BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penerjemahan perancangan berdasarkan hasil analisis ke dalam suatu bahasa pemrograman tertentu serta penerapan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Adapun pembahasan implementasi terdiri dari kebutuhan mimimum perangkat lunak , kebutuhan minimum perangkat keras dan implementasi antarmuka.

4.1.1 Kebutuhan Minimum Perangkat Lunak

Perangkat lunak yang digunakan dalam mengimplementasikan metode pitch detection untuk pencarian lirik lagu melalui media suara adalah sebagai berikut :

Tabel 4.1 Kebutuhan Minimum Perangkat Lunak

No	Perangkat Lunak
1	Sistem Operasi *Windows 7*
2	*Eclipse AdtBundle*
3	Sistem Operasi Android 4.1 Ice Cream Sendwich

4.1.2 Kebutuhan Minimum Perangkat Keras

Kebutuhan minimum perangkat keras yang diperlukan untuk mengimplementasikan program aplikasi yang dibangun adalah sebuah komputer PC atau *notebook* serta telepon seluler. Semakin baik spesifikasi perangkat keras yang membangun komputer yang digunakan, maka semakin baik aplikasi tersebut dapat diproses. Kebutuhan minimum spesifikasi perangkat keras , antara lain :

Tabel 4.2 Kebutuhan Minimum Perangkat Keras Komputer

Komputer	Spesifikasi Minimum
Prosesor	Inter core i3
Layar	Resolusi 1280 x 800
Monitor	LCD 14 inc
Memori	Memori 2GB
Harddisk	Harddisk

Tabel 4.3 Kebutuhan Minimum Perangkat Keras Smartphone

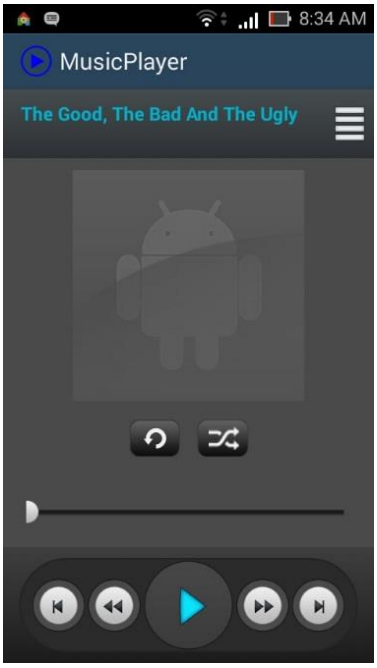
SmartPhone	Spesifikasi Minimum
Prosesor	Android
Layar	4.5 inc
Memori	1 GB

4.2 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan sistem yang dibuat dengan pengkodeannya dalam file program. Pada bagian ini akan ditampilkan implementasi antarmuka menu utama , menu pencarian judul melalui merekam suara, menu tampilan hasil pencarian , menu play musik, menu pause musik dan menu pesan terdapat pada antar muka pesan keluar aplikasi.

1. Implementasi Antarmuka Menu Utama

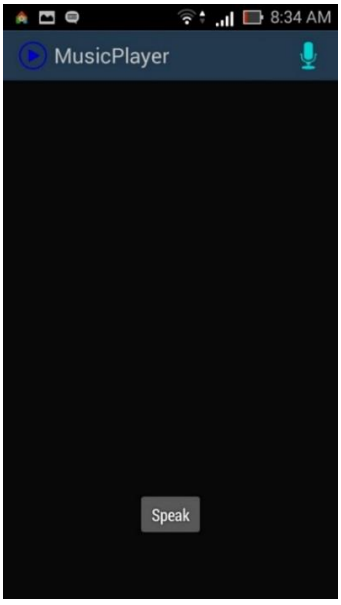
   Berikut adalah Implementasi Antarmuka Menu Utama yang ditunjuk oleh gambar 4.1 di bawah ini .Tombol yang tersedia adalah tombol pilihan yang berfungsi sebagai akses menuju perekaman suara untuk pencarian judul lagu.



