

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

Dalam bab ini penulis akan menguraikan dan menjelaskan beberapa hasil pengujian dari hasil penelitian tugas akhir ini.

4.1 Kebutuhan Sistem

Sebelum melakukan *denoising* sinyal suara jantung dibutuhkan perangkat lunak yang dapat menunjang penelitian. Perangkat keras dan lunak yang digunakan dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Kebutuhan Perangkat Keras.

Perangkat Keras	Spesifikasi
Processor	AMD E1-1200
Memori	4 Gb
Sistem Operasi	Windows 10

Tabel 4.2 Kebutuhan Perangkat Lunak.

Perangkat Lunak	Uraian
Matlab	Aplikasi yang digunakan untuk mengolah sinyal suara jantung.

4.2 Pengujian Program Penambahan *Noise*

Pengujian program penambahan *noise* bertujuan untuk perbandingan antara sinyal asli dengan sinyal yang tercampur dengan *noise* saat dilakukan proses denoising.

4.2.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah program yang telah dibuat sesuai dengan dasar teori rumus pada halaman 12.

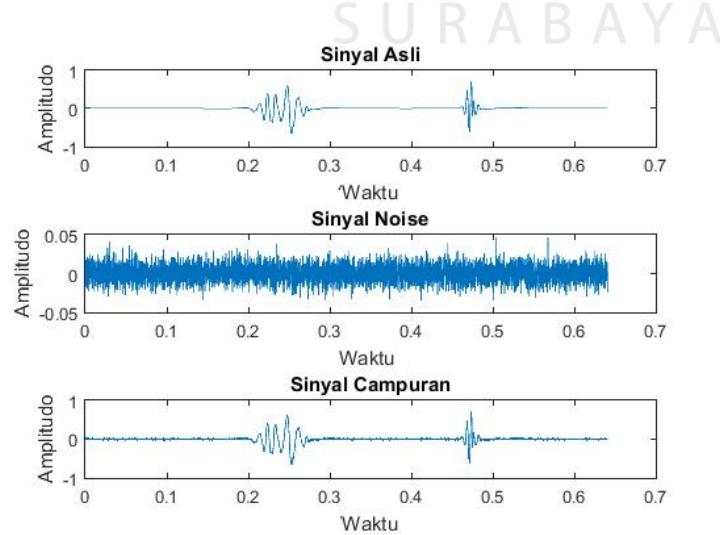
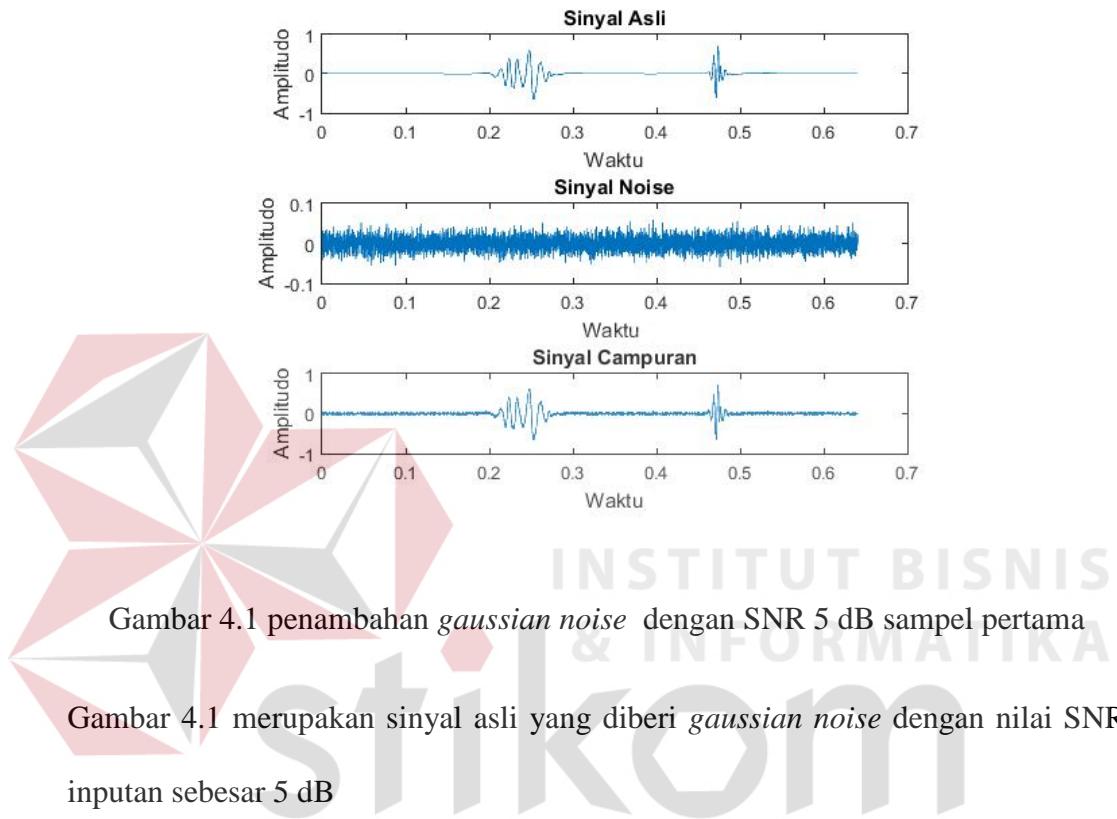
4.2.2 Prosedur Pengujian

Penambahan *noise* diakukan setelah didapatkan sebuah sinyal asli yang bersih dari *noise*. Prosedur pengujian program penambahan *noise* adalah sebagai berikut :

- a. Pengujian ini dilakukan dengan cara memberikan nilai SNR yang berbeda-beda pada *Gaussian Noise* yang telah ditambahkan pada sinyal suara jantung *non noise*. Penambahan SNR terdiri dari beberapa macam yaitu 5 dB, 10 dB, 15 dB, dan 20 dB
- b. Mendapatkan karakteristik dari sinyal setelah dilakukan penambahan *noise* dengan nilai SNR yang berbeda-beda.

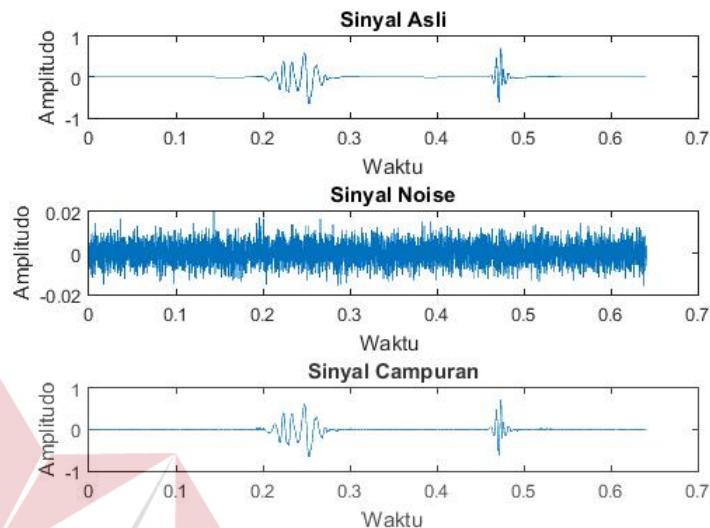
4.2.3 Hasil Pengujian

Berikut adalah gambar dari *noise* dengan nilai SNR yang berbeda-beda



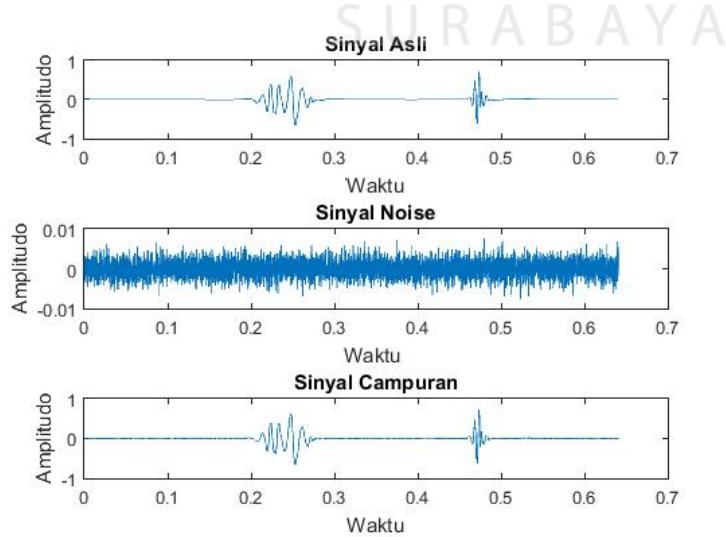
Gambar 4.2 penambahan *gaussian noise* dengan SNR 10 dB sampel pertama

