

REDUKSI DATA LATIH DENGAN K-SVNN SEBAGAI PEMROSESAN AWAL PADA ANN BACK-PROPAGATION UNTUK PENGURANGAN WAKTU PELATIHAN

Eko Prasetyo

Program Studi Teknik Informatika, Fakultas Teknik
Universitas Bhayangkara Surabaya
Email: eko@ubhara.ac.id

ABSTRAK

Sistem klasifikasi yang akurat membutuhkan masukan data yang baik juga. Agar didapatkan kualitas data masukan yang baik, maka pemrosesan awal dapat dilakukan. Pemrosesan awal data masukan yang dapat dilakukan pada masalah dimensi adalah reduksi dimensi dan reduksi data masukan. Data masukan tersebut yang digunakan sebagai data latih pada algoritma pelatihan *Artificial Neural Network* (ANN). Jumlah data masukan yang besar dapat menjadi masalah karena dapat menambah waktu eksekusi pelatihan. Pada sistem klasifikasi dengan algoritma ANN, semakin banyak jumlah data latih, maka iterasi pelatihan pada ANN semakin banyak dan waktu eksekusi juga semakin lama. Hal ini disebabkan pada tiap iterasi setiap data latih akan diproses tepat satu kali. Penelitian ini bertujuan untuk melakukan uji coba penggunaan *K-Support Vector Nearest Neighbor* (K-SVNN) sebagai pemrosesan awal data untuk mereduksi data latih yang kurang berpengaruh pada garis keputusan ANN. ANN yang digunakan dalam penelitian ini adalah ANN *Back-propagation*. Hasil yang dicapai adalah berkurangnya jumlah data latih yang dieksekusi di tiap iterasi tetapi tetap mempertahankan kualitas hasil pelatihan. Pengurangan jumlah data latih yang dieksekusi diukur dengan waktu yang digunakan selama proses pelatihan, sedangkan kualitas hasil pelatihan diukur dengan nilai akurasi yang didapat ketika dilakukan prediksi. Hasil yang dicapai cukup baik, waktu pelatihan berkurang sebesar 15% hingga 80%, sedangkan perbedaan akurasi prediksi ada penurunan, penurunan sebesar 0% hingga 4.76%.

Kata kunci: pemrosesan awal, reduksi data latih, waktu latih, K-SVNN, ANN *back-propagation*.

ABSTRACT

Accurate classification system requires good data input as well. In order to have a good quality of input data, preprocessing can be done. Preprocessing of input data that can be performed on dimensions problem is dimension reduction and reduction of input data. That input data is used as training data on training algorithms of Artificial Neural Network (ANN). A large amount of input data that can be a problem because it can add the execution time of training. In the classification system with ANN algorithm, the more amount of training data, then the iterating on ANN training will be more and more, and the execution time will be longer. This is due in each iteration of each training data will be processed exactly once. This study aimed to try the use of K-Support Vector Nearest Neighbor (K-SVNN) as preprocessing of the data to reduce training data that are less impact on the decision line of ANN. ANN used in this research is ANN Back-propagation. While the result is a reduced number of training data which is executed in each iteration while still keep the quality of training results. Reduction of the amount of executed training data is measured by the time spent during training process, while the quality of training results are measured by an accuracy value obtained when predictions. Results achieved quite well, training time is reduced by 15% to 80%, while there was a decrease difference prediction accuracy, decreased by 0% to 4.76%.

Keywords: *pre-processing, data reduction training, training time, K-SVNN, ANN back-propagation.*

1. PENDAHULUAN

1.1 Latar Belakang

Pemrosesan awal (*pre-processing*) data merupakan salah satu proses tahapan dalam sistem seperti data mining, visi komputer, pengolahan citra, dan sistem-sistem yang lain. Pemrosesan awal menjadi salah satu proses yang penting dilakukan sebelum masuk ke pemrosesan utama dalam sistem. Hal ini dikaitkan dengan kualitas data masukan yang diolah. Semakin baik kualitas data masukan maka hasil sistem secara keseluruhan juga semakin baik. Jika kualitas data masukan jelek maka meskipun sistem memberikan hasil, tetapi hasil tersebut bisa jadi terdistorsi dari hasil yang seharusnya didapatkan. Tahapan ini tentu beralasan untuk dilakukan mengingat kondisi

awal data biasanya masih ada masalah, seperti: data yang tidak lengkap, data tidak konsisten, data dengan fitur yang banyak sekali, data mengandung *noise*, termasuk jumlah data yang sangat besar juga menjadi masalah penting untuk diselesaikan diawal. Untuk sistem dengan metode-metode yang sensitif terhadap nilai numerik, sensitif terhadap *noise*, sensitif terhadap jumlah fitur dan data yang besar tentu pemrosesan awal penting dilakukan.

