Mengatasi Permasalahan *High Dimensional Space* dalam Klasifikasi Multikelas *Big Data* pada Data Gambar dengan DCSVM

Gina Purnama Insany [a,1], Siti Sarah Sobariah Lestari [b,2,*], Dede Sukmawan [c,3], Faiz Dzulfikar Yusuf [d,4]

[a] Teknik Informatika Universitas Nusa Putra, JL.Cibolang No.21, Sukabumi 43152, Indonesia
[b] Statistika Universitas Islam Bandung, JL.Ranggagading No.8, Bandung 40116, Indonesia
[c,d] Sistem Informasi Universitas Nusa Putra, JL.Cibolang No.21, Sukabumi 43152, Indonesia
[1] gina.purnama@nusaputra.ac.id; [2] sitisobariah@unisba.ac.id *; [3] dede.sukmawan@nusaputra.ac.id; [4] faiz.dzulfikar_si21@nusaputra.ac.id
* Penulis Korespondensi

ABSTRAK

Masalah pada *unstructured* data khususnya pada data gambar menjadi tantangan dalam pemodelan klasifikasi *big data* yang berkaitan dengan akurasi dari model, sehingga memerlukan solusi untuk mengatasinya. Data gambar memiliki atribut yang disajikan dalam satuan piksel, satuan data tersebut berada pada dimensi yang tinggi. Sehingga menimbulkan ketidak beraturan pada dimensi, fenomena ini disebut sebagai *the curse of dimensionality*. Kenaikan dimensi pada data secara bersamaan mengakibatkan kenaikan volume pada ruang yang menyebabkan fenomena *sparsity* yang terjadi pada *high dimensional space*. Ketersediaan data jenis huruf dalam bentuk gambar, membuat data berada pada kondisi *data sparsity* pada *high dimensional space*, sehingga diperlukan metode yang mampu melakukan proses klasifikasi yang memberikan akurasi tinggi. Tujuan yang ingin dicapai dari penelitian ini adalah memperoleh model yang tepat dan akurat dalam melakukan klasifikasi multikelas dengan kondisi *data sparsity* dalam *high dimensional space*. *Support Vector Machine* (SVM) merupakan salah satu metode klasifikasi yang paling sering digunakan, karena SVM sensitif terhadap noise dan kaidah *VC-dimension* mampu menangani masalah *the curse of dimensionality*. Algoritma *Divide and Conquer*, membantu proses SVM dalam melakukan klasifikasi pada *data sparsity* dengan baik. Hal ini ditunjukkan dengan akurasi DCSVM (*Divide and Conquer Support Vector Machine*) lebih tinggi dibandingkan dengan klasifikasi dengan menggunakan SVM biner dan *one vs one* SVM

KATA KUNCI

Data Sparsity
Multidimentional Space
Multiclass Classification
SVM

ABSTRACT

Problems in unstructured data especially in image data, are a challenge in modelling big data classification related to the accuracy of the model, so it requires a solution to overcome it. Image data has attributes that are presented in pixel, these data are in high dimensions. This phenomenon is referred to as the curse of dimensionality. The increase in the dimensions of the data simultaneously results in an increase in the volume of space which causes the phenomenon of sparsity that occurs in high dimensional space. The availability of font data in the form of images, makes the data in the condition of sparsity data in high dimensional space, so a method is needed that is able to perform a classification process that provides high accuracy. The purpose of this research is to obtain the right and accurate model in performing multiclass classification with sparsity data conditions in high dimensional space. Support Vector Machine (SVM) is one of the most commonly used classification methods, because SVM is sensitive to noise and the VC-dimension rule is able to handle the curse of dimensionality problem. The Divide and Conquer algorithm, helps the SVM process in classifying sparsity data well. This is shown by the higher accuracy of DCSVM (Divide and Conquer Support Vector Machine) compared to classification using binary SVM and one vs one SVM

KEYWORD

Data Sparsity
Multidimentional Space
Multiclass Classification
SVM

1. Pendahuluan

Perkembangan teknologi membawa kita ke dalam masalah big data yang digambarkan dengan tiga sifat umum yakni *velocity*, *volume*, dan *variety*. Keberagaman data tidak hanya dilihat dari jenis skala yang dimiliki pada data, namun juga pada bentuk data yang berupa *structured* dan *unstructured* data. Masalah pada *unstructured* data khususnya pada data gambar menjadi tantangan dalam pemodelan klasifikasi *big data* yang berkaitan dengan akurasi dari model, sehingga memerlukan solusi untuk mengatasinya. Data gambar memiliki atribut yang disajikan dalam satuan piksel, satuan data tersebut berada pada dimensi yang tinggi. Sehingga menimbulkan ketidak beraturan pada dimensi, fenomena ini disebut sebagai *the curse of dimentionality*. Kenaikan dimensi pada data secara bersamaan mengakibatkan kenaikan volume pada ruang yang menyebabkan fenomena *sparsity* yang terjadi pada *high dimentional space*.

Ketersediaan data jenis huruf dalam bentuk gambar, membuat data berada pada kondisi *sparsity* pada *high dimentional space*, sehingga diperlukan metode yang mampu melakukan proses klasifikasi yang memberikan akurasi tinggi. Tujuan yang ingin dicapai dari penelitian ini adalah memperoleh model yang tepat dan akurat dalam melakukan klasifikasi multikelas dengan kondisi *data sparsity* dalam *high dimentional space* dan mengelompokkan beberapa bentuk huruf yang belum masuk ke dalam *family font* berdasarkan bentuknya.

Support Vector Machine (SVM) merupakan salah satu metode klasifikasi yang paling sering digunakan, karena SVM sensitif terhadap noise dikarenakan adanya kaidah *VC-dimension*, mampu menangani masalah *the curse of dimensionality*. Algoritma *Divide and Conquer*, membantu proses SVM dalam melakukan klasifikasi pada data *sparsity* dengan baik. Hal ini ditunjukkan dengan akurasi DCSVM (*Devide and Conquer Support Vector Machine*) lebih tinggi dibandingkan dengan klasifikasi dengan menggunakan SVM biner dan *one vs one* SVM.

