JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id

ANALISA DAN VISUALISASI HASIL KUESIONER PERTANYAAN TERBUKA MENGGUNAKAN ELASTICSEARCH DAN KIBANA

Muchlis Polin[1], Nikmasari Pakaya[2], Budiyanto Ahaliki[3]
[1,2,3] Program Studi Sistem Informasi, Fakultas Teknik, Universitas Negeri Gorontalo
e-mail: mpolin@ung.ac.id, nikmasari.pakaya@ung.ac.id, budiyanto@ung.ac.id

Abstrak

Penggunaan kuesioner online untuk pengumpulan data saat ini semakin lazim. Secara umum, pertanyaan dalam kuesioner bisa dibagi menjadi pertanyaan terbuka ataupun pertanyaan tertutup. Jenis data yang dihasilkan oleh kuesioner pun bisa bermacam-macam, sehingga teknik analisis yang digunakan harus disesuaikan dengan data yang ada. Permasalahan yang ditemui adalah pada sulitnya analisis jawaban pertanyaan terbuka berupa teks, terutama untuk kuesioner dengan jumlah responden yang banyak. Data teks ini perlu di analisis dan dipresentasikan sedemikian rupa sehingga topik-topik ataupun temanya bisa dieksplorasi dengan mudah. Penelitian ini bertujuan untuk mengembangkan aplikasi analisis dan visualisasi data teks pada jawaban kuesioner pertanyaan terbuka. Dengan demikian, hasil kuesioner yang telah dikumpulkan dapat dianalisis lebih cepat serta divisualisasikan dengan mudah. Untuk itu dibuatlah sebuah sistem berdasarkan framework Capture Understand, and Present (CUP) dengan menggunakan bahasa pemrograman Python. Pada tahapan Capture, data harus melewati proses preprocessing dimana terjadi pembersihan dan pemrosesan awal data. Pada tahapan Understand, data yang telah dibersihkan akan dianalisis menggunakan sebuah algoritma topic modelling, yaitu GSDMM. Pada tahapan Present, hasil analisis kemudian disimpan menggunakan Elasticsearch dan divisualisasikan menggunakan dashboard Kibana. Visualisasi yang dibuat bersifat interaktif, sehingga memudahkan pengguna dalam mengeksplorasi hasil analisis data.

Kata kunci: *Machine Learning, GSDMM, Visualisasi, Elasticsearch, Kibana*

Abstract

The use of online questionnaires for data collection is now ubiquitous. In general, questionnaire questions can be categorized into open-ended questions and closed-ended questions. The types of data generated by the questionnaire can also vary. Hence, the appropriate analytical techniques must be used according to the type of data. The problem encountered is that analysing responses to open-ended questions in the form of text can be difficult, especially for questionnaires with a large number of respondents. This text data needs to be analyzed and presented in such a way that the topics or themes can be explored easily. This study aims to develop a system for analysing and visualizing text data from open-ended questionnaire. So that the collected data can be analyzed quickly and easily visualized. For this reason, a system based on the Capture Understand Present (CUP) framework was created using the Python programming language. First, at the Capture stage, the data must go through a preprocessing stage where initial processing of the data is performed. Second, at the Understand stage, the cleaned data will be analyzed using a topic modeling algorithm, namely GSDMM. Finally, the analysis results are then stored using Elasticsearch and visualized using the Kibana dashboard. The visualization made is interactive, making it easy for users to explore the results of data analysis.

Keywords*: Machine Learning, GSDMM, Visualization, Elasticsearch, Kibana*

1. PENDAHULUAN

Kuesioner online semakin banyak diminati dan digunakan untuk kepentingan pengumpulan data oleh berbagai pihak karena memiliki beberapa keunggulan dibandingkan dengan kuesioner manual [1]. Bagi para peneliti, kuesioner online mudah disebarkan dan tidak membutuhkan biaya banyak atau bahkan bisa gratis jika menggunakan layanan tertentu [2]. Selain itu, data responden bisa langsung terisi kedalam komputer tanpa perlu lagi diinput. Dari sudut pandang responden, kuesioner bisa

diisi kapan saja dan dimana saja selama ada akses terhadap internet dan perangkat smartphone ataupun komputer.

Untuk pertanyaan yang memiliki desain skala Likert, ataupun pertanyaan tertutup lainnya, analisis kuantitatif dapat dilakukan menggunakan software *spreadsheet* seperti Microsoft Excel, ataupun software analisis statistika seperti SPSS [3]. Hasil analisis jawaban pertanyaan tertutup ini juga dapat divisualisasikan dengan mudah. Sebagai contoh, pada Google Forms, grafik hasil analisis untuk setiap pertanyaan akan dibuatkan secara otomatis oleh sistem. Meskipun visualisasi ataupun grafiknya tidak bisa dikustomisasi, tetapi untuk kuesioner yang sederhana ataupun kebutuhan yang tidak terlalu kompleks, visualisasi yang disediakan sudah bisa mencukupi. Platform lainnya seperti SurveyMonkey menyediakan kemampuan analisis dan visualisasi yang lebih lengkap [3], namun terkadang pengguna harus membayar untuk mendapatkan fasilitas-fasilitas ini.

Tidak seperti pertanyaan tertutup, analisis pertanyaan terbuka membutuhkan teknik yang berbeda karena jawaban yang diberikan responden adalah berupa teks bebas yang bersifat kualitatif [3]. Analisis yang dilakukan dapat berupa *coding* atau penyortiran data kedalam tema-tema yang dapat diinterpretasi.

