BAB 3
ANALISIS KEBUTUHAN ALGORITMA

3.1 Analisis Masalah

Dalam pengenalan suara, proses ekstraksi ciri suara dan pencocokan suara sangatlah penting karena untuk menghasilkan suatu *speaker recognition* yang baik dibutuhkan keakurasian yang baik. Metode yang digunakan untuk ekstraksi ciri suara dan pencocokan suara pun dijadikan suatu pertimbangan. Di penelitian sebelumnya dijelaskan bahwa metode MFCC dan DTW pada pengujian dari sistem verifikasi suara menghasilkan rata-rata akurasi sebesar 73,260% [1].

Penelitian sebelumnya untuk ekstraksi ciri suara menjelaskan bahwa ekstraksi ciri MFCC mempunyai tingkat keakuratan tertinggi dengan tingkat pengenalan 85,3% dan waktu ekstraksi ciri tercepat dibandingkan dengan metode ekstraksi ciri yang lainnya [7].

Hasil dari ekstraksi fitur menggunakan metode MFCC akan digunakan untuk proses pencocokan, dimana proses pencocokan ini akan membandingkan hasil ekstraksi fitur dari data uji dengan hasil ekstraksi fitur data latih yang terdapat pada *database*. Pada saat dilakukan pencocokan, hasil filter pada MFCC yang tersebar tidak merata mempengaruhi banyaknya informasi yang diperoleh untuk mendapatkan karakter suara seseorang [16]. *Clustering* dapat mengelompokan data hasil filter MFCC berdasarkan kemiripan dari hasil filter MFCC tersebut, salah satunya dengan menggunakan *k-mean clustering*. *K-Means Clustering* digunakan untuk mempartisi vektor hasil ekstraksi fitur ke dalam (k) vektor pusat [2]. *K-Means Clustering* juga mengurangi waktu pemrosesan untuk pencocokan suara dengan membuat vektor data sebagai representasi dari keseluruhan data hasil filter MFCC. Sebagai klasifikator digunakan jaringan syaraf tiruan *Learning Vector Quantization* (LVQ). LVQ melakukan pembelajaran terhadap hasil *k-mean clustering* yang merupakan vektor dari hasil filter MFCC. Hasil dari pembelajaran LVQ akan disimpan sebagai model *pattern* pada *database* yang akan digunakan untuk proses pencocokan suara. Proses pencocokan suara dilakukan dengan pengukuran jarak

terdekat menggunakan *euclidean distance* untuk mengetahui seberapa besar kemiripan suara data uji dengan model *pattern*.

Berdasarkan uraian tersebut, pada penelitian ini akan diterapkan *K-Means Clustering* dan LVQ untuk optimasi akurasi pada pengenalan suara. Di harapkan dari proses verifikasi suara dengan penerapan metode tersebut diketahui seberapa besar pengaruh akurasinya.

3.2 Analisis Proses

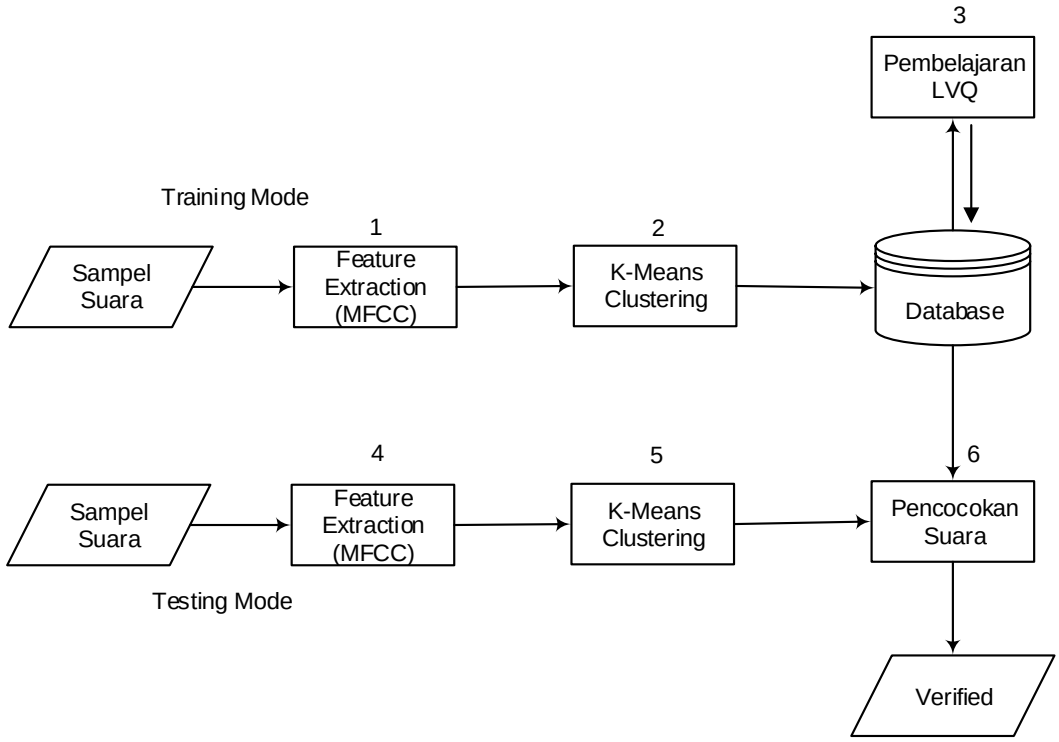
Proses dari sistem pengenalan suara menggunakan Metode Mel Frequency Cepstrum Coefficients (MFCC), *K-Means Clustering* dan *Learning Vector Quantization* (LVQ) adalah proses perekaman suara (*voice recorder*), proses latih (*training mode*) dan proses pengujian (*testing mode*).

Voice Recorder merupakan bagian yang bertugas merekam suara pembicara. Suara direkam dengan menggunakan format PCM *(pulse code modulation*). Alat yang dibutuhkan untuk proses ini adalah microphone dan audio speaker.

Training Mode adalah bagian yang bertugas mempelajari dan memodelkan suara pembicara. Pembelajaran terhadap suara pembicara dilakukan dengan membuat model dari suara pembicara kemudian menyimpannya ke dalam database.

Testing Mode merupakan bagian yang bertugas mengenali suara pembicara. Pengenalan suara dilakukan dengan melakukan pencocokan data uji dengan data model *pattern* pada *database*. Pada pencocokan, perbandingan yang dilakukan adalah perbandingan 1:1 dimana model suara dari A langsung dibandingkan dengan hasil pembelajaran pemodelan suara A yang ada dalam *database*.

Proses pengenalan suara menggunakan metode MFCC, *k-means clustering* dan *learning vector quantization* dapat dilihat pada Gambar 3.1 berikut :



Gambar 3.1 Alur Proses Pengenalan Suara

Gambar 3.1 merupakan alur proses pengenalan suara, berikut penjelasannya:

1. Pada sample suara *training mode* akan dilakukan *feature extraction* menggunakan metode MFCC.
2. Hasil *feature extraction* di kluster menggunakan *k-means clustering* dan disimpan dalam *database*.
3. Pembelajaran LVQ mengambil data latih dari *database* lalu melakukan pembelajaran sehingga menghasilkan pemodelan pembelajaran yang disimpan ke *database*.
4. Pada sample suara *testing mode* akan dilakukan *feature extraction* menggunakan metode MFCC.
5. Hasil *feature extraction* di kluster menggunakan k-means *clustering*.
6. Titik pusat hasil clustering pada langkah no.5 dicocokan dengan hasil pemodelan pembelajaran LVQ yang ada pada *database* dengan menggunakan

euclidean distance, setelah proses pencocokan selesai maka didapatkan hasil verifikasi.

3.3 Analisis Data Masukan

Analisis data terdiri dari analisis data untuk ektraksi ciri suara menggunakan metode MFCC dan analisis data untuk pencocokan suara menggunakan metode gabungan *k-mean clustering* dan LVQ. Analisis data tersebut terdiri dari analisis data masukan dan analisis data keluaran. Pada proses ekstraksi ciri suara, data masukan berbentuk berkas audio .wav maka data masukan berkas audio .wav yang bersifat sinyal analog akan dikonversikan dulu ke dalam bentuk sinyal digital melalui proses *pre-processing* dan akuisisi data. Pada proses pencocokan suara, data yang digunakan merupakan data hasil dari fitur ekstraksi ciri menggunakan metode MFCC. Berikut ini penjelasannya :

3.3.1 Analisis Data Masukan MFCC

Analisis data masukan MFCC terdiri dari :

3.3.1.1 Pre-Proccessing

Sinyal suara yang akan diproses bersifat *analog* sehingga jika akan dilakukan pengolahan secara *digital*, sinyal suara tersebut harus dikonversi menjadi sinyal digital, berupa urutan angka dengan tingkat presisi tertentu yang dinamakan analog to digital conversion dengan menggunakan *analog to digital converter* (ADC). Konsep kerja ADC terdiri dari tiga proses yaitu :

1. Sampling

Proses sampling adalah suatu proses untuk mengambil data *signal continue* untuk setiap periode tertentu. Data berupa sinyal diambil dengan cara merekam melalui *microphone*, data akan disimpan menggunakan format berkas file audio wav. Frekuensi sampling yang digunakan pada proses perekaman adalah 8000Hz. Durasi perekaman dilakukan dengan waktu yang pendek yaitu 1 detik.

2. Kuantisasi

Proses kuantisasi adalah proses untuk membulatkan nilai data ke dalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu. Proses ini melakukan pengkonversian nilai analog ke dalam suatu nilai diskrit. Selama proses kuantisasi, ADC menkonversi setiap nilai analog ke dalam bentuk diskrit.

3. Pengkodean

Proses pengkodean adalah proses pemberian kode untuk tiap-tiap data signal yang telah terkuantisasi berdasarkan level yang ditempati. Pada proses ini, tiap nilai diskrit yang telah didapat, direpresentasikan dengan angka binary n-bit. Cara konversinya adalah dengan membagi bilangan desimal dengan bilangan biner dengan memperhatikan hasil sisa pembagian.

Pada perekaman suara untuk kata "satu" yang dilakukan selama 1 detik pada frekuensi 8000Hz, dengan format berkas audio .wav menggunakan 16 bit dan *channel mono*. Diambil data sinyal sebanyak 10, data tersebut adalah 17, -454, -296, 23, 163, 1208, -1211, -511, 199, -255.

3.3.1.2 Akuisisi Data

Data diperoleh dengan cara merekam suara melalui mikrofon yang dihubungkan dengan komputer. Pada analisis ini perekaman suara menggunakan frekuensi 8000Hz dan menggunakan format berkas audio .wav. Suara dengan format .wav ini bisa menggunakan 16 bits/sample dan 1 untuk channel mono. Durasi suara dengan jangka waktu yang pendek akan lebih mudah untuk diambil perbedaan fiturnya.

Dalam analisis ini digunakan contoh durasi rekaman yang diambil dalam waktu 1 detik dan kata yang di ucapkan adalah kata "satu".

$$X = FS\ x\ dt(detik)\ x\ \left(\frac{bit}{8}\right) x\ j \quad (3.1)$$

X = Data sampling

FS = Frekuensi sampling

dt = durasi rekaman (detik)

bit = jumlah bit resolusi

j = 1 untuk *mono*, dan 2 untuk *stereo*

Perhitungan untuk proses akuisisi data adalah :

8000 x 1 x (16/8) x 1 = 16000 byte

Untuk menghitung sample rate digunakan cara data sample dibagi dengan waktu perekaman, maka :

$$Sample\ rate = \frac{16000}{1} = 16000\ Hz$$

Untuk mendapatkan sample point maka pada analisis ini melakukan pengambilan data pada setiap 20ms maka :

Sample point = sample rate x waktu pengambilan data (3.2)

Sample point = 16000 x 0,02 = 320

Untuk pengambilan data pada setiap 20 ms didapatkan 320 sample point.

Data masukan yang digunakan merupakan data sinyal suara berdasarkan pengguna dan kata yang diucapkan yang dapat dilihat pada Tabel 3.1. Pada tabel tersebut terdapat sinyal suara yang akan menjadi data masukan untuk melakukan pengenalan suara pada verifikasi suara, data sinyal suara tersebut diperoleh dengan menggunakan fungsi *wavread* pada Matlab, berikut Tabel 3.1 :

Tabel 3.1 Data Masukan

No	Identitas	Kata	Sinyal
1	User1	Satu	17, -454, -296, 23, 163, 1208, -1211, -511, 199, -255
2	User2	Satu	-394, -457, -600, -804, -736, -842, -21, 160, 278, 280
3	User3	Satu	-400, -391, -299, -322, -272, 20, -92, -229, -131, -113
4	User4	Satu	-66, -58, -90, -51 , 188, 344, 342, 188, -511, -507
5	User5	Satu	93, -46, -11, , 162, -53, 113, 41, -115, -737, -508

3.3.2 Analisis Data Keluaran MFCC

Analisis data keluaran dari metode MFCC merupakan hasil *feature extraction* sebagai berikut:

Tabel 3.2 Data Keluaran

Identitas	i	Fitur ke i
User1	0	44.91442503
User1	1	2.920313430
User1	2	53.501330636
User1	3	3.4044221681
User1	4	6.636070651
User1	5	3.4231766728
User1	6	16.820597761
User1	7	64.167837792
User1	8	48.658774043
User1	9	22.054725305

3.3.3 Analisis Data Masukan K-Means Clustering

Data masukan untuk *k-means clustering* adalah hasil dari *feature extraction* menggunakan metode MFCC sebagai berikut:

Tabel 3.3 Data Masukan

Identitas	i	Fitur ke i
User1	0	44.91442503
User1	1	2.920313430
User1	2	53.501330636
User1	3	3.4044221681
User1	4	6.636070651
User1	5	3.4231766728
User1	6	16.820597761
User1	7	64.167837792
User1	8	48.658774043
User1	9	22.054725305

3.3.4 Analisis Data Keluaran K-Means Clustering

Data keluaran untuk *k-means clustering* adalah hasil dari *clustering* sebagai berikut:

Tabel 3.4 Data Keluaran

Cluster	Nilai Centroid
0	45,49
1	33,94
2	17,02
3	93,846
4	58,79
5	6,374107

3.3.5 Analisis Data Masukan LVQ

Data masukan untuk LVQ adalah hasil dari *clustering* menggunakan metode k-means sebagai berikut:

Tabel 3.5 Data Masukan

Variabel	Nilai	Target Kelas
User1	45.49, 33.94, 17.02, 93.846, 58.79, 58.79	1
User1	119.229, 137.860, 118.015, 61.877, 39.471, 2.124	1
User2	151.004, 19.627, 183.137, 93.997, 69.1228, 3.187	2
User2	99.691, 105.655, 87.291, 45.977, 27.787, 1.593	2

3.3.6 Analisis Data Keluaran LVQ

Data masukan untuk LVQ adalah hasil dari *clustering* menggunakan metode *k-means clustering* sebagai berikut:

Tabel 3.6 Data Keluaran

Target	Nilai
1	84.502, 85.727, 69.456, 36.124, 21.303, 1.276
2	134.619, 67.084, 26.672, 14.948, 43.049, 1.062

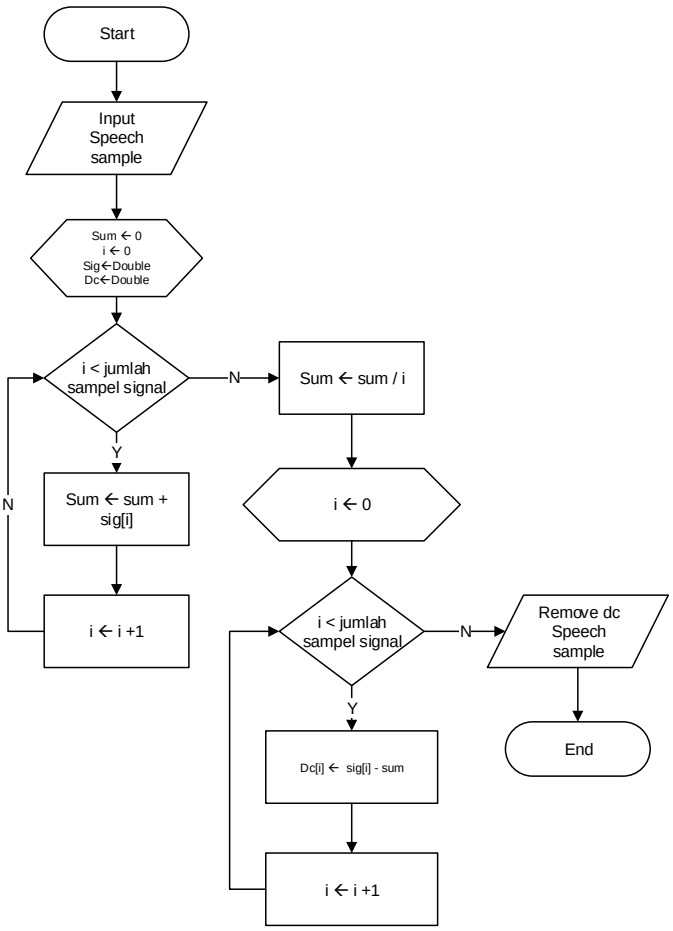
3.4 Analisis Metode

Metode yang diterapkan pada pengenalan suara meliputi MFCC dan gabungan *K-Means Clustering* dan LVQ. Algoritma MFCC terdiri dari beberapa algoritma lagi yaitu algoritma dc removal, pre-emphasize, frame blocking, windowing, dct, filterbank, dct dan cepstral liftering. Berikut ini adalah analisis dari masing-masing metode yang digunakan :

3.4.1 Metode MFCC

Metode MFCC terdiri dari beberapa proses yaitu *dc removal*, *pre-emphasize*, *frame blocking*, *windowing*, *discrte fourier transfrom*, *filterbank*, *discrete cosine transform* dan *cepstral liftering*.

