

# Aplikasi Tuner Alat Musik Sasando *Real-Time* Menggunakan Teknik Fast Fourier Transform (FFT) dan Harmonic Product Spectrum (HPS)

Amin A. Maggang<sup>1</sup>, Sarlince O. Manu<sup>2</sup>, Molina O. Odja<sup>3</sup>

<sup>1,2,3</sup> *Jurusan Teknik Elektro, Fakultas Sains dan Teknik, Universitas Nusa Cendana*

<sup>1</sup>*Email: amin\_maggang@staf.undana.ac.id*

<sup>2</sup>*Email: Sarlince\_manu@staf.undana.ac.id*

<sup>3</sup>*Email: Molina.odja@staf.undana.ac.id*

## ABSTRAK

Tujuan dari penelitian ini adalah untuk mendesain suatu aplikasi tuner sasando yang memiliki tingkat akurasi yang tinggi dalam mendeteksi frekuensi fundamental dari jenis sasando 32 senar menggunakan Algoritma Fast Fourier Transform (FFT) dan Harmonic Product Spectrum (HPS). Tingkat akurasi yang tinggi ditentukan dari resolusi frekuensi yang tinggi. Menurut Lourde, M (2009), kemampuan telinga manusia untuk membedakan bunyi adalah pada resolusi 0.5 Hz. Manusia tidak mampu membedakan bunyi jika resolusi atau selisih dua frekuensi yang berdekatan lebih kecil dari 0.5 Hz. Oleh karena itu, aplikasi yang didesain harus mampu memberikan resolusi 0.5 Hz. Berdasarkan hasil simulasi, Aplikasi Tuner sasando yang telah dibangun mampu memberikan output resolusi 0.5 Hz. Selain itu, aplikasi yang dibangun juga dapat menampilkan spektrum frekuensi dan mampu memberikan informasi apabila frekuensi dari senar yang sedang dituning sudah tepat pada frekuensi fundamentalnya. Tidak hanya itu, apabila frekuensinya masih kurang atau sudah melewati frekuensi dasarnya, maka akan ditampilkan juga pada outputnya.

**Kata Kunci:** *Tuner Sasando, Fast Fourier Transform (FFT), Harmonic Product Spectrum (HPS)*

## 1. PENDAHULUAN

Sasando merupakan suatu alat musik tradisional yang berasal dari pulau Rote, salah satu kabupaten di propinsi Nusa Tenggara Timur. Alat musik ini merupakan jenis alat musik petik seperti gitar atau kecapi. Sama seperti gitar, sasando juga memiliki jenis akustik dan listrik. Perbedaan yang mencolok antara sasando dan alat musik yang menggunakan senar lainnya adalah pada resonatornya. Sasando menggunakan daun lontar sedangkan alat musik petik lainnya menggunakan kayu sebagai resonator. Dilihat dari jumlah senar, sasando memiliki jumlah senar yang lebih banyak dari gitar. Sasando memiliki 28, 32, 40 bahkan 45 senar. Jumlah senar yang banyak memperlebar rentang oktaf dari sasando.

Namun dengan semakin banyaknya senar, seorang pemain sasando harus mampu untuk melakukan stem senar yang tepat sesuai dengan frekuensi pada setiap senarnya. Hal ini tentunya akan sangat sulit bagi seorang pemula yang ingin belajar memainkan alat musik ini. Selain itu satu jenis stem hanya dapat digunakan untuk memainkan dua nada dasar. Untuk dapat bermain dari nada dasar yang lain harus dilakukan penyeteman ulang untuk mendapatkan susunan nada yang sesuai dengan nada dasar yang ingin digunakan (edonsasando, 2011). Nada dasar yang dapat dimainkan pada sasando yaitu nada C, D, F, dan G. Untuk dapat melakukan stem sasando, diperlukan kemampuan mendengar yang tinggi dan hal ini tentunya bisa dilakukan oleh orang yang sudah berpengalaman menggunakan alat musik tersebut.

