

PENGEMBANGAN SISTEM PENGENALAN WAJAH DENGAN METODE PENGKLASIFIKASIAN HIBRID BERBASIS JARINGAN FUNGSI BASIS RADIAL DAN POHON KEPUTUSAN INDUKTIF

Rully Soelaiman¹, Diana Purwitasari², Ariadi Retno Tri Hayati²

¹Jurusan Sistem Informasi, Fakultas Teknologi Informasi ITS, Jl. Raya ITS, Sukolilo, Surabaya, 60111

²Jurusan Teknik Informatika, Fakultas Teknologi Informasi ITS, Jl. Raya ITS, Sukolilo, Surabaya, 60111
e-mail : rully@its-sby.edu

ABSTRAK: Pengklasifikasian wajah berkaitan dengan variasi data misalnya detil - detil kecil dari wajah atau transformasi saat proses pengambilan citra. Pengklasifikasian wajah dengan metode hibrid menggabungkan pembelajaran berbasis Jaringan Fungsi Basis Radial (JFBR) dan Pohon Keputusan Induktif. JFBR digunakan sebagai metode pembelajaran dalam arsitektur jaringan syaraf tiruan. Untuk meningkatkan kemampuan pengenalan dilakukan pengklasifikasian pada Pohon Keputusan Induktif. Selain menjadi metode penghubung pada pengklasifikasian hibrid, Himpunan JFBR (HJFBR) digunakan untuk penyediaan atribut pada pengklasifikasian Pohon Keputusan Induktif. Uji coba dilakukan pada 50 obyek dengan total ± 500 citra wajah dalam format grayscale. Data dipilih dengan memberi variasi mimik wajah, kemiringan (rotasi) data $\pm 5^0$ dan juga dipengaruhi oleh pencahayaan di dalam ataupun di luar ruangan. Rata - rata peningkatan keakurasian positif benar yang diberikan arsitektur HJFBR dibanding JFBR sebesar $\pm 13,86\%$ untuk HJFBR1 dan $\pm 15,93\%$ untuk HJFBR2. Namun menurunkan keakurasian negatif benar sebesar $\pm 5,8\%$ untuk HJFBR1 dan $\pm 5,6\%$ untuk HJFBR2. Penambahan pohon keputusan induktif pada metode hibrid memberikan keuntungan selain tetap dapat meningkatkan keakurasian positif benar juga mampu mengatasi permasalahan sebelumnya tentang penurunan keakurasian negatif benar.

Kata kunci: pengenalan wajah, jaringan fungsi basis radial, pengklasifikasian hibrid, pohon keputusan induktif.

***ABSTRACT:** Face recognition is a difficult task mostly because of the inherent variability of the image formation process ranging from the position/cropping of the face and its environment (distance and illumination) is totally controlled, to those involving little or no control over the background and viewpoint. Moreover, those are allowing for major changes in facial appearance due to factors expression, aging, and accessories such as glasses or changes in hairstyle. A solution has been proposed by considering hybrid classification architectures deal with the benefit of robustness via consensus provided by ensembles of Radial Basis Functions (RBF) networks and categorical classification using decision trees. A specific approach considers an ensemble of RBF Networks through its ability to cope with variability in the image formation. The experiments were carried out on images drawn randomly 50 unique subjects totalling to 500 facial images with rotation ± 50 encoded in grayscale. The faces are then normalized to account for geometrical and illumination changes using information about the eye location. Specifically performance true positive by Ensamble RBF1 (ERBF1) increased on $\pm 13,86\%$ measures up to RBF while ERBF2 by $\pm 15,93\%$. On the contrary the false negative rate decreased by amount of $\pm 5,8\%$ for ERBF1 and somewhat less to $\pm 5,6\%$ for ERBF2. When the connectionist ERBF model is coupled with an Inductive Decision Tree - C4.5 - the performance improves over the case while only the connectionist ERBF module is used.*

Keywords: face recognition, radial basis function networks, hybrid classification, inductive decision tree.

PENDAHULUAN

Dewasa ini, teknologi informasi yang kian berkembang telah banyak menghasilkan berbagai aplikasi menggunakan citra wajah sebagai sumber informasi. Hal ini dikarenakan secara umum citra wajah dapat memberikan informasi khusus berkaitan dengan identifikasi personal namun sebagai input, citra wajah mempunyai variasi yang tinggi. Secara umum variasi tersebut dapat disebabkan oleh dua hal

yaitu variasi pada wajah itu sendiri dan variasi karena transformasi obyek wajah menjadi citra wajah. Penyebab pertama dapat berupa variasi pada detil - detil kecil dari wajah, misalkan ekspresi, pose atau adanya perubahan asesoris wajah. Sedangkan variasi yang disebabkan oleh proses pengambilan citra dapat berupa variasi sudut pengambilan gambar, latar belakang ataupun intensitas. Variasi dari citra wajah tersebut harus mampu diselesaikan oleh sistem pengenalan.

Pengklasifikasian wajah dengan JFBR yang digunakan sebagai metode pembelajaran dalam arsitektur jaringan syaraf tiruan telah mulai dilakukan oleh Srinivas Gutta dan H. Wechsler (1996). Kemudian Srinivas Gutta, J. Huang, dan H. Wechsler (1996) mengubah model JFBR menjadi HJFBR untuk meningkatkan kemampuan pengenalan wajah. Implementasi metode hibrid diterapkan Srinivas Gutta, Vasant Philomin dan Miroslav Tjrajcovic (2000) dengan menggunakan data pada HJFBR tidak berasal dari data asli yang ditambahkan *noise* namun memang dikarenakan data aslinya sudah memiliki *noise*.

