

# Rancang Bangun Aplikasi Pendeteksi Suara Tangisan Bayi

Welly Setiawan Limantoro, Chastine Faticah, dan Umi Laili Yuhana

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: [chastine@if.its.ac.id](mailto:chastine@if.its.ac.id)

**Abstrak**—Suara tangisan bayi merupakan sebuah tanda dari bayi yang mengalami suatu masalah. Namun, tidak semua orang dapat mengenali arti tangis bayi. Beberapa penelitian tentang deteksi suara tangis bayi sudah dilakukan oleh beberapa peneliti, namun saat ini masih belum ada penelitian yang membuat sebuah aplikasi pendeteksi suara tangis bayi berbasis *web*. Pada penelitian ini, sebuah aplikasi dibuat untuk membantu pengguna mengenali suara tangis bayi berbasis *Dunstan Baby Language*. Metode yang diterapkan pada aplikasi ini adalah ekstraksi fitur suara tangis bayi dengan algoritma *Mel-Frequency Cepstrum Coefficient* (MFCC), normalisasi hasil ekstraksi fitur, dan klasifikasi *K-nearest Neighbor*. Dari berbagai pengujian yang dilakukan, dapat disimpulkan bahwa akurasi rata-rata terbaik sebesar 75,95% dapat dicapai ketika menggunakan parameter *wintime* pada ekstraksi fitur MFCC sebesar 0,08 detik, proporsi data latih 85% dan data uji 15% dari setiap kelas, normalisasi ekstraksi fitur dengan *Standard Deviation Normalization*, dan klasifikasi *K-nearest Neighbor* dengan  $k=1$ . Hasil ini lebih baik jika dibandingkan dengan metode klasifikasi lain seperti *Naive Bayes*, *Neural Network*, maupun *Support Vector Machine*. Pada pengujian aplikasi dengan seluruh data, akurasi rata-rata yang sebesar 96,57% dapat dicapai ketika menggunakan parameter *wintime* pada ekstraksi fitur MFCC sebesar 0,08 detik, proporsi data latih 85% setiap kelas, normalisasi ekstraksi fitur dengan *Standard Deviation Normalization*, dan klasifikasi *K-nearest Neighbor* dengan  $k=1$ .

**Kata Kunci**—*Dunstan Baby Language*, *K-nearest Neighbor*, *Mel-Frequency Cepstrum Coefficient*, Suara Tangis Bayi

## I. PENDAHULUAN

SUARA tangisan bayi merupakan sebuah tanda dari bayi yang mengalami suatu masalah. Suara tangisan tersebut dapat digunakan untuk mengidentifikasi masalah pada bayi, seperti kelaparan, kesakitan, rasa kantuk, rasa tidak nyaman, kedinginan atau kepanasan, dan lain-lain. Dari suara tangis itu, seseorang yang telah terbiasa mengasuh bayi dapat mengerti apa maksud tangisan tersebut. Namun, tidak semua orang mengerti arti tangis bayi.

Dari permasalahan tentang tangis bayi, ada beberapa penelitian yang mengangkat tema bagaimana cara untuk mendeteksi masalah pada tangis bayi. Beberapa penelitian tersebut ialah “*Application of Neuro-Fuzzy Approaches to Recognition and Classification of Infant Cry*” [1], “*Identifikasi Arti Tangis Bayi Versi Dunstan Baby Language Menggunakan Jarak Terpendek dari Jarak Mahalanobis*” [2], dan “*Normal and Hypoacoustic Infant Cry Signal Classification Using Time-Frequency Analysis and General Regression Neural Network*” [3]. Penelitian [1] dan [2] menggunakan *Dunstan Baby Language* sebagai acuan untuk mengidentifikasi jenis tangis

bayi, sementara penelitian [3] fokus pada identifikasi suara tangis bayi yang normal atau tuli. Berdasarkan tiga penelitian tersebut, dapat disimpulkan bahwa penelitian tentang arti tangis bayi terus berkembang, baik dari segi metode ekstraksi fitur maupun metode identifikasi. Namun, saat ini masih belum ada penelitian yang mengembangkan aplikasi pendeteksi tangis bayi berbasis *web* agar dapat digunakan oleh banyak orang untuk mengetahui arti suara tangis bayi.

