

PEMODELAN TOPIK CHATGPT PADA ULASAN TWITTER MENGUNAKAN METODE *LATENT DIRICHLET ALLOCATION*

Ricky Putra Sardika¹, Cerwyn Asyraq², Muhammad Rizky Pribadi³, Wijang Widhiarso⁴

^{1,2,3,4}Program Studi Informatika Fakultas Ilmu Komputer Universitas Multi Data Palembang
rickyputrasardika_2226250129p@mhs.mdp.ac.id

Abstrak

Maraknya tren penggunaan *Artificial Intelligence* (AI) di era teknologi saat ini. Salah satu contohnya yakni *chatbot* yang bernama *chatgpt*. Telah banyak topik obrolan atau pertanyaan yang telah dilontarkan melalui *chatgpt* ini dari berbagai kalangan seperti orang biasa, pelajar, guru atau dosen, hingga tenaga peneliti. Hal tersebut dapat dideteksi untuk mengetahui tren topik hangat yang paling dominan dibicarakan di twitter. Namun dengan ratusan metadata tersebut akan sulit dicari dan dibaca. Oleh karena itu dibutuhkan metode *Latent Dirichlet Allocation* (LDA) untuk menyelesaikan masalah tersebut. LDA mampu melakukan analisis tentang tren topik hangat yang sedang dibahas di twitter mengenai *chatgpt*. Dari 202304 ulasan twitter mengenai *chatgpt*, didapat sejumlah 30 kata tren secara keseluruhan hasil dari *topic modelling* yang dilakukan dengan masing – masing 10 topik berdasarkan yang sudah dimodelkan.

Kata Kunci: *Artificial Intelligence*, *Chatgpt*, *Topic Modelling*, LDA, Twitter

Abstract

The trend of using Artificial Intelligence (AI) in the current technological era. One example is a chatbot called chatgpt. There have been many chat topics or questions that have been raised through this chatgpt from various groups such as ordinary people, students, teachers or lecturers, to research staff. This can be detected to find out the trend of hot topics that are most dominantly discussed on Twitter. However, with hundreds of metadata it will be difficult to find and read. Therefore a Latent Dirichlet Allocation (LDA) method is needed to solve this problem. LDA is able to analyze trending hot topics being discussed on Twitter regarding chatgpt. From 202304 Twitter reviews regarding chatgpt, obtained a total of 30 trend words as a result of the overall topic modeling carried out with each of the 10 topics based on what has been modelled.

Keywords: *Artificial Intelligence*, *Chatgpt*, *Topic Modelling*, LDA, Twitter

1. Pendahuluan

Artificial Intelligence atau biasa kita kenal dengan sebutan kecerdasan buatan ini telah melebur tidak dapat dipisahkan dari kehidupan manusia era modern. Sekarang beragam aplikasi berbasis kecerdasan buatan telah digunakan pada bidang manajemen, bisnis, kesehatan, dan terkhusus di sektor teknologi informasi (Raya & Tengah, 2023). Chatgpt salah satu contoh yang sedang tren saat ini. Chatgpt adalah sebuah *chatbot* AI yang diproduksi oleh OpenAI, merupakan sebuah organisasi nirlaba yang berdiri pada tahun 2015 silam (Taecharungroj, 2023).

Chatbot adalah program yang dibuat untuk tujuan penggunaan otomatis pada kegiatan percakapan yang dilakukan oleh teknologi komputer terhadap manusia. Penerapan kecerdasan buatan di teknologi *chatbot* direncanakan untuk membuat program layanan obrolan yang tidak hanya mampu memahami maknanya dari kalimat yang diucapkan oleh seseorang pengguna, tetapi dapat menyediakan jawaban yang relevan dalam jangka waktu singkat (Dewonoto Laut Santoso et al., 2021). Dan, telah banyak topik pertanyaan atau bahasan yang telah dilontarkan melalui chatgpt *chatbot* ini dari berbagai kalangan pengguna seperti orang biasa, pelajar, guru atau dosen, hingga tenaga peneliti. Namun untuk melihat hal tersebut secara manual, pastinya akan menghabiskan banyak waktu, tenaga hingga memerlukan usaha yang besar. Oleh karena itu, diperlukan suatu metode untuk mengotomisasikan arah analisis tren topik yang telah dilontarkan oleh pengguna tweets twitter tentang chatgpt ini.

