



**TUGAS AKHIR - KI141502**

**KLASIFIKASI EMOSI DARI DATA SINYAL EEG MENGGUNAKAN  
INDEPENDENT COMPONENT ANALYSIS (ICA), WAVELET  
DENOISING DAN MULTIPLE DISCRIMINANT ANALYSIS (MDA)**

**ALDY SYAHDEINI  
NRP 5111100122**

**Dosen Pembimbing  
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.  
Dini Adni Navastara, S.Kom, M.Sc.**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**



**UNDERGRADUATE THESES - KI141502**

**EMOTION CLASSIFICATION FROM EEG SIGNAL DATA  
USING INDEPENDENT COMPONENT ANALYSIS (ICA),  
WAVELET DENOISING AND MULTIPLE DISCRIMINANT  
ANALYSIS (MDA)**

**ALDY SYAHDEINI  
NRP 5111100122**

**Advisor**

**Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.  
Dini Adni Navastara, S.Kom, M.Sc.**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2015**

## LEMBAR PENGESAHAN

### KLASIFIKASI EMOSI DARI DATA SINYAL EEG MENGUNAKAN *INDEPENDENT COMPONENT ANALYSIS (ICA)*, *WAVELET DENOISING* DAN *MULTIPLE DISCRIMINANT ANALYSIS (MDA)*

#### 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

**ALDY SYAHDEINI**  
**NRP : 51111100 122**

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Prof. Ir. Handayani Tjandrasa, S.Soc., Ph.D. ....  
NIP:194908231976032001 (Pembimbing 1)
2. Dini Adni Navastara, S.Kom, M.Sc. ....  
NIP:5100201405003 (Pembimbing 2)



**SURABAYA**  
**JANUARI, 2015**

## KATA PENGANTAR

Alhamdulillah penulis dapat menyelesaikan Tugas Akhir yang berjudul **“KLASIFIKASI EMOSI DARI DATA SINYAL EEG DENGAN MENGGUNAKAN INDEPENDENT COMPONENT ANALYSIS (ICA), WAVELET DENOISING DAN MULTIPLE DISCRIMINANT ANALYSIS (MDA)”**. Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Ayah, ibu, adik dan kakak memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
2. Ibu Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D. selaku pembimbing I yang telah membantu dan membimbing, serta dengan sabar mengajarkan ilmu kepada penulis dalam menyelesaikan Tugas Akhir.
3. Ibu Dini Adni Navastara, S.Kom, M.Sc. selaku pembimbing II yang telah memberikan motivasi, nasehat, bimbingan dan bantuan yang banyak kepada penulis dalam mengerjakan Tugas Akhir ini.
4. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, yang telah memberikan masukan-masukan dan ide mengenai pengerjaan tugas akhir ini.
5. Teman-teman *administrator* Laboratorium pemrograman yang telah memberikan dukungan dan menghibur penulis.
6. Senior-senior laboratorium pemrograman mas almas, tedjo, firman, demit, galang, dadang, andre, felix, sindung, ryan dan mbak nada

telah memberikan motivasi dan dorongan untuk berjuang mengerjakan tugas akhir.

7. Teman admin aida, kemal, asri, farhan, danang, supri, ali, kristo, otniel, karsten, ade, hamdi, pinas, dinar, ratih yang telah menyemangati penulis selama pengerjaan tugas akhir ini.
8. Teman-teman TC angkatan 2011 dessy, nisa, ade, haidar, hayam dll. yang selalu menjaga kebersamaan.
9. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Januari 2015

# **KLASIFIKASI EMOSI DARI DATA SINYAL EEG MENGUNAKAN INDEPENDENT COMPONENT ANALYSIS (ICA), WAVELET DENOISING DAN MULTIPLE DISCRIMNANT ANALYSIS (MDA)**

**Nama Mahasiswa** : ALDY SYAHDEINI  
**NRP** : 5111100122  
**Jurusan** : Teknik Informatika FTIF-ITS  
**Dosen Pembimbing 1** : Prof. Ir. Handayani Tjandrasa, M.Sc.,  
Ph.D.  
**Dosen Pembimbing 2** : Dini Adni Navastara, S.Kom, M.Sc.

## ***Abstrak***

*Emosi ialah keadaan psikologi pada alam sadar manusia sebagai reaksi terhadap sebuah objek atau keadaan yang juga dikaitkan dengan suasana hati, temperamen, jati diri, watak dan motivasi. Emosi dapat diekspresikan dengan tindakan verbal melalui ucapan, maupun non-verbal seperti intonasi suara atau gerakan tubuh. Emosi sangat penting dalam meningkatkan kualitas kehidupan manusia. Dengan adanya sistem kecerdasan emosi pada komputer, diharapkan manusia dapat merasa lebih nyaman dalam berinteraksi dan bekerja dengan komputer.*

*Emosi dapat dikenali dengan menggunakan sinyal electroencephalography (EEG). Pergerakan mata, kedipan mata, gerakan otot, sinyal jantung dapat membuat sinyal EEG menjadi tidak akurat sehingga diperlukan preprocessing data yang efektif.*

*Dalam tugas akhir ini penulis melakukan klasifikasi emosi dari sinyal EEG menggunakan Multiple Discriminant Analysis (MDA), dimana sebelumnya dilakukan preprocessing menggunakan Independent Component Analysis (ICA) dan wavelet denoising. Berdasarkan hasil percobaan, klasifikasi emosi dengan preprocess menggunakan ICA dengan merata-ratakan 100 fitur menghasilkan akurasi sebesar 70%, sedangkan tanpa menggunakan ICA*

*menghasilkan akurasi sebesar 65%. Hal ini membuktikan bahwa preprocess menggunakan ICA lebih efektif untuk mengurangi noise dan meningkatkan akurasi klasifikasi dibandingkan tanpa ICA.*

***Kata kunci: Emosi, electroencephalography, klasifikasi, Independent Component Analysis, wavelet denoising, Multiple Discriminant Analysis***

**EMOTION CLASSIFICATION FROM EEG SIGNAL  
DATA USING INDEPENDENT COMPONENT ANALYSIS  
(ICA), WAVELET DENOISING AND MULTIPLE  
DISCRIMINANT ANALYSIS (MDA)**

**Student's Name** : ALDY SYAHDEINI  
**Student's ID** : 5111100122  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Prof. Ir. Handayani Tjandrasa, M.Sc.,  
Ph.D.  
**Second Advisor** : Dini Adni Navastara, S.Kom, M.Sc.

**Abstract**

*Emotion is a psychological state of the human subconscious as a reaction from an object or a situation which also associated with mood, temperament, identity, character and motivation. Emotion can be expressed by the verbal action through speech or body movement.*

*Emotion is very important in improving the quality of human life. With the emotional intelligence in computer systems, it is expected that people can be more comfortable in interacting and working with computers.*

*Emotion can be identified by using electroencephalography (EEG). Eye movements, blinking, muscle movement and heart signal can make the EEG signal becomes inaccurate, so it is necessary to preprocess the data using the effective method.*

*In this thesis the author tries to classify emotions from EEG signals data using Multiple Discriminant Analysis (MDA) and to preprocess using Independent Component Analysis (ICA) and wavelet denoising. Based on the result of the trials, emotion classification using ICA and 100 features in average, the classification produces 70% of accuracy. On other hand without using ICA, it produces 65% of accuracy. Therefore we can*

*conclude that preprocessing signals using ICA are more effective for reducing the noise than without ICA.*

***Keywords: Emotion, electroencephalography, classification, Independent Component Analysis, wavelet denoising, Multiple Discriminant Analysis***

## DAFTAR ISI

LEMBAR PENGESAHAN .....	v
<i>Abstrak</i> .....	vii
Abstract .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
DAFTAR KODE SUMBER .....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	4
1.5 Manfaat .....	4
1.6 Metodologi .....	4
1.7 Sistematika Penulisan Laporan Tugas Akhir .....	6
BAB II TINJAUAN PUSTAKA .....	9
2.1 Model emosi .....	9
2.2 Sinyal EEG .....	11
2.3 <i>Independent Component Analysis</i> .....	13
2.4 <i>Wavelet Denoising</i> .....	15
2.4.1 <i>Discrete Wavelet Transform</i> .....	15
2.4.2 <i>Stationary Wavelet Transform</i> .....	17
2.5 <i>Multiple Discriminant Analysis</i> .....	18
BAB III DESAIN .....	21
3.1 Desain Metode Secara Umum .....	21
3.2 <i>Preprocessing data</i> .....	22
3.2.1 <i>Independent Component Analysis</i> .....	22
3.2.2 <i>Stationary Wavelet Transform</i> .....	23
3.3 Klasifikasi data .....	25

BAB IV IMPLEMENTASI.....	31
4.1 Lingkungan Implementasi.....	31
4.2 Implementasi <i>High Pass Filter</i> .....	31
4.3 Implementasi Fungsi ICA.....	31
4.4 Implementasi Fungsi SWT.....	32
4.5 Implementasi Kelas MDA.....	33
4.5.1 Fungsi <i>Training</i> .....	33
4.5.2 Fungsi <i>Testing</i> .....	35
BAB V UJI COBA DAN EVALUASI.....	37
5.1 Lingkungan Uji Coba.....	37
5.2 Data DEAP.....	37
5.3 Pemrosesan Data.....	38
5.4 Skenario dan Evaluasi Pengujian.....	40
5.4.1 Skenario dan evaluasi Uji Coba 1.....	45
5.4.2 Skenario dan evaluasi Uji Coba 2.....	45
5.4.3 Skenario dan evaluasi Uji Coba 3.....	46
5.4.4 Skenario dan evaluasi Uji Coba 4.....	46
5.4.5 Skenario Uji dan evaluasi Coba 5.....	47
5.4.6 Skenario Uji dan evaluasi Coba 6.....	47
5.4.7 Evaluasi Semua skenario.....	47
BAB VI KESIMPULAN DAN SARAN.....	57
6.1 Kesimpulan.....	57
6.2 Saran.....	57
DAFTAR PUSTAKA.....	59
LAMPIRAN.....	61
BIODATA PENULIS.....	67

## DAFTAR GAMBAR

Gambar 2.1 <i>Circumplex model</i> emosi .....	10
Gambar 2.2 Hasil percobaan Russel.....	10
Gambar 2.3 Objek sebelum melakukan percobaan .....	12
Gambar 2.4 Peta <i>channel</i> pada BCI .....	12
Gambar 2.5 Implementasi circumplex model pada video percobaan .....	13
Gambar 2.6 Hasil Koreksi Sinyal EEG Menggunakan ICA .....	16
Gambar 2.7 Dekomposisi Sinyal EEG dengan ICA.....	16
Gambar 2.8 Struktur dari DWT.....	17
Gambar 2.9 Struktur dari SWT .....	18
Gambar 2.10 Perbedaan Hasil dari DWT dan SWT.....	18
Gambar 2.11 Hasil dari Multiple Discriminant Analysis .....	19
Gambar 3.1 Metode secara umum.....	21
Gambar 3.2 Kerangka kerja preprocessing .....	22
Gambar 3.3 Alur dekomposisi SWT .....	25
Gambar 3.4 Alur kerja SWT .....	26
Gambar 3.5 Proses klasifikasi .....	26
Gambar 4.1 <i>High pass filter</i> dengan EEGLAB .....	32
Gambar 5.1 Reduksi sampling rate dengan EDFBrowser.....	39
Gambar 5.2 Hasil pengurangan sampling rate .....	39
Gambar 5.3 Plot data setelah re-reference dan high pass filter ...	41
Gambar 5.4 Plot data setelah di ICA .....	41
Gambar 5.5 Hasil dekomposisi dari SWT .....	42
Gambar 5.6 Sinyal hasil rata-rata semua channel pada satu trial	42
Gambar 5.7 Sinyal asli untuk tiap kelas data .....	43
Gambar 5.8 Contoh sinyal tanpa rata-rata pada kerangka kerja 1 .....	48
Gambar 5.9 Sinyal dengan rata-rata 50 fitur pada kerangka kerja 1 .....	48
Gambar 5.10 Sinyal dengan rata-rata 100 fitur pada kerangka kerja 1 .....	49
Gambar 5.11 Sinyal dengan rata-rata 500 fitur pada kerangka kerja 1 .....	49

Gambar 5.12 Hasil <i>output</i> program untuk skenario 1.....	49
Gambar 5.13 Hasil <i>output</i> program untuk skenario 2.....	50
Gambar 5.14 Hasil <i>output</i> program untuk skenario 3.....	50
Gambar 5.15 Contoh sinyal tanpa rata-rata pada kerangka kerja 2 .....	50
Gambar 5.16 Sinyal yang telah dirata-ratakan tiap 50 fitur pada kerangka kerja 2 .....	51
Gambar 5.17 Sinyal yang telah dirata-ratakan tiap 100 fitur pada kerangka kerja 2 .....	51
Gambar 5.18 Sinyal yang telah dirata-ratakan tiap 500 fitur pada kerangka kerja 2 .....	51
Gambar 5.19 Hasil <i>ouput</i> program untuk skenario 4.....	52
Gambar 5.20 Hasil <i>ouput</i> program untuk skenario 5.....	52
Gambar 5.21 Hasil <i>ouput</i> program untuk skenario 6.....	52
Gambar 5.22 Sinyal hasil prediksi kelas pada skenario 1 .....	53
Gambar 5.23 Sinyal hasil prediksi kelas pada skenario 2 .....	53
Gambar 5.24 Sinyal hasil prediksi kelas pada skenario 3 .....	54
Gambar 5.25 Sinyal hasil prediksi kelas pada skenario 4 .....	54
Gambar 5.26 Sinyal hasil prediksi kelas pada skenario 5 .....	55
Gambar 5.27 Sinyal hasil prediksi kelas pada skenario 6 .....	55
Gambar 5.28 Akurasi semua kerangka kerja.....	56

## DAFTAR TABEL

Tabel 3.1 Pemisahan band pada SWT .....	24
Tabel 5.1 Hasil Klasifikasi kedua kerangka kerja .....	56
Tabel A.1 Urutan <i>channel</i> pada data sinyal asli dari DEAP .....	61
Tabel A.2 <i>Video</i> yang digunakan pada percobaan .....	63

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Emosi ialah keadaan psikologi pada alam sadar manusia sebagai reaksi terhadap sebuah objek atau keadaan yang juga dikaitkan dengan suasana hati, temperamen, jati diri, watak dan motivasi. Emosi dapat diekspresikan dengan tindakan verbal melalui ucapan, maupun non-verbal seperti intonasi suara atau gerakan tubuh. Emosi seseorang sangat sulit diprediksi karena dipengaruhi oleh budaya dan lingkungan. Sistem interaksi manusia komputer untuk memahami emosi masih sangat kurang dan minim [1]. Padahal emosi sangat penting dalam meningkatkan kualitas kehidupan manusia. Dengan adanya sistem kecerdasan emosi pada komputer, diharapkan manusia dapat merasa lebih nyaman dalam berinteraksi dan bekerja dengan komputer.

