

RANCANG BANGUNG PERANGKAT LUNAK UNTUK PEMBELAJARAN MEMBACA AL QUR'AN MENGGUNAKAN PENGGABUNGAN SUKU KATA

Kholid Fathoni¹, Nur Rosyid M², Faza Syarof²
Mahasiswa Jurusan Teknologi Informasi¹, Dosen Pembimbing²
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
Email : kholid@eepis-its.edu

Makalah Proyek Akhir

ABSTRAK

This final project is a project to make a software that can read arabic text with the tajwid / rule of reading Al Quran. The steps to read arabic text are using several steps i.e. reading ascii code, converting arabic text to latin text, text filtering that adopted from the real condition in reading Al Qur'an, this filtering make easy about identifying the syllables and finding of reading so that the sound output of this program equals to arabic text input.

This final project is using easy method but difficult in improvisation due to the complexity of grammar and speech of arabic language. This improvisation is how to symbolize arabic text to latin text so that the reading of arabic text is much easier but this symbolization not change arabic text structure, erase useless character, change specific character to completing the arabic text reading and more improvement that can not describe one by one due to complexity of reading arabic text.

Kata kunci : ASCII code, converting arabic text, syllables

1. PENDAHULUAN

1.1 Latar Belakang

Dengan berkembangnya teknologi kecanggihan teknologi sekarang membuat segala hal dapat dikerjakan dengan mudah, teknologi komputer adalah bagian dari kecanggihan teknologi yang ada sekarang. Dengan komputer, orang dapat melakukan banyak hal seperti memainkan musik, melihat film, komunikasi antar tempat melalui jaringan dan lain-lain.

Sehubungan dengan hal diatas saya bermaksud membuat sesuatu yang bermanfaat melalui proyek akhir yang akan saya kerjakan. Pembuatan software yang dapat membantu seseorang dalam mengenal huruf arab dan bagaimana cara membacanya adalah hal yang akan saya kerjakan dalam proyek akhir ini.

Diharapkan dengan adanya software yang dapat membaca teks arab ini maka orang akan lebih mudah mempelajari teks arab beserta cara membacanya, intensitas belajar mereka juga dapat bertambah, disamping mereka belajar ke ustadz-ustadz mereka juga dapat menggunakan software ini untuk belajar

1.2 Rumusan Permasalahan

Adapun permasalahan pada pembuatan proyek akhir ini adalah:

- Bagaimana software dapat membaca teks arab kemudian melakukan simbolisasi ke teks latin
- Bagaimana teks latin yang dihasilkan dapat dimengerti oleh software
- Bagaimana menemukan bacaan/tajwid dari teks
- Bagaimana memisah-misah teks menjadi beberapa suku kata
- Bagaimana menentukan file audio yang sesuai dengan bunyi masing-masing suku kata
- Bagaimana menyamakan volume audio satu dengan yang lain
- Bagaimana membuat transisi load suara audio satu dengan audio berikutnya lebih halus sehingga tidak dihasilkan suara putus-putus

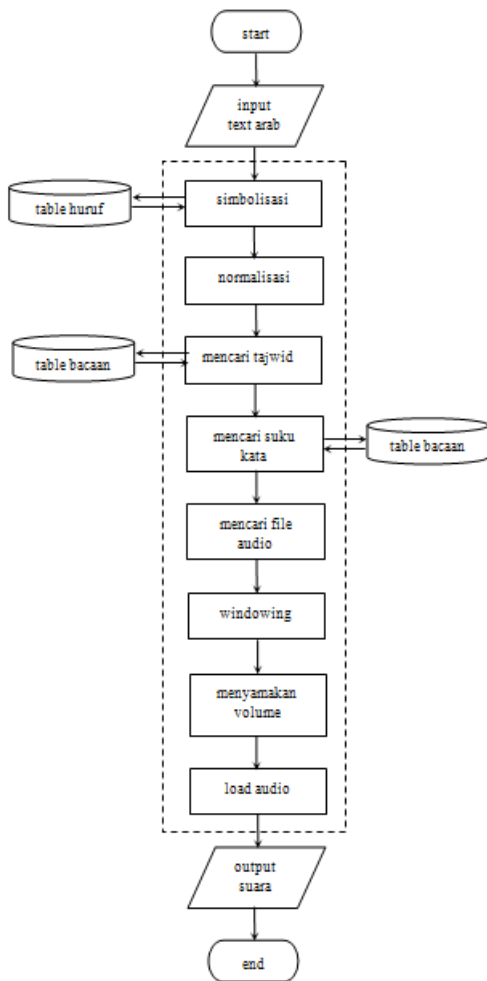
1.3 Batasan Permasalahan

Batasan masalah dalam proyek akhir ini adalah :

- Input yang dimasukkan adalah standart bacaan arab, artinya tidak dimodifikasi sedemikian rupa sehingga sulit dimengerti.

