

SYSTEMIC: Information System and Informatics Journal**ISSN: 2460-8092, 2548-6551 (e)****Vol 6 No 1 – Agustus 2020****Sistem Pembelajaran Hukum Baca Al-Qur'an Menggunakan Algoritma LPC dan KNN****Hafizh Achmad Dinan¹, Youllia Indrawaty N², Kurnia Ramadhan Putra³**^{1,2,3}) Institut Teknologi Nasionalhafizhdinan35@gmail.com¹, youllia1975@gmail.com², kurniaramadhanp91@gmail.com³**Kata Kunci***LPC, KNN, ekstraksi suara, python, tajwid.***Abstrak**

Seorang muslim harus dapat membaca ayat-ayat Al-Quran dengan baik sesuai yang diajarkan oleh Rasulullah saw. Membaca Al-Quran sesuai ilmu tajwid hukumnya wajib bagi setiap orang muslim, apabila seseorang membaca Al-Quran dengan tidak memakai tajwid, hukumnya berdosa. Pengembangan Aplikasi Pembelajaran Tajwid Al-Qur'an ini bertujuan untuk membantu seorang muslim agar pandai membaca al-Qur'an yang baik dan benar. Surat yang di pakai dalam aplikasi ini adalah surat Al-Fatihah. Aplikasi Pembelajaran Tajwid Al-Qur'an menggunakan Bahasa pemrograman Python dengan metode Linear Predictive Coding (LPC) sebagai ekstraksi ciri suara dan metode K-Nearest Neighbor sebagai pencocokan dengan data latih. Untuk pengujian pengucapan ayat ke-1 didapatkan akurasi data sebesar 83.3%, ayat ke-2 sebesar 86.7%, ayat ke-3 sebesar 85%, ayat ke-4 sebesar 80%, ayat ke-5 sebesar 88.3%, ayat ke-6 sebesar 93.3%

Keywords*LPC, KNN, python, , tajwid voice extraction.***Abstract**

A Muslim must be able to read the verses of the Qur'an properly as taught by the Prophet, Muhammad. Reading the Qur'an in accordance with tajwid is obligatory for every Muslim, if someone reads the Qur'an without using tajwid, the law is sinful. The development of the application of learning the Tajwid of Qur'an is aimed at helping a Muslim to be good at reading the Qur'an that is good and right. Al-Fatihah is uses in this application. Learning the Tajwid of Qur'an Application is using Linear Predictive Coding (LPC) method as sound feature extraction and K-Nearest Neighbor as matching with training data. For testing the pronunciation of the 1st verse obtained data accuracy of 83.3%, the 2nd verse is 86.7%, the 3rd verse is 85%, the 4th verse is 80%, the 5th verse is 88.3%, the 6th verse is 93.3%.

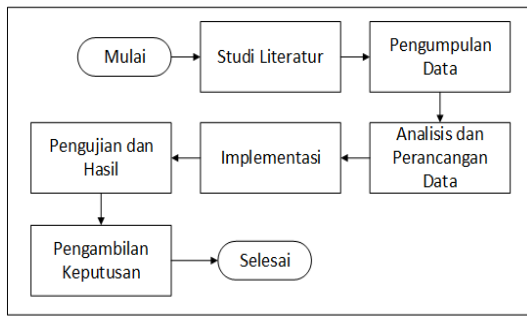
1. Pendahuluan

Seorang muslim harus dapat membaca ayat-ayat Al-Quran dengan baik sesuai yang diajarkan oleh Rasulullah saw. Membaca Al-Quran sesuai ilmu tajwid hukumnya wajib bagi setiap orang muslim, tidak bisa diwakili oleh orang lain. Apabila seseorang membaca Al-Quran dengan tidak memakai tajwid, hukumnya berdosa karena sesungguhnya Allah SWT menurunkan Al-Quran berikut tajwidnya[1]. Salah membaca hukum bacaan Al-Qur'an dapat menimbulkan arti yang berbeda, meskipun itu hanya panjang pendeknya suara. Surat Al-Fatihah merupakan surat pertama dari kitab suci Al-Qur'an, dan selalu diucapkan saat sholat.

Berdasarkan permasalahan tersebut, yaitu saat membaca Al-Qur'an orang masih sering salah dalam membaca tajwid Al-Qur'an. Oleh karena itu, penelitian ini fokus kepada merancang aplikasi mendeteksi suara baca Al-Qur'an yang dibangun dengan Algoritma LPC untuk ekstraksi ciri suara dan Algoritma KNN untuk pencocokan dengan database.

2. Metode Penelitian

Beberapa tahapan penelitian yang dilakukan meliputi studi literatur dan pengumpulan data, analisis dan perancangan sistem, implementasi sistem, pengujian dan hasil, serta kesimpulan. Alur dari tahapan penelitian tersebut dapat dilihat pada Gambar 1.



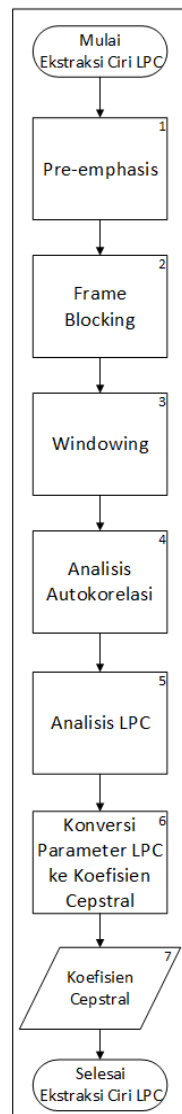
Gambar 1. Alur Metode Penelitian

2.1 Pengumpulan Data

Data penelitian yang digunakan merupakan hasil rekaman dan dari situs yang beralamat <https://islamdownload.net/> dalam bentuk file .wav.

