

Implementasi *Text Summarization* pada *Review Aplikasi Super* di *Google Play Store* Menggunakan Metode *Maximum Marginal Relevance*

Dion Alexander Louis, Silvia Rostianingsih, Leo Willyanto Santoso
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658

Email: dionalexander0@gmail.com, silvia@petra.ac.id, leow@petra.ac.id

ABSTRAK

Aplikasi Super adalah aplikasi untuk agen reseller yang menjual dan mendistribusikan sembako di kota-kota tingkat 2, 3 dan pedesaan Indonesia. Aplikasi Super sudah didownload sekitar 50.000 pengguna pada Play Store. Berbagai ulasan atau review juga telah diberikan oleh para pengguna yang telah mendownload aplikasi Super. Disadari atau tidak, opini-opini/ulasan pelanggan yang diberikan pada Google play, sedikit atau pun banyak, akan memberikan pengaruh pada calon pelanggan. Berdasarkan permasalahan yang terjadi, maka dalam penelitian ini akan dilakukan penerapan program peringkasan teks pada review Aplikasi Super dengan implementasi metode MMR dan TF-IDF, agar dari banyaknya jumlah review yang ada, dapat diekstrak beberapa kalimat penting saja, agar proses pengambilan kesimpulan akan menjadi lebih mudah. Hasil penelitian menggunakan metode MMR menghasilkan nilai rata-rata presisi 40,4% pada 3 kali percobaan, dan dengan nilai presisi tertinggi yaitu 60,4% pada percobaan menggunakan nilai parameter $\lambda = 0,7$.

Kata Kunci: *maximum marginal relevance, TF-IDF, cosine similarity, peringkasan review*

ABSTRACT

Super App is an app for reseller agents who sell and distribute basic necessities in tier 2, 3 cities and rural Indonesia. The Super app has been downloaded by around 50,000 users on the Play Store. Various reviews or reviews have also been given by users who have downloaded the Super application. Whether we realize it or not, customer opinions / reviews given on Google play, a little or a lot, will have an influence on potential customers. Based on the problems that occur, this research will implement a text summarization program on Super App reviews with the implementation of the MMR and TF-IDF methods, so that from the large number of existing reviews, only a few important sentences can be extracted, so that the conclusion making process will become easier. The results of the research using the MMR method produced an average precision value of 40.4% in 3 trials, and with the highest precision value of 60.4% in the experiment using the parameter value $\lambda = 0.7$.

Keywords: *maximum marginal relevance, TF-IDF, cosine similarity, text summarization*

1. LATAR BELAKANG

Aplikasi Super adalah aplikasi untuk agen reseller yang menjual dan mendistribusikan sembako di kota-kota tingkat 2, 3 dan

pedesaan Indonesia. Aplikasi Super memiliki persaingan yang ketat dengan kompetitornya, seperti Ula dan Warung Pintar. Agar dapat unggul dalam persaingan, Aplikasi Super harus terus meningkatkan kinerjanya agar dapat memuaskan penggunanya. Aplikasi Super sudah didownload sekitar 50.000 pengguna pada Play Store. Berbagai ulasan atau review juga telah diberikan oleh para pengguna yang telah mendownload aplikasi Super. Disadari atau tidak, opini-opini/ulasan pelanggan yang diberikan pada Google play, sedikit atau pun banyak, akan memberikan pengaruh pada calon pelanggan. Akan tetapi, memantau dan mengorganisasi opini dari masyarakat juga bukanlah hal yang mudah. Tidak hanya itu ulasan atau komentar review yang diberikan pengguna juga sangat penting bagi pengembang aplikasi dalam mencari tahu apa kekurangan dari aplikasi sehingga dapat meningkatkan kinerja aplikasi berdasarkan dari ribuan komentar tekstual pengguna. Ulasan dari pengguna sering digunakan sebagai alat yang efektif dan efisien dalam menemukan informasi terhadap suatu produk atau jasa. Penelitian baru-baru ini menemukan hampir 50% dari pengguna internet bergantung pada rekomendasi word-of-mouth (opini) sebelum menggunakan suatu [8]. Untuk dapat memperoleh informasi yang dibutuhkan dari banyaknya komentar yang ada diperlukan sebuah metode atau teknik khusus yang mampu meringkas semua komentar-komentar pengguna secara otomatis, pada permasalahan ini peringkasan teks atau Text Summarization dinilai cocok karena dengan meringkas teks dapat diperoleh inti sari dari kumpulan review pengguna. Text summarization atau peringkasan teks merupakan salah satu teknik pada bidang text mining yang bertujuan untuk mengurangi panjang dan detail dokumen dengan mempertahankan poin-poin terpenting dan makna umum [15]. Beberapa penelitian terdahulu terkait text summarization dilakukan salah satunya penelitian yang dilakukan oleh [10] membahas mengenai peringkasan kumpulan berita secara otomatis. Metode yang digunakan dalam penelitian ini adalah *maximum marginal relevance* (MMR), metode ini digunakan untuk meringkas dokumen dengan menghitung kesamaan (similarity) antar bagian teks. Hasil dari penelitian ini menunjukkan bahwa metode MMR dapat digunakan untuk meringkas multi dokumen dengan menggunakan kata kunci (masukan dari user) sebagai query. Berdasarkan permasalahan yang ada maka dilakukan penelitian dengan judul “Implementasi Text Summarization pada Review aplikasi Super di Google Play Store menggunakan metode *maximum marginal relevance*”. Penggunaan MMR akan menghasilkan ringkasan yang dimodelkan sedemikian rupa sehingga isi ringkasan yang dihasilkan harus terdiri dari informasi yang relevan dengan kueri serta kesamaan minimal di antara konten [14]. Hasil ringkasan menggunakan MMR akan menghasilkan nilai akurasi, presisi, recall dan F-1 Score. Tujuan dari penelitian ini adalah melakukan peringkasan teks pada review aplikasi Super

