

# PELACAKAN DAN PENGENALAN WAJAH MENGGUNAKAN METODE EMBEDDED HIDDEN MARKOV MODELS

**Arie Wirawan Margono, Ibnu Gunawan, Resmana Lim**

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

e-mail : [ariewm@petra.ac.id](mailto:ariewm@petra.ac.id) , [ibnu@petra.ac.id](mailto:ibnu@petra.ac.id) , [resmana@petra.ac.id](mailto:resmana@petra.ac.id)

**ABSTRAK:** Pelacakan dan pengenalan wajah manusia merupakan salah satu bidang yang cukup berkembang dewasa ini, dimana aplikasi dapat diterapkan dalam bidang keamanan (*security system*) seperti ijin akses masuk ruangan, pengawasan lokasi (*surveillance*), maupun pencarian identitas individu pada *database* kepolisian. Karena diterapkan dalam kasus keamanan, dibutuhkan sistem yang handal terhadap beberapa kondisi, seperti: pengaruh latar belakang, pose wajah *non-frontal* terhadap pria maupun wanita dalam perbedaan usia dan ras.

Tujuan penelitian ini adalah untuk membuat perangkat lunak yang menggabungkan sistem pelacakan wajah manusia dengan menggunakan algoritma *CamShift* dan sistem pengenalan wajah dengan menggunakan algoritma *Embedded Hidden Markov Models*. Sebagai input sistem digunakan video kamera (*webcam*) untuk input bersifat *real-time*, video AVI untuk input bersifat dinamis, dan file *image* untuk input statis. Pemrograman perangkat lunak menggunakan prinsip pemrograman berorientasi objek (OOP) dengan menggunakan bahasa pemrograman C++, kompiler Microsoft Visual C++ 6.0<sup>®</sup>, dan dibantu dengan *library* dari *Intel Image Processing Library* (IPL) dan *Intel Open Source Computer Vision* (OpenCV).

Hasil pengujian sistem menunjukkan bahwa pelacakan berdasarkan warna kulit manusia dengan menggunakan algoritma *CamShift* cukup baik, dalam melakukan pelacakan terhadap satu maupun dua objek wajah sekaligus. Sistem pengenalan wajah manusia menggunakan metode *Embedded Hidden Markov Models* mencapai tingkat akurasi pengenalan sebesar 82.76%, dengan *database* citra wajah sebanyak 341 citra yang terdiri dari 31 individu dengan 11 pose, dan jumlah citra pengujian sebanyak 29 citra wajah.

**Kata kunci:** *Computer Vision*, Pelacakan Objek, *CamShift*, Pengenalan Wajah, *Hidden Markov Model*.

**ABSTRACT:** *Tracking and recognizing human face becomes one of the important research subjects nowadays, where it is applicable in security system like room access, surveillance, as well as searching for person identity in police database. Because of applying in security case, it is necessary to have robust system for certain conditions such as: background influence, non-frontal face pose of male or female in different age and race.*

*The aim of this research is to develop software which combines human face tracking using CamShift algorithm and face recognition system using Embedded Hidden Markov Models. The software uses video camera (webcam) for real-time input, video AVI for dynamic input, and image file for static input. The software uses Object Oriented Programming (OOP) coding style with C++ programming language, Microsoft Visual C++ 6.0<sup>®</sup> compiler, and assisted by some libraries of Intel Image Processing Library (IPL) and Intel Open Source Computer Vision (OpenCV).*

*System testing shows that object tracking based on skin complexion using CamShift algorithm comes out well, for tracking of single or even two face objects at once. Human face recognition system using Embedded Hidden Markov Models method has reach accuracy percentage of 82.76%, using 341 human faces in database that consists of 31 individuals with 11 poses and 29 human face testers.*

**Keywords:** *Computer Vision*, Object Tracking, *CamShift*, Face Recognition, *Hidden Markov Model*.

## 1. PENDAHULUAN

Pada penelitian ini akan dibuat suatu sistem yang dapat digunakan secara sekaligus untuk melacak posisi wajah seseorang

(*tracking*) dan mengenali wajah tersebut (*recognition*) berdasarkan *database* wajah yang telah dibuat dan di-*training* sebelumnya.

Algoritma yang digunakan untuk melakukan *tracking* posisi wajah adalah algoritma *CamShift* yang menggunakan faktor warna sebagai dasar pelacakannya.

Setelah objek wajah dapat di-*track* dan dideteksi dengan jelas, baru kemudian dilakukan proses pengenalan wajah dengan menggunakan algoritma *Embedded Hidden Markov Models*.

Input data wajah diperoleh dari:

- Kamera (*webcam*), untuk input bersifat *real-time*.
- *File* video (dengan format: AVI), untuk input bersifat dinamis.
- *File* citra (dengan format: BMP, JPEG, PNG, TIFF), untuk input bersifat statis.

Output dari sistem berupa wajah dalam *database* yang telah dikenali dengan nilai kemiripan (*likelihood*) yang paling tinggi.

### 1.1 Algoritma CAMSHIFT

*CamShift* adalah singkatan dari *Continuously Adaptive Mean Shift*, yang merupakan pengembangan dari algoritma *Mean Shift* yang dilakukan secara terus menerus (berulang) untuk melakukan adaptasi atau penyesuaian terhadap distribusi probabilitas warna yang selalu berubah tiap pergantian *frame* dari *video sequence*.