Data *sparsity* terjadi dikarenakan data observasi tidak memadai menyebabkan akurasi model rendah. Bertambahnya dimensi pada saat *pre-processing* data, membuat volume ruang meningkat dengan cepat, sehingga mengakibatkan data memiliki rentang yang cukup jauh karena terdapat observasi yang mengandung nilai nol (*sparse*). Sehingga untuk mengetahui distirubusi dan pola dari data tersebut cukup sulit [1]. Secara umum, masalah data *sparsity* memberikan dampak pada pemodelan dengan metode yang memerlukan signifikansi secara statistik [2]. Penentuan *hyperplane* dalam proses pengkalsifikasian dengan SVM, penting dilakukan untuk memisahkan data input. Berdasarkan *VC-dimenssion*, pemisah *hyperplane* akan selalu berada pada dimensi *d-1*, sehingga apabila terdapat masalah *nonsepearable*, akan dilakukan transformasi pada dimensi yang lebih tinggi dan menyebabkan data berada pada *high dimension space*. Kondisi ini dapat memungkinkan terjadinya fenomena data *sparsity*.

Fungsi dari *VC-dimension* digunakan untuk generalisasi model *machine learning* sebelum dibuat. Fungsi dari *VC-dimension* kemungkinan besar dapat mengatasi masalah *the curse of dimensionality* yang memberikan dampak data *sparsity* [3]. Fenomena *high dimentional space* berkaitan dengan data *sparsity* yang menyebabkan kurangnya akurasi saat klasifikasi pada *unstructured* data. SVM merupakan salah satu metode klasifikasi *supervised learning* yang juga memanfaatkan konsep *higher dimentional space* dalam mentransformasi data yang sulit diklasifikasikan dikarenakan kondisi tertentu (*unseaperable*) sehingga diperlukan *hyperplane* yang optimum. Namun penelitian yang membahas mengenai klasifikasi multikelas dengan permasalahan data *sparsity* pada *high dimentional space*, masih belum banyak dikembangkan sedangkan ketersediaan data pada dunia nyata memiliki kecenderungan pada masalah data *sparsity*.

Perkembangan metode SVM dalam analisis klasifikasi multikelas masih dalam topik pembahasan yang menarik untuk diteliti penelitian yang membahas mengenai hal tersebut tersaji dalam Tabel 1.

Tabel 1. *State of The Art* Penelitian

No	Topik	Hasil	Referensi
1.	*Comparation Between Neural network and Support Vector Machine in Optical Character Recognition.*	Melakukan klasifikasi biner dari berbagai macam metode ekstraksi fitur untuk karakter jenis tulisan. Hal ini menyebabkan data *sparsity*.	[4]
2.	*Optical Character Recognitions Systems for Different Languages with Soft Computing.*	Menjelaskan teknis kerja OCR yang terbagi ke dalam digitalisasi pada dokumen (hanya untuk dokumen analog) dengan melakukan pemindaian dokumen untuk memperoleh informasi yang mengandung karakter, kemudian dilakukan proses penghilangan *noise* pada data, dilakukan proses segmentasi untuk mengetahui bagian dari gambar piandaian mana yang merupakan huruf atau karakter. Hasil segmentasi, dilakukan ekstrasi fitur dan selanjutnya dilakukan klasifikasi.	[5]
3.	Klasifikasi Kabupaten di Provinsi Jawa Timur Berdasarkan Indikator Daerah Tertinggal dengan Metode Support Vector Machine (SVM) dan Entropy Based Fuzzy Support Vector (ESVM)	Melakukan klasifikasi Multikelas pada kabupaten di provinsi Jawa Timur berdasarkan indikator daerah tertinggal dengan metode SVM dan EFSVM	[6]
4.	MC²ESVM : *Multiclassification based on coorperative evolution of support vector machines*	Mengembangkan metode SVM untuk pengklasifikasian multi kelas dengan konsep dekompisisi *M* kelas menjadi *M* Subkelas yang dikenal menjadi *single machine method* yang dikenal sebagai *Multi class classification based on coorperative evolution of support vector machine* (MC²ESVM)	[7]
5.	*Image Classification by Using Multiclass Support Vector Machines.*	Melakukan klasfisikasi sebagai *Content-based image retrieval* dengan menggunakan fitur visual gambar yakni warna, bentuk, tekstur, dan *spatial layout* untuk mewakilkan dan memberi indeks pada gambar untuk melakukan query pada data base. Skema yang digunakan adalah *one-vs-one* dan *pairwise* SVM. Namun dengan mempertimbangkan fitur kompleks yang ada dalam gambar kueri dan batasan komputasi dari algoritma ekstraksi fitur, metode ini tidak dapat memberikan hasil yang efektif.	[8]
6.	Penggunaan Metode Support Vector Machine Klasifikasi Multiclass pada Data Pasien Penyakit Tiroid	Menggunakan klasifikasi SVM dengan skema *one-vs-one* dan *one-vs-against* untuk mendiagnosa penyakit tiroid berdasarkan gambar hasil scan pasien. Namun penelitian ini tidak membahas mengenai kendala dan ekstrasi fitur pada gambar.	[9]
7.	*Improved Support Vector Machine and Image Processing Enabled Methodology for Detection and Classification of Grape Leaf Disease*	Menerapkan CLAHE dalam ekstraksi fitur pada data gambar kemudian dilakukan klasifiksi dengan menggunakan PSO-SVM, BPN, dan random forest. Hasil akurasi dengan PSO-SVM lebih tinggi.	[10]

Berdasarkan Tabel 1, penelitian yang membahas mengenai pengklasifikasian multikelas pada jenis tulisan masih jarang dilakukan umumnya penelitian mengenai karakter tulisan berfokus kepada sistem yang digunakan dalam ekstraksi fitur saja. Beberapa penelitian melakukan klasifikasi pada data gambar, namun mereka mengabaikan permasalahan yang terjadi akibat perubahan dimensi pada ekstraksi fitur. Hal ini karena ekstraksi fitur pada gambar untuk data teks dengan gambar biasa memiliki metode yang berbeda.