Metode hibrid untuk pengenalan wajah juga dilakukan oleh Atle Nes (2003). Namun pengklasifikasian dengan metode hibrid tersebut memerlukan normalisasi data yang merupakan praproses data. Proses normalisasi dimulai dari proses pendeteksian wajah untuk mengurangi ukuran matrik yang akan diproses saat pembelajaran dan dilanjutkan transformasi data dengan rotasi dan *scalling* yang diharapkan akan dapat mengurangi variasi pada data pelatihan dan uji coba. Terakhir, ukuran data akan disamakan.

Metode LMS dengan metode *pseudoinverse* memiliki tujuan yang sama yaitu mencari nilai bobot dengan dipengaruhi parameter Gaussian dan parameter target keluaran untuk JFBR. Penerapan metode *pseudoinverse* yang berpengaruh pada faktor kecepatan digunakan Rully Soelaiman (2003) agar dapat memperoleh selisih kesalahan seminimum mungkin dengan nilai sebenarnya melalui penerapan *singular value decomposition*.

Penentuan bobot awal untuk JFBR tidak dilakukan secara acak namun mencari nilai yang mendekati data sebenarnya. Richard O. Duda, Peter E. Hart dan David G Stork (2001) menunjukkan metode yang sering digunakan adalah *k-means clustering*.

Supaya data berbentuk numerik supaya dapat dijadikan atribut pada pohon keputusan, maka oleh Tom Mitchell (1997) dilakukan diskretisasi sebelumnya. Tipe pohon keputusan yang akan dibuat telah dilakukan sebelumnya oleh Salvatore Ruggereri (2000) yaitu C4.5.

Metode penghubung pengklasifikasian berbasis JFBR dan Pohon Keputusan Induktif adalah Himpunan Jaringan Fungsi Basis Radial (HJFBR) untuk penyediaan atribut pada pengklasifikasian Pohon Keputusan Induktif dan pemilihan data negatif (tidak dikenali) — positif (dikenali) pada masing-masing kelas. Data pelatihan diolah dengan mekanisme validasi silang dengan pendekatan K-Fold bertujuan

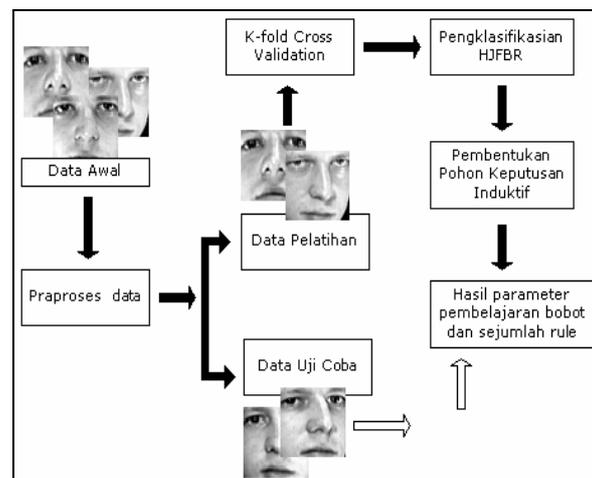
untuk mendapatkan prediksi nilai terbaik dalam pembentukan *rule* hasil Pohon Keputusan Induktif.

Perbedaan HJFBR dengan JFBR yaitu pada JFBR data diolah pada satu jaringan saja, sedangkan pada HJFBR data diolah dengan beberapa jaringan yang dipengaruhi oleh variasi data. Data pelatihan pada HJFBR diolah dengan mekanisme silang metode *K-fold* (dibatasi $k = 2$) sehingga terdapat data pembelajaran dan data validasi pada setiap *fold*. Seluruh node akan diolah secara terpisah dengan pembelajaran JFBR.

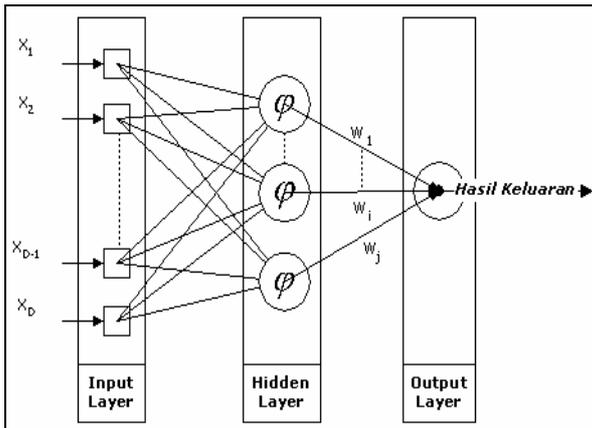
JARINGAN FUNGSI BASIS RADIAL

Pembelajaran JFBR menggabungkan pembelajaran *unsupervised* untuk mendapatkan nilai *center* dari proses *clustering* dengan metode *K-Means Clustering* dan selanjutnya menggunakan pembelajaran *supervised* untuk mendapatkan bobot sebagai parameter penyimpan hasil pembelajaran.

Pada *input layer* akan dihitung nilai *center* dengan metode *K-means clustering*. Inisialisasi *cluster center*, yaitu $\mu_1, \mu_2, \dots, \mu_k$ dengan nilai k telah ditentukan oleh sistem. Nilai inisialisasi dihitung dengan menggunakan *data range*, misalkan terdapat enam puluh data dan memiliki dua *cluster* maka indeks satu sampai indeks tiga puluh dari data akan dirata-rata dan digunakan menjadi *mean cluster* ke-1 dan indeks tigapuluh satu sampai indeks enam puluh akan dirata-rata yang digunakan sebagai *mean cluster* kedua. Nilai masukan, x_i , adalah dimensi dari setiap data dengan $i=1, 2, 3, \dots, D$ berupa sejumlah perkalian dari jumlah baris dan jumlah kolom gambar.