Oleh sebab itu, diperlukan suatu metode untuk melacak otomatisasi tersebut dalam membaca arah tren topik ulasan tweets menggunakan pendekatan *topic modelling*. *Topic Modelling* adalah sebuah pendekatan untuk menganalisis kumpulan dokumen teks dan akan mengelompokkan berdasarkan beberapa topik. Metode yang dipakai penulis dalam karya ilmiah ini yakni dengan *topic modelling Latent Dirichlet Allocation* (LDA) (Yoga et al., 2020). LDA merupakan metode *topic modelling* yang berfungsi dalam mengidentifikasi pola dalam teks agar dapat

menghasilkan suatu topik (Sahria & Hatta Fudholi, 2017).

Tentunya, *Latent Dirichlet Allocation* (LDA) sendiri sangat populer dalam ruang lingkup *topic modelling*. Dan telah banyak dikembangkan oleh *author* untuk kemajuan dalam menganalisis data dokumen ataupun teks hingga menjadi sebuah tren topik. Seperti penelitian sebelumnya yang membahas tentang bahasa pemrograman paling populer dalam *generate* kode di chatgpt yang dimana data tersebut diambil dari twitter dan reddit sehingga mendapatkan hasil tentang bahasa pemrograman tertinggi berdasarkan ulasan dari kedua sumber (Feng et al., n.d.). Terdapat penelitian tentang sentimen analisis mengenai reaksi tweets di twitter yang bertujuan untuk mendapatkan hasil topik yang bisa dilakukan oleh chatgpt menggunakan metode LDA (Taecharungroj, 2023). Penelitian selanjutnya mengenai identifikasi kata *translate* yang tidak sesuai menggunakan metode *Latent Dirichlet Allocation* (Vulić et al., 2011).

Berdasarkan penelitian sebelumnya yang telah dibahas tentang *topic modelling*, penulis akan menggunakan metode LDA dengan berfokus pada *topic modelling* dalam ulasan tweets pada twitter mengenai topik tren yang selama ini dibahas. Setelah data berhasil dikumpulkan, lalu akan diproses menggunakan *build* model LDA, dan hasil dari tujuan penelitian penulis membahas tentang top topik yang tren dikalangan chatgpt sehingga memudahkan *user* atau pengguna dalam lebih bijak menggunakan chatgpt tersebut.

2. Metode Penelitian

Untuk penelitian ini, penulis membagi proses menjadi beberapa tahap yakni :

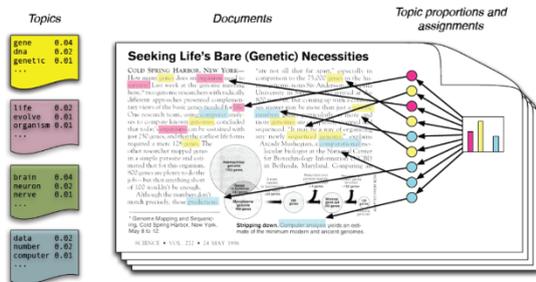
- a. Studi Literatur
Cara yang dipakai dalam mengumpulkan informasi pada tahap ini yaitu dengan melalui referensi penelitian terdahulu di perpustakaan, internet dan jurnal.
- b. Pengumpulan Data
Pengumpulan data yang dilakukan penulis diambil dari tweets atau ulasan twitter mengenai topik chatgpt yang telah tersedia di situs kaggle.com dataset.

c. Preprocessing

Tahap preprocessing ini berfungsi agar semua ulasan yang ada menjadi tersusun secara sistematis dan bersih. Tahap – tahap tersebut terdiri dari *upload* data atau file CSV yang telah berhasil di *scrapping* sebelumnya. Lalu langkah kedua melakukan *data cleaning* untuk menghilangkan link, emoticon, *stopword*, dan tanda baca yang tidak diperlukan. Kemudian langkah terakhir yaitu melakukan *explore* supaya data hasil *cleaning* sebelumnya terverifikasi kelayakannya untuk dimodelkan menggunakan LDA dan ditampilkan melalui *wordcloud* dan *pyLDAvis* agar lebih mudah dimengerti saat dilihat. Penulis menggunakan bantuan *software Anaconda3 Jupyter Notebook* dalam melakukan proses tersebut.

