

Implementasi Algoritma FBLMS Untuk Pereduksi Derau Pada Sinyal Suara Menggunakan TMS321C6416T

Dimitra Levina Hartono, Suwadi, Wirawan

Teknik Elektro, Fakultas Teknik Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: suwadi@ee.its.ac.id

Abstrak - Dalam sistem komunikasi, derau merupakan sinyal yang tidak diinginkan, yang menyertai sinyal informasi sehingga akan mempengaruhi kualitas sinyal yang akan dikirimkan. Pada dunia nyata, sinyal yang akan dikirimkan, juga telah tercampur dengan derau sebelum sinyal tersebut dilewatkan pada kanal transmisi. Oleh karena itu, diperlukan pemrosesan sinyal terlebih dahulu, dengan mereduksi derau pada sinyal sebelum proses transmisi. Pada aplikasi filter adaptif penghapus derau, sebagai masukannya adalah sinyal utama yaitu sinyal informasi yang telah tercampur dengan derau dan sinyal referensi dimana sinyal referensi tersebut memiliki korelasi dengan sinyal utama. Dalam hal ini sinyal referensinya adalah derau itu sendiri. Dan sebagai keluarannya adalah kesalahan (*error*) dan estimasi sinyal informasi. Selama pengujian dan analisa algoritma Fast Blok LMS (FBLMS) menggunakan modul DSK TMS320C6416T ini telah berhasil diimplementasikan sistem penghilang derau berbasis TMS320C6416T dengan menggunakan pembanding algoritma LMS dan FBLMS. Hal ini dapat dibuktikan dengan perbandingan SNR input yang berupa suara beserta *noise*, dan SNR output yang berupa suara yang telah bersih dari derau. Nilai SNR output lebih besar dari SNR input. Pemilihan *step size* dan *filter length* sangat menentukan filter yang akan mempengaruhi stabilitas dan kecepatan konvergensi. Hal ini dapat diambil kesimpulan bahwa pada perbandingan SNR untuk hasil simulasi antara LMS dan FBLMS, untuk hasil terbaik yaitu pada saat *step size* dengan besar 0.005, baik untuk *filter length* 16, 32, maupun 32. Nilai SNR terbesar dan konvergen untuk LMS adalah 28.2499 sedangkan FBLMS adalah 28.2362. Sehingga dapat disimpulkan bahwa hasil kinerja LMS masih lebih baik daripada FBLMS.

Kata kunci - LMS dan FBLMS, TMS320C6416T.

I. PENDAHULUAN

Dalam sistem komunikasi derau merupakan masalah utama yang senantiasa menyertainya. Derau menyebabkan penurunan kualitas sinyal informasi, bahkan dapat terjadi pada saat sebelum sinyal ditransmisikan. Untuk mengatasinya, diperlukan pemrosesan sinyal sebelum sinyal ditransmisikan sehingga derau dapat direduksi.

Pemfilteran merupakan salah satu cara untuk mereduksi derau dari sinyal informasi. Berdasarkan kemampuannya untuk beradaptasi, filter digolongkan menjadi dua, yaitu filter adaptif dan filter non-adaptif. Dalam hal ini diperlukan filter adaptif yang mempunyai kemampuan untuk mengubah karakteristik derau secara kontinyu. Algoritma adaptasi yang digunakan adalah algoritma LMS dan FBLMS untuk dibandingkan

karakteristiknya dalam menghapus derau dengan menggunakan perangkat DSP *board* TMS320C6416T.

Hasil implementasi yang diinginkan berupa evaluasi kualitas signal dan kecepatan konvergensi (atau kecepatan adaptasi) dan MSE.

II. DASAR TEORI

A. Definisi Filter

Filter adalah suatu alat untuk memisahkan sinyal-sinyal yang diinginkan dari sinyal-sinyal yang tidak diinginkan. Filter berkembang dalam pemakaiannya di bidang Elektroteknik menjadi sebagai alat untuk memisahkan signal dari derau.

B. Filter Digital

Filter digital menggunakan *digital processor* untuk melakukan kalkulasi numerik pada harga contoh sinyal. Prosesor digital yang digunakan yang biasa digunakan seperti PC atau DSP (*Digital Signal Processing*) *chip*.

Filter merupakan nama umum yang mengacu pada sistem LTI untuk melakukan seleksi frekuensi. Dengan demikian sistem LTI waktu-diskret juga disebut filter digital. Ada dua jenis filter digital, yaitu FIR dan IIR.

