

## **Separasi Suara Musik Pada Robot KRSTI Menggunakan *Band Pass Filter* FIR Untuk Sinkronasi**

**Muhammad Nur Mukhlis<sup>1</sup>, Yuning Widiarti<sup>2</sup>, Ryan Yudha Adhitya<sup>3</sup>**

<sup>1,2,3</sup>Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal

Politeknik Perkapalan Negeri Surabaya

Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111

E-mail : [muhammadnur15@student.ppns.ac.id](mailto:muhammadnur15@student.ppns.ac.id)

### **Abstrak**

KRSTI merupakan bagian dari ajang Kontes Robot Indonesia (KRI), dimana robot tersebut akan menari mengikuti sinyal suara musik yang diberikan. Dalam pelaksanaan lomba suara derau dapat mengganggu suara masukan yang diterima oleh robot, sehingga perlu dilakukan penyaringan terhadap sinyal asli. Maka dibutuhkan sebuah sistem yang dapat menyaring sinyal asli dari suara derau dengan menggunakan *Bandpass Filter FIR*. Dimana teori dari *Bandpass Filter FIR* dapat menghilangkan *noise* diantara 2 batas frekuensi, yaitu frekuensi batas bawah dan frekuensi batas atas. Untuk melakukan penelitian ini sinyal suara dalam domain waktu akan dirubah menjadi domain frekuensi dengan menggunakan algoritma *Fast Fourier Transform* (FFT) untuk dilakukan analisa, dan didapatkan 208 base frekuensi pada audio tari enggang. Untuk mendapatkan hasil filter yang baik maka digunakan besar orde pemfilteran yaitu orde 300. Selanjutnya bobot yang dihasilkan dari analisa digunakan untuk memfilter sinyal pada Mikrokontroler Raspberry pi. Keberhasilan pemfilteran dibuktikan menggunakan perhitungan *Root Mean Square Error* (RMSE) dengan nilai 0.1060 untuk derau berupa *Gaussian Noise*, 0.1363 untuk derau berupa suara tepuk tangan, dan 0.0243 untuk derau berupa suara dari Frekuensi Generator dengan rata rata dari ketiga jenis derau (*noise*) yaitu 0.2666 dimana nilai tersebut membuktikan bahwa perbedaan audio tanpa *noise* dan audio hasil pemfilteran hanya sedikit memiliki perbedaan

**Kata Kunci :** *Bandpass Filter, (FFT) Fast Fourier Transform, FIR, (IFFT) Invers Fast Fourier Transform, Raspberry Pi*

### **1. PENDAHULUAN**

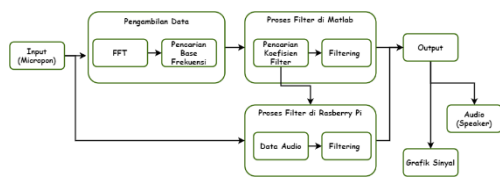
Saat ini kita telah memasuki era industri 4.0. Dimana teknologi berkembang sangat pesat. Perlahan sudah semua sudah beralih ke arah teknologi digital. Sehingga interaksi antara manusia dan teknologi sudah tidak dapat dipungkiri lagi. Semua pemenuhan kebutuhan kini telah tersedia secara digital, mulai dari jual-beli, jasa, hingga transaksi pembayaran serta robot pembantu pekerjaan manusia .

Robot sendiri berasal dari bahasa ceko “roboto” yang artinya pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot sendiri juga bisa diartikan sebagai seperangkat alat mekanik yang dapat melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, maupun dengan program yang telah didefinisikan terlebih dahulu atau biasa kita sebut kecerdasan buatan.

