TERAKREDITASI RISTEKDIKTI No. 36b/E/KPT/2016

Jurnal Rekayasa Elektrika

VOLUME 15 NOMOR 1 April 2019

JRE	Vol. 15	No. 1	Hal 1–74	Banda Aceh, April 2019	ISSN. 1412-4785 e-ISSN. 2252-620X

Jurnal Rekayasa Elektrika Vol. 15, No. 1, April 2019, hal. 7-17
ISSN. 1412-4785; e-ISSN. 2252-620X, Terakreditasi RISTEKDIKTI No. 36b/E/KPT/2016
DOI: 10.17529/jre.v15i1.13298

Grid SVM: Aplikasi Machine Learning dalam Pengolahan Data Akuakultur

Oskar Natan, Agus Indra Gunawan, dan Bima Sena Bayu Dewantara
Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jalan Raya ITS Sukolilo, Surabaya 60111
e-mail: theoskarnatan@gmail.com

***Abstrak*—Kondisi air merupakan faktor utama yang mempengaruhi tingkat kesuksesan dalam akuakultur khususnya pada budidaya udang. Akan tetapi, petani sering mengalami kesulitan dalam menentukan kondisi tersebut yang didasarkan pada hasil pengukuran berbagai parameter air. Oleh karena itu, diperlukan sebuah model klasifikasi yang tepat untuk membantu petani dalam mengklasifikasi kondisi air di tambak. Dengan mengetahui kondisi tersebut, maka *treatment* yang tepat dan sesuai dapat diberikan. Pada penelitian ini, algoritma *machine learning* bernama *support vector machine* (SVM) digunakan untuk membuat model dari sebuah dataset akuakultur. Teknik pemrosesan lain seperti normalisasi data dan penggunaan algoritma optimasi bernama *grid search* juga dilakukan untuk meningkatkan hasil pemodelan. Selanjutnya, sebuah skema pengujian dengan *k-fold cross validation* dilakukan untuk mengetahui performa dari model yang diukur berdasarkan nilai akurasi, presisi, *recall, f-measure*, dan *area under receiver operating characteristic curve* (AUROC). Kemudian, model dari SVM ini dibandingkan dengan model yang dibentuk dengan algoritma yang lain yaitu *K-nearest neighbor* (KNN), *random forest* (RF), logistic regression (LR), *complement naive bayes* (CNB), dan *multi-layer perceptron* (MLP) guna mengetahui model yang paling bagus untuk diimplementasikan pada proses budidaya. Dari hasil percobaan, model yang dibentuk dengan SVM dan optimasi grid search memiliki performa paling tinggi saat proses validasi dengan skor performa keseluruhan sebesar 3,54383.**

Kata kunci: ***machine learning, SVM, grid search, akuakultur***

***Abstract*—Water condition is the main factor that affects the success rate of aquaculture, especially in shrimp cultivation. However, the farmer often experiences difficulties in determining the condition which is stated based on the measurement of various water parameter. Therefore, a proper classification model is needed to help the farmer in classifying the water condition in a pond. By knowing the condition, then proper and correct treatment can be given. In this research, a machine learning algorithm called SVM is used to make a model from an aquaculture dataset. Another processing technique like data normalization and the usage of optimization algorithm named grid search is also performed to improve the modeling result. Furthermore, a test scheme with using k-fold cross-validation is performed to know the performance of the model which is measured by the value of accuracy, precision, recall, f-measure, and AUROC. Then, the SVM model is compared with several models which are made by using another machine learning algorithm such as KNN, CNB, RF, MLP, and LR in order to know the best model to be implemented on cultivation process. From the experiment results, the model which is made with SVM and grid search optimization has the best performance in the validation process with the performance score of 3.54383.**

Keywords: ***machine Learning, SVM, grid search, aquaculture***

I. PENDAHULUAN

Dalam dunia akuakultur, terdapat banyak faktor yang mempengaruhi kesuksesan dalam berbudidaya, khususnya pada budidaya udang. Salah satu faktor yang paling berpengaruh dalam budidaya perairan adalah kondisi air tambak yang menjadi habitat bagi tumbuh-kembang udang. Selain berpengaruh terhadap perkembangan udang, kondisi air juga menentukan tingkat *survival rate* udang. Hal ini dikarenakan udang sangat rentan terserang penyakit apabila kondisi air tambak di bawah kondisi standar untuk budidaya [1]. Oleh karena itu, mengetahui kondisi air tambak merupakan hal yang wajib bagi para petani. Akan tetapi, salah satu tantangan dari penentuan kondisi air tambak adalah pertimbangan terhadap berbagai parameter air yang cukup banyak. Selain itu, nilai parameter air seperti pH dan kadar oksigen terlarut cenderung untuk tidak berubah secara signifikan ketika kondisi air berubah dari baik ke buruk atau sebaliknya [2]. Bahkan kondisi air juga bisa dikatakan buruk meskipun beberapa parameter air dalam kondisi yang normal atau masih dalam batas aman untuk budidaya. Dengan demikian, petani sering mengalami kesulitan dalam mengklasifikasi kondisi air tambak berdasarkan pengukuran terhadap beberapa parameter air. Hal ini bisa menjadi sangat berbahaya jika petani salah mengambil tindakan atau salah melakukan

Received 11 March 2019; Revised 30 March 2019; Accepted 09 April 2019

treatment pada tambak mereka.

Berdasarkan paparan permasalahan dan tantangan tersebut, maka diperlukan sebuah model klasifikasi yang dapat membantu petani dalam mengklasifikasi kondisi air tambak. Model ini bekerja dengan mengestimasi kondisi air berdasarkan input pembacaan parameter air. Dengan kata lain, model klasifikasi ini memproses berbagai input data berupa data-data pengukuran parameter air tambak dan menghasilkan output berupa estimasi klasifikasi dari kondisi air tambak tersebut. Adapun parameter-parameter air yang diperhitungkan dalam pemrosesan adalah pH, alkalinitas, *dissolved oxygen* (DO), *total organic matter* (TOM), NH_4, NH_3, NO_2, NO_3, PO_4, *NP ratio*, salinitas, transparansi air, dan tinggi air. Dengan adanya bantuan dalam hal klasifikasi ini, diharapkan petani dapat mengambil tindakan atau *treatment* yang tepat untuk tambak mereka. Dengan demikian, yang menjadi tantangan dalam penelitian ini adalah bagaimana cara membentuk model klasifikasi yang bagus untuk digunakan oleh para petani udang.

Terkait dengan pembuatan model klasifikasi tersebut, pada penelitian ini diajukan sebuah algoritma *machine learning* bernama SVM untuk membuat model dari sebuah dataset akuakultur. Selain itu, beberapa teknik dalam *data preprocessing* juga diaplikasikan untuk mengolah data terlebih dahulu sebelum dimodelkan. SVM memiliki beberapa parameter yang menentukan seberapa baik algoritma tersebut bekerja. Oleh karena itu, pada penelitian ini juga digunakan sebuah algoritma optimasi yang disebut dengan *grid search* untuk men-*tuning* parameter-parameter yang ada pada SVM. Sebagai evaluasi tambahan, model yang dibentuk SVM ini juga dibandingkan dengan model yang dibentuk dari algoritma *machine learning* yang lain untuk menentukan model mana yang paling bagus. Sebuah model dikatakan bagus apabila model tersebut memiliki performa yang baik saat diuji pada sebuah dataset pengujian. Adapun parameter performa yang digunakan untuk mengukur seberapa baik suatu model klasifikasi adalah nilai akurasi, presisi, *recall*, *f-measure* dan luasan AUROC yang akan dijelaskan pada bab berikutnya. Dari berbagai penjelasan di atas, maka keterbaruan dari penelitian ini adalah studi analisa tentang penggunaan teknik *data preprocessing*, algoritma *machine learning*, dan algoritma optimasi dalam pembuatan model klasifikasi air tambak pada bidang akuakultur khususnya untuk budidaya udang.

