

SENTIMENT ANALYSIS FOR REVIEW MOBILE APPLICATIONS USING NEIGHBOR METHOD WEIGHTED K-NEAREST NEIGHBOR (NWKNN)

Indriati¹, Achmad Ridok²

^{1,2} Informatics Department, Faculty of Computer Science, Brawijaya University
Email: ¹indriati.tif@ub.ac.id, ²acridokb@ub.ac.id

ABSTRACT

Indonesia is a potential market for business because of a large number of smartphone users, especially developers of mobile applications. Each application stores allow the user to provide a review of the application used. The review is not only beneficial for prospective users of the application but also beneficial for the application developer. Review of the applications that are influenced by emotion (sentiment) are grouped or classified to determine positive and negative polarization. Therefore, it is necessary to have an application that can perform sentiment analysis for the mobile app reviews using Neighbor-Weighted K-Nearest Neighbor (NWKNN) classification method with high accuracy results. NWKNN method is able to classify mobile application review documents on the balanced data with current value of $k = 20$ gives the best f-measure average value of 0.9 with ratio of training data and test data 80%: 20%. However, for the imbalanced data with value of $k = 45$ gives the best f-measure average value of 0.797 with a ratio of training data and test data 80%: 20%. Based on the results, the effect of imbalanced data to the accuracy of the NWKNN methods by comparing NWKNN and KNN methods, it was found that the value of F-Measure NWKNN method is better than KNN method with gap of 0,27, due to the added weight on class minority overcome misclassification problem on minority class.

Keywords: *sentiment analysis, neighbor weighted K-nearest neighbor (NWKNN), review documents, imbalanced data*

1. PENDAHULUAN

Meningkatnya jumlah pengguna smartphone di Indonesia dimana pada akhir tahun 2014, Indonesia memiliki 38,3 juta pengguna *smartphone* dan akan terus bertambah sehingga di perkirakan tahun 2018 akan ada 100 jt pengguna *smartphone* dari Indonesia. Jumlah pengguna *smartphone* yang besar, menjadikan Indonesia sebagai pasar yang sangat potensial bagi para pebisnis terutama pengembang aplikasi *mobile*. Android, iOS, dan Windows Phone merupakan tiga *platform smartphone* yang banyak digunakan. Pada akhir desember 2014 terdapat kurang lebih 1,4 juta aplikasi yang ada di Google Play dan 1,2 juta aplikasi yang ada di ios App Store serta 300 ribu aplikasi di amazon apps store. Google play, ios AppStore dan amazon apps store merupakan tiga toko aplikasi terbesar yang menyediakan aplikasi-aplikasi yang bisa diunduh pengguna.

Setiap toko aplikasi mengizinkan pengguna aplikasi untuk memberikan *review* terhadap aplikasi yang digunakan. *Review* tersebut tidak hanya bermanfaat bagi calon pengguna aplikasi tetapi juga bermanfaat bagi pengembang aplikasi tersebut. Dari sisi pengguna, banyak sedikitnya *review* positif dan negatif dapat mempengaruhi keputusan untuk menggunakan atau tidak menggunakan aplikasi tersebut. Hal ini dikarenakan banyaknya aplikasi yang tersedia akan membingungkan pengguna untuk memilih aplikasi yang terbaik. Dari sisi pengembang, dengan membaca *review* maka pengembang dapat mengetahui fitur yang harus diperbaiki atau ditingkatkan.

Dari toko aplikasi yang ada saat ini, tidak ada yang memiliki fitur analisis sentimen untuk menyaring atau mengelompokkan antara *review* positif dan *review* negatif. Analisis sentimen adalah bidang ilmu yang mempelajari

bagaimana menganalisa opini, sentimen, evaluasi, penilaian, sikap dan emosi dari sebuah entitas yang dapat berupa produk, pelayanan, organisasi, individu, isu-isu, peristiwa, dan topik (Liu,2012). *Review* dari pengguna aplikasi yang dipengaruhi oleh emosi (sentimen) dikelompokkan atau diklasifikasikan untuk menentukan kepolarisasiannya yaitu positif, negatif, dan netral.

Analisis sentimen adalah riset komputasional dari opini, sentimen, dan emosi yang diekspresikan secara tekstual (Liu, 2010). Secara sederhana, analisis sentimen bertugas untuk klasifikasi teks dari sebuah dokumen, kalimat, atau konten lainnya menjadi kelompok sentimen positif dan negatif.

