

## DETEKSI UCAPAN ANGKA SATU SAMPAI SEPULUH BAHASA PALEMBANG MENGGUNAKAN MFCC DAN BOBOT DOMINAN

Heriyanto

Program Studi Teknik Informatika  
Jl. Babarsari 2 Tambakbayan 55281 Telp(0274) 485323  
e-mail : heriyanto@upnyk.ac.id

### Abstract

*Detecting speech with regional language, one of which is Palembang language, has uniqueness and distinctiveness in accent. Differences in dialects to check how precise and influential the accuracy of using MFCC and dominant weights. This study consists of three stages. The first stage, feature extraction of numerical numbers from one to ten using Mel Frequency Cepstral Coefficient (MFCC). The second stage is the selection of features that will be used as feature tables using the proposed model Normalized Dominant Weight (NBD) with threshold similarity, range, filtering, normalization of weights and dominant weights. The third stage is testing by checking by finding similarities in range, filtering, sequential multiplication and calculation of Suitability of Uniformity Patterns (CTF). The test results of checking MFCC and feature selection with normalization of dominant weights were 70% while without feature selection only 42%.*

**Keywords : extraction, weighting, dominant, normalization, range**

### Abstrak

Deteksi ucapan dengan berbahasa daerah salah satunya bahasa Palembang mempunyai keunikan dan kekhasan dalam logat berbahasa. Perbedaan logat bahasa untuk mengecek seberapa tepat dan berpengaruh terhadap akurasi menggunakan MFCC dan Bobot dominan. Penelitian ini terdiri atas tiga tahap. Tahap pertama, ekstraksi ciri angka bahasa angka satu sampai sepuluh menggunakan Mel Frequency Cepstral Coefficient (MFCC). Tahap kedua adalah pemilihan fitur yang akan dijadikan tabel fitur menggunakan model yang diusulkan Normalisasi Bobot Dominan (NBD) dengan kesamaan threshold, jangkauan, filtering, normalisasi bobot dan bobot dominan. Tahap ketiga adalah pengujian dengan pengecekan dengan cara mencari kesamaan jangkauan, filtering, perkalian sekuensial dan perhitungan Kesesuaian Keseragaman Pola (KKP). Hasil pengujian pengecekan terhadap MFCC dan pemilihan fitur dengan normalisasi bobot dominan sebesar 70% sedangkan tanpa pemilihan fitur hanya sebesar 42%.

**Kata kunci : ekstraksi, bobot, dominan, normalisasi, jangkauan**

## 1. PENDAHULUAN

Deteksi ucapan suara angka satu sampai sepuluh dengan berbahasa Palembang yang mempunyai ciri khas logat bahasa daerah tertentu dan berbeda logat daerah lainnya. Logat daerah tersebut yang berbeda diucapkan untuk mengecek dan mendeteksi suara ucapan dengan metode MFCC.

Metode MFCC diperkenalkan pertama kali oleh Davis dan Mermelstein sekitar tahun 1980. MFCC salah satu metode yang cukup baik dalam pengenalan suara dalam bidang *speech recognition* (Davis dan Mermelstein, 1980). MFCC merupakan ekstraksi ciri yang paling banyak digunakan dalam bidang *speaker recognition* dan *speech recognition*.

MFCC merupakan ekstraksi ciri yang menghasilkan fitur atau ciri yang merupakan pembeda satu dengan yang lain berupa parameter *cepstral coefficient* (Abriyono dan Harjoko, 2012). Ekstraksi ciri *Mel Frequency Cepstral Coefficient* (MFCC) mengubah gelombang suara menjadi beberapa tipe parameter seperti *cepstral coefficient* yang mempresentasikan *audio file* (Chamidy, 2016). Selain itu MFCC menghasilkan fitur vektor yang mengonversi sinyal suara menjadi beberapa vektor untuk pengenalan fitur suara (Putra dkk., 2011).

## 2. METODE PENELITIAN

Metode penelitian yang digunakan dengan ekstraksi ciri MFCC dan Normalisasi Bobot Dominan. MFCC mempunyai tahapan yaitu *pre-emphasis*, *frame blocking*, *windowing*, *Fast Fourier Transform* (FFT), *Mel Frequency Wrapping* (MFW), *Discrete Cosinus Transform* (DCT) dan *cepstral liftering* yang menghasilkan parameter sebagai fitur yaitu *frame* dan *cepstral coefficient* (Putra, 2011).

Ekstraksi ciri menggunakan *Mel Frequency Cepstral Coefficient* (MFCC) mempunyai tujuan untuk menghasilkan fitur dengan beberapa parameter. Parameter yang dihasilkan yaitu berupa *frame* dan *cepstral coefficient*. MFCC terdiri atas tujuh tahap *pre-emphasis*, *frame blocking*, *windowing*, *Fast Fourier Transform* (FFT), *Mel Frequency Wrapping* (MFW), *Discrete Cosine Transform* (DCT) dan *cepstral liftering*.

