

MODUL 1

PROSES PEREKAMAN DAN PENGEDITAN SINYAL WICARA

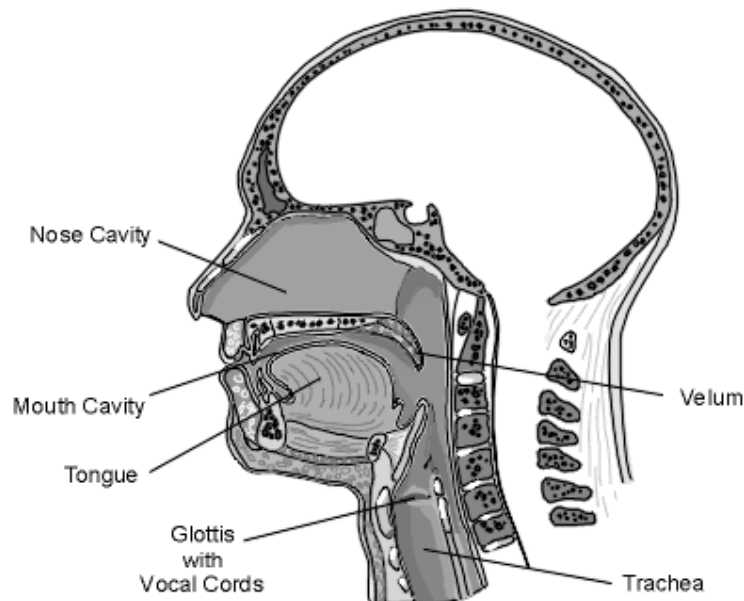
I. TUJUAN

- Mahasiswa mampu melakukan proses perekaman dan pengeditan sinyal wicara dengan menggunakan perangkat lunak.

II. DASAR TEORI

2.1. Pembangkitan Sinyal Wicara pada Manusia

Speech (wicara) dihasilkan dari sebuah kerjasama antara *lungs*(paru-paru), *glottis* (dengan *vocal cords*) dan *articulation tract* (*mouth/mulut* dan *nose cavity/rongga hidung*). Gambar 1 menunjukkan penampang melintang dari organ wicara manusia. Untuk menghasilkan sebuah *voiced sounds* (suara ucapan), paru-paru lungs menekan udara melalui *epiglottis*, *vocal cords* bergetar, menginterupt udara melalui aliran udara dan menghasilkan sebuah gelombang tekanan *quasi-periodic*.

