

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Voice recognition***

Pengenalan ucapan atau pengenalan suara dalam istilah bahasa Inggrisnya, *voice recognition* adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi sebagai sebuah komando untuk melakukan suatu pekerjaan, misalnya penekanan tombol pada telepon genggam yang dilakukan secara otomatis dengan komando suara. Alat pengenalan ucapan/suara, yang sering disebut dengan *voice recognizer*, membutuhkan sampel kata sebenarnya yang diucapkan dari pengguna. Sampel kata akan didigitalisasi, disimpan ke dalam komputer, dan kemudian digunakan sebagai basis data dalam mencocokkan kata yang diucapkan selanjutnya. Sebagian besar alat pengenalan ucapan sifatnya masih tergantung kepada pembicara. Alat yang dibuat pun hanya dapat mengenali kata yang diucapkan dari satu atau dua orang saja (Rusdi & Yani, 2018).

#### **2.2 *Speech processing***

*Speech processing* (pemrosesan ucapan) adalah metode mengekstrak informasi yang diinginkan dari sebuah sinyal suara. Untuk memproses sebuah sinyal dengan sebuah komputer digital, sinyal harus dihadirkan dalam bentuk digital sehingga sinyal tersebut dapat digunakan oleh sebuah komputer digital (Rabiner & Juang, 1933). *Speech processing* dilakukan agar komputer dapat mengenali sinyal ucapan manusia (*speech recognition*). *Speech recognition* dapat mempermudah

pekerjaan manusia dalam berinteraksi dengan gawai karena hanya perlu mengucapkan kata sebagai masukan interaksi dengan gawai.

### 2.3 Klasifikasi Suara

Tipe suara adalah berbagai jenis suara yang diklasifikasikan menggunakan kriteria tertentu. Klasifikasi suara adalah proses dimana suara manusia dinilai, kemudian akan digolongkan menjadi tipe-tipe suara tertentu. Ada banyak perbedaan tipe suara berdasarkan berbagai macam sistem klasifikasi. Jenis suara manusia dibedakan menjadi enam jenis berdasarkan *gender* perempuan dan laki-laki yang mempunyai sifat dan karakter yang berbeda-beda.

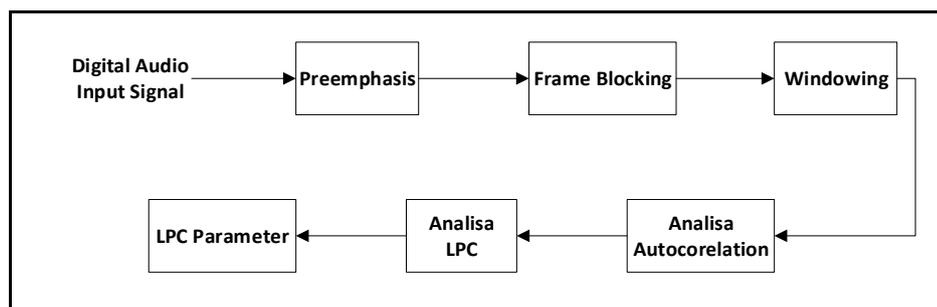
Tabel 1.1 Informasi Suara

Tipe Suara	<i>Range vocal</i>	Frekuensi <i>Range vocal</i>
Tenor	C3 - C5	130.813 - 523.251
Bariton	F2 - F4	87.3071 - 349.228
Bass	E2 - E4	82.4069 - 329.628
Sopran	C4 - C6	261.626 - 1046.50
Mezzo-Sopran	A3 - A5	220.000 - 880.000
Alto	F3 - F5	174.614 - 698.456

Sumber : (Sidabutar & Laksana, 2018)

### 2.4 Linear Predictive Coding (LPC)

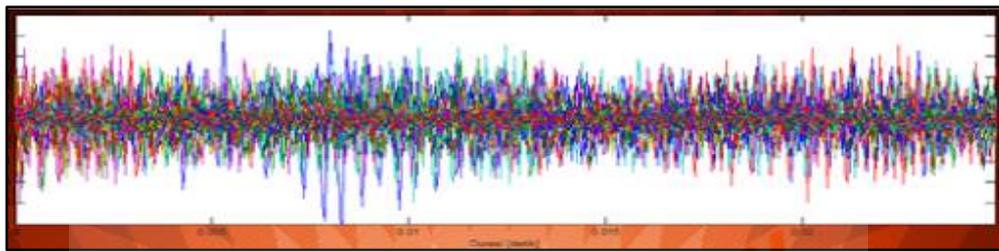
*Linear Predictive Coding* (LPC) merupakan salah satu metode analisis sinyal suara yang menyatakan ciri-ciri penting dari sinyal suara tersebut dalam bentuk koefisien-koefisien LPC. Enam langkah utama yang digunakan pada bagian ekstraksi ciri berbasis *Linear Predictive Coding* (LPC) adalah (Kurniawan, dkk, 2016) :