Gambar 1. Diagram Sistem Pengklasifikasian Hibrid



Gambar 2. Struktur Jaringan Fungsi Basis Radial (JFBR)

Algoritma *K-Means* menggunakan jarak *euclidean* kuadrat, $d(x, y)$, untuk menentukan kemiripan data satu dengan data yang lain.

$$d(x, y) = \sqrt{(x - y)^t (x - y)} = \left(\sum_{i=1}^D (x_i - y_i)^2 \right)^{\frac{1}{2}} \quad (1)$$

Pemindahan data dari *cluster* i ke *cluster* k jika jarak *euclidean* data terhadap *cluster* k nilainya lebih kecil dari nilai jarak *euclidean* data terhadap *cluster* i pada iterasi ke- m . Data akan dihitung untuk semua dimensi, $i=1, 2, 3, \dots, D$ berarti dimensi data.

Pembelajaran *K-Means* akan berhenti jika mencapai konvergen yang berdasarkan pada penghitungan nilai *sum-of-squared-error* yang dilambangkan dengan J_e .

$$J_e = \sum_{k=1}^K J_k \quad (2)$$

Nilai k adalah jumlah *cluster* yang digunakan atau jumlah unit *hidden* pada struktur JFBR dengan nilai maksimum sejumlah data pelatihan.

$$J_k = \sum_{x_j \in P_k} \|x_j - \mu_k\|^2 \quad (3)$$

Nilai μ_k adalah *mean(center)* dari subset P_k . Sedangkan P_k adalah subset ke- k dari S yaitu subset yang berisi data-data untuk *cluster* ke- k . Sehingga maksud dari $x_j \in P_k$ adalah x_j merupakan data vektor yang merupakan bagian dari *cluster* ke- k . Konvergen terjadi jika nilai J_e pada iterasi $m+1$ sama dibanding dengan iterasi m (iterasi sebelumnya). Iterasi maksimal diperlukan sebagai pembatasan penghitungan perubahan *center* setiap iterasi. Jika iterasi $m >$ daripada iterasi maksimal dan belum mencapai konvergen maka iterasi berhenti.

Fungsi aktivasi pada unit *hidden* disebut sebagai fungsi Gaussian (ϕ) atau fungsi radial yang dilakukan dengan komputasi non linier. Terdapat dua cara

penghitungan nilai Gaussian yaitu dipengaruhi nilai *overlap factor* (pengklasikasian HJFBR) dan tanpa dipengaruhi nilai *overlap factor* (pengklasikasian JFBR).

$$y_i = \phi(\|x - \mu_i\|) = \exp \left[\sum_{j=1}^D \frac{(x_j - \mu_{ij})^2}{2h\sigma_{ij}o} \right] \quad (4)$$

Pada proses ini akan menghitung titik tengah unit data (ϕ) dari seluruh dimensi secara non linier. Hasil penghitungan proses Gaussian akan digunakan untuk pencarian nilai bobot pada pembelajaran *supervised*.

Nilai $i=1, 2, 3, \dots, k$, dengan nilai k adalah jumlah *cluster* pada pembelajaran *K-means clustering* untuk mendapatkan nilai μ_i . Nilai j menunjukkan dimensi yang diacu oleh data. Sehingga μ_{ij} adalah *center* pada *cluster* ke- i dan dimensi ke- j . Nilai o adalah *overlap factor* sedangkan h adalah konstanta untuk varian (σ_{ij}). *Overlap factor* adalah nilai yang mempengaruhi hasil keluaran pada *output layer* agar bervariasi pada masing-masing jaringan untuk data yang sama.

Nilai varian (σ_{ij}) dibentuk berdasarkan sejumlah *cluster* yang terbentuk, nilai N_k adalah jumlah data pada *cluster* ke- i dengan $i=1, 2, 3, \dots, k$. Parameter x_{ih} adalah nilai data pada *cluster* ke- i dengan urutan data ke- h dari subset N_k .

$$\sigma_{ij}^2 = \frac{1}{N_k - 1} \sum_{h=1}^{N_k} (x_{ih} - \mu_{ij})^2 \quad (5)$$

Pembelajaran secara *supervised* pada *output layer* adalah sistem pembelajaran dengan menggunakan target keluaran (*desired output*) yang bernilai hanya 1 jika data tersebut telah masuk pada kelas yang diinginkan dan 0 pada kelas lain untuk mendapatkan bobot yang paling tepat agar mencapai keakurasian yang lebih tinggi. Selisih dari target keluaran adalah *kesalahan* yang digunakan untuk pengubah bobot jaringan sehingga kesalahan akan semakin kecil pada setiap kali iterasi telah diterapkan dengan pembelajaran *Least Mean Square* (LMS).

Pada *output layer* dilakukan komputasi linier untuk didapatkan hasil keluaran yang merupakan penjumlahan dari hasil perkalian bobot dengan nilai Gaussian (dari unit *hidden*).

$$f(x) = \sum_{i=1}^k w_i \phi_i + bias \quad (6)$$

Dengan $f(x)$ adalah nilai *actual output* dari JFBR, nilai w_i adalah bobot pada *cluster* ke- i dengan $i=1, 2, 3, \dots, k$ sedangkan k adalah jumlah *cluster* pada unit *hidden* dan nilai ϕ_i adalah nilai Gaussian pada *cluster* ke- i .

