

IMPLEMENTASI ALGORITMA *FAST FOURIER TRANSFORM* DAN *MEAN SQUARE PERCENTAGE ERROR* UNTUK MENGHITUNG PERUBAHAN SPEKTRUM SUARA SETELAH MENGGUNAKAN FILTER *PRE-EMPHASIS*

Fitri Mintarsih¹, Rizal Bahaweres², Ricky Aditya³

^{1,2,3}Teknik Informatika, Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

mintarsihf@gmail.com, rizalbroer@gmail.com, riickyaditya26@gmail.com

ABSTRAK

Suara saat ini banyak digunakan sebagai salah satu media atau alat input untuk melakukan perintah terhadap suatu aplikasi atau alat, seperti SIRI dan *ok google*. Dalam melakukan pengolahan suara dibutuhkan filter untuk membuat suara menjadi lebih baik, sehingga memudahkan dalam proses pengenalan suara. Salah satu filter yang digunakan dalam pengolahan suara adalah filter *pre-emphasis*. Menerapkan *pre-emphasis* jelas/terbukti mengubah (membuat distorsi) pada spektrum suara [1] Suara hasil filter akan berbeda dari suara *original*, namun seberapa besar perbedaan (perubahan) suara hasil filter *pre-emphasis* dari suara *original*? Untuk mengetahui seberapa besar perbedaan (perubahan) tersebut, penulis menggunakan Algoritma *Fast Fourier Transform* (FFT) untuk mendapatkan ekstraksi ciri (*magnitude*) dari spektrum suara yang difilter dan suara *original*, serta menggunakan *Mean Square Percentage Error* (MSPE) untuk menghitung perbedaan/perubahan spektrum suara dalam persen. *Mean Square Percentage Error* (MSPE) merupakan bentuk persen dari *Mean Square Error* (MSE). MSPE memiliki properti yang sama dengan MSE, hanya saja ditampilkan dalam bentuk persen. [2]

Kata Kunci : Filter *Pre-emphasis*, *Fast Fourier Transform*, *Mean Square Error*, *Mean Square Percentage Error*

1. PENDAHULUAN

1.1. Latar Belakang

Filter adalah suatu rangkaian elektronik yang berfungsi untuk mengolah frekuensi dari suatu sinyal, frekuensi tersebut dapat diloloskan atau diredam, hal ini disesuaikan dengan kebutuhan. Berdasarkan sifat ini filter dibedakan menjadi *low pass filter* (LPF), *high pass filter* (HPF), *band pass filter* (bpf), dan *band reject filter* (BRF). [3]

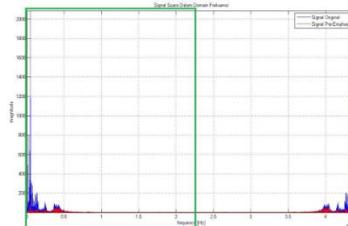
Menurut Shorty Stouffer pada *paper Explanation of "Flat Audio"*, suara (*speech*) terdiri dari (mengandung) beberapa frekuensi audio, kebanyakan berada pada frekuensi antara 125 sampai 4000 Hz. Tapi suara manusia (*human speech*) semua frekuensi tidak tercipta secara sama rata.

Suara yang dihasilkan dari mulut akan kehilangan informasi pada frekuensi tinggi, karena itu dibutuhkan proses *pre-emphasis* untuk mengkompensasi (mengatasi) kehilangan pada frekuensi tinggi. Operasi untuk menyelamatkan informasi pada frekuensi tinggi tersebut adalah *pre-emphasis*. [4]

Filter *pre-emphasis* meningkatkan energi suara sejumlah variabel tertentu dimana meningkat seiring frekuensi meningkat, variabel tersebut biasa didefinisikan sebagai *alpha*. Menerapkan *pre-emphasis* jelas/terbukti mengubah (membuat distorsi) pada spektrum suara. [1]

Filter *pre-emphasis* banyak digunakan sebagai tahap awal (*pre-processing*) pada proses pengolahan

suara. Untuk mengetahui seberapa besar suara hasil filter *pre-emphasis* berubah dari aslinya. Penulis menggunakan algoritma *Fast Fourier Transform* dan *Mean Square Percentage Error* untuk menghitung perbedaan/perubahan suara hasil filter *pre-emphasis* terhadap suara asli (*original*).



