



Pemodelan Topik Menggunakan Latent Dirichlet Allocation pada Ulasan Aplikasi PeduliLindungi

Jihadul Akbar¹, Tamrizal A.M¹, Yefta Tolla¹, Abdulrahmat E. Ahmad¹, Ainul Yaqin², Ema Utami²

¹Magister Teknik Informatika, Universitas Amikom Yogyakarta,
Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta, Indonesia 55283,

²Informatics. Faculty of Computer Science, Universitas Amikom Yogyakarta,
Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta, Indonesia 55283,

*Email Penulis Koresponden: jihadulakbar@students.amikom.ac.id

Abstrak

Pandemi covid-19 yang melanda seluruh dunia termasuk Indonesia, membutuhkan langkah-langkah pencegahan seperti penelusuran (tracing), pelacakan (tracking) dan pemberian peringatan (warning dan fencing). Salah satu langkah pencegahan yang dilakukan Pemerintah adalah melalui Keputusan Menteri Komunikasi dan Informatika Nomor 171 Tahun 2020 tentang Penetapan Aplikasi PeduliLindungi Dalam Rangka Pelaksanaan Surveilans Kesehatan Penanganan Coronavirus Disease 2019 (Covid-19), menetapkan aplikasi PeduliLindungi sebagai aplikasi surveilans kesehatan penanganan COVID-19. Berbagai komentar disampaikan masyarakat terhadap aplikasi PeduliLindungi termasuk melalui kolom ulasan pada *playstore*. Pada penelitian ini akan dilakukan pemodelan topik menggunakan LDA terhadap ulasan masyarakat tentang aplikasi PeduliLindungi. Jumlah data yang digunakan pada penelitian ini sebanyak 13.731 data yang didapatkan dengan melakukan scraping pada google play mulai tanggal 15 september s/d 6 desember 2021 menggunakan library google scrapping. Langkah-langkah yang dilakukan dalam penelitian ini adalah preprocessing dataset, melakukan proses word2vec, menghitung nilai coherence dan melakukan pemodelan topik. Berdasarkan perhitungan nilai coherence pembagian jumlah topik yang ideal adalah 5, kemudian setelah diproses menggunakan algoritma LDA kesimpulan dari ke 5 topik tersebut didefinisikan sebagai kendala pendaftaran, sertifikat vaksin, tanggal lahir yang tidak sesuai, kendala membuka aplikasi dan keluhan pengguna aplikasi.

Keywords:

Pemodelan Topik;
Latent Dirichlet Allocation;
PeduliLindungi;
Google Play Store;

Riwayat Artikel:

Diserahkan 01 Juni 2022
Direvisi 24 Agustus 2022
Diterima 07 September 2022
Dipublikasi 30 April 2023

DOI:

10.22441/incomtech.v13i1.15572

This is an open access article under the [CC BY-NC license](https://creativecommons.org/licenses/by-nc/4.0/)



1. PENDAHULUAN

Penyakit *Coronavirus Disease* (COVID-19) merupakan penyakit yang disebabkan oleh virus SARS-CoV-2 [1]. Covid-19 telah menyebar di seluruh dunia tidak terkecuali Indonesia. Sebagai langkah antisipasi penyebaran Covid-19 maka Indonesia mengambil langkah-langkah pencegahan. Seperti penelusuran (*tracing*), pelacakan (*tracking*) dan pemberian peringatan (*warning* dan *fencing*) terhadap seluruh masyarakat. Salah satu langkah pencegahan yang dilakukan pemerintah adalah melalui Keputusan Menteri Komunikasi dan Informatika Nomor 171 Tahun 2020 tentang Penetapan Aplikasi PeduliLindungi Dalam Rangka Pelaksanaan Surveilans Kesehatan Penanganan Corona Virus Disease 2019 (Covid-19). Berdasarkan keputusan Menteri Kominfo tersebut maka ditetapkan aplikasi PeduliLindungi sebagai aplikasi surveilans kesehatan penanganan COVID-19 [2].

Dengan menggunakan aplikasi PeduliLindungi telah memberikan tiga hal terkait kondisi penyebaran COVID-19 yakni jumlah masyarakat yang berada pada fasilitas publik, penelusuran masyarakat yang terpapar COVID-19 dan informasi zonasi paparan COVID-19 [3]. Isu terkait keamanan data pribadi sempat muncul pada aplikasi PeduliLindungi tetapi sebuah penelitian yang dilakukan oleh Wijayanto dkk [4], menunjukkan tidak ada potensi kebocoran data, hal ini dibuktikan dengan tidak ditemukannya *malware* dan *database* yang tersimpan pada smartphone. Selain sebagai aplikasi surveilans kesehatan, aplikasi PeduliLindungi juga merupakan bagian dari mitigasi bencana COVID-19 di Indonesia [5].

