

***DYNAMIC TOPIC MODELLING MENGGUNAKAN BERTOPIC***  
**DALAM PEMILIHAN PRESIDEN TAHUN 2019**



Universitas Islam Negeri

**PROGRAM STUDI MATEMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH**

**JAKARTA**

**2024 M/1445 H**

***DYNAMIC TOPIC MODELLING MENGGUNAKAN BERTOPIC***  
**DALAM PEMILIHAN PRESIDEN TAHUN 2019**

**SKRIPSI**

**Diajukan untuk Memenuhi Salah Satu Persyaratan  
dalam Memperoleh Gelar Sarjana Matematika (S.Mat)**



**Disusun Oleh :  
Muhammad Raihan  
11180940000022**

Universitas Islam Negeri  
**SYARIF HIDAYATULLAH JAKARTA**  
PROGRAM STUDI MATEMATIKA

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2024 M/1445 H**

### **PERNYATAAN**

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN

Jakarta, 5 Januari 2024



Muhammad Raihan  
NIM. 11180940000022

## **PERSEMBAHAN**

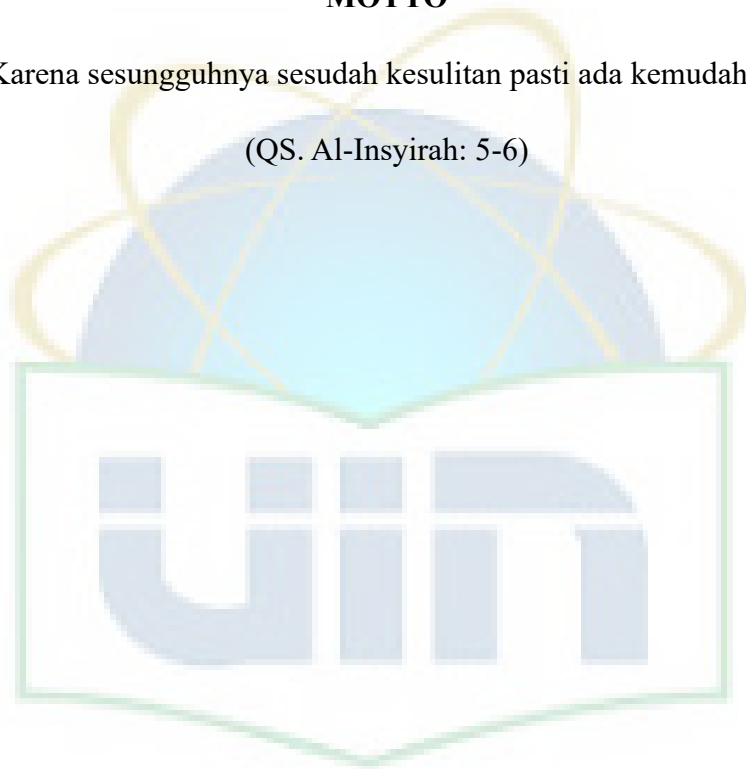
Skripsi ini saya persembahkan untuk kedua orang tua saya yang selalu mendukung dan mendoakan yang terbaik untuk saya

Bapak Mulliyadi dan Ibu Ratnawati

## **MOTTO**

“Karena sesungguhnya sesudah kesulitan pasti ada kemudahan.”

(QS. Al-Insyirah: 5-6)



Universitas Islam Negeri  
**YARIF HIDAYATULLAH JAKARTA**

## ABSTRAK

**Muhammad Raihan**, *Dynamic Topic Modelling Menggunakan BERTOPIC Dalam Pemilihan Presiden Tahun 2019*. Dibawah bimbingan **Dr. Gustina Elfiyanti, M.Si.** dan **M. Irvan Septiar Musti, S.Si., M.Si.**

Pesatnya perkembangan media sosial di Indonesia pada beberapa tahun terakhir, menyebabkan banyak pelaku politik baik individu maupun partai politik yang memanfaatkan hal ini. Menurut *Hootsuite (We are Social)* dalam laporan digital 2023, Indonesia memiliki 60.2% dari jumlah populasi yang aktif bermedia sosial twitter, sehingga akan banyak pelaku politik yang berkampanye di media sosial khususnya twitter. Di literatur yang membahas topik yang sejenis, jarang yang menganalisis pemodelan topik yang dilanjutkan dengan evolusinya dari waktu ke waktu. Maka dari itu dalam penelitian kali ini, Peneliti akan mencoba menganalisis topik apa saja yang dihasilkan dari *tweet* yang diunggah oleh masyarakat menjelang Pemilu 2019 dan disertai dengan evolusi topiknya dari waktu ke waktu. Metode pemodelan topik yang akan digunakan kali ini adalah *BERTopic*. Metode pemodelan topik ini di dasari *sentence embedding* dengan salah satu jenis arsitektur *neural network* yaitu *Siamese network* sehingga metode ini dapat mengelompokkan kata sesuai konteksnya dalam suatu kalimat. Metode *BERTopic* ini juga dilengkapi dengan fitur *Dynamic Topic Modelling* yaitu metode pemodelan topik yang dilanjutkan dengan mengevolusi setiap topiknya dari waktu ke waktu. Dengan data *tweet* yang ada, metode *BERTopic* mampu menghasilkan topik-topik yang ada dengan baik, hal ini dapat dibuktikan dengan hasil evaluasi dari nilai koheren yang dihasilkan yaitu 0.71. Topik yang dihasilkan juga relevan dan dapat dibuat narasinya.

**Kata Kunci :** *Dynamic Topic Modelling, BERTopic, BERT, Politik, Twitter*

Universitas Islam Negeri  
YARIF HIDAYATULLAH JAKARTA

## ABSTRACT

**Muhammad Raihan**, *Dnamic Topic Modelling* using *BERTOPIC* in the 2019 Presidential Election. Under the guidance of **Dr. Gustina Elfiyanti, M.Si. and M. Irvan Septiar Musti, S.Si., M.Si.**

The rapid development of social media in Indonesia in recent years has caused many political actors, both individuals and political parties, to take advantage of this. According to Hootsuite (We are Social) in its 2023 digital report, Indonesia has 60.2% of the population who are active on Twitter, so there will be many political actors campaigning on social media, especially Twitter. In the literature discussing similar topics, it is rare to analyze topic modeling which continues with its evolution over time. Therefore, in this research, researchers will try to analyze what topics are generated from tweets uploaded by the public ahead of the 2019 Election and accompanied by the evolution of the topics from time to time. The topic modeling method that will be used this time is BERTopic. This topic modeling method is based on embedding sentences with one type of neural network architecture, namely the Siamese network, so that this method can group words according to their context in a sentence. The BERTopic method is also equipped with the Dynamic Topic Modeling feature, which is a topic modeling method that continues by evolving each topic from time to time. With existing tweet data, the BERTopic method is able to produce existing topics well, this can be proven by the evaluation results of the coherent values is 0.71. The resulting topics are also relevant and can easily be narrated

**Keywords:** *Dynamic Topic Modelling, BERTopic, BERT, Politic, Twitter*

Universitas Islam Negeri  
YARIF HIDAYATULLAH JAKARTA

## KATA PENGANTAR

*Assalamu'alaikum Warrahmatullahi Wabarakatuh*

Alhamdulillah kita panjatkan puji dan syukur penulis kepada Allah SWT, karena atas karunia-Nya dapat menyelesaikan sebuah penelitian yang berjudul “*Dynamic Topic Modelling Menggunakan BERTOPIC Dalam Pemilihan Presiden Tahun 2019*” dengan baik dan lancar. Sholawat serta salam tidak lupa penulis panjatkan pada Baginda Nabi Besar Muhammad SAW, semoga kita semua mendapatkan *syafaat* beliau di *yaumul akhir* nanti. Tujuan penelitian ini adalah untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana strata satu (S1) pada Program Studi Matematika UIN Syarif Hidayatullah Jakarta.

Penulis sangat menyadari bahwa penelitian ini dapat diselesaikan dengan baik dengan bantuan dari banyak pihak. Untuk itu, pada kesempatan ini penulis ingin menyampaikan terima kasih kepada :

1. Bapak Ir. Nashrul Hakiem, S.Si., M.T., Ph.D. selaku Dekan Fakultas Sains dan Teknologi
2. Ibu Dr. Suma'Inna, M.Si. selaku Kepala Prodi Matematika dan Ibu Irma Fauziah, M.Sc. selaku Sekretaris Prodi Matematika, yang senantiasa membantu administrasi saya sebagai mahasiswa
3. Dr. Gustina Elfiyanti, M.Si. selaku pembimbing skripsi I dan Bapak Muhammad Irvan Septiar Musti S.Si., M.Si. selaku pembimbing skripsi II, yang selalu membantu dan meluangkan waktunya untuk penulis dalam melakukan penelitian ini
4. Bapak dan Ibu dosen di prodi matematika yang memberikan ilmunya yang sangat luar biasa kepada saya
5. Ayahanda tercinta Bapak Mulliyadi yang dengan luar biasa bekerja keras untuk menyekolahkan anaknya dan mendidik saya di rumah
6. Ibunda tercinta Ratnawati. yang perjuangan dan didikannya mengantarkan saya sampai ke titik ini

7. Kawan-kawan Kampus . Ka hany ,Alamsyah, Lambang , Yusmar , Lukman Riski , Lubis, Riski yang selalu menghadirkan tawa dalam kehidupan kuliah saya
8. Stevanny yang selalu menemani saya mengerjakan skripsi
9. Kawan-kawan Magang (Kementrian Pertanian) Lukman , Lubis , Nanda, Suci yang keren dan hebat
10. Himpunanku tercinta HIMATIKA, terimakasih telah memberikan pengalaman dan pelajaran selama menjadi mahasiswa
11. Kawan-kawan matematika 2018 yang sudah menjadi angkatan yang luar biasa, menjadi rumah, teman bercanda dan teman diskusi
12. Terakhir, terimakasih Raihan yang sudah sampai titik ini, berhasil melalui Hampir 5 tahun belajar. Terimakasih sudah jadi sarjana

Penulis menyadari bahwa penelitian ini tidak lepas dari banyak kekurangan. Oleh karena itu, penulis berharap agar para pembaca sekalian dapat memberikan kritik maupun saran yang dapat membangun tersebut agar dapat menjadi bahan evaluasi untuk penulis kedepannya. Terima kasih.

*Wassalamu'alaikum Warrahmatullahi Wabarakatuh*

Jakarta, 5 Januari 2024



Penulis

Universitas Islam Negeri  
YARIF HIDAYATULLAH JAKARTA



## DAFTAR ISI

PERNYATAAN .....	ii
PERSEMBAHAN .....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	5
1.3. Batasan Masalah .....	6
1.4. Tujuan Penelitian .....	6
1.5. Manfaat Penelitian .....	6
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI .....	7
2.1 <i>Text Mining</i> .....	7
2.1.1. <i>Scrapping</i> .....	8
2.1.2. <i>Preprocessing Text</i> .....	8
2.2. <i>Topic modelling</i> .....	10
2.3. <i>BERT</i> .....	11
2.4. <i>Dimensionality Reduction</i> .....	12
2.5. <i>Clustering</i> .....	13
2.6. <i>Class Based Term Frequency Inverse Document Frequency (c-TF-IDF)</i> 14	
2.7. Evaluasi Model .....	16

BAB III METODOLOGI PENELITIAN.....	18
3.1. Data Penelitian .....	18
3.2. Tahapan Penelitian.....	20
3.3. <i>Preprocessing Text</i> .....	22
3.4. Pemodelan Topik .....	23
3.4.1. <i>Sentence-BERT</i> .....	23
3.4.2. <i>UMAP (Uniform Manifold Approximation and Projection)</i> .....	26
3.4.3. <i>HDBSCAN</i> .....	32
3.4.4. <i>Class Based TF-IDF</i> .....	36
3.5. <i>Dynamic topic modelling</i> .....	37
3.6. Visualisasi dan Interpretasi.....	38
BAB IV HASIL DAN PEMBAHASAN.....	18
4.1. Pengolahan Data.....	18
4.2. Modelling .....	45
4.3. Analisa Hasil .....	46
BAB V KESIMPULAN DAN SARAN.....	51
DAFTAR PUSTAKA.....	53

## DAFTAR GAMBAR

<b>Gambar 1.1</b> Data Komisi Pemilihan Umum.....	3
<b>Gambar 1.2</b> Pengguna Sosial Media di Indonesia .....	4
<b>Gambar 3.1</b> Diagram Alur Penelitian .....	20
<b>Gambar 3.2</b> Diagram <i>BERT</i> [32]. .....	24
<b>Gambar 3.3</b> Cara Kerja <i>SBERT</i> [32]......	26
<b>Gambar 3.4</b> Contoh <i>Simplicial Complex</i> [35]. .....	27
<b>Gambar 3.5</b> Mencari Tetangga Terdekat ke- $n$ [35]. .....	27
<b>Gambar 3.6</b> Memperluas Radius Secara Lokal [35]......	28
<b>Gambar 3.7</b> Membuat Graf "Fuzzy" Antar Titik [35]......	28
<b>Gambar 3.8</b> Algoritma <i>UMAP</i> [24]. .....	29
<b>Gambar 3.9</b> Algoritma Pembuatan <i>Himpunan Local Fuzzy Simplicial</i> [24]. .....	30
<b>Gambar 3.10</b> Algoritma Perhitungan Normalisasi [24]. .....	30
<b>Gambar 3.11</b> Algoritma <i>Spectral Embedding</i> untuk Inisialisasi [24]. .....	31
<b>Gambar 3.12</b> Algoritma Optimalisasi <i>Embedding</i> [24]. .....	31
<b>Gambar 3.13</b> Contoh Visualisasi Data 3D (kiri) menjadi 2D (kanan)......	32
<b>Gambar 3.14</b> Algoritma <i>HDBSCAN</i> . .....	33
<b>Gambar 3.15</b> <i>Core Distance</i> dalam Algoritma <i>HDBSCAN</i> [37]. .....	33
<b>Gambar 3.16</b> Contoh Penetapan <i>Threshold</i> yang Terlalu Tinggi [37]. .....	34
<b>Gambar 3.17</b> Contoh Penetapan <i>Threshold</i> yang Terlalu Rendah [37]. .....	35
<b>Gambar 3.18</b> Membuat Pohon Hierarki [37]. .....	35
<b>Gambar 3.19</b> <i>Core Stability</i> [37]. .....	36
<b>Gambar 3.20</b> Diagram Alur Metode <i>Dynamic Topic Modelling</i> [38]. .....	37
<b>Gambar 4.1</b> Jumlah Data Sebelum dan Sesudah <i>Preprocessing</i> .....	41
<b>Gambar 4.2</b> Countplot <i>Frequency Tweet</i> Bulanan. ....	42
<b>Gambar 4.3</b> <i>Wordcloud</i> Dari Data <i>Tweet</i> . .....	43
<b>Gambar 4.4</b> Akun <i>Twitter</i> Dengan Jumlah <i>Like</i> Terbanyak. ....	44
<b>Gambar 4.5</b> Akun <i>Twitter</i> Dengan Jumlah <i>Tweet</i> Terbanyak. ....	44
<b>Gambar 4.6</b> Banyaknya Topik yang Dihasilkan. ....	47
<b>Gambar 4.7</b> Topik yang Dihasilkan dari Model .....	48
<b>Gambar 4.8</b> Evolusi Topik Pertama.....	49
<b>Gambar 4.9</b> Evolusi Topik Kelima .....	50

## DAFTAR TABEL

<b>Tabel 2.1</b> Contoh Hasil Tokenisasi .....	9
<b>Tabel 2.2</b> Contoh Hasil Penghapusan <i>Stopwords</i> .....	9
<b>Tabel 2.3</b> Contoh Hasil <i>Lemmatisasi</i> .....	10
<b>Tabel 3.1</b> Data Awal Hasil <i>Scrapping</i> .....	18
<b>Tabel 3.2</b> Contoh Hasil <i>Preprocessing</i> .....	23
<b>Tabel 4.1</b> Contoh Hasil <i>Preprocessing</i> .....	18



Universitas Islam Negeri  
YARIF HIDAYATULLAH JAKARTA

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Indonesia adalah negara yang sangat menjunjung nilai-nilai demokrasi. Diselenggarakannya pemilu di Indonesia menunjukkan bahwa Indonesia menganut paham demokrasi. Pemilu pertama kali diselenggarakan di Indonesia pada tahun 1955 untuk memilih anggota Dewan Perwakilan Rakyat (DPR) dan Konstituante. Sedangkan untuk pemilihan umum presiden dan wakil presiden pertama kali diselenggarakan pada tahun 2004.

Pemilihan umum atau yang biasa disebut pemilu merupakan salah satu unsur penting dalam mewujudkan negara yang demokratis. Negara-negara yang menganut paham demokrasi beranggapan bahwa pemilu merupakan tolak ukur demokrasi itu sendiri. Hal ini dikarenakan dengan pemilu demokrasi di suatu negara dapat berjalan. Sebagai pilar utama demokrasi, pemilu merupakan sarana dan momentum terbaik bagi rakyat, khususnya untuk menyalurkan aspirasi politiknya.

Sebelum dilaksanakannya pemilu seluruh kandidat akan melakukan suatu proses kegiatan komunikasi individu dan kelompok yang dilakukan secara terlembaga dan bertujuan menciptakan suatu efek maupun dampak tertentu atau sering disebut kampanye. Dengan adanya kampanye, masyarakat bisa mengenal dengan baik dan juga mengetahui latar belakang dari masing-masing kandidat. Selain itu masyarakat juga akan mengetahui visi dan misi yang akan dilakukan oleh kandidat setelah menjadi presiden nantinya.

Pada saat menuju pemilu 2019, kontestasi politik akan dimulai. Dalam melakukan kontestasi politik atau kampanye terdapat dua cara mulai dari kampanye yang sehat sampai dengan *black-campaign*. Salah satunya adalah menggunakan isu politik identitas. Politik identitas adalah sebuah alat politik suatu kelompok seperti etnis, suku, budaya, agama atau yang lainnya untuk tujuan

tertentu, misalnya sebagai bentuk perlawanan atau sebagai alat untuk menunjukkan jati diri suatu kelompok tersebut.

Pada pemilu serentak 2019 fenomena politik identitas dan agama juga diwarnai dengan berebut suara muslim. Sebagai negara yang mayoritasnya beragama islam, berebut suara muslim terjadi dalam setiap pemilu. Selain itu penyebaran isu *hoax* serta ujaran kebencian untuk menjatuhkan lawan juga sering kali terjadi pada pemilu 2019. Menurut Kementerian Komunikasi dan Informatika (Kominfo) ditemukan 3356 berita *hoax* pada periode Agustus 2018 sampai September 2019. Plt. Kepala Biro Humas Sekretariat Jenderal Kementerian Kominfo Ferdinandus Setu mengatakan bahwa angka berita *hoax* tertinggi terjadi pada bulan April 2019 yang mana bertepatan dengan Pemilu 2019 [1].