- Input pada saat penekanan tombol shift pada keyboard hanya sebatas harokat untuk teks arab, berupa fathah, kasroh, dhommah, dan sukun, symbol-simbol lain tidak diizinkan.
- Input yang dimasukkan memenuhi standart bacaan IDZHAR, IDZGHOM BILAGHUNNAH, IDZGHOM BIGHUNNAH, IQLAB, IKHFA', IDZGHOM MITSILI, IKHFA' SYAFAWI, IDZHAR SYAFAWI, MAD THOB'I, MAD LEN, GHUNNAH, IDZGHOM MUTAMATSILAIN, QOLQOLAH, LAM JALALAH, AL QOMARIYAH, AL SYAMSIYAH, dan PENGECUALIAN DALAM IDGHOM BIGHUNNAH.
- Input yang dimasukkan bukan termasuk bacaan MAD LAZIM dan MAD ARID LISSUKUN
- Output berupa suara yang diload dari database

2. PERANCANGAN SISTEM



Gambar 2.1 flowchart sistem

HARDWARE PENDUKUNG

1. 1 komputer dengan I sound card
2. Sound system

SOFTWARE APLIKASI PENDUKUNG

Aplikasi-aplikasi yang dijadikan dasar pembuatan sistem adalah :

1. Jdk 6 sebagai JVM
2. Jcreator 4.50 / NetBeans 5.5
3. Wampp server 2.0 untuk MySQL server
4. mysql-connector-java-5.0.5-bin.jar
5. jmusic.jar

PEMBUATAN DATABASE

Database yang digunakan dalam perangkat lunak ini adalah MySQL dengan nama database *atts*. Di dalam database *atts* terdapat dua table yaitu *bacaan* dan *huruf*. Gambar dibawah menunjukkan bentuk table *bacaan* dan *huruf*.

Database	Table	Field	Value
atts2 (2)	bacaan	xa	xa.wav
atts2 (2)	bacaan	ba	ba.wav
atts2 (2)	bacaan	ta	ta.wav
atts2 (2)	bacaan	tba	tba.wav
atts2 (2)	bacaan	ja	ja.wav
atts2 (2)	bacaan	cha	cha.wav
atts2 (2)	bacaan	kha	kha.wav
atts2 (2)	bacaan	da	da.wav
atts2 (2)	bacaan	dta	dta.wav
atts2 (2)	bacaan	ra	ra.wav
atts2 (2)	bacaan	za	za.wav
atts2 (2)	bacaan	sa	sa.wav
atts2 (2)	bacaan	sya	sya.wav
atts2 (2)	bacaan	sha	sha.wav
atts2 (2)	bacaan	dla	dla.wav
atts2 (2)	bacaan	tha	tha.wav
atts2 (2)	bacaan	dha	dha.wav
atts2 (2)	bacaan	la	la.wav
atts2 (2)	bacaan	gha	gha.wav
atts2 (2)	bacaan	fa	fa.wav

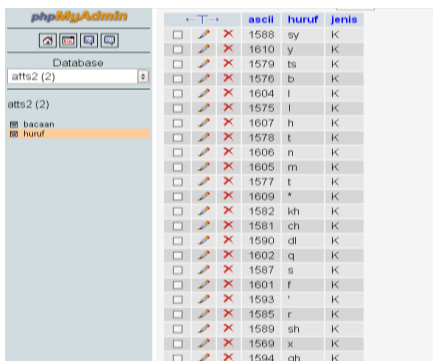
Gambar 2.2 Table bacaan dalam database atts

Table *bacaan* terdiri dari **2889 data** yang merupakan kumpulan dari seluruh kemungkinan suku kata yang terjadi sesuai dengan batasan masalah pada proyek ini, data tersebut diperoleh dari :

- **84 data** diperoleh dari kemungkinan **semua huruf** arab saat **berharokat fathah, kasroh, dan dlommah** → 28x3 huruf
- **28 data** diperoleh dari kemungkinan saat semua huruf arab yang berharokat **fathah bertemu alif** → 28x1 huruf
- **2352 data** diperoleh dari kemungkinan saat **semua huruf** arab yang **berharokat fathah, kasroh, dan dlommah bertemu dengan 28 huruf** (termasuk dirinya sendiri) → 28x28 huruf
- **84 data** diperoleh dari kemungkiina saat **semua huruf** yang **berharokat fathah, kasroh, dan dlommah** dibaca **mendengung mengerupai suara 'ng'**, bacaan ini terjadi saat ikhfa' → 28x3 huruf
- **84 data** diperoleh dari kemungkiina saat **semua huruf** yang **berharokat fathah, kasroh, dan**

dlommah dibaca **mendengung mengerupai suara nun mati yang dibaca panjang**, bacaan ini terjadi saat ikhfa' dan idghom bighunnah → 28x3 huruf

