PENERAPAN METODE MEL FREQUENCY CEPTRAL COEFFICIENT DAN LEARNING VECTOR QUANTIZATION UNTUK TEXT-DEPENDENT SPEAKER VERIFICATION

SUKORENO MUKTI W - 1112051

LATAR BELAKANG

- Kata sandi dalam bentuk teks dianggap kurang aman karena sering kali terjadi kebocoran. Maka dari itu dibutuhkan bentuk lain dari kata sandi, salah satunya adalah dalam bentuk suara.
- Dibangun sistem pengenalan pembicara untuk identifikasi dan verifikasi yang nanti nya dapat berguna sebagai bentuk kata sandi untuk mengakses sebuah sistem
- Pengenalan dibuat menggunakan metode MFCC sedangkan klasifikasi menggunakan LVQ

RUMUSAN MASALAH

- Bagaimanakah pengaruh nilai parameter-parameter MFCC dan LVQ terhadap hasil identifikasi yang diperoleh?
- Seberapa besar peningkatan akurasi yang dapat diperoleh dengan mengubah parameter-parameter MFCC ?
- Apakah metode MFCC mampu mengekstraksi ciri suara hingga mendapatkan warna suara yang unik dari masing-masing pembicara ?
- Bagaimana hasil dari identifikasi dan verifikasi suara pembicara-kata dengan menggunakan kombinasi metode MFCC dan LVQ ?

TUJUAN PENELITIAN

Untuk mengidentifikasi pembicara melalui pengenalan suara, dan untuk mengetahui performa dari kombinasi metode MFCC dan LVQ untuk identifikasi dan verifikasi suara pembicara.

BATASAN MASALAH

- Kata yang akan digunakan untuk melakukan verifikasi dibatasi hanya berupa buka, kunci, unlock dan lock.
- Aplikasi yang dibuat hanya menerima masukan berupa suara ucapan manusia dan keluarannya merupakan hasil identifikasi pembicara dan kata-kata terdaftar.

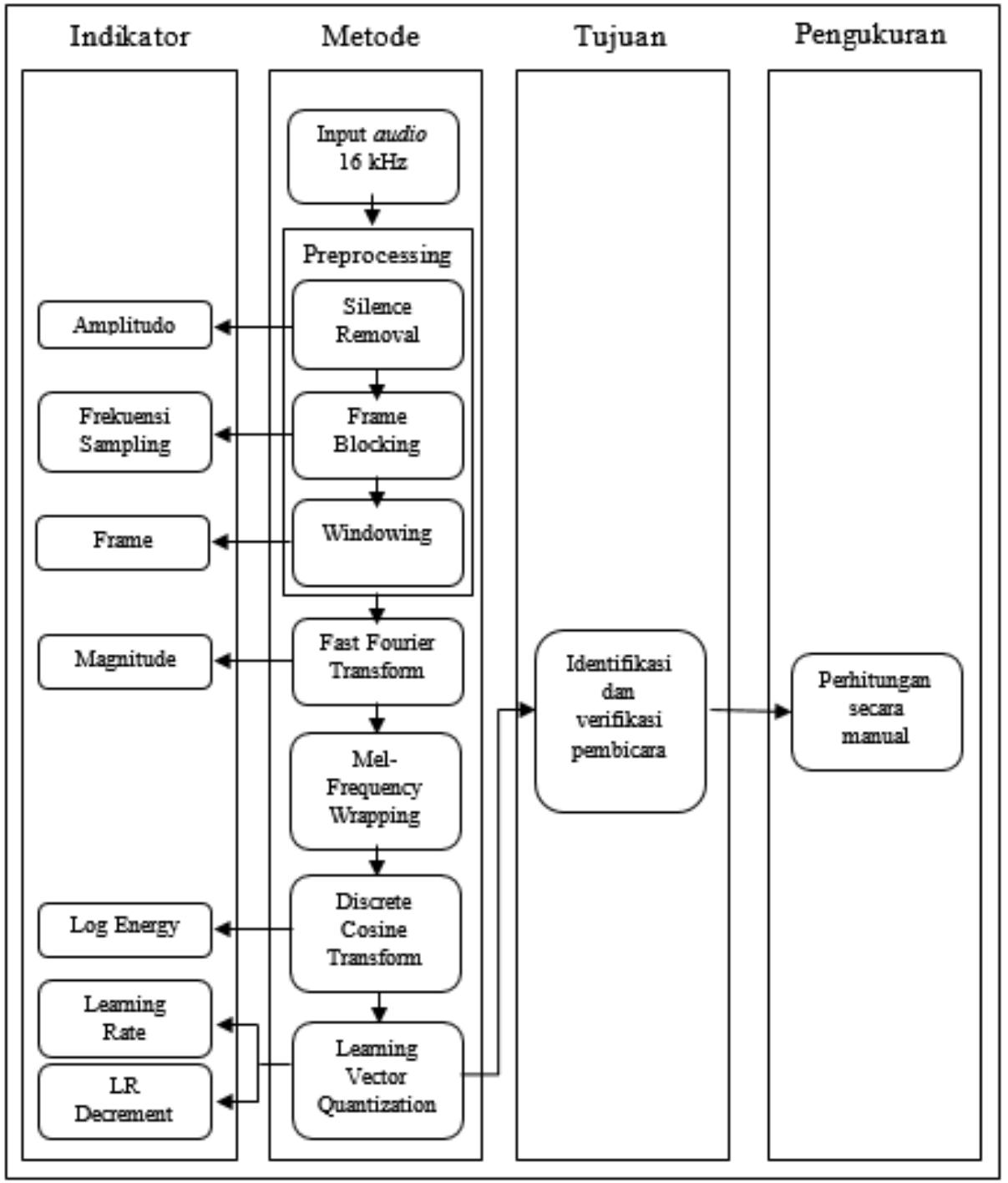
KONTRIBUSI PENELITIAN

Mendapatkan perbaikan akurasi yang lebih baik yang pada penelitian sebelumnya yang dilakukan oleh Daniel Christian T. dengan Judul "Penerapan Metode Mel-Frequency Cepstral Coefficients Dan K-Means Clustering Untuk Pengenalan Pembicara" yang sebelumnya menggunakan metode MFCC dan K-Mean Clustering dan akan dicoba dengan menggunakan kombinasi metode MFCC dan LVQ.

