

PENGENALAN UCAPAN METODA MFCC-HMM UNTUK PERINTAH GERAK ROBOT MOBIL PENJEJAK IDENTIFIKASI WARNA

Fitrilina, Rahmadi Kurnia, Siska Aulia

Program Studi Teknik Elektro, Fakultas Teknik

Universitas Andalas

Email: fitrilina@ft.unand.ac.id

ABSTRAK

Dewasa ini, teknologi human machine cukup berkembang dan mendapatkan perhatian.. Dua bidang yang cukup berperan dalam perkembangan teknologi human machine adalah pemrosesan suara dan pemrosesan image. Penelitian ini mengimplementasikan pemrosesan suara dan image pada sistem robot mobil sehingga robot dapat mendengar, melihat, memberi respon dan diperintah berdasarkan ucapan. Sistem pengenalan ucapan menggunakan Mel Frequency Cepstrum Coefficient (MFCC) dan Hidden Markov Model (HMM). Robot mobil melakukan tracking objek berdasarkan empat jenis warna yaitu merah kuning, hijau dan biru. Pada penelitian ini segmentasi menggunakan metode multilevel color thresholding pada ruang warna HIS. Keluaran dari sistem robot mobil dihubungkan ke motor servo standar sebagai penggerak kamera dalam proses tracking objek. Keluaran sistem berupa gerakan horizontal kamera dan respon maju yang dilakukan robot mobil berdasarkan ucapan yang dikenali. Hasil pengujian untuk pengenalan ucapan sumber terlatih 83,33% dan untuk sumber tidak dilatih sebesar 80,25%. Sedangkan pengujian untuk objek tracking berdasarkan ucapan yang dikenali mencapai keberhasilan 100%. Oleh karena itu dapat disimpulkan bahwa robot dapat mentracking dan merespon objek.

Kata kunci : pengenalan ucapan, segmentasi, tracking objek

I. PENDAHULUAN

Teknologi pada prinsipnya dikembangkan sebagai alat atau sarana yang dapat membantu dan memberi kemudahan bagi manusia untuk melakukan kegiatan dalam hidupnya. Seiring dengan perkembangan teknologi, manusia selalu menginginkan peningkatan kualitas dan kepraktisan dari alat-alat tersebut. Oleh karena itu dibutuhkan mesin-mesin yang dapat berinteraksi dengan manusia atau yang dikenal dengan teknologi *human machine*.

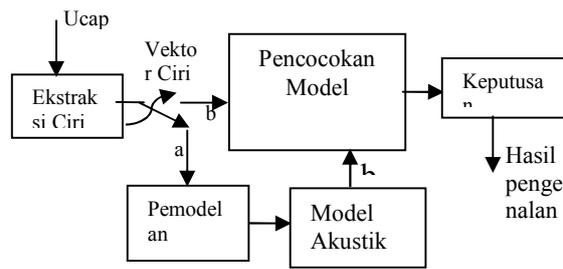
Dewasa ini, Teknologi *human machine* telah telah cukup berkembang. Salah satu bidang yang mendukung teknologi *human machine* adalah *Automatic Speech Recognition* (ASR). ASR berusaha menciptakan mesin yang memiliki kemampuan mengartikan informasi yang diucapkan manusia dan bertindak sesuai dengan informasi tersebut. Selain ASR, dalam bidang *human machine* juga berkembang teknologi visual mesin (*Machine vision*). Teknologi ini berusaha menciptakan mesin yang dapat mengenali area disekitarnya,

mempunyai persepsi yang sama dengan mata manusia dalam mengamati sebuah objek.

Pada penelitian ini, dikembangkan robot mobil yang dapat mendengar, melihat dan memberi respon berdasarkan perintah suara yang diberikan. Robot mobil dapat mencari warna yang sesuai dengan suatu perintah ucapan dari beberapa warna yang ditampilkan. Penelitian ini menggabungkan *speech processing* dengan *image processing* untuk *object tracking* video berdasarkan warna. Pada Penelitian ini sistem bersifat *isolated word recognitions* dengan ucapan merah, kuning, hijau dan biru, menggunakan kamera tunggal jenis *web camera* yang terhubung ke komputer dengan resolusi video adalah 320x240 piksel dan *frame rate* 30 fps, menggunakan robot jenis beroda, dimana motor servo DC satu poros dengan kamera. Motor servo yang digunakan adalah keluaran *parallax* jenis motor servo dc standar 180⁰ dan menggunakan Mikrokontroler jenis AT89C51.