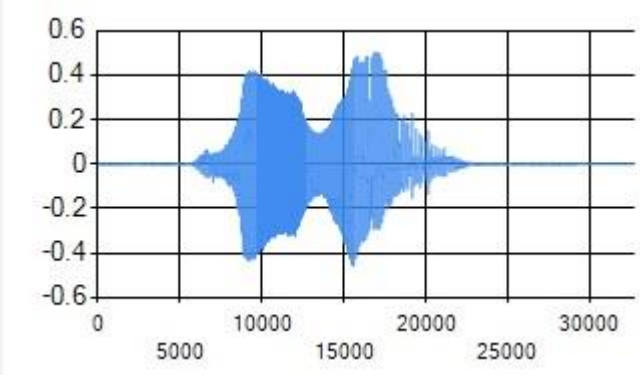
3.4.1.1 DC Removal



Gambar 3.2 Alur *DC Removal*

Proses DC *removal* dilakukan untuk menghitung rata-rata dari data sampel suara, dan mengurangkan nilai setiap sampel suara dengan nilai rata-rata tersebut. Tujuannya adalah mendapat normalisasi dari data suara input.

Perhitungan DC Removal untuk kata “satu”, dengan data sinyalnya dapat dilihat pada Gambar 3.3 berikut:



Gambar 3.3 Sinyal Suara

Data sinyal tersebut dalam numerik adalah sebagai berikut :

Tabel 3.7 Data Sinyal

i	Signal ke i
0	-0.0036
1	-0.0439
2	-0.0405
3	-0.0376
4	-0.0377
5	-0.0416
6	-0.0424
7	-0.0395
8	-0.0369
9	-0.0403
10	-0.0435
11	-0.0413
12	-0.0375
13	-0.0395
14	-0.0427
15	-0.0419
16	-0.038

i	Signal ke i
17	-0.033
18	-0.0348
19	-0.0357
20	-0.0352
21	-0.0308
22	-0.0291
23	-0.0303
24	-0.0327
25	-0.035
26	-0.0369
27	-0.0372
28	-0.0365
29	-0.036

Hitung nilai rata-rata sinyal :

$$\overline{\mathrm{x}} = \frac{s_0 + \cdots + s_n}{jumlah\ sinyal} \qquad (3.3)$$

$$\overline{\mathrm{x}} = \frac{(-0.0036) + \cdots + (-0.036)}{30} = 0{,}039$$

Hitung DC *Removal* :

$$D[n] = s[n] - \bar{x}\ ,\ 0 \leq n \leq N\text{-}1 \qquad (3.4)$$

D[n] = sampel signal hasil proses DC removal

s[n] = sampel signal asli

$\bar{x}$ = nilai rata-rata sampel signal asli.

N = panjang *signal*

$D_0 = s_0 - \overline{\mathrm{x}} = (-0.0036) - (0{,}039) = -0.0426$

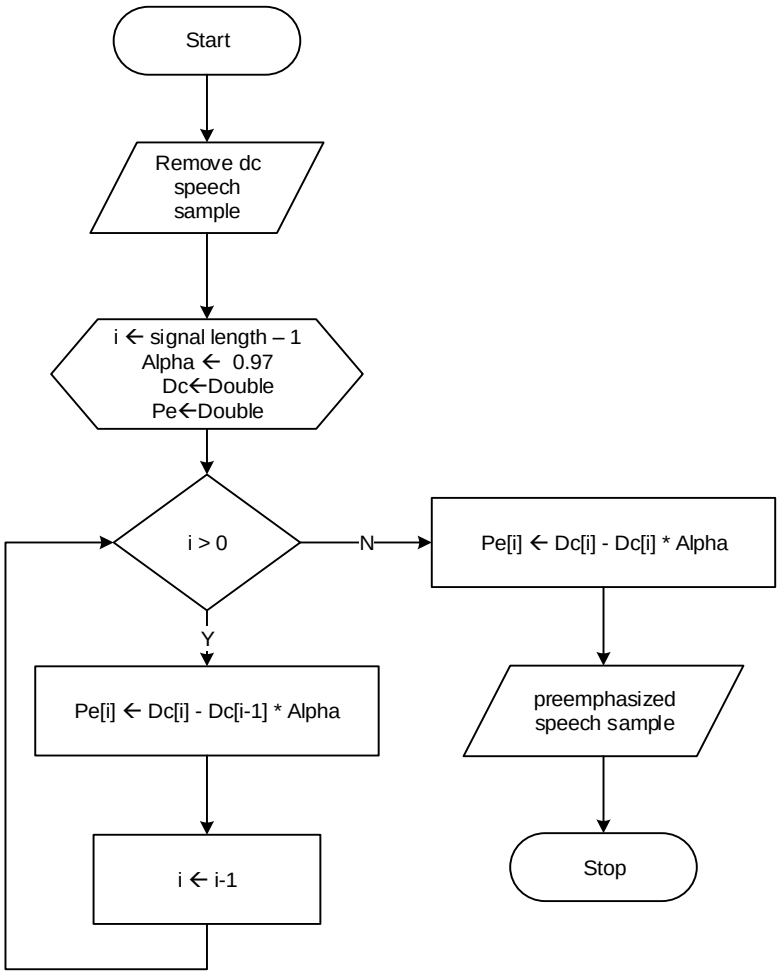
Hitung untuk data sinyal berikutnya dengan cara yang sama, data sinyal setelah dilakukan DC Removal adalah :

Tabel 3.8 Data *DC Removal*

i	DC Removal ke i
0	-0.0426
1	-0.0049

i	DC Removal ke i
2	-0.0015
3	0.0014
4	0.0013
5	-0.0026
6	-0.0034
7	-0.0005
8	0.0021
9	-0.0013
10	-0.0045
11	-0.0023
12	0.0015
13	-0.0005
14	-0.0037
15	-0.0029
16	0.001
17	0.006
18	0.0042
19	0.0033
20	0.0038
21	0.0082
22	0.0099
23	0.0087
24	0.0063
25	0.004
26	0.0021
27	0.0018
28	0.0025
29	0.0026

3.4.1.2 Pre-Emphasize



Gambar 3.4 Alur *Pre Emphasize*

Proses *pre-emphasize* merupakan salah satu jenis *filter* yang sering digunakan sebelum sebuah *signal* diproses lebih lanjut. *Filter* ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara. *Pre – emphasize* dilakukan untuk mengurangi *noise ratio* pada *signal*, sehingga dapat meningkatkan kualitas *signal* dan menyeimbangkan spektrum dari *voiced sound.*

$$y[n] = s[n] - \alpha\, s[n-1] \tag{3.5}$$

y[n] = *signal* hasil *pre-emphasize filter*

s[n] = *signal* sebelum *pre-emphasize filter*

Nilai α yang digunakan pada analisis ini adalah 0,97.

$y[0] = -0.0426$

$y[1] = s[1] - \alpha\, s[1-1] = (-0.0049) - 0{,}97 \text{ x } (-0.0426) = 0.036422$

Hitung untuk data sinyal berikutnya dengan cara yang sama, data sinyal setelah pre-emphasize adalah sebagai berikut :

Tabel 3.9 Data *Pre Emphasize*

i	Pre Empasize ke-i
0	-0.0426
1	0.036422
2	0.003253
3	0.002855
4	-0.005855
5	-0.003861
6	-0.000878
7	0.002798
8	0.002585
9	-0.003337
10	-0.003239
11	0.002065
12	0.003731
13	-0.001955
14	-0.003215
15	0.000689
16	0.003813
17	0.00503
18	-0.00162
19	-0.000774
20	0.000599
21	0.004514
22	0.001946
23	-0.000903
24	-0.002139
25	-0.002111
26	-0.00178
27	-0.000237
28	0.000754

i	Pre Empasize ke-i
29	0.000175

3.4.1.3 Frame Blocking

Data hasil perekaman suara merupakan sinyal analog yang berada dalam domain waktu yang bersifat *variant time*, yaitu suatu fungsi yang bergantung waktu. Proses *frame blocking* dilakukan dengan cara memotong-motong sinyal dalam slot-slot waktu tertentu agar dapat dianggap *invariant*. Setiap potongan tersebut disebut frame.

$$\text{Jumlah frame} = ((\text{I-N})/\text{M})+1) \tag{3.6}$$

I = Sample rate

N= Sample point

M = N/2

Hitung Jumlah Frame :

$$\text{Jumlah frame} = \frac{16000 - 320}{320/2} + 1 = 99$$

Jumlah frame adalah 99 frame.

3.4.1.4 Windowing

Setelah melakukan proses *frame blocking*, proses selanjutnya adalah *windowing*. Fungsi pada proses windowing ini menggunakan fungsi *Hamming Window* sebagai berikut :

$$w(n) = 0.54 - 0.46\, cos\, \frac{2\pi n}{M - 1} \tag{3.7}$$

n = 0,1,...,M-1

M = panjang *frame*

$$w(0) = 0{,}54 - 0{,}46 \cos\left(\frac{2x3{,}14x0}{320 - 1}\right) = 0{,}08$$

Hitung dengan cara yang sama untuk data yang lainnya.

Selanjutnya merepresentasikan fungsi *window* terhadap sinyal.

$$x(n) = x_i(n)w(n) \quad (3.8)$$

n = 0,1,…,N-1

$x(n)$ = nilai sampel *signal* hasil *windowing*

$x_i(n)$ = nilai sampel dari *frame signal* ke i

$w(n)$ = fungsi *window*

N = *frame size*

$x(0) = x_0\,(0) * w(0) = -0.0426 * 0{,}08 = -0.003408$

Hitung dengan cara yang sama untuk data yang lainnya, data setelah dilakukan proses windowing adalah :

Tabel 3.10 Data *Windowing*

i	Windowing ke-i
0	-0.003408
1	0.00291376
2	0.00026024
3	0.0002284
4	-4.64E-06
5	-0.00030888
6	-0.00007024
7	0.00022384
8	0.0002068
9	-0.00026696
10	-0.00025912
11	0.0001652
12	0.00029848
13	-0.0001564
14	-0.0002572
15	0.00005512
16	0.00030504
17	0.0004024
18	-0.0001296
19	-0.00006192
20	0.00004792
21	0.00036112

i	Windowing ke-i
22	0.00015568
23	-7.224E-05
24	-0.00017112
25	-0.00016888
26	-0.0001424
27	-0.00001896
28	0.00006032
29	0.000014

3.4.1.5 Fast Fourier Transform

DFT merupakan perluasan dari transformasi *fourier* yang berlaku untuk *signal-signal* diskrit dengan panjang yang terhingga. Semua *signal* periodik terbentuk dari gabungan *signal-signal* sinusoidal yang menjadi satu yang dapat dirumuskan sebagai berikut :

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-\frac{j2\pi nk}{N}}, 0 \leq k \leq N-1 \qquad (3.9)$$

N = jumlah sampel yang akan diproses ($N \in$ N)

S(n) = nilai sampel *signal*

K = variable frekuensi discrete

$$f(0) = \frac{1}{30}\left[-0.003408\left(\cos\left(\frac{2\pi*0*0}{30}\right)\right)\right] - j\ sin\frac{2\pi*0*0}{30} +$$
$$\cdots+\left[0.000014\left(\cos\left(\frac{2\pi*0*29}{30}\right)\right)\right] - j\ sin\frac{2\pi*0*29}{30} = 0.0002$$

Hitung untuk data berikutnya, maka didapatkan hasil sebagai berikut:

Tabel 3.11 Data FFT

i	FFT ke-i
0	0.0002
1	0.001
2	0.0002
3	0.0003
4	0.0009

i	FFT ke-i
5	0.0026
6	0.0044
7	0.0032
8	0.0033
9	0.0045
10	0.0048
11	0.0052
12	0.0058
13	0.0061
14	0.0058
15	0.0064
16	0.0058
17	0.0061
18	0.0058
19	0.0052
20	0.0048
21	0.0045
22	0.0033
23	0.0032
24	0.0044
25	0.0026
26	0.0009
27	0.0003
28	0.0002
29	0.001

3.4.1.6 Filterbank

Hasil dari FFT selanjutnya akan melalui tahap filterbank

Perhitungan Mel Scale:

$$Hi = \frac{mel\ f}{Xi/_2} \tag{3.10}$$

$$mel\ f = 2595 \times log\left(1 + \frac{f}{700}\right) \tag{3.11}$$

Mel f = Fungsi Mel Scale

f = frekuensi

Pada penelitian ini *f* = 1000

$$Hi = \frac{2595 \times log(1 + \frac{f}{700})}{Xi/_2} \tag{3.12}$$

$$H_0 = \frac{2595 \times log(1 + \frac{1000}{700})}{0.0002/_2} = 9999.855371$$

Berikut perhitungan untuk *filterbank* :

$$Y[i] = \sum_{j=1}^{N} S[j]H_i[j] \tag{3.13}$$

N = jumlah *magnitude spectrum* (*N* ∈ N)

S[j] = *magnitude spectrum* pada frekuensi j

$H_i[j]$ = koefisien *filterbank* pada frekuensi j ($1 \leq i \leq M$)

M = jumlah *channel* dalam *filterbank*

$$Y_0 = 0.0002 \times 9999.855371 = 37649.4857$$

Hitung dengan cara yang sama untuk data yang lainnya, didapatkan hasil sebagai berikut :

Tabel 3.12 Data *Filterbank*

i	Filterbank ke-i
0	37649.4857
1	34906.08019
2	174530.401
3	116353.6006
4	38784.53355
5	13425.41546
6	7933.200043
7	10908.15006
8	10577.60006
9	7756.906709
10	7272.10004
11	6712.707729
12	6018.289688
13	5722.308228
14	6018.289688
15	5454.07503
16	6018.289688
17	5722.308228
18	6018.289688
19	6712.707729
20	7272.10004
21	7756.906709
22	10577.60006
23	10908.15006
24	7933.200043
25	13425.41546
26	38784.53355
27	116353.6006
28	174530.401
29	34906.08019

3.4.1.7 Discrete Cosine Transform

Proses ini adalah proses akhir dari MFFC, menggunakan rumus :

$$C_n = \sum_{k=0}^{k} (logS_k)\cos[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}] \quad (3.14)$$

Koef (k) = 10

Data *filterbank* sebagai berikut:

$$n_0 = \log(37649.4857)\, cos\left[0\left(0 - \frac{1}{2}\right)\frac{3{,}14}{10}\right] + \cdots$$
$$+ \log(34906.08019)\, cos\left[0\left(30 - \frac{1}{2}\right)\frac{3{,}14}{10}\right] = 17.4437$$

Hitung dengan cara yang sama untuk data yang lainnya, didapatkan hasil sebagai berikut :

Tabel 3.13 Data DCT

i	DCT ke-i
0	17.4437
1	17.41414
2	17.38534
3	17.3572
4	17.32965
5	17.30259
6	16.67375
7	16.64597
8	16.61855
9	16.59144
10	17.12989
11	17.10545
12	17.08111
13	17.05685
14	17.03262
15	17.00841
16	16.9842
17	16.95997
18	16.9357
19	16.91138

i	DCT ke-i
20	16.88701
21	16.86257
22	16.83805
23	16.81347
24	16.7888
25	16.76407
26	16.73925
27	16.71437
28	16.68942
29	16.66441

3.4.1.8 Cepstral Liftering

Untuk meningkatkan kualitas pengenalan maka dilakukan proses *Cepstral Liftering*

$$w[n] = \left\{1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right)\right\} \quad (3.15)$$

L = jumlah *cepstral coefficients*

n = *index* dari *cepstral coefficients*

$$w[0] = 17.4437 \times 1 + \frac{30}{2}\sin\left(\frac{0x3,14}{30}\right) = 44,91442503$$

Hitung dengan cara yang sama untuk data yang lainnya, didapatkan hasil sebagai berikut :

