



Nama Formulir:

**Lembar
Pengesahan
Karya Ilmiah**

No.

FR-003/PR-003/KB-02-
01/MMP/UPM/2021

Issue/Revisi

Tgl Berlaku

Halaman

10 Juli 2021

Yang bertanda tangan di bawah ini:

1. Nama Lengkap Muhammad Darwis, M.Kom
Jabatan Dosen
Program Studi Teknik Informatika
NIP _____

Telah melakukan penelaahan karya ilmiah berjudul:

Analisis Sentimen Ulasan Aplikasi Mobile Banking M-Smile Menggunakan Metode Latent
Semantic Indexing (LSI)

Dan menyatakan bahwa karya ilmiah tersebut sudah memenuhi kaidah penulisan ilmiah dan oleh karenanya layak diajukan untuk keperluan unggah karya ilmiah di Universitas Paramadina, atas nama:

Nama Lengkap Retno Asih Murniati
Jenjang Sarjana
Program Studi Teknik Informatika
NIM 120103007

Demikian hasil penelaahan atas karya ilmiah ini dibuat dengan sesungguhnya untuk dapat dipergunakan bilamana diperlukan.

Jakarta, 30 Agustus 2023

Penelaah,

Muhammad Darwis, M.Kom

NIP: _____

Analisis Sentimen Ulasan Aplikasi *Mobile Banking* M-Smile Menggunakan Metode *Latent Semantic Indexing* (LSI)

Retno Asih Murniati¹

Program Studi Teknik Informatika, Fakultas Ilmu Rekayasa, Universitas Paramadina¹,

Email : ¹retno.murniati@students.paramadina.ac.id

Abstrak - Perkembangan teknologi di Indonesia memunculkan banyak aktivitas yang didukung dengan berbagai fasilitas digital salah satunya aplikasi *mobile banking* yang digunakan untuk melakukan transaksi pembelian maupun pembayaran. M-Smile merupakan aplikasi *mobile banking* dari PT. Bank Mega, Tbk yang memiliki *rating* 3,5 di Google Playstore. Untuk mengetahui apakah pengguna aplikasi *mobile banking* M-Smile ini puas atau tidak dengan fitur yang disediakan oleh aplikasi maka dilakukan analisis sentimen dari ulasan yang diberikan oleh pengguna. Dari ulasan dapat dilihat hasil dari sentimen yang digolongkan kedalam tiga kelas yaitu kelas positif, kelas negatif, dan kelas netral. Analisis sentimen dilakukan menggunakan metode *Latent Semantic Indexing* (LSI) yang memanfaatkan proses reduksi matriks *Singular Value Decomposition* (SVD) dimana matriks SVD merepresentasikan hubungan antara dokumen dan *term* dalam koleksi teks. Metode *Cosine Similarity* pada analisis sentimen ini digunakan untuk membagi penggolongan kelas yang dihitung dari kemiripan antar data. Berdasarkan hasil pengujian dihasilkan akurasi sebesar 81,67% dan dengan pembagian data ulasan positif, negatif, dan netral berturut-turut sebesar 31,67% untuk ulasan positif 13,33% untuk ulasan negatif dan 55% untuk ulasan netral sehingga dapat disimpulkan metode *Latent Semantic Indexing* (LSI) baik digunakan untuk mencari relevansi antar dokumen.

Kata kunci: Analisis sentimen, *Singular Value Decomposition*, *Latent Semantic Indexing*, *mobile banking*, *penggolongan kelas*

Abstract - The technology development in Indonesia supports numerous activities with various digital facilities, such as *mobile banking* for purchasing transactions or payments. M-Smile is a *mobile banking* application from PT. Bank Mega, Tbk. It has a *rating* of 3.5 stars on Google Play. To determine whether users of the *mobile banking* application are satisfied or not satisfied with the features provided by the application, an analysis of sentiment is conducted based on the reviews given by users. From the reviews, it can be observed that the sentiment results are categorized into three classes: positive, negative, and neutral. The sentiment analysis is performed using the *Latent Semantic Indexing* (LSI) method, which utilizes the *Singular Value Decomposition* (SVD) matrix reduction process. The SVD matrix represents the relationship between documents and terms in the text

collections. The cosine similarity method is used to calculate similarities between data within the categorized classes. Based on the testing results, an accuracy of 81,67% was achieved and with the data sharing of positive, negative, and neutral reviews, respectively, 31.67% for positive reviews, 13.33% for negative reviews, and 55% for neutral reviews, indicating that the Latent Semantic Indexing (LSI) method provides better results in determining relevance between documents.

