Handling Imbalanced Data pada Prediksi Churn menggunakan metode SMOTE dan KNN Based on Kernel

Handling Imbalanced Data on Churn Prediction using SMOTE and KNN Based on Kernel Methods

Oscar Febri Ramadhan[#1], Adiwijaya[#2], Annisa Aditsania[#3]

#School of Computing, Telkom University
Jl. Telekomunikasi No.01, Terusan Buah Batu, Bandung, Jawa Barat, Indonesia
[1]oscarramadhan.or@gmail.com
[2]adiwijaya@telkomuniversity.ac.id
[3]aaditsania@telkomuniversity.ac.id

Abstrak

Customer churn merupakan masalah umum yang ditemui diperindustrian telekomunikasi. Customer churn didefinisikan sebagai kecenderungan customer berhenti melakukan bisnis dengan suatu perusahaan. Tetapi hanya terdapat sedikit sekali churn customer yang ada. Kekurangan data yang menunjukan bahwa customer tersebut termasuk churn customer menyebabkan masalah imbalanced data. pada tugas akhir ini penulis membuat sebuah sistem yang dapat melakukan penanganan terkait imbalanced data menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*). *Classifier* yang digunakan untuk menentukan suatu customer apakah termasuk churn atau tidak, menggunakan metode *Improve KNN Algorithm Based on Kernel Method*. Metode ini merupakan perkembangan dari metode *KNN Standard*. Dimana pada metode *KNN Standard* proses klasifikasi dilakukan dengan melihat sejumlah *k* tetangga terdekat, dan akan diklasifikasikan berdasarkan jumlah kelas terbanyak pada sejumlah *k* tetangga terdekatnya. *Classifier* tersebut diuji menggunakan 3 fungsi *Kernel* dan 40 kombinasi parameter untuk menemukan performansi tertinggi. Performansi tertinggi yang didapat dari kombinasi parameter tersebut diukur menggunakan *f1-measure* dan akurasi secara berurut pada data tanpa smote, smote 1:3, smote 1:2, smote 3:4, dan smote 1:1, yaitu: 0,314 & 97,58%, 0,449 & 94,55%, 0,413 & 93,70%, 0,382 & 92,74% dan 0,363 & 92,08%.

Kata Kunci: *Churn Prediction, Over-sampling,* SMOTE (*Synthetic Minority Over-sampling Technique*), *Improve KNN Algorithm Based on Kernel Method*.

I. Pendahuluan

Customer churn merupakan masalah umum yang ditemui diperindustrian telekomunikasi. *Customer churn* didefinisikan sebagai kecenderungan pelanggan untuk berhenti melakukan bisnis dengan suatu perusahaan dalam jangka waktu tertentu [7]. Dapat dikatakan bahwa *Customer churn* merupakan suatu pola yang menunjukkan bahwa seorang *customer* akan berhenti menggunakan jasa suatu perusahaan. *Customer churn* telah menjadi masalah yang signifikan dan merupakan salah satu tantangan utama yang harus dihadapi banyak perusahaan di dunia [7].

Untuk dapat bertahan dikompetisi pasar, beberapa perusahaan mulai mencoba untuk memprediksi *customer churn* menggunakan pendekatan *Data Mining*. Pendekatan *Data Mining* digunakan untuk membuat model prediksi yang dapat membantu perusahaan dalam memahami kebiasaan *customer* [9]. Sehingga perusahaan dapat mengimplementasikan CRM (*Customer Relationship Management*) yang

tepat untuk membuat *customer* bertahan atau dengan kata lain, menyelamatkan pendapatan perusahaan [9].

Tetapi hanya terdapat sedikit sekali *churned customer* dari keseluruhan data yang ada [3]. Data yang mendeskripsikan *customer churn* biasanya memiliki persentasi yang kecil. Kekurangan data yang menunjukan bahwa *customer* tersebut termasuk *churned customer* menyebabkan masalah *imbalanced data*. *Imbalanced data* terjadi karena sedikitnya *churned customer* dari keseluruhan data yang ada [9]. *Imbalanced data* menyebabkan sulitnya mengembangkan model prediksi yang bagus.