Gambar 4.2 merupakan sinyal asli yang diberi *gaussian noise* dengan nilai SNR inputan sebesar 10 dB



Gambar 4.3 penambahan *gaussian noise* dengan SNR 15 dB sampel pertama

Gambar 4.3 merupakan sinyal asli yang diberi *gaussian noise* dengan nilai SNR inputan sebesar 15 dB



Gambar 4.4 penambahan *gaussian noise* dengan SNR 20 dB sampel pertama

Gambar 4.4 merupakan sinyal asli yang diberi *gaussian noise* dengan nilai SNR inputan sebesar 20 dB

Dari hasil penambahan *gaussian noise* pada sinyal asli dengan nilai SNR inputan yang berbeda-beda, bahwa semakin besar nilai SNR *input* maka nilai amplitudo dari *noise* semakin kecil dan semakin kecil nilai SNR *input* maka nilai amplitudo dari *noise* semakin besar.

4.3 Pengujian Program *Denoising* Menggunakan *Blind Source Separation*

Pengujian program *denoising* menggunakan *blind source separation* bertujuan untuk memisahkan antara *noise* dengan sinyal.

4.3.1 Tujuan

Mengetahui seberapa efektif proses *denoising* sinyal menggunakan *blind source separation*.

4.3.2 Prosedur Pengujian

Proses *denoising* menggunakan *blind source separation* dilakukan setelah didapatkan sebuah sinyal yang telah tercampur oleh *gaussian noise*. Prosedur pengujian program proses *denoising* menggunakan *blind source separation* adalah sebagai berikut :

- a. Tahapan pertama dalam melakukan proses *denoising* menggunakan *blind source separation* adalah dilakukan proses *centering*. *Centering* bertujuan untuk membuat rataan dari sinyal menjadi nol.

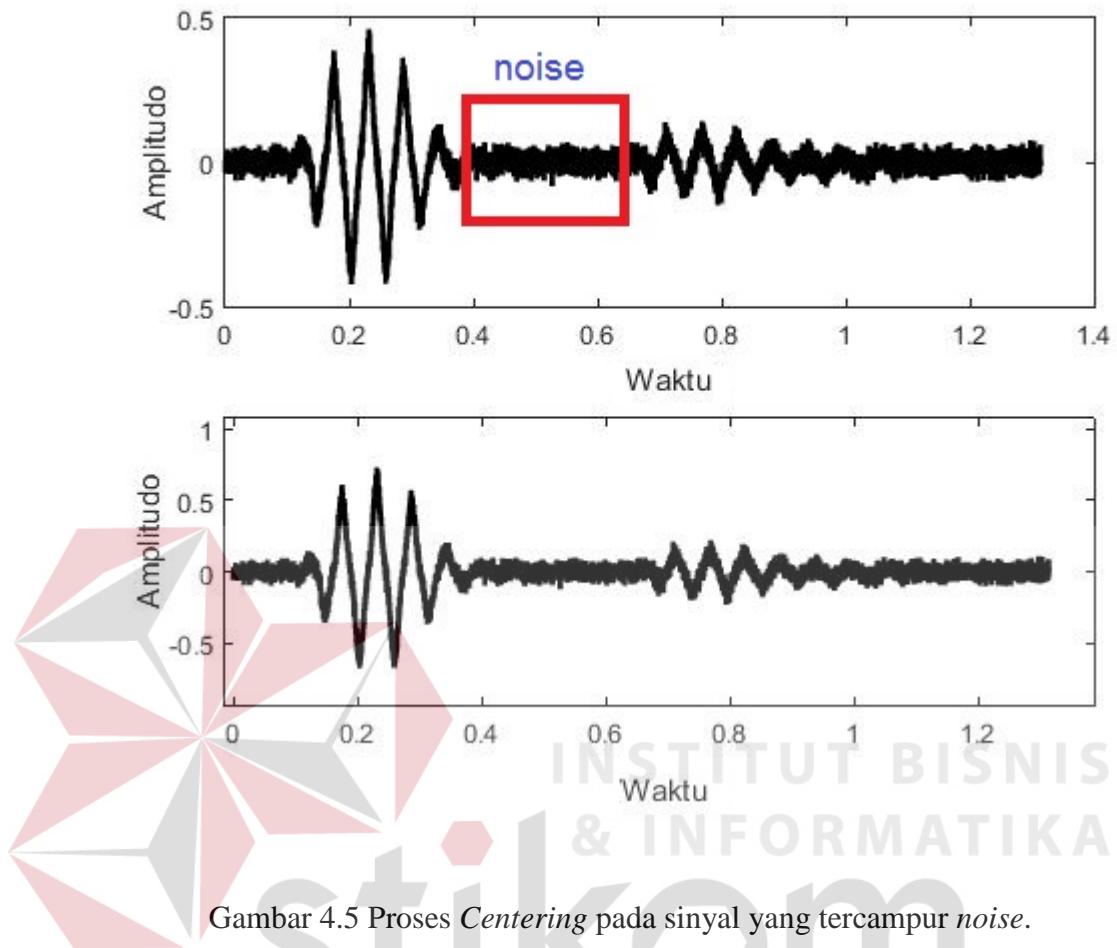
- b. Setelah mendapatkan hasil dari proses *centering*, dilakukan proses *whitening* yang berguna untuk mentransformasi data ke bentuk yang baru.
- c. Proses akhir dari *denoising* dengan menggunakan *blind source separation* adalah ICA. ICA atau *independent component analysis* adalah salah satu metode pada *blind source separation* yang digunakan untuk memisahkan sinyal dari *noise*

4.3.3 Hasil Pengujian

Berikut adalah gambar dari *denoising* menggunakan *Blind Source Separation* dengan tahapan *centering*, *whitening*, dan ICA

a. Proses *centering*.

Setelah didapatkan sebuah sinyal yang telah tercampur *noise* dilakukan proses *centering* agar didapatkan sebuah sinyal yang memiliki nilai rata-rata sama dengan nol. Proses *centering* bertujuan untuk meminimalkan *noise* yang tercampur pada sinyal agar mudah dalam proses menganalisis. Hasil proses *centering* ditunjukkan pada gambar 4.5

