

BAB 3

METODE PENELITIAN

1.1 ALAT DAN BAHAN

1.1.1 Dataset

Dataset ini merupakan data yang berisi rekaman suara Otak yang akan digunakan sebagai bahan dalam pengambilan data sinyal suara Otak. Sinyal suara Otak yang didapatkan dari *dataset* inilah yang digunakan dalam proses analisis sinyal suara Otak sebelum dan sesudah mengalami proses *denoising*. Data sinyal suara Otak yang akan digunakan dalam proses analisis merupakan data yang berisi 1 data yang terdiri dari 22 rekaman sinyal otak yang akan digunakan sebagai bahan dalam pengambilan data yang berisi rekaman sinyal EEG. Dataset didapatkan dari alamat berikut https://figshare.com/articles/Experiment_HaLT-SubjectM-161124-6States/5554642. .

1.1.2 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam proses penelitian ini yaitu sebuah PC (*Personal Computer*) dengan spesifikasi sebagai berikut :

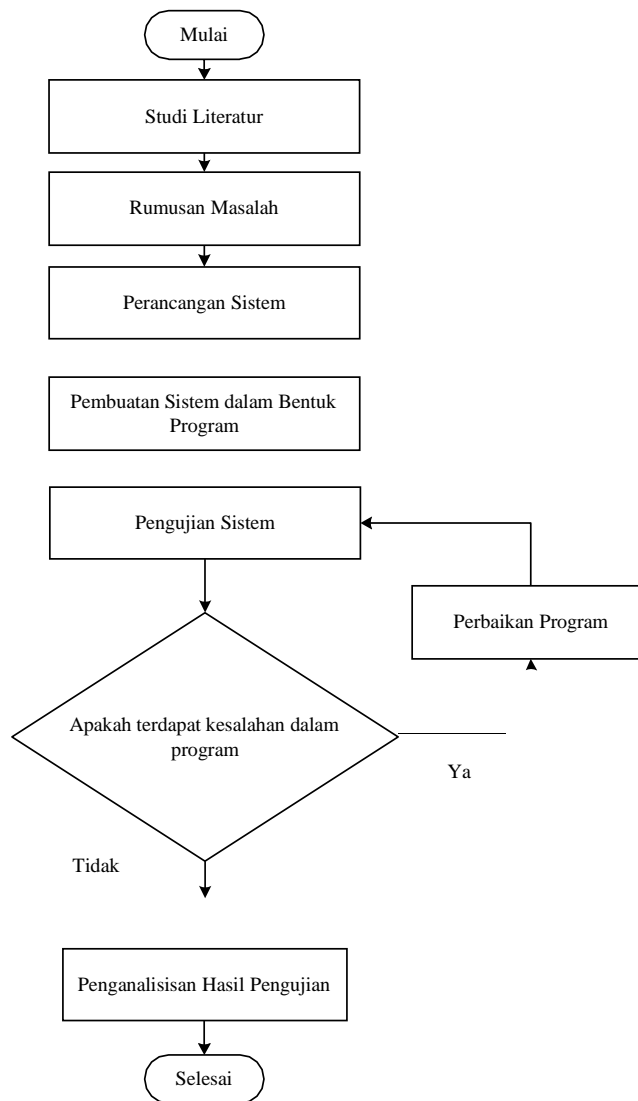
1. Intel ® Core™ i3-2330M
2. *Windows 7 (64-Bit)*.
3. RAM 2 GB.
4. Memory Harddisk 500 GB.

1.1.3 Perangkat Lunak (*Software*)

Software yang akan digunakan dalam pembuatan sistem *denoising* ini adalah Matlab R2016a. *Software* Matlab ini banyak digunakan dalam penelitian di bidang pengembangan sistem, desain sistem, dan lain-lain. Matlab merupakan *Software* yang akan digunakan untuk proses pemisahan antara sinyal EEG yang masih mengandung *noise* AWGN menjadi sinyal EEG yang tidak mengandung *noise*. Pemisahan *noise* AWGN dari sinyal EEG menggunakan metode TWD dan FHD.

