

**TUGAS AKHIR – KI141502**

**Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine**

**AHMAD MUSTOFA**

**NRP 51112100100**

**Dosen Pembimbing I**

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

**Dosen Pembimbing II**

**Bilqis Amaliah, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA 2016**

***(Halaman ini sengaja dikosongkan)***



**TUGAS AKHIR – KI141502**

Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine

AHMAD MUSTOFA

NRP 51112100100

Dosen Pembimbing I

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Dosen Pembimbing II

Bilqis Amaliah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA 2016

***(Halaman ini sengaja dikosongkan)***



**FINAL PROJECT – KI141502**

Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine

AHMAD MUSTOFA

NRP 51112100100

Supervisor I

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Supervisor II

Bilqis Amaliah, S.Kom., M.Kom.

**DEPARTMENT OF INFORMATICS**

**FACULTY OF INFORMATION TECHNOLOGY**

**SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY**

**SURABAYA 2016**

***(Halaman ini sengaja dikosongkan)***

# LEMBAR PENGESAHAN

**Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine**

**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 MUSTOFA**

**NRP: 5112 100 100**

**Disetujui oleh Pembimbing Tugas Akhir:**

|  |  |
| --- | --- |
| Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.  NIP: 19490823 197603 2 001 | …………………….  (pembimbing 1) |
| Bilqis Amaliah, S.Kom., M.Kom.  NIP: 19750914 200112 2 002 | …………………….  (pembimbing 2) |

**SURABAYA**

**JUNI, 2016**

***(Halaman ini sengaja dikosongkan)***

**DETEKSI PENYAKIT GLAUKOMA PADA CITRA FUNDUS RETINA MATA MENGGUNAKAN ADAPTIVE THRESHOLDING DAN SUPPORT VECTOR MACHINE**

Nama : Ahmad Mustofa

NRP : 5112100100

Jurusan : Teknik Informatika, FTIf-ITS

Dosen Pembimbing I : Prof. Ir. Handayani Tjandrasa, M.Sc.,

Ph.D.

Dosen Pembimbing II : Bilqis Amaliah, S.Kom., M.Kom.

# Abstrak

Glaukoma adalah penyebab kebutaan kedua terbanyak setelah katarak. Berbeda dengan katarak, kebutaan yang disebabkan oleh glaukoma bersifat permanen. Hal ini karena glaukoma disebabkan oleh tekanan besar pada bola mata yang menyebabkan tersumbatnya pembuluh darah yang menuju ke syaraf mata sehingga syaraf mata tidak mendapatkan suplai darah yang cukup dan akhirnya akan mengalami kerusakan.

Gejala glaukoma yang timbul biasanya tidak dapat dirasakan secara langsung. Sehingga perlu dilakukan pemeriksaan mata terlebih dahulu untuk mengetahui adanya glaukoma.

*Pada pengerjaan tugas akhir ini, sinyal electrooculogram (EOG) diekstrak dari sinyal electroenchepalogram (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* filter*isasi, 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 “*Deteksi Penyakit Glaukoma Pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding Dan Support Vector Machine*” 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 Laboraturium 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 pengrejaan 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](#_Toc422222348)

[Abstrak ix](#_Toc422222349)

[*Abstract* xi](#_Toc422222350)

[KATA PENGANTAR xiii](#_Toc422222351)

[DAFTAR ISI xv](#_Toc422222352)

[DAFTAR GAMBAR xvii](#_Toc422222353)

[DAFTAR TABEL xix](#_Toc422222354)

[BAB I 1](#_Toc422222355)

[1.1 Latar Belakang 1](#_Toc422222356)

[1.2 Rumusan Masalah 2](#_Toc422222357)

[1.3 Batasan Masalah 2](#_Toc422222358)

[1.4 Tujuan 3](#_Toc422222359)

[1.5 Metodologi 3](#_Toc422222360)

[1.6 Sistematika Penulisan 4](#_Toc422222361)

[BAB II 7](#_Toc422222362)

[2.1 Sinyal *Electroencephalogram* (EEG) 7](#_Toc422222363)

[2.2 *NeuroSky Mindwave* 8](#_Toc422222364)

[2.3 Sandi Morse 9](#_Toc422222365)

[2.4 *Empirical Mode Decomposition* (EMD) 10](#_Toc422222366)

[2.5 Transformasi *Fourier* 13](#_Toc422222367)

[2.6 Dynamic Time Warping (DTW) 14](#_Toc422222368)

[2.7 K-*Nearest* *Neighbor* (kNN) *Classifier* 15](#_Toc422222369)

[2.8 *K-Means* *Clustering* 17](#_Toc422222370)

[2.9 JSON *String* 17](#_Toc422222371)

[BAB III 19](#_Toc422222372)

[3.1 Desain Metode Secara Umum 19](#_Toc422222373)

[3.2 Perancangan Data 21](#_Toc422222374)

[3.2.1 Data Masukan 21](#_Toc422222375)

[3.2.2 Data Luaran 22](#_Toc422222376)

[3.3 Perancangan Proses 23](#_Toc422222377)

[3.3.1 Tahap Pengambilan Data 23](#_Toc422222378)

[3.3.2 Contoh Data 26](#_Toc422222379)

[3.3.3 Tahap Khusus Pengenalan Berdasarkan Aturan 27](#_Toc422222380)

[3.3.4 Tahap Khusus Pengenalan Berdasarkan Pembelajaran 39](#_Toc422222381)

[3.4 Tahap Evaluasi Metode 42](#_Toc422222382)

[3.5 Perancangan Antarmuka Perangkat Lunak 43](#_Toc422222383)

[BAB IV 47](#_Toc422222384)

[4.1 Lingkungan Implementasi 47](#_Toc422222385)

[4.2 Implementasi Modul *NeuroskyConnector* 47](#_Toc422222386)

[4.3 Implementasi Modul *StreamClass* 49](#_Toc422222387)

[4.4 Implementasi Modul *MonitorClass* 50](#_Toc422222388)

[4.5 Implementasi Modul Pendeteksi 51](#_Toc422222389)

[4.6 Implementasi Modul Evaluasi 54](#_Toc422222390)

[BAB V 59](#_Toc422222391)

[5.1 Lingkungan Uji Coba 59](#_Toc422222392)

[5.2 Data Uji Coba 59](#_Toc422222393)

[5.3 Uji Coba Proses Pengenalan 60](#_Toc422222394)

[5.4 Skenario Uji Coba 63](#_Toc422222395)

[5.4.1 Skenario Uji Coba 1 64](#_Toc422222396)

[5.4.2 Skenario Uji Coba 2 71](#_Toc422222397)

[5.4.3 Skenario Uji Coba 3 82](#_Toc422222398)

[5.4.4 Skenario Uji Coba 4 90](#_Toc422222399)

[5.4.5 Skenario Uji Coba 5 98](#_Toc422222400)

[5.5 Evaluasi 106](#_Toc422222401)

[BAB VI 109](#_Toc422222402)

[6.1 Kesimpulan 109](#_Toc422222403)

[6.2 Saran 109](#_Toc422222404)

[DAFTAR PUSTAKA 111](#_Toc422222405)

[LAMPIRAN 115](#_Toc422222406)

[BIODATA PENULIS 127](#_Toc422222407)

# DAFTAR GAMBAR

[Gambar 2.1 NeuroSky Mindwave 8](#_Toc423583217)

[Gambar 2.2 Sandi morse standar internasional 10](#_Toc423583218)

[Gambar 2.3 Sinyal masukan pada proses EMD 12](#_Toc423583219)

[Gambar 2.4 IMF yang dihasilkan pada proses EMD 12](#_Toc423583220)

[Gambar 2.5 Ilustrasi optimal warping path [8] 15](#_Toc423583221)

[Gambar 2.6 Ilustrasi algoritma kNN [11] 16](#_Toc423583222)

[Gambar 2.7 Json string 18](#_Toc423583223)

[Gambar 3.1 Diagram alir model sistem 20](#_Toc423583224)

[Gambar 3.2 Sinyal masukan 22](#_Toc423583225)

[Gambar 3.3 Puncak dan lembah tersegmentasi 22](#_Toc423583226)

[Gambar 3.4 Diagram alir penerimaan data 24](#_Toc423583227)

[Gambar 3.5 Paket data yang dikirimkan ke sitem *server* 25](#_Toc423583228)

[Gambar 3.6 Format data yang diterima oleh klien 25](#_Toc423583229)

[Gambar 3.7 Diagram alir *preprocessing* 28](#_Toc423583230)

[Gambar 3.8 Sinyal dalam domain waktu 29](#_Toc423583231)

[Gambar 3.9 Sinyal dalam domain frekuensi 29](#_Toc423583232)

[Gambar 3.10 Ilustrasi bandpass filtering 30](#_Toc423583233)

[Gambar 3.11 Ilustrasi *cut-off* frequency 30](#_Toc423583234)

[Gambar 3.12 Domain frekuensi yang telah terpotong 31](#_Toc423583235)

[Gambar 3.13 Gambar sinyal luaran *preprocessing* 32](#_Toc423583236)

[Gambar 3.14 Bagan tahap segmentasi ROI secara umum 32](#_Toc423583237)

[Gambar 3.15 *Pseudocode* normalisasi sinyal 33](#_Toc423583238)

[Gambar 3.16 *Pseudocode* pendeteksian puncak 34](#_Toc423583239)

[Gambar 3.17 *Pseudocode* pendeteksian ROI 35](#_Toc423583240)

[Gambar 3.18 Diagram tahap *postprocessing* secara umum 37](#_Toc423583241)

[Gambar 3.19 *Pseudocode* tahap *postprocessing* 38](#_Toc423583242)

[Gambar 3.20 Data asli (atas), data hasil pemotongan (bawah) 39](#_Toc423583243)

[Gambar 3.21 Pemilihan *template* sinyal menggunakan *K-Means* *Clustering* 40](#_Toc423583244)

[Gambar 3.22 Pemilihan *template* sinyal menggunakan *Hierarchical Clustering* 41](#_Toc423583245)

[Gambar 3.23 *Pseudocode* k-Nearest Neighbor 42](#_Toc423583246)