Sekarang ini banyak penelitian yang dilakukan untuk mengenali emosi melalui sinyal EEG [2]. *Electroencephalography* (EEG) ialah rekaman dari informasi aktivitas otak manusia berupa perubahan tegangan dan medan listrik pada kulit kepala. Sinyal EEG ditangkap oleh konduktor yang dipasangkan pada kepala manusia. Peneliti percaya bahwa perubahan sinyal yang ditangkap oleh EEG berhubungan dengan perasaan atau emosi manusia [2].

Pergerakan mata, kedipan mata, gerakan otot dan sinyal jantung dapat membuat sinyal EEG menjadi tidak akurat, karena artefak-artefak ini menimbulkan *noise*. Oleh karena itu perlu dilakukan *preprocessing* yang berguna untuk mengekstrak sinyal asli dan mengurangi *noise*.

*Preprocessing* EEG dapat dilakukan dengan *simple temporal & spatial filter* atau metode *filter* yang lebih canggih seperti *Independent Component Analysis* (ICA), *Common Spatial Patterns* (CSP), *Principal Component Analysis* (PCA) dan

lain sebagainya. ICA sendiri sudah digunakan secara luas sebagai metode untuk memisahkan artefak dari data EEG. ICA akan bertindak sebagai *blind source separation* dengan memisahkan data sinyal ke dalam bentuk data yang *independent* atau disebut *independent component (IC)* [3].

Banyak penelitian yang dilakukan untuk meningkatkan performa dari ICA. Salah satu metode yang digunakan untuk meningkatkan performa ICA ialah *Discrete Wavelet Transform (DWT)*. Kelemahan pada DWT ialah komputasi cenderung lama karena adanya *down sampling*, maka digunakan *Stationary Wavelet Transform (SWT)* yang merupakan fungsi modifikasi dari DWT. Sama seperti DWT, SWT menggunakan fungsi *low pass filter* dan *high pass filter* untuk memisahkan sinyal kedalam frekuensi *band* tertentu pada tiap level dekomposisinya. Pada SWT *down sampling* tidak ada sehingga jumlah fitur dari data tidak berubah dan mempercepat komputasi. [4]

Pada SWT sinyal EEG harus dipartisi sebanyak *level* tertentu serta perlu dilakukan pemilihan fungsi basis *wavelet* dan *level* untuk batas *denoising*. Pemilihan fungsi basis *wavelet* dan *level* harus dipilih dengan hati-hati karena terlalu banyak *dekomposisi* dapat melemahkan frekuensi data [5].

Algoritma klasifikasi EEG yang paling populer yaitu *Multiple Discriminant Analysis (MDA)* dan *Support Vector Machine (SVM)*. Algoritma ini menggunakan fungsi *linear* untuk membedakan kelas dari data. MDA merupakan sebuah algoritma klasifikasi yang cepat karena menggunakan perhitungan matematika yang jauh lebih sedikit dibandingkan dengan algoritma klasifikasi lain. Oleh karena itu MDA sangat cocok digunakan untuk mengklasifikasi data EEG secara *realtime* [6].

Dalam kasus klasifikasi dengan banyak kelas, MDA akan memisahkan fitur atau *variabel* dari sinyal ke bentuk yang *independent*. Kemudian akan dibentuk *hyperplane* yaitu berupa fungsi *discriminant* atau fungsi pembeda yang berguna sebagai fungsi prediksi untuk melakukan klasifikasi data [7].

Dalam tugas akhir ini penulis mengusulkan sebuah sistem “Klasifikasi emosi manusia dari data sinyal EEG menggunakan *Independent Component Analysis (ICA)*, *Wavelet Denoising* dan *Multiple Discriminant Analysis (MDA)*”. Pertama, data sinyal EEG akan diproses terlebih dahulu menggunakan ICA dan SWT untuk menghilangkan *noise* dan mengekstraksi fitur, kemudian data akan diklasifikasi menggunakan MDA.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana kerangka kerja yang paling baik untuk mengklasifikasikan emosi pada data sinyal EEG ?
2. Bagaimana membuktikan bahwa hasil sistem klasifikasi menggunakan ICA, SWT dan MDA mempunyai akurasi yang lebih baik dibandingkan dengan sistem klasifikasi yang menggunakan SWT dan MDA ?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Implementasi dilakukan dengan Bahasa pemrograman *python*, perangkat lunak *EDFBrowser* dan *EEGLAB*.
2. *Level* dekomposisi pada SWT yang digunakan sebesar lima.
3. Fungsi *wavelet* yang digunakan ialah *Haar*.
4. Data yang digunakan ialah *Datataset for emotion analysis using EEG, physiological and video signals (DEAP)*.
5. Jumlah data yang digunakan ialah data pertama sampai data ke 22.

## 1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Mengklasifikasikan emosi dari data sinyal EEG dengan melakukan *preprocessing* menggunakan ICA dan SWT serta algoritma klasifikasi MDA.
2. Membuktikan bahwa *preprocessing* menggunakan ICA mampu meningkatkan akurasi data dibandingkan tanpa menggunakan ICA.

## 1.5 Manfaat

Tugas Akhir ini diharapkan mampu membangun sebuah kerangka kerja untuk mendeteksi emosi dari data sinyal EEG, sehingga dapat meningkatkan interaksi antara manusia dan komputer. Dengan begitu sistem komputer yang ada dapat dibuat lebih berguna, berkesan, menyenangkan atau bahkan memberikan motivasi kepada pengguna. Selain itu, kita dapat memahami bagaimana komputer memberikan efek pada manusia.

## 1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan untuk menguji apakah *preprocessing* dengan ICA dapat meningkatkan akurasi pada klasifikasi data.

2. Studi literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi dan literatur yang diperlukan untuk pembuatan implementasi. Dasar informasi yang diperlukan pada pembuatan implementasi ini diantaranya pemodelan emosi, sinyal emosi DEAP, *Independent Component Analysis* (ICA), *Stationary Wavelet Transform* (SWT) dan *Multiple Discriminant Analysis* (MDA). Informasi dan literatur didapatkan dari literatur buku dan sumber-sumber informasi lain yang berhubungan.

### 3. Pengumpulan data

Data berupa sinyal EEG yang khusus untuk menganalisa emosi didapatkan dari *Database for emotion analysis using physiological signal* (DEAP), yang dapat diunduh pada halaman situs DEAP. Dataset yang diunduh mempunyai ukuran 5,8 *Gigabyte* berupa *file* dengan ekstensi *Biosemi Data Format* (BDF).

### 4. Perancangan perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

### 5. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga

menjadi sebuah program yang sesuai dengan apa yang telah direncanakan. Implementasi akan dilakukan dengan menggunakan EEGLAB dan python untuk tahap *preprocessing*.

## 6. Pengujian dan evaluasi

Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data serta program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan. Pada tahap ini akan dilakukan pengujian terhadap dua kerangka kerja terhadap enam skenario. Hasil klasifikasi berupa akurasi dari kedua kerangka kerja kemudian dievaluasi

## 7. Penyusunan buku Tugas Akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

### 1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

#### **BAB I. PENDAHULUAN**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

#### **BAB II. DASAR TEORI**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

### **BAB III. PERANCANGAN PERANGKAT LUNAK**

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

### **BAB IV. IMPLEMENTASI**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

### **BAB V. UJI COBA DAN EVALUASI**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

### **BAB VI. KESIMPULAN DAN SARAN**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

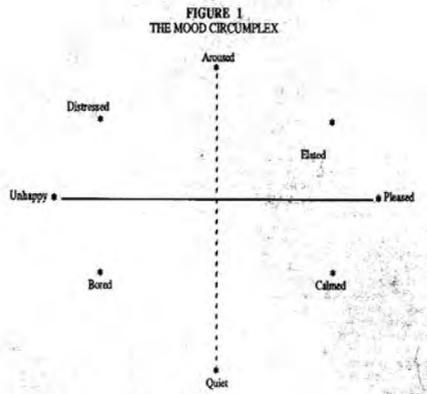
#### **2.1 Model emosi**

Emosi memegang peranan penting dalam kehidupan manusia. Emosi bersifat abstrak sehingga tidak mudah mengetahui emosi dari seseorang karena kompleksitas dari bermacam-macam budaya manusia. Di bidang psikologi banyak peneliti yang mencoba menentukan emosi dasar dari manusia [8]

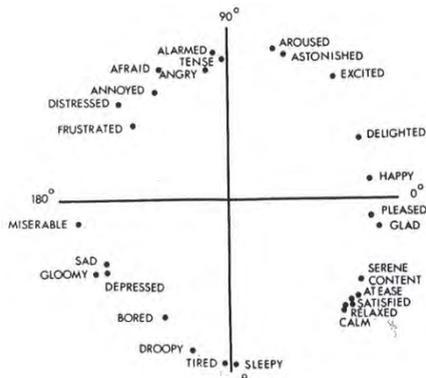
Pada tahun 1980 James A. Russel dari Universitas British Columbia mengemukakan sebuah model dari emosi yang diberi nama *circumplex model*. *Circumplex model* mencoba mengelompokkan emosi manusia kedalam empat dimensi dasar berdasarkan derajat dari gairah atau *arousal*, dan kesenangan atau *pleasure*. Gambar 2.1 memperlihatkan model dasar dari emosi.

Untuk membuktikan model yang dibuatnya, Russel dibantu dengan 36 orang mahasiswa membuat percobaan, dimana subjeknya ialah menempatkan 28 kata bahasa Inggris kedalam model emosi. Setelah melakukan rata-rata dari semua percobaan, didapat hasil berupa sebuah model yang dapat menggambarkan letak-letak keadaan emosi pada model seperti senang, sedih, marah dan mengantuk. Gambar 2.2 memperlihatkan hasil dari percobaan yang dilakukan Russel [8]

Dalam tugas akhir ini, penulis menggunakan *circumplex model* dari Russel, dimana emosi dari sinyal EEG akan dikelompokkan berdasarkan empat dimensi. Dimensi ini dapat dianggap sebagai sebuah kriteria dari emosi.



**Gambar 2.1 Circumplex model emosi**



**Gambar 2.2 Hasil percobaan Russel**

Dimensi pertama dapat mewakili emosi berupa perasaan marah, penderitaan dan kesukaran. Dimensi kedua dapat mewakili emosi berupa perasaan senang dan bergembira. Dimensi ketiga mewakili keadaan santai dan lega. Dimensi keempat mewakili perasaan bosan, sedih dan lelah.

## 2.2 Sinyal EEG

Sinyal *Electroencephalography* (EEG) adalah sebuah rekaman sinyal elektronik dari otak yang didapat dari perangkat yang disebut *Brain Computer Interface* (BCI). BCI merupakan alat yang mencatat aktivitas gelombang otak selama kurun waktu tertentu. Sinyal elektrik yang dihasilkan secara natural oleh otak akan ditangkap oleh kanal-kanal elektrode pada BCI.

EEG membuat peneliti dapat mengukur *impulse* elektrik pada semua permukaan otak dan mengamati perubahan yang terjadi tiap satuan waktu. Sinyal EEG dapat digunakan untuk menganalisa penyakit *epilepsi*, gangguan tidur, kelumpuhan otak ataupun emosi. Aktifitas sinyal EEG cenderung cukup kecil, biasanya dalam ukuran *microvolts* dengan frekuensi utama sampai di atas 30 *Hertz*.

BCI berupa semacam topi yang dipakaikan di kepala. Terdapat beberapa model dari BCI dengan jumlah elektrode atau *channel* yang berbeda-beda. Salah satu perangkat BCI yang sering digunakan dalam bidang kedokteran ialah BCI berupa topi dengan 32 *channel* atau elektrode yang diletakkan pada kepala partisipan, seperti terlihat pada Gambar 2.3.

Setiap elektrode dilambangkan dengan huruf dan angka. Huruf menandakan area dari kepala pada elektrode, semisal F ialah *Frontal Lobe* dan T ialah *Temporal lobe*. Angka genap menandakan bagian kanan dari otak dan angka ganjil menandakan bagian kiri otak. Gambar 2.4 menunjukkan penempatan dan penamaan dari 32 *channel* pada BCI.

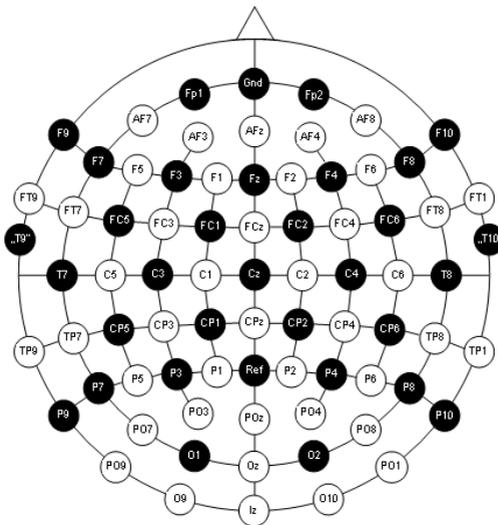
### **Database for Emotion Analysis using Physiological Signal (DEAP)**

*Database for emotion analysis using physiological signal* (DEAP) merupakan sebuah dataset untuk melakukan analisa psikologi yang berisi sinyal EEG, EOG, EMG, GSR, *Respiration belt*, *Plethysmograph*, dan temperatur tubuh. Selain itu, DEAP juga

menyertakan video wajah dari peserta. Dataset dari DEAP berupa *file* sinyal yang mempunyai ekstensi *Biosemi Data Format* (BDF).



**Gambar 2.3** Objek sebelum melakukan percobaan



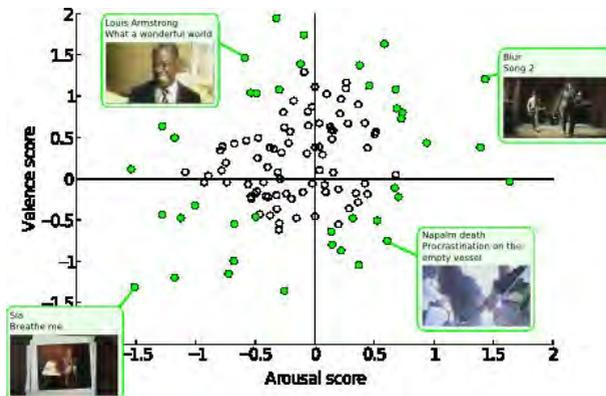
**Gambar 2.4** Peta *channel* pada BCI

Tim riset dari Universitas Tweente dan Universitas Geneva mengumpulkan 32 pelajar sebagai partisipan. Setiap partisipan akan mengikuti 40 *trial*, dimana setiap *trial* disuguhkan sebuah

*video* yang telah diberi nilai kelas emosi tertentu [9]. Pada tiap *trial*, sinyal dari partisipan akan disimpan. Sinyal yang didapat mempunyai *sampling rate* sebesar 512 Hertz. Setiap *video* pada *trial* melambangkan kelas dari region pada *circumplex model*, seperti yang terlihat pada Gambar 2.5. Informasi mengenai DEAP dapat dilihat di <http://www.eecs.qmul.ac.uk>.

