Penerapan *Synthetic Minority Oversampling Technique* (SMOTE) untuk *Imbalance Class* pada *Data Text* Menggunakan KNN

Sultan Maula Chamzah[1], Merinda Lestandy[2*], Nur Kasan[3], Adhi Nugraha[4]

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Malang
Jalan Raya Tlogomas No. 246, Malang, Jawa Timur
Email: *merindalestandy@umm.ac.id

Abstrak. *Marketplace* adalah salah satu jenis web e-niaga dimana informasi produk atau layanan disediakan oleh banyak pihak ketiga, salah satu contohnya yaitu Tokopedia. Tokopedia mendapatkan jumlah pengunjung website maupun aplikasi rata-rata 147,79 juta per bulan. Meski memiliki pengguna yang banyak, tentu saja dalam sebuah aplikasi memiliki kekurangan dan kelebihan. Hal tersebut disampaikan oleh pengguna melalui *review* atau ulasan yang terdapat pada Google Play Store. Pada ulasan tersebut terlihat bahwa pengguna yang memberikan ulasan rating bintang 5 lebih banyak daripada pengguna memberikan rating bintang 1. *Synthetic Minority Oversampling Technique* atau SMOTE merupakan metode yang populer diterapkan dalam rangka menangani ketidakseimbangan kelas. Penelitian ini bertujuan untuk mengetahui performa dari algoritma K-Nearest Neighbor dalam menangani *imbalance class* menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). Penelitian ini menggunakan 5000 data yang terdiri dari 3975 data negatif dan 1025 data positif. Dari 5000 data dibagi menjadi dua bagian, 70% data latih dan 30% data uji. Tujuan dari penelitian ini yaitu untuk mengetahui performansi penanganan ketidakseimbangan kelas pada data review aplikasi Tokopedia menggunakan algoritma *kNN* dan SMOTE. Metode SMOTE-kNN menunjukkan hasil akurasi yang lebih baik yaitu sebesar 90% dibandingkan hanya menggunakan *kNN* dengan nilai akurasi 82%.

Kata kunci: *Tokopedia, Imbalance Class, kNN, SMOTE-kNN*

1 Pendahuluan

Perkembangan internet telah tumbuh pada tingkat yang luar biasa di banyak negara termasuk Indonesia dimana jumlah penggunanya melesat dalam tiga tahun terakhir. Begitupula dengan pengguna internet di Indonesia yang semakin berkembang seiring dengan kemajuan teknologi informasi. Berdasarkan survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pada tahun 2019 – Q2 2020 menyebutkan bahwa pengguna internet di Indonesia mencapai 196,7 juta dari 266,9 juta penduduk [1]. Meningkatnya jumlah pengguna internet ini tentu saja menjadi peluang besar bagi para pelaku usaha untuk memasarkan produknya melalui internet atau berbasis *online*. Salah satu dari sekian banyak

Diterima 16 November 2022 Direvisi 20 Oktober 2022 Diterima untuk publikasi 31 Oktober 2022

cara yang digunakan dalam melakukan transaksi jual beli di dunia maya yaitu menggunakan *marketplace*. *Marketplace* sendiri juga berbagai macam, salah satunya ialah Tokopedia.

Tokopedia merupakan salah satu penyedia online *marketplace* di Indonesia yang memfasilitasi pengguna internet untuk melakukan jual beli secara *online*. Berdasarkan riset iPrice Group, Tokopedia mendapatkan jumlah pengunjung website maupun aplikasi rata-rata 147,79 juta per bulan [2]. Dengan jumlah pengunjung yang cukup tinggi, maka Tokopedia dapat menunjukkan bahwa masyarakat memiliki ketertarikan tersendiri maupun memiliki keunikan dibandingkan dengan *marketplace* yang lain. Tokopedia merupakan *marketplace* yang menghubungkan langsung antara penjual dengan pembeli dengan melibatkan pihak Tokopedia dalam melakukan pembayaran, hal ini sering disebut dengan rekening bersama (rekber). Sehingga para konsumen atau pembeli tidak perlu meragukan kualitas dan integritas dari pembeli sehingga muncullah rasa percaya antara penjual dengan pembeli.