Tabel 3.14 Data *Cepstral*

i	Cepstral ke-i
0	44.61842
1	46.5462
2	54.4755
3	73.40605
4	52.79217
5	42.72087
6	52.6505
7	65.58093
8	43.96283
9	53.9001

i	Cepstral ke-i
10	83.83763
11	23.77537
12	73.71319
13	35.65105
14	77.58892
15	48.52674
16	49.46445
17	43.40203
18	47.33949
19	58.27676
20	52.21384
21	53.15075
22	73.08744
23	64.02397
24	68.96027
25	65.89642
26	49.83239
27	47.7682
28	67.61842
29	59.5462

3.4.2 K-Means Clustering

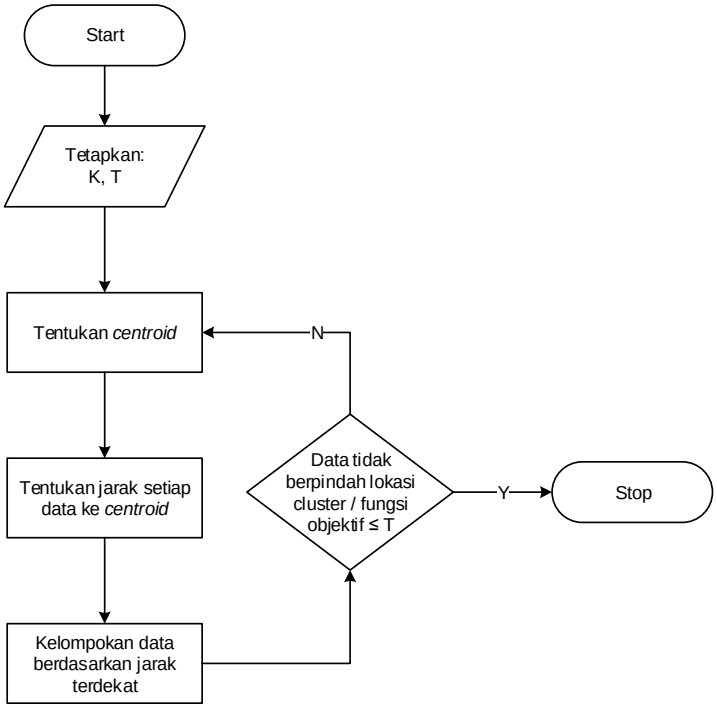
Berikut ini data fitur hasil ekstraksi ciri MFCC yang akan dilakukan proses *clustering* :

Tabel 3.15 Data Hasil Proses MFCC

i	Fitur ke-i
0	44.61842
1	46.5462
2	54.4755
3	73.40605
4	52.79217
5	42.72087
6	52.6505
7	65.58093

i	Fitur ke-i
8	43.96283
9	53.9001
10	83.83763
11	23.77537
12	73.71319
13	35.65105
14	77.58892
15	48.52674
16	49.46445
17	43.40203
18	47.33949
19	58.27676
20	52.21384
21	53.15075
22	73.08744
23	64.02397
24	68.96027
25	65.89642
26	49.83239
27	47.7682
28	67.61842
29	59.5462

Alur proses *k-mean clustering* dapat dilihat pada Gambar 3.5 berikut :



Gambar 3.5 Alur K-Means Clustering

Langkah-langkah yang dilakukan sebagai berikut :

1. Inisialisasi

   K = 6

   T = 0.001

   Keterangan:

   T = *Threshold*

   K = Jumlah *Cluster*

   Bangkitkan k *centroid* (titik pusat *cluster*) awal secara *random*. Berdasarkan perbandingan yang telah dilakukan terhadap nilai K, maka dipilih nilai K yang optimal pada penelitian ini yaitu K=6. Pada penelitian ini pemilihan K *centroid* awal diambil dari data hasil filter MFCC, dipilih data ke-0 sebagai *centroid cluster* 0, data ke-1 sebagai *centroid cluster* 1, data ke-2 sebagai *centroid cluster* 2, data ke-3 sebagai *centroid cluster* 3, data ke-4 sebagai *centroid cluster* 4 dan data ke-5 sebagai *centroid cluster* 5.

Centroid awal dapat dilihat pada Tabel 3.16 berikut:

Tabel 3.16 Data Centroid Awal

Centroid	Nilai
0	44.61842
1	46.5462
2	54.4755
3	73.40605
4	52.79217
5	42.72087

Karena data belum masuk dalam *cluster* maka nilai fungsi objektif diberi nilai nilai awal yang besar misalnya 1000.

2. Iterasi 1

Menghitung jarak setiap data ke *centroid* terdekat dengan menggunakan *Euclidean Distance*. Centroid terdekat akan menjadi *cluster* yang diikuti oleh data tersebut. Perhitungan *centroid* terdekat dengan menggunakan persamaan 2.16.

Berikut perhitungan jarak ke setiap *centroid* pada data ke -1:

$$d(x_1 - c_0) = \sqrt{(x_1 - c_0)^2} = \sqrt{(44.61842 - 44.61842)^2} = 0$$

$$d(x_1 - c_1) = \sqrt{(x_1 - c_1)^2} = \sqrt{(44.61842 - 46.5462)^2} = 1.92778$$

$$d(x_1 - c_2) = \sqrt{(x_1 - c_2)^2} = \sqrt{(44.61842 - 54.4755)^2} = 9.85708$$

$$d(x_1 - c_3) = \sqrt{(x_1 - c_3)^2} = \sqrt{(44.61842 - 73.40605)^2} = 28.78763$$

$$d(x_1 - c_4) = \sqrt{(x_1 - c_4)^2} = \sqrt{(44.61842 - 52.79217)^2} = 8.17375$$

$$d(x_1 - c_5) = \sqrt{(x_1 - c_5)^2} = \sqrt{(44.61842 - 42.72087)^2} = 1.89755$$

Berikut ini merupakan hasil *clustering* awal yang ditampilkan pada Tabel 3.17.

Tabel 3.17 *Clustering* Awal

Data ke-i	Jarak ke centroid						Terdekat	Cluster
	0	1	2	3	4	5		
0	0	1.92778	9.85708	28.78763	8.17375	1.89755	0	0
1	1.92778	0	7.9293	26.85985	6.24597	3.82533	0	1
2	9.85708	7.9293	0	18.93055	1.68333	11.75463	0	2
3	28.78763	26.85985	18.93055	0	20.61388	30.68518	0	3
4	8.17375	6.24597	1.68333	20.61388	0	10.0713	0	4
5	1.89755	3.82533	11.75463	30.68518	10.0713	0	0	5
6	8.03208	6.1043	1.825	20.75555	0.14167	9.92963	0.14167	4
7	20.96251	19.03473	11.10543	7.82512	12.78876	22.86006	7.82512	3
8	0.65559	2.58337	10.51267	29.44322	8.82934	1.24196	0.65559	0
9	9.28168	7.3539	0.5754	19.50595	1.10793	11.17923	0.5754	2
10	39.21921	37.29143	29.36213	10.43158	31.04546	41.11676	10.43158	3
11	20.84305	22.77083	30.70013	49.63068	29.0168	18.9455	18.9455	5
12	29.09477	27.16699	19.23769	0.30714	20.92102	30.99232	0.30714	3
13	8.96737	10.89515	18.82445	37.755	17.14112	7.06982	7.06982	5
14	32.9705	31.04272	23.11342	4.18287	24.79675	34.86805	4.18287	3
15	3.90832	1.98054	5.94876	24.87931	4.26543	5.80587	1.98054	1
16	4.84603	2.91825	5.01105	23.9416	3.32772	6.74358	2.91825	2
17	1.21639	3.14417	11.07347	30.00402	9.39014	0.68116	0.68116	5
18	2.72107	0.79329	7.13601	26.06656	5.45268	4.61862	0.79329	1
19	13.65834	11.73056	3.80126	15.12929	5.48459	15.55589	3.80126	2
20	7.59542	5.66764	2.26166	21.19221	0.57833	9.49297	0.57833	4
21	8.53233	6.60455	1.32475	20.2553	0.35858	10.42988	0.35858	4
22	28.46902	26.54124	18.61194	0.31861	20.29527	30.36657	0.31861	3
23	19.40555	17.47777	9.54847	9.38208	11.2318	21.3031	9.38208	3
24	24.34185	22.41407	14.48477	4.44578	16.1681	26.2394	4.44578	3
25	21.278	19.35022	11.42092	7.50963	13.10425	23.17555	7.50963	3
26	5.21397	3.28619	4.64311	23.57366	2.95978	7.11152	2.95978	4
27	3.14978	1.222	6.7073	25.63785	5.02397	5.04733	1.222	1
28	23	21.07222	13.14292	5.78763	14.82625	24.89755	5.78763	3
29	14.92778	13	5.0707	13.85985	6.75403	16.82533	5.0707	2

Fungsi objektif didapat dari jumlah jarak keseluruhan data ke centroid yaitu 97.94231.

Perubahan fungsi objektif = 1000 - 97.94231 = 902.05769 (masih di atas T)

Selanjutnya dihitung *centroid* yang baru untuk setiap cluster berdasarkan data yang bergabung pada setiap clusternya.

Untuk cluster 0, ada 2 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.18 Data Anggota *Cluster* 0

Data anggota	Nilai
0	44.61842
8	43.96283

Nk = 2

Jumlah nilai = 88.58125

Rata rata = Jumlah nilai/Nk = 44.29063

Untuk cluster 1, ada 4 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.19 Data Anggota *Cluster* 1

Data anggota	Nilai
1	46.5462
15	48.52674
27	47.7682
18	47.33949

Nk = 4

Jumlah nilai = 190.18063

Rata rata = Jumlah nilai/Nk = 47.5451575

Untuk cluster 2, ada 5 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.20 Data Anggota *Cluster* 2

Data anggota	Nilai
2	54.4755
9	53.9001
16	49.46445
19	58.27676
29	59.5462

Nk = 5

Jumlah nilai = 275.66301

Rata rata = Jumlah nilai/Nk = 55.1326

Untuk cluster 3, ada 10 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.21 Data Anggota *Cluster* 3

Data anggota	Nilai
3	73.40605
7	65.58093
10	83.83763
12	73.71319
14	77.58892
22	73.08744
23	64.02397
24	68.96027
25	65.89642
28	67.61842

Nk = 10

Jumlah nilai = 713.71324

Rata rata = Jumlah nilai/Nk = 71.371324

Untuk cluster 4, ada 5 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.22 Data Anggota *Cluster* 4

Data anggota	Nilai
4	52.79217
6	52.6505
20	52.21384
21	53.15075
26	49.83239

Nk = 5

Jumlah nilai = 260.63965

Rata rata = Jumlah nilai/Nk = 52.12793

Untuk cluster 5, ada 4 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.23 Data Anggota *Cluster* 5

Data anggota	Nilai
5	42.72087
11	23.77537
13	35.65105
17	43.40203

Nk = 4

Jumlah nilai = 145.54932

Rata rata = Jumlah nilai/Nk = 36.38733

Rata-rata yang didapatkan dari 6 cluster tersebut adalah centroid baru sebagai berikut :

Tabel 3.24 *Centroid* Baru

Centroid	Nilai Centroid
0	44.290625
1	47.5451575
2	55.132602
3	71.371324
4	52.12793
5	36.38733

3. Iterasi 2

Tabel 3.25 *Clustering* Iterasi 2

Data ke-i	Jarak ke centroid						Terdekat	Cluster
	0	1	2	3	4	5		
0	0.327795	2.926738	10.51418	26.7529	7.50951	8.23109	0.327795	0
1	2.255575	0.998958	8.586402	24.82512	5.58173	10.15887	0.9989575	1
2	10.18488	6.930342	0.657102	16.89582	2.34757	18.08817	0.657102	2
3	29.11543	25.86089	18.27345	2.034726	21.27812	37.01872	2.034726	3
4	8.501545	5.247013	2.340432	18.57915	0.66424	16.40484	0.66424	4
5	1.569755	4.824288	12.41173	28.65045	9.40706	6.33354	1.569755	0
6	8.359875	5.105343	2.482102	18.72082	0.52257	16.26317	0.52257	4
7	21.29031	18.03577	10.44833	5.790394	13.453	29.1936	5.790394	3
8	0.327795	3.582328	11.16977	27.40849	8.1651	7.5755	0.327795	0
9	9.609475	6.354943	1.232502	17.47122	1.77217	17.51277	1.232502	2
10	39.54701	36.29247	28.70503	12.46631	31.7097	47.4503	12.466306	3
11	20.51526	23.76979	31.35723	47.59595	28.35256	12.61196	12.61196	5
12	29.42257	26.16803	18.58059	2.341866	21.58526	37.32586	2.341866	3
13	8.639575	11.89411	19.48155	35.72027	16.47688	0.73628	0.73628	5
14	33.2983	30.04376	22.45632	6.217596	25.46099	41.20159	6.217596	3
15	4.236115	0.981582	6.605862	22.84458	3.60119	12.13941	0.9815825	1
16	5.173825	1.919293	5.668152	21.90687	2.66348	13.07712	1.9192925	1
17	0.888595	4.143128	11.73057	27.96929	8.7259	7.0147	0.888595	0
18	3.048865	0.205668	7.793112	24.03183	4.78844	10.95216	0.2056675	1
19	13.98614	10.7316	3.144158	13.09456	6.14883	21.88943	3.144158	2

Data ke-i	Jarak ke centroid						Terdekat	Cluster
	0	1	2	3	4	5		
20	7.923215	4.668683	2.918762	19.15748	0.08591	15.82651	0.08591	4
21	8.860125	5.605593	1.981852	18.22057	1.02282	16.76342	1.02282	4
22	28.79682	25.54228	17.95484	1.716116	20.95951	36.70011	1.716116	3
23	19.73335	16.47881	8.891368	7.347354	11.89604	27.63664	7.347354	3
24	24.66965	21.41511	13.82767	2.411054	16.83234	32.57294	2.411054	3
25	21.6058	18.35126	10.76382	5.474904	13.76849	29.50909	5.474904	3
26	5.541765	2.287232	5.300212	21.53893	2.29554	13.44506	2.2872325	1
27	3.477575	0.223042	7.364402	23.60312	4.35973	11.38087	0.2230425	1
28	23.3278	20.07326	12.48582	3.752904	15.49049	31.23109	3.752904	3
29	15.25558	12.00104	4.413598	11.82512	7.41827	23.15887	4.413598	2

Fungsi objektif didapat dari jumlah jarak keseluruhan data ke *centroid* yaitu 97.94231. Perubahan fungsi objektif = 97.94231-84.374075 = 13.568235 (masih di atas T).

Lakukan langkah-langkah seperti pada iterasi 1 sampai nilai fungsi objektif ≤ T.Maka didapatkan hasil terakhir sebagai berikut:

Tabel 3.26 *Clustering* Terakhir

Data ke-i	Jarak ke centroid						Terdekat	Cluster
	0	1	2	3	4	5		
0	1.79978	8.96742	20.84302	31.70818	10.00728	21.79758	1.79978	0
1	0.128	10.8952	22.7708	29.7804	8.0795	19.8698	0.128	0
2	8.0573	18.8245	30.7001	21.8511	0.1502	11.9405	0.1502	4
3	26.98785	37.75505	49.63065	2.92055	18.78035	6.99005	2.92055	3
4	6.37397	17.14117	29.01677	23.53443	1.83353	13.62383	1.83353	4
5	3.69733	7.06987	18.94547	33.60573	11.90483	23.69513	3.69733	0
6	6.2323	16.9995	28.8751	23.6761	1.9752	13.7655	1.9752	4
7	19.16273	29.92993	41.80553	10.74567	10.95523	0.83507	0.83507	5
8	2.45537	8.31183	20.18743	32.36377	10.66287	22.45317	2.45537	0
9	7.4819	18.2491	30.1247	22.4265	0.7256	12.5159	0.7256	4
10	37.41943	48.18663	60.06223	7.51103	29.21193	17.42163	7.51103	3
11	22.64283	11.87563	3E-05	52.55123	30.85033	42.64063	3E-05	2
12	27.29499	38.06219	49.93779	2.61341	19.08749	7.29719	2.61341	3
13	10.76715	5E-05	11.87565	40.67555	18.97465	30.76495	5E-05	1
14	31.17072	41.93792	53.81352	1.26232	22.96322	11.17292	1.26232	3
15	2.10854	12.87574	24.75134	27.79986	6.09896	17.88926	2.10854	0
16	3.04625	13.81345	25.68905	26.86215	5.16125	16.95155	3.04625	0

Data ke-i	Jarak ke centroid						Terdekat	Cluster
	0	1	2	3	4	5		
17	3.01617	7.75103	19.62663	32.92457	11.22367	23.01397	3.01617	0
18	0.92129	11.68849	23.56409	28.98711	7.28621	19.07651	0.92129	0
19	11.85856	22.62576	34.50136	18.04984	3.65106	8.13924	3.65106	4
20	5.79564	16.56284	28.43844	24.11276	2.41186	14.20216	2.41186	4
21	6.73255	17.49975	29.37535	23.17585	1.47495	13.26525	1.47495	4
22	26.66924	37.43644	49.31204	3.23916	18.46174	6.67144	3.23916	3
23	17.60577	28.37297	40.24857	12.30263	9.39827	2.39203	2.39203	5
24	22.54207	33.30927	45.18487	7.36633	14.33457	2.54427	2.54427	5
25	19.47822	30.24542	42.12102	10.43018	11.27072	0.51958	0.51958	5
26	3.41419	14.18139	26.05699	26.49421	4.79331	16.58361	3.41419	0
27	1.35	12.1172	23.9928	28.5584	6.8575	18.6478	1.35	0
28	21.20022	31.96742	43.84302	8.70818	12.99272	1.20242	1.20242	5
29	13.128	23.8952	35.7708	16.7804	4.9205	6.8698	4.9205	4

Selanjutnya dihitung *centroid* yang baru untuk setiap cluster berdasarkan data yang bergabung pada setiap clusternya.