Keywords: *Sentiment analysis, Singular Value Decomposition, Latent Semantic Indexing, mobile banking, categorized classes.*

I. PENDAHULUAN

Permasalahan yang sering dihadapi oleh pengembang perangkat lunak yaitu tidak dapat mengetahui secara langsung kepuasan pengguna terhadap fasilitas aplikasi. Informasi yang didapatkan dari pengguna dapat berupa masukan positif yang dapat digunakan untuk pengembangan selanjutnya. Salah satu contoh aplikasi yang menghadapi permasalahan ini adalah M-Smile, sebuah platform *mobile banking* yang dikeluarkan oleh PT. Bank Mega, Tbk. Aplikasi M-Smile menyediakan beragam fitur untuk memudahkan transaksi perbankan. Meskipun telah memperoleh peringkat 3,5 di Google Playstore dan diunduh sebanyak satu juta kali, evaluasi dari para pengguna menjadi faktor penting dalam mengarahkan langkah-langkah perbaikan aplikasi ini. Dengan mengetahui tingkat kepuasan dan masukan pengguna aplikasi, pengembang perangkat lunak dapat melakukan penyempurnaan aplikasi dengan tepat dan sesuai dengan kebutuhan pengguna. Berdasarkan kebijakan yang terdapat pada halaman Google Play, sistem peringkat aplikasi di Play Store serta representasi visual berbentuk diagram batang yang menunjukkan presentase ulasan bintang 1,2,3,4, dan 5, dihitung berdasarkan penilaian kualitas terkini yang diberikan oleh pengguna aplikasi. Ini berarti bahwa penilaian tidak diambil dari rata-rata nilai sepanjang riwayat ulasan pengguna, kecuali jika jumlah ulasan yang diberikan pada aplikasi tersebut sangat sedikit. Oleh karena itu, angka yang terlihat pada sistem peringkat tidak sepenuhnya mencerminkan tingkat kepuasan pengguna secara menyeluruh (Google Play, 2023).

Untuk mengetahui apakah pengguna aplikasi *mobile banking* M-Smile puas atau tidak dengan fitur yang disediakan oleh aplikasi maka dilakukan analisis sentimen dari ulasan yang diberikan oleh pengguna. Analisis sentimen adalah suatu langkah yang digunakan untuk mengekstrak data dalam bentuk teks guna untuk mendapatkan sebuah informasi didalamnya baik informasi positif maupun informasi negatif (Sari & Wibowo, 2019). Dengan ini klasifikasi sentimen juga membantu analisis intelegensi bisnis aplikasi dan merekomendasi sistem (Pang & Lee, 2008). Dari analisis sentimen perolehan informasinya dapat digunakan untuk membuat keputusan pengembangan aplikasi selanjutnya.

Analisis sentimen pada penelitian ini menggunakan metode *Latent Semantic Indexing* (LSI) yang dimana metode ini penting untuk *Search Engine Option* (SEO). Metode LSI memanfaatkan reduksi matriks *Singular Value Decomposition* (SVD) yang dimana merepresentasikan hubungan antara dokumen dan *term* dalam koleksi teks. Metode LSI dipilih untuk melakukan analisis sentimen karena dapat memperhitungkan sinonim, antonim, dan polisemi kata-kata dari dokumen. Metode LSI juga memperhitungkan hubungan antara dokumen dan konsep terkait sehingga cakupan analisis sentimen menjadi lebih luas.