[Gambar 3.24 Diagram alir evaluasi metode 43](#_Toc423583247)

[Gambar 3.25 Tatap muka utama 44](#_Toc423583248)

[Gambar 3.26 Kotak dialog *error* yang ditampilkan jika tidak ada perangkat terhubung 45](#_Toc423583249)

[Gambar 3.27 Tampilan hasil Analisis sinyal masukan 46](#_Toc423583250)

[Gambar 4.1 Kode program *NeuroskyConnector* 48](#_Toc423583251)

[Gambar 4.2 Kode program *StreamClass* 49](#_Toc423583252)

[Gambar 4.3 Kode program modul *MonitorClass* 51](#_Toc423583253)

[Gambar 4.4 Modul pengenalan sinyal 53](#_Toc423583254)

[Gambar 4.5 Kode program evalusi skenario 56](#_Toc423583255)

[Gambar 5.1 Sinyal masukan sistem 60](#_Toc423583256)

[Gambar 5.2 Hasil filter *window averaging* 60](#_Toc423583257)

[Gambar 5.3 Hasil filter *empirical mode decomposition* 61](#_Toc423583258)

[Gambar 5.4 Hasil filter *bandpass* 61](#_Toc423583259)

[Gambar 5.5 Hasil pendeteksian ROI oleh filter *window averaging* 62](#_Toc423583260)

[Gambar 5.6 Hasil pendeteksian ROI oleh filter *empirical mode decomposition* 62](#_Toc423583261)

[Gambar 5.7 Hasil pendeteksian ROI oleh filter bandpass 63](#_Toc423583262)

# DAFTAR TABEL

[Tabel 2.1 Tabel parameter persamaan Transformasi Fourier 13](#_Toc423583263)

[Tabel 3.1 Contoh data terekam dan pola sebenarnya 26](#_Toc423583264)

[Tabel 5.1 Hasil skenario 1 dengan Lebar *window* 10 64](#_Toc423583265)

[Tabel 5.2 Hasil skenario 1 dengan Lebar *window* 20 65](#_Toc423583266)

[Tabel 5.3 Hasil skenario 1 dengan Lebar *window* 30 66](#_Toc423583267)

[Tabel 5.4 Hasil skenario 1 dengan Lebar *window* 40 67](#_Toc423583268)

[Tabel 5.5 Hasil skenario 1 dengan Lebar *window* 50 67](#_Toc423583269)

[Tabel 5.6 Rekapitulasi skenario 1 69](#_Toc423583270)

[Tabel 5.7 Contoh pengolahan masing-masing window 69](#_Toc423583271)

[Tabel 5.8 Hasil skenario 2 dengan parameter 1:end-1 71](#_Toc423583272)

[Tabel 5.9 Hasil skenario 2 dengan parameter 1:end-2 72](#_Toc423583273)

[Tabel 5.10 Hasil skenario 2 dengan parameter 2:end-1 73](#_Toc423583274)

[Tabel 5.11 Hasil skenario 2 dengan parameter 2:end-2 74](#_Toc423583275)

[Tabel 5.12 Hasil skenario 2 dengan parameter 3:end-1 75](#_Toc423583276)

[Tabel 5.13 Hasil skenario 2 dengan parameter 3:end-2 76](#_Toc423583277)

[Tabel 5.14 Hasil skenario 2 dengan parameter 4:end-1 76](#_Toc423583278)

[Tabel 5.15 Hasil skenario 2 dengan parameter 4:end-2 77](#_Toc423583279)

[Tabel 5.16 Rekapitulasi skenario 2 79](#_Toc423583280)

[Tabel 5.17 Contoh hasil pengolahan variasi indeks 79](#_Toc423583281)

[Tabel 5.18 Hasil uji coba skenario 3 parameter [0,0; 10] 82](#_Toc423583282)

[Tabel 5.19 Hasil uji coba skenario 3 parameter [0,5; 10] 83](#_Toc423583283)

[Tabel 5.20 Hasil uji coba skenario 3 parameter [0,6; 10] 84](#_Toc423583284)

[Tabel 5.21 Hasil uji coba skenario 3 parameter [1,0; 10] 85](#_Toc423583285)

[Tabel 5.22 Hasil uji coba skenario 3 parameter [0,0; 20] 86](#_Toc423583286)

[Tabel 5.23 Hasil uji coba skenario 3 parameter [5,0; 20] 87](#_Toc423583287)

[Tabel 5.24 Rekapitulasi skenario 3 88](#_Toc423583288)

[Tabel 5.25 Contoh hasil pengolahan variasi frekuensi 89](#_Toc423583289)

[Tabel 5.26 Hasil dengan himpunan latih asli 91](#_Toc423583290)

[Tabel 5.27 Hasil dengan jumlah *template* 1 92](#_Toc423583291)

[Tabel 5.28 Hasil dengan jumlah *template* 5 93](#_Toc423583292)

[Tabel 5.29 Hasil dengan jumlah *template* 15 94](#_Toc423583293)

[Tabel 5.30 Hasil dengan jumlah *template* 1 95](#_Toc423583294)

[Tabel 5.31 Hasil dengan jumlah *template* 5 95](#_Toc423583295)

[Tabel 5.32 Hasil dengan jumlah *template* 15 96](#_Toc423583296)

[Tabel 5.33 Tabel rekapitulasi skenario 4 98](#_Toc423583297)

[Tabel 5.34 Hasil dengan himpunan latih asli 98](#_Toc423583298)

[Tabel 5.35 Hasil dengan jumlah *template* 1 99](#_Toc423583299)

[Tabel 5.36 Hasil dengan jumlah *template* 5 100](#_Toc423583300)

[Tabel 5.37 Hasil dengan jumlah *template* 15 101](#_Toc423583301)

[Tabel 5.38 Hasil dengan jumlah *template* 1 102](#_Toc423583302)

[Tabel 5.39 Hasil dengan jumlah *template* 5 103](#_Toc423583303)

[Tabel 5.40 Hasil dengan jumlah *template* 15 104](#_Toc423583304)

[Tabel 5.41 Rekapitulasi skenario 5 105](#_Toc423583305)

[Tabel 5.42 Perbandingan metode *filtering/preprocessing* 107](#_Toc423583306)

# 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.

## Latar Belakang

Glaukoma adalah penyebab kebutaan kedua terbanyak setelah katarak di seluruh dunia. Berbeda dengan katarak, kebutaan yang disebabkan oleh glaukoma bersifat permanen. Berdasarkan data WHO 2010, diperkirakan sebanyak 3.2 juta orang mengalami kebutaan akibat glaukoma [1]. Disamping itu, perkembangan teknologi informasi dan komunikasi yang pesat harusnya bisa dimanfaatkan untuk mempermudah tenaga kesehatan untuk melaksanakan tugasnya. Manfaat itulah yang diharapkan dari tugas akhir ini, yaitu mempermudah tenaga kesehatan untuk mendeteksi penyakit glaukoma dari citra retina mata pasien.

Dalam tugas akhir ini, fitur yang akan digunakan untuk deteksi glaukoma adalah CDR (Cup to Disc Ratio), luas NRR (Neuro Retinal Rim), dan luas pembuluh darah di daerah optic disc dalam ISNT [2]. Metode yang akan digunakan untuk segmentasi optic disc dan optic cup adalah adaptive thresholding, sedangkan untuk klasifikasi digunakan metode support vector machine.

Hasil yang diharapkan dari tugas akhir ini adalah sebuah aplikasi yang mampu mengklasifikasi apakah masukan yang berupa citra retina mata menderita glaukoma atau tidak. Aplikasi ini juga diharapkan dapat mempermudah tenaga medis dalam mendiagnosis penyakit glaukoma pada pasien.

## Rumusan Masalah

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

1. Bagaimana melakukan segmentasi optic cup, optic disc, dan pembuluh darah dari citra retina mata?
2. Bagaimana sistem dapat mendeteksi penderita glaukoma dari citra retina mata?

## Batasan Masalah

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

1. Kelas retina mata yang dapat diklasifikasi adalah kelas mata normal dan kelas mata glaukoma.
2. Implementasi program yang dilakukan pada lingkungan komputer desktop.

## Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut

1. Merancang dan membuat aplikasi yang bisa melakukan segmentasi optic cup, optic disc, dan pembuluh darah dari citra retina mata.
2. Merancang dan membuat aplikasi yang bisa mendeteksi penderita glaukoma dari citra retina mata.

## 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 pendeteksian penyakit glaukoma pada citra fundus retina mata menggunakan metode *adaptive thresholding* dan *support vector machine*.

* + - 1. **Studi Literatur**

Pada tahap ini dilakukan pencarian informasi dan studi literatur sejumlah referensi tentang deteksi penderita glaukoma dengan masukan citra retina mata. Informasi dan studi literatur tersebut didapat dari buku, internet, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

* + - 1. **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.

* + - 1. **Pengujian dan Evaluasi**

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

* + - 1. **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.

## 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.

1. Bab II. Tinjauan Pustaka

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

1. 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.

1. Bab IV. Implementasi

Bab implementasi berisi pembangunan aplikasi pendeteksi penyakit glaukoma dari citra retina mata menggunakan matlab.

1. 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.

1. 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.

***(Halaman ini sengaja dikosongkan)***

# 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.

## Glaukoma

Glaukoma merupakan penyebab kebutaan kedua terbanyak setelah katarak di seluruh dunia. Berbeda dengan katarak, kebutaan yang diakibatkan glaukoma bersifat permanen, atau tidak dapat disembuhkan. Hal ini menjadi tantangan tersendiri dalam upaya pencegahan dan penanganan kasus glaukoma. Berdasarkan data WHO 2010, diperkirakan sebanyak 3.2 juta orang mengalami kebutaan akibat glaukoma.

Glaukoma adalah penyakit mata dimana terjadi kerusakan saraf optik yang diikuti gangguan pada lapang pandangan yang khas. Kondisi ini utamanya diakibatkan oleh tekanan bola mata yang meninggi yang biasanya disebabkan oleh hambatan pengeluaran cairan bola mata (humour aquous). Penyebab lain kerusakan saraf optic, antara lain gangguan suplai darah ke saraf optic dan kelemahan /masalah saraf optiknya sendiri.

