
        vector_str = ' '.join(map(str, vector))
        f.write(f"{word} {vector_str}\n")
```

Adapun contoh hasil dari ekstraksi fitur ditunjukkan pada Tabel 4 berikut. Hasil lengkap dapat dilihat pada zenodo (Yuniarti and Rakhmawati, 2024)

Tabel 4 Contoh Hasil Vektor Word2Vec Sebelum *Debiasing*

kata_vektor
imin 0.2665534 -0.18040197 -1.7828639 -2.1073384 0.11963004 0.10967675 -0.486972 0.8062492 -2.0386117 -2.312363 0.4826379 0.021996262 -0.09360887 -0.9375656 0.74619657 1.1706706 -1.3262998 -0.9612835 0.15904224 -0.45724893 0.86042327 …..
pkb 1.6835288 -1.5464444 0.92888737 -0.45506835 0.5591797 -0.6709837 -0.23713824 -0.13383114 0.021622915 -0.5214966 -0.008010393 1.0463396 0.12615469 -1.0395048 -1.0419137 1.9281152 -2.99997 -1.8445562 0.45055103 -1.1137831 -0.25547528 ….
amin 0.9164716 -1.9183075 0.66771656 -2.2027643 -0.8571693 -2.4717803 2.6106129 2.144811 -1.360063 -2.0246172 -0.91102904 -1.0664496 -0.9896942 -0.6454314 0.16858661 -0.16590834 -2.132867 0.048724193 0.33491397 0.33967352 1.0804327 ….

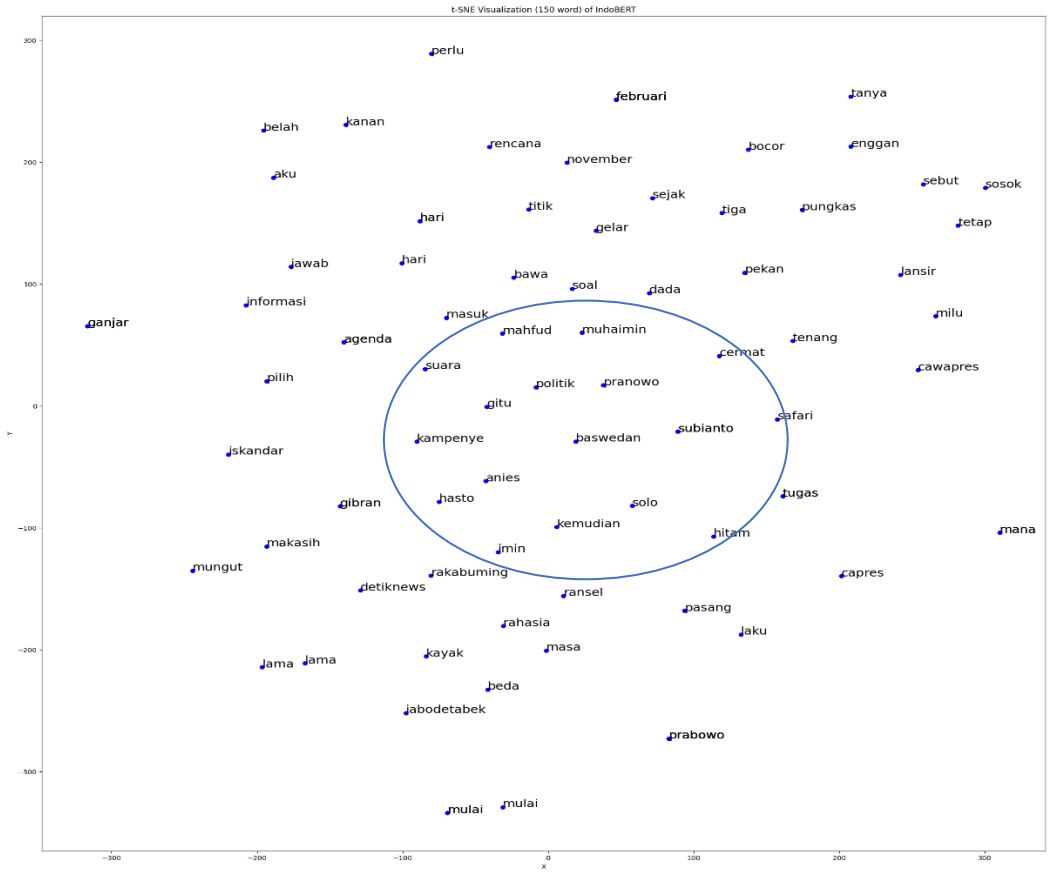
Selanjutnya dilakukan visualisasi pada *word embedding* dengan model IndoBERT, Berikut Kode 14 untuk visualisasi t-SNE indoBERT

Kode 14 Potongan Kode t-SNE IndoBERT

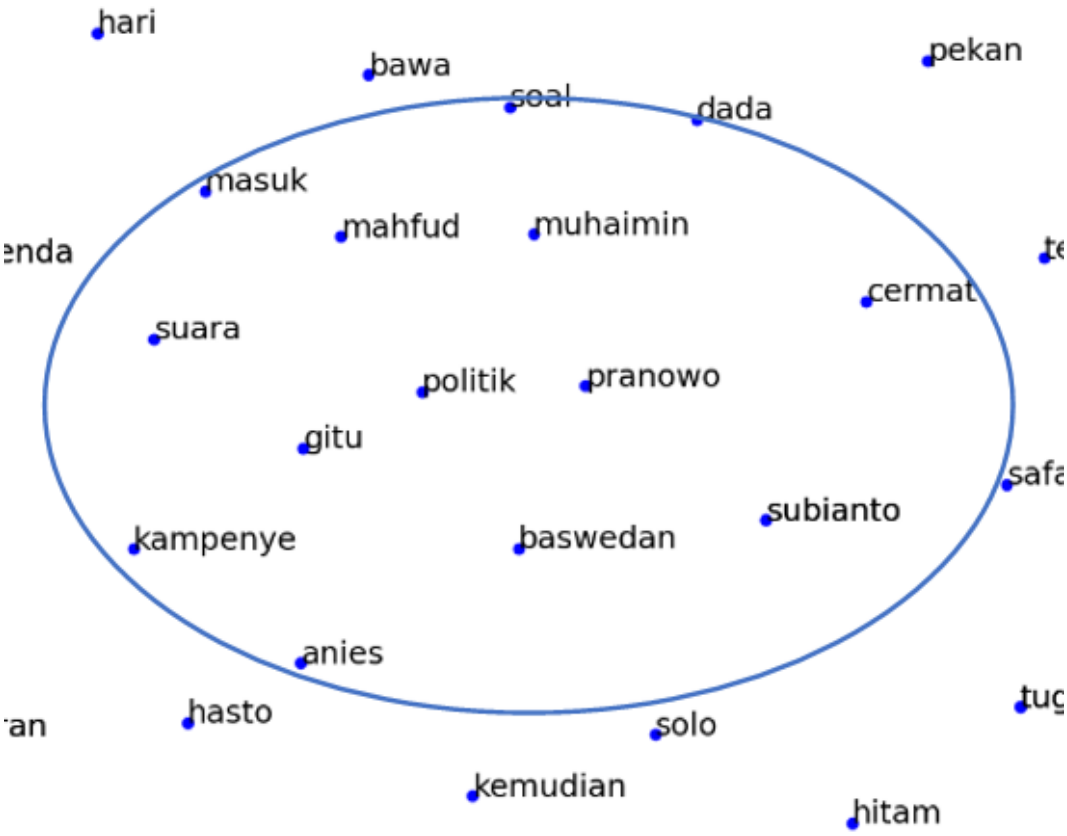
```
from sklearn.manifold import TSNE
```

```
tsne = TSNE(n_components=2, random_state=42)
tsne_embeddings = tsne.fit_transform(embeddings)
tsne_embeddings
import matplotlib.pyplot as plt

# Plot the reduced embeddings
plt.figure(figsize=(25, 25))
for i, word in enumerate(words):
    plt.scatter(tsne_embeddings[i, 0], tsne_embeddings[i, 1], color='blue')
    plt.text(tsne_embeddings[i, 0] + 0.01, tsne_embeddings[i, 1] + 0.01, word, fontsize=18)

plt.title('t-SNE Visualization (150 word) of IndoBERT')
# plt.xlim(-5, 8)  # Change these values based on your data range
# plt.ylim(-5, 8)
plt.xlabel('X')
plt.ylabel('Y')
#plt.grid(True)
plt.show()
```

Kode 14 merupakan potongan kode untuk visualisasi t-SNE dengn IndoBERT, dimana dilakukan pembuatan data frame dengan ‘tsne = TSNE(n_components=2, random_state=42)’ dilakukan untuk menginisiasi objek t-SNE untuk direkduksi. ‘tsne_embeddings = tsne.fit_transform(embeddings)’ yang digunakan untuk memperoleh reduksi *embeddings*nya. Selanjutnya dilakukan visualisasi dengan t-SNE menggunakan ‘plt.scatter’. Adapun hasil representasi berita menggunakan visualisasi t-SNE pada IndoBERT dapa dilihat pada Gambar 15.



Gambar 15 Visualisasi t-SNE IndoBERT



Gambar 16 Contoh Detail t-SNE IndoBERT

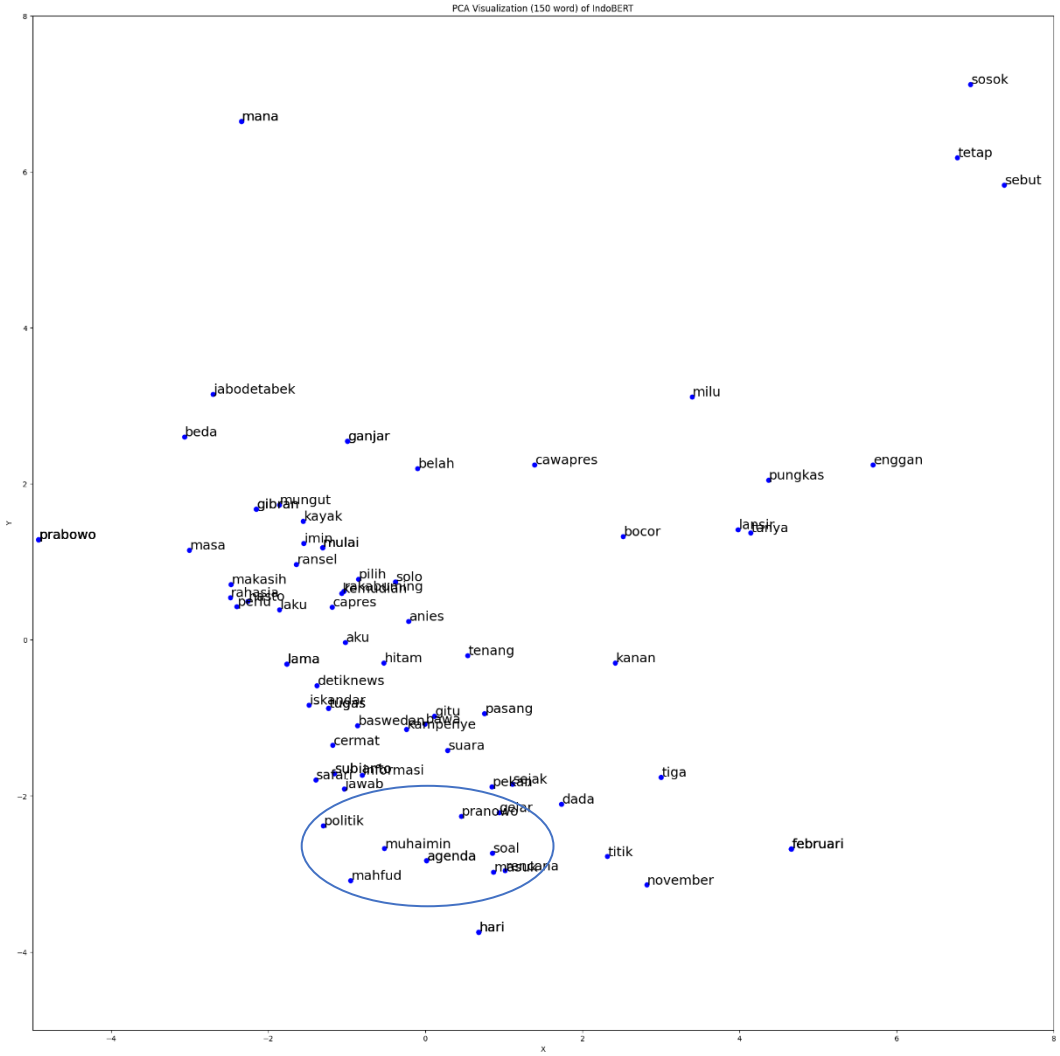
Adapun hasil visualisasi t-SNE pada IndoBERT dapat dilihat Gambar 15 bahwa terdapat beberapa kata yang berdekatan yang membahas mengenai politik dan berkumpul di pusat grafik. Secara detail dapat dilihat pada Gambar 16 diatas, dapat dilihat pada lingkaran berwarna biru menunjukkan kumpulan kata mengenai pemilu presiden di Indonesia, seperti nama kandidat presiden seperti “anies”, ‘”baswedan”, “mahfud”, “muhaimin”, “subianto”, “pranowo”, “kampanye”, “politik” . Berikut kode untuk melakukan visualiasi PCA pada IndoBERT

Kode 15 Potongan Kode PCA IndoBERT

```
from sklearn.decomposition import PCA


# Apply PCA to reduce to 2 dimensions
pca = PCA(n_components=2)
reduced_embeddings = pca.fit_transform(embeddings)
reduced_embeddings
```

