

# TRANSLASI BAHASA ISYARAT

Juniar Prima Rakhman<sup>1)</sup>, Nana Ramadijanti, S.Kom, M.Kom<sup>2)</sup>, Edi Satriyanto S.Si, M.Si<sup>3)</sup>

*Jurusan Teknik Informatika, PENS –ITS Surabaya  
Kampus ITS Sukolilo, Surabaya 60111  
081320445021*

**E-mail :** [jeteye\\_master@msn.com](mailto:jeteye_master@msn.com)

## Abstrak

Masyarakat tuna rungu pada umumnya menggunakan bahasa isyarat sebagai alat komunikasi utamanya. Bahasa isyarat mengutamakan komunikasi visual, pengguna bahasa ini menggunakan orientasi, bentuk dan gerakan tangan, lengan, tubuh, serta ekspresi wajah untuk mengungkapkan ekspresi mereka. Tetapi cara komunikasi ini sering menyulitkan/membatasi komunikasi dengan orang lain yang normal, karena perbedaan komunikasinya itu kurang dipahami oleh lawan komunikasinya. Untuk mengatasi keterbatasan komunikasinya tersebut diperlukan upaya penterjemahan bahasa isyarat menjadi lisan. Dengan demikian akan terjadi komunikasi yang lebih mudah antar kaum tuna rungu dengan masyarakat umum.

Untuk menyelesaikan proyek akhir ini, digunakan kamera webcam sebagai alat bantu untuk menangkap gambar dari tangan pengguna. Teknik yang digunakan adalah dengan menangkap posisi tangan, mengekstrak bentuk dari tangan tersebut, kemudian mengklasifikasinya. Untuk mencari letak tangan dari setiap frame yang dihasilkan, penulis menggunakan HaarClassifier yang sebelumnya telah dilakukan training terlebih dahulu. Kemudian untuk mengekstrak bentuk tangan digunakan skin detection dan noise removal yang kemudian dilanjutkan dengan thresholding dan normalisasi. Setelah bentuk tangan ini didapatkan, maka gambar biner bentuk tangan ini diklasifikasikan berdasarkan kumpulan gambar-gambar isyarat tangan yang digunakan sebagai data training. Algoritma klasifikasi yang digunakan penulis adalah algoritma K Nearest Neighbors.

Sistem ini mampu mengenali 19 isyarat huruf tangan dari 26 isyarat yang ditargetkan. Rata-rata akurasi yang dihasilkan sistem ini adalah 89.68%. Nilai akurasi ini dapat bervariasi tergantung dari konsistensi data training dan noise yang dihasilkan.

*Kata Kunci : Skin detection, HaarClassifier, OpenCV, Machine Learning.*

## 1. Pendahuluan

Teknologi komunikasi dengan perangkat komputer telah maju pesat dan telah diterapkan dalam berbagai bidang atau peralatan lain. Teknologi komunikasi tersebut antara lain yang berkaitan dengan pengolahan dan penterjemahan citra atau isyarat menjadi bahasa yang mudah dipahami oleh masyarakat umum, atau penterjemahan sinyal pada peralatan tertentu. Misalnya, penggunaan teknologi tersebut dalam prinsip kerja atau konstruksi peralatan yang digunakan dalam bidang keamanan, medis, atau robotik. Teknologi (penterjemahan citra) tersebut sangat mungkin dimanfaatkan untuk menyusun paket teknologi komunikasi khusus untuk mereka yang mempunyai keterbatasan fisik.

Karena kekurangan kondisi fisiknya, masyarakat tuna rungu umumnya menggunakan bahasa isyarat sebagai

alat komunikasi utamanya. Bahasa isyarat mengutamakan komunikasi visual, pengguna bahasa ini menggunakan orientasi, bentuk dan gerakan tangan, lengan, tubuh, serta ekspresi wajah untuk mengungkapkan pikiran mereka.

Sama halnya dengan bahasa yang kita gunakan untuk berbicara, bahasa isyarat tidak bersifat universal. Bahasa isyarat unik dalam jenisnya di setiap negara. Untuk Indonesia, sistem yang sekarang umum digunakan adalah Sistem Isyarat Bahasa Indonesia (SIBI) yang sama dengan bahasa isyarat Amerika (ASL - American Sign Language).

