

# Analisa dan Implementasi Transformasi Analog to Digital Converter (ADC) untuk Mengkonversi Suara Kebentuk Teks

Elgamar Syam

Fakultas Teknik, Jurusan Teknik Informatika Universitas Islam Kuantan Singingi  
[gamar.tc@gmail.com](mailto:gamar.tc@gmail.com)

## Abstrak

Teknologi telah banyak membawa perubahan pada kehidupan manusia, sehingga memberikan dampak positif dan negatif. Menurut hasil beberapa penelitian menunjukkan bahwa daya serap manusia terhadap sebuah informasi adanya ketidaksamaan tingkatan, salah satu yang menyebabkan demikian adalah pilihan jenis media yang digunakan. Jenis media komunikasi audio visual ternyata yang paling tinggi, audio, dan selanjutnya berupa teks. Transformasi Analog to Digital Converter (ADC) merupakan piranti yang berfungsi merubah sinyal analog menjadi sinyal digital. Dalam sub sistem ADC, untuk mendapatkan frekuensi yang di inginkan, melalui tiga tahapan yaitu *sampling*, *quantizing*, dan *pengkodean*, kemudian membandingkan frekuensi ini dengan frekuensi sinyal digital suara yang masuk dengan database kata-kata yang sudah di inputkan sebelumnya sehingga menghasilkan sebuah output berupa *text-on-screen*.

Kata Kunci : *Speech Recognition*, *Speech to Text*, *x-webkit*, *Analog to Digital Converter*.

## 1. Pendahuluan

Adanya perbedaan daya serap manusia terhadap informasi dan ketidaksamaan tingkatan, yang salah satunya disebabkan pemilihan media komunikasi. Salah satu perkembangan perangkat lunak audio adalah perangkat lunak yang dapat mendeteksi suara (*speech recognition*) yang diucapkan *user* dan mengolahnya menjadi sebuah *output*. Perangkat lunak jenis ini dapat membantu pekerjaan manusia khususnya dibidang pengetikan teks. Perangkat lunak ini menggunakan teknologi *voice recognition* (pengenalan suara) dimana suara pengguna akan dikenali berdasarkan amplitudo getaran suaranya.

Teknologi *voice recognition* 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 disimpan dalam suatu perangkat. Analisa proses teknologi transformasi *analog to digital converter* (ADC) ini memanfaatkan *google speech API* dan yang menjadi sumber masukan berupa sinyal analog suara.

Pemanfaatan *Google Speech Application Programming Interface* (*Google Speech API*) secara sederhana bisa diartikan sebagai kode program yang merupakan *interface* antara aplikasi berbasis web yang dibuat dengan fungsi-fungsi yang dikerjakan, untuk mengakses, menjalankan, memanfaatkan fungsi atau fitur yang disediakan *Google Speech-API*. Adapun rumusan masalah yang ditekankan dalam penelitian ini adalah : Bagaimana menganalisa dan implementasi transformasi *analog to digital converter* dari sinyal analog suara sehingga menghasilkan sinyal digital *text-on-screen*?

Berdasarkan latar belakang dan perumusan masalah yang di atas, maka penelitian ini diberi batasan yaitu : Sumber masukan berupa sinyal analog suara dalam bahasa inggris dengan menggunakan kata singkat seperti "*food*" dan "*okay*", agar proses penganalisaan bit sinyal analog tidak terlalu banyak.

Tujuan dari penelitian ini adalah memahami algoritma proses transformasi sinyal analog suara ke sinyal digital berupa *text-on-screen*.

## 2. Tinjauan Pustaka

Transformasi *analog to digital* merupakan teknologi *voice recognition* yaitu suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. 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 teks.

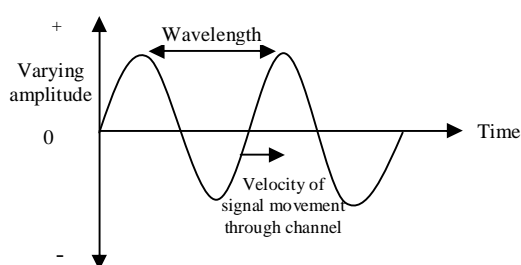
### 2.1 Sinyal Analog dan Digital

Sinyal analog bekerja dengan mentransmisikan suara dan gambar dalam bentuk gelombang kontinu (*continous varying*). Dua parameter atau karakteristik

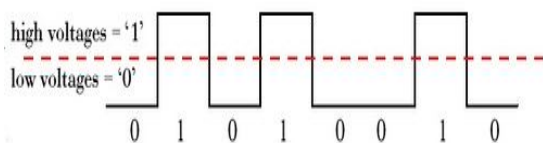
terpenting yang dimiliki oleh isyarat analog adalah amplitude dan frekuensi.

