

Implementasi Algoritma BLMS Untuk Pereduksi Derau Pada Sinyal Suara Menggunakan TMS320C6416T

Rayi Margina Putri Kurnia, Suwadi, Wirawan

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

Jl. Arief Rahman Hakim, Surabaya 60111

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

Abstrak— Dalam system 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 BLMS untuk dibandingkan karakteristiknya dalam mengurangi derau dengan menggunakan perangkat DSP board TMS320C6416T. Hasil implementasi yang diinginkan yaitu memperoleh sinyal keluaran informasi tanpa noise. Nilai SNR yang paling baik yaitu pada parameter panjang filter 16 dengan step size 0.05 sebesar 30,0193 dB dengan MSE minimum diperoleh 6,9129E-07. Sedangkan hasil simulasi diperoleh SNR terbesar dengan parameter panjang filter 16 pada step size 0,005 yaitu 28,4183 dB dengan nilai MSE minimum 5,47E-05. Semakin panjang filter (orde semakin tinggi) maka proses perhitungan semakin lama. Jika parameter konvergensi diperkecil maka filter membutuhkan waktu yang lebih lama untuk beradaptasi namun akan didapatkan sinyal yang mendekati sinyal yang diinginkan.

Kata Kunci—Filter adaptif, LMS, BLMS, DSP TMS320C6416.

I. PENDAHULUAN

Pada sistem telekomunikasi, salah satu permasalahan yang tidak dapat diabaikan adalah munculnya sinyal lain yang tidak diinginkan yang menyertai sinyal informasi, sinyal tersebut adalah derau. Derau merupakan masalah serius pada bidang telekomunikasi, karena dengan munculnya derau, sinyal informasi yang dikirimkan akan menjadi terganggu sehingga kualitas sinyal informasi menjadi buruk.

Derau dapat muncul pada sinyal informasi yang dikirimkan melalui beberapa sisi, yaitu sisi pemancar, transmisi dan penerima. Pada sisi pemancar, kadang-kala derau memang sudah bercampur dengan sinyal informasi sebelum sinyal informasi tersebut dilewatkan pada peralatan-peralatan pemancar, atau derau masuk pada sinyal informasi melalui rangkaian penguat pada pemancar. Pada sisi transmisi, yang dapat menjadi sumber derau adalah medium transmisi yang buruk. Sedangkan pada sisi penerima, rangkaian-rangkaian yang digunakan pada penerima juga dapat membangkitkan derau.

Derau mempunyai karakteristik acak, sehingga diperlukan filter yang mempunyai karakteristik yang selalu berubah secara terus-menerus mengikuti perubahan derau tersebut. Hal ini bertujuan agar kesalahan keluaran filter tersebut diharapkan selalu minimum, atau dengan kata lain filter bekerja secara optimal. Filter yang dapat selalu merubah karakteristiknya sendiri adalah filter adaptif.

II. GAMBARAN SISTEM

A. Definisi Filter

Filter digunakan untuk mengatur sinyal yang bercampur derau (*noise*) dengan cara memisahkan sinyal informasi dengan derau sehingga diperoleh sinyal yang diinginkan.

Filter mempunyai tiga fungsi dasar yaitu :

1. Pemfilteran (*Filtering*), dengan cara menghilangkan sinyal yang tidak diinginkan.
2. Penghalusan (*smoothing*),
3. Pemrediksi (*prediction*),

Filter yang menerapkan fungsi linier pada masukan filter sehingga pada keluaran diperoleh sinyal yang telah mengalami proses pemfilteran, penghalusan dan pemrediksi disebut filter linier [1].

B. Aplikasi Filter adaptif

Keistimewaan filter adaptif sesuai dengan namanya adalah kemampuan operasi dalam lingkungan yang tidak diketahui secara optimal dan penjejakan variasi waktu statistik masukan sehingga filter adaptif merupakan perangkat yang handal untuk pemrosesan sinyal dan aplikasi kontrol [2]. Filter adaptif dapat diaplikasikan menjadi empat aplikasi pokok sebagai berikut :

Tabel 1
Aplikasi Filter Adaptif

Jenis Filter Adaptif	Aplikasi
Identifikasi	Identifikasi sistem Pemodelan lapisan bumi
Pemodelan invers	Dekonvolusi prediktif Ekualisasi adaptif
Pemrediksi	Pengkodean prediktif linier Adaptif Differensial pulse-code modulation Analisa spektrum autoregresif Pendeteksi sinyal