II. TINJAUAN PUSTAKA

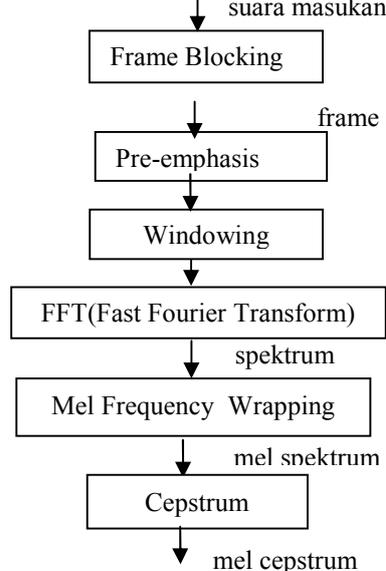
2.1 Sistem Pengenalan Ucapan



Gambar 1 : sistem pengenalan ucapan.
a).pelatihan b).pengenalan^[1]

Sistem pengenalan ucapan dibentuk melalui dua proses utama yaitu pelatihan (*training*) dan pengenalan (*recognition*). Secara umum proses pembentukan sistem pengenalan ucapan dapat dilihat pada gambar 1. Pada proses pelatihan dan pengenalan, dilakukan ekstraksi ciri terhadap setiap sinyal ucapan dari masing-masing kata sehingga dihasilkan vektor ciri untuk masing-masing ucapan. Pada proses pelatihan, vektor ciri digunakan untuk membentuk model akustik yang merepresentasikan suatu kata. Model akustik tersebut disimpan dan dijadikan sebagai pola referensi. Pada proses pengenalan, vektor ciri dibandingkan dengan semua model akustik pada basis data. Model akustik dengan peluang terbesar yang akan dipilih sebagai model akustik dari vektor ciri tersebut [1]. Pada penelitian ini ekstraksi ciri menggunakan MFCC dan proses pelatihan dan pengenalan menggunakan *Hidden Markov Model* (HMM).

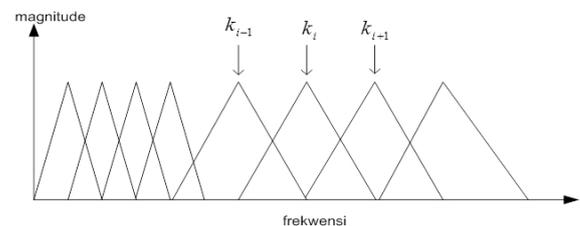
2.2 Mel Frequency Cepstral Coefficient



Gambar 2. Proses MFCC

Ekstraksi sinyal ucapan dengan menggunakan MFCC dapat dilihat seperti pada gambar 2 dan mengalami proses sebagai berikut :

- **Frame blocking** : membagi suara menjadi frame-frame dengan durasi 20-30 ms sehingga sinyal suara dapat dianggap stasioner
- **Pre-emphasis** : cuplikan sinyal dalam bentuk digital ditapis dengan menggunakan *FIR filter* orde satu untuk meratakan spektral sinyal .
- **Windowing** : dilakukan pada setiap frame untuk meminimalisasi ketidak kontinuan atau mengurangi distorsi spektrum pada awal dan akhir *frame*. Window yang digunakan adalah Hamming window
- **Fast Fourier Transform** : mengubah masing-masing frame N sampel dari domain waktu menjadi domain frekuensi.
- **Mel-Frequency Wrapping** :Skala frekuensi mel merupakan suatu skala yang bersifat linier untuk frekuensi di bawah 1000 Hz dan logaritmis pada frekuensi diatas 1000 Hz. Untuk mendapatkan koefisien mel maka digunakan mel filter bank. Mel filter bank adalah filter bank berbentuk segitiga seperti pada gambar 3



Gambar 3 Filter Bank Segitiga

Secara matematis *filter bank* segitiga ini dapat didefinisikan dalam bentuk hubungan seperti persamaan 4

$$F(i, k) = \begin{cases} 0 & k \leq k_{i-1} \\ (k - k_{i-1}) / (k_i - k_{i-1}) & k_{i-1} < k < k_i \\ 1 & k = k_i \\ (k - k_{i-1}) / (k_{i+1} - k_i) & k_i < k < k_{i+1} \\ 0 & k \geq k_{i+1} \end{cases} \quad (1)$$

Dimana: $F(i,k)$ = frekuensi dalam *filter bank*
 k = frekuensi yang bernilai 1000 Hz

- **Cepstrum** : Untuk mendapatkan koefisien cepstrum maka koefisien mel diproses

dengan menggunakan *Discrete cosinus transform (DCT)*.

2.3 Hidden Markov Model (HMM)

Hidden Markov Model berdasarkan deret observasi dapat dibedakan atas Diskrit HMM (DHMM) dan Continous HMM (CHMM). Pada penelitian ini deret observasi adalah hasil ekstraksi ciri dari sinyal suara sehingga diproses dengan menggunakan Continous HMM. Pada CHMM ada beberapa parameter model yaitu:

1. N adalah jumlah state pada model.
2. M adalah jumlah mixture gaussian
3. Matrik probabilitas transisi antar state adalah $(A=a_{ij})$, dimana

$$a_{ij} = P[q_{t+1} = j | q_t = i], 1 \leq i, j \leq N \quad (2)$$

4. Distribusi probabilitas observasi $\{B(j,k)\}$. Pada CHMM diperoleh dengan menggunakan Distribusi Gaussian
5. Probabilitas state awal, $\pi = \{\pi_i\}$ dimana

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N$$

Masalah - masalah Dasar Pada Hidden Markov Model

➤ **Masalah evaluasi**

Masalah evaluasi yaitu bagaimana menghitung probabilitas deret observasi $(O = O_1, O_2, \dots, O_T)$ terhadap model yang diberikan (λ) . Secara matematis dapat dirumuskan dengan $P(O|\lambda)$. Penyelesaian masalah ini dapat menggunakan metoda forward-backward [2].