d. Pemodelan LDA

Sumber ide dari pemodelan *topic modelling* yaitu tentang sebuah topik yang disusun dari dokumen atau kata – kata tertentu. Pemodelan LDA dalam kasus penelitian penulis ini yakni untuk mengedukasi pengguna chatgpt agar lebih bijak dalam menggunakan *chatbot* tersebut.



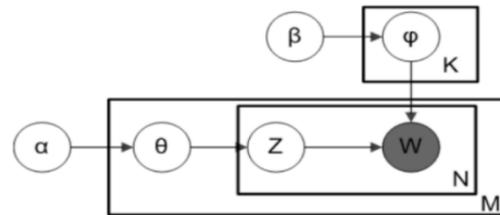
Gambar 1. Sistematis Alur Kerja LDA

Cara kerja LDA model tersebut yakni dengan cara mengasumsikan topik yang telah dispesifikasikan lalu menetapkan jumlah topik yang diinginkan terlebih dahulu. Seperti studi kasus diatas yakni:

1. Memiliki 3 topik (yang digambarkan dengan grafik disebelah kanan).
2. Untuk kata yang terletak di dalam dokumen:
 - a) Dipilih secara acak topik dan menyesuaikan warna dari setiap pasangan grafik (pada gambar lingkaran dan grafik).

- b) Total jumlah warna lingkaran yang telah ditentukan sebelumnya akan menyesuaikan dengan kata yang ada sesuai jumlah kosakata di dalam dokumen.

LDA divisualisasikan dengan model grafis menggunakan *plate notation* seperti berikut:



Gambar 2. LDA Plate Notation

Dimana:

- N = Kumpulan kata
- W = Kata
- K = Kumpulan topik
- Z = Topik *index assignment*
- M = Kumpulan dokumen
- θ = Dokumen
- β = *Dirichlet* parameter atas distribusi kata terhadap topik
- φ = Distribusi kata terhadap topik *corpus*
- α = *Dirichlet* parameter atas distribusi topik terhadap dokumen.

LDA memiliki rumus sebagai berikut:

$$p(w,z,\theta,\varphi|\alpha,\beta) = p(\varphi|\beta)p(\theta|\alpha)p(z|\theta)p(w|\varphi_k)$$

Saat implementasi LDA tidak dapat diterapkan, karena variabel Z , θ , dan φ tidak dapat ditemukan atau tersembunyi lalu untuk menemukan asosiasi antara z dan w itu sulit, karena sebuah kata dapat memiliki arti 2 topik atau lebih (Setijohatmo et al., 2020).

e. Analisis dan Evaluasi

Di tahap ini, hasil pengelompokan topik perlu diperhatikan *term-frequency* dan analisis kualitatif nya. Dari pengelompokkan data yang telah dibuat, kemudian dianalisa untuk memberi data yang bermanfaat dan seimbang.

3. Hasil dan Pembahasan

3.1 Pengumpulan Data

Tahap mengumpulkan data dalam penelitian ini yakni dengan mengambil dataset chatgpt terlebih dahulu melalui <https://www.kaggle.com/datasets/>. Kemudian mengunduh file csv yang telah di *scrapping* tersebut. Dataset itu memiliki 202304 ulasan tweets mengenai chatgpt dan ada 14 nama atribut atau *field* didalamnya. Namun penulis hanya membutuhkan 1 kolom yang bernama ‘tweets’ saja sebagai langkah awal dalam proses topik *modelling* ini. Lalu sisa kolom atau atribut akan dihilangkan didalam tahap *preprocessing* data.