untuk mendapatkan informasi terkait peluang apa yang dapat ditingkatkan pada layanan aplikasi berdasarkan dari komplain atau keluhan review yang diberikan pengguna di Google Play.

2. PENELITIAN SEBELUMNYA

2.1 Peringkasan Review Konsumen Restoran Menggunakan Weighted Frequent Itemset Mining

Yusron, Bachtiar, & Fauzi [15] melakukan penelitian dengan masalah yang diangkat adalah jumlah review yang besar menyulitkan calon konsumen dan pemilik untuk mengetahui pendapat orang lain terhadap restoran. Metode yang diusulkan dari penelitian ini adalah *weighted frequent itemset mining* untuk melakukan peringkasan *review* konsumen restoran. Metode ini melakukan peringkasan dengan memilih kalimat yang mengandung banyak frequent itemset dan nilai relevansi kalimat yang tinggi. Hasil dari penelitian ini adalah menunjukkan bahwa peringkasan dengan menggunakan metode *weighted frequent itemset mining* menghasilkan nilai rata-rata F-measure sebesar 0.279. Penggunaan tahapan penghapusan stopword yang ada dalam preprocessing juga berpengaruh terhadap hasil ringkasan. Hasil ringkasan tanpa penghapusan stopword diketahui memberikan hasil yang lebih baik dibandingkan kondisi sebaliknya. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah penggunaan metode, skripsi yang akan dilakukan menggunakan metode MMR sedangkan penelitian ini menggunakan *weighted frequent itemset mining*.

2.2 Text Summarization pada Berita Bola Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python

Rifano, et al [5] melakukan penelitian dengan masalah yang diangkat adalah teks yang terlalu panjang yang disajikan pada artikel berita online membuat pembaca kesulitan memahami isi berita. Penelitian ini memanfaatkan library NTLK pada pemrograman Python. Hasil dari penelitian yang dilakukan penggunaan library NLTK dapat mempermudah pembaca dalam menemukan informasi yang paling penting dalam sebuah berita bola. Terdapat 15 kalimat teks asli yang setelah dilakukan peringkasan dengan library NLTK memperoleh hasil ringkasan menjadi 6 kalimat inti berita. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah penggunaan metode dalam menyelesaikan masalah, skripsi yang akan dilakukan menggunakan metode *maximum marginal relevance* sedangkan penelitian ini memanfaatkan library NTLK pada pemrograman python.