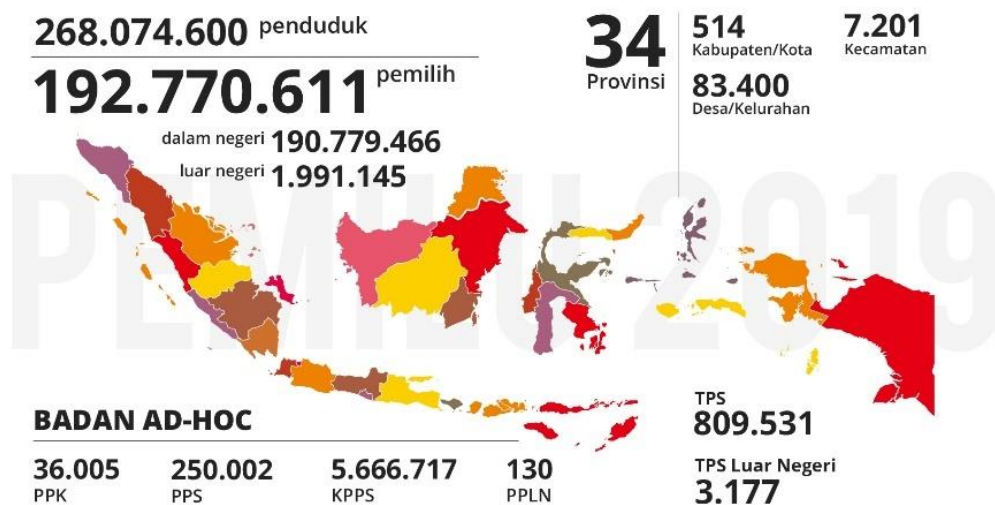
Maraknya isu *hoax* dikarenakan dari rendahnya literasi dan budaya baca dikalangan masyarakat. Agama islam menganjurkan umatnya untuk selalu berhati-hati dalam menerima informasi serta memvalidasi kebenarannya terlebih dahulu. Secara bahasa keagamaan hal ini disebut *tabayyun*. , seperti firman Allah pada surah An-Nuur ayat 11 yang berbunyi:

إِنَّ الَّذِينَ جَاءُوا بِالْإِفْكِ عُصْبَةٌ مِّنْكُمْ لَا تَحْسَبُوهُ شَرًّا لَّكُم بَلْ هُوَ خَيْرٌ لَّكُمْ لِكُلِّ أَمْرٍِ مِّنْهُمْ  
مَا أَكْتَسَبَ مِنَ الْإِثْمِ وَالَّذِي تَوَلَّى كِبْرَهُ مِنْهُمْ لَهُ عَذَابٌ عَظِيمٌ

*“Sesungguhnya orang-orang yang membawa berita bohong itu adalah dari golongan kamu juga. Janganlah kamu kira bahwa berita bohong itu buruk bagi kamu bahkan ia adalah baik bagi kamu. Tiap-tiap seseorang dari mereka mendapat balasan dari dosa yang dikerjakannya. Dan siapa di antara mereka yang mengambil bahagian yang terbesar dalam penyiaran berita bohong itu baginya azab yang besar.” (An-Nuur : 11).*

Pemilu tahun 2019 menjadi topik yang sering dibicarakan. Hal ini menjadi berbeda jika dibandingkan dengan pemilu pada tahun-tahun sebelumnya karena untuk pertama kalinya diadakan pemilu serentak yang melangsungkan pileg (pemilihan legislatif) dan pilpres (pemilihan presiden). Pemilu serentak jauh lebih kompleks dan rumit bagi semua aspek. Dilansir pada komisi pemilihan umum

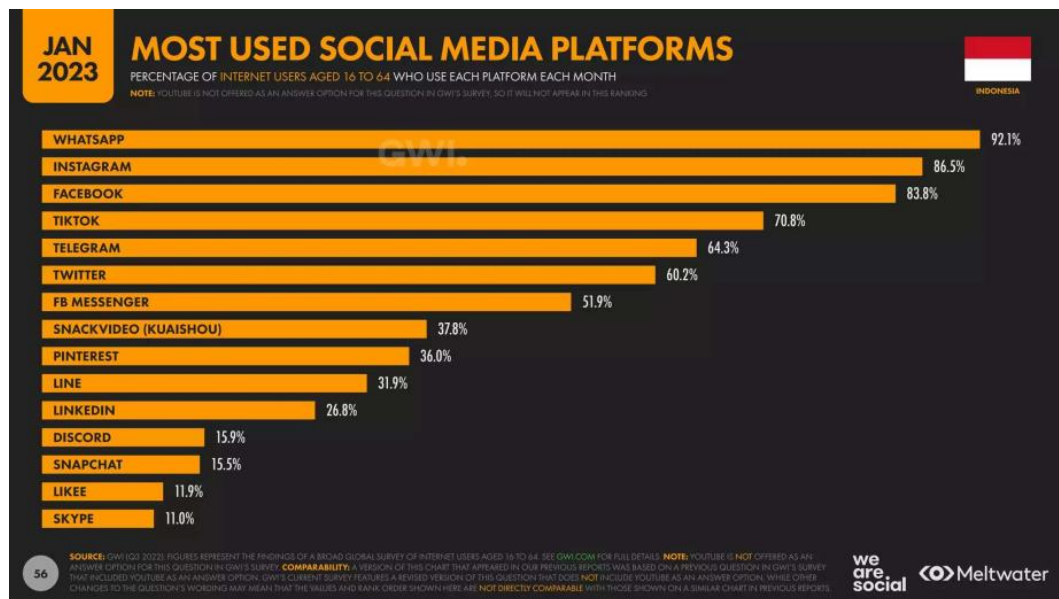
tercatat bahwa terdapat kurang lebih 192 juta jiwa penduduk di Indonesia akan melakukan pemilihan umum 2019 seperti yang terlihat pada gambar 1.2.



**Gambar 1.1** Data Komisi Pemilihan Umum [49].

Menurut World Population Review, Indonesia adalah negara keempat dengan jumlah populasi terbesar di dunia [2]. Hal ini mengakibatkan kontestasi politik di Indonesia akan melibatkan banyak elemen masyarakat. Kemudian dalam masa kampanye tersebut masyarakat dapat mengutarakan aspirasinya serta pendapatnya dari masing-masing kandidat melalui media sosial. Pertumbuhan media sosial mempengaruhi berbagai aktivitas, bahkan kita dapat ikut berpendapat dalam kegiatan kontestasi politik. Hal ini terjadi karena perkembangan teknologi dan informasi di zaman sekarang sangat pesat. Media sosial sendiri memungkinkan penggunaannya untuk membentuk opini publik yang merepresentasikan topik yang sedang dibicarakan. Salah satu media sosial yang populer digunakan masyarakat Indonesia adalah *twitter*. Menurut Hootsuite (*We are Social*) dalam laporan Digital 2023 seperti pada gambar 1.3, Indonesia memiliki pengguna aktif media sosial

twitter sekitar 60.2% dari jumlah populasi, sehingga cocok dijadikan alat politik baru di Indonesia [3].



Gambar 1.2 Pengguna Sosial Media di Indonesia [3].

Dengan adanya media sosial twitter , masyarakat Indonesia dapat menyampaikan aspirasinya serta keluhannya selama masa kampanye berlangsung dengan membuat sebuah *tweet* . Dan dari kumpulan *tweet* tadi dapat dijadikan sebuah tujuan untuk penelitian ini yaitu untuk membantu mencari tahu topik atau pembahasan apa yang sedang dibicarakan oleh masyarakat Indonesia menjelang pemilu 2019. Kemudian dari topik-topik yang dihasilkan tersebut juga bisa dijadikan sebagai *counter* narasi dari topik-topik yang menghasilkan isu-isu negatif agar menjadi tepat sasaran. Lalu peneliti juga ingin melihat apakah ada perubahan atau evolusi topiknya dari waktu ke waktu dengan tujuan ingin memahami bagaimana suatu topik diwakili diwaktu yang berbeda. Oleh karena itu, penelitian ini menggunakan *BERTopic* dengan fitur tambahan yaitu *Dynamic Topic Modelling* ,sebab metode ini selain untuk mengekstrak topik juga dapat melihat evolusinya dari waktu ke waktu.

Data yang digunakan pada penelitian ini diambil dari penelitian terdahulu , maka dari itu referensi utama penelitian ini adalah dari jurnal penelitian Fadlan Bima yang berjudul Analisis Eksplorasi Data Pada Kampanye Kandidat Pemilihan Uum Presiden Indonesia Tahun 2019 Di Media Sosial Twitter. Fadlan melakukan



sebuah Analisis data eksploratif (*exploratory data analysis*) pada pemilu 2019, khususnya pada pemilihan presiden 2019 yang melibatkan 2 pasangan calon yaitu Ir. Jokowi – K.H Ma'ruf Amien dan H.Prabowo – Sandiaga Salahudin Uno. Pada penelitian ini Fadlan Bima Hermawan melakukan beberapa analisis eksploratif untuk mendapatkan gambaran sesungguhnya pada kampanye media sosial di *twitter*. Selain itu penelitian ini juga melakukan analisis jejaring sosial (*social network analysis*) untuk mengetahui *user* yang berpengaruh saat kampanye media sosial berlangsung [4].

Kemudian referensi kedua adalah penelitian dari Lukas Lefebure yang berjudul *Exploring the UN General Debates with Dynamic Topic Models*, mendemostrasikan kemampuan metode *Dynamic Topic Model* untuk menemukan narasi yang berkembang dalam teks yang tidak terstruktur pada korpus pidato. Setiap musim gugur, para pemimpin berkumpul di New York untuk berpartisipasi dalam debat umum PBB. Pidato-pidato yang disampaikan pada Debat Umum merupakan catatan sejarah mengenai isu-isu yang menarik perhatian dunia internasional. Lukas menganalisis bagaimana topik yang dihasilkan dari pidato tersebut berubah seiring waktu dengan menggunakan *Dynamic Topic Model* yang dapat membantu memodelkan evolusi tersebut secara kuantitatif [5].

Terakhir ada jurnal dari Maarten Grootendorst pada tahun 2022 yang berjudul “*BERTopic: Neural topic modeling with a class-based TF-IDF procedure*”. Jurnal ini berisi pembelajaran terbaru untuk pemodelan topik dengan cara pengelompokkan. Selain itu, *BERTopic* juga melakukan tugasnya dengan menggunakan penyematan antar kata berbasis transformator yang telah dilatih sebelumnya. Untuk menghasilkan representasi topik digunakan prosedur *TF-IDF* berbasis kelas [6].

## **1.2. Rumusan Masalah**

Berdasarkan uraian pada latar belakang penelitian ini, rumusan masalah yang akan dibahas antara lain:

1. Bagaimana cara mengimplementasikan *Dynamic Topic Modelling* menggunakan *BERTopic*?

2. Bagaimana performa *Dynamic Topic Modelling* menggunakan *BERTopic* pada isu Pemilihan Presiden 2019?
3. Berapa banyak *Global Topic Representation* yang dihasilkan dari model?
4. Bagaimana Interpretasi dari *Global Topic Representation* dan evolusi topiknya dari waktu ke waktu ?

### 1.3. Batasan Masalah

Adapun Batasan masalah pada penelitian ini yaitu:

1. Data yang digunakan merupakan data *tweet* dengan *keyword*: “Jokowi”, “Jokowilagi”, dan “Jokowiamin” dari awal bulan Maret 2018 hingga Januari 2019 yang terdiri dari 290.134 *tweet*.
2. Data Penelitian diambil dari penelitian yang dilakukan oleh Fadlan Bima Hermawan *tweet* [4].
3. Metode yang digunakan adalah *BERTopic*.

### 1.4. Tujuan Penelitian

Berdasarkan uraian pada rumusan dan Batasan masalah, beberapa tujuan yang ingin dicapai pada penelitian kali ini adalah sebagai berikut:

1. Mendapatkan hasil akhir pemodelan dengan metode *BERTopic*.
2. Mengetahui bagaimana performa *Topic modelling* yang dihasilkan menggunakan fitur *Dynamic topic modelling* pada metode *BERTopic*.
3. Mengetahui *Global Topic* apa saja yang dibicarakan masyarakat Indonesia menjelang pemilu 2019.
4. Memberikan interpretasi dari *insight* yang dihasilkan pada model.

### 1.5. Manfaat Penelitian

Pada penelitian ini, penulis berharap hasil dari pemodelan topik yang dilakukan dapat membantu masyarakat mengetahui topik apa saja yang dibicarakan serta perubahan topiknya dari waktu ke waktu.

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

Pada penelitian ini memiliki landasan dari beberapa penelitian sebelumnya, yaitu jurnal penelitian tentang “*Dynamic Topic Models*” yang ditulis oleh David M. Blei dan John D. Lafferty [7].

#### 2.1 *Text Mining*

Seiring berjalannya waktu dokumen berupa teks akan bertambah banyak. Banyaknya dokumen teks berasal dari berbagai sumber seperti dari paper, jurnal, berita, opini, email, dan lain-lain. Teknik yang berkembang untuk menggali dokumen teks salah satunya adalah *text mining*. Cara kerja *text mining* yaitu memproses pengambilan intisari dari sebuah dokumen teks sehingga didapatkan hasil yang berguna untuk tujuan tertentu. Hal ini sesuai dengan pengertian *text mining* menurut para ahli yaitu Ronen Feldman dan James Sanger [8], *text mining* dapat didefinisikan secara luas sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan tools analisis yang merupakan komponen-komponen dalam data mining yang salah satunya adalah kategorisasi. Dalam melakukan *text mining*, berikut adalah langkah-langkah yang harus di perhatikan [9]:

1. Mengambil informasi dari data yang ada.
2. Mengidentifikasi informasi yang telah diperoleh.
3. Melakukan pemrosesan terhadap informasi yang diberikan oleh sebuah data teks.
4. Menganalisis pola yang didapatkan.
5. Mengelompokkannya sesuai dengan kategori.
6. Ekstrak informasi berharga dan menyimpannya kedalam database.

*Text Mining* berperan penting dalam Sentimen Analisis, Pemodelan Topik, dan lain- lain. Secara umum, terdapat beberapa proses yang perlu dijalankan dalam melakukan *Text Mining* contohnya untuk melakukan *Dynamic topic modelling*, yaitu sebagai berikut:

1. Pengambilan data dari penelitian terdahulu [4]

2. Pemrosesan data atau *Preprocessing Data*.
3. Pemodelan data yang akan di proses secara berkala.
4. Visualisasi data dari hasil pemodelan yang telah dilakukan.

#### **2.1.1. *Scrapping***

Dataset pada penelitian Fadlan Bima Hermawan [4] diperoleh menggunakan metode *scrapping*. Pada platform twitter setiap minggunya akan ada kata trending yang sering di unggah oleh setiap pengguna akun twitter. Salah satu cara yang bisa dilakukan untuk mengetahui dan mendapatkan informasi pada pembahasan tersebut adalah dengan cara *scrapping*. *Scrapping* yaitu proses pengambilan sebuah dokumen dari internet secara otomatis, atau dengan kata lain teknik mengekstrak informasi dari kumpulan data pada situs web secara otomatis yang selanjutnya bisa di analisa dan didapatkan informasi yang berharga [10].

#### **2.1.2. *Preprocessing Text***

Setelah pengumpulan data, langkah selanjutnya adalah melakukan *preprocessing*. Pada tahap ini akan dilakukan *preprocessing* pada dokumen-dokumen hasil dari *scrapping* yang akan dilakukan proses pemodelan topik. Hal ini bertujuan untuk mengurangi singkatan atau bentuk tidak beraturan pada dokumen yang nantinya akan mempersulit dalam proses analisa teks [11]. Tahapan *preprocessing* terdiri dari beberapa langkah yaitu diantaranya *case folding*, menambahkan titik pada akhir kalimat, menghapus simbol, tokenisasi, melakukan *stopwords*, dan *lemmatisasi*.

##### **1. *Case Folding***

*Case Folding* merupakan salah satu bentuk *preprocessing text* yang sederhana dan sangat efektif meskipun sering diabaikan dalam penggunaannya [12]. Tujuan utama dari *case folding* yaitu menyamaratakan penggunaan huruf kapital dengan mengubah semua huruf menjadi *lowercase* semua. Selain itu, karakter selain huruf dihilangkan dan dianggap sebagai *delimiter*. Ada beberapa cara yang dapat digunakan dalam melakukan *case folding*. Pertama, mengubah *text* menjadi *lowercase*. Kedua, menghapus angka, simbol dan tanda

baca. Dan yang terakhir menghapus *whitespace* atau karakter kosong [13].

## 2. Tokenisasi

Tokenisasi adalah tahap untuk memisahkan teks yang berupa paragraf atau kalimat menjadi token-token berdasarkan tiap kata yang menyusunnya atau dengan kata lain, tokenisasi merupakan suatu proses pemisahan kata-kata dari suatu dokumen [13]. Contoh proses tokenisasi dapat dilihat pada tabel berikut:

**Tabel 2. 1** Contoh Hasil Tokenisasi

Kalimat	“Statistika adalah ilmu yang berhubungan dengan data”
Tokenisasi	{statistika, adalah, ilmu, yang, berhubungan, dengan, data}

## 3. Penghapusan *Stopwords*

Setelah melakukan proses tokenisasi, perlu adanya pembuangan atau *stopwords*. *Stopwords* biasanya terdiri dari kata konjungsi ataupun kata keterangan [14]. Penghapusan *stopwords* juga bergantung terhadap kebutuhan penelitian yang dilakukan. Pada penelitian ini akan dilakukan proses dengan menambahkan *stopwords* ataupun tidak untuk mengamati hasil model yang diperoleh. Jika dilakukan proses *stopwords* pada hasil yang telah melalui proses tokenisasi maka akan diperoleh hasil seperti pada tabel berikut:

**Tabel 2. 2** Contoh Hasil Penghapusan *Stopwords*

Kalimat	“Statistika adalah ilmu yang berhubungan dengan data”
<i>Stopwords</i>	{statistika, ilmu, berhubungan, data}

## 4. Lemmatisasi

*Lemmatisasi* merupakan proses transformasi untuk menemukan normalisasi pada sebuah kata. Proses ini merubah suatu kata menjadi kata dasarnya, sehingga dapat mengelompokkan kata-kata yang memiliki makna yang sama. Hal ini dilakukan agar data yang dihasilkan tidak terlalu besar dan tentunya dapat mengurangi *noise* yang ada pada data. Contoh Proses *lemmatisasi* dapat dilihat pada tabel berikut:

**Tabel 2. 3** Contoh Hasil Lemmatisasi

Token	Statistika, adalah, ilmu, yang, berhubungan, dengan, data
<i>Lemmatisasi</i>	Statistika, adalah, ilmu, yang, hubung, dengan, data

## 2.2. *Topic modelling*

Pemodelan topik merupakan algoritma untuk menemukan tema utama yang berasal dari koleksi dokumen yang besar dan tidak terstruktur. Menurut David M. Blei [15], setiap dokumen dalam korpus mengandung proporsi topik tersendiri dari topik-topik yang dibahas sesuai kata-kata yang terkandung didalamnya. Konsep *topic modelling* menurutnya terdiri dari entitas-entitas yaitu “kata”, “dokumen”, dan “*corpus*”. Kata dianggap sebagai unit dasar dari data diskrit dalam dokumen, di definisikan sebagai item dari kosa kata yang diberi indeks  $n$  untuk setiap kata unik pada dokumen. Sedangkan “dokumen” adalah susunan  $n$  kata-kata dan sebuah *corpus* adalah kumpulan dari  $N$  dokumen. Sementara topik adalah beberapa kata yang dapat menggambarkan sekumpulan dokumen yang dapat merepresentasikannya. Secara sederhana, setiap dokumen dalam korpus mempunyai topik-topik yang dibahas berdasarkan kata-kata yang terkandung di dalamnya. *Topic modelling* di anggap sebagai teknik *Unsupervised Learning* karena tidak melakukan pelabelan pada data yang dilatih. *Unsupervised Learning* adalah penggunaan algoritma kecerdasan buatan atau *Artificial Intelligence* untuk mengidentifikasi pola dalam Kumpulan data yang berisik titik data yang tidak diklasifikasikan atau diberi label. Contoh implementasi dari *Unsupervised Learning* salah satunya yaitu *clustering* yang mana merupakan bagian dari *Topic*

*Modelling* [16]. Tujuan dari *topic modelling* adalah untuk mengungkap variabel laten yang membentuk makna dokumen dan korpus dengan membuat topik atau kumpulan kata dalam dokumen yang diamati [15]. Metode yang umum digunakan untuk melakukan pendekatan *topic modelling* adalah *LDA* (*Latent Dirichlet Allocation*), *LSA* (*Latent Semantic Analysis*) dan *CTM* (*Corelated Topic Model*).

