

TRANSFORM FOURIER CEPAT MATEMATIS UNTUK MENGANALISIS SPEKTRUM FREKUENSI LINIER SINYAL TUTUR

Salman Abd. Cadum¹, Prayoto², Adhi Susanto³, Kirbani Sri Brotopuspito⁴

¹ Jurusan Teknik Eektro Fakultas Teknik Universitas Gadjah Mada, email: salman_abd_2002@yahoo.com

² Jurusan Fisika Fakultas MIPA Universitas Gadjah Mada

³ Jurusan Teknik Elektro Fakultas Teknik Universitas Gadjah Mada

⁴ Jurusan Geofisika Fakultas MIPA Universitas Gadjah Mada

Abstrak

Spektrum menggunakan transform Fourier cepat FFT matematis dimaksudkan untuk melakukan analisis frekuensi. Resolusi frekuensi spektrum tergantung nilai-nilai cuplikan sinyal audio/tutur digital, yang terbagi dalam sejumlah poin data FFT, yaitu semakin besar jumlah poin data, semakin baik resolusi frekuensi spektrum. Skala frekuensi pada spektrum ini menggunakan skala frekuensi linier yang menyebarkan komponen frekuensi secara merata pada tampilan spektrum secara keseluruhan. Penanganan terhadap berkas gelombang tutur, atau masukan audio, dapat berjalan lebih cepat ketika menggunakan skala frekuensi linier daripada menggunakan skala frekuensi logaritmis. Penerapan fungsi berbagai window terhadap data dapat membantu mengurangi efek kebocoran yang terjadi pada spektrum frekuensi.

Kata kunci : FFT Matematis, Spektrum Frekuensi Linier, Sinyal Tutur

1. PENGANTAR

FFT biasanya dilaksanakan berdasar sejumlah poin data masukan yang selalu berupa kelipatan dua (128, 256, 512, 1024, 2048, atau 4096). Resolusi frekuensi spektrum untuk sinyal audio/tutur tergantung pada jumlah poin data ini, semakin besar jumlah poin data, semakin baik resolusi frekuensi spektrum. Frekuensi maksimum yang dihitung oleh FFT atau batas frekuensi tertinggi spektrum adalah benar setengah frekuensi cuplikan digital [1].

Dalam setiap proses analisis spektrum, resolusi waktu dan resolusi frekuensi memiliki hubungan terbalik, resolusi frekuensi yang sangat bagus berkaitan dengan resolusi waktu yang buruk, sebaliknya resolusi waktu yang sangat bagus berkaitan dengan resolusi frekuensi yang buruk. Hubungan antara resolusi waktu (*Time Resolution*, TR) dalam detik dan resolusi frekuensi (*Frequency Resolution*, FR) dalam Hz adalah:

$$TR = 1/FR \dots\dots\dots(1)$$

Akibat logis prinsip ketidakpastian ini adalah bahwa memilih resolusi frekuensi yang sangat bagus akan menutupi *detail* waktu dalam spektrum audio/tutur. Memilih resolusi frekuensi yang tepat dan resolusi waktu yang sesuai menjadi suatu strategi antara kebutuhan untuk mengamati *detail* frekuensi yang baik dalam spektrum dengan kebutuhan untuk mengamati variasi waktu yang cepat dalam spektrum [2].

Skala linier membagi sumbu frekuensi menjadi *interval* frekuensi yang sama, skala linier tidak banyak memiliki muatan komputasional karena resolusi frekuensi selalu konstan di sepanjang bidang frekuensi seluruhnya. Mengingat bahwa muatan komputasional FFT yang besar, dapat dibuktikan bahwa penggarapan terhadap berkas gelombang atau masukan audio/tutur dapat berjalan lebih cepat ketika menggunakan suatu skala frekuensi linier dari pada menggunakan skala frekuensi logaritmis.

Penting untuk memahami bagaimana pilihan terhadap suatu skala frekuensi linier mempengaruhi tampilan spektrum tingkat tekanan suara. Spektrum mengukur tingkat tekanan suara pada bidang frekuensi sempit diskrit di sepanjang spektrum. Ketika menggunakan suatu

skala frekuensi linier, lebar setiap bidang pengukuran yang konstan, sama dengan nilai resolusi frekuensi yang dipilih ketika memasang tampilan spektrum.

Analisis frekuensi untuk sinyal dengan tingkat tekanan suara yang konstan pada semua frekuensi seperti derau putih, atau jumlah energi yang sama terkandung dalam setiap subbidang pengukuran. Penganalisis spektrum yang dikalibrasi secara tepat dengan masukan derau putih akan menunjukkan suatu gambar spektrum datar ketika menggunakan skala frekuensi linier.