2.3 Extractive Text Summarization pada Berita Berbahasa Indonesia Menggunakan Algoritma Support Vector Machine

Aulia, Jamaludin, & Padilah [1] melakukan penelitian dengan masalah yang diangkat adalah berdasarkan data dari Programme for International Student Assessment (PISA) untuk tahun 2000 – 2018, Indonesia memiliki minat baca yang rendah. Akibat dari rendahnya minat membaca sering sekali ditemukan masyarakat kesulitan untuk memahami isi teks bacaan terutama jika teks panjang dan banyak. Metode yang digunakan dalam penelitian ini adalah algoritma *support vector machine* dengan ekstraksi fitur berdasarkan fitur kalimat, dengan ekstraksi fitur berdasarkan fitur kalimat poin utama pada berita akan diambil tanpa mengubah isi

bacaan. Hasil dari penelitian yang dilakukan adalah dengan skenario perbandingan data uji dan data latih 7:3, menggunakan kernel linear, dengan hasil *accuracy* 72,4%, *precision* 63,49%, *recall* 51,9%, dan *f-measure* 57,1%. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah ini adalah penggunaan metode dalam menyelesaikan masalah, skripsi yang akan dilakukan menggunakan metode *maximum marginal relevance* sedangkan penelitian ini menggunakan SVM. Pada tahap *preprocessing* penelitian ini menggunakan *case folding*, *tokenizing*, *stopword removal*, *stemming*, sedangkan pada skripsi yang akan dilakukan menggunakan *segmentasi*, *case folding*, *tokenizing*, *filtering*.

2.4 Automatic Text Summarization Menggunakan Metode Graph dan Ant Colony Optimization

Setyadi, Krisne, & Suyadnya [11] melakukan penelitian dengan masalah yang diangkat adalah sistem temu kembali dokumen merupakan sistem yang dapat mencari dokumen berdasarkan kata kunci, namun sistem temu kembali dokumen akan menemui kendala jika harus mencari informasi spesifik dari dokumen yang jumlahnya banyak dengan jumlah teks penyusun dokumen dengan jumlah yang tidak sedikit. Metode yang digunakan dalam penelitian ini adalah *graph* dan *ant colony optimization*. Hasil dari penelitian ini adalah berdasarkan uji coba yang telah dilakukan menunjukkan bahwa ringkasan sistem memiliki kemiripan 78,43% dibandingkan dengan ringkasan ahli, dengan tingkat kompresi 78,2%. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah pada penelitian ini menggabungkan dua metode yaitu *graph* dan *ant colony optimization* sedangkan skripsi yang akan dilakukan menggunakan MMR. Pada tahap *preprocessing* penelitian ini hanya menggunakan *stemming*.

2.5 Peringkasan Kumpulan Berita Secara Otomatis Menggunakan Metode Maximum Marginal Relevance

Setiadi, Djamal, & Ilyas [10] melakukan penelitian dengan masalah yang diangkat adalah sering kali berita yang sama dituliskan pada berbagai portal. Bahkan dapat disajikan dalam berbagai artikel pada portal yang sama, dengan penambahan sedikit informasi. Metode yang digunakan dalam penelitian ini adalah *maximum marginal relevance*. Hasil dari penelitian ini adalah sistem dapat digunakan sebagai peringkasan kumpulan berita secara otomatis, metode ini dapat digunakan untuk meringkas multi dokumen dengan menggunakan kata kunci (masuk dari user) sebagai *query*. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah pada pembobotan kata penelitian ini menggunakan TF-ISF sedangkan skripsi yang akan dilakukan menggunakan TF-IDF.

3. DATASET

Dataset yang digunakan dalam penelitian ini merupakan dataset yang berasal dari *review* aplikasi Super di Google Play Store.

4. METODE

4.1 Maximum Marginal Relevance

Algoritma *maximum marginal relevance* (MMR) merupakan salah satu metode ekstraksi ringkasan (*extractive summary*) yang digunakan untuk meringkas dokumen tunggal atau multi dokumen. MMR meringkas dokumen dengan menghitung kesamaan (similarity) antar bagian teks [10]. MMR merupakan Teknik peringkasan yang tidak mengandung redundansi dan mengambil informasi yang relevan [12]. Peringkasan dokumen dengan tipe

ekstraktif, nilai akhir diberikan pada kalimat dalam MMR dihitung dengan persamaan (1)

Pada saat parameter $\lambda=1$ maka nilai MMR yang diperoleh cenderung relevan terhadap dokumen asli [7].

$$MMR = \operatorname{argmax} [\lambda * \operatorname{Sim}1(Si, Q) - (1 - \lambda) * \operatorname{maxSim}2(Si, S')](2.1) \quad (1)$$

Keterangan:

λ = parameter yang mempengaruhi tingkat relevansi

Si = vektor bobot kata yang menjadi kandidat

S' = vektor bobot kata lainnya selain kandidat

Q = vektor bobot kata dari query

$\operatorname{Sim}1(Si, Q)$ = nilai similarity antar kalimat ke i dengan query

$\operatorname{Sim}2(Si, S')$ = nilai similarity antar kalimat ke i dengan kalimat hasil ekstraksi

Nilai dari parameter λ merupakan 1 atau 0 atau antara ($0 < \lambda < 1$). Jika parameter λ sama dengan 1 maka nilai MMR yang diperoleh akan cenderung relevan terhadap dokumen aslinya. Sedangkan ketika λ sama dengan 0 maka nilai dari *maximum marginal relevance* yang diperoleh akan cenderung relevan dengan kalimat yang sudah diekstrak sebelumnya. Oleh sebab itu, nilai dari parameter λ perlu dioptimalkan agar mendapatkan hasil ringkasan yang baik dengan nilai λ berada antara 0 sampai 1. Pada peringkasan dengan dokumen yang sedikit, seperti artikel akan menghasilkan hasil ringkasan dengan baik jika nilai dari parameter λ adalah 0,7 [12].

4.2 TF-IDF

Term weighting TF-IDF adalah salah satu pembobotan yang sering digunakan dan merupakan gabungan dari term frequency dan inverse document frequency [4]. Term frequency atau TF dihitung berdasarkan jumlah kemunculan setiap kata dalam tiap dokumen, sedangkan inverse document frequency atau IDF dihitung berdasarkan jumlah kemunculan kata dalam keseluruhan dokumen [2]. Nilai IDF sebuah term (kata) dapat dihitung menggunakan persamaan (2).

$$\operatorname{idf}_t = \log_{10} \left(\frac{N}{\operatorname{df}_t} \right) \quad (2)$$

Dimana

N = jumlah semua dokumen dalam dataset

df_t = jumlah dari dokumen yang mengandung term t di dalamnya.

TF-IDF adalah hasil perkalian antara TF dengan IDF pada setiap katanya. Untuk rumus TF-IDF dapat dilihat pada persamaan (3)

$$\operatorname{Wtd}(\operatorname{Term}1, \operatorname{Dokumen}1) = \operatorname{Nilai} \operatorname{Term} \operatorname{Dokumen}1 * \operatorname{IDF} \operatorname{Dokumen}1 \quad (3)$$

dimana

W = bobot term t pada dokumen t, d = kata ke- t dari term

d = dokumen ke- d

tf = jumlah kemunculan term t pada dokumen d df = jumlah dokumen yang memiliki term t

4.3 Cosine Similarity

Cosine similarity merupakan metode yang digunakan untuk menghitung similarity (tingkat kesamaan) antar dua objek (dokumen dengan query atau dokumen dengan dokumen) [9].

Untuk rumus *cosine similarity* dengan normalisasi [13] dapat dilihat pada persamaan (4).

$$\operatorname{CosSim}(dj, q) = \sum_{i=1}^t (\operatorname{Wijnorm} * \operatorname{Wiqnorm}) \quad (4)$$

Sedangkan untuk perhitungan *cosine similarity* tanpa menggunakan normalisasi [3] dapat dilihat pada persamaan (5).

$$\operatorname{CosSim}(dj, d) = \frac{\sum_{i=1}^t (\operatorname{Wij} * \operatorname{Wiq})}{\sum_{i=1}^t \operatorname{Wij}^2 * \sum_{i=1}^t \operatorname{Wiq}^2} \quad (5)$$

4.4 Pengukuran Keakuratan

Proses evaluasi hasil text summarization dilakukan menggunakan perhitungan *confusion matrix*. *Confusion matrix* merupakan sebuah tabel yang terdiri atas banyaknya baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi, tabel ini diperlukan untuk menentukan kinerja suatu model klasifikasi. Metode ini sering digunakan dengan kasus multiple classifiers atau kelas yang lebih dari dua. Maka dari itu metode ini cocok untuk digunakan dalam penelitian ini untuk mengukur seberapa akurat hasil klasifikasi dari model yang telah dibuat [6]. Pengukuran kinerja menggunakan confusion matrix, terdapat 4 istilah yaitu *true positive* (TP) yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem, *true negative* (TN) yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem, *false positive* (FP) yaitu jumlah data positif namun terklasifikasi salah oleh sistem, *false negative* (FN) yaitu jumlah data negatif namun terklasifikasi salah oleh sistem. Berikut merupakan rumus dari *confusion matrix*. Untuk rumus perhitungan *confusion matrix* dapat dilihat pada persamaan (6), (7), (8), dan (9).

