

LAMPIRAN A

AR MODEL METODE BURG

Salah satu metode yang dapat digunakan dalam penentuan parameter AR Model adalah metode Burg. Pembahasan mengenai *lattice filter* akan dibahas terlebih dahulu untuk menunjang pembahasan metode ini.

A.1 *Lattice filter*

Lattice filter merupakan salah satu struktur filter FIR (*Finite Impulse Response*). Persamaan umum dari filter FIR ditunjukkan dalam persamaan A.1. Nilai m pada persamaan tersebut menunjukkan orde polinomial.

$$H_m(z) = 1 + \sum_{k=1}^m \alpha_m(k)z^{-k} \quad m \geq 1 \quad \dots\dots\dots(A.1)$$

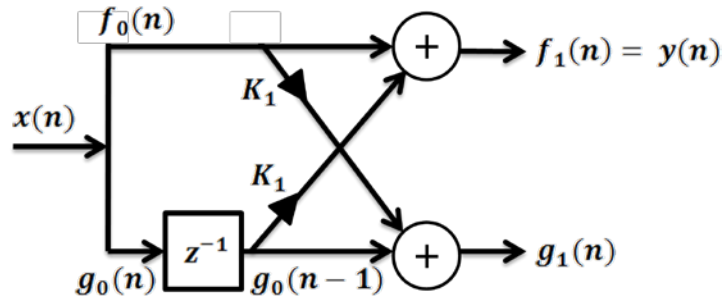
Bila diasumsikan bahwa deret input terhadap filter $H_m(z)$ adalah $x(n)$, dan deret output adalah $y(n)$, maka persamaannya dapat dituliskan dalam persamaan A.2.

$$y(n) = x(n) + \sum_{k=1}^m \alpha_m(k)x(n - k) \quad \dots\dots\dots(A.2)$$

Asumsikan bahwa filter memiliki orde satu, sehingga persamaan A.2 dapat dituliskan menjadi persamaan A.3.

$$y(n) = x(n) + \alpha_1(1)x(n - 1) \quad \dots\dots\dots(A.3)$$

Persamaan A.3 ternyata bisa didapatkan dari *single - stage lattice filter* yang ditunjukkan pada Gambar A.1. Kedua input bagi filter tersebut adalah $x(n)$, dan output yang dipilih adalah output pada cabang atas, yaitu $f_1(n)$.



Gambar A.1 Single – stage lattice filter

Persamaan output dari filter tersebut ditunjukkan pada persamaan A.4. Bila dipilih nilai $K_1 = \alpha_1(1)$, maka persamaan A.4 akan ekivalen dengan persamaan A.3. Parameter K_1 pada persamaan A.4 dikenal sebagai koefisien refleksi dari *lattice filter*.

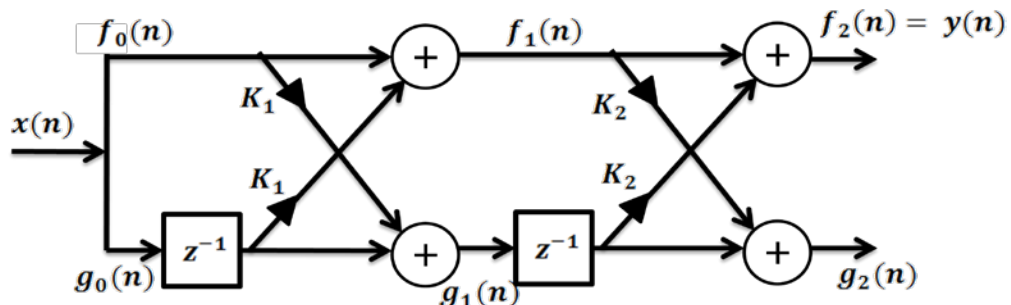
$$f_1(n) = x(n) + K_1x(n - 1)$$

$$g_1(n) = K_1x(n) + x(n - 1) \dots\dots\dots(A.4)$$

Bila orde pada filter adalah 2, maka persamaan deret outputnya dapat dituliskan dalam persamaan A.5

$$y(n) = x(n) + \alpha_2(1)x(n - 1) + \alpha_2(2)x(n - 2) \dots\dots\dots(A.5)$$

Persamaan A.5 juga bisa didapatkan dari persamaan output *two – stage lattice filter* yang ditunjukkan pada Gambar A.2.



Gambar A.2 Two – stage lattice filter

Persamaan output dari *two – stage lattice filter* ditunjukkan pada persamaan A.6. Nilai dari $f_1(n)$ dan $g_1(n)$ dari persamaan tersebut ekuivalen dengan nilai pada persamaan II.9.

$$\begin{aligned}
 f_2(n) &= f_1(n) + K_2 g_1(n - 1) \\
 g_2(n) &= K_2 f_1(n) + g_1(n - 1) \dots\dots\dots(A.6)
 \end{aligned}$$

Dengan meneruskan proses ini, dapat dilihat bahwa ada ekuivalensi antara FIR filter orde m dengan *m – stage lattice filter*. *Lattice filter* secara umum dapat dideskripsikan dengan *set of order – recursive equations* yang ditunjukkan pada persamaan A.7.

$$\begin{aligned}
 f_0(n) &= g_0(n) = x(n) \\
 f_m(n) &= f_{m-1}(n) + K_m g_{m-1}(n - 1) \dots\dots\dots(A.7) \\
 g_m(n) &= K_m^* f_{m-1}(n) + g_{m-1}(n - 1) \quad m = 1, 2, \dots, p
 \end{aligned}$$

A.2 Metode Burg Dalam Penentuan Parameter AR Model

Metode Burg merupakan sebuah metode yang didasari oleh proses minimalisasi nilai *forward* dan *backward error* dalam prediktor linear untuk menentukan nilai – nilai parameter modelnya. Metode Burg menggunakan realisasi *lattice filter* sehingga sering pula dikenal dengan nama metode *lattice least – square order – recursive*.