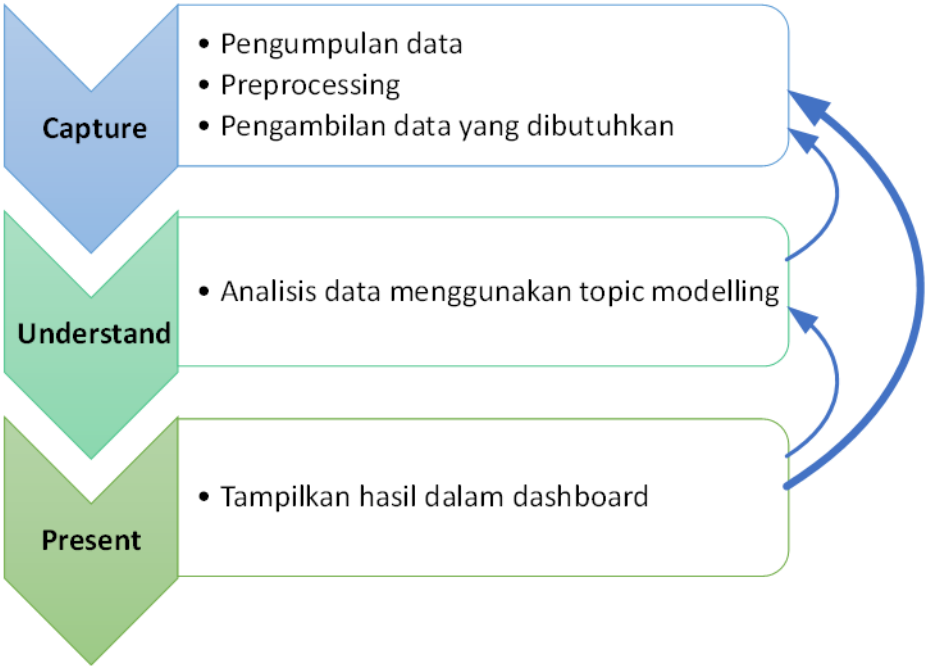
Unit Penjaminan Mutu (UPM) pada salah satu fakultas di kampus XYZ rutin menyelenggarakan berbagai survei sebagai salah satu bentuk evaluasi kegiatan tridharma perguruan tinggi. Survei-survei ini dilaksanakan secara online untuk mempermudah pengumpulan dan analisis data. Kuesioner yang disebar pada umumnya memiliki dua bentuk pertanyaan, yaitu pertanyaan dalam skala Likert dan pertanyaan terbuka. Pertanyaan terbuka sering digunakan oleh UPM fakultas untuk meminta kritik dan saran dari sivitas akademika maupun para *stakeholder*.

Permasalahan yang ditemukan dilapangan adalah banyaknya kuesioner yang disebar dan jumlah respon yang diterima menyebabkan analisis jawaban pertanyaan terbuka sulit untuk dilakukan secara manual. Hal ini karena bentuk jawabannya yang berupa teks bebas. Dengan demikian, penelitian ini bertujuan untuk mengembangkan sebuah sistem analisis dan visualisasi data teks yang berasal dari jawaban pertanyaan terbuka. Sistem ini diharapkan akan mempermudah analisis dan eksplorasi jawaban pertanyaan terbuka.

2. METODE

Metode yang digunakan pada penelitian ini adalah metode *Capture, Understand and Present (CUP)* yang dikemukakan oleh [4]. Framework CUP didesain untuk proses analisis data sosial media, namun bisa diadaptasi untuk kebutuhan penelitian ini. Secara garis besar, metode CUP terdiri dari tiga tahapan, yaitu tahapan pengumpulan data (*capture*), kemudian diikuti oleh tahapan analisis data (*understand*) dan yang terakhir adalah tahapan presentasi hasil analisis data (*present*). Metode CUP yang digunakan dalam penelitian ini ditampilkan pada Gambar 1.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id



Gambar 1. Tahapan penelitian (diadaptasi dari [4])

2.1. Tahapan Pengumpulan Data (Capture)

Pada tahapan ini, dilakukan pengumpulan data dari sumber yang dibutuhkan untuk kemudian diproses melalui langkah *preprocessing*, dimana data akan diproses dan dibersihkan. Tahapan ini termasuk pemilahan data yang diperlukan dan yang tidak diperlukan. Data-data yang tidak diperlukan tidak akan diproses lebih lanjut. Hasil dari tahapan ini adalah data yang telah dibersihkan dan diproses sehingga layak untuk dianalisis pada tahapan berikutnya.

2.2. Tahapan Analisis Data (Understand)

Pada tahapan *understand* dilakukan berbagai jenis analisis sesuai dengan jenis data yang ada serta tujuan analisis yang ingin dicapai. Beberapa teknik yang umum digunakan pada tahapan ini adalah *sentiment analysis*, *trend analysis*, *topic modelling* dan *social network analysis* [4]. Pada penelitian ini teknik yang digunakan pada tahapan analisis adalah *topic modelling*. Algoritma *topic modelling* digunakan untuk menemukan topik-topik yang terdapat dalam dataset dengan cara mengelompokkan teks berdasarkan kemiripan, sehingga respon kuesioner akan bisa dikelompokkan berdasarkan tema ataupun topik.

Beberapa contoh algoritma *topic modelling* yang umum digunakan adalah; *Latent Dirichlet Allocation* (LDA) [5] *Non-negative Matrix Factorization* (NMF) [6] dan *Latent Semantic Analysis* (LSA) [7]. Algoritma-algoritma ini didesain untuk mendeteksi topik pada teks yang panjang seperti pada sebuah buku, artikel ataupun blog post. Sedangkan teks yang tersedia pada jawaban kuesioner adalah teks pendek. Sehingga, penggunaan LDA, LSA dan NMF dinilai kurang tepat untuk kasus penelitian ini.

Untuk teks yang pendek seperti yang ada pada Twitter atau teks pendek lainnya, maka algoritma yang lebih tepat digunakan adalah *Gibbs Sampling Dirichlet Multinomial Mixture* (GSDMM) yang dirancang oleh [8]. Perbedaan GSDMM dan algoritma *topic modeling* lain adalah GSDMM mengasumsikan bahwa hanya ada satu topik dalam sebuah

teks, sedangkan algoritma lainnya seperti LDA mengasumsikan adanya lebih dari satu topik dengan proporsi yang berbeda-beda pada sebuah teks. GSDMM memiliki performa yang lebih baik dibandingkan LDA untuk analisis teks pendek [9]. Pada penelitian ini, teks yang dianalisis hanyalah teks pendek berisi saran, sehingga penggunaan GSDMM dinilai lebih tepat untuk penelitian ini.