Untuk cluster 0, ada 4 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.27 Data Anggota *Cluster* 0

Data anggota	Nilai
0	44.61842
5	42.72087
8	43.96283
17	43.40203

Nk =4

Jumlah nilai = 174.7042

Rata rata = Jumlah nilai/Nk = 43.67604

Untuk cluster 1, ada 6 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.28 Data Anggota *Cluster* 1

Data anggota	Nilai
1	46.5462
15	48.52674
16	49.46445
18	47.33949
26	49.83239

Data anggota	Nilai
27	47.7682

Nk =6

Jumlah nilai = 289.4775

Rata rata = Jumlah nilai/Nk = 48.24625

Untuk cluster 2, ada 4 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.29 Data Anggota *Cluster* 2

Data anggota	Nilai
2	54.4755
9	53.9001
19	58.27676
29	59.5462

Nk =4

Jumlah nilai = 226.1986

Rata rata = Jumlah nilai/Nk = 56.54964

Untuk cluster 3, ada 10 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.30 Data Anggota *Cluster* 3

Data anggota	Nilai
3	73.40605
7	65.58093
10	83.83763
12	73.71319
14	77.58892
22	73.08744
23	64.02397
24	68.96027
25	65.89642
28	67.61842

Nk =10

Jumlah nilai = 713.7132

Rata rata = Jumlah nilai/Nk = 71.37132

Untuk cluster 4, ada 4 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.31 Data Anggota *Cluster* 4

Data anggota	Nilai
4	52.79217
6	52.6505
20	52.21384
21	53.15075

Nk =4

Jumlah nilai = 210.8073

Rata rata = Jumlah nilai/Nk = 52.70182

Untuk cluster 5, ada 4 data yang tergabung di dalamnya sebagai berikut :

Tabel 3.32 Data Anggota *Cluster* 5

Data anggota	Nilai
11	23.77537
13	35.65105

Nk =2

Jumlah nilai = 59.42642

Rata rata = Jumlah nilai/Nk = 29.71321

Nilai *centroid* terakhir yang didapat adalah sebagai berikut :

Tabel 3.33 *Centroid* Terakhir

Centroid	Nilai Centroid
0	46.4182
1	48.24625
2	56.54964
3	71.37132
4	52.70182
5	29.71321

3.4.3 Learning Vector Quantization

Learning Vector Quantization (LVQ) digunakan sebagai klasifikator, setiap *centroid* hasil *clustering* dari k-mean digunakan sebagai 1 vektor nilai bobot sehingga terdapat 6 vektor nilai bobot untuk 1 *record* di data latih, pada proses pembelajaran LVQ jumlah target kelas ditentukan di awal, pada penelitian ini target kelas merupakan identitas untuk 1 orang pembicara, keterangan terkait target kelas dapat dilihat pada tabel berikut :

Tabel 3.34 Target Kelas

Target Kelas	Keterangan
1	Speaker 1
2	Speaker 2
3	Speaker 3
4	Speaker 4

Data latih yang digunakan sebagai berikut ;

Tabel 3.35 Data Latih

Variabel	Identitas	Nilai Bobot	Target Kelas
X1	User1	58.49, 35.94, 77.02, 37.84, 58.79, 81.74	1
X2	User1	53.22, 37.86, 75.01, 35.87, 60.47, 76.24	1
X3	User1	45.16, 47.38, 59.84, 52.12, 51.52, 69.22	1
X4	User1	40.74, 35.35, 66.67, 36.32, 68.39, 69.53	1
X5	User1	62.36, 49.37, 75.84, 45.49, 60.35, 68.18	1
X6	User1	43.62, 34.51, 56.66, 37.28, 64.16, 74.13	1
X7	User2	51.04, 96.62, 83.13, 35.99, 87.12, 48.18	2
X8	User2	43.69, 80.65, 78.29, 33.97, 98.78, 35.93	2
X9	User2	45.52, 61.93, 36.61, 55.46, 79.69, 57.55	2
X10	User2	46.73, 40.13, 39.17, 77.53, 82.63, 48.39	2
X11	User2	68.82, 70.64, 53.36, 53.49, 68.95, 80.53	2
X12	User2	66.04, 75.85, 76.42, 55.38, 79.02, 71.38	2
X13	User3	60.38, 52.16, 45.61, 80.32, 73.93, 60.96	3
X14	User3	60.42, 40.37, 56.39, 49.42, 75.83, 62.63	3
X15	User3	40.46, 56.54, 46.72, 75.37, 62.37, 66.82	3
X16	User3	57.33, 62.83, 42.28, 48.27, 82.38. 62.85	3

Variabel	Identitas	Nilai Bobot	Target Kelas
X17	User3	61.32, 83.53, 72.38, 43.04, 83.20, 72.69	3
X18	User3	67.36, 72.38, 55.67, 51.84, 73.26, 63.87	3
X19	User4	40.43, 75.32, 60.31, 70.17, 44.83, 53.24	4
X20	User4	76.68, 75.68, 37.61, 49.29, 54.93, 58.83	4
X21	User4	65.69, 45.41, 38.38, 44.47, 44.39, 76.72	4
X22	User4	38.67, 71.61, 59.46, 80.14, 66,34, 60.98	4
X23	User4	69.55, 79.26, 57.43, 76.86, 43.08, 48.82	4
X24	User4	72.04, 50.88, 52.62, 40.74, 82.31, 52.27	4

Pada penelitian ini bobot awal diambil berdasarkan inputan pertama untuk tiap target kelas, yang digunakan sebagai inisiasi bobot awal yaitu X1 untuk target kelas 1, X7 untuk target kelas 2, X13 untuk target kelas 3 dan X19 untuk target kelas 4, bobot awal dapat dilihat pada tabel 3.36.

Tabel 3.36 Bobot Awal

No	Variabel	Nilai Bobot	Kelas
1	X1	58.49, 35.94, 77.02, 37.84, 58.79, 81.74	1
2	X7	51.04, 36.62, 83.13, 35.99, 87.12, 48.18	2
3	X13	60.38, 52.16, 45.61, 80.32, 73.93, 60.96	3
4	X19	40.43, 75.32, 60.31, 70.17, 44.83, 53.24	4

Tetapkan parameter:

$\alpha = 0{,}05$

dec $\alpha = 0{,}1$

maksimum epoh = 88.

Pembelajaran LVQ pada data X2 sebagai berikut :

X2 = 53.22, 37.86, 75.01, 35.87, 60.47, 76.24

Epoh ke-1

Lakukan perhitungan jarak menggunakan persamaan 2.15.

Jarak X2 ke kelas 1

$$d(X_2 - Y_1) = \sqrt{(X2_{x1} - Y1_{y1})^2 + \cdots + (X2_{xn} - Y1_{yn})^2}$$

$$= \sqrt{\begin{matrix}(53.22 - 58{,}49)^2 + (37.86 - 35{,}94)^2 + (75.015 - 77{,}02)^2 + \\ (35.87 - 37{,}846)^2 + (60.47 - 58{,}79)^2 + (76.24 - 81{,}74)^2\end{matrix}}$$

$$= \sqrt{(-5.261)^2 + (1.92)^2 + (-2.05)^2 + (-1.969)^2 + (1.677)^2 + (-5.5)^2}$$

$$= \sqrt{27.678121 + 3.6864 + 4.2025 + 3.876961 + 2.812329 + 30.25}$$

$$= \sqrt{72.506311} = 8.51506377$$

Jarak X2 ke kelas 2

$$d(X_2 - Y_2) = \sqrt{(X2_{x1} - Y2_{y1})^2 + \cdots + (X2_{xn} - Y2_{yn})^2}$$

$$= \sqrt{\begin{matrix}(53.22 - 51.04)^2 + (37.86 - 36.62)^2 + (75.01 - 83.13)^2 + \\ (35.87 - 35.99)^2 + (60.47 - 87.12)^2 + (76.24 - 48.18)^2\end{matrix}}$$

$$= \sqrt{\begin{matrix}(2.225)^2 + (1.233)^2 + (-8.122)^2 + (-0.12)^2 + (-26.651)^2 + \\ (28.053)^2\end{matrix}}$$

$$= \sqrt{\begin{matrix}4.950625 + 1.520289 + 65.966884 + 0.0144 + 710.275801 + \\ 786.970809\end{matrix}}$$

$$= \sqrt{1569.698808} = 39.61942463$$

Jarak X2 ke kelas 3

$$d(X_2 - Y_3) = \sqrt{(X2_{x1} - Y3_{y1})^2 + \cdots + (X2_{xn} - Y3_{yn})^2}$$

$$= \sqrt{\begin{matrix}(53.22 - 60.38)^2 + (37.86 - 60.38)^2 + (75.01 - 60.38)^2 + \\ (35.87 - 60.38)^2 + (60.47 - 73.93)^2 + (76.24 - 60.96)^2\end{matrix}}$$

$$= \sqrt{\begin{matrix}(-7.16)^2 + (-14.3)^2 + (29.4)^2 + (-44.45)^2 + (-13.46)^2 + \\ (15.28)^2\end{matrix}}$$

$$= \sqrt{\begin{matrix}51.2656 + 204.49 + 864.36 + 1975.8025 + 181.1716 + \\ 233.4784\end{matrix}}$$

$$= \sqrt{3510.5681} = 59.25004726$$

Jarak X2 ke kelas 4

$$d(X_2 - Y_4) = \sqrt{(X2_{x1} - Y4_{y1})^2 + \cdots + (X2_{xn} - Y4_{yn})^2}$$

$$= \sqrt{\begin{matrix}(53.22 - 40.43)^2 + (37.86 - 75.32)^2 + (75.01 - 60.31)^2 + \\ (35.87 - 70.17)^2 + (60.47 - 44.83)^2 + (76.24 - 53.24)^2\end{matrix}}$$

$$= \sqrt{(12.79 + (-37.46)^2 + (14.7)^2 + (-34.3)^2 + (15.64)^2 + (23)^2}$$

$$= \sqrt{163.5841 + 1403.2516 + 216.09 + 1176.49 + 244.6096 + 529}$$

$$= \sqrt{3733.0253} = 61.09848852$$

Jarak terkecil bobot adalah ke kelas 1, maka bobot kelas 1 baru :

$$w11(baru) = w11(lama) + \alpha(\text{x11} - \text{w11(lama)})$$

$$= 58{,}49 + 0{,}05 * (53.229 - 58{,}49)$$

$$= 58{,}49 + (-0.26305) = 58.22695$$

$$w12(baru) = w12(lama) + \alpha(\text{x12} - \text{w12(lama)})$$

$$= 35{,}94 + 0{,}05 * (37.860 - 35{,}94)$$

$$= 35{,}94 + 0.096 = 36.036$$

$$w13(baru) = w13(lama) + \alpha(\text{x13} - \text{w13(lama)})$$

$$= 77{,}02 + 0{,}05 * (75.015 - 77{,}02)$$

$$= 35{,}94 + (-0.1025) = 76.9175$$

$$w14(baru) = w14(lama) + \alpha(\text{x14} - \text{w14(lama)})$$

$$= 37{,}846 + 0{,}05 * (35.877 - 37{,}846)$$

$$= 35{,}94 + (-0.09845) = 37.74755$$

$$w15(baru) = w15(lama) + \alpha(\text{x15} - \text{w15(lama)})$$

$$= 58{,}79 + 0{,}05 * (60.471 - 58{,}79)$$

$$= 58{,}79 + 0.08385 = 58.87385$$

$$w16(baru) = w16(lama) + \alpha(\text{x16} - \text{w16(lama)})$$

$$= 81{,}74 + 0{,}05 * (76.24 - 81{,}74)$$

$$= 81{,}74 + (-0.275) = 81.465$$

Pembelajaran LVQ data X2 pada epoh ke -1 selesai dengan bobot W1 pada epoh ke-1 adalah (58.22695, 36.036, 76.9175, 37.74755, 58.87385, 81.465). Lakukan hal sama seperti yang dilakukan pada data X2 terhadap data yang lainnya.

Setelah melakukan hal yang sama terhadap data yang lainnya maka didapatkan bobot akhir pada epoh ke-1 ysng ditampilkan pada Tabel 3.37.

Tabel 3.37 Bobot akhir epoh ke-1

No	Bobot Akhir	Nilai Bobot	Kelas
1	Y1	12.939087811986, 9.92973887280915, 8.74308214096916, 11.4843407279255, 9.33560096494201, 10.5954505604721	1
2	Y2	2.4145663118208, 8.14638546215996, 10.0215189868059, 9.34239703703147, 9.34239703703147, 10.9579174467273	2
3	Y3	9.15934157754774, 10.4998770481304, 2.9587909842855, 2.9587909842855, 8.28960377336395, 9.78803978814639	3
4	Y4	8.23576795168937, 12.2047747717045, 7.17311383474033, 9.81136715564363, 10.7230927902271, 9.04611584641443	4

Epoh ke-1 selesai, masuk ke epoh ke-2, pada epoh ke-2 α di update sebagai berikut :

$$\alpha = \alpha - \text{dec}\,\alpha * \alpha(\text{lama}) = 0.05 - 0.1 * 0.05 = 0.045$$

Lakukan hal yang sama untuk epoh selanjutnya sampai epoh maksimum. Maka di dapatkan bobot terakhir sebagai model *pattern* yang dibutuhkan untuk pencocokan suara.

Data bobot terakhir ditampilkan pada Tabel.3.38

Tabel 3.38 Data *pattern*

No	Bobot Akhir	Nilai Bobot	Kelas
1	Y1	10.16010935, 10.8076543681389, 9.98830598633498, 10.8241976777701, 9.097238566, 10.61104262	1
2	Y2	12.5966045627762, 10.14896, 12.0408276144914, 9.71670684750022, 11.6368381560248, 8.52768289792408	2
3	Y3	11.65296786895, 9.63268678103745, 13.0561636194078, 10.1573759729081, 9.10111676657628, 10.7680066696559	3

No	Bobot Akhir	Nilai Bobot	Kelas
4	Y4	8.30660160210718, 13.1338127925847, 9.18354449040888, 10.5899287053484, 9.83181500195254, 11.5703259731386	4

3.4.4 Euclidean Distance

Untuk dapat melakukan verifikasi suara, maka data signal baru yang masuk akan dicocokan dengan data yang telah ada dalam *database* sebelumnya yang sudah dilakukan pembelajaran oleh LVQ. Setiap vektor dari model yang diujicobakan, dibandingkan dan dihitung *euclidean distance* dengan semua vektor yang ada pada salah satu model *database*.

Data uji pencocokan suara ditampilkan pada Tabel 3.39 dan data pattern ditampilkan pada Tabel 3.39.

Tabel 3.39 Data Uji

Variabel	Identitas	Nilai Bobot	Target Kelas
Z1	User1	8.83450552130402, 13.1594266247076, 11.6300375755923, 9.36672086186148, 10.5963046433056, 9.92231103365605	1
Z2	User1	9.15153531410966, 12.0230971403831, 9.71482093661501, 11.0418541606356, 10.3039656937224, 13.5477686584838	1
Z3	User2	12.7729754032668, 9.60837349264792, 10.2786802564153 ,8.48269866209621, 11.158131937788, 9.06028184869652	2
Z4	User3	8.14638546215996, 10.9579174467273, 12.4145663118208, 9.34239703703147, 10.0215189868059, 8.76695323944555	3
Z5	User4	9.15934157754774, 8.28960377336395, 11.4209936378861, 9.78803978814639, 12.9587909842855, 10.4998770481304	4
Z6	User2	10.2786802564153, 12.7729754032668, 8.48269866209621, 11.158131937788, 9.06028184869652, 9.60837349264792	2

Variabel	Identitas	Nilai Bobot	Target Kelas
Z7	User4	11.4209936378861, 10.4998770481304, 9.78803978814639, 12.9587909842855, 8.28960377336395, 9.15934157754774	4
Z8	User3	8.14638546215996, 10.9579174467273, 12.4145663118208, 9.34239703703147, 10.0215189868059, 8.76695323944555	3

Tabel 3.40 Data *pattern*

No	Bobot Akhir	Nilai Bobot	Kelas
1	Y1	10.16010935, 10.8076543681389, 9.98830598633498, 10.8241976777701, 9.097238566, 10.61104262	1
2	Y2	12.5966045627762, 10.14896, 12.0408276144914, 9.71670684750022, 11.6368381560248, 8.52768289792408	2
3	Y3	11.65296786895, 9.63268678103745, 13.0561636194078, 10.1573759729081, 9.10111676657628, 10.7680066696559	3
4	Y4	8.30660160210718, 13.1338127925847, 9.18354449040888, 10.5899287053484, 9.83181500195254, 11.5703259731386	4

Pada pencocokan suara akan dibandingkan data uji (X) dengan data *pattern* (Y) menggunakan perhitungan jarak terdekat *euclidean distance* dengan persamaan 2.15.

