

## PENGARUH *COMPRESSIVE SENSING* PADA KINERJA FILTER ADAPTIF BERBASIS *LMS* UNTUK PENGHILANG DERAU

### *EFFECT OF COMPRESSIVE SENSING ON ADAPTIVE FILTER PERFORMANCE BASED ON LMS FOR NOISE REMOVER*

Sholichatur Rizkiyah<sup>1</sup>, Gelar Budiman<sup>2</sup>, Indrarini Diah Irawati<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

sholichaturrizikiyah@student.telkomuniversity.ac.id<sup>1</sup>, gelarbudiman@telkomuniversity.ac.id<sup>2</sup>,

indrarini@telkomuniversity.ac.id<sup>3</sup>

#### Abstrak

Masyarakat memanfaatkan teknologi untuk bertukar informasi. *File-file* yang dikirimkan atau diterima berukuran cukup besar, sehingga membutuhkan media penyimpanan cukup besar. Pertukaran informasi banyak dilakukan menggunakan internet akan membuat data lebih mudah terserang *noise*. Untuk menghindari hal tersebut digunakanlah metode *Compressive Sensing* (CS) dan *filter* digital. Pada Penelitian ini dilakukan analisa pengaruh CS terhadap kinerja *filter* adaptif. CS berfungsi untuk mengkompresi *audio host* yang sudah tercampur dengan *noise*, setelah proses CS sinyal hasil rekonstruksi CS dilakukan pemfilteran dengan *filter* adaptif. Pada tahap akuisisi CS digunakan metode DCT untuk mengubah sinyal dari domain waktu ke frekuensi, proses rekonstruksi menggunakan metode *Orthogonal Matching Pursuit* (OMP). *Filter* adaptif menggunakan algoritma *Least Mean Square* (LMS). Pengamatan dilakukan terhadap orde *filter* yang merupakan tingkatan *filter*,  $N$  merupakan nilai panjang awal sinyal *audio*,  $\mu$  merupakan koefisien kecepatan *filter*,  $L$  merupakan panjang *audio* yang diakuisisi serta  $M$  adalah panjang vektor dari sampel CS. Hasil yang diperoleh dari penelitian ini adalah CS mempengaruhi kinerja *filter* adaptif berbasis *LMS*. Sistem bekerja maksimal pada nilai koefisien kecepatan *filter* = 0.005, rasio kompresi 50% dan  $N=80000$ . Semakin kecil nilai  $M$  yang digunakan nilai SNR yang di hasilkan *filter* semakin menurun dan semakin kecil parameter  $L$  yang digunakan SNR yang dihasilkan semakin besar.

**Kata kunci :** *Compressive Sensing, Filter Adaptif, Least Mean Square, Orthogonal Matching Pursuit.*

#### Abstract

People use technology to exchange information. The files that are sent or received are pretty large, so they require large enough storage media. The exchange of data using the internet will make the data more susceptible to noise. The solution to this problem, *Compressive Sensing* (CS) and digital filters, are used. In this study, an analysis of the influence of CS on the performance of the adaptive filter was carried out. CS serves to compress the host audio that has been mixed with noise. After the CS process, the signal from the CS reconstruction was filtered with an adaptive filter. In the CS acquisition stage, the DCT method is used to convert the signal from the time domain to the frequency. In contrast, the reconstruction process uses the *Orthogonal Matching Pursuit* (OMP) method. The adaptive filter uses the *Least Mean Square* (LMS) algorithm. Observations were made on the filter order, which is the filter level.  $N$  is the initial length of the audio signal, is the filter speed coefficient,  $L$  is the length of the acquired audio, and  $M$  is the vector length of the CS sample. The results obtained from this study are that CS affects the performance of LMS-based adaptive filters. The system works optimally at the filter speed coefficient = 0.005. The compression ratio is 50% and  $N=80000$ . The smaller the  $M$  value used, the SNR value produced by the filter decreases, and the smaller the  $L$  parameter used, the greater the SNR created.

