ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 3, Nomor 4, Tahun 2014, Halaman 751 - 760
Online di: http://ejournal-s1.undip.ac.id/index.php/gaussian

KLASIFIKASI WILAYAH DESA-PERDESAAN DAN DESA-PERKOTAAN WILAYAH KABUPATEN SEMARANG DENGAN *SUPPORT VECTOR MACHINE* (SVM)

Mekar Sekar Sari[1], Diah Safitri[2], Sugito[3]
[1]Mahasiswa Jurusan Statistika FSM UNDIP
[2,3]Staf Pengajar Jurusan Statistika FSM UNDIP

ABSTRACT

This research will be carry out classification based on the status of the rural and urban regions that reflect the differences in characteristics/ conditions between regions in Indonesia with Support Vector Machine (SVM) method. Classification on this issue is working by build separation functions involving the kernel function to map the input data into a higher dimensional space. Sequential Minimal Optimization (SMO) algorithms is used in the training process of data classification of rural and urban regions to get the optimal separation function (hyperplane). To determine the kernel function and parameters according to the data, grid search method combined with the leave-one-out cross-validation method is used. In the classification using SVM, accuracy is obtained, which the best value is 90% using Radial Basis Function (RBF) kernel functions with parameters C=100 dan $\gamma=2^{-5}$.

Keywords : classification, support vector machine, sequential minimal optimization, grid search, leave-one-out, cross validation, rural, urban

1. PENDAHULUAN

Wilayah Indonesia dibagi ke dalam beberapa tingkat wilayah administratif, yaitu provinsi, kabupaten/kota, kecamatan, dan desa atau disebut dengan nama lain yang merupakan wilayah administratif terkecil (Badan Pusat Statistik, 2010). Untuk berbagai keperluan, data mengenai klasifikasi wilayah desa dan kota sangat bermanfaat terutama dalam hal perencanaan pembangunan. Perencanaan pembangunan wilayah mencakup berbagai aspek yang tentunya mempertimbangkan peran keterkaitan antara desa dan kota. Mike Douglass (1998) dalam Tarigan (2003), melalui konsep agropolitan menekankan bahwa pengembangan desa dapat tercapai dengan baik apabila desa tersebut dikaitkan dengan pengembangan kota dalam wilayah tersebut. BPS melakukan penggolongan wilayah desa-perdesaan dan desa-perkotaan yang melibatkan beberapa variabel yang telah ditetapkan berdasarkan Peraturan Kepala Badan Pusat Statistik Nomor 37 Tahun 2010 tentang Klasifikasi Perkotaan dan Perdesaan di Indonesia.

Support Vector Machine (SVM) adalah suatu teknik untuk menemukan fungsi pemisah (*hyperplane*) yang bisa memisahkan dua himpunan data dari dua kelas yang berbeda (Vapnik, 1995). Ide dasar SVM adalah memaksimalkan batas *hyperplane* (Prasetyo, 2012). Pada penelitian ini akan dilakukan klasifikasi terhadap wilayah desa-perdesaan dan desa-perkotaan di Kabupaten Semarang dengan menggunakan metode *Support Vector Machine* (SVM).

2. TINJAUAN PUSTAKA

2.1. Pengertian Desa-Perkotaan dan Desa-Perdesaan

Menurut Badan Pusat Statistik (2010), untuk memahami klasifikasi desa perkotaan perdesaan perlu dijelaskan tentang beberapa pengertian secara statistik, sebagai berikut:

a Daerah perkotaan, adalah suatu wilayah administratif setingkat desa/kelurahan yang memenuhi persyaratan tertentu dalam hal kepadatan penduduk, persentase rumah tangga pertanian, dan sejumlah fasilitas perkotaan, sarana pendidikan formal, sarana kesehatan umum, dan sebagainya.
b Daerah perdesaan, adalah suatu wilayah administratif setingkat desa/kelurahan yang belum memenuhi persyaratan tertentu dalam hal kepadatan penduduk, persentase rumah tangga pertanian, dan sejumlah fasilitas perkotaan, sarana pendidikan formal, sarana kesehatan umum, dan sebagainya.

2.2. Konsep *Support Vector Machine* (SVM)

Permasalahan pada SVM adalah memisahkan dua kelas dengan suatu fungsi yang didapatkan dari data *training* yang tersedia (Gunn, 1998). SVM memetakan vektor input ke dimensi ruang yang lebih tinggi di mana fungsi pemisah (*hyperplane*) maksimal dibangun (Vapnik, 1995). Dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. yang didefinisikan sebagai : $g(x) \coloneqq sign(f(x))$.
dengan $f(x) = \boldsymbol{w}^T\boldsymbol{x} + b,\ \boldsymbol{w}, \boldsymbol{x} \in \boldsymbol{R}^n$ dan b $\in \boldsymbol{R}$**,** $\boldsymbol{w}$ dan b adalah parameter-parameter yang dicari nilainya (Santosa, 2007).

Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua himpunan objek dari dua kelas (Santosa, 2007). Sampel yang terletak di sepanjang *hyperplane* disebut *support vector* (Srivastava, D.K. and Bhambhu, 2005).

Suatu permasalahan klasifikasi linier dengan data latih $x_i \in \boldsymbol{R}^n$ dan kelas label $y_i \in \{-1, +1\}$ dirumuskan pada: $\boldsymbol{D} = \{(\boldsymbol{x_1}, \boldsymbol{y_1}), .., (\boldsymbol{x_l}, \boldsymbol{y_l})\}, \boldsymbol{x} \in \boldsymbol{R}^n, i\ =\ 1, 2, \dots, l,$
Fungsi *hyperlane* SVM didefinisikan, sebagai berikut: $\boldsymbol{w}^T\boldsymbol{x} + b = 0$. Di dalam Prasetyo (2012), data x_i yang termasuk kelas -1 dapat dirumuskan sebagai data yang memenuhi pertidaksamaan : $\boldsymbol{w}^T\boldsymbol{x} + b \leq -1$. Sedangkan data x_i yang termasuk kelas +1 dirumuskan : $\boldsymbol{w}^T\boldsymbol{x} + b \geq +1$

2.3. SVM pada *Linearly Separable Data*

Secara matematika, formulasi problem optimasi SVM untuk kasus klasifikasi linier di dalam *primal space* adalah:

$$min \frac{1}{2} \|\boldsymbol{w}\|^2$$

syarat : $y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) \geq 1, i = 1,2, \dots l$
dengan $\boldsymbol{x}_i$ adalah data input, y_i adalah keluaran dari data $\boldsymbol{x_i}$, $\boldsymbol{w}$ dan b adalah parameter-parameter yang dicari nilainya (Santosa, 2007). Margin optimal dihitung dengan memaksimalkan jarak antara *hyperplane* dan data terdekat. Selanjutnya permasalahan tersebut diformulasikan ke dalam *Quadratic Programming* (QP) problem yang merupakan salah satu bentuk persamaan optimasi. Optimalisasi ini dapat diselesaikan dengan *Lagrange Multiplier*. Permasalahan tersebut kemudian diubah menjadi fungsi *Lagrangian*, sebagai berikut:

$$L_p = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i [y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1]$$

α_i adalah *Lagrange Multipliers*, yang berkorespondensi dengan x_i. α_i bernilai nol atau positif ($\alpha_i \geq 0$) (Prasetyo, 2012).

Di dalam Santosa (2007), solusi dari problem optimisasi dengan pembatas ditentukan dengan mencari *saddle point* dari fungsi Lagrangian L_p. Fungsi ini harus diminimalkan terhadap w dan b, dan memaksimalkan L_p terhadap α_i. Kemudian dicari turunan pertama dari fungsi L_p terhadap variabel w dan b dan disamakan dengan 0. Dengan melakukan proses ini akan didapatkan dua kondisi optimalitas berikut:

1 kondisi 1 :

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \ \rightarrow = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x_i}$$

2 kondisi 2 :

$$\frac{\partial L_p}{\partial b} = 0 \ \rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0$$

Menurut Prasetyo (2012), untuk menyelesaikan masalah tersebut, modifikasikan persamaan L_p menjadi kasus pemaksimalan, dengan syarat optimasi untuk dualitasnya menggunakan kendala *Karush-Khun Tucker* (KKT), yaitu $\alpha_i[y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1] = 0$ dan $\alpha_i \geq 0,\ i = 1,2,\dots,l$. L_p kemudian dijabarkan sebagai berikut :

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i(\boldsymbol{w}.\boldsymbol{x_i}) - b\sum_{i=1}^{l} \alpha_i\, y_i + \sum_{i=1}^{l} \alpha_i$$

Persamaan tersebut selanjutnya berubah menjadi dualitas *Lagrange Multiplier* dengan persamaan, sebagai berikut:

$$L_d = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\, \alpha_j y_i y_j \boldsymbol{x_i} \boldsymbol{x_j}$$

Selanjutnya fungsi dual tersebut dimaksimalkan, sehingga persamaannya menjadi, sebagai berikut:

$$max_\alpha \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\, \alpha_j y_i y_j x_i x_j$$

Dengan kondisi $\alpha_i \geq 0, i = 1,2,\dots,l$ dan $\sum_{i=1}^{l} \alpha_i y_i = 0$. $\boldsymbol{x_i}\boldsymbol{x_j}$ merupakan dot-product dua data dalam data latih. Data pelatihan yang memiliki nilai $\alpha_i > 0$ adalah *support vektor* sedangkan sisanya memiliki nilai $\alpha_i = 0$. Dengan demikian fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*. *Hyperplane* (fungsi keputusan) didapatkan dengan formula:

$$f(z) = \left(\sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x_i}.\boldsymbol{z}\right) + b$$

dengan *l* merupakan jumlah data yang menjadi *support vektor*, $\boldsymbol{x_i}$ merupakan *support vektor*, **z** merupakan data uji yang akan diprediksi kelasnya, dan $\boldsymbol{x_i}.\boldsymbol{z}$ merupakan inner product antara $\boldsymbol{x_i}$ dan **z**.

2.4. SVM pada *Non-linearly Separable Data*

Menurut Tan *et al.* (2006) dalam Nuha dkk. (2012), pada kasus klasifikasi linier SVM ketika terdapat data yang tidak dapat dikelompokkan dengan benar (*nonseparable case*), rumusan SVM ditambah dengan adanya variabel *slack*. Formulasi dari permasalahan sebelumnya kemudian diubah menjadi berikut

$$min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i$$

dengan kendala $y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) + \xi_i \geq 1, \xi_i \geq 0, i = 1,2,\dots l$

di mana ξ_i adalah variabel *slack* yang digunakan untuk memberikan penalti terhadap data yang tidak memenuhi persamaan *hyperplane* $y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) \geq 1$.

Untuk meminimalkan nilai variabel *slack*, pada rumusan diberikan penalti dengan menambahkan nilai *cost* (C). Parameter C berguna untuk mengontrol pertukaran antara margin dan error klasifikasi (Prasetyo, 2012).Menurut Tan *et al.* (2006) dalam Nuha, dkk (2012), untuk kasus *nonseparable case*, kendala $\alpha_i \geq 0$ diubah menjadi $0 \leq \alpha_i \leq \mathrm{C}$.

2.5. Metode Kernel dan Klasifikasi Nonlinier pada SVM

Untuk data yang distribusi kelasnya tidak linier biasanya digunakan pendekatan kernel pada fitur data awal (Prasetyo, 2012). Menurut Tan *et al.* (2006) dalam Nuha dkk.(2012), pada klasifikasi nonlinier SVM, data **x** dipetakan oleh fungsi Φ(x) ke ruang vektor dengan dimensi yang lebih tinggi. Proses pemetaan pada fase ini memerlukan perhitungan dot-product dua buah data pada ruang fitur baru yang dinotasikan sebagai $\Phi(x_i).\Phi(x_j)$. Trik komputasi ini sering dikenal dengan trik kernel, sebagai berikut: $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{\Phi}(\boldsymbol{x_i}).\boldsymbol{\Phi}(\boldsymbol{x_j})$

Dan prediksi pada data dengan dimensi fitur yang baru diformulasikan dengan

$$f(\Phi(z)) = sign(w.\Phi(z) + b) = sign(\textstyle\sum_{i=1}^{l} \alpha_i y_i \Phi(x_i).\Phi(z) + b)$$

Dengan l adalah jumlah data yang menjadi *support vector*, x_i adalah *support vektor*, dan z adalah data uji yang akan diprediksi (Prasetyo, 2012).

Menurut Haykin (1999), fungsi kernel yang biasanya dipakai dalam literatur SVM:

- Linier : $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}.\boldsymbol{y}$
- *Polynomial* : $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}.\boldsymbol{y} + \boldsymbol{c})^{\boldsymbol{d}}$
- *Radial Basis Function* (RBF) : $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y}) = exp(-\gamma\|\boldsymbol{x} - \boldsymbol{y}\|^2)$, dengan $\gamma = \frac{1}{2\sigma^2}$
- *Tangent hyperbolic* (sigmoid): $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y}) = tanh(\sigma(\boldsymbol{x}.\boldsymbol{y}) + c)$

$\boldsymbol{x}$ dan $\boldsymbol{y}$ adalah pasangan dua data dari semua bagian data latih. Parameter $\sigma, c, d > 0$, merupakan konstanta. Menurut Vapnik (1995) dan Haykin (1999) dalam Santosa (2007), fungsi kernel yang legitimate diberikan oleh Teori Mercer di mana fungsi tersebut harus memenuhi syarat kontinu dan positif definite.

2.6. Optimasi *Hyperplane* SVM

Sequential Minimal Optimization (SMO) adalah algoritma untuk menyelesaikan masalah *quadratic programming* (QP) yang muncul selama *training* pada *Support Vector Machine* (Platt, 1998). Menurut Karatzoglou *et al* (2006), setiap langkahnya SMO memilih dua α_i untuk dioptimalkan secara bersama-sama dan menemukan nilai-nilai optimal untuk nilai-nilai α_i tersebut secara analitik, sehingga dapat menghindari optimasi QP secara numerik, dan memperbarui SVM untuk memberikan nilai-nilai optimal yang terbaru.

Menurut Platt (1998), tahapan optimasi *hyperplane* pada SVM meliputi:

1. Menyelesaikan Dua Pengali *Lagrange*
2. Menghitung b

Tahapan optimasi *hyperplane* SVM tersebut dapat dijabarkan, sebagai berikut:

1. Menyelesaikan Dua Pengali *Lagrange*

Menurut Platt (1998), QP problem pada *training* SVM dinyatakan, sebagai berikut:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \boldsymbol{K}(\boldsymbol{x_i}\boldsymbol{x_j})\alpha_{\boldsymbol{i}}\, \alpha_j$$

$$0 \leq \alpha_i \leq \mathrm{C},\ \forall i \text{ dan } \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0$$

Permasalahan QP tersebut diselesaikan dengan algoritma SMO. Suatu titik pada persamaan tersebut merupakan titik optimum apabila kondisi *Karush-Kuhn-Tucker* (KKT) terpenuhi dan $y_i y_j K(\boldsymbol{x_i x_j})$ bernilai positif semi-definit. Permasalahan QP terpenuhi jika semua i:

$$\alpha_i = 0 \rightarrow y_i f(x_i) \geq 1$$
$$0 < \alpha_i < C \rightarrow y_i f(x_i) = 1$$
$$\alpha_i = C \rightarrow y_i f(x_i) \leq 1$$

Dengan $f(x_i)$ merupakan nilai hasil prediksi data x_i. Di dalam Keerthi and Gilbert (2002) disebutkan untuk QP kondisi KKT merupakan syarat cukup sekaligus syarat perlu.

Di dalam Platt (1998), untuk menyelesaikan dua pengali *Lagrange*, SMO pertama kali menghitung pembatas pada pengali-pengali tersebut. Jika target $y_i \neq y_j$, maka batas berikut berlaku untuk α_j: $L = \max\{0, \alpha_j - \alpha_i\}, H = \min\{C, C + \alpha_j - \alpha_i\}$

Jika target $y_i = y_j$, batas yang berlaku: $L = \max\{0, \alpha_j + \alpha_i - C\}, H = \min\{C, \alpha_j + \alpha_i\}$

Jika nilai tersebut jatuh di luar batas L dan H, nilai α_j tersebut akan dipotong agar berada pada jangkauan tersebut. Optimal α_j dapat dirumuskan, sebagai berikut:

$$\alpha_j^{new} = \alpha_j + \frac{y_j(E_i - E_j)}{\eta}$$

dengan $E_i = f(x_i) - y_i$ atau *error training* ke-i

$E_j = f(x_j) - y_j$ atau *error training* ke-j

$\eta = K(\boldsymbol{x_i}, \boldsymbol{x_j}) + K(\boldsymbol{x_j}, \boldsymbol{x_j}) - 2K(\boldsymbol{x_i}, \boldsymbol{x_j})$

Pada kondisi normal, η akan bernilai kurang dari 0. Pada langkah selanjutnya, nilai α_j dipotong agar berada dalam rentang [L,H] :

$$\alpha_j^{new,clipped} = \begin{cases} H, & jika\ \alpha_j^{new} \geq H \\ \alpha_j^{new}, & jika\ L < \alpha_j^{new} < H \\ L, & jika\ \alpha_j^{new} \leq L \end{cases}$$

Setelah menemukan solusi untuk α_j, dicari nilai α_i, sebagai berikut:

$$\alpha_i^{new} = \alpha_i + s(\alpha_j - \alpha_j^{new,clipped})$$

dengan $s = y_i y_j$ (Platt, 1998).

Menurut Platt (1998), diluar kondisi normal, η tidak akan bernilai negatif. Pada beberapa peristiwa, SMO akan tetap bekerja meskipun ketika η tidak bernilai negatif, dalam hal ini fungsi tujuan Ψ yang harus dievaluasi pada setiap akhir ruas garis.

$$f_i = y_i(E_i + b) - \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x_i}) - s\alpha_j K(\boldsymbol{x_i}, \boldsymbol{x_j})$$
$$f_j = y_j(E_j + b) - s\alpha_i K(\boldsymbol{x_i}, \boldsymbol{x_j}) - \alpha_j K(\boldsymbol{x_j}, \boldsymbol{x_j})$$
$$L_i = \alpha_i + s(\alpha_j - L)$$
$$H_i = \alpha_i + s(\alpha_j - H)$$
$$\Psi_L = L_i f_i + L f_j + \frac{1}{2} L_i^2 K(\boldsymbol{x_i}, \boldsymbol{x_i}) + \frac{1}{2} L^2 K(\boldsymbol{x_j}, \boldsymbol{x_j}) + sLL_i K(\boldsymbol{x_i}, \boldsymbol{x_j})$$
$$\Psi_H = H_i f_i + H f_j + \frac{1}{2} H_i^2 K(\boldsymbol{x_i}, \boldsymbol{x_i}) + \frac{1}{2} H^2 K(\boldsymbol{x_j}, \boldsymbol{x_j}) + sHH_i K(\boldsymbol{x_i}, \boldsymbol{x_j})$$

2. Menghitung b

Nilai b selalu dihitung ulang pada setiap akhir tahapan, sehingga kondisi KKT terpenuhi untuk kedua masalah optimasi. b_i berikut akan valid jika α_i baru tidak berada pada batas:

$$b_i = E_i + y_i(\alpha_i^{new} - \alpha_i)K(\boldsymbol{x_i}, \boldsymbol{x_i}) + y_j(\alpha_j^{new,clipped} - \alpha_j)K(\boldsymbol{x_i}, \boldsymbol{x_j}) + b$$

b_j berikut akan valid jika α_j baru tidak berada pada batas :

$$b_j = E_j + y_i(\alpha_i^{new} - \alpha_i)K(\boldsymbol{x_i}, \boldsymbol{x_j}) + y_j(\alpha_j^{new,clipped} - \alpha_j)K(\boldsymbol{x_j}, \boldsymbol{x_j}) + b$$

Ketika kedua b_i dan b_j valid, keduanya bernilai sama. Jika kedua pengali *Lagrange* baru berada pada batas, dan jika L tidak sama dengan H maka interval di antara b_i dan b_j, semuanya adalah ambang yang konsisten dengan syarat KKT. SMO memilih nilai tengah antara b_i dan b_j (Platt, 1998).

2.7. Seleksi Parameter

1 *Cross-Validation*

Menurut Leidiyana (2013), *cross-validation* adalah pengujian standar yang dilakukan untuk memprediksi *error rate*. Data *training* dibagi secara random ke dalam beberapa bagian dengan perbandingan yang sama kemudian *error rate* dihitung bagian demi bagian, selanjutnya hitung rata-rata seluruh *error rate* untuk mendapatkan *error rate* secara keseluruhan. Laju error dapat dihitung dengan formulasi berikut:

$$Laju\ Error\ (Error\ Rate) = \frac{Banyaknya\ data\ yang\ diprediksi\ secara\ salah}{Banyaknya\ data\ prediksi}$$

Dalam *cross-validation*, dikenal validasi *leave-one-out* (LOO). Dalam LOO, data dibagi ke dalam 2 subset, subset 1 berisi N-1 data untuk *training* dan satu data sisanya untuk *testing* (Santosa, 2007).

2 *Grid Search*

Menurut Wang (2012) dalam Yao, *et al.* (2014), algoritma *grid-search* membagi jangkauan pencarian parameter yang akan dioptimalkan ke dalam *grid* dan melintasi semua titik *grid* untuk mendapatkan nilai optimal. Di dalam Hsu, *et al.*(2004), sebuah algoritma *grid search* harus dipandu oleh beberapa metrik kinerja, biasanya diukur dengan *cross-validation* pada data *training*. Disarankan untuk mencoba menggunakan variasi pasangan parameter pada *hyperplane* SVM dan pasangan parameter yang menghasilkan akurasi terbaik yang didapatkan dari uji *cross-validation* yang akan diambil. Parameter-parameter yang diseleksi digunakan pada data *training* sehingga didapatkan model klasifikasi. Setelah itu, model klasifikasi tersebut digunakan untuk mengklasifikasikan data *testing* untuk mendapatkan generalisasi akurasi (Srivastava and Bhambhu, 2005).

2.8. Pengukuran Kinerja Klasifikasi

Menurut Prasetyo (2012), matriks konfusi merupakan tabel pencatat hasil klasifikasi. Misalkan elemen matriks konfusi untuk data klasifikasi dengan dua kelas dinyatakan dengan f_{ij}, maka setiap sel f_{ij} menyatakan jumlah record/data yang sebenarnya masuk dalam kelas i, tetapi hasil prediksinya mengklasifikasikan data tersebut pada kelas j. Tabel 1 menunjukkan gambaran mengenai matriks konfusi.

Tabel 1 Matriks Konfusi

Hasil Observasi (*actual class*)	Hasil Prediksi (*predicted class*)	
	Kelas 1	Kelas 2
Kelas 1	f_{11}	f_{12}
Kelas 2	f_{21}	f_{22}

Akurasi hasil klasifikasi yang dapat dihitung dengan formula, sebagai berikut:

$$Akurasi = \frac{Banyaknya\ data\ yang\ diprediksi\ dengan\ benar}{Banyaknya\ data\ prediksi} = \frac{f_{11} + f_{22}}{f_{11} + f_{12} + f_{21} + f_{22}}$$

3. METODOLOGI PENELITIAN

Dalam penelitian ini data yang digunakan adalah data sekunder yang diperoleh dari Badan Pusat Statistik (BPS) Kabupaten Semarang melalui web site http://semarangkab.bps.go.id/ dan Badan Pusat Statistik Republik Indonesia melalui web site http://www.bps.go.id/.

Pada penelitian ini, akan terdapat sepuluh variabel independen (x), yang terdiri dari kepadatan penduduk (x_1), persentase rumah tangga pertanian (x_2), ketersediaan sekolah Taman Kanak-Kanak (TK) (x_3). ketersediaan Sekolah Menengah Pertama (SMP) (x_4), ketersediaan Sekolah Menengah Atas (SMA) (x_5), ketersediaan pasar (x_6), ketersediaan minimarket (x_7), ketersediaan rumah sakit (x_8), ketersediaan hotel (x_9), dan persentase rumah tangga pengguna listrik (x_{10}). Data akan dikelompokkan ke dalam dua kelas yang merupakan variabel dependen (y), yaitu wilayah dengan kelas desa-perdesaan = -1 dan kelas desa-perkotaan = 1.

Langkah-langkah Analisis

1. Mempersiapkan data klasifikasi wilayah desa-perdesaan dan desa-perkotaan Kabupaten Semarang yang terdiri dari variabel independen dan variabel dependen.
2. Membagi data tersebut menjadi data *training* dan data *testing* dengan persentase proporsi tertentu secara acak dengan bantuan paket program R 2.15.1.
3. Melakukan klasifikasi data klasifikasi wilayah desa-perdesaan dan desa-perkotaan Kabupaten Semarang dengan metode SVM (*Support Vector Machine*) dengan tahapan sebagai berikut:
   a. Menentukan fungsi kernel, nilai-nilai parameter kernel dan parameter *cost* untuk optimasi *hyperplane* pada data *training*.
   b. Memilih nilai parameter terbaik untuk klasifikasi data pada penelitian ini menggunakan metode *grid-search* pada tiap *hyperplane* dengan fungsi kernel yang berbeda dengan bantuan paket program R 2.15.1, sebagai berikut:
      1) Membagi data *training* sebanyak baris data.
      2) Gunakan tiap bagian sebagai data *testing* dan bagian lainnya sebagai data *training*.
      3) Optimasi parameter α dan b untuk setiap fungsi pemisah dengan metode SMO.
      4) Menggunakan *hyperplane-hyperplane* tersebut untuk klasifikasi.
      5) Hitung error tiap hasil klasifikasi.
      6) Hitung rata-rata error hasil klasifikasi untuk setiap *hyperplane*.
      7) Menentukan *hyperplane* terbaik.
   c. Menggunakan *hyperplane* dengan parameter terbaik yang diperoleh untuk setiap fungsi kernel pada klasifikasi data testing.
   d. Evaluasi hasil klasifikasi data desa-perdesaan dan desa-perkotaan wilayah Kabupaten Semarang dengan metode SVM pada data testing untuk mengukur ketepatan klasifikasi dengan nilai akurasi.

4. ANALISIS DAN PEMBAHASAN

4.1. Penentuan Fungsi Kernel dan Parameter Terbaik Untuk *Hyperplane*

Pada penelitian ini dicobakan fungsi kernel linier, *polynomial*, dan *Radial Basis Function* (RBF) pada fungsi pemisah (*hyperplane*) SVM. Parameter terbaik pada setiap fungsi kernel ditentukan dengan mencobakan beberapa nilai pada rentang tertentu untuk membangun *hyperplane*. Untuk menentukan fungsi kernel beserta parameter yang terbaik pada permasalahan ini digunakan metode *grid search* yang dipadukan dengan metode *leave-one-out cross-validation* untuk menghitung nilai error klasifikasi untuk setiap nilai parameter yang dicobakan pada data *training*. Fungsi kernel beserta nilai parameter yang terbaik untuk *hyperplane* ditentukan dengan nilai error terkecil. Dari seleksi parameter yang dilakukan, didapatkan parameter terbaik untuk *hyperplane* dengan fungsi kernel linier adalah $C = 10^1$, fungsi kernel *polynomial* d=2 dan $C = 10^{-2}$, dan fungsi kernel RBF $\gamma = 2^{-5}$ dan $C = 10^2$.

Tabel 2 Error Hasil Klasifikasi SVM Untuk Penentuan Parameter Terbaik Dengan Metode *Grid Search*

Fungsi kernel	Parameter *hyperplane*		Error Klasifikasi
Linier		$C = 10^{-1}$	0,1515152
		$C = 10^{0}$	0,1333333
		$C = 10^{1}$	**0,1212121**
		$C = 10^{2}$	0,1212121
		$C = 10^{3}$	0,1272727
Polynomial	d = 2	$C = 10^{-5}$	0,2666667
		$C = 10^{-4}$	0,169697
		$C = 10^{-3}$	0,1575758
		$C = 10^{-2}$	**0,1030303**
		$C = 10^{-1}$	0,1151515
		$C = 10^{0}$	0,1393939
		$C = 10^{1}$	0,1393939
		$C = 10^{2}$	0,1818182
	d = 3	$C = 10^{-5}$	0,1575758
		$C = 10^{-4}$	0,1272727
		$C = 10^{-3}$	0,1151515
		$C = 10^{-2}$	0,1515152
		$C = 10^{-1}$	0,1636364
		$C = 10^{0}$	0,1636364
		$C = 10^{1}$	0,1757576
		$C = 10^{2}$	0,1757576
	d = 4	$C = 10^{-5}$	0,1090909
		$C = 10^{-4}$	0,1393939
		$C = 10^{-3}$	0,1454545
		$C = 10^{-2}$	0,1515152
		$C = 10^{-1}$	0,1818182
		$C = 10^{0}$	0,1818182
		$C = 10^{1}$	0,1818182
		$C = 10^{2}$	0,1818182
	d = 5	$C = 10^{-5}$	0,1515152
		$C = 10^{-4}$	0,1515152
		$C = 10^{-3}$	0,1757576
		$C = 10^{-2}$	0,1818182
		$C = 10^{-1}$	0,1818182
		$C = 10^{0}$	0,1818182
		$C = 10^{1}$	0,1818182
		$C = 10^{2}$	0,1818182

Fungsi kernel	Parameter *hyperplane*		Error Klasifikasi
RBF	$\gamma = 2^{-8}$	$C = 10^{-1}$	0,2727273
		$C = 10^{0}$	0,169697
		$C = 10^{1}$	0,1515152
		$C = 10^{2}$	0,1393939
		$C = 10^{3}$	0,1393939
		$C = 10^{4}$	0,1575758
		$C = 10^{5}$	0,1454545
	$\gamma = 2^{-7}$	$C = 10^{-1}$	0,2727273
		$C = 10^{0}$	0,1575758
		$C = 10^{1}$	0,1454545
		$C = 10^{2}$	0,1393939
		$C = 10^{3}$	0,1393939
		$C = 10^{4}$	0,1575758
		$C = 10^{5}$	0,1757576
	$\gamma = 2^{-6}$	$C = 10^{-1}$	0,2363636
		$C = 10^{0}$	0,1636364
		$C = 10^{1}$	0,1333333
		$C = 10^{2}$	0,1333333
		$C = 10^{3}$	0,1333333
		$C = 10^{4}$	0,1454545
		$C = 10^{5}$	0,1515152
	$\gamma = 2^{-5}$	$C = 10^{-1}$	0,169697
		$C = 10^{0}$	0,1575758
		$C = 10^{1}$	0,1333333
		$C = 10^{2}$	**0,1272727**
		$C = 10^{3}$	0,1393939
		$C = 10^{4}$	0,1636364
		$C = 10^{5}$	0,1575758

4.2. Perbandingan Akurasi Hasil Klasifikasi Dengan Metode SVM

Parameter terbaik yang diperoleh dari penerapan metode *grid search* pada data *training* selanjutnya diterapkan pada data *testing* sebagai evaluasi terhadap hasil klasifikasi yang ditunjukkan dengan nilai akurasi. Perbandingan akurasi hasil klasifikasi dengan SVM dengan parameter terbaik yang telah ditentukan sebelumnya pada setiap fungsi kernel dapat ditunjukkan pada Tabel 3.

Tabel 3 Akurasi Hasil Klasifikasi Dengan Metode SVM

Fungsi Kernel	Akurasi Hasil Klasifikasi
Linier	0,8571429
Polynomial	0,8714286
RBF	0,9000000

Akurasi tertinggi didapatkan dengan menggunakan *hyperplane* SVM dengan fungsi kernel *Radial Basis Function* (RBF) dengan parameter C=100 dan $\gamma = 2^{-5}$, yaitu sebesar 90%.

5. KESIMPULAN

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan bahwa klasifikasi status wilayah desa-perdesaan dan desa-perkotaan di Kabupaten Semarang menggunakan metode *Support Vector Machine* (SVM) sebanyak 63 wilayah dari 70 wilayah yang diuji dapat diklasifikasikan secara benar sesuai dengan kelas asli. Akurasi klasifikasi yang didapat yaitu sebesar 90 % dengan *hyperplane* terbaik yang menggunakan parameter *cost* (C) = 100 dan fungsi kernel *Radial Basis Function* (RBF) dengan parameter $\gamma = 2^{-5}$.

DAFTAR PUSTAKA

[BPS] Badan Pusat Statistik.2010.*Peraturan Kepala Badan Pusat Statistik Nomor 37 Tahun 2010 Tentang Klasifikasi Perkotaan Dan Pedesaan Di Indonesia*.[online].[diakses pada 28 Januari 2014].Tersedia pada:*www.bps.go.id/download_file/ MFD/MFD_2010 _Buku_2.pdf*

Gunn, S. R. 1998. *Support Vector Machines for Classification and Regression*. Technical Report. University of Southampton.

Haykin, S.1999.*Neural Network : A Comprehensive Foundation*. New Jersey:Prentice Hall

Hsu, C.W.,*et al*.2004.*A Practical Guide to Support Vektor Classification*. Department of Computer Science and Information Engineering, National Taiwan University.

Karatzoglou, A, et al.2006.*Support Vector Machines in R*. Journal of Statistical Software Volume 15, Issue 9.Tersedia pada: http://www.jstatsoft.org/

Keerthi, S.S. And Gilbert, E.G.2002.*Convergence of a Generalized SMO Algorithm for SVM Classifier Design*.Kluwer Academic Publishers.Machine Learning, 46, 351–360

Leidiyana, H.2013. *Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bemotor*.Jurnal Penelitian Ilmu Komputer, System Embedded & Logic 1(1) : 65-76

Nuha,M.U,dkk.2012.*Pengembangan Perangkat Lunak Prediktor Kebangkrutan Menggunakan Bagging Nearest Neighbor Support Vector Machine.*Jurnal Teknik Pomits Vol. 1, No. 1, (2012) 1-6.Surabaya

Platt J.C.1998. *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*. In B Scholkopf, CJC Burges, AJ Smola (eds.), "Advances in Kernel Methods – Support Vektor Learning," pp. 185–208. MIT Press, Cambridge, MA.

Prasetyo, E.2012.*Data Mining Konsep dan Aplikasi Menggunakna Matlab*. Yogyakarta:Andi

Santosa, B.2007.*Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis.*Yogyakarta:Graha Ilmu

Srivastava, D.K. and Bhambhu, L.2005.*Data Classification Using Support Vector Machine.Journal Of Theoretical And Applied Information Technology*.India

Tarigan, A.2003.*Rural - Urban Economic Lingkages*.[online]. [diunduh pada 17 Februari 2014].Tersedia pada:http://www.bappenas.go.id/index.php/download file/view/10656/2372/

Vapnik ,V.1995.The Nature of Statistical Learning Theory. Springer:Verlag

Yao, Y, *et al*.2014.*An Improved Grid Search Algorithm and Its Application in PCA and SVM Based Face Recognition*.Journal of Computational Information Systems 10: 3 (2014) 1219–1229.China