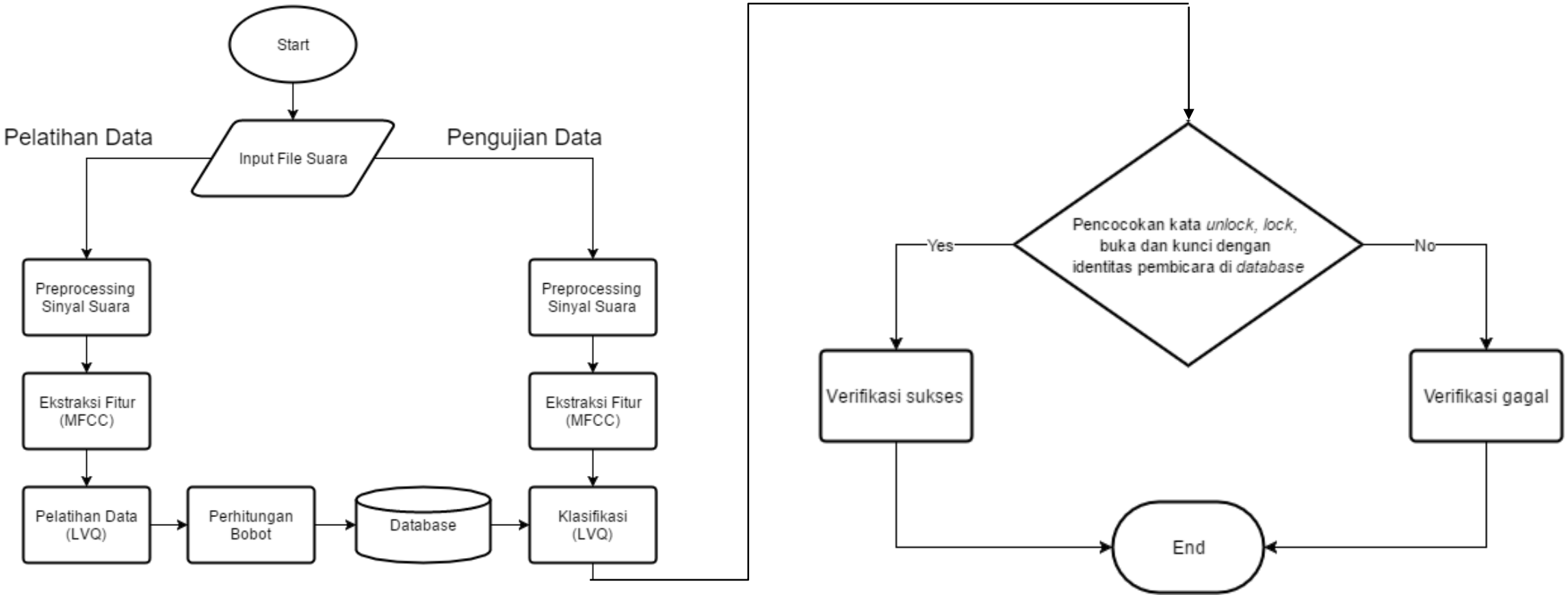
ANALISIS MASALAH

Bagaimana melakukan identifikasi dan verifikasi pembicara melalui sebuah *input* suara dengan ucapan terbatas. Untuk itu digunakan metode MFCC untuk ekstraksi fitur dan LVQ untuk klasifikasi identifikasi pembicara.

MFCC digunakan karena sudah terbukti menghasilkan ekstraksi fitur yang suara yang baik, sedangkan LVQ digunakan karena merupakan salah satu *supervised neural network*. Berdasarkan penelitian oleh Geeta Nijhawan pada tahun 2014, kombinasi MFCC dan LVQ untuk kasus Text Independent Speaker Verification menghasilkan akurasi 95%.


Indikator
Metode
Tujuan
Pengukuran
Input *audio* 16 kHz
Preprocessing
Silence Removal
Frame Blocking
Windowing
Fast Fourier Transform
Mel-Frequency Wrapping
Discrete Cosine Transform
Learning Vector Quantization
Amplitudo
Frekuensi Sampling
Frame
Magnitude
Log Energy
Learning Rate
LR Decrement
Identifikasi dan verifikasi pembicara
Perhitungan secara manual

PERANCANGAN



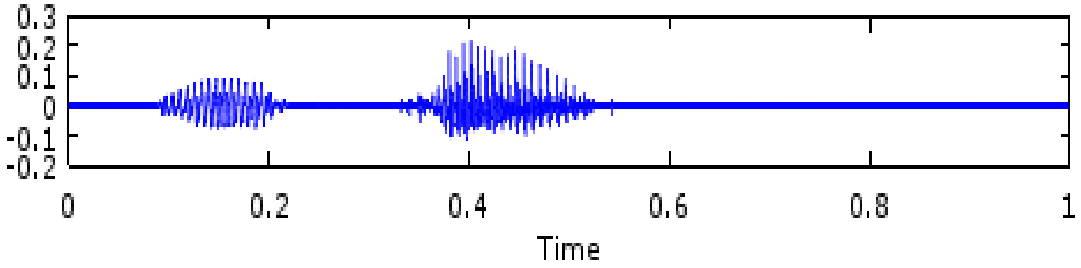
INPUT SUARA

- Berdurasi sekitar 1-3 detik tergantung kata yang diucapkan
- Sampling yang digunakan adalah 16 kHz, agar memenuhi *Nyquist-Shannon Criteria.*
- 5 sampel pembicara yang terdiri dari 3 pria dan 2 wanita (buka, kunci, *lock,* dan *unlock).*
- Setiap kata akan direkam sebanyak 5 kali, sehingga total untuk masing-masing pembicara adalah 20 kali perekaman suara.

INPUT SUARA (Cont.)

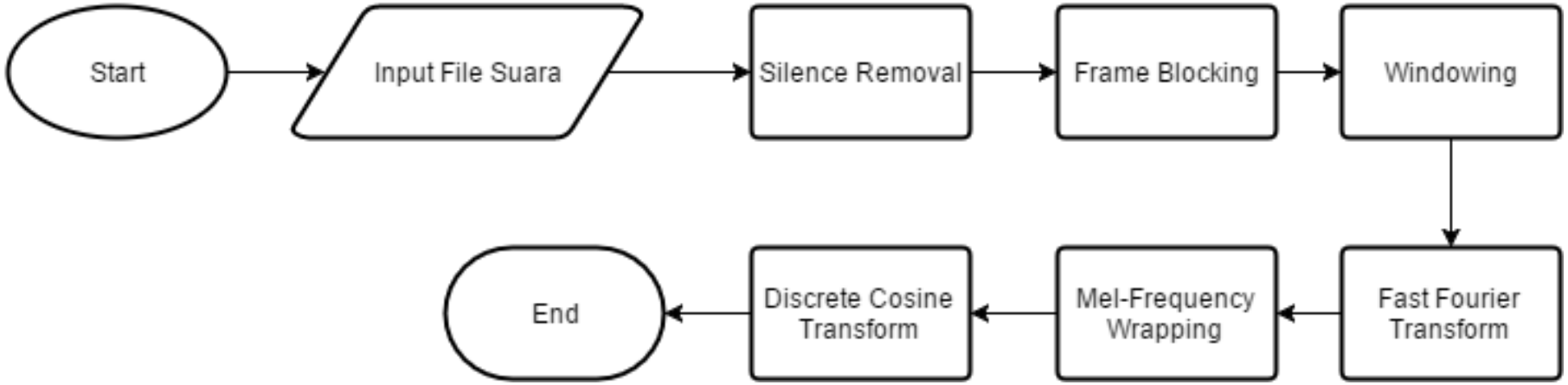
file suara yang digunakan adalah rekaman suara ucapan “buka” sepanjang 2 detik dengan frekuensi *sampling* sebesar 16 kHz. Jumlah sampel yang didapat dari rekaman tersebut adalah sebanyak 15975 sampel.