### 2.3 Independent Component Analysis

*Independent Component Analysis* (ICA) merupakan sebuah metode *preprocessing* pada data sinyal dengan melakukan dekomposisi sinyal data kedalam bentuk yang *independent* untuk mendeteksi artefak yang menyebabkan *noise*.



**Gambar 2.5 Implementasi circumplex model pada video percobaan**

ICA telah terbukti efisien dalam memisahkan data dari artefak seperti kedipan mata, gerakan otot, atau denyut jantung [10]. Tujuan awal algoritma ini ialah sebagai pengenalan asal suara dari

berbagai sumber yang bercampur atau disebut *cocktail party problem*.

Pada masalah *cocktail party*, diibaratkan terdapat dua orang yang berbicara secara bersamaan di depan pengeras suara pada lokasi yang berbeda. Suara yang dihasilkan oleh dua pengeras suara tercampur menjadi dua rekaman sinyal suara yang berbeda yaitu  $x_1$  dan  $x_2$ , dimana  $x(t)$  merupakan amplitudo dan  $t$  ialah waktu. Rekaman sinyal suara mempunyai variabel pemberat (*weight*) yaitu suara asli dari kedua pembicara yang dinotasikan dengan  $s_1(t)$  dan  $s_2(t)$ . Persamaan (2.1) dan (2.2) menunjukkan keterkaitan antara sumber, variabel pemberat dan rekaman sinyal suara

$$x_1(t) = a_{11} * s_1 + a_{12} * s_2 \quad (2.1)$$

$$x_2(t) = a_{21} * s_1 + a_{22} * s_2 \quad (2.2)$$

dimana  $a_{11}, a_{12}, a_{21}$  dan  $a_{22}$  ialah parameter yang bergantung pada jarak antara pengeras suara dan pembicara. Misalnya, kita berada pada jarak tertentu dari kedua sumber suara, sehingga suara dari kedua sumber sudah bercampur. ICA digunakan untuk mendapatkan suara asli  $s_1$  dan  $s_2$  dengan hanya menggunakan suara rekaman  $x_1$  dan  $x_2$ , dengan cara mengestimasi  $a_{11}, a_{12}, a_{21}$  dan  $a_{22}$  berdasarkan informasi independensinya.

Sinyal elektrik dari aktivitas otak yang terekam dalam bentuk EEG mengandung rekaman sinyal listrik dari banyak lokasi aktifas otak yang telah tercampur menjadi satu. ICA dapat menentukan informasi menarik seperti mengekstrak sinyal elektrik mana yang ingin didapatkan berdasarkan sifat setiap sumber sinyal yang *independent* [3].

ICA memodelkan data dalam bentuk *linear*, seperti yang terlihat pada persamaan (2.3), dimana  $A$  merupakan *mixing matrik*,  $s$  merupakan *vector* yang berisi *source* atau *independent component* (IC) dan  $x$  merupakan sinyal hasil campuran

$$x = As \quad (2.3)$$

Untuk mencari sumber  $s$  maka diperkenalkan matrik  $W$  yaitu matrik *inverse* dari  $A$ , sehingga dapat dibentuk persamaan (2.4)

$$s = Wx \quad (2.4)$$

karena matrik  $W$  ataupun  $A$  tidak diketahui maka nilai dari matrik  $W$  perlu diestimasi. *Non-gaussian* merupakan tolak ukur untuk mencari  $W$ . Apabila nilai *non-gaussian* dari  $W$  dan  $x$  besar maka *source* akan lebih *independent*. Untuk nilai *non-gaussian* digunakan fungsi *negentropy* pada persamaan (2.5)

$$J(y) = [E\{xg(W^T x)\} - E\{g'(W^t x)\}]^2 \quad (2.5)$$

dimana  $g(x)$  merupakan sebuah fungsi *kuadratik*. Gambar 2.6 merupakan contoh sinyal EEG yang telah di-*preprocess* dengan ICA. Pada Gambar 2.6 sebelah kanan sinyal yang telah dikoreksi dengan ICA mengungkapkan bahwa sinyal pada elektrode T3 dan T4 mengandung artefak yang mungkin disebabkan oleh pergerakan otot (perkiraan didasarkan pada posisi elektrode di kulit kepala). Pada Gambar 2.7 terlihat bagaimana ICA melakukan dekomposisi sinyal EEG dengan memisahkan sinyal asli dari artefak [3].

## 2.4 *Wavelet Denoising*

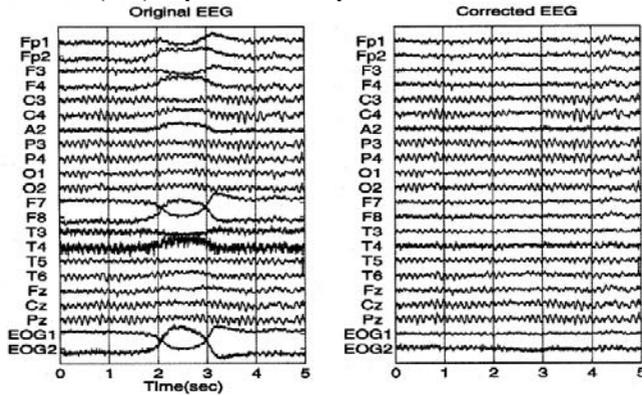
Deteksi dan diagnosis artefak menggunakan ICA terkadang masih terganggu oleh *noise* yang dapat menyembunyikan sinyal asli dan dapat merusak interpretasi dari data. Sehingga metode untuk menekan *noise* sangat krusial pada *preprocessing* sinyal EEG.

Salah satu metode yang paling sering digunakan saat ini ialah *wavelet denoising*. *Wavelet denoising* melakukan *preprocessing* data pada bidang *spatial*. Terdapat dua metode utama dalam melakukan *wavelet denoising* yaitu *Discrete Wavelet Transform* (DWT) dan *Stationary Wavelet Transform* (SWT) [11].

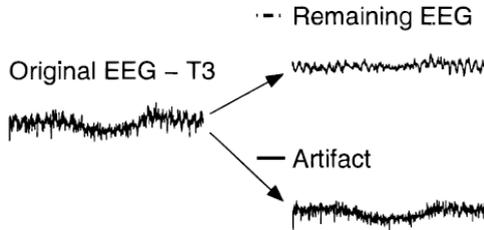
### 2.4.1 *Discrete Wavelet Transform*

*Discrete Wavelet Transform* (DWT) adalah metode yang digunakan untuk meningkatkan penekanan *noise*. Pada DWT data

yang dianalisa akan disaring melalui *Low-pass Filter* (LF) dan *High-pass Filter* (HF) seperti terlihat pada Gambar 2.8.



**Gambar 2.6 Hasil Koreksi Sinyal EEG Menggunakan ICA**

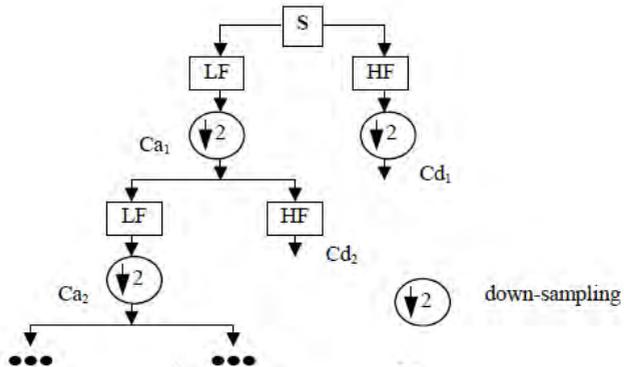


**Gambar 2.7 Dekomposisi Sinyal EEG dengan ICA**

Saat sinyal didekomposisi melalui LF dan HF serta diikuti dengan proses *down sampling*, menghasilkan konten dengan frekuensi rendah yang disebut aproksimasi dan konten dengan frekuensi tinggi yang disebut detail. Aproksimasi kemudian diproses lagi dengan HF dan LF untuk kemudian didekomposisi di *level* selanjutnya.

Pada DWT, metode *down sampling* memegang peranan kunci dari proses dekomposisi. *Down sampling* adalah proses untuk

mengurangi *ratio* dari sinyal dengan menghapus beberapa koefisien dari sinyal. *Down sampling* menghasilkan sinyal yang sepanjang setengah dari jumlah fitur awalnya.



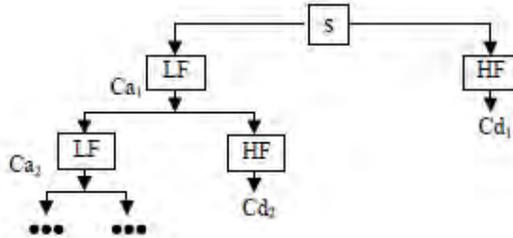
Gambar 2.8 Struktur dari DWT

#### 2.4.2 Stationary Wavelet Transform

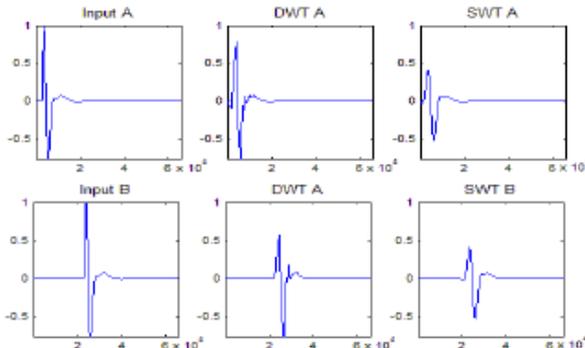
*Stationary Wavelet Transform* (SWT) mempunyai cara kerja yang hampir sama dengan DWT. Pada SWT metode *down sampling* dihilangkan sehingga jumlah semua koefisiennya pada semua *level* dekomposisinya tidak berubah, seperti terlihat pada Gambar 2.9. Hal ini membuat SWT mempunyai komputasi yang lebih cepat dibandingkan DWT dengan prosedur dari kedua metode tetap sama [5]. Redudansi pada SWT ini mempunyai efek yang baik untuk sinyal yang bergerak dalam persatuan waktu. Seperti terlihat pada Gambar 2.10, dimana gambar kiri atas menunjukkan sinyal awal dan dibawahnya menunjukkan sinyal yang hanya digeser pada sumbu-x.

Terlihat bahwa hasil *preprocessing* dengan DWT berbeda antara sinyal asli dan yang telah digeser. Sedangkan pada SWT, sinyal hasil *preprocessing* dari sinyal asli ataupun

yang sudah digeser tetap sama, hal ini disebabkan tidak adanya penghilangan koefisien pada sinyal.



**Gambar 2.9 Struktur dari SWT**



**Gambar 2.10 Perbedaan Hasil dari DWT dan SWT**

## 2.5 Multiple Discriminant Analysis

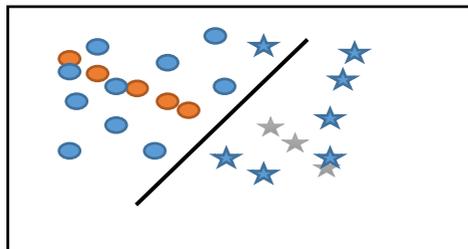
*Multiple Discriminant Analysis* (MDA) merupakan algoritma klasifikasi dari pengembangan algoritma *Linear Discriminant Analysis* (LDA), namun pada MDA kelas yang diklasifikasikan lebih dari dua. Tujuan utama dari MDA ialah menggunakan *hyperplane* untuk membedakan data dari kelas-kelas yang berbeda. *Hyperplane* yang dimaksud berupa fungsi *discriminant*. Fungsi *discriminant* ini akan melakukan klasifikasi

data berdasarkan variabel atau fitur mana yang paling *discriminative* antar kelas.

Sebagai contoh, semisal seorang peneliti ingin membuat sistem klasifikasi untuk membedakan makanan dari primata, burung, atau tupai. Untuk tujuan itu peneliti akan mengumpulkan data buah-buahan yang dimakan oleh masing-masing grup binatang tersebut. Kemudian dari setiap buah-buahan, dicari informasi variabel dari buah-buahan tersebut seperti bentuk, warna dan bau. Variabel-variabel ini lah yang disebut *independent variabel* [7].

Dengan MDA dapat ditentukan *independent variabel* mana yang paling berpengaruh sebagai pembeda atau paling *discriminative*. Selanjutnya, variabel ini digunakan sebagai acuan untuk menentukan buah mana yang akan dimakan oleh primata, burung atau tupai [7].

Metode klasifikasi dengan MDA mempunyai kemiripan dengan algoritma *Principal Component Analysis* (PCA) dimana data akan di proyeksi ke dalam basis baru dengan menggunakan *eigen vector*. Inti dari MDA ialah melakukan transformasi pada data prediktor ke bentuk yang paling baik dalam melakukan perbandingan diantara kelas. Setelah itu dilakukan klasifikasi berdasarkan *metric* tertentu seperti *euclidean distance*. Ilustrasinya dari MDA dapat dilihat pada Gambar 2.11, dimana data berbentuk bintang dan lingkaran berwarna biru merupakan variabel. Lingkaran berwarna oranye dan bintang berwarna putih merupakan *centroid* dari variabel yang diproyeksikan pada basis baru.



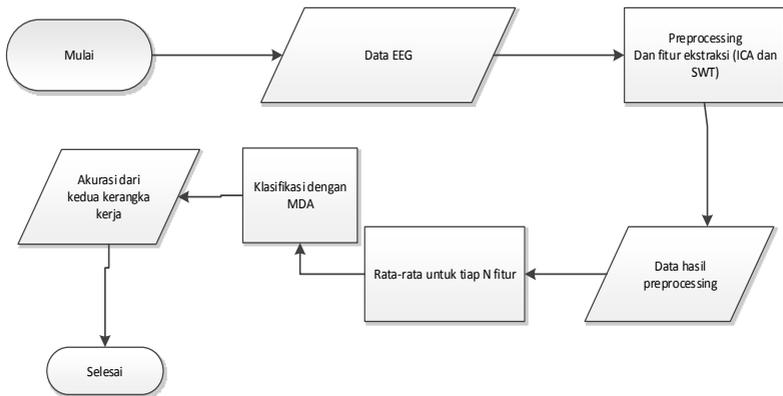
**Gambar 2.11 Hasil dari Multiple Discriminant Analysis**

## BAB III DESAIN

Pada bab ini akan dijelaskan perancangan program yang dibuat. Perancangan akan dibagi menjadi dua proses utama yaitu proses untuk mengolah data mentah yang nantinya akan dipakai sebagai data uji serta mengolah algoritma diskriminan analisis untuk diterapkan pada data yang telah dipilih fiturnya. Pada bab ini akan dijelaskan gambaran umum dalam bentuk *flowchart*. Selanjutnya untuk penjelasan lebih detail akan disajikan dalam *pseudocode*.