Berbagai metode telah dikembangkan dan diterapkan untuk klasifikasi teks. Salah satu metode tersebut adalah *K-Nearest Neighbor* (KNN). KNN merupakan metode yang melakukan klasifikasi berdasarkan kedekatan jarak atau kemiripan ciri/sifat dari suatu data dengan data yang lain (Prasetyo,2012) KNN melakukan klasifikasi suatu data dengan mengidentifikasi sejumlah k tetangga terdekat kemudian mengambil kelas terbanyak dalam menentukan jenis kelas dari data tersebut. KNN termasuk algoritma yang sederhana bila dibandingkan dengan algoritma lain karena KNN hanya mengandalkan memori dan tidak membangun model pada saat melakukan pembelajaran mesin. Namun KNN tidak dapat menangani data yang tidak terdistribusi secara rata, padahal kumpulan data yang tidak seimbang tersebut sering muncul. Sebuah kelas mayoritas digambarkan begitu banyak pada data latih, sedangkan kelas minoritas sebaliknya. Hal tersebut menjadikan tingkat akurasi KNN mengalami penurunan sehingga muncul metode yang merupakan pengembangan dari KNN yaitu *Neighbor-Weighted K-Nearest Neighbor* (NWKNN) (Tan, 2005)

Neighbor-Weighted K-Nearest Neighbor (NWKNN) merupakan pengembangan KNN yang bekerja dengan menerapkan prinsip pembobotan (Tan, 2005). Sejumlah k tetangga terdekat yang berasal dari kelas mayoritas akan diberi bobot lebih sedikit dari kelas minoritas. Berdasarkan hasil penelitian yang dilakukan oleh Songbo Tan, NWKNN mampu

melakukan klasifikasi dan memiliki kinerja yang baik untuk menyelesaikan masalah data teks yang tidak terdistribusi secara rata.

Sebagian besar proses klasifikasi teks menggunakan fitur kata atau term untuk melakukan klasifikasi. Semakin banyak kata yang digunakan akan menyebabkan proses komputasi semakin lama. Dalam penelitian ini, fitur yang digunakan untuk merepresentasikan suatu dokumen berdasarkan dari kamus (*lexicon-based*). Dokumen akan diekstrak menggunakan kata kunci positif, negatif, negasi positif, dan negasi negatif yang dilihat dari kamus dan dihitung jumlah kata tersebut dalam suatu dokumen. Diharapkan dengan ekstraksi kata kunci berbasis kamus ini dapat mempercepat proses komputasi dan dapat merepresentasikan suatu dokumen dalam sentimen positif atau negatif.

Berdasarkan penjelasan di atas maka diperlukan adanya aplikasi yang dapat melakukan analisis sentimen review aplikasi *mobile* dengan menggunakan metode klasifikasi NWKNN dengan hasil akurasi yang tinggi.

2. DASAR TEORI

2.1 Text Mining

Text mining adalah proses penemuan pola yang sebelumnya tidak terlihat pada dokumen atau sumber tertentu menjadi pola yang diinginkan untuk tujuan tertentu (Mustafa, 2009) *Text mining* sering digunakan untuk analisis informasi, pengambilan keputusan, dan tugas-tugas manajemen informasi lainnya yang terkait dengan data teks dalam jumlah besar. Sistem memanfaatkan peningkatan jumlah data yang tidak terstruktur dalam bentuk teks. Data tersebut diolah untuk memenuhi kebutuhan informasi menggunakan berbagai metode klasifikasi, *clustering*, analisis sentimen, dll.

Text mining adalah bagian dari *data mining*, namun proses *text mining* memerlukan lebih banyak tahapan dibanding *data mining* karena data teks memiliki karakteristik yang lebih kompleks daripada data biasa bahkan data yang sudah terstruktur. Berdasarkan ketidakteraturan struktur data teks, maka proses *text mining* memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan

agar teks dapat diubah menjadi lebih terstruktur.

2.2 Analisis Sentimen

Analisis sentimen adalah bidang ilmu yang mempelajari bagaimana menganalisa opini, sentimen, evaluasi, penilaian, sikap dan emosi dari sebuah entitas yang dapat berupa produk, pelayanan, organisasi, individu, isu-isu, peristiwa, dan topik (Liu, 2012). Analisis sentimen biasa digunakan untuk menilai kesukaan atau ketidaksukaan publik terhadap suatu barang atau jasa. Sentimen tersebut merupakan informasi tekstual yang bersifat subjektif dan memiliki polaritas positif dan negatif. Nilai polaritas ini dapat digunakan sebagai parameter dalam menentukan suatu keputusan.

