Journal of Information Technology and Computer Science (INTECOMS)
Volume 5 Nomor 2, Desember 2022
e-ISSN : 2614-1574
p-ISSN : 2621-3249

PENGELOMPOKAN ULASAN APLIKASI PEDULILINDUNGI DENGAN ALGORITMA K-MEDOIDS

PEDULILINDUNGI APPLICATION REVIEW GROUPING WITH THE K-MEDOIDS ALGORITHM

Ahmad Habib Husaini[1], Rini Mayasari[2], Susilawati[3]
[1,2,3] Universitas Singaperbangsa Karawang
ahmad.habib18204@student.unsika.ac.id[1]

ABSTRACT

By looking at the reviews of potential users, you can see the responses of other users who have used the application first, besides the application reviews can be used as input for developers. Clustering reviews can use text mining. In this study, the Kmedoids algorithm was used to cluster reviews with Fasttext word embedding to represent the review's word units into vectors. The data used in this study amounted to 2,729 taken from the PeduliLindung application comment column on the Google Playstore. The results of the evaluation using the Davies Bouldin index or abbreviated as DBI by comparing sixteen trials and the best experiment was obtained, namely, an experiment using data without stemming, cbow architecture, using the Manhattan distance metric, and an experiment using data without stemming, cbow architecture, and using the cityblock distance metric with the same DBI value of 2.93 which resulted in two clusters of reviews. In the first cluster there were 2,183 reviews while in the second cluster there were 546 reviews.

***Keyword** : Reviews, Applications, Review Clustering, Kmedoids, Word Embedding, Fasttext*

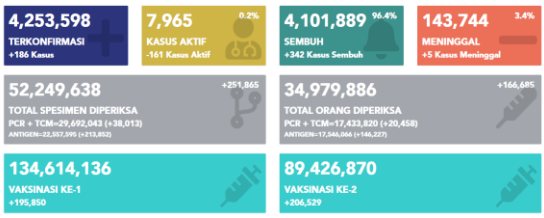
ABSTRAK

Dengan melihat ulasan calon pengguna dapat melihat tanggapan pengguna lain yang sudah menggunakan aplikasi lebih dahulu, selain itu ulasan aplikasi dapat dijadikan masukan untuk pengembang. Penglompokan ulasan dapat menggunakan teknik *text mining* atau penambangan *text*. Pada penelitian ini menggunakan algoritma Kmedoids untuk mengelompokkan ulasan dengan *fasttext word embedding* sebangai teknik untuk merepresentasikan satuan kata dalam ulasan menjadi vektor. Data yang digunakan pada penelitian ini berjumlah 2.729 yang diambil dari kolom komentar aplikasi PeduliLindungi pada google playstore. Hasil dari evaluasi menggunakan *davies bouldin index* atau disingkat DBI dengan membandingkan enam belas percobaan dan di dapatkan percobaan terbaik ialah percobaan menggunakan data tanpa *stemming, architecture cbow*, menggunakan *metric manhattan distance* dan percobaan menggunakan data tanpa *stemming, architecture cbow*, serta menggunakan *metric cityblock distance* dengan nilai DBI yang sama yakni 2,93 yang menghasilkan dua kelompok ulasan. Pada kelompok pertama terdapat sebanyak 2.183 ulasan sedangkan pada kelompok kedua terdapat sebanyak 546 ulasan.

Kata Kunci*:* Ulasan, Aplikasi, Pengelompokan Ulasan, Kmedoids, *Word Embedding, Fasttext.*

PENDAHULUAN

Saat ini dunia sedang menghadapi wabah COVID-19, tak terkecuali di Indonesia. Virus Corona atau *severe acute respiratory syndrome coronavirus* 2 disingkat SARS-CoV-2 berasal dari kota Wuhan, Provinsi Hubei China merupakan virus yang menyerang saluran pernapasan baik orang dewasa maupun anak-anak serta dapat menyebabkan gangguan ringan seperti batuk-batuk, demam hingga yang terparah yakni kematian (Ndwandwe & Wiysonge, 2021). Di Indonesia sendiri kasus pertama pasien positif virus corona diumumkan oleh pemerintah pada tanggal 2 maret 2020, sejak saat itu jumlah kasus terus meningkat hingga tanggal 30 November 2021 total sudah ada 4,25 juta kasus dengan kasus aktif sebanyak 7.965 pasien dan kasus sembuh sebanyak 4.1 juta pasien yang dapat dilihat pada gambar 1 (Chollisni et al., 2022).



Gambar 1. Peta Sebaran Covid-19 Indonesia Tanggal 30 November

Sumber : covid19.go.id/peta-sebaran

Dalam upaya menanggulangi penyebaran virus corona, pemerintah membuat serangkaian kebijakan tertulis. Seperti kebijakan berdiam diri di rumah, pembatasan sosial, pembatasan fisik, menggunakan masker, mencuci tangan, bekerj dan belajar dari rumah, serta tidak berkerumun. Bukti nyata dalam penerapan salah satu kebijakan pemerintah agar tidak berkerumn ialah dengan perilisan aplikasi PeduliLindungi (Sabrina et al., 2021; Herdiana, 2021).

PeduliLindungi merupakan aplikasi yang dikembangkan guna membantu instansi pemerintah serta aparatur negara dalam pelacakan untuk menghentikan penyebaran covid-19. Aplikasi ini mengidentifikasi lokasi pengguna serta memberikan informasi keramaian suatu lokasi secara berkala. Hal tersebut dapat membantu pemerintah dalam melakukan proses tracing untuk menghentikan mata rantai penyebaran Covid-19 di Indonesia (Haerani, & Rahmatulloh, 2021).

Aplikasi ini dapat diunduh secara gratis pada *Google Play Store*. *Google Play Store* memiliki fitur *review* atau ulasan yang dapat digunakan oleh pengguna untuk memberkan penilaian terhadap suatu aplikasi.