### 3.1 Desain Metode Secara Umum

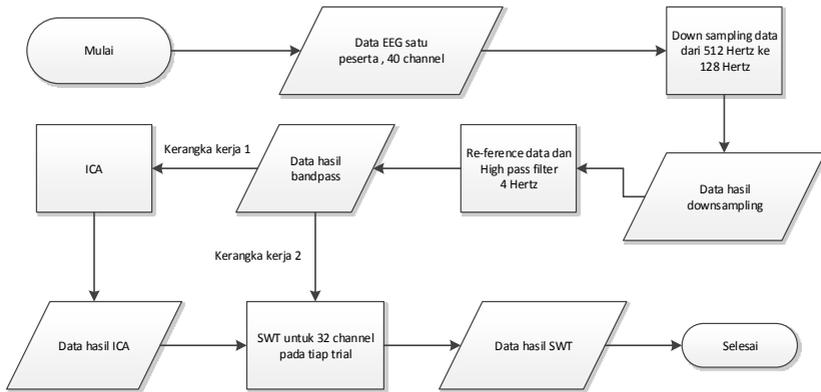
Secara umum terdapat tiga proses utama yaitu proses *preprocessing* untuk mengurangi *noise* dengan ICA dan SWT. Setelah itu, N fitur dari data dirata-ratakan kemudian diklasifikasi dengan MDA untuk mendapatkan akurasi dari kedua kerangka kerja, seperti yang terlihat pada Gambar 3.1.



**Gambar 3.1 Metode secara umum**

### 3.2 Preprocessing data

Preprocessing data dimaksudkan untuk mengubah data menjadi bentuk yang tepat untuk proses selanjutnya yaitu klasifikasi, serta mengurangi *noise* dari data sinyal EEG. Gambar 3.2 merupakan gambaran mengenai *preprocessing* data yang lebih rinci.



**Gambar 3.2 Kerangka kerja preprocessing**

#### 3.2.1 Independent Component Analysis

Setelah data dibagi bagi per-*trial*, dilakukan *Independent Component Analysis* (ICA) pada data untuk tiap *trial*. Hasil dari ICA berupa data dengan dimensi  $40 \times 40 \times$  banyak fitur per-*channel*, dimana terdapat 40 *trial* dan pada tiap *trial* terdapat 40 *channel*.

Pada dasarnya ICA akan mencari *Independent Component* (IC) pada data dengan merombak sumber dan *noise* pada data. Seperti yang telah dijelaskan sebelumnya bahwa ICA memerlukan matriks  $W$  sebagai *estimator* untuk menentukan *independent*

*componen*, dimana matrik W ditentukan dengan fungsi *negentropy*. Persamaan (3.1) dan persamaan (3.2) merupakan fungsi yang telah dijelaskan pada bagian 2.3. Berikut merupakan *pseudocode* untuk mencari matrik W.

$$g(u) = \tanh(a_i u) \quad (3.1)$$

$$g'(u) = u \exp\left(-\frac{u^2}{2}\right) \quad (3.2)$$

dimana  $a_1$  ialah bilangan konstan random dengan nilai  $1 < a_1 < 2$ .

Berikut ini merupakan pseudocode dari ICA

Pseudocode ICA	
1	Ambil tiap satuan sinyal x
2	/* Pilih <i>initial vector</i> w secara acak */
3	w= random()
4	wb=w
5	/* uji konvergensi matriks */
6	Selama w.dot(w.T) != 0
7	wb=w
8	$wb = E\{xg(w^T x)\} - E\{g'(w^T x)\}w$
9	wb=wb/sqrt(wb)

Pada baris ke 6, konvergen berarti matrik w yang sebelumnya dan matrik w yang baru, mempunyai arah yang sama. Konvergen dicapai jika *dot product* antara keduanya bernilai satu.

Setelah mendapatkan *vector* w, maka *vector* w dapat digabungkan menjadi *mixing matrix* W. *Independent component* didapatkan dengan mengalikan sinyal masukan x dengan matrik W seperti terlihat pada persamaan (3.3)

$$s = W * x \quad (3.3)$$

dimana s merupakan *Independent component* dari x.

### 3.2.2 Stationary Wavelet Transform

Pada Tugas akhir ini *Stationary Wavelet Transform* (SWT) digunakan untuk mengekstraksi fitur dan menyaring artefak seperti gerakan otot. Ide utama dari SWT ialah untuk menyaring sinyal ke

*sub band* tertentu dengan menggunakan *high pass filtering* dan *low pass filtering*. Dengan melakukan SWT pada tiap *level* dekomposisi didapatkan sinyal dengan frekuensi setengah dari asalnya. Hal ini dilakukan agar frekuensi *band* dari sinyal bisa dipisahkan. Alur dekomposisi SWT dengan lima *level* dapat dilihat pada Gambar 3.3 dan Tabel 3.1 berisi informasi frekuensi dari detail untuk tiap *level* dekomposisi.

**Tabel 3.1 Pemisahan band pada SWT**

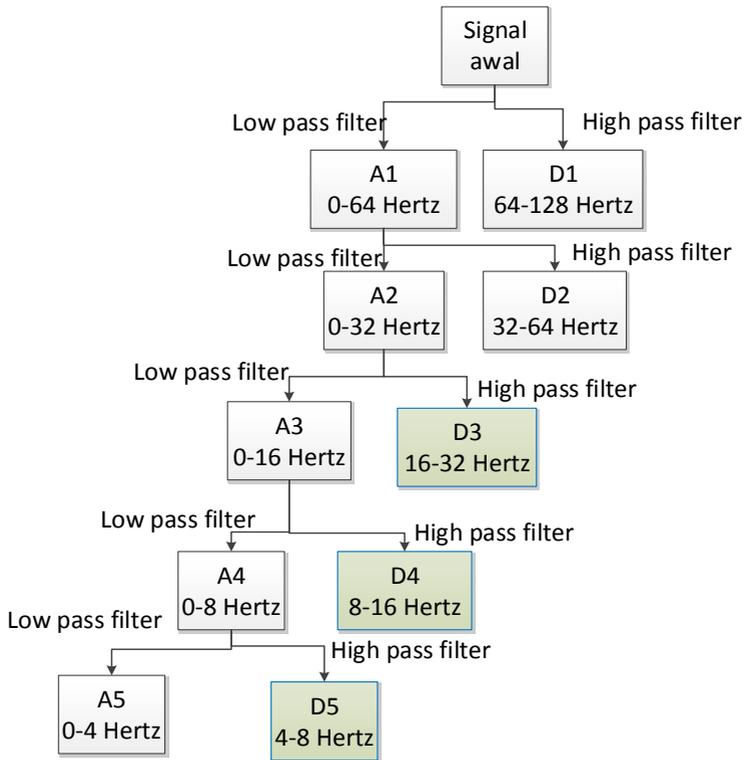
Jarak frekuensi	Level dekomposisi
64-128	1
32-64	2
16-32	3
4-16	4
4-8	5

Hasil dari SWT pada tiap *level* berupa dua buah komponen yaitu aproksimasi dan detail. SWT dilakukan untuk tiap *channel* per-*trial*. Komponen yang diambil ialah detail pada *level* 5, 4 dan 3. Setiap detail pada tiap *level* disimpan. Kemudian semua detail pada semua *channel* di rata-ratakan untuk tiap frekuensi pada tiap *level*-nya. Alur kerja dari SWT dapat dilihat pada Gambar 3.4. Setelah tahap ini selesai, setiap *trial* akan mempunyai fitur sebanyak 24.192.

Berikut merupakan *pseudocode* dari SWT

Pseudocode SWT	
1	/* input : x data sinyal dengan elemen $e_1, e_2 \dots e_N$
2	n_level : banyak level
3	output : $C_T$ = komponen Aproksimasi dan Detail pada tiap level
4	$[[C_A, C_D], [C_A, C_D]]$
5	*/
6	$C_T = []$ /* list kosong */
7	for j in n_level
8	/* low pass filtering */
9	$C_A = \frac{e_1+e_2}{\sqrt{2}}, \frac{e_2+e_3}{\sqrt{2}}, \frac{e_3+e_4}{\sqrt{2}} \dots \frac{e_{N-1}+e_N}{\sqrt{2}}$

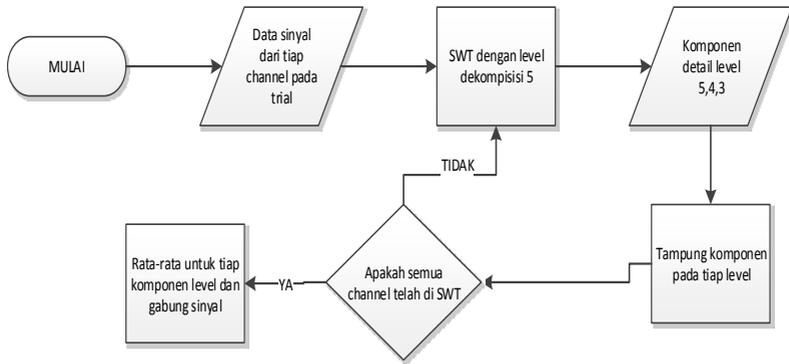
10	/*high pass filtering */
11	$C_D = \frac{e_1-e_2}{\sqrt{2}}, \frac{e_2-e_3}{\sqrt{2}}, \frac{e_3-e_4}{\sqrt{2}} \dots \frac{e_{N-1}-e_N}{\sqrt{2}}$
12	$C_T.push(C_A, C_D)$
13	$x=C_A$



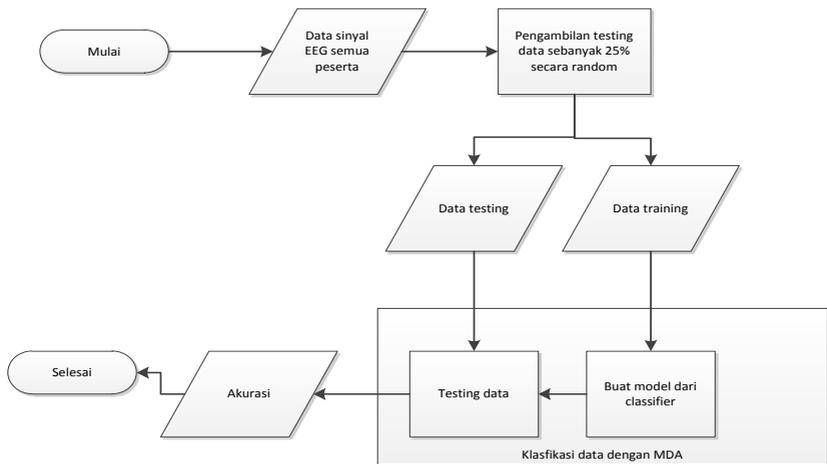
**Gambar 3.3** Alur dekomposisi SWT

### 3.3 Klasifikasi data

Data yang telah di-*preprocessing* dari kedua kerangka kerja kemudian diklasifikasikan untuk mencari akurasi. Gambar 3.5 memperlihatkan bagaimana alur dari proses klasifikasi.



**Gambar 3.4 Alur kerja SWT**



**Gambar 3.5 Proses klasifikasi**

### ***Multiple Discriminant Analysis***

Setelah data selesai di-*preprocess*, dihasilkan data pada kedua kerangka kerja yang kemudian digunakan membuat model

*classifier* dengan metode MDA. Model yang dibentuk oleh MDA berupa sejumlah *vector centroid* yang berjumlah sesuai dengan jumlah kelas, yakni berjumlah empat. *Vector centroid* sendiri sebenarnya merupakan proyeksi rata-rata pada tiap kelas pada basis baru, yang dibentuk oleh *eigen vector*.

Tujuan dari proyeksi data ke basis baru untuk membuat data menjadi lebih diskriminatif untuk proses klasifikasi. Seperti dalam *Principal Component Analysis*, jumlah fitur yang diproyeksikan dapat ditentukan dengan menggunakan *eigen value* dari tiap fitur, dengan cara memilih *eigen value* yang paling besar. Berikut ini merupakan *pseudocode* dari *training* atau pembentukan model pada MDA.

Pseudocode Training model MDA	
1	/* input : $[(data)]_{train}$ ialah matrik dari data yang akan diuji yang
2	mengandung vector dari data dengan kelas tertentu
3	kelas : ialah vector berdimensi satu yang perbarisnya menunjukkan kelas dari $[(data)]_{train}$
4	output : $projected\_centroid$ : berupa koefisien dari model yang berisi centroid dari tiap kelas */
5	/* data dengan dimensi $M \times N$ */
6	$x = [(data)]_{train}$
7	/* kelas pada data */
8	kelas = [1..4]
9	/* rata-rata tiap kolom pada data $x$ */
10	$mean\_vec = \text{meand}(x, \text{axis}=0)$
11	/* mencari within class matriks $Sw$ */
12	$Sw = \text{zeros}(N \times N)$ /*matrik kosong berukuran $N \times N$ */
13	/* untuk tiap kelas dan vector rata-rata lakukan iterasi */
14	for $cl$ and $\mu$ in kelas, $mean\_vector$
15	/* ambil data untuk tiap kelas $cl$ */
16	$x\_k = x[\text{kelas}==cl]$
17	/* lakukan iterasi pada tiap baris data $x\_k$ */
18	for row in $x\_k$
19	/*hitung kovariance dari tiap row pada kelas ke $cl$ */
20	$s\_k += (\text{row}-\mu).\text{dot}((\text{row}-\mu).\text{tranpose})$

21	/* sisipkan data covariance tiap kelas pada matrix S */
22	Sw+=s k
23	/* mencari between class matrix Sb */
24	/*mencari seluruh mean (grand mean) pada tiap fitur */
25	grand_mean=mean(mean_vec,axis=0)
26	/* inialisasi matriks kosong berukuran NxN*/
27	Sb=zeros(N,N)
28	/* dapatkan index i dan rata-rata vector pada index ke-i */
29	for cl,mu in kelas,mean_vector
30	n=x[kelas==cl].dimensi[0] /* dapatkan dimensi baris dari data */
31	n=(mu-grand_mean).dot((mu-grand_mean).T)
32	/* cari eigen value dan eigen vector dari matriks $[(Sw)]^{(-1)}.Sb$ */
33	eigValue,eigVector=eig(Sb.Sw+Sb)
34	/*lakukan sorting secara decreasing (lebih besar didahulukan) pada
35	eigen value dan eigen vector berdasarkan nilai eigenvalue*/
36	eigValue,eigeVector=sorted((eigenvalue,eigenvector),key=eigenvalue,decrease=true)
37	/*vector eigenvector menjadi element pada discriminant matrix */
38	discriminant_mat=array(eigenvector)
39	/* proyeksikan data rata-rata pada basis baru dengan melakukan dot product dengan matrik discriminant */
40	projected_centroid=mu.dot(discriminant_mat)

Setelah model selesai di-*training*, kemudian diambil data uji sebanyak 25% dari seluruh data atau sejumlah 220 data. Selanjutnya data uji akan diprediksi dengan model yang telah di *training*. Untuk memprediksi data, data uji akan diproyeksikan pada matrik *eigen vector* yang didapat pada saat *training*. Kemudian data hasil proyeksi ini dibandingkan dengan *centroid* dari semua kelas yang telah diproyeksi sebelumnya.