❖ **Metoda Forward**

- 1) Inisialisasi

$$\alpha_i(1) = \pi_i b_i(O_1); \quad 1 \leq i \leq N \quad (3)$$

- 2) Induksi

$$\alpha_j(t+1) = \left[\sum_{i=1}^N \alpha_i(t) a_{ij} \right] b_j(O_{t+1}); \quad 1 \leq t \leq T-1; \quad 1 \leq j \leq N \quad (4)$$

- 3) Terminasi (terakhir)

$$P(O|\lambda) = \sum_{i=1}^N \alpha_i(t) \quad (5)$$

Dimana: α_j = Variabel forward

- $P(O|\lambda)$ = Peluang observasi terhadap λ
- N = jumlah state
- a_{ij} = matrik antar state
- π = matrik peluang state awal
- $b_j(O_t)$ = distribusi probabilitas observasi

❖ **Metoda Backward**

- 1) Inisialisasi

$$\beta_T(i) = 1; \quad 1 \leq i \leq N \quad (6)$$

- 2) Induksi

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j); \quad t=T-1, T-2, \dots, 1; \quad 1 \leq i \leq N \quad (7)$$

- 3) Terminasi

$$P(O|\lambda) = \sum_{i=1}^N \pi_i b_i(O_1) \beta_1(i); \quad 1 \leq i \leq N \quad (8)$$

Dimana: β_T = Variabel backward

➤ **Masalah Decoding**

Masalah decoding yaitu bagaimana memilih deret state yang optimal sehingga bisa merepresentasikan dengan baik deret observasi dan parameter model yang diberikan. Untuk mengatasi masalah ini dapat digunakan Algoritma Viterbi. Algoritma Viterbi memilih salah satu deret state yang terbaik sehingga dapat memaksimalkan peluang deret state terhadap deret observasi dan model yang diberikan^{[2][3]}.

➤ **Masalah Pelatihan (Training)**

Masalah pelatihan (*training*) yaitu bagaimana menentukan atau menyesuaikan parameter model (A, B, π) untuk memaksimalkan peluang deret observasi yang diberikan terhadap model. Untuk mengatasi hal ini dapat dilakukan dengan menggunakan metoda Baum-Welch atau disebut juga dengan metoda EM (*Expectation-Maximalization method*)^{[3][4][5]}.

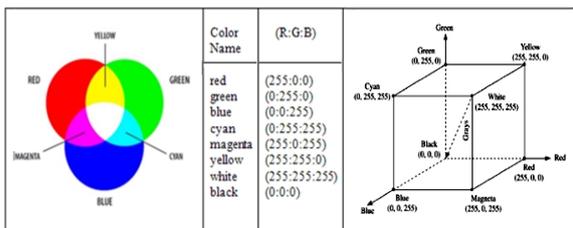
2.4 Ruang Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang (λ) yang berbeda. Warna sinar yang direspon oleh mata adalah sinar tampak (*visible*)

spectrum) dengan panjang gelombang berkisar dari 380 sampai 780 nm. Pada video digital ada banyak ruang warna yang telah dibuat. Namun pada penulisan ini hanya dijelaskan 2 jenis ruang warna yaitu RGB dan HIS karena ruang warna inilah yang dipakai dalam penelitian ini.

➤ **RGB (red, green, blue)**

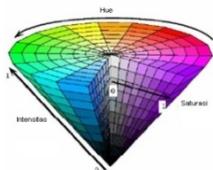
Warna pada citra disusun oleh tiga komponen warna yang disebut komponen RGB, yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Pada citra ini masing – masing warna primernya yaitu merah, hijau, dan biru terdiri dari 8 bit dan kemudian warna – warna ini digabungkan untuk menghasilkan warna yang sesuai [6]. Oleh karena itu, citra RGB terdiri dari 24 bit per piksel (bpp). Ruang warna RGB dapat dilihat pada gambar 4. Dalam ungkapan ini warna putih = RGB(255,255,255), sedangkan warna *hitam* = RGB(0,0,0).



Gambar 4. Ruang warna RGB

➤ **HSI (Hue Saturation Intensity)**

Sistem HSI adalah cara lain penggunaan ruang warna pada pengolahan citra, yang lebih intuitif untuk visi manusia. Sistem warna HSI merupakan warna pelengkap yang dihasilkan dari pengaturan warna RGB yang diproyeksikan ke diagonal utama yaitu dari warna hitam ke warna putih.