1.2 ALUR PENELITIAN

Alur penelitian merupakan salah satu komponen penting untuk memulai sebuah penelitian hingga berakhirnya penelitian. Pada alur penelitian ini berisikan langkah-langkah yang menjelaskan setiap kegiatan penelitian yang akan dilakukan. Berikut ini merupakan diagram alur penelitian yang disajikan dalam bentuk *flowchart*, yaitu :



Gambar 3.1 Alur Penelitian

Pada Gambar 3.1 terlihat bahwasannya penelitian yang akan dilakukan dalam tugas akhir ini dimulai dari perancangan sistem. Perancangan sistem merupakan suatu gambaran sistem yang akan digunakan untuk melakukan

pengolahan data. Sistem yang akan dibuat merupakan sistem dalam bentuk kode program yang ditulis pada sebuah *software* Matlab R2016a. Rancangan sistem yang telah selesai dibuat dalam bentuk program akan dilakukan pengujian terhadap sistem tersebut. Tujuan dari pengujian ini yaitu untuk mengetahui kemampuan sistem sesungguhnya dari sistem yang telah dibuat. Apabila terjadi kesalahan pada saat pengujian sistem maka langkah yang akan dilakukan yaitu melakukan perbaikan terhadap sistem dengan melakukan pengecekan kode program yang telah dibuat untuk memastikan bahwa kode program yang ditulis sudah sesuai dengan aturan yang semestinya. Apabila sistem tidak ada kesalahan saat sedang dilakukan pengujian maka langkah selanjutnya yaitu melakukan analisis terhadap data yang telah diolah oleh sistem.

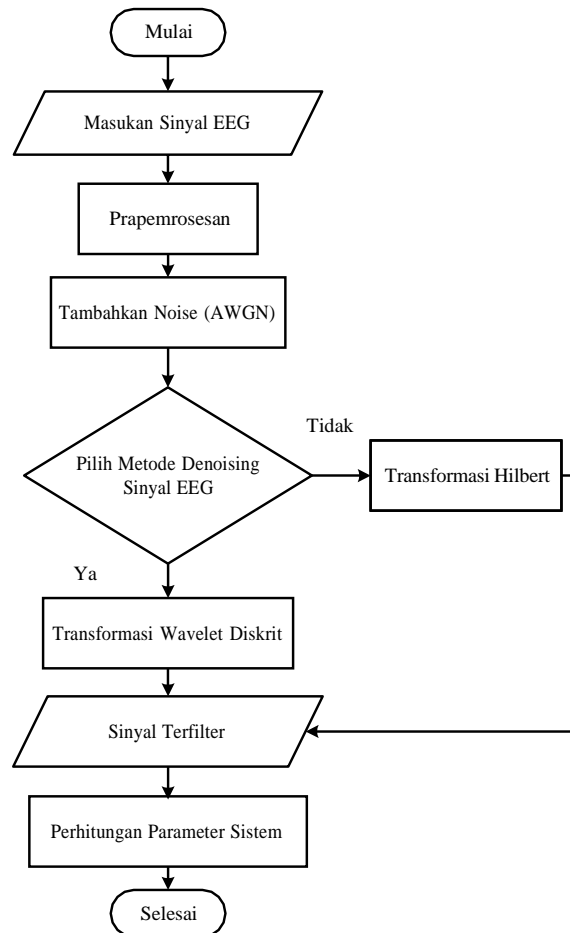
1.3 RANCANGAN SISTEM

Pada penelitian ini akan digambarkan mengenai rancangan sistem yang terdiri dari proses prapemrosesan sampai dengan proses *denoising noise* AWGN pada Sinyal EEG dengan menggunakan metode TWD dan FHD. Rancangan sistem ini merupakan tahap penggambaran dari sistem yang akan dibuat dalam bentuk kode program pada *software* Matlab R2016a. Rancangan sistem ini terdiri dari bentuk *flowchart* dan kode program. Pada bentuk *flowchart* akan memberikan gambaran rancangan sistem secara umum dan secara spesifik. Rancangan sistem secara spesifik menjelaskan teknik *denoising noise* AWGN dengan menggunakan metode transformasi TWD dan FHD. Penulisan kode program ini merupakan bentuk implementasi dari sebuah rumus matematika yang kemudian diterjemahkan menjadi kode program dengan menggunakan *software* Matlab R2016a. Penulisan kode program ini tidak hanya menggunakan rumus matematika, tetapi menggunakan fungsi *toolbox* yang mana telah disediakan oleh *software* Matlab R2016a. Berikut ini merupakan bentuk rancangan sistem yang akan dibuat, yaitu :

1. Rancangan Sistem Secara Umum

Rancangan sistem ini memberikan gambaran secara menyeluruh mengenai tahap-tahap *denoising noise* AWGN dengan menggunakan metode TWD dan

