JURNAL SAINS DAN SENI ITS Vol. 4, No.1, (2015) 2337-3520 (2301-928X Print)

Perbandingan *Reduced Support Vector Machine* dan *Smooth Support Vector Machine* untuk Klasifikasi *Large Data*

Epa Suryanto dan Santi Wulan Purnami
Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: santi_wp@statistika.its.ac.id

***Abstrak*—Klasifikasi merupakan pengelompokan objek ke dalam dua atau lebih kelompok yang didasarkan pada variabel yang diamati. *Support Vector Machine* merupakan metode berbasis machine learning yang sangat menjanjikan untuk dikembangkan karena memiliki performansi tinggi dan dapat diaplikasikan secara luas untuk klasifikasi dan estimasi. SVM memanfaatkan optimasi dengan *quadratic programming*, sehingga untuk data berdimensi tinggi dan berjumlah besar, SVM menjadi kurang efisien. Untuk mengatasi hal tersebut, dikembangkan *Smooth Support Vector Machine* (SSVM). Pada jumlah data yang besar SSVM juga tidak efisien kemudian dikembangkan *Reduced Support Vector Machine* (RSVM) yang melakukan klasifikasi dengan menggunakan sebagian karakteristik dari data yang dipilih secara random. Hasil penelitian ini menunjukkan pada jumlah data yang relatif kecil (kurang dari 1000) metode SSVM dan RSVM memberikan performansi yang sama, tetapi pada data yang relatif besar (lebih dari 1000) RSVM memberikan performansi yang lebih baik daripada SSVM.**

***Kata Kunci*—Klasifikasi, *Smooth Support Vector Machine, Reducedd Smooth Support Vector Machine*, Large Data**

I. PENDAHULUAN

KLASIFIKASI merupakan pengelompokan objek ke dalam dua atau lebih kelompok yang didasarkan pada variabel yang diamati. Pada umumnya metode klasifikasi yang digunakan adalah analisis diskriminan dan regresi logistik. Kedua metode tersebut memiliki kelemahan yaitu regresi logistik menghitung estimasi probabilitas untuk masuk pada kelas tertentu, sehingga kurang praktis digunakan [1]. Sedangkan untuk Analisis diskriminan terdapat beberapa syarat, yaitu variabel prediktor harus berskala rasio atau interval, matriks varian yang sama untuk setiap populasi dan data harus berdistribusi normal multivariat [2]. Karena metode tersebut memiliki kelemahan sehingga muncul banyak penelitian metode klasifikasi dengan pendekatan *computational programing*, misalnya, *Artificial Neural Network (ANN), Naive Bayes, Classification Adaptive Regression Tree* (*CART*) dan *Support Vector Machine (SVM)* [1].

Menurut Rachman(2011), Huang (2003) dan Byvatov (2003) *Support Vector Machine* memiliki tingkat akurasi yang lebih baik jika dibandingkan dengan metode regresi logistik, *ANN*, *Naive Bayes*, dan *CART* [3-5]. *Support Vector Machine* merupakan metode berbasis *machine learning* yang sangat menjanjikan untuk dikembangkan karena memiliki performansi tinggi dan dapat diaplikasikan secara luas untuk klasifikasi dan estimasi [6].

Menurut Lee dan Mangasarian, (2001) SVM memanfaatkan optimasi dengan *quadratic programming*, sehingga untuk data berdimensi tinggi dan data jumlah besar SVM menjadi kurang efisien. Oleh karena itu dikembangkan *smoothing technique* yang menggantikan *plus function* SVM dengan integral dari fungsi sigmoid *neural network* yang selanjutnya dikenal dengan *Smooth Support Vector Machine* (*SSVM*). Apabila dibandingkan dengan SSVM, SVM memiliki waktu *running* yang lebih lama dan akurasi yang lebih kecil daripada SSVM [7].

Pada jumlah data yang besar, SSVM juga tidak efisien karena memecahkan masalah optimasi tanpa kendala dan melibatkan fungsi kernel pada bidang pemisah non linier yang membutuhkan memori sangat besar dan pada umumnya komputer mengalami *out of memory* bahkan sebelum dimulai proses pencarian solusi. Pada jumlah data lebih besar dari 8000, SSVM tidak mampu memberikan solusi klasifikasi [8]. Pada tahun yang sama Lee dan Mangasarian memperkenalkan konsep *Reduced Support Vector Machine (RSVM)*. RSVM merupakan model kernel matriks yang diturunkan dari *General Suport Vector Machine* (*GSVM*) dan *Smooth Support Vector Machine* (*SSVM*). Konsep dasar RSVM adalah melakukan klasifikasi dengan menggunakan sebagian karakteristik dari data yang dipilih secara random.

Penelitian ini akan membandingkan performansi dari metode RSVM dengan metode SSVM dengan berbagai jumlah data baik untuk kasus linier maupun nonlinier. Perfomansi tersebut dinyatakan dengan tingkat akurasi dan waktu *running* yang diperlukan metode tersebut. Data yang digunakan merupakan data simulasi dan data riil. Pada penelitian ini juga digunakan algoritma *Newton-Armijo* dengan *stratified* 10-*fold cross validation* untuk menyelesaikan masalah optimasi SSVM dan RSVM. Sedangkan kernel yang digunakan adalah fungsi Kernel Gaussian dengan metode seleksi parameter *Uniform Design (UD)*.