SVM telah digunakan dalam beberapa penelitian untuk membuat model estimasi atau klasifikasi dari suatu data. Pada [3], model SVM digunakan untuk memprediksi atau mengestimasi suhu dari *multicore processors* berdasarkan jumlah konsumsi daya selama beroperasi. Selain itu, SVM juga dapat diaplikasikan pada klasifikasi gambar seperti yang dilakukan dalam penelitian [4]. Penelitian terkait dengan optimasi SVM juga dilakukan pada [5] dan diaplikasikan untuk diagnosa medis. Sementara pada penelitian ini, SVM digunakan untuk memprediksi kondisi air tambak berdasarkan beberapa input nilai parameter air. Adapun kernel SVM yang dipakai dalam penelitian ini adalah kernel *radial basis function* (RBF) dan dioptimasi dengan algoritma *grid search* untuk menemukan nilai *c* dan γ paling optimal. Untuk meningkatkan performa pemodelan data, teknik *data preprocessing* seperti normalisasi juga diaplikasikan dan dievaluasi.

II. Studi Pustaka

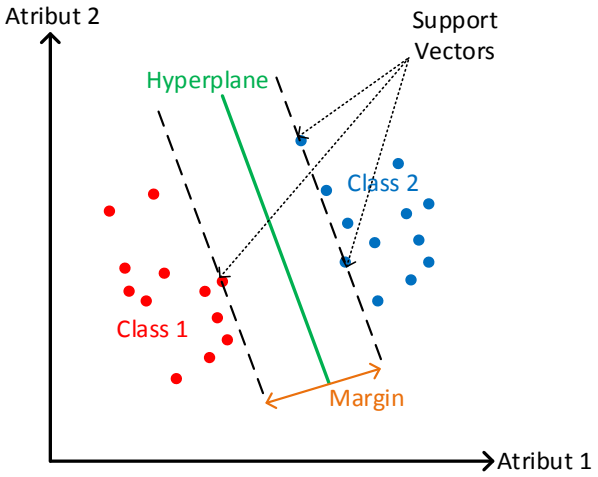
Seperti yang dijelaskan pada bab sebelumnya, pada penelitian ini digunakan sebuah algoritma *machine learning* yaitu *support vector machine* (SVM) untuk membentuk model klasifikasi dari dataset akuakultur dalam kasus *supervised learning*. SVM membentuk model klasifikasi dengan menemukan *hyperplane* yang bisa memaksimalkan margin dari 2 buah *class* dalam dataset [6]. Ilustrasi dari *hyperplane* yang memisahkan 2 buah *class* dalam data dapat dilihat pada Gambar 1.

Titik-titik warna merah dan biru adalah plot dari sampel data berdasarkan nilai atribut 1 dan atribut 2 dengan warna berbeda untuk masing-masing *class*. Kedua *class* ini dipisahkan oleh garis *hyperplane* berwarna hijau yang letaknya tepat di tengah margin. Lebar margin ini bergantung dari letak *support vectors* yang tidak lain adalah sampel yang letaknya paling berdekatan di antara 2 *class*. Dengan menggunakan konsep ini, SVM juga bisa diimplementasikan untuk dataset berdimensi tinggi (memiliki *feature vectors* yang banyak) atau dataset yang memiliki lebih dari 2 atribut. Pada contoh ilustrasi dalam Gambar 1, fungsi untuk membentuk garis *hyperplane* dapat diperoleh dengan menggunakan persamaan garis lurus seperti pada Persamaan (1),

$$y = mx + c, \tag{1}$$

dimana y adalah fungsi garis x, nilai atribut sampel, m koefisien *gradient* garis, dan koefisien bias garis.

Akan tetapi dalam permasalahan sebenarnya, data seringkali tidak mengelompok sesuai dengan *class*-nya dengan teratur sehingga sebuah garis lurus tidak bisa memisahkan 2 buah *class* dengan baik. Oleh karena itu, dibuatlah sebuah kernel untuk membentuk garis yang bisa melengkung-lengkung sesuai dengan sebaran data yang ada. Salah satu kernel SVM yang digunakan dalam



Gambar 1. Plot data dan *hyperplane* SVM

Tabel 1. *Cartesian product* pada *grid search*

Himpunan (Parameter)	γ			
c	Anggota Himpunan	j_1	j_2	j_n
	i_1	(i_1, j_1)	(i_1, j_2)	(i_1, j_n)
	i_2	(i_2, j_1)	(i_2, j_2)	(i_2, j_n)
	i_n	(i_n, j_1)	(i_n, j_2)	(i_n, j_n)

penelitian ini adalah kernel RBF [7]. Adapun fungsi kernel tersebut dapat dilihat seperti pada Persamaan (2) dan Persamaan (3).

$$y = k(x_1, x_2) + c \,, \tag{2}$$

$$k(x_1, x_2) = \exp\left(-\gamma \left\| x_1 - x_2 \right\|^2\right), \tag{3}$$

dimana k adalah fungsi kernel RBF, x_1 nilai atribut (*feature vectors*) sampel 1, x_2 nilai atribut (*feat* nilai atribut (*feature vectors*) sampel 2, γ koefisien gamma, c koefisien bias garis *hyperplane*, dan adalah kuadrat jarak *Euclidean* dari 2 nilai atribut dari kedua sampel.

Dari Persamaan (2) dan Persamaan (3), telah diketahui bahwa terdapat 2 buah parameter yang perlu di-*tuning* untuk menghasilkan garis *hyperplane* yang paling optimal yaitu koefisien c dan γ. Nilai c digunakan untuk kompensasi terhadap garis *hyperplane* SVM agar berada pada posisi yang tepat dalam memisahkan 2 buah *class* atau lebih. Sementara nilai γ dapat diibaratkan sebagai faktor pengali untuk jarak dari 2 buah sampel. Untuk mendapatkan nilai yang tepat atau optimal untuk kedua koefisien tersebut, maka diperlukan sebuah algoritma optimasi. Pada penelitian ini sebuah algoritma bernama *grid search* digunakan untuk menemukan nilai yang tepat untuk mengisi koefisien c dan γ. *Grid search* bekerja dengan mencacah nilai dalam batasan *range* tertentu lalu menginputkan segala kombinasi angka yang dimungkinkan untuk mengisi nilai c dan γ [8]. Sebagai contoh, dalam sebuah percobaan dibuat *range* nilai 1 sampai 200 untuk parameter c dan 0,01 sampai 1 untuk parameter γ. Kemudian masing-masing nilai dicacah setiap 1 dan 0,01 sehingga himpunan nilai untuk c adalah {1,2,3,…,200} dan γ adalah {0,01, 0,02, 0,03, …,1}. Masing-masing himpunan memiliki 200 dan 100 anggota nilai, sehingga jumlah kombinasi yang dapat dibuat adalah 20000 kombinasi. Kombinasi angka yang menghasilkan performa terbaiklah yang dipilih untuk menjadi nilai dari kedua koefisien tersebut. Secara garis besar, *grid search* adalah fungsi *cartesian product* seperti pada Persamaan (4).

$$c \,.\, \gamma = \{(i, j) | i \in c \quad dan \quad j \in \gamma\}, \tag{4}$$

dimana i anggota dari himpunan c, j anggota dari himpunan γ. Adapun ilustrasi pencacahan nilai dan *cartesian product* dapat dilihat pada Tabel 1.