Pada praktiknya pengembang suatu aplikasi tidak membaca tiap komentar untuk mengetahui tanggapan pengguna terhdadap aplikas yang dikembangkan tetapi memanfaatkan salah satu penerapan kecerdasan buaran khususnya pembelajaran mesin atau *machine learning* yaitu analisis sentiment (Hertina et al., 2021).

Analisis sentiment atau dapat disebut juga penggalian opini merupakan penyatuan berbagai bidang kecerdasan buatan yakni, *natural language processing, data mining* dan *text mining* dengan tujuan untuk menemukan pendapat seseorang dalam suatu ungkapan tertulis (Farhadloo & Rolland, 2016). Dibalik layar, analisis sentimen hanya mengelompokan data *text* menjadi kategori tertentu.

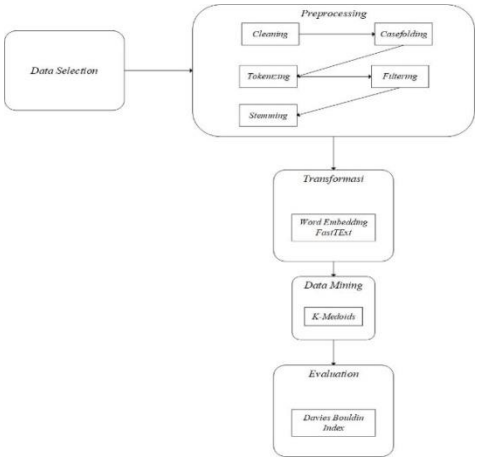
Penelitiian yang dilakukan oleh (Yudiarta et al., 2018)berhasil mengelompokkan teks dengan algoritma k-means *clustering* dengan menggunakan *term frequency inverse document frequency* (TF-IDF) sebagai cara merepresentasikan teks menjadi vector angka menghasilkan tingkat *purity* berkisar dari 78% hingga 83%.

(Nurdin et al., 2020) membandingkan kinerja *word embedding* yaitu Word2vec, GloVe dan FastText pada klasifikasi teks. Hasil yang diperoleh menghasilkan *Word Embedding* FastText memiliki kinerja terbaik dibandingkan dua *word embedding* lainya yaitu *Word2vec*, *GloVe* walau perbedaannya tidak terlalu signifikan.

Penelitian ini bertujuan untuk mengelompokkan ulasan aplikasi PeduliLindungi pada kolom komentar *Google Play Store* dengan algoritma k-medoids *clustering* serta menerapkan *Word Embedding* FastText pada proses transformasi.

METODE

Metode penelitian yang digunakan dalam penelitian ini ialah KDD (*Knowledge Discovery in Database)*. KDD merupakan proses terorganisir dalam melakukan identifikasi terhadap pola dari *dataset* yang kompleks(Gata, 2016). Secara formal KDD dapat didefinisikan sebagai proses *non-trivial* atau tidak sederhana untuk mengidentifikasi pola yang valid, baru, berpotensi berguna dan dapat dipahami dengan proses yang sangat *iterative* (Imberman, 2001).



Gambar 2. Alur Penelitian

Secara garis besar, tahapan *Knowledge Discovery in Database* (KDD) tersebut dapat dijelaskan sebagai berikut (Karsito & Sari, 2018)

1. Data Selection
   Pada tahapan ini berfokus untuk pemilihan data yang akan digunakan. Data yang digunakan didapat dari situs G*oogle Play Store*. Data diambil menggunakan library "*google-play-scraper*" dengan bahasa pemrograman *python*. Pada tahapan ini dilakukan dua kali seleksi, yang pertama hanya mengambil ulasan berbahasa Indonesia dan kedua data ulasan pada tanggal 1 november hingga 30 Desember 2021 yang akan digunakan untuk proses pemodelan.
2. *Preprocessing*
   Pada tahapan ini berfokus untuk mempersiapkan data dengan beberapa tahap yaitu:
   a) *Cleaning* tahapan yang dilakukan pada proses ini adalah membersihkan seluruh ulasan dari *email, mention, hashtag, special character, punctuation, multiple space, duplicate review, emoticon* dan ulasan kosong.
   b) *Case folding* merupakan tahapan untuk menyeragamkan kata, lebih spesifiknya lagi huruf yang semula kapital (upper case) diubah menjadi *lowercase*.
   c) *Tokenizing* , merupakan tahapan untuk mengubah data yang sebelumnya berbentuk kalimat menjadi per kata.
   d) *Filtering*, merupakan tahapan untuk menghilangkan kata yang kurang bermakna atau tidak terlalu berpengaruh seperti kata "ada", "adalah", "adapun", "aku", "yang", dan semisalnya makan kata-kata tersebut akan dihilangkan.
   e) *Stemming*, merupakantahapan untuk menghapus imbuhan pada kata dan mengubah kata menjadi kata dasar dengan menggunakan *library Sastrawi*.
3. *Transformation*
   Tahapan ini berfokus untuk mengubah data yang telah melewati tahapan *preprocessing* agar bisa diproses oleh algoritma dengan mengubah data menjadi bentuk vektor angka dengan menggunakan *Word Embedding Fasttext.*
4. *Data Mining*
   Tahapan ini berfokus untuk mencari pola atau informasi tersembunyi dari data menggunakan algoritma Kmedoids *clustering* dengan menggunakan empat *distance measure* yaitu *euclidean distance, manhattan distance dan cosine distance.* Berikut persamaan tiap *distance measure.*
   1. *Euclidean Distance*

$$d(x,y) = \left\{\sum_{i=1}^{n}(x_i - y_i)^2\right\}^{\frac{1}{2}}$$

   2. *Manhattan Distance*

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$$

   3. *Cosine Distance*

$$d(x,y) = 1 - \left(\frac{\vec{x}.\vec{y}}{\|\vec{x}\|.\|\vec{y}\|}\right)$$

5. *Evaluation*
   Tahap evaluasi akan menggunakan nilai *davies boundin index* atau disingkat DBI. Berikut persamaan *davies bouldin index.*

$$DBI = \bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i$$

Dengan $R_i$merupakan maksimum dari $R_{ij}, i \neq j$.