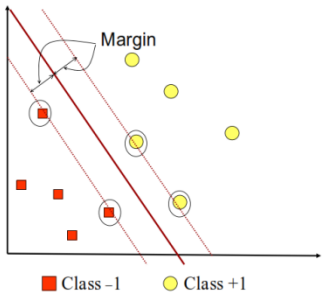
II. LANDASAN TEORI

A. Support Vector Machine

SVM pertama kali diperkenalkan oleh Vapnik tahun 1992. SVM berusaha menemukan fungsi pemisah (*hyperplane*) yang optimal sebagai pemisah dua buah kelas pada *input space*.

JURNAL SAINS DAN SENI ITS Vol. 4, No.1, (2015) 2337-3520 (2301-928X Print)

Gambar 1 menunjukkan sebuah data set yang memiliki dua kelas yaitu kelas {-1} dan {1}.



Gambar 1. Hyperplane Optimum

Hyperplane terbaik merupakan hiperplane yang memiliki margin maksimal yang diperoleh dari alternatif garis pemisah (*discriminant boundaries*). *Margin* adalah jarak antara *hyperplane* dengan titik terdekat dari masing-masing kelas, sedangkan titik terdekat tersebut disebut *support vector.* [9].

Masalah klasifikasi *m* titik dalam ruang dimensi *n* yang dinotasikan R^n , digambarkan dengan matriks **A** yang berukuran *m* x *n*, dengan anggotanya dinotasikan $\mathbf{A}_i^{\mathrm{T}}$ terhadap kelas {+1} dan {-1} didefinisikan pada diagonal matriks **D** berukuran m x m dengan 1 dan -1 pada setiap diagonalnya. Algoritma untuk SVM linier adalah sebagai berikut

$$\min_{(\boldsymbol{w},\gamma,\boldsymbol{y})\in R^{n+1+m}} v\mathbf{e}^{\mathrm{T}}\mathbf{y}+\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \tag{1}$$

dengan kendala $\mathbf{D}(\mathbf{Aw}-\mathbf{e}\gamma)+\mathbf{y}\geq\mathbf{e}$

$\mathbf{y}\geq\mathbf{0}$

dengan v merupakan parameter dalam SVM yang bernilai positif

- **y** vektor variabel *slack* berukuran *m*x1 yang mengukur kesalahan klasifikasi dan bernilai non negatif
- **e** vektor kolom berukuran *m* dan bernilai 1
- **w** vektor normal berukuran *n* x 1
- γ nilai bias yang menentukan lokasi relatif *hyperplane* terhadap kelas asli

persamaan fungsi kendala diatas membandingkan setiap elemen vektor tersebut. Apabila kedua kelas dapat terpisah secara sempurna oleh *hyperplane* yang didefinisikan $\mathbf{x}^{\mathrm{T}}\mathbf{w}+\gamma=0$, maka terdapat dua bidang yang sejajar yang merupakan batas kedua kelas yaitu $\mathbf{x}^{\mathrm{T}}\mathbf{w}+\gamma=+1$ untuk kelas +1 dan $\mathbf{x}^{\mathrm{T}}\mathbf{w}+\gamma=-1$ untuk kelas -1. SVM dengan bidang pemisah yang nonlinier diperoleh dengan mentransformasikan formulasi SVM standar sebagai berikut :

$$\mathbf{w}=\mathbf{A}^{\mathrm{T}}\mathbf{Du} \tag{2}$$

Dalam mencari solusi *masalah* non linier digunakan "*kernel trick*" yaitu menambahkan fungsi kernel Gaussian

$$\mathrm{K}(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\mu\|\mathbf{x}_i-\mathbf{x}_j\|^2)\ ,\ \mu>0 \tag{3}$$

dengan μ merupakan parameter kernel dan *i, j*=1,2,...,*m*. Dengan mensubtitusikan persamaan 2 kedalam model 2.1. maka masalah nonlinier diperoleh sebagai berikut

$$\min_{(\boldsymbol{w},\gamma,\boldsymbol{y})\in R^{n+1+m}} v\boldsymbol{e}^{T}\boldsymbol{y}+\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{DAA}^{\mathrm{T}}\mathbf{Du} \tag{4}$$

dengan kendala $\mathbf{D}(\mathbf{AA}^{\mathrm{T}}\mathbf{Du}-\mathbf{e}\gamma)+\mathbf{y}\geq\mathbf{e}$

$\mathbf{y}\geq\mathbf{0}$

Solusi dari fungsi diatas untuk masalah nonlinier adalah $K(\mathbf{x}^{\mathrm{T}},\mathbf{A}^{\mathrm{T}})\mathbf{Du}=\gamma$. Dengan menggantikan $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ dengan kernel nonlinier $\mathrm{K}(\mathbf{A},\mathbf{A}^{\mathrm{T}})$ dan variabel y diminimalisasi dengan bobot $\frac{\mathbf{v}}{2}$ menghasilkan *nonlinier generalized SVM* dengan persamaan

$$\min_{(\boldsymbol{w},\gamma,\boldsymbol{y})\in R^{n+1+m}} \frac{\mathrm{v}}{2}\mathbf{y}^{\mathrm{T}}\mathbf{y}+\frac{1}{2}\left(\mathbf{u}^{\mathrm{T}}\mathbf{u}+\gamma^2\right) \tag{5}$$

dengan kendala $\mathbf{D}(\mathrm{K}(\mathbf{A},\mathbf{A}^{\mathrm{T}})\mathbf{Du}-\mathbf{e}\gamma)+\mathbf{y}\geq\mathbf{e}$ dan $\mathbf{y}\geq\mathbf{0}$