Sebagai contoh, seorang pendatang baru di suatu daerah meminta pendapat orang di daerah tersebut tentang operator selular yang cocok digunakan untuk mengakses internet. Pendapat orang-orang terhadap operator A berbeda-beda, ada yang positif ada yang negatif. Jumlah pendapat positif dan negatif ini akan menentukan keputusan yang akan diambil pendatang baru tersebut dalam memilih operator selular. Jumlah pendapat yang sangat banyak akan membuat pengambilan keputusan menjadi tidak efisien.

2.3 Text Preprocessing

Text preprocessing adalah tahap awal yang digunakan untuk mendapatkan data yang siap untuk diproses. Tahapan dalam *text preprocessing* yaitu: *tokenizing*, *filtering*, dan *stemming*.

2.3.1 Tokenizing

Tokenizing adalah proses pengambilan kata yang menjadi penyusun suatu dokumen. Karakter-karakter pemisah kata akan dihilangkan karena tidak memiliki pengaruh terhadap pemrosesan teks. Karakter-karakter pemisah tersebut bisa berupa spasi, tanda baca, angka, dan karakter selain huruf. Dalam proses ini juga dilakukan perubahan semua huruf menjadi huruf kecil.

2.3.2 Filtering

Filtering adalah proses pemilihan kata dari hasil *tokenizing* yang akan digunakan

untuk merepresentasikan suatu dokumen. *Filtering* dapat dilakukan dengan dua cara yaitu *filtering* berdasarkan *stoplist* atau berdasarkan *wordlist*. *Stoplist* adalah kumpulan kata-kata tidak penting. Setiap kata dalam dokumen akan dibandingkan dengan setiap kata dalam *stoplist*. Kata dalam dokumen yang juga merupakan kata dalam *stoplist*, akan dibuang dari hasil *tokenizing*. *Wordlist* adalah kumpulan kata-kata penting yang berpotensi mampu merepresentasikan suatu dokumen. Sistem akan mengambil kata dari dokumen hasil *tokenizing* yang juga merupakan kata dalam *wordlist*. Dalam pengaplikasiannya, penggunaan *stoplist* lebih efisien daripada *wordlist*. Karena banyaknya kata yang tidak penting jauh lebih sedikit daripada kata yang penting.

2.3.3 Stemming

Proses *stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dan kata tersebut harus sesuai dengan struktur morfologi bahasa Indonesia yang benar. Penggunaan imbuhan berupa awalan atau akhiran pada suatu kata akan dihapus sehingga akan mendapatkan inti dari kata tersebut. Salah satu algoritma *stemming* yaitu *stemming* Arifin-Setiono.

2.4 Pembobotan

Pembobotan adalah proses merubah *term* yang merupakan data kualitatif menjadi data kuantitatif sehingga bisa diproses oleh komputer. Pembobotan dilakukan pada dokumen yang sudah dilakukan *preprocessing*. Beberapa metode pembobotan yang umum digunakan yaitu pembobotan TF dan TF-IDF.

Term frequency (TF) adalah jumlah kemunculan kata pada suatu dokumen. Semakin banyak suatu kata muncul pada dokumen, maka semakin besar kata tersebut berpengaruh pada dokumen tersebut. Sebaliknya, semakin sedikit suatu kata muncul pada dokumen, maka semakin kecil kata tersebut berpengaruh pada dokumen tersebut. Pembobotan menggunakan TF dijelaskan pada Persamaan (1) (Feldman, 2007).

$$w_{tf(t,d)} = 1 + \log TF(t, d) \quad (1)$$

Keterangan:

$w_{tf(t,d)}$ = bobot *term* t pada dokumen d,

$TF(t,d)$ = jumlah kemunculan kata t pada dokumen d .

Inverse document frequency (IDF) adalah pembobotan kata yang didasarkan pada banyaknya dokumen yang mengandung kata tertentu. Semakin banyak dokumen yang mengandung suatu kata tertentu, semakin kecil pengaruh kata tersebut pada dokumen. Sebaliknya, semakin sedikit dokumen yang mengandung suatu kata tertentu, semakin besar pengaruh kata tersebut pada dokumen. Pembobotan menggunakan TF-IDF dijelaskan pada Persamaan (2) (Feldman,20017)

$$w(t,d) = TF(t,d) * \log \frac{D}{Dt} \quad (2)$$

Keterangan:

$w(t,d)$ = bobot kata t pada dokumen d ,

$TF(t,d)$ = jumlah kemunculan kata t pada dokumen d ,

D = jumlah seluruh dokumen,

Dt = Jumlah dokumen yang mengandung kata t .

2.5 Klasifikasi

Dalam metode *text mining* dapat dikategorikan sebagai *supervised learning* ataupun *unsupervised learning*. Perbedaan antara *supervised learning* dengan *unsupervised learning* adalah *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada (data latih) dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data. Klasifikasi termasuk *supervised learning* dimana ada data/variabel yang telah diidentifikasi sebelumnya (Larose, 2005). Klasifikasi merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan kelas data dengan tujuan untuk memperkirakan kelas yang tidak diketahui dari suatu objek.