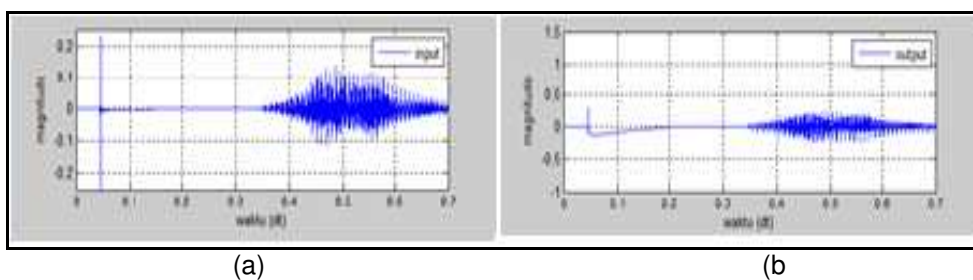
### 2.1 Pre-emphasis

*Pre-emphasis* menurut Chitode (2010) melakukan penekanan pada komponen frekuensi tinggi, menyelaraskan pada frekuensi rendah dan frekuensi tinggi. *Pre-emphasis* menurut Tokunbo Ogunfunmi (2015) mengurangi kebisingan untuk memperbaiki *Signal to Noise Rasio* (SNR) dan mengurangi suara yang tidak diinginkan. *Pre-emphasis* merupakan pemrosesan signal sederhana yang pada dasarnya adalah *filter linear* dan masih dalam ranah waktu (Holmes, 2003).

*Pre-emphasis* merupakan tahap awal dalam proses MFCC. Tahap ini, dilakukan karena sinyal sering sekali mengalami gangguan *noise*, sehingga diperlukan untuk mengurangi *noise* atau derau. *Pre-emphasis* merupakan cara yang sangat sederhana dalam mengurangi *noise*. *Pre-emphasis* mempunyai tujuan agar level *base band* pada bagian frekuensi tinggi masih memiliki kualitas sinyal yang baik. Proses *pre-emphasis* menurut Proakis dan Manolakis (1996) dengan nilai  $\alpha$  di antara 0 sampai 1 atau di antara  $0,9 \leq \alpha \leq 1,0$  menggunakan persamaan (2.1)

$$y(n) = s(n) - \alpha s(n - 1) \dots \dots \dots (2.1)$$

Dalam hal ini,  $y(n)$  adalah signal hasil *pre-emphasis*, sedangkan  $s(n)$  adalah signal sebelum *pre-emphasis*, simbol  $n$  merupakan nomor urut signal,  $\alpha$  merupakan konstanta *filter pre-emphasis* di antara 0,9-1,0 dan  $s$  adalah signal. Proses *pre-emphasis* yang dilakukan seperti pada Gambar 1 menunjukkan bagian (a) merupakan suara masukan sebelum dilakukan *pre-emphasis*, sedangkan Gambar 1 bagian (b) adalah hasil *output* pengolahan signal setelah proses *pre-emphasis*.



Gambar 1. *Pre-emphasis* (a) sebelum dan (b) sesudah (Putra, 2011)

Pengambilan signal ke- $n$  pada *pre-emphasis* dilakukan sepanjang bacaan satu kata atau dua kata dengan waktu satu sampai tiga detik.

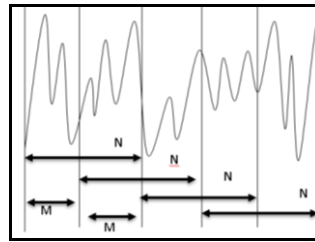
### 2.2 Frame blocking

Sinyal yang telah melalui *pre-emphasis* selanjutnya dilakukan proses *frame blocking* yang diblok dalam *frame* dengan  $N$  sampel dan digeser sebesar  $M$  sampel sehingga  $N=2M$  dengan  $M < N$ . Gambar 3.2 menunjukkan ilustrasi *frame blocking* (Abriyono dan Harjoko, 2012). Lebar *frame* dinotasikan dengan  $N$ , sedangkan lebar pergeseran setiap *frame* sebagai  $M$ . Lebar *overlap* dihitung selisih  $N-M$ .

*Frame blocking* menurut Holmes (2003) menganalisis sinyal ucapan ke dalam *frame* yang diwakili oleh fitur vektor tunggal digambarkan dalam spektrum rata-rata interval waktu. Waktu dalam *frame* diambil rata-rata antara 20-40 milidetik menurut Chamidy (2016). *Frame* diambil sepanjang mungkin untuk mendapatkan resolusi frekuensi yang baik, sedangkan waktu sependek mungkin dimaksudkan untuk mendapatkan ranah waktu yang terbaik. Perhitungan jumlah *frame blocking* menggunakan persamaan (2.2)

$$f_l(n) = y(Ml + n). \tag{2.2}$$

Dalam hal ini,  $f_l(n)$  merupakan hasil *frame blocking*, simbol  $n$  adalah  $0,1,\dots,N-1$ . Simbol  $N$  adalah jumlah sampel,  $M$  adalah panjang *frame*,  $l$  adalah  $0,1,\dots,L-1$ . Simbol  $L$  adalah seluruh signal dan  $y$  adalah hasil *pre-emphasis*.



Gambar 2. Ilustrasi *Frame blocking* (Abriyono dan Harjoko, 2012)

Gambar 2 menunjukkan  $M$  adalah *frame* pertama signal suara dalam formula disimbolkan  $f_l$  kemudian  $M+M=N$ .