```
import matplotlib.pyplot as plt

# Plot the reduced embeddings
plt.figure(figsize=(25, 25))
for i, word in enumerate(words):
    plt.scatter(reduced_embeddings[i, 0], reduced_embeddings[i, 1], color='blue')
    plt.text(reduced_embeddings[i, 0] + 0.01, reduced_embeddings[i, 1] + 0.01, word, fontsize=18)

plt.title('PCA Visualization (150 word) of IndoBERT')
plt.xlim(-5, 8)  # Change these values based on your data range
plt.ylim(-5, 8)
plt.xlabel('X')
plt.ylabel('Y')
#plt.grid(True)
plt.show()
```

Dimana dilakukan pereduksian PCA dengan kode “pca = PCA(n_components=2) reduced_embeddings = pca.fit_transform(embeddings)” kemudian dilakukan visualisasi PCA dengan kode selanjutnya. Adapun untuk visualisasi dengan PCA pada IndoBERT dapat dilihat pada berikut



Gambar 17 Visualisasi PCA IndoBERT

Gambar 18 Contoh Detail PCA IndoBERT

Visualisasi Gambar 17 diatas merupakan visualisasi PCA dengan model IndoBERT untuk merepresentasikan kata-kata dalam bahasa Indonesia. Setiap titik pada visualisasi diatas mewakili kata tertentu, adapun jarak-jarak tersentu pada visualisaisi diatas mencerminkan sejauh mana kata tersebut dianggap memiliki kemiripan oleh model IndoBERT. Dapat dilihat visualisasi diatas, kata-kata banyak berpusat pada pusat grafik (X (0) dan (Y (0)). Adapun visualisasi lebih dekat dapat dilihat pada Gambar 18 diatas dapat dilihat bahwa terdapat beberapa kata yang bedekatan seperti kata yang dilingkari dengan warna biru diatas, yang mencerminkan konteks politik, seperti mengenai nama kandidat "muhaimin", "mahfud", "pranowo", "ganjar".

Adapun hasil visualisasi kata pada PCA dan TSNE yang dilakukan pada Word2Vec dan IndoBERT dalam mengidentifikasi pola atau distribusi kata pada saat representasi memiliki perbedaan pendekatan dalam merepresentasikan kata, sehingga representasi kata yang ditampilkan pun berbeda. Model Word2Vec menggunakan metode pembelajaran yang terarah *(supervised learning)* dalam merepresentasikan kata dalam ruang vektor yang berdimensi tinggi berdasarkan konteks dari mana kata tersebut muncul. Pada representasi kata pada Word2Vec diatas memprediksi kata berdasarkan konteks sekitar, sehinggga saat dilakukan visualisasi dengan PCA menunjukkan beberapa kelompok kata yang cenderung sering muncul bersama dan memiliki hubungan semantik serta konteks yang sama. Pada saat visualisasi dengan t-SNE dapat dilihat sangat memperhatikan jarak kata yang cenderung tidak bertumpuk dalam visualisasi tetapi masih cenderung memiliki hubungan kontektual yang sama.

Model IndoBERT dalam merepresentasikan kata berbeda dengan model Wor2Vec. Model IndoBERT menggunakan model berbasis *Transformer* yang menggunakan arsitektur BERT yang menggunakan pendekatan tidak terarah *(unspervised learning)* dengan memahami konteks dua arah (dibirectional), sehingga pada visualisasi dengan PCA dapat dilihat bahwa pola distribusi kata lebih tersebar yang menangkap hubungan kontekstual yang lebih dalam dibandingkan dengan PCA pada Word2Vec. Pada visualisasi dengan t-SNE masih memperhatikan jarak dan menunjukkan kelompok yang cenderung berbeda dengan hubungan semantiknya. Dari kedua model tersebut dapat menampilkan

representasi kata dengan baik sehingga dapat memberikan pemahaman yang menyeluruh mengenai calon kandidat presiden, partai politik dan organisasi pendukung yang ada pada berita politik terkait pemilu presiden di Indonesia baik dalam visualisasi dengan t-SNE maupun dengan PCA yang dilakukan.

Setelah dilakukan visualisasi kata berita diatas, dilakukan pengambian vektor kata dengan memuat tokenizer yang telah dilatih sebelumnya dari model 'indobert-base-uncased' yang tidak membedakan huruf besar dan kecil, lebih akurat dalam memahami struktur dan idiom yang spesifik dalam bahasa Indonesia dari indobert dan menggunakan model yang sudah dilatih sebelumnya dari *chekpoint* yang sama yaitu 'indolem/indobert-base-uncased'. Pengambilan embeddings dilakukan dengan *last hidden layer* dengan fungsi 'last_hidden_state' yang dapat merepresentasikan kontekstual yang kaya serta lebih efisien dalam pengambilan emebddings dengan panjang maksimum 512 token dalam teks untuk mengurangi waktu komputasi dan memori dalam pemrosesan teks. Setelah itu dilakukan konversi teks menjadi array kata untuk memudahkan penyimpanan kata terakhir embedding disimpan dengan fungsi 'save_embeddings_to_file' dan emebddings akan disimpan dalam format .txt. Berikut merupakan potongan kodenya

Kode 16 Potongan Kode Embeddings IndoBERT

```
# Memuat tokenizer dan model IndoBERT yang telah dilatih sebelumnya
tokenizer = AutoTokenizer.from_pretrained("indolem/indobert-base-uncased")
model = AutoModel.from_pretrained("indolem/indobert-base-uncased")

# Fungsi untuk mendapatkan embeddings kata
def get_word_embeddings(words, max_length=512):
  inputs = tokenizer(words, return_tensors='pt', padding=True, truncation=True, max_length=max_length)
  with torch.no_grad():
    outputs = model(**inputs)
  # Mengambil embeddings dari last hidden state
  embeddings = outputs.last_hidden_state
  # Mengambil embeddings dari token pertama ([CLS])
```

```
    return embeddings[:, 0, :].numpy()

# Fungsi untuk membaca file teks dan mengonversinya menjadi array kata
def txt_to_word_array(filename):
    with open(filename, 'r') as file:
        content = file.read()
    words = content.split()
    return words

# Fungsi untuk menyimpan embeddings ke file teks
def save_embeddings_to_file(words, embeddings, filename):
    with open(filename, 'w') as f:
        for word, vector in zip(words, embeddings):
            vector_str = ' '.join(map(str, vector))
            f.write(f"{word} {vector_str}\n")

# Penggunaan contoh
filename = 'sebelum_debiasing.txt'
result = txt_to_word_array(filename)
words = result[:150]
embeddings = get_word_embeddings(words)

# Menyimpan embeddings ke file teks
save_embeddings_to_file(words, embeddings, 'Vektor_IndoBERT.txt')
```

Berikut merupakan contoh hasil dari pengambilam vektor dari model IndoBERT dapat dilihat pada tabel berikut. Lebih lengkap dapat dilihat pada zenodo (Yuniarti and Rakhmawati, 2024)

Tabel 5 Hasil Vektor Model IndoBERT Sebelum *Debiasing*

Kata_vektor
anies 0.6724698 -1.2966323 -1.142347 -0.054252006 -0.58826715 -0.16336522 -0.7015743 -0.06674037 2.2874014 0.13271603 -1.4463129 -0.96771973 -0.096001565 0.365588 0.034477256 0.16104454 0.6866186 1.7915136 -0.10252562 0.090993404 ….
amin 0.6353757 -1.1412323 -1.3564335 0.08725787 0.5958589 0.44780755 -0.6605006

0.38714358 1.144389 0.16178364 -2.0551345 0.23847656 -0.27286506 -0.31689754 -0.68202823 0.69153357 -1.1235257 -0.2788592 0.30416068 -0.9783985 -0.04208219 …
pkb 0.119966954 -0.99831045 -0.62748235 0.88891995 -0.58332705 0.2864304 -1.4242038 -0.24559413 1.7997913 -0.22741623 -0.6021398 -0.1788439 -0.021099092 0.5471926 -0.97455347 -1.2106196 0.3767721 0.04352097 0.123850144 -0.6282655 …

4.4 Identifikasi Bias

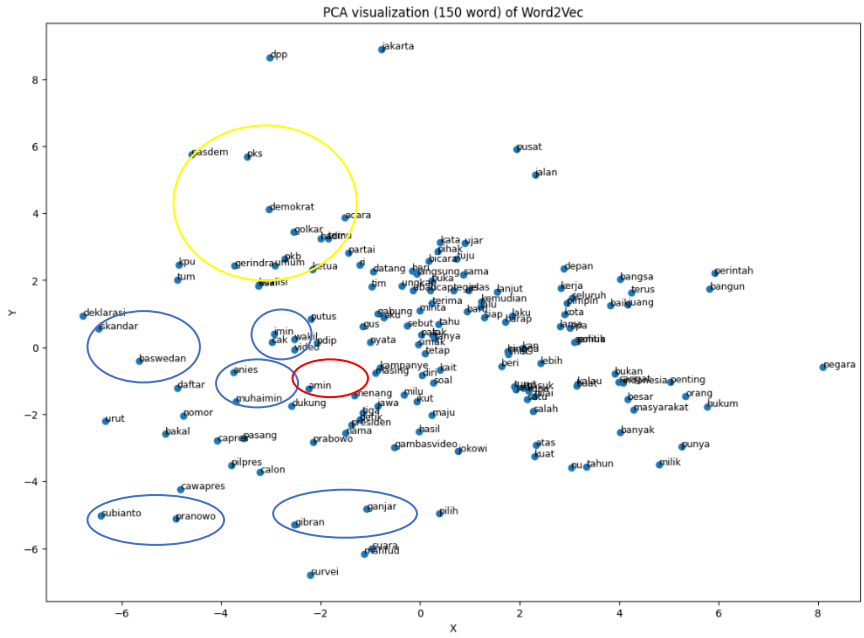
Bias dalam konteks berita mengacu pada penyajian informasi yang tidak netral, dimana terdapat kecenderungan atau preferensi tertentu yang memengaruhi cara berita tersebut diberitakan. Dalam konteks politik, berita yang mengandung bias merujuk pada kecenderungan media untuk memberitakan berita yang memihak, framing, mendukung, tidak mendukung atau merugikan pihak tertentu terkait kandidat, partai politik atau ideologi tertentu. Berita yang terdapat bias dapat memengaruhi persepsi publik terhadap isu politik atau tokoh politik tertentu sehingga dapat menyebabkan informasi yang tidak seimbang dan dipengaruhi dengan berita bias yang disediakan.

Dalam penelitian ini, bias dalam berita diidentifikasi berdasarkan kategori yang sudah ditentukan yaitu kemunculan nama kandidat capres-cawapres, partai politik dan organisasi tertentu pada berita sesuai dengan batasan penelitian yang dilakukan yakni, tidak memperhatikan konteks dari berita apa yang dibicarakan. Dimana terdapat beberapa kategori yang termasuk bias, seperti kategori nama orang terkait kandidat capres-cawapres Indonesia, organisasi tim suskes capres-cawapres dan partai politik yang terdapat pada berita. Identifikasi bias ini berdasarkan entitas (elemen spesifik) yang sudah dilabeli sebelumnya. Adapun kategori nama orang, partai politik dan organisasi yang terdapat dalam list kategori berikut didasari pada analisis dataset, dimana terdapat jumlah kategori nama orang terkait kandidat capres-cawapres sebanyak 6 pasangan calon (paslon), partai politik (parpol) yang ditemukan pada dataset sebanyak 10 partai politik dan organisasi ditemukan sebanyak 1. Adapun kategori dan contoh entitas tersebut dapat dilihat pada Tabel 6 berikut:

Tabel 6 Kategori Bias

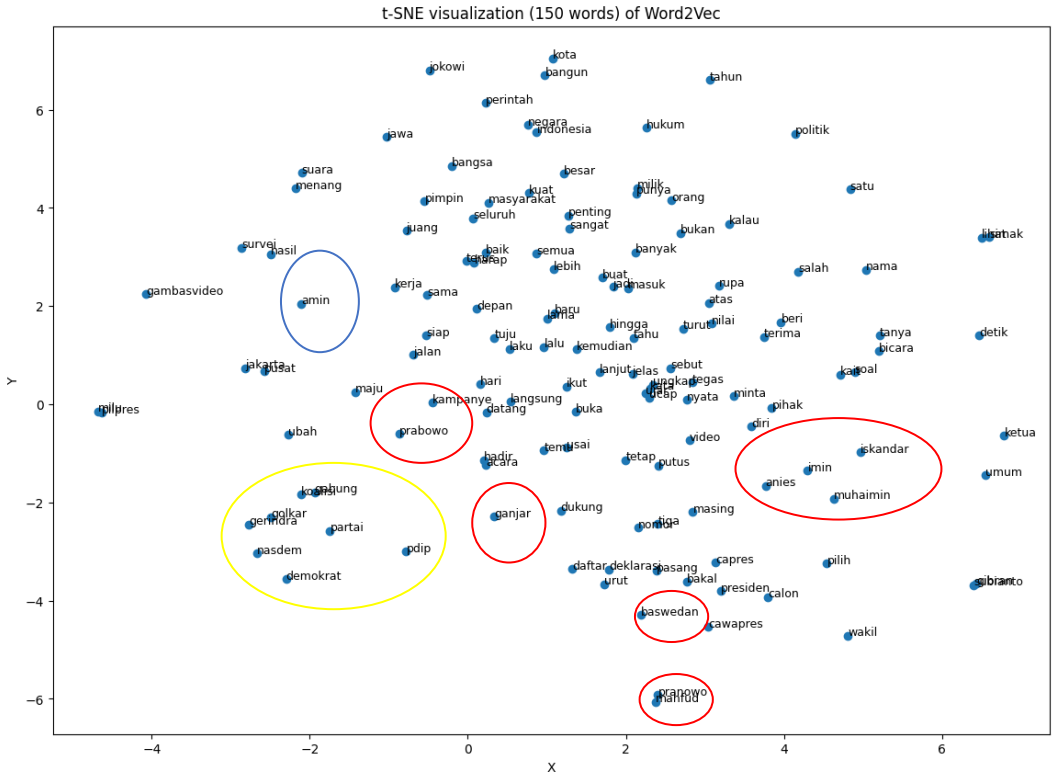
Kategori	Contoh Kata
Nama orang (kandidat capres-cawapres)	Prabowo Subianto (Prabowo)
	Gibran Rakabuming Raka (Gibran)
	Anies Rasyid Baswedan (Anies)
	Muhaimin Iskandar (Cak Imin)
	Ganjar Pranowo (Ganjar)
	Mahfud Md (Mahfud)
Partai politik	Partai Demokrat (Demokrat)
	Partai Demokrasi Indonesia Perjuangan (Pdip)
	Partai Golongan Karya (Golkar)
	Partai Gerakan Indonesia Raya (Gerindra)
	Partai Nasional Demorat (Nasdem)
	Partai Kebangkitan Bangsa (Pkb)
	Pertai Keadilan Sejahtera (Pks)
	Partai Amanat Nasional (Pan)
	Partai Persatuan Pembangunan (Ppp)
	Partai Bulan Bintang (Pbb)
Organisasi	Anis-Muhaimin (AMIN)

Berdasarkan representasi kata pada Word2Vec dan IndoBERT sebelumnya, maka dilakukan identifikasi bias berdasarkan hasil visualisasi pada PCA dan t-SNE. Pada visualisasi PCA dari model Word2Vec berdasarkan Gambar 19



Gambar 19 Identifikasi Bias berdasarkan PCA pada Word2Vec

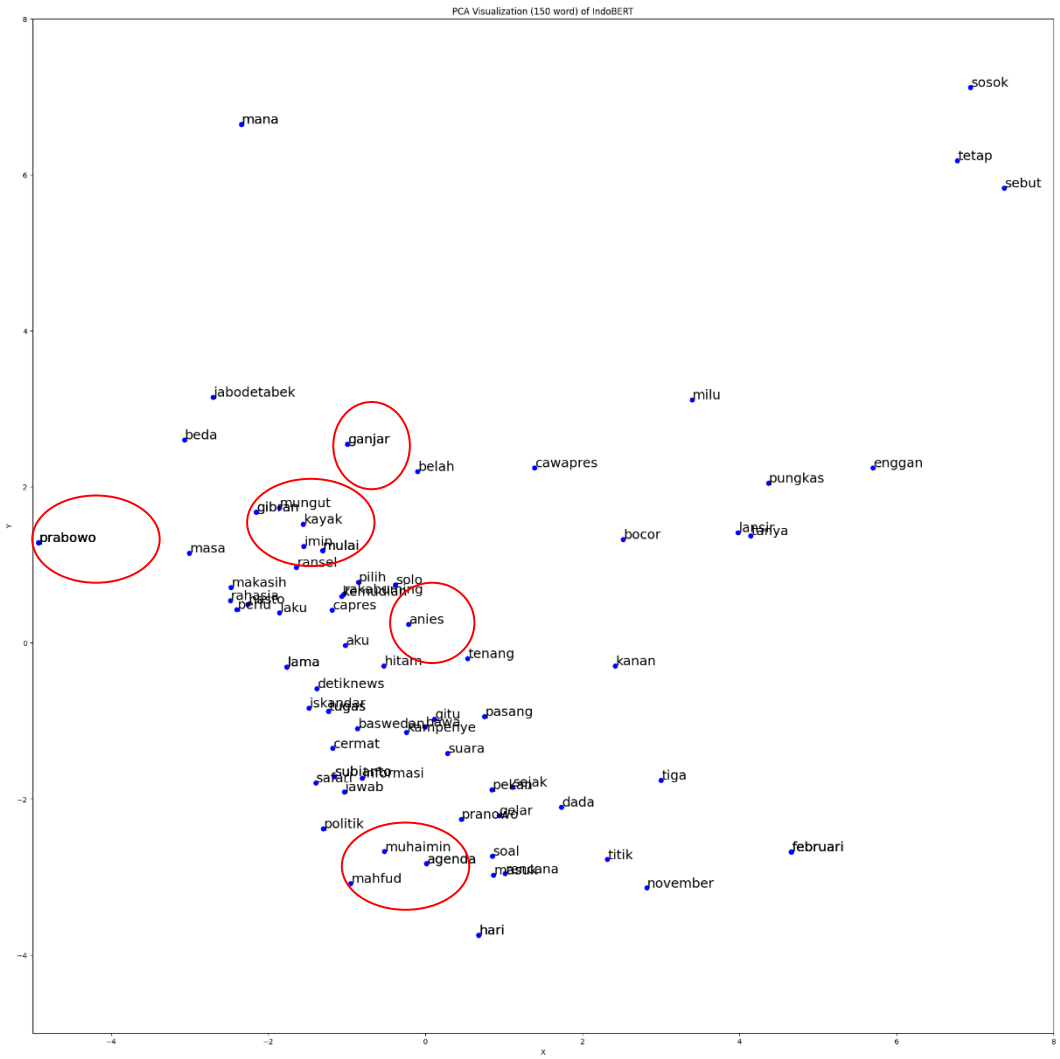
Berdasarkan visualisasi PCA Gambar 19 diatas terdapat beberapa kata yang menunjukkan identifikasi bias sesuai dengan kategori yang sudah ditentukan sebelumnya, nama orang, partai politik dan organisasi. Pada visualisasi diatas terdapat kata yang menunjukkan nama orang yang dilingkar dengan warna biru terdapat sebanyak 9 kata, kata tersebut seperti “baswedan”, “anies”, “muhaimin”, “subianto”, “pranowo”, “ganjar”, “mahfud”, “gibran”, “imin”. Kemudian mengenai partai politik yang dilingkari dengan warna kuning dalam visualisasi tersebut ditemukan sebanyak 6 kata, seperti kata “nasdem”, “pks”, “demokrat”, “golkar”, pkb”, “gerindra”, “demokrat”. Lalu kata organisasi yang tunjukkan dengan warna merah muncul sebanyak 1 kata, seperti kata”amin”. Lalu pada visualisasi dengan t-SNE dapat dilihat sebagai berikut



Gambar 20 Identifikasi Bias berdasarkan t-SNE pada Word2Vec

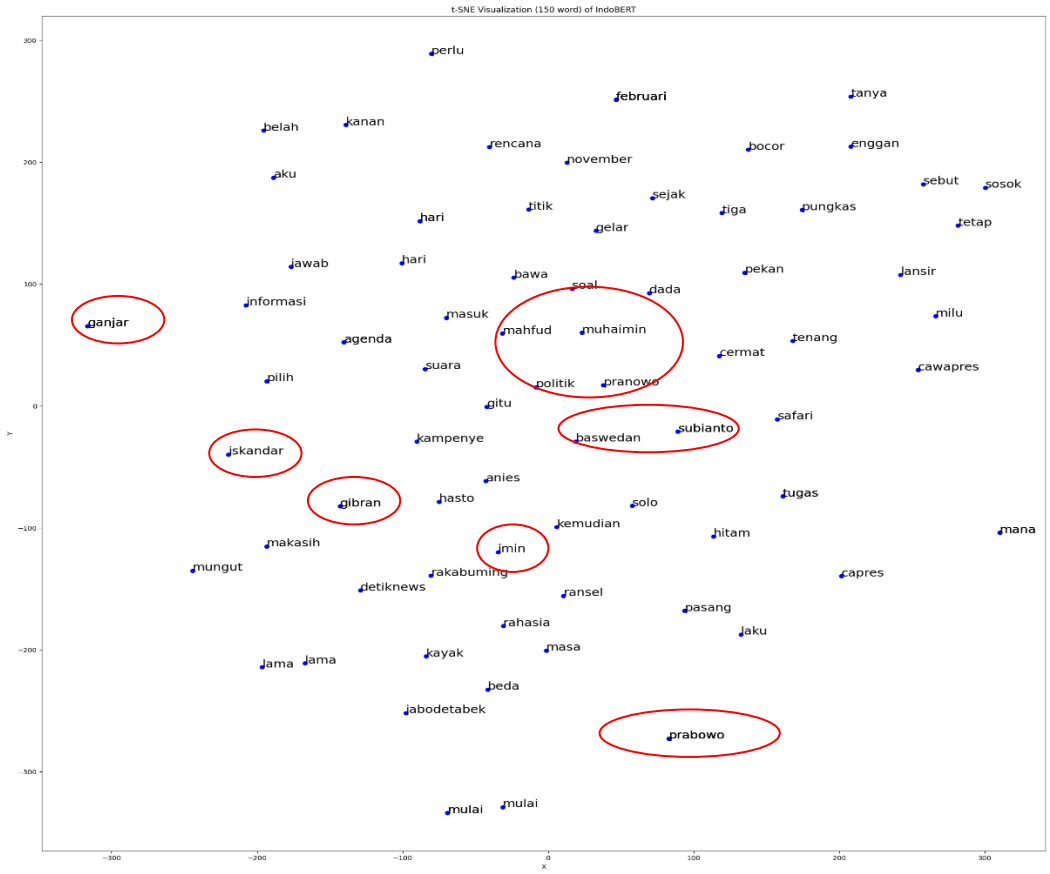
Pada visualisasi Gambar 20 diatas ditemukan sebanyak 9 kata yang termasuk bias dalam kategori nama orang yang dilingkari dengan warna merah, yaitu kata “anies”, “imin”, “muhaimin”, “iskandar”, “baswedan”, “mahfud”, “pranowo”, “prabowo”, “ganjar”. Lalu bias terkait partai politik dengan lingkaran warna kuning terdapat sebanyak 5 kata, seperti kata “golkar”, “pdip”, “demokrat”, “gerindra”, “nasdem”,. Dan bias terkait organisasi muncul sebanyak 1 kata yang dilingkari dengan warna bitu, yaitu kata “amin”.

Kemudian identifikasi bias dengan PCA pada IndoBERT dapat dilihat pada Gambar 21 berikut



Gambar 21 Identifikasi Bias dengan PCA pada IndoBERT

Pada visualisasi Gambar 21 diatas ditemukan 7 kata yang termasuk bias dalam kategori nama orang yang dilingkari dengan warna merah seperti “ganjar”, “mahfud”, “gibran”, “imin”, “muhaimin”, “prabowo”, “anies” dan tidak ditemukan bias terkait organisasi dan partai politik. Adapun pada visualisasi t-SNE pada IndoBERT dapat dilihat pada Gambar 22 dibawah ini



Gambar 22 Identifikasi Bias dengan t-SNE pada IndoBERT

Pada visualisasi Gambar 22 diatas terdapat kata yang menunjukkan bias terkait katagori nama orang (terkait kandidat) sebanyak 11 kata seperti “ganjar”, “prabowo”, “iskandar”, “gibran”, “imin”, “baswedan”, “subianto”, “mahfud”, “muhaimin”, “pranowo”. Pada identifikasi bias berdasarkan visualisasi *word embedding* tidak begitu terlihat banyak bias dikarenakan data yang digunakan sangat banyak, sehingga bias tidak dapat terdeteksi dengan baik. Oleh karena itu, dilakukan identifikasi bias kembali dengan cara menghitung jumlah entitas yang bias pada berita sebelum dilakukan *debiasing* guna untuk melihat banyaknya bias yang ada dalam berita tersebut. Berikut Tabel 7 merupakan hasil pengecekan jumlah bias tersebut

Tabel 7 Jumlah Kemunculan Bias

Kategori bias	**Bias**	**Jumlah kemunculan bias dalam berita (kali)**

Nama orang	Prabowo	10062
	Gibran	4250
	Rakabuming	7
	raka	20
	Mahfud md	1213
	Mahfud	414
	Ganjar	6408
	Pranowo	138
	Imin	20947
	Rayid Baswedan	2
	Anies	12857
	Baswedan	30
Partai politik	Partai Demokrat	1431
	Demokrat	375
	Pdip	2314
	Golkar	1605
	Gerindra	1264
	Nasdem	2089
	Pkb	9508
	Pks	2593
	Partai Amanat Nasional	61
	Pan	1278
	Ppp	589
	Pbb	320
Organisasi	Amin	3465
Total		83.240

Berdasarkan hasil perhitungan kemunculan entitas diatas dapat dilihat dari Tabel 7 diatas dapat dilihat bahwa jumlah kemunculan bias sangat banyak dalam berita baik terkait nama orang, partai politik dan organisasi dibandingan dengan visualisasi pada *word embedding* sebelumnya. Sebagai contoh dapat dilihat bahwa kemunculan entitas ‘prabowo’ pada kategori nama orang dalam berita muncul sebanyak 10062 kali, terkait kategori partai politik seperti ‘partai demokrat’

muncul sebanyak 1431 kali dalam berita dan kategori organisasi seperti ‘amin’ muncul sebanyak 3465 kali dalam berita.

4.5 Debiasing

Proses awal yang dilakukan dalam *debiasing* ini adalah labeling data. Pada proses pelabelan ini akan dilakukan pelabelan kata yang bias berdasarkan kriteria yang termasuk dalam Tabel 8 dengan tujuan untuk mengidentifikasi dan menandai entitas yang bias pada teks berita. Pelabelan entitas pada tahap ini menggunakan pelabelan *online* yang dapat custom tags sesuai dengan kebutuhan, setelah dilabeli maka dilakukan pengaplikasian pada NER (Named Entity Recognition) menggunakan SpaCy. Penggunaan NER dalam pelabelan sebelum *debiasing* dapat membantu memberikan struktur dan kejelasan dalam memahami konteks di mana entitas muncul pada teks serta NER dapat meningkatkan akurasi identifikasi entitas yang relevan sertaa lebih efisien dari segi waktu dan tenaga jika dibandingkan dengan debiasing manual untuk skala data yang besar, sehingga memungkinkan proses debiasing yang lebih terfokus dan efektif. Adapun entitas yang mengandung bias akan dilabeli dengan contoh sebagai berikut:

Tabel 8 Contoh Labeling Data

Entitas	Kategori	Label
Anies	Orang	ORANG
Cak imin	Orang	ORANG
Partai demokrat	Partai politik	PARTAI POLITIK
PDIP	Partai politik	PARTAI POLITIK
Amin	Organisasi	ORGANISASI

Berikut merupakan potongan gambar pada saat proses pelabelan menggunakan pelabelan *online* yang digunakan sebagai berikut:

Gambar 23 Proses Pelabelan

Pada gambar Gambar 23 diatas merupakan proses pelabelan entitas sesuai dengan tiga katagori yang sudah dijelaskan pada tabel Tabel 6 sebelumnya, pada proses pelabelan *online* kategori terkait nama orang dilabeli dengan warna merah, kategori organisasi dilabeli dengan warna hijau dan kategori terkait partai politik dilabeli dengan warna biru. Dalam penelitian ini, dari 5215 data berita dilakukan pelabelan data sebanyak 128 data teks berita. berikut merupakan kode pada saat diakukan pelabelan ke semua NER

Kode 17 Potongan Kode Proses Semua NER