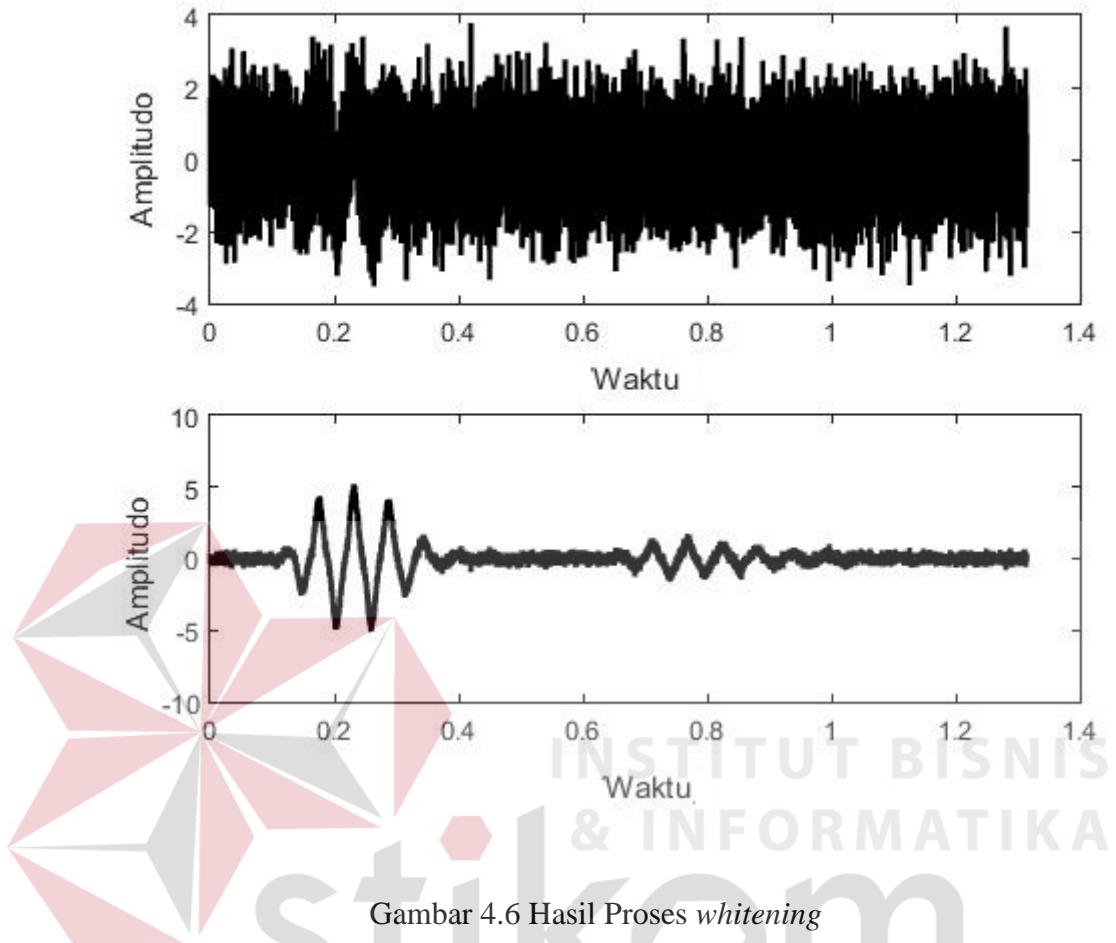


Gambar 4.5 Proses *Centering* pada sinyal yang tercampur *noise*.

Pada gambar 4.5 terdapat dua gambar, gambar yang pertama adalah hasil dari proses *centering* sedangkan gambar yang kedua adalah sebuah sinyal yang tercampur oleh *noise*.

b. Proses *Whitening*

Setelah dilakukan proses *centering*, dilakukan proses *whitening* yang bertujuan untuk menransformasikan sinyal ke bentuk baru dimana didapatkan nilai kovariannya adalah matriks identitas. Hasil dari proses *whitening* ditunjukkan pada gambar 4.6.