Salah satu program dari DIKTI yaitu Kontes Robot Indonesia atau yang biasa kita kenal sebagai KRI. KRI adalah kegiatan kontes bidang robotika yang dapat diikuti oleh tim mahasiswa dari institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di

Kementerian Riset, Teknologi dan Pendidikan Tinggi. Kontes Robot Indonesia ini sudah menjadi kegiatan tahunan yang dilaksanakan oleh Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia. Tujuan diadakannya KRI ini sendiri yaitu menumbuhkan-kembangkan ilmu pengetahuan dan teknologi tentang robot, mengaplikasikan ilmu pengetahuan dan teknologi ke dalam dunia desain dan aplikasi nyata, meningkatkan kepekaan mahasiswa dalam mengembangkan teknologi robotika, membudayakan sikap kompetitif yang positif di lingkungan perguruan tinggi, meningkatkan kepekaan mahasiswa terhadap seni budaya bangsa Indonesia, serta meningkatkan pengetahuan aplikasi sensor dan teknik kontrol yang terupdate. Terdapat beberapa kategori dalam ajang KRI ini antara lain 1. KRAI (Kontes Robot Abu Indonesia), 2. KRPAI (Kontes Robot Pemadam Api Indonesia), 3. KRSBI (Kontes Robot Sepak Bola Indonesia) Humanoid, 4. KRSBI (Kontes Robot Sepak Bola Indonesia) Beroda, 5. KRSTI (Kontes Robot Seni Tari Indonesia).

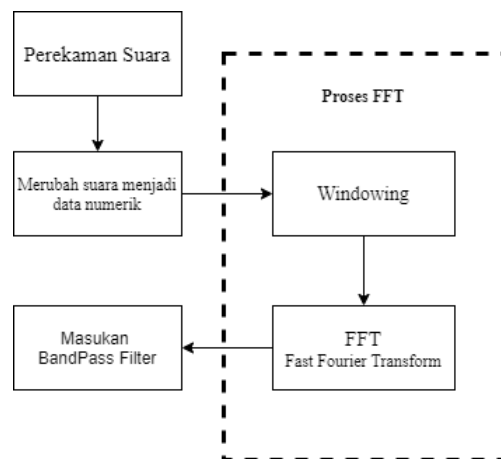




**Gambar 2.2** Diagram Blok Keseluruhan Sistem

## 2.4 Perancangan Sistem FFT

Diagram alir perancangan sistem *Fast Fourier Transform* (FFT) pada tugas akhir ini menjelaskan konsep awal dalam perancangan sistem transformasi menggunakan *Fast Fourier Transform* (FFT). Sistem ini menggunakan mikrofon yang dihubungkan di laptop untuk mendapatkan sinyal suara. Sinyal suara tersebut akan diolah menggunakan *software* matlab sehingga menjadi sebuah sinyal dalam domain waktu dan menjadi sebuah sinyal dalam grafik. Dimana pada sumbu X-nya berupa angka dengan satuan waktu dan sumbu Y-nya berupa angka satuan magnitude. Sebelum ditransformasi menggunakan *Fast Fourier Transform* (FFT) ada satu proses yang harus dilakukan yaitu proses *Windowing*. ditunjukkan pada **Gambar 2.3** dibawah ini.



**Gambar 2.3** Diagram Alir perancangan sistem *Fast Fourier Transform* (FFT)

## 2.5 Windowing

Saat suara sudah terekam dan sudah diubah dalam bentuk data numerik maka akan dibentuk sebuah gelombang. Dimana gelombang tersebut akan diproses dengan menggunakan metode *Windowing*. Proses ini bertujuan untuk meminimalisir sinyal yang tidak kontinyu atau mengenali sinyal baru yang berbeda dengan sinyal asli. Pada prinsipnya

penggunaan fungsi *windowing* adalah dengan cara melewati sinyal yang mempunyai frekuensi sembarang dikonvolusikan dengan fungsi *window* tertentu sehingga dapat mereduksi sinyal-sinyal yang tergolong bocor sebelum dilakukan proses transformasi. Ada beberapa fungsi *windowing* yang telah ada diantaranya *hann*, *hamming*, *triangular* dan *rectangular*.

