

PERBAIKAN AKURASI FUZZY K-NEAREST NEIGHBOR IN EVERY CLASS MENGGUNAKAN FUNGSI KERNEL

Harunur Rosyid¹⁾, Eko Prasetyo²⁾, Soffiana Agustin³⁾

^{1), 3)} Teknik Informatika Universitas Muhammadiyah Gresik
Jl. Sumatra 101 GKB, Gresik, 61121

²⁾ Teknik Informatika Universitas Bhayangkara Surabaya
Jl. A. Yani 114, Surabaya, 60231
email : fbharun7@gmail.com¹⁾, eko1979@yahoo.com²⁾, asoffi@gmail.com³⁾

Abstrak

Salah satu algoritma klasifikasi yaitu K-NN menjadi algoritma yang sangat populer karena kesederhanaan dalam proses penggunaannya. Kelemahan K-NN adalah harus menggunakan seluruh data latih untuk melakukan proses prediksi. Banyak variasi-variasi K-NN yang diusulkan untuk melakukan perbaikan kinerja K-NN, baik dari sisi akurasi prediksi maupun waktu komputasi yang digunakan selama proses prediksi, seperti Fuzzy K-Nearest Neighbor (KFK-NN), dan Fuzzy K-NN in every Class (FK-NNC). Dalam makalah ini dipaparkan hasil penelitian berupa penggunaan fungsi kernel dalam Kernel Based FK-NNC (KB-FK-NNC) untuk meningkatkan kinerja akurasi prediksi. Hasil pengujian menunjukkan bahwa penggunaan kernel cukup baik untuk digunakan dalam membantu meningkatkan akurasi meskipun peningkatannya kurang signifikan. Akurasi yang didapatkan oleh KB-FK-NNC juga terlihat stabil untuk beberapa pilihan K dan \dagger yang digunakan dalam pengujian.

Kata kunci :

fuzzy, K-Nearest Neighbor, class, kernel, klasifikasi.

1. Pendahuluan

Algoritma klasifikasi yang sangat populer penggunaannya karena kesederhanaan algoritma adalah Nearest Neighbor (kadang disebut juga K-Nearest Neighbor / K-NN). K-NN melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain [6]. Prinsip sederhana yang diadopsi oleh algoritma K-NN adalah: "Jika suatu hewan berjalan seperti bebek, bersuara kwek-kwek seperti bebek, dan penampilannya seperti bebek, maka hewan itu mungkin bebek". K-NN merupakan algoritma yang menggunakan seluruh data latih untuk melakukan proses klasifikasi (*complete storage*). Bisa dipastikan bahwa ketika harus memprediksi dengan komposisi data latih yang sangat besar maka proses prediksi menjadi sangat lama. Hal inilah yang menjadi kelemahan K-NN dibanding metode-metode klasifikasi yang lain.

K-NN versi klasik melakukan prediksi secara tegas pada data uji berdasarkan perbandingan K tetangga terdekat, sedangkan Keller membuat pendekatan lain yang dalam melakukan prediksi juga berdasarkan K tetangga terdekat tapi tidak secara tegas memprediksi kelas yang harus dikuti oleh data uji, pemberian label kelas data uji pada setiap kelas dengan memberikan nilai keanggotaan seperti halnya teori himpunan fuzzy [1]. Algoritma Fuzzy K-Nearest Neighbor (FK-NN) tersebut diperkenalkan oleh Keller et al. [2] dengan mengembangkan K-NN yang digabungkan dengan teori fuzzy dalam pemberian label kelas pada data uji yang diprediksi. Salah satu perbaikan yang dilakukan pada FK-NN untuk meningkatkan akurasinya adalah algoritma Adaptive Fuzzy K-Nearest Neighbor (AFK-NN) diusulkan oleh Chen et al. [1] dengan mengembangkan FK-NN yang merupakan gabungan teori fuzzy dengan optimasi parameter K dan fuzzy strength m dalam pemberian label kelas pada data uji yang diprediksi.

FK-NN juga mempunyai banyak variasi selain AFK-NN [1], diantaranya ada Kernel-based Fuzzy K-Nearest Neighbor (KFK-NN) yang diusulkan oleh Wu dan Zhou [9] untuk menangani kasus prediksi data yang non-linear menggunakan fungsi kernel. Weinberger [8] mengusulkan penggunaan metrik jarak Mahalanobis untuk digunakan sebagai basis pelatihan K-NN dengan data latih untuk menuju K tetangga terdekat yang sama kelasnya dengan meminimalkan fungsi obyektif. Prasetyo [5] mengusulkan Fuzzy K-NN in every Class (FK-NNC) untuk memperbaiki akurasi kinerja pada saat prediksi. Umumnya, metode-metode yang diusulkan tersebut bertujuan untuk memperbaiki akurasi atau mengurangi jumlah data latih yang dilibatkan pada saat prediksi.

