

Komparasi dan Analisis Kinerja Model Algoritma SVM dan PSO-SVM (Studi Kasus Klasifikasi Jalur Minat SMA)

Theopilus Bayu Sasongko

Program Studi D4 Teknik Informatika, Politeknik Harapan Bersama Tegal

Jalan Mataram No 9 Pesurungan Lor, Tegal

theopilus.27@gmail.com

Abstract — Attribute Selection is very important for classification process. This research has been done by doing attribute selection using PSO method (Particle Swarm Optimization) on SVM algorithm (Support Vector Machine). The development of the classification model uses three parameters especially data attribute, influence of the transformation of various kernel function and penalty factor (C) toward the performance of SVM and PSO-SVM classification. The analysis uses five kernels in mySVM library that existed in Rapidminer application namely dot, radial, polynomial, neural, and anova kernel. The training data used in the first model classification development is student interest data at ABC high school on 2013-2014 year academic. The first model is evaluated using accuracy, precision, recall, and auc value test. The first result shows that the anova kernel on PSO-SVM is able to work with accuracy level 99.30% using penalty factor 0.1. The second model has been developed to predict student interest in XYZ high school. The second result shows that PSO-SVM with kernel anova is able to classify students interest with 99.29% accuracy level.

Keywords— Optimization, SVM, PSO-SVM, Student Interest.

I. PENDAHULUAN

Membimbing siswa-siswi untuk memilih jurusan yang tepat sangat penting dalam jenis pembelajaran apapun. Keterlambatan mengetahui jurusan yang tepat bagi siswa-siswi adalah suatu kerugian, baik siswa maupun bagi lembaga yang bersangkutan.

Sudah ada alat bantu untuk melakukan prediksi maupun klasifikasi dengan kasus di lembaga pendidikan salah satunya adalah menggunakan metode *Support Vector Machine* [1]. SVM sebagai salah satu metode klasifikasi yang memiliki tingkat akurasi yang tinggi dalam melakukan prediksi pengklasifikasian potensi pada lembaga pendidikan. SVM memiliki kelebihan dalam mengklasifikasikan suatu pola dengan akurat walaupun dengan keterbatasan *dataset* seperti pada penelitian SVM yang diterapkan dalam lingkup kasus lembaga pendidikan yang dilakukan oleh Nimit Pattanasri, dkk pada tahun 2012

[2]. Penelitian yang bertujuan mengklasifikasikan jenis *slide* presentasi berdasarkan fitur pada tiap matakuliah ini hanya menggunakan data sebanyak 102 kuisisioner yang diberikan kepada mahasiswa. Namun SVM memiliki keterbatasan ketika jumlah atribut yang digunakan relatif banyak yang mengakibatkan beban komputasi menjadi berat dan akurasi menjadi kurang akurat.

Pada penelitian yang sebelumnya peneliti melakukan penelitian dengan menerapkan algoritma SVM pada kasus pengklasifikasian jalur minat SMA, dengan melakukan uji coba di semua *kernel* yang terdapat di *mySVM* dengan beberapa fungsi C (faktor pinalti) akan tetapi hasil *akurasi*, *presisi*, dan *recall* kurang maksimal karena tidak menggunakan metode seleksi atribut [3].

Salah satu metode seleksi atribut yang cukup terkenal adalah PSO (*particle swarm optimization*). PSO adalah suatu algoritma yang digunakan untuk menyelesaikan masalah optimasi. Beberapa kelebihan PSO adalah mudah untuk diimplementasikan dan hanya membutuhkan sedikit parameter, PSO lebih efisien dalam hal komputasi, dan PSO lebih bersifat fleksibel dalam menjaga keseimbangan antara pencarian global dan lokal optimal.

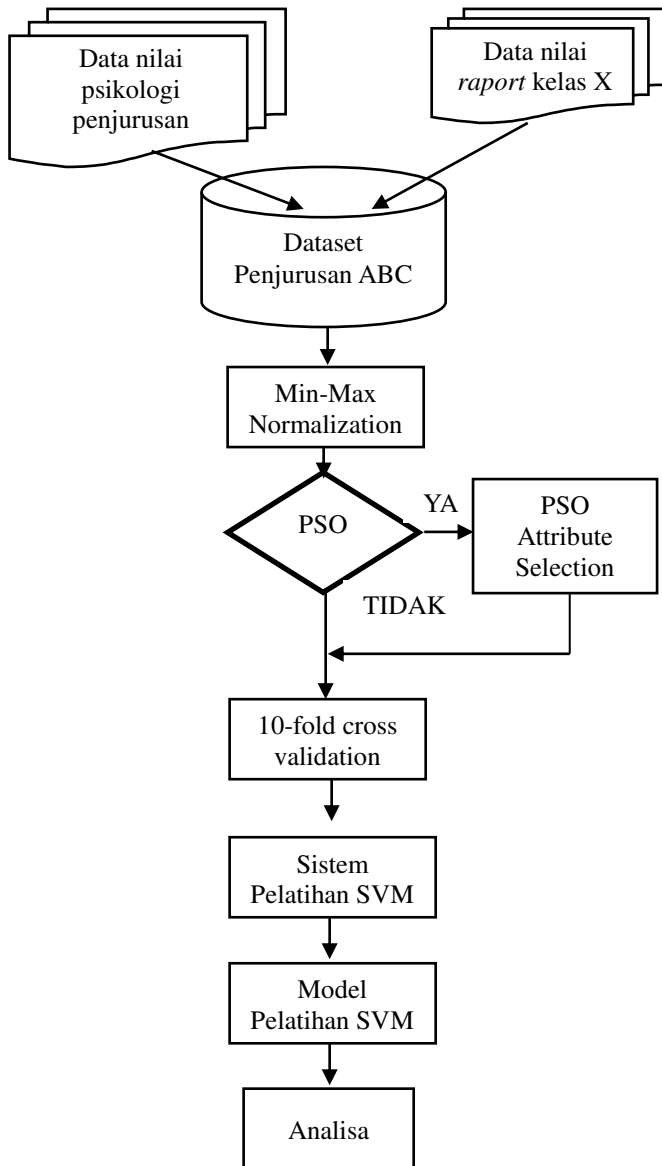
Chung-Jui Tu, dkk pernah melakukan penelitian mengenai seleksi atribut dengan menggunakan *PSO-Multi Class SVM* pada beberapa jenis *dataset*. Pada penelitian ini Cheng-Jui Tu, dkk [4] pernah mengimplementasikan *PSO-Multi Class SVM* pada beberapa jenis *dataset* yang atributnya berjumlah banyak seperti *vowel dataset*, *wine*, *wdbc*, *ionosphere*, dan *sonar dataset*. Hasil dari penelitian mereka disimpulkan bahwa penggunaan PSO dapat dengan baik diintegrasikan dengan *Multiclass SVM* dengan tingkat akurasi jauh lebih baik dibandingkan dengan SVM tanpa metode PSO. Di penelitiannya juga dijelaskan dampak negatifnya adalah waktu komputasi akan relatif lebih lama dibandingkan menggunakan SVM tanpa PSO [4]. Penelitian yang serupa juga dilakukan oleh Fatima Ardjani dan Kaddour Sadouni, mereka memanfaatkan PSO-SVM untuk mengoptimisasi multi kelas SVM [5].