2. LANDASAN TEORI

Teori Fourier menyatakan bahwa bentuk gelombang (*waveform*) periodis dapat dipecah kedalam serangkaian sinusoid yang memiliki frekuensi, amplitude, dan fase yang berbeda. Dalam sintesis *wavetable*, amplitude parsialnya tetap, dan sementara hal ini umumnya cukup untuk memberikan kesan (*impression*) tipe warna nada (*timbre*) tertentu, keterbatasannya mencegahnya menerima hasil yang lebih tahan dalam kelangsungan hidup (*life-like*), hal ini terutama untuk dua alasan: a) di alam, frekuensi parsial jarang merupakan perkalian bilangan bulat yang tepat (*exact*), dan b) amplitude parsial individu hampir selalu berbeda-beda relatif terhadap satu sama lain selama suatu nada (*tone*). Kedua efek ini mustahil dicapai dengan sintesis *wavetable*, dan sintesis aditif, oleh karena itu, dengan hubungannya yang jelas dengan analisis Fourier, maka Fourier menawarkan solusi masalah ini.

Penganalisis FFT mentransformasikan data dari domain waktu ke domain frekuensi dengan menghitung transform Fourier cepat. Hal ini didasarkan pada integral Fourier:

$$y = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt \quad (2)$$

Namun ini merupakan suatu bentuk yang dapat dihitung secara numeris. Integral ini mensyaratkan bahwa suatu sinyal kontinu diintegrasikan selama waktu yang tak terhingga, tentu saja, diinginkan hasil dalam waktu yang terhingga, dan karena komputer berhubungan dengan angka, maka diperlukan digitasi bentuk gelombang, yang dapat membuat waktu bersifat diskrit. Kedua perubahan terhadap sinyal ini mengakibatkan kesalahan dalam spektrum frekuensi yang dihitung. Cuplikan sinyal pada waktu diskrit dapat menyebabkan *aliasing* (yang dapat terlihat sebagai sinyal bayangan (*phantom*) pada tampilan). Pengubahan batas integral panjang tak terhingga menjadi panjang terhingga dapat menyebabkan kesalahan yang disebut kebocoran (yang muncul sebagai energi dari titik tertentu dalam spektrum terbaaur (*smearred*) naik dan turun melintasi spektrum). Karena ketidakmungkinan mengukur suatu sinyal untuk waktu yang tak terhingga, maka penganalisis mengubah batas integral ke panjang waktu yang dibutuhkan untuk mengumpulkan blok cuplikan. Blok cuplikan disebut *time record*. FFT mensyaratkan bahwa sinyal dalam *time record* diulang terus-menerus sepanjang waktu. Jika *time record* yang diulang secara aktual tampak seperti sinyal asli, maka tidak akan terjadi kebocoran. Jika, pada sisi lain, tak terlihat seperti sinyal aslinya, maka terjadi kebocoran. Penerapan fungsi *window* terhadap data yang ada akan dapat membantu mengurangi efek kebocoran dalam domain frekuensi [3].

Untuk menghitung spektrum y dari sinyal tutur X dapat digunakan FFT dengan menggunakan persamaan berikut:

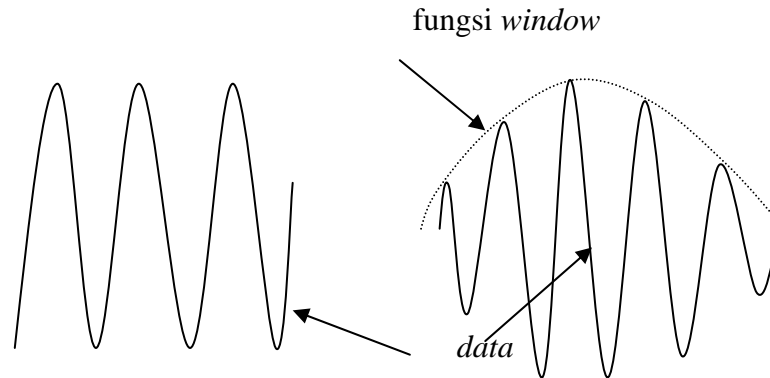
$$y_m = \sum_{k=0}^{N-1} X_k \cdot e^{2\pi i k m / N} \quad (3)$$

dengan $m = 0 \dots N-1$, dan $k = 0 \dots N-1$, demikian juga, untuk menghitung sinyal X dari spektrum y dapat digunakan invers transform Fourier cepat (IFFT) menggunakan persamaan berikut:

$$X_k = \frac{1}{N} \sum_{m=0}^{N-1} y_m \cdot e^{-2\pi i k m / N} \quad (4)$$

dengan $m = 0 \dots N-1$, dan $k = 0 \dots N-1$.

Memilih suatu *data window* biasanya dilakukan melalui suatu kompromi. Setiap *window* 'mengotori' (*smear*) spektrum frekuensi. Ini berarti bahwa suatu 'puncak yang runcing' yang berkaitan dengan suatu gelombang sinus dalam sinyal semakin melebar. Di sisi lain kebocoran palsu (*spurious*) ke dalam frekuensi di sekitarnya akan berkurang. Fungsi *window* disajikan pada Gambar 1.