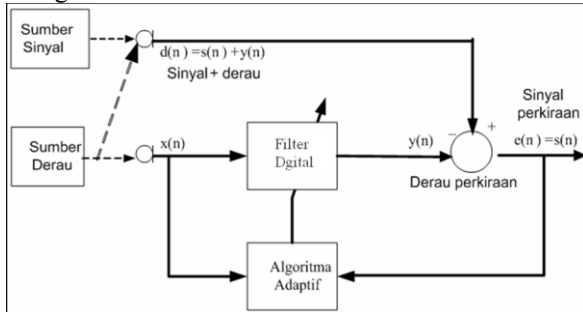
Beberapa keuntungan penggunaan filter digital antara lain:

1. Filter digital bisa di program (*programmable*). Operasi yang dilakukan dapat diprogram yang kemudian dapat disimpan di memori prosesor. Hal ini menunjukkan filter mudah untuk diubah melalui program tanpa mengubah rangkaian elektronik (*hardware*).
2. Filter digital lebih mudah di desain, dites dan diimplementasikan.
3. Karakteristik rangkaian filter analog tergantung perubahan temperatur, filter digital tidak terpengaruh perubahan temperatur dan sangat stabil.
4. Filter digital mampu bekerja pada sinyal frekuensi rendah dengan akurat.
5. Filter digital serbaguna dalam kemampuannya memproses berbagai sinyal, seperti filter adaptif yang mampu menyesuaikan terhadap perubahan sinyal.
6. Fast DSP Processor mampu menangani kombinasi kompleks filter paralel atau serial, membuat peralatan hardware lebih sederhana di bandingkan filter analog.

C. Filter Adaptif

Filter adaptif merupakan sebuah filter dengan pengatur koefisien dimana parameter filter diatur sedemikian rupa sehingga dapat mengoptimalkan sinyal dari *distorsi* (cacat) seminimal mungkin[1]. Filter yang paling adaptif adalah filter yang melakukan pemrosesan sinyal digital dan menyesuaikan kinerja berdasarkan sinyal input.

Fasilitas yang menarik dari filter adaptif adalah kemampuan untuk beroperasi secara maksimal pada lingkungan yang tidak dikenal dan juga variasi waktu pelacakan dari statistik input, membuat filter adaptif menjadiperalatan dalam pemrosesan sinyal. Beberapa contoh aplikasinya adalah gema, tapis kebisingan.



Gambar 3. Diagram Blok Filter Adaptif Sebagai Penghilang Derau [2]

D. Algoritma LMS (Least Mean Square)

Algoritma LMS yang dikembangkan oleh Widrow dan Hoff (1959)[5]. Setiap koefisien bobot dari filter digital diperbaharui dengan menggunakan algoritma adaptif yang berdasarkan pada setiap data masukan.

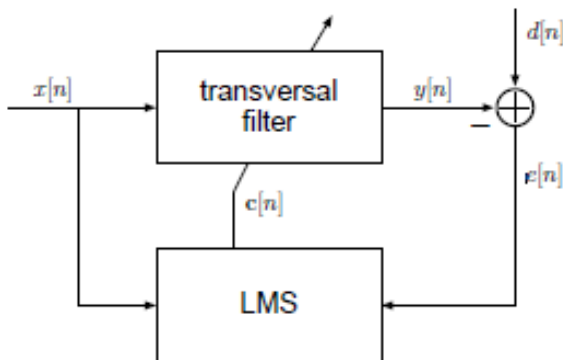
Sub program LMS ini melakukan proses-proses perhitungan berdasarkan pada algoritma Filter Wiener dan konsep dasar pereduksi derau.

Data masukan sub program LMS :

1. Sinyal masukan informasi
2. Sinyal derau
3. Jumlah sampling
4. Orde filter

Proses perhitungan algoritma LMS adalah sebagai berikut :

1. Menghitung nilai estimasi untuk sinyal campuran yang diperoleh dari perkalian karakteristik filter dengan derau $x(n)=h(n)y(n)$.
2. Menghitung nilai $e(n)$ yang merupakan selisih antara sinyal $x(n)$ dengan sinyal estimasinya, $e(n)=x(n)-x(n)$.
3. Mencari nilai karakteristik filter yang baru dengan persamaan $h(n-1)=h(n)+2\mu e(n)y(n)$.