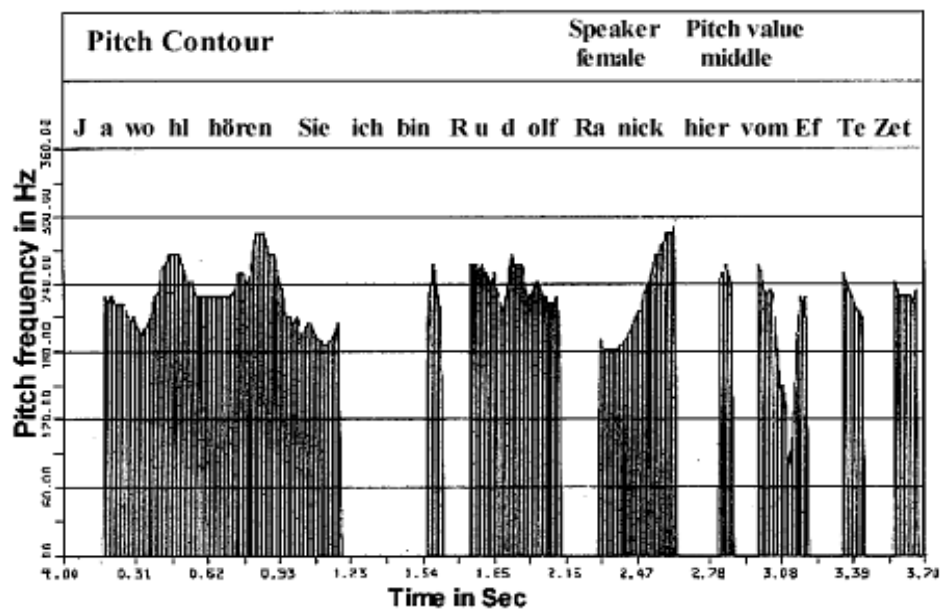


Gambar 1. Organ wicara manusia

Impuls tekanan pada umumnya disebut sebagai *pitch impulses* dan frekuensi sinyal tekanan adalah *pitch frequency* atau *fundamental frequency*. Di dalam Gambar 2a sederetan impuls (fungsi tekanan suara) dihasilkan oleh *vocal cords* untuk sebuah suara. Ini merupakan bagian dari sinyal voice (suara) yang mendefinisikan *speech melody* (melodi wicara). Ketika kita berbicara dengan sebuah frekuensi *pitch* konstan, suara sinyal wicara *monotonous* tetapi dalam kasus normal sebuah perubahan permanen pada frekuensi terjadi. Variasi frekuensi *pitch* dapat dilihat seperti pada Gambar 2b.



a. Sederetan impulse yang sama

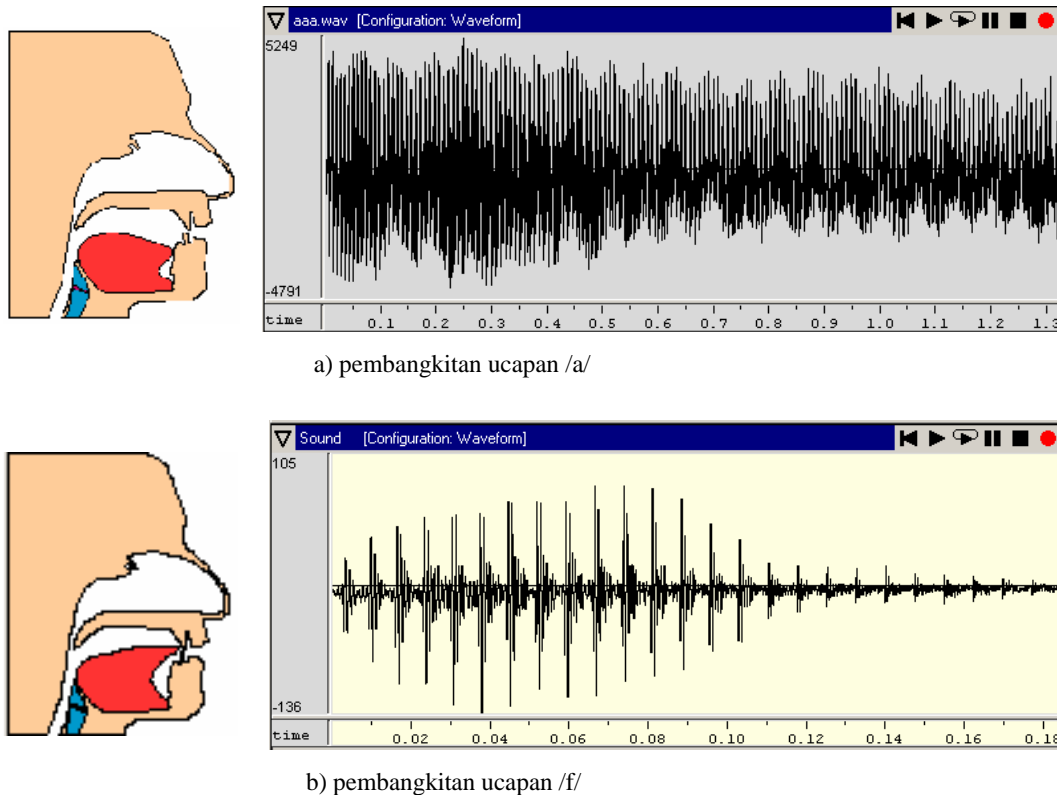


b. Variasi pada frekuensi pitch

Gambar 2. Sederetan impuls dan pitch pada sinyal wicara

Impuls *pitch* merangsang udara di dalam mulut, dan untuk suara tertentu (nasals) juga merangsang *nasal cavity* (rongga hidung). Ketika rongga beresonansi, akan menimbulkan radiasi sebuah gelombang suara yang mana merupakan sinyal wicara. Kedua rongga beraksi sebagai resonators dengan karakteristik frekuensi resonansi masing-masing, yang disebut *formant frequencies*. Pada saat rongga mulut dapat mengalami perubahan besar, kita mampu untuk menghasilkan beragam pola ucapan suara yang berbeda.

