



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KI141502**

# **Pengenalan Sandi Morse dari Sinyal Electroencephalogram yang Direkam Perangkat Neurosky Mindwave**

**AHMAD HAYAM BRILIAN**  
NRP 5111100091

Dosen Pembimbing I  
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Dosen Pembimbing II  
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA 2015**



**FINAL PROJECT - KI141502**

**MORSE CODE RECOGNITION FROM  
ELECTROENCEPHALOGRAM SIGNALS  
RECORDED BY NEUROSKY MINDWAVE DEVICE**

**AHMAD HAYAM BRILIAN  
NRP 5111100091**

**Supervisor I  
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.**

**Supervisor II  
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA 2015**

**LEMBAR PENGESAHAN**  
**Pengenalan Sandi Morse dari Sinyal**  
**Electroencephalogram yang Direkam**  
**Perangkat Neurosky Mindwave**  
**TUGAS AKHIR**

**Diajukan Untuk Memenuhi Salah Satu Syarat**  
**Memperoleh Gelar Sarjana Komputer**  
**pada**

**Rumpun Mata Kuliah Komputasi Cerdas dan Visi**  
**Program Studi S-1 Jurusan Teknik Informatika**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**

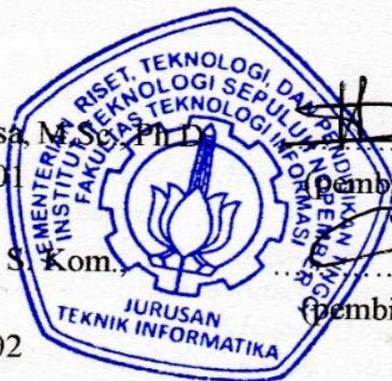
**Oleh:**

**AHMAD HAYAM BRILIAN**  
**NRP: 5111 100 091**

**Disetujui oleh Pembimbing Tugas Akhir:**

**Prof. Ir. Handayani Tjandrasa, M. Sc., Ph. D.**  
**NIP: 19490823 197603 2 001**

**Dr. Eng. Chastine Fatichah, S. Kom.**  
**M. Kom.**  
**NIP: 19751220 200112 2 002**



*[Signature]*  
Pembimbing 1)

*[Signature]*  
Pembimbing 2)

**SURABAYA**  
**JULI, 2015**

**Pengenalan Sandi Morse dari Sinyal  
Electroencephalogram yang Direkam  
Perangkat Neurosky Mindwave**

Nama : Ahmad Hayam Brilian  
NRP : 5111100091  
Jurusan : Teknik Informatika, FTIf-ITS  
Dosen Pembimbing I : Prof. Ir. Handayani Tjandrasa, M.Sc.,  
Ph.D.  
Dosen Pembimbing II : Dr. Eng. Chastine Fatichah, S. Kom.,  
M. Kom.

**Abstrak**

*Komunikasi terjadi antara makhluk hidup yang satu dengan yang lain. Komunikasi sehari-hari yang sering digunakan oleh manusia adalah komunikasi verbal. Namun, beberapa orang yang menderita kecacatan parah tidak mampu berkomunikasi dengan baik. Penderita cacat parah tidak mampu menuliskan kata yang dikehendaki, apalagi untuk berkomunikasi secara lisan karena mengalami gangguan syaraf. Namun beberapa penderita cacat parah masih bisa mengedipkan matanya dengan normal. Sehingga perlu dibangun sebuah metode yang mampu menerjemahkan kedipan mata menjadi kata verbal. Salah satu bentuk pengkodean yang sering dipakai adalah sandi morse. Pemilihan ini disebabkan karena pola-pola yang terdapat pada sandi morse dapat ditirukan dengan menggunakan kedipan mata.*

*Pada pengerjaan tugas akhir ini, sinyal electrooculogram (EOG) diekstrak dari sinyal electroencephalogram (EEG) yang didapatkan dari perangkat Neurosky Mindwave. Tahap pertama dalam pengerjaan tugas akhir ini adalah desain sistem penerima data dari perangkat. Data yang diperoleh akan diekstraksi dengan menggunakan filter bandpass. Filter bandpass cenderung memberikan tren dari sinyal EEG karena cukup baik dalam membersihkan noise. Tren sinyal yang didapatkan diasumsikan sebagai sinyal EOG. Dari hasil filterisasi, puncak dan lembah dideteksi untuk menentukan region of interest (ROI). ROI yang*

*telah tervalidasi oleh aturan-aturan yang telah ditetapkan akan mengalami pengklasteran oleh algoritma K-Means untuk ditentukan apakah elemen tersebut titik atau garis. Dengan menggunakan data yang diambil dari 3 subjek uji, didapatkan nilai rata-rata akurasi sebesar 89,2%.*

***Kata kunci: bandpass , EEG, EOG, kluster, ROI***

**MORSE CODE RECOGNITION FROM  
ELECTROENCEPHALOGRAM SIGNALS RECORDED  
BY NEROSKY MINDWAVE DEVICE**

Name : Ahmad Hayam Brilian  
NRP : 5111100091  
Department : Informatics Engineering, FTIf-ITS  
Supervisor I : Prof. Ir. Handayani Tjandrasa, M.Sc.,  
Ph.D.  
Supervisor II : Dr. Eng. Chastine Fatichah, S. Kom.,  
M. Kom.

***Abstract***

*Communication occurs among human. Verbal communication is used by human as a form of daily communication. But, there are people who suffered from severe disability and are not be able at doing communication. These people, can not write and say the words they want because of the nerve system disruption. But, many of them can control their eye movement well. So, there is a chance to build a method having ability to translate the eye blinking into words. One among the popular codification system is morse code. The choice of morse code is based on that the eye blinking can imitate the code pattern on the morse code system.*

*In this final project, electrooculogram (EOG) signals is extracted from electroencephalogram signals (EEG) taken from a Neurosky Mindwave device. First, the system for receiving data from the device must be designed and implemented. Then, the acquired data is extracted using bandpass filter. Bandpass filter tend to give a good signals' trend result. The achieved signals' trend is asumed as EOG signals. From the former process, peaks and valleys are detected to build region of interests (ROIs). After validating by the decisions system, ROIs will be clustered using K-*

*Means algorithm to determine whether it is dot or dash. The average of accuracy measurement with the use of dataset used in this final project is 89,2%.*

***Keywords: bandpass, cluster, EEG, EOG, ROI***

## KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “*Pengenalan Sandi Morse dari Sinyal Electroncephalogram yang Direkam Perangkat Neurosky Mindwave*” dengan tepat waktu.

Dalam pelaksanaan dan pembuatan tugas akhir ini penulis mendapatkan banyak bantuan dari berbagai pihak. Pada kesempatan ini, ucapan terima kasih penulis berikan kepada:

1. Allah SWT, karena atas limpahan rahmat-Nya, penulis diberikan kemudahan dan kelancaran dalam mengerjakan tugas akhir ini.
2. Orang tua dan keluarga penulis yang senantiasa memberikan doa dan dukungan kepada penulis untuk menyelesaikan pengerjaan tugas akhir ini.
3. Ibu Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D. dan Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. sebagai dosen pembimbing yang telah memberikan banyak arahan dan bantuan sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Bapak Dr. H. Agus Zainal Arifin, S.Kom., M.Kom. selaku dosen wali yang telah memberikan bimbingan dan nasihat selama masa perkuliahan.
5. Teman-teman *user* TA dan Administrator Laboratorium Komputasi Cerdas dan Visi, yang telah memberikan bantuan dalam penyelesaian tugas akhir ini.
6. Semua pihak yang telah membantu terselesaikannya tugas akhir ini.

Penulis menyadari adanya banyak kekurangan dalam pengerjaan tugas akhir baik dari segi program maupun laporan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk penyempurnaan tugas akhir ini. Akhir kata, penulis meminta maaf bila terdapat kesalahan dalam penulisan

laporan tugas akhir ini. Semoga hasil dari tugas akhir ini dapat memberikan manfaat bagi pembaca pada umumnya dan penulis pada khususnya.

Surabaya, Juni 2015

Ahmad Hayam B.

## DAFTAR ISI

LEMBAR PENGESAHAN .....	vii
Abstrak .....	ix
<i>Abstract</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
BAB I .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Metodologi .....	3
1.6 Sistematika Penulisan .....	4
BAB II .....	7
2.1 Sinyal <i>Electroencephalogram</i> (EEG) .....	7
2.2 <i>NeuroSky Mindwave</i> .....	8
2.3 Sandi Morse .....	9
2.4 <i>Empirical Mode Decomposition</i> (EMD) .....	10
2.5 Transformasi <i>Fourier</i> .....	13
2.6 Dynamic Time Warping (DTW) .....	14
2.7 <i>K-Nearest Neighbor</i> (kNN) <i>Classifier</i> .....	15
2.8 <i>K-Means Clustering</i> .....	17
2.9 <i>JSON String</i> .....	17
BAB III .....	19
3.1 Desain Metode Secara Umum .....	19
3.2 Perancangan Data .....	21
3.2.1 Data Masukan .....	21
3.2.2 Data Luaran .....	22

3.3	Perancangan Proses .....	23
3.3.1	Tahap Pengambilan Data .....	23
3.3.2	Contoh Data .....	26
3.3.3	Tahap Khusus Pengenalan Berdasarkan Aturan ..	27
3.3.4	Tahap Khusus Pengenalan Berdasarkan Pembelajaran .....	39
3.4	Tahap Evaluasi Metode .....	42
3.5	Perancangan Antarmuka Perangkat Lunak .....	43
<b>BAB IV</b>	.....	<b>47</b>
4.1	Lingkungan Implementasi .....	47
4.2	Implementasi Modul <i>NeuroskyConnector</i> .....	47
4.3	Implementasi Modul <i>StreamClass</i> .....	49
4.4	Implementasi Modul <i>MonitorClass</i> .....	50
4.5	Implementasi Modul Pendeteksi .....	51
4.6	Implementasi Modul Evaluasi .....	54
<b>BAB V</b>	.....	<b>59</b>
5.1	Lingkungan Uji Coba .....	59
5.2	Data Uji Coba .....	59
5.3	Uji Coba Proses Pengenalan.....	60
5.4	Skenario Uji Coba .....	63
5.4.1	Skenario Uji Coba 1 .....	64
5.4.2	Skenario Uji Coba 2 .....	71
5.4.3	Skenario Uji Coba 3 .....	82
5.4.4	Skenario Uji Coba 4 .....	90
5.4.5	Skenario Uji Coba 5 .....	98
5.5	Evaluasi .....	106
<b>BAB VI</b>	.....	<b>109</b>
6.1	Kesimpulan.....	109
6.2	Saran.....	109
<b>DAFTAR PUSTAKA</b>	.....	<b>111</b>
<b>LAMPIRAN</b>	.....	<b>115</b>
<b>BIODATA PENULIS</b>	.....	<b>127</b>

## DAFTAR TABEL

Tabel 2.1 Tabel parameter persamaan Transformasi Fourier	13
Tabel 3.1 Contoh data terekam dan pola sebenarnya.....	26
Tabel 5.1 Hasil skenario 1 dengan Lebar <i>window</i> 10 .....	64
Tabel 5.2 Hasil skenario 1 dengan Lebar <i>window</i> 20 .....	65
Tabel 5.3 Hasil skenario 1 dengan Lebar <i>window</i> 30 .....	66
Tabel 5.4 Hasil skenario 1 dengan Lebar <i>window</i> 40 .....	67
Tabel 5.5 Hasil skenario 1 dengan Lebar <i>window</i> 50 .....	67
Tabel 5.6 Rekapitulasi skenario 1 .....	69
Tabel 5.7 Contoh pengolahan masing-masing <i>window</i> .....	69
Tabel 5.8 Hasil skenario 2 dengan parameter 1:end-1 .....	71
Tabel 5.9 Hasil skenario 2 dengan parameter 1:end-2.....	72
Tabel 5.10 Hasil skenario 2 dengan parameter 2:end-1 .....	73
Tabel 5.11 Hasil skenario 2 dengan parameter 2:end-2.....	74
Tabel 5.12 Hasil skenario 2 dengan parameter 3:end-1 .....	75
Tabel 5.13 Hasil skenario 2 dengan parameter 3:end-2.....	76
Tabel 5.14 Hasil skenario 2 dengan parameter 4:end-1 .....	76
Tabel 5.15 Hasil skenario 2 dengan parameter 4:end-2.....	77
Tabel 5.16 Rekapitulasi skenario 2.....	79
Tabel 5.17 Contoh hasil pengolahan variasi indeks.....	79
Tabel 5.18 Hasil uji coba skenario 3 parameter [0,0; 10] .....	82
Tabel 5.19 Hasil uji coba skenario 3 parameter [0,5; 10] .....	83
Tabel 5.20 Hasil uji coba skenario 3 parameter [0,6; 10] .....	84
Tabel 5.21 Hasil uji coba skenario 3 parameter [1,0; 10] .....	85
Tabel 5.22 Hasil uji coba skenario 3 parameter [0,0; 20] .....	86
Tabel 5.23 Hasil uji coba skenario 3 parameter [5,0; 20] .....	87
Tabel 5.24 Rekapitulasi skenario 3.....	88
Tabel 5.25 Contoh hasil pengolahan variasi frekuensi .....	89

Tabel 5.26 Hasil dengan himpunan latih asli .....	91
Tabel 5.27 Hasil dengan jumlah <i>template</i> 1.....	92
Tabel 5.28 Hasil dengan jumlah <i>template</i> 5.....	93
Tabel 5.29 Hasil dengan jumlah <i>template</i> 15.....	94
Tabel 5.30 Hasil dengan jumlah <i>template</i> 1.....	95
Tabel 5.31 Hasil dengan jumlah <i>template</i> 5.....	95
Tabel 5.32 Hasil dengan jumlah <i>template</i> 15.....	96
Tabel 5.33 Tabel rekapitulasi skenario 4 .....	98
Tabel 5.34 Hasil dengan himpunan latih asli .....	98
Tabel 5.35 Hasil dengan jumlah <i>template</i> 1.....	99
Tabel 5.36 Hasil dengan jumlah <i>template</i> 5.....	100
Tabel 5.37 Hasil dengan jumlah <i>template</i> 15.....	101
Tabel 5.38 Hasil dengan jumlah <i>template</i> 1.....	102
Tabel 5.39 Hasil dengan jumlah <i>template</i> 5.....	103
Tabel 5.40 Hasil dengan jumlah <i>template</i> 15.....	104
Tabel 5.41 Rekapitulasi skenario 5 .....	105
Tabel 5.42 Perbandingan metode <i>filtering/preprocessing</i> ..	107

## DAFTAR GAMBAR

Gambar 2.1 NeuroSky Mindwave.....	8
Gambar 2.2 Sandi morse standar internasional .....	10
Gambar 2.3 Sinyal masukan pada proses EMD .....	12
Gambar 2.4 IMF yang dihasilkan pada proses EMD .....	12
Gambar 2.5 Ilustrasi optimal warping path [8] .....	15
Gambar 2.6 Ilustrasi algoritma kNN [11].....	16
Gambar 2.7 Json string.....	18
Gambar 3.1 Diagram alir model sistem.....	20
Gambar 3.2 Sinyal masukan.....	22
Gambar 3.3 Puncak dan lembah tersegmentasi.....	22
Gambar 3.4 Diagram alir penerimaan data.....	24
Gambar 3.5 Paket data yang dikirimkan ke sitem <i>server</i> .....	25
Gambar 3.6 Format data yang diterima oleh klien .....	25
Gambar 3.7 Diagram alir <i>preprocessing</i> .....	28
Gambar 3.8 Sinyal dalam domain waktu .....	29
Gambar 3.9 Sinyal dalam domain frekuensi .....	29
Gambar 3.10 Ilustrasi bandpass filtering.....	30
Gambar 3.11 Ilustrasi <i>cut-off</i> frequency.....	30
Gambar 3.12 Domain frekuensi yang telah terpotong.....	31
Gambar 3.13 Gambar sinyal luaran <i>preprocessing</i> .....	32
Gambar 3.14 Bagan tahap segmentasi ROI secara umum.....	32
Gambar 3.15 <i>Pseudocode</i> normalisasi sinyal.....	33
Gambar 3.16 <i>Pseudocode</i> pendeteksian puncak .....	34
Gambar 3.17 <i>Pseudocode</i> pendeteksian ROI .....	35
Gambar 3.18 Diagram tahap <i>postprocessing</i> secara umum .....	37
Gambar 3.19 <i>Pseudocode</i> tahap <i>postprocessing</i> .....	38
Gambar 3.20 Data asli (atas), data hasil pemotongan (bawah).....	39

Gambar 3.21 Pemilihan <i>template</i> sinyal menggunakan <i>K-Means Clustering</i> .....	40
Gambar 3.22 Pemilihan <i>template</i> sinyal menggunakan <i>Hierarchical Clustering</i> .....	41
Gambar 3.23 <i>Pseudocode</i> k-Nearest Neighbor .....	42
Gambar 3.24 Diagram alir evaluasi metode .....	43
Gambar 3.25 Tatap muka utama .....	44
Gambar 3.26 Kotak dialog <i>error</i> yang ditampilkan jika tidak ada perangkat terhubung .....	45
Gambar 3.27 Tampilan hasil Analisis sinyal masukan .....	46
Gambar 4.1 Kode program <i>NeuroskyConnector</i> .....	48
Gambar 4.2 Kode program <i>StreamClass</i> .....	49
Gambar 4.3 Kode program modul <i>MonitorClass</i> .....	51
Gambar 4.4 Modul pengenalan sinyal .....	53
Gambar 4.5 Kode program evaluasi skenario .....	56
Gambar 5.1 Sinyal masukan sistem .....	60
Gambar 5.2 Hasil filter <i>window averaging</i> .....	60
Gambar 5.3 Hasil filter <i>empirical mode decomposition</i> .....	61
Gambar 5.4 Hasil filter <i>bandpass</i> .....	61
Gambar 5.5 Hasil pendeteksian ROI oleh filter <i>window averaging</i> .....	62
Gambar 5.6 Hasil pendeteksian ROI oleh filter <i>empirical mode decomposition</i> .....	62
Gambar 5.7 Hasil pendeteksian ROI oleh filter <i>bandpass</i> .....	63

# BAB I

## PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

### 1.1 Latar Belakang

Sebagai makhluk sosial, manusia membutuhkan komunikasi dalam melakukan aktifitas sehari-hari. Komunikasi yang dilakukan dapat menggunakan bermacam-macam metode, baik yang langsung maupun tidak langsung. Komunikasi langsung terjadi jika lebih dari satu objek bertukar informasi dengan bertatap muka secara langsung. Namun, bagi penderita cacat parah, tipe komunikasi seperti ini tentu menimbulkan permasalahan tersendiri. Penderita cacat parah tidak mampu menuliskan kata yang dikehendaki, apalagi untuk berkomunikasi secara lisan karena mungkin mengalami gangguan syaraf yang berhubungan dengan indra yang bersangkutan. Oleh karena itu, diperlukan sebuah alat yang digunakan sebagai pembantu orang-orang dengan keadaan seperti ini untuk berkomunikasi. Salah satunya dengan membangun sebuah sistem otomatis yang memanfaatkan sinyal *Electroencephalogram* (EEG) sebagai media komunikasi [1].

Sinyal EEG yang didapatkan tentu tidak dapat digunakan sebagai media komunikasi secara langsung. Maka dari itu, sinyal yang didapatkan perlu diubah ke dalam bentuk kode, salah satunya adalah sandi Morse. Sandi Morse adalah metode komunikasi yang bisa memfasilitasi penderita cacat parah. Sandi Morse merepresentasikan karakter alfabet dan numerik ke dalam bentuk titik dan garis.

Dalam tugas akhir ini, akan digunakan metode *Bandpass Filtering*, *Window Averaging*, dan *Empirical Mode Decomposition*

(EMD) untuk mengisolasi sinyal *Electrooculogram* (EOG) dari sinyal EEG [2]. Sinyal EOG adalah sinyal yang dihasilkan dari pergerakan otot mata subjek uji. Pergerakan kelopak mata saat menutup dan membuka inilah yang akan menjadi dasar penentuan sandi Morse yang terbaca. Setelah pola sandi Morse diketahui, karakter yang dimaksud dapat terbaca berdasarkan pola sandi Morse.

## **1.2 Rumusan Masalah**

Berikut beberapa hal yang menjadi rumusan masalah dalam tugas akhir ini :

1. Bagaimana cara mendapatkan sinyal EEG dari perangkat *Neurosky Mindwave* dan bagaimana spesifikasi sinyal yang dihasilkan?
2. Bagaimana proses ekstraksi sinyal EOG dari sinyal EEG yang dibangkitkan dari perangkat *Neurosky Mindwave*?
3. Bagaimana proses penerjemahan sinyal EOG menjadi sandi Morse?

## **1.3 Batasan Masalah**

Berikut beberapa hal yang menjadi batasan masalah dalam pengerjaan tugas akhir ini:

1. Pengambilan sinyal EEG menggunakan perangkat *Neurosky Mindwave* satu elektroda yang diletakkan pada kening.
2. Data yang diambil adalah data kedipan mata subjek uji dan berapa pada kondisi tenang dan tidak terdapat banyak gerakan yang mengganggu.
3. Implementasi program yang dilakukan pada lingkungan komputer desktop.
4. Hasil akhir dari pengolahan data yang didapatkan adalah karakter tunggal dari sandi Morse yang bersangkutan.

## 1.4 Tujuan

Tujuan dari tugas akhir ini adalah untuk membangun aplikasi pembaca sandi Morse dari sinyal EEG yang didapatkan dari perangkat *Neurosky Mindwave* melalui kedipan mata pengguna.

## 1.5 Metodologi

Tahap yang dilakukan untuk menyelesaikan tugas akhir ini adalah sebagai berikut:

### 1. Penyusunan Proposal Tugas Akhir

Penulisan proposal ini merupakan tahap awal dalam pengerjaan tugas akhir. Pada proposal ini, penulis mengajukan gagasan pengenalan sandi Morse dari sinyal EEG menggunakan metode *empirical mode decomposition* (EMD), *bandpass* dan *window averaging*.

### 2. Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur untuk pengumpulan data dan desain aplikasi yang akan dibuat. Informasi didapatkan dari buku dan literatur lain yang berhubungan dengan algoritma yang digunakan dalam pengerjaan tugas akhir ini yaitu studi literatur mengenai metode *preprocessing* sinyal dan ekstraksi sinyal EOG.

### 3. Implementasi

Pada tahap ini dibangun perangkat lunak sesuai dengan rancangan yang diajukan pada proposal. Pembangunan perangkat lunak diimplementasikan sesuai dengan konsep yang telah didapatkan saat studi literatur.

### 4. Pengujian dan Evaluasi

Tahap ini merupakan tahap uji coba aplikasi dan evaluasi data keluaran. Tahap uji coba dilakukan dengan

membandingkan metode *preprocessing* serta membandingkan deteksi karakter dengan pengenalan berbasis aturan dan *learning based recognition*. Tahap evaluasi dilakukan dengan menghitung tingkat kebenaran pengenalan (akurasi) dan *confusion matrix* dari masing-masing kelas alfabet.

## **5. Penyusunan Buku Tugas Akhir**

Tahap ini merupakan tahap dokumentasi dari tugas akhir. Buku tugas akhir berisi dasar teori, perancangan, implementasi, serta hasil uji coba dan evaluasi dari aplikasi yang dibangun.

### **1.6 Sistematika Penulisan**

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

#### **1. Bab I. Pendahuluan**

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

#### **2. Bab II. Tinjauan Pustaka**

Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan tugas akhir.

#### **3. Bab III. Analisis dan Perancangan**

Bab analisis dan perancangan berisi penjelasan mengenai analisis kebutuhan, perancangan sistem dan perangkat yang digunakan dalam pengerjaan tugas akhir serta urutan pelaksanaan proses.

#### **4. Bab IV. Implementasi**

Bab implementasi berisi pembangunan aplikasi pengenalan sandi morse dari sinyal EEG menggunakan bahasa *Python*.

5. Bab V. Uji Coba dan Evaluasi

Bab ini berisi hasil evaluasi aplikasi dengan menggunakan aplikasi yang dibangun. Juga disertakan analisis dari hasil evaluasi perangkat lunak.

6. Bab VI. Kesimpulan dan Saran

Bab kesimpulan dan saran berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan tugas akhir.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab tinjauan pustaka berisi penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Sinyal *Electroencephalogram* (EEG)**

Sinyal elctroencephalogram (EEG) adalah sinyal aktivitas listrik dilapisan terluar kulit otak (*cerebral cortex*). Umumnya sinyal EEG digunakan untuk mendiagnosa penyakit yang berkaitan dengan otak dan kejiwaan, seperti epilepsi, tumor otak, mendeteksi posisi/lokasi otak yang terluka, serta membantu diagnosa gangguan mental. Pengukuran sinyal EEG dilakukan dengan cara meletakkan elektroda-elektroda pada kulit kepala (*scalp*) dan hasil pengukurannya dipengaruhi oleh beberapa variabel, seperti kondisi mental, aktivitas pada saat pengukuran, kondisi kesehatan, dan kondisi lingkungan pengukuran, serta faktor stimulus eksternal yang diterima pada saat pengukuran. Karena karakteristik dari sinyal EEG yang tidak periodik, tidak mempunyai pola baku, serta mempunyai amplitudo tegangan yang kecil, maka menyebabkan sinyal EEG sangat mudah tertimbun *noise* sehingga sulit untuk didiagnosa kecuali oleh dokter/ahli terlatih.

Perekaman sinyal EEG, selain dipengaruhi oleh kondisi mental, aktifitas pasien, dan kondisi lingkungan disekitar pengukuran, juga dipengaruhi oleh pemberian stimulus dari eksternal saat pengukuran, seperti suasana relaks namun sadar, suasana berpikir/beraktifitas dengan mata terbuka, suasana tidur ringan dengan memberi tekanan berupa cahaya lampu yang dihidup matikan, serta suasana tidur nyenyak [3]. Untuk menganalisa adanya penyakit di otak, maka dilakukan pengukuran sinyal EEG berdasarkan kondisi pikiran, frekuensi dan amplitudo

tegangan. Berdasarkan frekwensi, amplitudo tegangan, dan kondisi obyek, sinyal EEG dapat dibagi menjadi 4 gelombang, yaitu gelombang alpha, beta, theta, dan delta.

## 2.2 *NeuroSky Mindwave*

*Neurosky Mindwave* merupakan perangkat yang dipakai untuk mengambil data pada pengerjaan tugas akhir ini. Perangkat ini terhubung pada komputer secara nirkabel dengan baterai ukuran AA sebagai sumber daya. Jumlah elektroda yang terpasang pada alat ini adalah 2 buah, satu elektroda dipasang pada posisi kening bagian kiri (*fp1*) dan satu elektroda referensi berbentuk penjepit yang dipasang pada telinga kiri. Penggunaan perangkat ini tidak memerlukan larutan *saline* karena bertipe elektroda kering. *Sampling rate* yang dihasilkan berfrekuensi 512 Hz. Perangkat *Neurosky Mindwave* ditunjukkan pada gambar 2.1.