Gambar 2.1 Block Diagram LPC

### 2.4.1 Preemphasis

Sinyal suara digital,  $s(n)$ , dimasukkan melalui sistem digital berorde rendah (biasanya orde pertama filter FIR) untuk meratakan sinyal secara spektral dan untuk membuat sinyal rentan terhadap efek presisi dalam pemrosesan sinyal. Sistem digital yang digunakan dalam *preemphasizer* dapat berupa sistem yang konstan atau adaptif (misalnya, dengan kondisi transmisi rata-rata, latar belakang kebisingan, atau untuk spektrum sinyal rata-rata) (Rabiner & Juang, 1933).



Gambar 2.2 Sinyal *Preemphasis*

Sumber : (Silfiana, Ichwan, Dewi, 2018)

$$y(n) = s(n) - a s(n - 1) \quad 0,9 \leq a \leq 1 \quad (1)$$

Dimana :

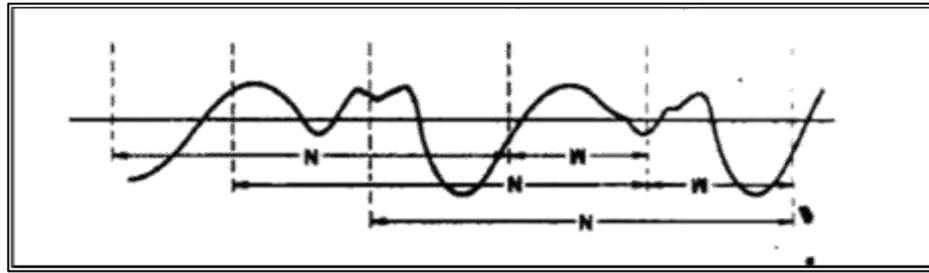
$\tilde{s}(n)$  = sinyal suara setelah dilakukan proses *preemphasis*.

$s(n)$  = sinyal suara sebelum dilakukan proses *preemphasis*.

$\tilde{a}$  = nilai yang paling sering digunakan yaitu 0.95.

### 2.4.2 Frame Blocking

Pada tahap ini, sinyal hasil tahapan *preemphasis* disegmentasi menjadi beberapa *frame* yang terdiri dari  $N$  sampel suara dengan jarak antar *frame* dipisahkan oleh  $M$  sampel. Jika  $M \leq N$ , beberapa *frame* yang berdekatan akan saling overlap dan hasil estimasi spektral akan berkorelasi dari *frame* ke *frame*. Sebaliknya, jika  $M > N$ , tidak akan ada overlap antara *frame* yang berdekatan sehingga beberapa isyarat sinyal suara akan hilang total. Jika *frame* ke- dari suara dinotasikan dengan dan terdapat *frame* di dalam keseluruhan sinyal suara (Amanda, dkk, 2019).



Gambar 2.3 Sinyal *Frame blocking*

Sumber: (Rabiner & Juang, 1933)

$$\begin{aligned}
 x_l n &= \tilde{x} Ml + n, \quad n = 0, 1, \dots, N - 1, \\
 l &= 0, 1, \dots, l - 1
 \end{aligned}
 \tag{2}$$

Dimana :

$x_l n$  = adalah sinyal ke- $n$  *frame blocking*

$M$  = adalah jarak antar *frame*

$N$  = adalah ukuran *frame* (*frame size*)

$l$  = adalah *frame*

### 2.4.3 Windowing

Tahap berikutnya pada pemrosesan sinyal adalah melakukan proses window pada setiap bagian sinyal yang telah dibuat sebelumnya untuk meminimalkan diskontinuitas di awal dan di akhir *frame*. Konsep *windowing* ini identik dengan interpretasi domain frekuensi waktu spektrum, menggunakan window untuk membuat sinyal pada awal dan akhir setiap *frame* menjadi nol. Jika window didefinisikan sebagai  $w(n)$ .

$$\tilde{x}_l(n) = x_l(n)w(n), \quad 0 \leq n \leq N - 1
 \tag{3}$$

Dimana :