Jenis Filter Adaptif	Aplikasi
Penghapus interferensi	Penghapus derau adaptif Penghapus echo Polarimetri radar Pembentuk beam adaptif

C. Alogartima Block Least Mean Square (LMS) [3-4]

Algoritma LMS mempunyai kinerja yang dapat menekan noise dari sinyal yang telah terkena noise. Algoritma LMS merupakan pendekatan dari algoritma steepest descant yang menggunakan estimasi sesaat dari gradien vektor sebagai fungsi yang dipertimbangkan. Nilai estimasi gradien didasarkan pada nilai sampel vektor masukan dan nilai galat (error) yang diperoleh. Metode adaptasi LMS didefinisikan sebagai :

$$w(n + 1) = w(n) + \mu \left(-\frac{\partial e^2(n)}{\partial w(n)} \right) \tag{1}$$

Dimana sinyal error $e(n)$ diberikan oleh :

$$e(n) = x(n) - w^T(n)x(n) \tag{2}$$

Gradien sesaat dari error kuadrat dapat kembali dinyatakan sebagai :

$$\begin{aligned} \frac{\partial e^2(n)}{\partial w(n)} &= \frac{\partial}{\partial w(n)} [x(n) - w^T(n)y(n)]^2 \\ &= -2y(n)[x(n) - w^T(n)y(n)]^2 \\ &= -2y(n)e(n) \end{aligned} \tag{3}$$

Dengan mensubstitusi persamaan (3) ke dalam persamaan pembaruan rekursi dari parameter filter Persamaan (1) menghasilkan persamaan adaptasi LMS :

$$w(n + 1) = w(n) + 2\mu[y(n)e(n)] \tag{4}$$

D. Alogartima Block Least Mean Square (BLMS)[5]

Algoritma block LMS (BLMS) adalah alogaritma yang bekerja berdasarkan alogaritma dasar LMS. Bobot filter akan diupdate sekali setelah setiap blok terkumpul dari sample data. Gradien vektor $-2e(n)\bar{X}(n)$, digunakan untuk meng-update bobot filter yang dijumlahkan pada blok saat itu. Menggunakan k sebagai simbol indeks blok, perulangan BLMS dapat diperoleh sebagai :

$$\bar{W}(k + 1) = \bar{W}(k) + 2\mu_B \frac{\sum_{i=0}^{L-1} e(kL+i)\bar{X}(kL+i)}{L} \tag{5}$$

Dimana L adalah panjang blok dan μ_B adalah parameter step-size.

$$e(kL + i) = d(kL + i) - y(kL + i), \quad i = 0,1,2, \dots, L - 1 \tag{6}$$

Definisi matrix (blok ke k)

$$X(k) = [\bar{X}(kL) \quad \bar{X}(kL + 1) \quad \dots \quad \bar{X}(kL + L - 1)]^T \tag{7}$$

Dan vektor kolom,

$$\bar{d}(k) = [d(kL) \quad d(kL + 1) \quad \dots \quad d(kL + L - 1)]^T$$

$$\bar{y}(k) = [y(kL) \quad y(kL + 1) \quad \dots \quad y(kL + L - 1)]^T$$

$$\bar{e}(k) = [e(kL) \quad e(kL + 1) \quad \dots \quad e(kL + L - 1)]^T \tag{8}$$

Catatan,

$$\bar{y}(k) = X(k)\bar{W}(k)$$

$$\bar{e}(k) = \bar{d}(k) - \bar{y}(k)$$

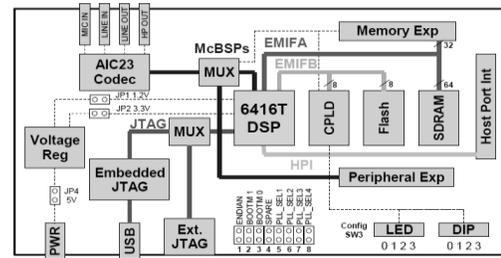
$$\sum_{i=0}^{L-1} e(kL + i)\bar{x}(kL + i) = X^T(k)\bar{e}(k) \tag{9}$$

Sehingga didapatkan persamaan,

$$\bar{W}(k + 1) = \bar{W}(k) + 2 \frac{\mu_B}{L} X^T(k)\bar{e}(k) \tag{10}$$

E. DSK TMS320C6416

Sistem yang telah direalisasikan menggunakan digital signal processor starter kit (DSK) TMS320C6416 yang di produksi Texas Instruments [6-7].