Sampel ke-	Nilai Sampel
1	0.00006
2	-0.00006
3	0.00012
…	…
15975	-0.00006



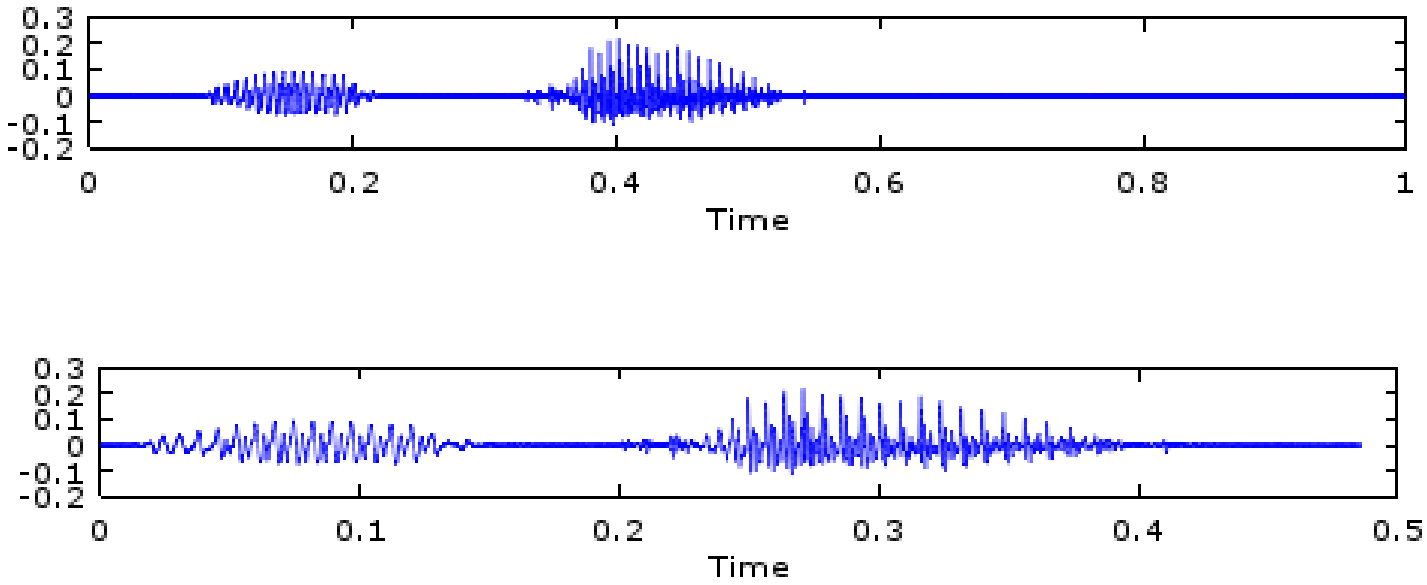
MEL FREQUENCY CEPTRAL COEFFICIENTS

Salah satu metode yang paling popular untuk mengekstraksi fitur suara dengan cara menghitung koefisien *cepstral* berdasarkan variasi frekuensi kritis pada sistem pendengaran manusia.

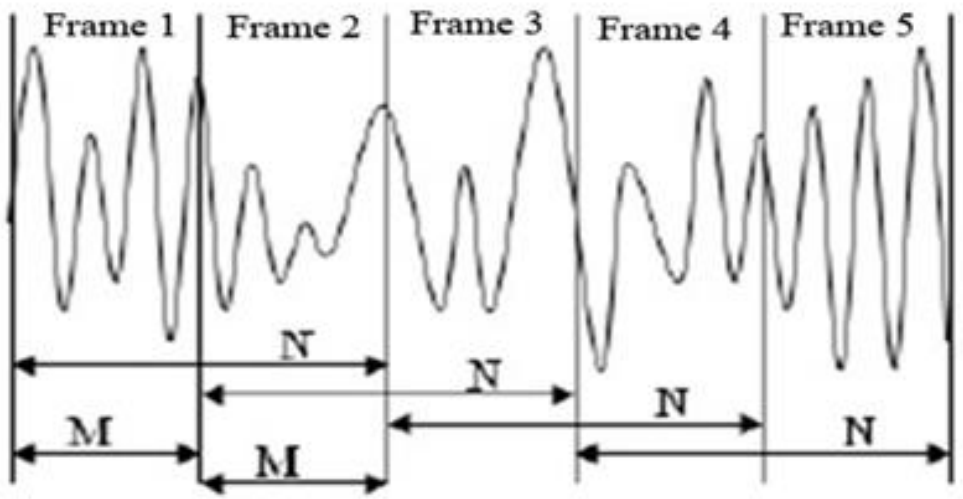


SILENCE REMOVAL

- *file* suara yang bernilai antara -0.00021 sampai 0.00021 akan dihilangkan dan tidak akan di proses karena nilai sampel tersebut dianggap suara diam yang tidak penting untuk ekstraksi fitur.
- sampel yang bernilai -0.00021 sampai 0.00021 ada sebanyak 8201 buah, maka jumlah sampel file audio tersebut akan menjadi 7774 sampel.



FRAME BLOCKING



Sinyal dibagi-bagi menjadi beberapa *frame* dengan panjang pada umumnya sekitar 20-30ms yang berisi M sampel dan masing-masing frame dipisahkan oleh N (M<N) dimana N adalah banyaknya pergeseran antar *frame*. *Frame blocking* perlu dilakukan karena sinyal suara mengalami perubahan dalam jangka waktu tertentu.

FRAME BLOCKING (Cont.)