**Keywords:** *Compressive Sensing, Filter Adaptif, Least Mean Square, Orthogonal Matching Pursuit.*

#### 1. Pendahuluan

Teknologi terus berkembang pesat di berbagai bidang saat ini, begitu pula dengan bidang komunikasi dan informasi hampir setiap individu memanfaatkan teknologi ini setiap hari. Banyak sekali pertukaran informasi yang terjadi setiap hari. Mulai dari informasi dalam bentuk teks, gambar, *audio* maupun video. Saat ini pertukaran informasi hampir semua dilakukan menggunakan internet, file-file yang dikirimkan dalam ukuran yang besar dan dalam jumlah yang cukup banyak, hal ini akan menyebabkan kebutuhan *bandwidth* yang lebar dalam pentransmisiannya dan juga membutuhkan media penyimpanan atau *storage* yang cukup besar. Selain itu data-data yang dikirimkan pun rentan untuk terserang *noise* sehingga akan menimbulkan kerusakan atau kecacatan pada data, hal ini akan merugikan bagi pihak pengirim dan penerima, karena informasi yang ingin disampaikan tidak bisa tersampaikan secara utuh karena data mengalami kerusakan.

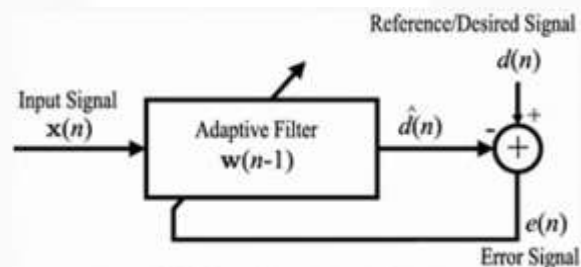
Namun saat ini sudah banyak teknik yang dikembangkan untuk mengatasi hal tersebut, salah satu teknik yang banyak digunakan saat ini yaitu teknik *Compressive Sensing* (CS), dimana data-data dalam ukuran yang besar akan dikompres menjadi ukuran yang lebih kecil agar lebih menghemat *bandwidth* transmisi dan juga menghemat media penyimpanan. Dan banyaknya data rusak yang diakibatkan oleh *noise* ini dapat diatasi dengan menggunakan

*filter*, salah satunya yaitu *filter* adaptif dimana *filter* adaptif merupakan sebuah *filter* dengan pengatur koefisien, pada proses penekanan derau parameter *filter* diatur sehingga dapat mengoptimalkan sinyal dari distorsi seminimal mungkin. *Filter* adaptif ini memiliki bermacam algoritma, dan salah satu algoritma yang sering digunakan yaitu algoritma *Least Mean Square* (LMS) [1]. Pada Tugas Akhir ini, akan dilakukan penelitian lebih mendalam mengenai bagaimana pengaruh CS terhadap kinerja *filter* adaptif berbasis LMS untuk penghilang derau. Dimana dalam prosesnya data akan diberikan *noise* terlebih dahulu sebelum dilakukan proses akuisisi CS, setelah itu data dan *noise* yang sudah melalui proses akuisisi CS akan dilakukan proses rekonstruksi CS pada penerima. Kemudian di-filter menggunakan filter adaptif dengan algoritma LMS, setelah itu akan dilihat bagaimana kualitas data yang dihasilkan apakah data tersebut masih baik dengan parameter-parameter yang ditentukan dan apakah *noise* dapat terfilter oleh *filter* adaptif LMS walaupun *noise* sudah dalam kondisi yang ikut dalam proses CS bersama dengan data sebelumnya, dan setelah itu akan dilakukan analisis dan penarikan kesimpulan untuk mengetahui seberapa besar pengaruh CS pada *filter* adaptif untuk menghilangkan derau berbasis LMS.

## 2. Dasar Teori

### 2.1 Filter Adaptif

*Filter* adaptif adalah sebuah *filter* digital. *Filter* menjadi adaptif ketika koefisien diperbarui (menggunakan algoritma tertentu) setiap kali sampel baru dari sinyal tersedia. Secara konseptual, algoritma yang dipilih harus dapat mengikuti evolusi dari sistem yang diteliti. Hal ini memberikan *filter* adaptif, kemampuan untuk beroperasi dalam waktu nyata dan meningkatkan kinerjanya tanpa perlu melibatkan pengguna. Gambar 2.1 menunjukkan skema masalah pemfilteran adaptif. Pada dasarnya, *filter* adaptif menerima sinyal *input*  $X(n)$  dan sinyal *referensi* / sinyal yang diinginkan  $d(n)$ . Pada saat  $n$ , koefisien *filter*  $w(n-1) = [w_0(n-1), w_1(n-1), \dots, w_{L-1}(n-1)]^T$ , dengan  $T$  menunjukkan *transposisi*, akan diperbarui sesuai dengan algoritma tertentu. Pada *filter* adaptif yang digunakan, banyak menggunakan beberapa algoritma [7].