Berbagai komentar disampaikan oleh masyarakat melalui berbagai media mengenai aplikasi PeduliLindungi termasuk melalui kolom ulasan pada play store. Komentar yang disampaikan sesungguhnya merupakan masukan yang dapat mendorong aplikasi PeduliLindungi menjadi lebih baik. Pada umumnya setiap komentar akan dianalisis untuk dapat memberikan respon terbaik terhadap komentar yang disampaikan, tetapi ribuan komentar yang terdapat pada kolom ulasan play store tersebut akan menjadi sulit untuk dianalisis karena jumlah data yang begitu besar. Salah satu cara untuk membantu analisis terhadap data komentar yang jumlahnya besar adalah dengan membagi data berdasarkan topik yang dibahas pada data komentar yang ada.

Permasalahan dari membagi data komentar berdasarkan topik adalah jika dilakukan secara manual maka akan membutuhkan waktu yang cukup lama, sehingga dibutuhkan sebuah pemodelan data untuk dapat menentukan topik dari data komentar yang ditentukan. Pada penelitian ini akan dilakukan pemodelan topik terhadap ulasan yang disampaikan masyarakat tentang aplikasi PeduliLindungi melalui *play store*.

Pemodelan topik termasuk dalam kategori algoritma *Unsupervised Learning* karena data yang digunakan tidak memiliki label. Salah satu pemodelan topik yang paling populer adalah *Latent Dirichlet Allocation* (LDA) sebuah metode yang pertama kali diperkenalkan pada tahun 2003 oleh Blei, Ng dan Jordan [6].

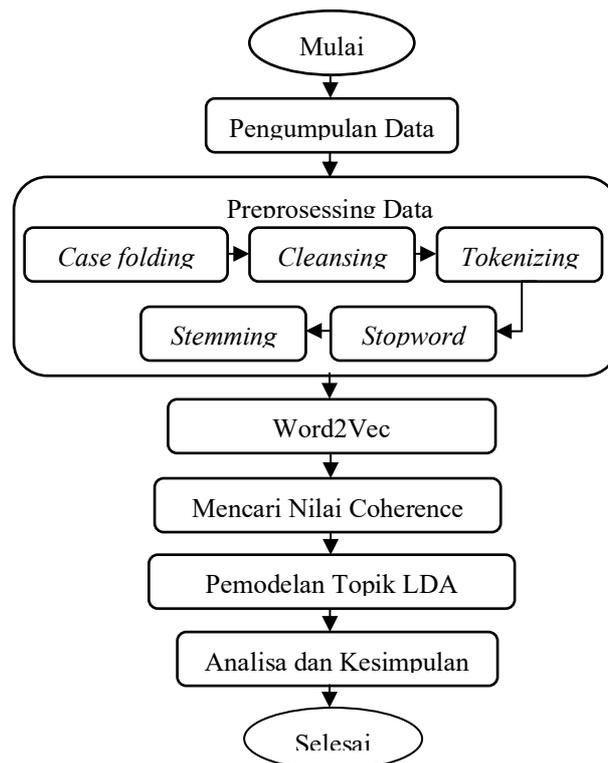
Beberapa penelitian pemodelan topik menggunakan LDA telah dilakukan diantaranya oleh Marjanen J dkk [7], yang membandingkan pemodelan topik *Dynamic Topic Model* (DTM) dengan *Latent Dirichlet Allocation* (LDA), pada penelitian ini didapatkan hasil LDA lebih baik dalam melakukan pemodelan topik

dibandingkan DTM. Penelitian lain dilakukan oleh Kim D dkk [8] yang melakukan klasifikasi sejumlah dokumen dengan menggunakan pemodelan topik LDA, kemudian penelitian yang dilakukan oleh Mutanga M B dkk [9] tentang pemodelan topik cuitan pada twitter yang membahas tentang Covid-19 di Afrika Selatan, dan penelitian yang dilakukan oleh Bastani K dkk [10] yang meneliti pemodelan topik komplain konsumen pada Consumer Financial Protection Bureau (CFPB). Beberapa penelitian diatas dalam menentukan jumlah topik yang ideal untuk diproses oleh LDA didapatkan dengan melakukan percobaan trial dan error. Pada penelitian ini akan dilakukan pemodelan topik ulasan aplikasi PeduliLindungi menggunakan LDA karena telah terbukti lebih baik pada peneliti sebelumnya.