3.2 Preprocessing Data

Tahap *preprocessing* bertujuan agar dapat merapikan, menstrukturkan, dan memastikan data siap dianalisis menggunakan LDA (Prasetyanto & Hartanto, 2022). *Preprocessing* akan membersihkan data guna menghindari data tidak sempurna, data tidak konsisten dan bermasalah nya data (Yoga et al., 2020), dengan menggunakan bahasa pemrograman python. Oleh karena itu, *preprocessing* akan menghilangkan links (‘http’), menghapus *emoticon*, melakukan *stopword*, menghilangkan kalimat yang tersusun kurang dari sama dengan tiga, dan menghilangkan tanda baca. Berikut *script preprocessing* dalam *cleaning* data dapat dilihat pada gambar 3 berikut:

```
In [4]: # Remove the columns
df = df.drop(columns=['ID', 'Date', 'Username', 'ReplyCount', 'RetweetCount', 'LikeCount',
                    'QuoteCount', 'OnlyDate', 'OnlyHour', 'OnlyMin', 'Processed_tweet', 'Sentiment_label',
                    'Sentiment_score'], axis=1)
df.head()

Out[4]:
```

	Tweet
0	ChatGPT is another woke machine
1	of the Atlantic, or only near the Atla #8899946 m...
2	This thread is saved to your Notion database t...
3	Prompt AI - ChatGPT #0018
4	Just had some interesting conversations with G...

Gambar 3. Remove Column

Gambar diatas merupakan perintah untuk menghilangkan kolom yang merupakan salah satu proses dalam *cleaning* data. Kolom yang penulis akan hilangkan yakni ‘ID’, ‘Date’, ‘Username’, ‘ReplyCount’, ‘RetweetCount’, ‘LikeCount’, ‘QuoteCount’, ‘OnlyDate’, ‘OnlyHour’, ‘OnlyMin’, ‘Processed_tweet’, ‘Sentiment_label’, ‘Sentiment_score’ dan hanya menyisakan kolom ‘tweets’ yang akan digunakan sebagai bahan topik *modelling* nantinya.

Setelah hanya kolom tweets yang tersisa, kemudian penulis menghilangkan kata yang

mengandung ‘chatgpt’ didalam dataset nya. Karena kata tersebut sudah pasti akan sering muncul dan mempengaruhi nilai koherensi untuk topik *modelling* nya. Setelah itu menghilangkan tanda baca dan *emoticon*. Berikut *script* serta *output* yang dihasilkan pada gambar 4 dibawah ini.

```
In [6]: # Cleaning the Dataset
def clean(text):
    text = str(text).lower()
    text = re.sub('[^a-z]', '', text)
    text = re.sub('[\s+]', ' ', text)
    text = re.sub('!', '', text)
    text = re.sub('\"', '', text)
    text = text.strip()
    return text
df['tweet'] = df['Tweet'].apply(clean)
df.head()

Out[6]:
```

	Tweet
0	chatgpt is another woke machine
1	of the atlantic or only near the atla more to ...
2	this thread is saved to your notion database t...
3	prompt ai chatgpt
4	just had some interesting conversations with g...

```
In [7]: # Menghapus kata pada borts-borts dataset
kata_yang_dihapus = 'chatgpt'
df['tweet'] = df['Tweet'].apply(lambda x: ' '.join(word for word in x.split() if word != kata_yang_dihapus))
df.head()

Out[7]:
```

	Tweet
0	is another woke machine
1	of the atlantic or only near the atla more to ...
2	this thread is saved to your notion database tags
3	prompt ai
4	just had some interesting conversations with g...

```
In [8]: # Remove punctuation
df['tweets_text_processed'] = df['tweet'].map(lambda x: re.sub('[^a-zA-Z]', '', x))
# Convert the titles to lowercase
df['tweets_text_processed'] = df['tweets_text_processed'].map(lambda x: x.lower())
# Print out the first row of df
df['tweets_text_processed'].head()

Out[8]:
```

0	is another woke machine
1	of the atlantic or only near the atla more to ...
2	this thread is saved to your notion database tags
3	prompt ai
4	just had some interesting conversations with g...