2.2 Linear Predictive Coding (LPC)

Terdapat beberapa penelitian yang menggunakan metode LPC karena menurut Rabiner [2] LPC memiliki kelebihan yaitu sederhana dan dapat diterapkan pada perangkat lunak dan keras.



Gambar 2. Langkah-langkah LPC

a. Pre-emphasis

Tahap pertama adalah menghilangkan noise suara yang tidak dibutuhkan. Noise ini berupa sinyal yang memiliki perbedaan signifikan di antara nilai di sekitarnya. Pada tahapan ini diterapkan Persamaan (1) (Deng, O'Shaughnessy, 2003):

$$y(n) = s(n) - \alpha \cdot s(n-1) \quad (1)$$

$\tilde{s}(n)$: sinyal suara setelah dilakukan proses preemphasize.

$y(n)$: sinyal suara sebelum dilakukan proses preemphasize.

α : nilai yang paling sering digunakan yaitu 0.95

b. Frame Blocking

Frame Blocking merupakan proses membagi sinyal suara menjadi beberapa blok suara.

c. Windowing

Pada tahapan frame blocking ada kemungkinan terdapat sinyal yang terputus-putus atau mengalami diskontinuitas antara blok satu dengan yang lainnya, sehingga pada tahapan windowing dilakukan untuk mengurangi efek tersebut. Dengan menerapkan Persamaan (3) [2] dengan asumsi jumlah sample masing-masing frame atau N adalah 640, sehingga diperoleh $w(n)$ sebagai berikut:

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (2)$$

$$\tilde{x}(n) = x(n) \cdot w(n) \quad (3)$$

$W(n)$ adalah nilai *hamming*

$\tilde{x}(n)$ adalah nilai *windowing*

$x(n)$ adalah nilai *frame blocking*

d. Analisis Autokorelasi

Bagian sinyal yang telah melalui tahap windowing, dibentuk autokorelasinya dengan Persamaan (4) berikut:

$$r(m) = \sum_{n=0}^{N-1-m} \tilde{x}(n) \cdot \tilde{x}(n+m) \quad (4)$$

$r(m)$ adalah nilai analisis autokorelasi

$\tilde{x}(n)$ adalah nilai *windowing*

e. Analisis LPC

Proses ini dilakukan untuk mengubah nilai autokorelasi menjadi parameter koefisien transformasi. Pada tahap ini akan menghasilkan sinyal suara melalui perhitungan berikut:

$$E^{(0)} = r(0) \quad (5)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (6)$$

$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} r(i-j)}{E^{(i-1)}} \quad (7)$$

$$\alpha_i^{(i)} = k_i \quad (8)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad (9)$$

E adalah nilai error

K adalah koefisien PARCOR

α adalah koefisien LPC

f. Konversi Parameter LPC ke Koefisien Cepstral
Koefisien cepstral ini adalah koefisien dari representasi transformasi Fourier pada spectrum logaritmis

$$c_m = \alpha_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) \cdot c_k \cdot \alpha_{m-k} \quad (10)$$

c adalah koefisien Cepstral

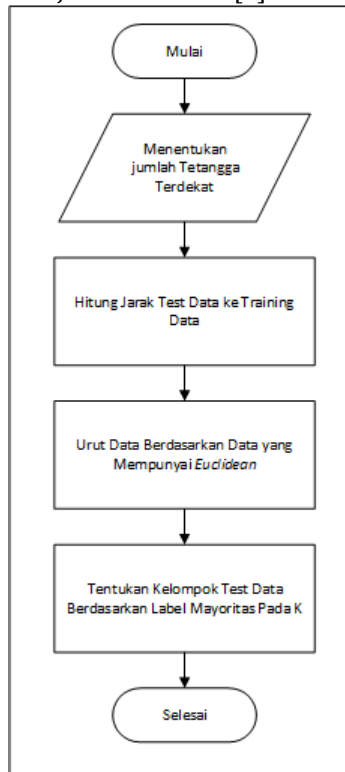
a adalah koefisien LPC

m adalah orde ke- n

2.3 K-Nearest Neighbor

Dalam pengenalan pola, algoritma tetangga-k terdekat (k-NN) adalah metode non-parametrik yang digunakan untuk klasifikasi dan regresi.[3]

Algoritma K-Nearest Neighbor mengelompokkan data baru yang belum diketahui class-nya dengan memilih data sejumlah k yang letaknya paling dekat dari data baru. Class paling banyak dari data terdekat sejumlah k akan dipilih sebagai class yang diprediksi untuk data baru [4]. Pada umumnya nilai k adalah jumlah ganjil supaya tidak terdapat jarak yang sama dalam proses klasifikasi. Jauh atau dekatnya tetangga dihitung menggunakan jarak Euclidean[5].