Perhitungan jarak pada data uji Z1 dengan semua data *pattern* (Y1, Y2, Y3 dan Y4) sebagai berikut :

Z1 = 8.83450552130402, 13.1594266247076, 11.6300375755923, 9.36672086186148, 10.5963046433056, 9.92231103365605.

Jarak Z1 ke Y1 :

$$d(Z_1 - Y_1) = \sqrt{(Z1_{x1} - Y1_{y1})^2 + \cdots + (Z1_{xn} - Y1_{yn})^2}$$

$$= \sqrt{\begin{matrix}(8.83450552130402 - 10.16010935)^2 + \cdots + \\ (9.92231103365605 - 10.61104262)^2\end{matrix}}$$

$$= 8.96438$$

Jarak Z1 ke Y2:

$$d(Z_1 - Y_2) = \sqrt{(Z1_{x1} - Y2_{y2})^2 + \cdots + (Z1_{xn} - Y2_{yn})^2}$$

$$= \sqrt{\begin{matrix}(8.83450552130402 - 12.5966045627762)^2 + \cdots + \\ (9.92231103365605 - 8.52768289792408)^2\end{matrix}}$$

$$= 9.968507$$

Jarak Z1 ke Y3:

$$d(Z_1 - Y_3) = \sqrt{(Z1_{x1} - Y3_{y1})^2 + \cdots + (Z1_{xn} - Y3_{yn})^2}$$

$$= \sqrt{\begin{matrix}(8.83450552130402 - 1.65296786895)^2 + \cdots + \\ (9.92231103365605 - 10.7680066696559)^2\end{matrix}}$$

$$= 13.264847$$

Jarak Z1 ke Y4:

$$d(Z_1 - Y_4) = \sqrt{(Z1_{x1} - Y4_{y1})^2 + \cdots + (Z1_{xn} - Y4_{yn})^2}$$

$$= \sqrt{\begin{matrix}(8.83450552130402 - 8.30660160210718)^2 + \cdots + \\ (9.92231103365605 - 11.5703259731386)^2\end{matrix}}$$

$$= 10.735293$$

Jarak terdekat adalah ke Y1, data uji Z1 target kelasnya = 1, data *pattern* Y1 target kelasnya =1. Maka kemungkinan Z1 merupakan merupakan *speaker* 1.

3.5 Analisis Kebutuhan Non Fungsional

Analisis non fungsional yang dilakukan dibagi ke dalam 2 tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

3.5.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras pada penelitian ini merupakan kebutuhan perangkat keras yang digunakan dalam pembangunan perangkat lunak ini. Adapun perangkat keras yang digunakan sebagai berikut:

a. Processor 2.10 GHz
b. RAM 2 GB
c. HDD 320GB
d. *Keyboard*
e. *Mouse*
f. *Microphone*
g. *Speaker*

3.5.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak pada penelitian ini merupakan *tools* yang digunakan untuk pembangunan aplikasi ini. *Tools* yang digunakan adalah sebagai berikut:

a. OS Windows 7
b. Microsoft Visual Studio 2010
c. MySQL

3.5.3 Analisis Pengguna

Spesifikasi pengguna yang dapat menjalankan aplikasi yang akan dibangun hanya terdapat satu jenis user yaitu seorang penguji. Spesifikasi pengguna yang dibutuhkan adalah sebagai berikut :

a. Menguasai dalam penggunaan komputer.
b. Mengerti secara teknis mengenai *speaker recognition*.
c. Mengerti maksud penggunaan MFCC, k-means *clustering* dan LVQ.

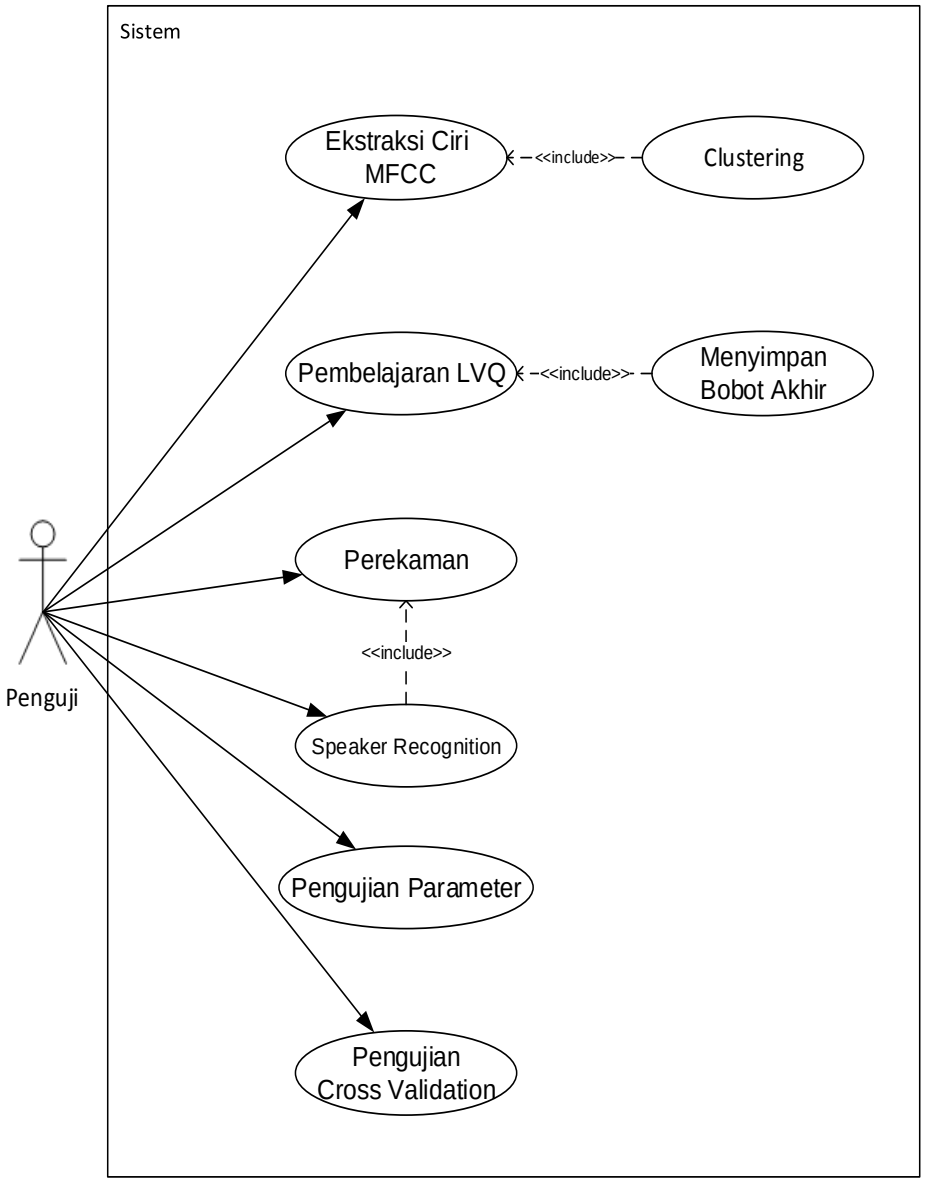
3.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menganalisis proses yang diterapkan dalam perangkat lunak dan menjelaskan kebutuhan yang diperlukan.

Perangkat lunak ini dimodelkan dengan menggunakan model *unified modeling language* (UML). Bagian-bagian yang dilakukan dalam analisis tersebut antara lain:

3.6.1 Use Case Diagram

Use case diagram menggambarkan proses yang ada dalam sistem, Gambar 3.6 adalah *use case* implementasi metode MFCC, *k-mean clustering* dan LVQ pada pengenalan suara:



Gambar 3.6 *Use Case Diagram*

3.6.2 Use Case Scenario

Use Case Scenario menjelaskan skenario untuk tiap *use case* yang menggambarkan urutan interaksi aktor dengan *use case* tersebut.

Tabel 3.41 *Use Case Scenario* Ekstraksi Ciri MFCC

Nama *Use Case*	Ektraksi Ciri MFCC.
Tujuan	Mendapatkan hasil ektraksi ciri MFCC.
Deskripsi	Fungsionalitas ini digunakan penguji untuk mendapatkan hasil ekstraksi ciri MFCC.
Aktor	Penguji
Related Case	-
Kondisi Sukses	Berhasil mendapatkan hasil ekstraksi ciri MFCC.
Kondisi Gagal	Gagal mendapatkan hasil ekraksi ciri MFCC.
Include Case	-
Trigger	*Open wave*
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol ekstraksi ciri MFCC.	
	2.Menampilkan menu ektraksi ciri MFCC.
3.Aktor menekan tombol *open wave*.	
	4.Menampilkan menu untuk memilih data suara.
5.Aktor memilih data suara.	
	6.Menampilkan grafik sinyal dari data suara yang dipilih.
	7.Menampilkan data sinyal dan hasil ektraksi ciri suara MFCC yang meliputi tahapan DC *removal*, *pre emphasize*, *frame blocking*, *windowing*, FFT, *filterbank*, DCT dan *ceptral liftering* yang ditampilkan pada satu tabel.

Tabel 3.42 *Use Case Scenario Clustering*

Nama *Use Case*	*Clustering*.
Tujuan	Mendapatkan hasil *clustering* dari hasil ektraksi ciri MFCC.
Deskripsi	Fungsionalitas ini digunakan penguji untuk mendapatkan hasil *clustering* dari ektraksi ciri MFCC.
Aktor	Penguji
Related Case	Ektrasi ciri MFCC
Kondisi Sukses	Berhasil mendapatkan hasil *clustering* dari ektraksi ciri MFCC.
Kondisi Gagal	Gagal mendapatkan hasil *clustering* dari ektraksi ciri MFCC.
Include Case	-
Trigger	*Open wave, Clustering.*
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol *clustering*.	
	2.Menampilkan titik pusat hasil *clustering* yang ditampilkan pada sebuah tabel.

Tabel 3.43 *Use Case Scenario* Pembelajaran LVQ

Nama *Use Case*	Pembelajaran LVQ.
Tujuan	Mendapatkan *pattern* dari hasil pembelajaran LVQ.
Deskripsi	Fungsionalitas ini digunakan penguji untuk mendapatkan *pattern* dari hasil pembelajaran LVQ.
Aktor	Penguji
Related Case	Data Latih
Kondisi Sukses	Berhasil mendapatkan *pattern* dari hasil pembelajaran LVQ.
Kondisi Gagal	Gagal mendapatkan *pattern* dari hasil pembelajaran LVQ.
Include Case	-
Trigger	Pembelajaran
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol pembelajaran LVQ	
	2.Menampilkan menu pembelajaran LVQ
3.Aktor menekan tombol pembelajaran.	
	4.Menampilkan tabel data latih.
	5.Menampilkan hasil akhir dari pembelajaran LVQ

Tabel 3.44 *Use Case Scenario* Menyimpan Bobot Akhir

Nama *Use Case*	Menyimpan bobot akhir.
Tujuan	Menyimpan bobot akhir.
Deskripsi	Fungsionalitas ini digunakan penguji untuk menyimpan bobot akhir.
Aktor	Penguji
Related Case	Pembelajaran LVQ
Kondisi Sukses	Berhasil menyimpan bobot akhir.
Kondisi Gagal	Gagal menyimpan bobot akhir..
Include Case	-
Trigger	Simpan
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol simpan bobot akhir	
	2.Menyimpan data bobot akhir ke *database*.

Tabel 3.45 *Use Case Scenario* Perekaman

Nama *Use Case*	Perekaman
Tujuan	Menyimpan hasik perekaman suara.
Deskripsi	Fungsionalitas ini digunakan penguji untuk menyimpan hasik perekaman suara..
Aktor	Penguji
Related Case	Uji
Kondisi Sukses	Berhasil menyimpan hasik perekaman suara..
Kondisi Gagal	Gagal menyimpan hasik perekaman suara..
Include Case	-
Trigger	Rekam
Skenario Utama	
Aksi aktor	Reaksi Sistem

1.Aktor menekan tombol Rekam.	
	2.Menampilkan menu rekam.
3.Aktor menekan Cek Device	
	4.Menampilkan *microphone* yang tersedia.
5.Aktor memilih *microphone*	
6.Aktor menekan tombol rekam	
	7.Menampilkan menu untuk menyimpan file hasil rekaman.
8.Aktor menginputkan suara.	
9.Aktor menekan tombol simpan.	
	10. Menyimpan hasil rekaman.

Tabel 3.46 *Use Case Scenario Speaker Recognition*

Nama *Use Case*	Recognition.
Tujuan	Mendapatkan hasil pencocokan suara.
Deskripsi	Fungsionalitas ini digunakan penguji untuk mendapatkan hasil pencocokan suara
Aktor	Penguji
Related Case	Perekaman
Kondisi Sukses	Berhasil mendapatkan hasil pencocokan suara
Kondisi Gagal	Gagal mendapatkan hasil pencocokan suara
Include Case	-
Trigger	*Recognition*
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol *recogniton.*	
	2.Menampilkan gambar sinyal dari data suara.
	3.Mendapatkan hasil ektraksi ciri MFCC dan *k-means clustering*.
	4. Menampilkan titik pusat hasil *k-means clustering*.
	5.Menampilkan hasil pencocokan suara

Tabel 3.47 *Use Case Scenario* Pengujian Parameter

Nama *Use Case*	Pengujian parameter
Tujuan	Mendapatkan hasil pengujian parameter.
Deskripsi	Fungsionalitas ini digunakan penguji untuk mendapatkan hasil pengujian parameter.
Aktor	Penguji
Related Case	-
Kondisi Sukses	Berhasil mendapatkan hasil pengujian parameter.
Kondisi Gagal	Gagal mendapatkan hasil pengujian parameter.
Include Case	-
Trigger	Pengujian parameter
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol pengujian.	
	2.Menampilkan tampilan pengujian.
3.Aktor menginputkan nilai *learning rate*.	
4.Aktor menginputkan nilai penurunan *learning rate*.	

5.Aktor menginputkan nilai maksimum epoh.	
6.Aktor menekan tombol Uji	
	7.Menampilkan hasil pengujian parameter.

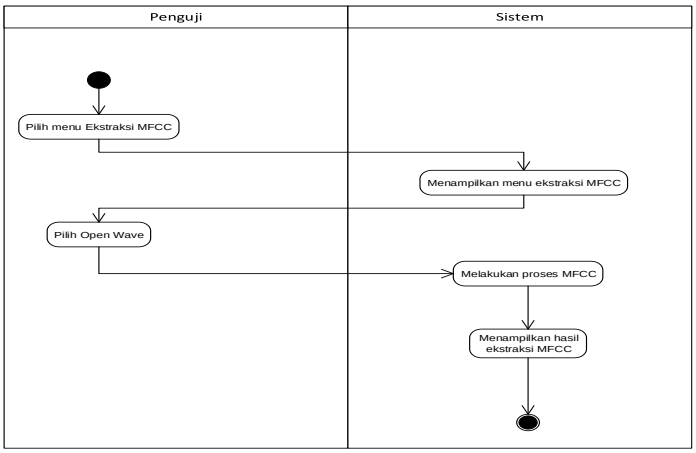
Tabel 3.48 *Use Case Scenario* Pengujian *Cross Validation*

Nama *Use Case*	Pengujian cross validation
Tujuan	Mendapatkan hasil pengujian cross validation.
Deskripsi	Fungsionalitas ini digunakan penguji untuk mendapatkan hasil pengujian cross validation.
Aktor	Penguji
Related Case	-
Kondisi Sukses	Berhasil mendapatkan hasil pengujian cross validation.
Kondisi Gagal	Gagal mendapatkan hasil pengujian cross validation.
Include Case	-
Trigger	Pengujian cross validation
Skenario Utama	
Aksi aktor	Reaksi Sistem
1.Aktor menekan tombol pengujian cross validation.	
	2.Menampilkan tampilan pengujian cross validation.
3.Aktor menekan tombol fold 1, fold 2, sampai dengan fold 10.	
	4.Menampilkan hasil pengujian tiap fold.
5.Aktor menekan tombol hasil.	
	6.Menampilkan rata-rata dari hasil akhir seluruh pengujian.