**2.3 Windowing**

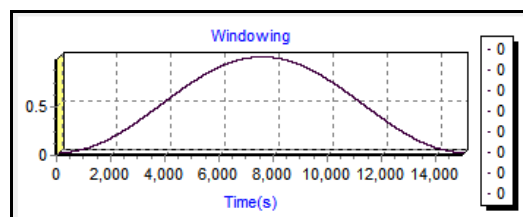
*Windowing* menurut Proakis dan Manolakis (1996) mempunyai fungsi efek menghaluskan pada spektrum setelah melalui proses *frame blocking*. *Windowing* mempunyai tujuan untuk mengurangi efek *diskontinue* pada ujung-ujung *frame* yang dihasilkan oleh proses *frame blocking*. *Windowing* yang digunakan yaitu *Rectangular Window*, *Hamming Window* dan *Hanning Window* (Chamidy, 2016). Dari ketiga fungsi *windowing* tersebut peneliti menggunakan *windowing Hanning* dikarenakan lebih halus dibandingkan dengan yang lain (Putra, 2008a). Representasi fungsi *windowing* menggunakan persamaan (2.3)

$$X(n) = f_l(n)w(n). \tag{2.3}$$

Dalam hal ini, fungsi  $X(n)$  adalah signal hasil *windowing*, dengan  $f_l$  adalah hasil *frame blocking*, dengan  $n$  adalah  $0,1,\dots,N-1$ . Simbol  $N$  adalah jumlah sampel dalam masing-masing *frame* dan  $w(n)$  adalah fungsi *window*. Sedangkan fungsi *windowing Hanning* menggunakan persamaan (2.4)

$$w(n) = 0,5 \left( 1 - \cos \left( \frac{2\pi n}{M-1} \right) \right) \tag{2.4}$$

Dalam hal ini,  $w(n)$  adalah fungsi *window* menggunakan *hanning*, dengan  $n$  adalah  $0,1,\dots,M-1$ ,  $M$  adalah panjang *frame*. Gambar 3 menjelaskan tentang hasil proses *windowing* menggunakan *Hanning window*.



Gambar 3. Ilustrasi *windowing* (Proakis dan Manolakis, 1996)

### 2.4 Fast Fourier Transform (FFT)

*Fast Fourier Transform* adalah pengembangan dari algoritma *Discrete Fourier Transform* (DFT) yang digunakan untuk mengubah sinyal digital pada ranah waktu ke ranah frekuensi (Abriyono dan Harjoko, 2012). Sinyal ini pada dasarnya merepresentasikan dekomposisi sinyal dalam hal komponen sinusoidal. Sinusoidal adalah sinusoid dari frekuensi yang sama tetapi amplitudo dan *fase* yang berbeda. FFT adalah algoritma yang dikembangkan oleh Cooley dan Turki merupakan proses signal dari ranah waktu menjadi ranah frekuensi.

Transformasi *fourier* menurut Proakis dan Manolakis (1996) salah satu dari beberapa alat matematika yang berguna dalam analisis dan desain sistem *Linier Time Invarian* (LTI) dan deret *fourier*.

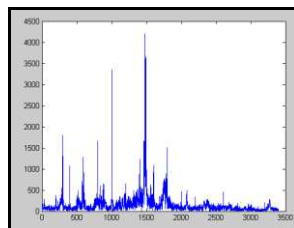
FFT adalah salah satu metode algoritma cepat untuk dapat mengimplementasikan *Discrete Fourier Transform* (DFT). DFT menurut Proakis dan Manolakis (1996) alat komputasi yang memainkan peran yang sangat penting dalam banyak aplikasi pemrosesan sinyal digital, seperti analisis frekuensi, estimasi spektrum daya, dan *filter linear*. Waktu komputasi DFT terlalu lama dan tidak efisien kemudian FFT dapat melakukan efisiensi perhitungan. Seperti yang dikatakan menurut Proakis dan Manolakis (1996) bahwa metode FFT digunakan sebagai cara yang efisien untuk dapat menghitung DFT. *Discrete Fourier Transform* (DFT) menggunakan persamaan (2.5)

$$d[m] = \sum_{n=0}^{N-1} X(n)e^{-j\frac{2\pi}{N}nm}; m = 0,1,2,\dots, N-1. \dots\dots\dots(2.5)$$

Dalam hal ini,  $d[k]$  adalah hasil perhitungan DFT, simbol  $X(n)$  adalah hasil *windowing*. Simbol  $N$  adalah bilangan natural,  $N$  adalah jumlah sampel yang akan diproses ( $N \in \mathbb{N}$ ). Simbol  $k$  adalah variabel *frequency discrete* bernilai ( $m=N/2, m \in \mathbb{N}$ ). *Fast Fourier Transform* mempunyai tujuan mendekomposisi sinyal menjadi sinyal sinusoidal yang berupa unit *real* dan *unit imajiner*. *Fast Fourier Transform* menggunakan persamaan (3.6)

$$T(m) = \sum_{n=0}^{N-1} X(n) \cos\left(\frac{2\pi mn}{N}\right) - \sum_{n=0}^{N-1} X(n) \sin\left(\frac{2\pi mn}{N}\right) \dots\dots\dots(2.6)$$

Dalam hal ini, fungsi  $T(m)$  adalah hasil perhitungan *Fast Fourier Transform* ke- $m$ , simbol  $X(n)$  adalah hasil perhitungan *windowing* ke- $n$ . Simbol  $n$  adalah nomor urut sinyal. Simbol  $m$  adalah indeks dari frekuensi ( $1,2,\dots,N$ ). Gambar 4 menunjukkan hasil spektrum dengan FFT.