Forward Linear Prediction digunakan untuk memprediksi nilai mendatang dari sebuah proses atau deret berdasarkan nilai – nilai terdahulunya. Asumsikan sebuah deret yang tersusun dari data $x(n)$, dengan nilai $n = 0, 1, \dots, N - 1$. Bila prediktor memiliki orde m, nilai $x(n)$ dapat diprediksi melalui kombinasi linear dari nilai - nilai $x(n - 1), x(n - 2), \dots, x(n - m)$ yang telah diboboti dengan koefisien tertentu. Persamaan matematis dari proses prediksi linear ini ditunjukkan pada persamaan A.8.

$$\hat{x}(n) = -\sum_{k=1}^m a_m(k)x(n-k) \dots\dots\dots(A.8)$$

Selisih antara nilai aktual, $x(n)$, dengan nilai prediksi, $\hat{x}(n)$, disebut *forward prediction error*, yang secara matematis dituliskan pada persamaan A.9.

$$f_m(n) = x(n) - \hat{x}(n) \dots\dots\dots(A.9)$$

Backward Linear Prediction digunakan untuk memprediksi nilai terdahulu dari sebuah proses atau deret berdasarkan nilai – nilai sesudahnya. Asumsikan sebuah deret yang tersusun dari data $x(n)$, dengan nilai $n = 0, 1, \dots, N - 1$. Bila prediktor memiliki orde m , nilai $x(n - m)$ dapat diprediksi melalui kombinasi linear dari nilai - nilai $x(n - m + 1), x(n - m + 2), \dots, x(n)$ yang telah diboboti dengan koefisien tertentu. Persamaan matematis dari proses prediksi linear ini ditunjukkan pada persamaan A.10.

$$\hat{x}(n - m) = -\sum_{k=1}^m a_m^*(k)x(n + k - m) \dots\dots\dots(A.10)$$

Selisih antara nilai aktual, $x(n - m)$, dengan nilai prediksi, $\hat{x}(n - m)$, disebut *backward prediction error*, yang secara matematis dituliskan pada persamaan A.11.

$$g_m(n) = x(n - m) - \hat{x}(n - m) \dots\dots\dots(A.11)$$

Berdasarkan persamaan A.9 dan persamaan A.11, bisa didapatkan nilai *average prediction error* yang dituliskan pada persamaan A.12.

$$\varepsilon_m = \frac{1}{2(N-m)} \sum_{n=m}^{N-1} [|f_m(n)|^2 + |g_m(n)|^2] \dots\dots\dots(A.12)$$

Nilai *average prediction error* dapat diminimalkan dengan menentukan nilai koefisien prediksi, a_m dan a_m^* , yang juga harus memenuhi *Levinson – Durbin recursion*. Persamaan dari koefisien prediksi tersebut ditunjukkan pada persamaan A.13.

$$a_m(k) = a_{m-1}(k) + K_m a_{m-1}^*(m - k) \quad 1 \leq k \leq m - 1 \dots\dots\dots(A.13)$$

$$1 \leq m \leq p$$

Pada persamaan A.13, $K_m = a_m$ adalah koefisien refleksi ke $-m$ dalam realisasi *lattice filter*. Bila persamaan A.13 disubstitusikan pada $f_m(n)$ dan $g_m(n)$, akan didapatkan pasangan persamaan *order - recursive* untuk *forward* dan *backward prediction error* yang ditunjukkan pada persamaan A.14.

$$f_0(n) = g_0(n) = x(n)$$

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1) \dots\dots\dots(A.14)$$

$$g_m(n) = K_m^* f_{m-1}(n) + g_{m-1}(n-1) \quad m = 1, 2, \dots, p$$

Persamaan A.14 dapat disubstitusikan pada persamaan A.12 untuk mendapatkan nilai error minimal, ϵ_m , yang merupakan fungsi dari K_m . Persamaan lengkapnya ditunjukkan pada persamaan A.15.

$$\epsilon_m = \frac{1}{2(N-m)} \sum_{n=m}^{N-1} [|f_{m-1}(n) + K_m g_{m-1}(n-1)|^2 + |K_m^* f_{m-1}(n) + g_{m-1}(n-1)|^2] \dots(A.15)$$

Persamaan A.15 kemudian didiferensialkan. Bila hasil diferensial bernilai nol, maka bisa didapatkan nilai koefisien refleksi yang ditunjukkan pada persamaan A.16.

$$\hat{K}_m = \frac{-\sum_{n=m}^{N-1} f_{m-1}(n) g_{m-1}^*(n-1)}{\frac{1}{2} \sum_{n=m}^{N-1} [|f_{m-1}(n)|^2 + |g_{m-1}(n-1)|^2]} \quad m = 1, 2, \dots, p \dots\dots(A.16)$$

Numerator pada persamaan A.16 merupakan estimasi dari *crosscorrelation* antara nilai *forward* dan *backward prediction error*. Normalisasi pada denominator mengakibatkan nilai $|K_m| < 1$, sehingga model yang dihasilkan stabil.

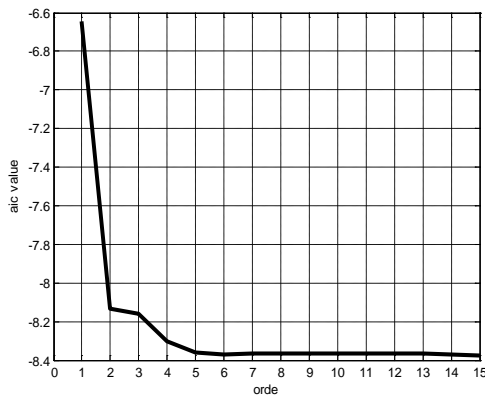
Algoritma Burg, dapat disimpulkan, melakukan perhitungan nilai koefisien refleksi dalam persamaan *lattice filter* pada persamaan A.16. Algoritma *Levinson - Durbin* yang ditunjukkan pada persamaan A.13 kemudian digunakan untuk mendapatkan nilai parameter dari AR Model.

LAMPIRAN B

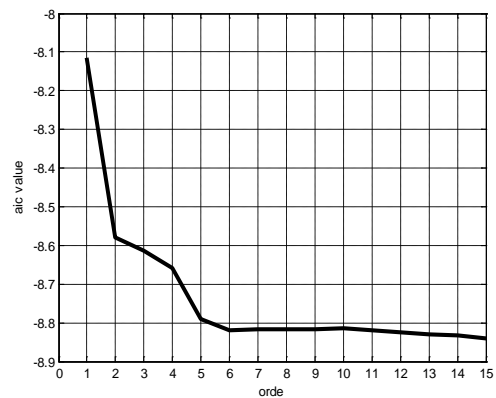
PLOT SINYAL UNTUK PENENTUAN ORDE MODEL

Pada bagian ini akan disajikan plot dari 20 sinyal hasil pencuplikan untuk penentuan orde model terbaik. Kriteria uji kecocokan dan AIC digunakan dalam penentuan orde model ini. Pengujian pada masing – masing sinyal dilakukan pada kanal MLII dan V1.

