JOURNAL OF FUNDAMENTAL MATHEMATICS
AND APPLICATIONS (JFMA) VOL. 4 NO. 2 (NOV 2021)
Available online at www.jfma.math.fsm.undip.ac.id

ANALISIS AKURASI DARI PERBEDAAN FUNGSI KERNEL DAN COST PADA SUPPORT VECTOR MACHINE STUDI KASUS KLASIFIKASI CURAH HUJAN DI JAKARTA

Noviana Pratiwi[1*], Yudi Setyawan[2]

[1,2]Jurusan Statistika, Fakultas Sains Terapan, IST AKPRIND Yogyakarta
Email: [1]novianapratiwi@akprind.ac.id, [2]yudista2003@yahoo.com
*Penulis korespondensi

Abstrak. Penelitian ini difokuskan pada perbandingan beberapa fungsi kernel, cost dan proporsi data training pada Support Vector Machine terhadap akurasi pengklasifikasian curah hujan di Jakarta. Fungsi-fungsi kernel linier, Gauss dan polynomial digunakan untuk memodifikasi metode Support Vector Machine guna menyelesaikan kasus nonlinier yang sering terjadi pada kondisi real. Variabel yang digunakan dalam penelitian ini meliputi temperatur, kelembaban, penyinaran matahari dan kecepatan angin. Hasil analisis menunjukkan bahwa nilai support vector terkecil tidak memberikan akurasi yang tertinggi pada masing-masing fungsi kernel. Selain itu, proporsi dataset (training:testing) sebesar 90%:10% memberikan akurasi sedikit lebih tinggi dibandingkan dengan akurasi untuk proporsi 80%:20% untuk masing-masing fungsi kernel. Secara keseluruhan, akurasi tertinggi diperoleh pada proporsi 90%:10% oleh fungsi kernel linier dan polinom untuk cost 1 dan 1000 secara bersamaan yaitu 78,38%.
Kata Kunci : *Cost*, *Gauss,* Kernel, *linear, polynomial,*

Abstract*. This research focuses on the comparison of several kernel functions, costs and proportions of data training on the Support Vector Machine to the accuracy of classifying rainfall in Jakarta. The linear, Gaussian and polynomial kernel functions were applied to modify the Support Vector Machine method to solve non-linear cases that often occur in actual conditions. The variables used in this study comprised of temperature, humidity, sunlight and wind speed. The analysis disclosed that the smallest support vector value did not provide the highest accuracy value for each kernel. In addition, the proportion of the dataset (training:testing) of 90%:10% provided a slightly higher accuracy compared to the accuracy for the proportion of 80%:20% for each kernel function. Overall, the highest accuracy attained at the proportion of 90%:10% by linear and polynomial kernel functions for cost 1 and 1000 simultaneously, which was 78.38%.*

I. PENDAHULUAN

Letak geografis Indonesia mengakibatkan Indonesia mempunyai iklim tropis yang berarti berpotensi memiliki curah hujan yang sangat tidak stabil dikarenakan penguapan air ke udara besar. Curah hujan di Indonesia memiliki tingkat hujan yang beragam berdasarkan ruang dan waktunya. Memasuki musim hujan, banyak daerah yang berpotensi banjir karena kurangnya resapan air hujan. Provinsi DKI Jakarta dan Banjir menjadi dua hal yang sulit dipasahkan untuk sekarang ini dikarenakan kurangnya resapan air. DKI Jakarta mempunyai enam kota yang semuanya berpotensi banjir karena kurangnya resapan air hujan. Enam kota tersebut antara lain Kepulauan seribu, Jakarta Utara, Jakarta Pusat, Jakarta Selatan, Jakarta Barat, dan Jakarta

Timur. Banyak kegiatan yang dilakukan pemerintah kota maupun provinsi untuk menanggulangi banjir di DKI Jakarta, namun penanganan ini belum berhasil 100% dikarenakan banyak faktor yang mempengaruhinya. Untuk itu perlu dilakukan klasifikasi dan predisksi curah hujan sebagai informasi untuk masyarakat agar masyarakat bisa bersiap-siap dalam menghadapi hujan yang mengguyur Jakarta. Curah hujan di Jakarta Utara menjadi fokus penelitian ini. Sesuai dengan letak geografisnya, Jakarta Utara mendapat serangan air baik dari hulu maupun hilir jika di Jakarta terjadi hujan. Bagian hulu merupakan sungai-sungai di Jakarta Pusat dan sekitarnya yang mengalir ke Jakarta Utara sedangkan bagian hilir adalah pesisir-pesisir pantai utara. Kepala Dinas Pekerjaan Umum menyebutkan bahwa 40% tanah di Jakarta tanahnya rendah sedangkan bagian yang lebih tinggi dari Jakarta Utara tidak menyerap air sehingga air mengalir dari daerah lain ditambah dengan curah hujan di Jakarta utara berpotensi terjadi banjir lebih besar dibanding wilayah sekitarnya.