3.6.3 Activity Diagram

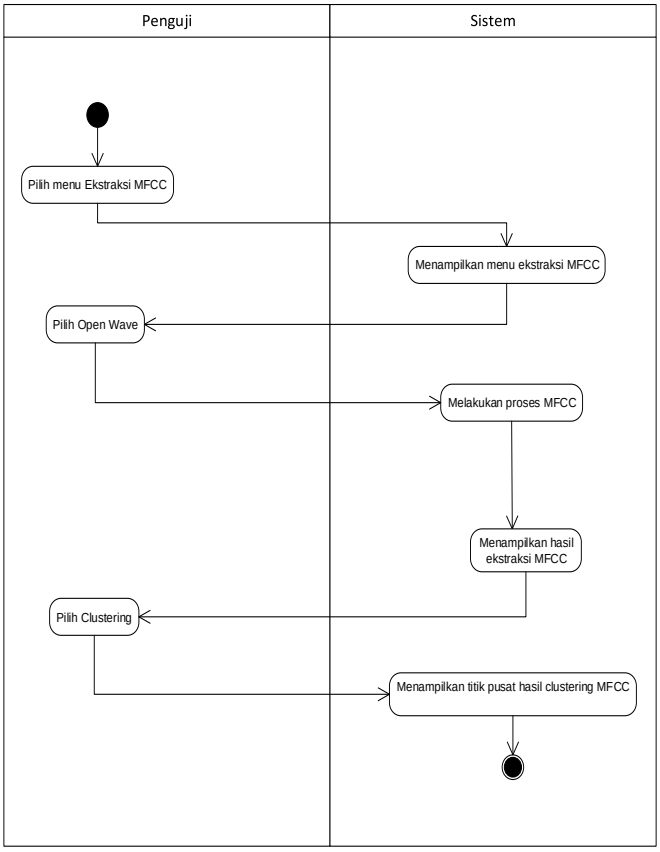
Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, decision (pilihan), yang mungkin terjadi dan bagaimana berakhir. Berikut *activity diagram* :

1. *Activity Diagram* Ekstraksi Ciri MFCC



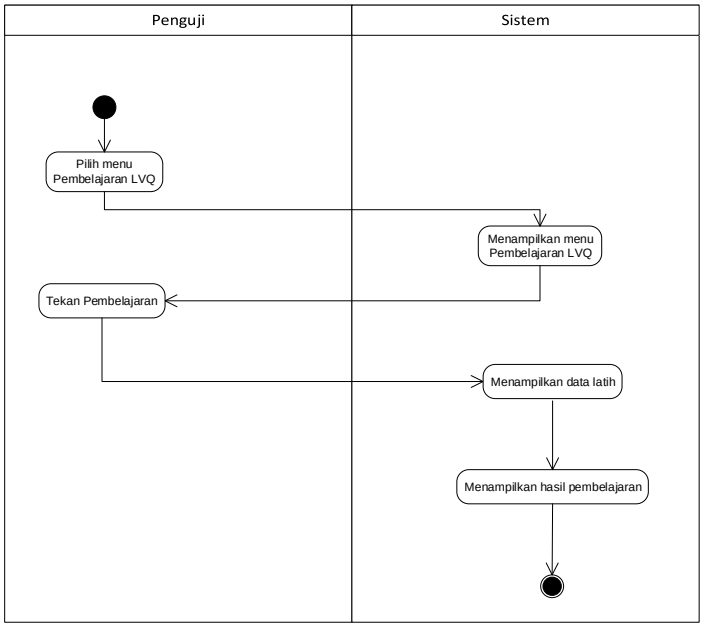
Gambar 3.7 *Activity Diagram* Ekstraksi Ciri MFCC

2. *Activity Diagram Clustering*



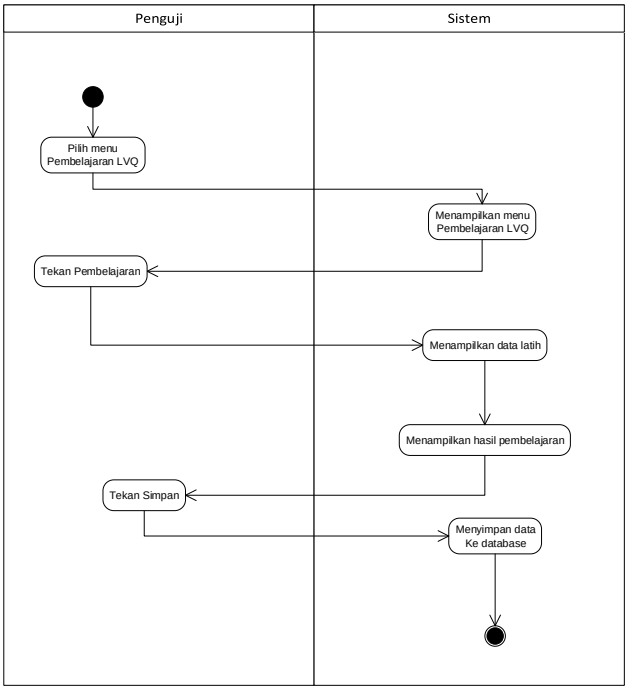
Gambar 3.8 *Activity Diagram Clustering*

3. *Activity Diagram* Pembelajaran LVQ



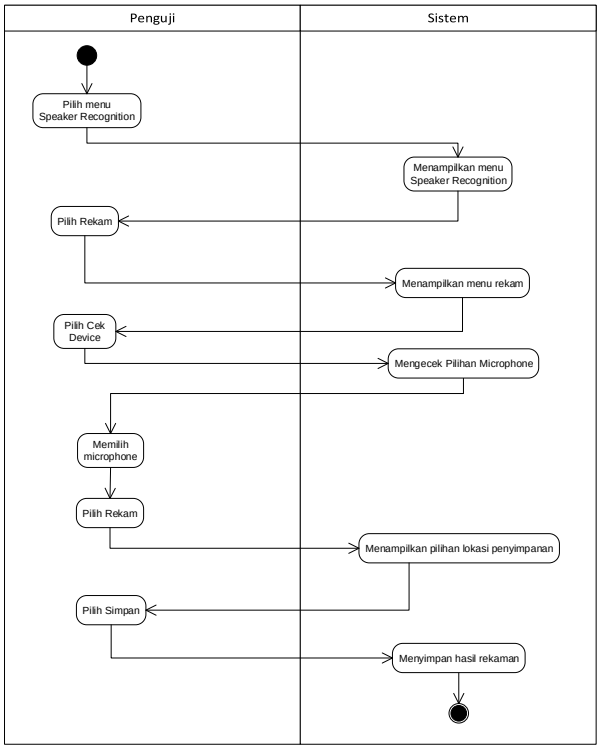
Gambar 3.9 Activity Diagram Pembelajaran

4. *Activity Diagram* Menyimpan Bobot Akhir



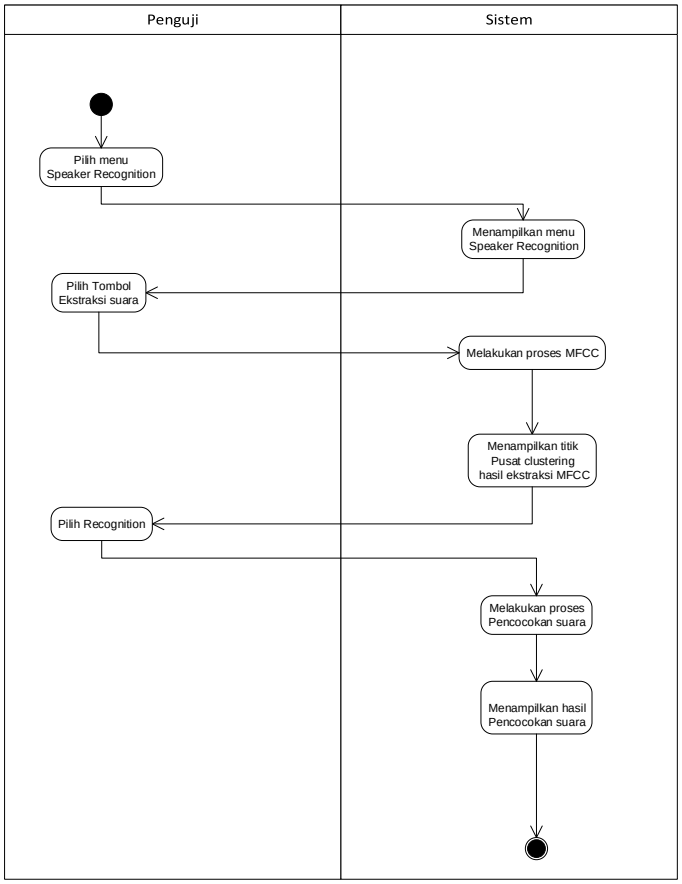
Gambar 3.10 Activity Diagram Simpan Bobot Akhir

5. *Activity Diagram* Perekaman



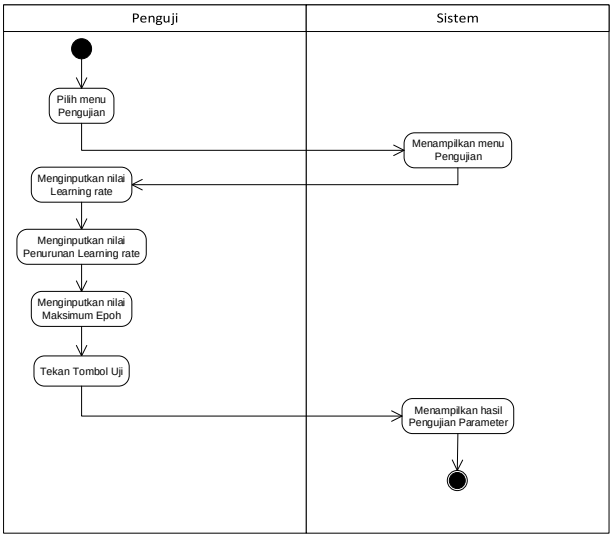
Gambar 3.11 Activity Diagram Perekaman

6. *Activity Diagram Recognition*



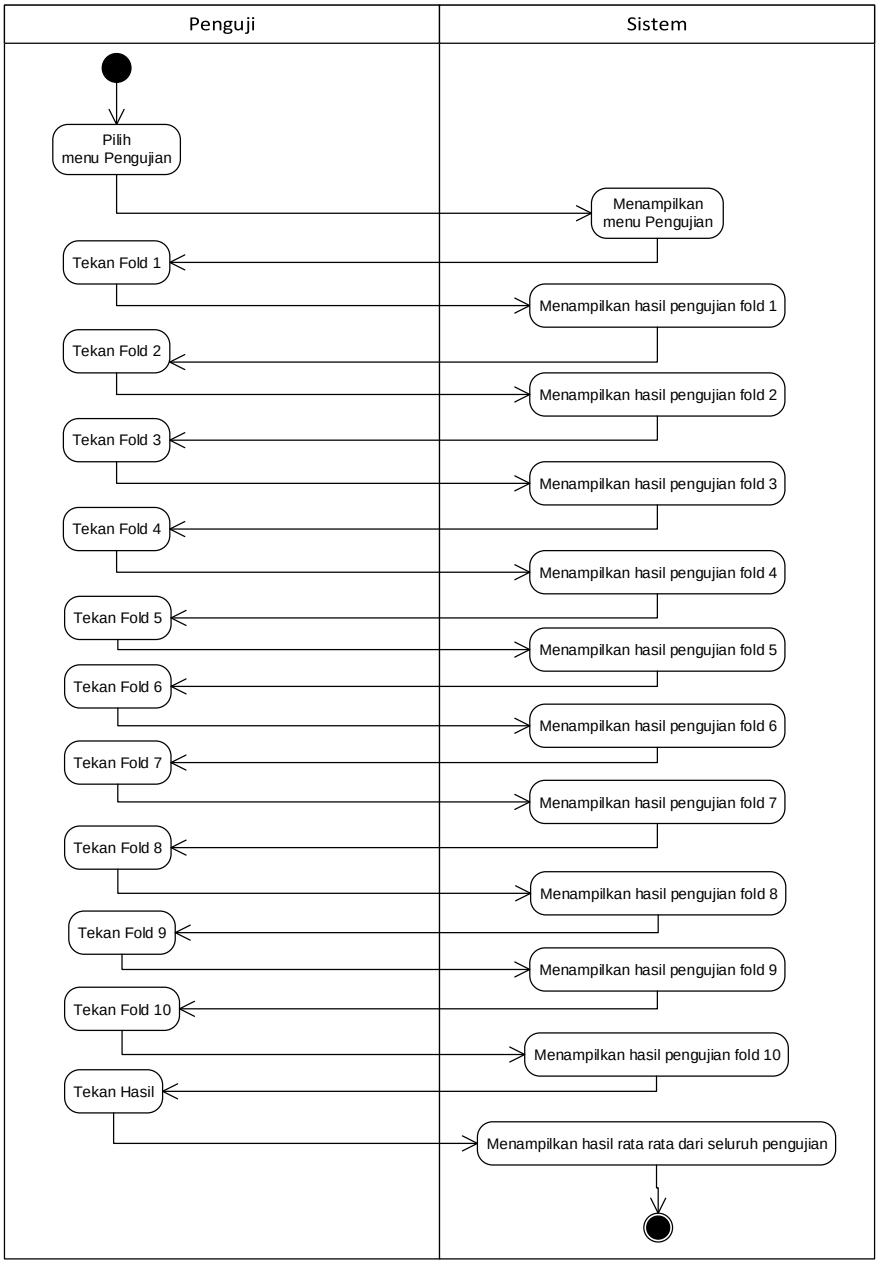
Gambar 3.12 Activity Diagram *Recognition*

7. *Activity Diagram Pengujian Parameter*



Gambar 3.13 *Activity Diagram* Pengujian Parameter

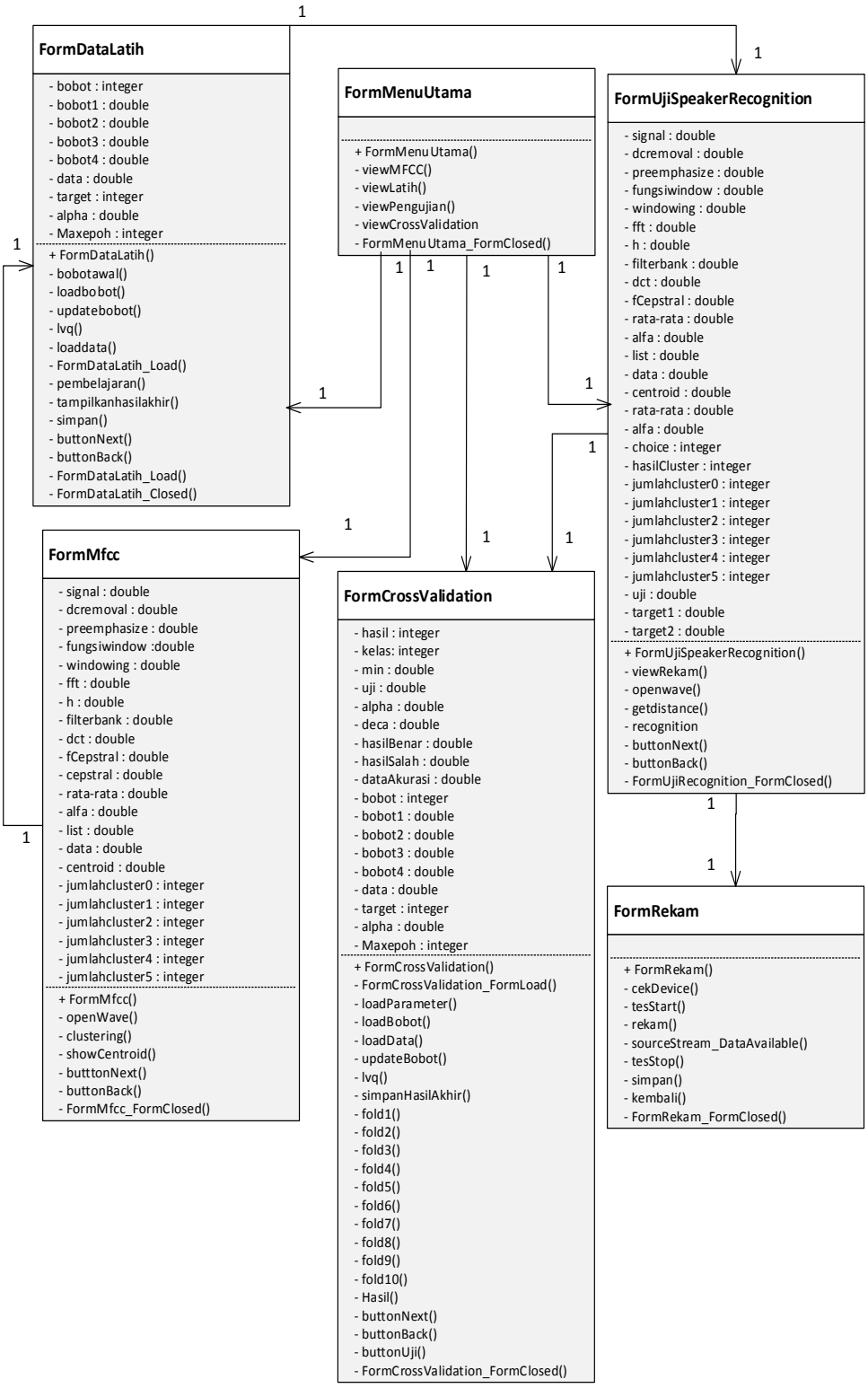
8. *Activity Diagram* Pengujian *Cross Validation*