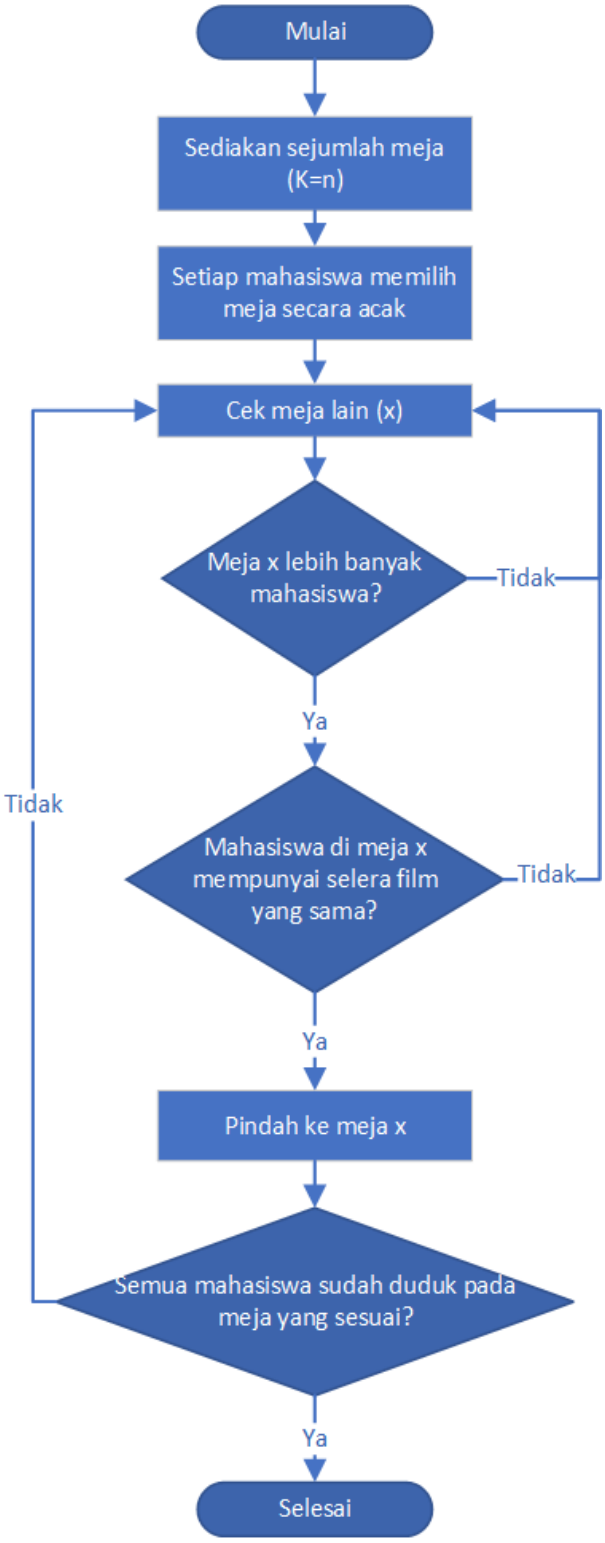
Proses dalam GSDMM bisa dianalogikan dalam sebuah proses yang disebut dengan *Movie Group Process* (MGP) [8]. Secara sederhana, penjelasan dari MGP bisa dianalogikan sebagai berikut. Seorang dosen ingin menugaskan kepada mahasiswanya untuk mendiskusikan film kesukaan mereka. Untuk itu, mahasiswa diminta menuliskan daftar film yang pernah mereka tonton dalam waktu beberapa menit saja agar film yang ditulis adalah film yang baru saja ditonton ataupun film kesukaan mereka. Karena waktu terbatas, daftar film yang ditulis tidak akan menjadi panjang. Dosen tersebut ingin membagi mahasiswanya menjadi beberapa kelompok, dimana setiap kelompok berisi mahasiswa dengan selera film yang sama. Dengan demikian, setiap kelompok akan mewakili topik atau selera film tertentu. Dosen tadi kemudian mengajak mahasiswanya kedalam sebuah restoran dan menyediakan sejumlah meja.

Pada awalnya, mahasiswa akan diminta untuk duduk secara acak, kemudian setelah semua mendapatkan tempat duduk, mahasiswa diminta untuk memilih meja baru, dengan dua peraturan sebagai berikut:

1. Peraturan pertama: Pilih meja baru dengan jumlah mahasiswa yang lebih banyak dari yang sekarang ditempati.
2. Peraturan kedua: Pilih meja dimana temannya yang duduk disitu memiliki ketertarikan film yang sama (pernah menonton lebih banyak film yang sama dibandingkan teman-teman pada meja yang lain).

Seiring berjalannya waktu, meja-meja tertentu akan menjadi lebih banyak mahasiswanya, sedangkan meja lainnya bisa jadi akan kosong. Sehingga pada akhirnya, masing-masing meja akan ditempati oleh mahasiswa-mahasiswa yang memiliki ketertarikan terhadap film (topik) yang sama. Flowchart dari MGP ditampilkan pada Gambar 2.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id