FHD. Berikut ini merupakan rancangan sistem secara umum yang mana dibuat dalam bentuk flowchart, yaitu :

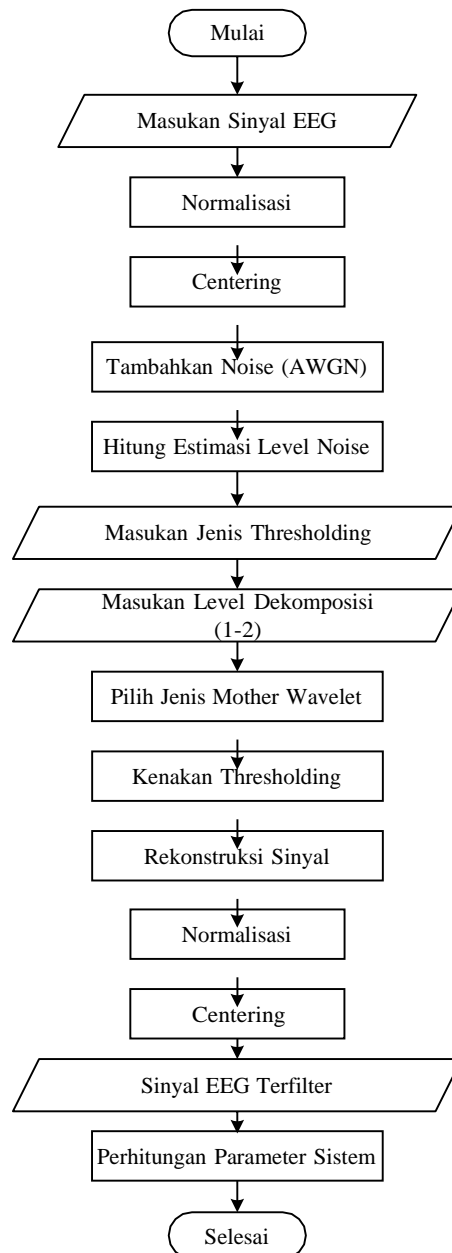


Gambar 3.2 Rancangan Sistem

Pada Gambar 3.2 merupakan rancangan sistem bagian pertama, yang mana rancangan sistem tersebut dimulai dengan memasukkan data. Data yang akan dimasukkan merupakan data rakaman Sinyal EEG. Sinyal EEG yang belum ditambahkan *noise* AWGN ini akan dilakukan tahap prapemrosesan terlebih dahulu yang terdiri dari normalisasi dan *centering*. Apabila tahap prapemrosesan telah selesai maka tahap selanjutnya yaitu penambahan AWGN. Apabila telah ditambahkan AWGN maka tahap selanjutnya yaitu menentukan metode mana yang akan digunakan untuk melakukan penghilangan *noise* AWGN yang ada pada Sinyal EEG. Jika pemilihan metode telah dilakukan maka proses selanjutnya adalah proses penghilangan *noise* AWGN dengan menggunakan metode yang telah dipilih. Tahap selanjutnya yaitu proses penghitungan parameter sistem sebelum dan sesudah proses penghilangan *noise*.

2. Rancangan Sistem Wavelet Denoising

Berikut ini merupakan rancangan sistem yang mana dibuat dalam bentuk *flowchart* dengan menggunakan metode TWD, yaitu :