$$\operatorname{Akurasi} = \frac{\operatorname{TP} + \operatorname{TN}}{\operatorname{TP} + \operatorname{TN} + \operatorname{FP} + \operatorname{FN}} \quad (6)$$

$$\operatorname{Presisi} = \frac{\operatorname{TP}}{\operatorname{FP} + \operatorname{TP}} \quad (7)$$

$$\operatorname{Recall} = \frac{\operatorname{TP}}{\operatorname{FN} + \operatorname{TP}} \quad (8)$$

$$\operatorname{F1} - \operatorname{Measure} = 2 * \frac{\operatorname{precision} * \operatorname{recall}}{\operatorname{precision} + \operatorname{recall}} \quad (9)$$

5. PENGUJIAN

Pada bab ini akan membahas mengenai pengujian program *text summarization* menggunakan metode *maximum marginal relevance*. Dataset yang digunakan dalam penelitian ini berupa *review* aplikasi Super yang didapatkan dari Google Play Store. Pengujian model *maximum marginal relevance* dilakukan untuk mencari nilai *precision* dari penelitian implementasi *text summarization* pada *review* aplikasi Super di Google Play Store. Proses pengujian dilakukan dengan beberapa tahapan yaitu preprocessing, menghitung TF-IDF, menghitung *cosine similarity*, menghitung *maximum marginal relevance* dan menghitung *confusion matrix*.

5.1 Preprocessing

Pada tahap ini akan dilakukan proses *preprocessing* data terhadap dataset yang digunakan dengan melakukan beberapa tahapan yaitu *splitting, case folding, normalisasi, stopwords, stemming*.

5.2 TF-IDF

Pada tahap ini akan dilakukan pembobotan kata dengan menggunakan TF-IDF. Hasil pembobotan menggunakan library TF-IDF *vectorizer*. Setiap kalimat akan diberi pembobotan, serta pembobotan query yaitu pembobotan terhadap seluruh kalimat di dataset.

5.3 Cosine Similarity

Pada tahap ini akan dilakukan perhitungan *cosine similarity*. Masing-masing dokumen akan di hitung nilai *cosine similarity* terhadap *query*.

5.4 Menghitung Maximum Marginal Relevance

Setelah hasil perhitungan cosine similarity pada masing-masing dokumen terhadap query diperoleh, maka tahap berikutnya menghitung nilai MMR. Algoritma *maximum marginal relevance* digunakan untuk mengurangi kemungkinan terjadinya redundansi pada hasil peringkasan. Perhitungan MMR dilakukan dengan iterasi mengkombinasikan 2 matrik *cosine similarity*, yaitu relevansi antara query terhadap kalimat dan similarity kalimat dengan kalimat yang telah di ekstrak. Prinsip perhitungan metode MMR adalah mengambil kalimat dengan nilai tertinggi dari setiap perhitungan iterasi, jumlah iterasi ditentukan oleh pengguna. Adapun nilai parameter λ yang digunakan pada percobaan perhitungan MMR pertama adalah $\lambda = 0,5$, dan nilai jumlah iterasi adalah 4. Proses perhitungan MMR sebagai berikut dengan catatan $Sim1(S_i, Q)$ adalah hasil perhitungan nilai cosine similarity pada tahap sebelumnya. Sedangkan $Sim1(S_i, S')$ adalah similarity kalimat terhadap kalimat yang diekstrak :

$$MMR(S_i) = \lambda \cdot Sim1(S_i, Q) - (1 - \lambda) \cdot Sim2(S_i, S')$$

Iterasi 1

Pada perhitungan proses iterasi 1, nilai Sim2 akan berisi 0, karena belum ada data yang telah terpilih sebagai pembanding cosine similarity.