Gambar 1. Block Diagram Adaptif Filter Dengan LMS

E. Algoritma FBLMS (Fast Block Least Mean Square)

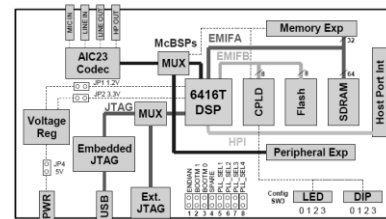
Filter adaptif harus bekerja secara real time untuk menyaring sinyal masuk. Dalam situasi seperti ini, dapat digunakan algoritma FBLMS [c]. FBLMS mengubah sinyal input $x(n)$ ke domain frekuensi menggunakan Fast Fourier Transform (FFT).

Operasi perkalian yang dibutuhkan untuk algoritma FBLMS jauh lebih rendah daripada algoritma LMS. proses sampling yang dilakukan adalah per blok. Jika kedua panjang filter dan ukuran blok adalah N , maka:

- Algoritma LMS memerlukan: $N(2N + 1)$
- Algoritma FBLMS hanya membutuhkan: $(10N \log 2N + 26N)$

F. Digital Signal Processor (DSP) TMS320C6T

Digital Signal Processor (DSP) seri TMS320C6T adalah mikroprosesor berkecepatan tinggi dengan tipe arsitektur yang cocok digunakan untuk mengolah sinyal [4].



Gambar 2. Diagram Blok DSK TMS320C6713

DSP dapat diaplikasikan sebagai pengontrol suara, pengolah gambar dan pengolah sinyal lainnya. DSP dapat ditemukan pada telepon seluler, *harddisk*, radio, printer, MP3 players, HDTV, kamera digital dan alat elektronik lainnya. DSP dapat melakukan banyak proses karena DSP dapat diprogram untuk aplikasi yang berbeda-beda. Selain itu, DSP sangat sedikit terpengaruh oleh perubahan kondisi lingkungan sekitar seperti suhu.

DSP dikemas dalam sebuah DSK (*Digital Signal Processor Starter Kit*) dengan ukuran sekitar 5 x 8 inch dimana terdapat C6713 *floating-point* dan sebuah 32 bit *stereo codec* TLV320AIC23 (AIC23) sebagai masukan dan keluaran. *Codec* AIC23 menyediakan ADC dan DAC dengan clock 12 MHz dan sampling rate 8 – 96 kHz. DSK memiliki memori tipe *Synchronous Dynamic Random Access Memory* (SDRAM) sebesar 16 MB dan flash memori sebesar 256 kB. Ada 4 konektor pada DSK yang berfungsi sebagai masukan dan keluaran yaitu :

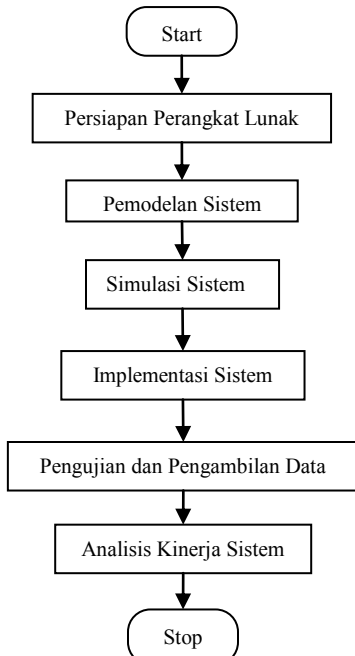
- a. MIC IN untuk masukan *microphone*.
- b. LINE IN untuk masukan *line*.
- c. LINE OUT untuk keluaran *line*.
- d. HEADPHONE untuk keluaran *headphone* yang tergabung dalam LINE OUT.

Dip switches dalam DSK dapat difungsikan sesuai program dan menyediakan fungsi kontrol. Selain itu terdapat pengatur tegangan yang menyediakan 1,26 volt untuk prosesor C6713 dan 3,3 volt untuk sistem memori dan kelengkapan lain. Fitur-fitur lain yang tersedia dalam DSK TMS320C6713 terdapat

dalam lampiran.

III. PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan ditampilkan mengenai gambaran sistem yang akan diteliti dan diuji secara umum meliputi perangkat lunak (*software*), simulasi sistem, proses implementasi sistem, dan metode pengujian yang digunakan. Untuk memudahkan, berikut gambar skema metodologi yang digunakan dalam tugas akhir ini.