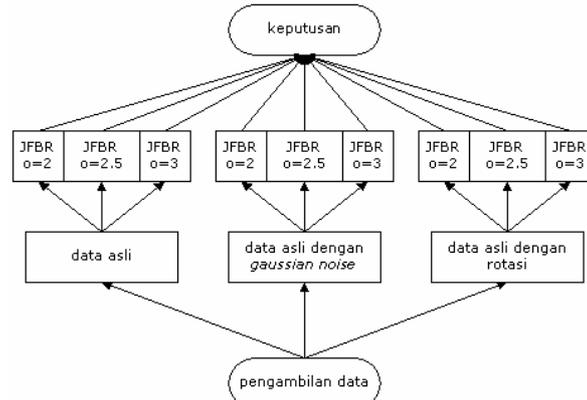
HIMPUNAN JARINGAN FUNGSI BASIS RADIAL

Model pertama pada HJFBR disebut HJFBR1 yang memiliki tiga komponen yaitu komponen data asal, komponen data asal dengan penambahan *gaussian noise* pada data dan komponen data asli dengan sedikit rotasi. Setiap komponen memiliki node sejumlah tiga node yang masing-masing node dipengaruhi oleh nilai *overlap factor* (o) dengan nilai 2, 2.5 dan 3 sehingga nantinya terdapat 9 variasi data. Dari seluruh hasil uji coba data validasi dari masing-masing jaringan (sejumlah 9) akan digabungkan sehingga menghasilkan pengklasifikasian HJFBR1 dengan *threshold* yang telah ditentukan. Jika rata-rata penjumlahan dari 9 jaringan \geq *threshold*, maka data tersebut masuk klasifikasi dikenali dan jika sebaliknya maka data tersebut tidak dikenali pada masing-masing kelas.

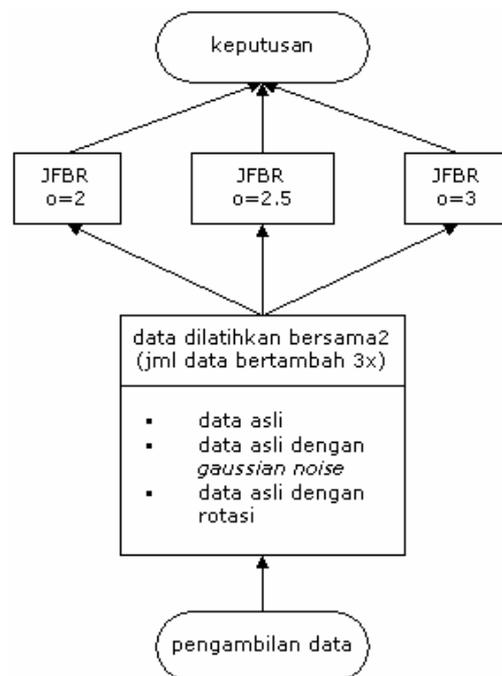
Model kedua pada yaitu HJFBR 2 juga memiliki tiga komponen yaitu komponen data asal, komponen data asal dengan penambahan *gaussian noise* pada data dan komponen data asli dengan sedikit rotasi. Perbedaan yang ada pada HJFBR 2 dibandingkan HJFBR 1 yaitu terjadi pengurangan node dari 9 node menjadi 3 node. Ketiga node pada HJFBR2 dipengaruhi oleh *overlap factor* (o) yaitu 2 untuk node 1, 2.5 untuk node 2 dan 3 untuk node 3. Setiap jaringan akan mengolah data asli, data dengan *gaussian noise* dan data dengan rotasi secara bersama-sama.

Pada HJFBR1 jumlah percobaan untuk mendapatkan bobot setiap jaringan dengan metode JFBR sebanyak 18 kali percobaan yaitu 9 percobaan pada *fold* 1 dan 9 percobaan pada *fold* 2 sehingga akhirnya dihasilkan 18 bobot. Iterasi pertama, data pelatihan JFBR dari 9 jaringan menggunakan data *fold* 1 dan data uji coba dari *fold* 2. Sebaliknya untuk iterasi kedua, data pelatihan menggunakan data *fold* 2 sedangkan uji coba dari *fold* 1. Nilai-nilai dari hasil uji coba data validasi dari setiap jaringan akan dipisahkan dan digabungkan berdasarkan kelasnya.

Proses pembelajaran dengan HJFBR1 dilakukan 2 kali. Untuk iterasi pertama ditunjukkan pada gambar 5 sedangkan untuk iterasi kedua perbedaannya hanya pada data pelatihan yang menggunakan *fold* 2 sedangkan pada data uji coba menggunakan *fold* 1. Untuk HJFBR2 sejumlah 6 kali percobaan untuk mendapatkan bobot yaitu 3 pada *fold* 1 dan 3 pada *fold* 2 dengan jumlah data tiga kali dari data pada HJFBR1 karena pada HJFBR2 data diolah bersamaan. Hasil dari ke-6 percobaan ini akan dihasilkan 3 bobot hasil pembelajaran.



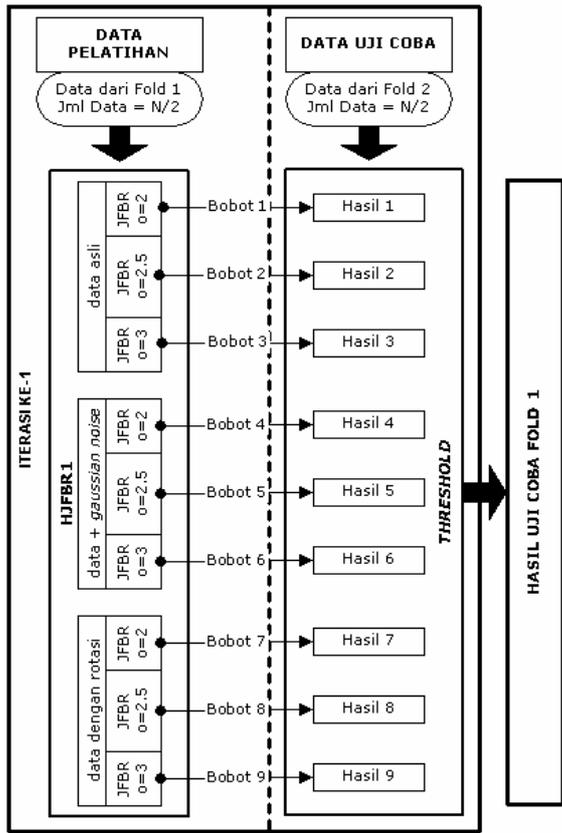
Gambar 3. Diagram Arsitektur HJFBR1



Gambar 4. Diagram Arsitektur HJFBR2

Pembelajaran pada HJFBR2 juga dilakukan 2 kali yang ditunjukkan pada gambar 6. Setelah proses pembelajaran JFBR pada masing-masing jaringan, nilai dari hasil uji coba data validasi dari setiap jaringan akan dipisahkan dan digabungkan berdasarkan kelasnya sesuai gambar 7.