II. TINJAUAN PUSTAKA

A. Analisis Sentimen

Analisis sentimen adalah sebuah langkah yang digunakan untuk mengevaluasi emosi dan sentimen yang dilakukan melalui tekstual. Untuk memperoleh sebuah analisis sentimen yaitu dengan mengekstrak dan mengolah data untuk memperoleh sebuah informasi. Analisis sentimen memegang peranan krusial dalam berbagai disiplin ilmu seperti manajemen, politik, ekonomi, dan sosial karena hasil analisis sentimen sangat dipengaruhi oleh pendapat atau opini yang ada (Septian, et al., 2019).

B. Term Frequency & Inverse Document Frequency (TF-IDF)

TF-IDF merupakan sebuah metode dalam *text mining* yang digunakan untuk melakukan sebuah pembobotan. Metode TF-IDF menggunakan dua faktor penting, yaitu frekuensi kata dalam dokumen dan inversi frekuensi kata dalam dokumen, yang digabungkan untuk menghasilkan skor yang relevan. Dalam analisis sentimen, *Term Frequency* (TF) digunakan untuk mengestimasi seberapa sering suatu kata muncul dalam sebuah teks (Septian, et al., 2019). Sedangkan, *Inverse Document Frequency* (IDF) mengukur seberapa umum atau jarang sebuah kata di seluruh dokumen. Untuk melakukan perhitungan TF-IDF dilakukan seperti Persamaan 1 berikut:

$$TF = \text{jumlah frekuensi kata} : \text{jumlah kata} \quad (1)$$

Untuk persamaan IDF dapat ditunjukkan oleh Persamaan berikut:

$$IDF = \log \frac{td}{df} \quad (2)$$

IDF merupakan kata yang dicari, *td* merupakan jumlah dokumen, dan *df* merupakan frekuensi kata yang terpilih.

C. Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) adalah sebuah teknik dimana digunakan untuk klasifikasi dengan memprediksi kelas dari hasil proses *training* (Alkadri, et al., 2018). Metode LSI menggunakan teknik dengan cara menghilangkan kata-kata yang dianggap tidak penting dari sebuah *dataset*. Pada metode LSI klasifikasi dilakukan dengan menggunakan garis pembatas untuk memisahkan opini positif dan opini negatif.

LSI merupakan metode yang menyusun informasi menjadi struktur sematik dengan metode temu balik informasi dimana memanfaatkan asosiasi kata-kata yang tersirat dalam sebuah dokumen (Fernando & Toba, 2020). Untuk mengetahui representasi dari LSI dapat ditunjukkan Persamaan 3 berikut.

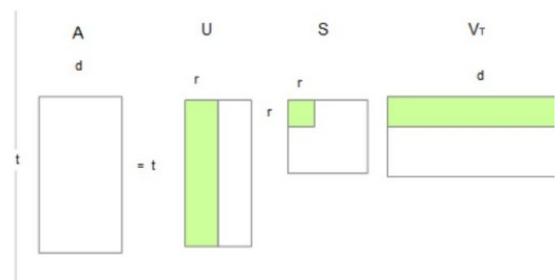
$$q' = q^T \cdot U_k \cdot S_k^{-1} \quad (3)$$

Sumber: (Alkadri, et al., 2018)

Dari persamaan diatas diketahui q' adalah kueri vektor, q^T adalah *transpose* dari metode TF-IDF. Untuk U_k merupakan reduksi dimensi k dari matriks U . S_k^{-1} merupakan *inverse* reduksi dimensi k matriks S (Sari, 2012).

D. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) adalah sebuah langkah yang digunakan untuk mereduksi dimensi matriks dan untuk mengurangi nilai kompleksitas dalam proses terms-dokumen matriks (Setiawan, et al., 2019). Pada metode SVD dekomposisi dibagi menjadi tiga buah matriks yaitu USV. Dekomposisi matriks dapat ditunjukkan oleh Gambar 1 berikut.



Gambar 1. Matriks SVD

Matriks SVD diurai menjadi tiga matriks pembentukannya seperti Gambar 1 diatas. Matriks ortogonal U adalah matriks dengan ukuran $(t \times r)$ adalah matriks yang terdiri dari t baris dan r kolom. Matriks ortogonal S adalah matriks dengan dimensi $(r \times r)$ yang

berisi nilai scalar. Matriks V adalah matriks dengan ukuran (dxr) yang merupakan *transpose* dari matriks ortogonal V .