Gambar 2. 1 Gambaran Sistem Filter Adaptif

### 2.2 Least Mean Square (LMS)

Algoritma LMS sejauh ini merupakan algoritma yang paling banyak digunakan dalam pemfilteran adaptif karena algoritma LMS memiliki kompleksitas yang rendah, dan kestabilan ketika diimplementasikan dengan *finite-precision arithmetic*. Algoritma LMS secara matematik dapat dilihat seperti berikut:

Algoritma LMS merupakan algoritma *steepest descent* dalam kondisi tertentu, dimana diketahui bahwa algoritma *steepest descent*:

$$\underline{W}_{n+1} = \underline{W}_n + \mu E\{e(n)\underline{x}^*(n)\} \quad (2.1)$$

Secara praktis Persamaan (2.1) tidak bisa dipakai karena umumnya  $E\{e(n)\underline{x}^*(n)\}$  tidak diketahui, sehingga dapat diestimasi dengan persamaan:

$$E\{e(n)\underline{x}^*(n)\} = \frac{1}{L} \sum_{l=0}^{L-1} e(n-l)\underline{x}^*(n-l) \quad (2.2)$$

Dari Persamaan (2.2) maka didapatkan algoritma *steepest descent*:

$$\underline{W}_{n+1} = \underline{W}_n + \mu \frac{1}{L} \sum_{l=0}^{L-1} e(n-l)\underline{x}^*(n-l) \quad (2.3)$$

Dalam kondisi tertentu bila  $L = 1$ , maka Persamaan (2.3) disebut algoritma LMS. Dalam hal ini:  $E\{e(n)\underline{x}^*(n)} = e(n)\underline{x}^*(n)$ , merupakan algoritma LMS untuk *filter adaptif non rekursif*, dengan parameter  $P =$  derajat *filter* dan  $\mu =$  ukuran langkah (*step size*), dan dengan inialisasi  $\underline{W}_0 = 0$ . Dan berikut adalah perhitungan algoritma LMS [14]:

$$\begin{aligned} \hat{d}(n) &= \sum_{k=0}^p w_n(k) \cdot x(n-k) = \underline{w}^T \underline{x}(n) \\ e(n) &= d(n) - \hat{d}(n) = d(n) - \underline{w}^T \underline{x}(n) \\ \underline{W}_{n+1} &= \underline{W}_n + \mu E\{e(n)\underline{x}^*(n)\} \rightarrow w_0(n+1) = w_0(n) + \mu e(n)x^*(n) \\ w_1(n+1) &= w_1(n) + \mu e(n)x^*(n-1) \\ w_p(n+1) &= w_p(n) + \mu e(n)x^*(n-p) \end{aligned} \quad (2.4)$$

### 2.3 Compressive Sensing

*Compressive Sampling* (CS) atau sering disebut *Compressive Sensing* merupakan sebuah teknik kompresi, yaitu dengan cara menggabungkan akuisisi dan kompresi dalam satu step. CS terdiri dari dua tahap, yaitu tahap akuisisi dan tahap rekonstruksi [9]. Secara matematis proses akuisisi dilakukan dengan cara perkalian matriks atau *linier*, yang ditunjukkan pada persamaan berikut:

$$Y = AX \quad (2.5)$$

$X \in \mathbb{R}^M$  dimana  $X$  adalah  $K$ -sparse pada  $\Psi$  dengan panjang vektor  $N$ , dan  $Y \in \mathbb{R}^M$  dimana  $Y$  adalah sinyal terkompresi dengan panjang vektor  $p$ . Sedangkan  $A \in \mathbb{R}^{M \times N}$  merupakan matriks yang digunakan dalam kompresi dengan ukuran  $M \times N$ . Persamaan (2.5) memiliki syarat bahwa  $K < M \ll N$ . [9].