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}$$

Untuk persamaan S_i sebagai berikut:

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} \left| X_j - A_i \right|^q \right\}^{1/q}$$

merupakan banyaknya data tiap cluster dan Ai adalah pusat dari cluster. Sedangkan persamaan M_{ij} sebagai berikut :

$$M_{ij} = \left\{ \sum_{k=1}^{N} \left| a_{ki} - a_{kj} \right|^p \right\}^{1/p}$$

HASIL DAN PEMBAHASAN

Proses pengumpulan data ulasan dilakukan dengan menggunakan *library python* atau bisa disebut juga *package* berbahasa *python* bernama *google_play_scraper*. Data yang dikumpulkan merupakan ulasan hingga tanggal 23 Juli 2022 dengan total data sebanyak 50.668 ulasan. Contoh data yang telah dikumpulkan dapat dilihat pada gambar 4 dibawah ini.

	content	at
48170	sangat membantu dalam rangka pandemi covid 19....	2021-02-26 19:19:23
25823	Loading terus waktu bikin akun	2021-09-15 11:19:26
9425	Ini lagi kenapa ya peduli lindungi, tiyap kali...	2022-01-08 15:24:15
12092	Seharian tidak ad NAR	2021-12-10 18:52:47
49040	jelek, tidak informatif terlalu kaku	2020-05-24 11:15:39
39376	Waktu open dia minta akses lokasi, setelah aks...	2021-08-13 08:44:32
32097	I can't check in everytime after scan the barcode	2021-09-03 13:12:40
9578	Really usefull	2022-01-07 20:15:22
46797	Kenapa gak bisa ya	2021-06-29 14:23:02
32798	ni aplikasi paling resek yang pernah saia inst...	2021-09-01 14:30:26

Gambar 3. Sampel Data Ulasan Berbahsa Indonesia

Data Selection

Data ulasan yang berhasil didapatkan sebanyak 50.665. Data tersebut terbagi menjadi 34.081 data berbahasa Indoensia dan 16.587 data berbahasa bukan bahasa Indonesia. Penyeleksi bahasa yang digunakan dalam peneliian ini ialah menggunakan *library googletrans.*

Data ulasan yang digunakan hanya dari rentang 1 November 2021 hingga 30 Desember 2021. Artinya dari 16.587 data berbahasa Indonesia hanya 2.729 ulasan saja yang berada di rentang tanggal tersebut. Hasil proses seleksi dapat dilihat pada gambar 5.

	content	at
14168	sangat bagus.	2021-11-07 20:20:47
10753	Habis update kok gak bisa cek serifikat dan de...	2021-12-31 09:36:39
12580	Sangat membantu 👍	2021-12-02 11:21:59
11418	daripada dihukum 🤬	2021-12-25 10:53:04
13048	Berikan yang terbaik buat Kita semua. Semangat	2021-11-24 06:24:36
11141	Bagus.	2021-12-28 07:11:04
11675	Setting yg sdh di buat berubah terus.	2021-12-20 07:07:09
11455	1. Bagaimana cara lihat sertifikat vaksin? Say...	2021-12-24 14:36:50
11564	Fitur nya biasa aja	2021-12-22 14:46:16
13682	Aplikasi ini tidak mendukung handphone android...	2021-11-14 11:18:51

Gambar 4. Hasil Proses *Cleaning*

Preprocessing

Data yang sudah diseleksi kemudian dilakukan *preprocessing*. Proses ini bertujuan untuk membersihkan data dari berbagai *noise*.

1. *Cleaning*

Tahap pertama dalam proses *preprocessing* ialah *cleaning*. Dalam tahap ini membersihkan ulasan dari *email, mention, hashtag, special character, punctuation, multiple space, duplicate review, emoticon* dan ulasan kosong. Hasil pada proses cleaning dapat dilihat pada gambar 6.

	content	at
14168	sangat bagus	2021-11-07 20:20:47
10753	Habis update kok gak bisa cek serifikat dan de...	2021-12-31 09:36:39
12580	Sangat membantu	2021-12-02 11:21:59
11418	daripada dihukum	2021-12-25 10:53:04
13048	Berikan yang terbaik buat Kita semua Semangat	2021-11-24 06:24:36
11141	Bagus	2021-12-28 07:11:04
11675	Setting yg sdh di buat berubah terus	2021-12-20 07:07:09
11455	1 Bagaimana cara lihat sertifikat vaksin Saya ...	2021-12-24 14:36:50
11564	Fitur nya biasa aja	2021-12-22 14:46:16
13682	Aplikasi ini tidak mendukung handphone android...	2021-11-14 11:18:51

Gambar 5. Hasil Proses *Case Folding*

2. *Case Folding*

Case folding merupakan tahapan untuk menyeragamkan kata, lebih spesifiknya lagi huruf yang semula kapital (*uppercase*) diubah menjadi *lowercase*. Hasil dari proses ini dapat dilihat pada gambar 7.

	content	at
11168	sangat membantu	2021-12-27 19:45:54
10753	habis update kok gak bisa cek serifikat dan de...	2021-12-31 09:36:39
11001	sangat bagus	2021-12-29 09:52:36
11418	daripada dihukum	2021-12-25 10:53:04
13048	berikan yang terbaik buat kita semua semangat	2021-11-24 06:24:36
10689	bagus	2021-12-31 20:03:40
11675	setting yg sdh di buat berubah terus	2021-12-20 07:07:09
11455	1 bagaimana cara lihat sertifikat vaksin saya ...	2021-12-24 14:36:50
11564	fitur nya biasa aja	2021-12-22 14:46:16
13682	aplikasi ini tidak mendukung handphone android...	2021-11-14 11:18:51