Dalam pemrosesan awal ada sejumlah strategi dan teknik berbeda yang saling berhubungan dengan sangat kompleks [1]. Pekerjaan-pekerjaan dalam pemrosesan awal yang dapat dilakukan, misalnya: agregasi, sampling, reduksi dimensi, pemilihan subset fitur, pembangkitan fitur, diskretisasi dan binerisasi, transformasi variabel, normalisasi, perlakuan data yang salah, dan sebagainya.

Jumlah data yang besar tentu juga menjadi masalah dalam proses utama karena dapat menambah durasi proses. Pada sistem klasifikasi, data latih yang digunakan untuk pembelajaran/ pelatihan model dalam memberikan hasil prediksi yang benar juga perlu menjadi perhatian. Pada sistem klasifikasi seperti *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM) tentu memerlukan pemrosesan awal untuk memberikan data masukan yang lebih berkualitas. Adanya *noise* dapat memberikan pengaruh kinerja [1]. Semakin banyak jumlah data latih, maka bisa jadi iterasi pelatihan yang diperlukan pada ANN juga semakin banyak, dimana pada tiap iterasi setiap data latih akan dieksekusi untuk dilatihkan. Pada prinsipnya, pelatihan pada ANN dan SVM adalah proses pencarian garis keputusan (*decision boundary*) dalam memberikan batas klasifikasi dua kelas atau lebih data latih. Secara kontekstual, seharusnya hanya data-data yang dekat dengan garis keputusan saja yang punya pengaruh dalam klasifikasi baik pelatihan maupun pengujian. Penelitian yang dilakukan Prasetyo [2] membuat skema *K-Support Vector Nearest Neighbor* (K-SVNN) untuk melakukan reduksi data latih pada data yang digunakan dalam klasifikasi dengan metode K-NN. Hasilnya, data latih yang kurang berpengaruh pada garis keputusan berhasil direduksi dengan tidak mengurangi akurasi kinerja.

Berdasarkan pemaparan diatas, dalam makalah ini dipaparkan hasil penelitian penggunaan K-SVNN yang digunakan sebagai pemrosesan awal data latih sebelum digunakan untuk pelatihan pada metode ANN. Pemrosesan awal yang dilakukan adalah reduksi data latih dengan membuang data latih yang tidak berpengaruh pada garis keputusan dengan metode K-SVNN.

Pemaparan 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 melakukan penelitian. Bagian 3 menyajikan metodologi penelitian. Bagian 4 menyajikan hasil penelitian dan pembahasan yang dilakukan pada hasil reduksi dan perbandingan kinerja. Dan bagian 5 menyajikan simpulan dari hasil penelitian dan saran untuk penelitian berikutnya.

1.2 Penelitian Sebelumnya

Penelitian yang dilakukan Prasetyo [2] adalah melakukan reduksi data latih dengan melakukan pencarian data yang punya pengaruh pada garis keputusan. Ciri data latih yang punya pengaruh pada garis keputusan dalam penelitian ini adalah data latih yang mempunyai tetangga dari kelas yang sama dan berbeda dari data tersebut. Semakin seimbang jumlah tetangga terdekat dari dua kelas tersebut maka semakin kuat pengaruh data tersebut pada garis keputusan. Dengan semakin kuatnya data tersebut, maka potensi menjadi *Support Vector* juga semakin tinggi. Diakhir proses akan tersisa data latih yang punya pengaruh pada garis keputusan setelah lolos seleksi dengan nilai *threshold* tertentu. Skema seperti tersebut disebut dengan *K-Support Vector Nearest Neighbor* (K-SVNN). *Support Vector* tersebut kemudian digunakan dalam proses prediksi K-NN.

Penelitian-penelitian yang banyak dilakukan yang berkaitan dengan pemrosesan awal data [3] [4]. Untuk pengurangan jumlah data, yang dapat dilakukan diantaranya adalah agregasi, reduksi dimensi, kompresi data, dan sejumlah teknik lain. Sisi lain dalam pengurangan jumlah data yang masih dapat dilakukan adalah pengurangan data latih yang kurang berpengaruh pada garis keputusan. Maka dalam penelitian ini fokus yang dilakukan adalah pengurangan jumlah data latih sebelum dilakukan proses pelatihan pada metode *Artificial Neural Network* (ANN).

