Identifikasi Pembicara Menggunakan Algoritme VFI5 dengan MFCC sebagai Pengekstraksi Ciri

Vicky Zilvan, S.Kom.
UPT LPSN - LIPI
v_q1e@yahoo.com

Furqon Hensan Muttaqien, S.Komp.
P2 Informatika - LIPI
fh.muttaqien@informatika.lipi.go.id

Abstract

Voting Feature Intervals (VFI) 5 algorithm has high accuracy rate in classification for text and image data. According to that fact, we have developed a method for speaker identification using VFI5 algorithm with Mel Frequency Cepstrum Coefficients (MFCC) as voice feature extraction. In this research, we also have tried to examine the method using noise. The result is this method which has been developed has high accuracy with highest accuracy about 97% for normal data (without noise). In addition, the results of this research also indicate that 11 is the optimum number of training data which obtain the highest accuracy. Whereas for noise with SNR about 30 dB, highest accuracy is 81.5 % and for noise with SNR about 20 dB about 59%.

Keywords: *Voting Feature Intervals 5 (VFI5), Mel Frequency Cepstrum Coefficients (MFCC), speaker identification*

Abstrak

Voting Feature Intervals (VFI) 5 memiliki akurasi yang cukup tinggi dalam mengklasifikasikan data berbasis teks dan citra. Berdasarkan hal tersebut dikembangkanlah metode identifikasi pembicara menggunakan algoritme VFI5 dengan Mel Frequency Cepstrum Coefficients (MFCC) sebagai pengekstraksi ciri suara untuk melihat keakuratan algoritme VFI5 dalam mengklasifikasikan data berbasis suara. Jenis identifikasi pembicara pada penelitian ini bersifat tertutup dan bergantung pada text. Pada penelitian ini juga dilakukan percobaan menggunakan suara ber-noise untuk melihat kehandalan VFI5 dalam mengklasifikasikan suara ber-noise. Dari hasil pengujian didapatkan bahwa metode yang telah dikembangkan ini memiliki akurasi cukup tinggi dengan akurasi tertinggi sebesar 97% untuk data suara tanpa noise. Selain itu juga diketahui bahwa jumlah data latih yang optimal untuk menghasilkan akurasi yang tinggi adalah 11. Sedangkan untuk suara bernoise dengan SNR sebesar 30 dB, akurasi tertinggi mencapai 81,5 % dan untuk suara bernoise dengan SNR sebesar 20 dB tingkat akurasi tertinggi mencapai 59 %.

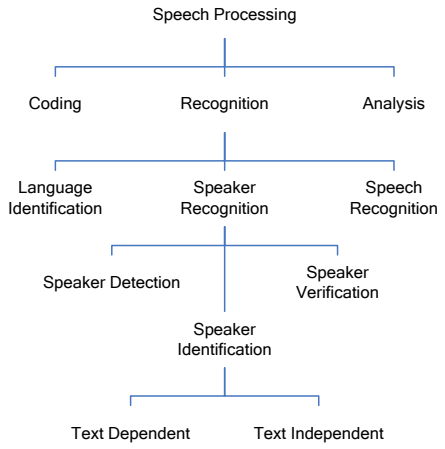
Kata kunci: *Voting Feature Intervals 5 (VFI5), Mel Frequency Cepstrum Coefficients (MFCC), identifikasi pembicara*

1. Pendahuluan

Biometrik merupakan ilmu yang mempelajari pengenalan identitas manusia berdasarkan pola ciri-ciri yang dimilikinya, baik ciri-ciri kimia, fisik, maupun tingkah laku, seperti wajah, sidik jari, suara, geometri tangan, ataupun iris mata [1].

Penelitian yang dilakukan dengan menggunakan data sinyal suara umumnya disebut dengan pemrosesan sinyal suara (*speech processing*). *Speech processing* sendiri memiliki beberapa cabang kajian sebagaimana terlihat pada Gambar 1. Salah

satu kajian dalam *speech processing* adalah identifikasi pembicara. Identifikasi pembicara (*speaker identification*) adalah suatu proses mengenali seseorang berdasarkan suaranya [2].



Gambar 1 Pemrosesan sinyal suara

Pengenalan pembicara berdasarkan aspek kebahasaan dibagi menjadi dua, yaitu pengenalan pembicara bergantung teks dan pengenalan pembicara bebas [2]. Pada pengenalan pembicara bergantung teks pembicara diharuskan untuk mengucapkan kata atau kalimat yang sama baik pada pelatihan maupun pengujian. Sedangkan pada pengenalan pembicara bebas teks pembicara tidak diharuskan untuk mengucapkan kata atau kalimat yang sama baik pada pelatihan maupun pengujian.

Penggunaan suara pada sistem biometrik saat ini telah diterima secara luas. Selain murah, penggunaan suara juga dinilai lebih praktis dibandingkan objek biometrik yang lain [2]. Walaupun begitu, penelitian pemrosesan sinyal suara masih terus dilakukan. Ini terkait dengan masih adanya beberapa tantangan dalam pengenalan suara, seperti kesalahan membaca atau pengucapan frasa yang telah ditetapkan, keadaan emosi pembicara, posisi mikrofon saat pengucapan, inkonsistensi ruang akustik (misalnya adanya noise), kesalahan *channel* transmisi (misalnya menggunakan mikrofon yang berbeda saat identifikasi dan verifikasi), penyakit pembicara yang dapat mempengaruhi *vocal tract*-nya (misalnya flu), usia pembicara dan peniruan suara (mimicry) [2].