Gambar 1. Gambar fungsi *window*

3. HASIL EKSPERIMEN

Ukuran FFT dapat dihitung pada spektrum biasanya adalah 128, 256, 512, 1024, 2048, atau 4096. Resolusi frekuensi paling tinggi pada spektrum adalah nilai cuplikan *digital* yang dibagi dengan ukuran FFT. FFT yang lebih besar hanya digunakan untuk analisis resolusi yang tinggi atau dengan skala frekuensi logaritmis. FFT dengan resolusi yang lebih tinggi membutuhkan waktu lebih banyak untuk menghitung spektrum. Untuk alasan ini, kadang-kadang sebaiknya nilai cuplikan dikurangi ketika merekam data *audio*/tutur, jika resolusi frekuensi yang tinggi dibutuhkan, daripada menggunakan suatu FFT dengan resolusi yang lebih tinggi.

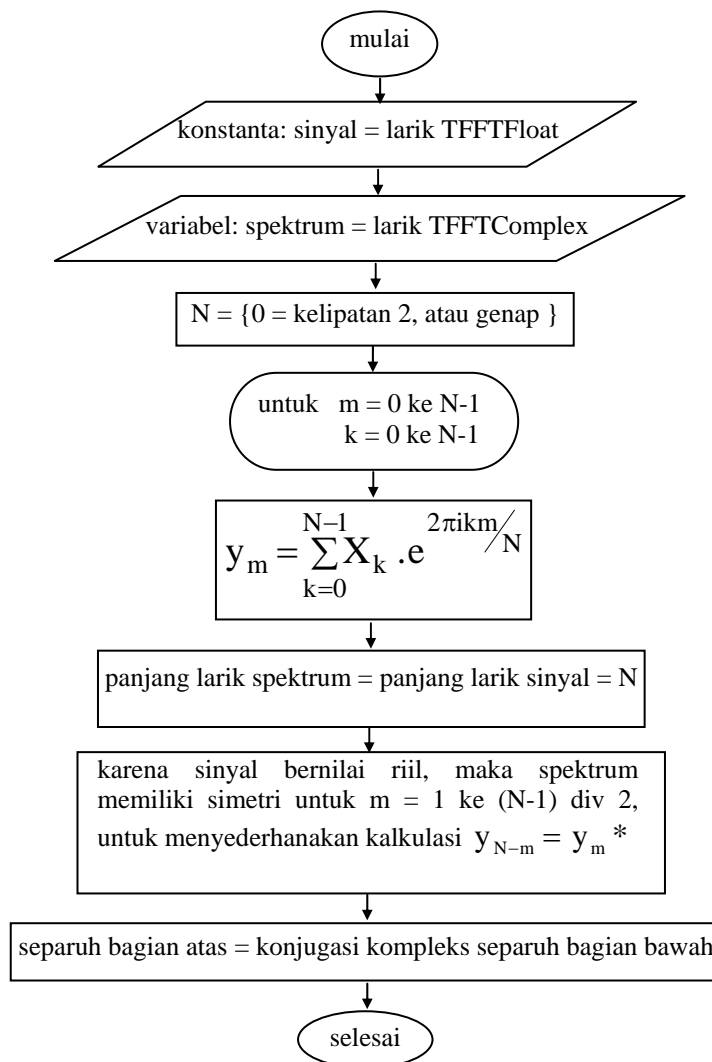
Algoritma dasar FFT berikut digunakan untuk menghitung spektrum y dari sinyal X (Diagram alir algoritma FFT dapat dilihat pada Gambar 2).

1. Mulai;
2. Konstanta : sinyal \leftarrow Larik TFFTFLOAT;
3. Variabel : spektrum \leftarrow Larik TFFTComplex;
4. Isikan data dengan nilai nol hingga ke kelipatan 2 berikutnya dapat dijadikan suatu pilihan bila kecepatan dianggap penting.
5. Untuk $m \leftarrow 0..N-1$, dan $k \leftarrow 0..N-1$;
6. Hitunglah spektrum y sinyal X :
$$y_m = \sum_{k=0}^{N-1} X_k \cdot e^{2\pi i k m / N}$$
7. Larik spektrum dan larik sinyal harus memiliki panjang yang sama N .
8. Algoritma ini berfungsi untuk setiap jumlah N , tetapi algoritma ini akan sangat cepat jika N adalah kelipatan 2, juga bekerja lebih cepat untuk N genap.
9. Karena sinyal masukan bernilai riil, spektrum memiliki simetri ini untuk $m \leftarrow 1..(N-1) \text{ div } 2$, yang mungkin berguna untuk menyederhanakan beberapa kalkulasi: $y_{N-m} = y_m^*$
10. Separuh bagian atas adalah konjugasi kompleks separuh bagian bawah.
11. Selesai.