- Ukuran sampel per *frame* sebesar 256 dan *overlapping* sebesar $1/2$ dari sampel per *frame*.

Jumlah Frame = ((sample-frameSize) / overlap) + 1

= ((7774 – 256) / 128)+ 1

= 59 *frames*

WINDOWING

Windowing dilakukan untuk meminimalisir diskontinuitas yang terjadi pada sinyal, yang disebabkan oleh kebocoran spektral pada saat proses *frame blocking* dilakukan dimana sinyal yang baru, memiliki frekuensi yang berbeda dengan sinyal aslinya.

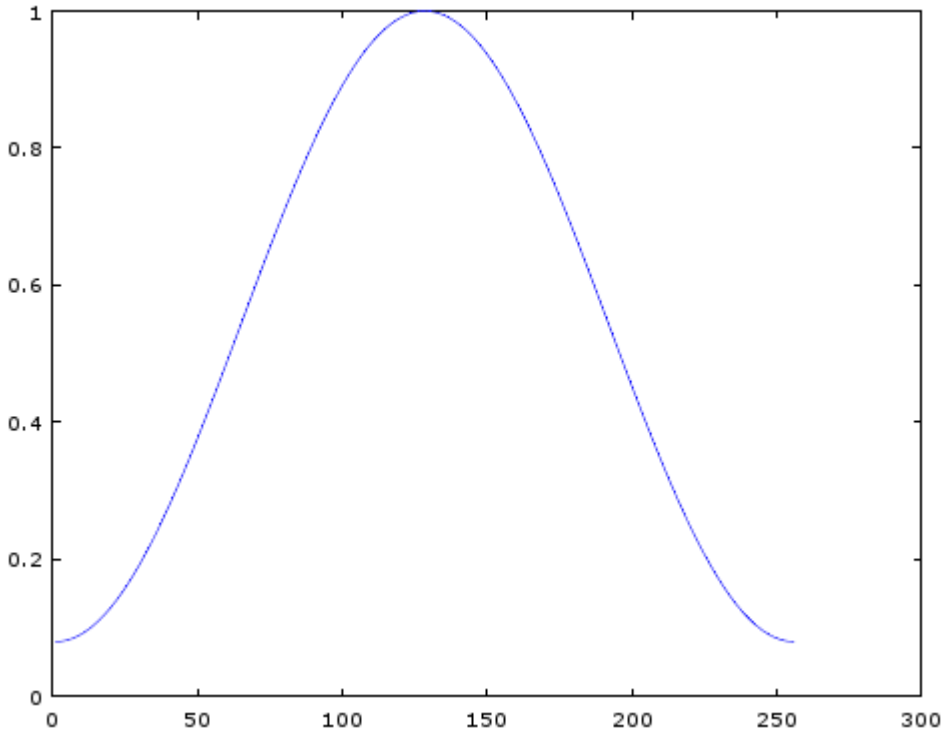
Konsep dari *windowing* adalah meruncingkan ujung sinyal menjadi nol pada bagian awal dan akhir setiap *frame.*

$$y(n) = x(n)w(n), \quad 0 \leq n \leq N-1 \tag{2.1}$$

$$w(n) = 0.54 - 0.46\, cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \leq n \leq N-1 \tag{2.2}$$

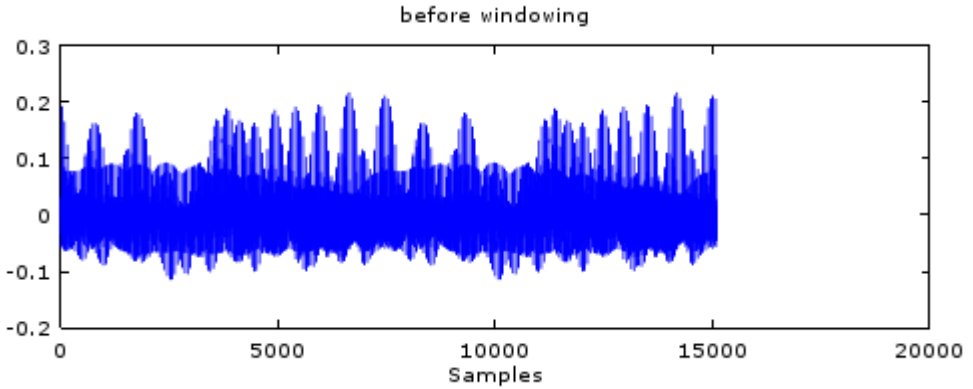
HAMMING WINDOW

Hamming	Nilai Hamming Window
1	0.080000
…	…
128	0.99997
…	…
256	0.080000

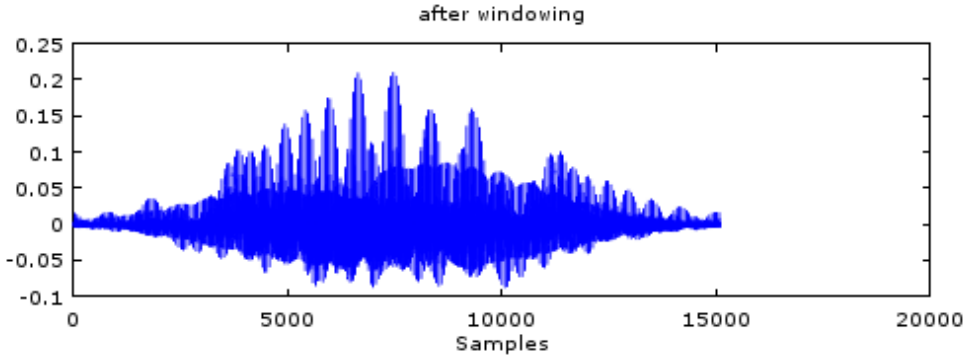


WINDOWING (Cont.)