untuk menyelesaikan persamaan 2.3 didefinisikan fungsi kendala berikut

$$\mathbf{y}=(\mathbf{e}-\mathbf{D}\,\mathrm{K}(\mathbf{A},\mathbf{A}^{\mathrm{T}})\mathbf{Du}-\mathbf{e}\gamma)_+ \tag{6}$$

dengan mensubtitusikan fungsi kendala tersebut ke persamaan 3 diperoleh persamaan masalah SVM yang ekivalen dengan SVM optimasi tanpa kendala sebagai berikut

$$\min_{(\boldsymbol{u},\gamma)\in R^{m+1}} \frac{v}{2}\left\|(\mathbf{e}-\mathbf{D}(\mathrm{K}(\mathbf{A},\mathbf{A}^{\mathrm{T}})\mathbf{Du}-\mathbf{e}\gamma))_+\right\|_2^2+\frac{1}{2}\left(\mathbf{u}^{\mathrm{T}}\mathbf{u}+\gamma^2\right) \tag{7}$$

dimana pada $(.)_+$ komponen yang bernilai negatif digantikan dengan nilai nol. Persamaan 7 memiliki solusi yang unik tetapi fungsi objektifnya tidak memiliki turunan kedua sehingga Lee dan Mangasarian (2001) mengusulkan *smoothing technique* yang menggantikan fungsi plus $(.)_+$ dengan integral dari fungsi *sigmoid neural network*

$(1+\exp(-\alpha\mathbf{x}))^{-1}$ yang dapat dituliskan sebagai berikut [7]

$$\min_{(\boldsymbol{u},\gamma)\in R^{m+1}} \frac{v}{2}\left\|(\mathbf{e}-\mathbf{D}(\mathrm{K}(\mathbf{A},\mathbf{A}^{\mathrm{T}})\mathbf{Du}-\mathbf{e}\gamma))_+\right\|_2^2+\frac{1}{2}\left(\mathbf{u}^{\mathrm{T}}\mathbf{u}+\gamma^2\right) \tag{8}$$

dimana α adalah parameter *smoothing,*sehingga diperoleh model SSVM sebagai berikut

$$\min_{(\mathbf{u},\gamma)\in R^{m+1}} \Phi_\alpha(\mathbf{w},\gamma)=\min_{(\mathbf{u},\gamma)\in R^{m+1}} \frac{v}{2}\left\|p\left(\mathbf{e}-\mathbf{D}\left(\mathrm{K}\left(\mathbf{A},\mathbf{A}^{\mathrm{T}}\right)\mathbf{Du}-\mathbf{e}\gamma\right),\alpha\right)\right\|_2^2 +\frac{1}{2}\left(\mathbf{u}^{\mathrm{T}}\mathbf{u}+\gamma^2\right) \tag{9}$$

Optimasi persamaan 9 dapat diselesaikan dengan pendekatan numerik salah satunya dengan metode *Newton-Armijo.* Langkah awal untuk memulai algoritma Newton Armijo adalah dengan menginisiasi $(\mathbf{w}^{(0)},\gamma^{(0)})\in \mathrm{R}^{m+1}$ dimana notasi $\mathbf{w}^{(i)}$ menunjukkan **w** iterasi ke-i. Selanjutnya mengulanginya sampai gradien dari fungsi objektif 9 sama dengan nol atau $\nabla\Phi_\alpha(\mathbf{w}^{(i)},\gamma^{(i)})=0$. Selanjutnya menghitung $(\mathbf{w}^{i+1},\gamma^{i+1})$ sebagai berikut

1. *Newton Direction* : menentukan arah $d^{(i)}\in R^{n+1}$ sebagai berikut

$$\nabla^2\Phi_\alpha\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)d^{(i)}=-\nabla\Phi_\alpha\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)^T \tag{10}$$

2. *Armijo Stepsize* : memilih stepsize $\lambda_i\in R$ sedemikian sehingga:

$$\left(\mathbf{w}^{(i+1)},\gamma^{(i)}\right)=\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)+\lambda_i d^{(i)}$$

dimana λ_i=maksimumkan $\left\{1,\frac{1}{2},\frac{1}{4},\ldots\right\}$ sehingga:

$$\Phi_\alpha\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)-\Phi_\alpha\left(\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)+\lambda_i d^{(i)}\right)\geq-\delta\lambda_i\nabla\Phi_\alpha\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)d^{(i)} \tag{11}$$

dengan $\delta=\left(0,\frac{1}{2}\right)$

Saat $\nabla\Phi_\alpha\left(\mathbf{w}^{(i)},\gamma^{(i)}\right)=0$, iterasi pada algoritma Newton-Armijo berhenti, dan diperoleh nilai **w** dan γ yang konvergen kemudian didapatkan fungsi pemisah untuk klasifikasi linier sebagai berikut

$$f(\mathbf{x})=sign\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}-\gamma\right) \tag{12}$$

JURNAL SAINS DAN SENI ITS Vol. 4, No.1, (2015) 2337-3520 (2301-928X Print)