Gambar 4.1. Antarmuka Menu Utama

2. Antarmuka menu pencarian judul lagu

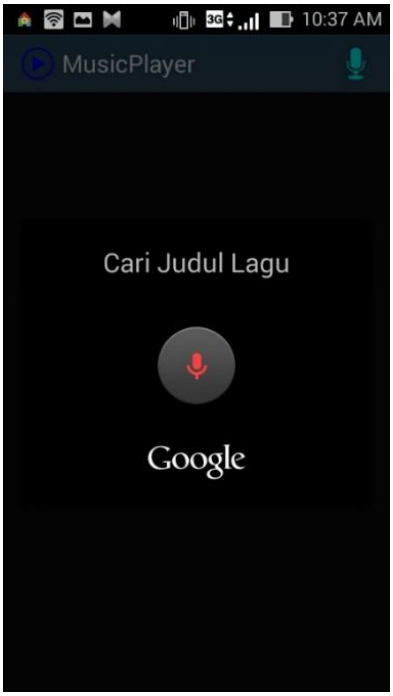
   Antarmuka menu pencarian judul lagu ditunjukkan pada gambar 4.2 dibawah ini. Tombol recognition yang tersedia menunjukan akses untuk proses perekaman suara terjadi.



Gambar 4.2 Antarmuka Pencarian Judul Lagu

3. Antarmuka Menu Rekam Suara

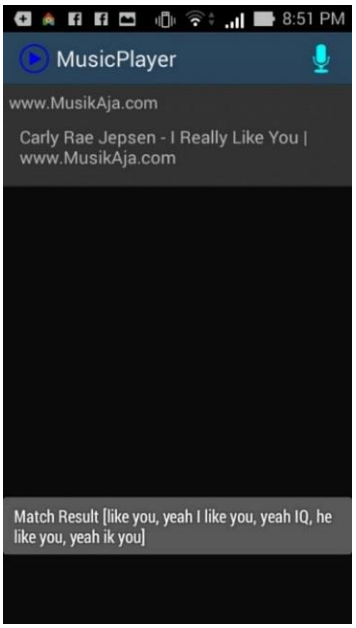
Antarmuka menu rekam suara ditunjukkan pada gambar 4.3. Antarmuka menu rekam berfungsi untuk merekam suara untuk selanjutnya terjadi proses pencarian judul lagu.



Gambar 4.3 Antarmuka Menu Rekam

4. Antarmuka Menu Hasil Pencarian Melalui suara

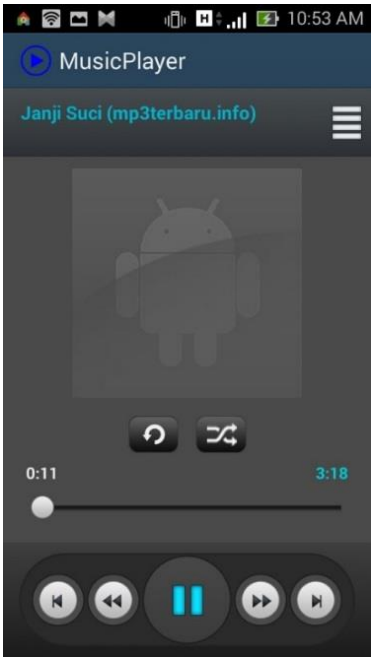
   Antarmuka menu hasil pencarian melalui suara ditunjukan pada gambar 4.4 dibawah ini. Antarmuka menu hasil pencarian memiliki fungsi menampilkan hasil berupa lagu dari masukan suara diawal.