### 2.4 Orthogonal Matching Pursuit

Dalam melakukan rekonstruksi CS dibutuhkan sebuah algoritma, salah satunya yaitu algoritma *Orthogonal Matching Pursuit* (OMP). Algoritma OMP termasuk dalam *greedy algorithm* dimana *greedy algorithm* ini bersifat *iterative* sampai mendapatkan nilai akhir yang optimal. OMP banyak digunakan dalam proses rekonstruksi CS karena kecepatan rekonstruksinya paling cepat [8].

Pada proses rekonstruksi ini, OMP akan mendapatkan sinyal  $x$  dari sinyal  $y$  yang sudah dikompresi. Dan untuk mendapatkan sinyal  $x$  maka ada tahapan-tahapan seperti berikut:

#### a. Calculate Contribution

Pada proses *calculate contribution* dilakukan proses mencari nilai atom yang paling besar ketika matriks  $A^T$  dikalikan dengan nilai  $y$ . Matriks  $A$  adalah matriks yang digunakan dalam proses kompresi.

$$w = A^T \cdot y \quad (2.10)$$

$$\begin{bmatrix} a & a \\ b & b \\ c & c \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (2.11)$$

Pada persamaan (2.10) akan didapatkan nilai  $w$ , dari nilai  $w$  tersebut kemudian dipilih nilai yang paling besar, pada penulisan ini diumpamakan nilai yang dipilih paling besar adalah  $\lambda_1$ . Nilai  $\lambda$  yang paling besar menunjukkan koherensi yang rendah terhadap  $y$ , maka basis yang dipilih adalah basis dengan nilai  $\lambda$  yang paling besar. Pada penulisan ini kita umpamakan adalah  $b_i$ . Gambar 2.3 menunjukkan ilustrasi antara basis dan  $y$  [10].

b. **Calculating Residue**

Proses ini yaitu dengan melakukan perhitungan *residue* antara  $y$  dengan basis yang sudah terpilih sebelumnya. Dari gambar 2.5 dapat diketahui bahwa residu tegak lurus dengan *contribution* basis pertama yang terpilih [10].

$$r_i = y - \lambda_1 \cdot \langle b_1, y \rangle \tag{2.12}$$

c. **Repeat the Iteration**

Proses ini dilakukan untuk mencari nilai atom yang paling besar ketika dikalikan dengan nilai  $y$  dengan menggunakan  $r_i$ . Pada proses ini menggunakan  $b_1$  untuk dijadikan sebuah basis baru, disini kita lambangkan sebagai  $A_{new}$ . Setelah itu dilakukan proses *selected* basis kembali dengan menggunakan nilai residu pada persamaan (2.12) sebagai pengali.

$$A_{new} = [b_1] = \begin{bmatrix} a \\ a \end{bmatrix} \tag{2.13}$$

$$w = \frac{A^T \cdot r_i}{b} = \begin{bmatrix} \lambda_2 \\ \lambda_3 \end{bmatrix} \tag{2.14}$$

$$\begin{bmatrix} b \\ c \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \lambda_2 \\ \lambda_3 \end{bmatrix} \tag{2.15}$$

Dari proses di atas diumpamakan nilai terbesar yaitu  $\lambda_2$ , maka basis yang *contribute* adalah  $b_2$ . Setelah itu basis yang terpilih yaitu  $b_1$  dan  $b_2$  dikumpulkan pada matrik  $A_{new}$  dan nilai  $r$  dikumpulkan menjadi matriks baru yaitu  $x_{rec}$ .

$$A_{new} = [b_1 \quad b_2] = \begin{bmatrix} a & b \\ a & b \end{bmatrix} \tag{2.16}$$

$$x_{rec} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 0 \end{bmatrix} \tag{2.17}$$

Setelah proses (2.13),(2.14),(2.15),(2.16) kemudian dilakukan lagi pencarian residu dengan persamaan berikut :