Gambar 2. Flowchart Movie Group Process

2.3. Tahapan Presentasi (Present)

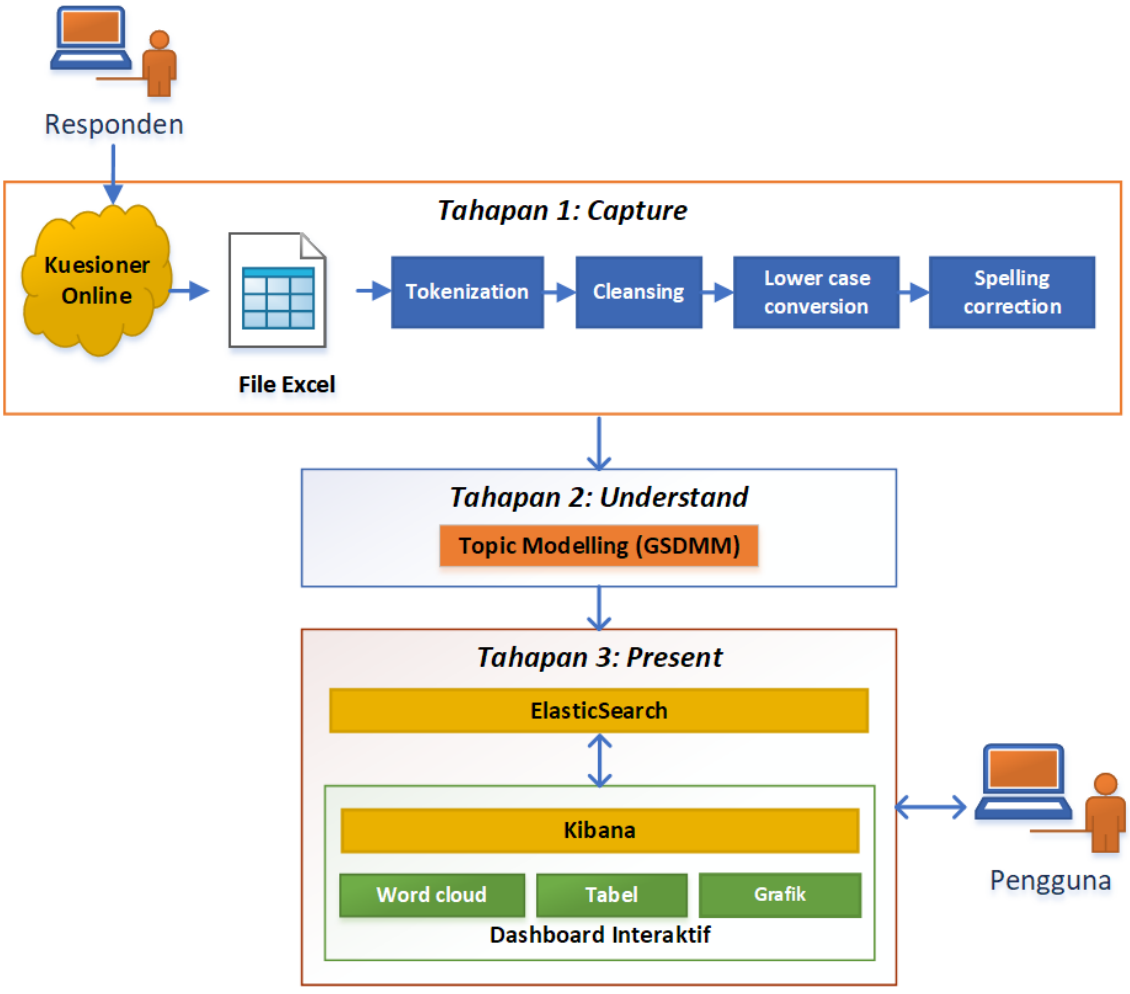
Pada tahapan ini, hasil analisis dirangkum, dievaluasi dan ditampilkan kepada pengguna dalam bentuk yang mudah dipahami oleh pengguna. Salah satu cara untuk

menampilkan data yang kompleks agar mudah dipahami adalah visualisasi. Tergantung dari jenis data ataupun informasi yang ingin ditampilkan, hasil analisis dapat divisualisasikan dalam beberapa bentuk seperti *word cloud*, grafik, histogram, tabel dan teks. Berbagai jenis visualisasi ini dapat digabungkan kedalam sebuah *dashboard*.

Pada penelitian ini, hasil analisis akan ditampilkan kedalam sebuah *dashboard* interaktif yang dibuat menggunakan Kibana. Karena sifatnya yang interaktif, maka pengguna akan dipermudah dalam menyaring ataupun mencari data yang diinginkan.

3. HASIL DAN PEMBAHASAN

Arsitektur sistem yang dibangun memiliki komponen-komponen yang disesuaikan dengan metode CUP yang digunakan. Sehingga sistem ini terbagi juga menjadi tiga bagian, yaitu bagian *capture*, *understand* dan *present*. Arsitektur sistem secara keseluruhan bisa dilihat pada Gambar 3.



Gambar 3. Arsitektur Sistem

Komponen tahapan *capture* dan *understand* dibuat menggunakan bahasa pemrograman Python karena mempunyai pustaka yang cukup lengkap untuk kebutuhan pemrograman text mining. Software Jupyter Notebook digunakan sebagai editor kode

program Python, yang memudahkan pengeditan kode secara interaktif dan bisa dijalankan per baris. Hal ini memudahkan proses pembuatan program dan eksplorasi data. Sedangkan untuk tahapan *present*, terdiri dari Elasticsearch versi 7.10 sebagai database NoSQL dan Kibana versi 7.10 sebagai software visualisasi.

Elasticsearch adalah sebuah software analisis dan pencarian data yang terdistribusi [10]. Maksud dari terdistribusi disini adalah Elasticsearch bisa diinstal pada beberapa komputer sekaligus sehingga membentuk sebuah *cluster*. Karena Elasticsearch bisa berjalan pada sebuah *cluster*, maka Elasticsearch mampu melakukan pemrosesan data dengan jumlah yang besar. Pada penelitian ini, Elasticsearch akan digunakan untuk menyimpan data hasil analisis kuesioner sebelum nantinya ditampilkan pada *dashboard* Kibana. Sedangkan Kibana adalah adalah software yang digunakan untuk menampilkan dan mengeksplorasi data yang tersimpan dalam Elasticsearch melalui berbagai visualisasi yang disatukan dalam sebuah *dashboard* [11].

3.1. Tahapan Pengumpulan Data (Capture)

Data yang digunakan dalam penelitian ini adalah data hasil survey kepuasan data mahasiswa terhadap layanan kemahasiswaan pada kampus XYZ yang dilaksanakan pada akhir semester ganjil tahun akademik 2020/2021. Survey ini dilaksanakan oleh Unit Penjaminan Mutu Fakultas melalui platform Google Forms. Responden survey adalah seluruh mahasiswa fakultas yang aktif pada tahun akademik 2020/2021. Jumlah responden survey adalah sebanyak 1019 orang.