Dalam makalah ini, dipaparkan hasil penelitian untuk pengembangan metode FK-NNC yang dikombinasikan dengan penggunaan fungsi kernel untuk menangani kasus data yang distribusinya non-linear. Karena kernel dalam SVM terbukti mampu memberikan kinerja akurasi yang lebih baik pada data set non-linear maka dalam penelitian yang dilakukan penulis digunakan fungsi kernel pada saat proses prediksi. Harapannya adalah meningkatkan akurasi prediksi FK-NNC yang lebih baik dari metode-metode yang diusulkan sebelumnya.

Penelitian ini juga melakukan pengamatan korelasi penggunaan fungsi kernel dengan peningkatan akurasi prediksi.

Sistematika dalam makalah ini terbagi menjadi 7, bagian 1 membahas pendahuluan, bagian 2 membahas ringkasan metode terkait, bagian 3 membahas kerangka kerja metode, bagian 4 membahas hasil pengujian dan analisis, bagian 5 membahas simpulan dan saran, dan bagian 7 daftar pustaka

2. Tinjauan Pustaka dan Metode Terkait

2.1. K-Nearest Neighbor

Pada algoritma K-NN [6], data berdimensi N , dilakukan perhitungan jarak dari data uji tersebut ke data latih, nilai jarak ini kemudian digunakan sebagai nilai kedekatan/ketidakmiripan antara data uji dengan data latih. Nilai K pada K-NN menunjukkan K -data latih sebagai tetangga terdekat dari data uji.

Pada algoritma K-NN, sebuah data uji $z = (x', y')$, dimana x' ada vektor/atribut data uji, sedangkan y' adalah label kelas data uji yang belum diketahui, kemudian dihitung jarak (atau ketidakmiripan) data uji ke setiap data latih $d(x', x)$, kemudian memilih K tetangga terdekat pertama dalam D_z . Setelah itu dihitung jumlah data latih yang mengikuti kelas yang ada dari K tetangga tersebut. Kelas dengan data terbanyak yang mengikutinya menjadi kelas pemenang yang diberikan sebagai label kelas pada data uji y' .

Algoritma Prediksi dengan K-NN

1. $z = (x', y')$, adalah data uji dengan vektor x' dan label kelas y' yang belum diketahui
2. Hitung jarak $d(x', x)$, jarak diantara data uji z ke setiap vektor data latih, simpan dalam D
3. Pilih $D_z \subseteq D$, yaitu K tetangga terdekat dari z
4. Berikan label kelas dari data latih tetangga terdekat yang jumlahnya mayoritas,
 $y' = \arg \max_v \sum_{(x_i, y_i) \in D_z} I(v = y_i)$

Salah satu masalah yang dihadapi K-NN adalah pemilihan nilai K yang sulit, cara voting mayoritas dari K -tetangga untuk nilai K yang besar bisa mengakibatkan distorsi data yang besar, jika K terlalu kecil bisa menyebabkan algoritma terlalu sensitif terhadap noise. Meskipun begitu, sebenarnya penentuan nilai K yang terbaik juga dapat dilakukan dengan menggunakan teknik cross-validation [6].

2.2. Fuzzy K-Nearest Neighbor

Metode K-NN telah mengalami banyak perbaikan, salah satunya oleh Chen et al. [1] yang melakukan optimasi pada parameter nilai K dan fuzzy strength m menggunakan metode Particle Swarm Optimization. Dalam sistem fuzzy, sebuah data dapat mempunyai nilai keanggotaan dari sejumlah kelas dengan derajat keanggotaan dalam interval 0 sampai 1. Teori himpunan fuzzy men-generalisasi teori K-NN klasik dengan

mendefinisikan nilai keanggotaan sebuah data pada masing-masing kelas. Formula yang digunakan [1][3]:

$$u(x, y_i) = \frac{\sum_{j=1}^K u(x_j, y_i) * d(x, x_j)^{\frac{-2}{(m-1)}}}{\sum_{j=1}^K d(x, x_j)^{\frac{-2}{(m-1)}}} \quad (1)$$

Dimana $u(x, y_i)$ adalah nilai keanggotaan data x ke kelas c_i , K adalah jumlah tetangga terdekat yang digunakan, $u(x_k, y_i)$ adalah nilai keanggotaan data tetangga dalam K tetangga pada kelas y_i , nilainya 1 jika data latih x_k milik kelas c_i atau 0 jika bukan milik kelas c_i , untuk $d(x, x_k)$ adalah jarak dari data x ke data x_k dalam K tetangga terdekat, m adalah bobot pangkat (*weight exponent*) yang besarnya $m > 1$.

