ISSN : 2302-3805

WEIGHT K-SUPPORT VECTOR NEAREST NEIGHBOR

Eko Prasetyo [1] , Rifki Fahrial Zainal [2] , Harunur Rosyid [3]

[1) , 2)] Teknik Informatika Universitas Bhayangkara Surabaya
Jl. A. Yani 114, Surabaya, 60231
[3)] Teknik Informatika Universitas Muhammadiyah Gresik
Jl. Sumatra 101 GKB, Gresik, 61121
email : eko1979@yahoo.com[1)], rifkifz@gmail.com[2)], harun.ac@gmail.com[3)]

Abstrak

Algoritma klasifikasi berbasis K-Nearest Neighbor (K-NN) mempunyai banyak variasi, seperti Template Reduction K-NN, Support Vector K-NN, dan K-Support Vector NN. Semuanya berusaha untuk memperbaiki kinerja K-NN, baik dari sisi akurasi prediksi,maupun waktu prediksi. Tetapi untuk hal tersebut memang harus dibayar dengan adanya waktu yang harus dialokasikan untuk pelatihan, sedangkan K-NN klasik tidak melakukan pelatihan sama sekali. Dalam makalah ini dipaparkan hasil penelitian berupa metode Weight K-Support Vector Nearest Neighbor (WK-SVNN) yang berusaha untuk meningkatkan akurasi prediksi dan mengurangi waktu pelatihan dan prediksi yang digunakan. Hasilnya, WK-SVNN membutuhkan waktu prediksi yang paling lama, tetapi berhasil meningkatkan akurasi prediksinya dibanding semua metode pembanding.

Kata kunci :
K-Nearest Neighbor, Support Vector, bobot, skor, derajat signifikansi.

1. Pendahuluan

Algoritma klasifikasi yang sangat tua dan menjadi pilihan utama dalam penelitian dibidang data mining adalah K-Nearest Neighbor (K-NN). Kederhanaan algoritmanyalah yang menjadi kunci kepopulerannya [2]. Algoritmanya bekerja dengan cara membandingkan ketidakmiripan antara data uji yang baru dengan sejumlah data latih, kemudian diambil K tetangga yang paling mirip untuk dilakukan voting mayoritas sebagai kelas hasil prediksi [6]. Kelebihan lainnya adalah mampu memodelkan fungsi tujuan yang kompleks dengan sejumlah perkiraan kompleks lokal, selain itu informasi yang tersimpan data data latih tidak pernah hilang.
Meskipun mempunyai kelebihan, K-NN juga mempunyai kekurangan, yaitu harus menyimpan keseluruhan data latih untuk digunakan pada saat prediksi. Hal inilah yang menyebabkan proses prediksi menjadi sangat lama. Sedangkan untuk menyimpan seluruh data latih saja, harus dialokasikan di memori. Disamping itu, keberadaan noise dalam data juga dapat mendistorsi hasil prediksi. Ada sejumlah penelitian yang sudah dipublikasikan, semuanya bertujuan untuk memperbaiki kinerja K-NN, seperti disajikan dibagian dua.
Berdasarkan hasil pengamatan penulis pada penelitian-penelitian sebelumnya, penulis melakukan pengembangan skema kerja metode K-Support Vector Nearest Neighbor (K-SVNN) untuk mendapatkan akurasi prediksi yang lebih baik dengan melakukan proses lebih lanjut pada support vector yang didapat sebelum digunakan pada saat proses prediksi.
Makalah ini dibagi menjadi 5 bagian. Bagian 1 menyajikan pendahuluan yang melatarbelakangi penulis melakukan penelitian. Bagian 2 menyajikan penelitian-penelitian terkait yang menjadi dasar bagi penulis untuk melanjutkan penelitiannya. Bagian 3 menyajikan kerangka kerja metode yang diteliti oleh penulis. Bagian 4 menyajikan pengujian dan analisis yang dilakukan untuk mengukur kinerja metode. Dan bagian 5 menyajikan simpulan dari hasil penelitian dan saran untuk penelitian berikutnya.