Maka dari itu, untuk mengenal arti tangis bayi berbasis *Dunstan Baby Language*, penulis menerapkan metode ekstraksi fitur *Mel-Frequency Cepstrum Coefficients* (MFCC) dan metode klasifikasi *K-nearest Neighbor* pada aplikasi pendeteksi suara tangisan bayi berbasis *web*. Metode ekstraksi fitur MFCC dipilih karena telah digunakan oleh banyak peneliti untuk menganalisa pola kecocokan suara, sedangkan metode klasifikasi *K-nearest Neighbor* dipilih karena waktu yang diperlukan untuk komputasi klasifikasi pada data uji yang cukup singkat dengan komposisi data latih yang cukup banyak. Pembuatan aplikasi berbasis *web* ini dipilih karena mempertimbangkan kemudahan penggunaan aplikasi di manapun dan kapanpun. Selain itu, pengguna tidak perlu mengalokasikan ruang pada media penyimpanan untuk pemasangan aplikasi, baik pada komputer atau perangkat bergerak lainnya, seperti *smartphone* atau tablet. Dengan adanya aplikasi tersebut, pengguna diharapkan dapat mengetahui arti dari suara tangisan bayi yang bersangkutan sehingga dapat memberikan penanganan yang tepat terhadap suara tangisan tersebut.

## II. TINJAUAN PUSTAKA

### A. *Dunstan Baby Language*

*Dunstan Baby Language* [4] adalah gagasan tentang arti tangisan bayi yang dicetuskan oleh Pricilla Dunstan pada tahun 2006. Menurut Pricilla, terdapat 5 jenis tangisan bayi universal (berlaku pada seluruh bayi di dunia) yang memiliki arti masing-masing, yaitu:

- *Neh (Aku lapar)* - Seorang bayi menggunakan refleks suara "Neh" untuk berkomunikasi bahwa ia sedang kelaparan. Suara yang dihasilkan berasal dari refleks menghisap dan lidah didorong di atap mulut.
- *Owh (Aku mengantuk)* - Seorang bayi menggunakan refleks suara "Owh" untuk berkomunikasi bahwa ia lelah. Suara yang dihasilkan terdegar seperti orang yang sedang menguap.
- *Heh (Aku merasa tidak nyaman)* - Seorang bayi

menggunakan refleksi suara "Heh" untuk berkomunikasi bahwa ia merasa tertekan, tidak nyaman, atau membutuhkan popok baru. Suara yang dihasilkan ialah respon pada masalah kulit bayi, biasanya rasa kepanasan (disertai keringat) atau rasa gatal.

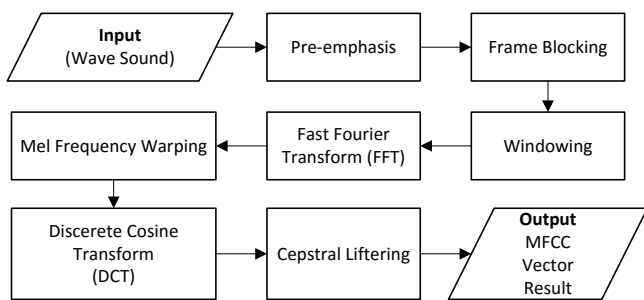
- *Eairh (Ada angin di perutku)* - Seorang bayi menggunakan refleksi suara "Eairh" untuk berkomunikasi bahwa ia ingin buang angin atau masalah perut. Suara tersebut dihasilkan ketika bayi tak dapat bersendawa, kemudian angin tersebut masuk ke perut dan menekan otot-otot perut agar angin tersebut dapat dikeluarkan. Seringkali, suara ini juga mengindikasikan bayi ingin buang air besar yang disertai dengan gerakan menekuk lutut ke tubuhnya.
- *Eh (Aku ingin bersendawa)* - Seorang bayi menggunakan suara refleksi "Eh" untuk berkomunikasi bahwa ia ingin bersendawa. Suara tersebut dihasilkan ketika angin tidak dapat keluar dan terperangkap di sekitar dada dan disertai dengan refleksi untuk mengeluarkan angin tersebut melalui mulut.

**B. Mel-Frequency Cepstrum Coefficient**

*Mel-Frequency Cepstral Coefficients* (MFCC) [5] adalah sebuah metode ekstraksi fitur suara yang merubah sinyal suara menjadi vektor. Keunggulan dari MFCC ialah sebagai berikut:

- Dapat menangkap karakter suara untuk mengetahui pola pada suara tertentu.
- Memberikan output berupa vektor dengan ukuran data yang kecil namun tidak menghilangkan karakteristik dari suara yang diekstrak.
- Cara kerja algoritma MFCC mirip dengan cara kerja pendengaran manusia dalam memberikan persepsi terhadap suara yang didengarkan.

Alur ekstraksi fitur MFCC terdapat pada **Gambar 1**.