Komputer analog adalah istilah yang digunakan untuk menggambarkan alat penghitung yang bekerja pada *level* analog. *Level* analog di sini adalah lawan dari *level* digital, yang mana *level* digital adalah *level* tegangan 'high' (tinggi) dan 'low' (rendah), yang digunakan dalam implementasi bilangan biner (Arief Budijanto dan Slamet Winardi, 2012).

Sinyal digital merupakan hasil teknologi yang dapat mengubah signal menjadi kombinasi urutan bilangan 0 dan 1, sehingga tidak mudah terpengaruh oleh derau, proses informasinya pun mudah, cepat dan akurat, tetapi transmisi dengan sinyal digital hanya mencapai jarak jangkauan pengiriman data yang relatif dekat.



Gambar 1. Sinyal Analog



Gambar 2. Sinyal Digital

## 2.2 Voice Recognition

Pengenalan suara (*voice recognition*) dalam perkembangan teknologinya dibagi menjadi dua jenis, yaitu *speech recognition* dan *speaker recognition*.

## 2.3 Speech Recognition

*Speech recognition* adalah proses identifikasi yang dilakukan komputer untuk mengenali kata yang diucapkan oleh seseorang tanpa memperdulikan identitas orang terkait dengan melakukan konversi sebuah sinyal akustik, yang ditangkap oleh audio device (perangkat input suara).

Teknologi *speech recognition* membutuhkan beberapa perangkat pendukung agar dapat mengubah percakapan menjadi teks *on-screen* atau perintah tertentu :

### 1. Sound card

Merupakan perangkat yang ditambahkan dalam suatu komputer yang fungsinya sebagai perangkat input

dan output suara untuk mengubah sinyal elektrik, menjadi analog maupun menjadi digital.

### 2. Microphone

Merupakan perangkat input suara yang berfungsi untuk mengubah suara yang melewati udara, air dan benda lainnya menjadi sinyal elektrik.

### 3. Computer

Proses suara digital menterjemahkan gelombang suara menjadi suatu symbol dan biasanya menjadi suatu nomor biner yang dapat diproses lagi, kemudian diidentifikasi dan dicocokkan dengan *database* yang berisi berkas suara agar dapat dikenali.

## 2.4 Speaker Recognition

*Speaker Recognition* disebut juga *Speaker Verification* merupakan proses verifikasi seorang pembicara, yang mana sebelumnya telah diketahui identitas pembicara tersebut berdasarkan data yang telah di inputkan. *Speaker verification* melakukan perbandingan *one to one* (1:1). Dalam arti bahwa fitur-fitur suara dari seorang pembicara dibandingkan secara langsung dengan fitur-fitur seorang pembicara tertentu yang ada dalam sistem. Bila hasil perbandingan (skor) tersebut lebih kecil atau sama dengan batasan tertentu (*threshold*), maka pembicara tersebut diterima, bila tidak maka akan ditolak (dengan asumsi semakin kecil skor berarti kedua sampel semakin mirip), (Rezza Aditya, 2012).

*Speaker Recognition* melalui dua tahapan proses identifikasi dan verifikasi. Identifikasi ialah mengenali identitas subyek, dilakukan perbandingan kecocokan antara data *biometric* subyek dalam *database* berisi record karakter subyek. Sedangkan verifikasi adalah menentukan apakah subyek sesuai dengan apa yang dikatakan terhadap dirinya.

## 2.5 Google Speech-API

Google *Speech API* bisa dikatakan bagian dari *Framework* Google yang menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang *web* maupun aplikasi desktop untuk memanfaatkan berbagai fitur yang disediakan oleh Google seperti *AdSense*, *Search Engine*, *Translation* maupun *YouTube*.