Salah satu riset yang dilakukan oleh Wang Xueli mengenai klasifikasi menggunakan *classifier KNN Standard* dan *KNN Based on Kernel*, menggunakan 12 kombinasi parameter menghasilkan akurasi tertinggi pada *KNN Based on Kernel* [1]. *Sampling* merupakan sebuah metode pengambilan sebagian data dari populasi [2]. Pada kasus berbeda, Nitesh V. Chawla melakukan riset untuk menangani permasalahan *churn* dengan menerapkan salah satu metode *oversampling* yaitu SMOTE dan didapatkan bahwa data yang dilakukan SMOTE menghasilkan performansi yang lebih baik dibandingkan data tanpa SMOTE [2]. Pada proses *handling imbalanced data* penulis akan menggunakan metode SMOTE (*Synthetic Minority Over-sampling Technique*).

SMOTE merupakan salah satu teknik *sampling*. Dalam proses klasifikasi untuk menentukan apakah suatu *customer* termasuk *churn* atau tidak, akan digunakan metode *Improved KKN Algorithm based on Kernel Method*. Metode ini merupakan pengembangan dari metode *KNN Standard*, dimana pada metode *KNN Standard* proses klasifikasi dilakukan dengan melihat sejumlah *k* tetangga terdekat, dan akan ditentukan kelasnya berdasarkan jumlah kelas terbanyak pada sejumlah *k* tetangga terdekatnya [1]. *KNN Based on Kernel* menggunakan *weight* dalam menentukan prediksi kelasnya, sehingga prediksi sistem yang dihasilkan lebih akurat [1].

II. Penulisan Terkait

Salah satu riset yang dilakukan oleh Wang Xueli mengenai klasifikasi menggunakan *classifier KNN Standard* dan *KNN Based on Kernel* menggunakan empat buah dataset dengan jumlah atribut secara berurut yaitu 9, 13, 8 dan 11, dengan menggunakan 12 kombinasi parameter menghasilkan akurasi tertinggi pada *KNN Based on Kernel* yaitu 86,7%[1].

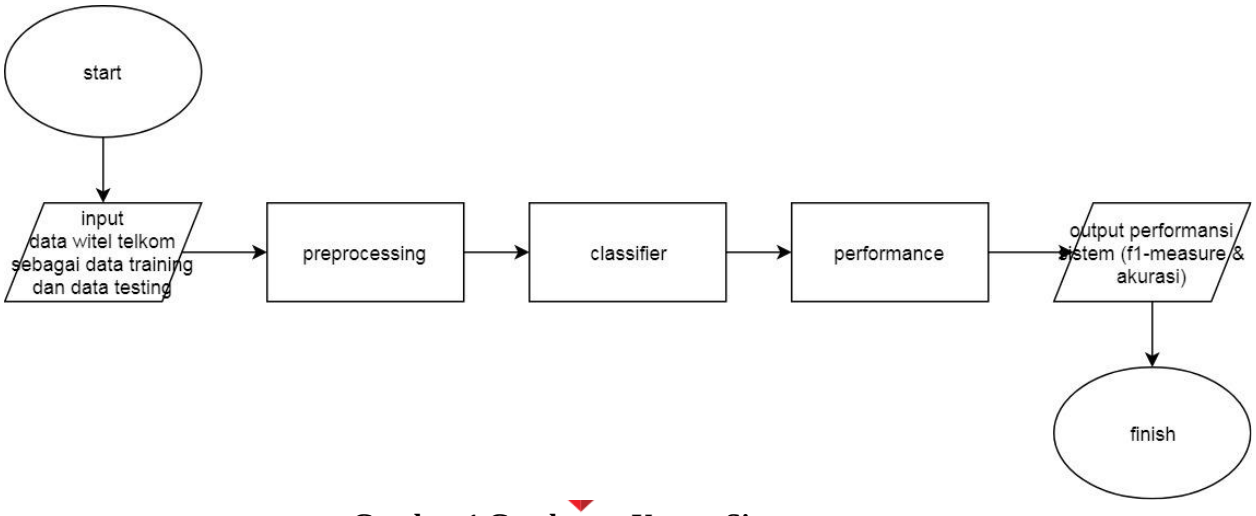
Pada kasus berbeda, Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O.Hall, dan W. Philip Kegelmeyer melakukan riset untuk menangani permasalahan *churn* dengan menerapkan salah satu metode *oversampling* yaitu SMOTE untuk menangani permasalahan perbandingan data minor dengan data mayor yang jauh berbeda atau yang disebut *imbalanced data*. Didapatkan bahwa data yang dilakukan SMOTE menghasilkan performansi yang lebih baik dalam memprediksi kelas dari data minor dibandingkan data tanpa SMOTE [2].