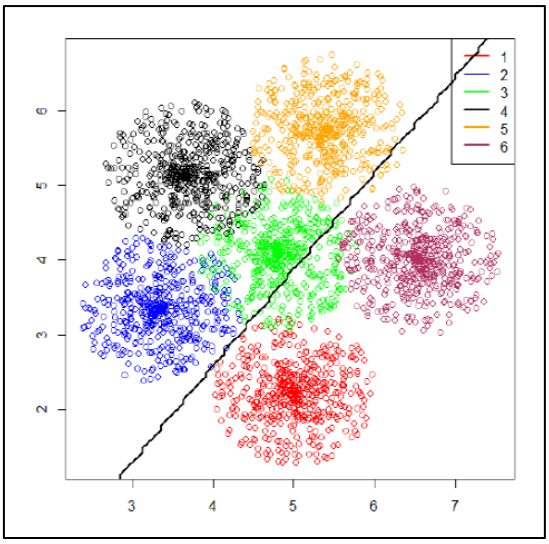
Metode yang diajukan adalah *Devided and Conquer Support Vector Machine* (DCSVM) metode ini dapat mempartisi *data trainig* ke dalam subset yang terpisah sehingga dapat dikelompokkan dengan mudah ke

dalam kelas klasifikasi dengan mengoptimalkan nilai *threshold*. Hal ini yang tidak dihiraukan dalam proses klasifikasi dengan metode multi kelas terdahulu. Fungsi dari *VC-dimension* dapat digunakan untuk generalisasi *mahine learning* dalam kasus SVM, generalisasi yang digunakan dengan pendekatan *VC-dimension* adalah SRM. Fungsi dari *VC-dimension* ini berkemungkinan besar dapat mengatasi masalah *the curse of dimensionality* [3] Dengan demikian DCSVM diklim mampu mengatasi hal tersebut.

2. Tinjauan Pustaka

2.1. *Devide and Conquer Support Vector Machine* (DCSVM)

DCSVM melakukan prediksi tunggal antara dua kelas yang dipartisi, dalam satu partisi dilakukan eliminasi satu kelas atau lebih yang menyisakan kandidat kelas yang telah tereduksi. Pada Gambar 1 terdapat beberapa kelas yang dapat dipisah dengan baik dan terdapat sebagian kelas lainnya tidak terpisah dengan baik sehingga menyebabkan *missclassification*.



Gambar 1. Ilustrasi Klasifikasi Enam Kelas

Jika classifier memprediksi kelas 1, maka selanjutknya kita akan memutuskan antara kelas ke 1,6,3, dan 5. Jika classifier memprediksi kelas 2 kemudian keputusan selanjutnya diantara kelas 2,4,3, dan 4. Algoritma DCSVM bekerja secara rekusif pada kasus kelas yang lebih sedikit. Mengingat banyaknya kelas adalah k, pada skenario terbaik, sebagian kelas akan dieliminasi pada setiap langkah dan algoritma akan selesai dalam waktu peluruhan $log_2 k$.

Pada Gambar 1, kelas 2 dan 4 terpisah secara sempurna dengan kelas 1 dan 6, sedangkan kelas 3 dan 5 tidak begitu jelas berada pada salah satu sisi dari garis pemisah. Terlepas dari pengklasifikasian pertama, kelas 3 dan 5 menjadi bagian untuk pengambilan keputusan pada langkah selanjutnya. Namun demikian, terdapat perbedaan yang signifikan antara kelas 3 dan 5. Meski garis pemisah hampir memisahkan setengah bagian dari kelas tiga, kelas 5 juga tetap dapat diprediksi sebagai kelas 2 dengan nilai error yang sangat kecil. DCSVM menggunakan nilai batas untuk menunjukkan maximum classification error yang diterima untuk mempertimbangkan masalah missclassification. Semakin tinggi nilai batas menghasilkan lebih sedikit kelas pada tahap selanjutnya juga akan memisahkan kelas lebih baik (less overlaping) nilai tersebut berpengaruh terhadap nilai akurasi dari hasil prediksi akhir.

Data set dari kelas ke-k ditunjukkan dalam D dimana, $\boldsymbol{x} \in D$ dengan label $l \in \{1, \dots, k\}$, fungsi $dcsvm$ dibangun dengan $D \rightarrow \{1, \dots, k\}$ sehingga $dcsvm\ (\boldsymbol{x}) = l$, dengan l merupakan label yang sesuai dari $\boldsymbol{x} \in D$. Dengan mempertimbangkan pembagian $D = R \cup T$ dari dataset D yang menjadi dua buah *disjoint* dengan R merupakan *training set* dan T merupakan *testing set*, data pada R digunakan untuk membangun fungsi $dcsvm()$ kemudian akurasi dari model diukur dengan menggunakan *testing set*. $R = R_1 \cup R_2 \cup \dots \cup R_k$ merupakan satu kesatuan *disjoint* dari R_1 dimana $\boldsymbol{x} \in R_1$ dengan label l untuk $l = 1, \dots, k$. Hal yang serupa berlaku pada *testing set*. Dalam metode DCSVM, fungsi $svm()$ yang digunakan lebih praktikal, dimana fungsi tersebut dipengaruhi oleh *missclassification errors*.

ISSN 2686-4797
Jurnal Restikom : Riset Teknik Informatika dan Komputer
Vol. 5, No. 3, Desember 2023, pp. 340-351

Misalkan $svm_{i,j} : D \rightarrow \{i, j\}$, merupakan *classifier* SVM biner yang dibentuk dati *training set* $R_i \cup R_j, i < j$ dan $i = 1, \dots, k-1, j = 2, \dots, k$. Sehingga didapat $k(k-1)/2$ hal ini serupa dengan *classifier* biner *one-vs-one*. Don (2018) menyebutkan bahwa penting untuk mengetahui bahwa dalam metode DCSVM, fungsi $svm()$ yang digunakan bukan hanya secara umum, namun lebih praktikal, dimana kemungkinan fungsi tersebut dipengaruhi oleh *missclassification errors* misalnya untuk beberapa $\boldsymbol{x} \in R_i \cup T_i$, fungsi SVM menjadi $svm_{i,j}(\boldsymbol{x}) = j$.