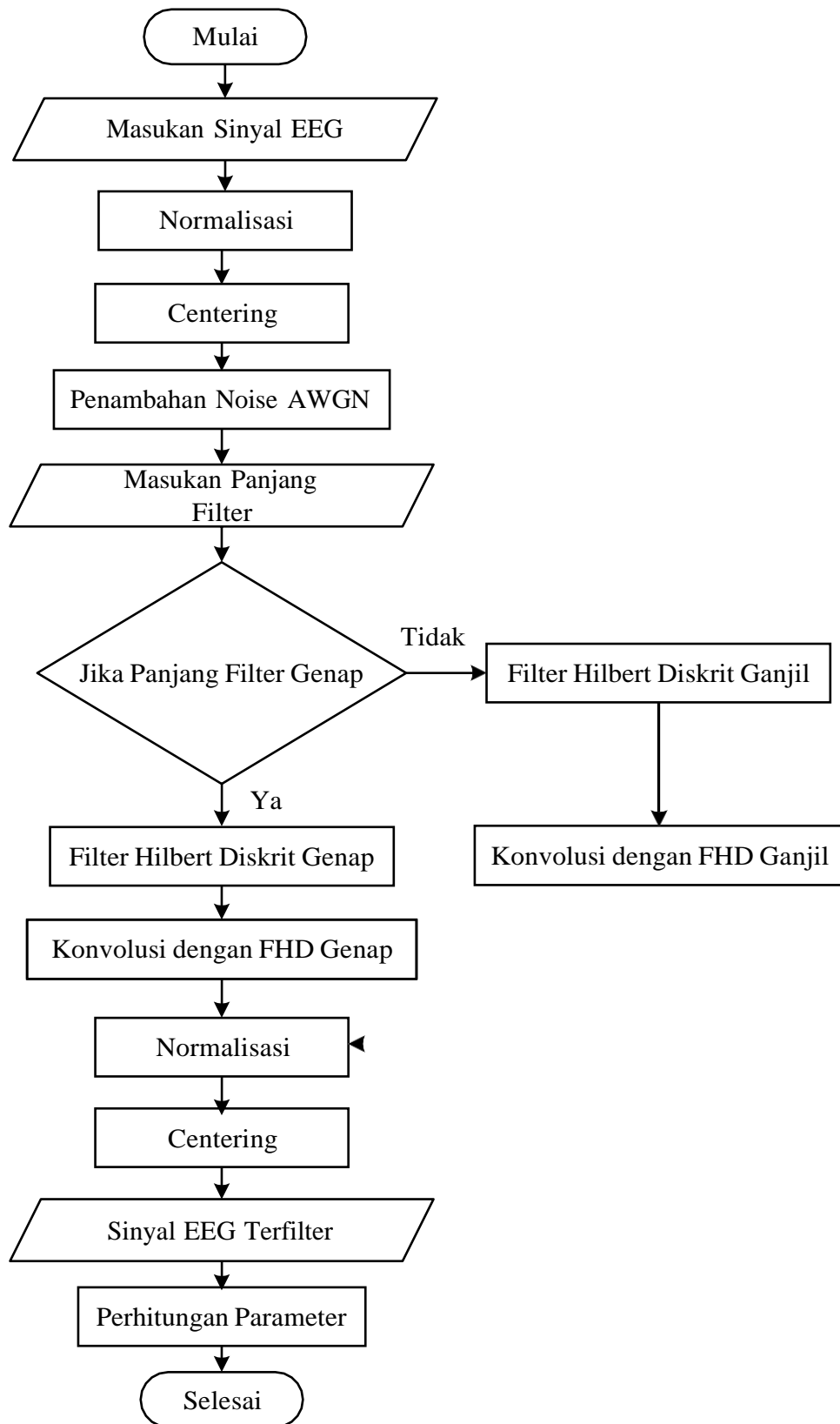
Gambar 3.3 Rancangan sistem TWD

Pada Gambar 3.3 merupakan tahap penghilangan *noise* AWGN pada Sinyal EEG dengan menggunakan metode TWD. Pada metode ini Sinyal EEG harus dinormalisasi dan *centering* terlebih dahulu sebelum ditambahkan *noise* AWGN. Sinyal EEG yang telah ditambahkan *noise* AWGN akan dihitung level

estimasi *noise*-nya dan diikuti dengan penentuan jenis *thresholding*nya yaitu *hard*, *soft*, atau *adaptive*. Masing-masing *thresholding* ini memiliki perhitungan masing-masing sehingga akan menghasilkan nilai *threshold* yang berbeda-beda untuk proses penghilangan *noise* AWGN. Apabila pada tahap ini telah selesai dilakukan, maka tahap selanjutnya adalah pemilihan jenis mother wavelet yang akan digunakan untuk proses penghilangan *noise* AWGN. Setelah dikenakan koefisien wavelet yang didapat dengan menggunakan *threshold* yang telah didapat sebelumnya untuk proses pemfilteran. Proses selanjutnya yaitu proses rekonstruksi Sinyal EEG yang telah difilter. Sinyal EEG yang telah difilter akan dikenakan proses normalisasi dan *centering*. Tahap terakhir yaitu perhitungan parameter sistem.