Algoritma dasar IFFT dapat menghitung sinyal X dari spektrum y (Diagram alir algoritma IFFT dapat dilihat pada Gambar 3).

1. Mulai;
2. Konstanta ; spektrum \leftarrow larik TFFTComplex
3. Variabel : sinyal \leftarrow Larik TFFTFLOAT;

4. Algoritma ini berfungsi untuk setiap jumlah N, namun sangat cepat jika N adalah kelipatan 2, juga bekerja lebih cepat bahkan untuk N genap.
5. Untuk $m \leftarrow 0..N-1$, dan $k \leftarrow 0..N-1$,
6. Hitunglah sinyal x spektrum y :
$$X_k = \frac{1}{N} \sum_{m=0}^{N-1} y_m \cdot e^{-2\pi i k m / N}$$
7. Larik spektrum dan larik sinyal harus memiliki panjang yang sama N.
8. Algoritma dasar FFT menghasilkan suatu sinyal bernilai kompleks tetapi hanya bagian riil yang disalin ke keluaran larik sinyal, sementara bagian imajiner diabaikan.
9. Terdapat nol di mana spektrum memiliki simetri untuk $m \leftarrow 1..(N-1) \text{ div } 2$: $y_{N-m} = y_m^*$
10. Separuh bagian atas adalah konjugasi kompleks separuh bagian bawah.
11. Selesai.

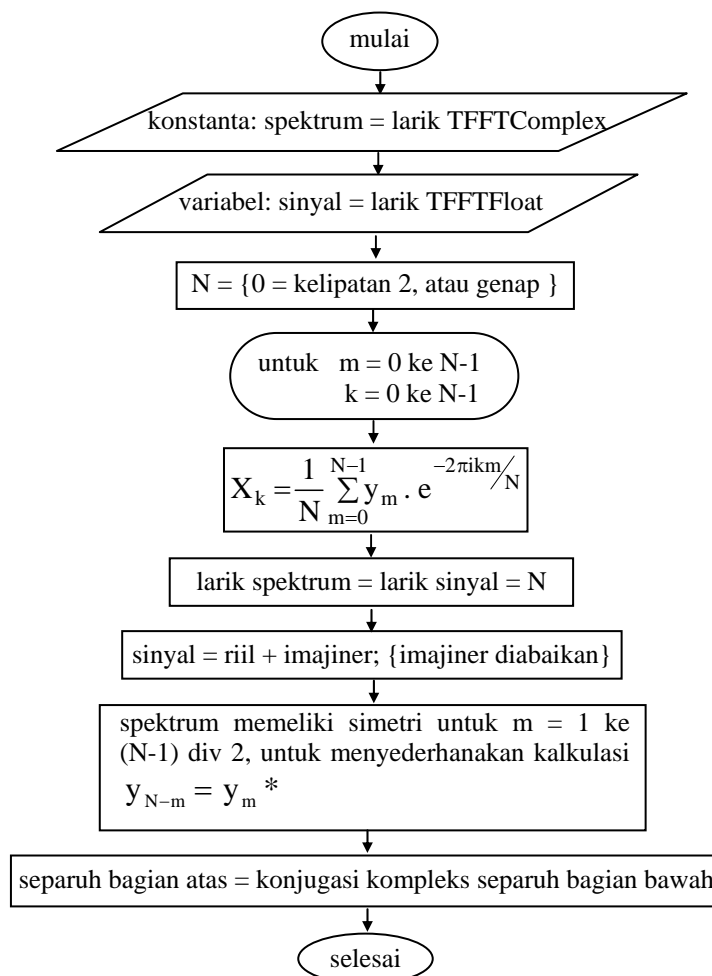


Gambar 2. Diagram alir algoritma dasar FFT

Skala frekuensi pada spektrum ini menggunakan skala frekuensi linier untuk menghitung suatu spektrum. Suatu skala linier akan menyebarkan komponen frekuensi secara merata pada spektrum keseluruhan. Suatu sinyal akan menghasilkan ketajaman, karena rutin FFT menganggap masukan sebagai suatu sinyal periodik yang dapat menimbulkan cacat pada

spektrum. Untuk mengurangi masalah tersebut suatu *window* data pertama-tama dapat diterapkan pada sinyal.