Dalam pengklasifikasian data terdapat tiga proses yang dilakukan yaitu:

1. Proses *training*

Pada proses *training* digunakan data latih yang telah diketahui label-labelnya untuk membangun model atau fungsi.

2. Proses *testing*

Untuk mengetahui keakuratan model atau fungsi yang akan dibangun pada proses

training, maka digunakan data yang disebut dengan data uji untuk memprediksi label-labelnya.

3. Proses validasi

Validasi dilakukan untuk meminimalkan kesalahan pada data validasi.

2.6 Neighbor-Weighted K-Nearest Neighbor (NWKNN)

Asumsi yang dimiliki bahwa data latih didistribusikan secara merata dan seimbang di setiap kelas atau kategori tidak selalu tepat. Pada kenyataannya ada kemungkinan bahwa data latih yang tidak tersebar secara merata dan seimbang.

Sebuah kelas mayoritas digambarkan begitu banyak pada data latih, sedangkan

kelas minoritas sebaliknya. *Neighbor-Weighted K-Nearest Neighbor* (NWKNN) dikembangkan oleh Songbo Tan pada tahun 2005 merupakan modifikasi dari KNN untuk menyelesaikan masalah tersebut (Tan, 2005)

Penerapan algoritma NWKNN serupa dengan KNN tradisional. Pada tahapan awal dilakukan penghitungan kemiripan atau jarak antara kelompok k data latih dengan obyek pada data baru atau data uji. Terdapat beberapa cara yang umum digunakan untuk mengukur jarak antar data lama dan data baru, diantaranya *euclidean distance* dan *cosine similarity*. Tahapan berikutnya, dilakukan pengurutan nilai similaritas berdasarkan nilai terbesar dan pemilihan k tetangga.

Setelah k tetangga didapatkan, untuk mendapatkan hasil akhir pada algoritma KNN dilakukan penghitungan skor terhadap setiap kategori. Rumus penghitungan skor pada algoritma KNN seperti pada persamaan (3).

$$Score(q, C_i) = \sum_{d_j \in KNN(q)} Sim(q, d_j) \delta(d_j, C_i) \quad (3)$$

dimana :

$$\begin{aligned} d_j \in KNN(q) &= \text{data latih } d_j \text{ yang berada pada} \\ &\text{kumpulan tetangga terdekat} \\ &\text{(nearest neighbor) dari} \\ &\text{dokumen uji } q \\ Sim(q, d_j) &= \text{similaritas antara dokumen} \\ &\text{uji } q \text{ dengan dokumen latih} \\ &d_j \\ \delta(d_j, C_i) &= \begin{cases} d_j \in C_i = 1 \\ d_j \notin C_i = 0 \end{cases} \end{aligned}$$

C_i = Kelas / kategori i

Pada tahapan penghitungan skor ini terdapat perbedaan antara algoritma KNN dan NWKNN. Pada algoritma NWKNN dilakukan tahapan penghitungan bobot terlebih dahulu. Setiap kategori (C_i) akan ditentukan bobotnya. Kategori mayoritas akan diberi bobot kecil, sementara kategori minoritas akan diberi bobot yang lebih besar. Rumus penghitungan bobot kategori ($Weight_i$) dapat dihitung dengan persamaan (4) :

$$Weight_i = \frac{1}{\left(\frac{Num(C_i^d)}{\min\{Num(C_j^d) | j=1, \dots, k^*\}} \right)^{1/exp}} \quad (4)$$

dimana :

$Num(C_i^d)$ = banyaknya data latih d pada kelas i

$Num(C_j^d)$ = banyaknya data latih d pada kelas j , dimana j terdapat dalam himpunan k tetangga terdekat

exp = eksponen, bilangan lebih dari 1

Setiap bobot kategori yang didapatkan kemudian akan digunakan untuk menghitung skor data uji q terhadap setiap kategori. Hasil dari penghitungan skor akan digunakan sebagai acuan untuk menentukan kategori dari data uji yang diproses. Penghitungan skor pada metode NWKNN hampir sama seperti pada KNN. Namun, pada metode NWKNN setiap jumlah skor dari kategori akan dikalikan dengan masing-masing bobot dari kategorinya. Perhitungan skor setiap data uji q terhadap kategori ke i ($Score(q, C_i)$) untuk metode NWKNN dihitung dengan persamaan (5).