Kalimat : sudah daftar tapi tidak bisa login

$$MMR : 0.5 * 0.40824829 - 0.5 * 0 = 0.20412415$$

Kalimat : kecewa banget dengan aplikasi ini tolong kalau memang mau tambah costumernya jangan php

$$MMR : 0.5 * 0.33671751 - 0.5 * 0 = [0.16835876$$

Kalimat : saya uninstall lagi aplikasi

$$MMR : 0.5 * 0.23328474 - 0.5 * 0 = 0.11664237$$

Kalimat : kenapa saya tidak bisa daftar ya padahal daftar sudah sesuai data nya telah mau login pemberitahuan nya maaf akun anda tidak aktif silah hubungi cs

$$MMR : 0.5 * 0.6244224 - 0.5 * 0 = 0.3122112$$

Kalimat : sementara saya ganti bintang

$$MMR : 0.5 * 0.11664237 - 0.5 * 0 = 0.05832118$$

Kalimat : sudah beberapa kali update telah habis update selalu buat kecewa

$$MMR : 0.5 * 0.43643578 - 0.5 * 0 = 0.21821789$$

Kalimat : update harus makin mudah customer bukan malah sulit dan makin banyak bug

$$MMR : 0.5 * 0.33197 - 0.5 * 0 = 0.165985$$

Kalimat : saya jadi telat order lewat jam

$$MMR : 0.5 * 0.14285714 - 0.5 * 0 = 0.07142857$$

Kalimat : gara suruh lengkap alamat tapi tidak bisa simpan padahal bag kirim juga sudah paham lokasi toko saya

$$MMR : 0.5 * 0.26082027 - 0.5 * 0 = 0.13041013$$

Kalimat : untuk daerah aceh tidak sedia tolong di update agar bisa guna untuk wilayah aceh

$$MMR : 0.5 * 0.35740731 - 0.5 * 0 = 0.17870365$$

Kalimat : mau masuk aja ada pesan maaf akun anda sedang tidak aktif silah hubungi customer servis bagaimana cara hubungi csnya bambang

$$MMR : 0.5 * 0.46850948 - 0.5 * 0 = 0.23425474$$

Kalimat : daftar kali kali tidak bisa login bagaimana coba aplikasinya

$$MMR : 0.5 * 0.5498574 - 0.5 * 0 = 0.2749287$$

Maka kalimat yang terpilih adalah “kenapa saya tidak bisa daftar ya padahal daftar sudah sesuai data nya telah mau login pemberitahuan nya maaf akun anda tidak aktif silah hubungi cs” dengan nilai MMR tertinggi yaitu 0.3122112. Proses diulang sesuai dengan jumlah iterasi yang diinginkan

Setelah seluruh proses selesai, didapati 4 kalimat yang terpilih dari total 12 kalimat yang diproses yaitu kalimat 3, 4, 6, dan 9. 4 kalimat yang terpilih adalah “kenapa saya tidak bisa daftar ya padahal daftar sudah sesuai data nya telah mau login pemberitahuan nya maaf akun anda tidak aktif silah hubungi cs”, “sudah beberapa kali update telah habis update selalu buat kecewa”, “gara suruh lengkap alamat tapi tidak bisa simpan padahal bag kirim juga sudah paham lokasi toko saya”, dan “saya uninstall lagi aplikasi”. 4 kalimat ini terpilih dikarenakan memiliki nilai MMR tertinggi pada setiap iterasi, yaitu 4 iterasi.

Pada percobaan kedua, dilakukan proses peringkasan pada dataset yang sama, namun dengan nilai parameter λ yang digunakan pada perhitungan MMR adalah $\lambda = 0,6$. Hasil yang diperoleh adalah kalimat 4, 6, 9, dan 12 yaitu “kenapa saya tidak bisa daftar ya padahal pendaftaran sudah sesuai data nya setelah mau login pemberitahuan nya maaf akun anda tidak aktif silahkan hubungi cs” dengan nilai MMR 0,37465344, “sudah beberapa kali update setelah habis update selalu buat kecewa” dengan nilai MMR 0,26186147, “gara disuruh lengkapi alamat tapi tidak bisa disimpan padahal bag pengiriman juga sudah paham lokasi toko saya” dengan nilai MMR 0,15649216, dan “daftar berkali kali tidak bisa login bagaimana coba aplikasinya” dengan nilai MMR 0,15353102.