Gambar 1 Diagram Alur Ekstraksi Fitur MFCC

**1) Pre-Emphasis**

*Pre-Emphasis* [6] digunakan untuk mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum yang umumnya tereduksi saat proses produksi suara. Pada tahap ini, *file* audio yang telah dibaca sebagai data sinyal frekuensi dalam tipe data *array* sepanjang *N* detik akan ditransformasi menjadi sinyal frekuensi baru dengan persamaan (1) [7]. Persamaan *Pre-Emphasis* tertulis pada persamaan (1), di mana  $y_n$  adalah sinyal hasil *Pre-Emphasis* dan  $x_n$  adalah sinyal sebelum *Pre-Emphasis*. Misalkan nilai dari  $\alpha = 0,97$ , maka 97% dari setiap sampel dianggap berasal dari sampel sebelumnya.

$$y_n = x_n - \alpha \times x_{n-1}, 0,9 \leq \alpha \leq 1,0 \tag{1}$$

**2) Frame Blocking**

*Frame Blocking* [6] digunakan untuk membuat segmentasi pada sinyal frekuensi sebesar 20-30 milidetik (dikenal sebagai *frame*). Sinyal suara dibagi menjadi *N* sampel *frame* dan *frame* yang berdekatan dipisahkan oleh *M* ( $M < N$ ) [5]. Pada umumnya nilai  $N=256$  dan  $M=100$ . Panjang *frame* akan menentukan keberhasilan analisa spektral. Ukuran *frame* harus sepanjang mungkin untuk menunjukkan resolusi frekuensi yang baik, namun juga harus cukup pendek untuk menunjukkan resolusi waktu yang baik [6].

Proses *frame blocking* ini akan dilakukan secara terus-menerus sampai seluruh sinyal dapat diproses. Pada umumnya, proses ini dilakukan secara *overlapping* pada setiap *frame*-nya. Panjangnya *overlap* yang umum digunakan ialah 30% hingga 50% dari panjang *frame*. Dengan adanya *overlapping*, hilangnya karakteristik suara pada perbatasan perpotongan setiap *frame* dapat dihindari.

Sebagai contoh, jika audio yang telah melewati proses *Pre-Emphasis* memiliki sample rate sebesar 16000 Hz, maka segmentasi selama 20 milidetik dilakukan setiap 320 *frame*. Hasil tersebut diperoleh dari perkalian sample rate dengan durasi segmentasi. Kemudian, jika *overlap* yang dilakukan sebesar 50%, maka segmen pertama ialah dengan *frame* urutan 1-320, diikuti dengan segmen kedua dengan *frame* urutan 160-480 dan seterusnya [7].

**3) Windowing**

*Windowing* [6] digunakan untuk mengurangi terjadinya efek *aliasing* pada sinyal yang diproses setelah proses *framing*. *Aliasing* adalah sinyal baru yang frekuensinya berbeda dengan frekuensi sinyal aslinya akibat dari rendahnya *sampling rate* atau proses *frame blocking* yang menyebabkan sinyal terputus. Pada tahap ini, setiap sinyal frekuensi berupa spektrum suara ke-*n* yang telah diproses dari *Frame Blocking* dikalikan dengan nilai *Hamming Window* [7]. Nilai *Hamming Window* didapat dari fungsi *window*. Persamaan *windowing* tertulis pada (2), di mana *N* adalah jumlah sampel pada setiap *frame*,  $y_n$  adalah nilai sampel sinyal hasil *windowing*,  $x_n$  adalah nilai sampel dari *frame* sinyal ke-*n*, dan  $w_n$  adalah fungsi *window*. Fungsi *window* yang sering digunakan pada proses *windowing* ini ialah *Hamming Window* (3) karena menghasilkan *sidelobe level* yang tidak terlalu tinggi dan *noise* yang tidak terlalu banyak.

$$y_n = x_n \times w_n, 0 \leq n \leq N - 1 \tag{2}$$

$$w_n = 0,54 - 0,46 \cos \frac{2\pi n}{N-1}, 0 \leq n \leq N - 1 \tag{3}$$

**4) Fast Fourier Transform**

*Fast Fourier Transform* [6] (FFT) digunakan mengubah tiap *frame* frekuensi pada sample *N* dari domain waktu menjadi domain frekuensi dan untuk menganalisa *spectral properties* berdasarkan sinyal yang dimasukkan. *Spectral properties* ini biasa dikenal sebagai *spectrogram*. *Spectrogram* berhubungan dengan waktu dan frekuensi. Jika resolusi waktu yang digunakan tinggi, maka frekuensi yang dihasilkan menjadi semakin rendah. Pada tahap ini, sinyal frekuensi dari proses *windowing* akan diproses dengan persamaan (4), di mana  $X_k$  adalah jumlah frekuensi *k* pada sinyal, *k* adalah frekuensi audio sebelum FFT (0 Hz sampai  $N - 1$  Hz),  $x_n$  adalah nilai sinyal pada waktu ke-*n*, dan *N* adalah jumlah waktu pada sampel [8].

$$X_k = \sum_{n=0}^{N-1} x_n \times e^{\frac{-2\pi kn}{N}}, n = 0 \leq n \leq N - 1 \quad (4)$$

5) *Mel Frequency Warping*

*Mel Frequency Warping* [7] dilakukan untuk menghasilkan *magnitude spectrum* yang bagus dan memperkecil ukuran fitur terkait. *Mel Frequency Warping* dilakukan dengan menerapkan *filterbank* untuk mengetahui ukuran energi *frequency band* tertentu dalam sinyal suara [6]. *Filterbank* ini dapat diterapkan pada domain frekuensi maupun domain waktu, namun karena digunakan untuk MFCC, maka domain frekuensi yang diterapkan pada penelitian ini.