API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang dibuat dengan fungsi-fungsi yang dikerjakan. Dalam hal ini Google *Speech API* berarti kode program yang disederhanakan sehingga dapat ditambahkan pada aplikasi atau *web* agar dapat mengakses, menjalankan, dan memanfaatkan fungsi atau fitur yang disediakan Google. Ada banyak API yang disediakan oleh Google, beberapa diantaranya adalah :

1. *Language* API, untuk memanfaatkan fitur *translation* yang dimiliki Google.
2. *Earth* API, memanfaatkan fitur yang ada pada *Google Earth*.
3. *Javascript* API.
4. *Maps* API, memanfaatkan fitur yang ada pada *Google Maps*.
5. *Search* API, memanfaatkan fitur pencarian pada *Google Search*.
6. *Visualization* API, membuat grafik maupun *chart* dengan *Google API*.
7. *YouTube* API, memanfaatkan fitur yang ada pada *YouTube* misalnya untuk pencarian video.

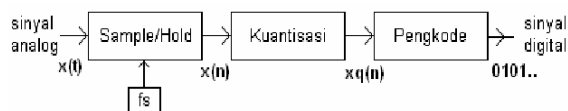
```
<input name="text" type="text" id="speechInput" style="font-size:25px;" size="45"
x-webkit-speech onwebkitspeechchange="onChange()" />
```

Gambar 3. Kode program *speech-API*

## 2.6 Analog to Digital Converter (ADC)

Analog to digital converter (ADC) adalah piranti pengkonversi sinyal analog ke sinyal digital. Analog to Digital Conversion (ADC) merupakan piranti elektronik yang berfungsi sebagai pengubah sinyal analog menjadi sinyal digital.

ADC berperan dalam hal memindahkan data dan informasi dari dua sistem yang berbeda. Hasil ubahan oleh ADC adalah dalam bentuk kode-kode biner berupa bit-bit karakter baik secara paralel ataupun seri.



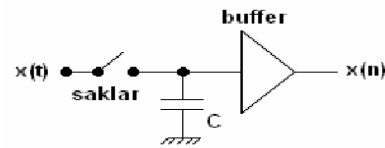
Gambar 4. Proses konversi sinyal analog menjadi sinyal digital

## 2.7 Algoritma Sinyal Analog ke Digital

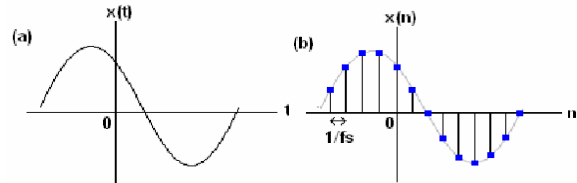
Dalam mengolah sinyal analog suara menjadi sinyal digital *text-on-screen*, teknologi *speech to text* melalui 7 proses sebagai berikut :

1. *Input Speech* (ucapan).  
Suara diinputkan melalui sebuah perangkat yang dihubungkan ke komputer yaitu *microphone*.
2. *Sampling*.  
Merupakan suatu proses mengambil data signal *continue* untuk setiap periode tertentu (per detik), yang nantinya diubah ke signal *discrete*. *Sampling* atau *sampling rate* diukur dalam *hertz* (Hz), semakin tinggi *sampling rate* maka akan semakin kuat resolusi *file* suara tersebut. Sebagai contoh, suara 16 bit dengan 44,1 Khz bermakna suara

tersebut disampling 44.100 kali per detik dan diukur dengan akurasi 16 bit.

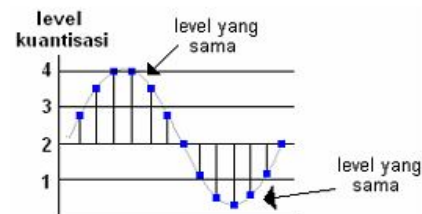


Gambar 5. Pengambilan sinyal analog ke digital pada *buffer* ADC.



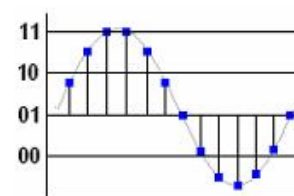
Gambar 6. Pencuplikan sinyal analog dan sinyal digital.