Pada penelitian ini digunakan pendekatan *hybrid* yang mengkombinasikan antara *support vector machine* (SVM)

klasifier dengan metode *particle swarm optimization* (PSO) dengan jumlah *dataset* sebanyak 281 records dan 11 atribut. Tujuan dan manfaat dari penelitian ini adalah melakukan proses seleksi atribut menggunakan PSO pada *dataset* yang digunakan sehingga akan meningkatkan akurasi dari model klasifikasi SVM.

II. METODOLOGI

Metodologi dan alur penelitian yang dilakukan hingga pembentukan model klasifikasi tercantum seperti pada skema Gambar 1 sebagai berikut:



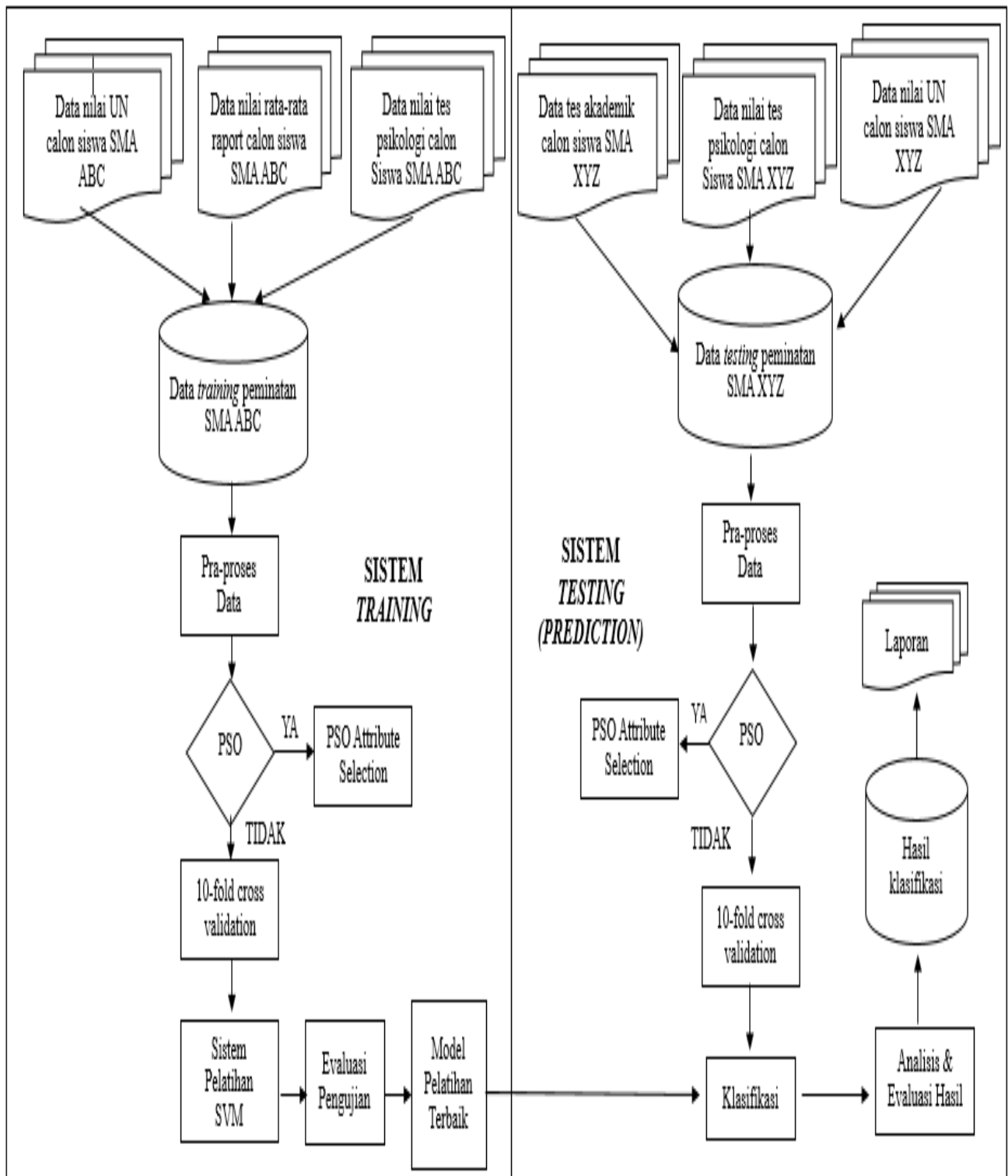
Gambar 1. Skema Metodologi Penelitian

Metodologi penelitian pada Gambar 1 dimulai dari pengumpulan *dataset* yang akan digunakan pada proses

selection attribute, training hingga pembentukan model klasifikasi. Bahan yang digunakan diambil dari nilai peserta didik saat mendaftar di SMA ABC pada tahun ajaran 2013-2014 yang meliputi nama, nilai UN pada jenjang sebelumnya, nilai rata-rata *raport*, dan nilai tes psikologi peminatan yang berjumlah 281 siswa. Siswa yang berlabel penjurusan IPA berjumlah 64 siswa dan siswa yang berlabel penjurusan IPS berjumlah 216 siswa.