**Gambar 2.1 NeuroSky Mindwave**

Nilai yang dihasilkan oleh perangkat ini adalah *rawEeg*, *theta*, *lowAlpha*, *highAlpa*, *lowBeta*, *highBeta*, *lowGamma*, *highGamma* dan *blinkStrength*. Selain *rawEeg*, semua nilai ditransmisikan setiap 1 detik, sedangkan nilai *rawEeg* yang dikembalikan setiap detiknya tidak lebih dari 512. Selain itu, nilai

*meditation* dan *attention* yang diperoleh dari perhitungan algoritma pribadi pengembang perangkat juga dikembalikan per detiknya. Penggunaan nilai-nilai yang disediakan akan disesuaikan dengan tujuan pemanfaatan perangkat. Dari nilai-nilai yang dikembalikan, hanya nilai *rawEeg* yang digunakan pada proses pengolahan.

### 2.3 Sandi Morse

Sandi Morse adalah sistem representasi huruf, angka, tanda baca dan sinyal dengan menggunakan kode titik dan garis yang disusun mewakili karakter tertentu pada alfabet dan angka. Sandi Morse diciptakan oleh Samuel F.B. Morse dan Alfred Vail pada tahun 1835 untuk mengirim pesan melalui kabel telegraf. Kegunaan utama dari sistem sandi ini adalah untuk komunikasi jarak jauh atau bila komunikasi secara langsung tidak memungkinkan untuk dipakai.

Sandi Morse biasa digunakan dalam dunia kepramukaan, kepanduan, keadaan darurat dan *Save Our Soul* (SOS). Dalam bidang militer, sandi morse digunakan untuk saling bertukar pesan. Dalam bidang pendidikan kepelayaran, sandi morse digunakan untuk bahan ujian sertifikasi bagi pelamar operator kapal dan operator stasiun pantai yang telah lulus ujian tertulis. Sandi morse saat ini paling populer di kalangan operator radio amatir, karena Sandi morse digunakan sebagai pola untuk tombol transmiter dan di radio komunikasi biasa disebut sebagai gelombang tetap atau *continuous wave*. Sandi morse juga digunakan sebagai teknologi bantuan, yaitu membantu orang yang cacat untuk berkomunikasi, misalnya orang yang kehilangan penglihatan atau orang yang cacat dalam pendengaran [4].

Terdapat beberapa macam tipe peralatan untuk membangkitkan sandi morse, sistem satu knop, dua knop, dan tiga knop. Perbedaan antara sistem ini adalah jumlah tipe elemen yang bisa dimasukkan. Pada sistem satu knop, hanya terdapat dua kondisi, saklar mati dan hidup. Perbandingan garis dan titik dalam

sistem ini adalah 3:1. Untuk sistem dua knop, terdapat saklar titik dan garis. Sedangkan untuk sistem tiga knop terdapat saklar titik, garis, dan spasi. Sistem yang cocok dipakai bagi penderita cacat parah adalah sistem satu knop karena kemudahan yang diberikan dalam mengirimkan sinyal. Gambar 2.2 menunjukkan daftar sandi morse.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	• — • —
D	— • • •	X	• • • — •
E	•	Y	• • — • —
F	• • — •	Z	— • — • •
G	— • • •		
H	• • • •		
I	• • •		
J	• — • — • —	1	• — • — • — • —
K	— • • —	2	• • — • — • —
L	• — • — •	3	• • • — • — • —
M	— • —	4	• • • • — • —
N	— • •	5	• • • • •
O	— • — • —	6	• • • • • •
P	• — • — • •	7	• • • • • • •
Q	— • — • • •	8	• — • — • • • •
R	• • • •	9	• — • — • — • •
S	• • •	0	— • — • — • —
T	— •		

**Gambar 2.2 Sandi morse standar internasional**

#### **2.4 Empirical Mode Decomposition (EMD)**

*Empirical Mode Decomposition* (EMD) adalah metode pendekomposisian sinyal dalam domain waktu. EMD seperti layaknya metode analisis sinyal yang lain seperti *Fourier Transform* dan *Wavelet Decomposition*. Proses yang dijalankan berfungsi untuk menganalisis sinyal asli baik sinyal non-linear dan non-stasioner [5].

EMD akan mendekomposisikan sinyal menjadi beberapa komponen IMF (*Intrinsic Mode Function*). Sebuah IMF adalah fungsi yang hanya memiliki satu titik ekstrik antara *zero-crossing* (perpindahan nilai positif ke negatif atau sebaliknya) dan memiliki nilai rata-rata nol. Berikut ini adalah proses kerja dari EMD (*shifting process*) [6].

1. Untuk sinyal  $X(t)$ ,  $m_1$  adalah rata-rata dari puncak atas dan bawah.
2. Komponen pertama

$$h_1 = X(t) - m_1 \quad (2.1)$$

3. Pada tahap *shifting process* yang kedua,  $h_1$  dijadikan sebagai data masukan dan  $m_{11}$  adalah rata-rata dari puncak atas dan bawah  $h_1$ , sehingga

$$h_{11} = h_1 - m_{11} \quad (2.2)$$

4. Shifting process dilakukan berulang-ulang sebanyak  $k$  kali, sampai  $h_{1k}$  adalah sebuah IMF dimana

$$h_{1(k-1)} - m_{1k} = h_{1k} \quad (2.3)$$

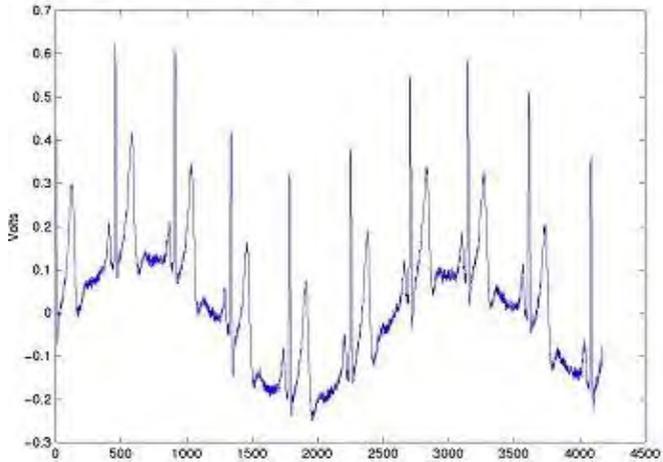
5. Nyatakan bahwa  $c_1 = h_{1k}$ , komponen IMF pertama dari data, yang mengandung periode komponen terpendek dari sinyal yang diproses. Pisahkan dari sisa data pada persamaan 2.4,

$$X(t) - c_1 = r_1 \quad (2.4)$$

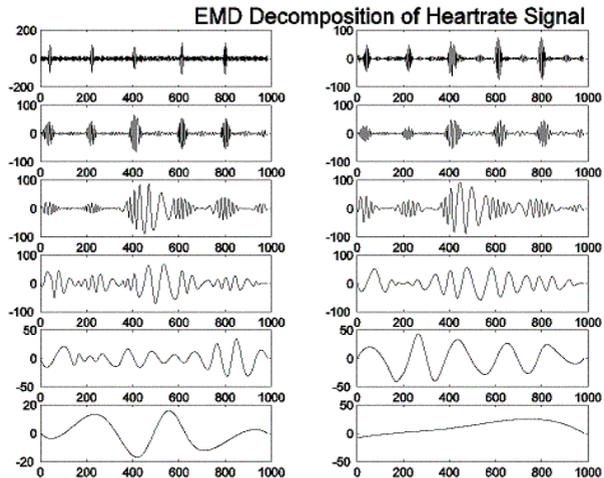
langkah ini diulangi berderet seperti pada persamaan 2.5

$$r_1 - c_2 = r_2, \dots, r_{n-1} - c_n = r_n \quad (2.5)$$

Gambar 2.3 dan 2.4 masing-masing menunjukkan sinyal masukan dan *shifting process* pada EMD.



**Gambar 2.3 Sinyal masukan pada proses EMD**



**Gambar 2.4 IMF yang dihasilkan pada proses EMD**

## 2.5 Transformasi *Fourier*

Transformasi *Fourier* dikenal sebagai alat yang handal untuk menganalisis sinyal. Transformasi *Fourier* mengubah sinyal dari domain waktu menjadi domain frekuensi. Transformasi *fourier* untuk sebuah sinyal waktu kontinu  $x(t)$  secara matematis ditunjukkan oleh persamaan 2.6.

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt, \quad (2.6)$$

dimana  $\omega \in (-\infty, \infty)$

Sementara *Discrete Fourier Transform* dibentuk dengan menggantikan integral berhingga dengan sederetan jumlahan pada suatu nilai berhingga. Persamaan DFT ditunjukkan pada persamaan 2.7.

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t_n)e^{-j\omega_k t_n}, \quad (2.7)$$

dengan  $k = 0, 1, N-1$

$X(t_n)$  selanjutnya akan disebut sebagai  $x(n)$  merupakan notasi sampel ke- $n$  pada sinyal masukan.  $X(\omega_k)$  juga dapat dijumpai sebagai  $X(k)$  merupakan sampel spektrum ke- $k$ .

Tabel 2.1 adalah tabel yang menunjukkan paramter-parameter pada persamaan Transformasi *Fourier*.

**Tabel 2.1 Tabel parameter persamaan Transformasi *Fourier***

Simbol Parameter	Nilai/Maksud
$j$	$\sqrt{-1}$
$e$	$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = 2,71828182..$
$\omega_k = k\theta$	Sampel frekuensi ke- $k$ . $\theta$ merupakan interval <i>sampling</i>

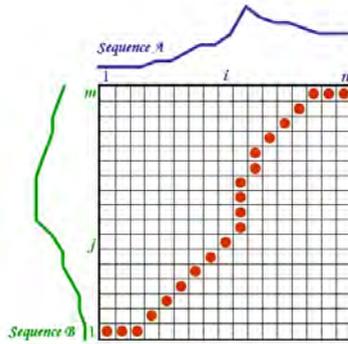
$N$	Sampel frekuensi yang digunakan (Hz)
-----	--------------------------------------

Dengan melihat Persamaan (2.7) diketahui bahwa DFT memiliki basis sinyal sinusoida dan merupakan bentuk kompleks. Sehingga representasi domain frekuensi yang dihasilkan juga akan memiliki bentuk kompleks [7].

## 2.6 Dynamic Time Warping (DTW)

Metode *dynamic time warping* (DTW) pertama kali diperkenalkan pada tahun 1959 oleh R. Bellman. Saat ini, metode DTW sudah diterapkan dalam berbagai bidang, diantaranya *handwriting* dan *online signature matching*, *computer vision* dan *computer animation*, *protein sequence alignment* dan *chemical engineering*, dan lain-lain.

DTW adalah metode untuk menghitung kesamaan antara dua deret waktu (*time series*) yang berbeda karena waktu dan kecepatan. Contohnya dalam pendeteksian kesamaan pola gerak berjalan. Kecepatan gerak seseorang mungkin berbeda, namun pola gerak dan naik turun lembahnya sama. Cara kerja DTW adalah dengan membandingkan *optimal warping path*. Dimana *optimal warping path* adalah jalur dengan cost terpendek dari selisih perbandingan sinyal-sinyal masukan [8]. Gambar 2.5 menunjukkan gambaran *optimal warping path* yang dihasilkan dari algoritma DTW.



**Gambar 2.5 Ilustrasi optimal warping path [8]**

Persamaan 2.8 adalah persamaan fungsi minimalisasi *warping path* dari 2 sinyal *time series*.

$$DTW[i, j] = d(s[i], t[j]) + \min(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1]) \quad (2.8)$$

Sinyal  $s$  dan  $t$  memiliki panjang masing-masing  $n$ ,  $m$ .  $DTW$  adalah variabel matriks yang berukuran  $n \times m$ . Nilai *path* minimum didapatkan dari penjumlahan antara selisih  $s_i$  dan  $t_j$  dengan nilai minimum pada perhitungan tiga arah sebelumnya (atas, samping, pojok kiri atas).

## 2.7 K-Nearest Neighbor (kNN) Classifier

Pengklasifikasi kNN adalah pengklasifikasi yang tidak membangkitkan model atau bisa disebut sebagai *lazy classifier*. Model pada pengklasifikasi ini adalah fitur-fitur dari data latih (*training*). Penentuan kelas suatu *instance* didapatkan dengan melakukan *voting* terhadap  $k$  data latih dengan jarak terdekat. Penghitungan jarak yang sering dipakai adalah *Euclidean distance* [10]. Karena penghitungan jarak yang dipakai berdasarkan selisih kedekatan fitur, maka data uji dan latih harus mempunyai dimensi fitur yang sama.

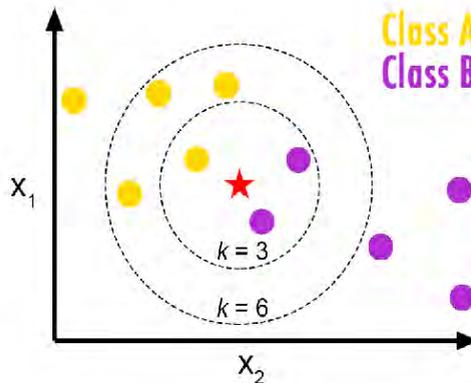
Terdapat dua langkah utama dalam algoritma pengklasifikasi kNN, yaitu

1. Penghitungan jarak antara *instance p* dan *instance q* yang berada di himpunan data latih dengan jumlah fitur  $f$ . Persamaan penghitungan jarak ditunjukkan oleh persamaan 2.9.

$$d(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^f (p_j - q_j)^2 \quad (2.9)$$

Variabel  $d$  adalah jarak antara *instance p* dan  $q$ .

2. Pemilihan kelas data uji berdasarkan *voting* pada  $k$  tetangga terdekat. Untuk semua nilai  $d$  yang didapatkan, diurutkan dari kecil ke besar. Kelas hasil prediksi adalah kelas dengan nilai *class majority* terbesar pada himpunan  $k$  jarak terdekat. Ilustrasi algoritma kNN ditunjukkan pada gambar 2.6.



**Gambar 2.6 Ilustrasi algoritma kNN [11]**

Pada Gambar 2.6 telah ditunjukkan ilustrasi cara kerja kNN. Jika nilai  $k = 3$ , maka kelas dari data merah adalah B, sedangkan

jika nilai  $k = 6$  maka kelasnya A. Pada tugas akhir ini, algoritma kNN digunakan sebagai metode pengenalan berbasis pembelajaran yang dikombinasikan dengan DTW.

## 2.8 *K-Means Clustering*

*K-Means* adalah salah satu metode pembelajaran tidak terbantu yang paling sederhana dan mampu memecahkan berbagai permasalahan pengelompokan (*clustering*). Cara kerja algoritma ini tergolong sederhana untuk mengelompokkan himpunan data ke dalam  $k$  kluster. Ide utama dari algoritma ini adalah menentukan  $k$  *centroid* (pusat kluster) untuk setiap kluster pada tahap inisialisasi. *Centroid-centroid* tersebut harus diletakkan pada titik-titik yang tepat karena titik yang berbeda akan mengakibatkan hasil yang berbeda. Langkah selanjutnya adalah dengan meletakkan setiap titik (data) pada *centroid* terdekat. Setelah semua titik telah diletakkan pada kluster-kluster yang ada berdasarkan pada kedekatan *centroid*, (penghitungan kedekatan bisa memakai persamaan 2.9) *centroid* baru dihitung berdasarkan titik-titik yang ada pada masing-masing kluster. Langkah ini diulangi sampai kriteria pemberhentian (*stopping criterion*) terpenuhi [12]. Beberapa kriteria pemberhentian yang dipakai adalah jumlah iterasi maksimal dan konvergensi nilai SSE (*Sum of Square Error*). Nilai SSE dihitung dengan menggunakan persamaan 2.10.

$$SSE_x = \sum_{i=1}^N \sum_{j=1}^D (p_j - c_j)^2 \quad (2.10)$$

Persamaan 2.10 menghitung jarak masing-masing data  $p$  di kluster  $x$  dengan *centroid*  $c$  dan mengembalikan nilai SSE untuk kluster  $x$  dengan jumlah anggota  $N$  dan dimensi fitur  $D$ .

## 2.9 *JSON String*

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis bagi

manusia. JSON adalah format teks yang secara penuh bebas terhadap bahasa pemrograman [13]. Json *string* akan digunakan sebagai media komunikasi program *thinkgear* dengan aplikasi yang akan dibangun. Gambar 2.7 mencontohkan format json *string*.

```
"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter","lastName":"Jones"}
]
```

**Gambar 2.7** Json string

## **BAB III**

### **DESAIN DAN PERANCANGAN**

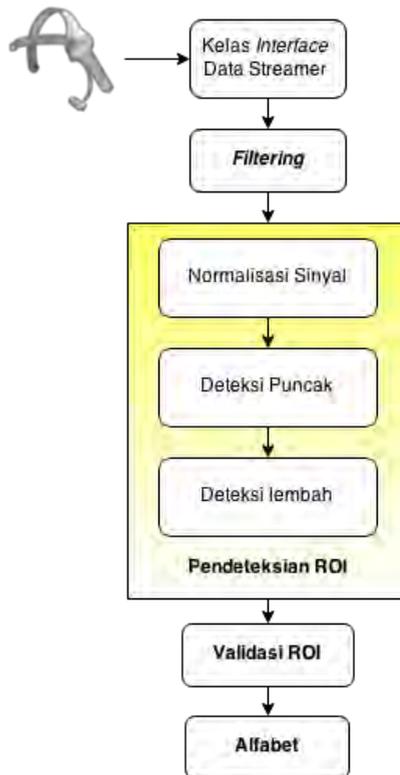
Pada Bab 3 akan dijelaskan mengenai perancangan sistem perangkat lunak untuk mencapai tujuan dari tugas akhir. Perancangan yang akan dijelaskan pada bab ini meliputi perancangan data, perancangan proses dan perancangan antar muka. Selain itu akan dijelaskan juga desain metode secara umum pada sistem.

#### **3.1 Desain Metode Secara Umum**

Pada tugas akhir ini akan dibangun suatu sistem untuk menerjemahkan sandi morse dari sinyal EEG. Terdapat dua macam metode yang digunakan sebagai pengenalan sandi morse pada tugas akhir ini. Yang pertama adalah pengenalan berdasarkan aturan dan pengenalan berdasarkan pembelajaran (*learning based detection*). Proses umum yang ada dalam sistem ini adalah pengambilan data alir secara waktu nyata dari perangkat *Neurosky Mindwave*. Proses khusus yang dijalankan pada pendeteksian berdasarkan aturan yaitu *preprocessing* atau filtering sinyal masukan, pendeteksian *Region of Interest (ROI)* dan pengenalan karakter dari sinyal masukan. Sedangkan pada pendeteksian berdasarkan pembelajaran digunakan pengklasifikasi *k-Nearest Neighbor* dengan nilai  $k = 1$  dan menggunakan pengukuran jarak *Dynamic Time Warping*.

Pengambilan data alir dari perangkat *Neurosky* adalah tahapan yang harus dilakukan untuk mendapatkan data sinyal masukan. Pada pendeteksian berdasarkan aturan, data sinyal yang didapatkan tidak dapat diolah secara langsung untuk mendapatkan pola sandi morse terdaftar. Data sinyal yang didapatkan akan menjadi data masukan pada tahap kedua, yaitu tahap *preprocessing*. Pada tahap ini, sinyal EEG akan diekstraksi dan menghasilkan sinyal EOG (Electrooculogram). Ekstraksi sinyal EOG dari EEG menggunakan *bandpass cut-off filtering*. Sinyal EOG yang didapatkan akan mengalami pemrosesan lanjut pada

tahap pendeteksian ROI. Pada tahap ini titik dan garis dari sinyal EOG akan dikenali secara otomatis. Pengenalan awal titik dan garis diawali dengan pendeteksian puncak. Bagian-bagian yang tersegmentasi akan mengalami validasi titik garis oleh aturan yang telah ditetapkan. Susunan titik garis yang telah divalidasi akan diterjemahkan ke dalam karakter yang terdapat pada tabel sandi morse. Bagan dari sistem yang dibangun ditunjukkan oleh Gambar 3.1.



**Gambar 3.1 Diagram alir model sistem**

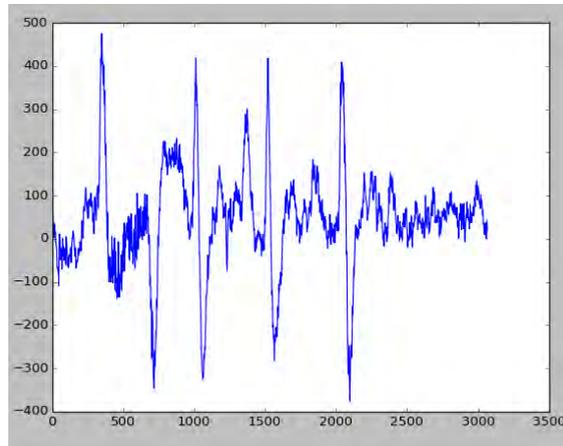
## 3.2 Perancangan Data

Perancangan data merupakan bagian yang terpenting dalam pengoperasian perangkat lunak karena diperlukan data yang tepat agar perangkat lunak dapat beroperasi dengan benar. Pada bagian ini akan dijelaskan mengenai perancangan data yang dibutuhkan untuk membangun perangkat lunak pengenalan sandi morse dari sinyal EEG. Data yang diperlukan dalam pengoperasian perangkat lunak adalah data masukan (*input*) dan data keluaran (*output*) yang memberikan hasil proses pengoperasian perangkat lunak untuk pengguna.

### 3.2.1 Data Masukan

Data masukan adalah data awal yang akan diproses pada sistem pengenalan sandi morse dari sinyal EEG. Data yang diproses berupa data sinyal yang direpresentasikan ke dalam baris angka desimal (*array of float number*) yang sebelumnya telah didapatkan dari proses pengambilan data. Frekuensi sinyal yang dihasilkan sensor perangkat tidak lebih dari 512 Hz dengan lama waktu pengambilan data berbeda-beda tiap sinyal.

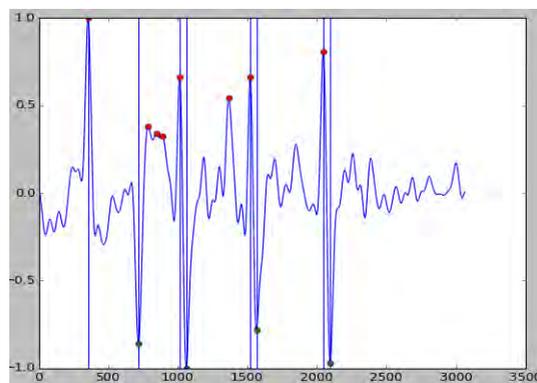
Data sinyal masukan memiliki beberapa puncak dan lembah yang mempunyai amplitudo yang tinggi. Puncak dan lembah ini dihasilkan oleh gerakan otot mata. Puncak yang tinggi menunjukkan pergerakan kelopak mata dari buka menjadi tutup. Sedangkan lembah yang tinggi menunjukkan pergerakan kelopak mata dari tutup menjadi buka. Pada pengerjaan tugas akhir ini, puncak dan lembah yang dihasilkan dari pergerakan kelopak mata inilah yang akan dideteksi sebagai ROI. Gambar 3.2 menunjukkan sinyal yang berhasil direkam.



**Gambar 3.2 Sinyal masukan**

### 3.2.2 Data Luaran

Data luaran dari sistem ini merupakan data hasil proses pengenalan sandi morse dari sinyal EEG. Data luaran yang didapatkan adalah hasil evaluasi akurasi berbagai metode pengenalan yang terdiri dari pengenalan berbasis aturan dan *learning based*, tampilan bagian-bagian sinyal yang tersegmentasi oleh sistem, *confusion matrix*, dan kelas sinyal data masukan.



**Gambar 3.3 Puncak dan lembah tersegmentasi**

Gambar 3.3 menunjukkan tampilan bagian-bagian puncak dan lembah yang tersegmentasi oleh sistem. Masing-masing ROI dibatas oleh dua garis biru. Dari gambar 3.3 diketahui bahwa terdapat empat ROI yang akan divalidasi.

### **3.3 Perancangan Proses**

Berikut ini adalah rancangan dari sistem pengenalan sandi morse dari sinyal EEG:

#### **3.3.1 Tahap Pengambilan Data**

Data diambil sebagai uji coba terhadap sistem yang dibangun. Pengguna perangkat melakukan aksi kedip panjang dan kedip singkat yang merepresentasikan sandi morse yang ingin diamati. Kedip singkat didefinisikan sebagai perubahan keadaan kelopak mata dari kondisi terbuka, tertutup dan terbuka lagi dalam selang waktu yang singkat (tidak lebih dari 1 detik). Sedangkan kedip panjang didefinisikan sebagai perubahan kondisi mata terbuka, tertutup dan terbuka lagi dengan lama waktu kelopak mata tertutup lebih dari 1 detik. Titik dan garis pada sandi morse diwakili dengan kedip singkat dan kedip panjang secara berurutan.

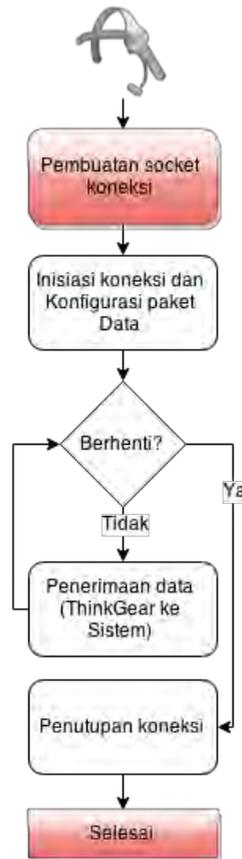
##### **3.3.1.1 Objek Pengambilan Data**

Data sinyal EEG diperoleh dari 3 orang yang berbeda. Setiap pengguna melakukan kedip singkat dan panjang sesuai dengan pola sandi morse yang akan diuji.

##### **3.3.1.2 Perancangan Pengambilan Data**

Pada bagian ini akan dijelaskan bagaimana proses pengambilan data yang dihasilkan perangkat dan penyimpanan data pada lingkungan implementasi sistem. Pengambilan data memanfaatkan program *ThinkGear* bawaan yang disertakan pada *Compact Disc* (CD) instalasi perangkat. Program bawaan ini akan bertindak sebagai penyedia data (*middleware*) bagi pengembang yang ingin memanfaatkan data yang dihasilkan perangkat [13].

*Host* dimana program *ThinkGear* berjalan selanjutnya akan disebut penyedia (*server*). Sistem yang membutuhkan data dari penyedia selanjutnya disebut klien. Terdapat beberapa langkah yang harus dijalankan ketika proses pengambilan data. Proses tersebut adalah pembuatan saluran koneksi penyedia dan klien, inisiasi hubungan antara *server*-klien dan pengaturan paket data, penerimaan data, dan penutupan saluran koneksi. Gambar 3.4 menunjukkan diagram alir sistem penerimaan data.