Pada tahap *Mel Frequency Warping*, sinyal frekuensi yang telah diproses pada tahap *Fast Fourier Transform* akan dirubah ke dalam frekuensi *Mel* dengan persamaan (5), di mana  $f$  adalah frekuensi sinyal. Kemudian, untuk mengetahui besarnya energi pada frekuensi *Mel*, dilakukan perhitungan *filterbank* [7]. Persamaan *filterbank* terdapat pada (6), di mana  $N$  adalah jumlah *magnitude* spektrum (nilai yang sering digunakan ialah 40),  $S[j]$  adalah *magnitude spectrum* yang pada frekuensi  $j$ ,  $H_i[j]$  adalah koefisien *filterbank* pada frekuensi  $j$ , dan  $M$  adalah jumlah channel pada *filterbank*.

$$Mel(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (5)$$

$$y[i] = \sum_{j=1}^N S[j] H_i[j], j = 1 \leq i \leq M \quad (6)$$

6) *Discrete Cosine Transform*

*Discrete Cosine Transform* [7] merupakan proses terakhir dari rangkaian proses MFCC yang digunakan untuk mengonversi *mel spectrum* menjadi domain yang mirip waktu (dikenal sebagai *quefrency*) sehingga spektral lokal dapat terepresentasi dengan baik. Hasil dari konversi ini disebut sebagai *Mel-Frequency Cepstrum Coefficient* yang berbentuk deretan vektor akustik. Pada tahap ini, frekuensi *Mel* ditransformasi dengan persamaan (7), di mana  $N$  adalah jumlah dari *Mel Frequency Warping* (nilai yang sering digunakan ialah 40),  $L$  adalah jumlah dari *Mel-Scale Cepstral Coefficients* (nilai yang sering digunakan ialah 12).

$$C_m = \sum_{k=1}^N \cos \left[ m \left( k - \frac{1}{2} \right) \frac{\pi}{N} \right] E_k, m = 1, 2, \dots, L \quad (7)$$

7) *Cepstral Liftering*

*Cepstral Liftering* [9] digunakan untuk memperhalus hasil dari seluruh proses MFCC agar lebih hasilnya lebih baik saat digunakan pada *pattern matching*. *Cepstral Liftering* diimplementasikan pada hasil *Discrete Cosine Transform* berupa *cepstral features* dengan menggunakan fungsi *window* pada persamaan, di mana  $L$  adalah jumlah *cepstral coefficients* dan  $n$  adalah indeks dari *cepstral coefficients*.

$$w_n = 1 + \frac{L}{2} \sin \left[ \frac{\pi n}{L} \right] \quad (8)$$

C. *Normalisasi*

Normalisasi adalah sebuah teknik penskalaan data agar distribusi data sesuai dengan standar tertentu. Beberapa teknik normalisasi yang digunakan dalam penelitian ini ialah sebagai berikut.

1) *Standard Deviation Normalization*

*Standard Deviation Normalization* [1] (9) adalah metode untuk menormalkan data berdasarkan besarnya nilai standar deviasi dari sebuah kumpulan data. Cara kerja metode ini ialah nilai data yang ke sekian dikurangi dengan nilai rata-rata dari kumpulan data terkait kemudian dibagi dengan nilai standar deviasi dari data terkait.

$$x_i = \frac{data_i - \mu(data)}{\sigma(data)} \quad (9)$$

2) *Min-Max Normalization*

*Min-Max Normalization* [1] (10) adalah metode untuk menormalkan data berdasarkan besarnya nilai maksimum dikurangi dengan nilai minimum dari sebuah kumpulan data. Cara kerja metode ini ialah nilai data yang ke sekian dikurangi dengan nilai minimum dari kumpulan data terkait kemudian dibagi dengan nilai maksimum yang dikurangi nilai minimum dari data terkait.

$$x_i = \frac{data_i - \min(data)}{\max(data) - \min(data)} \quad (10)$$

3) *Decimal Normalization*

*Decimal Normalization* [1] (11) adalah metode untuk menormalkan data berdasarkan besarnya nilai  $10^j$  ( $j$  adalah bilangan bulat). Cara kerja metode ini ialah data nilai data yang ke sekian dibagi dengan  $10^j$ .