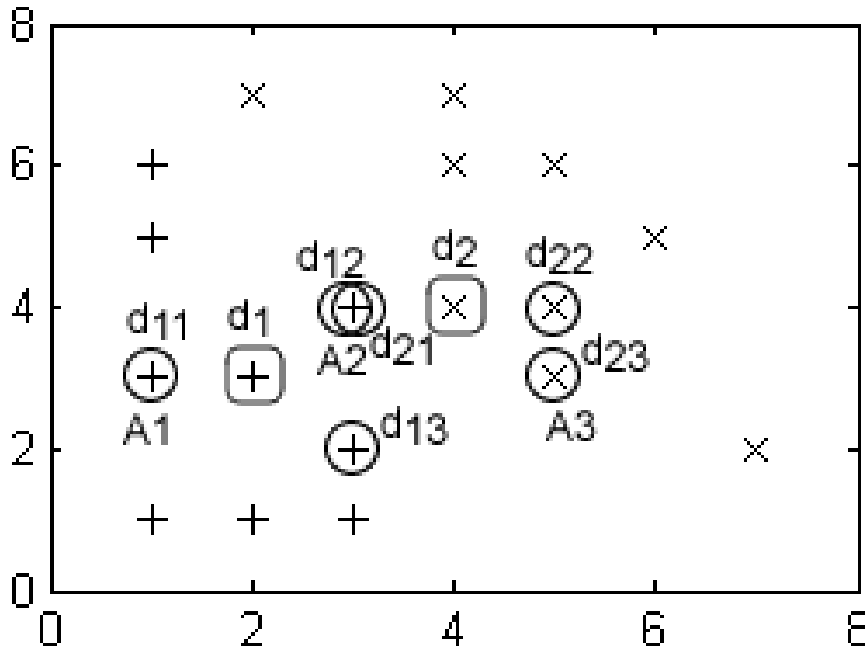
1.3 K-Support Vector Nearest Neighbor

K-Support Vector Nearest Neighbor (K-SVNN) berusaha mengurangi data latih berdasarkan properti skor (*score*) dan properti relevansi / derajat signifikansi (*significant degree*) pada setiap data latih berdasarkan prinsip K tetangga terdekat [2]. Setiap data latih mempunyai kedua properti tersebut. 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. Jumlah LV dan RV dari semua data latih sama dengan $N \times K$, seperti dinyatakan oleh persamaan (1).

$$\sum_{i=1}^N LV_i + \sum_{i=1}^N RV_i = N \times K \quad (1)$$

Properti relevansi / derajat signifikansi adalah nilai yang menyatakan tingkat signifikansi (relevansi) data latih tersebut pada fungsi tujuan (daerah *hyperplane*). 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. Dalam penelitian ini digunakan batas ambang T (*threshold*) > 0, yang artinya sekecil apapun nilai relevansi akan tetap digunakan sebagai support vector. Nilai 0 pada derajat signifikansi sebuah data latih berarti data latih tersebut harus dibuang (tidak digunakan sebagai *Support Vector*). Nilai derajat signifikansi (*Significant Degree / SD*) didapatkan dengan membagi LV terhadap RV atau RV terhadap LV sesuai syarat yang terpenuhi seperti pada persamaan (2).

$$SD_i = \begin{cases} 0 & , SV_i = RV_i = 0 \\ \frac{SV_i}{RV_i} & , SV_i < RV_i \\ \frac{RV_i}{SV_i} & , SV_i > RV_i \\ 1 & , SV_i = RV_i \end{cases} \quad (2)$$



Gambar 1. K-SVNN Dengan K=3

Ilustrasi proses untuk mendapatkan properti skor disajikan pada gambar 1. Pada gambar tersebut, digunakan K=3. Data latih d1 mempunyai 3 tetangga terdekat d11, d12, d13, karena kelas dari 3 tetangga tersebut sama dengan d1 maka nilai LV untuk data tetangga d11, d12, dan d13 masing-masing bertambah 1. Sedangkan data latih d2 mempunyai 3 tetangga terdekat d21, d22, d23, untuk data tetangga d22 dan d23 kelasnya sama dengan d2 maka nilai LV pada data tetangga d22 dan d23 bertambah 1, sedangkan data tetangga d21 nilai RV bertambah 1. Dari data latih d1 dan d2 saja, maka untuk data A1 nilai LV=1 dan RV=0, untuk data A2 nilai LV=1 dan RV=1, sedangkan data A3 nilai LV=1 dan RV=0. Hal tersebut dilakukan pada semua data latih.