Gambar 1.1 : Spektrum Suara *Original* Dan Filter *Pre-emphasis*

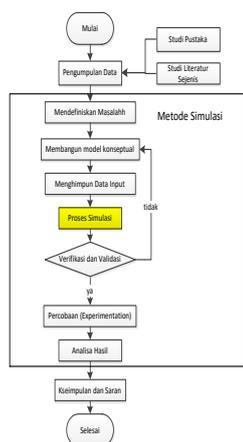
1.2. Rumusan Masalah

Rumusan masalah yang dibahas dalam penelitian ini, sesuai dengan apa yang telah dikemukakan pada bagian latar belakang adalah:

1. Bagaimanakah pengaruh filter *pre-emphasis* terhadap spektrum (sinyal) suara?
2. Bagaimana menghitung besar perubahan spektrum suara setelah dilakukan filter *pre-emphasis* (seberapa besar spektrum suara berubah dari aslinya)?
3. Bagaimana audio (suara) hasil filter *pre-emphasis*?

2. METODOLOGI PENELITIAN

Metode penelitian dalam penulisan ini merujuk pada metode simulasi Madani (2010) [5]. Berikut adalah alur penelitian yang penulis lakukan:



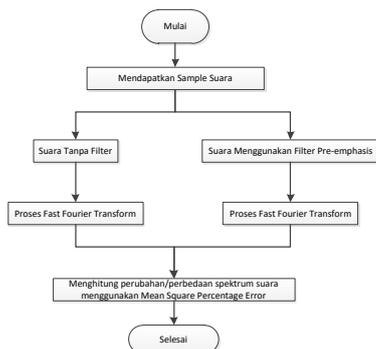
Gambar 3. 1 : Alur Penelitian

Studi pustaka dan studi literatur sejenis dilakukan untuk mendapatkan informasi terkait topik penelitian.

Membangun model konseptual dilakukan untuk menentukan langkah-langkah percobaan dalam proses simulasi untuk menjawab permasalahan yang ditentukan.

Menghimpun data *input*, tahap ini dilakukan untuk mendapatkan data *input* (masukkan) yang digunakan dalam simulasi.

Dalam proses simulasi terdapat beberapa proses sebagai berikut:



Gambar 3. 2 : Rincian Alur Proses Simulasi

3. PEMBAHASAN

4.1. Mendapatkan *Sample*

Penulis melakukan perekaman suara untuk digunakan sebagai *input* pada aplikasi Android yang penulis buat. Dikarenakan jumlah maksimal *sample* yang dapat diproses secara sekaligus pada proses *Fast Fourier Transform* yang digunakan pada *smartphone* Android dalam penelitian ini adalah 65536 (bilangan *power of 2*), sehingga panjang durasi suara dibatasi sesuai dengan *sample rate* yang digunakan agar tidak melebihi jumlah *sample* yang dapat diproses pada FFT. Oleh sebab itu, maka

penulis hanya mengucapkan satu buah kata karena durasinya cukup singkat.

Perekaman untuk proses simulasi yang penulis lakukan menggunakan *sample rate* 44100 Hz dengan durasi 1,4 detik dengan mengucapkan kata 'Sekolah'. Dari hasil suara rekaman diperoleh *sample* sebanyak 61740 *sample* (*sample rate* * durasi). *Sample* ini akan diproses. Suara hasil rekaman dalam bentuk mentah (*original*) juga disimpan menjadi file *wav*, hal ini dilakukan untuk dijadikan sebagai data *input* pada proses verifikasi menggunakan Matlab.



Gambar 4. 1 : Perekaman Suara Menggunakan Aplikasi Android

Berikut adalah data sebagian *sample* yang diperoleh dari hasil rekaman menggunakan Android:

Tabel 4. 1 : *Sample* Sinyal Suara *Original* Dalam Domain Waktu

Indeks	<i>Sample</i> Sinyal Suara <i>Original</i>
0 s/d 11999	0,0
12000	0,008941650390625
12001	0,00714111328125
12002	0,0074462890625
12003	0,0084228515625
12004	0,00946044921875
12005	0,0089111328125
12006	0,008331298828125
12007	0,007965087890625
12008	0,0064697265625

Terlihat bahwa *sample*[0] sampai *sample*[11999] bernilai 0,0 dan baru pada *sample*[12000] memiliki nilai.