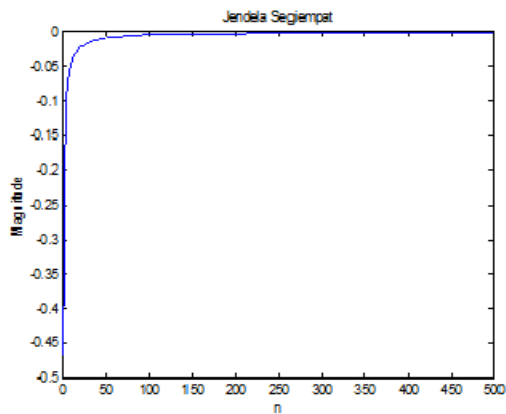
Window data digunakan ketika memproses larik terhingga diskrit, yaitu jumlah cuplikan N tetap. FFT mengasumsikan bahwa larik N -cuplikan bersifat periodik; jika tidak, maka transformasi yang dihasilkan dapat memperlihatkan kandungan frekuensi salah (*false*) yang dikenal sebagai kebocoran (*leakage*) karena diskontinuitas pada bagian tepi larik. Fungsi *window* memiliki amplitudo nol, atau mendekati nol pada bagian "tepi" larik, dan dapat digunakan untuk memodulasi masukan tak-periodis untuk memastikan bahwa fungsi produk bersifat periodis selama N cuplikan. Sebagai bukti dari apa yang disebut di atas, terdapat banyak fungsi *window* yang dapat dipilih, antara lain berupa *window* segiempat, Segitiga (Bartlett, atau Perzen), Hamming, Hanning, Blackman, Welch, FlatTop, dan Nuttall. Persamaannya tertera pada Tabel 1, dan *window* disajikan pada Gambar 4. Lebar *window* dapat dipilih kurang dari N , yang memungkinkan analisis terhadap spektrum fungsi *window* itu sendiri. Penerapan fungsi *window* terhadap data yang ada akan dapat membantu mengurangi efek kebocoran dalam domain frekuensi.



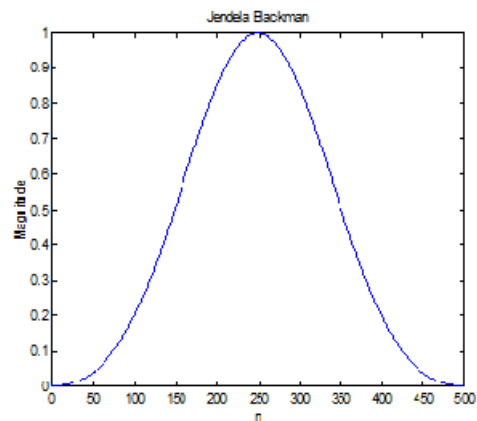
Gambar 3. Diagram alir algoritma dasar IFFT

Tabel 1. Persamaan *window* yang diterapkan pada spektrum

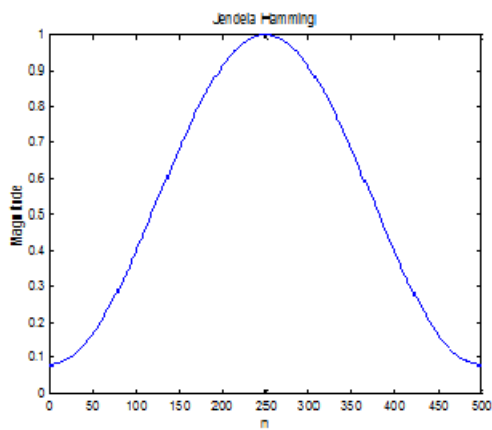
No	Tipe	Nilai cuplikan	Kisaran k
1	Segiempat	$w_k = \begin{cases} 1 \\ 0 \end{cases}$	$0 \leq k \leq N-1$ else
2	Segitiga Bartlett , atau Parzen	$w_k = 1 - \left 1 - \frac{2k}{N-1} \right $	$0 \leq k \leq N-1$
3	Hanning	$w_k = \frac{1}{2} \left(1 - \cos \frac{2k\pi}{N-1} \right)$	$0 \leq k \leq N-1$
4	Hamming	$w_k = 0.54 - 0.46 \cos \frac{2k\pi}{N-1}$	$0 \leq k \leq N-1$
5	Blackman	$w_k = 0.42 + 0.5 \cos \frac{2k\pi}{N-1} + 0.08 \cos \frac{4k\pi}{N-1}$	$0 \leq k \leq N-1$
6	Welch	$W_{lech}(x, \tau) = \begin{cases} 1 - \left(\frac{x}{\tau} \right)^2 \\ 0 \end{cases}$	$ x < \tau$ else
7	FlatTop	$w_k = 0.2810638602 - 0.520897135 * \cos(x) +$ $0.1980389663 * \cos(2x)$ $x = 2\pi k / (N-1)$	$0 \leq k \leq N-1$
8	Nuttall	$w_k = 0.375 - 0.5 * \cos(x/(n-1)) + 0.125 * \cos(2x/(n-1))$ $x = 2\pi k / (N-1)$	$0 \leq k \leq N-1$



