

## PEMISAHAN *REFF* DAN *VERSE* SECARA OTOMATIS PADA MUSIK MP3 MENGUNAKAN KORELASI ANTAR *FRAME* BERBASIS CIRI *FFT*

### *AUTOMATIC REFF AND VERSE SEPARATION ON MP3 MUSIC USING CORRELATION OF FRAME BASE ON FFT FEATURE*

Muhammad Fadly Mustakim<sup>1</sup>, Ir. Rita Magdalena, S.T., M.T.<sup>2</sup>, I Nyoman Apraz Ramatryana, S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[fadlydodi@students.telkomuniversity.ac.id](mailto:fadlydodi@students.telkomuniversity.ac.id), <sup>2</sup>[rmagdalena@telkomuniversity.ac.id](mailto:rmagdalena@telkomuniversity.ac.id),

<sup>3</sup>[ramatryana@telkomuniversity.ac.id](mailto:ramatryana@telkomuniversity.ac.id)

#### Abstrak

Lagu merupakan suatu kesatuan musik yang terdiri atas susunan nada yang berurutan. Lagu juga merupakan seni yang melukiskan pemikiran dan perasaan manusia lewat keindahan suara. Terkadang kita menyanyikan sebuah lagu, tetapi tidak mengetahui *reff* dan *verse* nya. Penelitian ini tentang pencarian *reff* dan *verse* dengan inputan lagu yang membutuhkan potongan *reff* dan *verse* dari lagu untuk disimpan pada database. Database yang terdiri dari 25 potongan *reff* dan *verse* dari data lagu diproses secara manual. Proses tersebut membutuhkan waktu yang lama dikarenakan peneliti harus terlebih dahulu menentukan letak *reff* dan *verse* secara manual. Hal ini akan menjadi masalah bila jumlah database *reff* dan *verse* ditambahkan dengan data baru, sehingga diperlukan penelitian lebih lanjut untuk melakukan pemisahan *reff* dan *verse* secara otomatis dengan menganalisis sinyal dari file musik pada mp3.

Pada penelitian ini akan merancang metode pemisahan letak *reff* dan *verse* menggunakan perhitungan kreorelasi antar frame. Sinyal audio dari file mp3 diubah menjadi frame-frame kecil pada proses framing, selanjutnya frame tersebut ditransformasi menggunakan metode Fast Fourier Transform (FFT). Hasil transformasi FFT kemudian akan dihitung nilai korelasi antara kumpulan frame untuk mencari pola kesamaan beberapa kumpulan frame tersebut. Dari hasil korelasi tersebut didapat pola korelasi yang sama pada kumpulan frame dan selanjutnya menentukan letak *reff* dan *verse*. Proses terakhir adalah pemotongan *reff* dan *verse* sesuai dengan letak yang telah ditentukan pada proses sebelumnya. Simulasi metode ini diaplikasikan pada software pemrograman matlab.

Penelitian ini menghasilkan akurasi lebih dari 50% dari ketepatan letak *reff* dan *verse* dalam detik yang berjumlah 25 lagu dari hasil metode dibandingkan dengan letak aktual dari hasil pemisahan letak secara manual oleh peneliti pada masing-masing lagu. Waktu komputasi terbaik yang dihasilkan pada tugas akhir ini 86 detik dengan frame 1000ms untuk pemotongan 1 lagu file mp3.

Kata kunci: *Fast Fourier Transform(FFT)*, *reff*, *verse*, *korelasi*, *mp3*.

#### Abstract

*Song is a unity of music which consists of sequences of sequential tones. Songs are also art that depicts human thoughts and feelings through the beauty of sound. Sometimes we sing a song, but don't know the verse and the verse. This research is about search referrals and verses with song input that requires reffers and verses from songs to be stored in the database. A database consisting of 25 rebate pieces and verses from song data is processed manually. This process takes a long time because the researcher must first determine the location of the referrals and verses manually. This will be a problem if the number of reff and verse databases is added with new data, so further research is needed to automatically separate the reff and verse by analyzing the signal from the music file in mp3.*

*In this research, we will design a method of separating the reff and verse locations using the calculation of the frame inter-creation. Audio signals from mp3 files are converted into small frames in the framing process, then the frame is transformed using the Fast Fourier Transform (FFT) method. The FFT transformation results will then be calculated the correlation value between the frames collection to find the similarity patterns of several sets of frames. From the results of the correlation obtained the same correlation pattern in the collection of frames and then determine the location of the reff and verse. The last process is cutting reff and verse in accordance with the location specified in the previous process. This method simulation is applied to matlab programming software.*

*This research results in an accuracy of more than 50% of the exact location of the reff and verse in seconds totaling 25 songs from the results of the method compared to the actual location of the results of the manual location separation by the researcher in each song. The best computing time generated in this final project is 86 seconds with a 1000ms frame for cutting 1 song mp3 file.*

*Keywords: Fast Fourier Transform (FFT), reff, verse, correlation, mp3.*

## 1 Pendahuluan

### 1.1 Latar Belakang

Saat bernyanyi banyak orang yang ingin mengetahui *reff dan verse* lagu tersebut, sehingga peneliti membuat simulasinya untuk mempermudah penikmat musik mengetahui letak *reff dan verse* lagu tersebut.