**Gambar 3.4 Diagram alir penerimaan data**

Ketika tahap inisiasi koneksi, klien mengirimkan data ke penyedia dengan format json. *Field* data yang dikirim adalah *appName* dan *appKey*. *AppName* adalah *field* yang berisi nama aplikasi yang bertindak sebagai klien sedangkan *appKey* adalah identitas aplikasi yang tercatat di penyedia yang berisi string dengan panjang 40 karakter. *Server* ThinkGear mengirimkan data dalam 2 format, yaitu format binary dan json *string*. Berikut ini adalah data yang dikirimkan klien ketika proses inisiasi koneksi dan konfigurasi paket data dalam bentuk json *string*. Gambar 3.5 dan 3.6 adalah paket data dalam format json yang dikirimkan ke *server* dan respon *server* terhadap aplikasi.

```
{
  "appName": "NeuroMorse",
  "appKey": "9f54141b4b4c567c558d3a76cb8d715cbde03096"
} {
  "enableRawOutput": true,
  "format": "Json"
}
```

**Gambar 3.5** Paket data yang dikirimkan ke sitem *server*

```
{
  "poorSignalLevel": 0,
  "eSense": {
    "attention": 38,
    "meditation": 43
  },
  "eegPower": {
    "delta": 1.15e-4,
    "theta": 1.41e-6,
    "lowAlpha": 1.35e-4,
    "highAlpha": 6.69e-5,
    "lowBeta": 1.47e-5,
    "highBeta": 6.95e-,
    "lowGamma": 5.26e-7,
    "highGamma": 1.40e-5
  }
} {
  "rawEeg": 238
} {
  "rawEeg": 282
} {
```

**Gambar 3.6** Format data yang diterima oleh klien

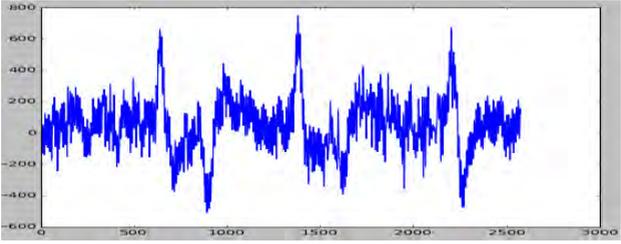
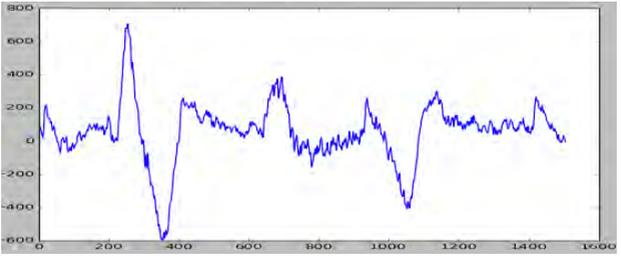
Setelah proses inisiasi koneksi dan konfigurasi paket data berhasil, penyedia akan mengirimkan data hasil pengukuran perangkat ke klien secara terus menerus. Gambar 3.6 menunjukkan data yang dikirimkan oleh *server* ke klien dalam format json *string*.

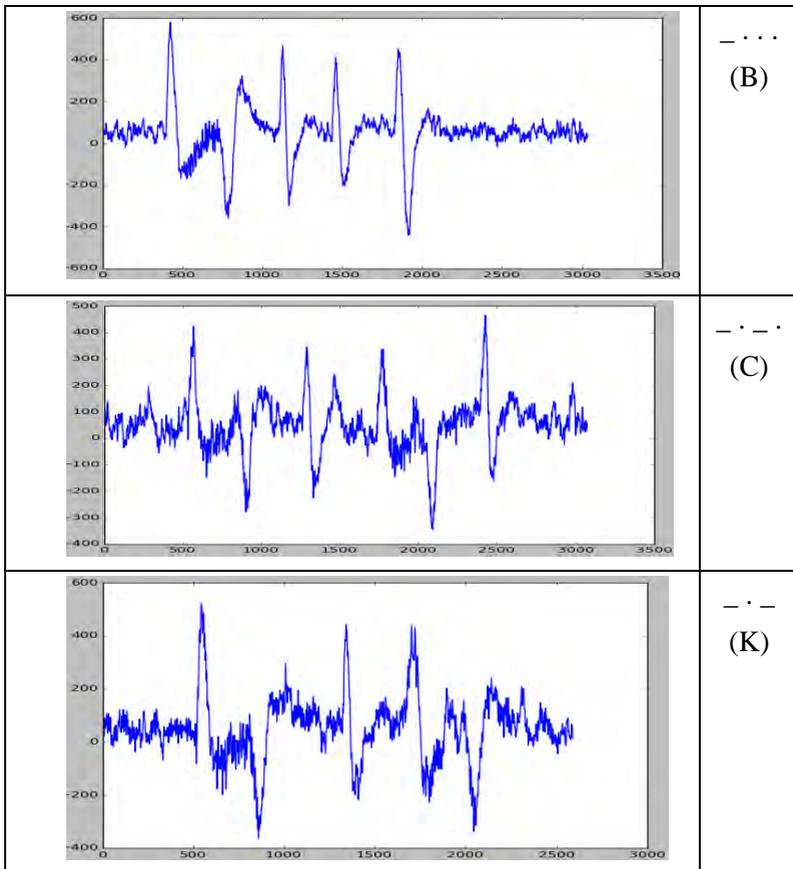
Dari data yang dikirimkan oleh penyedia, hanya *field rawEeg* yang dikirimkan sebanyak tidak lebih dari 512 kali setiap detik. Sedangkan untuk *field-field* yang lain dikirimkan sebanyak 1 kali setiap detik. Nilai dari *field rawEeg* inilah yang akan disimpan dan diolah pada proses selanjutnya. Nilai-nilai yang telah didapat pada *field rawEeg* disimpan ke dalam berkas tanpa ekstensi dengan tanda pemisah spasi untuk setiap nilai.

### 3.3.2 Contoh Data

Tabel 3.1 menunjukkan beberapa data sinyal yang berhasil diambil beserta pola sandi morse sebenarnya.

**Tabel 3.1** Contoh data terekam dan pola sebenarnya

Gambar	Pola
	<p>--- (G)</p>
	<p>· - (A)</p>



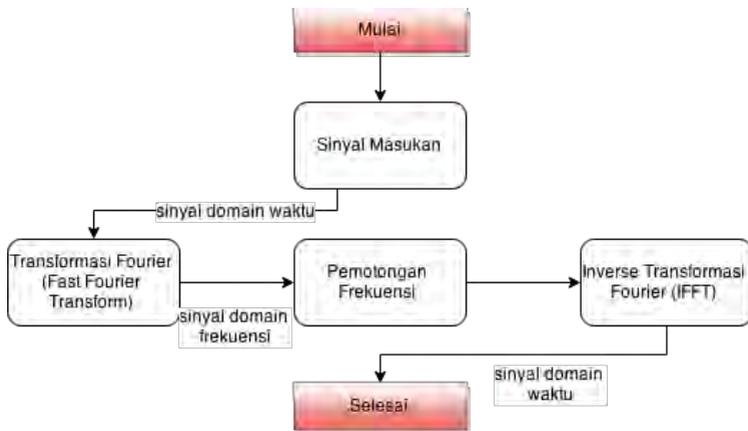
### 3.3.3 Tahap Khusus Pengenalan Berdasarkan Aturan

Pada bagian ini, akan dijelaskan tahap-tahap yang dilakukan pada metode pendeteksian berdasarkan aturan.

#### 3.3.3.1 Tahap *Preprocessing*

Pada tahap ini sinyal EEG yang telah didapatkan akan diekstrak menjadi sinyal EOG. Karena sinyal EEG yang didapatkan berasal dari satu elektroda yang ditempatkan pada

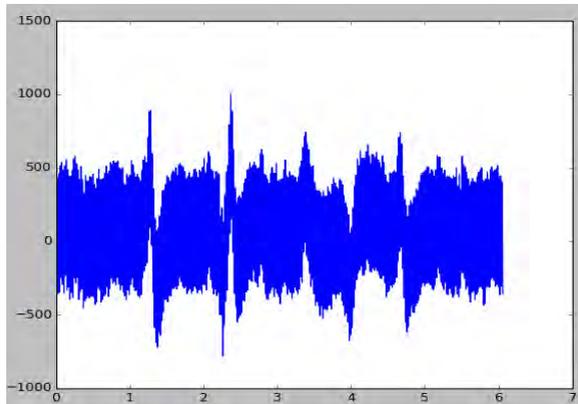
posisi fp1 (kening kiri), pergerakan dari otot mata sangat mendominasi sinyal EEG yang didapatkan. Pola dari sinyal EEG yang didapatkan ini adalah sinyal EOG. Metode ekstraksi yang umum digunakan adalah penghilangan *noise*. Beberapa diantaranya adalah *averaging*, *low pass filtering*, *bandpass filtering*, *frequency cut-off*, *empirical mode decomposition*, dan lain-lain. Pada pengerjaan tugas akhir ini, digunakan metode *frequency-cutoff* pada langkah penghapusan *noise*. Gambar 3.7 menunjukkan diagram alir yang menunjukkan keseluruhan tahap *preprocessing*.



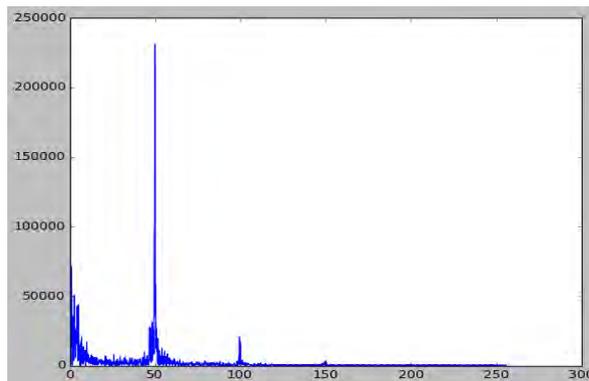
**Gambar 3.7** Diagram alir *preprocessing*

### 3.3.3.1.1 Transformasi Fourier

Sebelum proses filterisasi, sinyal masukan diubah kedalam domain frekuensi. Pengubahan sinyal dari domain waktu menjadi domain frekuensi menggunakan metode *Discrete Fourier Transform*. Gambar 3.8 dan 3.9 masing-masing menunjukkan representasi sinyal dalam domain waktu dan frekuensi.



**Gambar 3.8 Sinyal dalam domain waktu**

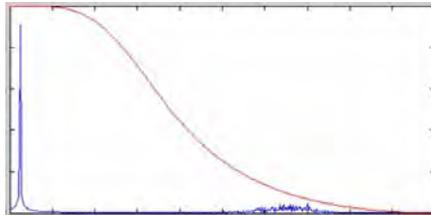


**Gambar 3.9 Sinyal dalam domain frekuensi**

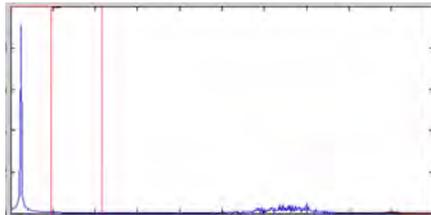
Implementasi metode *Discrete Fourier Transform* menggunakan *Fast Fourier Transform*. Fungsi *Fast Fourier Transform* yang digunakan pada pengerjaan tugas akhir ini didapat dari pustaka *scipy.fftpack* pada fungsi *fft*. Luaran dari fungsi ini adalah jajaran bilangan imajiner sejumlah jajaran sinyal masukan. Luaran yang dihasilkan bersisi ganda, sehingga harus dipotong pada posisi tengah.

### 3.3.3.1.2 *Frequency Cut-Off*

*Frequency cut-off* memiliki konsep umum yang hampir sama dengan *bandpass filtering*. Perbedaannya terletak pada bagaimana pemotongan frekuensi terjadi. Pada metode *frequency cut-off*, frekuensi langsung dipotong pada suatu titik sedangkan pada *bandpass filtering*, nilai spektrum frekuensi dikalikan dengan parameter distribusi beta. *Frequency cut-off* memiliki kelebihan lebih cepat dibanding dengan *bandpass filtering* dikarenakan tidak memerlukan parameter distribusi beta sebagai faktor pengali. Gambar 3.10 dan 3.11 masing-masing menunjukkan ilustrasi *bandpass* dan *cut-off frequency*.



**Gambar 3.10 Ilustrasi bandpass filtering**

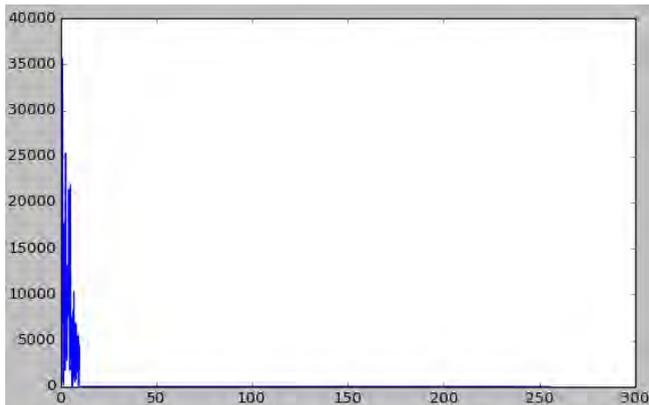


**Gambar 3.11 Ilustrasi *cut-off frequency***

Nilai frekuensi dikalikan dengan nilai yang ditunjukkan pada garis merah (bernilai 0-1). Pada gambar 3.11 daerah yang berada diantara dua garis merah akan dikalikan dengan satu, sedangkan untuk daerah yang lain dikalikan 0 (*cut-off*).

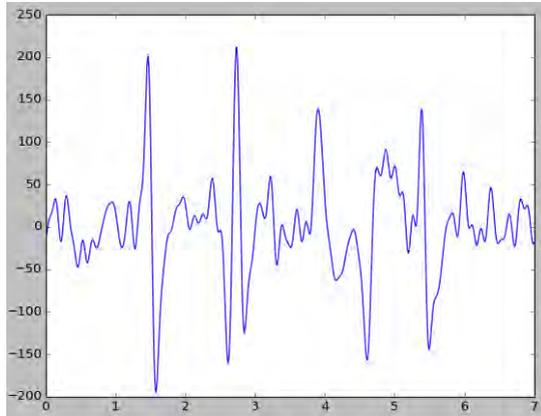
Pada gambar 3.9 diketahui bahwa frekuensi di sekitar 50 Hz memiliki nilai paling tinggi. Ini menunjukkan data sinyal masukan

memiliki frekuensi dominan pada 50 Hz. Hal ini dikarenakan interferensi gelombang listrik dari lingkungan sekitar. Frekuensi gelombang listrik yang berlaku di Indonesia adalah 50 Hz. Pemotongan frekuensi pada *frequency cut-off* ini dilakukan pada 0,0 Hz dan 10 Hz. Frekuensi di bawah 0,0 Hz dan diatas 10 Hz akan dikalikan dengan nol. Gambar 3.12 menunjukkan sinyal masukan pada gambar 3.8 dalam domain frekuensi yang telah dipotong.



**Gambar 3.12 Domain frekuensi yang telah terpotong**

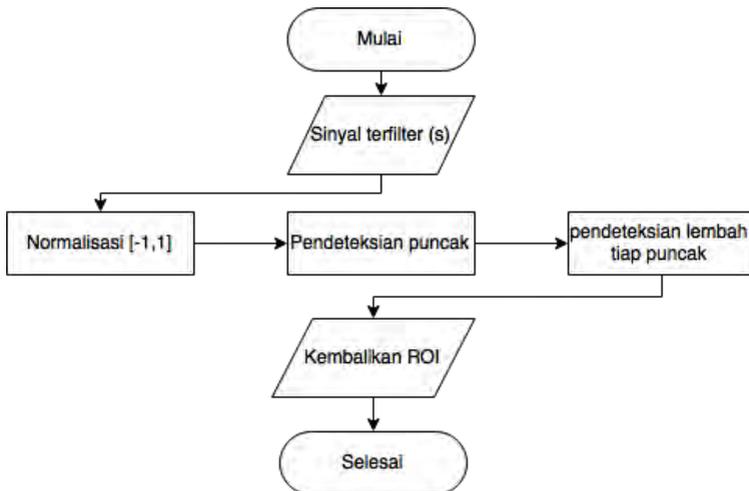
*Inverse Fourier Transform* akan diterapkan pada luaran domain frekuensi yang telah didapatkan sebelumnya. Implementasi *Inverse Fourier Transform* yang digunakan pada pengerjaan tugas akhir ini menggunakan pustaka *scipy.fftpack* pada fungsi *ifft*. Luaran dari fungsi ini adalah jajaran bilangan imajiner sejumlah jajaran masukan, dalam hal ini adalah sinyal domain frekuensi yang telah dipotong. Gambar 3.13 menunjukkan sinyal luaran pada *preprocessing*.



**Gambar 3.13** Gambar sinyal luaran *preprocessing*

### 3.3.3.2 Tahap Segmentasi *Region of Interest (ROI)*

Pada tahap pendeteksian ROI, terdapat dua proses utama, yaitu normalisasi sinyal masukan terfilter dan pendeteksian puncak. Gambar 3.14 menunjukkan bagan dari tahap segmentasi ROI secara umum.



**Gambar 3.14** Bagan tahap segmentasi ROI secara umum

Sinyal masukan terfilter selanjutnya akan disebut sebagai sinyal masukan dinormalisasi menjadi rentang -1 dan 1 dan memiliki rata-rata (*baseline*) 0. Gambar 3.14 menunjukkan *pseudocode* untuk normalisasi sinyal masukan.

1	<b>normalize(arr)</b>
2	MEAN = mean(arr)
3	arr_top = copy(arr)
4	arr_bot = copy(arr)
5	arr_top[arr_top <= MEAN] = MEAN
6	arr_bot[arr_bot > MEAN] = MEAN
7	arr_topnorm = 0 + (arr_top - min(arr_top)) * 1. / (max(arr_top) - min(arr_top))
8	arr_botnorm = -1 + (arr_bot - min(arr_bot)) * 1. / (max(arr_bot) - min(arr_bot))
9	arr_norm = arr_topnorm + arr_botnorm
10	<b>return</b> arr_norm

**Gambar 3.15 Pseudocode normalisasi sinyal**

Dengan *arr* adalah barisan bilangan desimal yang merepresentasikan sinyal masukan dan *arr\_norm* adalah sinyal masukan yang telah dinormalisasi.

Normalisasi dilakukan secara terpisah antara sinyal rata-rata atas dan sinyal rata-rata bawah. Hal ini dikarenakan untuk menjadikan rata-rata sinyal luaran mendekati 0.

Setelah mengalami normalisasi, deteksi puncak diterapkan pada sinyal masukan. Deteksi puncak yang digunakan adalah pendeteksian berdasarkan gradien / kemiringan antara nilai-nilai yang berdekatan. Suatu nilai dikatakan puncak jika dan hanya jika terjadi perubahan tanda dari positif menjadi negatif atau negatif menjadi positif dari variabel barisan gradien yang dihasilkan sinyal masukan. Pernyataan ini ditunjukkan oleh *pseudocode* gambar 3.16 pada baris 11.

1	<b>findpeaks (arr)</b>
2	N = length(arr)
3	MEAN = mean(arr)
4	arr norm = normalize(arr)
5	diff = [0..N-1]
6	peaks = []
7	<b>FOR</b> i <b>IN</b> [0..N-2]
8	diff[i] = arr[i] - arr[i + 1]
9	<b>END FOR</b>
10	<b>FOR</b> i <b>IN</b> [0..N-1]
11	<b>IF</b> diff[i] < 0 <b>AND</b> diff[i + 1] > 0
12	<b>AND</b> arr norm[i] > 0.3 + MEAN
13	peaks = [peaks i]
14	<b>END IF</b>
15	<b>END FOR</b>
16	<b>return</b> peaks

**Gambar 3.16 Pseudocode pendeteksian puncak**

Dengan *arr* adalah variabel sinyal bertipe barisan bilangan desimal (*array of float*) dan *peaks* adalah posisi-posisi dimana puncak pada variabel *arr* berada.

Tahap terakhir pendeteksian ROI adalah pemilihan lembah pada masing-masing puncak yang telah ditentukan. Setiap ROI dibatasi oleh satu titik puncak dan satu titik lembah. Titik-titik puncak yang didapatkan tidak semuanya akan menjadi puncak masing-masing ROI. Beberapa titik puncak yang tidak memenuhi kriteria akan dihapus. Kriteria-kriteria tersebut adalah titik-titik puncak harus memiliki nilai lebih besar dari 0,3 (gambar 3.16 baris 7) ditambah dengan rata-rata sinyal masukan serta titik puncak tersebut harus memiliki lembah yang memiliki nilai lebih kecil dari -0.3 dikurangi dengan rata-rata sinyal masukan [15]. Lembah dari masing-masing ROI adalah lembah yang memiliki nilai paling kecil diantara lembah-lembah yang terdapat diantara dua titik puncak. Gambar 3.17 menunjukkan *pseudocode* lengkap pendeteksian ROI.

1	<b>DetectROI (arr)</b>
2	arr_norm = normalize(arr)
3	peaks = findpeaks(arr)
4	MEAN = mean(arr_norm)
5	LEN = length(arr_norm)
6	LEN_peaks = lean(peaks)
7	arr_norm[arr_norm < 0.3 + MEAN] = 0
8	ROI = []
9	<b>FOR</b> idx <b>IN</b> [0..LEN_peaks-2]
10	bt = findpeaks(-1 * arr[peaks[idx]:peaks[idx + 1]])
11	max_id = argmax(-1 * arr_norm[bt + peaks[idx]])
12	ROI = [ROI arr_norm[arr_norm[idx] : bt[max_id] + arr_norm[idx]]]
13	<b>END FOR</b>
14	<b>return</b> ROI

**Gambar 3.17 Pseudocode pendeteksi ROI**

Dengan *arr* adalah variabel sinyal bertipe barisan bilangan desimal (*array of float*) dan *ROI* adalah jajaran rentang bilangan bulat yang merupakan potongan *region of interest* pada variabel *arr*.

### 3.3.3.3 Tahap *Postprocessing*

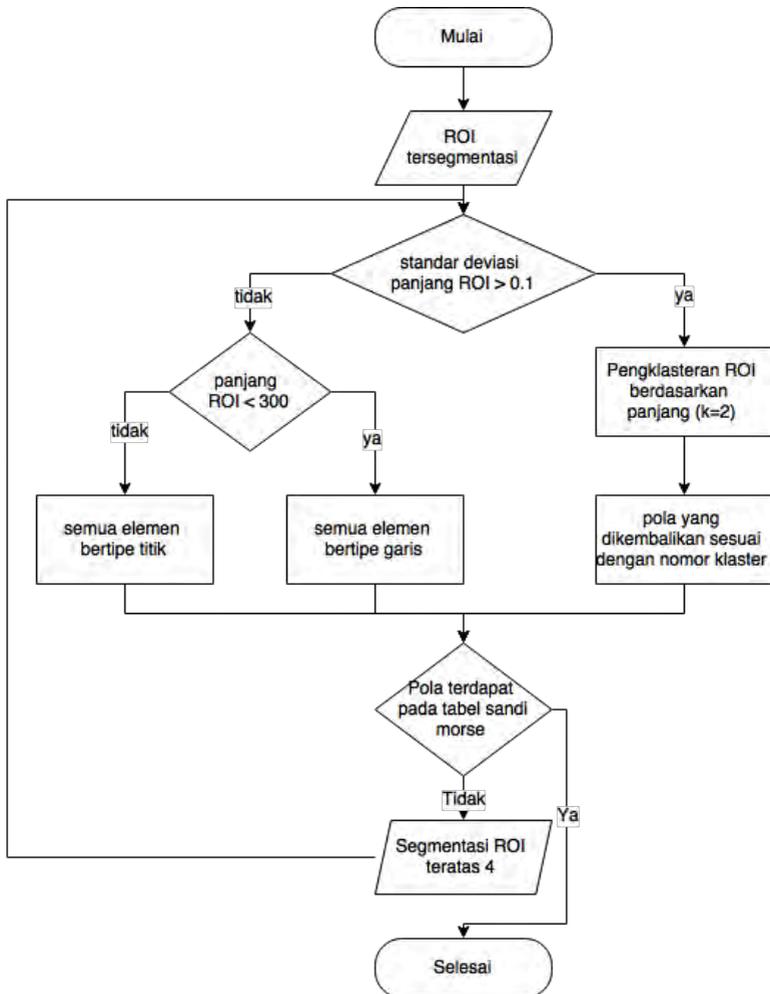
Pada tahap ini, ROI yang telah terdeteksi akan diterjemahkan menjadi titik atau garis. Terdapat dua kondisi utama sebelum penandaan titik garis dilakukan. Kemungkinannya adalah sinyal masukan terdiri dari elemen yang seragam (titik semua atau garis semua) atau sinyal masukan terdiri dari elemen yang berbeda (titik dan garis). Perbedaan kemungkinan elemen penyusun ini menggunakan standard deviasi. Jika standard deviasi panjang ROI ternormalisasi [0-1] melebihi 0,1, maka sinyal masukan terdiri dari elemen yang berbeda (heterogen), jika kurang dari nilai batas (*threshold*) maka sinyal masukan tersusun atas elemen yang sama (homogen). Gambar 3.18 menunjukkan bagan umum dari tahap

*postprocessing*. Sinyal masukan yang memiliki elemen penyusun yang sama (homogen) akan diperiksa, apakah elemen penyusun tersebut titik atau garis. Jika panjang maksimal dari ROI melebihi 300 maka elemen penyusun merupakan garis dan sebaliknya. *K-means Clustering* digunakan untuk menandai elemen pada sinyal heterogen. Penggunaan *K-means* membantu penandaan elemen secara otomatis karena tidak memerlukan nilai batas (*threshold*) untuk membedakan antara titik dan garis. Setelah setiap ROI pada sinyal masukan ditandai, maka kode yang diperoleh akan dibandingkan dengan tabel sandi morse yang tersedia. Terdapat dua kemungkinan, kemungkinan yang pertama jika kode ditemukan di dalam daftar sandi morse dan kemungkinan yang kedua kode tidak terdapat pada daftar sandi morse. Jika kode tidak terdapat pada sandi morse, pendeteksian ROI ulang akan dilakukan pada sinyal masukan.

Perbedaan pada pendeteksian ulang ini adalah jumlah ROI yang dikembalikan oleh fungsi pada gambar 3.17 dibatasi menjadi 4. Hal ini didasarkan pada jumlah maksimal elemen pada sandi morse adalah 4. Jika kode masih tidak ditemukan pada tabel sandi morse, maka sinyal masukan termasuk pada kelas *Unknown*. Gambar 3.19 adalah *pseudocode* untuk tahap *postprocessing*.

Fungsi *maskchar* pada Gambar 3.19 adalah fungsi yang mengembalikan kode menjadi bentuk titik dan garis dari ROI yang telah diberikan label pada proses sebelumnya. Proses pelabelan ROI dimulai dari baris ke 11.

Variabel-variabel penting dalam *pseudocode* ini antara lain : *arr* adalah variabel sinyal bertipe barisan bilangan desimal (*array of float*) dan *result* adalah kode karakter yang terdapat pada tabel sandi morse.