Gambar 3. Flowchart Metode KNN

Pada Gambar 3 dapat dilihat bahwa tahapan metode KNN melalui beberapa tahapan berikut:

1. Tahap awal adalah menentukan tetangga terdekat (k).
2. Selanjutnya menghitung dengan rumus euclidean distance antara sample latih dan sample uji dengan Persamaan (10).

$$D(a, b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \quad (11)$$

$D(a, b)$: Nilai Euclidean Distance

a : Data latih (training)

b : Data uji (testing)

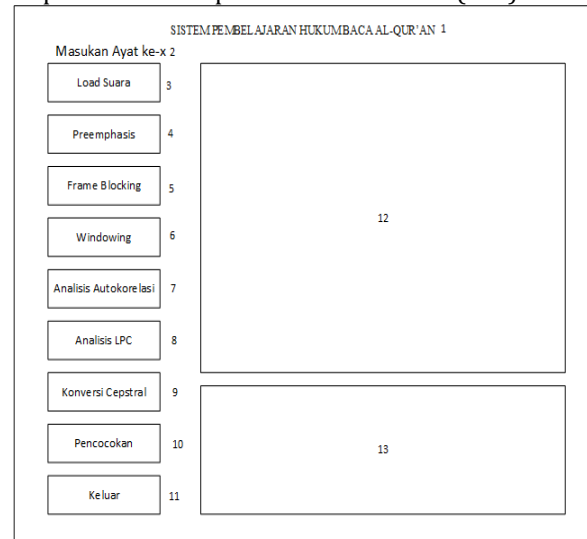
k : k (fitur) ke-

d : Dimensi (jumlah k)

3. Setelah itu mengurutkan data yang mempunyai nilai k .
4. Tahap akhir mengelompokkan sample latih berdasarkan mayoritas nilai k .

2.4 Implementasi

Implementasi pada penelitian ini terdiri dari implementasi Graphical User Interface (GUI).



Gambar 4. User Interface

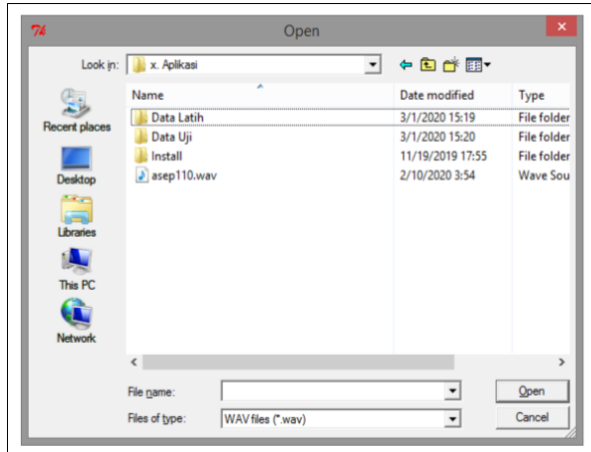
Keterangan dari Gambar 4 sebagai berikut:

1. Judul Aplikasi
2. Indikator Ayat ke-x
3. Tombol load suara
4. Tombol preemphasis
5. Tombol Frame Blocking
6. Tombol Windowing
7. Tombol Analisis Autokorelasi
8. Tombol Analisis LPC
9. Tombol Konversi Cepstral
10. Tombol Pencocokan
11. Tombol Keluar
12. Grafik
13. Textbox

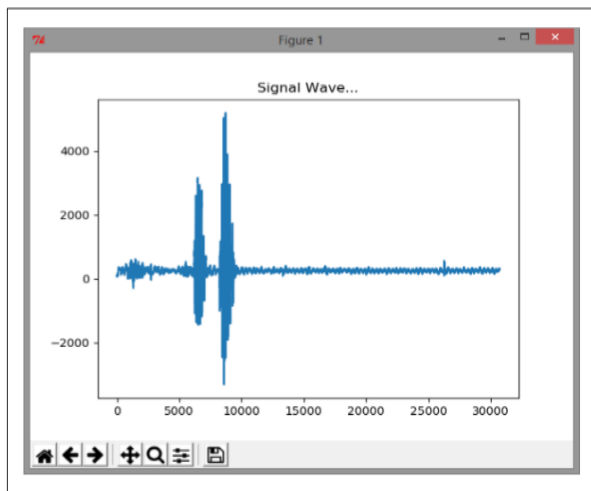
3. Hasil Dan Pembahasan

Sebelum dilakukan proses pencocokkan, data uji dan data latih diekstraksi dahulu menggunakan Linear Prediction Coding untuk mendapatkan nilai koefisien cepstral suara masukan. Pada penelitian penentuan kualitas suara ini terdapat tujuh pengucapan ayat surat Al-Fatihah. Dimulai dengan proses pengambilan data dengan cara ambil data suara yang sudah ada. Pengguna merekam atau mengambil data suara yang berformat .wav dengan tombol Browse suara yang berformat .wav.

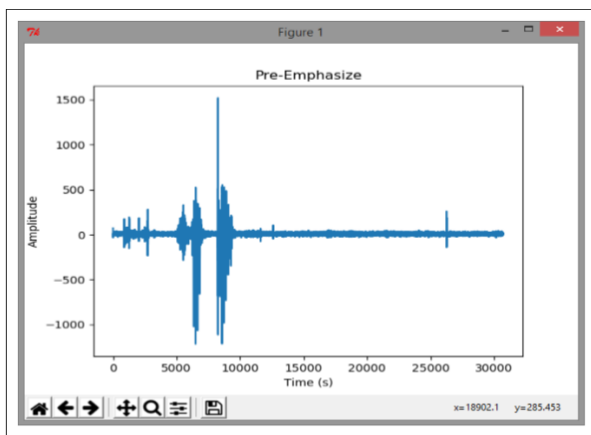
Gambar 5 menunjukkan pengguna mengambil data suara berformat .wav yang sudah ada sebelumnya dan suara yang diambil akan dijadikan data suara uji untuk penelitian ini. Gambar 6 merupakan tampilan sinyal dari data suara yang dipilih ditampilkan pada layar aplikasi. Dan Gambar 7 merupakan hasil dari data sinyal suara yang sudah dilakukan proses *Preemphasize*.