Oleh karena itu, pada penelitian ini penulis akan merancang suatu aplikasi Tuner untuk menolong

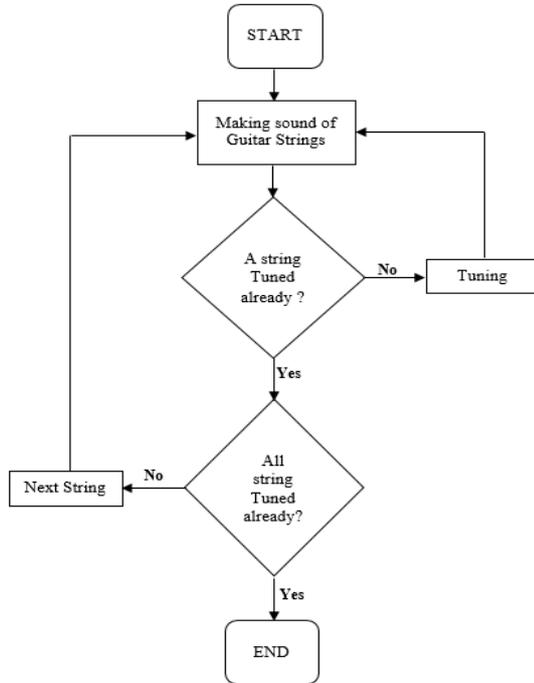
orang yang masih awam dalam menyetem senar sasando. Jenis tuner yang selama ini sudah dibuat adalah hanya untuk gitar dan piano. Sampai saat ini belum ada peneliti yang mencoba untuk mendesain suatu algoritma khusus untuk melakukan tuning senar pada sasando.

Algoritma yang akan dibuat untuk aplikasi tuner sasando diambil dari teknik sinyal prosesing yaitu FFT dan HPS. Kedua teknik ini sudah banyak digunakan dalam bidang penelitian pengenalan kord. Selain itu implementasi kedua algoritma paling banyak dilakukan pada software MATLAB. Oleh karena itu, MATLAB juga digunakan pada penelitian ini sebagai software untuk membangun aplikasi Tuner Sasando.

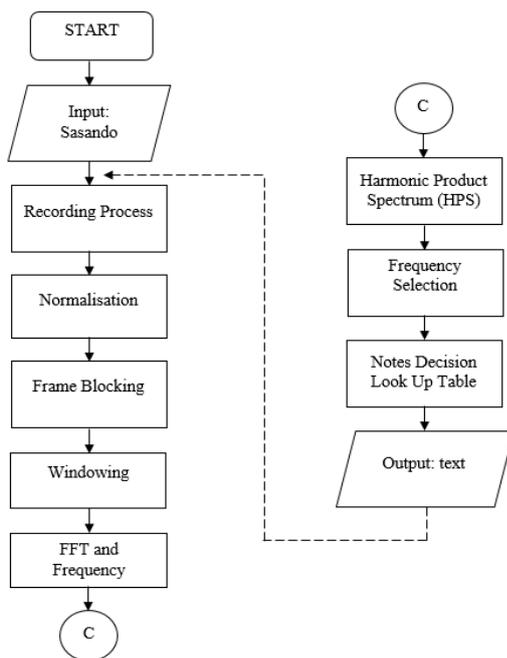
## 2. METODE PENELITIAN

Desain dan implementasi tuner sasando pada MATLAB dilakukan sesuai dengan dua diagram alur pada gambar 3. Pada gambar 3.a menjelaskan bagaimana pengguna dapat melakukan stem sasando secara real-time. Ketika sebuah senar dipetik, suaranya akan diverifikasi apakah ia memiliki frekuensi fundamental yang diinginkan atau tidak. Gambar 3.b merupakan alur kerja aplikasi. Sinyal input, dalam hal ini suara senar sasando, direkam dalam format .wav dan kemudiannya amplitudonya dinormalisasikan. Setelah itu sinyal tersebut harus di blok dalam frame dan dikonvolusikan dengan window Hann untuk mencegah kebocoran frekuensi ketika

proses FFT. Output dari proses FFT adalah spectrum frekuensi yang terdiri dari fundamental frekuensi dan harmoniknya. Spektrum ini yang dimasukkan ke dalam algoritma Harmonic Product Spectrum untuk dikompresi dan kemudian menentukan nilai Frekuensi Fundamental secara otomatis



Gambar 1.a Proses Tuning



Gambar 3.b Proses Aplikasi

### 3. HASIL DAN PEMBAHASAN

#### 4.1 Hasil Desain Aplikasi Tuner Sasando pada Matlab

Hasil yang diperoleh dalam penelitian adalah suatu aplikasi tuner sasando yang telah berhasil didesain menggunakan software Matlab. Aplikasi yang dibangun dapat dipakai untuk melakukan tuning sasando secara real-time. Berikut merupakan tampilan Grafik User Interface (GUI) aplikasi yang dibangun.