Gambar 4.4 Antarmuka Menu Hasil Cari

5. Antarmuka Menu Play

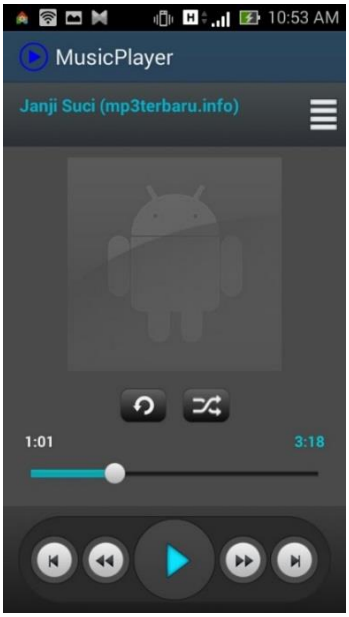
   Antarmuka menu play ditunjukan pada gambar 4.5 dibawah ini. Antarmuka menu play memiliki fungsi memainkan lagu yang telah berhasil keluar melalui hasil pencarian.



Gambar 4.5 Antarmuka Menu Play

6. Antarmuka Menu Pause

Antarmuka menu pause ditunjukan pada gambar 4.6 dibawah ini. Antarmuka menu pause memiliki fungsi berhentikan lagu yang sedang dimainkan melalui hasil pencarian.



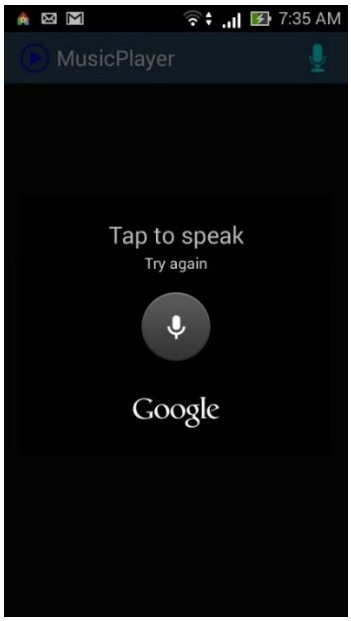
Gambar 4.6 Antarmuka Menu Pause

4.3 Implementasi Pesan

Implementasi pesan adalah bagian yang menunjukan bentuk tampilan setiap pesan pada aplikasi.

1. Pesan Gagal Merekam Suara

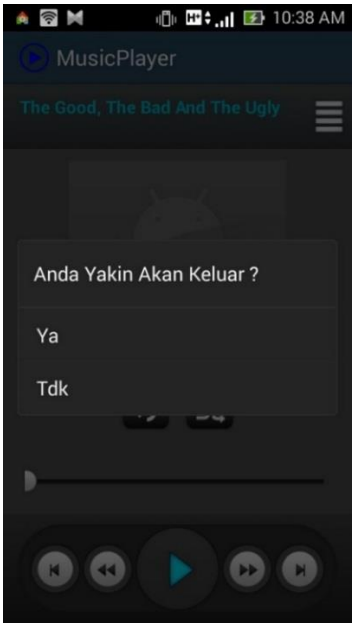
   Pesan gagal merekam suara di tunjukan pada gambar . Pesan gagal berfungsi sebagai pemberitahuan kepada pengguna agar mencoba lagi merekam suara dengan menekan tombol record suara.



Gambar 4.7 Antarmuka Menu pesan gagal merekam suara

2. Pesan Keluar Aplikasi

   Pesan keluar aplikasi di tunjukan pada gambar . Pesan keluar permainan berfungsi sebagai validasi jika pengguna ingin keluar dari menu utama dan kembali ke menu aplikasi.



Gambar 4.8 Antarmuka Menu Pesan Keluar

4.4 Pengujian Sistem

Pengujian sistem dilakukan bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang telah diuji.pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Dalam penelitian ini pengujian yang dilakukan terhadap sistem yaitu pengujian secara fungsional (*alpha).* Metode yang digunakan dalam pengujian ini adalah pengujian *blackbox* yang berfokus pada persyaratan fungsional dari penelitian yang dilakukan dan *whitebox* yang berfokus pada alur aplikasi yang dibangun.