Decission fuction $dcsvm()$ dibangun dengan meminimalkan jumlah *decission* biner untuk klasifikasi k kelas, tanpa mengorbankan akurasi pada klasifikasi. Metriks dari *decisssion tree* seperti *gini, impurity,* dan *gain information* dapat juga dimanfaatkan dalam metode ini. Nanmun demikian kebanyakan DCSVM dengan *decission tree* tidak dapat memisahkan set kelas ke dalam beebrapa *disjoint*. Karena beberapa kelas bisa jadi berada pada kedua akar *subtrees* dalam node *decission* yang sama. Beberapa pengukuran yang digunakan dalam proses mengidentifikasi jalur yang paling sederhana dalam pengambilan keputusan klasifikasi multikelas dengan DSCVM didefinisikan oleh Don (2018) sebagai berikut :

Class prediction Likelihoods dari *classifier* biner SVM $svm_{i,j}(.)$ untuk label $l \in \{1, \dots, k\}$ masing-masing dilambangkan dengan $C_{i,j}(l, i)$ dengan rumus matematis :

$$C_{i,j}(l,i) = \frac{\left|\{\boldsymbol{x} \in R_l | svm_{i,j}(\boldsymbol{x}) = i\}\right|}{|R_l|}$$

$$C_{i,j}(l,j) = \frac{\left|\{\boldsymbol{x} \in R_l | svm_{i,j}(\boldsymbol{x}) = j\}\right|}{|R_l|} = 1 - C_{i,j}(l,i) \qquad (2.1)$$

Setiap *class prediction likelihood* mewakili hasil *likelihood* yang diharapkan untuk i atau j ketika sebuah *classifier* biner $svm_{i,j}(.)$ digunakan untuk memprediksi pada seluruh item data pada R_l. *Likelihood* dihitung untuk setiap *classifier* biner dan setiap kelas pada data *training set*.

Tabel Prediksi, memuat hasil *class prediction likelihood* dimana setiap baris menunjukkan setiap *classifier* $svm_{i,j}$ biner dan setiap kolom menunjukkan setiap kelas $1, \dots, k$ untuk membentuk tabel $\mathcal{T}$ dimana setiap entri di isi dengan sepasang *prediction likelihood* yang dihitung dengan :

$$\mathcal{T}[svm_{i,j}l] = (C_{i,j}(l,i), C_{i,j}(l,j)) \qquad (2.2)$$

Untuk mengetahui kualitas dari kalsifikasi, maka dihitung melalui dua pengukuran yakni inkeds *impurity* dan indeks *balanced*. Indeks *Impurity* mengukur seberapa baiknya *classifier* biner dalam mengklasifikasikan semua kelas sebagai i atau j berdasrkan nilai *threhold* θ yang menurpakan kekuatan memisahkan suatu label kelas dengan label kelas lainnya. Singkatnya sebuah kelas l diklasifikasikan dengan jelas sebagai i dengan $svm_{i,j}(.)$ jika $C_{i,j}(l,i) \geq 1 - \theta$ dan diklasifikasikan sebagai j jika $C_{i,j}(l,j) \geq 1 - \theta$ selain itu maka akan diklasifikasikan sebagai "*undecideced*" i atau j.

Indeks *impurity* menghitung seberapa banyak "*undecideced*" yang dihasilkan oleh *classifier* biner. Semakin rendah indeks menghasilkan *seperation* yang semakin baik. Indeks *balanced* menghitung seberapa imbangnya suatu pemisah dalam sejumlah kelas yang di prediksi sebagai i dan j. Semakin besar nilai indeksnya menunjukkan *seperation* yang semakin baik. Indeks *Impurity* dan *Balanced* SVM dengan θ dari *classifier* SVM $svm_{i,j}(.)$ didefinisikan sebagai berikut [2] :

Indeks *Impurity* dilambangkan dengan $\mathcal{P}_{i,j}(\theta)$:

$$\mathcal{P}_{i,j}(\theta) = \left(\sum_{l=1}^{k} (\chi_\theta(C_{i,j}(l,i)) + \chi_\theta(C_{i,j}(l,j))\right) - k \qquad (2.3)$$

Indeks *Balanced* dilambangkan dengan $\mathcal{B}_{i,j}(\theta)$:

$$\mathcal{B}_{i,j}(\theta) = min\left(k - \sum_{l=1}^{k}(\chi_\theta(C_{i,j}(l,j)), k - \sum_{l=1}^{k}(\chi_\theta(C_{i,j}(l,i))\right) \quad (2.4)$$

Dimana χ_θ adalah fungsi :

$$\chi_\theta(x) = \begin{cases} 1 & \text{Jika } x > \theta \\ 0 & \text{JIka } x \leq \theta \end{cases}$$

Semakin tinggi nilai SVM ($S_{i,j}$) menujukkan semakin baik *classifier* tersebut. Secara matematis, dihitung dengan :

$$S_{i,j} = \frac{C_{i,j}(i,i) + C_{i,j}(j,j)}{2} \quad (2.5)$$

Decission fuction $dcsvm()$ dibangun dengan meminimalkan jumlah *decission* biner untuk klasifikasi k kelas, tanpa mempengaruhi akurasi pada klasifikasi. Metode ini juga menggunakan metriks dari *decisssion tree* seperti *gini, impurity,* dan *gain information*. DCSVM dengan *decission tree* tidak dapat memisahkan set kelas ke dalam beberapa *disjoint*. Karena beberapa kelas bisa jadi berada pada kedua akar *subtrees* dalam node *decission* yang sama. Pengukuran yang digunakan dalam pengambilan keputusan klasifikasi multikelas dengan DSCVM sebagai berikut.

1. *Class prediction Likelihoods* dari *classifier* biner SVM $svm_{i,j}(.)$ untuk label $l \in \{1, \dots, k\}$ masing-masing dilambangkan dengan $C_{i,j}(l,i)$. Setiap *class prediction likelihood* mewakili hasil *likelihood* untuk i atau j ketika sebuah *classifier* biner $svm_{i,j}(.)$ digunakan untuk memprediksi pada seluruh item data pada R_l. *Likelihood* dihitung untuk setiap *classifier* biner dan setiap kelas pada data *training set*.
2. Tabel Prediksi, memuat hasil *class prediction likelihood* setiap baris menunjukkan *classifier* $svm_{i,j}$ biner dan setiap kolom menunjukkan kelas $1, \dots, k$.
3. Indeks *Impurity* mengukur seberapa baiknya *classifier* biner dalam mengklasifikasikan semua kelas sebagai i atau j berdasrkan nilai *thrasehold* θ yang memisahkan suatu label kelas dengan label kelas lainnya. Sebuah kelas l diklasifikasikan dengan jelas sebagai i dengan $svm_{i,j}(.)$ jika $C_{i,j}(l,i) \geq 1-\theta$ dan diklasifikasikan sebagai j jika $C_{i,j}(l,j) \geq 1-\theta$ selain itu maka akan diklasifikasikan sebagai "*undecideced*" i atau j. Indeks *impurity* menghitung seberapa banyak "*undecideced*" yang dihasilkan oleh *classifier* biner, semakin rendah indeks menghasilkan *seperation* yang semakin baik.