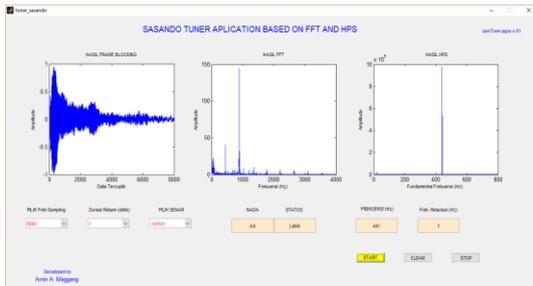
Gambar 4 Hasil Desain GUI Tuner Sasando

Berdasarkan gambar 4 terdapat tiga *pushbutton*, tiga *axes*, dan tiga *popupmenu*, dan empat *edit text*. Fungsi dari setiap menu dijelaskan sebagai berikut:

- Pushbutton* “START” untuk memulai proses tuning sasando
- Pushbutton* “CLEAR” untuk mereset atau membersihkan semua fungsi pada tuner sasando menjadi nol.
- Pushbutton* “STOP” untuk memberhentikan proses tuning
- Popupmenu* “PILIH frekuensi sampling” untuk memudahkan user memilih frekuensi sampling yang diinginkan
- Popupmenu* “Durasi Rekam” untuk menentukan waktu rekaman
- Popupmenu* “Pilih Senar” untuk memilih salah satu senar dari 32 jenis senar
- Edit text* “NADA” untuk menampilkan nada dari jenis senar yang dipilih pada popupmenu “pilih senar”
- Edit text* “FREKUENSI (Hz)” untuk menampilkan keluaran frekuensi yang sedang dituning
- Edit text* “Frek. Resolusi (Hz)” untuk menampilkan keluaran resolusi frekuensi dari setiap pemilihan frekuensi sampling dan durasi rekam
- Axis* “Hasil *Frame Blocking*” untuk menampilkan sinyal hasil *Frame Blocking*
- Axis* “FFT” untuk menampilkan spectrum sinyal hasil *Fast Fourier Transform*
- Axis* “HPS” untuk menampilkan spectrum fundamental frekuensi hasil *Harmonic Product Spectrum*

### 4.2 Penjelasan coding program Tuner Sasando secara Real Time

Agar tuner dapat melakukan tuning secara *Real Time* maka setiap komponen pada GUI akan diberikan sintaks program tertentu sehingga proses tuning secara *Real Time* dapat berjalan sesuai dengan yang diinginkan. Aplikasi tuner secara real-time ini lah yang akan digunakan untuk pengujian dan analisis data. Berikut merupakan tampilan aplikasi ketika proses tuning sasando secara real-time.



Gambar 5 Tampilan proses tuning secara real time

a. Pushbutton “START”  
Ketika *pushbutton* START diklik, sistem akan mulai untuk merekam bunyi senar sasando yang dibunyikan dan akan diproses lebih lanjut sesuai dengan tahapan-tahapan algoritma tuningnya. Untuk lebih jelasnya, proses selanjutnya dan sintaks programnya dapat dijelaskan sebagai berikut

- Proses merekam bunyi senar sasando  
Sintaks program yang digunakan ditampilkan pada gambar dibawah ini

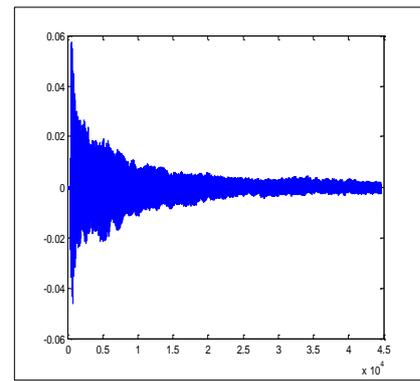
```
global henti;
henti = 0;
while henti == 0
    disp('tuning process..')
    pause(1);

    durasi = handles.durasi;
    fs = handles.fs;
    frame = fs; % length frame blocking
    hfs = fs/2;

    sg = wavrecord(durasi*frame,fs,'double'); % sinyal sasando masukan untuk wav
```