3. *Quantizing* (kuantisasi).  
Proses untuk mengelompokkan elemen-elemen yang nilainya kontinyu. Penggunaan umum kuantisasi adalah konversi dari sinyal kontinyu menjadi sinyal diskrit oleh *quantizing*. Kedua langkah ini (pengambilan sampel dan *quantizing*) dilakukan dalam *analog-to-digital converter* dengan tingkat kuantisasi ditentukan dalam bit. Sinyal suara dideteksi di bagian puncak, baik bagian puncak positif maupun negatif.



Gambar 7. Proses kuantisasi

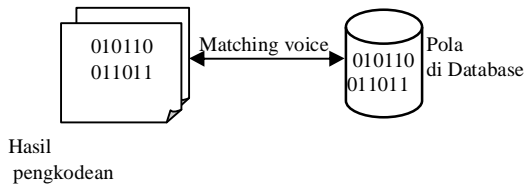
4. *Pengkodean*.  
Proses pengkodean dalam ADC (*analog-to-digital conversion*) menetapkan bilangan biner tertentu pada tiap *level* kuantisasi.



Gambar 8. Proses pengkodean

5. *Matching Voice*.  
*Matching voice* merupakan suatu tahapan di mana suara masuk dan yang sudah di rubah kedalam sinyal diskrit dibandingkan dengan pola yang telah ada didalam *database*. Apabila suara yang masuk

cocok dengan pola yang ada, maka akan menampilkan teks yang sesuai, namun apabila tidak cocok maka akan menampilkan teks yang salah.



**Gambar 9. Matching voice**

#### 6. Validation

Tahap validasi merupakan tahap pembuktian sinyal digital hasil pengolahan *analog to digital*, yang terjadi dalam perangkat ADC (*Analog to Digital Converter*) merupakan suara manusia. Jika frekuensi suara (20 Hz -20.000 Hz) maka diketahui bahwa sinyal suara tersebut adalah suara manusia.

#### 7. Text on Screen.

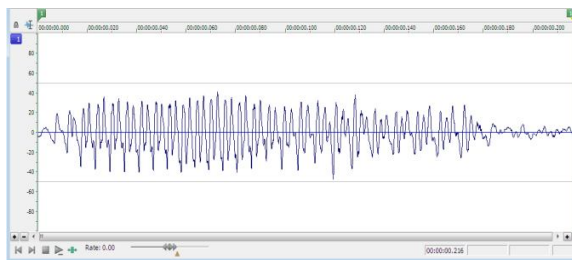
Hasil atau *output* yang berupa teks yang ditampilkan pada *interface* atau aplikasi yang dibuat dari proses pengolahan suara yang memanfaatkan *google speech-API*. Teks yang dihasilkan bisa bernilai benar atau salah, tergantung pada fonem dan gangguan (*noise*) suara dari lingkungan yang ada disekitar.

### 3. Analisa dan Pengolahan Data Sampel

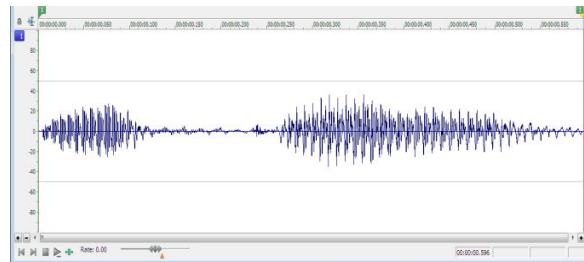
Proses pengolahan data dilakukan melalui proses perekaman suara (*voice recording*), kemudian di analisa dan dilakukan implementasi sistem.

#### Voice Recording

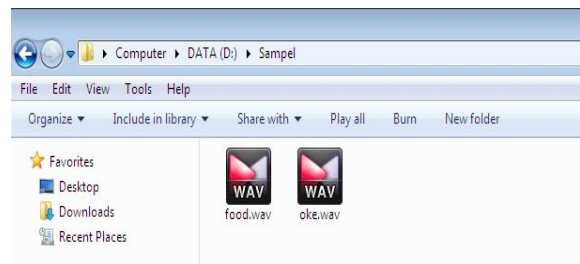
Proses perekaman suara (*voice recording*), dilakukan dengan pengucapan 2 sampel suara “*food*” dan “*okay*” dalam bahasa inggris, menggunakan sebuah *software* (sound forge 10) dengan sampel rate 44.100 Hz dan bit depth 16 bit, kemudian dilakukan identifikasi terhadap spektrum suara tersebut, sehingga menghasilkan sebuah informasi.