Pada kasus yang sama, Veronikha Effendy, Adiwijaya dan Z.K.A. Baizal melakukan riset serupa untuk menangani permasalahan *churn* dengan menggunakan *combined sampling* (*undersampling* dan salah satu metode *oversampling* yaitu SMOTE) untuk menangani permasalahan *imbalanced data* dan menggunakan *classifier weight random forest* dan didapatkan hasil tertinggi yaitu dengan kombinasi parameter *undersampling* 8/10 + SMOTE 50x dengan *f1-measure* 0,66 [5].

III. Metodologi dan Desain Sistem

A. Gambaran Umum Sistem

Berikut merupakan gambaran umum sistem klasifikasi yang digunakan:



Gambar 1 Gambaran Umum Sistem

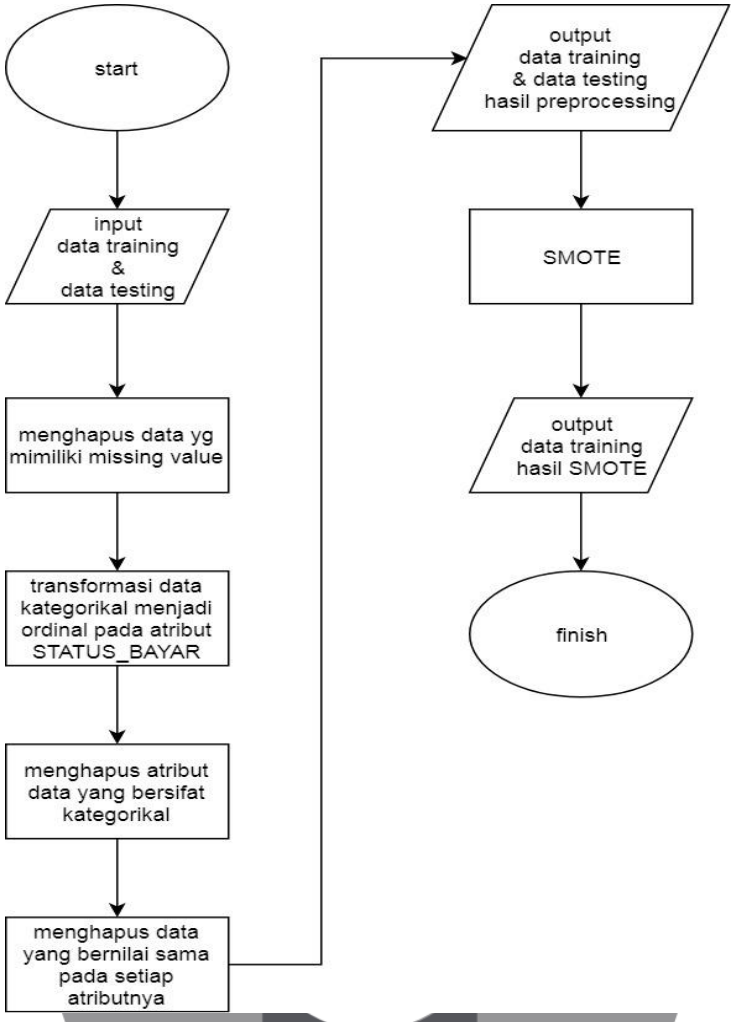
Secara umum ada 3 proses utama dari sistem klasifikasi yang dibuat yaitu *preprocessing*, *classifier*, dan *performance*.