Dengan demikian, pada penelitian ini merancang metode pemisahan letak *reff dan verse* secara otomatis menggunakan perhitungan korelasi antar *frame*. Sinyal audio dari *file mp3* diubah menjadi *frame-frame* kecil pada proses *framing*, selanjutnya *frame* tersebut ditransformasi menggunakan metode *Fast Fourier Transform* (FFT). Hasil transformasi FFT dihitung korelasi antara kumpulan *frame* untuk mencari pola kesamaan beberapa kumpulan *frame* tersebut. Dari hasil korelasi tersebut didapat pola korelasi yang sama pada kumpulan *frame* dan selanjutnya ditentukan letak *reff dan verse*. Proses terakhir adalah pemotongan *reff dan verse* sesuai dengan letak yang telah ditentukan pada proses sebelumnya. Simulasi metode ini diaplikasikan pada *software* pemrograman *matlab*.

Penelitian ini akurasi lebih dari 50% dari ketepatan letak *reff dan verse* dalam detik dari hasil metode dibandingkan dengan letak aktual dari hasil pemisahan letak secara manual oleh peneliti pada masing-masing lagu. Waktu komputasi terbaik yang dihasilkan pada tugas akhir ini 86 detik dengan *frame* 1000ms untuk pemotongan 1 lagu *file mp3*.

### 1.2 Tujuan penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang metode pemisahan letak *reff dan verse* secara otomatis menggunakan korelasi antar *frame* yang telah dilakukan transformasi FFT.
2. Menganalisis pengaruh parameter FFT terhadap performansi penentuan letak *reff dan verse* secara otomatis.

### 1.3 Rumusan Masalah

Berikut Batasan masalah yang ditentukan penulis agar pembahasan materi pada penelitian ini tidak meluas:

1. Bagaimana merancang algoritma untuk pemisahan letak *reff dan verse* secara otomatis dengan mengaplikasikan korelasi antar *frame* yang telah ditransformasi FFT?
2. Bagaimana performansi parameter FFT terhadap proses pemisahan letak?

### 1.4 Batasan Masalah

Berikut Batasan masalah yang ditentukan penulis agar pembahasan materi pada penelitian ini tidak meluas:

1. Simulasi yang dirancang menggunakan masukan lagu dengan lima genre yang berbeda.
2. Lagu yang diinput berformat \*.mp3.
3. Tidak melakukan proses pemisahan *vocal* dan *non-vocal* pada sinyal audio musik mp3 yang menjadi input dari penelitian.
4. Simulasi dari metode yang dirancang menggunakan *software matlab R2017a*.
5. Tidak melakukan *pre-processing* pada sinyal audio yang menjadi input.
6. *Reff* pada setiap lagu berjumlah tiga dan *verse* berjumlah dua.
7. Menggunakan ekstraksi ciri *Fast Fourier Transform*.

## 2 Landasan Teori

### 2.1 Konsep Dasar Suara

Suara adalah kompresi mekanikal atau gelombang longitudinal yang merambat melalui medium. Kebanyakan dari suara merupakan gabungan beberapa sinyal, tetapi suara murni secara teoritis dapat dijelaskan dengan frekuensi yang diukur dalam *Hertz* (Hz) dan amplitudo atau tekanan suara dengan pengukuran dalam decibel [9].

### 2.2 Parameter Karakteristik Suara

Terdapat beberapa parameter untuk mengetahui karakteristik suara suatu sumber suara. Diantaranya yaitu frekuensi, intensitas dan amplitudo. Frekuensi adalah ukuran jumlah putaran ulang per peristiwa dalam selang waktu yang diberikan. Untuk memperhitungkan frekuensi, seseorang menetapkan jarak waktu, menghitung jumlah kejadian peristiwa, dan membagi hitungan ini dengan panjang jarak waktu. Hasil perhitungan ini dinyatakan dalam satuan hertz (Hz).

### 2.3 Struktur Lagu

Struktur lagu merupakan susunan dan hubungan antara unsur-unsur musik dalam lagu sehingga menghasilkan suatu lagu yang bermakna. Bentuk atau struktur lagu itu merupakan susunan dan hubungan antar unsur musik dalam suatu lagu yang menghasilkan lagu memiliki suatu komposisi. Lagu memiliki tujuh komponen berikut struktur yang terdapat pada lagu *Intro, Bridge, Reff, Verse, Interlude, Modulation, dan Ending*.