E. Web Crawling

Web crawling merupakan cara yang digunakan untuk membantu proses ekstraksi data yang didapatkan dari berbagai sumber data (Fernando & Toba, 2020). Proses ini digunakan karena apabila menggunakan teknik manual akan memakan waktu yang cukup lama. Proses *web crawling* dilakukan dengan menggunakan pemrograman Python.

Proses *web crawling* ini dilakukan dengan mengumpulkan data dari sebuah halaman *web* kemudian didalam pemrograman akan diatur untuk menyimpan data yang dibutuhkan kedalam sistem lokal komputer. Proses *web crawling* menunjukkan sebuah kemampuan program yang dapat menavigasi halaman *web* secara mandiri (Fernando & Toba, 2020).

F. Python

Dalam dunia pemrograman, Python digunakan secara luas untuk mengembangkan aplikasi, melakukan analisis data, dan menjalankan perintah komputer. Untuk melakukan pemrograman melalui Python ada beberapa peraturan yang harus dipenuhi akan tetapi pemrograman Python memiliki *syntax* yang tidak terlalu rumit (Mahawardana, et al., 2022). Kegunaan Python lainnya yaitu membantu dalam bidang pekerjaan matematika, digunakan untuk *machine learning*, *web development*, dan untuk *software development*.

G. Cosine Similarity

Deviyanto dan Wahyudi (2018) menyatakan bahwa untuk menghitung kemiripan setiap data latih maka diperlukan *cosine similarity*. *Cosine similarity* menggunakan pemodelan dua vektor dan kemudian jarak antara dua vektor tersebut diukur. Apabila nilai yang dihasilkan vektor tersebut yang berupa sudut semakin kecil maka kesamaan antara data latih semakin besar (Wahid & SN, 2016). Untuk menghitung *cosine similarity* digunakan Persamaan 4 berikut.

$$\text{similarity} = \cos(\theta) \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (4)$$

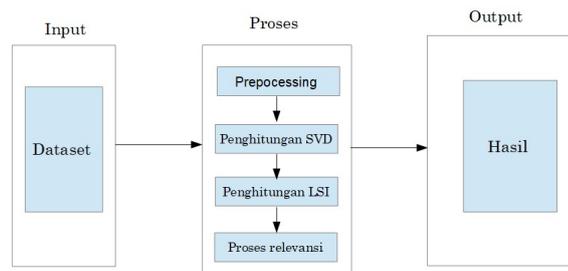
Dari Persamaan 4 diatas diketahui A merupakan hasil perhitungan dari LSI dan B merupakan nilai dari matriks SVD. Sedangkan $\|A\|$ merupakan panjang vektor A dan $\|B\|$ merupakan panjang vektor B .

III. METODE PENELITIAN

A. Perancangan Sistem

Dalam perancangan sistem untuk analisis sentimen menggunakan metode LSI ada beberapa proses dan tahapan. Untuk proses pertama yaitu input untuk

melakukan proses sistem. Perancangan sistem dapat dilihat pada Gambar 2 berikut ini.



Gambar 2. Model Rancangan Sistem

Adapun penjelasan dari model perancangan sistem diatas sebagai berikut:

Dataset

Pada tahap ini dilakukan proses pengambilan data yang berupa ulasan dari pengguna aplikasi *mobile banking*. Pengambilan dataset dilakukan dengan *web crawling* pada halaman *website mobile banking* M-Smile di Google Playstore. Dataset yang diambil berupa ulasan pengguna yang menggunakan bahasa Inggris dan semua data yang terkumpul disimpan dalam *file csv*.