B. Analisis Data

Data yang digunakan yaitu data WITEL Telkom regional 7,dimana data yang digunakan memiliki 55 buah atribut dan 160217 record data pada data training, dan 55 buah atribut dan 40054 record data pada data testing. Customer dikategorikan churn direpresentasikan dengan 1, dan 0 jika customer dikategorikan non-churn pada atribut terakhir dari data yang diberikan, yaitu atribut churn. Atribut pada data yang digunakan yaitu SND, SEGMEN_ID, UMUR_PLG, PAKET_SPEEDY_ID, PREFIX_ND, WITEL, TAG_N (12 atribut), STATUS_BAYAR_N (12 atribut), GGN_N (12 atribut), USAGE_N (12 atribut), CHURN. Atribut data yang digunakan hanya atribut yang bersifat numerik saja.

C. *Preprocessing*

Berikut merupakan tahapan yang dilakukan pada proses preprocessing:

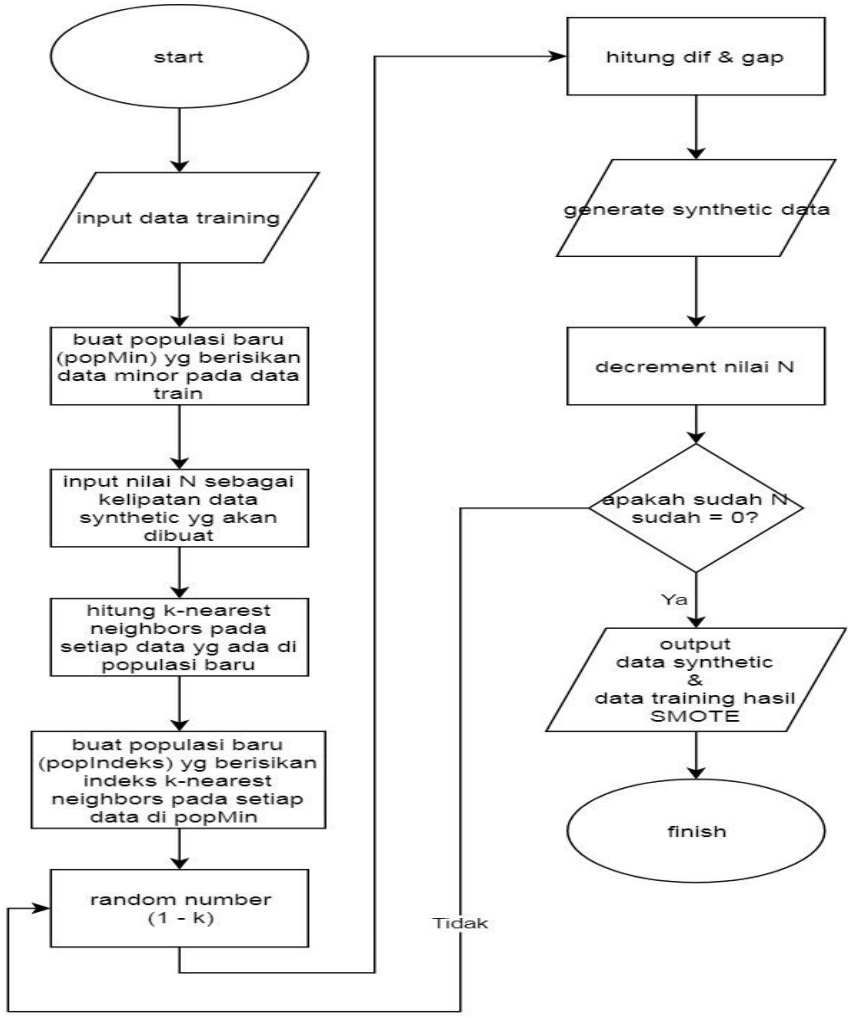


Gambar 2 Tahap Preprocessing

Tahap *preprocessing* dilakukan untuk menangani berbagai masalah pada data agar data siap digunakan. Tahap *preprocessing* yang dilakukan yaitu: menghapus data yang memiliki *missing value*, melakukan transformasi data kategorikal menjadi *ordinal* pada 12 atribut STATUS_BAYAR_N, menghapus atribut data yang bersifat kategorikal, menghapus data yang bernilai sama pada setiap atributnya. Lalu