```
# Load the trained spaCy NER model from the specified path
# nlp = spacy.load('//content/drive/MyDrive/Custom NER/trained_models/output/model-last')

# Nama file CSV
file_path = '//content/drive/MyDrive/Custom NER/data_bersih.csv'

# Membaca hanya kolom "prep data" dari file CSV
df_prep_data = pd.read_csv(file_path, usecols=['prep_text'])
ner_text = []

# len(df_prep_data)

for index, row in df_prep_data.iterrows():
  print("NER ke-"+str(index))
```

```
try:
  text = row['prep_text']


  # Process the extracted text using the loaded spaCy NER model
  ner_text.append(text)
except:
  print("Error NER for text-"+str(index))
```

Kode 17 diatas merupakan kode yang dilakukan untuk memproses pelabelan semua data pada 'prep_text'. Pelabelan 128 teks berita tersebut didasarkan pada asumsi bahwa jumlah data tersebut sudah cukup mewakili keberagaman entitas yang ada dalam dataset, seperti keberagam bias pada entitas nama orang dan partai politik yang sudah dilabeli cukup beragam. Setelah pelabelan data secara manual dengan media *online* selesai, selanjutnya dilakukan pengaplikasian pada model NER menggunakan SpaCy untuk dilakukan ekstraksi informasi secara otomatis pada teks berita lainnya sesuai dengan data yang telah di labeli sebelumnya. Adapun hasil labeling menggunakan NER dengan SpaCy sebagai berikut:

```
-=-=-=-=- NER Text-1 =--=-=-=-=-
anies baswedan   ->>>>   ORANG
amin   ->>>>   ORGANISASI
amin   ->>>>   ORGANISASI
m syaugi   ->>>>   ORANG
anies   ->>>>   ORANG
syaugi   ->>>>   ORANG
anies baswedan   ->>>>   ORANG
syaugi   ->>>>   ORANG
anies   ->>>>   ORANG
anies   ->>>>   ORANG
amin   ->>>>   ORGANISASI
syaugi   ->>>>   ORANG
muhaimin iskandar   ->>>>   ORANG
cak imin   ->>>>   ORANG
syaugi   ->>>>   ORANG
cak imin   ->>>>   ORANG
pkb   ->>>>   PARTAI POLITIK
anies   ->>>>   ORANG
anies   ->>>>   ORANG
anies   ->>>>   ORANG
anies   ->>>>   ORANG
anies   ->>>>   ORANG
```

Gambar 24 Contoh Hasil labeling NER

Setelah dilakukan pengaplikasian pada semua data menggunakan NER, dilakukan evaluasi untuk melihat keakuratan pendeteksian dari labeling yang sudah dilakukan, evaluasi yang dilakukan adalah dengan mengecek kinerja model pada saat training pada NER pada 128 data dengan training set sebanyak 102 data

dan testing set sebanyak 26 data, berikut merupakan hasil dari metrik evaluasi kinerja model tersebut:

Tabel 9 Evaluasi Metrik NER

E	#	LOSS TOK2VEC	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	69.64	3.66	7.14	2.46	0.04
25	200	213.01	4874.54	74.49	73.60	75.41	0.74
50	400	39.96	150.65	76.27	78.95	73.77	0.76
75	600	48.50	121.66	73.55	74.17	72.95	0.74
100	800	29.37	66.02	78.95	84.91	73.77	0.79
125	1000	72.57	123.73	77.27	86.73	69.67	0.77
250	2000	261.42	321.81	75.54	79.28	72.13	0.76
275	2200	238.04	223.20	82.14	90.20	75.41	0.82
300	2400	12.96	23.01	81.82	91.84	73.77	0.82
325	2600	59.77	79.88	78.92	87.13	72.13	0.79
350	2800	16.42	21.34	79.82	85.85	74.59	0.80
375	3000	111.13	82.48	80.52	85.32	76.23	0.81
400	3200	20.00	19.77	76.65	82.86	71.31	0.77
425	3400	0.00	0.00	77.39	82.41	72.95	0.77
450	3600	0.00	0.00	77.39	82.41	72.95	0.77
475	3800	0.00	0.00	77.39	82.41	72.95	0.77

Pada Tabel 9 diatas merupakan metrik evaluasi model NER, adapun penjelasan dari Tabel 9 diatas sebagai berikut:

- E : Epoch, yang menunjukkan iterasi ke-berapa dalam pelatihan.
- #: jumlah *batch* dari data pelatihan yang telah diproses.
- LOSS TOK2VEC: kerugian (loss) untuk tok2vec (token-to-vector)
- LOSS NER: kerugian (loss) untuk komponen NER
- ENTS_P: precision untuk entitas yang benar-benar dikenali model
- ENTS_F: F1 (gabungan precision dan recall) score untuk entitas yang dikenali oleh model.

- ENTS_R: recall untuk entitas yang benar-benar ada yang berhasil dikenali model.
- SCORE: skor keseluruhan model.

Pada Tabel 9 diatas merupakan hasil metrik evaluasi dari model, pada epoch 0 (awal pelatihan) belum ada *batch* dalam pelatihan, sehingga belum ada kerugian yang dihasilkan oleh token2vec pada tahap awal, kerugian NER cukup tinggi sebesar 69.64 dan skor F1, presisi dan recall sangat rendah dengan nilai F1 sebesar 3.66, presisi sebesar 7.14 dan recall sebesar 2.46, secara keseluruhan skor dari modelnya sangat rendah yaitu sebesar 0.04. Hal ini menunjukkan bahwa model belum banyak belajar dari data testing. Namun, pada epoch 25 kerugian NER turun signifikan menjadi 4874.54 dan skor F1 nya naik menjadi 74.49 dan secara keseluruhan model NER sebesar 0.74 yang menunjukkan adanya perbaikan dalam model dari sebelumnya. Dari keseluruhan model dapat dilihat bahwa kerugian model menurun dan metrik kinerja model, seperti F1, presisi dan recall meningkat yang menunjukkan bahwa model baik dalam mendeteksi entitas dari data.

Setelah dilakukan pelabelan maka akan dilakukan penggantian kata atau *debiasing* berdasarkan pelabelan kata yang bias dengan NER sebelumnya dan di gantikan dengan kata yang telah ditentukan agar lebih netral. Berikut merupakan Tabel 10 kriteria penggantian kata yang digunakan:

Tabel 10 Penggantian Data *Debiasing*

Kategori bias	**Entitas**	**Label**	***Debiasing***
Nama orang	Prabowo Subianto (Prabowo, Subianto)	ORANG	orang
	Gibran Rakabuming Raka (Gibran, Rakabuming, Raka)	ORANG	orang
	Mahfud Md (Mahfud)	ORANG	orang
	Ganjar Pranowo (Ganjar, Pranowo)	ORANG	orang
	Muhaimin Iskandar (Cak Imin, Muhaimin,	ORANG	orang

	Iskandar)		
	Anies Rayid Baswedan (Anies, Rasyid, Baswedan)	ORANG	orang
Partai politik	Partai Demokrat (Demokrat)	PARTAI POLITIK	partai politik
	Partai Demokrasi Indonesia Perjuangan (Pdip)	PARTAI POLITIK	partai politik
	Partai Golongan Karya (Golkar)	PARTAI POLITIK	partai politik
	Partai Gerakan Indonesia Raya (Gerindra)	PARTAI POLITIK	partai politik
	Partai Nasional Demokrat (Nasdem)	PARTAI POLITIK	partai politik
	Partai Kebangkitan Bangsa (Pkb)	PARTAI POLITIK	partai politik
	Partai Keadilan Sejahtera (Pks	PARTAI POLITIK	partai politik
	Partai Persatuan Pembangunan (Ppp)	PARTAI POLITIK	partai politik
	Partai Amanat Nasional (Pan)	PARTAI POLITIK	partai politik
	Partai Bulan Bintang (Pbb)	PARTAI POLITIK	partai politik
Organisasi	Amin	ORGANISASI	organisasi

Adapun kode yang dilakukan untuk *debiasing* sebagai berikut

Kode 18 Potongan Kode *Debiasing* NER