3. Rancangan Sistem Diskrit Hilbert Filter

Pada Gambar 3.5 merupakan *flowchart* untuk proses penghilangan *noise* AWGN yang ada pada Sinyal EEG. Pada Gambar 3.5 Sinyal EEG akan dinormalisasi dan di-*centering* terlebih dahulu sebelum ditambahkan *noise* AWGN. Langkah selanjutnya yaitu menentukan panjang filter yang akan digunakan dari FHD yang terdiri dari panjang filter genap dan ganjil. Apabila panjang filter telah selesai dipilih, maka proses selanjutnya yaitu melakukan proses *filtering*. Sinyal yang telah difilter akan dilakukan proses normalisasi dan *centering* seperti pada tahap sebelumnya. Setelah itu akan dilakukan perhitungan parameter sistem dan proses penghilangan *noise* AWGN pada sinyal EEG telah selesai dilakukan. Berikut ini merupakan bentuk rancangan sistem yang mana dibuat dalam bentuk *flowchart* dengan menggunakan metode FHD, baik yang tidak menggunakan BPF ataupun yang menggunakan BPF yaitu :



Gambar 3.4 Rancangan sistem FHD

1.3.1 Masukan Sinyal Electroencephalogram

Berikut merupakan tahap memasukan Sinyal EEG ke dalam sistem yang telah dibuat dengan menggunakan *software* Matlab R2016a, yaitu :

1.3.2 Prapemrosesan

Pada tahap prapemrosesan ini terdiri dari dua tahap yaitu tahap normalisasi dan *centering*. Tahap prapemrosesan merupakan salah satu tahap yang penting sebelum melakukan pengolahan Sinyal EEG.

1.3.2.1 Normalisasi

Tujuan dari normalisasi ini yaitu agar setiap elemen dalam vektor suara bisa diskalakan ke dalam kisaran antara -1.0 dan $+1.0$ [22]. Sehingga amplitudo pada sebelum dan sesudah proses penghapusan *noise* tidak ada yang berbeda. Berikut ini merupakan persamaan matematis yang digunakan untuk melakukan proses normalisasi pada tahap prapemrosesan, yaitu :

$$s_{norm}(t) = \frac{s(t)}{\max(|s(t)|)} \quad (3.1)$$

Dimana,

$s(t)$ = Sinyal EEG

$s_{norm}(t)$ = Sinyal Hasil Normalisasi

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
% normalisasi data mentah agar berada pada -1 hingga +1 volt  
data = data ./ max(abs(data));
```

1.3.2.2 Centering

Centering mengubah semua konsentrasi menjadi fluktuasi disekitar nol sebagai gantinya yaitu disekitar rata-rata data. Dengan ini, ia menyesuaikan perbedaan di antara data tinggi dan rendah yang berlimpah. Oleh karena itu digunakan untuk fokus pada bagian data yang berfluktuasi, dan hanya menyisakan variasi yang relevan. Tujuan dari *centering* ini yaitu untuk menghapus *offset* yang

ada pada suatu data [23]. Berikut ini merupakan persamaan matematis untuk melakukan *centering*, yaitu :

$$\tilde{x}_j = x_{ij} - \bar{x} \quad (3.2)$$

Dimana,

\tilde{x}_j = Sinyal Hasil *Centering*

x_{ij} = Sinyal EEG Asli

\bar{x} = Rata-rata Sinyal EEG Asli

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
% centering
data = data - mean(data(:));
```