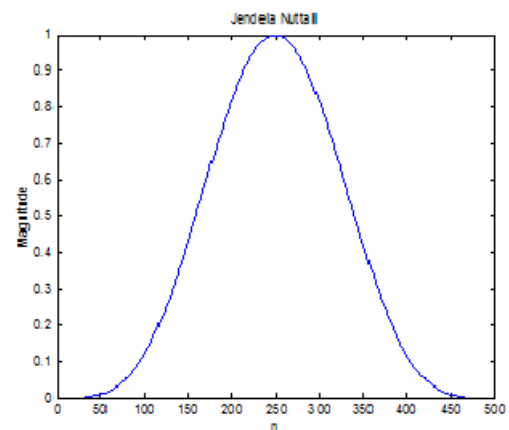
(a) Jendela Segiempat



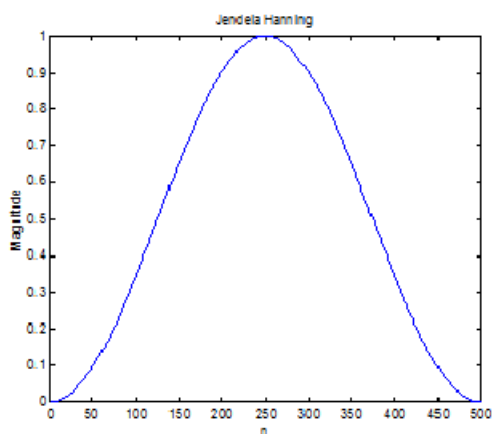
(b) Jendela Blackman



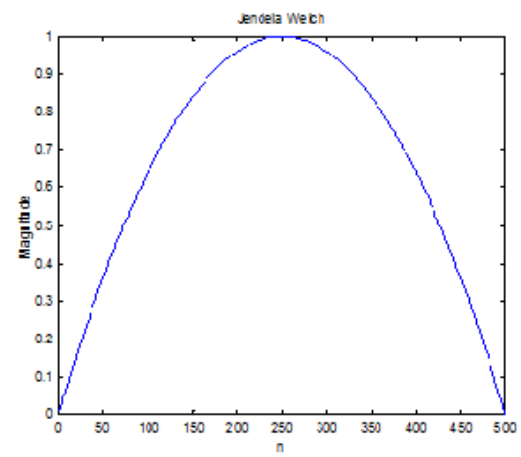
(c) Jendela Hamming



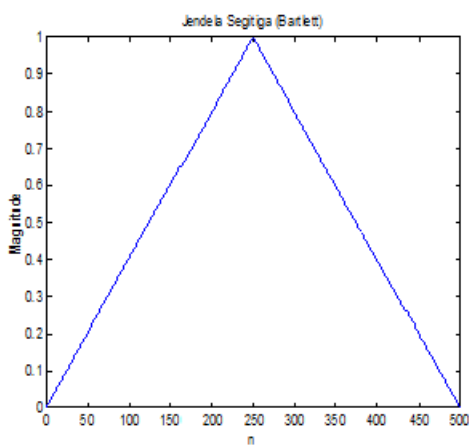
(d) Jendela Nuttall



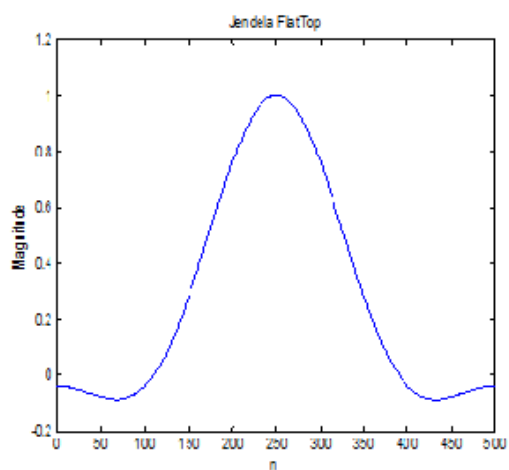
(e) Jendela Hanning



(f) Jendela Welch



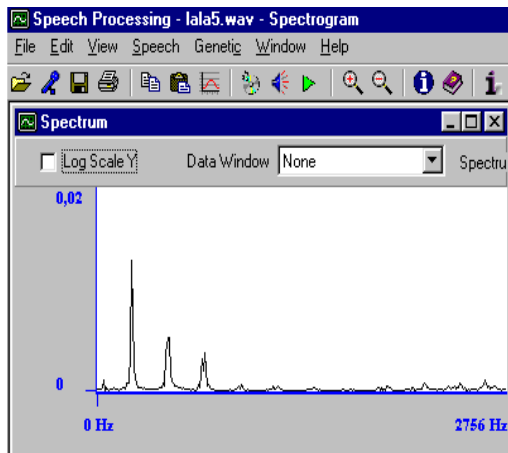
(g) Jendela Segitiga (Bartlett)