4.4.1 Pengujian Alpha

Pengujian Alpha dilakukan pada sisi pengembangan masalah pemakaian. Pengujian yang dilakukan mengamati hasil pengujian dan memeriksa fungsionalitas dari perangkat lunak.

4.4.2 Skenario Pengujian Alpha

Rencana pengujian adalah pengujian terhadap fungsionalitas yang terdapat dalam aplikasi, apakah fungsionalitas tersebut sesuai yang diharapkan atau tidak. Rencana pengujian dari sistem yang dibangun dapat dilihat pada tabel 4.3.

Tabel 4.3 Skenario Pengujian Alpha

No	Komponen yang di uji	Detail pengujian	Jenis Pengujian
1.	Mulai Aplikasi	Membuka aplikasi pencarian judul lagu	*Black box*
2	Mulai Merekam Pencarian	Menekan Tombol Recognition	*Black box*
3	Play Hasil Pencarian	Memilih Daftar Lagu yang keluar	*Black box*
4	Pause Lagu	Menekan Tombol Pause	*Black box*
5	Previous	Menekan Tombol Previous	*Black box*
6	Fast-forward	Menekan Tombol Fast-Forward	*Black box*
7	Keluar aplikasi	Menekan Tombol keluar	*Black box*

4.4.3 Pegujian *Blackbox*

Pengujian *black box* adalah metode pengujian berfokus pada persyaratan perangkat lunak. Tujuan *black box* adalah menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang, menguji hasil keluaran tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut dan mengetahui kemampuan perangkat lunak dalam memenuhi kebutuhan pengguna. Kasus dan hasil pengujian dapat dilihat pada tabel 4.4.

Tabel 4.4 Kasus dan Hasil Pengujian Black Box

No	Kelas Uji	Detail Pengujian	Hasil yang diharapkan	Hasil pengujian
1.	Menu Utama	Triger menuju menu pencarian	Menampilkan Menu mencari judul lagu	[√] Berhasil [ ] Gagal
		Triger untuk memulai lagu hasil pencarian	Lagu akan dimainkan setelah pencarian berhasil	[√] Berhasil [ ] Gagal
		Triger menjeda lagu yang sedang diputar	Lagu akan berhenti	[√] Berhasil [ ] Gagal
		Triger untuk menurunkan kecepatan lagu yang diputar	Lagu akan perlambat	[√] Berhasil [ ] Gagal
		triger untuk mempercepat lagu yang diputar.	Lagu akan percepat	[√] Berhasil [ ] Gagal
		Triger untuk keluar dari menu utama	Keluar Aplikasi	[√] Berhasil [ ] Gagal
2.	Menu pengenalan	Triger pengenalan suara masukan	Sistem meminta merekam suara	[√] Berhasil [ ] Gagal
3	Menu Hasil Pencarian Lagu	Triger hasil pencarian lagu	Hasil lagu akan keluar sesuai pencarian	[√] Berhasil [ ] Gagal

4.4.4 Pengujian *White Box*

Pengujian *whitebox* merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur control dari desain aplikasi secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Serta untuk menguji alur logika, perulangan, dan validasi parameter masukannya. Di bawah ini merupakan kasus Algoritma Short Time Fourier Transform untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode white-box, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan.