$\tilde{x}_l(n)$  = adalah sinyal hasil proses *Windowing*

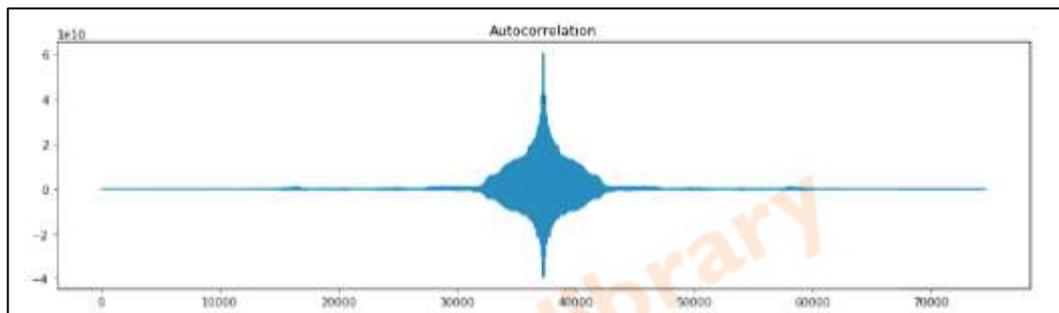
$x_l$  = adalah sinyal hasil proses *Frame blocking* ke- $n$

$W(n)$  = adalah fungsi *Window*

$N$  = adalah *Frame Size*, kelipatan 2

#### 2.4.4 Analisa Autocorelation

Setiap sampel suara yang telah dilakukan proses *windowing* maka akan dilakukan korelasi untuk menghasilkan sejumlah nilai yang dapat dibentuk menjadi toeplist matriks. Nilai tertinggi dari proses autocorellation ini adalah nilai orde analisis LPC (P). Nilai P umumnya berada pada interval 8 sampai 16. Autocorellation ini berfungsi untuk mengkorelasikan bentuk gelombang dengan dirinya sendiri. Analisis autocorellation ini dilakukan untuk mendapatkan elemen – elemen *autocorellation* dengan cara :



Gambar 2.4 Sinyal Autocorrelation

$$r_i(m) = \sum_{n=0}^{N-i-m} \tilde{x}_i(n) \cdot \tilde{x}_i(n+m), m = 0, 1, 2, \dots, p \quad (4)$$

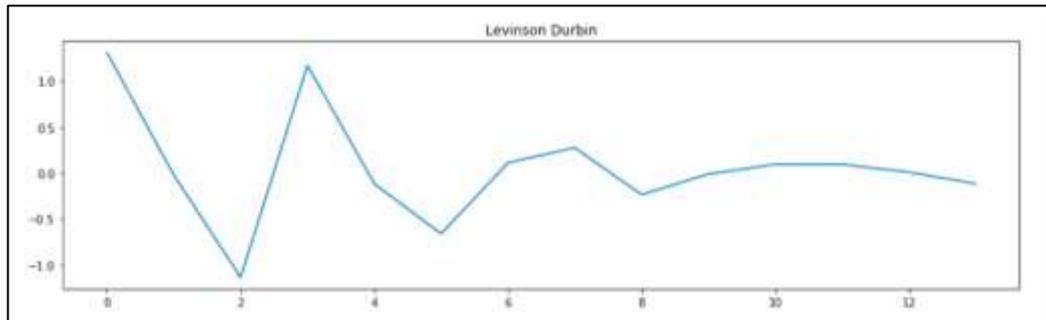
Dimana :

$r_i(m)$  = hasil sinyal *autocorrelation*

$\tilde{x}_i(n)$  = hasil sinyal *windowing* ke-n

#### 2.4.5 Analisis LPC

Proses selanjutnya adalah analisis LPC, yang mengubah setiap *frame* autokorelasi p+1 ke dalam bentuk parameter-parameter LPC atau yang biasa disebut dengan koefisien LPC. Metode yang biasa digunakan dalam analisis LPC ini adalah metode Durbin yang mempunyai algoritma sebagai berikut :



Gambar 2.5 Sinyal Analisis LPC

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

$$k_i = \frac{\left\{ r^{(i)} - \sum_{j=1}^{i-1} a_j^{(i-1)} r^{(i-j)} \right\}}{E^{(i-1)}}, 1 \leq i \leq p \quad (6)$$

$$a_i^{(i)} = k_i \quad (7)$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, 1 \leq j \leq i-1 \quad (8)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (9)$$

Dimana :