Kuesioner dijalankan pada Google Forms, sehingga data yang didapatkan berupa file Excel hasil export dari Google Forms. Karena yang diperlukan hanyalah data saran, maka data kuesioner kuantitatif tidak diambil. Pengolahan data kuesioner kuantitatif berada diluar ruang lingkup penelitian ini. Selain kolom saran, kolom yang diambil lainnya adalah kolom Prodi dan Timestamp. Kedua kolom ini diambil sebagai informasi tambahan untuk mengetahui prodi mahasiswa yang memberikan saran. Data-data terkait identitas juga tidak diproses oleh sistem, untuk menjaga anonimitas responden.

Dari 1019 respon, sebanyak 338 tidak berisi kritik ataupun saran, sehingga total data yang bisa dianalisis adalah sebanyak 681 respon. Sebelum data dianalisis, perlu dilakukan *preprocessing*. Adapun tahapan-tahapan *preprocessing* yang dilakukan adalah sebagai berikut:

1. Tokenization

Seluruh kalimat yang ada akan dipisah menjadi komponen token yang menyusun kalimat tersebut. Hasil dari proses ini adalah sejumlah token yang sudah terpisah-pisah. Proses *tokenization* ini sendiri menggunakan implementasi yang tersedia pada library *Natural Language Toolkit* (NLTK) [12]. Ada beberapa macam tokenizer yang tersedia pada NLTK, namun pada penelitian ini, *TweetTokenizer* dipilih karena memang dirancang untuk memproses bahasa yang tidak baku seperti yang ada pada Twitter, ataupun teks pendek lainnya. Pada penelitian ini, frase yang umum seperti ‘terima kasih’ akan tetap dianggap sebagai sebuah token, sehingga tidak akan merubah makna dari frase tersebut. Contoh dari hasil *TweetTokenizer* dari NLTK ditampilkan pada Gambar 4, dimana kalimat “Menurut saya sudah cukup OK :-)” diproses oleh *TweetTokenizer* sehingga menjadi 6 token.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id

```
from nltk.tokenize.casual import TweetTokenizer
tokenizer = TweetTokenizer()
tokens = tokenizer.tokenize("Menurut saya sudah cukup OK :-)")
```

```
print(tokens)
```

```
['Menurut', 'saya', 'sudah', 'cukup', 'OK', ':-)']
```

Gambar 4. Contoh Penggunaan *TweetTokenizer*

2. Lower case conversion:

Pada tahapan ini, seluruh token hasil dari tokenization akan dikonversi menjadi huruf kecil. Hal ini dilakukan untuk menyederhanakan pemrosesan data. Sebagai contoh, tanpa *lower case conversion*, maka token 'Saya' akan dianggap berbeda dengan 'saya', padahal dari segi makna, kedua token atau kata ini adalah sama. Proses konversi ke *lower case* dilakukan dengan menggunakan fungsi *lower()* yang merupakan bawaan pada Python. Contoh konversi ini ditampilkan pada Gambar 5, dimana seluruh huruf kapital dirubah menjadi huruf kecil.

```
print(tokens)
```

```
['Menurut', 'saya', 'sudah', 'cukup', 'OK', ':-)']
```

```
# konversi menjadi lower case
lower = [i.lower() for i in tokens]
print(lower)
```

```
['menurut', 'saya', 'sudah', 'cukup', 'ok', ':-)']
```

Gambar 5. *Lower case conversion*

3. Cleansing

Pada tahapan ini kata-kata yang dinilai tidak penting untuk pemrosesan data, atau disebut dengan *stopwords*, akan dibuang. Beberapa contoh dari *stopwords* dalam Bahasa Indonesia adalah; 'yang', 'dan', 'atau', 'ke', 'dari', dst. Penelitian ini menggunakan *stopwords* yang telah dikompilasi oleh [13], ditambah dengan beberapa *stopwords* yang dikompilasi sendiri berdasarkan pengamatan terhadap data jawaban kuesioner.

Proses *cleansing* juga akan membuang karakter-karakter yang tidak penting seperti titik, tanda tanya, koma, ataupun karakter *whitespace* lainnya. Selain itu, alamat website, email, dan angka juga akan dibuang. Pada akhir dari proses cleansing ini hanya token yang memiliki makna penting untuk proses analisis yang akan tersisa. Contoh dari proses *cleansing* disajikan pada Gambar 6, dimana token-token 'wow', 'keren', 'sekali', 'ya', 'saya', 'jadi', 'tertarik', 'beli' disaring sehingga token yang tersisa adalah 'wow', 'keren', 'tertarik' dan 'beli'.

```
# cetak tokens yang akan di cleansing
print(tokens)
```

```
['wow', 'keren', 'sekali', 'ya', 'saya', 'jadi', 'tertarik', 'beli']
```

```
# lakukan proses cleansing
sw_processor = StopwordsProcessor(Language.ID)
sw_processor.remove_stopwords(tokens)
```

```
E:\bigdata\sources\listener\listener\ml\datasets\stopwords\tala_2003.txt loaded.
['wow', 'keren', 'tertarik', 'beli']
```

Gambar 6. Contoh Proses *Cleansing* Pada Sebuah Kalimat

4. Spelling correction

Respon yang ada pada kuesioner terkadang ditulis menggunakan kata tidak baku. Sehingga diperlukan adanya perbaikan ejaan kata. Pada langkah ini, kata-kata yang penulisannya salah akan dikoreksi, sesuai dengan daftar kata yang telah disediakan sebelumnya. Contohnya kata “tdk” akan dikoreksi menjadi “tidak” dan “kmps” akan dikoreksi menjadi “kampus”. Dari proses *spelling correction* akan dihasilkan daftar kata (*token*) yang akan siap untuk dianalisis pada tahapan berikutnya.