Gambar 5. Browse Suara



Gambar 6. Hasil Sinyal Suara



Gambar 7. Hasil Preemphasis

Data sinyal suara yang telah diambil akan dilakukan proses pertama yaitu *Preemphasize* yang bertujuan untuk mendapatkan data sinyal suara

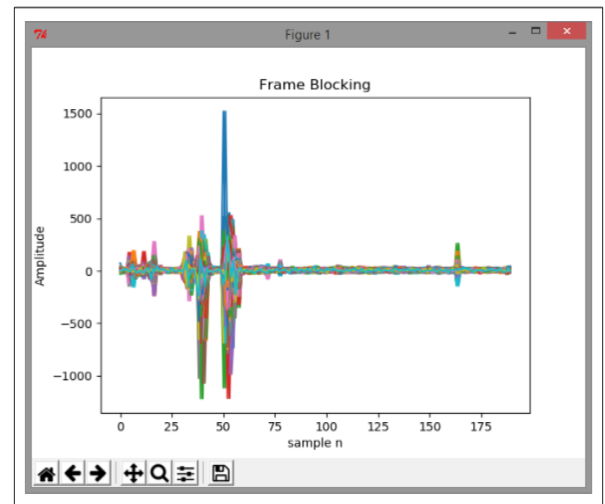
tanpa noise.

Misalkan nilai adalah $(-20.125, -36.125, -25.325, -4.425, 19.175, 25.275, 42.875, 27.875)$, sampel sinyal : $(-20.125, -36.125, -25.325, -4.425, 19.175, 25.275, 42.875, 27.875)$, dengan $\alpha = 0.95$ dengan menerapkan Persamaan (1), maka diperoleh

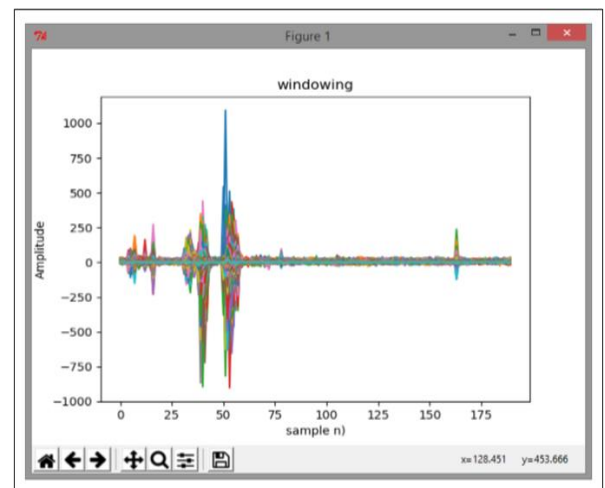
$$\begin{aligned} y(0) &= s(0) = -20.125 \\ y(1) &= s(1) - \alpha * y(1 - 1) \\ &= -36.125 - (-20.125 * 0.95) \\ &= -17.006 \end{aligned}$$

Begitu juga dengan ketujuh data yang lainnya, sehingga didapatkan nilai sinyal setelah preemphasis adalah $(-20.125, -17.006, 8.994, 19.634, 23.379, 7.059, 18.864, -12.856)$.

Gambar 8 merupakan hasil dari data sinyal suara yang sudah dilakukan proses *Frame Blocking*. Hasil data sinyal suara yang sudah diproses pada tahapan *Preemphasize* sinyal akan disegmentasi menjadi beberapa frame ditahapan *Frame Blocking* ini.



Gambar 8. Hasil Frame Blocking



Gambar 9. Hasil Windowing

Gambar 9 merupakan hasil dari data sinyal suara yang sudah dilakukan proses *Windowing*. Hasil data sinyal suara yang sudah disegmentasi pada tahapan *Frame Blocking* akan menimbulkan efek diskontinuitas di awal dan di akhir frame. Saat

dilakukan proses windowing, data sinyal suara yang memiliki diskontinuitas di awal dan di akhir diubah nilainya menjadi nol.

Diketahui nilai sinyal setelah Frame Blocking adalah : (1.824 , 1.1771, 1.1664, 1.2551, 0.8692, 0.0165, -0.7363, -0.0334) dengan nilai N = 320 dari perhitungan N pada bagian frame blocking sebelumnya. Pertama mencari nilai *hamming* dengan persamaan (2):

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

$$w(0) = 0.54 - 0.46 \cos\left(\frac{2 * 3.14 * 0}{320 - 1}\right)$$

$$w(0) = 0.54 - 0.46 \cos(0)$$

$$w(0) = 0.54 - 0.46 * 1$$

$$w(0) = 0.54 - 0.46 = 0.08$$

Kemudian mencari nilai windowing dengan persamaan (3):

$$\tilde{x}(0) = x(0) \cdot w(0)$$

$$\tilde{x}(0) = 1.824 * 0.08 = 0.14592$$

Begitu juga dengan ketujuh data yang lainnya, sehingga didapatkan nilai sinyal setelah *windowing* yaitu (0.14592 , 0.094168, 0.093312 , 0.100408, 0.069536, 0.00132, -0.058936 , -0.002672)

Selanjutnya tahap Analisis Autokorelasi yaitu korelasi silang terhadap sinyal dengan dirinya sendiri.