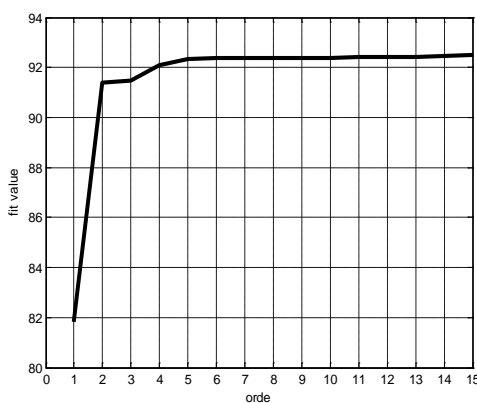
1. Data *record* 112 kondisi normal



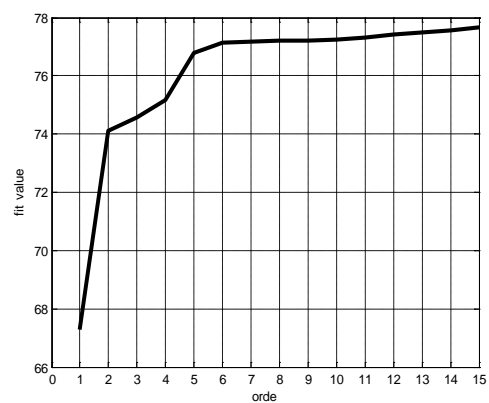
(a)



(b)



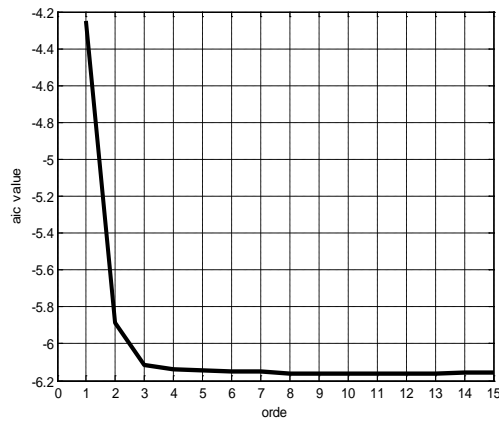
(c)



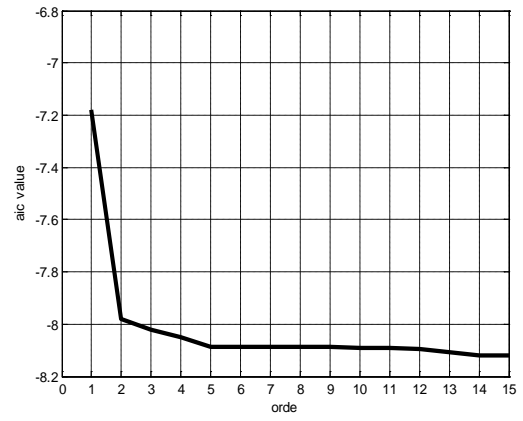
(d)

Gambar B.1 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

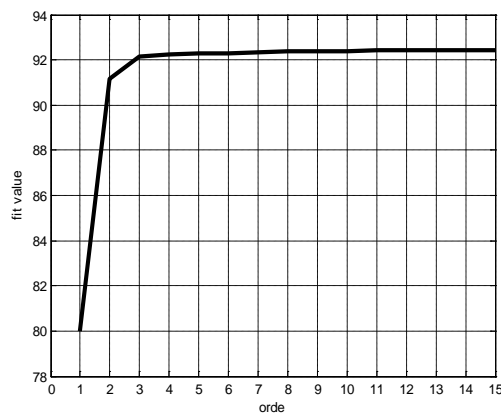
2. Data record 116 kondisi normal



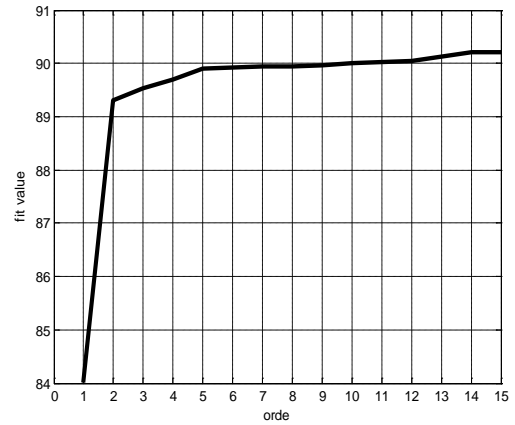
(a)



(b)



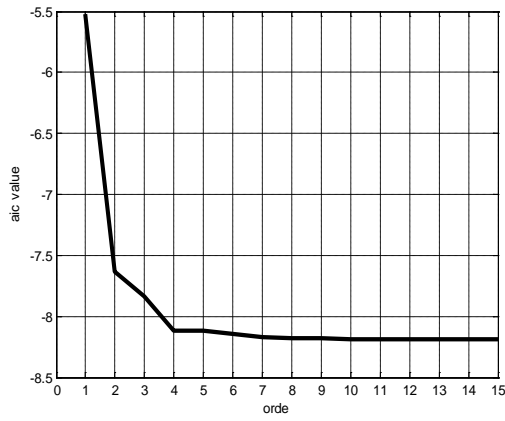
(c)



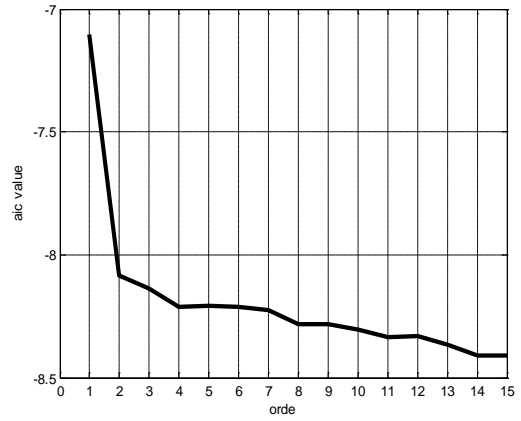
(d)