Nilai keanggotaan suatu data pada kelas sangat dipengaruhi oleh jarak data itu ke tetangga terdekatnya, semakin dekat ke tetangganya maka semakin besar nilai keanggotaan data tersebut pada kelas tetangganya, begitu pula sebaliknya. Jarak tersebut diukur dengan N dimensi (fitur) data.

Pengukuran jarak (ketidakmiripan) dua data yang digunakan dalam FK-NN digeneralisasi dengan [1][3]:

$$d(x_i, x_j) = \left(\sum_{l=1}^N |x_{il} - x_{jl}|^p \right)^{\frac{1}{p}} \quad (2)$$

Dimana N adalah dimensi (jumlah fitur) data. Untuk p adalah penentu jarak yang digunakan, jika $p=1$ maka jarak yang digunakan adalah Manhattan, jika $p=2$ maka jarak yang digunakan adalah Euclidean, jika $p=\infty$ maka jarak yang digunakan adalah Chebyshev.

Meskipun FK-NN menggunakan nilai keanggotaan untuk menyatakan keanggotaan data pada setiap kelas, tetapi untuk memberikan keluaran akhir, FK-NN tetap harus memberikan kelas akhir hasil prediksi, untuk keperluan ini, FK-NN memilih kelas dengan nilai keanggotaan terbesar pada data tersebut.

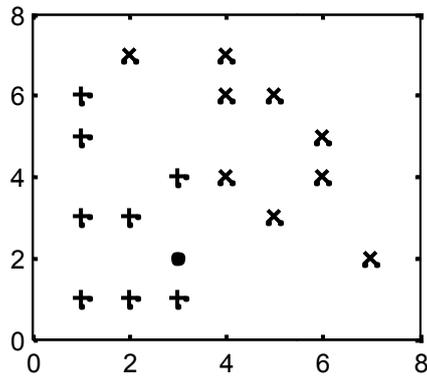
Algoritma Prediksi dengan FK-NN

1. Normalisasikan data menggunakan nilai terbesar dan terkecil data pada setiap fitur.
2. Cari K tetangga terdekat untuk data uji x menggunakan persamaan (2).
3. Hitung nilai keanggotaan $u(x, y_i)$ menggunakan persamaan (1) untuk setiap i , dimana $1 \leq i \leq C$.
4. Ambil nilai terbesar: $v = \arg \max_i (u(x, y_i))$ untuk semua $1 \leq i \leq C$, C adalah jumlah kelas.
5. Berikan label kelas v ke data uji x yaitu y_i .

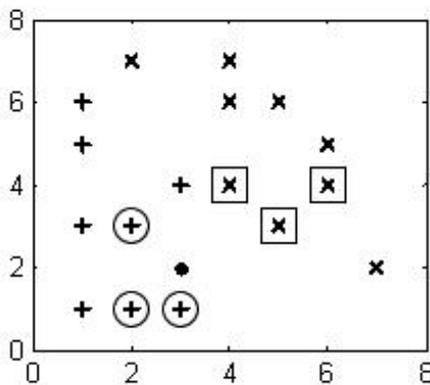
2.3. Fuzzy K-Nearest Neighbor in every Class

Metode Fuzzy K-Nearest Neighbor in every Class (FK-NNC) menggunakan sejumlah K tetangga terdekat pada setiap kelas dari sebuah data uji, bukan K tetangga terdekat seperti pada K-NN dan FK-NN. Gambar 1 memberikan gambaran K tetangga terdekat dari setiap kelas pada sebuah data uji pada metode FK-NNC. Pada gambar tersebut, untuk $K=3$, tiga tetangga terdekat

dikelas + dan tiga tetangga dikelas x yang ditemukan oleh FK-NNC [5].



a. Tanda dot hitam (solid) adalah data uji



b. Tiga tetangga dikelas + dan tiga tetangga dikelas x

Gambar 1. Konsep K tetangga terdekat dari setiap kelas, untuk K = 3

FK-NNC memberikan nilai keanggotaan data pada setiap kelas dalam interval 0 sampai 1 [5]. Jumlah nilai keanggotaan sebuah data pada semua kelas sama dengan 1, seperti pada persamaan (3).

$$\sum_{j=1}^C u_{ij} = 1, \quad 0 \leq u_{ij} \leq 1 \quad (3)$$

Dimana u_{ij} adalah nilai keanggotaan data uji x_i ke kelas ke- j .

