



TESIS - SS235401

***MULTICLASS SUPPORT VECTOR MACHINE DENGAN
GAUSSIAN TIME WARP EDIT DISTANCE KERNEL UNTUK
KLASIFIKASI KONDISI MENTAL BERDASARKAN
REKAMAN ELEKTROENSEFALOGRAM***

NURIS FARANDHINI HIDAYAT
NRP 6003221024

Dosen Pembimbing
Santi Wulan Purnami, S.Si., M.Si., Ph.D.
Shofi Andari, S.Stat., M.Si., Ph.D.

Program Studi Magister Statistika
Departemen Statistika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya
2024



TESIS - SS235401

***MULTICLASS SUPPORT VECTOR MACHINE DENGAN GAUSSIAN
TIME WARP EDIT DISTANCE KERNEL UNTUK KLASIFIKASI
KEADAAN MENTAL BERDASARKAN REKAMAN
ELEKTROENSEFALOGRAM***

NURIS FARANDHINI HIDAYAT
NRP 6003221024

Dosen Pembimbing
Santi Wulan Purnami, S.Si., M.Si., Ph.D.
Shofi Andari, S.Stat., M.Si., Ph.D.

Program Studi Magister Statistika
Departemen Statistika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya
2024



THESIS - SS235401

**MULTICLASS SUPPORT VECTOR MACHINE WITH GAUSSIAN
TIME WARP EDIT DISTANCE KERNEL FOR MENTAL STATE
CLASSIFICATION BASED ON ELECTROENCEPHALOGRAM**

NURIS FARANDHINI HIDAYAT
NRP 6003221024

Advisors

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

Shofi Andari, S.Stat., M.Si., Ph.D.

Master of Statistics Study Program
Departemen of Statistics
Faculty of Science and Data Analytics
Institut Teknologi Sepuluh Nopember
Surabaya
2024

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Statistika (M.Stat)

di

Institut Teknologi Sepuluh Nopember

Oleh:

NURIS FARANDHINI HIDAYAT

NRP: 6003221024

Tanggal Ujian: 31 Juli 2024

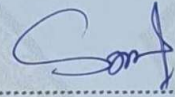
Periode Wisuda: September 2024

Disetujui oleh:

Pembimbing:

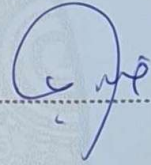
1. Santi Wulan Purnami, S.Si., M.Si., Ph.D.

NIP: 19720923 199803 2 001



2. Shofi Andari, S.Stat., M.Si., Ph.D.

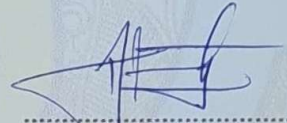
NIP: 19871207 201404 2 001



Penguji:

1. Dr.rer.pol. Dedy Dwi Prastyo, S.Si, M.Si

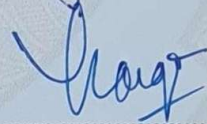
NIP: 19831204 200812 1 002



2. Prof. Dr. dr. Margarita M. Maramis, Sp.KJ.,

Subsp.B.P.(K)., FISCN

NIP: 19610819 198803 2 005



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

Dr. Dra. Kartika Fithriasari, M.Si

NIP: 19691212 199303 2 002



Multiclass Support Vector Machine dengan *Gaussian Time Warp Edit Distance Kernel* untuk Klasifikasi Kondisi Mental berdasarkan Rekaman Elektroensefalogram

Nama mahasiswa/NRP : Nuris Farandhini Hidayat / 6003221024

Departemen : Statistika

**Dosen Pembimbing : Santi Wulan Purnami, S.Si., M.Si., Ph.D.
Shofi Andari, S.Stat., M.Si., Ph.D.**

ABSTRAK

Electroencephalography (EEG) merupakan suatu metode untuk merekam aktivitas listrik otak yang dilakukan pada permukaan kulit kepala. EEG dapat memberikan informasi dengan resolusi temporal yang tinggi karena berskala detik atau milidetik. Hal ini serupa dengan kerangka waktu aktivitas saraf. EEG sudah digunakan pada layanan kesehatan untuk tujuan diagnostik dan ke depannya diharapkan penggunaannya dapat diaplikasikan lebih luas dengan efektivitas biaya. Pemeriksaan kondisi pasien melibatkan penilaian apakah pasien sedang dalam suatu keadaan mental tertentu, yaitu kondisi konsentrasi, netral atau dalam keadaan rileks. Terdapat lima jenis gelombang berdasarkan frekuensinya, di antaranya: gelombang *delta* (< 4 Hz), *theta* (4-8 Hz), *alpha* (8-13 Hz), *beta* (13-30 Hz) dan *gamma* (>50 Hz). Penggunaan metode *Multiclass SVM* diketahui dapat memberikan nilai akurasi yang tinggi untuk klasifikasi data dengan kategori lebih dari dua. Data EEG adalah data yang berbasis waktu, sehingga diperlukan metode yang tepat untuk mengatasi permasalahan deret waktu tersebut. Pada penelitian ini akan dilakukan klasifikasi keadaan mental terdiri dari tiga kelas yaitu keadaan relaks, netral dan konsentrasi berdasarkan hasil dari data rekaman EEG dengan menggunakan metode klasifikasi *Multiclass SVM* dengan kernel *Gaussian Time Warp Edit Distance* (GTWED). Hasil klasifikasi *multiclass SVM* pendekatan OAA memiliki AUC sebesar 0,561 dan AUC pada pendekatan OAO adalah sebesar 0,651, kedua pendekatan menunjukkan bahwa hasil klasifikasi yang kurang baik, namun berdasarkan nilai AUC pendekatan OAO lebih baik dibandingkan dengan pendekatan OAA.

Kata Kunci : *Electroencephalography*, Kondisi mental, *Multiclass SVM*, *Time series kernel*

Multiclass Support Vector Machine with Gaussian Time Warp Edit Distance Kernel Mental State Classification based on Electroencephalogram

Student Name/NRP : Nuris Farandhini Hidayat / 6003221024
Departement : Statistics
Advisor : Santi Wulan Purnami, S.Si., M.Si., Ph.D.
Shofi Andari, S.Stat., M.Si., Ph.D.

ABSTRACT

Electroencephalography (EEG) is a method for recording the electrical activity of the brain carried out on the surface of the scalp. EEG can provide information with high temporal resolution because the time scale can be seconds or milliseconds, which is the same as the time frame of neural activity. EEG has been used in many clinics for diagnostic purposes and in the future it is hoped that its use can be applied more widely with cost effectiveness. Examination of the patient's condition involves assessing whether the patient is in a certain mental state, namely a state of concentration, neutral or relaxed. There are five types of waves based on their frequency, including: delta waves (< 4 Hz), theta (4-8 Hz), alpha (8-13 Hz), beta (13-30 Hz) and gamma (> 50 Hz). The use of the Multiclass SVM method is known to provide high accuracy values for classifying data with more than two categories. EEG data is time-based data, so an appropriate method is needed to overcome this time series problem. In this research, mental state classification will be carried out into three classes, namely relaxed, neutral and concentrated states based on the results of EEG recording data using the Multiclass SVM classification method with the Gaussian Time Warp Edit Distance (GTWED) kernel. The multiclass SVM classification results of the OAA approach have an AUC of 0.561 and the AUC in the OAO approach is 0.651, both approaches show that the classification results are not good, but based on the AUC value the OAO approach is better than the OAA approach.

Keywords : Electroencephalography, Mental states, Multiclass SVM, Time series kernel

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT karena dengan rahmat, karunia, serta taufik dan hidayah-Nya penulis dapat menyelesaikan laporan tesis yang berjudul “***Multiclass Support Vector Machine dengan Gaussian Time Warp Edit Distance Kernel untuk Klasifikasi Keadaan Mental berdasarkan Rekaman Sinyal Elektroensefalogram***”. Dalam penyusunan dan penyelesaian laporan tesis ini, penulis mendapat bantuan dan dukungan dari berbagai pihak, oleh sebab itu penulis mengucapkan terima kasih kepada :

1. Santi Wulan Purnami, S.Si., M.Si., Ph.D. selaku dosen pembimbing 1 sekaligus selaku sekretaris Sekretaris Departemen I (Bidang Akademik, Kemahasiswaan, Penelitian dan Pengabdian Kepada Masyarakat) yang telah membimbing dan mengarahkan serta memberikan dukungan pada penulis untuk dapat menyelesaikan proposal tesis.
2. Shofi Andari, S.Stat., M.Si., Ph.D. selaku dosen pembimbing 2 yang telah membimbing dan mengarahkan serta memberikan dukungan pada penulis untuk dapat menyelesaikan proposal tesis.
3. Dr. Dra. Kartika Fithriasari, M.Si. selaku Kepala Departemen Statistika yang telah memberikan fasilitas untuk penyelesaian proposal tesis.
4. Ibu Prof. Dr. Vita Ratnasari, M.Si selaku Sekretaris Departemen II (Bidang Sumber Daya Keuangan, Sumber Daya Manusia, dan Sarana Prasarana)
5. Dr.rer.pol. Dedy Dwi Prastyo, M.Si selaku Ketua Program Studi Pascasarjana Statistika, Fakultas Sains dan Analitika Data Institut Teknologi Sepuluh Nopember sekaligus dosen penguji. Prof. Dr. dr. Margarita M. Maramis, Sp.KJ., Subsp.B.P.(K)., FISCAM selaku dosen penguji yang telah memberikan bimbingan, kritik dan saran untuk kesempurnaan Tugas Akhir ini.
6. Prof. Dr. I Nyoman Budiantara, M.Si selaku dosen wali yang telah memberikan motivasi, saran belajar dan dukungan selama proses perkuliahan.
7. Dra. Christina Ririn, M.Si., Psikolog selaku operator pada tes EEG di klinik swasta milik Prof. Dr. dr. Margarita M. Maramis, Sp.KJ., Subsp.B.P.(K)., FISCAM.
8. Seluruh dosen Departemen Statistika ITS yang telah memberikan bekal ilmu selama menempuh masa perkuliahan, sehingga ilmu tersebut dapat diterapkan dalam tugas akhir ini.
9. Seluruh karyawan Departemen Statistika ITS yang telah membantu kelancaran dalam administrasi Tugas Akhir.
10. Suami, kedua orang tua, dan keluarga yang selalu memberikan doa, bimbingan, dukungan, kasih sayang serta kesabarannya dalam mendidik baik secara materiil, moril, maupun spiritual.
11. Annisa, Saffanah, Farel, Aldo, Aufar, teman-teman mahasiswa pascasarjana tahun 2022 serta sahabat-sahabat yang selalu mendukung dan memberikan saya semangat dalam menjalankan masa perkuliahan dan dalam penyelesaian laporan tesis ini.
12. Pimpinan dan seluruh rekan-rekan Urban Group yang selalu memberikan saya semangat dan motivasi dalam menyelesaikan proposal tesis ini.
13. Semua pihak yang telah membantu dalam penyusunan draft tesis ini, baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu oleh penulis. Semoga bantuan dan kerjasama yang telah dilakukan mendapat pahala yang setimpal dari Allah SWT.

Penulis menyadari bahwa laporan tesis ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun agar dapat dijadikan pertimbangan untuk penelitian selanjutnya.

Surabaya, Juli 2024

Penulis

DAFTAR ISI

Halaman	
HALAMAN JUDUL.....	ii
TITLE PAGE.....	iii
LEMBAR PENGESAHAN TESIS	iviii
ABSTRAK.....	v
ABSTRACT	xi
KATA PENGANTAR.....	viiix
DAFTAR ISI	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Metode Filterisasi <i>Finite Impulse Response</i>	9
2.3 <i>Time Epoching</i> dan <i>Overlapping</i>	10
2.4 Ekstraksi Fitur	11
2.4 <i>Support Vector Machine (SVM)</i>	15
2.4.1 SVM Kasus <i>Linearly Separable</i>	15
2.4.2 SVM Kasus <i>Linearly Non-Separable</i>	18
2.4.3 SVM Kasus <i>Non-Linearly Separable</i>	20
2.5 <i>Gaussian Elastic Metric Kernel</i>	21
2.5.1 <i>Time Warp Edit Distance</i>	22
2.6 Multiclass SVM.....	24
2.7 Evaluasi Performansi.....	26
2.8 <i>Mental State</i>	28

2.9	<i>Electroencephalography (EEG)</i>	29
BAB III METODOLOGI PENELITIAN		33
3.1	Kajian <i>Gaussian Time Warp Edit Distance Kernel (GTWED)</i> sebagai Fungsi Kernel pada <i>Multiclass SVM</i>	33
3.2	Klasifikasi Keadaan Mental dengan Menggunakan Hasil Rekaman EEG	34
3.2.1	Sumber Data	34
3.2.2	Variabel Penelitian.....	34
3.2.3	Langkah Analisis	36
BAB IV HASIL DAN PEMBAHASAN		43
4.1	Kajian <i>Gaussian Time Warp Edit Distance (GTWED)</i> sebagai Fungsi Kernel pada <i>Multiclass SVM</i>	43
4.2	Klasifikasi Kondisi Mental dengan menggunakan kernel <i>Gaussian Time Warp Edit Distance (GTWED)</i>	46
4.2.1	Karakteristik Data Awal	46
4.2.2	Filterisasi dan Segmentasi Data.....	49
4.2.3	Ekstraksi Fitur.....	51
4.2.4	Klasifikasi dengan menggunakan <i>Kernel Time Series GTWED</i>	58
BAB V KESIMPULAN DAN SARAN.....		63
5.1	Kesimpulan	63
5.2	Saran	64
DAFTAR PUSTAKA		65
LAMPIRAN.....		73
BIODATA PENULIS		97

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Klasifikasi Dua Kelas oleh Hyperplane.....	16
Gambar 2.2 Ilustrasi Data SVM Linearly Non-Separable.....	18
Gambar 2.3 Confusion Matrix.....	26
Gambar 2.4 Ilustrasi Penempatan Elektroda 10-20.....	30
Gambar 2.5 EEG Sensor O2.....	31
Gambar 3.1 Penempatan Elektroda	35
Gambar 3.2 Proses Pengolahan Sinyal EEG	36
Gambar 3.3 Proses Filterisasi	37
Gambar 3.4 Time Epoching	38
Gambar 3.5 Diagram Alir.....	42
Gambar 4.1 Ilustrasi Operasi Edit Distance	44
Gambar 4.2 Boxplot Beda Tegangan Sebelum Preprocessing Data berdasarkan Kondisi Mental	47
Gambar 4.3 Time Series Plot Sinyal Awal.....	48
Gambar 4.4 Plot PSD <i>Channel</i> O2 Sebelum Filterisasi (a) Kondisi Netral, (b) Kondisi Konsentrasi, (c) Kondisi Rileks	49
Gambar 4.5 Plot PSD (a) Sebelum Filterisasi, (b) Sub-Band Alfa, (c) Sub-Band Beta, (d) Sub-Band Beta, (e) Sub-Band Gamma, (f) Sub-Band Teta.....	50
Gambar 4.6 <i>Epoching</i> Data Orang ke-1 (a) Kondisi Netral, (b) Kondisi Rileks, (c) Kondisi Konsentrasi	51
Gambar 4.7 <i>Boxlot</i> Fitur Minimum	53
Gambar 4.8 <i>Boxlot</i> Fitur Median.....	54
Gambar 4.9 <i>Boxlot</i> Fitur Maksimum.....	54
Gambar 4.10 <i>Boxlot</i> Fitur Varians.....	55
Gambar 4.11 <i>Boxlot</i> Fitur <i>Skewness</i>	56
Gambar 4.12 <i>Boxlot</i> Fitur Kurtosis	56
Gambar 4.13 <i>Boxlot</i> Fitur Energi	57
Gambar 4.14 <i>Boxlot</i> Fitur Entropi	58
Gambar 4.15 ROC Pendekatan OAA	60
Gambar 4.16 ROC Pendekatan OAO	61

DAFTAR TABEL

	Halaman
Tabel 2.1 Penelitian Terdahulu	7
Tabel 2.2 Confusion Matrix dengan 3 Kelas Klasifikasi	27
Tabel 2.3 Perhitungan Sensitivitas dan Spesifisitas.....	27
Tabel 2.4 Tingkat Keakuratan berdasarkan Nilai AUC.....	28
Tabel 2.5 Jenis Sinyal EEG	31
Tabel 3.1 Struktur Data Awal Sinyal EEG Channel O2.....	35
Tabel 3.2 Struktur Data Hasil Filterisasi.....	37
Tabel 3.3 Struktur Data Hasil Segmentasi.....	38
Tabel 3.4 Struktur Data Ekstraksi Fitur	40
Tabel 3.5 Pembagian Data Training dan Testing.....	41
Tabel 4.1 Rangkuman Hasil Ekstraksi Fitur	51
Tabel 4.2 Confusion Matrix Kernel GTWED Pendekatan OAA	59
Tabel 4.3 Confusion Matrix Kernel GTWED Pendekatan OAO.....	59
Tabel 4.4 Keباikan Model Pendekatan OAA dan OAO	60

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Keterangan Legalitas Data.....	73
Lampiran 2. Data Sinyal Perekaman EEG.....	74
Lampiran 3. Data Sinyal Perekaman EEG.....	75
Lampiran 4 Syntax Preprocessing Data.....	76
Lampiran 5 Syntax <i>Ekstraksi Fitur</i>	79
Lampiran 6 Syntax Multiclass SVM	81
Lampiran 7 Syntax Visualisasi Data.....	86

BAB I PENDAHULUAN

1.1 Latar Belakang

Electroencephalography (EEG) merupakan suatu metode untuk merekam aktivitas listrik otak yang dilakukan pada permukaan kulit kepala (Khakim, 2021). Penggunaan EEG pada manusia pertama kali dilakukan oleh Hans Berger di tahun 1924, yang kemudian mulai menuai perhatian pada tahun 1930 (Borck, 2005). EEG dapat memberikan informasi dengan resolusi temporal yang tinggi karena berskala detik atau milidetik, dimana hal ini sama dengan kerangka waktu aktivitas saraf (Maharto, 2019).

Penelitian telah banyak dilakukan untuk pendeteksian ketidaknormalan sinyal otak menggunakan EEG, diantaranya pendeteksian *movement intention* (Lana, 2015), deteksi aktivitas bicara (Kocúrová, 2021) dan deteksi pasien epilepsi dengan sinyal EEG (Purnami, 2021; Wulandari, 2019; Islamiyah, 2020; Astuti, 2021). Selain itu data EEG dapat dikembangkan untuk penelitian pada *neuropsychiatric disorder* (De Gennaro, 2008). Penelitian mengenai data EEG untuk deteksi *neuropsychiatric disorder* di antaranya, deteksi pasien depresi dan normal (Nikmah, 2023), deteksi keadaan mental (Karimah, 2023), deteksi stress (TuerxunWaili, 2020) dan lain-lain. Variasi tegangan yang dihasilkan dari arus ionik yang mengalir di dalam neuron otak membantu dalam diagnosis gangguan mental (Yasin, 2021).

Kesehatan mental menjadi permasalahan global. Menurut WHO penyebab utama pada masalah kesehatan mental adalah karena tingginya tingkat stress yang mengakibatkan depresi (Saputra, 2023). Adanya pandemi COVID-19 menyebabkan kenaikan pada gangguan kesehatan mental dengan kenaikan secara global sebesar 25% selama tahun pertama pandemi (WHO, 2022). Namun, depresi seringkali tidak terdiagnosis dan tidak diobati karena terapi yang kurang efektif dan sumber daya kesehatan mental yang tidak memadai (Liu, 2022).

Identifikasi yang efektif untuk depresi berat sangat diperlukan dalam menentukan langkah yang efektif untuk pengobatan pasien. Alasan utama terjadinya kesalahan diagnosis adalah karena tidak adanya biomarker pada pasien *Major Depressive Disorder* (MDD) atau gangguan jiwa lainnya. Diagnosis depresi dilakukan berdasarkan kriteria klasifikasi depresi menggunakan *Diagnostic and Statistical Manual of Mental Disorders-Fourth Edition (DSM-IV)*, *Beck Depression Inventory (BDI)* dan *Hamilton Rating Scale (HAM-D)* yang bergantung pada beberapa kuesioner yang ditanyakan dan berdasarkan hasil observasi perilaku selama sesi wawancara antara pasien dan praktisi kesehatan (Maharto, 2019).

Akurasi dari penilaian perilaku klasik yang umumnya berbentuk tes yang menuntut jawaban tertulis mampu ditingkatkan dengan penggunaan alat ukur berbasis fisiologis (Kessels, 2019). *Electroencephalogram* (EEG) sudah digunakan di banyak klinik untuk tujuan diagnostic dan kedepannya diharapkan penggunaannya dapat diaplikasikan lebih luas dengan efektivitas biaya (Olbrich, 2013). Analisis berdasarkan kriteria yang objektif dapat membantu mendeteksi depresi pada tahap lebih awal agar mempermudah proses penyembuhan. Pemeriksaan kondisi pasien melibatkan penilaian apakah pasien sedang dalam suatu keadaan mental tertentu, yaitu kondisi konsentrasi, netral atau dalam keadaan rileks. *Mental State* (keadaan mental) mengacu pada keadaan pikiran yang mungkin terjadi dan dapat dilihat dari berbagai perpektif. Menurut berbagai penelitian, keadaan mental dapat diketahui berbasis kesadaran, berbasis intensionalitas dan fungsionalisme (Baron-Cohen S, 1994). Berdasarkan studi kebiasaan kognitif terdapat tiga keadaan mental yang dapat diklasifikasikan, yaitu keadaan relaks, netral dan konsentrasi (Bird, 2018).

Dominasi aktivitas yang sedang dilakukan adalah sebagai karakteristik gelombang sinyal EEG karena menghasilkan frekuensi yang berbeda. Terdapat lima jenis gelombang pada otak manusia yang diklasifikasikan berdasarkan frekuensinya, diantaranya: gelombang *delta* (< 4 Hz), *theta* (4-8 Hz), *alpha* (8-13 Hz), *beta* (13-30 Hz) *gamma* (>50 Hz). Gelombang *alpha* dominan muncul dalam keadaan sadar, mata tertutup dan kondisi rileks, gelombang *theta* dominan muncul ketika sedang tidur ringan, mengantuk atau stress, gelombang *delta* dominan muncul ketika seseorang sedang tidur nyenyak, gelombang *beta* dominan muncul saat orang tersebut sedang berpikir dan gelombang *gamma* dominan muncul ketika seseorang dalam kesadaran penuh (Karmila, 2016). Oleh karena itu sinyal EEG adalah suatu kumpulan data yang sangat kompleks, sehingga pemrosesan sinyal dengan menggunakan bantuan komputer diperlukan untuk mengklasifikasikan subjek yang normal dan depresi secara otomatis. Hal itu merupakan tantangan terbesar dalam pengaplikasian *brain-machine interface* untuk mendapatkan kesimpulan atau informasi mengenai bagaimana keadaan mental terdefiniskan oleh pola tertentu yang dihasilkan oleh aktivitas otak (Bird, 2018). Berdasarkan penelitian yang telah dilakukan oleh (Shen, 2008), diketahui bahwa area otak yang paling ideal untuk dilakukan deteksi kelelahan mental yaitu pada area oksipital. Hal ini juga disebutkan dalam review oleh (R. P. Balandong, 2018) bahwa kondisi seseorang sedang kelelahan mental berkaitan dengan seseorang dalam kondisi kantuk yang berlebih banyak dideteksi oleh peneliti lain mendekati area oksipital. Diketahui bahwa pada area oksipital terdapat bagian *channel* O1, Oz dan O2, dimana O1 merupakan letak *channel* oksipital bagian kiri, Oz berada di tengah, sedangkan O2

berada di bagian kanan. Berdasarkan penelitian oleh (Lim, Duc, Thien, Yuvaraj, & Fogarty, 2022) diketahui bahwa fluktuasi di *channel* O2 berefek positif pada fungsi kognitif yang tinggi, seperti visual, pemrosesan informasi, dan gerakan sukarela.

Penelitian terdahulu yang telah mempelajari bagaimana metode untuk klasifikasi keadaan mental dari hasil rekaman EEG, di antaranya: penelitian oleh You (2021) mengenai klasifikasi keadaan mental antara relaksasi dan konsentrasi berdasarkan sinyal EEG dengan menggunakan metode SVM dan *Back Propagation Neural Network* (BPNN), oleh Rahman (2022) mengenai deteksi keadaan mental berdasarkan hasil sinyal EEG menggunakan metode SVM, *Gradient Boosting* (GrB), *K-Nearest Neighbors* (KNN) dan *XGBoost* (XGB), dan oleh Xiaoon Li., et al (2014) dengan menggunakan *multiple kernel* diperoleh bahwa metode terbaik adalah dengan menggunakan SVM dengan tingkat akurasi secara berurutan adalah lebih dari 90%, 95,36%. dan diketahui bahwa pada klasifikasi dua kelas secara rata-rata diperoleh nilai akurasi sebesar 99,25 sedangkan untuk klasifikasi multi-kelas diperoleh nilai akurasi diatas 75%. Sehingga dapat diperoleh kesimpulan bahwa klasifikasi dengan menggunakan metode SVM memberikan nilai akurasi yang tinggi.

Diketahui bahwa data EEG adalah data yang berbasis waktu, sehingga diperlukan metode yang tepat untuk mengatasi permasalahan deret waktu tersebut. Sebelumnya telah dilakukan berbagai kombinasi pendekatan ekstraksi fitur sehingga SVM dapat diadopsi sebagai metode *plug-in* dalam mengatasi permasalahan klasifikasi deret waktu. Fungsi kernel klasik seperti Gaussian RBF (GRBF) dan *polynomial* umumnya tidak cocok digunakan untuk klasifikasi berbasis deret waktu berbasis SVM karena terdapat permasalahan pada distorsi sumbu waktu (Zhang, 2010). Sebelumnya banyak penelitian yang telah menggunakan metode jarak *Dynamic Time Warping* (DTW) dan disarankan untuk menggunakan ukuran yang elastis dalam membangun kernel yang sesuai untuk mengakomodir permasalahan deret waktu. Kemudian metode Gaussian DTW (GDTW) diusulkan untuk klasifikasi SVM berbasis deret waktu. Penggunaan kernel GDTW pertama diusulkan untuk dikombinasikan dengan SVM adalah untuk identifikasi tulisan tangan (Hamdani, 2011). Berdasarkan hasil penelitian Lei dan Sun (2007) serta penelitian oleh Gudmundsson dan Runarsson (2008) diketahui bahwa kernel GDTW tidak *positive semi-definite* (PSD) yang dapat diterima oleh SVM. Hasil eksperimen diperoleh bahwa metode SVM dengan kernel GDTW (GDTW-SVM) tidak dapat menghasilkan performansi yang lebih baik dibandingkan dengan kernel GRBF (GRBF-SVM) atau *Nearest Neighbor* dengan DTW (1NN-DTW). Oleh karena itu selanjutnya penelitian mengenai pengembangan GRBF dilakukan oleh (Zhang, 2010) dengan mengusulkan *Gaussian Elastic*

Metric Kernel (GEMK) dan menghasilkan dua contoh penerapan GEMK, yaitu dengan menggunakan kernel *Gaussian Time Warp Edit Distance* (GTWED) dan kernel *Gaussian Edit Distance with Real Penalty* (GERP). Hasil penelitian menunjukkan bahwa SVM dengan GEMK jauh lebih baik dibandingkan menggunakan SVM dengan kernel Gaussian RBF dan kernel Gaussian DTW. Diketahui bahwa TWED merupakan pengembangan dari *elastic metric* sebelumnya yaitu ERP, dimana pada TWED yang dapat mendukung pergeseran waktu lokal sementara menjadi metrik.

Berdasarkan permasalahan dan studi literatur yang telah dilakukan, maka pada penelitian ini akan dilakukan klasifikasi keadaan mental yang terdiri dari tiga kelas yaitu keadaan relaks, netral dan konsentrasi berdasarkan hasil dari data rekaman EEG. Klasifikasi akan dilakukan dengan menggunakan metode SVM dengan kernel GTWED dan *channel* yang digunakan pada penelitian ini adalah *channel* O2. Hasil penelitian ini diharapkan dapat menjadi salah satu pengembangan metode statistik untuk deteksi keadaan mental dengan menggunakan hasil rekaman sinyal EEG sehingga dapat memberikan deteksi lebih objektif, cepat dan akurat. Hasil yang diperoleh diharapkan dapat membantu dokter atau psikiater dalam analisis kondisi pasien.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka permasalahan yang akan dibahas dalam penelitian ini adalah sebagai berikut.

1. Bagaimana cara kerja *Gaussian Time Warp Edit Distance* (GTWED) sebagai fungsi kernel dalam metode klasifikasi *Multiclass Support Vector Machine* (MSVM)?
2. Bagaimana model terbaik MSVM dengan kernel GTWED untuk klasifikasi kondisi mental konsentrasi, netral dan rileks berdasarkan hasil rekaman sinyal EEG?

1.3 Batasan Masalah

Batasan penelitian yang digunakan adalah perekaman pada subjek berkondisi mental normal dengan menggunakan *channel* EEG yaitu O2. Dilakukan *epoching* (pemotongan sinyal) dengan durasi 10 detik dan *overlapping* (tumpang tindih pemotongan sinyal) selama 5 detik. Klasifikasi keadaan mental yang digunakan adalah kondisi netral, rileks, konsentrasi dengan durasi perekaman setiap kondisi mental adalah 10 menit atau total durasi perekaman setiap subjek adalah 1800 detik, namun pada penelitian ini data perekaman yang digunakan adalah 8 menit setiap kondisi mental atau total durasi perekaman setiap subjek adalah 1440 detik. Pelabelan kondisi mental pada setiap subjek adalah 10 menit pertama adalah kondisi netral, 10

meni kedua adalah kondisi rileks dan 10 menit terakhir adalah kondisi konsentrasi. Pelabelan ini didasarkan pada proses perekaman EEG yang akan dijelaskan lebih detail pada Bab 3.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini berdasarkan rumusan masalah adalah sebagai berikut:

1. Mengkaji *Gaussian Time Warp Edit Distance* (GTWED) sebagai fungsi kernel dalam klasifikasi *Multiclass Support Vector Machine* (MSVM).
2. Mendapatkan model terbaik untuk klasifikasi kondisi mental konsentrasi, netral dan rileks berdasarkan hasil rekaman sinyal EEG menggunakan MSVM dengan GTWED.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah diharapkan model terbaik yang didapatkan dapat menjadi acuan atau *baseline* sehingga dapat menjadi *input* pada pengembangan *software* perekaman EEG untuk membantu ahli medis dalam melakukan klasifikasi kondisi mental seseorang. Selain itu diharapkan penelitian ini dapat menambah wawasan terkait pengembangan metode klasifikasi dengan menggunakan metode *Multiclass Support Vector Machine* (MSVM).