Meski memiliki pengguna yang banyak, tentu saja dalam sebuah aplikasi memiliki kekurangan dan kelebihan. Hal tersebut disampaikan oleh pengguna melalui *review* atau ulasan yang terdapat pada *Google Play Store*. Sebanyak 5.185.251 ulasan telah diunggah oleh pengguna aplikasi Tokopedia pada *Google Play Store*. Pada ulasan tersebut terlihat bahwa pengguna yang memberikan ulasan rating bintang 5 lebih banyak daripada pengguna memberikan rating bintang 1. Data tidak seimbang adalah ketika distribusi kelas data tidak seimbang dan jumlah kelas data (*instance*) lebih sedikit atau lebih dari jumlah kelas data lainnya. Sekelompok kelas data yang tidak dikenal sebagai kelompok minoritas, kelompok kelas data lainnya disebut kelompok mayoritas. Kondisi ini menyulitkan metode klasifikasi untuk melakukan fungsi generalisasi dalam proses machine learning. Hampir semua algoritma klasifikasi, seperti *Naive Bayes*, *Decision Tree*, dan *k-Nearest Neighbor*, berkinerja sangat buruk ketika kelas menangani data yang tidak proporsional secara signifikan. Metode klasifikasi di atas tidak memiliki kemampuan untuk mengatasi masalah ketidakseimbangan kelas [3]. Untuk mengatasi ketidaksamaan kelas dapat dilakukan dengan menggunakan teknik *Synthetic Minority Oversampling Technique* (SMOTE).

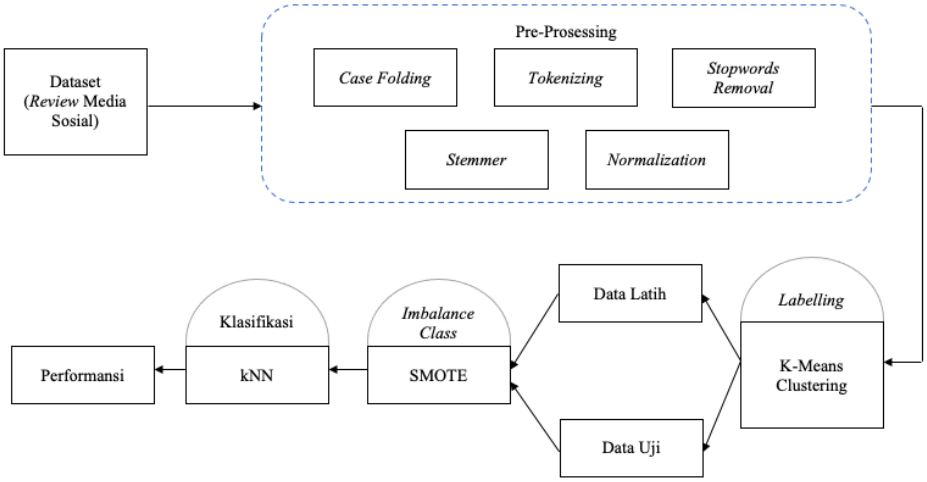
Synthetic Minority Oversampling Technique atau SMOTE merupakan metode yang populer diterapkan dalam rangka menangani ketidak seimbangan kelas. Teknik ini mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan dataset dengan cara sampling ulang sampel kelas minoritas [3]. Penelitian mengenai SMOTE sebelumnya dilakukan oleh [4] yang membahas Penerapan SMOTE untuk mengatasi *imbalance class* dalam klasifikasi objektivitas berita *online* menggunakan algoritma *kNN*. Kemudian [5] melakukan penelitian yang sama tetapi menggunakan data yang berbeda yaitu identifikasi keluhan pelanggan jasa pengiriman barang. Penelitian lainnya juga

dilakukan oleh [6] membahas SMOTE dan metode *Neural Network Backpropagation* untuk menangani data tidak seimbang pada prediksi pemakaian alat kontrasepsi implan.

Penelitian ini bertujuan untuk mengetahui performa dari algoritma *k-Nearest Neighbor* dalam menangani *imbalance class* menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). Hasil akhir yaitu performancy dari kedua teknik ini juga menjadi bahasan yang diangkat pada penelitian. Berdasarkan tujuan tersebut, penelitian ini dapat membantu untuk mengetahui performancy algoritma *k-Nearest Neighbor* dengan SMOTE dalam menangani ketidakseimbangan kelas.