- **84** data diperoleh dari kemungkina saat **semua huruf** yang **berharokat fathah, kasroh, dan dlommah** dibaca **mendengung mengerupai suara yak mati yang dibaca panjang**, bacaan ini terjadi saat idghom bighunnah → 28x3 huruf
- **84** data diperoleh dari kemungkina saat **semua huruf** yang **berharokat fathah, kasroh, dan dlommah** dibaca **mendengung mengerupai suara mim mati yang dibaca panjang**, bacaan ini terjadi saat idghom bighunnah → 28x3 huruf
- **84** data diperoleh dari kemungkina saat **semua huruf** yang **berharokat fathah, kasroh, dan dlommah** dibaca **mendengung mengerupai suara wawu mati yang dibaca panjang**, bacaan ini terjadi saat idghom bighunnah → 28x3 huruf



ascii	huruf	jenis
1588	sy	K
1610	y	K
1579	ts	K
1576	b	K
1604	l	K
1575	l	K
1607	h	K
1578	t	K
1606	n	K
1605	m	K
1577	t	K
1609	*	K
1582	kh	K
1581	ch	K
1590	di	K
1602	q	K
1587	s	K
1601	f	K
1593	'	K
1585	r	K
1589	sh	K
1569	x	K
1594	ah	K

Gambar 2.3 Table huruf dalam database atts

Table *huruf* menyimpan kode ascii dari semua huruf arab serta jenis huruf tersebut (konsonan / K dan vocal / V)

SIMBOLISASI TEKS ARAB KE TEKS LATIN

Input text arab akan dibaca per-karakter kemudian dicari kode ascii-nya, selanjutnya kode ascii tersebut dicocokkan dengan table *huruf* pada database *atts*, hasil pencocokan adalah berupa penggantian huruf arab dengan huruf latin (simbolisasi).

```
try{
    Class.forName("org.gjt.mm.mysql.Driver");
    Connection
    con=DriverManager.getConnection("jdbc:mysql://localhost/atts2","root","");
    Statement st1=con.createStatement();
    ResultSet rs1=st1.executeQuery("SELECT
* FROM huruf");
    int l=txtArab.getText().length();
    int count=0;
    String kata1="";
```

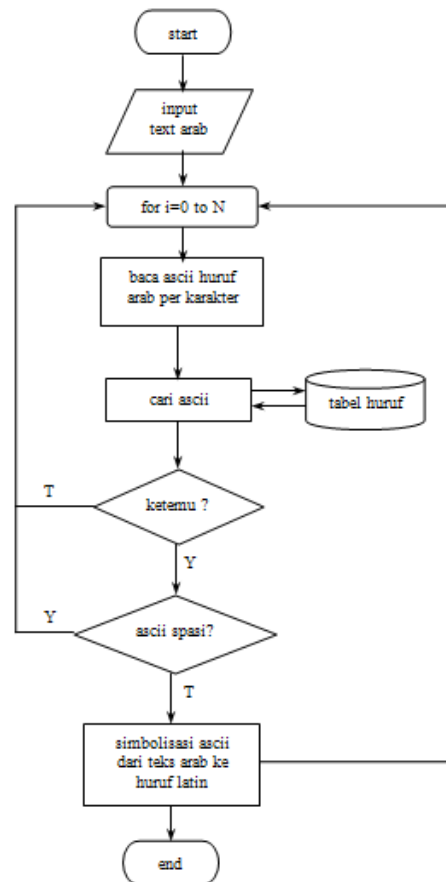
```
String huruf[]=new String[1000];
String jenis[]=new String[1000];

//menemukan huruf
for(int i=0; i<l; i++){
    int
    ascii=(int)txtArab.getText().charAt(i);
    rs1.beforeFirst();
    while(rs1.next()){

if(ascii==Integer.parseInt(rs1.getString("ascii")))
    break;
    }
    if(!rs1.getString("huruf").equals("-")){

huruf[count]=rs1.getString("huruf");
jenis[count]=rs1.getString("jenis");
kata1=kata1+huruf[count];
count++;
    }
}
//CETAK DI TEXTBOX PERTAMA
txtLatin.setText(kata1);
.....
}catch(Exception e){
}
```

Gambar 2.4 Program untuk simbolisasi teks arab ke teks latin



Gambar 2.5 flowchart proses simbolisasi teks arab ke teks latin

NORMALISASI TEKS

Teks latin yang dihasilkan dari poin diatas selanjutnya dinormalisasi, yaitu proses menghilangkan atau mengganti karakter-karakter tertentu dalam teks latin hasil simbolisasi, hal ini dimaksudkan agar teks latin dapat dimengerti / dibaca oleh program.