Beberapa metode yang dikembangkan para peneliti untuk melakukan identifikasi suara antara lain *Dynamic Time Warping* (DTW), Model Markov Tersembunyi, *Vector Quantization* (VQ), *Bayesian classifiers, Principal Components Analysis* (PCA), algoritma *K-Means clustering*, jaringan syaraf tiruan maupun logika *Fuzzy*. Pada penelitian ini akan dicoba metode lain untuk melakukan identifikasi pembicara, yaitu Voting Feature Intervals (VFI) 5, dengan MFCC sebagai pengektraksi ciri suara.

Penelitian dengan menggunakan algoritme VFI5 sebelumnya telah banyak digunakan untuk klasifikasi berbasis teks dan juga citra. Pada penelitian diagnosis penyakit demam berdarah dengue algoritme VFI5 memiliki akurasi sebesar 100%, sedangkan saat menggunakan ANFIS akurasi yang diperoleh adalah sebesar 86,67% [3]. Demikian pula akurasi yang dihasilkan pada klasifikasi citra juga cukup menjanjikan, yaitu 97,5% pada pengenalan tanda tangan menggunanakan dengan praproses wavelet level 1 dan 100% pada pengenalan wajah dengan partisi berbasis histogram [3], [4].

Mel-Frequency Cepstrum Coefficients (MFCC) merupakan salah satu teknik ekstraksi ciri berbasis transformasi Fourier yang telah banyak dipakai dalam pemrosesan sinyal suara, termasuk pengenalan pembicara [5]. Penggunaan teknik ini pada sistem pemrosesan sinyal suara memberikan pengenalan yang lebih baik dibandingkan dengan metode lain yang sudah ada [6].

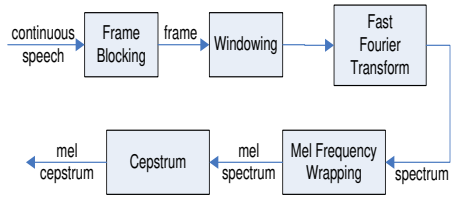
Dari beberapa hal yang telah dipaparkan, penelitian ini bertujuan untuk mengembangkan model pengidentifikasi pembicara bersifat *text-dependent* menggunakan algoritme VFI5. Dari penelitian ini juga diharapkan dapat diketahui jumlah data latih terkecil yang digunakan untuk menghasilkan akurasi yang optimal.

2. Tinjauan Pustaka

2.1 MFCC

MFCC didasarkan pada variasi yang telah diketahui dari jangkauan kritis telinga manusia dengan frekuensi. Filter dipisahkan secara linear pada frekuensi rendah dan logaritmik pada frekuensi tinggi. Hal ini telah dilakukan untuk menangkap karakteristik penting dari sinyal suara.

Tujuan utama MFCC adalah untuk meniru perilaku telinga manusia. Selain itu MFCC telah terbukti bisa menyebutkan variasi dari gelombang suara [7]. Diagram blok dari proses MFCC dapat dilihat pada Gambar 2.



Gambar 2 Proses MFCC [7]

Penjelasan tiap tahapan pada proses MFCC sebagai berikut [7]:

a. *Frame Blocking*. Pada tahap ini sinyal suara (*continous speech*) dibagi ke dalam *frame-frame*. Tiap *frame* terdiri dari N sampel.

b. *Windowing*. Proses selanjutnya adalah melakukan *windowing* pada tiap *frame* untuk meminimalkan diskontinuitas sinyal pada awal dan akhir tiap *frame*. Konsepnya adalah meminimisasi distorsi spektral dengan menggunakan *window* untuk memperkecil sinyal hingga mendekati nol pada awal dan akhir tiap *frame*. Jika *window* didefinisikan sebagai w(n), $0 \le n \le N-1$, dengan N adalah banyaknya sampel tiap *frame*, maka hasil dari *windowing* adalah sinyal dengan persamaan (1):

$$Yt(n)=x1(n)w(n). \quad (1)$$

c. Pada umumnya, *window* yang digunakan adalah *hamming window*, dengan persamaan (2):

$$w(n)=0.54-0.46\cos(2\pi n/N-1). \quad (2)$$

d. *Fast Fourier Transform* (FFT). Tahap ini mengkonversi tiap *frame* dengan N sampel dari *time domain* menjadi *frequency domain*. FFT adalah suatu algoritme untuk mengimplementasikan *Discrete Fourier Transform* (DFT) yang didefinisikan pada himpunan N sampel {xn} sebagai persamaan (3):

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N} \quad (3)$$

e. j digunakan untuk menotasikan unit imajiner, yaitu $j = \sqrt{-1}$. Secara umum Xn adalah bilangan kompleks. Barisan {Xn} yang dihasilkan diartikan sebagai berikut: frekuensi nol berkorespondensi dengan n = 0, frekuensi positif $0 < f < Fs/2$ berkorespondensi dengan nilai $1 \le n \le N/2-1$, sedangkan frekuensi negatif $-Fs/2 < f < 0$ berkorespondensi dengan $N/2+1 < n < N-1$. Dalam hal ini Fs adalah *sampling frequency*. Hasil yang didapatkan dalam tahap ini biasa disebut dengan spektrum sinyal atau periodogram.