Glaukoma dapat diklasifikasikan menjadi glaukoma primer, glaukoma sekunder, dan glaukoma kengenital. Glaukoma primer adalah glaukoma yang tidak diketahui penyebabnya. Glaukoma primer sudut terbuka biasanya merupakan glaukoma kronis, sedangkan glaukoma primer sudut tertutup bisa berupa glaukoma sudut tertutup akut atau kronis. Glaukoma sekunder adalah glaukoma yang timbul sebagai akibat dari penyakit mata lain, trauma, pembedahan, penggunaan kortikosteroid yang berlebihan atau penyakit sistemik lainnya. Glaukoma kongenital adalah glaukoma yang ditemukan sejak dilahirkan dan biasanya disebabkan oleh sistem saluran pembuangan di dalam mata tidak berfungsi dengan baik sehingga menyebabkan pembesaran mata bayi. Di samping itu glaukoma dengan kebutaan total disebut juga sebagai glaukoma absolut. [1]

## *Optic Disc* dan *Optic Cup*

*Optic disc* didefinisikan sebagai bagian dari saraf optik yang membentang dari bagian mielin saraf tepat dibelakang sklera sampai permukaan retina. Umumnya berbentuk sedikit oval dengan diameter vertikal sekitar 9% lebih besar dari diameter horizontal. Rata – rata pada orang normal, diameter vertikalnya mendekati 1500 mikrometer [3].

Sedangkan *optic cup* adalah sebuah lekukan berbentuk mangkok yang dibentuk oleh invaginasi dari gelembung optik yang berkembang menjadi lapisan sensorik dan berpigmen dari retina [4].

Pada umumnya *optic disc* merupakan daerah yang paling terang dalam citra retina, sedangkan *optic cup* merupakan daerah paling terang di daerah *optic disc*.[paper utama]

## Aturan ISNT

ISNT adalah sebuah aturan yang menitikberatkan pada 4 daerah pada optic disc, yaitu Inferior, Superior, Nasal, dan Temporal. Keempat daerah inilah yang memiliki peran dalam membedakan mata yang normal dan yang tidak [5], baik itu jika dinilai dari keberadaan pembuluh darah maupun optic cup di masing-masing daerah [2].



Gambar 2.1 *Masks* yang digunakan untuk mendapatkan luasan di daerah a Superior; b Temporal; c Inferior; dan d Nasal dari optic disc

## *Cup to Disc Ratio*

*Cup to Disc Ratio* (CDR) adalah perbandingan dari *optic cup* dan *optic disc* baik dari segi diameter vertikal, diameter horizontal, ataupun luas. CDR adalah salah satu fitur yang populer digunakan dalam pengevaluasian glaukoma [2]. Dalam tugas akhir ini, CDR yang digunakan adalah rasio jejari dari optic cup dan optic disk yang didapatkan dengan rumus sebagai berikut.

## Neuro Retinal Rim

Neuro Retinal Rim (NRR) adalah daerah yang tersisa jika daerah optic cup dihapus dari optic disc. Bentuk dari NRR dapat digunakan untuk mengukur kerusakan pada optic disc [2].

## *Histogram Smoothing*

*Histogram smoothing* adalah sebuah metode untuk memperhalus histogram dari sebuah image dengan filter tertentu. Dalam tugas akhir ini filter yang digunakan adalah filter gaussian dengan rerata 50, standar deviasi 6, dan ukuran window 1x100.

## Bottom Hat Transormation

Bottom Hat Transformation adalah sebuah operasi morfologi yang dapat digunakan untuk mendapatkan objek yang gelap pada background yang terang.

dimana adalah operasi dilasi dan adalah operasi erosi.

## Otsu Thresholding

Dalam *image processing,* metode otsu digunakan untuk melakukan *thresholding* berbasis *clustering* sehingga didapatkan batas yang dapat dijadikan acuan untuk melakukan segmentasi pada suatu citra. Dalam metode ini jumlah kelas yang dapat ditangani hanya ada 2, yaitu objek dan latar belakang. Konsep utama dari metode ini adalah dengan memilih nilai batas yang menghasilkan kelas dengan varian intra kelasnya terkecil dan varian antar kelasnya terbesar. Rumus untuk menghitung varian antar kelas adalah sebagai berikut.

dimana dan

serta dan dengan

Berikut ini pseudocode dari metode otsu.

* Hitung histogram dan peluang dari setiap level intensitas
* Inisialisasi nilai awal
* For t = 1 to 256
  + Update
  + Hitung
* Pilih t dengan nilai maksimum

## Median Filtering

*Median filtering* adalah sebuah metode *filtering* yang pada umumnya digunakan untuk menghilangkan *noise.* Berikut pseudocode dari median filtering dua dimensi.

|  |  |
| --- | --- |
| 1 | **MedianFiltering(I,size)** |
| 2 | Foreach p = pixels in I |
| 3 | p = median(part(I(position(p).x – size, position(p).y + size)) |
| 4 | return I |

## Morfologi Erosi dan Dilasi

Operasi morfologi erosi dan dilasi adalah metode yang biasa dipakai dalam image processing dengan tujuan – tujuan tertentu. Dalam tugas akhir ini, operasi morfologi erosi dan dilasi digunakan untuk menutup gap yang dihasilkan oleh pembuluh darah pada saat melakukan segmentasi optic disc dan optic cup.

Operasi morfologi erosi dan dilasi membutuhkan elemen struktur yang berbeda-beda sesuai kebutuhan. Elemen struktur yang digunakan pada tugas akhir ini adalah elemen struktur berbentuk disk dengan ukuran 50.

## *Support Vector Machine*

*Support vector machine* (SVM) adalah metode klasifikasi *supervised* (data latih sudah diketaui kelasnya) yang mengklasifikasikan dua kelas. Pada metode ini, setiap data akan di-*plotting* ke dalam ruang *n-*dimensi (dimana *n* adalah jumlah fitur) dengan nilai masing-masing fitur menjadi nilai tertentu pada koordinat. Kemudian akan dilakukan klasifikasi dengan mencari *hyperplane* (bidang pembatas) untuk membedakan antara dua kelas sebaik mungkin. Gambar merupakan beberapa contoh dari kemungkinan *hyperplane* yang digunakan untuk memisahkan kelas satu dengan lainnya. Metode SVM akan mencari h*yperplane* yang memiliki *margin* yang maksimal. *Support vector* adalah data-data yang dijadikan pembatas dengan kelas lain.



Gambar 2.5 Contoh alternatif hyperplane.

(Sumber gambar : [www.sine.ni.com](http://www.sine.ni.com))

Diberikan data masukan dan masing-masing kelas dinotasikan untuk dimana *n* adalah banyaknya data. Fungsi *hyperplane* dibuat dengan persamaan

dengan batasan yang ditulis dalam persamaan

untuk mencari nilai *w* dan *b* yang optimal digunakan persamaan

dimana nilai *C* adalah nilai pinalti dari kesalahan klasifikasi. Fungsi tujuan persamaan di atas berbentuk kuadrat, sehingga untuk menyelesaikannya, bentuk tersebut ditransformasikan ke dalam bentuk *dual space*. Persamaan *dual space* dapat ditulis menggunakan persamaan

dengan batasan

Untuk mencari nilai , digunakan *sequential minimal optimization* (SMO). Persamaan *hyperplane* dilakukan dengan persamaan:

dimana *z* adalah data masukan *support vector machine*. Pada banyak kasus, data yang diklasifikasikan tidak bisa langsung dipisahkan dengan garis yang linear. Oleh karena itu, digunakan metode kernel untuk mengatasi permasalahan tersebut. Dengan metode kernel, suatu data di *input space* dipetakan ke fitur *space*  dengan dimensi yang lebih tinggi. Salah satu kernel yang biasa dipakai adalah kernel RBF dan *Polynomial*. Persamaan kernel RBF dan *Polynomial* berurutan dapat dilihat pada persamaan di bawah ini:

Penggunaan fungsi kernel mengubah persamaan *dual space* menjadi

Dan juga mengubah persamaan *hyperplane* menjadi

*(Halaman ini sengaja dikosongkan)*

# 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.

## 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 pendeteksi penyakit glaukoma dari citra fundus retina mata. Data yang diperlukan dalam pengoperasian perangkat lunak adalah data masukan (*input*) dan data keluaran (*output*) yang memberikan hasil proses pengoperasian perangkat lunak untuk pengguna.

### Data Masukan

Data masukan adalah data awal yang akan diproses pada sistem pendeteksi penyakit glaukoma dari citra fundus retina mata. Data yang diproses berupa data citra yang direpresentasikan oleh tiga buah matriks dua dimensi (*channel* *red, green,* dan *blue*) dengan ukuran yang variatif ( tidak ada batasan dalam ukuran citra).

Dalam tugas akhir ini, digunakan dataset citra retina mata yang didapat dari *database* RIM-ONE revisi 2 yang dapat diunduh langsung dari [*http://medimrg.webs.ull.es/research/retinal-imaging/rim-one/*](http://medimrg.webs.ull.es/research/retinal-imaging/rim-one/). Dataset berisi citra retina mata yang sudah dipotong sehingga citra hanya terfokus pada daerah yang terdapat optic disk yang hendak dievaluasi. Label kelas pada dataset terdiri dari dua macam, yaitu normal dan glaukoma. Berikut contoh citra inputan yang digunakan.



Gambar 3.1 Citra Retina Mata Glaukoma

### Data Luaran

Data luaran dari sistem ini adalah hasil klasifikasi dari data masukan dengan label glaukoma untuk kelas 1 dan label normal untuk kelas 0. Selain itu, didapat juga hasil segmentasi optic disk, optic cup, dan pembuluh darah dari citra masukan.



Gambar 3.2 Hasil segmentasi (a) optic cup, (b) optic disk, (c) pembuluh darah di daerah optic disc

## Gambar 3.2 menunjukkan tampilan hasil segmentasi optic cup, optic disk, dan pembuluh darah dari citra retina mata pada gambar 3.1 .