4.2. Proses Filter *Pre-emphasis*

Pada tahap ini penulis melakukan proses filter *pre-emphasis* terhadap *sample* yang sudah direkam sebelumnya untuk mendapatkan *sample* kedua (variabel data kedua). Sehingga terdapat dua data, yaitu variabel 1 berupa data suara sebelum filter (*original*/asli) dan variabel 2 berupa data suara setelah dilakukan filter *pre-emphasis*. Filter *pre-emphasis* digunakan tepat setelah dilakukan proses mendapatkan *sample* suara, dengan kata lain proses

filter *pre-emphasis* dilakukan terhadap suara pada domain waktu.

Berikut adalah sebagian *sample* yang telah dilakukan filter *pre-emphasis*:

Tabel 4. 2 : Sample Sinyal Suara Filter *Pre-emphasis* Dalam Domain Waktu

Indeks	Sample Sinyal Suara Filter <i>Pre-emphasis</i>
0 s/d 11999	0,0
12000	0,008941650390625
12001	-0,0015322875976562494
12002	5.194091796875003E-4
12003	0,0011999511718749998
12004	0,001290283203125001
12005	-2,6550292968749965E-4
12006	-3,125000000000003E-4
12007	-1,1627197265625007E-4
12008	-0.0012564086914062496

Sama seperti sinyal *original*, terlihat bahwa *sample*[0] sampai *sample*[11999] bernilai 0,0 dan baru pada *sample*[12000] memiliki nilai.

4.3. Proses *Fast Fourier Transform*

Dikarenakan Algoritma FFT yang penulis gunakan pada aplikasi *smartphone* Android mensyaratkan jumlah *sample* harus berupa bilangan *power of 2*, maka *sample* yang didapat (sinyal suara *original* dan hasil filter) dimasukkan ke dalam sebuah variabel yang memiliki panjang (*size*) bilangan *power of 2* supaya bisa diproses oleh Algoritma FFT. Sinyal suara *original* & filter yang digunakan pada proses FFT adalah keseluruhan sinyal (*whole signal*) tanpa dibagi menjadi beberapa segmen.

Sample suara *original* dan *sample* suara hasil filter *pre-emphasis* masing – masing diproses dengan algoritma FFT untuk dicari nilai *magnitude*-nya. Hasil dari algoritma FFT tidak langsung *magnitude*, akan tetapi berupa bilangan *complex* (*real* dan *imaginary*). Bilangan *complex* tersebut yang digunakan untuk memperoleh *magnitude* dari masing-masing *sample*. Pada saat akan melakukan transformasi dari domain waktu ke domain frekuensi menggunakan algoritma *Fast Fourier Transform*, *sample* suara hasil rekaman (sinyal suara *original*) disusun ulang terlebih dahulu dengan pengelompokkan genap – ganjil (berdasarkan indeks elemen).

4.3.1. Proses FFT sinyal *original*

Berikut adalah *sample* suara *original* yang telah disusun ulang urutannya (*re-ordered input*).

Tabel 4. 3 : Sinyal Suara *Original* Setelah *Re-ordered Input*

Indeks	Sample Sinyal Suara <i>Original</i> Setelah <i>Re-ordered Input</i>
0	0,0
1	-0,35968017578125
2	0,00128173828125

3	0,1199951171875
4	0,0
5	0,356201171875
6	-0,004302978515625
7	0,020782470703125
8	0,0
9	-0,132843017578125

Terlihat bahwa urutan *sample* berbeda dari sinyal suara sebelum disusun ulang. Dimana pada sinyal suara yang sebelum disusun ulang memiliki nilai 0,0 dari indeks ke 0 sampai dengan indeks 11999, pada sinyal yang telah disusun ulang nilai 0,0 tidak lagi berurutan dari indeks ke 0 sampai dengan indeks 11999.

Setelah dilakukan proses perhitungan *Fast Fourier Transform*, dihasilkan *output* (hasil) dari algoritma FFT berupa bilangan *complex* (bilangan yang terdiri dari *real* & *imaginary*) dari sinyal suara *original* sebagai berikut.