2 Metode Penelitian

Metode yang digunakan dalam penilitian ini yaitu menggabungkan antara *Synthetic Minority Oversampling Technique* (SMOTE) dan *k-Nearest Neighbor* (*kNN*). Dimana metode SMOTE berfungsi untuk menangani ketidakseimbangan kelas dan metode *kNN* berfungsi untuk mengklasifikasikan data kedalam sentimen positif dan negatif berdasarkan data *review* yang telah dikumpulkan. Model penelitian yang diusulkan pada penelitian ini sebagai berikut :



Gambar 1. Diagram Blok Sistem Sentimen Analisis dengan Metode SMOTE dan algoritma *kNN*

Berdasarkan gambar diatas, sistem yang akan dibangun melalui beberapa tahap yaitu pengumpulan data atau *crawl data*, *preprocessing*, *labelling*, dan klasifikasi data menggunakan metode gabungan antara *K-Nearest Neighbor* dan SMOTE untuk mengatasi ketidakseimbangan kelas yang kemudian akan menghasilkan sebuah performancy dari data yang telah diolah. Proses

pengumpulan data atau *crawl data* dalam penelitian ini menggunakan data teks *review*, dimana data teks *review* tersebut diambil menggunakan *library python* yang bernama *google-play-scraper*. Kemudian data yang sudah didapatkan akan di-*preprocessing* dengan 5 tahapan yaitu *case folding*, *tokenizing*, *stopward removal*, *normalization*, dan *stemmer*. Setelah data diproses dalam tahapan *preprocessing* menghasilkan data yang bersih, sehingga dapat dilakukan *labelling* dengan menggunakan *K-Means Clustering* dan menghasilkan 2 *cluster* yaitu positif dan negatif. Selanjutnya data akan diproses dengan teknik SMOTE untuk mengatasi ketidakseimbangan kelas yang terjadi pada dataset. Kemudian dilakukan klasifikasi dengan menggunakan metode *K-Nearest Neighbor*, dimana akan didapatkan hasil akhir dari klasifikasi tersebut dalam sebuah *performancy*. Hal tersebut bertujuan untuk menunjukkan keakuratan dari gabungan antara SMOTE dengan algoritma *K-Nearest Neighbor* yang digunakan.

2.1 *Dataset*

Sumber data yang digunakan pada penelitian ini berasal dari *Google Play Store*. Data *review* diambil menggunakan *crawl data* atau *scrapping data* menggunakan bahasa *python* dengan *library google-play-scrapper* dan menggunakan batasan yaitu regional Indonesia sebanyak 5000 data teks *review*, dan di dalam data tersebut terdiri dari beberapa atribut yaitu userName, score, at, dan content. userName merupakan *user* yang memberikan *review*, *score* merupakan penilaian terhadap aplikasi, yaitu berupa penilaian dari bintang 1–5, at merupakan tanggal saat *review* diunggah, dan content merupakan isi dari *review* yang diunggah *user*. Kemudian data teks *review* yang terkumpul disimpan dengan format .csv.

	userName	score	at	content
0	Dede Fajar	5	2021-12-29 17:55:45	yeeeee dapat helem dari even Rp 0 terimakasih ...
1	Mohammad Harish	1	2021-12-29 16:58:21	Tokopedia ini curang ya.. saya bayar pesanan t...
2	alvin kurniawan	3	2021-12-30 08:19:58	Tolong dong tokopedia. Saya pengguna baru. Mas...
3	Tuan Bosan	1	2021-12-29 23:25:32	Sistem kacau, tidak bisa membuka/memilih metod...
4	Hasan Almacani	4	2021-12-29 12:43:51	Tolong dong buat developer. Apk nya di bikin r...

