



TESIS - KS185411

**KLASIFIKASI SINYAL *ICTAL*  
DAN *OCULAR ARTIFACTS* MENGGUNAKAN  
*INPUT TIME FREQUENCY FEATURES*  
PADA HASIL PEREKAMAN *EEG* PASIEN EPILEPSI**

**YOGA PRASTYA IRFANDI**  
062117 5001 0002

**DOSEN PEMBIMBING**  
Santi Wulan Purnami, M.Si., Ph.D.  
Irhamah, M.Si., Ph.D.

Program Magister  
Departemen Statistika  
Fakultas Sains Dan Analitika Data  
Institut Teknologi Sepuluh Nopember  
SURABAYA  
2020





TESIS - KS185411

**KLASIFIKASI SINYAL *ICTAL* DAN *OCULAR*  
*ARTIFACTS* MENGGUNAKAN *INPUT*  
*TIME FREQUENCY FEATURES*  
PADA HASIL PEREKAMAN *EEG* PASIEN EPILEPSI**

Yoga Prastya Irfandi  
062117 5001 0002

DOSEN PEMBIMBING  
Santi Wulan Purnami, M.Si., Ph.D.  
Irhamah, M.Si., Ph.D.

Departemen Statistika  
Fakultas Sains dan Analitika Data  
Institut Teknologi Sepuluh Nopember  
2020





TESIS - KS185411

***CLASSIFICATION OF ICTAL AND OCULAR  
ARTIFACTS USING TIME FREQUENCY FEATURES  
FROM EPILEPTIC PATIENS EEG RECORDING***

Yoga Prastya Irfandi  
NRP. 062117 5001 0002

Supervisors  
Santi Wulan Purnami, M.Si., Ph.D.  
Irhamah, M.Si., Ph.D.

Department of Statistics  
Faculty of Science and Data Analytics  
Institut Teknologi Sepuluh Nopember  
2020



## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
**Magister Statistika (M.Stat)**

di

Institut Teknologi Sepuluh Nopember

Oleh:

**YOGA PRASTYA IRFANDI**

**NRP: 06211750010002**

Tanggal Ujian: 21 Januari 2020

Periode Wisuda: Maret 2020

Disetujui Oleh:

**Pembimbing:**

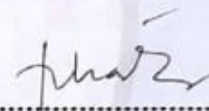
1. Santi Wulan Purnami, M.Si., Ph.D

NIP:19720923 199803 2 001



2. Irhamah, M.Si., Ph.D

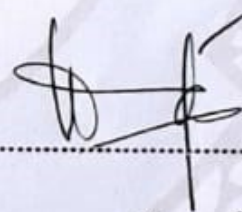
NIP: 19780406 200112 2 002



**Penguji:**


1. Dr. Bambang Widjanarko Otok, M.Si

NIP:19681124199412 1 001



2. Dr.rer.pol. Dedy Dwi Prastyo, M.Si

NIP: 19831204 200812 1 002



**Kepala Departemen Statistika  
Fakultas Sains dan Analitika Data**

**Dr. Dra. Kartika Fithriasari, M.Si**

**NIP: 19691212 199303 2 002**



(Halaman ini sengaja dikosongkan)



**Klasifikasi Sinyal *Ictal* dan *Ocular Artifacts* menggunakan *Input Time Frequency Features* pada Hasil Perekaman *EEG* Pasien Epilepsi**

Nama Mahasiswa : Yoga Prastya Irfandi  
NRP : 062117 5001 0002  
Pembimbing : Santi Wulan Purnami, M.Si., Ph.D  
Co-Pembimbing : Irhamah, M.Si., Ph.D

**ABSTRAK**

Epilepsi merupakan gangguan fungsi otak yang menyerang kurang lebih 50 juta orang di seluruh dunia, dengan gejala khas kejang secara tiba-tiba tanpa ditandai penyebab yang jelas. Dokter atau ahli saraf pada umumnya menggunakan alat *Elektroencephalogram (EEG)* untuk merekam sinyal otak pasien untuk keperluan diagnosa penyakit epilepsi. Saat perekaman terdapat sinyal lain yang tertangkap *EEG*, dengan kejadian terbanyak adalah akibat dari pergerakan mata (*ocular artifacts*) pasien selama perekaman. *Ocular Artifacts (OA)* dapat meningkatkan resiko kesalahan dalam melakukan diagnosa penyakit epilepsi. Sehingga perlu dilakukan klasifikasi untuk membedakan sinyal *ictal* dan *OA*. Data hasil pengamatan dipotong sesuai catatan dokter dengan durasi 2 detik dan 5 detik di setiap potongnya. Algoritma dekomposisi yang digunakan adalah *empirical mode decomposition (EMD)*, dan diperoleh sebanyak 4 *IMF*. Dari *IMF* selanjutnya diekstraksi menjadi fitur  $B_{AM}$ ,  $B_{FM}$  untuk *input* klasifikasi skenario pertama dan *time frequency features* seperti *Root Mean Square (RMS) frequency  $f_r$*  dan *dominant frequency  $f_d$*  dan *RMIFS* untuk skenario pengelompokan kedua. Klasifikasi dilakukan dengan menggunakan *SVM* dengan kernel *RBF*, *linier* dan *polynomial*, dengan optimasi tuning parameter menggunakan metode *Grid Search*. Hasil klasifikasi terbaik diperoleh dari data dengan pemotongan berdurasi 2 detik dan fitur  $B_{FM}$  untuk skenario pertama serta *RMIFS* untuk skenario kedua.

**Kata Kunci :** *Epilepsi, EEG, ictal, OA, EMD, time frequency features , SVM*

(Halaman ini sengaja dikosongkan)

**Classification of Ictal and Ocular Artifacts  
Using Time Frequency Features  
From Epileptic Patients EEG Recording**

Student Name : Yoga Prastya Irfandi  
NRP : 062117 5001 0002  
Supervisor : Santi Wulan Purnami, M.Si., Ph.D  
Co-Supervisor : Irhamah, M.Si., Ph.D

***ABSTRACT***

Epilepsy is a brain function disorder that attacks approximately 50 million people worldwide, with the typical symptoms of sudden seizures without marked causes. Doctors or neurologists generally use an Electroencephalogram (EEG) to record the patient's brain signals for the purpose of diagnosing epilepsy. When recording there is another signal that is captured by EEG, with the most occurrence being the result of the patient's ocular artifacts during recording. Ocular Artifacts (OA) can increase the risk of errors in diagnosing epilepsy. So it is necessary to do a classification to distinguish ictal and OA signals. Observation data were cut according to doctor's notes with a duration of 2 seconds and 5 seconds in each pieces. The decomposition algorithm used is empirical mode decomposition (EMD), and obtained as many as 4 IMF. From the IMF then extracted into BAM, BFM features for the *input* of the first scenario classification and time frequency features such as Root Mean Square (RMS) frequency  $f_r$  and dominant frequency  $f_d$  and RMIFS for the second classification scenario. Classification is done using SVM with RBF, linear and polynomial kernels, with optimization of tuning parameters using the Grid Search method. The best classification results are obtained from data with a 2 second cut and BFM features for the first scenario and RMIFS for the second scenario.

***Keyword : Epilepsi, EEG, ictal, OA, EMD, time frequency features , SVM***

(Halaman ini sengaja dikosongkan)

## KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT yang maha menguasai segala ilmu dan alam. Atas rahmat, ridho dan hidayah-Nya sehingga pengerjaan serta penulisan Tesis dengan judul “*Klasifikasi Sinyal Ictal dan Ocular Artifacts menggunakan Input Time Frequency Features pada Hasil Perekaman EEG Pasien Epilepsi*” dapat terselesaikan dengan baik dan lancar.

Penulisan Tesis ini adalah salah satu syarat yang harus dipenuhi dalam memperoleh gelar Magister sesuai dengan kurikulum Departemen Statistika FSAD-ITS Surabaya. Dalam penyelesaian Tesis serta laporan ini penulis tidak terlepas dari bantuan serta dukungan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Dr. Dra. Kartika Fithriasari, M.Si. selaku Ketua Departemen Statistika ITS yang telah banyak memberikan inspirasi kepada mahasiswa untuk senantiasa berkarya.
2. Santi Wulan Purnami, M.Si., Ph.D. dan Irhamah, M.Si., Ph.D. selaku dosen pembimbing yang dengan sabar memberikan bimbingan, arahan, dan masukan selama pengerjaan Tesis.
3. Dr. Bambang Widjanarko Otok, M.Si. dan Dr.rer.pol. Dedy Dwi Prastyo, M.Si. selaku dosen penguji yang telah memberikan banyak tambahan ilmu selama proses perbaikan laporan Tesis.
4. Dr.rer.pol. Dedy Dwi Prastyo, M.Si selaku dosen wali yang telah membimbing dan mengarahkan semenjak awal perkuliahan.
5. Dr.rer.pol. Dedy Dwi Prastyo, M.Si selaku Ketua Program Studi Pascasarjana Statistika ITS yang memberikan motivasi dalam pendidikan.
6. Seluruh dosen pengajar serta karyawan di departemen Statistika ITS, yang telah memberikan bantuan dan ilmunya sebagai bekal dalam pengerjaan Tesis.
7. Bapak, Ibu dan adek serta keluarga besar penulis atas segala doa, dukungan materi, motivasi, kepercayaan dan rasa kasih sayang.

8. Teman-teman Sony, Ocid, Sulhan, Hendy, Try, Ciko, Muniroh, Nendy yang sudah membantu, mendukung, dan mendoakan dalam suksesnya pengerjaan tesis ini dan yang selalu mendukung penulis selama melanjutkan studi S2.
9. Teman-teman keluarga besar yaitu Indira Ihnu Brilillant, Baiq Siska, Mbak Dasty Dewi, Senpai Islahul Kamal, Miss Barbara, dan Deby Fakhriyana yang selalu setia menemani penulis selama kuliah S2 Statistika ITS, yang selalu memberi semangat, mendukung, membantu penulis dalam segala hal dan mendoakan dalam suksesnya pengerjaan tesis ini.
10. Mak Linda Chrestina Fadlilah Rachman yang selalu menyediakan tempat mengerjakan tesis dan selalu memberikan semangat serta dukungan selama ini.
11. Teman-teman S2 Statistika ITS angkatan 2017, S2 Statistika ITS angkatan 2018 dan teman-teman BPS angkatan 2017 yang senantiasa semangat belajar dan selalu menginspirasi.
12. Pihak-pihak lain yang telah mendukung dan membantu dalam penyusunan Tesis ini yang tidak mungkin penulis sebutkan satu per satu. Terima kasih.

Penulis menyadari bahwa penyusunan Tesis ini masih jauh dari sempurna, maka kritik dan saran yang membangun akan senantiasa penulis harapkan demi kesempurnaan di masa mendatang. Semoga laporan ini dapat memberikan sum-bangan yang bermanfaat bagi semua pihak

## DAFTAR ISI

	Halaman
<b>LEMBAR PENGESAHAN TESIS .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>ix</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvii</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Tujuan.....	6
1.4 Manfaat.....	7
1.5 Batasan Masalah.....	7
<b>BAB 2 TINJAUAN PUSTAKA.....</b>	<b>9</b>
2.1 Kajian Jurnal Utama dan Jurnal Pendukung .....	9
2.2 <i>Empirical Mode Decomposition</i> .....	11
2.3 <i>Hilbert Transform</i> terhadap <i>IMF</i> .....	12
2.4 Ekstraksi Fitur .....	13
2.4.1 <i>Root Mean Square Frequency, Dominant Frequency</i> dan <i>Root Mean Instaneous Frequency Square</i> .....	13
2.5 <i>Fold Cross Validation</i> .....	16
2.6 <i>Support Vector Machines</i> .....	17
2.6.1 <i>SVM Linearly Separable</i> .....	18
2.6.2 <i>SVM Linearly Nonseparable</i> .....	21
2.6.3 <i>SVM Nonlinearly Separable</i> .....	23
2.7 Evaluasi Performa Metode Klasifikasi.....	27
2.7.1 <i>Accuracy</i> .....	28
2.7.2 <i>Sensitifty</i> .....	28
2.7.3 <i>Specificity</i> .....	29

2.7.4	<i>AUROC</i> .....	29
2.8	Epilepsi .....	31
2.9	<i>Electroencephalogram (EEG)</i> .....	32
2.9.1	<i>EEG</i> untuk Diagnosa Epilepsi .....	35
2.9.2	Pola Sinyal <i>EEG</i> .....	36
<b>BAB 3</b>	<b>METODOLOGI PENELITIAN</b> .....	<b>39</b>
3.1	Sumber Data .....	39
3.2	Struktur Data .....	42
3.3	Langkah Analisis .....	43
3.3.1	Preprocessing .....	43
3.3.2	Klasifikasi Sinyal <i>Ictal</i> dan <i>Ocular Artifacts</i> .....	50
3.4	Diagram Alir Penelitian .....	52
<b>BAB 4</b>	<b>ANALISIS DAN PEMBAHASAN</b> .....	<b>53</b>
4.1	Pemotongan Sampel Sinyal <i>EEG</i> .....	53
4.1.1	Sinyal <i>Ictal</i> .....	54
4.1.2	Sinyal <i>Ocular Artifacts</i> .....	55
4.2	Dekomposisi Sinyal menggunakan <i>Empirical Mode Decomposition</i> ....	56
4.3	Ekstraksi Fitur .....	58
4.3.1	<i>Bandwidth Amplitude (B<sub>AM</sub>)</i> Sinyal <i>Ictal</i> dan Sinyal <i>OA</i> .....	58
4.3.2	<i>Bandwidth Frequency (B<sub>FM</sub>)</i> Sinyal <i>Ictal</i> dan Sinyal <i>OA</i> .....	59
4.3.3	<i>RMS Frequency (f<sub>r</sub>)</i> Sinyal <i>Ictal</i> dan Sinyal <i>OA</i> .....	60
4.3.4	<i>Dominant Frequency (f<sub>d</sub>)</i> Sinyal <i>Ictal</i> dan Sinyal <i>OA</i> .....	60
4.3.5	<i>RMIFS (f<sub>R</sub>)</i> Sinyal <i>Ictal</i> dan Sinyal <i>OA</i> .....	61
4.4	Klasifikasi Sinyal <i>Ictal</i> dan <i>Ocular Artifacts</i> .....	62
4.4.1	Klasifikasi Sinyal <i>Ictal</i> dan <i>Ocular Artifacts</i> Skenario Pertama .....	63
4.4.2	Klasifikasi Sinyal <i>Ictal</i> dan <i>Ocular Artifacts</i> Skenario Kedua .....	65
<b>BAB 5</b>	<b>KESIMPULAN DAN SARAN</b> .....	<b>67</b>
5.1	Kesimpulan .....	67
5.2	Saran .....	67
<b>DAFTAR PUSTAKA</b>	.....	<b>69</b>
<b>LAMPIRAN</b>	.....	<b>75</b>
<b>BIOGRAFI PENULIS</b>	.....	<b>119</b>



## DAFTAR TABEL

<b>Tabel 2.1</b> <i>Confusion Matrix</i> .....	28
<b>Tabel 2.2</b> Kategori Pengklasifikasian Model berdasarkan Nilai <i>AUC</i> .....	30
<b>Tabel 2.3</b> Data untuk Ilustrasi Membuat Kurva <i>ROC</i> .....	30
<b>Tabel 2.4</b> Etiologi Epilepsi .....	31
<b>Tabel 3.1</b> Struktur Sampel Beda Potensial Hasil Pemoangan .....	42
<b>Tabel 3.2</b> Variabel untuk Tahapan Klasifikasi Hasil Ekstraksi Fitur .....	49
<b>Tabel 3.3</b> Struktur data pada tahapan klasifikasi .....	49
<b>Tabel 3.4</b> Kombinasi Fitur untuk Klasifikasi .....	50
<b>Tabel 4.1</b> Kebaikan Klasifikasi Skenario Pertama untuk Data Potongan 2 detik	63
<b>Tabel 4.2</b> Kebaikan Klasifikasi Skenario Pertama untuk Data Potongan 5 detik	64
<b>Tabel 4.3</b> Kebaikan Klasifikasi Skenario Kedua untuk Data Potongan 2 detik ...	65
<b>Tabel 4.4</b> Kebaikan Klasifikasi Skenario Kedua untuk Data Potongan 5 detik ...	66

(Halaman ini sengaja dikosongkan)

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Ilustrasi 5-Folds Cross Validation .....	17
<b>Gambar 2.2</b>	Klasifikasi Linier <i>SVM</i> : (a) <i>Linearly Separable</i> , (b) <i>Linearly Nonseparable</i> .....	18
<b>Gambar 2.3</b>	Klasifikasi <i>Linearly Separable</i> (Härdle dkk., 2014).....	19
<b>Gambar 2.4</b>	Klasifikasi <i>Linearly Nonseparable</i> (Härdle dkk., 2014) .....	22
<b>Gambar 2.5</b>	Fungsi $\psi$ Memetakan Data ke Ruang Berdimensi Lebih Tinggi ..	24
<b>Gambar 2.6</b>	<i>Curva</i> Kernel Linier.....	26
<b>Gambar 2.7</b>	<i>Curva</i> Kernel <i>Polynomial</i> .....	27
<b>Gambar 2.8</b>	<i>Curva</i> Kernel <i>Gaussian Radial Basis Function</i> .....	27
<b>Gambar 2.9</b>	Kurva AUC-ROC .....	30
<b>Gambar 2.10.</b>	Ilustrasi Kejang Epilepsi.....	31
<b>Gambar 2.11</b>	Ilustrasi Perekaman EEG.....	33
<b>Gambar 2.12</b>	Standar Sistem Peletakan Elektroda 10-20 pada (A) Sisi Kiri Kepala dan (B) Tampak Atas Kepala. A = Ear lobe, C = central, Pg = nasopharyngeal, P = parietal, F = frontal, Fp = frontal polar, O = occipital .....	34
<b>Gambar 2.13</b>	Ilustrasi <i>Montase EEG</i> .....	34
<b>Gambar 2.14</b>	Ilustrasi impuls listrik abnormal saat kejang .....	36
<b>Gambar 2.15</b>	Contoh Macam-macam Gelombang.....	37
<b>Gambar 3.1</b>	Hasil Rekaman <i>EEG</i> dengan Montase <i>Cz</i> .....	39
<b>Gambar 3.2</b>	Pemotongan Sinyal <i>Ictal</i> dan <i>Ocular Artifacts</i> .....	40
<b>Gambar 3.3</b>	Langkah Analisis Secara Umum .....	43
<b>Gambar 3.4</b>	Display Pengaturan di Aplikasi <i>Nihon Kohden</i> .....	44
<b>Gambar 3.5</b>	Hasil <i>Export</i> Data dari Aplikasi <i>Nihon Kohden</i> .....	44
<b>Gambar 3.6</b>	Sinyal <i>Ictal</i> pada Perekaman <i>EEG</i> .....	46
<b>Gambar 3.7</b>	Sinyal <i>OA</i> pada Perekaman <i>EEG</i> .....	46
<b>Gambar 3.8</b>	<i>Empirical Mode Decomposition</i> .....	47
<b>Gambar 3.9</b>	Ekstraksi Fitur.....	48
<b>Gambar 3.10</b>	Diagram Alir Penelitian.....	52

<b>Gambar 4.1</b>	Jumlah Sinyal Hasil Pemotongan.....	54
<b>Gambar 4.2</b>	Pola Sinyal <i>Ictal</i> Saat Terjadi Kejang Epilepsi .....	54
<b>Gambar 4.3</b>	Pola Sinyal <i>Ocular Artifacts</i> yang Terekam <i>EEG</i> .....	55
<b>Gambar 4.4</b>	Titik <i>Extrema</i> Hasil <i>Cubic Spline Interpolation</i> dari Satu Potong Sinyal <i>Ictal</i> .....	56
<b>Gambar 4.5</b>	<i>IMF</i> dan Residual dari Proses <i>EMD</i> Potongan Pertama Sinyal <i>Ictal</i> .....	56
<b>Gambar 4.6</b>	Titik <i>Extrema</i> hasil <i>Cubic Spline Interpolation</i> dari Satu Potong Sinyal <i>OA</i> .....	57
<b>Gambar 4.7</b>	<i>IMF</i> dan Residual dari Proses <i>EMD</i> Potongan Pertama Sinyal <i>OA</i> .....	58
<b>Gambar 4.8</b>	<i>Boxplot</i> $B_{AM}$ dari <i>IMF1-IMF4</i> Dari Data potongan 2 detik (A) dan Dari Data Potongan 5 Detik (B).....	59
<b>Gambar 4.9</b>	<i>Boxpot</i> $B_{FM}$ Dari <i>IMF1-IMF4</i> Dari Data Potongan 2 detik (A) dan Dari Data Potongan 5 Detik (B).....	59
<b>Gambar 4.10</b>	<i>Boxplot</i> $fr$ dari <i>IMF1-IMF4</i> Dari Data Potongan 2 Detik (A) dan Dari Data Potongan 5 Detik (B).....	60
<b>Gambar 4.11</b>	<i>Boxplot</i> $fd$ Dari <i>IMF1-IMF4</i> Dari Data Potongan 2 Detik (A) dan Dari Data Potongan 5 Detik (B).....	61
<b>Gambar 4.12</b>	<i>Boxplot</i> $RMIFS$ dari <i>IMF1-IMF4</i> Dari Data Potongan 2 Detik (A) dan Dari Data Potongan 5 Detik (B).....	61

## DAFTAR LAMPIRAN

<b>Lampiran 1.</b> Sertifikat Sidang Etik.....	75
<b>Lampiran 2.</b> Data Hasil Ekstraksi $B_{AM}$ dan $B_{FM}$ untuk sinyal dengan potongan 2 detik .....	76
<b>Lampiran 3.</b> Data Hasil Ekstraksi $B_{AM}$ dan $B_{FM}$ untuk sinyal dengan potongan 5 detik .....	77
<b>Lampiran 4.</b> Data Hasil Ekstraksi Fitur Time Frequency untuk sinyal dengan potongan 2 detik .....	78
<b>Lampiran 5.</b> Data Hasil Ekstraksi Fitur Time Frequency untuk sinyal dengan potongan 5 detik .....	79
<b>Lampiran 6.</b> Script Potong Data Sinyal <i>Ictal</i> dengan durasi 2 detik .....	80
<b>Lampiran 7.</b> Script Potong Data Sinyal <i>Ocular artifact</i> dengan durasi 2 detik ..	81
<b>Lampiran 8.</b> Script Potong Data Sinyal <i>Ictal</i> dengan durasi 5 detik .....	82
<b>Lampiran 9.</b> Script Potong Data Sinyal <i>Ocular artifact</i> dengan durasi 5 detik ..	83
<b>Lampiran 10.</b> Script Ekstraksi Fitur $B_{AM}$ dan $B_{FM}$ untuk data pemotongan 2 detik .....	84
<b>Lampiran 11.</b> Script Ekstraksi Fitur $B_{AM}$ dan $B_{FM}$ untuk data pemotongan 5 detik .....	88
<b>Lampiran 12.</b> Script Ekstraksi Fitur <i>time frequency</i> untuk data pemotongan 2 detik .....	91
<b>Lampiran 13.</b> Script Ekstraksi Fitur <i>time frequency</i> untuk data pemotongan 5 detik .....	96
<b>Lampiran 14.</b> Script <i>Tunning</i> Parameter <i>SVM</i> .....	104
<b>Lampiran 15.</b> Script Klasifikasi Sinyal <i>Ictal</i> dan Sinyal OA menggunakan <i>SVM</i> .....	105
<b>Lampiran 16.</b> Hasil tuning parameter dan Keباikan Klasifikasi dengan <i>input</i> fitur $B_{AM}$ dan $B_{FM}$ untuk data potongan 2 detik .....	108
<b>Lampiran 17.</b> Hasil tuning parameter dan Keباikan Klasifikasi dengan <i>input</i> fitur $B_{AM}$ dan $B_{FM}$ untuk data potongan 5 detik .....	110

<b>Lampiran 18.</b> Hasil tuning parameter dan Kebaikan Klasifikasi dengan <i>input</i> fitur <i>Time Frequency</i> untuk data potongan 2 detik .....	112
<b>Lampiran 19.</b> Hasil tuning parameter dan Kebaikan Klasifikasi dengan <i>input</i> fitur <i>Time Frequency</i> untuk data potongan 5 detik.....	115
<b>Lampiran 20.</b> Salah satu contoh catatan dari dokter.....	118

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Epilepsi adalah gangguan fungsi otak dengan gejala tunggal yang khas, yaitu kejang berulang akibat lepasnya muatan listrik *neuron* otak secara berlebihan dan peroksimal (Fisher dkk., 2005). Jumlah penderita epilepsi di seluruh dunia diperkirakan mencapai 50 juta jiwa dan menyerang disemua kalangan usia (World Health Organization, 2018), sehingga menjadikan penyakit ini sebagai penyakit saraf yang paling populer di dunia (Supriya dkk., 2016). Sekitar 80% penderita epilepsi hidup di negara miskin dan berkembang, dan sekitar tiga per empatnya tidak mendapatkan penanganan yang benar (World Health Organization, 2018). Salahnya penanganan terhadap penderita epilepsi tidak lepas dari anggapan masyarakat yang mengkaitkan epilepsi sebagai hal mistis, bukan sebagai penyakit atau gangguan pada saraf otak (Kandel dkk., 2000). Di Indonesia, epilepsi lebih dikenal dengan istilah “ayan”. Pada tahun 2018, WHO memperkirakan 1,85-3,71 juta penduduk Indonesia menderita penyakit epilepsi dan meningkat sekitar 80-130 ribu penderita setiap tahunnya. Gejala umum yang muncul pada penderita epilepsi adalah kejang atau lebih dikenal dengan kejang epilepsi (Fisher dkk., 2014).

*Ictal* atau kejang epilepsi merupakan ciri khas dari penyakit epilepsi yang disebabkan adanya aktivitas berlebihan pada sistem saraf manusia, terutama pada sistem saraf pusat (otak dan sumsum tulang belakang) (Fisher dkk., 2005). Fungsi sel saraf manusia dibagi menjadi 3 yaitu saraf *sensory*, saraf *intermediates*, dan saraf *motoric* (Kandel dkk., 2000) yang terdiri atas tiga komponen utama, yaitu dendrit, axon, dan badan sel. Dendrit berfungsi sebagai penerima pesan dari sel saraf lain ataupun rangsangan yang diterima oleh tubuh dari luar. Pesan yang diterima oleh dendrit akan diteruskan ke badan sel yang selanjutnya diteruskan melalui *axon* menuju percabangan axon (*axon terminal*).

Proses transfer sinyal atau pesan antar sel saraf terjadi pada ujung *axon terminal* dan ujung dendrit. Sel saraf mengirim impuls dari sistem saraf pusat ke otot atau kelenjar yang menghasilkan respon tubuh terhadap rangsangan. Untuk

meneruskan impuls pada jaringan saraf, zat kimia yang disebut *neurotransmitter* harus dilepaskan. Pesan berjalan dari celah-celah kecil di setiap saraf yang disebut celah sinaptik. *Neurotransmitter* yang dilepas dan yang diterima harus dapat menyeimbangkan impuls, agar impuls dapat melakukan perjalanan dari masing-masing sel saraf. Kejang akan terjadi apabila saat menghambat dan menginduksi *neurotransmitter* tidak seimbang.

Sel saraf mengirimkan dan menangkap sinyal yang salah dapat memicu aktivitas listrik yang tidak terkendali di otak. Gerakan acak tubuh manusia (kejang) akan terjadi ketika saraf menerima impuls/pesan campuran dan mencoba untuk melaksanakan semua pada saat yang sama. Kejang yang terjadi pada penderita epilepsi dikenal dengan kejang epilepsi, yang juga merupakan ciri khas dari penderita epilepsi (Fisher dkk., 2005). Penderita epilepsi pasti pernah mengalami kejang, namun orang yang mengalami kejang belum tentu merupakan penderita epilepsi. Perbedaan umumnya adalah pada penderita epilepsi kejang dapat terjadi sewaktu-waktu tanpa harus ada kejadian atau penyebab yang menyertai (Kandel dkk., 2000). Sedangkan kejang yang bukan karena epilepsi terjadi karena ada pemicunya seperti demam, dan benturan kepala yang menyebabkan kondisi otak tidak stabil (Siuly dkk., 2016). Ahli syaraf atau *neurologist* menggunakan *electroencephalogram (EEG)* untuk merekam aktivitas otak berupa sinyal (Malmivuo dan Plonsey, 1995).

Sinyal *EEG* merupakan rekaman aktivitas listrik berupa fluktuasi tegangan yang dihasilkan oleh arus ion di dalam neuron otak (Schomer dan Silva, 2011). Aktivitas listrik tersebut ditangkap oleh elektroda yang dipasangkan pada kulit kepala sesuai dengan standar *International Federation of Societies for Electroencephalography and Clinical Neurophysiology* (Klem dkk., 1999), atau dikenal dengan istilah *10-20 Electrode Placement International Standard* (Schomer dan Silva, 2011). Rekaman sinyal *EEG* didapatkan berdasarkan perbandingan antara dua tegangan elektroda yang berdekatan atau terhadap rata-ratanya maupun suatu titik tertentu, yang dikenal dengan istilah *montase* (Siuly dkk., 2016). Sinyal-sinyal yang terekam mencirikan kondisi pasien atau orang yang diamati, menjadikan *EEG* sebagai komponen penting untuk diagnosa dan analisis epilepsi (Alotaiby dkk., 2014).



Terdapat dua kategori aktivitas abnormal yang dapat diamati dari sinyal *EEG*, yaitu kondisi saat terjadinya kejang epilepsi (*ictal*) dan kondisi antar kejang (*interictal*) (Siuly dkk., 2016). Seringkali timbulnya kejang klinis ditandai oleh perubahan frekuensi yang tiba-tiba dalam perekaman *EEG*. Pada umumnya di dalam sinyal *alpha* frekuensinya akan menurun, tetapi meningkatkan amplitudo selama periode kejang (Siuly dkk., 2016). *EEG* membantu menentukan jenis kejang dan sindrom pasien epilepsi, sehingga dapat membantu menentukan pilihan obat anti epilepsi dan prediksi prognosis. Kontribusi lainnya dari informasi di dalam *EEG* adalah menentukan diagnosa *multi-axial* apilepsi, dalam hal ini apakah gangguan kejang bersifat *focal* atau *generalized*, *idiopahic* atau *symptomatic* (Smith, 2005). Namun, rekaman *EEG* juga mengandung sinyal *artifacts* yang secara umum disebabkan oleh pasien (gerakan mata, otot, serta jantung) dan mesin seperti lepasnya elektroda. Sinyal *artifacts* cenderung memiliki amplitudo yang besar dan sering disalah artikan sebagai sinyal *ictal*.

*Artifact* merupakan sinyal pengganggu yang berdasarkan asalnya dapat dibagi menjadi dua, yaitu *biological artifacts* dan *external artifacts*. *Biological artifact* disebabkan oleh aktivitas mata *electro-oculogram* (EOG) atau juga disebut *ocular artifacts* (*OA*), aktivitas otot *electromyogram* (EMG), serta aktivitas jantung *electrocardiogram* (ECG). Asal munculnya *artifacts* kedua yaitu *external artifacts* yang disebabkan oleh faktor-faktor teknis seperti lepasnya kabel atau elektroda saat perekaman sinyal *EEG* (Jafarifarmand dan Badamchizadeh, 2013).

*OA* merupakan jenis *artifacts* yang paling sering muncul pada perekaman *EEG* (Singh B dan Wagatsuma H, 2018). *Artifacts* secara langsung dapat mempengaruhi hasil identifikasi dalam melakukan diagnosa terhadap pasien. Apabila tidak ditangani, maka akan memperbesar resiko salah identifikasi. Karena jenis *artifacts* yang paling sering muncul akibat dari aktifitas pergerakan mata, maka penelitian ini bertujuan untuk membedakan sinyal *ictal* dan *OA* dengan cara klasifikasi.

*Support Vector Machine* (*SVM*) merupakan salah satu metode klasifikasi yang awalnya diusulkan oleh Vapnik pada tahun 1995 (Vapnik dan Cortes, 1995). *SVM* telah diaplikasikan secara sukses dalam banyak masalah klasifikasi dan regresi di berbagai bidang. Prinsipnya yaitu menemukan sekumpulan pemisah

optimal (*hyperplane*) dari data klasifikasi yang dilatih dengan suatu algoritma sehingga dapat memisahkan dataset menjadi dua atau lebih kelas yang berbeda yang dapat membantu memprediksi kategori dari data baru (Huang dkk., 2014; Purnami dkk., 2009). Keuntungan menggunakan *SVM* antara lain memiliki *background* matematika yang solid, kemampuan generalisasi yang tinggi dan kemampuan dalam menemukan solusi klasifikasi yang global dan tidak linier (Batuwata dan Paladeva, 2013). Klasifikasi biner menggunakan *SVM* pernah dilakukan oleh Purnami dkk (2009) untuk diagnosa penyakit diabetes, Mahaptra dan Horio (2018) untuk klasifikasi sinyal *ictal* dan *interictal*, bahkan Supriya, Siuly dan Zhang (2016) menggunakan *SVM* kernel *polynomial* untuk sinyal *ictal* dan *interictal* dengan akurasi tinggi. Akan tetapi, sinyal *EEG* hasil perekaman adalah data sinyal otak yang tidak dapat secara langsung digunakan untuk klasifikasi

Rekaman *EEG* mengandung data dengan jumlah yang besar. Hal utama dan yang paling penting adalah bagaimana merepresentasikan sinyal *EEG* hasil rekaman untuk analisis lebih lanjut, seperti klasifikasi. Ekstraksi sinyal mentah menjadi fitur-fitur yang penting diperlukan sebagai bahan untuk klasifikasi (Siuly dkk., 2016). *Fitur-fitur* dalam klasifikasi sinyal *EEG* dapat diperoleh dari proses dekomposisi, karena sinyal yang dijadikan *input* merupakan satu band/pita gelombang (Huang dkk., 1998).

*Empirical Mode Decomposition (EMD)* merupakan salah satu metode dekomposisi yang dapat diterapkan pada sinyal *EEG* (Fu dkk., 2014; Mahaptra dan Horio, 2018; Bajaj dan Pachori, 2012). *EMD* mendekomposisi sinyal *EEG* menjadi beberapa *intrinsic mode functions (IMF)* dan satu residual yang konstan (Mahaptra dan Horio, 2018). Penerapan algoritma *EMD* dapat dikonvolusi dengan beberapa transformasi sebagai *conjugatonya*. Mahaptra dan Horio (2018) menggunakan *EMD* dengan konvolusi *Hilbert transform*, sedangkan Fu dkk (2014) menggunakan *Hilbert-Huang transform* dan berhasil mendapatkan *IMF* dan residual yang konstan. Dengan demikian, *EMD* dengan *Hilbert transform* sebagai konvolusinya dapat pula digunakan dalam penelitian ini untuk mendapatkan *IMF* dari sinyal yang dijadikan *input*. *IMF-IMF* yang diperoleh dari proses dekomposisi selanjutnya di ekstrak menjadi fitur-fitur untuk keperluan klasifikasi (Mahaptra dan Horio, 2018).

Ekstraksi fitur terdiri atas beberapa macam cara dan metode. Pada klasifikasi sinyal *ictal* dan *interictal EEG* data *Bonn* dan menggunakan metode *SVM* sebagai metode klasifikasinya, dapat menghasilkan hasil yang berbeda-beda akibat dari penggunaan *fitur* yang berbeda. Pada kasus yang sama, penggunaan 6 *temporal* dan *spectral features* menghasilkan *accuracy* klasifikasi sebesar 93% (Riaz dkk., 2016), penggunaan fitur *Bandwidth amplitude* dan *Bandwidth frequency* menghasilkan akurasi sebesar 96,3% (Putri, 2018), penggunaan fitur *coefficient of variation* dan *fluctuation index* menghasilkan akurasi 97,4% (Li dkk., 2013), penggunaan fitur *dominant frequency*, *RMS frequency*, *RMIFS frequency* yang merupakan *time frequency features* masing-masing menghasilkan akurasi sebesar 99,88%, 99,91%, dan 98,9% (Mahapatra dan Horio, 2018).

*Time frequency features* pada penjelasan sebelumnya dapat menghasilkan akurasi yang lebih besar daripada fitur-fitur yang lain. Karena fitur merupakan penghubung dari masing-masing metode, ekstraksi fitur menjadi salah satu hal penting dalam proses klasifikasi yang dapat mempengaruhi keakuratan klasifikasi. Di sisi lain sinyal *interictal* secara umum memiliki pola yang hampir sama dengan dengan *OA*, terutama *OA* yang diakibatkan oleh kedipan yang berlebihan. Untuk menghasilkan klasifikasi sinyal *ictal* dan *OA* yang terbaik, maka dalam penelitian ini fitur yang digunakan adalah *dominant frequency*, *RMS frequency*, *RMIFS frequency* dan parameternya.

## 1.2 Rumusan Masalah

Hasil rekaman sinyal *EEG* masih mengandung *noise* atau dalam dunia medis lebih dikenal dengan istilah “*artifacts*” yang dihasilkan saat perekaman. Jenis *artifacts* yang paling sering muncul adalah *artifacts* yang disebabkan oleh pergerakan mata atau *ocular artifacts (OA)*. Munculnya *OA* dapat meningkatkan resiko kesalahan diagnosa pasien epilepsi, karena secara visual pada kondisi tertentu sinyal *ictal* memiliki pola yang hampir sama dengan *OA*. Sehingga diperlukan cara yang dapat membedakan keduanya dengan akurasi yang tinggi, dan secara objektif.

Klasifikasi menggunakan *SVM* dapat dilakukan untuk membedakan sinyal *ictal* dan *OA*. Namun rekaman *EEG* adalah beberapa sub-band sinyal. Hal utama

dan yang paling penting adalah bagaimana merepresentasikan sinyal *EEG* hasil rekaman untuk analisis lebih lanjut, seperti klasifikasi. Ekstraksi sinyal mentah menjadi fitur-fitur yang penting diperlukan sebagai bahan untuk klasifikasi. Berdasarkan uraian pada sub-bab latar belakang, sinyal *EEG* dapat didekomposisi menggunakan *EMD* hingga didapat *IMF*. Setelah mendapatkan *IMF*, dapat dilakukan ekstraksi sinyal menjadi beberapa fitur.

*Time frequency features* dipilih sebagai fitur utama dalam penelitian ini, karena pada penelitian sebelumnya dapat menghasilkan nilai akurasi yang terbaik. Hal ini penting dilakukan mengingat kesalahan klasifikasi dapat menyebabkan kesalahan fatal dalam dunia medis. Resiko-resiko ini harus diminimalkan dengan berusaha meningkatkan akurasi klasifikasi.

Berdasarkan uraian tersebut, pada penelitian ini menggunakan sinyal *EEG* sebagai *input*, selanjutnya dipotong berdasarkan kategori sinyal (*ictal* dan *OA*). Hasil potongan tersebut kemudian didekomposisi menggunakan *EMD* dan diekstraksi hingga mendapatkan beberapa fitur yaitu *dominant frequency*, *RMS frequency*, *RMIFS frequency*. Setelah didapatkan fitur-fitur tersebut, tahapan selanjutnya adalah klasifikasi menggunakan *SVM* dengan beberapa kombinasi fitur yang dijelaskan pada Sub-sub-bab 3.3.2. Tahapan terakhir yaitu menentukan model klasifikasi terbaik berdasarkan nilai evaluasi kebaikan model. Secara ringkas, rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana dekomposisi sinyal menggunakan *EMD* dan ekstraksi sinyal menggunakan *Time frequency features* dan rasio parameternya?
2. Bagaimana hasil klasifikasi sinyal *ictal* dan *OA* menggunakan *SVM* berdasarkan hasil fitur ekstraksi?
3. Bagaimana model terbaik untuk klasifikasi sinyal *ictal* dan *OA*?

### **1.3 Tujuan**

Berdasarkan rumusan masalah yang telah dipaparkan, tujuan yang ingin dicapai dari penelitian ini adalah

1. Melakukan dekomposisi sinyal menggunakan *EMD* dan ekstraksi sinyal menggunakan *time frequency features* dan rasio parameternya

2. Mengklasifikasikan sinyal *ictal* dan *OA* menggunakan *SVM* berdasarkan hasil fitur ekstraksi
3. Mendapatkan model terbaik untuk klasifikasi sinyal *ictal* dan *OA*

#### **1.4 Manfaat**

Manfaat yang diharapkan dapat diperoleh dari penelitian ini berdasarkan tujuan penelitian adalah.