$$Score(q, C_i) = Weight_i \left(\sum_{d_j \in KNN(q)} Sim(q, d_j) \delta(d_j, C_i) \right) \quad (5)$$

Keterangan :

$Weight_i$ = Bobot kelas i
 $d_j \in KNN(q)$ = data latih d_j yang berada pada kumpulan tetangga terdekat

(*nearest neighbor*) dari dokumen uji q

$Sim(q, d_j)$ = similaritas antara dokumen uji q dengan dokumen latih d_j

$\delta(d_j, C_i)$ = $\begin{cases} d_j \in C_i = 1 \\ d_j \notin C_i = 0 \end{cases}$

C_i = Kelas / kategori i

Berdasarkan langkah-langkah yang umum digunakan pada penerapan metode KNN dapat implementasikan pula pada penerapan metode NWKNN dengan melakukan perubahan pada penghitungan skor metode KNN. Langkah-langkah yang umum digunakan pada algoritma KNN:

1. Menentukan parameter k (jumlah tetangga paling dekat).
2. Menghitung jarak kesamaan masing-masing obyek atau data uji terhadap data sampel atau data latih yang diberikan.
3. Kemudian mengurutkan obyek-obyek tersebut ke dalam kelompok yang mempunyai jarak kesamaan terkecil.
4. Mengumpulkan kategori klasifikasi *nearest neighbor*.
5. Dengan menggunakan kategori *nearest neighbor* yang paling mayoritas maka dapat diprediksi nilai skor yang telah dihitung. Pada tahapan ini penghitungan skor algoritma KNN akan diubah dengan cara penghitungan skor algoritma NWKNN seperti pada persamaan (5).

3. PENGUJIAN

3.1 Hasil Pengujian Pengaruh Nilai k Tetangga yang Bervariasi Terhadap Akurasi Dari Metode Klasifikasi NWKNN pada data yang seimbang

Pengujian pertama ini dilakukan untuk mengetahui pengaruh nilai k yang bervariasi terhadap performa metode NWKNN pada data yang seimbang yaitu data yang jumlah data positif sama dengan data negatif. Nilai k tetangga yang digunakan adalah 2, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50. Data review yang digunakan berjumlah 200 data terdiri dari 100 data positif dan 100 data negatif. Data akan dibagi menjadi dua kelompok data latih dan data uji dengan menggunakan perbandingan data latih berbanding data uji yang berubah-ubah.

Pada pengujian pertama untuk skenario 1 dilakukan perbandingan data uji dan data latih sebesar 50%:50%. Hasil dari pengujian

pertama untuk skenario 1 dapat dilihat pada Tabel 1.

Tabel 1. Hasil pengujian 1 skenario 1

Nilai k	Precision	Recall	F-Measure
2	0.802	0.800	0.800
3	0.804	0.800	0.799
5	0.816	0.810	0.809
10	0.816	0.810	0.809
15	0.811	0.810	0.810
20	0.837	0.830	0.829
25	0.822	0.820	0.820
30	0.822	0.820	0.820
35	0.822	0.820	0.820
40	0.822	0.820	0.820
45	0.822	0.820	0.820
50	0.822	0.820	0.820

Pada pengujian pertama untuk skenario 2 dilakukan perbandingan data uji dan data latih sebesar 60%:40%. Hasil dari pengujian pertama untuk skenario 2 dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian 1 skenario 2

Nilai k	Precision	Recall	F-Measure
2	0.800	0.800	0.800
3	0.828	0.825	0.825
5	0.828	0.825	0.825
10	0.843	0.838	0.837
15	0.828	0.825	0.825
20	0.839	0.838	0.837
25	0.825	0.825	0.825
30	0.826	0.825	0.825
35	0.813	0.813	0.812
40	0.838	0.838	0.837
45	0.826	0.825	0.825
50	0.826	0.825	0.825

Pada pengujian pertama untuk skenario 3 dilakukan perbandingan data uji dan data latih sebesar 70%:30%. Hasil dari pengujian pertama untuk skenario 3 dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian 1 skenario 3

Nilai k	Precision	Recall	F-Measure
2	0.685	0.683	0.683
3	0.805	0.800	0.799
5	0.811	0.783	0.778
10	0.813	0.800	0.798
15	0.826	0.817	0.815
20	0.839	0.833	0.833
25	0.820	0.817	0.816
30	0.817	0.817	0.817
35	0.817	0.817	0.817
40	0.817	0.817	0.817
45	0.817	0.817	0.817
50	0.817	0.817	0.817

Pada pengujian pertama untuk skenario 4 dilakukan perbandingan data uji dan data latih sebesar 80%:20%. Hasil dari pengujian pertama untuk skenario 4 dapat dilihat pada Tabel 4.