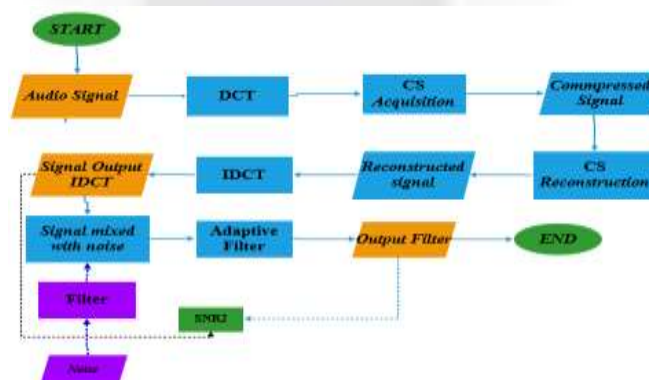
$$r = y - A_{new} \cdot \lambda \tag{2.18}$$

Proses ini akan berhenti ketika didapatkan nilai residu 0, setelah mendapatkan hasil residu 0 maka tidak perlu dilakukan iterasi lagi. Dan disini sinyal hasil rekonstruksi yang didapatkan adalah  $x_{rec}$  [10].

$$x_{rec} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 0 \end{bmatrix} \tag{2.19}$$

**3 Perancangan Sistem**

Perancangan sistem pada Tugas Akhir ini dibagi menjadi tiga tahapan, yaitu proses akuisisi, proses rekonstruksi dan proses denoising. Proses pertama adalah proses akuisisi *Compressive Sensing* (CS) dimana *file audio* yang sudah tercampur dengan *noise* akan dilakukan CS. Setelah didapatkan *file* yang terkompresi, kemudian dilakukan proses tahap kedua yaitu proses rekonstruksi CS. Pada proses ini dilakukan pengembalian sinyal ke bentuk awal seperti sebelum dilakukan proses akuisisi CS. Setelah itu, dilakukan proses pemfilteran untuk memisahkan antara sinyal *audio* dengan *noise*. Hasil pemfilteran ini dapat dihitung parameter SNR dan MSE.



Gambar 3. 1 Diagram Alir Sistem

Beikut adalah langkah – langkah dari proses Tugas Akhir ini:

1. Membaca sinyal *audio*.
2. Merubah sinyal dari domain waktu ke domain frekuensi menggunakan transformasi DCT.
3. Setelah didapatkan sinyal dalam domain frekuensi, kemudian dilakukan proses akuisisi CS.
4. Setelah didapatkan sinyal hasil kompresi, dilakukan proses rekonstruksi CS menggunakan metode OMP.
5. Sinyal hasil rekonstruksi dirubah ke bentuk sinyal domain waktu menggunakan transformasi IDCT.
6. Sinyal hasil IDCT di tambahkan dengan *noise* terfilter.
7. Sinyal yang sudah tercampur dengan noise kemudian dilakukan filtering untuk memisahkan antara sinyal *audio* dan *noise* menggunakan *filter* adaptif dengan algoritma LMS.

Perhitungan algoritma LMS:

- a. Pembacaan sinyal *input* yang berupa sinyal *audio*.
  - b. Penentuan nilai *koefisien kecepatan filter* dan *orde filter*.
  - c. Perhitungan sinyal keluaran  $y(n) = x(n)*w(n)$ .
  - d. Perhitungan nilai *audio estimation*  $e(n) = d(n) - y(n)$ .
  - e. Perbaikan *weight filter*  $w(n+1) = w(n) + \mu x(n)e(n)$ .
8. Melakukan perhitungan nilai SNR dari sinyal *output* hasil pemfilteran.

#### 4. Pengujian dan Analisis Sistem

Bab ini menjelaskan mengenai hasil pengujian sistem *filter* adaptif dengan algoritma LMS menggunakan *Compressive Sensing* (CS). Pengujian dan analisis ini dilakukan dengan tujuan untuk mengetahui bagaimana pengaruh CS pada kinerja *filter* adaptif dengan algoritma LMS. Pengujian dilakukan dengan menggunakan *host* berupa *audio* dengan format *.wav*. *Host* yang digunakan pada pengujian ini adalah *Africa-toto.wav*. Analisis ini dilakukan dengan melakukan perubahan parameter disetiap pengujian. Parameter yang digunakan untuk melakukan pengujian ini adalah *orde filter*,  $N$ , koefisien kecepatan *filter*,  $M$ , dan  $L$ . Hasil pengujian akan dibandingkan dengan hasil proses pemfilteran tanpa menggunakan CS. SNR1 pada penelitian ini adalah hasil perhitungan nilai SNR dari proses pemfilteran sinyal hasil rekonstruksi CS yang tidak terganggu dengan *noise*. SNR2 adalah hasil dari perhitungan SNR dari proses pemfilteran sinyal hasil rekonstruksi CS yang terganggu *noise*. Sedangkan SNR3 adalah hasil dari perhitungan SNR dari proses pemfilteran tanpa dipengaruhi CS.