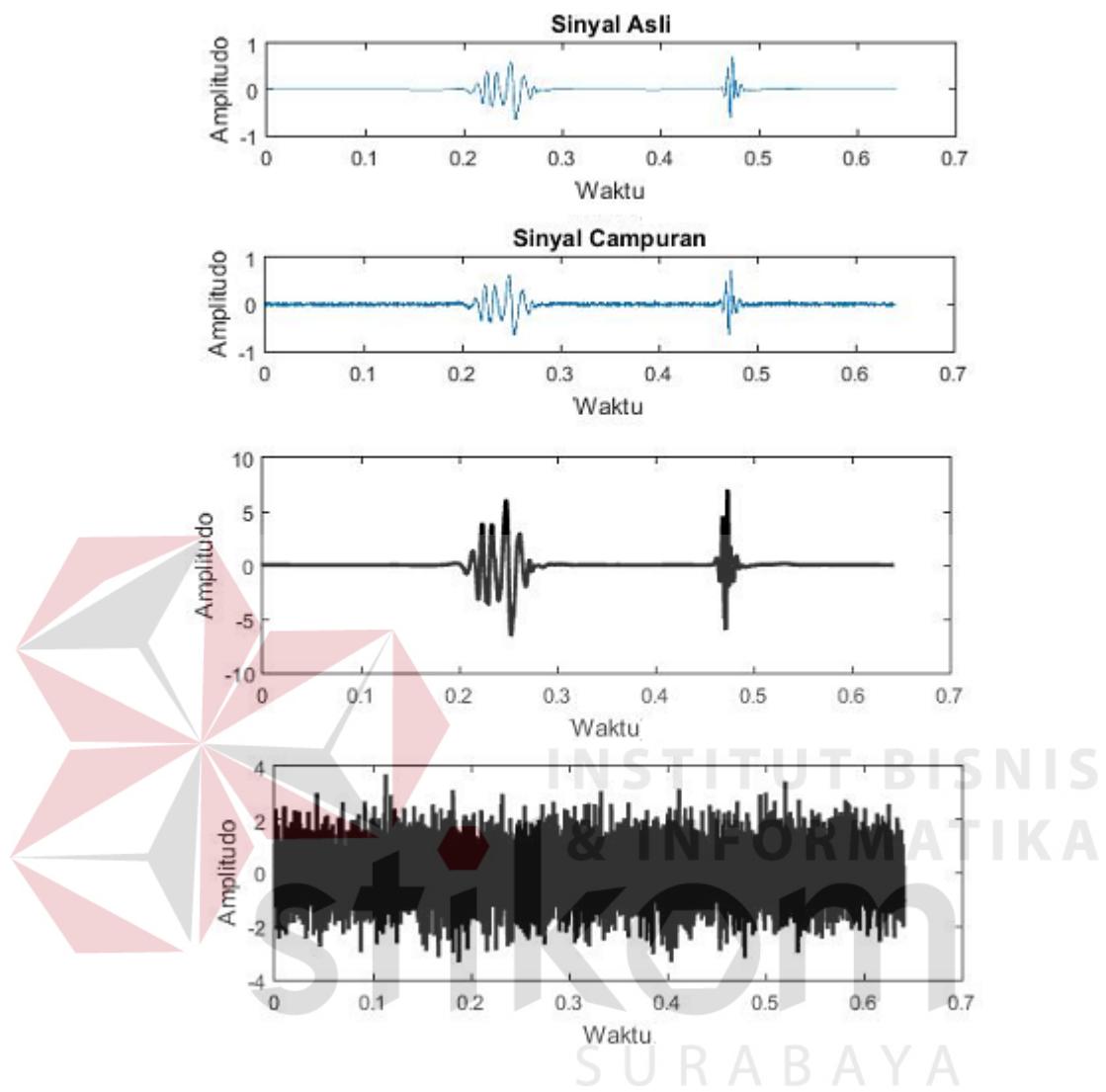


Gambar 4.6 Hasil Proses *whitening*

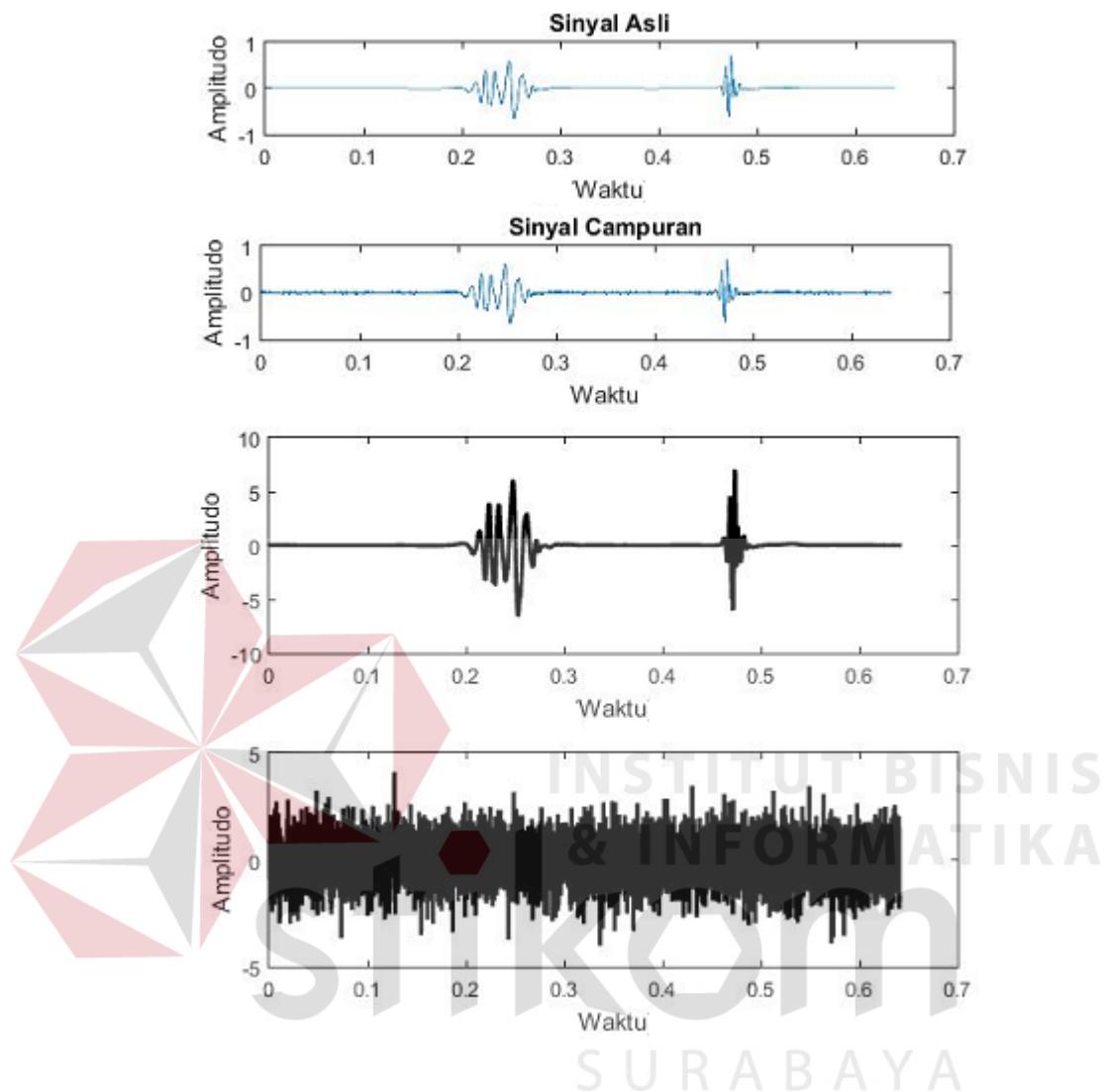
Pada gambar 4.6 terlihat bahwa terdapat dua gambar. Pada gambar yang pertama adalah sebuah *noise*, sedangkan pada gambar yang kedua adalah hasil dari proses *whitening*.

c. Proses ICA

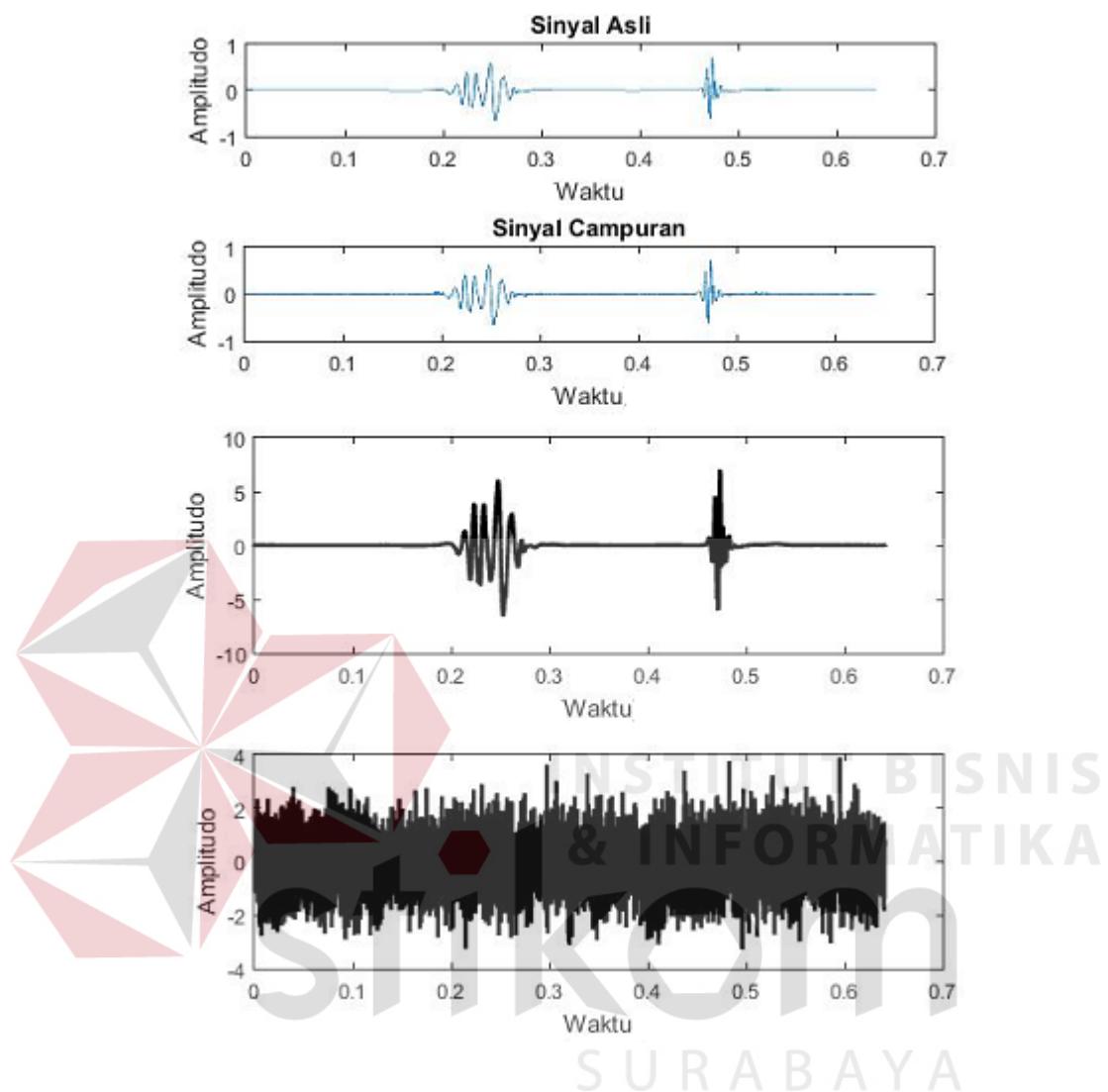
Setelah proses *centering* dan *whitening* tahapan yang terakhir dari proses *denoising* menggunakan *blind source separation* adalah proses *independent component analysis* (ICA). Hasil dari proses ICA ditunjukkan pada gambar di 4.7, 4.8, 4.9, dan 4.10.