2. METODE

2.1. Jenis Penelitian

Penelitian yang dilakukan merupakan penelitian eksperimental dengan pendekatan kuantitatif, karena pada penelitian ini akan fokus pada mencari topik yang dihasilkan pada dataset ulasan aplikasi PeduliLindungi. Alur penelitian dapat dilihat pada Gambar 1.



Gambar 1 Alur Penelitian

2.2. Ulasan Aplikasi

Ulasan merupakan teks yang berisikan pendapat yang mengupas seta memberi nilai terhadap produk atau karya sebagai tolak ukur untuk meningkatkan kualitas produk atau karya. Menurut Kamus Besar Bahasa Indonesia (KBBI) ulasan memiliki arti kutipan, tafsiran, atau komentar. Ulasan disebut juga *review*. Sedangkan aplikasi merupakan program yang digunakan untuk menyelesaikan

pekerjaan dengan menjalankan fungsi-fungsi tertentu [11]. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa ulasan aplikasi adalah memberikan komentar atau pendapat terhadap aplikasi. Tujuannya adalah untuk mengetahui aplikasi tersebut sudah memenuhi kebutuhan pengguna atau tidak.

2.3. Pengumpulan Data

Data yang digunakan dalam penelitian ini bersumber dari ulasan aplikasi PeduliLindungi pada *Play Store* dengan alamat URL: <https://play.google.com/store/apps/details?id=com.telkom.tracencare>. Data dikumpulkan tanggal 15 September 2021 sampai dengan 6 Desember 2021.

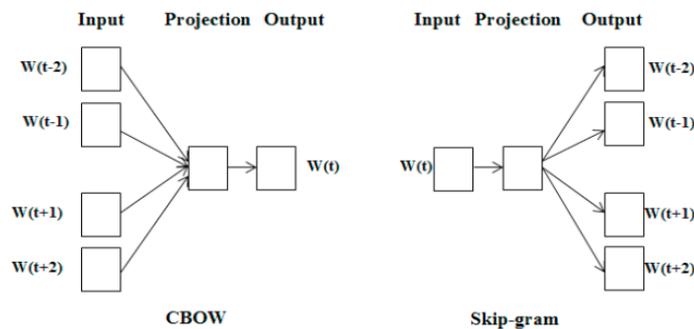
2.4. Preprocessing Data

Data yang telah dikumpulkan dilakukan pemrosesan dengan tahapan:

1. *Case folding*, merubah huruf kapital menjadi huruf kecil.
2. *Cleansing*, membersihkan data dari karakter yang tidak diperlukan dengan cara menghapus tanda bada, menghapus angka, karakter kosong (*whitespace*).
3. *Tokenizing*, memecah kalimat menjadi per kata.
4. *Stopword*, menghilangkan kata umum yang sering muncul dalam jumlah besar dan tidak memiliki makna. Dalam bahasa Indonesia contohnya seperti yang, dan, di, dari dll. Namun dalam dataset ini juga kami membuat list *stopword* yakni : “nya”, “yg”, “dg”, “gk”, “gak”, “sy”, “aja”, “ya”, “d”, “sdh”, “udah”, “ga”, “mohon”, ”udh”, ”gw”, ”tpi”, ”tp”, ”ndk”, ”lg”, ”utk” karena banyak dataset yang menggunakan singkatan.
5. *Stemming*, mengubah kata menjadi bentuk dasar dengan menghilangkan imbuhan dan akhiran [12], seperti kesembuhan: sembuh, pertolongan: tolong, membersihkan : bersih. *Stemming* yang kami gunakan yakni *library* sastrawi [13].

2.5. Word2Vec

Word2vec berfungsi untuk merubah kata-kata menjadi *vektor* yang bertujuan mencari kedekatan vektor antar kata [14],[15]. Data yang telah diproses pada tahap pemrosesan diubah menjadi *word vektor* dengan membuat model. Dalam model yang dihasilkan kata dalam kalimat bisa memiliki lebih dari satu *vektor* dengan konteks yang berbeda. *Word2vec* memiliki dua model yakni *Continuous Bag Of Word* (CBOW) dan *Skip-gram*. CBOW bekerja dengan memprediksi probabilitas sebuah kata yang diberikan kontaknya serta lebih cepat dengan kinerja lebih baik pada kumpulan kata yang besar. Sedangkan *Skip-gram* bekerja dengan memprediksi kontek yang diberikan kata target, model ini berfungsi dengan baik pada kumpulan data kecil dan representasi yang baik untuk kata-kata langka. Gambar 2 merupakan arsitektur dari dua model *Word2Vec*.