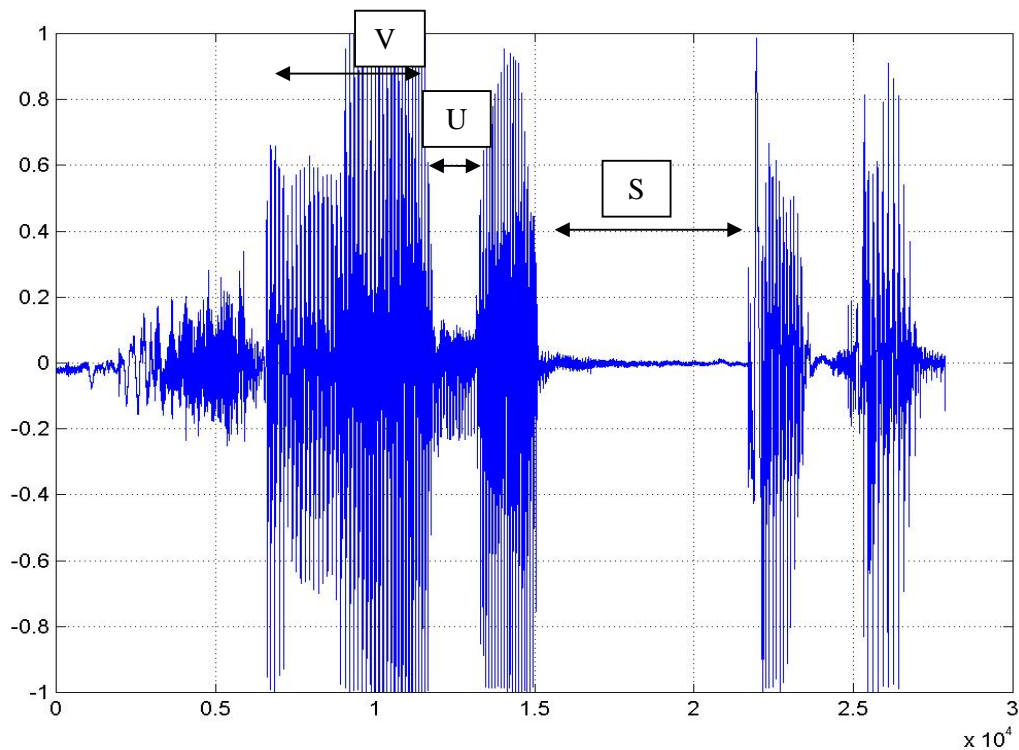
Di dalam kasus *unvoiced sounds* (suara tak terucap), exitasi pada *vocal tract* lebih menyerupai *noise* (derau). Gambar 3 menampilkan proses produksi suara-suara /a/, dan /f/. Untuk sementara perbedaan bentuk dan posisi pada organ *articulation* diabaikan saja.



Gambar 3. Proses produksi suara

2.2. Bentuk Sinyal Wicara dalam Domain Waktu

Sinyal wicara merupakan sinyal yang bervariasi lambat sebagai fungsi waktu, dalam hal ini ketika diamati pada durasi yang sangat pendek (5 sampai 100 mili detik) karakteristiknya masih stasioner. Tetapi bilamana diamati dalam durasi yang lebih panjang ($\geq 1/5$ detik) karakteristik sinyalnya berubah untuk merefleksikan suara ucapan yang keluar dari pembicara.



Gambar 4. Contoh sinyal wicara ucapan “Selamat Datang”

Salah satu cara dalam menyajikan sebuah sinyal wicara adalah dengan menampilkannya dalam tiga kondisi dasar, yaitu *silence* (S) atau keadaan tenang dimana sinyal wicara tidak diproduksi, *unvoice* (U) dimana *vocal cord* tidak berfibrasi, dan yang ketiga adalah *voiced* (V) dimana *vocal cord* bervibrasi secara periodik sehingga menggerakkan udara ke kerongkongan melalui mekanisme akustik sampai keluar mulut dan menghasilkan sinyal wicara.