Dalam *datamining*, klasifikasi merupakan salah satu teknik yang merupakan proses mengumpulkan data sesuai dengan cirinya tertentu dengan cara membuat model/fungsi berdasarkan data sesuai ciri tertentu tersebut. Ada banyak metode untuk klasifikasi dalam data mining diantaranya : *K-Nearest Neighbor, Artificial Neural Network, Classification Tree, Naïve Bayes Classifier* dan *Support Vector Machine* dan lain-lain. Prediksi cuaca dan curah hujan menggunakan data mining pernah diteliti oleh [1]. [1] melakukan klasifikasi dengan beberapa metode seperti *Association Rule, Random forest, classification tree* dan C4.4. Metode terbaik digunakan adalah C4.4 dengan nilai akurasi 69.5%. Tingkat akurasi lebih tinggi dimungkinkan bisa diperoleh jika kita melakukan analisis dengan metode lain yang lebih memiliki kecocokan denga data. Metode lain diharapkan akan menghasilkan nilai akurasi yang lebih baik jika dibandingkan dengan metode sebelumnya. Angka tersebut bisa dibilang kurang baik dan metode lain dimungkinkan akan memberikan hasil yang lebih baik.

Pada penelitian ini difokuskan pada penerapan metode Support Vector Machine (SVM) karena metode termasuk pada metode nonparametrik yang ini tidak memerlukan asumsi sehingga metode ini merupakan salah satu metode untuk mengatasi pelanggaran asumsi tertentu pada metode pengklasifikasian data. Selain itu metode ini juga bisa meminimalkan *eror* pada data training. SVM dapat mengeneralisasikan data yang tidak termasuk dalam data training secara cepat sehingga eror yang dihasilkan bisa lebih minimal, kelebihan lainnya adalah proses generalisasi tersebut juga tidak dipengaruhi oleh dimensi peubah yang dari objek yang diamati. Selain itu, [2] menyebutkan bahwa Support Vector Machine mampu melakukan generalisasi pada kasus dengan data terbatas. Hal lain juga diungkapkan [3], memberikan hasil bahwa Algoritma SVM memberikan hasil lebih baik dibanding *Naive Bayes* karena mempunyai akurasi yang lebih tinggi. Alasan tersebut diperkuat oleh [4] yang menyebutkan bahwa metode *Support Vector Machine* hanya tidak membutuhkan banyak observasi dalam pembentukan fungsi keputusannya. Belum lama ini, [5] juga melakukan klasifikasi curah hujan dengan membandingkan 2 algoritma dalam *machine learning* yaitu SVM dan *Naive Bayes Classifier* dengan hasil akurasi lebih tinggi diberikan oleh metode SVM.

Secara Teknis, *Support Vector Machine* adalah salah satu mesin pembelajaran yang memerlukan ruang hipotesis fungsi-fungsi linier dalam sebuah dengan ruang dimensi tinggi yang akan dilakukan *training* menggunakan algoritma pembelajaran yang berbasis pada teori optimasi. Penggunaan fungsi kernel juga membuat metode ini bisa digunakan pada data *time series*. [6] melakukan penelitian tentang *Support Vector Machine* dengan tiga kernel yang berbeda. Ternyata dari ketiga kernel tersebut memberikan hasil yang tidak jauh berbeda. [7] juga menggunakan metode SVM ini untuk melakukan klasifikasi pada sentiment vaksinasi

COVID-19 hasilnya menunjukkan bahwa dari keempat metode tersebut mempunyai aurasi yang tidak jauh berbeda. Penelitian terdahulu belum melihat metode ini dari sisi perbedaan nilai cost sebagai *hypermarameter*nya. Beberapa penelitian mengambil nilai cost yang sama untuk melakukan analisis SVM ini. Untuk itu peneliti ingin mengetahui apakah penerapan ketiga macam kernel dengan parameter yang jauh berbeda ini akan menghasilkan output yang tidak jauh berbeda juga jika diterapkan pada klasifikasi curah hujan dengan berbagai penggunaan hyperparameter cost.