Kelas pada data didapatkan dengan menghitung *euclidean distance* dari data uji dengan *centroid* dari tiap kelas. Setelah itu dicari kelas yang mempunyai *euclidean distance* terkecil. Berikut merupakan pseudocode dari *testing* data

Psedocode Testing data
------------------------

1	/*
2	input : $data_{tes}$ ialah data berupa matriks yang akan diuji,
3	yang tersusu dari vector data yang kelasnya akan diprediksi
4	output : kelas berupa vector yang berisi kelas dari tiap vector data
5	*/
6	/*x_test merupakan data yang akan diuji */
7	x_test= $data_{test}$
8	projected_data=x_test.dot(discriminant_mat)
9	/* ambil setiap data testing pada projected data pada index ke- i*/
10	for i,data in projected data
11	/* untuk menampung euclidean distance */
12	dict={}
13	/*ambil setiap centroid pada projected centroid */
14	for centroid in projected centroid
15	ed=0
16	/*cari euclidean distance dari data test dan centroid */
17	ed = sqrt(sum(sqr(datum-dat_centroid) in data,centroid )
18	dict.add(ed)
19	dict=sorted(dict)
20	/* dapatkan kelas prediksi untuk data ke-i */
21	kelas_pred[i]=dict.getMin.idx

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi utama. Perlu diperhatikan bahwa *preprocess* ini dilakukan untuk sinyal tiap satu partisipan, dan akhirnya seluruh hasil dari *preprocess* dari semua partisipan akan digabungkan untuk diklasifikasi.

### **4.1 Lingkungan Implementasi**

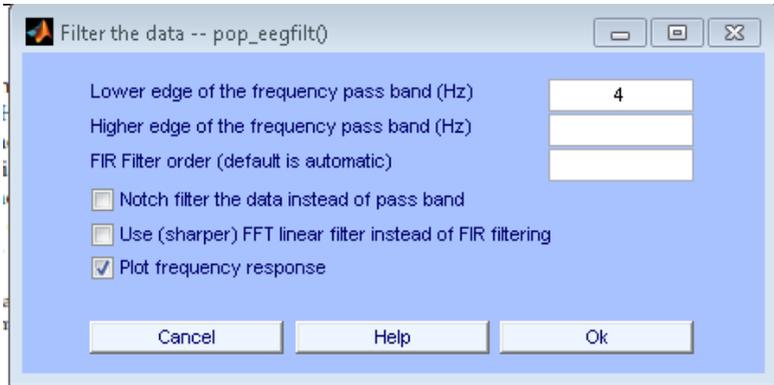
Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah bahasa pemrograman *Python* dengan versi *compiler 2.7*, *library numpy*, *scipy*, *sklearn* dan perangkat lunak *EDFBrowser* dan *EEGLAB* untuk melakukan *preprocessing* yang diinstal pada sistem operasi Linux Ubuntu 13.10 dan Windows 7.

### **4.2 Implementasi *High Pass Filter***

Sinyal yang mempunyai frekuensi dibawah 4 *Hertz* dianggap sebagai *electrical noise*. Oleh karena itu dilakukan *High pass filter* dengan *cut off* frekuensi sebesar 4 *Hertz*. *EEGLAB* digunakan untuk melakukan *High pass filter* seperti terlihat pada Gambar 4.1

### **4.3 Implementasi Fungsi ICA**

Fungsi ICA digunakan untuk mendapatkan matrik *Independent Component* pada data. Fungsi ini juga merupakan implementasi dari persamaan yang telah dijelaskan pada bab 2.3 dan pseudocode ICA terdapat pada bab 3.2.2.



**Gambar 4.1 High pass filter dengan EEGLAB**

Untuk mengimplementasikan ICA digunakan fungsi ICA pada EEGLAB. Proses ICA menggunakan 40 *channel*, yaitu 32 *channel* EEG dan 6 *channel* EOG.

#### 4.4 Implementasi Fungsi SWT

Fungsi SWT digunakan untuk mendapatkan komponen aproksimasi dan detail pada data. SWT dilakukan pada sinyal tiap *channel* data. Algoritma SWT diimplementasikan oleh fungsi SWT pada *library* *pywavelet* dengan parameter pertama yaitu data sinyal satu dimensi, parameter kedua merupakan *mother function wavelet* dan parameter terakhir *level* dekomposisi.

Pada percobaan ini *level* dekomposisi yang dipilih ialah *level 5*, dan *mother wavelet* yang digunakan ialah fungsi *haar*. Dengan menggunakan 5 *level* dihasilkan lima buah aproksimasi dan detail pada tiap elemen. Karena untuk melakukan SWT jumlah fitur dari tiap sinyal harus bernilai genap, oleh karena itu diambil

data dengan ukuran 8064, yang berarti 1 menit perekaman sinyal EEG. Seperti yang telah di jelaskan pada bab 2.3.3, komponen detail dari level 5, 4 dan 3 diambil digunakan sebagai fitur dari data. Contoh implementasi SWT dapat dilihat pada Kode sumber 4.1.

1	hasilSWT=[]
2	for id,eachTrial in enumerate(hasilIca):
3	hasilPerTrial=[]
4	for eachChannel in eachTrial:
5	[(cA5,cD5),(cA4,cD4),(cA3,cD3),(cA2,cD2),(cA1,cD1)]=pywt.swt(eachChannel,'haar',5)
6	list_cD5.append(cD5)
7	list_cD4.append(cD4)
8	list_cD3.append(cD3)
9	t_list=[]
10	t_list.extend(averageSignal.averageSignal2(list_cD5))
11	t_list.extend(averageSignal.averageSignal2(list_cD4))
12	t_list.extend(averageSignal.averageSignal2(list_cD3))
13	hasilSWT.append(t_list)

**Kode sumber 4.1 Stationary Wavelet Transform**

## 4.5 Implementasi Kelas MDA

Fungsi MDA adalah fungsi yang digunakan untuk melakukan proses klasifikasi pada data. Terdapat dua proses yang dilakukan pada MDA yaitu *training* data dan *testing* data. Pada tahap implemmentasi, kedua proses ini dipisah menjadi fungsi yang berbeda.

### 4.5.1 Fungsi *Training*

*Training* data dilakukan dengan melakukan proyeksi *centroid* pada tiap kelas, seperti yang telah dijelaskan pada 3.2.6. *Pseudocode* dari *training* data dapat dilihat pada Kode sumber 4.2

1	# X data , Y class
2	def Train(self,X,Y):
3	# dapatkan semua kelas
4	label_kelas = np.unique(Y)
5	jum_kelas = len(label_kelas)
6	#dapatkan dimensi dari data
7	datanum,dim = X.shape
8	#datapatkan rata-rata dari setiap data perkolom
9	totalMean = np.mean(X,0)
10	# partisi setiap data
11	partition = [np.where(Y==label)[0] for label in label_kelas]
12	rataKelas = [(np.mean(X[idx],0),len(idx)) for idx in partition]
13	
14	# menghitung within-class scatter matrix
15	W = np.zeros((dim,dim))
16	for idx in partition:
17	W += np.cov(X[idx],rowvar=0)*len(idx)
18	# menghitung between-class scatter matrix
19	B = np.zeros((dim,dim))
20	for mu,ukuran_kelas in rataKelas:
21	offset = mu - totalMean
22	B += np.outer(offset,offset)*ukuran_kelas
23	# ew=eigen value, ev= eigen vector
24	ew, ev = linalg.eig(B,W+B)
25	sorted_eg_ev=sorted(enumerate(ew), key=operator.itemgetter(1), reverse=True)
26	selected_ind = [ind for ind,val in sorted_eg_ev[:jum_kelas]]
27	self.discriminant_vector = ev[:,selected_ind]
28	self.projected_centroid = [np.dot(mu,self.discriminant_vector) for mu,ukuran_kelas in rataKelas]

**Kode sumber 4.2 *Training* pada MDA**

#### 4.5.2 Fungsi *Testing*

Testing dari data dilakukan dengan memproyeksikan data uji pada basis *eigen vector* yang didapat pada tahap *training* seperti terlihat pada Kode sumber 4.3 baris 2. Lalu dicari jarak terpendek dari data dengan semua *centroid* dengan *euclidean distance* dari tiap kelas seperti pada baris ke 4, untuk menentukan kelas dari data uji.

1	<code>def Test(self,X):</code>
2	<code>    projected_data = np.dot(X,self.discriminant_vector)</code>
3	<code>    projected_centroid = self.projected_centroid</code>
4	<code>    # cari euclidian distanceance dari data masukan dan rata"</code> <code>centroid perkelas</code>
5	<code>    distance = [linalg.norm(data-centroid) for data in</code> <code>projected_data for centroid in projected_centroid]</code>
6	<code>    Y_raw=np.reshape(np.array(distance),(len(X),len(projected_centroid)))</code>
7	<code>    return Y_raw.argmax(axis=1)</code>

**Kode sumber 4.3 *Testing* pada MDA**

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji kinerja serta analisa setiap pengujian.

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

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji kerangka kerja klasifikasi emosi pada data sinyal EEG dengan ICA, Wavelet denoising dan MDA. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak. Lingkungan uji coba yang dijelaskan sebagai berikut:

1. Perangkat keras
  - a. Prosesor: Intel® Core™ i3-2430M CPU @ 2.40GHz
  - b. Memory(RAM): 8,00 GB
  - c. Tipe system: 64-bit sistem operasi
2. Perangkat lunak
  - a. Sistem operasi: Windows 7 Home Premium dan Linux ubuntu 13.10
  - b. EEGLAB
  - c. EDFBrowser
  - d. Perangkat pengembang: Pycharm IDE versi 10.

#### **5.2 Data DEAP**

Data yang digunakan dalam penelitian ini berupa data sinyal EEG yang didapat dari DEAP. Data sebanyak 32 *file*, yang berekstensi BDF. Setiap *file* berisi sinyal dari tiap partisipan, dimana 1-22 orang didapatkan dari partisipan Universitas Tweente, sedangkan sisanya 23-32 didapatkan dari partisipan Universitas Geneva.

Sinyal mengandung 48 *channel*, dimana 32 *channel* merupakan sinyal EEG, sedangkan sisanya merupakan sinyal EOG, GSR, ERG, repirsasi, temperatur tubuh dan status *channel*. Tabel A.1 merupakan deksripsi dari data sinyal. Kolom *channel name* menunjukkan letak dan nama *channel*. Kolom keempat dan kelima menunjukkan transisi *channel* karena adanya perbedaan posisi *channel* pada kedua tempat.

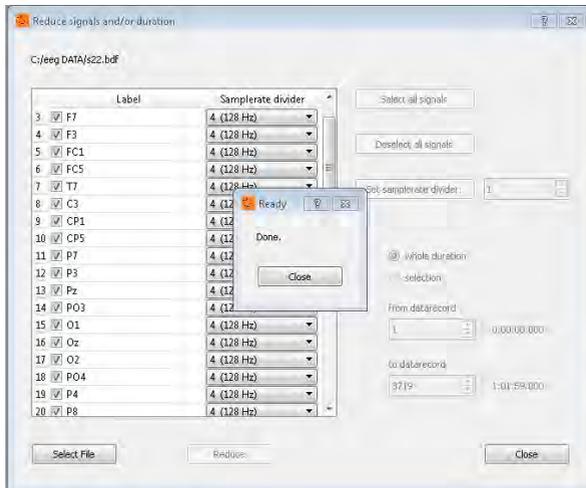
Terdapat dua buah *file* yang berisi kelas dari tiap *trial* yaitu kelas penilaian dari peneliti (VAQ\_Online) dan penilaian dari peserta (VAQ\_estimate), namun pada percobaan ini kelas yang digunakan ialah kelas dengan penilaian dari peneliti. Informasi kelas tersedia pada *file* video\_list.csv pada kolom VAQ\_Online. Pada Tabel A.2 diperlihatkan kelas yang sudah diurutkan berdasarkan kolom *experiment\_id*.

Dalam mengawali proses pengerjaan program, harus disiapkan terlebih dahulu data yang akan digunakan dalam proses pengerjaan. Data awal yang didapatkan adalah 32 data *file* berektensi BDF. Setiap *file* merepresentasikan satu orang objek pengujian yang berisi sinyal EEG dan tambahan *channel*. Semua urutan *channel* dapat dilihat pada Tabel A.1. Data terdiri dari 40 *channel* dengan *sampling rate* 512 hertz.

### 5.3 Pemrosesan Data

Karena keterbatasan memori untuk melakukan proses pada data, sehingga data perlu diturunkan *sampling rate*-nya dari 512 hertz ke 128 hertz dengan menggunakan perangkat lunak *EDFBrowser*, seperti yang terlihat pada Gambar 5.1. Dengan melakukan pengurangan *sampling rate*, atribut data akan menjadi satu perempat dari ukuran semula.

Hasil dari *EDFBrowser* ialah 22 *file* sinyal berformat BDF, dimana setiap *file* berisi sinyal untuk setiap peserta. Setiap data sinyal mengandung 40 *channel* berupa sinyal EEG dan EOG. Hasil dari pengurangan *sampling rate* sebanyak 22 data yang dapat dilihat pada Gambar 5.2. Tiap *file* berisi sinyal untuk tiap partisipan.