**Gambar 10. Sampel spektrum suara “food.wav”**



**Gambar 11. Sampel spektrum suara “okay.wav”**



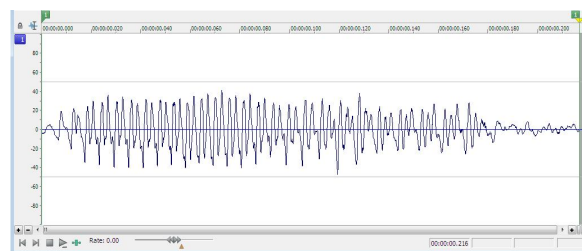
**Gambar 12. File recording, sampel spektrum “food.wav” dan “okay.wav”**

#### Analisa Voice Recording

Dalam proses transformasi sinyal analog (suara) ke bentuk teks semuanya terjadi di dalam piranti *analog to digital converter* (ADC). Secara umum terjadi melalui 7 proses, *input speech*, *sampling*, *quantizing*, *pengkodean*, *matching voice*, *validation*, *text-on-screen*.

##### 1. Tahap satu, *input speech*.

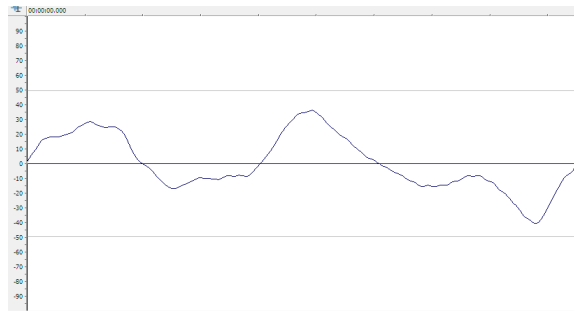
Pada tahap pertama, suara diinputkan melalui sebuah perangkat keras yaitu “*microphone*” atau “*earphone*” sebagai alat input suara, yang dihubungkan pada komputer pengolah suara. Pada proses pengujian ini di ambil salah satu sampel suara hasil *recording*, yaitu suara “*food*”.



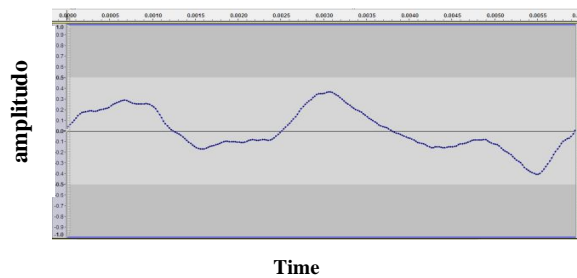
**Gambar 13. Sampel spektrum “food.wav”**

##### 2. Tahap dua, *sampling*.

Pada proses *sampling*, dilakukan pengambilan sinyal *continue* (analog) suara, hasil *input speech* dari *microphone* yang berupa pengucapan kata *food*.



**Gambar 14. Sinyal Asli “food.wav” dalam 1 Periode**



**Gambar 15. Sinyal Hasil Sampling “food.wav” dalam 1 Periode**

Pada gambar 14 memperlihatkan sinyal hasil *sampling* “*food.wav*” dalam titik-titik periode waktu, dengan jarak 0.00580 seconds, dan sebanyak 50 sampel.

**Tabel 1. Jarak titik-titik amplitude waktu**

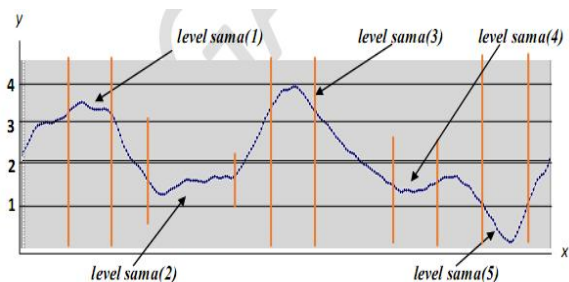
food-sampling.csv 1 channel (mono)	
One column per channel.	
Sample Rate: 44100 Hz. Sample values on linear scale.	
Length processed: 50 samples 0.00580 seconds.	
No.	Amplitudo /Waktu (seconds)
1	0.01074
2	0.03723
3	0.06009
4	0.07864
5	0.09644
6	0.11938
7	0.1413
8	0.15955
9	0.1705
10	0.17514
11	0.17896
12	0.18329
13	0.18549
14	0.1846
15	0.18314
16	0.18344
17	0.18747
18	0.19269