Indeks *balanced* menghitung seberapa imbangnya suatu pemisah dalam sejumlah kelas yang diprediksi *i* dan *j*. Semakin besar nilai indeks artinya tidak ada masalah *misclassification*.

3. Metodologi Penelitian

3.1. Data Penelitian

Data pada penelitian ini merupakan data beberapa karakter huruf dan tanda baca dari media elektronik berupa poster, *flyer*, dan buku elektronik. Karakter huruf dan tanda baca tersaji dalam bentuk gambar. terdiri dari 153 jenis huruf yang terbagi ke dalam empat *family font* yakni *serif, sans-serif, script,* dan *decorative*. Data diperoleh dari UCI *Learning*. Setiap obervasi berasal dari dokumen yang berbeda dan hanya diambil satu huruf dari setiap dokumen. Terdapat 412 fitur yang dapat dijadikan sebgai variabel untuk klasifikasi. Setiap *family font* diambil sebanyak 2.500 data gambar sehingga total observasi sebanyak 10.000.

3.2. Alur Kerja DCSVM

1. Pengenalan karakter pada huruf dengan menggunakan sistem OCR
2. *Data spliting* pada tahap ini data dibagi ke dalam dua bagian yakni data *training* yang digunakan suntuk membangun model klasifikasi dan data *testing* untuk melihat akurasi model.
3. **Algoritma 1** pada algortima ini dilakukan prosedur *traning* DCSVM, semua pasangan kelas dilatih dengan *clasifier* SVM biner, kemudian dihitung prediksi *likelihood* sebagaimana ditulis pada persamaan 3.2 kemudian disimpan dalam tabel prediksi. selanjutnya dibuat fungsi untuk *dcsvm* dengan menyediakan *empty tree* sebagai tempat dari hasil proses selanjutnya yang merupakan

proses DCSVM-*Subtree* memunculkan hasil melalui proses yang berjalan secara rekusif kemudian membuat node dari kiri atau kanan untuk setiap langhaknya atau prosesnya berhenti kemudiian membuat label. Setiap node baru berasosiasi dengan $\boldsymbol{svm_{i,j}}$ biner yang merupakan *decider* pada node tersebut atau dengan label kelas jika node berhenti berproses secara rekusif. Bagian terpenting dari proses DCSVM-*Subtree* adalah pemilihan svm yang optimal berdasarkan prediksi *likelihood*. Oleh karenanya, digunakan indeks *impurity, balance,* dan *score* seperti pada persamaan 3.3 dan 3.4 untuk megetahui kualitas dari klasifikasi dengan pengukuran indeks tersebut kemungkinan dapat berpengaruh terhadap bentuk dan presisi dari *decission tree*. Jika *score* yang digunakan, maka indeks *impurity* dan *balanced* dapat memisahkan cabang-cabang pohon lebih banyak sehingga membuat bentuk *decission tree* terlihat lebih "lebat" namun jika hal ini dilakukan sebaliknya maka, *decission tree* akan memberikan hasil yang lebih baik.

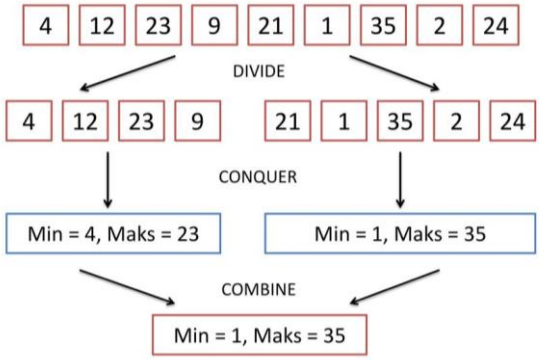
4. **Algoritma 2** *classifier* DCSVM bergantung pada *dcsvm decission tree* yang merupaka hasil dari algoritma 1. Algoritma 2 dimulai pada keputusan simpul akar pohon yang dihasilkan pada algoritma 1, kemudian setiap node SVM yang berasosiasi, melakukan prediksi hingga node mencapai daun. Label dari node daun merupakan prediksi DCSVM untuk input data. Pada algortima ini klasifikasi multikelas dari setiap data item paling banyak menghasilkan *k* langkah keputusan biner dengan kecepatan waktu peluruhan setidaknya $\boldsymbol{\log_2 k}$

4. Hasil dan Pembahasan

4.1. Pembentukan Model Klasifikasi

Pembentukan model klasifikasi diawali dengan membagi 10.000 data ke dalam dua bagian. Pertama sebanyak 80% dari keseluruhan data, diambil untuk membuat model (data *training*), kemudian sebanyak 20% dari keseluruhan data digunakan untuk evaluasi model (data *testing*). Ide pada penelitian ini adalah menerapkan algoritma *Divide and Conquer (DC)* ke dalam SVM untuk dilakukan proses klasifikasi pada multi kelas.

Terdapat tiga tahap pengerjaan pada algortima *DC*. Tahap pertama adalah membagi masalah kedalam beberapa sub-masalah dimana pada setiap sub-masalah memiliki karkateristik yang sama dengan masalah yang semula namun memiliki ukuran yang lebih kecil tahap ini disebut dengan *divide,* selanjutnya adalah tahap *conquer* pada alur ini diperoleh solusi dari sub-masalah yang berkerja secara berulang-ulang atau rekusif. Tahap ketiga disebut sebagai *combine* pada tahap ini digabungkan solusi dari sub-masalah sehingga membentuk solusi, seperti masalah awal. Ilustrasi algoritma *divide and conquer* tersaji pada gambar 2 yang menggambarkan proses pencarian nilai minimum dan maksimum (*MinMax*) dari suatu data.