Frame ke-	Sampel ke-	Nilai Sampel
1	1	-0.00021362
…	…	…
1	256	-0.0024719



Frame ke-	Sampel ke-	Nilai setelah *windowing*
1	1	-0.000017
…	…	…
1	256	-0.00019775



DFT & FFT

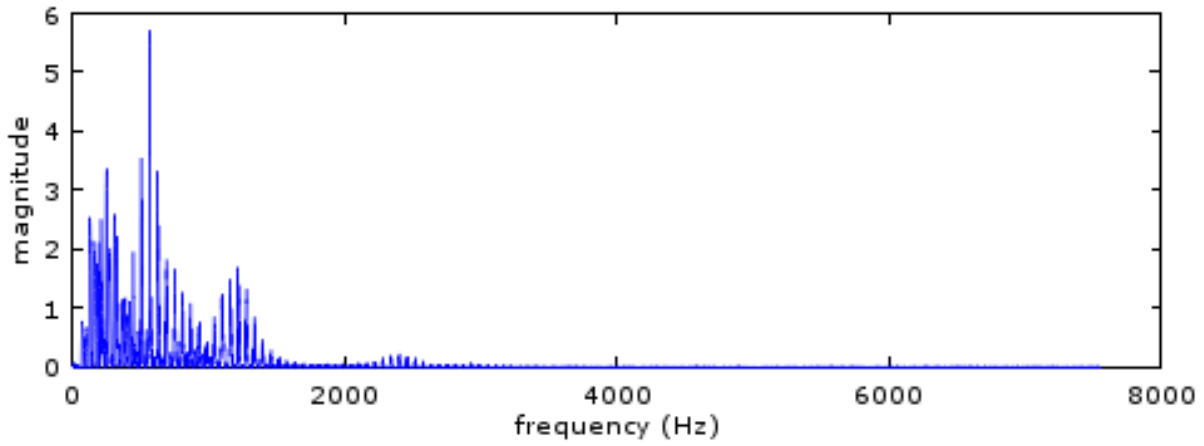
- Discrete Fourier Transform (DFT) adalah prosedur yang digunakan dalam pemrosesan sinyal digital dan filterisasi digital. DFT memungkinkan kita untuk menganalisa, memanipulasi dan mensintesis sinyal. Mengubah sinyal dari domain waktu ke domain frekuensi. DFT membutuhkan operasi $O(N^2)$ FFT mengurangi operasi komputasinya menjadi $O(N \log_2(N))$

$$X(m) = \sum_{n=0}^{N-1} x(n)[cos\left(\frac{2\pi nm}{N}\right) - jsin\left(\frac{2\pi nm}{N}\right)] \quad (2.5)$$

$$Xmag(m) = |X(m)| = \sqrt{X_{real}(m)^2 + X_{imag}(m)^2} \quad (2.7)$$

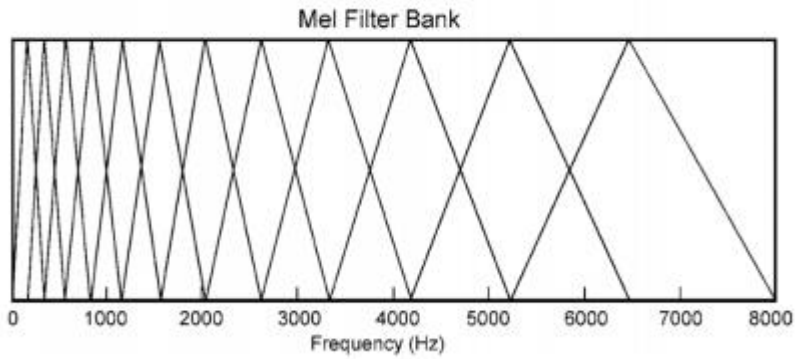
DFT & FFT (Cont.)

Frame ke-	Sampel ke-	Nilai Magnitude
1	1	0.0015689
1	2	0.0036518
…	…	…
1	128	0.0036518



MEL-FREQUENCY WRAPPING

- Mel-Frequency Wrapping menggunakan Filterbank untuk menyaring sinyal suara.
- Filterbank adalah sistem yang membagi input sinyal ke dalam kumpulan analisis sinyal, yang masing-masing sesuai dengan wilayah yang berbeda sesuai spectrum.
- mel-filterbank yang terdiri dari rangkaian Triangular Window yang saling overlap akan menyaring sinyal sebanyak N sampel.



$$mel(f) = 1125 \ln(1 + \frac{f}{700}) \quad (2.10)$$

$$mel^{-1}(f) = 700 \left(\exp\left(\frac{mel(f)}{1125}\right) - 1\right) \quad (2.11)$$

MEL-FREQUENCY WRAPPING (Cont.)

$$f[m] = \left(\frac{N}{F_s}\right) mel^{-1}\left(mel(f_i) + m\frac{mel(f_h) - mel(f_i)}{M+1}\right) \quad (2.12)$$

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{k - f[m-1]}{f[m] - f[m-1]} & f[m-1] \le k \le f[m] \\ \dfrac{f[m+1] - k}{f[m+1] - f[m]} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \quad (2.13)$$

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right], \qquad 1 \le m \le M \quad (2.14)$$

MEL-FREQUENCY WRAPPING (Cont.)

- Jumlah *filter* yang digunakan adalah 32 *filter*.
- Batas bawah adalah 300 Hz dan batas atas adalah Frekuensi *Sampling* / 2 yaitu 8000 Hz.
- *mel filterbank* memiliki *triangular window* sebanyak 32 buah, dan memiliki 34 titik.

Mel ke-	Nilai Mel	Nilai Frekuensi	FFT Bin
1	401.26	300	4
…	…	…	…
34	2835	8000	128

frame ke-	*filter ke-*	Nilai *log energy*
1	1	0.028768
1	32	0.0075837

DISCRETE COSINE TRANSFORM

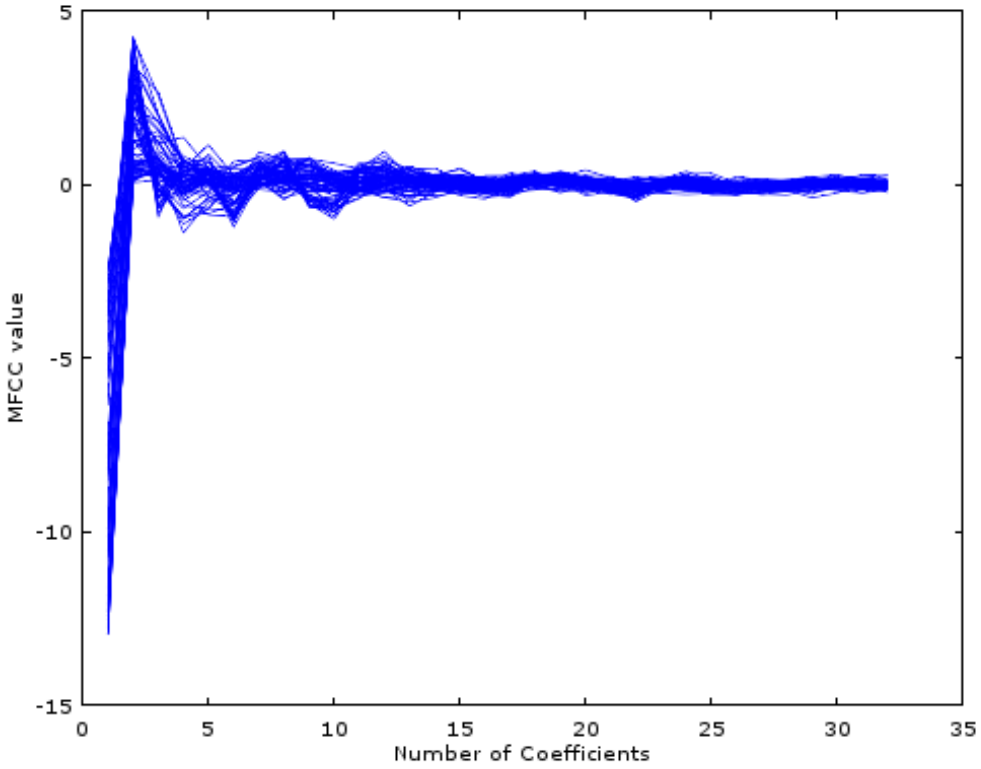
- Nilai mel akan dikonversikan kembali ke dalam domain waktu, yang hasilnya disebut Mel Frequency Ceptral Coefficient.
- Konversi ini dilakukan dengan menggunakan Discrete Cosine Transform (DCT). Koefisien DCT adalah nilai amplitudo dari spektrum yang dihasilkan.