Gambar 6. Hasil Proses *Tokenization*

3. *Tokenizing*

Pada tahapan ini data ulasan yang sudah diseragamkan menjadi bentuk *lowercase* akan dilakukan proses tokenisasi. Data ulasan yang sebelumnya berbentuk kalimat akan dipecah menjadi perkata. Hasil proses *tokenization* dapat dilihat pada gambar 8.

	content_tokenize	at
11168	[sangat, membantu]	2021-12-27 19:45:54
10753	[habis, update, kok, gak, bisa, cek, serifikat, dan, detail, layan...	2021-12-31 09:36:39
11001	[sangat, bagus]	2021-12-29 09:52:36
11418	[daripada, dihukum]	2021-12-25 10:53:04
13048	[berikan, yang, terbaik, buat, kita, semua, semangat]	2021-11-24 06:24:36
10689	[bagus]	2021-12-31 20:03:40
11675	[setting, yg, sdh, di, buat, berubah, terus]	2021-12-20 07:07:09
11455	[1, bagaimana, cara, lihat, sertifikat, vaksin, saya, sendiri, jug...	2021-12-24 14:36:50
11564	[fitur, nya, biasa, aja]	2021-12-22 14:46:16
13682	[aplikasi, ini, tidak, mendukung, handphone, android, berumur, dia...	2021-11-14 11:18:51

Gambar 7. Hasil Proses *Tokenization*

4. *Filtering*

Pada proses *filtering* kata yang kurang bermakna atau tidak terlalu berpengaruh akan dihapus. Jika pada data terdapat kata stopword seperti “ada”, “adalah”, “dapaun”, “aku”, dan “yang”, makan kata-kata tersebut akan dihilangkan. Hasil filtering dapat dilihat pada gambar 9.

	content_filtered	at
11168	[sangat, membantu]	2021-12-27 19:45:54
10753	[habis, update, gak, cek, serifikat, detail, layan, covid, terkini...	2021-12-31 09:36:39
11001	[sangat, bagus]	2021-12-29 09:52:36
11418	[dihukum]	2021-12-25 10:53:04
13048	[terbaik, semangat]	2021-11-24 06:24:36
10689	[bagus]	2021-12-31 20:03:40
11675	[setting, yg, sdh, berubah]	2021-12-20 07:07:09
11455	[1, lihat, sertifikat, vaksin, lupa, vaksin, nya, 2, log, out, log...	2021-12-24 14:36:50
11564	[fitur, nya, aja]	2021-12-22 14:46:16
13682	[aplikasi, mendukung, handphone, android, berumur, diatas, 5, thn,...	2021-11-14 11:18:51

Gambar 8. Hasil Proses *Filtering*

5. *Stemming*

Pada penelitian kali ini terdapat skenario dan salah satu skenario tersebut melewati proses *stemming*. Proses *stemming* akan dilakukan penghapusan imbuhan pada kata dan mengubah kata menjadi kata dasar dengan menggunakan library Sastrawi. Tabel 1 merupakan contoh kata sebelum dan sesudah dilakukan *stemming*. Hasil proses stemming dapat dilihat pada gambar 10.

Tabel 1. Contoh Penerapan *Stemming*

Kata sebelum stemming	Kata sesudah stemming
Mendukung	Dukung
Ketinggalan	Tinggal
Dibilang	Bilang
Perjalanan	Jalan
Melihat	Lihat
Berubah	Ubah
Terdaftar	Daftar
Diatas	Atas

	content_stemming	at
11168	[sangat, bantu]	2021-12-27 19:45:54
10753	[habis, update, gak, cek, serifikat, detail, layan, covid, kini, g...	2021-12-31 09:36:39
11001	[sangat, bagus]	2021-12-29 09:52:36
11418	[hukum]	2021-12-25 10:53:04
13048	[baik, semangat]	2021-11-24 06:24:36
10689	[bagus]	2021-12-31 20:03:40
11675	[setting, yg, sdh, ubah]	2021-12-20 07:07:09
11455	[1, lihat, sertifikat, vaksin, lupa, vaksin, nya, 2, log, out, log...	2021-12-24 14:36:50
11564	[fitur, nya, aja]	2021-12-22 14:46:16
13682	[aplikasi, dukung, handphone, android, umur, atas, 5, thn, cth, re...	2021-11-14 11:18:51

Gambar 9. Hasil Proses *Stemming*

Transformation

Pada tahapan ini mengubah data ulasan komentar menjadi vektor angka menggunakan *Fasttext Wordembedding.* Penerapan *Fasttext Wordembedding* akan menggunakan fungsi yang telah disediakan oleh *library Gensim* yaitu *models.Fasttext.train* sebanyak dua kali sesuai jumlah skenario, yakni pada data yang dilakukan *stemming* dan data yang tidak dilakukan *stemming*.

Output dari *Fasttext WordEmbedding* adalah vektor berdimensi (n,). Pada penelitian ini nilai n yang digunakan adalah tiga ratus sesuai dengan nilai default dari *library Gensim* selain itu penulis menggunakan kedua *architecture* yakni *skipgram* dan *cbow* pada masing-masing skenario yang telah ditentukan. Sebagai

contoh pada gambar 11 merupakan *syntax* untuk proses *training fasttext* dan pada gambar 12 merupakan *syntax* untuk melakukan transformasi pada data ulasan, keduanya dengan menggunakan data *stemming*.

```
sentences = [tokens for tokens in tqdm(df["content_stemming"])]
model = FastText(sg=0, vector_size=300, workers=6)
model.build_vocab(sentences)
model.epochs = 200
model.train(sentences, total_examples=model.corpus_total_words, epochs=model.epochs,
            compute_loss=True, callbacks=[callback()])
model.save("model/cbow/model300_stemming.bin")
```

Gambar 10. Proses *Training Fasttext Cbow*

```
wv = model.wv
def simple_encode_sentence(sentence, w2v, stopwords=None):
    if stopwords is None:
        vecs = [w2v[word] for word in sentence]
    else:
        vecs = [w2v[word] for word in sentence if word not in stopwords]
    sentence_vec = np.mean(vecs, axis=0)
    return sentence_vec
vecs = [simple_encode_sentence(sentence, wv) for sentence in tqdm(df.content_stemming)]
vecs = np.array(vecs)
```