Gambar 2. Ilustrasi Algoritma Divide and Conquer

Tahap awal pemodelan klasifikasi menggunakan DCSVM adalah menentukan kelas prediksi *liklihood* dengan melakukan klasifikasi SVM biner, dengan kemungkinan banyaknya *clasifier* SVM sebanyak $\boldsymbol{k(k-1)/2}$ dimana $\boldsymbol{k=4}$ menunjukkan terdapat empat buah kelas, sehingga akan terbentuk sebanyak enam buah *classifier* SVM yang akan digunakan untuk menghitung kelas prediksi *likelihood* dengan menggunakan persamaan 2.1. Diperoleh hasil kelas prediksi *likelihood* yang tersaji pada tabel 2 perhitungan prediksi kelas *likelihood* merupakan realisasi dari ide awal DCSVM yang diilustrasikan pada gambar 1.

Tabel 2. Kelas Prediksi Likelihood dengan Threshold = 5%

Model SVM	Label Kelas			
	Decorative	*Script*	*Serif*	*Sans-Serif*
Decorative vs Script	Decorative=98,9% Script=1.01%	Decorative=0,84% Script=99,16%	Decorative=0,76% Script=99,24%	Decorative=2,0% Script=97,97%
Decorative vs Serif	Decorative=99,3% Serif=0,61%	Decorative=3,52% Serif=96,48%	Decorative=0,4% Serif=99,6%	Decorative=2,9% Serif=97,03%
Decorative vs Sans-Serif	Decorative=98,5% Sans-Serif=1,47%	Decorative=3,32% Sans-Serif=96,68%	Decorative=0,7% Sans-Serif=99,3%	Decorative=1,1% Sans-Serif=98,81%
Script vs Serif	Script=3,39% Serif=96,61%	Script=99,55% Serif=0,45%	Script=0,7% Serif=99,3%	Script=3,17% Serif=96,83%
Script vs Sans-Serif	Script=2,83% Sans-Serif=97,17%	Script=99,01% Sans-Serif=0,99%	Script=1,06% Sans-Serif=98,94%	Script=1,19% Sans-Serif=98,81%
Serif vs Sans-serif	Serif=0,91% Sans-Serif=99,09%	Serif=1,68% Sans-Serif=98,32%	Serif=98,89% Sans-Serif=1,11%	Serif=0,69% Sans-Serif=99,31%

Hasil perhitungan pada tabel 2 akan digunakan untuk menentukan kelas mana yang akan menjadi *root* dan *leaf node* yang ditentukan berdasarkan nilai *impurity* dan sensitifitas tertinggi serta nilai *balance* terendah dengan menggunakan persamaan 2.3 sampai 2.5.

- Menghitung nilai *impurity* dengan persamaan 2.3 untuk *classifier* SVM *decorative-vs-Serif* dengan $\boldsymbol{\theta}$ sebesar 5%
- Diketahui χ_θ adalah fungsi :

$$\chi_\theta(x) = \begin{cases} 1 & \text{Jika } x > \theta \\ 0 & \text{JIka } x \leq \theta \end{cases}$$

Berdasarkan tabel 2, maka diperoleh

$$\boldsymbol{\mathcal{P}_{\text{decorative,serif}}(0.05) = \big((1+0)+(0+1)+(0+1)+(0+1)\big) - 4 = 0}$$

$$\boldsymbol{\mathcal{B}_{\text{decorative,serif}}(0.05) = \min\big(4-(0+1+1+1)\big), \big(4-(1+0+0+0)\big)}$$

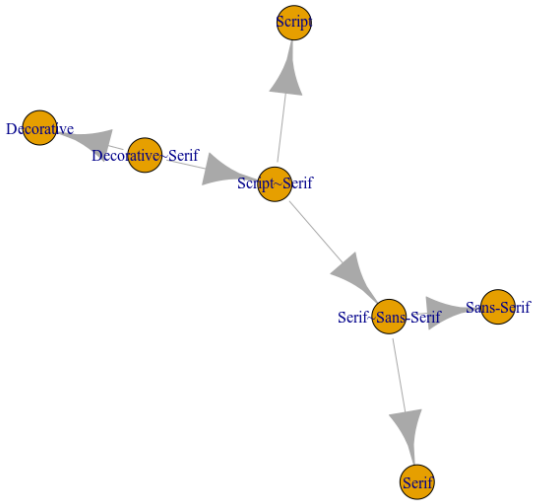
Sehingga, nilai *impurity* dan *balance* untuk *classifier* SVM *decorative-vs-serif* berturut-turut adalah 0 dan 1. Hasil hitung nilai *impurity, balance,* dan sensitifitas untuk seluruh $\boldsymbol{\chi_\theta}$ tersebut tersaji pada tabel 3.

Tabel 3. Nilai *Impurity, Balance,* dan Sensitivitas

Model SVM	***Impurity***	***Balance***	**Sensitivitas**
Decorative vs. Script	0	1	0.99075
Decorative vs. Serif	0	1	0.99495
Decorative vs. Sans-Serif	0	1	0.9867
Script vs. Serif	0	1	0.99425
Script vs. Sans-Serif	0	1	0.9891
Serif vs. Sans-Serif	0	1	0.991

ISSN 2686-4797
Jurnal Restikom : Riset Teknik Informatika dan Komputer
Vol. 5, No. 3, Desember 2023, pp. 340-351

Berdasrkan tabel 3, kita dapat menentukkan *root* pada *tree* yang kemudian bisa kita gunakaan untuk melakukan pemodelan DCSVM. Nilai *impurity* dan *balance* pada ke enam *classifier,* memiliki nilai yang sama dengan tingkat sensitivitas yang berbeda. Berdasarkan nilai sensitivitas paling tinggi, *classifier Decorative*-vs-*Serif* dipilih menjadi *root node* hal ini merupakan tahap *divide* yang dimaksud pada algoritma *DC*. Untuk pemilihan *leaf node,* dilihat dari nilai *impurity, balance,* dan sensitivitas dari *classifier* pada kombinasi label kelas yang mungkin berdasarkan tabel 2 diperoleh label kelas *decorative* mengisi *leaf node* sebelah kiri dan label kelas *script, serif* dan *sans serif* mengisi *leaf node* sebelah kanan dimana untuk *leaf node* pertama diisi oleh *classifier script vs serif*. Untuk menentukan kandidat kelas mana yang akan menjadi *leaf node* selanjutnya ditentukan berdasarkan tabel 3 dengan melihat label kelas yang mungkin menjadi kombinasi *classifier* tanpa mempertimbangkan kolom dengan label kelas *decorative*. Sehingga diperoleh label kelas yang mungkin adalah *script* mengisi *leaf node* sebelah kanan dan *leaf node* sebelah kiri diisi oleh *classifier serif vs sans serif* dan menghasilkan *leaf node* dengan label kelas *serif* di sebelah kiri dan *sans serif* di sebelah kanan. Gambar 3 menunjukkan hasil *tree* DCSVM sebagai hasil akhir dari algoritma DCSVM