Tabel 4. 4 : *Output* FFT Sinyal Suara *Original*

Indeks	Sinyal Suara <i>Original</i> Setelah FFT
0	24,144439697265625
1	-31,108949101376364 + 27,686524029723977i
2	22,845525524011688 – 31,12293471748543i
3	4,194115465643595 + 6,253078968557782i
4	6,243474587219134 + 5,9310431824313605i
5	-23,921595302975895 + 2,392459956416322i
6	33,700433164852534 + 2,3156203404207i
7	-31,682530314901907 + 20,019162920723563i
8	5,635633132952314 – 31,379415318453482i
9	5,167542707869074 + 23,48089868179609i

Dari bilangan *complex* tersebut dapat diperoleh *magnitude* spektrum suara *original*. Berikut adalah sebagian *magnitude* spektrum suara *original* dari indeks 0 sampai indeks 9 (pada awal frekuensi):

Tabel 4. 5 : *Magnitude* Suara *Original* Pada Frekuensi Awal

Indeks	<i>Magnitude</i> Spektrum Suara <i>Original</i>
0	24,144439697265625
1	41,64505165131277
2	38,60770780423424
3	7,529382519580901
4	8,611518341913733
5	24,04093564074439

6	33,779894805337435
7	37,47732129169788
8	31,88146901473053
9	24,04279726950307

Berikut adalah *magnitude* sinyal suara *original* pada frekuensi yang lebih tinggi, sekitar pada frekuensi 21264 Hz atau *magnitude* dari data (*sample*) ke 31600 sampai 31609.

Tabel 4. 6 : *Magnitude* Suara *Original* Pada Frekuensi Sekitar 21264 Hz

Indeks	<i>Magnitude</i> Spektrum Suara <i>Original</i>
31600	0,17859130241194462
31601	0,30856901537081144
31602	0,328897978313349
31603	0,2730524459256943
31604	0,16071020977579317
31605	0,03072451459807093
31606	0,1335714837433994
31607	0,2004441301469685
31608	0,15259457509028215
31609	0,16679296123320603

4.3.2 Proses FFT sinyal filter *pre-emphasis*

Proses *Fast Fourier Transform* pada sinyal hasil filter *pre-emphasis* sama seperti sinyal *original*. Berikut adalah *sample* suara filter *pre-emphasis* yang telah disusun ulang urutannya (*re-ordered input*), *output* fft (bilangan *complex*), serta *magnitude* spektrum suara filter *pre-emphasis* pada frekuensi awal dan *magnitude* spektrum suara filter *pre-emphasis* pada frekuensi yang lebih tinggi, sekitar pada frekuensi 21264 Hz atau *magnitude* dari data (*sample*) ke 31600 sampai 31609.

Tabel 4. 7 : Sinyal Suara Filter *Pre-emphasis* Setelah *Re-ordered Input*

Indeks	<i>Sample</i> Sinyal Suara Filter <i>Pre-emphasis</i> Setelah <i>Re-ordered Input</i>
0	0,0
1	-0,014905090332031246
2	-0,00392822265625
3	-4,260253906250011E-4
4	0,0
5	0,137264404296875
6	9,957885742187499E-4
7	-0,0018334960937499978
8	0,0
9	-0,031130371093749998

Tabel 4. 8 : *Output* FFT Sinyal Suara Filter *Pre-emphasis*

Indeks	Sinyal Suara Filter <i>Pre-emphasis</i> Setelah FFT
0	0,7331842041015706
1	-0,927572095961676 + 0,830853359002095i

2	0,69776297150146 – 0,9235510034220853i
3	0,1281583153728862 + 0,19661769394796175i
4	0,18611444363930382 + 0,18904673171239594i
5	-0,7209432313965173 + 0,06922909827622989i
6	1,0046353868465396 + 0,09551369503329263i
7	-0,9708497079198397 + 0,5849084582786166i
8	0,18379854034254678 – 0,9351799417225395i
9	0,12660329085969435 + 0,7075794752806228i