```
import spacy


# Load the trained spaCy NER model from the specified path
```

```
nlp = spacy.load('//content/drive/MyDrive/Custom NER/trained_models/output/model-last')

# List nama-nama yang akan diganti
names_to_replace = [
    "prabowo", "subianto", "gibran", "gibran rakabuming raka", "rakabuming", "raka",
    "mahfud md", "mahfud", "ganjar", "pranowo", "imin", "muhaimin", "muhaimin iskandar",
    "anies", "anies baswedan", "rasyid baswedan", "baswedan", "cak imin"
]

# List nama-nama partai politik yang akan diganti
parties_to_replace = [
    "partai demokrat", "demokrat", "pdip", "partai demokrasi indonesia perjuangan",
    "golkar", "partai golongan karya", "partai gerakan indonesia raya", "gerindra",
    "partai nasional demokrat", "nasdem", "partai kebangkitan bangsa", "pkb",
    "partai keadilan sejahtera", "pks", "partai amanat nasional", "pan",
    "partai persatuan pembangunan", "ppp", "pbb"
]

# List nama-nama organisasi yang akan diganti
organizations_to_replace = [
    "amin"
]

debiased_text = []
iterasi = 1

# Function to replace names in text
def replace_names(text, ent, names_to_replace):
    for name in names_to_replace:
        if name in ent.text.lower():
            text = text.replace(ent.text, "orang")
            break
```

```
    return text

# Function to replace parties in text
def replace_parties(text, ent, parties_to_replace):
    for party in parties_to_replace:
        if party in ent.text.lower():
            text = text.replace(ent.text, "partai politik")
            break
    return text

# Function to replace organizations in text
def replace_organizations(text, ent, organizations_to_replace):
    for organization in organizations_to_replace:
        if organization in ent.text.lower():
            text = text.replace(ent.text, "organisasi")
            break
    return text

# Open the output file once before the loop
with open("debiased_text2.txt", "a") as output:
    for text in ner_text:
        try:
            doc = nlp(text)
            print(f"-=-=-=-=-= NER ke-{iterasi} =-=-=-=-=-=-")
            iterasi += 1

            # Iterate over each entity in the document
            for ent in doc.ents:
                # Replace entities based on their labels and predefined names
                if ent.label_ == "ORANG":
                    text = replace_names(text, ent, names_to_replace)
                elif ent.label_ == "PARTAI POLITIK":
                    text = replace_parties(text, ent, parties_to_replace)
                elif ent.label_ == "ORGANISASI":
```

```
                text = replace_organizations(text, ent, organizations_to_replace)

            debiased_text.append(text)
            print("debiased text2 =", text)

            output.write(text + "\n")