Gambar 4. FFT menghasilkan spektrum (Kumar, 2013)

### 2.5 Mel Frequency Wrapping (MFW)

*Mel Frequency Wrapping* (MFW) merupakan *filter* berupa *filterbank* untuk mengetahui ukuran energi dari *frequency band* tertentu dalam signal suara (Miftahuddin dan Hakim, 2017; Putra, 2011). MFW menurut Laha (2007) mengubah frekuensi ke dalam mel.

Sedangkan MFW menurut Tshilidzi Marwala (2012) berisi *filterbank*, yang diberi jarak pada skala mel. *Filterbank* memiliki respon frekuensi lewat jalur segitiga yang jarak dan besarnya ditentukan oleh interval frekuensi yang konstan. Proses *output* yang diperoleh dari *filter* dikenal

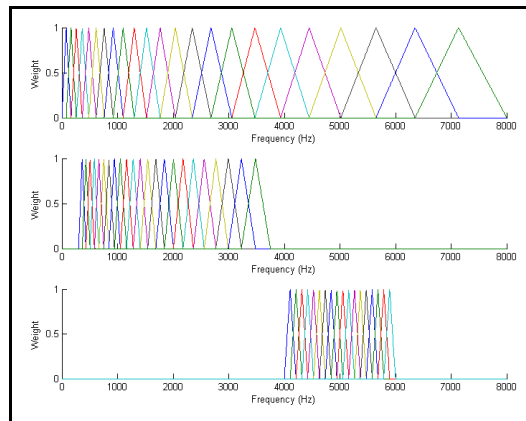
dengan spektrum mel. MFW mempunyai tujuan menghasilkan spektrum mel menggunakan persamaan (2.7)

$$Y[i] = \sum_{j=1}^G T[j]H_r[i,j] \quad (2.7)$$

Dalam hal ini,  $Y[i]$  adalah hasil perhitungan *mel frequency wrapping* ke- $i$  dengan  $G$  adalah jumlah *magnitude spectrum* ( $G \in \mathbb{N}$ ), simbol  $T[j]$  adalah hasil FFT,  $H_r[i,j]$  adalah *coefficient filterbank* pada frekuensi  $j$  ( $1 \leq i \leq E$ ), dan  $E$  adalah jumlah *channel* dalam *filterbank*. Pendekatan yang digunakan dalam bentuk mel menggunakan persamaan (2.8)

$$mel(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (2.8)$$

Dalam hal ini, mel menggunakan frekuensi dengan skala mel,  $f$  sebagai *frequency*. MFW menghasilkan berupa spektrum mel. *Mel frequency scale* adalah *linear* frekuensi skala pada frekuensi di bawah 1.000 Hz dan merupakan *logarithmic* skala pada frekuensi di atas 1.000 Hz (Putra, 2011). Gambar 5 menunjukkan proses spektrum mel.



Gambar 5. Spektrum mel (Hidayat dkk., 2015; Hassan dkk., 2007)

Gambar 5 menunjukkan skala mel dibuat dari *filter bank* menggunakan jenis *filter triangular* dalam warna sedangkan *weight* dalam hal ini *db* merupakan *amplitude*.

## 2.6 Discrete Cosine Transform (DCT)

DCT menurut Smith (2000) merupakan kerabat dari transformasi *fourier* terdekomposisi sinyal ke gelombang *cosinus*. DCT juga banyak dipakai untuk mengolah *image processing* misalkan file JPEG. Pada dasarnya konsep dari DCT memiliki kesamaan dengan *inverse fourier transform*. Namun hasil dari DCT adalah mendekati *Principle Component Analysis* (PCA). PCA adalah metode *static* klasik yang digunakan secara luas dalam analisis data dan kompresi.

DCT diasumsikan menggantikan *inverse fourier transform* dalam proses ekstraksi ciri MFCC (Putra, 2011). *Discrete Cosine Transforms* (DCT) adalah anggota dari kelas transformasi unit sinusoidal (Britanak dkk., 2007). DCT mempunyai tujuan menghasilkan seprtrum mel untuk meningkatkan kualitas pengenalan. DCT menggunakan persamaan (2.9)

$$C_r = \sum_{k=1}^K (\log_{10} Y[i] \cos \left[ r \left( i - \frac{1}{2} \right) \frac{\pi}{K} \right]); r = 1, 2, \dots, K. \quad (2.9)$$

Dalam hal ini,  $C_m$  adalah *Coefficient*, di mana  $Y[i]$  adalah keluaran dari proses *filterbank* pada indek,  $r$  adalah banyaknya *coefficient* dan  $K$  merupakan jumlah *coefficient* yang diharapkan. Proses DCT menghasilkan seprtrum mel.