Tetapi cara komunikasi tersebut sering menyulitkan/membatasi komunikasi dengan orang lain yang normal, karena perbedaan komunikasinya itu kurang dapat difahami oleh masyarakat umum atau lawan komunikasinya. Untuk mengatasi keterbatasan komunikasi tersebut diperlukan upaya penterjemahan bahasa isyarat (khusus yang digunakan oleh tuna rungu) tersebut menjadi bahasa yang umum atau mudah difahami orang lain/normal. Dengan demikian akan terjadi komunikasi

yang mudah antar masyarakat tuna rungu dengan masyarakat umumnya.

Berkaitan dengan upaya tersebut di atas, maka akan dikaji penterjemahan bahasa isyarat menjadi bahasa tulis sehingga memungkinkan terjadinya komunikasi yang mudah antara kaum tuna rungu dengan masyarakat yang kondisi fisiknya normal.

## 2. Model, Analisa, Desain dan Implementasi

### 2.1 Dasar Teori

#### 2.1.1 Skin Color Detection

Untuk mengimplementasikan skin color detection, penulis menggunakan metode peer model yang telah dimodifikasi. Adapun pemodifikasian metode ini adalah dengan mendapatkan rata-rata dari kulit manusia dari berbagai macam ras mayoritas, dan juga dalam kondisi yang berbeda-beda pula. Namun alangkah lebih baik jika dilakukan normalisasi sebelum dilakukan proses diatas. Berikut ini adalah rumus hasil modifikasi dari metode peer model:

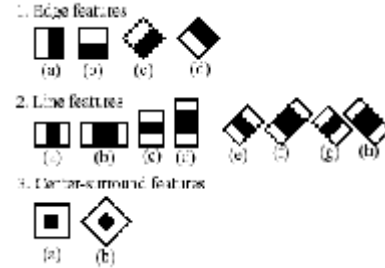
$$r > 95 \text{ and } g > 40 \text{ and } b > 20 \text{ and } \max(r,g,b) - \min(r,g,b) > 15 \text{ and } \text{abs}(r-g) > 15 \text{ and } r > g \text{ and } r > b \text{ and } g > b$$

$$H \leq 19 \text{ and } S \geq 48$$

Figure 2.1 model peer modifikasi

#### 2.1.2 Object Detection

Metode yang kami pakai untuk mendeteksi adanya obyek adalah HaarClassifier, yaitu metode object detection yang membangun sebuah boosted rejection cascade, yang akan membuang data training negative sehingga didapatkan suatu keputusan untuk menentukan data positif. Haar Classifier adalah metode supervised learning yaitu metode yang membutuhkan data training untuk dapat mendeteksi obyek-obyek tertentu. Untuk itu, dilakukan training data terlebih dahulu sebelum akan menggunakan metode ini.



Gambar 2.1 Haar fitur

Tiga (3) tipe kotak(rectangular) fitur:

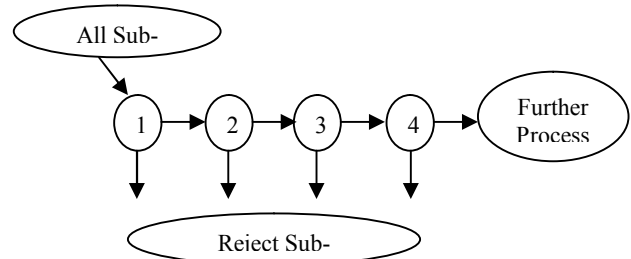
- Tipe two-rectangle feature
- Tipe three-rectangle feature
- Tipe four-rectangle feature

Sedangkan untuk menghitung nilai dari rectangle tersebut digunakanlah Integral Image, adalah representasi tengah dari suatu citra yang terdiri dari jumlah nilai keabu-abuan.

$$ii(x, y) = \sum_{x' < x, y' < y} i(x', y')$$

Figure 2.1 persamaan integral image

Setelah mendapatkan nilai integral image, langkah selanjutnya adalah mencari cascade classifier, adalah sebuah rantai stage classifier, dimana setiap stage classifier digunakan untuk mendeteksi apakah didalam image subwindow terdapat obyek yang diinginkan (object of interest).