19	0.19675
20	0.20123
21	0.20532
22	0.21057
23	0.22003
24	0.23383
25	0.24722
26	0.25558
27	0.26361
28	0.27203
29	0.27774
30	0.28342
31	0.28671
32	0.28229
33	0.27475
34	0.2673
35	0.26044
36	0.25574
37	0.25204
38	0.24942
39	0.24954
40	0.25101
41	0.25214
42	0.25272
43	0.25137
44	0.24533
45	0.23608
46	0.2244
47	0.20651
48	0.18185
49	0.15195
50	0.12045

Tabel 1 memperlihatkan jarak setiap titik-titik amplitudo dalam 1 periode per satuan waktu dengan 50 sampel, dan jarak tiap titik 0.00580 seconds, sample rate 44.100 Hz dan terdapat nilai positif dan negatif.

3. Tahap Tiga “*Quantizing*”.

Tahapan kuantisasi (*quantizing*) merupakan tahapan pengelompokan titik-titik sinyal, sehingga nantinya akan ditentukan nilai biner pada setiap titik. Pada tahapan ini dibagi menjadi 4 level dan dikelompokkan titik-titik level yang sama.

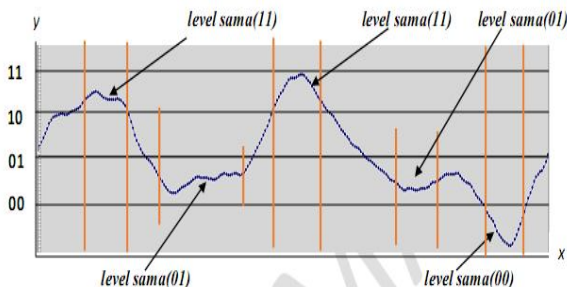


**Gambar 16. Sinyal Hasil Kuantisasi “food.wav” dalam 1 periode**

Pada gambar 15 terlihat pengelompokan titik-titik pada level-level yang sama, yang berarti memiliki nilai biner yang sama.

#### 4. Tahap Empat (Pengkodean)

Proses pengkodean merupakan tahapan akhir dalam pengolahan analog ke digital. Selanjutnya nilai-nilai biner ini dibandingkan dengan *database* pola (*database server google speech-API*) dan dibutuhkan 2 bit saja.

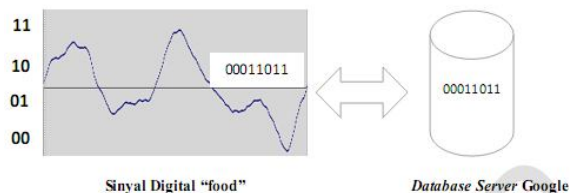


**Gambar 17. Pengkodean Sinyal "food" dalam 1 Periode**

Gambar 16 memperlihatkan klasifikasi sinyal suara "food", berdasarkan nilai biner. Setelah kode biner didapat maka dilakukan *matching voice* terhadap pola yang ada di *database server google*.

#### 5. Tahap Lima (Matching Voice)

Proses *matching voice* dilakukan untuk mempresentasikan sinyal input yang telah berubah menjadi sinyal digital atau bernilai biner, dengan sinyal digital bernilai biner yang ada pada *database server google*. Kemiripan sinyal input dengan sinyal digital yang ada pada *database* inilah yang dibandingkan sehingga nantinya menghasilkan *text-on-screen* yang sesuai atau tidak, jika sinyal tersebut mendekati maka hasil teks yang muncul akan sesuai dengan yang diucapkan, kemudian sebaliknya jika sinyal input hasil digitalisasi tidak mirip dengan nilai biner yang ada pada pola maka teks yang ditampilkan tidak sesuai.



**Gambar 18. Ilustrasi proses matching voice**

#### 6. Tahap Enam (Validasi)

Tahap validasi merupakan tahap pembuktian sinyal digital hasil pengolahan *analog to digital*, yang terjadi dalam perangkat ADC (*Analog to Digital Converter*) merupakan suara manusia. Jika frekuensi suara (20 Hz

-20.000 Hz) maka diketahui bahwa sinyal suara tersebut adalah suara manusia.