(Halaman ini sengaja dikosongkan)

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Telah dilakukan beberapa penelitian terkait klasifikasi keadaan mental dengan menggunakan beberapa metode klasifikasi. Penelitian terkait klasifikasi keadaan mental ini juga menggunakan berbagai keadaan mental, berbagai fitur yang digunakan dengan berbagai *channel* yang digunakan. Berikut rangkuman mengenai penelitian mengenai klasifikasi keadaan mental yang pernah dilakukan sebelumnya.

Tabel 2.1 Penelitian Terdahulu

Pengarang	Klasifikasi Keadaan Mental	Metode Filterisasi	Fitur yang digunakan	Metode Klasifikasi	Hasil Akurasi	Channel
You, S.D (2021)	<ul style="list-style-type: none"> - Relaks - Konsentrasi 	<i>Time-to-frequency transformation module</i>	<ul style="list-style-type: none"> - Rata-rata energi semua band - Eenergi Beta - Energi beta dan alpha 	SVM	81,29%	FP1 dan FP2
				BPNN	76,07%	
Jung, Dawoon., et al (2022)	<ul style="list-style-type: none"> - Netral - Kurang Bergairah - Bergairah Tinggi - Kecemasan Sosial 	<i>Independent Component Analysis (ICA)</i>	Kategori Fitur FP, DASM, RASM, CC dan FD	XGBoost	89,50%	AF7, AF8, AF9, TP10
				<i>Support Vector Classifier (SVC)</i>	72,50%	
				<i>Logistic Regression</i>	56,47%	
Saputra, N.H., et al (2022)	<ul style="list-style-type: none"> - Stres - Normal 	<i>Finite Impulse Response (FIR)</i>	Rata-rata, Standar deviasi & <i>Zero Crossing</i>	SVM	88,90%	F3 dan F4
				Random Forest	86,10%	
				K-NN	77,80%	
				Decision Tree	77,80%	
Rahman, Ahnaf Akif., et al (2022)	<ul style="list-style-type: none"> - Keadaan Tenang - Konsentrasi - Keadaan <i>Impartial</i> 	-	900 fitur dimana 500 fitur lebih diperhatikan	SVM	95,36%	TP9, AF7, AF8 dan TP10
				Gradient Boosting (GrB)	95,24%	
				KNN	93,10%	
				XGBoost	93,10%	
				AdaB	79%	

Tabel 2.1 Penelitian Terdahulu (Pendahuluan)

Pengarang	Klasifikasi Mental State	Metode Filterisasi	Fitur yang digunakan	Metode Klasifikasi	Hasil Akurasi	Channel
Bashivan, Pouya., et al (2015)	- In-truksional (logis) - <i>Recreational</i> (emosional)	<i>Dynamic Time Warping (DWT)</i>	<i>Average power, variance power, hemispheric difference in observed power</i>	<i>Logistic Regression</i>	Metode terbaik menggunakan SVM	T9, FP1, FP2, T10
				SVM		
				<i>Deep Belief Network</i>		
				<i>Random Forest</i>		
Khanal, Bidur., et al (2018)	- Normal - Konsentrasi	<i>Infinite Impulse Response (IIR) Butterworth filter</i>	Rata-rata daya band delta, teta, alfa, beta, gamma	SVM	87,55	-
Edla., Mangalorakar., et al (2018)	- Perhatian - Emosional	-	Rata-rata, standar deviasi, maksimum dan minimum	Neural Networks	88%	FP1
				SVM	82%	
				Random Forest	75%	
Bird, Manso, Ribeiro, Ekart, & Faria (2018)	- Konsentrasi - Netral - Rileks	-	<i>Statistical Features, Maksimum, Minimum, Derivatives, Log-covariance features, Shannon entropy and log-energy entropy, frequency domain, Accumulative features as energy model</i>	<i>Naïve Bayes</i>	56,3	TP9, AF7, AF8, TP10
				<i>Bayes Net</i>	72,69	
				J48	77,05	
				<i>Random Tree</i>	75,85	
				<i>Random Forest</i>	84,17	
				<i>Multilayer Perceptron (MLP)</i>	80,82	
SVM	75,24					

Berdasarkan hasil kajian teori terdahulu dari Tabel 2.1 diketahui bahwa dengan menggunakan metode *Support Vector Machine (SVM)* dapat dilakukan klasifikasi baik secara biner maupun *multiclass* untuk mendeteksi keadaan mental dan jika dibandingkan dengan

metode klasifikasi lain diperoleh tingkat akurasi yang tinggi yaitu rata-rata diatas 80%, sehingga dapat disimpulkan bahwa metode SVM dapat dengan baik digunakan untuk klasifikasi keadaan mental berdasarkan dari hasil rekaman *Electroencephalography* (EEG).

2.2 Metode Filterisasi *Finite Impulse Response*

Filter dalam konteks elektronika diartikan sebagai rangkaian yang melewatkan suatu pita frekuensi tertentu (frekuensi yang diinginkan) dengan meredam pita frekuensi lainnya. Terdapat dua jenis filter, yaitu filter analog dan filter digital. Filter analog adalah filter yang dibuat dengan menggunakan rangkaian elektronika seperti resistor, kapasitor dan op-amp, sedangkan filter digital merupakan filter yang dibuat dalam bentuk operasi *software*.

Filter digital dapat dibagi menjadi dua yaitu *Infinite Impulse Response* (IIR) dan *Finite Impulse Response* (FIR). Perbedaan kedua filter tersebut terletak pada respon impuls filter tersebut. FIR memiliki respon impuls yang panjangnya terbatas, sedangkan IIR panjang impulsnya tidak terbatas (Mooniarsih, 2010). Diketahui bahwa filter IIR mempunyai fase tidak linier dan mempunyai potensi untuk tidak stabil, sedangkan FIR tidak memiliki pola sehingga lebih memiliki kestabilan (Nurgiyatna, 2004). Output FIR $y[k]$ dapat ditulis pada persamaan sebagai berikut:

$$y[k] = \sum_{n=0}^{M-1} h[n]x[k-n] \quad (2.1)$$

Keterangan :

M : panjang filter digital

$h[n]$: response impulse filter / koefisien filter

$x[k]$: sampel sinyal masukan

$x[k-n]$: sampel sinyal masukan yang ditahan dalam TDL (*Tap Delay Line*)

$y[k]$: *output* filter digital

(Dimurtadha, 2019)

Terdapat beberapa jenis metode dalam merancang filter digital FIR. Metode yang biasa digunakan adalah metode *windowing*. Metode *windowing* adalah metode dengan memperoleh respon frekuensi ideal filter dan menghitung respon *impuls*nya. *Window* dapat dikategorikan sebagai *fixed window* atau *adjustable window* (Islam, Islam, Rahman, & et al, 2014). Beberapa fungsi window yang tersedia memiliki karakteristik yang berbeda dalam *domain* waktu dan *domain* frekuensi (Mooniarsih, 2010). Beberapa metode *fixed window* diantaranya *Rectangular*

Window, *Hanning Window*, *Hamming Window* dan *Blackman Window* yang hanya memiliki satu parameter independen panjang *window*. Sedangkan *adjustable window* memiliki dua atau lebih parameter independen, diantaranya panjang *window* dan satu parameter tambahan yang dapat mengontrol karakteristik *window* lain (Islam, Islam, Rahman, & et al, 2014). Persamaan *window* pada koefisien filter dapat ditulis sebagai berikut :

$$h(n) = h_d(n) \times w(n) \quad (2.2)$$

Diketahui bahwa dari kelima jenis fungsi tersebut, metode *Hamming Window* adalah jenis fungsi yang lebih sering digunakan. Hasil penelitian juga menunjukkan bahwa teknik *windowing* yang terbaik adalah tipe *Hamming Window* untuk nilai orde yang sama (Caroline., 2020). *Transition Bandwidth* yang dihasilkan dengan metode *Hamming Window* tidak terlalu besar, ketepatan frekuensi *cut off* yang cukup mendekati spesifikasi serta redaman yang baik. Persamaan untuk *Hamming Window* adalah sebagai berikut (Mooniarsih, 2010).

$$w(n) = \begin{cases} 0,54 - 0,46 \cos\left(\frac{2\pi n}{M-1}\right), & 0 \leq n \leq M-1 \\ 0, & \text{lainnya} \end{cases} \quad (2.3)$$

Keterangan :

$h_d(n)$: response *impulse*

$w(n)$: fungsi *window*

M : Panjang filter

n : indeks nilai sampel waktu diskrit.

2.3 Time Epoching dan Overlapping

Sinyal EEG memiliki karakteristik yang dapat berubah-ubah dan tidak stasioner (Lionitama, Djamal, & Kasyidi, 2024). Diketahui bahwa sinyal EEG stasioner hanya pada interval yang pendek (Bird, 2018). Oleh karena itu diperlukan proses segmentasi pada pemrosesan datanya. *Epoching* dalam sinyal EEG diartikan dengan proses segmentasi atau membagi data sinyal EEG (Warren J, 1987). Pemotongan *epoch* pada data sinyal EEG umumnya adalah satu hingga sepuluh detik. Menurut hasil penelitian oleh (Warren J, 1987) pemilihan panjang *epoch* berdampak pada variansi yang dihasilkan dan kecepatan identifikasi perubahan data. *Overlap* merupakan proses dimana ketika setiap *epoch* dapat saling bertumpuk dengan *epoch* lain (Wulansari, Dajalam, & Darmanto, 2016). Ukuran *overlap* yang digunakan pada analisis EEG umumnya adalah sebesar 50% dari ukuran *epoch* (Tzimourta., 2018).

Penggunaan *overlapping* dapat meningkatkan hasil akurasi hasil filtrasi, terutama jika frekuensi yang diinginkan terletak disekitar batas antara dua *epoch* (Karimah, 2023).

2.4 Ekstraksi Fitur

Proses ekstraksi fitur adalah tahapan yang penting untuk mengetahui kemampuan generalisasi klasifikasi. Ekstraksi fitur merupakan proses untuk mengekstraksi fitur dari fitur asli baik dengan transformasi linier atau non-linier. Pada data *training* ketika menjalankan metode SVM, *hyperplane* pemisah dibentuk untuk memaksimalkan margin. Koefisien vektor dari *hyperplane* adalah *orthogonal* terhadap *hyperplane* dan arena data *training* dari berbagai kelas di proyeksikan terhadap koefisien vektor sejauh mungkin, maka SVM dapat di interpretasikan dengan menggunakan ekstraktor fitur (Tajiri, 2010). Fitur ekstraksi yang dapat digunakan adalah sebagai berikut:

1. Nilai Minimum

Nilai minimum merupakan nilai sampel terkecil dimana memiliki amplitude terkecil yang dimiliki oleh sinyal itu dari setiap *sub-band*.

2. Median

Median merupakan nilai tengah dari suatu kumpulan titik pada sinyal EEG setelah diurutkan dari yang terkecil hingga terbesar. Persamaan untuk menghitung nilai median data berjumlah ganjil adalah sebagai berikut.

$$Me(x) = x_{\left[\frac{n+1}{2}\right]} \quad (2.4)$$

Persamaan untuk menghitung median pada data berjumlah genap adalah sebagai berikut.

$$Me(x) = \frac{x_{\left[\frac{n}{2}\right]} + x_{\left[\frac{n}{2}+1\right]}}{2} \quad (2.5)$$

Keterangan :

n : jumlah data

x : data ke-

3. Nilai Maksimum

Nilai maksimum merupakan nilai sampel terbesar dimana memiliki amplitude terbesar yang dimiliki oleh sinyal itu dari setiap *sub-band*.

4. Varians

Varians merupakan nilai persebaran dari suatu data. Pada penelitian ini nilai varians dapat digunakan untuk melihat bagaimana tingkat persebaran dari titik beda potensial listrik dari hasil perekaman EEG. Persamaan untuk menghitung nilai varians EEG adalah sebagai berikut.

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})}{n-1} \quad (2.6)$$

Keterangan :

n : jumlah data pada masing-masing *sub-band*

x_i : data ke- i

\bar{x} : rata-rata

5. *Skewness*

Skewness (kemiringan) adalah suatu nilai yang digunakan untuk menggambarkan tingkat ketidak simetrisan suatu distribusi. Diketahui bahwa suatu distribusi yang tidak memiliki nilai rata-rata, median dan modus yang tidak sama, maka distribusi tersebut tidak simetris. Hal ini menyebabkan kurva distribusi tersebut miring karena hanya terkonsentrasi pada salah satu sisi saja (Hidayati, Handayani, & Ikasari, 2019). *Skewness* pada distribusi *epoch* didefinisikan sebagai perbedaan dari nilai rata-rata dan median dari *epoch* (Frederick & Lubar, 2002).

Momen non-sentral ke- r dari nilai $x_1, x_2, x_3, \dots, x_n$ dilambangkan dengan m_r . Persamaan dapat ditulis sebagai berikut (Pasaribu, Pengantar Statistik, 1975).

$$m_r' = \left(\frac{1}{n}\right) \sum_{i=1}^n x_i^r, \text{ dimana } r = 0, 1, 2, \dots, n \quad (2.7)$$

Momen sentral ke- r dari nilai $x_1, x_2, x_3, \dots, x_n$ dilambangkan dengan m_r . Persamaan dapat ditulis sebagai berikut.

$$m_r = \left(\frac{1}{n}\right) \sum_{i=1}^n (x_i - \bar{x})^r \quad (2.8)$$

Nilai koefisien *skewness* berdasarkan persamaan 2.8 selanjutnya dapat dtuliskan sebagai berikut.

$$\begin{aligned} \alpha_3 &= \frac{m_3}{\sigma^3} = \left[\frac{E(x - \mu)}{\sigma} \right]^3 \\ &= \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\sigma^3} = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{(n-1)s^3} \end{aligned} \quad (2.9)$$

dimana:

- α_3 : koefisien *skewness*
- m_3 : momen ke - 3
- x_i : nilai ke- i
- \bar{x} : nilai rata-rata sampel
- n : jumlah data pada masing-masing *sub-band*
- σ : standar deviasi populasi
- s : standar deviasi sampel

Suatu distribusi yang memiliki nilai $\alpha_3 = 0$ maka distribusi tersebut memiliki kurva simetris dimana nilai *mean* = median = modus, jika $\alpha_3 > 0$, maka kurva memiliki ekor memanjang ke kanan atau menceng positif dimana nilai modus < median < *mean*, dan apabila $\alpha_3 < 0$ maka kurva condong ke kiri dimana nilai modus > median > *mean*.

6. Kurtosis

Kurtosis adalah derajat ketinggian puncak atau keruncingan dari suatu distribusi data terhadap distribusi normalnya. Salah satu ukuran yang digunakan untuk menyatakan derajat keruncingan kurva distribusi adalah dengan menggunakan momen keempat di sekitar nilai rata-rata (Hidayati T. d., 2019), dengan menggunakan persamaan 2.30 didapatkan persamaan sebagai berikut.

$$\alpha_4 = \frac{m_4}{\sigma^4} = \left[\frac{E(x - \mu)}{\sigma} \right]^4 \tag{2.10}$$

$$= \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\sigma^4} = \frac{\sum_{i=1}^n (x_i - \bar{x})^4}{(n-1)s^4}$$

dimana:

- α_4 : koefisien kurtosis
 - m_4 : momen ke - 4
 - x_i : nilai ke- i
 - \bar{x} : nilai rata-rata sampel
 - n : jumlah data pada masing-masing *sub-band*
 - σ : deviasi standar populasi
- (Pasaribu, Pengantar Statistik, 1975)

Distribusi data yang puncaknya normal memiliki kurva yang tidak terlalu runcing dan tidak terlalu datar disebut *mesokurtik* dengan nilai $\alpha_4 = 3$ (sekitar 3), sedangkan kurva data yang runcing disebut dengan *leptokurtik* dengan $\alpha_4 > 3$ dan kurva yang datar disebut dengan *platikurtik* dengan $\alpha_4 < 3$.

7. Energi

Energi merupakan fitur yang sangat penting untuk klasifikasi data sinyal *electroencephalogram* (EGG) pada *brain computer interfaces* (BCIs). Hal ini tidak hanya rasional secara fisiologis, namun juga efektif secara empiris (Sun, 2008). Energi sinyal EEG dapat berfungsi sebagai indikator intensitas aktivitas sinyal EEG dalam periode waktu tertentu. Diketahui bahwa pada kondisi rileks fitur energi paling tinggi dimiliki oleh beta dan gamma, sedangkan pada kondisi netral, fitur energi terbesar dihasilkan oleh subband alfa dan beta dan pada kondisi konsentrasi, fitur energi paling besar dihasilkan pada subband beta dan gamma (Karimah, 2023). Energi dapat didefinisikan sebagai jumlah kuadrat dari nilai amplitudo (Stancin, 2021), persamaan untuk menghitung nilai energi sebagai berikut:

$$Energi = \sum_{i=1}^n (x_i)^2 \quad (2.11)$$

Dimana :

x_i : nilai ke- i

8. Entropi

Entropi merupakan alat yang mudah digunakan untuk mengukur keseluruhan sistem dan telah digunakan secara efektif di bidang pemrosesan sinyal. Jika spectrum frekuensi suatu sinyal dianggap sebagai suatu sistem, maka distribusinya dapat dicirikan oleh entropi (Luo, 2017). Shannon mendefinisikan bahwa entropi merupakan suatu teknik persamaan yang dapat digunakan untuk mengestimasi rata-rata jumlah bit minimum yang diperlukan untuk mengkodekan simbol string berdasarkan simbol itu sendiri. Persamaan entropi dapat ditulis sebagai berikut.

$$H_{ShanEn}(x) = - \sum_{i=0}^{N-1} (p_i(x)) \left(\log_2(p_i(x)) \right) \quad (2.11)$$

Dimana :

p_i : energi probabilitas dari x_i ; $x_i = \frac{x_i^2}{\sum_{i=1}^N x_i^2}$

2.4 Support Vector Machine (SVM)

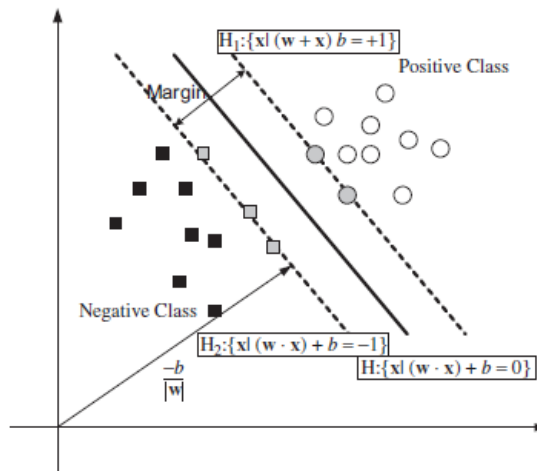
Support Vector Machine (SVM) merupakan suatu sistem yang klasifikasinya menggunakan ruang hipotesis berupa fungsi linier pada ruang fitur berdimensi tinggi yang dijalankan menggunakan algoritma berdasarkan teori optimasi dengan menerapkan *al learning bias*. SVM dikembangkan untuk menyelesaikan permasalahan klasifikasi karena SVM memiliki kemampuan untuk menjeneralisir data jika dibandingkan dengan metode sebelumnya yang sudah ada. Pendekatan dengan menggunakan SVM memiliki kelebihan dimana model yang sedang dibangun mempunyai ketergantungan eksplisit pada semua subset titik data, serta vektor bantuan yang dapat membantu interpretasi model (Ovirianti, 2022).

SVM saat ini sudah dikembangkan di berbagai kerangka kerja teori statistika dan telah berhasil diaplikasikan dari prediksi data deret waktu (Muller, 1997), pengenalan wajah (Guodong Guo, 2000), hingga ke pemrosesan data biologis untuk diagnosa medis (Widodo, 2007). Pendekatan pembelajaran klasik dirancang untuk meminimalkan kesalahan pada kumpulan data *training* dan ini disebut *Empirical Risk Minimization* (ERM). Metode tersebut mengikuti prinsip ERM dan jaringan syaraf tiruan (*neural networks*) adalah contoh paling umum dari ERM. Di sisi lain, SVM didasarkan pada *Structural Risk Minimization* (SRM) yang berakar pada teori pembelajaran statistik. Ini memberikan kemampuan generalisasi yang lebih baik dan SRM dicapai melalui minimalisasi batas atas kesalahan generalisasi (Widodo, 2007).

SVM mempunyai potensi untuk menangani ruang fitur yang sangat besar, dikarenakan *training* dari SVM dilakukan sehingga dimensi vektor yang diklasifikasikan tidak memiliki pengaruh yang berbeda terhadap kinerja SVM, dimana hal ini berbeda dengan proses yang dimiliki oleh pengklasifikasi konvensional (Widodo, 2007). Konsep SVM secara sederhana yaitu mencari *hyperplane* terbaik yang berfungsi sebagai pembagi dua kelas misalkan kelas -1 dan +1 dengan suatu *hyperplane* pada ruang yang memiliki banyak dimensi. Partisi atau pembagi tersebut diselesaikan secara non linier dengan menggunakan *hyperplane* optimum yang diperoleh dengan *quadratic program* (Bertsimas, 2007).

2.4.1 SVM Kasus Linearly Separable

Jika suatu kelas terdiri dari dua kelas yang *linearly separable*, misalkan kategori dibedakan menjadi kelas +1 dan -1 yang dapat ditulis menjadi $y_i \in \{+1, -1\}$ dimana $(y_i \in \mathbf{R}^n)$. Diketahui suatu data set D dengan $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, dimana x_i merupakan himpunan data yang memiliki hubungan dengan $y_i (x_i \in \mathbf{R}^n)$. Ilustrasi gambar dari dua kelas yang dipisahkan oleh *hyperplane* adalah sebagai berikut.



Gambar 2.1 Klasifikasi Dua Kelas oleh *Hyperplane*

Sumber : (Widodo, 2007)

Berdasarkan gambar di atas diketahui bahwa w adalah vektor bobot (*weight vector*) dengan $w = [w_1, w_2, \dots, w_n]$ dimana n adalah jumlah variabel dan b adalah sebuah skalar yang sering disebut bias. Selanjutnya variabel b dituliskan sebagai w_0 dan jika diketahui bahwa terdapat dua *training* data yaitu $\mathbf{x}^T = (x_1, x_2)$, sehingga persamaan diperoleh persamaan sebagai berikut.

$$w_0 + w_1x_1 + w_2x_2 = 0 \quad (2.12)$$

Persamaan yang membatasi *hyperplane* pada batas atas adalah sebagai berikut.

$$w_0 + w_1x_1 + w_2x_2 > 0 \quad (2.13)$$

Sedangkan persamaan yang membatasi *hyperplane* pada batas bawah adalah sebagai berikut.

$$w_0 + w_1x_1 + w_2x_2 < 0 \quad (2.14)$$

Estimasi parameter b dan w diperoleh saat proses *training*. Saat menentukan vektor pendukung, maka dibutuhkan margin yang maksimal. Margin merupakan koridor antara batas pemisah. *Hyperplane* yang menjelaskan kedua sisi dari margin dengan setiap observasi $i = 1, 2, \dots, n$ yang dapat ditulis dalam persamaan sebagai berikut.

$$w_0 + w_1x_1 + w_2x_2 \geq 1 \text{ untuk } y_i = +1 \quad (2.15)$$

$$w_0 + w_1x_1 + w_2x_2 \leq -1 \text{ untuk } y_i = -1 \quad (2.16)$$

Support vector berada diantara persamaan 2.15 dan 2.16 di setiap *training tuples*. Penentuan *support vector* adalah dengan memilih nilai dari w dan b sehingga dapat membentuk *margin* yang optimum. Margin dilambangkan dengan d_- dan d_+ , yang bertanda (-) maka

dimasuk dalam kategori -1 dan jika bertanda (+) makan masuk dalam kategori +1 (Hardle, 2014). Panjang vektor dari \mathbf{w} dapat didefinisikan sebagai berikut $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T \mathbf{w}} = \sqrt{w_1^2 + w_2^2 + \dots + w_p^2}$. Sedangkan nilai d dapat dihitung dengan mengurangi jarak tegak lurus bidang pembatas kelas negatif dan titik asal. Sehingga bidang pemisah optimum dapat ditulis dengan fungsi obyektif sebagai berikut.

$$d = \frac{(1-b) - (-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \quad (2.17)$$

Sehingga dari persamaan diatas untuk memperoleh bidang pemisah yang optimum dapat dilakukan dengan meminimumkan *norm Euclidean* dari vektor bobot \mathbf{w} . Optimasi kuadratik untuk mencari *hyperplane* yang optimal dapat menggunakan fungsi *lagrange multipliers*, fungsi *lagrange* dapat ditulis sebagai berikut.

$$L_p(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \left[\mathbf{y}_i (\mathbf{x}_i^T \mathbf{w} + b) - 1 \right] \quad (2.18)$$

Dimana:

α_i : *Lagrange multipliers*

$\|\mathbf{w}\|$: Panjang vektor bobot

\mathbf{x}_i : himpunan data

b : bias

Selanjutnya untuk meminimumkan *Lagrangian* dengan cara mencari turunan pertama dari fungsi *lagrang multipliers* terhadap \mathbf{w} dan b (Hardle, 2014).

$$\frac{\partial L_p(\mathbf{w}, b)}{\partial \mathbf{w}} = 0 \quad (2.19)$$

$$\mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \text{ sehingga diperoleh } \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \quad (2.20)$$

$$\text{Diketahui bahwa } \frac{\partial L_p(\mathbf{w}, b)}{\partial \mathbf{w}} = 0, \text{ sehingga diperoleh } \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.21)$$

Kemudian persamaan 2.19 disubstitusikan kedalam persamaan 2.18 menjadi persamaan 2.2 berikut ini.

$$\frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.22)$$

$$\begin{aligned} \sum_{j=1}^n \alpha_j \left[y_j (\mathbf{x}_j^T \mathbf{w} + b) - 1 \right] &= \frac{1}{2} \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^n \alpha_j y_j \mathbf{x}_j - \sum_{j=1}^n \alpha_j \\ &= \frac{1}{2} \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^n \alpha_j y_j \mathbf{x}_j - \sum_{j=1}^n \alpha_j \end{aligned}$$

Sehingga diperoleh *lagrangian* untuk *dual problem* adalah sebagai berikut.

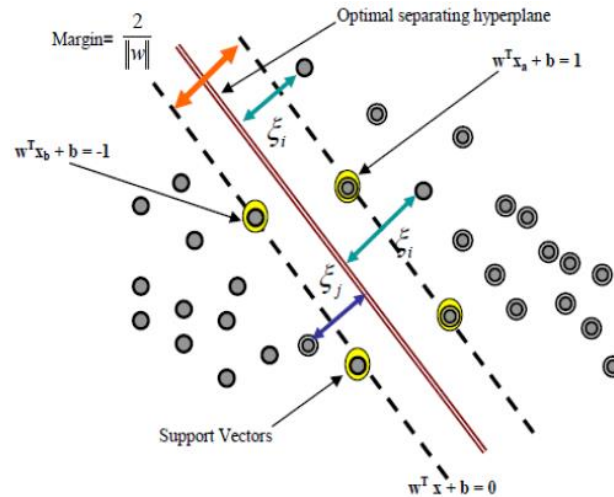
$$\max_{\alpha} L_p(\alpha) = \sum_{j=1}^n \alpha_j - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.23)$$

Selanjutnya melakukan klasifikasi objek dengan menggunakan persamaan sebagai berikut.

$$\hat{g}(x) = \text{sign}(x^T \hat{\mathbf{w}} + \hat{b}) \text{ dengan } \hat{\mathbf{w}} = \sum_{i=1}^n \hat{\alpha}_i y_i \mathbf{x}_i \quad (2.24)$$

2.4.2 SVM Kasus Linearly Non-Separable

SVM pada dasarnya merupakan metode yang digunakan untuk menyelesaikan permasalahan yang *linearly seperable*, namun pada perkembangannya SVM dapat menyelesaikan permasalahan *linearly non-separable*. *Linearly non-separable* artinya ketika dua kelas tidak dapat terpisah secara linier sempurna. Ilustrasi data yang *linearly non-separable* dapat dilihat pada Gambar 2.2 sebagai berikut.



Gambar 2.2 Ilustrasi Data SVM *Linearly Non-Separable*

Sumber : (Bekkari, 2013)

Berdasarkan Gambar 2.2 diatas, diketahui bahwa terdapat data yang seharusnya termasuk dalam kategori 1 namun letak data berada dibawah *hyperplane* kategori 2 dan begitupun sebaliknya, terdapat data yang seharusnya masuk ke kategori 2 namun letak data dibawah *hyperplane* kategori 1. Oleh karena itu pada kasus SVM *Linearly Non-Separable* terdapat penambahan variabel ξ_i atau yang dapat disebut dengan vektor *slack* yang berfungsi sebagai

bidang pemisah kelas. Vektor *slack* bernilai $0 \leq \xi_i \leq 1$ (non negatif) dan berukuran $m \times 1$ (Hardle, 2014). Sehingga persamaan dapat ditulis sebagai berikut.

$$\begin{aligned} \mathbf{x}^T \mathbf{w} + b &\geq 1 - \xi_i \text{ untuk } y_i = 1 \\ \mathbf{x}^T \mathbf{w} + b &\geq -(1 - \xi_i) \text{ untuk } y_i = -1 \\ \xi_i &\geq 0 \end{aligned} \quad (2.25)$$

Fungsi *constrain* yang terbentuk adalah sebagai berikut

$$\begin{aligned} y_i (\mathbf{x}^T \mathbf{w} + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned} \quad (2.26)$$

Jika terjadi kesalahan pemisahan kelas dimana $\xi_i \geq 0$, maka untuk memaksimalkan margin dapat menggunakan formula sebagai berikut.

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^n \xi_i \quad (2.27)$$

Fungsi *lagrange* yang digunakan untuk menyelesaikan permasalahan *primal* adalah sebagai berikut.

$$\begin{aligned} L_p(\mathbf{w}, b, \xi) &= \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i (\mathbf{x}^T \mathbf{w} + b) - 1 + \xi_i\} - \sum_{i=1}^n \mu_i \xi_i \\ \alpha_i \geq 0; \mu_i \geq 0 &\text{ adalah } \textit{lagrange multipliers} \end{aligned} \quad (2.28)$$

Masalah *primal* dirumuskan dengan meminimumkan fungsi *lagrange* dengan menghitung turunan pertama terhadap \mathbf{w}, b dan ξ . Kondisi *first-order* adalah sebagai berikut.

$$\frac{\partial L_p(\mathbf{w}, b, \xi)}{\partial \mathbf{w}} = 0; \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \quad (2.29)$$

$$\frac{\partial L_p(\mathbf{w}, b, \xi)}{\partial b} = 0; \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.30)$$

$$\frac{\partial L_p(\mathbf{w}, b, \xi)}{\partial \xi_i} = 0; c - \alpha_i - \mu_i = 0 \quad (2.31)$$

Sehingga fungsi *constraints* yang diperoleh berdasarkan persamaan diatas adalah sebagai berikut.

$$\alpha_i \geq 0, \quad (2.32)$$

$$\mu_i \geq 0, \quad (2.32)$$

$$\alpha_i \{y_i (\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i\} = 0, \quad (2.34)$$

$$\mu_i \xi_i = 0$$

Jika diketahui bahwa $\sum_{i=1}^n \alpha_i y_i b = 0$, maka persamaan *primal* akan dibawa kedalam penyelesaian permasalahan dual, sehingga diperoleh penyelesaian sebagai berikut (Hardle, 2014).

$$\begin{aligned}
L_p &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^n \alpha_j y_j \mathbf{x}_j + c \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n \mu_i \xi_i \\
&= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \underbrace{\sum_{i=1}^n \xi_i (C - \alpha_i - \mu_i)}_0
\end{aligned} \tag{2.35}$$

$$\max_{\alpha} L_p(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

Dimana $0 \leq \alpha_i \leq c$ dan $\sum_{i=1}^n \alpha_i y_i = 0$ merupakan fungsi *constrain* untuk permasalahan dual, pengamatan x_i untuk $\alpha_i > 0$ merupakan *support vector* yang berada diatas atau didalam margin ketika *soft margin* digunakan.