**Gambar 5.1** Reduksi sampling rate dengan EDFBrowser

Name	Date modified	Type	Size
s01_reduced.bdf	1/4/2015 8:15 AM	BDF File	58,046 KB
s02_reduced.bdf	1/4/2015 8:21 AM	BDF File	55,556 KB
s03_reduced.bdf	1/4/2015 8:23 AM	BDF File	58,286 KB
s04_reduced.bdf	1/4/2015 8:25 AM	BDF File	49,528 KB
s05_reduced.bdf	1/4/2015 8:43 PM	BDF File	58,753 KB
s06_reduced.bdf	1/4/2015 8:29 AM	BDF File	53,801 KB
s07_reduced.bdf	1/4/2015 8:31 AM	BDF File	53,576 KB
s08_reduced.bdf	1/4/2015 8:37 PM	BDF File	52,766 KB
s09_reduced.bdf	1/4/2015 8:35 AM	BDF File	56,516 KB
s10_reduced.bdf	1/4/2015 8:38 AM	BDF File	52,693 KB
s11_reduced.bdf	1/4/2015 8:39 AM	BDF File	62,426 KB
s12_reduced.bdf	1/4/2015 8:41 AM	BDF File	52,481 KB
s13_reduced.bdf	1/4/2015 8:43 AM	BDF File	55,676 KB
s14_reduced.bdf	1/4/2015 8:44 AM	BDF File	59,066 KB
s15_reduced.bdf	1/4/2015 8:46 AM	BDF File	61,000 KB
s16_reduced.bdf	1/4/2015 8:48 AM	BDF File	55,256 KB
s17_reduced.bdf	1/4/2015 8:49 AM	BDF File	51,700 KB
s18_reduced.bdf	1/4/2015 8:51 AM	BDF File	54,423 KB
s19_reduced.bdf	1/4/2015 8:52 AM	BDF File	56,203 KB
s20_reduced.bdf	1/4/2015 8:54 AM	BDF File	52,646 KB
s21_reduced.bdf	1/4/2015 8:55 AM	BDF File	53,276 KB
s22_reduced.bdf	1/4/2015 8:57 AM	BDF File	55,796 KB

**Gambar 5.2** Hasil pengurangan sampling rate

Satu persatu data BDF diproses menggunakan EEGLAB untuk melakukan *re-reference* dengan rata-rata dan *high pass filter* sebesar 4 *Hertz*. Hasil dari tahap ini dapat dilihat Gambar 5.3. Kemudian data diproses menggunakan ICA untuk mendapatkan

*Independent Component (IC)*. Hasil dari data yang telah di-ICA terlihat pada Gambar 5.4.

Data hasil dari ICA kemudian disimpan kedalam bentuk *file text* dengan jumlah 22 *file*. *File-file* ini kemudian dibaca oleh python untuk dipotong kedalam 40 *trial*. Pemotongan dilakukan dengan membagi fitur kedalam 40 bagian. Data hasil pemotongan kemudian disimpan kebentuk *serialisasi* objek dalam bentuk *pickle*. Selanjutnya dilakukan SWT pada data untuk tiap *trial* dengan *level* dekomposisi bernilai lima. Hasil dari SWT pada salah satu *level* dekomposisi dapat dilihat pada Gambar 5.5.

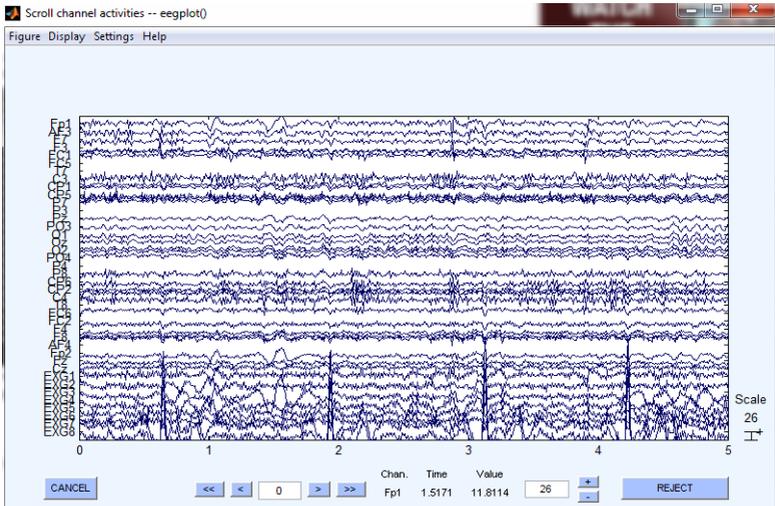
Detail tiap *channel* pada *level* 5, 4 dan 3 kemudian di rata-ratakan untuk tiap satu *trial*. Contoh hasil dari rata-rata detail bisa dilihat pada Gambar 5.6. Hasil rata-rata ini kemudian digabung menjadi satu dengan urutan sinyal level 3, 4 dan 5.

Fitur satu *trial* pada tiap orang kemudian disatukan, menghasilkan data dengan jumlah 880 baris dan 24.192 fitur. Untuk data uji digunakan 25% dari total data, yaitu sebanyak 220 data dan sisanya 660 data digunakan untuk *training model*. Pengambilan data uji dilakukan secara acak. Pada Gambar 5.7 disajikan contoh dari data asli pada tiap kelas.

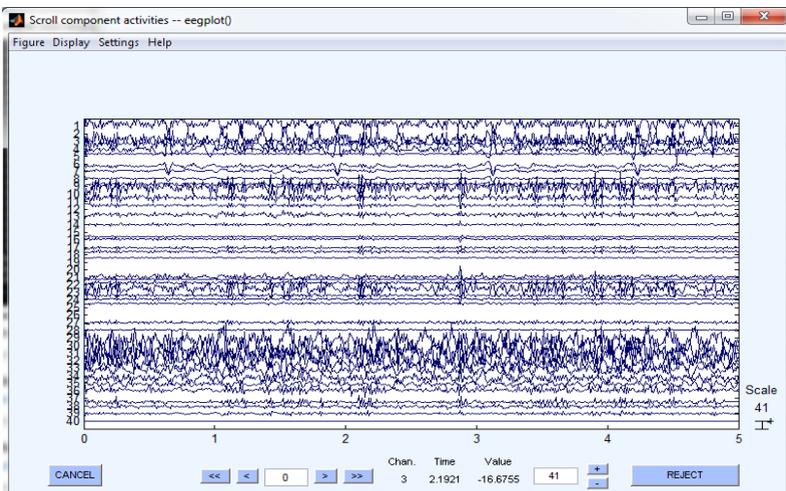
Uji coba ini dilakukan untuk menguji apakah fungsionalitas program telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya. Karena data yang direkam di Tweente mempunyai urutan *channel* yang berbeda dengan data yang direkam pada Geneva, maka digunakan data dari Tweente sebab data lebih banyak, yaitu sebanyak 22 orang atau 880 baris data.

## 5.4 Skenario dan Evaluasi Pengujian

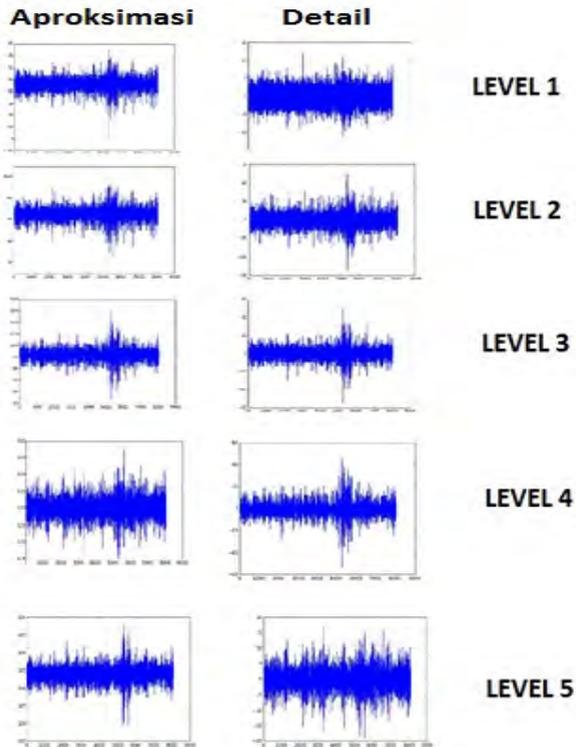
Uji coba akan didasarkan pada beberapa skenario untuk menguji kesesuaian dan kinerja aplikasi. Pada skenario uji coba ini, fitur pada data dirata-ratakan tiap N fitur. Klasifikasi dilakukan dengan 10-*fold cross validation*.



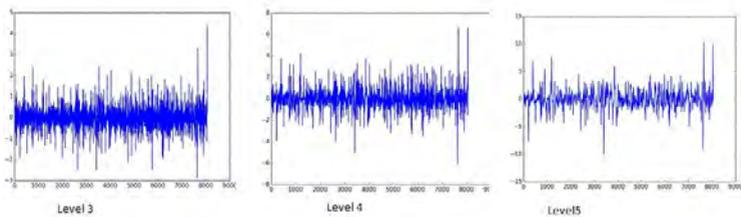
**Gambar 5.3** Plot data setelah re-reference dan high pass filter



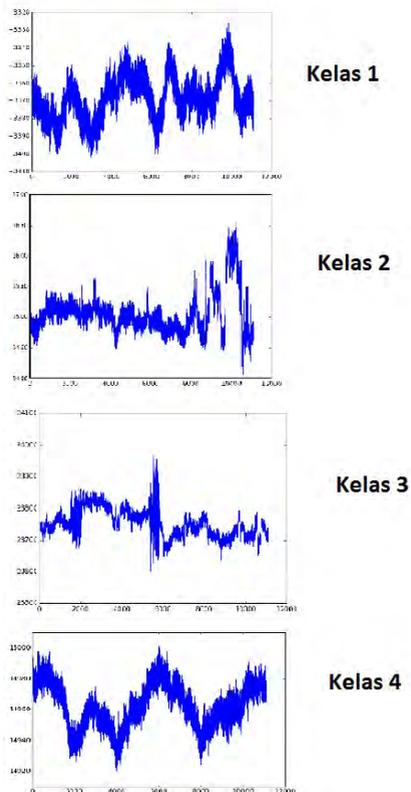
**Gambar 5.4** Plot data setelah di ICA



**Gambar 5.5 Hasil dekomposisi dari SWT**



**Gambar 5.6 Sinyal hasil rata-rata semua channel pada satu trial**



**Gambar 5.7** Sinyal asli untuk tiap kelas data

Pada Skenario dilakukan pengujian dengan menggunakan dua buah kerangka kerja yaitu:

1. Kerangka kerja 1 : *preprocess* dengan menggunakan ICA dan SWT.
2. Kerangka kerja 2 : *preprocess* dengan menggunakan SWT.

Dengan detail skenario sebagai berikut :

1. Skenario uji coba 1, pada kerangka kerja 1 dengan rata-rata setiap 50 fitur.
2. Skenario uji coba 2, pada kerangka kerja 1 dengan rata-rata setiap 100 fitur.
3. Skenario uji coba 3, pada kerangka kerja 1 dengan rata-rata setiap 500 fitur.
4. Skenario uji coba 4, pada kerangka kerja 2 dengan rata-rata setiap 50 fitur.
5. Skenario uji coba 5, pada kerangka kerja 2 dengan rata-rata setiap 100 fitur.
6. Skenario uji coba 6, pada kerangka kerja 2 dengan rata-rata setiap 500 fitur.

Tujuan dari pengujian dengan menggunakan dua kerangka kerja untuk membuktikan bahwa ICA efektif untuk mengurangi *noise* pada data DEAP. Pada Tugas Akhir ini kelas dibagi menjadi empat dimensi berdasarkan *circumplex model* emosi dari Russel, yaitu :

1. Dimensi pertama menandakan marah, menderita dan kesukaran.
2. Dimensi kedua menandakan senang dan bergembira.
3. Dimensi ketiga menandakan bosan, sedih dan lelah.
4. Dimensi keempat menandakan santai dan lega.

Akurasi data didapatkan dengan persamaan 5.1,

$$\frac{\text{Jumlah Prediksi benar}}{\text{Jumlah data uji}} * 100\% \quad (5.1)$$

Akurasi dihitung pada tiap *fold*. Tiap kelas prediksi dibandingkan dengan kelas asli dari data uji. Jika kelas prediksi data sama dengan kelas asli, maka nilai prediksi benar ditambah satu. Data yang digunakan berjumlah 880 data dengan jumlah fitur sebanyak 24.192. Tiap 40 baris data secara berurutan menandakan 40 *rial* dari partisipan ke-*i*. Kelas dari tiap *trial* didapatkan melalui *file* metadata, *video\_list.csv* dan *participant\_rating.csv*.

#### 5.4.1 Skenario dan evaluasi Uji Coba 1

Skenario uji coba 1 adalah pengujian aplikasi menggunakan kerangka kerja satu, yaitu *preprocessing* ICA dan SWT dengan melakukan rata-rata setiap 50 fitur. Gambar 5.8 menunjukkan plot dari data asli sebelum dilakukan rata-rata fitur. Garis horizontal menunjukkan indek dari data dan garis vertikal menunjukkan *amplitudo* dari data. Gambar 5.9 merupakan sinyal hasil rata-rata tiap 50 fitur. Sinyal asli dari data mempunyai fitur sebanyak 24.192. Data kemudian dirata-rata per-50 fitur menghasilkan data dengan 484 fitur.

Data hasil rata-rata per-N fitur kemudian diklasifikasi. Hasil keluaran dari program dapat dilihat pada Gambar 5.12. Pada Gambar 5.12 terlihat akurasi dari sepuluh kali *fold*, dengan akurasi rata-rata sebesar 62% dan akurasi terbaik sebesar 64%. Pada Gambar 5.22 diperlihatkan gambar sinyal hasil prediksi pada tiap kelas untuk skenario 1.

#### 5.4.2 Skenario dan evaluasi Uji Coba 2

Skenario uji coba 2 adalah pengujian aplikasi dengan menggunakan kerangka kerja pertama dan melakukan rata-rata pada setiap 100 fitur. Data awal yang dipakai mempunyai fitur sebanyak 24.192 yang dapat dilihat pada Gambar 5.8.

Pada Gambar 5.10 ditunjukkan data yang telah dirata-ratakan setiap 100 fitur, dengan jumlah fitur setelah dirata-ratakan sebanyak 242 fitur. Data hasil rata-rata per-N fitur kemudian diklasifikasi. Hasil keluaran dari program ditunjukkan pada Gambar 5.13.

Pada Gambar 5.13 baris kedua dan baris ketiga menunjukkan dimensi data sebelum fiturnya dirata-ratakan dan setelah dirata-ratakan. Dapat dilihat hasil dari 10 *fold* dengan rata-rata akurasi sebesar 70%, dengan akurasi terbaik sebesar 76%.

Pada Gambar 5.23 diperlihatkan sinyal hasil prediksi pada tiap kelas untuk skenario 2.