Dataset yang telah dikumpulkan kemudian dijadikan menjadi satu kesatuan yang dinamakan *dataset* Penjurusan ABC yang kemudian dilanjutkan dengan melakukan *pre-processing* data dalam bentuk *scaling method*. Peneliti menggunakan *Min-Max Normalization* dengan nilai minimal adalah -1 dan nilai maksimal adalah 1 karena menurut peneliti akan menunjukkan hasil yang baik dibandingkan *scaling method* yang lain berdasarkan studi pustaka penelitian lain [6]. Data yang telah melewati proses *pre-processing* kemudian dilakukan dua perlakuan yang berbeda untuk mengetahui perbandingan kinerja model performansi dari SVM dan PSO-SVM. Model yang dibangun dengan SVM kemudian akan langsung dilakukan *10-fold cross validation* sedangkan model yang dibangun dengan PSO-SVM akan dilakukan pembobotan dan penyeleksian atribut dengan menggunakan teknik *particle swarm optimization* (PSO) kemudian Atribut data yang telah dilakukan seleksi kemudian baru dibagi menjadi dua bagian yaitu *data training* dan *data testing* menggunakan *10-fold cross validation* untuk menguji performansi model SVM hingga menghasilkan model pelatihan yang paling optimal. Untuk melakukan pengujian model terbaik ini dilakukan dengan membandingkan kinerja atau performansi model dari SVM dan PSO-SVM. Kriteria pengujiannya meliputi uji akurasi, uji presisi, uji *recall* dan uji nilai AUC (*area under curve*). Model terbaik yang telah terbentuk kemudian akan diterapkan (uji coba) kedalam *dataset* SMA XYZ yang berbeda dengan *data training* SMA ABC yang digunakan untuk menghasilkan model. *Dataset* SMA XYZ berjumlah 288 siswa dengan label peminatan IPA sebanyak 150 siswa IPA, dan 138 siswa IPS. Model yang telah diuji coba kedalam *dataset* SMA XYZ kemudian akan dilakukan analisis dan evaluasi hasil dengan cara mencocokkan label yang telah terbentuk pada saat pengklasifikasian data dengan label asli yang diberikan oleh SMA XYZ.

Berikut adalah gambaran secara menyeluruh dari proses pembentukan model hingga proses klasifikasi data seperti pada Gambar 2.



Gambar 2. Proses Pembentukan Model hingga Klasifikasi Data

A. Data Mining

Data mining adalah proses untuk menggali (ekstraksi) pola dari sekumpulan data yang berpotensi guna memperoleh suatu informasi yang dibutuhkan [11]. Menurut Han dan Kamber *data mining* berkembang karena adanya kebutuhan yang sangat mendasar yaitu adanya sejumlah besar data yang sebenarnya dapat digunakan untuk menghasilkan suatu informasi yang berguna. Han dan Kamber menyatakan *data mining* memiliki tujuh tahap atau langkah-langkah yang pada umumnya biasa diterapkan yaitu [11].

- 1) *Data cleaning* yaitu proses penghapusan suatu bagian dari data yang tidak konsisten atau yang mengandung *noise*.
- 2) *Data integration* yaitu proses penggabungan atau pengintegrasian sumber-sumber data dari berbagai sumber data.
- 3) *Data selection* yaitu proses penyeleksian atau pengambilan data yang sesuai dengan kebutuhan.
- 4) *Data transformation* yaitu proses perubahan data menjadi suatu bentuk yang sesuai dengan tujuan untuk pengambilan keputusan.
- 5) *Data mining* yaitu suatu proses mengolah data dengan menggunakan suatu metode atau algoritme tertentu guna menghasilkan suatu pola data.
- 6) *Pattern evaluation of data* yaitu proses untuk menguji kebenaran dari pola data yang mewakili pengetahuan yang ada di dalam data itu sendiri.
- 7) *Knowledge representation* yaitu proses menyajikan pengetahuan untuk menampilkan hasil pengolahan menjadi informasi kepada pihak pengguna.

B. Klasifikasi Data

Klasifikasi data adalah suatu proses memisahkan serta mengelompokkan sekumpulan data dengan kumpulan data yang lain [11]. Menurut Han dan Kamber klasifikasi data memiliki dua tahap proses. Pertama yaitu membangun suatu model atau pola berdasarkan serangkaian kelas data yang selanjutnya akan di klasifikasikan. Tahap pertama ini juga sering disebut dengan tahap *training* atau proses pembelajaran, proses pembangunan model ini dengan cara menganalisis data *training* yang ada [11].

Pada umumnya proses pembangunan model ini diterjemahkan ke dalam bentuk *classification rules*, *decision tree*, atau *mathematical* model lainnya. Selanjutnya pada tahap kedua yaitu proses pengklasifikasian menggunakan data *testing*, model yang telah ditentukan akan digunakan untuk memprediksi data yang belum diketahui labelnya [11].

C. Algoritma SVM

Support Vector Machine adalah suatu metode atau algoritma untuk melakukan klasifikasi maupun prediksi [7]. Prinsip kerja dari metode ini adalah mencari ruang pemisah yang paling *optimal* dari suatu *dataset* dalam kelas yang berbeda. Dalam kehidupan sehari-hari, kita sering diperhadapkan pada persoalan-persoalan yang tidak *linear* / data yang tidak dapat benar-benar dipisahkan secara *linear* yaitu suatu kondisi dimana tidak ada sebuah garis atau bidang yang dapat dibuat untuk menjadi pemisah antar kelas data. Pada permasalahan ini ada 2 langkah yang dapat dilakukan, yaitu:

- 1) Menggunakan soft margin hyperplane.

Tujuan dari soft margin hyperplane adalah mengubah data yang bersifat non linear ke dalam bentuk linear dengan tetap mempertahankan bidang pembatas tetap fleksibel.