1.3.3 Penambahan AWGN

AWGN merupakan salah satu metode untuk menambahkan *noise* pada sebuah sinyal secara distribusi dan merata ke dalam semua frekuensi. Sinyal EEG sebelum masuk tahap *denoising* dengan menggunakan TWD dan FHD akan ditambahkan AWGN terlebih dahulu. Penambahan AWGN pada Sinyal EEG ini bertujuan untuk menurunkan kualitas Sinyal EEG dan digunakan untuk mengukur seberapa baik sistem yang dibuat dapat menghilangkan AWGN yang telah bercampur dengan Sinyal EEG. Berikut ini merupakan penerapan rumus dari persamaan 2.5 menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
% berikan noise acak N(0,1)
% datan = wgn(length(data), 1, 0);
datan = awgn(data, 10, 'Measured');
datax = data + datan;
```

1.3.4 Denoising Transformasi Wavelet Diskrit

Ada beberapa tahapan yang perlu dilakukan dalam melakukan *denoising* dengan menggunakan metode TWD, yaitu :

3.3.5.1 Estimasi Level Noise

Metode *thresholding* bisa dikelompokkan menjadi dua kategori yaitu global *thresholding* dan bergantung level *thresholding*. Hal ini berarti bahwa pemilihan sebuah nilai tunggal untuk *thresholding* λ untuk diterapkan secara global untuk semua empirical koefisien wavelet, yang mana dapat diartikan bahwamemungkinkan perbedaan nilai λ_j yang dipilih untuk setiap level resolusi j . Sebuah *thresholding* membutuhkan membutuhkan sebuah estimasi dari sebuah *noise* level σ . Biasanya standar deviasi dari nilai data secara jelas tidak bagus untuk sebuah estimasi, kecuali kalau yang mendasari fungsi S-nya adalah masuk akal. Berdasarkan Donoho dan Johnstone (1994) estimasi σ dalam domain wavelet dan disarankan bahwa sebuah estimasi yang kuat yang mana berdasarkan hanya pada koefisien wavelet pada level resolusi yang paling baik. Alasannya yaitu berdasarkan pertimbangan hanya pada level yang paling baik yang koefisien wavelet empirikalnya cenderung berhubungan yang mana sebagian besar terdiri dari *noise*. Berdasarkan pada level estimasi yang diusulkan oleh Donoho dan Johnstone (1994) dapat dituliskan secara matematis sebagai berikut [10]. Berikut ini merupakan penerapan rumus pada persamaan 2.9 menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
lvl0 = 1;  
mot = 'db1';  
[C, L] = wavedec(x, lvl0, mot);  
cD1 = detcoef(C, L, lvl0);  
xs = median(abs(cD1));  
nLevel = xs / 0.6745;
```

3.3.5.2 Aturan *Shrinkage*

Aturan *Shrinkage* menentukan untuk proses perhitungan nilai *threshold*, yang mana nilai *threshold* dinotasikan dengan simbol λ . Aturan lain dalam perhitungan *threshold* independen yaitu dari fungsi *Shrinkage* ketika yang lain memperoleh *threshold* yang berbeda untuk fungsi *threshold* yang berbeda. Aturan yang lainnya menganggap bahwa sebuah unit skala *noise* , $\sigma = 1$, sedangkan yang lainnya tidak [10].

1) SURE

Berikut ini merupakan penerapan rumus pada persamaan 2.19 menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
a = sort(abs(x)) .^ 2;           % x merupakan Sinyal EEG
b = cumsum(a);
n = length(x);                 % n merupakan panjang Sinyal EEG
c = linspace(n - 1, 0, n);
s = b + c .* a;                % nLevel merupakan Estimasi Level Noise
risk = (n - (2 .* (1 : n)) + s) / n;
[~, ibest] = min(risk);
th = sqrt(a(ibest));
th = th * nLevel ;
```

3.3.5.3 Thresholding

Ada tiga jenis fungsi *thresholding* yang akan digunakan yaitu *hard*, *soft*, dan *adaptive thresholding*. Berikut merupakan fungsi *thresholding* yang akan digunakan, yaitu :

1) Adaptive Threshold

Pada *Adaptive Threshold* menggunakan aturan SURE untuk mencari nilai *thresholdnya*. Proses *filtering* dengan menggunakan SURE *threshold* dapat dituliskan secara matematis sebagaimana persamaan 2.23. Berikut ini merupakan penerapan rumus pada persamaan 2.23 menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
xf = x .* (abs(x) > th);
```

3.3.5.4 Rekonstruksi Transformasi Wavelet Diskrit

Rekosntruksi digunakan untuk menggabungkan sinyal yang telah di-*thresholding* baik menggunakan *adaptive*. Rekostruksi ini adalah tahap terakhir dari *denoising* Sinyal EEG dengan menggunakan TWD. Berikut adalah kode program yang digunakan untuk proses rekonstruksi dengan menggunakan TWD, yaitu :

```

% rekonstruksi
dden = waverec(xf, L, mot{1});
% normalisasi ke -1 - +1 volt
dden = dden ./ max(abs(dden));
% centering
dden = dden - mean(dden);

```

3.3.5 Denoising Filter Hilbert Diskrit

Proses *denoising* dengan menggunakan FHD dilakukan dengan menggunakan panjang filter yang berbeda-beda. Ada dua jenis panjang filter yang digunakan untuk proses *denoising* dengan menggunakan FHD yaitu panjang filter genap dan ganjil. Berikut merupakan Filter Hilbert Diskrit yang akan digunakan, yaitu :