Secara matematis penggunaan metode *windowing* dapat dinyatakan dalam persamaan sebagai berikut

$$X(n) = \sum_{k=0}^{N-1} w(k) \times X(k) W_N^{kn}$$

Dimana :

$X[n]$  = Output Sinyal

$X[k]$  = Input Sinyal

$W[k]$  = *Hamming Window*

$N$  = Jumlah Sampel tiap frame

## 2.6 Fast Fourier Transform

*Fast Fourier Transform* (FFT) memiliki fungsi yang sama dengan *Discrete Fourier Transform* (DFT). Hanya saja *Fast Fourier Transform* (FFT) mempunyai algoritma perhitungan yang lebih efisien dibandingkan dengan *Discrete Fourier Transform* (DFT)[1]. *Fast Fourier Transform* (FFT) sendiri ditemukan pada tahun 1965 yang merupakan pengembangan dari *Fourier Transform* (FT). *Fourier Transform* (FT) digunakan untuk membagi sebuah sinyal menjadi berbeda beda dalam fungsi eksponensial yang kompleks. *Fast Fourier Transform* (FFT) merupakan suatu algoritma untuk menghitung transformasi *fourier discrete* dengan cepat dan efisien.

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-2\pi jkn/N} \quad (1)$$

Dimana :

$k$  = indeks domain waktu = 0, 1,..., N-1

$n$  = indeks domain frekuensi = 0, 1,...,N-1

$N$  = Jumlah total data

$J = \sqrt{-1}$

$X(k)$  dipilih menjadi genap dan ganjil sehingga persamaan 1 menjadi

$$\begin{aligned} X(n) &= \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k) e^{-2\pi j(2k)n/N} + \\ &\sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k+1) e^{-2\pi j(2k+1)n/N} \\ &= \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k) e^{-2\pi j(2k)n/N} + \\ &e^{-2\pi jkn/N} \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k+1) e^{-2\pi j(2k+1)n/N} \end{aligned}$$

(2)

Dengan mendefinisikan  $W_N = e^{(-2\pi j/N)}$ , maka persamaan 2 menjadi

$$X(n) = \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k)W_N^{2kn} + W_N^n \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k+1)W_N^{2kn} \quad (3)$$

Karena

$$W_N^2 = e^{-j\left(\frac{2\pi}{N}\right)^2} = e^{-j\left(\frac{2\pi}{N}\right)}, \text{ maka } W_N^2 = W_{\frac{N}{2}}.$$

Jadi persamaan 3 menjadi

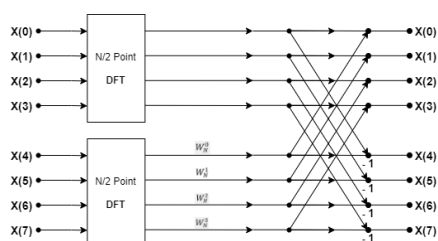
$$X(n) = \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k)W_{\frac{N}{2}}^{kn} + W_N^n \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k+1)W_{\frac{N}{2}}^{kn} \quad (4)$$

Setelah domain waktu dibagi dua, domain frekuensi juga dibagi menjadi dua menjadi

$$X(n + N/2) = \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k)W_{\frac{N}{2}}^{k(n+\frac{N}{2})} + W_N^{(n+\frac{N}{2})} \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k+1)W_{\frac{N}{2}}^{k(n+\frac{N}{2})} \quad (5)$$

$$= \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k)W_{\frac{N}{2}}^{kn} - W_N^n \sum_{k=0}^{\left(\frac{N}{2}\right)-1} x(2k+1)W_{\frac{N}{2}}^{kn}$$

Persamaan 4 dan 5 merupakan *Decimation in Time* (DIT) yang mana sequence data dipilah menjadi dua bagian menjadi genap dan ganjil dan menggambarkan gabungan dua DFT  $N/2$  data. Penggunaan sifat periodik dari fungsi kernel pada FFT membuat perhitungan menjadi lebih cepat dan lebih efisien karena cukup mengganti tanda operasi menjadi minus. Secara sederhana persamaan 4 dan 5 digambarkan menggunakan diagram kupu-kupu (*butterfly diagram*) seperti pada **Gambar 2.4** berikut