Contoh untuk kondisi dimana terdapat siddah (disimbolkan dengan '2') dalam teks :

```
if(huruf[i].equals("2")){
    try{
        //kondisi jika ada huruf bersukun
        bertemu dg huruf yg sama
        // tetapi bersiddah maka siddah
        dihilangkan
        if(huruf[i-1].equals(huruf[i-3]) &&
        huruf[i-2].equals("0"))
            huruf[i]="";
        //kondisi jika ada huruf bertanwin
        bertemu dengan huruf bersiddah
        //maka siddah dibuang, terdapat pd
        bacaan idghom dan ikhfa'
        else if(huruf[i-2].equals("an0") ||
        huruf[i-3].equals("an0") || huruf[i-
        2].equals("in0") || huruf[i-2].equals("un0"))
            huruf[i]="";
        //siddah diganti dengan
        sukun+huruf sebelum siddah
        else
            huruf[i]="0"+huruf[i-1];
        }catch(Exception e){
        }
    }
```

Gambar 2.6 Contoh program untuk normalisasi teks

PENCARIAN TAJWID

Setelah teks dinormalisasi maka selanjutnya adalah pencarian bacaan. Bacaan-bacaan yang dimaksud disini adalah sebagai berikut :

- IDZHAR, artinya : jelas atau tampak, suara ن َّ َّ َّ َّ dibaca apa adanya saat bertemu huruf ء ه ح خ غ contoh : مِنْهُمْ
- IDZGHOM BILAGHUNNAH, artinya : masuk tanpa dengung, suara ن َّ َّ َّ َّ masuk pd ل ر tanpa dengung, contoh : مِنْ رَبِّهِمْ
- IDZGHOM BIGHUNNAH, artinya : masuk dengan dengung, suara ن َّ َّ َّ َّ masuk pd و م ن ي dengan dengung, contoh : مَنْ يَعْمَلُ

- IQLAB, artinya : membalik atau merubah, suara ن َّ َّ َّ َّ berubah menjadi mim saat bertemu huruf ب contoh : مِنْ بَعْدِ
- IKHFA', artinya : samar, suara ن َّ َّ َّ َّ َّ َّ dibaca berdengung saat bertemu huruf 15 selain huruf-huruf yang terdapat pd IDZHAR, IDZGHOM dan IQLAB, contoh : مِنْ فَضْلِ
- IDZGHOM MITSLI, artinya : masuk pada huruf yang sama, suara م َّ masuk pd م contoh : لَهُمْ مَثَلٌ
- IKHFA' SYAFAWI, artinya : samar pd bibir, suara م َّ dibaca samar dan berdengung ketika bertemu ب contoh : إِنَّهُمْ بِذَالِكَ
- IDZHAR SYAFAWI, artinya : jelas pada bibir, suara م َّ dibaca apa adanya ketika bertemu huruf selain م dan ب contoh : أَمْ لَمْ
- MAD THOBI'I, bacaan panjang ketika fathah bertemu ا kasroh bertemu ي dlommah bertemu و contoh : ذَالِكَ
- MAD LEN, bacaan pendek ketika fathah tidak bertemu ا kasroh tidak bertemu ي dlommah tidak bertemu و contoh : فَوْقَ
- GHUNNAH, semua َ dan ِ dibaca dengung, contoh : إِنَّ
- IDZGHOM MUTAMATSILAIN, semua huruf yang bersukun bertemu dengan dirinya sendiri, contoh : يُسْرِفُ فِي

- QOLQOLAH, artinya : bacaan ق ط
د ج ب yang bersukun dibaca seakan-akan menyangkut ditenggorokan, contoh : أَقَوْمٌ
- LAM JALALAH, artinya : bacaan lam yang terdapat pada lafadl اَللّٰه
- AL QOMARIYAH, artinya : AL yang tetap terbaca AL ketika bertemu 14 huruf ب ج ح خ ع غ ف ق ك م و ه ء ي
- AL SYAMSYIYAH, artinya : AL yang tidak terbaca AL ketika bertemu 14 huruf ت ث د ذ ر ز س ش ص ض ط ظ ل ن
- PENGECUALIAN DALAM IDGHOM BIGHUNNAH, artinya : bacaan ن ّ ّ ّ yang bertemu و dan ي dalam satu ّ kata wajib dibaca terang, contoh : بُنْيَانٌ