$$c_i = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n\,(m+0.5)}{M}\right) \quad 0 \leq n < M \qquad (2.15)$$

Hasil MFCC

- Jumlah *ceptrum* yang diambil adalah sebanyak 13 buah pada satu *frame*. Dimulai dari index setelah pertama.

frame ke-	DCT *ke-*	Nilai DCT
1	2	0.257731
1	3	0.347545
…	…	…
1	13	0.059163



LEARNING VECTOR QUANTIZATION

- Learning Vector Quantization (LVQ) adalah satu metode untuk melakukan klasifikasi pola yang dimana setiap keluarannya merepresentasikan sebuah kelas atau kategori tertentu.

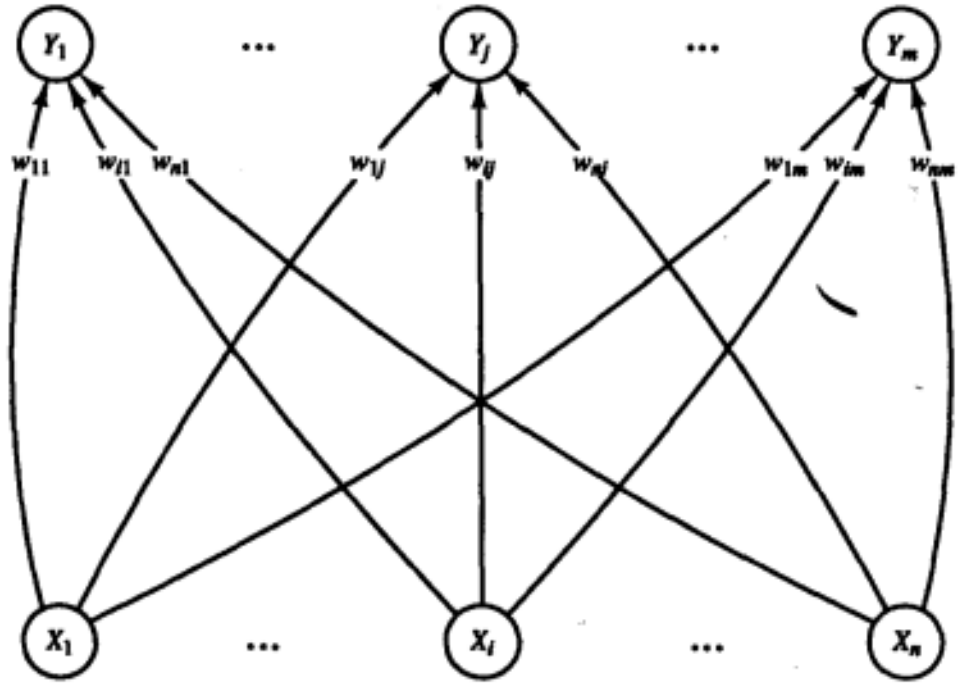
Jika $T = C_j$ maka

$$w_j(baru) = w_j(lama) + \alpha[x - w_j(lama)] \quad (2.19)$$

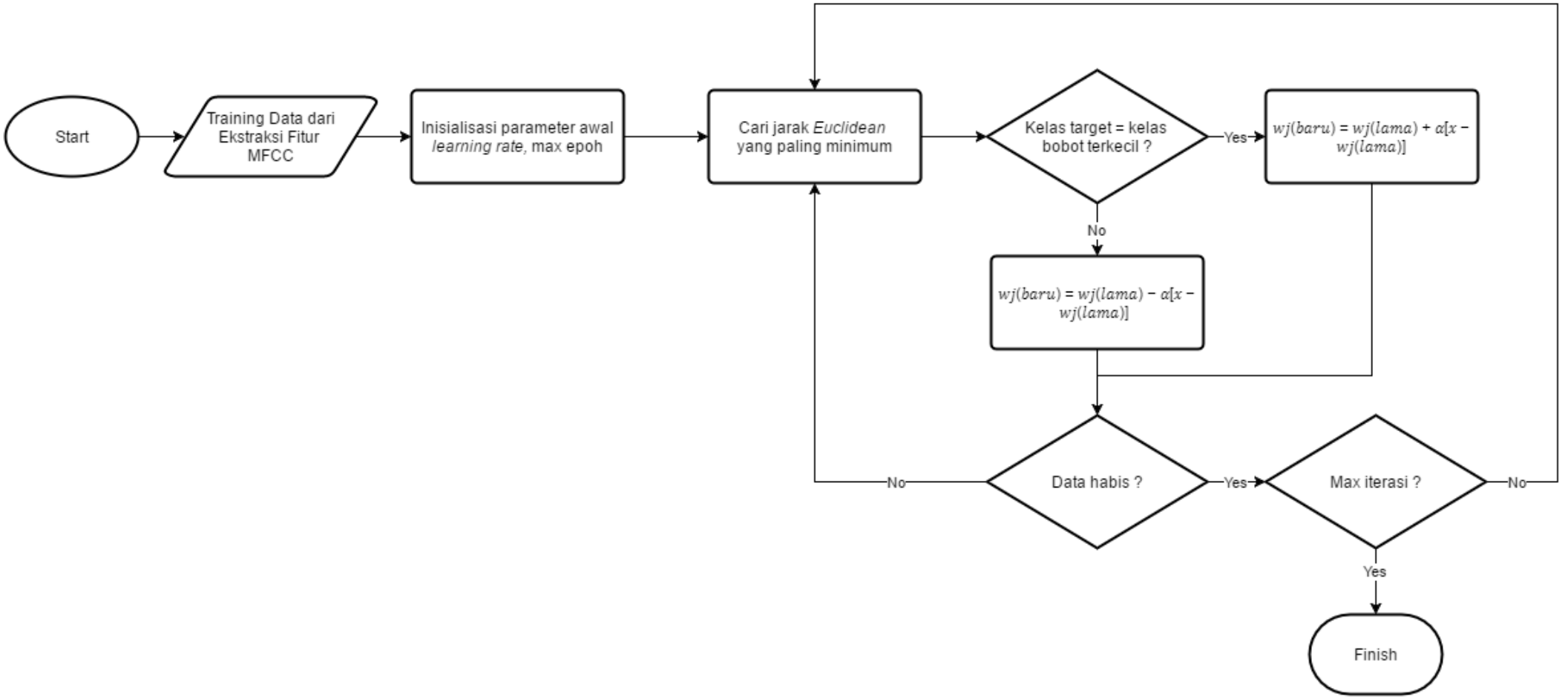
Jika $T \neq C_j$ maka

$$w_j(baru) = w_j(lama) - \alpha[x - w_j(lama)] \quad (2.20)$$

$$\|x - w_j\| = \sqrt{\sum_{i=1}^{n}(x - w_j)^2} \quad (2.21)$$



Flowchart LVQ



Input LVQ (pelatihan data)

- neuron Input ada 13, Output neuron ada 4 yaitu, pembicara-Buka, pembicara-Kunci, pembicara-Unlock, pembicara-Lock.
- Learning rate ditetapkan sebesar 0.05 dan pengurangannya adalah 0.977, maksimum iterasi adalah 1000.