B. Reduced Support Vector Machine

Terdapat dua masalah besar dalam klasifikasi data besar yang non linier, yang pertama kesulitan komputasi dalam memecahkan masalah optimasi tanpa kendala dan melibatkan fungsi kernel yang membutuhkan memori sangat besar. Pada umumnya komputer mengalami *out of memory* bahkan sebelum dimulai proses pencarian solusi. Yang kedua kesulitan dalam menggunakan formula yang tidak terlihat, untuk bidang pemisah . untuk menyelesaikan masalah tersebut muncul sebuah ide menyelesakan bidang pemisah non linier pada data besar dengan hanya menggunakan sebagian karakteristik dari data. Hal inilah yang mengawali ide dasar dari RSVM. Dengan formulasi data penuh (*full set*) $\mathbf{A} \in R^{m \times n}$ dengan *square kernel* $K(\mathbf{A}, \mathbf{A}^{\mathrm{T}}) \in R^{m \times m}$dimodifikasi sedemikian hingga *reducedd* dataset $\bar{\mathbf{A}} \in R^{\bar{m} \times n}$ dengan diagonal matriks $\bar{\mathbf{D}}$ dan matriks kernel $K(\mathbf{A}, \bar{\mathbf{A}}^{\mathrm{T}}) \in R^{m \times \bar{m}}$. Selanjutnya algoritma tersebut diselesaikan dengan *smoothing technique*. Hasil modifikasi tersebut dirumuskan dengan menggantikan $\mathbf{A}^{\mathrm{T}}$ dengan $\bar{\mathbf{A}}^{\mathrm{T}}$ sebagai berikut

$$\min_{(\bar{\mathbf{u}},\gamma,\boldsymbol{y}) \in R^{\bar{m}+1+m}} \frac{v}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} + \frac{1}{2}\left(\bar{\mathbf{u}}^{\mathrm{T}}\bar{\mathbf{u}} + \gamma^2\right) \qquad (13)$$

dengan kendala $\mathbf{D}(\mathrm{K}(\mathbf{A}, \bar{\mathbf{A}}^{\mathrm{T}})\bar{\mathbf{D}}\bar{\mathbf{u}} - \mathbf{e}\gamma) + y \geq \mathbf{e}$ dan $\mathbf{y} \geq \mathbf{0}$.

Persamaan diatas menghasilkan solusi untuk masalah penggunaan matriks yang besar dan waktu pemrosesan data. Menurut Lee dan Mangasarian (2001) pada [9] menyebutkan bahwa secara garis besar algoritma RSVM dapat dituliskan sebagai berikut:

1. Memilih subset matriks $\bar{\mathbf{A}} \in R^{\bar{m} \times n}$ dari matriks awal $\mathbf{A} \in R^{m \times n}$ secara random dengan $\bar{m}$ sebesar 1% hingga 10%.
2. Menyelesaikan persamaan SSVM yang telah dimodifikasi berikut dan diselesaikan dengan algoritma *Newton-Armijo* dimana $\mathbf{A}^{\mathrm{T}}$ digantikan oleh $\bar{\mathbf{A}}^{\mathrm{T}}$ dengan $\bar{\mathbf{D}} \subset \mathbf{D}$.

$$\min_{(\bar{\mathbf{u}},\gamma) \in R^{\bar{m}+1}} \frac{v}{2}\left\| p\left(\mathbf{e} - \mathbf{D}\left(\mathrm{K}\left(\mathbf{A}, \bar{\mathbf{A}}^{\mathrm{T}}\right)\bar{\mathbf{D}}\bar{\mathbf{u}} - \mathbf{e}\gamma\right), \alpha\right)\right\|_2^2 + \frac{1}{2}\left(\bar{\mathbf{u}}^{\mathrm{T}}\bar{\mathbf{u}} + \gamma^2\right) \qquad (14)$$

3. Bidang pemisah dengan $\mathbf{A}^{\mathrm{T}}$ digantikan oleh $\bar{\mathbf{A}}^{\mathrm{T}}$ sehingga $(K(\mathbf{x}^{\mathrm{T}}, \bar{\mathbf{A}}^{\mathrm{T}})\bar{\mathbf{D}}\bar{\mathbf{u}} - \gamma)$ dimana $(\bar{\mathbf{u}}, \gamma) \in R^{\bar{m}+1}$ adalah solusi unik dari persamaan 2.18 dan $\mathbf{x} \in R^n$ yang merupakan variabel input untuk data yang baru.
4. Input yang baru diklasifikasikan kedalam kelas {+1} dan {-1} tergantung dari fungsi

$$f(\mathbf{x}) = \left(\mathrm{K}\left(\mathbf{x}^{\mathrm{T}}, \bar{\mathbf{A}}^{\mathrm{T}}\right)\bar{\mathbf{D}}\bar{\mathbf{u}} - \gamma\right)_+ \qquad (15)$$

dengan *f(x)* adalah *plus function* yang nilainya adalah +1 atau 0.

C. Seleksi Parameter

Dalam penelitian ini digunakan fungsi kernel gaussian dengan metode seleksi parameter *Uniform Design (UD)*. Parameter yang diseleksi adalah parameter *v* dan μ. Parameter μ merupakan kunci perfomansi dari model, sehingga nilai μ menjadi parameter pertama yang akan diseleksi. Parameter kernel μ menunjukkan *non linier mapping* dan data menjadi *feature space* berdimensi tinggi.

Kernel gaussian tidak hanya bergantung pada parameter μ tetapi juga pada jarak antara dua titik. Huang dkk,(2007) memberikan alternatif metode untuk menentukan batas jangkauan nilai μ [10].

Parameter *v* menentukan waktu *trade off* antara meminimalisir *training error* dan kompleksitas model. Letak nilai v yang optimal tergantung pada model SVM yang digunakan. Nilai v yang optimal untuk SVM terletak diantara 10^{-2} hingga 10^4. Pada RSVM saat dilakukan *reduced* kernel pada dasarnya mengubah kompleksitas dari model sehingga untuk RSVM nilai v yang optimum terletak diantara 10^0 hingga 10^6.