Gambar 6 Sintaks rekam bunyi senar sasando

Pada gambar 6 diatas, tampak bahwa terdapat sintaks while yang berfungsi untuk melakukan proses perulangan atau biasa disebut dengan *looping* selama kondisi ekspresi terpenuhi. Proses *looping* ini dapat dihentikan dengan menggunakan sintaks global. Hasil keluaran dari tahapan ini berupa sinyal suara berformat wav. gambar 7 dibawah ini merupakan bentuk sinyal keluaran nada A4 yang dihasilkan dari tahap ini



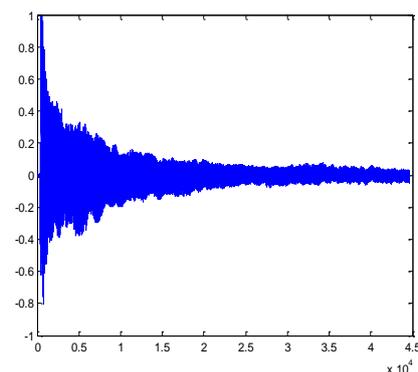
Gambar 7 Sinyal Keluaran A4 pada Matlab

- Proses Normalisasi  
Sintaks program yang digunakan untuk normalisasi adalah sebagai berikut

```
% normalisasi
y1 = sg/max(abs(sg));
```

Gambar 8 Sintaks program untuk Normalisasi

Proses awal yang akan dilakukan oleh sistem setelah mendapatkan masukan sinyal suara sasando, adalah normalisasi. Proses ini dilakukan dengan membagi data masukan (data sinyal suara senar sasando) dengan nilai absolut data tersebut. Tujuannya adalah untuk membuat amplitudo maksimum dari sinyal selalu sama yaitu satu. Proses ini menghasilkan sinyal seperti gambar berikut



Gambar 9 Sinyal audio setelah Normalisasi

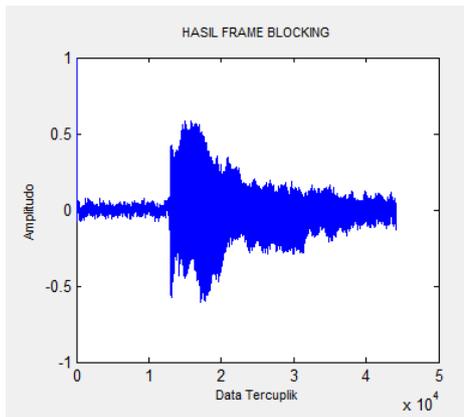
- Proses *Frame Blocking*  
Sintaks program untuk melakukan *frame blocking* adalah sebagai berikut

```
%frame blocking
length_frame = durasi*fs;
x0 = y1(1:length_frame);

axes(handles.axes1)
plot(x0)
xlabel('Data Tercuplik','FontSize',8)
ylabel('Amplitudo','FontSize',8)
```

Gambar 10 Sintaks program untuk Frame Blocking

Setelah proses normalisasi, proses selanjutnya adalah *Frame blocking*. Proses ini bertujuan untuk mencuplik sebagian data sesuai panjang nilai *Frame blocking*. Panjang *frame blocking* bergantung dari durasi rekaman dikalikan dengan frekuensi sampling. Keluarannya kemudian akan ditampilkan pada kotak *axes 1* (Hasil *Frame blocking*).



Gambar 11 Sinyal Audio setelah proses Frame blocking

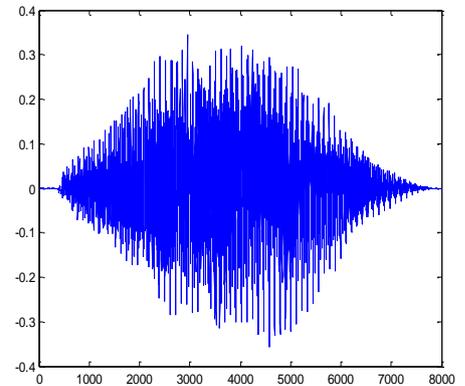
Sinyal yang telah dicuplik ini memiliki panjang sama seperti pada frekuensi samplingnya. Gambar diatas diambil dengan frekuensi sampling sebesar 44100 Hz sehingga tampak pada gambar, panjang sinyalnya pada sumbu x dimulai dari nol sampai sekitar 44100.