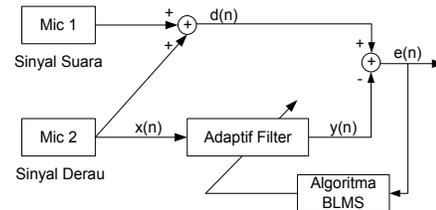
Gambar 1 Blok Diagram DSK TMS320C6416T

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.

III. RANCANGAN SISTEM

A. Pemodelan Sistem

Pemodelan sistem dapat digambarkan dalam bentuk blok diagram seperti yang diperlihatkan pada gambar 3.1. Sistem terdiri atas adaptive filter dan algoritma adaptasi. Struktur filter yang diimplementasikan adalah finite impulse respon (FIR). Algoritma adaptasi menggunakan algoritma BLMS dan dibandingkan dengan algoritma LMS. Kriteria dari kinerja blok ini adalah mendeteksi sinyal yang tidak diinginkan, yaitu sinyal noise.

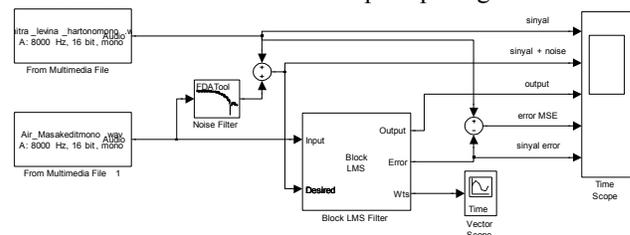


Gambar 2 Diagram Blok Sistem

Input pada mic 1 dimodelkan berupa sinyal informasi $s(n)$, sedangkan pada mic 2 dimodelkan dengan sinyal derau $n_0(n)$. Hasil campuran dari kedua sinyal tersebut $d(n)$ akan masuk pada sistem canceller. Disamping itu, noise $n_0(n)$ juga akan masuk ke dalam sistem filter adaptif dan menghasilkan output $y(n)$ yang merupakan replika terdekat yang mungkin dari $n_0(n)$. Output ini akan dikurangi dengan input $s(n)+n_0(n)$ untuk menghasilkan output sistem $z(n)= s(n)+n_0(n)-y(n)$ [8].

B. Implementasi Sistem Filter Adaptif BLMS

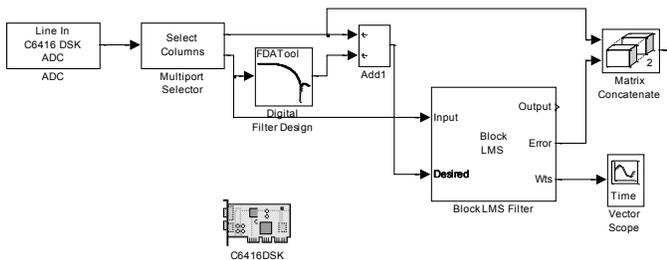
Secara umum pemodelan sistem filter adaptif terbagi menjadi 3 bagian utama, yaitu sinyal masukan, proses filter adaptif, dan sinyal keluaran yang diinginkan, maka dapat dibuat simulasi Simulink Matlab seperti pada gambar 3.



Gambar 3 Diagram Blok Simulasi filter adaptif BLMS

Sinyal masukan berupa rekaman sinyal suara dan sinyal referensi untuk pengujian sistem dapat menggunakan blok *from multimedia file* untuk memanggil rekaman suara dan referensinya yang diperoleh dari simulink *signal processing blockset – signal processing sources – from multimedia file*. Pengaturan parameter dalam blok *from multimedia file* hanya dengan memasukkan nama file suara dalam bentuk *.wav*.

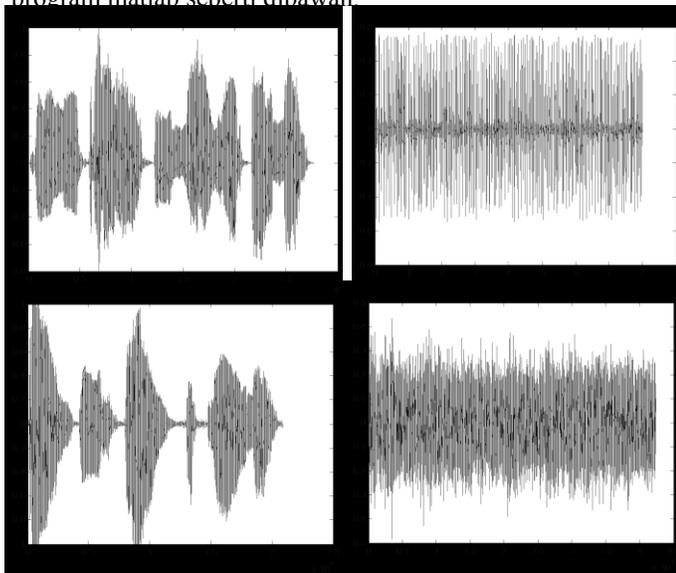
Pada blok diagram implementasi menggunakan sistem atau blok *multiport selector*. Keluaran blok *multiport selector* bagian atas merupakan *left channel*, sedangkan pada bagian bawahnya adalah *right channel*. *Left channel* digunakan sebagai input sinyal suara sedangkan *right channel* digunakan sebagai sinyal noise. Agar lebih jelasnya dapat melihat pada gambar 4 yang menunjukkan blok implementasi sistem *multiport*.