2. Tinjauan Pustaka dan Penelitian Terkait

Sejumlah penelitian yang bertujuan untuk perbaikan metode K-NN telah banyak dipublikasikan, baik yang bertujuan untuk meningkatkan akurasi kinerja maupun mengurangi kebutuhan memori untuk menyimpan data latih.
Prasetyo [4] membuat skema K-Support Vector Nearest Neighbor (K-SVNN) untuk melakukan reduksi data latih, kemudian support vector yang didapat tersebut digunakan untuk melakukan prediksi menggunakan K-NN klasik. Salah satu peluang pengembangan lebih lanjut yang dapat dilakukan diantaranya adalah penggunakan skema prediksi, hal ini menjadi penting karena adanya penurunan jumlah data latih (berupa Support Vector) yang berkontribusi pada saat prediksi sehingga dimungkinkan adanya penurunan akurasi kinerja prediksi jika dibandingkan dengan K-NN klasik.

Angiulli [1] mengusulkan Fast Condensed Nearest Neighbor (FCNN) Rule dengan menambahkan properti dalam menghitung bagian dari data latih yang berpengaruh pada garis keputusan prediksi. Hasil pengujian menunjukkan bahwa akurasi prediksi yang

ISSN : 2302-3805

diberikan secara umum lebih baik dari metode berbasis condensed yang lain.

Srisawat et al. [5] mengusulkan skema SV-KNN dengan melakukan reduksi jumlah data latih untuk mempercepat kinerja K-NN, ada 3 langkah utama yang dilakukan yaitu menggunakan SVM untuk mendapatkan support vector (sebagian dari data latih) yang mempunyai pengaruh signifikan pada fungsi tujuan, menggunakan K-Means Clustering untuk mempartisi dan memampatkan support vector menjadi sejumlah centroid dengan bobot centroid berdasarkan prosentasi komposisi data dari masing-masing kelas, kemudian menggunakan prototype tersebut sebagai template K-NN yang baru untuk proses prediksi dengan bobot setiap data (centroid) pada setiap kelas. Meskipun akurasinya meningkat, tetapi harus dibayar dengan proses pelatihan yang lama yaitu penggunaan SVM dan K-Means dalam mendapatkan prototype terbobot tersebut.

Fayed dan Atiya [3] mengusulkan *condensing approach* dengan nama Template Reduction K-Nearest Neighbor (TR-KNN) yang berusaha mengurangi data latih yang tidak berpengaruh siginifikan pada saat proses prediksi. Data latih tersebut adalah data latih yang posisinya tidak berada diposisi dekat *hyperplane* fungsi tujuan. Data ini tidak punya pengaruh apa-apa pada saat proses prediksi sehingga harus dibuang dari template utama. Selanjutnya data latih yang tersisa merupakan data latih yang digunakan sebagai template baru K-NN untuk proses prediksi. TR-KNN berhasil mengurangi data latih secara signifikan dibandingkan dengan metode sebelumnya, tetapi akurasi yang didapatkan secara umum tidak lebih tinggi dari lain [5].

3. Kerangka Kerja Metode

Kerangka kerja yang ditambahkan pada K-SVNN dalam makalah ini dapat menjadi bagian proses pelatihan dalam K-SVNN. Pada K-SVNN [4] proses pelatihannya berisi algoritma untuk menghitung skor dan derajat signifikansi kemudian dilakukan seleksi data latih yang nilai derajat signifikansinya lebih dari atau sama dengan threshold yang ditetapkan. Dalam makalah ini, dilakukan penambahan dalam proses pelatihan, yaitu dengan menambahkan bobot selama pelatihan, kemudian diberikan skema proses prediksi yang baru dengan mewarisi algoritma prediksi K-NN klasik. Perbaikan metode ini diberi nama Weight K-Support Vector Nearest Neighbor (WK-SVNN).