Properti lain yang juga menentukan kinerja K-SVNN baik pada saat pelatihan maupun pada saat prediksi adalah nilai K. Secara umum, nilai K yang disarankan untuk digunakan adalah K>1. Pengaruh besarnya nilai K yang digunakan pada saat pelatihan adalah semakin kecil nilai K maka semakin sedikit jumlah *Support Vector* (semakin besar reduksinya), semakin cepat waktu kerjanya saat prediksi, dan semakin kecil pula akurasi yang didapatkan, demikian pula sebaliknya. Tetapi nilai K yang besar tidak

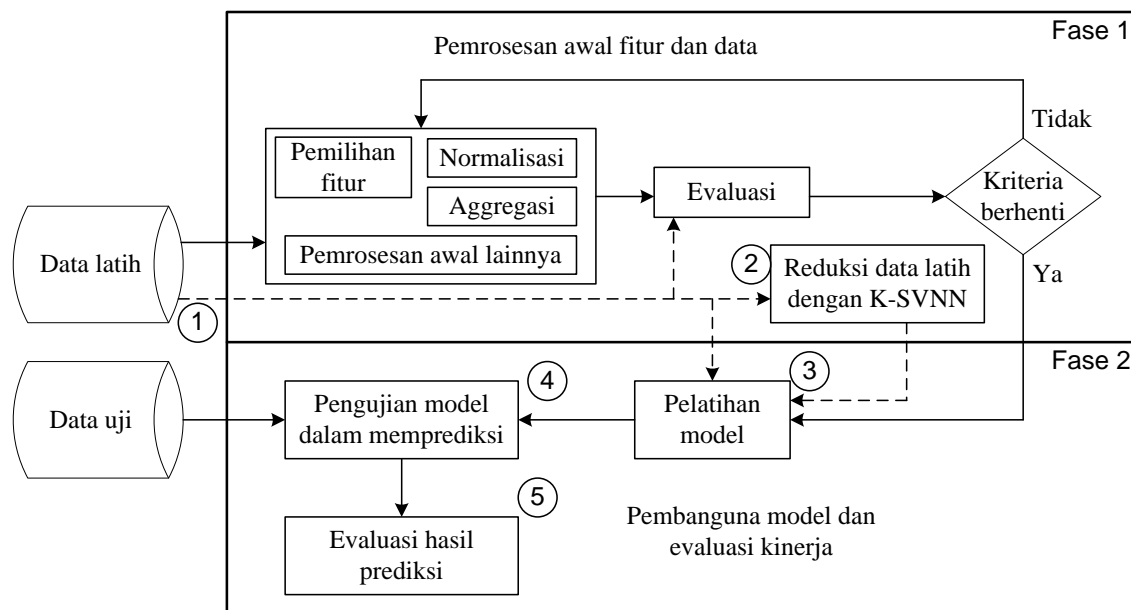
menjamin akurasi kinerja yang semakin baik. Algoritma pelatihan K-SVNN dapat dijelaskan sebagai berikut:

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 $d_i \in D$, 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 $SD \geq T$, simpan dalam memori (variabel) sebagai template untuk prediksi.

2. METODOLOGI PENELITIAN

Penelitian ini melakukan eksperimen penggunaan K-SVNN sebagai pemrosesan awal data latih sebelum digunakan sebagai masukan pada proses pelatihan ANN *Back-propagation*. Algoritma pelatihan *Back-propagation* mengharuskan untuk membaca semua data latih pada tiap iterasi pelatihan [5][6]. Misalnya jika jumlah data latih ada 100 dan iterasi pelatihan yang dilakukan ada 300, maka data latih tersebut harus dilakukan pembacaan sebanyak 30.000 kali. Jika data latih dapat dikurangi, misalnya 20%, maka data latih yang tersisa adalah 80 data, dengan asumsi jumlah iterasi sama, maka pembacaan data dilakukan 24.000 kali, dengan harapan kualitas model dalam melakukan klasifikasi sama atau tidak jauh berbeda.

Penelitian ini melakukan hal tersebut dimana dilakukan pengurangan data latih dengan K-SVNN, kemudian *Support Vector* yang didapat digunakan sebagai data masukan pada pelatihan metode ANN *Back-propagation*. Selanjutnya, model hasil pelatihan digunakan untuk melakukan prediksi data uji. Hasil prediksinya juga dibandingkan dengan model hasil pelatihan dengan masukan data latih tanpa melalui K-SVNN.