Gambar B.2 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

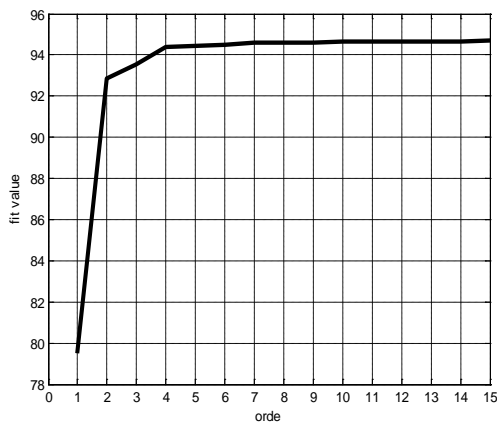
3. Data record 234 kondisi normal



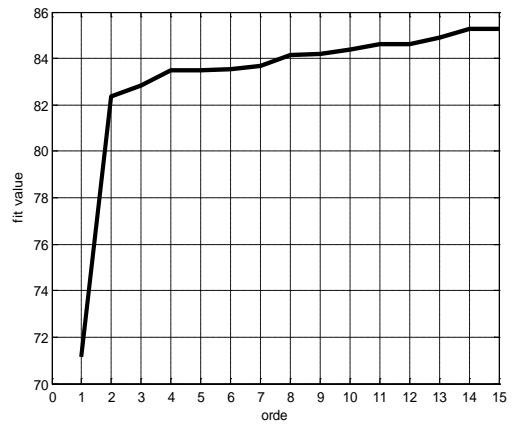
(a)



(b)



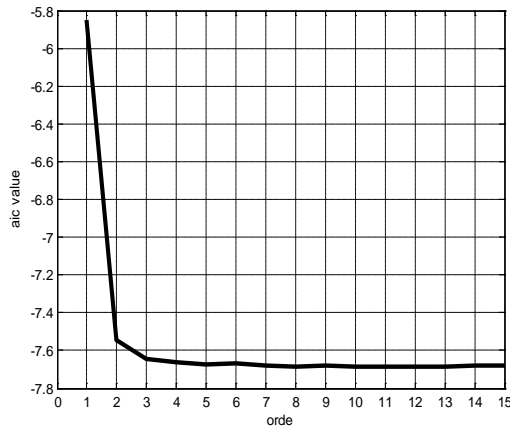
(c)



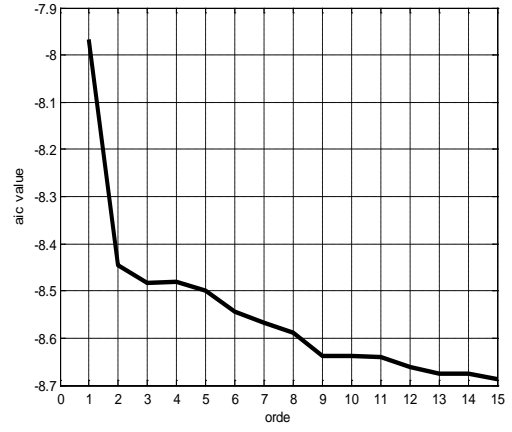
(d)

Gambar B.3 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

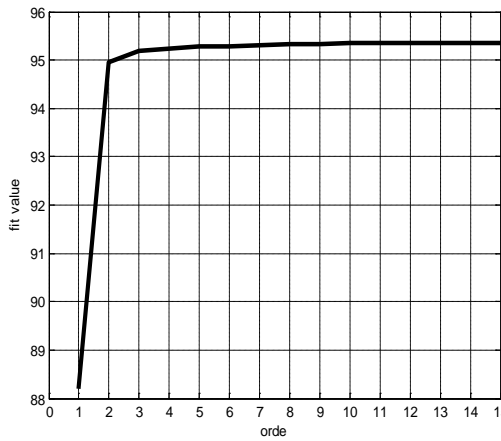
4. Data record 200 kondisi *ventricular tachycardia*



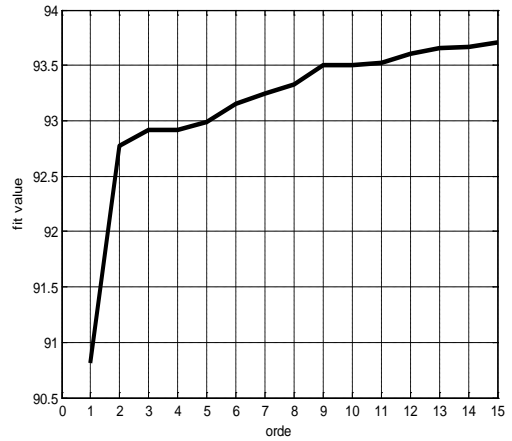
(a)



(b)



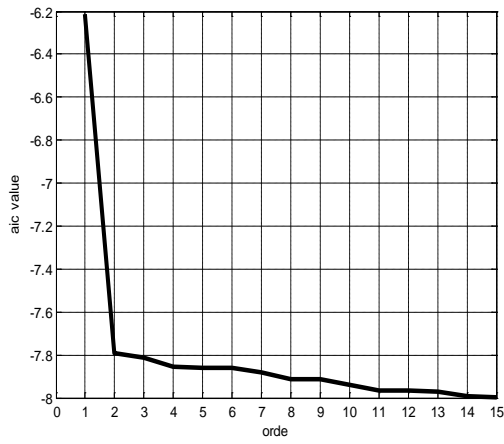
(c)



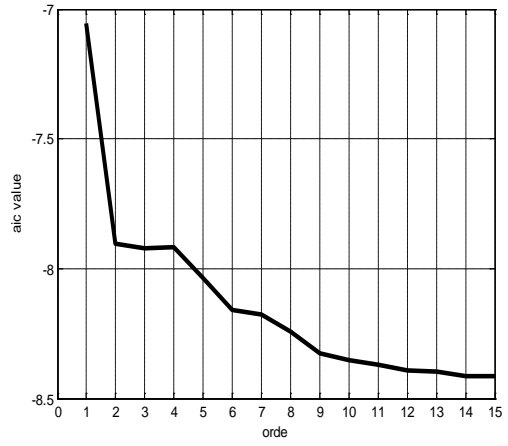
(d)

Gambar B.4 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

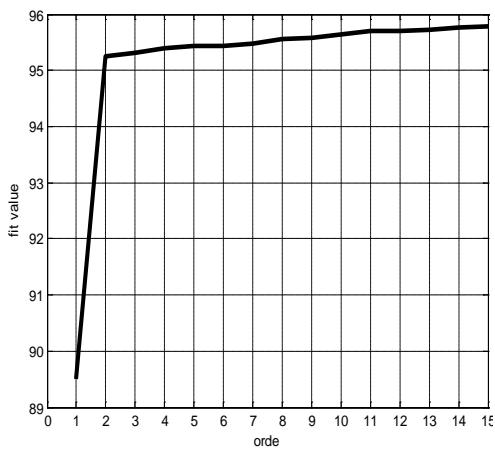
5. Data record 214 kondisi *ventricular tachycardia*