Pada Gambar 7 ditampilkan contoh penggunaan spelling correction dimana token ‘sy’ dikoreksi menjadi ‘saya’, dan token ‘kmps’ dikoreksi menjadi ‘kampus’. Koreksi dilakukan berdasarkan data yang telah tersimpan dalam variabel *spelling_dic*.

```
print("Sebelum spelling correction:")
words = ['hari', 'ini', 'sy', 'mau', 'datang', 'ke', 'kmps']
print(words)
words2 = []

for word in words:
    if word in spelling_dic:
        words2.append(spelling_dic[word])
    else:
        words2.append(word)

print("\nSetelah spelling correction:")
print(words2)
```

```
Sebelum spelling correction:
['hari', 'ini', 'sy', 'mau', 'datang', 'ke', 'kmps']

Setelah spelling correction:
['hari', 'ini', 'saya', 'mau', 'datang', 'ke', 'kampus']
```

Gambar 7. Contoh *spelling correction*

3.2. Tahapan Analisis Data (Understand)

Penelitian ini menggunakan implementasi algoritma *topic modelling* GSDMM yang dibuat oleh [14] menggunakan bahasa pemrograman Python. Setelah melalui proses ini, saran-saran yang memiliki kemiripan akan dimasukkan kedalam kelompok yang

sama, dimana setiap kelompok akan diberikan ID kelompok. Teks saran yang telah dikelompokkan ini kemudian diteruskan kepada Elasticsearch dan Kibana untuk ditampilkan.

Salah satu perbedaan antara LDA dan GSDMM adalah dari segi penentuan jumlah cluster atau topik. Pada algoritma LDA, jumlah topik perlu ditentukan terlebih dahulu. Dalam prosesnya, bisa jadi jumlah topik yang telah ditentukan tidak optimal untuk dataset yang dianalisis [15]. Sedangkan pada GSDMM, pengguna algoritma cukup menentukan jumlah topik maksimum, kemudian pada setiap iterasi GSDMM akan mencari jumlah topik yang sesuai dengan data yang dianalisis.

Algoritma GSDMM memiliki dua parameter yang mengontrol proses pengelompokan topik, yaitu *alpha* dan *beta*. *Alpha* adalah probabilitas dari mahasiswa (dokumen atau teks) untuk memilih sebuah meja (topik) kosong, semakin besar *alpha* maka akan semakin besar kemungkinan mahasiswa (teks) memilih meja (topik) kosong. Jika *alpha* = 0, maka mahasiswa tidak akan memilih meja kosong. Parameter *beta* mengontrol ketertarikan mahasiwa (teks) terhadap mahasiswa lain dengan minat film (topik) yang sama. Nilai *beta* yang rendah akan menyebabkan mahasiswa untuk berkelompok dengan mahasiswa lain yang memiliki minat sama. Sedangkan nilai *beta* yang tinggi akan menyebabkan mahasiswa tidak terlalu memperhatikan minat, melainkan jumlah mahasiswa yang ada dalam sebuah meja [14].

Pada penelitian ini, parameter yang digunakan adalah *alpha* = 0.1 dan *beta* = 0.1. Jumlah topik maksimum yang dipilih adalah 30, dengan iterasi sebanyak 50 kali. Dari dataset yang ada, algoritma GSDMM menghasilkan 18 topik. Beberapa sampel topik yang terdeteksi oleh GSDMM disajikan pada Tabel 1. Karena keterbatasan ruang, maka jumlah kata yang ditampilkan pada Tabel 1 dibatasi sebanyak 10 kata per topik.

Tabel 1. Sampel Topik Pada Dataset

ID	Token pada topik
26	mahasiswa, peningkatan, layanan, saran, semoga, pelayanan, tingkatkan, kemahasiswaan, melayani, informasi
7	pelayanan, semoga, mahasiswa, saran, tingkatkan, layanan, kemahasiswaan, fakultas, peningkatan
11	hujan, parkir, tingkatkan, lahan, memadai, saran, kampus, mahasiswa, terima kasih
19	mahasiswa, kemahasiswaan, saran, informasi, fakultas, kegiatan, surat, media, birokrasi, sosial
1	kondisi, mata kuliah, menjalani, pandemi, maksimal, pengajarannya, perancangan, daring, perkuliahan

JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id

topic	saran_text
26	yang baik dipertahankan dan ditingkatkan untuk menjadi yg lebih baik lgi
26	Saran dari saya, semoga tempat pelayanan menjadi lebih baik lagi.
26	Semoga dengan adanya survey ini bisa meningkatkan layanan kemahasiswaan
26	Lebih di maksimalkan lagi
26	sudah cukup baik, namun lebih baiknya lagi jika terus ditingkatkan layanan kemahasiswaannya agar mahasiswa menjadi puas dengan layanan yang di berikan.
26	pelayanan dalam segala aspek lebih baik lagi

Gambar 8. Beberapa sampel respon dengan ID topik 26 (peningkatan layanan).

Gambar 9. *Word cloud* untuk ID topik 26 (peningkatan layanan)

Dari Tabel 1 bisa dilihat bahwa topik dengan ID 26 secara umum berisi tentang peningkatan layanan. Hal ini bisa dikonfirmasi dengan melihat beberapa sampel teks dengan ID topik 26 (Gambar 8) yang berisi harapan dari mahasiswa akan adanya peningkatan pelayanan. Hal yang sama juga nampak pada *word cloud* untuk topik dengan ID 26 (Gambar 9), dimana kata-kata yang memiliki bobot tinggi adalah kata ‘tingkatkan’, ‘pelayanan’, ‘mahasiswa’ dan ‘kemahasiswaan’. Dari teks jawaban mahasiswa dan visualisasi yang ada bisa disimpulkan bahwa responden merasa bahwa layanan kemahasiswaan sudah baik, tetapi masih bisa ditingkatkan. Dari contoh kasus ini bisa

JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id

dilihat bahwa GSDMM bisa mengelompokkan saran yang ada pada kuesioner sesuai dengan kemiripan isi atau topiknya.