### 2.4 Framing

*Framing* adalah proses yang digunakan untuk membago sinyal audio menjadi beberapa *frame*. Pada perancang sistem ini digunakan beberapa nilai *frame* diantaranya yaitu dengan nilai

0.25 second, 0,5 second, dan 1 second. Semakin besar *frame*, maka semakin cepat waktu komputasinya dikarenakan ciri yang dicocokkan oleh sistem menjadi lebih sedikit. Satu *frame* terdiri dari beberapa *sample*, tergantung tiap berapa detik suara yang akan di sampling dan berapa besar frekuensi samplingnya [8]. Sinyal hasil dari framing adalah sinyal terpotong yang diskontinu [8]. Berikut persamaan *frame/buffer*:

$$\text{Jumlah Frame} = ((I - N)/M) + 1 \quad (2.1)$$

Keterangan:

I = *Sample rate*

N = *Sample point (sample rate\*waktu framing (s))*

M = N/2

## 2.5 Windowing

Dalam melakukan pemrosesan sinyal, maka dari input yang dimasukan akan terbentuk sinyal yang magnitudenya bervariasi pada awal maupun akhir frame. Hal tersebut menghambat pemrosesan sinyal dan menghasilkan keluaran yang kurang akurat. Untuk itu perlu diaplikasikan suatu window penghalus pada setiap frame dengan mengalikan *Nframe* dengan window sehingga dapat dibangkitkan suatu feature yang lebih halus sepanjang durasi waktu tersebut.

Fungsi window yang digunakan disini adalah:

- 1 Window Hamming

$$Wham(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1} \quad (2.2)$$

- 2 Window Hanning

$$Whan(n) = 0,5 (1 - \cos \frac{2\pi n}{N-1}) \quad (2.3)$$

- 3 Window Rectangular

$$Wrec(n) = 1 \quad (2.4)$$

N merepresentasikan lebar, dalam sampel, dari waktu diskrit, fungsi jendela simetris  $w[n], 0 \leq n \leq N - 1$ .

## 2.6 Fast Fourier Transform (FFT)

*Fast Fourier Transform (FFT)* adalah metoda yang sangat efisien untuk menghitung koefisien dari *fourier* diskrit ke suatu *finite* sekuen dari data yang kompleks. Karena substansi waktu yang tersimpan lebih daripada metoda konvensional. Algoritma *FFT* berdasarkan atas prinsip pokok dekomposisi perhitungan *discrete fourier transform* dari suatu sekuen sepanjang N kedalam transformasi diskrit *Fourier* secara berturut-turut lebih kecil. Cara prinsip ini diterapkan memimpin kearah suatu variasi dari algoritma yang berbeda, dimana semuanya memperbandingkan peningkatan kecepatan perhitungan [7].

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (2.5)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-kn} \quad (2.6)$$

$$W_N = e^{-j(\frac{2\pi}{N})}, W_N \quad (2.7)$$

Dimana x merupakan nilai posisi argument dengan nilai  $W_N$  yang merupakan koefisien FFT.

FFT dengan *input* tunggal *argumen*x, menghitung nilai *fourier transform* dari *input* berupa vektor atau matrik. Jika x adalah sebuah vektor, *Fast Fourier Transform (FFT)* menghitung *fourier transform* dari vektor, bila x adalah *array* lingkaran, maka *fft* menghitung nilai *fourier transform* dari setiap kolom *array*.

## 3 Landasan Teori

### 3.1 Perancangan Sistem

#### 3.1.1 Blok Diagram

Sistem yang dibuat pada tugas akhir ini yaitu pemisahan *reff* dan *verse* dari sebuah lagu asli. Pada blok diagram sistem ini terdapat beberapa tahapan dari pemrosesan data lagu sehingga menghasilkan *reff* dan *verse* lagu yang sesuai.

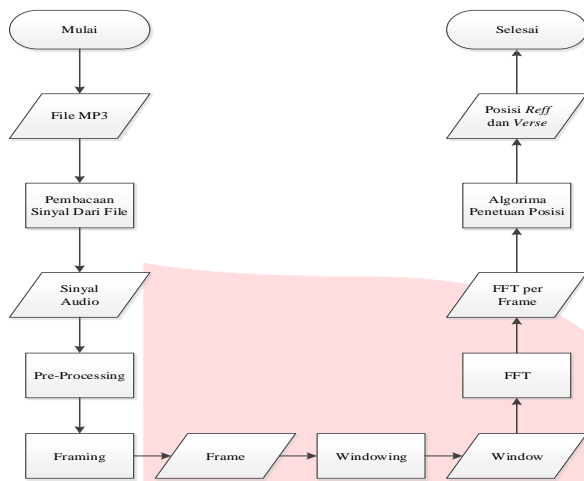


Gambar 3. 1 Blok Diagram Sistem

Dalam perancangan tugas akhir ini terdapat 3 tahapan penting yang digambarkan pada sistem ini yang pertama yaitu file mp3 untuk mengetahui *reff* dan *verse* tersebut setelah di proses oleh otomatis kemudian akan didapat letak *reff* dan *verse* selanjutnya letak itu akan digunakan untuk pemotongan file mp3 itu sesuai dengan letak.

#### 3.1.2 Alur Kerja Sistem

Proses data uji merupakan proses utama pada sistem yang dirancang untuk klasifikasi pemisahan *reff* dan *verse* pada lagu secara otomatis. Penentuan posisi menggunakan metode *Fast Fourier Transform* untuk ekstraksi ciri, korelasi dua dimensi untuk mendapatkan koefisien korelasi 2-D. Proses dimulai dengan memilih *file* MP3 dan selanjutnya didapatkan data sinyal audio. Lalu dilakukan pemotongan sinyal audio menjadi potongan kecil yang disebut *frame* dan proses ini disebut *framing*. Setiap *frame* dilakukan proses transformasi FFT untuk mendapatkan nilai koefisien FFT. Koefisien FFT disini merupakan ciri yang membedakan satu *frame* dengan *frame* yang lain. Dan proses terakhir penentuan posisi *reff* dan *verse* dengan menggunakan proses korelasi 2-D.



