JOIN (Jurnal Online Informatika)
Volume 2 No. 2 | Desember 2017 : 84-91
DOI: 10.15575/join.v2i2.109

Pengenalan Wajah Menggunakan SVM Multi Kernel dengan Pembelajaran yang Bertambah

Muhammad Athoillah
Universitas PGRI Adi Buana, Jl. Ngagel Dadi III-3B No.37, Wonokromo, Kota Surabaya, 60234
Athoillah.Muhammad@gmail.com

Abstract-Automatic face recognition has an important role in the life of today's society. Basically face recognition problem can be solved with classification method or algorithm, one of them is Support Vector Machine (SVM). Although it is very good to solve classification problem, SVM can only classify linier separable data. So to be able to classify non-linier separable data, SVM must be modified using kernel function. It is hard to find the best suitable kernel function for every characteristics data. To solve that difficulties, the researchers develop the method by combine some kernel function to be one kernel function, this method call Multi Kernel SVM. In this framework we build a face recognition system using Multi Kernel SVM with Incremental Learning. It means if there is additional data or new information or class, this system does not have to delete the existing learning information and restart the system to get the new learning information, in other words this method make the system more dynamic. The result of this framework shows the system that has built can recognize human face well, this proved by the average of all accuracy value which is achieve 89%, then the precision value is 41.67% and recall value is 47.67%. The framework result also shows the system with incremental learning method only need 6.4856 second in average of all experiment.

Keywords **- Face Recognition, Incremental Learning, Multi Kernel, SVM**

Abstrak**-Pengenalan wajah secara otomatis merupakan sebuah kebutuhan yang memiliki peran penting dalam kehidupan masyarakat saat ini. Pada dasarnya masalah pengenalan wajah dapat dipecahkan dengan menggunakan algoritma atau metode klasifikasi, salah satunya adalah *Support Vector Machine* (SVM). Walaupun sangat baik dalam menyelesaikan permasalahan klasifikasi, SVM hanya dapat digunakan pada data yang bersifat linear saja, sehingga untuk dapat digunakan pada data non-linear maka SVM dimodifikasi dengan menggunakan fungsi kernel. Sulitnya menemukan fungsi kernel yang sesuai dengan karakteristik data yang dipakai membuat para peneliti melakukan pengembangan dengan menggunakan kombinasi dari beberapa kernel atau disebut Multi kernel. Pada penelitian ini dibangun sebuah sistem pengenalan wajah yang berbasis SVM multi kernel dengan metode pembelajaran yang bertambah, artinya apabila terjadi penambahan data atau informasi baru tidak harus menghapus pengetahuan yang lalu dan mengulang pembelajarannya dari awal. Sehingga pembelajaran sistem yang dibangun lebih dinamis. Hasil yang didapat pada penelitian ini menunjukkan bahwa sistem yang dibangun dapat mengenali wajah manusia dengan baik, hal ini dibuktikan dengan nilai rata-rata keseluruhan akurasi yang mencapai 89%, kemudian nilai *precision* 41.67% serta nilai *recall* sebesar 47.67%. Hasil penelitian juga menunjukkan bahwa dengan metode pembelajaran yang bertambah, sistem hanya membutuhkan waktu selama 6.4853 detik secara rata-rata pada keseluruhan uji coba.**

Kata kunci **- Multi Kernel, Pembelajaran yang Bertambah, Pengenalan Wajah, SVM**

I. PENDAHULUAN

Manusia memiliki bentuk wajah yang berbeda antara satu dengan lainnya, perbedaan inilah yang menjadikannya salah satu identitas bagi tiap manusia. Wajah sebagai objek dua dimensi digambarkan dengan berbagai macam illuminasi, pose dan ekspresi wajah untuk diidentifikasi berdasarkan citra dua dimensi dari wajah tersebut [1]. Seiring dengan berkembangnya zaman, kebutuhan akan mesin cerdas yang mampu mengidentifikasi identitas dari sebuah wajah sangatlah diperlukan, hal ini tentunya akan sangat membantu dalam banyak aspek kehidupan, mulai dari sistem keamanan, penanganan kejahatan criminal, identifikasi pelaku pelanggaran lalu-lintas atau alat-alat otomatis lain khususnya yang berhubungan dengan keamanan dan privasi dari seseorang.

Pada dasarnya, sistem pengenalan wajah bekerja dengan membandingkan citra masukan *(input)* dengan citra yang telah tersimpan dalam sebuah *database* dan menemukan kecocokan wajah yang paling sesuai dengan data masukan yang ada sebelumnya. Hal ini berarti masalah pengenalan wajah dapat dipecahkan dengan menggunakan algoritma atau metode klasifikasi. Salah satu algortima yang mampu menyelesaikan masalah klasifikasi dengan baik adalah *Support Vector Machine*

ISSN 2527-1682 (Print)
ISSN 2527-9165 (Online)

(SVM) [2][3]. SVM bekerja dengan cara mendefinisikan batas antara dua kelas dengan jarak maksimal dari data yang terdekat [4], jarak maksimal ini didapatkan dengan menemukan *hyperplane* (garis pemisah) terbaik pada *input space* yang diperoleh dengan mengukur *margin hyperplane*, m*argin* merupakan jarak antara *hyperplane* dengan titik terdekat dari masing-masing kelas.