Gambar 3.14 *Activity Diagram* Pengujian *Cross Validation*

3.6.4 Class Diagram

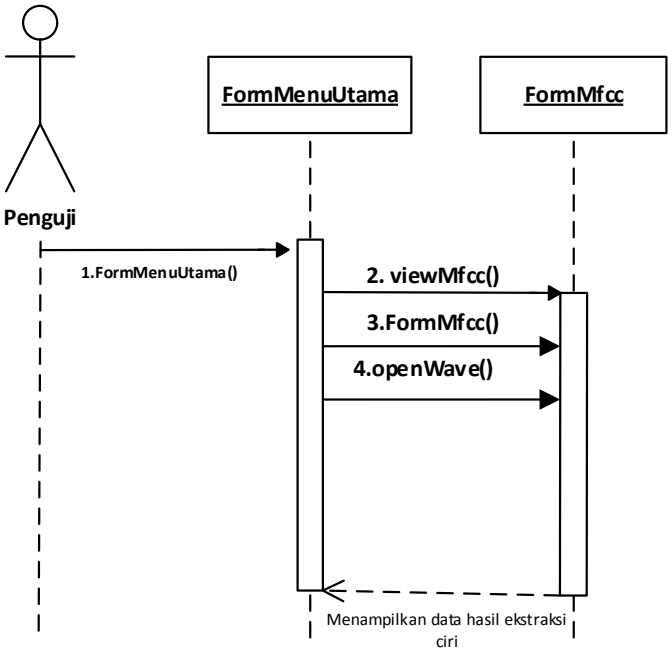
Class diagram adalah sebuah spesifikasi yang dapat menghasilkan sebuah objek dan merupakan inti dari desain berorientasi objek. *Class diagram* dapat dilihat pada Gambar 3.15.



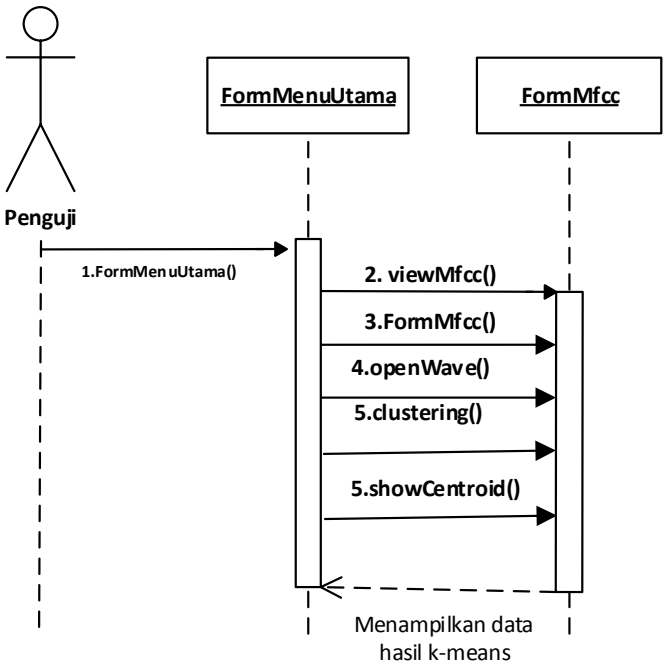
Gambar 3.15 Class Diagram

3.6.5 Sequence Diagram

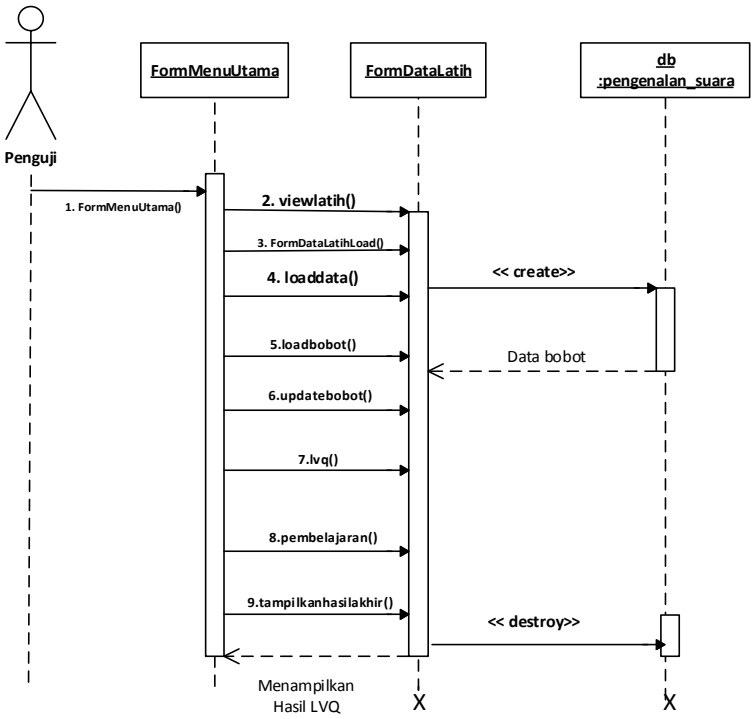
Berikut ini adalah *sequence diagram* dari pembangunan aplikasi pengenalan suara :



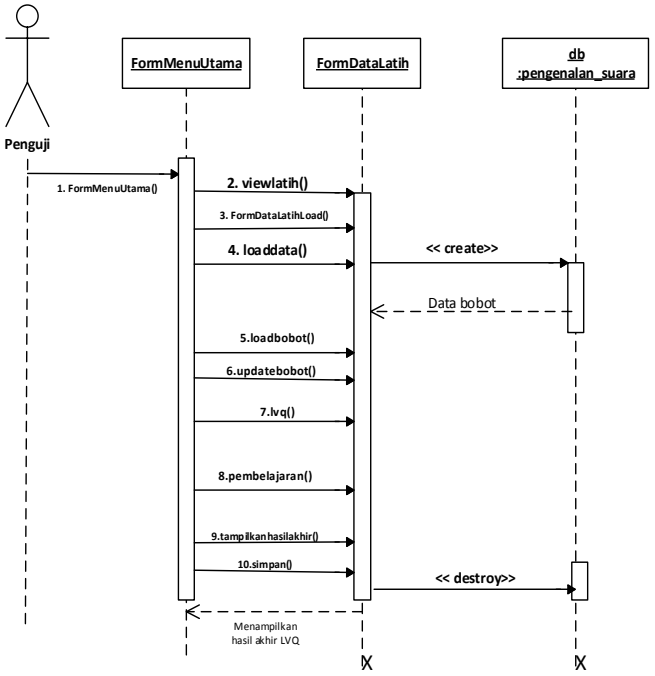
Gambar 3.16 Sequence Diagram Ekstraksi Ciri MFCC



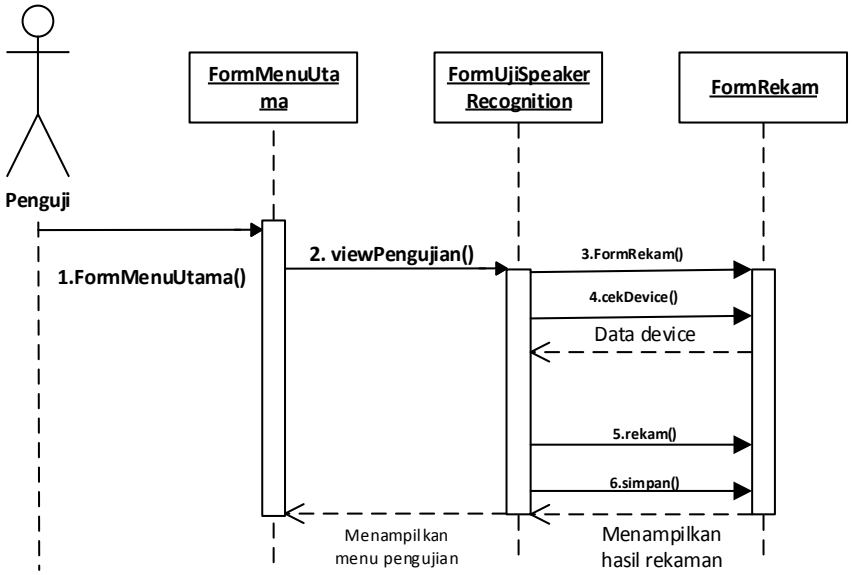
Gambar 3.17 Sequence Diagram Clustering



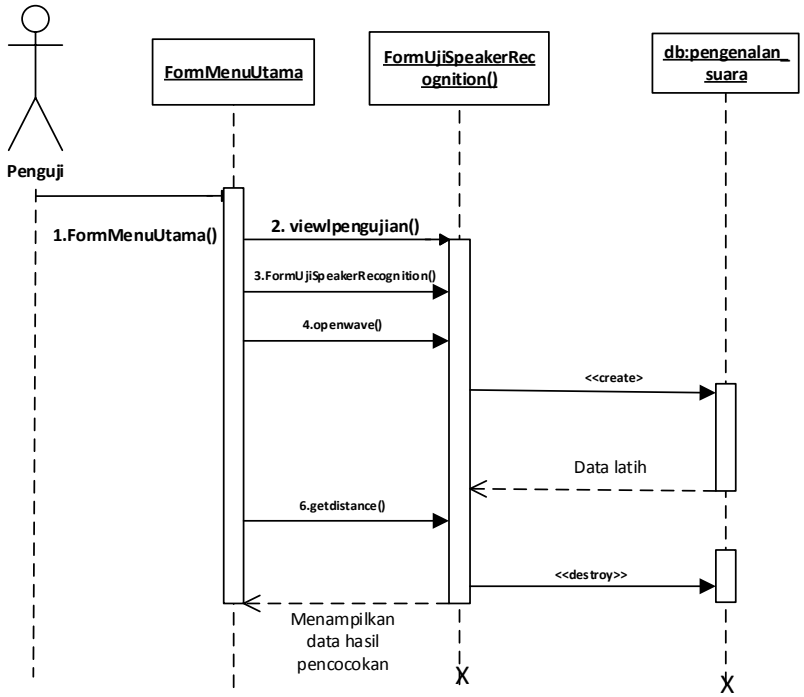
Gambar 3.18 *Sequence Diagram* Pembelajaran LVQ



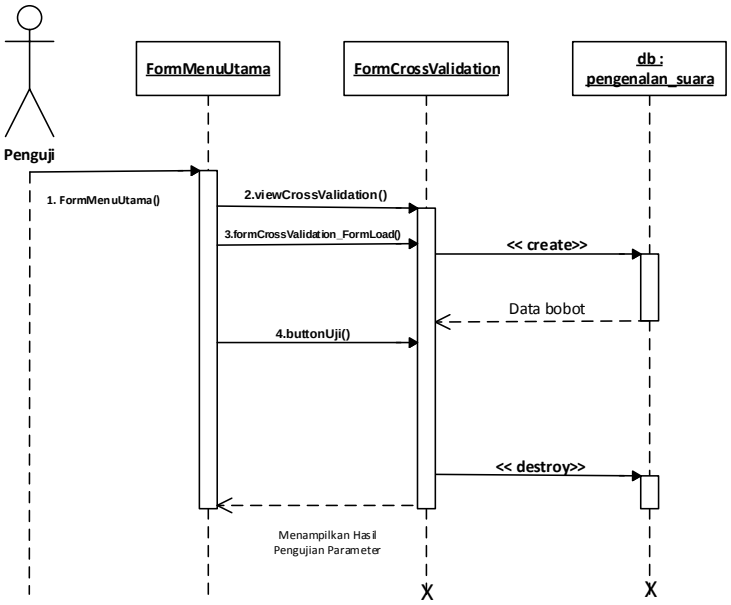
Gambar 3.19 *Sequence Diagram* Menyimpan Bobot Akhir



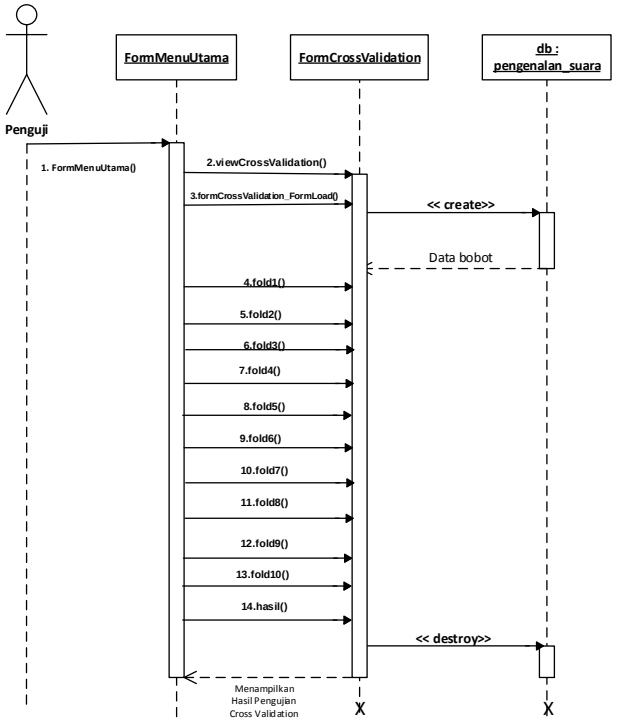
Gambar 3.20 *Sequence Diagram* Perekaman



Gambar 3.21 *Sequence Diagram* Speaker Recognition



Gambar 3.22 *Sequence Diagram* Pengujian Parameter



Gambar 3.23 *Sequence Diagram* Pengujian *Cross Validation*

3.7 Perancangan Sistem

Perancangan sistem merupakan penggambaran dan perencanaan sistem untuk mengimplementasikan hasil analisis yang dilakukan sebelumnya. Perancangan sistem yang dibuat terdiri dari perancangan basis data, perancangan struktur menu, perancangan antarmuka dan perancangan method.

3.7.1 Perancangan Basis Data

Struktur tabel menggambarkan detail basis data pada tabel yang berisi *field*, tipe data, dan keterangan lainnya.