Gambar 3. 2 Alur Kerja Sistem

### 3.2 Pengambilan Data

Data lagu berupa lagu dengan format \*mp3 yang memiliki kualitas lagu yang paling bagus dengan frekuensi sampling 44100 Hz. Selanjutnya lagu dikelompokkan berdasarkan lagu instrument, lagu edm, lagu pop, lagu hip hop, lagu rock dan lagu funk. Jumlah data lagu yang diambil dalam pengujian adalah 25 lagu, dari total genre yang ada. Masing-masing lagu akan dicari *reff dan verse* pertamanya untuk menemukan *reff dan verse* selanjutnya.

### 3.3 Sinyal Audio

Sinyal audio yang digunakan adalah sinyal diskrit hasil dari proses pembacaan pada bahasa pemrograman matriks dengan frekuensi sampling  $F_s=44100$  sampel/detik. Sinyal audio terdiri dari dua kanal audio atau stereo.

### 3.4 Pre-processing

*Pre-processing* adalah proses awal yang dilakukan setelah perangkat mendapatkan data audio masukan. Tujuan dilakukan *Pre-processing* yaitu untuk memperbaiki kualitas data sebelum masuk ke tahap ekstraksi ciri. Adapun tahapan yang dilakukan dalam *Pre-processing* yaitu sebagai berikut:

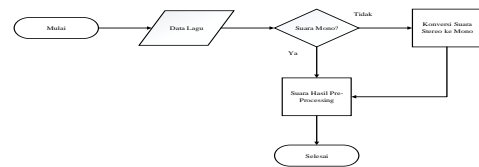
#### 3.4.1 Stereo to Mono

Pada tahap ini data akan dirubah dari stereo ke mono dengan mencari nilai rata-rata kedua kanal pada stereo untuk diubah menjadi satu kanal saja (mono).

$$\text{Rumus: Output} = \frac{k1+k2}{2} \quad (3.1)$$

Keterangan: k1 = kanal pertama

k2 = kanal kedua



Gambar 3.5 Proses Pre-Processing

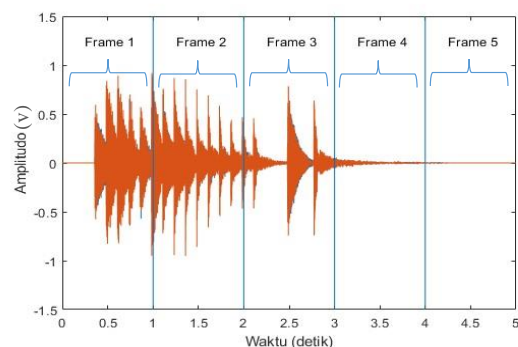
#### 3.4.2 Pre-Processing Normalisasi Amplitudo

Melakukan proses pembagian seluruh data terhadap semua nilai maksimum, agar amplitudo sinyal terbatas dari rentang nilai -1 sampai 1.  $S(i)$  merupakan sinyal input dan  $S_{max}$  merupakan sinyal maksimum dari sinyal  $S(i)$  dengan persamaan berikut:

$$S(i) = \frac{s(i)}{s_{max}} \quad (3.2)$$

### 3.5 Framing

Proses *framing* merupakan proses pemotongan sinyal audio menjadi potongan kecil dengan ukuran yang sama yaitu sebanyak  $N_{window}$ . Tujuan dari proses ini agar proses sinyal menjadi lebih fokus pada jumlah data yang terbatas. Proses *framing* tidak menggunakan overlap sesuai dengan penjelasan pada gambar 3.6.



Gambar 3. 6 Proses Framing dengan Window 1 Detik

Gambar 3.6 menunjukkan panjang pemotongan frame dengan  $N_{window} = 44100$ , sinyal audio  $x(n)$  dengan frekuensi sampling  $F_s = 44100$  memiliki durasi  $d$  selama 5 detik sehingga panjang data diskrit dari  $x(n)$  adalah:

$$Nx = dFs = (5)(44100) = 220500 \text{ sampel} \quad (3.3)$$

sehingga jumlah Nframe dihitung dengan rumus:

$$Nframe = \frac{Nx}{N_{window}} \quad (3.4)$$

dan hasil dari proses framing ini disimpan dalam sebuah matriks  $x_{frame}$  dengan dimensi  $N_{window} \times N_{frame}$ , pada gambar 3.6 akan memiliki dimensi matriks  $44100 \times 5$ .