Jika data hasil windowing sampel data latih adalah (0.192 , 0.168, 0.312, 0.408, 0.536 , 0.32, -0.936 , -0.672) maka nilai autokorelasinya dengan menggunakan persamaan (4) adalah:

$$r(0) = \tilde{x}(0) * \tilde{x}(0) + \tilde{x}(1) * \tilde{x}(1) + \tilde{x}(2) * \tilde{x}(2) + \tilde{x}(3) * \tilde{x}(3) + \tilde{x}(4) * \tilde{x}(4) + \tilde{x}(5) * \tilde{x}(5) + \tilde{x}(6) * \tilde{x}(6) + \tilde{x}(7) * \tilde{x}(7)$$

$$r(0) = (0.192 * 0.192) + (0.168 * 0.168) + (0.312 * 0.312) + (0.408 * 0.408) + (0.536 * 0.536) + (0.32 * 0.32) + (-0.936 * (-0.936)) + (-0.672 * (-0.672))$$

$$r(0) = 2.046272$$

Begitu juga dengan ke tujuh data lainnya sehingga didapatkan hasil analisis autokorelasi menggunakan order 8 adalah (2.046, 0.932, -0.290, -0.474, -0.410, -0.305, -0.293, -0.129).

Selanjutnya tahap Analisis LPC yaitu untuk mengubah nilai autokorelasi menjadi parameter koefisien transformasi dengan menggunakan persamaan (5), (6), (7), (8), (9):

$$E^{(0)} = 2.046$$

$$k_1 = \frac{r(1) - \sum_{j=1}^{1-1} \alpha_j^{(1-1)} r(|1-j|)}{E^{(1-1)}}$$

$$k_1 = \frac{r(1)}{E^{(0)}} = \frac{0.932}{2.046} = 0.45552297165 = 0.456$$

$$\alpha_1^{(1)} = k_1$$

$$\alpha_1^{(1)} = 0.456$$

$$E^{(1)} = (1 - k_1^2) E^{(1-1)}$$

$$E^{(1)} = (1 - (0.456)^2) 2.046 = 1.620562944 = -1.621$$

Begitu juga dengan ke tujuh data lainnya sehingga didapatkan hasil analisis LPC adalah (1.654, 1.415, 0.522, -0.578, -1.146, -1.641, 1.057)

Tahap terakhir dalam metode LPC adalah Konversi koefisien LPC menjadi koefisien Cepstral dengan menggunakan persamaan (10):

$$c_0 = \ln(p) = \ln(7) = 1.94591014906 = 1.946$$

$$c_1 = \alpha_1 + \sum_{k=1}^{1-1} \left(\frac{k}{1}\right) \cdot c_k \cdot a_{1-k} = \alpha_1 = 1.645$$

$$c_2 = \alpha_2 + \sum_{k=1}^{2-1} \left(\frac{k}{2}\right) \cdot c_k \cdot a_{2-k} = \alpha_2 + \left(\frac{1}{2} * c_1 * \alpha_1\right)$$

Begitu juga dengan data lainnya sehingga didapatkan hasil koefisien Cepstral adalah (1.94591, -0.01799, -0.12958, 0.10819, -0.19576, -0.41561, 0.06887, -0.11295). Kemudian, setelah tahapan mendapatkan nilai cepstral, dilakukan pencocokan dengan metode KNN.

Tabel 1 merupakan nilai cepstral dari 6 file suara wav terdiri dari 3 file suara ayat 1 dan 3 file suara ayat 2. Nilai koefisien cepstral yang akan di ui adalah C0 = 1.946 , C1= -0.304, C2= -0.058, C3= 0.109, C4= -0.375, C5=0.105, C6=0.261, C7=0.067. Perhitungan KNN ada dalam Tabel 2 menggunakan persamaan (11). Setelah dihitung *square distance* nya, Selanjutnya mengurutkan jarak tersebut dan tentukan tetangga mana yang terdekat berdasarkan jarak minimum ke -K sebagaimana Tabel 3. Selanjutnya menentukan kategori dari tetangga terdekat. Kategori dari tetangga terdekat (Y) tidak termasuk karena rangking dari data ini lebih dari 3 (=K). terdapat kategori dari yaitu satu dan dua karena di database hanya terdapat klasifikasi satu dan dua sebagaimana Tabel 4.