3.1 Properti-properti WK-SVNN

Properti-properti yang dimiliki oleh WK-SVNN dalam penelitian ini masih menggunakan property K-SVNN dalam penelitian sebelumnya [4], ditambah properti bobot dalam penelitian ini. Properti-properti tersebut dijelaskan sebagai berikut:

1. Skor (*score*)
   Properti skor untuk setiap data latih ada 2 nilai: nilai kiri (Left Value / LV) dan nilai kanan (Right Value / RV), nilai yang kiri untuk kelas yang sama, sedangkan nilai yang kanan untuk kelas yang berbeda [4]. Jumlah LV dan RV dari semua data latih sama dengan N×K, seperti dinyatakan oleh persamaan (1).
   $$\sum_{i=1}^{N} LV_i + \sum_{i=1}^{N} RV_i = N \times K \quad (1)$$
2. Derajat siginifikan (*significant degree*)
   Properti relevansi / derajat signifikansi adalah nilai yang menyatakan tingkat signifikansi (relevansi) data latih tersebut pada fungsi tujuan (daerah hyperplane) [4]. Nilainya dalam rentang 0 sampai 1 [0,1], semakin tinggi nilainya maka relevansinya untuk menjadi *support vector* (data latih yang digunakan pada saat prediksi) juga semakin tinggi.
   Nilai derajat signifikansi (*Significant Degree* / SD) didapatkan dengan membagi LV terhadap RV atau RV terhadap LV sesuai syarat yang terpenuhi seperti pada persamaan (2).
   $$SD_i = \begin{cases} 0 & , SV_i = RV_i = 0 \\ \dfrac{SV_i}{RV_i} & , SV_i < RV_i \\ \dfrac{RV_i}{SV_i} & , SV_i > RV_i \\ 1 & , SV_i = RV_i \end{cases} \quad (2)$$
3. Bobot (*weight*)
   Bobot merupakan properti baru yang ditambahkan ke kerangka kerja WK-SVNN dalam makalah ini. Bobot menyatakan derajat pengaruh support vector pada setiap kelas. Nilainya dalam jangkauan 0 sampai 1 [0,1]. Nilai 0 menyatakan tidak ada pengaruh sama sekali, sedangkan nilai 1 menyatakan mempunyai pengaruh penuh pada kelas tersebut. Karena dalam kerangka WK-SVNN ada 2 kelas, maka untuk bobot pada setiap support vector juga ada 2: W_1 dan W_2. W_1 menyatakan bobot ke kelas 1, W_2 menyatakan bobot ke kelas 2. Untuk mendapatkan W maka harus dihitung koefisien bobot (α) untuk setiap jarak data uji ke K tetangga terdekat.
   Untuk menghitung koefisien bobot (α) setiap tetangga dari K tetangga digunakan persamaan (3).
   $$\sim = \frac{1}{1 + e^{\log 10(jarak)}} \quad (3)$$
   Untuk menghitung bobot W sebuah support vector pada kelas 1 digunakan persamaan (4), sedangkan pada kelas 2 digunakan persamaan (5).
   $$W_{j1} = W_{j1} + \sim \quad (4)$$
   $$W_{j2} = W_{j2} + \sim \quad (5)$$
   Untuk bobot akhir (U) yang akan diberikan pada data uji yang diprediksi kelasnya, didapatkan dengan mengakumulasi perkalian koefisien bobot (α) dengan bobot W dari kelas 1 dan kelas 2 dari K tetangga terdekat yang didapat. Untuk mengakumulasi bobot di kelas 1 digunakan