**2.7 Cepstral Liftering**

Cepstral coefficient menurut Proakis dan Manolakis (1996) merupakan *the fourier series coefficients*. Septrum didefinisikan oleh Rabiner dan Schafer (2007) adalah kebalikannya transformasi *fourier*. Cepstral liftering meningkatkan akurasi digunakan untuk *pattern matching*, baik *speaker recognition* maupun *speech recognition* (Putra, 2011). Cepstral coefficient menggunakan persamaan (3.10)

$$w(k) = 1 + \frac{C}{2} \sin\left(\frac{b\pi}{C}\right); b = 1, 2, \dots, C \dots\dots\dots(2.10)$$

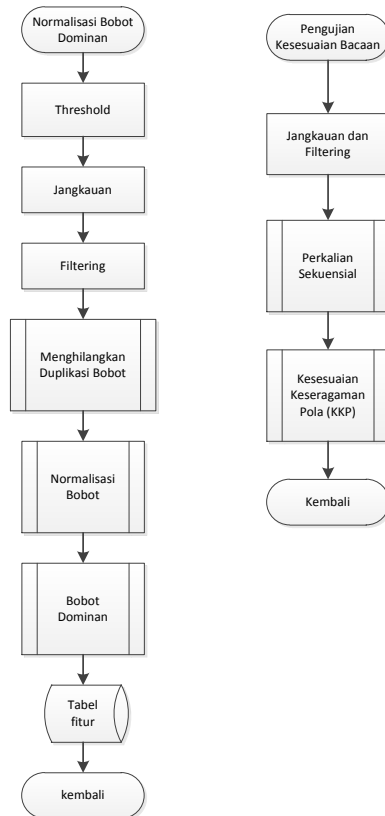
Dalam hal ini,  $w(k)$  adalah fungsi *window* terhadap *cepstral features*,  $C$  adalah *cepstral coefficients*, simbol  $k$  adalah index dari *cepstral coefficients*. Pengolahan *cepstral liftering* mempunyai hasil berupa *frame* dan *cepstral coefficients* selanjutnya diproses ke pemilihan fitur.

**2.8 Pemilihan fitur.**

Pemilihan fitur diperlukan, karena dalam hal ini fitur yang dihasilkan dari ekstraksi ciri MFCC sangat berpengaruh pada pencocokan pengenalan suara (Heriyanto, 2018).

**3. HASIL DAN PEMBAHASAN**

Penggunaan data dengan merekam format wav untuk sumber satu referensi. 44 Setelah dilakukan ekstraksi ciri menggunakan MFCC dilakukan pemilihan fitur. Pemilihan fitur menggunakan model Normalisasi Bobot Dominan (NBD). Model NBD mempunyai enam tahap, yaitu menentukan *threshold*, membuat jangkauan, *filtering*, menghilangkan duplikasi bobot, normalisasi bobot dan bobot dominan yang menghasilkan tabel fitur (Heriyanto, 2018). Model NBD diambil dari pembobotan pada Neural Network (NN) dimana hasil dari skema pelatihan yang berupa bobot JST digunakan kembali pada skema pengujian (Abriyono dan Harjoko, 2012).



**Gambar 6. Normalisasi Bobot Dominan dan pengujian kesesuaian**

### 3.1 Menentukan *Threshold*

Model NBD dimulai dengan ekstraksi ciri MFCC yang menghasilkan fitur berupa *frame* dan *cepstral coefficient*. Fitur tersebut yang digunakan untuk *threshold*. *Threshold* menjadi pemisah antar satu bagian dengan bagian yang lain. Penentuan *threshold* menggunakan persamaan (5.1) hingga (5.6).

Penentuan *threshold* berdasarkan minimum (awal) dari *cepstral coefficient*, maksimum (akhir) juga dari *cepstral coefficient*, minimum ditambah maksimum dibagi dua atau median, rata-rata, minimum ditambah median dibagi dua dan maksimum ditambah median dibagi dua yang kesemuanya dari *cepstral coefficient*.

*Threshold* digunakan sebagai fitur untuk memisah batas dari masing-masing *cepstral coefficient* yang tersebar dalam *frames*. *Cepstral coefficient* yang bervariasi dan banyak perlu dilakukan pemisahan atau pengelompokan mulai dari *threshold* yang sama, menetapkan minimum dan maksimum. Pertama-tama ditetapkan *threshold* untuk nilai *min* (minimum) dan *max* (maksimum) dari *cepstral coefficient* (*c*).

*Threshold* menggunakan persamaan (3.1) sampai (3.6)

$$b_1 = \min(w(k)), \dots \dots \dots (3.1)$$

$$b_2 = \frac{\min(w(k)) + \left(\frac{\min(w(k)) + \max(w(k))}{z}\right)}{2} \dots \dots \dots (3.2)$$

$$b_3 = \frac{\min(w(k)) + \max(w(k))}{2} \dots \dots \dots (3.3)$$

$$b_4 = \text{rata-rata}(w(k)) \dots \dots \dots (3.4)$$

$$b_5 = \frac{\left(\frac{\min(w(k)) + \max(w(k))}{z}\right) + \max(w(k))}{2} \dots \dots \dots (3.5)$$

$$b_6 = \max(w(k)) \dots \dots \dots (3.6)$$

### 3.2 Pembuatan Jangkauan

Pembuatan jangkauan dilakukan berdasarkan *threshold* yang sudah terbentuk. Penentuan jangkauan dimulai dari *threshold* ke-1 sampai dengan *threshold* ke-6 ( $b_1$  sampai dengan  $b_6$ ). Penentuan jangkauan menghasilkan beberapa kondisi yang terbentuk sehingga menjadi aturan atau *rule*.