Gambar 4 Rangkaian blok Implementasi Sistem Multiport

C. Pengujian dan Analisa

Pada pengujian sistem kali ini akan digunakan sample sinyal input suara yang telah di plot dengan menggunakan program matlab seperti dibawah



Gambar 5 Tampilan empat sinyal input

Dari keempat suara yang dipakai sebagai input tersebut mempunyai spesifikasi yang sama, yaitu :

- Bit rate : 128 kbps
- Audio sample size : 16 bit
- Channels : 1 (mono)
- Audio sample rate : 8 kHz
- Audio format : PCM

Pengujian dan analisa pada sistem kali ini akan dibagi menjadi 4 skema, diantaranya yaitu :

a. Pengujian algoritma BLMS secara simulink.

Pengujian dengan menggunakan program simulink ini nantinya menghasilkan output 5 plot sinyal yang nantinya akan dianalisis. 5 plot sinyal tersebut diantaranya.

- Plot pertama merupakan gambar sinyal suara,
- Plot kedua merupakan sinyal pengganggu,
- Plot ketiga merupakan sinyal hasil campuran antara sinyal suara dan sinyal pengganggu,
- Plot keempat merupakan sinyal selisih dari sinyal suara dengan sinyal outputnya,
- Plot kelima merupakan sinyal error atau sinyal estimasi keluaran dari filter adaptif.

Kelima tampilan tersebut dititik beratkan pada plot keempat yang merupakan hasil keluaran sinyal errornya. Untuk dapat dengan mudah membandingkan hasil keluaran sinyal yang dihasilkan sistem pada pengujian secara simulasi dengan simulink maka sistem akan diberi masukan dengan variable tetap berupa panjang sinyal dan variable ubah berupa konstanta *step size* μ .

Tabel 2
Tabel Variasi Parameter

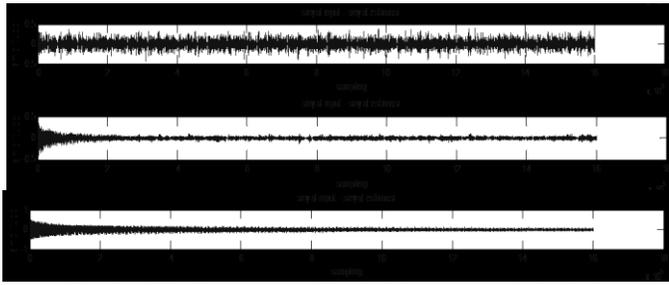
Simulasi	Length	μ
BLMS_I	16	0.05
BLMS_II	16	0.005
BLMS_III	16	0.0005
BLMS_IV	32	0.05
BLMS_V	32	0.005
BLMS_VI	32	0.0005
BLMS_VII	64	0.05
BLMS_VIII	64	0.005
BLMS_IX	64	0.0005

Berikut adalah contoh plot sinyal hasil keluaran simulasi filter adaptif menggunakan algoritma BLMS dengan *length* 32 dan *step size* 0.005.



Gambar 6 Filter length 32 step size 0.005

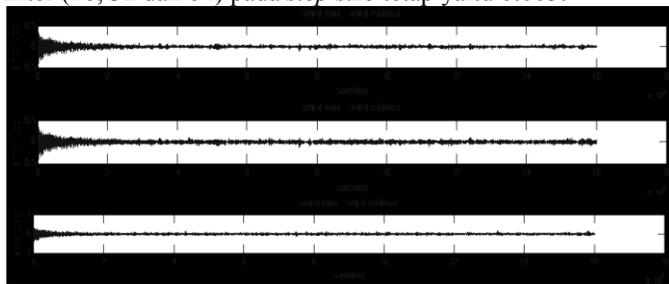
Tampilan plot sinyal hasil keluaran simulasi filter adaptif menggunakan algoritma BLMS dengan *step size* yang berbeda sebagai berikut.