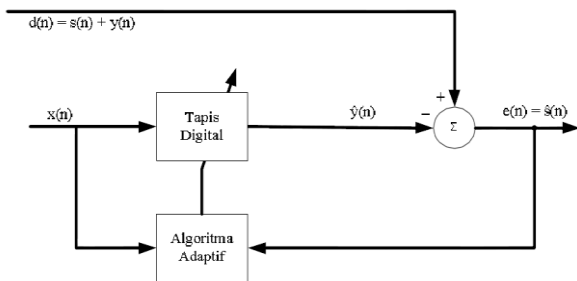
Gambar 4. Skema Metodologi

A. Persiapan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam proses ini adalah Matlab versi R2007a dan Composer Studio (CCS) versi 3.1. Semua perangkat lunak ini berjalan pada *Operating System* Windows XP.

B. Pemodelan Sistem

Pemodelan sistem untuk blok LMS dan FBLMS terdapat dua macam bentuk. Yaitu pemodelan saat simulasi, yaitu pemodelan saat disimulasikan pada simulink, dan pemodelan pada simulink yang diaplikasikan pada TMS.



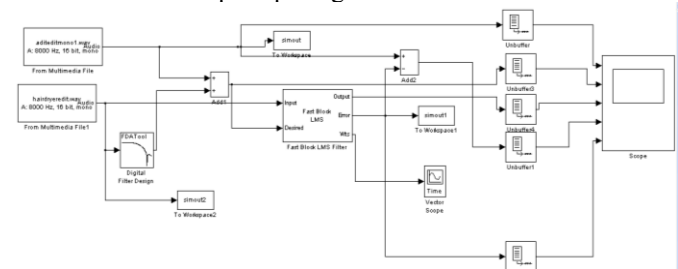
Gambar 5. Blok Diagram Filter Adaptif [e]

Berdasarkan pada gambar 2 terdapat dua buah sinyal masukan, yaitu $d(n)$ dan $x(n)$. $d(n)$ berfungsi sebagai sinyal masukan yang terdiri dari sinyal informasi (suara) $s(n)$ yang tercampur dengan derau $y(n)$. Sedangkan $x(n)$ adalah derau

yang akan diproses untuk mendapatkan nilai pendekatan $\hat{y}(n)$ dari derau yang terdapat pada sinyal $d(n)$ sehingga derau yang terdapat pada $d(n)$ dapat dikurangi atau dihilangkan.

C. Simulasi Sistem

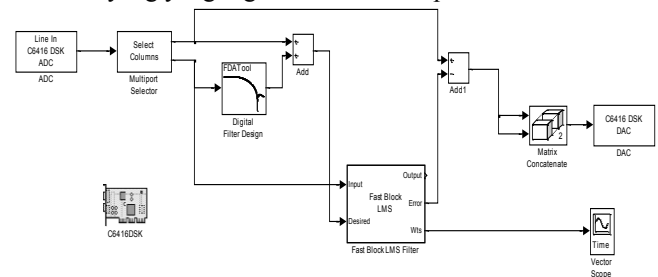
Berdasarkan pemodelan sistem algoritma FBLMS dengan satu blok bagian utama, maka dapat dibuat simulasi Simulink Matlab seperti pada gambar 6.



Gambar 6. Diagram Blok Simulasi FBLMS

D. Implementasi Sistem

Implementasi sistem dilakukan dengan cara memprogram DSK TMS320C6416T sesuai dengan sistem yang kita buat. DSK dapat deprogram dengan menggunakan bahasa C, assembly, atau integrasi Simulink Matlab. Setelah menyiapkan file Simulink yang akan diimplementasikan, ada beberapa blok tambahan yang digunakan untuk implementasi DSK.



Gambar 7. Diagram Blok Implementasi FBLMS

E. Metode Pengujian dan Pengolahan Data

Pengujian MSE dilakukan dengan cara menyampling hasil keluaran errornya lalu dirata-rata. Rumus dari MSE adalah sebagai berikut:

$$MSE = \frac{1}{n} \sum_{i=1}^n (s - se)^2$$

Dimana n = jumlah sampel data

S = sinyal asli

se = sinyal estimasi

Sedangkan untuk perhitungan SNR nya didapatkan dengan rumus sebagai berikut :

$$SNR = 10 \log \frac{\text{sinyal asli}}{\text{noise}}$$

IV. PENGUJIAN DAN ANALISA SISTEM

Dalam bab ini akan dibahas mengenai pengujian dari implementasi dengan menggunakan algoritma LMS dan FBLMS yang telah dijelaskan pada bab III. Ada dua model pengujian yang akan digunakan, yaitu :

1. Pengujian secara simulink.

2. Pengujian secara implementasi.

Tabel 1.
Variansi Parameter untuk Simulasi dan Implementasi

Simulasi FBLMS	Filter Length	Step size (mu)
1	16	0.05
2	16	0.005
3	16	0.0005
4	32	0.05
5	32	0.005
6	32	0.0005
7	64	0.05
8	64	0.005
9	64	0.0005