### 3.6 Windowing

Langkah berikutnya adalah melakukan proses *window* pada setiap bagian sinyal yang telah dibuat sebelumnya. Hal ini dilakukan untuk

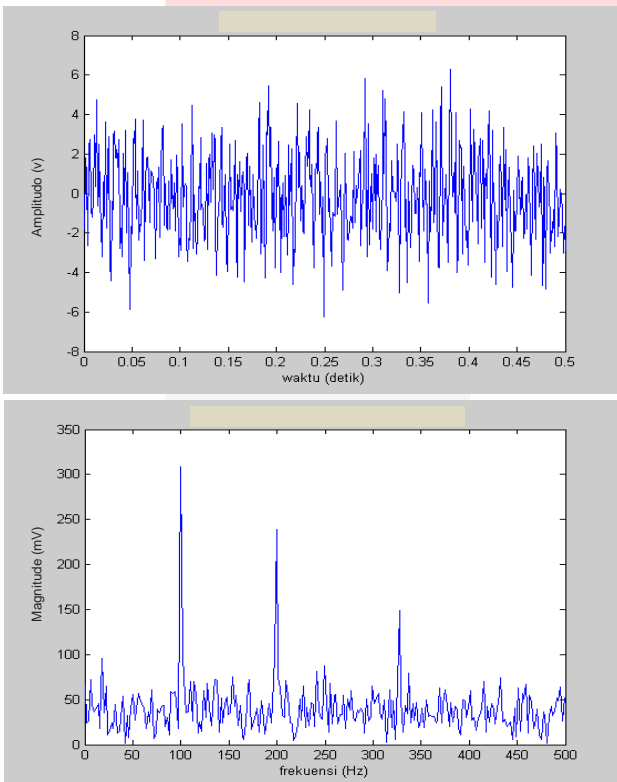


meminimalkan diskontinuitas pada bagian awal dan akhir sinyal. Jika didefinisikan sebuah *window*  $w(n)$  dan sinyal tiap bagian adalah  $x(n)$  maka sinyal hasil proses *windowing* ini adalah sebagai berikut:

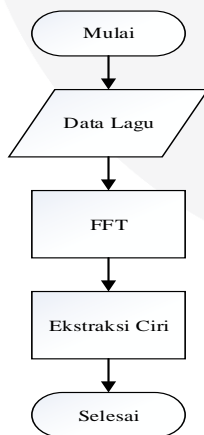
$$\delta(n) = x(n) * w(n); 0 \leq n \leq N - 1 \quad (3.5)$$

**3.7 Ekstraksi Ciri Fast Fourier Transform (FFT)**

*Fast Fourier Transform (FFT)* ini bertujuan untuk mengubah sinyal yang awalnya didomain waktu menjadi didomain *frekuensi*. Tahap ini menggunakan FFT per *frame* untuk mengambil nilai frekuensi data yang bertujuan untuk mengambil ciri pada input sinyal lagu. Setelah itu nilai *Fast Fourier Transform (FFT)* per *frame* akan mendeteksi kesamaan nada pada setiap *frame* yang ada.



Gambar 3.7 Ekstraksi Ciri FFT



Gambar 3.8 Flowchart Ekstraksi Ciri

**3.8 Algoritma Penentuan Posisi**

Algoritma penentuan posisi menggunakan *High Frequency Content (HFC)* dengan perhitungan sesuai persamaan 3.3

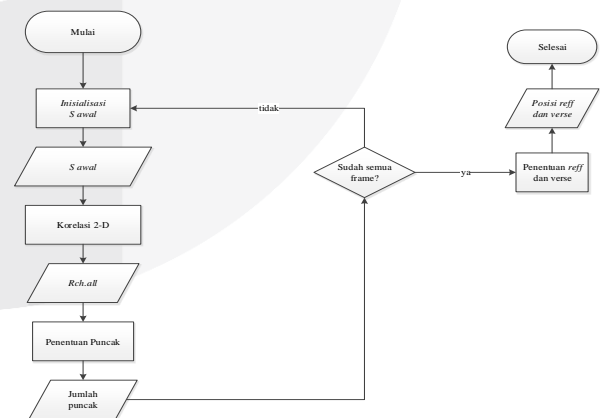
$$HFC(k, frame) = kx(kframe) \quad (3.6)$$

Proses selanjutnya adalah melakukan iterasi segmen awal yang berubah dari detik 10 sampai dengan 70 disebut sebagai  $Sawal=10,20,30,\dots,70$  yang merupakan referensi awal untuk chorus dari  $HFC(k, frame)$ .  $Sawal$  merupakan awal segmen dari  $HFC(k, frame)$  dan sepanjang  $Spanjang$  sehingga  $Sakhir = Sawal + Spanjang$ . Referensi chorus disimbolkan sebagai  $Schorus = HFC(k, Sawal: Sakhir)$  dan selanjutnya referensi tersebut dihitung koefisien korelasi 2-D terhadap seluruh  $HFC(k, frame)$  dengan proses overlapping sebesar 1 frame. Perhitungan koefisien korelasi 2-D ( $Rch, all$ ) sesuai dengan persamaan 3.4.