Gambar 11. Proses *Transformation*

Hasil proses *transformation* pada ulasan sangat panjang, misalkan ulasan "sangat bagus" setelah diubah kebentuk vektor, kata sangat akan mempunyai vektor sendiri berdimensi (300,) begitu juga dengan bagus. Sehingga dimensi dari ulasan "sangat bagus" adalah (300, 2). Sebagai contoh vektor kata "PeduliLindungi" dapat dilihat pada gambar 13, gambar 14, gambar 15 dan gambar 16.

```
array([-3.77618283e-01,  2.92605460e-01, -3.17217290e-01, -2.20138784e-02,
        7.59841144e-01,  3.47584113e-02, -3.09854418e-01,  1.95416376e-01,
       -1.88262448e-01,  3.50544572e-01, -5.11140883e-01, -2.13104799e-01,
       -6.69891059e-01, -3.19963068e-01, -1.94502786e-01,  1.97231770e-01,
       -5.47033548e-01, -2.54695028e-01, -7.30141759e-01,  9.44024175e-02,
        5.14105082e-01,  4.93742347e-01, -4.56359565e-01,  6.42818093e-01,
       -1.58741355e-01,  8.77839103e-02,  2.31774133e-02, -6.40506148e-01,
       -5.96270263e-01,  2.92183369e-01, -1.65154953e-02,  1.07575074e-01,
        3.82823497e-01,  2.05034137e-01, -3.66567492e-01,  1.58154920e-01,
        2.30823025e-01,  1.96088240e-01,  3.48240167e-01, -3.41044873e-01,
       -2.55442023e-01, -6.83446527e-02,  3.01188052e-01,  3.60040575e-01,
       -1.91461220e-01,  1.52940750e-01, -6.06640875e-01, -9.22143683e-02,
        1.88909918e-02, -2.41957471e-01, -4.44603264e-02,  1.27997652e-01,
       -2.20224917e-01, -4.06621397e-02,  3.97747397e-01,  4.85184988e-01,
       -2.96692569e-02, -2.90958602e-02,  1.86838377e-02,  1.88982949e-01,
       -4.92097586e-02,  4.87427413e-01, -2.29110047e-01,  9.19995308e-02,
       -2.70083994e-01,  1.04116499e-01, -5.02052784e-01,  5.87120771e-01,
       -3.61591190e-01, -4.26649630e-01,  2.87237734e-01,  2.23672941e-01,
       -5.59728257e-02,  6.15269691e-02, -6.69520870e-02,  5.97388804e-01,
        5.80741763e-01, -4.10785824e-01,  1.81689113e-01, -2.89934158e-01,
        1.46153077e-01, -5.79164922e-02, -2.64895558e-01,  1.12351581e-01,
       -6.77140653e-02,  7.39088282e-03,  2.63554662e-01, -1.48700834e-01,
        4.17158231e-02,  1.73840180e-01,  8.18538964e-02, -1.61511645e-01,
        5.30207276e-01,  9.68398079e-02, -6.42311648e-02,  5.87476254e-01,
       -1.74900591e-01,  3.14464182e-01,  5.14250845e-02, -6.03024244e-01,
       ...
       -2.99404502e-01, -1.96292356e-01, -5.35406709e-01, -4.13927227e-01,
       -2.59783603e-02, -1.52667714e-02,  4.17721868e-01,  5.76864332e-02,
        1.70210972e-01, -1.05013512e-01, -2.33589768e-01,  4.06450219e-02,
        3.59310746e-01, -1.40631840e-01,  8.98267552e-02,  5.56407273e-01],
      dtype=float32)
```

Gambar 12. *Vektor Kata PeduliLindungi Data Tanpa Stemming dengan Architecture CBOW*

```
array([-2.41187423e-01, -3.90954874e-02, -1.05505802e-01,  1.73360556e-01,
        8.81157145e-02, -1.53092176e-01, -2.38271862e-01, -8.99680853e-01,
       -8.71425152e-01,  1.36493528e-02, -3.17950279e-01, -5.97186029e-01,
        2.62530893e-02,  1.58543736e-01, -6.63308740e-01,  2.81748325e-01,
       -5.09449661e-01, -1.38611242e-01, -3.53969574e-01,  2.47140706e-01,
       -3.52838606e-01,  1.88633427e-01, -2.37027556e-01,  1.53981358e-01,
       -1.28764600e-01, -2.75251389e-01, -9.08193052e-01, -4.92755920e-02,
       -1.18443049e-01,  3.89899671e-01,  2.02715605e-01, -9.30849016e-02,
        6.08790480e-02,  2.60047466e-01, -7.09092617e-03,  6.46175593e-02,
       -6.23375535e-01, -4.65713488e-03,  3.12268734e-02, -2.80389547e-01,
       -6.29824162e-01, -2.68753707e-01, -3.61263126e-01,  6.92787111e-01,
       -8.35235417e-02,  4.54406083e-01,  1.71032310e-01, -6.49917006e-01,
        7.09322914e-02, -3.92076403e-01, -3.75926018e-01,  2.47628257e-01,
       -3.67572345e-02, -2.62124091e-02, -2.26792526e-02,  3.04789306e-03,
        3.55745666e-02, -1.63281500e-01,  1.18893415e-01,  5.78920007e-01,
        1.76355705e-01,  1.51349321e-01, -2.44545951e-01, -3.13536078e-01,
       -1.17626846e-01, -4.85908873e-02,  1.49603993e-01, -1.48170382e-01,
       -2.16597077e-02,  7.30670542e-02, -4.38334569e-02,  2.91667372e-01,
       -1.84453741e-01,  4.37903285e-01, -1.40725836e-01,  2.53368199e-01,
       -3.38962555e-01, -2.67128140e-01,  4.18058246e-01,  3.13809067e-02,
        1.36387914e-01, -1.89667102e-02,  9.12834927e-02,  3.36389363e-01,
        8.62700343e-02,  1.64868817e-01,  4.77168672e-02, -7.82305747e-02,
        1.55915692e-01, -1.70450881e-01,  9.28039104e-02, -2.29967579e-01,
       -1.93853378e-01, -3.06054652e-01,  2.47241035e-02, -1.39832228e-01,
       -5.10863513e-02,  4.62878466e-01,  1.64264470e-01, -2.30893075e-01,
       ...
       -2.30344340e-01, -8.35744888e-02, -1.35466531e-01, -1.76324576e-01,
        8.59302357e-02, -1.02375880e-01, -1.29452338e-02,  3.96191627e-01,
        3.70036304e-01, -2.31151327e-01, -9.27509665e-02,  3.15067053e-01,
        4.06975657e-01, -4.00363922e-01, -1.70438178e-02,  4.90843534e-01],
      dtype=float32)
```