1. Sebagai alat bantu ahli neurologi dalam mengambil keputusan mengenai diagnosa epilepsi berdasarkan sinyal *EEG* sehingga dapat segera dilakukan tindakan preventif untuk pengobatan pasien serta membantu memudahkan kerja tenaga medis dalam mendiagnosa apakah pasien tersebut mengalami kejang epilepsi atau hanya *artifacts* psikologis
2. Memberikan informasi mengenai penerapan *time frequency features* pada kasus klasifikasi untuk meningkatkan performa klasifikasi sinyal *ictal* dan *OA*.

#### **1.5 Batasan Masalah**

Batasan masalah pada penelitian ini adalah klasifikasi dilakukan berdasarkan data hasil rekam sinyal *EEG* dari Rumah Sakit Universitas Airlangga Surabaya. Sinyal dipilih hanyalah sinyal *ictal* dan sinyal *ocular artifacts*. Metode klasifikasi menggunakan *SVM* dengan kernel linier, *RBF* dan *polynomial*

(Halaman ini sengaja dikosongkan)

## **BAB 2**

### **TINJAUAN PUSTAKA**

Bab ini terdiri atas tiga bagian utama, yaitu kajian jurnal pendukung, kajian metode statistika yang digunakan dan kajian non-statistika. Kajian jurnal pendukung berisi tentang jurnal-jurnal terdahulu yang relevan dengan metode dekomposisi sinyal, ekstraksi sinyal dan klasifikasi. Kajian Statistika berisi tentang algoritma dan metode-metode statistika yang digunakan untuk analisis data pada penelitian ini. Kajian non statistika membahas tentang pengertian dan ulasan kasus yang dibahas, yaitu beberapa definisi dan pemahaman mengenai penyakit epilepsi serta alat pengukurannya.

#### **2.1 Kajian Jurnal Utama dan Jurnal Pendukung**

Sub-bab ini mengkaji dua jurnal utama dan delapan jurnal pendukung terkait metode dan analisis yang digunakan dalam penelitian ini. Secara umum jurnal utama merupakan jurnal yang membahas penggunaan metode dan penerapan metode untuk menyelesaikan masalah yang akan diselesaikan.

Jurnal utama yang pertama (Mahapatra dan Horio, 2018) membahas mengenai metode ekstraksi fitur dengan pendekatan *time frequency*. Data yang digunakan berasal dari data publik berupa sinyal *ictal* dan *interictal*. Langkah analisis dalam jurnal ini diawali dengan dengan dekomposisi menggunakan algoritma *EMD* yang diconvolusi dengan *Hilbert transform*. Hasil dekomposisi berupa *IMF* dan residual yang merupakan pembentuk sinyal asli. Fitur-fitur yang digunakan adalah *RMS frequency*, *dominant frequency*, *root mean instantaneous frequency square*. Setelah mendapatkan fitur-fitur tersebut, selanjutnya dilakukan klasifikasi menggunakan metode *SVM*. Akurasi klasifikasi yang dihasilkan sebesar 99,88% pada klasifikasi yang terbaik. Langkah-langkah di jurnal ini dijadikan sebagai rujukan utama dalam penulisan penelitian ini.

Jurnal lain digunakan untuk mendukung analisis, baik dari segi metode analisis maupun konsep teori kasus yang dibahas, dalam hal ini yang dijadikan topik utama adalah kasus klasifikasi sinyal epilepsi.

Jurnal pendukung yang pertama (Bajaj dan Pachori, 2012) membahas mengenai klasifikasi sinyal *EEG* dengan dekomposisi *EMD*, hingga menghasilkan *IMF* yang dianggap sebagai satu set sinyal modulasi amplitudo dan frekuensi (AM-FM). Sama halnya dengan jurnal utama, *IMF* juga diconvolusi dengan *Hilbert transform*. Fitur yang dihasilkan adalah *Bandwidth Amplitude* ( $B_{AM}$ ) dan *Bandwidth modulated frequency* ( $B_{FM}$ ). Metode LS-SVM digunakan sebagai alat untuk mengklasifikasikan sinyal kejang dan non-kejang. Hasil akurasi yang diperoleh antara 97.82% - 98.51%.

Jurnal pendukung yang ke-dua (Loughlin dan Tacer, 1997) membahas interpretasi dan konsep dari *instantaneous frequency*, yang merupakan salah satu dari fitur untuk klasifikasi pada thesis ini. Secara umum jurnal ini menjelaskan cara mendapatkan nilai *instantaneous frequency* dan menunjukkan kelebihan dan kekurangannya, terutama dari sisi konsep dasarnya.

Jurnal pendukung ke-tiga (Huang dkk., 1998) membahas tentang metode *preprocessing* untuk data non-stasioner, proses *shifting* menggunakan *EMD* dan konsep dasar, validasi dan kalibrasi *Hilbert transform*. Jurnal ini dapat digunakan sebagai pendukung dalam membentuk *IMF* dan juga dapat membantu sebagai referensi saat pembentukan fitur-fitur untuk klasifikasi.

Jurnal pendukung ke-empat (Mutlu, 2018) membahas cara mengekstrak satu sinyal tunggal menggunakan *Hilbert Vibration Decomposition*, sehingga dapat memisahkan antara sinyal *EEG* normal dengan sinyal *EEG* pasien epilepsi. Dalam hal ini yang dimaksud dengan *EEG* pasien epilepsi adalah saat dalam kondisi kejang. Metode ini dapat dijadikan sebagai rujukan untuk merancang algoritma *EMD* dengan convolusi *Hilbert transform* dan membentuk fitur-fitur berbasis *time frequency* dari satu sinyal tunggal yang tidak stasioner.

Jurnal pendukung ke-lima (Singh B dan Wagatsuma H, 2018) membahas mengenai konsep dasar *OA* dengan tipe kedipan maupun lirikan. Jurnal ini menjelaskan karakteristik khas dari *OA* ditinjau dari amplitudo dan frekuensinya. Dengan demikian, pengetahuan dari jurnal ini dapat membantu saat melakukan pemotongan data dan proses dekomposisi.

Jurnal pendukung ke-enam (Supriya dkk., 2016) membahas mengenai metodologi yang dapat digunakan secara otomatis untuk mendeteksi penyakit



epilepsi, dengan cara mengelompokkan kondisi kejang dan tidak kejang dari sinyal *EEG*. Metode yang digunakan adalah *Linear Discriminant Analysis* (LDA) dan *SVM*. Hasil klasifikasi menunjukkan bahwa tingkat akurasi tertinggi diperoleh dari metode *SVM* dengan kernel *polynomial* yang mencapai akurasi 100%.

Jurnal pendukung ke-tujuh (Fu dkk., 2015) membahas mengenai klasifikasi untuk mendeteksi sinyal kejang pada sinyal *EEG* menggunakan *SVM*. Metode dekomposisi yang digunakan adalah *EMD*, dan *Hilbert marginal spectrum* berdasarkan entropi dan energi untuk ekstraksi fitur. Hasil akurasi klasifikasi dengan menggunakan *10-fold cross-validation* mencapai 99.85%.

Jurnal pendukung ke-delapan (Fu dkk., 2014) membahas tentang klasifikasi *seizure* bersarkan *time-frequency image* dari sinyal *EEG* menggunakan kombinasi antara *Hilbert-Huang transform* (HHT) dan *SVM*. HHT yang berdomain *time-frequency* diubah menjadi berdomain *time frequency image*. Fitur yang digunakan adalah rata-rata, varians, skewness dan kurtosis. *SVM* yang dapat digunakan untuk klasifikasi biner menghasilkan nilai akurasi yang tinggi dalam algoritma ini, yakni mencapai 99,125%, dengan kernel RBF.

## **2.2 Empirical Mode Decomposition**

*Empirical Mode Decomposition* (*EMD*) menganalisis sinyal sepenuhnya tergantung pada data, tanpa memaksakan model yang telah ada sebelumnya (Huang dkk., 1998). Cara kerjanya adalah dengan menguraikan atau mendekomposisi satu atau sekumpulan sinyal (Mahapatra dan Horio, 2018), dalam hal ini sinyal *EEG*, menjadi beberapa bagian yang disebut *Intrinsic Mode Function* (*IMF*). Terdapat dua syarat agar *EMD* dapat menghasilkan beberapa *IMF* (Mahapatra dan Horio, 2018), yaitu:

1. Jumlah *extreme* dan jumlah *zero-crossing* harus sama jika terdapat selisih paling banyak berjumlah satu.
2. Pada titik manapun nilai rata-rata dari *envelope* yang didefinisikan oleh *local minima* adalah nol.

Jika syarat tersebut terpenuhi, maka *EMD* dapat dilakukan dengan mengikuti algoritma berikut ini:

1. Mendeteksi *extrema* (*maxima* dan *minima*) dari sinyal awal  $z(t)$ , dengan  $t$  merupakan waktu.
2. Menggunakan *cubic spline interpolation* untuk interpolasi *local maxima* guna mendapatkan *upper envelop*  $U_{max}(t)$
3. Menggunakan *cubic spline* lagi untuk interpolasi *local minima* guna mendapatkan *lower envelope*  $L_{min}(t)$
4. Menghitung rata-rata *envelope*  $Avg(t) = \frac{1}{2} (U_{max}(t) + L_{min}(t))$ .
5. Mengurangi sinyal asli dengan rata-rata *envelop*,  $h(t) = z(t) - Avg(t)$
6. Menyatakan  $h(t)$  sebagai *IMF* ketika memenuhi dua kondisi dan tetapkan  $h(t)$  menjadi  $c(t)$
7. Menghitung  $r(t) = z(t) - c(t)$ . Jika  $r(t)$  tidak konstan, maka tetapkan  $z(t) = r(t)$  dan ulangi kembali dari langkah 1.

Keseluruhan proses tersebut berlaku secara rekursif yang mana disebut juga *sifting process* sampai *residu*  $r(t)$  monoton. Sehingga penjumlahan seluruh *IMF* hasil dekomposisi dan *residu* yang konstan dapat membentuk sinyal asli atau sinyal awal. Kondisi tersebut dapat dituliskan seperti pada Persamaan (2.1).

$$z(t) = \sum_{k=1}^K c_k(t) + r_K(t), \quad (2.1)$$

dengan  $K$  adalah jumlah *IMF* dan  $c_k(t)$  merupakan *IMF* ke- $i$ .  $r_K(t)$  adalah *residu* yang monoton pada iterasi terakhir. Namun pada penelitian ini dibatasi hanya empat (4) *IMF* yang dihasilkan.

### 2.3 Hilbert Transform terhadap IMF

Pada penelitian ini, *Hilbert transform* akan diterapkan pada algoritma *EMD* untuk menghasilkan *IMF*. *Hilbert transform* membawa nilai asli *IMF*  $c(t)$  ke domain *time frequency* dengan cara memproyeksikannya pada sumbu nyata domain yang kompleks, yang dapat dinyatakan dalam bentuk analitik  $s_t$  sebagai

$$s(t) = c(t) + jc_H(t) \quad (2.2)$$

di mana  $c(t)$  yang dikenai *Hilbert transform* dituliskan sebagai  $c_H(t)$  yang merupakan konvolusi dari  $c(t)$  dengan sinyal  $\frac{1}{\pi t}$  (Mahapatra dan Horio, 2018), atau dapat dituliskan  $c_H(t) = c(t) * \frac{1}{\pi t}$ . Dengan menggunakan konsep *Hermitian*

*Symmetry*, hanya bagian yang nyata yang digunakan, mengabaikan bagian imajiner yang merepresentasikan frekuensi yang negatif atau tidak diharapkan. Dengan demikian, maka Persamaan (2.2) dapat ditulis kembali menjadi Persamaan berikut (Huang dkk., 1998):

$$s(t) = a(t)e^{j\varphi(t)}. \quad (2.3)$$

*Instantaneous phase*  $\varphi(t)$  dan amplitudo  $a(t)$  dapat dituliskan masing-masing pada Persamaan (2.4) dan Persamaan (2.5)

$$\varphi(t) = \arctan \left[ \frac{c_H(t)}{c(t)} \right]. \quad (2.4)$$

$$a(t) = \sqrt{c^2(t) + c_H^2(t)} \quad (2.5)$$

Amplitudo  $a(t)$  dinormalisasi. *instantaneous frequency* didefinisikan sebagai turunan dari *instantaneous phase* (Loughlin dan Tacer, 1997), sehingga dapat dituliskan seperti pada Persamaan (2.6).

$$\omega(t) = \varphi'(t) \quad (2.6)$$

## 2.4 Ekstraksi Fitur

Ekstraksi fitur merupakan salah satu tahapan penting dalam sinyal *processing*. Tahapan ini memiliki pengaruh yang besar terhadap akurasi klasifikasi, yang artinya fitur yang baik mampu menunjukkan karakteristik yang membedakan ciri khas antar kelas dan mengarahkan pada hasil kinerja klasifikasi yang diharapkan. Setelah melakukan dekomposisi menggunakan *EMD*, maka dapat dibentuk fitur *time frequency* dan fitur *bandwidth*.

### 2.4.1 Root Mean Square Frequency, Dominant Frequency dan Root Mean Instantaneous Frequency Square

*Root Mean Square (RMS) frequency*  $f_r$  adalah akar kuadrat dari penjumlahan *square bandwidth* dan kuadrat dari *center frequency*. Hal tersebut dapat dibentuk dari *square bandwidth* pada persamaan berikut.

$$\sigma_\omega^2 = \int (\omega - \langle \omega \rangle)^2 |S(\omega)|^2 d\omega \quad (2.7)$$

$$B^2 = \langle \omega^2 \rangle_S - \langle \omega \rangle_S^2 \quad (2.8)$$

Pada penelitian ini, *square bandwidth*  $\sigma_\omega^2$  direpresentasikan sebagai  $B^2$  dan semua integral berada pada interval waktu  $[t_1, t_2]$ , yaitu  $[0,2]$  yang merupakan

waktu durasi segmen sinyal *EEG* disetiap satu potongan. Berdasarkan Persamaan (2.8), diperoleh rumus untuk mencari nilai *RMS frequency* seperti pada Persamaan (2.10).

$$\langle \omega^2 \rangle_S = B^2 + \langle \omega \rangle_S^2 \quad (2.9)$$

$$f_r = \sqrt{B^2 + \langle \omega \rangle_S^2}, \quad (2.10)$$

di mana  $\langle \omega^2 \rangle_S$  adalah *mean square frequency* dan  $\langle * \rangle_S$  mewakili untuk *spectral domain*.

$$\langle \omega^2 \rangle_S = \int \omega^2 |S(\omega)|^2 d\omega. \quad (2.11)$$

dan  $\langle \omega \rangle_S$  merepresentasikan *spectral* atau *center frequency* seperti pada Persamaan (2.12) (Cohen dan Lee, 1990).

$$\langle \omega \rangle_S = \int \omega |S(\omega)|^2 d\omega. \quad (2.12)$$

*Time Average*  $\langle * \rangle_T$  dari *instantaneous frequency*

$$\langle \omega \rangle_T = \int \varphi'(t) |s(t)|^2 dt. \quad (2.13)$$

$S(\omega)$  adalah *Fourier transform* dari sinyal  $s(t)$ .

$$S(\omega) = \frac{1}{\sqrt{2\pi}} \int e^{-j\omega t} s(t) dt, \quad (2.14)$$

dengan menggunakan teorema *parseval's*,  $\int |S(\omega)|^2 d\omega = \int |s(t)|^2 dt$  (Cohen, 1995). Total energi dari semua frekuensi yaitu  $\int |S(\omega)|^2 d\omega$  atau dapat dihitung dalam bentuk sinyal waktu menjadi  $\int |s(t)|^2 dt$  di *normalized* ke satu.

Persamaan (2.12) dapat dituliskan dalam bentuk lain dengan cara menghilangkan  $\omega$  dengan menggunakan *hermitian time frequency operator*  $\left(\frac{1}{j} \frac{d}{dt}\right)$  dan mengoperasikannya pada sinyal, lalu mengalikannya dengan *complex conjugate* sinyal  $s^*(t)$  dan diintegalkan. *Center frequency* dapat dituliskan sebagai berikut

$$\begin{aligned} \langle \omega \rangle_S &= \int \omega |S(\omega)|^2 d\omega \\ &= \int s^*(t) \frac{1}{j} \frac{d}{dt} s(t) dt \\ &= \int s^*(t) \left( \frac{1}{j} \frac{d}{dt} s(t) \right) dt, \end{aligned} \quad (2.15)$$

$$\begin{aligned}
&= \int s^*(t) \left( \left\{ \frac{1}{j} \frac{a'(t)}{a(t)} + \varphi'(t) \right\} s(t) \right) dt, \\
&= \int \left( \varphi'(t) + \frac{1}{j} \frac{a'(t)}{a(t)} \right) a^2(t) dt.
\end{aligned}$$

Persamaan (2.15) dapat disederhanakan dengan cara disubstitusikan dalam Persamaan (2.3). Bagian imajiner nya diabaikan karena akan bernilai nol dan  $a^2(t)$  adalah *density* dalam waktu (Cohen dan Lee, 1990). Sehingga dapat diperoleh *center frequency* sebagai berikut (Cohen, 1995):

$$\langle \omega \rangle_S = \int \varphi'(t) a^2(t) dt = \langle \omega \rangle_T. \quad (2.16)$$

Dengan membandingkan Persamaan (2.12) dan Persamaan (2.13) menunjukkan bahwa  $\langle \omega \rangle_S = \langle \omega \rangle_T$  (Cohen dan Lee, 1988). *Square bandwidth* dapat dinyatakan sebagai berikut (Cohen, 1995; Cohen dan Lee, 1990).

$$\langle \omega \rangle_S = \int \varphi'(t) a^2(t) dt = \langle \omega \rangle_T. \quad (2.17)$$

$$B^2 = \int (\omega - \langle \omega \rangle)^2 |S(\omega)|^2 d\omega,$$

$$B^2 = \int s^* t \left( \frac{1}{j} \frac{d}{dt} - \langle \omega \rangle \right)^2 s(t) dt,$$

$$B^2 = \int \left\{ \left( \frac{a'(t)}{a(t)} \right)^2 + (\varphi'(t) - \langle \omega \rangle)^2 \right\} a^2(t) dt,$$

$$B^2 = \int s^* t \left( \frac{1}{j} \frac{d}{dt} - \langle \omega \rangle \right)^2 s(t) dt,$$

$$B^2 = \int \left\{ \left( \frac{a'(t)}{a(t)} \right)^2 + (\varphi'(t) - \langle \omega \rangle)^2 \right\} a^2(t) dt,$$

$$B^2 = \int \sigma_\omega^2(t) a^2(t) dt + \sigma_T^2,$$

$$B^2 = \int a'^2(t) dt + \int (\varphi'(t) - \langle \omega \rangle)^2 a^2(t) dt,$$

$$B^2 = B_{AM}^2 + B_{FM}^2,$$

$$B = \sqrt{B_{AM}^2 + B_{FM}^2}.$$

dengan komponen *modulated (AM)* dan *frequency modulated (FM)* maka,

$$B_{AM}^2 = \int a'^2(t) dt \quad (2.18)$$

$$B_{FM}^2 = \int (\varphi'(t) - \langle \omega \rangle)^2 a^2(t) dt, \quad (2.19)$$

di mana  $a'(t)$  merupakan turunan atau perubahan amplitudo.  $B^2$  standar deviasi global atau *spectral square bandwidth*  $\sigma_\omega^2$ , yaitu penyebaran frekuensi yang ada selama total durasi pengamatan sinyal (Cohen dan Lee, 1990). Secara rumus yang digunakan untuk mencari nilai *time frequency features* sebagai berikut

1. *RMS Frequency*

$$f_r = \sqrt{B^2 + \langle \omega \rangle_S^2}, \quad (2.20)$$

dengan ratio parameter  $\Gamma_{MS} = \frac{B^2}{\langle \omega \rangle_T^2}$ , yang dapat didefinisikan juga dalam

$$\Gamma_{MS_{AM}} = \frac{B_{AM}^2}{\langle \omega \rangle_T^2}. \quad (2.21)$$

$$\Gamma_{MS_{FM}} = \frac{B_{FM}^2}{\langle \omega \rangle_T^2}. \quad (2.22)$$

2. *Dominant Frequency*

$$f_d = \sqrt{B_{AM}^2 + \langle \omega \rangle_T^2} \quad (2.23)$$

dengan ratio parameter  $\Delta_{DF}^2$  seperti pada persamaan berikut

$$\Delta_{DF}^2 = \frac{B_{FM}^2 + \langle \omega \rangle_T^2}{B_{AM}^2}. \quad (2.24)$$

3. *RMIFS (root mean instantaneous frequency square)*

$$f_R = \sqrt{B_{FM}^2 + \langle \omega \rangle_T^2} \quad (2.25)$$

dengan ratio parameter  $E_{MIFS}$  seperti pada persamaan berikut

$$E_{MIFS} = \frac{\langle \omega \rangle_T^2}{B_{FM}^2} \quad (2.26)$$

## 2.5 *Fold Cross Validation*

*Stratified k-fold cross validation* adalah metode pembagian data *training* dan *testing* dengan mempertimbangkan proporsi antar kelas. Metode ini bekerja dengan cara partisi data secara acak menjadi  $k$  bagian atau *folds* dengan jumlah sampel

sama yaitu  $D_1, D_2, \dots, D_k$  di mana masing-masing *fold* memiliki proporsi yang sama antar kelas data dengan validasi menggunakan *training* dan *testing* dilakukan sebanyak  $k$  kali. Pada iterasi ke- $i$ , partisi  $D_i$  akan diatur sebagai data *testing* dan partisi lainnya akan digunakan sebagai data *training*. Artinya, pada iterasi yang pertama, partisi  $D_2, D_3, \dots, D_k$  akan menjadi data *training* untuk mendapatkan model pertama yang akan diuji dengan partisi  $D_1$ . Pada iterasi kedua partisi  $D_1, D_3, \dots, D_k$  akan menjadi data *training* kemudian  $D_2$  akan menjadi data *testing*, begitu seterusnya sampai dengan  $D_k$  yang akan menjadi data *testing* (Han dkk., 2012). Ilustrasi pembagian data training dan testing pada *stratified k-fold cross validation* dengan  $k=5$  ditampilkan pada Gambar 2.1.

Original dataset (100%)		Model 1	Model 2	Model 3	Model 4	Model 5
	Fold 1	Testing (20%)	Training (20%)	Training (20%)	Training (20%)	Training (20%)
	Fold 2	Training (20%)	Testing (20%)	Training (20%)	Training (20%)	Training (20%)
	Fold 3	Training (20%)	Training (20%)	Testing (20%)	Training (20%)	Training (20%)
	Fold 4	Training (20%)	Training (20%)	Training (20%)	Testing (20%)	Training (20%)
	Fold 5	Training (20%)	Training (20%)	Training (20%)	Training (20%)	Testing (20%)

**Gambar 2.1** Ilustrasi 5-Folds Cross Validation

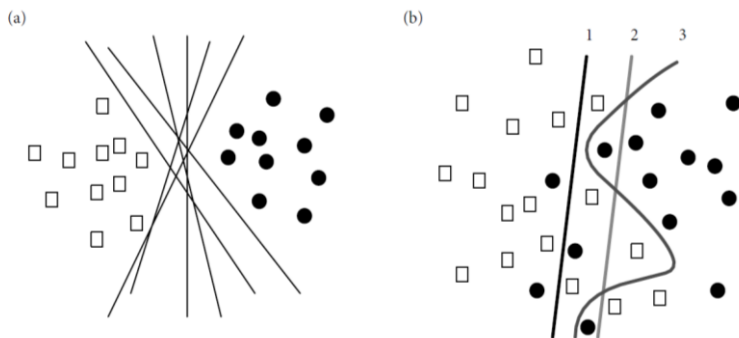
Data *training* digunakan untuk keperluan pembentukan model dan sebagai *input fitur* dalam analisis *SVM* yang akan dijelaskan pada Sub-bab 2.6. Sedangkan data *testing* digunakan untuk keperluan evaluasi performa klasifikasi yang dihasilkan oleh masing-masing model yang terbentuk.

## 2.6 Support Vector Machines

*Support Vector Machine (SVM)* adalah salah satu teknik dalam *machine learning* untuk mengatasi permasalahan klasifikasi biner (Vapnik dan Cortes, 1995; Purnami dkk., 2009). *SVM* tidak seperti metode statistika tradisional yang secara empiris mengurangi *error*, tetapi *SVM* bertujuan untuk meminimalkan batas *error* dengan memaksimalkan margin *hyperplane* untuk mengklasifikasikan suatu data (Purnami dkk., 2009). *Hyperplane* merupakan garis pembatas antar kelas sementara *margin* merupakan jarak terpendek dari *hyperplane* dengan kedua kelas (Han dkk., 2012). Terdapat dua langkah pada *SVM* yang digunakan untuk mendapatkan

*hyperplane* optimum. Langkah pertama adalah menemukan data-data terluar pada kedua kelas yang berada di perbatasan. Langkah kedua adalah menemukan *hyperplane* optimum dengan memperhitungkan data-data terluar tersebut sedangkan data-data lainnya tidak diperhitungkan (Suyanto, 2018). Dengan kata lain, *SVM* merupakan kombinasi dari garis pemisah (*hyperplane*), fungsi kernel, serta pendukung lain dalam ruang fitur berdimensi tinggi.

Klasifikasi menggunakan *SVM* pada awalnya adalah memisahkan obyek menjadi dua kelas, misal angka 1 mewakili kelas pertama dan angka -1 mewakili kelas ke-dua (Boser dkk., 1992). Tetapi seiring dengan perkembangan pada metode tersebut, klasifikasi berdasarkan *SVM* adalah mencari garis pemisah optimum yang dapat memisahkan dua kelas. Garis pemisah dikatakan optimum apabila dapat mengklasifikasikan tanpa terdapat error dan jarak antara vektor yang berdekatan adalah maksimal, di mana vektor terdekat dengan garis pemisah disebut *support vector* (Gunn, 1998). *SVM* menggunakan konsep klasifikasi linier (*linier classification*), di mana suatu objek tidak dapat dipisahkan dengan tepat, tidak terpisah dengan baik sehingga menimbulkan *missclassification* atau kesalahan dalam pengklasifikasian. Pada klasifikasi linier, *SVM* dibedakan menjadi dua, yaitu *linearly separable* dan *linearly nonseparable* dengan ilustrasi klasifikasi seperti pada Gambar 2.2.



**Gambar 2.2** Klasifikasi Linier *SVM* : (a) *Linearly Separable*, (b) *Linearly Nonseparable* (Härdle dkk., 2014)

### 2.6.1 *SVM Linearly Separable*

Setiap observasi terdiri atas  $p$  pasang prediktor  $\mathbf{x}'_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ ,  $\mathbf{x}'_i \in \mathcal{R}^p, i = 1, \dots, n$ , dimana  $n$  merupakan banyak data dan label kelas dari data  $\mathbf{x}_i$  yang termasuk kelas klasifikasi (label/target)  $y_i \in \mathbf{y} = \{-1, 1\}$ . Jika  $\mathbf{x}_i$  adalah anggota



dari kelas (1) maka  $\mathbf{x}_i$  diberi label  $y_i = 1$ , demikian pula sebaliknya, sehingga data yang diberikan akan berupa data pasangan atau juga disebut dengan himpunan data *training* dari kedua kelas yang akan diklasifikasikan dengan *SVM*, sehingga himpunan tersebut dapat dituliskan dalam Persamaan (2.27)

$$D_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\} \in \mathbf{X} \times \{-1, 1\}. \quad (2.27)$$

Pertama akan dijelaskan *SVM* pada kasus *linearly separable*. Sebuah konsep utama untuk menetapkan pemisah yang bersifat linier adalah *dot product* atau bisa disebut sebagai *inner product* atau *scalar product*, antara dua vektor yang ditetapkan sebagai  $\mathbf{x}'\mathbf{w} = \sum_{i=1}^n x_i w_i$ . Keluarga  $F$  dari fungsi klasifikasi yang terdapat pada ruang data diberikan sebagai berikut:

$$F = \mathbf{x}'\mathbf{w} + b, \mathbf{w} \in \mathcal{R}^p, b \in \mathcal{R}$$

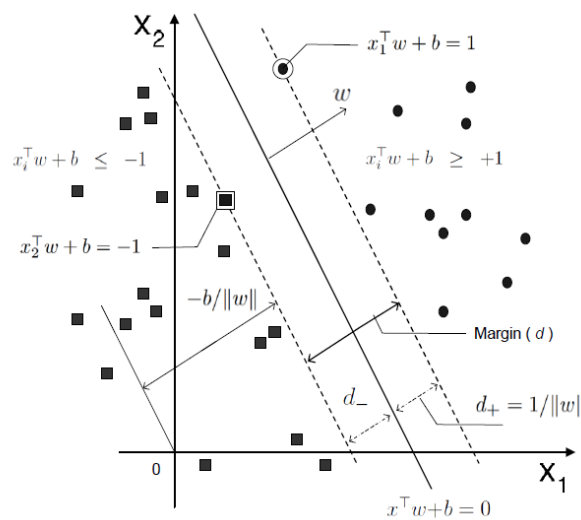
dimana  $\mathbf{w}$  merupakan vektor bobot yang tegak lurus terhadap *hyperplane* dan  $b$  merupakan posisi bidang relatif terhadap pusat koordinat. Bidang pemisah (*separating hyperplane*) dapat dituliskan sebagai berikut:

$$h(x) = \mathbf{x}'\mathbf{w} + b = 0. \quad (2.28)$$

$$\text{Bidang pembatas kelas 1: } \mathbf{x}'_i\mathbf{w} + b \geq 1 \text{ untuk } y_i = 1. \quad (2.29)$$

$$\text{Bidang pembatas kelas 2: } \mathbf{x}'_i\mathbf{w} + b \leq -1 \text{ untuk } y_i = -1. \quad (2.30)$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}; \mathbf{x}'_i = [x_{i1} \quad x_{i2} \quad \dots \quad x_{ip}]; \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}; \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$



**Gambar 2.3** Klasifikasi *Linearly Separable* (Härdle dkk., 2014)

Gambar 2.3 menunjukkan bahwa panjang vektor  $\mathbf{w}$  adalah  $\|\mathbf{w}\| = \sqrt{\mathbf{w}'\mathbf{w}}$ ,  $\|\mathbf{w}\| = \sqrt{w_1^2 + w_2^2 + \dots + w_p^2}$ . Bidang pembatas pertama membatasi kelas (1) sedangkan bidang pembatas kedua membatasi kelas (-1). Bidang pembatas pertama  $\mathbf{x}'_i\mathbf{w} + b = 1$  mempunyai bobot  $\mathbf{w}$  dan jarak tegak lurus dari titik asal sebesar  $\frac{|1-b|}{\|\mathbf{w}\|}$ , sedangkan bidang pembatas kedua  $\mathbf{x}'_i\mathbf{w} + b = -1$  mempunyai bobot  $\mathbf{w}$  dan jarak tegak lurus dari titik asal sebesar  $\frac{|-1-b|}{\|\mathbf{w}\|}$ . Jarak antara *margin* dengan bidang pemisah adalah  $d_+ = d_- = \frac{1}{\|\mathbf{w}\|}$ . Nilai maksimum *margin* (berdasarkan rumus jarak garis ke titik pusat) adalah  $\frac{1-b-(-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$  atau ekuivalen dengan  $\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}^2\|$ .

Menentukan *hyperplane* terbaik dapat menggunakan *Quadratic Programming* (QP) *problem* yaitu mencari titik minimal persamaan (2.31) dengan memperhatikan *constraint* persamaan (2.32)

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}^2\| \quad (2.31)$$

$$y_i(\mathbf{x}'_i\mathbf{w} + b) - 1 \geq 0, \forall i \quad (2.32)$$

Untuk mempermudah menyelesaikan persamaan (2.31) dan (2.32) dapat dipecahkan dengan berbagai teknik komputasi, salah satunya adalah *lagrange multiplier*. Berikut ini rumus *lagrange multiplier*.

$$L_P(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{x}'_i\mathbf{w} + b) - 1), \quad (2.33)$$

untuk  $i = 1, \dots, n$ .  $\alpha_i$  merupakan *Lagrange Multiplier*, yang bernilai nol atau positif  $\alpha_i \geq 0$ . Nilai optimal dari persamaan (2.33) dapat dihitung dengan meminimalkan  $L$  terhadap  $\mathbf{w}$  dan  $b$  serta memaksimalkan  $L$  terhadap  $\alpha_i$  sehingga diperoleh persamaan sebagai berikut:

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha} \left( \min_{\mathbf{w}, b} L_P(\mathbf{w}, b) \right). \quad (2.34)$$

Meminimalkan  $L$  terhadap  $\mathbf{w}$  dan  $b$  dapat diberikan sebagai berikut:

$$\begin{aligned} \frac{\partial L_P(\mathbf{w}, b)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \\ \Rightarrow \mathbf{w} &= \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i. \end{aligned} \quad (2.35)$$

$$\frac{\partial L_P(\mathbf{w}, b)}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0. \quad (2.36)$$

Persamaan (2.35) dan (2.36) disubstitusikan ke persamaan (2.34) sehingga diperoleh persamaan sebagai berikut:

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha} \left( \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}'_i \mathbf{x}_j \right). \quad (2.37)$$

*Constraint* yang digunakan untuk memaksimalkan  $\alpha_i$  pada persamaan (2.37) adalah sebagai berikut:

$$\sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0, \quad (2.38)$$

nilai  $\alpha_i$  dapat diperoleh dan nilai tersebut digunakan untuk mencari nilai  $\mathbf{w}$ . Jika nilai  $\alpha_i > 0$  atau sebuah titik data ke- $i$  untuk setiap  $y_i(\mathbf{x}'_i \mathbf{w} + b) = 1$ . Setelah menyelesaikan optimasi dengan *lagrange multiplier*, maka suatu pengamatan baru  $\mathbf{x}_{new}$  dapat diklasifikasikan menggunakan aturan klasifikasi sebagai berikut:

$$f(\mathbf{x}_{new}) = \text{sign}(\mathbf{x}'_{new} \hat{\mathbf{w}} + \hat{b}) \quad (2.39)$$

dimana  $\hat{\mathbf{w}} = \sum_{i=1}^n \hat{\alpha}_i y_i \mathbf{x}_i$  dan  $\hat{b} = \frac{1}{n_{sv}} \left( \sum_{i=1}^{n_{sv}} \frac{1}{y_i} - (\mathbf{x}'_{new} \hat{\mathbf{w}}) \right)$  dengan  $\mathbf{x}_i$  adalah *support vector*,  $\mathbf{x}_{new}$  adalah data yang diklasifikasikan dan  $n_{sv}$  adalah jumlah *support vector*.

### 2.6.2 SVM Linearly Nonseparable

Berdasarkan asumsi bahwa kedua buah kelas dapat terpisah secara sempurna oleh *hyperplane*. Namun, pada umumnya kedua buah kelas pada *input space* tidak dapat terpisah secara sempurna. Menurut Härdle, Prastyo, & Hafner (2014), kasus *linearly nonseparable* adalah mengklasifikasikan data linier yang tidak dapat dipisahkan sehingga *constraint* pada persamaan (2.29) dan (2.30) harus diubah secara linier dengan penambahan variabel *slack*  $\xi$  (*soft margin*) yang menunjukkan pelanggaran terhadap ketelitian atau kesalahan klasifikasi yang memungkinkan suatu titik berada di dalam marjin *error*  $0 \leq \xi_i \leq 1, \forall_i$  atau disebut misklasifikasi ( $\xi > 1$ ), sehingga  $\mathbf{x}_i$  diklasifikasikan menjadi

$$\mathbf{x}'_i \mathbf{w} + b \geq 1 - \xi, \text{ untuk } y_i = 1. \quad (2.40)$$

$$\mathbf{x}'_i \mathbf{w} + b \geq -(1 - \xi) \text{ untuk } y_i = -1. \quad (2.41)$$

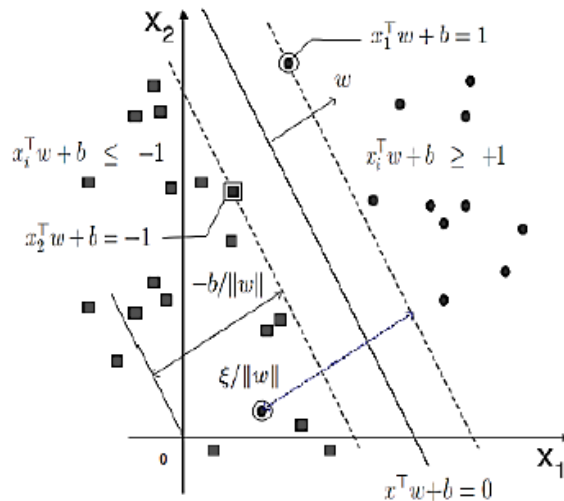
Ilustrasi klasifikasi *linearly nonseparable* dengan metode SVM dapat dilihat pada Gambar 2.4. Kedua *constraint* pada persamaan (2.40) dan (2.41) dapat digabung menjadi

$$y_i(\mathbf{x}'_i \mathbf{w} + b) \geq 1 - \xi_i, \forall_i, \xi_i > 0. \quad (2.42)$$

Formula pencarian bidang pemisah terbaik atau fungsi tujuan pada persamaan (2.31) diubah menjadi

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}^2\| + C \sum_{i=1}^n \xi_i, \quad (2.43)$$

dengan nilai  $\xi_i \geq 0$  dan parameter pinalti  $C > 0$  dimana  $C$  adalah parameter yang menentukan besar biaya akibat kesalahan klasifikasi (*misclassification*) dari data *training* selama proses pembelajaran dan nilai parameter ditentukan oleh pengguna. Ketika nilai  $C$  besar, maka margin akan menjadi lebih kecil, yang mengindikasikan bahwa tingkat toleransi kesalahan akan menjadi lebih kecil ketika suatu kesalahan terjadi. Sebaliknya, ketika nilai  $C$  kecil, tingkat toleransi kesalahan akan menjadi lebih besar (Huang dkk., 2014).



**Gambar 2.4** Klasifikasi *Linearly Nonseparable* (Härdle dkk., 2014)

Fungsi *lagrange multiplier* untuk kasus linearly nonseparable adalah

$$L_P(\mathbf{w}, b, \xi) = \frac{1}{2} \|\mathbf{w}^2\| + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i (\mathbf{x}'_i \mathbf{w} + b) - 1 + \xi_i) - \sum_{i=1}^n \mu_i \xi_i, \quad (2.44)$$

dimana  $\alpha_i \geq 0$  dan  $\mu_i \geq 0$  adalah *lagrange multiplier*,  $\mu_i \xi_i = 0$  dan  $\sum_{i=1}^n \alpha_i (y_i (\mathbf{x}'_i \mathbf{w} + b) - 1 + \xi_i) = 0$ . Nilai optimal dari persamaan (2.44) dapat dihitung dengan meminimalkan  $L$  terhadap  $\mathbf{w}$ ,  $b$  dan  $\xi_i$  serta memaksimalkan  $L$  terhadap  $\alpha_i$  sehingga diperoleh persamaan sebagai berikut:

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha} (\min_{\mathbf{w}, b, \xi} L(\mathbf{w}, b, \xi)) \quad (2.45)$$

Meminimalkan  $L$  terhadap  $\mathbf{w}$ ,  $b$  dan  $\xi_i$  dapat diberikan sebagai berikut:

$$\begin{aligned} \frac{\partial L_P(\mathbf{w}, b, \xi)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \\ \Rightarrow \mathbf{w} &= \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i. \end{aligned} \quad (2.46)$$

$$\frac{\partial L_P(\mathbf{w}, b, \xi)}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0. \quad (2.47)$$

$$\begin{aligned} \frac{\partial L_P(\mathbf{w}, b, \xi)}{\partial \xi_i} &= C - \alpha_i - \mu_i = 0 \\ \Rightarrow \alpha_i &= C - \mu_i. \end{aligned} \quad (2.48)$$

Persamaan (2.46), (2.47) dan (2.48) disubstitusikan ke persamaan (2.45) sehingga diperoleh persamaan sebagai berikut:

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha} (\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i' \mathbf{x}_j). \quad (2.49)$$

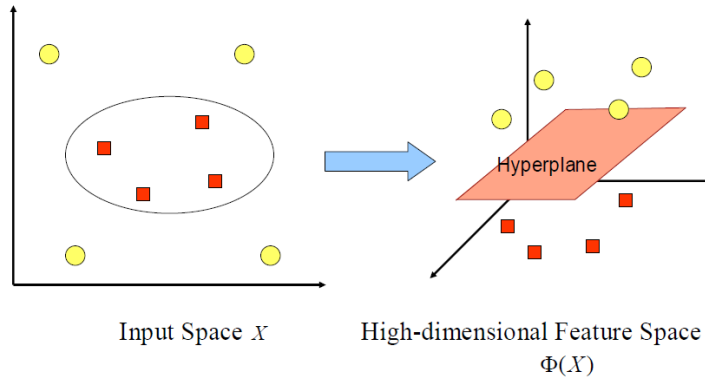
*Constraint* yang digunakan untuk memaksimalkan  $\alpha_i$  pada persamaan (2.49) adalah sebagai berikut:

$$\sum_{i=1}^n \alpha_i y_i = 0, 0 \leq \alpha_i \leq C. \quad (2.50)$$

Sampel  $\mathbf{x}_i$  untuk  $\alpha_i > 0$  (*support vector*) merupakan titik yang berada di atas marjin atau dalam marjin ketika *soft margin* digunakan. *Support vector* sering menyebar dan level penyebarannya berada pada batas atas (*upper bound*) untuk *misclassification rate* (Schölkopf dan Smola, 2001).