Langkah-langkah dari algoritma *Mean Shift* adalah sebagai berikut:

- a. Ukuran *search window* yang sudah ditentukan.
- b. Lokasi awal *search window* yang sudah ditentukan.
- c. Hitung daerah *mean* dalam *search window*.
- d. Posisikan *search window* ke tengah daerah *mean* seperti dihitung pada *step c*.
- e. Ulangi *step c* dan d hingga konvergen (atau hingga pergeseran daerah *mean* kurang dari *threshold*/batas yang ditentukan).

Sedangkan langkah-langkah dari algoritma *CamShift* adalah sebagai berikut:

- a. Tentukan ukuran awal *search window*.
- b. Tentukan lokasi awal dari *search window*.

- c. Tentukan daerah kalkulasi (*calculation region*) pada bagian tengah *search window* dengan ukuran lebih besar dari *search window*.
- d. *Frame* citra video dikonversi ke dalam sistem warna HSV (*Hue, Saturation, Value*), dan dilakukan *color histogram lookup* dalam *calculation region* yang akan menghasilkan citra distribusi probabilitas warna kulit.
- e. Lakukan algoritma *Mean Shift* seperti di atas (satu atau banyak iterasi) dengan input berupa ukuran dan lokasi *search window* serta citra distribusi probabilitas warna, simpan *zeroth moment*.
- f. Set nilai *x, y, z*, dan *head roll* yang diperoleh dari *step e*.
- g. Nilai *x, y* dipakai untuk set titik tengah *search window*, ( $2 * area^{1/2}$ ) untuk set ukuran *search window*.
- h. Ulangi *step c* untuk setiap pergantian *frame* citra video.

Untuk citra distribusi probabilitas warna kulit, daerah *mean* (*centroid*) di dalam *search window* bisa dicari dengan persamaan:

Cari *zeroth moment*:

$$M_{00} = \sum_x \sum_y I(x,y) \tag{1}$$

Cari *first moment* untuk *x* dan *y*:

$$M_{10} = \sum_x \sum_y xI(x,y) \tag{2}$$

$$M_{01} = \sum_x \sum_y yI(x,y)$$

Maka lokasi *mean* dalam *search window* (*centroid*) adalah:

$$x_c = \frac{M_{10}}{M_{00}} \quad y_c = \frac{M_{01}}{M_{00}} \tag{3}$$

dimana  $I(x,y)$  adalah nilai warna piksel di posisi (*x, y*) pada citra dan *x, y* berada di dalam *search window*.

Orientasi 2 dimensi (2D) dari objek wajah diperoleh dengan melakukan perhitungan *second moments* dengan persamaan:

$$M_{20} = \sum_x \sum_y x^2 I(x,y) \tag{4}$$

$$M_{02} = \sum_x \sum_y y^2 I(x,y)$$

dimana orientasi objek adalah:

$$q = \frac{\arctan \left( \frac{2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right)}{\left( \frac{M_{20}}{M_{00}} - x_c^2 \right) - \left( \frac{M_{02}}{M_{00}} - y_c^2 \right)} \right)}{2} \quad (5)$$

jika

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \quad b = 2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right) \quad (6)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2$$

maka panjang l dan lebar w dari distribusi centroid yaitu:

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (7)$$

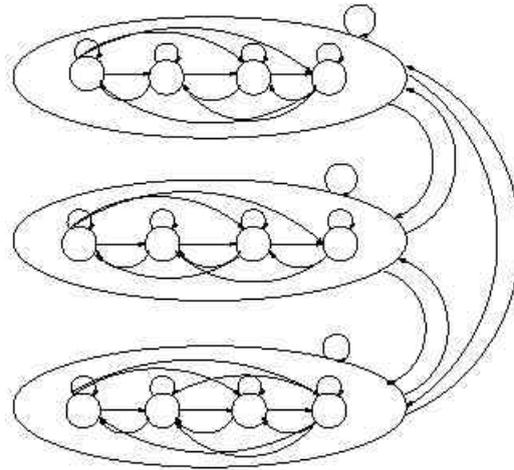
$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}$$

Pemakaian persamaan tersebut pada sistem pelacakan wajah akan menghasilkan nilai x, y, rotasi wajah, panjang dan lebar (area atau nilai z).

### 1.2 Algoritma Embedded Hidden Markov Models

Sebuah *embedded* HMM (seperti ditunjukkan pada Gambar 1) adalah pengembangan dari HMM (*Hidden Markov Model*), dimana suatu *state* pada 1-D HMM adalah juga sebuah HMM.