Contoh untuk bacaan Idghom bilaghunnah, idghom bighunnah, dan ikhfa' :

```
//-----IDZGHOM BILAGHUNNAH-----
text=text.replaceAll("n0l","l0l");
//nun bertemu lam
text=text.replaceAll("n0r","r0r");
//nun bertemu ro'
//-----
//-----IDZGHOM BIGHUNNAH-----
text=text.replaceAll("n0y","y#y");
//nun bertemu ya'
text=text.replaceAll("n0n","n#n");
//nun bertemu nun
text=text.replaceAll("n0m","m#m");
//nun bertemu mim
text=text.replaceAll("n0w","w#w");
//nun bertemu wawu
//-----
//-----IKHFA'-----
text=text.replaceAll("n0ts","n#ts");
//nun bertemu tsa'
text=text.replaceAll("n0j","n#j");
//nun bertemu jim
text=text.replaceAll("n0dz","n#dz");
//nun bertemu dzal
text=text.replaceAll("n0z","n#z");
//nun bertemu zain
```

```
text=text.replaceAll("n0sy","ngsy");
//nun bertemu syin
text=text.replaceAll("n0sh","ngsh");
//nun bertemu shod
text=text.replaceAll("n0s","n#s");
//nun bertemu sin
text=text.replaceAll("n0dl","ngdl");
//nun bertemu dlo'
text=text.replaceAll("n0th","ngth");
//nun bertemu tho'
text=text.replaceAll("n0t","n#t");
//nun bertemu ta'
text=text.replaceAll("n0dh","ngdh");
//nun bertemu dho'
text=text.replaceAll("n0d","n#d");
//nun bertemu dal
text=text.replaceAll("n0f","ngf");
//nun bertemu fa'
text=text.replaceAll("n0q","ngq");
//nun bertemu qof
text=text.replaceAll("n0k","ngk");
//nun bertemu kaf
//-----
```

Gambar 2.7 Contoh program untuk pencarian bacaan

Listing program diatas adalah mengganti simbol nun sukun/tanwin dengan simbol yang sesuai dengan bacaan yang benar, misalnya simbol n# berarti bacaan nun dengung dst

PENCARIAN SUKU KATA

Setelah input teks sudah difilter dan dicari bacaan/tajwid didalamnya maka selanjutnya dicari suku kata-suku kata yang ada dalam teks (d disesuaikan dengan teks arab). Penentuan suku kata ini disesuaikan dengan data yang ditemukan dalam table *bacaan* dalam database *atts2*. Adapun suku kata dalam teks arab dibagi menjadi 2 jenis yaitu suku kata tertutup dan suku kata terbuka, suku kata tertutup mempunyai pola KVK, misalnya 'qab', 'zab', 'kaf', dll. Sedangkan suku kata terbuka mempunyai pola KV, misalnya 'fa', 'za' dll, dan KVV, misalnya 'faa', 'zaa' dll

```
String suku_kata[]=new String[1000];
int count=0;
try{
    Class.forName("org.gjt.mm.mysql.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/atts2","root","");
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("SELECT * FROM
bacaan");

    int i=0, j=kata.length();
    int ketemu=0;
    txtBacaan.setText("");
    while(j>i){
        String temp_suku=kata.substring(i,j);

        rs.beforeFirst();
        while(rs.next()){

            if(rs.getString("bacaan").equals(temp_suku)){

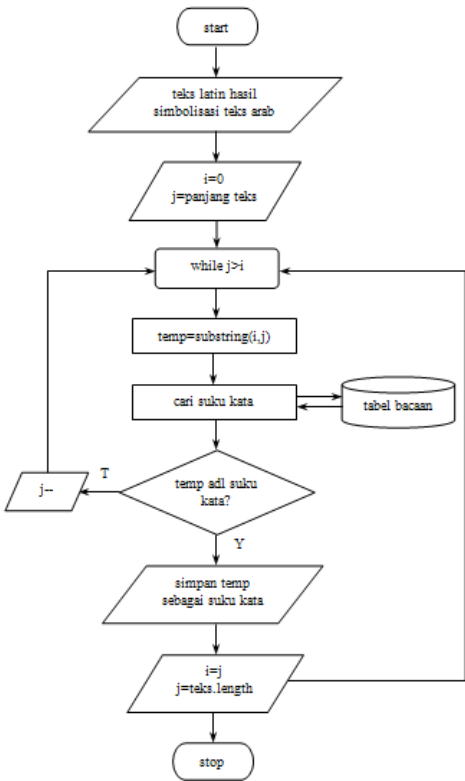
                System.out.println(i+" "+j+" "+temp_suku);
```

```

suku_kata[count]=rs.getString("suara");
count++;
ketemu=1;
break;
}
}
rs.last();
if(ketemu==0)
j--;
else{
i=j;
j=kata.length();
ketemu=0;
}
}
}

```

Gambar 2.8 Program untuk pencarian suku kata



Gambar 2.9 flowchart proses pencarian suku kata

PENCARIAN FILE AUDIO YANG SESUAI

Dari point diatas sebenarnya sudah didapatkan file wave mana saja yang harus di-load

```

.....
suku_kata[count]=rs.getString("suara");
.....