        except Exception as e:
            print(f"Error processing text-{iterasi}: {e}")
```

Dimana, kode tersebut hanya melakukan *debiasing* pada kata yang dilabeli Adapun hasil potongan berita yang sudah di*debiasing* dapat dilihat pada Tabel 11 sebagai berikut:

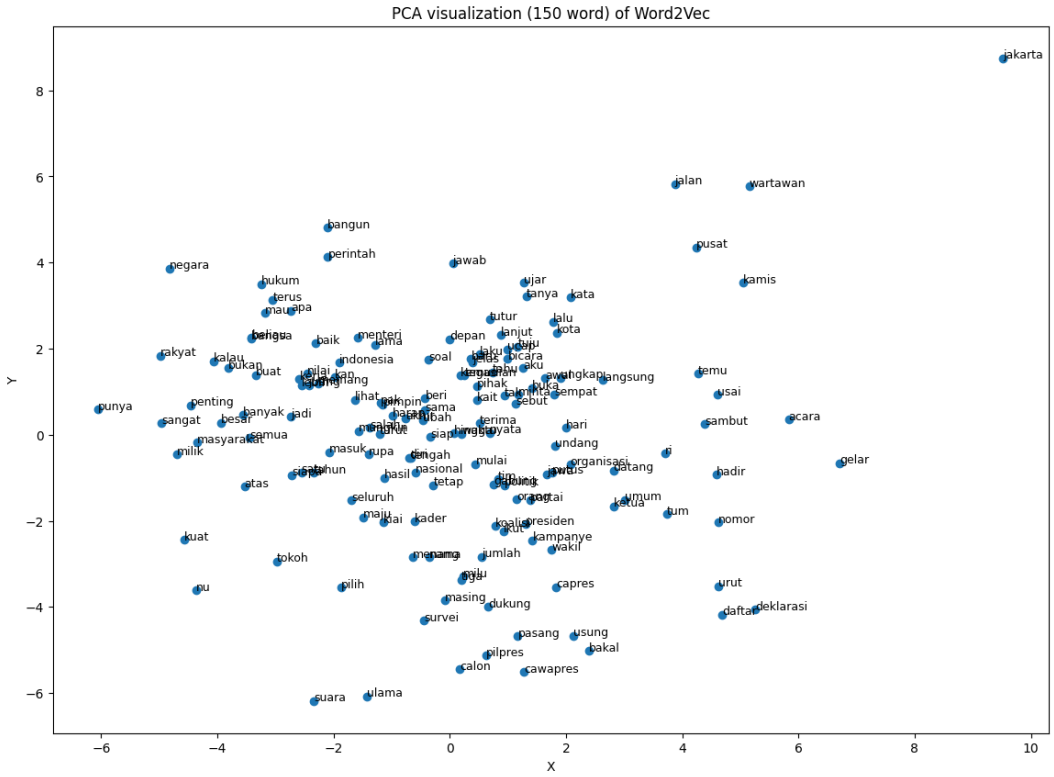
Tabel 11 Contoh Potongan Berita setelah *Debiasing*

Isi berita sebelum *debiasing*	Isi berita setelah *debiasing*
calon presiden capres nomor urut **anies baswedan** janji bangun mattoanging international stadium hadap suporter psm makassar janji sebut ikat penandatangan kontrak politik penandatangan kontrak politik bangun mattoanging international stadium laksana gori artisan building makassar sabtu khusus **anies** bahkan siap desain maket stadion bangun pilih jadi presiden ri	calon presiden capres nomor urut **orang** janji bangun mattoanging international stadium hadap suporter psm makassar janji sebut ikat penandatangan kontrak politik penandatangan kontrak politik bangun mattoanging international stadium laksana gori artisan building makassar sabtu khusus bahkan siap desain maket stadion bangun pilih jadi presiden ri
ketua umum **psi** kaesang pangarep sama dewan bina **psi** grace natalie isyana bagoes oka unjung tokoh agama islam jayapura thaha alhamid kata datang rangkul semua golong tanah papua semua kan rangkul anti intoleransi kata kaesang rumah abah thaha jayapura papua senin kaesang jelas sosok abah thaha rupa juru bicara pasang capres cawapres nomor **anies baswedan**	ketua umum **partai politik** kaesang pangarep sama dewan bina **partai politik** grace natalie isyana bagoes oka unjung tokoh agama islam jayapura thaha alhamid kata datang rangkul semua golong tanah papua semua kan rangkul anti intoleransi kata kaesang rumah abah thaha jayapura papua senin kaesang jelas sosok abah thaha rupa juru bicara pasang

muhaimin iskandar cak imin amin yakin abah thaha bakal beri dukung **psi**	capres cawapres nomor **orang orang orang** amin yakin abah thaha bakal beri dukung **partai politik**

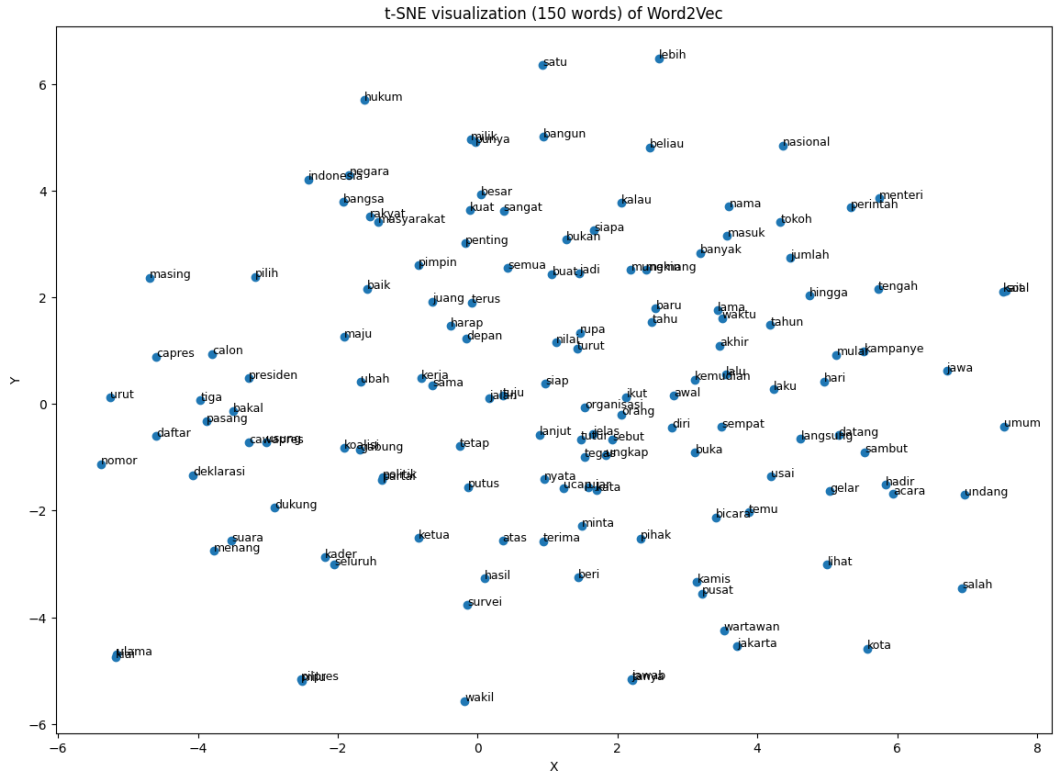
Tabel 11 merupakan contoh hasil dari berita yang belum dan sudah di *debiasing,* kolom isi berita sebelum *debiasing* menunjukkan potongan berita yang sudah ditandai bias dan akan digantikan menjadi kata yang lebih netral ditandai dengan kata yang becetak tebal dan kolom isi berita setelah *debiasing* menjelaskan tentang isi berita yang sudah di*debiasing* dengan kata yang lebih umum ditandai dengan kata yang bercetal tebal. Dapat dilihat bahwa kata yang bercetak tebal merupakan kata yang sudah di*debiasing* dengan kata yang lebih netral sesuai dengan ketentuan sebelumnya pada Tabel 10. Adapun contoh kata ‘anies’ diganti menjadi kata ‘orang’, kata ‘psi diganti menjadi ‘partai politik’ dan kata ‘amin’ diganti menjadi kata ‘organisasi’.

Setelah dilakukan proses *debiasing*, selanjutnya dilakukan kembali pengengecekan kata yang bias setelah *debiasing* dengan visualisasi pada 150 kata dengan PCA dan t-SNE menggunakan Word2Vec dan IndoBERT. Adapun hasil representasi kata setelah dilakukan proses *debiasing* pada Word2Vec dapat dilihat sebagai berikut



Gambar 25 Visualisasi PCA setelah Debiased

Pada Gambar 25 diatas merupakan visualisasi 150 kata pada Word2Vec setelah dilakukan proses *debiasing*. Pada Gambar 25 diatas dapat dilihat bahwa tidak ada kata yang termasuk dalam kategori bias (orang, organisasi atau partai politik) dalam representasi kata diatas. Kemudian dapat pada visualisasi menggunakan t-SNE dapat dilihat Gambar 26 pada berikut



Gambar 26 Visualisasi t-SNE Word2Vec setelah Debiased

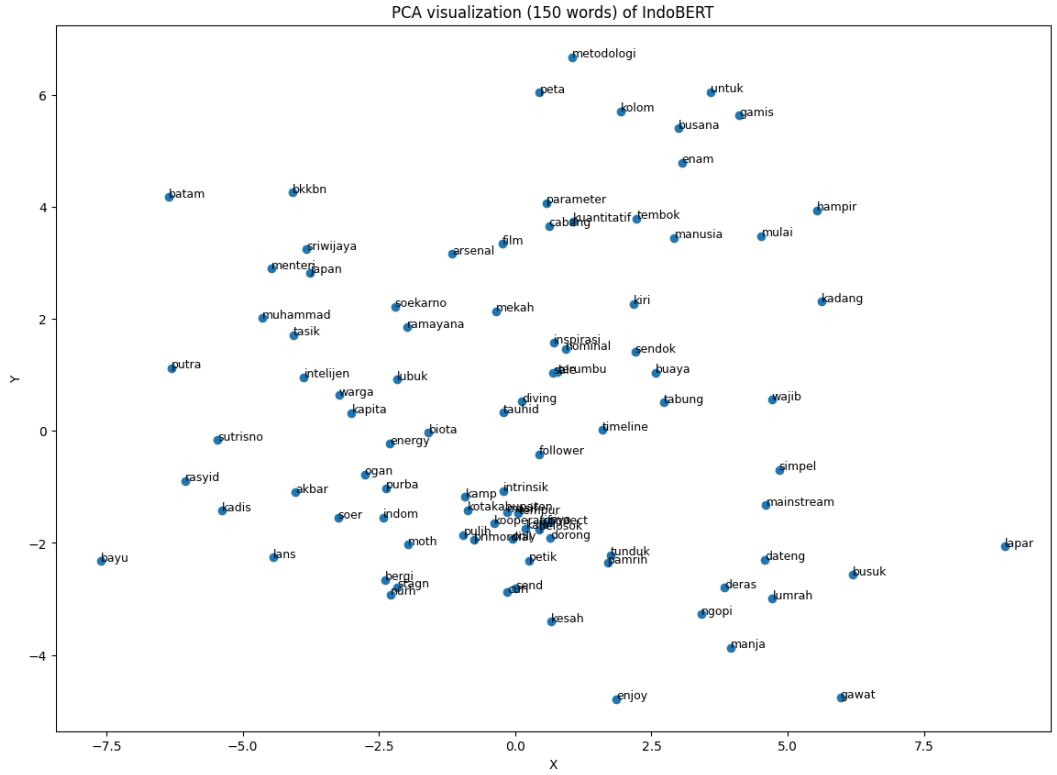
Pada gambar Gambar 26 diatas dapat dilihat bahwa pada visualisasi diatas juga tidak ada kategori bias (orang, organisasi atau partai politik) dalam representasi kata diatas. Setelah dilakukan representasi bias setelah melakukan *debiasing* diatas dilakukan pengambilan vektor pada kata yang sudah di*debiasing* dengan model Word2Vec, berikut Tabel 12 merupakan contoh hasil vektor kata yang sudah di*debiasing* berikut. Data yang lebih lengkap dapat dilihat pada zenodo (Yuniarti and Rakhmawati, 2024)

Tabel 12 Hasil Vektor Kata Word2Vec Setelah *Debiasing*

Kata_vektor
orang -1.7919993 -0.75585645 -0.23530123 -0.5471965 -0.92797095 0.6036797 -2.9496846 0.09206333 -1.2477518 -0.59605145 0.55658007 0.7203254 0.9722388 -0.29517817 0.066934526 0.04491824 1.7952992 -0.029754875 -0.89911973 …
partai -0.5950459 0.018881489 0.7602052 -1.3346157 0.92720246 2.7389424 -1.3585181 -0.522554 -1.5346799 -0.43356514 -0.23878813 -0.37522545 2.3639286 0.88415 0.63871175 1.3992571 -0.868354 -1.5772934 -2.7449117 -0.15598375 …

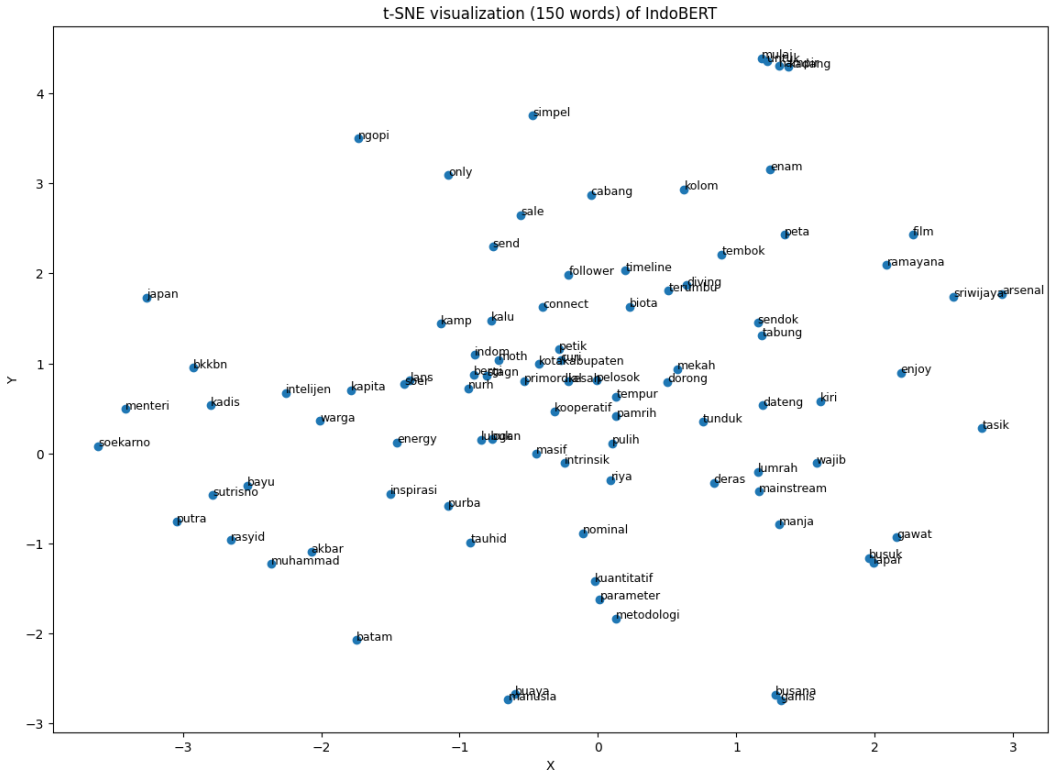
organisasi 0.5966423 -1.51447 -1.6771958 0.06251103 -0.4757628 1.7646862 -2.0917923 2.0081553 -1.0688295 -0.3381503 0.72088134 0.86758095 1.2708659 0.31085637 -1.6931707 -0.51845056 0.40188643 0.89140344 -1.9844824 -0.4158258

Adapun hasil visualisasi PCA dan t-SNE pada IndoBERT dapat pada Gambar 27 berikut



Gambar 27 Visualisasi PCA IndoBERT setelah Debiased

Pada Gambar 27 diatas dapat dilihat bahwa tidak ada bias berdasarkan representasi kata diatas. Selanjutnya, pada t-SNE dapat dilihat pada gambar dibawah



Gambar 28 Visualisasi t-SNE IndoBERT setelah Debiased

Pada Gambar 28 diatas dapat dilihat bahwa pada representasi kata diatas juga tidak ada kata yang merepresentasikan bias (orang, organisasi atau partai politik). Setelah dilakukan representasi bias setelah melakukan *debiasing* diatas dilakukan pengambilan vektor pada kata yang sudah di*debiasing* dengan model IndoBERT, berikut merupakan Tabel 13 contoh hasil vektor kata yang sudah di*debiasing* berikut. Data lebih lengkap dapat dilihat pada zenodo (Yuniarti and Rakhmawati, 2024)

Tabel 13 Hasil Vektor Model IndoBERT Setelah *Debiasing*

Kata_vektor
orang 0.51101124 -0.06594853 -0.033904407 0.05901861 0.2566596 -0.16190836 0.17609678 0.62716526 0.8151611 0.9077667 -1.6999695 0.19605383 0.09221752 0.87042636 -0.8953364 -1.0026771 -0.94511837 0.25407448 -0.60761124 -0.16969061...
organisasi 0.76193714 0.407702 -0.92929536 0.43181497 0.9172193 -0.30109093 0.3810981 0.6596441 1.332468 0.1612901 -3.0753539 -0.22605963 -0.14647055 -

0.0713483 -1.3541214 0.6683393 -0.4898442 1.0137593 0.6029091 -0.21086046 …
partai 0.18610957 0.1015562 -0.15483943 0.8245009 0.76713187 -1.0802267 -2.2557724 -1.4674615 1.1325992 0.16479737 0.5088599 -0.35931793 0.5669724 0.69897306 -0.30377012 0.43848512 0.10170061 -0.9577311 0.03514536 -0.880551 -0.4685978 -1.6496868 0.74779576 0.72137487 -0.8610119 -0.6171747 -0.6704368 -

Selain dilakukan pengecekan menggunakan *word embedding* diatas dilakukan juga pengecekan jumlah kemunculan entitas yang bias pada berita yang sudah dilakukan proses *debiasing*, hal ini dilakukan untuk memastikan apakah bias masih ada ketika sudah dilakukan *debiasing*. Adapun hasil dari pengecekan kemunculan bias sebagai berikut

Tabel 14 Pengecekan Kemunculan Bias Setelah *Debiasing*

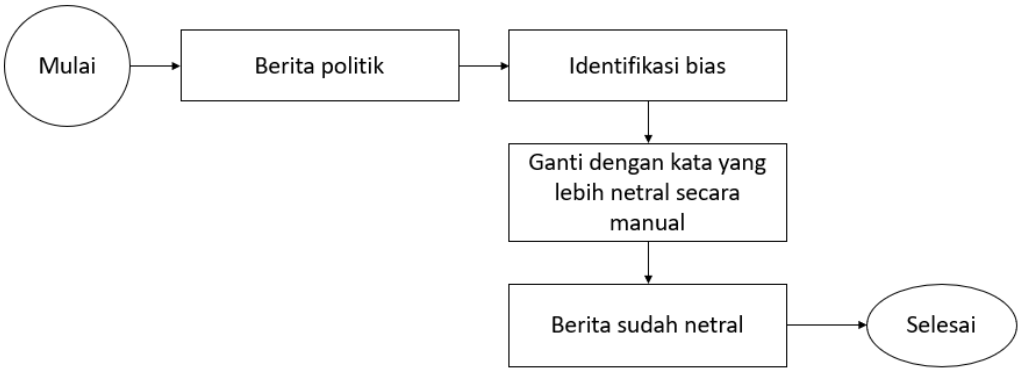
Kategori bias	**Bias**	**Jumlah kemunculan Bias (kali)**
Nama orang	Prabowo Subianto (Prabowo, Subianto)	Prabowo 16, Subianto 196
	Gibran Rakabuming Raka (Gibran, Rakabuming, Raka)	Gibran 6, Rakabuming 136, Raka 111
	Mahfud Md (Mahfud)	Mahfud Md 2, Mahfud 1023
	Ganjar Pranowo (Ganjar, Pranowo)	Ganjar 30, Pranowo 146
	Muhaimin Iskandar (Cak Imin, Muhaimin, Iskandar)	Cak Imin 2, Muhaimin 85, muhaimin Iskandar 67
	Anies Rayid Baswedan (Anies, Rasyid, Baswedan)	Anies 36, Rasyid Baswedan 38, baswedan 244
Partai politik	Partai Demokrat (Demokrat)	Demokrat 777
	Partai Demokrasi Indonesia Perjuangan (Pdip)	Pdip 777
	Partai Golongan Karya (Golkar)	Golkar 17
	Partai Gerakan Indonesia Raya (Gerindra)	Gerindra 662
	Partai Nasional Demokrat	Nasdem 138

	(Nasdem)	
	Partai Kebangkitan Bangsa (Pkb)	Pkb 4
	Partai Keadilan Sejahtera (Pks	0
	Partai Persatuan Pembangunan (Ppp)	Ppp 3
	Partai Amanat Nasional (Pan)	Pan 15
	Partai Bulan Bintang (Pbb)	Pbb 7
Organisasi	Amin	Amin 38
Total		3915

Setelah proses *debiasing* dilakukan pengecekan kembali dengan cara menghitung jumlah kemuncul bias kembali setelah dilakukan *debiasing* untuk memastikan apakah penggantian kata yang dilakukan sudah berhasil untuk mengubah bias dalam teks. Dapat dilihat pada Tabel 14 diatas bahwa bias terkait nama orang, partai politik dan organisasi masih ada, dibuktikan dengan contoh bias kategori orang pada kata ‘prabowo’ setelah dihitung kemunculan biasnya muncul 16 kali dalam berita, kemudian pada kategori partai politik ‘ppp’ setelah dihitung kemunculan biasnya sebanyak 3 kali dan pada kategori organisasi pada kata ‘amin’ setelah dihitung kemunculan biasnya 38 kali dalam berita. Dari hasil perhitungan kemunculan bias setelah dilakukan *debiasing* terdapat pengurangan dari sebelum debiasing berjumlah 83.240 setelah didebiasing sisa bias sebanyak 3.915, total kata bias yang berhasil di*debiasing* sebanyak 79.325 kata. Adapun kemunculan bias tetap ada meskipun setelah proses *debiasing* disebabkan oleh beberapa hal seperti kesalahan dalam melabeli entitas, keterbatasan data katihan yang digunakan untuk melatih model NER dalam dataset yang besar sehingga tidak semua bias dapat ter*debiasing.* Berdasarkan hasil yang didapat dapat simpulkan bahwa proses *debiasing* yang dilakukan berhasil.

Selain dilakukan *debiasing* untuk keseluruhan data dengan otomatis, dilakukan juga *debiasing* dengan cara manual dengan masing-masing data sebanyak 20 berita pertama dari keseluruhan berita (5212), kemudian dilakukan perbandingan dari hasil kedua cara tersebut untuk melihat efisiensi dari kedua

metode *debiasing* manual dan otomatis. Adapun untuk cara manual alur kerjanya dapat dilihat pada Gambar 29 sebagai berikut



Gambar 29 Cara Kerja *Debiasing* Manual

Proses *debiasing* secara manual dilakukan dengan beberapa langkah seperti, sebelum dilakukan *debiasing* berita politik diidentifikasi terlebih dahulu terkait bias yang ada pada berita sesuai dengan kategori pada Tabel 6 sebelumnya. Kemudian bias yang sudah diidentifikasi sebelumnya dilakukan *debiasing* secara manual dengan menggantikan dengan kata yang lebih netral sesuai ketentuan pada Tabel 10. Setelah itu berita sudah netral, adapun pengecekan berita yang sudah netral dilakukan dengan mengecek kata yang berhasil terganti pada proses *debiasing.* Berikut merupakan contoh dari berita yang dilakukan proses *debiasing* manual

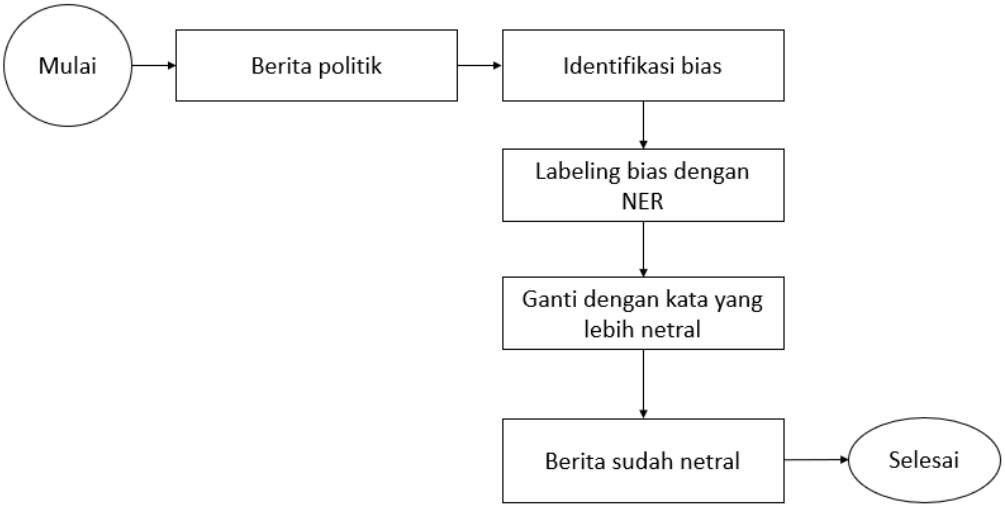
Tabel 15 Contoh Potongan Berita *Debiasing* Manual

Berita	Hasil *debiasing* manual