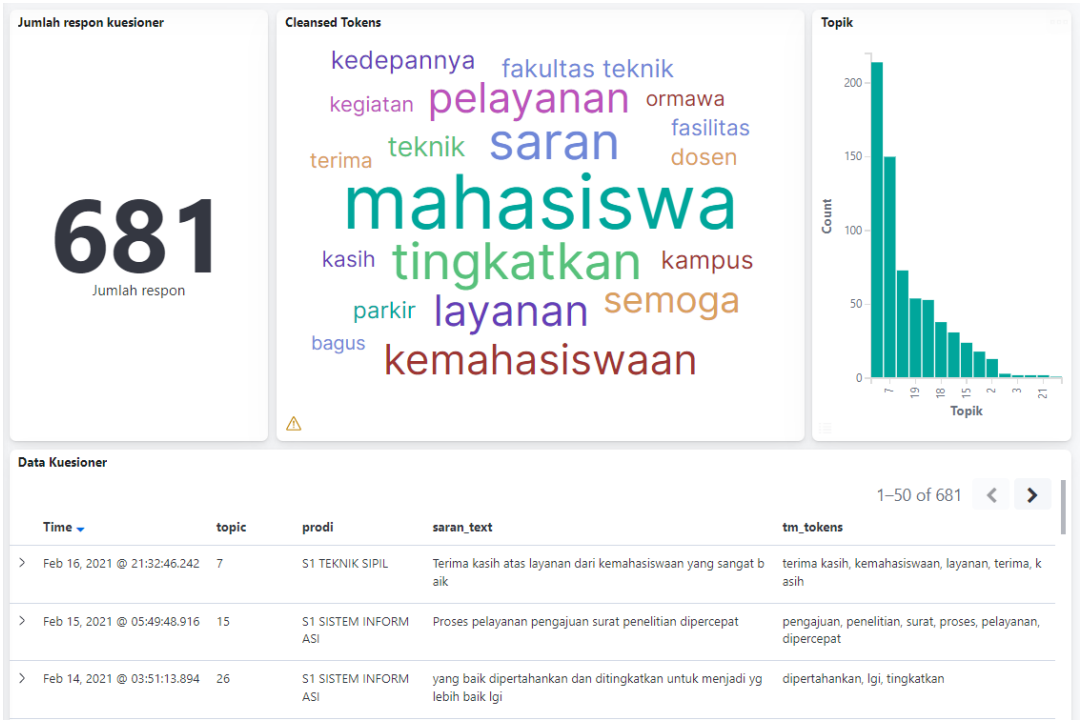
3.3. Tahapan Presentasi (Present)

Pada tahapan ini, hasil analisis harus disimpan dalam database Elasticsearch terlebih dahulu sebelum bisa ditampilkan dalam bentuk visualisasi oleh Kibana. *Mapping* index Elasticsearch yang digunakan pada penelitian ini ditampilkan pada Tabel 2.

Tabel 2. Konfigurasi *mapping* Elasticsearch

No	Field	Tipe	Keterangan
1	id	Long	Digunakan untuk menyimpan ID dari setiap saran.
2	timestamp	Date	Tanggal mahasiswa menjawab kuesioner
3	prodi	Keyword	Program studi mahasiswa
4	saran_text	Text	Jawaban saran mahasiswa yang belum melewati tahapan *pre-processing*. Jawaban ini disimpan utuh apa adanya.
5	tm_tokens	Keyword	Teks saran yang telah melewati pre-processing.
6	topic	Integer	ID kelompok topik yang dideteksi oleh GSDMM.

Hasil analisis divisualisasikan dalam beberapa bentuk visualisasi pada Kibana yaitu *word cloud*, *chart*, tabel dan teks. Visualisasi ini disusun menjadi satu kesatuan dalam sebuah *dashboard* interaktif, seperti tampil pada Gambar 10.



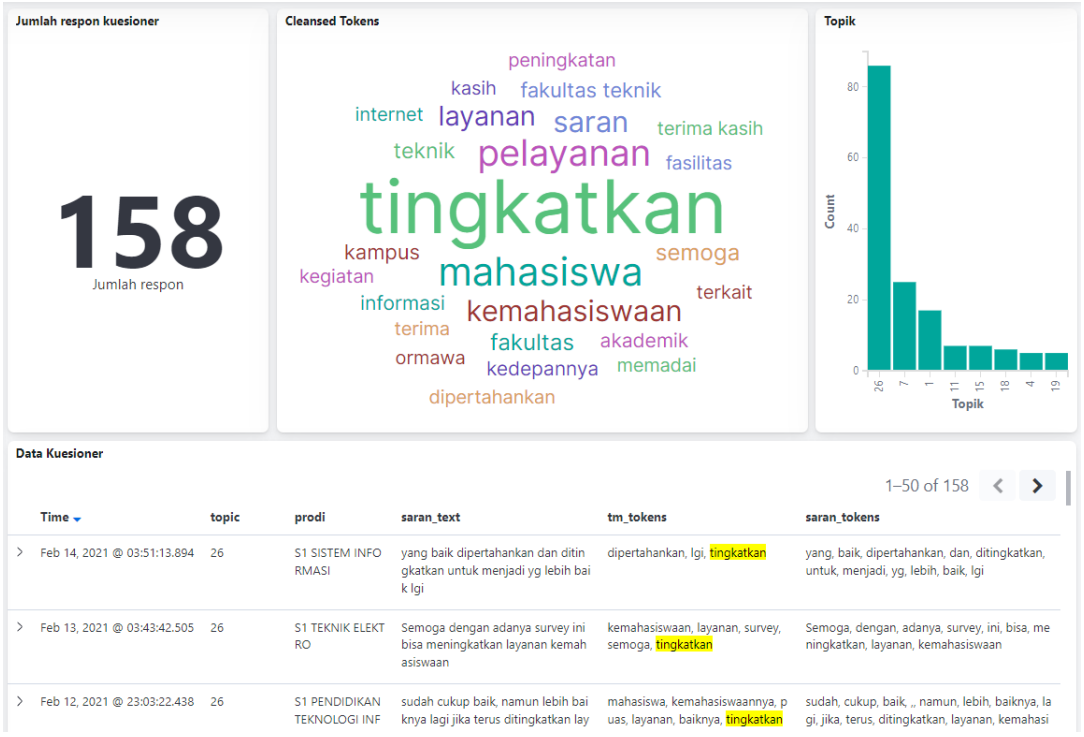
Gambar 10. *Dashboard* Kibana

Ada empat komponen visualisasi yang terdapat pada dashboard. Dari kiri ke kanan, komponen pertama menampilkan jumlah respon yang berada dalam database

JSI : Jurnal Sistem Informasi *(E-Journal), VOL.14, NO.2, Oktober 2022*
ISSN Print : 2085-1588
ISSN Online :2355-4614
http://ejournal.unsri.ac.id/index.php/jsi/index
email: jsifasilkom@unsri.ac.id

Elasticsearch. Komponen kedua adalah visualisasi *word cloud*, yang menunjukkan kata-kata atau token yang populer. Semakin besar ukuran huruf kata tersebut, maka semakin banyak kata tersebut muncul dalam database. Kemudian, visualisasi ketiga adalah *bar chart* yang menunjukkan topik yang telah dideteksi oleh GSDMM, yang diurutkan berdasarkan jumlah respon atau saran yang tergabung pada masing-masing topik. Terakhir, pada bagian bawah *dashboard* terdapat tabel yang menyajikan data respon dari mahasiswa.