Walaupun sangat baik dalam menyelesaikan permasalahan klasifikasi, SVM hanya dapat digunakan pada data yang bersifat linier saja, sehingga diperlukan sebuah pengembangan untuk dapat membuat SVM mampu memisahkan data non-linier, salah satunya dengan menambahkan fungsi kernel. Dengan menggunakan fungsi kernel data yang pada awalnya tidak dapat dipisahkan secara linear akan dibawa ke dimensi yang lebih tinggi sehingga pada dimensi baru tersebut *hyperplane* dapat dikonstruksikan. Pengunaan fungsi kernel tidak selalu memberikan hasil klasifikasi yang maksimal, karena setiap fungsi memiliki karakteristik tersendiri dan tidak selalu sesuai dengan karakteristik data yang diklasifikasikan. Karena sulitnya menentukan kernel yang terbaik, maka para peneliti di bidang pembelajaran mesin mencoba untuk mengembangkan pembelajaran kernel yang lebih fleksibel yaitu dengan pembelajaran multi kernel atau lebih dikenal dengan *Multiple Kernel Learning (MKL)* [2].

Setiap proses pelatihan atau *training* pada algoritma pembelajaran mesin pada umumnya berbentuk satuan kelompok, dimana hal tersebut berarti setiap data *training* memiliki prioritas yang sama selama proses *training* berlangsung. Ini berarti jika terjadi penambahan data *training* baru akan merubah batas parameter pada data yang telah di-*training* sebelumnya. Sehingga, meskipun data penambahan berjumlah relatif sedikit dan berpengaruh kecil pada hasil klasifikasi sebelumnya, pembelajaran yang baru tetap harus dilakukan sehingga seolah-olah pembelajaran yang lalu tidaklah berarti. Untuk itulah dikembangkan sebuah pendekatan baru yang dinamakan dengan pembelajaran yang bertambah (*incremental learning*), dimana dengan cara ini proses pembelajaran mesin akan lebih dinamis [5]. Pembahasan lebih lanjut dapat dilihat pada bab selanjutnya.

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, pada penelitian ini akan dibangun sebuah sistem pengenalan wajah dengan menggunakan SVM Multi Kernel serta dengan menggunakan teknik pembelajaran yang bertambah sehingga pembelajaran sistem yang dibangun lebih dinamis.

II. DASAR TEORI

A. PENGENALAN WAJAH

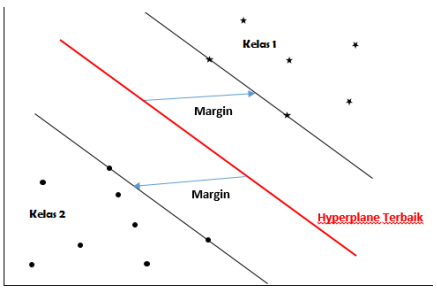
Manusia memiliki bentuk wajah yang berbeda antara satu dengan lainnya, perbedaan inilah yang menjadikannya salah satu identitas bagi tiap manusia. Wajah sebagai objek dua dimensi digambarkan dengan berbagai macam illuminasi, pose dan ekspresi wajah untuk diidentifikasi berdasarkan citra dua dimensi dari wajah tersebut [1].

Sistem pengenalan wajah merupakan sistem yang secara otomatis dapat mengidentifikasi atau memverifikasi seseorang dari sebuah gambar digital dan atau bingkai video dari kamera pengawas [1]. Pada dasarnya, sistem pengenalan wajah bekerja dengan membandingkan citra masukan (input) dengan citra yang telah tersimpan dalam sebuah *database* dan menemukan kecocokan wajah yang paling sesuai dengan data masukan yang ada sebelumnya.

Gambar 1. Ilustrasi Pengenalan Wajah

B. *SUPPORT VECTOR MACHINE* (SVM)

Support Vector Machine (SVM) merupakan metode klasifikasi yang diperkenalkan pertama kali oleh Vapnik pada tahun 1998. Pada dasarnya, metode ini bekerja dengan cara mendefinisikan batas antara dua kelas dengan jarak maksimal dari data yang terdekat [4]. Untuk mendapatkan batas maksimal antar kelas maka harus dibentuk sebuah *hyperplane* (garis pemisah) terbaik pada input space yang diperoleh dengan mengukur *margin hyperplane* dan mencari titik maksimalnya. *Margin* merupakan jarak antara *hyperplane* dengan titik terdekat dari masing-masing kelas. Titik terdekat inilah yang disebut sebagai *support vector* [6].