*Embedded* HMM terdiri dari himpunan *super state* dimana masing-masing berisi himpunan *state* pula yang disebut dengan *embedded state*. *Super state* memodelkan data citra 2 dimensi pada arah vertikal, sedangkan *embedded state* memodelkan pada arah *horizontal*



**Gambar 1. Embedded HMM Dengan 3 Super State**

Berikut adalah elemen dari *embedded* HMM:

- a. Himpunan *super state* sebanyak  $N_0$ ,  $S_0 = \{S_{0,i}\} \quad 1 \leq i \leq N_0.$  (8)
- b. *Initial super state probability distribution*,  $\mathbf{p}_0 = \{\mathbf{p}_{0,i}\}$ , dimana  $\mathbf{p}_{0,i}$  adalah nilai probabilitas berada pada *super state*  $i$  pada waktu  $t=0$ .
- c. Himpunan transisi *state* antar *super state* yang berupa matriks dua dimensi,  $A_0 = \{a_{0,ij}\}$ , dimana  $a_{0,ij}$  adalah nilai probabilitas untuk melakukan transisi dari *super state*  $i$  ke *super state*  $j$ .
- d. Pada metode ini, tiap *super state* adalah juga merupakan HMM (disebut *embedded* HMM), dan struktur dari *embedded* HMM ini sama dengan 1-D HMM. Namun tidak seperti 1-D HMM, jumlah *state*, *initial state probability*, dan *state transition* matriks, sangat bergantung pada *super state* dimana *embedded* HMM tersebut berada. Karena itu masing-masing *embedded* HMM juga memiliki elemen sebagai berikut:
  - i. Jumlah *embedded state* pada *super state* yang ke  $k$ ,  $N_1^k$ , dan himpunan *embedded state*,  $S_1^k = \{S_{1,i}^k, i=1,2,\dots, N_1^k\}$  (9)

- ii. *Initial state probability distribution* pada *embedded state*,  $\mathbf{p}_1^k = \{p_{1,i}^k\}$ , dimana  $p_{1,i}^k$  adalah nilai probabilitas berada pada *state*  $i$  dari *super state*  $k$  pada waktu  $t = 0$ .
- iii. *State Transition* Matriks untuk *embedded state*,  $A_1^k = \{a_{1,ij}^k\}$ , dimana  $a_{1,ij}^k$  adalah nilai probabilitas untuk melakukan transisi dari *state*  $i$  ke *state*  $j$  pada *super state*  $k$ .
- iv. *Probability Distribution* Matriks,  $B^k = \{b_j^k(\mathbf{o}_{t_0,t_1})\}$ , dimana  $b_j^k(\mathbf{o}_{t_0,t_1})$  adalah nilai probabilitas dari observasi  $\mathbf{o}_{t_0,t_1}$  yang dihasilkan pada *embedded state*  $j$  dan *super state*  $k$ . Untuk *observation vector*  $\mathbf{o}_{t_0,t_1}$  terdapat dua *subscript*, yaitu  $t_0$  dan  $t_1$ , dimana  $t_0$  menyatakan waktu dari *super state* dan  $t_1$  menyatakan waktu dari *embedded state*, bisa juga dikatakan pada baris  $t_0$  dan kolom  $t_1$ .

Untuk *continuous density embedded HMM*, observasi diperoleh dari *continuous probability density function* (seperti pada 1-D HMM), yang berupa *finite Gaussian mixtures*, dengan bentuk:

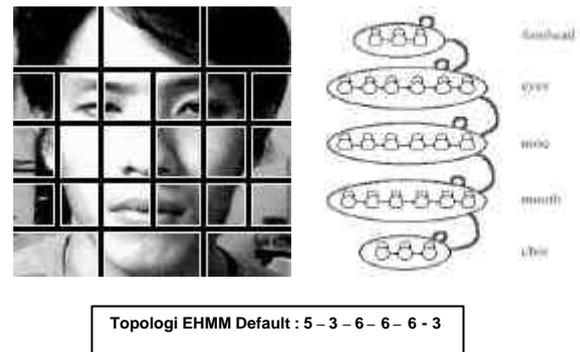
$$b_i^k(\mathbf{o}_{t_0,t_1}) = \sum_{m=1}^{M_i^k} c_{im}^k N(\mathbf{o}_{t_0,t_1}, \mathbf{m}_{im}^k, U_{im}^k) \quad (10)$$

untuk  $1 \leq i \leq N_1^k$ , dimana  $c_{im}^k$  adalah *mixture coefficient* untuk *mixture* ke  $m$  pada *state*  $i$  *super state*  $k$ , dan  $N(\mathbf{o}_{t_0,t_1}, \mathbf{m}_{im}^k, U_{im}^k)$  adalah *Gaussian density* dengan *mean* vektor  $\mathbf{m}_{im}^k$  dan kovarian matriks  $U_{im}^k$ .

Gambar 2 berikut menunjukkan pembagian citra wajah sesuai dengan topologi HMM (5-3-6-6-6-3) dimana wajah dibagi menjadi 5 bagian (*Super State*), yaitu:

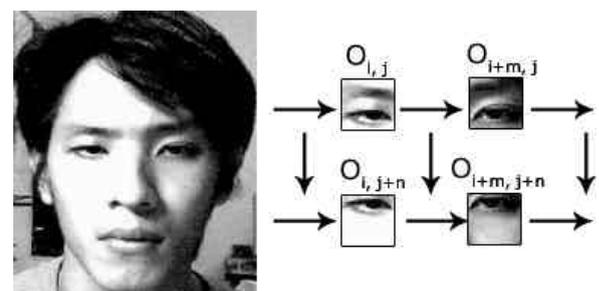
- Dahi, dibagi lagi menjadi 3.
- Mata, dibagi lagi menjadi 6.
- Hidung, dibagi lagi menjadi 6.
- Mulut, dibagi lagi menjadi 6.
- Daggu, dibagi lagi menjadi 3.