II. TINJAUAN PUSTAKA

2.1. Pengumpulan data

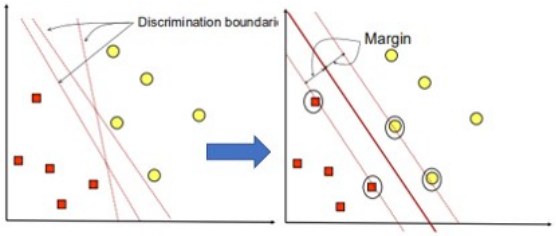
Data yang akan digunakan adalah data curah hujan yang bisa diakses melalui website www.bmkg.go.id. Variabel dalam penelitian ini ada dua jenis, yaitu variabel dependen (Y) berupa status curah hujan yang terdiri dari hujan dan tidak hujan, serta variabel independen (X) berupa temperatur, kelembapan, penyinaran matahari dan kecepatan angin. Klasifikasi curah hujan ini akan dilakukan dengan metode yang *Support Vector Machine* (SVM) dengan berbagai macam kernel. Langkah-langkah dalam melakukan metode SVM adalah sebgai berikut :

1. Mempersiapkan dataset untuk analisis
2. Jika terdapat data missing maka dilakukan penanganan data missing dengan bantuan metode interpolasi linier
3. Analisis Deskriptif untuk melihat karakteristik dari masing-masing variabel
4. Untuk memulai klasifikasi, pertama membagi data menjadi data training dan data testing. Disini kita menggunakan data training dan testing dengan proporsi 80%-20% dan 90%-10%.
5. Menentukan fungsi kernel yang akan digunakan untuk mencari fungsi *hyperplane*. Fungsi kernel yang akan digunakan adalah linier, Gauss dan polynomial.
6. Untuk mendapatkan nilai *support vector* atau biasa disebut *alpha*, ditentukan parameter C. Nilai parameter C yang digunakan adalah 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000
7. Menentukan persamaan *hyperplane* dengan alpha dan bias
8. Melakukan Prediksi pada kelas data testing berdasarkan persamaan *hyperplane*
9. Menghitung nilai akurasi dengan terlebih dahulu membuat tabel *confusion matrix*
10. Mengulangi langkah 6-9 dengan kernel yang berbeda
11. Melakukan perbandingan nilai akurasi
12. Melakukan analisis

2.2. Perbandingan fungsi kernel pada *Support Vector Machine*

Support Vector Machine (SVM) merupakan bagian dari data mining yang berfokus pada proses klasifikasi. SVM pertama kalinya dikenalkan oleh Vapnik pada tahun 1992. Dalam proses trainingnya, algoritma metode SVM menggunakan bantuan pemetaan non linier dimana proses SVM memetakan data ke ruang vektor dengan dimensi yang lebih tinggi. Ruang vektor dimensi yang lebih tinggi ini bertugas mencari *hyperplane* yang akan memisahkan secara linier dengan pemetaan nonlinier. Metode SVM digunakan untuk menyelesaikan masalah yang linier, namun dalam dunia nyata, kasus linier jarang sekali terjadi, kebanyakan kasus bersifat non linier sehingga perlu dilakukan modifikasi dalam metode SVM. Salah satu cara untuk memodivikasi agar metode ini bisa digunakan dalam kasus non-linier adalah dengan memasukkan fungsi kernel kedalam metode SVM. Dalam

penelitian ini, kita akan melakukan modifikasi beberapa fungsi kernel ke dalam metode SVM. [8] mengilustrasikan SVM pada Gambar 1 untuk lebih mudah dalam memahami metode.



. Konsep hyperplane (*Sumber:* [8])

Pada Gambar 1 sebelah kiri memperlihatkan *pattern. Pattern* tersebut anggota dari 2 kelas yang kita namakan -1 (kotak, merah) dan +1 (lingkaran, kuning). Tujuan akhir dari klasifikasi menggunakan metode SVM adalah menemukan *hyperplane* yang akan memisahkan kedua kelas tersebut. Proses pemisahan ini berada dalam ruang vector berdimensi d, terdapat *Affine subspace* berdimensi d-1 yang membagi dua bagian dimana dua bagian ini akan berhubungan dengan masing-masing kelas. Beberapa alternatif garis pemisah bisa dilihat pada gambar 1 bagian kanan. Untuk mendapatkan hyperplane, perlu diukur margin (gambar 1 sebelah kanan) dan dicari titik maksimalnya. Margin adalah jarak terdekat antara objek-objek dengan garis *hyperplane* dari masing-masing kelas. Objek terdekat dengan garis *hyperplane* ini dinamakan *support vector.* Garis pemisah yang baik adalah garis yang terletak persis diantara kedua kelas, secara matematis diperoleh :