2.4.3 SVM Kasus *Non-Linearly Separable*

Terdapat beberapa permasalahan klasifikasi yang tidak cocok untuk dilakukan dengan klasifikasi linier, oleh karena itu perlu dilakukan transformasi dari *linearly separable* ke *non-linearly separable*. Pemetaan data *non linear* dilakukan dengan menggunakan kernel untuk dicari *hyperplane* yang optimal (Nikmah, 2023). Transformasi *linearly separation* ke *nonlinearly separation* dapat dilambangkan dengan Φ . Transformasi dari n-dimensi pada vektor \mathbf{x} ke \mathbf{m} dimensi vektor \mathbf{z} pada ruang *euclidean* R^m dapat ditulis sebagai berikut.

$$\Phi : \begin{cases} R^n \rightarrow H \\ \mathbf{x} = ([x]_1, \dots, [x]_n)^T \rightarrow \mathbf{z} = (z_1, z_2, \dots, z_n)^T = \Phi(x) \end{cases} \tag{2.36}$$

Dimana persamaan data *training* awal dan *training* setelah dilakukan transformasi sebagai berikut.

$$\begin{aligned}
T_{awal} : T &= \{(x_1, y_1), \dots, (x_n, y_n)\} \\
T_{trans} : T &= \{(z_1, y_1), \dots, (z_n, y_n)\} \\
x_i \in R^n, y_i \in Y &= \{+1, -1\}, i = 1, 2, \dots, n
\end{aligned} \tag{2.37}$$

Pada kasus nonlinier kernel dibutuhkan untuk membentuk suatu *hyperplane*. Fungsi dasar kernel adalah sebagai berikut (Scholkopf & Smola, 2002).

$$\left(\phi^T(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)\right) = K(\mathbf{x}_i, \mathbf{x}_j) \quad (2.38)$$

Matrik gram merupakan matrik *positive semidefinite*, sehingga $k(\mathbf{x}_i, \mathbf{x}_j)$ merupakan matriks kernel, dimana $\Phi(x)$ merupakan fungsi pemetaan nonlinier. Berikut adalah beberapa fungsi kernel yang sering digunakan.

- 1) Kernel Linier

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \quad (2.39)$$

- 2) Kernel Polinomial

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r\right)^d \quad (2.40)$$

- 3) *Radial Basis Function (RBF) Kernels*

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \\ &= \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \end{aligned} \quad (2.41)$$

- 4) Kernel *Sigmoid*

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r\right) \quad (2.42)$$

Keterangan:

d : Parameter *degree* (derajat) atau orde dari polinomial

γ : Parameter gamma

r : Parameter coef0

2.5 *Gaussian Elastic Metric Kernel*

Sebelum mendefinisikan GEMK, sebelumnya telah diketahui bahwa kernel GRBF merupakan salah satu fungsi kernel yang paling umum digunakan dalam pengklasifikasi SVM. Diberikan dua deret waktu x dan y dengan panjang n yang sama, kernel GRBF didefinisikan sebagai berikut.

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (2.43)$$

Dimana σ adalah simpangan baku. Kernel GRBF adalah kernel PDS yang memasukkan jarak Euclidean ke dalam bentuk fungsi Gaussian. Kernel GRBF memerlukan deret waktu yang memiliki panjang yang sama dan tidak dapat menangani masalah distorsi sumbu waktu. Jika

panjang dua deret waktu berbeda, biasanya diperlukan pengambilan sampel ulang untuk menormalkannya ke panjang yang sama sebelum diproses lebih lanjut. Jadi SVM dengan kernel GRBF (GRBF-SVM) biasanya tidak cocok untuk klasifikasi deret waktu.

Oleh karena itu, untuk menangani adanya masalah distorsi waktu, terdapat ide oleh Zhang., et al (2010) untuk memasukkan *elastic distance* kedalam model SVM *time series*. Menurut Zhang , et.al (2010) bahwa buruknya kinerja kernel GDTW mungkin disebabkan oleh DTW yang non-metrik, dan menyarankan untuk memperluas kernel GRBF menggunakan metrik elastis. Oleh karena itu, pada penelitian ini akan digunakan fungsi *Gaussian Elastic Metric Kernel* (GEMK). Secara umum persamaan GEMK dapat ditulis sebagai berikut.

$$K_{EM}(x_i, x_j) = \exp\left(-\frac{D^2(x_i, x_j)}{2\sigma^2}\right) \quad (2.44)$$

Dimana X merupakan *finite set* dari *time series* dan D adalah nilai jarak yang *elastic metric* dimana merupakan penggabungan antara *elastic distance* dan l_p norm. L_p norm didefinisikan sebagai berikut.

$$L_p(x_i, x_j) = \left(\sum_{i=1}^n \|x_i - x_j\|^p\right)^{\frac{1}{p}} \quad (2.45)$$

Ketika $p = 2$ maka disebut dengan jarak *euclidean*. L_2 norm optimal digunakan pada penghitungan *Maximum Likelihood* dan banyak digunakan untuk perhitungan perbedaan jarak (Yi & Faloutsos, 2000). Salah satu pengaplikasian *Gaussian Elastic Metric Kernel* (GEMK) yaitu *Gaussian Time Warp Edit Distance* (GTWED) Kernel (Zhang, 2010).

2.5.1 Time Warp Edit Distance

Time Wrap Edit Distance (TWED) pertama kali diusulkan oleh Marteau PF (2008), dimana TWED mengembangkan matrik dengan memasukkan *stamp* waktu dari data *time series*. Diberikan U merupakan suatu himpunan deret waktu diskrit yang terbatas pada : $U = \{A_i^p / p \in N^+\} \cup \{\Omega\}$, dimana Ω merupakan himpunan kosong dari suatu deret waktu. Diketahui bahwa A_i^p merupakan deret waktu dengan memiliki indeks diskrit yang bervariasi antara 1 dan p . Diberikan a_i^j adalah sampel ke $-i$ dari deret waktu A dengan $a_i^j \in S \times T$ dimana $S \subset R^k$ dengan $k \geq 1$ pada variabel ruang multidimensional dan $T \subset R$ melekat pada variabel

time stamp, sehingga dapat ditulis menjadi $a'_i = (a_i, t_{a_i})$ dengan kondisi bahwa $t_{a_i} > t_{a_j}$, $i > j$.

Time Warp Edit Distance dapat didefinisikan sebagai berikut (Marteau P.-F. , 2008).

$$\delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \min \begin{cases} \delta_{\lambda,\gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^q) + \Gamma(a'_p \rightarrow \Lambda) & \text{delete - A} \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^{q-1}) + \Gamma(a'_p \rightarrow b'_q) & \text{match} \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^{q-1}) + \Gamma(\Lambda \rightarrow b'_q) & \text{delete - B} \end{cases} \quad (2.46)$$

$$\Gamma(a'_p \rightarrow \Lambda) = d(a'_p, a'_{p-1}) + \lambda$$

dengan, $\Gamma(a'_p \rightarrow \Lambda) = d(a'_p, b'_q) + d(a'_p, b'_{q-1})$

$$\Gamma(\Lambda \rightarrow b'_q) = d(b'_{q-1}, b'_q) + \lambda$$

Diketahui bahwa d merupakan jarak yang berada pada R^{k+1} , dengan $d(a', b') = d_{LP}(a, b) + \gamma \cdot d_{LP}(t_a, t_b)$. Didefinisikan $\delta_{\lambda,\gamma}$ merujuk pada kedua parameter yaitu γ yang merupakan parameter untuk menggambarkan *stiffness* dari jarak elastis $\delta_{\lambda,\gamma}$ dan λ adalah elemen nilai konstan positif di ruang R yang koresponden terhadap pinalti kesenjangan, sehingga diketahui bahwa $\lambda \geq 0, \gamma > 0$, dimana $p \geq 1, q \geq 1$ dan Γ merupakan nilai dari fungsi *cost*.

Inisialisai fungsi pengulangan adalah sebagai berikut (Marteau P.-F. , 2008).

$$\begin{aligned} \delta_{\lambda,\gamma}(\mathbf{A}_1^0, \mathbf{B}_1^0) &= 0 \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^0, \mathbf{B}_1^j) &= \infty \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^i, \mathbf{B}_1^0) &= \infty \end{aligned} \quad (2.47)$$

dengan $a'_0 = b'_0 = 0$

Parameter λ dan γ dioptimumkan untuk setiap *dataset* pada data *training* dengan meminimumkan tingkat eror klasifikasi dari klasifikasi terdekat. Nilai *stiffness* (γ) yang biasa digunakan adalah $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ dan nilai λ dipilih pada renang nilai $\{0, 0.25, 0.5, 0.75, 1\}$ (Jia, Zhang, & Li, 2014). Sehingga dengan mensubstitusikan persamaan 2.47 dengan TWED matriks maka persamaan GTWED dapat ditulis sebagai berikut.

$$k_{gtwed}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \exp\left(-\frac{\delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q)}{2\sigma^2}\right) \quad (2.48)$$

2.6 Multiclass SVM

Klasifikasi *multiclass* merupakan suatu proses klasifikasi setiap titik data pada kelas yang berbeda, dengan jumlah kelas lebih dari dua. SVM pada dasarnya dibuat untuk mengatasi permasalahan klasifikasi biner, namun selanjutnya terdapat pengembangan metode SVM untuk dapat mengatasi permasalahan *multiclass*. Klasifikasi SVM secara *multiclass* dapat dilakukan dengan melakukan partisi dataset berdasarkan kelas untuk dibentuk pengklasifikasian secara biner. Terdapat beberapa pendekatan untuk pengklasifikasian *multiclass*, yaitu *One Against One* dan *One Against All* (Tantika, 2022).

2.6.1 One-Against-All (OAA)

Metode One-Against-All merupakan metode klasifikasi *multiclass* melalui pendekatan dengan melakukan pembagian terhadap permasalahan *multiclass* kedalam m sub masalah biner, dimana jika setiap pengklasifikasian memperlakukan salah satu kelas sebagai kelas positif, maka kelas lain akan diklasifikasikan dengan kelas negatif (Tantika, 2022). Apabila terdapat data sebanyak n untuk dilakukan *training* dimana x_i merupakan data input dan y_i adalah kelas dari data x_i dimana $i = 1, 2, \dots, n$, maka persamaan optimasi untuk menyelesaikan *hyperplane* ke- p adalah sebagai berikut.

$$\min_{w^p, b^p, \xi_i^p} \left\{ \frac{1}{2} (w^p)^T w^p + c \sum_{v=1}^V \xi_i^p \right\} \quad (2.49)$$

Hyperplane yang menentukan sisi margin dimana data dikategorikan kelas p adalah dengan menggunakan persamaan sebagai berikut.

$$(w^p)^T \phi(x_i) + b^p \geq 1 - \xi_i^p \text{ untuk } y_i = p \quad (2.50)$$

Sedangkan *hyperplane* data yang dikategorikan selain kelas p ditunjukkan pada persamaan sebagai berikut.

$$(w^p)^T \phi(x_i) + b^p \geq -1 - \xi_i^p \text{ untuk } y_i \neq p \quad (2.51)$$

Dimana :

u : Banyak variabel predictor

i : sampel data ke- i

e : Banyak model SVM biner

ξ_i^p : Variabel *slack* atau *soft margin hyperplane*

Terdapat e fungsi pemisah yang dapat dituliskan dengan $(w^1)^T \phi(x_i) + b^1, \dots, (w^e)^T \phi(x_i) + b^e$. Prediksi kelas dari data x merupakan nilai terbesar dari

fungsi *hyperplane* yang dapat dituliskan dengan persamaan berikut (Yang, Yu, He, & Guo, 2013).

$$y(\mathbf{x}) = \arg \max_{p=1,2,\dots,e} (\mathbf{w}^p)^T \phi(\mathbf{x}) + b^p \quad (2.52)$$

Diketahui pada sub sub bab 2.4.3 bahwa $k(\mathbf{x}_i, \mathbf{x}_j)$ merupakan matriks kernel, dimana $\Phi(x)$ merupakan fungsi pemetaan nonlinier, maka persamaan untuk fungsi klasifikasi MSVM menggunakan pendekatan OAA dengan kernel GTWED adalah dengan men substitusikan persamaan 2.48 kedalam persamaan 2.52 menjadi persamaan 2.53 sebagai berikut.

$$\begin{aligned} y(\mathbf{x}) &= \arg \max_{p=1,2,\dots,e} (\mathbf{w}^p)^T k(\mathbf{x}_i, \mathbf{x}_j) + b^p \\ &= \arg \max_{p=1,2,\dots,e} (\mathbf{w}^p)^T \exp\left(-\frac{\delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q)}{2\sigma^2}\right) + b^p \end{aligned} \quad (2.53)$$

2.6.2 One-Against-One (OAO)

Pendekatan *One-Against-One* (OAO) adalah dengan membagi permasalahan *multiclass* dengan e kelas kedalam $e(e-1)/2$ biner sub permasalahan, dan masing-masing pengklasifikasi pada OAO mendiskriminasi antar setiap pasang kelas $\{c_i, c_j\}$ (Galar, 2011). Persamaan optimasi untuk menyelesaikan *hyperplane* ke- p data *training* dari kelas ke- p dan ke- q dengan jumlah data sebanyak n adalah sebagai berikut.

$$\min_{\mathbf{w}^{pq}, b^{pq}, \xi_i^{pq}} \left\{ \frac{1}{2} (\mathbf{w}^{pq})^T \mathbf{w}^{pq} + c \sum_{i=1}^n \xi_i^{pq} \right\} \quad (2.54)$$

Hyperplane yang menentukan sisi margin dimana data dikategorikan kelas p adalah dengan menggunakan persamaan sebagai berikut.

$$(\mathbf{w}^{pq})^T \phi(x_i) + b^{pq} \geq 1 - \xi_i^{pq} \text{ untuk } y_i = p \quad (2.55)$$

Sedangkan *hyperplane* data yang dikategorikan kelas q ditunjukkan pada persamaan sebagai berikut.

$$(\mathbf{w}^p)^T \phi(x_i) + b^p \geq -1 - \xi_i^p \text{ untuk } y_i = q \quad (2.56)$$

Dimana :

u : Banyak variabel prediktor

i : sampel data ke- i

ξ_i^p : Variabel *slack* atau *soft margin hyperplane*

Prediksi kelas dari data x merupakan nilai terbesar dari fungsi *hyperplane* yang dapat dituliskan dengan persamaan berikut (Yang, Yu, He, & Guo, 2013).

$$y(x) = \arg \max_{p=1,2,\dots,e^{(e-1)/2}} \sum_{q=1,q \neq 1}^{e^{(e-1)/2}} \text{sign} \left((w^{pq})^T \phi(x) + b^{pq} \right) \quad (2.57)$$

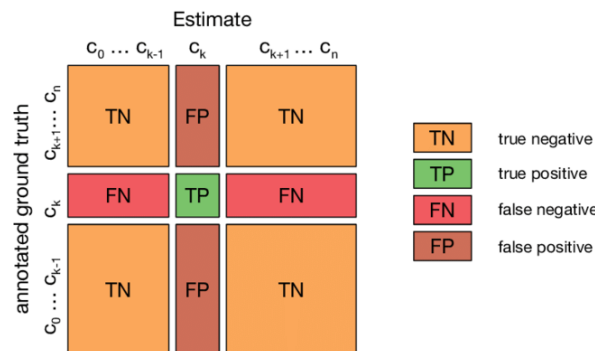
Diketahui pada sub sub bab 2.4.3 bahwa $k(x_i, x_j)$ merupakan matriks kernel, dimana $\Phi(x)$ merupakan fungsi pemetaan nonlinier, maka persamaan untuk fungsi klasifikasi MSVM menggunakan pendekatan OAO dengan kernel GTWED adalah dengan men substitusikan persamaan 2.48 kedalam persamaan 2.57 menjadi persamaan 2.58 sebagai berikut.

$$y(x) = \arg \max_{p=1,2,\dots,e^{(e-1)/2}} \sum_{q=1,q \neq 1}^{e^{(e-1)/2}} \text{sign} \left((w^{pq})^T k(x_i, x_j) + b^{pq} \right) \quad (2.58)$$

$$= \arg \max_{p=1,2,\dots,e^{(e-1)/2}} \sum_{q=1,q \neq 1}^{e^{(e-1)/2}} \text{sign} \left((w^{pq})^T \exp \left(-\frac{\delta_{\lambda,\gamma}(A_1^p, B_1^q)}{2\sigma^2} \right) + b^{pq} \right)$$

2.7 Evaluasi Performansi

Evaluasi performa adalah suatu proses yang dilakukan untuk melihat bagaimana kemampuan dari suatu metode dalam menyelesaikan permasalahan. Semakin tinggi akurasi klasifikasi maka dapat disimpulkan bahwa metode tersebut memiliki teknik klasifikasi yang semakin baik dibandingkan dengan yang lain. Cara untuk menilai perfoma suatu metode biasanya menggunakan confusion matrix. Confusion matrix menunjukkan objek yang diprediksi secara benar dan salah (Gorunescu, Data Mining: Concepts, Models, and Techniques, 2011). Tabel confusion matrix akan menunjukkan hasil prediksi yang benar dan sesuai dengan kelas aktual, sedangkan yang di luar diagonal merupakan hasil prediksi yang salah (Cahyo, 2018). Confusion matrix untuk klasifikasi n kelas ditunjukkan pada Fig. 1. (Krüger, 2016).



Gambar 2.3 Confusion Matrix
Sumber : (Krüger, 2016)

Keterangan :

True Negative : Kelas data yang diprediksi adalah negative sesuai dengan kelas sebenarnya.

True Positive : Kelas data yang diprediksi adalah positive sesuai dengan kelas sebenarnya.

False Positive : Kelas data yang diprediksi adalah positive namun kelas sebenarnya adalah negative.

False Negative : Kelas data yang diprediksi adalah negative namun kelas sebenarnya adalah positive.

Tabel *confusion matrix* pada 3 kelas klasifikasi adalah sebagai berikut (Witten & Frank, 2005).

Tabel 2.2 *Confusion Matrix* dengan 3 Kelas Klasifikasi

		Predksi		
		Kelas 1	Kelas 2	Kelas 3
Aktual	Kelas 1	A	B	C
	Kelas 2	D	E	F
	Kelas 3	G	H	I

Untuk mendapatkan ukuran klasifikasi mana yang optimal, maka dapat dilakukan perhitungan nilai akurasi, sensitivitas dan spesivitas (Han, Kamber, & Pei, 2012). Persamaan dapat dilihat sebagai berikut.

$$Accuracy = \left(\frac{A + E + I}{A + B + C + D + E + F + G + H + I} \times 100\% \right) \quad (2.59)$$

Tabel 2.3 Perhitungan Sensitivitas dan Spesifisitas

	Sensitivitas	Spesifisitas
Kelas 1	$\left(\frac{A}{A + B + C} \times 100\% \right)$	$\left(\frac{E + F + H + I}{D + E + F + G + H + I} \times 100\% \right)$
Kelas 2	$\left(\frac{E}{D + E + F} \times 100\% \right)$	$\left(\frac{A + C + G + I}{A + B + C + G + H + I} \times 100\% \right)$
Kelas 3	$\left(\frac{I}{G + H + I} \times 100\% \right)$	$\left(\frac{A + B + D + E}{A + B + C + D + E + F} \times 100\% \right)$

Area Under the Curve (AUC) merupakan salah satu metode untuk mengukur luas area di bawah grafik *Receiver Operating Curve (ROC)* (Gorunescu, Data Mining: Concepts, Models, and Techniques, 2011). Berikut persamaan untuk mengukur AUC pada kasus *multiclass*, yaitu dengan mengguakan rata-rata AUC (Hand & Till, 2001; Horng, 2009).

$$AUC = \frac{1}{2} \times (\text{sensitivity} + \text{specificity}) \quad (2.60)$$

Dimana :

$$Sensitivity = \frac{\sum_{i=0}^M Sensitivity_i (tp_i + fn_i)}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}}, \quad Specificity = \frac{\sum_{i=0}^M Specificity_i (tp_i + fn_i)}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}}$$

Nilai AUC untuk kasus multikelas dapat dihitung dengan AUC rata-rata dengan persamaan sebagai berikut (Hand, 2011).

$$AUC_{Average} = \frac{2}{k(k-1)} \sum_{i=1}^k AUC_i \quad (2.61)$$

Presisi dapat menjadi ukuran kepastian ketepatan classifier dalam memprediksi kelas. Persamaan 2.60 adalah untuk menghitung nilai presisi kelas, sedangkan persamaan 2.61 adalah untuk menghitung presisi kelas secara keseluruhan (Krüger, 2016).

$$Precision_{class} = \frac{tp_{class}}{tp_{class} + fp_{class}} \quad (2.62)$$

$$Precision = \frac{\sum_{i=0}^M Precision_i (tp_i + fn_i)}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}} \quad (2.63)$$

F1-score adalah nilai yang dapat menggambarkan kemampuan *classifier* dalam mengklasifikasikan kelas.

$$F1-score_{class} = \frac{2tp_{class}}{2tp_{class} + fn_{class} + fp_{class}} \quad (2.64)$$

$$F1-score = \frac{\sum_{i=0}^M F1-score_i (tp_i + fn_i)}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}} \quad (2.65)$$

Tingkat keakuratan klasifikasi berdasarkan nilai AUC ditampilkan oleh Tabel 2.4 sebagai berikut (Gorunescu, Data Mining: Concepts, Models, and Techniques, 2011).

Tabel 2.4 Tingkat Keakuratan berdasarkan Nilai AUC

Nilai AUC	Keterangan
0,90-1,00	<i>Excellent classification</i>
0,80-0,90	<i>Good classification</i>
0,70-0,80	<i>Fair classification</i>
0,60-0,70	<i>Poor clasification</i>
0,50-0,60	<i>Failure</i>

2.8 Mental State

Mental State adalah suatu keadaan mental yang mengacu pada keadaan pikiran yang mungkin terjadi dilihat dari berbagai sudut pandang seperti berbasis kesadaran, berbasis intensionalitas dan fungsionalisme (Baron-Cohen, 1994). Keadaan mental dapat dinilai dengan

melakukan pengamatan pada kebiasaan seseorang, mood, persepsi dalam konteks realitas dan dapat dideteksi dari aktivitas fungsional otak (Islam M. L., 2022). Terdapat beberapa klasifikasi kondisi mental diantaranya :

1. *Concentrating state* : kondisi ini merupakan keadaan yang berkaitan dengan kemampuan berpikir dimana dalam pikiran seseorang sedang bekerja dengan keras. Saat seseorang sedang memusatkan perhatiannya maka akan dominan muncul gelombang beta (You, 2021).
2. *Relaxed state* : merupakan kondisi dibidang psikofisiologis dan emosi. Kondisi ini berkaitan dengan perasaan damai, tenang dan santai dimana pikiran seseorang sedang tidak bekerja dengan keras. Saat kondisi relaks maka gelombang *alpha* akan dominan muncul (You, 2021).
3. *Neutral state* : merupakan kondisi yang berada diantara *concentrating state* dan *relaxed state*. Kondisi ini terjadi ketika seseorang sedang berada dalam keadaan seimbang secara emosional dan mental. Kondisi netral merupakan fase saat akan dimulainya perekaman EEG. Pada fase perekaman kondisi netral, subjek tidak diberikan instruksi yang spesifik (Richer, 2018).

2.9 Electroencephalography (EEG)

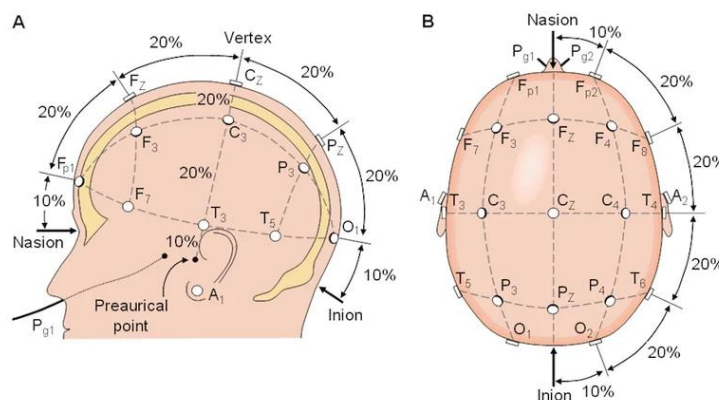
Electroencephalography (EEG) merupakan suatu metode untuk merekam aktivitas otak pada permukaan kulit kepala. EEG menjadi metode untuk melakukan pengukuran fungsi perilaku otak secara non-invasif dan pengukuran terhadap aktivitas otak yang terjadi secara spontan sebagai respons terhadap stimulus (Khakim, 2021). Pemansangan elektroda di kepala dapat melalui dua metode, yaitu metode invasif dan non invasif. Metode invasif dengan cara penanaman elektroda di bawah tengkorak atau di atas korteks. Tentunya untuk melakukan metode invasive diperlukan tindakan operasi sehingga memerlukan biaya yang besar dan memiliki resiko yang tinggi. Oleh karena itu metode ini jarang untuk digunakan. Sedangkan untuk metode non invasive menggunakan elektroda yang dipasang pada sekitar tempurung kepala di atas permukaan kulit kepala. Tentunya metode ini lebih mudah dilakukan dibandingkan dengan metode invasif dan bersifat aman (Nguyen, 2015).

Penggunaan EEG pada manusia pertama kali dilakukan oleh Hans Berger di tahun 1924, yang kemudian mulai menuai perhatian pada tahun 1930 (Borck, 2005). Diketahui bahwa otak manusia terdiri dari setidaknya 86 miliar sel/neuron (Azevedo, 2009). Aktivitas sel-sel otak menghasilkan osilasi. Osilasi adalah irama dalam aktivitas saraf yang diamati di berbagai skala temporal dan spasial (Cole, 2017), dalam hal ini biasa disebut dengan gelombang otak yang

memiliki frekuensi (jumlah gelombang per detik), amplitude (kekuatan gelombang) dan fase yang berbeda-beda. Hasil percampuran dari beberapa osilasi yang muncul bersamaan pada frekuensi yang berbeda yang terekam oleh EEG akan membentuk gelombang. Pada umumnya terdapat 4 proses yang berjalan dalam pengukuran aktivitas otak dengan EEG, diantaranya (Khakim, 2021):

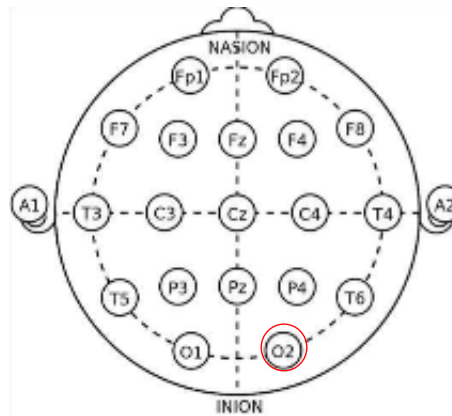
1. Transduksi dari sinyal bioelektris menjadi arus elektrik oleh elektroda.
2. Transmisi sinyal dari sensor elektroda menuju *amplifier*.
3. Amplifikasi sinyal EEG oleh *amplifier*.
4. Konversi dari sinyal analog menjadi digital.

Terdapat risiko dalam berkurangnya kualitas data pada setiap proses tersebut. Penempatan elektroda telah distandarkan melalui *International Federation of Societies for Electroencephalography and Clinical Neurophysiology* (IFSECN) yang secara umum dikenal dengan *10-20 Electrode Placement International Standard* (Tamura & Chen, 2028). Angka “10” dan “20” merujuk pada jarak antara elektroda yang berdekatan, yaitu 10% atau 20% dari total jarak (depan-belakang atau kanan-kiri) tengkorak. Jarak total didasarkan pada lokasi anatomi pada kulit kepala: nasion dan inion (arah depan-belakang) dan dua titik *preauricular* (arah kanan-kiri) seperti terlihat pada Gambar 2.4. Penggunaan penanda anatomi ini, penempatan elektroda dapat ditentukan dengan proporsi yang telah ditentukan sebelumnya: 10% digunakan dari penanda anatomi dan elektroda pertama pada arah tersebut, dan 20% digunakan di antara elektroda lainnya (TMSi, 2022). Ilustrasi penempatan elektroda dengan sistem 10-20 dapat dilihat pada gambar berikut.



Gambar 2.4 Ilustrasi Penempatan Elektroda 10-20

Sumber : (TMSi, 2022)



Gambar 2.5 EEG Sensor O2

Berdasarkan Gambar 2.4 penggunaan kode huruf menunjukkan lokasi penempatan elektroda, dimana “F” melambangkan Frontal, “Fp” melambangkan Frontolar, “C” melambangkan Central, “P” melambangkan Parietal, “O” melambangkan Occipital dan “T” melambangkan lokasi temporal dan symbol “z” merupakan penempatan garis tengah. Pada penelitian ini akan digunakan *channel O2*.

Tabel 2.5 Jenis Sinyal EEG

Gelombang	Frekuensi
Delta	< 4 Hz
Theta	4-8 Hz
Alpha	8-13 Hz
Beta	13-30 Hz
Gamma	>30 Hz

Sub band gamma menunjukkan aktivitas yang kompleks adanya proses visual, audio dan motorik. Oleh karena itu gelombang gamma terjadi dalam kondisi kesadaran penuh dan ketika seseorang mengalami aktivitas mental yang tinggi. Sehingga adanya aktivitas gelombang gamma yang tinggi menyebabkan kecemasan dan stres. Gelombang beta menunjukkan adanya kondisi fokus dan ketika seseorang sedang berpikir logis, sehingga pada kondisi gelombang beta yang cukup akan mendorong seseorang untuk produktif. Namun, jika berlebihan akan dapat menimbulkan kecemasan dan stress. Selanjutnya kemunculan gelombang *alpha* adalah ketika seseorang dalam kondisi mengantuk, bersantai atau berkhayal. Sehingga apabila seseorang sedang dalam kondisi gelombang *alpha* yang cukup maka akan membuat orang tersebut dalam keadaan rileks, namun gelombang *alpha* yang tinggi akan membuat seseorang menjadi kurang konsentrasi dan melamun. Ketika seseorang sedang dalam kondisi melamun atau tidur maka akan muncul gelombang *theta*. Gelombang *theta* terhubung dengan perasaan dan emosi paling

dasar manusia. Sehingga jika seseorang dengan kondisi gelombang *theta* yang cukup akan membawa kepada kondisi rileks, namun jika berlebihan maka akan membuat seseorang rawan untuk terkena depresi dan mudah terpengaruh oleh orang lain. Ketika seseorang sedang dalam kondisi mengantuk dan tertidur lelap, maka akan muncul gelombang *delta* (Abhang, 2016).