#### 7. Tahap Tujuh (Text On Screen)

Tahap ketujuh merupakan tahap akhir dan output dalam proses *speech to text*, teks ditampilkan pada aplikasi, teks yang tampil dipengaruhi oleh *fonem, lexical dan noise*.



**Gambar 19. Hasil pengucapan "food"**

### Analisa Hasil

Analisa hasil memaparkan hasil yang didapat dalam 7 tahapan atau proses pembentukan *speech to text*. Berdasarkan hasil pengolahan data suara, pembentukan sinyal suara melalui tiga tahap awal (*sampling, quantizing, dan pengkodean*) untuk mengkonversi sinyal analog (suara) menjadi sinyal digital (teks). *Output* yang ditampilkan berupa *text-on-screen* dipengaruhi beberapa faktor sehingga menghasilkan teks yang sesuai. *Fonem, lexical dan noise* merupakan hambatan dalam pembentukan sinyal ke teks.

Dari hasil *voice recording*, maka dapat diketahui sebuah informasi dari setiap hasil *recording* suara.

**Tabel 2. Informasi Sampel Data Berdasarkan Hasil Recording Menggunakan Sound Forge 10.0.**

File Name	File Type	Bit Rate/Bit Depth	Sample Rate	File Size	Length/Second
Food	wav	705 Kbps /16 Bit	44,100 Hz	18.9 KB	00:00:00.217
Okay	wav	705 Kbps /16 Bit	44,100 Hz	51.6 KB	00:00:00.596

Berdasarkan dari analisa *sample rate* dan *bit depth*, dapat diketahui bahwa semakin tinggi *sample rate* dan *bit depth* suatu suara maka akan menghasilkan suara yang lebih jernih.

### 4. Kesimpulan

Dari hasil analisa terhadap pengolahan data, maka dapat disimpulkan suatu kondisi yang mana sebagai acuan nantinya dalam proses implementasi dan

pengujian. Adapun kesimpulan yang dapat ditarik adalah :

1. Proses *input speech* dipengaruhi oleh beberapa faktor antara lain, *fonem*, *lexical*, dan *noise* (kebisingan di sekitar *user*).
2. Tinggi dan rendahnya *sample rate* dan *bit depth* sangat mempengaruhi kejernihan suatu suara.
3. Semakin pendek durasi (*length*) suara, maka akan menghasilkan *bit rate* yang kecil.
4. Seluruh kegiatan atau proses yang terjadi pada pembentukan sinyal analog ke sinyal digital berlangsung dalam sebuah perangkat ADC (*analog to digital*) yang melakukan transformasi suara.

Proses *matching voice* dan validasi suara dapat terjadi, dengan menghubungkan aplikasi yang dibuat dengan layanan internet (*database server google*).

## Referensi

- [1] Sonia Sunny., David Peter S., and K Poulouse Jacob, (2012). "*Recognition Of Speech Signals : An Experimental Comparison Of Linear Predictive Coding And Discrete Wavelet Transforms.*" India :
- [2] Cochin University Of Science and Technology.1594-1601.
- [2] Anusuya, M.A. and Katti, S.K. (2009). "*Speech Recognition by Machine : A Review.*"India : Sri Jaya Chamarajendra College Of Engineering Mysore. 181-205.
- [3] Syaiful Alam (2008), "Perancangan dan Simulasi Sistem Kompresi Suara dengan Transformasi DCT" Indonesia : Universitas Lampung.
- [4] Ferda Ernawan and NurAzman Abu. (2011), "*Spectrum Analysis Of Speech Recognition Via Discrete Tchebichef Transform*" Indonesia, Malaysia :Universitas Dian Nuswantoro (UDINUS) and Universiti Teknikal Malaysia Melaka (UTeM).
- [5] Rezza Aditya (2012), "Prototipe Pengenalan Suara Sebagai Penggerak Dinamo Starter Pada Mobil" Indonesia : Universitas Gunadarma.
- [6] Darma Putra dan Adi Resmawan. (2011). Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW" Indonesia : Universitas Udayana.
- [7] Fadlisyah, Bustami, dan Ikhwanus, M (2013). "Pengolahan Suara." 1<sup>th</sup>. Yogyakarta. viii + 1-130.