```

Namun didalam program/software ini terdapat dua pilihan suara yaitu *Faza Syarof* dan *Khusnul Khotimah* karena itu terdapat dua folder yang menyimpan file wave, untuk menentukan folder mana yang dipilih adalah dengan memilih pilihan dalam *combobox* di tampilan program

kemudian pengaruh dari pemilihan tersebut adalah seperti dibawah ini

```

.....
String pilih="";
if(pilSuara.getSelectedltem().equals("Faza Syarof"))
pilih="Faza/";
else
pilih="Ima/";
.....

```

WINDOWING

Proses windowing ini berfungsi untuk mengubah gelombang sinyal audio pada bagian ujung file audio menjadi mengerucut, hal ini dimaksudkan agar transisi saat load suara anta suara satu dengan yang lain terdengar lebih smooth



Gambar 2.10 Perubahan gelombang audio saat dilakukan windowing

```

import jm.util.*;

public class windowing{
    public windowing(String au){
        float data[] = Read.audio(au);
        System.out.println(au);
        float window[] = new float[data.length];
        for(int i=0; i<data.length; i++){
            float x=(float) (2*3.14*(i-1)/(data.length-1));
            window[i]=(float) (data[i]*(0.54-0.46*Math.cos(x)));
        }
        Write.audio(window, "temp/"+au);
    }
}

```

Gambar 2.11 Program untuk windowing

proses dimulai dari sampling file audio $data[]=Read.audio(au)$ proses ini mengkonversi data sinyal menjadi digital, data digital selanjutnya dimasukkan dalam rumus windowing dengan metode hamming yaitu :

$h_d[n]=w[n]h[n]$
dimana

$$w[n] = 0.541 - 0.46 \cos \frac{2\pi n}{N-1} \quad 0 \leq n \leq N-1$$

- N : jumlah data
- $h_d[n]$: data baru ke-n
- $h[n]$: data ke-n
- $w[n]$: windowing ke-n

hasil windowing adalah berupa data digital baru yang belum dapat di-load sebagai file audio, maka agar data tersebut dapat di-load perlu dilakukan pengembalian dari data digital

ke data sinyal dengan perintah `Write.audio(window, "temp/"+au)`

MENYAMAKAN VOLUME

Sebelum file audio yang sudah ditemukan pada proses diatas akan di-load, terlebih dahulu dilakukan proses menyamakan volume antara file audio satu dengan file audio yang lain, hal ini bertujuan agar aoutput akhir suara yang dihasilkan tidak terkesan naik turun.

Perbedaan volume diantara file-file audio ini disebabkan oleh proses perekaman yang kurang sempurna, kondisi fisik, emosi dan daya pada saat perekaman oleh pengisi suara sangat mempengaruhi perbedaan ini, selain itu faktor perbedaan waktu saat perekaman juga mempengaruhi tinggi rendahnya volume masing-masing file audio yang dihasilkan.

Permasalahan perbedaan volume ini sedikit di-minimalis dengan adanya proses menyamakan volume. Hal pertama yang dilakukan sebelum menyamakan volume adalah dengan mendeteksi perbedaan volume antara file audio satu dengan yang lain, kemudian dicari nilai-nilai perbedaan yang ada. Berikut ini adalah class `volControl` yang berfungsi untuk mendeteksi perbedaan volume :

```
import jm.util.Read;
import jm.util.Write;

public class volume {
    int d1,d2;
    float avg1, avg2;
    float nilai_atas[]=new float[1000];
    float nilai_bawah[]=new float[1000];
    public volume(String au, int count) {
        float data[]=Read.audio(au);
        for(int i=0; i<data.length; i++){
            if(data[i]>0.01){
                avg1=avg1+data[i];
                d1++;
            }
        }
        nilai_atas[count]=avg1/d1;
        for(int j=0; j<data.length; j++){
            if(data[j]<-0.01){
                avg2=avg2+data[j];
                d2++;
            }
        }
        nilai_bawah[count]=avg2/d2;
    }
}
```

Gambar 2.12 Program untuk mendeteksi perbedaan volume

Setelah perbedaan ditemukan dan didapatkan nilai-nilai perbedaan maka selanjutnya akan dilakukan penyamaan volume. Berikut adalah potongan program untuk menyamakan volume :

```
String au[]=new String[count];
float signal[]=new float[count];
float min=1000;
String nmFile="";
double setVol[]=new double[count];

for(int x=0; x<count; x++){
    au[x]="temp/"+pilih+suku_kata[x];
}

for(int x=0; x<count; x++){
    volume v=new volume(au[x],x);
    signal[x]=v.nilai_atas[x]-
    v.nilai_bawah[x];
    if(min>signal[x]){
        min=signal[x];
        nmFile=suku_kata[x];
    }
}

for(int x=0; x<count; x++){
    setVol[x]=min/signal[x];
    System.out.println(setVol[x]);
}
```

Gambar 2.13 Program untuk menyamakan volume

Kode program diatas menghasilkan nilai volume yang sesuai untuk masing-masing file audio, nilai tersebut kemudian dikirim ke proses load suara.