BAB III METODOLOGI PENELITIAN

3.1 Kajian *Gaussian Time Warp Edit Distance Kernel* (GTWED) sebagai Fungsi Kernel pada *Multiclass SVM*

Time Warp Edit Distance (TWED) merupakan pengembangan untuk menyelesaikan persoalan *time shifting* pada data *time series* yang menggunakan matrik elastis. Tahapan untuk mencapai tujuan pertama dari penelitian ini adalah sebagai berikut.

1. Menyusun algoritma TWED untuk mendapatkan *elastic distance* (δ_{TWED}).
 - (i) Diberikan dua data *time series* $A_1^p = [(a_1, t_1), \dots, (a_i, t_i), \dots, (a_p, t_p)]$ sebanyak p elemen dan $B_1^q = [(b_1, t_1), \dots, (b_j, t_j), \dots, (b_q, t_q)]$ sebanyak q elemen, dimana t_{a_i} merupakan *stamps* waktu dari A_1^p dan t_{b_j} merupakan *stamps* waktu dari B_1^q .
 - (ii) Membuat *cost matrix* berukuran $p \times q$ dengan elemen matriks (i, j) berupa nilai jarak. Inisialisasi pengulangan menggunakan persamaan 2.46 sebagai berikut

$$\delta_{\lambda, \gamma}(\mathbf{A}_1^0, \mathbf{B}_1^0) = 0$$

$$\delta_{\lambda, \gamma}(\mathbf{A}_1^0, \mathbf{B}_1^j) = \infty$$

$$\delta_{\lambda, \gamma}(\mathbf{A}_1^i, \mathbf{B}_1^0) = \infty$$

$$\text{dengan } a_0' = b_0' = 0$$

- (iii) Menghitung jarak TWED antara A dan B dengan menggunakan persamaan 2.45 sebagai berikut.

$$\delta_{\lambda, \gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \min \begin{cases} \delta_{\lambda, \gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^q) + \Gamma(a_p' \rightarrow \Lambda) & \text{delete - A} \\ \delta_{\lambda, \gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^{q-1}) + \Gamma(a_p' \rightarrow b_q') & \text{match} \\ \delta_{\lambda, \gamma}(\mathbf{A}_1^p, \mathbf{B}_1^{q-1}) + \Gamma(\Lambda \rightarrow b_q') & \text{delete - B} \end{cases}$$

$$\Gamma(a_p' \rightarrow \Lambda) = d(a_p', a_{p-1}') + \lambda$$

$$\text{dengan, } \Gamma(a_p' \rightarrow \Lambda) = d(a_p', b_q') + d(a_p', b_{q-1}')$$

$$\Gamma(\Lambda \rightarrow b_q') = d(b_{q-1}', b_q') + \lambda$$

2. Substitusikan *elastic distance* (δ_{TWED}) kedalam persamaan fungsi kernel *Gaussian Elastic Kernel* (GEMK), sehingga menjadi fungsi kernel GTWED sebagai berikut.

$$k_{GTWED}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \exp\left(-\frac{\delta_{TWED}(\mathbf{A}_1^p, \mathbf{B}_1^q)}{2\sigma^2}\right)$$

3. Subtitusikan kernel *Gaussian Time Warp Edit Distance (GTWED)* kedalam fungsi optimasi dan untuk mendapatkan prediksi kelas dari data.

Fungsi optimasi adalah sebagai berikut.

$$\min_{\mathbf{w}^{pq}, b^{pq}, \xi_i^{pq}} \left\{ \frac{1}{2} (\mathbf{w}^{pq})^T \mathbf{w}^{pq} + c \sum_{i=1}^n \xi_i^{pq} \right\}$$

Prediksi kelas dari data untuk metode *One-Against-All (OAA)* adalah sebagai berikut.

$$y(\mathbf{x}) = \arg \max_{p=1,2,\dots,e} (\mathbf{w}^p)^T \phi(\mathbf{x}) + b^p$$

Prediksi kelas dari data untuk metode *One-Against-One (OAO)* adalah sebagai berikut.

$$y(\mathbf{x}) = \arg \max_{p=1,2,\dots,e} \sum_{q=1, q \neq p}^{e(e-1)/2} \text{sign}\left((\mathbf{w}^{pq})^T \phi(\mathbf{x}) + b^{pq}\right)$$

3.2 Klasifikasi Keadaan Mental dengan Menggunakan Hasil Rekaman EEG

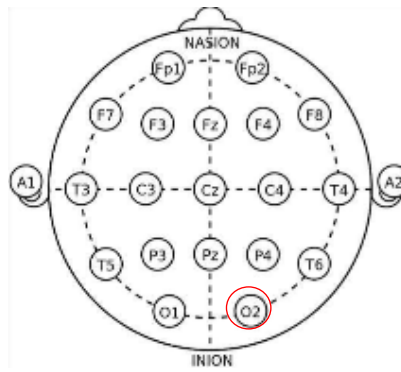
Langkah analisis untuk pengaplikasian metode MSVM dengan GTWED sebagai fungsi kernel untuk mencapai tujuan kedua dari penelitian ini adalah sebagai berikut.

3.2.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah menggunakan data primer yaitu data rekaman sinyal otak dengan *electroencephalogram (EEG)*, dimana perekaman dilakukan secara langsung di klinik swasta psikiatri yang dimiliki oleh salah satu psikiater di Surabaya. Data sukarelawan yang digunakan mengikuti kriteria inklusi adalah telah dilakukan *screening* awal kesehatan mental dan dinyatakan sehat atau tidak sedang mengalami gangguan mental tertentu. Berdasarkan kriteria tersebut terdapat sebanyak 10 sukarelawan yang memenuhi. Sehingga pada penelitian ini menggunakan data sebanyak 10 subjek.

3.2.2 Variabel Penelitian

Data pada penelitian ini menggunakan 1 *channel* yaitu O2. Penempatan elektroda berdasarkan *channel* yang digunakan ditunjukkan pada Gambar 3.1 sebagai berikut.



Gambar 3.1 Penempatan Elektroda

Data memiliki durasi perekaman setiap subjek yaitu selama 30 menit atau 1800 detik dengan kondisi subjek mata terbuka selama proses perekaman EEG. Proses perekaman EEG yang dilakukan terdiri dari 10 menit pertama adalah fase keadaan mental netral, dimana subjek diberikan gambar ikan yang bergerak dan pada fase ini subjek tidak diberikan suatu instruksi khusus. Pada 10 menit berikutnya subjek diberikan gambar jamur, dimana jika subjek semakin rileks maka gambar jamur tersebut akan semakin besar, sehingga pada fase 10 menit kedua disebut sebagai kondisi mental rileks. Selanjutnya 10 menit terakhir subjek diberikan gambar topi berwarna merah. Pada fase ini jika subjek semakin membuka mata, gambar akan terlihat semakin besar, namun sebaliknya jika subjek mengedipkan mata maka gambar akan semakin mengecil, oleh karena itu pada fase ini disebut sebagai kondisi mental konsentrasi.

Pada penelitian ini menggunakan batasan masalah yaitu dengan menggunakan data sinyal selama 8 menit pada setiap kondisi mental, sehingga 1 data akan memiliki durasi waktu selama 480 detik. Durasi pemotongan sinyal (segmentasi) yang digunakan adalah *epoch* 10 detik, *overlap* 5 detik dan *sampling rate* 500 Hz. Sehingga untuk setiap data didapatkan jumlah titik beda potensial listrik sebanyak $480 \times 500 = 240.000$.

Tabel 3.1 Struktur Data Awal Sinyal EEG Channel O2

Kondisi Mental	Subjek	Beda Potensial Listrik			
		1	2	...	240.000
Netral	1	$X_{1,1,1}$	$X_{1,1,2}$...	$X_{1,1,240.000}$
	⋮	⋮	⋮	⋮	⋮
	10	$X_{1,10,1}$	$X_{1,10,2}$...	$X_{1,10,240.000}$
Rileks	1	$X_{2,1,1}$	$X_{2,1,2}$...	$X_{2,1,240.000}$
	⋮	⋮	⋮	⋮	⋮
	10	$X_{2,10,1}$	$X_{2,10,2}$...	$X_{2,10,240.000}$
Konsentrasi	1	$X_{3,1,1}$	$X_{3,1,2}$...	$X_{3,1,240.000}$
	⋮	⋮	⋮	⋮	⋮
	10	$X_{3,10,1}$	$X_{3,10,2}$...	$X_{3,10,240.000}$

Keterangan:

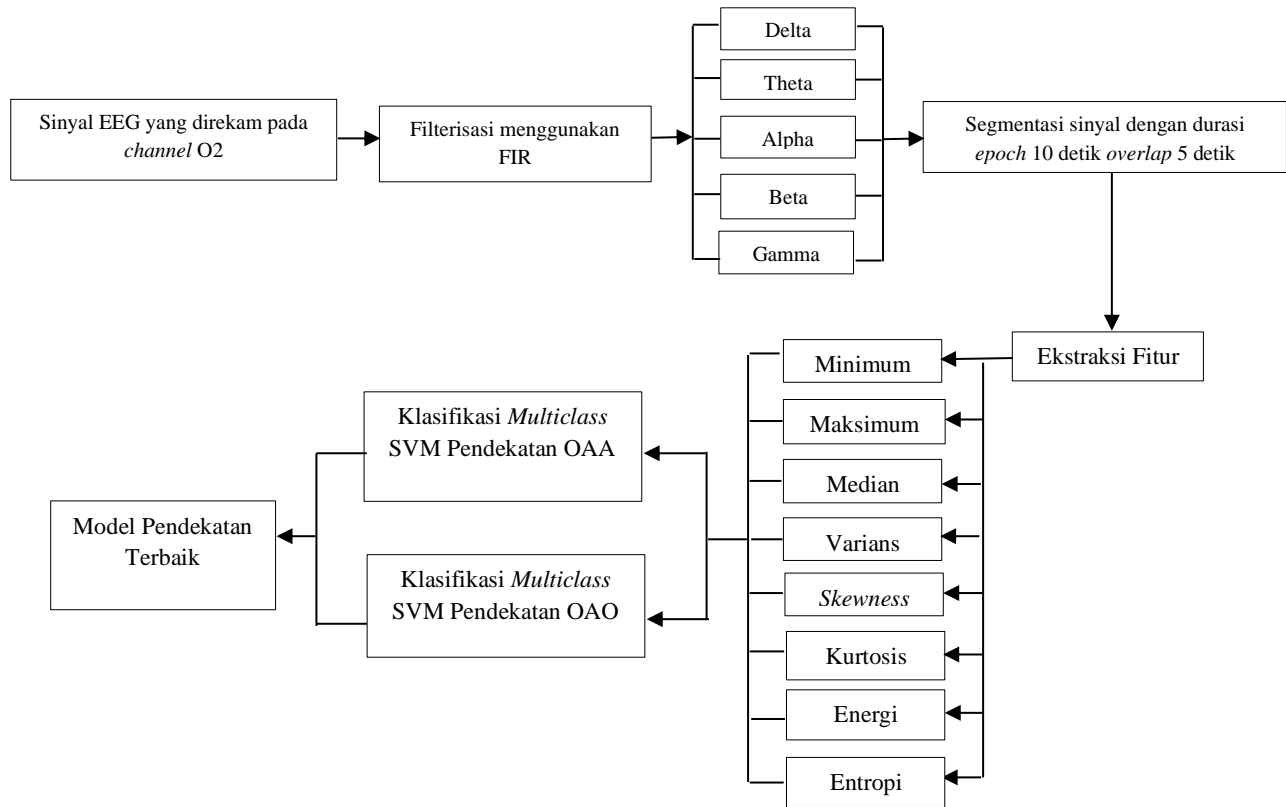
i : Keadaan mental ke- i , dimana $i = 1, 2, 3$

j : Subjek ke- j , dimana $j = 1, 2, 3, \dots, 10$

k : Beda potensial listrik ke- k , dimana $k = 1, 2, 3, \dots, 240.000$

3.2.3 Langkah Analisis

Proses pengolahan data perekaman EEG sebagai berikut.



Gambar 3.2 Proses Pengolahan Sinyal EEG

Berdasarkan Gambar 3.2 di atas merupakan diagram untuk memproses sinyal EEG selanjutnya dapat dijelaskan tahapan secara rinci sebagai berikut.

1. Filterisasi Sinyal

Filterisasi sinyal merupakan tahapan dimana sinyal akan dibagi ke dalam lima *subband* yaitu *delta*, *theta*, *alpha*, *beta* dan *gamma*, dimana *channel* yang dipilih pada penelitian ini adalah O2. Filterisasi menggunakan metode *Finite Impulse Response (FIR) Hamming Windows*. Proses filterisasi sinyal dapat dilihat pada Gambar 3.2 dan struktur data setelah proses filterisasi dapat dilihat pada Tabel 3.2 sebagai berikut.

Tabel 3.2 Struktur Data Hasil Filterisasi

Keadaan Mental (<i>i</i>)	Subjek (<i>j</i>)	Gelombang (<i>k</i>)	Beda Potensial Listrik (<i>l</i>)			
			1	2	...	47.500
Netral	1	Delta	$X_{1,1,1,1}$	$X_{1,1,2}$...	$X_{1,1,1,47.500}$
		⋮	⋮	⋮	⋮	⋮
		Gamma	$X_{1,1,5,1}$	$X_{1,1,5,2}$...	$X_{1,1,5,47.500}$
	⋮	⋮	⋮	⋮	⋮	⋮
	10	Delta	$X_{1,10,1,1}$	$X_{1,10,1,2}$...	$X_{1,10,1,47.500}$
		⋮	⋮	⋮	⋮	⋮
Gamma		$X_{1,10,5,1}$	$X_{1,10,5,2}$...	$X_{1,10,5,47.500}$	
Rileks	1	Delta	$X_{2,1,1,1}$	$X_{2,1,1,2}$...	$X_{2,1,1,47.500}$
		⋮	⋮	⋮	⋮	⋮
		Gamma	$X_{2,1,5,1}$	$X_{2,1,5,2}$...	$X_{2,1,5,47.500}$
	⋮	⋮	⋮	⋮	⋮	⋮
	10	Delta	$X_{2,10,1,1}$	$X_{2,1,10,1,2}$...	$X_{2,10,1,47.500}$
		⋮	⋮	⋮	⋮	⋮
Gamma		$X_{2,10,5,1}$	$X_{2,1,10,5,2}$...	$X_{2,10,5,47.500}$	
Konsentrasi	1	Delta	$X_{3,1,1,1}$	$X_{3,1,1,2}$...	$X_{3,1,1,47.500}$
		⋮	⋮	⋮	⋮	⋮
		Gamma	$X_{3,1,5,1}$	$X_{3,1,5,2}$...	$X_{3,1,5,47.500}$
	⋮	⋮	⋮	⋮	⋮	⋮
	10	Delta	$X_{3,10,1,1}$	$X_{3,10,1,2}$...	$X_{3,10,1,47.500}$
		⋮	⋮	⋮	⋮	⋮
Gamma		$X_{3,10,5,1}$	$X_{3,10,5,2}$...	$X_{3,10,5,47.500}$	

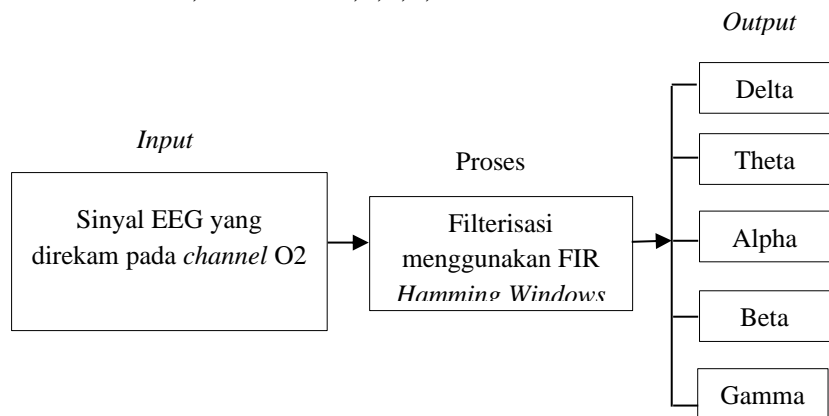
Keterangan:

i : Keadaan mental ke-*i* , dimana *i* = 1, 2, 3

j : Subjek ke-*j* , dimana *j* = 1, 2, 3, ..., 10

k : Gelombang ke-*k* , dimana *k* = 1, 2, 3, 4, 5

l : Beda potensial listrik ke-*l* , dimana *l*=1,2,3,...,47.500

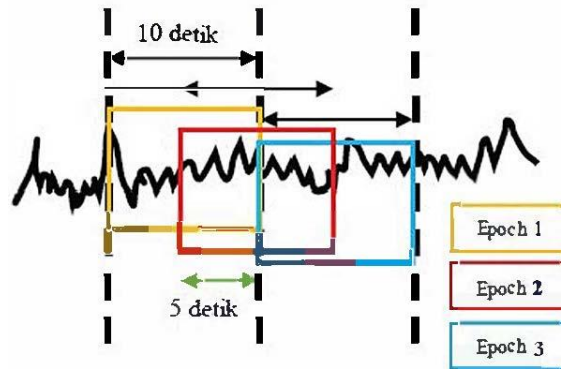


Gambar 3.3 Proses Filterisasi

2. Segmentasi Data

Segmentasi data merupakan tahap dimana dilakukan pemotongan sepuluh detik dengan *overlapping* lima detik, sehingga untuk satu data dengan durasi perekaman selama 8 menit (480

detik) akan menghasilkan 95 *epoch*. Masing-masing *epoch* akan memiliki $10 \times 500 = 5000$ beda tegangan. Ilustrasi segmentasi sinyal ditunjukkan pada Gambar 3.4 dan struktur data hasil segmentasi data ditunjukkan pada Tabel 3.3 sebagai berikut.



Gambar 3.4 Time Epoching

Sumber : (Nikmah, 2023)

Tabel 3.3 Struktur Data Hasil Segmentasi

Keadaan Mental (<i>i</i>)	Subjek (<i>j</i>)	Gelombang (<i>k</i>)	Epoch (<i>l</i>)	Beda Tegangan (<i>m</i>)			
				1	2	...	5000
Netral	1	Delta	1	$X_{1,1,1,1,1}$	$X_{1,1,1,1,2}$...	$X_{1,1,1,1,5000}$
			⋮	⋮	⋮	⋮	⋮
			119	$X_{1,1,119,1}$	$X_{1,1,119,2}$...	$X_{1,1,119,5000}$
		⋮	⋮	⋮	⋮	⋮	
		Gamma	1	$X_{1,1,5,1,1}$	$X_{1,1,5,1,2}$...	$X_{1,1,5,1,5000}$
			⋮	⋮	⋮	⋮	⋮
	119		$X_{1,1,5,119,1}$	$X_{1,1,5,119,2}$...	$X_{1,1,5,119,5000}$	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	10	Delta	1	$X_{1,10,1,1,1}$	$X_{1,10,1,1,2}$...	$X_{1,10,1,1,5000}$
			⋮	⋮	⋮	⋮	⋮
			119	$X_{1,10,1,119,1}$	$X_{1,10,1,119,2}$...	$X_{1,10,1,119,5000}$
		⋮	⋮	⋮	⋮	⋮	
Gamma		1	$X_{1,10,5,1,1}$	$X_{1,10,5,1,1}$...	$X_{1,10,5,1,5000}$	
		⋮	⋮	⋮	⋮	⋮	
	119	$X_{1,10,5,119,1}$	$X_{1,10,5,119,2}$...	$X_{1,10,5,119,5000}$		

Tabel 3.3 Struktur Data Hasil Segmentasi (Lanjutan)

Keadaan Mental (<i>i</i>)	Subjek (<i>j</i>)	Gelombang (<i>k</i>)	Epoch (<i>l</i>)	Beda Tegangan (<i>m</i>)			
				1	2	...	5000
Rileks	1	Delta	1	$X_{2,1,1,1,1}$	$X_{2,1,1,1,2}$...	$X_{2,1,1,1,5000}$
			∴	∴	∴	∴	∴
			119	$X_{2,1,119,1}$	$X_{2,1,119,2}$...	$X_{2,1,119,5000}$
		∴	∴	∴	∴	∴	
		Gamma	1	$X_{2,1,5,1,1}$	$X_{2,1,5,1,2}$...	$X_{2,1,5,1,5000}$
			∴	∴	∴	∴	∴
	119		$X_{2,1,5,119,1}$	$X_{2,1,5,119,2}$...	$X_{2,1,5,119,5000}$	
	∴	∴	∴	∴	∴	∴	∴
	10	Delta	1	$X_{2,10,1,1,1}$	$X_{2,10,1,1,2}$...	$X_{2,10,1,1,5000}$
			∴	∴	∴	∴	∴
			119	$X_{2,10,1,119,1}$	$X_{2,10,1,119,2}$...	$X_{2,10,1,119,5000}$
		∴	∴	∴	∴	∴	
Gamma		1	$X_{2,10,5,1,1}$	$X_{2,10,5,1,2}$...	$X_{2,10,5,1,5000}$	
		∴	∴	∴	∴	∴	
	119	$X_{2,10,5,119,1}$	$X_{2,10,5,119,2}$...	$X_{2,10,5,119,5000}$		
∴	∴	∴	∴	∴	∴		
Konsentrasi	1	Delta	1	$X_{3,1,1,1,1}$	$X_{3,1,1,1,2}$...	$X_{3,1,1,1,5000}$
			∴	∴	∴	∴	∴
			119	$X_{3,1,1,119,1}$	$X_{3,1,1,119,2}$...	$X_{3,1,1,119,5000}$
		∴	∴	∴	∴	∴	
		Gamma	1	$X_{3,1,5,1,1}$	$X_{3,1,5,1,2}$...	$X_{3,1,5,1,5000}$
			∴	∴	∴	∴	∴
	119		$X_{3,1,5,119,1}$	$X_{3,1,5,119,2}$...	$X_{3,1,5,119,5000}$	
	∴	∴	∴	∴	∴	∴	
	10	Delta	1	$X_{3,10,1,1,1}$	$X_{3,10,1,1,2}$...	$X_{3,10,1,1,5000}$
			∴	∴	∴	∴	∴
			119	$X_{3,10,1,119,1}$	$X_{3,10,1,119,2}$...	$X_{3,10,1,119,5000}$
		∴	∴	∴	∴	∴	
Gamma		1	$X_{3,10,5,1,1}$	$X_{3,10,5,1,2}$...	$X_{3,10,5,1,5000}$	
		∴	∴	∴	∴	∴	
	119	$X_{3,10,5,119,1}$	$X_{3,10,5,119,2}$...	$X_{3,10,5,119,5000}$		
∴	∴	∴	∴	∴	∴		

Keterangan:

i : Keadaan mental ke-*i*, dimana *i* = 1, 2, 3

j : Subjek ke-*j*, dimana *j* = 1, 2, 3, ..., 10

k : Gelombang ke-*k*, dimana *k* = 1, 2, 3, 4, 5

l : Epoch ke-*l*, dimana *l* = 1, 2, 3, ..., 199

m : Beda potensial listrik ke-*m*, dimana *m* = 1, 2, 3, ..., 5.000

3. Ekstraksi Fitur

Setelah dilakukan segmentasi data, selanjutnya dilakukan ekstraksi fitur. Fitur yang digunakan diantaranya median, minimum, maksimum, varians, *skewness*, kurtosis, energi, dan entropi. Perhitungan ekstraksi fitur dilakukan sebanyak beda tegangan yang berada pada setiap

epoch masing-masing gelombang setiap individu. Sehingga akan dihasilkan sebanyak 40 fitur ekstraksi berdasarkan perhitungan $8 \text{ fitur} \times 5 \text{ gelombang} \times 1 \text{ channel} = 40 \text{ fitur}$. Diketahui bahwa terdapat 10 subjek orang normal dengan lama perekaman masing-masing kondisi mental adalah 8 menit, sehingga jumlah *epoch* setiap kondisi mental pada satu orang subjek adalah 95 *epoch*. Total *epoch* adalah $10 \text{ subjek} \times 95 \text{ epoch} \times 3 \text{ kondisi mental} = 2.850 \text{ data}$. Struktur data pada tahapan klasifikasi adalah sebagai berikut.

Tabel 3.4 Struktur Data Ekstraksi Fitur

Konsisi Mental ke- (<i>i</i>)	Subjek ke- (<i>j</i>)	<i>Epoch</i> ke- (<i>k</i>)	Fitur Ekstraksi (<i>l</i>)			
			X ₁	X ₂	...	X ₄₀
Y ₁	1	1	X _{1,1,1,1}	X _{1,1,1,2}	...	X _{1,1,1,40}
⋮		⋮	⋮	⋮	...	⋮
Y ₁		95	X _{1,1,95,1}	X _{1,1,95,2}	...	X _{1,1,95,40}
⋮	⋮	⋮	⋮	⋮	...	⋮
Y ₁	10	571	X _{1,10,571,1}	X _{1,10,571,2}	...	X _{1,10,571,40}
⋮		⋮	⋮	⋮	...	⋮
Y ₁		950	X _{1,10,950,1}	X _{1,10,950,2}	...	X _{1,10,950,40}
Y ₂	1	951	X _{2,1,951,1}	X _{2,1,951,2}	...	X _{2,1,951,40}
⋮		⋮	⋮	⋮	...	⋮
Y ₂		1046	X _{2,1,1046,1}	X _{2,1,1046,2}	...	X _{2,1,1046,40}
⋮	⋮	⋮	⋮	⋮	...	⋮
Y ₂	10	1805	X _{2,10,1805,1}	X _{2,10,1805,2}	...	X _{2,10,1805,40}
⋮		⋮	⋮	⋮	...	⋮
Y ₂		1900	X _{2,10,1900,1}	X _{2,10,1900,2}	...	X _{2,10,1900,40}
Y ₃	1	1996	X _{3,1,1996,1}	X _{3,1,1996,2}	...	X _{3,1,1996,40}
⋮		⋮	⋮	⋮	...	⋮
Y ₃		2091	X _{3,1,2091,1}	X _{3,1,2091,2}	...	X _{3,1,2091,40}
⋮	⋮	⋮	⋮	⋮	...	⋮
Y ₃	10	2755	X _{3,10,2755,1}	X _{3,10,2755,2}	...	X _{3,10,2755,40}
⋮		⋮	⋮	⋮	...	⋮
Y ₃		2850	X _{3,10,2850,1}	X _{3,10,2850,2}	...	X _{3,10,2850,40}

Keterangan:

i : Keadaan mental ke-*i*, dimana $i = 1, 2, 3$

Dimana, Y₁ : Keadaan mental netral

Y₂ : Keadaan mental rileks

Y₃ : Keadaan mental konsentrasi

j : Subjek ke-*j*, dimana $j = 1, 2, 3, \dots, 10$

k : *Epoch* ke-*k*, dimana $k = 1, 2, \dots, 2850$

l : *Epoch* ke-*l*, dimana $l = 1, 2, 3, \dots, 40$

4. Pembagian Data *Training* dan *Testing*

Pembagian data *training* dan *testing* adalah dengan menggunakan metode *holdout* dengan perbandingan 70% untuk data *testing* dan 30% untuk data *training*. Sehingga berdasarkan Tabel 3.4, pembagian data *training* dan *testing* adalah sebagai berikut.

Tabel 3.5 Pembagian Data Training dan Testing

Kondisi Mental	Data <i>Training</i>	Data <i>Testing</i>
Netral	Data ke- 1 hingga 665	Data ke – 666 hingga 950
Rileks	Data ke- 951 hingga 1615	Data ke – 1616 hingga 1900
Konsentrasi	Data ke – 1901 hingga 2565	Data ke – 2566 hingga 2850

Merujuk langkah analisis pertama pada sub bab 3.1 diketahui terdapat deret waktu $A_1^p = [(a_1, t_1), \dots, (a_i, t_i), \dots, (a_p, t_p)]$ dan $B_1^q = [(b_1, t_1), \dots, (b_j, t_j), \dots, (b_q, t_q)]$, bahwa $a_i = (a_i, t_{a_i})$ merupakan sampel ke- i dari deret waktu A , p merupakan indeks untuk menerangkan elemen terakhir pada deret waktu A sedangkan q merupakan indeks untuk menerangkan elemen terakhir pada deret waktu B . Sehingga definisi deret *time series* untuk melakukan prediksi model SVM dari data *training* pada penelitian ini berdasarkan Tabel 3.4 dan Tabel 3.5 adalah jika ${}_1A_1^{N=665} = [(X_{i,j,k,l}, t_1), (X_{i,j,k,l}, t_2), \dots, (X_{i,j,k,l}, t_{665})]$ merupakan data *training* pada kelas 1, maka untuk pasangan deret waktu B merupakan deret waktu pada data *testing* kelas 1, sehingga ${}_2B_1^{N=666} = [(X_{i,j,k,l}, t_1), (X_{i,j,k,l}, t_2), \dots, (X_{i,j,k,l}, t_{665})]$.