**Gambar 2.4** Diagram Kupu-Kupu

Sehingga perhitungan *Discrete Fourier Transform* (DFT) dapat disederhanakan

dengan menggunakan perhitungan *Fast Fourier Transform* (FFT) cukup setengah periode saja lalu setengah periode berikutnya dapat dihitung menggunakan persamaan

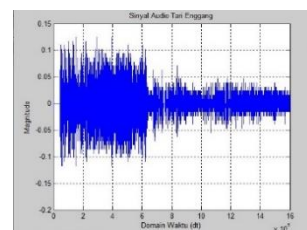
## 2.7 Pembuatan Laporan Akhir

Penyusunan laporan akhir bertujuan sebagai bentuk tanggung jawab dan otentik atas terselenggaranya penelitian ini.

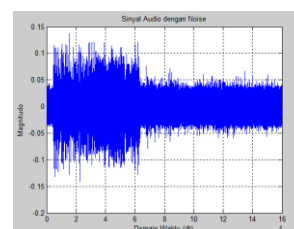
## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengambilan Data Audio dan Penambahan Noise

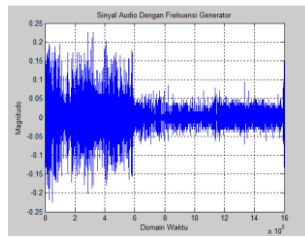
Pengambilan data audio tari enggang dilakukan melalui perekaman dengan durasi 3 menit 20 detik atau setengah dari audio asli yaitu 6 menit 40 detik. Dilakukannya pengambilan setengah durasi dikarenakan audio pada tari tersebut diputar sebanyak 2 kali. Sinyal dari audio tari enggang dapat dilihat pada **Gambar 3.1**. Pengambilan data kali ini dilakukan dengan cara penambahan audio tari enggang dengan *noise* di matlab. Terdapat 3 jenis derau dalam penambahan noise pada audio, diantaranya yaitu *Gaussian Noise* pada **Gambar 3.2**, Suara tepuk tangan pada **Gambar 3.3**, dan Suara dari Frekuensi Generator pada **Gambar 3.4** yang sudah di set pada frekuensi tertentu. Untuk derau berupa *Gaussian Noise* Penambahan dilakukan setelah dilakukan perekaman.



**Gambar 3.1.** Sinyal Audio tanpa *noise*



**Gambar 3.2.** Sinyal Audio dengan *Gaussian Noise*



**Gambar 3.4.** Sinyal Audio dengan *Noise* Frekuensi Generator

### 3.2 Perhitungan Koefisien

Secara matematis nilai dari koefisien filter dapat dicari dengan menggunakan persamaan 10 berikut..

$$h(n) = h_d(n) \cdot w(n) \\ = \frac{\sin w_{c2}\pi(n-\alpha) - \sin w_{c1}\pi(n-\alpha)}{\pi(n-\alpha)} \cdot \left[ 0,54 - 0,46 \cos \frac{2\pi n}{M-1} \right]$$

dimana :

- $h(n)$  = Koefisien Filter
- $h_d(n)$  = Respon Impuls
- $w(n)$  = Hamming Window
- $M$  = Panjang Filter

Pada persamaan diatas respon impuls  $h_d(n)$  mempunyai panjang yang tak terbatas. Sehingga harus dipotong untuk mendapatkan persamaan filter FIR dengan panjang  $M$  dengan cara mengalikan  $h_d(n)$  dengan *Hamming window*. Dalam Analisa ini koefisien filter diambil dari nilai hasil *training* dari matlab yang digunakan untuk memfilter data sinyal. Jumlah koefisien filter bergantung dengan jumlah orde + 1. Jadi jika dalam proses *training* data jumlah orde yang digunakan 16 maka jumlah koefisien atau panjang filter yang akan keluar berjumlah 17 data. Untuk mendapatkan nilai koefisien filter maka harus dilakukan normalisasi frekuensi, yaitu dengan membagi nilai batas atas dan batas bawah dengan setengah frekuensi sampling.