Gambar 5. HSI color space

Gambar 5 menunjukkan rentang warna nilai HSI dimana *Hue* menyatakan spektrum warna dominan dalam cahaya. *Hue* menunjukkan kandungan warna sebenarnya dari suatu citra. Nilai warna dalam *hue* dari 0 derajat sampai 360 derajat. *Saturation* menyatakan kejernihan spektrum warna dalam cahaya (ukuran kemurnian suatu warna). Semakin

jernih suatu warna, maka nilai saturasinya semakin tinggi. *Intensity* atau *value* menyatakan tingkat kecerahan cahaya dari warna yang dilihat dari batas yang paling redup atau gelap sampai batas yang paling terang. Berikut formula untuk konversi RGB ke HSI [8] :

$$H_1 = \arccos \frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \quad (9)$$

$$H = \begin{cases} H_1, & \text{if } B \leq G \\ 360^\circ - H_1, & \text{if } B > G \end{cases} \quad (10)$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \quad (11)$$

$$I = \frac{R + G + B}{3} \quad (12)$$

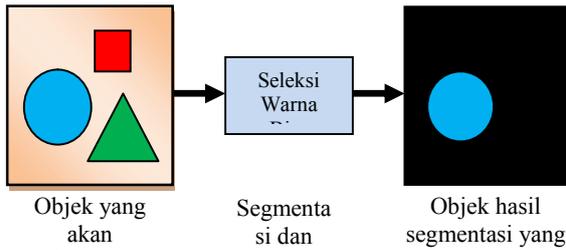
2.5 Multilevel Color Thresholding

Multilevel Color Thresholding merupakan metode segmentasi yang menggunakan banyak komponen warna pada ruang warna tertentu, seperti komponen *Red, Green, Blue* pada RGB atau komponen *Hue, Sat, Int* pada HSI. Dalam format HSI sebuah warna masih dikatakan sama jika nilai *hue* dari warna tersebut masih berada pada spektrum yang sama. Oleh karena itu nilai *hue* adalah parameter pemilih warna yang paling penting dalam segmentasi *multilevel color thresholding*, sedangkan pada saturasi hanya menggunakan range warna murni yakni 70 – 240 dan intensitas pada range 70 – 190 [7], dimana intensitas cahaya tidak terlalu gelap dan tidak terlalu terang. Beberapa sampel warna beserta nilai *hue* dapat dilihat pada tabel 1.

Tabel 1. RGB dan range *hue*[7]

Warna	RGB	Hue
Hitam	0,0,0	-
Merah	255,0,0	0 - 5
Hijau	0,255,0	80 - 110
Biru	0,0,255	120 - 160
Ungu	128,0,128	170 - 210
Kuning	255,255,0	20 - 60
Cokelat	150,75,0	10 - 20
Putih	255,255,255	-

Proses *thresholding* selanjutnya adalah melewati warna yang masuk kedalam *range hue* warna yang ditentukan sesuai dengan nilai pada tabel 1 dan memblok semua warna yang berada diluar *range* tersebut, maka hasil dari segmentasi adalah piksel-piksel dengan warna yang sama (disebut objek) tidak terkecuali warna pada latar belakang yang warnanya juga sama dengan warna objek. Proses segmentasi dapat dilihat pada gambar 6.



Gambar 6. Proses Segmentasi

2.6 Pelabelan dan Filtering

Proses pelabelan komponen hanya diperlukan bila dalam suatu citra biner terdapat lebih dari satu objek dengan tujuan memberi tanda yang membedakan objek yang satu dengan yang lainnya. Proses labelisasi nantinya akan digunakan untuk menghitung luas setiap objek. Penentuan luas dari piksel yang ditempati oleh suatu objek yang merepresentasikan ukuran dari objek tersebut.

$$Rumusnya: A = \sum_{i=0}^n \sum_{j=0}^m f(i, j) \quad (13)$$

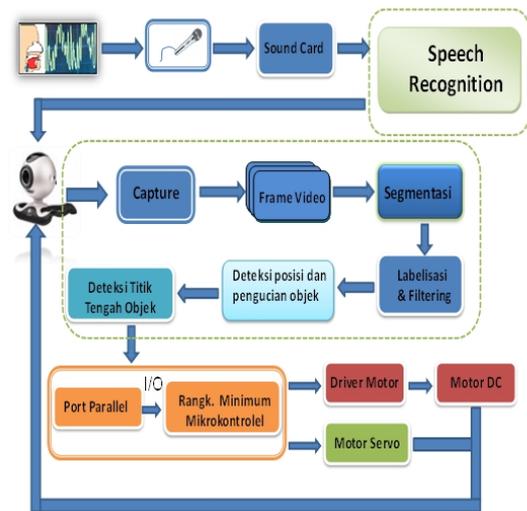
A = Area; Jumlah piksel.
 f(i,j) = 1, jika (i,j) adalah piksel objek.

Filtering merupakan proses untuk menghilangkan *noise* yang terdapat pada objek. *Filtering* dilakukan jika ditemukan objek yang memiliki luas kurang dari *threshold* luas yang telah ditetapkan. Jika objek dengan nilai luas ini ditemukan, maka objek tersebut dianggap sebagai *noise* dan difilter dengan cara memberi label 0 pada objek tersebut.