LOAD SUARA

Pada java, apabila kita ingin me-load suara lebih dari satu file audio maka kita perlu menggunakan delay yang sesuai dengan panjang/durasi file audio yang akan diload, jika tidak maka semua file audio akan diload secara bersamaan sehingga menghasilkan suara yang saling menimpa (*overlap*) dan tidak teratur.

Dibawah ini adalah program untuk mengetahui durasi audio file

```
.....
framesize = format.getFrameSize();
framerate = (int) format.getSampleRate();
l = (float)f.length();
msec = 1 / (framesize * framerate) *
1000;
.....
```

Gambar 2.14 Program untuk mendapatkan durasi audio

Sebelum dilakukan load suara maka akan dicari dulu nilai volume untuk load setiap suara / file audio, seperti yang telah dijelaskan diatas. Selanjutnya setelah durasi dan nilai volume untuk masing-masing suara / file audio ditemukan maka barulah suara di-load. Berikut potongan program untuk load suara :

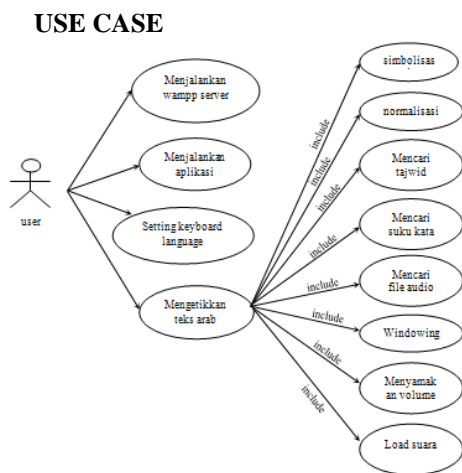
```

//load suara-----
-----
for(int x=0; x<count; x++){
    new
    sound("Temp/suara"+(x+1)+".wav",setVol[x]);

    System.out.println("Temp/suara"+(x+1)+".
    wav : "+msec[x]);
    try{
        Thread.sleep(msec[x]-
msec[x]/3);
    }catch(Exception e){}
}
//-----

```

Gambar 2.15 Program untuk load suara



Gambar 2.16 Use Case

3. UJI COBA DAN ANALISA

Berikut ini beberapa hasil percobaan dalam pembacaan masing-masing bacaan :

Teks arab	Bacaan	Hasil
مِنْهُمْ	Idhar	Terbaca
مِنْ رَبِّهِمْ	Idghom bilaghunnah	Terbaca
مَنْ يَعْمَلْ	Idghom bighunnah	Terbaca
مِنْ بَعْدِ	Iqlab	Terbaca
مِنْ فَصْلِ	Ikhfa'	Terbaca
لَهُمْ مَثَلٌ	Idghom mitsli	Terbaca
إِنَّهُمْ بِذَلِكَ	Ikhfa' syafawi	Terbaca
أَمْ لَمْ	Idhar syafawi	Terbaca

ذَالِكَ	Mad thobi'i	Terbaca
فَوْقَ	Mad len	Terbaca
إِنَّ	Ghunnah	Terbaca
يُسْرِفُ فِي	Idghom mutamatsilain	Terbaca
أَقْوَمُ	Qolqolah	Terbaca
اللَّهِ	Lam jalalah	Terbaca
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ	Lam jalalah, al syamsyiyah, mad thobi'i	Terbaca
الْقَارِءَ مَا الْقَارِءَ	Al qomariyah	Terbaca
بُنْيَانٌ	Pengecualian dalam idghom bighunnah	Terbaca
لِيَحَاجُّكُمْ	Mad lazim	Tidak terbaca
وَلَا الضَّالِّينَ	Mad lazim	Tidak terbaca
بَعِيدٌ	Mad arid lissukun	Tidak terbaca
شُكُورٌ	Mad arid lissukun	Tidak terbaca

Tabel 3.1 Percobaan pembacaan teks arab

Alasan mengapa bacaan mad lazim dan mad arid lissukun tidak bisa dibaca adalah karena untuk dua bacaan ini tidak dapat didefinisikan suku katanya, seperti yang telah dijelaskan oleh wikipedia.com bahwa ada dua jenis suku kata dalam bahasa arab yaitu suku kata tertutup dan suku kata terbuka.