Setiap data uji x_i , harus dicarikan K tetangga terdekat pada setiap kelas menggunakan formula (2), sehingga untuk 2 kelas akan ada $2 \times K$ tetangga yang didapat, untuk 3 kelas akan ada $3 \times K$ tetangga yang didapat, begitu seterusnya. Sehingga jika ada C kelas maka ada $C \times K$ tetangga terdekat yang harus dipilih [5]. Selanjutnya jarak data uji x_i ke semua K tetangga dari setiap kelas ke- j dijumlahkan, formula yang digunakan:

$$S_{ij} = \sum_{r=1}^K d(x_i, x_r)^{\frac{-2}{m-1}} \quad (4)$$

Nilai d sebagai akumulasi jarak data uji x_i ke K tetangga dalam kelas ke- j dilakukan sebanyak C kelas. Nilai m disini merupakan pangkat bobot (*weight exponent*) seperti pada FK-NN, nilai $m > 1$.

Selanjutnya, akumulasi jarak data uji x_i ke setiap kelas digabungkan, disimbolkan D , formula yang digunakan:

$$D_i = \sum_{j=1}^C (S_{ij}) \quad (5)$$

Untuk mendapatkan nilai keanggotaan data uji x_i pada setiap kelas ke- j (ada C kelas), menggunakan formula:

$$u_{ij} = \frac{S_{ij}}{D_i} \quad (6)$$

Untuk menentukan kelas hasil prediksi data uji x_i , dipilih kelas dengan nilai keanggotaan terbesar dari data x_i . Formula yang digunakan:

$$y' = \arg \max_{j=1}^C (u_{ij}) \quad (7)$$

Algoritma prediksi pada FK-NNC untuk sebuah data uji, disajikan sebagai berikut:

1. Cari K tetangga terdekat pada setiap kelas, menggunakan formula (2)
2. Hitung S sebagai akumulasi jarak K tetangga pada setiap kelas, menggunakan formula (4)
3. Hitung D sebagai akumulasi semua jarak dari $C \times K$ tetangga, menggunakan formula (5)
4. Hitung u sebagai nilai keanggotaan data pada setiap kelas, menggunakan formula (6)
5. Pilih nilai keanggotaan terbesar menggunakan formula (7), kelas dengan nilai keanggotaan terbesar menjadi kelas hasil prediksi untuk data uji tersebut.

Dalam makalah ini, modifikasi dilakukan pada cara perhitungan kuantitas jarak untuk mengukur ketidakmiripan, kuantitas jarak dihitung dalam dimensi tinggi \mathfrak{R} menggunakan pemetaan fungsi kernel.

3. Metode Penelitian

Penelitian yang dilakukan oleh Wu dan Zhou [9] membuktikan bahwa penggunaan fungsi kernel untuk memetakan dimensi data yang lama ke dimensi data yang baru terbukti dapat meningkatkan akurasi prediksi data uji, maka dalam penelitian ini juga digunakan fungsi kernel untuk memetakan data latih dari dimensi lama ke dimensi baru yang relatif lebih tinggi untuk perhitungan jarak antara data uji dengan data latih. Penggunaan fungsi kernel dalam perhitungan jarak tersebut diterapkan pada metode Fuzzy K-Nearest Neighbor in every Class (FK-NNC) [5]

Masalah yang harus diselesaikan dalam kasus nyata biasanya tidak linear, sehingga sangat cocok jika menggunakan fungsi kernel yang memetakan data dari dimensi yang lama ke dimensi yang baru (\mathfrak{R}) pada proses pelatihan atau prediksinya. Tetapi penggunaan jarak Euclidean dalam persamaan (2) tidak cukup kompleks untuk menghitung jarak dua data dalam dimensi yang baru (\mathfrak{R}) [9]. Hal ini disebabkan tidak dapat dilakukan perhitungan $W(x)$ secara langsung, tetapi harus menggunakan pasangan dua data antara $W(x)$ dengan $W(x_r)$. Persamaan (2) dapat dituliskan

kembali secara universal untuk menghitung jarak data uji ke data latih menjadi persamaan (8).

$$d = \|x - x_r\| \quad (8)$$

Dalam penelitian ini digunakan fungsi kernel Gaussian RBF untuk menghitung jarak (ketidakmiripan) pada dua data dalam dimensi yang lebih tinggi (\mathfrak{R}) pada persamaan (8), seperti yang yang diusulkan Wu dan Zhou [9].