## Desain Metode Secara Umum

Pada tugas akhir ini akan dibangun suatu sistem untuk mendeteksi penyakit glaukoma dari citra masukan dengan menggunakan metode segmentasi *adaptive thresholding* dan metode klasifikasi *support vector machine*. Tahap pertama yang dilakukan opleh sistem adalah tahap *preprocessing.* Pada tahap ini dilakukan proses standarisasi, morfologi erosi dan dilasi pada segmentasi optic cup dan optic disk, serta bottom hat filtering, morfologi erosi dan dilasi pada segmentasi pembuluh darah. Tahap kedua adalah tahap segmentasi dengan menggunakan metode otsu. Tahap ketiga adalah tahap ekstraksi fitur. Fitur yang digunakan pada tugas akhir ini adalah CDR, ISNT pembuluh darah, dan ISNT NRR. Tahap yang terakhir adalah tahap klasifikasi dengan menggunakan pengklasifikasi *support vector machine*.

Bagan dari sistem yang dibangun ditunjukkan oleh Gambar 3.3.

F:\5112100100\Semester 8\Tugas Akhir\Program\tugasAkhir\Bismillah Buku TA\Alur sistem.png

Gambar 3.3 Diagram Alir Model Sistem

## Perancangan Proses

Berikut ini adalah rancangan dari sistem pendeteksi penyakit glaukoma dari citra fundus retina mata:

### Tahap Preprocessing

Pada tahap ini, dilakukan proses awal sebelum masuk tahap segmentasi. Tahap ini dipecah menjadi dua proses yaitu preprocessing optic cup dan optic disk serta preprocessing pembuluh darah.

#### Preprocessing Optic Cup dan Optic Disk

Pada tahap ini terbagi menjadi dua proses yaitu, proses morfologi dan standarisasi. Berikut diagram alur dari tahap ini.

F:\5112100100\Semester 8\Tugas Akhir\Program\tugasAkhir\Bismillah Buku TA\preprocessing cupdisk.png

**Gambar 3.3 Diagram Alir Preprocessing Optic Cup dan Optic Disk**

##### **Operasi Morfologi**

Pada proses ini dilakukan operasi morfologi erosi dan dilasi dengan menggunakan elemen struktur berbentuk disk dengan ukuran 25 untuk menutup gap pembuluh darah pada *red channel*.

##### **Standarisasi Citra**

Pada tahap ini, dilakukan proses standarisasi citra dengan pseudocode sebagai berikut.

|  |  |
| --- | --- |
| 1 | **Standarize(I,tipe)** |
| 2 | MEAN = mean(I) |
| 3 | Std = StandardDeviation(I) |
| 4 | if tipe == ‘disk’ |
| 5 | I = I – MEAN |
| 6 | Iter = 5 |
| 7 | else |
| 8 | I = I – (MEAN+Std) |
| 9 | Iter = 7 |
| 10 | For x = 1 to iter |
| 11 | I = I – minimum(nonzero(I)) |
| 12 | **return** arr\_norm |

#### Preprocessing Pembuluh Darah

Berikut alur proses pada tahap ini.

F:\5112100100\Semester 8\Tugas Akhir\Program\tugasAkhir\Bismillah Buku TA\bloodpre.png

Gambar 3.4 Diagram Blood Preprocessing

### Tahap Segmentasi

Pada tahap ini terbagi menjadi tiga bagian, yaitu segmentasi pembuluh darah, segmentasi optik disk, dan segmentasi optic cup.

1. **Segmentasi Optic Cup**

Pada tahap ini, dilakukan proses segmentasi optic cup dari citra yang telah melalui tahap preprocessing. Berikut diagram alurnya.

1. **Segmentasi Optic Disk**
2. **Segmentasi Pembuluh Darah**

Tabel 3.1 Contoh data terekam dan pola sebenarnya

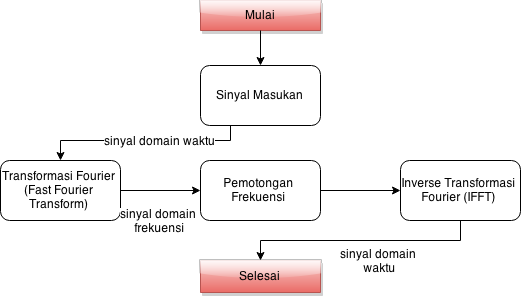
|  |  |
| --- | --- |
| **Gambar** | **Pola** |
|  | \_ \_ .  (G) |
|  | . \_  (A) |
|  | \_ . . .  (B) |
|  | \_ . \_ .  (C) |
|  | \_ . \_  (K) |

### Tahap Ekstraksi Fitur

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

#### Cup to Disk Ratio

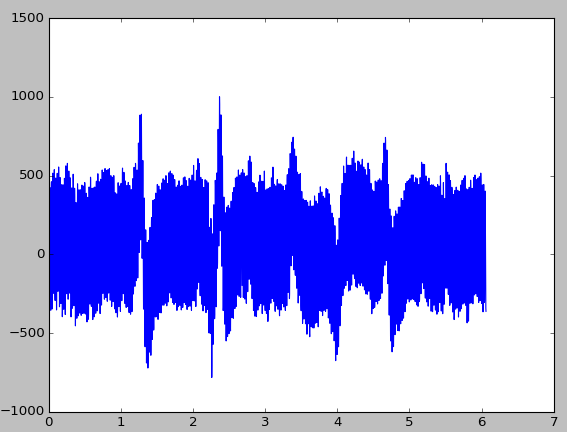
Pada tahap ini sinyal EEG yang telah didapatkan akan diekstrak menjadi sinyal EOG. Karena sinyal EEG yang didapatkan berasal dari satu elektroda yang ditempatkan pada posisi fp1 (kening kiri), pergerakan dari otot mata sangat mendominasi sinyal EEG yang didapatkan. Pola dari sinyal EEG yang didapatkan 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.*



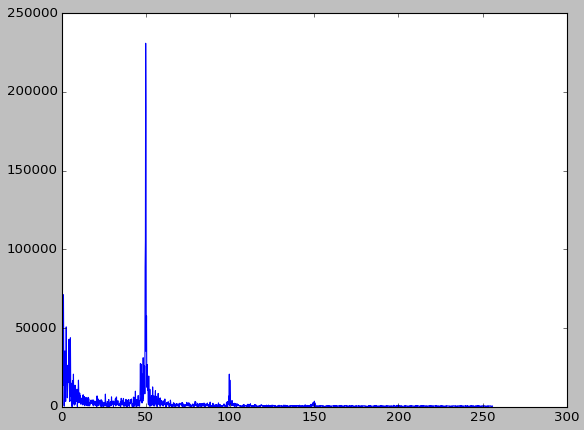
Gambar 3.7 Diagram alir *preprocessing*

##### 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

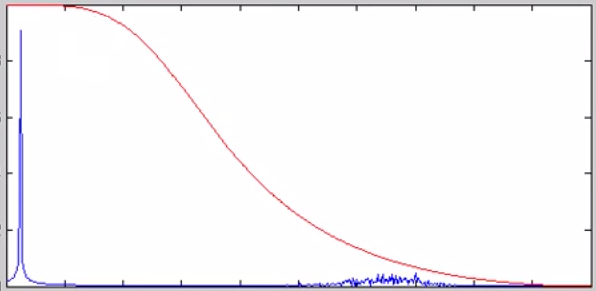


Gambar 3.9 Sinyal dalam domain frekuensi

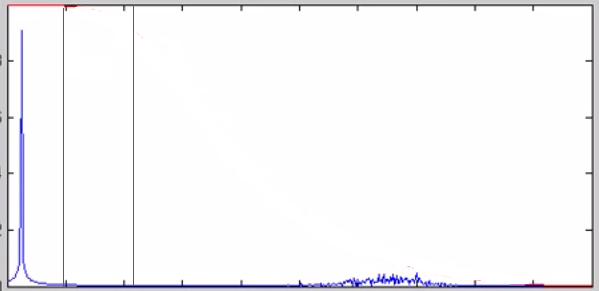
Implementasi metode *Discrete Fourier Transform* menggunakan *Fast Fourier Transform*. Fungsi *Fast Fourier Transform* yang digunakan pada pengerjaan tugas akhir ini didapat dari pustaka *scipy.fftpack* pada fungsi *fft*. Luaran dari fungsi ini adalah jajaran bilangan imajiner sejumlah jajaran sinyal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

##### *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 paramter distribusi beta sebagai faktor pengali. Gambar 3.10 dan 3.11 masing-masing menujukkan ilustrasi *bandpass* dan *cut-off frequency*.



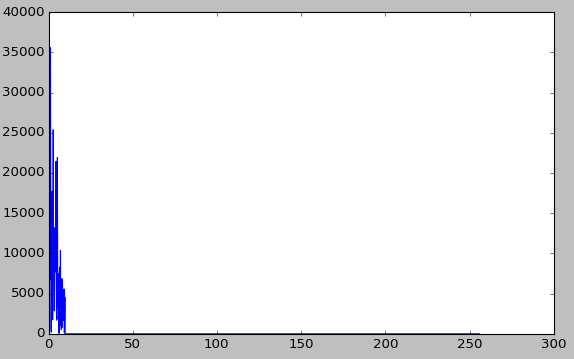
Gambar 3.10 Ilustrasi bandpass filtering



Gambar 3.11 Ilustrasi *cut-off* frequency

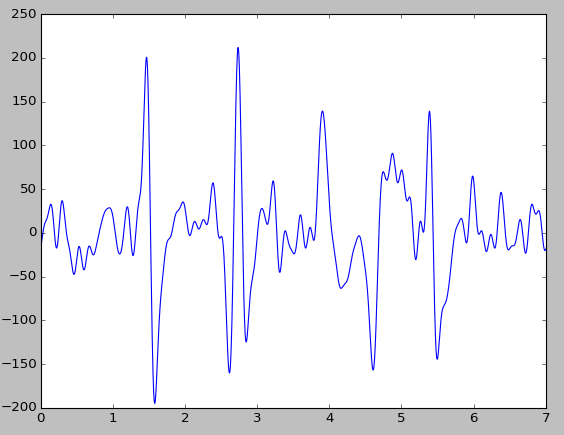
Nilai frekuensi dikalikan dengan nilai yang ditunjukkan pada garis merah (bernilai 0-1). Pada gambar 3.11 daerah yang berada diantara dua garis merah akan dikalikan dengan satu, sedangkan untuk daerah yang lain dikalikan 0 (*cut-off*).