Tabel 1. Cepstral dan Klasifikasi

C0 Data Latih	C1 Data Latih	C2 Data Latih	C3 Data Latih	C4 Data Latih	C5 Data Latih	C6 Data Latih	C7 Data Latih	Klasifikasi Data Latih (Ayat)
1.946	-0.018	-0.13	0.108	-0.196	-0.416	0.069	-0.113	1
1.946	0.099	0.182	-0.619	-0.474	0.009	0.061	0.183	1
1.946	-0.131	0.043	0.244	-0.549	0.11	-0.091	0.11	1
1.946	0.041	0.512	0.235	0.679	-0.194	-0.18	0.211	2
1.946	-0.047	0.724	-0.8	-0.113	0.01	-0.04	0.016	2
1.946	0.063	-0.203	-0.281	-0.541	0.59	0.085	0.251	2

Tabel 2. Square Distance

Square Distance
$(1.946 - 1.946)^2 + (-0.018 - (-0.304))^2 + (-0.13 - (-0.058))^2 + (0.108 - 0.109)^2 + (-0.196 - (-0.375))^2 + (-0.416 - 0.105)^2 + (0.069 - 0.261)^2 + (-0.113 - 0.067)^2 = 0.459727$
$(1.946 - 1.946)^2 + (0.099 - (-0.304))^2 + (0.182 - (-0.058))^2 + (-0.619 - 0.109)^2 + (-0.474 - (-0.375))^2 + (0.009 - 0.105)^2 + (0.061 - 0.261)^2 + (0.183 - 0.067)^2 = 0.822466$
$(1.946 - 1.946)^2 + (-0.131 - (-0.304))^2 + (0.043 - (-0.058))^2 + (0.244 - 0.109)^2 + (-0.549 - (-0.375))^2 + (0.11 - 0.105)^2 + (-0.091 - 0.261)^2 + (0.11 - 0.067)^2 = 0.214409$
$(1.946 - 1.946)^2 + (0.041 - (-0.304))^2 + (0.152 - (-0.058))^2 + (0.235 - 0.109)^2 + (0.679 - (-0.375))^2 + (-0.194 - 0.105)^2 + (-0.18 - 0.261)^2 + (0.211 - 0.067)^2 = 1.594535$
$(1.946 - 1.946)^2 + (-0.047 - (-0.304))^2 + (0.724 - (-0.058))^2 + (-0.8 - 0.109)^2 + (-0.113 - (-0.375))^2 + (0.1 - 0.105)^2 + (-0.04 - 0.261)^2 + (0.016 - 0.067)^2 = 1.665725$
$(1.946 - 1.946)^2 + (0.063 - (-0.304))^2 + (-0.203 - (-0.058))^2 + (-0.281 - 0.109)^2 + (-0.541 - (-0.375))^2 + (0.59 - 0.105)^2 + (0.085 - 0.261)^2 + (0.251 - 0.067)^2 = 0.635427$

Tabel 3. Mengurutkan data berdasarkan jarak minimum-K

Square Distance	Urutan Jarak	Apakah termasuk kategori k=3 ? (rangking 1,2 dan 3)
0.459727	2	Ya
0.822466	4	Tidak
0.214409	1	Ya
1.594535	5	Tidak
1.665725	6	Tidak
0.635427	2	Ya

Tabel 4. Menentukan Kategori

Square Distance	Urutan Jarak	Apakah termasuk kategori k=3 ? (rangking 1,2 dan 3)	Y = Kategori dari KNN (Class yang terdapat pada Data Latih)
0.459727	2	Ya	Ayat Satu
0.822466	4	Tidak	-
0.214409	1	Ya	Ayat Satu
1.594535	5	Tidak	-
1.665725	6	Tidak	-
0.635427	2	Ya	Ayat Dua

Selanjutnya digunakan kategori mayoritas yang sederhana dari tetangga yang terdekat tersebut sebagai nilai prediksi dari data yang baru. Dari data tersebut di dapatkan dua kategori angka 1 dan satu kategori angka 2, karena kategori angka 1 lebih banyak maka data koefisien cepstral uji yang di masukkan adalah termasuk dalam kategori satu atau ayat 1.

Hasil dari pencocokan KNN seperti pada Gambar 10 jika benar dan jika pencocokan knn salah maka seperti pada Gambar 13. jika ayatnya benar tetapi bacaan madnya terlalu panjang atau pendek akan muncul pesan seperti Gambar 11 dan Gambar 12 dan menambah total salah.

```
Prediksi =['satu']
Benar, Suara ini Ayat ke-1
```

Gambar 10. Pencocokan KNN Benar

```
Prediksi =['satu']
Benar, Suara ini Ayat ke-1, tetapi bacaannya terlalu pendek
```

Gambar 11. Pencocokan Benar Bacaan Pendek

```
Prediksi =['satu']
Benar, Suara ini Ayat ke-1, tetapi bacaannya terlalu panjang
```

Gambar 12. Pencocokan Benar Bacaan Panjang

```
Prediksi =['satu']
Salah, Suara ini bukan Ayat ke-2
```

Gambar 13. Pencocokan KNN Salah

Jika user melakukan kesalahan, maka counter total salah akan bertambah 1 seperti pada Gambar 14.

```
Total Salah = 1 x
```


Gambar 14. Total Salah

Setelah semua ayat telah di uji oleh aplikasi, akan muncul berapa kali *user* melakukan kesalahan pada ayat ke-n tersebut. Hasil akhir program seperti pada Gambar 15.

Ayat 1 Salah = 0 x
Ayat 2 Salah = 2 x
Ayat 3 Salah = 0 x
Ayat 4 Salah = 1 x
Ayat 5 Salah = 0 x
Ayat 6 Salah = 4 x

Gambar 15. Hasil Akhir Program

Hasil perhitungan ekstraksi ciri LPC pada 5 orang sampel berada pada Tabel 5. Sedangkan Tabel 6 adalah contoh pengujian KNN pada orang ke-1.