3.3.6.1 Filter Hilbert Diskrit Genap

Dikatakan sebagai FHD Genap karena panjang filter yang digunakan untuk proses *filtering* Sinyal EEG yang terkena *noise* AWGN adalah bernilai genap. Berikut merupakan penulisan secara matematis dari FHD Genap, yaitu [24]:

$$h(i) = \frac{2^{N-1}}{N} \sum_{k=0}^{N-1} \sin(w) = \frac{2}{N} \sin^2\left(\frac{\pi i}{2}\right) \cot\left(\frac{\pi i}{N}\right) \quad (3.3)$$

$$i = 0, 1, \dots, N-1, w = \frac{2\pi ki}{N}$$

Dimana,

$h(i)$ = Respon impuls diskrit Hilbert filter

N = Panjang FHD genap

w = Koefisien Detail

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```

x1 = 0:N-1;
hi1 = (2/N)*sin((pi*x1)/2).^2.*cot((pi*x1)/N);
hi1 = hi1(2 : end).';

```

3.3.6.2 Filter Hilbert Diskrit Ganjil

Dikatakan sebagai FHD Ganjil karena panjang filter yang digunakan untuk proses *filtering* Sinyal EEG yang terkena *noise* AWGN adalah bernilai ganjil. Berikut merupakan penulisan secara matematis dari FHD Ganjil, yaitu [24]:

$$h(i) = \frac{1}{N} \left[1 - \frac{\cos\left(\frac{\pi i}{N}\right)}{\cos\left(\frac{\pi i}{N}\right)} \cot\left(\frac{\pi i}{N}\right) \right] \quad (3.4)$$
$$i = 0, 1, \dots, N - 1$$

Dimana,

$h(i)$ = Respon impuls diskrit Hilbert filter

N = Panjang FHD ganjil

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
x3 = 0:N-1; % Menghasilkan nilai awal Inf atau NaN
a1 = cos(pi*x3);
a2 = cos((pi*x3/N));
a3 = cot((pi*x3/N));
hi1 = (1/N)*(1-((a1./a2).*a3));
hi1 = hi1(2 : end).';
```

3.3.6.3 Konvolusi Diskrit Hilbert Filter

FHD yang telah dibuat baik yang panjang filternya ganjil maupun genap akan dikonvolusikan dengan Sinyal EEG yang terkena *noise* AWGN. Hasil konvolusi antara Sinyal EEG yang terkena AWGN dengan FHD inilah yang disebut sebagai Sinyal EEG terfilter. Berikut merupakan bentuk matematis dari konvolusi Sinyal EEG yang terkena *noise* AWGN dengan FHD, yaitu [24]:

$$v(i) = -x(i) \ h(i) \quad (3.5)$$
$$i = 0, 1, \dots, N - 1$$

Dimana,

$v(i)$ = Konvolusi

$x(i)$ = Sinyal Asli

$h(i)$ = Respon impuls diskrit Hilbert filter

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
dden1 = conv(-datax, hi1, 'same'); % Konvolusi Filter Hilbert Genap  
dden2 = conv(-datax, hi1, 'same'); % Konvolusi Filter Hilbert Ganjil
```

1.4 PARAMETER PENELITIAN

Parameter penelitian yang akan digunakan ada dua macam yaitu parameter pengujian dan parameter sistem. Berikut merupakan parameter-parameter yang akan digunakan dalam penelitian ini, yaitu :

1.4.1 Parameter Pengujian

Parameter pengujian adalah parameter yang digunakan untuk menguji sistem yang telah dibuat untuk proses *denoising* Sinyal EEG dengan menggunakan metode TWD dan FHD. Berikut merupakan parameter pengujian yang akan digunakan, yaitu :

1.4.1.1 Parameter Transformasi Wavelet Diskrit

Berikut ini merupakan parameter pengujian yang akan digunakan dalam *denoising* AWGN pada Sinyal EEG dengan menggunakan metode TWD, yaitu :

Tabel 3.1 Parameter Pengujian TWD

| No | Jenis Mather Wavelet | Level Dekomposisi | <i>Thresholding</i> |
|----|-------------------------|----------------------|---------------------|
| 1 | Sym2, sym5, sym8 | 1-2 | <i>Adaptive</i> |

1.4.1.2 Parameter Filter Hilbert Diskrit

Berikut ini merupakan parameter pengujian yang akan digunakan dalam *denoising noise* AWGN pada Sinyal EEG dengan menggunakan metode Diskrit Hilbert Filter, yaitu :

Tabel 3.2 Parameter Pengujian FHD Genap

| No | Panjang Filter | Kelipatan | | |
|----|----------------|---------------|------------------|--------------------|
| 1 | Tahap 1 | 2, 4, ...,20 | 22, 24, ...,40 | 42, 44, ...,60 |
| 2 | Tahap 2 | 2, 6, ..., 38 | 42, 46, ..., 78 | 82, 86, ..., 118 |
| 3 | Tahap 3 | 2, 8, ..., 56 | 62, 68, ..., 116 | 122, 128, ..., 176 |

Tabel 3.3 Parameter Pengujian FHD Ganjil

| No | Panjang Filter | Kelipatan | | |
|----|----------------|---------------|------------------|--------------------|
| 1 | Tahap 1 | 1, 3, ...,19 | 21, 23, ...,39 | 41, 43, ...,59 |
| 2 | Tahap 2 | 1, 5, ..., 37 | 41, 45, ..., 77 | 81, 85, ..., 117 |
| 3 | Tahap 3 | 1, 7, ..., 55 | 61, 67, ..., 115 | 121, 127, ..., 175 |

1.4.2 Parameter Sistem

Parameter sistem yang akan digunakan adalah SNR dan MSE, parameter sistem ini berfungsi untuk mengetahui kinerja dari sistem yang telah dibuat untuk proses *denoising*. Berikut merupakan parameter sistem yang akan digunakan, yaitu :

1.4.2.1 Signal Noise to Ratio

SNR digunakan untuk mengetahui kinerja dari sistem yang dibuat. Pengukuran kinerja sistem dilakukan dengan cara membandingkan SNR sebelum dan sesudah dilakukan proses *denoising* pada Sinyal EEG. SNR ini biasanya merupakan ukuran yang paling penting daripada kekuatan *noise*, ketika mempertimbangkan bahwa kekuatan sinyal untuk *denoising noise* dapat bervariasi. Semakin besar nilai SNR yang dihasilkan maka kualitas sinyal akan semakin bagus dan sebaliknya. Secara matematis bentuk persamaan dari SNR, yaitu [8] :

$$SNR_{before} = 10 \log_{10} \left(\frac{(\bar{x}(n))^2}{(\bar{x}'(n))^2} \right) \quad (3.6)$$

Dimana,

$\bar{x}(n)$ = Rata-rata nilai Sinyal EEG Asli

$\bar{x}'(n)$ = Nilai rata-rata Sinyal EEG yang ditambahkan AWGN

$$SNR_{after} = 20 \log_{10} \left(\frac{\text{Peak to Peak (EEG)}}{4 \cdot \text{std(EEG)}} \right) \quad (3.7)$$

Dimana,

Peak to Peak (EEG) = Mengukur nilai amplitudo maximum-minus minimum S1 Sinyal EEG

std(EEG) = Mengukur *noise* pada Sinyal EEG

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
% SNR Sebelum Denoising tanpa toolbox
snr_before = mean(data.^2) / mean(datax.^2);
SNR1 = 10 * log10(snr_before);
% SNR Sebelum Denoising dengan toolbox
% SNR1 = snr(data, datax);
% SNR Sesudah Denoising
p2p = peak2peak(dden);
stdp = std(dden);
cpr = p2p ./ (4 * stdp);
SNR2 = abs(20 * log10(cpr));
fprintf(' SNR1 : %.2f', SNR1(id, ix));
fprintf(' SNR2 : %.2f\n', SNR2(id, ix));
```

1.4.2.2 Mean Square Error

MSE adalah parameter yang digunakan untuk mengetahui tingkat *error* dari proses *denoising* yang telah dilakukan. Jika nilai MSE semakin mendekati 0 maka itu menandakan bahwa sistem yang telah dibuat memiliki kehandalan yang bagus. Secara matematis MSE dapat dituliskan sebagai berikut :

$$MSE = \frac{\sum_{i=1}^n (s - s_e)^2}{n} \quad (3.8)$$

Dimana,

s = Sinyal Asli

s_e = Sinyal hasil *denoising*

n = Panjang sinyal

Berikut ini merupakan penerapan rumus menjadi kode program dengan menggunakan *software* Matlab R2016a, yaitu :

```
MSE = (mean((data - dden).^2))*100;  
fprintf('[MSE : %.2f%%', MSE);
```