Gambar 2. Ilustasi Pemisahan kelas pada SVM

Hyperplane terbaik seperti yang terlihat pada Gambar 2 dapat ditemukan dengan

$$\max_{\alpha} L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j$$

dengan syarat

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \qquad \alpha_i \geq 0$$

Dengan $\{x_1, \ldots, x_n\}$ adalah *dataset* input, dan $y_i \in \{+1, -1\}$ adalah label kelas dari data x_i. Dengan hasil

tersebut akan terdapat nilai α_i untuk setiap data *training*. Data *training* yang memiliki nilai $\alpha_i > 0$ adalah *support vector* sedangkan sisanya memiliki nilai $\alpha_i = 0$

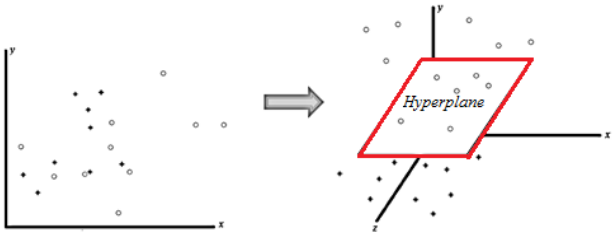
Setelah solusi permasalahan *quadratic programming* ditemukan (nilai α_i), maka kelas dari data pengujian x dapat ditentukan berdasarkan nilai dari fungsi keputusan:

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i x_i x_d + b$$

x_i adalah *support vector*, ns = jumlah *support vector* dan x_d adalah data yang diklasifikasikan.

C. MULTI KERNEL SVM

Tidak semua data dapat dipisahkan secara linier, sedangkan SVM pada dasarnya hanya mampu memisahkan data secara linier saja, sehingga diperlukan sebuah pengembangan untuk dapat membuat SVM mampu memisahkan data non-linier, salah satunya dengan menambahkan fungsi kernel. Dengan menambahkan fungsi kernel pada SVM nantinya data x akan dipetakan ke ruang vektor yang lebih tinggi hingga *hyperplane* dapat dikonstruksikan [7].



Gambar 3. Ilustasi *Hyperplane* pada dimensi yang lebih tinggi

Pemetaan tersebut bertujuan untuk tetap menjaga topologi data, artinya dua data yang semula dekat pada *input space* akan tetap dekat pada *feature space*, begitu pula sebaliknya data yang pada awalnya jauh akan tetap jauh. Kemudian dari data yang sudah ditransformasikan pada ruang yang lebih tinggi tersebut dilakukan perhitungan *dot product* untuk mendapatkan titik-titik *support vector*. Karena sulitnya menemukan fungsi dari transformasi atau fungsi Φ tersebut, maka perhitungan *dot product* dapat digantikan dengan fungsi kernel $K(x_i,x_j)$ yang mendefinisikan fungsi transformasi sebelumnya secara implisit inilah yang disebut dengan *"kernel trick"* [8] dirumuskan dengan:

$$K(x_i,x_j) = \langle \Phi(x_i).\Phi(x_j) \rangle$$

berikut adalah beberapa jenis fungsi kernel yang biasa digunakan dalam penelitian menurut [7], yaitu:

a) Kernel Linier: $K(x_i,x_j) = x_i x_j^T$
b) Kernel Polynomial: $K(x_i,x_j) = (x_i x_j^T + 1)^p$
c) Kernel RBF :$K(x_i,x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$

Pengunaan fungsi-fungsi kernel tersebut tidaklah selalu baik untuk memecahkan persoalan klasifikasi yang ditemui, karena setiap fungsi memiliki karakteristik tersendiri dan tidak selalu sesuai dengan karakteristik data yang diklasifikasikan sehingga tidak jarang menghasilkan klasifikasi yang tidak maksimal. Karena sulitnya menentukan kernel yang terbaik, seiring dengan berkembangnya penelitian, para peneliti di bidang pembelajaran mesin mencoba untuk mengembangkan pembelajaran kernel yang lebih fleksibel yaitu dengan pembelajaran multi kernel atau lebih dikenal dengan *Multiple Kernel Learning (MKL)* [2]. Beberapa penelitian terbaru menunjukkan bahwa pembelajaran multi kernel mampu memberikan hasil yang lebih baik dalam memprediksi maupun mengklasifikasi [9][10].

Jika fungsi kernel didefinisikan dengan $K(x,x')$ maka *Multiple Kernel Learning (MKL)* didefinisikan dengan

$$K(x,x') = \sum_{m=1}^{M} d_m K_m(x,x')$$

dengan kendala

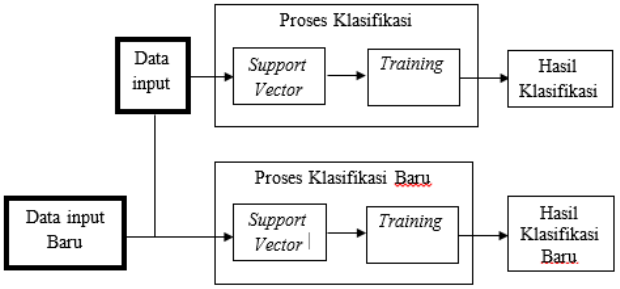
$$d_m \geq 0, \sum_{m=1}^{M} d_m = 1$$

M adalah jumlah dari fungsi kernel [2].