**Gambar 3.18** Diagram tahap *postprocessing* secara umum

1	<b>Guesschar (arr)</b>
2	VISIT = false
3	ROIs = DetectROI(arr)
4	DO AGAIN::
5	len ROIs = []

6	result = []
7	<b>FOR</b> i <b>IN</b> [0..length(ROIs)]
8	len ROIs = [len ROIs length(ROIs[i])]
9	<b>END FOR</b>
10	STD ROI = std(norm(len ROIs))
11	<b>IF</b> STD ROI > 0.1
12	idx = KMeans(STD ROI, 2)
13	<b>FOR</b> i <b>IN</b> [0..length(len ROIs)-2]
14	<b>IF</b> idx[i] != idx[i + 1]
15	<b>IF</b> idx[i] > idx[i + 1]
16	idx==0 is ' '
17	idx==1 is '.'
18	<b>ELSE</b>
19	idx==0 is '.'
20	idx==1 is ' '
21	<b>END IF</b>
22	<b>END IF</b>
23	<b>END FOR</b>
24	result = maskchar(result)
25	<b>ELSE</b>
26	<b>IF</b> len ROIs[0] > 300
27	result = maskchar(result, ' ')
28	<b>ELSE</b>
29	result = maskchar(result, '.')
30	<b>END IF</b>
31	<b>END IF</b>
32	<b>IF</b> result <b>IN</b> MORSE TABLE
33	return result
34	<b>ELSE</b>
35	ROIs = DetectROI(arr, max = 4)
36	<b>IF</b> not VISIT
37	VISIT = true
38	GOTO DO AGAIN
39	<b>END IF</b>
40	<b>END IF</b>

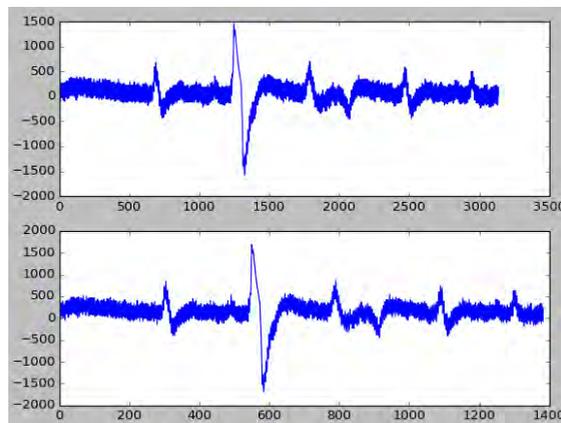
**Gambar 3.19 Pseudocode tahap postprocessing**

### 3.3.4 Tahap Khusus Pengenalan Berdasarkan Pembelajaran

Pada bagian ini, akan dijelaskan tahap-tahap yang dilakukan pada metode pendeteksian berdasarkan pembelajaran menggunakan pengklasifikasi *k-Nearest Neighbor* dengan *Dynamic Time Warping* sebagai pengukuran jarak.

#### 3.3.4.1 Persiapan Data Latih

Data latih yang digunakan pada metode ini berasal dari 80 % data yang diperoleh pada proses pengambilan data. Untuk mempercepat proses pengenalan, diterapkan standardisasi ukuran data. Standardisasi ukuran data dilakukan dengan memotong frekuensi data pada posisi seragam untuk semua data. Hal ini dilakukan karena pemotongan data pada domain frekuensi tidak merubah bentuk data secara signifikan namun menghasilkan data dengan ukuran yang lebih kecil. Gambar 3.20 menunjukkan data asli (atas) dan data yang telah mengalami pemotongan (bawah).



**Gambar 3.20 Data asli (atas), data hasil pemotongan (bawah)**

Kumpulan data latih yang telah distandardisasi selanjutnya diolah untuk membentuk *template*. *Template* adalah sinyal-sinyal dalam satu kelas yang telah mengalami penyusunan ulang. Metode penyusunan ulang (*realignment*) sinyal yang digunakan pada tugas

akhir ini adalah *Dynamic Time Warping*. Penggunaan fungsi *Dynamic Time Warping* pada pengerjaan tugas akhir ini menggunakan pustaka *mlpy* pada bahasa pemrograman *Python*. Fungsi pemilihan *template* menghasilkan *n template*, dimana masing-masing *template* mewakili *subclass* / kluster dalam masing-masing kelas. Pemilihan *subclass* yang digunakan dalam tugas akhir ini menggunakan *K-Means* dan *Hierarchical Clustering*. Fungsi *K-Means* yang digunakan diambil dari pustaka *sklearn* pada bahasa pemrograman *Python*. Gambar 3.21 menunjukkan *pseudocode* pembentukan *template* dengan *K-Means*.

1	<b>KMeansTemplateBuilder</b> (candidates, n template)
2	signal = []
3	N = length(candidates)
4	e = min(n template, N)
5	cluster = KMeans(e, candidates)
6	<b>FOR</b> i <b>IN</b> [0..e]:
7	j = argwhere(cluster == i)
8	M = length(j)
9	cand = candidates[j]
10	template = cand[0]
11	<b>FOR</b> ix <b>IN</b> [1..M]:
12	pathDtw = dtw_path(template, cand[ix])
13	template = template[pathDtw]
14	<b>END FOR</b>
15	signal = [signal template]
16	<b>END FOR</b>
17	<b>return</b> signal

**Gambar 3.21** Pemilihan *template* sinyal menggunakan *K-Means Clustering*

Metode pembentukan *template* selanjutnya berdasarkan pada *hierarchical clustering*. Gambar 3.22 adalah *pseudocode* pembentukan *template* dengan metode *hierarchical clustering*.

1	<b>HierTemplateBuilder</b> (candidates, n_template)
2	signal = candidates
3	N = length(signal)
4	n_template = min(n_template, len(signal))
5	<b>WHILE</b> length(signal) > n_template:
6	minval = inf
7	a = -1
8	b = -1
9	<b>FOR</b> i <b>IN</b> [0..length(signal)]:
10	<b>FOR</b> j <b>IN</b> [i + 1..length(signal)]:
11	dist = dtw(signal[i], signal[j])
12	<b>IF</b> dist < minval:
13	minval = d
14	a = i
15	b = j
16	<b>END IF</b>
17	<b>END FOR</b>
18	<b>END FOR</b>
19	siga = pop(signal, a)
20	sigb = pop(signal, b - 1)
21	pathDtw = dtw_path(siga, sigb)
22	signal = [signal siga[pathDtw]]
23	<b>END WHILE</b>
24	<b>RETURN</b> signal

**Gambar 3.22** Pemilihan *template* sinyal menggunakan *Hierarchical Clustering*

*Template* akan dibangun dari  $N$  kandidat. Pemilihan *template* pada *hierarchical clustering* diawali dengan menggabungkan dua sinyal terdekat. Penggabungan sinyal menggunakan algoritma DTW. Setelah mengalami penggabungan, jumlah kandidat menjadi  $N-1$ . Proses ini akan diulang selama jumlah kandidat lebih dari  $n\_template$ .

Pada *pseudocode* di atas, *candidates* adalah himpunan sinyal untuk masing-masing kelas, *n\_template* adalah jumlah *template* yang ingin dibentuk, dan *signal* adalah daftar *template* sebagai luaran fungsi.

### 3.3.4.2 Tahap Klasifikasi

Pada tahap ini, sinyal masukan akan diklasifikasikan menggunakan *k-Nearest Neighbor* dengan nilai  $k = 1$ . Gambar 3.23 menunjukkan *pseudocode* dari *k-Nearest Neighbor* dengan nilai  $k = 1$ .

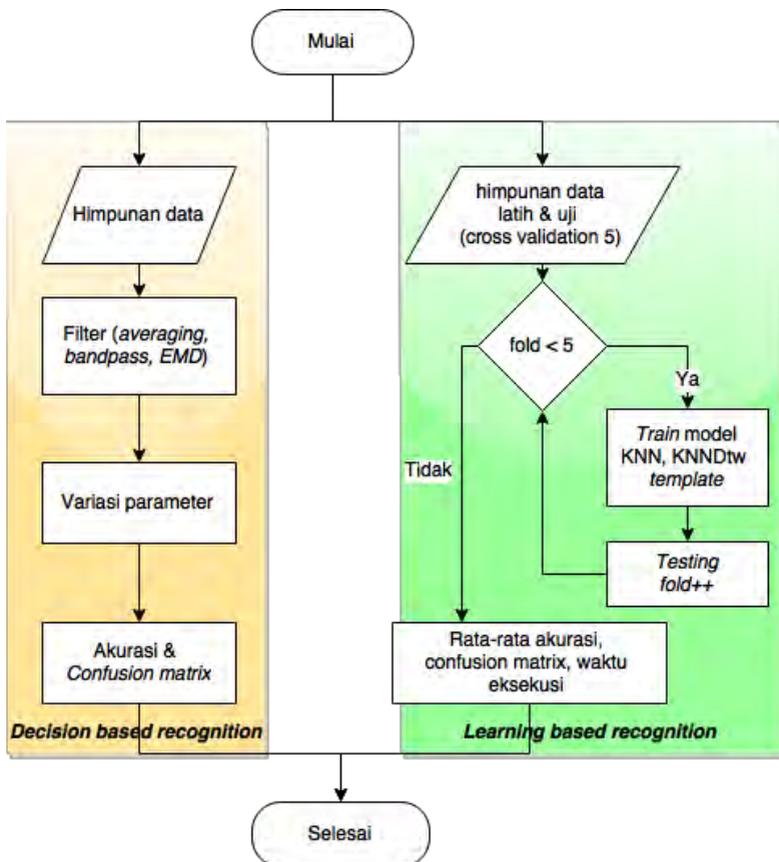
1	<b>kNN</b> ( <i>test</i> , <i>templates</i> , <i>templatesClass</i> )
2	$M = \text{length}(\text{templates})$
3	$\text{minDist} = \text{inf}$
4	$\text{class} = \text{None}$
5	<b>FOR</b> $j$ <b>IN</b> $[0..M]$
6	$\text{dist} = \text{DynamicTimeWarping}(\text{test}, \text{templates}[j])$
7	<b>IF</b> $\text{minDist} > \text{dist}$
8	$\text{minDist} = \text{dist}$
9	$\text{class} = \text{templateClass}[j]$
10	<b>END IF</b>
11	<b>END FOR</b>
12	<b>return</b> $\text{class}$

**Gambar 3.23 Pseudocode k-Nearest Neighbor**

Dengan *test* adalah sinyal, *templates* adalah himpunan sinyal yang telah didapatkan pada *pseudocode* gambar 3.21 atau 3.22, dan *templatesClass* adalah label untuk masing-masing *templates*.

### 3.4 Tahap Evaluasi Metode

Pada tahap ini, akan dijelaskan alur evaluasi yang digunakan untuk menguji metode yang telah dibahas pada Bab 3. Evaluasi yang dijalankan dibedakan menjadi dua jenis, yaitu pada metode pengenalan berbasis aturan dan pengenalan berbasis pembelajaran. Gambar 3.22 menunjukkan diagram alir evaluasi metode.

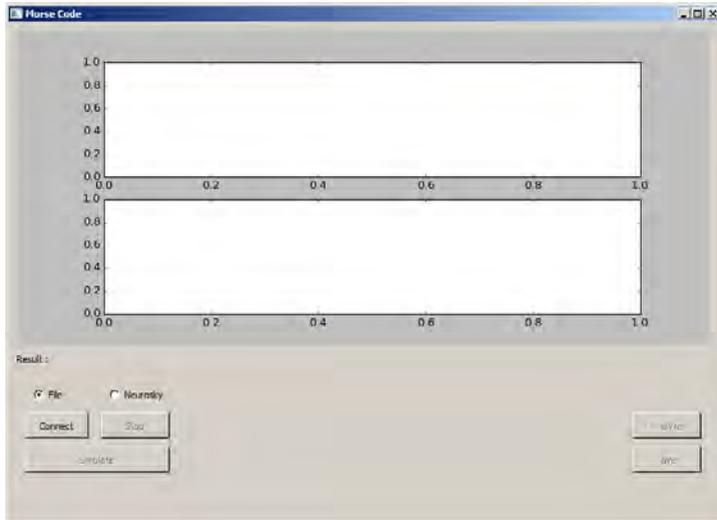


**Gambar 3.24** Diagram alir evaluasi metode

### 3.5 Perancangan Antarmuka Perangkat Lunak

Perancangan antarmuka perangkat lunak merupakan penggambaran halaman antarmuka yang akan digunakan pengguna ketika berinteraksi dengan aplikasi. Implementasi perancangan antarmuka menggunakan pustaka *PyQt* pada bahasa pemrograman *Python*. Gambar 3.25 menunjukkan rancangan tatap muka program.

Terdapat dua mode masukan, yaitu berkas sinyal simulasi yang telah tersimpan dan aliran sinyal waktu nyata dari perangkat.



**Gambar 3.25** Tatap muka utama

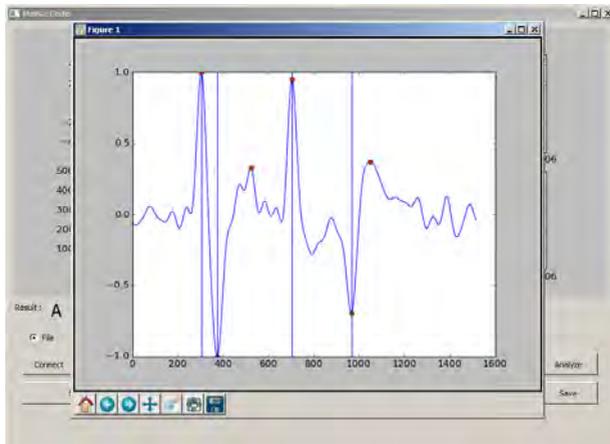
Tombol *Connect* digunakan untuk memulai simulasi. Jika mode masukan *File*, maka akan muncul kotak dialog buka yang bertujuan untuk memilih berkas yang akan disimulasikan. Jika mode masukan *NeuroSky* maka program akan menginisiasi koneksi dengan perangkat. Apabila tidak ditemukan perangkat *NeuroSky* yang terhubung pada mode ini, maka kotak dialog *error* seperti pada gambar 3.26 akan ditampilkan sebagai peringatan. Untuk menjalankan simulasi digunakan tombol *Simulate*. Ketika simulasi berjalan, aliran data sinyal akan ditampilkan pada elemen plot pada bagian atas, sedangkan elemen plot pada bagian bawah menunjukkan terfilter. Untuk menghentikan simulasi digunakan tombol *Stop*.



**Gambar 3.26 Kotak dialog *error* yang ditampilkan jika tidak ada perangkat terhubung**

Tombol *Analyze* dan *Save* hanya akan aktif jika simulasi yang telah dihentikan (*Stop*). Tombol *Analyze* digunakan untuk melakukan prediksi sinyal yang telah tertangkap. Kotak dialog *plot* tambahan akan muncul dan menampilkan ROI pada sinyal masukan. Tampilan akhir analisis sistem ditunjukkan gambar 3.27.

Hasil prediksi karakter dari sinyal input akan ditampilkan di samping label *Result*. Tombol *Save* digunakan untuk menyimpan sinyal masukan yang telah tertangkap. Sinyal disimpan ke dalam berkas tanpa ekstensi dengan format barisan angka yang dipisahkan dengan spasi.



**Gambar 3.27** Tampilan hasil Analisis sinyal masukan

## **BAB IV**

### **IMPLEMENTASI**

Bab implementasi berisi pembahasan mengenai implementasi perangkat lunak berdasarkan perancangan yang telah dibuat. Tahap perancangan merupakan tahap dasar dari implementasi perangkat lunak.

#### **4.1 Lingkungan Implementasi**

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat Keras
  - a. Processor: Intel® Core™ i5-2410M CPU @ 2.30GHz
  - b. Memory (RAM) : 4.00 GB
  - c. Tipe Sistem: 64-bit
2. Perangkat Lunak
  - a. Sistem operasi : *Windows 7 Professional* 64 bit.
  - b. *Python 2.7.6*.
  - c. *Sublime Text Editor*.
  - d. Pustaka *numpy, sklearn, scipy, mlp, pyqt, dan matplotlib*.

#### **4.2 Implementasi Modul *NeuroskyConnector***

Modul ini adalah bagian dari sistem yang berhubungan dengan penerimaan data dari perangkat *Neurosky*. Modul ini bergantung pada aplikasi *Thinkgear* bawaan perangkat. Modul *NeuroskyConnector* akan menginisiasi saluran koneksi dan menerima aliran data dari aplikasi *Thinkgear*. Setiap detik data yang diterima oleh modul ini tidak lebih dari 512 nilai. Data yang diterima selanjutnya akan diproses atau disimpan. Gambar 4.1

menunjukkan kode program yang berfungsi melakukan pengambilan data.

1	<code>def __read_data(self, a, b):</code>
2	<code>while self.__recording:</code>
3	<code>    buff = self.c_sock.recv(8192)</code>
4	<code>    if self.__islogging:</code>
5	<code>        self.temp += buff</code>
6	<code>        curr_time = time()</code>
7	<code>        parts = buff.split('\r')</code>
8	<code>        for part in parts:</code>
9	<code>            if part.startswith('{'):</code>
10	<code>                try:</code>
11	<code>                    parseObj = json.loads(part)</code>
12	<code>                    for key in parseObj.keys():</code>
13	<code>                        if key == 'eSense' or key == 'eegPower':</code>
14	<code>                            for key2 in parseObj[key].keys():</code>
15	<code>                                if len(self.data[key2]) &gt; 4000000:</code>
16	<code>                                    self.data[key2].pop(0)</code>
17	<code>                                    self.data[key2].append((curr_time,</code>
18	<code>  parseObj[key][key2]))</code>
18	<code>                        else:</code>
19	<code>                            if len(self.data[key]) &gt; 4000000:</code>
20	<code>                                self.data[key].pop(0)</code>
21	<code>                                self.data[key].append((curr_time,</code>
22	<code>                                    parseObj[key]))</code>
22	<code>                        except:</code>
23	<code>                            print key</code>
24	<code>                        try:</code>
25	<code>                            sleep(0.05)</code>
26	<code>                        except:</code>
27	<code>                            pass</code>

**Gambar 4.1** Kode program *NeuroskyConnector*

Pada potongan kode yang ditunjukkan gambar 4.1, program melakukan penerimaan data secara terus menerus selama sistem berjalan. Data yang diterima berupa json *string* yang memiliki format seperti yang ditunjukkan gambar 3.6. Field yang digunakan hanya field *rawEeg*.

### 4.3 Implementasi Modul *StreamClass*

Modul ini berperan sebagai kelas abstrak yang menghubungkan modul *NeuroskyConnector* dan aplikasi utama. Terdapat dua model masukan yang dapat ditangani oleh aplikasi, model masukan yang berasal dari berkas sinyal yang tersimpan dan model masukan langsung dari aliran data perangkat. Gambar 4.2 menunjukkan potongan kode program representasi kelas *streamer*.

1	<code>class DataStream(HAbstractStream):</code>
2	<code>def __init__(self, filename, window_length = 500):</code>
3	<code>    HAbstractStream.__init__(self)</code>
4	<code>    self.window_length = window_length</code>
5	<code>    self.filename = filename</code>
6	<code>    self.data = np.append(np.array([np.nan for i     in range(100)]),     self.__open_data(self.filename))</code>
7	<code>def next(self):</code>
8	<code>    if self.current_cursor &lt; len(self.data):</code>
9	<code>        start = self.current_cursor</code>
10	<code>        self.current_cursor += self.delta</code>
11	<code>        temp = len(self.data[start:start +         self.window_length])</code>
12	<code>        if temp == 0:</code>
13	<code>            return []</code>
14	<code>        return np.append(self.data[start:start +         self.window_length], np.array([np.nan for i         in range(self.window_length - temp)]))</code>
15	<code>    else:</code>
16	<code>        print '0'</code>
17	<code>        return []</code>
18	<code>class NeuroStream(HAbstractStream):</code>
19	<code>def next(self):</code>
20	<code>    self.currentData =     np.array(self.neuroObj.get_recorded(withtime=     False)[-self.window_length:])</code>
21	<code>    return self.currentData</code>

**Gambar 4.2** Kode program *StreamClass*

Terdapat dua kelas yang digunakan, yaitu kelas *DataStream* dan *NeuroStream*. Kelas *DataStream* berfungsi untuk menangani aliran data yang berasal dari berkas simulasi, sedangkan *NeuroStream* berfungsi untuk menangani aliran data dari perangkat. Fungsi *next* digunakan untuk mendapatkan data setiap permunculan per detik.

#### 4.4 Implementasi Modul *MonitorClass*

Modul ini digunakan untuk menampilkan aliran data (sinyal) yang diperoleh dari perangkat maupun berkas tersimpan. Terdapat dua sub tampilan yang dihasilkan dari modul ini, yaitu tampilan sinyal asli dari perangkat dan tampilan hasil pengolahan dari sinyal dengan *bandpass (frequency cut-off) filtering*. Gambar 4.3 menunjukkan potongan kode pada kelas *Monitor*.

1	<code>class Monitor(<i>FigureCanvas</i>):</code>
2	<code>def timerEvent(<i>self</i>, <i>evt</i>):</code>
3	<code>dummy = <i>self</i>.__obj.next()</code>
4	<code>print len(dummy)</code>
5	<code>if len(dummy) == 0:</code>
6	<code>self.__obj.reset()</code>
7	<code>self.endSimulation()</code>
8	<code>return</code>
9	<code>y = bandpass(dummy, 0.6, 20)</code>
10	<code>if len(y) == 0:</code>
11	<code>self.__obj.reset()</code>
12	<code>self.endSimulation()</code>
13	<code>return</code>
14	<code>x = np.arange(len(y))</code>
15	<code>self.axes.clear()</code>
16	<code>self.axes.plot(x, y)</code>
17	<code>self.axes2.clear()</code>
18	<code>self.axes2.plot(np.abs(fft(y[y!= np.nan]))[:len(y) / 2])</code>
19	<code>self.axes.set_ylim(-500, 500)</code>
20	<code>self.axes2.set_ylim(0, 50000)</code>
21	<code>self.infig.canvas.draw()</code>
22	<code>def setDataStreamer(<i>self</i>, <i>obj</i>):</code>
23	<code>self.__obj = obj</code>

24	<code>def startSimulation(self, time):</code>
25	<code>self.timer = self.startTimer(time)</code>
26	<code>def endSimulation(self):</code>
27	<code>self.killTimer(self.timer)</code>

**Gambar 4.3** Kode program modul *MonitorClass*

Bagian terpenting dari kelas ini adalah fungsi *timerEvent*. Fungsi ini berfungsi untuk mengembalikan data yang diperoleh dari *StreamerClass* ke media plot. Setiap tampilan yang dikembalikan terdapat dua representasi sinyal, yaitu sinyal asli dan sinyal terfilter.

#### 4.5 Implementasi Modul Pendeteksi

Modul ini digunakan untuk melakukan pendeteksian sinyal. Sinyal masukan akan dideteksi pola sandi morsenya. Terdapat dua macam pengenalan, pengenalan berbasis aturan dan *learning based detection*. Metode *learning* (pembelajaran) yang digunakan adalah *k-Nearest Neighbor*. Gambar 4.4 menunjukkan fungsi-fungsi yang digunakan pada modul pendeteksi.

1	<code>MAX_DOT_THRES = 300</code>
2	
3	<code>def guesschar(s):</code>
4	<code>  r = populate_ROI(s)</code>
5	<code>  k = np.array([len(i) for i in r],</code> <code>              dtype=np.float64)</code>
6	<code>  rep = {1:'.', 0:''}</code>
7	<code>  predict = ''</code>
8	<code>  if np.std(k / np.max(k)) &gt; 0.1:</code>
9	<code>    idx = KMeans(n_clusters=2).</code> <code>          fit_predict(np.array([[len(i) for i in r],</code> <code>                          np.zeros((1, len(r)))[0]],</code> <code>                      dtype=np.float64).T)</code>
10	<code>    for i in range(len(idx) - 1):</code>
11	<code>      if idx[i] != idx[i + 1] and abs(len(r[i])</code> <code>      - len(r[i + 1])) &gt; 0:</code>
12	<code>      if len(r[i]) &gt; len(r[i + 1]):</code>
13	<code>        rep[idx[i]] = '.'</code>
14	<code>        rep[idx[i + 1]] = ''</code>



47	rep[idx[i + 1]] = '.'
48	else:
49	rep[idx[i]] = '.'
50	rep[idx[i + 1]] = '_'
51	break
52	predict_candidate = [rep[i] for i in idx]
53	predict = ''.join(predict_candidate)
54	else:
55	if min([len(i) for i in r]) < MAX_DOT_THRES:
56	predict = ''.join(['.' for i in range(len(r))])
57	else:
58	predict = ''.join(['_' for i in range(len(r))])
59	return predict

**Gambar 4.4 Modul pengenalan sinyal**

Terdapat dua fungsi utama dalam modul ini, yaitu *guesschar* dan *exceptional*. Fungsi *guesschar* adalah fungsi pendeteksi pengenal standar yang digunakan pada sinyal masukan. Hasil yang dikembalikan oleh sinyal ini adalah representasi titik garis dengan panjang yang tidak memiliki batasan. Fungsi ini diawali dengan pendeteksian ROI dan setiap ROI yang dideteksi akan ditentukan sebagai titik atau garis dengan menggunakan *K-Means* dengan nilai  $K = 2$ . Fungsi *exceptional* mempunyai cara kerja yang sama dengan fungsi *guesschar*, yang membedakan adalah fungsi ini digunakan ketika hasil yang diperoleh fungsi *guesschar* tidak terdapat pada tabel sandi morse. Jumlah ROI yang dikembalikan pada fungsi *exceptional* selalu tidak lebih dari empat. Hal ini berdasarkan bahwa jumlah elemen maksimal yang terdapat pada karakter sandi morse adalah empat buah. Terdapat kemungkinan bahwa kombinasi elemen sebanyak empat tidak terdapat pada tabel sandi morse. Jika hal ini terjadi, maka data akan diklasifikasikan sebagai kelas *Unknown*.

#### 4.6 Implementasi Modul Evaluasi

Modul ini berisi skenario-skenario uji coba dan evaluasi. Terdapat dua skenario utama, yaitu pengenalan berdasarkan pembelajaran dan pengenalan berdasarkan aturan. Gambar 4.5 menunjukkan potongan kode program yang digunakan pada saat evaluasi sistem pengenalan.