f. *Mel-frequency Wrapping*. Studi psikofisik menunjukkan bahwa persepsi manusia terhadap frekuensi sinyal suara tidak berupa skala linear. Oleh karena itu, untuk setiap nada dengan frekuensi aktual f (dalam Hertz), tinggi subjektifnya diukur dengan skala 'mel'. Skala *mel-frequency* adalah selang frekuensi di bawah 1000 Hz dan selang logaritmik untuk frekuensi di atas 1000 Hz, sehingga pendekatan persamaan (4) dapat digunakan untuk menghitung *mel-frequency* untuk frekuensi f dalam Hz:

$$Mel(f) = 2595*\log10(1+f/700). \quad (4)$$

Cepstrum. Langkah terakhir, konversikan log *mel spectrum* ke domain waktu. Hasilnya disebut *mel frequency cepstrum coefficients*. Representasi cepstral spektrum suara merupakan representasi properti spektral lokal yang baik dari suatu sinyal untuk

analisis *frame*. *Mel spectrum coefficients* (dan logaritmanya) berupa bilangan riil, sehingga dapat dikonversikan ke domain waktu dengan menggunakan *Discrete Cosine Transform* (DCT).

2.2 Voting Feature Intervals 5

Voting Feature Intervals adalah salah satu algoritme yang digunakan dalam pengklasifikasian data. Algoritme tersebut dikembangkan oleh Gülşen Demiroz dan H. Altay Güvenir pada tahun 1997 [8]. Algoritme klasifikasi VFI5 merepresentasikan deskripsi sebuah konsep oleh sekumpulan interval nilai-nilai *feature* atau atribut. Pengklasifikasian *instance* baru berdasarkan *voting* pada klasifikasi yang dibuat oleh nilai tiap-tiap *feature* secara terpisah. VFI5 merupakan algoritme klasifikasi yang bersifat *non-incremental* dan *supervised* [8]. Algortima VFI5 membuat interval yang berupa *range* atau *point interval* untuk setiap *feature*. *Point interval* terdiri atas seluruh *end point* secara berturut-turut. *Range interval* terdiri atas nilai-nilai antara 2 *end point* yang berdekatan namun tidak termasuk kedua *end point* tersebut.

Keunggulan algoritma VFI5 adalah algoritma ini cukup kokoh (*robust*) terhadap *feature* yang tidak relevan namun mampu memberikan hasil yang baik pada *real-world datasets* yang ada. VFI5 mampu menghilangkan pengaruh yang kurang menguntungkan dari *feature* yang tidak relevan dengan mekanisme *voting*-nya [9]. Berikut adalah *Pseudocode* pelatihan pada algoritme VFI5.

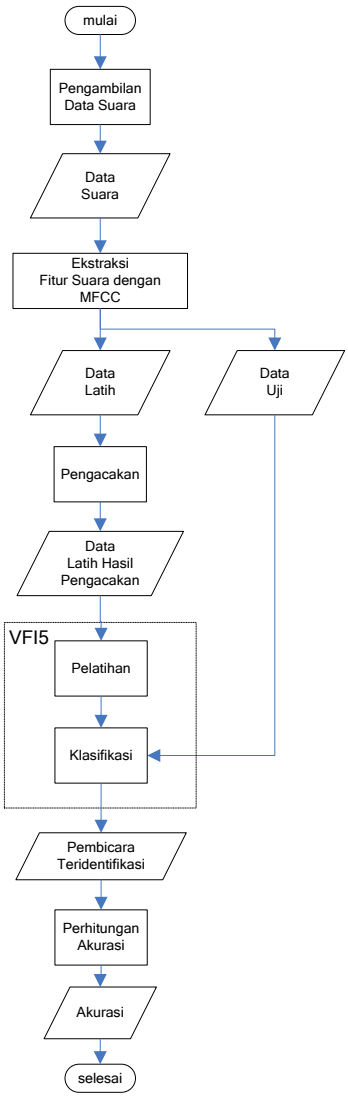
```
train(Training Set);
begin
   for each feature f
         for each class c
               EndPoints[f] =
EndPoints[f] U
find_end_points(TrainingSet, f, c);
         sort(EndPoints[f]);
         If f is linear
               for each end point p
in EndPoints[f]
                     form a poin
interval from end point p
                     form a range
interval between p and the next
endpoint ≠ p
         else /*f is nominal*/
               each distinct point in
EndPoints[f] forms a point interval
         for each interval I on
feature dimension f
               for each class c

interval_count[f, I, c] = 0
         count_instances(f,
TrainingSet);
         for each interval I on
feature dimension f
               for each class c
                     interval_vote[f,
I, c] = interval_count[f, I,
c]/class_count[c]
                     normalize
interval_vote[f, i, c]
                     /*such that Σc
interval_vote[f, I, c] = 1*/
   end
```

Sedangkan Pseudocode klasifikasi algoritma VFI5 adalah sebagai berikut:

```
classify(e); /*e:example to be
classified*/
begin
   for each class c
         vote[c] = 0
   for each feature f
         for each class c
               feature_vote[f, c] = 0
/*vote of feature f for class c*/
         if ef value is known
               i=find_interval(f, ef)
               for each class c
                     feature_vote[f,
c] = interval_vote[f, I, c]
                     vote[c] =
vote[c] + feature_vote[f, c] *
weight[f];
   return the class c with highest
vote[c];
end
```

3. Metode Penelitian

Ada beberapa tahap yang dilakukan dalam proses identifikasi menggunakan model yang akan dikembangkan ini. Tahapan ini ditunjukkan pada Gambar 5.