Gambar 3. DCSVM *Tree*

Berdasarkan Gambar 3 diperoleh skenario dalam pengklasifikasian jenis font ke dalam *family font,* ketika jenis huruf yang baru hendak di klasifikasikan ke dalam *family font* tertentu, maka klasifikasi akan mengikuti *leaf node* berdasarkan kesamaan pada karakteristik pada setiap *clasifier* pada *leaf node* nya. Ketika terdapat suatu jenis huruf yang hendak diklasifikasikan dengan model yang diperoleh DCSVM, maka akan diproses melalui *classifier family font decorative-vs-serif* dengan skenario sebagai berikut :

1. Hasil klasifikasi pada *root node* ditentukan dengan, nilai prediksi *likelihood* ($C_{i,j}(l,i)$) dengan ketentuan, suatu jenis huruf akan diklasifikasikan kedalam *family font decorative,* jika $C_{i,j}(l,i) \geq 1-\theta$ dan diklasifikasikan kedalam *family font serif* jika $C_{i,j}(l,j) \geq 1-\theta$ selain itu maka akan diklasifikasikan sebagai "*undecideced*" yang akan ditinjau melalui Indeks *impurity, balance,* dan sensitivitas. Kemudian, klasifikasi jenis huruf yang diklasifikasikan ke dalam *family font serif,* ditentukan berdasarkan nilai $C_{i,j}(l,i)$ pada *leaf node* dengan *classifier family font script-vs-serif* .
2. Dengan ketentuan yang sama pada saat menentukan kalsifikasi yang dihasilkan pada *root node,* jenis huruf akan diklasifikasikan kedalam *family font script,* jika $C_{i,j}(l,i) \geq 1-\theta$ dan diklasifikasikan kedalam *family font serif* jika $C_{i,j}(l,j) \geq 1-\theta$ dengan memperhitungkan kembali nilai $C_{i,j}(l,j)$ pada *leaf node* dengan *classifier family font serif-vs-sans serif* .
3. Ketentuan jenis huruf yang mengikuti *leaf node* dengan *classifier family font serif-vs-sans serif* . akan diklasifikasikan kedalam *family font serif,* jika $C_{i,j}(l,i) \geq 1-\theta$ dan diklasifikasikan kedalam *family font sans serif* jika $C_{i,j}(l,j) \geq 1-\theta$. Pada tahap inilah sudah tidak terdapat data yang tidak termasuk kedalam *family font.*

Setelah dilakukan pemodelan DCSVM dengan menggunakan *data training*, maka dilakukan validasi model untuk mengetahui seberapa baik model DCSVM yang sudah dibangun dalam mengklasifikasikan jenis huruf ke dalam *family font* dengan melihat akurasi klasifikasi pada *data training* dan melihat matriks konfusi untuk mengetahui seberapa banyak kejadian *miss classification* pada pengklasifikasian jenis huruf dengan menggunakan model DCSVM.

Tabel 4. Matriks Konfusi *Data Training* pada DCSVM dengan *Threshold* 5%

Aktual	Prediksi			
	Decorative	*Sans serif*	*Script*	*Serif*
Decorative	1964	4	4	4
Script	71	4	1939	4
Serif	8	17	8	1953
Sans serif	60	1942	10	8

4.2. Uji Kelayakan Model

Untuk mengetahui, apakah model klasifikasi DCSVM layak digunakan, maka dilakukan proses klasifikasi menggunakan data *testing* pada besaran *threshold* yang berbeda-beda yang kemudian dibandingkan dengan model SVM biner dan *one vs one* (OVO). Dalam penelitian ini, nilai *threshold* (θ) yang digunakan sebesar 2%, 3%, dan 5%. Hasil perbandingan akurasi tersaji pada tabel 5.

Tabel 5. Perbandingan Akurasi SVM Biner dan OVO dengan DCSVM dengan *Threshold* yang Berbeda

OVO	SVM Biner	DCSVM		
		$\theta = 2\%$	$\theta = 3\%$	$\theta = 5\%$
80,38%	79,65	79,15	80,4	82,85

Hasil perbandingan akurasi pada tabel 5 menunjukkan, kalsifikasi dengan DCSVM pada nilai θ dengan *rough grid search* sebesar 2%, 3% dan 5% memiliki akurasi yang berbeda dimana model DCSVM yang dipilih berdasarkan nilai θ optimal yang memiliki kemampuan dalam sparasi data ke dalam kelas tertantu dengan jelas, sehingga dapat memberikan akurasi yang lebih tinggi. Untuk melihat performasi klasifikasi dengan DCSVM maka berdasarkan nilai θ optimal dipilih pada $\theta = 5\%$ disajikan kedalam matriks konfusi, menunjukkan sebesar 82,85 % DCSVM mampu mengklasifikasi jenis huruf kedalam *family font* yang tersaji pada tabel 6.

Tabel 6. Matriks Konfusi *Data Testing* pada DCSVM dengan Threshold 5%

Aktual	Prediksi			
	Decorative	*Sans serif*	*Script*	*Serif*
Decorative	424	95	3	0
Script	15	86	376	4
Serif	1	116	2	393
Sans serif	14	464	4	3

Han dalam Hendarsin (2018) model dengan akurasi tinggi dan memiliki nilai sensitivitas tinggi, dapat dikatakan mampu melakukan klasifikasi dengan baik berdasarkan informasi pada tabel 4.6 maka diperoleh sensitivitas untuk setiap *family font* sebagai tersaji pada tabel 7.