Gambar 2. Pemrosesan Awal Reduksi Data Latih Dengan K-SVNN

Kerangka kerja penggunaan K-SVNN sebagai pemrosesan awal digambarkan pada gambar 2. Penelitian penggunaan K-SVNN sebagai pemrosesan awal reduksi data ditunjukkan berturut-turut oleh nomor 1-2-3-4-5, dimulai dari data latih yang dilakukan reduksi dengan K-SVNN kemudian hasilnya dijadikan masukan dalam proses pelatihan model, model hasil pelatihan tersebut kemudian digunakan untuk melakukan pengujian kinerja prediksi dengan masukan data uji, hasil prediksi kemudian dilakukan evaluasi. Untuk pemrosesan awal yang lain tetap dilakukan seperti biasa, tetapi dalam penelitian ini, pada set data yang digunakan sebagai uji coba dilakukan normalisasi linear dengan mentransformasi nilai data

setiap fitur menjadi rentang 0 sampai 1. Sedangkan perbandingan dengan klasifikasi tanpa reduksi data ditunjukkan oleh nomor 1-3-4-5.

Metode klasifikasi yang dijadikan fokus dalam penelitian ini adalah ANN *Back-propagation*. Metode ini sangat luas penggunaannya karena kemampuannya dalam menangani data-data yang non linear. Hal tersebut ditangani dengan digunakannya layer tersembunyi. Neuron dalam hidden layer mendeteksi fitur-fitur tersembunyi, bobot dari neuron dalam hidden layer merepresentasikan fitur tersembunyi dalam vektor masukan. Fitur-fitur tersembunyi ini kemudian digunakan oleh layer keluaran dalam penentuan pola/kelas keluaran. Dengan satu hidden layer, dapat merepresentasikan sembarang fungsi kontinyu dari sinyal masukan, dan dengan dua hidden layer fungsi diskontinyupun dapat direpresentasikan [6]. Penggunaan ANN *Back-propagation* dalam implementasinya diperlukan untuk mengatur parameter-yang menentukan kinerja ANN, seperti: jumlah layer tersembunyi, jumlah neuron dalam layer tersembunyi, laju pembelajaran, fungsi aktivasi, target error, kriteria error, momentum, dan jumlah maksimal iterasi pelatihan. Dalam penelitian ini digunakan batasan parameter ANN *Back-propagation* sebagai berikut:

- 1 jumlah layer tersembunyi = 1
- 2 jumlah neuron dalam layer tersembunyi = 150
- 3 laju pembelajaran = 0.1
- 4 fungsi aktivasi = sigmoid biner
- 5 target error = 0.001
- 6 kriteria error = mse
- 7 momentum = 0.95
- 8 jumlah maksimal iterasi pelatihan = 1000

3. HASIL PENELITIAN DAN PEMBAHASAN

Pengujian dilakukan terhadap 6 set data publik yang diunduh dari UCI Machine Learning Repository [7] yaitu: Iris (150 record, 4 fitur), Vertebral Column (310 record, 6 fitur), Wine (178 record, 13 fitur), Glass (214 record, 9 fitur), Blood (748 record, 4 fitur), dan Ionosphere (351 record, 33 fitur). Sistem pengujian menggunakan *5-fold Cross Validation*, dimana 80% digunakan sebagai data latih dan 20% digunakan sebagai data uji. Kinerja diukur dari nilai rata-rata dari 5 sesi *Cross Validation*.

Pengujian dilakukan dengan mengukur kinerja antara 2 sistem: (1) klasifikasi tanpa pemrosesan awal K-SVNN; (2) klasifikasi dengan pemrosesan awal K-SVNN. Parameter yang diukur adalah waktu yang dibutuhkan untuk pelatihan, dan akurasi prediksi. Disajikan juga prosentase data yang dikeluarkan dari data latih. Waktu pelatihan untuk sistem yang kedua terdiri dari waktu yang digunakan untuk pemrosesan awal K-SVNN ditambah waktu pelatihan ketika menggunakan *Support Vector* yang dihasilkan dari K-SVNN. Untuk reduksi data, nilai K yang diujikan pada K-SVNN adalah K bernilai 7, 9, 11, 13.

