MAKARA, SAINS, VOL. 8, NO. 3, DESEMBER 2004: 89-95

KAJIAN KEMAMPUAN GENERALISASI *SUPPORT VECTOR MACHINE* DALAM PENGENALAN JENIS *SPLICE SITES* PADA BARISAN *DNA*

Djati Kerami dan Hendri Murfi

Departemen Matematika, FMIPA, Universitas Indonesia, Depok 16424, Indonesia;
Kelompok Kajian Kuantitatif, FMIPA, Universitas Indonesia, Depok 16424, Indonesia

E-mail: djatikr@makara.cso.ui.ac.id , hendri@makara.cso.ui.ac.id

Abstrak

Beberapa tahun terakhir ini, *Support Vector Machine* (*SVM*) telah populer digunakan sebagai model *machine learning*. Hal ini terutama karena *SVM* dapat dianalisis secara teoritis, dan secara bersamaan dianggap memberikan kinerja yang lebih baik daripada model *machine learning* yang biasa digunakan sebelumnya. Pada makalah ini dibahas pendekatan matematis model *SVM* dalam memecahkan masalah pengenalan pola. Selanjutnya dibahas pula penggunaan model tersebut berupa kajian awal penentuan jenis *splice site* pada suatu barisan *DNA* terutama dari segi kemampuan generalisasi atau tingkat keakuratannya. Hasil yang diperoleh menunjukkan bahwa kemampuan generalisasi *SVM* sangat baik yaitu sekitar 95.4 %.

Abstract

Study on Generalization Capability of Support Vector Machine in Splice Site Type Recognition of DNA Sequence. Recently, support vector machine has become a popular model as machine learning. A particular advantage of SVM over other machine learning is that it can be analyzed theoretically and at same time can achieve a good performance when applied to real problems. This paper will describe analytically the using of SVM to solve pattern recognition problem with a preliminary case study in determining the type of splice site on the DNA sequence, particularity on the generalization capability. The result obtained show that SVM has a good generalization capability of around 95.4 %.

Keywords: *Support vector machine, generalization test, pattern recognition*, *splice sites*, *DNA*

1. Pendahuluan

Support Vector Machine (*SVM*) dikenal sebagai teknik pembelajaran mesin (*machine learning*) paling mutakhir setelah pembelajaran mesin sebelumnya yang dikenal sebagai *Neural Network* (*NN*). Baik *SVM* maupun *NN* tersebut telah berhasil digunakan dalam pengenalan pola. Pembelajaran dilakukan dengan menggunakan pasangan data input dan data output berupa sasaran yang diinginkan. Pembelajaran dengan cara ini disebut dengan pembelajaran terarah (*supervised learning*). Dengan pembelajaran terarah ini akan diperoleh fungsi yang menggambarkan bentuk ketergantungan input dan outputnya. Selanjutnya, diharapkan fungsi yang diperoleh mempunyai kemampuan generalisasi yang baik, dalam arti bahwa fungsi tersebut dapat digunakan untuk data input di luar data pembelajaran.

Sebelum digunakan teknik *SVM*, teknik *NN* (khususnya berdasarkan *backpropagation neural network*) telah berhasil digunakan pada masalah pengenalan pola. Akan tetapi, teknik ini memiliki beberapa kelemahan, antara lain optimisasi yang digunakan tidak selalu mencapai nilai minimal global dari kurva fungsi galatnya [1]. Di samping itu, kadang-kadang terjadi fenomena *over-learning* yang sering dapat menurunkan kemampuan generalisasinya [2]. Berbagai upaya pengembangan dilakukan untuk mengatasi kelemahan tersebut, antara lain pendekatan menggunakan

teknik *radial basis function networks* oleh Moody *et al.* [1], *projection generalizing neural networks* oleh Ogawa [2] dan yang terakhir adalah *SVM* seperti yang dikemukakan oleh Burges [3], Hearst *et al.* [4], Cristianini *et al.* [5], dan Shawe-Taylor *et al.* [6], yang semuanya memperoleh ide dari Vapnik [7].

Pada beberapa tahun terakhir ini, *SVM* mulai menjadi model yang favorit sebagai suatu pembelajaran mesin. Hal ini terutama karena terhadap *SVM* dapat dilakukan secara analitis (analisis secara matematis) dan disamping itu dapat memberikan kemampuan generalisasi yang baik pada penerapannya dibanding model *NN* [4].

Pada tulisan ini akan dibahas tentang *SVM* serta penggunaannya dalam pemecahan masalah pengenalan pola. Di sini dilakukan eksperimen (simulasi komputer) pada studi kasus berupa studi awal dalam pengenalan jenis *splice site* pada suatu barisan *DNA*. Dalam hal ini hanya akan ditentukan apakah suatu barisan *DNA* merupakan *splice site* berjenis *donor* atau bukan. Tinjauan yang hampir sama dilakukan oleh Yamamura dan Gutoh [8] yang menunjukkan bahwa penggunaan *SVM* memberikan kemampuan generalisasi (tingkat akurasi) sekitar 94,49 % lebih baik dibandingkan dengan penggunaan model Markov (sekitar 91,29 %) maupun model lainnya.

Dalam penelitian yang telah dikerjakan, dilakukan pengelompokan data ke dalam tiga kelompok, yaitu data pembelajaran, data validasi, dan data uji yang digunakan secara berturutan. Setiap kelompok mempunyai fungsi tertentu dalam pembentukan *SVM*. Dari simulasi komputer yang dilakukan dengan menggunakan teknik pengelompokan data tersebut, kemampuan generalisasi yang diberikan sedikit lebih baik (sekitar 95,42 %).

2. Metode Penelitian

Penelitian dilakukan melalui simulasi komputer menggunakan model *SVM* linear dan *SVM* non linear. Berikut ini akan diuraikan secara ringkas tentang ide dasar *SVM* [3-7]: Misalkan diberikan himpunan X= $\{x_1, x_2, \ldots, x_m\}$, dengan $x_i \in R^n$, i=1, …. , m. Telah diketahui bahwa X berpola tertentu, yaitu apabila x_k termasuk dalam suatu klas maka x_k diberikan label (merupakan target) y_k =+1, jika tidak diberi label y_k =-1. Dengan demikian data yang diberikan berupa pasangan $(x_1, y_1), \ldots\ldots, (x_m, y_m) \in X \times \{+1,-1\}$. Dalam masalah pembelajaran, kumpulan pasangan tersebut tersebut merupakan data pembelajaran bagi *SVM*. Berbekal pengalaman pembelajaran menggunakan data pembelajaran tersebut, *SVM* harus mampu menentukan pola (generalisasi) dari $x \notin X$.

Masalah dasar dari *SVM* adalah menentukan suatu *hyperplane* $\langle w, x\rangle + b = 0$ memisahkan data x_i yang terdiri dari dua kelas, yaitu $y_i = \{+1, -1\}$, dengan margin maksimal. Margin disini merupakan jarak antara *hyperplane* ke masing-masing kelas data (Gambar 1).
Selanjutnya, *hyperplane* ini akan menjadi fungsi keputusan $f(x)$ untuk masalah klasifikasi dua kelas di atas.

$$f(x) = \text{sign}(\langle w, x\rangle + b) \quad (1)$$

dengan $f(x) = 1$ jika $\langle w, x\rangle + b \geq 0$ dan
$f(x) = -1$ jika $\langle w, x\rangle + b < 0$.

Hyperplane tersebut dapat dibuat secara tunggal sesuai dengan w dan b yang akan diperoleh. Selanjutnya x_i yang berupa subhimpunan data pembelajaran yang terletak pada margin disebut dengan *support vector*.

Menurut teorema Vapnik [7], pemilihan margin maksimum ini akan memberikan kemampuan generalisasi yang paling baik

Untuk menentukan persamaan *hyperplane* $\langle w, x\rangle + b = 0$ harus diketahui lebih dahulu nilai w dan b. Di sini nilai w merupakan nilai yang berkaitan dengan margin. Untuk memperolehnya, digunakan bantuan *hyperplane* kanonik yaitu:

$$\langle w, x^+\rangle + b = +1 \quad (2)$$
$$\langle w, x^-\rangle + b = -1 \quad (3)$$

Dalam hal ini, x^+ adalah data yang terletak pada kelas $y = +1$ dan terdekat ke *hyperplane*, dan x^- adalah data yang terletak pada kelas $y = -1$ dan terdekat ke *hyperplane* (Gambar 2).

Lebar margin γ adalah jarak x^+ ke *hyperplane* atau jarak x^- ke *hyperplane*,

MAKARA, SAINS, VOL. 8, NO. 3, DESEMBER 2004: 89-95

Jadi, memaksimalkan margin adalah ekivalen dengan meminimalkan $\|w\|$ dengan syarat:

$$\begin{aligned} &\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq +1 \text{ jika } y_i = +1 \\ &\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b < -1 \text{ jika } y_i = -1 \end{aligned} \quad (5)$$

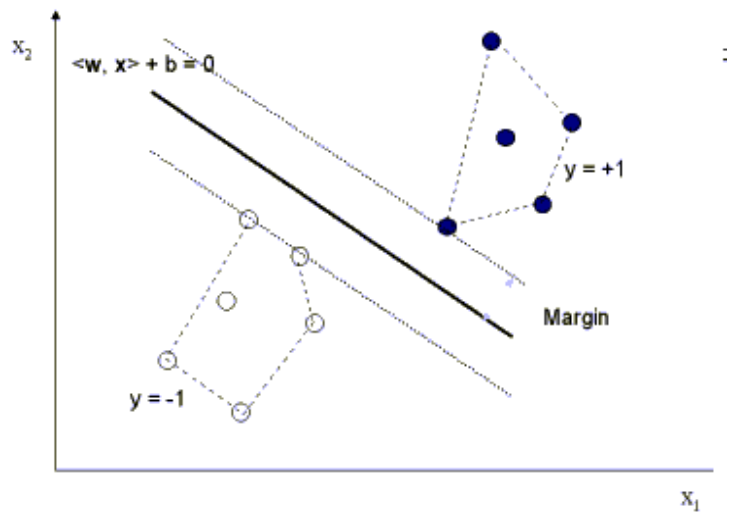
Dua syarat di atas dapat dijadikan satu syarat, yaitu

$$y_i (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \ \forall i \quad (6)$$

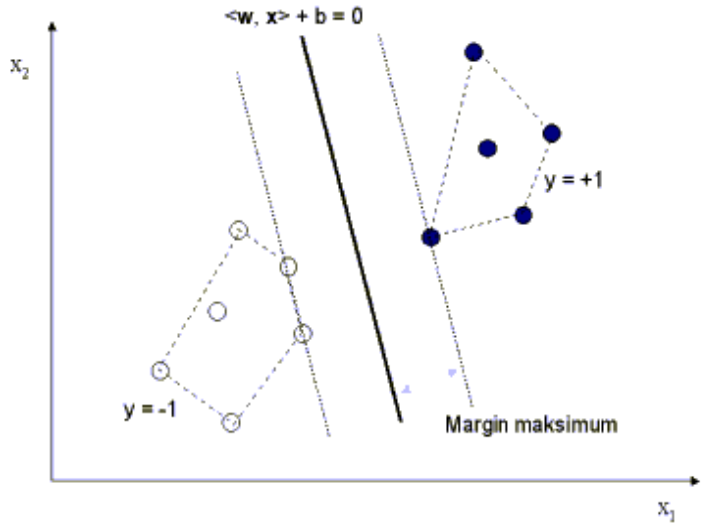
Dengan demikian masalah pembelajaran *SVM* menjadi masalah Pemrograman Kuadratik (PK) yang sajikan dalam beberapa proposisi seperti berikut ini :

Proposisi 1. Diberikan data pembelajaran yang *linearly separable* berikut:

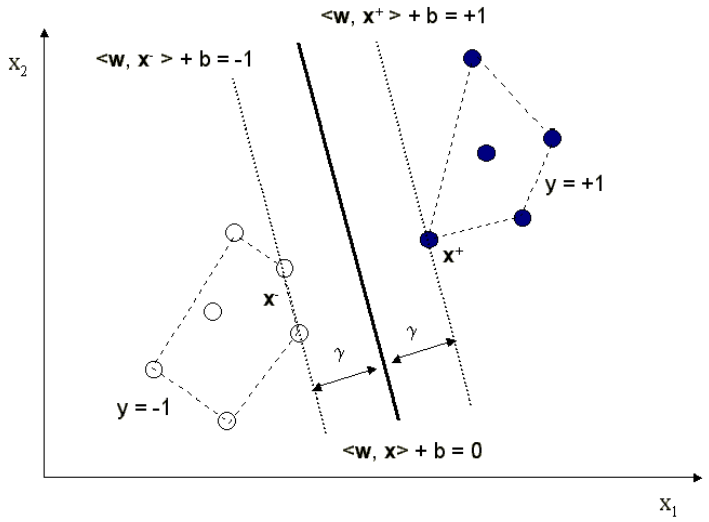
$S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$



MAKARA, SAINS, VOL. 8, NO. 3, DESEMBER 2004: 89-95



Gambar 1. Margin dari hyperplane



Gambar 2. *Hyperplane* dan *hyperplane* kanonik

Hyperplane dengan w dan b yang merupakan penyelesaian dari PK berikut ini:

$$\text{Min.} \quad \langle w,w \rangle \qquad (7)$$

dengan kendala

$$y_i (\langle w, x_i \rangle + b) \geq 1, \; i = 1, \ldots, m$$

adalah merupakan *hyperplane* dengan margin maksimum, yaitu $\gamma = 1/\|w\|$.

Bentuk primal PK persamaan (7) dapat dibawa ke dalam bentuk fungsi *Langrange*, sebagai berikut:

(8)

dengan $\alpha_i \geq 0$ adalah pengali Lagrange.

Formulasi bentuk dual dari bentuk (7) diperoleh dengan menggunakan kondisi Karush-Kuhn-Tucker, melalui derivatif L terhadap w dan b, yaitu:

MAKARA, SAINS, VOL. 8, NO. 3, DESEMBER 2004: 89-95

(9)

(10)

Dari persamaan (9) dan persamaan (10) diperoleh

(11)

(12)

Substitusi persamaan (11) dan persamaan (12) ke dalam fungsi *Lagrange* persamaan (8), memberikan :

(13)

Selanjutnya, bentuk dual dari PK pada Proposisi 1 adalah Proposisi 2. Diberikan data pembelajaran yang *linearly separable* berikut ini

$S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$

Misal α^* adalah penyelesaian dari PK berikut

max.

dengan kendala (14)

Maka vektor bobot $w^* =$ merupakan bobot *hyperplane* dengan margin maksimal, i.e $\gamma = 1/\|w^*\|$

Nilai bias b dapat dicari dari kendala PK (7) pada Proposisi 1. Sedangkan b^* merupakan b optimal menentukan penggeseran *hyperplan* ke kiri (data maksimal pada kelas -1) dan ke kanan (data minimal pada kelas +1) yang paling jauh.

(15)

Oleh karena fungsi tujuan dan kendala dari PK (14) adalah berupa fungsi yang konveks, maka penyelesaian PK tersebut merupakan penyelesaian optimal global.

Metode pembelajaran pada Proposisi 2 hanya berlaku jika data pembelajarannya bersifat *linearly separable*. Untuk data yang *non linearly separable*, metode di atas akan gagal diterapkan. Untuk mengatasinya, digunakan teknik tambahan yang disebut metode Kernel. Ide dasar dari metode ini adalah: (i) Memetakan data yang *non linearly separable* ke ruang dimensi lebih tinggi yang disebut ruang fitur, sehingga menjadi *linearly separable*. Di sini, $K(x',x) = \langle \phi(x_i), \phi(x) \rangle$, dengan ϕ pemetaan dari X ke ruang fitur F dan K merupakan fungsi kernel. (ii) Mencari *hyperplane* dengan margin maksimum pada ruang fitur tersebut (Gambar 3). Selanjutnya, *SVM* yang dibentuk dengan menggunakan metode ini dikenal dengan nama *SVM* non-linear.

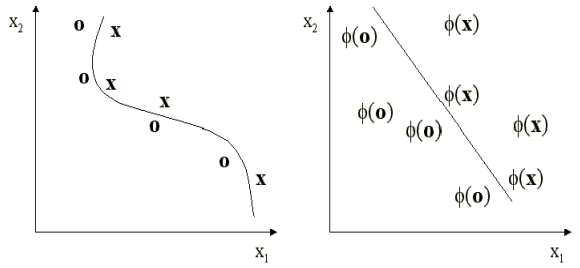
MAKARA, SAINS, VOL. 8, NO. 3, DESEMBER 2004: 89-95

Berdasarkan nilai bobot w^* yang diperoleh dari Proposisi 2, fungsi keputusan *f* dapat dihitung sebagai perkalian dalam dari data pembelajaran dan data uji, yaitu:

(16)

Jika fungsi yang memetakan data ke ruang fitur kita sebut ϕ, maka fungsi keputusan $f(x)$ pada persamaan (16) dapat ditulis sebagai

(17)



Gambar 3. Pemetaan ke ruang fitur oleh fungsi ϕ

Selanjutnya, dengan bantuan teorema Mercer [3-6] kita dapat menghitung nilai perkalian dalam $<\phi(x_i),\phi(x)>$ secara langsung tanpa perlu mengetahui ϕ secara eksplisit.

Dengan demikian, fungsi keputusan persamaan (17) dapat juga ditulis sebagai:

(18)

Dewasa ini, terdapat banyak fungsi Kernel yang sudah dikembangkan. Yang umum digunakan pada *SVM* adalah fungsi polinomial, yaitu:

$$K(x', x) = <x_i, x>^d,\ d \in Z^+ \qquad (19)$$

Masalah utama dari metode *hyperplane* dengan margin maksimal seperti yang dijelaskan sebelumnya adalah selalu menghasilkan *hyperplane* yang sempurna, dengan anggapan tidak adanya galat data pembelajaran. Untuk data pembelajaran yang mengandung gangguan (*noise*), yang umum terjadi pada data dari masalah praktis, teknik ini akan mengalami masalah. Selanjutnya, metode tersebut juga mengalami masalah jika data pembelajaran masih *non-linear separable* pada ruang fitur. Untuk mengatasi masalah di atas digunakan teknik tambahan yang toleran terhadap gangguan dan *outlier*, serta memberikan perhatian yang lebih pada data yang terdekat dengan *hyperplane* pemisah antara dua kelas yang ada. Teknik ini selanjutnya dikenal dengan nama margin lunak (*soft margin*), sementara teknik sebelumnya dikenal dengan nama margin kokoh (*hard margin*) [5-7].

Pada teknik margin lunak, diperkenalkan variabel *slack* (ξ_i), yaitu variabel yang merupakan galat dari masing-masing data pembelajaran, yang memungkinkan kendala margin diabaikan. Selanjutnya, metode pembelajarannya menjadi masalah mencari *hyperplane* optimal yang memaksimalkan margin dan meminimalkan galat data pembelajaran. Teknik ini dikenal dengan *Structural Risk Minimization* (SRM), yang berbeda dengan teknik *Empirical Risk Minimization* (ERM) yang hanya meminimalkan galat data pembelajaran tanpa memperhatikan aspek generalisasi [6].

Proposisi 3. diberikan data pembelajaran berikut:

$$S = ((x_1, y_1), \ldots, (x_l, y_m))$$

Hyperplane dengan margin lunak adalah *hyperplane* dengan *w* dan *b* yang berupa penyelesaian dari PK berikut ini:

Min. $\langle w, w\rangle + C$ (20)

dengan kendala $y_i (\langle w, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i$, $i = 1, \ldots, m$

$\xi_i \geq 0$, $i = 1, \ldots, m$

Bentuk dual dari PK (20) dapat dibangun dengan menggunakan prosedur seperti pada persamaan (8) – (13).

Proposisi 4. Diberikan data pembelajaran berikut ini:

$S = ((x_1, y_1), \ldots, (x_m, y_m))$

Dengan menggunakan ruang fitur yang didefinisikan secara implisit oleh fungsi kernel $K(x', x)$, dan misal α^* adalah penyelesaian dari PK berikut:

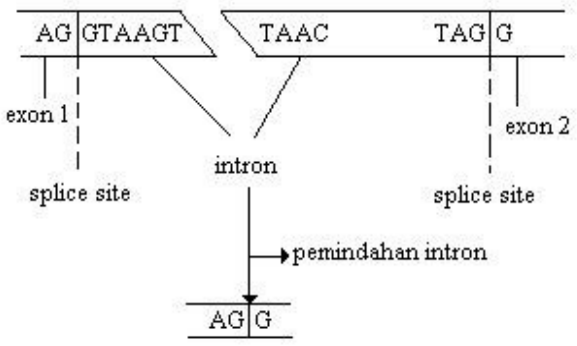
Max. dengan kendala (21)

Misal , dengan b^* dipilih sedemikian sehingga $y_i f(\boldsymbol{x}_i) = 1$ untuk setiap i dengan $C > \alpha^*_i > 0$. Maka fungsi keputusan yang didefinisikan oleh $sign(f(\boldsymbol{x}))$ adalah ekivalen dengan *hyperplane* pada ruang fitur yang secara implisit didefinisikan oleh $K(x', x)$, dengan margin $\gamma = (\quad)^{-1/2}$

Dilakukan simulasi komputer berdasarkan pendekatan *SVM* untuk memecahkan masalah pengenalan pola. Dalam melakukan simulasi digunakan perangkat lunak Matlab 6.5.1.

Studi kasus yang digunakan adalah pengenalan jenis *splice site* pada suatu barisan *Deoxyribo Nucleic Acid* (*DNA*). Barisan *DNA* ini tersusun dari nukleotida *Guanine* (*G*), *Adenine* (*A*), *Thymine* (*T*), *Cytosin* (*C*). Dalam simulasi yang telah dilakukan hanya ditentukan apakah suatu barisan *DNA* merupakan *splice site* berjenis donor atau bukan. Penentuan *splice site* ini diperlukan dalam pemindahan *intron-intron* yang akan berakibat bergabungnya *exon-exon* (donor). *Exon* hasil penggabungan ini diperlukan dalam proses pembentukan protein.

Dalam menggunakan model *SVM*, data yang digunakan dipilah menjadi data pembelajaran, data validasi, dan data uji. Data pembelajaran digunakan untuk membentuk *SVM*, sementara nilai parameter bebasnya dipilih dari nilai parameter yang membuat galat dari data validasi bernilai minimal. Selanjutnya, *SVM* yang dihasilkan akan digunakan untuk memprediksi data uji.



Gambar 4. *Splice sites* dan pemindahan *intron*

Sedangkan data yang digunakan merupakan barisan *DNA* yang terdapat dalam basis data *Splice-juction Gene Sequences* [9]. Setiap barisan memiliki panjang 60 *base-pair* (*bp*). Selanjutnya, masing-masing setiap nukleotida disajikan dalam 4 digit bilangan biner (*bit*),
yaitu *A*: 1000, *T*: 0100, *G*: 0010 dan *C*: 0001. Dengan demikian, dimensi data input adalah 60 x 4 = 240 *bit*. Sementara outputnya berdimensi satu, yaitu bernilai 1 jika merupakan *donor site* dan –1 jika bukan. Nilai parameter bebas yang ada pada model *SVM*, yaitu (i) nilai C pada (21), yaitu nilai *trade-off* antara lebar margin dan galat data estimasi, dan (ii) parameter pada kernel, yaitu *d* pada fungsi polinomial atau simpangan σ pada fungsi basis radial, diidentifikasi dengan menggunakan teknik *cross-validation*. Pada teknik ini, data yang ada dibagi menjadi tiga kelompok, yaitu data estimasi , data validasi, dan data uji. Pada simulasi yang telah dilakukan, digunakan data sebanyak 750 dan didistribusikan secara acak sebanyak 250 data (tetap) untuk data estimasi, dan sisanya sebanyak 500 data digunakan untuk kelompok data validasi dan kelompok data uji. Dalam simulasi komputer, dilakukan 5(lima) kali percobaan, yaitu menggunakan banyaknya data estimasi yang banyaknya tetap (250) dan pasangan data validasi dan data uji atau (*V*,*U*) yang berubah, sebagai berikut: (150, 350), (200, 300), (250, 250), (300, 200), dan (350,150).

Uji generalisasi dilakukan dengan menggunakan data uji
sebanyak yang telah dipersiapkan. Hasil pengujian berupa nilai akurasi (dalam %), menyatakan berapa banyaknya data input yang menghasilkan *splice site donor* yang benar. Angka ini menentukan kemampuan *SVM* dalam generalisasi.

3. Hasil dan Pembahasan

Dari simulasi yang dilakukan diperoleh hasil akurasi (kemampuan generalisasi) dari *SVM* linear maupun non linear (menggunakan fungsi polinomial derajat dua sebagai kernelnya).

Tabel 1. Uji akurasi pada generalisasi

# data V	# data U	*SVM* Linear	*SVM* Non Linear
150	350	94,2%	94,2%
200	300	94,9%	94,8%
250	250	96,8%	96,8%
300	200	96,2%	96,1%
350	150	95,1%	95.1%

Dari Tabel 1 terlihat bahwa banyaknya data validasi maupun banyaknya data uji tidak berpengaruh banyak terhadap hasil generalisasinya. Akan tetapi hasilnya akan lebih baik apabila banyaknya data pembelajaran lebih banyak atau sama dengan banyaknya data validasi maupun data uji. Dari hasil simulasi komputer tersebut diperoleh bahwa ternyata *SVM* linear dan *SVM* non-linear memberikan hasil generalisasi yang sama, yaitu dapat memprediksi data uji dengan kemampuan
generalisasi sebesar 95.42 %. Disamping itu, hasil pada Tabel tersebut juga memberikan petunjuk bagi kita bahwa data input barisan *DNA* yang digunakan nampaknya bersifat *linearly separable*.

4. Kesimpulan

Telah dibahas dalam makalah ini tentang penggunaan *SVM* dalam memecahkan masalah pengenalan pola. Studi kasus yang digunakan adalah masalah pengenalan *splice site* pada suatu barisan *DNA*. Dalam hal ini adalah menentukan apakah barisan *DNA* yang diberikan adalah *splice site* jenis *donor* atau bukan. Berdasarkan simulasi komputer yang telah dilakukan, *SVM* dapat memberikan kemampuan generalisasi yang baik yaitu sekitar 95.4 %.

Daftar Acuan

[1] J. Moody, C. Darken, Neural Computation 1 (1989) 281.

[2] H. Ogawa, Proceeding of International Conference on Intelligent Information Processing System , Beijing, RRC, 1992.
[3] C.J.C. Burges, Data Mining and Knowledge Discovery 2 (1998) 955.
[4] M.A. Hearst, B. Schölkopf, S. Dumais. E. Osuna, J. Platt, IEEE Intelligent Systems. 13 (1998) 18.
[5] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning method, Cambridge University Press, New York, 2000.
[6] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, 2004.
[7] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1999.
[8] M. Yamamura, O. Gotoh, Genome Informatics. 14 (2003) 426.
[9] Molecular Biology Data Base, http://www.ics.edu/~mlearn/ Mlsummary. html, 2004.