Membuat Jangkauan	
Jangkauan dengan cek kondisi ada di setiap <i>frame</i> $i=0,1,\dots,F$ .	
dengan <i>cepstral coefficient</i> ( $c$ )= $w(k)$ Bobot=1	
1. Kondisi $r_1$	adalah if ( $b_1$ ) $\min=w(k)$ then $p_{i1}$ =bobot elseif bobot=0
2. Kondisi $r_2$	adalah if ( $b_1$ ) $\geq w(k)$ and ( $w(k) < b_2$ ) then $p_{i2}$ =bobot elseif bobot=0
3. Kondisi $r_3$	adalah if ( $b_2$ ) $\geq w(k)$ and ( $w(k) < b_3$ ) then $p_{i3}$ =bobot elseif bobot=0
4. Kondisi $r_4$	adalah if ( $b_3$ ) $\geq w(k)$ and ( $w(k) < b_4$ ) then $p_{i4}$ =bobot elseif bobot=0
5. Kondisi $r_5$	adalah if ( $b_4$ ) $\geq w(k)$ and ( $w(k) < b_5$ ) then $p_{i5}$ =bobot elseif bobot=0
6. Kondisi $r_6$	adalah if ( $b_5$ ) $\geq w(k)$ and ( $w(k) < b_6$ ) then $p_{i6}$ =bobot elseif bobot=0
7. Kondisi $r_7$	adalah if ( $b_6$ ) $\max= w(k)$ then $p_{i7}$ =bobot elseif bobot=0

**Gambar 4.6.** Kondisi  $r_1$  sampai  $r_7$  atau *rule* dengan pengecekan *cepstral coefficients* ( $c$ ) atau  $w(k)$  dari jangkauan

Gambar 4.6. menjelaskan kondisi  $r_1$  sampai  $r_7$  atau *rule* dengan pengecekan *cepstral coefficients* ( $c$ ) atau  $w(k)$  dari jangkauan menggunakan bobot yang dituliskan dengan simbol  $p_{i1}$  sampai  $p_{i7}$  menjadi bobot sama dengan satu, jika memenuhi *rule*. Pembuatan jangkauan bertujuan membuat *rule* yang digunakan untuk *filtering* setiap *frame* dari  $i=0,1,\dots,F$ .

**3.3. Filtering**

*Filtering* mempunyai tujuan untuk memisahkan atau memilah *cepstral coefficient* menjadi bobot. Pemilahan menggunakan aturan “jika memenuhi sesuai jangkauan atau *range*, maka bobot menjadi satu dan terakumulasi di masing-masing  $p_1$  sampai dengan  $p_7$ ”.

Penjumlahan hasil *filtering* menggunakan persamaan (3.7) dan (3.8)

$$G_j = \sum_{i=0}^F p_{ij}, \dots\dots\dots (3.7)$$

$$U = \sum_{j=1}^a G_j, \dots\dots\dots (3.8)$$

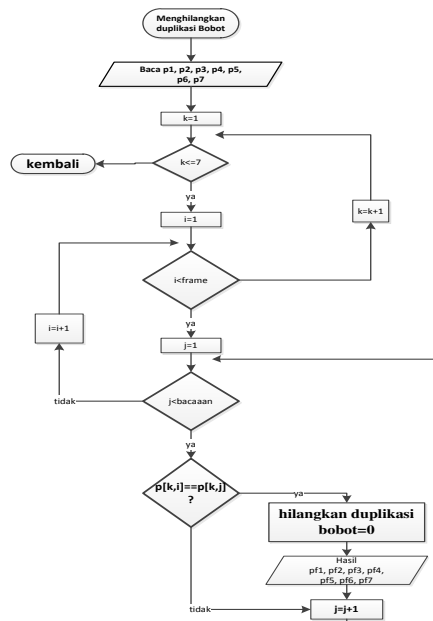
Dalam hal ini,  $U$  adalah jumlah keseluruhan  $G_j$ . Simbol  $G_j$  adalah hasil jumlah  $p_{ij}$ , keseluruhan, simbol  $j$  adalah 1 sampai 7, sedangkan *Frame* yang dibentuk berupa  $i$  adalah *frame* ke-0 sampai *frame* ke- $F$ . Simbol  $p_{ij}$  adalah hasil *filtering* atau akumulasi bobot. Simbol  $a=1,2,\dots,7$ .

**3.4 Menghilangkan Duplikasi Bobot**

Duplikasi bobot dihilangkan dengan cara mencari adanya kesamaan dengan algoritma pencarian yang sama pada masing-masing  $p_1$  sampai  $p_7$  yang telah terakumulasi bobot. Simbol  $k_p$  merupakan keseluruhan jumlah masing-masing  $p_1$  sampai  $p_7$ . Duplikasi bobot dihilangkan kemudian dilakukan perhitungan jumlah bobot menggunakan persamaan (3.9)

$$Z_j = \sum_{i=0}^F Q_{ij} \dots\dots\dots (3.9)$$

Dalam hal ini simbol  $Z$  adalah perhitungan jumlah bobot  $Q$ . Bobot  $Q_{ij}$  adalah hasil *filtering* bobot yang sudah dihilangkan duplikasinya,  $j$  adalah 1 sampai 7,  $i$  adalah *frame* ke-0 sampai *frame* ke- $F$ .