$$H_1: x_i w + b \geq 1 \text{ untuk } y_1 = +1 \quad (1)$$

$$H_2: x_i w + b \leq -1 \text{ untuk } y_2 = -1 \quad (2)$$

Dengan w adalah vektor normal berukuran $1 \times p$ dan tegak lurus sengan *hyperplane*, x merupakan vektor pengamatan berukuran $p \times 1$ sedangkan saklar b disebut dengan simpangan dan y merupakan label kelas. Penggabungan kedua persamaan di atas menghasilkan persamaan baru yaitu

$$y_i(x_i w + b) \geq 1, \text{untuk } \forall i = 1, \cdots, n \quad (3)$$

Perhitungan margin dperoleh dengan menghitung jarak antara dua margin *hyperplane* yang disebut dengan H_1 dan H_2. Dengan meminimkan $x^T x$ kitadapat memperolah jarak terdekat antara titik yang berada di bidang H_1 terhadap pusat dengan tetap memperhatikan kendala $(x_i w + b) \geq 1$. Dengan bantuan fungsi Lagrange maka diperoleh

$$min\, x^T x - \lambda(w^T x + b - 1) \text{ atau}$$

$$\frac{d}{dx}(x^T x - \lambda(w^T x + b - 1)) = 0$$

$$\Rightarrow 2x - \lambda w = 0$$

$$\Rightarrow x = \frac{\lambda}{2} w$$

dengan mensubtitusikan x ke bidang $H_1: x_i w + b = 1$ diperoleh:

$$\Rightarrow \lambda = \frac{2(1-b)}{w^T w}$$

sehingga dengan mensubtitusikan kembali λ dan x diperoleh:

$$x = \frac{(1-b)}{w^T w} w$$

$$x^T x = \frac{(1-b)^2}{(w^T w)^2} w^T w = \frac{(1-b)^2}{w^T w}$$

maka jarak H_1 ke pusat adalah:

$$\|x\| = \sqrt{x^T x} = \sqrt{\frac{(1-b)^2}{w^T w}} = \frac{(1-b)}{\|w\|} \quad (4)$$

Hal serupa diatas analog dengan mencari jarak terdekat suatu titik pada bidang H_2 terhadap titik pusat, sehingga diperoleh jarak H_2 ke pusat:

$$\|x\| = \sqrt{x^T x} = \sqrt{\frac{(-b-1)^2}{w^T w}} = \frac{(-b-1)}{\|w\|} \quad (5)$$

Margin yang optminal dapat diperoleh dengan memaksimalkan jarak dari kedua bidang H_1 dan H_2. Jarak antara H_1 dan H_2 dapat dilihat sebagai berikut :

$$\left| \frac{(1-b)}{\|w\|} - \frac{(-b-1)}{\|w\|} \right| = \frac{2}{\|w\|}$$

memaksimalkan $\frac{1}{\|w\|}$ akan memberikan hasil yang sama dengan meminimumkan $\|w\|^2$ dan untuk menyederhanakan penyelesaian, seperti yang telah dibahas [9] perlu ditambahkan faktor $\frac{1}{2}$. Dan persamaannya menjadi:

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

dengan kendala $y_i(x_i w + b) \geq 1$, untuk $\forall i = 1,2,\cdots,n$ (n merupakan jumlah data training).

Untuk menyelesaikan persamaan di atas diperlukan pengali *Lagrange* α_i, dengan i=1,2,...,n, sehingga model masalah sebelumnya dapat ditulis:

Min $L_P = \frac{1}{2}\|w\|^2 + (0 - a_i \sum_{i=1}^{n} [y_i(x_i w + b) - 1])$

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i y_i x_i . w + \sum_{i=1}^{n} a_i y_i b + \sum_{i=1}^{n} a_i \quad (6)$$

Untuk mendapatkan nilai min L_P maka kita turnunkan persamaan (6) terhadap masing masing *primal variable* yaitu w dan b dan disamadengankan dengan nol sehingga diperoleh :

$$\frac{\partial \left(\frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i y_i x_i . w + \sum_{i=1}^{n} a_i y_i b + \sum_{i=1}^{n} a_i \right)}{\partial_b} = 0$$

$$\sum_{i=1}^{n} a_i y_i = 0 \quad (7)$$

dan

$$\frac{\partial \left(\frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i y_i x_i . w + \sum_{i=1}^{n} a_i y_i b + \sum_{i=1}^{n} a_i \right)}{\partial_w} = 0$$

$$w - \sum_{i=1}^{n} a_i y_i x_i = 0$$

$$w = \sum_{i=1}^{n} a_i y_i x_i \quad (8)$$

dan kemudian mensubstitusikan persamaan (7) dan (8) ke persamaan (6) untuk mendapatan nilai maksimal L_P terhadap variable a_i (*dual variable*), sehingga :