### 2.6.3 SVM Nonlinearly Separable

Pada umumnya masalah dalam dunia nyata jarang yang bersifat linier, kebanyakan bersifat *non linear*. Banyaknya data yang bersifat *nonlinear* dalam dunia nyata, maka *SVM* dimodifikasi dengan memasukkan fungsi Kernel yaitu fungsi *non linear*. Langkah pertama untuk *SVM nonlinearly separable* adalah data dengan struktur *nonlinear* dipetakan dengan menggunakan fungsi pemetaan  $\psi: \mathcal{R}^p \rightarrow \mathbf{H}$  ke dalam ruang berdimensi lebih tinggi  $\mathbf{H}$  dimana aturan klasifikasi bersifat linier. *SVM nonlinearly separable* dapat diilustrasikan pada Gambar 2.5.



**Gambar 2.5** Fungsi  $\psi$  Memetakan Data ke Ruang Berdimensi Lebih Tinggi

Gambar 2.5 kiri menunjukkan bahwa data pada kelas kuning dan data pada kelas merah tidak dapat dipisahkan secara linier jika menggunakan ruang berdimensi dua. Sedangkan Gambar 2.5 kanan menunjukkan bahwa ruang berdimensi 3 dapat dipisahkan secara linier oleh *hyperplane*. Menemukan titik-titik *support vector* dalam proses pembelajaran *SVM* bergantung pada *dot product* dari data yang sudah ditransformasikan ke ruang vektor baru yang berdimensi lebih tinggi  $\psi(\mathbf{x}_i)' \psi(\mathbf{x}_j)$ . Fungsi transformasi pada *SVM* adalah menggunakan *kernel trick* (Schölkopf dan Smola, 2001). Kegunaan *kernel trick* untuk menghitung *scalar product* melalui sebuah fungsi *kernel*. Proyeksi fungsi  $\psi: \mathcal{R}^p \rightarrow \mathbf{H}$  memastikan bahwa *inner product*  $\psi(\mathbf{x}_i)' \psi(\mathbf{x}_j)$  ditunjukkan oleh fungsi *kernel*.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \psi(\mathbf{x}_i)' \psi(\mathbf{x}_j). \quad (2.51)$$

Persamaan optimasi pada persamaan (2.49) dapat diubah menjadi,

$$\max_{\alpha} L_D(\alpha) = \max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j). \quad (2.52)$$

*Constraint* yang digunakan untuk memaksimalkan  $\alpha_i$  pada Persamaan (2.52) adalah sebagai berikut:

$$\sum_{i=1}^n \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, \dots, n. \quad (2.53)$$

Nilai dari fungsi klasifikasi (*score*) dapat dirumuskan sebagai berikut:

$$\begin{aligned} f(\mathbf{x}_i) &= \left( \sum_{j=1}^n \alpha_j y_j \left( \psi(\mathbf{x}_i)' \psi(\mathbf{x}_j) \right) + b \right) \\ &= \left( \sum_{j=1}^n \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right). \end{aligned} \quad (2.54)$$

Klasifikasi linier tidak cocok digunakan untuk beberapa permasalahan klasifikasi, sehingga perlu dilakukan transformasi terhadap data yang ada.

Transformasi *linearly separation* ke *nonlinearly separation* dilambangkan dengan simbol  $\Phi$ , melakukan transformasi dari  $n$ -dimensi pada vektor  $x$  ke  $m$  dimensi vektor  $z$  pada ruang *Euclidean*  $R^m$ , dapat di ilustrasikan seperti berikut ini.

$$\Phi: \begin{cases} R^n \rightarrow \mathcal{H} \\ \mathbf{x} = ([x]_1, \dots, [x]_n)^T \rightarrow \mathbf{z} = (z_1, z_2, \dots)^T = \Phi(\mathbf{x}) \end{cases} \quad (2.55)$$

dimana data *training* awal dan *training* setelah dilakukan transformasi diberikan seperti persamaan berikut.

$$\text{Training awal: } T = \{(x_1, y_1), \dots, (x_n, y_n)\} \quad (2.56)$$

$$\text{Training setelah transformasi: } T_\Phi = \{(z_1, y_1), \dots, (z_n, y_n)\}$$

dimana  $x_i \in R^n, y_i \in Y = \{+1, -1\}, i = 1, 2, \dots, n$ . Algoritma dari proses klasifikasi SVM untuk kasus *nonlinear separation* adalah sebagai berikut.

- 1) Masukkan data training dengan  $T = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , dimana  $x_i \in R^n, y_i \in Y = \{+1, -1\}, i = 1, 2, \dots, n$ .
- 2) Memilih sebuah *map* yang tepat  $\Phi: \mathbf{x} = \Phi(\mathbf{x})$  dari *space*  $R^n$  ke dalam *Hilbert space* dan parameter *penalty*  $C > 0$ .
- 3) Menyusun dan menyelesaikan permasalahan *convex quadratic programming* dengan menggunakan persamaan berikut.

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) - \sum_{j=1}^n \alpha_j \quad (2.57)$$

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (2.58)$$

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, n \quad (2.59)$$

dengan  $\alpha^* = (\alpha_1^*, \dots, \alpha_n^*)^T$

- 4) Menghitung nilai  $b^*$ : memilih komponen  $\alpha^*, \alpha_j^* \in (0, C)$

$$b^* = \sum_{i=1}^n y_i - \sum_{i=1}^n y_i \alpha_i^* (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)); \quad (2.60)$$

- 5) Menyusun fungsi keputusan dengan menggunakan persamaan berikut

$$f(\mathbf{x}) = \text{sign}(g(\mathbf{x})) \quad (2.61)$$

$$g(\mathbf{x}) = \sum_{i=1}^n y_i \alpha_i^* (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) + b^*. \quad (2.62)$$

Pada kasus nonlinear, membutuhkan sebuah kernel yang berbeda untuk membentuk suatu hyperplane. Fungsi kernel didefinisikan sebagai  $R^n \times R^n$  yang dipanggil dengan kernel  $R^n \times R^n$  seperti pada persamaan (2.55). Berikut adalah fungsi dasar kernel (Deng dkk., 2013).

$$K(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')) \quad (2.63)$$

dimana  $(\cdot)$  merupakan *inner product* dari *Hilbert space*, untuk setiap  $x_i, \dots, x_n \in R^n$ , berdasarkan *gram matrix*  $(K(\mathbf{x}_i, \mathbf{x}_j))_{n \times n}$  untuk  $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})f(\mathbf{x}')$ , dengan  $f(\cdot)$  fungsi real pada  $R^n$ . Setiap vektor  $\alpha = (\alpha_1, \dots, \alpha_l)^T \in R^n$ , dari uraian tersebut fungsi kernel yang didapat adalah sebagai berikut (Deng dkk., 2013).

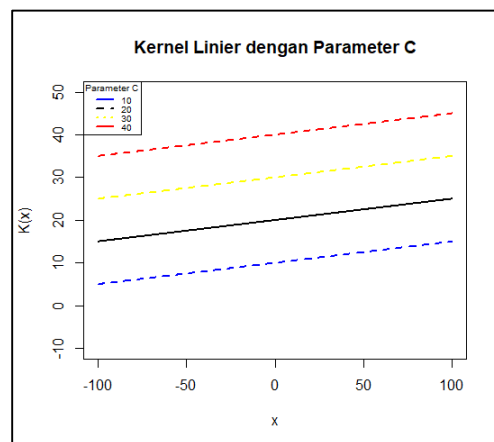
$$\begin{aligned} \alpha^T (K(\mathbf{x}, \mathbf{x}'))_{n \times n} \alpha &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j f(\mathbf{x}_i) f(\mathbf{x}_j) = \sum_{i=1}^n \alpha_i f(\mathbf{x}_i) \sum_{j=1}^n \alpha_j f(\mathbf{x}_j) \\ &= \left( \sum_{i=1}^n \alpha_i f(\mathbf{x}_i) \right)^2 \geq 0. \end{aligned} \quad (2.64)$$

Berikut adalah beberapa fungsi kernel yang sering digunakan (Deng dkk., 2013; Scholkopf, 2002).

1) Kernel *Linier*

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}'\mathbf{x} + C \quad (2.65)$$

dengan plot *density kernel linier* dapat ditunjukkan seperti pada **Gambar 2.6**.



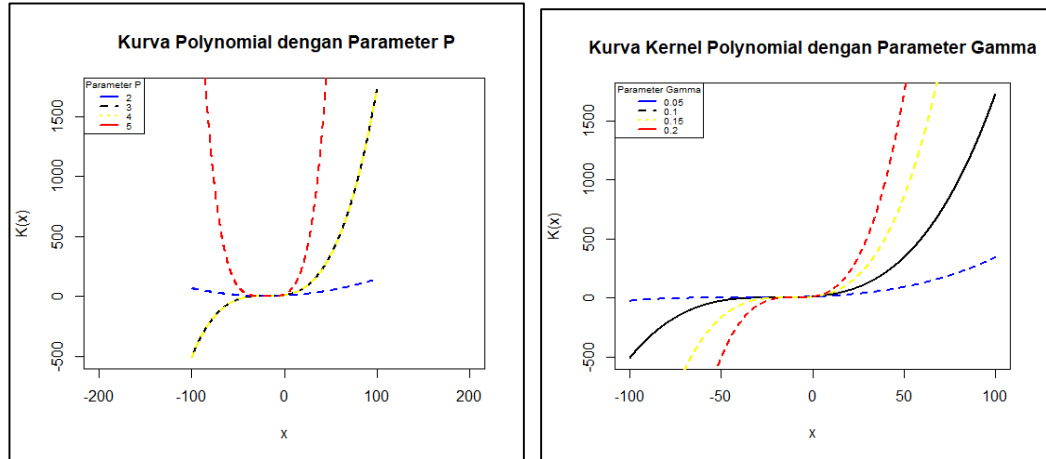
**Gambar 2.6** *Curva Kernel Linier*



2) Kernel *Polynomial*

$$K(\mathbf{x}, \mathbf{x}') = \gamma(\mathbf{x}_i^T \mathbf{x} + 1)^P \quad (2.66)$$

Plot *density kernel Polynomial* dapat ditunjukkan seperti pada **Gambar 2.7**

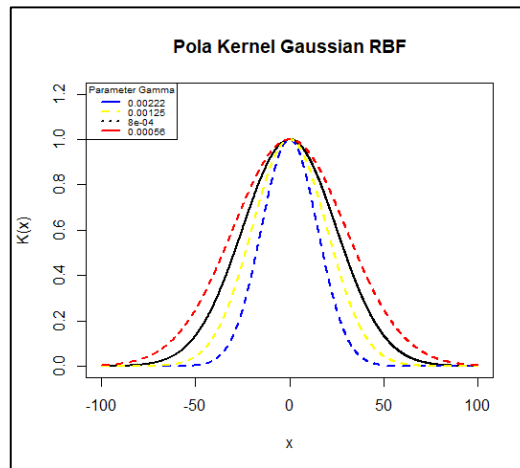


**Gambar 2.7** *Curva Kernel Polynomial*

3) Kernel *Gaussian Radial Basis Function*

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i'\|^2}{2\sigma}\right) \quad (2.67)$$

Plot *density Kernel Gaussian Radial Basis Function* dapat ditunjukkan seperti pada Gambar 2.8, dengan  $-\infty < \mathbf{x} < \infty$ .



**Gambar 2.8** *Curva Kernel Gaussian Radial Basis Function*

## 2.7 Evaluasi Performa Metode Klasifikasi

Performa metode klasifikasi dapat dievaluasi dengan alat ukur yang dapat menunjukkan kemampuan metode-metode dalam memprediksi kelas data atau respon secara tepat.

Ukuran ketepatan klasifikasi dapat dilihat dari besar nilai *accuracy*, *sensitifiy*, *specificity*, dan *Area Under the Receiver Operating Characteristics Curve (AUROC)* (Han dkk., 2012). Untuk mempermudah penghitungan nilai-nilai tersebut, maka terlebih dahulu dibuat *confusion matrix* seperti yang ditampilkan pada Tabel 2.1

**Tabel 2.1 Confusion Matrix**

	Prediksi (+)	Prediksi (-)	Jumlah
Aktual (+)	<i>TP (True Positives)</i>	<i>FN (False Negatives)</i>	TP+FP
Aktual (-)	<i>FP (False Positives)</i>	<i>TN (True Negatives)</i>	FN+TN
Jumlah	TP+FP	FN+TN	Total

Keterangan:

TP : jumlah prediksi benar pada kelas positif

TN : jumlah prediksi benar pada kelas negatif

FP : jumlah prediksi salah pada kelas positif

FN : jumlah prediksi salah pada kelas negatif

Pada penelitian ini, kelas positif untuk kategori *ictal* dan kelas negatif untuk kategori *OA*. Pengertian dan rumus untuk menghitung ukuran ketepatan klasifikasi adalah sebagai berikut (Han dkk., 2012):

### 2.7.1 Accuracy

*Accuracy* merupakan ukuran yang mengukur ketepatan hasil klasifikasi secara keseluruhan, semakin tinggi nilai akurasi, maka performansi metode klasifikasi yang digunakan semakin baik. Rumus untuk menghitung nilai *accuracy* dapat dilihat pada Persamaan (2.68)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.68)$$

### 2.7.2 Sensitifiy

*Sensitifiy* merupakan kelas positif yang benar dibandingkan total data yang diklasifikasikan sebagai kelas positif. Dalam hal ini, *sensitifiy* menyatakan jumlah sinyal *ictal* yang diklasifikasikan dengan benar, dibandingkan dengan jumlah sinyal *ictal* keseluruhan. Rumus untuk menghitung nilai *sensitifiy* dapat dilihat pada Persamaan (2.69).

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (2.69)$$

### 2.7.3 Specificity

*Specificity* merupakan kelas negatif yang benar dibandingkan total data yang diklasifikasikan sebagai kelas negatif. *Sensitivity* dalam penelitian ini menyatakan jumlah sinyal *OA* yang diklasifikasikan dengan benar, dibandingkan dengan jumlah sinyal *OA* keseluruhan. Rumus untuk menghitung nilai *sensitivity* dapat dilihat pada Persamaan (2.70).

$$Specificity = \frac{TN}{FN + TN} \times 100\% \quad (2.70)$$

### 2.7.4 AUROC

*AUROC* merupakan nama lain dari *AUC* (*Area Under the Curve*) – *ROC* (*Receiver Operating Characteristics*). Nilai *AUC* dapat diperoleh dengan cara menghitung luas daerah di bawah kurva *ROC*, sehingga nilai *AUC* berada dalam interval 0 hingga 1.

Kurva *ROC* adalah grafik yang digunakan untuk membandingkan performansi klasifikasi berdasarkan *sensitivity* pada sumbu vertikal dan pada sumbu horizontal adalah *1-specificity* atau proporsi data kelas negatif yang terklasifikasi dengan salah (Lakso dkk., 2005). *1-specificity* dalam penelitian ini adalah proporsi kategori sinyal *OA* yang terklasifikasi menjadi sinyal *ictal*. Kurva *ROC* dapat digunakan karena adanya ketidaksesuaian ukuran ketepatan klasifikasi antar kelompok, sehingga dapat mempersulit dalam menentukan metode mana yang lebih baik (Metz, 2006).

Kurva *ROC* mempunyai kemampuan evaluasi secara menyeluruh dan cukup baik, sehingga sering digunakan untuk evaluasi hasil klasifikasi (Cheng dkk., 2010). Kurva *ROC* yang baik berada di sebelah atas dari garis diagonal (0,0) dan (1,1) dan membuat nilai *AUC* kurang dari 0,5 (*random*). Semakin dekat kurva *ROC* ke sudut kiri atas mengindikasikan bahwa semakin tinggi *accuracy* dari klasifikasi yang dihasilkan. Rumus untuk menghitung nilai *AUC* dapat dilihat pada Persamaan (2.71).

$$AUC = \frac{1}{2} (sensitivity + spesificity) \quad (2.71)$$

Semakin besar nilai *AUC*, maka hasil klasifikasi dari metode yang digunakan semakin baik. Nilai *AUC* memiliki interval tertentu dalam mengkategorikan kebaikan hasil klasifikasi, seperti yang ditunjukkan pada Tabel 2.2.

**Tabel 2.2** Kategori Pengklasifikasian Model berdasarkan Nilai *AUC*

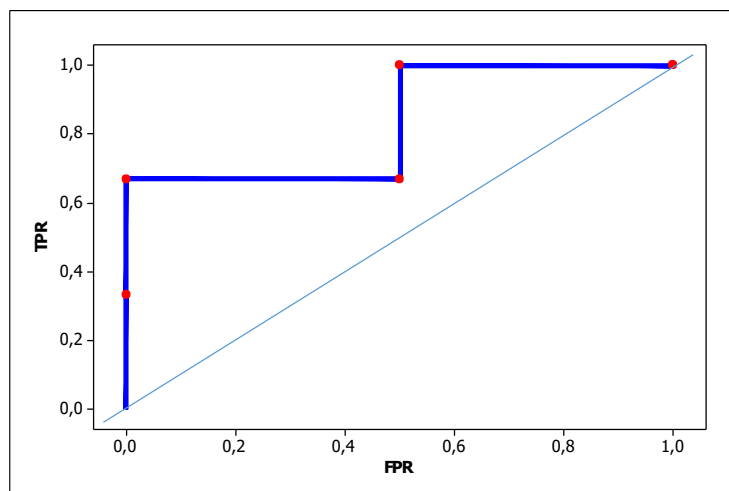
Nilai <i>AUC</i>	Model Diklasifikasikan Sebagai
0.90-1.00	<i>Excellent</i>
0.80-0.90	<i>Good</i>
0.70-0.80	<i>Fair</i>
0.60-0.70	<i>Poor</i>
0.50-0.60	<i>Fail</i>

Ilustrasi penghitungan nilai *AUC-ROC* menggunakan 5 pasang data dapat digunakan untuk mempermudah menghitung nilai *AUC* dan menggambarkan kurva *ROC*. Jika *threshold* antara kelas *ictal* (0) dan *OA* (1) ditentukan sebesar 0,5, maka  $\hat{Y}=0$  apabila  $p(y = 1|X) \leq 0,50$  dan  $\hat{Y}=1$  apabila  $p(y = 1|X) > 0,50$ . Berikut ini contoh penghitungan untuk mendapatkan *ROC* dan nilai *AUC*.

**Tabel 2.3** Data untuk Ilustrasi Membuat Kurva *ROC*

No.	Y	$\hat{Y}$	p	TP	FP	TN	FN	TPR	FPR
1	0	0	0,48	1	0	2	2	0,33	0
2	1	1	0,72	2	0	2	1	0,67	0
3	0	0	0,30	2	1	1	1	0,67	0,50
4	1	1	0,60	3	1	1	0	1	0,50
5	0	1	0,55	3	2	0	0	1	1

Berdasarkan perhitungan pada Tabel 2.3, dapat dibuat kurva *ROC* seperti pada Gambar 2.9. Garis lurus pembagi dari sumbu (0,0) hingga (1,1) mengindikasikan luas *AUC* 0,5, dan diharapkan nilai *AUC* dapat lebih besar dari 0,5.



**Gambar 2.9** Kurva *AUC-ROC*

## 2.8 Epilepsi

Epilepsi adalah gangguan fungsi otak dengan berbagai etiologi, dengan gejala tunggal yang khas, yaitu kejang berulang akibat lepasnya muatan listrik *neuron* otak secara berlebih dan peroksimal (Fisher dkk., 2005). Epilepsi dianggap sebagai penyakit otak yang didefinisikan oleh salah satu kondisi berikut (Fisher dkk., 2014): (1) Setidaknya terdapat dua kejang tidak beralasan (refleks) terjadi lebih dari 24 jam; (2) satu kejang tidak beralasan (refleks) dan kemungkinan kejang lebih lanjut mirip dengan resiko kekambuhan umum (setidaknya 60%) setelah dua kejang tak beralasan terjadi selama 10 tahun ke depan; (3) Diagnosis sindrom epilepsi. Epilepsi dianggap dapat diatasi untuk individu yang memiliki sindrom epilepsi *self-limited* yang tergantung usia tetapi yang sekarang melewati usia yang berlaku, atau bagi mereka yang tetap bebas kejang selama 10 tahun terakhir, tanpa obat kejang pada 5 tahun terakhir. Ilustrasi kejang epilepsi seperti pada Gambar 2.10.



**Gambar 2.10.** Ilustrasi Kejang Epilepsi

Terdapat dua kategori kejang epilepsi berdasarkan etiologinya, yaitu kejang fokal dan kejang umum dengan rincian pada Tabel 2.4.

**Tabel 2.4** Etiologi Epilepsi

<b>Kejang Focal</b>	<b>Kejang General</b>
a. Trauma kepala	a. Penyakit metabolik
b. Stroke	b. Reaksi obat
c. Infeksi	c. Idioptik
d. Malformasi vaskuler	d. Faktor Genetik
e. Tumor (Neoplasma)	e. Kejang fotosensitif
f. Displasia	
g. <i>Mesial Temporal Sclerosis</i>	

Epilepsi dapat terjadi pada semua kalangan usia dan sekitar 50 juta orang di seluruh dunia terkena epilepsi, sehingga menjadikannya sebagai salah satu penyakit *neurologis* yang paling umum di dunia (World Health Organization, 2018). Dalam kebanyakan kasus, penyebab epilepsi tidak diketahui, walaupun beberapa orang

menderita epilepsi sebagai akibat dari cedera otak, stroke, kanker otak dan penyalahgunaan obat dan alkohol (Fisher dkk., 2005). Epilepsi tidak dapat disembuhkan, tetapi serangan-serangannya bisa dikontrol dengan pengobatan pada sekitar 70% kasus (Eadie, 2012).

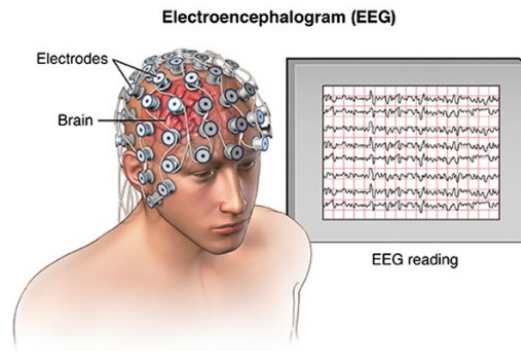
## **2.9 *Electroencephalogram (EEG)***

*EEG* merekam aktivitas elektrik di otak manusia. *EEG* mengukur fluktuasi tegangan yang dihasilkan oleh arus ion di dalam neuron otak (Schomer dan Silva, 2011). Dalam konteks klinis, *EEG* mengacu kepada perekam aktivitas elektrik spontan dari otak selama periode tertentu, biasanya 20-40 menit yang direkam dari elektroda yang dipasang di kulit kepala. *EEG* banyak digunakan oleh dokter dan peneliti yang mempelajari sistem kerja otak hingga diagnosa penyakit yang disebabkan kerusakan saraf. Penelitian tentang aktivitas kelistrikan di otak, menggunakan rekaman *EEG*, adalah salah satu alat yang paling penting untuk diagnosa penyakit syaraf seperti epilepsi, tumor otak, cedera otak, hingga pemantauan anastesi saat dilakukan operasi (Adeli dkk., 2003).

Mesin perekam *EEG* pertamakali diperkenal oleh Hans Beger pada tahun 1929. Dia adalah seorang *neuropsychiatrist* dari University of Jena di Jerman, yang memperkenalkan *EEG* dengan istilah “elektrenkephalogramm” berupa grafis dari arus listrik yang dihasilkan oleh otak. Beger mengungkapkan bahwa arus otak berubah-ubah tergantung aktivitas atau status fungsionalnya seperti tidur, epilepsi ataupun anastesi. Ide revolusionernya itu memunculkan cabang ilmu kedokteran baru yang disebut *neurophysiology* (Siuly dkk., 2016)

Saat dilakukan perekaman *EEG*, terdapat beberapa alat kecil yang disebut elektroda yang ditempatkan pada permukaan kepala seperti yang ditampilkan pada Gambar 2.11. Masing-masing elektroda dihubungkan dengan aplifier disetiap pasang elektroda dan sebuah mesin perekam sinyal. Sehingga aktivitas listrik didalam otak dapat diterima oleh elektroda yang pada akhirnya direkam dan dikonversikan menjadi sinyal *EEG* serta dapat dilihat berupa sinyal pada layar komputer. Berdasarkan sinyal-sinyal tersebutlah para *neuropsychiatrist* membaca informasi penting dan membuat suatu diagnosa (Malmivuo dan Plonsey, 1995). Penggunaan *EEG* dapat terdiri dari 1 hingga 256 elektroda yang merekam secara

paralel, atau disebut juga dengan *multichannel EEG*. Satu pasang *EEG* pada umumnya terekam dalam satu *channel* yang menghasilkan sinyal rekaman *EEG* dari letak elektroda tertentu.

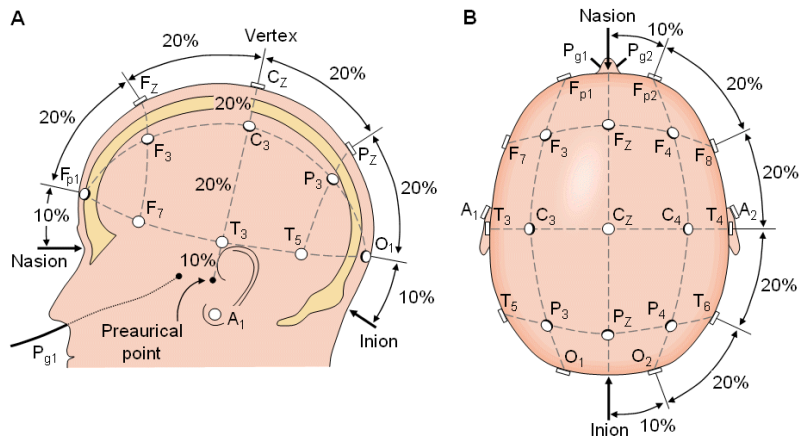


**Gambar 2.11** Ilustrasi Perekaman *EEG*  
(Sumber: Saint Luke's Healthy Sistem (Siuly dkk., 2016))

Amplitudo gelombang hasil perekaman *EEG* untuk orang dewasa umumnya berada pada range 1 hingga 100  $\mu V$ . Struktur arsitek otak tidak seragam dan korteks terorganisasi secara fungsionalnya dapat menyebabkan perekaman sinyal *EEG* tergantung pada lokasi peletakan elektroda.

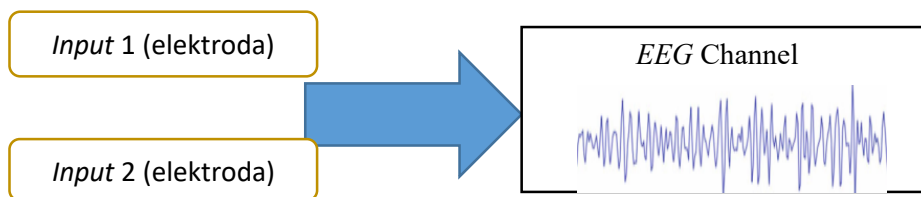
Peletakan elektroda menjadi hal yang penting dalam pengamatan sinyal *EEG* karena lobus atau bagian-bagian otak memiliki peran yang berbeda-beda. Terdapat dua tipe cara perekaman *EEG* berdasarkan letak penangkapan sinyal otak, yaitu *scalp* atau *intracranial*. Tipe peletakan *scalp* adalah metode peletakan elektroda-elektroda di permukaan kulit kepala dengan aturan peletakan standar internasional 10-20 seperti pada Gambar 2.12. Angka “10” dan “20” merepresentasikan jarak antar elektroda yang berada diantara 10% atau 20% dari total luas kepala dari depan ke belakang maupun dari kiri ke kanan (Klem dkk., 1999). Peletakan *intracranial* merupakan peletakan elektroda yang ditanam di dalam kepala manusia, pada umumnya metode peletakan ini dilakukan saat melakukan operasi yang membutuhkan pantauan sinyal *EEG* (Siuly dkk., 2016).

Penentuan peletakan elektroda mengikuti dua titik utama yaitu *nasion* dan *inion*. *Nasion* merupakan titik antara dahi dan hidung, setingkat dengan mata, sedangkan *inion* merupakan tonjolan pada tulang tengkorak pada bagian belakang. Ilustrasi peletakan elektroda pada Gambar 2.12 memiliki kode-kode unik huruf dan angka.



**Gambar 2.12** Standar Sistem Peletakan Elektroda 10-20 pada (A) Sisi Kiri Kepala dan (B) Tampak Atas Kepala. A = Ear lobe, C = central, Pg = nasopharyngeal, P = parietal, F = frontal, Fp = frontal polar, O = occipital (Malmivuo dan Plonsey, 1995)

Huruf F, T, C, P, O masing-masing mewakili bagian kepala *frontal*, *temporal*, *central*, *parietal* dan *occipital* yang dituliskan pada awal kode. Pada kode ke-dua terdiri atas angka, dengan angka ganjil yang menandakan bagian kiri sedangkan angka genap mewakili bagian kanan. Huruf z menunjukkan letak elektroda pada bagian tengah yang diletakkan pada garis lurus antara *nasion* dan *inion*. Karena tegangan sinyal merepresentasikan perbedaan antara dua elektroda seperti ilustrasi pada, sehingga tampilan *EEG* pada monitor mesin perekam dapat diatur pada beberapa kondisi yang disebut *montase* (Siuly dkk., 2016). *Montase* merupakan kombinasi derivasi untuk semua *channel* (Nihon Kohden Corp., n.d.).



**Gambar 2.13** Ilustrasi *Montase EEG*

Macam-macam *montase* yang dapat digunakan untuk menampilkan sinyal *EEG* adalah sebagai berikut:

1. *Montase Bipolar*:

Sepasang elektroda pada umumnya akan membangkitkan sebuah *channel* seperti yang ditampilkan pada Gambar 2.13. Masing-masing *channel* merepresentasikan perbedaan dua elektroda yang saling berdekatan (Schomer dan Silva, 2011). Contohnya berdasarkan peletakan elektroda pada Gambar



2.12 dapat diketahui bahwa F4 berdekatan dengan C4 sehingga terbentuk *montase bipolar* “F4-C4”. Rekaman sinyal *EEG* pada F4-C4 merupakan representasi beda potensial antara elektroda F4 dan C4. Aturan tersebut berlaku untuk semua *channel* yang terbentuk dari keseluruhan elektroda-elektroda yang terpasang.

2. *Montase Referensial:*

Masing-masing elektroda merepresentasikan perbedaan antara elektroda tertentu dengan dengan suatu elektroda tertentu yang ditentukan sebagai referensi (Schomer dan Silva, 2011). Tidak ada standar pemilihan elektroda sebagai referensi, namun demikian tetap terdapat perbedaan posisi peletakan elektroda-elektrodanya. Elektroda-elektroda di posisi garis tengah (z) seringkali digunakan sebagai referensi karena tidak memperkuat sinyal-sinyal siantar belahan otak dengan belahan lainnya (Siuly dkk., 2016).

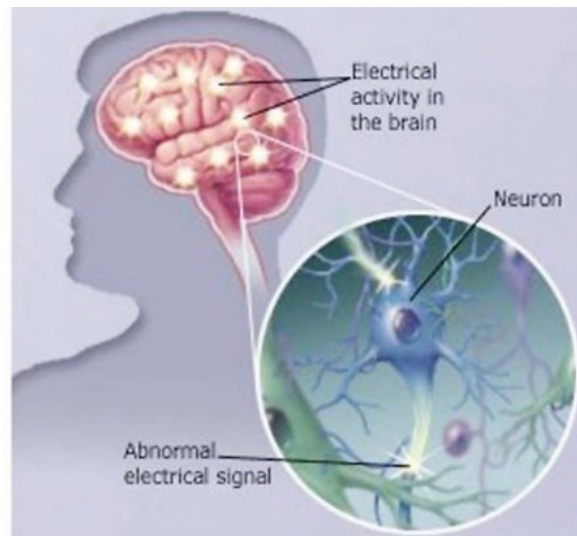
3. *Montase Referensi Rata-rata:*

Output sinyal *EEG* yang ditampilkan merupakan hasil penjumlahan dan selanjutnya dirata-ratakan, dengan nilai referensi adalah rata-rata sinyal dari semua elektroda (Lemos dan Fisch, 1991)

### 2.9.1 *EEG* untuk Diagnosa Epilepsi

Epilepsi adalah salah satu penyakit saraf yang paling sering terjadi di seluruh dunia (Supriya dkk., 2016). Ciri khas epilepsi adalah adanya kejang yang disebut kejang epilepsi (Fisher dkk., 2005). Kejang terjadi karena adanya perubahan fungsi kelistrikan pada otak secara tiba-tiba, sehingga merubah keadaan seseorang menjadi hilang kesadaran, hilang ingatan, gerakan tubuh yang serentak dan tak beraturan, sesak nafas, dan hilang ingatan. Perubahan fungsi kelistrikan tersebut umumnya terjadi di korteks atau di luar lingkaran otak. Epilepsi dapat berkembang karena adanya kelainan pada jaringan otak, ketidakseimbangan *neurotransmitter* atau kombinasi keduanya (Siuly dkk., 2016).

Ilustrasi pada **Gambar 2.14** adalah kondisi impuls listrik abnormal yang muncul saat terjadinya kejang. *Neuron* menghasilkan impuls elektrokimia yang bekerja pada neuron lain, kelenjar, dan otot untuk menghasilkan pikiran, perasaan, dan suatu tindakan manusia.



**Gambar 2.14** Ilustrasi impuls listrik abnormal saat kejang  
 (Sumber: <https://www.stabroeknews.com/2011/features/07/03/seizures/>)

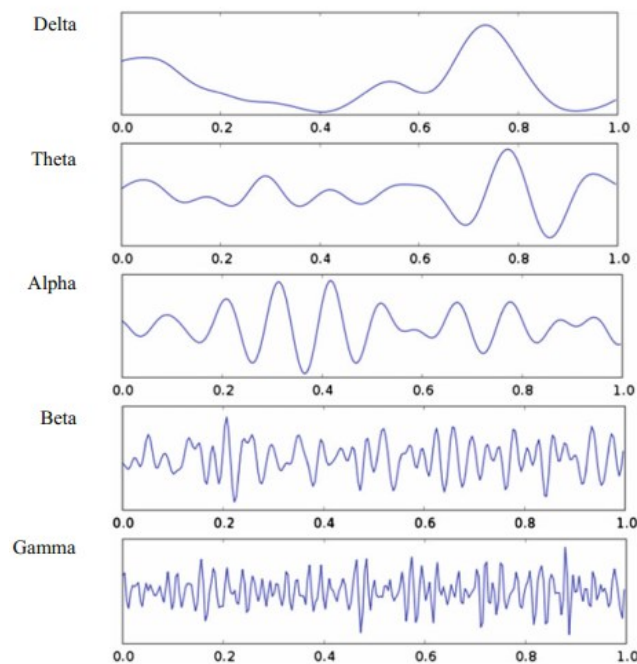
Pada kasus epilepsi, pola normal aktivitas *neuronal* menjadi terganggu sehingga menyebabkan sensasi aneh, emosi dan perilaku, atau terkadang kejang dan kehilangan kesadaran. Terdapat semacam “badai” listrik singkat yang timbul pada *neuron* yang tidak stabil karena cacat genetik (epilepsi turunan), atau karena kelainan metabolik seperti gula darah rendah atau karena alkohol. Pada kasus lain, adanya cedera di kepala juga dapat menyebabkan cairan otak yang tidak normal. Saat terjadi kejang, *neuron* akan meletup sekitar 500 kali dalam satu detik, jauh lebih cepat banyak daripada saat kondisi normal 1-100  $\mu V$ . Pada beberapa orang hal ini akan terjadi sesekali, namun bagi sebagian lainnya dapat terjadi ratusan kali dalam sehari (Siuly dkk., 2016).

### 2.9.2 Pola Sinyal *EEG*

Sinyal hasil perekaman *EEG* dimungkinkan dapat dibedakan atau dipisahkan berdasarkan frekuensinya. Frekuensi suatu sinyal menjadi salah satu kriteria yang penting dalam memahami *EEG* secara klinis. Frekuensi adalah banyaknya gelombang dalam 1 detik (Hz), dengan 1 gelombang terdiri atas satu bukit dan satu lembah gelombang. Adanya milyaran syaraf yang saling terhubung sebagai sumbernya, membuat potensial *EEG* manusia digolongkan sebagai sesuatu yang tidak dapat diprediksi secara periodik, dan dapat berubah-ubah sewaktu-waktu. Pada orang dewasa yang sehat, amplitudo dan frekuensi sinyalnya dapat berubah dari suatu keadaan ke keadaan lainnya, seperti bangun dan tidur. Secara garis besar,

gelombang otak dapat dibedakan menjadi 5 kelompok yang berbeda berdasarkan frekuensinya, yaitu gelombang delta ( $\delta$ ), theta ( $\theta$ ), alpha ( $\alpha$ ), beta ( $\beta$ ) dan gamma ( $\gamma$ ) (Siuly dkk., 2016). Pola-pola gelombang tersebut ditampilkan pada Gambar 2.15.

Range frekuensi 5 gelombang dari yang terendah ke yang tertinggi adalah  $\delta$ (0,5-4 Hz),  $\theta$  (>4-8 Hz),  $\alpha$  (>8-13 Hz),  $\beta$  (>13-30 Hz) dan  $\gamma$  (>30Hz) (Schomer dan Silva, 2011). Frekuensi yang tinggi seringkali mengindikasikan adanya gangguan aktivitas otak yang tidak normal, seperti epilepsi.



**Gambar 2.15** Contoh Macam-macam Gelombang  
(Sumber: Brainwave Entrainment, Itsu sync (Siuly dkk., 2016))

Deskripsi karakteristik umum macam-macam gelombang yang ditampilkan pada Gambar 2.15 adalah sebagai berikut.

1. Gelombang *Delta* ( $\delta$ )

Gelombang  $\delta$  berada pada kisaran frekuensi 0,5-4 Hz umumnya memiliki amplitudo tertinggi dan gelombang yang terlambat (frekuensi terendah). Gelombang ini terutama berkaitan dengan kondisi seseorang saat tertidur lelap, adanya gangguan otak yang serius dan kondisi orang yang sedang terjaga (Siuly dkk., 2016).

2. Gelombang *Theta* ( $\Theta$ )

Gelombang  $\Theta$  berada dikisaran frekuensi  $>4-8$  Hz yang pada umumnya memiliki amplitudo lebih besar dari  $20 \mu V$  (Schomer dan Silva, 2011). Gelombang  $\Theta$  umumnya muncul saat seseorang berada dalam kondisi stress, frustrasi, kecewa, memiliki inspirasi kreatif dan saat sedang bermeditasi dengan fokus yang dalam (Siuly dkk., 2016).

3. Gelombang *Alpha* ( $\alpha$ )

Gelombang  $\alpha$  merupakan gelombang dengan range frekuensi  $30-50$  Hz, dengan amplitudo sekitar  $30-50 \mu V$ . Gelombang ini seringkali muncul pada bagian posterior kepala (occipital lobe) ketika subjek pengamatan sedang menutup mata atau saat berada dalam keadaan relaksasi. Gelombang ini pada umumnya dikaitkan dengan aktivitas mental yang intens, stress dan dalam kondisi yang menegangkan (Siuly dkk., 2016). Aktivitas Gelombang  $\alpha$  yang direkam di area motor sensori juga disebut aktivitas *mu*.