1	MORSE_TABLE = json.loads(JSON_STRING)
2	def RuleBased(fileList, a, b):
3	dict_result = [[0 for i in range(27)] for j in range(27)]
4	time_detection = []
5	accur = 0
6	for n in fileList:
7	s = open_data('Data/' + n)
8	start = clock()
9	pred = guesschar(s, mfilter=1, params=[a, b])
10	if pred in MORSE_TABLE:
11	if MORSE_TABLE[pred] == n[0]:
12	accur += 1
13	dict_result[ord(n[0]) - ord('a')][ord(MORSE_TABLE[pred]) - ord('a')] += 1
14	else:
15	pred = exceptional(s, mfilter=1, params=[a, b])
16	if pred in MORSE_TABLE:
17	if MORSE_TABLE[pred] == n[0]:
18	accur += 1
19	dict_result[ord(n[0]) - ord('a')][ord(MORSE_TABLE[pred]) - ord('a')] += 1
20	else:
21	dict_result[ord(n[0]) - ord('a')][26] += 1
22	time_detection.append(clock() - start)
23	return dict_result, float(accur) / len(fileList), time_detection
24	def KNNdtw(fileList, a = -1):
25	kf = KFold(len(na), n_folds=5, shuffle=True,

	<code>random_state=3)</code>
26	<code>feat = [open_data('New/' + n) for n in filelist]</code>
27	<code>result = []</code>
28	<code>for trains, tests in kf:</code>
29	<code>    templates, chClass = [], []</code>
30	<code>    learning_start = clock()</code>
31	<code>    for ch in range(97, 97 + 26):</code>
32	<code>        cand = [feat[i] for i in trains if                 na[i][0] == chr(ch)]</code>
33	<code>        if len(cand) != 0:</code>
34	<code>            temp = templateBuilder(cand, e = a)</code>
35	<code>            for t in temp:</code>
36	<code>                templates.append(t)</code>
37	<code>                chClass.append(chr(ch))</code>
38	<code>    learning_end = clock()</code>
39	<code>    print 'Template created'</code>
40	<code>    accur = 0</code>
41	<code>    avg_detect_time = 0.0</code>
42	<code>    for test in tests:</code>
43	<code>        detect_start = clock()</code>
44	<code>        minVal, cla = np.inf, None</code>
45	<code>        for i in range(len(templates)):</code>
46	<code>            d = mlpw.dtw_std(feat[test],                             templates[i])</code>
47	<code>            if minVal &gt; d:</code>
48	<code>                minVal = d</code>
49	<code>                cla = chClass[i]</code>
50	<code>        detect_end = clock()</code>
51	<code>        if cla == na[test][0]:</code>
52	<code>            accur += 1</code>
53	<code>            avg_detect_time += (detect_end -                                 detect_start)</code>
54	<code>    result.append((learning_end -                   learning_start, avg_detect_time /                   len(tests), float(accur) / len(tests) *100))</code>
55	<code>    return result</code>
56	<code>def KNNEuclid(filelist, a = -1):</code>
57	<code>    kf = KFold(len(na), n_folds=5, shuffle=True,                random_state=3)</code>
58	<code>    feat = [open_data('New/' + n) for n in</code>

	filelist]
59	result = []
60	for trains, tests in kf:
61	templates, chClass = [], []
62	learning_start = clock()
63	for ch in range(97, 97 + 26):
64	cand = [feat[i] for i in trains if na[i][0] == chr(ch)]
65	if len(cand) != 0:
66	temp = templateBuilder(cand, e = a)
67	for t in temp:
68	templates.append(t)
69	chClass.append(chr(ch))
70	learning_end = clock()
71	accur = 0
72	avg_detect_time = 0.0
73	for test in tests:
74	detect_start = clock()
75	minVal, cla = np.inf, None
76	for i in range(len(templates)):
77	templates[i] = np.real (ifft(fft(templates[i]) [:len(feat[tests[0]])])))
78	for i in range(len(templates)):
79	d = np.sqrt(np.sum((feat[test] - templates[i]) ** 2))
80	if minVal > d:
81	minVal = d
82	cla = chClass[i]
83	detect_end = clock()
84	if cla == na[test][0]:
85	accur += 1
86	avg_detect_time += (detect_end - detect_start)
87	result.append((learning_end - learning_start, avg_detect_time / len(tests), float(accur) / len(tests) * 100))
88	return result

**Gambar 4.5 Kode program evaluasi skenario**

Terdapat tiga fungsi utama pada potongan kode yang ditunjukkan gambar 4.5, yaitu *RuleBased* (baris ke 2), *KNNEuclid* (baris 56) dan *KNNDtw* (baris 24). Skenario *RuleBased* terdiri dari 3 sub skenario, yaitu dengan metode *filtering bandpass*, *window averaging* dan EMD. Perbedaan skenario ini ditentukan oleh parameter masukan fungsi yang direpresentasikan dengan variabel *mfilter*. Pada fungsi *KNNEuclid* dan *KNNDtw* terdapat dua variasi untuk membangkitkan *template*, yaitu dengan algoritma *K-Means* atau *Hierarchical clustering*. Perbedaan fungsi *KNNEuclid* dan *KNNDtw* adalah penggunaan DTW pada fungsi *KNNDtw* sebagai pengukur jarak.

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini, dibahas uji coba dan evaluasi metode-metode yang digunakan. Hal-hal yang diujikan dalam bab ini adalah: performa metode filtering *fourier transform* dan *empirical mode decomposition* untuk *preprocessing* sinyal, serta metode pendeteksian puncak pada data masukan.

#### **5.1 Lingkungan Uji Coba**

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

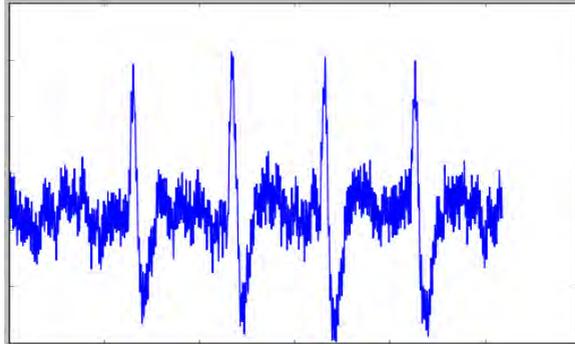
1. Perangkat Keras
  - a. Processor: Intel® Core™ i5-2410M CPU @ 2.30GHz
  - b. Memory (RAM) : 4.00 GB
  - c. Tipe Sistem: 64-bit
2. Perangkat Lunak
  - a. Sistem operasi : *Windows 7 Professional* 64 bit.
  - b. *Python 2.7.6*.
  - c. *Sublime Text Editor*.
  - d. Pustaka *numpy*, *sklearn*, *scipy*, *mly*, *pyqt*, dan *matplotlib*.

#### **5.2 Data Uji Coba**

Data yang digunakan sebagai uji coba didapatkan dari tiga pengguna. Perekaman data menggunakan perangkat tidak selalu menghasilkan data yang bagus. Hal ini bisa diakibatkan oleh pemasangan perangkat, kondisi lingkungan, kondisi mental subjek yang direkam dan lain-lain. Perekaman data diulang untuk mendapatkan data yang sesuai asumsi. Sehingga didapatkan 552 data dari sekitar 1500 perekaman yang dilakukan.

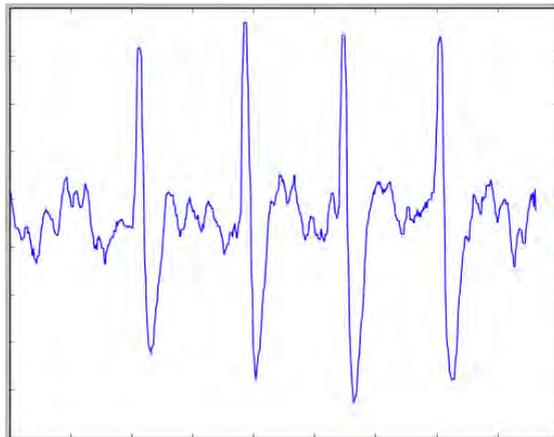
### 5.3 Uji Coba Proses Pengenalan

Pada bagian ini akan ditampilkan hasil dari masing-masing proses pada pengenalan sinyal berbasis aturan.

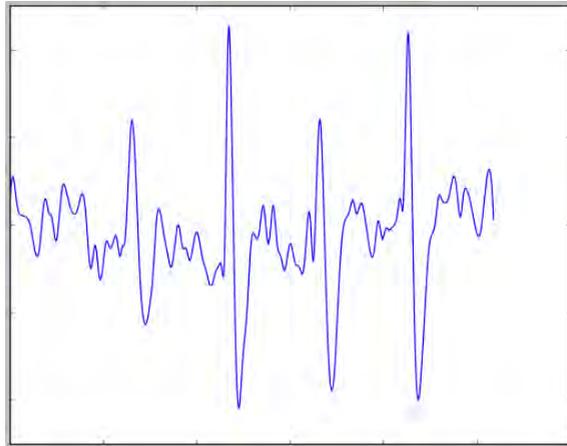


**Gambar 5.1 Sinyal masukan sistem**

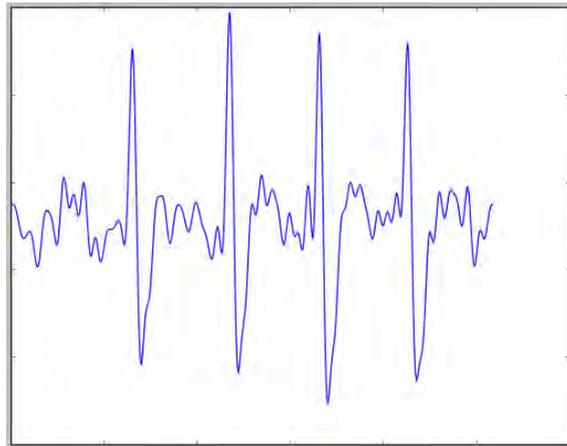
Pada tahap *preprocessing*, digunakan filter *window averaging*, *empirical mode decomposition* dan *bandpass*. Gambar 5.2, 5.3 dan 5.4 menunjukkan masing-masing dari hasil *preprocessing*.



**Gambar 5.2 Hasil filter *window averaging***

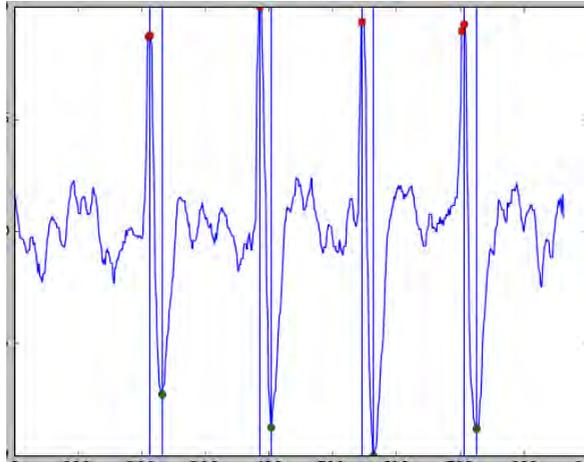


**Gambar 5.3 Hasil filter *empirical mode decomposition***

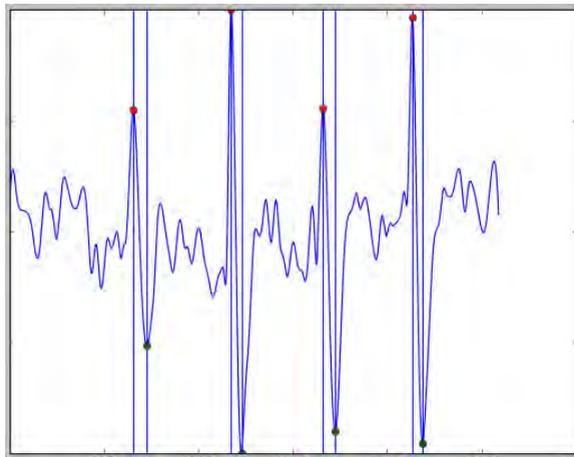


**Gambar 5.4 Hasil filter *bandpass***

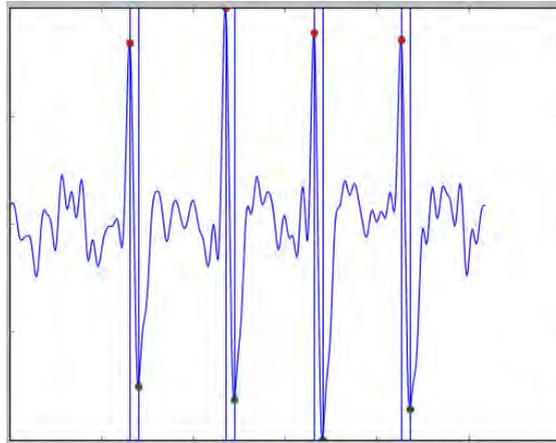
Setelah tahap ini, ROI pada sinyal dideteksi dengan menggunakan aturan-aturan yang telah dijelaskan pada Tahap *Postprocessing*. Gambar 5.5, 5.6 dan 5.7 menunjukkan hasil pendeteksian ROI untuk masing-masing filter.



**Gambar 5.5** Hasil pendeteksian ROI oleh filter *window averaging*



**Gambar 5.6** Hasil pendeteksian ROI oleh filter *empirical mode decomposition*



**Gambar 5.7 Hasil pendeteksian ROI oleh filter bandpass**

Untuk semua filter yang digunakan, pola yang dikenali adalah  $(\cdot \cdot \cdot \cdot)$  yang merupakan huruf H. Hal seperti ini dimungkinkan karena sinyal memiliki bentuk morfologis yang bagus.

#### 5.4 Skenario Uji Coba

Di bagian ini dijelaskan skenario uji coba yang dilakukan. Terdapat lima skenario, yaitu penggunaan filter *sliding window*, *empirical mode decomposition*, dan *bandpass* sebagai metode *preprocessing*, *k-Nearest Neighbor* dengan dan tanpa penyusunan ulang menggunakan *Dynamic Time Warping*. Parameter yang digunakan untuk evaluasi yaitu akurasi pengenalan dan lama eksekusi program (*running time*). Pada tabel hasil uji coba, terdapat 5 kolom. Kolom *kelas* menunjukkan kelas yang diuji, kolom *Benar* menunjukkan jumlah data pada kelas tersebut yang terklasifikasikan secara benar, kolom *Salah* menunjukkan jumlah data yang salah klasifikasi, kolom *Klasifikasi Salah* menunjukkan kelas-kelas lain yang dikenali oleh sistem dalam himpunan per kelas pada setiap baris, dan kolom *Akurasi* menunjukkan akurasi pendeteksian per kelas pada setiap baris.

### 5.4.1 Skenario Uji Coba 1

Skenario uji coba 1 adalah penggunaan filter averaging pada sliding *window* sebagai metode *preprocessing* sinyal. Filter *averaging* akan diterapkan pada sinyal masukan dengan ukuran *window* yang bervariasi. Selanjutnya, sinyal hasil *preprocessing* akan mengalami proses pendeteksian ROI dan pengenalan karakter seperti yang telah dijelaskan pada bagian Tahap Khusus Pengenalan Berdasarkan Aturan. Pada skenario uji coba 1, terdapat lima paket variasi parameter lebar *window*, yaitu 10, 20, 30, 40, dan 50. Tabel 5.1, 5.2, 5.3, 5.4 dan 5.5 masing-masing menunjukkan hasil pengenalan untuk masing-masing variasi parameter.

**Tabel 5.1 Hasil skenario 1 dengan Lebar *window* 10**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	15	6	I,T	71,43
B	18	3	D,N	85,71
C	10	9	A,G,K,M,Q,R	52,63
D	14	7	B,N,O,Z	66,67
E	8	13	A,I,J,M,R,S,T,W,#	38,10
F	9	13	A,E,G,H,K,N,R,T	40,91
G	10	12	K,M,N,P,R,S	45,45
H	19	4	B,R,S	82,61
I	7	14	C,H,M,R,S,T	33,33
J	14	8	A,H,U	63,64
K	20	2	D,M	90,91
L	9	12	A,D,R	42,86
M	0	21	I	0,00
N	20	2	T	90,91
O	0	21	M,S,W	0,00
P	11	10	G,M,R,U,W,Y	52,38
Q	4	17	I,K,M,N,O,S,T,U	19,05
R	11	10	A,H,N,S,W	52,38
S	16	5	H	76,19
T	17	4	A,I,N	80,95
U	17	4	A,T	80,95

V	8	13	A,W	38,10
W	19	2	M	90,48
X	14	8	K,S,U,V,Y	63,64
Y	15	6	K,O,W	71,43
Z	10	11	G,H,I,M,O,T	47,62
<b>Akurasi</b>				<b>57,07</b>
<b>Waktu Pengenalan</b>				<b>0,062</b>

**Tabel 5.2 Hasil skenario 1 dengan Lebar *window* 20**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	19	2	I	90,48
B	20	1	D	95,24
C	15	4	G,K,R	78,95
D	17	4	B,H,N,X	80,95
E	11	10	A,H,I,J,O,S,T,W	52,38
F	10	12	A,C,E,H,N,R,T,U	45,45
G	14	8	M,N,R,S	63,64
H	21	2	B,I	91,30
I	12	9	B,H,O,R,S,U	57,14
J	16	6	H	72,73
K	21	1	C	95,45
L	13	8	H,R	61,90
M	0	21	I	0,00
N	22	0		100,00
O	0	21	M,S,W	0,00
P	14	7	F,G,U,W,Y	66,67
Q	6	15	A,G,I,K,O,S,T,U,Y,Z	28,57
R	16	5	A,H,P	76,19
S	18	3	H	85,71
T	18	3	I,N	85,71
U	16	5	A,T,#	76,19
V	15	6	A,U,W	71,43
W	19	2	I,M	90,48
X	18	4	K,R,V,Y	81,82
Y	19	2	D,X	90,48
Z	13	8	G,I,S,T	61,90

<b>Akurasi</b>	<b>69,38</b>
<b>Waktu Pengenalan</b>	<b>0,062</b>

**Tabel 5.3 Hasil skenario 1 dengan Lebar *window* 30**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	18	3	I,T	85,71
B	19	2	D,H	90,48
C	15	4	K,R,#	78,95
D	16	5	B,H,N,S	76,19
E	12	9	A,I,M,S,T	57,14
F	11	11	A,E,G,H,N,R,U	50,00
G	14	8	N,R,S	63,64
H	23	0		100,00
I	11	10	D,M,S,#	52,38
J	16	6	H	72,73
K	20	2	U,Y	90,91
L	12	9	H,R	57,14
M	0	21	I	0,00
N	22	0		100,00
O	1	20	S	4,76
P	15	6	F,G,N,R	71,43
Q	6	15	A,G,I,K,M,O,R,S,T,U	28,57
R	17	4	A,N,S	80,95
S	19	2	H	90,48
T	18	3	E,I	85,71
U	18	3	A,#	85,71
V	19	2	U	90,48
W	19	2	I,M	90,48
X	17	5	K,Q,R,S,T	77,27
Y	19	2	O,X	90,48
Z	15	6	G,I,T	71,43
<b>Akurasi</b>				<b>71,01</b>
<b>Waktu Pengenalan</b>				<b>0,08</b>

**Tabel 5.4 Hasil skenario 1 dengan Lebar *window* 40**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	19	2	I	90,48
B	18	3	D,H	85,71
C	15	4	F,G,K,R	78,95
D	18	3	B,N,S	85,71
E	10	11	I,M,N,O,T	47,62
F	9	13	A,E,G,H,N,R,S	40,91
G	14	8	D,M,R,S	63,64
H	23	0		100,00
I	13	8	D,M,O,S	61,90
J	17	5	H	77,27
K	20	2	U,Y	90,91
L	11	10	H,R	52,38
M	0	21	I	0,00
N	22	0		100,00
O	1	20	S	4,76
P	16	5	F,G,R	76,19
Q	4	17	A,K,O,R,S,T,U,Y,Z	19,05
R	14	7	A,H,N,S,W	66,67
S	19	2	H	90,48
T	20	1	I	95,24
U	15	6	A,T,#	71,43
V	17	4	K,U	80,95
W	17	4	I,M	80,95
X	14	8	B,K,Q,R,S,T	63,64
Y	18	3	K,O	85,71
Z	11	10	G,I,O,T	52,38
<b>Akurasi</b>				<b>67,93</b>
<b>Waktu Pengenalan</b>				<b>0,11</b>

**Tabel 5.5 Hasil skenario 1 dengan Lebar *window* 50**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	19	2	I	90,48
B	18	3	D,H	85,71
C	14	5	F,G,K,R	73,68

D	18	3	B,N,S	85,71
E	11	10	A,I,M,N,T	52,38
F	9	13	A,E,G,H,I,N,R,S,W	40,91
G	13	9	D,M,N,R,S	59,09
H	23	0		100,00
I	12	9	H,M,R,S	57,14
J	18	4	H	81,82
K	19	3	S,U,Y	86,36
L	13	8	H,R	61,90
M	0	21	I	0,00
N	22	0		100,00
O	1	20	S	4,76
P	15	6	F,G,M,R,W	71,43
Q	6	15	A,I,K,M,N,O,S,T,U,Y	28,57
R	14	7	A,N,S,W	66,67
S	16	5	H	76,19
T	18	3	E,I,N	85,71
U	12	9	A,E,T	57,14
V	11	10	A,K,M,U,W	52,38
W	14	7	I,M	66,67
X	11	11	Q,S,T	50,00
Y	16	5	H,O,X	76,19
Z	12	9	G,I,M,O,T	57,14
<b>Akurasi</b>				<b>64,31</b>
<b>Waktu Pengenalan</b>				<b>0,11</b>

Berdasarkan hasil yang telah didapatkan pada uji coba skenario 1, didapat bahwa nilai akurasi tertinggi adalah 71,01% dengan panjang window 30. Besar *overlapping array* antar window adalah 3 petak. Dari akurasi masing-masing ukuran *window* yang diujicobakan, dapat diketahui bahwa semakin besar ukuran *window*, semakin besar waktu yang dihabiskan untuk mengenali pola sinyal. Namun perbedaan lama waktu pengenalan antar variasi tidak terlalu signifikan, sehingga hal seperti ini bisa diabaikan.

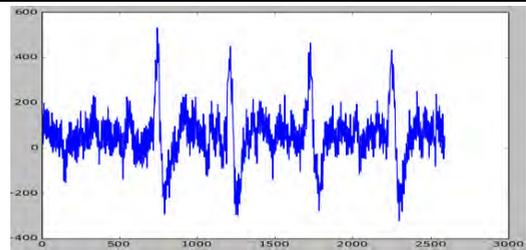
Dari semua nilai akurasi pada uji coba ini, didapatkan bahwa ukuran *window* 20-50 memiliki nilai signifikansi yang rendah

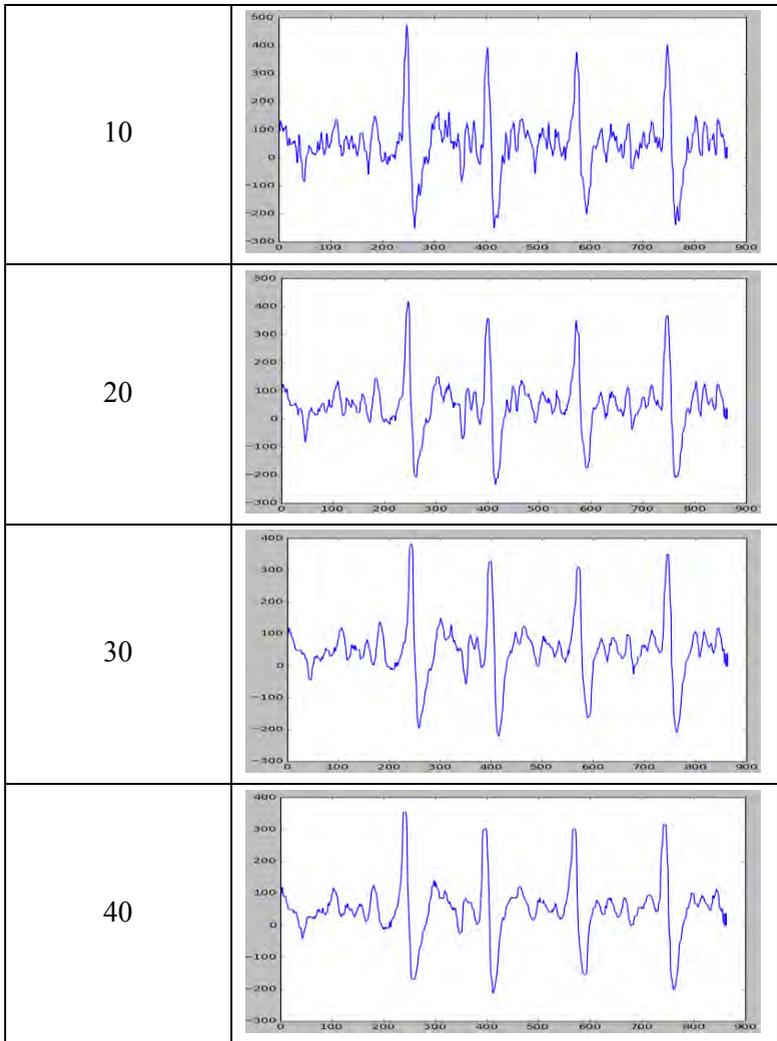
sehingga nilai *window* 20-50 adalah pilihan terbaik untuk digunakan sebagai parameter pada metode ini. Pada semua hasil uji coba, diketahui bahwa kelas M tidak pernah dideteksi sebagai kelas M. Hal ini dikarenakan ukuran sinyal berubah dan pola huruf M (- -) memiliki kesamaan dengan I (. .). Kesamaan pola namun berbeda penyusun ini memberikan peluang kesalahan ketika proses klastering elemen. Hasil rekapitulasi skenario uji coba 1 ditunjukkan oleh tabel 5.6, sedangkan tabel 5.7 menunjukkan contoh hasil pengolahan untuk masing-masing ukuran *window*.

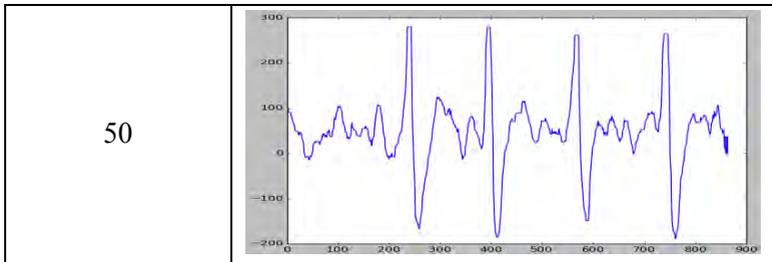
**Tabel 5.6 Rekapitulasi skenario 1**

Lebar <i>Window</i>	Akurasi (%)	Rata-rata waktu pengenalan (det)
10	57,07	0,062
20	69,38	0,062
30	71,01	0,08
40	67,93	0,11
50	64,31	0,11

**Tabel 5.7 Contoh pengolahan masing-masing *window***

Lebar <i>Window</i>	Hasil Pengolahan
Asli	





### 5.4.2 Skenario Uji Coba 2

Skenario uji coba 2 adalah penggunaan filter *empirical mode decomposition* sebagai metode *preprocessing* sinyal. Metode ini akan diterapkan pada sinyal masukan dengan jumlah *intrinsic mode function* terpilih yang bervariasi. Selanjutnya sinyal akan dikenali polanya seperti yang telah dijelaskan bagian Tahap Khusus Pengenalan Berdasarkan Aturan. Pada skenario ini, terdapat variasi parameter pengambilan IMF. Hal ini bertujuan untuk mengetahui indeks IMF terbaik yang merepresentasikan sinyal EOG. Variasi parameter tersebut adalah (format a:b, a = indeks awal IMFs, b = indeks akhir IMFs dalam bilangan negatif) 1:end-1, 1:end-2, 2:end-1, 2:end-2, 3:end-1, 3:end-2, 4:end-1, dan 4:end-2. Tabel 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 dan 5.15 masing-masing menunjukkan hasil pengenalan untuk masing-masing variasi parameter.