Gambar 7 hasil output error Simulasi dengan *step size* berbeda

Pengamatan dilihat dengan membandingkan plot atau gambar keempat yang merupakan perubahan errornya. Sumbu x merupakan hasil samplingnya, sedangkan sumbu y merupakan amplitudo sinyal. Sampling pada simulasi ini menggunakan batas sampling hingga 16×10^5 . Amplitudo sinyal input dan noise maksimal sebesar 1. Dari hasil simulasi yang di dapat diatas, terlihat bahwa hasil amplitudo keluaran error dengan parameter *step size* 0.05 masih belum stabil. Karena sinyal yang ditampilkan tidak menuju atau mendekati nilai 0 pada sumbu y. Kesatabilan mulai terlihat dengan *step size* sebesar 0.005 dengan sinyal error yang mendekati nilai 0.

Pada simulasi selanjutnya dengan mengubah panjang filter (16, 32 dan 64) pada *step size* tetap yaitu 0.005.



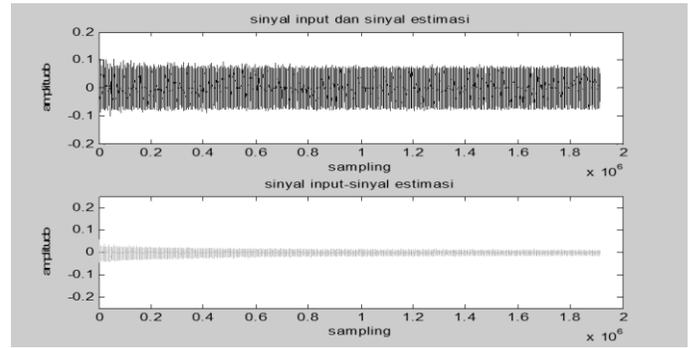
Gambar 8 hasil output error Simulasidengan panjang filter berbeda

b. Pengujian secara implementasi.

Setelah melakukan pengujian sistem dengan simulasi, maka langkah selanjutnya adalah melakukan pengambilan data dan pengujian sistem yang telah diimplementasikan kedalam TMS. Hal ini dimaksudkan untuk membuktikan apakah hasil simulasi sama dengan hasil implementasi. Dengan menggunakan tabel variasi parameter yang sama, pada pengujian secara implementasi ini akan ditampilkan output plot sinyal diantaranya :

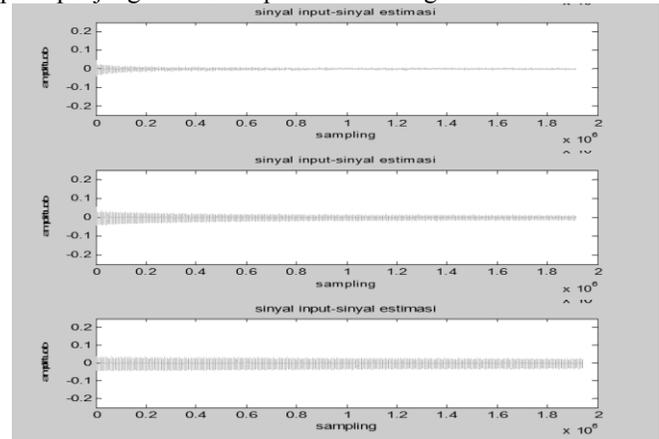
- Tampilan pertama merupakan sinyal gabungan dari sinyal suara dan sinyal referensinya,
- Tampilan kedua merupakan perubahan errornya.

Berikut adalah contoh plot sinyal hasil keluaran implementasi sistem filter adaptif menggunakan algoritma BLMS dengan *length* 32 dan *step size* 0.005.