$$Rch, all = \frac{\sum m \sum n (Amn - \bar{A})(Bmn - \bar{B})}{\sqrt{(\sum m \sum n (Amn - \bar{A})^2)(\sum m \sum n (Bmn - \bar{B})^2)}} \quad (3.7)$$

dimana  $\bar{A} = \frac{1}{N} \sum_{i=1}^N Ai$  dan  $\bar{B} = \frac{1}{N} \sum_{i=1}^N Ai$ . Contoh hasil  $Rch, all$  dengan  $Sawal = 50$  dan  $Spanjang = 30$ . Dari hasil proses korelasi 2-D dihitung berapa nilai puncak dari korelasi yang paling tinggi selain terhadap *Schorus* karena untuk nilai korelasi 2-D yang bernilai  $Rch, all = 1$  berarti merupakan korelasi terhadap dirinya sendiri atau autokorelasi. Dua nilai puncak terdapat pada nilai  $Rch, all = 0.53$  dan  $Rch, all = 0.47$  dan dari dua puncak tersebut dihitung nilai jarak detik antar puncak sehingga didapat  $R_{ch}^1$  dan  $R_{ch}^2$ .

Dari hasil  $R_{ch}^1$  dan  $R_{ch}^2$ , kita bisa menentukan letak *Reff*. Namun agar proses menjadi otomatis maka kita harus melakukan algoritma agar  $Sawal$  bisa ditentukan secara otomatis sehingga proses penentuan posisi ini dilakukan secara berulang dan ditambahkan.



Gambar 3.9 Alur Kerja Sistem Algoritma Penentuan Posisi

### 3.8.1 Korelasi

Korelasi merupakan teknik analisis yang termasuk dalam salah satu teknik pengukuran asosiasi/hubungan. Pengukuran asosiasi merupakan istilah umum yang mengacu pada sekelompok teknik dalam statistic bivariate yang digunakan untuk mengukur kekuatan hubungan antara dua variabel. Korelasi bermanfaat untuk mengukur kekuatan hubungan antara dua variabel dengan skala tertentu.

$$r = \frac{\sum_m \sum_n (Amn - \bar{A})(Bmn - \bar{B})}{\sqrt{\sum_m \sum_n (Amn - \bar{A})^2} \sqrt{\sum_m \sum_n (Bmn - \bar{B})^2}} \quad (3.8)$$

- r = nilai korelasi
- A = nilai variabel pertama
- B = nilai variabel kedua
- $\bar{A}$  = nilai rata-rata variabel pertama
- $\bar{B}$  = nilai rata-rata variabel kedua

Tabel 3.1 Tabel Korelasi

Ko rel asi Tin ggi	Ti ng gi	Re nd ah	Re nd ah	Ta npa Ko rel asi	Tid ak ada Ko rel asi	Ta npa Ko rel asi	Re nd ah	Re nd ah	Ti ng gi	Ko rel asi Tin ggi
-1	< 0,9	> 0,9	< 0,4	> 0,4	0	< 0,4	> 0,4	< 0,9	> 0,9	+1

### 3.9 Skenario Evadimensis

Tahap selanjutnya yang dikerjakan adalah pengujian sistem. Pengujian ini dilakukan untuk mengetahui performansi sistem sehingga dapat diketahui kekurangan dan kelebihan sistem. Performansi sistem diukur berdasarkan pada parameter-parameter berikut:

#### 3.9.1 Parameter Pengujian Sistem

- a. Error  
Error merupakan tingkat kesalahan sistem dalam memisahkan *reff* dan *verse* input lagu yang diberikan terhadap jumlah data lagu secara keseluruhan. Error secara matematis dapat ditulis sebagai berikut:

$$Error = \frac{|Hasil\ pengujian\ sistem - Panjang\ referensi|}{Panjang\ referensi} \quad (3.9)$$

- b. Akurasi  
Sedangkan akurasi merupakan ukuran ketelitian sistem dalam memisahkan *reff* dan *verse*, input lagu yang diberikan sehingga menghasilkan output yang benar. Akurasi sistem secara keseluruhan dapat ditulis sebagai berikut:

$$Akurasi = (1 - error) \times 100 \% \quad (3.10)$$

#### 3.9.2 Waktu Komputasi

Waktu komputasi adalah waktu yang dibutuhkan oleh sistem untuk melakukan suatu proses. Pada sistem ini waktu komputasi dihitung

dengan menggunakan waktu selesai dikurangi waktu memulai sehingga diperoleh waktu komputasi sistem. Adapun secara matematis dapat ditulis sebagai berikut:

$$Waktu\ komputasi = waktu\ akhir - waktu\ awal \quad (3.11)$$

## 4 Pengujian dan Analisis

Pada bab ini dilakukan beberapa pengujian terhadap sistem yang telah dirancang. Setelah dilakukan pengujian maka hasil pengujian tersebut dianalisis dan disimpulkan hasilnya.

Untuk mengetahui performansi sistem yang telah dirancang, maka dilakukan pengujian terhadap sistem dengan beberapa skenario pengujian yaitu:

1. Pengaruh ukuran frame terhadap koefisien korelasi dua dimensi
2. Pengaruh jenis window terhadap koefisien korelasi dua dimensi
3. Analisis sistem pemisahan *reff* dan *verse* dengan referensi
4. Analisis sistem pemisahan *reff* dan *verse* tanpa referensi (otomatis).