Tabel 2. Spesifikasi dataset akuakultur

Spesifikasi	
Jenis Permasalahan	Klasifikasi
Jumlah Atribut Input	13
Jumlah Atribut Output	1 (*Class*)
Jumlah Sampel	174
Tipe Data	*Binomial* dan Bilangan
Missing Value	Ya

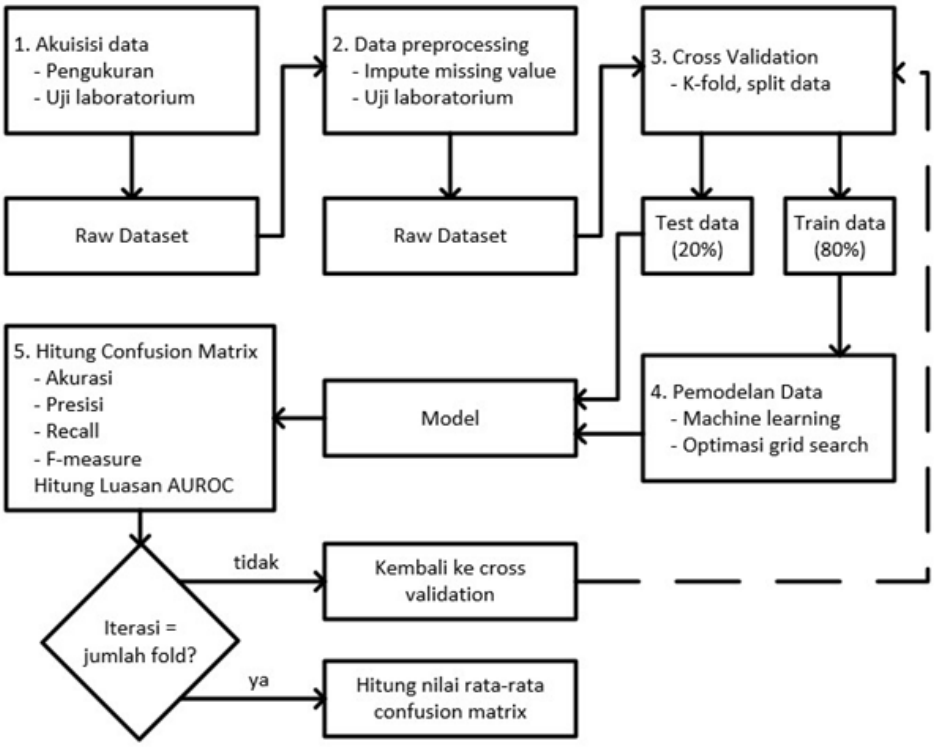
III. METODE

Penelitian ini bertujuan untuk mendapatkan model terbaik untuk klasifikasi kondisi air tambak. Oleh karena itu, dilakukan evaluasi performa pada berbagai macam algoritma *machine learning* dalam proses pemodelan data akuakultur. Untuk mendapatkan hasil yang optimal maka diperlukan beberapa tahapan mulai dari proses pengambilan data hingga proses pemodelan data. Pada bab ini dijelaskan bagaimana penelitian dilakukan dan penjelasan secara detail pada setiap tahapan penelitian. Adapun proses akuisisi hingga pemodelan data dan validasi model *machine learning* dapat diringkas sesuai dengan blok diagram pada Gambar 2.

A. Dataset

Dalam persoalan *supervised learning* maka diperlukan sebuah dataset untuk dimodelkan dengan suatu algoritma *machine learning* tertentu. Oleh karena itu, dalam penelitian ini digunakan sebuah dataset akuakultur yang diakuisisi dari beberapa tambak udang Vannamei di daerah Bulukumba, Sulawesi Selatan. Adapun spesifikasi dari dataset ini dapat dilihat pada Tabel 2.

Data ini berisi tentang kondisi air tambak berdasarkan hasil pengukuran 13 parameter air (atribut input). Seperti yang dijelaskan pada bab sebelumnya, parameter-parameter tersebut meliputi pH, alkalinitas, DO, TOM, NH_4, NH_3, NO_2, NO_3, PO_4, *NP ratio*, salinitas, transparansi air, dan tinggi air. Pengukuran parameter-parameter air ini dilakukan dengan menggunakan sensor untuk besaran fisis dan juga berdasarkan hasil uji lab untuk zat-zat kimia tertentu. Pada pengambilan data parameter fisis air, sensor air dicelupkan langsung ke dalam tambak. Sementara untuk mengukur bahan kimia terlarut, pengukuran dilakukan dengan mengambil sampel air dan dibawa ke laboratorium. Sedangkan untuk kondisi air tambak (atribut output) diklasifikasikan menjadi 2 kategori yaitu kondisi baik (1) dan buruk (0). Adapun pelabelan klasifikasi kondisi air tambak ini didasarkan pada justifikasi para *expert* di bidang akuakultur yang menilai kondisi tambak secara langsung. Data diakuisisi dari 14 tambak yang berada dalam 1 wilayah yang sama selama proses budidaya udang berlangsung. Pengambilan data dilakukan setiap hari sehingga didapatkan sampel data 174 sampel. Pada dataset ini juga terdapat beberapa data *missing* yang disebabkan kerusakan sensor maupun *human error*.



Gambar 2. Diagram proses pemodelan *machine learning*

Tabel 3. Penjelasan parameter air tambak

No.	Parameter	Penjelasan	Satuan / Range
1	PH	Kadar PH dalam air tambak	0 – 14
2	Alkalinitas	Jumlah HCO3 (Bikarbonat) dan CO3 (Karbonat)	part per million (ppm)
3	DO	Jumlah Oksigen terlarut dalam air	(ppm)
4	TOM	Total organic matter dalam air	(ppm)
5	NH4	Jumlah ammonium dalam air	(ppm)
6	NH3	Jumlah ammonia dalam air	(ppm)
7	NO2	Jumlah nitrogen dioksida dalam air	(ppm)
8	NO3	Jumlah nitrat dalam air	(ppm)
9	PO4	Jumlah fosfat dalam air	(ppm)
10	NP Ratio	Ratio nitrat dan fosfat dalam air	-
11	Salinitas	Tingkat keasinan air	part per thousand (ppt)
12	Transparansi Air	Jarak pandang air dari permukaan tambak	centimeter (cm)
13	Tinggi Air	Ketinggian air tambak	(cm)
14	Kondisi Air	Klasifikasi kondisi air tambak	baik (1) dan buruk (0)

Adapun penjelasan untuk setiap parameter air tambak dalam atribut input dan atribut output pada data akuakultur tersebut dapat dilihat pada Tabel 3.

B. Data Preprocessing

Seperti dijelaskan pada sub bab sebelumnya, dataset yang digunakan dalam penelitian ini terdapat *missing value* di beberapa sampel. Selain itu masing-masing atribut juga memiliki sebaran nilai yang berbeda. Jika kedua hal tersebut diabaikan, maka akan mengakibatkan buruknya hasil pemodelan data [9]. Oleh karena itu, diperlukan suatu teknik *preprocessing* untuk mengisi nilai yang *missing* dan menyamakan sebaran nilai data pada setiap atribut. Secara umum, tujuan dari implementasi *preprocessing* adalah untuk mempersiapkan dataset sebelum diproses lebih lanjut dengan algoritma *machine learning*. Pada penelitian ini digunakan teknik *averaging* dimana nilai yang ada pada suatu atribut dirata-rata untuk mengisi nilai yang *missing* pada atribut yang bersangkutan. Adapun persamaan matematis untuk teknik *averaging* ini dapat dilihat pada Persamaan (5).

$$y_i = \frac{\sum_{n=1}^{N} x_{in}}{N_i}, \quad (5)$$

dimana y_i adalah hasil *averaging* untuk pengisian *missing value* dalam atribut *i*, x_{in} nilai sampel yang tidak *missing* *n* dalam atribut *i,* N_i jumlah sampel yang tidak *missing* dalam atribut *i.*

Sedangkan untuk menyamakan sebaran nilai yang berbeda disetiap atribut, pada penelitian ini digunakan teknik *min-max scaling* dimana setiap nilai data di-*scaling* pada batasan nilai 0 sampai 1. Tujuan dari tahap ini adalah supaya setiap atribut input (parameter air tambak) memiliki pengaruh yang sama terhadap atribut output (klasifikasi kondisi air tambak) saat proses pemodelan dengan algoritma *machine learning*. Adapun persamaan matematis dari teknik *min-max scaling* ini dapat dilihat pada Persamaan (6).

$$x'_{in} = \frac{x_{in} - \min(x_i)}{\max(x_i) - \min(x_i)}, \quad (6)$$

dimana x_{in} adalah nilai dari sampel *n* dalam atribut *i,* hasil scaling dari x_{in}, dan x_i semua nilai sampel yang ada dalam atribut *i*.