Salah satu kelebihan dari *dashboard* Kibana adalah visualisasi yang interaktif. Sebagai contoh, pengguna bisa mengklik salah satu kata pada *word cloud* untuk menyaring data. Misalnya dengan mengklik kata “tingkatkan”, maka pengguna bisa menampilkan seluruh saran mahasiswa yang memiliki kata tersebut. Seluruh elemen visualisasi pada dashboard akan berubah menyesuaikan dengan penyaringan ini. Hal ini bisa dilihat pada Gambar 11, dimana ditemukan ada 158 respon yang berisi kata “tingkatkan”. Dengan adanya fitur ini, pengguna bisa dengan mudah memvisualisasikan dan mengeksplorasi data yang ada secara cepat.



Gambar 11. Tampilan *dashboard* setelah disaring menggunakan kata "tingkatkan"

4. KESIMPULAN

Penelitian ini bertujuan untuk mempermudah analisis teks jawaban pertanyaan bebas yang diperoleh dari kuesioner seperti komentar, kritik dan saran responden. Untuk itu, sebuah sistem telah dirancang menggunakan metode *Capture, Understand and Present* dan diimplementasikan menggunakan bahasa pemrograman Python, database Elasticsearch dan *dashboard* Kibana.

Pada tahapan *capture*, sistem dapat melakukan pengumpulan dan pembersihan data untuk kemudian dianalisis lebih lanjut pada tahapan *understand* menggunakan algoritma GSDMM. Tahapan *understand* digunakan untuk mengelompokkan jawaban

kuesioner berdasarkan topik secara otomatis. Pengelompokan berdasarkan topik dilakukan untuk menemukan tema-tema umum yang ada pada jawaban kuesioner. Keluaran dari tahapan ini kemudian diteruskan kepada tahapan *present* yang menyediakan sebuah *dashboard* interaktif. Karena sifatnya yang interaktif, maka pengguna sistem dapat melakukan pencarian maupun penyaringan data dengan lebih mudah. Pencarian dan penyaringan data dapat dilakukan berdasarkan kelompok topik atau keyword yang ada pada jawaban kuesioner.

Saran untuk penelitian selanjutnya adalah agar sistem bisa dikembangkan dengan menambahkan algoritma text mining lainnya seperti *sentiment analysis* untuk mendeteksi sentimen dari responden kuesioner. Selain itu, penambahan *user interface* yang lebih baik pada tahapan pengumpulan dan analisis data akan mempermudah pengguna dalam menjalankan sistem.

REFERENSI

[1] P. Ward, T. Clark, R. Zabriskie, and T. Morris, "Paper/Pencil Versus Online Data Collection," *https://doi.org/10.1080/00222216.2014.11950314*, vol. 46, no. 1, pp. 84–105, 2017, doi: 10.1080/00222216.2014.11950314.

[2] E. Heiervang and R. Goodman, "Advantages and limitations of web-based surveys: evidence from a child mental health survey," *Social Psychiatry and Psychiatric Epidemiology 2009 46:1*, vol. 46, no. 1, pp. 69–76, Nov. 2009, doi: 10.1007/S00127-009-0171-9.

[3] D. Thwaites Bee and D. Murdoch-Eaton, "Questionnaire design: the good, the bad and the pitfalls," *Archives of Disease in Childhood - Education and Practice*, vol. 101, no. 4, pp. 210–212, Aug. 2016, doi: 10.1136/ARCHDISCHILD-2015-309450.

[4] W. Fan and M. D. Gordon, "The power of social media analytics," *Commun ACM*, vol. 57, no. 6, pp. 74–81, 2014, doi: 10.1145/2602574.

[5] D. M. Blei, A. Y. Ng, and J. B. Edu, "Latent Dirichlet Allocation Michael I. Jordan," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003, doi: 10.5555/944919.944937.

[6] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," p. 267, 2003, doi: 10.1145/860435.860485.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[8] J. Yin and J. Wang, "A Dirichlet multinomial mixture model-based approach for short text clustering," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 233–242, 2014, doi: 10.1145/2623330.2623715.

[9] J. Mazarura and A. de Waal, "A comparison of the performance of latent Dirichlet allocation and the Dirichlet multinomial mixture model on short text," *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, PRASA-RobMech 2016*, Jan. 2017, doi: 10.1109/ROBOMECH.2016.7813155.

[10] Elastic, "Elasticsearch: The Official Distributed Search & Analytics Engine | Elastic," 2022. https://www.elastic.co/elasticsearch/ (accessed Apr. 07, 2022).

[11] Elastic, "Kibana: Explore, Visualize, Discover Data | Elastic," 2022. https://www.elastic.co/kibana/ (accessed Apr. 07, 2022).

[12] S. Bird, "NLTK: The natural language toolkit," in *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Interactive Presentation Sessions*, 2006.

[13] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," *M.Sc. Thesis, Appendix D*, vol. pp, 2003.

[14] R. Walker, "GitHub - rwalk/gsdmm: GSDMM: Short text clustering," 2021. https://github.com/rwalk/gsdmm (accessed Apr. 07, 2022).

[15] J. Gan and Y. Qi, "Selection of the Optimal Number of Topics for LDA Topic Model—Taking Patent Policy Analysis as an Example," *Entropy*, vol. 23, no. 10, Oct. 2021, doi: 10.3390/E23101301.