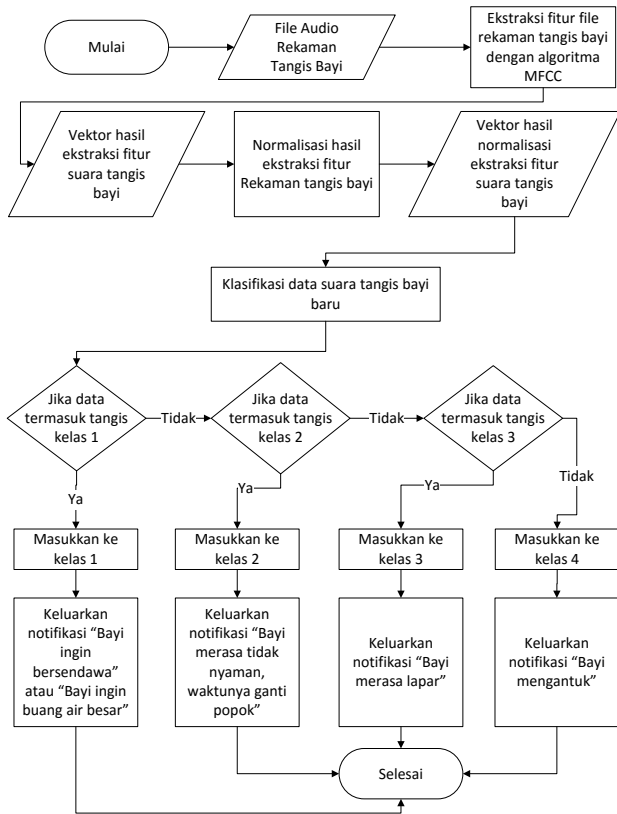
$$x_i = \frac{data_i}{10^j} \quad (11)$$

D. *K-nearest Neighbor*

$$d_E = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (12)$$

*K-Nearest Neighbor* (KNN) [10] (12) adalah algoritma sederhana yang menyimpan semua kasus yang tersedia dan mengklasifikasikan kasus baru berdasarkan ukuran kesamaan menggunakan fungsi jarak. Cara kerja algoritma ini ialah sebuah data uji diklasifikasikan menjadi suatu kelas tertentu berdasarkan jumlah tetangga terdekat dengan menggunakan perhitungan jarak *Euclidian*. Saat perhitungan, jarak antar data uji dengan seluruh data latih beserta label kelas disimpan pada vektor, kemudian diurutkan berdasarkan jarak terpendek. Data uji akan masuk ke kelas tertentu berdasarkan jumlah label terbanyak dari nilai  $k$  yang ditentukan.

### III. PERANCANGAN SISTEM



Gambar 2 Diagram Proses Klasifikasi Suara Tangis Bayi

Pada penelitian ini akan dibangun suatu sistem yang dapat melakukan klasifikasi terhadap rekaman suara tangis bayi kedalam empat kelas yaitu kelas “Eairh/Eh” (bayi ingin bersendawa, kentut, atau buang air besar), kelas “Heh” (bayi merasa tidak nyaman, waktunya ganti popok), kelas “Neh” (bayi merasa lapar), atau kelas “Owh” (bayi mulai mengantuk), di mana rekaman suara tangis bayi yang dipakai memiliki durasi maksimal 10 detik. Pada tahap ini dilakukan beberapa proses meliputi ekstraksi fitur dengan algoritma MFCC, normalisasi hasil ekstraksi fitur, dan tahap klasifikasi berdasarkan fitur yang telah didapatkan. Diagram alur klasifikasi suara tangis bayi dapat dilihat pada Gambar 2.

### IV. UJI COBA DAN EVALUASI

#### A. Data Uji Coba

Tabel 1  
Jumlah Data Tiap Jenis Tangis Bayi

Jenis Tangis	Kelas (angka)	Jumlah rekaman
Eairh/Eh	1	46
Heh	2	23
Neh	3	35
Owh	4	35

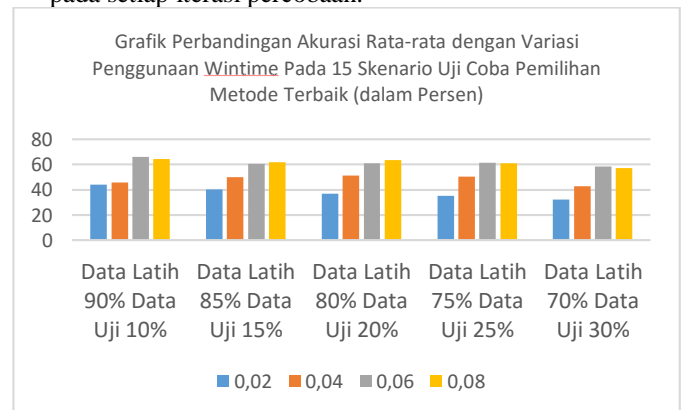
Data uji coba yang digunakan pada penelitian ini adalah rekaman suara tangis bayi yang didapatkan dari video tangis bayi *Dunstan Baby Language* yang telah dipotong-potong berdasarkan durasi waktu tertentu (maksimal 10 detik) dan dikonversi menjadi *file* audio dengan format “.wav”, *channel* audio *mono*, dan *sample rate* 11205 Hz dengan bantuan *tools* Adobe Premiere Pro CS6. Video tersebut didapatkan dari