tahap *preprocessing* akan menghasilkan output berupa *data training* dan *data testing* baru, kemudian akan dilakukan tahap SMOTE menggunakan *data training* hasil *preprocessing*.

- SMOTE

  Berikut ini merupakan tahap yang dilakukan pada proses SMOTE:

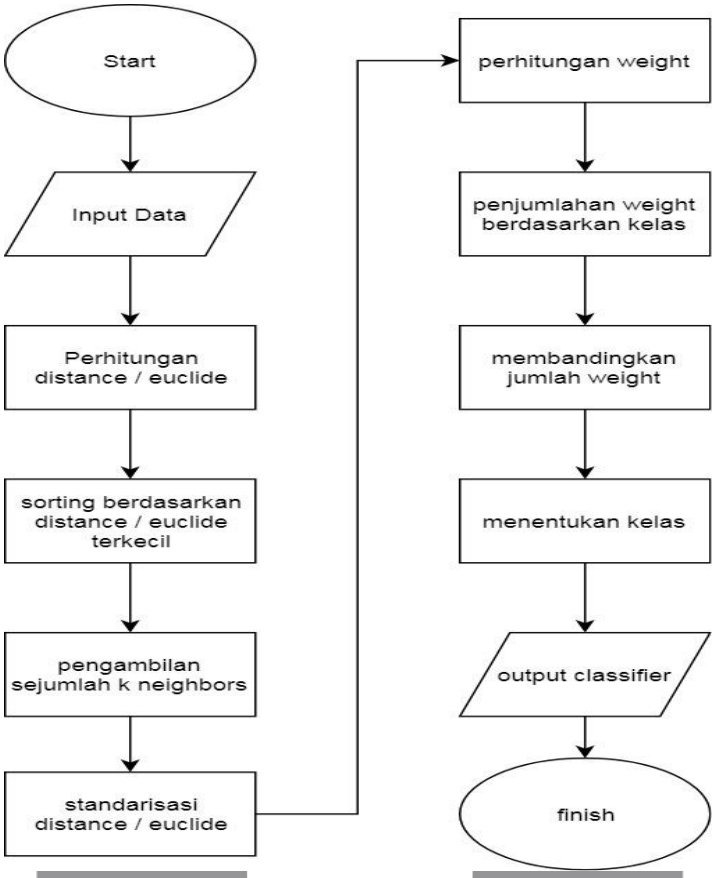


Gambar 3 Tahap SMOTE

Tahap SMOTE dilakukan untuk menangani masalah *imbalanced data* pada *data training* yang akan digunakan. Pada tahap ini, akan dilakukan pembuatan *data synthetic* baru dengan kelas 1 (churn). SMOTE dilakukan pada *data training* hasil *preprocessing*, dan akan menghasilkan *output* berupa *data synthetic* yang telah dibuat, dan *data training* baru hasil SMOTE.

D. *Classifier*

Berikut ini merupakan tahap yang dilakukan pada *classifier* yang dibuat:



Gambar 4 Tahap Classifier

Tahap pada *Classifier* yang dibuat menerima inputan berupa *data training* dan *data testing* hasil *preprocessing*. *Data training* yang menjadi inputan berupa *data training* hasil *preprocessing* tanpa SMOTE dan menggunakan SMOTE. *Classifier* yang dibuat akan menghasilkan *output* berupa kelas hasil prediksi, yang kemudian akan digunakan untuk mengukur performansi dari sistem yang akan dibuat.