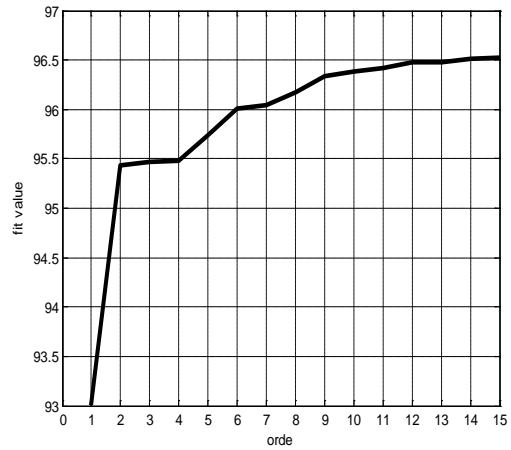
(a)



(b)



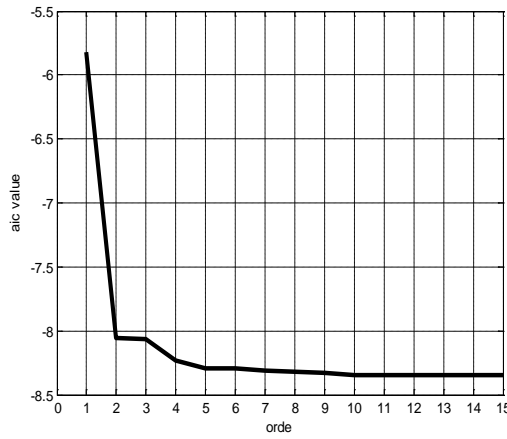
(c)



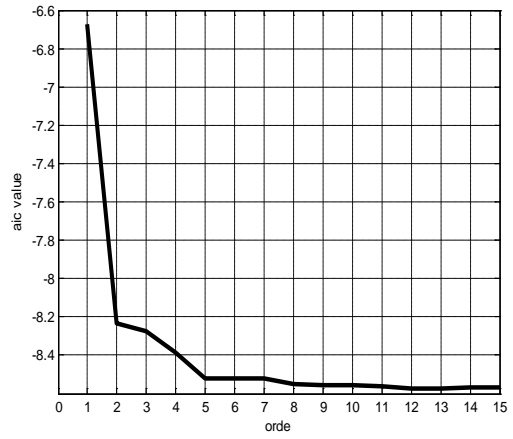
(d)

Gambar B.5 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

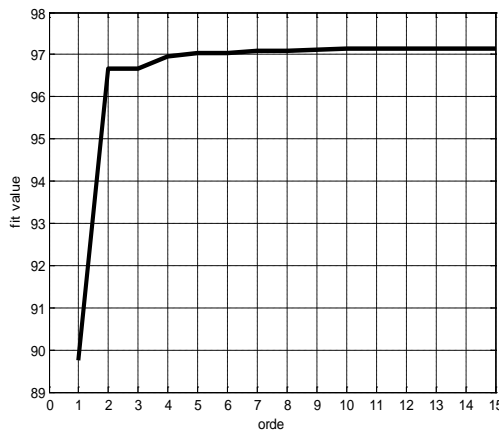
6. Data record 223 kondisi *ventricular tachycardia*



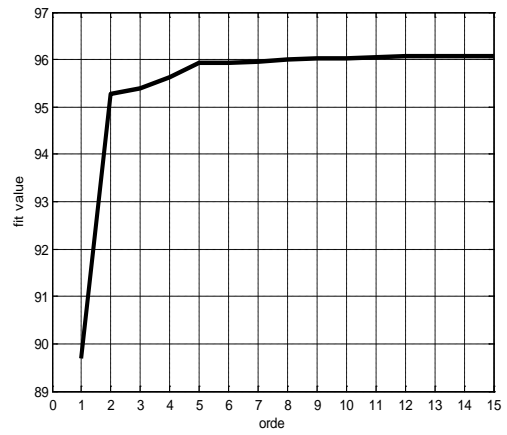
(a)



(b)



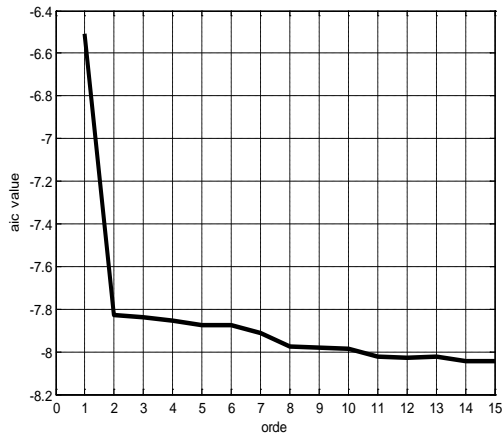
(c)



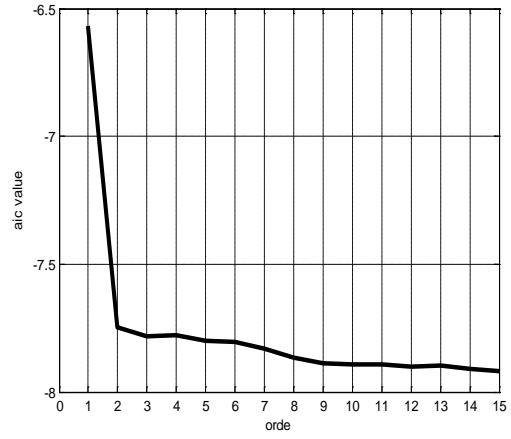
(d)

Gambar B.6 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

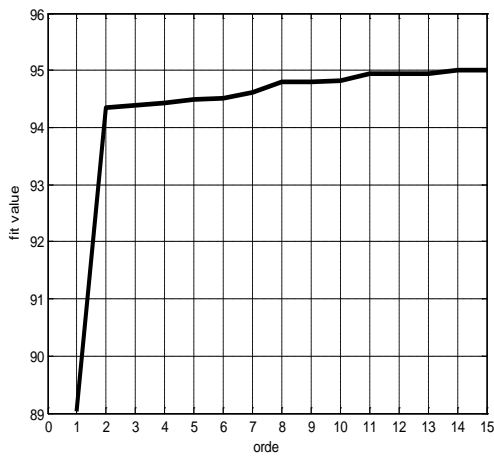
7. Data record 106 kondisi *ventricular bigeminy*