D. PEMBELAJARAN YANG BERTAMBAH

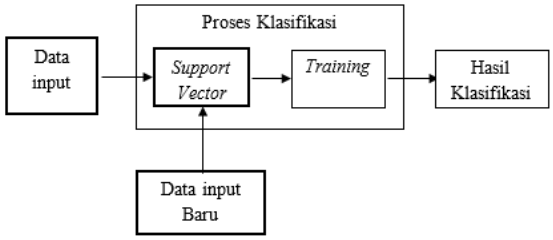
Setiap proses pelatihan atau *training* pada algoritma pembelajaran mesin pada umumnya berbentuk satuan kelompok, dimana hal tersebut berarti setiap data *training* memiliki prioritas yang sama selama proses *training* berlangsung. Hal ini menjadi sebuah kerugian jika pada suatu kasus terdapat penambahan data training baru, penambahan data *training* baru akan mengubah batas parameter pada data yang telah di-*training* sebelumnya. Artinya, meskipun data penambahan berjumlah relatif sedikit dan berpengaruh kecil pada hasil klasifikasi sebelumnya pembelajaran yang baru tetap harus dilakukan sehingga seolah-olah pembelajaran yang lalu tidaklah berarti.

Jika algoritma pada umumnya hasil dari *training* ditentukan oleh seluruh data yang ikut diproses pada *training*, maka pada SVM yang mempengaruhi hasil *training* adalah *support vector*-nya. Dengan karakteristik tersebut masalah pembelajaran mesin yang menjadi lambat pada proses training dengan menggunakan data besar dapat diatasi, yaitu dengan melakukan pembelajaran yang bertambah (*incremental*) dengan cara memilih kandidat *support vector* sebelum *training*, sehingga mampu mengurangi jumlah data pada saat proses *training* yang tentunya mengurangi kebutuhan akan memori dan waktu [11] seperti yang ditunjukkan pada Gambar 4 dan Gambar 5 berikut :



Gambar 4. Proses Pembelajaran Pada Umumnya

ISSN 2527-1682 (Print)
ISSN 2527-9165 (Online)



Gambar 5. Proses Pembelajaran yang Bertambah dengan SVM

Setiap proses training dengan SVM menghasilkan tiga kondisi data x_i, yaitu data x_i yang memenuhi $\alpha_i = 0$ atau data yang berada diluar dari *margin*, data x_i yang memenuhi $0 \leq \alpha_i \leq C$ atau disebut data *support vector*, serta data dengan $C = \alpha$ merupakan data yang berada didalam *margin*. Atau dapat dirumuskan sebagai berikut:

$$\alpha_i = 0 \rightarrow |f(x_i)| \geq 1$$

$$0 \leq \alpha_i \leq C \rightarrow |f(x_i)| = 1$$

$$\alpha = C \rightarrow |f(x_i)| \leq 1$$

Dimana setiap data yang berada pada bidang pembatas $(x) = \pm 1$ merupakan data *support vector*. Jika pada persamaan sebelumnya diketahui bahwa mendapatkan *hyperplane* terbaik adalah dengan menemukan nilai α_i pada persamaan

$$\max_{\alpha} L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j$$

dengan syarat

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \qquad \alpha_i \geq 0$$

Kemudian berdasarkan [2] dengan penambahan fungsi multi kernel persamaan optimasinya didapatkan menjadi

$$\max_{\alpha} -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \sum_{m}^{M} d_m K_m(x_i, x_j) + \sum_{i=1}^{n} \alpha_i$$

dengan

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \ \forall i$$

Kemudian dengan memisalkan

$$H_{i,j} = y_i y_j \sum_{m}^{M} d_m K_m(x_i, x_j)$$

maka

$$\max_{\alpha} -L_D = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i H_{ij} \alpha_j - \sum_{i=1}^{n} \alpha_i$$

Jika diturunkan terhadap α diperoleh

$$P_i = \frac{\partial L}{\partial \alpha_i} = \frac{1}{2} \sum_{i,j=1}^{n} H_{ij} \alpha_j - 1$$

Kemudian apabila s merupakan penambahan data baru maka P_i berubah mengikuti:

$$\Delta g_i = H_{i,s} \Delta \alpha_s + \sum_{i,j=1}^{n} H_{ij} \alpha_j$$

$$0 = y_s \Delta \alpha_s + \sum_{j \in 1}^{n} y_j \Delta \alpha_j$$

Dengan kata lain jika terdapat penambahan data baru s, maka

$$P_s = H_{ss} \Delta \alpha_s + \sum_{s,j}^{n} H_{sj} (\alpha_j + \Delta \alpha_j) - 1 = 0$$

dengan s adalah *support vector* sehingga α dapat diperbaharui [12] menurut persamaan berikut