C. Pemodelan Data dengan Machine Learning

Setelah tahapan *data preprocessing* dilakukan maka didapatkan sebuah dataset yang sudah siap untuk dimodelkan dengan algoritma *machine learning*. Seperti yang telah dijelaskan, algoritma yang digunakan dalam penelitian ini adalah SVM. Selain itu, juga dilakukan studi perbandingan performa model SVM dengan model dari beberapa algoritma *machine learning* lainnya yaitu KNN [10], RF [11], LR [12], CNB [13], dan MLP [14]. Masing-masing algoritma ini memiliki beberapa parameter yang menentukan performa algoritma dalam proses pemodelan data. Untuk menentukan nilai parameter yang tepat dan menghasilkan performa model yang maksimal, maka dalam penelitian ini digunakan algoritma optimasi bernama *grid search*. Algoritma ini bekerja dengan mencoba seluruh nilai yang ada dalam batasan nominal tertentu [8]. Kemudian nilai terbaik diambil berdasarkan hasil performa terbaik dari proses pemodelan data. Akan tetapi, masing-masing algoritma memiliki jenis parameter yang berbeda sehingga *tuning* yang dilakukan *grid search* juga berbeda. Adapun nilai *tuning* yang digunakan algoritma optimasi *grid search* untuk setiap parameter algoritma *machine learning* yang digunakan dapat dilihat pada Tabel 4.

Tabel 4. Nilai tuning optimasi *grid search*

Algoritma	Parameter	Nilai Tuning	Total
KNN	Jumlah K	1 –>100, step 1	100
SVM	*c*	1 –> 200, step 1	20000
	Gamma (γ)	0,01 –>1, step 0,01	
RF	Jumlah tree (T)	1 –>100, step 1	100
LR	*c*	0,05 –>5, step 0,05	100
CNB	Alpha (α)	0.,01 –>3, step 0,01	300
MLP	Learning rate (L)	0,0001 –> 0,001, step 0,0001	120
	Solver (S)	SGD dan Adam	
	Jumlah neuron (N)	100, 200, dan 300	
	Jumlah iterasi (I)	1000 dan 2000	

Sesuai dengan Tabel 4, maksud dari nilai *tuning* $1 \rightarrow 100$ dengan *step* 1 adalah nilai antara 1 sampai 100 dicacah setiap 1 sehingga terdapat 100 nilai yang akan dicobakan untuk mengoptimasi algoritma yaitu {1, 2, 3, 4, …, 100}, hal ini juga berlaku untuk nilai *tuning* yang lain. Selanjutnya pada algoritma SVM digunakan kernel RBF sehingga terdapat 2 buah parameter yang perlu dioptimasi yaitu *c* dan γ. Sedangkan untuk algoritma MLP digunakan 2 jenis *solver* yang berbeda yaitu *stochastic gradient descent* (SGD) [15] dan *adaptive moment estimation* (Adam) [16]. Fungsi optimasi *grid search* ini seperti *cartesian product* yang dapat dilihat kembali pada Persamaan (4) dan Tabel 1.

D. Validasi Model

Suatu mekanisme pengujian perlu dilakukan untuk mengetahui seberapa baik performa suatu model *machine learning*. Secara umum, performa suatu model diukur saat model digunakan untuk mengklasifikasi sampel pada sebuah *test data*. Hasil pengklasifikasian atau *predicted class* tersebut selanjutnya dibandingkan dengan *ground truth* atau *actual class* yang ada di atribut output dalam suatu *test data*.

Dengan demikian didapatkan jumlah *true positive, false positive, true negative,* dan *false negative* seperti yang ditunjukkan oleh *confusion matrix* pada Tabel 5. Dengan mengetahui *confusion matrix* tersebut, maka dapat dihitung nilai akurasi, presisi, *recall*, *f-measure*, dan luasan AUROC yang menunjukkan tingkat baik-buruknya suatu model klasifikasi. Perbedaan akurasi, presisi, *recall, f-measure,* dan AUROC serta teknik perhitungan kelima nilai parameter performa tersebut dapat dilihat pada Persamaan (7) sampai dengan persamaan (14) [17]-[19]. Secara matematis, akurasi dapat dihitung dengan menggunakan Persamaan (7),

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}, \quad (7)$$

secara matematis, presisi dapat dihitung dengan menggunakan Persamaan (8),

$$Presisi = \frac{TP}{TP + FP}, \quad (8)$$

secara matermatis, *recall* dapat dihitung dengan menggunakan Persamaan (9),

Tabel 5. *Confusion matrix*

Class (Atribut output dalam *dataset*)		*Actual Class*	
		Baik (1)	Buruk (0)
Predicted Class	Baik (1)	*True Positive* (TP)	*False Positive* (FP)
	Buruk (0)	*False Negative* (FN)	*True Negative* (TN)

Tabel 6. Ilustrasi *K-fold cross validation*

Dataset	Iterasi 1	Iterasi 2	Iterasi 3	Iterasi 4	Iterasi 5
Bag. 1	O	X	X	X	X
Bag. 2	X	O	X	X	X
Bag.3	X	X	O	X	X
Bag.4	X	X	X	O	X
Bag.5	X	X	X	X	O

Keterangan:
O : test data
X : data untuk pemodelan (train data)

$$Recall = TPR = \frac{TP}{TP + FN}, \quad (9)$$

secara matematis, *f-measure* dapat dihitung dengan menggunakan Persamaan (10),

$$f - measure = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}. \quad (10)$$

Berbeda dengan keempat nilai parameter akurasi, presisi, *recall*, dan *f-measure*, nilai AUROC didapatkan dengan menghitung luasan area di bawah kurva *receiver operating characteristic* (ROC). Dengan kata lain besar nilai luasan AUROC ini bergantung dari bentuk lengkungan kurva ROC itu sendiri. Untuk mendapatkan kurva ROC tersebut, 2 hal yang perlu dikalkulasi terlebih dahulu adalah nilai *true positive rate* (TPR) dan *false positive rate* (FPR). TPR juga disebut sebagai *recall* sehingga untuk mengkalkulasi nilai TPR dapat menggunakan Persamaan (9). Sedangkan untuk mengkalkulasi FPR dapat menggunakan rumus seperti yang ditunjukkan pada Persamaan (11),

$$FPR = \frac{FP}{FP + TN}, \quad (11)$$

setelah mendapatkan kedua nilai TPR dan FPR di setiap iterasi *k-fold cross validation*, maka kurva ROC dapat dibentuk dan selanjutnya luasan AUROC dapat dihitung. Pola kurva ROC ini berbeda-beda disetiap algoritma dan model data sehingga mengakibatkan luasan AUROC yang berbentuk sembarang. Oleh karena itu, untuk menghitung luasan AUROC secara diskrit dapat menggunakan pendekatan *trapezoidal rule* yang ditunjukkan oleh Persamaan (12) sampai dengan Persamaan (14),

$$a_i = TPR\left(FPR\left(x_i\right)\right) + TPR\left(FPR\left(x_{i+1}\right)\right), \quad (12)$$

$$h_i = \left|FPR\left(x_{i+1}\right) - FPR\left(x_i\right)\right|, \quad (13)$$

$$\text{AUROC} = \sum_{i=0}^{1} \frac{a_i \times h_i}{2}, \quad (14)$$

dimana α_i meurpakan jumlah sisi sejajar trapesium (*trapezoid*) dan adalah h_i tinggi trapesium. Dengan menggunakan pendekatan *trapezoid rule*, luasan bangun sembarang yang dibentuk di bawah kurva ROC dapat dihitung secara diskrit.

Suatu model *machine learning* dikatakan memiliki performa yang baik dalam melakukan proses klasifikasi sampel data dalam sebuah dataset jika memiliki hasil perhitungan akurasi, presisi, *recall, f-measure,* dan luasan AUROC yang tinggi. Akurasi merepresentasikan seberapa akurat suatu model dalam melakukan klasifikasi. Sementara presisi, *recall*, dan *f-measure* merepresentasikan tingkat kekonsistenan model dalam melakukan klasifikasi. Nilai presisi dan *recall* ini tidak boleh terlalu timpang karena mengindikasikan kecenderungan prediksi model tersebut (lebih banyak ke *class* 1 atau 0) dan inilah yang direpresentasikan dengan nilai *f-measure*. Sementara luasan AUROC mengindikasikan kinerja model dalam melakukan klasifikasi yaitu perbandingan nilai TPR dan FPR atau *rate* prediksi benar dengan yang salah. Semakin besar TPR terhadap FPR maka semakin bagus. TPR yang lebih tinggi dibanding FPR menyebabkan kurva ROC semakin melengkung tinggi dan berdampak pada luasan AUROC yang semakin luas. Sehingga dapat dikatakan bahwa semakin luas AUROC maka mengindikasikan semakin bagus kinerja model tersebut [18]-[19]. Untuk mendapatkan nilai parameter-parameter performa atau indikator kinerja tersebut maka diperlukan sebuah mekanisme validasi terhadap model *machine learning* yang terbentuk.