Formulasi pada *soft margin hyperplane* yang menggunakan variabel slack (ξ) dirumuskan dengan Persamaan (1)[6].

$$\begin{aligned} x_i \cdot w + b &\geq 1 - \xi \text{ untuk } y_i = \text{kelas 1} \\ x_i \cdot w + b &\leq -1 + \xi \text{ untuk } y_i = \text{kelas 2} \end{aligned} \quad (1)$$

Dengan Persamaan (1), maka pencarian bidang pemisah terbaik dapat diformulasikan menjadi Persamaan (2)[6].

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (2)$$

Dengan fungsi pemisah $y_i(x_i \cdot w + b) \geq 1 - \xi \forall i = 1, \dots, n$

C adalah parameter yang menentukan besarnya pinalti yang dikarenakan oleh kesalahan pengklasifikasian data.

- 2) Mencari hyperplane pemisah linear dalam ruang dimensi yang baru (feature space).

Mengubah *input space* (*dot product*) kedalam bentuk *feature space* sering dikenal dengan teknik kernel *trick* $\Phi(\bar{x}_i) \cdot \Phi(\bar{x}_j)$ yang kemudian berkembang menjadi fungsi Kernel $K(x_i, x_j)$ [6]. Perubahan dari *input space* ke *feature space* mengakibatkan komputasi yang sangat besar, karena ada kemungkinan dimensi *feature space* yang sangat banyak bahkan tak terhingga. Oleh karena itu SVM menjembatani hal itu dengan fungsi *Kernel*.

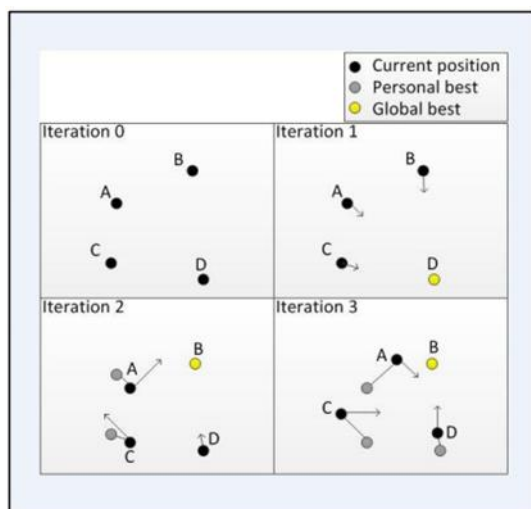
Pada Penelitian ini digunakan lima kernel meliputi kernel *dot*, *radial*, *polynomial*, *neural*, dan *anova* (*analysis of variable*).

D. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan algoritma *swarm intelligence* yakni studi mengenai sistem komputasi yang terinspirasi oleh kecerdasan kolektif [8]. Kecerdasan kolektif muncul berdasarkan populasi atau kerjasama *homogen* dalam sebuah lingkungan. Dalam PSO, sebuah kawanan diasumsikan memiliki ukuran tertentu dengan setiap partikel memiliki posisi awal yang terletak acak di suatu lokasi dalam satu ruang. Setiap partikel diasumsikan memiliki dua karakter: posisi dan kecepatan [9]. Setiap partikel bergerak dalam ruang tertentu dan mengingat posisi

terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan. Setiap partikel menyampaikan informasi atau posisi terbaiknya kepada partikel lain. Pada algoritma PSO, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel.

Populasi dibangkitkan secara *random* dengan batasan nilai terkecil dan terbesar (batas bawah dan batas atas). Setiap partikel mencari solusi dengan melintasi ruang pencarian dengan cara melakukan penyesuaian terhadap posisi terbaiknya (*local best*) dan penyesuaian terhadap posisi partikel terbaik dari seluruh kawanan (*global best*) selama melintasi *search space*. Dilakukan sejumlah iterasi untuk mencari posisi terbaik setiap partikel sampai didapatkan posisi yang relatif tetap atau mencapai batas iterasi yang ditetapkan. Pada tiap *iterasi*, setiap solusi (posisi partikel) dievaluasi performanya dengan cara memasukkan solusi tersebut ke dalam *fitness function*.



Gambar 3. Proses Partikel Mencari Posisi Terbaik

PSO terbukti merupakan algoritma yang baik dan efektif untuk permasalahan optimasi karena kemudahannya dalam penerapan kode dan performanya yang konsisten [10].

E. Min-Max Normalization

Min-Max normalization adalah bentuk *scaling* normalisasi data yang digunakan untuk mencegah dominasi atribut yang memiliki jangkauan nilai yang besar terhadap atribut dengan nilai jangkauan yang kecil, selain itu *min-max normalization* juga dapat mencegah permasalahan numerik selama perhitungan dengan Persamaan (3) [11].

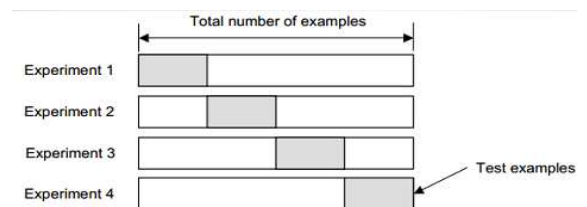
$$D'(i) = \frac{D(i) - \min(D)}{\max(D) - \min(D)}(U - L) + L \quad (3)$$

$D'(i)$ adalah nilai dari data i dari atribut D yang telah dilakukan proses normalisasi, $D(i)$ adalah nilai data i yang asli, U dan L adalah batas atas (*upper*) dan batas bawah (*lower*) normalisasi. $\min(D)$ adalah nilai minimum dari sebuah atribut data D . $\max(D)$ adalah nilai maksimum

dari sebuah atribut D [6]. Penggunaan metode *Min-Max normalization* ini didasarkan penelitian komparasi metode-metode normalisasi data masukan untuk *Support vector machine* yang pernah dilakukan oleh Ali dan Smith[12] dimana hasil menunjukkan bahwa metode *min-max normalization* memberikan tingkat akurasi dan performansi yang lebih baik dibanding dengan normalisasi yang menggunakan metode *zero mean* dan *log scaling*.