### 4.1 Tujuan Pengujian Sistem

Untuk mengetahui performansi dan kelayakan sistem yang telah dirancang, maka diperlukan pengujian sistem dengan beberapa parameter. Tujuan dari pengujian sistem adalah sebagai berikut:

1. Mengetahui performansi sistem yang telah dikembangkan berdasarkan parameter akurasi dan waktu komputasi dengan melakukan pengujian pemisahan letak *reff* dan *verse* secara otomatis menggunakan kolerasi antar *frame*.
2. Mengetahui parameter yang menunjang sistem agar dapat bekerja secara maksimal.
3. Menganalisis hasil dari kerja sistem yang telah dirancang agar dapat diketahui kekurangan dari sistem tersebut.

### 4.2 Pengujian Sistem

#### 4.2.1 Skenario 1 Pengujian dan Analisis Pengaruh Ukuran Frame Terhadap Koefisien Korelasi 2 Dimensi

Tabel 4.1 Pengaruh Ukuran Frame Terhadap Koefisien Korelasi 2 Dimensi

Ukuran Frame (s)	<i>Verse 2</i>	<i>Reff 2</i>	<i>Reff 3</i>
	rata-rata nilai KK-2D	rata-rata nilai KK-2D	rata-rata nilai KK-2D
0.25	0.14	0.29	0.21
0.5	0.15	0.32	0.23
1	0.15	0.34	0.25

Dari hasil uji yang dilakukan, didapatkan hasil yang terbaik dari ukuran frame 1 s, karena mendapatkan rata-rata nilai koefisien korelasi 2 dimensi terbesar.

**Tabel 4.2** Pengaruh Ukuran Frame Terhadap Ranking Posisi

Ukuran Frame (s)	<i>Verse 2</i>	<i>Reff 2</i>	<i>Reff 3</i>
	ranking posisi	rangking posisi	rangking posisi
0.25	256.56	386.88	329.36
0.5	170.76	179.44	187.12
1	27.52	14.92	19.44

Dari hasil uji yang dilakukan, didapatkan hasil pengaruh ukuran frame terbaik di 1 s, karena mendapatkan rata-rata nilai ranking terendah.

**Tabel 4.3** Pengaruh Ukuran Frame Terhadap Waktu Proses

Ukuran Frame (s)	waktu proses	
	<i>verse</i>	<i>reff</i>
0.25	11.32	11.02
0.5	5.72	5.56
1	3.14	3.02

Dari hasil uji yang dilakukan, didapatkan hasil pengaruh ukuran frame terbaik di 1s pula, karena mendapatkan waktu proses tercepat. Jadi, ukuran frame terbaik dibanding 2 frame lainnya adalah di ukuran frame 1s karena mendapatkan hasil ujian terbaik dibanding yang lainnya.

#### 4.3.2 Skenario 2 Pengujian dan Analisis Pengaruh Jenis Window Terhadap Koefisien Korelasi 2 Dimensi

**Tabel 4.4** Pengaruh Jenis Window terhadap Koefisien Korelasi 2 Dimensi

Jenis Windowing	<i>Verse 2</i>	<i>Reff 2</i>	<i>Reff 3</i>
	rata-rata nilai KK-2D	rata-rata nilai KK-2D	rata-rata nilai KK-2D
Rectangular	0.15	0.34	0.25
hanning	0.15	0.33	0.23
hamming	0.15	0.33	0.23

Dari hasil uji yang dilakukan, dapat dilihat jenis window mendapatkan rata-rata hasil yang mirip, jadi dapat disimpulkan semua jenis window bagus terhadap korelasi 2 dimensi.

**Tabel 4.5** Pengaruh Jenis Window terhadap Ranking

Jenis Windowing	<i>Verse 2</i>	<i>Reff 2</i>	<i>Reff 3</i>
	rata-rata ranking posisi	rata-rata ranking posisi	rata-rata ranking posisi
Rectangular	27.52	14.92	19.44
hanning	38.48	21.8	25.48

hamming	37.4	20.88	25.12
---------	------	-------	-------

Dari hasil uji yang dilakukan, didapatkan hasil rata-rata ranking terendah yang merupakan hasil paling baik terdapat pada jenis window rectangular.

**Tabel 4.6** Pengaruh jenis Window terhadap Waktu Proses

Jenis Windowing	waktu proses	
	<i>verse</i>	<i>reff</i>
Rectangular	3.27	3.20
hanning	3.16	3.09
hamming	3.16	3.07

Dari hasil uji yang dilakukan, tidak terdapat perbedaan yang signifikan antar jenis window jadi dapat disimpulkan semua jenis window terhadap waktu proses berbanding lurus atau sama.

#### 4.3.3 Skenario 3 Pengujian dan Analisis Sistem Pemisahan *Reff* dan *Verse* dengan Referensi

Dalam skenario ini dilakukan pengujian pengaruh ukuran frame pada proses ekstraksi ciri. Dalam pengujian digunakan data 25 lagu yang dikelompokkan menjadi 5 genre yaitu pop, EDM, funk, rock, dan hip-hop. Dari hasil pengujian, dilakukan analisis *Reff* dan *Verse* dengan referensi.