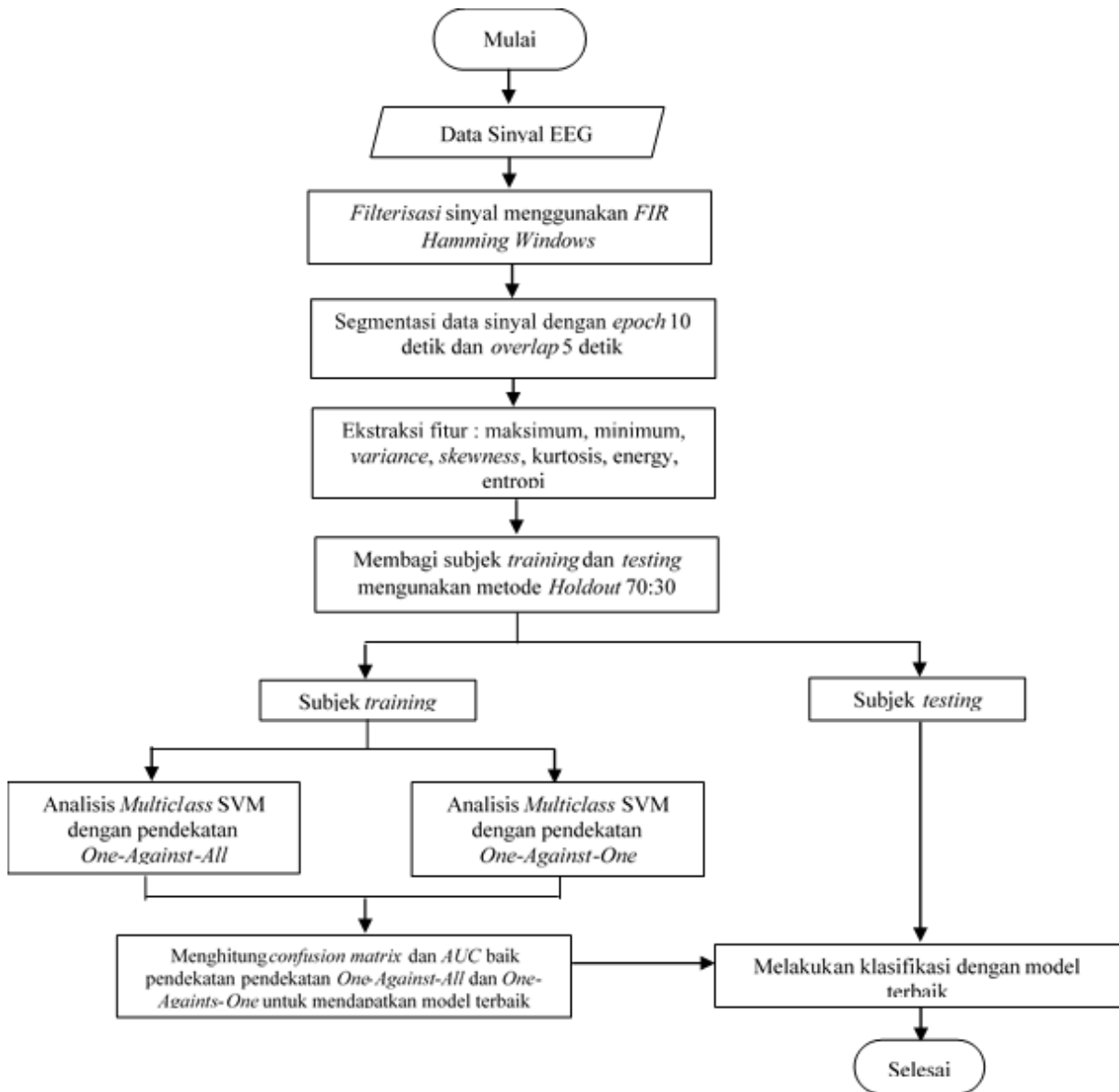
5. Klasifikasi *Multiclass* SVM

Klasifikasi *Multiclass* SVM dilakukan dengan pendekatan *One-Against-All* (OAA) dan *One-Against-One* (OAO).

6. Pemilihan Model Terbaik

Setelah melakukan klasifikasi dari pendekatan *One-Against-All* (OAA) dan *One-Against-One* (OAO), maka selanjutnya menghitung *confusion matrix* dan nilai *Area Under Curve* (AUC) untuk mengetahui model terbaik dengan menggunakan pendekatan OAA atau OAO.

Langkah-langkah analisis dapat digambarkan pada diagram alir sebagai berikut.



Gambar 3.5 Diagram Alir

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil analisis sesuai dengan rumusan masalah dan tujuan penelitian. Pada sub-bab 4.1 dilakukan pembahasan mengenai kajian *Gaussian Time Warp Edit Distance Kernel* (GTWED) sebagai fungsi kernel pada *Multiclass Support Vector Machine* (MSVM). Selanjutnya pada sub-bab 4.2 akan membahas model terbaik untuk klasifikasi kondisi mental konsentrasi, netral dan rileks berdasarkan hasil rekaman sinyal EEG menggunakan MSVM dengan kernel GTWED.

4.1 Kajian *Gaussian Time Warp Edit Distance* (GTWED) sebagai Fungsi Kernel pada *Multiclass SVM*

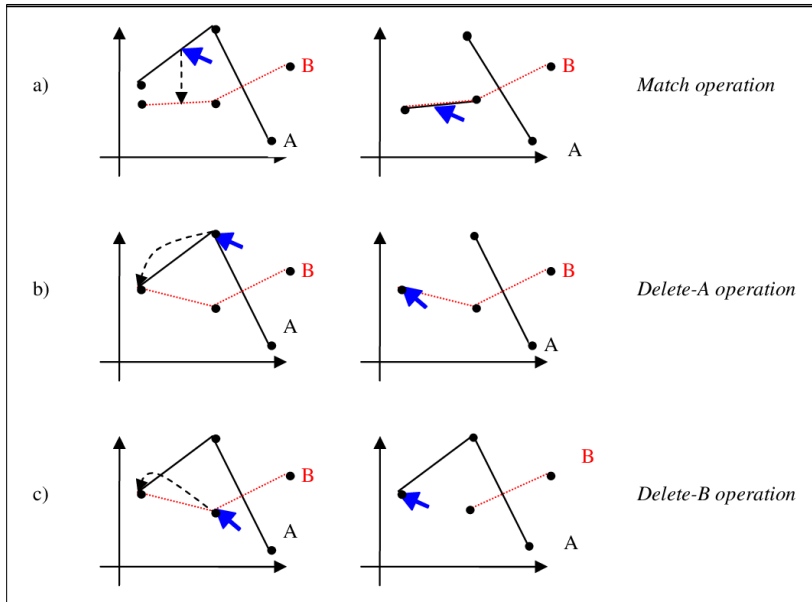
1. Menyusun Algoritma *Time Warp Edit Distance* (TWED) untuk mendapatkan *elastic distance* (δ_{TWED}).
- (i) Diberikan dua data *time series* $A_1^p = [(a_1, t_1), \dots, (a_i, t_i), \dots, (a_p, t_p)]$ sebanyak p elemen dan $B_1^q = [(b_1, t_1), \dots, (b_j, t_j), \dots, (b_q, t_q)]$ sebanyak q elemen, dimana t_{a_i} merupakan *stamps* waktu dari A_1^p dan t_{b_j} merupakan *stamps* waktu dari B_1^q .
- (ii) Membuat *cost matrix* berukuran $p \times q$ dengan elemen matriks (i, j) berupa nilai jarak. Inisialisasi pengulangan menggunakan persamaan 2.46 sebagai berikut.

$$\begin{aligned} \delta_{\lambda, \gamma}(\mathbf{A}_1^0, \mathbf{B}_1^0) &= 0 \\ \delta_{\lambda, \gamma}(\mathbf{A}_1^0, \mathbf{B}_1^j) &= \infty \\ \delta_{\lambda, \gamma}(\mathbf{A}_1^i, \mathbf{B}_1^0) &= \infty \end{aligned} \tag{4.1}$$

dengan $a_0 = b_0 = 0$

- (iii) Menghitung jarak TWED antara A dan B dengan operasi *delete – A*, *delete – B*, dan *match*.

Untuk menghitung TWED antar A dan B dengan operasi *delete – A*, *delete – B*, dan *match* terlebih dahulu digambarkan operasi edit secara grafik sebagai berikut.



Gambar 4.1 Ilustrasi Operasi *Edit Distance*

Sumber : (Marteau P. F., 2008)

a. *delete – A*

Operasi *delete – A* (hapus di dalam deret waktu pertama) (Gambar 4.1.a) terdiri dari mengklik titik yang mewakili sampel di A yang akan dihapus (a'_i) dan menyeret lalu menjatuhkan titik ini ke titik sampel sebelumnya (a'_{i-1}). Dimana dengan adanya pergeseran titik dari panjang vektor ($a'_i - a'_{i-1}$) mengakibatkan adanya biaya (*cost*) pergeseran terkait operasi edit ini dengan menambahkan constant penalti $\lambda \geq 0$.

b. *delete – B*

Operasi *delete – B* (hapus di dalam deret waktu kedua) (Gambar 4.1.b) terdiri dari mengklik titik yang mewakili sampel di B yang akan dihapus (b'_i) dan menyeret lalu menjatuhkan titik ini ke titik sampel sebelumnya (b'_{i-1}). Dimana dengan adanya pergeseran titik dari panjang vektor ($b'_i - b'_{i-1}$) mengakibatkan adanya biaya (*cost*) pergeseran terkait operasi edit ini dengan menambahkan constant penalti $\lambda \geq 0$.

c. *match*

Operasi *match* (Gambar 4.1.a) terdiri dari mengklik segmen yang akan dicocokkan pada deret waktu pertama ($a'_{i-1}a'_i$) dan kemudian menyeret dan melepaskan segmen ini ke posisi grafik yang sesuai dengan segmen yang cocok ($b'_{j-1} - b'_j$) pada deret waktu kedua. Oleh karena itu biaya (*cost*) pergeseran terkait dengan operasi *match*

ini adalah sebanding dengan jumlah panjang dari kedua vektor $(b'_j - a'_i)$ dan $(b'_{j-1} - a'_{i-1})$.

Sehingga operasi *delete - A*, *match*, dan *delete - B* dapat ditulis sebagai berikut.

$$\delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \min \begin{cases} \delta_{\lambda,\gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^q) + \Gamma(a'_p \rightarrow \Lambda) & \text{delete - A} \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^{q-1}) + \Gamma(a'_p \rightarrow b'_q) & \text{match} \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^{q-1}) + \Gamma(\Lambda \rightarrow b'_q) & \text{delete - B} \end{cases} \quad (4.2)$$

$$\Gamma(a'_p \rightarrow \Lambda) = d(a'_p, a'_{p-1}) + \lambda$$

dengan, $\Gamma(a'_p \rightarrow \Lambda) = d(a'_p, b'_q) + d(a'_p, b'_{q-1})$

$$\Gamma(\Lambda \rightarrow b'_q) = d(b'_{q-1}, b'_q) + \lambda$$

(iv) Memberikan 'stiffness' pada $\delta_{\lambda,\gamma}$

Penalti atau biaya yang terkait dengan setiap operasi pengeditan harus sebanding dengan perpindahannya. Jika memisahkan perpindahan spasial di S dari perpindahan temporal di T , maka harus mempertimbangkan penalti spasial oleh jarak yang diukur dalam S dan penalty temporal yang sebanding dengan jarak yang diukur di T . Sehingga diparameterkan sebuah jarak di antara *Euclidian Distance* yaitu 'infinite stiffness' dan DTW yang di karakteristikkann dengan 'null stiffness' yang dapat diformulasikan sebagai berikut.

$$d(a', b') = d_{LP}(a, b) + \gamma \cdot d_{LP}(t_a, t_b) \quad (4.3)$$

dimana, $\delta_{\lambda,\gamma}$ Merupakan jarak dan $\gamma > 0$ merupakan karakteristik 'stiffness' dari $\delta_{\lambda,\gamma}$.

Sehingga jika persamaan 4.3 disubtitusikan kedalam persamaan 4.2, maka persamaan final untuk menghitung $\delta_{\lambda,\gamma}$ adalah sebagai berikut.

$$\delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \min \begin{cases} \delta_{\lambda,\gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^q) + d_{LP}(a_p, a_{p-1}) + \gamma \cdot d_{LP}(t_{a_p}, t_{a_{p-1}}) + \lambda & \text{delete - A} \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^{p-1}, \mathbf{B}_1^{q-1}) + d_{LP}(a_p, b_q) + \gamma \cdot d_{LP}(t_{a_p}, t_{b_q}) + d_{LP}(a_{p-1}, b_{q-1}) & \text{match} \\ + \gamma \cdot d_{LP}(t_{a_p}, t_{b_{q-1}}) \\ \delta_{\lambda,\gamma}(\mathbf{A}_1^p, \mathbf{B}_1^{q-1}) + d_{LP}(b_q, b_{q-1}) + \gamma \cdot d_{LP}(t_{b_q}, t_{b_{q-1}}) + \lambda & \text{delete - B} \end{cases} \quad (4.2)$$

2. Subtitusikan *elastic distance* (δ_{twed}) kedalam persamaan fungsi kernel *Gaussian Elastic etric Kernel* (GEMK), sehingga menjadi fungsi kernel GTWED sebagai berikut.

$$k_{gtwed}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \exp\left(-\frac{\delta_{twed}(\mathbf{A}_1^p, \mathbf{B}_1^q)}{2\sigma^2}\right)$$

3. Subtitusikan kernel *Gaussian Time Warp Edit Distance (GTWED)* kedalam fungsi optimasi dan untuk mendapatkan prediksi kelas dari data.

Fungsi optimasi adalah sebagai berikut.

$$\min_{w^{pq}, b^{pq}, \xi_i^{pq}} \left\{ \frac{1}{2} (\mathbf{w}^{pq})^T \mathbf{w}^{pq} + c \sum_{i=1}^n \xi_i^{pq} \right\}$$

Prediksi kelas dari data untuk metode *One-Against-All (OAA)* adalah sebagai berikut.

$$y(\mathbf{x}) = \arg \max_{p=1,2,\dots,e} (\mathbf{w}^p)^T \phi(\mathbf{x}) + b^p$$

Prediksi kelas dari data untuk metode *One-Against-One (OAO)* adalah sebagai berikut.

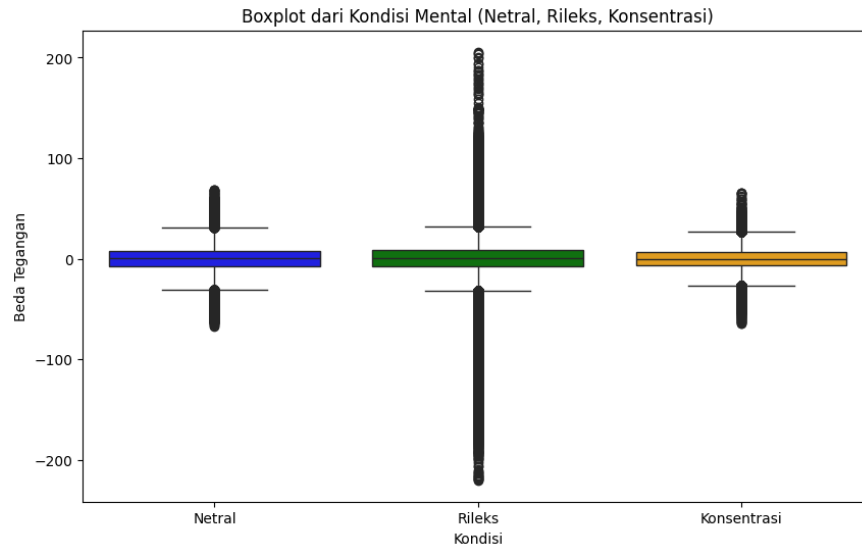
$$y(x) = \arg \max_{p=1,2,\dots,e} \sum_{q=1, q \neq p}^{e-1} \text{sign}\left((\mathbf{w}^{pq})^T \phi(x) + b^{pq}\right)$$

4.2 Klasifikasi Kondisi Mental dengan menggunakan kernel *Gaussian Time Warp Edit Distance (GTWED)*

Sebelum melakukan klasifikasi terlebih dahulu dilakukan proses filterisasi, segmentasi dan ekstraksi fitur. Pada sub-sub-bab berikut akan dijelaskan bagaimana tahapan hingga memperoleh metode klasifikasi terbaik untuk klasifikasi kondisi mental dengan menggunakan kernel *Gaussian Time Warp Edit Distance (GTWED)*

4.2.1 Karakteristik Data Awal

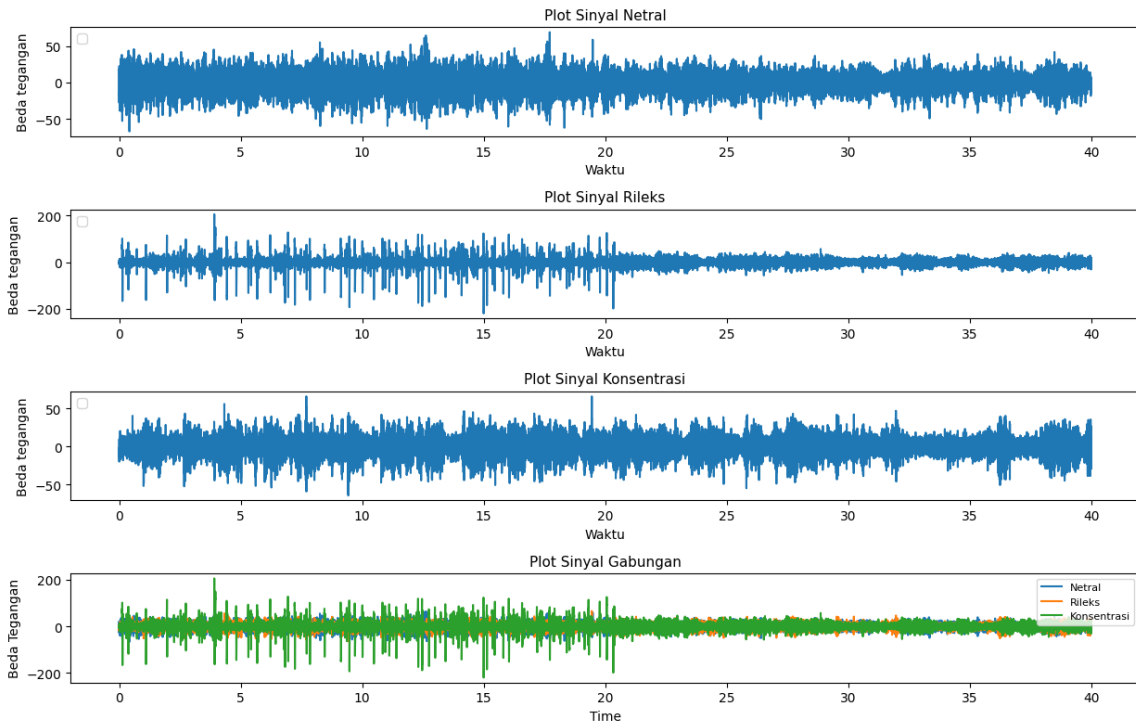
Data kondisi mental memiliki tiga jenis yaitu saat orang dalam keadaan netral, konsentrasi dan rileks. Pada penelitian ini menggunakan channel perekaman EEG yaitu O2, dimana dalam satu data memiliki durasi perekaman setiap subjek yaitu selama 30 menit atau 1800 detik yang terdiri dari 10 menit pertama pada kondisi mental netral, 10 menit berikutnya pada kondisi mental konsentrasi dan 10 menit terakhir pada kondisi mental rileks, sehingga 1 data akan memiliki durasi waktu selama 600 detik. Namun pada penelitian kali ini peneliti membatasi durasi perekaman yang diambil pada masing-masing kondisi mental adalah selama 8 menit. Berikut merupakan salah satu contoh boxplot pada salah satu orang yang menggambarkan data sinyal kondisi mental sebelum melalui tahapan prosesing.



Gambar 4.2 *Boxplot* Beda Tegangan Sebelum *Preprocessing* Data berdasarkan Kondisi Mental

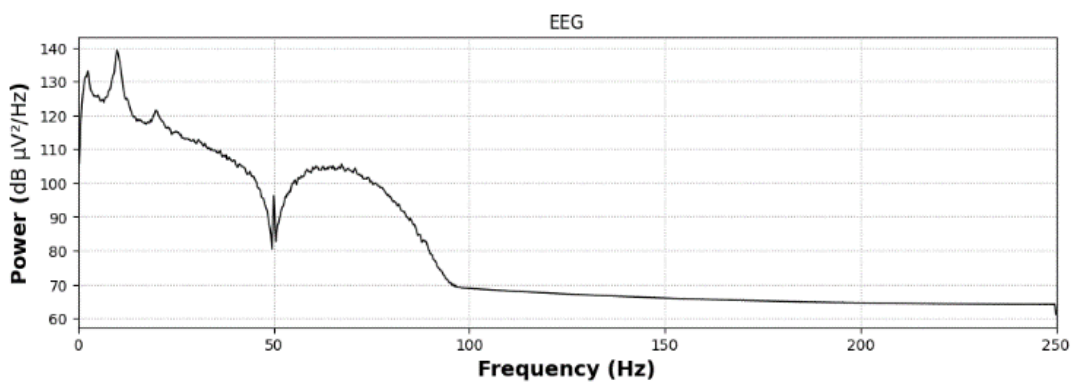
Secara umum berdasarkan Gambar 4.2 menunjukkan bahwa *boxplot* data sinyal sebelum melalui preprocessing data dengan channel otak yang dipakai adalah O2, diketahui bahwa baik pada kondisi mental netral, konsentrasi maupun rileks memiliki banyak data *outlier*. Sehingga perlu dilakukan beberapa tahapan *preprocessing* data untuk mempermudah dalam pengolahan data hingga proses klasifikasi. Berikut merupakan salah satu contoh *time series plot channel O2* pada orang pertama.

Berdasarkan Gambar 4.3 dapat dilihat bahwa sinyal pada kondisi mental netral, konsentrasi dan rileks memiliki rentang beda tegangan yang berbeda. Diketahui bahwa sinyal EEG ketika dalam kondisi rileks memiliki rentang beda tegangan yang paling kecil dan sinyal EEG ketika dalam kondisi konsentrasi memiliki rentang beda tegangan yang paling tinggi, dimana hal ini menandakan bahwa sedang terdapat aktivitas yang intens pada otak orang tersebut. Diketahui pula bahwa ketika dalam kondisi netral rentang beda tegangan tidak berbeda signifikan jika dibandingkan ketika dalam kondisi rileks, namun pada keadaan mental normal memiliki rentang beda tegangan yang lebih tinggi dibandingkan saat dalam keadaan mental rileks. Hal ini menunjukkan bahwa aktivitas gelombang otak berada di antara kondisi yang relative stabil, yaitu tidak terlalu banyak aktivitas namun belum pada kondisi rileks.

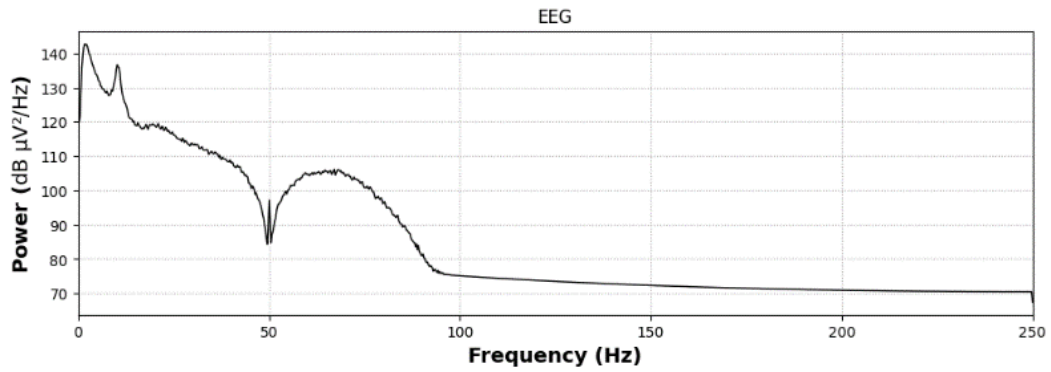


Gambar 4.3 Time Series Plot Sinyal Awal

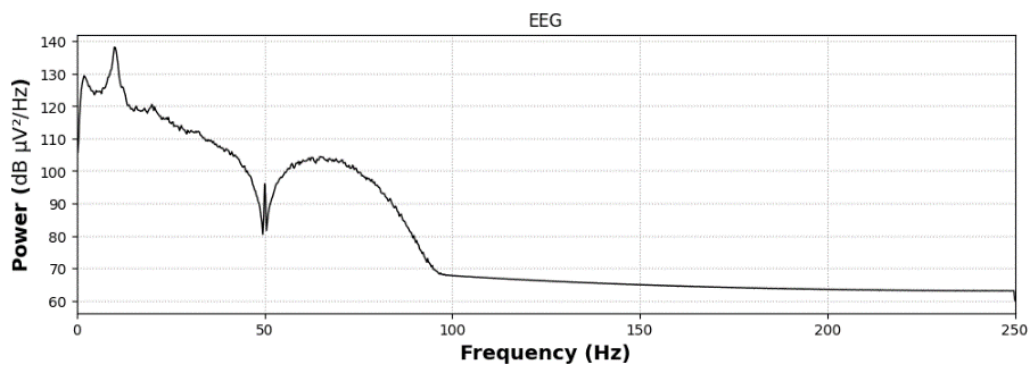
Selanjutnya karakteristik data dapat dilihat melalui plot *Power Spectral Density* (PSD). Plot PSD merupakan suatu grafik yang dapat menggambarkan distribusi dari energy frekuensi pada sinyal gelombang otak. Sehingga dengan plot PSD dapat terlihat banyaknya energi yang ada pada setiap frekuensi pada sinyal gelombang otak. Plot PSD pada masing-masing kondisi mental dapat dilihat pada Gambar 4.3 sebagai berikut.



(a)



(b)



(c)

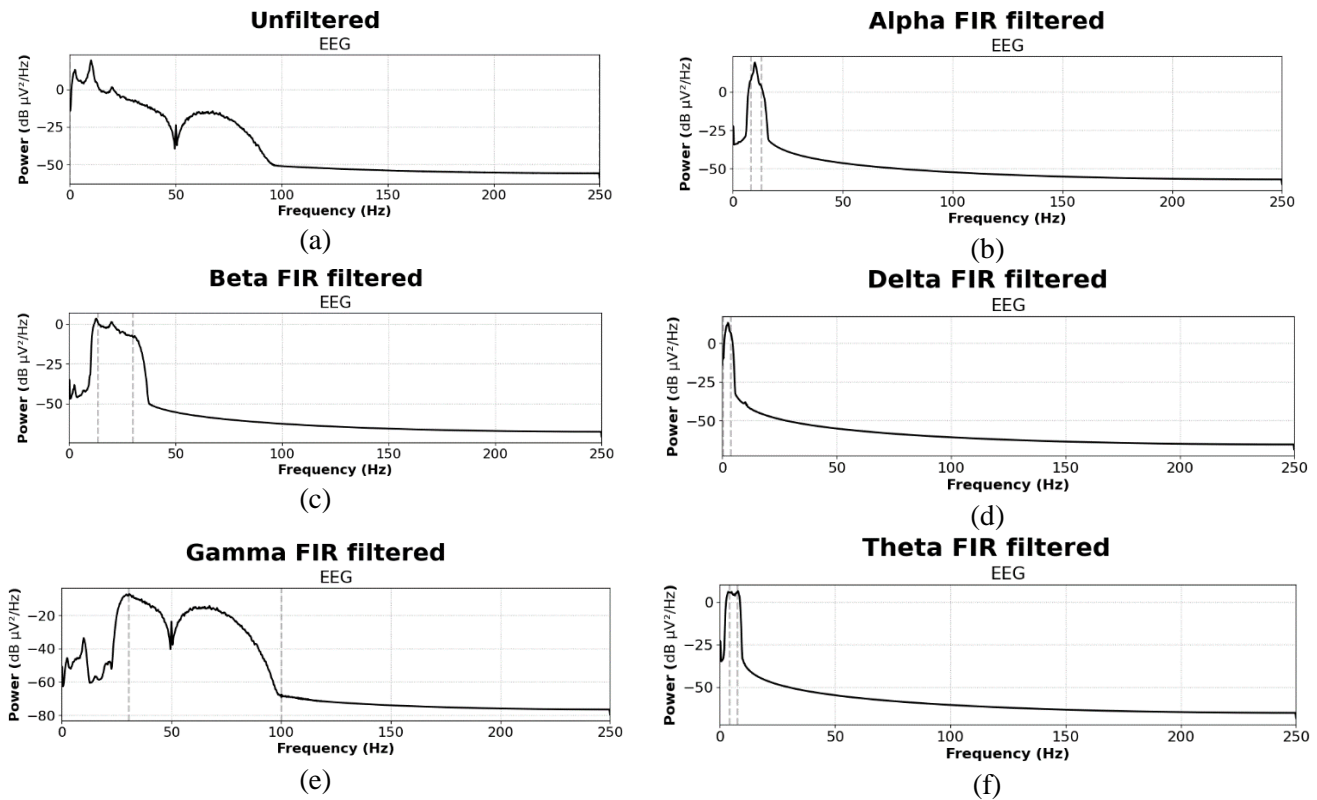
Gambar 4.4 Plot PSD *Channel O2* Sebelum Filterisasi (a) Kondisi Netral, (b) Kondisi Konsentrasi, (c) Kondisi Rileks

Berdasarkan Gambar 4.3 diatas menunjukkan plot PSD pada masing-masing kondisi mental sebelum dilakukan proses filterisasi. Diketahui bahwa baik pada kondisi mental netral, konsentrasi maupun rileks memiliki rentang frekuensinya antara 0 – 150 Hz, hal ini menunjukkan bahwa pada masing-masing kondisi mental mengandung *sub-band* gelombang delta, teta, alfa, beta dan gamma. Dapat dilihat bahwa plot PSD kondisi netral dan konsentrasi memiliki rentang nilai PSD yang hampir sama, namun pada kondisi mental konsentrasi masih memiliki rentang nilai PSD yang lebih tinggi jika dibandingkan dengan kondisi netral dan rileks. Nilai rentang PSD yang paling rendah adalah saat seseorang dalam keadaan mental rileks. Hal ini menunjukkan bahwa ketika seseorang sedang dalam kondisi konsentrasi memiliki distribusi energi frekuensi yang lebih tinggi dibandingkan saat seseorang dalam keadaan mental netral maupun rileks.

4.2.2 Filterisasi dan Segmentasi Data

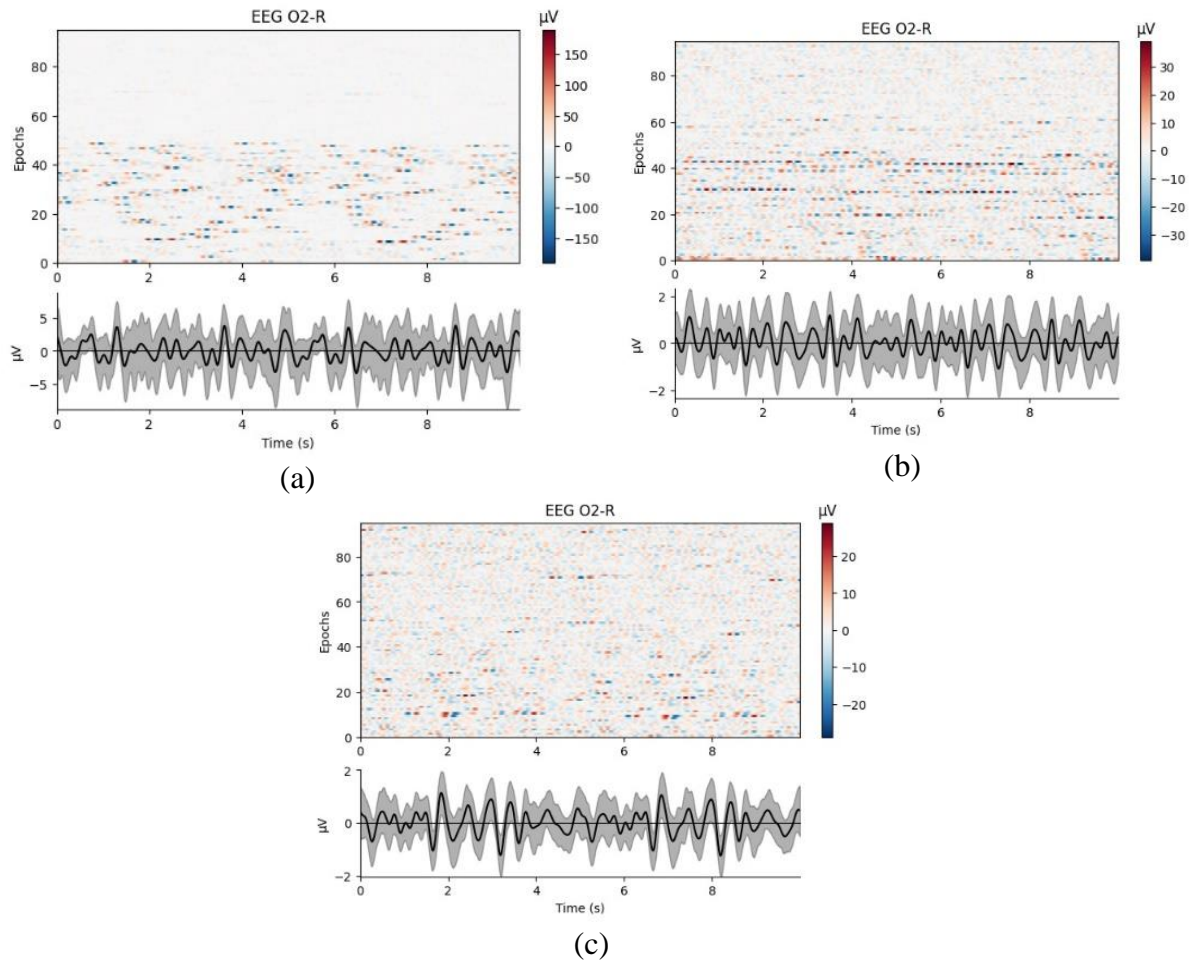
Filterisasi sinyal merupakan tahapan dimana sinyal akan dibagi ke dalam lima subband yaitu delta, teta, alfa, beta dan gamma, dimana *channel* yang dipilih pada penelitian ini adalah O2. Filterisasi menggunakan metode *Finite Impulse Response (FIR) Hamming Windows*. Hasil

dari proses filterisasi dapat digambarkan melalui grafik *Power Spectral Density* (PSD) sebagai berikut.