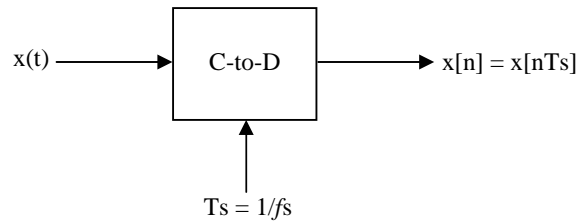
2.3. Proses Sampling

Perhatikan sinyal sinus berikut ini:

$$x(t) = A \cos(\omega t + \phi) \quad (1)$$

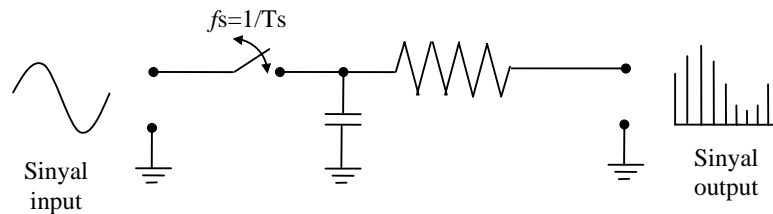
Sinyal tersebut merupakan contoh sinyal waktu kontinyu. Kita juga seringkali menggunakan terminologi sinyal analog untuk menyebutnya.

Untuk proses komputasi, sinyal waktu kontinyu harus dirubah menjadi bentuk waktu diskrit dan dilanjutkan dengan proses digitalisasi. Untuk memperoleh bentuk sinyal waktu diskrit, sinyal waktu kontinyu harus di-sampel.



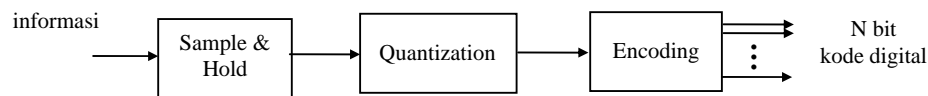
Gambar 5. Blok diagram konversi sinyal kontinyu menjadi sinyal diskrit

Sekuen $x[n]$ didapatkan setelah proses perubahan dari *continues to discrete* (C-to-D). Kondisi realnya secara hardware adalah menggunakan rangkaian sampling seperti Gambar 6 berikut ini.



Gambar 6. Rangkaian Sampling

Rangkaian sampling diatas merupakan sebuah ujung tombak dari sebuah *analog to digital conversion* (ADC).



Gambar 7. Blok diagram rangkaian ADC

Persyaratan frekuensi sampling menurut teorema *Shannon* harus sama dengan atau melebihi 2 kali frekuensi sinyal yang di sample.

$$f_s \geq 2 \times f_i \quad (2)$$

Jika sinyal informasi yang kita sample memiliki komponen frekuensi beragam, misalnya untuk sinyal wicara memungkinkan untuk memiliki frekuensi dari 20 sampai 4000 Hz, maka sinyal informasi tersebut bisa dituliskan sebagai:

$$x(t) = \sum_{i=1}^{i_{\max}} \sin(2\pi f_i t) \quad (3)$$

Dan persyaratan untuk frekuensi smpling menjadi:

$$f_s \geq 2 \times f_{\max} \quad (4)$$

Frekuensi sampling seringkali dikatakan dengan terminology *sampling rate*, yaitu jumlah sample yang diambil setiap detik, $f_s = 1/T_s$ yang juga dikenal sebagai *Nyquist rate*.

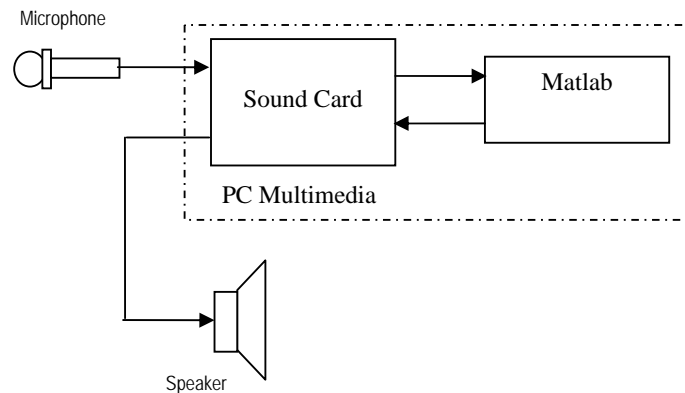
III. PERANGKAT YANG DIPERLUKAN

- 1 (satu) buah PC Multimedia lengkap sound card dan microphone
- Satu perangkat lunak Matlab

IV. LANGKAH-LANGKAH PERCOBAAN

4.1. Penataan Perangkat

Sebelum melakukan percobaan siswa harus melakukan penataan seperti pada Gambar 8 berikut ini.