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 interfensi 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

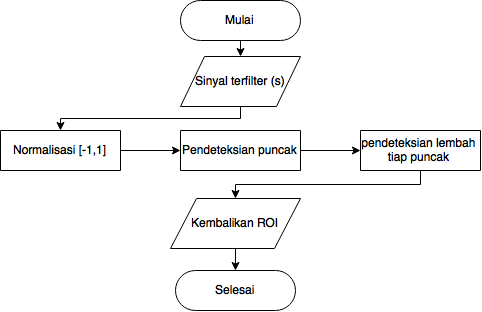
*Inverse Fourier Transform* akan diterapkan pada luaran domain frekuensi yang telah didapatkan sebelumnya. Implementasi *Inverse Fourier Transform* yang digunakan pada pengerjaan tugas akhir ini menggunakan pustaka *scipy.fftpack* pada fungsi *ifft*. Luaran dari fungsi ini adalah jajaran bilangan imajiner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



Gambar 3.13 Gambar sinyal luaran *preprocessing*

#### ISNT Neuro Retinal Rim

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 | **normalize(arr)** |
| 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 | **return** 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 | **findpeaks(arr)** |
| 2 | N = length(arr) |
| 3 | MEAN = mean(arr) |
| 4 | arr\_norm = normalize(arr) |
| 5 | diff = [0..N-1] |
| 6 | peaks = [] |
| 7 | **FOR** i **IN** [0..N-2] |
| 8 | diff[i] = arr[i] - arr[i + 1] |
| 9 | **END FOR** |
| 10 | **FOR** i **IN** [0..N-1] |
| 11 | **IF** diff[i] < 0 **AND** diff[i + 1] > 0 **AND** arr\_norm[i] > 0.3 + MEAN |
| 12 | peaks = [peaks i] |
| 13 | **END IF** |
| 14 | **END FOR** |
| 15 | **return** 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 masin-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 | **DetectROI(arr)** |
| 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 | **FOR** idx **IN** [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 | **END FOR** |
| 14 | **return** ROI |

Gambar 3.17 *Pseudocode* pendeteksian 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*.

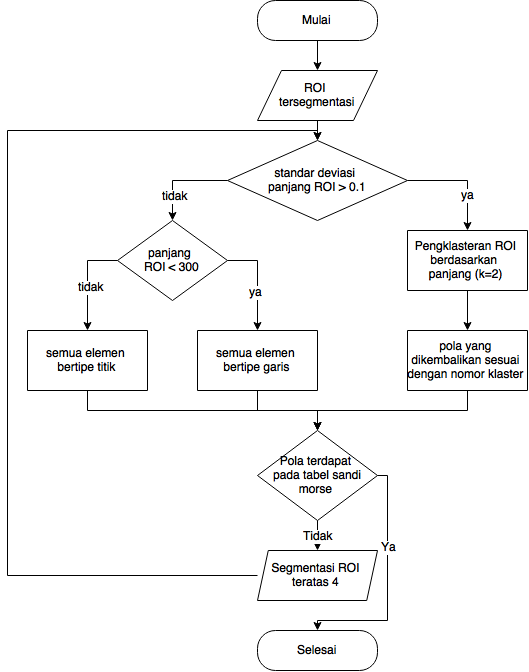
#### ISNT Pembuluh Darah

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). Pembedaan 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 tertandai, 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 | **Guesschar(arr)** | | 2 | VISIT = false | | 3 | ROIs = DetectROI(arr) | | 4 | *DO\_AGAIN::* | | 5 | len\_ROIs = [] | | 6 | result = [] | | 7 | **FOR** i **IN** [0..length(ROIs)] | | 8 | len\_ROIs = [len\_ROIs length(ROIs[i])] | | 9 | **END FOR** | | 10 | STD\_ROI = std(norm(len\_ROIs)) | | 11 | **IF** STD\_ROI > 0.1 | | 12 | idx = KMeans(STD\_ROI, 2) | | 13 | **FOR** i **IN** [0..length(len\_ROIs)-2] | | 14 | **IF** idx[i] != idx[i + 1] | | 15 | **IF** idx[i] > idx[i + 1] | | 16 | idx==0 is '\_' | | 17 | idx==1 is '.' | | 18 | **ELSE** | | 19 | idx==0 is '.' | | 20 | idx==1 is '\_' | | 21 | **END IF** | | 22 | **END IF** | | 23 | **END FOR** | | 24 | result = maskchar(result) | | 25 | **ELSE** | | 26 | **IF** len\_ROIs[0] > 300 | | 27 | result = maskchar(result, '\_') | | 28 | **ELSE** | | 29 | result = maskchar(result, '.') | | 30 | **END IF** | | 31 | **END IF** | | 32 | **IF** result **IN** MORSE\_TABLE | | 33 | return result | | 34 | **ELSE** | | 35 | ROIs = DetectROI(arr, *max* = 4) | | 36 | **IF** not VISIT | | 37 | VISIT = true | | 38 | GOTO DO\_AGAIN | | 39 | **END IF** | | 40 | **END IF** | |

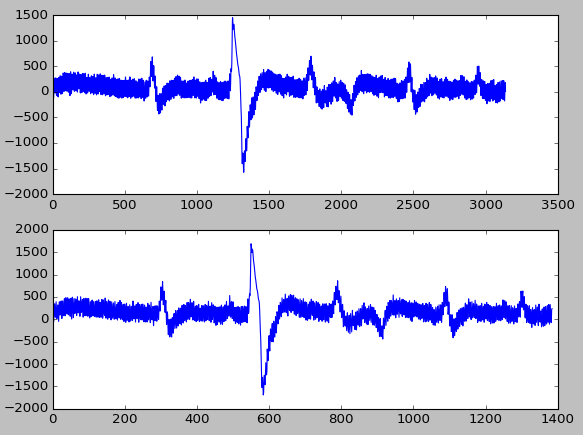
Gambar 3.19 *Pseudocode* tahap *postprocessing*

### Tahap Klasifikasi

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.

#### 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 | **KMeans*Template*Builder**(candidates, n\_*template*) |
| 2 | signal = [] |
| 3 | N = length(candidates) |
| 4 | e = min(n\_*template*, N) |
| 5 | cluster = KMeans(e, candidates) |
| 6 | **FOR** i **IN** [0..e]: |
| 7 | j = argwhere(cluster == i) |
| 8 | M = length(j) |
| 9 | cand = candidates[j] |
| 10 | *template* = cand[0] |
| 11 | **FOR** ix **IN** [1..M]: |
| 12 | pathDtw = dtw\_path(*template*, cand[ix]) |
| 13 | *template* = *template*[pathDtw] |
| 14 | **END FOR** |
| 15 | signal = [signal *template*] |
| 16 | **END FOR** |
| 17 | **return** 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 | **Hier*Template*Builder**(candidates, n\_*template*) |
| 2 | signal = candidates |
| 3 | N = length(signal) |
| 4 | n\_template = min(n\_*template*,      len(signal)) |
| 5 | **WHILE** length(signal) > n\_template: |
| 6 | minval = inf |
| 7 | a = -1 |
| 8 | b = -1 |
| 9 | **FOR** i **IN** [0..length(signal)]: |
| 10 | **FOR** j **IN** [i +              1..length(signal)]: |
| 11 | dist = dtw(signal[i],                  signal[j]) |
| 12 | **IF** dist < minval: |
| 13 | minval = d |
| 14 | a = i |
| 15 | b = j |
| 16 | **END IF** |
| 17 | **END FOR** |
| 18 | **END FOR** |
| 19 | siga = pop(signal, a) |
| 20 | sigb = pop(signal, b - 1) |
| 21 | pathDtw = dtw\_path(siga, sigb) |
| 22 | signal = [signal siga[pathDtw]] |
| 23 | **END WHILE** |
| 24 | **RETURN** 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.

#### 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.

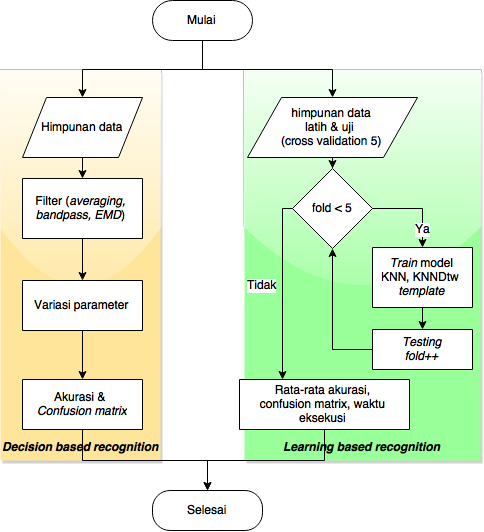
|  |  |
| --- | --- |
| 1 | **kNN**(test, *template*s, *template*sClass) |
| 2 | M = length(*template*s) |
| 3 | minDist = inf |
| 4 | *class* = None |
| 5 | **FOR** j **IN** [0..M] |
| 6 | dist = DynamicTimeWarping(test,  *template*s[j]) |
| 7 | **IF** minDist > dist |
| 8 | minDist = dist |
| 9 | *class* = *template*Class[j] |
| 10 | **END IF** |
| 11 | **END FOR** |
| 12 | **return** *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 *template*s.