- Suku kata tertutup mempunyai pola

KVK, contoh : مِنْ هَلْ

Jika dua teks arab tersebut didefinisikan dalam ejaan latin maka menjadi MINDan HAL, dua suku kata ini berpola KVK

- Suku kata tertutup mempunyai dua pola yaitu KV dan KVV,

contoh : قُوْ نَا ثُ وَ, jika keempat text tersebut didefinisikan dalam ejaan latin

maka menjadi WA, TU mempunyai pola KV, dan NAA, QUU mempunyai pola KVV

Sedangkan untuk bacaan mad lazim dan mad arid lissukun, sulit untuk didefinisikan suku katanya, contoh :

- Mad lazim : **لِيَحَاجُّكُمْ, وَلَا الضَّالِّينَ** jika didefinisikan dalam ejaan latin maka menjadi WA LADLAALLIINA dan LIYUCHAAJJUUKUM, bagian yang bercetak tebal adalah yang dimaksud dengan bacaan mad lazim, jika dilihat polanya adalah KVVK (huruf CH dan DL dianggap satu huruf karena mewakili ح dan ض), format KVVK tidak sesuai dengan standart suku kata bahasa arab seperti yang dijelaskan *wikipedia.com*
- Mad arid lissukun : **شَكُورٌ, بَعِيدٌ** jika didefinisikan dalam ejaan latin maka menjadi SYAKUUR dan BA'IID, bagian yang bercetak tebal adalah yang dimaksud dengan bacaan mad arid lissukun, jika dilihat polanya adalah KVVK, format KVVK tidak sesuai dengan standart suku kata bahasa arab seperti yang dijelaskan *wikipedia.com*

Kesimpulan

Setelah melaksanakan pengujian dan analisa pada sistem rancang bangun perangkat lunak untuk pembelajaran membaca Al Qur'an, maka dapat diambil beberapa kesimpulan sebagai berikut :

- Hasil load suara dapat lebih maksimal jika teks arab yang diketikkan mengandung fathah yang sama, misalnya : baqara, qaraxa, hilmi, tsulutsu dsb, hal ini karena proses perekaman yang berbeda dalam hal waktu, kondisi, dan emosi pengisi suara dalam hal perekaman harokat yang berbeda, waktu yang berbeda ini adalah karena faktor sangat banyaknya suara yang disimpan.
- Dari hasil pengujian, sistem mampu membaca semua bacaan yang sudah ditentukan dalam batasan masalah seperti IDZHAR, IDZGHOM BILAGHUNNAH, IDZGHOM BIGHUNNAH, IQLAB, IKHFA', IDZGHOM MITSLI, IKHFA' SYAFAWI, IDZHAR SYAFAWI, MAD THOBI'I, MAD LEN, GHUNNAH, IDZGHOM MUTAMATSILAIN, QOLQOLAH, LAM JALALAH, AL QOMARIYAH, AL SYAMSYIYAH,

dan PENGECEUALIAN DALAM IDGHOM BIGHUNNAH.

Sistem mampu membaca teks arab yang cukup panjang dan mengandung beberapa macam jenis bacaan yang telah disebutkan diatas.

Saran

Tugas akhir ini masih terdapat kekurangan yang masih bisa diperbaiki pada proyek akhir selanjutnya di tahun yang akan datang. Adapun beberapa kekurangan yang perlu diperbaiki pada tugas akhir yang akan datang adalah :

1. Output suara yang dihasilkan kurang maksimal karena keterbatasan alat yang digunakan saat proses perekaman sehingga suara yang dihasilkan terdapat banyak noise, kendala waktu, kondisi, dan emosi yang berbeda juga menjadi faktor yang mempengaruhi hasil perekaman suara.
2. Diharapkan untuk pembuatan perangkat lunak sejenis diwaktu yang akan datang, alat yang digunakan adalah alat perekam khusus yang mempunyai kualitas perekaman yang baik dan harus terpisah dengan komputer, karena apabila kita menggunakan headset yang terdapat microphone atau alat-alat lain yang terhubung secara langsung ke komputer, seperti yang digunakan dalam proyek akhir ini, akan terdapat banyak noise yang dihasilkan.
3. Selain alat perekam yang baik, penyamaan pitch untuk setiap audio file harus dilakukan, dalam proyek akhir ini kami melakukannya secara manual menggunakan tool audacity, namun ternyata hasilnya kurang maksimal, proses penyamaan pitch yang bagus adalah dengan mengirimkan file audio ke Diphone Processor di Belgia. (pemrosesan memerlukan biaya proses 2 Euro/diphone).
4. Diharapkan selanjutnya dapat dikembangkan untuk pembacaan Al-Qur'an digital serta untuk fasilitas pembacaan text arab lain yang terintegrasikan dengan OS Windows atau Linux yang mendukung versi arab.
5. Diharapkan selanjutnya sistem dapat berjalan sebagai background, seperti halnya 'narrator' yang terdapat di Windows.

5. DAFTAR PUSTAKA

- [1] <http://www.didien.net/v2r1/>
- [2] <http://www.wikipedia.com>
- [3] Tri Budi Santoso, Hary Octavianto, Titon Dutono , “Windowing dan Pengamatan Spectrum Frekuensi ”
- [4]. <http://www.developer.com/java>