Kemudian pada tahun 2006 ilmuwan David M. Blei mengembangkan metode *Latent Dirichlet Allocation* bersama dengan Lafferty [7]. Mereka memperkenalkan komponen temporal untuk pemodelan topik. Pada pemodelan ini akan diteliti bagaimana topik dalam kumpulan dokumen berkembang dari waktu ke waktu dengan memanfaatkan keadaan. Pemodelan topik dinamis (*DTM*) adalah kumpulan teknik yang ditujukan untuk menganalisis evolusi topik dari waktu ke waktu. Metode ini memungkinkan kita untuk memahami bagaimana suatu topik direpresentasikan pada waktu yang berbeda. Misalnya, pada tahun 1995 mungkin pembicaraan tentang kesadaran lingkungan berbeda dibandingkan pada tahun 2015. Meskipun topiknya sendiri tetap sama yaitu kesadaran lingkungan tetapi representasi sebenarnya dari topik itu mungkin berbeda.

Atau secara sederhana pengertian dari *Dynamic Topic Model* adalah teknik *machine learning* yang digunakan untuk mengungkapkan topik abstrak dalam sebuah korpus atau dokumen serta memperhatikan aspek temporal ke dalam analisis yang memungkinkan untuk mempelajari bagaimana topik tertentu berubah dari waktu ke waktu. *DTM* ini merupakan perluasan dari *topic modelling* biasa karena dapat menangani dokumen yang berurutan.

Jadi, yang membedakan antara *topic modelling* dengan *dynamic topic modelling* adalah aspek temporal. Pada *topic modelling*, model tidak memperhatikan dimensi waktu. Sedangkan untuk *DTM* memperkenalkan dimensi temporal ke dalam analisis sehingga topik tertentu dapat berubah dari waktu ke waktu.

### 2.3. **BERT**

Peneliti ahli dari *Google AI* mengembangkan model bahasa terlatih pada tahun 2018. Penemuan ini diberi nama *BERT* atau *Bidirectional Encoder Representation*



*From Transformer* [17]. Secara sederhana, teknik ini memanfaatkan transformer, sebuah mekanisme yang mempelajari hubungan kontekstual antara kata atau sub-kata dalam sebuah teks [18]. Transformer menyertakan dua mekanisme terpisah yaitu, *encoder* yang membaca input teks dan *decoder* yang menghasilkan prediksi dari teks tersebut.

*Encoder* merupakan salah satu bagian dari arsitektur *Transformer* yang memiliki dua *sub-layer*, yang pertama adalah *multi-head attention* dan yang kedua adalah *fully-connected feed-forward network*. *Multi-head attention layer* berguna untuk membantu encoder untuk fokus kepada suatu kata dan melihat konteks secara keseluruhan dari sebuah *input*. Sedangkan untuk layer *fully-connected feed-forward network* berfungsi untuk menghasilkan vektor yang akan dilanjutkan ke *decoder* [19].

Arsitektur *decoder* sama seperti *encoder* dengan tambahan *sub-layer* ketiga, yaitu *masked multi-head attention*. Layer ini akan menyembunyikan sebagian vektor yang dihasilkan *encoder*. Hal ini bertujuan untuk mencegah *decoder* melihat kata selanjutnya. Dengan begitu, model tidak akan menyalin input dari *encoder*, tetapi mempelajari output yang dikeluarkan oleh *encoder*. Input pertama dari *decoder* adalah sebuah *token start*. Hal ini bertujuan untuk mencegah kekosongan dari pergeseran *output decoder*. *Token end* juga digunakan sebagai penanda akhir kalimat dan *decoder* tidak akan memproses *token end* tersebut [19].

Dalam proses melatih model *BERTopic*, peneliti akan menggunakan salah satu varian *BERT* yaitu, *Sentence-BERT* atau lebih dikenal dengan sebutan *SBERT* [20]. Varian ini merupakan hasil modifikasi dari jaringan pretrained *BERT* yang menggunakan struktur jaringan *siamese* sehingga dapat memperoleh *sentence embedding* yang dapat dibandingkan menggunakan *cosine-similarity* [20].

#### **2.4. Dimensionality Reduction**

Reduksi dimensi atau *Dimensionality Reduction* adalah suatu teknik untuk mengurangi dataset [21]. Teknik ini merupakan hal terpenting dalam menjalankan prosedur *BERTopic* karena hasil dari proses *Sentence-BERT* memiliki dimensi yang cukup tinggi, setidaknya 384 dimensi [22]. Hal ini mengakibatkan proses



“Clustering” kesulitan untuk memproses data yang memiliki dimensi terlalu tinggi. Maka dari itu solusi terbaik adalah dengan mereduksi data vektor dengan “Dimensionality Reduction” yang akan menghasilkan dimensi yang lebih rendah sehingga proses data tersebut bisa diproses oleh algoritma *clustering* [23].

Metode pengurangan dimensi dapat di kategorikan secara luas dalam dua kelompok yaitu *linear* dan *non-linear*. Untuk *linear* terdapat metode *PCA* (*Principal Component Analysis*) dan *tICA* (*time-structure Independent Component Analysis*). Cara kerja dari metode ini yaitu membangun variabel kolektif baru dengan melakukan kombinasi *linear* dari variabel masukan. Untuk *non-linear*, seperti metode *t-Distributed Stochastic Neighbor Embedding (t-SNE)* bekerja dengan membuat variabel kolektif baru dengan memetakan variabel input ke fungsi *non-linear*. Teknik ini bagus dalam menangkap struktur *non-linear* dalam data berdimensi tinggi. Jika terdapat dua titik berdekatan dalam ruang berdimensi tinggi, mereka memiliki probabilitas tinggi untuk berdekatan dalam dimensi yang rendah.

*UMAP (Uniform Manifold Approximation and Projection)* adalah teknik terbaru pembelajaran *manifold* untuk reduksi dimensi [24] yang ditemukan pada tahun 2018. Teknik ini sangat mirip dengan *t-SNE* karena ketika kita membandingkan visualisasi yang dibuat dengan *t-SNE* dan *UMAP* akan kesulitan membandingkannya. *UMAP* dibuat dari kerangka teoritis yang didasarkan pada geometri dan topologi aljabar Riemannian. *UMAP* juga memiliki beberapa keunggulan signifikan dibandingkan *t-SNE* diantaranya dapat menangkap struktur global lebih baik, lebih cepat dan tidak memiliki keterbatasan komputasi dimensi embedding yang membuatnya layak menjadi tujuan utama dalam mengurangi dimensi yang tinggi untuk *machine learning* [24].

## 2.5. Clustering

*Clustering* atau *klasterisasi* merupakan metode pengelompokan data. *Clustering* dapat diartikan sebagai sebuah proses untuk mengelompokkan data ke dalam beberapa cluster sehingga data dalam satu cluster tersebut memiliki tingkat kemiripan yang maksimum dan data antar cluster memiliki kemiripan yang minimum [25]. Pada *Dynamic topic modelling* klasterisasi sangat berpengaruh

dalam mengelompokkan kata pada data yang telah di *preprocessing* sehingga terbentuk kelompok topik yang ada dalam suatu dokumen.

Hasil *clustering* yang baik akan menghasilkan tingkat kesamaan yang tinggi dalam satu kelas dan tingkat kesamaan yang rendah antar kelas. Kesamaan yang dimaksud merupakan pengukuran secara numerik terhadap dua buah objek. Nilai kesamaan antar kedua objek akan semakin tinggi jika kedua objek yang dibandingkan memiliki kemiripan yang tinggi. Begitu juga dengan sebaliknya. Kualitas hasil *clustering* sangat bergantung pada metode yang dipakai.

Pada penelitian ini algoritma yang digunakan untuk proses *clustering* adalah *HDBSCAN (Hierarchical Density Based Spatial Clustering of Application with Noise)*. Metode ini bekerja dengan pengelompokan bertingkat atau hierarkis yang berbasis kepadatan yang telah dimodifikasi secara teoritis dan praktikal [26]. Algoritma ini diturunkan atau diperluas dari algoritma sebelumnya yaitu *DBSCAN (Density Based Spatial Clustering of Application with Noise)*. Dengan kata lain, algoritma ini akan mengatasi beberapa keterbatasan sebelumnya. *HDBSCAN* menggunakan pendekatan *soft-clustering*, dimana *noise* dimodelkan sebagai *outlier* sehingga dokumen yang tidak terkait tidak akan dimasukkan kedalam cluster. Hal ini akan meningkatkan representasi topik yang akan dihasilkan nantinya.

## **2.6. Class Based Term Frequency Inverse Document Frequency (c-TF-IDF)**

Dalam pencarian informasi, *TF-IDF* adalah statistik numerik yang bertujuan untuk mencerminkan betapa pentingnya sebuah kata adalah dokumen dalam koleksi atau korpus. Tahapan klasik *TF-IDF* yaitu menggabungkan dua statistik “*Term Frequency*” dan “*Inverse Document Frequency*” [27]. Terdapat variasi formula dalam mengimplementasikan metode *TF-IDF* pada pembobotan kata. Nilai *TF-IDF* meningkat secara proporsional berdasarkan jumlah atau banyaknya kata yang muncul pada dokumen, tetapi diimbangi dengan frekuensi kata dalam korpus.

$$W_{t,d} = tf_{t,d} \cdot \log \left( \frac{N}{df_t} \right) \quad (2.1)$$

$tf_{t,d}$  = Frekuensi kata  $t$  pada dokumen  $d$

$df_t$  = Jumlah dokumen yang mengandung kata  $t$

$N$  = Jumlah dokumen di sebuah korpus  $N$

Dari persamaan diatas dapat disimpulkan bahwa, “*Term Frequency*” memodelkan frekuensi kata  $t$  pada dokumen  $d$ , sedangkan “*Inverse Document Frequency*” mengukur jumlah dokumen yang mengandung kata  $t$  atau kata yang tersedia untuk dokumen dan dihitung dengan mengambil logaritma dari jumlah dokumen di sebuah korpus  $N$  dibagi dengan jumlah dokumen yang mengandung  $t$ .

Kemudian persamaan diatas digeneralisasikan ke dalam *cluster* dokumen. Pertama, semua dokumen dalam sebuah *cluster* diperlakukan sebagai satu dokumen. Kemudian, *TF-IDF* disesuaikan untuk representasi ini dengan menerjemahkan dokumen ke dalam *cluster*; sehingga terbentuk rumus berikut:

$$W_{t,c} = tf_{t,c} \cdot \left( 1 + \frac{A}{tf_t} \right) \quad (2.2)$$

$tf_{t,c}$  = Frekuensi kata  $t$  pada *cluster*  $c$

$tf_t$  = Frekuensi kata  $t$  pada semua *cluster*

$A$  = Jumlah rata-rata kata per kelas  $A$

Pada “*Term Frequency*” mempunyai tugas untuk memodelkan frekuensi kata  $t$  pada kelas  $c$  (kelas  $c$  adalah kumpulan dokumen yang digabungkan menjadi satu dokumen dalam *cluster*). Sedangkan untuk “*Inverse Document Frequency*” diubah menjadi “*Inverse Class Frequency*” yang bertujuan untuk mengukur berapa banyak informasi yang disediakan oleh sebuah kata untuk suatu kelas. Kemudian

dihitung dengan mengambil logaritma dari jumlah rata-rata per kelas  $A$  dibagi dengan frekuensi kata  $t$  di semua kelas.

Dengan demikian, prosedur *TF-IDF* berbasis kelas ini memungkinkan kita untuk menghasilkan distribusi topik dari setiap kata untuk setiap kelompok dokumen. Kemudian kita juga dapat menggabungkan *c-TF-IDF* secara iteratif dari topik yang paling tidak umum dengan topik yang paling umum. Sehingga kita dapat mengurangi jumlah topik menjadi nilai yang ditentukan oleh pengguna.

## 2.7. Evaluasi Model

Pada tahap ini sangatlah penting, karena evaluasi model dilakukan untuk melihat seberapa bagus model itu bekerja, sehingga evaluasi model dapat menjadi cara untuk membandingkan satu model dengan model lainnya. Terdapat beberapa metode untuk mengevaluasi sebuah model contohnya, *topic coherence*, dan *running time* [28]. Proses *running time* nantinya akan mendapatkan nilai seberapa cepat model tersebut dapat memprediksi topik dari sebuah dokumen atau data set. Penilaian ini menjadi sangat penting dikarenakan pada pengaplikasiannya kecepatan sebuah model dalam memprediksi suatu topik merupakan hal yang di pertimbangkan ketika model diterapkan pada sebuah program.

Cara lain dalam mengevaluasi model adalah menghitung *topic coherence*. Selain kecepatan, ketepatan sebuah model juga harus dipertimbangkan. Salah satu cara untuk mengukur ketepatan atau seberapa baik performa sebuah model *Topic modelling* adalah dengan mengukur *topic coherence* dari model tersebut [29]. *Topic coherence* menangkap informasi semantik dari topik yang dihasilkan dan menilai interpretasi topik tersebut. Semakin tinggi nilai *topic coherence* maka semakin baik model tersebut. Konsep dari *topic coherence* yaitu menggabungkan sejumlah ukuran ke dalam kerangka kerja untuk mengevaluasi koherensi antara topik yang disimpulkan oleh sebuah model. *Topic coherence* dihitung dengan melakukan perbandingan berpasangan antar kata dalam sebuah topik tertentu yang nantinya akan menghasilkan sebuah ukuran standar kualitas suatu topik [29]. Berikut ini adalah rumus untuk menghitung nilai koheren [30] :

$$\phi_k = \sum_{j=2}^N \sum_{i=1}^{j-1} \log \frac{P(w_i^{(k)}, w_j^{(k)}) + \epsilon}{\frac{P(w_i^{(k)}) P(w_j^{(k)})}{-\log(P(w_i^{(k)}, w_j^{(k)}) + \epsilon)}} \quad (2.3)$$

Keterangan:

$P(w_i^{(k)}, w_j^{(k)})$  = Peluang munculnya kata  $w_i$  dan kata  $w_j$  yang muncul secara bersama-sama dalam koleksi dokumen pada topik ke- $k$ .

$P(w_i^{(k)})$  = Peluang munculnya kata  $w_i$  yang muncul dalam koleksi dokumen pada topik ke- $k$

$P(w_j^{(k)})$  = Peluang munculnya kata  $w_j$  yang muncul dalam koleksi dokumen pada topik ke- $k$



Universitas Islam Negeri  
YARIF HIDAYATULLAH JAKARTA

## BAB III

### METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan langkah-langkah dalam melakukan penelitian ini. Adapun beberapa hal yang akan dijelaskan yaitu meliputi sumber data, tahapan, dan diagram alur penelitian.

#### 3.1. Data Penelitian

Dataset yang digunakan dalam penelitian ini diperoleh dari penelitian yang berjudul “Analisis Eksplorasi Data Pada Kampanye Kandidat Pemilihan Umum Presiden Indonesia Tahun 2019 di Media Sosial Twitter” [4]. Dataset yang digunakan dalam penelitian kali ini adalah data *tweet* berupa teks dari media sosial twitter (gambar, *retweet*, dan *like* diabaikan) yang diunggah oleh pengguna twitter. Dengan pengambilannya yang dibatasi dari awal Maret 2018 hingga Januari 2019. Kata kunci yang digunakan adalah : “*Jokowi*”, “*Jokowilagi*”, dan “*jokowiamin*”.

Dataset yang diambil sudah dalam bentuk *comma separated value (CSV)*. Total data awal yang ada di dataset sebanyak 243.121 baris dengan 18 kolom. Berikut ini adalah tampilan dataset awal.

**Tabel 3.1.** Dataset Awal

Created_ at	Username	Like_cou nt	Provinsi	Text
01/03/20 18 07:01	RustamIbrahi m	2	Jakarta	kritiklah presiden jokowi sekeras-kerasnya. tapi jangan sekedar pintar mempermainkan kata. tampilkan fakta, angka dan data dan analisislah berdasarkan fakta, angka dan data tersebut. dan jangan berbohong

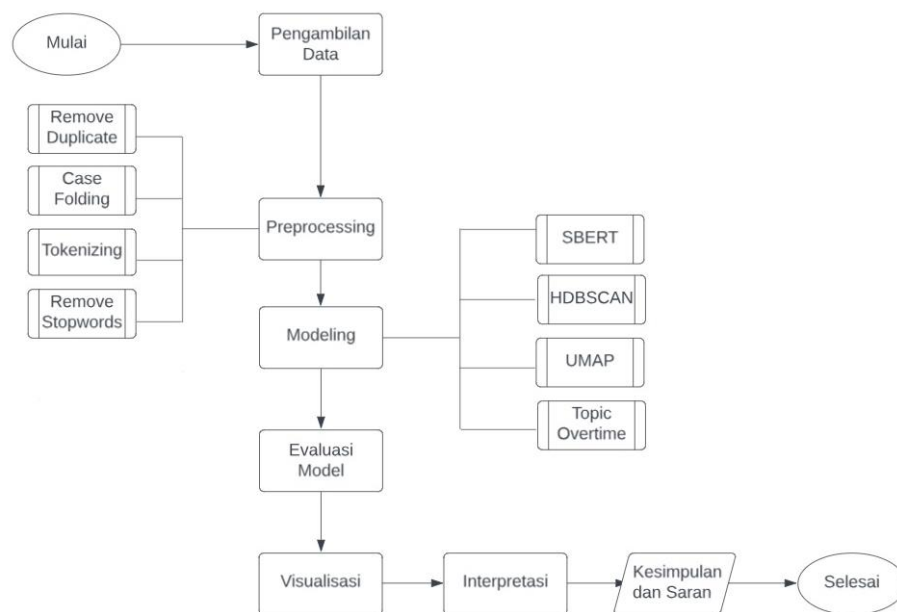
01/03/20 18 07:05	yaya_violin	0	Kepulaua n Riau	@permadiaktivis @azzamizzulhaq @hilmi28 @na_dirs bukan nya terlahir dari orang2 cebong seperti anda ya?? pak presiden, ini ada si penebar kebencian ni pak. mohon diproses @jokowi
01/03/20 18 07:13	Jeffry_joris	10	Jakarta	@liputan6dotcom cc: pakde @jokowi, pak @ganjarpranowo, tolong dibantu pak... mohon kasusnya diluruskan sebenar2nya... kasian ini menyangkut masa depan dan cita2 anak bangsa.
01/03/20 18 07:20	Royke_Yonath an	6	Lampung	@kompascom @yudiwijayaa awas entar ada komentar 'jokowi pencitraan' 😏😏 susahnya jadi presiden, maka biarlah jokowi aja lagi
01/03/20 18 07:24	WidiBudiatmo ko	7	Kalimant an Selatan	@ruhutsitompul kita tak perlu risau dgn yg marah'2 bang..sesungguhnya hasil kerja nyata @jokowi s.d 2018 telah membuka mata hati masyarakat utk tetap memilih beliau pada

				<p>pilpres 2019</p> <p>#jokowimembangunindonesia @pdi_perjuangan</p>
--	--	--	--	--

Dalam penelitian ini dari 18 kolom yang tersedia pada dataset hanya lima kolom yang akan digunakan untuk analisis lebih lanjut. Kolom yang akan digunakan dalam penelitian ini antara lain adalah *created\_at* kolom ini berisi data tentang tanggal kapan suatu *tweet* dibuat, *username* kolom ini berisi tentang nama identitas seseorang yang dipakai di dunia maya, *like\_count* kolom ini berisi tentang seberapa banyak *user* yang menyukai *tweet* dari seorang *user*, *provinsi* kolom ini berisi keberadaan dimana *user* mengunggah *tweet* tersebut dan yang terakhir adalah *text* yang mana kolom ini tentang isi pesan dari sebuah *tweet*.