A. Pengujian Secara Simulink

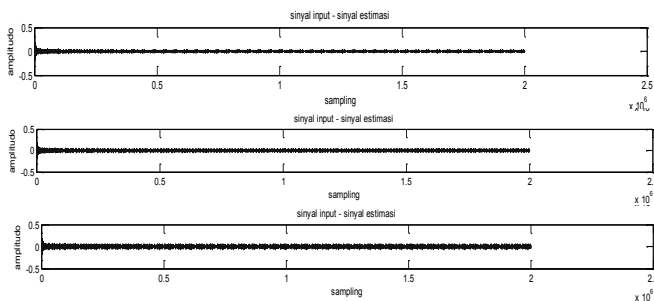
Sebelum implementasi ke TMS320C6416T dan pengambilan data, maka dilakukan pengujian dan pengambilan data dengan simulasi. Hal ini bertujuan agar data simulasi dapat menjadi acuan dan pembanding dari data hasil imolementasi, dan menjadi tolak ukur keberhasilan dari implementasi sistem algoritma FBLMS pada tugas akhir ini.

Agar lebih mudah dalam membandingkan hasil keluaran sinyal yang dihasilkan oleh sistem secara simulasi, makan sistem akan diatur dengan variable tetap yang berupa panjang filter dan variable ubah berupa konstanta *step size* μ seperti pada tabel data 4.2.

Dengan macam variansi yang digunakan, ada 9 macam simulasi yang akan dilakukan, yaitu ditampilkan pada tabel 4.2 di bawah ini :

Tabel 2
Perbandingan MSE LMS dan FBLMS Terhadap Filter Length (Simulasi)

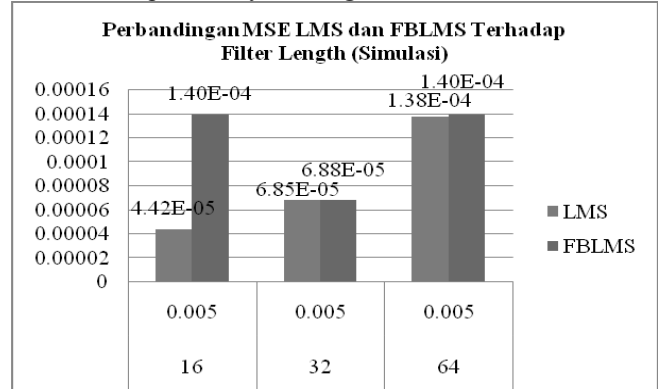
NO	L	mu	SNR input	MSE Simulasi	
				LMS	FBLMS
2	16	0.005	6.9383	4.42E-05	4.40E-04
5	32	0.005	6.9383	6.85E-05	6.88E-05
8	64	0.005	6.9383	1.38E-04	1.40E-04



Gambar 8. Hasil Output Error Terhadap Filter Length (Simulasi FBLMS)

Dari hasil plot ketiga gambar di atas, dapat dianalisa bahwa semakin panjang *filter length* maka pada plot sinyal input - sinyal estimasi yaitu sinyal errornya hampir mendekati

nol. Dapat dilihat bahwa pada saat *filter length* 64 nilai MSE lebih kecil daipada saat *filter length* 16.

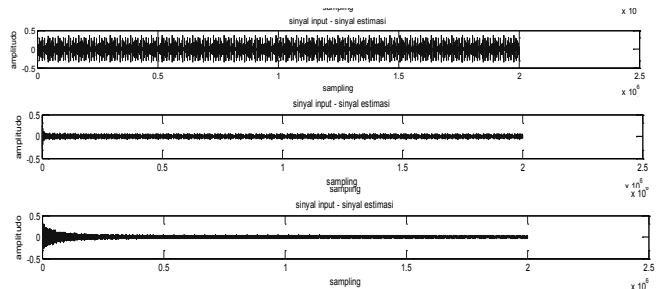


Gambar 9. Perbandingan MSE LMS dan FBLMS Terhadap Filter Length (FBLMS)

Untuk selanjutnya, akan dibandingkan antara hasil plot dari keluaran sinyal dengan *step size* yang berbeda dan panjang filer yang sama.