Urutan observasi untuk citra wajah dibentuk dari blok *image* berukuran  $L_x \times L_y$  (*Sampling Window*) yang diperoleh dengan melakukan *scanning* citra dari kiri ke kanan dan atas ke bawah, seperti ditunjukkan pada Gambar 3. Blok *image* yang bersebelahan *overlap* dengan jarak  $P_y$  baris pada arah vertikal, dan  $P_x$  kolom pada arah *horizontal*.



Gambar 2. Struktur EHMM Citra Wajah

*Observation vector* diperoleh dari enam 2-D DCT *coefficient* (3x2 *low-frequency array*). Jika sebagai *observation vector* digunakan nilai piksel pada tiap blok *image* maka ukuran dari *observation vector* adalah  $L_x \times L_y$ .



Gambar 3. Urutan Observasi Pada Citra Wajah

Misal  $L_x = 8$  dan  $L_y = 10$  maka terdapat 80 nilai observasi, yaitu sekitar 15 kali lipat dibandingkan penggunaan enam 2-D DCT *coefficient*, sehingga bisa mengurangi kompleksitas perhitungan. Disamping itu pemakaian DCT *coefficient* juga bisa mengurangi masalah *noise*, rotasi dan pergeseran citra, serta faktor pencahayaan.

Berikut adalah persamaan 2D DCT *Transform* yang dilakukan pada tiap *sampling window*:

$$X[k_1, k_2] = \mathbf{a}[k_1] \mathbf{a}[k_2] \sum_{y=0}^{N_2-1} \sum_{x=0}^{N_1-1} X[x, y] \cos\left[\frac{(2x+1)k_1p}{2N_1}\right] \cos\left[\frac{(2y+1)k_2p}{2N_2}\right] \quad (11)$$

untuk  $k_1 = 0, 1, \dots, N_1 - 1$

dan  $k_2 = 0, 1, \dots, N_2 - 1$

$$\mathbf{a}[k] = \begin{cases} \sqrt{\frac{1}{N}} & \text{untuk } k = 0 \\ \sqrt{\frac{2}{N}} & \text{untuk } k = 1, 2, \dots, N - 1 \end{cases}$$

dimana:

$k_1$  = posisi kolom piksel dari image

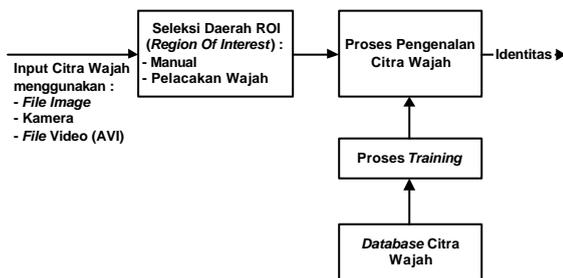
$k_2$  = posisi baris piksel dari image

$N_1$  = lebar image dalam piksel

$N_2$  = tinggi image dalam piksel

## 2. ANALISIS DAN DESAIN

Garis besar sistem kerja perangkat lunak ditunjukkan pada Gambar 4. berikut.



Gambar 4. Garis Besar Sistem Kerja Perangkat Lunak

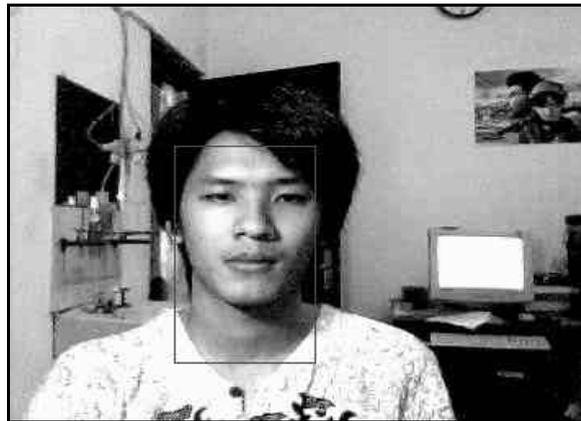
Hasil dari proses pelacakan citra wajah berupa sebuah persegi pada *frame citra* kamera atau video AVI untuk menunjukkan daerah wajah yang terlacak, seperti ditunjukkan pada Gambar 5.

Secara garis besar sistem kerja dari proses pengenalan citra wajah dibagi menjadi dua bagian, yaitu: proses pelatihan *face database*, dan proses pengenalan citra wajah. Bagian pertama, yaitu proses pelatihan *face database*, memiliki cara kerja sebagai berikut:

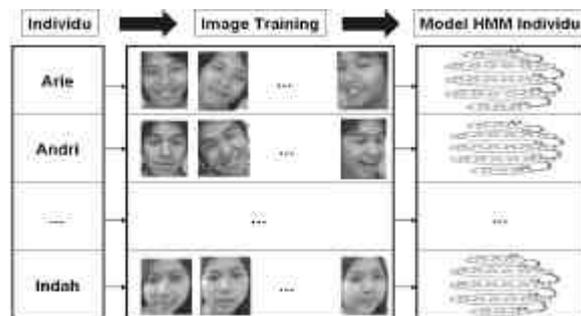
- Pada tiap individu di *database* diambil beberapa variasi pose wajah.