Gambar 2. *Dataset Review* Aplikasi Tokopedia

2.2 *K-Nearest Neighbor* (*kNN*)

Larose [7] menyatakan bahwa *kNN* adalah algoritma berbasis pembelajaran di mana dataset pelatihan disimpan. *kNN* merupakan salah satu teknik data mining yang paling banyak digunakan dalam klasifikasi masalah *kNN* biasa disebut *k-Memory Based Classification* karena *data training* harus berada di memori pada saat *run-time* [8]. Selain digunakan untuk klasifikasi, algoritma *kNN* juga digunakan untuk estimasi dan prediksi. Perhitungan jarak objek *Euclidean* terhadap *data training* yang diberikan dinyatakan dalam Persamaan 1.

$$d(x,y) = \| x - y \| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

2.3 *Synthetic Minority Oversampling Technique* (SMOTE)

Pendekatan *Synthetic Minority Oversampling Technique* (SMOTE) merupakan teknik untuk menyeimbangkan kelas yang berbeda dengan *oversampling*. Pendekatan SMOTE membuat duplikasi data minor agar seimbang dengan data mayor. Teknik SMOTE mampu mengurangi *overfitting* yang merupakan kelemahan dari teknik *oversampling* [9].

2.4 Performansi

Pengukuran kinerja algoritma klasifikasi pada penelitian ini yaitu dengan menggunakan *confusion matrix. Confusion matrix* menunjukkan hasil identifikasi antara jumlah data prediksi yang benar dan jumlah data yang salah dibandingkan dengan fakta yang dihasilkan [10]. Tabel *confusion matrix* ditunjukkan pada Tabel 1.

Tabel 1. *Confusion Matrix*

		Prediksi	
Aktual		Positif	Negatif
	Positif	TP	FN
	Negatif	FP	TN

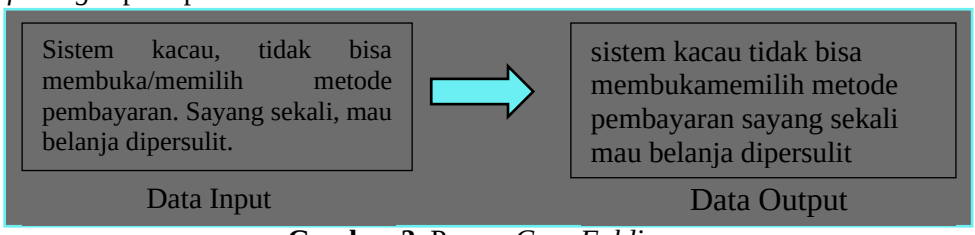
Dimana:
TP (*True Positive)* : data positif *review* Tokopedia yang diprediksi dengan benar.
TN (*True Negative*) : data negatif *review* Tokopedia yang diprediksi dengan salah.
FN (*False Negative*) : data positif *review* Tokopedia yang diprediksi sebagai data negatif.
FP (*False Positive*) : data negatif *review* Tokopedia yang diprediksi sebagai data positif.

3 Hasil dan Pembahasan

Dataset yang sudah terbentuk kemudian dilakukan dengan proses *prepocessing*. Proses *prepocessing* memegang peranan penting dalam analisis sentimen, tujuannya adalah untuk membangun dan mengatur teks dengan menganalisis hubungan-hubungan dan aturan-aturan yang ada pada data teks, baik semi tersruktur maupun tidak terstruktur. Dalam prosesnya, beberapa langkah dilakukan untuk mengubah data ke dalam suatu format, adapun tahapannya antara lain :

3.1 Case Folding

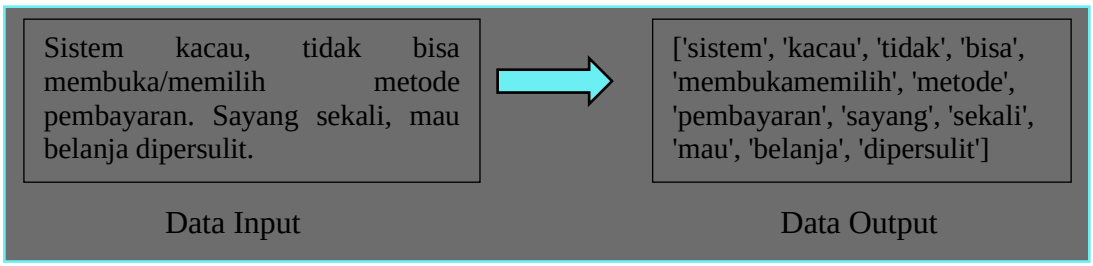
Case folding adalah tahapan untuk mengubah semua huruf dalam data teks menjadi huruf kecil atau *lowercase*, hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter. Proses *case folding* seperti pada Gambar 3.