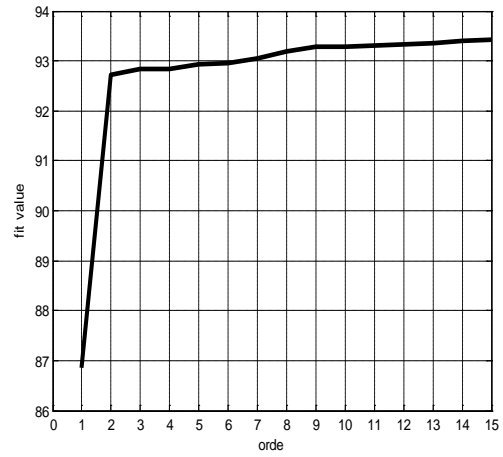
(a)



(b)



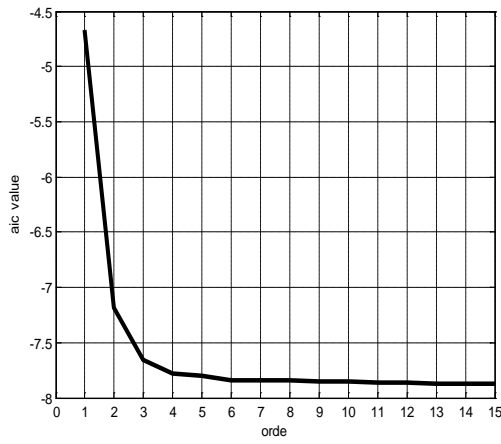
(c)



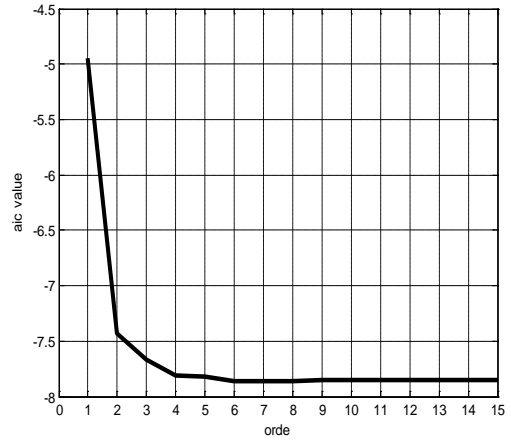
(d)

Gambar B.7 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

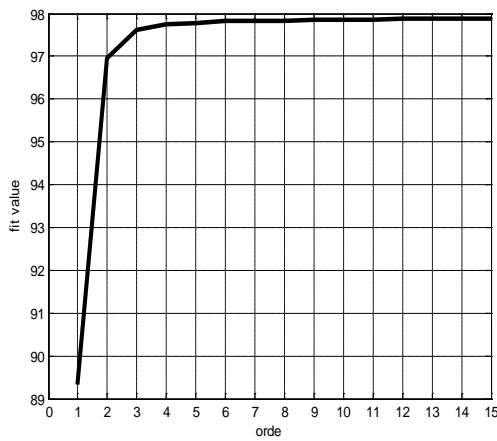
8. Data record 213 kondisi *ventricular bigeminy*



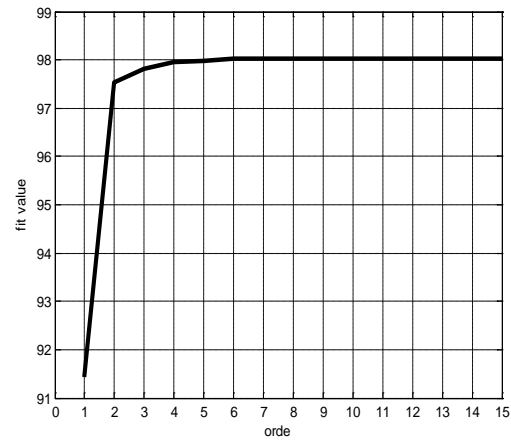
(a)



(b)



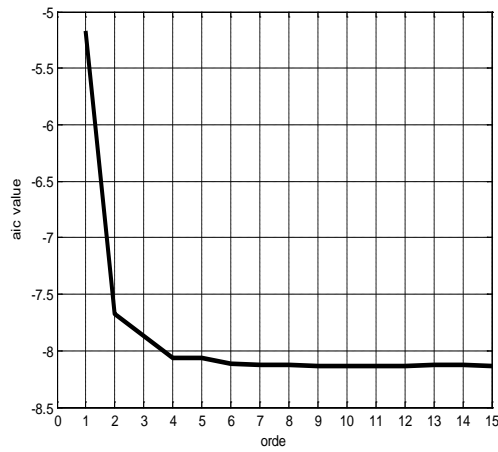
(c)



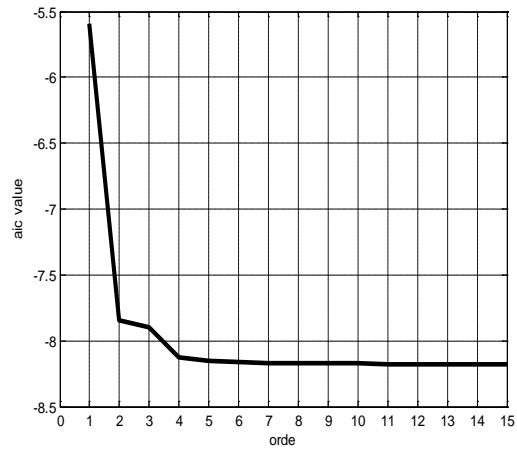
(d)

Gambar B.8 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

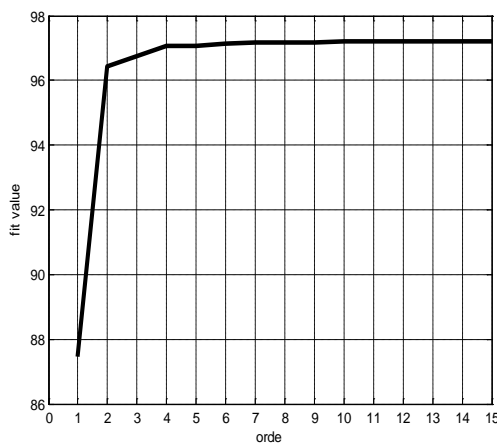
9. Data record 233 kondisi *ventricular bigeminy*



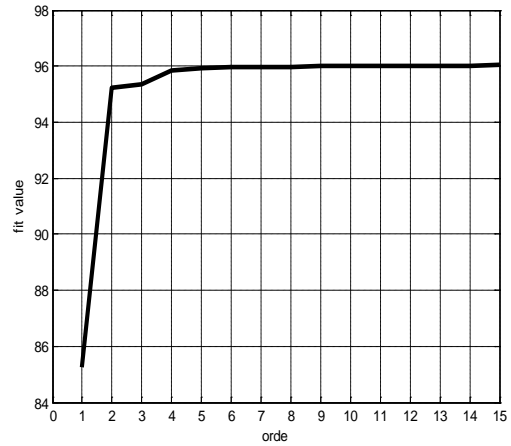
(a)