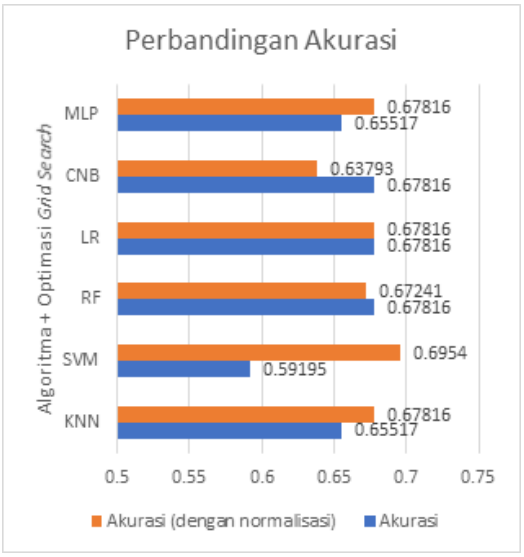
Pada penelitian ini digunakan mekanisme validasi yang disebut dengan *k-fold cross validation* dimana satu dataset utuh dibagi ke dalam sejumlah bagian *k*. Pada *k* bagian tersebut, 1 bagian digunakan sebagai *test data* dan bagian yang lain digunakan untuk pemodelan (*train data*) [20]. Pada iterasi pertama, proses *training* dilakukan dengan menggunakan *train data* dan didapatkan sebuah model yang merepresentasikan *train data* tersebut. Selanjutnya, model itu diuji untuk memprediksi nilai atribut output pada *test data* dan didapatkan hasil perhitungan nilai-nilai parameter performa model tersebut. Kemudian pada iterasi berikutnya, bagian data yang lain digunakan untuk *test data* dan sisanya untuk pemodelan (*train data*). Begitu seterusnya sehingga setiap data pernah menjadi *test data* dan juga data untuk pemodelan (*train data*) [21]. Pada penelitian ini digunakan 5-*fold cross validation* sehingga ada 5 kali iterasi dimana setiap iterasi 20% dataset menjadi *test data* dan 80% sisanya menjadi data untuk pemodelan dengan algoritma *machine learning*. Ilustrasi dari konsep *k-fold cross validation* ini dapat dilihat pada Tabel 6.

IV. Hasil dan Pembahasan

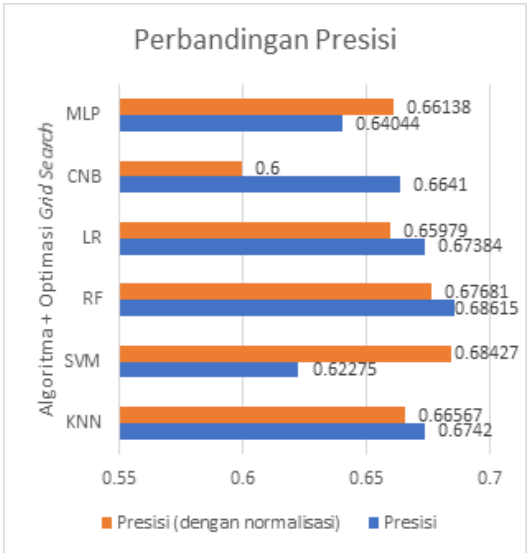
Pada bab ini dijelaskan perbandingan performa dari berbagai algoritma *machine learning* yang digunakan. Komparasi dilakukan dengan membandingkan hasil perhitungan akurasi, presisi, *recall*, *f-measure*, dan luasan AUROC yang didapat dari proses pengolahan data dan validasi dengan mekanisme *5-fold cross validation*. Selain itu, pengaruh normalisasi data terhadap perolehan nilai *confusion matrix* dan AUROC juga diperhitungkan.

A. Perbandingan Akurasi

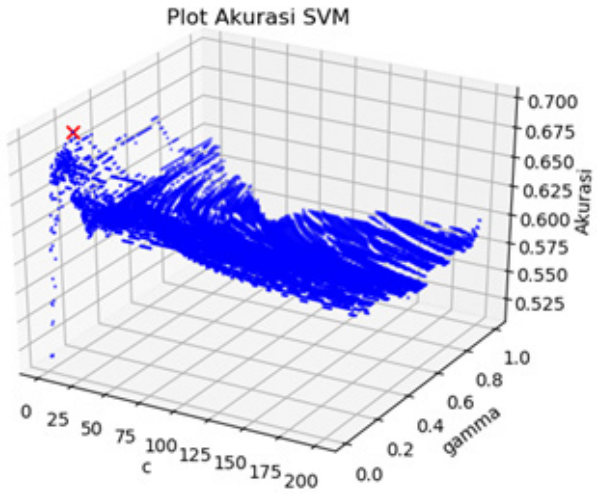
Dalam bidang *machine learning* khususnya pada



Gambar 3. Grafik perbandingan akurasi



Gambar 5. Grafik perbandingan presisi



Keterangan:
. (biru) : bukan akurasi tertinggi
X (merah) : akurasi tertinggi (pada c = 3, γ= 0,17)

Gambar 4. Nilai akurasi SVM pada setiap nilai *tuning*

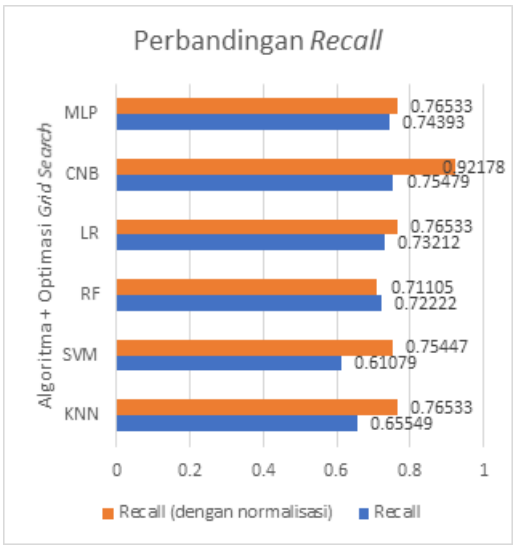
persoalan klasifikasi, suatu model dikatakan memiliki akurasi yang tinggi jika model tersebut dapat memprediksi output dari sejumlah input data dengan benar. Sesuai dengan namanya, nilai akurasi ini merepresentasikan seberapa akurat sebuah model dalam melakukan prediksi suatu *class*. Pada penelitian ini, tingkat akurasi diukur berdasarkan seberapa banyak *predicted class* yang sesuai dengan *actual class* dari sejumlah sampel data dalam *test data*. Sesuai dengan Persamaan (7), akurasi dihitung berdasarkan jumlah prediksi benar yaitu TP + TN dibagi dengan seluruh jumlah prediksi yang dilakukan yaitu TP + TN + FP + FN. Dengan menggunakan mekanisme 5-*fold cross validation*, maka didapatkan 5 nilai akurasi dari pengujian model terhadap 5 *test data* yang berbeda pada setiap iterasi. Oleh karena itu, nilai akurasi diambil dengan melakukan rata-rata terhadap kelima nilai tersebut dan didapatkan akurasi untuk setiap algoritma seperti ditunjukkan pada Gambar 3.