ISSN : 2302-3805

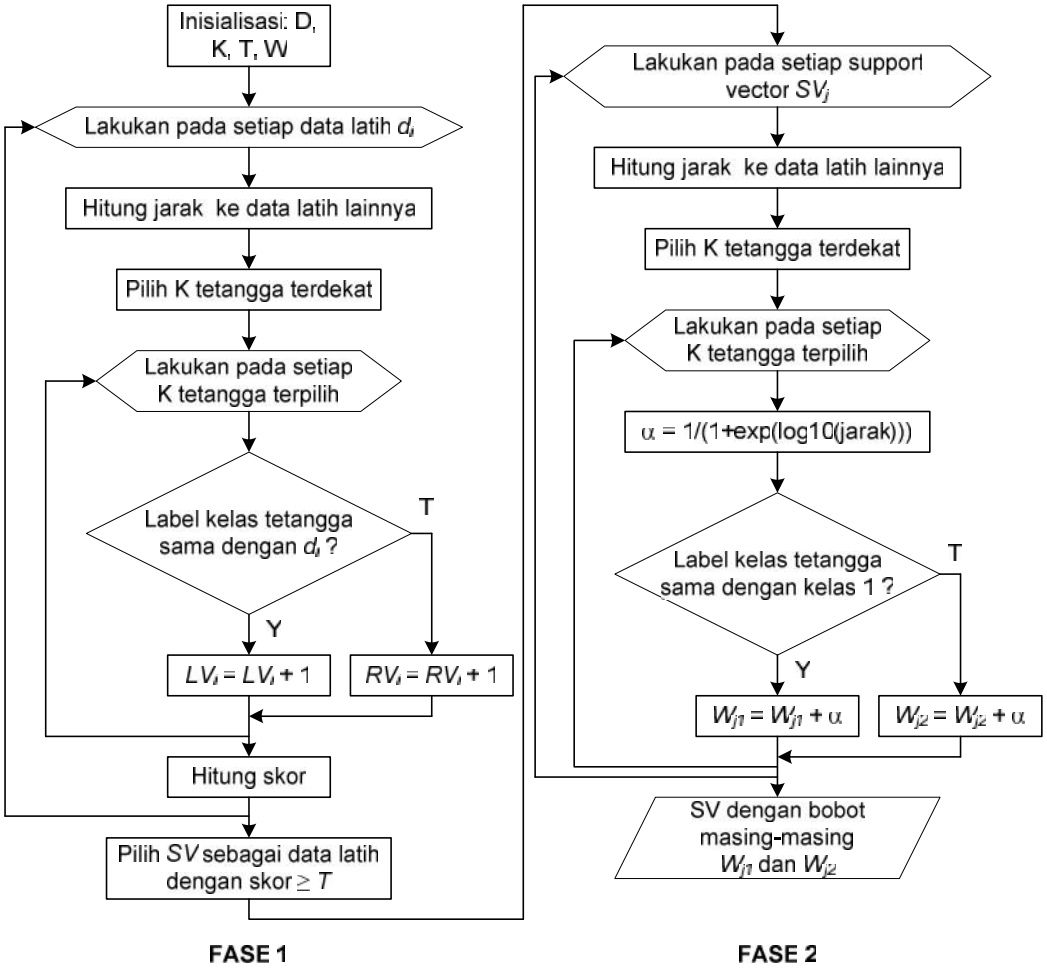
persamaan (6), untuk kelas 2 digunakan persamaan (7).

$$Ud_{i1} = Ud_{i1} + W_{j1} \times \sim_j \quad (6)$$

$$Ud_{i2} = Ud_{i2} + W_{j2} \times \sim_j \quad (7)$$

3.2 Pelatihan model WK-SVNN

Skema kerangka kerja pelatihan K-SVNN yang baru diberikan pada gambar 1. Pada gambar tersebut, terbagi menjadi 2 fasa: fasa 1 dan fasa 2. Fasa 1 merupakan proses pelatihan dalam K-SVNN yang diusulkan Prasetyo [4], sedangkan fasa 2 merupakan tambahan proses yang diberikan dalam penelitian ini untuk mendapatkan bobot setiap support vector pada setiap kelas.