calon presiden nomor urut **anies baswedan** mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang **amin** timnas **amin** m syaugi jelas tanah merah rupa tempat sejarah **anies** betul tanah merah rupa sejarah beliau mudah mudah jadi tanda baik zaman gubernur masa lalu kata syaugi pendopo **anies baswedan** jakarta selatan	calon presiden nomor urut **orang** mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang **organisasi** timnas **organiasi** m syaugi jelas tanah merah rupa tempat sejarah **orang** betul tanah merah rupa sejarah beliau mudah mudah jadi tanda baik zaman gubernur masa lalu kata syaugi pendopo **orang** jakarta selatan

ketua umum **psi** kaesang pangarep sama dewan bina **psi** grace natalie isyana bagoes oka unjung tokoh agama islam jayapura thaha alhamid kata datang rangkul semua golong tanah papua semua kan rangkul anti intoleransi kata kaesang rumah abah thaha jayapura papua senin kaesang jelas sosok abah thaha rupa juru bicara pasang capres cawapres nomor **anies baswedan muhaimin iskandar cak imin amin**	ketua umum **partai politik** kaesang pangarep sama dewan bina **partai politik** grace natalie isyana bagoes oka unjung tokoh agama islam jayapura thaha alhamid kata datang rangkul semua golong tanah papua semua kan rangkul anti intoleransi kata kaesang rumah abah thaha jayapura papua senin kaesang jelas sosok abah thaha rupa juru bicara pasang capres cawapres nomor **orang orang orang organisasi**

Adapun proses yang dilakukan dalam *debiasing* manual dari berita tersebut adalah yang pertama dilakukan identifikasi bias sesuai pada Tabel 6. Pada contoh diatas kata yang bercetak tebal merupakan kata yang bias, dapat dilihat pada contoh berita pertama terdapat beberapa kata yang bias seperti nama orang pada kata ‘anies’, ‘anies baswedan’ dan ‘amin’. Kemudian pada berita kedua dapat diidentifikasi beberapa kata yang bias seperti kata ‘psi’, ‘anies baswedan’, ‘muhaimin iskandar’, ‘cak imin’ dan ‘amin’. Setelah dilakukan *debiasing* secara manual ini maka berita yang sebelumnya bias sudah menjadi netral.

Adapun *debiasing* secara otomatis, alur kerjanya dapat dilihat pada Gambar 30 dibawah ini



Gambar 30 Cara Kerja *Debiasing* Otomatis

Sesuai dengan Gambar 30 diatas langkah pertama pada *debiasing* otomatis adalah berita terkait politik dilakukan identifikasi bias sesuai dengan kategori bias pada Tabel 6. Setelah bias diidentifikasi dilakukan pelabelan kata yang bias dengan bantuan pelabelan *online,* pelabelan kata tersebut dilakukan pada kata yang termasuk dalam kategori bias nya, adapun contoh kategori pelabelan kata dapat dilihat pada Tabel 8 dan contoh pelabelan dapat dilihat pata Gambar 24. Setelah dilakukan labeling kemudian diimplementasikan ke model NER untuk melabeling semua data dan langkah terakhir adalah dilakuakn *debiasing* berdasarkan kata yang sudah dilakukan labeling dengan NER tersebut. Penggunaan NER dalam pelabelan sebelum *debiasing* otomatis dapat membantu memberikan struktur dan kejelasan dalam memahami konteks di mana entitas muncul pada teks serta NER dapat meningkatkan akurasi identifikasi entitas yang relevan, sehingga memungkinkan proses debiasing yang lebih terfokus dan efektif. Hasil berita yang sudah di*debiasing* maka berita sudah netral. Adapun hasil dari contoh 2 berita yang di*debiasing* secara manual dapat dilihat sebagai berikut

Tabel 16 Hasil *Debiasing* Manual

Berita	Hasil *debiasing* manual
calon presiden nomor urut **anies baswedan**	calon presiden nomor urut **orang** mulai

mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang **amin** timnas **amin** m syaugi jelas tanah merah rupa tempat sejarah **anies** betul tanah merah rupa sejarah beliau mudah mudah jadi tanda baik zaman gubernur masa lalu	kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang **organisasi** timnas **organiasi** m syaugi jelas tanah merah rupa tempat sejarah **orang** betul tanah merah rupa sejarah beliau mudah mudah jadi tanda baik zaman gubernur masa lalu
ketua umum **psi** kaesang pangarep sama dewan bina **psi** grace natalie isyana bagoes oka unjung tokoh agama islam jayapura thaha alhamid kata datang rangkul semua golong tanah papua semua kan rangkul anti intoleransi kata kaesang rumah abah thaha jayapura papua senin kaesang jelas sosok abah thaha rupa juru bicara pasang capres cawapres nomor **anies baswedan muhaimin iskandar cak imin amin**	ketua umum **partai politik** kaesang pangarep sama dewan bina **partai politik** grace natalie isyana bagoes oka unjung tokoh agama islam jayapura thaha alhamid kata datang rangkul semua golong tanah papua semua kan rangkul anti intoleransi kata kaesang rumah abah thaha jayapura papua senin kaesang jelas sosok abah thaha rupa juru bicara pasang capres cawapres nomor **orang orang orang organisasi**

Hasil dari *debiasing* secara manual dan otomatis dengan model selanjutnya akan dilakukan perbandingan dari berapa bias yang berhasil dilakukan secara keseluruhan pada 20 berita tersebut, hal ini dilakukan guna untuk mengevaluasi efektivitas dari kedua metode dalam mengurangi atau menghilangkan bias, mengidentifikasi kelebihan dan kekurangan masing-masing metode tersebut. Adapun perbandingan hasil dari kedua metode *debiasing* tersebut dapat dilihat pada Tabel 17 dibawah ini.

Tabel 17 Perbandingan Hasil Debiasing Manual dan Debiasing Otomatis

Berita ke-	Isi Berita	Jumlah Kata Bias		
		Sebelum Debiasing	**Setelah Debiasing Otomatis**	**Setelah Debiasing Manual**
Berita ke-1	calon presiden nomor urut anies baswedan ….	25 bias	0 bias	0 bias
Berita ke-2	ketua umum psi kaesang pangarep…	23 bias	0 bias	0 bias
Berita ke-3	calon presiden capres nomor urut anies baswedan…	15 bias	0 bias	0 bias
Berita ke-4	pks sulawesi selatan sulsel tegas tim kampanye…	23 bias	0 bias	0 bias
Berita ke-5	ketua dewan timbang partai nasdem jawa barat…	21 bias	1 bias	0 bias
Berita ke-6	mampu sang kapten orkestrasi tim menang…	43 bias	3 bias	0 bias
Berita ke-7	pasang capres cawapres anies baswedan muhaimin iskandar..	21 bias	1 bias	0 bias
Berita ke-8	bakal pasang calon koalisi indonesia maju kim prabowo..	43 bias	0 bias	1 bias
Berita ke-9	bakal capres koalisi ubah anies baswedan …	36 bias	2 bias	0 bias
Berita ke-10	pasang prabowo subianto gibran rakabuming..	78 bias	1 bias	1 bias
Berita ke-11	bacapres koalisi indonesia maju kim prabowo..	47 bias	1 bias	1 bias
Berita ke-12	bacapres koalisi indonesia maju kim prabowo subianto ….	35 bias	3 bias	0 bias
Berita ke-13	koalisi indonesia maju kim resmi dukung prabowo..	58 bias	0 bias	0 bias
Berita ke-14	bakal capres cawapres koalisi ubah anies baswedan muhaimin..	37 bias	0 bias	0 bias
Berita ke-15	bakal capres cawapres koalisi ubah anis baswedan muhaimin…	21 bias	0 bias	0 bias

Berita ke-16	partai gerindra unggah buah cuplik video menko polhukam…	43 bias	1 bias	0 bias
Berita ke-17	pasang anies baswedan muhaimin iskandar alias cak imin amin..	25 bias	0 bias	0 bias
Berita ke-18	bacawapres ubah tum pkb muhaimin iskandar cak imin…	34 bias	0 bias	0 bias
Berita ke-19	tum pkb muhaimin iskandar cak imin sempat kelakar jadi…	28 bias	0 bias	0 bias
Berita ke-20	ketua umum pkb muhaimin iskandar cak imin cerita…	27 bias	0 bias	0 bias
total		683	13 bias	3 bias

Adapun kata yang tidak berhasil ter*debiasing* pada proses otomatis adalah kata 'nasdem' pada berita 5, kata 'mahfud', 'muhaimin', 'demokrat' pada berita 6, kata 'imin' pada berita 7, kata 'rakabuming' dan 'raka' pada berita 9, kata 'mahfud', 'muhaimin', 'muhaimin iskandar' pada berita 12 dan kata 'pranowo pada berita 16' adapun untuk proses manual kata yang tidak berhasil ter*debiasing* adalah kata 'pan'. Pada Tabel 17 dapat dilihat bahwa masih ada 13 kata bias yang tidak berhasil ter*debiasing* dengan baik dengan metode otomatis dibandingan dengan metode manual hal ini dikarenakan beberapa hal seperti pada saat labeling terjadi kesalahan identifikasi entitas (kurang akurat) sehingga proses *debiasing* terdapat entitas yang terlewat. Dibandingkan dengan metode manual dapat mengidentikasi bias dengan lebih akurat, namun dalam penggunaan data yang besar penggunaan *debiasing* otomatis dapat membantu untuk mengefisiensi waktu dan tenaga dibandingkan dengan metode manual. Agar mendapatkan hasil lebih baik dapat menggunakan kedua metode tersebut untuk mendapatkan dan memastikan hasil yang lebih akurat dan lebih efisien dalam proses debiasing.

Pada penelitian yang dilakukan debiasing berita tidak memperhatikan konteks isi berita sehingga informasi yang dihasilkan kurang relevan. Adapun contoh potongan berita yang diakukan proses debiasing dengan memerhatikan nilai dari isi beritanya sehingga tidak kehilangan makna berita tersebut. Adapun preses debiasing sebelumnya nama terkait kandidat presiden dan calon presiden diganti dengan ‘orang’ dan nama partai politik diganti dengan ‘partai politik’ dan organisasi tim kemenangan capres-cawapres diganti dengan ‘organisasi’. Namun, penggantian nama kandidat presiden dan calon wakil presiden dengan ‘orang’ terlalu umum dan mengakibatkan informasi yang relevan menjadi kurang jelas. Oleh karena itu, istilah ‘kandidat calon presiden’ untuk calon presiden dan ‘kandidat calon wakil presiden’ untuk calon wakil presiden. Istilah ini dapat memberikan informasi mengenai posisi seseorang tanpa menyebutkan identitas spesifik seperti nama untuk mengurangi bias. Namun, tetap mempertahankan konteks dan makna berita secara umum. potongan berita tersebut dapat dilihat pada Tabel 18

Tabel 18 Pembaruan Contoh Berita Debiasing

Sebelum debiasing	Setelah debiasing
calon presiden nomor urut **anies baswedan** mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kapten tim menang **amin** timnas **amin** m syaugi jelas tanah merah rupa tempat sejarah **anies** betul tanah merah rupa sejarah beliau mudah mudah jadi tanda baik zaman gubernur masa lalu	calon presiden nomor urut **calon presiden** mulai kampanye hari pertama unjung kawasan tanah merah jakarta utara kaptem tim menang **organisasi** timnas **organisasi** m syaugi jelas tanah merah rupa tempat sejarah **calon presiden** betul tanah merah rupa sejarah beliau mudah mudah menjadi tanda baik zaman gubernur masa lalu

4.6 Evaluasi Hasil *Debiasing*

Langkah terkahir adalah melakukan evaluasi dengan membandingkan sebelum dan sesudah *debiasing* menggunakan metode sentimen analisis berbasis *lexicon-based* menggunakan kamus bahasa Indonesia (Barasa-ID) pada data teks berita terkait politik yang diteliti. Tujuan pengukuran ini adalah untuk mengukur

efektivitas proses *debiasing* dalam mengurangi bias yang ada pada data, selain itu evaluasi ini juga dilakukan untuk memastikan model untuk memberikan penilaian yang lebih adil dan seimbang. Berikut merupakan potongan Kode 19 untuk memuat leksikon Barasa-ID dan Kode 20 dengan fungsi 'defget_sentimen_scores untuk mendapatkan skor sentimen dan hasil dari sentimen tersebut dilakukan dengan fungsi 'resul'. Berikut merupakan contoh potongan kode tersebut yang digunakan pada saat melakukan evaluasi sebelum dilakukan proses *debiasing*:

Kode 19 Memuat Leksikon Barasa-ID