Gambar 5 Proses identifikasi pembicara

3.1 Pengambilan data suara

Data pada penelitian ini merupakan data yang sama yang digunakan pada [10]. Data tersebut merupakan suara dari 10 pembicara yang telah didijitasi yang terdiri dari 5 pembicara laki-laki dan 5 pembicara perempuan dengan rentang usia 20-25 tahun. Masing-masing pembicara diambil suaranya dalam jangka waktu yang sama dan tanpa pengarahan (*unguided*) sehingga pembicara dapat menggunakan cara pengucapan, intonasi, dan logat apapun pada saat perekaman.

Karena jenis identifikasi pembicara yang dilakukan bersifat bergantung pada teks, maka kata yang digunakan telah ditentukan terlebih dahulu, yaitu kata "komputer". Pengambilan data ini dilakukan sebanyak 60 kali untuk masing-masing pembicara sehingga terdapat 600 berkas data. Dari 60 data tersebut, 40 data pengambilan pertama digunakan sebagai data latih dan sisanya sebanyak 20 data digunakan sebagai data uji.

Untuk mengetahui kehandalan model yang dikembangkan terhadap *noise* diperlukan juga data ber-*noise* dengan jumlah yang sama. Data ber-*noise* tersebut diperoleh dengan menambahkannya secara manual menggunakan fungsi awgn pada Matlab sesuai dengan besarnya SNR. Besarnya SNR yang akan ditambahkan adalah 20 dB dan 30 dB. Dengan demikian, total seluruh data suara yang didapat sebanyak 1800 *file*, yaitu 600 *file* data suara asli dan 600 *file* data suara yang dengan SNR sebesar 20 dB dan 600 *file* data suara yang dengan SNR sebesar 30 dB.

3.2 Ekstraksi suara dengan MFCC

Ekstraksi ciri sinyal suara pada penelitian ini menggunakan MFCC. Pada implementasi MFCC ini, kecuali tahap *frame blocking*, digunakan fungsi dari *Auditory Toolbox* yang dikembangkan oleh Slanley pada tahun 1998. Fungsi ini menggunakan lima parameter, yaitu:

a. *Input*, yaitu masukan suara yang berasal dari tiap pembicara.
b. *Sampling rate*, yaitu banyaknya nilai yang diambil dalam satu detik. Dalam penelitian ini digunakan *sampling rate* sebesar 16000 Hz.
c. Time *frame*, yaitu waktu yang diinginkan untuk satu *frame* (dalam milidetik). *Time frame* yang digunakan adalah 30 ms.
d. *Lap*, yaitu *overlaping* yang diinginkan (harus kurang dari satu). *Lap* yang digunakan sebesar 0.5.
e. *Cepstral coefficient*, yaitu jumlah *cepstrum* yang diinginkan sebagai

output. *Cepstral coefficient* yang digunakan sebanyak 13.

Setiap data suara dari setiap pembicarakan dibagi menjadi 66 *frame* dimana masing-masing *frame* berukuran 30 ms dengan *overlap* 50%. Hasil dari ekstraksi ciri ini merupakan masukan bagi model yang akan dikembangkan, yaitu VFI 5. 3.3 Pengacakan

Proses pengacakan ini dilakukan untuk mengurangi pengaruh data latih yang digunakan pada hasil percobaan. Dengan demikian diharapkan semua pola suara dari pembicara dapat tercakup dalam data latih yang digunakan. Proses pengacakan ini diulang sebanyak 3 kali untuk tiap percobaan dengan jumlah data latih yang sama. Adapun jumlah data latih yang digunakan pada penelitian ini dimulai dari 39 hingga 1 data latih.

3.4 VFI 5

Terdapat dua tahapan dalam algoritme VFI 5. Kedua tahapan tersebut adalah pelatihan dan klasifikasi.

3.4.1 Pelatihan

Fitur yang digunakan untuk pelatihan pada algoritme VFI 5 merupakan elemen matriks hasil ektraksi fitur menggunakan MFCC pada tahap sebelumnya. Adapun matriks yang dihasilkan berukuran 1 x 858, sehingga jumlah fitur yang digunakan berjumlah 858 fitur.

3.4.2 Klasifikasi

Setiap nilai fitur dari *instance* data uji diperiksa letak interval nilai fitur tersebut pada hasil vote yang telah dinormalisasi. *Vote-vote* setiap kelas untuk setiap fitur pada setiap interval yang bersesuaian diambil dan kemudian dijumlahkan. Kelas yang memiliki nilai total *vote* tertinggi menjadi kelas prediksi *instance* tersebut.