Gambar 4.5 Plot PSD (a) Sebelum Filterisasi, (b) Sub-Band Alfa, (c) Sub-Band Beta, (d) Sub-Band Beta, (e) Sub-Band Gamma, (f) Sub-Band Teta

Grafik PSD merupakan grafik yang dapat mencerminkan besaran amplitude sinyal pada suatu frekuensi tertentu. Pada Gambar 4.4 (a) diketahui bahwa rentang frekuensi yang ditunjukkan dengan nilai PSD masih menyebar, lalu setelah dilakukan filterisasi diperoleh bahwa sinyal EEG berada sesuai dengan rentangnya. Setelah dilakukan filterisasi selanjutnya dilakukan segmentasi data. Segmentasi data merupakan tahap dimana dilakukan pemotongan sepuluh detik dengan overlapping lima detik, sehingga untuk satu data dengan durasi perekaman selama 8 menit (480 detik) akan menghasilkan 95 *epoch*. Masing-masing *epoch* akan memiliki 5.000 beda tegangan. Contoh proses hasil segmentasi data pada orang pertama dapat dilihat pada gambar di bawah ini.



Gambar 4.6 Epoching Data Orang ke-1 (a) Kondisi Netral, (b) Kondisi Rileks, (c) Kondisi Konsentrasi

4.2.3 Ekstraksi Fitur

Ekstraksi fitur merupakan proses untuk mengekstraksi fitur dari fitur asli baik dengan transformasi linier atau non-linier. Proses ekstraksi fitur dilakukan setelah proses filterisasi yaitu untuk mendapatkan sinyal dengan *sub-band* delta, teta, alfa, beta dan gamma dan setelah proses dilakukan segmentasi data. Fitur yang diekstrak terdiri dari fitur minimum, median, maksimum, varians, *skewness*, kurtosis, energi dan entropi. Dalam penelitian ini diperoleh sebanyak 40 fitur. Berikut merupakan rangkuman rentang dari fitur hasil ekstraksi yang disajikan pada Tabel 4.1.

Tabel 4.1 Rangkuman Hasil Ekstraksi Fitur

Kolom	Nilai Maksimum	Nilai Minimum
minimum_delta_O2	$-5,852 \times 10^{-6}$	-0,00049
minimum_teta_O2	$-2,859 \times 10^{-6}$	-0,00018
minimum_alfa_O2	$-4,263 \times 10^{-9}$	$-1,802 \times 10^{-7}$
minimum_beta_O2	$-6,893 \times 10^{-12}$	$-2,517 \times 10^{-10}$
minimum_gamma_O2	$-1,119 \times 10^{-14}$	$-3,212 \times 10^{-13}$

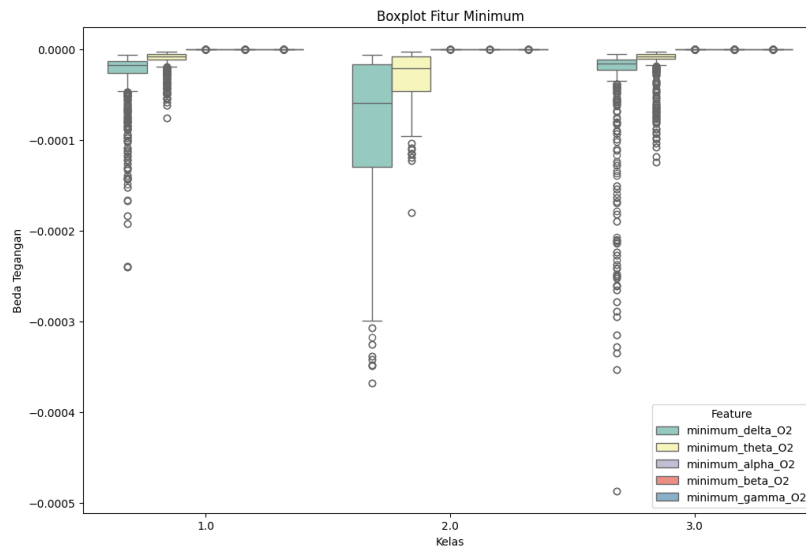
Tabel 4.1 Rangkuman Hasil Ekstraksi Fitur (Lanjutan)

Kolom	Nilai Maksimum	Nilai Minimum
median_delta_O2	$4,461 \times 10^{-5}$	$-1,126 \times 10^{-5}$
median_teta_O2	$5,216 \times 10^{-7}$	$-5,464 \times 10^{-7}$
median_alfa_O2	$1,192 \times 10^{-9}$	$-1,477 \times 10^{-9}$
median_beta_O2	$2,869 \times 10^{-12}$	$-1,538 \times 10^{-12}$
median_gamma_O2	$2,549 \times 10^{-15}$	$-5,087 \times 10^{-15}$
maksimum_delta_O2	0,00061	$5,187 \times 10^{-6}$
maksimum_teta_O2	0,00019	$2,958 \times 10^{-6}$
maksimum_alfa_O2	$1,823 \times 10^{-7}$	$4,063 \times 10^{-9}$
maksimum_beta_O2	$2,428 \times 10^{-10}$	$6,717 \times 10^{-12}$
maksimum_gamma_O2	$3,296 \times 10^{-13}$	$1,186 \times 10^{-14}$
variance_delta_O2	$9,174 \times 10^{-9}$	$5,472 \times 10^{-12}$
variance_teta_O2	$1,937 \times 10^{-9}$	$1,172 \times 10^{-12}$
variance_alfa_O2	$2,218 \times 10^{-15}$	$2,985 \times 10^{-18}$
variance_beta_O2	$6,323 \times 10^{-21}$	$6,416 \times 10^{-24}$
variance_gamma_O2	$9,992 \times 10^{-27}$	$1,622 \times 10^{-29}$
skewness_delta_O2	3,694	-4,744
skewness_teta_O2	3,026	-2,223
skewness_alfa_O2	1,059	-0,678
skewness_beta_O2	0,818	-0,734
skewness_gamma_O2	0,597	-0,679
kurtosis_delta_O2	39,395	-0,995
kurtosis_teta_O2	23,688	-1,012
kurtosis_alfa_O2	15,759	-0,849
kurtosis_beta_O2	12,999	-1,068
kurtosis_gamma_O2	13,362	-0,928
energi_delta_O2	$4,587 \times 10^{-5}$	$2,736 \times 10^{-8}$
energi_teta_O2	$9,685 \times 10^{-6}$	$5,862 \times 10^{-9}$
energi_alfa_O2	$1,109 \times 10^{-11}$	$1,492 \times 10^{-14}$
energi_beta_O2	$3,162 \times 10^{-17}$	$3,208 \times 10^{-20}$
energi_gamma_O2	$4,996 \times 10^{-23}$	$8,110 \times 10^{-26}$
entropi_delta_O2	1,380	0,729
entropi_teta_O2	1,385	1,170
entropi_alfa_O2	2,212	1,221
entropi_beta_O2	2,514	1,085
entropi_gamma_O2	2,405	1,223

Selanjutnya dilakukan visualisasi data dengan menggunakan *boxplot* untuk masing-masing kondisi mental dan masing-masing *sub-band*. Kategori kelas 1 adalah kondisi mental netral, kategori kelas 2 merupakan kondisi mental konsentrasi dan kategori kelas 3 merupakan kondisi mental rileks. Data yang digunakan untuk visualisasi adalah data ketiga kondisi mental pada orang pertama. Hasil visualisasi dapat dilihat sebagai berikut.

a) Fitur Minimum

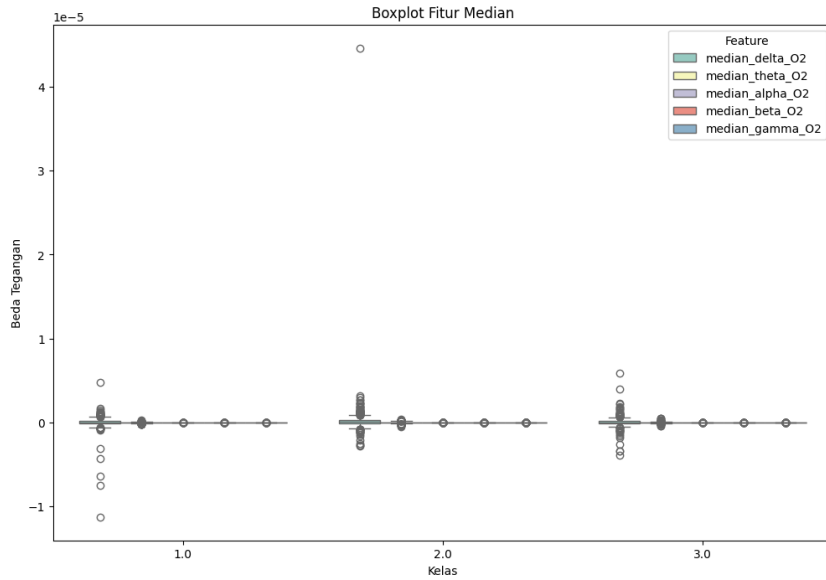
Nilai minimum merupakan nilai sampel terkecil dimana memiliki amplitude terkecil yang dimiliki oleh sinyal itu dari setiap *sub-band*. Pada Gambar 4.5 merupakan *boxplot* dari hasil ekstraksi fitur minimum pada ketiga kondisi mental dan masing-masing *sub-band* nya. Gambar 4.6 menunjukkan bahwa *sub-band* delta memiliki rentang nilai yang paling lebar dibandingkan dengan *sub-band* yang lainnya. Terlihat bahwa terdapat data *outlier* pada ketiga kondisi mental. Diketahui pula bahwa banyakk terdapat *outlier* pada *sub-band* delta dan teta.



Gambar 4.7 *Boxlot* Fitur Minimum

b) Fitur Median

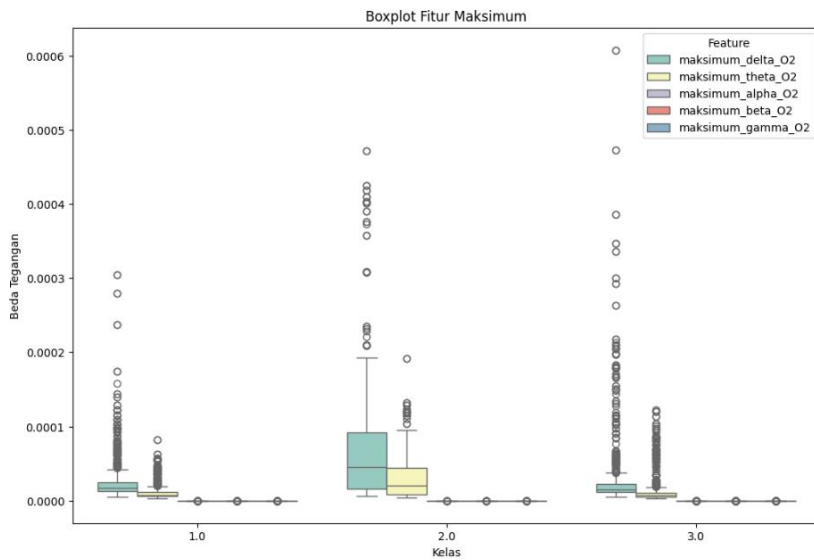
Median merupakan nilai tengah dari suatu kumpulan titik pada sinyal EEG setelah diurutkan dari yang terkecil hingga terbesar. *Boxplot* dari hasil ekstraksi fitur median pada ketiga kondisi mental dan masing-masing *sub-band* nya ditunjukkan pada Gambar 4.7. Diketahui bahwa *sub-band* delta memiliki rentang nilai yang paling lebar dibandingkan dengan *sub-band* yang lainnya. Terlihat bahwa terdapat data *outlier* pada ketiga kondisi mental dan *outlier* paling tinggi terdapat pada kondisi mental konsentrasi dan pada *sub-band* delta.



Gambar 4.8 *Boxlot* Fitur Median

c) Fitur Maksimum

Nilai maksimum merupakan nilai sampel terbesar dimana memiliki *amplitude* terbesar yang dimiliki oleh sinyal itu dari setiap sub-band. *Boxplot* dari hasil ekstraksi fitur maksimum pada ketiga kondisi mental dan masing-masing *sub-band* nya ditunjukkan pada Gambar 4.8.

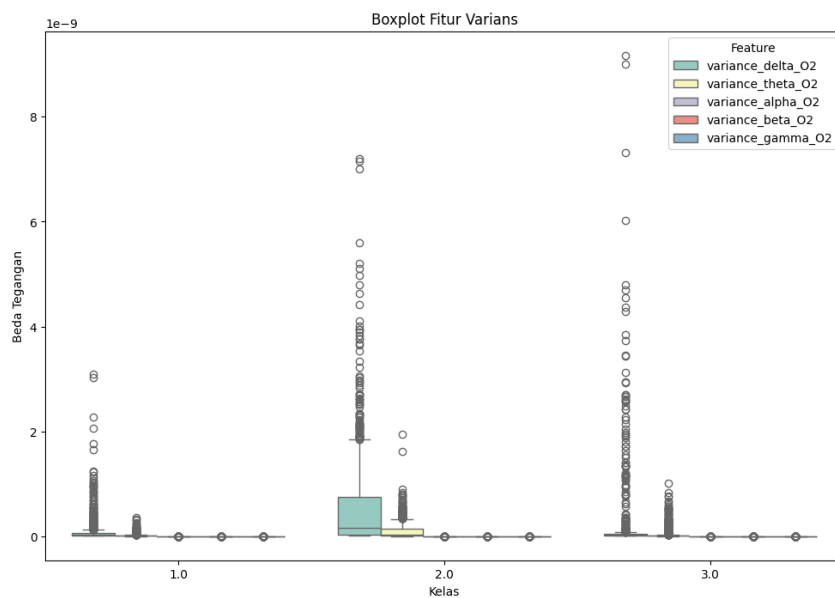


Gambar 4.9 *Boxlot* Fitur Maksimum

Diketahui bahwa *sub-band* delta memiliki rentang nilai yang paling lebar dibandingkan dengan *sub-band* yang lainnya. Terlihat bahwa terdapat data *outlier* pada ketiga kondisi mental dan *outlier* paling tinggi terdapat pada kondisi mental konsentrasi dan pada *sub-band* delta.

d) Fitur Varians

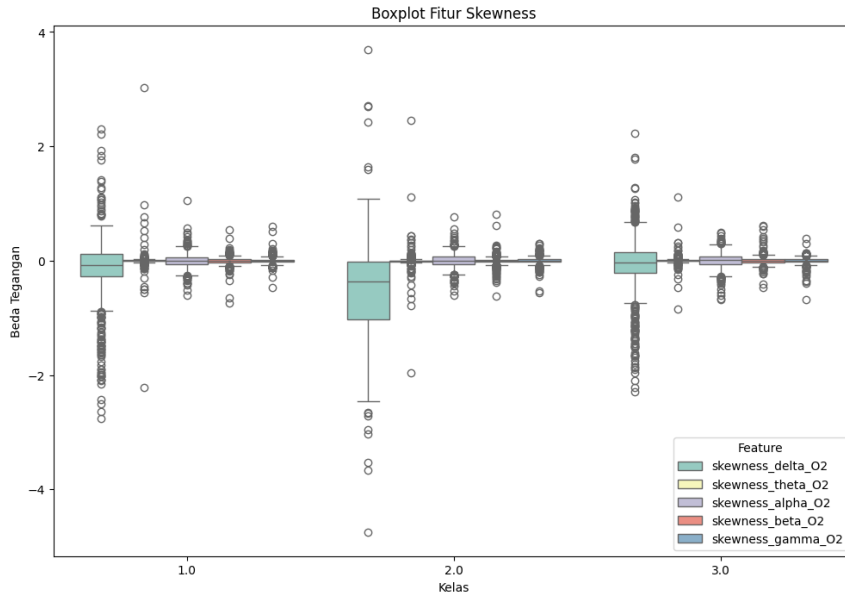
Nilai varians dapat digunakan untuk melihat bagaimana tingkat persebaran dari titik beda potensial listrik dari hasil perekaman EEG. Gambar 4.9 menunjukkan *boxplot* dari hasil ekstraksi fitur varians pada ketiga kondisi mental dan masing-masing *sub-band* nya. Diketahui bahwa pada kondisi mental netral dan rileks kelima *sub-band* tidak terlihat jauh perbedaan rentang nilainya. *Sub-band* delta pada kondisi mental konsentrasi memiliki rentang nilai yang paling lebar dibandingkan dengan yang lainnya. Terlihat bahwa terdapat data *outlier* pada ketiga kondisi mental dan *outlier* paling tinggi terdapat pada kondisi mental konsentrasi dan pada *sub-band* delta.



Gambar 4.10 *Boxlot* Fitur Varians

e) Fitur Skewness

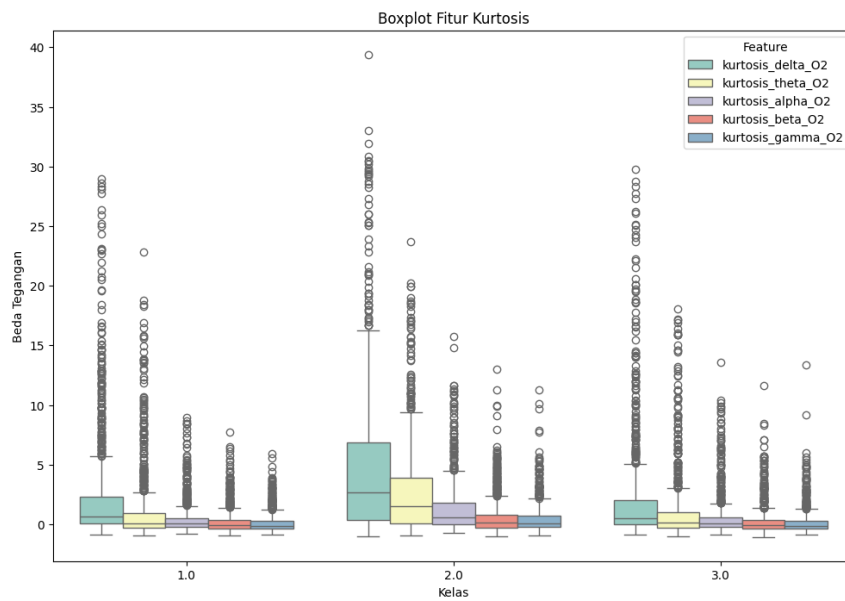
Skewness (kemiringan) adalah suatu nilai yang digunakan untuk menggambarkan tingkat ketidaksimetrisan suatu distribusi. Gambar 4.10 menunjukkan *boxplot* dari hasil ekstraksi fitur *skewness/* pada ketiga kondisi mental dan masing-masing *sub-band* nya. *Sub-band* delta pada kondisi mental konsentrasi memiliki rentang nilai yang paling lebar dibandingkan dengan yang lainnya dan terdapat data *outlier* pada ketiga kondisi mental.



Gambar 4.11 *Boxlot* Fitur *Skewness*

f) Fitur Kurtosis

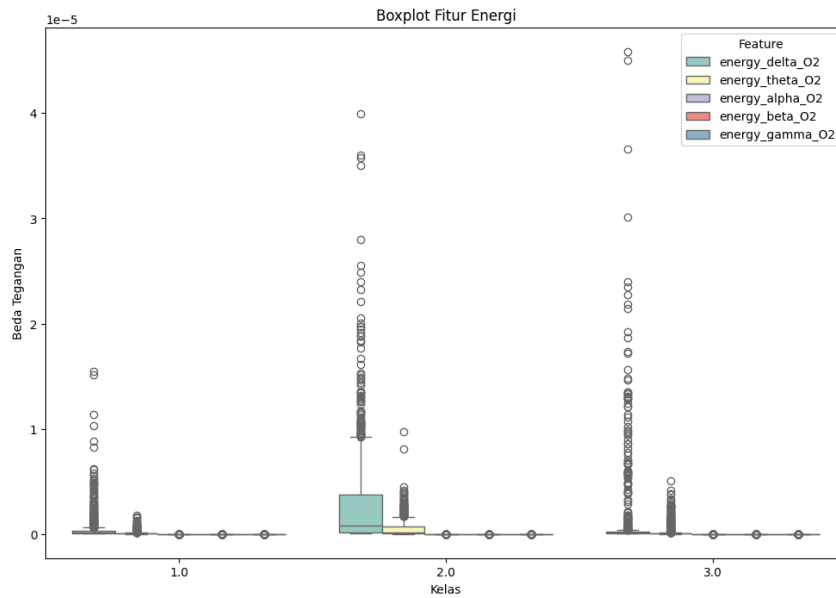
Kurtosis adalah derajat ketinggian puncak atau keruncingan dari suatu distribusi data terhadap distribusi normalnya. Gambar 4.11 menunjukkan *boxplot* dari hasil ekstraksi fitur kurtosis pada ketiga kondisi mental dan masing-masing *sub-band* nya. Dapat dilihat bahwa *sub-band* delta memiliki rentang nilai yang paling lebar dibandingkan dengan yang lainnya dan terdapat data *outlier* pada ketiga kondisi mental dan pada masing-masing *sub-band* nya.



Gambar 4.12 *Boxlot* Fitur *Kurtosis*

g) Fitur Energi

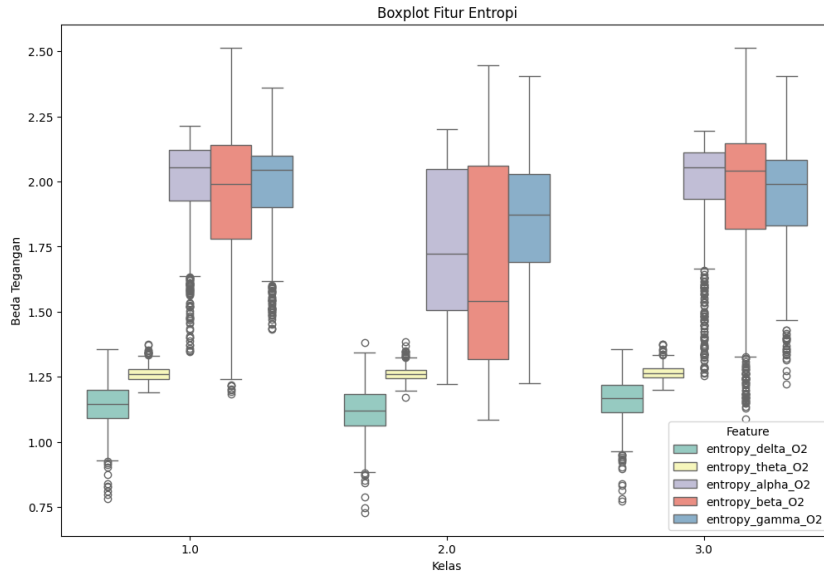
Energi sinyal EEG dapat berfungsi sebagai indikator intensitas aktivitas sinyal EEG dalam periode waktu tertentu. Pada Gambar 4.12 merupakan *boxplot* dari hasil ekstraksi fitur energi pada ketiga kondisi mental dan masing-masing *sub-band* nya. Gambar 4.12 menunjukkan bahwa tidak terdapat perbedaan rentang yang signifikan antar *sub-band* pada masing-masing kondisi mental. Terlihat bahwa *sub-band* delta pada kondisi mental konsentrasi memiliki rentang nilai yang paling lebar. Diketahui pula bahwa pada *sub-band* delta dan teta memiliki banyak data *outlier* ke atas.



Gambar 4.13 *Boxlot* Fitur Energi

h) Fitur Entropi

Fitur entropi dapat dijadikan sebagai ciri distribusi jikaspektrum frekuensisuatu sinyal dianggap sebagai suatu sistem. Pada Gambar 4.13 merupakan *boxplot* dari hasil ekstraksi fitur entropi pada ketiga kondisi mental dan masing-masing *sub-band* nya. Gambar 4.13 menunjukkan bahwa *sub-band* alfa, beta dan gamma memiliki nilai yang lebih tinggi dan rentang yang paling lebar dibandingkan dengan *sub-band* delta dan teta. Terlihat bahwa *sub-band* delta, alfa, beta dan gamma cenderung memiiki data *outlier* ke bawah, sedangkan pada *sub-band* gamma cenderung memiliki data *outlier* ke atas.



Gambar 4.14 Boxlot Fitur Entropi

4.2.4 Klasifikasi dengan menggunakan *Kernel Time Series* GTWED

Setelah dilakukan proses preprocessing data yang meliputi filterisasi, segmentasi dan ekstraksi fitur selanjutnya dilakukan klasifikasi *Multiclass Support Vector Machine* (MSM) dengan menggunakan kernel berbasis time series yaitu *Gaussian Time Warp Edit Distance* (GTWED). *Multiclass Support Vector Machine* merupakan proses klasifikasi dengan jumlah kelas lebih dari dua. Pada penelitian kali ini dilakukan dengan menggunakan pendekatan *One-Against-All* (OAA) atau *One-Versus-Rest* (OVR) dan *One-Against-One* (OAO) atau *One-Versus-One* (OVO).

Sebelum dilakukan pemodelan, terlebih dahulu data dibagi menjadi data training dan data testing menggunakan metode *hold out* 70:30, yang berarti bahwa 70% data akan masuk sebagai data training dan 30% sisanya akan masuk sebagai data testing. Pada penelitian ini data *training* adalah 7 orang pertama pada masing-masing kondisi mental, dan 3 orang sisanya adalah sebagai data *testing*. Pemodelan MSM dengan menggunakan GTWED dilakukan dengan menggunakan bantuan *software* Python. Pada penelitian ini akan dilakukan evaluasi model terbaik apakah menggunakan OAO atau OAA.

4.2.4.1 Pendekatan *One-Against-All* (OAA)

Metode OAA adalah suatu metode klasifikasi dalam permasalahan *multiclass* dimana permasalahan *multiclass* dibagi kedalam m -sub masalah biner, dimana jika salah satu kelas adalah kelas positif maka kelas yang lain akan diklasifikasikan sebagai kelas negatif. Setelah dilakukan pemodelan pada data *training*, selanjutnya dilakukan evaluasi model pada data

testing. Hasil klasifikasi dalam *confusion matrix* dengan menggunakan nilai parameter $\gamma = 0, \lambda = 1 \times 10^{-5}, \sigma = 1$ dan $C = 1$ adalah sebagai berikut.

Tabel 4.2 *Confusion Matrix* Kernel GTWED Pendekatan OAA

		Prediksi		
		1 : Netral	2 : Rileks	3 : Konsentrasi
Aktual	1 : Netral	15	150	120
	2 : Rileks	49	134	102
	3 : Konsentrasi	5	74	206

Tabel 4.2 menunjukkan hasil *confusion matrix* klasifikasi kondisi mental pendekatan OAA dengan menggunakan kernel GTWED. Diketahui bahwa ketika menggunakan kernel GTWED pada ketiga kondisi memiliki hasil prediksi yang tidak sesuai dengan data aktualnya. Diketahui bahwa pada kondisi netral terdapat 15 data yang hasil prediksinya sesuai dengan data aktual dan sisanya diprediksi salah. Pada kondisi rileks diketahui terdapat 134 data yang hasil prediksinya sesuai dengan data aktual dan sisanya diprediksi salah, sedangkan pada kondisi konsentrasi diketahui bahwa terdapat 206 data yang hasil prediksinya sesuai dengan data aktual dan sisanya diprediksi salah.

4.2.4.2 Pendekatan *One-Against-One* (OAO)

Metode OAO adalah suatu metode klasifikasi dalam permasalahan multiclass dimana permasalahan multiclass dengan e kelas kedalam $e(e-1)/2$ biner sub permasalahan. Setelah dilakukan pemodelan pada data *training*, selanjutnya dilakukan evaluasi model pada data testing. Hasil klasifikasi disajikan dalam *confusion matrix* dengan menggunakan nilai parameter $\gamma = 0, \lambda = 1 \times 10^{-5}, \sigma = 1$ dan $C = 1$ adalah sebagai berikut.

Tabel 4.3 *Confusion Matrix* Kernel GTWED Pendekatan OAO

		Prediksi		
		1 : Netral	2 : Rileks	3 : Konsentrasi
Aktual	1 : Netral	179	13	93
	2 : Rileks	68	96	121
	3 : Konsentrasi	94	9	182

Tabel 4.3 menunjukkan hasil *confusion matrix* klasifikasi kondisi mental pendekatan OAO dengan menggunakan kernel GTWED. Diketahui bahwa ketika menggunakan kernel GTWED pada ketiga kondisi memiliki hasil prediksi yang tidak sesuai dengan data aktualnya. Diketahui bahwa pada kondisi netral terdapat 179 data yang hasil prediksinya sesuai dengan data aktual dan sisanya diprediksi salah. Pada kondisi rileks diketahui terdapat 96 data yang

hasil prediksinya sesuai dengan data aktual dan sisanya diprediksi salah, sedangkan pada kondisi konsentrasi diketahui bahwa terdapat 182 data yang hasil prediksinya sesuai dengan data aktual dan sisanya diprediksi salah.