- Variasi pose dari tiap individu tersebut digunakan untuk pelatihan, yaitu membuat model *Hidden Markov* (HMM) yang mewakili tiap individu.

Semakin banyak variasi pose yang diambil, maka model HMM akan semakin mewakili karakteristik individu tersebut, sehingga hasil pengenalan akan lebih akurat.



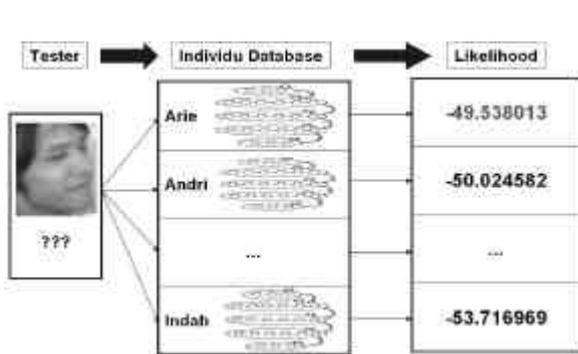
Gambar 5. Hasil Pelacakan (*Search indow*) Dengan *Outline Merah*



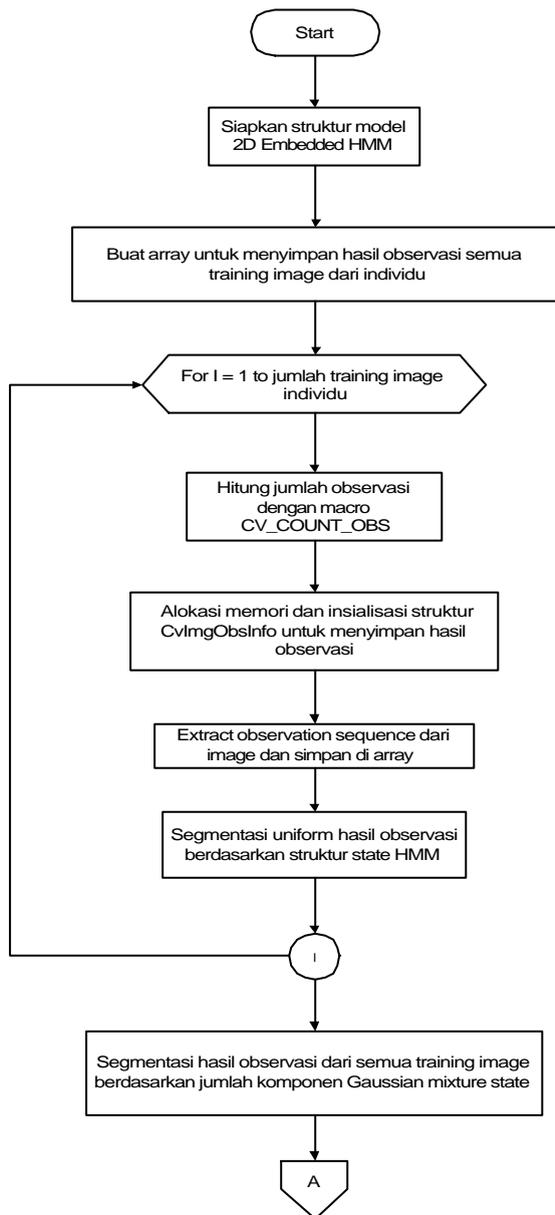
Gambar 6. Sistem Kerja Pelatihan *Face Database*

Bagian kedua, yaitu proses pengenalan citra wajah, memiliki cara kerja sebagai berikut:

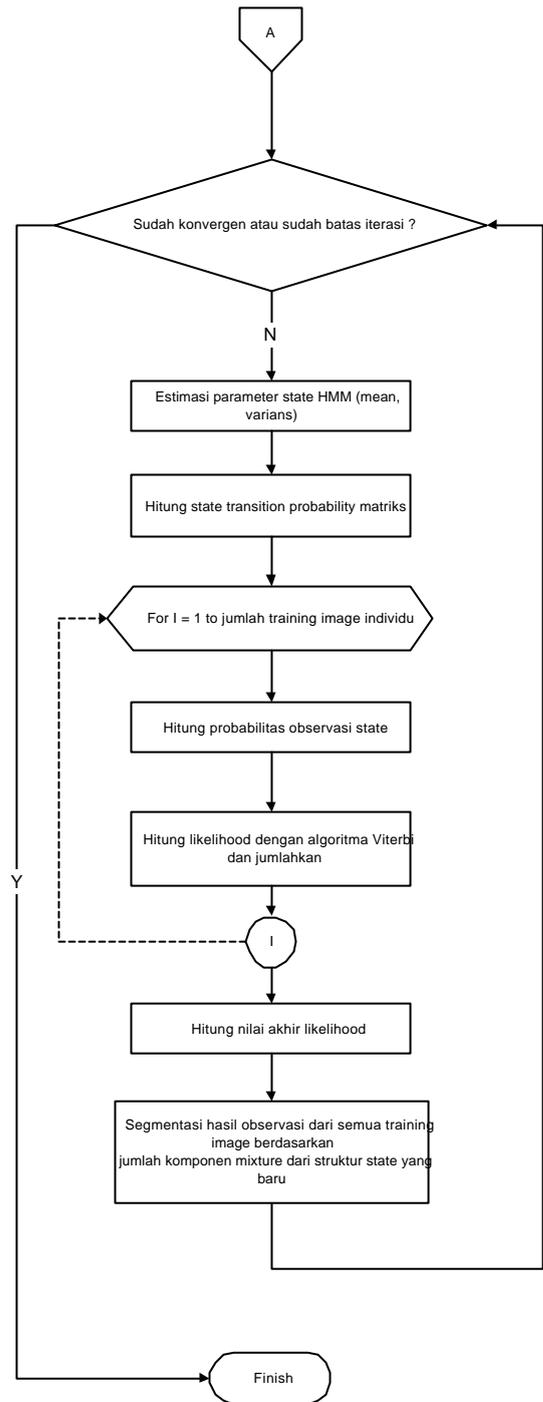
- Diambil satu pose wajah unik dari individu yang ada di *database*, dimana pose tersebut belum ada di *database* pose wajah individu tersebut sebelumnya.
- Citra wajah yang diujikan tersebut dibandingkan satu persatu dengan tiap model HMM dari individu, dan menghasilkan nilai kemiripan (*likelihood*).
- Hasil nilai *likelihood* yang tertinggi adalah identitas dari individu yang diuji tersebut.