$$\Delta \alpha_s = \frac{\sum_{j \in s}^{n} H_{sj} \alpha_j - 1}{H_{ss}}$$

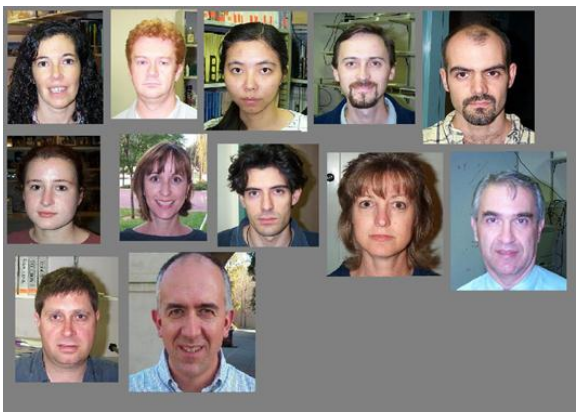
dan

$$\alpha = (\alpha + \Delta \alpha, \alpha_s)^T$$

III. RANCANG BANGUN SISTEM

A. *DATASET*

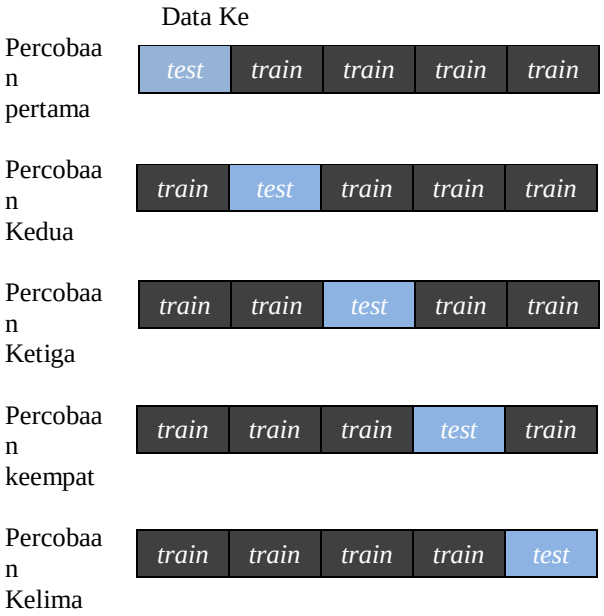
Dataset yang digunakan dalam penelitian ini merupakan data dari citra wajah berwarna dengan jumlah 10 macam wajah dengan masing-masing berjumlah 20 citra dengan latar belakang dan ekspresi yang berbeda satu sama lain. Kemudian untuk proses penambahan kelas dilakukan 2 kali penambahan dengan masing – masing berjumlah 15 citra untuk penambahan kelas pertama dan 10 citra untuk penambahan kelas kedua.



Gambar 6. Citra Data Penelitian

Untuk proses validasi sistem yang telah dibangun, dilakukan dengan menggunakan metode *k-fold cross validation* [13]. Dimana data akan dibagi menjadi 2 bagian yaitu 80% dari data citra digunakan untuk proses *training* dan 20% dari data citra digunakan untuk proses *testing*. Dari pembagian tersebut akan dikelompokkan

menjadi 5 kelompok dengan 4 kelompok sebagai kelompok data *training* dan 1 kelompok sebagai data *testing* kemudian dilakukan percobaan sebanyak 5 kali dengan data *training* dan *testing* yang berbeda, sesuai dengan skema berikut:



Gambar 7. Skema *k-fold cross validation*

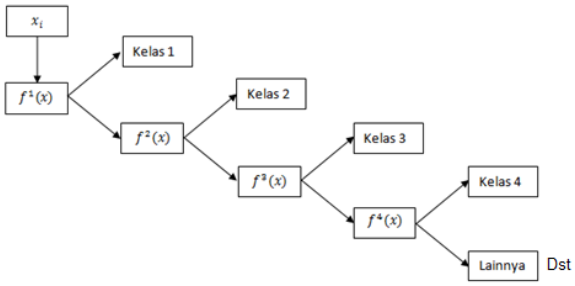
B. IMPLEMENTASI

Pada tahap implementasi, hal pertama yang dilakukan pada penelitian ini adalah dengan melakukan pre-proses pada data citra, yaitu dengan mendapatkan histogram pada citra tersebut. Histogram didapatkan dengan cara menghitung frekuensi kemunculan nisbi (*relative*) dari intensitas pada citra tersebut. Misalkan sebuah citra memiliki L derajat keabuan (0 sampai L-1) atau dapat dirumuskan dengan:

$$h_i = \frac{n_i}{n}, \quad i = 0,1, \dots, L-1$$

dengan:
n_i adalah jumlah *pixel* yang memiliki derajat keabuan i
n adalah jumlah seluruh *pixel* di dalam citra

Setelah histogram didapatkan maka selanjutnya adalah melakukan pembelajaran atau *training* pada data tersebut. Karena SVM hanya dapat mengklasifikasikan data secara biner (hanya dua kelas saja) maka dilakukan teknik pembelajaran dengan cara *One-against-all*. Untuk mendapatkan hasil klasifikasi dengan banyak kelas. Adapun skema pembelajaran dengan metode *One-against-all* adalah sebagaimana yang ditunjukkan pada Gambar 8 berikut:



Gambar 8. Skema klasifikasi *one-against-all*

Pada proses *training* ini, fungsi kernel yang digunakan adalah fungsi multi kernel yang didapatkan dengan cara mengkombinasikan fungsi-fungsi kernel diantaranya kernel linier, kernel polynomial dengan order 3 dan kernel RBF. Kombinasi ini dilakukan sesuai dengan persamaan

$$K(x, x') = \sum_{m=1}^{M} d_m K_m(x, x')$$

dengan kendala

$$d_m \geq 0, \sum_{m=1}^{M} d_m = 1$$

dengan M adalah jumlah dari fungsi kernel [2].