4. Gelombang *Beta* ( $\beta$ )

Gelombang  $\beta$  berada pada range frekuensi  $>13-30$  Hz. Gelombang ini terlihat dalam amplitudo rendah dan frekuensinya bervariasi simetris yang sama pada kedua sisi area frontal. Jika otak terangsang dan aktif terlibat dalam aktivitas mental, maka akan menghasilkan gelombang  $\beta$  (Siuly dkk., 2016). Gelombang ini merupakan mencirikan kondisi berpikir keras. Gelombang  $\beta$  umumnya berkaitan dengan pikiran aktif, perhatian aktif dan saat fokus dalam menyelesaikan suatu permasalahan.

5. Gelombang *Gamma* ( $\gamma$ )

Gelombang  $\gamma$  memiliki frekuensi lebih dari  $30$  Hz. Gelombang ini terkadang didefinisikan memiliki frekuensi maksimal disekitar  $80$  atau  $100$  Hz. Gelombang  $\gamma$  dikaitkan dengan fungsi kognitif dan motorik (Siuly dkk., 2016).

## BAB 3

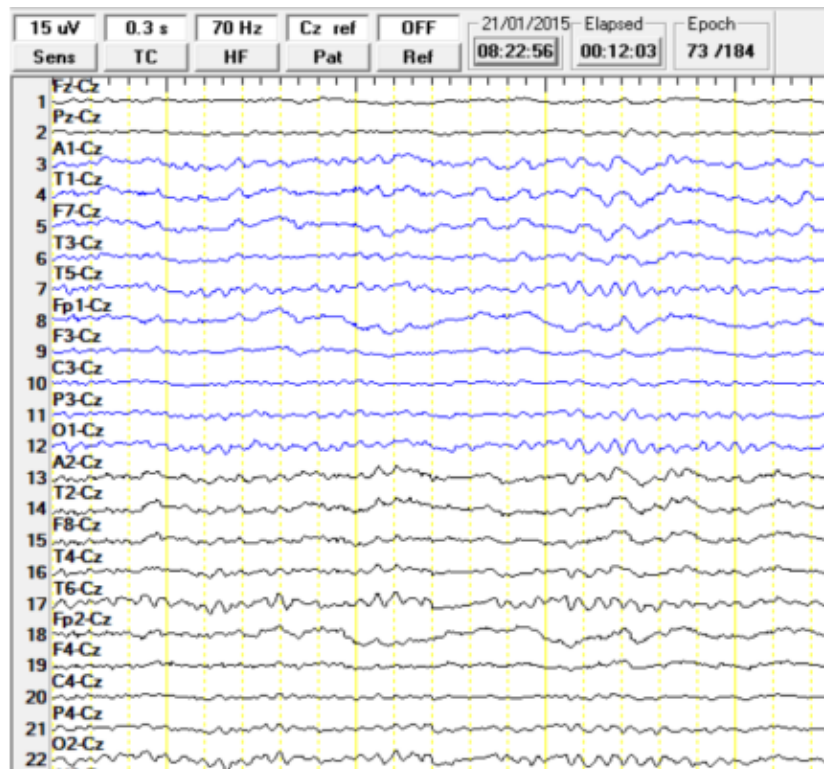
### METODOLOGI PENELITIAN

Penelitian ini dilaksanakan sesuai dengan metodologi penelitian yang secara umum terdiri atas tiga proses utama. Proses pertama adalah pengumpulan data, identifikasi variabel, dan langkah-langkah analisis penelitian.

#### 3.1 Sumber Data

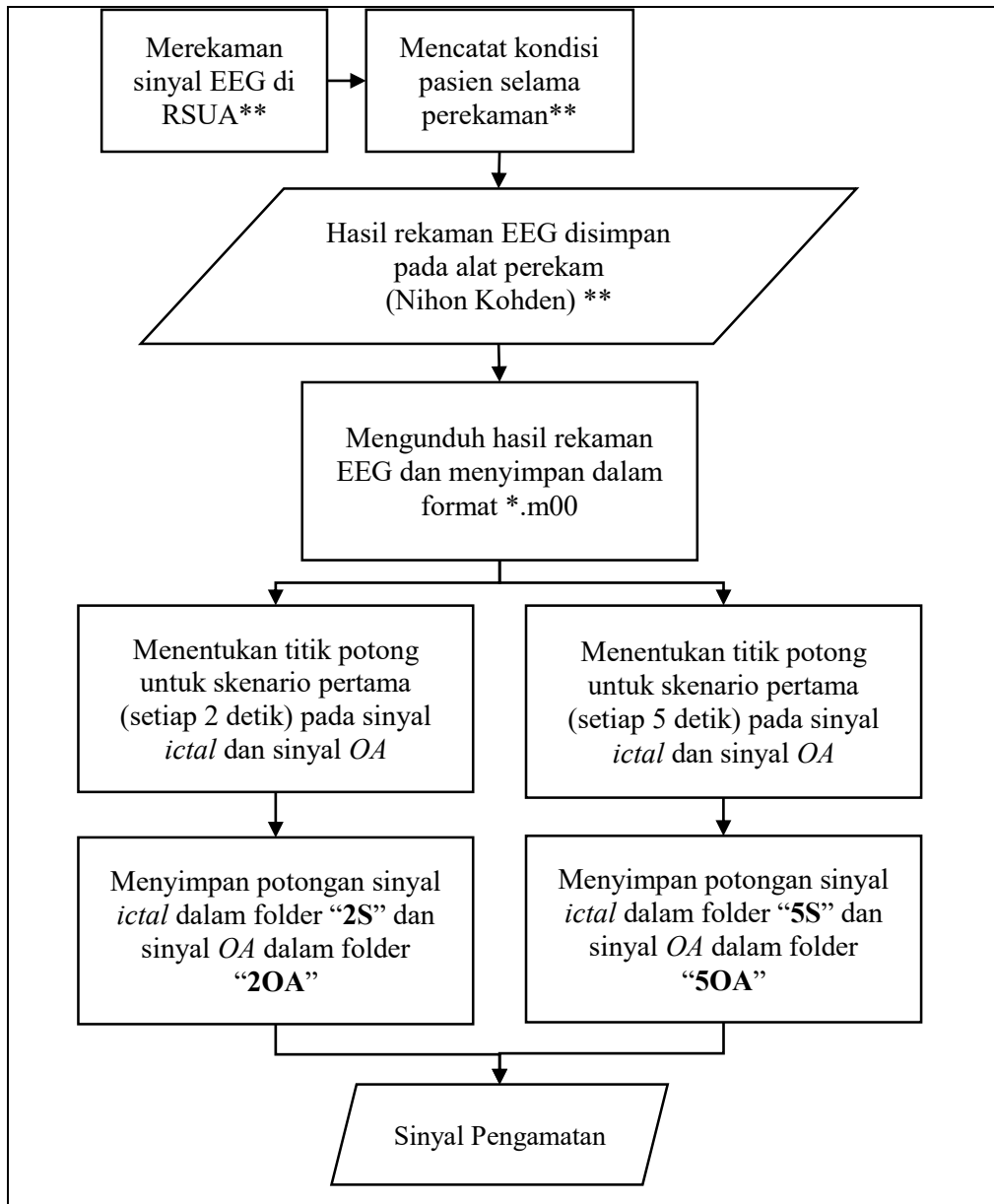
Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari hasil perekaman sinyal *EEG* secara *ekstrakranial* pada pasien epilepsi di Rumah Sakit Universitas Airlangga Surabaya (RSUA). Data tersebut berupa hasil perekaman sinyal *EEG* sebanyak 22 *channel* beda potensial dengan *montase Cz* dari elektroda yang diletakkan pada kulit kepala masing-masing pasien dengan aturan peletakan 10-20 seperti pada Gambar 2.12. Penggunaan data telah disetujui oleh RSUA, dengan bukti kode etik pada

Lampiran 1.



Gambar 3.1 Hasil Rekaman *EEG* dengan Montase *Cz*

Perekaman *EEG* dilakukan selama 30 menit disetiap sesinya. Data yang didapat dari semua elektroda kemudian direferensikan menggunakan montase Cz, sehingga diperoleh tampilan hasil rekaman *EEG* untuk masing-masing *channel* seperti pada Gambar 3.1. Pada penelitian ini, fokus penelitian hanya pada *channel* FP1 dan FP2, karena hanya pada kedua elektroda tersebut yang dapat merekam sinyal *ictal* dan sinyal *OA*.



Keterangan: \*\*=kegiatan yang dilakukan oleh dokter dan teknisi selama perekaman

**Gambar 3.2** Pematangan Sinyal *Ictal* dan *Ocular Artifacts*

Proses perekaman sinyal EEG dilakukan oleh dokter dan tim teknisi RSUD menggunakan metode ekstrakranial dan metode peletakan elektroda 10-20 seperti

pada Gambar 2.12. Selama perekaman dicatat pula keadaan klinis pasien pada beberapa kondisi seperti kejang dan mata berkedip. Catatan ini digunakan sebagai dasar titik awal pemotongan data yang digunakan pada penelitian ini. Semua hasil rekaman disimpan dalam aplikasi *Nihon Kohden* berupa ASCII (American Standard Code for Information Interchange) dengan format \*m00. Guna mempermudah dalam proses analisis, hasil unduhan data yang berupa format \*m00 dirubah menjadi format \*.csv *comma delimited*. Perekaman dilakukan selama 30 menit dengan *Sampling frequency rate (sfr)* dua *milisecond (ms)* sehingga dalam satu file unduhan terdiri atas 900.000 data di masing-masing elektroda yang sudah dimontase ke *cz*. Dalam satu rekaman ini hanya dipilih data yang merepresentasikan kondisi kejang (*ictal*) yang selanjutnya disebut sinyal *ictal* dan kondisi mata berkedip (*ocular artifacts*) yang selanjutnya disebut sinyal *OA*. Selanjutnya sinyal *ictal* dan sinyal *OA* akan dipotong dengan dua skenario pemotongan.

Skenario pemotongan pertama dengan durasi dua detik untuk masing-masing sinyal, dengan *sfr* 2 ms, sehingga dalam satu potong sinyal terdapat 1000 data. Sedangkan skenario pemotongan kedua dengan durasi lima detik, sehingga dalam satu potong sinyal terdapat 2500 data. Detail langkah-langkah teknis dan contoh pemotongan data dijelaskan pada bagian 3.3.1A. Kedua skenario pemotongan data ini dilakukan karena sinyal *ictal* memiliki kecenderungan terjadi dalam waktu lebih lama daripada munculnya sinyal *OA*.

Hasil pemotongan sinyal *EEG* kemudian disimpan dalam folder untuk mempermudah perhitungan komputasional dengan ketentuan sebagai berikut:

1. Potongan sinyal *ictal* skenario pertama disimpan dalam folder **II\_S**
2. Potongan sinyal *ictal* skenario kedua disimpan dalam folder **V\_S**
3. Potongan sinyal *OA* skenario pertama disimpan dalam folder **II\_OA**
4. Potongan sinyal *OA* skenario pertama disimpan dalam folder **V\_OA**

Pelambangan sinyal *ictal* diwakili oleh huruf kapital “S”. Angka romawi didepannya menunjukkan durasi pemotongan, dengan II memiliki arti pemotongan dengan durasi 2 detik dan V untuk durasi 5 detik.

### 3.2 Struktur Data

Struktur data setelah dilakukan pemotongan sinyal dapat disusun seperti pada Tabel 3.1. Angka 1 pada  $j$  mengindikasikan sinyal *ictal* sedangkan angka 2 mengindikasikan sinyal *OA*. Potongan sinyal ( $k$ ) sebanyak  $N$ , dan titik sampel ( $t$ ) merupakan urutan pengambilan sampel beda potensial setiap dua ms sebanyak 1000 sampel setiap potongnya.  $N1$  adalah jumlah potongan sinyal *ictal* dan  $N2$  adalah jumlah potongan sinyal *OA*, sehingga jumlah potongan keseluruhan dapat dihitung dengan menjumlahkan  $N1$  dan  $N2$ .

**Tabel 3.1** Struktur Sampel Beda Potensial Hasil Pemotongan

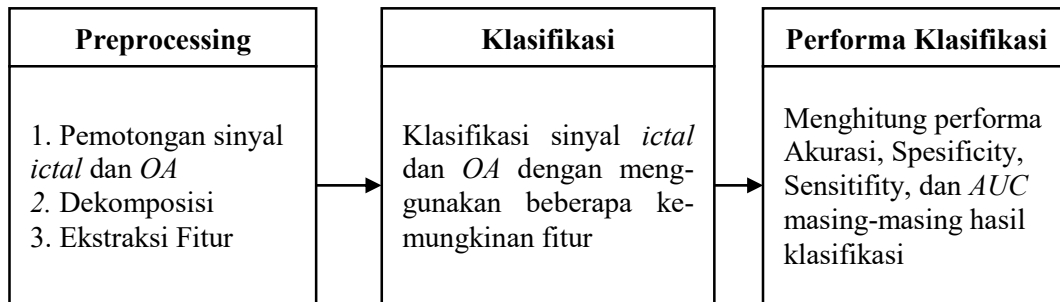
Durasi Pemotongan	Set	Potongan sinyal ke-	Beda Potensial Ke-			
			1	2	...	1000
2 detik	<i>II_S</i>	1	$z_{1,1,1}$	$z_{1,1,2}$	...	$z_{1,1,1000}$
		2	$z_{1,2,1}$	$z_{1,2,2}$	...	$z_{1,2,1000}$
		⋮	⋮	⋮	...	⋮
		$N1$	$z_{1,N1,1}$	$z_{1,N1,2}$	...	$z_{1,N1,1000}$
	<i>II_OA</i>	1	$z_{2,1,1}$	$z_{2,1,2}$	...	$z_{2,1,1000}$
		2	$z_{2,2,1}$	$z_{2,2,2}$	...	$z_{2,2,1000}$
		⋮	⋮	⋮	...	⋮
		$N2$	$z_{2,N2,1}$	$z_{2,N2,2}$	...	$z_{2,N2,1000}$
Durasi Pemotongan	Set	Potongan sinyal ke-	Beda Potensial Ke-			
			1	2	...	2500
5 detik	<i>V_S</i>	1	$z_{3,1,1}$	$z_{1,1,2}$	...	$z_{1,1,2500}$
		2	$z_{3,2,1}$	$z_{1,2,2}$	...	$z_{1,2,2500}$
		⋮	⋮	⋮	...	⋮
		$N3$	$z_{3,N3,1}$	$z_{1,N3,2}$	...	$z_{1,N3,2500}$
	<i>V_OA</i>	1	$z_{4,1,1}$	$z_{2,1,2}$	...	$z_{2,1,2500}$
		2	$z_{4,2,1}$	$z_{2,2,2}$	...	$z_{2,2,2500}$
		⋮	⋮	⋮	...	⋮
		$N4$	$z_{4,N4,1}$	$z_{2,N4,2}$	...	$z_{2,N4,2500}$

Nilai  $z_{jkt}$  pada Tabel 3.1 merupakan beda potensial listrik antara dua elektroda yang dipasangkan di kulit kepala dengan konsep *montase Cz* yang dijelaskan pada sub-bab 2.9. Perbedaan muatan positif dan negatif pada bagian elektroda yang berpasangan akan menyebabkan penurunan atau kenaikan nilai dari nilai  $z_{jkt}$  sebelumnya. Jika tidak terdapat perbedaan potensial maka nilai  $z_{jkt}$  cenderung stabil dari nilai  $z_{jkt}$  sebelumnya. Langkah-langkah pemotongan sampel sinyal *Ictal* dan *OA* dijelaskan pada sub-sub bab 3.3.1.



### 3.3 Langkah Analisis

Langkah-langkah analisis untuk mendapatkan tujuan penelitian ini dibagi menjadi tiga tahapan utama yaitu *preprocessing*, klasifikasi sinyal *EEG* kondisi *ictal* dan *OA* dan pemilihan hasil klasifikasi yang paling optimum. Langkah-langkah analisis secara umum dapat ditunjukkan seperti pada **Gambar 3.3**



Gambar 3.3 Langkah Analisis Secara Umum

#### 3.3.1 Preprocessing

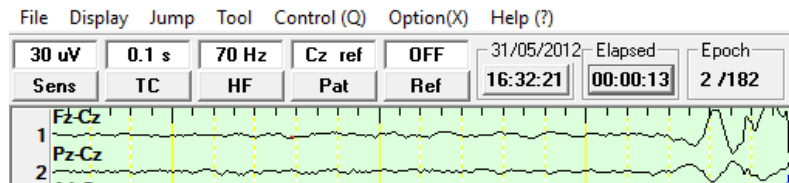
Tahapan *preprocessing* pada penelitian ini merupakan tahapan untuk mendapatkan data baru yang memenuhi syarat klasifikasi. Tahapan ini terdiri atas pemotongan sinyal *ictal* dan sinyal *OA* dari hasil rekaman *EEG* yang tersimpan dalam aplikasi *Nihon Kohden*. Setelah mendapatkan sinyal *ictal* dan *OA*, selanjutnya dilakukan dekomposisi untuk mendapatkan *IMF* dengan menggunakan *EMD* yang selanjutnya ditransformasi menggunakan *Hilbert transform*. *IMF-IMF* yang terbentuk dari *EMD* selanjutnya diekstrak menjadi beberapa fitur *Bandwidth* dan fitur *time frequency*.

##### A. Pemotongan Sinyal *Ictal* dan *OA*

Pemotongan data merupakan hal yang sangat penting dalam penelitian ini, karena dapat mempengaruhi hasil analisis secara keseluruhan. Pemilihan sinyal *ictal* dan sinyal *OA* akan melibatkan pengamatan secara subjektif dari dokter saraf RSUA sebagai validator, dengan didukung catatan kejadian-kejadian yang muncul saat perekaman *EEG* oleh tim teknisi RSUA. Langkah-langkah pemotongan sinyal adalah sebagai berikut:

1. Membuka data hasil rekaman pada aplikasi *Nihon Kohden*.
2. Memilih nama pasien yang akan diamati hasil rekamannya.

- Melakukan montase hasil rekaman ke *channel Cz*, dengan contoh seperti pada Gambar 3.1. Pengaturan tampilan dan motase dapat dilihat pada bagian kanan atas tampilan layar utama, seperti yang ditampilkan pada Gambar 3.4.



**Gambar 3.4** Display Pengaturan di Aplikasi *Nihon Kohden*

Tombol “Pat” berfungsi untuk mengatur jenis montase, tombol “HF” digunakan untuk mengatur nilai *frequency* maksimal yang digunakan. Sedangkan tombol lainnya digunakan untuk mengatur tampilan gelombang agar mudah dikenali polanya.

- Melakukan *export* data dengan klik tombol “File” → “Export”, untuk menyimpan data dalam bentuk *ASCII* hingga didapat file \*.txt. Contoh data hasil *export* dapat dilihat pada Gambar 3.5.

	Fz-Cz	Pz-Cz	A1-Cz	T1-Cz	F7-Cz	T3-Cz	T5-Cz	Fp1-Cz	F3-Cz	C3-Cz	P3-Cz	O1-Cz	A2-Cz
1													
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

**Gambar 3.5** Hasil *Export* Data dari Aplikasi *Nihon Kohden*

Kolom pertama “Fz-Cz” merupakan montase dari *channel Fz* dengan Cz, dan kolom lainnya merupakan *montase* semua *channel* terhadap Cz. Jumlah data pada masing-masing *channel* yang didapat dari proses *export* sebanyak 907500 data.

- Menentukan data ke berapa yang akan dipotong berdasarkan catatan dokter dan tim teknisi yang berisi informasi waktu terjadinya *seizure* atau *ictal*. Berikut adalah langkah-langkah untuk mendapatkan titik awal pemotongan.

- a. Mengubah semua satuan waktu ke dalam detik. format waktu (*jj.mm.dd*), di mana *jj*= jam, *mm*=menit, dan *dd*=detik, maka titik awal (*start poin*) dapat dihitung dengan rumus berikut ini.

$$Start\ point = 1 + ((jj \times 3600) + (mm \times 60) + (dd)) \times sfr \quad (3.1)$$

*Sampling frequency rate sfr* pada penelitian ini ditentukan sebesar 500, artinya terdapat 500 data terekam dalam 1 detik, yang struktur datanya dituliskan pada Tabel 3.1.

- b. Menentukan titik akhir pemotongan untuk 1 potong sinyal.

$$End\ point = Start\ poin + ((2 \times sfr) - 1) \quad (3.2)$$

Angka 2 pada Persamaan (3.2) merupakan durasi waktu rekaman sinyal dalam 1 potongan sinyal, yaitu duadetik.

### **Contoh Penerapan**

Pada Catatan dokter menyatakan bahwa di detik ke 17 terjadi *seizure* (Gambar 3.6). Sehingga dapat ditentukan pemotongan data sebagai berikut.

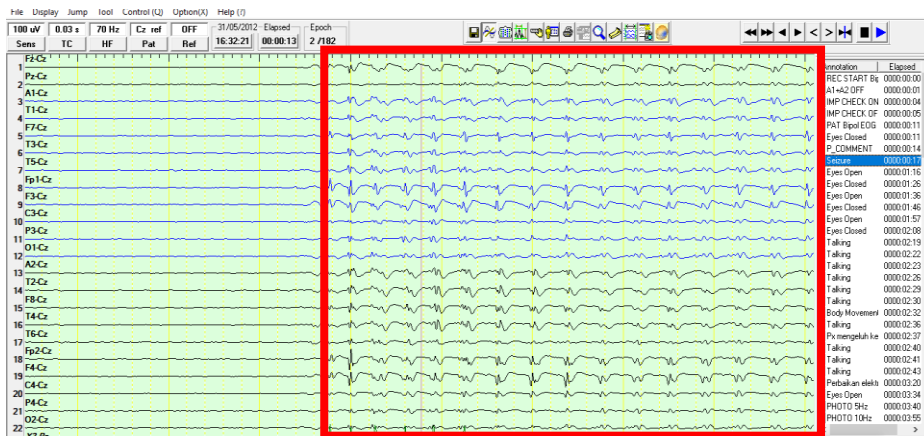
$$Start\ point = 1 + ((0 \times 3600) + (0 \times 60) + (17)) \times 500 = 8501$$

$$End\ point = 8501 + 999 = 9500$$

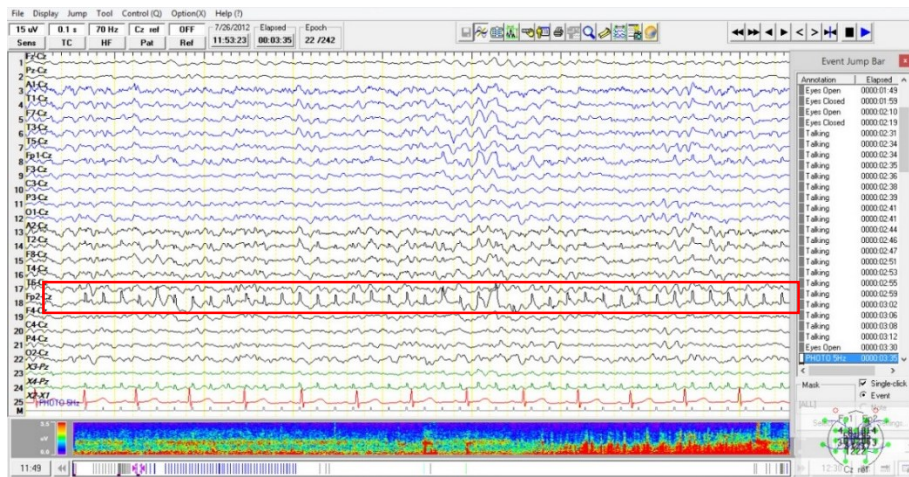
Secara visual, sinyal *Ictal* dan sinyal *OA* memiliki ciri-ciri yang berbeda. Sinyal *ictal* cenderung memiliki amplitudo gelombang yang lebih tinggi seperti yang ditampilkan pada Gambar 3.6.

Sinyal *ictal* muncul ketika orang yang diamati mengalami kejang epilepsi dengan durasi tertentu dan dapat diamati dengan jelas pergerakan kejangnya. Saat mengalami kejang, letupan kelistrikan di otak dapat berupa *focal* dan *general*. Kondisi *focal* menunjukkan lokasi letupan kelistrikan di otak hanya terjadi pada bagian tertentu saja, yang dapat dilihat dari elektroda dengan pola beda potensial tertinggi. Sedangkan kondisi *general* menunjukkan lokasi letupan kelistrikan di otak terjadi di seluruh bagian seperti yang ditampilkan oleh pola gelombang di dalam kotak merah pada Gambar 3.6.

Sinyal *OA* memiliki pola yang berbeda dengan sinyal *ictal*. Gambar 3.7 merupakan hasil rekaman *EEG* yang mengandung *OA* pada *channel Fp-Cz* (di dalam kotak merah). Pola sinyalnya cenderung lebih tenang daripada sinyal *ictal* dan naik di interval waktu tertentu, tetapi tidak sebesar sinyal *ictal*.



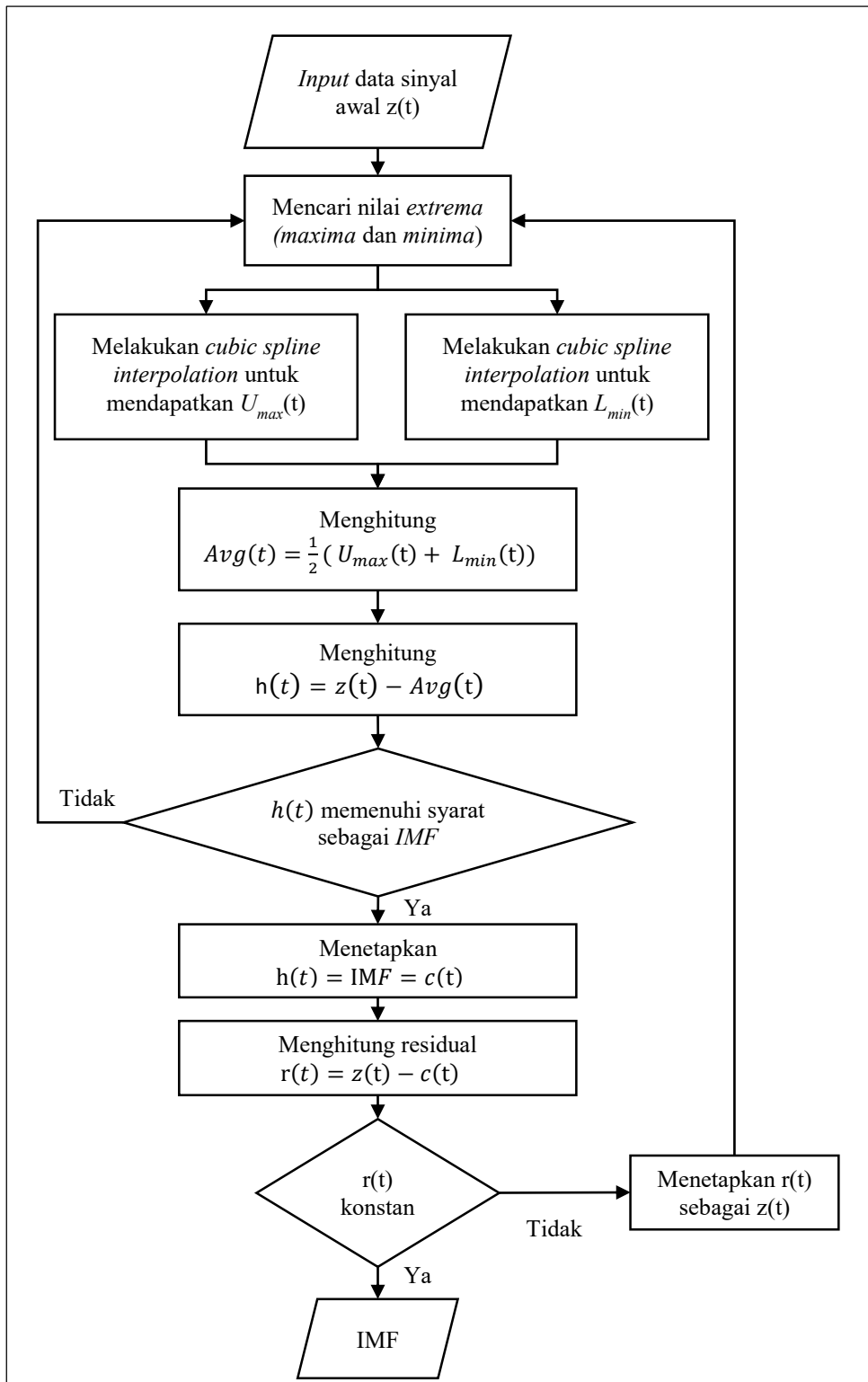
Gambar 3.6 Sinyal *Ictal* pada Perikaman EEG



Gambar 3.7 Sinyal *OA* pada Perikaman EEG

## B. Dekomposisi Sinyal menggunakan *EMD*

Dekomposisi dapat dilakukan setelah tahap pemotongan data selesai dilakukan. *EMD* dapat mendekomposisi sinyal mentah menjadi beberapa sinyal yang disebut *IMF* dan satu sinyal residu. Tahap ini juga menggunakan *Hilbert transform* sebagai konvolusi untuk pembentukan *IMF*. Algoritma *EMD* pada Sub-bab 2.2 dapat dituliskan kembali dalam diagram alir pada Gambar 3.8.



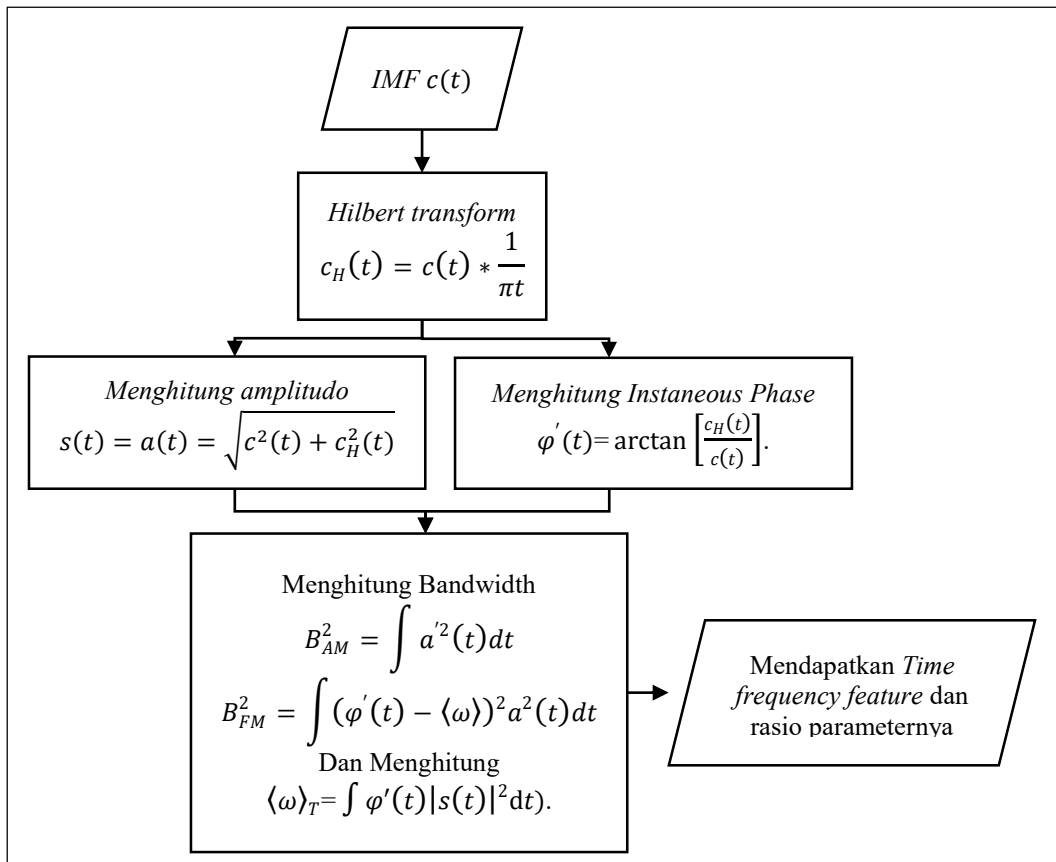
Gambar 3.8 Empirical Mode Decomposition

Setelah menjalankan algoritma EMD, didapatkan IMF sebanyak  $K$  dan satu residu yang merupakan peyusun dari  $z(t)$  atau secara matematis dapat dituliskan seperti pada Persamaan (2.1). Selanjutnya IMF  $c(t)$  akan dibawa kedalam domain

*time frequency* dengan transformasi menggunakan *Hilbert transform* dan dinyatakan sebagai  $s(t)$  yang dapat dihitung menggunakan Persamaan (2.3).

### C. Ekstraksi Fitur

Ekstraksi fitur dilakukan untuk mendapatkan fitur-fitur yang dapat digunakan sebagai prediktor dalam klasifikasi sinyal *ictal* dan *OA*. Fitur yang digunakan dalam penelitian ini berasal dari *bandwidth* sinyal yang didapatkan dari *IMF* hasil dekomposisi pada langkah sebelumnya.



Gambar 3.9 Ekstraksi Fitur

Hilbert transform digunakan untuk mengubah domain *time* pada hasil rekaman *EEG* menjadi domain *time frequency* agar dapat dilanjutkan ke tahapan selanjutnya. *Time frequency features* terdiri atas  $f_r$  dan rasio parameter  $f_r$  ( $\Gamma_{MS}$ ),  $f_d$  dan rasio parameter  $f_d$  ( $\Delta_{DF}^2$ ),  $f_R$  dan rasio parameter  $f_R$  ( $\Delta_{DF}^2$ ). Rumus untuk menghitung *time frequency features* dan rasio parameternya dapat dilihat pada Sub-sub bab 2.4.1.

**Tabel 3.2** Variabel untuk Tahapan Klasifikasi Hasil Ekstraksi Fitur

Nama Variabel	Kode Variabel	Keterangan
Respon/Class	Y	Kelas sinyal <i>EEG</i> 0: <i>Ictal</i> 1: <i>OA</i>
<i>Bandwidth amplitude</i> (X1)	X1.1	<i>Bandwidth amplitude</i> ( $B_{AM}$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>Bandwidth frequency</i> (X2)	X1.4	$B_{AM}$ <i>IMF</i> dari ke-4
	⋮	⋮
<i>RMS frequency</i> (X3)	X2.1	<i>Bandwidth amplitude</i> ( $B_{FM}$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>RMS frequency</i> (X3)	X2.4	$B_{AM}$ <i>IMF</i> dari ke-4
	⋮	⋮
<i>Rasio Parameter <math>f_r</math></i> (X4)	X3.1	<i>RMS frequency</i> ( $f_r$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>Dominant frequency</i> (X5)	X3.4	$f_r$ <i>IMF</i> dari ke-K
	⋮	⋮
<i>Rasio Parameter <math>f_d</math></i> (X6)	X4.1	Rasio Parameter $f_r$ ( $\Gamma_{MS}$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>RMIFS Frequency</i> (X7)	X4.4	$\Gamma_{MS}$ dari <i>IMF</i> ke-4
	⋮	⋮
<i>Rasio Parameter <math>f_d</math></i> (X6)	X5.1	<i>Dominant frequency</i> ( $f_d$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>Rasio Parameter <math>f_d</math></i> (X6)	X5.4	$f_d$ dari <i>IMF</i> ke-4
	⋮	⋮
<i>Rasio Parameter <math>f_R</math></i> (X8)	X6.1	Rasio Parameter $f_d$ ( $\Delta_{DF}^2$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>Rasio Parameter <math>f_R</math></i> (X8)	X6.4	$\Delta_{DF}^2$ dari <i>IMF</i> ke-4
	⋮	⋮
<i>Rasio Parameter <math>f_R</math></i> (X8)	X7.1	<i>RMIFS Frequency</i> ( $f_R$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>Rasio Parameter <math>f_R</math></i> (X8)	X7.4	$f_R$ dari <i>IMF</i> ke-4
	⋮	⋮
<i>Rasio Parameter <math>f_R</math></i> (X8)	X8.1	Rasio Parameter $f_R$ ( $E_{MIFS}$ ) dari <i>IMF</i> ke-1
	⋮	⋮
<i>Rasio Parameter <math>f_R</math></i> (X8)	X8.4	$(E_{MIFS})$ dari <i>IMF</i> ke-4
	⋮	⋮

Jumlah *IMF* ditentukan sebanyak 4 ( $K=4$ ), maka masing-masing fitur atau variabel prediktor pada Tabel 3.2 terdiri atas 4 fitur. Dengan demikian, dapat disusun struktur data untuk tahapan klasifikasi seperti pada Gambar 3.4, dengan jumlah pengamatan sebanyak  $N$ , maka struktur data setelah ekstraksi fitur dapat disusun seperti pada Tabel 3.3.

**Tabel 3.3** Struktur data pada tahapan klasifikasi

Sinyal	$X_{1.1,i}$	...	$X_{1.4,i}$	...	$X_{8.1,i}$	...	$X_{8.4,i}$	$Y_i$
1	$X_{1.1,1}$	...	$X_{1.4,1}$	...	$X_{8.1,1}$	...	$X_{8.4,1}$	$Y_1$
2	$X_{1.1,2}$	...	$X_{1.4,2}$	...	$X_{8.1,2}$	...	$X_{8.4,2}$	$Y_2$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$i$	$X_{1.1,i}$	...	$X_{1.4,i}$	...	$X_{8.1,i}$	...	$X_{8.4,i}$	$Y_i$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$N$	$X_{1.1,N}$	...	$X_{1.4,N}$	...	$X_{8.1,N}$	...	$X_{8.4,N}$	$Y_N$

$X_{1.1,1}$  adalah nilai fitur *RMS frequency* pada *IMF* pertama dari potongan sinyal pertama,  $X_{1.2,1}$  adalah nilai fitur *rasio parameter RMS frequency* pada *IMF* ke-dua dari potongan sinyal pertama, dan seterusnya. Variabel respon atau kelas akan

bernilai 0 jika sinyal ke- $i$  adalah sinyal *ictal* dan bernilai 1 jika sinyal ke- $i$  adalah sinyal *OA*, dengan  $i = 1, 2, \dots, N$ .

### 3.3.2 Klasifikasi Sinyal *Ictal* dan *Ocular Artifacts*

Klasifikasi sinyal *ictal* dan *OA* menggunakan metode *SVM* dengan kernel linier, *RBF* dan *Polynomial* dapat dilakukan setelah diperoleh hasil ekstraksi fitur dengan langkah-langkah pada sub-bab sebelumnya. Langkah-langkah klasifikasi menggunakan metode *SVM* setelah diperoleh fitur-fitur yang dibutuhkan adalah.

1. *Input* data respon (**Y**), dan fitur **X1, X2, X3, X4, X5, X6, X7** dan **X8**. Fitur yang tergabung dalam **X1** terdiri atas fitur  $X1.1, X1.2, \dots, X1.4$ . Fitur yang tergabung dalam **X2** terdiri atas fitur  $X2.1, X2.2, \dots, X2.4$ . Fitur yang tergabung dalam **X3** terdiri atas fitur  $X3.1, X3.2, \dots, X3.4$ . Fitur yang tergabung dalam **X4** terdiri atas fitur  $X4.1, X4.2, \dots, X4.4$ . Fitur yang tergabung dalam **X5** terdiri atas fitur  $X5.1, X5.2, \dots, X5.4$ . Fitur yang tergabung dalam **X6** terdiri atas fitur  $X6.1, X6.2, \dots, X6.4$ . Fitur yang tergabung dalam **X7** terdiri atas fitur  $X7.1, X7.2, \dots, X7.4$ . Fitur yang tergabung dalam **X8** terdiri atas fitur  $X8.1, X8.2, \dots, X8.4$ .