Gambar 2. Arsitektur CBOW dan Skip-gram sumber [15]

2.6. Menghitung Nilai Coherence Topik

Coherence Topik melakukan penilaian suatu topik dengan mengukur tingkat kemiripan semantik antar kata-kata yang terdapat dalam topik tersebut [16]. Pengukuran ini dapat digunakan untuk membedakan antara topik yang dapat ditafsirkan secara semantik dengan topik yang merupakan artefak dari inferensi statistik. Perhitungan nilai *coherence* dapat digunakan untuk membantu menentukan jumlah topik yang akan digunakan dalam proses pemodelan topik menggunakan LDA [17]. Rumus yang digunakan dalam menghitung nilai coherence dapat dilihat pada persamaan 1.

$$coherence(V) = \sum_{(v_i, v_j) \in V} score(v_i, v_j, \epsilon) \quad (1)$$

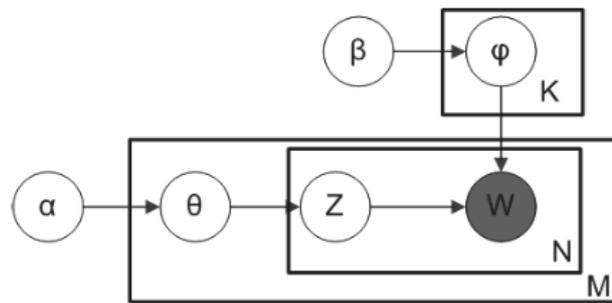
Dimana V adalah kumpulan kata yang mencerminkan topik dan ϵ merupakan faktor yang menjamin skor akan kembali menjadi bilangan real [18].

2.7. Latent Dirichlet Allocation (LDA)

Konsep kerja dari LDA adalah dokumen dianggap sebagai kombinasi dari beberapa topik, karakteristik suatu topik ditentukan oleh distribusi kata. Kata didalam LDA sebetulnya istilah, kumpulan kata-kata disebut dokumen, kumpulan dokumen disebut corpus dan kumpulan semua istilah yang ada dalam corpus disebut kosakata. Metode LDA digunakan untuk menemukan topik dimana ciri topik ditentukan oleh distribusi kata [18]. Tahapan yang digunakan untuk menentukan topik dalam LDA yakni :

1. Tentukan jumlah topik.
2. Berikan inisialisasi topik acak dalam kata-kata.
3. Hitung nilai probabilitas topik pada dokumen dan probabilitas kata pada topik untuk melihat prevalensi topik pada dokumen dan probabilitas kata pada topik.
4. Perbaharui topik pada setiap kata berdasarkan nilai probabilitas tertinggi.

Model graf probabilistik dari LDA pada langkah 2 dapat dilihat pada Gambar 3 dalam bentuk *Directed Acyclic Graf* (DAG).



Gambar 3. Model probabilistik graf LDA

Berdasarkan Gambar 3, α dan β merupakan parameter pada corpus. θ merupakan variabel level dokumen. Dimana z dan w adalah variabel tingkat kata (istilah). Variabel adalah simbol objek yang dapat diisi dengan konten bergantian tetapi tipe yang dapat didistribusikan. Sehingga isi z dan w dapat diubah. Sebab nilai berubah mengikuti distribusi dari variabel yang memiliki parameter tertentu. Distribusi DAG pada Gambar 2 dapat ditulis pada persamaan (2).

$$p(w, z, \theta, \phi | \alpha, \beta) = p(\theta | \alpha) p(z | \theta) p(\phi | \beta) p(w | z, \phi). \quad (2)$$

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan Data

Pengumpulan data ulasan aplikasi PeduliLindungi dilakukan dengan teknik *scraping* menggunakan *library google play scraper* [19] pada tanggal 15 September 2021 sampai dengan 6 Desember 2021. Data yang berhasil dikumpulkan sejumlah 13.731 baris. Atribut data yang dikumpulkan yakni *reviewId*, *userName*, *userImage*, *content*, *score*, *thumbsUpCount*, *reviewCreatedVersion*, *at*, *replyContent*, dan *repliedAt*. Atribut yang digunakan hanya isi dari atribut *content*. Hasil data yang dikumpulkan dapat dilihat pada Gambar 4.