Name: tweets_text_processed, dtype: object

Gambar 4. Remove Emoticon dan Punctuation

Setelah data telah di *preprocessing* dan menjadi *clean*, maka dilakukan tahap selanjutnya yaitu *stopword*. *Stopword* adalah proses menghilangkan kata – kata umum yang memiliki fungsi tapi tidak memiliki arti. Berikut *script* pada gambar 5 dibawah ini.

```
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])

# Define functions for stopwords, bigrams, trigrams and lemmatization
def remove_stopwords(texts):
    return [word for word in simple_preprocess(str(doc)) if word not in stop_words for doc in texts]

def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

def make_trigrams(texts):
    return [trigram_mod[bigram_mod[doc]] for doc in texts]

def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
    return texts_out
```

Gambar 5. Stopword

Dalam tahap ini, ditetapkan bahwa *stopword* akan mendeteksi teks dalam bahasa ‘english’, karena penulis menggunakan dataset tersebut.

3.3 Pemodelan LDA

Tahap topik *modelling* LDA di penelitian ini menggunakan *library* yang disediakan *python* yakni mengunduh *pip install LDA*. Setelah install *library*, dilanjutkan dengan *build* model LDA nya seperti gambar 6 berikut.

```
# Build LDA model
lda_model = gensim.models.LdaMulticore(corpus=corpus,
                                       id2word=id2word,
                                       num_topics=10,
                                       random_state=100,
                                       chunksize=100,
                                       passes=10,
                                       per_word_topics=True)
```

Gambar 6. Build Model LDA

Kemudian mencari nilai koherensi setiap topik kata yang telah di *modelling*.

```
from gensim.models import CoherenceModel
from gensim import corpora
import string

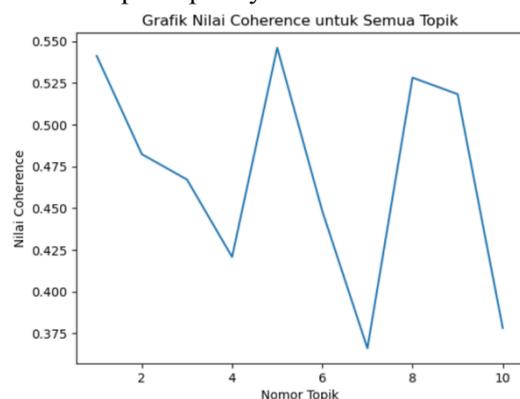
# Menghitung skor koherensi model
coherence_model_lda = CoherenceModel(model=lda_model, texts=texts, dictionary=id2word, coherence='c_v')
coherence_scores = coherence_model_lda.get_coherence_per_topic()

# Menampilkan nilai Coherence untuk semua topik
for topic_num, coherence_score in enumerate(coherence_scores):
    print("Nilai Coherence untuk Topik {}: {}".format(topic_num+1, coherence_score))

Nilai Coherence untuk Topik 1: 0.541187226996475
Nilai Coherence untuk Topik 2: 0.4823808822831266
Nilai Coherence untuk Topik 3: 0.4671238873831912
Nilai Coherence untuk Topik 4: 0.4207533220780588
Nilai Coherence untuk Topik 5: 0.5469713754867564
Nilai Coherence untuk Topik 6: 0.4482861798833746
Nilai Coherence untuk Topik 7: 0.3668525984581769
Nilai Coherence untuk Topik 8: 0.528188124559317
Nilai Coherence untuk Topik 9: 0.5182880351163137
Nilai Coherence untuk Topik 10: 0.37828769949625777
```

Gambar 7. Nilai Koherensi Tiap Topik *Modelling*

Adapun tampilan grafik dari tiap nilai koherensi per topik nya.