Pada metode ini digunakan dua tahap percobaan, pertama mencobakan 13 kombinasi v dan μ, kemudian dari kombinasi tersebut dipilih yang memiliki akurasi terbaik. Hasil kombinasi pada tahap pertama menjadi titik tengah wilayah penyeleksian yang baru pada tahap kedua. Pada tahap kedua mencobakan kembali 8 titik kombinasi, sehingga total kombinasi titik yang dicobakan sebanyak 21 titik. Huang,dkk pada [10] , menggunakan nilai parameter logaritma berbasis 2.

Pada setiap kombinasi v dan μ digunakan metode *k-fold cross validation* untuk membagi data menjadi *training* dan *testing*. Metode ini membagi data menjadi k bagian secara random. Setiap bagian memiliki proporsi kelas yang sama dengan proporsi kelas awal. Setiap bagian akan dijadikan data testing dan sisanya dijadikan data training, sehingga akan didapatkan sebanyak k akurasi, hasil akurasi menggunakan metode ini merupakan rata-rata dari k akurasi tersebut.

D. Evaluasi Performansi Model

Efektivitas model dievaluasi menggunakan ketepatan akurasi dan lama waktu *running* model tersebut. Ketepatan akurasi dapat ditentukan menggunakan nilai yang terdapat dalam tabel kontingensi berikut ini [8].

Tabel 1.
Kontingensi Ketepatan Klasifikasi

Aktual	Prediksi	
	I (*Negative*)	II (*Positive*)
Negative	*True Negative* (TN)	*False Positive* (FP)
Positive	*False Negative* (FN)	*True Posistive* (TP)

Dengan menggunakan Tabel 1 maka tingkat ketepatan akurasi suatu klasifikasi dapat diukur sebagai berikut:

$$\text{Ketepatan akurasi (\%)} = \frac{TN + TP}{TP + FP + FN + TN}$$

model yang lebih baik merupakan model dengan ketepatan akurasi yang lebih tinggi dan lama waktu *running* yang singkat

III. METODOLOGI PENELITIAN

A. Metode Analisis Data

Metode analisis yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Mengidentifikasi karakteristik data hasil simulasi dan data riil

JURNAL SAINS DAN SENI ITS Vol. 4, No.1, (2015) 2337-3520 (2301-928X Print)

2. Melakukan Klasifikasi data untuk setiap tipe dan jumlah data menggunakan *Smooth Support Vector Machine* dengan tahapan
   a. Menentukan kombinasi parameter yang paling tepat untuk model fungsi kernel Gaussian dengan teknik KCV berdasarkan tingkat akurasi tertinggi.
   b. Membentuk data training dan testing untuk setiap dataset.
   c. Menentukan parameter RSVM menggunakan Algoritma Newton Armijo
   d. Membangun model SSVM menggunakan fungsi kernel Gaussian.
   e. Menghitung akurasi dari prediksi model yang terbentuk.
3. Melakukan klasifikasi data untuk setiap tipe dan jumlah data menggunkan *Reduced Support Vector Machine* dengan tahapan
   a. Menentukan kombinasi parameter yang paling tepat untuk model fungsi kernel Gaussian dengan teknik KCV berdasarkan tingkat akurasi tertinggi.
   b. Membentuk data training dan testing untuk setiap dataset.
   c. Me*reduced* dataset sebesar 10% secara *stratified.*
   d. Menentukan parameter RSVM menggunakan Algoritma Newton Armijo
   e. Membangun model RSVM menggunakan fungsi kernel Gaussian..
   f. Menghitung akurasi dari prediksi model yang terbentuk.
4. Membandingkan tingkat akurasi ketepatan klasifikasi dan waktu untuk *running* metode RSVM dan SSVM.

B. Data Eksperimen

1) Data Simulasi

Data simulasi yang digunakan terdiri dari tiga tipe data yaitu data untuk kasus linier, lingkaran dan data *checkerboard.* Masing-masing tipe data dibagi menjadi 4 kelompok dengan jumlah data yang berbeda yaitu kelompok dengan jumlah data 500, 1000, 3000 dan 10.000. Variabel tersebut merupakan data yang dibangkitkan melalui distribusi normal dengan rata-rata dan varian yang sama**.** Variabel yang digunakan dalam penelitian ini adalah x_1 dan x_2 yang merupakan variabel prediktor dan y sebagai variabel respon kategorik 1 dan -1 untuk lebih jelasnya variabel penelitian untuk data simulasi dapat dilihat pada Tabel 2.

Tabel 2.

Variabel Simulasi

Kasus	Jumlah Data	Variabel Prediktor	Variabel Respon
Linier, Lingkaran dan *checkerboard*	500	x1 (skala rasio) x2 (skala rasio)	y:kelas (skala nominal: 1,-1)
	1000	x1 (skala rasio) x2 (skala rasio)	y:kelas (skala nominal: 1,-1)
	3000	x1 (skala rasio) x2 (skala rasio)	y:kelas (skala nominal: 1,-1)
	10000	x1 (skala rasio) x2 (skala rasio)	y:kelas (skala nominal: 1,-1)

Fungsi yang digunakan untuk mengklasifikasikan data linier adalah sebagai berikut

$$f(x_1, x_2) = \begin{cases} 1 & x_1 < x_2 \\ -1 & x_1 \geq x_2 \end{cases} \quad (16)$$

Sedangkan untuk data simulasi lingkaran menggunakan fungsi berikut

$$f(x_1, x_2) = \begin{cases} 1 & \sqrt{x_1^2 + x_2^2} < 1,2 \\ -1 & \sqrt{x_1^2 + x_2^2} \geq 1,2 \end{cases} \quad (17)$$

sedangkan untuk data checkerboard tidak digunakan sebuah fungsi tetapi hanya menyusun titik-titik dengan kategori {+1} dan {-1}. Dari data simulasi tersebut digunakan proporsi mendekati 0,5 dengan rentang antara 0,45 hingga 0,55, hal ini dimaksudkan agar memiliki proporsi yang cenderung *balance* sehingga mendukung untuk proses klasifikasi.