```
# Memuat leksikon Barasa-ID
barasa_lexicon = load_barasa_lexicon('/content/barasa-ID.txt')
```

Kode 20 Sentimen Analisis Sebelum Proses *Debiasing*

```
# Fungsi untuk mendapatkan skor sentimen
def get_sentiment_scores(text):
    return sid.polarity_scores(text)

# Menganalisis sentimen setiap teks dalam kolom prep_text
results = []
for text in df['prep_text']:
    sentiment_scores = get_sentiment_scores(text)
    results.append({
        'text': text.strip(),
        'neg': sentiment_scores['neg'],
        'neu': sentiment_scores['neu'],
        'pos': sentiment_scores['pos'],
        'compound': sentiment_scores['compound']
    })
```

Berikut merupakan hasil sentimen analisis sebelum dilakukan proses *debiasing* dapat dilihat pada Gambar 31 berikut:

	text	neg	neu	pos	compound
0	calon presiden nomor urut anies baswedan mulai...	0.000	0.751	0.249	0.9791
1	ketua umum psi kaesang pangarep sama dewan bin...	0.000	0.882	0.118	0.9684
2	calon presiden capres nomor urut anies basweda...	0.000	0.793	0.207	0.9598
3	pks sulawesi selatan sulsel tegas tim kampanye...	0.000	0.901	0.099	0.8472
4	ketua dewan timbang partai nasdem jawa barat j...	0.000	0.966	0.034	0.1593
...	...	...	...	...	...
5212	calon presiden capres prabowo subianto beri pu...	0.000	0.821	0.179	0.7543
5213	wali kota medan bobby nasution resmi dukung pa...	0.000	0.872	0.128	0.7073
5214	baliho resmi bakal capres cawapres koalisi ind...	0.000	0.894	0.106	0.8176
5215	mahkamah konstitusi mk gelar sidang atas gugat...	0.007	0.839	0.154	0.9250
5216	calon presiden capres ganjar pranowo tegas kom...	0.000	0.889	0.111	0.0323

5217 rows × 5 columns

Gambar 31 Sentimen Analisis sebelum *Debiasing*

Pada Gambar 31 diatas terdapat beberapa keterangan sebagai berikut:

- Text : text yang di analisis
- Neg : skor sentimen negatif dengan nilai beriksar antara 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen negatif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen negatif.
- Neu : skor sentimen netral dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen netral dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen netral.
- Pos : skor sentimen positif dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen positif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen positif.

Adapun penjelasan dari Gambar 31 diatas pada baris 0 dengan teks “calom presiden nomor urut anies baswedan mulai…” mendapatkan skor sentimen negatif sebesar 0.000 yang menunjukkan bahwa teks tersebut tidak memiliki sentimen negatif, kemudian menujukkan skor sentimen netral sebesar 0.751 yang menunjukkan bahwa sebagian besar teks memiliki sentimen netral dan

mendapatkan skor 0.249 yang menujukkan bahwa teks tersebut sebagian kecil ada sentimen positif.

Pada baris 1 dengan teks “ketua umum psi kaesang pangarep sama dewan bin…” mendapatkan skor negatif sebesar 0.000 yang menujukkan teks tersebut tidak bersentimen negatif, kemudian untuk sentimen netral mendapat skor sebesar 0.882 yang menunjukkan sebagian besar teks memiliki sentimen netral, pada sentimen positif bernilai 0.118 yang menunjukkan sebagian kecil teks memiliki sentimen positif.

Setelah proses *debiasing* dilakukan kembali sentimen analisis untuk mengukur dan mengevaluasi perubahan dalam sentimen teks yang dianalisis untuk memastikan bahwa teks yang dianalisis memberikan gambaran yang lebih seimbang dan adil. Proses ini menggunakan metode yang sama, yaitu sentimen analisis dengan metode *lexicon-based* berdasarkan kamus bahasa Indonesia (Barasa-ID). Adapun potongan Kode 21 yang dilakukan pada proses setelah *debiasing* sebagai berikut

Kode 21 Sentimen Analisis Setelah Proses *Debiasing*