Gambar 1. Kerangka kerja pelatihan K-SVNN terbobot

Algoritma pelatihan WK-SVNN dapat dijelaskan sebagai berikut:

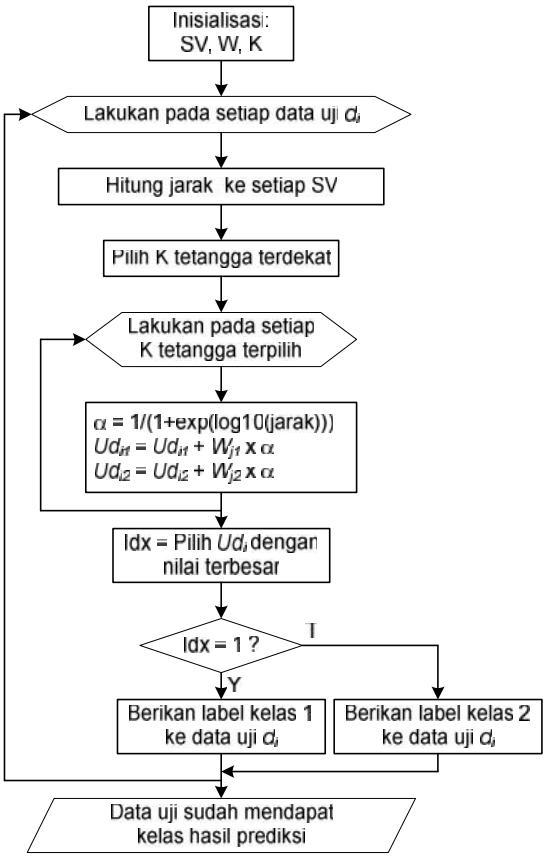
Fasa 1

1. Inisialisasi: D adalah set data latih, K adalah jumlah tetangga terdekat, T adalah threshold SD yang dipilih, LV dan RV untuk semua data latih = 0.
2. Untuk setiap data latih d_iČD, lakukan langkah 3 sampai 5
3. Hitung ketidakmiripan (jarak) dari d_i ke data latih yang lain.
4. Pilih *dt* sebagai K data latih tetangga terdekat (tidak termasuk d_i).
5. Untuk setiap data latih dalam *dt*, jika label kelas sama dengan d_i, maka tambahkan nilai 1 pada LV_i, jika tidak sama maka tambahkan nilai 1 pada RV_i.
6. Untuk setiap data latih d_i, hitung SD_i menggunakan persamaan (2)
7. Pilih data latih dengan SD ≥ T, simpan dalam memori (variabel) sebagai support vector untuk prediksi.

Fasa 2

1. Untuk setiap support vector SV_i Č *SV*, lakukan langkah 2 sampai 6
2. Hitung ketidakmiripan (jarak) dari SV_i ke data latih yang lain.
3. Pilih *dt* sebagai K data latih tetangga terdekat (tidak termasuk SV_i).

4. Untuk setiap data latih dalam *dt*, lakukan langkah 5 sampai 6
5. Hitung α menggunakan persamaan (3).
6. Jika label kelasnya sama dengan 1, gunakan persamaan (4) untuk mengakumulasikan ke bobot kelas 1, jika tidak, gunakan persamaan (5) untuk mengakumulasikan ke bobot kelas 2.