penelitian [1] yang terdiri atas lima jenis tangis bayi (“Eairh”, “Eh”, “Heh”, “Neh”, dan “Owh”) yang nantinya akan dikelompokkan menjadi empat kelas yaitu kelas 1 “Eairh/Eh” (bayi ingin bersendawa, kentut, atau buang air besar), kelas 2 “Heh” (bayi merasa tidak nyaman, waktunya ganti popok), kelas 3 “Neh” (bayi merasa lapar), dan kelas 4 “Owh” (bayi mulai mengantuk). Hasil potongan video berupa *file* audio disempurnakan dengan menghilangkan gelombang audio di bagian awal yang bernilai nol menggunakan *tools* Adobe Audition CS5.5. Jumlah data tiap jenis tangis bayi terdapat pada Tabel 1.

#### B. Uji Coba Pemilihan Metode

Uji coba dilakukan dengan mengklasifikasi data tangis bayi dari video *Dunstan Baby Language* yang dikonversi menjadi audio berformat “.wav” hasil ekstraksi fitur yang sudah dinormalisasi pada empat jenis tangis bayi. Hasil dari ekstraksi fitur yang sudah dinormalisasi adalah sebuah *data frame* dataset yang kemudian diklasifikasi dengan *K-nearest Neighbor* untuk menentukan akurasi pada masing-masing kelas. Terdapat 15 macam skenario dari uji coba dengan menggunakan kombinasi beberapa parameter. Berikut ini merupakan penjelasan kombinasi parameter yang digunakan pada 15 skenario uji coba.

- *Parameter wintime*: merupakan kombinasi panjang *window* dalam detik pada sebuah *file* audio yang akan diproses melalui ekstraksi fitur MFCC. Kombinasi nilai yang digunakan pada panjang *window* ini adalah 0,02 sampai 0,08 dengan kenaikan 0,02 pada setiap iterasi percobaan.
- *Parameter Proporsi Data Latih dan Data Uji*: merupakan kombinasi persentase data uji dan data latih pada masing-masing kelas yang terdiri dari 70% data latih 30% data uji hingga 90% data latih dan 10% data uji dengan kenaikan 5% data latih dan penurunan 5% data uji pada setiap iterasi percobaan.
- *Parameter Normalisasi Hasil Ekstraksi Fitur*: merupakan kombinasi pemakaian metode normalisasi pada hasil ekstraksi fitur MFCC. Kombinasi metode normalisasi yang digunakan adalah *Decimal Normalization*, *Min-Max Normalization*, dan *Standard Deviation Normalization* pada setiap iterasi percobaan.
- *Parameter Nilai k pada K-nearest Neighbor*: merupakan kombinasi penggunaan nilai *k* pada *K-nearest Neighbor*. Nilai yang digunakan ialah 1 sampai 5 dengan kenaikan 1 pada setiap iterasi percobaan.



Gambar 3 Akurasi Rata-rata Penggunaan Kombinasi *wintime* pada 15 skenario percobaan

Berdasarkan hasil uji coba pemilihan metode dari 15 skenario yang telah dilakukan, rata-rata akurasi terbaik sebesar 75,95% didapat ketika menggunakan *wintime* 0,08 dengan proporsi data latih 85% data uji 15%, nilai  $k=1$  pada *K-nearest Neighbor*, dan normalisasi hasil ekstraksi fitur dengan *Standard Deviation Normalization*. Dari 15 skenario uji coba tersebut, parameter *wintime* merupakan parameter yang paling berpengaruh terhadap akurasi klasifikasi. Semakin lama durasi dari audio yang akan diproses dengan ekstraksi fitur MFCC, maka penentuan *wintime* harus semakin besar agar fitur yang didapat semakin baik. Perbandingan akurasi penggunaan durasi *wintime* dapat dilihat pada Gambar 3.

C. Uji Coba dengan Data Latih Seimbang pada Tiap Kelas

Pengujian dilakukan dengan menggunakan parameter *wintime* sebesar 0,08 detik, *K-nearest Neighbor* dengan  $k=1$  sampai  $k=5$ , normalisasi hasil ekstraksi fitur dengan *Standard Deviation Normalization*, dan data latih seimbang pada tiap kelas dilakukan untuk menganalisis kecenderungan salah deteksi dari jenis tangis satu ke jenis tangis lainnya. Data latih pada masing-masing kelas jenis tangis bayi akan diatur sebanyak 19, kemudian sisa data yang lain akan digunakan sebagai data uji. Akurasi hasil uji coba dengan data latih seimbang dapat dilihat pada Tabel 2.