Simbol W menyatakan pemetaan non-linear dari dimensi yang lama (X) ke dimensi yang baru (\mathfrak{R}).

$$X = (x_1, \dots, x_M) \rightarrow W(X) = (W(X_1), \dots, W(X_N)) \quad (9)$$

Maka persamaan (8) harus dipetakan juga ke fitur dimensi tinggi \mathfrak{R} :

$$d = \|x - x_r\| \rightarrow d = \|W(x) - W(x_r)\| \quad (10)$$

Jika \mathfrak{R} adalah dimensi yang sangat tinggi maka tidak mungkin dapat menghitung $d = \|W(x) - W(x_r)\|$. Maka untuk menyelesaikan keterbatasan tersebut digunakan metode kernel untuk menghitung jarak tersebut.

Kernel $K(x_i, x_j)$ menghitung dot-product dalam fitur dimensi tinggi \mathfrak{R} [9][4], seperti pada persamaan (11)

$$K(x_i, x_j) = \langle W(x_i), W(x_j) \rangle \quad (11)$$

Sehingga perhitungan product skalar dalam dimensi awal ditransformasi kedalam dimensi tinggi dengan pemetaan non-linear menggunakan fungsi kernel.

$$\langle x_i, x_j \rangle \rightarrow \langle W(x_i), W(x_j) \rangle = K(x_i, x_j)$$

Kernel harus menggunakan kernel Mercer [4], dari beberapa macam fungsi kernel, kernel Gaussian RBF merupakan salah satu kernel yang memerlukan perhitungan jarak dalam menghitung hasil pemetaannya, maka dalam penelitian ini digunakan kernel Gaussian RBF untuk menghitung jarak dalam dimensi tinggi, seperti yang juga digunakan oleh Wu dan Zhou [9][4].

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (12)$$

Suku kanan pada persamaan (4) bisa dituliskan menjadi:

$$\frac{1}{d(x_i, x_r)^{\frac{2}{m-1}}}$$

Dan untuk $m=2$, maka dapat dihitung jarak pada persamaan (10) menjadi [9]:

$$\begin{aligned} \|W(x) - W(x_r)\|^2 &= \\ &= \langle (W(x) - W(x_r)), (W(x) - W(x_r)) \rangle \quad (13) \\ &= K(x, x) + K(x_r, x_r) - 2K(x, x_r) \end{aligned}$$

Nilai $K(x, x)$ dalam pemetaan kernel Gaussian RBF selalu bernilai 1. Jadi jika hasil pemetaan persamaan (12)

dimasukkan ke persamaan (13) maka persamaan (13) berubah menjadi persamaan (14).

$$\|W(x) - W(x_r)\|^2 = 2 - 2K(x, x_r) \quad (14)$$

Persamaan (14) hanya berlaku untuk kernel Gaussian RBF untuk melakukan perhitungan jarak dalam dimensi tinggi \mathfrak{R} dalam KB-FK-NNC.

Untuk mendapatkan nilai keanggotaan data uji x_i pada setiap kelas ke- j (ada C kelas) dalam dimensi tinggi \mathfrak{R} didapat dari gabungan persamaan (4), (5) dan (6) yang sudah dipetakan dalam dimensi tinggi \mathfrak{R} oleh persamaan (14). Setelah dimasukkan dan disederhanakan, didapatkan persamaan (15).

$$u_{ij}(W(x)) = \frac{\sum_{r=1}^K \frac{1}{1 - K(x, x_r)}}{\sum_{q=1}^C \sum_{r=1}^K \frac{1}{1 - K(x, x_{qr})}} \quad (15)$$

Dimana K adalah jumlah K tetangga terdekat dari setiap kelas j pada data uji x_i , C adalah jumlah kelas. Sehingga ada $C \times K$ tetangga terdekat yang dilibatkan dalam perhitungan. Persamaan (15) berlaku untuk $m=2$ dalam FK-NNC pada dimensi tinggi \mathfrak{R} .