Tabel 4. 9 : *Magnitude* Suara Filter *Pre-emphasis* Pada Frekuensi Awal

Indeks	<i>Magnitude</i> Spektrum Suara Filter <i>Pre-emphasis</i>
0	0,7331842041015706
1	1,2452739848610832
2	1,1575057755020006
3	0,23469782992739943
4	0,2652871140901989
5	0,72425949144253
6	1,009165559481302
7	1,1334315417941687
8	0,9530705256339759
9	0,7188164627322512

Tabel 4. 10 : *Magnitude* Suara Filter *Pre-emphasis* Pada Frekuensi Sekitar 21264 Hz

Indeks	<i>Magnitude</i> Spektrum Suara Filter <i>Pre-emphasis</i>
31600	0,3424737078372963
31601	0,6126191531307301
31602	0,6490756577949269
31603	0,5306857586837651
31604	0,32494780536773676
31605	0,06207551498101116
31606	0,26816885729606843
31607	0,3856525140089778
31608	0,30071601003846515
31609	0,3341233264812005

Dari nilai *magnitude* yang diperoleh spektrum suara *original* dan spektrum suara hasil filter *pre-emphasis*, terlihat bahwa:

- Nilai *magnitude* spektrum suara hasil filter *pre-emphasis* lebih rendah dari *magnitude* spektrum suara *original* pada frekuensi rendah (frekuensi awal).
- Nilai *magnitude* spektrum suara hasil filter *pre-emphasis* lebih tinggi (lebih kuat) dari *magnitude* spektrum suara

original pada frekuensi yang lebih tinggi (*higher frequency*).

4.4. Menghitung Perubahan/Perbedaan Spektrum Suara Menggunakan Mean Square Percentage Error

Setelah melakukan proses FFT serta mendapatkan nilai *magnitude* dari *sample original* dan *sample* suara hasil filter *pre-emphasis*, proses selanjutnya adalah menghitung besar perubahan/perbedaan dari spektrum suara. *Magnitude* dari *sample original* dan *sample* filter *pre-emphasis* dihitung perbedaannya menggunakan *Mean Square Percentage Error*, hasil dari perhitungan ini akan menunjukkan besar perubahan spektrum suara hasil filter *pre-emphasis* dari spektrum suara *original* (aslinya) dalam bentuk persen. Semakin besar nilai *Mean Square Percentage Error*, maka semakin besar pula perbedaan (perubahan) spektrum suara hasil filter *pre-emphasis* dari spektrum suara *original* (asli). Hasil perhitungan menggunakan Android sebesar 43,34224132317319%.



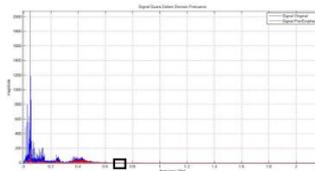
Gambar 4. 2 : Besar Perubahan Spektrum Suara Setelah Menggunakan Filter *Pre-emphasis*

4.5. Verifikasi

Penulis menggunakan Matlab untuk memverifikasi hasil perhitungan perubahan spektrum suara setelah menggunakan filter *pre-emphasis* yang telah dilakukan menggunakan *smartphone* Android. Matlab dipilih dikarenakan pada proses *Fast Fourier Transform* di Matlab tidak mensyaratkan *sample* yang digunakan harus berupa bilangan *power of 2*, sedikit berbeda dengan yang digunakan pada proses *Fast Fourier Transform* di *smartphone* Android.

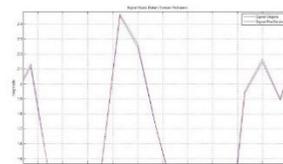
Dalam melakukan verifikasi, penulis menggunakan file *wav* yang merupakan hasil rekaman suara tanpa filter (*original*) dari aplikasi *smartphone* Android sebagai data *input* pada Matlab. File *wav* yang digunakan adalah pengucapan kata ‘Sekolah’ dengan menggunakan *sample rate* 44100 Hz dan memiliki durasi 1,4 detik, kemudian dilakukan langkah-langkah seperti pada proses simulasi. Berikut adalah gambar pola spektrum suara *original* (warna biru) dan pola

spektrum suara hasil filter *pre-emphasis* (warna merah).



Gambar 4. 3 : Sinyal Suara ‘Sekolah’ Dalam Domain Frekuensi

Spektrum tersebut di-*zoom* pada frekuensi sekitar 7520Hz.