Berdasarkan uji coba dengan data latih seimbang, akurasi rata-rata terbaik yang didapat untuk mengklasifikasikan empat jenis tangis bayi sebesar 64,53% ketika nilai  $k=1$  pada *K-nearest Neighbor*. Dari hasil uji dengan *K-nearest Neighbor*  $k=1$ , masing-masing suara tangis bayi ada yang dominan salah dideteksi sebagai jenis tangis lain. Kesalahan deteksi suara tangis “Eairh/Eh” dominan pada tangis “Owh”, suara tangis “Heh” ada satu file yang salah dideteksi sebagai tangis “Eairh/Eh”, suara tangis “Neh” dideteksi sebagai tangis “Eairh/Eh” dan “Heh”, dan suara tangis “Owh” dominan salah dideteksi sebagai tangis “Neh”.

Secara garis besar, jika dibandingkan dengan pembagian proporsi data latih berdasarkan persentase data dari tiap jenis bayi, akurasi yang dihasilkan jauh lebih baik karena data latih dari tiap jenis tangis bayi yang digunakan jauh lebih banyak.

Tabel 2  
Perbedaan Hasil Klasifikasi KNN dengan Variasi Nilai  $k$  pada Data Latih Seimbang

Jenis Tangis	Klasifikasi KNN				
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Eairh/Eh	51,85%	70,37%	55,56%	51,85%	55,56%
Heh	75,00%	75,00%	100,00%	100,00%	75,00%
Neh	75,00%	50,00%	56,25%	62,50%	56,25%
Owh	56,25%	31,25%	43,75%	37,50%	37,50%
Akurasi rata-rata	<b>64,53%</b>	56,66%	63,89%	62,96%	56,08%

D. Uji Coba dengan Metode Klasifikasi Lain

Tabel 3  
Perbandingan Akurasi Metode Klasifikasi

Jenis Tangis	Metode Klasifikasi			
	KNN	<i>Naive Bayes</i>	<i>Neural Network</i>	SVM
Eairh/Eh	57,14%	28,57%	42,86%	42,86%
Heh	66,67%	100,00%	100,00%	66,67%
Neh	80,00%	20,00%	40,00%	60,00%
Owh	100,00%	80,00%	80,00%	100,00%
Rata-Rata Akurasi	<b>75,95%</b>	57,14%	65,71%	67,38%

Uji coba dilakukan dengan menggunakan metode klasifikasi selain *K-nearest Neighbor* dengan  $k=1$  untuk mengetahui perbandingan akurasi dari metode lain. Metode klasifikasi yang akan dibandingkan ialah *Naive Bayes*, *Support Vector Machine*, dan *Neural Network*. Pada klasifikasi *Support Vector Machine*, dengan metode *One-against-one*, kernel yang digunakan ialah *polynomial* dan tidak menggunakan *cross validation*. Pada metode klasifikasi *Neural Network*, jumlah *hidden layer* yang digunakan sebanyak dua dengan 7 dan 5 node pada masing-masing *layer*, fungsi aktivasi yang digunakan ialah fungsi logistik, algoritma *training* yang digunakan ialah *resilient backpropagation* dengan *weight backtracking*, dan koefisien *learning rate* yang digunakan sebesar 0,02. Proporsi data latih dan data uji yang digunakan pada tiga metode perbandingan ialah 85% data latih dan 15% data uji dari tiap kelas. Nilai *wintime* yang digunakan pada ekstraksi fitur MFCC dari *package tuneR* sebesar 0,08 detik. Normalisasi hasil ekstraksi fitur yang digunakan ialah *Standard Deviation Normalization*.

Berdasarkan skenario uji coba perbandingan tiga metode klasifikasi selain *K-nearest Neighbor* dengan  $k=1$  pada tiap jenis tangis bayi, rata-rata akurasi klasifikasi terbaik sebesar 75,95% didapat ketika menggunakan metode klasifikasi *K-nearest Neighbor* dengan  $k=1$ . Perbandingan akurasi tiap metode klasifikasi dapat dilihat pada Tabel 3. Perbedaan akurasi hasil klasifikasi ini wajar karena setiap metode memiliki kelebihan maupun kekurangan masing-masing dalam menangani data yang akan diklasifikasi

E. Uji Coba Aplikasi Pendeteksi Suara Tangisan Bayi

Uji coba aplikasi dilakukan untuk menguji apakah aplikasi mampu berjalan dengan baik dalam melakukan klasifikasi suara tangis bayi. Uji coba aplikasi dilakukan dengan menerapkan parameter berikut ini:

- *wintime* = 0,08.
- data latih 85% dari setiap kelas jenis tangis bayi.
- nilai  $k=1$  pada *K-nearest Neighbor*.
- normalisasi hasil ekstraksi fitur dengan *Standard Deviation Normalization*.