Hasil penggabungan ini akan digunakan untuk pengklasifikasian HJFBR yang dipengaruhi oleh nilai *threshold*. Setelah proses penggabungan data sesuai gambar 7 dilanjutkan pengklasifikasian dengan HJFBR yang akan menghasilkan data positif dan data negatif dari masing-masing kelas dipengaruhi *threshold*. Hasil uji coba pengklasifikasian HJFBR dari masing-masing *fold* digabungkan menjadi satu kemudian dilanjutkan proses pemisahan data yang menghasilkan data pelatihan dan data uji coba (validasi) untuk pengklasifikasian Pohon Keputusan Induktif.



Gambar 5. Skema Pengklasifikasian HJFBR1 pada Iterasi ke-1

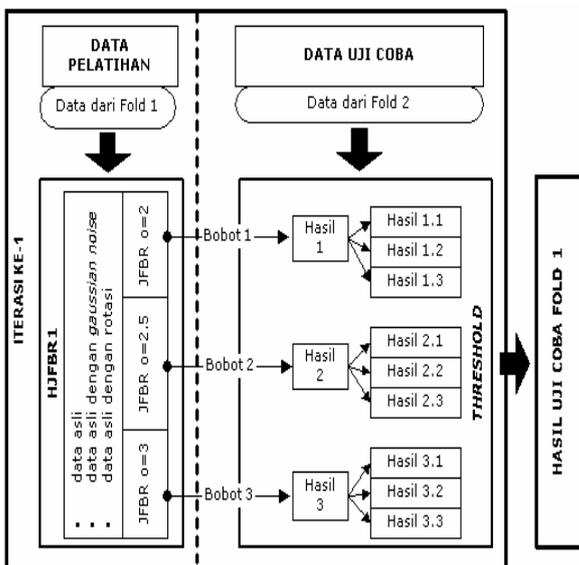
Proses persiapan data untuk pohon keputusan induktif adalah sebagai berikut. Data positif hasil uji coba HJFBR1 dari *fold* 1 dan hasil uji coba HJFBR1 dari *fold* 2 akan digabungkan sebagai hasil positif. Hasil penggabungan ini akan dirandom sehingga menghasilkan data-data pelatihan dan data-data uji coba dengan nilai positif. Begitu juga dengan data nilai negatif. Misal dari gambar 9, data hasil positif benar pada pelatihan HJBR1 dengan HJBR2 akan digabung menjadi data positif benar yang kemudian diacak.

Hal yang sama berlaku untuk persiapan data untuk pohon keputusan induktif dari data yang sebelumnya diolah dengan HJFBR2.

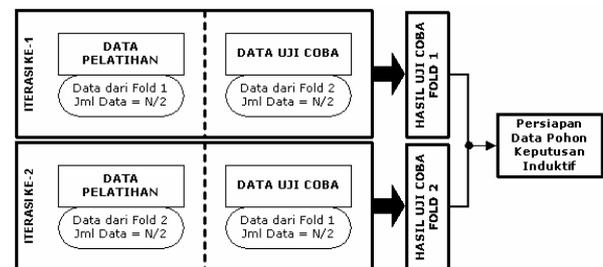
JFBR1				JFBR2				JFBR3			
id	Kelas			id	Kelas			id	kelas		
	1	2	3		1	2	3		1	2	3
1	0.53	0.06	0.04	1	0.55	0.03	0.02	1	0.73	0.01	0.03
2	0.62	0.01	0.02	2	0.72	0.01	0.02	2	0.65	0.02	0.01
3	-0.22	0.67	0.02	3	-0.21	0.87	0.04	3	0.25	0.71	0.01

a. Atribut Kelas 1				b. Atribut Kelas 2				c. Atribut Kelas 3			
id	Atribut			id	Atribut			id	Atribut		
	1	2	3		1	2	3		1	2	3
1	0.53	0.55	0.73	1	0.06	0.03	0.01	1	0.04	0.02	0.03
2	0.62	0.72	0.65	2	0.01	0.01	0.02	2	0.02	0.02	0.01
3	-0.22	-0.21	0.25	3	0.67	0.87	0.71	3	0.02	0.04	0.01

Gambar 7. Skema Contoh Penggabungan Data Berdasarkan Kelas



Gambar 6. Skema Pengklasifikasian HJFBR2 pada Iterasi ke-1



Gambar 8. Mekanisme Pembentukan Data untuk Pohon Keputusan Induktif

METODE HIBRID HJFBR DAN POHON KEPUTUSAN INDUKTIF

Pembelajaran secara induktif adalah pembelajaran berdasarkan data-data yang telah dipelajari, dalam hal ini adalah data-data dari data pelatihan untuk mendapatkan sejumlah *rule* guna memprediksi klasifikasi. Untuk tujuan klasifikasi maka kelas pada masing-masing data perlu diketahui sebelumnya.