**Tabel 3.4** Kombinasi Fitur untuk Klasifikasi

Skenario	Respon	Fitur	Prediktor/fitur untuk Kombinasi Ke-				
			1	2	3	4	5
1 (Bandwidth)	Y	$B_{AM}$	X1.1	X1.2	X1.3	X1.4	X1.1, X1.2, X1.3, X1.4
	Y	$B_{FM}$	X2.1	X2.2	X2.3	X2.4	X2.1, X2.2, X2.3, X2.4
	Y	$B_{AM}, B_{FM}$	X1.1, X2.1	X1.2, X2.2	X1.3, X2.3	X1.4, X2.4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4
2 (Time Frequency)	Y	$f_r$	X3.1	X3.2	X3.3	X3.4	X3.1, X3.2, X3.3, X3.4
	Y	$\Gamma_{MS}$	X4.1	X4.2	X4.3	X4.4	X4.1, X4.2, X4.3, X4.4
	Y	$f_d$	X5.1	X5.2	X5.3	X5.4	X5.1, X5.2, X5.3, X5.4
	Y	$\Delta_{DF}^2$	X6.1	X6.2	X6.3	X6.4	X6.1, X6.2, X6.3, X6.4
	Y	$f_R$	X7.1	X7.2	X7.3	X7.4	X7.1, X7.2, X7.3, X7.4
	Y	$E_{MIFS}$	X8.1	X8.2	X8.3	X8.4	X8.1, X8.2, X8.3, X8.4
	Y	$f_r, \Gamma_{MS}, f_d$	X3.1, X4.1, X5.1	X3.2, X4.2, X5.2	X3.3, X4.3, X5.3	X3.4, X4.4, X5.4	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4, X5.1, X5.2, X5.3, X5.4
	Y	$\Delta_{DF}^2, f_R, E_{MIFS}$	X6.1, X7.1, X8.1	X6.2, X7.2, X8.2	X6.3, X7.3, X8.3	X6.4, X7.4, X8.4	X6.1, X6.2, X6.3, X6.4, X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4

Kombinasi klasifikasi pada Tabel 3.4 merupakan kombinasi keseluruhan pada penelitian ini. Klasifikasi pertama pada skenario pertama menggunakan fitur X1.1, klasifikasi kedua menggunakan fitur X1.2, dan seterusnya hingga

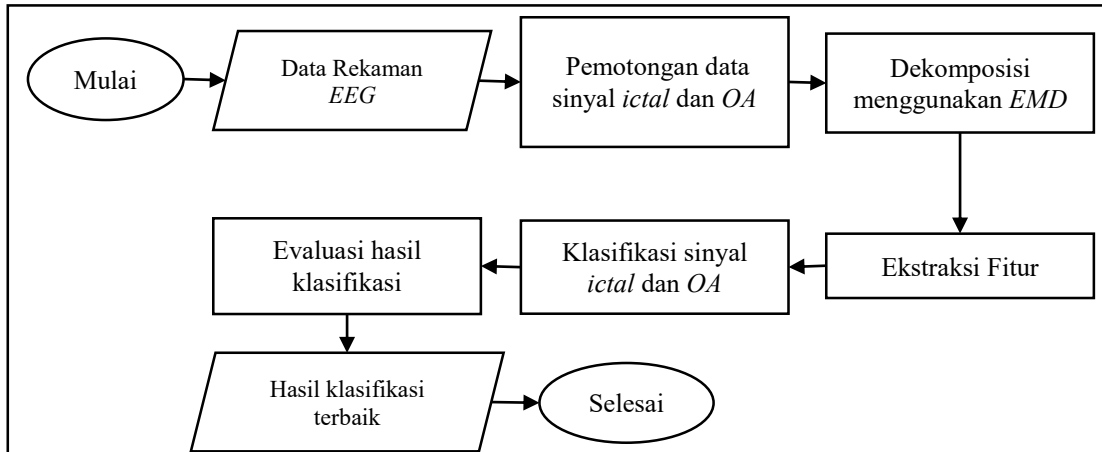


kombinasi terakhir pada skenario kedua menggunakan fitur X6.1, X6.2, X6.3, X6.4, X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4 secara bersamaan dalam satu kali klasifikasi.

2. Membagi data menjadi training dan *testing*, dengan 90% data sebagai training dan 10% data testing. Kombinasi penggunaan fitur-fitur untuk klasifikasi menggunakan semua kemungkinan kombinasi fitur pada Tabel 3.4. Skenario klasifikasi yang pertama adalah menggunakan *input* fitur *Bandwidth* ( $B_{AM}$  dan  $B_{FM}$ ), sedangkan skenario klasifikasi yang kedua menggunakan *input Time frequency features* ( $f_r$ ,  $f_d$ ,  $f_R$  dan rasio parameternya). Kombinasi fitur ini digunakan pada data dengan potongan 2 detik dan 5 detik.
3. Melakukan klasifikasi menggunakan *SVM* dengan kernel linier, *RBF* dan *polynomial* menggunakan tahapan-tahapan berikut (Vijayakumar dan Wu, 1999):
  - a. *Input* data dengan struktur seperti pada kemungkinan pertama dalam Tabel 3.4, dan membagi data *training* dan *testing* dengan aturan *10-fold cross validation* sehingga diperoleh 10 kombinasi data *training* dan *testing*.
  - b. Menentukan Kernel yang akan digunakan.
  - c. Menentukan range nilai parameter  $C$  dan  $\gamma$  menggunakan *exhaustive grid search*.
  - d. Melakukan klasifikasi *SVM* dengan kombinasi nilai parameter  $C$  dan  $\gamma$  beserta kernel yang telah ditentukan.
  - e. Menghitung rata-rata *accuracy*, *sensitify*, *specificity*, dan *AUC* data *testing* dari *10-fold cross validation*, sehingga didapatkan satu nilai terbesar dari masing-masing kriteria kebaikan model.
  - f. Mengulangi langkah 3b sampai 3e, sampai semua kombinasi parameter telah dilakukan
4. Mengulangi langkah 3 untuk semua kemungkinan pada Tabel 3.4
5. Menentukan hasil klasifikasi terbaik dari semua kemungkinan yang terbantu berdasarkan nilai *accuracy*, *sensitify*, *specificity*, dan *AUC* terbesar.

### 3.4 Diagram Alir Penelitian

Diagram alir pada langkah analisis sub bab sebelumnya merupakan diagram alir masing-masing tahap. Secara umum, tahapan tersebut dapat disatukan dalam diagram alir yang ditampilkan pada Gambar 3.10.



Gambar 3.10 Diagram Alir Penelitian

## **BAB 4**

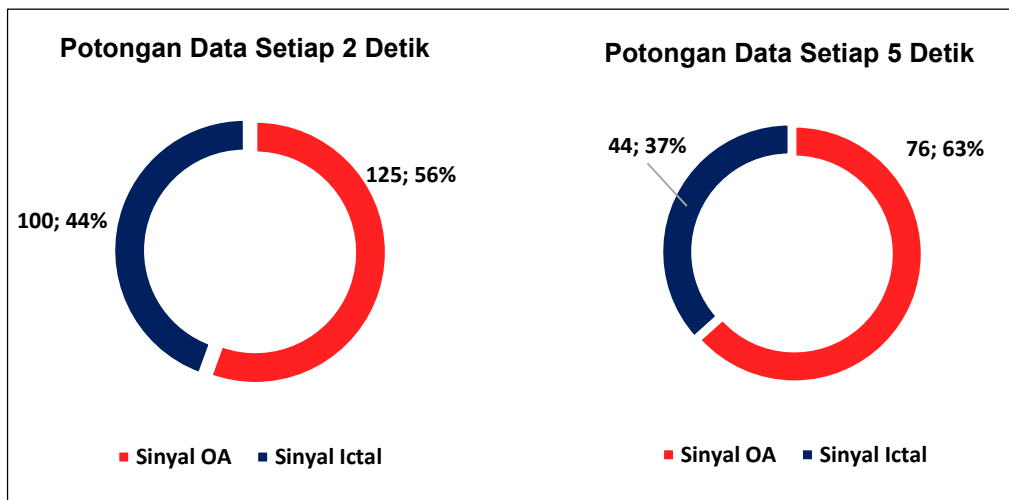
### **ANALISIS DAN PEMBAHASAN**

Bab ini membahas mengenai hasil dari langkah-langkah analisis yang telah dijelaskan pada Bab 3. Hasil analisis yang dibahas meliputi pemotongan sinyal *ictal* dan *OA*, dekomposisi sinyal *EEG*, ekstraksi fitur, klasifikasi sinyal, dan evaluasi hasil klasifikasi dari model yang dibentuk.

#### **4.1 Pemotongan Sampel Sinyal *EEG***

Perekaman *EEG* selama 30 menit menghasilkan 22 pasang sinyal *EEG* berupa beda potensial antar elektroda. Jumlah sampel yang terambil setiap detiknya adalah 500 data. Fokus penelitian ini adalah pada sinyal *OA* yang muncul saat perekaman, dan adanya sinyal ini hanya pada channel *FPI* dan *FP2*. Sehingga dari 22 channel hanya diamati dua channel tersebut. Hal ini juga berlaku pada sinyal aturan pemotongan sinyal *ictal*, dengan hanya mengambil sinyal *ictal* yang muncul pada channel *FPI* dan *FP2*. Pada penelitian ini jenis *montase* ditentukan terhadap *Cz*, karena *Cz* merupakan titik tengah dari peletakan elektorda dalam aturan 10-20. Proses pemotongan *ictal* telah dijelaskan pada Sub-bab 3.3.1. Hasil perekaman *EEG* terdapat berbagai macam jenis sinyal jika ditinjau dari karakteristik pasien saat perekaman. Namun pada penelitian ini dibatasi hanya pada pengamatan sinyal saat terjadi kejang epilepsi yang selanjutnya disebut sinyal *ictal* dan sinyal saat pasien melakukan gerakan mata berkedip yang selanjutnya disebut *Ocular Artifacts (OA)*. Kode pemotongan data menggunakan software *python* dapat dilihat pada Lampiran 5, Lampiran 6, Lampiran 7, dan Lampiran 8.

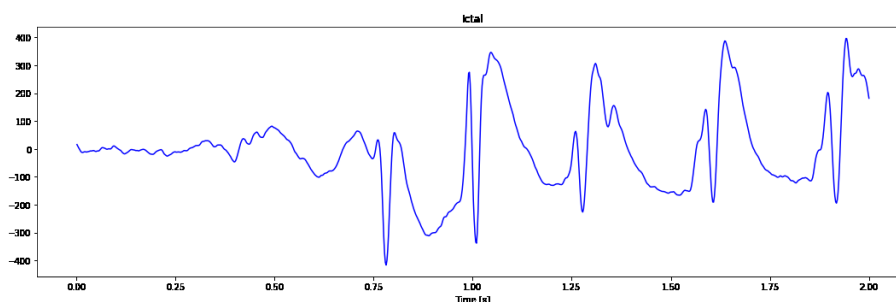
Gambar 4.1 menunjukkan bahwa dari hasil pemotongan dengan durasi lima detik, diperoleh sinyal *OA* sebanyak 76 potongan dan sinyal *ictal* sebanyak 44 potongan. Sedangkan pada pemotongan dengan durasi dua detik, diperoleh sinyal *OA* sebanyak 125 potongan dan sinyal *ictal* sebanyak 100 potongan. Sehingga pada analisis dengan durasi pemotongan lima terdapat 120 sinyal dan pada analisis dengan durasi potong dua detik terdapat 225 sinyal. Berdasarkan jumlah potongan data sinyal *ictal* dan sinyal *OA* untuk potongan data setiap 2 detik dan 5 detik, dianggap bahwa tidak ada kasus *inbalance*.



Gambar 4.1 Jumlah Sinyal Hasil Pemetongan

#### 4.1.1 Sinyal Ictal

Sinyal *ictal* merepresentasikan kondisi gelombang otak pasien saat terjadinya kejang epilepsi. Secara visual pola sinyal *ictal* pada Gambar 4.2 memiliki pola *nonstasioner* dan *nonlinear*. Sinyal *ictal* memiliki frekuensi gelombang yang menyerupai gelombang *alpha*. Pola sinyal *ictal* diawali dengan frekuensi gelombang yang tinggi, lalu amplitudo naik namun ke arah bawah, dan selanjutnya mengalami kenaikan yang signifikan dengan disertai menurunnya frekuensi sinyal. Pola tersebut tertangkap dan berulang-ulang saat kejang epilepsi. Pola ini hampir sama dengan sinyal *sharpwave*, namun pada sinyal *sharpwave* hanya terjadi dalam waktu yang singkat dan bahkan seringkali terjadi dengan durasi kurang dari satu detik. Sedangkan pada sinyal *ictal*, memiliki durasi yang panjang dan seringkali kondisi pasien dapat dilihat dengan mata normal sedang mengalami pergerakan acak secara bersamaan atau kejang.

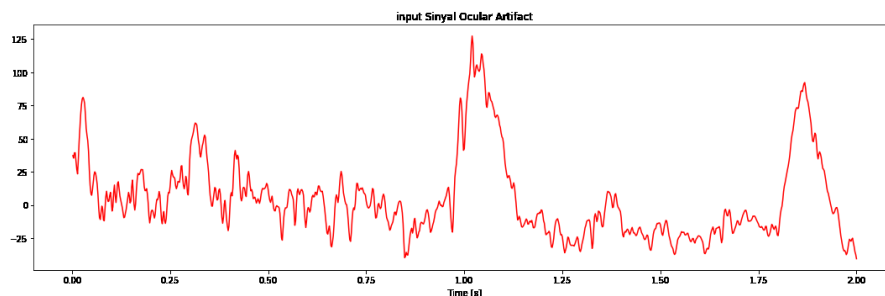


Gambar 4.2 Pola Sinyal Ictal Saat Terjadi Kejang Epilepsi

Secara klinis sinyal *ictal* menangkap letupan gelombang listrik otak saat terjadi kejang epilepsi. Kejang epilepsi terjadi akibat dari adanya ketidakseimbangan antara gelombang informasi yang dibawa oleh ujung akson terhadap gelombang informasi yang mampu ditangkap oleh dendrit dalam jaringan syaraf otak. Akibatnya letupan-letupan listrik yang tidak beraturan akan terjadi pada *neurotransmitter*. Letupan ini memiliki potensial listrik yang tinggi, yang selanjutnya ditangkap oleh elektroda dan dapat digambarkan dengan sinyal beramplitudo tinggi.

#### 4.1.2 Sinyal *Ocular Artifacts*

Sinyal *Ocular Artifact (OA)* adalah salah satu sinyal yang tidak diharapkan saat perekaman *EEG*. Sinyal ini memiliki ciri-ciri frekuensi yang tinggi, sehingga amplitudonya lebih kecil daripada sinyal *ictal*. Pada saat-saat tertentu akan mengalami pola penurunan atau kenaikan sesaat yang secara visual tampak lebih lancip dibandingkan saat kenaikan atau penurunan sinyal *ictal*, seperti yang dapat dilihat pada Gambar 4.3. Sinyal ini muncul ketika perekaman *EEG*, akibat dari pergerakan berkedip yang berulang secara cepat dan terjadi dalam durasi tertentu.

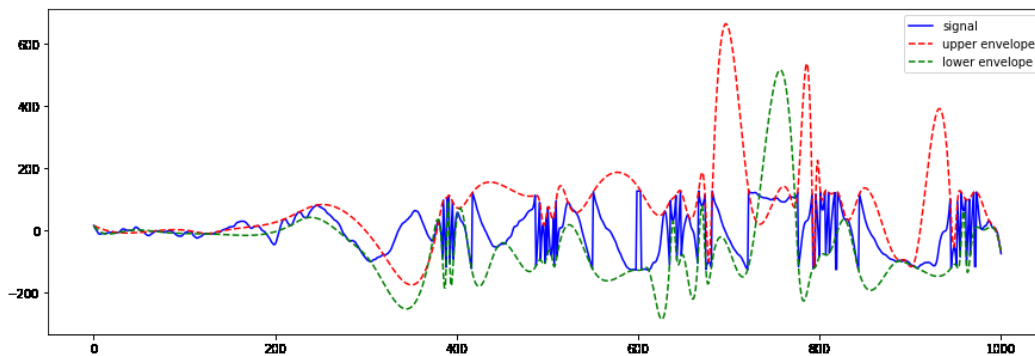


**Gambar 4.3** Pola Sinyal *Ocular Artifacts* yang Terekam *EEG*

Pola sinyal turun lancip pada Gambar 4.3 terjadi akibat gerakan berkedip pasien, di mana saat berkedip bagian hitam mata akan berubah ke arah atas. Warna hitam mata akan ditangkap elektroda sebagai sinyal positif, sehingga pola akan naik. Ilustrasi yang dapat dilihat dari Gambar 4.3 adalah terdapat empat kali kenaikan tajam, yang menunjukkan terjadi enam kali kedipan dalam dua detik saat perekaman. Adanya sinyal ini dapat meningkatkan resiko kesalahan diagnosa pasien epilepsi, karena dapat mempengaruhi frekuensi, amplitudo dan pola sinyal sinyal *EEG* murni.

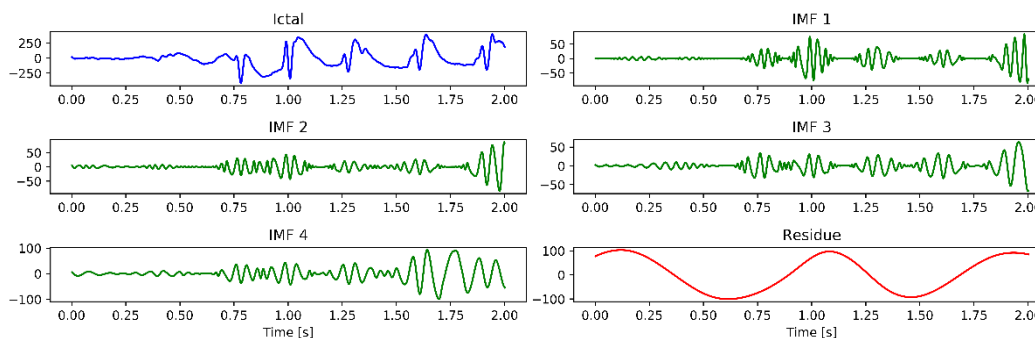
## 4.2 Dekomposisi Sinyal menggunakan *Empirical Mode Decomposition*

*Empirical Mode Decomposition* (EMD) digunakan untuk mendekomposisi sinyal menjadi *IMF*. Dekomposisi dilakukan disetiap potong sinyal, baik sinyal *ictal* maupun *artifact*. Masing-masing sinyal dicari nilai *extrema* menggunakan *cubic spline interpolation* dan menandainya pada masing-masing *extrema* (*local minima* dan *local maxima*). Pada sub-bab ini akan ditampilkan satu potong sinyal *ictal* dan satu potong sinyal *OA* yang akan didekomposisi menggunakan *EMD*. Dekomposisi sinyal *ictal* pada Gambar 4.2 menghasilkan titik-titik *extrema* seperti pada Gambar 4.4.



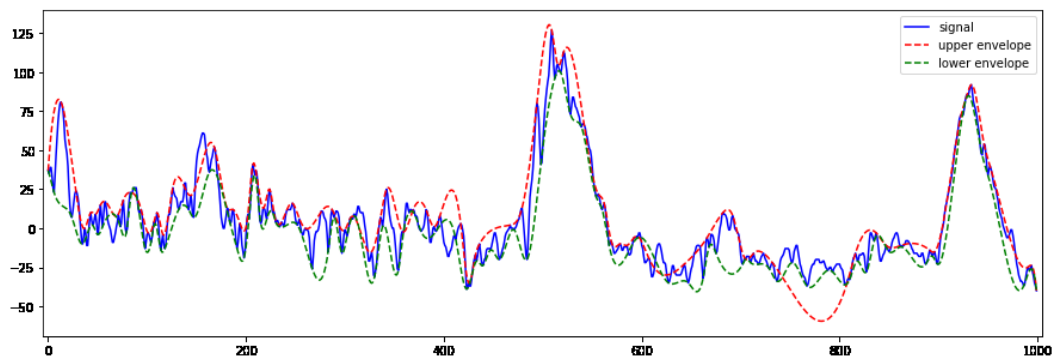
**Gambar 4.4** Titik *Extrema* Hasil *Cubic Spline Interpolation* dari Satu Potong Sinyal *Ictal*

*Extrema* pada Gambar 4.4 merupakan hasil interpolasi menggunakan *cubic spline* terhadap potongan pertama sinyal *ictal*. Plot sinyal dibawah sinyal *ictal* asli disebut dengan *local minima*, sedangkan plot yang berada di atas sinyal *ictal* disebut *local maxima*. Hasil penjumlahan *local minima* dan *local minima* lalu dibagi dua akan menghasilkan nilai *average*. Selisih antara nilai sinyal asli dengan *average* akan menghasilkan *IMF*. Langkah ini berulang, hingga mendapatkan residual monoton seperti yang ditampilkan pada Gambar 4.5.



**Gambar 4.5** *IMF* dan Residual dari Proses *EMD* Potongan Pertama Sinyal *Ictal*

Gambar 4.5 menunjukkan bahwa terdapat empat *IMF* dihasilkan dari proses *EMD* pada potongan pertama sinyal *ictal*. *IMF* pertama (*IMF* 1) memiliki nilai amplitudo pada kisaran amplitudo 50 mV. Pada detik-detik awal hingga mendekati detik ke 0,7, *IMF* 1 memiliki nilai amplitudo yang rendah, namun memiliki frekuensi yang tinggi. Pada beberapa saat amplitudo mengalami kenaikan, dan secara bersamaan terjadi pengurangan frekuensi sinyal. Saat terjadinya kenaikan amplitudo adalah saat bersamaan dengan naiknya amplitudo pada sinyal asli. *IMF* 3 dan *IMF* 4 memiliki ciri-ciri yang hampir sama dengan *IMF* 1, dengan amplitudo maksimal berada pada absolute 10 mV.

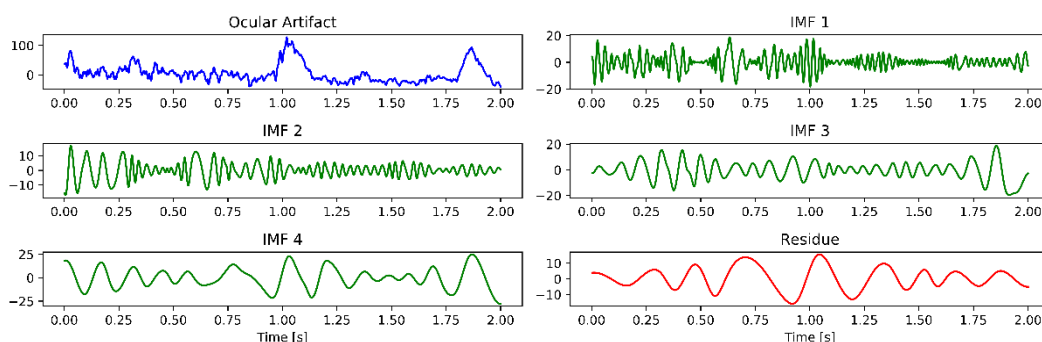


**Gambar 4.6** Titik *Extrema* hasil *Cubic Spline Interpolation* dari Satu Potong Sinyal *OA*

Plot pada Gambar 4.6 merupakan salah satu hasil interpolasi menggunakan *cubic spline* terhadap satu potong sinyal *OA* yang memiliki amplitudo 125 mV. Gambar 4.7 merupakan *IMF* yang dihasilkan dari *EMD* terhadap sinyal *OA*. *IMF* pertama menghasilkan frekuensi yang tinggi dengan amplitudo berada disekitar absolute 20 mV, sehingga dilanjutkan ketahap iterasi yang ke-dua. Iterasi ke-dua menghasilkan *IMF* yang kedua (*IMF* 2). *IMF* 2 memiliki karakteristik amplitudo dikisaran absolute 10 mV dengan frekuensi yang lebih kecil daripada *IMF* 1. Iterasi selanjutnya menghasilkan *IMF* 3 dan *IMF* 4 yang memiliki amplitudo dan frekuensi lebih kecil daripada *IMF* 2. Proses *EMD* berakhir ketika mencapai jumlah maksimal dari yang ditentukan dalam penelitian ini yaitu 4 *IMF*. Langkah dekomposisi ini dilakukan terhadap semua sinyal *OA* dan sinyal *ictal* yang dijadikan sebagai data dalam penelitian ini.

Plot *IMF* hasil *EMD* pada Gambar 4.5 dan Gambar 4.7 secara umum menunjukkan bahwa karakteristik yang membedakan antara sinyal *ictal* dan sinyal *OA* adalah nilai frekuensi dan amplitudonya. Amplitudo sinyal *ictal* cenderung

lebih besar namun memiliki frekuensi lebih kecil daripada sinyal *OA*. Dengan adanya kondisi ini, maka pendekatan ekstraksi fitur yang memperhatikan kondisi amplitudo dan frekuensi diindikasikan mampu menunjukkan perbedaan ciri khas dari sinyal *ictal* dan sinyal *OA*.



**Gambar 4.7** IMF dan Residual dari Proses EMD Potongan Pertama Sinyal *OA*

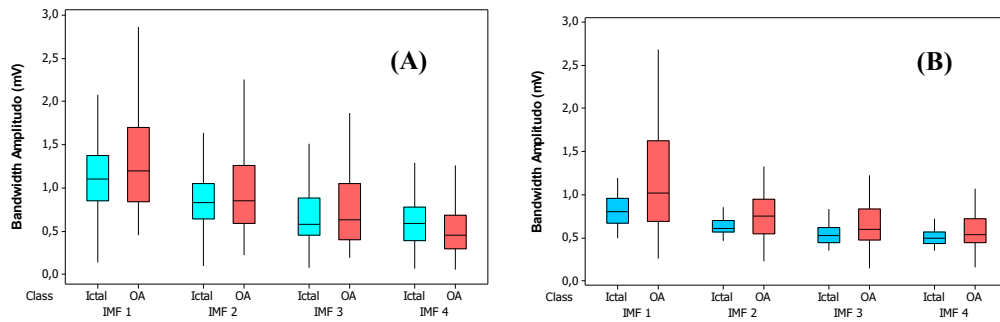
### 4.3 Ekstraksi Fitur

Setelah mendapatkan nilai *IMF* dari proses dekomposisi, langkah selanjutnya adalah membentuk fitur-fitur yang dapat digunakan sebagai *input* klasifikasi. Fitur-fitur yang dibentuk merupakan fitur-fitur *time-frequency* meliputi  $B_{AM}$ ,  $B_{FM}$ ,  $fr$ ,  $fd$ ,  $f_R$  dan rasio parameternya

#### 4.3.1 Bandwidth Amplitude ( $B_{AM}$ ) Sinyal *Ictal* dan Sinyal *OA*

$B_{AM}$  merupakan nilai yang menunjukkan jangkauan amplitudo suatu sinyal.  $B_{AM}$  dapat dihitung menggunakan Persamaan (2.18) atau dapat juga dihitung menggunakan *software python* menggunakan *script* pada Lampiran 10. Plot pada Gambar 4.8 merupakan nilai  $B_{AM}$  hasil ekstraksi terhadap semua potongan sinyal dengan durasi potongan 2 detik (A) dan durasi potongan 5 detik (B). Secara visual pada *boxplot* data potongan 2 detik dapat dilihat bahwa amplitudo sinyal *ictal* memiliki *range* lebih rendah daripada sinyal *OA*, namun memiliki nilai tengah atau rata-rata hampir sama serta saling berhimpitan di setiap *IMF*. Nilai dari hasil ekstraksi  $B_{AM}$  untuk data dengan durasi pemotongan 2 detik dapat dilihat pada Lampiran 2, sedangkan untuk  $B_{AM}$  untuk data dengan durasi pemotongan 5 detik dapat dilihat pada Lampiran 3.



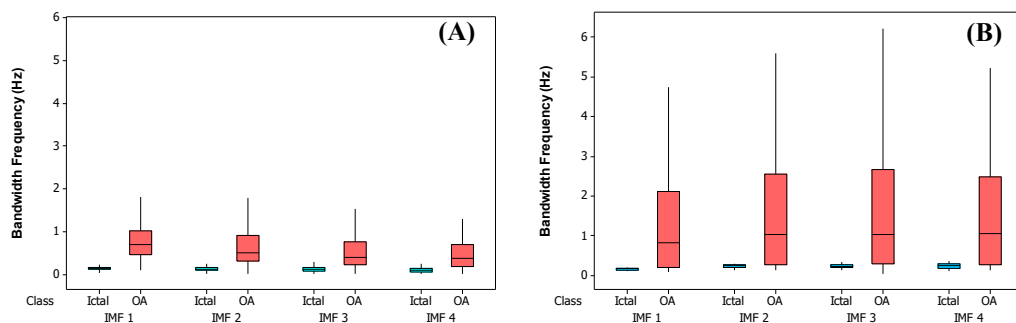


**Gambar 4.8** *Boxplot*  $B_{AM}$  dari  $IMF1$ - $IMF4$  Dari Data potongan 2 detik (A) dan Dari Data Potongan 5 Detik (B)

Hal yang sama juga dapat dilihat dari *boxplot* pada Gambar 4.8 (B), dimana  $B_{AM}$  sinyal *ictal* dan sinyal *OA* pada potongan 5 detik juga saling berpotongan di semua *IMF*. Perbedaan utama yang dapat dilihat pada potongan 2 detik dan 5 detik hanyalah penurunan nilai  $B_{AM}$  nya. Tampak pada *boxplot* sinyal *ictal* pada potongan 5 detik lebih rendah daripada potongan 2 detik dikarenakan amplitudo cenderung menurun pada pengamatan dengan durasi yang lebih lama.

### 4.3.2 Bandwidth Frequency ( $B_{FM}$ ) Sinyal *Ictal* dan Sinyal *OA*

$B_{FM}$  merupakan nilai yang menunjukkan jangkauan frekuensi suatu sinyal.  $B_{FM}$  dapat dihitung menggunakan Persamaan (2.19) atau dapat juga dihitung dengan *software python 3* menggunakan script pada Lampiran 10 dan Lampiran 11. Pada penelitian ini,  $B_{FM}$  digunakan sebagai fitur untuk membedakan sinyal *ictal* dan sinyal *OA* berdasarkan rentang frekuensinya.



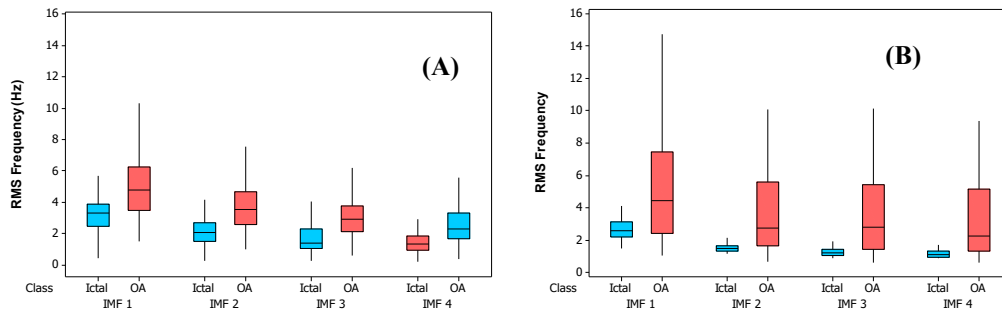
**Gambar 4.9** *Boxplot*  $B_{FM}$  Dari  $IMF1$ - $IMF4$  Dari Data Potongan 2 detik (A) dan Dari Data Potongan 5 Detik (B)

Plot pada Gambar 4.9 merupakan nilai  $B_{FM}$  hasil ekstraksi pada semua potongan sinyal. *Boxplot*  $B_{FM}$  pada durasi pemotongan 2 detik maupun 5 detik

memiliki perbedaan yang cukup jauh antara *boxplot* sinyal *ictal* dan sinyal *OA*. Meskipun masih saling berpotongan, akan tetapi range-nya masih berada di bawah nilai kuartil yang pertama, terutama  $B_{FM}$  pada *IMF* 1 dan *IMF* 2. Berdasarkan hal tersebut, maka dapat diindikasikan fitur  $B_{FM}$  dapat digunakan sebagai *input* yang baik pada proses klasifikasi.

### 4.3.3 RMS Frequency (*fr*) Sinyal *Ictal* dan Sinyal *OA*

*Root Mean Square (RMS) Frequency* atau dilambangkan menjadi *fr* dapat didefinisikan sebagai nilai rata-rata frekuensi yang bervariasi dari hasil *instantaneous frequency* atau frekuensi sesaat dalam satu siklus tertentu. Dalam penelitian ini, *fr* dapat digunakan untuk menunjukkan akar rata-rata dari frekuensi dalam satu potong sinyal, baik sinyal dari hasil pemotongan selama 2 detik maupun 5 detik.

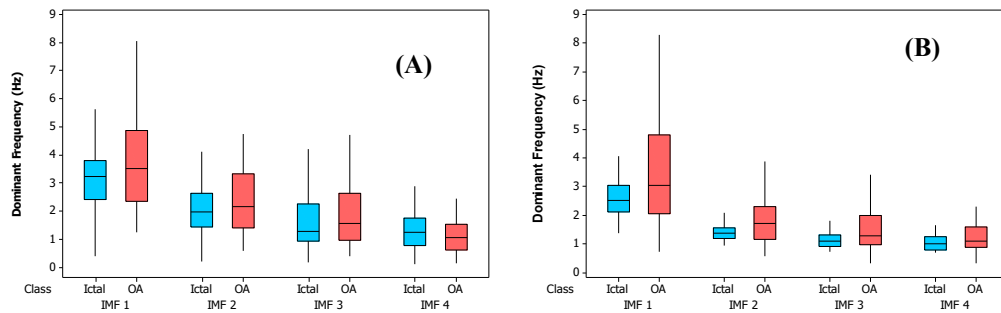


**Gambar 4.10** *Boxplot fr* dari *IMF*1-*IMF*4 Dari Data Potongan 2 Detik (A) dan Dari Data Potongan 5 Detik (B)

*Boxplot fr* pada Gambar 4.10 menunjukkan pada kedua potongan, memiliki rentang yang saling beririsan. Nilai *fr* sinyal *OA* pada potongan 5 detik memiliki rentang lebih tinggi daripada *fr* sinyal *ictal*. Hal ini menunjukkan bahwa meskipun masih pada rentang interval yang sama, frekuensi sinyal *OA* lebih tinggi daripada sinyal *ictal*.

### 4.3.4 Dominant Frequency (*fd*) Sinyal *Ictal* dan Sinyal *OA*

Terdapat beberapa situasi ketika suatu data atau suatu sinyal menunjukkan perilaku/pola periodik karena satu frekuensi. Frekuensi inilah yang disebut dengan *dominant frequency*, yang membawa energi maksimum di antara semua frekuensi yang ditemukan dalam suatu spektrum sinyal.

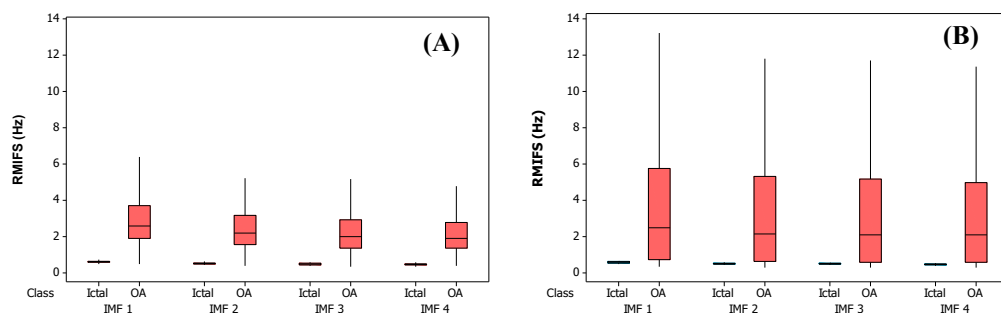


**Gambar 4.11** *Boxplot*  $fd$  Dari  $IMF1$ - $IMF4$  Dari Data Potongan 2 Detik (A) dan Dari Data Potongan 5 Detik (B)

*Boxplot* di Gambar 4.11 (A) menunjukkan bahwa pada potongan setiap 2 detik,  $fd$  antara sinyal *ictal* dan sinyal *OA* memiliki kemiripan di  $IMF 1$  hingga  $IMF 4$ . Kondisi ini mengindasikan fitur  $fd$  kurang baik untuk digunakan sebagai *input* dalam proses klasifikasi, karena tidak dapat menjelaskan adanya perbedaan ciri atau karakteristik yang unik dari masing masing sinyal. Kondisi yang sama juga dapat dilihat dari Gambar 4.11 (B) yang merupakan  $fd$  dari potongan setiap 5 detik. Meskipun dari *boxplot* tampak terdapat perbedaan *range*, akan tetapi setiap pasang *boxplot*  $fd$  sinyal *ictal* maupun sinyal *OA* saling berpotongan di semua  $IMF$ .

#### 4.3.5 $RMIFS (f_R)$ Sinyal *Ictal* dan Sinyal *OA*

$RMIFS$  merupakan akar dari rata-rata kuadrat nilai *instantaneous frequency* atau frekuensi sesaat. Pada perekaman *EEG*, frekuensi sesaat memberikan informasi tentang perilaku atau sifat gelombang otak yang mempengaruhi perubahan frekuensi akibat adanya suatu aktifitas tertentu.



**Gambar 4.12** *Boxplot*  $RMIFS$  dari  $IMF1$ - $IMF4$  Dari Data Potongan 2 Detik (A) dan Dari Data Potongan 5 Detik (B)

Frekuensi sesaat memiliki rentang frekuensi dari (-) Frekuensi *Nyquist* sampai (+) Frekuensi *Nyquist*, meskipun sebagian besar frekuensinya positif. Adanya kemungkinan frekuensi negatif inilah yang membuat pendekatan frekuensi sesaat lebih mudah dan rasional menggunakan *RMIFS*.

*Boxplot* Gambar 4.12 menunjukkan nilai *RMIFS* pada sinyal potongan 2 detik (A) dan sinyal potongan 5 detik (B) memiliki karakter yang hampir sama. Secara visual dapat diindikasikan bahwa fitur *RMIFS* dari sinyal *ictal* dan sinyal *OA* memiliki karakteristik yang berbeda. Adanya perbedaan karakteristik ini juga mengindikasikan *RMIFS* dapat membedakan sinyal *ictal* dan *OA* dengan baik pada proses klasifikasi. Nilai *RMIFS* sinyal *ictal* berada dibawah 1 Hz, sedangkan *RMIFS* sinyal *OA* memiliki nilai dengan range jauh lebih besar hingga 6 Hz pada potongan 2 detik dan mencapai 12 Hz pada potongan 5 detik.

Ekstraksi fitur untuk sinyal *ictal* dan sinyal *OA* menghasilkan berbagai macam fitur seperti yang telah dijelaskan pada pembahasan sebelumnya. Fitur yang baik untuk klasifikasi adalah fitur-fitur yang dapat membedakan kondisi di dalam kelas atau respon, dengan kata lain dapat mencirikan kondisi masing-masing kelas. Ditinjau dengan menggunakan *boxplot* yang dapat menunjukkan jangkauan nilai fitur-fitur dari masing-masing kelas, maka dapat diketahui bahwa fitur  $B_{FM}$ , dan *RMIFS* dapat menggambarkan ciri khas antara sinyal *ictal* dan *OA*. Setelah mendapatkan fitur-fitur dari ekstraksi fitur, maka selanjutnya adalah melakukan klasifikasi menggunakan metode *SVM*. Kernel yang digunakan adalah kernel linier, RBF dan *polynomial*, dengan nilai parameter yang digunakan didapat dari hasil *tunning* parameter menggunakan metode *gridsearch*. dilakukan untuk mendapatkan nilai parameter terbaik dari kernel yang digunakan.

#### **4.4 Klasifikasi Sinyal *Ictal* dan *Ocular Artifacts***

Klasifikasi kedua sinyal dapat dilakukan setelah diperoleh semua fitur dari proses ekstraksi fitur. Metode klasifikasi yang digunakan dalam penelitian ini adalah *SVM* dengan kernel *RBF*, *linear*, dan *polynomial*. Klasifikasi dilakukan dengan beberapa kombinasi fitur dengan nilai parameter yang digunakan merupakan hasil tuning menggunakan metode *gridsearch*. Hasil *tunning* parameter

*SVM* pada penelitian ini dilakukan dengan semua kombinasi dari fitur-fitur seperti yang dijelaskan pada Tabel 3.4

#### 4.4.1 Klasifikasi Sinyal *Ictal* dan *Ocular Artifacts* Skenario Pertama

Skenario pertama klasifikasi sinyal *ictal* dan sinyal *ocular artifacts* menggunakan *input*  $B_{AM}$  dan  $B_{FM}$ . Karena jumlah *IMF* sebanyak 4, maka terdapat 8 fitur yang digunakan pada skenario pertama. Semua kemungkinan kombinasi fitur dicari parameter optimum di masing-masing kernel dan dari nilai parameter tersebut dicari kebaikan klasifikasi dengan menghitung nilai akurasi, sensitifity, spesificity dan *AUC* data *testing*, dengan *fold* 10. Pada pembahasan ini, hanya ditampilkan nilai akurasi dan *AUC* dari *input* fitur terbaik di setiap *IMF* dan masing-masing kernel seperti yang ditampilkan pada Tabel 4.1. Sedangkan untuk detail *tunning* parameter dan kebaikan klasifikasi untuk skenario pertama secara lengkap dapat dilihat pada Lampiran 16.