E. *Performance*

Analisa terkait performansi model prediksi yang telah dibuat akan dilakukan dengan cara berikut:

Tabel 1 Confusion Matrix

	Predicted Positive	*Predicted Negative*
Actual Positive	TP	FN
Actual Negative	FP	TN

Prediksi data yang menghasilkan *churn* dan pada data *training* merupakan *churn*, direpresentasikan sebagai TP (*true positive*). Prediksi data yang menghasilkan *churn*, tetapi pada data training merupakan *non-churn*, direpresentasikan sebagai FN (*false negative*). Prediksi data yang menghasilkan *non-churn*, tetapi pada data *training* merupakan *churn*, direpresentasikan sebagai FP (*false positif*). Prediksi data yang menghasilkan *non-churn* dan pada data *training* merupakan *non-churn*, direpresentasikan sebagai TN (*true negative*).

Proses perhitungan *f1-measure:*

$$F = \frac{2TP}{2TP + FP + FN} \quad (1)$$

Proses perhitungan akurasi:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

IV. Hasil dan Analisa

A. Hasil

1. Membandingkan akurasi dan *f1-measure* berdasarkan *classifier* yang digunakan pada data *training* tanpa SMOTE.

Tabel 2 Akurasi & *f1-measure* menggunakan *classifier KNN Standard* & *KNN Kernel* pada data tanpa SMOTE

k	*KNN Standard*		*KNN Kernel Triangular*	
	akurasi	*f1-measure*	akurasi	*f1-measure*
3	97.31%	0.295	96.55%	0.297
5	97.55%	0.293	97.06%	0.298
7	97.64%	0.281	97.39%	0.310
9	97.65%	0.257	97.53%	0.312

KNN Kernel menghasilkan *f1-measure* yang lebih besar karena dalam menentukan kelas dilihat berdasarkan jumlah *weight* terbesar pada masing-masing kelasnya, sehingga semakin dekat jarak tetangganya, maka semakin besar pula weight yang dimiliki tetangga tersebut. *Classifier* yang digunakan menghasilkan *f1-measure* yang kecil karena data *training* yang digunakan memiliki data yang *imbalanced* pada masing-masing kelasnya.

2. Membandingkan akurasi dan *f1-measure* berdasarkan data *training* yang digunakan.

Tabel 3 Akurasi & *f1-measure* pada *data train* tanpa SMOTE

Triangular										
k	akurasi					*f1-measure*				
	Data1	Data2	Data3	Data4	Data5	Data1	Data2	Data3	Data4	Data5
4	96.86%	94.56%	93.70%	92.70%	92.00%	0.296	0.445	0.413	0.381	0.362
6	97.25%	94.55%	93.47%	92.42%	91.59%	0.307	0.448	0.406	0.374	0.353
8	97.48%	94.55%	93.38%	92.14%	91.30%	0.313	0.449	0.404	0.366	0.345

Dapat dilihat bahwa *f1-measure* yang dihasilkan berdasarkan data2 memiliki nilai f1-measure tertinggi, karena data 2 memiliki jumlah data churn yang lebih banyak dibandingkan dengan data1. Tetapi pada data3, data4 dan data5 nilai *f1-measure* yang dihasilkan turun, disebabkan kesalahan prediksi pada data yang seharusnya *negative* namun diprediksi *positive* / *False Positive* (FP), karena pada data3, data4 dan data5 memiliki jumlah kelas *churn* yang jauh lebih banyak dibandingkan data2.

3. Membandingkan *f1-measure* berdasarkan beberapa nilai *neighbors*.

Tabel 4 *f1-measure* berdasarkan beberapa nilai *neighbors*

Triangular					
k	*f1-measure*				
	Data1	Data2	Data3	Data4	Data5
3	0.297	0.440	0.410	0.382	0.362
4	0.296	0.445	0.413	0.381	0.362
5	0.298	0.447	0.409	0.376	0.361
6	0.307	0.448	0.406	0.374	0.353
7	0.310	0.449	0.407	0.372	0.349
8	0.313	0.449	0.404	0.366	0.345
9	0.312	0.447	0.402	0.362	0.341
10	0.314	0.447	0.401	0.360	0.336