### 3.2. Tahapan Penelitian

Sub-bab ini akan membahas tahapan-tahapan dalam melakukan penelitian. Seperti yang sudah dijelaskan sebelumnya. Berikut alur penelitian yang dilakukan.



**Gambar 3.1** Diagram Alur Penelitian



Tahap pertama dalam penelitian ini adalah pengambilan data yang sudah ada pada penelitian yang dilakukan oleh Fadlan Bima Hermawan yang berjudul “Analisis Eksplorasi Data Pada Kampanye Kandidat Pemilihan Umum Presiden Indonesia Tahun 2019 di Media Sosial Twitter” [4]. Dalam dataset ini proses pengambilannya menggunakan metode *scrapping* dengan *keyword*: “Jokowi”, “jokowolagi”, dan “jokowiamin”. Dataset yang diperoleh ini sudah dalam bentuk *CSV*. Data awal yang didapatkan berjumlah 243.121 baris dengan 18 kolom yang berisikan: *created\_at*, *user\_id*, *bio*, *url*, *username*, *screen\_name*, *followers\_count*, *friends\_count*, *status\_count*, *text*, *reply\_count*, *retweet\_count*, *like\_count*, *quote\_count*, *media*, *provinsi*, *keywords*, *cleaned\_text*. Tipe data tersebut terdiri dari 4 macam yaitu: *datetime*, *float64*, *int64*, dan *object*.

Data yang sudah didapatkan kemudian akan di-*preprocessing*. *Preprocessing* dilakukan menggunakan bahasa pemrograman python. Pertama-tama, data yang duplikat akan dihapus. Setelah dihapus data yang tersisa menjadi 198.913 baris. Selanjutnya, dilakukan penambahan titik di akhir kalimat dikarenakan model yang berbasis *sentence embedding*. Kemudian dilanjutkan dengan *case folding*, penghapusan simbol dan tanda baca, penghapusan huruf dan kata-kata yang berulang, dan yang terakhir penghapusan *stopword*. *Stopword* yang akan digunakan berformat txt yang berisi kata ganti, kata sambung, kata *slang*, dan lain sebagainya.

Setelah data melewati proses *preprocessing*, yang selanjutnya akan dilakukan yaitu *modelling*. Tahap pertama yang akan dilakukan adalah *sentence embedding* menggunakan salah satu modifikasi dari Transformer *BERT* yaitu *sentence-BERT* atau lebih dikenal dengan *SBERT*. Setiap kata akan dikonversi kedalam bentuk numerik. Data yang didapatkan dari konversi tadi berupa kumpulan kata-kata dalam bentuk matriks berdimensi tinggi. Sebelum diproses menggunakan algoritma *clustering* data matriks berdimensi tinggi harus diubah menjadi data berdimensi rendah menggunakan algoritma *dimensionality reduction*, yang pada penelitian kali ini menggunakan metode *UMAP* (*Uniform*

*Manifold Approximation and Projection*). Setelah data yang akan diproses memiliki dimensi rendah, data baru akan diproses menggunakan algoritma *clustering*, yaitu *HDBSCAN (Hierarchical Density Based Spatial Clustering of Application with Noise)*. Hasilnya, kata-kata dalam data yang masih berbentuk matriks akan dikelompokkan menjadi *cluster-cluster* yang telah membentuk sejumlah topik yang sedang dibicarakan. Selanjutnya, data yang sebelumnya berbentuk matriks akan diubah menjadi bentuk semula dan ditokenisasi. Lalu, kata-kata dalam setiap *cluster* akan diberikan bobot menggunakan algoritma *c-TF-IDF (Class Based - Term Frequency Invers Document Frequency)*. Kemudian untuk mendapatkan representasi topik menjadi berevolusi dari waktu ke waktu yaitu dengan merata-ratakan *c-TF-IDF*. Topik-topik yang telah diekstrak dari data yang kita miliki akan divisualisasikan ke dalam *line chart* untuk melihat topik apa saja yang berevolusi.

### 3.3. *Preprocessing Text*

Pada tahap ini akan menjelaskan terkait *preprocessing* dan hasilnya. Data teks twitter memiliki format yang tidak terstruktur sehingga informasi yang didapat belum bisa tersampaikan dengan baik. *Preprocessing text* bertujuan untuk membuat data menjadi lebih terstruktur dan mengurangi pencilaan data seperti singkatan, simbol, dan bentuk lain yang tidak beraturan. Tahapan ini menjadi hal yang sangat penting karena pada tahap ini data akan diolah agar dapat dianalisis lebih lanjut nantinya. Berikut ini adalah tahapan *preprocessing* yang dilakukan:

1. Melakukan *Case Folding*.
2. Pemisahan kata atau tokenisasi.
3. Menambahkan titik pada akhir kalimat.
4. Pembersihan simbol-simbol dan *stopwords*.
5. Melakukan *Lemmatisasi*

Setelah dilakukan *preprocessing*, struktur dan format data menjadi jelas dan akan lebih mudah untuk diproses. Berikut contoh hasil *preprocessing* yang telah dilakukan:

**Tabel 3. 2.** Contoh Hasil *Preprocessing*

Sebelum <i>preprocessing</i>	@kunenmanai @sandiuno 😂😂😂 pdhl basis gerindra minut ibu vonny panambunan 😊 mar ttp bosan kalah 😊 warga minut so tau bapak presiden jokowi yg fokus bekerja membgn indonesia lebih baik #01jokowiamin #2019tetapjokowi skrg rakyat so cerdas
Setelah <i>preprocessing</i>	basis gerindra minut vonny panambunan mar bosan kalah warga minut so presiden jokowi fokus kerja membgn indonesia jokowiamin tetapjokowi rakyat so cerdas

### 3.4. Pemodelan Topik

Tahap selanjutnya dalam penelitian ini adalah mengekstrak topik-topik yang telah diunggah oleh pengguna twitter dengan kata kunci: “Jokowi”, “Jokowilagi”, dan “jokowiamin”. Metode yang digunakan untuk mengekstrak topik-topik dari dalam dokumen disebut *Topic modelling*. Dalam penelitian kali ini, metode yang digunakan adalah *BERTopic*. *BERTopic* merupakan sebuah prosedur pemodelan topik yang memanfaatkan *sentence embedding* dalam memberikan makna semantik sebuah kata dalam kalimat dan juga menggunakan *class based TF-IDF* sebagai algoritma pemberi bobot untuk mengekstrak topik dari *cluster* yang terbentuk dari data. Berikut ini akan di jelaskan langkah-langkah dari *BERTopic*.

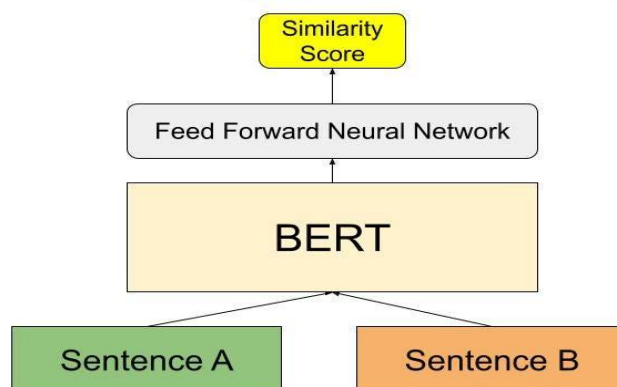
#### 3.4.1. *Sentence-BERT*

Tahap awal *BERTopic* adalah *sentence embedding*. Data *tweet* yang telah di-*preprocessing* akan dimasukkan ke dalam algoritma *sentence embedding*. Pada penelitian kali ini, algoritma *sentence embedding* yang digunakan adalah *Sentence-BERT (SBERT)*. *Sentence-BERT* merupakan salah satu *sentence embedding* yang populer saat ini. *SBERT* adalah pengembangan dari *transformer BERT* [20].

Pengertian *Embedding* yaitu proses merepresentasikan data dalam bentuk vektor numerik dengan memenuhi beberapa sifat yaitu adanya sebuah teks yang

kemudian dilakukan teknik untuk merepresentasikannya ke dalam vektor yang panjangnya sudah ditentukan [31].

Pada umumnya, cara kerja *SBERT* sama seperti *BERT*, namun ada beberapa modifikasi. Modifikasi yang dilakukan yaitu dengan melakukan *sentence embedding* pada setiap kalimat tanpa menggabungkannya. Selain itu, *SBERT* juga menggunakan arsitektur *Siamese Network* yang berisi dua arsitektur *BERT* yang indentik dan memiliki beban yang sama. Salah satu perhitungan *BERT* adalah menghitung skor semantik. Skor ini dihitung dengan cara memasangkan 2 kalimat dan dihitung seberapa besar kemiripannya. Kemudian pada penggunaan transformer *BERT* terdapat kekurangan dalam melakukan prosesnya. Salah satu kekurangannya yaitu waktu yang diperlukan cukup banyak jika digunakan untuk data yang sangat besar. Sebagai contoh misalkan kita memiliki 100 ribu data maka kita perlu memasangkan *sentence* 1 dengan *sentence* 2, *sentence* 1 dengan 3 sampai dengan seratus. Kemudian *sentence* 2 dengan 3, *sentence* 2 dengan 4, dan seterusnya sampai dengan *sentence* 100.



**Gambar 3.2** Diagram *BERT* [32].

Dari permasalahan tadi terbitlah *SBERT* yang di temukan oleh Reimers dan Iryna Gurevich. Cara kerja *SBERT* adalah pertama memasukkan masing-masing *sentence* kedalam *BERT* menggunakan arsitektur *Siamese Network*. Kata dalam setiap tweet akan dikonversi kedalam bentuk *list vector* menggunakan *word embedding*. Algoritma *word embedding* dapat digunakan dalam topic modelling untuk membantu menghasilkan representasi kata-kata yang lebih informatif dan kontekstual. Kemudian *list vector* tersebut di

transformasi sehingga menghasilkan *sentence embedding* dengan berukuran 512 dimensi atau 768 dimensi. Kemudian dilakukan *pooling operation*. *Pooling* sendiri merupakan teknik untuk menggeneralisir fitur dalam *network*. Sedangkan *mean pooling* bekerja dengan menghitung rata-rata dari kumpulan fitur dalam *BERT*. Setelah *pooling* selesai, maka akan menghasilkan dua *embeddings* berbentuk vektor yaitu vektor  $u$  dan satu lagi vektor  $v$ . Saat model dilatih, *SBERT* menggabungkan dua vektor tadi yang kemudian dijalankan melalui *softmax classifier*. *Softmax* merupakan sebuah fungsi yang mengubah vektor dengan nilai  $K$  real menjadi suatu probabilitas. Adapun fungsi *softmax* sebagai berikut [33] :

$$sm(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.1)$$

Pada inferensi, kedua penyematan tersebut kemudian dibandingkan menggunakan fungsi “*Cosine Similarity*” , yang akan menampilkan skor kesamaan untuk kedua kalimat tersebut. *Similarity* adalah suatu fungsi yang disimbolkan dengan huruf  $s$ . Fungsi tersebut dapat ditulis sebagai berikut:

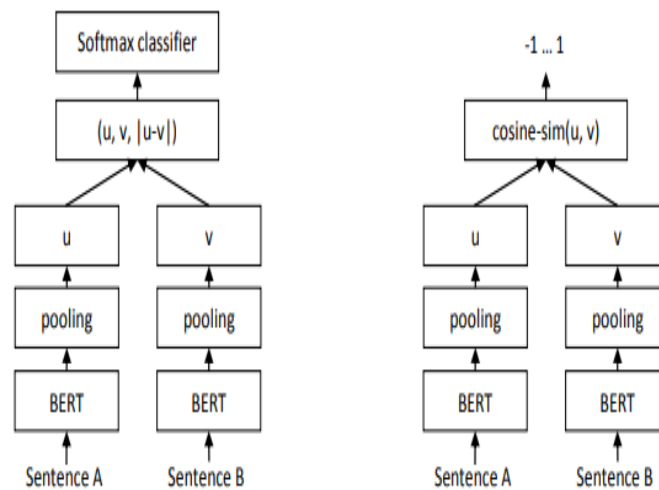
$$s : X \times X \rightarrow [0, u] \text{ dimana } x, y \in X \quad (3.2)$$

Dari fungsi di atas, dapat diketahui bahwa *similarity* adalah suatu fungsi yang terdefinisi antar objek-objek dengan *range* fungsinya 0 sampai 1. Kemudian fungsi ini juga harus memenuhi beberapa sifat diantaranya:

1.  $d(x, y) \geq 0$  (*Non – negatify*).
2.  $d(x, y) \leq u$  (*Boundedness*).
3.  $d(x, y) = u \iff x = y$  (*Identity*).
4.  $d(x, y) = d(y, x)$  (*Symmetry*).

Dari 4 sifat di atas, suatu fungsi dikatakan *similarity* jika memenuhi beberapa sifat yaitu *Non-negativity*, *Boundedness*, *Identity*, *Symmetry* [34].

Berikut ini adalah diagram *SBERT* untuk *fine tuning* dan pada *Inference*

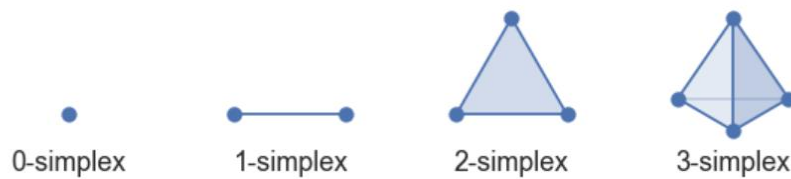


**Gambar 3.3** Cara Kerja SBERT [32].

### 3.4.2. UMAP (Uniform Manifold Approximation and Projection)

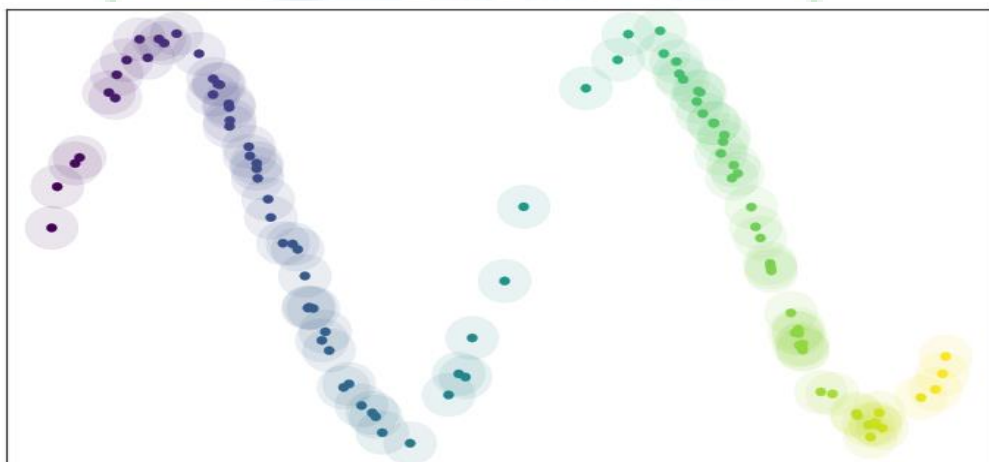
Tahapan selanjutnya adalah mereduksi data vektor tadi dengan menggunakan algoritma *UMAP*. Salah satu fungsi dari algoritma ini adalah membuat data vektor yang memiliki dimensi tinggi menjadi rendah, namun tetap menjaga struktur global dan lokal dari data tersebut [24]. Tahap pertama dalam menjalankan algoritma *UMAP* yaitu dengan membuat representasi graf berdimensi tinggi dari data, kemudian memindahkannya menjadi graf berdimensi rendah tanpa menghilangkan struktur lokal serta global dari data.

Untuk membuat graf berdimensi tinggi, *UMAP* membuat *Fuzzy Simplicial Complex* [24] yang merupakan himpunan yang terdiri dari titik, segmen garis, segitiga dan  $n$  dimensional. Contohnya *0-simplex* hanya ada 1 titik, *1-simplex* menghubungkan antara 2 titik dan menjadi segmen garis. *2-simplex* berarti terdapat 3 titik yang saling menghubungkan dan menjadi segitiga. Untuk *3-simplex* yaitu menghubungkan 4 buah titik. Contoh tersebut digambarkan pada Gambar 3.4



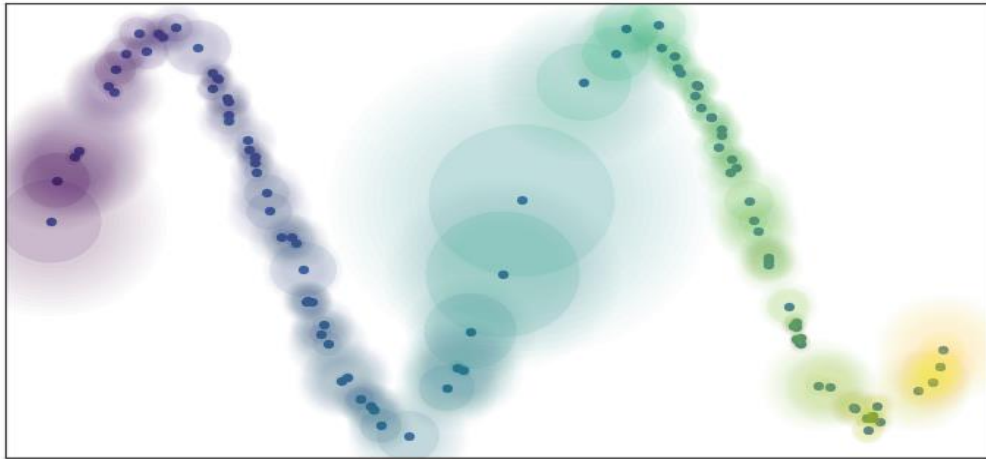
**Gambar 3.4** Contoh *Simplicial Complex* [35].

Untuk menentukan keterhubungan tersebut, *UMAP* mencari tetangga terdekat dengan memperluas radius keluar dari setiap titik yang kemudian menghubungkan titik-titik ketika jari-jari tersebut bertumpukan. Penjelasan ini dapat dilihat pada Gambar 3.5. Memilih radius sangat penting, untuk pemilihan yang terlalu kecil akan menghasilkan cluster yang kecil. Sementara radius yang terlalu besar akan menghubungkan semuanya. Untuk mengatasi hal ini, *UMAP* memilih radius secara lokal berdasarkan jarak ke tetangga terdekat ke- $n$  setiap titik. Penjelasan ini dapat dilihat pada Gambar 3.6. Kemudian *UMAP* membuat graf “fuzzy” dengan mengurangi kemungkinan koneksi saat radius bertambah. *UMAP* memastikan bahwa struktur lokal dipertahankan seimbang dengan struktur global dengan menetapkan bahwa setiap titik harus terhubung setidaknya dengan tetangga terdekatnya [36]. Penjelasan ini dapat dilihat pada Gambar 3.7

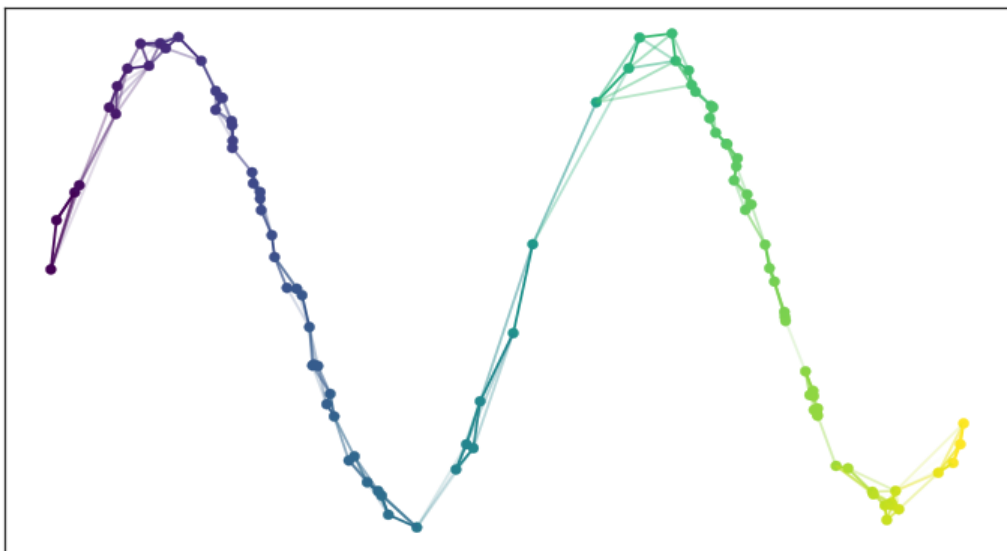


**Gambar 3.5** Mencari Tetangga Terdekat ke- $n$  [35].





**Gambar 3.6** Memperluas Radius Secara Lokal [35].



**Gambar 3.7** Membuat Graf "Fuzzy" Antar Titik [35].

Berikut ini akan dijelaskan bagaimana algoritma *UMAP* dalam implementasinya yang disertakan dengan parameternya.