Gambar 2. Kerangka kerja prediksi K-SVNN terbobot

3.3 Prediksi menggunakan model WK-SVNN

Skema kerangka kerja prediksi WK-SVNN dalam makalah ini disajikan pada gambar 2. Algoritma prediksinya dapat dijelaskan sebagai berikut:

1. Lakukan inisialisasi: SV adalah support vector, W adalah bobot untuk setiap support vector, dan K adalah jumlah tetangga terdekat yang akan digunakan.
2. Untuk setiap data uji d_i, lakukan langkah 3 sampai 9
3. Hitung jarak dari data uji d_i ke setiap support vector
4. Pilih K tetangga dengan jarak terdekat.
5. Pada setiap K tetangga terpilih, lakukan langkah 6 sampai 7
6. Hitung α menggunakan persamaan (3).
7. Hitung Ud_{i1} dan Ud_{i2} menggunakan persamaan (6) dan (7).
8. Dapatkan Idx dengan memilih Ud_i terbesar
9. Jika Idx=1, berikan label kelas 1 ke data uji d_i, jika tidak berikan label kelas 2

4. Pengujian dan Analisis

Pengujian metode WK-SVNN dilakukan pada 4 data set. utama yang diambil dari data set publik [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 5 fold, dimana 80% digunakan sebagai data latih dan 20% digunakan sebagai data uji. K-SVNN yang diuji dalam penelitian ini masih bekerja hanya pada dua kelas saja, sehingga harus dilakukan penggabungan beberapa kelas berbeda menjadi satu kelas pada data set yang komposisi kelasnya lebih dari dua.

Pengujian WK-SVNN dibandingkan dengan 4 metode sebelumnya. Hasil pengujian dengan membandingkan kinerja WK-SVNN untuk perbandingan akurasi prediksi disajikan pada tabel 1, sedangkan untuk perbandingan waktu yang dibutuhkan selama proses pelatihan dan prediksi diberikan pada tabel 2 dan 3

Tabel 1. Perbandingan akurasi prediksi

Data set	K-NN	TR-KNN	SV-KNN	K-SVNN	WK-SVNN
Iris	96.0%	57.3%	90.7%	94.7%	96.0%
Vert. Col.	84.8%	68.1%	85.8%	84.2%	87.1%
Wine	73.1%	57.1%	74.3%	76.0%	77.1%
Glass	91.9%	85.2%	89.0%	90.0%	91.0%

Dari hasil pengujian yang disajikan pada tabel 1, dapat diamati bahwa akurasi prediksi yang diberikan oleh WK-SVNN selalu diatas metode lain sebelumnya pada 2 dari 4 data set yaitu Vertebral Column, dan Wine. Bahkan dibandingkan dengan K-SVNN, WK-SVNN mempunyai akurasi yang lebih tinggi. Khusus untuk data set Glass, WK-SVNN mempunyai akurasi kinerja prediksi dibawah K-NN sekitar 0.9%, tetapi untuk ukuran nilai diatas 90% sudah termasuk bagus.

Tabel 2. Perbandingan waktu pelatihan (mili detik)

Data set	K-NN	TR-KNN	SV-KNN	K-SVNN	WK-SVNN
Iris	0	10.27	787.48	23.37	26.83
Vert. Col.	0	18.31	7,011.28	43.76	58.73
Wine	0	9.58	1,463.04	29.11	38.20
Glass	0	12.60	2,283.12	31.96	33.15

Dari hasil pengujian yang disajikan pada tabel 2, dapat diamati bahwa WK-SVNN memerlukan waktu pelatihan yang lebih lama dari pada K-SVNN, TR-KNN dan K-NN, hal ini karena WK-SVNN sebenarnya adalah K-SVNN yang mendapat tambahan langkah kerja pada proses pelatihannya untuk mendapatkan bobot pada setiap support vectornya.

Tabel 3. Perbandingan waktu prediksi (mili detik)

Data set	K-NN	TR-KNN	SV-KNN	K-SVNN	WK-SVNN
Iris	2.71	2.25	3.36	2.27	3.75
Vert. Col.	6.64	4.67	7.85	4.87	8.59
Wine	3.57	2.91	4.09	3.08	4.96
Glass	4.22	3.06	5.66	3.18	5.86

Dari hasil pengujian yang disajikan pada tabel 3, dapat diamati bahwa WK-SVNN membutuhkan waktu prediksi yang lebih lama dibandingkan dengan 4 metode sebelumnya, hal ini terjadi karena WK-SVNN harus memproses nilai-nilai bobot untuk setiap support vector