Tabel 7. Sensitivitas DCSVM untuk Setiap Family Font

Family Font	Sensitivitas	
Decorative	$\frac{424}{424+15+1+14} = 0{,}934$	93,4%
Script	$\frac{376}{376+3+2+4} = 0{,}977$	97,7%
Serif	$\frac{393}{393+0+4+3} = 0{,}983$	98,3%
Sans serif	$\frac{464}{464+95+86+116} = 0{,}610$	61,0%

Hasil perhitungan sesitivitas menunjukkan bahwa DCSVM memiliki kinerja yang baik dalam mengklasifikasikan jenis huruf kedalam *family font decorative, script,* dan *serif* yang ditunjukkan melalui nilai sensitivitas di atas 90%. Kemudian, model bekerja relatif kurang baik jika mengklasifikasikan jenis huruf ke dalam *family font sans serif* ditunjukkan dengan nilai sensitivitas yang hanya mencapai 61%.

Setelah dilakukan pemodelan menggunakan algoritma *DC* pada SVM dan diperoleh hasil evaluasi yang baik, maka dilakukan proses klasifikasi jenis huruf yang belum diketahui jenis *family font* nya. jenis huruf ini adalah *italic* dengan observasi sebanyak 3.712. Hasil klasifikasi menggunakan DCSVM menunjukkan sebanyak 16 observasi dari jenis huruf *italic,* termasuk ke dalam *family font decorative* dan sebanyak 3.696 termasuk ke dalam *family font sans serif*. Sehingga, dapat disimpulkan bahwa jenis huruf *italic* merupakan *family font* dari *sans serif*.

4.3 Pembahasan

Hasil perhitungan diperoleh nilai akurasi model DCSVM pada *data training* dengan menggunakan *threshold* (θ) sebesar 5%, adalah 97,48% artinya model DCSVM mampu mengklasifikasikan jenis huruf ke dalam *family font* dengan keakuratan yang cukup tinggi penggunaan model ini berlainan dengan teknik yang digunakan pada penelitian terdahulu yang hanya mengkombinasikan proses ektraksi fitur dan tidak melakukan klasifikasi multikelas [6],[8] permasalahan *sparsity* dalam penelitian ini diatasi berdasarkan algoritma *DC* hal ini akan berbeda dengan hasil penelitian pada [8] sebab pada penelitian ini lebih fokus menangani masalah *sparsity* yang disebabkan esktraksi fitur namun penelitian terdahulu lebih memilih konsep pemilihan fitur saja. Hal ini dapat juga dilihat melalui matriks konfusi yang tersaji pada tabel 6 dimana secara umum, kejadian *miss classification* pada jenis huruf ke dalam *family font* yang dilakukan dengan model DCSVM cukup rendah. Sehingga dapat disimpulkan bahwa, model DCSVM mengklasifikasikan jenis huruf ke dalam *family font* sesuai dengan label kelas *family font* pada data asli.

5. Penutup

5.1 Kesimpulan

DCSVM dapat memberikan model klasifikasi yang baik ditunjukkan dengan hasil akurasi yang diperoleh dari klasifikasi DCSVM lebih baik dari SVM biner dan SVM *one vs one*. Model DCSVM yang dipilih berdasarkan nilai θ optimal yang memiliki kemampuan dalam sparasi data ke dalam kelas tertantu dengan jelas, sehingga dapat memberikan akurasi yang lebih tinggi. Sehingga, penelitian ini menunjukkan bahwa model klasifikasi dengan DCSVM pada *threshold* optimal baik digunakan untuk melakukan klasifikasi jenis font ke dalam *family font*.

5.2 Ucapan Terimakasih

Terimakasih peneliti ucapkan kepada Kementrian Pendidikan, Kebudayaan, Riset, dan Teknologi Indonesia yang telah memberikan bantuan dana sehingga penelitian ini dapat terlaksana dengan baik dan sebagaimana mestinya.

Daftar Pustaka

[1] Allison B., Guthrie D., Gutrhie L. Another Look at the Data Sparsity Problem. International Conference on Text, Speech and Dialogue. TDS Springer [internet]. 2006 :327-334. Available from: https://doi.org/ 10.1007/11846406_41.

[2] Arshiya S. Ansari, et.al. "Improved Support Vector Machine and Image Processing Enabled Methodology for Detection and Classification of Grape Leaf Disease", Journal of Food Quality. 2022; vol. 2022(2). Article ID 9502475. Available from: https://doi.org/10.1155/2022/9502475

[3] Chauduri A., Badelia P., Ghost S.K., Mandaviya K. Optical Character Recognitions Systems for Different Languages with Soft Computing. Studies in Fuzziness and Soft Computing. Springer [internet]. 2017; 352. Available from: http://doi.org/10.1007/978-3-319-50252-6_1.

[4] Don, R Duleep. DCSVM : Fast Multiclass Classification using Support Vector Machine. International Journal of Machine Learning and Cybermetics. Springer [internet]. 2019; 11 (2): 433 – 447. Available from: http://doi.org/10.1007/s13042-019-00984-9.

[5] Maulana P., Irhamah. Klasifikasi Kabupaten di Provinsi Jawa Timur Berdasarkan Indikator Daerah Tertinggal dengan Metode Support Vector Machine (SVM) dan Entropy Based Fuzzy Support Vector (ESVM). INFERENSI. 2018; Vol.1 (1). Available from: http://doi.org/10.12962/j27213862.v1i1.6715.

[6] Padole Vishay, Mankar Vijay. 2019. Image Classification by Using Multiclass Support Vector Machines. WSEAS TRANSACTIONS on COMPUTER RESEARCH. 2019;Vol.7.

[7] Perez A., Gracia S., Marin H., Coello CAC, Herrera F. Mc2esvm : Multiclassification based on Coorperative Evolution of Support Vector Machines. IEEE Computat Intell Mag. 2018; 13(2):18-29.Available From: http://doi.org/10.1109/MCI.2018.2806997

[8] Phangtriasu M.R., Hareta J., Tanoto D.F. Comparation Between Neural Network and Support Vector Machine in Optical Character Recognition. 2nd International Conference Computer Science and Computational Intelligence, Bali. 2017; 351-357. Available from: http://doi.org/10.1016/j.procs.2017.10.061.

[9] Tantika Silviany R. Penggunaan Metode Support Vector Machine Klasifikasi Multiclass pada Data Pasien Penyakit Tiroid. Bandung Conference Series : Statistics. 2022; Vol.2 (2). Available From: http://doi.org/10.29313/bcss.v2i2.3590.

[10] Vapnik, N, Vladimir. The Nature of Statistical Learning Theory. Springer [internet]. 2000; 2: XX : 314. Available from: https://doi.org/10.1007/978-1-4757-3264-1