Preprocessing

Setelah dataset terkumpul tahap selanjutnya yaitu *preprocessing*. *Preprocessing* dilakukan menggunakan kumpulan komentar hasil dari *web crawling*. Pada tahap *preprocessing* yang pertama dilakukan *case folding* yaitu mengubah semua huruf dan kata-kata menjadi huruf kecil. Selanjutnya, proses *tokenizing* yaitu memisahkan kalimat menjadi kata-kata. Proses *tokenizing* kalimat akan dipisah menjadi kata-kata berdasarkan tanda ‘ ‘ pada tahap *tokenizing* ini juga dilakukan proses pembuangan tanda baca. Proses *preprocessing* selanjutnya yaitu *stop word removal* menghilangkan kata penghubung atau kata yang dianggap tidak penting. Proses terakhir dari *preprocessing* yaitu *stemming*. Dalam upaya untuk mengurangi variasi kata-kata, *stemming* digunakan untuk mengubah kata-kata menjadi bentuk dasar yang lebih umum.

Perhitungan SVD

Pada proses perhitungan SVD dilakukan terlebih dahulu proses TF-IDF untuk mendapatkan pembobotan kata dari setiap dokumen. Pada proses TF-IDF dilakukan proses ekstraksi fitur untuk mendapatkan nilai vektor dari data hasil *preprocessing* yang telah dilabeli. Selanjutnya, proses perhitungan SVD untuk mendapatkan nilai matriks *term* dokumen.

Perhitungan LSI

Pada perhitungan LSI menggunakan dekomposisi dari SVD untuk mengurangi dimensi matriks *term-dokumen*. Selanjutnya, dengan

menggunakan varian pada setiap komponen utama akan menunjukkan seberapa baik komponen tersebut menjelaskan variasi data asli. Lalu dengan akumulasi varian dari komponen terpilih akan dapat diperkirakan seberapa banyak varian yang berhasil dipertahankan dan nilai ini menjadi indikator dalam representasi LSI.

Proses Relevansi

Pada proses relevansi didapatkan dari hasil *cosine similarity*. Perhitungan dari *cosine similarity* akan digunakan untuk mengukur presisi, akurasi, dan *recall* dari proses klasifikasi.

IV. Hasil dan Pembahasan

Hasil penelitian analisis sentimen dengan menggunakan data ulasan dari Google Play sebanyak tiga ratus ulasan yang telah dilabeli. Data diambil menggunakan *library* dari python *google_play_scraper*. Pada Gambar 3 merupakan *source code* untuk mengambil data ulasan sebanyak tiga ratus dengan ulasan hanya berbahasa Inggris.

```
from google_play_scraper import Sort, reviews
result, continuation_token = reviews(
    'com.bankmega.msml',
    lang='en',
    country='id',
    sort = Sort.MOST_RELEVANT,
    count = 300,
    filter_score_with=None
)
```

Gambar 3. Fungsi Python *Scraping* Ulasan

Hasil dari *scraping* menggunakan fungsi diatas berupa tiga ratus ulasan. Berikut contoh hasil data ulasan yang diambil yang ditunjukkan pada Gambar 4.

userName	userImage	content	score
azareru lolaz	https://play-lh.googleusercontent.com/a-/ACB-R...	At first it easy to used but when i comeback a...	1
F. C. P	https://play-lh.googleusercontent.com/a-/ACB-R...	Can't even login. It says choose product, I ta...	1
Novi AH	https://play-lh.googleusercontent.com/a/AAcHTt...	Already updated the newest one but still can n...	1
Panji Poluan	https://play-lh.googleusercontent.com/a-/ACB-R...	Just updated the app and it does	1

Gambar 4. Contoh Hasil *Scraping* Data Ulasan

Dari data diatas diambil kolom bagian content untuk disimpan kedalam *file csv* dan *download* ke lokal komputer kemudian dilakukan proses *preprocessing*.

Pada proses *preprocessing* data hasil dilakukan *cleaning* untuk menghilangkan karakter yang tidak diinginkan. Selanjutnya, *preprocessing* dilakukan proses *case folding* yang bertujuan untuk menyamakan semua huruf dalam kalimat menjadi huruf kecil. Setelah dilakukan proses *case folding* selanjutnya dilakukan proses *tokenizing* yaitu untuk memisahkan kalimat menjadi kata-kata. Setelah proses *tokenizing* dilakukan selanjutnya melakukan proses *stop word removal* yaitu penghilangan kata-kata yang dianggap tidak signifikan atau tidak memberikan kontribusi makna yang penting dalam kalimat. Proses terakhir *preprocessing* yaitu *stemming* yaitu mengubah kata-kata menjadi bentuk dasarnya. Pada Gambar 5 akan ditunjukkan contoh data hasil dari proses *preprocessing*.