1. Tabel data

Tabel data merupakan tabel yang akan digunakan untuk menyimpan *data titik pusat hasil k-means clustering*. Untuk lebih jelasnya ditampilkan pada tabel berikut:

Tabel 3.49 Data Latih

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_suara	int	10	*PRIMARY KEY, NOT NULL*
bobot0	double	100	*NOT NULL*
bobot1	double	100	*NOT NULL*
bobot2	double	100	*NOT NULL*
bobot3	double	100	*NOT NULL*
bobot4	double	100	*NOT NULL*
bobot5	double	100	*NOT NULL*
target	integer	10	*NOT NULL*

2. Tabel bobot

Tabel bobot merupakan tabel yang akan digunakan untuk menyimpan data titik bobot awal untuk pembelajaran LVQ. Untuk lebih jelasnya ditampilkan pada tabel berikut:

Tabel 3.50 Data Bobot

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_bobot	int	10	*PRIMARY KEY, NOT NULL*
bobot0	double	100	*NOT NULL*

Nama *Field*	Tipe Data	Ukuran	Keterangan
bobot1	double	100	*NOT NULL*
bobot2	double	100	*NOT NULL*
bobot3	double	100	*NOT NULL*
bobot4	double	100	*NOT NULL*
bobot5	double	100	*NOT NULL*
target	integer	10	*NOT NULL*

3. Tabel *pattern*

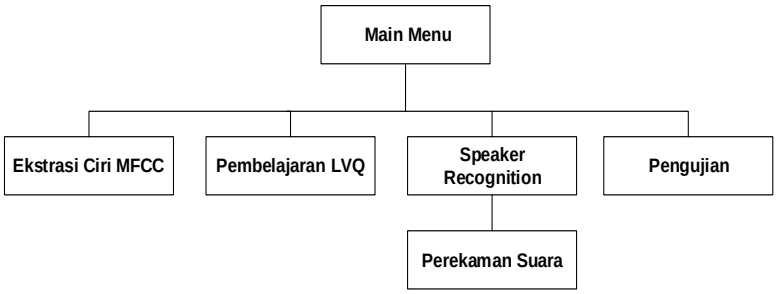
Tabel bobot merupakan tabel yang akan digunakan untuk menyimpan data titik bobot awal untuk pembelajaran LVQ. Untuk lebih jelasnya ditampilkan pada tabel berikut:

Tabel 3.51 Data *Pattern*

Nama *Field*	Tipe Data	Ukuran	Keterangan
target	int	10	*PRIMARY KEY, NOT NULL*
bobot0	double	100	*NOT NULL*
bobot1	double	100	*NOT NULL*
bobot2	double	100	*NOT NULL*
bobot3	double	100	*NOT NULL*
bobot4	double	100	*NOT NULL*
bobot5	double	100	*NOT NULL*

3.7.2 Perancangan Struktur Menu

Struktur menu merepresentasikan organisasi atau modul secara hirarki. Adapun struktur menu untuk perangkat lunak ini sebagai berikut :



Gambar 3.24 Perancangan Struktur Menu

3.7.3 Perancangan Antarmuka

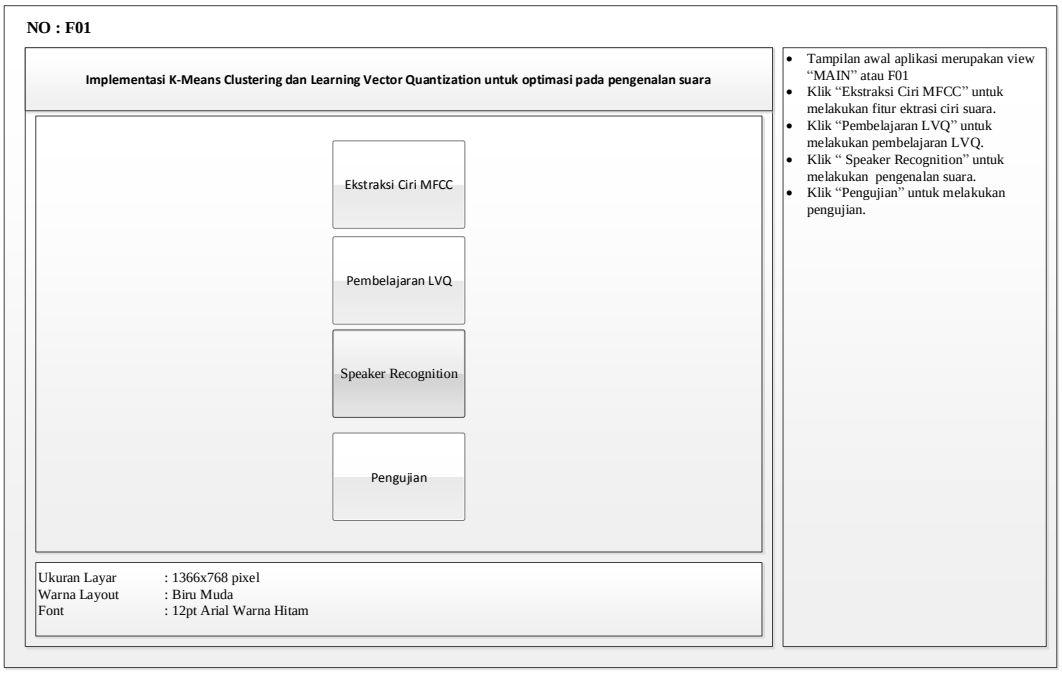
Perancangan antarmuka mendeskripsikan rencana tampilan yang akan digunakan pada aplikasi yang akan dibangun. Perancangan antarmuka yang ada terdiri dari perancangan form, perancangan pesan dan jaringan semantik.

3.7.3.1 Perancangan Form

Perancangan antarmuka *form* merupakan rancangan dalam menentukan tata letak dari tampilan sistem yang akan dibuat. Rancangan tersebut dibuat sebagai perancangan *form* pada sistem. Berikut adalah perancangan *form :*

1. F01 – *Form Main Menu*

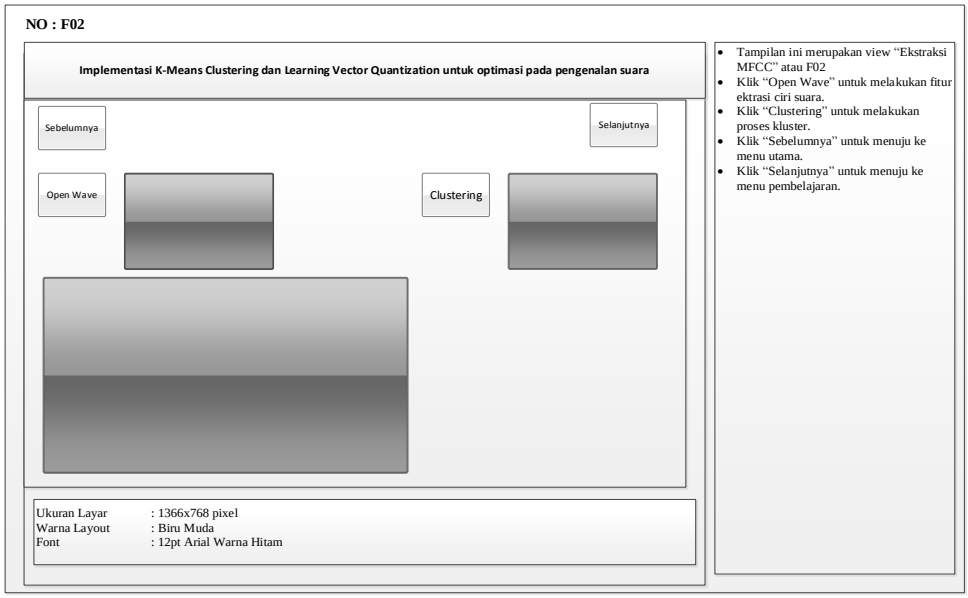
Perancangan F01 merupakan halaman utama pada saat pertama kali menjalankan program.. Perancangan F01 dapat dilihat pada gambar berikut:



Gambar 3.25 F01 – *Form Main Menu*

2. F02 – *Form* MFCC

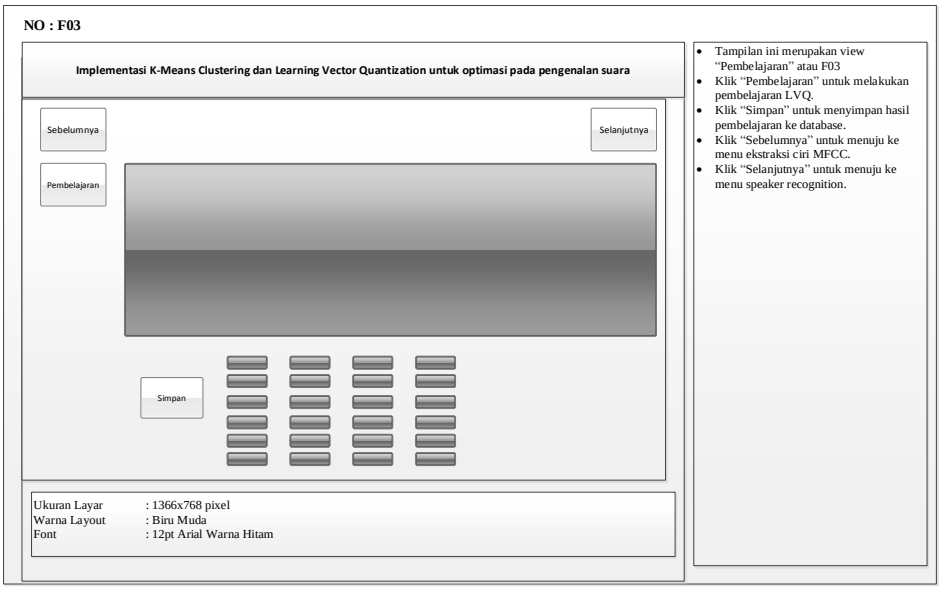
Perancangan F02 merupakan halaman untuk melakukan proses ektraksi ciri. Perancangan F02 dapat dilihat pada gambar berikut:



Gambar 3.26 F02 – *Form* MFCC

3. F03 – *Form* Pembelajaran LVQ

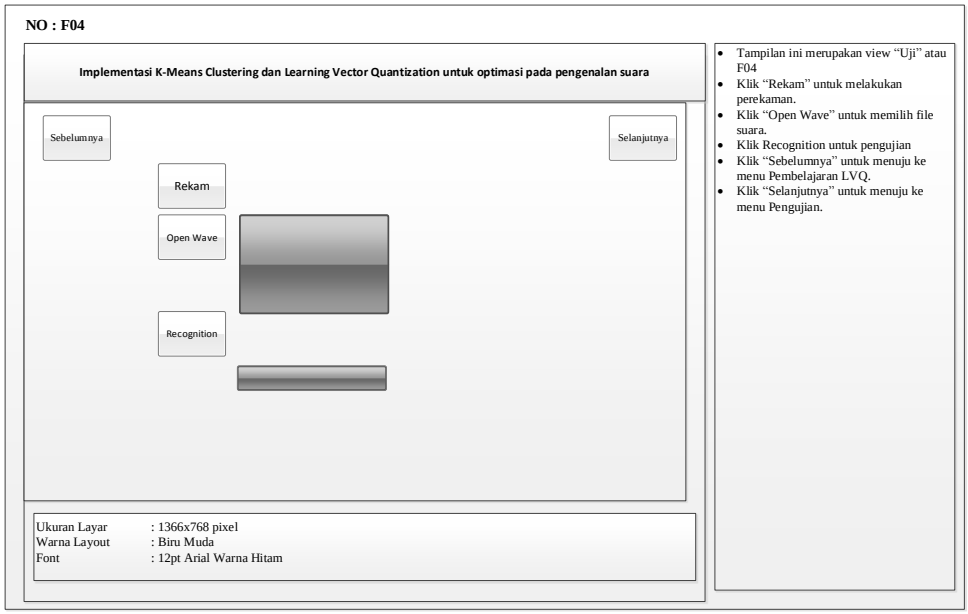
Perancangan F03 merupakan halaman untuk melakukan proses pelatihan/pembelajaran. Perancangan F03 dapat dilihat pada gambar berikut:



Gambar 3.27 F03 – *Form* Pembelajaran LVQ

4. F04 – *Form Speaker Recognition*

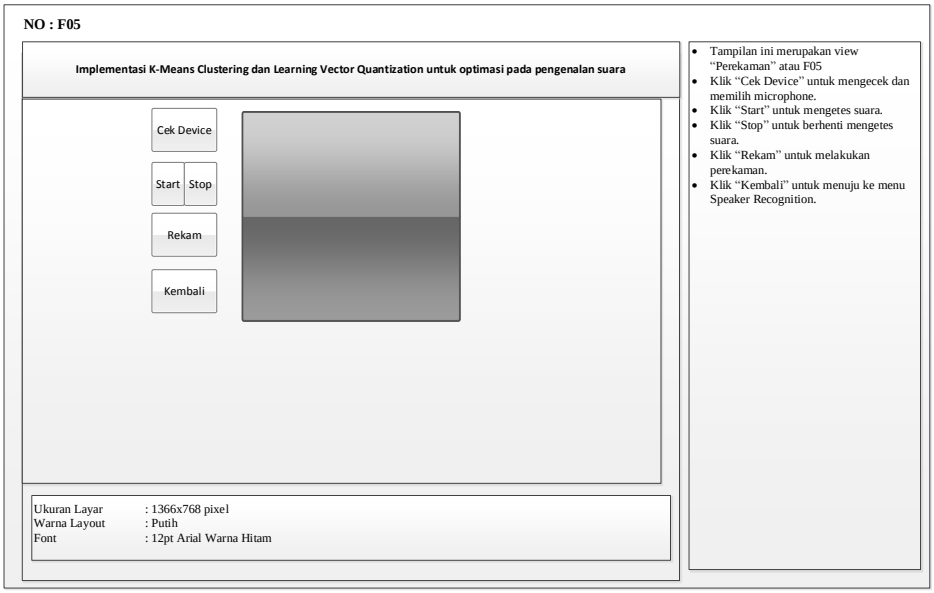
Perancangan F04 merupakan halaman untuk melakukan proses pencocokan suara. Perancangan F04 dapat dilihat pada gambar berikut:



Gambar 3.28 F04 – *Form Speaker Recognition*

5. F05 – *Form* Perekaman

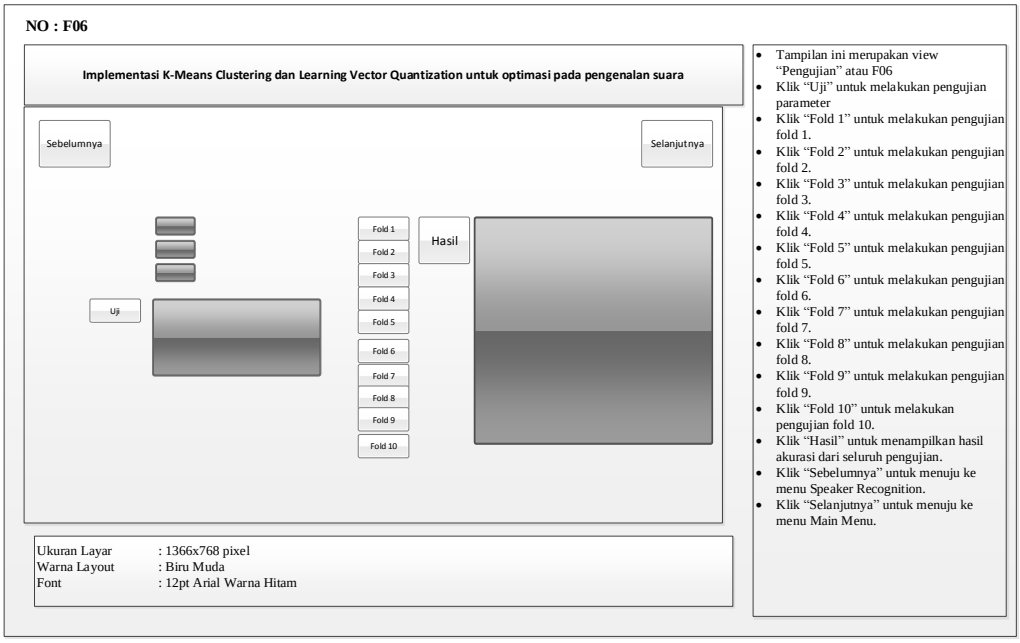
Perancangan F05 merupakan halaman untuk melakukan proses perekaman. Perancangan F05 dapat dilihat pada gambar berikut:



Gambar 3.29 F05 – *Form* Perekaman

6. F06 – *Form* Pengujian

Perancangan F06 merupakan halaman untuk melakukan proses pengujian. Perancangan F06 dapat dilihat pada gambar berikut:



Gambar 3.30 F06 – *Form* Pengujian

3.7.3.2 Perancangan Pesan

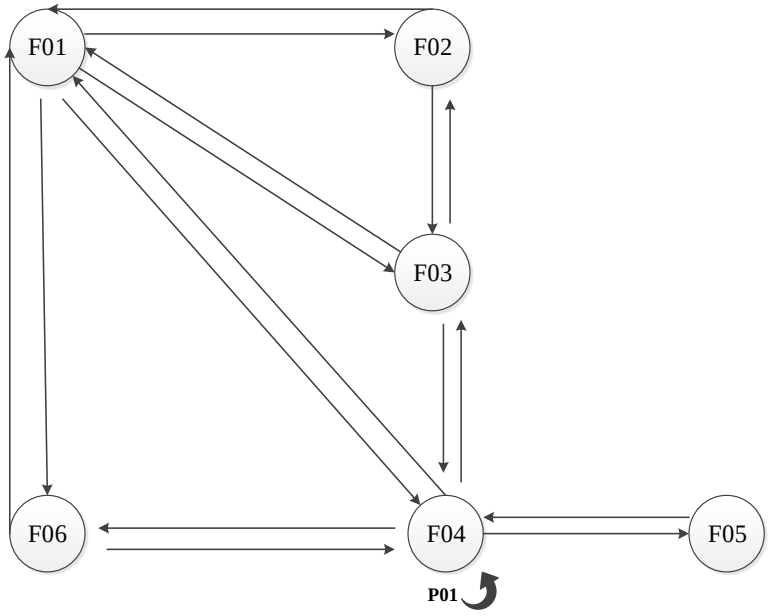
Perancangan pesan merupakan validasi dan pesan yang akan disampaikan oleh dalam keadaan-keadaan tertentu. Adapun perancangan pesan dapat dilihat pada gambar berikut:



Gambar 3.31 Perancangan Pesan

3.7.3.3 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu antarmuka ke antarmuka lain. Jaringan semantik yang terbentuk pada sistem ini dapat dilihat pada berikut:



Gambar 3.32 Jaringan Semantik