Source Code Algoritma Short Time Fourier Transform

```
  public static void main(String[] args) {
    int N = 8;

1.      STFT fft = new STFT(N);
    double[] window = stft.getWindow();
     double[] re = new double[N];
     double[] im = new double[N]; 2.    // Impulse
     re[0] = 1; im[0] = 0;
     for(int i=1; i<N; i++)
       re[i] = im[i] = 0;
     beforeAfter(stft, re, im);

3.      // nilai
     for(int i=0; i<N; i++) {
       re[i] = Math.pow(-1, i);
      im[i] = 0;
     }
     beforeAfter(stft, re, im);
4.       // cos
     for(int i=0; i<N; i++) {
       re[i] =   Math.cos(2*Math.PI*i / N);
       im[i] = 0;
     }
     beforeAfter(stft, re, im);
 5.      //frame
     for(int i=0; i<N; i++) {
       re[i] = i;
       im[i] = 0;
     }
     beforeAfter(stft, re, im);

6. long time =  System.currentTimeMillis();
     double iter = 30000;
     for(int i=0; i<iter; i++)
       stft.stft(re,im);
     time = System.currentTimeMillis() - time;
     System.out.println("Averaged " + (time/iter) + "ms per
iteration");
   }
 7.   protected static void   beforeAfter(STFT stft, double[] re,
double[] im) {
     System.out.println("Before: ");
     printReIm(re, im);
     stft.fft(re, im);
     System.out.println("After: ");
     printReIm(re, im);
   }
   protected static void printReIm(double[] re, double[] im) {
8.    System.out.print("Re: [");
     for(int i=0; i<re.length; i++)
       System.out.print(((int)(re[i]*1000)/1000.0) + " ");
   System.out.print("]\nIm: [");
     for(int i=0; i<im.length; i++)
       System.out.print(((int)(im[i]*1000)/1000.0) + " ");
```

```
        System.out.println("]");
        }
9.  }
```

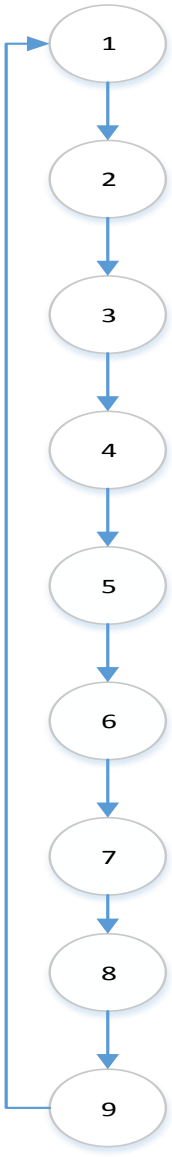
1. **Flow Graph**

Berikut merupakan Flow Graph dari algoritma Short Time Fourier Transform yang dapat dilihat pada gambar berikut.

Keterangan:

◯ = Menggambarkan kondisi

⟶ = Menggambarkan Aksi



2. Cyclomatic Complexity

V(G) = *Edge- Node* + (2P)

V(G) = 9-9+2 = 2

Dari hasil perhitungan *cyclomatic complexity* terdapat 2 independent path yaitu :

Path 1 : 1 – 2 – 3 – 4 – 5- 6 – 7 – 8 - 9

Path 2 : 1 – 2 – 3 – 4 – 5- 6 – 7 – 8 – 9-1

Keterangan:

E (Edge) : Jumlah busur pada flowgraph

N (Node) : Jumlah simpul pada flowgraph

V (G) : Cyclomatic Complexity

3. Graph Matriks Testing

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**N = (E + 1)**
1		1							1	1
2			1							0
3				1						0
4					1					0
5						1				0
6							1			0
7								1		0
8									1	0
Jumlah										1 + 1 = 2

Keterangan:

1. Baris dan kolom merepresentasikan simpul
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

4. Kesimpulan

Berdasarkan hasil pengujian *white box* dapat disimpulkan bahwa jumlah *cyclomatic complexity* pada *graph matrix* sama dengan jumlah *cyclomatic complexity* V(G) yaitu 3, sehingga tidak terjadi error pada setiap baris *sourcecode*.

4.4.5 Pengujian Performansi

Pengujian performansi merupakan proses menentukan kecepatan atau efektivitas, perangkat keras, ataupun perangkat lunak. Pengujian performansi yang dilakukan dalam tugas akhir ini yaitu mengukur tingkat keberhasilan metode yang dilakukan dengan menerapkan algoritma *Short time fourier transform* pada aplikasi pencarian judul lagu melalui media suara. Metode pengujian yang dilakukan untuk mengetahui bahwa sampai dimana lagu berhasil didapatkan dalam beberapa kata sebagai masukkan awal

1. Pencarian lagu dengan masukkan data suara menggunakan 1 kata sebagai acuan

   Pengujian pencocokan suara uji dengan lagu referensi menggunakan 1 kata.