**Tabel 5.8 Hasil skenario 2 dengan parameter 1:end-1**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	8	13	I,T	38,10
B	2	19	D,N	9,52
C	4	15	G,M	21,05
D	2	19	H,K,N,T,Z	9,52
E	6	15	I,K,N,R,T	28,57
F	8	14	A,C,E,G,N,R,T,U,V	36,36
G	12	10	K,M,N,O,R,S	54,55
H	16	7	B,N,R,S	69,57
I	9	12	A,B,D,N,R,S	42,86

J	6	16	A,M,O,W	27,27
K	7	15	M,T	31,82
L	3	18	A,N,R,T	14,29
M	19	2	T	90,48
N	12	10	T	54,55
O	18	3	M	85,71
P	7	14	B,F,G,M,R,W	33,33
Q	9	12	G,K,M,O	42,86
R	6	15	G,L,M,N,S,T	28,57
S	16	5	D,F,H	76,19
T	20	1	A	95,24
U	11	10	A,K,T	52,38
V	13	8	A,D,U,W	61,90
W	16	5	A,M,T	76,19
X	8	14	A,K,M,N,T,Y	36,36
Y	13	8	K,O,W	61,90
Z	7	14	D,G,M,O,T	33,33
<b>Akurasi</b>				<b>46,74</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

**Tabel 5.9 Hasil skenario 2 dengan parameter 1:end-2**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	7	14	I,T	33,33
B	3	18	A,D,N,T	14,29
C	4	15	G,K,M,N,O,U	21,05
D	2	19	H,I,K,N,Q,T	9,52
E	5	16	A,D,I,K,N,O,T,#	23,81
F	7	15	A,C,E,G,H,N,R,T,U	31,82
G	11	11	M,N,O,R,T	50,00
H	17	6	B,N,R,S	73,91
I	9	12	A,C,D,N,S,T	42,86
J	7	15	A,M,O	31,82
K	6	16	M,Q	27,27
L	2	19	D,N,R,S,T,W	9,52
M	18	3	O,T	85,71
N	10	12	G,H,M,T	45,45

O	19	2	M	90,48
P	7	14	A,B,F,G,M,R,T,U,W	33,33
Q	5	16	G,M,N,O,#	23,81
R	3	18	A,C,K,L,N,S,U,W	14,29
S	13	8	B,D,H,I,N,T	61,90
T	17	4	A,M	80,95
U	9	12	A,K,T,W,X	42,86
V	10	11	A,T,U,W	47,62
W	15	6	M	71,43
X	8	14	A,K,M,T	36,36
Y	11	10	K,M,O,W	52,38
Z	4	17	D,G,O,T	19,05
<b>Akurasi</b>				<b>41,49</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

**Tabel 5.10 Hasil skenario 2 dengan parameter 2:end-1**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	19	2	T	90,48
B	20	1	N	95,24
C	14	5	G,K	73,68
D	15	6	G,N,S,Z	71,43
E	9	12	B,N,T	42,86
F	11	11	A,D,E,G,N,R,U,W,X	50,00
G	16	6	D,K,N,Q,R	72,73
H	19	4	B,S	82,61
I	10	11	A,B,D,L,N,R,S	47,62
J	20	2	O	90,91
K	20	2	M	90,91
L	13	8	A,N,R	61,90
M	20	1	T	95,24
N	22	0		100,00
O	18	3	J,M	85,71
P	18	3	H,J,R	85,71
Q	11	10	B,D,K,N,O,#	52,38
R	12	9	A,F,H,L,O,P,W	57,14
S	16	5	F,H,I	76,19

T	19	2	I,N	90,48
U	18	3	A	85,71
V	15	6	A,K,U	71,43
W	14	7	A,J,M,R	66,67
X	15	7	A,B,K,T,V,Y	68,18
Y	14	7	D,M,O,X	66,67
Z	11	10	G,M,O,T	52,38
<b>Akurasi</b>				<b>74,09</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

**Tabel 5.11 Hasil skenario 2 dengan parameter 2:end-2**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	18	3	I,T	85,71
B	16	5	D,N,V	76,19
C	13	6	G,K,N,R	68,42
D	11	10	I,N,S,Z	52,38
E	8	13	A,G,I,K,N,T,#	38,10
F	12	10	A,C,E,G,L,N,R,#	54,55
G	13	9	D,M,N,R,S,T	59,09
H	18	5	B,S,U	78,26
I	9	12	A,B,D,E,H,J,N,R,S	42,86
J	20	2	O	90,91
K	16	6	A,D,M,U,Y	72,73
L	11	10	A,D,N,P,R,S	52,38
M	19	2	T	90,48
N	18	4	G,R,S	81,82
O	19	2	M,W	90,48
P	11	10	C,F,G,H,I,J,R,#	52,38
Q	9	12	B,D,K,N,O,Z,#	42,86
R	9	12	A,K,L,N,P,S	42,86
S	14	7	B,D,H,I	66,67
T	16	5	A,I,K,M	76,19
U	10	11	A,K,T,W,#	47,62
V	9	12	A,F,T,U,#	42,86
W	15	6	M	71,43
X	10	12	D,K,M,T,V	45,45

Y	11	10	C,M,O,X	52,38
Z	11	10	G,N,O,T,U,W	52,38
<b>Akurasi</b>				<b>62,68</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

**Tabel 5.12 Hasil skenario 2 dengan parameter 3:end-1**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	19	2	T	90,48
B	13	8	D,G,K,Z	61,90
C	13	6	K,M,O,Y	68,42
D	11	10	B,G,N,S,Z	52,38
E	8	13	G,I,N,T,#	38,10
F	3	19	A,C,D,H,L,M,N,R, U,V,W,Y,#	13,64
G	3	19	D,M,N,P,U,Z	13,64
H	9	14	A,B,D,P	39,13
I	6	15	A,D,G,K,N,R,T	28,57
J	21	1	O	95,45
K	20	2	U,X	90,91
L	16	5	A,D,R,T	76,19
M	19	2	O,T	90,48
N	17	5	M,T	77,27
O	17	4	K,W,#	80,95
P	7	14	D,G,H,J,M,R,W,Z,#	33,33
Q	2	19	C,O,P,U,Z	9,52
R	6	15	A,F,I,L,N,S,T	28,57
S	7	14	B,F,I,N,O,R,T,X	33,33
T	16	5	E,I,M,#	76,19
U	2	19	A,I,S,T,W	9,52
V	4	17	A,H,M,N,S,T,U,W	19,05
W	9	12	G,I,M,N,P,R,T	42,86
X	2	20	J,K,M,Q,T	9,09
Y	4	17	C,D,H,K,R,#	19,05
Z	5	16	D,M,O,Q,T,W	23,81
<b>Akurasi</b>				<b>46,92</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

**Tabel 5.13 Hasil skenario 2 dengan parameter 3:end-2**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	18	3	E,I,T	85,71
B	10	11	D,N,T	47,62
C	11	8	G,K,M,O,W,Y	57,89
D	5	16	A,B,G,H,I,N,Q,T,Z	23,81
E	4	17	A,K,N,R,T,U,#	19,05
F	3	19	C,E,H,N,P,R,U,#	13,64
G	2	20	A,C,D,F,I,O,P,T	9,09
H	7	16	A,B,C,F,M,N,S,U,V,X	30,43
I	3	18	A,E,K,M,T,U	14,29
J	21	1	O	95,45
K	16	6	D,M,U	72,73
L	15	6	A,N,P,R,T,Z	71,43
M	18	3	A,N,O	85,71
N	13	9	A,M,S,T	59,09
O	15	6	K,M,W	71,43
P	3	18	C,D,G,H,J,T,U,W,Y,#	14,29
Q	1	20	B,K,O,W	4,76
R	7	14	A,E,F,L,M,N,S,T,U,W	33,33
S	7	14	A,B,D,E,H,I,K,M,U,X	33,33
T	10	11	A,E,I,M,N,U,X	47,62
U	2	19	A,I,M,N,T,X	9,52
V	2	19	A,K,M,R,S,T,W,X,Y,#	9,52
W	7	14	I,M,N,T,U	33,33
X	4	18	G,K,M,Q,T,W	18,18
Y	3	18	C,G,N,O,R,W,#	14,29
Z	5	16	A,G,N,S,T,U,W	23,81
<b>Akurasi</b>				<b>38,41</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

**Tabel 5.14 Hasil skenario 2 dengan parameter 4:end-1**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
--------------	--------------	--------------	--------------------------	----------------

A	10	11	D,I,N,T	47,62
B	2	19	A,E,N,T	9,52
C	0	19	A,B,K,M,O,T,U,W,Z	0,00
D	3	18	A,G,I,M,N,T,X	14,29
E	1	20	A,T	4,76
F	1	21	D,M,N,T	4,55
G	0	22	A,D,L,M,O,R,T,U,V	0,00
H	0	23	A,D,M,N,R,S,T,U,V,#	0,00
I	0	21	A,G,N,T,#	0,00
J	7	15	K,L,O,P,T,W	31,82
K	4	18	A,N,O,Q,T,U,Y	18,18
L	3	18	B,D,F,G,M,P,T,U	14,29
M	10	11	A,D,I,N,T	47,62
N	2	20	A,M,T	9,09
O	6	15	B,D,G,K,Q,U,W,X	28,57
P	1	20	A,G,M,N,O,R,T	4,76
Q	0	21	F,G,L,N,O,P,R,T, U,V,W	0,00
R	3	18	A,D,E,I,M,N,T	14,29
S	0	21	A,D,I,N,O,T,#	0,00
T	14	7	A,I,M,N,#	66,67
U	0	21	M,N,T	0,00
V	0	21	A,D,E,M,N,T,U	0,00
W	0	21	A,D,G,K,M,N,O	0,00
X	0	22	A,G,I,M,N,O,R,T,W,Z	0,00
Y	0	21	A,B,C,F,G,L,M,N,O, R,T,U,W,X	0,00
Z	2	19	A,D,G,M,N,R,T,U	9,52
<b>Akurasi</b>				<b>12,50</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

Tabel 5.15 Hasil skenario 2 dengan parameter 4:end-2

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
-------	-------	-------	-------------------	---------

A	8	13	I,M,N,T,U	38,10
B	1	20	A,D,G,H,N,T	4,76
C	0	19	A,D,J,K,M,O,R,T,U,V	0,00
D	3	18	A,B,G,M,N,T,Z	14,29
E	1	20	A,T,#	4,76
F	1	21	A,T,U,#	4,55
G	0	22	A,D,F,M,N,O,T,V,#	0,00
H	0	23	A,B,D,F,G,K,M,N,R, S,T,U	0,00
I	0	21	A,G,M,N,O,R,T,U,#	0,00
J	2	20	A,F,O,T,#	9,09
K	0	22	A,B,D,G,M,N,O,Q, S,T,U,V	0,00
L	1	20	A,D,T	4,76
M	9	12	A,D,G,N,T,X	42,86
N	3	19	E,T,U,#	13,64
O	6	15	D,G,K,M,N,Q,R,U,V,W	28,57
P	1	20	A,G,N,O,R,X	4,76
Q	0	21	A,G,M,N,O,R,T,U, W,Y,Z	0,00
R	3	18	A,D,M,T	14,29
S	0	21	A,D,E,G,K,M,N,T,U	0,00
T	9	12	A,I,M,O,U,V,W	42,86
U	2	19	A,E,G,K,M,R,T	9,52
V	0	21	A,D,E,K,M,N,T,#	0,00
W	0	21	A,M,O,R,T	0,00
X	0	22	A,D,K,M,N,T	0,00
Y	0	21	D,F,K,M,N,O,R,T,U,W	0,00
Z	2	19	D,F,M,T,X	9,52
<b>Akurasi</b>				<b>9,42</b>
<b>Waktu Pendeteksian</b>				<b>0,15</b>

Hasil yang ditunjukkan pada tabel 5.7 dan 5.8 menunjukkan tingkat *error* yang tinggi. Ini dikarenakan pengambilan elemen pertama tidak banyak menghilangkan *noise*, sehingga mengakibatkan pendeteksian ROI yang salah, bisa jadi garis dikenali sebagai titik atau titik dan garis tidak dikenali sama sekali.

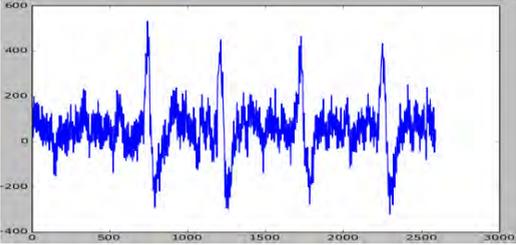
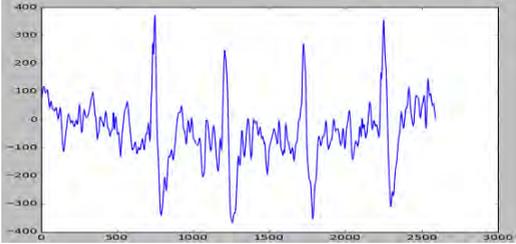
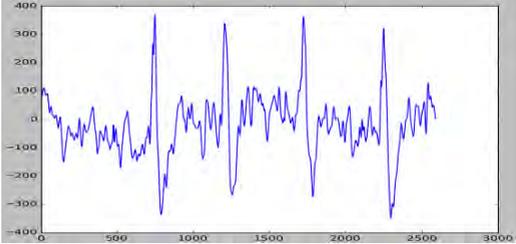
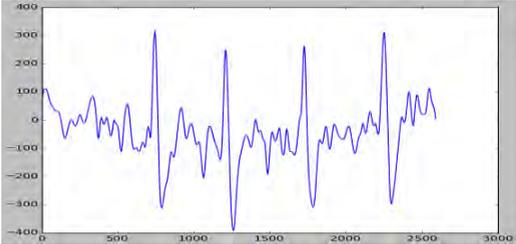
Hasil yang ditunjukkan pada tabel 5.11, 5.12, 5.13 dan 5.14 diakibatkan karena pengambilan IMF mulai dari indeks 3 dan 4 menjadikan sinyal hasil *preprocessing* menjadi sinyal gelombang *sinus* dan *cosinus* dengan frekuensi teratur. Banyak karakter terdeteksi berasal dari kelas A, karena interval gelombang menjadi lebih besar sehingga elemen yang terdeteksi hanya sedikit.

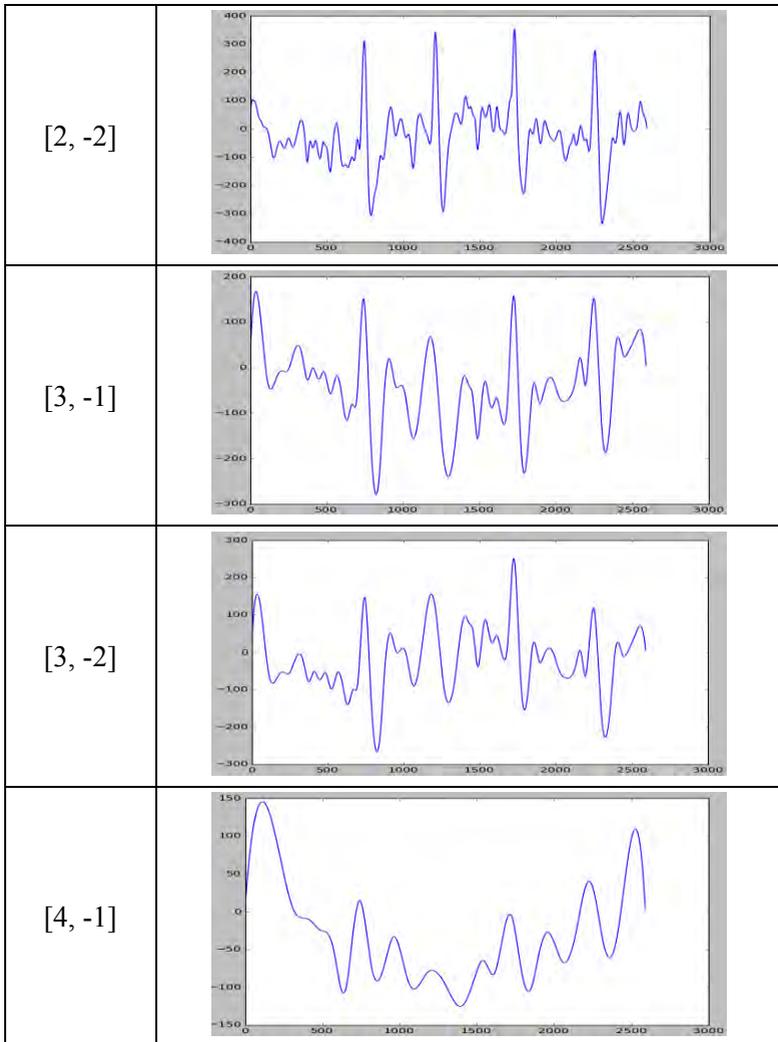
Berdasarkan hasil yang telah didapatkan pada uji coba skenario 2, didapat bahwa nilai akurasi tertinggi adalah 74,09% dengan elemen IMF ke 3 sampai sebelum terakhir (residu). Hal ini mengindikasikan bahwa elemen IMF ke 3 sampai terakhir sebelum residu (IMF terakhir) adalah elemen penyusun terbaik untuk rekonstruksi sinyal EOG dari EEG. Variasi parameter ini memberikan hasil yang signifikan jika dibanding dengan variasi parameter yang lain. Tabel 5.16 menunjukkan hasil rekapitulasi skenario 2 sedangkan tabel 5.17 menunjukkan contoh hasil pengolahan untuk masing-masing variasi indeks.

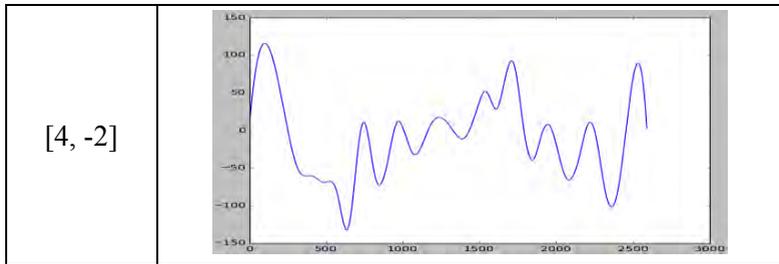
**Tabel 5.16 Rekapitulasi skenario 2**

IMF ke	Akurasi (%)	Rata-rata waktu pengenalan (det)
[1, -1]	46,74	0,15
[1, -2]	41,49	0,15
[2, -1]	74,09	0,15
[2, -2]	62,68	0,15
[3, -1]	46,92	0,15
[3, -2]	38,41	0,15
[4, -1]	12,50	0,15
[4, -2]	9,42	0,15

**Tabel 5.17 Contoh hasil pengolahan variasi indeks**

IMK ke	Hasil
Asli	 A line plot showing a highly oscillatory signal. The x-axis ranges from 0 to 3000 with major ticks every 500 units. The y-axis ranges from -400 to 600 with major ticks every 200 units. The signal fluctuates rapidly around a mean of zero, with several prominent peaks reaching approximately 500 and troughs reaching approximately -300.
[1, -1]	 A line plot showing a filtered version of the original signal. The axes are the same as the original plot. The signal is significantly smoother, with the high-frequency noise removed. The overall shape of the signal is preserved, showing a series of peaks and troughs that are much broader than in the original plot.
[1, -2]	 A line plot showing a filtered version of the original signal. The axes are the same as the original plot. The signal is smoother than the [1, -1] filter, with even fewer high-frequency components. The peaks and troughs are further smoothed and broadened.
[2, -1]	 A line plot showing a filtered version of the original signal. The axes are the same as the original plot. The signal is very smooth, with only the most prominent low-frequency components remaining. The high-frequency noise is almost entirely removed, leaving a clear, broad signal.





### 5.4.3 Skenario Uji Coba 3

Skenario uji coba 3 adalah penggunaan filter *bandpass* sebagai metode *preprocessing* sinyal. Metode ini akan diterapkan pada sinyal masukan dengan batasan frekuensi yang bervariasi. Selanjutnya sinyal akan mengalami pendeteksian pola seperti yang telah dijelaskan pada bagian Tahap Khusus Pengenalan Berdasarkan Aturan. Pada skenario ini, terdapat tujuh variasi parameter frekuensi yang diambil. Variasi nilai parameter ini bertujuan untuk mengetahui berapa rentang frekuensi yang mewakili representasi dari sinyal EOG yang akan diproses. Variasi parameter tersebut adalah (format [a; b] dengan a adalah frekuensi minimal dan b adalah frekuensi maksimal yang diambil) [0; 10], [0,5; 10], [0,6; 10], [1; 10], [0; 20], dan [5; 40]. Tabel 5.18, 5.19, 5.20, 5.21, 5.22, dan 5.23 masing-masing menunjukkan hasil uji coba untuk masing-masing variasi parameter.

**Tabel 5.18 Hasil uji coba skenario 3 parameter [0,0; 10]**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	21	0		100,00
C	19	0		100,00
D	17	4	B,H,Z	80,95
E	14	7	A,I,N,Z	66,67
F	18	4	E,N,R	81,82

G	17	5	K,N,R	77,27
H	23	0		100,00
I	14	7	B,D,S,Z	66,67
J	22	0		100,00
K	20	2	U,Y	90,91
L	20	1	A	95,24
M	21	0		100,00
N	22	0		100,00
O	21	0		100,00
P	18	3	F,G	85,71
Q	11	10	M,N,O,X,Z,#	52,38
R	20	1	#	95,24
S	20	1	H	95,24
T	20	1	I	95,24
U	19	2	A,#	90,48
V	19	2	U	90,48
W	19	2	J,M	90,48
X	20	2	Q,R	90,91
Y	20	1	X	95,24
Z	18	3	G,T	85,71
<b>Akurasi</b>				<b>89,49</b>
<b>Waktu Pendeteksian</b>				<b>0,03</b>

**Tabel 5.19 Hasil uji coba skenario 3 parameter [0,5; 10]**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	21	0		100,00
C	19	0		100,00
D	17	4	B,H,Z	80,95
E	14	7	A,I,N,Z	66,67
F	17	5	E,N,R,U	77,27
G	17	5	K,N,R	77,27
H	23	0		100,00
I	16	5	B,D,S,Z	76,19
J	22	0		100,00

K	21	1	Y	95,45
L	20	1	A	95,24
M	21	0		100,00
N	22	0		100,00
O	21	0		100,00
P	18	3	F,G	85,71
Q	10	11	K,N,O,X,Z,#	47,62
R	20	1	C	95,24
S	20	1	H	95,24
T	19	2	I,M	90,48
U	19	2	A,#	90,48
V	15	6	F,U,#	71,43
W	19	2	J,M	90,48
X	20	2	Q	90,91
Y	20	1	X	95,24
Z	19	2	G	90,48
<b>Akurasi</b>				<b>88,95</b>
<b>Waktu Pendeteksian</b>				<b>0,03</b>

**Tabel 5.20 Hasil uji coba skenario 3 parameter [0,6; 10]**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	21	0		100,00
B	20	1	D	95,24
C	18	1	D	94,74
D	16	5	B,H,Z	76,19
E	15	6	A,I,N,Z	71,43
F	18	4	E,N,R	81,82
G	17	5	K,N,R	77,27
H	23	0		100,00
I	15	6	B,D,S,Z	71,43
J	22	0		100,00
K	21	1	Y	95,45
L	20	1	A	95,24
M	21	0		100,00
N	22	0		100,00

O	20	1	Q	95,24
P	18	3	F,G	85,71
Q	10	11	K,N,O,X,Z,#	47,62
R	20	1	C	95,24
S	20	1	H	95,24
T	18	3	I,M	85,71
U	18	3	A,#	85,71
V	15	6	F,U,#	71,43
W	19	2	J,M	90,48
X	20	2	Q	90,91
Y	20	1	X	95,24
Z	19	2	G,U	90,48
<b>Akurasi</b>				<b>88,04</b>
<b>Waktu Pendeteksian</b>				<b>0,03</b>

**Tabel 5.21 Hasil uji coba skenario 3 parameter [1,0; 10]**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	19	2	W	90,48
B	19	2	D,V	90,48
C	18	1	F	94,74
D	14	7	B,H,L,Z	66,67
E	16	5	A,I,N,S,Z	76,19
F	15	7	E,G,H,R,V	68,18
G	10	12	F,H,K,P,Q,R,Z,#	45,45
H	22	1	B	95,65
I	14	7	B,D,R,S,U,Z	66,67
J	17	5	A,F,L,W	77,27
K	16	6	C,Y	72,73
L	20	1	A	95,24
M	12	9	K,O,T,#	57,14
N	21	1	G	95,45
O	6	15	C,K,M,Q,W,#	28,57
P	15	6	F,L,U	71,43
Q	10	11	B,R,V,X,Z,#	47,62
R	15	6	A,F,H,L,W	71,43
S	20	1	F	95,24

T	15	6	A,I,M,N	71,43
U	18	3	V,#	85,71
V	18	3	F,U	85,71
W	10	11	J,K,U,V	47,62
X	18	4	F,Q	81,82
Y	12	9	C,K,Q,U,X,#	57,14
Z	14	7	B,C,#	66,67
<b>Akurasi</b>				<b>73,19</b>
<b>Waktu Pendeteksian</b>				<b>0,03</b>

**Tabel 5.22 Hasil uji coba skenario 3 parameter [0,0; 20]**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	20	1	T	95,24
B	20	1	D	95,24
C	16	3	K,Y	84,21
D	16	5	B,H,L,N	76,19
E	9	12	A,D,F,H,I,K,N	42,86
F	12	10	C,E,N,R,U	54,55
G	13	9	M,N,P,Q,R,S	59,09
H	20	3	B,S	86,96
I	10	11	A,D,H,R,S	47,62
J	22	0		100,00
K	21	1	B	95,45
L	17	4	R	80,95
M	21	0		100,00
N	20	2	T	90,91
O	19	2	M,W	90,48
P	16	5	B,F,G,R,W	76,19
Q	13	8	G,K,N,O,X,#	61,90
R	17	4	A,C,N,W	80,95
S	18	3	H	85,71
T	18	3	A,M,N	85,71
U	19	2	T,#	90,48
V	17	4	A,B,U	80,95
W	19	2	M	90,48

X	18	4	K,T,V,Y	81,82
Y	16	5	K,O,W,X	76,19
Z	16	5	D,G,T	76,19
<b>Akurasi</b>				<b>80,25</b>
<b>Waktu Pendeteksian</b>				<b>0,03</b>

**Tabel 5.23 Hasil uji coba skenario 3 parameter [5,0; 20]**

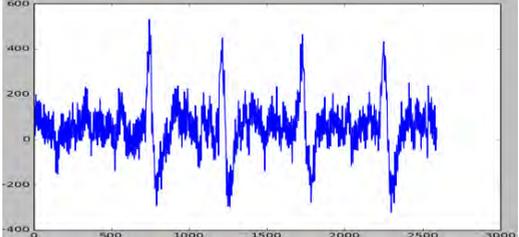
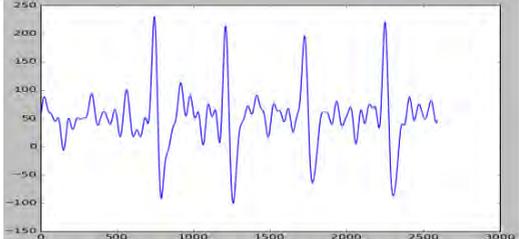
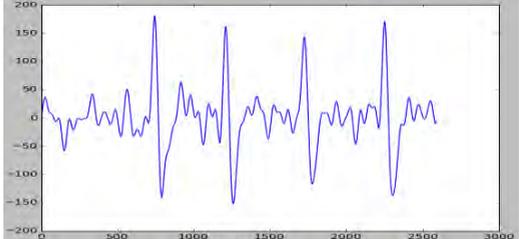
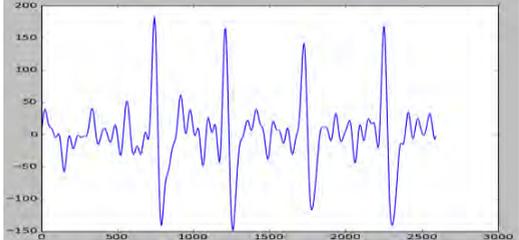
Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	1	20	F,H,L,S,V,W	4,76
B	3	18	C,F,G,H,L,R,V,#	14,29
C	3	16	L,P,Q,V,Z,#	15,79
D	0	21	B,C,G,L,P,U,V,X,Z,#	0,00
E	0	21	B,D,F,I,L,R,S,U,V,X	0,00
F	4	18	B,C,H,L,P,S,Z,#	18,18
G	0	22	B,C,F,H,L,P,V,X,Z,#	0,00
H	0	23	B,C,F,L,P,Y,Z,#	0,00
I	0	21	B,C,F,L,R,T,Z	0,00
J	0	22	F,H,L,P,R,V,Y,Z,#	0,00
K	1	21	B,C,F,L,S,X,Z,#	4,55
L	3	18	B,C,Z,#	14,29
M	0	21	B,D,F,H,L,P,S,V,Z,#	0,00
N	1	21	F,H,L,R	4,55
O	0	21	C,D,F,H,L,P,Y,Z	0,00
P	4	17	B,C,F,Z,#	19,05
Q	0	21	C,D,F,J,L,P,X,Y,Z,#	0,00
R	1	20	B,C,F,Z,#	4,76
S	2	19	B,F,G,L,P,V,Z	9,52
T	0	21	A,B,I,L,N,R,S,U,V,W	0,00
U	0	21	B,C,F,H,L,P,V,Z,#	0,00
V	0	21	A,C,F,H,J,P,Q,X,Y,#	0,00
W	0	21	B,C,F,P,Q,U,V,Z,#	0,00
X	0	22	B,C,F,L,U,#	0,00
Y	2	19	C,F,L,P,V,Z,#	9,52
Z	4	17	C,D,L,#	19,05
<b>Akurasi</b>				<b>5,25</b>
<b>Waktu Pendeteksian</b>				<b>0,03</b>