Gambar 9 Filter lengh 32 *step size* 0.005

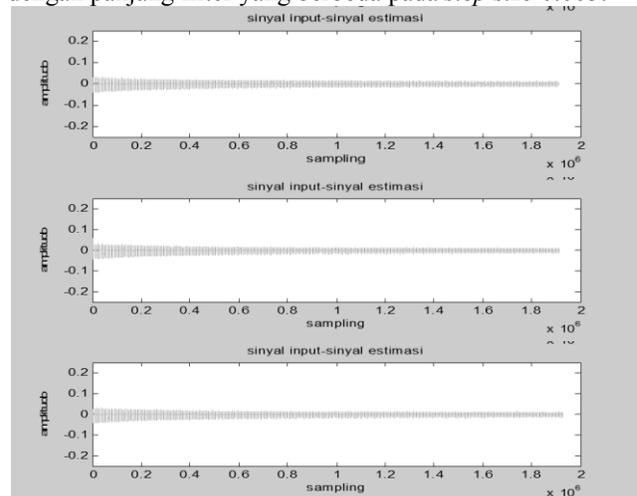
Plot sinyal hasil keluaran implementasi filter adaptif menggunakan algoritma BLMS dengan *step size* yang berbeda pada panjang filter 32 dapat dilihat sebagai berikut.



Gambar 10 hasil output error Implementasi dengan *step size* berbeda

Pengamatan dilihat dengan membandingkan plot atau gambar kedua yang merupakan perubahan errornya. Berbeda dengan hasil simulasi sebelumnya, hasil implementasi yang di dapat diatas, terlihat bahwa hasil keluaran error dengan parameter *step size* 0.05 terlihat sudah stabil. Karena sinyal yang ditampilkan menuju atau mendekati nilai 0 pada sumbu y.

Berikut adalah plot sinyal hasil keluaran implementasi dengan panjang filter yang berbeda pada *step size* 0.005.



Gambar 11 hasil output error Implementasi dengan panjang filter berbeda

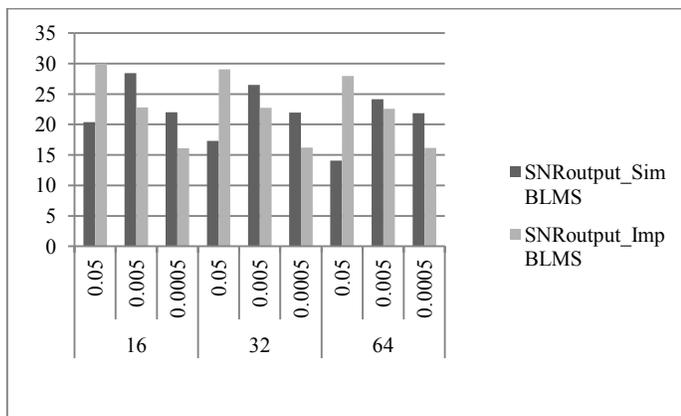
Dari hasil output error dengan panjang filter yang berbeda belum terlihat perbedaannya secara tampilan visual. Perbedaan akan terlihat pada pengukuran SNR yang akan dibahas lebih lanjut.

c. Pengujian SNR pada input dan SNR pada output.

Perhitungan SNR dilakukan dengan bantuan program matlab dalam bentuk *m-file* dan ditampilkan nilai dari SNR input dan SNR output dalam bentuk satuan *decibel* (dB).

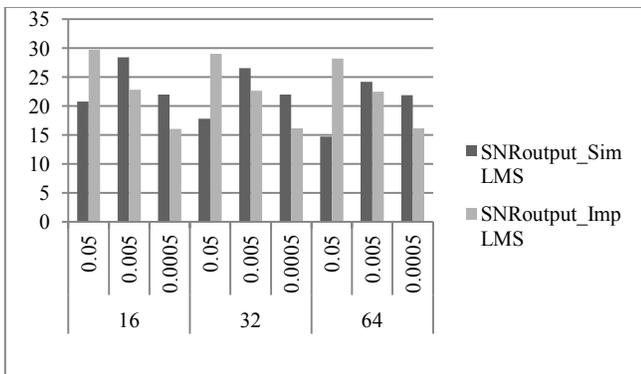
Tabel 3
Hasil SNR output keseluruhan

L	μ	SNR	SNR	SNR	SNR
		output Sim LMS	output Sim BLMS	output Imp LMS	output Imp BLMS
16	0.05	20,7692	20,4076	29,7708	30,0193
16	0.005	28,4472	28,4183	22,8399	22,781
16	0.0005	22,017	22,0149	16,0633	16,1109
32	0.05	17,8197	17,305	29,0471	29,0395
32	0.005	26,5607	26,5201	22,6463	22,7262
32	0.0005	21,9802	21,9778	16,1527	16,2057
64	0.05	14,7477	14,0688	28,1839	27,9871
64	0.005	24,1973	24,1443	22,5139	22,5972
64	0.0005	21,8615	21,8586	16,1616	16,1566



Gambar 12 Perbandingan SNR simulasi dan Implementasi algoritma BLMS

Dari tabel dan grafik diatas terlihat bahwa nilai terbesar SNR output secara simulasi adalah 28,4183 dB dengan parameter panjang filter 16 dengan *step size* sebesar 0.005, sedangkan nilai terbesar SNR output secara implementasi sebesar 30,0193 dB dengan parameter panjang filter 16 pada *step size* 0.05. Tetapi pada parameter *step size* 0.05 keluaran sinyal secara visual dan didengar masih belum stabil.