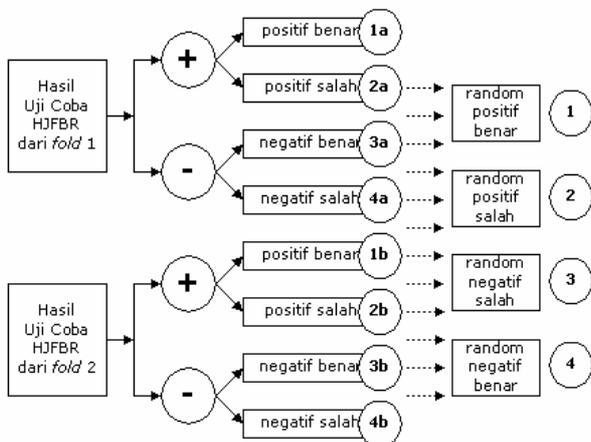
Atribut yang akan digunakan pada pohon keputusan induktif bernilai numerik disebabkan hasil dari pengklasifikasian HJFBR berupa nilai numerik. Untuk nilai numerik maka perlu dilakukan proses diskretisasi (*dicretization*) melalui dua tahap, yaitu:

Setelah data-data pada seluruh fold telah diklasifikasikan dengan metode HJFBR, maka pada gambar 8 data hasil uji coba dari setiap fold akan digabungkan dan dipilih sebagian data untuk pelatihan Pohon Keputusan Induktif.

- Mengurutkan data
Setiap atribut (data setiap kelas) dengan nilai numerik perlu diurutkan dari nilai yang terbesar menjadi terkecil sesuai dengan karakteristik hasil JFBR.
- Pemilihan *threshold*
Setiap atribut memiliki beberapa pilihan *threshold* dimana penentuannya setiap kali terjadi perubahan target keluaran (*desired output*) pada suatu kelas dari nilai 1 menjadi 0 atau sebaliknya maka akan terbentuk *threshold*. Nilai terpilih adalah nilai yang akan menghasilkan *Gain Ratio Criterion* terbesar.

Langkah-langkah pembentukan pohon keputusan induktif C4.5 dengan menghitung frekuensi terhadap kelas pada masing-masing atribut, jika *node* telah berstatus *stopping criteria* maka *node* diubah menjadi *leaf*; jika tidak, maka menghitung *Gain Ratio Criterion* yang terbaik untuk memilih atribut dan proses pembentukan pohon keputusan induktif berulang kembali.

Setelah Pohon Keputusan Induktif telah terbentuk, maka dilakukan uji coba data yang belum pernah dibelajarkan pada sistem menggunakan parameter bobot dari salah satu *fold* dengan keakurasian terbaik dan sejumlah *rule* hasil pengklasifikasian Pohon Keputusan Induktif.



Gambar 9. Data Pelatihan dan Uji Coba pada Pohon Keputusan Induktif

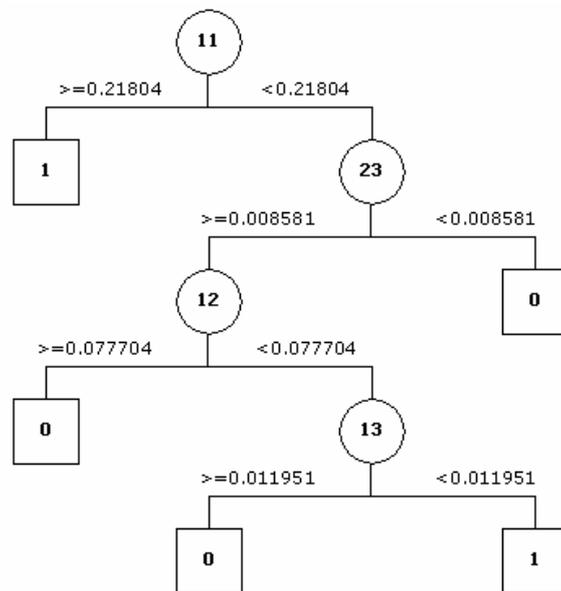
UJI COBA DAN EVALUASI

Data yang digunakan pada uji coba yaitu *database* Ollivetti yang terdiri dari 40 orang dengan 10 variasi perorang dan *database* Archive yang terdiri dari 14 orang dan 8 variasi perorang. Pada *database* Ollivetti terdapat 2 percobaan: percobaan 1 adalah data-data dari kelas 1 sampai kelas 31 dengan data terpilih dan percobaan 2 diambil dari kelas 21 sampai

kelas 40 dengan data terpilih. Keakurasian pada hasil uji coba dipengaruhi oleh 4 kriteria sesuai tabel 2. Pada setiap percobaan, data terbagi menjadi 3 bagian yaitu data pelatihan, data validasi dan data uji coba. Variasi mimik wajah pada *database* Archive tidak variatif dibandingkan Ollivetti dan hanya sedikit kelas yang digunakan sehingga cukup satu kali percobaan pada *database* Archive.

Tabel 1. Data Terurut untuk Pemilihan *Threshold*

Id	Atribut	Kelas
4	0.72	1
3	0.67	1
2	0.66	1
1	0.58	1
8	-0.07	0
14	-0.07	0
13	-0.08	0
7	-0.08	0
10	-0.09	0
11	-0.09	0
9	-0.10	0
12	-0.10	0
5	-0.16	0
6	-0.16	0



Gambar 10. Contoh Hasil Pembentukan Pohon Keputusan Induktif untuk Kelas 3

Pada percobaan 1 dan percobaan 2, nilai *threshold* diberikan dari 0.005 hingga 0.29. Data dengan pengaruh *Gaussian noise* sebelumnya dirubah dahulu secara manual sehingga setiap *pixel*-

nya mendapat *noise* dengan radius sebesar 0.0005. Sedangkan data dengan pengaruh rotasi akan mendapat *blur image*. Pengklasifikasian berdasarkan *threshold* pada HJFBR 1 bertujuan meningkatkan keakurasian pengenalan. Nilai *threshold* yang tepat diambil untuk mendapatkan kegagalan terkecil.