- Masing-masing hasil suara, memiliki jumlah *input* yang berbeda-beda, tergantung dari jumlah fitur yang diperoleh dari hasil ekstraksi fitur.

Input	X_1	X_2	X_3	…	X_{13}	Output
1	-12.554	0.25773	0.34754	…	0.05916	Reno-Buka
60	-12.06	-0.00362	0.27599	…	0.49975	Reno-Kunci
151	-12.235	0.41048	0.73952	…	0.07273	Reno-Lock
297	-12.273	0.19314	0.40462	…	0.34727	Reno-Unlock

LVQ

Tabel bobot awal

No	Vektor	Kelas
1	[-12.554, 0.25773, 0.34754, …,0.05916]	Reno-Buka
2	[-12.060, -0.00362, 0.27599,…, 0.49975]	Reno-Kunci
3	[-12.235, 0.41048, 0.73952, …, 0.07273]	Reno-Lock
4	[-12.8, 0.24217, 0.60348, …, 0.10297]	Reno-Unlock

Tabel bobot akhir

No	vektor	Kelas
1	[-20.087, 0.41237, 0.55607, …,0.09466]	Reno-Buka
2	[-19.296, -0.0058071, 0.44158,…, 0.7996]	Reno-Kunci
3	[-19.576, 0.65676, 1.1832, …, 0.11637]	Reno-Lock
4	[−20.48, 0.38747, 0.96556, …, 0.16475]	Reno-Unlock

LVQ (pengujian data)

Input	X_1	X_2	X_3	…	X_{13}
1	-12.061	0.71374	0.59465	…	0.07847
2	-11.872	0.55795	0.58062	…	0.02105
…	…	…		…	…
90	-12.432	0.37681	0.70664	…	0.01507

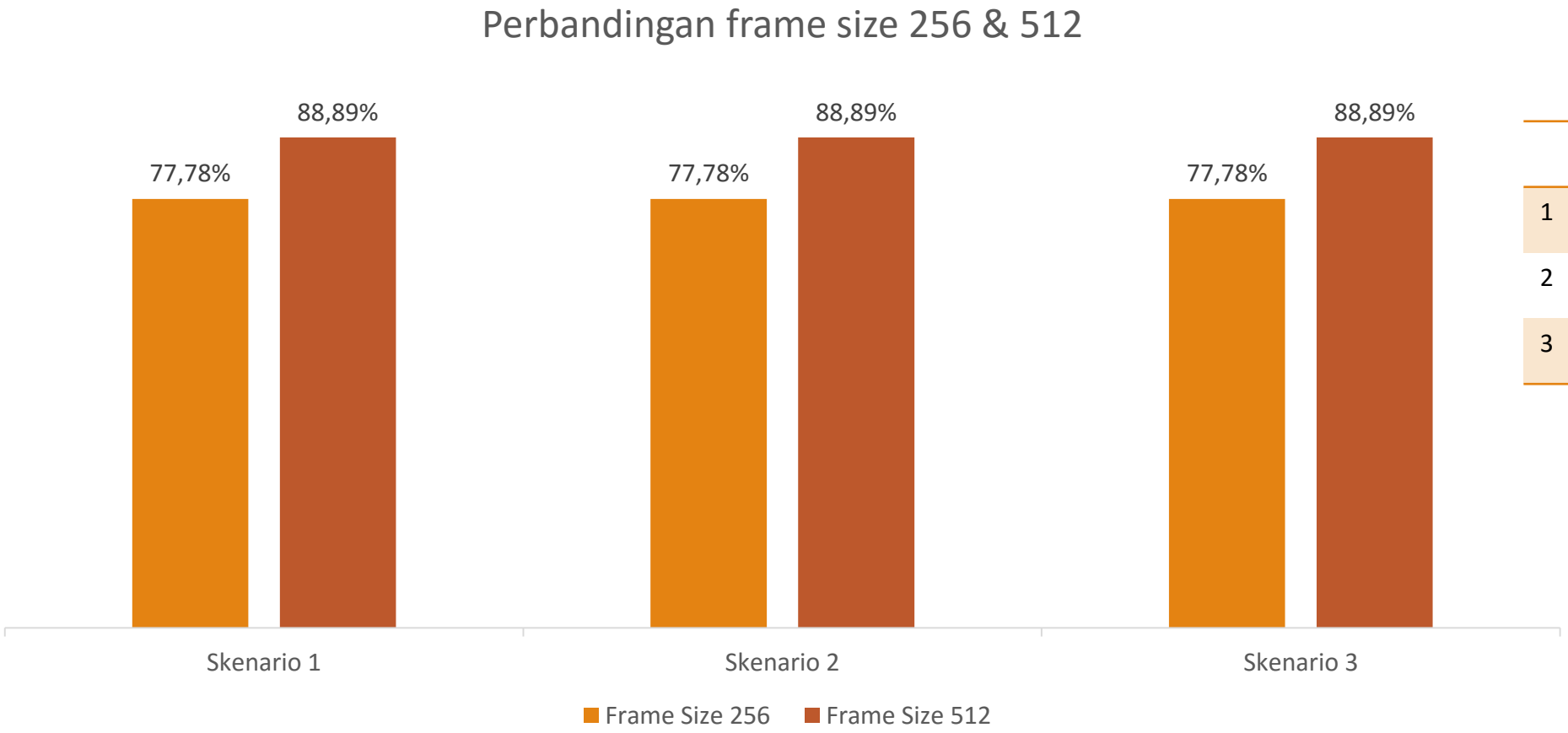
	Reno-Buka	Reno-Kunci	Reno-Lock	Reno-Unlock
Jumlah hasil	0	76	11	0