Gambar 3. Proses *Case Folding*

3.2 Tokenizing

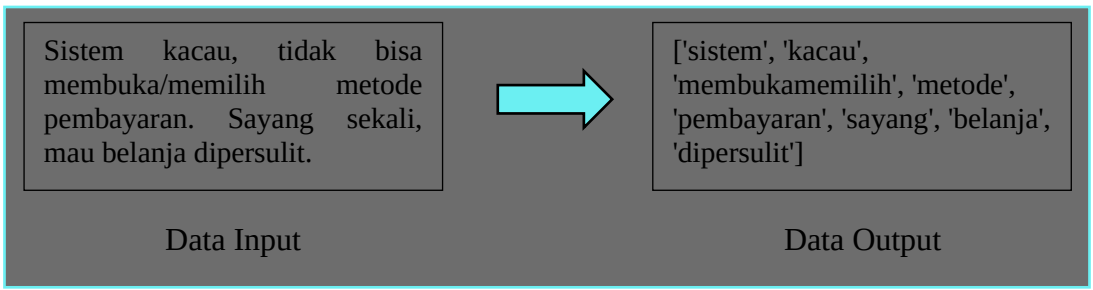
Sebelum data/teks dapat diproses lebih lanjut, maka data tersebut harus disegmentasi ke dalam kata-kata, proses ini dikenal sebagai *tokenizing*. Fase *tokenizing* adalah fase pemotongan string input berdasarkan kata-kata yang menyusunnya atau dengan kata lain memecah kalimat menjadi kata. Proses *tokenizing* dapat dilihat pada Gambar 4.



Gambar 4. Proses *Tokenizing*

3.3 Stopword Removal

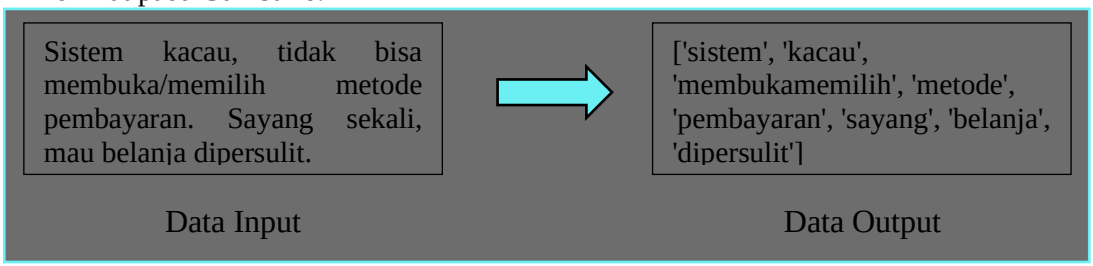
Selain komponen khas yang disebutkan di atas, ulasan aplikasi juga menyertakan banyak singkatan atau kata yang tidak memiliki arti atau tidak relevan. Proses ini menghilangkan kata-kata ini karena tidak mempengaruhi analisis sentiment. Proses *stopword removal* dapat dilihat pada Gambar 5.



Gambar 5. Proses *Stopword Removal*

3.4 Normalization

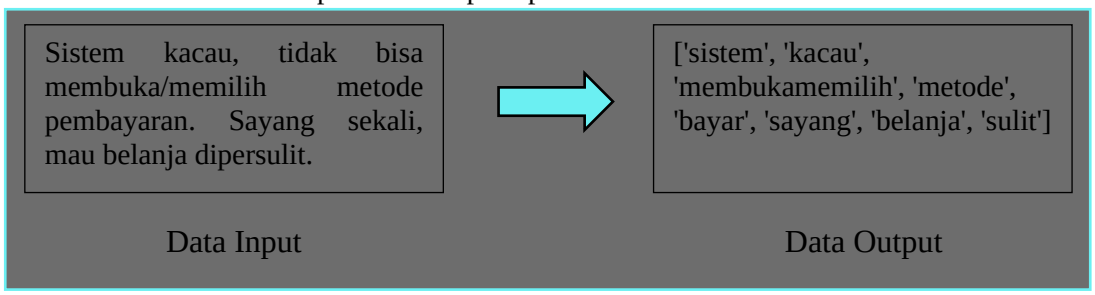
Normalization atau normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Proses ini berfungsi untuk mengatur istilah yang memiliki makna yang sama, tetapi dengan penulisan yang berbeda, umumnya disebabkan oleh kesalahan menulis, penyingkatan kata atau penggunaan bahasa gaul. Proses n*ormalization* dapat dilihat pada Gambar 6.