- Proses Windowing  
Sintaks program yang digunakan ditunjukkan pada gambar dibawah ini

```
% windowing
h = hann(length_frame);
y2 = x0.*h;
```

Gambar 12 Sintaks program untuk proses windowing

Proses windowing dilakukan setelah melewati proses *frame blocking*. Tujuan dari windowing adalah untuk menghilangkan diskontinuitas yang diakibatkan oleh proses *frame blocking*. Tampak pada gambar diatas bahwa jenis window yang digunakan adalah jenis "hann".



Gambar 13 Hann Window

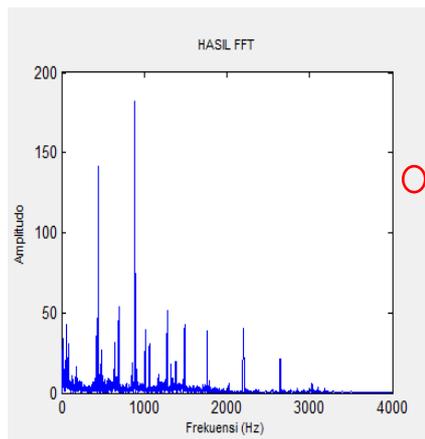
Gambar 13 adalah jenis *hann window* dengan panjang window adalah 8000 titik. Pada proses diatas, sinyal keluaran *y2* adalah hasil perkalian antara sinyal hasil *frame blocking* *x0* dan window *hann*.

- Proses FFT  
Sintaks program FFT yang digunakan adalah sebagai berikut

```
%FFT
FFT = fft(y2, length_frame)
FFT = FFT(1:size(FFT,1)/2);
y3 = abs(FFT)
axes(handles.axes2)
plot(y3)
xlabel('Frekuensi (Hz)','FontSize',8)
ylabel('Amplitudo','FontSize',8)
```

Gambar 14 Sintaks program untuk proses FFT

Proses selanjutnya adalah transformasi sinyal menggunakan teknik FFT untuk menghasilkan spektrum dari sinyal hasil windowing. Spektrum yang dibangkitkan inilah yang akan dianalisis untuk mengetahui frekuensi fundamental dari senar yang sedang dituning. Keluarannya akan ditampilkan pada *axes 2* (Hasil FFT). Berikut adalah spektrum frekuensi hasil FFT dari nada A4.



Gambar 15 Spektrum Frekuensi hasil FFT

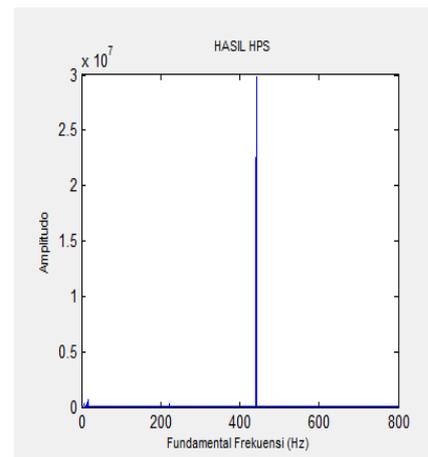
Tampak pada gambar 15 bahwa komponen frekuensi yang muncul adalah harmonik dari nada dasar A4 (440 Hz). Spektrum frekuensi fundamental nada A4 yang diberi lingkaran merah. Dapat dilihat bahwa spektrum harmonik secara keseluruhan memiliki jarak yang sama.

- Proses HPS  
Sintaks program yang dituliskan untuk HPS adalah sebagai berikut

```
% HPS
hps1 = downsample(y3,1);
hps2 = downsample(y3,2);
hps3 = downsample(y3,3);
hps4 = downsample(y3,4);
hps5 = downsample(y3,5);
hps6 = downsample(y3,6);
y = [];
for i=1:length(hps5)
    Product = hps1(i) * hps2(i) * hps3(i) * hps4(i) * hps5(i) * hps6(i);
    y(i) = [Product];
end
[~,n]=findpeaks(y, 'SORTED', 'descend');
Maximum = n(1);
axes(handles.axes3)
plot(y)
xlabel('Fundamental Frekuensi (Hz)', 'FontSize', 8)
ylabel('Amplitudo', 'FontSize', 8)
```