Akurasi tertinggi sebesar 0,6954 diperoleh dengan menggunakan algoritma SVM kernel RBF yang dioptimasi dengan *grid search*. Dengan optimasi *grid search* didapatkan parameter kernel RBF paling optimal yaitu untuk c sebesar 3 dan γ sebesar 0,17. Dari hasil ini dapat dilihat juga bahwa dengan mengaplikasian teknik normalisasi *min-max scaling* pada tahapan *data preprocessing* cukup berpengaruh dalam meningkatkan nilai akurasi. Hal ini terbukti dari skor tertinggi didapatkan dengan mengaplikasikan normalisasi data terlebih dahulu sebelum diproses SVM. Selain itu penggunaan normalisasi juga berpengaruh pada peningkatan akurasi algoritma MLP dan KNN tetapi tidak dengan CNB dan RF. Pada algoritma LR, normalisasi tidak membawa pengaruh terhadap pencapaian akurasi. Adapun plot grafik nilai akurasi SVM kernel RBF pada setiap nilai *tuning grid search* untuk c = 1 sampai dengan 200 dan γ = 0,01 sampai dengan 1 dapat dilihat pada Gambar 4.

B. Perbandingan Presisi

Selain harus mempunyai performa akurasi yang tinggi, sebuah model dikatakan baik dalam mengklasifikasi apabila juga memiliki nilai presisi yang tinggi. Presisi sendiri mengindikasikan tingkat kekonsistenan suatu model dalam melakukan klasifikasi. Pada penelitian ini, presisi didapatkan dengan menghitung jumlah prediksi suatu *class* yang benar atau TP dibagi dengan seluruh jumlah total prediksi yang ditujukan pada *class* tersebut atau TP + FP. Seperti dijelaskan pada sub bab sebelumnya, dengan mekanisme *5-fold cross validation* maka didapatkan 5 nilai presisi pada setiap iterasi validasi. Kemudian nilai presisi akhir diambil dengan merata-rata kelima nilai tersebut. Adapun perbandingan nilai presisi untuk setiap algoritma dan pengaruh normalisasi pada tahap *data preprocessing* dapat dilihat pada Gambar 5.

Nilai presisi tertinggi didapatkan oleh algoritma RF yang dioptimasi *grid search* dengan nilai sebesar 0,68615. Sementara algoritma yang diusulkan dalam penelitian ini yaitu SVM berada di urutan kedua dengan nilai 0,68427. Dengan optimasi *grid search* pada algoritma RF didapatkan



Gambar 6. Grafik perbandingan *recall*

parameter jumlah *tree* paling optimal yaitu sejumlah 28 *trees*. Dari grafik tersebut juga dapat dilihat bahwa normalisasi justru menurunkan nilai presisi pada setiap algoritma kecuali SVM dan MLP. Dengan normalisasi *min-max scaling* pada tahap *data preprocessing*, nilai presisi SVM hampir menyamai presisi RF dimana selisih nilainya hanya sebesar 0,00188.

C. Perbandingan Recall

Sama halnya dengan nilai presisi, nilai yang ditunjukkan *recall* juga mengindikasikan kekonsistenan model dalam melakukan klasifikasi Perbedaannya adalah pada cara menghitungnya yaitu jumlah prediksi suatu *class* yang benar atau TP dibagi dengan jumlah *actual class* yang ada pada *class* yang diprediksi atau dibagi dengan TP + FN. Dengan kata lain, jika presisi membandingkan jumlah total prediksi benar dengan jumlah prediksi yang ditujukan pada *class* yang dituju, maka *recall* membandingkan jumlah prediksi benar dengan jumlah *actual class* yang dituju. Adapun *recall* akhir diambil dengan melakukan *averaging* pada kelima nilai *recall* yang diperoleh dengan mekanisme *5-fold cross validation*. Hasil perbandingan nilai *recall* dari seluruh algoritma yang digunakan dalam penelitian ini dapat dilihat pada Gambar 6.

Sesuai dengan grafik pada Gambar 6, nilai *recall* tertinggi didapatkan dengan menggunakan algoritma CNB yang dioptimasi dengan *grid search*. Dengan optimasi *grid search* didapatkan parameter α paling optimal yaitu sebesar 0,01. Nilai *recall* tertinggi ini juga dipengaruhi oleh normalisasi data sehingga didapatkan nilai sebesar 0,92178. Dengan menggunakan normalisasi pada tahap *data preprocessing*, setiap algoritma mengalami peningkatan nilai *recall* kecuali algoritma RF yang justru mengalami penurunan nilai dengan selisih 0,01117.

D. Perbandingan f-measure

Sama halnya dengan presisi dan *recall*, *f-measure* juga menunjukkan kekonsistenan model dalam melakukan proses klasifikasi. Perbedaannya adalah jika presisi membandingkan dengan jumlah total prediksi yang ditujukan pada suatu *class* (TP + FP) dan *recall* membandingkan dengan jumlah *actual class* (TP + FN), maka *f-measure* membandingkan dengan keduanya (TP x (FP + FN)). Baik presisi, *recall,* maupun *f-measure* perlu diketahui nilainya untuk melihat kecenderungan model dalam melakukan prediksi. Jika nilai presisi jauh lebih tinggi dibanding nilai *recall* atau sebaliknya maka model tersebut terindikasi cenderung mengarahkan prediksinya hanya pada salah satu *class* saja (0 atau 1). Akan tetapi jika nilai *f-measure*nya tinggi, hal ini menandakan model tersebut dapat melakukan prediksi yang setimbang di kedua *class* yang ada. Sesuai dengan persamaan 10, nilai *f-measure* didapat dengan menghitung 2 × presisi × *recall* dibagi dengan presisi + *recall*. Dengan melakukan subtitusi pada Persamaan (10) dengan Persamaan (8) dan Persamaan (9) maka didapatkan rumus untuk menghitung *f-measure* seperti pada Persamaan (15),

$$f-measure = 2\times\frac{Presisi \times Recall}{Presisi + Recall}$$

$$f-measure = 2\times\frac{\frac{TP}{TP+FP}\times\frac{TP}{TP+FN}}{\frac{TP}{TP+FP}+\frac{TP}{TP+FN}}$$

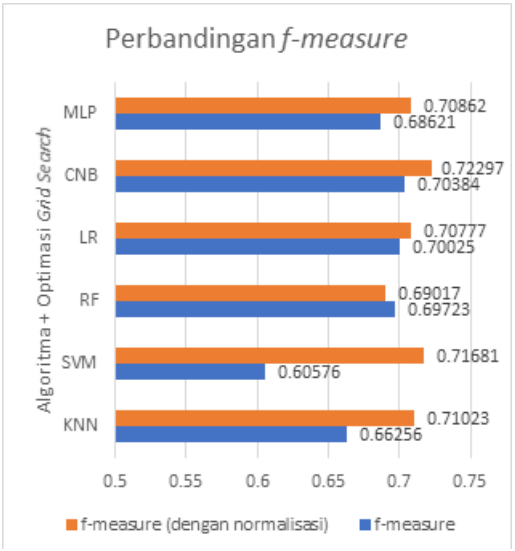
$$f-measure = 2\times\frac{\frac{TP^2}{TP^2+TPFN+TPFP+FPFN}}{\frac{2TP^2+TPFN+TPFP}{TP^2+TPFN+TPFP+FPFN}}$$

$$f-measure = 2\times\frac{TP^2}{2TP^2+TPFN+TPFP}$$

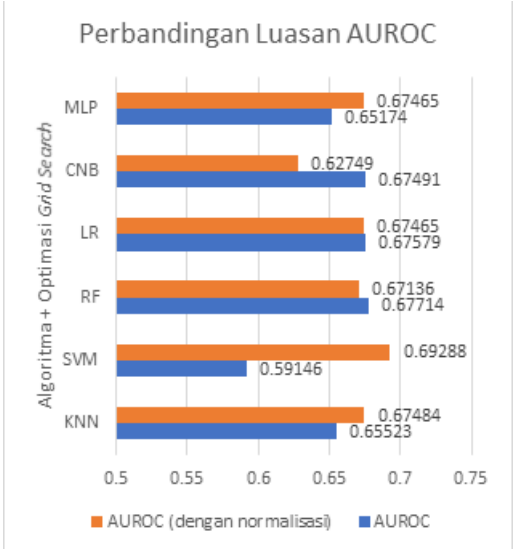
$$f-measure = \frac{2TP}{2TP+FN+FP}. \quad (15)$$

Dari penurunan persamaan tersebut dapat dikatakan bahwa semakin nilai presisi suatu model mendekati nilai *recall*nya maka semakin besar nilai *f-measure*. Kemudian semakin tinggi nilai *f-measure* maka semakin bagus suatu model karena memiliki tingkat kekonsistenan yang tinggi dalam melakukan klasifikasi. Proses berikutnya adalah sama seperti sub bab sebelumnya dimana kelima nilai *f-measure* dari *5-fold cross validation* dirata-rata sehingga didapatkan satu nilai *f-measure* untuk setiap performa model algoritma *machine learning*. Adapun komparasi nilai *f-measure* dari masing-masing algoritma *machine learning* dapat dilihat pada Gambar 7.