C. HASIL UJI COBA

Berikut adalah *user interface* dari aplikasi pengenalan wajah yang telah dibagun dalam penelitian ini.

Gambar 9. *User Interface* Sistem Pengenalan Wajah

Adapun beberapa spesifikasi perangkat keras dan perangkat lunak yang digunakan pada uji coba sistem adalah sebagai berikut:

Tabel 1. Spesifikasi Lingkungan Perancangan Sistem

Perangkat Keras	Prosesor	:	Intel® Core™ i5-5200U CPU @ 2,20 GHz
	Memory	:	4 GB DDR4
Perangkat Lunak	Sistem Operasi	:	Windows 8.1 Pro
	Tools	:	MatlabR2012b x64

ISSN 2527-1682 (Print)
ISSN 2527-9165 (Online)

Uji coba dilakukan dengan menghitung waktu yang dibutuhkan sistem pada saat proses *training dataset* citra input, proses *training* dilakukan sebanyak tiga kali, dimana proses *training* yang pertama dilakukan terhadap 10 macam wajah dengan jumlah 160 citra, kemudian *training* berikutnya adalah proses pembelajaran bertambah yaitu melakukan *training* dengan menambahkan satu kelompok macam wajah berjumlah 12 citra, kemudian untuk *training* pada *update* atau penambahan kelas kedua dilakukan dengan menambahkan satu kelompok wajah berjumlah 8.

Untuk proses *testing*, dilakukan uji coba dengan cara menguji kelompok *database testing* yeng terdiri dari 10 macam wajah dengan masing-masing berjumlah 4 citra sehingga jumlah keseluruhan data testing pada kelompok 10 wajah awal adalah 40 citra. Berikutnya untuk proses *testing* pada hasil klasifikasi penambahan kelas yang pertama, data citra pada *dataset testing* ditambahkan 3 citra sehingga jumlah total wajah yang diuji pada hasil klsifikasi dengan penambahan satu kelas pertama adalah 43 citra demikian juga dengan penambahan kelas yang kedua ditambahkan jumlah citra *testing* sebanyak 2 citra. Dari percobaan tersebut didapatkan sejumlah citra yang dikenali dengan benar maupun yang tidak, dari jumlah tersebut kemudian dilakukan perhitungan *precision*, *recall* dan akurasi dengan rumusan sebagai berikut:

Tabel 2. *Confusion matrix* nilai prediksi dan nilai sebenarnya

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	*True Positive*	*False Positif*
	False	*False Negative*	*True Negative*

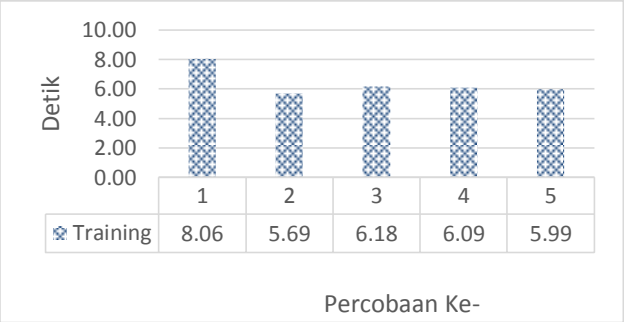
$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} x\ 100\%$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} x\ 100\%$$

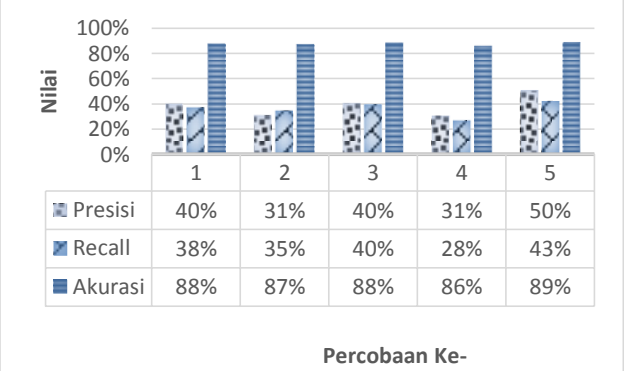
$$\text{Akurasi} = \frac{True\ Positive + True\ Negative}{\text{Jumlah keseluruhan data}} x\ 100\%$$

Dari uji coba yang telah dilakukan didapatkan hasil sebagaimana yang ditunjukkan oleh grafik berikut:

1. Hasil uji coba untuk pengenalan 10 macam wajah

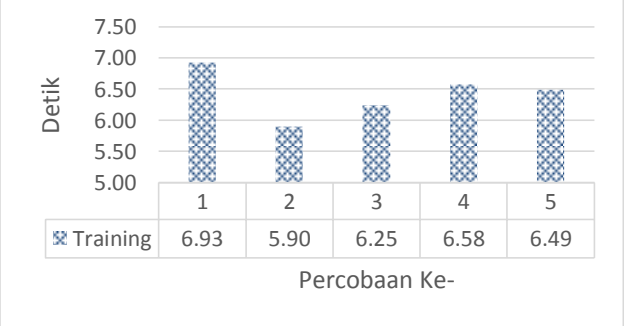


Gambar 10. Waktu Komputasi Proses *Training* 10 Macam Wajah

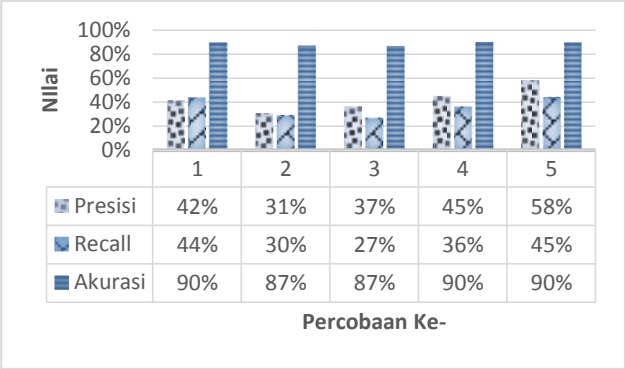


Gambar 11. Nilai *Precision*, R*ecall*, Akurasi Hasil Klasifikasi 10 Macam Wajah

2. Hasil uji coba untuk pengenalan 11 macam wajah atau penambahan satu kelas baru (*update*) yang pertama

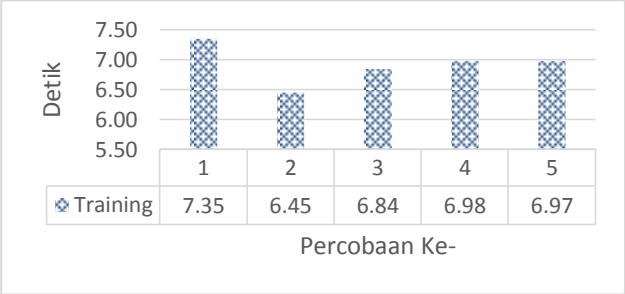


Gambar 12. Waktu Komputasi Proses *Training* Pada *Update* Kelas Pertama

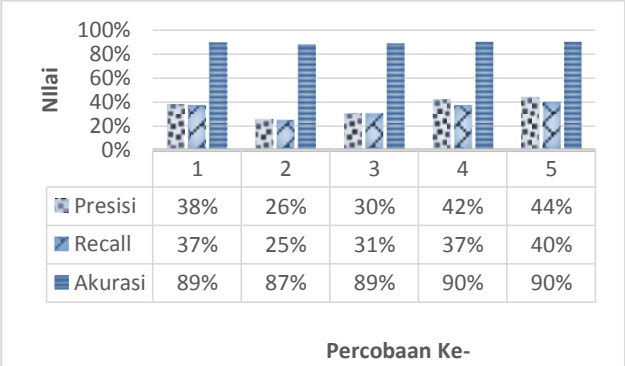


Gambar 13. Nilai *Precision*, *Recall*, Akurasi Hasil Klasifikasi Pada *Update* Kelas Pertama

3. Hasil uji coba untuk pengenalan 12 macam wajah atau penambahan satu kelas baru (*update*) yang kedua



Gambar 14. Waktu Komputasi Proses *Training* Pada *Update* Kelas Kedua



Gambar 15. Nilai *Precision*, *Recall*, Akurasi Hasil Klasifikasi Pada *Update* Kedua

Dari hasil yang ditunjukkan Gambar 10 sampai dengan Gambar 15 jika dihitung rata-rata, terlihat bahwa pengenalan wajah dengan menggunakan SVM multi kernel memiliki performa yang baik dengan rata-rata akurasinya mencapai 89% dengan 10 macam wajah yang dikenali. Kemudian dengan metode pembelajaran yang bertambah yang dilakukan dengan dua kali penambahan kelas, nilai rata-rata akurasinya cenderung tidak berubah yaitu tetap 89%. Di lain sisi, nilai *precision* pada 10 kelas awal memiliki rata-rata 38% dan tidak banyak berubah pada saat penambahan kelas yaitu 43% dan 44%. Untuk nilai *Recall* pada 10 kelas awal nilai rata-ratanya mencapai 43% kemudian cenderung menurun menjadi 37% dan 35% saat penambahan kelas pertama dan kedua.

Jika dilihat dari segi waktu komputasi selama proses *training*, waktu rata-rata yang dibutuhkan sistem untuk melakukan *training* sebanyak 160 citra adalah 5.993 detik sedangkan saat terjadi penambahan kelas yang artinya data input yang di *training* pun bertambah waktu rata-rata yang dibutuhkan untuk proses *training* tidaklah banyak berubah yaitu 6.494 detik dan 6.969 detik. Untuk lebih mudahnya nilai rata-rata hasil dari uji coba dapat dilihat pada Tabel 3 berikut ini:

Tabel 3. Hasil Uji Coba Pengenalan Wajah

	Waktu *Training*	*Precision*	*Recall*	Akurasi
I	5.993	38%	43%	89%
II	6.494	43%	37%	89%
III	6.969	44%	35%	89%

Ket:
I = Pengenalan 10 kelas Awal
II = Pengenalan dengan penambahan 1 kelas pertama
III= Pengenalan dengan penambahan 1 kelas kedua

Dari hasil uji coba, bisa diketahui bahwa pengenalan wajah menggunakan SVM multi kernel dengan pembelajaran yang bertambah memilik performa yang baik dari segi waktu maupun dari segi *precision*, *recall* dan akurasinya. Dari hasil performa saat penambahan kelas menujukkan bahwa tidak ada perubahan atau penurunan performa yang signifikan meskipun pembelajaran (*training*) yang dilakukan saat penambahan kelas tidak menggunakan data keseluruhan dari pembelajaran yang sebelumnya atau dengan kata lain tidak mengulang pembelajaran baru dengan adanya penambahan kelas atau data baru.