Algoritma prediksi Kernel-Based Fuzzy K-Nearest Neighbor in every Class (KB-FK-NNC) untuk sebuah data uji x disajikan sebagai berikut:

1. Gabungkan data uji x kedalam data latih X .
2. Lakukan pemetaan X dari dimensi lama ke dimensi baru: $\mathfrak{R} : X \rightarrow W(X)$ menggunakan persamaan (12)
3. Tentukan K , dengan syarat $1 \leq K \leq N$
4. Hitung jarak dari $W(x)$ ke $W(x_i)$ menggunakan metode kernel (dalam hal ini adalah matrik dimensi tinggi hasil dari fungsi kernel) dalam persamaan (14)
5. Pilih K tetangga dari setiap kelas, sehingga didapat $C \times K$ tetangga terdekat.
6. Hitung nilai keanggotaan data uji x ke setiap kelas menggunakan persamaan (15)
7. Berikan label kelas j dari nilai keanggotaan u_{ij} pada data uji x .

4. Pengujian dan Analisis

Metode klasifikasi Kernel-Based Fuzzy K-Nearest Neighbor in every Class (KB-FK-NNC) diuji pada 4 dataset publik yang diunduh dari website [7] yaitu: Iris (150 record, 4 fitur), Vertebral Column (310 record, 6 fitur), Wine (178 record, 13 fitur), dan Glass (214 record, 9 fitur). Sistem pengujian menggunakan model split 5-fold, dimana 80% digunakan sebagai data latih dan 20% digunakan sebagai data uji. Pada penelitian ini dilakukan pengujian pada kasus 2 kelas, maka untuk data set yang berisi lebih dari satu kelas akan dilakukan penggabungan beberapa kelas hingga menjadi dua kelas secara keseluruhan pada data set tersebut.

Pengujian perilaku kinerja secara lokal pada KB-FK-NNC dilakukan dengan memberikan nilai K dari 1 sampai 15, dengan pilihan nilai σ yang digunakan adalah: 1.2, 2, 2.6, 3, 3.5, 4. Hasilnya disajikan pada tabel 1 untuk data set Iris.

Tabel 1. Hasil prediksi KB-FK-NNC untuk data set Iris (dalam %)

K	σ					
	1.2	2	2.6	3	3.5	4
1	94.67	94.67	94.67	94.67	94.67	94
2	94	94	94	94	94	94
3	94.67	94.67	94.67	94.67	94.67	94
4	94.67	94.67	94.67	94.67	94.67	94
5	94.67	94.67	94	94.67	94.67	94
6	94.67	94.67	94.67	94.67	94.67	94
7	94.67	94.67	94.67	94.67	94.67	94
8	94.67	94.67	94.67	94.67	94.67	94
9	94	94.67	94.67	94.67	94	94
10	94	94	94.67	94.67	94	94
11	94	94	94.67	94.67	94	94
12	93.33	94	94.67	94.67	94	94
13	93.33	94	94.67	94.67	94	94
14	93.33	94	94.67	94.67	94	94
15	93.33	94	94.67	94.67	94	94

Dari data hasil prediksi seperti yang disajikan pada tabel 1, dapat diamati bahwa akurasi yang diberikan untuk nilai K dari 1 sampai 15 dengan pilihan σ yang digunakan, didapatkan rata-rata semuanya adalah 94.33% dengan simpangan baku adalah 0.39. Artinya disini, akurasi yang didapat cukup stabil untuk semua pilihan K yang digunakan dan tidak sensitif terhadap pilihan σ yang digunakan.

Tabel 2. Hasil prediksi KB-FK-NNC untuk data set Vertebral Column (dalam %)

K	σ					
	1.2	2	2.6	3	3.5	4
1	80.97	80.65	80.65	78.39	79.68	80.97
2	78.71	80.65	79.68	79.03	78.39	79.35
3	80	81.29	80	77.74	78.39	79.68
4	80.65	81.29	80.97	77.42	78.39	80
5	81.29	81.94	81.61	78.39	80.32	80.32
6	81.94	81.61	81.29	79.35	80.65	80
7	81.94	82.26	81.61	79.68	80.32	80.32
8	81.94	82.58	82.26	79.68	81.61	80.65
9	82.9	82.9	82.9	80	81.94	80.97
10	83.55	83.23	82.58	80.65	81.61	81.29
11	83.87	83.23	82.26	80.65	81.61	81.29
12	83.55	83.23	82.58	80.32	80.97	81.29
13	83.55	82.58	82.26	80.65	80.97	81.94
14	83.55	82.9	82.58	80.97	80.97	82.26
15	82.9	83.23	82.58	80.97	80.65	81.61

Pengujian perilaku kinerja secara lokal juga diujikan pada data set Vertebral Column, Wine, dan Glass, seperti disajikan pada tabel 2, tabel 3, dan tabel 4. Hasil akurasi rata-rata semua variasi K dan σ , masing-masing berturut-turut 81.14%, 96.85%, dan 93.05%. Sedangkan simpangan baku masing-masing berturut-turut adalah 1.47, 0.90, dan 0.97. Nilai yang didapat untuk semua data set tersebut masing dapat disimpulkan stabil, dan tidak bisa diputuskan pilihan nilai K yang terbaik.