Gambar 4. 4 : Sinyal Suara ‘Sekolah’ Dalam Domain Frekuensi Pada Frekuensi Sekitar 7520Hz

Dari gambar spektrum tersebut terlihat bahwa pada frekuensi lebih tinggi (sekitar frekuensi $\pm 7520\text{Hz}$), nilai *magnitude* suara hasil filter *pre-emphasis* lebih besar (kuat) dibandingkan *magnitude* suara *original*. Terlihat grafik berwarna merah (sinyal suara hasil filter *pre-emphasis*) sudah sedikit lebih tinggi dari grafik berwarna biru (sinyal suara *original*).

Berikut adalah beberapa *sample magnitude* yang dimulai dari *sample* ke 10526 (posisi frekuensi sekitar 7520 Hz).

Variables - mag		Variables - magPre	
mag	magPre	mag	magPre
10526	0.3516	10526	0.3620
10527	0.1743	10527	0.1829
10528	0.6796	10528	0.6861
10529	1.9014	10529	1.9070
10530	2.1133	10530	2.1330
10531	1.4152	10531	1.4154
10532	0.6839	10532	0.6963
10533	0.3953	10533	0.4059
10534	1.4707	10534	1.4769
10535	2.4545	10535	2.4672
10536	2.2456	10536	2.2683
10537	1.7149	10537	1.7194
10538	1.2766	10538	1.2850
10539	0.7567	10539	0.7696
10540	0.6422	10540	0.6429

Gambar 4. 5 : Perbandingan *Magnitude Original* (Kiri) dan Filter (Kanan) dari *Sample* Ke 10526

Berikut adalah hasil perhitungan di Matlab menunjukkan spektrum suara hasil filter *pre-emphasis* berubah (berbeda) sebesar 43,457309049774835% dari spektrum suara *original*.

Variables - MSPE	
MSPE	
1	43.457309049774835
2	
3	

Gambar 4. 6 : Hasil Verifikasi Perhitungan Perubahan/Perbedaan Spektrum Suara di Matlab

Hasil ini hampir sama dengan perhitungan di *smartphone* Android, dimana hasilnya sebesar 43,34224132317319%. Hasil perhitungan di Matlab dan *smartphone* Android sedikit berbeda dikarenakan Algoritma *Fast Fourier Transform* yang digunakan pada Matlab berbeda dengan yang digunakan pada *smartphone* Android.

Dari hasil verifikasi ini dapat dikatakan bahwa filter *pre-emphasis* membuat spektrum suara berubah sebesar $\pm 43\%$ dari suara *original* (asli)

4.6. Validasi

Validasi dilakukan dengan melakukan pengecekan terhadap audio hasil filter *pre-emphasis*. Audio hasil filter *pre-emphasis* dikatakan valid apabila audio yang dihasilkan sesuai dengan teori. Dalam proses validasi ini penulis menampilkan spektrogram dari audio *original* dan audio hasil filter *pre-emphasis*, untuk memperlihatkan secara visual perubahan yang terjadi terhadap spektrum suara (*human speech*). Untuk menampilkan spektrogram, penulis menggunakan *software* Spek – *Acoustic Spectrum Analyser*.



Gambar 4. 7 : Spektrogram ‘Sekolah2.wav’ (Original)



Gambar 4. 8 : Spektrogram ‘Sekolah2 filter.wav’

Terlihat bahwa spektrum suara hasil filter *pre-emphasis* lebih merata dan lebih bersih daripada spektrum suara *original*. Serta suara hasil filter *pre-emphasis* terdengar lebih halus dan jelas dibandingkan suara *original*.

4.7. Percobaan

4.7.1. Skenario *sample rate* 44100 Hz durasi 1,4 detik dengan pengucapan kata ‘Sekolah’.

Berikut adalah hasil percobaan :

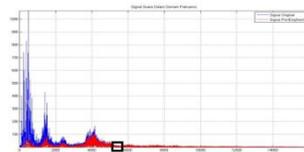
Tabel 4. 11 : Besar Perubahan Spektrum Suara Skenario 44100 Hz

Percobaan	Besar Perubahan Spektrum Suara	
	Hasil Android	Hasil Verifikasi Matlab
1	43,50659685901288%	43,344661559439840%
2	43,34224132317319%	43,457309049774835%
3	43,18713368487691%	43,176459210975935%
4	43,24802483090293%	43,281345201021566%
5	43,22345853766464%	43,229458277808230%

6	43,39835360236998%	43,365001242810920%
7	43,153758460130746%	43,205052324003160%
8	43,84244591104133%	44.093416445187430%

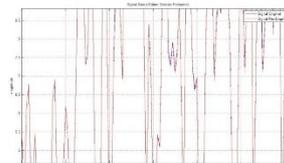
4.7.2 Skenario *sample rate* 32000 Hz durasi 2 detik dengan pengucapan kata ‘Perpustakaan’.