Tabel 5. Nilai Koefisien Cepstral

Orang ke-	Ayat ke-					
	1	2	3	4	5	6
1	1.94591	1.94591	1.94591	1.94591	1.94591	1.94591
	-0.01799	0.04074	0.13428	0.00618	0.0409	0.00404
	-0.12958	0.51162	1.17207	0.36951	0.11517	-0.20692
	0.10819	0.2346	0.31266	0.1153	0.04061	-0.42255
	-0.19576	0.67899	-0.20616	-0.15279	0.43143	0.42252
	-0.41561	-0.19398	0.04201	0.08586	0.06945	-0.09284
	0.06887	-0.17948	-0.22339	-0.12729	0.052	0.04057
	-0.11295	0.21143	-0.04667	0.10328	-0.0798	0.09415
2	1.94591	1.94591	1.94591	1.94591	1.94591	1.94591
	0.0999	-0.04718	0.13613	0.07253	-0.04409	0.0215
	0.1822	0.72355	1.44149	1.1112	-0.03485	0.56928
	-0.6192	-0.79969	-0.78311	-0.12762	0.49943	-0.04506
	-0.47412	-0.1129	0.29446	0.18066	-0.34606	-0.28338
	0.00861	0.00967	-0.19702	-0.22659	0.06432	-0.20474
	0.06136	-0.03951	0.21343	0.02258	0.13542	-0.10947
	0.18327	0.01588	0.13034	0.13908	0.02815	-0.08204
3	1.94591	1.94591	1.94591	1.94591	1.94591	1.94591
	-0.13134	0.06323	0.22712	0.08494	0.07202	0.0526
	0.0426	-0.20343	-0.4161	0.41539	0.76058	-0.67125
	0.24386	-0.28123	-0.35287	0.22398	-0.12786	-0.27071
	-0.54857	-0.54047	0.09688	-0.08801	0.18567	-0.09328
	0.11005	0.05895	-0.15741	0.23639	-0.44394	0.05121
	-0.09054	0.08528	0.35055	0.0903	-0.19414	0.12524
	0.11042	0.25059	-0.16364	-0.05854	0.14731	0.04725
4	1.94591	1.94591	1.94591	1.94591	1.94591	1.94591
	-0.14684	0.14846	0.24895	0.14806	0.15863	-0.12284
	0.26266	-0.25471	0.08419	1.54646	1.53458	-0.17955
	0.1467	-0.18829	-0.03937	-0.80319	-0.79396	-0.24963
	0.11007	-0.06999	0.18201	0.36836	0.35265	-0.21152
	-0.18807	0.0119	0.02719	-0.25703	-0.24802	-0.11265
	-0.09203	0.53442	0.0655	0.2561	0.24993	0.415
	0.51436	-0.1714	0.02945	0.13789	0.14275	-0.04152
5	1.94591	1.94591	1.94591	1.94591	1.94591	1.94591
	0.16758	0.17228	-0.2593	0.16496	0.16839	0.16938
	1.51104	1.39253	0.89231	0.20365	0.39622	1.47274
	-0.77235	-0.7871	0.13424	-0.00554	0.04305	-0.75962
	0.30698	0.13404	-0.25098	-0.30007	-0.25721	0.266
	-0.22149	-0.23708	-0.07297	-0.11607	0.35844	-0.20197
	0.22203	0.11949	0.038	0.07304	-0.18693	0.19826
	0.16163	0.14803	-0.01332	-0.22912	0.06786	0.16128

Tabel 6. Pencocokan KNN pada orang ke-1

Ayat	C0	C1	C2	C3	C4	C5	C6	C7	
1	1.946	-0.018	-0.13	0.108	-0.196	-0.416	0.069	-0.113	Berhasil
2	1.946	0.041	0.512	0.235	0.679	-0.194	-0.18	0.211	Berhasil
3	1.946	0.134	1.172	0.313	-0.206	0.042	-0.223	-0.047	Gagal
4	1.946	0.006	0.37	0.115	-0.153	0.086	-0.127	0.103	Gagal

5	1.946	0.041	0.115	0.041	0.431	0.069	0.052	-0.08	Berhasil
6	1.946	0.004	-0.207	-0.423	0.423	-0.093	0.041	0.094	Berhasil

Persamaan (12), (13), dan (14) adalah persamaan untuk menghitung *precision*, *recall*, dan *accuracy* pada setiap ayat.

$$precision = \frac{TP}{(TP+FP)} \times 100\% \quad (12)$$

$$recall = \frac{TP}{(TP+FN)} \times 100\% \quad (13)$$

$$accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% \quad (14)$$

TP : True Positive, jumlah ayat ke-n yang benar yang terdeteksi oleh aplikasi

TN : True Negative, jumlah ayat ke-n yang terdeteksi benar oleh aplikasi tetapi bukan ayat ke-n

FP : False Positive, jumlah suara ayat ke-n yang benar tetapi tidak terdeteksi oleh aplikasi

FN : False Negative, jumlah suara yang bukan ayat ke-n

Tabel 7. Data Pengujian

Ayat ke-	TP	FP	FN	TN
1	6	6	4	44
2	5	3	5	47
3	6	5	4	45
4	4	6	6	44
5	6	3	4	47
6	8	2	2	48

Tabel 3 adalah data pengujian dari 60 file suara wav dengan setiap ayat 10 file suara wav.