##### 4.1 Analisis Pengaruh Parameter $M$

Parameter  $M$  merupakan panjang vektor dari sampel CS. Perhitungan proses CS didapatkan menggunakan persamaan (2.5). Panjang dari  $y$  merupakan hasil dari  $M \times I$ . Dalam penelitian ini dilakukan analisa mengenai pengaruh  $M$  yang diubah nilainya terhadap hasil dari CS. Pada pengujian ini dilakukan pengujian nilai SNR1 dan SNR2 yang dihasilkan. Hasil pengujian menunjukkan bahwa nilai SNR1 tidak mengalami perubahan, nilai SNR1 mengalami kestabilan pada nilai 311,492 dB. Sedangkan pada SNR2 akan bernilai semakin besar apabila rasio kompresi yang digunakan mulai dari 37,5% – 62,5%, dari pengujian ini dapat disimpulkan bahwa parameter  $M$  mempengaruhi kinerja *filter* dalam menghilangkan *noise*, karena apabila parameter  $M$  berubah maka rasio kompresi yang dihasilkan akan berubah. Parameter  $M$  ini tidak mempengaruhi *weight filter*, dimana ketika dilakukan perubahan nilai parameter  $M$ , maka *weight filter* yang dihasilkan akan sama di setiap nilai  $M$  yang digunakan, dan keakurasian dari *actual weight* dan *estimated weight* memiliki keakurasian yang sama dalam setiap nilai  $M$  yang digunakan



**Tabel 4. 1 Hasil Pengujian Parameter M**

No	M	SNR2	Rasio Kompresi	MSE
1	5	4.301	1.2	0.015
2	10	4.880	2.5	0.014
3	50	4.124	12.5	0.016
4	100	4.538	25	0.014
5	150	5.245	37.5	0.013
6	200	5.012	50	0.013
7	250	5.220	62.5	0.014
8	300	4.542	75	0.015
9	350	4.718	87.5	0.014
10	400	4.777	100	0.014

#### 4.2. Analisis Pengaruh Parameter L

Pada pengujian kali ini dilakukan perubahan untuk parameter  $L$ , dimana  $L$  merupakan panjang dari sinyal *audio* yang diakuisisi. Pada pengujian ini dilakukan perubahan nilai  $L$  mulai dari 100 sampai dengan 9000. Dari Tabel 4.2 dapat diketahui bahwa rasio kompresi akan mengalami perubahan ketika nilai  $L$  berubah, nilai rasio kompresi yang dihasilkan akan semakin kecil apabila nilai  $L$  yang digunakan semakin besar.

Perubahan parameter  $L$  juga mempengaruhi nilai SNR yang dihasilkan. Dari pengujian ini dapat kita ketahui bahwa kinerja filter akan semakin menurun apabila nilai  $L$  yang digunakan semakin besar. Parameter  $L$  ini tidak mempengaruhi weight filter, ketika dilakukan perubahan nilai parameter  $L$ , maka weight filter yang dihasilkan akan sama disetiap nilai  $L$  yang digunakan, dan keakurasian dari actual weight dan estimated weight memiliki keakurasian yang sama dalam setiap nilai  $L$  yang digunakan

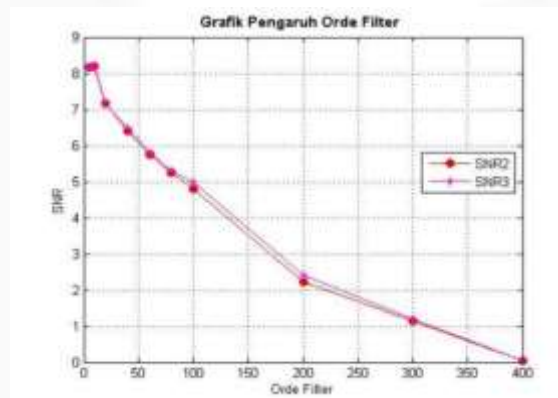
**Tabel 4. 2 Hasil Pengujian Parameter L**