Gambar spektrum suara *original* (biru) dan filter *pre-emphasis* (merah).



Gambar 4. 9 : Sinyal Suara ‘Perpustakaan’ Percobaan 1 Domain Frekuensi

Gambar spektrum tersebut di-zoom pada frekuensi sekitar 5428Hz.



Gambar 4. 10 : Sinyal Suara ‘Perpustakaan’ Percobaan 1 Domain Frekuensi Pada Frekuensi Sekitar 5428Hz

Terlihat bahwa pada frekuensi sekitar ± 5428 Hz, *magnitude* (energi) pada suara hasil filter *pre-emphasis* sudah lebih besar daripada *magnitude* suara *original*.

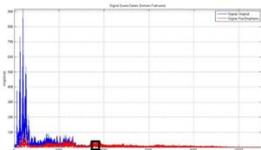
Berikut adalah hasil percobaan :

Tabel 4. 12 : Besar Perubahan Spektrum Suara Skenario 32000 Hz

Percobaan	Besar Perubahan Spektrum Suara	
	Hasil Android	Hasil Verifikasi Matlab
1	43,11906606519568%	43,116019974944685%
2	43,30843151217772%	43,292914665528870%
3	43,17440342202%	43,166896909572920%
4	43,77203149606681%	44,028342773470050%
5	43,128647741091164%	43,124832178840450%
6	43,12084841112952%	43,125072097954650%
7	43,55883701370916%	43,202035660168164%
8	43,19437274151283%	43,258871822035770%

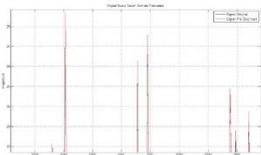
4.7.3. Skenario *sample rate* 22050 Hz durasi 2 detik dengan pengucapan kata ‘Universitas’.

Gambar spektrum suara *original* (biru) dan filter *pre-emphasis* (merah).



Gambar 4. 11 : Sinyal Suara 'Universitas'
Percobaan 1 Domain Frekuensi

Gambar spektrum tersebut di-zoom pada frekuensi sekitar 3740Hz.



Gambar 4. 12 : Sinyal Suara 'Universitas'
Percobaan 1 Domain Frekuensi Pada Frekuensi
Sekitar 3740Hz

Terlihat bahwa pada frekuensi sekitar ± 3740 Hz, *magnitude* (energi) pada suara hasil filter *pre-emphasis* sudah lebih besar daripada *magnitude* suara *original*.

Berikut adalah hasil percobaan :

Tabel 4. 13 : Besar Perubahan Spektrum Suara
Skenario 22050 Hz

Percobaan	Besar Perubahan Spektrum Suara	
	Hasil Android	Hasil Verifikasi Matlab
1	43,26405859550806%	43,226459971361436%
2	43,11825850839186%	43,118834075954716%
3	43,23195718032391%	43,997552667878130%
4	43,116233044426295%	43,116285942835950%
5	43,17068524647999%	43,148481821512790%
6	43,13811877604813%	43,137525252883450%
7	43,12695733122509%	43,148074634683820%
8	43,23842549000237%	43,287245429874430%

4.8. Analisa Hasil

Hasil percobaan tersebut menunjukkan *sample rate* mempengaruhi posisi frekuensi tinggi (*high frequency*) dimana energi (*magnitude*) hasil filter *pre-emphasis* lebih besar dari energi (*magnitude*) *original*, walaupun konstanta filter *pre-emphasis* α sama ($\alpha=0,97$). Pada *sample rate* 44100Hz *magnitude* hasil filter lebih besar dari *magnitude original* pada frekuensi sekitar ± 7520 Hz, pada *sample rate* 32000Hz *magnitude* lebih besar pada frekuensi sekitar ± 5428 Hz, dan pada *sample rate* 22050Hz *magnitude* lebih besar pada frekuensi sekitar ± 3740 Hz.