**Algorithm 1** UMAP algorithm

---

```
function UMAP( $X, n, d, \text{min-dist}, \text{n-epochs}$ )

    # Construct the relevant weighted graph
    for all  $x \in X$  do
         $\text{fs-set}[x] \leftarrow \text{LOCALFUZZYSIMPLICIALSET}(X, x, n)$ 
     $\text{top-rep} \leftarrow \bigcup_{x \in X} \text{fs-set}[x]$     # We recommend the probabilistic t-conorm

    # Perform optimization of the graph layout
     $Y \leftarrow \text{SPECTRALEMBEDDING}(\text{top-rep}, d)$ 
     $Y \leftarrow \text{OPTIMIZEEMBEDDING}(\text{top-rep}, Y, \text{min-dist}, \text{n-epochs})$ 
    return  $Y$ 
```

**Gambar 3.8** Algoritma *UMAP* [24].

Saat melakukan penggabungan *fuzzy* dari himpunan *simplicial fuzzy* lokal hal yang paling efektif adalah menggunakan fungsi probabilitas *t-conorm*. Parameter untuk algoritma *UMAP* adalah sebagai berikut:

- a)  $X$  adalah dataset yang dimensinya akan dikurangi,
- b)  $n$  adalah jarak antar tetangga,
- c)  $d$  adalah target data setelah dimensi dikurangi,
- d) *min – dist* adalah jarak minimum antara titik yang disematkan
- e) *n – epoch* adalah mengontrol jumlah pekerjaan optimalisasi yang harus dilakukan.

Selanjutnya, fungsi individu untuk membangun himpunan *simplicial fuzzy* lokal, menentukan *spectral embedding*, dan mengoptimalkan penyisipan terkait dengan himpunan *fuzzy cross entropy*, dijelaskan lebih rinci di bawah.

---

**Algorithm 2** Constructing a local fuzzy simplicial set

---

```
function LOCALFUZZYSIMPLICIALSET( $X, x, n$ )
  knn, knn-dists  $\leftarrow$  APPROXNEARESTNEIGHBORS( $X, x, n$ )
   $\rho \leftarrow$  knn-dists[1] # Distance to nearest neighbor
   $\sigma \leftarrow$  SMOOTHKNNDIST(knn-dists,  $n, \rho$ ) # Smooth approximator to
  knn-distance
  fs-set0  $\leftarrow X$ 
  fs-set1  $\leftarrow \{([x, y], 0) \mid y \in X\}$ 
  for all  $y \in$  knn do
     $d_{x,y} \leftarrow \max\{0, \text{dist}(x, y) - \rho\} / \sigma$ 
    fs-set1  $\leftarrow$  fs-set1  $\cup ([x, y], \exp(-d_{x,y}))$ 
  return fs-set
```

**Gambar 3.9** Algoritma Pembuatan *Himpunan Local Fuzzy Simplicial* [24].

Pada Gambar 3.9 algoritma ini menjelaskan bagaimana himpunan simplisial *fuzzy* lokal bekerja. Untuk merepresentasikan himpunan simplisial *fuzzy*, kita batasi citra himpunan *fuzzy* antara 0 dan 1 seperti yang dinyatakan pada gambar 3.9 yaitu *fs-set0* dan *fs-set1*. Kita dapat membangun himpunan simplisial *fuzzy* lokal ke titik tertentu  $x$  dengan menemukan  $n$  nearest neighbor atau jarak tetangga terdekat, kemudian akan menghasilkan normalisasi yang sesuai jarak pada ruang topologi, dan setelah itu mengubahnya ke ruang metrik.

---

**Algorithm 3** Compute the normalizing factor for distances  $\sigma$ 

---

```
function SMOOTHKNNDIST(knn-dists,  $n, \rho$ )
  Binary search for  $\sigma$  such that  $\sum_{i=1}^n \exp(-(knn-dists_i - \rho) / \sigma) = \log_2(n)$ 
  return  $\sigma$ 
```

**Gambar 3.10** Algoritma Perhitungan Normalisasi [24].

Pada Gambar 3.10, algoritma ini menjelaskan penghitungan normalisasi. Fungsi yang digunakan adalah *KNN smoothed version* yang memperbaiki kardinalitas himpunan *fuzzy 1-simplices* ke nilai tetap. Untuk  $\log_2(n)$  dipilih berdasarkan percobaan empiris.

**Algorithm 4** Spectral embedding for initialization

---

```

function SPECTRALEMBEDDING(top-rep,  $d$ )
   $A \leftarrow$  1-skeleton of top-rep expressed as a weighted adjacency matrix
   $D \leftarrow$  degree matrix for the graph  $A$ 
   $L \leftarrow D^{1/2}(D - A)D^{1/2}$ 
  evec  $\leftarrow$  Eigenvectors of  $L$  (sorted)
   $Y \leftarrow$  evec[1.. $d + 1$ ] # 0-base indexing assumed
  return  $Y$ 

```

---

**Gambar 3.11** Algoritma *Spectral Embedding* untuk Inisialisasi [24].

Pada Gambar 3.11, algoritma ini menjelaskan *spectral embedding*. *Spectral embedding* dilakukan dengan mempertimbangkan salah satu kerangka dari representasi topologi *fuzzy* global sebagai grafik berbobot dan menggunakan metode *Laplacian* yang dinormalisasi secara simetris. Komponen terakhir dari UMAP adalah optimalisasi *embedding* melalui minimalisasi himpunan *fuzzy cross entropy* seperti yang dijelaskan pada Gambar 3.12.

**Algorithm 5** Optimizing the embedding

---

```

function OPTIMIZEEMBEDDING(top-rep,  $Y$ , min-dist, n-epochs)
   $\alpha \leftarrow 1.0$ 
  Fit  $\Phi$  from  $\Psi$  defined by min-dist
  for  $e \leftarrow 1, \dots, \text{n-epochs}$  do
    for all  $([a, b], p) \in \text{top-rep}_1$  do
      if RANDOM( )  $\leq p$  then # Sample simplex with probability  $p$ 
         $y_a \leftarrow y_a + \alpha \cdot \nabla(\log(\Phi))(y_a, y_b)$ 
        for  $i \leftarrow 1, \dots, \text{n-neg-samples}$  do
           $c \leftarrow$  random sample from  $Y$ 
           $y_a \leftarrow y_a + \alpha \cdot \nabla(\log(1 - \Phi))(y_a, y_c)$ 
     $\alpha \leftarrow 1.0 - e/\text{n-epochs}$ 
  return  $Y$ 

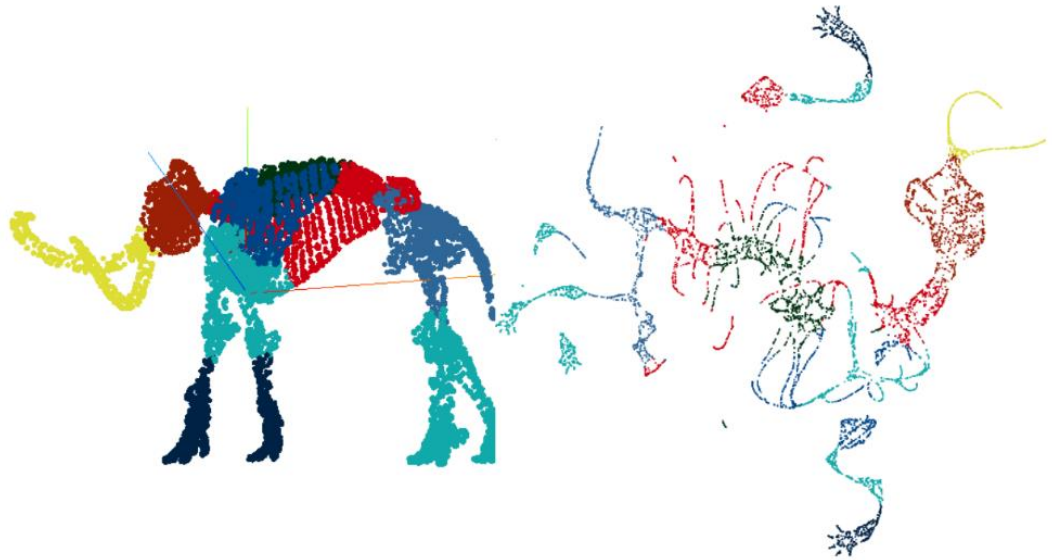
```

---

**Gambar 3.12** Algoritma Optimalisasi *Embedding* [24].

Dengan penjelasan diatas maka data *tweet* yang berbentuk vektor dengan dimensi yang tinggi dapat diubah menjadi data dengan dimensi yang lebih rendah, sehingga data tersebut dapat dilakukan *clustering*. Berikut ini adalah

contoh visualisasi data berdimensi tinggi (3 dimensi) yang telah di reduksi dimensinya menjadi data dengan dimensi yang lebih rendah (2 dimensi), dapat dilihat pada Gambar 3.13.



**Gambar 3.13** Contoh Visualisasi Data 3D (kiri) menjadi 2D (kanan) [37].

#### 3.4.3. *HDBSCAN*

Setelah melewati proses mereduksi data vektor ke dimensi yang lebih rendah, data tersebut bisa diproses dengan algoritma *clustering* agar kata-kata dalam data bisa dikelompokkan sesuai dengan kelompoknya. Salah satu kegunaan utama dari algoritma *clustering* adalah eksplorasi data. Pada penelitian kali ini, algoritma *clustering* yang akan digunakan adalah *HDBSCAN* (*Hierarchical Density Based Spatial Clustering of Application with Noise*). *HDBSCAN* bekerja berdasarkan kepadatan persebaran data. Selain itu, *HDBSCAN* lebih efektif dalam memproses data yang kurang terstruktur karena *HDBSCAN* meninggalkan *noise* pada proses *clustering* data. Hal ini yang membuat keunggulan dari *HDBSCAN* karena kita dapat memproses data walaupun terdapat *noise* dan juga ukuran cluster beserta kepadatan yang berbeda-beda. Algoritma ini merupakan pengembangan dari *DBSCAN* yang dibuat oleh Campello, Muavi, dan Sander. Perbedaan *HDBSCAN* dengan *DBSCAN* adalah dengan menambahkan konsep hirarki dalam menentukan

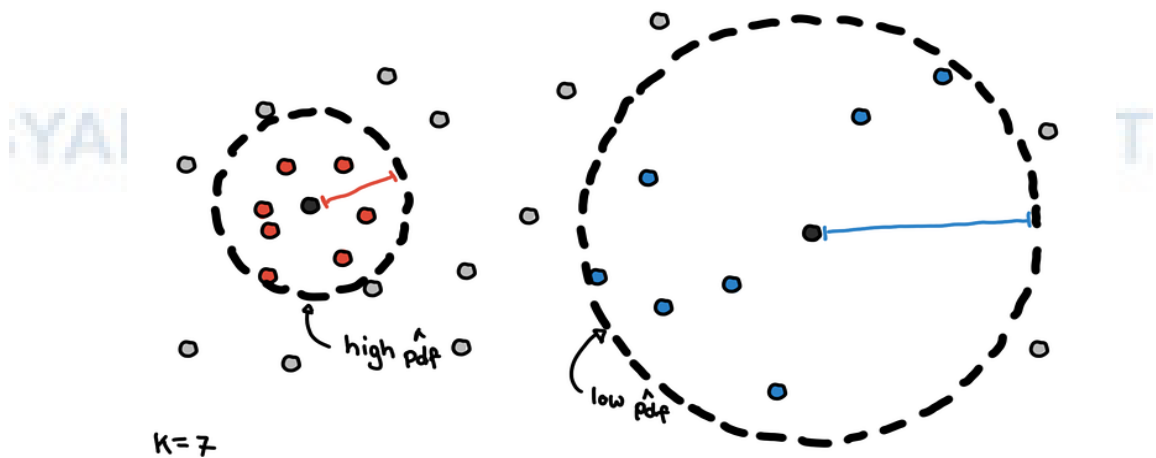
jumlah cluster yang optimal dan dapat menangannya dengan berbagai kepadatan.

**Algorithm 1.** HDBSCAN main steps

1. Compute the core distance w.r.t.  $m_{pts}$  for all data objects in  $X$ .
2. Compute an MST of  $G_{m_{pts}}$ , the Mutual Reachability Graph.
3. Extend the MST to obtain  $MST_{ext}$ , by adding for each vertex a “self edge” with the core distance of the corresponding object as weight.
4. Extract the HDBSCAN hierarchy as a dendrogram from  $MST_{ext}$ :
  - 4.1 For the root of the tree assign all objects the same label (single “cluster”).
  - 4.2 Iteratively remove all edges from  $MST_{ext}$  in decreasing order of weights (in case of ties, edges must be removed simultaneously):
    - 4.2.1 Before each removal, set the dendrogram scale value of the current hierarchical level as the weight of the edge(s) to be removed.
    - 4.2.2 After each removal, assign labels to the connected component(s) that contain(s) the end vertex(-ices) of the removed edge(s), to obtain the next hierarchical level: assign a new cluster label to a component if it still has at least one edge, else assign it a null label (“noise”).

**Gambar 3.14** Algoritma *HDBSCAN* [38].

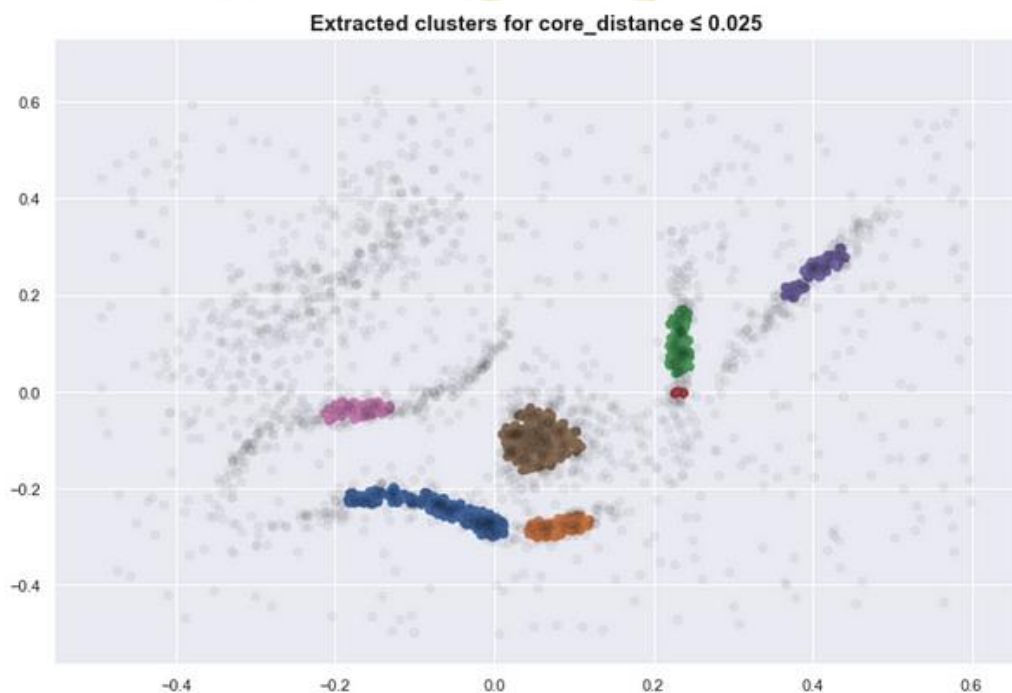
Tahapan dari proses *HDBSCAN* dapat dilihat pada algoritma di Gambar 3.14. Berikutnya algoritma *HDBSCAN* akan dijelaskan secara sederhana dengan disertai ilustrasi agar lebih mudah dipahami. Tahap pertama, *HDBSCAN* akan memperkirakan kepadatan di sekitar titik-titik tertentu pada data dengan menggunakan *core distance* yaitu mengukur jarak suatu titik menuju tetangga terdekatnya [26]. Titik dengan daerah yang lebih padat akan memiliki *core distance* yang lebih kecil dan begitu juga sebaliknya. Hal ini yang membuat algoritma *HDBSCAN* menjadi “*Density Based*” [39].



**Gambar 3.15** *Core Distance* dalam Algoritma *HDBSCAN* [37].

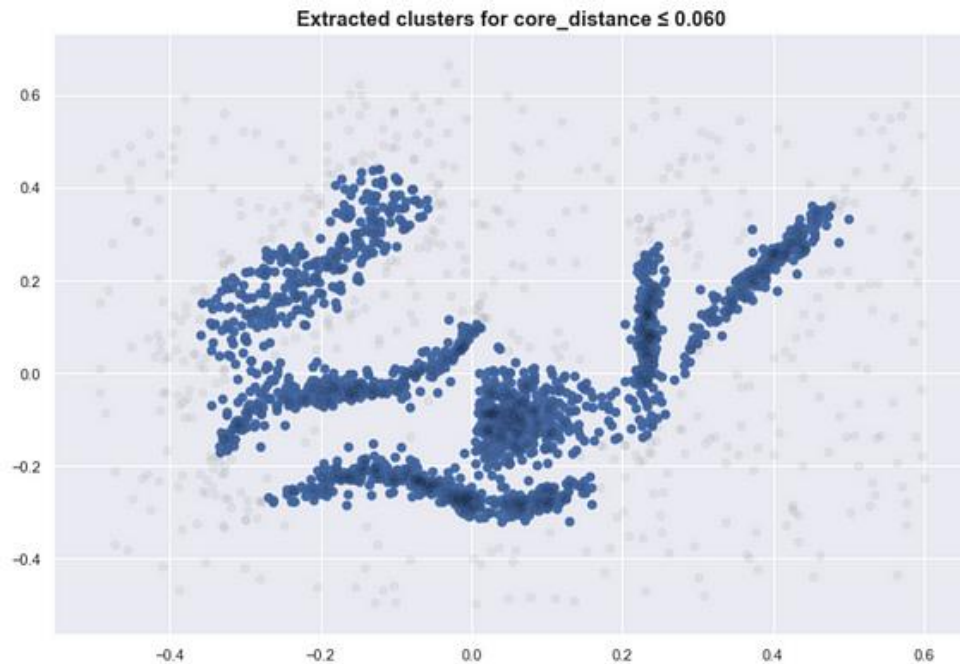


Tahap selanjutnya adalah mengekstrak *cluster* dari data yang telah terbentuk persebaran kepadatan datanya dengan menentukan nilai *threshold*. Dengan mendapatkan titik yang kepadatannya di atas nilai *threshold* dan menggabungkan titik-titik tersebut, maka terbentuklah *cluster-cluster* dari data. Namun, jika *threshold* yang digunakan terlalu tinggi, maka akan banyak titik yang menjadi *noise*. Sedangkan nilai *threshold* terlalu rendah, maka semua titik akan berada di satu cluster yang sama.