Gambar 2.2. Cascade Classifier

#### 2.1.3 K Nearest Neighbors Classifier

Algoritma *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut.

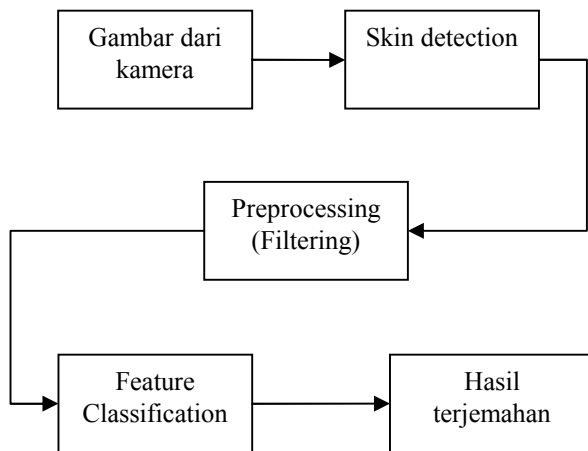
Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas *c* jika kelas *c* merupakan klasifikasi yang paling banyak ditemui pada *k* buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean.

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah  $k$  buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Ketika jumlah data mendekati tak hingga, algoritma ini menjamin error rate yang tidak lebih dari dua kali *Bayes error rate* (error rate minimum untuk distribusi data tertentu).

## 2.2 Implementasi

Sebelum system diimplementasikan, terlebih dahulu dibuat sistematika perancangan system yang merepresentasikan keseluruhan system agar didapatkan hasil yang optimal. Berikut ini adalah blok diagram system kami:



Gambar 2.3 blok diagram system

Blok diagram diatas merupakan keseluruhan proses untuk mengimplemtasikan proyek akhir. Dimana setelah gambar didapatkan dari kamera, proses yang pertama kali dilakukan adalah deteksi warna kulit. Kemudian dilakukan proses filtering. Setelah didapatkan obyek yang dikehendaki, maka langkah yang terakhir adalah menklasifikasi data tersebut.

## 2.3 Analisa

Pada pembahasan berikut ini berisi tentang pengujian perangkat lunak. Pengujiana ini dimaksudkan untuk melihat apakah perangkat lunak yang telah dibuat sesuai dengan yang telah direncanakan atau belum. Pengujian dilakukan dengan melalui beberapa tahap secara berkesinambungan. Pada pengujian ini yang pertama kali dilakukan adalah pengujian bagian-bagian rutin yang telah dibuat, kemudian dilakukan pengujian perangkat lunak secara keseluruhan.

Pengujian perangkat lunak ini bertujuan untuk mengetahui hasil dari perencanaan perangkat lunak yang telah dibuat, apakah sesuai dengan yang diharapkan, selain itu juga berfungsi untuk mengetahui kelemahan dari system yang ada.

Karena sebagai media untuk pengambilan gambar menggunakan kamera, maka factor cahaya sangat berperan penting dalam proses capture gambar.

### 2.3.1 Skin Detection

Berikut ini adalah hasil pengujian deteksi warna kulit pada model RGB an HSV dengan sumber cahaya berbeda:

Original	Skin RGB	Skin HSV
30 FPS	23-24 FPS	12-13 FPS
Original	Skin RGB	Skin HSV
20 FPS	18-19 FPS	11-12 FPS
Original	Skin RGB	Skin HSV
13 FPS	11-12 FPS	10-11 FPS

Gambar 2.4 deteksi warna kulit

Dari hasil pengujian pada tabel 4.2, 4.3, dan 4.4 dapat kita lihat bahwa metode skin detection pada

HSV colorspace mempunyai konsistensi yang lebih baik baik pada lingkungan dengan intensitas cahaya yang rendah, cukup, maupun tinggi..