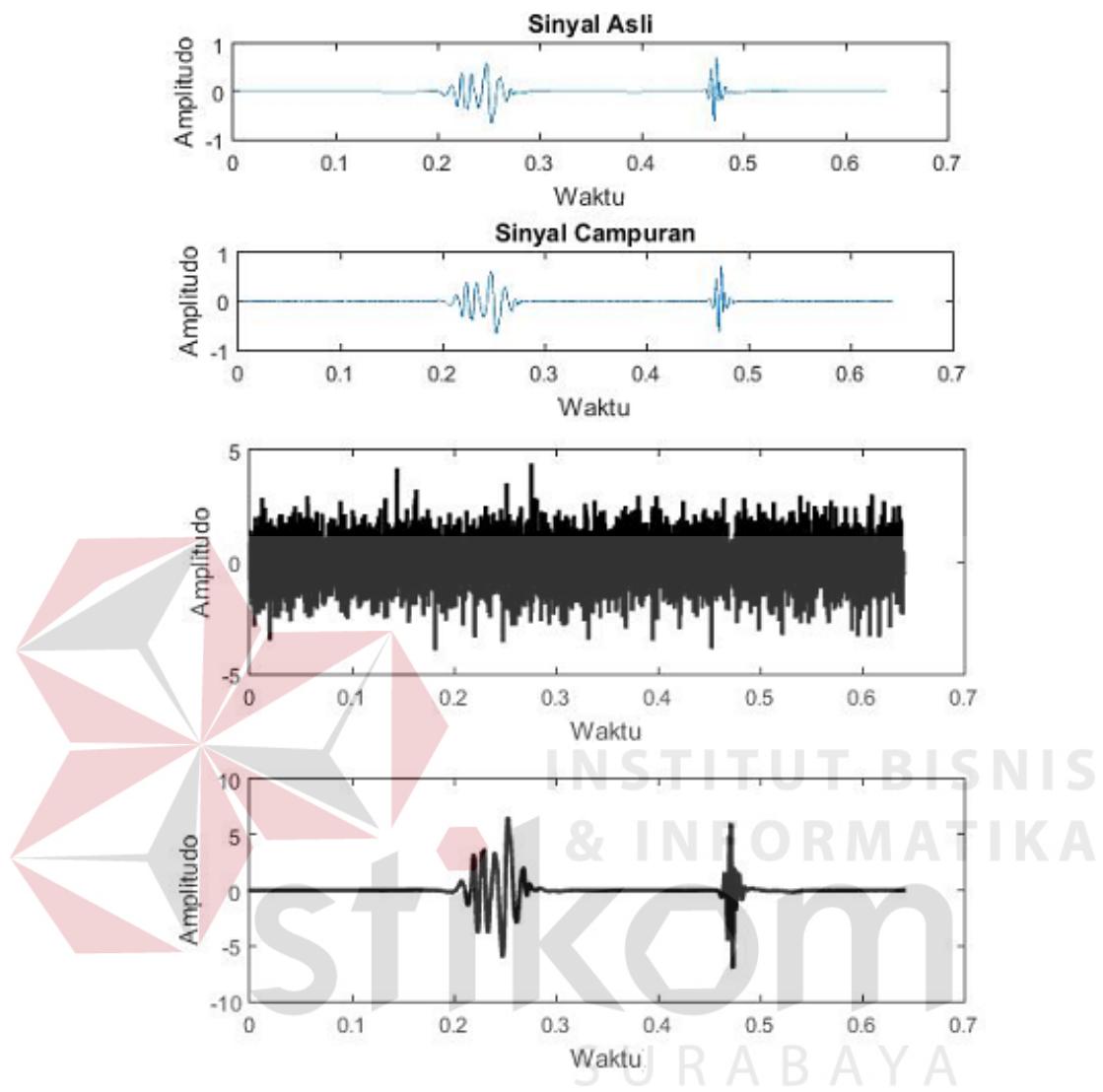
Gambar 4.7 Hasil Proses *denoising* dengan ICA dengan inputan SNR 5 dB



Gambar 4.8 Hasil Proses *denoising* dengan ICA dengan inputan SNR 10 dB



Gambar 4.9 Hasil Proses *denoising* dengan ICA dengan inputan SNR 15 dB



Gambar 4.10 Hasil Proses *denoising* dengan ICA dengan inputan SNR 20 dB

Pada gambar 4.7, 4.8, 4.9 dan 4.10 ditunjukkan hasil proses dari ICA yang terdiri atas sinyal asli, sinyal yang tercampur *noise* dan sinyal hasil *denoising* dengan ICA

4.4 Pengujian Program *Denoising* dengan Transformasi *Wavelet Diskrit*

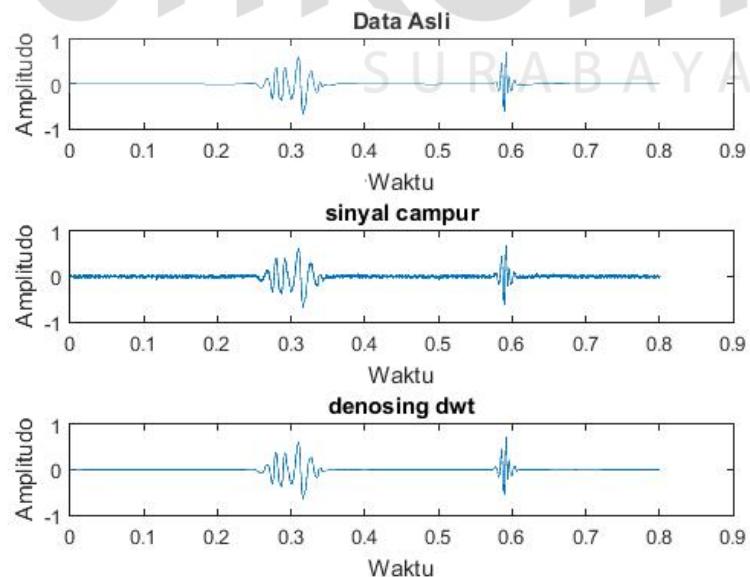
Pengujian *Denoising* dengan mencari nilai *threshold* dengan menerapkan *threshold rules* yaitu *global thresholding* dan *threshold adapt (heursure)*. Setelah mendapatkan nilai *threshold* yang didapatkan berdasarkan karakteristik dari sinyal input, Metode *denoising* yang digunakan adalah *soft thresholding*.

4.4.1 Tujuan

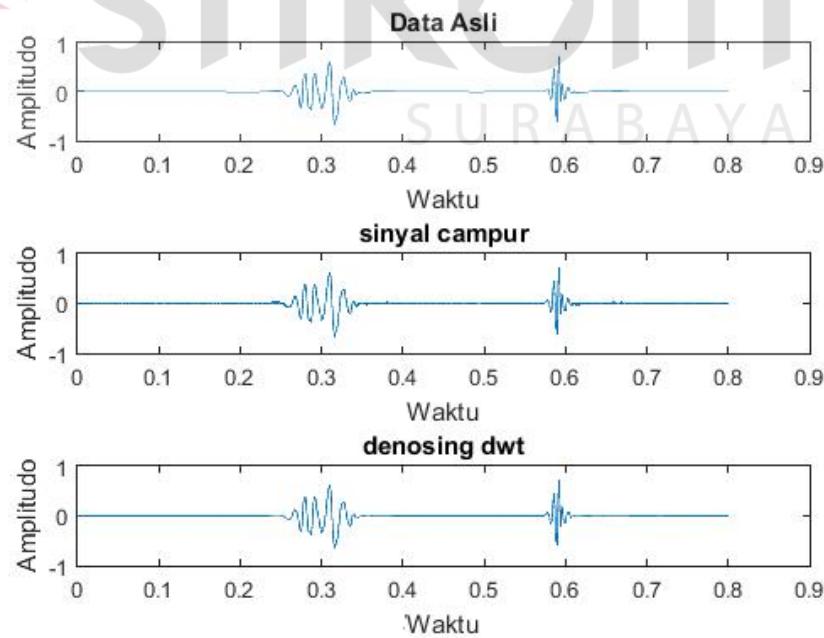
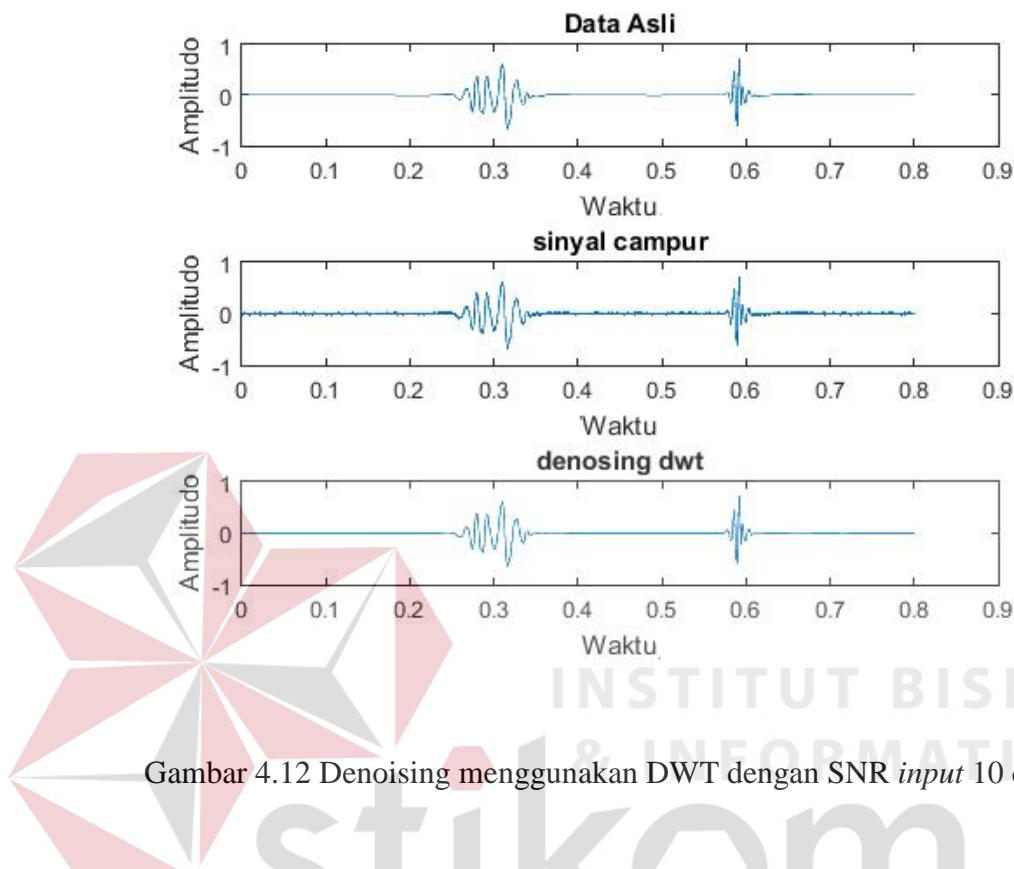
Pengujian ini dilakukan untuk mengetahui kesesuaian antara program yang telah dibuat sesuai dengan dasar teori rumus yang ada.