$$Maks\ L_D = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i [y_i(x_i . w + b) - 1]$$

$$\|w\|^2 = (w . w)$$

$$\|w\|^2 = \left(\sum_{i=1}^{n} a_i\, y_i x_i . \sum_{j=1}^{n} a_j y_j x_j \right) \quad (9)$$

dan

$$\sum_{i=1}^{n} a_i [y_i(x_i . w + b) - 1] = \sum_{i=1}^{n} a_i y_i x_i w + \sum_{i=1}^{n} a_i y_i b - \sum_{i=1}^{n} a_i$$
$$= \sum_{i=1}^{n} a_i y_i x_i \sum_{j=1}^{n} a_j y_j x_j + 0 - \sum_{i=1}^{n} a_i \quad (10)$$

sehingga diperoleh

$$L_p = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1,j=1}^{n} a_i y_i x_i a_j y_j x_j \quad (11)$$
$$\sum_{i=1}^{n} a_i y_i = 0, \text{ dan } 0 \leq a_i, i = 1, 2, \ldots, n$$

Nilai a_i akan diperoleh dengan penyelesaian persamaan (11) yang akan digunakan untuk mencari *primal variable* dengan rumus:

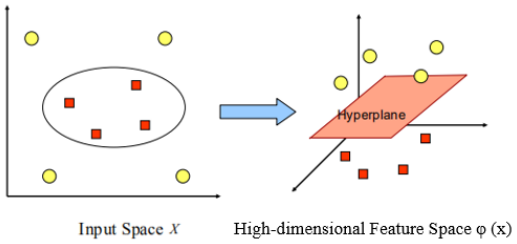
$$w = \sum_{i=1}^{n} a_i y_i x_i\,,\ b = y_i - x_i w \quad (12)$$

Nilai a_i yang dihasilkan dan lebih dari nol dinamakan *support vector*. Hal itu berarti fungsi keputusan yang dihasilkan dipengaruhi oleh *support vector*.

Nilai *support vector* akan mempenharuhi tingkat akurasi dalam menentukan metode yang lebih baik, beberapa peneliti menyebutkan bahwa semakin tinggi support vector akan menghasilkan keakuratan yang semakin tinggi pula. Parameter lain selain support vektor yang mempengaruhi tingkat akurasi adalah cost. Oleh karena itu pemilihan parameter cost juga menjadi fokus permasalahan pada penelitian ini. Teknik pemilihan *trial and eror* dilakukan untuk mendapatkan nilai akurasi yang optimal.

Metode Kernel

Fungsi kernel dalam SVM digunakan untuk menyelesaikan kasus non linier karena kebanyakan kasus dalam dunia nyata (*real world problem*) jarang terdapat kasus yang bersifat linier. Fungsi kernel ini akan dimasukkan kedalam algoritma SVM [8]



Gambar 2. Hyperplane [8]

Pada proses input, data x yang diinputkan akan dipetakan ke feature space F dengan dimensi yang lebih tinggi melalui *map* φ ($\phi: x \rightarrow \phi(x)$) menggunakan fungsi kernel. Fungsi *φ(x)* sering tidak bisa dihitung namun dot product dari kedua vector dapat dihitung di dalam *input space* maupun *future space* seperti pada persamaan (13) berikut:

$$\phi(x_i).\phi(x_j) \quad (13)$$

Pada gambar 2 sebelah kiri terlihat *input space* x berisi data data (kelas kuning lingkaran dan merah kotak) berdimensi dua dan tidak bisa dipisahkan secara linier. Dan gambar 2 sebelah kanan memperlihatkan bahsa sebuah fungsi φ berhasil memetakan data input ke dalam ruang vector baru dengan dimennsi 3 (lebih tinggi) dengan garis pemisah *hyperplane* yang linier.

Fungsi kernel disini lah yang menggantikan perhitungan dot product diatas sesuai dengan teori mercer. Secara implisit fungsi kernel $K(x_i.x_j)$ didefinisikan sebagai transformasi φ. Proses ini dikenal dengan sebutan *kernel Trick* dan dapat dirumuskan dalam persamaan (14) berikut :

(14)

Pada proses *training* dalam SVM, support vector akan lebih mudah diperoleh dengan *kernel trick* ini.kita tidak perlu mengetahui fungsi nonlinier φ karena fungsi φ sudah ditransformasi menjadi fungsi kernel $K(x_i.x_j)$, jadi kita hanya perlu mengetahui fungsi kernelnya [8]. Proses selanjutnya setelah diperoleh support vector dengan bantuan kernel diperoleh, maka akan dilakukan klasifikasi dari data x dengan persamaan (15) berikut :

$$f(x) = sign\left(\sum_{i=1}^{n} a_i y_i K\left(x_i^T.x_j\right) + b\right) \quad (15)$$

dimana:
x_i = Data *input* x baris ke-i
x_j = Data *input* x kolom ke-j
y_i = Kelas output baris ke-i
b = Nilai Bias
a_i = Nilai Alpha atau sebagai *support vector*
sign = Nilai yang besar dari 0 dilabelkan +1, Semua nilai yang kecil dari 0 dilabelkan -1.