### 5.4.3 Skenario dan evaluasi Uji Coba 3

Skenario uji coba 3 adalah pengujian aplikasi dengan menggunakan kerangka kerja pertama dengan melakukan rata-rata pada tiap data setiap 500 fitur. Pada Gambar 5.11 ditunjukkan data sinyal yang telah dirata-ratakan, dengan jumlah fitur sebanyak 48.

Hasil dari klasifikasi dengan menggunakan 500 fitur ditunjukkan pada Gambar 5.14, dengan baris kedua dan baris ketiga menunjukkan dimensi data sebelum fiturnya dirata-ratakan dan setelah dirata-ratakan. Gambar 5.14 memperlihatkan hasil dari 10 *fold* dengan rata-rata akurasi sebesar 45%, dengan akurasi terbaik didapatkan sebesar 46%. Pada Gambar 5.24 diperlihatkan sinyal hasil prediksi pada tiap kelas untuk skenario 3.

### 5.4.4 Skenario dan evaluasi Uji Coba 4

Skenario uji coba 4 adalah pengujian aplikasi dengan menggunakan kerangka kerja kedua yaitu melakukan *preprocessing* tanpa menggunakan ICA. Untuk skenario ini dilakukan rata-rata pada setiap 50 fitur. Contoh data awal sebelum dilakukan rata-rata per-N fitur dapat dilihat pada Gambar 5.15, dengan jumlah fitur awal sebanyak 24.192.

Pada Gambar 5.16 ditunjukkan data yang telah dirata-ratakan, dengan jumlah 484 fitur. Hasil dari klasifikasi dengan menggunakan rata-rata setiap 50 fitur ditunjukkan pada Gambar 5.19, dengan baris kedua dan baris ketiga menunjukkan dimensi data sebelum dan setelah fiturnya di dirata-ratakan.

Terlihat bahwa pada sinyal kerangka kerja dua, frekuensi dari sinyal tinggi dengan rata-rata amplitudo yang rendah dibandingkan dengan kerangka kerja pertama yang mempunyai amplitudo yang tinggi dan frekuensi yang lebih rendah.

Pada Gambar 5.19 memperlihatkan hasil dari 10 *fold* dengan rata-rata akurasi sebesar 61%, dengan akurasi terbaik

sebesar 65%. Pada Gambar 5.25 diperlihatkan hasil prediksi pada tiap kelas untuk skenario 4.

#### **5.4.5 Skenario Uji dan evaluasi Coba 5**

Skenario uji coba 5 adalah pengujian aplikasi dengan menggunakan kerangka kerja kedua. Untuk skenario ini dilakukan rata-rata pada setiap 100 fitur, sehingga menghasilkan data sinyal dengan jumlah fitur sebanyak 242.

Pada Gambar 5.17 ditunjukkan data yang telah dirata-ratakan. Hasil dari klasifikasi dengan menggunakan 100 fitur ditunjukkan pada Gambar 5.20. Pada skenario ini bisa dilihat hasil dari 10 *fold* dengan rata-rata akurasi sebesar 65%, dengan akurasi terbaik sebesar 68%. Pada Gambar 5.26 diperlihatkan hasil prediksi pada tiap kelas untuk skenario 5.

#### **5.4.6 Skenario Uji dan evaluasi Coba 6**

Skenario uji coba 6 adalah pengujian aplikasi dengan menggunakan kerangka kerja kedua. Untuk skenario ini dilakukan rata-rata pada setiap 500 fitur, menghasilkan data sinyal dengan jumlah fitur sebanyak 48, seperti terlihat Gambar 5.18.

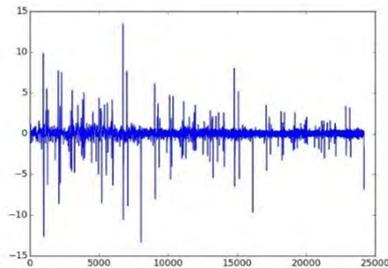
Hasil dari klasifikasi dengan menggunakan 500 fitur ditunjukkan dengan Gambar 5.21. Pada skenario ini bisa dilihat hasil dari 10 *fold* dengan rata-rata akurasi sebesar 42%, dengan akurasi terbaik sebesar 45%. Pada Gambar 5.27 diperlihatkan hasil prediksi pada tiap kelas untuk skenario 6

#### **5.4.7 Evaluasi Semua skenario**

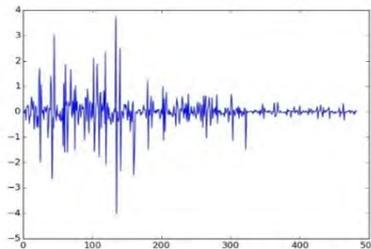
Terlihat pada Gambar 5.22, Gambar 5.23, Gambar 5.24, Gambar 5.25, Gambar 5.26 dan Gambar 5.27 bahwa kerangka kerja pertama kurang dapat membandingkan antara kelas 2 dan kelas 3. Sedangkan pada kerangka kerja kedua, kurang dapat membandingkan antara kelas 2, 3 dan 4.

Pada Tabel 5.1 diperlihatkan nilai akurasi dari kedua kerangka kerja, dengan jumlah fitur sebagai parameternya. Kerangka kerja 1 mempunyai nilai akurasi diatas kerangka kerja 2 pada semua rata-rata N fitur.

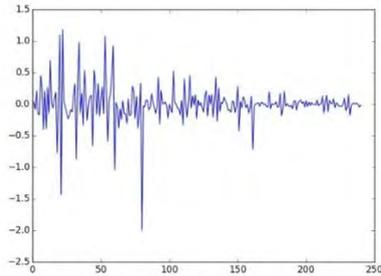
Dari Gambar 5.28 dapat dilihat bahwa nilai rata-rata 100 fitur mempunyai nilai akurasi yang paling tinggi pada semua kerangka kerja, yaitu 70% untuk kerangka kerja pertama dan 65% untuk kerangka kerja kedua. Hasil evaluasi ini menunjukkan bahwa *preprocessing* dengan ICA efektif untuk mereduksi *noise* sehingga data menjadi lebih diskriminatif antar kelas.



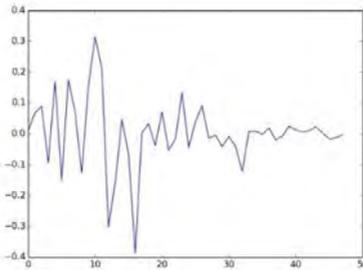
**Gambar 5.8 Contoh sinyal tanpa rata-rata pada kerangka kerja 1**



**Gambar 5.9 Sinyal dengan rata-rata 50 fitur pada kerangka kerja 1**



**Gambar 5.10 Sinyal dengan rata-rata 100 fitur pada kerangka kerja 1**



**Gambar 5.11 Sinyal dengan rata-rata 500 fitur pada kerangka kerja 1**

```

C:\Anaconda\python.exe "E:/Disea TA fur geht Deutchs/TA/untitled/TA/ta_ujicoba.py"
dataset dimensi (880L, 24192L)
setelah di rata dataset dimensi (880L, 483L)
kelas dimensi (880L, 1L)
141 220 64.0909090909
126 220 57.2727272727
127 220 57.7272727273
142 220 64.5454545455
144 220 65.4545454545
137 220 62.2727272727
137 220 62.2727272727
137 220 62.2727272727
131 220 59.5454545455
141 220 64.0909090909
61.9545454545

```

**Gambar 5.12 Hasil *output* program untuk skenario 1**

```

C:\Users\SYAHDEINI\Anaconda\python.exe D:/TAAAA/ta_ujicoba.py
dataset dimensi (880L, 24192L)
setelah di rata dataset dimensi (880L, 241L)
kelas dimensi (880L, 1L)
167 220 75.9090909091
154 220 70.0
150 220 68.1818181818
154 220 70.0
148 220 67.2727272727
154 220 70.0
151 220 68.6363636364
159 220 72.2727272727
157 220 71.3636363636
146 220 66.3636363636
70.0

Process finished with exit code 0

```

Gambar 5.13 Hasil *output* program untuk skenario 2

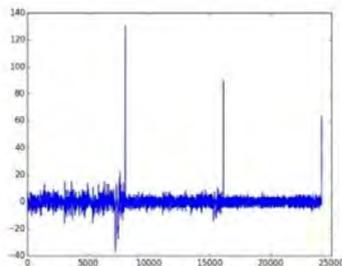
```

C:\Users\SYAHDEINI\Anaconda\python.exe D:/TAAAA/ta_ujicoba.py
dataset dimensi (880L, 24192L)
setelah di rata dataset dimensi (880L, 48L)
kelas dimensi (880L, 1L)
101 220 45.9090909091
101 220 45.9090909091
112 220 50.9090909091
96 220 43.6363636364
105 220 47.7272727273
92 220 41.8181818182
94 220 42.7272727273
100 220 45.4545454545
94 220 42.7272727273
104 220 47.2727272727
45.4090909091

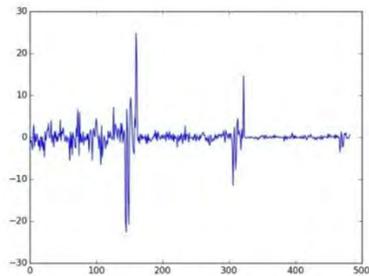
Process finished with exit code 0

```

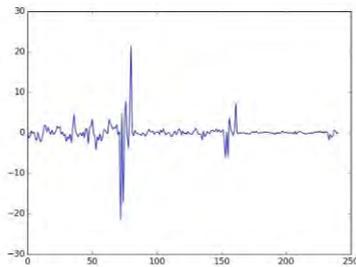
Gambar 5.14 Hasil *output* program untuk skenario 3



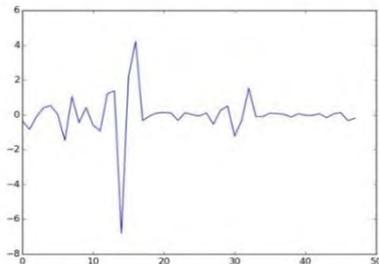
Gambar 5.15 Contoh sinyal tanpa rata-rata pada kerangka kerja 2



**Gambar 5.16 Sinyal yang telah dirata-ratakan tiap 50 fitur pada kerangka kerja 2**



**Gambar 5.17 Sinyal yang telah dirata-ratakan tiap 100 fitur pada kerangka kerja 2**



**Gambar 5.18 Sinyal yang telah dirata-ratakan tiap 500 fitur pada kerangka kerja 2**

```

C:\Anaconda\python.exe "E:/Diese IA fur geht Deuch?
dataset dimensi (880L, 24192L)
setelah di rata dataset dimensi (880L, 483L)
kelas dimensi (880L, 1L)
144 220 65.4545454545
120 220 54.5454545455
133 220 60.4545454545
137 220 62.2727272727
134 220 60.9090909091
123 220 55.9090909091
142 220 64.5454545455
138 220 62.7272727273
126 220 57.2727272727
149 220 67.7272727273
61.1818181818

```

Gambar 5.19 Hasil *ouput* program untuk skenario 4

```

kelas dimensi (880L, 1L)
146 220 66.3636363636
140 220 63.6363636364
145 220 65.9090909091
146 220 66.3636363636
128 220 58.1818181818
142 220 64.5454545455
160 220 72.7272727273
132 220 60.0
149 220 67.7272727273
146 220 66.3636363636
65.1818181818

```

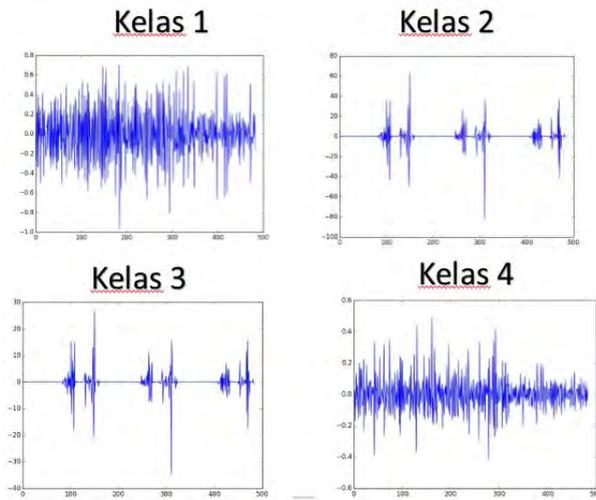
Gambar 5.20 Hasil *ouput* program untuk skenario 5

```

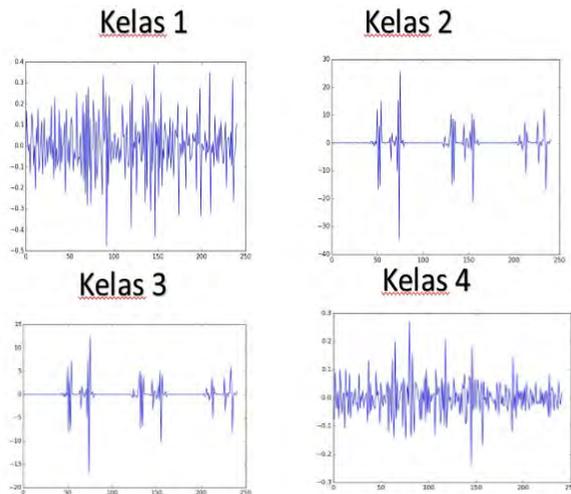
C:\Anaconda\python.exe "E:/Diese IA fur geht Deuch?/IA/u
dataset dimensi (880L, 24192L)
setelah di rata dataset dimensi (880L, 48L)
kelas dimensi (880L, 1L)
88 220 40.0
90 220 40.9090909091
89 220 40.4545454545
99 220 45.0
92 220 41.8181818182
84 220 38.1818181818
98 220 44.5454545455
89 220 40.4545454545
94 220 42.7272727273
94 220 42.7272727273
41.6818181818

```

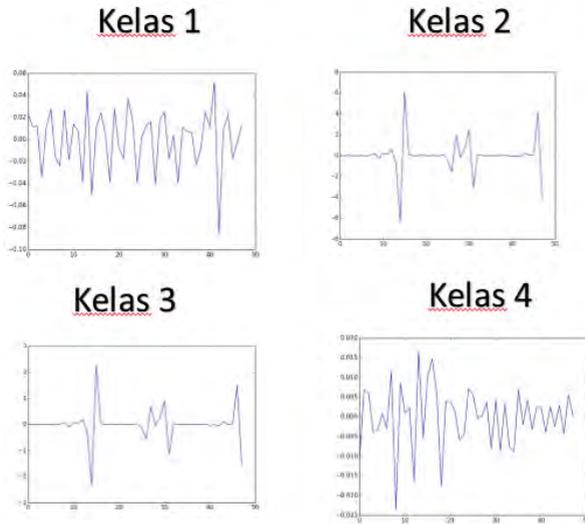
Gambar 5.21 Hasil *ouput* program untuk skenario 6