Gambar 7. Proses menghilangkan duplikasi bobot



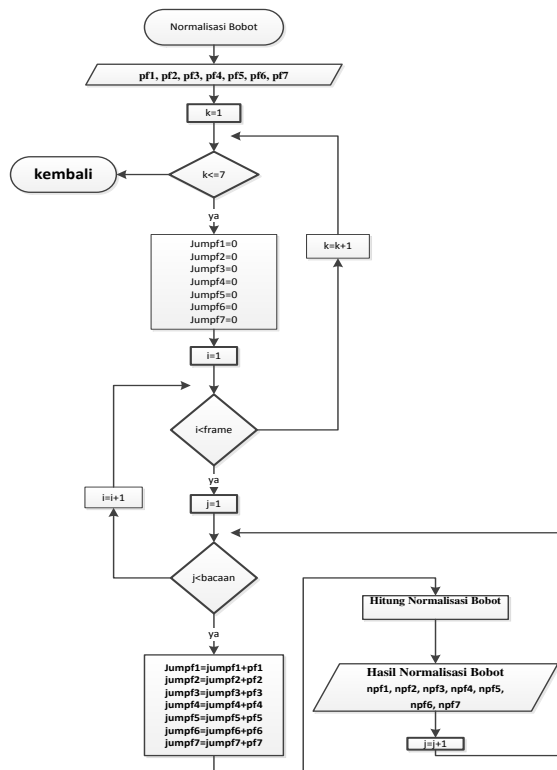
Gambar 7 menunjukkan proses penghilangan duplikasi dimulai dengan *filtering* dari  $p_1$  sampai  $p_7$  ( $P_{i1}$  sampai  $P_{i7}$ ). Pencarian duplikasi bobot dilakukan dengan membandingkan  $p_1$  sampai  $p_7$  yang sama. Contoh, bacaan “iqra” pada *frame* ke-0 dibandingkan dengan bacaan “kholaqq” pada *frame* ke-0, “jika terdapat kesamaan bobot, maka dihilangkan menjadi bobot 0”.

**3.5 Normalisasi Bobot**

Setelah proses duplikasi bobot kemudian dilakukan kesetaraan terhadap bacaan dari masing-masing bacaan yang ada pada tabel fitur. Perlunya dilakukan normalisasi agar setiap bacaan mempunyai standar yang sama atau sederajat. Perhitungan Normalisasi bobot menggunakan persamaan (5.10)

$$S_j = \sum_{i=0}^F \frac{Q_{ij}}{Z_j} \dots\dots\dots (3.10)$$

Dalam hal ini,  $S$  adalah hasil normalisasi bobot. Simbol  $j$  dengan  $j=2, 3, 5$  dan  $6$ , sedangkan  $Z$  adalah perhitungan jumlah bobot. Bobot  $Q_{ij}$  adalah hasil *filtering* bobot yang sudah dihilangkan duplikasinya. Simbol  $i$  adalah *frame*. *Frame* yang diambil mulai dari  $i=0$  sampai  $F$ .



**Gambar 8. Proses normalisasi bobot**

Gambar 8 menunjukkan proses normalisasi bobot yang dimulai dari menghilangkan duplikasi bobot sampai dilakukan normalisasi. Duplikasi bobot yang sudah hilang selanjutnya dilakukan perhitungan jumlah bobot pada masing-masing  $pf$  menjadi normalisasi bobot  $npf$  menggunakan persamaan (3.9).

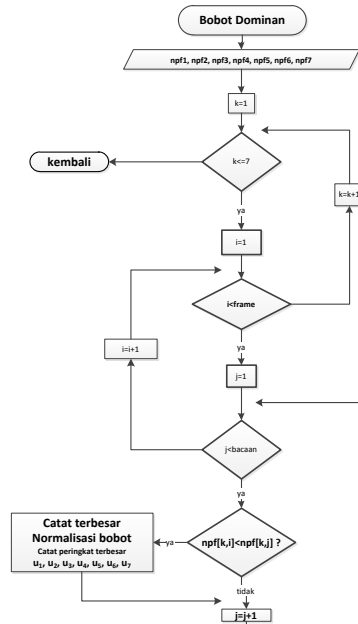
Sebelum dilakukan normalisasi bobot maka dilakukan perhitungan jumlah bobot  $pf_1$  sampai dengan  $pf_7$ . Perhitungan jumlah bobot tersebut dilakukan karena pada saat normalisasi diperlukan pembagi jumlah yang sama.

Selanjutnya dari masing-masing *frame* yang terdapat  $pf_1$  sampai dengan  $pf_7$  dibagi dengan  $pb$  yang kemudian disimpan menjadi  $npf_2, npf_3, npf_5,$  dan  $npf_6$ . Simbol  $pb$  adalah perhitungan

jumlah bobot  $pf$ . Simbol  $pf$  adalah hasil pencocokan kesamaan duplikasi bobot yang dihilangkan.