Dapat dilihat bahwa *f1-measure* yang dihasilkan berdasarkan jumlah *neighbor*, pada data1 semakin besar jumlah *neighbors* (nilai *k*) yang digunakan, makan semakin besar pula *f1-measure* yang didapat. Namun pada data3, data4 dan data5 semakin besar jumlah *neighbors* (nilai *k*) pada umumnya menyebabkan *f1-measure* menurun, ini disebabkan bertambahnya nilai FP jauh lebih besar dibandingkan bertambahnya nilai TP. Nilai FP bertambah jauh lebih besar, dikarenakan pada data3, data4 dan data5 memiliki data minor yang jauh lebih banyak dibandingkan data1. Karena data3, data4 dan data5 merupakan data yang telah dilakukan SMOTE.

4. Membandingkan *f1-measure* berdasarkan fungsi *Kernel*, yaitu: *Triangular*, *Epanechnikov* dan *Gaussian*.

Tabel 5 *f1-measure* berdasarkan fungsi *kernel*

k	*Triangular*			*Epanechnikov*			*Gaussian*		
	f1-measure			*f1-measure*			*f1-measure*		
	Data1	Data2	Data3	Data1	Data2	Data3	Data1	Data2	Data3
4	0.296	0.445	0.413	0.296	0.443	0.413	0.302	0.438	0.403
6	0.307	0.448	0.406	0.305	0.444	0.406	0.297	0.431	0.384
8	0.313	0.449	0.404	0.308	0.446	0.402	0.285	0.376	0.376

Dapat dilihat bahwa *f1-measure* yang dihasilkan oleh fungsi *Triangular* dan *Epanechnikov* memiliki nilai *f1-measure* yang lebih tinggi dibandingkan dengan fungsi *Gaussian*, *f1-measure* yang didapat menggunakan fungsi *Triangular* dan *Epanechnikov* memiliki nilai yang tidak jauh berbeda, sedangkan *f1-measure* yang dihasilkan fungsi *Gaussian* terlihat cukup jauh berbeda jika dibandingkan dengan fungsi *Triangular* dan *Epanechnikov*, karena dari ketiga fungsi tersebut, fungsi *Gaussian* memiliki perhitungan *weight* yang sangat berbeda jika dibandingkan dengan kedua fungsi lainnya.

B. Analisis Keseluruhan

Hasil pengujian diatas yang menampilkan hasil pengujian dari setiap skenario, bahwa pada data yang dilakukan SMOTE pada umumnya menyebabkan bertambahnya nilai *f1-measure* dibandingkan dengan data yang tidak dilakukan SMOTE. Pengaruh data menggunakan SMOTE menyebabkan bertambahnya jumlah nilai *True Positive* (TP) menjadi jauh lebih besar, namun juga menyebabkan kesalahan prediksi kelas negatif menjadi kelas positif / *False Positive* (FP) yang besar pula. Pada umumnya setiap data yang dilakukan SMOTE menyebabkan bertambahnya nilai FP menjadi lebih besar ketika bertambahnya nilai *k* pada *classifier* yang membuat nilai *f1-measure* turun.

Dapat dilihat pada hasil pengujian diatas, semakin banyak data *synthetic* yang dibuat, semakin banyak pula kesalahan prediksi kelas *negative* menjadi *positive*, sehingga metode SMOTE ini memiliki batas tertentu dalam membuat data *synthetic* untuk mendapatkan f1-measure terbaik. Pada skenario pengujian yang dilakukan, perbandingan data minor: data mayor yang terbaik adalah perbandingan data minor : data mayor (1:3).