Gambar 13. Vektor Kata PeduliLindungi Data *Stemming* dengan *Architecture CBOW*

```
array([-0.02921467,  0.28122932,  0.08878961, -0.28338164,  0.08929393,
       -0.00103734,  0.07771084,  0.14241885, -0.00809861,  0.12109866,
       -0.2779311 , -0.12277497, -0.09956637, -0.09183042, -0.1973225 ,
        0.05425942, -0.06759305,  0.3393004 ,  0.13235673, -0.04192852,
        0.14245161,  0.01445482, -0.24474174,  0.04349061,  0.2571183 ,
        0.0406233 ,  0.2549272 , -0.05989997, -0.25980708,  0.13664278,
       -0.04194231,  0.02466573,  0.0740727 , -0.0623523 , -0.04415405,
       -0.10580142, -0.10236765, -0.06100788,  0.1393188 , -0.11934748,
       -0.20687513, -0.04173228, -0.03947796,  0.03894548, -0.0997466 ,
        0.22489022, -0.23800682,  0.05483951,  0.15335491,  0.21232873,
       -0.02193235,  0.09258812, -0.17054649, -0.3272747 , -0.06746966,
        0.00591518,  0.01780232, -0.06265787,  0.01711154,  0.14332883,
        0.2312249 ,  0.12891495,  0.01761908,  0.13868691, -0.10348927,
       -0.05975169, -0.09592745,  0.0030231 , -0.1900484 , -0.03591866,
        0.24207956,  0.10170221, -0.083648  ,  0.06262375, -0.12330907,
        0.13241805, -0.22413349, -0.10873086,  0.05028035, -0.12229854,
        0.02995068,  0.1498902 , -0.07675326,  0.22295736, -0.1966729 ,
        0.18443988,  0.03060116,  0.16738888,  0.13073106,  0.02632899,
       -0.05906091, -0.04662152, -0.08075994, -0.14477488, -0.09450933,
        0.12665594, -0.06814748,  0.01904551,  0.17137624, -0.15415817,
       -0.05363683, -0.08712891, -0.00688202,  0.10142172,  0.16537385,
       -0.03802546,  0.14344177, -0.06902905, -0.12722842,  0.09540129,
       -0.00160406,  0.06921078,  0.03358058,  0.1604113 ,  0.04226235,
        0.02497088, -0.07237767, -0.14699201,  0.10518701,  0.0081682 ,
        0.10649948,  0.14956065, -0.28247073,  0.01015579, -0.02340234,
       ...
        0.18541017,  0.03319095,  0.21316983, -0.07332794, -0.27135792,
       -0.2489036 , -0.23100163,  0.14070082, -0.15858026,  0.18919551,
        0.0456036 , -0.04462582,  0.04110328,  0.09771133, -0.16653645,
       -0.03974074, -0.03292914, -0.10443422,  0.08418509, -0.05214289],
      dtype=float32)
```

Gambar 14. Vektor Kata PeduliLindungi Data Tanpa *Stemming* dengan *Architecture Skip Gram*

```
array([ 0.1851505 ,  0.35067764,  0.13889584, -0.1105586 ,  0.07913386,
        0.05096395, -0.05412698,  0.14471969, -0.0949545 ,  0.11950387,
       -0.29656702,  0.13899891, -0.08306404, -0.03432307, -0.14322487,
       -0.09128447,  0.29474732,  0.16875671,  0.09627079,  0.14820088,
        0.03928399, -0.06135125, -0.06586922,  0.11482556,  0.26980892,
        0.14540036,  0.19346741, -0.02441563,  0.06349449,  0.01223324,
        0.01483231,  0.1107349 ,  0.15442958, -0.10832415, -0.05574226,
       -0.06658792, -0.04096847, -0.07488931, -0.10684343,  0.01764476,
       -0.13351984,  0.00475065, -0.00957889, -0.08927947,  0.15448207,
        0.01510418, -0.15568852,  0.11742535,  0.32037205,  0.14768635,
       -0.0104772 ,  0.18936053, -0.05901895, -0.10597362, -0.07670602,
       -0.02900569,  0.06101946, -0.04771706,  0.08891583,  0.31398514,
        0.0633963 , -0.07675378, -0.0167177 ,  0.03725076, -0.09911071,
       -0.1184433 , -0.09738346,  0.20845707, -0.25170374,  0.03064316,
        0.02158413,  0.09430493, -0.01308865, -0.09886998, -0.15665856,
        0.0440721 , -0.35924625, -0.048419  ,  0.06512429,  0.18107864,
       -0.19706964,  0.1182285 , -0.1075025 ,  0.1374622 , -0.04952466,
        0.2376303 , -0.00978015,  0.16406871,  0.19064467, -0.13392454,
        0.04481178, -0.10645019, -0.10698982, -0.1779121 , -0.2855724 ,
        0.19955048,  0.00522591,  0.10654911,  0.01235917,  0.0547511 ,
       -0.2154169 , -0.22274533, -0.22092313, -0.08117765,  0.02457034,
        0.23568752, -0.10215883, -0.09938505, -0.12872672, -0.12096366,
        0.01249516,  0.01566556,  0.03136267,  0.25099725, -0.04917486,
       -0.12457574,  0.22091866, -0.10417601,  0.23884365,  0.0335961 ,
       -0.00336167,  0.22730152, -0.09928229,  0.06683553, -0.08690628,
       ...
       -0.012048  , -0.07450604,  0.29727614, -0.2543257 , -0.03053215,
        0.01742264, -0.14857239, -0.02124626, -0.07011709,  0.1449537 ,
        0.1372072 , -0.05107933, -0.01769362,  0.18003714, -0.00673115,
        0.02596211, -0.15411368, -0.03156608,  0.12660192, -0.07766955],
      dtype=float32)
```