Tabel 2. Kriteria Keakurasian

KRITERIA		KETERANGAN
P O S I T I F	BENAR	data terklasifikasikan pada kelas dengan benar
	SALAH	data terklasifikasikan pada kelas dengan salah
N E G A T I F	BENAR	data tidak terklasifikasikan pada kelas dan memang seharusnya tidak terklasifikasikan
	SALAH	data tidak terklasifikasikan pada kelas tetapi seharusnya terklasifikasikan

Untuk percobaan 1 Olivetti diambil 5 data dari setiap kelas yang dibagi menjadi dua *fold* yaitu *fold* 1 dengan data pelatihan diambil 3 dan data validasi diambil 2 data. Sedangkan untuk *fold* 2 diambil 2 data untuk data pelatihan dan 3 sisanya untuk data validasi. Untuk percobaan 2 Olivetti, jumlah data pelatihan secara keseluruhan sama dengan percobaan 1 yaitu 5 data perkelas yang terbagi lagi menjadi dua *fold*. Pada *fold* 1, sama dengan percobaan 1 jumlah data perkelasnya yaitu 3 data dan pada *fold* 2 jumlah data perkelasnya yaitu 2 data.

Untuk percobaan Archive, jumlah data pelatihan yaitu 2 data pada setiap kelas, yang dibagi menjadi dua *fold* yaitu *fold* 1 dengan jumlah data pelatihan 1 data dan jumlah data validasi 1 data pada masing-masing kelas. *Fold* 2 dengan jumlah data pelatihan 1 data dan jumlah data validasi 1 data pada masing-masing kelas.

Adanya pengaruh pembatasan dengan *threshold* terlihat pada tabel 3. Baris [a] dilakukan tanpa pemberian *threshold*, sedangkan baris [b] adalah hasil pengujian dengan nilai *threshold*. Pada percobaan 1 *threshold* = 0,13 untuk HJFBR1 dan 0,17 untuk HJFBR2. Pengaruh *threshold* pada percobaan 1 dapat dilihat pada tabel 3(a). Pada percobaan 2, dipilih nilai *threshold* = 0,16 untuk HJFBR1 dan 0,22 untuk HJFBR2. Pengaruh *threshold* pada percobaan 2 dapat dilihat pada tabel 3(b). Sedangkan untuk percobaan Archive dapat dilihat pada tabel 3(c).

Tabel 3. Hasil Pengujian Data Validasi dengan Pengaruh *Threshold*

(a). Percobaan 1 Olivetti

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
JFBR	37,10%	62,90%	97,90%	2,09%
HJFBR1 [a]	59,31%	40,68%	98,64%	1,35%
HJFBR1 [b]	74,73%	25,27%	93,46%	6,54%
HJFBR2 [a]	75,65%	24,34%	99,18%	0,81%
HJFBR2 [b]	93,28%	6,72%	93,21%	6,79%

(b). Percobaan 2 Olivetti

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
JFBR	40,00%	60,00%	96,84%	3,16%
HJFBR1 [a]	61,45%	39,18%	97,97%	2,03%
HJFBR1 [b]	73,75%	26,25%	91,55%	8,44%
HJFBR2 [a]	78,28%	21,71%	98,85%	1,14%
HJFBR2 [b]	92,50%	7,50%	92,83%	7,17%

(c). Percobaan Archive

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
JFBR	100,00%	0,00%	100,00%	0,00%
HJFBR1 [a]	100,00%	0,00%	100,00%	0,00%
HJFBR1 [b]	100,00%	0,00%	93,95%	6,04%
HJFBR2 [a]	100,00%	0,00%	100,00%	0,00%
HJFBR2 [b]	100,00%	0,00%	100,00%	0,00%

Pada percobaan 1, pembatasan dengan nilai *threshold* meningkatkan keakurasian positif benar dengan $\pm 15,42\%$ untuk HJFBR1 dan $\pm 17,63\%$ untuk HJFBR2. Sedangkan percobaan 2, meningkatkan sebesar $\pm 12,30\%$ untuk HJFBR1 dan $\pm 14,22\%$ untuk HJFBR2. Namun kelemahannya, *threshold* menurunkan keakurasian negatif benar sebesar $\pm 5,18\%$ pada HJFBR1 dan $\pm 5,97\%$ pada HJFBR2 untuk percobaan 1. Dan pada percobaan 2 terjadi kenaikan $\pm 6,42\%$ untuk HJFBR1 serta $\pm 6,02\%$ untuk HJFBR2. Percobaan Archive, pengaruh *threshold* tidak terlihat nyata dikarenakan keterbatasan variasi data pembelajaran.

Setelah melalui proses mekanisme pembelajaran JFBR + HJFBR1 + pohon keputusan induktif (metode hibrid 1) maupun JFBR + HJFBR2 + pohon keputusan induktif (metode hibrid 2), maka dilakukan uji coba yang menghasilkan tabel 4. Pengujian metode hibrid menggunakan data validasi ditunjukkan pada baris Hibrid 1 [a] dan Hibrid 2 [a]. Sedangkan untuk data uji coba ditunjukkan pada baris Hibrid 1 [b] dan Hibrid 2 [b].

Tabel 4. Hasil Pengujian dengan Metode Hibrid

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
Hibrid 1 [a]	88,70%	11,29%	80,99%	19,00%
Hibrid 1 [b]	89,03%	10,96%	86,43%	13,56%
Hibrid 2 [a]	96,77%	3,22%	91,02%	8,97%
Hibrid 2 [b]	91,61%	8,38%	93,95%	6,04%

(a). Percobaan 1 Olivetti

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
Hibrid 1 [a]	85,00%	15,00%	87,76%	12,23%
Hibrid 1 [b]	94,00%	6,00%	84,78%	15,21%
Hibrid 2 [a]	97,50%	2,50%	90,62%	9,37%
Hibrid 2 [b]	95,00%	5,00%	93,94%	6,05%

(b). Percobaan 2 Olivetti

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
Hibrid 1 [a]	100,00%	0,00%	100,00%	0,00%
Hibrid 1 [b]	100,00%	0,00%	99,26%	0,73%
Hibrid 2 [a]	100,00%	0,00%	99,45%	0,54%
Hibrid 2 [b]	100,00%	0,00%	99,26%	0,73%

(c). Percobaan Archive

Pada tabel 4, hasil yang ditunjukkan tidak terlalu jauh antara metode Hibrid baris [a] dengan baris [b] karena pembobotan pada arsitektur dan *rule* yang telah dihasilkan sama. Perbedaannya hanya terletak pada serangkaian data yang diujicobakan.