ISSN : 2302-3805

yang tidak dilakukan oleh metode lainnya, SV-KNN juga memproses bobot, tetapi representasi bobot yang digunakan berupa nilai integer mewakili jumlah kontribusi data dalam setiap kelas, sedangkan bobot dalam WK-SVNN merepresentasikan berat setiap support vector pada dua kelas dengan jangkauan nilai 0 sampai 1 [0,1].
Secara umum, dapat dikatakan bahwa metode WK-SVNN mempunyai kinerja akurasi yang lebih baik dibandingkan metode lain sebelumnya, tetapi memerlukan waktu yang lebih lama baik untuk proses pelatihan maupun prediksi.

5. Kesimpulan dan Saran

Dari pembahasan pada bagian sebelumnya, dapat disimpulkan sebagai berikut:

1. Metode WK-SVNN bisa menjadi alternatif yang lebih baik untuk digunakan sebagai mesin klasifikasi, meskipun akurasi prediksinya hanya sedikit lebih baik dibanding metode yang lain.
2. Algoritma reduksi data set menjadi support vector dan pembentukan bobot yang sederhana bisa menjadi kelebihan tersendiri bagi WK-SVNN ketika dibandingkan dengan metode klasifikasi yang lain.

Saran-saran yang dapat diberikan dari penelitian yang sudah dilakukan sebagai berikut:

1. WK-SVNN masih bekerja hanya untuk dua kelas, perlu penelitian lebih lanjut untuk dapat menangani kasus klasifikasi dengan jumlah kelas lebih dari dua.
2. Karena harus menghitung bobot, hal ini mengakibatkan proses prediksi menjadi lambat. Bahkan dibandingkan dengan K-NN klasik yang dikenal mempunyai waktu prediksi yang lama, ternyata WK-SVNN masih lebih lambat pada proses prediksinya.

Daftar Pustaka

[1] Angiulli, F., 2007. *Fast Nearest Neighbor Condensation for Large Data Sets Classification*, IEEE Transaction Knowledge and Data Engineering, Vol. 19, No. 11, pp. 1450-1464

[2] Dhanabal, S., Chandramathi, S., 2011. *A Review of various k-Nearest Neighbor Query Processing Techniques*, International Journal of Computer Applications (0975 – 8887), Vol. 31, No.7, pp.14-22.

[3] Fayed, H.A., Atiya, A.F., 2009. *A Novel Template Reduction Approach for the K-Nearest Neghbor Method*. IEEE Transaction on Neural Network, 20(5), pp.890-896.

[4] Prasetyo, E. 2012. K-Support Vector Nearest Neighbor untuk Klasifikasi Berbasis K-NN, Jurusan Sistem Informasi ITS, *Seminar Nasional Sistem Informasi Indonesia*. Surabaya, 3 Nopember 2012, ITS Press: Surabaya

[5] Srisawat, A., Phienthrakul, T., Kijsirikul, B., 2006. SV-KNNC: An Algorithm for Improving the Efficiency of K-Nearest Neighbor. In: Qiang Yang, Geoffrey I. Webb. *The 09th Pacific Rim International Conference on Artificial Intelligence* (PRICAI-2006). Guilin, China, 7-11 August 2006. Springer-Verlag Berlin Heidelberg.

[6] Tan, P., Steinbach, M., Kumar, V., 2006. *Introduction to Data Mining*, 1st Ed, Pearson Education: Boston San Fransisco New York

[7] *UCI Machine Learning Repository* , 20 Mei 2012, http://archive.ics.uci.edu/ml/datasets.html

Biodata Penulis

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.

Rifki Fahrial Zainal, memperoleh gelar Sarjana Teknik (ST.), Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Surabaya, lulus tahun 2005. Tahun 2008 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.

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.

ISSN : 2302-3805