Tabel 3.
Perbandingan MSE LMS dan FBLMS Terhadap Step Size (Simulasi)

NO	L	mu	SNR input	MSE Simulasi	
				LMS	FBLMS
7	64	0.05	6.9383	0.0013	0.0015
8	64	0.005	6.9383	1.38E-04	1.40E-04
9	64	0.0005	6.9383	1.91E-04	1.91E-04



Gambar 10. Hasil Output Error Terhadap Step Size (Simulasi FBLMS)

Dari hasil plot ketiga data di atas dapat dianalisa bahwa *step size* mempengaruhi kestabilan hasil keluaran error. Pada gambar terlihat pada saat *step size* 0.0005 hasil keluaran errornya lebih baik dan hampir mendekati nol pada saat *step size* ke 0.0005, lebih baik daripada saat *step size* 0.05.

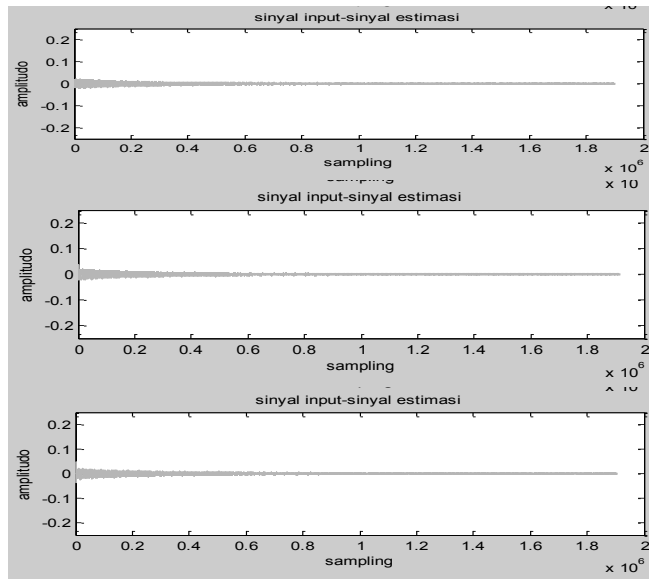
B. Pengujian Hasil Implementasi

Setelah melakukan pengujian dengan simulasi, maka langkah selanjutnya adalah mengujian dengan implementasi ke TMS320C6416T.

Tabel 4
Perbandingan MSE LMS dan FBLMS Terhadap Filter Length

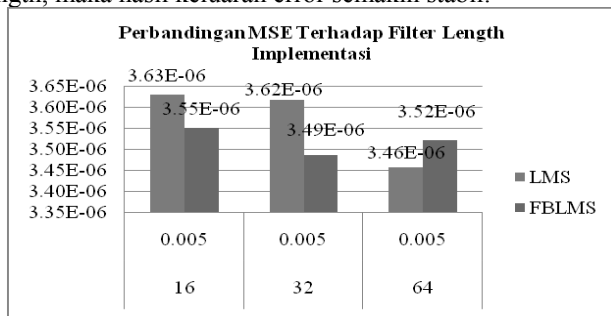
NO	L	mu	SNR input	MSE Implementasi	
				LMS	FBLMS
2	16	0.005	6.9383	3.63E-06	3.55E-06
5	32	0.005	6.9383	3.62E-06	3.49E-06
8	64	0.005	6.9383	3.46E-06	3.52E-06

(Implementasi)



Gambar 11. Hasil Output Error Terhadap Filter Length (Implementasi FBLMS)

Dari perbedaan simulasi dan implementasi dari ketiga gambar diatas, dengan filter length yang berbeda dan step size yang sama, maka dapat dilihat bahwa semakin besar nilai filter length, maka hasil keluaran error semakin stabil.

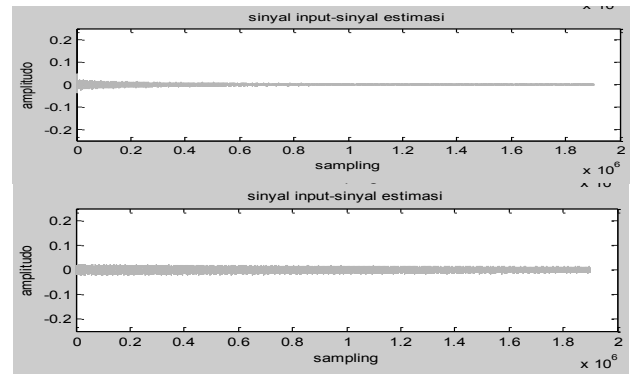
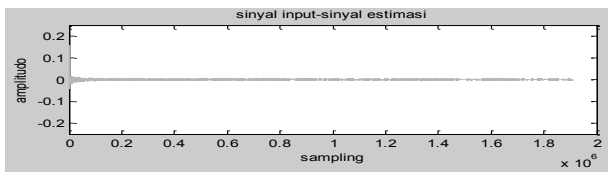


Gambar 14. Perbandingan MSE LMS dan FBLMS Terhadap Filter Length (Implementasi)

Kemudian untuk selanjutnya adalah perbandingan dari keluaran sinyal dengan panjang filter yang sama dan step size yang berbeda, yang akan dibandingkan antara hasil simulasi dan implementasi.