| | content | at |
|-------|-----------------------------------------------------|---------------------|
| 0 | Kalo belum profesional untuk membuat sebuah ap... | 2021-12-04 18:49:50 |
| 1 | Terimakasih atas kepeduliannya,, udah saya te... | 2021-12-05 14:14:49 |
| 2 | 1. Barcode tdk bisa digunakan 2. Sertifikasi v... | 2021-12-04 22:32:09 |
| 3 | aplikasi bisa diunduh, dan ketika habis vaksin ... | 2021-12-04 14:33:20 |
| 4 | Apakah harus banget pakai Android versi 7 ke a... | 2021-12-02 08:30:56 |
| ... | ... | ... |
| 13726 | Ini memang diperlukan agar masyarakat umum mau... | 2021-11-29 23:51:26 |
| 13727 | aplikasi apa ini gaess? mau liat sertifikat v... | 2021-12-01 16:42:50 |
| 13728 | Sangat membantu..., setelah vaksin kedua, hp say... | 2021-12-03 08:25:11 |
| 13729 | Ini gimana sih. Muncul popup lokasi buatan ter... | 2021-11-28 20:46:00 |
| 13730 | Mohon petunjuk, untuk mengaktifkan Scan QR (b... | 2021-11-29 08:42:03 |

13731 rows × 2 columns

Gambar 4. Hasil pengumpulan data

3.2. Preprocessing Data

Pada tahap preprocessing data ini dilakukan untuk mengubah agar data lebih mudah dibaca, mengurangi waktu proses data mining yang dibutuhkan dan memudahkan proses analisa data. Berikut contoh hasil dari 5 tahapan preprocessing yang telah dilakukan dapat dilihat pada Tabel 1.

Tabel 1. Hasil Preprocessing

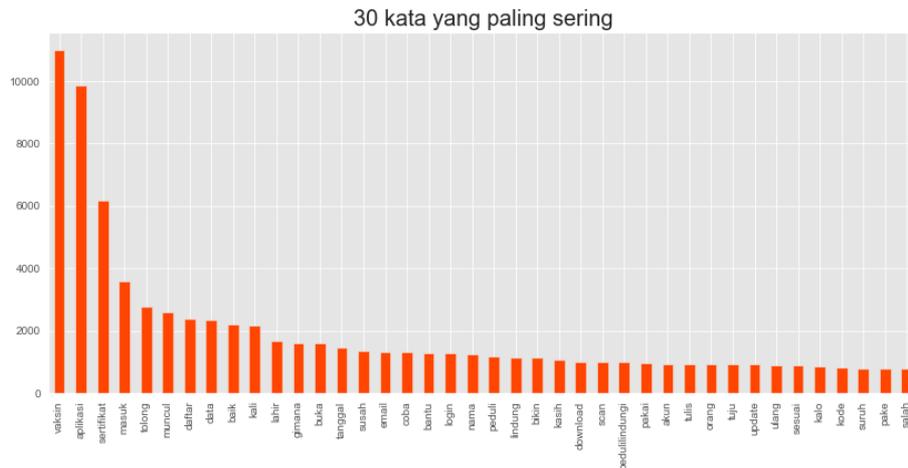
| Proses | Hasil |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ulasan | Aplikasi apa ini? mau liat sertifikat vaksin malah keluar dengan sendirinya. Gimana mau mencetak, udah capek2 nurutin pemerintah buat vaksin. |
| Case folding | aplikasi apa ini? mau liat sertifikat vaksin malah keluar dengan sendirinya. gimana mau mencetak, udah capek2 nurutin pemerintah buat vaksin. |
| Cleansing | aplikasi apa ini mau liat sertifikat vaksin malah keluar dengan sendirinya gimana mau mencetak, udah capek nurutin pemerintah buat vaksin. |
| Tokenizing | ['aplikasi', 'apa', 'ini', 'mau', 'liat', 'sertifikat', 'vaksin', 'malah', 'keluar', 'dengan', 'sendirinya', 'gimana', 'mau', 'mencetak', 'udah', 'capek', 'nurutin', 'pemerintah', 'buat', 'vaksin'] |
| Stopword | ['aplikasi', 'liat', 'sertifikat', 'vaksin', 'gimana', 'mencetak', 'udah', 'capek', 'nurutin', 'pemerintah', 'vaksin'] |
| Stemming | ['aplikasi', 'liat', 'sertifikat', 'vaksin', 'gimana', 'cetak', 'udah', 'capek', 'nurutin', 'pemerintah', 'vaksin'] |

Setelah dilakukan preprocessing, distribusi kemunculan kata dalam dataset dapat dianalisis dengan menggunakan *Wordcloud*. Kata dengan ukuran yang lebih besar merupakan kata yang paling banyak muncul. Berikut adalah *wordcloud* ulasan aplikasi PeduliLindungi yang ada pada *google play store*.