Tabel 1. Hasil perbandingan kinerja pada set data Iris

| K | WAKTU LATIH | | | | REDUKSI WAKTU LATIH (%) | AKURASI (%) | | | DATA TER REDUKSI (%) |
|----|---------------------------|----------------------------|---------|---------|-------------------------|---------------|----------------|-------|----------------------|
| | TANPA REDUKSI (milidetik) | DENGAN REDUKSI (milidetik) | | | | TANPA REDUKSI | DENGAN REDUKSI | TURUN | |
| | | REDUKSI | LATIH | JUMLAH | | | | | |
| 7 | 1476.30 | 13.06 | 1092.90 | 1105.96 | 25.09 | 95.00 | 95.00 | 0.00 | 71.25 |
| 9 | 2260.34 | 8.74 | 1149.70 | 1158.44 | 42.64 | 92.00 | 92.00 | 0.00 | 66.00 |
| 11 | 2410.90 | 10.06 | 1141.63 | 1151.69 | 46.26 | 93.00 | 94.00 | -1.00 | 61.00 |
| 13 | 2351.22 | 12.74 | 1234.78 | 1247.52 | 39.69 | 95.00 | 93.00 | 2.00 | 54.25 |

Tabel 2. Hasil perbandingan kinerja pada set data Vertebral Column

| K | WAKTU LATIH | | | | REDUKSI WAKTU LATIH (%) | AKURASI (%) | | | DATA TER REDUKSI (%) |
|----|---------------------------|----------------------------|---------|---------|-------------------------|---------------|----------------|-------|----------------------|
| | TANPA REDUKSI (milidetik) | DENGAN REDUKSI (milidetik) | | | | TANPA REDUKSI | DENGAN REDUKSI | TURUN | |
| | | REDUKSI | LATIH | JUMLAH | | | | | |
| 7 | 10499.00 | 24.40 | 8375.30 | 8399.70 | 20.00 | 83.87 | 87.10 | -3.23 | 38.31 |
| 9 | 10569.06 | 31.86 | 8718.56 | 8750.42 | 17.23 | 84.84 | 83.87 | 0.97 | 36.29 |
| 11 | 10572.06 | 34.06 | 8784.98 | 8819.04 | 16.52 | 85.48 | 83.23 | 2.26 | 32.26 |
| 13 | 10547.68 | 36.81 | 8922.48 | 8959.29 | 15.00 | 87.10 | 85.81 | 1.29 | 29.84 |

Tabel 3. Hasil perbandingan kinerja pada set data Wine

| K | WAKTU LATIH | | | | REDUKSI WAKTU LATIH (%) | AKURASI (%) | | | DATA TER REDUKSI (%) |
|----|---------------------------------|----------------------------|---------|---------|----------------------------------|------------------|-------------------|-------|-------------------------------|
| | TANPA REDUKSI (milidetik) | DENGAN REDUKSI (milidetik) | | | | TANPA REDUKSI | DENGAN REDUKSI | TURUN | |
| | | REDUKSI | LATIH | JUMLAH | | | | | |
| 7 | 2116.50 | 37.40 | 2325.00 | 2362.40 | -11.62 | 94.29 | 94.29 | 0.00 | 74.83 |
| 9 | 1962.70 | 18.89 | 2435.08 | 2453.97 | -31.13 | 97.78 | 91.57 | 6.21 | 67.14 |
| 11 | 1539.42 | 18.17 | 1797.78 | 1815.95 | -19.81 | 96.05 | 96.06 | -0.02 | 61.66 |
| 13 | 1549.48 | 20.15 | 2131.52 | 2151.67 | -39.18 | 96.63 | 94.38 | 2.25 | 57.16 |

Tabel 4. Hasil perbandingan kinerja pada set data Glass

| K | WAKTU LATIH | | | | REDUKSI WAKTU LATIH (%) | AKURASI (%) | | | DATA TER REDUKSI (%) |
|----|---------------------------------|----------------------------|---------|---------|----------------------------------|------------------|-------------------|-------|-------------------------------|
| | TANPA REDUKSI (milidetik) | DENGAN REDUKSI (milidetik) | | | | TANPA REDUKSI | DENGAN REDUKSI | TURUN | |
| | | REDUKSI | LATIH | JUMLAH | | | | | |
| 7 | 7244.60 | 15.42 | 1416.30 | 1431.72 | 80.24 | 95.24 | 90.48 | 4.76 | 72.67 |
| 9 | 8828.02 | 18.52 | 2372.88 | 2391.40 | 73.43 | 92.99 | 91.58 | 1.41 | 68.33 |
| 11 | 7591.64 | 24.34 | 2918.60 | 2942.94 | 61.05 | 94.41 | 92.07 | 2.34 | 62.74 |
| 13 | 8223.36 | 22.11 | 2697.02 | 2719.13 | 67.56 | 92.53 | 93.01 | -0.49 | 57.01 |