Tabel 4.5 Kasus dan Hasil Pengujian Performansi data suara masukan dengan 1 kata

No	Judul Lagu	Data suara Masukan berupa 1 kata	Yang diharapkan	Keterangan
1	Carly rae jepsen – i really like you	really	Carly rae jepsen – i really like you	Lagu di temukan setelah pencarian ke-1
2	Mocca – I remember	remember	Mocca – I remember	Lagu di temukan setelah pencarian ke-1
3	Meghan Trainor – marvin gaye	marvin	Meghan Trainor – marvin gaye	Lagu di temukan setelah pencarian ke-1
4	Olly murs feat demi lovato – Up	never	Olly murs feat demi lovato – Up	Lagu di temukan setelah pencarian ke-1
5	Marron 5 – Sugar	broken	Marron 5 – Sugar	Lagu di temukan setelah pencarian ke-1

Banyaknya pengujian : 5 sampel lagu

Jumlah suara dikenali : 5 lagu

Presentasi keakuratan : 100%

2. Pencarian lagu dengan masukkan data suara menggunakan 2 kata sebagai acuan

Pengujian pencocokan suara uji dengan lagu referensi menggunakan 2 kata.

Tabel 4.6 Kasus dan Hasil Pengujian Performansi data suara masukan dengan 2 kata

No	Judul Lagu	Data suara Masukan berupa 1 kata	Yang diharapkan	Keterangan
1	Carly rae jepsen – i really like you	Really like	Carly rae jepsen – i really like you	Lagu di temukan setelah pencarian ke-1
2	Mocca – I remember	I remember	Mocca – I remember	Lagu di temukan setelah pencarian ke-1
3	Meghan Trainor – marvin gaye	Marvin gaye	Meghan Trainor – marvin gaye	Lagu di temukan setelah pencarian ke-1
4	Olly murs feat demi lovato – Up	I never	Olly murs feat demi lovato – Up	Lagu di temukan setelah pencarian ke-1
5	Marron 5 – Sugar	Broken down	Marron 5 – Sugar	Lagu di temukan setelah pencarian ke-1

Banyaknya pengujian : 5 sampel lagu

Jumlah suara dikenali : 5 lagu

Presentasi keakuratan : 100 %

3. Pencarian lagu dengan masukkan data suara menggunakan 3 kata sebagai acuan

Pengujian pencocokan suara uji dengan lagu referensi menggunakan 3 kata.

Tabel 4.7 Kasus dan Hasil Pengujian Performansi data suara masukan dengan 3 kata

No	Judul Lagu	Data suara Masukan berupa 1 kata	Yang diharapkan	Keterangan
1	Carly rae jepsen – i really like you	Really like you	Carly rae jepsen – i really like you	Lagu di temukan setelah pencarian ke-1
2	Mocca – I remember	Do you remember	Mocca – I remember	Lagu di temukan setelah pencarian ke-1
3	Meghan Trainor – marvin gaye	Let's marvin gaye	Meghan Trainor – marvin gaye	Lagu di temukan setelah pencarian ke-1
4	Olly murs feat demi lovato – Up	I never meant	Olly murs feat demi lovato – Up	Lagu di temukan setelah pencarian ke-1
5	Marron 5 – Sugar	I'm something weak	Marron 5 – Sugar	Lagu di temukan setelah pencarian ke-1

Banyaknya pengujian : 5 sampel lagu

Jumlah suara dikenali : 5 lagu

Presentasi keakuratan : 100%

4. Pencarian lagu dengan masukkan data suara menggunakan 4 kata sebagai acuan

Pengujian pencocokan suara uji dengan lagu referensi menggunakan 4 kata.