Sebanyak 139 audio tangis bayi diunggah ke aplikasi penerjemah tangis bayi secara satu per satu. Akurasi per kelas dan akurasi rata-rata pada uji coba aplikasi dapat dilihat pada Tabel 4.

Berdasarkan skenario uji coba aplikasi pendeteksi suara tangis bayi yang dilakukan, akurasi tiap kelas jenis tangis bayi dan akurasi rata-rata aplikasi sangat baik. Hasil akurasi yang tinggi pada aplikasi tersebut dapat dicapai karena 85% data latih pada tiap kelas jenis tangis bayi yang digunakan pada aplikasi pendeteksi suara tangis bayi juga digunakan sebagai data uji. Aplikasi dapat dikunjungi pada *link* di bawah ini:

[https://liemwellys.shinyapps.io/Aplikasi\\_Penerjemah\\_Tangis\\_Bayi/](https://liemwellys.shinyapps.io/Aplikasi_Penerjemah_Tangis_Bayi/).

Tabel 4  
Hasil Uji Coba Aplikasi Pendeteksi Suara Tangis Bayi

Eairh/Eh	Heh	Neh	Owh	Akurasi Rata-Rata
93,48%	95,65%	97,14%	100,00%	96,57%

V. KESIMPULAN

- Parameter *wintime* pada proses ekstraksi fitur MFCC dari *package tuneR* ditentukan berdasarkan panjangnya durasi



audio. Semakin panjang durasi audio yang akan diproses melalui ekstraksi fitur, maka nilai *wintime* harus semakin besar.

- Rata-rata akurasi terbaik sebesar 79,95% pada klasifikasi *K-nearest Neighbor* dengan  $k=1$  didapat ketika proporsi data latih 85% data uji 15% dan proses ekstraksi fitur MFCC dengan durasi *wintime* 0,08 detik yang dinormalisasi dengan *Standard Deviation Normalization*.
- Hasil klasifikasi dengan pembagian proporsi data latih berdasarkan persentase data dari masing-masing jenis tangis bayi menghasilkan akurasi yang lebih baik jika dibandingkan dengan klasifikasi yang menggunakan data latih seimbang dari tiap jenis tangis bayi.
- Aplikasi dapat mendeteksi masing-masing jenis tangis bayi dengan akurasi rata-rata 96,57%. Hasil ini diperoleh karena 85% dari data uji pada tiap kelas digunakan sebagai data latih dalam aplikasi.

#### DAFTAR PUSTAKA

- [1] K. Srijanon dan N. Eiamkanitchat, "Application of Neuro-Fuzzy Approaches to Recognition and Classification of Infant Cry," dalam *TENCON 2014 - 2014 IEEE Region 10 Conference*, Bangkok, 2014.
- [2] M. D. Renanti, "Identifikasi Arti Tangis Bayi Versi Dunstan Baby Language Menggunakan Jarak Terpendek dari Jarak Mahalanobis," *Jurnal Sains Terapan*, vol. 3, no. 5, pp. 39-44, 2013.
- [3] M. Hariharan, R. Sindhu dan S. Yaacob, "Normal and Hypoacoustic Infant Cry Signal Classification Using Time-Frequency Analysis and General Regression Neural Network," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 2, pp. 559-569, November 2012.
- [4] P. Dunstan, "Open Up and Discover Your Baby's Language," 2006. [Online]. Available: [http://www.babytaal.nl/media/PDF/ComprehensiveBooklet\(2\).pdf](http://www.babytaal.nl/media/PDF/ComprehensiveBooklet(2).pdf). [Diakses 2 Juli 2016].
- [5] L. Muda, M. Begam dan I. E. , "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques," *JOURNAL OF COMPUTING*, vol. 2, no. 3, pp. 138-143, 2010.
- [6] D. Putra dan A. Resmawan, "Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW," *Lontar Komputer*, vol. 2, pp. 8-21, 2011.
- [7] R. Jang, "12-2 MFCC," [Online]. Available: <http://mirilab.org/jang/books/audiosignalprocessing/speechFeatureMfcc.asp?title=12-2%20MFCC>. [Diakses 3 Juli 2016].
- [8] Kalid, "An Interactive Guide To The Fourier Transform," [Online]. Available: <http://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/>. [Diakses 5 Juli 2016].
- [9] Z.-H. Tan dan B. Lindberg, "Use of Cepstral Liftering for MFCC Variance Normalization," dalam *Automatic Speech Recognition on Mobile Devices and over Communication Networks*, Springer Science & Business Media, 2008, pp. 148-150.
- [10] "Pengertian, kekurangan dan kelebihan K-NN," 2012. [Online]. Available: <http://cgeduntuksemua.blogspot.com/2012/03/pengertian-kelebihan-dan-kekurangan-k.html>. [Diakses 15 Januari 2016].