3.5 Perhitungan Akurasi

Hasil yang diamati pada penelitian ini adalah tingkat akurasi algoritme VFI5 dalam mengklasifikasikan data pengujian. Tingkat akurasi diperoleh dengan persamaan (5).

$$\text{akurasi} = \frac{\sum \text{data uji benar diklasifikasi}}{\sum \text{data uji}} \times 100\% \quad (5)$$

4. Hasil dan Pembahasan

4.1 Praproses dengan MFCC

Implementasi MFCC ini, kecuali tahap *frame blocking*, menggunakan fungsi dari *Auditory Toolbox* yang dikembangkan oleh Slanley pada tahun 1998. Setiap data suara akan dibagi menjadi *frame* berukuran masing-masing 30 ms dengan *overlap* 50%, dengan demikian akan dihasilkan 66 *frame*. Hasil dari analisis fitur suara MFCC ini adalah 13 koefisien *mel cepstrum* untuk masing-masing *frame*. Pemilihan nilai *time frame, lap,* dan *cepstral coefficient* berturut-turut sebesar 30 ms, 0.5, dan 13 didasarkan pada penelitian sebelumnya yang dilakukan [11] dan [12].

Seluruh data suara yang telah didapat (1800 *file*), baik data suara asli maupun data suara yang telah diberikan *noise* dengan SNR sebesar 20 dB dan *noise* dengan SNR sebesar 30 dB, selanjutnya dilakukan praproses MFCC. Dari hasil praproses ini, maka setiap data berubah dari matriks yang berukuran 16000 x 1 menjadi matriks 13 x 66. Selanjutnya, matrik berukuran 13 x 66 tersebut dijadikan 1 baris, sehingga didapat matriks berukuran 1 x 858. Matriks ini lah yang kemudian akan digunakan sebagai masuan untuk algoritme VFI5.

4.2 Penentuan Data Latih secara Acak beserta Pengulangannya

Pada model ini, data latih digunakan untuk membangun pola pada proses identifikasi. Banyaknya data latih akan dicobakan sebanyak 1-39 data dari masing-masing pembicara. Sehingga, akan terdapat 39 perlakuan untuk setiap jenis data suara (data suara asli, data suara ber-*noise* dengan SNR sebesar 20 dB, maupun data suara ber-*noise* dengan SNR sebesar 30 dB), terhadap model yang telah dikembangkan.

Perlakuan 1 merupakan perlakuan kepada model dengan banyaknya data latih

per pembicara sebanyak 1 data, perlakuan 2 merupakan perlakuan kepada model dengan banyaknya data latih per pembicara sebanyak 2 data, dan seterusnya. Semua jenis perlakuan yang akan dicobakan ditunjukkan pada Tabel 1.

Untuk mencegah suatu kebetulan/eksklusifitas penentuan data latih, serta untuk mendapatkan kesimpulan yang umum mengenai akurasi model yang dikembangkan, maka data yang dipilih sebagai data latih pun dilakukan secara acak pada masing-masing jenis data (data suara asli, data suara ber-*noise* dengan SNR sebesar 20 dB, maupun data suara ber-*noise* dengan SNR sebesar 30 dB), serta setiap perlakuan akan diulang sebanyak 3 kali.

Proses pengacakan ini menggunakan fungsi random untuk membangkitkan nilai random yang banyaknya sesuai dengan data latih yang digunakan, serta tidak terdapat pengulangan data yang digunakan pada jenis perlakuan yang sama.

Tabel 1 Jenis perlakuan

Perlakuan	Jumlah data latih per pembicara	Jumlah data latih seluruhnya
1	1	10
2	2	20
3	3	30
4	4	40
5	5	50
6	6	60
7	7	70
8	8	80
9	9	90
10	10	100
11	11	110
12	12	120
13	13	130
14	14	140
15	15	150
16	16	160
17	17	170
18	18	180
19	19	190
20	20	200
21	21	210
22	22	220
23	23	230
24	24	240
25	25	250
26	26	260
27	27	270
28	28	280
29	29	290
30	30	300
31	31	310
32	32	320
33	33	330
34	34	340
35	35	350
36	36	360
37	37	370
38	38	380
39	39	390

4.3 Voting Feature Intervals 5

4.3.1 Pelatihan

Dari proses pelatihan ini, untuk setiap perlakuan akan terbentuk suatu *vote* interval yang telah dinormalisasi. Dengan demikian akan terbentuk 39 *vote* interval ternormalisai dari seluruh perlakuan pada setiap jenis data (data suara asli, data suara ber-*noise* dengan SNR sebesar 20 dB, maupun data suara ber-*noise* dengan SNR sebesar 30 dB). Selanjutnya *vote* interval ini akan digunakan pada proses klasifikasi.