Tabel 4. Hasil pengujian 1 skenario 4

Nilai k	Precision	Recall	F-Measure
2	0.800	0.800	0.800
3	0.776	0.775	0.775
5	0.876	0.875	0.875
10	0.876	0.875	0.875
15	0.904	0.900	0.900
20	0.904	0.900	0.900
25	0.876	0.875	0.875
30	0.904	0.900	0.900
35	0.876	0.875	0.875
40	0.876	0.875	0.875
45	0.876	0.875	0.875
50	0.876	0.875	0.875

Pengujian pertama dengan menggunakan empat skenario pengujian menghasilkan nilai F-Measure yang paling besar yaitu 0.9 dengan menggunakan perbandingan data latih dan data uji sebesar 80% :20% dan k=20 seperti diperlihatkan pada Tabel 5.

Tabel 5. Hasil pengujian terhadap variasi jumlah data latih dan data uji

rasio	k	Data Latih	Data Uji	P	R	F
80% : 20%	20	160	40	0.904	0.900	0.900
70% : 30%	20	140	60	0.839	0.833	0.833
60% : 40%	20	120	80	0.839	0.838	0.837
50% : 50%	20	100	100	0.837	0.830	0.829

3.2 Hasil Pengujian Pengaruh Nilai k Tetangga yang Bervariasi Terhadap Akurasi Dari Metode Klasifikasi NWKNN pada data yang tidak seimbang

Pengujian kedua ini dilakukan untuk mengetahui pengaruh nilai k yang bervariasi terhadap performa metode NWKNN pada data yang tidak seimbang yaitu data yang jumlah data positif lebih kecil dibandingkan dengan data negatif. Nilai k tetangga yang digunakan adalah 2, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50. Data review yang digunakan berjumlah 550 data terdiri dari 50 data positif dan 500 data negatif. Perbandingan data latih berbanding data uji akan berubah-ubah.

Pada pengujian kedua untuk skenario 1 dilakukan perbandingan data uji dan data latih sebesar 50%:50%. Hasil dari pengujian kedua untuk skenario 1 dapat dilihat pada Tabel 6.

Tabel 6. Hasil pengujian 2 skenario 1

Nilai k	Precision	Recall	F-Measure
2	0.496	0.496	0.496
3	0.498	0.498	0.498
5	0.498	0.498	0.498
10	0.500	0.500	0.500
15	0.631	0.732	0.656
20	0.628	0.730	0.653
25	0.622	0.726	0.645
30	0.611	0.718	0.631
35	0.606	0.714	0.624
40	0.611	0.718	0.631
45	0.625	0.728	0.649
50	0.500	0.500	0.500

Pada pengujian kedua untuk skenario 2 dilakukan perbandingan data uji dan data latih

sebesar 60%:40%. Hasil dari pengujian kedua untuk skenario 2 dapat dilihat pada Tabel 7.

Tabel 7 Hasil pengujian 2 skenario 2

Nilai k	Precision	Recall	F-Measure
2	0.522	0.525	0.523
3	0.451	0.465	0.458
5	0.518	0.515	0.516
10	0.514	0.513	0.513
15	0.514	0.513	0.513
20	0.589	0.638	0.603
25	0.585	0.635	0.599
30	0.612	0.685	0.632
35	0.608	0.683	0.627
40	0.594	0.673	0.610
45	0.594	0.673	0.610
50	0.588	0.668	0.602

Pada pengujian kedua untuk skenario 3 dilakukan perbandingan data uji dan data latih sebesar 70%:30%. Hasil dari pengujian kedua untuk skenario 3 dapat dilihat pada Tabel 8.

Tabel 8. Hasil pengujian 1 skenario 3

Nilai k	Precision	Recall	F-Measure
2	0.500	0.500	0.499
3	0.452	0.470	0.461
5	0.452	0.470	0.461
10	0.586	0.570	0.576
15	0.586	0.570	0.576
20	0.452	0.470	0.461
25	0.593	0.640	0.607
30	0.593	0.640	0.607
35	0.645	0.740	0.673
40	0.645	0.740	0.673
45	0.639	0.737	0.666
50	0.654	0.770	0.685

Pada pengujian kedua untuk skenario 4 dilakukan perbandingan data uji dan data latih sebesar 80%:20%. Hasil dari pengujian kedua untuk skenario 4 dapat dilihat pada Tabel 9.