Untuk menemukan fungsi keputusan persamaan (15) maka perlu memecahkan masalah tersebut, sebagai berikut:

$$MaksL_p = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, K\left(x_j . x_i^T\right)$$

dengan kendala : $0 \le a_i \le C$, i = 1,2,…,n dan $\sum_{i=1}^{n} a_i y_i = 0$, Sehingga

$$\sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, K\left(x_j . x_i^T\right) = \left(\sum_{j=1}^{n} a_j y_j \phi\,(x_j) . \sum_{i=1}^{n} a_i y_i \phi(x_i)\right) \quad (16)$$

Fungsi kernel yang akan kita gunakan dalam penelitian ini adalah kernel dasar yang fungsinya didefinisikan sebagai berikut :

a. Kernel Linier

$$K\left(x_i, x_j\right) = x_i^T . x_j \quad (17)$$

b. Kernel Polynomial

$$K(\vec{x_i}, \vec{x_j}) = (\vec{x_i} . \vec{x_j} + 1)^p \quad (18)$$

c. Kernel Gauss atau RBF

$$K(\vec{x_i}, \vec{x_j}) = exp\left(-\frac{||\vec{x_i} - \vec{x_j}||^2}{2\sigma^2}\right) \quad (19)$$

2.3 Confusion Matrix

Confusion Matrix (matriks konfusi) digunakan untuk mengukur kinerja dari proses klasifikasi. Matriks konfusi merupakan matriks yang berisi hasil kerja dari klasifikasi yang dituangkan dalam tabel. Matriks konfusi berbentuk seperti tabel 1 sebagai berikut [10]

Tabel 1. Tabel *Confusion Matrix*

	Prediksi		
Aktual		True	False
	True	TP	FN
	False	FP	TN

Keterangan :
TN = Jumlah prediksi yang tepat bersifat negatif (*True Negative*)
FN = Jumlah prediksi yang salah bersifat negatif (*False Negative*)
FP = Jumlah prediksi yang salah bersifat positif (*False Positive*)
TP = Jumlah prediksi yang tepat bersifat positif (*True Positive*)

Tingkat akurasi sebuah metode bisa dilihat dari matrik konfusi. Kinerja klasifikasi dapat dilihat dari tingkat akurasi melalui matiks konnfusi [11]. Beberapa nilai akurasi yang menjadi perhatian adalah sebagai berikut :

a Tingkat akurasi yang bisa dilihat dari proporsi klasifikasi yang benar dalam melakukan predisksi. Rumusnya bisa dilihat dalam persamaan (20) berikut

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (20)$$

b *sensitivity* adalah proporsi kasus dimana hasil klasifikasi prediksi positif kenyataanya diklasifikasi sebagai positif, dengan perhitungan persamaan sebagai berikut:

$$sensitivity = \frac{TP}{TP+FN} \tag{21}$$

c *True Negative* (TN) atau *specificity* adalah proporsi kasus yang diprediksi negatif yang kenyataanya diklasifikasi sebagai negatif, dengan perhitungan persamaan sebagai berikut:

$$TrueNegative = \frac{TN}{TN+FP} \tag{22}$$

d *Error Rate* merupakan proporsi dari prediksi yang salah, dengan perhitungan persamaan sebagai berikut:

$$ErrorRate = \frac{FP+FN}{TP+TN+FP+FN} \tag{23}$$

III. PEMBAHASAN

Pada Pembahasan ini akan dilakukan klasifikasi dari curah hujan. *Suppoert Vector Machine* (SVM) digunakan untuk melakukan klasifikasi ini. Modifikasi fungsi kernel diterapkan pada SVM. Fungsi kernel yang dipakai adalah kernel *Linier,Gauss dan Polinom.* Untuk klasifikasi menggunakan *support vector machine* ada beberapa nilai C (Cost) yang digunakan untuk mengetahui ketepatan klasifikasi yang terbaik, nilai cost yang digunakan dalam analisis ini yaitu 0.000001, 0.0001, 0.01, 1, 100, 10000, dan 1000000 untuk masing-masing fungsi kernel. Nilai cost ini yang akan digunakan untuk mencari support vector pada data training. Nilai Support Vector yang diperole untuk masing-masing kernel dan masing-masing Cost bisa dilihat pada tabel 2 berikut:

Tabel 2. Jumlah Support Vector

Nilai Cost	Jumlah Support Vector (α)		
	Linier	Gauss	Polinom
0,000001	244	244	244
0,001	245	246	245
1	169	186	169
1000	169	168	168
100000	159*	151*	166*

Tabel 2 memberikan hasil bahwa semua nilai cost 00.000001, 0.0001, 0.01, 1, 100, 10000, 1000000 menghasilkan jumlah *support vector* yang tidak jauh berbeda. Sehingga kita akan menggunakan semua model untuk melakukan klasifikasi. Nilai jumlah *support vector* (α) tersebut selanjutnya digunakan untuk mencari nilai bias. Selanjutnya, nilai *alpha* (α) dan *bias*(b) ini digunakan untuk membuat model yang akan digunakan untuk memprediksi terhadap data testing sesuai dengan persamaan (15) yang biasa disebut persamaan *hyperplane*. Perlu diketahui bahwa dalam *software R* status Hujan dilabelkan dengan kelas -1 sedangkan Tidak hujan dilabelkan dengan kelas +1 secara otomatis. Persamaan (15) untuk kernel linear adalah sebagai berikut:

$$f(x)_{linier} = sign\left(\sum_{i=1}^{n} a_i^T y_i^T K\left(x_i^T x_j\right) + 1.00004\right)$$

maka dapat dibentuk matriks fungsi kernel linier, dengan salah satu contohnya seperti berikut

$$f_1 = sign\left(\begin{bmatrix}0{,}000001\\0{,}000001\\\vdots\\0{,}000001\end{bmatrix}_{244\times1}\begin{bmatrix}1\\-1\\\vdots\\-1\end{bmatrix}_{244\times1}\begin{bmatrix}30 & 30{,}1 & \cdots & 29{,}5\\69 & 67 & \vdots & 70\\7{,}3 & 9{,}1 & \ddots & 8{,}1\\3 & 4 & \cdots & 2\end{bmatrix}_{4\times244}[29{,}7 \quad 72 \quad 5{,}7 \quad 1] + \begin{bmatrix}1.00004\\1.00004\\\cdots\\1.00004\end{bmatrix}\right)$$

$$f_1 = sign(-12329422037518) = -1.$$

Berdasarkan hasil yang diperoleh yaitu -1 yang artinya prediksi data testing pertama adalah hujan. Setelah proses klasifikasi selesai, maka selanjutnya kita mencari nilai akurasi yang dicari dengan matiks konfusi. Nilai akurasi tersebut beserta jumlah support vectornya (nSV:number *Support Vector*) dapat dilihat pada Tabel 3.

Tabel 3. Nilai Kinerja Klasifikasi berdasarkan nilai jumlah support vector

Nilai Cost	SV dan Akurasi untuk proporsi 90%:10%					
	Linier		Gauss		Polinom	
	nSV(α)	Akurasi	nSV(α)	Akurasi	nSV(α)	Akurasi
0,000001	244	0.3784	244	0.3784	244	0.3784
0,001	245	0.3784	246	0.3784	245	0.3784
1	169	0.7838*	186	0.7568*	169	0.7838*
1000	169	0.7838*	168	0.7297	168	0.7838*
100000	159**	0.3784	151**	0.7027	166**	0.3784

*nilai akurasi terbesar
**nilai *support vector* terbesar

Tabel 3 di atas menunjuukan bahwa tingkat akurasi yang dihasilkan tidak jauh beda untuk masing-masing kernel, dan nilai akurasi tertinggi pada masing-masing kernel diperoleh pada nilai cost 1 dengan angka akurasi yang tidak jauh beda untuk tiap kernelnya. Selain itu juga bisa diperoleh bahwa untuk nilai (nSV:number *Support Vector*) terkecil tidak menghasilkan akurasi yang besar. Untuk menambah analisis, dirubah proporsi data training dan data testing menjadi 80% :20%. Hasil perhitungan untuk proporsi 80%:20% bisa dilihat pada Tabel 4.