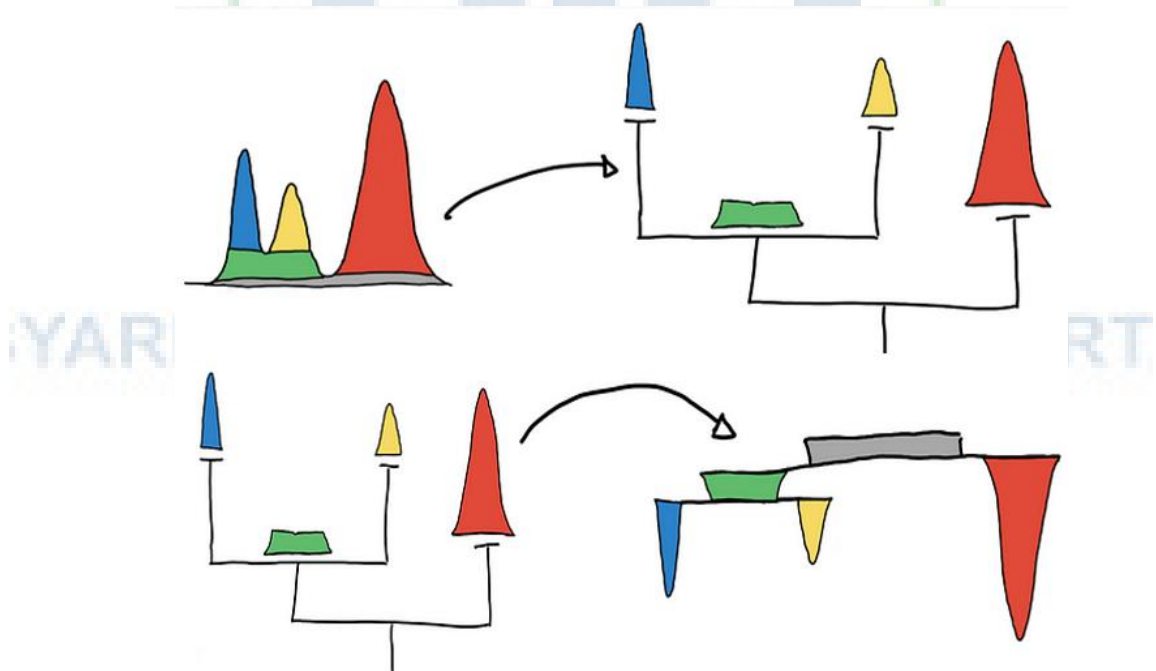
**Gambar 3.16** Contoh Penetapan *Threshold* yang Terlalu Tinggi [37].

YARIF HIDAYATULLAH JAKARTA



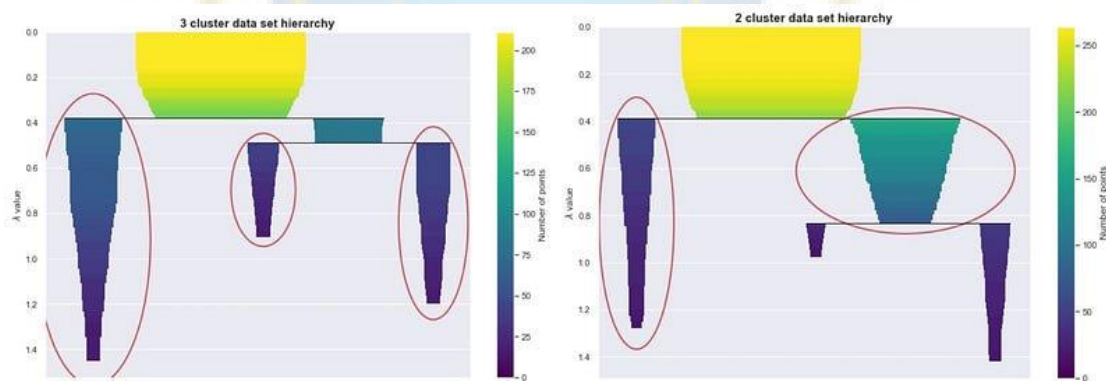
**Gambar 3.17** Contoh Penetapan *Threshold* yang Terlalu Rendah [37].

Untuk mengatasi hal tersebut, *HDBSCAN* membuat hierarki atau tingkatan agar dapat menentukan titik mana yang bergabung menjadi sebuah *cluster*. Kita dapat membuat suatu hierarki dengan cara memvisualisasikannya sebagai pohon seperti berikut:



**Gambar 3.18** Membuat Pohon Hirarki [37].

*HDBSCAN* menggunakan “*Cluster Stability*” untuk menentukan titik mana yang akan ‘bertahan’ dan yang tidak. Dalam proses *clustering* ini, kita membutuhkan nilai *threshold* yang berbeda untuk pengelompokan optimal. Ketika dua titik yang memiliki alas yang sama, untuk menentukan apakah masing-masing titik tersebut adalah sebuah cluster adalah dengan melihat perbandingan volume alas dan puncaknya. Ketika kedua titik tadi merupakan dua *cluster*, maka volume dari dua titik tersebut akan lebih besar dari pada alasnya, Namun, ketika dua titik tersebut sebenarnya adalah fitur dari sebuah *cluster*, maka volume alasnya akan lebih besar dari pada volume dua titik tersebut seperti pada Gambar 3.13. Dengan cara tersebut, *HDBSCAN* dapat memutuskan apakah cluster akan dibagi menjadi subcluster atau tidak.



**Gambar 3.19** Core Stability [37].

#### 3.4.4. Class Based TF-IDF

Pada tahap sebelumnya, data yang sudah terbagi menjadi *cluster* masih berbentuk vector. Untuk menentukan topik apa yang ada di dalam sebuah *cluster*, data vektor harus diubah menjadi kumpulan token kata dan setiap *cluster* akan menjadi sebuah “*Bag of Word*”. Hal ini dilakukan dengan cara tokenisasi menggunakan modul “*CountVectorizer*”. Setelah semua *cluster* berubah menjadi kumpulan kata, barulah kita akan mengekstrak topik dari tiap cluster menggunakan algoritma *c-TF-IDF* [22].

*Class based TF-IDF* merupakan versi dari *TF-IDF* yang memungkinkan untuk mengekstrak topik-topik menarik dari kumpulan dokumen. Tujuan dari *c-TF-IDF* adalah untuk menyediakan semua dokumen dalam satu kelas dengan

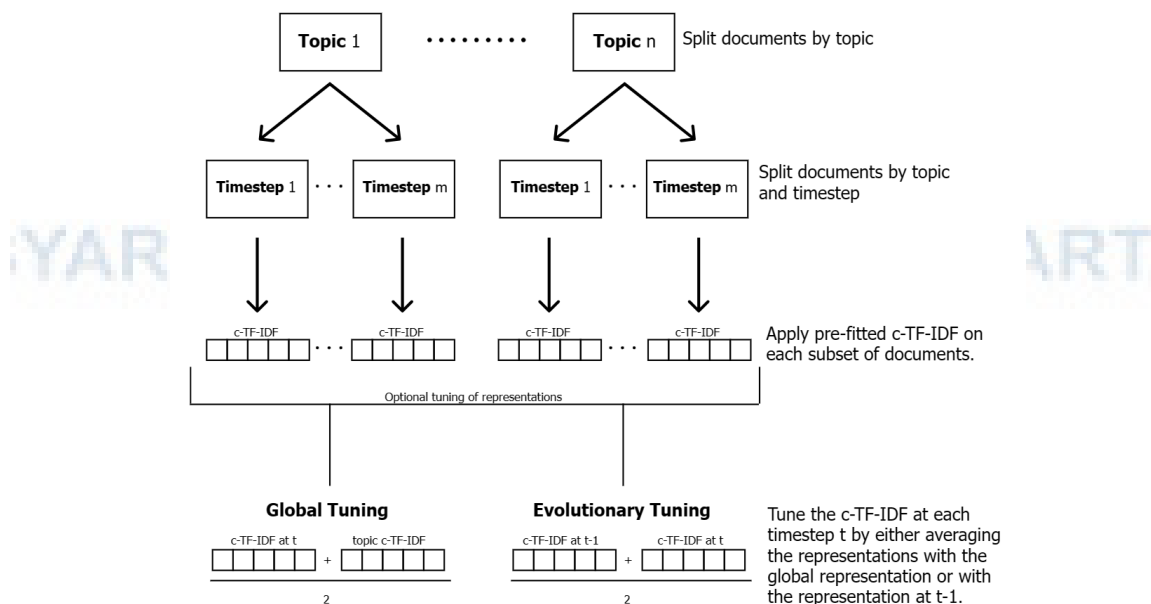


vector kelas yang sama. Maka dari itu, diharuskan untuk melihat *TF-IDF* dari sudut pandang berbasis kelas, bukan dokumen secara individual.

Bobot akan diberikan pada setiap kata menggunakan rumus *c-TF-IDF* dimana setiap kata akan dibandingkan kepentingannya diantara dokumen yang ada. Semakin suatu kata identik dengan suatu *cluster*, sedangkan tidak untuk *cluster* yang lain, maka kata tersebut akan memiliki bobot yang tinggi dalam *cluster* tersebut. Dari sini, data *tweet* sudah berubah menjadi *cluster-cluster* yang memiliki representasi topik dan bisa diinterpretasikan pada setiap *cluster*-nya.

### 3.5. *Dynamic topic modelling*

*Dynamic topic modelling* atau pemodelan topik dinamis merupakan kumpulan teknik yang bertujuan menganalisis evolusi topik dari waktu ke waktu [40]. *Dynamic topic modelling (DTM)* merupakan solusi utama untuk mengekstrak topik dari teks pendek yang dihasilkan dalam *Online Social Networks (OSNs)*. *DTM* sangat berguna pada data twitter. Kita dapat menganalisis topik-topik yang dibicarakan oleh pengguna twitter selama bertahun-tahun mereka menggunakan twitter. Metode ini memungkinkan kita untuk memahami bagaimana representasi dari suatu topik yang dibicarakan di waktu yang berbeda.



**Gambar 3.20** Diagram Alur Metode *Dynamic Topic Modelling* [38].

Terdapat dua cara utama untuk menyempurnakan representasi topik secara spesifik, yaitu *global* dan *evolusionary* [40]. Representasi topik pada *timestep*  $t$  dapat disesuaikan secara global dengan merata-ratakan representasi *c-TF-IDF* dengan representasi *global*. Hal ini memungkinkan setiap representasi topik untuk sedikit mengarah ke representasi *global* sambil tetap mempertahankan kata-kata spesifiknya.

Representasi topik pada *timestep*  $t$  dapat disempurnakan secara evolusioner dengan merata-ratakan representasi *c-TF-IDF* dengan representasi *c-TF-IDF* pada *timestep*  $t - 1$ . Hal ini dilakukan untuk setiap representasi topik yang memungkinkan representasi berkembang seiring waktu. Kedua metode *fine-tuning* tersebut diatur sebagai *default* dan memungkinkan terbentuknya representasi yang menarik.

Pertama, kita perlu memuat dan membersihkan datanya melalui tahap *preprocessing*. Topik yang sama sangat mungkin muncul pada waktu yang berbeda, namun dengan representasi yang berbeda. Dengan demikian, kita perlu menghasilkan representasi secara *global* dari sebuah topik, sebelum kemudian mengembangkan representasi lokal dengan membuat dan melatih *BERTopic* model. Dari topik-topik tersebut akan menghasilkan representasi topik pada setiap waktu untuk setiap topik.

### **3.6. Visualisasi dan Interpretasi**

Tahap akhir dari penelitian ini adalah melakukan visualisasi dari topik yang berhasil diekstrak dari data *tweet* dan menginterpretasikannya. Setelah data *tweet* diproses menggunakan *BERTopic* maka akan terlihat topik apa saja yang menjadi perbincangan oleh pengguna twitter dalam beberapa waktu terakhir. Kata-kata dalam setiap topik akan divisualisasikan dengan probabilitasnya agar dapat lebih mudah dianalisa. Selain itu, persebaran *cluster* dalam data *tweet* juga akan divisualisasikan untuk melihat seberapa baik algoritma *clustering* bekerja. Visualisasi dari hasil yang diperoleh akan memudahkan proses interpretasi topik.

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini akan ditampilkan secara jelas hasil dari proses pemodelan topik serta evolusinya yang telah dilakukan, dari mulai pengolahan data, performa dari pemodelan yang dipakai dan juga analisa evolusi topik dari waktu ke waktu pada penelitian ini.

#### 4.1. Pengolahan Data

Data tweet dari *user* twitter yang diambil dengan keyword “Jokowi”, ”Jokowilagi”, ”Jokowiamin” disimpan dalam format *CSV*. Kemudian data yang sudah dalam format *CSV* di upload ke *google drive*. Proses pengolahan data ini menggunakan fitur yang disediakan oleh google yaitu *google colab*. Setelah itu hubungkan antara *google drive* dengan *google colab*.

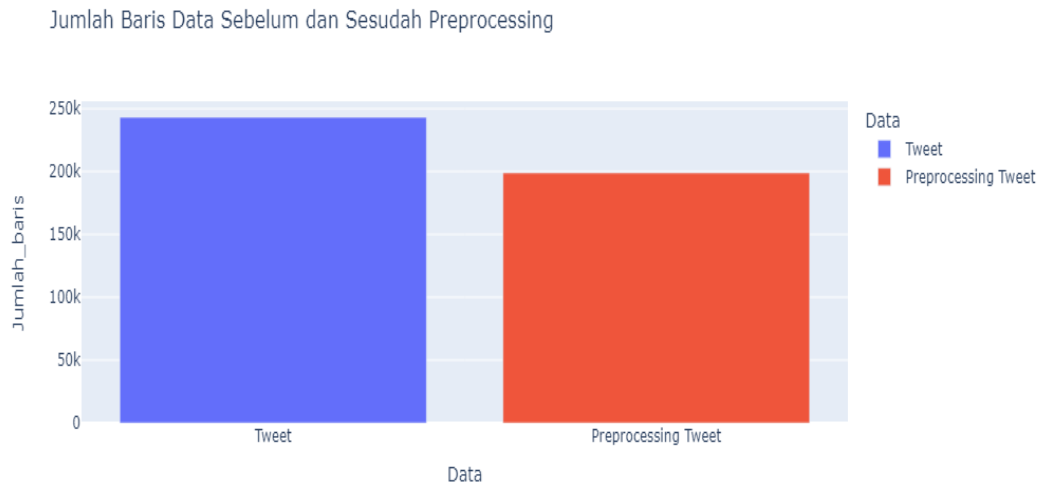
Setelah menghubungkan *google drive* dengan *google colab*, langkah selanjutnya yaitu melakukan *preprocessing*. Pada tahap ini dikarenakan data yang digunakan merupakan data teks yang tidak terstruktur maka *processing* yang dilakukan adalah *Case Folding*, menghapus simbol, menghapus stopwords, lemmatisasi dan *postagging*. Hasil dari tahap *preprocessing* seperti pada Tabel

**Tabel 4. 1.** Contoh Hasil *Preprocessing*

Data sebelum <i>preprocessing</i>	Data setelah <i>preprocessing</i>
kritiklah presiden jokowi sekeras- kerasnya. tapi jangan sekedar pintar mempermainkan kata. tampilkan fakta, angka dan data dan analisislah berdasarkan fakta, angka dan data tersebut. dan jangan berbohong	kritiklah presiden jokowi sekeras kerasnya sekedar pintar main tampilkan fakta angka data analisislah dasar fakta angka data bohong
@mkhumaini dulu sby dipuja puja sekarang pindah ke jokowi, kayaknya dia lagi lapar... 😊😊😊	Dulu sby dipuja puja sekarang pindah jokowi kayaknya dia lapar

<p>kabinet kerja pak jokowi @jokowi k 😊 mpak #salut ibu menteri sri mulyani @kemenkeu dan bapak-bapak menteri @budikaryas @hanifdhakiri @kemenpupr @triawanmunaf kerja serius tapi santai 🙌</p> <p>#kabinetkerjajokowi seru abs 😊</p> <p><a href="https://t.co/q5mgzgvn4q">https://t.co/q5mgzgvn4q</a></p>	<p>kabinet kerja jokowi kmpak salut menteri sri mulyani menteri kerja serius santai kabinetkerjajokowi seru abs</p>
<p>@kompascom @yudiwijayaa awas entar ada komentar 'jokowi pencitraan' 😊 😊 susahnya jadi presiden, maka biarlah jokowi aja lagi</p>	<p>awas komentar jokowi pencitraan susahnya presiden biarlah jokowi</p>
<p>kalau saya bandingkan hasil dua survai poltracking terlihat bahwa elektabilitas presiden @jokowi terus meningkat, tapi elektabilitas prabowo subianto stagnan. dan semakin sedikit rakyat pemilih yang belum menentukan sikap</p> <p><a href="https://t.co/ulzcbp3mng">https://t.co/ulzcbp3mng</a></p>	<p>bandingkan hasil survai poltracking lihat elektabilitas presiden Jokowi tingkat elektabilitas prabowo subianto stagnan rakyat pilih sikap</p>

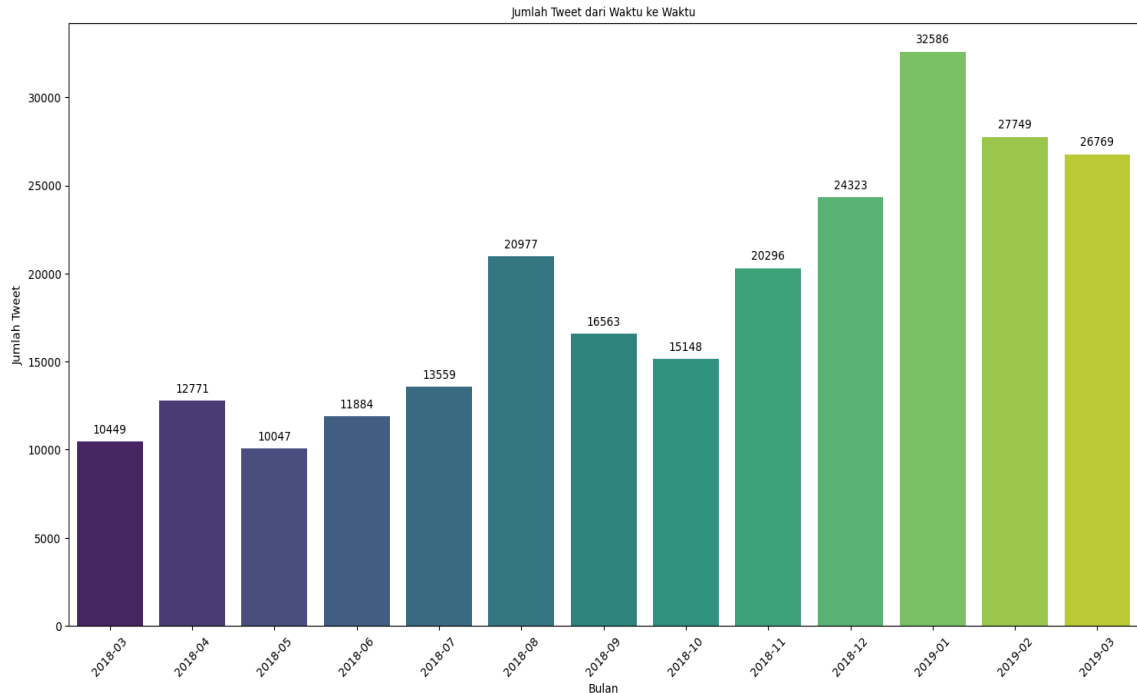
Pada tabel 4.1 diatas terlihat jelas bahwa ada perbedaan pada data tweet sebelum dan setelah dilakukan *preprocessing*. Sebelum dilakukannya tahap *preprocessing* terlihat banyak tweet yang mengandung emoji, kata-kata beruloh, serta tanda baca. Hal itu dihilangkan setelah di *preprocessing*. Proses *case folding* juga terlihat berhasil karena pada kolom tweet yang telah dibersihkan sudah tidak ada lagi kata yang menggunakan huruf kapital. Menghilangkan angka serta karakter yang tidak valid juga telah dilakukan pada tahap *preprocessing*.