Gambar 5. Wordcloud Review PeduliLindungi

Gambar 5 merupakan hasil *wordcloud* ulasan aplikasi PeduliLindungi dengan kata yang sering muncul aplikasi, vaksin, masyarakat, android, sertifikat. Selain itu, analisis yang dilakukan adalah dengan membuat grafik 30 kata yang paling sering muncul dalam ulasan aplikasi PeduliLindungi.



Gambar 6. Grafik 30 kata yang paling sering muncul

Gambar 6 memperlihatkan grafik 30 kata yang sering muncul dalam dokumen dimana kata yang paling banyak muncul adalah 1). Vaksin dengan nilai kemunculan lebih dari 10.000, 2). Aplikasi dengan jumlah kurang lebih 9.000, 3). Sertifikat dengan jumlah 6.000 lebih dan sisanya kurang dari 4.000.

3.3. Proses *Word2Vec*

Parameter yang digunakan untuk melatih model *word2vec* yakni *min_count* 600, *window* : 10, *vector_size*: 250, *alpha*=0.03, *min_alpha*=0.0007, *workers* = 4, dan *seed* = 42. Maksud dari parameter tersebut adalah *min_count* artinya membatasi pada kata-kata yang memiliki frekuensi di atas 600, *vector_size* merupakan ukuran *vektor* dari sebuah kata diatur menjadi 250, untuk *window* 10 berarti untuk sebuah kata akan terlihat 10 kata di sebelah kanan dan 10 kata di sebelah kiri. *Alpha* dan *min_alpha* adalah *learning rate*, maka dimulai dengan learning rate 0,03, dan kemudian mendekati optimasi, *learning rate* akan turun ke 0,0007, untuk menemukan bobot yang dioptimalkan. Hasil dari model *word2vec* dapat dilihat dengan melihat kemiripan dari kata yang dihasilkan, Gambar 7 merupakan hasil dari model *word2vec* dengan kata kunci vaksin. Kata vaksin memiliki kemiripan dengan kata sertifikatnya, sertifikat, muncul, status, pedulilindungi, email, kasih, gimana, data dan kali.

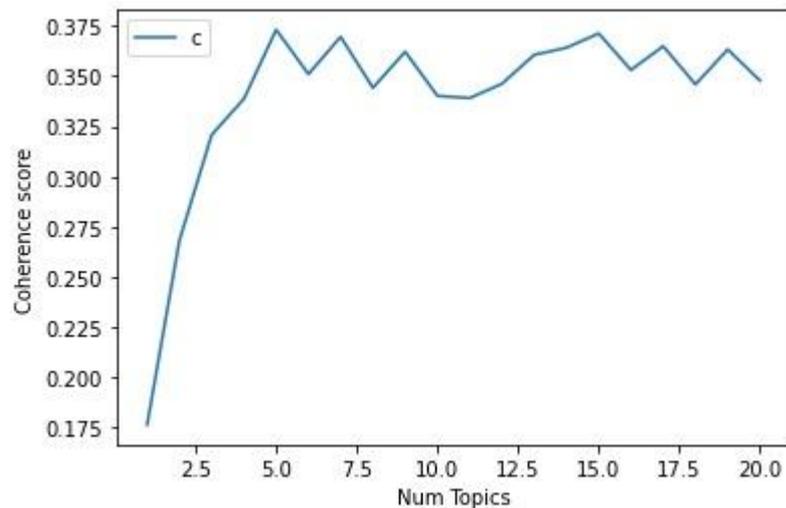
```
sim_words = w2v_model.wv.most_similar('vaksin')
print(sim_words)

[('sertifikatnya', 0.9933081269264221), ('sertifikat', 0.9928893446922302), ('muncul', 0.9836862683296204), ('status', 0.9628976583480835), ('pedulilindungi', 0.9302995204925537), ('email', 0.8521705269813538), ('kasih', 0.8116745948791504), ('gimana', 0.6957372426986694), ('data', 0.6735671162605286), ('kali', 0.6540766954421997)]
```

Gambar 7. Hasil *word2vec* kemiripan kata vaksin

3.4. Menghitung Nilai *Coherence* Topik

Selanjutnya dilakukan perhitungan nilai *coherence* untuk membantu menentukan jumlah topik yang akan digunakan. Hasil perhitungan nilai *coherence* dapat dilihat pada Gambar 8.