2) Data Riil

Data riil yang digunakan dalam penelitian ini adalah data *Spambase* sebanyak 4601 data dengan 57 variabel prediktor dan y sebagai variabel respon kategorik. Data ini didapatkan dari website *University of California, Irvine* (http://archive.ics.uci.edu/ml/datasets.html) yang diunggah oleh George Forman. Variabel penelitian untuk data riil ini merupakan persentase kata-kata atau karakter tertentu terhadap keseluruhan kata dalam suatu email.. Variabel responnya adalah y atau kelas dengan skala nominal yaitu spam dan bukan spam, sedangkan semua variabel prediktor berskala rasio.

IV. HASIL DAN PEMBAHASAN

A. Deskripsi Data Simulasi

Data simulasi dibangkitkan menggunakan distribusi yang sama. Rasio dari kelas {+1} terhadap kelas {-1} untuk semua tipe dan jumlah data diperlihatkan pada tabel berikut.

Tabel 3.

Proporsi kelas +1 terhadap kelas -1

Tipe Data	Jumlah Data			
	500	1.000	3000	10.000
Linier	0,528	0,511	0,504	0,504
Lingkaran	0,496	0,498	0,514	0,515
Checkerboard	0,494	0,491	0,480	0,480

Tabel 3 menunjukkan bahwa secara keseluruhan nilai proporsi kelas +1 terhadap kelas -1 terletak disekitar nilai 0,5. Hal ini menunjukkan bahwa proporsi antara kelas tersebut mendekati *balance* sehingga data tersebut mendukung untuk dilakukan proses klasifikasi.

B. Klasifikasi Data Simulasi menggunakan SSVM dan RSVM

Proses seleksi parameter dan pembuatan model pada tugas akhir ini menggunakan software Matlab versi R2010a 32bit. Spesifikasi komputer yang digunakan memiliki prosesor intel pentium 2020M @2,40GHz dengan RAM sebesar 2 gigabit.

Data simulasi akan diklasifikasikan menggunakan SSVM dan RSVM, dengan terlebih dahulu melakukan proses seleksi parameter. Parameter v dan μ yang telah diseleksi, kemudian dijadikan input pada proses klasifikasi dan didapatkan hasil klasifikasi dengan tingkat akurasi seperti pada

JURNAL SAINS DAN SENI ITS Vol. 4, No.1, (2015) 2337-3520 (2301-928X Print)

Tabel 4.

Akurasi SSVM dan RSVM pada Data Linier

Tipe	Metode	Jumlah Data			
		500	1000	3000	10000
Linier	SSVM	100%	100%	100%	NA
	RSVM	100%	100%	100%	99.98%
Lingkaran	SSVM	99.53%	99.93%	99.84%	NA
	RSVM	99.39%	99.83%	99.77%	99.95%
Checker-board	SSVM	97.96%	99.83%	99.93%	NA
	RSVM	95.37%	99.63%	99.80%	99.95%

Pada tabel 4, akurasi yang dicetak tebal merupakan akurasi tertinggi untuk setiap jumlah data. Dalam tabel tersebut data linier diklasifikasikan dengan sempurna untuk data berjumlah 500, 1000 dan 3000, sedangkan untuk data berjumlah 10000 SSVM tidak memberikan hasil karena komputer mengalami *out of memory* saat dilakukan proses klasifikasi. Hasil yang sama dihasilkan oleh metode RSVM yang memberikan akurasi sempurna pada data simulasi dengan jumlah data 500, 1000, dan 3000. Pada data berjumlah 10000 metode juga menghasilkan akurasi yang mendekati sempurna yaitu 99,98%.

Pada data lingkaran akursi dari SSVM lebih besar daripada RSVM pada jumlah data 500, 1000 dan 3000, tetapi perbedaan akurasi antara kedua metode tersebut relatif kecil secara berturut-turut yaitu sebesar 0,14%, 0,1% dan 0,07%. Sedangkan pada data yang berjumlah 10000 SSVM sudah tidak mampu untuk melakukan proses klasifikasi karena komputer mengalami *out of memory* sama seperti data linier.

Pada data *checkerboard* bahwa metode SSVM juga memiliki tingkat akurasi yang lebih tinggi daripada metode RSVM pada jumlah data 500,1000 dan 3000, sedangkan pada jumlah data 10000 SSVM juga tidak mampu melakukan proses klasifikasi. Metode RSVM masih dapat melakukan proses klasifikasi dengan jumlah data 10000 dengan akurasi sebesar 99,95%.