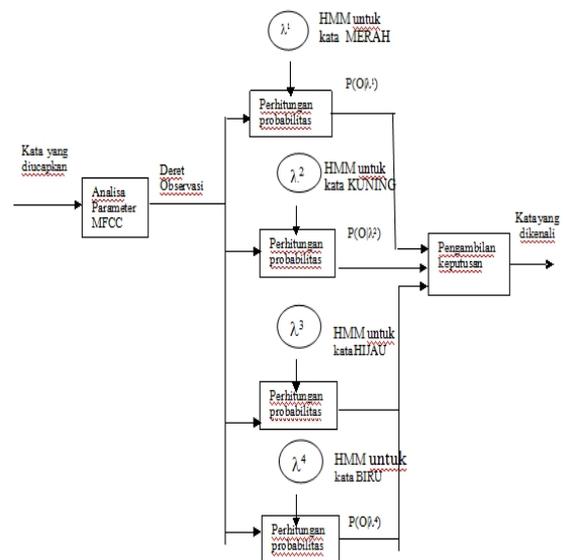
III. PERANCANGAN SISTEM

Pada penelitian ini dirancang sebuah robot mobil yang mampu mendengar, melihat, merespon dan diperintah dengan suara. Adapun

prosesnya dapat dilihat pada gambar 7. Proses perancangan diawali dengan pembentukan sistem pengenalan ucapan. Pada penelitian ini, sistem pengenalan ucapan menggunakan metoda HMM yang bersifat *isolated word*. Proses pengenalan ucapan untuk jenis *isolated word* yang digunakan pada penelitian ini dapat dilihat pada gambar 8.



Gambar 7. Blok diagram penelitian



Gambar 8. Blok Diagram pengenalan HMM tipe Isolated word [2]

Gambar 8 menunjukkan bahwa sinyal ucapan akan mengalami ekstraksi parameter menggunakan MFCC sehingga menghasilkan vektor ciri. Vektor ciri akan dihitung probabilitasnya terhadap semua model kata

yang sudah dibentuk ketika proses pelatihan dan akan dipilih probabilitas yang terbesar.

➤ **Capture dan Frame Video**

Setelah perintah dapat dikenali, penelitian ini dilanjutkan dengan proses penangkapan (*capture*) video digital menggunakan *web camera* berupa *frame-frame image*. Pertama dibuatkan *window* yang berada pada koordinat $x=0$ dan $y=0$, lebar jendela 320 piksel dan tinggi jendela 240 piksel. Jadi jumlah total piksel dari *window* adalah 320×240 piksel yaitu sebanyak 76.800 piksel. Selanjutnya *window capture* tersebut dikoneksikan dengan *driver capture (web camera)*. Kemudian sistem akan menangkap format video yang digunakan oleh *driver capture* untuk selanjutnya diubah dalam format *bitmap (bmp)*.

Tahap selanjutnya adalah melakukan pengecekan terhadap memori. Jika dalam memori terdapat *image*, baik *image* yang diolah maupun *image* hasil *capture* pada kondisi sebelumnya, maka *image* tersebut dihapus dari memori. Setelah *image* yang ada pada memori sebelumnya dihilangkan, maka kita mulai mengalokasikan alamat memori baru untuk *image* yang baru. Sistem akan memasukkan atau mengubah nilai yang sudah ada pada *property window*. Pada sistem dibuat sebuah fungsi yang berguna untuk memanggil *frame-frame* video kembali dari *capture window* yang terkoneksi dengan *capture driver*. Kecepatan *frame-nya* dibuat sebesar 30 *frame per second* (fps). Memori dialokasikan untuk setiap *frame*, sehingga tampilan video bisa dimunculkan.

➤ **Algoritma Segmentasi**

Adapun langkah-langkah untuk memisahkan objek dengan *background* adalah sebagai berikut:

1. *Scanning frame* pada titik (i,j) mulai dari titik $(0,0)$ sampai titik (m,n) sebagai representasi piksel pada baris i dan kolom j , dimana:
 $m = \text{lebar frame}$
 $n = \text{tinggi frame}$
2. Definisikan piksel pada titik (i,j) sebagai warna. Karena piksel warna dikenali dengan 24 bit, maka masing-masing piksel terdiri dari tiga komponen warna yaitu komponen *red*, *green* dan *blue*. Komponen *blue* terletak pada 8 bit pertama, komponen *green* terletak pada 8

bit kedua dan komponen *red* terletak pada 8 bit ketiga, jadi seolah-olah piksel dibaca sebagai warna *blue*, *green* dan *red*.

3. Konversi warna RGB tersebut kedalam ruang warna HSI.
4. Tentukan nilai *threshold* yang tepat pada parameter hue, saturasi dan intensitas untuk warna objek target.
5. Piksel yang berada pada nilai *threshold* akan di anggap sebagai objek dan akan diberi label 1. Sedangkan yang tidak masuk kedalam nilai *threshold* akan dianggap sebagai latar belakang dan diberi label 0.