Tabel 5. Hasil perbandingan kinerja pada set data Blood

| K | WAKTU LATIH | | | | REDUKSI WAKTU LATIH (%) | AKURASI (%) | | | DATA TER REDUKSI (%) |
|----|---------------------------------|----------------------------|----------|----------|----------------------------------|------------------|-------------------|-------|-------------------------------|
| | TANPA REDUKSI (milidetik) | DENGAN REDUKSI (milidetik) | | | | TANPA REDUKSI | DENGAN REDUKSI | TURUN | |
| | | REDUKSI | LATIH | JUMLAH | | | | | |
| 7 | 18979.70 | 93.19 | 12689.10 | 12782.29 | 32.65 | 78.52 | 82.55 | -4.03 | 34.56 |
| 9 | 19044.90 | 81.38 | 14256.00 | 14337.38 | 24.74 | 79.02 | 77.95 | 1.07 | 29.61 |
| 11 | 19497.18 | 92.12 | 16231.10 | 16323.22 | 16.15 | 79.14 | 78.33 | 0.80 | 24.56 |
| 13 | 19537.62 | 106.93 | 15102.76 | 15209.69 | 22.03 | 79.01 | 78.07 | 0.94 | 20.22 |

Hasil perbandingan kinerja ketika digunakan set data Iris disajikan pada tabel 1. Hasil tersebut menunjukkan bahwa waktu latihan yang dibutuhkan pada saat pelatihan ANN *Back-propagation* dengan pemrosesan awal K-SVNN lebih pendek daripada tanpa pemrosesan awal, pengurangan waktu yang berhasil didapatkan dari 25.09% hingga 46.26%, ini nilai yang cukup signifikan. Sedangkan akurasi kinerja hampir sama pada semua pilihan K yang digunakan, selisih 1% hingga 2%, dan ini bisa dikatakan hampir sama. Untuk data latihan yang berhasil direduksi mulai 54.25% hingga 71.25%, jumlah yang juga signifikan. Dari hasil yang didapat, bisa dinyatakan bahwa pada set data Iris, pemrosesan awal dengan K-SVNN dapat membantu dengan sangat baik.

Hasil perbandingan kinerja ketika digunakan set data Vertebral Column disajikan pada tabel 2. Hasil tersebut menunjukkan bahwa waktu latihan yang dibutuhkan pada saat pelatihan ANN *Back-propagation* dengan pemrosesan awal K-SVNN juga lebih pendek daripada tanpa pemrosesan awal seperti pada set data Iris, pengurangan waktu yang berhasil didapatkan dari 15% hingga 20%, ini nilai yang juga signifikan. Sedangkan akurasi kinerja mempunyai selisih yang lebih besar daripada set data Iris pada semua pilihan K yang digunakan, nilainya turun naik hingga 3%. Untuk data latihan yang berhasil direduksi mulai 29.84% hingga 38.31%, jumlah yang juga signifikan. Dari hasil yang didapat, bisa dinyatakan bahwa pada set data Vertebral Column, pemrosesan awal dengan K-SVNN dapat membantu dengan cukup baik.

Hasil perbandingan kinerja ketika digunakan set data Wine disajikan pada tabel 3. Hasil tersebut agak berbeda dari 2 set data sebelumnya. Pada set data Wine kinerja waktu pelatihan untuk klasifikasi dengan pemrosesan awal K-SVNN justru lebih panjang daripada tanpa pemrosesan awal. Waktu latihan secara keseluruhan justru naik minimal 11.62% (nilai negatif pada tabel menunjukkan bahwa nilai pada kolom jumlah lebih besar daripada kolom tanpa reduksi), meskipun sebenarnya waktu untuk K-SVNN juga jauh lebih pendek dari pada pelatihan ANN *Back-propagation* itu sendiri. Akurasi kinerja yang diberikan juga masih perlu diragukan karena nilai selisih dari 0% hingga 6.21% lebih rendah. Reduksi

data cukup besar yang diberikan menjadi tidak berguna ketika waktu latih keseluruhan menjadi lebih panjang dan akurasi yang malah turun. Jadi bisa dinyatakan bahwa pada set data Wine, pemrosesan awal dengan K-SVNN tidak dapat membantu dan malah merusak kinerja.