Tabel 4.8 Kasus dan Hasil Pengujian Performansi data suara masukan dengan 4 kata

No	Judul Lagu	Data suara Masukan berupa 1 kata	Yang diharapkan	Keterangan
1	Carly rae jepsen – i really like you	I really wanna stop	Carly rae jepsen – i really like you	Lagu di temukan setelah pencarian ke-2
2	Mocca – I remember	You smile at me	Mocca – I remember	Lagu di temukan setelah pencarian ke-2
3	Meghan Trainor – marvin gaye	That pulls me closer	Meghan Trainor – marvin gaye	Lagu di temukan setelah pencarian ke-2
4	Olly murs feat demi lovato – Up	From your side to mine	Olly murs feat demi lovato – Up	Lagu di temukan setelah pencarian ke-2
5	Marron 5 – Sugar	I don’t wanna be needing	Marron 5 – Sugar	Lagu di temukan setelah pencarian ke-2

Banyaknya pengujian : 5 sampel lagu

Jumlah suara dikenali : 5 lagu

Presentasi keakuratan : 100%

5. Pencarian lagu dengan masukkan data suara menggunakan 5- 6 kata sebagai acuan atau diambil 1 baris dari lirik lagu

   Pengujian pencocokan suara uji dengan lagu referensi menggunakan 5-6 kata.

Tabel 4.9 Kasus dan Hasil Pengujian Performansi data suara masukan dengan 1 baris lirik lagu

No	Judul Lagu	Data suara Masukan berupa 1 kata	Yang diharapkan	Keterangan
1	Carly rae jepsen – i really like you	Late night watching television	Carly rae jepsen – i really like you	Lagu di temukan setelah pencarian ke-3
2	Mocca – I remember	The way you drank you coffe	Mocca – I remember	Lagu di temukan setelah pencarian ke-3
3	Meghan Trainor – marvin gaye	Let's Marvin gaye and get it on	Meghan Trainor – marvin gaye	Lagu di temukan setelah pencarian ke-3
4	Olly murs feat demi lovato – Up	I never meant to break your heart	Olly murs feat demi lovato – Up	Lagu di temukan setelah pencarian ke-3
5	Marron 5 – Sugar	I'm hurting baby, i'm broken down	Marron 5 – Sugar	Lagu di temukan setelah pencarian ke-3

Banyaknya pengujian : 5 sampel lagu

Jumlah suara dikenali : 5 lagu

Presentasi keakuratan : 80%

Pada pengujian dengan data suara masukan menggunakan 2 baris dari lirik lagu kemampuan pencocokan mengalami penurunan akurasi ini diakibatkan karena kata yang menjadi data masukan suara meningkat atau lebih banyak dari 6 kata yang berakibat pada sulitnya pencocokan data suara masukan dengan lagu referensi.

Kesimpulan Pengujian :

1. Pengujian akurasi dilakukan dengan memberikan presentase 60 – 100 %

   Dengan ketentuan presentase :

   60 %: pencarian ditemukan setelah 5-6 kali

   80 % : pencarian ditemukan setelah 3-4 kali

   100% : pencarian ditemukan setelah 1-2 kali

2. Berdasarkan pengujian diatas yang telah dilakukan kesimpulannya adalah pencocokan lirik dengan lagu berbeda-beda penemuan pencarian hal ini dikarenakan. Semakin besar ukuran kosakata, semakin tinggi kesalahan dalam pengenalan, dengan meningkatkan ukuran *vocabulary* maka bisa terjadi kesalahan. Sementara itu setiap bahasa mempunyai huruf ketika diucapkan sulit untuk dibedakan. Sebagai contoh C, D, J . Berbeda lagi kalau yang dicari adalah kata tunggal , akan lebih mudah mencari 1 kata tunggal hasilnya akan lebih minim pengulangan pencarian.

3. Pencarian judul lagu melalu lirik lagu ini lebih berhasil ketika menggunakan bahasa inggris dibandingkan menggunakan bahasa indonesia.