content
At first it easy to used but when i comeback a...
Can't even login. It says choose product, I ta...
Already updated the newest one but still can n...
Just updated the app and it does not work anym...
I tried to log in with fingerprints but consta...
Preprocessed Text
first easi use comeback devic broke cant open...
ca nt even login say choos product tap credi...
alreadi updat newest one still work app forc ...
updat app work anymor respons get cs network...
tri log fingerprint constantli fail honestli ...

Gambar 5. Contoh Hasil *Preprocessing*

Dapat dilihat dari Gambar 5 contoh hasil dari proses *preprocessing*. Pada kolom content merupakan ulasan yang diambil dari Google Playstore sedangkan pada kolom *Preprocessed Text* merupakan hasil dari semua proses *preprocessing*. Hasil dari *preprocessing* masih terdapat kesalahan dari ejaan yang sehingga pada proses *labelling* data menggunakan *library* *TextBlob*. Di dalam *library* *TextBlob* terdapat koreksi untuk ejaan termasuk dalam penulisan bahasa Inggris.

Selanjutnya, proses *labelling* data dimana data yang telah melalui proses *preprocessing* diberi label positif, negatif, atau netral. *Labelling* data dengan menggunakan kata positif, negatif, dan netral merupakan jenis *labelling* data diskrit. Hasil dari proses *labelling* data dapat ditunjukkan pada Gambar 6 berikut.

Preprocessed Text	Label
first easi use comeback devic broke cant open...	positif
ca nt even login say choos product tap credi...	positif
alreadi updat newest one still work app forc ...	netral
updat app work anymor respons get cs network...	netral
tri log fingerprint constantli fail honestli ...	negatif

Gambar 6. Hasil Proses *Labelling* Data

Proses dari *labelling* sebelum menentukan positif, negatif, atau netral terdapat proses pembobotan nilai dengan menggunakan *sentiment polarity*. Dalam pembobotan nilai setiap kalimat akan dihitung nilai *polarity*-nya. Untuk mengklasifikasikan nilai *polarity* apabila nilai *polarity* lebih dari nol kalimat tersebut termasuk dalam kategori positif. Apabila nilai *polarity* kurang dari nol maka kalimat tersebut masuk kedalam kategori negatif. Dan apabila kalimat nilai *polarity*-nya sama dengan nol maka termasuk kedalam kategori netral.

Setelah melalui proses *labelling* data selanjutnya adalah proses membagi data uji dengan data latih. Dalam pembagian tersebut terdapat X_{train} , X_{test} , y_{train} , dan y_{test} . Pembagian tersebut X merupakan variabel yang dari data yang akan dibagi. Sedangkan, untuk y merupakan vektor target yang sesuai dengan matriks fitur X . X_{train} merupakan variabel yang akan dibagi yang digunakan untuk pelatihan model. X_{test} merupakan variabel yang akan dibagi yang digunakan untuk pengujian model. Sedangkan, untuk y_{train} merupakan vektor target yang digunakan pelatihan model. Lalu untuk y_{test} merupakan vektor target yang digunakan untuk pengujian model. Untuk total data yang akan dilatih dan diuji menggunakan tiga ratus ulasan yang telah dilabeli lalu data tersebut dibagi menjadi 20% total data sebagai pengujian dan 80% total data sebagai data pelatihan. Untuk mengontrol pembagian data secara acak menggunakan parameter `random_state` dan telah diatur sebesar 70 yang artinya menggunakan nilai 70 sebagai biji (*seed*) untuk menghasilkan pembagian data secara acak. Fungsi `random_state` juga bertujuan untuk memastikan reproduktibilitas hasil saat membagi data dan berguna untuk memastikan hasil yang konsisten dan dapat dikembangkan kembali saat kode dijalankan berulang.