## 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.

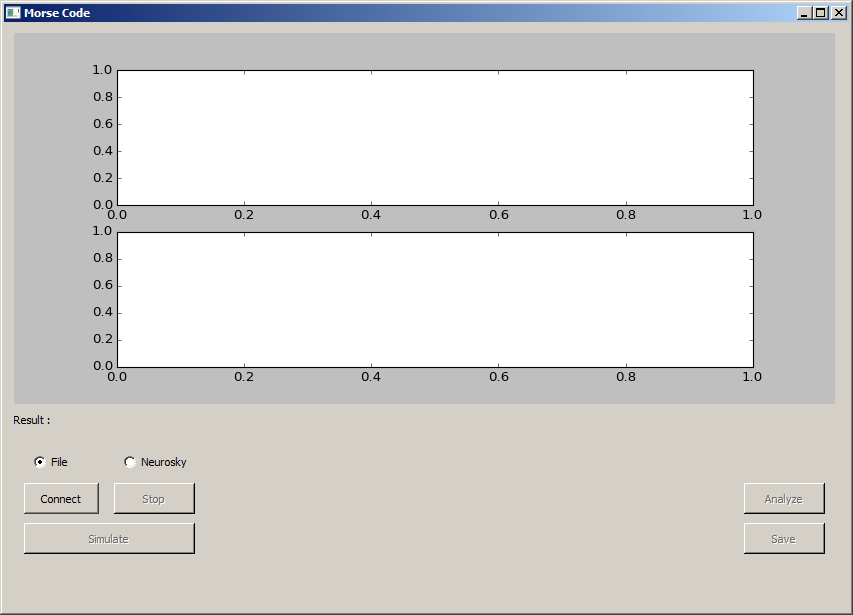


Gambar 3.24 Diagram alir evaluasi metode

## Perancangan Antarmuka Perangkat Lunak

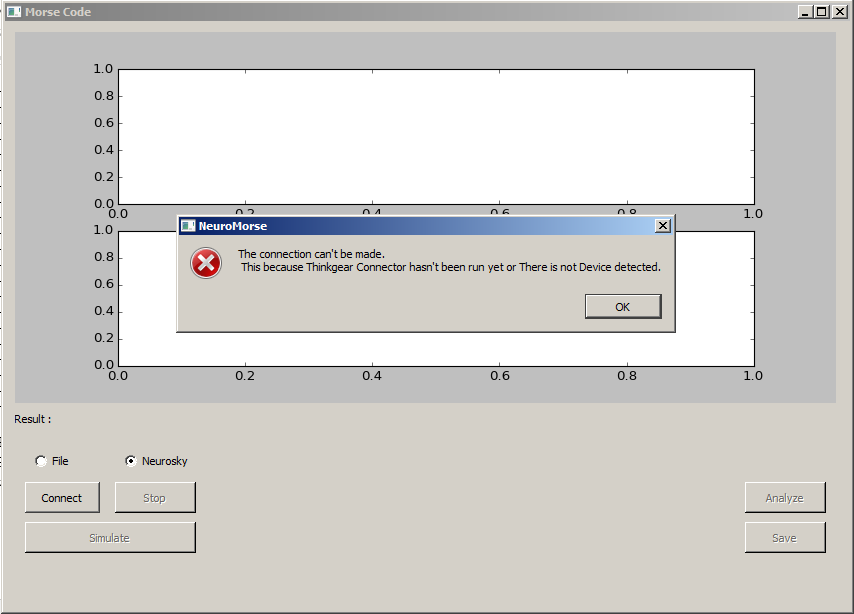
Perancangan antarmuka perangkat lunak merupakan penggambaran halaman antarmuka yang akan digunakan pengguna ketika berinteraksi dengan aplikasi. Implementasi perancangan antarmuka menggunakan pustaka *PyQt* pada bahasa pemrograman *Python*. Gambar 3.25 menunjukkan rancangan tatap muka program.

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



Gambar 3.25 Tatap muka utama

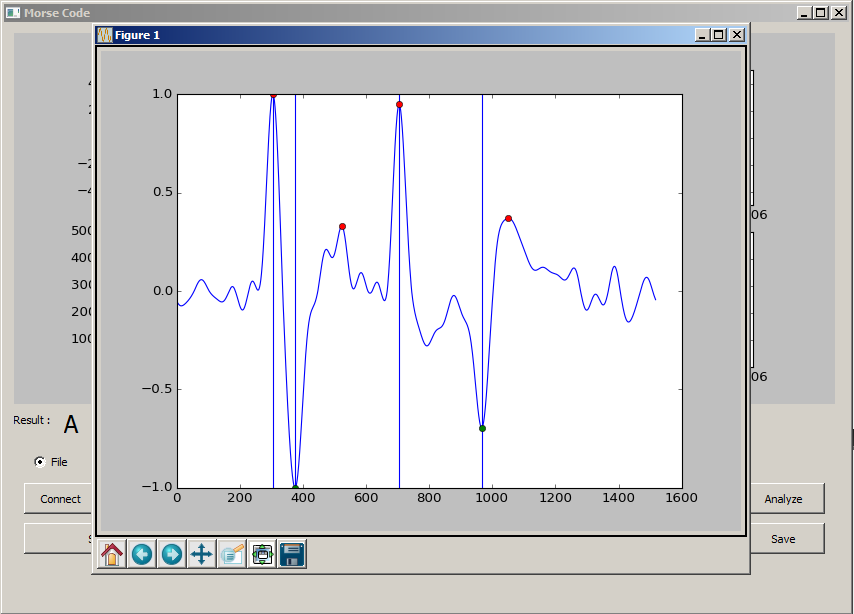
Tombol *Connect* digunakan untuk memulai simulasi. Jika mode masukan *File*, maka akan muncul kotak dialog buka yang bertujuan untuk memilih berkas yang akan disimulasikan. Jika mode masukan *NeuroSky* maka program akan menginisiasi koneksi dengan perangkat. Apabila tidak ditemukan perangkat *NeuroSky* 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.

## Lingkungan Implementasi

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

1. Perangkat Keras
2. Processor: Intel® Core™ i5-2410M CPU @ 2.30GHz
3. Memory (RAM) : 4.00 GB
4. Tipe Sistem: 64-bit
5. Perangkat Lunak
6. Sistem operasi : *Window*s 7 Professional 64 bit.
7. *Python* 2.7.6.
8. Sublime Text Editor.
9. Pustaka *numpy, sklearn, scipy, mlpy, pyqt, dan matplotlib*.

## 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 menujukkan kode program yang berfungsi melakukan pengambilan data.

|  |  |
| --- | --- |
| 1 | *def* \_\_read\_data(*self*, *a*, *b*): |
| 2 | while self.\_\_recording: |
| 3 | buff = self.c\_sock.recv(8192) |
| 4 | if self.\_\_islogging: |
| 5 | self.temp += buff |
| 6 | curr\_time = time() |
| 7 | parts = buff.split('\r') |
| 8 | for part in parts: |
| 9 | if part.startswith('{'): |
| 10 | try: |
| 11 | parseObj = json.loads(part) |
| 12 | for key in parseObj.keys(): |
| 13 | if key == 'eSense' or key == 'eegPower': |
| 14 | for key2 in parseObj[key].keys(): |
| 15 | if len(self.data[key2]) > 4000000: |
| 16 | self.data[key2].pop(0) |
| 17 | self.data[key2].append((curr\_time,            parseObj[key][key2])) |
| 18 | else: |
| 19 | if len(self.data[key]) > 4000000: |
| 20 | self.data[key].pop(0) |
| 21 | self.data[key].append((curr\_time,           parseObj[key])) |
| 22 | except: |
| 23 | print key |
| 24 | try: |
| 25 | sleep(0.05) |
| 26 | except: |
| 27 | pass |

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*.

## 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 | *class* DataStream(*HAbstractStream*): |
| 2 | *def* \_\_init\_\_(*self*, *filename*, *window\_length* = 500): |
| 3 | HAbstractStream.\_\_init\_\_(self) |
| 4 | self.*window*\_length = *window*\_length |
| 5 | self.filename = filename |
| 6 | self.data = np.append(np.array([np.nan for i    in range(100)]),    self.\_\_open\_data(self.filename)) |
| 7 | *def* next(*self*): |
| 8 | if self.current\_cursor < len(self.data): |
| 9 | start = self.current\_cursor |
| 10 | self.current\_cursor += self.delta |
| 11 | temp = len(self.data[start:start +     self.*window*\_length]) |
| 12 | if temp == 0: |
| 13 | return [] |
| 14 | return np.append(self.data[start:start +     self.*window*\_length], np.array([np.nan for i     in range(self.*window*\_length -temp)])) |
| 15 | else: |
| 16 | print '0' |
| 17 | return [] |
| 18 | *class* NeuroStream(*HAbstractStream*): |
| 19 | *def* next(*self*): |
| 20 | self.currentData =    np.array(self.neuroObj.get\_recorded(*withtime*=    False)[-self.*window*\_length:]) |
| 21 | return self.currentData |

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* berfunsi untuk menangani aliran data dari perangkat. Fungsi *next* digunakan untuk mendapatkan data setiap permunculan per detik.

## 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 | *class* Monitor(*FigureCanvas*): |
| 2 | *def* timerEvent(*self*, *evt*): |
| 3 | dummy = self.\_\_obj.next() |
| 4 | print len(dummy) |
| 5 | if len(dummy) == 0: |
| 6 | self.\_\_obj.reset() |
| 7 | self.endSimulation() |
| 8 | return |
| 9 | y = bandpass(dummy, 0.6, 20) |
| 10 | if len(y) == 0: |
| 11 | self.\_\_obj.reset() |
| 12 | self.endSimulation() |
| 13 | return |
| 14 | x = np.arange(len(y)) |
| 15 | self.axes.clear() |
| 16 | self.axes.plot(x, y) |
| 17 | self.axes2.clear() |
| 18 | self.axes2.plot(np.abs(fft(y[y!=  np.nan])[:len(y) / 2])) |
| 19 | self.axes.set\_ylim(-500, 500) |
| 20 | self.axes2.set\_ylim(0, 50000) |
| 21 | self.infig.canvas.draw() |
| 22 | *def* setDataStreamer(*self*, *obj*): |
| 23 | self.\_\_obj = obj |
| 24 | *def* startSimulation(*self*, *time*): |
| 25 | self.timer = self.startTimer(time) |
| 26 | *def* endSimulation(*self*): |
| 27 | self.killTimer(self.timer) |

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.