**Gambar 4.1** Jumlah Data Sebelum dan Sesudah *Preprocessing*.

Pada gambar 4.1 menunjukkan histogram perbedaan jumlah data *tweet* sebelum dan sesudah *preprocessing*. Terlihat perbedaan jumlah ulasan pada data yang menunjukkan bahwa jumlah ulasan yang muncul pada histogram setelah *preprocessing* jauh lebih sedikit dibanding yang sebelum di *preprocessing*, jumlah tweet tentang jokowi yang awalnya sebanyak 243.121 baris berkurang menjadi 198.913 baris. Hal tersebut terjadi dikarenakan tahap *preprocessing* yang menghapus kata-kata dan komponen lain yang tidak dibutuhkan, dan banyak dari tweet hanya berisi komponen-komponen yang tidak butuhkan sehingga sebagian data tweet dihapus.

Setelah dilakukan *Preprocessing* akan dilakukan *Eksploratory Data Analysis* yang diantaranya yaitu membuat analisis runtun waktu dengan cara mengelompokkan jumlah *tweet* per bulan dari waktu ke waktu. Tujuan dari membuat analisis runtun waktu adalah untuk memahami, memodelkan dan memanfaatkan informasi yang terkandung dalam data data sepanjang waktu.



**Gambar 4.2** Countplot *Frequency Tweet* Bulanan.

Gambar 4.2 adalah grafik persebaran *tweet* setiap bulan, sumbu *x* pada grafik menunjukkan waktu dari bulan Maret 2018 sampai Maret 2019 dan sumbu *y* menunjukkan *frequency tweet* perbulan. Secara umum jika dilihat pada grafik trend yang terlihat pada data yang ada cenderung naik yang mana intensitasnya semakin tinggi dari waktu ke waktu. Hal ini dapat disebabkan karena semakin mendekati waktu pemilihan umum yakni 17 April 2019 maka intensitasnya semakin tinggi. Kemudian pada Gambar 4.2 dapat dilihat juga bahwa *frequency tweet* bulanan paling rendah ada dibulan Mei 2018 yaitu dengan frekuensi sebesar 10.047 *tweet*. Sedangkan untuk *frequency tweet* bulanan paling tinggi ada dibulan Januari 2019 yaitu sebesar 32.586 *tweet*.

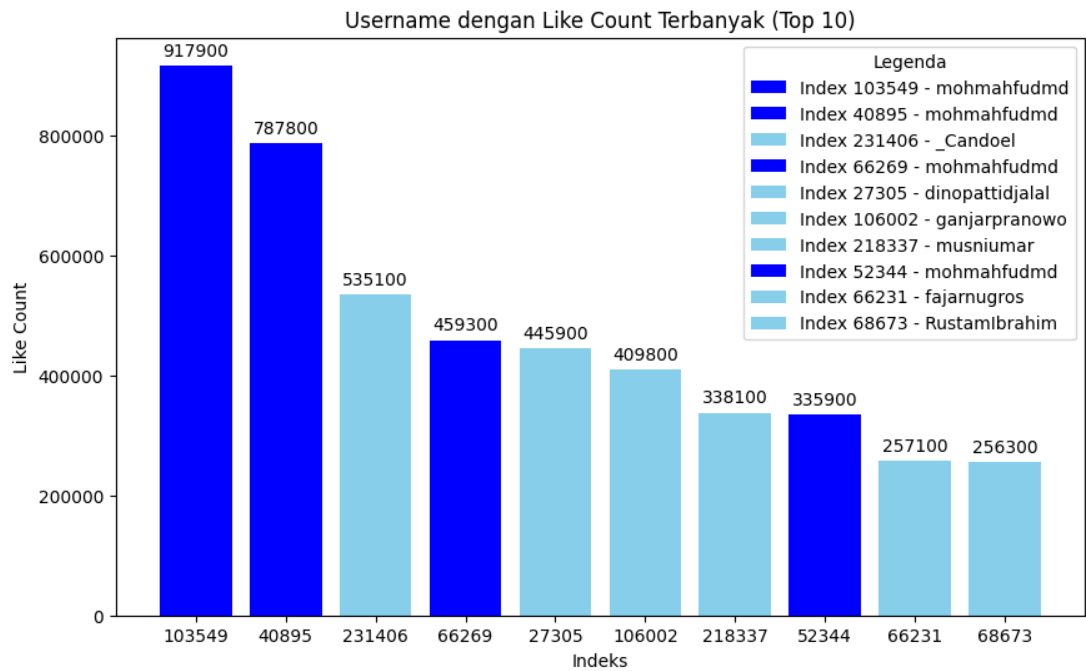
Selanjutnya peneliti akan mencoba memahami narasi yang muncul pada data yang ada. Alat yang digunakan untuk mendapatkan hasil analisis pada tahap ini adalah *wordcloud*. *Wordcloud* adalah suatu metode untuk menampilkan daftar kata yang digunakan dalam sebuah teks, kata dengan ukuran lebih besar yang muncul pada *wordcloud* artinya kata tersebut sering muncul dalam teks.



**Gambar 4.3** *Wordcloud* Dari Data *Tweet*.

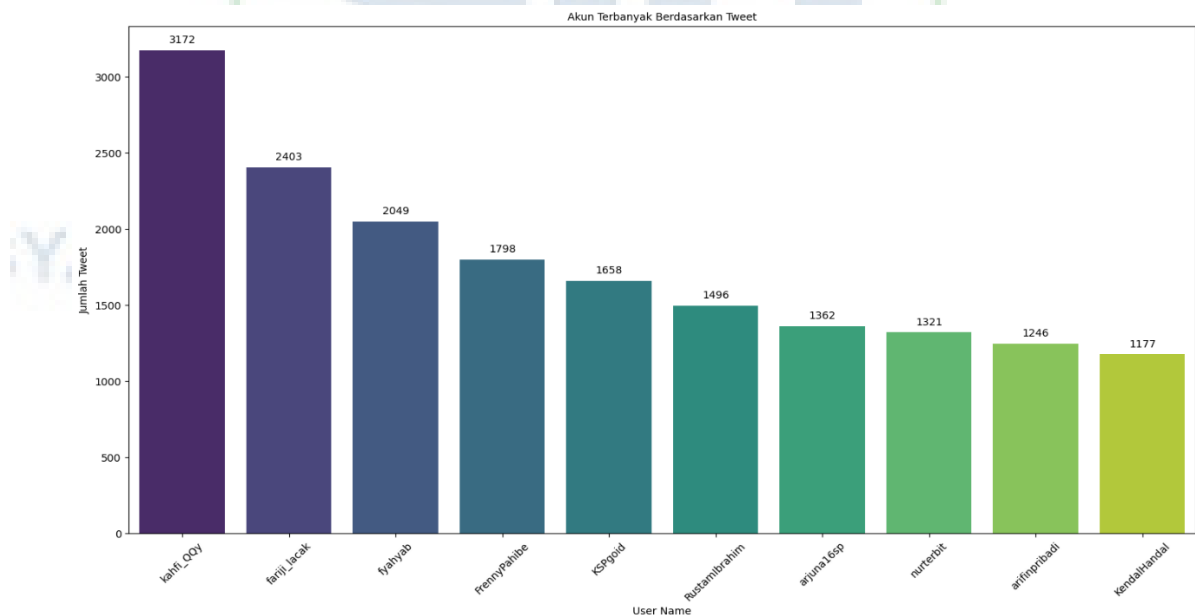
Gambar 4.3 adalah visualisasi teks dalam bentuk *wordcloud*. Kata yang muncul pada *wordcloud* ini adalah kata-kata setahun sebelum pemilihan umum. *Wordcloud* ini diperoleh dengan menggunakan *voyant tools* yang merupakan alat untuk visualisasi data teks. Pada analisis media sosial, banyak literatur yang mengatakan bahwa media sosial tidak efektif di daerah pedesaan yang jauh dari ibukota yang disebabkan internet di daerah tersebut tidak dapat dijangkau. Namun pada Gambar 4.3 *wordcloud* yang menampilkan daftar kata pada teks media sosial yang terkait dengan Jokowi memperlihatkan hasil yang berbeda. Hasilnya menunjukkan bahwa isu terkait desa sering muncul seperti Jokowi membangun desa dan lain sebagainya. Hal ini menjadi penting dibicarakan saat kampanye Jokowi, yang mana keterlibatan desa menjadi penting disini. Selain itu pada Gambar 4.3 banyak muncul kata-kata yang memperlihatkan keberhasilan Jokowi sebagai presiden pada periode pertamanya diantaranya bangun, jalan, ekonomi, program dan lain sebagainya.

Setelah melihat narasi yang muncul pada *wordcloud*, peneliti akan menampilkan akun *twitter* terbanyak berdasarkan jumlah *like* dan juga *tweet*.



**Gambar 4.4** Akun *Twitter* Dengan Jumlah *Like* Terbanyak.

Pada Gambar 4.4 menunjukkan 10 akun *twitter* teratas dengan jumlah *like* terbanyak pada periode Maret 2018 hingga Maret 2019. Sumbu *x* pada grafik menunjukkan posisi *tweet* pada data, sedangkan untuk sumbu *y* menunjukkan *frequency like* pada *tweet*. Pada gambar juga dapat dilihat bahwa untuk *username* mohmahfudmd memiliki *like* terbanyak dari *tweet* yang diunggah olehnya.



**Gambar 4.5** Akun *Twitter* Dengan Jumlah *Tweet* Terbanyak.



Selanjutnya pada Gambar 4.5 menunjukkan 10 akun *twitter* teratas dengan jumlah *tweet* terbanyak pada periode Maret 2018 hingga Maret 2019. Sumbu *x* pada grafik adalah *username* dari yang mengunggah *tweet*, sedangkan untuk sumbu *y* menunjukkan seberapa banyak *tweet* yang diunggah. Pada gambar juga dapat dilihat bahwa untuk *username* kahfi\_QQY paling banyak mengunggah *tweet* pada periode tersebut dengan *keyword* yang ditentukan pada penelitian ini.

## 4.2. Modelling

Pada subbab ini akan dibahas bagaimana model diproses dengan algoritma *BERTopic* dengan fitur tambahan yaitu “*Dynamic topic modelling*” dalam menganalisa perubahan topik dari waktu ke waktu pada data *tweet* dengan keyword “Jokowi”, “Jokowilagi” dan “Jokowiamin” pada bulan maret 2018 hingga bulan maret 2019. Data *tweet* yang telah melewati tahap *preprocessing* akan dimasukkan kedalam algoritma *BERTopic* yang kemudian dilanjutkan dengan fitur “*Dynamic Topic Model*”. Namun, tidak semua data di-*input* kedalam algoritma *BERTopic*, data yang telah di *cleaning* dilakukan pengambilan secara random yaitu 50.000 data yang nantinya akan di-*input* kedalam algoritma *BERTopic*.

Kemudian langkah selanjutnya adalah menjalankan modul *BERTopic*, adapun langkah-langkah serta parameternya sebagai berikut:

1. **`language`**: Ini adalah parameter yang menentukan bahasa yang akan digunakan dalam model. Dalam contoh ini, **`"multilingual"'** digunakan, yang berarti model akan mendukung berbagai bahasa.

2. **`embedding\_model`**: Ini adalah model yang digunakan untuk mengekstraksi representasi *embedding* dari teks. Dalam kasus ini, perlu diberikan model *embedding* yang telah di-training sebelumnya, seperti *BERT* atau model lainnya,

3. **`umap\_model`**: *UMAP (Uniform Manifold Approximation and Projection)* adalah teknik reduksi dimensi yang digunakan untuk memproyeksikan representasi *embedding* yang lebih tinggi ke dimensi yang lebih rendah. Ini membantu dalam visualisasi dan analisis lebih lanjut.

4. ``hdbscan_model``: *HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)* adalah algoritma pengelompokan yang digunakan untuk mengelompokkan representasi *embedding* yang telah direduksi. Ini membantu dalam membentuk klaster topik.

5. ``vectorizer_model``: Ini adalah model yang digunakan untuk mengubah teks menjadi vektor. Vektor ini akan digunakan dalam langkah-langkah selanjutnya untuk analisis topik.

6. ``ctfidf_model``: *C-TF-IDF (Term Frequency-Inverse Document Frequency)* adalah metode yang digunakan untuk mengukur pentingnya suatu kata dalam suatu dokumen relatif terhadap korpus keseluruhan. Model ini digunakan untuk menghitung probabilitas topik berdasarkan istilah dalam teks.

7. ``calculate_probabilities``: Parameter ini menentukan apakah model akan menghitung probabilitas topik atau tidak. Dalam kasus ini, model diatur sebagai ``True``, yang berarti model akan menghitung probabilitas topik.

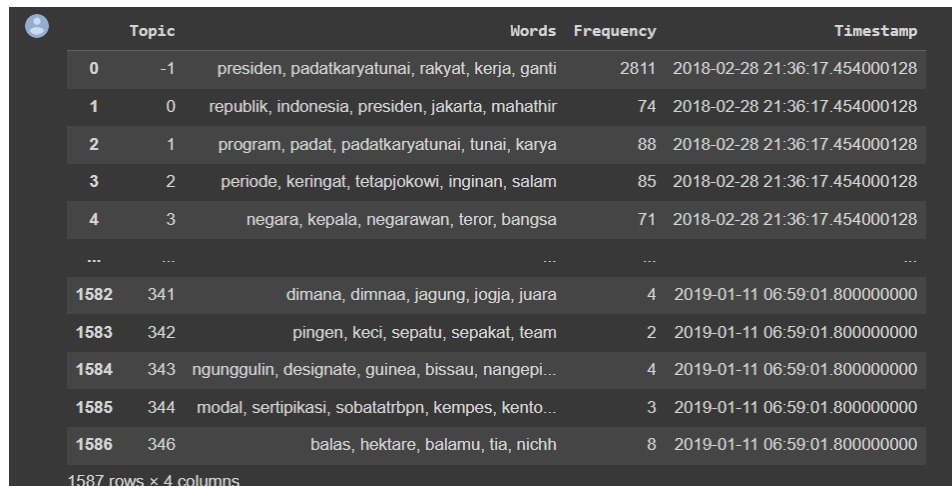
8. ``verbose``: Parameter ini mengontrol apakah keluaran *verbose (detail)* akan ditampilkan selama proses pemodelan atau tidak. Tahapan ini diatur sebagai ``True``, agar bisa melihat lebih banyak informasi selama proses.

Jadi, secara keseluruhan, model *BERTopic* ini bekerja dengan mengambil representasi *embedding* teks dari ``embedding_model``, mereduksi dimensi menggunakan ``umap_model``, mengelompokkan klaster menggunakan ``hdbscan_model``, melakukan tokenisasi topik dengan ``vectorizer_model``, dan menghitung probabilitas topik menggunakan ``ctfidf_model``, dengan opsi untuk menampilkan keluaran detail melalui parameter ``verbose``.

#### 4.3. Analisa Hasil

Pada subbab ini akan dianalisa terhadap kumpulan beberapa kata yang di kelompokkan kedalam beberapa topik serta evolusinya dari waktu dengan menggunakan algoritma *BERTopic* dilanjutkan fitur “*Dynamic Topic Model*” pada data teks yang diambil dari media sosial twitter. Proses algoritma ini menghasilkan 346 topik dari data yang diambil secara acak sebanyak 50.000. Penentuan topik ini berasal dari pengelompokan dokumen, dengan menggunakan *c-Tf-idf* yang mampu

mengekstrak setiap kumpulan dokumen unik. Jika menggunakan *tf-idf* biasa pada dokumen, pada dasarnya yang dilakukan hanya membandingkan kata penting antar dokumen. Jadi kita harus memperlakukan semua dokumen dalam satu kategori. Dengan kata lain mengkluster dokumen sebagai satu dokumen dan menerapkan *tf-idf*, hasilnya dokumen akan sangat panjang per kategori dan skor *tf idf* yang dihasilkan akan menunjukkan kata kata penting dalam satu topik.



Topic	Topic	Words	Frequency	Timestamp
0	-1	presiden, padatkaryatunai, rakyat, kerja, ganti	2811	2018-02-28 21:36:17.454000128
1	0	republik, indonesia, presiden, jakarta, mahathir	74	2018-02-28 21:36:17.454000128
2	1	program, padat, padatkaryatunai, tunai, karya	88	2018-02-28 21:36:17.454000128
3	2	periode, keringat, tetapjokowi, inginan, salam	85	2018-02-28 21:36:17.454000128
4	3	negara, kepala, negarawan, teror, bangsa	71	2018-02-28 21:36:17.454000128
...	...	...	...	...
1582	341	dimana, dimnaa, jagung, jogja, juara	4	2019-01-11 06:59:01.800000000
1583	342	pingen, keci, sepatu, sepatat, team	2	2019-01-11 06:59:01.800000000
1584	343	ngunggulin, designate, guinea, bissau, nangepi...	4	2019-01-11 06:59:01.800000000
1585	344	modal, sertifikasi, sobatatrbtn, kempes, kento...	3	2019-01-11 06:59:01.800000000
1586	346	balas, hektare, balamu, tia, nichh	8	2019-01-11 06:59:01.800000000

**Gambar 4.6** Banyaknya Topik yang Dihasilkan.

Pada Gambar 4.6 menunjukkan *output* dari algoritma *BERTopic* yang menghasilkan topik apa saja yang muncul dengan frekuensinya beserta dengan evolusi topiknya. Adapun parameter yang digunakan untuk melihat topik tersebut berevolusi di setiap waktunya yaitu, *global\_tuning* dan *evolutionary\_tuning* yang disetel ke *True* sebagai *default* dengan tujuan representasi yang dihasilkan akan dipengaruhi oleh topik global dan berevolusi dari waktu ke waktu. Parameter kedua adalah *bins*, jika kita memiliki lebih dari 100 *iterasi*, maka akan ada representasi topik yang dibuat untuk masing-masing stempel waktu tersebut yang dapat mempengaruhi representasi topik secara negatif. Maka dari itu disarankan untuk melakukan *iterasi* di bawah 50. Untuk melakukan ini, kita cukup mengatur jumlah *iterasi* yang dibuat saat menghitung representasi topik.