(b)



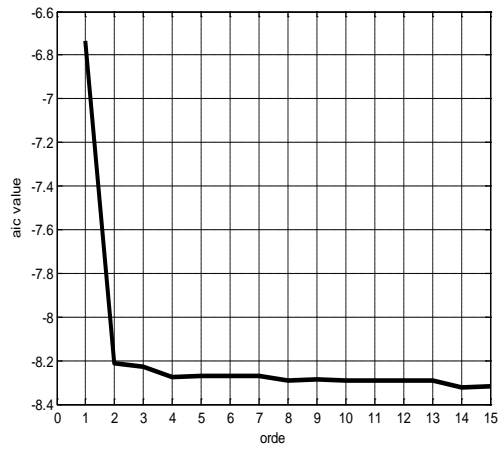
(c)



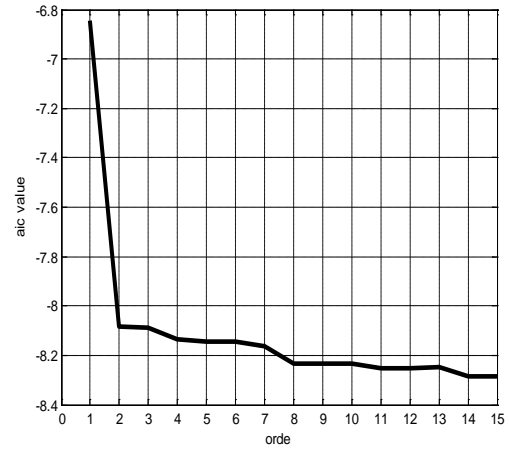
(d)

Gambar B.9 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

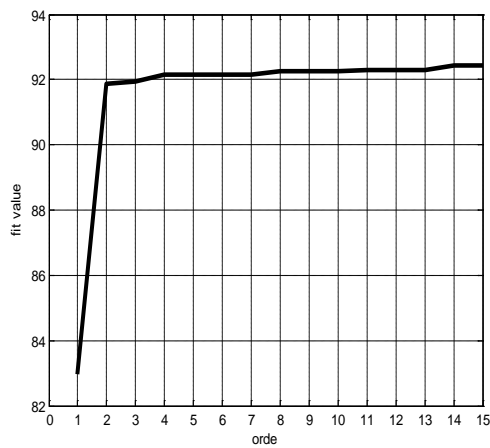
10. Data record 201 kondisi atrial fibrillation



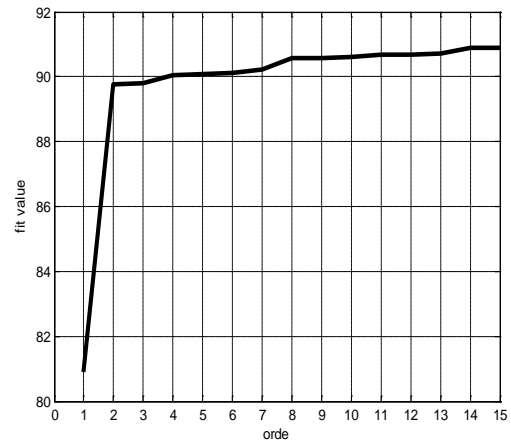
(a)



(b)



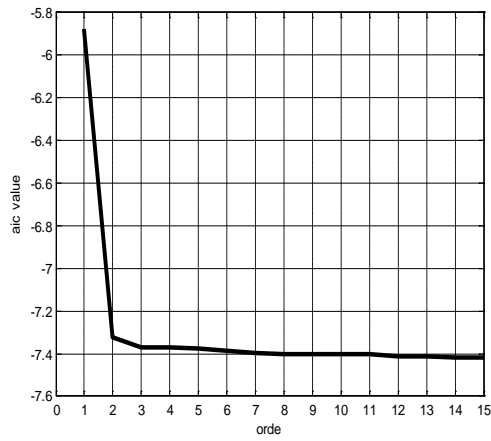
(c)



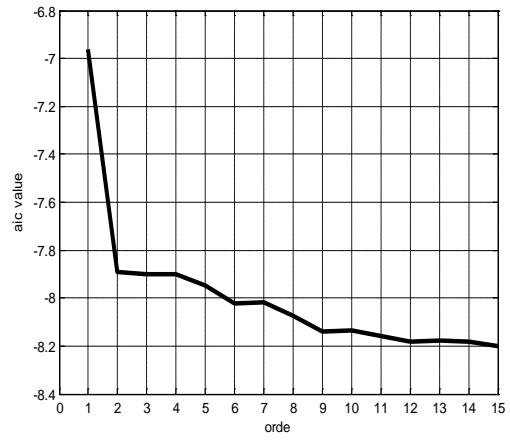
(d)

Gambar B.10 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

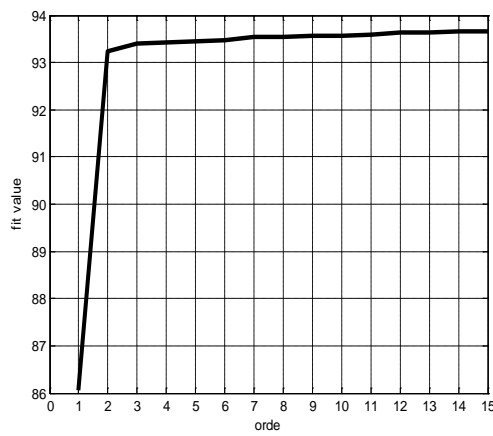
11. Data record 203 kondisi atrial fibrilation



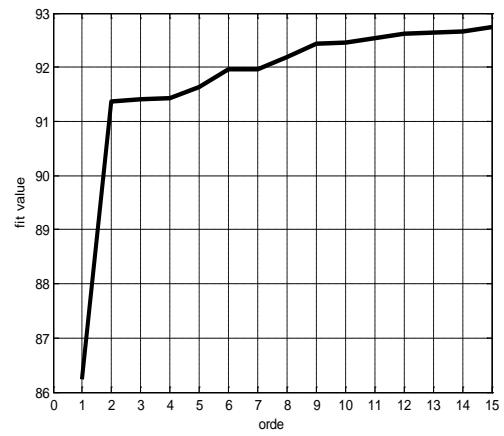
(a)



(b)