## 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 | MAX\_DOT\_THRES = 300 |
| 2 |  |
| 3 | *def* guesschar(*s*): |
| 4 | r = populate\_ROI(s) |
| 5 | k = np.array([len(i) for i in r],  *dtype*=np.float64) |
| 6 | rep = {1:'', 0:''} |
| 7 | predict = '' |
| 8 | if np.std(k / np.max(k)) > 0.1: |
| 9 | idx = KMeans(*n\_clusters*=2).      fit\_predict(np.array([[len(i) for i in r],      np.zeros((1, len(r)))[0]],  *dtype*=np.float64).T) |
| 10 | for i in range(len(idx) - 1): |
| 11 | if idx[i] != idx[i + 1] and abs(len(r[i])        - len(r[i + 1])) > 0: |
| 12 | if len(r[i]) > len(r[i + 1]): |
| 13 | rep[idx[i]] = '\_' |
| 14 | rep[idx[i + 1]] = '.' |
| 15 | else: |
| 16 | rep[idx[i]] = '.' |
| 17 | rep[idx[i + 1]] = '\_' |
| 18 | break |
| 19 | predict\_candidate = [rep[i] for i in idx] |
| 20 | predict\_final = [] |
| 21 | roi = populate\_ROI(s, *offset*=False) |
| 22 | for i in range(len(predict\_candidate)): |
| 23 | if predict\_candidate[i] == '.': |
| 24 | peaks = findpeaks(roi[i]) |
| 25 | if len(peaks) == 0: |
| 26 | predict\_final.append(predict\_candidate            [i]) |
| 27 | else: |
| 28 | predict\_final.append(predict\_candidate[i          ]) |
| 29 | predict = ' '.join(predict\_final) |
| 30 | else: |
| 31 | if min([len(i) for i in r]) < MAX\_DOT\_THRES: |
| 32 | predict = ' '.join(['.' for i in        range(len(r))]) |
| 33 | else: |
| 34 | predict = ' '.join(['\_' for i in        range(len(r))]) |
| 35 | return predict |
| 36 | *def* exceptional(*s*): |
| 37 | r = populate\_ROI(s, *e* = 4) |
| 38 | k = np.array([len(i) for i in r],  *dtype*=np.float64) |
| 39 | rep = {1:'', 0:''} |
| 40 | predict = '' |
| 41 | if np.std(k / np.max(k)) > 0.1: |
| 42 | idx = KMeans(*n\_clusters*=2).      fit\_predict(np.array([[len(i) for i in r],      np.zeros((1, len(r)))[0]],  *dtype*=np.float64).T) |
| 43 | for i in range(len(idx) - 1): |
| 44 | if idx[i] != idx[i + 1] and abs(len(r[i])        - len(r[i + 1])) > 0: |
| 45 | if len(r[i]) > len(r[i + 1]): |
| 46 | rep[idx[i]] = '\_' |
| 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*.

## 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 pengenal.

|  |  |
| --- | --- |
| 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, *m*filter=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, *m*filter=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,  *random\_state*=3) |
| 26 | feat = [open\_data('New/' + n) for n in    filelist] |
| 27 | result = [] |
| 28 | for trains, tests in kf: |
| 29 | *template*s, chClass = [], [] |
| 30 | learning\_start = clock() |
| 31 | for ch in range(97, 97 + 26): |
| 32 | cand = [feat[i] for i in trains if        na[i][0] == chr(ch)] |
| 33 | if len(cand) != 0: |
| 34 | temp = *template*Builder(cand, *e* = a) |
| 35 | for t in temp: |
| 36 | *template*s.append(t) |
| 37 | chClass.append(chr(ch)) |
| 38 | learning\_end = clock() |
| 39 | print '*Template* created' |
| 40 | accur = 0 |
| 41 | avg\_detect\_time = 0.0 |
| 42 | for test in tests: |
| 43 | detect\_start = clock() |
| 44 | minVal, cla = np.inf, None |
| 45 | for i in range(len(*template*s)): |
| 46 | d = mlpy.dtw\_std(feat[test],  *template*s[i]) |
| 47 | if minVal > d: |
| 48 | minVal = d |
| 49 | cla = chClass[i] |
| 50 | detect\_end = clock() |
| 51 | if cla == na[test][0]: |
| 52 | accur += 1 |
| 53 | avg\_detect\_time += (detect\_end –        detect\_start) |
| 54 | result.append((learning\_end –      learning\_start, avg\_detect\_time /      len(tests), *float*(accur) / len(tests) \*100)) |
| 55 | return result |
| 56 | *def* KNNEuclid(*filelist*, *a* = -1): |
| 57 | kf = KFold(len(na), *n\_folds*=5, *shuffle*=True,  *random\_state*=3) |
| 58 | feat = [open\_data('New/' + n) for n in    filelist] |
| 59 | result = [] |
| 60 | for trains, tests in kf: |
| 61 | *template*s, 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 = *template*Builder(cand, *e* = a) |
| 67 | for t in temp: |
| 68 | *template*s.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(*template*s)): |
| 77 | *template*s[i] = np.real          (ifft(fft(*template*s[i])          [:len(feat[tests[0]])])) |
| 78 | for i in range(len(*template*s)): |
| 79 | d = np.sqrt(np.sum((feat[test] –  *template*s[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 evalusi 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. Pembedaan 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.

***(Halaman ini sengaja dikosongkan)***

# 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.

## Lingkungan Uji Coba

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

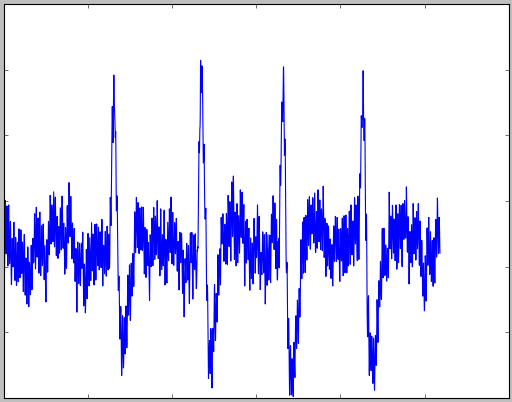
1. Perangkat Keras
   1. Processor: Intel® Core™ i5-2410M CPU @ 2.30GHz
   2. Memory (RAM) : 4.00 GB
   3. Tipe Sistem: 64-bit
2. Perangkat Lunak
3. Sistem operasi : *Window*s 7 Professional 64 bit.
4. *Python* 2.7.6.
5. Sublime Text Editor.
6. Pustaka *numpy, sklearn, scipy, mlpy, pyqt, dan matplotlib*.

## 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.

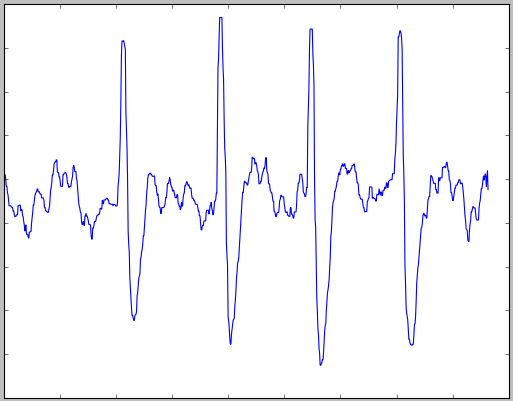
## Uji Coba Proses Pengenalan

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

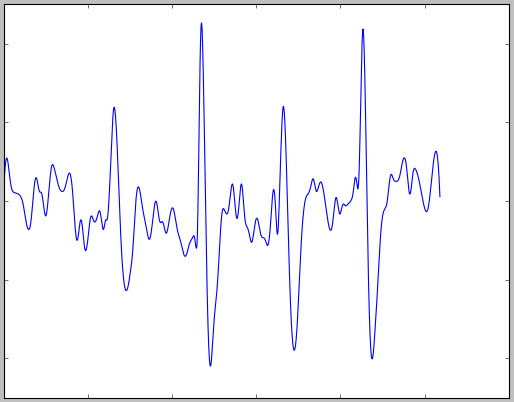


Gambar 5.1 Sinyal masukan sistem

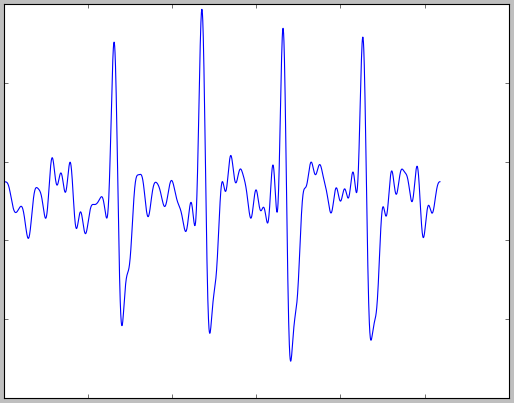
Pada tahap *preprocessing,* digunakan filter *window averaging*, *empirical mode decomposition* dan *bandpass*. Gambar 5.2, 5.3 dan 5.4 menunjukkan masing-masing dari hasil *preprocessing*.