Pada percobaan ketiga, dilakukan proses peringkasan pada dataset yang sama, namun dengan nilai parameter λ yang digunakan pada perhitungan MMR adalah $\lambda = 0,7$. Hasil yang diperoleh adalah kalimat 4, 6, 11, dan 12 yaitu “kenapa saya tidak bisa daftar ya padahal pendaftaran sudah sesuai data nya setelah mau login pemberitahuan nya maaf akun anda tidak aktif silahkan hubungi cs” dengan nilai MMR 0,37465344, “sudah beberapa kali update setelah habis update selalu buat kecewa” dengan nilai MMR 0,26186147, “daftar berkali kali tidak bisa login bagaimana coba aplikasinya” dengan nilai MMR 0,22923782, dan “mau masuk aja ada pesan maaf akun anda sedang tidak aktif silahkan hubungi customer servis bagaimana cara menghubungi csnya bambang” dengan nilai MMR 0,23704291.

5.5 Menghitung Confusion Matrix

Dari hasil perhitungan *maximum marginal relevance*, maka pada tahap ini dilakukan pengujian algoritma tersebut menggunakan confusion matrix untuk menghitung nilai presisi. Untuk pengujian confusion matrix, telah dikumpulkan 7 responden mahasiswa dan alumni Ilmu Komunikasi. Hasil perhitungan presisi pada ketiga percobaan menghasilkan rata-rata sebesar 40,4% dan hasil presisi terbaik sebesar 60,4% dengan nilai parameter λ adalah 0,7. Berdasarkan hasil precision 3 kali percobaan, diketahui bahwa peringkasan metode MMR dengan pembobotan kata menggunakan

metode TF-IDF menghasilkan nilai presisi yang baik dengan nilai parameter λ sebesar 0,7. Dari referensi jurnal [7] bahwa Sim adalah nilai *cosine similarity* antara dua vektor. λ merupakan nilai koefisien yang mengatur relevansi kalimat dan mengurangi redundansi. Nilai parameter λ adalah 1 atau 0 atau antara ($0 < \lambda < 1$). Pada saat parameter $\lambda=1$ maka nilai MMR yang diperoleh cenderung relevan terhadap dokumen asli. Ketika $\lambda = 0$ maka nilai MMR yang diperoleh akan cenderung relevan dengan kalimat yang diekstrak sebelumnya. Oleh karena itu, nilai λ perlu dioptimalkan agar mendapatkan ringkasan yang baik dengan nilai λ antara 0 sampai 1. Untuk peringkasan dengan dokumen yang kecil, seperti dataset akan menghasilkan hasil ringkasan yang baik jika nilai parameter $\lambda = 0,7$. Apabila jumlah dokumen banyak, maka nilai λ yang kecil akan mengurangi kemungkinan hasil yang redundan.

6. KESIMPULAN

Algoritma *maximum marginal relevance* dapat digunakan untuk meringkas teks review pengguna aplikasi Super seperti yang terlihat pada sistem bahwa algoritma *maximum marginal relevance* dapat dilakukan untuk meringkas teks review secara otomatis sesuai dengan dataset yang tersedia. Hasil evaluasi peringkasan teks yang didapatkan pada implementasi *text summarization* yang menggunakan pembobotan TF-IDF dengan penerapan algoritma *maximum marginal relevance* dalam meringkas teks dari review aplikasi Super adalah rata-rata presisi sebesar 40,4%, dan nilai presisi terbaik sebesar 60,4% dengan nilai parameter $\lambda=0,7$. Pada dokumen kecil, terbukti nilai $\lambda = 0,7$ memiliki hasil terbaik, namun pada dokumen dalam jumlah besar, semakin kecil nilai λ akan mengurangi kemungkinan hasil ringkasan yang redundan. Implementasi *text summarization* dalam membantu pihak terkait dalam memberikan informasi terkait ringkasan review aplikasi Super di Google Play Store yaitu dengan membuat aplikasi dalam bentuk website yang dapat meringkas teks review pengguna aplikasi Super menggunakan metode *maximum marginal relevance* dibuat dengan bahasa pemrograman *python* dan *framework flask* sebagai *frontend* serta database *MySQL* sebagai manajemen basis datanya. Pada aplikasi *text summarization* yang telah dibuat dapat menghasilkan *text summarization* berdasarkan dataset yang telah dikumpulkan dari Google Play Store terkait review pengguna aplikasi Super. Sehingga hasil dari *text summarization* mengasilkan informasi baru yang bermanfaat untuk memberikan ringkasan review. Dari kesimpulan yang telah dibuat, maka dalam penelitian ini memberikan saran yang bermanfaat untuk pengembangan aplikasi *text summarization* yang telah dibuat untuk penelitian selanjutnya. Saran-saran yang diberikan adalah untuk penelitian selanjutnya diharapkan dapat melakukan peringkasan otomatis dengan menggunakan metode peringkasan yang abstraktif dan untuk penelitian selanjutnya diharapkan menggunakan metode yang lainnya, seperti metode *latend semantic analysis* atau yang lain.