Gambar 6. Proses *Normalization*

3.5 Stemmer

Fase ini berfungsi untuk mengembalikan kata dalam bentuk dasarnya. Dalam proses pengerjaannya, tahap ini menggunakan *library* dari Sastrawi. Proses *stemmer* dapat dilihat seperti pada Gambar 7.



Gambar 7. Proses *Stemmer*

3.6 Clustering

Fase berikutnya yaitu melakukan *labelling* pada *dataset* menggunakan *K-Means Clustering*. *Cluster* adalah kumpulan dari beberapa objek yang serupa di antara banyak objek data dan berbeda dengan objek dari *cluster* lain, sehingga *clustering* dapat dipahami sebagai upaya untuk mengatur objek menjadi anggota dari grup yang sama dalam beberapa cara.

Langkah-langkah algoritma *K-Means* adalah sebagai berikut [11] :

1. Tentukan nilai *k* atau jumlah *cluster* pada *dataset*.
2. Menentukan nilai pusat (*centroid*). Penentuan nilai *centroid* pada tahap awal dilakukan secara random, sedangkan pada tahap iterasi digunakan rumus seperti dibawah ini:

$$Vij = 1\ Ni \sum Xkj\ Ni\ k=0 \quad (2)$$

Keterangan:

Vij = Centroid rata-rata cluster ke-*i* untuk variabel ke-*j*
Ni = Jumlah anggota cluster ke-*i*
i, k = Indeks dari *cluster*
j = Indeks dari variabel
Xkj = Nilai data ke-*k* variabel ke-*j* untuk *cluster* tersebut

3. Menghitung jarak antara titik *centroid* dengan titik tiap objek menggunakan *Euclidean Distance*. *Euclidean Distance* merupakan jarak garis lurus biasa antara dua titik dalam ruang *Euclidean*, dengan rumus seperti dibawah ini:

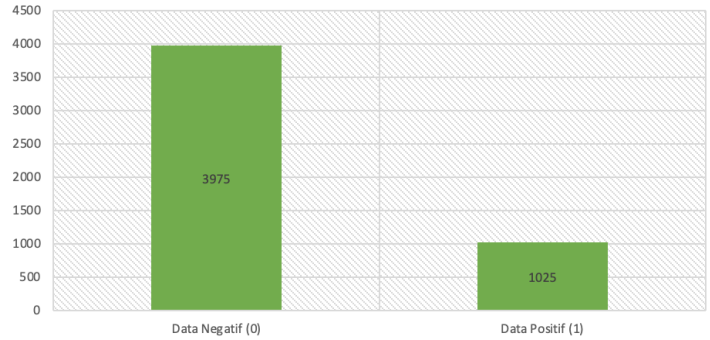
$$De = \sqrt{(xi - si)^2 + (yi - ti)^2} \quad (3)$$

Keterangan:

De = *Euclidean Distance*
i = Banyaknya objek
(*x, y*) = Koordinat objek
(*s, t*) = Koordinat *centroid*

4. Kelompokkan objek berdasarkan jarak ke *centroid* terdekat 5. Ulangi langkah ke-2 hingga ke-4, lakukan iterasi hingga *centroid* bernilai optimal.

Pada *dataset review* aplikasi Tokopedia menghasilkan sebanyak 3975 data negatif dan 1025 data positif seperti pada Gambar 8. Hasil *labelling dataset review* aplikasi Tokopedia dapat dilihat pada Tabel 2.