V. Kesimpulan dan Saran

Metode *KNN Based on Kernel* menghasilkan nilai *f1-measure* lebih tinggi dibandingkan *KNN Standard*, disebabkan penentuan kelas pada *KNN Kernel* berdasarkan *weight* dari tetangga terdekatnya. Metode SMOTE dapat meningkatkan *f1-measure* pada permasalahan *churn*, tetapi memiliki batas tertentu pada banyaknya data *synthetic* yang dibuat agar mendapatkan *f1-measure* terbaik. Dari ketiga skenario data yang dilakukan SMOTE berdasarkan hasil pengujian diketahui bahwa perbandingan data minor berbanding data mayor paling bagus adalah data minor : data mayor (1:3) karena memiliki nilai *f1-measure* paling tinggi diantara empat jenis skenario data keseluruhan dengan nilai *k*=8, yaitu: 0,449 pada fungsi *Triangular*, 0,449 pada fungsi *Epanechnikov* dan 0,421 pada fungsi *Gaussian*.

Semakin banyak jumlah *neighbors* (nilai *k*) yang digunakan pada *classifier* pada data yang dilakukan SMOTE pada umumnya menyebabkan semakin kecil akurasi dan *f1-measure* yang didapat. Semakin besar *data synthetic* yang akan dibuat dapat mempengaruhi nilai *True Positive* (TP) dan *False Positive*

(FP) semakin besar. Dari ketiga fungsi yang diujikan, fungsi *Triangular* dan *Epanechnikov* memiliki hasil *f1-measure* yang lebih tinggi dibandingkan fungsi *Gaussian*.

Untuk kedepannya mungkin dapat diterapkan metode *handling imbalanced data* yang lain untuk menangani permasalahan serupa. Juga menerapkan metode klasifikasi yang lain untuk membandingkan performansi classifier yang digunakan. Melakukan *feature selection*, jika jumlah atribut pada data yang didapat sangat banyak.

Daftar Pustaka

[1] Xueli, W., Zhiyong, J. and Dahai, Y., 2015, September. *An Improved KNN Algorithm Based on Kernel Methods and Attribute Reduction*. In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)* (pp. 567-570). IEEE.

[2] Chawla, N.V., Lazarevic, A., Hall, L.O. and Bowyer, K.W., 2003. SMOTEBoost: Improving prediction of the minority class in boosting. In*Knowledge Discovery in Databases: PKDD 2003* (pp. 107-119). Springer Berlin Heidelberg.

[3] Burez, J. and Van den Poel, D., 2009. *Handling class imbalance in customer churn prediction*. *Expert Systems with Applications*, *36*(3), pp.4626-4636.

[4] Shaaban, E., Helmy, Y., Khedr, A. and Nasr, M., 2012. A proposed *churn* prediction model. *IJERA*, *2*, pp.693-697.

[5] Effendy, V., Adiwijaya and Baizal, Z.A., 2014, May. *Handling* imbalanced data in *customer churn* prediction using combined sampling and weighted random forest. In *Information and Communication Technology (ICoICT), 2014 2nd International Conference on* (pp. 325-330). IEEE.

[6] Yap, B.W., Rani, K.A., Rahman, H.A.A., Fong, S., Khairudin, Z. and Abdullah, N.N., 2014. *An application of oversampling, undersampling, bagging and boosting in handling imbalanced* datasets. In *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)* (pp. 13-22). Springer Singapore.

[7] Lazarov, V. and Capota, M., 2007. *Churn* prediction. *Bus. Anal. Course. TUM Comput. Sci.*

[8] Dwiyanti, E., Adiwijaya, and Ardiyanti, A., 2016, August. *Handling Imbalanced Data in Churn Prediction Using RUSBoost and Feature Selection* (Case Studi: PT. Telekomunikasi Indonesia Regional 7). *In International Conference on Soft Computing and Data Mining* (pp. 376-385). Springer, Cham.

[9] Kantardzic, M., 2011. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons.

[10] Friedman, J., Hastie, T. and Tibshirani, R., 2001. *The elements of statistical learning* (Vol. 1). Springer, Berlin: Springer series in statistics.

[11] D. M. Maharaj, "Evaluating *Customer* Relations in The Cell phone Industry," *IJBSM (International Journal for Business, Srategy & Management) Vol 1 No1* , 2011.

[12] Adiwijaya. 2014. Aplikasi Matriks dan Ruang Vektor. Graha Ilmu, Yogyakarta.