No	L	SNR1 (dB)	SNR2 (dB)	Rasio Kompresi (%)	MSE
1	100	315.222	4.926	50	0.014
2	200	312.838	5.074	25	0.014
3	400	311.491	4.588	12.5	0.015
4	500	309.334	4.373	10	0.015
5	1000	309.832	4.641	5	0.014
7	2000	309.018	4.719	2.5	0.014
9	3000	308.813	4.621	1.66	0.016
11	4000	307.878	4.336	1.25	0.019
13	5000	308.300	3.889	1	0.016

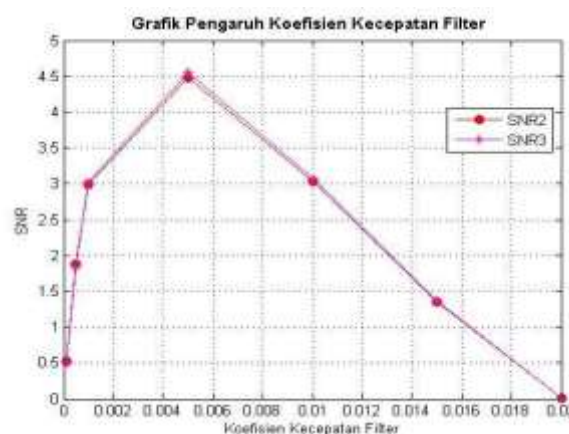
#### 4.2 Analisis Pengaruh CS dan Tanpa CS

Pada penelitian ini dilakukan perbandingan antara SNR2 dan SNR3 untuk mengetahui bagaimana kinerja *filter* apabila sinyal *input* melalui proses CS dan tanpa melalui proses CS, dalam pengujian ini dilakukan pengambilan sampel SNR sebanyak 10 kali disetiap nilai parameter yang dirubah, kemudian diambil nilai rata-rata. SNR2 adalah hasil SNR dengan sinyal *input* melalui proses CS terganggu *noise* sebelum dilakukan proses pemfilteran, sedangkan untuk SNR3 adalah hasil SNR dari sinyal *input* tanpa melalui proses CS. Dalam pengujian ini parameter yang akan dilakukan perubahan adalah parameter *orde filter*,  $N$ , koefisien kecepatan *filter*.

Hasil pengujian parameter *orde filter* disajikan dalam grafik pada Gambar 4.1. Sehingga, dapat diketahui bahwa perubahan nilai *orde filter* pada proses pemfilteran menghasilkan nilai SNR3 yang lebih tinggi dibandingkan dengan SNR2. Hal ini menunjukkan bahwa proses CS sebelum pemfilteran berpengaruh terhadap nilai SNR yang dihasilkan. Hasil yang sama juga didapatkan pada pengujian parameter koefisien kecepatan *filter* yang menghasilkan nilai SNR3 lebih besar dibandingkan dengan nilai SNR2, seperti yang ditunjukkan pada grafik perbandingan SNR2 dan SNR3 pada Gambar 4.2. Dari pengujian yang dilakukan pada parameter *orde filter* dan koefisien kecepatan *filter* tersebut dapat diketahui bahwa proses filtering tanpa melalui proses CS menghasilkan nilai SNR yang lebih besar.



Gambar 4. 1 Pengaruh Orde Filter Terhadap SNR2 dan SNR3



Gambar 4. 2 Pengaruh Koefisien Kecepatan Filter Terhadap SNR2 dan SNR3

#### 5. Kesimpulan

Dari penelitian dengan judul “Pengaruh *Compressive Sensing* Pada Kinerja *Filter* Adaptif Berbasis LMS Untuk Penghilang Derau” dapat diambil kesimpulan bahwa, *Compressive Sensing* mempengaruhi kinerja *filter* adaptif berbasis LMS. Semakin kecil nilai  $M$  yang digunakan maka nilai SNR yang di hasilkan *filter*

semakin menurun dan semakin kecil nilai parameter  $L$  yang digunakan SNR yang dihasilkan semakin besar. Semakin besar nilai parameter  $N$  yang digunakan pada sistem maka SNR yang dihasilkan semakin baik. Sistem yang dirancang bekerja maksimal pada saat nilai koefisien kecepatan  $filter = 0.005$ , rasio kompresi 50% dan nilai parameter  $N = 80000$ . Hasil pemfilteran yang dipengaruhi *Compressive Sensing* menghasilkan nilai SNR yang lebih kecil dibandingkan dengan hasil pemfilteran yang tidak dipengaruhi proses *Compressive Sensing*.