Dari tabel 5 dapat diambil kesimpulan pengenalan wajah dengan JFBR yang telah dimodifikasi sehingga mampu menangani variasi data (HJFBR) serta pemberian batasan *threshold* dapat meningkatkan keakurasian positif benar. Antara HJFBR1 dengan HJFBR2, penggunaan arsitektur HJFBR2 memberikan hasil yang lebih baik. Namun memiliki kelemahan dengan meningkatkan nilai hasil negatif benar. Dikombinasikan pohon keputusan induktif setelah pembelajaran dengan HJFBR yang disebut metode Hibrid dapat mengurangi kegagalan pada pengklasifikasian dengan keakurasian negatif benar juga meningkat.

Namun untuk percobaan Archive tidak terlalu mencolok perbedaannya. Dikarenakan data pada *database* Archive berasal dari data RGB yang dinormalisasi. Terlihat keakurasian tetap 100% pada setiap metode dan terjadi sedikit penurunan pada Hibrid 1 dan terjadi perbaikan pada Hibrid 2.

Tabel 5. Evaluasi Percobaan dengan Data Validasi

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
JFBR	37,10%	62,90%	97,90%	2,09%
HJFBR1	74,73%	25,27%	93,46%	6,54%
HJFBR2	93,28%	6,72%	93,21%	6,79%
Hibrid 1	88,70%	11,29%	80,99%	19,00%
Hibrid 2	96,77%	3,22%	91,02%	8,97%

(a). Percobaan 1 Olivetti

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
JFBR	40,00%	60,00%	96,84%	3,16%
HJFBR1	73,75%	26,25%	91,55%	8,44%
HJFBR2	92,50%	7,50%	92,83%	7,17%
Hibrid 1	85,00%	15,00%	87,76%	12,23%
Hibrid 2	97,50%	2,50%	90,62%	9,37%

(b). Percobaan 2 Olivetti

Metode	Positif		Negatif	
	Benar	Salah	Benar	Salah
JFBR	100,00%	0,00%	100,00%	0,00%
HJFBR1	100,00%	0,00%	93,95%	6,04%
HJFBR2	100,00%	0,00%	100,00%	0,00%
Hibrid 1	100,00%	0,00%	100,00%	0,00%
Hibrid 2	100,00%	0,00%	99,45%	0,54%

(c). Percobaan Archive

KESIMPULAN

1. Metode HJFBR1 dan HJFBR2 mampu meningkatkan keakurasian benar positif jika dibandingkan metode JFBR sebesar 12.26% pada HJFBR1 dan 14.84% pada HJFBR2 untuk percobaan 1 Olivetti. Sedangkan untuk percobaan 2 terjadi peningkatan keakurasian sebesar 3% pada HJFBR1 dan 4 % pada HJFBR2. Untuk percobaan Archive keduanya terjadi peningkatan 1.59%.
2. HJFBR2 memiliki kegagalan yang lebih sedikit jika dibandingkan HJFBR1. Pada percobaan 1 terjadi pengurangan kegagalan sebesar 2.58 % pada positif benar dan 7.52% pada negatif benar. Pada percobaan 2 terjadi pengurangan kegagalan sebesar 1% pada positif benar dan 9.16% pada negatif benar.
3. Jika dilakukan pencarian citra wajah untuk 1 orang, metode pelatihan JFBR memberikan hasil yang baik dibanding HJFBR1 dan HJFBR2. Namun jika pencarian citra >1 HJFBR1 dan HJFBR2 menunjukkan kelebihan terutama metode HJFBR2.

DAFTAR PUSTAKA

1. Duda, Richard O., Hart, Peter E. dan Stork, David G., *Pattern Classification*, edisi ke-2, John Wiley & Sons Inc, New York. 2001.
2. Gutta, Srinivas, J. Huang, H. Wechsler, *Face Recognition Using Ensembles of Networks*, Intl. Conf. On Pattern Recognition (ICPR), Vienna, Austria. 1996.
3. Gutta, Srinivas, H. Wechsler, *Face Recognition Using Hybrid Classifier Systems*, Department of Computer Science, George Masson University, Fairfax, VA. 1996.
4. Gutta, Srinivas, Vasant Philomin, Miroslav Tjrajcovic, *An Investigation into the use of Partial-Faces for Face Recognition*, Philips Research – USA. 2000.
5. Mitchell, Tom M., *Machine Learning*, The McGraw Hill Companies, Inc., New York. 1997.
6. Nes, Atle, *Hybrid Systems for Face Recognition*, Norwegian University of Science and Technology Faculty of Information Technology, Mathematics and Electrical Engineering Department of Computer and Information Science Division of Intelligent Systems / Image Processing, 16 June 2003.
7. Ruggereri, Salvatore, *Efficient C4.5*, Department Informatics, Pisa University, Pisa, Italy. 2000.
8. Soelaiman, Rully, *Sistem Pengenalan wajah dengan penerapan Algoritma Genetika pada Optimasi Basis Eigenface dan Proyeksi Fisherface*, Tesis: Program Pascasarjana Universitas Indonesia. 2003.
9. Zheng, Zijian, *Constructing New Attributes for Decision Tree Learning*, Basser Department Computer of Science, The University of Sydney. 1996.