### 2.3.2 Posisi Tangan

Pengujian selanjutnya adalah pendeteksian tangan dengan posisi yang berbeda, tetapi menggunakan jarak yang sama dan intensitas cahaya yang sama pula. Berikut adalah tabel hasil pengujian:

Subyek	Tegak	30 <sup>0</sup>	45 <sup>0</sup>	90 <sup>0</sup>	Terbalik
1	V	V	V	X	X
2	V	X	X	X	X
3	V	V	X	X	X
4	V	X	X	X	X
5	V	X	X	X	X
6	V	V	X	X	X
7	V	V	X	X	X
8	V	V	X	X	X
9	V	V	X	X	X
10	V	X	X	X	X
<b>rata</b>	<b>100%</b>	<b>60%</b>	<b>10%</b>	<b>0%</b>	<b>0%</b>

Tabel 2.2 deteksi tangan dengan posisi berbeda

Dari data diatas menunjukkan bahwa system pendeteksian tangan yang dibuat hanya dapat digunakan untuk mendeteksi tangan dalam posisi tegal, dan tidak berlaku untuk posisi yang lainnya. Hal ini terjadi karena sesuai dengan data training yang kami pakai selama training Haar Classifier.



Gambar 2.5 deteksi tangan dengan posisi berbeda, (a) tegak, (b) 45<sup>0</sup>, (c) 90<sup>0</sup>

### 2.3.3

Hal tersebut terjadi karena metode Haar Classifier mencari classifier ke seluruh window untuk dideteksi, sehingga hal tersebut menyebabkan program menjadi tidak begitu cepat. Oleh karena itu, system ini hanya optimal jika mendeteksi obyek dalam keadaan diam atau bergerak tetapi dalam keadaan lambat.

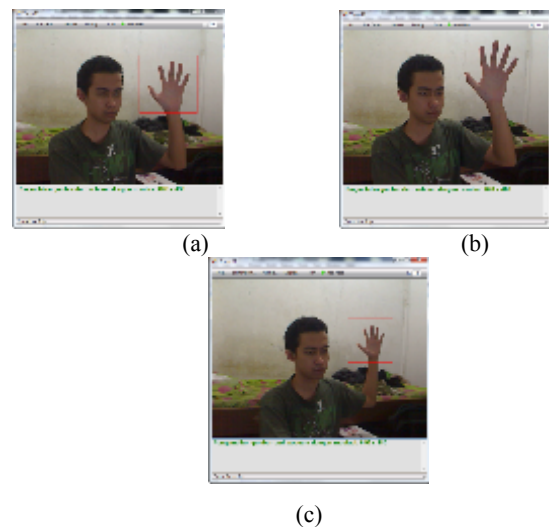
### 2.3.4 Jarak

Pengujian ini dilakukan dengan menggunakan parameter jarak yang berbeda tetapi dengan jumlah cahaya yang sama dan posisi telapak tangan yang sama. Berikut ini adalah hasil pengujiannya:

Subyek	30 cm	50 cm	70 cm	90 cm	110 cm
1	V	V	V	V	X
2	V	V	X	X	X
3	V	V	V	V	X
4	V	V	V	V	X
5	V	V	V	X	X
6	V	V	V	X	X
7	V	V	V	X	X
8	V	V	V	X	X
9	V	V	V	X	X
10	V	V	V	V	X
<b>Rata-rata</b>	<b>100%</b>	<b>100%</b>	<b>90%</b>	<b>40%</b>	<b>0%</b>

Tabel 2.4 deteksi tangan terhadap jarak

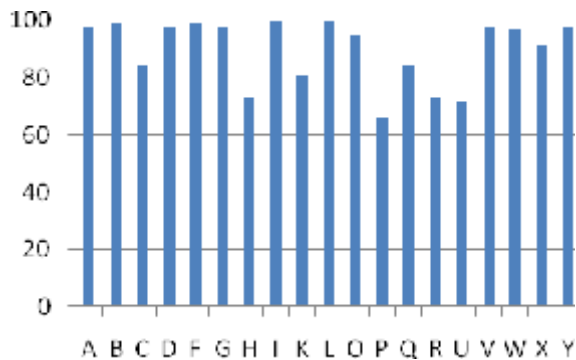
Dari tabel diatas dapat kita lihat bahwa jarak optimal yang mampu dideteksi oleh system adalah jarak 30 cm sampai dengan 70 cm. Hal ini terjadi karena pada saat training data Haar Classifier, kami menggunakan data tangan sebesar 20x25 pixel. Jika anda menginginkan pendeteksian tangan dengan jarak yang lebih jauh dari 1 meter, maka lebih baik kita menggunakan data tangan sebesar 30x30 pixel.