Berikut adalah contoh perhitungan dengan persamaan (11), (12), (13) menggunakan contoh ayat ke-1:

$$\text{Precision ayat 1} = \frac{TP}{(TP+FP)} \times 100\% = \frac{6}{6+6} \times 100\% = 50\%$$

$$\text{Recall ayat 1} = \frac{TP}{(TP+FN)} \times 100\% = \frac{6}{6+4} \times 100\% = 60\%$$

$$\text{Accuracy ayat 1} = \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% = \frac{6+44}{(6+6+4+44)} \times 100\% = 83.3\%$$

Setelah dihitung menggunakan persamaan (12), (13), dan (14) maka didapatkan hasil seperti pada Tabel 8.

Tabel 8 Akurasi Data

Ayat ke-	Precision	Recall	Accuracy
1	50%	60%	83.3%
2	62.5%	50%	86.7%
3	54.6%	60%	85%
4	40%	40%	80%
5	66.7%	60%	88.3%
6	80%	80%	93.3%

4. Kesimpulan

Berdasarkan penelitian mengenai sistem ukum baca Al-Qur'an yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

Untuk mendapatkan nilai koefisien cepstral dilakukan dengan cara pencarian metode Linear Prediction Coding dan pencocokan antara data suara Koefisien Cepstral latih dengan data suara Koefisien Cepstral uji menggunakan metode K-nearest Neighbor (KNN). Sistem yang dibangun dapat digunakan untuk mengenali suara berupa ayat.

Untuk pengujian pengucapan ayat ke-1 didapatkan akurasi data sebesar 83.3%, ayat ke-2 sebesar 86.7%, ayat ke-3 sebesar 85%, ayat ke-4 sebesar 80%, ayat ke-5 sebesar 88.3%, ayat ke-6 sebesar 93.3%.

Daftar Pustaka

- [1] Suryani, D., Irfan, M., Uriawan, W., & Budiawan, W. (2016). IMPLEMENTASI ALGORITMA DIVIDE AND CONQUER PADA APLIKASI BELAJAR ILMU TAJWID. JOIN Volume I No. 1.
- [2] Rabiner, L., & Juang, B.-H. (1933). Fundamental of Speech Recognition. New Jersey: PTR Prentice-Hall, Inc.
- [3] Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. American Statistical Association, 175-185.
- [4] Shofia, E. N., Putri, R. R., & Arwan, A. (2017). Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode K-Nearest Neighbor - Certainty Factor. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 426-435.
- [5] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining. Concept and Techniques*. Waltham: Morgan Kaufmann.
- [6] Dewi, I. A., Zulkarnain, A., & Lestari, A. A. (2018). Identifikasi Suara Tangisan Bayi menggunakan Metode LPC dan Euclidean Distance. *ELKOMIKA, Vol. 6, No. 1*, 153-164.
- [7] Fatoni, C. S., & Noviandha, F. D. (2017). Case Based Reasoning Diagnosis Penyakit Difteri dengan Algoritma K-Nearest Neighbor. *Citec Journal*, 220-232.
- [8] Hanafi, M., Fadillah, N., & Ihsan, A. (2019). Optimasi Algoritma K-Nearest Neighbor untuk Klasifikasi Tingkat Kematangan Buah Alpukat Berdasarkan Warna. *IT Journal Research and Development (ITJRD)*, 10-18.
- [9] S Harsemadi, G., Sudarma, M., & Pramaita, N.

- (2017). Implementasi Algoritma K-Nearest Neighbor pada Perangkat Lunak Pengelompokan Musik untuk Menentukan Suasana Hati. *Teknologi Elektro*, 15-19.
- [10] Idwal, A. Y., Nurhasanah, Y. I., & Utami, D. B. (2017). Sistem Pengenalan Suara Bahasa Indonesia Untuk Mengenali Aksen Melayu Pontianak Dan Sunda Garut. *Jurnal Teknik Informatika dan Sistem Informasi Volume 3 Nomor 3*, 461-471.
- [11] Indrawaty, Y., Dewi, I. A., & Lukman, R. (2019). Ekstraksi Ciri Pelafalan Huruf Hijaiyyah Dengan Metode Mel-Frequency Cepstral Coefficients. *MIND Journal*, 01-16.
- [12] Mustakim, & Oktaviani F, G. (2016). Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. *Jurnal Sains, Teknologi dan Industri*, 195-202
- [13] Nurhasanah, Y. I., Dewi, I. A., & Saputro, B. A. (2018). Iqro Reading Learning System through Speech Recognition Using Mel Frequency Cepstral Coefficient (MFCC) and Vector Quantization (VQ) Method. *Int. Journal of Applied IT Vol. 02 No. 01*, 29-42.
- [14] Rivki, M., & Bachtiar, M. (2017). Implementasi Algoritma K-Nearest Neighbor Dalam Pengklasifikasian Follower Twitter Yang Menggunakan Bahasa Indonesia. *Jurnal Sistem Informasi (Journal of Information Systems)*, 31-37.
- [15] Sudaradjat, D. (2019). Digitalisasi Sinyal Suara Manusia Dengan Algoritma Linear Predictive Coding. *Jurnal Ilmu Pengetahuan Dan Teknologi Komputer*, 177-184
- [16] A Triansyah, E., & Indrawaty N, Y. (2017). Implementasi Metode Pattern Recognition Untuk Pengenalan Ucapan Huruf Hijaiyyah. *Jurnal Ilmiah Teknologi Informasi Terapan*, 1-10.