Tabel 9. Hasil pengujian 1 skenario 4

Nilai k	Precision	Recall	F-Measure
2	0.500	0.500	0.500

3	0.451	0.465	0.458
5	0.451	0.465	0.458
10	0.453	0.480	0.466
15	0.453	0.480	0.466
20	0.453	0.480	0.466
25	0.453	0.480	0.466
30	0.734	0.815	0.766
35	0.734	0.815	0.766
40	0.734	0.815	0.766
45	0.756	0.865	0.797
50	0.756	0.865	0.797

Pengujian kedua dengan menggunakan empat skenario pengujian menghasilkan nilai F-Measure yang paling besar yaitu 0.797 dengan menggunakan perbandingan data latih dan data uji sebesar 80% :20% dan k=45 seperti diperlihatkan pada Tabel 10.

Tabel 10. Hasil pengujian terhadap variasi jumlah data latih dan data uji

Rasio	k	Data Latih	Data Uji	P	R	F
80% : 20%	45	440	110	0.756	0.865	0.797
70% : 30%	45	385	165	0.639	0.737	0.666
60% : 40%	45	330	220	0.594	0.673	0.610
50% : 50%	45	275	275	0.625	0.728	0.649

3.3 Hasil Pengujian Pengaruh Nilai k tetangga yang Bervariasi Terhadap Akurasi Dari Metode Klasifikasi NWKNN dan KNN pada data yang seimbang

Pengujian ketiga dilakukan dengan cara membandingkan akurasi antara metode NWKNN dan KNN. Jumlah data yang digunakan pada pengujian ini sebanyak 200 data terdiri dari 100 data positif dan 100 data negatif. Data dibagi menjadi dua kelompok data latih dan data uji dengan perbandingan 80% : 20%. Perbandingan tersebut merupakan perbandingan data yang menghasilkan nilai *f-measure* tertinggi pada pengujian sebelumnya.

Hasil dari pengujian ini akan didapatkan selisih nilai *f-measure* antara metode NWKNN dan KNN. Hasil dari pengujian ini dapat dilihat pada Tabel 11.

Tabel 11. Hasil pengujian metode NWKNN dan KNN pada data seimbang

k	NWKNN			KNN		
	P	R	F	P	R	F
2	0.800	0.800	0.800	0.598	0.550	0.487
3	0.776	0.775	0.775	0.776	0.775	0.775
5	0.876	0.875	0.875	0.876	0.875	0.875
10	0.876	0.875	0.875	0.884	0.875	0.874
15	0.904	0.900	0.900	0.904	0.900	0.900
20	0.904	0.900	0.900	0.904	0.900	0.900
25	0.876	0.875	0.875	0.876	0.875	0.875
30	0.904	0.900	0.900	0.884	0.875	0.874
35	0.876	0.875	0.875	0.854	0.850	0.850
40	0.876	0.875	0.875	0.854	0.850	0.850
45	0.876	0.875	0.875	0.854	0.850	0.850
50	0.876	0.875	0.875	0.854	0.850	0.850

3.4 Hasil Pengujian Pengaruh Nilai k Tetangga yang Bervariasi Terhadap Akurasi Dari Metode Klasifikasi NWKNN dan KNN pada data yang tidak seimbang

Pengujian keempat dilakukan dengan cara membandingkan akurasi antara metode NWKNN dan KNN. Jumlah data yang digunakan pada pengujian ini sebanyak 550 data terdiri dari 50 data positif dan 500 data negatif. Data dibagi menjadi dua kelompok data latih dan data uji dengan perbandingan 80% : 20%. Perbandingan tersebut merupakan perbandingan data yang menghasilkan nilai *f-measure* tertinggi pada pengujian sebelumnya. Hasil dari pengujian ini akan didapatkan selisih nilai *f-measure* antara metode NWKNN dan KNN. Selisih tersebut akan membuktikan tingkat perbaikan yang dilakukan oleh metode NWKNN terhadap metode KNN pada saat data tidak seimbang. Hasil dari pengujian ini dapat dilihat pada Tabel 12.

Tabel 12 Hasil pengujian metode NWKNN dan KNN pada data tidak seimbang

k	NWKNN	KNN
---	-------	-----

	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>
2	0.500	0.500	0.500	0.528	0.520	0.522
3	0.451	0.465	0.458	0.453	0.485	0.469
5	0.451	0.465	0.458	0.455	0.500	0.476
10	0.453	0.480	0.466	0.455	0.500	0.476
15	0.453	0.480	0.466	0.455	0.500	0.476
20	0.453	0.480	0.466	0.455	0.500	0.476
25	0.453	0.480	0.466	0.455	0.500	0.476
30	0.734	0.815	0.766	0.455	0.500	0.476
35	0.734	0.815	0.766	0.455	0.500	0.476
40	0.734	0.815	0.766	0.455	0.500	0.476
45	0.756	0.865	0.797	0.455	0.500	0.476
50	0.756	0.865	0.797	0.455	0.500	0.476