Gambar 13 Perbandingan SNR simulasi dan Implementasi algoritma LMS

Dari tabel dan grafik diatas terlihat bahwa nilai terbesar SNR output secara simulasi adalah 28,4472 dB dengan parameter panjang filter 16 dengan *step size* sebesar 0.005, sedangkan nilai terbesar SNR output secara implementasi sebesar 29,7708 dB dengan parameter panjang filter 16 pada *step size* 0,05. Pola keluaran SNR pada algoritma LMS ini sama dengan Algoritma BLMS dimana pada parameter *step size* 0.05 masih belum stabil tetapi pada implementasi menghasilkan nilai SNR yang lebih besar.

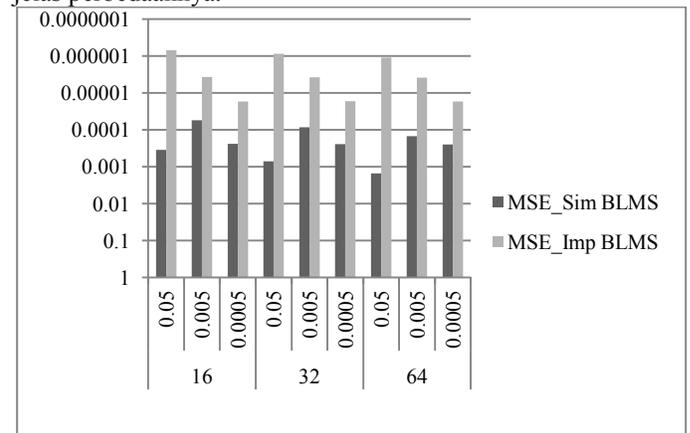
d. Pengujian MSE.

Hasil perhitungan MSE menggunakan bantuan program matlab sehingga didapat langsung nilai MSE dari masing-masing parameter. Setelah di dapat nilai MSE lalu hasil tersebut di-plot pada grafik dengan sumbu y merupakan nilai MSE, sedangkan sumbu x adalah perubahan parameternya.

Tabel 4
Perbandingan MSE algoritma LMS dan BLMS

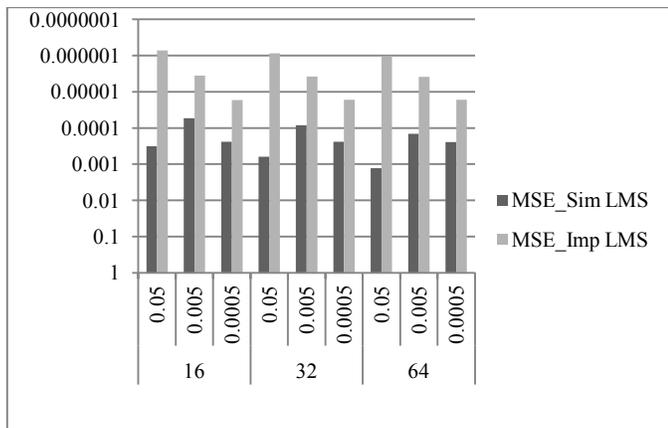
L	μ	MSE_Sim	MSE_Sim	MSE_Imp	MSE_Imp
		LMS	BLMS	LMS	BLMS
16	0.05	0,00031824	0,00034587	7,256E-07	6,9129E-07
	0.005	0,00005432	5,4682E-05	3,589E-06	3,6696E-06
	0.0005	0,00023877	0,00023888	1,709E-05	1,7058E-05
32	0.05	0,00062763	0,0007066	8,623E-07	8,6804E-07
	0.005	8,3871E-05	8,4658E-05	3,77E-06	3,7189E-06
	0.0005	0,0002408	0,00024093	1,682E-05	1,6675E-05
64	0.05	0,0013	0,0015	1,054E-06	1,1077E-06
	0.005	0,00014453	0,0001463	3,894E-06	3,8327E-06
	0.0005	0,00024747	0,00024764	1,679E-05	1,6871E-05

Dari tabel diatas dapat dibuat grafik perbandingan nilai MSE secara simulasi dan implementasi sehingga terlihat lebih jelas perbedaannya.