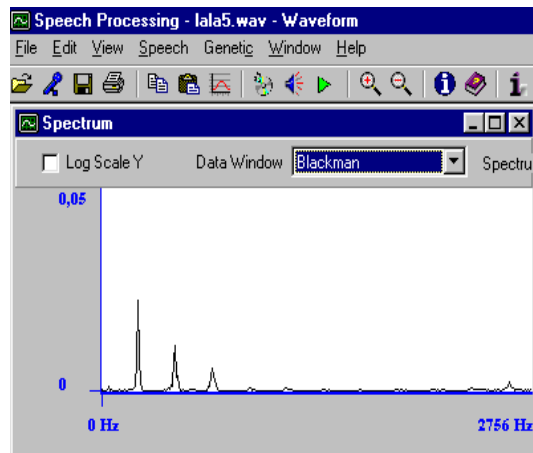
(h) Jendela FlatTop

Gambar 4. Berbagai jendela yang diterapkan pada spektrum

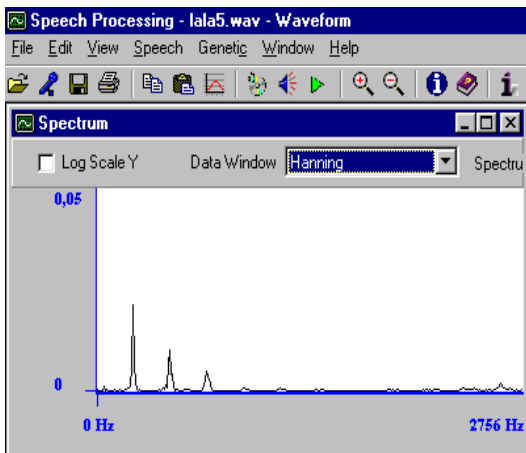
Spektrum dengan skala frekuensi linier dapat disajikan pada Gambar 5.



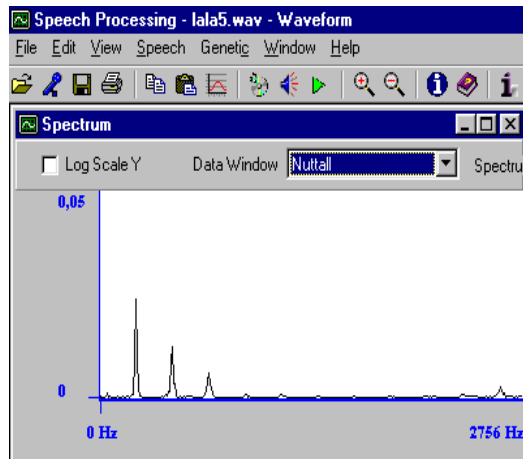
(a) Spektrum skala linier dengan *non-window*



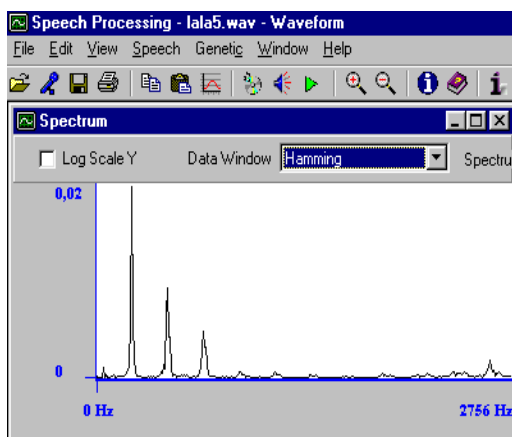
(b) Spektrum skala linier dengan *window* Blackman



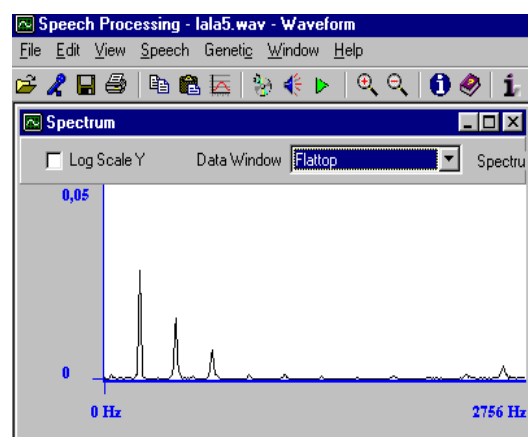
(c) Spektrum skala linier dengan *window* Hanning



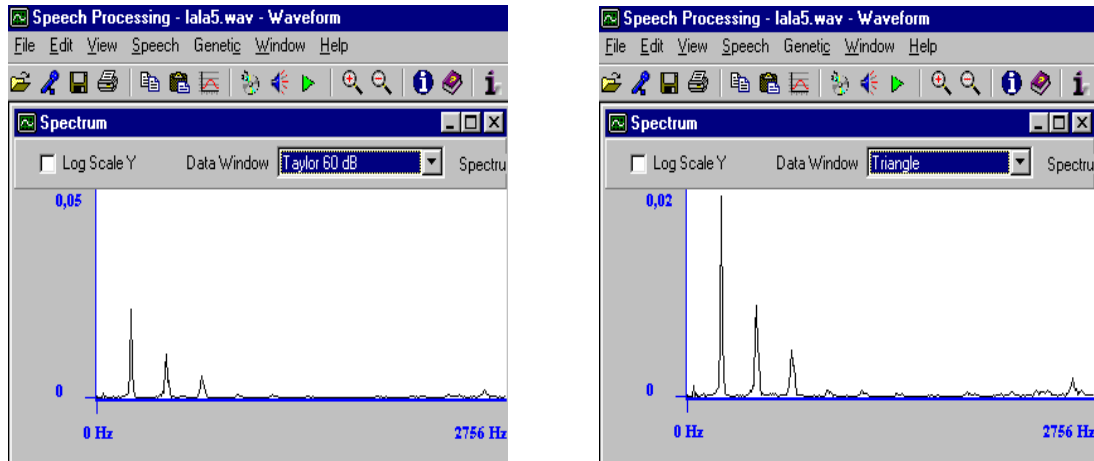
(d) Spektrum skala linier dengan *window* Nuttall