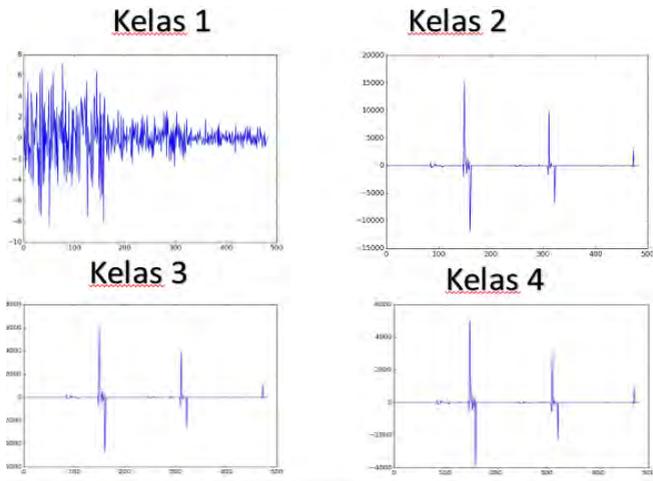
**Gambar 5.22 Sinyal hasil prediksi kelas pada skenario 1**



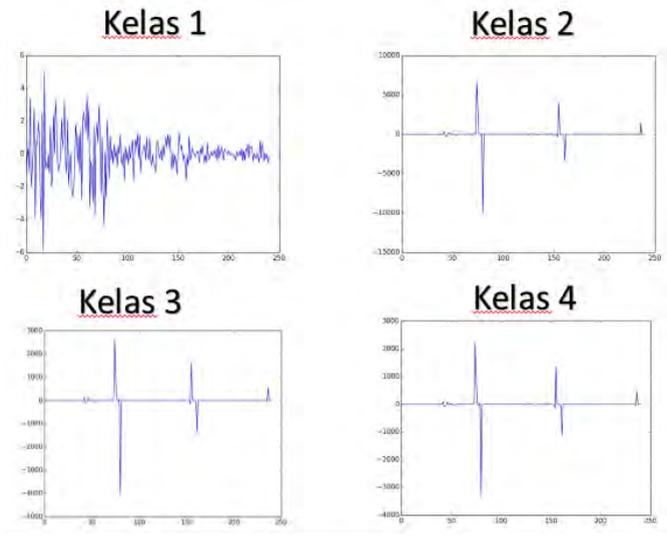
**Gambar 5.23 Sinyal hasil prediksi kelas pada skenario 2**



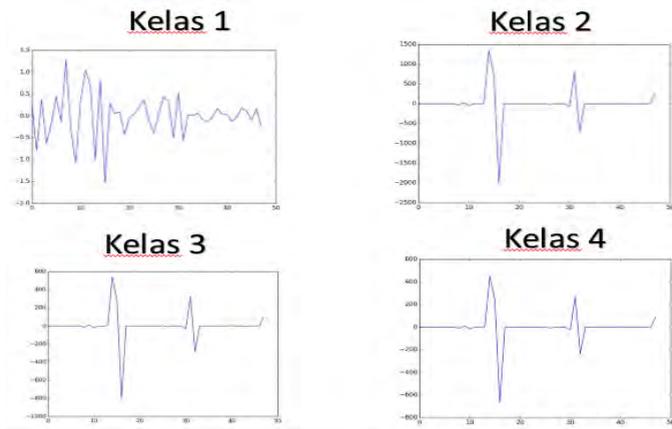
**Gambar 5.24** Sinyal hasil prediksi kelas pada skenario 3



**Gambar 5.25** Sinyal hasil prediksi kelas pada skenario 4



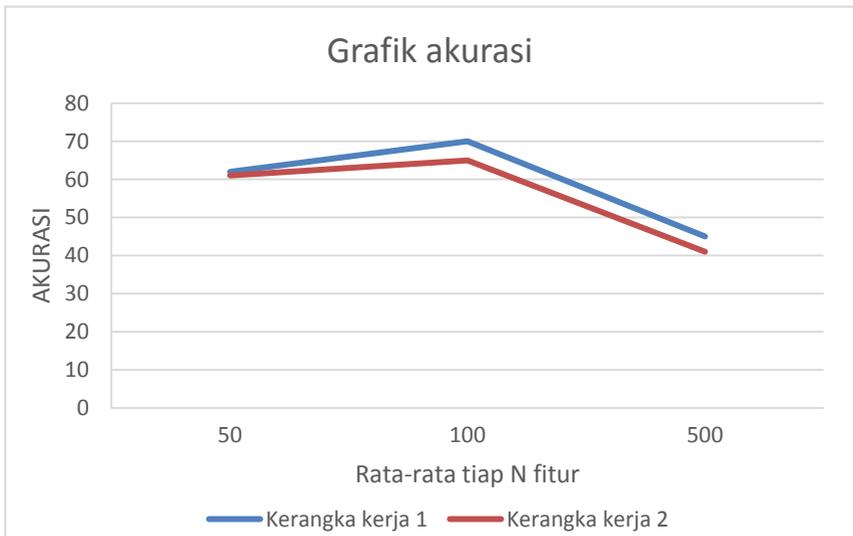
**Gambar 5.26** Sinyal hasil prediksi kelas pada skenario 5



**Gambar 5.27** Sinyal hasil prediksi kelas pada skenario 6

**Tabel 5.1 Hasil Klasifikasi kedua kerangka kerja**

Kerangka kerja 1 (%)	Kerangka kerja 2 (%)	Rata-rata per N fitur
62	61	50
70	65	100
45	42	500

**Gambar 5.28 Akurasi semua kerangka kerja**

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

#### **6.1 Kesimpulan**

Dari hasil uji coba yang telah dilakukan terhadap kedua kerangka kerja dengan merata-ratakan setiap N fitur dari data, dapat diambil kesimpulan sebagai berikut:

1. ICA dapat mengurangi *noise* pada sinyal EEG sehingga klasifikasi data menjadi lebih baik.
2. Akurasi terbaik sebesar 70% didapatkan pada model uji coba dengan *preprocess* menggunakan ICA, SWT dan metode klasifikasi MDA dengan merata-ratakan setiap 100 fitur.

#### **6.2 Saran**

Saran yang diberikan untuk klasifikasi emosi pada sinyal EEG dengan menggunakan *Independent Component Analysis*, *wavelet denoising* dan *Multiple Discriminant Analysis*:

1. Pemotongan data dapat dilakukan dengan lebih detail pada EEGLAB sesuai dengan *channel* status, apabila digunakan komputer yang mempunyai memori yang lebih besar.
2. Pada tugas akhir ini kelas dari data masih dalam bentuk yang umum. Untuk mendapatkan emosi yang lebih khusus dapat digunakan derajat *arousal* dan *valence* dari data dan *classifier* yang khusus untuk memprediksi lebih dari satu nilai kelas.

## DAFTAR PUSTAKA

- [1] Y. Liu, O. Sourina and M. K. Nguyen, "Real-time EEG-based Emotion Recognition and its Application," in *Transactions on computational science XII*, Springer-Verlag Berlin, ACM, 2011, pp. 256-277.
- [2] K. Ishino and M. hagiwara, "A feeling estimation system using a simple electroencephalograph," in *IEEE International Conference on Systems, Man and Cybernetics*, 2003.
- [3] J. TP1, M. S, H. C, L. TW, M. MJ, I. V and S. TJ, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, no. 2, 2000.
- [4] M. T. Akhtara, W. Mitsuhashib and C. J. Jamesc, "Employing spatially constrained ICA and wavelet denoising, for automatic removal of artifacts from multichannel EEG data," *Signal Processing*, vol. 92, no. 2, pp. 401-416, 2012.
- [5] X. Zhou, S. &. D. G. C. U. Sch. of Eng., C. Zhou and B. Stewart, "Comparisons of discrete wavelet transform, wavelet packet transform and stationary wavelet transform in denoising PD measurement data," *IEEE Electrical Insulation*, pp. 237-240, 2006.
- [6] F. Lotte, M. Congedo, A. Lecuyer, F. lamarche and B. Arnaldi, "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces," *Journal of Nerual Engineering*, vol. 4, no. 2, 2007.
- [7] Johnson, R. A and D. W. Wichern, APPLIED MULTIVARIATE STATISTICAL ANALYSIS, PEARSON, Prentice Hall, 2007.
- [8] J. W. Kalat and M. N. Shiota, Emotion, 2011.

- [9] S. Koelstra, C. Muhl, J.-S. Lee, T. Ebrahimi, T. Pun, A. Nijholt and I. Patras, "DEAP: A Database for Emotion Analysis using Physiological Signal," *The IEEE Transactions on Affective Computing*, vol. 3, 2011.
- [10] M. Ungureanu, C. Bigan, R. Strungaru and V. Lazarescu, "Independent Component Analysis Applied in Biomedical Signal Processing".
- [11] N. Mammone, M. U. o. R. C. R. C. I. Dept. DIMET, F. La Foresta and F. Morabito, "Automatic Artifact Rejection From Multichannel Scalp EEG by Wavelet ICA," *Sensors Journal, IEEE* , vol. 12, no. 3, pp. 533-542, 2012.

## LAMPIRAN

**Tabel A.1** Urutan *channel* pada data sinyal asli dari DEAP

Channel no.	Ch. name Twente	Ch. name Geneva	Geneva > Twente	Twente > Geneva
1	Fp1	Fp1	1	1
2	AF3	AF3	2	2
3	F7	F3	4	4
4	F3	F7	3	3
5	FC1	FC5	6	6
6	FC5	FC1	5	5
7	T7	C3	8	8
8	C3	T7	7	7
9	CP1	CP5	10	10
10	CP5	CP1	9	9

11	P7	P3	12	12
12	P3	P7	11	11
13	Pz	PO3	16	14
14	PO3	O1	13	15
15	O1	Oz	14	16
16	Oz	Pz	15	13
17	O2	Fp2	32	30
18	PO4	AF4	31	29
19	P4	Fz	29	31
20	P8	F4	30	27
21	CP6	F8	27	28
22	CP2	FC6	28	25
23	C4	FC2	25	26
24	T8	Cz	26	32
25	FC6	C4	22	23
26	FC2	T8	23	24
27	F4	CP6	20	21

28	F8	CP2	21	22
29	AF4	P4	18	19
30	Fp2	P8	17	20
31	Fz	PO4	19	18
32	Cz	O2	24	17

**Tabel A.2 *Video* yang digunakan pada percobaan**

Experiment id	Online_id	Lastfm_tag	Artist	Title	VAQ Estimate	VAQ Online
1	2	fun	EmilAana Torrini	Jungle Drum	1	1
2	7	exciting	Lustra	Scotty Doesn't Know	1	1
3	15	joy	Jackson 5	Blame It On The Boogie	1	1
4	62		The B52'S	Love Shack	1	1
5	63		Blur	Song 2	1	1

<b>6</b>	70		Blink 182	First Date	1	1
<b>7</b>	72		Benny Benassi	Satisfaction	1	1
<b>8</b>	84		Lily Allen	Fuck You	2	1
<b>9</b>	88		Queen	I Want To Break Free	2	1
<b>10</b>	118		Rage Against The Machine	Bombtrack	4	1
<b>11</b>	9	happy	Michael Franti & Spearhead	Say Hey (I Love You)	1	2
<b>12</b>	17	cheerful	Grand Archives	Miniature Birds	2	2
<b>13</b>	24	love	Bright Eyes	First Day Of My Life	2	2
<b>14</b>	27	happy	Jason Mraz	I'm Yours	2	2
<b>15</b>	29	lovely	Bishop Allen	Butterfly Nets	2	2
<b>16</b>	37	sentimental	The Submarines	Darkest Things	3	2
<b>17</b>	76		Air	Moon Safari	2	2
<b>18</b>	80		Louis Armstrong	What A Wonderful World	2	2
<b>19</b>	83		Manu Chao	Me Gustas Tu	2	2
<b>20</b>	85		Taylor Swift	Love Story	2	2

<b>21</b>	108		Diamanda Galas	Gloomy Sunday	4	3
<b>22</b>	31	sentimental	Porcupine Tree	Normal	3	3
<b>23</b>	33	melancholy	Wilco	How To Fight Loneliness	3	3
<b>24</b>	41	sad	James Blunt	Goodbye My Lover	3	3
<b>25</b>	44	depressing	A Fine Frenzy	Goodbye My Almost Lover	3	3
<b>26</b>	45	mellow	Kings Of Convenience	The Weight Of My Words	3	3
<b>27</b>	95		Madonna	Rain	3	3
<b>28</b>	96		Sia	Breathe Me	3	3
<b>29</b>	97		Christina Aguilera	Hurt	3	3
<b>30</b>	98		Enya	May It Be (Saving Private Ryan)	3	3
<b>31</b>	51	terrible	Mortemia	The One I Once Was	4	4
<b>32</b>	53	shock	Marilyn Manson	The Beautiful People	4	4
<b>33</b>	56	hate	Dead To Fall	Bastard Set Of Dreams	4	4

<b>34</b>	107		Dj Paul Elstak	A Hardcore State Of Mind	4	4
<b>35</b>	111		Napalm Death	Procrastination On The Empty Vessel	4	4
<b>36</b>	112		Sepultura	Refuse Resist	4	4
<b>37</b>	113		Cradle Of Filth	Scorched Earth Erotica	4	4
<b>38</b>	115		Gorgoroth	Carving A Giant	4	4
<b>39</b>	116		Dark Funeral	My Funeral	4	4
<b>40</b>	120		Arch Enemy	My Apocalypse	4	4

## BIODATA PENULIS



Aldy syahdeini, lahir di Selong, pada tanggal 2 Februari 1992. Penulis menempuh pendidikan mulai dari SDN 4 Pancor (1998-2004), SMPN 1 Selong (2004-2007), SMAN 1 Selong (2007-2010) dan S1 Teknik Informatika ITS (2011-2015). Selama masa kuliah, penulis pernah menjadi Google Student Ambassador (GSA) 2012-2013, staff ristek organisasi Himpunan Mahasiswa Teknik Computer (HMTC) 2012-2013, administrator laboratorium pemrograman (LP) 2012-2015. Penulis juga telah memenangkan kompetisi mobile game development war (MGDW 2013) dan sempat aktif dalam kegiatan sosial Sunday sharring loperman (SSL). Selama kuliah di teknik informatika ITS penulis mengambil bidang minat Komputasi Cerdas Visual (KCV). Komunikasi dengan penulis melalui : [syahdeini@gmail.com](mailto:syahdeini@gmail.com).