**Tabel 4.1** Kebaikan Klasifikasi Skenario Pertama untuk Data Potongan 2 detik

<i>IMF</i>	Fitur	Parameter	Kernel	Akurasi	<i>AUC</i>
<i>IMF</i> 1	<i>X2.1</i>	$C=100$	Linier	94.15	94.28
	<i>X2.1</i>	$C=10^3; \sigma=100$	RBF	<b>95.49</b>	<b>95.5</b>
	<i>X2.1</i>	$C=100; P=1$	<i>Polynomial</i>	94.15	94.28
<i>IMF</i> 2	<i>X1.2, X2.2</i>	$C=100$	Linier	<b>91.05</b>	<b>91.49</b>
	<i>X1.2, X2.2</i>	$C=1000; \sigma=0.1$	RBF	89.74	90.31
	<i>X1.2, X2.2</i>	$C=100; P=1$	<i>Polynomial</i>	<b>91.05</b>	<b>91.49</b>
<i>IMF</i> 3	<i>X1.3, X2.3</i>	$C=10$	Linier	86.56	87.39
	<i>X1.3, X2.3</i>	$C=10; \sigma=0.1$	RBF	<b>86.58</b>	<b>87.51</b>
	<i>X1.3, X2.3</i>	$C=10; P=1$	<i>Polynomial</i>	86.56	87.39
<i>IMF</i> 4	<i>X2.4</i>	$C=100$	Linier	<b>81.70</b>	<b>82.63</b>
	<i>X2.4</i>	$C=1; \sigma=10$	RBF	81.68	82.83
	<i>X2.4</i>	$C=100; P=1$	<i>Polynomial</i>	<b>81.70</b>	<b>82.63</b>
<i>IMF</i> 1-4	<i>X1.1, X1.2, X1.3, X1.4</i>	$C=10$	Linier	57,75	54,10
		$C=1000; \sigma=0,1$	RBF	70,20	69,78
		$C=10; P=1$	<i>Polynomial</i>	57,75	54,10
<i>IMF</i> 1-4	<i>X2.1, X2.2, X2.3, X2.4</i>	$C=100$	Linier	94,58	94,75
		$C=1; \sigma=100$	RBF	94,62	94,42
		$C=100; P=1$	<i>Polynomial</i>	94,58	94,75
<i>IMF</i> 1-4	<i>X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4</i>	$C=100$	Linier	96,40	96,45
		$C=1000; \sigma=0,01$	RBF	96,86	96,83
		$C=100; P=1$	<i>Polynomial</i>	96,40	96,45

Hasil klasifikasi skenario pertama untuk data potongan 2 detik pada Tabel 4.1 menunjukkan bahwa klasifikasi sinyal *ictal* dan sinyal *OA* terbaik diperoleh dari *input* semua fitur *Bandwidth* yang ada dengan nilai akurasi 96,4 dan *AUC* sebesar

96.83, dengan kernel *RBF* ( $C=1000$ ;  $\sigma=0.01$ ). Namun, apabila mempertimbangkan efisiensi, maka pilihan terbaik terdapat pada *input* fitur  $B_{FM}$  dari *IMF* 1, dengan nilai akurasi 95,49 dan *AUC* sebesar 95,5, dengan kernel *RBF* ( $C=10$ ;  $\sigma=100$ ). Hal ini didukung dari nilai kebaikan klasifikasi fitur *IMF* 1 yang selalu lebih baik daripada fitur *IMF* 2, *IMF* 3 dan *IMF* 4. Penggunaan *input* fitur  $B_{AM}$  dan  $B_{FM}$  juga digunakan pada data durasi pemotongan 5 detik, dengan hasil kebaikan klasifikasi pada Tabel 4.2.

**Tabel 4.2** Kebaikan Klasifikasi Skenario Pertama untuk Data Potongan 5 detik

<i>IMF</i>	Fitur	Parameter	Kernel	Akurasi	<i>AUC</i>
<i>IMF</i> 1	<i>X2.1</i>	$C=10$	Linier	80.19	84.38
	<i>X2.1</i>	<b><math>C=1</math>; <math>\sigma=1000</math></b>	<b>RBF</b>	<b>83.27</b>	<b>86.5</b>
	<i>X2.1</i>	$C=10$ ; $P=1$	<i>Polynomial</i>	80.19	84.38
<i>IMF</i> 2	<i>X1.2, X2.2</i>	$C=100$	Linier	80.19	84.38
	<i>X1.2, X2.2</i>	$C=10$ ; $\sigma=100$	RBF	83.26	85.5
	<i>X1.2, X2.2</i>	$C=100$ ; $P=1$	<i>Polynomial</i>	80.19	84.38
<i>IMF</i> 3	<i>X1.3, X2.3</i>	$C=100$	Linier	78.59	83.12
	<i>X1.3, X2.3</i>	$C=1$ ; $\sigma=100$	RBF	83.68	85
	<i>X1.3, X2.3</i>	$C=1000$ ; $P=3$	<i>Polynomial</i>	80.13	84.38
<i>IMF</i> 4	<i>X2.4</i>	$C=100$	Linier	77.76	82.5
	<i>X2.4</i>	$C=1$ ; $\sigma=1000$	RBF	80.90	84.25
	<i>X2.4</i>	$C=100$ ; $P=1$	<i>Polynomial</i>	77.76	82.50
<i>IMF</i> 1-4	<i>X1.1, X1.2, X1.3, X1.4</i>	$C=10$	Linier	57,75	54,10
		$C=1000$ ; $\sigma=0,1$	RBF	70,20	69,78
		$C=10$ ; $P=1$	<i>Polynomial</i>	57,75	54,10
<i>IMF</i> 1-4	<i>X2.1, X2.2, X2.3, X2.4</i>	$C=10$	Linier	80,19	84,38
		$C=1$ ; $\sigma=100$	RBF	82,21	82,62
		$C=10$ ; $P=1$	<i>Polynomial</i>	80,19	84,38
<i>IMF</i> 1-4	<i>X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4,</i>	$C=100$	Linier	81,03	84,25
		$C=0,1$ ; $\sigma=1$	RBF	81,67	85,62
		$C=1000$ ; $P=1$	<i>Polynomial</i>	81,03	84,25

Kebaikan klasifikasi skenario pertama untuk data dengan durasi pemotongan 5 detik pada Tabel 4.2 menunjukkan bahwa *Input* fitur  $B_{FM}$  dari *IMF* 1 memberikan hasil klasifikasi terbaik.. Kernel *SVM* yang digunakan adalah *RBF* ( $C=1$ ;  $\sigma=1000$ ) dengan nilai akurasi dan *AUC* sebesar 83,27 dan 86,5. Namun jika dibandingkan dengan akurasi dan *AUC* pada data pemotongan 2 detik masih lebih baik. Dengan demikian dapat diketahui bahwa *input* terbaik untuk skenario pertama adalah menggunakan  $B_{FM}$  dari *IMF* 1, dengan metode klasifikasi *SVM* dengan kernel *RBF*. Hal ini sesuai dengan *boxplot*  $B_{FM}$  pada Gambar 4.9 yang secara visual mampu membedakan karakteristik antara sinyal *ictal* dan sinyal *OA*.

#### 4.4.2 Klasifikasi Sinyal *Ictal* dan *Ocular Artifacts* Skenario Kedua

Skenario klasifikasi sinyal *Ictal* dan sinyal *Ocular Artifact* yang kedua menggunakan *input Time frequency feature*. Jumlah *IMF* sebanyak 4, sehingga terdapat 24 fitur dalam skenario kedua ini. Namun tidak semua hasil klasifikasi ditampilkan pada bagian ini. Hanya hasil klasifikasi terbaik yang ditampilkan, sedangkan hasil tuning dan klasifikasi secara lengkap ditampilkan pada Lampiran 18 untuk potongan 2 detik dan Lampiran 19 untuk potongan 5 detik. Semua kemungkinan kombinasi fitur dicari parameter optimum di masing-masing kernel dan dari nilai parameter tersebut dicari kebaikan klasifikasi dengan menghitung nilai akurasi, sensitifity, spesificity dan *AUC* data testing, dengan *fold* 10. Hasil nilai akurasi dan *AUC* dari *input* fitur *time frequency* terbaik di setiap *IMF* dan masing-masing kernel ditampilkan pada Tabel 4.3.

**Tabel 4.3** Kebaikan Klasifikasi Skenario Kedua untuk Data Potongan 2 detik

<i>IMF</i>	Fitur	Parameter	Kernel	Akurasi	<i>AUC</i>
<i>IMF</i> 1	X7.1	C=0.1	Linier	97.31	97.45
	X7.1	C=0.1'; $\sigma=1$	RBF	97.75	97.83
	X7.1	C=0.1; P=1	<i>Polynomial</i>	97.31	97.45
<i>IMF</i> 2	X7.2	C=10	Linier	97.77	97.75
	X7.2	C=10; $\sigma=0.1$	RBF	97.77	97.75
	X7.2	C=10; P=1	<i>Polynomial</i>	97.77	97.75
<i>IMF</i> 3	X7.3	C=0.1	Linier	97.31	97.45
	X7.3	C=0.1; $\sigma=10$	RBF	97.77	97.75
	X7.3	C=0.1; P=1	<i>Polynomial</i>	97.31	97.45
<i>IMF</i> 4	X7.4	C=10	Linier	97.31	97.45
	X7.4	C=100; $\sigma=0.1$	RBF	97.31	97.45
	X7.4	C=10; P=1	<i>Polynomial</i>	97.31	97.45
<i>IMF</i> 1-4	X7.1, X7.2, X7.3, X7.4	C=1000	Linier	98,22	98,28
		C=1000; $\sigma=1$	RBF	96.03	95.78
		<b>C=100; P=1</b>	<b><i>Polynomial</i></b>	<b>98.68</b>	<b>98.7</b>

Hasil klasifikasi skenario pertama untuk data potongan 2 detik pada Tabel 4.3 menunjukkan bahwa klasifikasi sinyal *ictal* dan sinyal *OA* terbaik diperoleh dari *input RMIFS* gabungan *IMF* 1 hingga *IMF* 4 dengan nilai akurasi 98,68 dan *AUC* sebesar 98,70, dengan kernel linier (C=1000). *Input* terbaik didominasi oleh *RMIFS* yang mengungguli fitur-fitur *time frequency* lainnya. Hal ini sesuai dengan *boxplot*

*RMIFS* yang mampu menunjukkan keunikan karakteristik dari sinyal *ictal* dan sinyal *OA* yang dapat dilihat pada Gambar 4.12. Pada skenario klasifikasi kedua ini juga diterapkan pada data dengan durasi potongan 5 detik. Hasil tuning parameter kernel terbaik dan *input* terbaik dari setiap *IMF* ditampilkan pada Tabel 4.4.

Kebaikan klasifikasi skenario kedua untuk data dengan durasi pemotongan 5 detik pada Tabel 4.4 menunjukkan bahwa *Input* fitur *RMIFS* dari *IMF* 1 hingga *IMF* 4 mendominasi sebagai *input* terbaik dalam di skenario kedua, baik untuk data dengan durasi pemotongan 2 detik ataupun 5 detik. Hasil klasifikasi terbaik untuk data potongan 5 detik pada skenario kedua diperoleh dari *input RMIFS* dengan kernel *RBF*, yang menghasilkan akurasi sebesar 85,71 dan *AUC* sebesar 88. Pada skenario kedua dapat diketahui bahwa klasifikasi dengan data potongan 2 detik lebih baik daripada data potongan 5 detik. Hal yang sama juga diperoleh dari klasifikasi pada skenario pertama,

**Tabel 4.4** Kebaikan Klasifikasi Skenario Kedua untuk Data Potongan 5 detik

<i>IMF</i>	Fitur	Parameter	Kernel	Akurasi	<i>AUC</i>
<i>IMF</i> 1	<i>X7.1</i>	C=10	Linier	80.96	85
	<i>X7.1</i>	C=0.1; $\sigma=100$	RBF	85.71	88
	<i>X7.1</i>	C=1; P=1	<i>Polynomial</i>	80.96	85
<i>IMF</i> 2	<i>X7.2</i>	C=10	Linier	80.96	85
	<i>X7.2</i>	C=0.1; $\sigma=100$	RBF	85.71	88
	<i>X7.2</i>	C=1; P=2	<i>Polynomial</i>	80.96	85
<i>IMF</i> 3	<i>X7.3</i>	C=1	Linier	80.96	85
	<i>X7.3</i>	C=1; $\sigma=1000$	RBF	85.71	88
	<i>X7.3</i>	C=1; P=1	<i>Polynomial</i>	80.96	85
<i>IMF</i> 4	<i>X7.4</i>	C=10	Linier	80.19	84.38
	<i>X7.4</i>	C=0.1; $\sigma=100$	RBF	85.71	88
	<i>X7.4</i>	C=10; P=1	<i>Polynomial</i>	80.19	84.38
<i>IMF</i> 1-4	<i>X7.1, X7.2, X7.3, X7.4</i>	C=1	Linier	80.96	85
		C=0.1; $\sigma=10$	RBF	84.87	87.38
		C=1; P=1	<i>Polynomial</i>	80.96	85



## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan analisis yang telah dilakukan, maka hasil penelitian ini dapat disimpulkan sebagai berikut.

1. Hasil dekomposisi sinyal *ictal* dan sinyal *ocular artifacts* menghasilkan 4 *IMF* yang dapat diekstraksi menjadi fitur-fitur klasifikasi. Fitur terbaik yang dapat digunakan adalah *Bandwidth Frequency* dan *RMIFS*.
2. Hasil klasifikasi terbaik pada skenario pertama (*input* fitur *bandwidth*) dapat disimpulkan bahwa  $B_{FM}$  pada *IMF* 1 menghasilkan klasifikasi terbaik. Sedangkan untuk skenario kedua (*input* fitur *time frequency*), fitur *RMIFS* menghasilkan klasifikasi terbaik.
3. Akurasi terbaik diperoleh dengan menggunakan metode *SVM* dengan kernel *Polynomial* ( $C=100$ ,  $P=1$ ) menggunakan *input RMIFS* dari semua *IMF*, yang dapat menghasilkan nilai akurasi sebesar 98,68% serta nilai *AUC* sebesar 98,7% dan pemotongan data terbaik dengan durasi 2 detik untuk setiap potongannya.

#### **5.2 Saran**

Saran yang dapat diberikan berdasarkan proses dan hasil analisis yang telah dilakukan adalah.

1. Pemotongan data dengan durasi lebih pendek pada kasus sinyal *ictal* dan sinyal *OA* dapat lebih baik dalam menangkap karakteristik dasar seperti frekuensi dari sinyal.
2. Klasifikasi menggunakan *SVM* memerlukan waktu yang lama pada saat *tunning* parameter. Pada penelitian selanjutnya diharapkan dapat memangkas waktu *tunning* jika ingin menggunakan *SVM* atau dengan metode lain yang lebih baik.

(Halaman ini sengaja dikosongkan)

## DAFTAR PUSTAKA

- Adeli, H., Zhou, Z. dan Dadmehr, N. (2003), "Analysis of EEG records in An Epileptic Patient Using Wavelet Transform", *Journal of Neuroscience Methods*, pp. 69-87, doi:10.1016/s0165-0270(02)00340-0.
- Alotaiby, T.N., Alshebeili, S.A., Alshawi, T., Ahmad, I. dan El-Samie, F.E.A. (2014), "EEG seizure detection and prediction algorithms: a survey", *EURASIP Journal on Advances in Signal Processing*, Vol. 183, doi:10.1186/1687-6180-2014-183, diakses dari: <http://asp.eurasipjournals.com/content/2014/1/183>.
- Bajaj, V. dan Pachori, R.B. (2012), "Classification of Seizure and Nonseizure EEG", *IEEE Transactions on Information Technology in Biomedicine*, Vol. 16, No. 6, pp. 1135-1142, doi:10.1109/TITB.2011.2181403, diakses dari: <https://ieeexplore.ieee.org/document/6111481>.
- Batuwata, R. dan Paladeva, V. (2013), "Class Imbalance Learning Methods for Support Vector Machine", dalam *Imbalance Learning: Foundation, Algorithms, and Applications*, Haibo He, d.M.Y. , John Wiley & Sons, Berlin, pp. 83-99.
- Boser, B.E., Guyon, I.M. dan Vapnik, V.N. (1992), "A Training Algorithm for Optimal Margin Classifiers", *5th Annual Workshop on Computational Learning Theory (COLT 1992)*, Association for Computing Machinery, Pittsburgh , pp. 144-152.
- Cheng, H.D., Shan, J., Guo, Y. dan Zhang, L. (2010), "Automated breast cancer detection and classification using ultrasound images: A Survey", *Pattern Recognition*, Vol. 43, pp. 299-317, doi:10.1016/j.patcog.2009.05.012, diakses dari: <https://www.sciencedirect.com/science/article/pii/S0031320309002027>.
- Cohen, L. (1995), *Time-Frequency Analysis*, Prentice-Hall PTR, New Jersey.
- Cohen, L. dan Lee, C. (1988), "Instantaneous Frequency, Its Standard Deviation and Multicomponent Signals", *Advanced Algorithms and Architectures for Signal Processing III*, Vol. 975, pp. 186-208, doi:10.1117/12.948504.
- Cohen, L. dan Lee, C. (1990), "Instantaneous Bandwidth For Signals and Spectrogram", *International Conference on Acoustics, Speech, and Signal Processing* , Vol. ICASSP-90, pp. 2451-2454, doi:10.1109/ICASSP.1990.116086 , diakses dari: <https://ieeexplore.ieee.org/abstract/document/116086>.

- Eadie, J. (2012), "Shortcomings In The Current Treatment of Epilepsy", *Epilepsy*, pp. 1419–1427.
- Fisher, R.S., Acevedo, C., Arzimanoglou, A., Bogacz, A., Cross, H., Elger, C.E., Engel Jr, J., Forsgren, L., French, A.J., Glynn, M., Hesdorffer, D.c., Lee, B.I., Mathern, G.W., Moshe, S.L., Perucca, E., Scheffer, I.E., Tomson, T., Watanabe, M. dan Wiebe, S. (2014), "A Practical Clinical Definition of Epilepsy", *International League Against Epilepsy*, pp. 475-482, doi:10.1111/epi.12550.
- Fisher, R.S., Boas, W.V.E., Blume, W., Elger, C., Genton, P., Lee, P. dan Engel, J. (2005), "Epileptic Seizure and Epilepsy: Definitions Proposed by The International League Against Epilepsy (ILAE) and The International Bureau for Epilepsy (IBE)", *Epilepsia*, pp. 470-472, doi:10.1111/j.0013-9580.2005.66104.x.
- Fu, K., Qu, J., Chai, Y. dan Dong, Y. (2014), "Classification of seizure based on the time-frequency image of EEG signals using HHT and SVM", *Biomedical Signal Processing and Control*, Vol. 13, pp. 15-22, doi:10.1016/j.bspc.2014.03.007.
- Fu, K., Qu, J., Chai, Y. dan Zou, T. (2015), "Hilbert marginal spectrum analysis for automatic seizure detection in EEG Signals", *Biomedical Signal Processing and Control*, Vol. 18, pp. 179-185, doi:10.1016/j.bspc.2015.01.002.
- Gunn, S. (1998), *Support Vector Machines for Classification and Regression*, ISIS Technical Report, Image Speech and Intelligent System Group University of Southampton.
- Han, J., Kamber, M. dan Pei, J. (2012), *Data Mining : Concepts and Technique 3rd edition*, Morgan Kaufman, USA.
- Härdle, W.K., Prastyo, D.D. dan Hafner, C.M. (2014), "Support Vector Machines with Evolutionary Model Selection for Default Prediction", dalam *Applied Nonparametric and Semiparametric Enometrics and Statistics*, eds. S.R. Racine dan A. Ullah , Oxford University Press, New York, pp. 346-376.
- Huang, M.L., Hung, Y.H., Lee, W.M., Li, R.K. dan Jiang, B.R. (2014), ""SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier"", *The Scientific World Journal*, Vol. 2014, pp. 1-10, doi:http://dx.doi.org/10.1155/2014/795624.
- Huang, M.-L., Hung, Y.-H., Lee, W.M., Li, R.K. dan Jiang, B.-R. (2014), "SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier", *The Scientific World Journal* , pp. 1-10.

- Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C. dan Liu, H.H. (1998), "The Empirical Mode Decomposition and The Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis", *The Royal Society*, Vol. 454, pp. 903-995.
- Jafarifarmand, A. dan Badamchizadeh, M.A. (2013), "Artifacts Removal in EEG Signal Using a New Neural Network Enhanced Adaptive Filter", *Neurocomputing*, Vol. 103, pp. 222-231, doi:dx.doi.org/10.1016/j.neucom.2012.09.024.
- Kandel, E.R., Schwartz, J.H. dan Jassell, T.M. (2000), *Principles of Neural Science*, 4th ed., McGraw-Hill Companies, New York.
- Klem, G.H., Luders, H.O., Jasper, H.H. dan Elger, C. (1999), *The ten-twenty electrode system of the International Federation*, Elsevier Science B.V, Cleveland.
- Lakso, T.A., Bhagwat, J.G., Zou, K.H. dan Ohno-Machado, L. (2005), "The Use of Receiver Operating Characteristic Curves in Biomedical Informatics", *Journal of Biomedical Informatics*, Vol. 38, pp. 404-415, doi:10.1016/j.jbi.2005.02.008, diakses dari: <https://www.ncbi.nlm.nih.gov/pubmed/16198999>.
- Lemos, M.S. dan Fisch, B.J. (1991), "The weighted average reference montage", *Electroencephalography and clinical Neurophysiology*, pp. 361-370, doi:10.1016/0013-4694(91)90201-e.
- Li, S., Zhou, W., Yuan, Q., Geng, S. dan Cai, D. (2013), "Feature extraction and recognition of ictal EEG using EMD and SVM", *Computers in Biology and Medicine*, Vol. 43, No. 7, pp. 807-816, doi:10.1016/j.combiomed.2013.04.002.
- Loughlin, P.J. dan Tacer, B. (1997), "Comments on the Interpretation of Instantaneous Frequency", *IEEE Signal Processing Letters*, Vol. 4, No. 5, pp. 123-125, doi:10.1109/97.575553.
- Mahapatra, G. dan Horio, K. (2018), "Classification of ictal and interictal EEG using RMS frequency, Dominant Frequency, Root Mean Instantaneous Frequency Square and Their parameters Ratio", *Biomedical Signal Processing and Control*, Vol. 44, pp. 168-180, doi:10.1016/j.bspc.2018.04.007.
- Malmivuo, J. dan Plonsey, R. (1995), *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*, Oxford University Press, New York.
- Metz, C.E. (2006), "Receiver Operating Characteristic Analysis: A Tool for the Quantitative Evaluation of Observer Performance and Imaging Systems",

- American Collage of Radiology*, Vol. 3, No. 6, pp. 413-422, doi:10.1016/j.jacr.2006.02.021, diakses dari: <https://www.ncbi.nlm.nih.gov/pubmed/17412096>.
- Mutlu, A.Y. (2018), "Detection of Epileptic Dysfunctions in EEG signals Using Hilbert Vibration Decomposition", *Biomedical Signal Processing and Control*, Vol. 40, pp. 33-40, doi:10.1016/j.bspc.2017.08.023.
- Nihon Kohden Corp. (n.d.), *Electroencephalograph*, 91009200th ed., Tokyo.
- Purnami, S.W., Embong, A., Zain, J.M. dan Rahayu, S.P. (2009), "A New Smooth Support Vector Machine and Its Applications in Diabetes Disease Diagnosis", *Journal of Computer Science* 5, pp. 1003-1008.
- Putri, N.G.P. (2018), *Classification Seizure and Non Seizure Based on EEG Signal Using EMD and Bandwidth Feature Extraction*, Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.
- Riaz, F., Hassan, A., Niazi, I.K. dan Dremstrup, K. (2016), "EMD-Based Temporal and Spectral Features for the Classification of EEG Signals Using Supervised Learning", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 24, No. 1, pp. 28-35, doi:<https://ieeexplore.ieee.org/document/7119578>, diakses dari: [10.1109/TNSRE.2015.2441835](https://doi.org/10.1109/TNSRE.2015.2441835).
- Schölkopf, B. dan Smola, J. (2001), *Learning with Kernel: Support Vector Machines, Regularization, Optimization and Beyond*, The MIT Press, London.
- Schomer, D.L. dan Silva, F.H.L.d. (2011), *Niedermeyer's Electroencephalography Basic Principles, Clinical Applications, and Related Fields*, Lippincott Williams & Wilkins, a Wolters Kluwer Business, Philadelphia.
- Singh B dan Wagatsuma H (2018), "Two-stage wavelet shrinkage and EEG-EOG signal contamination model to realize quantitative validations for the artifact removal from multiresource biosignals", *Biomedical Signal Processing and Control*, Vol. 47, pp. 96-114, doi:<https://doi.org/10.1016/j.bspc.2018.08.014>.
- Siuly, S., Li, Y. dan Zhang, Y. (2016), *EEG Signal Analysis and Classification Techniques and Applications*, Springer, Melbourne.
- Smith, S. (2005), "EEG in the diagnosis, classification, and management of patients with epilepsy", *J Neurol Neurosurg Psychiatry*, Vol. 76, doi:10.1136/jnnp.2005.069245.
- Supriya, S., Siuly, S. dan Zhang, Y. (2016), "Automatic epilepsy detection from EEG introducing a new edge weight method in the complex network",

*Electronics Letters*, Vol. 52, No. 17, pp. 1430-1432, doi:10.1049/el.2016.1992.

Suyanto (2018), *Machine Learning*, Informatika, Bandung.

Vapnik, V. dan Cortes, C. (1995), ""Support Vector Networks"", *Machine Learning*, pp. 273-297.

Vijayakumar, S. dan Wu, S. (1999), "Sequential Support Vector Classifier and Regression", *Proc. International Conference on Soft Computing*, pp. 610-619.

World Health Organization (2005), *Mental Health*. [Online] diakses dari: [http://www.who.int/mental\\_health/neurology/epilepsy/regional\\_reports/en/](http://www.who.int/mental_health/neurology/epilepsy/regional_reports/en/).

World Health Organization (2018), *Epilepsy*. [Online] diakses dari: <http://www.who.int/en/news-room/fact-sheets/detail/epilepsy> [pada 28 April 2018].

(Halaman ini sengaja dikosongkan)



# LAMPIRAN

## Lampiran 1. Sertifikat Sidang Etik



**KOMITE ETIKA DAN HUKUM  
RUMAH SAKIT UNIVERSITAS AIRLANGGA**

**KETERANGAN LOLOS KAJI ETIK**  
No :104/KEH/2019

Komite Etika dan Hukum Rumah Sakit Universitas Airlangga dalam upaya melindungi hak asasi dan kesejahteraan subyek penelitian, telah mengkaji dengan teliti protokol penelitian yang berjudul:

*Klasifikasi Fase, Spektrum dan Respon Terapi pada Penderita Epilepsi menggunakan Support Vector Machine (SVM) Berdasarkan Sinyal Electro Encephalo Graphy (EEG) dan Electro Cardio Graphy (ECG)*

Peneliti Utama : Santi Wulan Purnami, S.Si.,M.Si.,Ph.D

Anggota Peneliti : 1. Dr.Diah P Wulandari,ST.,M.Sc  
2. Anda Isviana Juniani, ST.,MT  
3. Wardah Rahmatul Islamiyah,dr.,Sp.S

Institusi Penelitian : Rumah Sakit Universitas Airlangga

**DINYATAKAN LAIK ETIK.**

Surabaya, 22 Januari 2019



Ketua

Prof.Dr.Med. (K) dr. Erfan Kusuma, dr.Sp.F (K), DFM

**Lampiran 2.** Data Hasil Ekstraksi B<sub>AM</sub> dan B<sub>FM</sub> untuk sinyal dengan potongan 2 detik

No.	Bam1	Bam2	Bam3	Bam4	Bfm1	Bfm2	Bfm3	Bfm4	Class
0	1,179	2,062	0,595	0,495	0,071	0,0	0,128	0,12	<i>Ictal</i>
1	0,817	0,73	0,578	0,584	0,129	0,141	0,08	0,095	<i>Ictal</i>
2	0,636	0,742	0,512	2,609	0,425	0,318	0,412	0,049	<i>Ictal</i>
3	1,053	0,774	0,508	0,657	0,328	0,353	0,312	0,468	<i>Ictal</i>
4	1,749	2,151	1,513	1,092	0,112	0,045	0,039	0,016	<i>Ictal</i>
5	0,869	0,855	1,023	0,769	0,138	0,191	0,074	0,049	<i>Ictal</i>
6	1,206	0,521	0,876	1,291	0,135	0,154	0,016	0,015	<i>Ictal</i>
7	0,908	1,044	1,247	0,447	0,099	0,11	0,073	0,019	<i>Ictal</i>
8	1,012	0,894	1,349	0,441	0,117	0,003	0,02	0,033	<i>Ictal</i>
9	0,947	1,542	0,815	0,448	0,194	0,119	0,092	0,192	<i>Ictal</i>
10	1,259	0,58	0,385	0,74	0,151	0,222	0,26	0,123	<i>Ictal</i>
∴	∴	∴	∴	∴	∴	∴	∴	∴	∴
203	4,166	3,929	1,131	0,427	0,531	0,441	0,184	0,596	OA
204	1,834	1,221	1,623	0,29	0,6	0,911	0,358	0,907	OA
205	1,61	0,939	0,865	0,633	0,268	0,458	0,285	0,098	OA
206	1,61	0,939	0,865	0,633	0,268	0,458	0,285	0,098	OA
207	1,249	0,711	1,17	0,298	0,668	0,57	0,194	0,65	OA
208	0,995	0,552	0,35	0,317	0,753	0,544	0,848	0,679	OA
209	3,415	2,61	0,793	0,594	0,395	0,247	0,196	0,116	OA
210	1,373	0,651	0,54	0,142	0,437	0,323	0,02	0,11	OA
211	2,536	0,792	0,623	0,994	0,617	0,423	0,386	0,156	OA
212	1,817	0,766	0,626	0,548	0,861	0,665	0,39	0,319	OA
213	2,239	0,842	0,885	0,588	1,813	1,16	0,596	0,422	OA
214	1,206	0,85	0,895	0,437	1,857	1,012	0,599	1,291	OA
215	1,682	0,907	0,584	0,453	1,629	0,925	0,873	0,879	OA
216	0,975	0,544	0,533	0,511	1,492	0,699	1,171	1,185	OA
217	1,779	0,875	0,876	0,509	1,908	1,096	0,744	1,481	OA
218	1,243	0,531	0,46	0,734	1,298	1,505	1,611	0,143	OA
219	0,565	0,425	0,332	0,214	0,86	0,423	0,199	0,336	OA
220	0,572	0,216	0,193	0,257	0,512	0,418	0,467	0,259	OA
221	1,17	0,987	0,403	0,178	0,199	0,211	0,08	0,225	OA
222	0,451	0,318	0,227	0,054	0,095	0,066	0,047	0,108	OA
223	3,636	2,867	0,974	0,803	0,231	0,276	0,393	0,298	OA
224	1,102	0,617	0,308	0,117	0,197	0,083	0,091	0,167	OA

**Lampiran 3.** Data Hasil Ekstraksi B<sub>AM</sub> dan B<sub>FM</sub> untuk sinyal dengan potongan 5 detik

No.	Bam1	Bam2	Bam3	Bam4	Bfm1	Bfm2	Bfm3	Bfm4	Class
0	0.49	0.563	0.409	0.463	0.126	0.189	0.202	0.176	<i>Ictal</i>
1	1,083	0.619	0.513	0.605	0.145	0.279	0.294	0.287	<i>Ictal</i>
2	0.873	0.591	0.543	0.589	0.171	0.197	0.202	0.105	<i>Ictal</i>
3	0.515	0.54	0.394	0.461	0.189	0.202	0.276	0.227	<i>Ictal</i>
4	0.728	0.626	0.93	0.49	0.165	0.278	0.179	0.312	<i>Ictal</i>
5	0.71	0.575	0.469	0.504	0.17	0.228	0.199	0.194	<i>Ictal</i>
6	0.762	0.693	0.444	0.45	0.157	0.259	0.269	0.337	<i>Ictal</i>
7	0.928	0.572	0.447	0.412	0.163	0.241	0.246	0.296	<i>Ictal</i>
8	1,004	0.565	0.374	0.349	0.164	0.284	0.329	0.358	<i>Ictal</i>
9	0.822	0.497	0.512	0.644	0.263	0.406	0.285	0.264	<i>Ictal</i>
10	0.586	0.672	0.665	0.533	0.184	0.241	0.184	0.205	<i>Ictal</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
108	1,643	1,135	0.65	0.552	3.0	2,457	3,313	3,402	OA
109	1,665	0.95	0.707	0.506	2,467	3,109	2,701	2,678	OA
110	0.883	0.718	0.454	0.437	0.825	0.905	1,019	0.861	OA
111	1,134	0.743	0.76	0.572	2,034	2,794	2,605	2,398	OA
112	1,801	0.882	0.913	0.538	2,874	3,684	2,813	4,121	OA
113	0.993	0.926	0.526	0.61	1,819	1,728	2,522	1,851	OA
114	4,684	2,292	1,063	1,062	1,819	1,635	2,301	1.7	OA
115	0.955	0.687	0.597	0.487	1,753	1,624	1,625	1,394	OA
116	1,139	0.692	0.617	0.265	1,941	2,113	1,534	2,256	OA
117	0.735	0.574	0.385	0.374	2,135	1,749	2,259	2,426	OA
118	1,845	1,164	0.735	0.472	4,291	3,129	2,675	3,819	OA
119	0.877	0.599	0.489	0.533	2.61	2,561	2,386	2,466	OA
120	1,148	0.751	0.62	0.411	3,153	3,179	2,715	3,242	OA
121	1,372	0.814	0.717	0.847	1,036	1,759	1,706	1,605	OA
122	0.763	0.575	0.646	0.465	2,708	2,904	2,088	2.48	OA
123	0.879	0.562	0.353	0.449	3,133	2.43	2,853	1,789	OA
124	1,013	0.812	0.59	0.585	3,772	3,427	3,428	3,377	OA
125	1,914	1,028	0.965	0.704	1,025	1,346	0.93	0.88	OA
126	1,969	0.86	0.743	0.547	0.453	0.629	0.571	0.565	OA
127	1,103	0.586	0.472	0.3	0.491	0.595	0.481	0.679	OA
128	1,991	0.631	0.497	0.332	0.292	0.446	0.357	0.529	OA
129	1,071	0.724	0.553	0.614	0.79	1,309	1,327	1,307	OA

**Lampiran 4.** Data Hasil Ekstraksi Fitur *Time Frequency* untuk sinyal dengan potongan 2 detik

No.	fr1	fr2	fr3	fr4	...	R_RMIFS2	R_RMIFS3	R_RMIFS4	Class
0	339,648	401,627	149,251	11,176	...	0.00064	0.00013	7.00E-05	Ictal
1	283,711	168,039	147,018	133,721	...	0.0002	0.00017	0.00012	Ictal
2	223,506	179,966	123,155	592,254	...	0.00029	6.00E-05	0.00119	Ictal
3	344,224	224,772	152,214	200,835	...	0.00025	7.00E-05	0.00018	Ictal
4	355,913	397,993	421,539	240,403	...	0.00066	0.00073	0.00028	Ictal
5	239,017	188,812	287,638	171,529	...	0.00025	0.00041	0.00016	Ictal
6	300,147	118,945	23,587	232,878	...	0.00011	0.00036	0.00028	Ictal
7	325,427	309,429	233,025	0.92519	...	0.00064	0.00035	7.00E-05	Ictal
8	290,285	19,457	368,668	102,207	...	0.00028	0.00062	5.00E-05	Ictal
9	289,884	327,272	197,221	0.9945	...	0.00095	0.00029	8.00E-05	Ictal
10	253,368	147,316	107,724	17,707	...	0.00011	6.00E-05	0.00017	Ictal
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
203	1,310,246	1,218,441	297,838	21,478	...	0.01554	0.00042	4.00E-05	OA
204	638,895	413,139	523,631	268,728	...	0.00099	0.00244	3.00E-05	OA
205	490,525	294,669	24,296	155,042	...	0.00047	0.0004	0.00011	OA
206	490,525	294,669	24,296	155,042	...	0.00047	0.0004	0.00011	OA
207	40,751	25,514	338,292	192,228	...	0.00023	0.00066	3.00E-05	OA
208	428,378	274,581	239,176	23,873	...	0.00024	5.00E-05	3.00E-05	OA
209	986,863	752,702	244,306	186,571	...	0.00384	0.00022	0.00011	OA
210	433,732	22,379	168,607	135,542	...	0.00032	0.00013	1.00E-05	OA
211	610,193	281,457	247,896	273,209	...	0.0004	0.00032	0.00036	OA
212	484,702	301,504	243,032	232,794	...	0.0004	0.00022	0.00017	OA
213	602,994	426,657	378,997	329,601	...	0.00053	0.00048	0.0001	OA
214	476,571	400,866	37,126	326,973	...	0.00035	0.00042	8.00E-05	OA
215	531,527	424,348	384,065	336,223	...	0.00052	0.00017	9.00E-05	OA
216	486,274	404,703	368,574	340,846	...	0.00019	0.00019	0.00011	OA
217	600,245	45,171	435,277	399,724	...	0.00057	0.00026	0.00012	OA
218	5,522	404,234	394,277	371,129	...	0.00018	0.0001	0.00038	OA
219	209,842	1,912	1,571	146,109	...	0.00011	6.00E-05	1.00E-05	OA
220	207,902	159,004	141,928	147,491	...	3.00E-05	1.00E-05	1.00E-05	OA
221	377,779	34,533	154,033	127,733	...	0.00087	7.00E-05	1.00E-05	OA
222	146,064	0.99926	0.58685	0.35444	...	8.00E-05	1.00E-05	0.0	OA
223	1,232,116	834,419	268,098	1,836	...	0.00829	0.00042	0.00012	OA
224	343,425	226,287	108,537	0.68276	...	0.00049	4.00E-05	0.0	OA

**Lampiran 5.** Data Hasil Ekstraksi Fitur *Time Frequency* untuk sinyal dengan potongan 5 detik

	fr1	fr2	fr3	fr4	...	R_RMIFS2	R_RMIFS3	R_RMIFS4	Class
0	1,474	1.25	0.91	1,032	...	0.0	0.0	0.0	<i>Ictal</i>
1	4,087	1,251	1,209	1,473	...	0.0	0.0	0.0	<i>Ictal</i>
2	2,174	1,551	1,301	1,606	...	0.0	0.0	0.0	<i>Ictal</i>
3	1,648	1.46	0.916	1,025	...	0.0	0.0	0.0	<i>Ictal</i>
4	2,309	1,299	1,791	1,037	...	0.0	0.0	0.0	<i>Ictal</i>
5	2,264	1,291	1,068	1,227	...	0.0	0.0	0.0	<i>Ictal</i>
6	2,603	1.64	0.992	0.912	...	0.0	0.0	0.0	<i>Ictal</i>
7	3,013	1,251	1,026	0.885	...	0.0	0.0	0.0	<i>Ictal</i>
8	2,891	1,572	1,026	1,049	...	0.0	0.0	0.0	<i>Ictal</i>
9	2,752	1,122	1,208	1,584	...	0.0	0.0	0.0	<i>Ictal</i>
10	1,851	2,132	1,634	1,194	...	0.0	0.0	0.0	<i>Ictal</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
98	13,595	9,959	9,748	9,051	...	0.001	0.001	0.0	OA
99	6,455	3,884	4,408	3,332	...	0.001	0.001	0.0	OA
100	7,559	6,782	6,591	6,416	...	0.0	0.0	0.0	OA
101	17,918	12,339	11,877	11.57	...	0.001	0.001	0.001	OA
102	9,712	8,389	8,174	7,791	...	0.0	0.0	0.0	OA
103	22,025	16,629	14,685	14,366	...	0.005	0.001	0.001	OA
104	10,111	9,228	8,666	8,544	...	0.0	0.0	0.0	OA
105	10,912	7.51	6,491	6,611	...	0.001	0.0	0.0	OA
106	6,457	5,537	5,318	5,157	...	0.0	0.0	0.0	OA
107	17,197	10,084	9,921	8,809	...	0.002	0.001	0.0	OA
108	8.31	6,993	6,264	6,412	...	0.001	0.0	0.0	OA
109	9,477	7,376	6,832	6,752	...	0.001	0.0	0.0	OA
110	3,455	2.66	2,266	2,167	...	0.0	0.0	0.0	OA
111	7,142	5,782	5,616	5,336	...	0.0	0.0	0.0	OA
112	10,067	7,978	7,631	7,412	...	0.0	0.0	0.0	OA
113	6,433	5,514	5,013	5,089	...	0.001	0.0	0.0	OA
114	13,094	7,205	5,021	5,117	...	0.001	0.0	0.0	OA
115	3,943	3,324	3,157	2,957	...	0.0	0.0	0.0	OA
116	5,532	4,603	4.36	4,052	...	0.0	0.0	0.0	OA
117	4,941	4,571	4,155	4.08	...	0.0	0.0	0.0	OA
118	9,234	8,349	7,486	7,149	...	0.001	0.0	0.0	OA
119	5,968	5,461	5,234	5.03	...	0.0	0.0	0.0	OA