$r^{(i)}$  = nilai *autocorrelation* ke-i

$k_i$  = Koefesien refleksi

$E^{(i)}$  = Error prediksi

$a_j^{(i)}$  = persamaan koefesien LPC

Kemudian koefesien LPC yang dihasilkan sebagai berikut :

$$a_m = a_m^{(p)} 1 \leq m \leq p \quad (10)$$

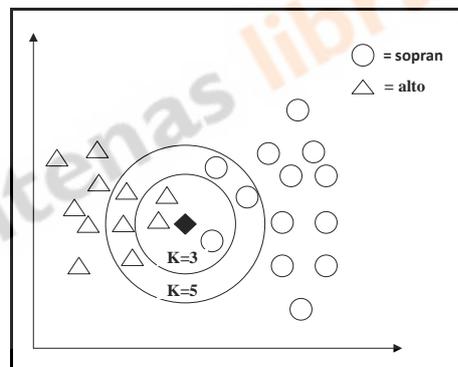
Dimana :

$a_m$  = Koefesien LPC

$p$  = Orde LPC

## 2.5 *K-Nearest Neighbor (KNN)*

Klasifikasi suara adalah suatu proses pengelompokan jenis suara yang dimana pada pengelompokan tersebut dilakukan berdasarkan kelas atau tipe atau jenis suara yang tersedia pada data latih. *K-Nearest Neighbor* merupakan suatu metode pendekatan untuk melakukan klasifikasi data dengan memperkirakan seberapa besar kemungkinan titik data uji untuk menjadi anggota dari suatu grup atau kelompok lainnya tergantung pada jarak terdekat dari titik pengujian. *K-Nearest Neighbor* merupakan jenis *supervised classification*, yang dimana hasil dari klasifikasi tersebut didapatkan berdasarkan *class majority* yang ada. Cara kerja dari algoritma *K-Nearest Neighbor* yaitu dengan mencari jarak data terdekat dengan titik data uji berdasarkan tipe atau jenis suara yang terdapat pada data latih (Alfath, dkk, 2019). Berikut ini merupakan ilustrasi cara kerja dari metode *K-Nearest Neighbor* yang dapat dilihat pada Gambar 2.5.



Gambar 2.6 Ilustrasi KNN

Metode klasifikasi KNN dipengaruhi oleh pemilihan ciri yang benar karena dapat memberikan kinerja klasifikasi yang lebih baik. Pada metode KNN klasifikasi ditinjau dari jarak (tetangga) yang paling dekat, sehingga tidak bisa dibandingkan dengan tetangga lainnya. Penerapan klasifikasi menggunakan KNN menjadi lebih sedikit, karena yang dipilih merupakan jarak yang paling dekat. Adapun langkah-langkah penyelesaian masalah menggunakan metode KNN adalah sebagai berikut (Rusydy, Irawan, & Setianingsih, 2017) :

1. Menentukan nilai parameter K (Jumlah tetangga terdekat)
2. Menghitung jarak setiap sampel data dengan data yang diuji
3. Mengurutkan data berdasarkan jarak dari yang terkecil hingga yang terbesar

4. Mengamati jumlah keputusan yang terbanyak untuk K data yang diambil
5. Jika terdapat dua atau lebih kelas yang merupakan tetangga terdekat dari data uji (x), maka terjadilah kondisi seimbang (konflik) dan digunakan strategi pemecahan konflik.
6. Untuk masing-masing kelas yang terlibat dalam konflik, tentukan jarak ( $d_i$ ) antara (x) dengan kelas berdasarkan (E) tetangga terdekat yang ditemukan pada kelas.
7. Jika pola pelatihan ke-m dari kelas yang terlibat dalam konflik maka jarak I antara x dengan kelas adalah :

$$d_i = \bar{E} \sum_{j=1}^n |(X_j - Y_j)| \quad (11)$$

Pengelompokan data diukur dengan menentukan dua objek mirip atau tidak mirip. Untuk menentukan kemiripan tersebut dapat digunakan pengukuran yang disebut dengan *distance measure*. Berikut merupakan jenis-jenis pada *distance measure* (Widya, dkk, 2020) :

1. *Euclidean Distance*

$$D_{L_2}(X_2, X_1) = \| X_2 - X_1 \|_2 = \sqrt{\sum_{j=1}^p (X_{2j} - X_{1j})^2} \quad (12)$$

Keterangan :

- P = Dimensi Data
- X1 = Posisi Titik 1
- X2 = Posisi Titik 2

2. *Manhattan Distance*

$$D_{L_1}(X_2, X_1) = \| X_2 - X_1 \|_{11} = \sqrt{\sum_{j=1}^p |X_{2j} - X_{1j}|} \quad (13)$$

Keterangan :

- P = Dimensi Data
- X1 = Posisi Titik 1
- X2 = Posisi Titik 2
- || = Nilai Absolut