Setelah proses pembagian data selanjutnya yaitu mengekstraksi fitur menggunakan metode TF-IDF yang dimana hasil dari metode TF-IDF digunakan untuk dekomposisi matriks SVD. Dalam penghitungan matriks SVD menggunakan sebuah fungsi dari *library* `scikit-learn` yaitu `TruncatedSVD` yang digunakan untuk melakukan dekomposisi nilai singular value (SVD) dan metode `TruncatedSVD` adalah varian dari SVD yang digunakan untuk matriks yang berdimensi besar. Selanjutnya, didalam fungsi `TruncatedSVD` terdapat `n_components = 60` yang artinya untuk menentukan jumlah komponen utama yang akan diekstraksi dari matriks term-dokumen dan dalam hal ini nilai 60 dipilih

untuk mendapatkan 60 komponen utama dari matriks term-dokumen. Dalam dekomposisi SVD memilih jumlah komponen utama berguna untuk mengurangi dimensi matriks, sehingga dapat mengompresi informasi tetapi tetap mempertahankan sebagian besar variasi dari data. Untuk hasil dari TF-IDF dan SVD ditunjukkan oleh Gambar 7 dan Gambar 8 berikut ini.

0	(0, 68)\t0.31752138976009786\n (0, 648)\t0....
1	(0, 696)\t0.14780924067323994\n (0, 411)\t0...
2	(0, 650)\t0.22742078581404573\n (0, 537)\t0...
3	(0, 544)\t0.22100884904868834\n (0, 89)\t0....
4	(0, 805)\t0.18159421498178357\n (0, 95)\t0....
...	...
295	(0, 715)\t0.6934756123738632\n (0, 547)\t0....
296	(0, 762)\t0.4386925414170435\n (0, 685)\t0....
297	(0, 590)\t0.4668519149216445\n (0, 773)\t0....
298	(0, 498)\t0.7882394107521238\n (0, 771)\t0....
299	(0, 332)\t0.6445918341815436\n (0, 483)\t0....

Gambar 7. Hasil TF-IDF

```
array([[ 0.24003004,  0.19900827,  0.02132545, ..., -0.01385016,
        -0.0523691, -0.04908969],
       [ 0.15929637, -0.02194593, -0.11806501, ...,  0.03014372,
         0.05615753,  0.11936954],
       [ 0.47655888,  0.01807337,  0.04769792, ..., -0.00440923,
        -0.07429927,  0.01523533],
       ...,
       [ 0.29356099, -0.07457999, -0.44107969, ..., -0.03286969,
         0.04025315,  0.05852851],
       [ 0.07873556,  0.43792623, -0.02862841, ..., -0.02188223,
         0.06937043,  0.0290119 ],
       [ 0.02512593,  0.23990581, -0.00350899, ...,  0.01357565,
         0.02193468,  0.01968629]])
```

Gambar 8. Hasil Dekomposisi Matriks SVD

Setelah mendapatkan nilai dari SVD selanjutnya menghitung akurasi dari metode LSI. Setelah mengetahui nilai akurasi lalu melakukan klasifikasi menggunakan metode *cosine similarity*. Setelah dilakukan klasifikasi selanjutnya menghitung nilai *recall*, presisi, dan akurasi klasifikasi. Setelah melakukan penghitungan nilai *recall*, presisi, dan akurasi klasifikasi selanjutnya menghitung banyaknya presentase pembagian komentar positif, negatif, dan netral. Dan proses yang terakhir yaitu menghitung nilai *True Positive Rate* dan *True Negative Rate*. Pengujian *True Positive Rate* mengukur keberhasilan dalam mengidentifikasi data uji positif dengan benar, dinyatakan dalam bentuk presentase. Sedangkan untuk *True Negative Rate* mengukur keberhasilan dalam mengidentifikasi data uji negatif dengan benar dan dinyatakan dalam bentuk presentase. Untuk melihat hasil keseluruhan dapat ditunjukkan oleh Gambar 9 berikut.