maka data *input* tersebut paling banyak masuk ke kelas Reno-Kunci, sehingga dapat disimpulkan kelas akhir dari data *input* adalah Reno-Kunci.

Pengujian

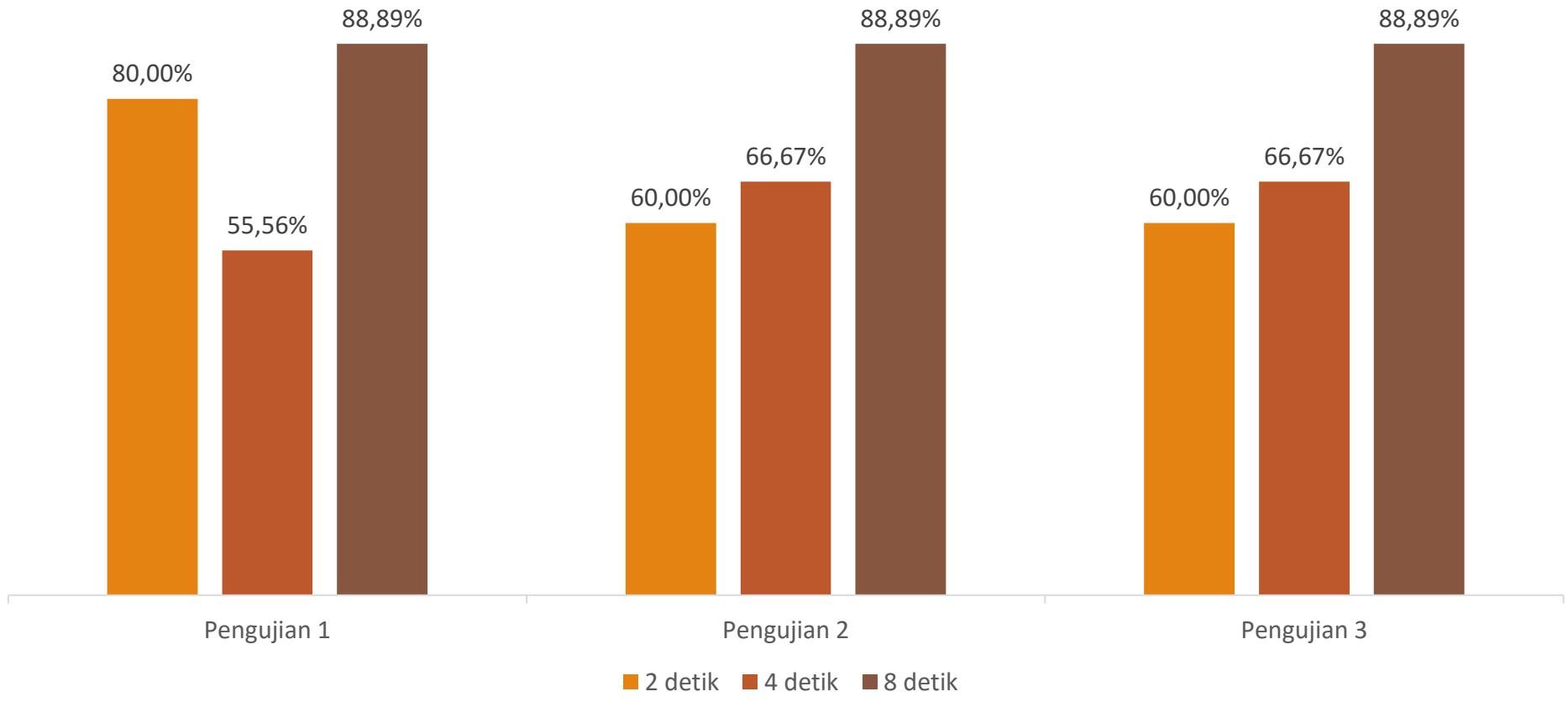
- Setiap orang akan diambil 1 rekaman suara untuk setiap kata, jumlah data uji setiap pembicara adalah 4 suara, sehingga total data uji adalah 20 buah suara.
- Dilakukan 3 kali pelatihan data dengan parameter yang sama untuk mendapatkan 3 bobot akhir yang berbeda.
- Akurasi akhir yang didapat adalah hasil akurasi rata-rata dari 3 kali pengujian pada parameter yang sama.

HASIL PENGUJIAN



	Alpha	Decay	Epoch
1	0.05	0.1	1000
2	0.03	0.1	1000
3	0.03	0.095	1000

HASIL PENGUJIAN



KESIMPULAN

- Nilai *alpha* dan *alpha decay* cukup mempengaruhi akurasi identifikasi dan verifikasi pembicara. Semakin kecil nilai *alpha* dan nilai *alpha decay* makan akurasi yang didapatkan akan semakin baik.
- Nilai *frame size* sebesar 512 menghasilkan akurasi yang lebih baik sekitar 10-15% dibandingkan dengan nilai *frame size* sebesar 256.
- Akurasi identifikasi pembicara dengan data latih hanya satu kata yang didapatkan lebih baik daripada akurasi identifikasi pembicara pada data latih banyak kata
- Metode MFCC menghasilkan akurasi yang lebih baik untuk identifikasi pembicara saja, akurasi yang didapat paling tinggi untuk pembicara adalah sebesar 80%. Sehingga metode ini lebih cocok untuk pengenalan pembicara daripada pengenalan kata.

SARAN

- Metode MFCC sebaiknya hanya digunakan untuk pengenalan pembicara saja, dapat dilihat hasil akurasi dalam mengenali pembicara berikut kata yang diucapkan masih kurang.
- Optimasi-optimasi parameter LVQ dan MFCC dapat dikembangkan lebih jauh lagi untuk mendapatkan akurasi pengenalan yang lebih baik lagi.