Gambar 5.2 Hasil filter *window averaging*

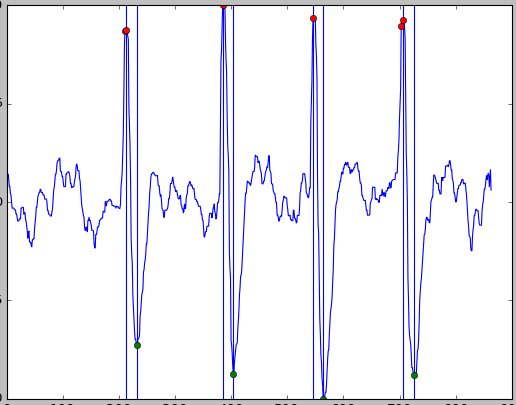


Gambar 5.3 Hasil filter *empirical mode decomposition*

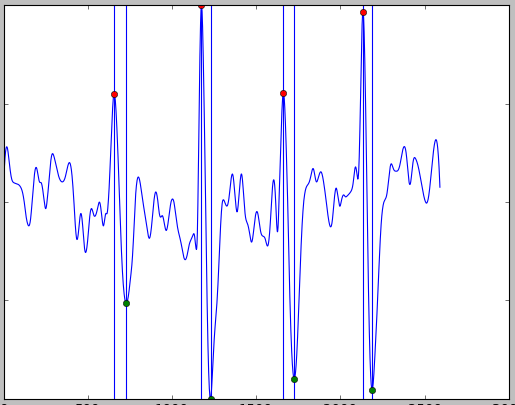


Gambar 5.4 Hasil filter *bandpass*

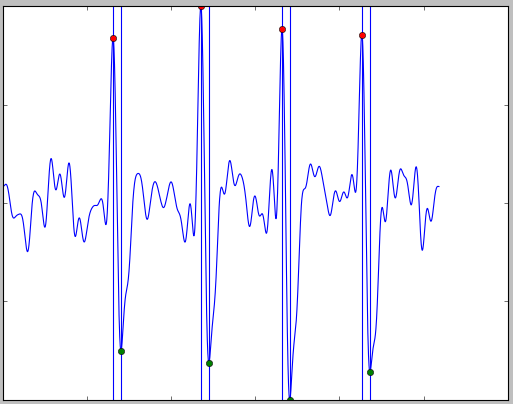
Setelah tahap ini, ROI pada sinyal dideteksi dengan menggunakan aturan-aturan yang telah dijelaskan pada Tahap *Postprocessing*. Gambar 5.5, 5.6 dan 5.7 menunjukkan hasil pendeteksian ROI untuk masing-masing filter.



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 (**∙ ∙ ∙ ∙)** yang merupakan huruf H. Hal seperti ini dimungkinkan karena sinyal memiliki bentuk morfologis yang bagus.

## 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.

### 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 pengenelan 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 |
| **Akurasi** | | | | **57,07** |
| **Waktu Pengenalan** | | | | **0,062** |

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 |
| **Akurasi** | | | | **69,38** |
| **Waktu Pengenalan** | | | | **0,062** |

Tabel 5.3 Hasil skenario 1 dengan Lebar *window* 30

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **71,01** |
| **Waktu Pengenalan** | | | | **0,08** |

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 |
| **Akurasi** | | | | **67,93** |
| **Waktu Pengenalan** | | | | **0,11** |

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 |
| **Akurasi** | | | | **64,31** |
| **Waktu Pengenalan** | | | | **0,11** |

Berdasarkan hasil yang telah didapatkan pada uji coba skenario 1, didapat bahwa nilai akurasi tertinggi adalah 71,01% dengan panjang window 30. Besar *overlaping 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 *Window*** | **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 Window** | **Hasil Pengolahan** |
| Asli |  |
| 10 |  |
| 20 |  |
| 30 |  |
| 40 |  |
| 50 |  |

### 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 *intrisic 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 |
| **Akurasi** | | | | **46,74** |
| **Waktu Pendeteksian** | | | | **0,15** |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **41,49** |
| **Waktu Pendeteksian** | | | | **0,15** |

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 |
| **Akurasi** | | | | **74,09** |
| **Waktu Pendeteksian** | | | | **0,15** |

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 |
| **Akurasi** | | | | **62,68** |
| **Waktu Pendeteksian** | | | | **0,15** |

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 |
| **Akurasi** | | | | **46,92** |
| **Waktu Pendeteksian** | | | | **0,15** |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **38,41** |
| **Waktu Pendeteksian** | | | | **0,15** |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **12,50** |
| **Waktu Pendeteksian** | | | | **0,15** |

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 |
| **Akurasi** | | | | **9,42** |
| **Waktu Pendeteksian** | | | | **0,15** |

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 |  |
| [1, -1] |  |
| [1, -2] |  |
| [2, -1] |  |
| [2, -2] |  |
| [3, -1] |  |
| [3, -2] |  |
| [4, -1] |  |
| [4, -2] |  |

### 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 |
| **Akurasi** | | | | **89,49** |
| **Waktu Pendeteksian** | | | | **0,03** |

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 |
| **Akurasi** | | | | **88,95** |
| **Waktu Pendeteksian** | | | | **0,03** |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **88,04** |
| **Waktu Pendeteksian** | | | | **0,03** |

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 |
| **Akurasi** | | | | **73,19** |
| **Waktu Pendeteksian** | | | | **0,03** |

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 |
| **Akurasi** | | | | **80,25** |
| **Waktu Pendeteksian** | | | | **0,03** |

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 |
| **Akurasi** | | | | **5,25** |
| **Waktu Pendeteksian** | | | | **0,03** |

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

|  |  |  |
| --- | --- | --- |
| **Batas Frekuensi** | **Akurasi (%)** | **Rata-rata waktu pengenalan (det)** |
| [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 |  |
| [0; 10] |  |
| [0,5; 10] |  |
| [1; 10] |  |
| [0; 20] |  |
| [5; 20] |  |

### 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 pengenal. 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 latih 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 |
| **Akurasi** | | | | **41,85** |
| **Waktu Deteksi** | | | | **0,5** |
| **Waktu Pembangkitan Model** | | | | **-** |

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 |
| **Akurasi** | | | | **5,98** |
| **Waktu Deteksi** | | | | **0,04** |
| **Waktu Pembangkitan Model** | | | | **30,2** |

Tabel 5.28 Hasil dengan jumlah *template* 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **17,21** |
| **Waktu Deteksi** | | | | **0,18** |
| **Waktu Pembangkitan Model** | | | | **16,15** |

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 |
| **Akurasi** | | | | **40,22** |
| **Waktu Deteksi** | | | | **0,51** |
| **Waktu Pembangkitan Model** | | | | **4,28** |

Tabel 5.30, 5.31 dan 5.32 masing-masing adalah hasil dari variasi *template* dengan metode pembangkitan *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 |
| **Akurasi** | | | | **4,89** |
| **Waktu Deteksi** | | | | **0,04** |
| **Waktu Pembangkitan Model** | | | | **1011,64** |

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 |
| **Akurasi** | | | | **17,21** |
| **Waktu Deteksi** | | | | **0,18** |
| **Waktu Pembangkitan Model** | | | | **910,6** |

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 |
| **Akurasi** | | | | **40,40** |
| **Waktu Deteksi** | | | | **0,5** |
| **Waktu Pembangkitan Model** | | | | **257,76** |

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 *Temp-late*** | ***K-Means*** | | | ***Hierarchical Clustering*** | | |
| **Rata-rata Akurasi (%)** | **Waktu Penge-nalan (detik)** | **Pem-bang-kitan Model (detik)** | **Rata-rata Akurasi (%)** | **Waktu Penge-nalan (detik)** | **Pem-bang-kitan 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 |

### 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 |
| **Akurasi** | | | | **96,56** |
| **Waktu Pendeteksian** | | | | **10,1** |
| **Waktu Pembangkitan Model** | | | | **-** |

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 |
| **Akurasi** | | | | **52,90** |
| **Waktu Pendeteksian** | | | | **2,65** |
| **Waktu Pembangkitan Model** | | | | **32,29** |

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 |
| **Akurasi** | | | | **72,28** |
| **Waktu Pendeteksian** | | | | **6,83** |
| **Waktu Pembangkitan Model** | | | | **17,31** |

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 |
| **Akurasi** | | | | **95,29** |
| **Waktu Pendeteksian** | | | | **11,94** |
| **Waktu Pembangkitan Model** | | | | **4,21** |

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

Tabel 5.38 Hasil dengan jumlah *template* 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **Benar** | **Salah** | **Klasifikasi Salah** | **Akurasi** |
| 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 |
| **Akurasi** | | | | **56,34** |
| **Waktu Pendeteksian** | | | | **1,87** |
| **Waktu Pembangkitan Model** | | | | **619,7** |

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 |
| **Akurasi** | | | | **69,75** |
| **Waktu Pendeteksian** | | | | **4,3** |
| **Waktu Pembangkitan Model** | | | | **577,31** |

Tabel 5.40 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 | 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 |
| **Akurasi** | | | | **94,75** |
| **Waktu Pendeteksian** | | | | **8,61** |
| **Waktu Pembangkitan Model** | | | | **181,49** |

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

Tabel 5.41 Rekapitulasi skenario 5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Jumlah *Temp-late*** | ***K-Means*** | | | ***Hierarchical Clustering*** | | |
| **Rata-rata Akurasi (%)** | **Waktu Penge-nalan (detik)** | **Pem-bang-kitan Model (detik)** | **Rata-rata Akurasi (%)** | **Waktu Penge-nalan (detik)** | **Pem-bang-kitan 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.

## Evaluasi

Data yang telah direkam tervariasi untuk masing-masing kelas, sehingga semakin banyak himpunan data latihnya akan semakin memperbagus 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 kurangn 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 penggunakan pengenal berbasis aturan adalah tidak diperlukan waktu untuk pembentukan model dan durasi waktu pendeteksian sinyal masukan yang singkat. Sedangkan nilai akurasi tertinggi pada pengenal 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*

|  |  |  |
| --- | --- | --- |
| ***Preprocessing*** | **Kelebihan** | **Kekurangan** |
| Averaging | mudah dipahami, menghilangkan *noise* mayor. | Rentan terhadap *noise* minor. |
| EMD | Menghilangkan *noise* minor pada indeks tertentu. | Rentan terhadap pelengkungan *baseline* karena penyertaan residu. |
| Bandpass | Tidak merubah posisi *baseline*, menghilangkan *noise* 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.

***(Halaman ini sengaja dikosongkan)***

# 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.

## 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 interfensi 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.

## 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 Elektroensephalogram (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. [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. [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. |

***(Halaman ini sengaja dikosongkan)***

# 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** | | | | | | | | | | | | | |
| **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** | **#** |
| 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** |
| ***GROUNDTRUTH*** | **A** |  | 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 |

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** | | | | | | | | | | | | |
| **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 | 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** |
| ***GROUNDTRUTH*** | **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 |

Lampiran 7.5 *Confusion matrix* uji coba skenario 5 semua *training set*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | 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 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 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 | 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 | 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 | 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 18 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 21 | 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 | 0 | 0 | 21 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 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 | 21 | 0 |
| 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 |

Lampiran 7.6 Contoh hasil IMF

|  |  |
| --- | --- |
| Indeks IMF ke | Gambar |
| Asli |  |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |

# 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).