Gambar 2.6 deteksi tangan dengan jarak berbeda, (a) 30 cm, (b) 50 cm, (c) 70 cm

### 3. Hasil

Apabila kondisi-kondisi seperti yang telah dijelaskan pada bab sebelumnya terpenuhi, maka tahap adalah mengklasifikasinya. Pada pengujian akan diklasifikasi 19 isyarat huruf yang dapat dibedakan dengan mudah yaitu huruf A, B, C, D, F, G, H, I, K, L, O, P, Q, R, U, V, W, X dan Y. Pengujian ini dijalankan oleh 10 orang dan setiap orang ini memberikan 10 isyarat tangan per huruf. Jumlah data training yang digunakan adalah 100 data per huruf dan nilai variabel K yang digunakan adalah 32.



Gambar 3.3 Akurasi klasifikasi secara keseluruhan.

Dapat kita lihat bahwa pada beberapa huruf seperti huruf P, H, R, dan U akurasi yang didapatkan kurang dari 80%. Ini disebabkan oleh beberapa faktor yaitu miripnya bentuk isyarat P dengan K dan R dengan U. Faktor lain yang mempengaruhi adalah konsistensi datang training yang dibuat, konsistensi ini dipengaruhi oleh beberapa hal seperti pencahayaan dan noise yang bervariasi.

### 4. Kesimpulan

Setelah melaksanakan pengujian dan analisa pada system “Translasi Bahasa Isyarat”, maka dapat diambil beberapa kesimpulan sebagai berikut:

- Jarak maksimal yang masih mampu dideteksi adalah 90 cm, selebihnya system tidak dapat mendeteksi adanya tangan.
- Untuk mendapatkan hasil yang optimal dalam pendeteksian tangan, maka posisi tangan harus dalam keadaan tegak.
- Metode modifikasi peer untuk mendeteksi warna kulit pada HSV color model dapat mendeteksi warna kulit dengan lebih baik dibanding metode

modifikasi peer untuk mendeteksi warna kulit pada RGB color model.

- Tahap noise removal dan normalisasi sangat efektif untuk menghasilkan data training yang konsisten.
- Algoritma KNN sebagai metode klasifikasi dapat memberi akurasi pendeteksian isyarat tangan yang cukup akurat. Ini ditunjukkan oleh akurasi sebesar 89.68%.
- Beberapa isyarat seperti huruf M, N, S, T, J dan Z belum dapat dideteksi karena sulit dibedakan dan menggunakan gerakan tangan.

### Daftar Pustaka

1. Open computer vision library. <http://sourceforge.net/project/opencvlibrary/>.
2. Tom Cassey, Tommy Chheng, Jefery Lien, “Camera Based Hand and Body Driven Interaction with Very Large Display”, ucsd lab, June 15, 2007.
3. Erdem Yoruk, Ender Konukoglu, Bulent Sankur, Jerome Darbon, “Shape-Based Hand Recognition”, Bogazici Lab, July 2006.
4. Neo, Naotoshi. “OpenCV Haar Training”. 2007.
5. Sun, Min. “OpenCV and face detection”. 2006.
6. Gary Bradski, Adrian Kaehler. “Learning OpenCV”. Oreilly team’s Intel. 2005.
7. Paul Viola and Michaels J. Jones. “Rapid Object Detection using boosted Cascade of Simple Features”. IEEE CVPR, 2001.
8. Jure Kovac, Peter Peer and Franc Solina. “Human Skin Colour Clustering For Face Detection”. University of Ljubljana Faculty of Computer and Information Science. Slovenia.
9. Robin Hewitt. “seeing with openCV part 2 – Finding Face in image”. Jan, 2007.
10. Gary Agam. “introduction to programming with OpenCV”. 2006.