Tabel 3. Hasil prediksi KB-FK-NNC untuk data set Wine (dalam %)

K	σ					
	1.2	2	2.6	3	3.5	4
1	96.06	96.06	94.94	94.94	94.35	94.41
2	96.06	96.06	95.51	94.94	95.48	95.52
3	95.49	96.06	96.62	95.49	95.49	96.1
4	96.63	96.06	96.62	96.06	95.49	96.1
5	97.19	96.63	96.62	96.06	96.06	96.65
6	97.75	96.63	96.62	96.06	96.62	96.65

7	97.75	96.63	97.75	97.19	97.19	96.65
8	97.75	96.63	97.75	97.19	97.19	97.21
9	97.75	96.06	97.75	97.19	97.19	97.21
10	97.75	97.19	97.75	96.62	97.19	97.21
11	97.75	97.19	97.75	97.19	97.19	97.21
12	97.75	97.19	97.75	97.19	97.19	97.76
13	97.75	97.19	97.75	97.19	97.19	97.76
14	97.75	98.32	97.75	97.76	97.19	97.76
15	97.75	98.32	97.75	97.76	97.19	97.76

Tabel 4. Hasil prediksi KB-FK-NNC untuk data set Glass (dalam %)

K	σ					
	1.2	2	2.6	3	3.5	4
1	94.39	94.4	92.99	93.45	93.48	93.93
2	94.87	93.48	93.93	93.45	94.87	94.4
3	94.87	93.94	94.39	92.98	95.33	94.87
4	94.4	93.47	93.93	92.98	94.87	94.87
5	94.4	93.47	93.93	92.98	93.93	94.4
6	93.48	93.47	93.93	92.98	93.93	93.94
7	93.02	93.47	93.45	92.51	92.53	93.48
8	93.02	93.47	93.45	92.51	92.53	92.55
9	93.02	93	92.51	92.51	92.53	92.09
10	92.54	93	92.05	92.99	92.53	92.09
11	92.54	92.53	92.05	92.99	92.53	92.09
12	92.54	92.53	92.05	92.53	92.53	91.63
13	92.54	92.53	92.05	92.06	92.06	91.63
14	92.54	92.05	91.57	92.06	92.06	91.63
15	92.08	92.05	91.1	92.06	92.53	91.15

Tabel 5. Pebandingan akurasi kinerja prediksi

Data Set	K-NN	FK-NN	FK-NNC	KB-FK-NNC
Iris	95.60%	95.33%	95.20%	93.38%
Vert. Col.	78.32%	80.00%	80.77%	81.40%
Wine	95.51%	93.22%	96.74%	96.74%
Glass	90.94%	91.63%	94.49%	93.34%

Pada sesi berbeda, pengujian data set yang sama juga dilakukan dengan membandingkan KB-FK-NNC pada metode yang lain yaitu: K-NN, FK-NN, dan FK-NNC. Hasilnya disajikan pada tabel 5. Dari tabel 2, dapat diamati bahwa akurasi prediksi yang didapat oleh KB-FK-NNC tidak selalu lebih baik daripada metode yang lain. Untuk data set Vertebral Column, nilai akurasi KB-FK-NNC secara signifikan lebih baik daripada K-NN yaitu lebih tinggi 3.08%, dan sedikit lebih baik daripada FK-NN dan FK-NNC dengan selisih 0.63%. Untuk data set Wine, KB-FK-NNC memberikan akurasi yang sama persis dengan FK-NNC yaitu 96.74%, sedangkan untuk data set Iris dan Glass masih kalah nilai akurasinya dibandingkan dengan metode sebelumnya.

Secara umum, akurasi prediksi yang diberikan oleh KB-FK-NNC tidak jauh berbeda dengan metode-metode sebelumnya, tetapi dalam KB-FK-NNC diperlukan penggunaan fungsi kernel, yang secara komputasi berharga mahal. Hal ini karena fungsi kernel melakukan transformasi data latih dan data uji dari dimensi lama ke dimensi baru yang relatif lebih tinggi. Jumlah dimensi yang lebih tinggi inilah yang diproses oleh KB-FK-NNC, tetapi memberikan kinerja akurasi yang tidak jauh berbeda dari metode sebelumnya.