## Lampiran 6. Script Potong Data Sinyal *Ictal* dengan durasi 2 detik

```
import pandas as pd
import numpy as np
SN= pd.read_csv('E:/1. Thesis/EEG/Data_EEG/MONTASE_CZ/3_SN_An.csv', sep=',')
Il_S1=SN.iloc[8001-1:9000,7]
Il_S2=SN.iloc[8001-1:9000,17]
Il_S3=SN.iloc[9001-1:10000,7]
Il_S4=SN.iloc[9001-1:10000,17]
Il_S5=SN.iloc[10001-1:11000,7]
Il_S6=SN.iloc[10001-1:11000,17]
Il_S7=SN.iloc[11001-1:12000,7]
Il_S8=SN.iloc[11001-1:12000,17]
Il_S9=SN.iloc[12001-1:13000,7]
Il_S10=SN.iloc[12001-1:13000,17]
.
.
.

Il_S94=SN.iloc[810001-1:811000,7]
Il_S95=SN.iloc[810001-1:811000,17]
Il_S96=SN.iloc[811001-1:812000,7]
Il_S97=SN.iloc[811001-1:812000,17]
Il_S98=SN.iloc[812001-1:813000,7]
Il_S99=SN.iloc[812001-1:813000,17]
Il_S100=SN.iloc[813001-1:814000,7]

Il_S1.to_csv('Il_S1.txt',index=False)
Il_S2.to_csv('Il_S2.txt',index=False)
Il_S3.to_csv('Il_S3.txt',index=False)
Il_S4.to_csv('Il_S4.txt',index=False)
Il_S5.to_csv('Il_S5.txt',index=False)
Il_S6.to_csv('Il_S6.txt',index=False)
Il_S7.to_csv('Il_S7.txt',index=False)
Il_S8.to_csv('Il_S8.txt',index=False)
Il_S9.to_csv('Il_S9.txt',index=False)
Il_S10.to_csv('Il_S10.txt',index=False)
.
.
.

Il_S94.to_csv('Il_S94.txt',index=False)
Il_S95.to_csv('Il_S95.txt',index=False)
Il_S96.to_csv('Il_S96.txt',index=False)
Il_S97.to_csv('Il_S97.txt',index=False)
Il_S98.to_csv('Il_S98.txt',index=False)
Il_S99.to_csv('Il_S99.txt',index=False)
Il_S100.to_csv('Il_S100.txt',index=False)
Il_S101.to_csv('Il_S101.txt',index=False)
```

### Lampiran 7. Script Potong Data Sinyal *Ocular artifacts* dengan durasi 2 detik

```
RM= pd.read_csv(r'E:/1. Thesis/EEG/Data_EEG/MONTASE_CZ/4_RM_An.csv', sep=',')
RMa= pd.read_csv(r'E:/1. Thesis/EEG/Data_EEG/MONTASE_CZ/4a_RM_An.csv', sep=',')
II_OA1=SN.iloc[38501-1:39500,7]
II_OA2=SN.iloc[39501-1:40500,7]
II_OA3=SN.iloc[40501-1:41500,7]
II_OA4=SN.iloc[41501-1:42500,7]
II_OA5=SN.iloc[38501-1:39500,17]
II_OA6=SN.iloc[39501-1:40500,17]
II_OA7=SN.iloc[40501-1:41500,17]
II_OA8=SN.iloc[41501-1:42500,17]
II_OA9=SN.iloc[50001-1:51000,7]
II_OA10=SN.iloc[51001-1:52000,7]
.
.
.
II_OA115=RMa.iloc[1001-1:2000,17]
II_OA116=RMa.iloc[2001-1:3000,7]
II_OA117=RMa.iloc[2001-1:3000,17]
II_OA118=RMa.iloc[3001-1:4000,7]
II_OA119=RMa.iloc[3001-1:4000,17]
II_OA120=RMa.iloc[4001-1:5000,7]
II_OA121=RMa.iloc[4001-1:5000,17]
II_OA122=RMa.iloc[24001-1:25000,7]
II_OA123=RMa.iloc[24001-1:25000,17]
II_OA124=RMa.iloc[25001-1:26000,7]
II_OA125=RMa.iloc[25001-1:26000,17]

II_OA1.to_csv('II_OA1.txt',index=False)
II_OA2.to_csv('II_OA2.txt',index=False)
II_OA3.to_csv('II_OA3.txt',index=False)
II_OA4.to_csv('II_OA4.txt',index=False)
II_OA5.to_csv('II_OA5.txt',index=False)
II_OA6.to_csv('II_OA6.txt',index=False)
II_OA7.to_csv('II_OA7.txt',index=False)
II_OA8.to_csv('II_OA8.txt',index=False)
II_OA9.to_csv('II_OA9.txt',index=False)
II_OA10.to_csv('II_OA10.txt',index=False)
.
..
.
. II_OA115.to_csv('II_OA115.txt',index=False)
II_OA116.to_csv('II_OA116.txt',index=False)
II_OA117.to_csv('II_OA117.txt',index=False)
II_OA118.to_csv('II_OA118.txt',index=False)
II_OA119.to_csv('II_OA119.txt',index=False)
II_OA120.to_csv('II_OA120.txt',index=False)
II_OA121.to_csv('II_OA121.txt',index=False)
II_OA122.to_csv('II_OA122.txt',index=False)
II_OA123.to_csv('II_OA123.txt',index=False)
II_OA124.to_csv('II_OA124.txt',index=False)
II_OA125.to_csv('II_OA125.txt',index=False)
```

## Lampiran 8. Script Potong Data Sinyal *Ictal* dengan durasi 5 detik

```
RMfull= pd.read_fwf('E:/1. Thesis/EEG/Data_EEG/MONTASE_CZ/RMa.txt', sep=" ", header=None,
names=["Fp1-F7", " F7-T1", "T1-T3", "T3-T5", "T5-O1", "Fp2-F8", "F8-T2", "T2-T4", "T4-T6", "T6-
O2", "T1-T2", "A1-A2", "Fp1-F3", "F3-C3", "C3-P3", "P3-O1", " Fp2-F4", "F4-C4", "C4-P4", "P4-O2", "Fz-
Cz", "Cz-Pz", "X3-Pz", " X4-Pz", "X2-X1"])
V_S1=SN.iloc[8500:11000,7]
V_S2=SN.iloc[8500:11000,17]
V_S3=SN.iloc[11000:13500,7]
V_S4=SN.iloc[11000:13500,17]
V_S5=SN.iloc[548500:551000,7]
V_S6=SN.iloc[548500:551000,17]
V_S7=SN.iloc[551000:553500,7]
V_S8=SN.iloc[551000:553500,17]
V_S9=SN.iloc[553500:556000,7]
V_S10=SN.iloc[553500:556000,17]
.
.
.
V_S38=SN.iloc[810500:813000,17]
V_S39=SN.iloc[312500:315000,7]
V_S40=SN.iloc[312500:315000,17]
V_S41=SN.iloc[315000:317500,7]
V_S42=SN.iloc[315000:317500,17]
V_S43=SN.iloc[317500:320000,7]
V_S44=SN.iloc[317500:320000,17]
V_S1.to_csv('V_S1.txt',index=False)
V_S2.to_csv('V_S2.txt',index=False)
V_S3.to_csv('V_S3.txt',index=False)
V_S4.to_csv('V_S4.txt',index=False)
V_S5.to_csv('V_S5.txt',index=False)
V_S6.to_csv('V_S6.txt',index=False)
V_S7.to_csv('V_S7.txt',index=False)
V_S8.to_csv('V_S8.txt',index=False)
V_S9.to_csv('V_S9.txt',index=False)
V_S10.to_csv('V_S10.txt',index=False)
.
.
.
V_S38.to_csv('V_S38.txt',index=False)
V_S39.to_csv('V_S39.txt',index=False)
V_S40.to_csv('V_S40.txt',index=False)
V_S41.to_csv('V_S41.txt',index=False)
V_S42.to_csv('V_S42.txt',index=False)
V_S43.to_csv('V_S43.txt',index=False)
V_S44.to_csv('V_S44.txt',index=False)
```



**Lampiran 9.** Script Potong Data Sinyal *Ocular artifacts* dengan durasi 5 detik

```
V_OA1=SN.iloc[38500:41000,7]
V_OA2=SN.iloc[38500:41000,17]
V_OA3=SN.iloc[49500:52000,7]
V_OA4=SN.iloc[49500:52000,17]
V_OA5=SN.iloc[52000:54500,7]
V_OA6=SN.iloc[52000:54500,17]
V_OA7=SN.iloc[94500:97000,7]
V_OA8=SN.iloc[94500:97000,17]
V_OA9=SN.iloc[117500:120000,7]
V_OA10=SN.iloc[117500:120000,17]
.
.
.
V_OA70=RMfull.iloc[1175500:1178000,17]
V_OA71=RMfull.iloc[1178000:1180500,7]
V_OA72=RMfull.iloc[1178000:1180500,17]
V_OA73=RMfull.iloc[1196000:1198500,7]
V_OA74=RMfull.iloc[1196000:1198500,17]
V_OA75=RMfull.iloc[1198500:1201000,7]
V_OA76=RMfull.iloc[1198500:1201000,17]

V_OA1.to_csv('V_OA1.txt',index=False)
V_OA2.to_csv('V_OA2.txt',index=False)
V_OA3.to_csv('V_OA3.txt',index=False)
V_OA4.to_csv('V_OA4.txt',index=False)
V_OA5.to_csv('V_OA5.txt',index=False)
V_OA6.to_csv('V_OA6.txt',index=False)
V_OA7.to_csv('V_OA7.txt',index=False)
V_OA8.to_csv('V_OA8.txt',index=False)
V_OA9.to_csv('V_OA9.txt',index=False)
.V_OA10.to_csv('V_OA10.txt',index=False)
.
.
.
V_OA70.to_csv('V_OA70.txt',index=False)
V_OA71.to_csv('V_OA71.txt',index=False)
V_OA72.to_csv('V_OA72.txt',index=False)
V_OA73.to_csv('V_OA73.txt',index=False)
V_OA74.to_csv('V_OA74.txt',index=False)
V_OA75.to_csv('V_OA75.txt',index=False)
V_OA76.to_csv('V_OA76.txt',index=False)
```

## Lampiran 10. Script Ekstraksi Fitur $B_{AM}$ dan $B_{FM}$ untuk data pemotongan 2 detik

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from time import sleep
from scipy.signal import hilbert, savgol_filter
import scipy as sp
from numpy import sin, tan, pi
from scipy.signal import argrelemin, argrelmax
from scipy.interpolate import splrep, splev
import warnings
warnings.filterwarnings("ignore", category=np.VisibleDeprecationWarning)
%matplotlib notebook
np.seterr(divide='ignore', invalid='ignore')
import math
warnings.filterwarnings('ignore')
from numpy import linalg as la
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn import SVM
from sklearn.metrics import roc_curve, AUC
from sklearn.model_selection import StratifiedShuffleSplit
%matplotlib inline
import seaborn as sns

def buka_folder(buka):
    # Buka folder
    data = []
    for file in os.listdir(buka):
        fl = buka+ file
        data.append(fl)
    data = sorted(data)

    te = []
    st = 'file'
    for i in range(len(data)):
        x = pd.read_table(data[i],header=None)
        # print(x)
        x.columns=[st+str(i)]
        te.append(x)

    return te

def tabel(tabel):
    folder_table = None
    for te in tabel:
        folder_table = pd.concat([folder_table, te],axis=1)
    return folder_table

def buat_data(ket,head):
```

**Lampiran 10.** Script Ekstraksi Fitur  $B_{AM}$  dan  $B_{FM}$  untuk data pemotongan 2 detik (Lanjutan1)

```
ket_ = np.zeros((len(ket),(len(head))))
for i in range(len(head)):
    ket[:,i] = ket[head[i]]
    sleep(0.01)
return ket_

stop = 0.05

def batas(s):
    #mendapatkan syarat yaitu zero crossing harus >=1"
    n_max = argrelmax(s)[0].shape[0]
    n_min = argrelmin(s)[0].shape[0]
    n_zeros = (s[:-1] * s[1:] < 0).sum()
    return abs((n_max + n_min) - n_zeros)

def dekomposisi(s):
    #mendapatkan IMF
    atas = argrelmax(s)[0]
    bawah = argrelmin(s)[0]
    b_atas = np.zeros((atas.shape[0] + 2,))
    b_atas[1:-1] = atas
    b_atas[-1] = s.shape[0] - 1
    b_bawah = np.zeros((bawah.shape[0] + 2,))
    b_bawah[1:-1] = bawah
    b_bawah[-1] = s.shape[0] - 1
    tck = splrep(b_atas, s[b_atas.astype(int)],k = 1)
    envelop_atas = splev(np.arange(s.shape[0]), tck)
    tck = splrep(b_bawah, s[b_bawah.astype(int)])
    envelop_bawah = splev(np.arange(s.shape[0]), tck)
    amp_m = np.abs(envelop_atas - envelop_bawah) / 2
    n_tengah = (envelop_atas + envelop_bawah) / 2
    amp_g = np.abs(n_tengah) / amp_m
    return s - n_tengah, amp_g.sum()

def emd(s, n_IMF):
    IMF = np.zeros((n_IMF + 1, s.shape[0]))
    for i in range(n_IMF):
        h = s - IMF.sum(0)
        while (batas(h) > 1):
            h,amp_g = dekomposisi(h)
            if amp_g <= stop:
                break
        IMF[i, :] = h
    IMF[-1, :] = s - IMF.sum(0)
    return IMF

def fitur(c):
    durasi = 2
    fs = 500
```

**Lampiran 10.** Script Ekstraksi Fitur  $B_{AM}$  dan  $B_{FM}$  untuk data pemotongan 2 detik (Lanjutan2)

```
samples = int(fs*durasi)
t = np.arange(samples) / fs

h=hilbert(c)
amp =np.abs(h)
fase_sesaat =np.unwrap(np.angle(h))
freq=(np.diff(fase_sesaat) / (2.0*np.pi)*fs)
fase = savgol_filter(fase_sesaat, window_length=7, polyorder=3, deriv=1)
frek = fase / (2.0*np.pi)*fs * np.conj(h) / (h * np.conj(h))
sleep(0.01)
return (amp,frek)

def b_am (amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    return bam

def b_fm (freq,amp,s):
    hei= sum(abs(s)**2)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    t_f= (((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    return bfm

def hasil (direktori):
    buka=buka_folder(direktori)
    bigE = tabel(buka)
    head = list(bigE.columns.values)
    bigE.head(4)
    matE = buat_data(bigE,head)
    matE = np.nan_to_num(matE)

    s = []
    IMFs = []
    c = []
    ampli=[]
    frek=[]
    bam=[]
    bfm=[]
    M_B = []

    for i in range(25):
        print('File ke: ', i)
        s.append(matE[:,i])
```

**Lampiran 10.** Script Ekstraksi Fitur  $B_{AM}$  dan  $B_{FM}$  untuk data pemotongan 2 detik (Lanjutan1)

```
IMFs.append([i])
ampli.append([i])
frek.append([i])
bam.append([i])
bfm.append([i])
#IMFs[i] = EMD(s[i]).decompose()
IMFs[i]=emd(s[i],8)
c.append(np.nan_to_num(fitur(IMFs[i])))
ampli[i], frek[i] = c[i]
bam[i] = b_am (ampli[i],s[i])
bfm[i]= b_fm(frek[i], ampli[i],s[i])

while (len(bam[i]) < 4) or (len(bfm[i]) < 4):
    if len(bam[i]) < 4:
        bam[i] = np.insert(bam[i], len(bam[i]), 0,)
    if len(bfm[i]) < 4:
        bfm[i] = np.insert(bfm[i], len(bfm[i]), 0,)

M_B.append(np.concatenate((bam[i][:4],bfm[i][:4])))

return M_B
```

## Lampiran 11. Script Ekstraksi Fitur $B_{AM}$ dan $B_{FM}$ untuk data pemotongan 5 detik

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from time import sleep
from scipy.signal import hilbert, savgol_filter
import scipy as sp
from numpy import sin, tan, pi
from scipy.signal import argrelemin, argrelmax
from scipy.interpolate import splrep, splev
import warnings
warnings.filterwarnings("ignore", category=np.VisibleDeprecationWarning)
%matplotlib notebook
np.seterr(divide='ignore', invalid='ignore')
import math
warnings.filterwarnings('ignore')
from numpy import linalg as la
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn import SVM
from sklearn.metrics import roc_curve, AUC
from sklearn.model_selection import StratifiedShuffleSplit
%matplotlib inline
import seaborn as sns

def buka_folder(buka):
    # Buka folder
    data = []
    for file in os.listdir(buka):
        fl = buka+ file
        data.append(fl)
    data = sorted(data)

    te = []
    st = 'file'
    for i in range(len(data)):
        x = pd.read_table(data[i],header=None)
        # print(x)
        x.columns=[st+str(i)]
        te.append(x)

    return te

def tabel(tabel):
    folder_table = None
    for te in tabel:
```

## Lampiran 11 Script Ekstraksi Fitur $B_{AM}$ dan $B_{FM}$ untuk data pemotongan 5 detik (lanjutan)

```
folder_table = pd.concat([folder_table, te],axis=1)
return folder_table

def buat_data(ket,head):
    ket_ = np.zeros((len(ket),(len(head))))
    for i in range(len(head)):
        ket_[i] = ket[head[i]]
        sleep(0.01)
    return ket_

stop = 0.05

def batas(s):
    #mendapatkan syarat yaitu zero crossing harus >=1"
    n_max = argrelmax(s)[0].shape[0]
    n_min = argrelmin(s)[0].shape[0]
    n_zeros = (s[:-1] * s[1:] < 0).sum()
    return abs((n_max + n_min) - n_zeros)

def dekomposisi(s):
    #mendapatkan IMF
    atas = argrelmax(s)[0]
    bawah = argrelmin(s)[0]
    b_atas = np.zeros((atas.shape[0] + 2,))
    b_atas[1:-1] = atas
    b_atas[-1] = s.shape[0] - 1
    b_bawah = np.zeros((bawah.shape[0] + 2,))
    b_bawah[1:-1] = bawah
    b_bawah[-1] = s.shape[0] - 1
    tck = splrep(b_atas, s[b_atas.astype(int)],k = 1)
    envelop_atas = splev(np.arange(s.shape[0]), tck)
    tck = splrep(b_bawah, s[b_bawah.astype(int)])
    envelop_bawah = splev(np.arange(s.shape[0]), tck)
    amp_m = np.abs(envelop_atas - envelop_bawah) / 2
    n_tengah = (envelop_atas + envelop_bawah) / 2
    amp_g = np.abs(n_tengah) / amp_m
    return s - n_tengah, amp_g.sum()

def emd(s, n_IMF):
    IMF = np.zeros((n_IMF + 1, s.shape[0]))
    for i in range(n_IMF):
        h = s - IMF.sum(0)
        while (batas(h) > 1):
            h,amp_g = dekomposisi(h)
```

**Lampiran 11** Script Ekstraksi Fitur  $B_{AM}$  dan  $B_{FM}$  untuk data pemotongan 5 detik (lanjutan)

```
        if amp_g <= stop:
            break
        IMF[i, :] = h
    IMF[-1, :] = s - IMF.sum(0)
    return IMF

def fitur(c):
    durasi = 5
    fs = 500
    samples = int(fs*durasi)
    t = np.arange(samples) / fs

    h=hilbert(c)
    amp =np.abs(h)
    fase_sesaat =np.unwrap(np.angle(h))
    freq=(np.diff(fase_sesaat) / (2.0*np.pi)*fs)
    fase = savgol_filter(fase_sesaat, window_length=7, polyorder=3, deriv=1)
    frek = fase / (2.0*np.pi)*fs * np.conj(h) / (h * np.conj(h))
    sleep(0.01)
    return (amp,frek)

def b_am (amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    return bam

def b_fm (freq,amp,s):
    hei= sum(abs(s)**2)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    t_f= (((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    return bfm
```



## Lampiran 12. Script Ekstraksi Fitur *time frequency* untuk data pemotongan 2 detik

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from time import sleep
from scipy.signal import hilbert, savgol_filter
import scipy as sp
from numpy import sin, tan, pi
from scipy.signal import argrelemin, argrelmax
from scipy.interpolate import splrep, splev
import warnings
warnings.filterwarnings("ignore", category=np.VisibleDeprecationWarning)
%matplotlib notebook
np.seterr(divide='ignore', invalid='ignore')
import math
warnings.filterwarnings('ignore')
from numpy import linalg as la
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn import SVM
from sklearn.metrics import roc_curve, AUC
from sklearn.model_selection import StratifiedShuffleSplit
%matplotlib inline
import seaborn as sns
from PyEMD import EMD
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn import SVM

def buka_folder(buka):
    # Buka folder
    data = []
    for file in os.listdir(buka):
        fl = buka+ file
        data.append(fl)
    data = sorted(data)

    te = []
    st = 'file'
    for i in range(len(data)):
        x=pd.read_table(data[i], delim_whitespace=True,header=None)
        #x = pd.read_table(data[i],header=None)
        #print(x)
        x.columns=[st+str(i)]
        te.append(x)
    return te

def tabel(tabel):
    folder_table = None
    for te in tabel:
        folder_table = pd.concat([folder_table, te],axis=1)
```

## Lampiran 12 Script Ekstraksi Fitur *time frequency* untuk data pemotongan 2 detik (Lanjutan)

```
return folder_table

def buat_data(ket,head):
    ket_ = np.zeros((len(ket),len(head)))
    for i in range(len(head)):
        ket[:,i] = ket[head[i]]
        sleep(0.01)
    return ket_

stop = 0.05

def batas(s):
    #mendapatkan syarat yaitu zero crossing harus >=1"
    n_max = argrelmax(s)[0].shape[0]
    n_min = argrelmin(s)[0].shape[0]
    n_zeros = (s[:-1] * s[1:] < 0).sum()
    return abs((n_max + n_min) - n_zeros)

def dekomposisi(s):
    #mendapatkan IMF
    atas = argrelmax(s)[0]
    bawah = argrelmin(s)[0]
    b_atas = np.zeros((atas.shape[0] + 2,))
    b_atas[1:-1] = atas
    b_atas[-1] = s.shape[0] - 1
    b_bawah = np.zeros((bawah.shape[0] + 2,))
    b_bawah[1:-1] = bawah
    b_bawah[-1] = s.shape[0] - 1
    tck = splrep(b_atas, s[b_atas.astype(int)],k = 1) #APA
    envelop_atas = splev(np.arange(s.shape[0]), tck)
    tck = splrep(b_bawah, s[b_bawah.astype(int)]) #APA
    envelop_bawah = splev(np.arange(s.shape[0]), tck)
    amp_m = np.abs(envelop_atas - envelop_bawah) / 2
    n_tengah = (envelop_atas + envelop_bawah) / 2
    amp_g = np.abs(n_tengah) / amp_m
    return s - n_tengah, amp_g.sum()

def emd(s, n_IMF):
    IMF = np.zeros((n_IMF + 1, s.shape[0]))
    for i in range(n_IMF):
        h = s - IMF.sum(0)
        while (batas(h) > 1):
            h,amp_g = dekomposisi(h)
            if amp_g <= stop:
                break
        IMF[i, :] = h
    IMF[-1, :] = s - IMF.sum(0)
    return IMF

def fitur(c):
    durasi = 2
```

## Lampiran 12 Script Ekstraksi Fitur *time frequency* untuk data pemotongan 2 detik (Lanjutan)

```
fs = 500
samples = int(fs*durasi)
t = np.arange(samples) / fs
h=hilbert(c).real
#amp =np.abs(h)
amp=np.abs(h)
fase_sesaat =np.unwrap(np.angle(h))
freq=(np.diff(fase_sesaat) / (2.0*np.pi)*fs)
fase = savgol_filter(fase_sesaat, window_length=7, polyorder=3, deriv=1)
frek = np.abs(fase / (2.0*np.pi)*fs * np.conj(h) / (h * np.conj(h)))
sleep(0.01)
return (amp,frek)

def b_am (amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    return bam

def b_fm (freq,amp,s):
    hei= sum(abs(s)**2)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    t_f= (((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    return bfm

def RMS_freq(freq,amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    w1=w.sum(axis=1)
    t_f= (((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    B=np.sqrt(bam**2+bfm**2)
    RMSfreq=np.sqrt(bam**2+bfm**2+w1**2)
    R_RMSfreq=(bam+bfm)/w1**2
    return (RMSfreq,R_RMSfreq)
    #return R_RMSfreq

def Dom_freq(freq,amp,s):
    hei= sum(abs(s)**2)
    fs = 500
```

**Lampiran 12** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 2 detik (Lanjutan)

```

t_a=(np.diff(amp) / (2.0*np.pi)*fs)
t_a2=t_a**2
hasil_a= t_a2.sum(axis=1)
bam=np.sqrt(1/hei*hasil_a)
i_f=(freq*(amp**2))
w=1/hei*i_f
w1=w.sum(axis=1)
t_f= (((freq-w)**2)*(amp**2))
tt_f=t_f.sum(axis=1)
bfm=np.sqrt(1/hei*tt_f)
B=np.sqrt(bam**2+bfm**2)
Domfreq=np.sqrt((bam**2)+w1) #eq20
R_Domfreq= ((bfm**2)+w1)/(bam**2)
return (Domfreq,R_Domfreq)
#return R_Domfreq

def RMI_fs(freq,amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    w1=w.sum(axis=1)
    t_f= (((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    B=np.sqrt(bam**2+bfm**2)
    RMIFS=np.sqrt((bfm**2)+w1**2)
    R_RMIFS=(w1**2)/(bfm**2)
    return (RMIFS,R_RMIFS)
#return R_RMIFS

def hasil (direktori):
    buka=buka_folder(direktori)
    bigE = tabel(buka)
    head = list(bigE.columns.values)
    bigE.head(6)
    matE = buat_data(bigE,head)
    matE = np.nan_to_num(matE)

    s = []
    IMFs = []
    c = []
    ampli=[]
    frek=[]
    bam=[]
    bfm=[]
    M_B = []
    RMSfreq=[]
    R_RMSfreq=[]

```

**Lampiran 12** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 2 detik (Lanjutan)

```

Domfreq=[]
R_Domfreq=[]
RMIFS=[]
R_RMIFS=[]

for i in range(25):
    print('File ke: ', i)
    s.append(matE[:,i])
    IMFs.append([i])
    ampli.append([i])
    frek.append([i])
    bam.append([i])
    bfm.append([i])
    RMSfreq.append([i])
    R_RMSfreq.append([i])
    Domfreq.append([i])
    R_Domfreq.append([i])
    RMIFS.append([i])
    R_RMIFS.append([i])
    #IMFs[i] = EMD(s[i]).decompose()
    IMFs[i]=emd(s[i],4)
    c.append(np.nan_to_num(fitur(IMFs[i])))
    ampli[i], frek[i] = c[i]
    bam[i] = b_am (ampli[i],s[i])
    bfm[i]= b_fm(frek[i], ampli[i],s[i])
    RMSfreq[i],R_RMSfreq[i]= RMS_freq(frek[i], ampli[i],s[i])
    # R_RMSfreq[i]= RMS_freq(frek[i], ampli[i],s[i])
    Domfreq[i],R_Domfreq[i]= Dom_freq(frek[i], ampli[i],s[i])
    #R_Domfreq[i]= Dom_freq(frek[i], ampli[i],s[i])
    RMIFS[i],R_RMIFS[i]= RMI_fs(frek[i], ampli[i],s[i])
    #R_RMIFS[i]= RMI_fs(frek[i], ampli[i],s[i])

    while (len(RMSfreq[i]) < 4) or (len(R_RMSfreq[i]) < 4) or (len(Domfreq[i]) < 4) or
    (len(R_Domfreq[i]) < 4) or (len(RMIFS[i]) < 4) or (len(R_RMIFS[i]) < 4):
        if len(RMSfreq[i]) < 4:
            RMSfreq[i] = np.insert(RMSfreq[i], len(RMSfreq[i]), 0,)
        if len(R_RMSfreq[i]) < 4:
            R_RMSfreq[i] = np.insert(R_RMSfreq[i], len(R_RMSfreq[i]), 0,)
        if len(Domfreq[i]) < 4:
            Domfreq[i] = np.insert(Domfreq[i], len(Domfreq[i]), 0,)
        if len(R_Domfreq[i]) < 4:
            R_Domfreq[i] = np.insert(R_Domfreq[i], len(R_Domfreq[i]), 0,)
        if len(RMIFS[i]) < 4:
            RMIFS[i] = np.insert(RMIFS[i], len(RMIFS[i]), 0,)
        if len(R_RMIFS[i]) < 4:
            R_RMIFS[i] = np.insert(R_RMIFS[i], len(R_RMIFS[i]), 0,)

M_B.append(np.concatenate((RMSfreq[i][:4],R_RMSfreq[i][:4],Domfreq[i][:4],R_Domfreq[i][:4],
,RMIFS[i][:4],R_RMIFS[i][:4])))

return M_B

```

### Lampiran 13. Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from time import sleep
from scipy.signal import hilbert, savgol_filter
import scipy as sp
from numpy import sin, tan, pi
from scipy.signal import argrelemin, argrelmax
from scipy.interpolate import splrep, splev
import warnings
warnings.filterwarnings("ignore", category=np.VisibleDeprecationWarning)
%matplotlib notebook
np.seterr(divide='ignore', invalid='ignore')
import math
warnings.filterwarnings('ignore')
from numpy import linalg as la
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn import SVM
from sklearn.metrics import roc_curve, AUC
from sklearn.model_selection import StratifiedShuffleSplit
%matplotlib inline
import seaborn as sns
from PyEMD import EMD
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn import SVM

def buka_folder(buka):
    # Buka folder
    data = []
    for file in os.listdir(buka):
        fl = buka+ file
        data.append(fl)
    data = sorted(data)

    te = []
    st = 'file'
    for i in range(len(data)):
        x=pd.read_table(data[i], delim_whitespace=True,header=None)
        #x = pd.read_table(data[i],header=None)
        #print(x)
        x.columns=[st+str(i)]
        te.append(x)
    return te
```

**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```
def tabel(tabel):
    folder_table = None
    for te in tabel:
        folder_table = pd.concat([folder_table, te],axis=1)
    return folder_table

def buat_data(ket,head):
    ket_ = np.zeros((len(ket),(len(head))))
    for i in range(len(head)):
        ket[:,i] = ket[head[i]]
        sleep(0.01)
    return ket_

stop = 0.05

def batas(s):
    #mendapatkan syarat yaitu zerro crossing harus >=1"
    n_max = argrelmax(s)[0].shape[0]
    n_min = argrelmin(s)[0].shape[0]
    n_zeros = (s[:-1] * s[1:] < 0).sum()
    return abs((n_max + n_min) - n_zeros)

def dekomposisi(s):
    #mendapatkan IMF
    atas = argrelmax(s)[0]
    bawah = argrelmin(s)[0]
    b_atas = np.zeros((atas.shape[0] + 2,))
    b_atas[1:-1] = atas
    b_atas[-1] = s.shape[0] - 1
    b_bawah = np.zeros((bawah.shape[0] + 2,))
    b_bawah[1:-1] = bawah
    b_bawah[-1] = s.shape[0] - 1
    tck = splrep(b_atas, s[b_atas.astype(int)],k = 1) #APA
    envelop_atas = splev(np.arange(s.shape[0]), tck)
    tck = splrep(b_bawah, s[b_bawah.astype(int)]) #APA
    envelop_bawah = splev(np.arange(s.shape[0]), tck)
    amp_m = np.abs(envelop_atas - envelop_bawah) / 2
    n_tengah = (envelop_atas + envelop_bawah) / 2
    amp_g = np.abs(n_tengah) / amp_m
    return s - n_tengah, amp_g.sum()

def emd(s, n_IMF):
    IMF = np.zeros((n_IMF + 1, s.shape[0]))
    for i in range(n_IMF):
        h = s - IMF.sum(0)
        while (batas(h) > 1):
```

**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```
h,amp_g = dekomposisi(h)
    if amp_g <= stop:
        break
    IMF[i, :] = h
    IMF[-1, :] = s - IMF.sum(0)
    return IMF

def fitur(c):
    durasi = 2
    fs = 500
    samples = int(fs*durasi)
    t = np.arange(samples) / fs
    h=hilbert(c).real
    #amp =np.abs(h)
    amp=np.abs(h)
    fase_sesaat =np.unwrap(np.angle(h))
    freq=(np.diff(fase_sesaat) / (2.0*np.pi)*fs)
    fase = savgol_filter(fase_sesaat, window_length=7, polyorder=3, deriv=1)
    frek = np.abs(fase / (2.0*np.pi)*fs * np.conj(h) / (h * np.conj(h)))
    sleep(0.01)
    return (amp,frek)

def b_am (amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    return bam

def b_fm (freq,amp,s):
    hei= sum(abs(s)**2)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    t_f= (((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    return bfm

def RMS_freq(freq,amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
```



**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```

w1=w.sum(axis=1)
t_f=(((freq-w)**2)*(amp**2))
tt_f=t_f.sum(axis=1)
bfm=np.sqrt(1/hei*tt_f)
B=np.sqrt(bam**2+bfm**2)
RMSfreq=np.sqrt(bam**2+bfm**2+w1**2)
R_RMSfreq=(bam+bfm)/w1**2
return (RMSfreq,R_RMSfreq)
#return R_RMSfreq

def Dom_freq(freq,amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    w1=w.sum(axis=1)
    t_f=(((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    B=np.sqrt(bam**2+bfm**2)
    Domfreq=np.sqrt((bam**2)+w1) #eq20
    R_Domfreq= ((bfm**2)+w1)/(bam**2)
    return (Domfreq,R_Domfreq)
#return R_Domfreq

def RMI_fs(freq,amp,s):
    hei= sum(abs(s)**2)
    fs = 500
    t_a=(np.diff(amp) / (2.0*np.pi)*fs)
    t_a2=t_a**2
    hasil_a= t_a2.sum(axis=1)
    bam=np.sqrt(1/hei*hasil_a)
    i_f=(freq*(amp**2))
    w=1/hei*i_f
    w1=w.sum(axis=1)
    t_f=(((freq-w)**2)*(amp**2))
    tt_f=t_f.sum(axis=1)
    bfm=np.sqrt(1/hei*tt_f)
    B=np.sqrt(bam**2+bfm**2)
    RMIFS=np.sqrt((bfm**2)+w1**2)
    R_RMIFS=(w1**2)/(bfm**2)
    return (RMIFS,R_RMIFS)
#return R_RMIFS

```

**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```
def hasil (direktori):
    buka=buka_folder(direktori)
    bigE = tabel(buka)
    head = list(bigE.columns.values)
    bigE.head(6)
    matE = buat_data(bigE,head)
    matE = np.nan_to_num(matE)
    s = []
    IMFs = []
    c = []
    ampli=[]
    frek=[]
    bam=[]
    bfm=[]
    M_B = []
    RMSfreq=[]
    R_RMSfreq=[]
    Domfreq=[]
    R_Domfreq=[]
    RMIFS=[]
    R_RMIFS=[]

    for i in range(44):
        print('File ke: ', i)
        s.append(matE[:,i])
        IMFs.append([i])
        ampli.append([i])
        frek.append([i])
        bam.append([i])
        bfm.append([i])
        RMSfreq.append([i])
        R_RMSfreq.append([i])
        Domfreq.append([i])
        R_Domfreq.append([i])
        RMIFS.append([i])
        R_RMIFS.append([i])
        #IMFs[i] = EMD(s[i]).decompose()
        IMFs[i]=emd(s[i],4)
        c.append(np.nan_to_num(fitur(IMFs[i])))
        ampli[i], frek[i] = c[i]
        bam[i] = b_am (ampli[i],s[i])
        bfm[i]= b_fm(frek[i], ampli[i],s[i])
        RMSfreq[i],R_RMSfreq[i]= RMS_freq(frek[i], ampli[i],s[i])
        # R_RMSfreq[i]= RMS_freq(frek[i], ampli[i],s[i])
        Domfreq[i],R_Domfreq[i]= Dom_freq(frek[i], ampli[i],s[i])
        #R_Domfreq[i]= Dom_freq(frek[i], ampli[i],s[i])
        RMIFS[i],R_RMIFS[i]= RMI_fs(frek[i], ampli[i],s[i])
        #R_RMIFS[i]= RMI_fs(frek[i], ampli[i],s[i])
```

**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```

while (len(RMSfreq[i]) < 4) or (len(R_RMSfreq[i]) < 4) or (len(Domfreq[i]) < 4) or
(len(R_Domfreq[i]) < 4) or (len(RMIFS[i]) < 4) or (len(R_RMIFS[i]) < 4):
    if len(RMSfreq[i]) < 4:
        RMSfreq[i] = np.insert(RMSfreq[i], len(RMSfreq[i]), 0,)
    if len(R_RMSfreq[i]) < 4:
        R_RMSfreq[i] = np.insert(R_RMSfreq[i], len(R_RMSfreq[i]), 0,)
    if len(Domfreq[i]) < 4:
        Domfreq[i] = np.insert(Domfreq[i], len(Domfreq[i]), 0,)
    if len(R_Domfreq[i]) < 4:
        R_Domfreq[i] = np.insert(R_Domfreq[i], len(R_Domfreq[i]), 0,)
    if len(RMIFS[i]) < 4:
        RMIFS[i] = np.insert(RMIFS[i], len(RMIFS[i]), 0,)
    if len(R_RMIFS[i]) < 4:
        R_RMIFS[i] = np.insert(R_RMIFS[i], len(R_RMIFS[i]), 0,)