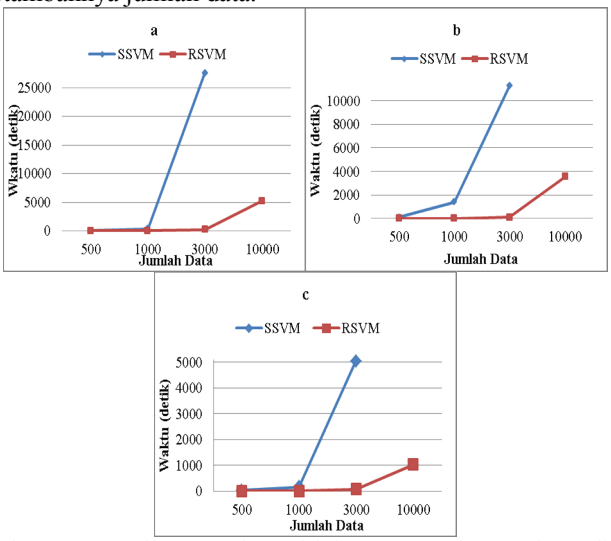
Sedikit berbeda dengan data bertipe lingkaran, pada data bertipe *checkerboard* ini selisih akurasi antara metode SSVM dan RSVM pada jumlah data sebanyak 500 menghasilkan akurasi yang sedikit lebih tinggi yaitu sebesar 2,59%. Sedangkan selisih metode SSVM dan RSVM untuk data dengan jumlah 1000 dan 3000 tidak terlalu berbeda dengan tipe data lingkaran yaitu sebesar 0,2% dan 0,18%. Dari kasus ini terlihat bahwa dengan bertambahnya jumlah data, akurasi dari metode RSVM akan mendekati metode SSVM.

Tabel 5.

Waktu Seleksi Parameter SSVM dan RSVM pada Data Linier

Tipe	Metode	Jumlah Data			
		500	1000	3000	10000
Linier	SSVM	57.45	375.52	27573.00	
	RSVM	2.25	5.05	228.26	5261.50
Lingkaran	SSVM	117.26	1410.40	11248.00	
	RSVM	2.27	7.05	107.12	3535.55
Checker-board	SSVM	36.13	167.99	5045.00	
	RSVM	1.92	5.52	71.30	1032.10

Waktu yang dicetak tebal pada Tabel 5 menunjukkan waktu yang lebih kecil. Terlihat bahwa waktu yang diperlukan untuk seleksi parameter metode SSVM secara keseluruhan lebih lama daripada metode RSVM. Berikut ini adalah visualisasi dari pertambahan waktu seleksi parameter seiring dengan bertambahnya jumlah data.



Gambar 2. Waktu untuk seleksi parameter a. data linier b. data lingkaran c. data *checkerboard*

Berdasarkan Gambar 2 di semua tipe data saat berjumla 3000 data terdapat kenaikan waktu proses yang sangat tinggi pada metode SSVM dan dengan data 10000, komputer sudah mengalami *out of memory*. Sedangkan waktu untuk metode RSVM mengalami kenaikan yang cenderung landai.

Tabel 6.

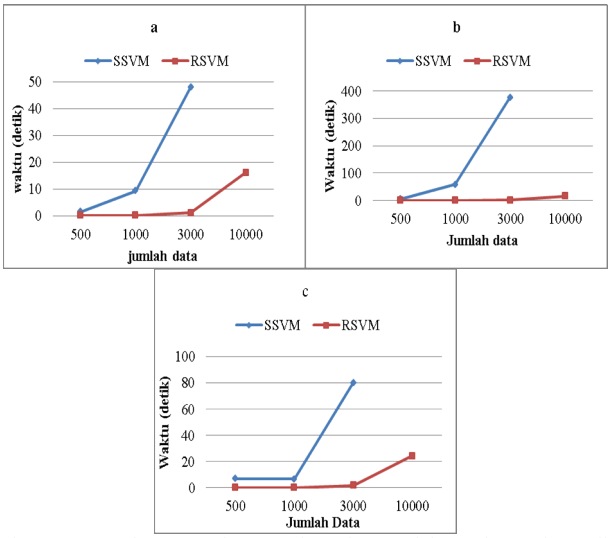
Waktu Membentuk Model SSVM dan RSVM pada Data Linier

Tipe	metode	Jumlah Data			
		500	1000	3000	10000
Linier	SSVM	1.51	9.38	48.05	NA
	RSVM	0.17	0.18	1.16	16.19
lingkaran	SSVM	4.75	58.42	376.97	NA
	RSVM	0.09	0.20	1.07	15.25
Checker-board	SSVM	6.97	6.75	80.23	NA
	RSVM	0.20	0.23	1.88	24.31

Tabel 6 menunjukkan bahwa waktu yang dibutuhkan untuk membentuk model RSVM lebih cepat dibanding membentuk model SSVM. Pada data yang berjumlah 3000 terlihat perbedaan yang jelas terlihat, SSVM membutuhkan waktu 48 detik sedangkan RSVM membutuhkan waktu 1,16 detik. Perbedaan tersebut terjadi karena pada metode SSVM menggunakan full kernel (tanpa *reducedd*) yang melibatkan perhitungan besar sehingga waktu yang dibutuhkan menjadi lebih lama dibandingkan RSVM yang menggunakan sebagian kernel saja.

Pada Gambar 3 diperlihatkan bahwa waktu yang digunakan untuk membentuk model SSVM selalu lebih lama daripada metode RSVM. Pada metode SSVM dengan jumlah data 3000 mengalami kenaikan yang sangat besar, sedangkan pada metode RSVM kenaikan waktu membentuk model cenderung lebih landai.