### 3.3 Respons Impulse dan Frekuensi

Setelah mendapatkan koefisien filter maka dapat diketahui bentuk dari *Respons Impulse* dan *Respons Frekuensi*. Pada **Tabel 3.1 – Tabel 3.3** merupakan contoh dari respons impuls salah satu base frekuensi, yaitu pada base frekuensi tengah yaitu pada frekuensi 1730.79Hz-1733.23Hz.

**Tabel 3.1** Respon Orde 50

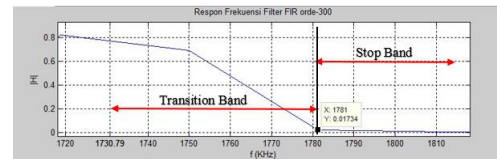
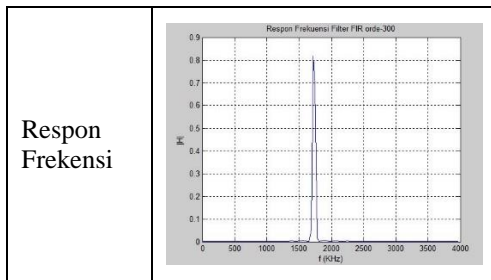
Orde 50m	
Respon Impulse	
Respon Frekuensi	

**Tabel 3. 2** Respon Orde 150

Orde 150	
Respon Impulse	
Respon Frekuensi	

**Tabel 3.3** Respon Orde 300

Orde 300	
Respon Impulse	



Gambar 3.5 Transition band orde 300

Pada tabel 3 diatas pada orde 300 terdapat gambar *respon frekuensi* yang mempunyai *transition band* sangat pendek, dikarenakan sifat dari metode FIR membutuhkan orde yang besar untuk menghasilkan *output* yang halus, dengan begitu pemfilteran dengan base frekuensi yang banyak yaitu orde 300 sudah cukup dalam menghasilkan *output* yang diinginkan.. Transition band pada orde 300 dapat dilihat pada **Gambar 3.5** berikut ini..

Pada **Gambar 3.5** transition band pada komponen frekuensi sinyal *input* dengan nilai orde 300 mulai mengalami peredaman, dan pada daerah stop band komponen frekuensi sinyal *input* mengalami peredaman yang sangat besar hingga bernilai nol sehingga bisa dikatakan daerah stop band frekuensi sinyal *input* dihentikan

### 3.4 Pengujian Keseluruhan Sistem

Hasil Pemfilteran yang telah dianalisa yaitu berupa grafik sinyal dengan spesifikasi frekuensi sampling 8000 dan panjang filter 301. Grafik sinyal *output* dapat dilihat pada Tabel berikut

Tabel 3.4 Hasil Pemfilteran

Jenis Derau (Noise)	Hasil Pemfilteran Matlab R2014a	Hasil Pemfilteran RaspberryPi3
Gaussian Noise		
Suara Tepuk Tangan		
Frekuensi Generator		

### 3.5 Perhitungan RMSE

Root Meand Square Error (RMSE)

digunakan sebagai metode pengukuran dengan mengukur perbedaan nilai dari audio hasil



pemfilteran suara dengan hasil filter yang diinginkan atau audio tanpa noise. Persamaan untuk mencari nilai RMSE menggunakan persamaan 11 sebagai berikut.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (At - Ft)^2}{n}}$$

Dimana :

RMSE = Root Mean Square Error

n = Panjang Data

At = Nilai Data Actual (Audio Hasil Filter)

Ft = Nilai Data Hasil Peramalan (Audio Tanpa Noise)