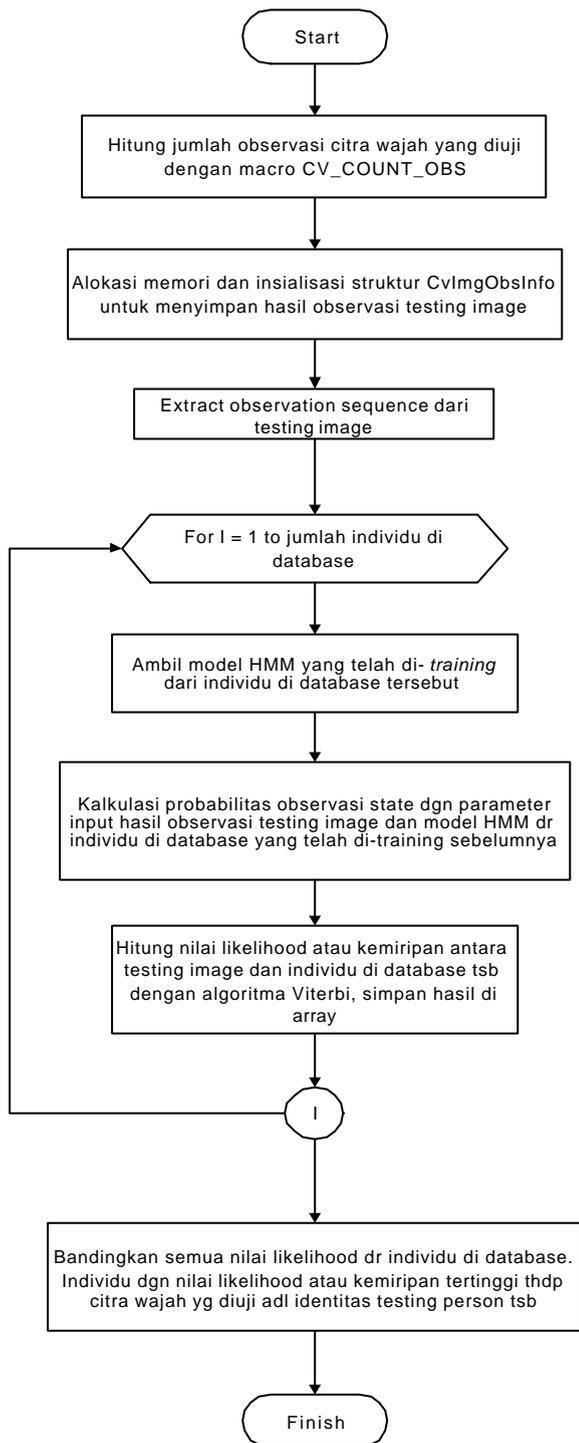
Gambar 7. Sistem Kerja Pengenalan Wajah



Gambar 8a. Flowchart Proses Training Database



Gambar 8b. Flowchart Proses Training Database (lanjutan)

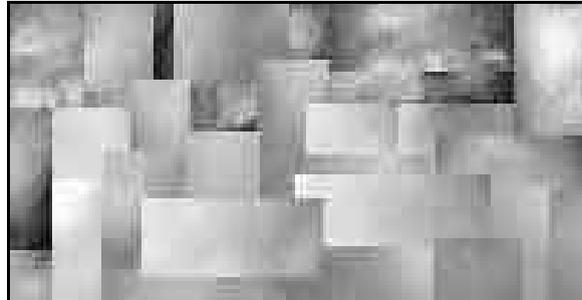


**Gambar 9. Flowchart Proses Pengenalan Citra Wajah**

### 3. HASIL DAN PENGUJIAN

Pelacakan dengan menggunakan algoritma *CamShift* berdasar pada warna kulit manusia, oleh karena itu, sebelum dapat diimplementasikan, perlu disiapkan terlebih dahulu citra warna kulit manusia yang akan

menjadi sampel (*template*). Pada pembuatan sampel warna kulit, beragam citra warna kulit manusia dari berbagai ras digabung menjadi satu sampel, sehingga sistem pelacakan dapat diterapkan terhadap berbagai jenis warna kulit manusia.