Gambar 8. Diagram Batang *Clustering Data Review* Aplikasi Tokopedia

Tabel 2. Hasil *Labelling*

Cluster	Teks
0	Tokopedia ini curang ya.. saya bayar pesanan tokopedia saya via alfamart, beberapa jam setelah check out.. setelah beres pembayaran tiba2 uang saya di kembalikan dengan potongan sekitar Rp 1.500.. loh kok bisa. Alesannya mungkin barangnya sudah keburu sold out. Tpi kalo begitu jgan curang dong, dikembalikan 100% seharusnya uang saya.. atau kalau memang sudah sold, ya seharusnya kode pembayaran saya jdi tidak berlaku lagi, atau bagaimana.. Cari untung kok dengan cara yg licik. Gak akan berkah gan
1	Traktiran pengguna baru muluu, pengguna lama? Wkwkwk Semenjak pakai Tokopedia sampai sekarang aku sudah hemat +200rb rupiah. Pengalaman dari semua e-commerce, Tokopedia yang paling berkesanðŸ‘ðŸ‘ðŸ‘ Terima kasih banyak Tokopedia, aplikasinya simpel, tidak ribet, tampilannya sangat rapi, mudah digunakan, intinya very satisfied menggunakan Tokopedia, banyak promo-promo dan diskon, pengiriman cepat, dan ongkirnya murah banget.

3.7 Proses SMOTE

Tahapan dalam melakukan SMOTE dimulai dari menghitung jarak antar data pada data minoritas, selanjutnya menentukan nilai presentase SMOTE kemudian menentukan k terdekat, dan terakhir menghasilkan ringkasan data dengan persamaan berikut[4] :

$$x\_syn = x\_i+(x\_knn-x\_i) \times \delta \quad (4)$$

Keterangan:

x_syn = Data sintesis yang akan diciptakan
x_i = Data yang akan direplikasi
x_knn = Data yang memiliki jarak terdekat dari data yang akan direplikasi
δ = nilai random antara 0 dan 1

3.8 Klasifikasi

Algoritma yang dikaji pada penelitian yaitu *kNN* dan penerapan SMOTE pada *kNN*. Pengujian dilakukan dengan membagi komposisi data latih sebesar

70% dan data uji sebesar 30% dari *dataset*. Hasil klasifikasi kedua metode tersebut dibandingkan untuk memperoleh metode yang tepat dalam menangani ketidakseimbangan kelas data *review* Tokopedia.

```
[15] from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

[16] print(confusion_matrix(y_test, y_pred))

     [[ 126  177]
      [  85 1105]]

[17] print(classification_report(y_test, y_pred))

                   precision    recall  f1-score   support

                0       0.60      0.42      0.49       303
                1       0.86      0.93      0.89      1190

         accuracy                           0.82      1493
        macro avg       0.73      0.67      0.69      1493
     weighted avg       0.81      0.82      0.81      1493


[18] accuracy = accuracy_score(y_test, y_pred)
     print('Accuracy', accuracy)

     Accuracy 0.8245144005358339
```

Gambar 9. Hasil Akurasi Metode *kNN*

Pada Gambar 9. menunjukkan bahwa hasil presisi, *recall*, dan *f1-score* data positif berturut-turut sebesar 0.60, 0.42, dan 0.49. Sedangkan pada data negatif sebesar 0.86, 0.93 dan 0.89. Akurasi yang dihasilkan dengan menggunakan metode *kNN* adalah 82%.

```
[20] from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

[21] print(confusion_matrix(y_train_smote, y_pred))

     [[2477  311]
      [ 240 2548]]

[22] print(classification_report(y_train_smote, y_pred))

                   precision    recall  f1-score   support

                0       0.91      0.89      0.90      2788
                1       0.89      0.91      0.90      2788

         accuracy                           0.90      5576
        macro avg       0.90      0.90      0.90      5576
     weighted avg       0.90      0.90      0.90      5576


[23] accuracy = accuracy_score(y_train_smote, y_pred)
     print('Accuracy', accuracy)

     Accuracy 0.901183644189383
```

Gambar 10. Hasil Akurasi Metode SMOTE-*kNN*

Hasil presisi, *recall*, dan *f1-score* data positif berturut-turut sebesar 0.91, 0.89, dan 0.90 pada metode SMOTE-*kNN*. Sedangkan pada data negatif sebesar

0.89, 0.91 dan 0.90. Akurasi yang dihasilkan dengan menggunakan metode *kNN* dengan SMOTE adalah 90%.