F. K-fold Cross Validation

K-fold cross validation adalah teknik untuk mengestimasi performansi dari model pelatihan yang telah dibangun [11]. Metode ini membagi data *training* dan data *testing* sebanyak k bagian data. Fungsi dari *k-fold cross validation* adalah agar tidak ada *overlapping* pada data *testing*. Berikut ilustrasi sederhana dari *k-fold cross validation* ditunjukkan pada Gambar 4.



Gambar 4. Ilustrasi k-fold cross validation

Pada ilustrasi yang tertera pada Gambar 4 dapat dijabarkan bahwa percobaan menggunakan *4-fold cross validation*. Hal ini tertera pada jumlah eksperimen yang dilakukan Kotak yang berwarna abu-abu merupakan *test set* dan sisanya (kotak berwarna putih) merupakan *training set*. Sebagai contoh diberikan data sebanyak 40 *instances* pada ilustrasi Gambar 4, maka eksperimen pertama kotak yang berwarna abu-abu dari *instances* 1 hingga *instances* dengan urutan yg ke 10 yang berjumlah 10 *instances* merupakan *test set* dan sisanya *instances* urutan ke 11 hingga urutan ke 30 yang berjumlah 30 *instances* (kotak yang berwarna putih) merupakan *training set*. Dari eksperimen pertama didapatkan hasil nilai rata-rata *error*. Kemudian dilanjutkan cara yang sama untuk eksperimen kedua seperti pada eksperimen pertama hingga eksperimen ke empat. Setelah itu dihitung rata-rata keseluruhan estimasi *error* dari setiap eksperimen yang telah dilakukan hingga akhir.

III. PENGUJIAN PERFORMANSI MODEL

Pengujian performansi model pada penelitian ini dilakukan dengan membandingkan kernel SVM dan PSO-SVM pada *library* mySVM yang terdapat pada aplikasi *Rapidminer* hingga didapatkan suatu model dengan performansi tertinggi. Parameter yang digunakan sebagai evaluasi komparasi kernel ini adalah akurasi, presisi, *recall*, kurva ROC (AUC), dan uji statistik *anova* [6]. Berikut akan

dibahas mengenai parameter evaluasi yang akan digunakan sebagai pengujian performansi model SVM.

1) Akurasi, presisi, dan recall

Akurasi dapat didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Presisi menunjukkan tingkat ketepatan atau ketelitian dalam pengklasifikasian. Sedangkan recall berfungsi untuk mengukur proporsi positif aktual yang benar diidentifikasi. Untuk mengukur akurasi, presisi, dan recall biasanya digunakan *confusion matrix*. *Confusion matrix* adalah alat ukur berbentuk matrix yang digunakan untuk mendapatkan jumlah ketepatan klasifikasi terhadap kelas dengan algoritme yang dipakai. Berikut akan disajikan bentuk *confusion matrix* pada Tabel 1.

TABEL I
BENTUK CONFUSION MATRIX DARI DUA KELAS

Confusion Matrix		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	FALSE	FN (False Negative) Missing result	TN (True Negative) Correct absence of result

Pada Tabel I nilai TP (*true positive*) dan TN (*true negative*) menunjukkan tingkat ketepatan klasifikasi. Umumnya semakin tinggi nilai TP dan TN semakin baik pula tingkat klasifikasi dari akurasi, presisi, dan recall. Jika label prediksi keluaran bernilai benar (*true*) dan nilai sebenarnya bernilai salah (*false*) disebut sebagai *false positive* (FP). Sedangkan jika prediksi label keluaran bernilai salah (*false*) dan nilai sebenarnya bernilai benar (*true*) maka hal ini disebut sebagai *false negative* (FN) [11]. Berikut formulasi untuk menghitung akurasi, presisi, dan recall pada pembentukan model klasifikasi ditunjukkan pada Persamaan (4), Persamaan (5), dan Persamaan (6) [11].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{4}$$

$$Presisi = \frac{TP}{TP + FP} \times 100\% \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{6}$$

A. Kurva ROC

Kurva ROC (*receiver operating characteristic*) adalah salah satu alat ukur untuk menilai kemampuan sistem klasifikasi. Pada penelitian ini akan digunakan alat ukur kurva ROC untuk membandingkan kernel-kernel SVM dengan PSO-SVM pada *library mySVM* yang terdapat pada program *Rapidminer*. Kurva ROC pertama kali diimplementasikan pada teori deteksi sinyal. Kemudian berkembang dan digunakan pada bidang kedokteran, radiologi, serta bidang yang lain. Kurva ROC sering digunakan untuk mengevaluasi pengklasifikasian karena memiliki kemampuan evaluasi algoritma dengan cukup baik [6].

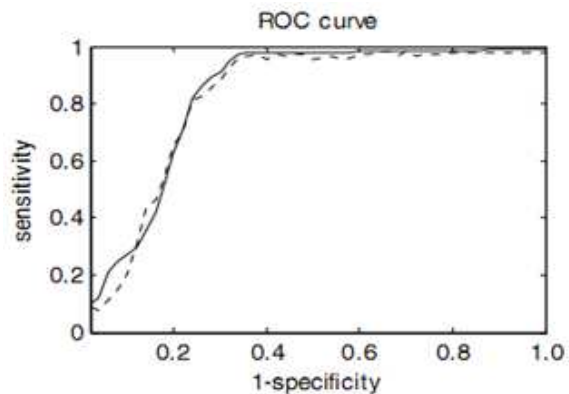
Kurva ROC merupakan grafik perbandingan antara *sensitivity* (*true positive rate* (TPR)) yang diterjemahkan kedalam sumbu vertikal atau sumbu koordinat y dengan *specificity* (*false positive rate* (FPR)) yang diterjemahkan dalam bentuk kurva. Berikut formulasi dari *sensitivity* dan *specificity* dipaparkan pada Persamaan (7), dan Persamaan (8) [6].

$$\tag{7}$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{8}$$

$$Specificity = \frac{FP}{FP + TN} \times 100\%$$

Kurva ROC dapat digunakan sebagai komparasi beberapa metode (*classifier*) ataupun model *classifier* yang memiliki perbedaan parameter guna mendapatkan model yang paling baik. Berikut adalah contoh penerapan komparasi performansi dari dua *classifier* yang berbeda pada Gambar 5.