➤ **Algoritma Pelabelan dan Pemfilteran**

1. *Scan frame* hasil segmentasi (label *layer*) mulai dari titik $(0,1)$ sampai titik $(m,n-1)$ sebagai representasi piksel pada baris i dan kolom j .
2. Jika piksel (i,j) adalah piksel objek (label *layer* = 1) dan belum memiliki label (label 1 dan 0 dianggap bukan label), periksa 8 tetangganya (kiri atas, atas, kanan atas, samping kanan, kanan bawah, bawah, kiri bawah, dan samping kiri). Jika salah satu saja dari piksel tetangga tersebut memiliki label (label tetangga $\neq 1$ atau $\neq 0$), maka salin label tersebut ke piksel (i,j) . Namun jika tidak satupun piksel tetangganya memiliki label, maka beri label baru pada piksel (i,j) dan salinkan labelnya ke 8 tetangga sekitarnya.
3. Karena ada kemungkinan dalam satu objek terdapat lebih dari 1 label, maka deteksi 2 label yang berdekatan (nol dianggap bukan label) dan simpan nilainya pada suatu tabel ekuivalen (label besar dan label kecil) label-label yang berdekatan.
4. Urutkan data pada tabel ekuivalen berdasarkan nilai label besar dari kecil ke besar.
5. *Scan ulang frame*, kemudian ganti label yang besar dengan label kecil yang merupakan pasangannya dalam tabel. Hasil dari proses ini yaitu setiap objek akan memiliki 1 label saja dan label ini berbeda dari objek lainnya.
6. Kembali ke langkah 3 jika masih ada piksel-piksel dalam sebuah objek belum

berlabel sama (berdekatan tapi berbeda), sehingga didapatkan 1 objek 1 label.

7. Hitung luas dari masing-masing objek yang terdeteksi.
8. Jika ditemukan objek yang memiliki jumlah piksel (luas) lebih kecil dari jumlah piksel objek yang telah ditentukan, maka objek tersebut dianggap sebagai *noise* dan difilter dengan cara memberi label 0 pada objek tersebut.

➤ **Algoritma Menghitung Luas Objek dan Filtering**

1. *Scanning* piksel hasil proses pelabelan pada titik (i,j) mulai dari titik (0,0) sampai titik (m,n) sebagai representasi piksel pada baris i dan kolom j. Lakukan hal ini pada setiap *frame*.
2. Jumlahkan setiap piksel yang memiliki label yang sama. Simpan hasil penjumlahan tersebut sesuai dengan labelnya. Lakukan hal ini pada setiap *frame*.
3. Jika ditemukan objek yang memiliki jumlah piksel diluar dari jumlah piksel objek yang ditentukan, maka objek tersebut dianggap sebagai *noise* dan dihilangkan (pemfilteran) dengan cara memberi label 0 pada objek tersebut.

➤ **Algoritma Penentuan Posisi Objek**

Posisi objek dapat ditentukan dari 4 titik terluar dari objek. Ke-4 titik ini membentuk sebuah rektanguler sebagai indikator objek terdeteksi. Titik tengah dari rektanguler ini bisa diwakilkan sebagai posisi dari objek. Sebagian besar algoritma menggunakan proses *scanning* frame yang berulang-ulang dengan lebar *m* dan tinggi *n*. Prosedur penentuan 4 titik tersebut sebagai berikut:

1. *Scan frame* dari kiri ke kanan dan atas ke bawah mulai piksel (0,0) ke (m,n) sampai ditemukan piksel objek (titik y_{min}).
2. *Scan frame* dari kiri ke kanan dan bawah ke atas mulai piksel (m,0) ke (0,0) sampai ditemukan piksel objek (titik y_{max}).
3. *Scan frame* dari atas ke bawah dan kiri ke kanan mulai piksel (0,0) ke (m,n) sampai ditemukan piksel objek (titik x_{min}).
4. *Scan frame* dari atas ke bawah dan kanan ke kiri mulai piksel (0,n) ke (m,0)

sampai ditemukan piksel objek (titik x_{max}).

5. Tentukan titik tengah 4 titik tersebut dengan rumus: $x = x_{max} + x_{min} / 2$ (14)
 $y = y_{max} + y_{min} / 2$ (15)

➤ **Algoritma Kalibrasi Pixel Kamera terhadap Sudut Servo**

Resolusi frame adalah 320x240 piksel sedangkan motor servo mampu bergerak sejauh 180 derajat. Sehingga diperlukan kalibrasi antara satuan piksel terhadap derajat yang dapat ditentukan dengan cara:

1. Tentukan jarak objek terhadap kamera (*a*), pada penelitian ini digunakan jarak 50cm.
2. Ukur cakupan frame video terhadap jarak yang telah ditentukan sebelumnya (*b*) masing-masing untuk lebar dan tinggi *frame*.
3. Tentukan korelasi antara *pixel* dengan jarak (cm). 320 pixel = *b* cm, sehingga

$$1cm = \frac{320}{b} \text{ pixel} \quad (16)$$

4. Tentukan korelasi antara satuan derajat dengan piksel, yaitu dengan mencari nilai keliling lingkaran dengan jarak yang telah ditentukan, Keliling Lingkaran = $2\pi \times a$, maka

$$1^{\circ} = \frac{2\pi a}{360} \text{ cm} \quad (17)$$

5. Tentukan korelasi antara satuan derajat terhadap *pixel*, berdasarkan persamaan pada *point* 3 dan 4.

$$1^{\circ} = \frac{\pi \cdot a}{180 \cdot b} \times 320 \text{ pixel} \quad (18)$$