Akurasi: 81.67%
Presisi: 0.75
Recall: 0.75
Akurasi: 0.75
True Positive Rate (TPR): 90.00%
True Negative Rate (TNR): 80.00%
Persentase Kelas Positif: 31.67%
Persentase Kelas Negatif: 13.33%
Persentase Kelas Netral: 55.00%

Gambar 9. Hasil Akurasi LSI

Penjelasan dari Gambar 7 adalah nilai akurasi dari metode LSI yaitu sebesar 81,67%. Selanjutnya, nilai klasifikasi menggunakan *cosine similarity* nilai *recall*, presisi, dan akurasi dihasilkan berturut-turut sebesar 0,75 untuk nilai *recall* 0,75 untuk nilai presisi dan 0,75 untuk akurasi. Selanjutnya, untuk pembagian data ulasan positif, negatif, dan netral berturut-turut sebesar 31,67% untuk ulasan positif 13,33% untuk ulasan negatif dan 55% untuk ulasan netral. Lalu untuk nilai *True Positive Rate* dan *True Negative Rate* yang dihasilkan sebesar 90,00% dan 80,00%.

V. KESIMPULAN

Penggunaan metode *Latent Semantic Indexing* (LSI) dapat membantu dalam menganalisis sentimen ulasan yang diberikan oleh pengguna aplikasi M-Smile. Akurasi dari metode LSI dengan menggunakan tiga ratus ulasan berbahasa Inggris yang telah dilabeli mencapai 81,67%.

Dalam pembagian presentase ulasan yang telah dihitung nilai polaritasnya untuk menentukan kategori ulasan positif, ulasan negatif, dan ulasan netral didapatkan 31,67% ulasan positif dari pengguna, 13,33% ulasan negatif dari pengguna, dan sebanyak 55% merupakan ulasan netral dari pengguna.

Dalam penghitungan nilai klasifikasi menggunakan *cosine similarity* didapatkan nilai *recall*

sebesar 0,75 nilai presisi sebesar 0,75 dan nilai akurasi klasifikasi sebesar 0,75. Sedangkan, untuk nilai *True Positive Rate* dan *True Negative Rate* didapatkan nilai sebesar 90,00% dan 80,00%.

REFERENSI

- [1] P. B & L. L., "Thumbs Up? Sentiment Classification using Machine Learning Techniques," 2008.
- [2] F. V. Sari & A. Wibowo, "Analisis Sentimen Pelanggan Toko Online JD.ID Menggunakan Metode Naive Bayes Classifier Berbasis Ikon Emosi," *Jurnal Simetris*, 2019, doi: [https://doi.org/10\(1\),%20xx-xx](https://doi.org/10(1),%20xx-xx).
- [3] J. A. Septian & T. M. Fahrudin, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepkabolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *Journal of Intelligent Systems and Computation*, 2019.
- [4] F. D. Alkadri & Y. A. Sari, "Analisis Sentimen Ulasan Video Animasi Menggunakan Metode Latent Semantic Indexing," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, pp. 8091–8097, 2019.
- [5] E. H. Fernando & H. Toba, "Pemanfaatan Latent Semantic Indexing untuk Mengukur Potensi Kerjasama Jurnal Ilmiah Lintas Universitas," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, 2020, doi: <https://doi.org/10.28932/jutisi.v6i3.2894>.
- [6] P. P. Mahawardana and G. A. Sasmita, "Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap 'Figure Pemimpin' Menggunakan Python," *Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, 2022.
- [7] A. Deviyanto & M. D. Wahyudi, "Penerapan Analisis Sentimen pada Pengguna Twitter Menggunakan Metode K-Nearest Neighbor," *Jurnal Informatika Sunan Kalijaga*, vol. 3, 2018.
- [8] D. H. Wahid & S. A., "Peringkasan Sentimen Ekstraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity," *IJCCS*, vol. 10, pp. 207–218, 2018.
- [9] Sari & Ridok, "Penentuan Lirik Lagu Berdasarkan Emosi Menggunakan Sistem Temu Kembali Informasi dengan Metode Latent Semantic Indexing (LSI)," *Dalam Seminar Nasional Teknologi Informasi & Komputasi*, pp. 73–79, Nov. 2012.
- [10] Google Play, "Cara Penghitungan Rating". *Google Play*. Diakses 23 Agustus 2023, dari https://play.google.com/intl/id_id/about/comment-posting-policy/.