4. ANALISIS PENGUJIAN

Analisis yang didapatkan dari hasil keempat pengujian yang dilakukan yaitu algoritma NWKNN dapat digunakan untuk untuk data yang seimbang maupun data tidak seimbang. Pengujian pertama dan kedua menunjukkan bahwa algoritma NWKNN dapat digunakan diberbagai variasi jumlah data latih dan data uji dengan nilai F Measure terbaik diperoleh ketika perbandingan data latih dan data uji sebesar 80%:20%. Kalau data seimbang nilai dari klasifikasi menggunakan KNN dan NWKNN hampir sama dan tidak menunjukkan peningkatan yang signifikan. Akan tetapi peningkatan akan terlihat ketika data tidak seimbang dikarenakan NWKNN menerapkan bobot/skor pada data yang mempunyai jumlah yang sedikit sehingga kesalahan klasifikasi untuk jumlah data yang sedikit dapat dikurangi. Hal ini terlihat pada pengujian ketika nilai F-Measure tertinggi pada KNN dan NWKNN sebesar 0,9 akan tetapi pada pengujian keempat F-Measure terbesar pada KNN yaitu 0,522 dan pada NWKNN yaitu 0,797. Perbedaan yang signifikan pada jumlah data uji dan data latih tidak seimbang dikarenakan NWKNN menerapkan prinsip pembobotan. Sejumlah k tetangga terdekat yang berasal dari kelas minoritas akan diberikan bobot yang lebih besar dibandingkan dari kelas yang mayoritas. Oleh karena itu, kesalahan klasifikasi untuk klas yang minoritas dapat dikurangi. Selain itu kesalahan klasifikasi juga terjadi apabila jumlah kata positif dan negatifnya sama dalam

sebuah review. Hal ini akan membuat sistem memilih salah satu kelas saja padahal jumlah fitur positif dan negatifnya sama.

5. KESIMPULAN

Metode NWKNN dapat digunakan untuk mengklasifikasikan dokumen review aplikasi mobile dengan melalui tahapan *text processing* terlebih dahulu, yakni tokenisasi, *filtering*, *stemming*. Selanjutnya dilakukan proses pembentukan kamus fitur dan pembobotannya.

Metode NWKNN mampu melakukan klasifikasi dokumen review aplikasi mobile dengan baik pada data seimbang saat nilai $k=20$ dengan rata-rata nilai *f-measure* terbaik sebesar 0,9 dengan perbandingan data latih dan data uji 80% : 20%. Sedangkan pada data tidak seimbang saat nilai $k=45$ dengan rata-rata nilai *f-measure* terbaik sebesar 0,797 pada perbandingan data latih dan data uji 80% : 20%.

Berdasarkan hasil pengujian pengaruh data latih tidak seimbang terhadap akurasi metode NWKNN dengan membandingkan metode NWKNN dan KNN, didapatkan bahwa metode NWKNN lebih baik 0.27 dibandingkan metode KNN dalam melakukan klasifikasi terhadap dokumen review aplikasi mobile dikarenakan adanya penambahan bobot pada klas minoritas sehingga kesalahan klasifikasi data pada klas minoritas dapat dikurangi.

DAFTAR PUSTAKA

- CHAOVALIT, P., ZHOU, L. 2005. *Movie review mining: A comparison between supervised and unsupervised classification approaches*, in Proceedings of the Hawaii International Conference on System Sciences.
- FELDMAN, R., & SANGER, J. 2007. *The Text Mining Handbook*. New York: Cambridge University Press.
- LAROSE, DANIEL T. 2005. *Discovering Knowledge in Data: An Introduction to Data Mining*.
- LIU, BING. 2010. *Sentiment Analysis and Subjectivity, Handbook of natural language processing 2: 627-666*.

- LIU, BING. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
- MUSTAFA, ATIKA, ALI AKBAR, and AHMER SULTAN. 2009, *Knowledge Discovery using Teks Mining: A Programmable Implementation on Information Extraction and Categorization*, international Journal of Multimedia and Ubiquitous Engineering Vol. 4, No. 2, April.
- NURFALAH, ADIYASAN. 2013, *Sentiment Analysis Menggunakan Pendekatan Lexicon-Based*,
- PRASETYO, EKO. 2012. *Data Mining: Konsep dan Aplikasi menggunakan Matlab*, Andi
- TAN, SONGBO. 2005. *Neighbor-weighted K-nearest neighbor for unbalanced text corpus*, Expert Systems with Applications 28 (2005) 667–671.