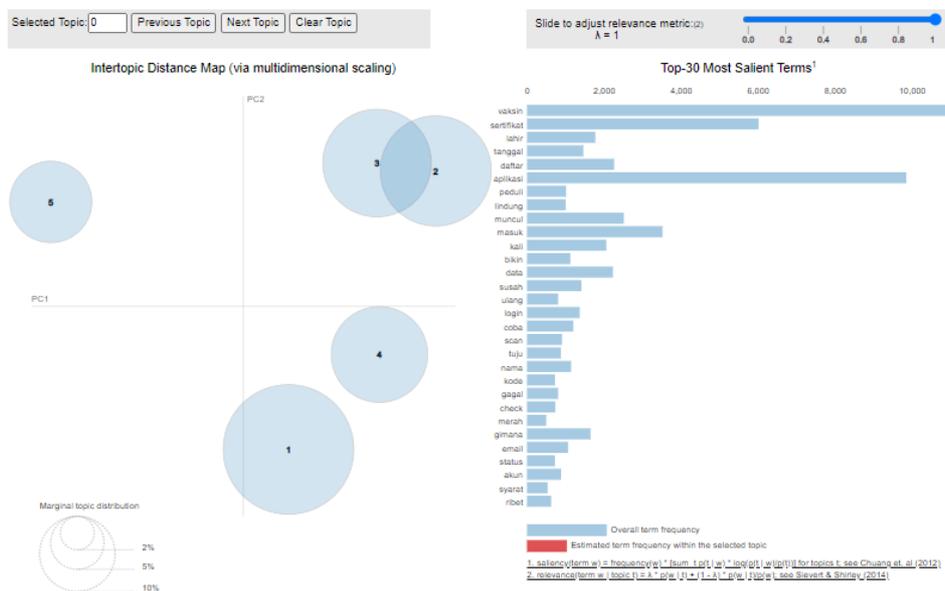


Gambar 8. Grafik *coherence score*

Berdasarkan data hasil perhitungan nilai *coherence* pada Gambar 8, ditentukan jumlah topik yang akan digunakan adalah 5 karena grafik menunjukkan tidak terjadi kenaikan nilai *coherence* yang signifikan setelah angka 5.

3.4. Pemodelan Topik

Hasil dari proses pemodelan menggunakan LDA disajikan dalam bentuk visualisasi pada Gambar 9.



Gambar 9. Hasil *Latent Dirichlet Allocation (LDA)*

Gambar 9 memperlihatkan hasil pemodelan dengan LDA dari 5 topik yang dihasilkan. Dari Gambar 9 tersebut dapat dilihat terdapat irisan antara topik 2 dan 3 yang berarti topik tersebut memiliki keterkaitan antar topik. Untuk lebih memperjelas kata-kata dominan yang terdapat pada masing-masing topik, akan ditampilkan dalam bentuk *wordcloud* dari masing-masing topik.

Gambar 10. *Wordcloud* dari 5 topik

Dari *Wordcloud* pada Gambar 10 dilakukan analisis hasil pemodelan dari masing-masing topik.

Tabel 2. Analisis Topik LDA

| No | Topik | Kata | Informasi |
|----|---------|------------------------------------------------------------------------------------------------------|------------------------------------|
| 1 | Topik 0 | “masuk”, “daftar”, “kali”, “aplikasi”, “coba”, “kode”, “gimana”, “gagal” | Kendala pendaftaran |
| 2 | Topik 1 | “vaksin”, “sertifikat”, “muncul”, “aplikasi”, “data”, “nama”, “sesuai”, “tolong”, “bantu”, “gimana”. | Sertifikat Vaksin |
| 3 | Topik 2 | “lahir”, “tanggal”, “peduli”, “lindungi”, “data”, “merah”, “status”, “ngisi”, “gakbisa” | Tanggal Lahir yang tidak sesuai |
| 4. | Topik 3 | “aplikasi”, “tuju”, “buka”, “masuk”, “baik”, “tolong”, “kasih”, “login”, “guna”, “bijak” | Kendala membuka aplikasi |
| 5. | Topik 4 | “aplikasi”, “susah”, “bikin”, “scan”, “masuk”, “check”, “ribet”, “bange”, “kalau” | Keluhan dalam menggunakan aplikasi |

Dari data yang ditampilkan pada Tabel 2 dapat disimpulkan bahwa pada topik 0 tentang kendala pendaftaran, pada topik 1 tentang informasi sertifikat vaksin, pada topik 2 tentang informasi tanggal lahir yang tidak sesuai, topik 3 tentang kendala membuka aplikasi dan pada topik 4 tentang keluhan dalam menggunakan aplikasi.