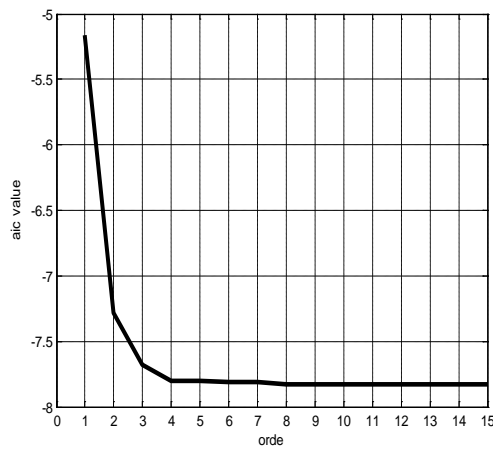
(c)



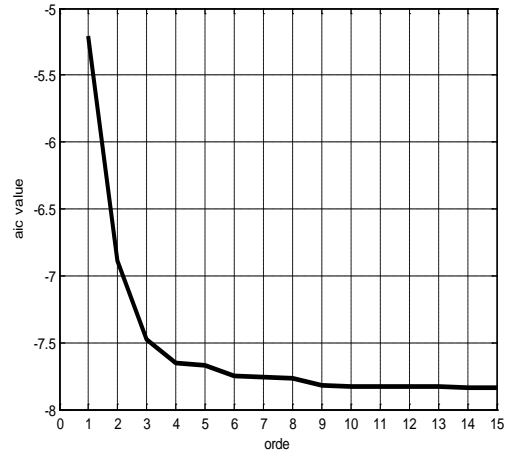
(d)

Gambar B.11 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

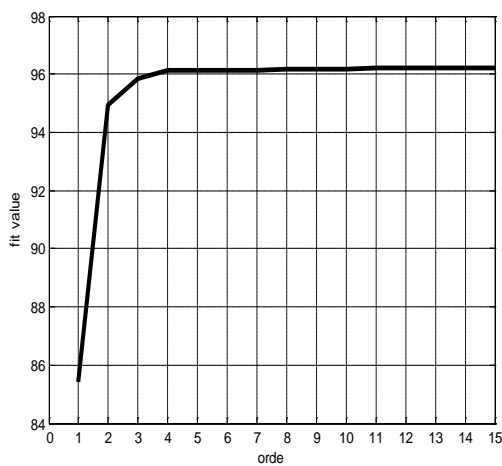
12. Data record 219 kondisi atrial fibrillation



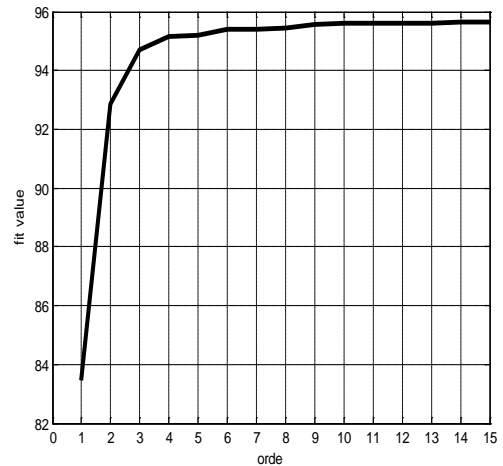
(a)



(b)



(c)



(d)

Gambar B.12 (a) Plot AIC kanal MLII (b) Plot AIC kanal V1 (c) Plot uji kecocokan kanal MLII (d) Plot uji kecocokan kanal V1

LAMPIRAN C

CONTOH PEMROGRAMAN MATLAB

Pada bagian ini akan disajikan contoh – contoh pemrograman pada MATLAB yang digunakan dalam penyusunan Tugas Akhir ini. Beberapa contoh yang akan disajikan adalah contoh program untuk mendesain Jaringan Saraf Tiruan dan perhitungan kriteria uji kecocokan dan AIC. Program selengkapnya dapat dilihat pada CD yang akan disertakan bersama laporan ini.

C.1. Desain dan Pengujian dengan Jaringan Saraf Tiruan

Beberapa desain Jaringan Saraf Tiruan telah digunakan dalam Tugas Akhir ini. Salah satu contoh desain Jaringan Saraf Tiruan yang digunakan adalah dengan menggunakan arsitektur 3 *layer* dengan masing – masing jumlah *neuron* 30, 20, dan 4. Algoritma pembelajaran yang digunakan adalah *backpropagation* metode *gradient descent with momentum* dengan orde model 5. Proses pengujian untuk kategori data uji sama dengan data latih dan data uji tidak sama dengan data latih juga dilakukan secara langsung setelah proses pembelajaran dilakukan. Berikut ini adalah list program yang digunakan.

```
disp('backpropagation neural network(30-20-4),traingdm,orde 5');
load koefburg.mat; %load input dan target
load inafib.mat;
load inb.mat;
load invt.mat;
load innormal.mat;
p = p(:,4:15);
t1 = [0;0;0;1];
t2 = [0;0;1;0];
t3 = [0;1;0;0];
t4 = [1;0;0;0];
t = [t1 t1 t1 t2 t2 t2 t3 t3 t3 t4 t4 t4];
netgdmI1 = newff(minmax(p),[30 20 4],{'purelin' 'tansig' 'logsig'},'traingdm');
netgdmI1.trainParam.show = 50;
netgdmI1.trainParam.lr = 0.5;
netgdmI1.trainParam.mc = 0.9;
```

```

netgdmI1.trainParam.goal = 1e-3;
netgdmI1.trainParam.epochs = 50000;
netgdmI1 = init(netgdmI1);
[netgdmI1,tr] = train(netgdmI1,p,t);
sim(netgdmI1,p)
sim(netgdmI1,pafib)
sim(netgdmI1,pb)
sim(netgdmI1,pvt)
sim(netgdmI1,pnormal)

```

C.2. Perhitungan Kriteria Uji Kecocokan dan AIC

Pengujian kriteria uji kecocokan dan AIC dilakukan pada 20 data seperti yang disajikan pada bagian Lampiran B. Berikut adalah list program yang digunakan.

```

disp('fit and aic for arburg model');
s = input('signal=' ); %input sinyal asli
s = s';
s = iddata(s);
for i=1:15
    bu = ar(s,i,'burg');
    [yh,fit] = compare(s,bu,1);
    fitt(i) = fit;
    am = aic(bu);
    aics(i) = am;
end
angka = 1:15;
plot(angka,aics);
xlabel('orde')
ylabel('aic value')
figure
plot(angka,fitt);
xlabel('orde')
ylabel('fit value')

```