Gambar 16 Sintaks Program untuk proses HPS

Seperti ditunjukkan pada gambar 16, proses HPS adalah proses yang bertujuan untuk menganalisa susunan spektrum hasil FFT dan menentukan frekuensi fundamentalnya. Berdasarkan gambar 20, jumlah downsampling yang dilakukan adalah sebanyak 6 kali. Hasil HPS dapat dilihat pada gambar berikut



Gambar 17 Spektrum Hasil proses HPS

Gambar 17 adalah tampilan hasil algoritma Harmonic Product Spectrum (HPS). Tampak bahwa setelah proses HPS, Spectrum yang muncul hanya satu frekuensi, yaitu frekuensi fundamentalnya. Dalam hal ini frekuensi nada A4, 440 Hz.

- Frekuensi *output*  
Untuk menghasilkan suatu keluaran frekuensi fundamental, maka sintaks yang digunakan adalah sebagai berikut

```
frek = ( Maximum / length_frame ) * handles.fs ; % CorrectFactor
set(handles.edit1, 'string', frek);
set(hObject, 'BackgroundColor', [1 1 0]);
```

Gambar 18 Sintaks Program untuk frekuensi Fundamental

Sintaks ini akan menampilkan frekuensi fundamental yang diambil dari hasil perhitungan HPS, yaitu frekuensi keluaran yang memiliki amplitudo tertinggi. Hasilnya akan ditampilkan pada *edit text 1* (Frekuensi (Hz)) dan berupa *string*.

- Teks Nada keluaran  
Untuk menghasilkan output nada keluaran yang sesuai dengan frekuensi standar, maka digunakan teknik *look up table*. Berikut merupakan sintaksnya untuk beberapa nada.

```
% cek KURANG, PAS, atau LEBIH
status = get(handles.popupmenu1, 'value');
if status == 2 %
set(handles.edit2, 'string', 'G2');
if frek < 97.49
set(handles.edit3, 'string', 'Kurang');
elseif (frek >= 97.49) && (frek <= 98.49)
set(handles.edit3, 'string', 'TEPAT');
else
set(handles.edit3, 'string', 'Lebih');
end
elseif status == 3
set(handles.edit2, 'string', 'G3');
if frek < 195.5
set(handles.edit3, 'string', 'Kurang');
elseif (frek >= 195.5) && (frek <= 196.5)
set(handles.edit3, 'string', 'TEPAT');
else
set(handles.edit3, 'string', 'Lebih');
end
elseif status == 4
set(handles.edit2, 'string', 'D4');
if frek < 293.16
set(handles.edit3, 'string', 'Kurang');
elseif (frek >= 293.16) && (frek <= 294.16)
set(handles.edit3, 'string', 'TEPAT');
else
set(handles.edit3, 'string', 'Lebih');
end
end
```

Gambar 19 Sintaks Program untuk Teks Nada Keluaran

Berdasarkan sintaks program pada gambar 19, frekuensi standar keluarannya akan ditunjukkan pada *edittext2*. Sedangkan kata “Kurang”, “Tepat”, dan “Lebih” akan ditampilkan pada *edittext3*. Kurang menunjukkan bahwa proses tuning yang sedang dilakukan belum mencapai frekuensi stadarnya. Sebaliknya kata lebih menunjukkan bahwa frekuensi yang sedang dituning sudah melebihi frekuensi standarnya. Proses tuning harus menunjukkan kata “Tepat”, yang artinya proses tuning sudah berada pada nada standarnya. Sintaks yang dituliskan di atas akan menampilkan kata “Tepat” apabila frekuensi senar yang sedang dituning dengan jarak mutlak lebih kecil dari 0.5 Hz.

$$|x - \text{frekuensi standard}| \leq 0.5 \text{ Hz.}$$

b. *Pushbutton* “CLEAR”

Untuk menghapus hasil tuning sebelumnya, maka dibuatkan button clear dengan sintaks sebagai berikut

```
axes(handles.axes1)
plot(0)
axes(handles.axes2)
plot(0)
axes(handles.axes3)
plot(0)
set(handles.edit1, 'string', '');
set(handles.edit2, 'string', '');
set(handles.edit3, 'string', '');
clear all
clc
```

Gambar 20 Sintaks program untuk pushbutton CLEAR

Ketika *Pushbutton* CLEAR ditekan maka *axes* dan *edit* pada GUI akan dibersihkan dari data hasil tuning.

c. *Pushbutton* “STOP”