4. KESIMPULAN

Dari penelitian yang dilakukan terhadap 13.731 data pada ulasan *playstore* aplikasi PeduliLindungi, hasil perhitungan nilai coherence menunjukkan jumlah topik yang ideal adalah 5. Selanjutnya dengan algoritma LDA diusulkan kesimpulan dari 5 topik tersebut adalah tentang kendala pendaftaran (topik 0), sertifikat vaksin (topik 1), tanggal lahir yang tidak sesuai (topik 2), kendala membuka aplikasi (topik 3) dan keluhan pengguna aplikasi (topik 4). Untuk penelitian yang selanjutnya dapat dilakukan eksplorasi menggunakan algoritma pemodelan topik yang lain seperti *Latent Semantic Analysis* (LSA) atau *Non-negative matrix factorization* (NMF) untuk membandingkan rekomendasi pemodelan topik yang dihasilkan.

REFERENSI

- [1] Johns Hopkins University, "What Is Coronavirus?," *Johns Hopkins Medicine*, 2022. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus> (accessed Jan. 04, 2022).
- [2] Menteri Komunikasi Dan Informatika, "Keputusan Menteri Komunikasi Dan Informatika Republik Indonesia Nomor 171 Tahun 2020," 2020.
- [3] D. Herdiana, "Aplikasi Peduli Lindungi: Perlindungan Masyarakat Dalam Mengakses Fasilitas Publik Di Masa Pemberlakuan Kebijakan Ppkm," *J. Inov. Penelit.*, vol. 2, no. 6, pp. 1685–1694, 2021.
- [4] H. Wijayanto and S. Nasiroh, "Analisis Forensik Pada Aplikasi Peduli Lindungi Terhadap Kebocoran Data Pribadi," vol. 9, no. 2, 2021.
- [5] C. Eka Putri and H. Radja Erland, "Aplikasi Pedulilindungi Mitigasi Bencana Covid-19 Di Indonesia," *J. Pustaka Komun.*, vol. 4, no. 1, pp. 66–78, 2021.
- [6] L. Hagen, "Content analysis of e-petitions with topic modeling: How to train and evaluate LDA models?," *Inf. Process. Manag.*, vol. 54, no. 6, pp. 1292–1307, 2018, doi: 10.1016/j.ipm.2018.05.006.
- [7] J. Marjanen, E. Zosa, S. Hengchen, L. Pivovarov, and M. Tolonen, "Topic modelling discourse dynamics in historical newspapers," *CEUR Workshop Proc.*, vol. 2865, pp. 63–77, 2020.
- [8] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec," *Inf. Sci. (Ny.)*, vol. 477, pp. 15–29, 2019, doi: 10.1016/j.ins.2018.10.006.
- [9] M. B. Mutanga and A. Abayomi, "Tweeting on COVID-19 pandemic in South Africa: LDA-based topic modelling approach," *African J. Sci. Technol. Innov. Dev.*, vol. 0, no. 0, pp. 1–10, 2020, doi: 10.1080/20421338.2020.1817262.
- [10] K. Bastani, H. Namavari, and J. Shaffer, "Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints," *Expert Syst. Appl.*, vol. 127, pp. 256–271, 2019, doi: 10.1016/j.eswa.2019.03.001.
- [11] A. N. Royyan, Indriati, and L. Muflikhah, "Analisis Sentimen Review Aplikasi Mobile Dengan Menggunakan Metode Modified K-Nearest Neighbour (MK-NN)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3157–3161, 2018.
- [12] D. Farrar and J. H. Hayes, "A comparison of stemming techniques in tracing," *Proc. - 2019 IEEE/ACM 10th Int. Work. Softw. Syst. Traceability, SST 2019*, pp. 37–44, 2019, doi: 10.1109/SST.2019.00017.
- [13] "Sastrawi · PyPI." <https://pypi.org/project/Sastrawi/> (accessed Aug. 26, 2022).
- [14] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling Word2Vec on Big Corpus," *Data Sci. Eng.*, vol. 4, no. 2, pp. 157–175, 2019, doi: 10.1007/s41019-019-0096-6.
- [15] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in English words," *Procedia Comput. Sci.*, vol. 157, pp. 160–167, 2019, doi: 10.1016/j.procs.2019.08.153.
- [16] D. Maier *et al.*, "Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology," *Commun. Methods Meas.*, vol. 12, no. 2–3, pp. 93–118, 2018, doi: 10.1080/19312458.2018.1430754.
- [17] V. Gangadharan and D. Gupta, "Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1337–1345, 2020, doi: 10.1016/j.procs.2020.04.143.
- [18] S. Qomariyah, N. Iriawan, and K. Fithriasari, "Topic modeling Twitter data using Latent Dirichlet Allocation and Latent Semantic Analysis," *AIP Conf. Proc.*, vol. 2194, no. December 2019, 2019, doi: 10.1063/1.5139825.
- [19] "google-play-scraper · PyPI." <https://pypi.org/project/google-play-scraper/> (accessed Aug. 26, 2022).