JURNAL SAINS DAN SENI ITS Vol. 4, No.1, (2015) 2337-3520 (2301-928X Print)



Gambar 3. Waktu untuk membentuk model pada a.data linier b. data lingkaran c. data *checkerboard*

Berdasarkan performansi kedua metode pada data simulasi, dapat disimpulkan bahwa SSVM dan RSVM memiliki performansi yang cenderung sama untuk data yang berjumlah relatif kecil (kurang dari 1000) sedangkan untuk jumlah data yang besar (lebih dari 1000) RSVM memiliki performansi yang baik daripada SSVM, terutama dalam hal waktu yagn dibutuhkan untuk seleksi parameter dan pembentukan model.

C. *Klasifikasi Data Spambase*

Nilai parameter v dan μ pada seleksi parameter dijadikan input untuk proses klasifikasi sehingga menghasilkan tingkat akurasi, waktu yang diperlukan untuk seleksi parameter dan membentuk model adalah sebagai berikut

Tabel 7.

Performansi SSVM dan RSVM pada Data *Spambase*

Metode	Akurasi	Waktu seleksi Parameter (detik)	Waktu membentuk model (detik)
SSVM	91,72%	76770,00	3033,92
RSVM	91,16%	719,65	19,60

Tabel 7 menunjukkan bahwa tingkat akurasi dari metode SSVM sebesar 91,72% lebih tinggi dari metode RSVM sebesar 91,16%. Waktu yang dibutuhkaan metode SSVM untuk seleksi parameter dan membentuk model metode SSVM lebih lama daripada metode RSVM. Metode SSMV memerlukan waktu 76770 detik atau sekitar 21 jam lebih 19 menit sedangkan metode RSVM hanya 719,65 detik atau sekitar 12 menit. Akurasi dari metode SSVM memang lebih tinggi dari RSVM tetapi selisihnya sangat kecil yaitu sebesar 0,56% sedangkan jika dilihat dari perbandingan waktu yang dibutuhkan untuk seleksi parameter dan membentuk model sangat berbeda jauh, sehingga dari kasus ini disarankan untuk menggunakan RSVM.

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Dari hasil dan pembahasan yang telah diperoleh didapatkan kesimpulan sebagai berikut.

1) Secara umum pada data simulasi RSVM menghasilkan akurasi yang tinggi dengan waktu seleksi parameter dan pembentukan model yang singkat. Akurasi RSVM pada data simulasi lebih besar dari 99%. Waktu untuk seleksi parameter cenderung landai dan ditandai kenaikan yang besar pada data berjumlah 10000. Waktu yang dibutuhkan untuk membentuk model kurang dari 2 detik dengan data 500,1000, dan 3000, sedangkan pada data 10.000 berturut turut dari linier, lingkaran dan *checkerboard* sebesar 15, 16 dan 24 detik.
2) Pada berbagai jumlah data metode SSVM dan RSVM memberikan akurasi yang cenderung sama. Dengan jumlah data yang relatif kecil (kurang dari 1000) metode SSVM membutuhkan waktu *running* yang juga hampir sama dengan RSVM, tetapi untuk jumlah data lebih dari 1000 waktu yang dibutuhkan metode SSVM mengalami peningkatan yang tinggi. Pada metode RSVM waktu yang dibutuhkan dengan berbagai jumlah data lebih cepat dibandingkan SSVM. Berdasarkan kedua hal tersebut performansi RSVM lebih baik daripada SSVM untuk jumlah data yang relatif besar (lebih dari 1000). Sedangkan pada data yang relatif kecil (kurang dari 1000) kedua metode memberikan performansi yang sama.

B. *Saran*

Saran untuk penelitian yang akan datang adalah sebagai berikut.

1) Menggunakan metode SSVM dan RSVM untuk permasalahan klasifikasi multikelas sehingga dapat diketahui bagaimana performansi kedua metode tersebut.
2) Data simulasi yang dibangkitkan tidak hanya berdasarkan banyaknya data tetapi juga berdasarkan banyaknya variabel.

DAFTAR PUSTAKA

[1] Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Technique.* Burlington: Elsevier Inc.

[2] Johnson, R., & Winchern, D. (2007). *Applied Multivariate Statistical Analysis.* New Jersey: Pearson Prentice Hall.

[3] Rachman, F., & Purnami, S. W. (2012). Klasifikasi Tingkat keganasan Breast Cancer dengan menggunakan Regresi Logistik Ordinal dan Support Vector Machine (SVM). *Jurnal Sains dan Seni ITS Vol.1 NO.1*, D130.

[4] Huang, J., Jingjing, L., & Ling, C. (2003). Comparing Naive Bayes, Decision Trees and SVM with AUC and accuracy. *International Conference on Data Mining*.

[5] Byvatov, E. e. (2003). Comparison of Support Vector Machine and Artificial Neural Network System for Drug/Nondrug classification. *Chem Inf Compu Sci*, 1882-1889.

[6] Wu, Q., & Wang, W. (2013). Peicewise-Smooth Support Vector Machine for Classification. *Mathematical Problems in Engineering*, 7.

[7] Lee, Y. J., & Mangasarian, O. L. (2001). RSVM:Reduced Support Vector Machine. *IN Proceedings of the First SIAM International Conference on Data Mining*.

[8] Purnami, S. W., Zain, J. M., & Heriawan, T. (2011). An alternative algorithm for classification large categorical dataset:k-mode clustering reduced support vector machine . *International Journal of Database Theory and Application*, Vol.4,No.1.

[9] Nugroho, A. S. (2003). Support Vector Machine Teori dan Aplikasinya dalam BioInformatika. *Ilmu Komputer*.

[10] Huang, C., Lin, D., & Huang, S. (2007). Model Selection for Support Vector machine via uniform Design. *Computational Statistic and Data Analysis vol 52*, 335-346.