Nilai *f-measure* tertinggi didapatkan oleh algoritma CNB yang dioptimasi *grid search* dan dilakukan normalisasi pada tahap *data preprocessing* dengan nilai sebesar 0,72297. Dengan optimasi *grid search* didapatkan parameter α paling optimal yaitu sebesar 0,01. Secara garis besar, hampir semua algoritma *machine learning* kecuali RF mengalami peningkatan nilai *f-measure* ketika dilakukan proses normalisasi data terlebih dahulu sebelum proses pengolahan data.



Gambar 7. Grafik perbandingan *f-measure*



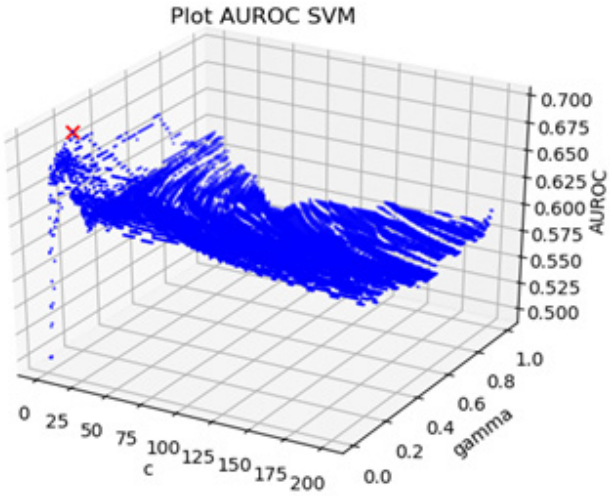
Gambar 8. Grafik perbandingan luasan AUROC

E. *Perbandingan luasan AUROC*

Seperti yang telah dijelaskan pada bab sebelumnya, AUROC merupakan luasan area yang berada di bawah kurva ROC. ROC sendiri merupakan kurva yang dibentuk berdasarkan nilai TPR terhadap FPR. Mengetahui luasan AUROC sangat penting untuk mengetahui kinerja model yang ditunjukkan dengan nilai TPR dan FPRnya. Semakin tinggi TPR terhadap nilai FPR maka semakin tinggi lengkungan kurva yang dibentuk dan semakin luas area di bawah kurva tersebut (AUROC). Semakin luas luasan AUROC maka semakin bagus juga kinerja model *machine learning* tersebut. Sebelum menghitung AUROC, terdapat 2 hal yang perlu dikalkulasi terlebih dahulu yaitu TPR dan FPR di setiap iterasi *cross validation*. Adapun cara untuk mengkalkulasi kedua nilai tersebut dapat dilihat kembali pada Persamaan (9) dan Persamaan (11). Kemudian dari 2 nilai tersebut dapat dibentuk kurva ROC dan selanjutnya dapat dihitung luasan AUROC dengan Persamaan (14). Dari percobaan yang telah dilakukan didapatkan hasil perhitungan AUROC seperti pada Gambar 8.

Seperti ditunjukkan oleh grafik pada Gambar 8, dengan menggunakan teknik normalisasi *min-max scaling* dan algoritma SVM kernel RBF yang dioptimasi *grid search,* didapatkan AUROC terluas dengan luasan sebesar 0,69288. Dengan optimasi *grid search* didapatkan kombinasi nilai *tuning* parameter SVM kernel RBF paling optimal yaitu untuk γ sebesar 0,17 dan *c* sebesar 3. Pada perbandingan ini, 3 algoritma *machine learning* mengalami peningkatan dengan pengaplikasian normalisasi sedangkan 3 algoritma lainnya mengalami penurunan. Adapun plot grafik luasan AUROC SVM kernel RBF pada setiap nilai *tuning grid search* untuk $c = 1$ sampai dengan 200 dan $\gamma = 0{,}01$ sampai dengan 1 dapat dilihat pada Gambar 9.

Sesuai dengan plot grafik pada Gambar 4 dan Gambar 9 dapat dilihat bahwa pola grafik akurasi dan AUROC relatif sama pada setiap nilai *tuning* yang diberikan *grid search*. Hal ini menunjukkan bahwa semakin tinggi nilai akurasi model maka semakin luas nilai luasan AUROCnya.



Keterangan:
. (biru) : bukan AUROC terluas
X (merah) : AUROC terluas (pada c = 3, γ= 0,17)

Gambar 9. Luasan AUROC SVM pada setiap nilai *tuning*

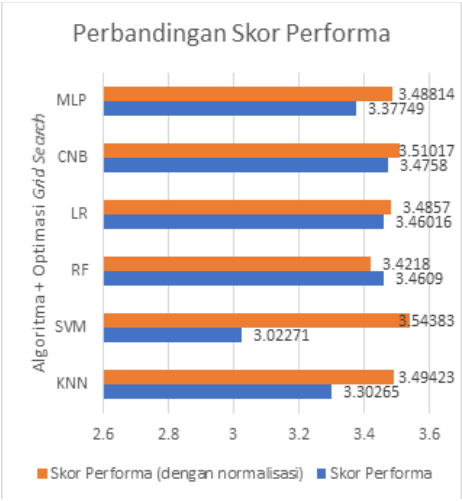
F. *Perbandingan Total*

Seperti dijelaskan sebelumnya, suatu model *machine learning* dikatakan memiliki performa yang bagus jika memiliki nilai akurasi, presisi, *recall*, *f-measure* dan luasan AUROC yang tinggi. Oleh karena itu, untuk menentukan model algoritma terbaik maka pada penelitian ini kelima parameter tersebut dijumlahkan. Dengan demikian maka setiap parameter performa memberikan pengaruh yang sama dalam menentukan model algoritma terbaik. Semakin tinggi skor performa yang didapat maka semakin bagus model algoritma tersebut. Adapun hasil akumulasi nilai performa model *machine learning* dan perbandingannya dapat dilihat pada Gambar 10.

Dari grafik yang ditunjukkan pada Gambar 10, dapat disimpulkan bahwa performa terbaik diperoleh oleh model dari algoritma SVM yang dioptimasi *grid search* dengan *c* optimal sebesar 3 dan γ optimal sebesar 0,17 serta dengan melakukan normalisasi pada tahap *data preprocessing*.

Tabel 7. Rekapitulasi data perbandingan performa model *machine learning*

Normalisasi	Algoritma	Grid Tuning	Akurasi	Presisi	Recall	f-measure	AUROC	Skor Total
Tidak	KNN	K = 17	0,65517	0,6742	0,65549	0,66256	0,65523	3,30265
Ya		K = 29	0,67816	0,66567	0,76533	0,71023	0,67484	3,49423
Tidak	SVM	c = 2 γ = 0,02	0,59195	0,62275	0,61079	0,60576	0,59146	3,02271
Ya		c = 3 γ = 0,17	0,6954	0,68427	0,75447	0,71681	0,69288	3,54383
Tidak	RF	T = 28	0,67816	0,68615	0,72222	0,69723	0,67714	3,4609
Ya		T = 22	0,67241	0,67681	0,71105	0,69017	0,67136	3,4218
Tidak	LR	c = 3,25	0,67816	0,67384	0,73212	0,70025	0,67579	3,46016
Ya		c = 0,4	0,67816	0,65979	0,76533	0,70777	0,67465	3,4857
Tidak	CNB	α = 0,01	0,67816	0,6641	0,75479	0,70384	0,67491	3,4758
Ya		α = 0,01	0,63793	0,6	0,92178	0,72297	0,62749	3,51017
Tidak	MLP	N = 100 L = 0,0009 I = 2000 S = SGD	0,65517	0,64044	0,74393	0,68621	0,65174	3,37749
Ya		N = 200 L = 0,0001 I = 2000 S = ADAM	0,67816	0,66138	0,76533	0,70862	0,67465	3,48814



Gambar 10. Grafik perbandingan performa model algoritma

Model SVM dikatakan yang terbaik karena memiliki jumlahan skor performa paling tinggi dibandingkan dengan model algoritma *machine learning* yang lain. Adapun skor performa yang didapatkan adalah sebesar 3,54383 yang didapatkan dari akumulasi nilai rata-rata akurasi, presisi, *recall*, *f-measure,* dan luasan AUROC. Akumulasi nilai dilakukan karena kelima parameter performa atau indikator kinerja tersebut diperhitungkan seluruhnya dalam pemilihan model terbaik untuk klasifikasi kondisi air tambak.