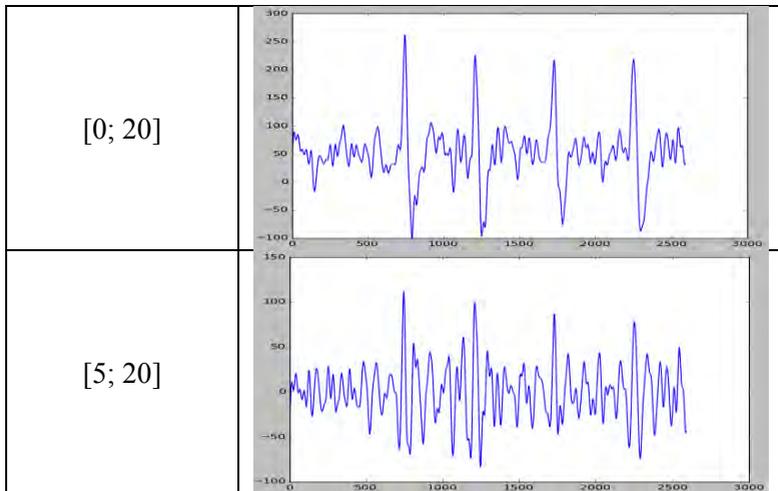
Berdasarkan hasil yang telah didapatkan dari uji coba skenario 3, didapat bahwa nilai akurasi tertinggi adalah 89,49% dengan batasan frekuensi yang diambil adalah [0,10]. Pada variasi satu, dua, dan tiga didapatkan hasil yang tidak signifikan. Ini menunjukkan bahwa pengurangan rentang seperti demikian tidak terlalu berpengaruh terhadap performa sistem. Dari percobaan ini, diketahui bahwa frekuensi dari sinyal EOG terdistribusi pada rentang [0, 10]. Hasil rata-rata akurasi pada tabel 5.16, 5.17, 5.18, 5.19 dan 5.20 menunjukkan bahwa pengambilan frekuensi pada bagian awal sangat berpengaruh terhadap rata-rata akurasi sistem. Frekuensi *baseline* (garis dasar) sinyal terkonsentrasi pada bagian awal domain frekuensi. Sedangkan kesalahan-kesalahan klasifikasi yang lain disebabkan karena salah pendeteksian titik dan garis. Data-data pada kelas E banyak mengandung *noise* mayor, sehingga seringkali terdeteksi puncak-puncak yang bukan merupakan bagian dari ROI. Kelas Q sering dikenali sebagai kelas K, M, N, O, X dikarenakan terdapat ROI yang tidak terdeteksi karena puncak terlalu rendah atau ROI yang terpilih tidak memenuhi kondisi titik atau garis. Pada tabel 5.24 disajikan hasil rekapitulasi skenario 3 sedangkan tabel 5.25 menunjukkan contoh hasil pengolahan untuk masing-masing variasi rentang frekuensi.

**Tabel 5.24 Rekapitulasi skenario 3**

<b>Batas Frekuensi</b>	<b>Akurasi (%)</b>	<b>Rata-rata waktu pengenalan (det)</b>
[0; 10]	89,49	0,03
[0,5; 10]	88,95	0,03
[0,6; 10]	88,04	0,03
[1; 10]	73,19	0,03
[0; 20]	80,25	0,03
[5; 20]	5,25	0,03

Tabel 5.25 Contoh hasil pengolahan variasi frekuensi

Batas Frekuensi	Hasil
Asli	 <p>The original signal plot shows a complex waveform with a peak amplitude of approximately 500 and a duration of 3000 units. The y-axis ranges from -400 to 600, and the x-axis ranges from 0 to 3000.</p>
[0; 10]	 <p>The filtered signal plot shows a waveform with a peak amplitude of approximately 250 and a duration of 3000 units. The y-axis ranges from -150 to 250, and the x-axis ranges from 0 to 3000.</p>
[0,5; 10]	 <p>The filtered signal plot shows a waveform with a peak amplitude of approximately 150 and a duration of 3000 units. The y-axis ranges from -200 to 200, and the x-axis ranges from 0 to 3000.</p>
[1; 10]	 <p>The filtered signal plot shows a waveform with a peak amplitude of approximately 150 and a duration of 3000 units. The y-axis ranges from -150 to 200, and the x-axis ranges from 0 to 3000.</p>



#### 5.4.4 Skenario Uji Coba 4

Skenario uji coba 4 merupakan skenario berbasis pembelajaran (*learning based*) menggunakan kNN. Pada skenario ini, semua data sinyal akan distandardisasi ukurannya. Standardisasi ukuran memanfaatkan pemotongan frekuensi sinyal yang telah ditransformasi ke dalam domain frekuensi. Frekuensi sinyal akan dipotong pada posisi dimana panjang sinyal terpendek terdapat pada semua data. Sinyal baru didapatkan dari invers sinyal barisan angka dalam domain frekuensi yang telah dipotong.

Lima *Fold Cross-validation* akan digunakan untuk membangkitkan himpunan uji dan latih. Pada setiap himpunan latih akan dibangkitkan atau tidak dibangkitkan himpunan *template*. *Template* inilah yang akan menjadi himpunan latih baru bagi data masukan. Pembangkitan *template* menggunakan metode *K-Means* dan *hierarchical clustering* dengan jumlah *template* bervariasi pada setiap eksekusi program. Pengukuran jarak yang digunakan dalam pembangkitan *template* adalah *Dynamic Time Warping*.

Pembedaan metode pembangkitan *template* bertujuan untuk mengetahui metode mana yang memiliki performa dan keefektifan yang lebih baik. Untuk masing-masing metode pembangkitan *template* divariasikan jumlah *template* yang dibangkitkan, yaitu tanpa pembangkitan *template* (menggunakan *training set* asli), satu, lima, dan lima belas *template*. Hal ini bertujuan untuk mengetahui pengaruh jumlah *template* dengan akurasi sistem pengenalan. Tabel 5.26, 5.27, 5.28 dan 5.29 secara terurut menunjukkan hasil dari masing-masing variasi *template* dengan metode pembangkitan *K-Means*.

**Tabel 5.26 Hasil dengan himpunan latihan asli**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	14	7	E,T	66,67
B	5	16	A,C,D,E,K,S	23,81
C	7	12	A,B,D,E,S,T	36,84
D	13	8	E,S	61,90
E	21	0		100,00
F	2	20	A,C,D,P,Z	9,09
G	3	19	A,D,E,I,M,R,S,T	13,64
H	10	13	D,E,S	43,48
I	5	16	E,T	23,81
J	8	14	C,D,H,P,S,T	36,36
K	6	16	B,D,E,S	27,27
L	6	15	A,B,D,E,N,S,T	28,57
M	14	7	A,D,E	66,67
N	16	6	A,E,I	72,73
O	11	10	C,D,K,L,R,U,Z	52,38
P	3	18	D,E,H,I,S,V	14,29
Q	12	9	A,C,D,H,S	57,14
R	7	14	A,E,S,T	33,33
S	3	18	A,E,T	14,29
T	10	11	E	47,62
U	7	14	A,E,M,T	33,33
V	8	13	B,C,D,K,S,U	38,10
W	17	4	E,I,T,U	80,95

X	7	15	A,B,C,D,H,K,Z	31,82
Y	10	11	A,B,C,E,K,U,V	47,62
Z	6	15	A,D,O,P,Q,S,V,X	28,57
<b>Akurasi</b>				<b>41,85</b>
<b>Waktu Deteksi</b>				<b>0,5</b>
<b>Waktu Pembangkitan Model</b>				-

**Tabel 5.27 Hasil dengan jumlah *template* 1**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	0	21	E,I	0,00
B	0	21	E,I,R	0,00
C	0	19	E,I,R	0,00
D	0	21	E,I	0,00
E	17	4	I	80,95
F	0	22	E,I,T	0,00
G	0	22	E,I,S	0,00
H	0	23	E,I	0,00
I	12	9	E,R,S	57,14
J	0	22	E,I,R	0,00
K	0	22	E,I,R	0,00
L	0	21	E,I,R,T	0,00
M	0	21	E,I,U	0,00
N	1	21	E,I,R	4,55
O	0	21	E,I,R	0,00
P	0	21	E,I,R,S	0,00
Q	0	21	E,I,T	0,00
R	1	20	E,I	4,76
S	1	20	E,I	4,76
T	0	21	E,I,R	0,00
U	0	21	E,I	0,00
V	0	21	E,I	0,00
W	1	20	E,I	4,76
X	0	22	E,I,R,S	0,00
Y	0	21	E,I,R	0,00
Z	0	21	E,I,R	0,00

<b>Akurasi</b>	<b>5,98</b>
<b>Waktu Deteksi</b>	<b>0,04</b>
<b>Waktu Pembangkitan Model</b>	<b>30,2</b>

**Tabel 5.28 Hasil dengan jumlah *template* 5**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	5	16	E,T	23,81
B	1	20	E,T	4,76
C	1	18	A,E,T	5,26
D	0	21	A,E,T	0,00
E	20	1	T	95,24
F	0	22	A,D,E,H,K,T	0,00
G	3	19	A,E,H,S,T,U	13,64
H	0	23	E,T,U	0,00
I	0	21	E,R,T	0,00
J	1	21	D,E,H,M,T	4,55
K	0	22	A,E,N,T	0,00
L	4	17	A,E,I,T	19,05
M	6	15	A,E,T,U,W	28,57
N	8	14	A,E,T	36,36
O	5	16	A,D,E,K,R,S, T,U,W	23,81
P	0	21	E,H,O,T,V,W	0,00
Q	1	20	E,T	4,76
R	3	18	A,E,N	14,29
S	0	21	A,E,T	0,00
T	8	13	A,E	38,10
U	4	17	E,G,M,T,W	19,05
V	1	20	E,K,T,U	4,76
W	14	7	A,E,N,T	66,67
X	0	22	A,D,E,G,K,R,T,U,V	0,00
Y	9	12	A,C,E,H,T,W,X	42,86
Z	1	20	E,T,U,X	4,76
<b>Akurasi</b>				<b>17,21</b>
<b>Waktu Deteksi</b>				<b>0,18</b>
<b>Waktu Pembangkitan Model</b>				<b>16,15</b>

Tabel 5.29 Hasil dengan jumlah *template* 15

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	14	7	E,T	66,67
B	5	16	A,K,R,S,U	23,81
C	1	18	E,S,T	5,26
D	8	13	N,S,T	38,10
E	21	0		100,00
F	1	21	A,C,D,U	4,55
G	5	17	A,D,E,I,M,R,S	22,73
H	10	13	E,S,T	43,48
I	6	15	E,T	28,57
J	6	16	B,C,D,H,K,P,V	27,27
K	6	16	B,D,E,M,S	27,27
L	6	15	A,D,E,G,I,T	28,57
M	15	6	A,D,T	71,43
N	19	3	A,E,I	86,36
O	12	9	A,C,D,E,K,L,U	57,14
P	3	18	A,D,E,I,N,S,U,V	14,29
Q	8	13	A,C,D,E,K,N,S,U,X	38,10
R	6	15	A,E,S,T	28,57
S	5	16	A,E,T	23,81
T	9	12	E	42,86
U	7	14	A,E,H,M,R,S,T	33,33
V	8	13	B,C,M,P,S,U	38,10
W	18	3	E,S,T	85,71
X	9	13	A,B,C,D,K,R,U	40,91
Y	9	12	A,C,D,H,K,P,U,V	42,86
Z	5	16	A,B,C,D,E,K,O,P,Q, S,U,V,X	23,81
<b>Akurasi</b>				<b>40,22</b>
<b>Waktu Deteksi</b>				<b>0,51</b>
<b>Waktu Pembangunan Model</b>				<b>4,28</b>

Tabel 5.30, 5.31 dan 5.32 masing-masing adalah hasil dari variasi *template* dengan metode pembangunan *Hierarchical clustering*.

**Tabel 5.30 Hasil dengan jumlah *template* 1**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	0	21	E,R	0,00
B	0	21	E,R	0,00
C	0	19	E,R	0,00
D	0	21	E	0,00
E	20	1	R	95,24
F	0	22	E,T	0,00
G	0	22	E	0,00
H	0	23	E	0,00
I	0	21	E,R	0,00
J	0	22	E	0,00
K	0	22	E,R	0,00
L	0	21	E,R,T	0,00
M	0	21	E,R	0,00
N	0	22	E,R	0,00
O	0	21	E,R,T	0,00
P	0	21	E,R	0,00
Q	0	21	E,U	0,00
R	2	19	E	9,52
S	0	21	E,R	0,00
T	4	17	E,R	19,05
U	1	20	E	4,76
V	0	21	E	0,00
W	0	21	E,R	0,00
X	0	22	E,R	0,00
Y	0	21	E,R	0,00
Z	0	21	E,R	0,00
<b>Akurasi</b>				<b>4,89</b>
<b>Waktu Deteksi</b>				<b>0,04</b>
<b>Waktu Pembangkitan Model</b>				<b>1011,64</b>

**Tabel 5.31 Hasil dengan jumlah *template* 5**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	3	18	E	14,29
B	0	21	A,E,T	0,00

C	0	19	A,E,R,T	0,00
D	0	21	E,I,T	0,00
E	21	0		100,00
F	0	22	A,E,T,U	0,00
G	3	19	A,E,M	13,64
H	4	19	E,T	17,39
I	1	20	E,T	4,76
J	0	22	A,B,C,D,E,H ,K,N,S,T,U	0,00
K	0	22	D,E,H	0,00
L	3	18	E,T,U	14,29
M	8	13	A,E,T	38,10
N	4	18	A,E,I,T	18,18
O	6	15	A,B,D,E,K,N,T,U	28,57
P	0	21	A,E,H,S,T,U	0,00
Q	0	21	A,C,E,T,U	0,00
R	0	21	E,T	0,00
S	1	20	E,T	4,76
T	10	11	E	47,62
U	5	16	A,E,T	23,81
V	5	16	A,D,E,K,U	23,81
W	12	9	A,E,T	57,14
X	1	21	A,B,E,T	4,55
Y	8	13	A,E,H,T,W	38,10
Z	0	21	A,B,C,E,I,U,X	0,00
<b>Akurasi</b>				<b>17,21</b>
<b>Waktu Deteksi</b>				<b>0,18</b>
<b>Waktu Pembangkitan Model</b>				<b>910,6</b>

Tabel 5.32 Hasil dengan jumlah *template* 15

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	14	7	E,N,T	66,67
B	4	17	A,E,K	19,05
C	1	18	A,E,S,T	5,26
D	11	10	A,E,M,S	52,38
E	21	0		100,00

F	2	20	A,D,E,O,R,T,U	9,09
G	3	19	A,D,E,I,K,M,R,S,U	13,64
H	12	11	E,I,S	52,17
I	4	17	E,T	19,05
J	6	16	B,C,D,H,K,P,S,U	27,27
K	7	15	B,D,H,I,N,S,U	31,82
L	6	15	A,D,E,T,U	28,57
M	16	5	A,D,N	76,19
N	18	4	A,E,I	81,82
O	12	9	A,C,D,K,L,R	57,14
P	2	19	B,D,G,R,S,T,U,X	9,52
Q	8	13	A,B,D,E,R,U,W,X	38,10
R	8	13	A,E,S,T	38,10
S	5	16	A,E,G,I,R,T	23,81
T	10	11	I	47,62
U	7	14	A,C,E,M,T	33,33
V	6	15	C,D,H,K,M,R,S,U	28,57
W	17	4	E,I,S,T	80,95
X	7	15	A,B,C,D,H,K,R,S,Z	31,82
Y	11	10	C,E,H,K,R,S,V	52,38
Z	5	16	A,B,C,D,E,H,K,O, P,Q,S,V,X	23,81
<b>Akurasi</b>				<b>40,40</b>
<b>Waktu Deteksi</b>				<b>0,5</b>
<b>Waktu Pembangkitan Model</b>				<b>257,76</b>

Pada skenario 4, seringkali data teridentifikasi sebagai kelas E. Penyebabnya adalah panjang data kelas E adalah yang paling pendek. Sebelum skenario 4 dijalankan, semua panjang data distandardisasi menjadi panjang minimal dari semua sinyal. Pola sinyal pada data di kelas E hanya satu puncak dan satu lembah maksimal sehingga selalu menghasilkan pengukuran jarak yang kecil jika dibandingkan dengan data lainnya. Demikian juga untuk kelas I dan R yang memiliki panjang sinyal yang pendek. Tabel 5.33 menunjukkan rekapitulasi masing-masing variasi parameter pada skenario 4.

**Tabel 5.33** Tabel rekapitulasi skenario 4

Jumlah Template	<i>K-Means</i>			<i>Hierarchical Clustering</i>		
	Rata-rata Akurasi (%)	Waktu Pengenalan (detik)	Pembangkitan Model (detik)	Rata-rata Akurasi (%)	Waktu Pengenalan (detik)	Pembangkitan Model (detik)
Asli	41,85	0,5	-	41,85	0,5	-
1	5,98	0,04	30,2	4,89	0,04	1011,64
5	17,21	0,18	16,15	17,21	0,18	910,6
15	40,22	0,51	4,28	40,4	0,5	257,76

#### 5.4.5 Skenario Uji Coba 5

Skenario uji coba 5 memiliki kesamaan dengan skenario 4. Perbedaannya terletak pada pengukuran jarak yang digunakan pada *classifier* kNN. Pada skenario 4 digunakan *Eucliden Distance*, sedangkan pada skenario 5 digunakan *Dynamic Time Warping*. Pada skenario ini, dibandingkan dua metode pembangkitan *template* seperti pada skenario uji coba 4. Tabel 5.34, 5.35, 5.36 dan 5.37 adalah hasil uji coba untuk metode pembangkitan *K-Means*.

**Tabel 5.34** Hasil dengan himpunan latih asli

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	21	0		100,00
C	17	2	G,P	89,47
D	20	1	B	95,24
E	21	0		100,00
F	20	2	C,Z	90,91
G	20	2	H,P	90,91
H	23	0		100,00
I	19	2	A,S	90,48
J	22	0		100,00
K	22	0		100,00

L	21	0		100,00
M	21	0		100,00
N	22	0		100,00
O	21	0		100,00
P	18	3	F,Q	85,71
Q	20	1	Z	95,24
R	21	0		100,00
S	18	3	I,L,U	85,71
T	21	0		100,00
U	21	0		100,00
V	21	0		100,00
W	21	0		100,00
X	22	0		100,00
Y	21	0		100,00
Z	18	3	Q	85,71
<b>Akurasi</b>				<b>96,56</b>
<b>Waktu Pendeteksian</b>				<b>10,1</b>
<b>Waktu Pembangkitan Model</b>				<b>-</b>

Tabel 5.35 Hasil dengan jumlah *template* 1

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	21	0		100,00
C	9	10	A,B,K,Q,U	47,37
D	0	21	A,E,N,S	0,00
E	21	0		100,00
F	5	17	G,H,P,Q,U,V,Y	22,73
G	3	19	R,S,U,W	13,64
H	2	21	S,U	8,70
I	4	17	A,E,G,Q,T,W	19,05
J	14	8	L,W	63,64
K	18	4	M,S	81,82
L	17	4	B,W	80,95
M	16	5	A	76,19
N	8	14	A	36,36

O	9	12	K,W	42,86
P	1	20	F,U,W,Y	4,76
Q	21	0		100,00
R	4	17	A,E,G,S,W	19,05
S	11	10	A,E,I,L,R,U	52,38
T	10	11	A	47,62
U	17	4	A,S	80,95
V	13	8	U,W	61,90
W	21	0		100,00
X	4	18	B,K,U,V	18,18
Y	21	0		100,00
Z	1	20	B,Q,Y	4,76
<b>Akurasi</b>				<b>52,90</b>
<b>Waktu Pendeteksian</b>				<b>2,65</b>
<b>Waktu Pembangkitan Model</b>				<b>32,29</b>

Tabel 5.36 Hasil dengan jumlah *template* 5

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	18	3	D	85,71
C	10	9	B,G,K,M,Q	52,63
D	20	1	M	95,24
E	21	0		100,00
F	5	17	U,V,Z	22,73
G	14	8	C,H,S,W	63,64
H	18	5	B,D,U	78,26
I	5	16	A,E,H,T	23,81
J	20	2	C,L	90,91
K	14	8	C,D,M	63,64
L	21	0		100,00
M	17	4	A	80,95
N	19	3	A	86,36
O	17	4	C,J,K	80,95
P	8	13	C,H,J,L,W,Z	38,10
Q	19	2	Z	90,48

R	8	13	A,D,E,S,T,W	38,10
S	4	17	A,E,H,I,L,T,U	19,05
T	14	7	A,E	66,67
U	19	2	A	90,48
V	20	1	U	95,24
W	21	0		100,00
X	17	5	J,O,U,V	77,27
Y	20	1	O	95,24
Z	9	12	C,G,J,L,O,Q,Y	42,86
<b>Akurasi</b>				<b>72,28</b>
<b>Waktu Pendeteksian</b>				<b>6,83</b>
<b>Waktu Pembangkitan Model</b>				<b>17,31</b>

**Tabel 5.37 Hasil dengan jumlah *template* 15**

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	21	0		100,00
C	17	2	G,P	89,47
D	21	0		100,00
E	21	0		100,00
F	19	3	C,G,Z	86,36
G	19	3	P,S,W	86,36
H	23	0		100,00
I	20	1	A	95,24
J	22	0		100,00
K	22	0		100,00
L	21	0		100,00
M	21	0		100,00
N	22	0		100,00
O	21	0		100,00
P	16	5	F,Q,W	76,19
Q	19	2	Z	90,48
R	20	1	S	95,24
S	18	3	I,L,U	85,71
T	21	0		100,00

U	21	0		100,00
V	21	0		100,00
W	21	0		100,00
X	22	0		100,00
Y	21	0		100,00
Z	15	6	B,G,L,Q	71,43
<b>Akurasi</b>				<b>95,29</b>
<b>Waktu Pendeteksian</b>				<b>11,94</b>
<b>Waktu Pembangunan Model</b>				<b>4,21</b>

Tabel 5.38, 5.39 dan 5.40 adalah hasil dari masing-masing variasi *template* dengan metode pembangunan *Hierarchical clustering*.

**Tabel 5.38 Hasil dengan jumlah *template* 1**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	21	0		100,00
B	20	1	D	95,24
C	0	19	A,B,D,M,Q,U	0,00
D	16	5	A	76,19
E	21	0		100,00
F	5	17	G,P,Q,S,U,V,W	22,73
G	6	16	R,U,W	27,27
H	12	11	D,R,U	52,17
I	0	21	A,E,G,Q,R,T,W	0,00
J	13	9	L,R,U	59,09
K	10	12	A,B,D,M	45,45
L	20	1	D	95,24
M	14	7	A	66,67
N	11	11	A,E,T	50,00
O	14	7	B,K,U	66,67
P	4	17	B,G,H,Q,R,U,V,Y	19,05
Q	21	0		100,00
R	6	15	A,G,N	28,57
S	2	19	A,E,N,R,U	9,52
T	17	4	A,E	80,95
U	16	5	A,R,S	76,19
V	11	10	U,W	52,38

W	17	4	M,R,U	80,95
X	9	13	B,J,S,U,V,Y	40,91
Y	21	0		100,00
Z	4	17	B,L,P,Q,R,W,Y	19,05
<b>Akurasi</b>				<b>56,34</b>
<b>Waktu Pendeteksian</b>				<b>1,87</b>
<b>Waktu Pembangunan Model</b>				<b>619,7</b>

Tabel 5.39 Hasil dengan jumlah *template* 5

Kelas	Benar	Salah	Klasifikasi Salah	Akurasi
A	21	0		100,00
B	20	1	D	95,24
C	4	15	B,D,G,M,P	21,05
D	17	4	B,C	80,95
E	21	0		100,00
F	1	21	C,V	4,55
G	12	10	D,R,U,W	54,55
H	13	10	D,S,U	56,52
I	6	15	A,E,N,T,U	28,57
J	20	2	L,U	90,91
K	14	8	B,C,D,M	63,64
L	21	0		100,00
M	19	2	A	90,48
N	20	2	A,M	90,91
O	20	1	W	95,24
P	4	17	C,D,H,R,U,W,Y,Z	19,05
Q	19	2	Z	90,48
R	10	11	A,C,N,T	47,62
S	5	16	A,D,E,U	23,81
T	19	2	E	90,48
U	20	1	A	95,24
V	13	8	U	61,90
W	21	0		100,00
X	17	5	M,O,U	77,27
Y	21	0		100,00
Z	7	14	B,C,D,L,M,O,Q,V,W	33,33

<b>Akurasi</b>	<b>69,75</b>
<b>Waktu Pendeteksian</b>	<b>4,3</b>
<b>Waktu Pembangkitan Model</b>	<b>577,31</b>

**Tabel 5.40 Hasil dengan jumlah *template* 15**

<b>Kelas</b>	<b>Benar</b>	<b>Salah</b>	<b>Klasifikasi Salah</b>	<b>Akurasi</b>
A	21	0		100,00
B	21	0		100,00
C	17	2	G,P	89,47
D	20	1	B	95,24
E	21	0		100,00
F	19	3	C,G,Z	86,36
G	20	2	H,W	90,91
H	23	0		100,00
I	20	1	A	95,24
J	22	0		100,00
K	22	0		100,00
L	21	0		100,00
M	21	0		100,00
N	22	0		100,00
O	21	0		100,00
P	15	6	F,J,Q,W	71,43
Q	19	2	Z	90,48
R	18	3	N,S	85,71
S	18	3	I,L,U	85,71
T	21	0		100,00
U	21	0		100,00
V	21	0		100,00
W	21	0		100,00
X	22	0		100,00
Y	21	0		100,00
Z	15	6	B,G,O,Q	71,43
<b>Akurasi</b>				<b>94,75</b>
<b>Waktu Pendeteksian</b>				<b>8,61</b>
<b>Waktu Pembangkitan Model</b>				<b>181,49</b>

Tabel 5.41 menunjukkan rekapitulasi skenario 5 untuk masing-masing variasi parameter.