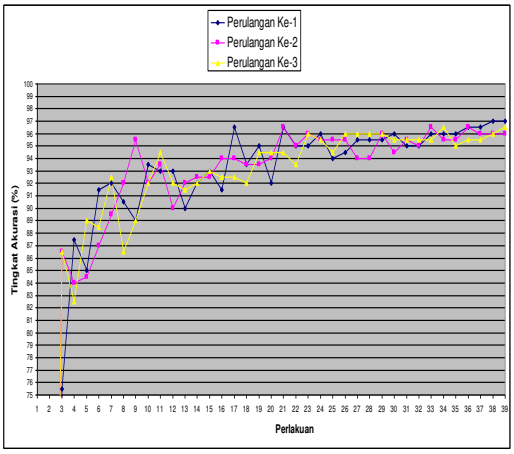
4.3.2 Klasifikasi

Pada proses klasifikasi, akan dilakukan klasifikasi dari setiap data tes yang diuji dengan menggunkan *vote* interval yang telah terbentuk dari proses pelatihan.

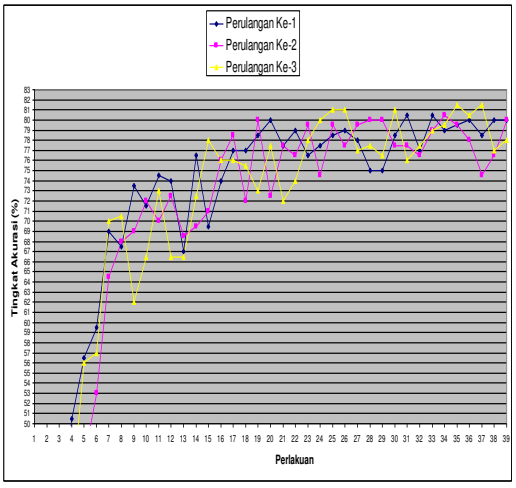
Hasil klasifikasi dari seluruh data tes ini selanjutnya akan dilakukan perhitungan untuk mengetahui akurasi model dari setiap perlakuan. Hasil perhitungan akurasi untuk setiap perulangan pada setiap jenis data disajikan pada Gambar 6, 7, dan 8.

Berdasarkan hasil yang didapat yang terlihat pada Gambar 6, 7, dan 8 terlihat bahwa semakin banyak data latih, semakin

meningkat rata-rata akurasi identifikasi model.

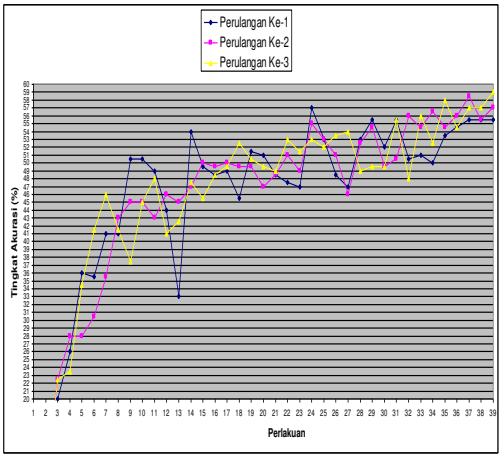


Gambar 6 Tingkat Akurasi Model untuk Data Suara Asli

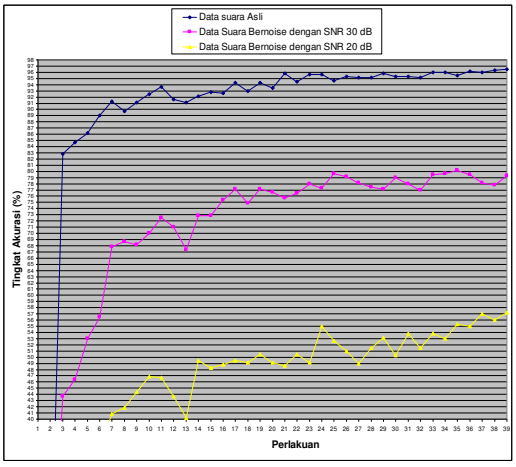


Gambar 7 Tingkat Akurasi Model untuk Data Suara dengan SNR 30 dB

Di sisi lain, dilihat dari perulangan yang dilakukan, terlihat bahwa terdapat perbedaaan tingkat akurasi yang dihasilkan antara perulangan ke-1, perulangan ke-2, dan perulangan ke-3 pada perlakuan yang sama. Hal ini dipengaruhi oleh tidak seragamnya data latih yang dihasikan pada saat perekaman suara akibat perbedaan cara pengucapan.



Gambar 8 Tingkat Akurasi Model untuk Data Suara dengan SNR 30 dB



Gambar 9 Tingkat Akurasi Model untuk setiap jenis data suara

Dari Gambar 9 terlihat bahwa besarnya *noise* yang diberikan memberikan pengaruh yang cukup signifikan terhadap akurasi model yang dihasilkan. Berdasarkan studi [5], penurunan ini sangat mungkin disebabkan penggunaan MFCC sebagai pengektraksi ciri. Dalam studi tersebut dijelaskan bahwa masukan yang digunakan pada proses MFCC, yaitu spektrum energi yang diperoleh dari transformasi Fourier, memiliki sensitivitas yang cukup tinggi terhadap gangguan *noise*. Dari penelitian tersebut diketahui juga bahwa pada data dengan *noise* 20 dB, akurasi sistem akan turun menjadi sekitar 40% dari semula 99%

dibandingkan data tanpa penambahan *noise*. Dengan demikian, model ini belum cukup handal untuk mengidentifikasi suara ber-*noise* cukup tinggi. Penggunaan metode pengekstraksi ciri yang *robust* terhadap *noise* diharapkan akan mampu meningkatkan nilai akurasi dari model identifikasi ini.