Gambar 15. Vektor Kata PeduliLindungi Data *Stemming* dengan *Architecture Skip Gram*

Data Mining

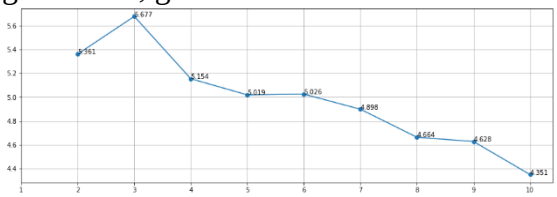
Di tahap ini melakukan *clustering* dengan algoritma K-Medoid sesuai dengan jumlah scenario dan jumlah transformasi serta tiga *distance measure* pada algoritma Kmedoids yaitu *euclidean*, *manhattan*, dan *cosine*. Metode yang digunakan sebagai acuan untuk memilih jumlah *cluster*, skenario dan *distance measure* terbaik adalah *davies bouldin index* atau disingkat DBI

Implementasi *clustering* pada penelitian ini akan menggunakan bahasa pemrograman *python* serta *library sklearn_extra* dan *sklearn*. Proses *clustering* akan dilakukan beberapa percobaan untuk mencari nilai *cluster* dengan nilai DBI paling mendekati nol. Sebagai contoh pada gambar 17 akan dilakukan *clustering* pada data yang melewati proses *stemming*

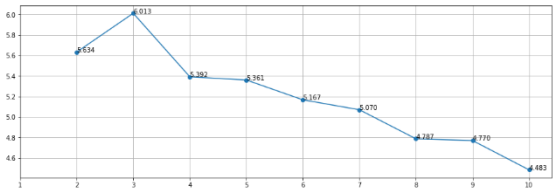
```
model_stemming_cbow = FastText.load("model/cbow/model300_stemming.bin")
wv = model_stemming_cbow.wv
vecs = [simple_encode_sentence(sentence, wv) for sentence in tqdm(df2.content_stemming)]
vecs = np.array(vecs)
dbi_Score = {}
for k in range(2, 11):
    m = KMedoids(n_clusters=k, random_state=0, metric="euclidean")
    m.fit(vecs)
    label = m.predict(vecs)
    dbi_Score[k] = davies_bouldin_score(vecs, label)
fig = plt.figure(figsize=(15, 5))
plt.plot(dbi_Score.keys(),dbi_Score.values(), marker="o")
plt.grid(True)
plt.xticks(range(1, 11))
plt.show()
```

Gambar 16. Proses *Clustering* dengan *Metrics Euclidean Distance*

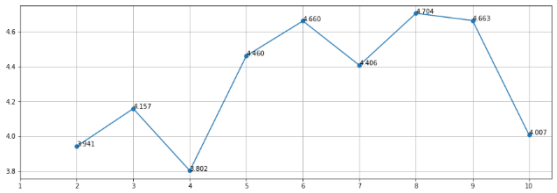
Nilai DBI dari beberapa percobaan *clustering* dengan jumlah *cluster* dimulai dari dua hingga sepuluh menggunakan data *stemming*, *architecture cbow* serta beberapa *distance measure* yaitu *euclidean distance*, *manhattan distance* dan *cosine distance* dapat dilihat pada gambar 17, gambar 18, gambar 19.



Gambar 17. Nilai DBI pada Percobaan *Metrics Euclidean Distance*



Gambar 18. Nilai DBI pada Percobaan M*etrics Manhattan Distance*



Gambar 19. Nilai DBI pada Percobaan *Metrics Cosine Distance*

Hasil yang didapat dari *clustering* data dengan *stemming* dengan menggunakan *architecture cbow* menghasilkan *metrics euclidean distance* dengan nilai *davies bouldin index* atau DBI paling mendekati nol. Hasil lengkap perbandingan nilai DBI dari seluruh percobaan dapat dilihat pada sub bab *Evaluation.*

Evaluation

Hasil yang diperoleh dari semua perocobaan *clustering* yang dilakukan akan dibandingkan menggunakan *davies bouldin index* atau disingkat DBI. Hasil perbandingan semua percobaan tersebut dapat dilihat pada tabel 2 dibawah ini.

Tabel 2. Tabel Perbandingan Nilai DBI

Skenario	*Architecture Transformations*	*Distance measure*	*cluster*	Nilai DBI
Data dengan *stemming*	*Skip Gram*	*Euclidean*	2	5.9
		Manhattan	10	6.36
		Cosine	10	5.85
	Cbow	*Euclidean*	10	4.35
		Manhattan	10	4.48
		Cosine	4	3.80
Data tanpa *stemming*	*Skip Gram*	*Euclidean*	9	6.24
		Manhattan	10	6.21
		Cosine	10	5.89
	Cbow	*Euclidean*	2	3.89
		Manhattan	2	2.93
		Cosine	10	3.64