5. Kesimpulan dan Saran

Dari penelitian yang sudah dilakukan, dapat disimpulkan sebagai berikut:

1. Fungsi kernel terbukti dapat digunakan dalam metode klasifikasi KB-FK-NNC dengan hasil akurasi yang stabil pada sejumlah pilihan nilai K dan σ yang digunakan. Hal ini memberikan pandangan bahwa nilai K yang σ yang dipilih tidak berpengaruh signifikan pada penurunan atau peningkatan akurasi.
2. Penggunaan kernel dalam KB-FK-NNC pada beberapa situasi dapat membantu meningkatkan nilai akurasi prediksi, tetapi tidak dapat memberikan peningkatan yang signifikan dari metode-metode sebelumnya yang tidak menggunakan kernel.

Saran-saran yang data diberikan untuk penelitian berikutnya sebagai berikut:

1. Penggunaan fungsi kernel yang lain masih perlu dilakukan penelitian untuk dibandingkan dengan fungsi kernel Gaussian RBF yang digunakan dalam penelitian ini.
2. Dalam penelitian ini, KB-FK-NNC baru melakukan prediksi pada data set dengan 2 kelas, sehingga diperlukan pengujian lebih lanjut untuk kasus prediksi kelas lebih dari 2.

Daftar Pustaka

- [1] Chen, HL., Yang, B., Wang, G., Liu, J., Vu, X., Wang, SJ., Liu, DY., 2011. *A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method*, Knowledge-Based Systems, Vol. 24, Issue 8, pp. 1348–1359
- [2] Keller, JM., Gray, MR., Givens, JA. 1985, *A Fuzzy K-Nearest Neighbor Algorithm*, IEEE Transaction on Systems, Man, and Cybernetics. Vol 15. No 4:580-585
- [3] Li, D., Deogun, JS., Wang, K. 2007, *Gene Function Classification Using Fuzzy K-Nearest Neighbor Approach*, IEEE International Conference on Granular Computing, IEEE Computer Society.
- [4] Muller, KR., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B., 2001, *An Introduction to Kernel-Based Learning Algorithms*, IEEE Transaction on Neural Networks, Vol. 12, No. 2, pp.181-202.
- [5] Prasetyo, E., 2012, *Fuzzy K-Nearest Neighbor in Every Class Untuk Klasifikasi Data*, Seminar Nasional Teknik Informatika (SANTIKA 2012), Teknik Informatika-Fakultas Teknologi Industri Universitas Pembangunan Nasional “Veteran” Jawa Timur, pp 57-60
- [6] Tan, P., Steinbach, M., Kumar, V., 2006, *Introduction to Data Mining*, Pearson Education: Boston San Fransisco New York
- [7] *UCI Machine Learning Repository*, 20 Mei 2012, <http://archive.ics.uci.edu/ml/datasets.html>
- [8] Weinberger, KQ., Blitzer, J., dan Saul, LK., 2005, *Distance Metric Learning for Large Margin Nearest Neighbor Classification*, Department of Computer and Information Science, University of Pennsylvania Levine Hall, 3330 Walnut Street, Philadelphia, PA 19104
- [9] Wu, XH., Zhou, JJ., 2005, *Kernel-based Fuzzy K-nearest-neighbor Algorithm*, Proceedings of the 2005

International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)

Biodata Penulis

Harunur Rosyid, memperoleh gelar Sarjana Teknik (ST.), Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia, lulus tahun 2000. Tahun 2012 memperoleh gelar Magister Komputer (M.Kom) dari Program Pasca Sarjana Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Saat ini sebagai Staf Pengajar program Sarjana Teknik Informatika Universitas Muhammadiyah Gresik.

Eko Prasetyo, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Gresik, lulus tahun 2005. Tahun 2011 memperoleh gelar Magister Komputer (M.Kom) dari Program Pasca Sarjana Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Saat ini sebagai Staf Pengajar program Sarjana Teknik Informatika UBHARA Surabaya.

Soffiana Agustin, memperoleh gelar Sarjana Komputer (S. Kom), Program Studi Teknik Informatika Jurusan Teknik Informatika STMIK AKAKOM Yogyakarta, lulus tahun 2002. Tahun 2010 memperoleh gelar Magister Komputer (M.Kom) dari Program Pasca Sarjana Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Saat ini sebagai Staf Pengajar program Sarjana Teknik Informatika Universitas Muhammadiyah Gresik.