4.4 Uji Statistika

4.4.1 Analisis Ragam (ANOVA)

Untuk membuat suatu kesimpulan berdasarkan data yang diperoleh mengenai tingkat akurasi model dari setiap perlakuan untuk setiap jenis data (data suara asli, data suara ber-*noise* dengan SNR sebesar 20 dB, maupun data suara ber-*noise* dengan SNR sebesar 30 dB), maka akan dilakukan uji ANOVA untuk melihat pengaruh perlakuan terhadap respon (pengaruh banyaknya data latih terhadap tingkat akurasi model) dengan menggunakan Minitab 15. Hipotesis yang akan diuji adalah sebagai berikut.

H_0 : Banyaknya data latih (perlakuan) tidak berpengaruh terhadap tingkat akurasi model (respon)

H_1 : Banyaknya data latih (perlakuan) berpengaruh terhadap tingkat akurasi model (respon)

Tabel 2 Hasil uji ANOVA

Jenis data suara	P-value
Data suara asli	0.000
Data suara bernoise dengan SNR sebesar 30 dB	0.000
Data suara bernoise dengan SNR sebesar 20 dB	0.000

Dari uji ANOVA sebagai mana terlihat dari Tabel 2 diperoleh bahwa P-value < α sebesar 5 % untuk setiap jenis data sehingga diputuskan untuk menolak H_0. Dari keputusan yang diambil tersebut, dapat disimpulkan bahwa secara statistik perbedaan jumlah data latih yang digunakan pada model ini akan menghasilkan nilai akurasi yang berbeda nyata pula.

4.4.2 Uji Lanjut Duncan

Dari hasil analisis ragam (ANOVA) untuk masing-masing jenis data terlihat bahwa terdapat pengaruh nyata (P-value < α sebesar 5 %), maka dapat dilakukan uji lanjut. Pada penelitian ini untuk uji lanjut digunakan Uji Lanjut Duncan menggunakan SPSS 16.0. Uji Lanjut Duncan dipilih karena uji lanjut Duncan lebih teliti dan bisa digunakan untuk membandingkan pengaruh perlakuan dengan jumlah perlakuan yang besar.

Dari hasil uji lanjut Duncan untuk data suara asli diketahui bahwa perlakuan yang memberikan respon optimal adalah pelakuan ke 11, yaitu perlakuan dengan banyak data latih sebanyak 11 data dari setiap pembicara. Hasil klasifikasi dengan menggunakan data latih yang optimum pada jenis suara asli ditunjukan pada Tabel 3.

Tabel 3 Hasil Klasfikasi dengan jumlah data latih optimum pada data suara asli

Pembicara	Hasil Klasifikasi		Akurasi (%)
	Benar	Salah	
1	18	2	90
2	18	2	90
3	20	0	100
4	19	1	95
5	17	3	85
6	19	1	95
7	20	0	100
8	19	1	95
9	17	3	85
10	19	1	95

Untuk data ber-*noise*, hasil uji Duncan untuk SNR sebesar 30 dB menunjukan bahwa perlakuan yang memberikan respon optimal adalah perlakuan 16, yaitu perlakuan dengan banyak data latih sebayak 16 dari setiap pembicara. Hasil klasifikasi dengan menggunakan data latih yang optimum pada jenis suara bernoise dengan SNR sebesar 30dB ditunjukan pada Tabel 4.

Tabel 4 Hasil klasfikasi dengan jumlah data latih optimum pada data suara bernoise dengan SNR sebesar 30 dB

Pembicara	Hasil Klasifikasi		Akurasi (%)
	Benar	Salah	
1	8	12	40
2	14	6	70
3	14	6	70
4	15	5	75
5	17	3	85
6	19	1	95
7	15	5	75
8	15	5	75
9	17	3	85
10	18	2	90

Untuk data suara ber-*noise* dangan SNR sebesar 20 dB, perlakuan yang memberikan respon optimal adalah perlakuan 24, yaitu perlakuan dengan banyak data latih sebayak 24 dari setiap pembicara. Hasil klasifikasi dengan menggunakan data latih yang optimum pada jenis suara bernoise dengan SNR sebesar 20dB ditunjukan pada Tabel 5.

Tabel 5 Hasil klasfikasi dengan jumlah data latih optimum pada data suara bernoise dengan SNR sebesar 30 dB

Pembicara	Hasil Klasifikasi		Akurasi (%)
	Benar	Salah	
1	7	13	35
2	8	12	40
3	11	9	55
4	9	11	45
5	4	16	20
6	18	2	90
7	11	9	55
8	7	13	35
9	15	5	75
10	16	4	80

4.5 Penelitian sebelumnya

Perbandingan akurasi penelitian ini dengan penelitian sebelumnya disajikan pada Tabel 6. Dari Tabel 6 terlihat bahwa akurasi optimal yang dihasilkan dari model yang dikembangkan pada penelitian ini tidak berbeda jauh dari akurasi pada model yang telah lebih dahulu dikembangkan. Jika dilihat dari perbandingan jumlah data latih dan data uji dengan akurasi yang dihasilkan, secara umum model VFI5-MFCC menghasilkan akurasi yang lebih optimal karena hanya dengan menggunakan 11 data latih dapat menghasilkan akurasi sebesar 94.5% untuk data uji dengan jumlah yang lebih banyak, yaitu 20 buah data.