M_B.append(np.concatenate((RMSfreq[i][:4],R_RMSfreq[i][:4],Domfreq[i][:4],R_Domfreq[i][:4]
,RMIFS[i][:4],R_RMIFS[i][:4])))

    return M_B

#Panggil Sinyal Ictal
klik_S= []
klik_S=hasil("E:/1. Thesis/EEG/Program/Potong data/27des/DATA POTONGAN/V_S/")

def hasil (direktori):
    buka=buka_folder(direktori)
    bigE = tabel(buka)
    head = list(bigE.columns.values)
    bigE.head(6)
    matE = buat_data(bigE,head)
    matE = np.nan_to_num(matE)

    s = []
    IMFs = []
    c = []
    ampli=[]
    frek=[]
    bam=[]
    bfm=[]
    M_B = []
    RMSfreq=[]
    R_RMSfreq=[]
    Domfreq=[]
    R_Domfreq=[]
    RMIFS=[]
    R_RMIFS=[]

    for i in range(76):

```

**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```

print('File ke: ', i)
s.append(matE[:,i])
IMFs.append([i])
ampli.append([i])
frek.append([i])
bam.append([i])
bfm.append([i])
RMSfreq.append([i])
R_RMSfreq.append([i])
Domfreq.append([i])
R_Domfreq.append([i])
RMIFS.append([i])
R_RMIFS.append([i])
#IMFs[i] = EMD(s[i]).decompose()
IMFs[i]=emd(s[i],4)
c.append(np.nan_to_num(fitur(IMFs[i])))
ampli[i], frek[i] = c[i]
bam[i] = b_am (ampli[i],s[i])
bfm[i]= b_fm(frek[i], ampli[i],s[i])
RMSfreq[i],R_RMSfreq[i]= RMS_freq(frek[i], ampli[i],s[i])
# R_RMSfreq[i]= RMS_freq(frek[i], ampli[i],s[i])
Domfreq[i],R_Domfreq[i]= Dom_freq(frek[i], ampli[i],s[i])
#R_Domfreq[i]= Dom_freq(frek[i], ampli[i],s[i])
RMIFS[i],R_RMIFS[i]= RMI_fs(frek[i], ampli[i],s[i])
#R_RMIFS[i]= RMI_fs(frek[i], ampli[i],s[i])

while (len(RMSfreq[i]) < 4) or (len(R_RMSfreq[i]) < 4) or (len(Domfreq[i]) < 4) or
(len(R_Domfreq[i]) < 4) or (len(RMIFS[i]) < 4) or (len(R_RMIFS[i]) < 4):
    if len(RMSfreq[i]) < 4:
        RMSfreq[i] = np.insert(RMSfreq[i], len(RMSfreq[i]), 0,)
    if len(R_RMSfreq[i]) < 4:
        R_RMSfreq[i] = np.insert(R_RMSfreq[i], len(R_RMSfreq[i]), 0,)
    if len(Domfreq[i]) < 4:
        Domfreq[i] = np.insert(Domfreq[i], len(Domfreq[i]), 0,)
    if len(R_Domfreq[i]) < 4:
        R_Domfreq[i] = np.insert(R_Domfreq[i], len(R_Domfreq[i]), 0,)
    if len(RMIFS[i]) < 4:
        RMIFS[i] = np.insert(RMIFS[i], len(RMIFS[i]), 0,)
    if len(R_RMIFS[i]) < 4:
        R_RMIFS[i] = np.insert(R_RMIFS[i], len(R_RMIFS[i]), 0,)

M_B.append(np.concatenate((RMSfreq[i][:4],R_RMSfreq[i][:4],Domfreq[i][:4],R_Domfreq[i][:4]
,RMIFS[i][:4],R_RMIFS[i][:4])))

return M_B

co= np.zeros((44,1))
cl= np.ones((76,1))
def gabung1(klik,c):

```

**Lampiran 13. .** Script Ekstraksi Fitur *time frequency* untuk data pemotongan 5 detik (Lanjutan)

```
data = np.concatenate([np.round(np.concatenate([klik]).real, 3).reshape(44,24),c],axis=1)
total_data=
pd.DataFrame(data,columns=['fr1','fr2','fr3','fr4','R_fr1','R_fr2','R_fr3','R_fr4','fd1','fd2','fd3','fd4','
R_fd1','R_fd2','R_fd3','R_fd4','RMIFS1','RMIFS2','RMIFS3','RMIFS4','R_RMIFS1','R_RMIFS2','R
_RMIFS3','R_RMIFS4','Class'])
return (total_data)
def gabung2(klik,c):
data = np.concatenate([np.round(np.concatenate([klik]).real, 3).reshape(76,24),c],axis=1)
total_data=
pd.DataFrame(data,columns=['fr1','fr2','fr3','fr4','R_fr1','R_fr2','R_fr3','R_fr4','fd1','fd2','fd3','fd4','
R_fd1','R_fd2','R_fd3','R_fd4','RMIFS1','RMIFS2','RMIFS3','RMIFS4','R_RMIFS1','R_RMIFS2','R
_RMIFS3','R_RMIFS4','Class'])
return (total_data)

total_data_S1=gabung1(klik_S,co)
total_data_OA1=gabung2(klik_OA,cl)

#Menggabungkan hasil dari ekstraksi fitur
TotalDataset = pd.concat([total_data_S1,total_data_OA1],ignore_index=True)
visDat = TotalDataset.copy(deep=True)
visDat['Class'] = visDat['Class'].map({0:'Ictal',1:'OA'})
TotalDatseta = pd.concat([total_data_S1],ignore_index=True)
TotalDatsete = pd.concat([total_data_OA1],ignore_index=True)

visDat.to_csv('Ekstrak_IctalvsOA_TfF_5_String_NEW.csv')
TotalDataset.to_csv('Ekstrak_IctalvsOA_TfF_5_NEW.csv')
```

## Lampiran 14. Script *Tunning* Parameter *SVM*

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, StratifiedKFold, GridSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn import SVM

#POLYNOMIAL

X = X_11
def svc_param_selection(X, y, nfolds):
    Cs=[0.001, 0.01, 0.1, 1, 10, 10,100,1000,10000]
    #Cs =[0.001, 0.025, 0.01, 0.1, 0.25, 1, 10, 100]
    degrees = [1, 2, 3]
    param_grid = {'C': Cs, 'degree' : degrees}
    grid_search = GridSearchCV(SVM.SVC(kernel='poly'), param_grid, cv=nfolds)
    grid_search.fit(X, y)
    grid_search.best_params_
    return grid_search.best_params_
svc_param_selection(X, y, 10)

X = X5
def svc_param_selection(X, y, nfolds):
    Cs = [0.001, 0.01, 0.1, 1, 10, 10,100,1000,10000]
    gammas = [0.001, 0.01, 0.1, 1, 10,100,1000,10000]
    param_grid = {'C': Cs, 'gamma' : gammas}
    grid_search = GridSearchCV(SVM.SVC(kernel='rbf'), param_grid, cv=nfolds)
    grid_search.fit(X, y)
    grid_search.best_params_
    return grid_search.best_params_
svc_param_selection(X, y, 10)

X = X7
def svc_param_selection(X, y, nfolds):
    Cs = [0.001, 0.025, 0.01, 0.1, 0.25, 1, 10, 100]
    param_grid = {'C': Cs}
    grid_search = GridSearchCV(SVM.SVC(kernel='linear'), param_grid, cv=nfolds)
    grid_search.fit(X, y)
    grid_search.best_params_
    return grid_search.best_params_
svc_param_selection(X, y, 10)
```

## Lampiran 15. Script Klasifikasi Sinyal *Ictal* dan Sinyal OA menggunakan *SVM*

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV,
cross_val_score, StratifiedKFold
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import confusion_matrix, accuracy_score, log_loss
from sklearn import SVM
import time

TotalDataset= pd.read_csv(r'/media/baiqsiska/Ical/BY/Persiapan
Sidang/Ekstrak_IctalvsOA_Tff_5.csv', index_col=None)
#%#%
##Kernel RBF
def SVM_rbf(X1, y1,C1,Gamma,n):
    k1=StratifiedKFold(n_splits=n)
    k1.get_n_splits(X1, y1)
    acc=[]
    spe=[]
    sen=[]
    AUC=[]
    klas=[]
    for train_index, test_index in k1.split(X1, y1):
        #print("TRAIN:", train_index, "TEST:", test_index)
        X1_train, X1_test = X1[train_index], X1[test_index]
        y1_train, y1_test = y1[train_index], y1[test_index]
        m1_clf =SVM.SVC(C=C1,gamma=Gamma,kernel='rbf')
        m1_clf.fit(X1[train_index], y1[train_index])
        m1_pred = m1_clf.predict(X1[test_index])
        tn, fp, fn, tp= list(confusion_matrix(y_true = y1[test_index],
                                             y_pred = m1_clf.predict(X1[test_index])).ravel())
        tn = float(tn);i_tn=("0.0f" % (tn));klas.append(i_tn)
        fp = float(fp);i_fp=("0.0f" % (fp));klas.append(i_fp)
        fn = float(fn);i_fn=("0.0f" % (fn));klas.append(i_fn)
        tp = float(tp);i_tp=("0.0f" % (tp));klas.append(i_tp)
        akurasi = ((tp+tn)/(tp+tn+fp+fn))*100;klas.append(akurasi)
        spesifikasi = (tn/(fp+tn))*100;klas.append(spesifikasi)
        sensitifitas = (tp/(fn+tp))*100;klas.append(sensitifitas)
        AUC=((tp/(fn+tp))*100)+((tn/(fp+tn))*100)/2;klas.append(AUC)
        acc.append(akurasi)
        spe.append(spesifikasi)
        sen.append(sensitifitas)
        AUC.append(AUC)
    data = np.reshape(klas, (n,8))
    total_data=
pd.DataFrame(data,columns=["TN",'FP','FN','TP','Akurasi','Spesifitas','Sensitifitas','AUC'])
    print("min: %0.2f %0.2f %0.2f %0.2f "
%(np.min(acc),np.min(sen),np.min(spe),np.min(AUC)))
    print("max: %0.2f %0.2f %0.2f %0.2f "
%(np.max(acc),np.max(sen),np.max(spe),np.max(AUC)))
```

**Lampiran 15** Script Klasifikasi Sinyal *Ictal* dan Sinyal OA menggunakan *SVM* (Lanjutan)

```

print("rata-rata: %0.2f %0.2f %0.2f %0.2f" %
(np.mean(acc),np.mean(sen),np.mean(spe),np.mean(AUC)))
return(total_data)

def SVM_linear(X1, y1,C1,n):
    k1=StratifiedKFold(n_splits=n)
    k1.get_n_splits(X1, y1)
    acc=[]
    spe=[]
    sen=[]
    AUC=[]
    klas=[]
    for train_index, test_index in k1.split(X1, y1):
        #print("TRAIN:", train_index, "TEST:", test_index)
        X1_train, X1_test = X1[train_index], X1[test_index]
        y1_train, y1_test = y1[train_index], y1[test_index]
        m1_clf=SVM.SVC(C=C1,kernel='linear')
        m1_clf.fit(X1[train_index], y1[train_index])
        m1_pred = m1_clf.predict(X1[test_index])
        tn, fp, fn, tp= list(confusion_matrix(y_true = y1[test_index],
                                             y_pred = m1_clf.predict(X1[test_index])).ravel())
        tn = float(tn);i_tn=("%.0f" %(tn));klas.append(i_tn)
        fp = float(fp);i_fp=("%.0f" %(fp));klas.append(i_fp)
        fn = float(fn);i_fn=("%.0f" %(fn));klas.append(i_fn)
        tp = float(tp);i_tp=("%.0f" %(tp));klas.append(i_tp)
        akurasi = ((tp+tn)/(tp+tn+fp+fn))*100;klas.append(akurasi)
        spesifikasi = (tn/(fp+tn))*100;klas.append(spesifikasi)
        sensitifitas = (tp/(fn+tp))*100;klas.append(sensitifitas)
        AUC=((((tp/(fn+tp))*100)+((tn/(fp+tn))*100))/2);klas.append(AUC)
        acc.append(akurasi)
        spe.append(spesifikasi)
        sen.append(sensitifitas)
        AUC.append(AUC)

    data = np.reshape(klas, (n,8))
    total_data=
pd.DataFrame(data,columns=['TN','FP','FN','TP','Akurasi','Spesifitas','Sensitifitas','AUC'])
    print("min: %0.2f %0.2f %0.2f %0.2f "
%(np.min(acc),np.min(spe),np.min(sen),np.min(AUC)))
    print("max: %0.2f %0.2f %0.2f %0.2f "
%(np.max(acc),np.max(spe),np.max(sen),np.max(AUC)))
    print("rata-rata: %0.2f %0.2f %0.2f %0.2f" %
(np.mean(acc),np.mean(spe),np.mean(sen),np.mean(AUC)))
    return(total_data)

def SVM_Polynom(X1, y1,C,degree1,n):
    k1=StratifiedKFold(n_splits=n)
    k1.get_n_splits(X1, y1)
    acc=[]
    spe=[]
    sen=[]

```



**Lampiran 15** Script Klasifikasi Sinyal *Ictal* dan Sinyal OA menggunakan *SVM* (Lanjutan)

```

AUC=[]
klas=[]
for train_index, test_index in k1.split(X1, y1):
    #print("TRAIN:", train_index, "TEST:", test_index)
    X1_train, X1_test = X1[train_index], X1[test_index]
    y1_train, y1_test = y1[train_index], y1[test_index]
    m1_clf =SVM.SVC(C=C,degree=degree1, kernel='poly')
    m1_clf.fit(X1[train_index], y1[train_index])
    m1_pred = m1_clf.predict(X1[test_index])
    tn, fp, fn, tp= list(confusion_matrix(y_true = y1[test_index],
                                         y_pred = m1_clf.predict(X1[test_index])).ravel())
    tn = float(tn);i_tn="%0.0f" %(tn);klas.append(i_tn)
    fp = float(fp);i_fp="%0.0f" %(fp);klas.append(i_fp)
    fn = float(fn);i_fn="%0.0f" %(fn);klas.append(i_fn)
    tp = float(tp);i_tp="%0.0f" %(tp);klas.append(i_tp)
    akurasi = ((tp+tn)/(tp+tn+fp+fn))*100;klas.append(akurasi)
    spesifikasi = (tn/(fp+tn))*100;klas.append(spesifikasi)
    sensitifitas = (tp/(fn+tp))*100;klas.append(sensitifitas)
    AUC=((tp/(fn+tp))*100)+((tn/(fp+tn))*100)/2;klas.append(AUC)
    acc.append(akurasi)
    spe.append(spesifikasi)
    sen.append(sensitifitas)
    AUC.append(AUC)
data = np.reshape(klas, (n,8))
total_data=
pd.DataFrame(data,columns=['TN','FP','FN','TP','Akurasi','Spesificity','Sensitifity','AUC'])
print("min: %0.2f %0.2f %0.2f %0.2f "
      %(np.min(acc),np.min(spe),np.min(sen),np.min(AUC)))
print("max: %0.2f %0.2f %0.2f %0.2f "
      %(np.max(acc),np.max(spe),np.max(sen),np.max(AUC)))
print("rata-rata: %0.2f %0.2f %0.2f %0.2f " %
      (np.mean(acc),np.mean(spe),np.mean(sen),np.mean(AUC)))
return(total_data)

```

**Lampiran 16.** Hasil tuning parameter dan Kebaikan Klasifikasi dengan *input* fitur  $B_{AM}$  dan  $B_{FM}$  untuk data potongan 2 detik

<i>IMF</i> ke-	Kombinasi Fitur	Kernel	Kebaikan Klasifikasi			
		Linier	Accuracy	Specificity	Sensitifity	<i>AUC</i>
		C				
<i>IMF</i> ke-1	X1.1	0,001	55,53	0	100	50
	X2.1	100	94,15	96	92,56	94,28
	X1.1, X2.1	10	93,74	98	90,19	94,1
<i>IMF</i> ke-2	X1.2	0,001	55,53	0	100	50
	X2.2	100	86,98	94	81,15	87,58
	X1.2, X2.2	100	91,05	96	86,99	91,49
<i>IMF</i> ke-3	X1.3	0,001	55,53	0	100	50
	X2.3	100	82,51	93	73,91	83,46
	X1.3, X2.3	10	86,56	96	78,78	87,39
<i>IMF</i> ke-4	X1.4	0,001	55,53	0	100	50
	X2.4	100	81,7	92	73,27	82,63
	X1.4, X2.4	100	81,7	92	73,27	82,63
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4	10	57,75	20	88,21	54,1
<i>IMF</i> 1-4	X2.1, X2.2, X2.3, X2.4	100	94,58	97	92,5	94,75
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4	100	96,4	97	95,9	96,45

<i>IMF</i> ke-	Kombinasi Fitur	Kernel		Kebaikan Klasifikasi			
		RBF		Acc.	Spec.	Sens.	<i>AUC</i>
		C	Sigma				
<i>IMF</i> ke-1	X1.1	1	10	59,53	76,79	38	57,4
	X2.1	10	100	95,49	95	96	95,5
	X1.1, X2.1	100	1	91,98	92,63	91	91,81
<i>IMF</i> ke-2	X1.2	100	10000	60,1	69,68	48	58,84
	X2.2	100	100	88,32	84,42	93	88,71
	X1.2, X2.2	1000	0,1	89,74	84,62	96	90,31
<i>IMF</i> ke-3	X1.3	0,001	0,001	55,53	100	0	50
	X2.3	10	1000	84,35	78,85	91	84,92
	X1.3, X2.3	10	0,1	86,58	78,01	97	87,51
<i>IMF</i> ke-4	X1.4	100	100	63,18	75,45	48	61,72
	X2.4	1	10	81,68	71,67	94	82,83
	X1.4, X2.4	1000	0,01	81,7	73,27	92	82,63
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4	1000	0,1	70,2	74,55	65	69,78
<i>IMF</i> 1-4	X2.1, X2.2, X2.3, X2.4	1	100	94,62	95,83	93	94,42
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4	1000	0,01	96,86	96,67	97	96,83

## Lampiran 16

Hasil tuning parameter dan Kebaikan Klasifikasi dengan *input* fitur  $B_{AM}$  dan  $B_{FM}$  untuk data potongan 2 detik (Lanjutan)

<i>IMF</i> ke-	Kombinasi Fitur	Kernel		Kebaikan Klasifikasi			
		<i>Polynomial</i>		Acc.	Spec.	Sens.	<i>AUC</i>
		C	Degree				
<i>IMF</i> ke-1	X1.1	0,001	1	55,53	0	100	50
	X2.1	100	1	94,15	96	92,56	94,28
	X1.1, X2.1	10	1	93,74	98	90,19	94,1
<i>IMF</i> ke-2	X1.2	0,001	1	55,53	0	100	50
	X2.2	100	1	86,98	94	81,15	87,58
	X1.2, X2.2	100	1	91,05	96	86,99	91,49
<i>IMF</i> ke-3	X1.3	0,001	1	55,53	0	100	50
	X2.3	100	1	82,51	93	73,91	83,46
	X1.3, X2.3	10	1	86,56	96	78,78	87,39
<i>IMF</i> ke-4	X1.4	0,001	1	55,53	0	100	50
	X2.4	100	1	81,7	92	73,27	82,63
	X1.4, X2.4	100	1	81,7	92	73,27	82,63
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4	10	1	57,75	20	88,21	54,1
<i>IMF</i> 1-4	X2.1, X2.2, X2.3, X2.4	100	1	94,58	97	92,5	94,75
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4	100	1	96,4	97	95,9	96,45

**Lampiran 17.** Hasil tuning parameter dan Keباikan Klasifikasi dengan *input* fitur  $B_{AM}$  dan  $B_{FM}$  untuk data potongan 5 detik

IMF ke-	Kombinasi Fitur	Linier	Keباikan Klasifikasi			
		C	Accuracy	Specificity	Sensitifity	AUC
IMF ke-1	X1.1	0,001	63,4	0	100	50
	X2.1	10	80,19	100	68,75	84,38
	X1.1, X2.1	0,001	63,67	0	100	50
IMF ke-2	X1.2	0,001	63,4	0	100	50
	X2.2	100	80,19	100	68,75	84,38
	X1.2, X2.2	100	80,19	98	70	84
IMF ke-3	X1.3	0,001	63,4	0	100	50
	X2.3	100	78,59	100	66,25	83,12
	X1.3, X2.3	100	78,59	100	66,25	83,12
IMF ke-4	X1.4	0,001	63,4	0	100	50
	X2.4	100	77,76	100	65	82,5
	X1.4, X2.4	1	76,99	100	63,75	81,88
IMF 1-4	X1.1, X1.2, X1.3, X1.4	0,001	63,4	0	100	50
IMF 1-4	X2.1, X2.2, X2.3, X2.4	10	80,19	100	68,75	84,38
IMF 1-4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4	100	81,03	96	72,5	84,25

IMF ke-	Kombinasi Fitur	Kernel		Keباikan Klasifikasi			
		RBF		Acc.	Spec.	Sens.	AUC
		C	Sigma				
IMF ke-1	X1.1	1	10	69	70,36	66,5	68,43
	X2.1	1	1000	83,27	75	98	86,5
	X1.1, X2.1	1	10	81,73	71,25	100	85,62
IMF ke-2	X1.2	10	10	72,49	71,07	74,5	72,79
	X2.2	10	100	83,26	77,5	93,5	85,5
	X1.2, X2.2	0,1	10	81,73	73,75	96	84,88
IMF ke-3	X1.3	0,001	0,001	63,4	100	0	50
	X2.3	0,1	100	82,35	77,5	90,5	84
	X1.3, X2.3	1	100	83,68	80	90	85
IMF ke-4	X1.4	0,001	0,001	63,4	100	0	50
	X2.4	10	1000	80,9	72,5	96	84,25
	X1.4, X2.4	10	100	81,11	78,75	85,5	82,12
IMF 1-4	X1.1, X1.2, X1.3, X1.4	10	1	69,42	67,68	72	69,84
IMF 1-4	X2.1, X2.2, X2.3, X2.4	1	100	82,21	81,25	84	82,62
IMF 1-4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4	0,1	1	81,67	71,25	100	85,62

**Lampiran 17** Hasil tuning parameter dan Keباikan Klasifikasi dengan *input* fitur  $B_{AM}$  dan  $B_{FM}$  untuk data potongan 5 detik (Lanjutan)

<i>IMF</i> ke-	Kombinasi Fitur	Kernel		Keباikan Klasifikasi			
		<i>Polynomial</i>		Accuracy	Specificity	Sensitifity	<i>AUC</i>
		C	Degree				
<i>IMF</i> ke-1	X1.1	0,001	1	63,4	0	100	50
	X2.1	10	1	80,19	100	68,75	84,38
	X1.1, X2.1	100	1	80,19	100	68,75	84,38
<i>IMF</i> ke-2	X1.2	0,001	1	63,4	0	100	50
	X2.2	100	1	80,19	100	68,75	84,38
	X1.2, X2.2	1000	1	80,19	98	70	84
<i>IMF</i> ke-3	X1.3	0,001	1	63,4	0	100	50
	X2.3	1000	3	80,13	100	68,75	84,38
	X1.3, X2.3	1000	3	79,36	100	67,5	83,75
<i>IMF</i> ke-4	X1.4	0,001	1	63,4	0	100	50
	X2.4	100	1	77,76	100	65	82,5
	X1.4, X2.4	1	1	76,99	100	63,75	81,88
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4	0,001	1	63,4	0	100	50
<i>IMF</i> 1-4	X2.1, X2.2, X2.3, X2.4	10	1	80,19	100	68,75	84,38
<i>IMF</i> 1-4	X1.1, X1.2, X1.3, X1.4, X2.1, X2.2, X2.3, X2.4	1000	1	81,03	96	72,5	84,25

**Lampiran 18.** Hasil tuning parameter dan Keباikan Klasifikasi dengan *input* fitur *Time Frequency* untuk data potongan 2 detik

<i>IMF</i>		Linier C	Keباikan Klasifikasi			
			Acc.	Spec.	Sens.	AUC
IMF 1	X3.1	1000	70,51	76	66,28	71,1
	X4.1					
	X5.1	0,001	55,53	0	100	50
	X6.1	1000	87,81	95	81,67	88,3
	X7.1	0,1	97,31	99	95,9	97,5
	X8.1	0,001	55,53	0	100	50
	X3.1, X4.1, X5.1, X6.1, X7.1, X8.1					
IMF 2	X3.2	1	73,68	77	71,03	74
	X4.2					
	X5.2	0,001	55,53	0	100	50
	X6.2	1	80,3	97	66,6	81,8
	X7.2	10	97,77	98	97,5	97,8
	X8.2	0,001	55,53	0	100	50
	X3.2, X4.2, X5.2, X6.2, X7.2, X8.2					
IMF 3	X3.3	1	95,45	100	100	95
	X4.3					
	X5.3	0,01	55,53	0	100	50
	X6.3	10	79,86	91	70,71	80,9
	X7.3	0,1	97,31	99	95,9	97,5
	X8.3	0,001	55,53	0	100	50
	X3.3, X4.3, X5.3, X6.3, X7.3, X8.3					
IMF 4	X3.4	0,01	72,31	71	73,01	72
	X4.4					
	X5.4	0,001	55,53	0	100	50
	X6.4	1	87,53	96	80,71	88,4
	X7.4	10	97,31	99	95,9	97,5
	X8.4	0,001	55,53	0	100	50
	X3.4, X4.4, X5.4, X6.4, X7.4, X8.4					
IMF 1-4	X3.1, X3.2, X3.3, X3.4	1	76,4	82	71,67	76,8
	X4.1, X4.2, X4.3, X4.4					
	X5.1, X5.2, X5.3, X5.4	0,001	55,53	0	100	50
	X6.1, X6.2, X6.3, X6.4	10	91,92	97	87,69	92,4
	X7.1, X7.2, X7.3, X7.4	1000	98,22	99	97,56	98,3
	X8.1, X8.2, X8.3, X8.4	0,001	55,53	0	100	50
	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4 X5.1, X5.2, X5.3, X5.4, X6.1, X6.2, X6.3, X6.4 X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4	1000				

**Lampiran 18** Hasil tuning parameter dan Kebaikan Klasifikasi dengan *input* fitur *Time Frequency* untuk data potongan 2 detik (Lanjutan)

<i>IMF</i>		RBF		Kebajikan Klasifikasi			
		C	sigma	Accuracy	Specificity	Sensitifity	AUC
<i>IMF 1</i>	X3.1	1	0,01	70,06	63,01	79	71,01
	X4.1						
	X5.1	1000	0,1	58,32	56,28	61	58,64
	X6.1	100	1	88,26	82,5	95	88,75
	X7.1	0,1	1	97,75	96,67	99	97,83
	X8.1	100	1000	60,77	44,04	82	63,02
	X3.1, X4.1, X5.1, X6.1, X7.1, X8.1						
<i>IMF 2</i>	X3.2	10	10	76,42	79,23	73	76,12
	X4.2						
	X5.2	100	1000	59,49	67,82	49	58,41
	X6.2	0,1	100	84,35	83,4	85	84,2
	X7.2	10	0,1	97,77	97,5	98	97,75
	X8.2	0,001	0,001	55,53	100	0	50
	X3.2, X4.2, X5.2, X6.2, X7.2, X8.2						
<i>IMF 3</i>	X3.3	100	0,1	78,7	91,03	63	77,01
	X4.3						
	X5.3	1	10	56,05	74,36	33	53,68
	X6.3	0,1	10	83	80,51	86	83,26
	X7.3	0,1	10	97,77	97,5	98	97,75
	X8.3	0,001	0,001	55,53	100	0	50
	X3.3, X4.3, X5.3, X6.3, X7.3, X8.3						
<i>IMF 4</i>	X3.4	0,1	10	72,43	94,23	45	69,62
	X4.4						
	X5.4	1	10	58,68	76,79	36	56,4
	X6.4	1000	0,001	87,53	80,71	96	88,35
	X7.4	100	0,1	97,31	95,9	99	97,45
	X8.4	0,001	0,001	55,53	100	0	50
	X3.4, X4.4, X5.4, X6.4, X7.4, X8.4						
<i>IMF 1-4</i>	X3.1, X3.2, X3.3, X3.4	1	1	81,72	85,96	76	80,98
	X4.1, X4.2, X4.3, X4.4						
	X5.1, X5.2, X5.3, X5.4	1000	0,01	66,32	69,74	62	65,87
	X6.1, X6.2, X6.3, X6.4	1	0,1	91,46	87,69	96	91,85
	X7.1, X7.2, X7.3, X7.4	1000	1	96,03	97,56	94	95,78
	X8.1, X8.2, X8.3, X8.4	1000	100	59,82	44,68	79	61,84
	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4, X5.1, X5.2, X5.3, X5.4, X6.1, X6.2, X6.3, X6.4, X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4	0,001	0,001	55,53	100	0	50

**Lampiran 18** Hasil tuning parameter dan Kebaikan Klasifikasi dengan *input* fitur *Time Frequency* untuk data potongan 2 detik (Lanjutan)

<i>IMF</i>		<i>Polynomial</i>		Kebaikan Klasifikasi			
				Acc.	Spec.	Sens	<i>AUC</i>
		C	Degree				
<i>IMF 1</i>	X3.1	1000	1	70,51	76	66,28	71,14
	X4.1						
	X5.1	0,001	1	55,53	0	100	50
	X6.1	1000	1	87,81	95	81,67	88,33
	X7.1	0,1	1	97,31	99	95,9	97,45
	X8.1	0,001	1	55,53	0	100	50
	X3.1, X4.1, X5.1, X6.1, X7.1, X8.1						
<i>IMF 2</i>	X3.2	1	1	73,68	77	71,03	74,01
	X4.2						
	X5.2	0,001	1	55,53	0	100	50
	X6.2	10	100	80,3	97	66,6	81,8
	X7.2	10	1	97,77	98	97,5	97,75
	X8.2	0,001	1	55,53	0	100	50
	X3.2, X4.2, X5.2, X6.2, X7.2, X8.2						
<i>IMF 3</i>	X3.3	1	1	74,68	72	76,6	74,3
	X4.3						
	X5.3	0,01	1	55,53	0	100	50
	X6.3	10	1	79,86	91	70,71	80,85
	X7.3	0,1	1	97,31	99	95,9	97,45
	X8.3	0,001	1	55,53	0	100	50
	X3.3, X4.3, X5.3, X6.3, X7.3, X8.3						
<i>IMF 4</i>	X3.4	0,01	1	72,31	71	73,01	72,01
	X4.4						
	X5.4	0,01	1	55,53	0	100	50
	X6.4	1	1	87,53	96	80,71	88,35
	X7.4	10	1	97,31	99	95,9	97,45
	X8.4	0,001	1	55,53	0	100	50
	X3.4, X4.4, X5.4, X6.4, X7.4, X8.4						
<i>IMF 1-4</i>	X3.1, X3.2, X3.3, X3.4	1	2	81,74	86	78,08	82,04
	X4.1, X4.2, X4.3, X4.4						
	X5.1, X5.2, X5.3, X5.4	100	3	56,44	69	46,6	57,8
	X6.1, X6.2, X6.3, X6.4	10	1	91,92	97	87,69	92,35
	X7.1, X7.2, X7.3, X7.4	1000	2	98,68	99	98,4	98,7
	X8.1, X8.2, X8.3, X8.4	0,001	1	55,53	0	100	50
	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4						
	X5.1, X5.2, X5.3, X5.4, X6.1, X6.2, X6.3, X6.4						
X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4	0,001	1	94,6	96	93,46	94,73	



**Lampiran 19.** Hasil tuning parameter dan Keباian Klasifikasi dengan *input* fitur Time Frequency untuk data potongan 5 detik

IMF	Fitur	Linear	Keباian Klasifikasi			
		C	Accuracy	Specificity	Sensitifity	AUC
IMF 1	X3.1	1	70,34	83	62,5	72,75
	X4.1					
	X5.1	0,001	63,4	0	100	50
	X6.1	1000	80,12	91,5	73,75	82,62
	X7.1	1	80,96	100	70	85
	X8.1	0,001	63,4	0	100	50
	X3.1, X4.1, X5.1, X6.1, X7.1, X8.1	0,1	74,12	90,5	64,82	77,66
IMF 2	X3.2	10	73,83	89	65	77
	X4.2					
	X5.2	0,001	63,4	0	100	50
	X6.2	10	81,79	98	72,5	85,25
	X7.2	10	80,96	100	70	85
	X8.2	0,001	63,4	0	100	50
	X3.2, X4.2, X5.2, X6.2, X7.2, X8.2					
IMF 3	X3.3	1	72,08	84	65	74,5
	X4.3					
	X5.3	0,001	63,4	0	100	50
	X6.3	1	79,35	98	68,57	83,29
	X7.3	1	80,96	100	70	85
	X8.3	0,001	63,4	0	100	50
	X3.3, X4.3, X5.3, X6.3, X7.3, X8.3	f				
IMF 4	X3.4	10	76,7	84	72,5	78,25
	X4.4					
	X5.4	0,001		63,4	0	100
	X6.4	0,1	79,29	100	67,5	83,75
	X7.4	1	80,19	100	68,75	84,38
	X8.4	0,001	63,4	0	100	50
	X3.4, X4.4, X5.4, X6.4, X7.4, X8.4	y				
IMF 1-4	X3.1, X3.2, X3.3, X3.4	0,1	73,07	97,5	58,75	78,12
	X4.1, X4.2, X4.3, X4.4					
	X5.1, X5.2, X5.3, X5.4	0,001	63,4	0	100	50
	X6.1, X6.2, X6.3, X6.4	1000	80,26	96	71,25	83,62
	X7.1, X7.2, X7.3, X7.4	1	80,96	100	70	85
	X8.1, X8.2, X8.3, X8.4	0,001	63,4	0	100	50
	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4					
	X5.1, X5.2, X5.3, X5.4, X6.1, X6.2, X6.3, X6.4					
X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4						

**Lampiran 19.** Hasil tuning parameter dan Keباikan Klasifikasi dengan *input* fitur Time Frequency untuk data potongan 5 detik (Lanjutan)

IMF	Fitur	RBF		Keباikan Klasifikasi			
		C	Gamma	Acc.	Spec.	Sens.	AUC
IMF 1	X3.1	1000	0,01	73,69	62,32	93,5	77,91
	X4.1	1	0,01	67,84	88,75	32	60,38
	X5.1	1000	0,01	64,94	98,75	6	52,38
	X6.1	10	1000	83,33	75	98	86,5
	X7.1	0,1	100	85,71	80	96	88
	X8.1	0,001	0,001	63,4	100	0	50
	X3.1, X4.1, X5.1, X6.1, X7.1, X8.1	1	0,01	67,84	88,75	32	60,38
IMF 2	X3.2	1	10	78,31	80	75,5	77,75
	X4.2	1	0,001	66,1	88,75	27	57,88
	X5.2	1	10	70,96	72,5	68,5	70,5
	X6.2	10	100	83,33	76,25	96	86,12
	X7.2	0,1	100	85,71	80	96	88
	X8.2	0,001	0,001	63,4	100	0	50
	X3.2, X4.2, X5.2, X6.2, X7.2, X8.2	1	0,001	66,1	88,75	27	57,88
IMF 3	X3.3	1	10	78,15	81,25	72,5	76,88
	X4.3	10	0,001	68,47	88,75	33,5	61,12
	X5.3	0,001	0,001	63,4	100	0	50
	X6.3	0,1	10	81,79	72,5	98	85,25
	X7.3	1	1000	85,71	80	96	88
	X8.3	0,001	0,001	63,4	100	0	50
	X3.3, X4.3, X5.3, X6.3, X7.3, X8.3	10	0,001	68,47	88,75	33,5	61,12
IMF 4	X3.4	1	100	82,21	81,25	84	82,62
	X4.4	1	0,001	66,1	88,75	27	57,88
	X5.4	50	1000	66,92	76,79	49,5	63,14
	X6.4	10	100	82,22	76,25	93	84,62
	X7.4	0,1	100	85,71	80	96	88
	X8.4	0,001	0,001	63,4	100	0	50
	X3.4, X4.4, X5.4, X6.4, X7.4, X8.4	1	0,001	66,1	88,75	27	57,88
IMF 1-4	X3.1, X3.2, X3.3, X3.4	100	1	73,84	78,75	66	72,38
	X4.1, X4.2, X4.3, X4.4	1	0,001	66,1	88,75	27	57,88
	X5.1, X5.2, X5.3, X5.4	1	10	65,89	81,96	38	59,98
	X6.1, X6.2, X6.3, X6.4	0,1	1	81,79	72,5	98	85,25
	X7.1, X7.2, X7.3, X7.4	0,1	10	84,87	78,75	96	87,38
	X8.1, X8.2, X8.3, X8.4	0,001	0,001	63,4	100	0	50
	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4, X5.1, X5.2, X5.3, X5.4, X6.1, X6.2, X6.3, X6.4, X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4	1	0,001	66,1	88,75	27	57,88

**Lampiran 19.** Hasil tuning parameter dan Kebaikan Klasifikasi dengan *input* fitur Time Frequency untuk data potongan 5 detik (Lanjutan)

IMF	Fitur	RBF		Kebajikan Klasifikasi			
		C	Gamma	Acc.	Spec.	Sens.	AUC
IMF 1	X3.1	1000	0,01	73,69	62,32	93,5	77,91
	X4.1	1	0,01	67,84	88,75	32	60,38
	X5.1	1000	0,01	64,94	98,75	6	52,38
	X6.1	10	1000	83,33	75	98	86,5
	X7.1	0,1	100	85,71	80	96	88
	X8.1	0,001	0,001	63,4	100	0	50
	X3.1, X4.1, X5.1, X6.1, X7.1, X8.1	1	0,01	67,84	88,75	32	60,38
IMF 2	X3.2	1	10	78,31	80	75,5	77,75
	X4.2	1	0,001	66,1	88,75	27	57,88
	X5.2	1	10	70,96	72,5	68,5	70,5
	X6.2	10	100	83,33	76,25	96	86,12
	X7.2	0,1	100	85,71	80	96	88
	X8.2	0,001	0,001	63,4	100	0	50
	X3.2, X4.2, X5.2, X6.2, X7.2, X8.2	1	0,001	66,1	88,75	27	57,88
IMF 3	X3.3	1	10	78,15	81,25	72,5	76,88
	X4.3	10	0,001	68,47	88,75	33,5	61,12
	X5.3	0,001	0,001	63,4	100	0	50
	X6.3	0,1	10	81,79	72,5	98	85,25
	X7.3	1	1000	85,71	80	96	88
	X8.3	0,001	0,001	63,4	100	0	50
	X3.3, X4.3, X5.3, X6.3, X7.3, X8.3	10	0,001	68,47	88,75	33,5	61,12
IMF 4	X3.4	1	100	82,21	81,25	84	82,62
	X4.4	1	0,001	66,1	88,75	27	57,88
	X5.4	50	1000	66,92	76,79	49,5	63,14
	X6.4	10	100	82,22	76,25	93	84,62
	X7.4	0,1	100	85,71	80	96	88
	X8.4	0,001	0,001	63,4	100	0	50
	X3.4, X4.4, X5.4, X6.4, X7.4, X8.4	1	0,001	66,1	88,75	27	57,88
IMF 1-4	X3.1, X3.2, X3.3, X3.4	100	1	73,84	78,75	66	72,38
	X4.1, X4.2, X4.3, X4.4	1	0,001	66,1	88,75	27	57,88
	X5.1, X5.2, X5.3, X5.4	1	10	65,89	81,96	38	59,98
	X6.1, X6.2, X6.3, X6.4	0,1	1	81,79	72,5	98	85,25
	X7.1, X7.2, X7.3, X7.4	0,1	10	84,87	78,75	96	87,38
	X8.1, X8.2, X8.3, X8.4	0,001	0,001	63,4	100	0	50
	X3.1, X3.2, X3.3, X3.4, X4.1, X4.2, X4.3, X4.4, X5.1, X5.2, X5.3, X5.4, X6.1, X6.2, X6.3, X6.4, X7.1, X7.2, X7.3, X7.4, X8.1, X8.2, X8.3, X8.4	1	0,001	66,1	88,75	27	57,88

**Lampiran 20.** Salah satu contoh catatan dari dokter

Contoh catatan dokter yang boleh ditampilkan. Untuk catatan yang lainnya tidak diperkenankan untuk dipublish

Epoch	Elapsed	Time	Ket
7/242	01.04	11.50.52	Buka mata
8/242	01.13	11.51.01	Tutup mata
9/242	01.22	11.51.10	Buka mata
10/242	01.36	11.51.24	tutup mata
11/242	01.45	11.51.33	Buka mata
12/242	01.54	11.51.42	tutup mata
13/242	02.02	11.51.50	Buka mata
14/242	02.15	11.52.03	Buka lalu tutup mata
16/242	02.30	11.52.18	Melirik kanan kiri
17/242	02.41	11.52.29	Melirik kiri kanan
18/242	02.57	11.52.45	Berkedip terus tiap detik
21/242	03.23	11.53.11	sampai
22/242	03.35	11.53.23	kedutan otot sekitar mata mengikuti fohk/ cahaya lampu dg frek 5 Hz (terutama R: Fp2-F8 Fp2-F4)

## BIOGRAFI PENULIS



**Yoga Prastya Irfandi** dilahirkan di Kabupaten Lamongan, Jawa Timur pada tanggal 16 Juni 1993. Penulis merupakan anak pertama dari dua bersaudara dari pasangan Bapak Sarsan dan Ibu Sining. Penulis menempuh pendidikan SD MI Rembon (1999-2001), SD Negeri Lamongrejo IV (2001-2005), SMP Negeri 1 Ngimbang (2005-2008), dan SMA Negeri 03 Jombang (2008-2011). Penulis diterima menjadi mahasiswa di Program Studi DIII Statistika ITS(2011-2014), Program Studi S1 Lintas Jalur Statistika ITS (2014-2016). Kemudian penulis melanjutkan pendidikan untuk jenjang Magister di Departemen Statistika, Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2017. Penulis sangat menerima dengan tangan terbuka terhadap kritik dan saran yang bersifat membangun. Apabila ada kritik, saran, dan pertanyaan yang ingin diajukan kepada penulis, maka dapat menghubungi penulis melalui email [yogaprastyairfandi@gmail.com](mailto:yogaprastyairfandi@gmail.com).

(Halaman ini sengaja dikosongkan)