➤ **Algoritma Penentuan Lebar Pulsa Pergerakan Motor Servo**

Motor servo standar dapat bergerak pada setiap sudutnya yaitu pada *range* 0° sampai 180° , dimana setiap sudut memiliki lebar pulsa yang berbeda-beda, sehingga untuk dapat menentukan lebar pulsa tersebut dapat ditentukan dengan:

1. Tentukan sudut acuan, yaitu pada sudut 0° dan 180° .
2. Tentukan lebar pulsa untuk 1 derajat, dengan menggunakan rumus:

$$T_{per1^0} = \frac{T_{S0^0} - T_{S180^0}}{180} \quad (19)$$

3. Tentukan lebar pulsa sudut lain dengan cara mengalikannya dengan lebar pulsa 1 derajat.

$$T_{sudut_x} = sudut_x \cdot T_{per1^0} \quad (20)$$

IV. HASIL DAN ANALISA

➤ Hasil Pengenalan Ucapan

Pada penelitian ini dilakukan pengujian terhadap sistem pengenalan ucapan dan pengujian terhadap kemampuan tracking objek. Pengujian *tracking object* dilakukan setelah ucapan dikenali. Pengujian terhadap sistem pengenalan ucapan, dilakukan terhadap sumber terlatih dan sumber yang tidak dilatih. Pengujian terhadap sumber terlatih bertujuan untuk mengetahui apakah sistem dapat mengenali kata yang diucapkan dari sumber yang sama dengan sumber pembentuk model kata tetapi dengan pengucapan yang berbeda dari pengucapan ketika pelatihan, seperti perbedaan intonasi, perbedaaan posisi mikrophone. Pengujian terhadap sumber yang tidak dilatih bertujuan untuk mengetahui kinerja sistem dalam mengenali kata meskipun diucapkan oleh sumber yang berbeda dengan sumber pembentuk model. Hasil pengujian ini dapat dilihat pada tabel 2.

Tabel 2. Hasil pengujian pengenalan ucapan

Kata yang diucapkan	Sumber terlatih	Sumber tidak terlatih
MERAH	93,33 %	100%
KUNING	66,67%	63%
HIJAU	80%	59%
BIRU	93,33%	99%
Rata-rata	83,33 %	80,25%

➤ Hasil Segmentasi

Sampel yang digunakan sebagai *frame* input segmentasi dapat dilihat pada gambar 9, dan hasil segmentasi setiap warna dapat dilihat pada tabel 3.



Gambar 9. Sampel Objek

Tabel 3. Hasil segmentasi untuk setiap warna

No	Warna dan Range Threshold	Gambar
1	Merah : $0 \leq H \leq 5$ $80 \leq I \leq 190$ $80 \leq S \leq 240$	
2	Kuning : $20 \leq H \leq 60$ $70 \leq I \leq 190$ $80 \leq S \leq 240$	
3	Hijau : $80 \leq H \leq 110$ $70 \leq I \leq 190$ $60 \leq S \leq 240$	
4	Biru : $120 \leq H \leq 160$ $80 \leq I \leq 190$ $120 \leq S \leq 240$	

➤ Proses tracking objek

Setelah ucapan warna berhasil dikenali kemudian warna yang dicari berhasil disegmentasi maka robot akan bergerak menuju warna tersebut dan robot berhenti sampai buzzer berbunyi. Proses tracking ini dapat dilihat pada tabel 4.

Tabel 4. Proses hasil tracking objek dan respon robot mobil berdasarkan kata yang dikenali (untuk kasus warna biru)

Posisi	Deteksi Objek	Gerakkan Kamera dan respon robot mobil	Keterangan
1			Hasil rekaman suara, dibaca., Dikenali sebagai biru, Lakukan tracking objek biru Awal diaktifkan, robot mobil mencari sudut 90°.
2			Deteksi warna biru dan penguncian objek (kotak persegi warna merah), Servo bergerak ke sebelah kanan (>90°).
3			Cari titik tengah objek, Putar motor DC disini, sampai motor servo kamera berada pada posisi 90°.
4			Titik tengah objek dipertahankan (digeser) sampai sama dengan titik tengah frame. Robot mobil dirancang mampu mengontrol informasi koordinat x, Putar motor DC disini untuk proses pengaturan sampai titik tengah objek berada pada titik tengah frame
5			Jumlah piksel >10000 Motor DC berhenti dan buzzer berbunyi.

- Hasil tracking objek sesuai kata yang dikenali
Setelah robot mampu melakukan tracking terhadap warna yang dicari, maka dilakukan pengujian terhadap semua ucapan yang diujikan dan semua warna. Kemampuan robot mobil untuk mencari semua warna yang sesuai dengan perintah ucapan yang diberikan dapat dilihat pada tabel 5

Tabel 5. Tracking objek berdasarkan perintah warna yang dikenali

No	Perintah kata dan warna Objek	Proses tracking robot mobil
1	Merah  $0 \leq H \leq 5, 80 \leq I \leq 190$ dan $80 \leq S \leq 240$	
2	Kuning  $20 \leq H \leq 60, 70 \leq I \leq 190$ dan $80 \leq S \leq 240$	
3	Hijau  $80 \leq H \leq 110, 70 \leq I \leq 190$ dan $60 \leq S \leq 240$	
4	Biru  $120 \leq H \leq 160, 80 \leq I \leq 190$ dan $120 \leq S \leq 240$	

Berdasarkan tabel 5, robot mobil akan men-tracking dan memberi respon maju terhadap objek ketika perintah ucapan telah dikenali dan ketika jumlah piksel objek lebih besar 2000 piksel dan kurang dari 10000. Ketika robot mobil telah selesai merespon maju (mendekati objek) sesuai dengan batas yang telah diberikan maka buzzer akan berbunyi.