Dari grafik yang ditunjukkan pada Gambar 10, dapat dikatakan juga bahwa penggunaan teknik normalisasi *min-max scaling* sangat berpengaruh untuk algoritma *machine learning* dimana setiap algoritma (kecuali algoritma RF) mengalami peningkatan performa setelah dilakukan normalisasi *min-max scaling* pada tahap *data preprocessing*.

Fenomena ini sesuai dengan yang telah dijelaskan pada pembahasan sebelumnya yaitu dengan mengaplikasikan normalisasi maka didapatkan atribut input data dengan sebaran nilai yang sama.

Dengan sebaran nilai yang sama maka setiap atribut input (parameter air) memiliki pengaruh yang sama terhadap atribut output (klasifikasi kondisi air). Dengan kata lain, akibat baik dari normalisasi data adalah tidak terjadinya dominasi pada atribut input data dalam proses pemodelan. Teknik normalisasi *min-max scaling* sangat berpengaruh bagi algoritma-algoritma *machine learning* yang memproses data berdasarkan perhitungan matematis. Peningkatan paling signifikan terjadi pada algoritma SVM dengan selisih nilai 0,52112. Sementara itu, normalisasi tidak memberi pengaruh signifikan pada algoritma lain.

Adapun hasil rekapitulasi seluruh data percobaan dalam penelitian ini dapat dilihat pada Tabel 7. Dapat dilihat bahwa model yang dibentuk dengan algoritma SVM dengan kernel RBF serta dengan optimasi algoritma *grid search* tidak menghasilkan nilai akurasi diatas 70%. Hal ini bisa disebabkan oleh karakteristik dataset yang menyebar dimana data-data sampel dengan *class* yang sama tidak berdekatan namun membaur dengan data-data dari *class* lain. Fenomena ini dapat terjadi jika setiap parameter atau atribut data tidak menunjukkan nilai-nilai yang berbeda secara signifikan dalam merepresentasikan *class*nya masing-masing, sehingga mengakibatkan algoritma SVM kesulitan dalam membentuk model klasifikasi yang dapat memisahkan data-data dari 2 buah *class* yang berbeda. Hal ini juga sesuai dengan penjelasan pada bab sebelumnya dimana parameter air tambak cenderung tidak berubah secara signifikan ketika air tambak berubah kondisi dari baik ke buruk ataupun sebaliknya.

V. KESIMPULAN

Dari serangkaian percobaan yang telah dilakukan, dapat disimpulkan bahwa model terbaik untuk klasifikasi kondisi air tambak didapatkan oleh model yang dibentuk dengan algoritma SVM kernel RBF yang dioptimasi dengan menggunakan algoritma *grid search* dimana didapatkan nilai optimasi paling optimal yaitu 3 untuk parameter *c* dan 0,17 untuk parameter γ . Adapun skor performa paling tinggi yang didapatkan oleh model algoritma SVM ini adalah 3,54383 yang didasarkan dari akumulasi perhitungan nilai akurasi, presisi, *recall, f-measure,* dan luasan AUROC saat proses pemodelan dan validasi. Selain itu, normalisasi yang dilakukan dengan menggunakan teknik *min-max scaling* pada tahap *data preprocessing* juga memberi pengaruh pada performa model yang dibentuk. Hal ini dibuktikan dari peningkatan skor performa pada seluruh algoritma *machine learning* kecuali algoritma RF. Peningkatan performa paling signifikan terjadi pada algoritma SVM kernel RBF dimana selisih skor performa antara model dengan normalisasi dan model tanpa normalisasi sebesar 0,52112. Sementara itu, normalisasi tidak memberi pengaruh yang signifikan bagi model algoritma yang lain. Dengan demikian kesimpulan akhir yang dapat ditarik dalam penelitian ini adalah bahwa model algoritma yang paling bagus untuk dipakai dalam mengatasi permasalahan klasifikasi kondisi air tambak selama proses budidaya perairan (akuakultur) adalah model yang dibentuk dengan algoritma SVM kernel RBF dengan optimasi *grid search* pada parameter *c* dan γ serta dilakukan proses normalisasi *min-max scaling* pada tahap *data preprocessing.*

REFERENSI

[1] H. M. Atmomarsono, Supito, M. Mangampa, and H. W. Pitoyo. Better Management Practice Budidaya Udang Vanamei, 1st ed., Jakarta, WWF Indonesia, 2014.

[2] D. Yuswantoro, O. Natan, A. N. Angga, A. I. Gunawan, Taufiqurrahman, B. S. B. Dewantara, dan A. Kurniawan, "Fuzzy logic-based control system for dissolved oxygen control on indoor shrimp cultivation," *IES-ETA,* pp. 37-42, 2018.

[3] C. Liao dan C. H. Wen, "SVM-Based Dynamic Voltage Prediction for Online Thermally Constrained Task Scheduling in 3-D Multicore Processors," *IEEE Embedded Systems Letters*, vol. 10, no. 2, pp. 49-52, 2018.

[4] Y. Guo, X. Jia, dan D. Paull, "Effective Sequential Classifier Training for SVM-Based Multitemporal Remote Sensing Image Classification," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3036-3048, 2018.

[5] A. Rojas-Domínguez, L. C. Padierna, J. M. Carpio Valadez, H. J. Puga-Soberanes, dan H. J. Fraire, "Optimal Hyper-Parameter Tuning of SVM Classifiers with Application to Medical Diagnosis," *IEEE Access*, vol. 6, pp. 7164-7176, 2018.

[6] C. Cortes dan V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[7] J. P. Vert, K. J. Tsuda, dan B. Scholkopf, "A Primer on Kernel Methods," *Kernel Methods in Computational Biology*, 2004.

[8] J. Bergstra dan Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[9] D. Pyle, *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, Los Altos, California. 1999.

[10] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[11] H. T. Kam, "Random Decision Forests," *International Conference on Document Analysis and Recognition*," vol. 14, pp. 278–282, 1995.

[12] D. A. Freedman, "Statistical Models: Theory and Practice," *Cambridge University Press*., p. 128, 2009.

[13] J. D. Rennie, L. Shih, J. Teevan, dan D. R. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," *20th International Conference on Machine Learning*, vol. 3, pp. 616-623, 2003.

[14] R. Collobert dan S. Bengio, "Links between Perceptrons, MLPs and SVMs," *21st International Conference on Machine Learning,* 2004.

[15] S. Mei, "A mean field view of the landscape of two-layer neural networks," *National Academy of Sciences*, vol. 33, p. 115, 2018.

[16] D. P. Kingma dan L. B. Jimmy, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 2015.

[17] F. Tom, "An Introduction to ROC Analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[18] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.

[19] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[20] R. Kohavi. "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proc. 14th International Joint Conference on Artificial Intelligence, 1995, p. 1137.

[21] S. Arlot dan A. Celisse, "A Survey of Cross-Validation Procedures for Model Selection," *Statistics Surveys,* vol. 4, pp. 40–79, 2010.

Penerbit:
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala
Jl. Tgk. Syech Abdurrauf No. 7, Banda Aceh 23111
website: http://jurnal.unsyiah.ac.id/JRE
email: rekayasa.elektrika@unsyiah.net
Telp/Fax: (0651) 7554336