Gambar 5 Komparasi Classifier dengan Kurva ROC

Pada Gambar 5 dapat dilihat bahwa terdapat dua buah *classifier* yang disimbolkan dengan garis putus-putus dan garis utuh. Jika pada Gambar 5 menunjukkan letak koordinat

(0, 1) hal tersebut mewakili *sensitivity* dan *specificity* sebesar 100%. Untuk menghitung dan memastikan *classifier* mana yang lebih unggul maka digunakan penghitungan AUC (*area under curve*).

AUC (*area under curve*) adalah luas area dibawah kurva. Luas dari AUC selalu berada diantara nilai 0 hingga 1. AUC dihitung berdasarkan gabungan luas trapesium titik-titik (*sensitivity* dan *specificity*). Pada Gambar 5 memperlihatkan bahwa garis yang utuh memiliki area dibawah kurva yang lebih besar dibandingkan garis yang putus-putus, hal ini berarti bahwa tingkat performansi klasifikasi dari *classifier* yang dilambangkan dengan garis utuh lebih baik dibandingkan tingkat performansi klasifikasi dari *classifier* yang dilambangkan dengan garis putus-putus.

erikut adalah standar Tabel kategori pengklasifikasian berdasarkan nilai AUC pada Tabel II

TABEL II.

KATEGORI PENGKLASIFIKASIAN BERDASARKAN NILAI AUC

Nilai AUC	Kategori Pengklasifikasian
0.90 - 1.00	Excellent
0.80 - 0.90	Good
0.70 - 0.80	Fair
0.60 - 0.70	Poor
0.50 - 0.60	Fail

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini digunakan data pelatihan guna pembentukan model klasifikasi yaitu nilai peserta didik saat mendaftar di salah satu SMA di Jawa Tengah pada tahun pelajaran 2013-2014 yang meliputi nama, nilai UN pada jenjang sebelumnya, nilai rata-rata *raport*, dan nilai tes psikologi peminatan sebanyak 280 siswa.

Tujuan dari penelitian ini adalah melakukan komparasi dan menganalisis kinerja model SVM yang tidak menggunakan PSO (SVM) dan model SVM yang berbasis PSO (PSO-SVM) yang diterapkan pada lima fungsi kernel SVM pada *library mySVM* yang terdapat pada aplikasi *Rapidminer* meliputi kernel *dot*, *radial*, *polynomial*, *neural*, dan *anova*. Setiap percobaan dengan berbagai model SVM kemudian dievaluasi menggunakan evaluator akurasi, presisi, *recall* guna mendapatkan model terbaik.

Pada Tabel III dan Tabel IV memperlihatkan perbandingan tingkat akurasi pengklasifikasian data menggunakan SVM biasa dan PSO-SVM.

TABEL III

TINGKAT AKURASI KLASIFIKASI SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neu-ral	Anova
0.0	84.72	80.42	80.09	66.23	95.37
0.1	84.72	76.87	86.13	73.63	97.87
0.2	85.78	76.87	82.56	69.33	96.08
0.4	84.70	77.23	83.29	68.62	93.23
0.8	86.12	78.65	80.80	67.56	96.08
1.0	86.83	80.42	85.09	66.23	95.73
1.2	87.19	81.83	82.59	66.95	97.16
1.4	87.19	82.19	84.72	65.17	97.16
1.8	87.19	81.85	84.73	66.96	97.16

TABEL IV

TINGKAT AKURASI PSO-SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neu-ral	Anova
0.0	87.94	93.25	81.86	76.17	98.60
0.1	87.56	85.03	91.10	80.10	99.30
0.2	87.94	82.93	92.21	80.10	98.60
0.4	87.23	93.25	83.80	80.10	97.88
0.8	87.57	95.04	84.20	77.97	98.30
1.0	87.92	93.25	86.21	79.03	98.60
1.2	87.23	93.60	81.86	77.64	98.50
1.4	87.56	92.55	90.10	77.98	98.30
1.8	87.92	92.91	83.80	78.29	98.60

Dilihat dari Tabel III dan Tabel IV dapat dilakukan perbandingan akurasi antara SVM dengan PSO-SVM dan ditarik kesimpulan bahwa optimasi dengan pemilihan atribut (parameter) yang tepat menggunakan PSO (*particle swarm optimization*) dapat meningkatkan nilai akurasi model SVM pada semua jenis kernel yang diujikan dengan beberapa faktor pinalti dari 0.0,0.1,0.2,0.4,0.8,1.0,1.2,1.4, dan 1.8.

Tingkat akurasi yang paling besar sebesar 99.30% diperoleh ketika menerapkan PSO-SVM dengan parameter C (pinalti) sebesar 0.0 dengan menggunakan kernel *anova*.

Pengujian kedua adalah dengan menggunakan evaluator *presisi*. Pada Tabel V dan Tabel VI memperlihatkan perbandingan tingkat presisi pengklasifikasian data menggunakan SVM biasa dengan PSO-SVM.

TABEL V
TINGKAT PRESISI KLASIFIKASI SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neural	Anova
0.0	88.03	79.80	81.64	83.56	95.05
0.1	88.03	76.87	89.29	87.47	98.65
0.2	89.81	76.87	87.12	84.11	96.73
0.4	89.96	77.15	89.16	88.10	97.56
0.8	90.38	78.32	85.75	82.57	97.55
1.0	91.12	79.80	89.42	83.60	98.65
1.2	90.62	81.01	88.11	82.80	98.65
1.4	90.88	81.33	90.47	84.93	98.65
1.8	90.88	81.50	92.07	82.96	98.65

TABEL VI
TINGKAT PRESISI KLASIFIKASI PSO-SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neural	Anova
0.0	90.71	92.04	82.52	84.47	98.86
0.1	88.19	76.87	92.50	86.03	99.57
0.2	90.71	83.16	94.12	90.70	98.66
0.4	89.89	95.55	83.20	88.49	98.18
0.8	89.20	97.76	84.40	85.27	98.40
1.0	90.30	92.04	85.12	84.78	99.57
1.2	91.00	97.71	93.12	86.60	98.86
1.4	89.87	93.11	95.16	84.73	98.23
1.8	91.06	94.74	96.12	84.47	98.86

Dilihat dari Tabel V dan Tabel VI dapat dilakukan perbandingan nilai presisi antara SVM dengan PSO-SVM. Ditarik kesimpulan bahwa optimasi dengan pemilihan atribut (parameter) yang tepat menggunakan PSO (*particle swarm optimization*) secara garis besar dapat meningkatkan nilai presisi model SVM pada hampir semua jenis kernel yang diujikan.