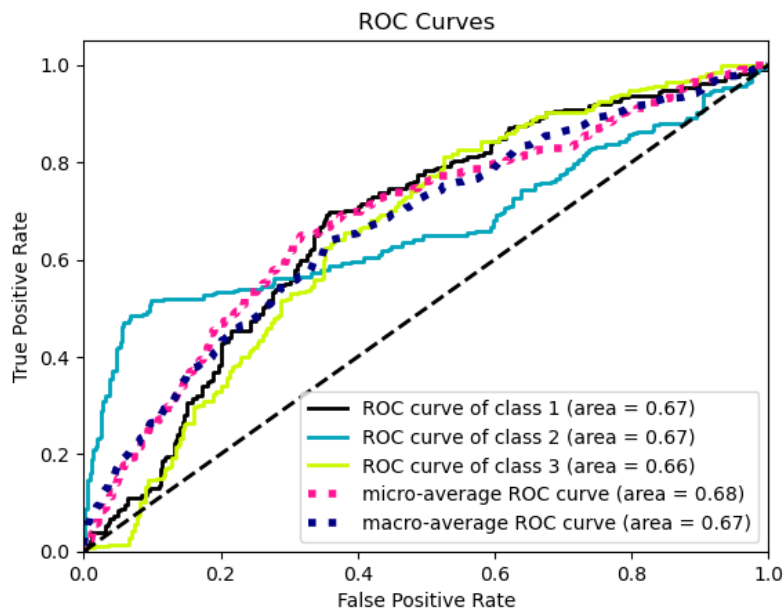
4.2.4.3 Pemilihan Model Terbaik

Selanjutnya, pemilihan model terbaik didasarkan pada nilai kebaikan model yang paling tinggi. Nilai pendekatan model memiliki rentang nilai 0 – 1, semakin mendekati nilai 1 adalah model yang terbaik.

Tabel 4.4 Kebaikan Model Pendekatan OAA dan OAO

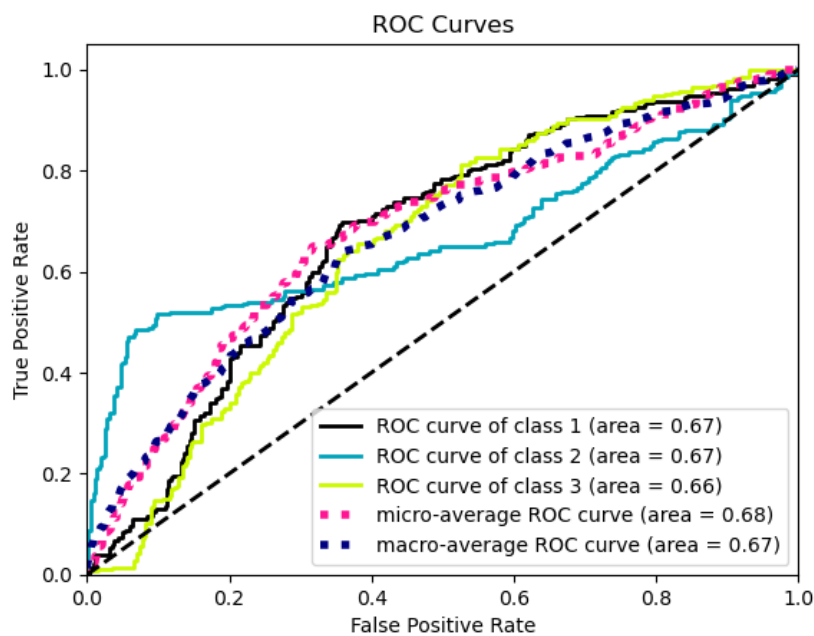
Pendekatan	Sensitivitas	Spesifisitas	Akurasi	Presisi	<i>F1-Score</i>	AUC
OAA	0,415	0,708	0,415	0,358	0,424	0,561
OAO	0,535	0,767	0,534	0,60	0,467	0,651

Berdasarkan hasil kebaikan model pada Tabel 4.4 diketahui bahwa pada pendekatan OAA dengan menggunakan kernel GTWED memiliki nilai sensitivitas sebesar 0,415, nilai spesifisitas sebesar 0,708, nilai akurasi sebesar 0,415, nilai presisi sebesar 0,358, *F1-Score* sebesar 0,424 dan AUC sebesar 0,561. Sedangkan pada pendekatan OAA dengan menggunakan kernel GTWED memiliki nilai sensitivitas sebesar 0,535, nilai spesifisitas sebesar 0,767, nilai akurasi sebesar 0,534, nilai presisi sebesar 0,6, *F1-Score* sebesar 0,467 dan AUC sebesar 0,651. Selanjutnya dilakukan pembentukan grafik ROC untuk mengetahui ilustrasi performansi yang dapat dilihat pada gambar sebagai berikut.



Gambar 4.15 ROC Pendekatan OAA

Berdasarkan Gambar 4.15 diketahui bahwa nilai *ROC curve of class 1* adalah sebesar 0,67, untuk kelas 2 adalah sebesar 0,67, dan untuk kelas 3 adalah sebesar 0,66. Sedangkan nilai *micro average ROC curve* dan *macro-average ROC curve* diperoleh masing-masing sebesar 0,68 dan 0,67. Oleh karena itu berdasarkan tingkat keakuratan berdasarkan nilai AUC pada Tabel 2.4 dapat diartikan bahwa model dengan kernel GTWED dengan pendekatan OAA kurang baik untuk mengklasifikasikan keadaan mental. Selanjutnya dilakukan pembentukan grafik ROC pendekatan OAO untuk mengetahui ilustrasi performansi yang dapat dilihat pada Gambar 4.16 sebagai berikut.



Gambar 4.16 ROC Pendekatan OAO

Berdasarkan Gambar 4.16 diketahui bahwa nilai *ROC curve of class 1* adalah sebesar 0,67, untuk kelas 2 adalah sebesar 0,67, dan untuk kelas 3 adalah sebesar 0,66. Sedangkan nilai *micro average ROC curve* dan *macro-average ROC curve* diperoleh masing-masing sebesar 0,68 dan 0,67. Oleh karena itu dapat diartikan bahwa model dengan kernel GTWED dan pendekatan OAO juga kurang baik untuk mengklasifikasikan keadaan mental. Namun secara keseluruhan pendekatan OAO memiliki nilai performansi lebih baik dibandingkan ketika menggunakan pendekatan OAA.

(Halaman ini sengaja dikosongkan)

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, didapatkan kesimpulan untuk menjawab rumusan masalah pada penelitian ini adalah sebagai berikut.

1. TWED mengembangkan matrik dengan memasukkan stamp waktu dari data *time series*. Kernel GTWED didapatkan dengan mensubstitusikan perhitungan jarak TWED kedalam persamaan kernel *Gaussian* sehingga didapatkan persamaan kernel GTWED sebagai berikut.

$$k_{gtwed}(\mathbf{A}_1^p, \mathbf{B}_1^q) = \exp\left(-\frac{\delta_{\lambda, \gamma}(\mathbf{A}_1^p, \mathbf{B}_1^q)}{2\sigma^2}\right)$$

2. Diketahui bahwa sinyal EEG ketika dalam kondisi rileks memiliki rentang beda tegangan yang paling kecil, hal ini menandakan bahwa seseorang tidak sedang melakukan aktivitas yang intens. Sinyal EEG ketika dalam kondisi netral memiliki beda tegangan yang lebih kecil dibandingkan sinyal EEG ketika dalam kondisi konsentrasi, menandakan bahwa gelombang otak berada diantara kondisi yang relative stabil, yaitu tidak terlalu banyak aktivitas namun belum berada pada kondisi rileks. Diketahui pula bahwa sinyal EEG ketika dalam kondisi konsentrasi memiliki rentang beda tegangan yang paling tinggi, dimana hal ini menandakan bahwa sedang terdapat aktivitas yang intens pada otak seseorang. Pada tahapan segmentasi data dilakukan pemotongan sepuluh detik dengan overlapping lima detik, sehingga untuk satu data dengan durasi perekaman selama 8 menit (480 detik) akan menghasilkan 95 *epoch*. Hasil dari ekstraksi fitur dengan visualisasi *boxplot* pada masing-masing *sub-band* terlihat bahwa *sub-band* delta memiliki penyebaran data yang cenderung lebih lebar dibandingkan dengan *sub-band* lainnya, namun jika dibandingkan pada setiap kondisi mental *boxplot* hasil ekstraksi fitur menunjukkan bahwa penyebaran data terlihat tidak jauh berbeda. Hasil kalifikasi *multiclass* SVM pendekatan OAA memiliki AUC sebesar 0,561 dan AUC pada pendekatan OAO adalah sebesar 0,651, kedua pendekatan menunjukkan bahwa hasil klasifikasi yang kurang baik, namun berdasarkan nilai AUC pendekatan OAO lebih baik dibandingkan dengan pendekatan OAA.

5.2 Saran

Berdasarkan hasil analisis dan kesimpulan yang didapatkan, berikut merupakan saran untuk penelitian selanjutnya sebagai berikut.

1. Melakukan eksplorasi terhadap metode filterisasi yang digunakan untuk bisa lebih baik dalam menghilangkan *noise*, seperti menggunakan filterisasi *butterworth* dan lainnya.
2. Memperhatikan label *time stamp* pada data agar terbentuk data *time series* yang tepat. Pada penelitian ini label *time stamp* masih berurutan dari subjek 1 dilanjutkan dengan subjek 2 dan seterusnya, sedangkan sebenarnya setiap subjek memiliki masing-masing *time stamp* yang sama.
3. Mencari metode yang lebih tepat dalam pembagian data *training* dan data *testing* agar dapat merepresentasikan karakteristik data.
4. Mengetahui *true label* kondisi mental yang sedang dialami subjek pada setiap rentang waktu nya dengan memperhatikan kondisi gelombang otak berdasarkan pengamatan ahli. Label kondisi mental yang digunakan pada penelitian ini berdasarkan pada intruksi saat dilakukan perekaman EEG.
5. Melakukan eksplorasi *channel* lain yang dapat digunakan untuk klasifikasi kondisi mental, karena pada penelitian ini masih berfokus pada satu *channel*.
6. Melakukan eksplorasi mengenai kernel-kernel *time series* yang dapat digunakan untuk klasifikasi data sinyal yang lebih *cost efficient*.
7. Melakukan *tunning* parameter untuk mengetahui parameter yang optimum sehingga mendapatkan hasil performansi yang baik.

DAFTAR PUSTAKA

- Abhang, P. A. (2016). Technical Aspects of Brain Rhythms and Speech Parameters. In P. A. Abhang, *Introduction to EEG- and Speech-Based Emotion Recognition*, 51–79. doi:10.1016/b978-0-12-804490-2.00003-8 (pp. 51-76). Elsevier.
- Azevedo, F. e. (2009). Equal Numbers of Neuronal and Nonneuronal Cells Make the Human Brain an Isometrically Scaled-Up Primate Brain. *The Journal of Comparative Neurology*, 532-541.
- Baron-Cohen S, R. H. (1994). Recognition of Mental State Terms Clinical findings in children with autism and a functional neuroimaging study of normal adults. *British Journal of Psychiatry*, 640-649.
- Baron-Cohen, S. R. (1994). Recognition of Mental State Terms (Clinical Findings in Children with Autism and a Functional Neuroimaging Study of Normal Adults). *British Journal of Psychiatry*, 640-649.
- Bekkari, A. e. (2013). SVM Classification of Urban High-Resolution Imagery Using Composite Kernels and Contour Information. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 126-141.
- Bertsimas, D. &. (2007). Classification and Regression via Integer Optimization. *Operations Research*, 252-271.
- Bird, J. J. (2018). A Study on Mental State Classification using EEG-based Brain-Machine Interface. *2018 International Conference on Intelligent Systems (IS)* (pp. 795-800). Funchal-Madeira, Portugal: IEEE.
- Borck, C. (2005). *Brainwaves : A Cultural History of Electroencephalography*. New York: Routledge.
- Cahyo, L. B. (2018). *Implementasi Metode Support Vector Machine Untuk Melakukan Klasifikasi Pada Data Bioinformatika*. Yogyakarta: Universitas Islam Indonesia.
- Caroline., S. N. (2020). Analisis Aplikasi Filter FIR dan Filter IIR dalam Pra-pemrosesan Sinyal Elektroensefalografi. *ULTIMA Computing*, 40-48.
- Cole, S. R. (2017). Brain Oscillations and the Importance of Waveform Shape. *Trends in Cognitive Sciences*, 137-149.
- De Gennaro, L. ,. (2008). The electroencephalographic fingerprint of sleep is genetically determined: A twin study. *Annals of Neurology*, 455-460.
- Dimurtadha, M. E. (2019). Analisis Filter Finite Impulse Response (FIR) pada Sinyal Electroensefalogram (EEG). *Seminar Nasional dan Expo Teknik Elektro*, 101-104.

- Frederick, J. A., & Lubar, J. F. (2002). Skewness in the Time Series of EEG Magnitude and Spectral Correlation. ResearchGate.
- Galar, M. e. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognit*, 1761-1776.
- Gorunescu, F. (2011). *Data Mining: Concepts, Models, and Techniques*. Germany: Springer.
- Gorunescu, F. (2011). *Data Mining: Concepts, Models, and Techniques*. Jerman: Springer.
- Guodong Guo, S. Z. (2000). Face recognition by support vector machines. *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)* (pp. 196-201). Grenoble, France: IEEE.
- H. Lei., a. B. (2007). A Study on the Dynamic Time Warping in Kernel Machines. *Third International IEEE Conference on Signal-Image Technologies and Internet-Based* (pp. 839-845). Shanghai, China: IEEE.
- Hamdani, T. e. (2011). An Iterative Method for Deciding SVM and Single Layer Neural Network Structures. *Neural Process Lett*, 171-186.
- Han, J. K. (2012). *Data Mining : Concepts and Technique 3rd edition*. USA: Morgan Kaufman.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. USA: Elsevier Inc.
- Hand, D. T. (2011). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. In *Machine Learning* (pp. 171-186). Netherlands: Kluwer Academic Publishers.
- Hardle, W. P. (2014). *Support Vector Machines with Evolutionary Feature Selection for Default Prediction dalam The Oxford Handbook of Aplpied Nonparametric and Semiparametric of Applied Nonparametric and Semiparametric Econometrics and Statistics*. Oxford: Oxford University.
- Hidayati, T. d. (2019). *Statistika Dasar Panduan Bagi Dosen dan Mahasiswa*. Purwokerto: CV. Pena Persada.
- Hidayati, T., Handayani, I., & Ikasari, I. H. (2019). *Statistika Dasar Panduan Bagi Dosen dan Mahasiswa*. Purwokerto: CV. Pena Persada.
- Islam, M. L. (2022). Multivariate Empirical Mode Decomposition of EEG for Mental State Detection at Localized Brain Lobes. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 11-15.

- Islam, M. S., Islam, M. S., Rahman, S. K., & et al. (2014). Design of FIR Filter Using Hamming Window. *International Journal of Emerging Reseaech in Management & Technology*, 2278-9359.
- Islamiyah, W. &. (2020). Identification of Epilepsy Phase Based on Time Domain Feature Using ECG Signals. *2020 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM 2020)* (pp. 91-96). Surabaya: IEEE.
- Jia, D., Zhang, D., & Li, N. (2014). Pulse Waveform Classification Using Support Vector Machine with Gaussian Time Warp Edit Distance Kernel. *Computational and Mathematical Methods in Medicine*.
- Jung, D. e. (2022). EEG-Based Identification of Emotional Neural State Evoked by Virtual Environment Interaction. *International Journal of Enviromental Research and Public Health*, 1-15.
- Karimah, S. (2023). *Klasifikasi Mental State berdasarkan Data Rekaman Electroencephalography (EEG) menggunakan Multiclass Support Vector Machine*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Karmila, R. e. (2016). Identifikasi Tingkat Konsentrasi Dari Sinyal EEG Dengan Wavelet dan Adaptive Backpropagation. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 23-27.
- Kessels, R. P. (2019). Improving precision in neuropsychological assessment: Bridging the gap between classic paper-and-pencil tests and paradigms from cognitive neuroscience. *The Clinical Neuropsychologist*, 357-368.
- Khakim, Z. K. (2021). Dasar - Dasar Electroencephalography (EEG) bagi Riset Psikologi. *Buletin Psikologi*, 92-115.
- Khanal, B. e. (2018). Mental State Prediction by Deployment of Trained SVM Model on EEG Brain Signal. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (pp. 82-85). Kathmandu (Nepal): IEEE.
- Koctúrová, M. &. (2021). EEG-based Speech Activity Detection. *Acta Polytechnica Hungarica*, 65-77.
- Krüger, F. (2016). *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models*. Rostock: University of Rostock.
- Lana, E. A. (2015). Detection of movement intention using EEG in a human-robot interaction environment. *Research on Biomedical Engineering*, 285-294.

- Li, X. C. (2014). Classification of EEG Signals Using a Multiple Kernel Learning Support Vector Machine. *Sensors*, 12784-12802.
- Lim, K., Duc, M., Thien, M., Yuvaraj, R., & Fogarty, J. (2022). Investigating the Effects of Microclimate on Physiological Stress and Brain Function with Data Science and Wearables. *Sustainability*, 10769.
- Lionitama, V., Djamal, E., & Kasyidi, F. (2024). Identifikasi Emosi Melalui Sinyal Elektroensefalogram Menggunakan Graph Convolutional Network. *MIND (Multimedia Artificial Intelligent Networking Database)*, 42-51.
- Liu, B. K. (2022). An End-to-End Depression Recognition Method Based on EEGNet. *Frontiers in Psychiatry*, 1-9.
- Luo, G. e. (2017). Recognition of Traveling Surges in HVDC with Wavelet Entropy. *Entropy*, 1-13.
- Maharto, S. P. (2019). Electroencephalogram (EEG) Signal Analysis for Diagnosis of Major Depressive Disorder (MDD): A Review. *Nanoelectronics, Circuits and Communication Systems*, 323-335.
- Marteau, P. F. (2008). Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 306-318.
- Marteau, P.-F. (2008). Time Warp Edit Distance. *Technical Report, Valoria Universite Europeenne De Bretagne*, lv5.
- Marzbani, H. R. (2016). Methodological Note: Neurofeedback: A Comprehensive Review on System Design, Methodology and Clinical Applications. *Basic and Clinical Neuroscience*, 143-157.
- Mohajon, J. (2020, Mei 29). *Confusion Matrix for Your Multi-Class Machine Learning Model*. Retrieved from Towards Data Science: <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>
- Mooniarsih, N. T. (2010). Desain dan Simulasi Filter menggunakan Metode Windowing. *ELKHA*, 41-47.
- Muller, K. e. (1997). Predicting Time Series with Support Vector Machine. *International Conference on Artificial Neural Networks 1997*, 999-1004.
- Nguyen, T. e. (2015). EEG signal classification for BCI applications by wavelets and interval type-2 fuzzy logic systems. *Expert Systems with Applications*, 4370-4380.

- Nikmah, A. (2023). *Smooth Support Cevtor Machine berbasis Polynomial Function untuk Deteksi Depresi menggunakan Sinyal Electroencephalogram (EEG)*. Surabaya: Intitut Teknologi Sepuluh Nopember.
- Nurgiyatna, N. &. (2004). Implementasi Filter Digital Infinite Impulse Response pada DSP TMS320C6711. *Jurnal Penelitian Sains & Teknologi*, 68-80.
- Olbrich, S. M. (2013). EEG biomarkers in major depressive disorder: Discriminative power and prediction of treatment response. *International Review of Psychiatry*, 604-618.
- Ovirianti, N. e. (2022). Support Vector Machine Using A Classification Algorithm. *SinkrOn (Jurnal & Penelitian Teknik Informatika)*, 2103-2107.
- Pasaribu, A. (1975). *Pengantar Statistik*. Jakarta: Gahlia Indonesia.
- Pasaribu, A. (1975). *Pengantar Statistik*. Jakarta: Gahlia Indonesia.
- Purnami, S. N. (2021). Least Square Support Vector Machine for Interictal Detection Based on EEG of Epilepsy Patients at Airlangga University Hospital Surabaya-Indonesia. In: Hemanth, J., Yigit, T., Patrut, B., Angelopoulou, A. (eds) Trends in Data Engineering Methods for Intel. *ICAIAME 2020. Lecture Notes on Data Engineering and Communications Technologies, vol 76*, 198-210.
- R. P. Balandong, R. F. (2018). A Review on EEG-Based Automatic Sleepiness Detection Systems for Driver. *IEEE Access, vol. 6*, 22908-22919.
- Rahman, A. &. (2022). Detection of Mental State from EEG Signal Data: An Investigation with Machine Learning Classifiers. *2022 International Conference on Knowledge and Smart Technology (KST)*. Chonburi, Thailand: IEEE.
- Richer, R. e. (2018). Real-time Mental State Recognition using a Wearable EEG. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society* (pp. 5495-5498). Honolulu, Hawaii, USA: IEEE.
- Runarsson, T. P. (2008). Support Vector Machines and Dynamic Time Warping for Time Series. *IJCNN'08* (pp. 2772-2776). Hongkong, China: IEEE.
- Saputra, N. H. (2023). EEG-based Statistical Analysis on Determining the Stress Mental State on Police Personnel. *2022 1st International Conference on Information System & Information Technology (ICISIT)* (pp. 181-186). Yogyakarta, Indonesia: IEEE.
- Scholkopf, B., & Smola, A. J. (2002). *Learning with Kernels*. London: The MIT Press.

- Shen, K.-Q. L.-P.-J.-Y.-S. (2008). EEG-based mental fatigue measurement using multi-class support vector machines with confidence estimat. *Clinical Neurophysiology*, 1524-1533.
- Stancin, I. e. (2021). A Review of EEG Signal Features and Their Application in Driver Drowsiness Detection Systems. *Sensors*, 1-29.
- Sun, S. (2008). The Extreme Energy Ratio Criterion for EEG Feature Extraction. *International Conference on Artificial Neural Networks*, 919-928.
- Tajiri, Y. e. (2010). Feature Extraction Using Support Vector Machines. *ICONIP*, 108-109.
- Tamura, T., & Chen, W. (2028). *Seamless Healthcare Monitoring*. Switzerland: Springer International Publishing.
- Tantika, R. (2022). Penggunaan Metode Support Vector Machine Klasifikasi Multiclass pada Data Pasien Penyakit Tiroid. *Bandung Conference Series: Statistics*, 159-166.
- TMSi. (2022, April 14). *Explaining the International 10-20 System for EEG*. Retrieved from What Is the 10-20 System for EEG?: <https://info.tmsi.com/blog/the-10-20-system-for-eeeg>
- TuerxunWaili, & A. (2020). Stress recognition using Electroencephalogram (EEG) signal. *Journal of Physics: Conference Series*, 1-6.
- Tzimourta., e. a. (2018). Evaluation of Window Size in Classification of Epileptic Short-Term EEG Signals Using a Brain Computer Interface Software. *Engineering, Technology & Applied Science Research*, 3093-3097.
- Wang, L. e. (2008). Two Criteria for Model Selection in Multiclass Support Vector Machines. *IEEE Transactions on Systems, Man and Cybernetics*, 1432-1448.
- Warren J, L. (1987). Effect of Epoch Length on Power Spectrum Analysis of the EEG. *Anesthesiology*, 66:489-495.
- WHO. (2022, Juni 16). *World mental health report: Transforming mental health for all*. Retrieved from World Health Organization: <https://www.who.int/publications/i/item/9789240049338>
- Widodo, A. &.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing* 21, 2560-2574.
- Witten, I. H., & Frank, E. (2005). *Data Mining: practical Machine Learning Tools and Techniques*. San Fransisco: Elsevier Inc.
- Wulandari, D. P. (2019). Epileptic Seizure Detection Based on Bandwidth Features of EEG Signals. *Procedia Computer Science*, 568-576.

- Wulansari, R., Dajalam, E., & Darmanto, T. (2016). KLASIFIKASI SINYAL EEG TERHADAP RANGSANGAN SUARA MENGGUNAKAN POWER SPECTRAL DENSITY DAN MULTILAYER PERCEPTRON. *Seminar Nasional Sains Dan Teknologi Fakultas Teknik* (pp. 126-131). Semarang: Fakultas Teknik Universitas Wahid Hasyim Semarang.
- Yang, X., Yu, Q., He, L., & Guo, T. (2013). The one-against-all partition based binary tree support vector machine algorithms for multi-class classification. *Neurocomputing*, 1-7.
- Yasin, S. e. (2021). Neural Networks based approaches for Major Depressive Disorder and Bipolar Disorder Diagnosis using EEG signals: A review. *Computer Methods and Programs in Biomedicine*, 1-29.
- Yi, B.-K., & Faloutsos, C. (2000). Fast Time Sequence Indexing for Arbitrary Lp Norms. *Proceedings of the 26th International Conference on Very Large Databases* (pp. 297-306). Cairo, Egypt: ResearchGate.
- You, S. (2021). Classification of Relaxation and Concentration Mental States with EEG. *Information*, 1-16.
- Zhang, D. e. (2010). Time Series Classification Using Support Vector Machine with Gaussian Elastic Metric Kernel. *2010 International Conference on Pattern Recognition* (pp. 29-32). Istanbul: IEEE.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1. Surat Keterangan Legalitas Data



**KOMITE ETIK PENELITIAN
RUMAH SAKIT UNIVERSITAS AIRLANGGA**

KETERANGAN LOLOS KAJI LAIK ETIK

ETHICAL APPROVAL

Nomor : 181/KEP/2023

Komite Etik Penelitian Rumah Sakit Universitas Airlangga dalam upaya melindungi hak asasi dan kesejahteraan subjek penelitian, telah mengkaji dengan teliti protokol penelitian yang berjudul:

The Research Ethics Committee of Rumah Sakit Universitas Airlangga with regards of the protection of human rights and welfare of research subjects, has carefully reviewed the research protocol entitled:

“Pengembangan *Support Vector Machines* Berbasis *Smoothing Function* untuk Klasifikasi Tingkat Depresi Menggunakan Sinyal *Elektroensefalogram (EEG)*”

Nomor Protokol : NA-02-23259
Protocol Number

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

Anggota Peneliti : 1. Dr. Diah P. Wulandari, S.T., M.Sc.
Research Member 2. Shofi Andari, Ph.D.
3. Yoyok Setyo Hadiwidodo, S.T., M.T., Ph.D.
4. Dr. dr. Wardah R. Ismalyah, Sp.S.
5. Dr. dr. Margarita M. Maramis, Sp.KJ(K)
6. Prof. Dr. Jasni Muhamad Zain

Nama Institusi : Fakultas Sains dan Analitika Data Institut Teknologi Sepuluh Nopember
Name of the Institution

Lokasi Penelitian : Rumah Sakit Universitas Airlangga
Site

Tanggal Persetujuan : 30 November 2023
Date of Approval (valid for one year beginning from the date of approval)

Dokumen Disetujui : Protokol Penelitian, Version 1.0 tanggal 13 November 2023
Document Approved

dan telah menyetujui protokol berikut dokumen terlampir.
And approves the above mentioned protocol including the attached document.

Ditetapkan di : Surabaya
Specified in

Ketua
Chair

Prof. Dr. Nancy Margarita Rehatta, dr., Sp.An., KMN., KNA
NIP 19501003 197703 2 001

- ❖ Peneliti berkewajiban:
1. Menjaga kerahasiaan identitas subjek penelitian;
 2. Memberitahukan status penelitian apabila:
 - a. Setelah masa berlakunya keterangan lolos kaji laik etik, penelitian masih belum selesai, dalam hal ini *ethical approval* harus diperpanjang;
 - b. Penelitian berhenti di tengah jalan;
 - c. Penyimpangan dari protokol penelitian yang diajukan.
 3. Melaporkan kejadian serius yang tidak diinginkan (*serious adverse events*);
 4. Peneliti tidak boleh melakukan tindakan apapun pada subjek sebelum protokol penelitian mendapat lolos kaji laik etik dan sebelum memperoleh *informed consent* dari subjek penelitian;
 5. Menyampaikan laporan akhir, bila penelitian sudah selesai;
 6. Mencantumkan nomor protokol pada setiap komunikasi dengan KEP RS UNAIR

Semua prosedur persetujuan dilakukan sesuai dengan standar ICH-GCP dan pedoman CIOMS-WHO 2016.
All procedures of Ethical Approval are performed in accordance with ICH-GCP standard and CIOMS – WHO Guidelines 2016.