Tabel 4. Nilai Kinerja Klasifikasi berdasarkan nilai jumlah support vector proporsi 80%:20%)

Nilai Cost	SV dan Akurasi					
	Linier		Gauss		Polinom	
	nSV(α)	Akurasi	nSV(α)	Akurasi	nSV(α)	Akurasi
0,000001	214	0.4865	214	0.4865	214	0.4865
0,001	214	0.4865	215	0.4865	214	0.4865
1	146	0.7568*	159	0.7162	146	0.7568*
1000	145**	0,7568*	141*	0.7297*	145	0.7568*
100000	145**	0.7432	142	0.7027	127*	0.4865

*nilai akurasi terbesar
**nilai *support vector* terbesar

Sama seperti proporsi data training-testing 90%:10%, dari tabel 4 menunjukkan bahwa tingkat akurasi yang dihasilkan tidak jauh beda untuk masing-masing kernel, sedangkan tingkat akurasi tertinggi untuk masing-masing kernel diperoleh pada nilai cost 1 untuk kernel linier dan polynomial dan cost 1000 untuk kernel Gauss dengan angka akurasi yang tidak jauh beda untuk tiap kernelnya. *Support Vector* terkecil menghasilkan akurasi yang terbesar untuk kernel Linier dan Gauss namun tidak berlaku untuk kernel Polinom

IV. KESIMPULAN

Klasifikasi dan perkiraan curah hujan dengan metode kernel yang berbeda menghasilkan tingkat akurasi yang berbeda namun tidak terlalu jauh antara masing-masing kernel. Masing-masing proporsi data training dan data testing (90%:10% dan 80%:20%) menunjukkan hasil akurasi yang berbeda dengan kisaran 8%. Dari kedua proporsi juga tidak menunjukkan bahwa semakin kecil *support vector*, maka akan semakin besar tingkat akurasi, atau dengan kata lain tingkat akurasi tidak diperoleh dari nilai (α) terkecil. Penyebaran nilai akurasi terbesar dan terkecil untuk masing-masing proporsi menunjukkan hasil yang sama. Namun, secara keseluruhan, nilai akurasi tertinggi diperoleh pada proporsi 90%:10% oleh kernel linier dan Polinom untuk cost 1 dan 1000 secara bersamaan yaitu sebesar 0,7838 atau 78,38%. Nilai akurasi terbesar untuk kernel Gauss sebesar 75,68% dicapai untuk cost 1, Hal ini berarti klasifikasi curah hujan paling mendekati data asli adalah dengan metode SVM kernel linier ataupun polinom dengan proporsi 90%:10% pada cost 1 ataupun 1000.

REFERENSI

[1] S. Mujiasih, "Pemanfaatan Data Mining Untuk Prakiraan Cuaca," *BMKG,* pp. 189-195, 2011.

[2] E. W. Fridayanthie, "Analisa Data Mining Untuk Prediksi Penyakit Hepatitis Dengan Menggunakan Metode Naive Bayes dan Support Vector Machine," *Jurnal Khatulistiwa Informatika,* pp. 24-36, 2015.

[3] S. &. D. S. Vijayarani, "Prediksi penyakit hati menggunakan algoritma SVM dan Naive Bayes," *International Journal of Science, Engineering and Technology Research (IJSETR),* pp. 816-820, 2015.

[4] A. Siregar, Pemodelan Support Vector Machine Untuk Klasifikasi Curah Hujan Bulanan Di Kabupaten Indramayu, Bogor: IPB, 2017.

[5] M. &. S. Y. Laila, "Perbandingan Hasil Klasifikasi Curah Hujan menggunakan Metode SVN dan NBC," *Jurnal Statistika dan Komputasi, IST AKPRIND Yogyakarta,* 2020.

[6] F. S. E. Lumbanraja, "Prediksi Posisi Asetilasi Pada Protein Lisin Menggunakan Support Vector Machine," *Prosiding Seminar Nasional Sains, Matematika, Informatika dan Aplikasinya , Universitas Lampung,* vol. 5, no. 1, 2019.

[7] T. M. P. Aulia, N. Arifin and R. Mayasari, "PERBANDINGAN KERNEL SUPPORT VECTOR MACHINE (SVM) DALAM PENERAPAN ANALISIS SENTIMEN VAKSINISASI COVID-19," *SINTECH JOURNAL ,* Vols. 04-2, p. 139, 2021.

[8] S. A. &. W. Nugroho, "Support Vector Machine Teori dan Aplikasinya dalam Bionformatika," *Jurnal Ilmu Komputer,* pp. 1-11, 2003.

[9] S. N. &. F. K. Asiyah, "Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K-Nearest Neighbor," *Jurnal Sains & Seni,* pp. 317-322, 2016.

[10] A. &. O. I. Novandya, "Penerapan Algoritma Klasifikasi Data Mining C4.5 Pada Dataset Cuaca Wilayah Bekasi," *Jurnal Format,* pp. 98-106, 2017.

[11] M. &. N. D. Faisal, "Belajar Data Science Klasifikasi dengan Bahasa Pemrograman R," Banjarbaru, Kalimanta Selatan, Scripta Cendekia., 2019.