IV. PENUTUP DAN KESIMPULAN

Pada penelitian ini dibangun sebuah sistem pengenalan wajah dengan menggunakan metode SVM multi kernel dengan pembelajaran yang bertambah. Metode klasifikasi SVM multi kernel dilakukan dengan mengkombinasikan kernel-kernel diantaranya Kernel Linear, Kernel Polynomial, dan Kernel RBF. Sedangkan metode pembelajaran bertambah dilakukan dengan mengubah *support vector* pada pembelajaran sebelumnya menjadi data input pada pembelajaran selanjutnya, sehingga dapat mereduksi jumlah data pembelajaran (*training*) yang pada akhirmya membuat sistem berjalan lebih efisien. Hasil yang diperoleh dari uji coba sistem menunjukkan bahwa sistem dapat mengenali citra wajah dengan baik ditunjukkan dengan nilai rata-rata *precision*, *recall*, akurasi maupun kebutuhan waktu komputasi yang baik.

Penelitian selanjutnya, diharapakan sistem pengenalan wajah dengan SVM multi kernel dapat dilakukan dengan kombinasi dari fungsi kernel yang berbeda dari yang telah

ISSN 2527-1682 (Print)
ISSN 2527-9165 (Online)

dilakukan pada penelitian ini, atau dengan menggunakan pre-proses selain dengan menggunakan nilai histogram pada *database* citra yang digunakan. Metode pembelajaran yang bertambah juga dapat dicoba dengan mengaplikasikannya pada jumlah data yang jauh lebih besar atau dengan objek lain selain citra.

V. REFERENSI

[1] Kurniawan, D.E, 2012. Rancang Bangun Sistem Pengenalan Wajah Menggunakan Filter Gabor. Universitas Diponegoro, Semarang.

[2] Rakotomamonjy, A., Bach, F.R., Canu, S. Dan Grandvalet, Y. (2008). SimpleMKL. *Journal Machine Learning Research 9*, hal 2491-2521.

[3] Athoillah, M., Irawan, M.I., dan Imah, E.M. (2015). Study Comparison of SVM-, K-NN- and Backpropagation-Based Classifier for Image Retrieval. *Journal of Computer Science and Information*, Vol 8, No 1, hal-11-19

[4] Clarke, B., Fokoue, E., dan Zhang, H.H. (2009). Principles and Theory for Data Mining and Machine Learning, *Springer Science + Bussiness Media*, New York, USA.

[5] Gosselin, P.H., Precioso, F. dan Philip-Foliguet, S. (2010). Incremental Kernel Learning for Active Image Retrieval without Global Dictionaries, Jurnal *Pattern Recognition 44*, hal 2244-2254

[6] Campbell, C dan Ying, Y. (2011). Learning with Support Vector Machines, Buku seri *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Morgan & Claypool Publisher, UK

[7] Scholkopf, B dan Smola, A.j. (2001). Learning with Kernels, Buku seri *The Adaptive Computations and Machine Learning*, MIT Press, Massachusetts Institute of Technology, USA.

[8] Hamel, L. (2009). Knowledge Discovery with Support Vector Machine Learning, John Wiley & Sons, Inc, New Jersey, USA

[9] Shrivastava, A., Pillai, J.K., dan Patel, V.M. (2015). Multiple Kernel-based Dictionary Learning for Weakly Supervised Classification, *Jurnal Pattern Recognition*, Vol 48, No 8, Hal 2667-2675

[10] Athoillah, M., Irawan, M.I., and Imah, E.M, (2015). Support Vector Machine with Multiple Kernel Learning for Image Retrieval. *Proceeding of International Conference on Information, Communication Technology and System (ICICTS)*, ITS, Surabaya.

[11] Shinya, K dan Shigeo, A. (2006). Incremental Training of Support Machine Learning using Truncated Hypercones, *Lecture Notes in Computer Science-Artificial Neural Network in Pattern Recognition*, Kobe University Repository, Jepang. Hal 153-164

[12] Zhang, Y., Hu, G., Zhu, F., dan Yu, J., (2009). A new Incremental Learning Support Vector Machine, *proceeding of International Converence on Artificial Intelligence and Computational Intelligence AICI*, Shanghai, hal 7-10.

[13] Mohri, M., Rostamizadeh, A. dan Talwalkar, A. (2012). Foundations of Machine Learning, Buku seri *The Adaptive Computations and Machine Learning*, Massachusetts Institute of Technology. USA: MIT Press.