4.4.2 Hasil Pengujian Program Transformasi *Wavelet Diskrit*

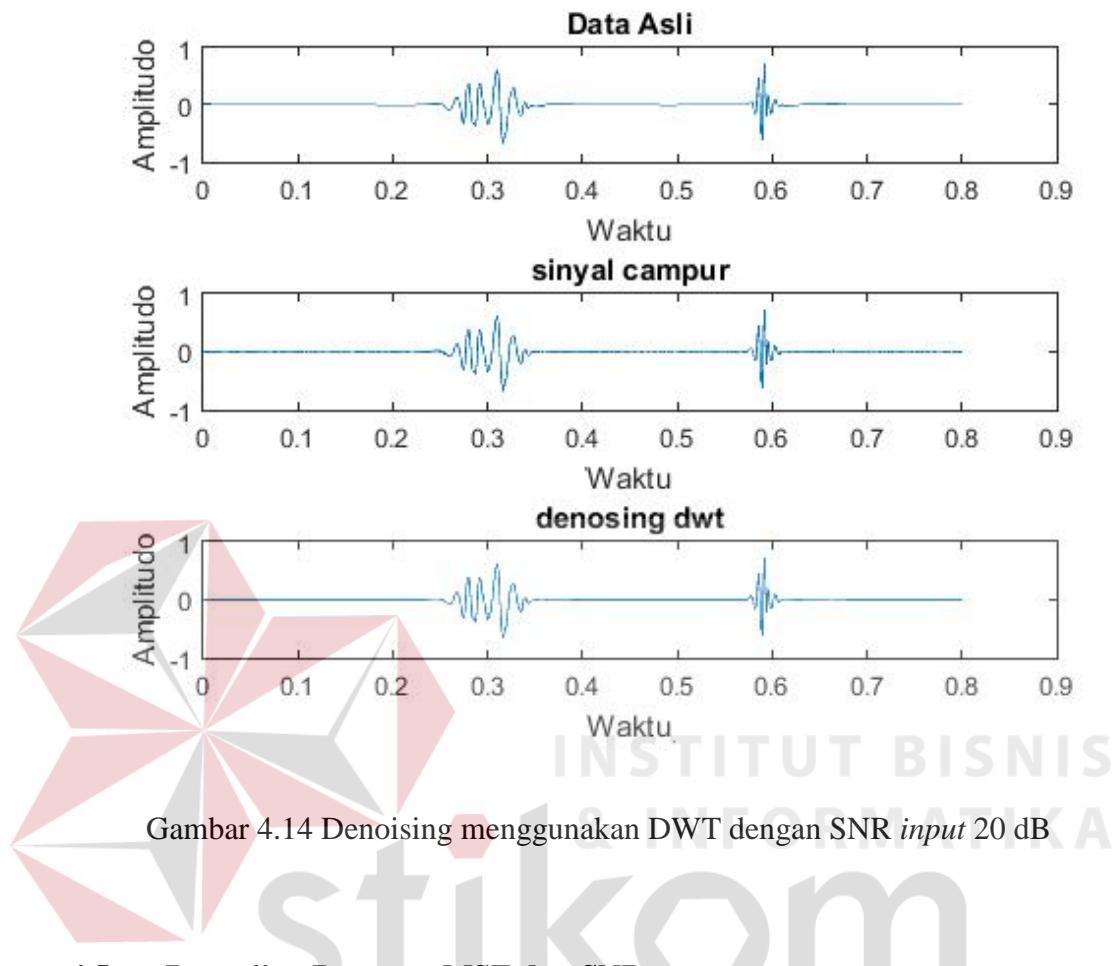
Berikut Hasil proses *denoising* sinyal menggunakan DWT setelah dilakukan penambahan *gaussisan noise* dengan SNR input 5 dB, 10 dB, 15 dB dan 20 dB. Hasil dari proses *denoising* ditunjukkan pada gambar 4.11, 4.12, 4.13, dan 4.14



Gambar 4.11 Denoising menggunakan DWT dengan SNR *input* 5 dB



Gambar 4.13 Denoising menggunakan DWT dengan SNR *input* 15 dB



4.5 Pengujian Program MSE dan SNR

Pengujian program untuk meghitung nilai *Signal to Noise Ratio* (SNR), dan *Mean Square Error* (MSE). Keberhasilan program dapat dilihat dari grafik perbandingan MSE terhadap SNR. Semakin besar nilai SNR dan semakin kecil nilai MSE maka proses denoising semakin baik.

4.5.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah program yang telah dibuat sesuai dengan dasar teori dan rumus yang ada.

4.5.2 Prosedur Pengujian

Prosedur pengujian program untuk meghitung nilai *Signal to Noise Ratio* (SNR), dan *Mean Square Error* (MSE) adalah sebagai berikut:

- Pengujian ini dilakukan dengan cara memberikan nilai SNR yang berbeda-beda pada *Gaussian Noise* yang telah ditambahkan pada sinyal suara jantung *non noise*. Penambahan SNR terdiri dari beberapa macam yaitu 5 dB, 10 dB, 15 dB, dan 20 dB.

- Menghitung nilai *signal to noise ratio* (SNR) output, untuk mengetahui tingkat keberhasilan *denoising*. Dengan rumus sebagai berikut :

$$SNR = 10 \log_{10} \left(\frac{\sigma^2_{noisedsignal}}{\sigma^2_{noisedsignal} - \sigma^2_{denoisedsignal}} \right) \quad (4.1)$$

Dimana :

$\sigma^2_{noisedsignal}$ = varian sinyal yang tercampur *noise*.

$\sigma^2_{denoisedsignal}$ = varian sinyal output yang telah di *denoising*.

- Validitas proses *denoising* di ukur dengan *mean square error* (MSE). Nilai MSE didapatkan dari rumus berikut :

$$MSE = \sum_{i=1}^n (s - s_e)^2 \quad (4.2)$$

Dimana :

MSE = *Mean Square Error*

n = panjang sinyal

S = sinyal input

Se = sinyal output

- d. Menganalisa Keberhasilan program dengan melihat grafik perbandingan MSE terhadap SNR. Semakin besar nilai SNR dan semakin kecil nilai MSE maka proses denoising semakin baik.

4.5.3 Hasil Pengujian

Berikut adalah hasil perhitungan SNR dan MSE untuk masing – masing metode.

a. Metode ICA

1) MSE

Untuk menghitung seberapa besar pergeseran data antara sinyal asli dan sinyal hasil *denoising*, dimana sinyal asli dan sinyal hasil *denoising* memiliki ukuran yang sama maka dilakukan perhitungan MSE. Nilai MSE yang baik adalah mendekati 0 ($MSE \approx 0$). Perhitungannya dilakukan dengan menggunakan simulasi sebanyak 100 kali untuk membangkitkan *noise* yang berbeda.