(e) Spektrum skala linier dengan *window* Hamming



(f) Spektrum skala linier dengan *window* Flattop



(g) Spektrum skala linier dengan *window* Taylor 80 dB (h) Spektrum skala linier dengan *window* Welch

Gambar 5. Spektrum dengan skala frekuensi linier

Pada Tabel 2 tertera keterangan yang dihasilkan ketika diterapkan beberapa *window* pada sinyal tutur.

Tabel 2. Keterangan efek *window* terhadap spektrum

No.	Nama <i>window</i>	Keterangan
1.	(tanpa <i>window</i>)	<i>Lobe</i> utama yang mungkin yang paling sempit, tetapi terjadi kebocoran yang cukup besar.
2.	Bartlett (<i>Triangle</i>)	<i>Welch</i> memiliki suatu <i>lobe</i> utama yang sedikit lebih sempit dibanding <i>Bartlett</i> , tetapi <i>sidelobe</i> -nya agak lebih tinggi.
3.	Welch (<i>parabola</i>)	<i>Welch</i> memiliki suatu <i>lobe</i> utama yang sedikit lebih sempit dibanding <i>Bartlett</i> , tetapi <i>sidelobe</i> -nya agak lebih tinggi.
4.	Hanning	<i>Lobe</i> utama yang agak sempit, reduksi kebocoran jauh asimptotik yang bagus.
5.	Hamming	<i>Lobe</i> utama yang agak sempit, reduksi kebocoran dekat yang bagus.
6.	Blackman	<i>Lobe</i> utama yang lebih luas dibanding <i>Hanning</i> dan <i>Hamming</i> , reduksi kebocoran dekat dan jauh yang bagus.
7.	Nuttall	Reduksi kebocoran dekat yang lebih bagus daripada <i>window Blackman</i> tetapi <i>lobe</i> utama yang lebih luas.
8.	FlatTop	<i>Window Flattop</i> digunakan ketika menghitung amplitude spektrum suatu puncak dengan energi spektrum mendekati aslinya dan menyediakan akurasi amplitude yang bagus. <i>Window Flattop</i> memiliki suatu <i>lobe</i> utama yang luas dengan suatu ' <i>flat top</i> ' melewati dua garis spektrum. Ini berarti bahwa setiap frekuensi sinyal yang berada di antara dua garis spektrum akan terlihat pada garis spektrum yang terdekat tanpa kehilangan amplitude. <i>Window FlatTop</i> digunakan ketika pengukuran komponen spektrum amplitude dari frekuensi arbitrer dianggap penting.

4. KESIMPULAN

Dari hasil penelitian dapat ditarik kesimpulan sebagai berikut:

1. Pemindaian (*scanning*) berkas gelombang atau masukan *audio*/tutur dapat berjalan lebih cepat ketika menggunakan suatu skala frekuensi linier dari pada menggunakan suatu skala frekuensi logaritmis.
2. Penerapan fungsi berbagai *window* terhadap data dapat membantu mengurangi efek kebocoran yang terjadi pada spektrum frekuensi.
3. Semakin besar jumlah poin data, semakin baik resolusi frekuensi pada spektrum.

4. Metode ini berjalan lebih cepat jika poin data merupakan kelipatan dua (128, 256, 1024, 2048, atau 4096, dan seterusnya), atau merupakan bilangan genap.
5. Memilih suatu resolusi frekuensi yang tepat dan resolusi waktu yang sesuai menjadi suatu kesesuaian antara kebutuhan untuk mengamati detail frekuensi yang baik dalam spektrum dengan kebutuhan untuk mengamati variasi waktu yang cepat dalam spektrum.

DAFTAR PUSTAKA

- [1] Rabiner, L. R., and R.W. Schafer, "***Digital Processing of Speech Signals***" Prentice Hall, 1978.
- [2] Sugamura, N., and F. Itakura, "***Speech Analysis and Synthesis Methodes Developed at ECL in NTT- from LPC to LSP***" *Speech Communication*, 5:199-215, 1986.
- [3] Fallside, F., and W.A. Woods, "***Computer Speech Processing***", Cambridge University Engineering Department, Printice Hall, 1985.