## 6. Refrensi

- [1] S. Syahid, "Simulasi Penekanan Derau dengan Metode Finite Impulse Response (FIR) secara Adaptif Menggunakan Algoritma Least Mean Square (LMS)," *Rekayasa Apl. dan Peranc. Ind. – RAPI 2004*, pp. 381–388, 2004.
- [2] N. Sireesha, K. Chithra, and T. Sudhakar, "Adaptive filtering based on least mean square algorithm," *Int. Symp. Ocean Electron. SYMPOL*, pp. 42–48, 2013, doi: 10.1109/sympol.2013.6701910.
- [3] K. Prajna and Charishma, "Implementation of an adaptive filter robust to impulsive noise," *Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIECS 2017*, vol. 2018-January, pp. 1–4, 2018, doi: 10.1109/ICIECS.2017.8275950.
- [4] C. Ye, G. Gui, S. Matsushita, and L. Xu, "Robust stochastic gradient-based adaptive filtering algorithms to realize compressive sensing against impulsive interferences," *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 1946–1951, 2016, doi: 10.1109/CCDC.2016.7531301.
- [5] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Compressed sensing block MAP-LMS adaptive filter for sparse channel estimation and a Bayesian Cramer-Rao bound," *Mach. Learn. Signal Process. XIX - Proc. 2009 IEEE Signal Process. Soc. Work. MLSP 2009*, 2009, doi: 10.1109/MLSP.2009.5306268.
- [6] G. Budiman, A. B. Suksmono, and D. Danudirdjo, "Compressive sampling with multiple bit spread spectrum-based data hiding," *Appl. Sci.*, vol. 10, no. 12, 2020, doi: 10.3390/app10124338.
- [7] L. Vega and H. Rey, *A Rapid Introduction to Adaptive Filtering*, 1st ed. Springer-Verlag Berlin Heidelberg, 2013.
- [8] P. S. R. Diniz, *Adaptive Filtering Algorithms and Practical Implementation*. Springer US, 2013.
- [9] S. L. Brunton and J. N. Kutz, "Data Driven Science & Engineering - Machine Learning, Dynamical Systems, and Control." p. 572, 2017, [Online]. Available: databook.uw.edu.
- [10] Usman, Koredianto, "Introduction to Orthogonal Matching Pursuit". Telkom University, Agustus 2017. [Online]. Tersedia: <http://korediantousman.staff.telkomuniversity.ac.id> [Diakses: 26 November 2020].
- [11] K. Usman, H. Gunawan, and A. B. Suksmono, "Sparse signal reconstruction using weight point algorithm," *J. ICT Res. Appl.*, vol. 12, no. 1, pp. 35–53, 2018, doi: 10.5614/itbj.ict.res.appl.2018.12.1.3.
- [12] J. Park, C. H. Lee, B. S. Kim, and J. Laskar, "A low flicker noise CMOS mixer using two resonating inductors for direct conversion receivers," *IEEE MTT-S Int. Microw. Symp. Dig.*, pp. 1705–1708, 2006, doi: 10.1109/MWSYM.2006.249707.
- [13] H. G. Alfarizi. 2018. "Penghapusan Derau Pada Sinyal Wicara Menggunakan Sistem Least Mean Square". Bandung. Telkom University.
- [14] Tjondronegoro Suhartono (2005). *Pengolahan Sinyal Dijital Lanjut & Aplikasi Bab 5, Edisi sem II – 2004/2005*. Bandung: LTRGM-ITB.
- [15] A. P. Bayu. 2005. "Studi dan Implementasi Non Blind Watermarking Dengan Metode Spread Spectrum". Bandung. Institut Teknologi Bandung.
- [16] Sunarya, Unang. 2012. "Simulasi dan Analisi Metode Least Mean Square Untuk Penghapusan Derau Secara Adaptif". Institut Teknologi Telkom Bandung : Tidak dterbitkan.