Gambar 8. Grafik Nilai Koherensi

```
Topic 1:
['business', 'next', 'free', 'help', 'money', 'make', 'idea', 'marketing', 'product', 'build']

Topic 2:
['work', 'get', 'go', 'use', 'time', 'day', 'say', 'thread', 'prompt', 'save']

Topic 3:
['ai', 'model', 'gpt', 'text', 'datum', 'web', 'powerful', 'train', 'llm', 'base']

Topic 4:
['api', 'user', 'pay', 'people', 'good', 'get', 'see', 'take', 'access', 'man']

Topic 5:
['ai', 'job', 'technology', 'chatbot', 'tech', 'future', 'replace', 'launch', 'artificial_intelligence', 'generative']

Topic 6:
['code', 'create', 'content', 'write', 'human', 'ask', 'less', 'website', 'use', 'make']

Topic 7:
['bing', 'ai', 'new', 'search', 'version', 'make', 'release', 'story', 'update', 'run']

Topic 8:
['ai', 'tool', 'power', 'article', 'new', 'read', 'learn', 'game', 'check', 'app']

Topic 9:
['ask', 'know', 'answer', 'question', 'get', 'think', 'use', 'well', 'give', 'need']

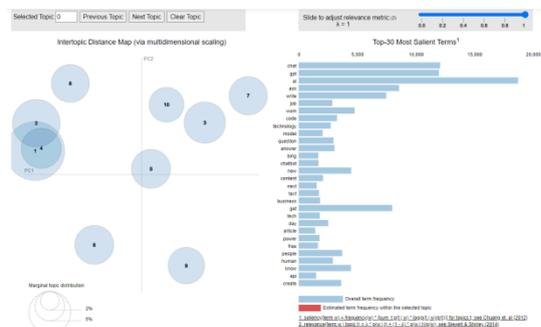
Topic 10:
['chat', 'gpt', 'write', 'ask', 'get', 'tweet', 'thank', 'make', 'use', 'lol']
```

Gambar 9. Hasil Model Topik

3.4 Visualisasi Pemodelan Topik LDA

Setelah berhasil melakukan proses *build* model LDA dan mendapatkan hasil model topik beserta nilai koherensi dari masing – masing topik, kemudian akan dilakukan visualisasi data menggunakan *pyLDAvis*. *pyLDAvis* dirancang agar bisa mempermudah pengguna menafsirkan atau menterjemahkan

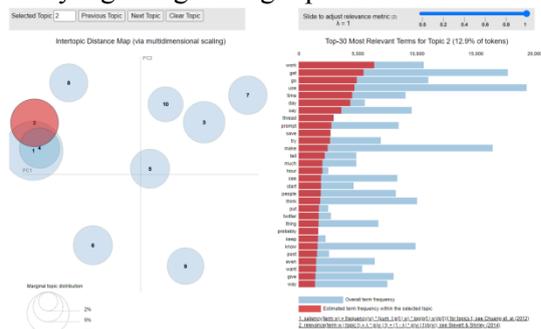
topik *modelling* yang sesuai dengan kumpulan informasi atau data teks yang telah dicari. *Packages* ini bisa mengekstraksi informasi dari model LDA topik kedalam tampilan visual berbasis web interaktif (Luvian chisni chilmi, 2021).



Gambar 10. Visualisasi Topik Dengan *pyLDAvis*

Pada gambar 10 merupakan tampilan visual dengan bentuk *pyLDAvis* yang bisa menentukan visual dari setiap topik serta kata yang paling sering muncul dalam tweets tentang chatgpt. Ada sekitar 30 kata penting yang dominan tersusun didalam *corpus*. Kemudian di sebelah kiri merupakan simbol lingkaran (*circle*) mewakili setiap topik. Alhasil ada 10 pembagian topik *modelling* dari penggunaan dataset chatgpt tersebut.

Jika salah satu simbol lingkaran topik tersebut dipilih (contoh: topik 2), maka disebelah kanan akan menampilkan semua isi kata yang mengandung topik tersebut.



Gambar 11. Visualisasi Model Topik 2

Selanjutnya penulis akan menampilkan visualisasi *wordcloud*. Semakin besar visual kata maka segitu sering kata tersebut muncul. Dengan *wordcloud* akan semakin mudah dalam memahami pemodelan topik yang telah dihasilkan. Terdapat 10 visualisasi *wordcloud* berdasarkan topik yang telah

of the Association for Computational Linguistics: Human Language Technologies, 2, 479–484.

Yoga, S., Nurul Isnaini, F., & Pamulatsih Dwi, O. (2020). Pemodelan Topik Penelitian Bidang Keperawatan Indonesia pada Repository Jurnal Sinta Menggunakan Metode Topic Modelling LDA (Latent Dirichlet Allocation). *Seminar Nasional Teknologi Informasi & Ilmu Komputer*, 1, 90–102.