Tabel 5
Perbandingan MSE LMS dan FBLMS Terhadap Step Size (Implementasi)

NO	L	mu	SNR input	MSE Implementasi	
				LMS	FBLMS
7	64	0.05	6.9383	8.16E-07	7.94E-07
8	64	0.005	6.9383	3.46E-06	3.52E-06
9	64	0.0005	6.9383	1.73E-05	1.77E-05



Gambar 12. Hasil Output Error Terhadap Step Size (Implementasi FBLMS)

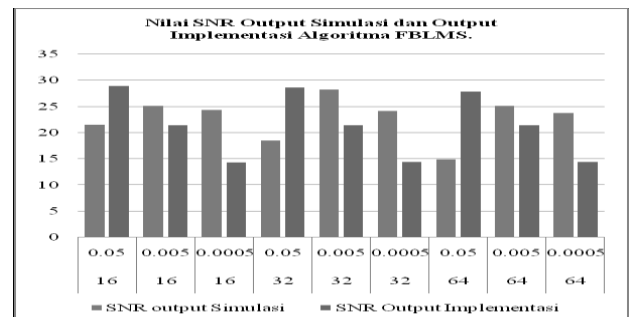
C. Perhitungan SNR

Untuk perhitungan SNR ini dilakukan dengan bantuan matlab dengan bentuk *m-file*, yang kemudian ditampilkan nilai dari SNR input dan SNR output dalam bentuk *decibel* (dB), sehingga untuk perhitungan dengan algoritma LMS dan FBLMS dihasilkan nilai seperti pada tabel 4.4 dibawah ini.

Tabel 6.
Perbandingan Nilai SNR Output Simulasi dan Output Implementasi Algoritma FBLMS

L	mu	SNR input	SNR output Simulasi	SNR Output Implementasi
16	0.05	6.9383	21.4976	28.9682
16	0.005	6.9383	25.1528	21.4008
16	0.0005	6.9383	24.3425	14.306
32	0.05	6.9383	18.5026	28.6972
32	0.005	6.9383	28.2362	21.4762
32	0.0005	6.9383	24.2151	14.3951
64	0.05	6.9383	14.9296	27.909
64	0.005	6.9383	25.1528	21.4379
64	0.0005	6.9383	23.7885	14.4318

Hal ini menunjukkan bahwa panjang filter tidak telalu mempengaruhi noise, namun kenaikan panjang filter ini mempengaruhi SNR, dimana nilai SNR bertambah. Dapat dilihat bahwa perubahan nilai SNR yang tidak signifikan



Gambar 13. Perbandingan Nilai SNR Output Simulasi dan Output Implementasi Algoritma FBLMS

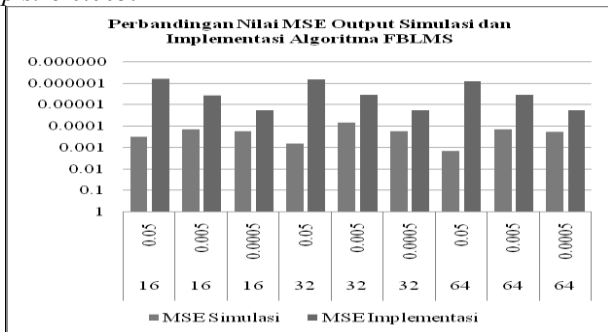
D. Perhitungan MSE

Untuk perhitungan MSE ini dilakukan dengan bantuan matlab dengan bentuk *m-file*, sehingga untuk perhitungan dengan algoritma LMS dan FBLMS dihasilkan nilai seperti pada tabel 7.