Tabel 3. Hasil Klasifikasi *kNN*

30% : 70%					
	0	**1**	**Accuracy**	**Macro Avg**	**Weighted Avg**
Precission	0.60	0.86	-	0.73	0.81
Recall	0.42	0.93	-	0.67	0.82
F1-Score	0.49	0.89	0.82	0.69	0.81
Support	303	1190	1493	1493	1493
Accuracy			**0.82**		

Tabel 4. Hasil Klasifikasi *SMOTE* dan *kNN*

30% : 70%					
	0	**1**	**Accuracy**	**Macro Avg**	**Weighted Avg**
Precission	0.91	0.89	-	0.90	0.90
Recall	0.89	0.91	-	0.90	0.90
F1-Score	0.90	0.90	0.90	0.90	0.90
Support	2788	2788	5576	5576	5576
Accuracy			**0.90**		

4 Kesimpulan

Penelitian ini berhasil dilakukan dengan membandingkan kinerja *kNN* dan SMOTE-*kNN* pada data tidak seimbang. *Dataset* yang digunakan pada penelitian ini sebesar 5000 *data review* Tokopedia dengan pembagian 3975 data negatif dan 1025 data positif. Metode SMOTE pada *kNN* menunjukkan hasil akurasi yang lebih baik yaitu sebesar 90% dibandingkan hanya menggunakan *kNN* dengan nilai akurasi 82%.

5 Ucapan Terima Kasih

Tim peneliti mengucapkan terima kasih dan penghargaan kepada Direktorat Penelitian dan Pengabdian Kepada Masyarakat (DPPM) Universitas Muhammadiyah Malang atas terselenggaranya pekerjaan ini melalui skema Penelitian Dasar Keilmuan Tahun 2022.

6 Referensi

[1] APJII, “Buletin APJII Edisi 94-September 2021,” *Apjii*, no. September, p.

2, 2021.
[2] R. Y. Endra and D. Hermawan, "Analisis dan Uji Kualitas Pengguna Website Tokopedia.Com Menggunakan Metode Webqual (case : Pengguna Tokopedia.com di Universitas Bandar Lampung)," *Explor. J. Sist. Inf. dan Telemat.*, vol. 8, no. 2, 2017, doi: 10.36448/jsit.v8i2.957.
[3] R. Siringoringo, "Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan k-Nearest Neighbor," *J. ISD*, vol. 3, no. 1, pp. 44–49, 2018.
[4] U. Kasanah, A. N., Muladi, M., & Pujianto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 1, no. 3, pp. 196–201, 2019.
[5] N. Ruhyana and D. Rosiyadi, "Klasifikasi Komentar Instagram untuk Identifikasi Keluhan Pelanggan Jasa Pengiriman Barang dengan Metode SVM dan Naïve Bayes Berbasis Teknik Smote," *Fakt. Exacta*, vol. 12, no. 4, p. 280, 2020, doi: 10.30998/faktorexacta.v12i4.4981.
[6] R. Agustika, "Penerapan Kombinasi SMOTE dan Tomek Links untuk Klasifikasi Data Tidak Seimbang dengan Metode Random Forest," 2021, [Online]. Available: http://etd.repository.ugm.ac.id/penelitian/detail/199065
[7] S.-H. Wu, "Machine Learning Notation," *IEEE Softw.*, vol. 33, pp. 1–2, 2009, doi: 10.1109/MS.2016.114.
[8] E. Alpaydin, "Voting over Multiple Condensed Nearest Neighbors," *Artif. Intell. Rev.*, vol. 11, no. 1–5, pp. 115–132, 1997, doi: 10.1007/978-94-017-2053-3_4.
[9] L. Chen, B. Fang, Z. Shang, and Y. Tang, "Tackling class overlap and imbalance problems in software defect prediction," *Softw. Qual. J.*, vol. 26, no. 1, pp. 97–125, 2018, doi: 10.1007/s11219-016-9342-6.
[10] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 238–242, 2020, doi: 10.1109/ITIS50118.2020.9320958.
[11] Z. Nabila, A. Rahman Isnain, and Z. Abidin, "Analisis Data Mining Untuk Clustering Kasus Covid-19 Di Provinsi Lampung Dengan Algoritma K-Means," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 2, p. 100, 2021, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI