Prediksi Perkembangan Kondisi Pasien Terapi HIV dengan Menggunakan Representasi ALE-index sebagai Invariant Nucleotida sequence dan Support Vector Machine

Al Azhar[1] Jondri[2] Untari Novia Wisesty[2]

[1,2,3]Fakultas Informatika – Universitas Telkom
Jl. Telekomunikasi, Dayeuhkolot Bandung 40257 Indonesia
[1]alazhar1092@gmail.com, [2]jondri@telkomuniversity.ac.id, [3]untarinw@telkomuniversity.ac.id

Abstrak

Human Immunodeficiency Virus atau disingkat HIV merupakan salah satu jenis virus yang sangat berbahaya. HIV menyerang system immune yang menyebabkan pasien HIV mengalami kegagalan sistem kekebalan tubuh. Dalam beberapa tahun terakhir, inveksi HIV sudah ditangani dengan berbagai terapi. Salah satu terapi paling efektif adalah dengan mengkonsumsi obat antiretroviral yang akan menekan virus HIV agar tidak menduplikasikan diri, ataupun menginfeksi sel darah putih. Namun, virus biasanya akan bermutasi terhadap obat obatan yang diberikan dalam penanganan, sehingga virus kebal terhadap obat yang biasa diberikan di terapi. Untuk itu dibutuhkan suatu sistem prediksi untuk memprediksi kondisi pasien terapi yang akan membaik, agar mempermudah dalam pengambilan keputusan penangan dini pada pasien. Dengan menggunakan 4 parameter yaitu jumlah CD4, Viral Load, PR sequence dan RT sequence, penulis berusaha membangun sistem prediksi perkembangan kondisi pasien terapi HIV. Sistem prediksi ini dibangun dengan salah satu metode klasifikasi machine learning yaitu metode Support Vector Machine (SVM) dan representasi numerik dari urutan nukleotida yaitu ALE-index. Metode ALE-index pada sistem berfungsi untuk mentranslasi parameter RT sequence dan PR sequence yang masih dalam bentuk urutan nukleotida menjadi data numerik agar bisa diinputkan kedalam SVM. Pada metode ALE-index ini juga terdapat beberapa penangan karakter yang bukan merupakan empat unsur utama penyusun urutan nukleotida. Hasil pengujian menunjukkan kombinasi penanganan Random-Delete row dengan menggunakan kernel RBF pada SVM memperoleh akurasi yang lebih tinggi dibandingkan kombinasi penanganan dan parameter lainnya yaitu 77.46%. Dan dengan menggunakan keempat parameter, akurasi yang diperoleh lebih tinggi dibandingkan dengan mengilangkan salah satu fitur.

Kata kunci : **:** *HIV, Support Vector Machine, Nukleotida, ALE-index*

Abstract

Human Immunodeficiency Virus (HIV) is one of many dangerous type of virus. HIV attacks the immune system that causes HIV patients experienced failure of the immune system. In recent years, HIV infection has been treated with various therapies. Currently, the most effective treatment is to take antiretroviral drugs that would suppress the HIV virus in order to not duplicate themselves, or infects white blood cells. However, the virus will mutate to a drug normally prescribed by the treatment, which makes the virus resistant to drugs commonly given in therapy. That requires a prediction system to predict the HIV progression of HIV patient, in order to facilitate the early decision-making in the patient. By using four parameters: CD4 count, viral load, PR and RT sequences sequences, the authors tried to build prediction system of HIV patient's condition. This prediction system built with one of machine learning classification method Support Vector Machine (SVM) and numerical representation of the nucleotides sequence ALE-index. ALE-index method is used to translate parameter RT sequences and PR sequences that still in the form of nucleotide sequences into numeric data to be entered into system. The results showed the combination of Random-Delete row using RBF kernel in the SVM obtain higher accuracy, 77.46% than the other combination of parameters. And by using four parameters, the system obtain higher accuracy compared to using just three parameters.

Keyword : *HIV, Support Vector Machine, Nucleotide, ALE-index*

1. PENDAHULUAN

HIV adalah sebuah virus yang menyerang system immune sehingga menyebabkan pengidap HIV mengalami kegagalan sistem kekebalan tubuh. Dampak dari hal tersebut adalah tubuh pasien menjadi lemah dalam melawan infeksi sehingga rentan terhadap serangan virus memungkinkan infeksi lain seperti kanker dapat berkembang [1]. Berdasarkan Joint United Nations Programme on HIV/AIDS, pada tahun 2009 HIV telah menyebabkan
30 juta kematian diseluruh dunia sejak ditemukan pada tahun 1981 [8].

Dalam beberapa tahun terakhir, inveksi HIV sudah ditangani dengan berbagai terapi. Saat ini terapi yang paling efektif adalah dengan mengkonsumsi obat antiretroviral yang dapat
menekan virus HIV agar tidak menduplikasikan diri, ataupun menginfeksi sel darah putih. Namun, biasanya virus dapat bermutasi terhadap obat obatan yang diberikan, sehingga virus menjadi kebal terhadap obat yang diberikan pada saat terapi. [13]

Penelitian ini bertujuan untuk membangun sebuah sistem yang dapat memprediksi kondisi pasien terapi yang mengalami perkembangan. Dengan prediksi tersebut, akan mempermudah dalam pengambilan keputusan penanganan pada pasien. Parameter yang biasanya digunakan dalam memonitor kondisi pasien adalah Viral Load (VL) dan jumlah CD4. Kemudian pada penelitian kali ini, terdapat tambahan dua parameter selain yang disebutkan sebelumnya yaitu Protease (PR) dan Reverse transcriptase (RT) Sequence [11]. Dengan parameter-parameter tersebut Pasien terapi HIV akan

diprediksi apakah kondisinya akan membaik atau tidak dalam 16 minggu kedepan.

Metode yang digunakan pada penelitian ini adalah metode Support Vector Machine (SVM). SVM adalah salah satu metode klasifikasi yang membagi data berdasarkan kelas-kelas yang sudah ditentukan. Metode ini telah terbukti keefektifannya di berbagai bidang ilmu, salah satunya bidang bioinformatika. Kelebihan metode SVM dibandingkan metode lainnya yaitu kemampuannya dalam menemukan hyperplane terbaik yang memisahkan dua kelas [12]. Namun metode SVM membutuhkan representasi numerik untuk data inputnya, untuk itu dibutuhkan suatu metode yang dapat merubah nucleotide sequence pada fitur kedalam representasi numerik. Oleh karena itu, penulis menggunakan metode representasi ALE- index sebagai representasi numerik dari nucleotide sequence. Metode ALE-index adalah metode yang diperkenalkan oleh Chun Li dan Jun Wang [3] sebagai invariant dari urutan nukleotida.

2. DASAR TEORI

2.1 Urutan Nukleotida

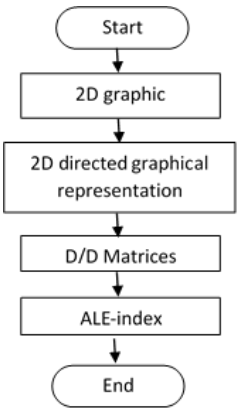
Urutan Nukleotida (Nucleotides sequence) adalah pengurutan susunan nukleotida ke dalam kodon sepanjang peregangan DNA untuk ditranskripsikan. Atau dapat diartikan sebagai metode yang digunakan untuk menentukan urutan basa nukleotida dalam suatu gen dan asam nukleat. DNA (*deoxyribonucleic acid*) merupakan suatu molekul yang terdapat dalam sel makhluk hidup yang memiliki skema biologis yang unik. Barisan ini memiliki kemampuan untuk merepresentasikan informasi.

Urutan nukleotida dihasilkan dari material biologikal DNA melalui suatu metode yang dinamakan DNA sequencing. Urutan nukleotida biasanya direpresentasikan dalam bentuk kumpulan string yang terdiri dari A, G, C, T. Empat karakter pada urutan tersebut merupakan asam nukleat yaitu adenine, guanine, cytosine, and thymine.

2.2 ALE-index

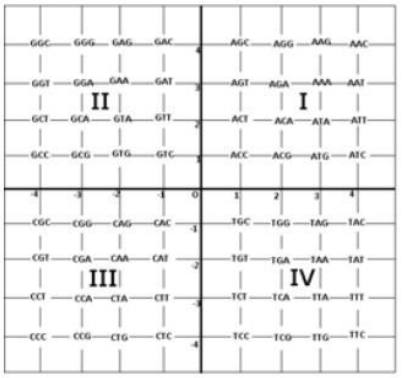
ALE-index merupakan salah satu metode representasi numerik dari urutan nukleotida. Jumlah nukleotida yang sangat panjang pada sequence memberikan kesulitan tersendiri dalam membandingkan urutan nukleotida. Dengan menggunakan metode ALE-index akan diperoleh sebuah invariant dari masing-masing urutan nukleotida. Invarian ini akan mempermudah dalam melihat perbandingan dari 2 urutan nukleotida tersebut.

Nafiseh Jafarzadeh and Ali Iranmanesh [6] dalam papernya menggunakan *D/D matrix based on codon* untuk mendapatkan nilai ALE-index dari sequence. Kodon merupakan urutan tertentu dari tiga nukleotida yang berdekatan untuk menentukan informasi kode genetik untuk sintesis asam amino tertentu.



Gambar 2-1 Alur representasi urutan nukleotida

Setiap kodon yang ada pada urutan nukleotida dipetakan kedalam koordinat grafik 2D yang terdiri dari empat kuadran seperti pada gambar 2-2

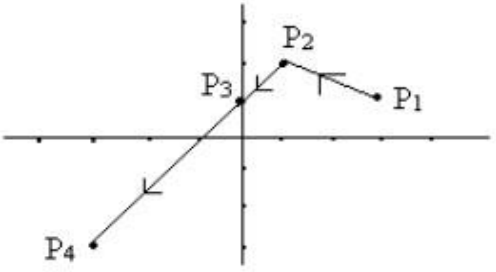


Gambar 2-2 Distribusi kodon kedalam koordinat kartesian

Untuk merepresentasikan kedalam grafik 2D, urutan nukleotida akan dipotong kedalam bentuk kodon-kodon. S=$c_1, c_2, c_3 \ldots c_n$ dimana S merupakan sequence dari nukleotida. Misalkan terdapat urutan nukleotida S= ATGGTGCACCCC. Dari urutan itu kita mendapatkan 4 kodon yaitu c_1=ATG, c_2=GTG, c_3=CAC, c_4=CCC. Dengan mengacu pada Gambar 3 maka didapatkan titik titik koordinat untuk setiap kodon (c_1, c_2)=(3,1) ,(c_2, c_2)=(-2,1) ,(c_3, c_3)=(-1,-1) ,(c_4, c_4)= (-4,-4). Untuk setiap n ϵ {1,2,…. N}, didapat

$$P_n = \left(\sum_{i=1}^{n} c_i, \sum_{i=1}^{n} c_i\right) \quad (1)$$

dimana (c_i c_i) adalah koordinat dari kodon c_i Dari contoh sebelumnya didapat P_1 = (3,1), P_2 = (1,2), P_3 = (0,1), P_4 = (-4,-3). Maka diperoleh grafik berarah dari sequence S.



Gambar 2-3 Grafik berarah dari urutan nukleotida

Selanjutnya dibangun matrix distance/distance [D/D] berdasarkan grafik berarah dari sequence S. Matrix [D/D] didefinisikan dengan:

[□□□□= [□□□□ untuk i≠j,
[□□□□

[□□□□= 0 untuk i=j

Dimana ED adalah Euclidean-distance antara titik □□dan □□pada grafik berarah. ED dihitung dengan persamaan 3 dimana pi = (xi1 , xi2) , pj = (xj1 , xj2)

$$[\square\square_{\square\square} = \sqrt{(\square_{\square} - \square_{\square})^2 + (\square_{\square} - \square_{\square})^2} \quad (2)$$

Dan GD adalah teori graf (topological) distance antara titik Pi dan Pj. GD menggunakan persamaan

$$[\square\square]_{\square\square} = \begin{cases} \square\text{–}\square\square\square\square\square > \square \\ \infty \end{cases} \quad (3)$$

Matrix [D/D] yang merepresentasikan 2D graf berarah adalah *upper triangular matrix* yaitu matrix yang elemen-elemen dibawah diagonal bernilai 0.

Nilai ALE-index suatu sequence ditentukan dengan persamaan

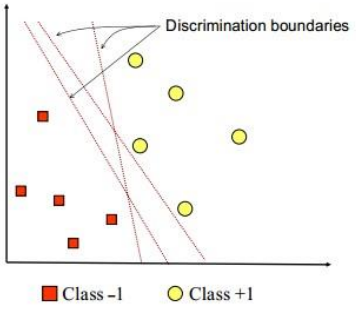
$$\square\square(\square\square) = \frac{1}{\|\square\square\|} \left(\frac{1}{2} \right) + \sqrt{\square\text{–}1} \quad (4)$$

dimana $\|\square\square\|_{\square} = \sum_{\square\square} |\square_{\square}|$ dan $\|\square\square\|_{f} = (\square\square[\square\square]^{T}[\square\square])^{1/2}$

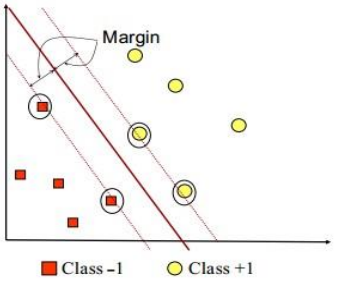
2.3 Support Vector Machine

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition. [4] [5] [7] Metode SVM termasuk dalam kelas supervised learning.

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space. [12]



(a)



(b)

Gambar 2-5b SVM dalam menemukan hyperplane

Pada Gambar 2-5 terlihat beberapa pola anggota kelas dari dua buah kelas yaitu kelas -1 yang disimbolkan dengan kotak warna merah dan kelas +1 yang disimbolkan dengan lingkaran berwarna kuning. Masalah klasifikasi dapat diartikan dengan usaha menemukan garis hyperplane yang dapat memisahkan antara kedua kelas tersebut. [12]

Hyperplane pemisah terbaik antara kedua class dapat ditemukan dengan mengukur margin hyperplane tersebut dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing class. Pattern ini yang disebut sebagai support vector. Garis solid pada gambar 2-5 di sisi kanan menunjukkan hyperplane yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua class, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada SVM. [12]

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik terdekatnya. Hal ini dapat dirumuskan sebagai Quadratic Programming (QP) problem, yaitu mencari titik minimal persamaan (5), dengan memperhatikan constraint persamaan (6).

$$\square\square\square\square(\square) = \frac{T}{2}\|\square\|^2 \quad (5)$$

$$\square(\square_{\square}.\ \square + b) - 1 \geq 0 \quad (6)$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya Lagrange Multiplier sebagaimana ditunjukkan pada persamaan (7).

$$\square\square, \square\square) = \frac{1}{2}\|\square\|^2 - \sum_{\square=1}^{\square} \square (\square((\square\square + \square - 1)) \quad (7)$$

Gambar 2-5a SVM dalam menemukan hyperplane

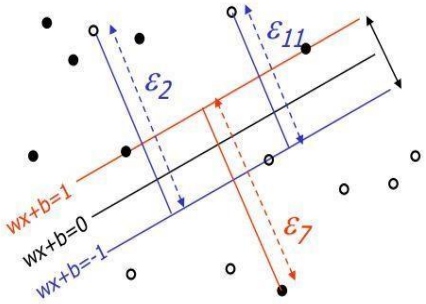
α_i adalah Lagrange multipliers, yang bernilai nol atau positif. Nilai optimal dari persamaan (7) dapat dihitung dengan meminimalkan L terhadap $\vec{w}$ dan b, dan memaksimalkan L terhadap αi. Dengan meperhatikan sifat bahwa pada titik optimal gradient L=0, persamaan (7) dapat dimodifikasi sebagai maksimalisasi yang hanya mengandung αi saja, yaitu

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \quad (8)$$

Subject to $\alpha_i > 0$, $(i = 1,2,\ldots,n)$ $\sum_{i=1}^{n} \alpha_i y_i = 0$

Dari hasil perhitungan di atas didapatlah α_i yang bernilai nol dan positif. Data yang berkorelasi dengan α_i yang positif inilah yang disebut sebagai support vector.

Penjelasan tersebut berlaku jika kedua kelas dapat dipisahkan secara sempurna oleh hyperplane. Akan tetapi, pada umumnya input space tidak dapat dipisahkan secara sempurna yang menyebabkan batasan pada persamaan (8) tidak dapat terpenuhi, dan optimisasi tidak dapat dilakukan. Untuk mengatasi permasalahan ini, maka SVM dirumus ulang dengan menggunakan teknik soft margin.[12]



Gambar 2-6 Soft margin SVM

Dengan menggunakan teknik soft margin, persamaan (6) dimodifikasi dengan memasukan slack variabel $\xi_i(\xi_i \geq 0)$ dan sebuah fungsi penalty sebagai berikut:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (9)$$

dimana $\xi_i \geq 0$ dan dengan demikian persamaan (5) dimodifikasi menjadi

$$\min_{\vec{w}} \Phi(w) = \frac{1}{2}\|w\| + C\sum_{i=1}^{l} \xi_i \quad (10)$$

Parameter C dipilih untuk mengatur trade off antara margin dan error klasifikasi ξ. Nilai C yang besar berarti akan memberikan penalty yang lebih besar terhadap error klasifikasi tersebut.

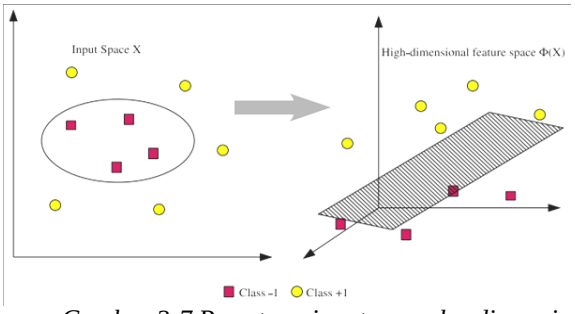
Dengan demikin persamaan Lagrange multipliers (8) menjadi :

$$\text{Maximize } \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i x_j \rangle \quad (11)$$

$$\text{Subject to } \alpha_i \geq 0 \ (i = 1,2 \ldots . l), \sum_{i=1}^{l} \alpha_i y_i = 0 \ , 0 \leq \alpha_i \leq C$$

2.4 Kernel Trick

Pada umumnya masalah dalam domain dunia nyata (*real world problem*) jarang yang bersifat *linear separable*, kebanyakan bersifat non *linear*. Untuk menyelesaikan problem non *linear*, SVM dimodifikasi dengan memasukkan fungsi Kernel. Dalam non *linear* SVM, pertama-tama data x dipetakan oleh fungsi Φ(x) ke ruang vektor yang berdimensi lebih tinggi. Pada ruang vektor yang baru ini, *hyperplane* yang memisahkan kedua class tersebut dapat dikonstruksikan.[12]



Gambar 2-7 Pemetaan input space berdimensi dua dengan pemetaan ke dimensi tinggi

Ilustrasi dari konsep ini dapat dilihat pada gambar 2-7. Pada gambar, sisi kiri diperlihatkan data pada *class* kuning dan data pada *class* merah yang berada pada input space berdimensi dua tidak dapat dipisahkan secara *Linear*. Selanjutnya pada gambar sisi kanan menunjukkan bahwa fungsi *Φ* memetakan tiap data pada input space tersebut ke ruang vektor baru yang berdimensi lebih tinggi dimana kedua *class* dapat dipisahkan secara *Linear* oleh sebuah *hyperplane*. Notasi matematika dari mapping ini adalah sebagai berikut.

$$\Phi : \Re^d \rightarrow \Re^q, x < x \quad (12)$$

Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*. Proses pembelajaran pada *SVM* dalam menemukan titik-titik *support vector*, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi, yaitu $\Phi(x_i) . \Phi(x_j)$.[12]

Umumnya transformasi Φ ini tidak diketahui, dan sangat sulit untuk difahami secara mudah,maka perhitungan *dot product* tersebut sesuai teori Mercer dapat digantikan dengan fungsi kernel $K(x_i, x_j)$ yang mendefinisikan secara implisit transformasi Φ. Hal ini disebut *Kernel Trick* [12]. Dengan ini, persamaan (11) diubah menjadi:

$$\text{Maximize } \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (16)$$

$$\text{Subject to } \alpha_i \geq 0 \ (i = 1,2 \ldots . l), \sum_{i=1}^{l} \alpha_i y_i = 0$$

Dimana $K(x_i, x_j)$ merupakan fungsi kernel.

2.5 Normalisasi data

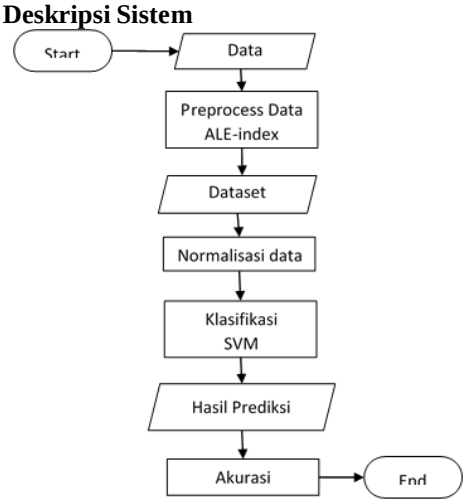
Normalisasi data adalah teknik menstandarkan nilai fitur atau variabel dari data. Hal ini bertujuan untuk menghindari atribut dengan range yang besar mendominasi atribut dengan range kecil. Untuk setiap data x, dilakukan perhitungan :

$$\frac{x - \mu}{\sigma} \quad (17)$$

Dimana x adalah data dalam sebuah fitur, μ adalah nilai mean dari matriks sebuah fitur, dan σ adalah nilai standar deviasi dari sebuah fitur [16]. Menurut Jason Weston, akurasi SVM akan menurun jika data tidak dinormalisasi [2].

3. ANALISIS DAN PERANCANGAN SISTEM

3.1 Deskripsi Sistem



Gambar 3-1 Gambaran sistem secara global

Secara umum proses yang dilakukan sistem yang dibangun dalam tugas akhir ini terdiri dari : *preprocessing* data, dan klasifikasi dengan *Support vector machine* yang nantinya menghasilkan prediksi yang diperoleh dan akurasi sistem.

3.2 Dataset yang digunakan

Data yang digunakan pada penelitian kali ini diambil dari situs kaggle [9]. Pada data terdapat 6 atribut yang terdiri dari ID pasien, Kondisi pasien setelah terapi (dimana 1 menunjukkan pasien mengalami perkembangan dan 0 berarti tidak mengalami perkembangan), PR sequence, RT sequence Viral load dan jumlah CD4.

3.3 Penanganan Noise

Pada kasus ini, terdapat karakter-karakter yang tidak dikenali oleh sistem preprosesing ini seperti karakter RYKMSW dan BDHV.

Tabel 3-1 Unsur pembentuk urutan nukleotida

Symbol	Description	Bases represented				
A	**A**denine	A				1
C	**C**ytosine		C			
G	**G**uanine			G		
T	**T**hymine				T	
W	**W**eak	A			T	2
S	**S**trong		C	G		
M	a**M**ino	A	C			
K	**K**eto			G	T	
R	pu**R**ine	A		G		
Y	p**Y**rimidine		C		T	
B	not A		C	G	T	3
D	not C	A		G	T	
H	not G	A	C		T	
V	not T	A	C	G		
N	any	A	C	G	T	

Unsur RYKMSW merupakan suatu unsur yang terbentuk dari gabungan 2 unsur utama pembantuk urutan nukleotida, sedangkan unsur BDHV dideskripsikan dengan unsur yang masih diragukan unsur utamanya namun terdapat diantara 3 unsur utama. Untuk itu dilakukan beberapa penanganan. Penangannanya adalah:

1. Jika menemukan karakter diantara RYKMSW maka dipilih salah satu penanganan dari beberapa penanganan berikut:
   a. **Skip**. Jika ditemukan karakter R,Y,K,M,S,W pada sequence, maka dilakukan penghapusan karakter tersebut pada sequence
   b. **Random.** Jika ditemukan karakter R,Y,K,M,S,W pada sequence, maka dilakukan random terhadap karakter yang membentuknya pada karakter tersebut.
   c. **Replace.** Jika ditemukan karakter R,Y,K,M,S,W pada sequence, maka karakter tersebut diganti dengan karakter yang membentuknya.
2. Jika menemukan karakter diantara BDHV maka dipilih salah satu penanganan dari beberapa penanganan berikut:
   a. **Skip**. Jika ditemukan karakter B,D,H,V pada sequence, maka dilakukan penghapusan karakter tersebut pada sequence
   b. **Random.** Jika ditemukan karakter B,D,H,V pada sequence, maka dilakukan random terhadap karakter yang membentuknya pada karakter tersebut.
   c. **Delete row.** Jika ditemukan karakter B,D,H,V pada sequence, maka sequence tersebut tidak dimasukkan kedalam proses selanjutnya

Dari kombinasi penanganan diatas, didapatkan 9 dataset yang digunakan dalam pengujian sistem.

Tabel 3-2 Dataset

Dataset	RYKMSW	BDHV
1	Random	Random
2	Random	Skip
3	Random	Delete Row
4	Skip	Random
5	Skip	Skip
6	Skip	Delete Row
7	Replace	Random
8	Replace	Skip
9	Replace	Delete Row

4. PENGUJIAN

Metode pengukuran akurasi yang akan digunakan pada pengujian ini adalah Balance Accuracy. Penggunaan metode ini dikarenakan jumlah data positif dan data negatif pada dataset yang digunakan tidak seimbang, yaitu dengan perbandingan 1:7. Hal ini mencegah sistem memperoleh akurasi yang tinggi ketika sistem memprediksi data dengan baik hanya pada kelas negative saja.[10]

$$\text{[illegible]} = 0.5 * \left(\frac{TP}{\text{[illegible]}} + \frac{T\text{[illegible]}}{\text{[illegible]}}\right)$$

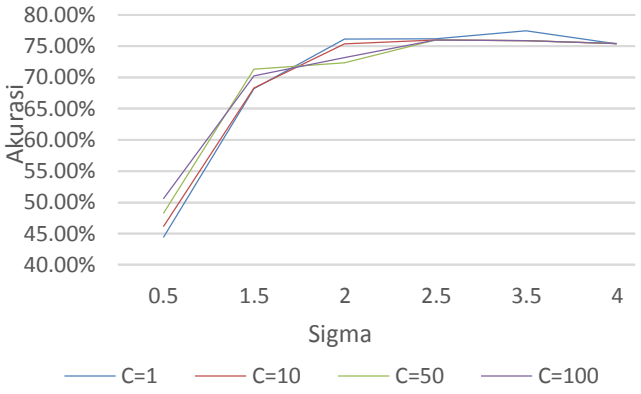
4.1 Pengujian terhadap dataset dan parameter SVM yang digunakan

Pengujian ini akan menggunakan 2 tipe kernel dengan beberapa parameter. Kernel RBF, dan kernel Polynomial. Untuk kernel RBF, akan diujikan dengan nilai sigma= 0.5, 1.5, 2, 2.5, 3.5, 4 dan kernel polynomial akan diujikan dengan menggunakan parameter poly ordo =2, 3, 4, 5. Nilai C yang digunakan yaitu 1, 10, 50, 100. Pembagian data dalam pengujian ini adalah 60% data latih dan 40% data uji. Hasil pengujian dapat dilihat pada table dan gambar.

Tabel 4-1 Hasil pengujian

Dataset	Kernel	C	*γ/d*	Akurasi (%)
Rand-Rand	RBF	10	3.5	75.0068
Rand-Skip	Poly	50	3	61.3061
Rand-Delete	RBF	**1**	**3.5**	**77.4686**
Skip-Rand	RBF	1	3.5	67.2721
Skip-Skip	RBF	1	2.5	67.7823
Skip-Delete	RBF	10	2	67.3113
Replace-Rand	RBF	10	2.5	75.3873
Replace-Skip	RBF	10	2.5	74.6803
Replace-Del	RBF	50	2.5	75.8491

Dari pengujian diketahui dataset Random-delete row dengan menggunakan SVM kernel RBF memperoleh akurasi yang lebih baik dibandingkan dengan dataset lain. Berikut merupakan grafik presentase perbandingan akurasi SVM kernel RBF pada tiap parameternya.



Gambar 4-1 Perbandingan akurasi kernel RBF tiap parameter

Dari grafik pada gambar 4-1, penambahan nilai sigma pada beberapa parameter terbukti mampu meningkatkan akurasi yang diperoleh sistem. Dan perbedaan parameter C dari hasil pengujian menghasilkan perbedaan akurasi yang tidak terlalu signifikan

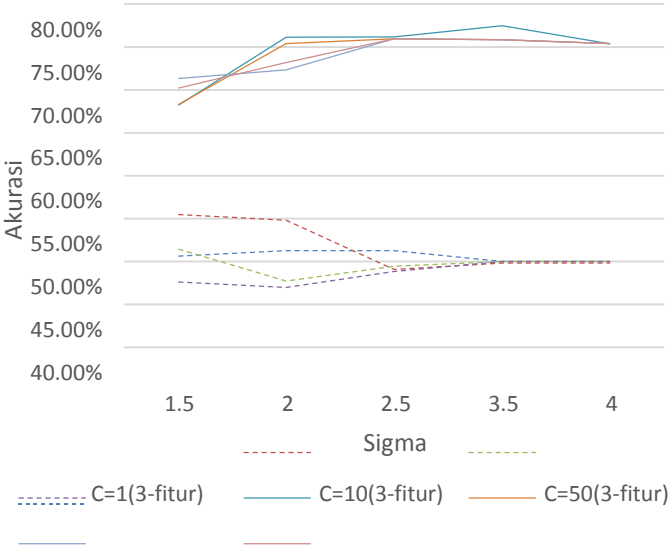
4.2 Pengujian terhadap fitur yang digunakan

Pada pengujian ini dilakukan pengujian sistem dengan menghilangkan salah satu fitur. Dataset yang digunakan pada pengujian ini adalah dataset 3, karna pada pengujian sebelumnya mendapatkan akurasi tertinggi. Untuk kernel polynomial, penulis sudah mencoba dengan beberapa parameter, namun belum memperoleh akurasi yang memuaskan. Pembagian data dalam pengujian ini adalah 60% data latih dan 40% data uji. Pada tahap uji pertama, fitur VL akan dihapus. Berikut hasil pengujian pada tahap uji pertama:

Tabel 4-2 Hasil pengujian tanpa fitur VL

Kernel	*γ*	C	Akurasi(%)
rbf	1	1.5	50.6446541
rbf	1	2	51.2893082
rbf	1	2.5	51.2893082
rbf	1	3.5	50
rbf	1	4	50
rbf	**10**	**1.5**	**55.4874214**
rbf	10	2	54.827044
rbf	10	2.5	49.0566038
rbf	10	3.5	49.8113208
rbf	10	4	49.8113208
rbf	50	1.5	51.4465409
rbf	50	2	47.7358491
rbf	50	2.5	49.4339623
rbf	50	3.5	50
rbf	50	4	50
rbf	100	1.5	47.6100629
rbf	100	2	46.9811321
rbf	100	2.5	48.8679245
rbf	100	3.5	50
rbf	100	4	50

Pada table terlihat akurasi yang diperoleh dengan menghilangkan fitur VL, ternyata sangat mempengaruhi akurasi yang didapatkan. Pada gambar 4-2, terlihat akurasi yang diperoleh dengan menghilangkan fitur VL jauh dibawah akurasi yang diperoleh dengan menggunakan fitur VL.



C=100(3-fitur) C=1(4Fitur)
C=10(4Fitur)

C=50(4Fitur) C=100(4Fitur)

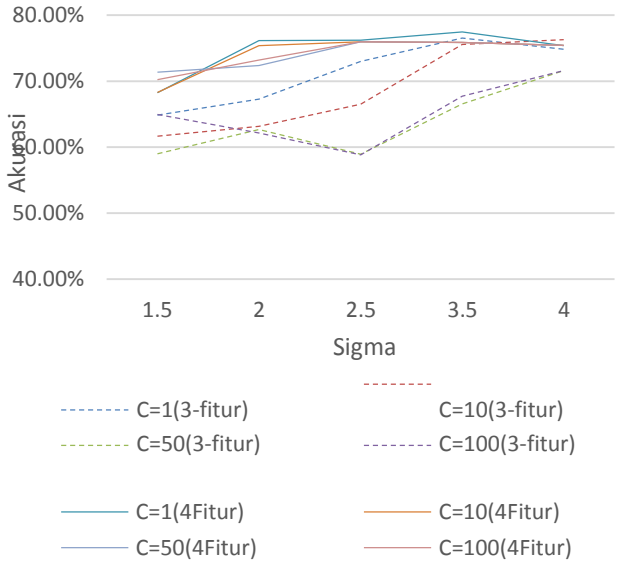
Gambar 4-2 Perbandingan akurasi 4-fitur dengan 3 - fitur

Pada tahap uji kedua, fitur CD4 dihapus dalam proses klasifikasi. Berikut hasil pengujian pada tahap uji kedua:

Tabel 4-3 Hasil pengujian tanpa fitur CD4

Kernel	γ	C	Akurasi(%)
rbf	1	1.5	64.827044
rbf	1	2	67.2641509
rbf	1	2.5	72.9874214
rbf	**1**	**3.5**	**76.509434**
rbf	1	4	74.8427673
rbf	10	1.5	61.6194969
rbf	10	2	63.1289308
rbf	10	2.5	66.4622642
rbf	10	3.5	75.5660377
rbf	10	4	76.2893082
rbf	50	1.5	58.9465409
rbf	50	2	62.672956
rbf	50	2.5	58.9150943
rbf	50	3.5	66.5408805
rbf	50	4	715880503
rbf	100	1.5	64.9371069
rbf	100	2	62.1069182
rbf	100	2.5	58.8050314
rbf	100	3.5	67.672956
rbf	100	4	71.5880503

Pada table 4-3 dan gambar 4-3, terlihat akurasi yang diperoleh dengan menghilangkan fitur CD4 hampir menyamai akurasi yang diperoleh dengan 4 fitur. Namun akurasi optimum yang didapatkan yaitu sebesar 76.5% belum bisa melampaui akurasi yang diperoleh dengan menggunakan keempat fitur.



Gambar 4-3 Perbandingan akurasi 4-fitur dengan 3-fitur

4.3 Pengujian terhadap pembagian data latih

Pada pengujian ini dilakukan pengujian sistem terhadap pembagian data latih yang berbeda. Pertama akan dilakukan pembagian data latih sebanyak 50%. Kedua pembagian data latih sebanyak 60%. Dan terakhir akan digunakan pembagian data latih sebanyak 70%. Data uji yang digunakan untuk menguji ketiga kondisi tersebut sebanyak 30%. Pengujian ini menggunakan kernel RBF dengan sigma pada range 1.5 sampai dengan 4, dan nilai

Tabel 4-4 Hasil pengujian pada jumlah data latih

50% Data Latih			60% Data Latih			70% Data Latih		
C	γ	Akurasi	C	γ	Akurasi	C	γ	Akurasi
1	1.5	70.7	1	1.5	71	1	1.5	69.0
1	2	78.5	**1**	**2**	**79.5**	1	2	73.7
1	**2.5**	**79.9**	1	2.5	77.6	1	2.5	77.3
1	3.5	79.1	1	3.5	78.7	1	3.5	76.6
1	4	78.1	1	4	77.6	1	4	76.4
10	1.5	74.7	10	1.5	73.6	10	1.5	71.8
10	2	75.5	10	2	77.5	10	2	71.6
10	2.5	79.1	10	2.5	77.5	10	2.5	76.0
10	3.5	77.9	10	3.5	78	**10**	**3.5**	**78.2**
10	4	77.9	10	4	77	**10**	**4**	**78.2**
50	1.5	72.2	50	1.5	75.4	50	1.5	73.4
50	2	72.1	50	2	75.4	50	2	71.3
50	2.5	79.1	50	2.5	77.4	50	2.5	73.7
50	3.5	77.9	50	3.5	78	**50**	**3.5**	**78.2**
50	4	77.9	50	4	77	**50**	**4**	**78.2**
100	1.5	69.3	100	1.5	74.4	100	1.5	72
100	2	73.1	100	2	77.0	100	2	68.8
100	2.5	79.1	100	2.5	77.5	100	2.5	72.4
100	3.5	77.9	100	3.5	78	100	3.5	76.7
100	4	77.9	100	4	77	**100**	**4**	**78.2**

Pada table diatas dapat dilihat penambahan jumlah data latih yaitu dari 50% data menjadi 60% data belum bisa meningkatkan akurasi optimum yang diperoleh sistem. Begitu juga dengan penambahan jumlah data latih dari 60% data menjadi 70% data.

5. KESIMPULAN

Berdasarkan hasil pengujian dapat diambil kesimpulan:

1. Metode SVM dengan menggunakan kernel *Radial Basis Function* pada kasus ini menghasilkan akurasi yang jauh lebih baik dibandingkan dengan akurasi yang diperoleh dengan menggunakan kernel *polynomial.*
2. Parameter C=1 dan sigma=3.5 pada kernel RBF, merupakan kombinasi parameter yang memperoleh akurasi terbaik pada kasus ini.
3. Parameter C pada kernel RBF terbukti mampu meningkatkan akurasi yang diperoleh oleh sistem,

C={1,10,50,100}. Hasil pengujian ketiga kondisi tersebut dapat dilihat pada table

akan tetapi pengaruh yang diberikan tidak terlalu signifikan terhadap akurasi yang didapatkan.

4. Metode Representasi numerik ALE-index sebagai invariant dari urutan nukleotida dapat menyelesaikan permasalahan dengan baik dalam menganalisis kemiripan antar urutan nukleotida.
5. Penananganan terbaik untuk karakter noise yang terdapat pada data adalah dengan metode Random pada karakter RYKMSW dan Delete-row pada karakter BDHV.
6. Keempat fitur dalam sistem prediksi ini yaitu Viral Load, jumlah CD4, RT sequence dan PR sequence masing-masing memiliki pengaruh dalam meningkatkan akurasi hasil prediksi yang diperoleh.

REFERENSI

[1] AIDS. (2013). *HIV-AIDS Basics*. Retrieved from http://aids.gov/hiv-aids-basics/

[2] Asa, B.-H., & Jason, W. (2010). A User's Guide to Support Vector Machines. *Data Mining Techniques for the Life Sciences*.

[3] Chun L and Wang J. (2005). New Invariant of DNA Sequences. *American Chemical Society*, *45*, 115-120.

[4] Cristianini, N. (2001). Support Vector and Kernel Machines.

[5] Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press.

[6] Jafarzadeh N and Iranmanesh A. (2012). A Novel Graphical and Numerical Representation for Analyzing DNA Sequences Based on Codons. *Communications in Mathematical and in Computer Chemistry*, *68*, 611-620.

[7] Jen Lin, C. (2006). Support Vector Machines and Kernel Methods. *Machine Learning Summer School.* Taipei.

[8] Joint United Nations Programmeon HIV/AIDS. (2009). *UNAIDS 2010 Global Report: Fact Sheet.*

[9] Kaggle. (n.d.). *Predict HIV Progression*. Retrieved from http://www.kaggle.com/c/hivprogression

[10] Marina, S., Nathalie, J., & Stan, S. (n.d.). Beyond Accuracy, F-score and ROC:a Family of Discriminant Measures for Performance Evaluation.

[11] Matthew, R., Lance, M., Milena, B., & Chan, A. (2006). Accurate prediction of HIV-1 drug response from the reverse transcriptase and protease amino acid sequences using sparse models created by convex optimization. *Bioinformatics, 22*, 541-549.

[12] Nugroho, Anto, S., Witarto, A. B., & Handoko., D. (2003). Application of Support Vector Machine in Bioinformatics. *FUSII*.

[13] *Onhealth, HIV-Basics.* (n.d.). Retrieved from http://www.onhealth.com/human_immunodeficiency_virus_hiv_aids

[14] Ramadhan, D. P. (2012). *Analisis Perbandingan Opinion Mining Berbahasa Indonesia Menggunakan Support Vector Machine dengan Kernel Linear dan Radial Basic Function.* Bandung: Fakultas Informatika Universitas Telkom.

[15] Shafer, R., & Rhee, S. (2007). HIV-1 protease and reverse transcriptase mutations for drug resistance surveillance. *Aids, 21*, 215-223.

[16] T, J., & Dr, A. (2011). Statistical Normalization and Back Propagation for classification. *International Journal of Computer Theory and Engineering, 3*, 89-93.

[17] Zhang Z, Meng L, Xiaoqing Y, Chun L. (2009). An ALE-index based algorithm facilitated prediction of success for polymerase chain reactions. *IEEE*.