**Tabel 5.41 Rekapitulasi skenario 5**

Jumlah <i>Template</i>	<i>K-Means</i>			<i>Hierarchical Clustering</i>		
	Rata-rata Akurasi (%)	Waktu Pengenalan (detik)	Pembangkitan Model (detik)	Rata-rata Akurasi (%)	Waktu Pengenalan (detik)	Pembangkitan Model (detik)
Asli	96,56	10,1	-	96,56	10,1	-
1	52,9	2,65	32,29	56,34	1,87	619,7
5	72,28	6,83	17,31	69,75	4,3	577,31
15	95,29	11,94	4,21	94,75	8,6	181,49

Semakin jumlah data latih, semakin bagus rata-rata akurasi yang dihasilkan. Hal ini karena setiap kelas mempunyai variasi perekaman, bisa diakibatkan oleh interval dalam kedipan atau interval antar kedipan. DTW berhasil mengatasi pergeseran peristiwa penting (puncak dan lembah) dalam sinyal. Kesalahan banyak terjadi pada kelas yang mempunyai jumlah elemen yang sama, tidak seperti pada sistem pendeteksi berbasis aturan yang sering disebabkan karena penghilangan elemen tertentu di bagian depan dan belakang. Contoh, data di kelas P (. \_ \_ .) sering dideteksi sebagai F (. . \_ .).

Pada skenario uji coba 4 dan 5, terdapat dua pembahasan. Pembahasan pertama yaitu tentang jenis metode yang dipakai untuk membangkitkan *template*, sedangkan pembahasan kedua yaitu pengaruh DTW terhadap performa klasifikasi dengan kNN. Waktu minimal dan maksimal yang diperlukan untuk membangkitkan *template* dari metode *K-Means* adalah 4,21 detik dan 32,29 detik, sedangkan untuk metode *hierarchical clustering* adalah 181,49 detik dan 1011,64 detik. Terlihat jelas bahwa algoritma *K-Means* memiliki keunggulan dalam waktu pembentukan *template*, namun nilai rata-rata akurasi yang dihasilkan tidak terlalu signifikan. Penggunaan DTW untuk

menyusun ulang sinyal sangat berpengaruh signifikan terhadap performa klasifikasi. Sehingga kesimpulan yang didapatkan dari skenario 4 dan 5 adalah *K-Means* sebagai metode pembangkit *template* dengan jumlah *template* yang dibangkitkan 15 dan pengukuran jarak menggunakan algoritma DTW. Pembangkitan *template* pada sistem mempercepat proses pengenalan, namun hal ini diimbangi dengan penurunan akurasi rata-rata.

## 5.5 Evaluasi

Data yang telah direkam bervariasi untuk masing-masing kelas, sehingga semakin banyak himpunan data latihnya akan semakin memperbaiki rata-rata akurasi sistem. Kesalahan klasifikasi sering terjadi karena puncak yang terdeteksi pada data bukan puncak yang dihasilkan dari kedipan. Data yang direkam dengan perangkat Neurosky rentan mengalami gangguan dari lingkungan luar. Ketahanan penggunaan perangkat ini juga kurang bagus, bahkan kalau perlu setiap 15 menit sekali, penggunaan perangkat harus dihentikan.

Dari skenario uji coba yang telah dilakukan dengan menggunakan jenis filter yang berbeda maupun pengenalan berbasis pembelajaran memberikan pengaruh terhadap hasil pengenalan. Parameter yang digunakan untuk masing-masing skenario adalah lebar *window* sebesar 30, indeks *intrinsic mode function* yang diambil adalah 2 sampai sebelum terakhir, batas frekuensi yang diambil sebesar 0-10 Hz, penggunaan pengukuran jarak *euclidean* dan penggunaan *dynamic time warping* pada pendeteksian berbasis pembelajaran secara terurut. Parameter-parameter tersebut menghasilkan nilai akurasi terbaik sebesar 89,49% untuk pendeteksian berdasarkan aturan pada skenario uji coba 3 dan 96,29% untuk pendeteksian berdasarkan pembelajaran pada skenario uji coba 5.

Kelebihan menggunakan pengenalan berbasis aturan adalah tidak diperlukan waktu untuk pembentukan model dan durasi

waktu pendeteksian sinyal masukan yang singkat. Sedangkan nilai akurasi tertinggi pada pengenalan berbasis pembelajaran dihasilkan dari skenario 5 namun dengan rata-rata lama waktu pendeteksian terlama. Berdasarkan hasil nilai akurasi dan performa skenario yang disajikan, skenario 3 memberikan hasil terbaik jika diimplementasikan pada sistem pendeteksian waktu nyata karena waktu deteksi yang singkat dan nilai akurasi yang cukup tinggi. Tabel 5.42 menunjukkan perbandingan untuk masing-masing metode preprocessing.

**Tabel 5.42 Perbandingan metode *filtering/preprocessing***

<b><i>Preprocessing</i></b>	<b>Kelebihan</b>	<b>Kekurangan</b>
Averaging	mudah dipahami, menghilangkan <i>noise</i> mayor.	Rentan terhadap <i>noise</i> minor.
EMD	Menghilangkan <i>noise</i> minor pada indeks tertentu.	Rentan terhadap pelengkungan <i>baseline</i> karena penyertaan residu.
Bandpass	Tidak merubah posisi <i>baseline</i> , menghilangkan <i>noise</i> minor dan mayor.	Perlu mengubah ke dalam domain frekuensi.

Hasil-hasil yang didapatkan hanya berlaku untuk dataset yang digunakan pada tugas akhir ini. Dengan kata lain, skenario dan parameter terbaik akan menghasilkan nilai akurasi dan rata-rata lama waktu pendeteksian terbaik jika diterapkan pada dataset sinyal yang digunakan pada pengerjaan tugas akhir ini.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

#### **6.1 Kesimpulan**

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan antara lain :

1. Terkadang sinyal yang dihasilkan tampak tidak berpola karena pengaruh kondisi sekitar seperti interferensi gelombang elektromagnetik dan suara, pemasangan alat yang kurang tepat serta kondisi mental subjek saat perekaman data berlangsung. Hal seperti ini akan memperburuk hasil pendeteksian.
2. Filter terbaik yang diterapkan pada *preprocessing* sinyal masukan adalah filter bandpass dengan frekuensi yang diambil 0-10 Hz dengan nilai akurasi 89,3%.
3. Penggunaan DTW sebagai metode pengukur jarak pada sinyal masukan memberikan pengaruh yang signifikan dibanding dengan pengukuran jarak menggunakan *euclidean*. Namun kekurangan dari metode ini adalah waktu pengenalan yang lama, yaitu sekitar 8,5 detik sehingga kurang cocok diimplementasikan sebagai metode pada sistem waktu nyata.

#### **6.2 Saran**

Beberapa saran yang disampaikan terkait pengerjaan tugas akhir ini adalah :

1. Pengambilan sebaiknya dilakukan di lingkungan yang tenang dan kondisi mental subjek yang stabil.

2. Pengambilan data kedipan mata pengguna seharusnya menggunakan perangkat yang khusus untuk sinyal EOG. Terdapat dua jenis perangkat yang digunakan, sinyal EOG vertikal dan sinyal EOG horizontal.
3. Perlu ditambahkan sistem pendeteksi yang mampu mengenali jeda antara huruf.
4. Perlu dikembangkan sistem pengkodean yang lebih singkat dan mampu menangani masukan per kata sehingga menjadi media komunikasi baru bagi penyandang cacat parah.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

#### **6.1 Kesimpulan**

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan antara lain :

1. Terkadang sinyal yang dihasilkan tampak tidak berpola karena pengaruh kondisi sekitar seperti interferensi gelombang elektromagnetik dan suara, pemasangan alat yang kurang tepat serta kondisi mental subjek saat perekaman data berlangsung. Hal seperti ini akan memperburuk hasil pendeteksian.
2. Filter terbaik yang diterapkan pada *preprocessing* sinyal masukan adalah filter bandpass dengan frekuensi yang diambil 0-10 Hz dengan nilai akurasi 89,3%.
3. Penggunaan DTW sebagai metode pengukur jarak pada sinyal masukan memberikan pengaruh yang signifikan dibanding dengan pengukuran jarak menggunakan *euclidean*. Namun kekurangan dari metode ini adalah waktu pengenalan yang lama, yaitu sekitar 8,5 detik sehingga kurang cocok diimplementasikan sebagai metode pada sistem waktu nyata.

#### **6.2 Saran**

Beberapa saran yang disampaikan terkait pengerjaan tugas akhir ini adalah :

1. Pengambilan sebaiknya dilakukan di lingkungan yang tenang dan kondisi mental subjek yang stabil.

2. Pengambilan data kedipan mata pengguna seharusnya menggunakan perangkat yang khusus untuk sinyal EOG. Terdapat dua jenis perangkat yang digunakan, sinyal EOG vertikal dan sinyal EOG horizontal.
3. Perlu ditambahkan sistem pendeteksi yang mampu mengenali jeda antara huruf.
4. Perlu dikembangkan sistem pengkodean yang lebih singkat dan mampu menangani masukan per kata sehingga menjadi media komunikasi baru bagi penyandang cacat parah.

**DAFTAR PUSTAKA**

- [1] C.-M. Wu, C.-H. Luo, S.-W. Lin, S.-C. Chen, M.-C. Hsieh, C.-T. Chao dan C.-C. Tai, "Morse Code Recognition System with Adaptive Fuzzy Algorithm for the Disabled," *Journal of Medical and Biological Engineering*, 2002.
- [2] Anonim, "Morse Codes for Computer Access," [Online]. Available: <http://www.makoa.org/jlubin/morsecode.htm>. [Diakses 18 Desember 2014].
- [3] Kemalasari, M. Hery P, R. W dan N. S, "Pengolahan Sinyal Elektroensefalogram (EEG) Sistem Peletakan 8 Elektrode dengan Metode Wavelet Transform," 2003.
- [4] S. Rahmaditya, "Pembuatan Decoder Morse dengan Keluaran Suara dan Cahaya Berbasis Mikrokontroler AT89S52," 2009.
- [5] M. Lambert, A. Engroff, D. Matt dan B. Ben, "Empirical Mode Decomposition (Introduction)," [Online]. Available: <https://www.clear.rice.edu/elec301/Projects02/empiricalMode/index.html>. [Diakses 18 Desember 2014].
- [6] M. Lambert, A. Engroff, D. Matt dan B. Ben, "Empirical Mode Decomposition (Process)," [Online]. Available: <https://www.clear.rice.edu/elec301/Projects02/empiricalMode/process.html>. [Diakses 18 Desember 2014].
- [7] M. H. Tri Budi Santoso, "Transformasi Fourier Diskrit," Surabaya.

- [8] Anonim, “Landasan Teori,” 1 2012. [Online]. Available: <http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2012-1-00533-mtif%202.pdf>. [Diakses 12 Juli 2015].
- [9] Anonim, “DTW Algorithm,” [Online]. Available: <http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWAlgorithm.htm>. [Diakses 12 Juni 2015].
- [10] scikit-learn, “Nearest Neighbors,” [Online]. Available: <http://scikit-learn.org/stable/modules/neighbors.html>. [Diakses 12 6 2015].
- [11] B. DeWilde, “Classification of Hand-written Digits,” [Online]. Available: <http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/>. [Diakses 12 Juni 2015].
- [12] Anonim, “A Tutorial on Clustering Algorithm,” [Online]. Available: [http://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/kmeans.html](http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html). [Diakses 16 Juni 2015].
- [13] Anonim, “JSON,” [Online]. Available: <http://json.org/>. [Diakses 6 July 2015].
- [14] NeuroSky Inc, “ThinkGear Socket Protocol,” 14 December 2009. [Online]. Available: [http://developer.neurosky.com/docs/lib/exe/fetch.php?media=app\\_notes:thinkgear\\_socket\\_protocol.pdf](http://developer.neurosky.com/docs/lib/exe/fetch.php?media=app_notes:thinkgear_socket_protocol.pdf). [Diakses 1 2 2015].
- [15] M. Abo-Zahhad, S. M. Ahmed dan S. N. Abbas, “A Novel Biometric Approach for Human Identification and

Verification Using Eye Blinking Signal,” *Signal Processing Letters*, 2015.

## LAMPIRAN

Pada bagian lampiran, akan dilampirkan *confusion matrix* untuk masing-masing skenario dengan parameter terbaik.

		HASIL SISTEM												
		A	B	C	D	E	F	G	H	I	J	K	L	M
GROUDNTRUTH	A	18	0	0	0	0	0	0	0	2	0	0	0	0
	B	0	19	0	1	0	0	0	1	0	0	0	0	0
	C	0	0	15	0	0	0	0	0	0	0	1	0	0
	D	0	1	0	16	0	0	0	1	0	0	0	0	0
	E	1	0	0	0	12	0	0	0	5	0	0	0	1
	F	1	0	0	0	1	11	1	2	0	0	0	0	0
	G	0	0	0	0	0	0	14	0	0	0	0	0	0
	H	0	0	0	0	0	0	0	23	0	0	0	0	0
	I	0	0	0	1	0	0	0	0	11	0	0	0	2
	J	0	0	0	0	0	0	0	6	0	16	0	0	0
	K	0	0	0	0	0	0	0	0	0	0	20	0	0
	L	0	0	0	0	0	0	0	8	0	0	0	12	0
	M	0	0	0	0	0	0	0	0	21	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0
	O	0	0	0	0	0	0	0	0	0	0	0	0	0
	P	0	0	0	0	0	1	3	0	0	0	0	0	0
	Q	2	0	0	0	0	0	1	0	1	0	2	0	1
	R	1	0	0	0	0	0	0	0	0	0	0	0	0
	S	0	0	0	0	0	0	0	2	0	0	0	0	0
	T	0	0	0	0	1	0	0	0	2	0	0	0	0
	U	2	0	0	0	0	0	0	0	0	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0
	W	0	0	0	0	0	0	0	0	1	0	0	0	1
	X	0	0	0	0	0	0	0	0	0	0	0	0	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0
	Z	0	0	0	0	0	0	3	0	0	0	0	0	0

Lampiran 7.1 *Confusion matrix* uji coba skenario 1 parameter window 30

**HASIL SISTEM**

<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	<b>#</b>
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0	0	0	0	0	1
1	0	0	0	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0
2	0	0	0	2	0	0	2	0	0	0	0	0	0
3	0	0	0	2	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	6	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	19	0	0	0	0	0	0	0	0
1	0	15	0	1	0	0	0	0	0	0	0	0	0
0	2	0	6	1	2	1	2	0	0	0	0	0	0
1	0	0	0	17	2	0	0	0	0	0	0	0	0
0	0	0	0	0	19	0	0	0	0	0	0	0	0
0	0	0	0	0	0	18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	18	0	0	0	0	0	1
0	0	0	0	0	0	0	2	19	0	0	0	0	0
0	0	0	0	0	0	0	0	0	19	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	19	0	0
0	0	0	0	0	0	1	0	0	0	0	0	9	0

		HASIL SISTEM												
		A	B	C	D	E	F	G	H	I	J	K	L	M
GROUNDTRUTH	A	19	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	20	0	0	0	0	0	0	0	0	0	0	0
	C	0	0	14	0	0	0	3	0	0	0	2	0	0
	D	0	0	0	15	0	0	1	0	0	0	0	0	0
	E	0	1	0	0	9	0	0	0	0	0	0	0	0
	F	1	0	0	1	1	11	1	0	0	0	0	0	0
	G	0	0	0	1	0	0	16	0	0	0	1	0	0
	H	0	2	0	0	0	0	0	19	0	0	0	0	0
	I	1	1	0	1	0	0	0	0	10	0	0	1	0
	J	0	0	0	0	0	0	0	0	0	20	0	0	0
	K	0	0	0	0	0	0	0	0	0	0	20	0	2
	L	1	0	0	0	0	0	0	0	0	0	0	13	0
	M	0	0	0	0	0	0	0	0	0	0	0	0	20
	N	0	0	0	0	0	0	0	0	0	0	0	0	0
	O	0	0	0	0	0	0	0	0	0	1	0	0	2
	P	0	0	0	0	0	0	0	1	0	1	0	0	0
	Q	0	1	0	1	0	0	0	0	0	0	1	0	0
	R	3	0	0	0	0	1	0	1	0	0	0	1	0
	S	0	0	0	0	0	1	0	3	1	0	0	0	0
	T	0	0	0	0	0	0	0	0	1	0	0	0	0
	U	3	0	0	0	0	0	0	0	0	0	0	0	0
	V	1	0	0	0	0	0	0	0	0	0	1	0	0
	W	1	0	0	0	0	0	0	0	0	2	0	0	3
	X	1	1	0	0	0	0	0	0	0	0	1	0	0
	Y	0	0	0	1	0	0	0	0	0	0	0	0	1
	Z	0	0	0	0	0	0	7	0	0	0	0	0	1

Lampiran 7.2 Confusion matrix uji coba skenario 2  
parameter [2; -1]

N	O	P	Q	R	S	T	U	V	W	X	Y	Z	#
0	0	0	0	0	0	2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	1	0
8	0	0	0	0	0	3	0	0	0	0	0	0	0
1	0	0	0	2	0	0	1	0	2	1	0	0	0
1	0	0	1	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0	0	0	0
1	0	0	0	4	2	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	6	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0
0	18	0	0	0	0	0	0	0	0	0	0	0	0
0	0	18	0	1	0	0	0	0	0	0	0	0	0
1	5	0	11	0	0	0	0	0	0	0	0	0	1
0	1	1	0	12	0	0	0	0	1	0	0	0	0
0	0	0	0	0	16	0	0	0	0	0	0	0	0
1	0	0	0	0	0	19	0	0	0	0	0	0	0
0	0	0	0	0	0	0	18	0	0	0	0	0	0
0	0	0	0	0	0	0	4	15	0	0	0	0	0
0	0	0	0	1	0	0	0	0	14	0	0	0	0
0	0	0	0	0	0	2	0	1	0	15	1	0	0
0	4	0	0	0	0	0	0	0	0	1	14	0	0
0	1	0	0	0	0	1	0	0	0	0	0	11	0

**HASIL SISTEM**

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	20	0	1	0	0	0	0	0	0	0	0	0
C	0	0	18	1	0	0	0	0	0	0	0	0	0
D	0	2	0	16	0	0	0	1	0	0	0	0	0
E	2	0	0	0	15	0	0	0	1	0	0	0	0
F	0	0	0	0	1	18	0	0	0	0	0	0	0
G	0	0	0	0	0	0	17	0	0	0	1	0	0
H	0	0	0	0	0	0	0	23	0	0	0	0	0
I	0	2	0	1	0	0	0	0	15	0	0	0	0
J	0	0	0	0	0	0	0	0	0	22	0	0	0
K	0	0	0	0	0	0	0	0	0	0	21	0	0
L	1	0	0	0	0	0	0	0	0	0	0	20	0
M	0	0	0	0	0	0	0	0	0	0	0	0	21
N	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	2	1	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	1	0	0
R	0	0	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	1	0	0	0	0	0
T	0	0	0	0	0	0	0	0	1	0	0	0	2
U	1	0	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	1	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	1	0	0	1
X	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	1	0	0	0	0	0	0

**GROUNDTRUTH**

**Lampiran 7.3 Confusion matrix uji coba skenario 3 parameter  
[0, 10]**

### HASIL SISTEM

N	O	P	Q	R	S	T	U	V	W	X	Y	Z	#
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	0
1	0	0	0	0	0	0	0	0	0	0	0	2	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0
0	20	0	1	0	0	0	0	0	0	0	0	0	0
0	0	18	0	0	0	0	0	0	0	0	0	0	0
1	2	0	10	0	0	0	0	0	0	2	0	2	3
0	0	0	0	20	0	0	0	0	0	0	0	0	0
0	0	0	0	0	20	0	0	0	0	0	0	0	0
0	0	0	0	0	0	18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	18	0	0	0	0	0	2
0	0	0	0	0	0	0	2	15	0	0	0	0	3
0	0	0	0	0	0	0	0	0	19	0	0	0	0
0	0	0	2	0	0	0	0	0	0	20	0	0	0
0	0	0	0	0	0	0	0	0	0	1	20	0	0
0	0	0	0	0	0	0	1	0	0	0	0	19	0

		HASIL SISTEM												
		A	B	C	D	E	F	G	H	I	J	K	L	M
GROUNDTRUTH	A	14	0	0	0	5	0	0	0	0	0	0	0	0
	B	3	5	3	3	2	0	0	0	0	0	1	0	0
	C	1	1	7	2	5	0	0	0	0	0	0	0	0
	D	0	0	0	13	7	0	0	0	0	0	0	0	0
	E	0	0	0	0	21	0	0	0	0	0	0	0	0
	F	9	0	3	6	0	2	0	0	0	0	0	0	0
	G	4	0	0	9	1	0	3	0	1	0	0	0	1
	H	0	0	0	2	8	0	0	10	0	0	0	0	0
	I	0	0	0	0	14	0	0	0	5	0	0	0	0
	J	0	0	2	4	0	0	0	2	0	8	0	0	0
	K	0	1	0	11	1	0	0	0	0	0	6	0	0
	L	1	1	0	2	1	0	0	0	0	0	0	6	0
	M	4	0	0	2	1	0	0	0	0	0	0	0	14
	N	1	0	0	0	4	0	0	0	1	0	0	0	0
	O	0	0	1	1	0	0	0	0	0	0	3	1	0
	P	0	0	0	4	2	0	0	1	2	0	0	0	0
	Q	1	0	1	1	0	0	0	1	0	0	0	0	0
	R	2	0	0	0	10	0	0	0	0	0	0	0	0
	S	2	0	0	0	15	0	0	0	0	0	0	0	0
	T	0	0	0	0	11	0	0	0	0	0	0	0	0
	U	4	0	0	0	8	0	0	0	0	0	0	0	1
	V	0	1	2	4	0	0	0	0	0	0	2	0	0
	W	0	0	0	0	1	0	0	0	1	0	0	0	0
	X	1	2	3	5	0	0	0	1	0	0	2	0	0
	Y	2	1	3	0	1	0	0	0	0	0	2	0	0
	Z	1	0	6	1	0	0	0	0	0	0	0	0	0

Lampiran 7.4 *Confusion matrix uji coba skenario 4 semua training set*

**HASIL SISTEM**

<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
0	0	0	0	0	0	2	0	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	2	0	0	0	0	0	0
0	0	1	0	0	4	1	0	0	0	0	0	0
0	0	0	0	0	3	0	0	0	0	0	0	0
1	0	0	0	0	2	7	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0
0	11	0	0	1	0	0	2	0	0	0	0	1
0	0	3	0	0	5	3	0	1	0	0	0	0
0	0	0	12	0	5	0	0	0	0	0	0	0
0	0	0	0	7	1	1	0	0	0	0	0	0
0	0	0	0	0	3	1	0	0	0	0	0	0
0	0	0	0	0	0	10	0	0	0	0	0	0
0	0	0	0	0	0	1	7	0	0	0	0	0
0	0	0	0	0	1	0	3	8	0	0	0	0
0	0	0	0	0	0	1	1	0	17	0	0	0
0	0	0	0	0	0	0	0	0	0	7	0	1
0	0	0	0	0	0	0	1	1	0	0	10	0
0	1	1	1	0	2	0	0	1	0	1	0	6

## HASIL SISTEM

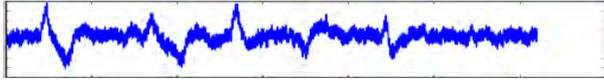
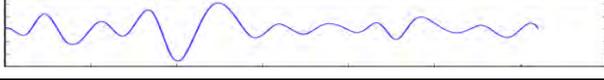
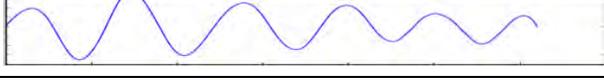
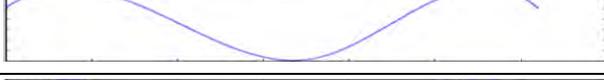
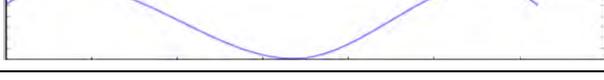
	A	B	C	D	E	F	G	H	I	J	K	L	M
A	21	0	0	0	0	0	0	0	0	0	0	0	0
B	0	21	0	0	0	0	0	0	0	0	0	0	0
C	0	0	17	0	0	0	1	0	0	0	0	0	0
D	0	1	0	20	0	0	0	0	0	0	0	0	0
E	0	0	0	0	21	0	0	0	0	0	0	0	0
F	0	0	1	0	0	20	0	0	0	0	0	0	0
G	0	0	0	0	0	0	20	1	0	0	0	0	0
H	0	0	0	0	0	0	0	23	0	0	0	0	0
I	1	0	0	0	0	0	0	0	19	0	0	0	0
J	0	0	0	0	0	0	0	0	0	22	0	0	0
K	0	0	0	0	0	0	0	0	0	0	22	0	0
L	0	0	0	0	0	0	0	0	0	0	0	21	0
M	0	0	0	0	0	0	0	0	0	0	0	0	21
N	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	2	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	1	0	0	1	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUNDTRUTH

Lampiran 7.5 Confusion matrix uji coba skenario 5 semua training set



**Lampiran 7.6 Contoh hasil IMF**

<b>Indeks IMF ke</b>	<b>Gambar</b>
Asli	 A plot of the original signal, showing a complex, noisy waveform with multiple peaks and troughs.
0	 A plot of IMF 0, showing a dense, noisy signal that appears as a thick blue band.
1	 A plot of IMF 1, showing a signal with a higher frequency than IMF 0, but still containing some noise.
2	 A plot of IMF 2, showing a signal with a higher frequency than IMF 1, appearing as a more regular, oscillating waveform.
3	 A plot of IMF 3, showing a signal with a higher frequency than IMF 2, appearing as a smooth, oscillating waveform.
4	 A plot of IMF 4, showing a signal with a higher frequency than IMF 3, appearing as a smooth, oscillating waveform.
5	 A plot of IMF 5, showing a signal with a higher frequency than IMF 4, appearing as a smooth, oscillating waveform.
6	 A plot of IMF 6, showing a signal with a higher frequency than IMF 5, appearing as a smooth, oscillating waveform.
7	 A plot of IMF 7, showing a signal with a higher frequency than IMF 6, appearing as a smooth, oscillating waveform.
8	 A plot of IMF 8, showing a signal with a higher frequency than IMF 7, appearing as a smooth, oscillating waveform.

## BIODATA PENULIS



Ahmad Hayam Brilian, penulis dari buku tugas akhir ini lahir di kota Bojonegoro tanggal 9 Oktober 1992. Penulis telah menempuh pendidikan di SD Negeri Campurejo 1 (1999-2005), SMP Negeri 1 Bojonegoro (2005-2008), SMA Negeri 1 Bojonegoro (2008-2011) dan Teknik Informatika ITS Surabaya (2011-2015). Selama masa perkuliahan, penulis pernah menjadi asisten pada mata kuliah

Grafika Komputer dan Software Perkantoran PIKTI ITS. Penulis juga aktif sebagai anggota organisasi Keluarga Muslim Informatika (KMI), Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS dan Jamaah Masjid Manurul Ilmi (JMMI) ITS. Penulis memilih bidang minat Komputasi Cerdas dan Visi (KCV) dan tertarik pada topik pengenalan pola dan kecerdasan buatan. Penulis dapat dihubungi melalui email [hayam11@mhs.if.its.ac.id](mailto:hayam11@mhs.if.its.ac.id).