Gambar 14 Perbandingan MSE simulasi dan Implementasi algoritma BLMS

Dari tabel dan grafik diatas dapat dilihat bahwa nilai MSE minimum secara simulasi sebesar 5,47E-05 dengan parameter panjang filter 16 pada *step size* 0,005, sedangkan nilai MSE minimum untuk implementasi didapatkan nilai 6,9129E-07 dengan parameter panjang filter 16 pada *step size* 0.05. Hasil grafik menunjukkan nilai MSE pada implementasi lebih kecil daripada nilai MSE pada Simulasi, hal ini dikarenakan adanya peredaman sinyal pada sistem dengan menggunakan blok multiport selektor. Semakin kecil nilai MSE akan mempengaruhi nilai SNR output yang lebih besar.



Gambar 15 Grafik perbandingan nilai MSE simulasi dengan implementasi algoritma LMS

Untuk algoritma LMS, nilai MSE minimum secara simulasi sebesar $5,43E-05$ dengan panjang filter 16 pada *step size* 0,005 sedangkan secara implementasi MSE dengan nilai $7,2559E-07$ dengan parameter panjang filter 16 pada *step size* 0.05. karena hasil keluaran algoritma BLMS hampir mendekati hasil dari algoritma LMS, sehingga tampilan grafik nilai MSE nya hampir sama yaitu nilai MSE implementasi lebih kecil dari nilai hasil simulasinya karena adanya redaman pada sistemnya.

Semakin kecil nilai MSE maka semakin bagus performansi hasil error nya. Dari hasil nilai MSE yang didapat secara keseluruhan antara algoritma LMS dan BLMS, dapat disimpulkan bahwa nilai MSE pada Algoritma BLMS mendekati nilai MSE algoritma LMS. Hal ini sesuai dengan teori yang menyatakan bahwa hasil output yang dihasilkan algoritma BLMS akan sama dengan atau mendekati algoritma LMS.

IV. 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 Tugas Akhir ini telah berhasil diimplementasikan sistem pereduksi derau berbasis TMS320C6416T menggunakan algoritma LMS dan BLMS.
2. Hasil simulasi BLMS yang dirancang dengan menggunakan input suara rekaman oleh simulink matlab menghasilkan MSE minimum sebesar $5,47E-05$ dengan panjang filter 16 pada *step size* 0,005 sedangkan pada hasil implementasi menghasilkan MSE minimum sebesar $6,9129E-07$ dengan panjang filter 16 pada *step size* 0,05.
3. Dari nilai minimum MSE BLMS didapat nilai SNR paling besar pada simulasi yaitu sebesar 28,4183 dB dan pada implementasi sebesar 30,0193 dB.

4. Dalam tugas akhir ini, jika parameter *step size* diperkecil hingga 0.0005 maka filter membutuhkan waktu yang lebih lama untuk beradaptasi terbukti dengan nilai SNR terkecil pada implementasi BLMS sebesar 16.1109 dB. Namun dengan *step size* diperkecil maka akan didapatkan respon sinyal yang mendekati sinyal yang diinginkan.

DAFTAR PUSTAKA

- [1] Nurwati, Tri., *Modul Pengolahan Sinyal*, BAB V FILTER.
- [2] Haykin, Simon., *Adaptive Filter Theory 3rd Edition*, Prentice Hall, 1995.
- [3] Novi Tomponu, Alan., *Implementasi Algoritma Least Mean Square untuk Peningkatan Kualitas Suara Penderita Tuna Laring Berbasis Processor TMS320C6713*, Jurnal Teknomatika, Poltek Negeri Sriwijaya, Palembang, Vol. 2, No. 2, Mei 2012.
- [4] M.S.E.Abadi, S.Z.Mousavi, A.Hadei, *Variable Step-Size Least Mean Square Adaptive Filters*, ICIS, August 2006.
- [5] Gregory A, Clark. *Block Implementation of Adaptive Digital Filters*. IEEE Trans. Vol, ASSP-29. No.3. June 1981.
- [6] Chassaing, Rulph., *Digital Signal Processing And Applications With The C6713 And C6416 Dsk*.
- [7] Spectrum Digital., *TMS320C6416T DSK Technical Reference*, INC. 2004.
- [8] Widrow, Bernard, et. Al., *Adaptive Noise Canceling Principles and Applications*, Proceedings of the IEEE, Vol.63, No.12, Page(s):1692-1716, 1975.