Pengujian terhadap lagu pada bagian *verse* dengan genre FUNK

**Tabel 4.7** Hasil Data Lagu *Verse* FUNK

Jenis Lagu	Ukuran frame	Akurasi	Waktu Komputasi
FUNK	1000	58.62 %	5
		76.47 %	4
		100%	6
		100%	5
		93.33 %	3

Dari tabel 4.7 didapat nilai akurasi terbaik pada angka 100% dan waktu komputasi tercepat pada angka 3 detik.

#### 4.3.4 Skenario 4 Pengujian dan Analisis Sistem Pemisahan *Reff* dan *Verse* Tanpa Referensi (otomatis)

Dalam skenario ini dilakukan pengujian pengaruh ukuran frame pada proses ekstraksi ciri. Dalam pengujian digunakan data 25 lagu yang dikelompokkan menjadi 5 genre yaitu pop, EDM, funk, rock, dan hip-hop. Dari hasil pengujian, dilakukan analisis akurasi menggunakan persamaan rumus (3.3).

Pengujian terhadap lagu dengan genre EDM

**Tabel 4.17** Hasil Data Lagu EDM

Jenis Lagu	Ukuran frame	Waktu Komputasi	Akurasi
EDM	1000	85	84.22 %
		98	87.50 %
		93	86.67 %
		97	92.86 %
		89	61.12 %

Dari tabel 4.12 didapat nilai akurasi terbaik pada angka 92.86% dan waktu komputasi tercepat pada angka 85 detik.

## 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan hasil pengujian sistem dan analisis sistem dalam Tugas Akhir ini, dapat diambil kesimpulan sebagai berikut:

1. Pada pengujian sistem yang sudah dilakukan dapat disimpulkan bahwa dengan menggunakan sistem ini FFT dapat bekerja dengan baik dalam menentukan *reff* lagu.
2. Sistem dapat menampilkan potongan *reff* lagu setelah lagu diinputkan kedalam program.
3. Sistem dapat menentukan *reff* lagu dari seluruh genre yang diuji dengan baik, dengan akurasi rata rata diatas 50% dengan rata-rata lama komputasi 97 detik pada ukuran frame 1000 ms.

### 5.2 Saran

Adapun saran yang dapat digunakan untuk penelitian Tugas Akhir selanjutnya, yaitu:

1. Pengembangan metode ekstraksi ciri lain yang dapat meningkatkan akurasi yang lebih baik.
2. Pengembangan cara kerja sistem yang mempunyai waktu komputasi lebih baik.
3. Pengembangan terhadap lagu dengan genre yang berbeda.
4. Pengembangan pencarian bagian-bagian lagu untuk semua genre lagu.
5. Menambahkan proses pemisahan vocal dan non-vocal.

## 6 Daftar Pustaka

- [1] Agatha Rizka Prassiwi. 2017. Analisis Dan Simulasi Klasifikasi Judul Lagu Dari Senandung Manusia Menggunakan Ekstraksi Ciri Fast Fourier Transform (FFT). Bandung: Universitas Telkom.
- [2] Ganang Wicaksono. 2017. Analisis Dan Simulasi Klasifikasi Judul Lagu Dari Senandung Manusia Menggunakan Ekstraksi Ciri Mell Frequency Cepstral Coefficient (MFCC). Bandung: Universitas Telkom.
- [3] Ignatius Yoslan Kurniawan. 2016. Analisis Dan Simulasi Identifikasi Judul Lagu Melalui Senandung Manusia Menggunakan Ekstraksi Ciri LPC (Linear Predictive Coding). Bandung: Universitas Telkom.
- [4] Arintyo Archamadi. 2016. Analisis Dan Simulasi Identifikasi Judul Lagu Dari Senandung Manusia Menggunakan Ekstraksi Ciri DCT (Discrete Cosine Transform). Bandung: Universitas Telkom.
- [5] Obed Gabriel Filemon Simanungkalit. 2017. Perancangan Dan Simulasi Pemisahan Reff Lagu Dengan Metode Fast Fourier Transform (FFT). Bandung: Universitas Telkom.
- [6] Shimon Anterio Armando Sinaga. 2017. Analisis Dan Simulasi Pencarian Reff Dan Verse Lagu Pada Musik Digital Dengan Metode Linear Predictive Coding (LPC). Bandung: Universitas Telkom.
- [7] Heru, Chandra. 2017. Analisis dan Simulasi Pencarian *Reff dan verse* Lagu pada Musik Digital dengan Metode *Modified Discrete Cosine Transform*. Bandung: Universitas Telkom, 2017
- [8] Firmansyah Patriandhika. 2017. Simulasi Dan Analisis Pencari Reff Dan Verse Lagu Pada Musik Digital Dengan Metode Korelasi. Bandung: Universitas Telkom.
- [9] Nur Rohman, Sigit. 2012. Aplikasi Pencirian Dengan Linear Predictive Coding Untuk Pembelajaran Pengucapan Nama Hewan Dalam Bahasa Inggris Menggunakan Jaringan Saraf Tiruan Propagasi Balik. Semarang: Universitas Diponegoro.