Berdasarkan tabel 2 dapat dilihat bahwa hasil *clustering* dengan nilai DBI paling mendekati nol adalah dengan menggunakan data tanpa *stemming*, *architecture cbow*. Dalam penelitian ini

metrics distace manhattan menjadi yang terbaik dengan jumlah *cluster* dua. Pada gambar 20 dapat dilihat beberapa sampel ulasan yang sudah *cluster.*

	content	label
13048	berikan yang terbaik buat kita semua semangat	0
10689	bagus	0
11675	setting yg sdh di buat berubah terus	0
11455	1 bagaimana cara lihat sertifikat vaksin saya sendiri juga sudah l...	1
11564	fitur nya biasa aja	0
13682	aplikasi ini tidak mendukung handphone android berumur diatas 5 th...	0
12793	sudah selesai makan	1
13411	lambat kurang proaktif banyak yang sudah vaksin tapi tidak terdaftar	1
11321	aplikasi tdk bisa buka sertifikasi vaksin setelah diupdate mohon p...	1
13168	ka sejauh ini 2 bintang dulu ya nenden udah coba daftar tapi gagal...	1
10918	setelah update versi ini malah ga bisa buka sertifikat vaksin	1

Gambar 20. Sampel Data Ulasan yang Sudah Berlabel

SIMPULAN

Penelitian ini berhasil mengelompokkan *reviews* atau ulasan-ulasan aplikasi PeduliLindungi pada google playstore menggunakan algoritma Kmedoids serta menerapkan kedua *architecture Fasttext Word Embedding* pada proses *transformation* dengan menggunakan metodologi Knowledge Discovery in Database atau disingkat KDD yang terdiri dari data *selection*, data *preprocessing*, data *transformation*, *data mining* dalam penelitian ini adalah *clustering* dan *evaluation*.

Hasil pada proses *clustering* dan *evaluation* pada beberapa percobaan yang telah dipaparkan menghasilkan *clustering* dengan menggunakan data tanpa *stemming*, *architecture cbow* pada *transformation* dan menggunakan *metrics manhattan distance* menjadi skenario terbaik pada penelitian ini dengan jumlah *cluster* sama dengan dua yang memiliki nilai DBI yang paling mendekati nol yaitu 2,93. Kata terbanyak yang paling sering muncul pada *cluster* pertama yaitu kata "aplikasi" sebanyak 703 kata sedangkan kata yang paling sering muncul pada *cluster* kedua yaitu kata "vaksin" sebanyak 581 kata.

Adapun saran untuk penelitian selanjutnya yaitu :

1. Dapat menggunakan sumber data lain seperti cuitan pada media sosial.
2. Dapat menambahkan *spelling correction* pada tahap *preprocessing.*
3. Dapat mencoba beberapa percobaan dengan mengganti nilai dimensi pada vektor yang dihasilkan saat proses *transformation.*
4. bisa mencoba algoritma *clustering* yang lebih *advance* seperti penerapan algoritma berbasis *neural network.*
5. bisa menambahkan metode evaluasi lain seperti *Silhouette score, Caliski-Harabaz Score* atau yang lainya.

DAFTAR PUSTAKA

Chollisni, A., Syahrani, S., Dewi, S., Utama, A. S., & Anas, M. (2022). The concept of creative economy development-strengthening post covid-19 pandemic in Indonesia: Strategy and public policy management study. *Linguistics and Culture Review*, *6*, 413-426.

Farhadloo, M., & Rolland, E. (2016). Fundamentals of sentiment analysis and its applications. *Studies in Computational Intelligence*, *639*(August 2018), 1–24. https://doi.org/10.1007/978-3-319-30319-2_1

Gata, W. (2016). *Akurasi Text Mining Menggunakan Algoritma K-Nearest Neighbour pada Data Content Berita SMS*.

Haerani, E., & Rahmatulloh, A. (2021). Analisis User Experience Aplikasi Peduli Lindungi untuk Menunjang Proses Bisnis Berkelanjutan. *SATIN-Sains dan Teknologi Informasi*, *7*(2), 01-10.

Herdiana, D. (2021). Aplikasi peduli lindungi: perlindungan masyarakat dalam mengakses fasilitas publik di masa pemberlakuan kebijakan ppkm. *Jurnal Inovasi Penelitian*, *2*(6), 1685-1694.

Hertina, H., Nurwahid, M., Haswir, H., Sayuti, H., Darwis, A., Rahman, M.,

... & Hamzah, M. L. (2021). Data mining applied about polygamy using sentiment analysis on Twitters in Indonesian perception. *Bulletin of Electrical Engineering and Informatics*, *10*(4), 2231-2236.

Imberman, S. P. (2001). *Effective Use Of The Kdd Process And Data Mining For Computer Performance Professionals. Effective Use Of The Kdd Process And Data Mining For Computer Performance Professionals*. https://www.researchgate.net/publication/221445402

Karsito, & Sari, W. M. (2018). Prediksi Potensi Penjualan Produk Delifrance Dengan Metode Naive Bayes Di Pt. Pangan Lestari. *SIGMA –Jurnal Teknologi Pelita Bangsa*, 67–78.

Ndwandwe, D., & Wiysonge, C. S. (2021). COVID-19 vaccines. *Current opinion in immunology*, *71*, 111-116.

Nurdin, A., Anggo, B., Aji, S., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal TEKNOKOMPAK*, *14*(2), 74.

Roziqin, A., Mas'udi, S. Y., & Sihidi, I. T. (2021). An analysis of Indonesian government policies against COVID-19. *Public Administration and Policy*, *24*(1), 92-107.

Sabrina, A., Siregar, I., & Sosrohadi, S. (2021). Lingual Dominance and Symbolic Power in the Discourse of Using the PeduliLindungi Application as a Digital Payment Tool. *International Journal of Linguistics Studies, 1*(2), 52-59.

Yudiarta, N. G., Sudarma, M., & Ariastina, W. G. (2018). Penerapan Metode Clustering Text Mining Untuk Pengelompokan Berita Pada Unstructured Textual Data. *Majalah Ilmiah Teknologi Elektro*, *17*(3), 339. https://doi.org/10.24843/mite.2018.v17i03.p06