Tabel 6 Perbandingan akurasi dengan penelitian sebelumnya

Metode	Akurasi tertinggi	Data latih	Data uji
HMM - MFCC [12]	71.25%	20	40
	77.92%	30	30
	86.25%	40	20
HMM - LPC [13]	98.90%	10	5
PNN - MFCC [14]	84%	20	20
	90%	30	15
	94%	40	10
PNN bertingkat - MFCC [10]	67%	20	20
	82%	30	20
	96%	40	20
HMM - 2D MFCC [5]	88%	20	60
	92%	40	40
	99%	60	20
VFI5 - MFCC	94.5%	11	20
	97%	38	20

5. Kesimpulan

Dari penelitian yang telah dilakukan, diperoleh suatu model *Voting Feature Intervals 5* untuk identifikasi pembicara.

Model yang dikembangkan ini sudah mampu mengidentifikasi suara dengan akurasi identifikasi tertinggi mencapai 97 %. Tingkat akurasi yang dihasilkan model sangat dipengaruhi oleh banyaknya data latih yang digunakan. Semakin banyak data latih yang digunakan, semakin tinggi akurasi model. Secara statistika penggunaan 11 data latih telah mampu menghasilkan akurasi yang optimal.

Pada percobaan menggunakan data yang memiliki *noise* dengan SNR sebesar 30 dB, model ini masih memiliki tingkat akurasi yang cukup tinggi mencapai 81,5%. Sedangkan untuk *noise* dengan SNR sebesar 20 dB, tingkat akurasi tertinggi mencapai 59%. Dengan demikian, model

pengidentifikasi pembicara menggunakan algoritme VFI5 dengan MFCC sebagai pengekstraksi ciri yang telah dikembangkan belum cukup handal untuk mengidentifikasi suara ber-*noise* cukup tinggi.

6. Daftar pustaka

[1] Jain, A.K. dan Ross, A. Introduction to Biometrics. Di dalam Jain, A.K., Flynn, P., Ross, A.A. editor. *Handbook of Biometrics*, Springer, New York, 2008.

[2] Campbell, Jr. J.P. Speaker Recognition: A Tutorial. *Proceedings of The IEEE*, *85* (9). 1437-1461.

[3] Musyaffa, F.A. Skripsi. *Pengenalan Tanda Tangan Menggunakan Algoritme VFI5 Melalui Praproses Wavelet*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor, 2009.

[4] Santoso, A. Skripsi. *Pengenalan Wajah dengan Partisi Menggunakan Algoritme VFI5 Berbasis Histogram*, Institut Pertanian Bogor, Bogor, 2011.

[5] Buono, A., Jatmiko, W. dan Kusumoputro, B. Perluasan Metode MFCC 1D ke 2D Sebagai Esktraksi Ciri Pada Sistem Identifikasi Pembicara Menggunakan Hidden Markov Model (HMM). *MAKARA, 13* (1). 87-93.

[6] Ganchev, T.D. Tesis. *Speaker Recognition*. Wire Communications Laboratory, Department of Computer and Electrical Engineering, University of Patras, Greece, 2005.

[7] Do, M.N. *Digital Signal Processing Mini-Project: An Automatic Speaker Recognition System*. Audio Visual Communications Laboratory, Swiss Federal Institute of Technology, Laussanne, Switzerland, 1994. Diakses pada 12 Juli 2006 dari http://lcavwww.epfl.ch/~minhdo/asr_project.pdf.

[8] Güvenir, H.A., Demiröz, G. dan İlter, N. 1998. Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. *Artificial Intelligence in Medicine, 1998* (13). 147-165.

[9] Güvenir, H.A. A Classification Learning Algorithm Robust to Irrelevant Features. Di Dalam Giunchiglia F. editor. *Artificial Intelligence: Methodology, Systems Applications. Proceeding of AIMSA '98* (Sozopol, 21-23 September 1998), Sozopol: Springer-Verlag. 281-290.

[10] Zilvan, V. Skripsi. *Pengembangan Model Probabilistic Neural Network Bertingkat Menggunakan Fuzzy C-Means untuk Identifikasi Pembicara*, Institut Pertanian Bogor, Bogor, 2007.

[11] Mandasari, Y. Skripsi. *Pengembangan Model Markov Tersembunyi untuk Pengenalan kata Berbahasa Indonesia*. Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor, 2005.

[12] Purnamasari, W. Skripsi. *Pengembangan Model Markov Tersembunyi untuk Identifikasi Pembicara*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor, 2006.

[13] Ihsan, M. Tesis. *Pengembangan Model Markov Tersembunyi pada Identifikasi Pembicara*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor, 2006.

[14] Suhartono, M.N. Skripsi. *Pengembangan Model Identifikasi Pembicara dengan Probabilistic Neural Netwok*, Institut Pertanian Bogor, Bogor, 2007.