Gambar 8. Penataan perangkat percobaan recording dan editing

PC yang digunakan harus dilengkapi dengan peralatan multimedia seperti sound card, speaker aktif dan *microphone*. Untuk *microphone* dan speaker aktif bias juga digantikan dengan *head set* lengkap. Sebelum memulai praktikum, sebaiknya dites dulu, apakah seluruh perangkat multimedia sudah terintegrasi dengan PC.

4.2 Perekaman dengan Matlab

Praktikum Pengolahan Informasi Wicara dengan langkah pertama adalah melakukan proses recording (perekaman) suara. Untuk itu ikuti langkah berikut ini.

1. Aktifkan Matlab, seperti pada pertemuan pertama saat mengenal Matlab Audio.

Siapkan microphone dan perangkat multimedia.

2. Buat program berikut ini untuk melakukan recording dengan Matlab

```
clear all;
Fs = 8000;
y = wavrecord(5.0*Fs, Fs, 'double');
wavwrite(y,Fs,'aiueo.wav')
```

3. Untuk mengetahui apakah proses perekaman yang telah dilakukan berhasil, cobalah untuk membuat perintah tambahan pada program diatas sehingga bisa membaca file *.wav hasil perekaman.
4. Tambahkan juga pada program bagaimana cara memainkan file *.wav yang sudah direkam. Kalau mengalami kesulitan buka kembali catatan tentang Matlab Audio yang ada di bagian lain pada buku ini.

4.3. Proses Pengeditan untuk Pemisah Vokal

Apabila telah selesai berdiskusi dengan teman terdekat, coba aktifkan kembali Matlab, dan lanjutkan dengan langkah berikut ini.

1. Buka file hasil rekaman “aiueo.wav”. Tampilkan dengan perintah plot, dan coba perhatikan dengan seksama durasi sinyal hasil dari proses perekaman tersebut.
2. Coba tampilkan sebagian saja dari seluruh sinyal, caranya adalah sebagai berikut:

```
y1=wavread('aiueo.wav');
t=length(y1);
y2=y1(1:10000);
plot(y2)
```

3. Sekarang coba buat program yang hanya membuat tampilan untuk vokal 'a' saja, dan jangan lupa mengujinya dengan melalui gambar dan mendengarkan suara yang dihasilkan.
4. Simpan ke dalam suatu file a.wav.

```
wavwrite(y1,Fs,'a.wav');
```
5. Lakukan hal yang sama untuk vocal 'i', 'u', 'e' dan 'o'. Semuanya juga harus disimpan dalam bentuk file *.wav.

4.4. Perubahan Nilai Sampling

1. Coba panggil kembali file hasil perekaman dengan cara sebagai berikut.

```
clear all;  
Fs = 8000;  
y=wavread('aiueo.wav')  
wavplay(y,Fs)
```

2. Rubah nilai frekuensi *sampling* menjadi $F_s=10000$, perhatikan apa yang terjadi. Lakukan hal yang sama, dengan nilai $F_s = 14000, 16000, 24000, 44000$ Hz.
3. Coba rubah dengan menurunkan nilai *sampling* $F_s=7000, 6000$, dan 5000 . Amati kejadian apa yang dapat diamati pada sinyal suara tsb.

5. ANALISA DATA DAN TUGAS

1. Untuk grafik “*aiueo.wav*”, coba beri tanda mana yang termasuk kategori *silence*, *unvoice*, dan *voice*.
2. Dari hasil percobaan yang telah dilakukan, coba buat catatan tentang durasi perekaman masing-masing vokal yang telah dilakukan.
3. Berikan catatan nilai magnitudo tertinggi pada masing-masing vokal yang telah direkam.
4. Berikan analisa terhadap perubahan nilai *Sampling Rate* terhadap suara yang dihasilkan.