Tabel 4.3 Hasil perhitungan MSE pada metode ICA

SNR <i>input</i> (dB)	MSE		
	Sampel Pertama	Sampel Kedua	Sampel Ketiga
5 dB	67.53	57.37	179,98
10 dB	66.41	56.64	173,31
15 dB	65.92	56.14	169,76
20 dB	65.76	56.03	169.10

Pada tabel 4.3 perbandingan nilai SNR *input* dengan nilai MSE didapatkan bahwa semakin kecil nilai SNR *input* maka nilai MSE dari sinyal hasil *denoising* semakin besar dan semakin besar nilai SNR *input* maka nilai MSE dari sinyal hasil *denoising* semakin kecil

2) SNR

SNR merupakan metode untuk mengukur kekuatan sinyal terhadap *noise*, yang berarti bahwa semakin besar nilai SNR output dari SNR input maka akan semakin baik hasil denoising tersebut. Pengujian SNR dilakukan dengan SNR input 5 dB, 10 dB, 15 dB dan 20 dB dan dengan menggunakan simulasi sebanyak 100 kali untuk membangkitkan *noise* yang berbeda.

Tabel 4.4 Hasil perhitungan SNR pada metode ICA

SNR <i>input</i> (dB)	SNR <i>output</i> (dB)		
	Sampel Pertama	Sampel Kedua	Sampel Ketiga
5 dB	-0.000943	-0.000809	-0.007006
10 dB	-0.000567	-0.000396	-0.003045
15 dB	-0.000078	-0.000110	-0.024803
20 dB	0.000004	-0.000015	-0.021928

Pada tabel 4.4 perbandingan nilai SNR *input* dengan nilai SNR *output* didapatkan bahwa semakin kecil nilai SNR *input* maka nilai SNR *output* dari sinyal hasil *denoising* semakin kecil dan semakin besar nilai SNR *input* maka nilai SNR *output* dari sinyal hasil *denoising* semakin besar.

b. Metode DWT

1) MSE

Untuk menghitung seberapa besar pergeseran data antara sinyal asli dan sinyal hasil *denoising*, dimana sinyal asli dan sinyal hasil *denoising* memiliki ukuran yang sama maka dilakukan perhitungan MSE. Nilai MSE yang baik adalah mendekati 0 ($MSE \approx 0$). Perhitungannya dilakukan dengan menggunakan simulasi sebanyak 100 kali untuk membangkitkan *noise* yang berbeda.

Tabel 4.5 Hasil perhitungan MSE pada metode DWT

SNR <i>input</i> (dB)	MSE		
	Sampel Pertama	Sampel Kedua	Sampel Ketiga
5 dB	67.60	57.43	179,42
10 dB	66.57	56.46	173,31
15 dB	65.90	56.16	169,50
20 dB	65.75	56.00	169.08

Pada tabel 4.5 perbandingan nilai SNR *input* dengan nilai MSE didapatkan bahwa semakin kecil nilai SNR *input* maka nilai MSE dari sinyal hasil *denoising* semakin besar dan semakin besar nilai SNR *input* maka nilai MSE dari sinyal hasil *denoising* semakin kecil

2) SNR

SNR merupakan metode untuk mengukur kekuatan sinyal terhadap *noise*, yang berarti bahwa semakin besar nilai SNR output dari SNR input maka akan semakin baik hasil denoising tersebut. Pengujian SNR dilakukan dengan SNR input 5 dB, 10 dB, 15 dB dan 20 dB dan dengan

menggunakan simulasi sebanyak 100 kali untuk membangkitkan *noise* yang berbeda.

Tabel 4.6 Hasil perhitungan SNR pada metode DWT

SNR <i>input</i> (dB)	SNR <i>output</i> (dB)		
	Sampel Pertama	Sampel Kedua	Sampel Ketiga
5 dB	-0.108920	-0.092715	-0.275240
10 dB	-0.046611	-0.037449	-0.110480
15 dB	-0.010209	-0.008715	-0.023937
20 dB	-0.002031	-0.001569	-0.004267

Pada tabel 4.6 perbandingan nilai SNR *input* dengan nilai SNR *output* didapatkan bahwa semakin kecil nilai SNR *input* maka nilai SNR *output* dari sinyal hasil *denoising* semakin kecil dan semakin besar nilai SNR *input* maka nilai SNR *output* dari sinyal hasil *denoising* semakin besar.

4.6 Hasil perbandingan nilai SNR *input* dan *output* hasil metode ICA dan

DWT

Tabel 4.7 Hasil Perbandingan nilai SNR *input* 5 dB dan SNR *output* hasil metode

ICA dan DWT

Sampel	SNR <i>Input</i>	SNR <i>output</i> ICA (dB)	SNR <i>output</i> DWT (dB)
Sampel 1	5	-0.000943	-0.108920
Sampel 2	5	-0.000809	-0.092715
Sampel 3	5	-0.007006	-0.275240
Rata-Rata		-0.002920	-0.158958

Tabel 4.8 Hasil Perbandingan nilai SNR *input* 10 dB dan SNR *output* hasil metode ICA dan DWT

Sampel	SNR <i>Input</i>	SNR <i>output</i> ICA (dB)	SNR <i>output</i> DWT (dB)
Sampel 1	10	-0.000567	-0.046611
Sampel 2	10	-0.000396	-0.037449
Sampel 3	10	-0.003045	-0.110480
Rata-Rata		-0.001336	-0.064847

Tabel 4.9 Hasil Perbandingan nilai SNR *input* 15 dB dan SNR *output* hasil metode ICA dan DWT

Sampel	SNR <i>Input</i>	SNR <i>output</i> ICA (dB)	SNR <i>output</i> DWT (dB)
Sampel 1	15	-0.000078	-0.010209
Sampel 2	15	-0.000110	-0.008715
Sampel 3	15	-0.024803	-0.023937
Rata-Rata		-0.008330	-0.014287

Tabel 4.10 Hasil Perbandingan nilai SNR *input* 20 dB dan *output* hasil metode ICA dan DWT

Sampel	SNR <i>Input</i>	SNR <i>output</i> ICA (dB)	SNR <i>output</i> DWT (dB)
Sampel 1	20	0.000004	-0.002031
Sampel 2	20	-0.000015	-0.001569
Sampel 3	20	-0.021928	-0.004267
Rata-Rata		-0.007313	-0.002622

Berdasarkan hasil pada tabel 4.7, tabel 4.8, tabel 4.9 ICA menghasilkan nilai SNR yang lebih besar maka hasil *denoising* yang dihasilkan semakin baik, sedangkan pada tabel 4.10 ketika SNR input yang diberikan mencapai 20 dB metode DWT menghasilkan nilai SNR *output* yang besar. Sebagai contoh pada tabel 4.7, 4.8, 4.9 metode ICA menghasilkan hasil *denoising* yang lebih baik dan pada tabel 4.10 metode DWT menghasilkan hasil *denoising* yang lebih baik.

4.7 Hasil Perbandingan *Mean Square Error* (MSE) untuk metode DWT

dan ICA

Tabel 4.11 Hasil Perbandingan *Mean Square Error* (MSE) untuk metode DWT dan ICA menggunakan SNR 5 dB

Sampel	SNR	MSE ICA	MSE DWT
Sampel 1	5	67.53	67.60
Sampel 2	5	57.37	57.43
Sampel 3	5	179.98	179.42
Rata-Rata		101.63	101.48
Selisih		0,2	

Tabel 4.12 Hasil Perbandingan *Mean Square Error* (MSE) untuk metode DWT

dan ICA menggunakan SNR 10 dB

Sampel	SNR	MSE ICA	MSE DWT
Sampel 1	10	66.41	66.57
Sampel 2	10	56.64	56.46
Sampel 3	10	173.31	173.31
Rata-Rata		98.78	98.78
Selisih		0	

Tabel 4.13 Hasil Perbandingan *Mean Square Error* (MSE) untuk metode DWT dan ICA menggunakan SNR 15 dB

Sampel	SNR	MSE ICA	MSE DWT
Sampel 1	15	65.92	65.90
Sampel 2	15	56.14	56.16
Sampel 3	15	169.76	169.50
Rata-Rata		97.27	97.19
Selisih			0,08