Jika sistem pengenalan ucapan salah dalam mengenali kata yang diucapkan maka robot mobil akan tetap mencari warna yang dikenali, sebagai contoh jika diucapkan warna BIRU dan sistem pengenalan mengenali sebagai MERAH maka robot mobil akan mentracking objek berwarna MERAH. Berdasarkan percobaan, didapatkan bahwa

robot mobil selalu berhasil mentracking objek berdasarkan warna yang dikenali. Hal ini menunjukkan kemampuan tracking objek berdasarkan identifikasi warna sebesar 100 % .

Hal yang sangat menentukan kinerja sistem adalah akurasi dari sistem pengenalan ucapan. Berdasarkan tabel 2 didapatkan akurasi sistem pengenalan untuk sumber terlatih adalah 83,33% (kesalahan 16,67 %) dan sumber tidak terlatih adalah 80,25 % (kesalahan 19,75 %). Kesalahan pengenalan disebabkan sistem pengenalan ucapan dibentuk menggunakan metoda HMM dengan pemodelan bersifat *single observation*, sehingga model yang terbentuk belum maksimal dalam mewakili ucapan kata MERAH, KUNING, HIJAU dan BIRU untuk semua sumber dengan variasi pengucapan kata.

V. PENUTUP

5.1 Kesimpulan

1. Hasil Pengenalan ucapan sumber terlatih sebesar 83,33% dan untuk sumber tidak dilatih sebesar 80,25% .
2. Keberhasilan robot mobil mengejar atau memilih objek berdasarkan identifikasi warna sesuai dengan kata yang dikenali sebesar 100%.
3. Media *tracking* berupa motor servo dc dan motor dc telah mampu melakukan *tracking* objek dengan menggerakkan kamera sehingga kamera mampu membaca posisi objek, dan respon maju yang dilakukan membuktikan motor DC mampu mempertahankan titik tengah objek untuk sama dengan titik tengah frame.

5.2 saran

Pada penelitian selanjutnya sebaiknya menggunakan model ucapan yang dibentuk dari multiple observation. Penelitian selanjutnya sebaiknya real time dan Dalam melakukan deteksi terhadap objek, pergerakan kamera tidak hanya sebatas horizontal saja tapi juga mampu bergerak secara vertikal.

DAFTAR PUSTAKA

- [1] Fitrilina, Sistem Pengenalan Isolated Digit yang Robust Dengan Menggunakan Spectral Subtraction Berdasarkan

Minimum Statistics, Teknika No 35 Vol 1. THNXVII April 2011

- [2] Rabiner, "A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol 77, No 2, 1989
- [3] Tomi Aarnio, "Speech Recognition with Hidden Markov Models in Visual Communication," Master of Science Thesis, University of Turki, 1999
- [4] Ting Chee Ming, " Malay Continous Speech Recognition using Continous Density Hidden Markov Model", Thesis, Universiti Teknologi Malaysia, 2007.
- [5] Mikael Nilson, marcus Ejnaronson, "Speech recognition using Hidden Markov Model, Performance Evaluation in Noisy Environment", Theiss Blekinge institute of Technology, Sewen, 2002.
- [6] Yao, Wang, J.Ostermann & Y.Q.Zhang. 2002. "Video Processing and Communications". New Jersey: Prentice Hall.
- [7] Microsoft Corporation.2001. "Microsoft Paint Software". New York: Microsoft Corporation.
- [8] Hearn, Donald dan M. Pauline Baker. 1996. "Computer Graphics C Version 2nd Edition". New York: Prentice Hall.

Biodata Penulis

Fitrilina, Lahir di Padang tanggal 25 Agustus 1981, Menamatkan S1 di Teknik Elektro Fakultas Teknik UNAND 2005, Menamatkan S2 di Teknik Elektro STEI-ITB 2010. Saat ini terdaftar sebagai dosen Teknik Elektro FT-UNAND

Rahmadi Kurnia, Lahir tanggal 20 Agustus 1969. Menamatkan S1 di Teknik Elektro Universitas Indonesia tahun 1995. Menamatkan S2 di Teknik Elektro Universitas Indonesia tahun 1998, bidang Image Processing. Menamatkan S3 di Saitama University, Jepang 2006. Saat ini terdaftar sebagai dosen Teknik Elektro FT-UNAND

Siska Aulia, Lahir di Payakumbuh tanggal 4 Maret 1988. Menamatkan S1 di Teknik Elektro Universitas Andalas tahun 2011. Saat ini terdaftar sebagai mahasiswa Pasca Sarjana di Teknik Elektro Universitas Andalas