**Gambar 10. Sampel Warna Kulit Manusia**

Untuk pengujian terhadap sistem pelacakan pergerakan wajah, dilakukan:

- Pengujian terhadap pergerakan objek melintas (*Occlusion*).

Dilakukan pengujian pergerakan objek tangan melintasi wajah, dari hasil pengujian terlihat bahwa sistem pelacakan wajah cukup handal terhadap pergerakan objek melintas, karena *search window* akan melakukan penyerapan (*absorpsi*) terhadap objek melintas tersebut. Saat objek tangan melintasi wajah, ukuran dari *search window* berubah, yang juga mencakup objek tangan di samping objek wajah. Namun kemudian *search window* akan tetap melekat pada objek dengan nilai distribusi yang lebih dominan yaitu objek wajah, dimana saat objek tangan menjauh dari citra wajah, maka *search window* berubah kembali ke ukuran normal.

- Pengujian terhadap pergerakan latar belakang objek (*Background Distractor*).

Dari hasil pengujian terlihat bahwa sistem pelacakan wajah cukup handal terhadap pergerakan latar belakang objek, dimana *search window* tidak berpindah ke objek *background*, hal ini karena *search window* hanya menjangkau nilai distribusi probabilitas yang saling terhubung (*connected*) pada jarak terdekat, sehingga

*search window* akan tetap berada pada posisi objek wajah yang sedang dilacak walaupun terjadi pergerakan pada objek latar belakang.

- c. Pengujian terhadap variasi *noise* pada input citra wajah.

Dari hasil pengujian terlihat bahwa kemampuan *CamShift* melakukan pelacakan terhadap kondisi nilai distribusi probabilitas yang selalu berubah memberikan pengaruh terhadap kemampuan melakukan pelacakan dengan penambahan *noise*.

Penggunaan model warna sebagai basis pelacakan wajah, memberikan kehandalan terhadap pengaruh *noise*, hal ini karena warna *noise* memiliki nilai probabilitas yang sangat kecil untuk menjadi serupa dengan warna kulit.

- d. Pengujian terhadap pelacakan pergerakan dua objek wajah sekaligus.

Dari hasil pengujian terlihat bahwa proses pelacakan wajah berjalan baik walaupun dilakukan pelacakan terhadap dua objek wajah sekaligus. Dengan proses komputasi dari algoritma *CamShift* yang tidak terlalu kompleks dan cukup cepat, bisa dilakukan pelacakan wajah secara *real-time*, yaitu setiap pergantian *frame* kamera dengan kecepatan sekitar 15 *frame/second* terhadap dua objek wajah sekaligus.

Untuk pengujian terhadap sistem pengenalan wajah, dilakukan:

- Pengujian terhadap pose wajah *frontal*.
- Pengujian terhadap pose wajah *non-frontal*.
- Pengujian terhadap variasi *noise* pada input citra wajah.
- Pengujian terhadap variasi iluminasi atau pencahayaan pada input citra wajah.
- Pengujian terhadap variasi ekspresi pada input citra wajah.
- Pengujian terhadap variasi parameter HMM (HMM *Topology*).
- Perbandingan tingkat akurasi pengenalan wajah antara metode *Embedded Hidden Markov Model* dengan metode *Gabor Filter*.
- Pengujian terhadap proses pengenalan citra wajah secara *real-time*.

Sebelum melakukan uji pengenalan wajah, dibuat terlebih dahulu *database* citra wajah (*training image*) dan disiapkan pula citra wajah yang akan diujikan (*test image*).

Dalam pengujian ini, digunakan dua kelompok *database* citra wajah, *database* pertama adalah citra wajah dari mahasiswa/i jurusan Teknik Informatika Universitas Kristen Petra, Surabaya-Indonesia (Gambar 11), yang digunakan pada pengujian a hingga d, dan pengujian f. Terdapat 341 citra wajah pria maupun wanita (31 individu x 11 pose), yaitu pose wajah: *frontal*, miring ke kiri dan kanan, hadap kiri dan kanan, hadap atas dan bawah, hadap kiri atas, kanan atas, kiri bawah, dan kanan bawah, dengan ekspresi wajah normal.

*Database* kedua adalah *Yale Face Database* yang di-download dari internet dengan alamat URL sebagai berikut:

<ftp://plucky.cs.yale.edu/%2f/CVC/pub/images/yalefaces/yalefaces.tar> yang digunakan pada pengujian e (Gambar 12). Terdapat 126 citra wajah pria maupun wanita (14 individu x 9 ekspresi wajah), yaitu ekspresi wajah: normal dengan cahaya tengah (*centerlight*), berkacamata, tanpa kacamata, gembira, sedih, normal, mengantuk, terkejut, mengedipkan mata, dengan pose *frontal*.

Sedangkan citra wajah yang digunakan sebagai *test image*, merupakan gabungan antara citra wajah yang juga berada di *database* maupun tidak, namun masih dalam kelompok citra wajah yang sama (mahasiswa/i Petra dan *Yale*).