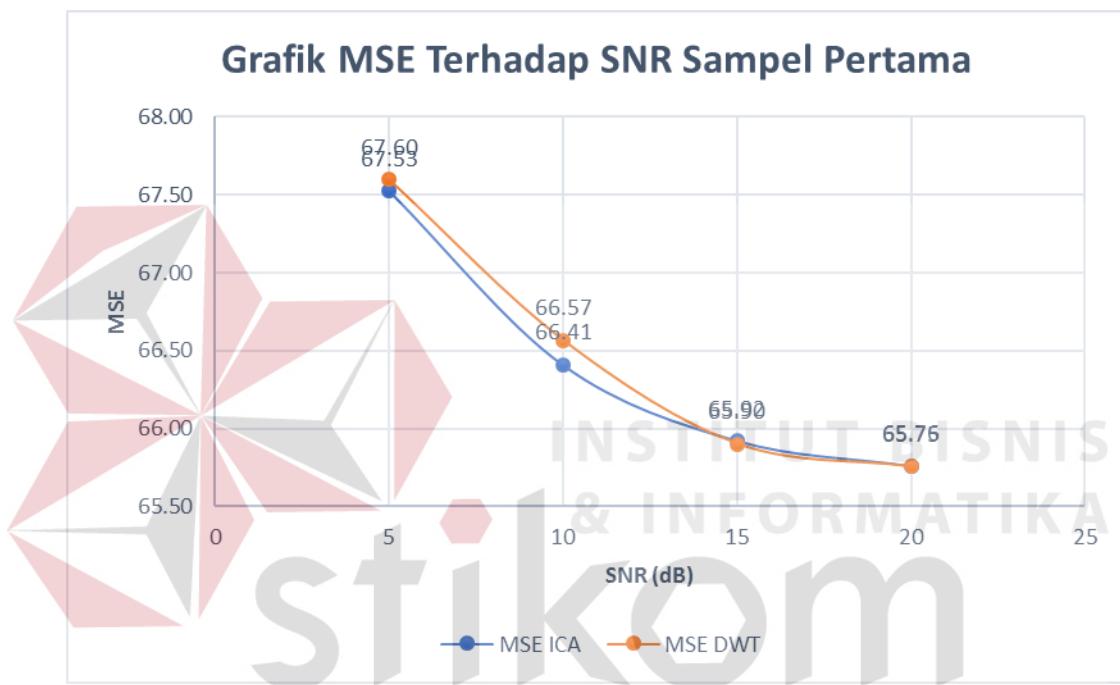
Tabel 4.14 Hasil Perbandingan *Mean Square Error* (MSE) untuk metode DWT dan ICA menggunakan SNR 20 dB

Sampel	SNR	MSE ICA	MSE DWT
Sampel 1	20	65.76	65.75
Sampel 2	20	56.03	56.00
Sampel 3	20	169.10	169.08
Rata-Rata		96.97	96.95
Selisih			0,02

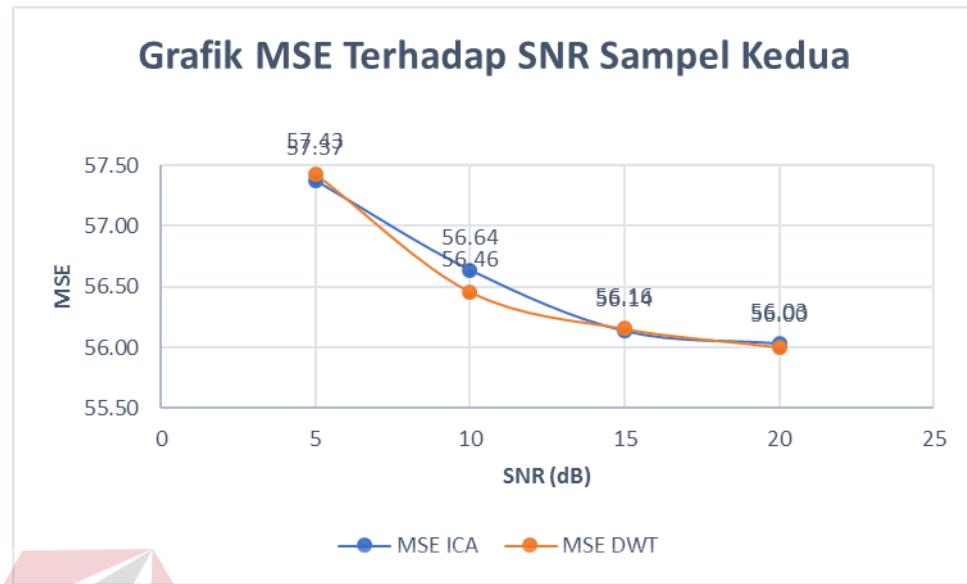
Berdasarkan hasil pengukuran nilai *mean square error* (MSE) pada tabel 4.11, 4.12, 4.13 dan 4.14 untuk nilai SNR yang berbeda-beda dapat disimpulkan bahwa hasil dari MSE untuk kedua metode (ICA dan DWT) tidak berbeda signifikan antara satu sama lain, yang artinya bahwa kedua metode tersebut dapat berfungsi dengan baik pada proses *denoising* sinyal jantung PCG.

4.8 Hasil perbandingan nilai MSE terhadap SNR

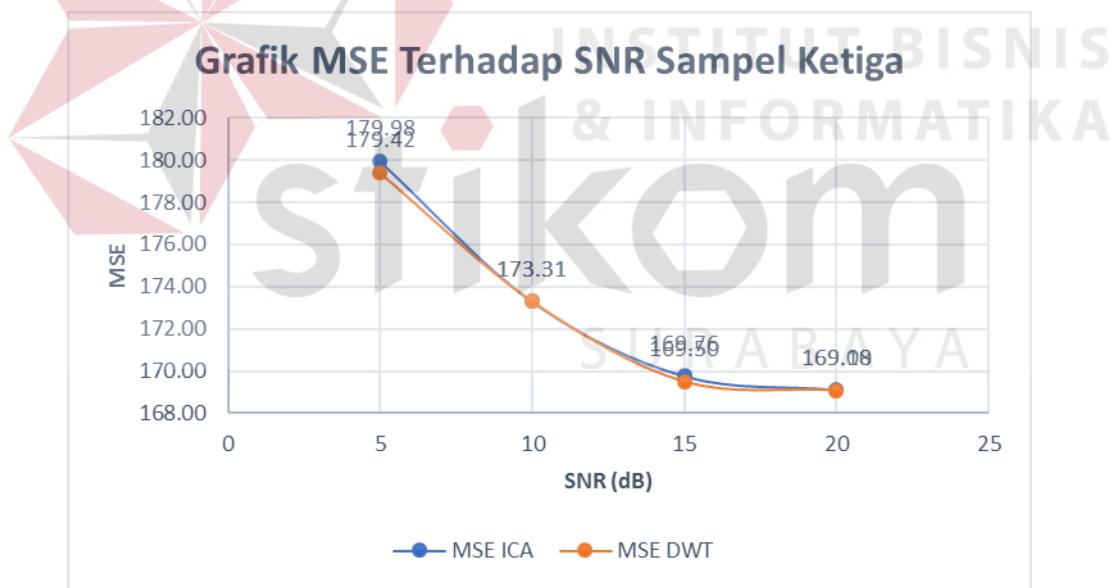
Hasil dari program SNR dan MSE akan di plot pada grafik dapat dilihat pada gambar 4.15, 4.16 dan 4.17. Guna melihat proses denoising telah berjalan sesuai rumus dan teori yang ada.



Gambar 4.15 Grafik perbandingan MSE terhadap SNR pada sampel pertama



Gambar 4.16 Grafik perbandingan MSE terhadap SNR pada sampel kedua



Gambar 4.17 Grafik perbandingan MSE terhadap SNR pada sampel ketiga

Berdasarkan grafik pada gambar 4.12, 4.13, dan 4.14 menunjukkan perbandingan antara MSE dan SNR. Dari grafik di atas, tampak bahwa semakin meningkatnya nilai SNR maka nilai MSE akan semakin menurun. Artinya, semakin kecil noise,

nilai Nilai MSE akan semakin kecil pula. Hal ini dikarenakan noise berpengaruh pada penurunan nilai amplitudo sinyal. Serta grafik di atas menunjukkan proses program *denoising* berhasil, dan pada SNR 20 dB nilai MSE antara ICA dan DWT terlihat mendekati pada angka yang sama.