Sintaks yang digunakan untuk memberhentikan proses tuning adalah

```
global henti
henti = 1;
```

Gambar 21 Sintaks program untuk Pushbutton STOP

Dapat dilihat bahwa terdapat fungsi global digunakan untuk menghentikan proses tuning. Hal ini karena pada awalnya digunakan fungsi *while* untuk melakukan *looping* sehingga diakhir program harus menggunakan fungsi global untuk menghentikan proses *looping* tersebut.

d. *Popupmenu* “Pilih Frek Sampling”

*Popupmenu* ini digunakan untuk memilih jenis frekuensi sampling yang diinginkan untuk proses tuning.

e. *Popupmenu* “Pilih durasi”

*Popupmenu* “pilih durasi” ini berfungsi untuk memudahkan user dalam memilih durasi rekaman yang secara tidak langsung menentukan resolusi frekuensi yang ada pada edit text “frekuensi resolusi”.

f. *Popupmenu* “Pilih Senar”

*Popupmenu* ini digunakan untuk memilih frekuensi sampling. Tujuannya adalah untuk memudahkan pengujian dengan frekuensi sampling yang berbeda. Sintaks pada *popupmenu* ini sudah tergabung dengan *look up table* yang terdapat berbagai nada frekuensi standar setiap senar sasando.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian, tahap pembuatan tuner sasando menggunakan software MATLAB telah selesai dilaksanakan. Dari hasil tersebut maka terdapat beberapa hal yang dapat disimpulkan dari aplikasi tuner sasando tersebut.

1. Aplikasi tuner yang dibangun menggunakan MATLAB tersebut mampu melakukan tuning sasando secara *real-time*.
2. Dilihat dari sisi *input* atau masukan, pengguna bisa memilih tiga jenis frekuensi sampling, yaitu 8000 Hz, 16000 Hz dan 44100 Hz. Selain itu, Aplikasi yang telah dibangun dapat melakukan tuning 32 senar sasando dengan nada dasar yang berbeda. Dengan menggunakan *popupmenu*,

pengguna dapat memilih jenis senar yang ingin dituning

3. Dilihat dari sisi output, aplikasi tersebut dapat menampilkan Jenis Nada dan frekuensi fundamentalnya pada dua *edit text* yang berbeda. Lebih dari itu, aplikasi tersebut dapat memberikan informasi nilai frekuensi, apakah “Kurang”, “Tepat”, atau “Lebih” dari frekuensi standarnya. Spektrum frekuensi hasil frame blocking, hasil FFT dan Harmonic product spectrum juga dapat dilihat secara *real time*.

## DAFTAR PUSTAKA

- [1] Da Costa A. M, “Stem Gitar Bass Menggunakan Harmonic Product Spectrum” Yogyakarta. 2017.
- [2] Edonsasando, “Petunjuk Cara memainkan Sasando”. <https://edonsasando.wordpress.com/2011/12/02/petunjuk-cara-memainkan-sasando/> . 2011.
- [3] Gunawan, Dadang & Juwono, Filbert Hilman. “*Pengolahan sinyal digital dengan pemrograman matlab*”. Yogyakarta: Graha Ilmu. 2012.
- [4] Ludemen, Lonnie C., “Fundamental of Digital Signal Processing”, Canada, John Wiley & Sons, Inc. 1987.
- [5] Lay Ipank, “Sasando”. <http://www.buset-online.com/sasando/> . 2016.
- [6] M. Soeharto, “Belajar Notasi Balok”, Jakarta: Gramedia. 1978.
- [7] N. Siripya dan T. Nagarijan. “Pitch Estimation Using Harmonic Product Spectrum derived from DCT,” IEEE. 2013. Setiawan, Paramita, dan Setiamaanaha, “Brain Stimulating Effects of Acoustic Sasando” Jurnal pendidikan penabur: 22-29. 2009.
- [8] S. Wiguna, “Frequency Detection of Piano Basic Tone Based on Correlation, Discrete Cosine Transform, and Fast Fourier Transform,” Telkom university. 2009.
- [9] T.G. Permana., “ Identifikasi Akor Gitar Menggunakan Algoritma Harmonic Product Spectrum,” Telkom Univerity. 2014. Y. Maulana, ” Implementation of guitar tuner based on Fast Fourier Transform,” Telkom University. 2012.