**Gambar 11. Database Citra Wajah Mahasiswa/i Jurusan Teknik Informatika, Universitas Kristen Petra**



Gambar 12. Yale Face Database

Tabel 1. Perbandingan Tingkat Akurasi Pengenalan Wajah Antara Metode EHMM Dan Gabor Filter

Pengujian	Tingkat Akurasi Pengenalan Wajah	
	Metode EHMM	Metode Gabor Filter
Pose Frontal (Pengujian a.)	82.76 %	79.31 %
Pose Non-Frontal (Pengujian b.)	100 %	24.19 %
Variasi Noise (Pengujian c.)	44.44 %	88.88 %
Variasi Iluminasi (Pengujian d.)	22.22 %	22.22 %

#### 4. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Sistem pelacakan wajah berdasarkan warna kulit menggunakan algoritma *CamShift* berjalan cukup baik dan cepat karena kompleksitas perhitungan yang tidak terlalu tinggi, hal ini terbukti dengan penggunaan algoritma tersebut untuk pengembangan sistem, yaitu pelacakan dua objek wajah sekaligus yang dilakukan tiap pergantian *frame* kamera atau video AVI (dengan *framerate* hingga 15 *frame per second*), yang berarti dilakukan sekitar 30 kali pelacakan tiap detiknya, dan sistem masih berjalan normal.
- Algoritma *Embedded Hidden Markov Models* dengan pengambilan 9 koefisien DCT (*Discrete Cosine Transform*) dari blok *image* sebagai *observation vector* dapat digunakan untuk mengenali pose

wajah *frontal* dengan cukup akurat, yaitu dengan tingkat akurasi pengenalan mencapai 82.76 %, dengan jumlah citra wajah pada *database* sebanyak 341 citra (31 individu x 11 pose).

Untuk pose wajah *non-frontal*, tingkat akurasi pengenalan bisa mencapai sempurna 100 %, walaupun hal ini tidak bersifat mutlak, karena dipengaruhi oleh beberapa faktor, seperti: tingkat pencahayaan, kualitas *training image*, *testing image* yang digunakan dalam pengujian.

- Variasi *noise* dan iluminasi atau pencahayaan, mempunyai pengaruh yang cukup besar terhadap tingkat akurasi pengenalan wajah. Hal ini dibuktikan dengan penurunan tingkat akurasi pengenalan secara signifikan, yaitu sebesar 44.44% dengan variasi *noise*, dan 22.22 % dengan variasi iluminasi.
- Variasi ekspresi wajah bisa dikenali dengan cukup baik oleh sistem dengan tingkat akurasi pengenalan mencapai sempurna 100 %, walaupun hal ini juga tidak bersifat mutlak, karena dipengaruhi oleh beberapa faktor lain seperti telah disebutkan sebelumnya.
- Pemilihan struktur *state* atau topologi dari model HMM yang tepat, memiliki pengaruh yang cukup penting dalam tingkat akurasi pengenalan. Struktur *state* yang ideal adalah dengan jumlah *super state* sebanyak 5, dan jumlah *embedded state* pada masing-masing *super state*, berturut-turut 3-6-6-6-3. Struktur *state* dibuat untuk dapat memodelkan struktur wajah manusia ke dalam *state-state* dari *Hidden Markov Model* dengan sebaik mungkin. Jumlah *embedded state* pada *super state* untuk bagian wajah yang penting dan merupakan ciri khas dari tiap individu seperti mata, hidung, dan mulut diperbanyak. Tingkat akurasi pengenalan wajah terhadap pengujian dengan jumlah *super state* sebanyak 6, dan jumlah *embedded state* berturut-turut 3-6-6-6-6-3 yaitu sebesar 70 %, terjadi penurunan sekitar 10 % jika dibandingkan dengan struktur

*state* ideal, di samping itu kompleksitas perhitungan juga meningkat akibat bertambahnya jumlah *super state* dan *embedded state*, sehingga waktu yang dibutuhkan untuk proses *training* dan *recognition* lebih lama.

Tingkat akurasi pengenalan wajah terhadap pengujian dengan jumlah *super state* sebanyak 3, dan jumlah *embedded state* berturut-turut 3-6-3, sebesar 50%, terjadi penurunan sekitar 30% jika dibandingkan dengan struktur *state* ideal, walaupun proses *training* dan *recognition* menjadi lebih cepat, karena kompleksitas perhitungan menurun akibat berkurangnya jumlah *super state* dan *embedded state* pada topologi HMM.

## DAFTAR PUSTAKA

1. Bradski, Gary R. *Computer Vision Face Tracking For Use in a Perceptual User Interface*. Microcomputer Research Lab, Intel Corporation, Santa Clara, CA: Intel Corporation, 1998.
2. Rabiner, Lawrence R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of IEEE*, vol. 77, no. 2: pp. 257-286. February 1989.
3. Nefian, Ara V. and Hayes, M.H. *A Hidden Markov Model-Based Approach for Face Detection and Recognition*. Atlanta, GA: Georgia Institute of Technology, August 1999.
4. Pantouw, Roy O. *Pelacakan dan Pengenalan Wajah menggunakan Metode Gabor Filter*. Surabaya, Indonesia: Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra, Juli 2002.
5. Yale Face Database. New Haven, Connecticut: Center for Computational Vision and Control, Yale University.  
<ftp://plucky.cs.yale.edu/%2f/CVC/pub/images/yalefaces/yalefaces.tar>.