Tingkat presisi yang paling besar sebesar 98.86% diperoleh ketika menerapkan PSO-SVM dengan parameter C (pinalti) sebesar 0.0 dengan menggunakan kernel *anova*.

Pengujian ketiga adalah dengan menggunakan evaluator *recall*. Pada Tabel VII dan Tabel VIII memperlihatkan perbandingan tingkat *recall* pengklasifikasian data menggunakan SVM biasa dengan PSO-SVM.

TABEL VII
TINGKAT RECALL KLASIFIKASI SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neural	Anova
0.0	93.55	100.00	95.80	72.27	99.52
0.1	93.55	100.00	93.48	80.04	98.59
0.2	92.62	100.00	90.65	77.34	98.59
0.4	90.78	100.00	89.24	71.45	93.38
0.8	92.14	100.00	90.28	75.58	97.66
1.0	92.14	100.00	91.65	72.27	97.66
1.2	88.33	100.00	89.33	74.68	97.66
1.4	93.05	100.00	89.26	70.13	97.66
1.8	93.05	99.07	87.49	74.20	97.66

TABEL VIII
TINGKAT RECALL PSO-SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neural	Anova
0.0	94.03	100.00	92.27	84.46	99.55
0.1	96.80	100.00	96.32	89.37	99.55
0.2	94.03	84.00	97.32	83.96	99.55
0.4	94.05	95.84	90.42	85.69	99.07
0.8	95.41	95.84	80.69	87.06	99.07
1.0	94.48	100.00	84.54	89.35	99.07
1.2	92.64	94.03	92.45	84.04	99.55
1.4	94.48	97.71	95.67	87.47	99.55
1.8	91.06	93.62	92.32	88.44	99.55

Dilihat dari Tabel VII dan Tabel VIII dapat dilihat dan dilakukan perbandingan nilai *recall* antara SVM dengan PSO-SVM. Ditarik kesimpulan bahwa optimasi dengan pemilihan atribut (parameter) yang tepat menggunakan PSO (*particle swarm optimization*) secara garis besar dapat meningkatkan nilai *recall* model SVM pada hampir semua jenis kernel yang diujikan.

Tingkat *recall* yang paling besar sebesar 99.55% diperoleh ketika menerapkan PSO-SVM dengan parameter C (pinalti) sebesar 0.0,0.1,0.2,1.2,1.4 dan 1.8 dengan menggunakan kernel *anova*.

Pengujian selanjutnya adalah mengukur nilai AUC (*area under curve*) pada kurva ROC. Pada Tabel IX dan Tabel X memperlihatkan perbandingan tingkat AUC pengklasifikasian data menggunakan SVM dibandingkan dengan PSO-SVM.

TABEL IX
NILAI AUC (AREA UNDER CURVE) SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neu-ral	Anova
0.0	0.926	0.969	0.835	0.729	0.991
0.1	0.935	0.894	0.911	0.892	0.995
0.2	0.928	0.935	0.936	0.880	0.996
0.4	0.925	0.952	0.836	0.879	0.994
0.8	0.930	0.952	0.850	0.776	0.995
1.0	0.935	0.969	0.837	0.749	0.996
1.2	0.923	0.948	0.912	0.777	0.994
1.4	0.936	0.947	0.946	0.826	0.994
1.8	0.937	0.946	0.835	0.724	0.995

TABEL X
NILAI AUC (AREA UNDER CURVE) PSO-SVM

Par C	Tipe Kernel				
	Dot	Radial	Poli-nomial	Neu-ral	Anova
0.0	0.938	0.916	0.755	0.822	0.987
0.1	0.938	0.915	0.817	0.799	0.994
0.2	0.935	0.915	0.830	0.755	0.996
0.4	0.922	0.915	0.718	0.756	0.983
0.8	0.934	0.916	0.718	0.734	0.996
1.0	0.930	0.916	0.815	0.757	0.995
1.2	0.936	0.916	0.806	0.751	0.994
1.4	0.936	0.916	0.843	0.752	0.985
1.8	0.937	0.914	0.838	0.756	0.994

Dilihat dari Tabel IX dan Tabel X dapat dilihat dan dilakukan perbandingan nilai AUC (*area under curve*) antara SVM dengan PSO-SVM. Ditarik kesimpulan bahwa optimasi dengan pemilihan atribut (parameter) yang tepat menggunakan metode PSO (*particle swarm optimization*) secara garis besar dapat meningkatkan nilai AUC (*area under curve*) model SVM pada beberapa jenis kernel yang diujikan seperti *radial* kernel.

Tingkat AUC (*area under curve*) yang paling besar sebesar 0.996% diperoleh ketika menerapkan PSO-SVM dengan parameter C (pinalti) sebesar 0.2 dengan menggunakan kernel *anova*.

Berdasarkan Tabel pengujian SVM dan PSO-SVM yang berbasiskan uji akurasi, presisi, *recall*, dan nilai AUC pada Tabel III, Tabel IV, Tabel V, Tabel VI, Tabel VII, Tabel VIII, Tabel IX, dan Tabel X diatas maka dapat ditarik kesimpulan bahwa PSO-SVM relatif lebih unggul dibandingkan SVM dengan performansi kernel *anova* sebagai kernel yang memberikan tingkat akurasi, presisi,

recall, dan nilai AUC paling besar dibandingkan dengan kernel-kernel yang lain.