**3.6 Bobot Dominan**

Setelah normalisasi bobot terbentuk, kemudian dicari bobot dominan dengan cara mencari daerah mana saja yang unggul dan menjadi dominan dari masing-masing *frame* yang berbeda dengan mengurutkan terbesar sampai yang terkecil.



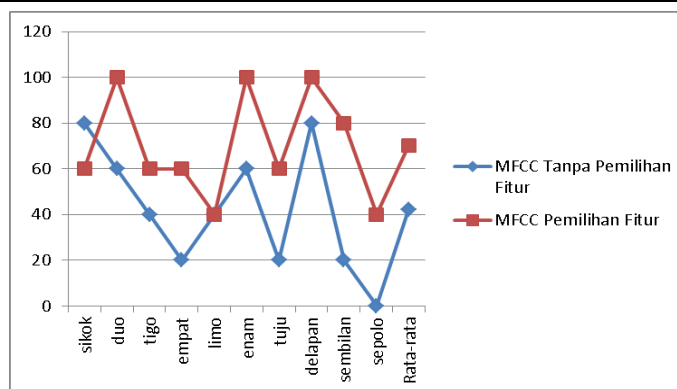
**Gambar 9. Proses pencarian bobot dominan**

Gambar 9 menunjukkan penentuan bobot dominan dengan cara mencari nilai yang tertinggi sampai dengan yang terendah yang bukan dominan. Pencarian bobot dominan dilakukan pengurutan atau sortir dengan *descending* nilai terbesar di peringkat atas yang dijadikan pemilihan fitur yang tepat sebagai tabel fitur.

Setelah selesai dilakukan sortir, maka dilakukan pemeringkatan dengan nilai tertinggi parameter  $u_1, u_2$  dan seterusnya. Hasil dari proses pencarian bobot dominan menjadi tabel fitur. Hasil tabel fitur kemudian dilakukan pengujian. Berikut tabel hasil dari pengecekan ucapan bahasa Palembang satu sampai sepuluh

**Tabel 1. Hasil deteksi ucapan bahasa Palembang angka satu sampai sepuluh**

no	B.Indonesia	B.Palembang	MFCC Tanpa Pemilihan Fitur	MFCC Pemilihan Fitur
1	satu	sikok	80	60
2	dua	duo	60	100
3	tiga	tigo	40	60
4	empat	empat	20	60
5	lima	limo	40	40
6	enam	enam	60	100
7	tujuh	tuju	20	60
8	delapan	delapan	80	100
9	sembilan	sembilan	20	80
10	sepuluh	sepolo	0	40
		Rata-rata	42	70



Gambar 10 Hasil perbandingan MFCC pemilihan fitur dan tanpa pemilihan fitur

## KESIMPULAN

Telah berhasil dilakukan Ekstraksi ciri, dan pemilihan fitur dengan MFCC sebesar 70% dan tanpa pemilihan fitur dengan MFCC sebesar 42%. Pemilihan fitur dengan cara mencari jumlah *cepstral coefficient* dan jumlah *frame* dengan menentukan *threshold*, jangkauan, *filtering*, menghilangkan duplikasi bobot, normalisasi bobot dan bobot dominan.

## DAFTAR PUSTAKA

- Abriyono dan Harjoko, A., 2012. Pengenalan Ucapan Suku Kata Bahasa Lisan Menggunakan Ciri LPC, MFCC, dan JST. *Indonesian Journal of Computing and Cybernetics Systems*, 6(2), hal.23–34.
- Aibinu, A.M., Salami, M.J.E., Najeeb, A.R., Azeez, J.F. dan Rajin, S.M.A.K., 2011a. Evaluating the effect of voice activity detection in isolated Yoruba word recognition system. *2011 4th International Conference on Mechatronics: Integrated Engineering for Industrial and Societal Development, ICOM'11 - Conference Proceedings*, (May), hal.17–19.
- Chamidy, T., 2016. Metode Mel Frequency Cepstral Coefficients (MFCC) Pada klasifikasi Hidden Markov Model (HMM) Untuk Kata Arabic pada Penutur Indonesia. *Matics*, 8(1), hal.36–39. Available at: <http://ejournal.uin-malang.ac.id/index.php/saintek/article/view/3482>.
- Davis, S.B. dan Mermelstein, P., 1980. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), hal.357–366.
- Heriyanto, 2015. Analisa Deteksi Huruf Hijaiyah Melalui Voice Recognition Menggunakan Kombinasi Energy. *Telematika*, 12(1), hal.11–22.
- Heriyanto, dkk 2018, Advanced Image Video Processing-Volume 6 No 2 Apr 2018-AIVP-DOI-DOI: <http://dx.doi.org/10.14738/aivp.62.4268>
- Suyanto, S. dan Putra, A.E., 2014b. Automatic Segmentation of Indonesian Speech into Syllables using Fuzzy Smoothed Energy Contour with Local Normalization, Splitting, and Assimilation. *Journal of ICT Research and Applications*, 8(2), hal.97–112. Available at: [http://journal.itb.ac.id/index.php?li=article\\_detail&id=1804](http://journal.itb.ac.id/index.php?li=article_detail&id=1804).
- Suyanto dan Hartati, S., 2013. Design of Indonesian LVCSR using Combined Phoneme The Approaches of LVCSR. *Icts*, hal.191–196.