**Tabel 3.5** Hasil Perhitungan RMS

No.	Jenis Derau (Noise)	Nilai RMSE
1	<i>Gaussian Noise</i>	0.1060
2	Suara Tepuk Tangan	0.1363
3	Frekuensi Generator	0.0243
Rata-rata		0.2666

**Tabel 3.5** merupakan hasil perhitungan dari *Root Mean Square Error*. Dimana dari ketiga audio tersebut dengan variasi noise yang berbeda beda memiliki perbedaan yang sedikit dengan rata rata nilai RMSE yaitu 0.2666

## 4. PENUTUP

### 4.1 Kesimpulan

Setelah melakukan pengujian dan analisis dari simulasi perancangan rangkaian band pass filter FIR menggunakan software MATLAB versi 2014a, diperoleh kesimpulan yaitu sebagai berikut:

4. Dari 3 jenis orde yang dilakukan orde 300 sudah dapat meredam suara *noise* dengan baik. Analisa dilakukan dengan melihat grafik sinyal *output* dan mendengarkan *output* audio dengan 3 kali percobaan pada orde 50, orde 100 dan orde 300.
5. Dari analisa yang dilakukan pada audio asli Tari Enggang terdapat base frekuensi sebanyak 208 data, sehingga orde yang digunakan dalam proses pemfilteran dibuat untuk memiliki transition band yang kecil. Orde 300 dalam pemfilteran ini dapat menghasilkan *output* yang baik dengan panjang filter 301.
6. Hasil pemfilteran medapatkan nilai RMSE 0.1060, 0.1363, dan 0.0243 dengan nilai rata rata yaitu 0.2666. dimana nilai tersebut membuktikan bahwa perbedaan audio tanpa noise dan audio hasil pemfilteran hanya sedikit memiliki perbedaan.

7. Penerapan filter pada *mikrokontroller* dilakukan dengan mengambil data koefisien filter pada matlab selanjutnya data tersebut diproses dengan menggunakan Mikrokontroller Rasberry pi dengan bahasa pemrograman Phyton, dengan durasi 177 jam. Pemilihan Mikrokontroller Rasberry pi dikarenakan dalam analisa ini membutuhkan device yang memiliki sampling rate yang tinggi.

## 4.2 Saran

Dalam pengujian yang telah dilakukan dalam penelitian ini, maka terdapat beberapa hal yang harus diperhatikan dan dikembangkan lagi untuk penelitian selanjutnya, yaitu untuk mendapatkan *output* filter yang bagus dari audio yang memiliki frekuensi yang sama antara audio asli dan noise dapat menggunakan transformasi 3 dimensi menggunakan STFT(*Short Time Fourier Transform*) dengan 3 sumbu grafik yaitu x, y dan z dimana sumbu z merupakan domain waktu yang bisa membedakan jika ada 2 sinyal pada frekuensi yang sama dan untuk Proses analisa dalam perhitungan data yang besar disarankan untuk menggunakan mikrokontroller yang mempunyai spesifikasi yang tinggi.

## 5. DAFTAR PUSTAKA

- [1] B. Auditiakusuma, "Rancang Bangun Pembuatan Aplikasi Sampling Frekuensi File Audio Menggunakan Metode Filter FIR(Finite Impulse Response)," 2011.
- [2] R. Y. Sipasulta, A. S. M. L. St, and S. R. U. A. Sompie, "Simulasi Sistem Pengacak Sinyal Dengan Metode Fft (Fast Fourier Transform)," *E-Journal Tek. Elektro Dan Komput.*, vol. 3, no. 2, pp. 1–9, 2014.
- [3] R. Y. Adhitya *et al.*, "Rotor bars fault detection by DFT spectral analysis and Extreme Learning Machine," *2016 Int. Symp. Electron. Smart Devices, ISESD 2016*, pp. 103–108, 2016, doi: 10.1109/ISESD.2016.7886701.