Hasil perbandingan kinerja ketika digunakan set data Glass disajikan pada tabel 4. Hasil tersebut menunjukkan bahwa waktu latih yang dibutuhkan pada saat pelatihan ANN *Back-propagation* dengan pemrosesan awal K-SVNN juga lebih pendek daripada tanpa pemrosesan awal seperti pada set data Iris dan Vertebral Column, pengurangan waktu yang berhasil didapatkan dari 61.05% hingga 80.24%, ini nilai yang sangat signifikan. Sedangkan akurasi kinerja lebih banyak turunnya hingga 4.76%. Jumlah data yang berhasil direduksi hingga 72.67%. Dari hasil yang didapat, bisa dinyatakan bahwa pada set data Glass, pemrosesan awal dengan K-SVNN dapat membantu dengan cukup baik.

Hasil perbandingan kinerja ketika digunakan set data Blood disajikan pada tabel 5. Hasil tersebut menunjukkan bahwa waktu latih yang dibutuhkan pada saat pelatihan ANN *Back-propagation* dengan pemrosesan awal K-SVNN juga lebih pendek daripada tanpa pemrosesan awal. Pengurangan waktu yang berhasil didapatkan dari 16.15% hingga 32.65%, nilai yang cukup signifikan. Sedangkan akurasi kinerja naik turun, tetapi turunnya hanya mencapai 1.07%. Jumlah data yang berhasil direduksi hingga 34.56%. Dari hasil yang didapat, bisa dinyatakan bahwa pada set data Glass, pemrosesan awal dengan K-SVNN dapat membantu dengan baik.

Dari hasil pengujian pada lima set data tersebut, dapat diringkas hasil bahwa penggunaan K-SVNN sebagai pemrosesan awal data latih untuk mereduksi data latih yang tidak signifikan dapat membantu mengurangi waktu pelatihan pada metode ANN *Back-propagation* dengan tetap mempertahankan akurasi kinerja prediksi model yang dihasilkan. Sedangkan data yang berhasil dikurangi juga cukup signifikan, meskipun untuk hasil tersebut masih dipengaruhi oleh penggunaan K pada K-SVNN itu sendiri.

4. KESIMPULAN

Dari penelitian yang dilakukan, maka dapat disimpulkan sebagai berikut:

- 1) K-SVNN dapat digunakan sebagai pemrosesan awal data untuk mereduksi data latih dengan tetap mempertahankan kualitas hasil pelatihan.
- 2) Penggunaan K-SVNN menunjukkan bahwa waktu pelatihan berkurang sebesar 15% hingga 80%, sedangkan perbedaan akurasi prediksi ada penurunan sebesar 0% hingga 4.76%.

Saran yang dapat penulis berikan dari penelitian ini sebagai petunjuk untuk penelitian berikutnya adalah sebagai berikut:

- 1) Penggunaan K pada K-SVNN masih menjadi pilihan yang penting untuk dipertimbangkan, karena dapat mempengaruhi jumlah data yang direduksi.
- 2) Pada set data tertentu, pemrosesan awal dengan K-SVNN tidak dapat bekerja dengan ini, hal ini perlu kajian lebih lanjut untuk mengetahui penyebabnya.

DAFTAR PUSTAKA

- [1] Tan, P.N., M. Steinbach, V. Kumar. 2006. *Introduction to Data Mining*, 1st Ed, Boston San Francisco New York: Pearson Education.
- [2] Prasetyo, E. 2012. K-Support Vector Nearest Neighbor Untuk Klasifikasi Berbasis K-NN, in *Proceeding of Seminar Nasional Sistem Informasi Indonesia*, Institut Teknologi Sepuluh Nopember, Surabaya.
- [3] Ladha, L., T. Deepa. 2011, *Feature Selection Methods And Algorithms*, International Journal on Computer Science and Engineering, Vol. 3 No. 5, May 2011
- [4] Beniwal, S., J. Arora. 2012. *Classification and Feature Selection Techniques in Data Mining*, International Journal of Engineering Research & Technology, Vol. 1, No. 6, August – 2012.
- [5] Theodoridis, S., K. Koutroubas. 2009. *Pattern Recognition – 4th edition*, Academic Press: Burlington, MA, USA.
- [6] Prasetyo, E. 2014. *Data Mining – Mengolah Data Menjadi Informasi Menggunakan Matlab*, Yogyakarta: Andi Offset

[7] UCI Machine Learning Repository , 1 Juni 2014, <http://archive.ics.uci.edu/ml/datasets.html>