Tabel 7.
Perbandingan Nilai MSE Output Simulasi dan Implementasi Algoritma FBLMS

L	mu	MSE Simulasi		MSE Implementasi	
		LMS	FBLMS	LMS	FBLMS
16	0.05	3.20E-04	3.24E-04	6.91E-07	6.22E-07
16	0.005	4.42E-05	1.40E-04	3.63E-06	3.55E-06
16	0.0005	1.69E-04	1.69E-04	1.82E-05	1.82E-05
32	0.05	6.15E-04	6.47E-04	7.52E-07	6.62E-07
32	0.005	6.85E-05	6.88E-05	3.62E-06	3.49E-06
32	0.0005	1.73E-04	1.74E-04	1.76E-05	1.79E-05
64	0.05	0.0013	0.0015	8.16E-07	7.94E-07
64	0.005	1.38E-04	1.40E-04	3.46E-06	3.52E-06
64	0.0005	1.91E-04	1.91E-04	1.73E-05	1.77E-05

Dalam tabel di atas dapat dilihat bahwa nilai yang mendekati hasil simulasi adalah pada *filter length* 32 dengan *step size* 0.005.



Gambar 14. Perbandingan Nilai MSE Output Simulasi dan Implementasi Algoritma FBLMS

V. KESIMPULAN/RINGKASAN

Dari segala yang dilakukan dalam tugas akhir ini, meliputi tahap perencanaan, simulasi, dan implementasinya, terdapat beberapa kesimpulan yang dapat diambil sebagai berikut.

1. Pada TA telah berhasil diimplementasikan sistem penghilang derau berbasis TMS320C6416T dengan menggunakan algoritma LMS dan FBLMS. Hal ini dapat dibuktikan dengan perbandingan rata-rata SNR input yang berupa suara beserta noise, dan SNR *output* yang berupa suara yang telah bersih dari derau. Nilai SNR *output* lebih besar dari snr *input*.
2. Pemilihan step size dan filter length sangat menentukan filter yang akan mempengaruhi stabilitas dan kecepatan konvergensi. Sehingga pada perbandingan SNR untuk hasil simulasi antara LMS dan FBLMS, untuk hasil terbaik yaitu pada saat *step size* dengan besar 0.005, baik untuk *filter length* 16 (LMS=30.1508, FBLMS=25.1528), 32 (LMS=28.2499, FBLMS=28.2362), maupun untuk *filter length* 64 (LMS=25.1988, FBLMS=25.1528).
3. Semakin kecil nilai MSE, maka semakin besar pula nilai SNR nya. Nilai MSE yang terendah adalah pada saat *step size* dengan besar 0.005, baik untuk *filter length* 16 (LMS=4.42E-05, FBLMS=1.40E-04), 32 (LMS=6.85E-05,

FBLMS=6.88E-05), maupun untuk *filter length* 64 (LMS=1.38E-04, FBLMS=1.40E-04).

4. Hasil SNR terbesar dan paling konvergen untuk FBLMS antara simulasi dan implementasi adalah pada saat *filter length* 32, yaitu 28.2362 untuk simulasi dan 21.4762 untuk implementasi.
5. Nilai SNR terbesar dan konvergen untuk LMS adalah 28.2499 sedangkan FBLMS adalah 28.2362. Sehingga dapat disimpulkan bahwa hasil kinerja LMS masih lebih baik daripada FBLMS.

UCAPAN TERIMA KASIH

Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungan yang telah diberikan selama proses pembuatan tugas akhir ini kepada :

1. Ayah dan Ibu tercinta, atas semua doa,dukungan baik materiil maupun spiritual.
2. Ibu Devy Kuswidiastuti, ST, MSc. selaku Kepala program studi Telekomunikasi Teknik Elektro ITS.
3. Bapak Dr. Ir. Suwadi, MT. dan Dr. Ir. Wirawan, DEA. selaku dosen pembimbing yang selama ini sudah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
4. Para dosen penguji yang telah membantu dalam pengembangan dan perbaikan tugas akhir ini.
5. Para dosen pengajar Teknik Elektro ITS.
6. Semua teman-teman angkatan LJ Genap 2010.

DAFTAR PUSTAKA

- [1] Haykin, Simon. 1995. *Adaptive Filter Theory 3rd Edition*. Prentice Hall.
- [2] Arif, Sulamul. Perbandingan Kerja Algoritma LMS dan NLMS Untuk Peredaman Derau Secara Adaptif. Semarang: Universitas Diponegoro.
- [3] Jebastine, J. 2012. Design And Implementation of Noise Free Audio Speech Signal Using Fast Block Least Mean Square Algorithm. Chennai : Satyabama University Chennai.
- [4] Chassaing, R., "Digital Signal Processing and Application with the C6713 and C6416 DSK", Wiley-Interscience, United State of America, Ch. 1, 2005.
- [5] Sudarmono, Bambang. Penggunaan Tapis Adaptif Dalam Proses Editing Suara Pada Pembuatan Film Layar Lebar. Semarang: Universitas Diponegoro.