7. REFERENSI

- [1] Aulia, T. M., Jamaludin, A., & Padilah, T. N. (2021). Extractive Text Summerrization Pada Berita Berbahasa Indonesia Menggunakan Algoritma Support Vector Macahine. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2), 727-735. DOI=10.30645/j-sakti.v5i2.371
- [2] Irmanda, H. N., & Astriratma, R. (2020). Klasifikasi Jenis Pantun dengan Metode Support Vector Machine (SVM). *RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4 No. 5, 915 - 922. DOI=10.29207/resti.v4i5.2313
- [3] Manatap Dolok Lauro, E. V. (2021). APLIKASI CLUSTERING BERITA DENGAN METODE K MEANS DAN PERINGKAS BERITA DENGAN METODE MAXIMUM MARGINAL RELEVANCE. *Jurnal Ilmu*. DOI=10.24912/jiksi.v9i1.11560
- [4] Paulina, W., Bachtiar, F. A., & Rusydi, A. N. (2020). Analisis Sentimen Berbasis Aspek Ulasan Pelanggan Terhadap Kertanegara Premium Guest House Menggunakan Support Vector Machine. *Kertanegara Premium Guest House Menggunakan Support Vector Machine*, 4 No. 4, 1141-1149.
- [5] Rifano, E. J., Fauzan, A. C., Makhii, A., Nadya, E., Nasikin, Z., & Putra, F. N. (2020). Text Summarization pada Berita Bola Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 2(1), 8-17. DOI=10.28926/ilkomnika.v2i1.32
- [6] Rivki, M., & Bachtiar, A. M. (2017). Implementasi Algoritma K-Nearest Neighbor Dalam Pengklasifikasian Follower Twitter Yang Menggunakan Bahasa Indonesia. DOI=10.21609/jsi.v13i1.500
- [7] Rizal Setya Perdana, N. I. (2018). Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5494- 5502.
- [8] Saputra, S., Rosiyadi, D., Gata, W., & Husain, S. (2019). Analisis Sentimen E-Wallet Pada Google Play Menggunakan Algoritma Naive Bayes Berbasis Particle Swarm Optimization. *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)* Vol 3, 377-382. DOI=10.29207/resti.v3i3.1118
- [9] Saraswati, N. F., Indriati, & Perdana, R. S. (2018). Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), 5494-5502.
- [10] Setiadi, F., Djamal, E. C., & Ilyas, R. (2018). Peringkasan Kumpulan Berita Secara Otomatis Menggunakan Metode Maximum Marginal Relevance. *Prosiding SNATIF*, 235-242.
- [11] Setyadi, I. W., Krisne, D. C., & Suyadnya, I. M. (2018). Automatic Text Summarization Menggunakan Metode Graph dan Ant Colony Optimization. *17(1)*, 127-130. Suputra, I. P. (2017). PERINGKASAN TEKS OTOMATIS UNTUK DOKUMEN BAHASA BALI BERBASIS METODE EKTRAKTIF. *Jurnal Ilmiah Ilmu Komputer*. DOI=10.24843/mite.2018.v17i01.p17
- [12] Susanto, E., Mawardi, V. C., & Lauro, M. D. (2021). Aplikasi Clustering Berita dengan Metode K Means dan Peringkasan Berita dengan Metode Maximum Marginal Relevance. *Jurnal Ilmu Komputer dan Sistem Informasi*, 9(1), 62-68.
- [13] Tri Sutrisno, D. D. (2022). APLIKASI PERINGKASAN DOKUMEN MENGGUNAKAN METODE MAXIMUM. DOI=10.24912/jiksi.v10i1.17820
- [14] Verma, P., & Verma, A. (2020). A Review on Text Summarization Techniques. *Journal of Scientific Research*, 64(1), 351-357.
- [15] Yusron, M. I., Bachtiar, F. A., & Fauzi, M. A. (2019). Peringkasan Review Konsumen Restoran Menggunakan Weighted Frequent Itemset Mining. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(5), 4961-4970.