Langkah selanjutnya dari penelitian ini adalah menguji model klasifikasi PSO-SVM terbaik kedalam *dataset* yang berbeda. Pada kasus ini digunakan *dataset* SMA XYZ dengan total data adalah 288 siswa. Model yang digunakan sebagai acuan prediksi adalah model PSO-SVM dengan kernel *anova* dengan parameter faktor pinalti (C) sebesar 0.1.

Hasil dari pengujian model kemudian dianalisis dengan melakukan pencocokan prediksi label dengan label peminatan sebenarnya sehingga didapatkan performansi akurasi, presisi, *recall* dengan menggunakan *confusion matrix*.

Berikut adalah Tabel XI hasil *confusion matrix* dari model PSO-SVM kernel *anova* parameter 0.1 yang diterapkan (implementasi) pada data peminatan siswa SMA XYZ.

TABEL XI. CONFUSION MATRIX DARI MODEL PSO-SVM KERNEL ANOVA PARAMETER C 0.1 PADA DATA TESTING SMA XYZ

PSO-SVM KERNEL ANOVA (C=0.1)	true IPA	true IPS	class precision
pred. IPA	150	2	98.23%
pred. IPS	0	136	100.00%
class recall	100.00%	98.82%	Akurasi : 99.29%

V. KESIMPULAN DAN SARAN

Dari hasil penelitian dan pembahasan yang dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

- 1) Penelitian dilakukan dengan melakukan komparasi antara algoritma SVM dengan PSO-SVM.
- 2) Untuk melakukan percobaan atau proses *training* digunakan *dataset* peminatan Siswa SMA ABC yang melibatkan 280 siswa.
- 3) Komparasi algoritma dilakukan dengan menggunakan beberapa kernel yaitu *dot*, *radial*, *polynomial*, *neural*, dan *anova kernel*.
- 4) Optimasi yang dilakukan dengan menggunakan algoritma PSO (*particle swarm optimization*) terbukti dapat meningkatkan nilai akurasi, presisi, *recall*, AUC dari model SVM yang dibangun. Nilai performansi paling tinggi yang diperoleh PSO-SVM adalah pada pengujian kernel *anova* dengan nilai akurasi sebesar

99.30%. Dengan nilai akurasi lebih dari 70% dapat dikatakan bahwa pembangunan model klasifikasi menggunakan PSO- SVM berhasil.

- 5) Model yang telah terbentuk digunakan untuk proses prediksi/klasifikasi data peminatan yang ada di sekolah SMA XYZ dengan jumlah data adalah 288 siswa.
- 6) Performansi prediksi yang diimplementasikan mendapatkan akurasi sebesar 99.29%. Dengan hasil akurasi prediksi lebih dari 70%, maka model PSO-SVM ini layak digunakan untuk *dataset* sekolah lainnya.

Beberapa saran kedepan untuk dapat diperbaiki (*improve*) dari penelitian ini adalah:

- 1) Hasil uji coba pemilihan kernel yang sesuai dapat digantikan dengan *automatic kernel selection* sehingga tidak memerlukan waktu yang banyak untuk pemilihan *kernel* yang tepat.
- 2) Hasil uji coba pemilihan variasi faktor pinalti (C) dapat digantikan dengan metode *parameter selection* sehingga tidak memerlukan waktu yang banyak untuk pemilihan faktor pinalti (C) yang tepat.

DAFTAR PUSTAKA

- [1] F. Jui-Hsi, C. Jui-Hung, H. Yueh-Min, and C. Han-Chieh, "A Support Vector Regression based Prediction of Student's School Performance", Computer, Consumer and Control (IS3C), International Symposium, pp.84-87, 2012.
- [2] P. Nimit, M. Masayuki, M. Michihiko, "Learning to Estimate Slide Comprehension in Classroom with Support Vector Machines". IEEE Computer Society, vol. 5, pp.52-61, March. 2012.
- [3] B. Theopilus, "Klasifikasi Jalur Minat Siswa Menggunakan Algoritme Support Vector Machine (SVM) (Kasus: SMA Negeri 1 dan SMA Negeri 2 Sragen)", Tesis, Jurusan Teknik Elektro dan Teknologi Informasi, UGM, Yogyakarta, 2014.
- [4] J.T. Chung, Y.C. Li, C.Y. Jun, and Y. Cheng-Hong, "Feature Selection using PSO-SVM", IAENG International Journal of Computer Science, IJCS, February, 2007.
- [5] A. Fatima and S. Kaddour, "Optimization of SVM Multiclass by Particle Swarm (PSO-SVM)". Modern Education and Computer Science (MECS), vol. 2, pp.32-38, 2010.
- [6] I.H. Witten, E. Frank, and M.A. Hall, "Data Mining Practical Machine Learning Tools and Techniques", Third Edition, Elsevier Publisher, USA, 2011.
- [7] J. Agustinus, "Sistem Deteksi Intrusi Jaringan dengan Metode Support Vector Machine", M. Eng. Thesis, Jurusan Ilmu Komputer FMIPA UGM, Yogyakarta, 2012.
- [8] J. Brownlee, "Clever Algorithms," 2013. [Online]. Available: <http://www.cleveralgorithms.com/nature-inspired/swarm.html>. [Accessed 28 Juni 2016].
- [9] B. Santosa, "Tutorial Particle Swarm Optimization," [Online]. Available: <http://oc.its.ac.id/ambilfile.php?i dp=1221>. [Accessed 28 Juni 2014].
- [10] N. Dhanasaputra and B. Santosa, "Pengembangan Algoritma Cat Swarm Optimization (CSO) untuk Klasifikasi". Jurnal ITS, p. 3, 2010.
- [11] J. Han, and M. Kamber, "Data mining: Concepts and Techniques", Third Edition, Morgan Kaufmann Publishers, San Fransisco, 2013.
- [12] S. Ali, and A. Smith, "Improved Support Vector Machine Generalization Using Normalized Input Space". AI 2010: Advances in Artificial Intelligence, vol.4304, pp.362-371,2010.