Selanjutnya akan diperlihatkan bagaimana performa yang dihasilkan dari algoritma *BERTopic* dalam mengekstrak topik pada data penelitian ini. Performa dari algoritma *BERTopic* yang akan dievaluasi adalah dengan melihat seberapa

cepat algoritma tersebut menyelesaikan tugasnya dan seberapa besar “*Coherence Score*” yang dihasilkan oleh algoritma tersebut.

```
# Evaluate
coherence_model = CoherenceModel(topics=topic_words,
                                  texts=tokens,
                                  corpus=corpus,
                                  dictionary=dictionary,
                                  coherence='c_v')

coherence = coherence_model.get_coherence()

[ ] print(coherence)

0.7116116960883678

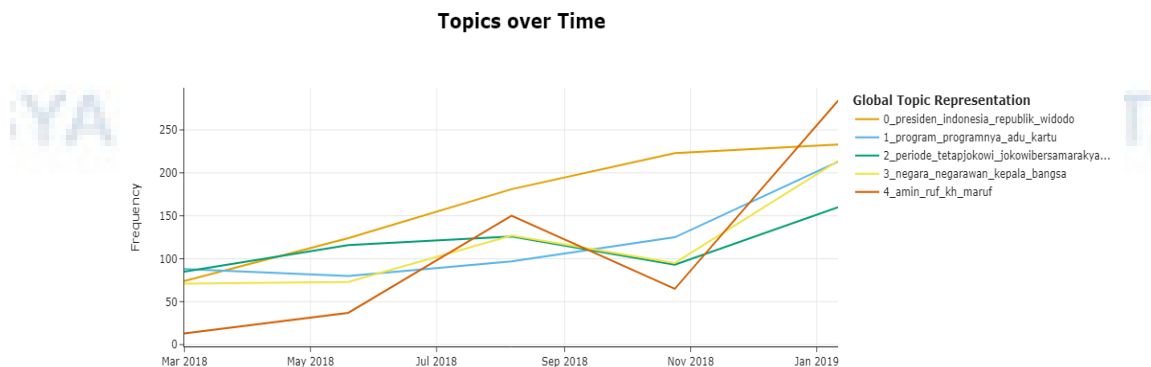
[ ] stop = timeit.default_timer()
print('Time', stop - start, 'seconds')

Time 3144.132465046 seconds
```

**Gambar 4.7** Evaluasi Model

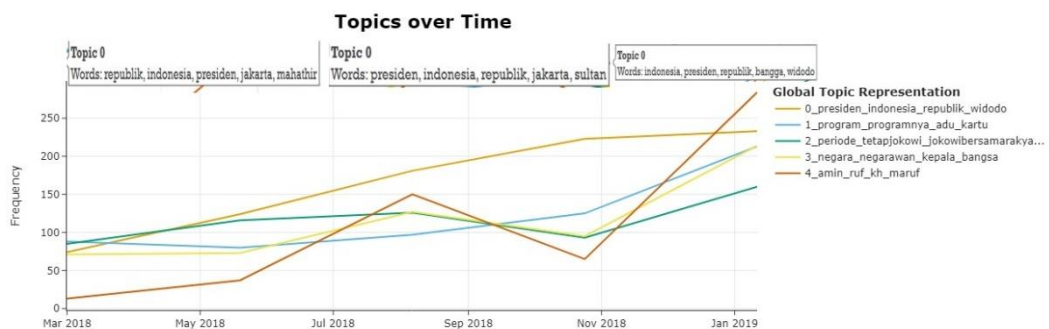
Hasil perolehan dari *running time* yang didapatkan algoritma *BERTopic*, menghasilkan *running time* selama 52.4 menit. Kemudian algoritma *BERTopic* berhasil mendapatkan *coherence score* yaitu 0,71 yang artinya kumpulan kata-kata dalam topik yang dihasilkan oleh algoritma *BERTopic* memiliki keterkaitan satu sama lain.

Kemudian topik-topik yang dihasilkan beserta evolusinya akan di visualisasikan serta di analisa secara lebih jelas sebagai berikut:



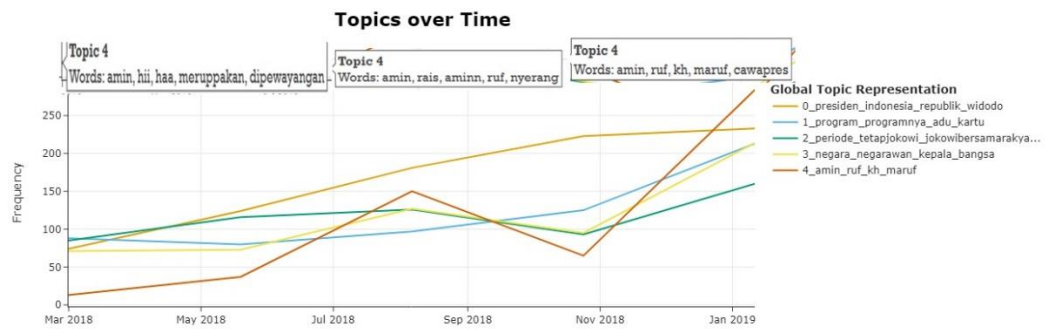
**Gambar 4.8** Topik yang Dihasilkan dari Model

Dari model yang sudah di proses, terdapat 5 topik yang akan dianalisa beserta evolusinya dari waktu ke waktu. Dari grafik tersebut dapat dilihat bahwa topik ketiga dan ke empat memiliki frekuensi yang sama pada periode tersebut yaitu Agustus 2018 hingga November 2018.. Kemudian untuk semua topik menjelang pemilu 2019 cenderung naik walaupun sebelumnya mengalami fluktuatif dari bulan Maret 2018 hingga bulan November 2018. Hal ini disebabkan karena pemilu 2019 akan segera diselenggarakan yang menyebabkan pembicaraan seputar politik akan ramai dibahas oleh masyarakat Indonesia.



**Gambar 4.9** Evolusi Topik Pertama

Dari Gambar 4.9 bisa dilihat bahwa untuk topik pertama konsisten naik dari bulan Maret 2018 hingga November 2018. Topik yang muncul berkaitan dengan beberapa kata antara lain Presiden, Indonesia, Republik dan Widodo. Pembahasan pada topik ini diawali dengan bertemunya Presiden Jokowi dengan perdana menteri Malaysia yaitu Dr. Mahathir Muhammad. Kemudian topik tersebut berevolusi pada bulan Juni 2018 yaitu bertemunya Presiden Jokowi dengan Sultan Brunei. Lalu pada bulan Agustus 2018 Topik pertama tidak lagi membicarakan tentang pertemuan Presiden Jokowi dengan negara tetangga melainkan topik ini berubah menjadi bangga terhadap Presiden Jokowi dan topik ini terus menaik dibicarakan hingga menjelang Pemilu 2019.



**Gambar 4.10** Evolusi Topik Kelima

Dari Gambar 4.10 terdapat *insight* yang menarik yaitu pada bulan november 2018 hingga Januari 2019 topik kelima membahas tentang calon wakil presiden yaitu K.H Maaruf Amin padahal pada pembukaan topik kelima yaitu bulan Maret 2018 topik ini membicarakan Amin Rais tentang mafia tanah. Namun topik kelima ini mengalami cukup banyak penurunan diantara topik-topik yang lain. Tetapi kemudian pada bulan November 2018 topik ini kembali *populer* lagi dikarenakan diumumkannya Calon Wakil Presiden yaitu K.H Maaruf Amin. Topik ini menjadi topik teratas diantara topik-topik yang lain hingga Januari 2019.

## BAB V

### KESIMPULAN DAN SARAN

Pada bab ini berisi tentang beberapa kesimpulan yang telah dihasilkan dari penelitian yang telah dilakukan serta saran yang bisa digunakan untuk penelitian yang akan datang.

#### 5.1 Kesimpulan

Terdapat beberapa langkah untuk mengimplementasikan *Dynamic Topic Modelling* menggunakan algoritma *BERTopic* yang pertama mempersiapkan data terstruktur dengan informasi waktu, misalnya kolom berisi *timestamp* untuk setiap dokumen. Kedua, melakukan *preprocessing* untuk mempermudah proses modelling. Ketiga, menjalankan algoritma *BERTopic* diantaranya *SBERT*, *UMAP*, *HDBSCAN* dan melakukan *Dynamic Topic Modelling* dengan cara menghitung nilai *C-TF-IDF* untuk menemukan evolusi topiknya dari waktu ke waktu. Terakhir, memvisualisasikan model dan diberikan interpretasinya.

Dari segi performa, algoritma *BERTopic* mendapatkan *coherence score* yang dihasilkan diangka 0.711 dengan 346 topik dari 50.000 data *sample*. Sedangkan untuk kecepatan, *runningtime* yang dihasilkan adalah 52.4 menit. Hasil tersebut tampaknya sedikit lebih lama dari algoritma pemodelan topik lainnya.

Kemudian dari analisa yang didapat menggunakan algoritma *BERTopic* terdapat *top 5 global topic representation* yang dapat di interpretasikan dengan evolusinya dari waktu ke waktu. Dari segi frekuensi yang digambarkan pada grafik, semua topik pada bulan november 2018 mengalami kenaikan walaupun pada bulan sebelumnya mengalami fluktuatif. Hal ini disebabkan karena pemilu 2019 akan segera diselenggarakan yang menyebabkan pembicaraan seputar politik akan ramai dibahas oleh masyarakat Indonesia. Kemudian untuk contoh evolusinya bisa dilihat pada topik kelima pada akhir periode membahas tentang calon wakil presiden yaitu K.H Maaruf Amin padahal sebelumnya pada pembukaan topik kelima yaitu bulan Maret 2018 topik ini membicarakan Amin Rais tentang mafia tanah.



## 5.1 Saran

Berdasarkan kesimpulan dari penelitian yang telah dilakukan diperoleh beberapa saran untuk penelitian serupa di masa yang akan datang diantaranya :

1. Penggunaan arsitektur *neural network* pada algoritma pemodelan topik membuat waktu yang dibutuhkan untuk memproses data menjadi lebih lama. Jika dibandingkan dengan algoritma pemodelan topik lainnya seperti *LDA*, *LSA* dan Sebagainya. Dari hal tersebut disarankan untuk penelitian berikutnya agar menemukan cara bagaimana algoritma *BERTopic* dan juga algoritma pemodelan topik yang menggunakan *neural network* lainnya dapat lebih efisien dalam segi waktu pemrosesan
2. Dalam penelitian ini algoritma pemodelan topik yang digunakan adalah *BERTopic* dimana parameter yang dimiliki sangat banyak dan beragam sehingga bisa disesuaikan untuk *dataset* yang ada, namun parameter yang digunakan pada penelitian ini lebih banyak menggunakan parameter *default*. Maka dari itu disarankan bagi penelitian selanjutnya untuk memperdalam setiap parameter yang digunakan dalam *BERTopic* untuk mendapatkan *output* topik yang terbaik.
3. Dalam penelitian ini bagian *topic coherence* belum dilakukan secara *temporal* maka dari itu peneliti menyarankan untuk penelitian serupa berikutnya agar menganalisis *coherence score* menggunakan aspek temporal. Untuk rumus *temporal topic coherence* terdapat beberapa perbedaan dengan *topic coherence* biasa yaitu terletak pada  $L$  dan juga  $t$ . Karena pada *temporal topic coherence* terdapat kumpulan kata yang berevolusi. Sedangkan untuk  $L$ -nya menunjukkan jendela waktu [41].
4. Terakhir, Peneliti menyarankan untuk penelitian serupa berikutnya agar bisa menganalisis pemilu berikutnya pada tahun 2024 dengan menggunakan data yang berasal dari media sosial lain seperti *youtube* dan *facebook* atau menggabungkan serta membandingkan hasil analisis antar media sosial untuk mendapatkan hasil yang lebih *insightful*.



## DAFTAR PUSTAKA

- [1] kominfo, "KOMINFO Temukan 3.356 Hoaks , Terbanyak saat pemilu 2019," 2019. [Online]. Available: [https://www.kominfo.go.id/content/detail/21876/kominfo-temukan-3356-hoaks-terbanyak-saat-pemilu-2019/0/berita\\_satker](https://www.kominfo.go.id/content/detail/21876/kominfo-temukan-3356-hoaks-terbanyak-saat-pemilu-2019/0/berita_satker).
- [2] W. P. Review, "World Population by Country," [Online]. Available: <https://worldpopulationreview.com/>.
- [3] A. Link, "Hootsuite (We are Social): Indonesian Digital Report 2023," Hootsuite, 2023. [Online]. Available: <https://andi.link/hootsuite-we-are-social-indonesian-digital-report-2023/>.
- [4] F. B. Hermawan, "Analisis Eksplorasi Data Pada Kampanye Kandidat Pemilihan Umum Presiden Indonesia Tahun 2019 di Media Sosial Twitter," 23 Juni 2023. [Online]. Available: <https://repository.uinjkt.ac.id/dspace/handle/123456789/71759>.
- [5] L. Lefebure, "Exploring the UN General Debates with Dynamic Topic Models," 17 oktober 2018.
- [6] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," p. 10, 2022.
- [7] D. M. Blei and J. D. Lafferty, "Dynamic Topic Models," in *Proceedings of The 23rd International Conference on Machine Learning*, 2006, pp. 113-120.
- [8] R. Feldman and J. Sanger, *The Text Mining Handbook: Adavanced Approaches in Analyzing Unstructured Data*, New York: Cambridge University Press, 2007.

- [9] S. Dang and P. H. Ahmad, "Text Mining: Techniques and Its Application," *International Journal of Engineering & Technology Innovations*, vol. 1, 2014.
- [10] R. Mitchell, "Web Scraping with Python," in *O'Reilly Media*, Sebastopol, 2015.
- [11] T. Aksoy, S. Celik and S. Gulsecen, "Data Pre-processing in Text Mining," in *Who Runs The World: Data*, Istanbul, Istanbul University Press, 2020, pp. 123-144.
- [12] Salsabila, "Begini Cara Implementasi Teknik Analisis untuk Text Preprocessing," DQ Lab, 9 May 2022. [Online]. Available: <https://dqlab.id/begini-cara-implementasi-teknik-analisis-data-untuk-text-preprocessing>.
- [13] R. Tineges, "Tahapan Text Preprocessing dalam Teknik Pengolahan Data," DQLab, 17 06 2021. [Online]. Available: <https://dqlab.id/tahapan-text-preprocessing-dalam-teknik-pengolahan-data>. [Accessed 20 05 2023].
- [14] S. Sarica and J. Luo, "Stopwords in Technical Language Processing," *PLoS ONE*, 05 08 2021.
- [15] D. M. Blei, A. Y. Ng and M. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 601-608, 2001.
- [16] S. M. Rezkia, "Mengenal Lebih Dalam Algoritma Unsupervised Learning," DQLab, 14 desember 2020. [Online]. Available: <https://dqlab.id/mengenal-leboh-dalam-algoritma-unsupervised-learning>.
- [17] J. D. Google, M.-W. Chang, K. Lee and K. N. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *NAACL 2019*, 2018.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is All You Need," 12 June 2017.

- [Online]. Available: <https://arxiv.org/abs/1706.03762>. [Accessed 12 May 2023].
- [19] "Encoder dan Decoder," Perpustakaan Multi Media Nusantara, [Online]. Available: [https://kc.umn.ac.id/13998/4/BAB\\_II.pdf](https://kc.umn.ac.id/13998/4/BAB_II.pdf).
- [20] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Computer Science, Computer and Language*, 27 08 2019.
- [21] L. Afifah, "2 Teknik Reduksi Dimensi Populer dengan Python," IlmuDataPy, [Online]. Available: <https://ilmudatapy.com/teknik-reduksi-dimensi/>.
- [22] M. Grootendorst, "BERTopic: Neural Topic Modeling with a Class-based TF-IDF Procedure," 11 March 2022. [Online]. Available: <https://arxiv.org/abs/2203.05794>. [Accessed 14 03 2023].
- [23] P. O. Box, L. V. D. Maaten, E. Postma and J. V. D. Herik, "Tilburg centre for Creative Computing Dimensionality Reduction: A Comparative Review," 2009. [Online]. Available: <http://www.uvt.nl/ticc>.
- [24] L. McInnes and J. Healy, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," 02 2018.
- [25] P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, *Introduction to Data Mining*, New York: Pearson, 2019.
- [26] R. J. G. B. Campello, D. Moulavi and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in *Advances in Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, 2013, pp. 166-172.
- [27] T. Joachims, "A Probabilistic Analysis of The Rocchio Relevance Feedback Algorithm," 1996.

- [28] H. Wallach, I. Murray and D. Mimno, "Evaluation Methods for Topic Models," in *International Conference on Machine Learning*, 2009.
- [29] D. Mimno, H. Wallach, A. McCallum, A. Talley and T. Leenders, "Optimizing Semantic Coherence in Topic Models," in *Conference on Empirical Methods in Natural Language Processing*, 2011.
- [30] M. Roder, A. Both and A. Hinneburg, "Exploring The Space of Topic Coherence Measures," in *WSDM 2015 - Proceedings of The 8th ACM International Conference on Web Search and Data Mining*, 2015.
- [31] E. R. J. I. Yoonjoo Ahn, "Dual Embedding With Input Embedding and Output Embedding For Better Word Representation," *Indonesian Journal Of Electrical Engineering and Computer Science*, p. 9, 2022.
- [32] J. Briggs, "Sentence Transformers : Meanings in Disguise," Pinecone, 2017. [Online]. Available: <https://www.pinecone.io/learn/series/nlp/sentence-embeddings/>.
- [33] K. B. e. al, "Exploring Alternatives to Softmax Function," 2020.
- [34] A. S. Shirkhorshidi, "A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data," 2015.
- [35] jlmelville, "Topological Data Analysis and Simplicial Complexes," Leland McInnes Revision, 2018. [Online]. Available: [https://umap-learn.readthedocs.io/en/latest/how\\_umap\\_works.html](https://umap-learn.readthedocs.io/en/latest/how_umap_works.html).
- [36] N. Oskolkov, "How Exactly UMAP Works and Why Exactly is Better than tSNE," 2019.
- [37] "UMAP projections of a 3D woolly mammoth skeleton," [Online]. Available: [https://www.researchgate.net/figure/UMAP-projections-of-a-3D-woolly-mammoth-skeleton-10\\_fig1\\_368664623](https://www.researchgate.net/figure/UMAP-projections-of-a-3D-woolly-mammoth-skeleton-10_fig1_368664623).

- [38] "How HDBSCAN Works," [Online]. Available:  
[https://hdbscan.readthedocs.io/en/latest/how\\_hdbscan\\_works.html](https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html).
- [39] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," 1996.  
[Online]. Available: <http://www.aaai.org/>.
- [40] Maarten, "Maartengr," 2023. [Online]. Available:  
[https://maartengr.github.io/BERTopic/getting\\_started/topicsovertime/topicsovertime.html](https://maartengr.github.io/BERTopic/getting_started/topicsovertime/topicsovertime.html). [Accessed 12 June 2023].
- [41] J. Vang, "Coherence Scores," One-Off Coder, 2019. [Online]. Available:  
<https://datascience.oneoffcoder.com/topic-modeling-gensim.html>.
- [42] C. M. Suci, "Analisis Penyelenggaraan Pemilihan Umum Serentak Thun 2019 Terhadap Nilai-Nilai Demokrasi di Indonesia," 2019.
- [43] F. Gao, B. Li, L. Chen, Z. Shang, X. Wei and C. He, "A Softmax Classifier for Hogh Precision Classification of Ultrasonic Similar Signal," *Ultrasonics*, vol. 112, 2021.
- [44] T. Kitasuka, M. Arisugi and F. Rahutomo, "Semantic Cosine Similarity," 2012.
- [45] P. Berba, "pberba," 17 Jan 2020. [Online]. Available:  
<https://pberba.github.io/stats/2020/01/17/hdbscan/>. [Accessed 09 July 2023].
- [46] D. Sanjaya, "Pemanfaatan Media Sosial Dalam Iklan Kampanye Politik Pemilu 2019 Bagian 1," ELSAM Multimedia, 07 02 2020. [Online]. Available: <https://multimedia.elsam.or.id/pemanfaatan-media-sosial-dalam-iklan-kampanye-politik-pemilu-2019-bagian-i/>. [Accessed 10 05 2023].
- [47] N. Wibisono, "Pemanfaatan Media Sosial," p. 47, 2015.

- [48] I. H. Harahap, "Kampanye Pilpres 2019 Melalui Media Sosial dan Pengaruhnya Terhadap Demokrasi Indonesia," *Komunikologi* , 2020. [Online]. Available: <https://komunikologi.esaunggul.ac.id/index.php/KM/article/view/234>.
- [49] "OPEN DATA KPU," [Online]. Available: <https://opendata.kpu.go.id/dataset/7380cce74-9197c30ea-3618b8927-a9d24>.