Dalam penelitian ini juga menunjukkan bahwa filter *pre-emphasis* dengan konstanta $\alpha=0,97$ memberikan perubahan cukup besar terhadap suara, suara hasil filter *pre-emphasis* berubah sekitar $\pm 43\%$ dari suara *original* (asli). Besar perubahan itu konsisten (hampir sama) walaupun menggunakan *sample rate* yang berbeda dan durasi berbeda.

4. KESIMPULAN

Dari pembahasan yang sudah diuraikan sebelumnya, dapat diambil kesimpulan bahwa:

1. Filter *pre-emphasis* membuat energi (*magnitude*) lebih besar pada frekuensi tinggi dan melemahkan energi pada frekuensi rendah. Dikarenakan filter *pre-emphasis* tidak menggunakan frekuensi *cut-off* seperti halnya *lowpass filter*, *highpass filter*, dan *bandpass filter*, membuat pemilihan *sample rate* yang digunakan turut mempengaruhi posisi frekuensi tinggi (*high frequency*) dimana energi (*magnitude*) hasil filter *pre-emphasis* lebih besar dari energi (*magnitude*) *original*, walaupun konstanta filter *pre-emphasis* α tetap ($\alpha=0,97$). Dalam penelitian ini pada *sample rate* 44100 Hz, *magnitude* hasil filter lebih besar dari *magnitude original* pada frekuensi sekitar ± 7520 Hz. Pada *sample rate* 32000 Hz, *magnitude* hasil filter lebih besar pada frekuensi sekitar ± 5428 Hz. Pada *sample rate* 22050 Hz, *magnitude* hasil filter lebih besar pada frekuensi sekitar ± 3740 Hz.
2. Dengan menggunakan algoritma *Fast Fourier Transform* dan *Mean Square Percentage Error* diketahui seberapa besar perubahan (distorsi) yang terjadi terhadap spektrum suara setelah menggunakan filter *pre-emphasis*. Hasilnya menunjukkan bahwa spektrum suara hasil filter *pre-emphasis* mengalami perubahan sekitar $\pm 43\%$ dari suara aslinya (*original*), walaupun menggunakan *sample rate* dan durasi yang berbeda. Terlihat bahwa filter *pre-emphasis* membuat suara berubah cukup besar dari aslinya.
3. Dengan menampilkan spektrogram dari audio hasil filter *pre-emphasis* dan audio *original*, terlihat bahwa audio hasil filter memiliki bentuk spektrum yang lebih jelas dan memiliki energi yang lebih merata sepanjang frekuensi dibandingkan audio *original*. Serta suara yang dihasilkan dari hasil filter terdengar lebih jernih, halus, dan jelas.

Saran dimana dalam penelitian ini menggunakan transformasi *fourier* (*Fast Fourier Transform*), untuk penelitian selanjutnya dapat menggunakan transformasi lain untuk melakukan analisa perubahan spektrum suara hasil filter *pre-emphasis*. Seperti menggunakan transformasi *z*, dsb. Selanjutnya juga dapat dikembangkan penelitian (dapat dibuat aplikasi *benchmark*) untuk beberapa jenis filter suara.

DAFTAR PUSTAKA

- [1] Tanudjaja, Harlianto. 2007. *Pengolahan Sinyal Digital & Sistem Pemrosesan Sinyal*. Yogyakarta: Penerbit Andi.
- [2] Swanson, David. A., Jeff Tayman., & T. M. Bryan. 2011. *MAPE-R: A RESCALED MEASURE OF ACCURACY FOR CROSS-SECTIONAL FORECASTS*.

- [3] Deng, Li & Douglas O'Shaughnessy. 2003. *Speech Processing: A Dynamic and Optimization-Oriented Approach*. New York: Marcel Dekker.
- [4] Pan, Linlin. 2013. *Research and Simulation on Speech Recognition by Matlab*. University of Gavle.
- [5] Madani, S. A., Jawad Kazmi, & Stefan Mahlke. 2010. *Wireless sensor networks: modeling and simulation*. InTech.