```
# Inisialisasi SentimentIntensityAnalyzer dan tambahkan leksikon Barasa-ID
sid = SentimentIntensityAnalyzer()

# Memuat leksikon Barasa-ID
barasa_lexicon = load_barasa_lexicon('/content/barasa-ID.txt')

# Menambahkan leksikon Barasa-ID
sid.lexicon.update(barasa_lexicon)

# Membaca dataset dari file debiased_text.txt
with open('/content/debiased_text2.txt', 'r') as file:
    texts = file.readlines()

# Fungsi untuk mendapatkan skor sentimen
def get_sentiment_scores(text):
    return sid.polarity_scores(text)
```

```
# Menganalisis sentimen setiap teks dalam dataset
results = []
for text in texts:
    sentiment_scores = get_sentiment_scores(text)
    results.append({
        'text': text.strip(),
        'neg': sentiment_scores['neg'],
        'neu': sentiment_scores['neu'],
        'pos': sentiment_scores['pos'],
        'compound': sentiment_scores['compound']
    })
```

Setelah proses *debiasing* data yang digunakan bernama 'debiased_text2' kemudian dilakukan penghitungan sentimen negatif, positi dan netral dengan 'get_sentimen_scores' untuk setiap teks berdasarkan dengan daftar kamus barasa_ID.

Hasil pengecekan evaluasi *debiasing* manual dan otomatis pada 20 berita dapat dilihat pada Gambar 32 berikut

	text	neg	neu	pos	compound
0	calon presiden nomor urut orang mulai kampanye...	0.00	0.743	0.257	0.9791
1	ketua umum psi kaesang pangarep sama dewan bin...	0.00	0.880	0.120	0.9684
2	calon presiden capres nomor urut orang janji b...	0.00	0.793	0.207	0.9598
3	partai politik sulawesi selatan sulsel tegas t...	0.00	0.904	0.096	0.8422
4	ketua dewan timbang partai partai politik jawa...	0.00	0.965	0.035	0.1593
5	mampu sang kapten orkestrasi tim menang salah ...	0.01	0.891	0.099	0.9636
6	pasang capres cawapres orang orang orang hadir...	0.00	0.720	0.280	0.9100
7	bakal pasang calon koalisi indonesia maju kim ...	0.00	0.870	0.130	0.9052
8	bakal capres koalisi ubah orang respons pasang...	0.00	0.889	0.111	0.8674
9	pasang orang orang orang rencana daftar pasang...	0.00	0.895	0.105	0.8945
10	bacapres koalisi indonesia maju kim orang umum...	0.00	0.887	0.113	0.8591
11	bacapres koalisi indonesia maju kim orang umum...	0.00	0.851	0.149	0.6280
12	koalisi indonesia maju kim resmi dukung orang ...	0.00	0.836	0.164	0.9453
13	bakal capres cawapres koalisi ubah orang orang...	0.00	0.877	0.123	0.7488
14	bakal capres cawapres koalisi ubah orang orang...	0.00	0.881	0.119	0.7073
15	partai partai politik unggah buah cuplik video...	0.00	0.892	0.108	0.7830
16	pasang orang orang alias orang organisasi resm...	0.00	0.847	0.153	0.8866
17	bacawapres ubah tum partai politik orang orang...	0.00	0.851	0.149	0.7830
18	tum partai politik orang orang sempat kelakar ...	0.00	0.860	0.140	0.6956
19	ketua umum partai politik orang orang cerita d...	0.00	0.825	0.175	0.8750

Gambar 32 Sentimen Analisi *Debiasing* Manual

Adapun hasil *debiasing* sacara manual dapat dilihat pada Gambar 32 diatas dengan keterangan sebagai berikut:

- Text : text yang di analisis
- Neg : skor sentimen negatif dengan nilai beriksar antara 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen negatif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen negatif.
- Neu : skor sentimen netral dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen netral dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen netral.
- Pos : skor sentimen positif dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen positif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen positif.

Pada Gambar 32 diatas dapat dilihat bahwa dari ke 2 berita yang dianalisis tidak ada berita yang mengandung sentimen negatif dan dari 20 berita diatas mengandung sentimen netral dengan sebagian besar skornya mendekati 1 (diatas 0.5), yang menandakan bahwa sentimen pada berita diatas memiliki sentimen netral dan positif dengan skor rata-rata mendekati 1 (dibawah 0.5) yang menandakan sebagian kecil berita mengandung sentimen negatif. Adapun hasil evaluasi *debiasing* dengan 20 berita secara otomatis dapat dilihat sebagai berikut.

	text	neg	neu	pos	compound
0	calon presiden nomor urut orang mulai kampanye...	0.00	0.748	0.252	0.9791
1	ketua umum psi kaesang pangarep sama dewan bin...	0.00	0.880	0.120	0.9684
2	calon presiden capres nomor urut orang janji b...	0.00	0.793	0.207	0.9598
3	partai politik sulawesi selatan sulsel tegas t...	0.00	0.902	0.098	0.8472
4	ketua dewan timbang partai nasdem jawa barat j...	0.00	0.965	0.035	0.1593
5	mampu sang kapten orkestrasi tim menang salah ...	0.01	0.889	0.101	0.9648
6	pasang capres cawapres orang orang orang hadir...	0.00	0.720	0.280	0.9100
7	bakal pasang calon koalisi indonesia maju kim ...	0.00	0.874	0.126	0.9052
8	bakal capres koalisi ubah orang respons pasang...	0.00	0.885	0.115	0.8750
9	pasang orang orang rencana daftar pasang capre...	0.00	0.894	0.106	0.9001
10	bacapres koalisi indonesia maju kim orang umum...	0.00	0.888	0.112	0.8591
11	bacapres koalisi indonesia maju kim orang umum...	0.00	0.851	0.149	0.6280
12	koalisi indonesia maju kim resmi dukung orang ...	0.00	0.837	0.163	0.9453
13	bakal capres cawapres koalisi ubah orang orang...	0.00	0.880	0.120	0.7488
14	bakal capres cawapres koalisi ubah orang orang...	0.00	0.884	0.116	0.7073
15	partai politik unggah buah cuplik video menko ...	0.00	0.894	0.106	0.7830
16	pasang orang orang alias orang organisasi resm...	0.00	0.847	0.153	0.8866
17	bacawapres ubah tum partai politik orang orang...	0.00	0.856	0.144	0.7830
18	tum partai politik orang orang sempat kelakar ...	0.00	0.859	0.141	0.6956
19	ketua umum partai politik orang orang cerita d...	0.00	0.827	0.173	0.8750

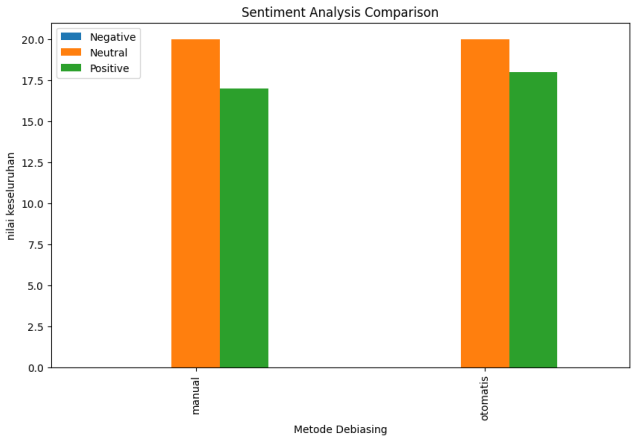
Gambar 33 Sentimen Metode Otomatis

Pada Gambar 33 diatas terdapat beberapa keterangan sebagai berikut:

- Text : text yang di analisis

- Neg : skor sentimen negatif dengan nilai beriksar antara 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen negatif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen negatif.
- Neu : skor sentimen netral dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen netral dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen netral.
- Pos : skor sentimen positif dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen positif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen positif.

Dapat dilihat pada Gambar 33 diatas dapat dilihat bawah sebagian berita memiliki nilai sentimen mendekati 1 (diatas 0.5) pada sentimen netral, yang menandakan bahwa sebagian berita diatas mengandung sentimen netral dan pada sentimen positif mendekati 1 (dibawah 0.5) yang menandakan sebagian kecil berita diatas mengandung sentimen negatif dan tidak ada sentimen netral pada kedua berita diatas. Kemudian dilakukan perhitungan skor sentimen secara keseluruhan baik negatif, positif dan netral. Didapatkan hasil sebagai berikut



Gambar 34 Perbandingan Hasil Sentimen Manual dan Otomatis

Pada Gambar 34 diatas dapat dilihat bahwa dari kedua metode diatas tidak ada sentimen negatif dengan metode manual, untuk sentimen netral didapatkan skor sebanyak 20 pada masing-masing metode dan sentimen positif sebanyak 17 untuk metode manual dan 18 untuk metode otomatis dan tidak didapatkan sentimen negatif dari kedua metode diatas. Dari hasil perbandingan ini dapat disimpulkan

bahwa metode manual lebih efektif untuk mengurangi kecenderungan sentimen pada berita politik diatas.

Adapun hasil analisis sentimen pada seluruh teks berita dapat dilihat pada Gambar 35 dibawah ini.

	text	neg	neu	pos	compound
0	calon presiden nomor urut orang mulai kampanye...	0.000	0.748	0.252	0.9791
1	ketua umum psi kaesang pangarep sama dewan bin...	0.000	0.880	0.120	0.9684
2	calon presiden capres nomor urut orang janji b...	0.000	0.793	0.207	0.9598
3	partai politik sulawesi selatan sulsel tegas t...	0.000	0.902	0.098	0.8472
4	ketua dewan timbang partai politik jawa barat ...	0.000	0.965	0.035	0.1593
...	...	...	...	...	...
5207	calon presiden capres orang beri puji presiden...	0.000	0.819	0.181	0.7543
5208	wali kota medan bobby nasution resmi dukung pa...	0.000	0.884	0.116	0.6833
5209	baliho resmi bakal capres cawapres koalisi ind...	0.000	0.892	0.108	0.8176
5210	mahkamah konstitusi mk gelar sidang atas gugat...	0.007	0.839	0.154	0.9250
5211	calon presiden capres orang komitmen daulat ma...	0.000	0.862	0.138	0.0323

5212 rows × 5 columns

Gambar 35 Sentimen Analisis Setelah *Debiasing*

Pada Gambar 35 diatas terdapat beberapa keterangan sebagai berikut:

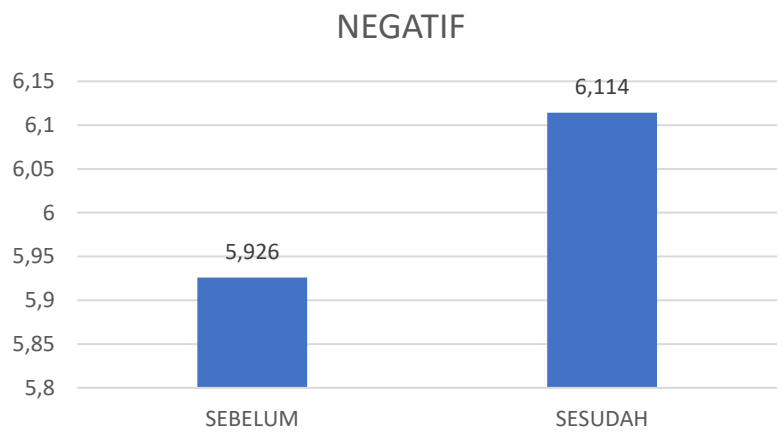
- Text : text yang di analisis
- Neg : skor sentimen negatif dengan nilai beriksar antara 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen negatif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen negatif.
- Neu : skor sentimen netral dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen netral dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen netral.
- Pos : skor sentimen positif dengan nilai berkisar 0 hingga 1. Nilai 0 menunjukkan bahwa tidak ada sentimen positif dalam teks dan nilai 1 menunjukkan bahwa seluruh teks bersentimen positif.

Dari Gambar 35 diatas dapat dilihat pada baris 0 pada teks “calon presiden nomor urut orang mulai kampanye…” mendapat skor negatif sentimen sebesar 0.000 yang menunjukkan bahwa tidak ada sentimen negatif pada teks tersebut, kemudian mendapat skor sentimen netral sebesar 0.748 yang menunjukkan sebagian bear teks memiliki nilai netral yang tinggi dan pada sentimen positif

mendapat skor 0.9791 yang menunjukkan tidak ada sentimen positif pada teks tersebut.

Pada baris 1 dengan teks "ketua umum partai politik orang sama dewan bin…" mendapatkan skor negatif sebesar 0.000 yang menunjukkan teks tersebut tidak memiliki sentimen negatif, kemudian mendapat skor netral 0.880 menunjukkan bahwa teks tersebut memilik sentimen netral yang cukup tinggi, skor sentimen positif mendapat skor 0.120 yang menunjukkan sentimen positif yang kecil pada teks tersebut.

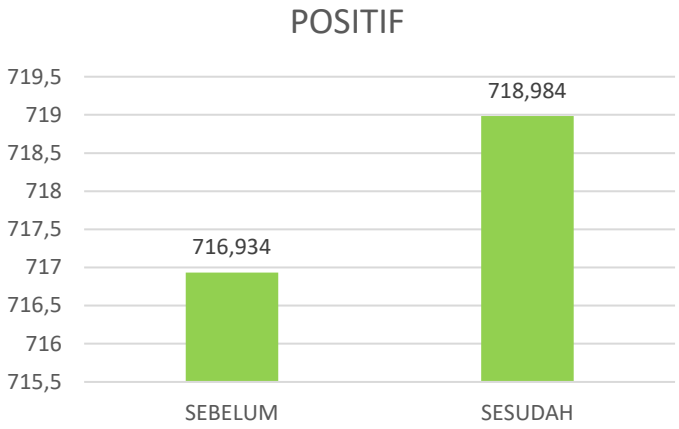
Adapun analisis sentimen negatif, positif dan netral pada keseluruhan data sebelum *debiasing* dan setelah proses *debiasing* dapat dilihat pada dibawah ini. Berikut merupakan analisis sentimen negatif secara keseluruhan berita baik sebelum dan sesudah dilakukan sebagai berikut.



Gambar 36 Perbandingan Sentimen Negatif Sebelum dan Sesudah

Pada Gambar 36 diatas merupakan total keseluruhan dari sentimen negatif yang terdapat dalam semua teks pada data pada sentimen sebelum dan sesudah dilakukan proses *debiasing*. Perhitungan *fairness measuremenet* yang dilakukan adalah dengan dilakukan perbandingan total nilai sentimen negatif pada semua dataset. Sebelum dilakukan proses *debiasing* jumlah skor sentimen negatif pada seluruh data sebesar 5,926 dan setelah dilakukan *debiasing* dilakukan jumlah skor sentimen negatif naik menjadi 6,114.

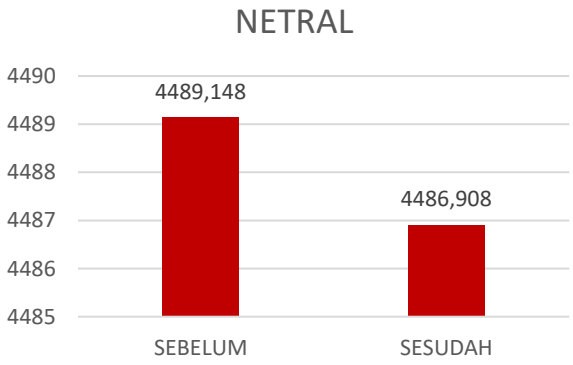
Adapun jumlah keseluruhan sentimen positif sebelum dilakukan proses *debiasing* dan setelah proses *debiasing* dapat dilihat pada Gambar 37



Gambar 37 Perbandingan Sentimen Positif Sebelum dan Sesudah

Adapun hasil yang didapat berdasarkan Gambar 37 diatas menunjukkan bahwa terjadi kenaikan nilai sentimen sentimen positif. Dimana, sebelum *debiasing* jumlah skor sentimen positif sebesar 716,934 dan sesudah dilakukan *debiasing* jumlah skor sentimen positif menjadi 718,984.

Adapun jumlah keseluruhan sentimen netral sebelum dilakukan proses *debiasing* dan sesudah dilakukan proses *debiasing* dapat dilihat pada Gambar 38 berikut



Gambar 38 Perbandingan Sentimen Netral Sebelum dan Sesudah

Pada Gambar 38 diatas dapat dilihat bahwa terjadi penuruan sentimen netral sebelum dilakukan *debiasing* dan sesudah dilakukan *debiasing*. Dimana, sebelum dilakukan *debiasing* total skor sentimen netral sebanyak 4489,148 dan sesudah dilakukan *debiasing* total skor menurun menjadi 4486,908. Berdasarkan hasil dari keseluruhan sentimen negatif, positif dan netral diatas dapat disimpulkan bahwa terdapat peningkatan sentimen positif dan negatif, sedangkan sentimen netralnya

mengalami penurunan. Hal ini dapat terjadi karena beberapa hal, seperti teknik *debiasing* yang dilakukan mungkin tidak cukup efektif dalam menghapus atau mengurangi bias yang ada dalam data dan metode berbasis leksikon yang digunakan memiliki keterbatasan dalam menangkap jenis bias yang ada pada daftar kata yang disediakan sehingga bias mungkin tidak terdeteksi dengan baik, sehingga metode *lexicon-based* yang digunakan tidak cocok untuk mengukur evaluasi bias pada penelitian yang dilakukan ini.

BAB 5
PENUTUP

Pada bab ini berisikan kesimpulan yang didapatkan dari penelitian yang telah dilakukan dan saran bagi pengembangan penelitian selanjutnya yang dapat diuraikan sebagai berikut:

5.1 Kesimpulan

Berdasarkan dari investigasi yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Dalam merepresentasikan kata, kedua IndoBERT dan Word2Vec cukup baik, dapat dilihat pada visulisasi dengan PCA dan t-SNE dari kedua model tersebut dapat terlihat kmpulan kata yang merepresentasikan bias dalam data. Setelah proses *debiasing* baik manual dan otomatis, kumpulan kata yang merepresentasikan bias sudah tidak terlihat baik dari represetasi *word embedding* dengan perhitungan kemunculan bias dalam berita dari sebelum *debiasing* dan setelah *debiasing* terdapat pengurangan kata yang bias, sehingga dapat disimpulkan bahwa proses *debiasing* yang dilakukan diasumsikan berhasil untuk menghilangkan bias.
2. Pengukuran evaluasi hasil *debiasing* pada berita terkait pemilu presiden di Indonesia menggunakan sentimen analisis dengan metode *lexicon-based* dari hasil *debiasing* yang dilakukan terjadi kenaikan sentimen negatif sebesar 0,18 dan positif sebesar 2,05 serta terjadi penurunan nilai sentimen netral sebesar 2,24 setelah dilakukan proses *debiasing,* dari hasil tersebut disimpulkan bahwa terdapat kecenderungan dalam hasil analisis sehingga metode ini dianggap kurang efektif dalam mengevaluasi hasil *debiasing*.

5.2 Saran

Berdasarkan kesimpulan yang diperoleh dari hasil penelitian ini, terdapat beberapa saran dari penulis yang dapat dijadikan sebagai bahan pertimbangan dan pengembangan untuk melakukan penelitian lebih lanjut. Adapaun beberapa saran yang penulis rangkum sebagai berikut:

1. Pada penelitian selanjutnya dapat menggunakan metode *word embedding* yang lain seperti BERT, Glove atau lainnya agar dapat memberikan gambaran yang beragam dalam merepresentasikan kata.
2. Penelitian selanjutnya disarankan untuk memperbanyak jumlah dataset yang dilakukan serta memastikan ketepatan pelabelan agar variasi kata yang dilabeli beragam dan tepat, guna untuk meningkatkan keakuratan pengenalan dan pelabelan pada entitas. Sehingga model NER dapat meningkatkan performa dalam mengidentifikasi atau mendeteksi entitas secara menyeluruh dan akurat.
3. Penelitian selanjutnya dapat menggunakan metode pengukuran yang lain dalam mengukur evaluasi *debiasing* seperti *statisical methods*, *contextual analysis methods* atau yang lain agar dapat memberikan hasil yang lebih akurat, maksimal dan mendalam. Selain itu, penting untuk memperhatikan konteks dari berita apa yang dibicarakan guna untuk mendeteksi agar bias yang di deteksi lebih akurat dalam pengukuran evauasi *debiasing*.

DAFTAR PUSTAKA

A. Yani, D. D., Pratiwi, H. S. and Muhardi, H. (2019) 'Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace', *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 7(4), p. 257. doi: 10.26418/justin.v7i4.30930.

Adinugroho, B. *et al.* (2019) 'MEDIA SOSIAL DAN INTERNET DALAM KETELIBATAN INFORMASI POLITIK DAN PEMILIHAN UMUM', *representamen*, 5(02). doi: 10.30996/representamen.v5i02.2943.

Arini, D. (2020) 'Penyuluhan Dampak Positif dan Negatif Media Sosial Terhadap Kalangan Remaja Di Desa Way Heling Kecamatan Lengkiti Kabupaten Ogan Komering Ulu', *Abdimas Universal*, 2(1), pp. 49–53. doi: 10.36277/abdimasuniversal.v2i1.38.

Bessa, A. (2023) *Lexicon-based sentiment analysis: What it is & how to conduct one*.

Chen, W.-F. *et al.* (2020) 'Analyzing Political Bias and Unfairness in News Articles at Different Levels of Granularity'.

Davis, S. R., Worsnop, C. J. and Hand, E. M. (2022) 'Gender bias recognition in political news articles', *Machine Learning with Applications*. Elsevier BV, 8, p. 100304. doi: 10.1016/j.mlwa.2022.100304.

Devlin, J. *et al.* (2018) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'.

Eriyanto (2011) *ANALISIS FRAMING Konstruksi, Ideologi, dan Politik Media*. VI. Edited by N. Huda SA. Yogyakarta: LKiS Printing Cemerlang.

Farid, A. S. (2023) 'Peran Media Massa Dalam Memoderasi Dialog Politik', 1(3), pp. 151–161. doi: 10.59246/aladalah.v1i3.343.

Flores, V. A., Permatasari, P. A. and Jasa, L. (2020) 'Penerapan Web Scraping Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan Keyword', *Majalah Ilmiah Teknologi Elektro*, 19(2), p. 157. doi: 10.24843/MITE.2020.v19i02.P06.

Gordon, J., Babaeianjelodar, M. and Matthews, J. (2020) 'Studying Political Bias via Word Embeddings', in *The Web Conference 2020 - Companion of the World Wide Web Conference, WWW 2020*. Association for Computing Machinery, pp. 760–764. doi: 10.1145/3366424.3383560.

Indonesiabaik.id (2023) *Media Digital Semakin Mendominasi*, *Indonesiabaik.id*.

Intyaswati, D. (2021) 'Peran Media Massa Terhadap Partisipasi Politik Mahasiswa Pada Pemilihan Umum 2019', *Jurnal Penelitian Pers dan Komunikasi Pembangunan*, 25(2). doi: 10.46426/jp2kp.v25i2.142.

Jatnika, D., Bijaksana, M. A. and Suryani, A. A. (2019) 'Word2Vec Model Analysis for Semantic Similarities in English Words', *Procedia Computer Science*, 157, pp. 160–167. doi: 10.1016/j.procs.2019.08.153.

Khatimah, H. (2018) ‘POSISI DAN PERAN MEDIA DALAM KEHIDUPAN MASYARAKAT’, *TASAMUH*, 16(1), pp. 119–138. doi: 10.20414/tasamuh.v16i1.548.

Kiritchenko, S. and Mohammad, S. M. (2018) *Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems*. Available at: https://competitions.codalab.org/competitions/17751.

Kleinnijenhuis, J., van Hoof, A. M. J. and van Atteveldt, W. (2019) ‘The Combined Effects of Mass Media and Social Media on Political Perceptions and Preferences’, *Journal of Communication*, 69(6), pp. 650–673. doi: 10.1093/joc/jqz038.

Komisi Pemilihan Umum Republik Indonesia (2023) *KPU Tetapkan Tiga Pasangan Calon Presiden dan Wakil Presiden Pemilu 2024*, *Komisi Pemilihan Umum Republik Indonesia*.

Liang, P. P. *et al.* (2020a) ‘Towards Debiasing Sentence Representations’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 5502–5515. doi: 10.18653/v1/2020.acl-main.488.

Liang, P. P. *et al.* (2020b) ‘Towards Debiasing Sentence Representations’.

Liu, R. *et al.* (2021) *Political Depolarization of News Articles Using Attribute-aware Word Embeddings*. Available at: www.aaai.org.

May, C. *et al.* (2019) *On Measuring Social Biases in Sentence Encoders*. Available at: http://github.com/W4ngatang/sent-bias.

Mikolov, T., Yih, W.-T. and Zweig, G. (2013) *Linguistic Regularities in Continuous Space Word Representations*. Association for Computational Linguistics. Available at: http://research.microsoft.com/en-.

Muliawanti, L. (2018) ‘JURNALISME ERA DIGITAL: DIGITALISASI JURNALISME DAN PROFESIONALITAS JURNALISME ONLINE’, *LENTERA: Jurnal Ilmu Dakwah dan Komunikasi*, 2(1). doi: 10.21093/lentera.v2i1.1168.

Najib, A. C. *et al.* (2019) ‘Perbandingan Metode Lexicon-based dan SVM untuk Analisis Sentimen Berbasis Ontologi pada Kampanye Pilpres Indonesia Tahun 2019 di Twitter’, *Fountain of Informatics Journal*, 4(2), p. 41. doi: 10.21111/fij.v4i2.3573.

Nangia, N. *et al.* (2020) ‘CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models’.

Nugraha, A. and Mulyandari, A. (2016) ‘Pilkada Langsung Dan Pilkada Tidak Langsung Dalam Perspektif Fikih Siyasah’, *Mazahib*, 15(2). doi: 10.21093/mj.v15i2.630.

Nurdin, A. *et al.* (2020) ‘PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS’, *Jurnal*

TEKNOKOMPAK, 14(2), p. 74.

Petreski, D. and Hashim, I. C. (2023) 'Word embeddings are biased. But whose bias are they reflecting?', *AI & SOCIETY*, 38(2), pp. 975–982. doi: 10.1007/s00146-022-01443-w.

Putra, H. K., Arif Bijaksana, M. and Romadhony, A. (2021) 'Deteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia Menggunakan Metode IndoBERT', *Jurnal Tugas Akhir Fakultas Informatika* , 8(2).

Rahmatulloh, A. and Gunawan, R. (2020) 'Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar', *Indonesian Journal of Information Systems*, 2(2), pp. 95–104. doi: 10.24002/ijis.v2i2.3029.

Romanyshyn, N., Chaplynskyi, D. and Zakharov, K. (2023) 'Learning Word Embeddings for Ukrainian: A Comparative Study of FastText Hyperparameters', in *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 20–31. doi: 10.18653/v1/2023.unlp-1.3.

Solihah, R. (2018) 'Peluang dan tantangan pemilu serentak 2019 dalam perspektif politik', *Jurnal Ilmiah Ilmu Pemerintahan*, 3(1), p. 73. doi: 10.14710/jiip.v3i1.3234.

Spinde, T., Hamborg, F. and Gipp, B. (2020) 'Media Bias in German News Articles: A Combined Approach', in, pp. 581–590. doi: 10.1007/978-3-030-65965-3_41.

Styawati, S. *et al.* (2022) 'Sentiment Analysis on Online Transportation Reviews Using Word2Vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm', in *2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*. IEEE, pp. 163–167. doi: 10.1109/ISMODE53584.2022.9742906.

Subiyanto, A. E. (2020) 'Pemilihan Umum Serentak yang Berintegritas sebagai Pembaruan Demokrasi Indonesia', *Jurnal Konstitusi*, 17(2), p. 355. doi: 10.31078/jk1726.

Suleiman, D. and Awajan, A. (2018) 'Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications', in *2018 International Arab Conference on Information Technology (ACIT)*. IEEE, pp. 1–7. doi: 10.1109/ACIT.2018.8672674.

Suryo, H. and Aji, H. K. (2020) 'MEDIA SOSIAL DAN PESAN POLITIK (PERSEPSI PEMILIH PEMULA DALAM MENERIMA PESAN POLITIK PADA PEMILIHAN UMUM 2019 MELALUI MEDIA SOSIAL)', *RESEARCH FAIR UNISRI*, 4(1). doi: 10.33061/rsfu.v4i1.3390.

Swinger, N. *et al.* (2018) 'What are the biases in my word embedding?'

Syarifudin, F. (2019) 'Urgensi tabayyun dan kualitas informasi dalam

membangun komunikasi', *Al-Kuttab : Jurnal Kajian Perpustakaan, informasi dan kearsipan*, 1(2), pp. 29–39. doi: 10.24952/ktb.v1i2.1994.

Tan, Y. C. and Celis, L. E. (2019) *Assessing Social and Intersectional Biases in Contextualized Word Representations*. Available at: https://github.com/openai/gpt-2-output-dataset.

The MathWork, I. (2023) *Explore Fairness Metrics for Credit Scoring Model*, *The MathWork, Inc.*

Xiao, Z. *et al.* (2022) 'Detecting Political Biases of Named Entities and Hashtags on Twitter'.

Yu, L.-C. *et al.* (2018) 'Refining Word Embeddings Using Intensity Scores for Sentiment Analysis', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3), pp. 671–681. doi: 10.1109/TASLP.2017.2788182.

Yuniarti, L. and Rakhmawati, N. A. (2024) *Data Vektor Embeddings Menggunakan IndoBERT dan Word2Vec Terkait Pemilu Pemilu Presiden Indonesia*, *zenodo*.

Zhu, J.-J. and Ren, Z. J. (2023) 'The evolution of research in resources, conservation & recycling revealed by Word2vec-enhanced data mining', *Resources, Conservation and Recycling*, 190, p. 106876. doi: 10.1016/j.resconrec.2023.106876.

BIODATA PENULIS

Penulis dilahirkan di Bangka Tengah pada tanggal 08 Agustus 2000 sebagai anak bungsu dari lima bersaudara. Pendidikan formal penulis diawali di SD 9 Lubuk Besar, kemudian dilanjutkan di MTs. Al-Muhajirin Koba, dan MA Al-Muhajirin Koba. Setelah lulus dari MA Al-Muhajirin pada tahun 2019, penulis meraih Program Beasiswa Santri Berprestasi (PBSB) dan diterima di Departemen Sistem Informasi, FTEIC - ITS dengan NRP 05211940007001. Selama berkuliah di Departemen Sistem Informasi penulis aktif dalam berbagai kegiatan dan organisasi baik di dalam maupun di luar kampus seperti, UKM Rebana, CSSMoRA ITS, serta sebagai Pendamping Produk Halal (PPH). Skripsi penulis yang berjudul "Analisis Persebaran Data Privasi pada Mesin Pencari Berdasarkan UU PDP di Indonesia" merupakan hasil dari dedikasi dan kerja keras penulis. Pada saat masih menjalani pendidikan S1 tepatnya disemester 7, penulis mendapatkan kesempatan meraih beasiswa S2 *Fast Track* dari ITS di departemen yang sama, yaitu Sistem Informasi, FTEIC – ITS dengan NRP 6026231027. Penulis dapat dihubungi melalui email diyayulid@gmail.com. Semoga thesis ini dapat memberikan manfaat serta maslahat untuk kontribusi bagi penelitian selanjutnya.