Lampiran 2. Data Sinyal Perekaman EEG

No.	Netral Orang ke-1	...	Netral Orang ke- 10	Konsentrasi Orang ke- 1	...	Konsentrasi Orang ke- 10	Rileks Orang ke- 1	...	Rileks Orang ke- 10
1	-14,658	∴	17,052	-6,450	∴	-2,857	-2,141	∴	13,578
2	-15,477	∴	16,480	-7,561	∴	-3,112	-3,116	∴	12,724
3	-16,576	∴	15,230	-8,575	∴	-4,226	-4,689	∴	11,174
4	-19,071	∴	13,731	-8,913	∴	-5,793	-7,022	∴	9,267
5	-22,854	∴	12,447	-8,205	∴	-7,253	-9,985	∴	7,485
6	-26,409	∴	11,666	-6,489	∴	-8,317	-13,222	∴	6,037
7	-27,878	∴	11,411	-4,162	∴	-9,075	-16,114	∴	4,707
8	-26,513	∴	11,411	-1,907	∴	-9,889	-18,025	∴	2,993
9	-23,270	∴	11,287	-0,347	∴	-11,055	-18,668	∴	0,532
10	-20,014	∴	10,744	0,212	∴	-12,622	-18,343	∴	-2,569
11	-18,083	∴	9,720	-0,042	∴	-14,347	-17,850	∴	-5,697
12	-17,343	∴	8,430	-0,542	∴	-15,852	-17,973	∴	-8,147
13	-16,569	∴	7,298	-0,640	∴	-16,825	-18,876	∴	-9,539
14	-14,697	∴	6,687	-0,074	∴	-17,148	-19,916	∴	-10,042
15	-11,766	∴	6,665	0,849	∴	-16,905	-20,059	∴	-10,161
16	-8,855	∴	6,885	1,440	∴	-16,311	-18,649	∴	-10,314
17	-7,028	∴	6,693	1,115	∴	-15,570	-15,984	∴	-10,506
18	-6,417	∴	5,522	-0,178	∴	-14,727	-13,151	∴	-10,455
19	-6,144	∴	3,259	-1,907	∴	-13,606	-11,272	∴	-9,901
20	-5,078	∴	0,373	-3,343	∴	-11,903	-10,746	∴	-8,928
21	-2,784	∴	-2,354	-4,091	∴	-9,414	-11,006	∴	-7,926
22	0,238	∴	-4,311	-4,318	∴	-6,291	-11,051	∴	-7,281
23	3,013	∴	-5,414	-4,552	∴	-3,066	-10,219	∴	-7,061
24	4,904	∴	-6,093	-5,221	∴	-0,441	-8,640	∴	-6,970
25	5,990	∴	-6,851	-6,326	∴	1,126	-7,035	∴	-6,631
26	6,900	∴	-7,870	-7,496	∴	1,646	-6,099	∴	-5,901
27	8,258	∴	-8,871	-8,328	∴	1,544	-6,079	∴	-4,973
28	10,273	∴	-9,375	-8,569	∴	1,318	-6,632	∴	-4,181
29	12,658	∴	-9,160	-8,172	∴	1,199	-7,171	∴	-3,706
30	14,926	∴	-8,424	-7,165	∴	1,126	-7,269	∴	-3,462
31	16,726	∴	-7,694	-5,579	∴	0,928	-6,781	∴	-3,202
32	18,020	∴	-7,355	-3,519	∴	0,600	-5,787	∴	-2,704
∴	∴	∴	∴	∴	∴	∴	∴	∴	∴
23995	4,787	∴	-0,441	-26,818	∴	-1,341	26,040	∴	-11,768
23996	4,410	∴	-0,984	-28,008	∴	-4,402	26,358	∴	-11,140
23997	4,176	∴	-2,059	-29,061	∴	-6,738	26,625	∴	-9,884
23998	4,092	∴	-3,332	-30,036	∴	-7,853	26,800	∴	-8,232
23999	4,092	∴	-4,305	-30,660	∴	-7,638	26,670	∴	-6,399
24000	4,170	∴	-4,577	-30,523	∴	-6,387	25,923	∴	-4,526
24001	4,391	∴	-3,972	-29,405	∴	-4,588	24,279	∴	-2,676

Lampiran 3. Data Sinyal Perekaman EEG

minimu m_delta	minimu m_theta	...	kurtosis _delta	kurtosis _theta	kurtosis _alpha	...	entropy _alpha	entropy _beta	entropy_ gamma	kel as
-2,80E-05	-1,32E-05	∴	0,791	1,399	0,187	∴	1,949	1,616	2,044	1
-2,80E-05	-1,32E-05	∴	1,713	1,477	0,628	∴	2,049	1,660	2,077	1
-1,90E-05	-9,52E-06	∴	1,479	2,025	0,435	∴	2,056	1,702	2,062	1
-1,46E-05	-3,54E-06	∴	1,893	-0,517	-0,563	∴	2,186	1,878	2,154	1
-1,72E-05	-1,12E-05	∴	1,983	3,340	-0,096	∴	2,018	1,781	2,156	1
-1,72E-05	-1,12E-05	∴	1,593	1,564	-0,184	∴	2,054	1,694	2,119	1
-1,40E-05	-6,62E-06	∴	1,677	0,220	-0,058	∴	2,118	1,766	2,107	1
-1,52E-05	-6,62E-06	∴	1,177	0,029	0,138	∴	2,041	1,827	2,065	1
-1,52E-05	-6,47E-06	∴	1,334	0,016	0,109	∴	2,144	1,875	2,122	1
-1,36E-05	-6,94E-06	∴	1,308	0,062	0,130	∴	2,159	1,765	2,126	1
-2,49E-05	-6,94E-06	∴	1,738	-0,154	-0,419	∴	1,962	1,652	2,070	1
-2,49E-05	-6,42E-06	∴	1,517	0,277	-0,241	∴	2,083	1,800	2,102	1
-1,70E-05	-7,14E-06	∴	0,511	0,529	0,500	∴	2,153	2,037	2,027	1
-1,54E-05	-7,14E-06	∴	0,588	0,176	0,156	∴	2,117	1,940	2,032	1
-1,54E-05	-5,51E-06	∴	0,260	-0,193	-0,265	∴	2,118	1,827	2,121	1
-1,25E-05	-5,30E-06	∴	0,049	0,587	-0,191	∴	2,155	1,883	2,144	1
-1,07E-05	-5,46E-06	∴	0,378	0,688	-0,124	∴	2,141	2,068	2,093	1
-1,33E-05	-5,46E-06	∴	0,299	0,035	0,555	∴	2,139	2,062	2,042	1
-2,88E-05	-1,03E-05	∴	2,581	0,394	-0,338	∴	2,130	1,824	2,063	1
-2,88E-05	-1,50E-05	∴	2,061	2,069	0,245	∴	2,145	1,882	2,105	1
-2,02E-05	-1,50E-05	∴	3,997	7,703	0,876	∴	2,133	1,920	2,121	1
-1,85E-05	-8,59E-06	∴	1,078	1,282	-0,199	∴	2,095	1,779	2,078	1
-1,89E-05	-6,48E-06	∴	0,436	-0,003	-0,412	∴	1,979	1,818	2,059	1
-1,89E-05	-6,48E-06	∴	1,302	0,473	-0,387	∴	2,194	2,101	1,965	1
-9,51E-06	-4,31E-06	∴	-0,464	-0,199	-0,149	∴	2,125	2,145	1,810	1
-2,34E-05	-6,93E-06	∴	2,442	0,577	0,225	∴	2,006	1,691	2,065	1
-2,34E-05	-6,93E-06	∴	2,462	0,754	-0,129	∴	2,114	1,855	2,061	1
-1,59E-05	-5,53E-06	∴	2,484	0,221	-0,488	∴	1,932	2,152	1,739	1
-2,85E-05	-1,16E-05	∴	2,589	3,869	1,421	∴	2,077	1,702	2,103	1
-3,58E-05	-1,92E-05	∴	0,702	1,427	1,233	∴	1,455	1,278	1,632	1
-3,58E-05	-1,92E-05	∴	1,591	2,035	2,099	∴	1,373	1,202	1,556	1
-1,45E-05	-6,64E-06	∴	0,062	-0,019	-0,267	∴	2,093	1,868	2,129	1
-1,19E-05	-5,35E-06	∴	0,687	0,186	-0,063	∴	2,182	2,051	2,091	1
-1,85E-05	-5,54E-06	∴	2,126	0,104	0,079	∴	2,126	2,013	2,110	1
-1,85E-05	-5,54E-06	∴	1,338	-0,250	-0,342	∴	2,103	1,792	2,095	1
∴	∴	∴	∴	∴	∴	∴	∴	∴	∴	∴
-4,91E-05	-1,48E-05	∴	9,463	-0,092	0,062	∴	1,990	1,916	1,959	3
-4,91E-05	-1,48E-05	∴	9,919	0,302	0,586	∴	1,986	1,871	2,037	3
-2,12E-05	-8,62E-06	∴	1,016	-0,300	-0,188	∴	1,945	1,700	2,072	3
-2,47E-05	-9,88E-06	∴	0,244	0,032	-0,048	∴	2,074	1,946	2,013	3
-2,47E-05	-9,88E-06	∴	0,077	-0,597	0,056	∴	2,126	2,072	1,904	3

Lampiran 4 Syntax Preprocessing Data

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
!pip install mne
!pip install mne_features
import mne
from sklearn.model_selection import StratifiedKFold
import os
!pip install tslearn
import tslearn
from tslearn.utils import to_time_series
from tslearn.svm import TimeSeriesSVC
from tslearn.generators import random_walk_blobs
!pip install pyedflib
import pyedflib
from scipy import stats
import mne_features
!pip install scikit-plot
import scikitplot as skplt
import seaborn as sns

#input dataset
from glob import glob
file_path=glob("C:/Users/Students/Documents/Data Nuris/Data Tesis
EDF/Modified/*.csv")
print(len(file_path))
#Split data
Netral_path=[i for i in file_path if 'Netral' in i.split("_")[0]]
Konsentrasi_path=[i for i in file_path if 'Rileks' in i.split("_")[0]]
Rileks_path=[i for i in file_path if 'Konsentrasi' in i.split("_")[0]]
print(len(Netral_path),len(Rileks_path),len(Konsentrasi_path))
data = pd.read_csv('C:/Users/Students/Documents/Data Nuris/Data Tesis
EDF/Modified/Netral_1_Modified.csv', header=0)
data

#input dataset
from glob import glob
file_path=glob("C:/Users/Students/Documents/Data Nuris/Data Tesis
EDF/Modified/edf/*.edf")
print(len(file_path))
#Split data sesuai kondisi mental
Netral_path=[i for i in file_path if 'Netral' in
i.split("C:/Users/Students/Documents/Data Nuris/Data Tesis
EDF/Modified/edf/*.edf") [0]]
```

Lampiran 4 Syntax Preprocessing Data (Lanjutan)

```
Rileks_path=[i for i in file_path if 'Rileks' in
i.split("C:/Users/Students/Documents/Data Nuris/Data Tesis
EDF/Modified/edf/*.edf")][0]
Konsentrasi_path=[i for i in file_path if 'Konsentrasi' in
i.split("C:/Users/Students/Documents/Data Nuris/Data Tesis
EDF/Modified/edf/*.edf")][0]
print(len(Neutral_path),len(Rileks_path),len(Konsentrasi_path))

#contoh visualisasi grafik eeg dan informasi terkait perekaman eeg
raw = mne.io.read_raw_edf(Neutral_path[1], preload = True)
raw.info['ch_names']
#contoh visualisasi grafik eeg dan informasi terkait perekaman eeg
raw = mne.io.read_raw_edf(Neutral_path[0], preload = True)
channel_name = ['EEG O2-R'] #sesuai dengan nama di proses sebelumnya
raw_channel = raw.copy().pick_channels(channel_name)
raw_channel.plot(start=0, duration=4)
raw.info
eeg_picks = mne.pick_types(raw.info, eeg=True)
delta_raw_fir = raw.copy().filter(l_freq = 0.1, h_freq = 3.5,
picks=eeg_picks)
theta_raw_fir = raw.copy().filter(l_freq = 4, h_freq = 7.5,
picks=eeg_picks)
alpha_raw_fir = raw.copy().filter(l_freq = 8, h_freq = 13,
picks=eeg_picks)
beta_raw_fir = raw.copy().filter(l_freq = 13.5, h_freq = 30,
picks=eeg_picks)
gamma_raw_fir = raw.copy().filter(l_freq = 30.5, h_freq = 100,
picks=eeg_picks)
for title, data in zip(['Un', 'Delta FIR ', 'Theta FIR ', 'Alpha FIR
', 'Beta FIR ', 'Gamma FIR '], [raw, delta_raw_fir, theta_raw_fir,
alpha_raw_fir, beta_raw_fir, gamma_raw_fir]):
    fig = data.compute_psd().plot(average=True)
    fig.subplots_adjust(top=0.85)
    fig.suptitle('{}filtered'.format(title), size='xx-large',
weight='bold')
# Definisikan Rentang frekuensi
lowc = [0.1, 4, 8, 13.5, 30.5] #batas bawah
highc = [3.5, 7.5, 13, 30, 100] #batas atas

def data_input(path):
    raw = mne.io.read_raw_edf(path, preload=True)

    data = []
    # Loop through each passband
    for i in range(5):
        # Filter raw data
```

Lampiran 4 Syntax Preprocessing *Data* (Lanjutan)

```
raw_filtered = raw.filter(l_freq=lowc[i], h_freq=highc[i],
fir_window='hamming')

# Epoch data
epoch = mne.make_fixed_length_epochs(raw_filtered,
duration=10, overlap=5)
epoch.plot_image(picks=['EEG O2-R'])
#epoch.plot_image(picks=['AF8'])
#epoch.plot_image(picks=['TP9'])
#epoch.plot_image(picks=['TP10'])

# Jadi array
array = epoch.get_data()
data.append(array)

data = np.concatenate(data, axis=1)
return data

# Mengambil data untuk setiap jenis data
Netral_array = [data_input(i) for i in Netral_path]
Rileks_array = [data_input(i) for i in Rileks_path]
Konsentrasi_array = [data_input(i) for i in Konsentrasi_path]
Netral_label= [len(i)*[1] for i in Netral_array]
Rileks_label= [len(i)*[2] for i in Rileks_array]
Konsentrasi_label= [len(i)*[3] for i in Konsentrasi_array]
len(Netral_label),len(Rileks_label), len(Konsentrasi_label)

Data_list=Netral_array+Rileks_array+Konsentrasi_array
Label_list=Netral_label+Rileks_label+Konsentrasi_label
group_list=[[i]*len(j) for i,j in enumerate(Data_list)]

Data_array=np.vstack(Data_list)
label_array=np.hstack(Label_list)
Group_array=np.hstack(group_list)
print(Data_array.shape, label_array.shape, Group_array.shape)

# Mengubah menjadi dua dimensi dengan menggunakan reshape()
Data_array_2d = Data_array.reshape(-1, Data_array.shape[-1])
print(Data_array_2d.shape)

# Mendapatkan label sesuai dengan Data_array_2d
label_array_2d = np.repeat(label_array, Data_array_2d.shape[0] //
len(label_array))
```

Lampiran 5 Syntax Ekstraksi Fitur

```
from scipy import stats
import mne_features

def minimum(x):
    return np.min(x, axis=-1)
def median(x):
    return np.median(x, axis=-1)
def maksimum(x):
    return np.max(x, axis=-1)
def variance(x):
    return np.var(x, axis=-1)
def skewness(x):
    return stats.skew(x, axis = -1)
def kurtosis(x):
    return stats.kurtosis(x, axis = -1)
def energy(x):
    return np.sum(x**2, axis=-1)
def entropy(x):
    return mne_features.univariate.compute_spect_entropy(sfreq=500,
data=x)
def concatenate_features(x):
    return np.concatenate((minimum(x),median(x),
maksimum(x),variance(x),skewness(x), kurtosis(x), energy(x),
entropy(x)), axis=-1)

fitur= []
for i in Data_array:
    fitur.append(concatenate_features (i))

fitur_array = np.array(fitur)

#Penggabungan fitur array dengan label kelas
label_array1 = label_array.reshape(2850, 1)
combined_array = np.concatenate((fitur_array, label_array1), axis=1)

# Cetak bentuk hasil gabungan
print(combined_array.shape)

df_fitur = pd.DataFrame(combined_array)

# Definisikan daftar nama fitur
fitur = ['minimum', 'median', 'maksimum', 'variance', 'skewness',
'kurtosis', 'energy', 'entropy']

# Definisikan daftar nama sub-band
sub_band = ['delta', 'theta', 'alpha', 'beta', 'gamma']
```

Lampiran 5 Syntax Ekstraksi Fitur (Lanjutan)

```
# Definisikan daftar nama channel EEG
channel = ['O2']

# Inisialisasi list nama kolom
nama_kolom = []

# Looping untuk menggabungkan nama kolom
for f in fitur:
    for sb in sub_band:
        for ch in channel:
            nama_kolom.append(f + '_' + sb + '_' + ch)

nama_kolom.append('kelas')

# Inisialisasi list fitur
fitur = []

# Loop melalui Data_array dan tambahkan fitur hasil concatenate ke
dalam list fitur
for i in Data_array:
    fitur.append(concatenate_features(i))

# Konversi list fitur menjadi numpy array
fitur_array = np.array(fitur)

#Gabungkan fitur array sama group dan label
label_array1 = label_array.reshape(2850, 1)
combined_array = np.concatenate((fitur_array, label_array1), axis=1)

# Buat DataFrame dari fitur_array dengan nama kolom
df_fitur = pd.DataFrame(combined_array, columns=nama_kolom)

# Simpan DataFrame ke dalam file CSV
df_fitur.to_csv('fitur.csv', index=False)

# Menggabungkan kolom secara vertikal menjadi satu kolom
df_gabungan = pd.melt(df_fitur, value_vars=['minimum_delta_O2',
'minimum_theta_O2', 'minimum_alpha_O2',
'minimum_beta_O2', 'minimum_gamma_O2'], var_name='Fitur',
value_name='Ekstraksi Fitur')

# Menampilkan nilai minimum dan maksimum kolom Gabungan
nilai_min = df_gabungan['Ekstraksi Fitur'].min()
nilai_max = df_gabungan['Ekstraksi Fitur'].max()
```

Lampiran 5 Syntax Ekstraksi Fitur (Lanjutan)

```
print("Nilai Minimum:", nilai_min)
print("Nilai Maksimum:", nilai_max)
df_gabungan
```

Lampiran 6 Syntax Multiclass SVM

```
from sklearn import svm
from sklearn.multiclass import OneVsRestClassifier
from sklearn.multiclass import OneVsOneClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve, auc
from sklearn.utils.multiclass import type_of_target
from sklearn.preprocessing import label_binarize
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer, roc_auc_score
#Pembagian Data Training dan Testing
import numpy as np
selected_rows_train = np.concatenate([np.arange(0, 665),
np.arange(950, 1615), np.arange(1900, 2565)])
train_df_fitur = df_fitur.iloc[selected_rows_train, :]

selected_rows_test = np.concatenate([np.arange(665, 950),
np.arange(1615, 1900), np.arange(2565, 2850)])
test_df_fitur = df_fitur.iloc[selected_rows_test, :]
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from scipy.spatial.distance import cdist
# Example function to compute Time Warp Edit Distance (TWED)
def twed_distance(x, y, nu=0, lambda_=1e-5):
    n, m = len(x), len(y)
    D = np.zeros((n + 1, m + 1))
    D[:, 0] = np.inf
    D[0, :] = np.inf
    D[0, 0] = 0
```

Lampiran 6 Syntax Multiclass SVM (Lanjutan)

```
    for i in range(1, n + 1):
        for j in range(1, m + 1):
            cost = (x[i - 1] - y[j - 1]) ** 2
            D[i, j] = min(D[i - 1, j] + lambda_,
                          D[i, j - 1] + lambda_,
                          D[i - 1, j - 1] + cost + nu * (i + j))

    return D[n, m]

# Define the GTWED kernel function
def gtwed_kernel(X, Y, sigma=1):
    twed_dist_matrix = cdist(X, Y, metric=twed_distance)
    gtwed_k = np.exp(-twed_dist_matrix ** 2 / (2 * sigma ** 2))
    return gtwed_k

# Print label counts
print(Y.value_counts())
x_train = train_df_fitur
x_test = test_df_fitur

train_y_1 = 7*95*[1]
train_y_2 = 7*95*[2]
train_y_3 = 7*95*[3]

y_train = train_y_1 + train_y_2 + train_y_3

test_y_1 = 3*95*[1]
test_y_2 = 3*95*[2]
test_y_3 = 3*95*[3]

y_test = test_y_1 + test_y_2 + test_y_3
# Normalize the data
scaler = StandardScaler()
x_train_normalized = scaler.fit_transform(x_train)
x_test_normalized = scaler.transform(x_test)
# Compute the GTWED kernel matrix for training and test sets
K_train = gtwed_kernel(x_train_normalized, x_train_normalized)
K_test = gtwed_kernel(x_test_normalized, x_train_normalized)
# Train the SVM with the custom kernel
svm = SVC(kernel='precomputed')
svm.fit(K_train, y_train)

## Pendekatan OAA ##
import numpy
import matplotlib.pyplot as plt

from tslearn.datasets import CachedDatasets
from tslearn.preprocessing import TimeSeriesScalerMinMax
```


Lampiran 6 Syntax Multiclass SVM (Lanjutan)

```
from tslearn.svm import TimeSeriesSVC
import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import make_blobs
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
!pip install --upgrade scikit-learn
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from mlxtend.plotting import plot_decision_regions

# Create the SVM
svm_1 = SVC(kernel='precomputed', probability=True)
# Make it an OvR classifier
ovr_classifier_gtwed = OneVsRestClassifier(svm_1)
# Fit the data to the OvR classifier
ovr_classifier_gtwed = ovr_classifier_gtwed.fit(K_train, y_train)
# Compute and print the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
            xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score,
roc_auc_score

# Compute confusion matrix
cm_gtwed = confusion_matrix(y_test,
ovr_classifier_gtwed.predict(K_test))
```

Lampiran 6 Syntax Multiclass SVM (Lanjutan)

```
# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm_gtwed, annot=True, fmt="d", cmap="Blues")
plt.title('Confusion matrix for OvR classifier GTWED')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import confusion_matrix, accuracy_score,
roc_auc_score

# Compute confusion matrix
cm_gtwed_1 = confusion_matrix(y_train,
ovr_classifier_gtwed.predict(K_train))

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm_gtwed_1, annot=True, fmt="d", cmap="Blues")
plt.title('Confusion matrix data training for OvR classifier GTWED')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
from joblib import dump
svm_1.fit(K_train, y_train)
dump(svm_1, "Model OVR.joblib")
predict_train = svm_1.predict(K_train)
np.save('prediksi_train.npy', predict_train)
predict_test = svm_1.predict(K_test)
np.save('prediksi_test.npy', predict_test)
#ROC
##Train
y_proba = svm_1.predict_proba(K_train)
skplt.metrics.plot_roc(y_train, y_proba)

plt.savefig('ROC_train_OAA.png')
plt.show()
# ROC
##Test
y_proba =svm_1.predict_proba(K_test)
skplt.metrics.plot_roc(y_test, y_proba)
plt.savefig('ROC_test_OAA.png')
plt.show()
## Pendekatan OAO ##
```

Lampiran 6 Syntax Multiclass SVM (Lanjutan)

```
# Create the SVM
svm_2 = SVC(kernel='precomputed', probability=True)
# Make it an OvR classifier
ovo_classifier_gtwed = OneVsOneClassifier(svm_2)
# Fit the data to the OvR classifier
ovo_classifier_gtwed = ovo_classifier_gtwed.fit(K_train, y_train)
# Predict and evaluate the SVM
y_pred_1 = ovo_classifier_gtwed.predict(K_test)
accuracy_1 = accuracy_score(y_test, y_pred_1)
print(f'Accuracy: {accuracy_1:.2f}')

# Compute and print the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
            xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score,
roc_auc_score

# Compute confusion matrix
cm_2 = confusion_matrix(y_train,
                        ovo_classifier_gtwed.predict(K_train))

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm_2, annot=True, fmt="d", cmap="Blues")
plt.title('Confusion matrix data training for OvO classifier GTWED')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
from sklearn.metrics import confusion_matrix
```

Lampiran 6 Syntax Multiclass SVM (Lanjutan)

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score,
roc_auc_score

# Compute confusion matrix
cm_3 = confusion_matrix(y_test, ovo_classifier_gtwed.predict(K_test))

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm_3, annot=True, fmt="d", cmap="Blues")
plt.title('Confusion matrix data testing for OvO classifier GTWED')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
from joblib import dump
svm_2.fit(K_train, y_train)
dump(svm_2, "Model OVO GTWED.joblib")
predict_train_ovo = svm_2.predict(K_train)
np.save('prediksi train ovo GTWED.npy', predict_train_ovo)
predict_test_ovo = svm_2.predict(K_test)
np.save('prediksi test ovo GTWED.npy', predict_test_ovo)
# ROC
##Test
y_proba_2_test =svm_2.predict_proba(K_test)
skplt.metrics.plot_roc(y_test, y_proba)
plt.savefig('ROC_test_OAO_GTWED.png')
plt.show()
```

Lampiran 7 Syntax Visualisasi Data

```
import numpy as np
import pandas as pd
from scipy import signal
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from scipy.signal import butter, lfilter
from google.colab import files
filenya=files.upload()
df1=pd.read_csv('fitur.csv',sep="," , decimal=".",encoding='latin-1')
df1.drop(['Unnamed: 0'], axis=1, inplace=True)
print(df1)
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
import pandas as pd

# Membaca data dari csv
df = pd.read_csv('fitur.csv', sep=",", decimal=".", encoding='latin-1')
df.drop(['Unnamed: 0'], axis=1, inplace=True)

# Membuat DataFrame kosong untuk menyimpan hasil perhitungan
summary_data = []

# Looping untuk setiap kolom
for kolom in df.columns:
    data = df[kolom]

    # Menghitung nilai maksimum, minimum, dan median
    nilai_maksimum = data.max()
    nilai_minimum = data.min()
    nilai_median = data.median()

    # Menambahkan hasil perhitungan ke dalam list
    summary_data.append({'Kolom': kolom, 'Nilai Maksimum':
nilai_maksimum, 'Nilai Minimum': nilai_minimum, 'Nilai Median':
nilai_median})

# Membuat DataFrame dari list
summary_df = pd.DataFrame(summary_data)

# Menyimpan DataFrame ke dalam file Excel
summary_df.to_excel('summary_file.xlsx', index=False)

#Hubungin ke gdrive

from google.colab import drive
drive.mount('/content/drive/')

# Menyimpan DataFrame ke dalam file CSV
summary_df.to_excel('/content/drive/MyDrive/DataTesis/Hasil/summary_fi
tur.xlsx', index=False)
!pip install seaborn
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep=",", decimal=".",encoding='latin-1')
selected_columns = ['minimum_delta_O2', 'minimum_theta_O2',
'minimum_alpha_O2', 'minimum_beta_O2', 'minimum_gamma_O2', 'kelas']
df_selected_min = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['minimum_delta_O2', 'minimum_theta_O2',
'minimum_alpha_O2', 'minimum_beta_O2', 'minimum_gamma_O2']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Minimum')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='lower right')
plt.show()

# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep=",", decimal=".",encoding='latin-1')
selected_columns = ['median_delta_O2', 'median_theta_O2',
'median_alpha_O2', 'median_beta_O2', 'median_gamma_O2', 'kelas']
df_selected_median = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['median_delta_O2', 'median_theta_O2',
'median_alpha_O2', 'median_beta_O2', 'median_gamma_O2']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Median')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='upper right')
plt.show()
# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep=",", decimal=".",encoding='latin-
1')
selected_columns = ['maksimum_delta_O2', 'maksimum_theta_O2',
'maksimum_alpha_O2', 'maksimum_beta_O2', 'maksimum_gamma_O2', 'kelas']
df_selected_maksimum = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['maksimum_delta_O2', 'maksimum_theta_O2',
'maksimum_alpha_O2', 'maksimum_beta_O2', 'maksimum_gamma_O2']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Maksimum')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='upper right')
plt.show()
# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep=",", decimal=".",encoding='latin-
1')
selected_columns = ['variance_delta_O2', 'variance_theta_O2',
'variance_alpha_O2', 'variance_beta_O2', 'variance_gamma_O2', 'kelas']
df_selected_maksimum = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['variance_delta_O2', 'variance_theta_O2',
'variance_alpha_O2', 'variance_beta_O2', 'variance_gamma_O2']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
var_name='Feature', value_name='Value')
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Varians')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='upper right')
plt.show()
# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep="," , decimal=".",encoding='latin-
1')
selected_columns = ['kurtosis_delta_02', 'kurtosis_theta_02',
'kurtosis_alpha_02', 'kurtosis_beta_02', 'kurtosis_gamma_02', 'kelas']
df_selected_maksimum = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['kurtosis_delta_02', 'kurtosis_theta_02',
'kurtosis_alpha_02', 'kurtosis_beta_02', 'kurtosis_gamma_02']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Kurtosis')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='upper right')
plt.show()
# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep="," , decimal=".",encoding='latin-
1')
selected_columns = ['energy_delta_02', 'energy_theta_02',
'energy_alpha_02', 'energy_beta_02', 'energy_gamma_02', 'kelas']
df_selected_maksimum = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['energy_delta_02', 'energy_theta_02',
'energy_alpha_02', 'energy_beta_02', 'energy_gamma_02']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
```


Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
        var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Energi')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='upper right')
plt.show()

# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep="," , decimal=".",encoding='latin-
1')
selected_columns = ['entropy_delta_02', 'entropy_theta_02',
'entropy_alpha_02', 'entropy_beta_02', 'entropy_gamma_02', 'kelas']
df_selected_maksimum = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['entropy_delta_02', 'entropy_theta_02',
'entropy_alpha_02', 'entropy_beta_02', 'entropy_gamma_02']

# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
palette='Set3')
plt.title('Boxplot Fitur Entropi')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='lower right')
plt.show()

# Load data from CSV
df_min=pd.read_csv('fitur.csv',sep="," , decimal=".",encoding='latin-
1')
selected_columns = ['skewness_delta_02', 'skewness_theta_02',
'skewness_alpha_02', 'skewness_beta_02', 'skewness_gamma_02', 'kelas']
df_selected_maksimum = df[selected_columns]

# List of feature columns for boxplot
feature_columns = ['skewness_delta_02', 'skewness_theta_02',
'skewness_alpha_02', 'skewness_beta_02', 'skewness_gamma_02']
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
# Melt the DataFrame to long format
df_melted = pd.melt(df, id_vars=['kelas'], value_vars=feature_columns,
                    var_name='Feature', value_name='Value')

# Create a boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(x='kelas', y='Value', hue='Feature', data=df_melted,
            palette='Set3')
plt.title('Boxplot Fitur Skewness')
plt.xlabel('Kelas')
plt.ylabel('Beda Tegangan')
plt.legend(title='Feature', loc='lower right')
plt.show()

## PSD Plot ##
!pip install mne
import mne
!pip install --upgrade mne
#Sambungkan
from google.colab import drive
drive.mount('/content/drive/')
import numpy as np
import pandas as pd

#Netral
data_netral =
pd.read_csv('/content/drive/MyDrive/DataTesis/New/Modified/Netral_1_Mo
dified.csv', header=0)
data_netral
#NETRAL
data_array = data_netral.to_numpy()

# Create MNE info object
n_channels = data_array.shape[1]
ch_names = ['ch{}'.format(i) for i in range(n_channels)]
ch_types = ['eeg'] * n_channels
sfreq = 500 # sampling rate
info = mne.create_info(ch_names=ch_names, sfreq=sfreq,
                       ch_types=ch_types)

# Create MNE RawArray object
raw_n = mne.io.RawArray(data_array.T, info)
# Create MNE RawArray object
raw_n = mne.io.RawArray(data_array.T, info)

print (raw_n)
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
#Plot PSD
raw_n.compute_psd().plot(average=True)
## Konsentrasi ##
data_konsentrasi =
pd.read_csv('/content/drive/MyDrive/DataTesis/New/Modified/R_1_Modified.csv', header=0)
data_konsentrasi
data_array1 = data_konsentrasi.to_numpy()

# Create MNE info object
n_channels = data_array1.shape[1]
ch_names = ['ch{}'.format(i) for i in range(n_channels)]
ch_types = ['eeg'] * n_channels
sfreq = 500 # sampling rate
info = mne.create_info(ch_names=ch_names, sfreq=sfreq,
ch_types=ch_types)

# Create MNE RawArray object
raw_k = mne.io.RawArray(data_array1.T, info)
# Create MNE RawArray object
raw_k = mne.io.RawArray(data_array1.T, info)

print (raw_k)

#Plot PSD
raw_k.compute_psd().plot(average=True)
## Rileks ##
data_rileks =
pd.read_csv('/content/drive/MyDrive/DataTesis/New/Modified/K_1_Modified.csv', header=0)
data_rileks
data_array2 = data_rileks.to_numpy()

# Create MNE info object
n_channels = data_array2.shape[1]
ch_names = ['ch{}'.format(i) for i in range(n_channels)]
ch_types = ['eeg'] * n_channels
sfreq = 500 # sampling rate
info = mne.create_info(ch_names=ch_names, sfreq=sfreq,
ch_types=ch_types)

# Create MNE RawArray object
raw_r = mne.io.RawArray(data_array2.T, info)
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
# Create MNE RawArray object
raw_r = mne.io.RawArray(data_array2.T, info)

print (raw_r)

#Plot PSD
raw_r.compute_psd().plot(average=True)
## Time Series Plot ##
from google.colab import files
filenya=files.upload()
import pandas as pd
df_ts=pd.read_excel('Data Orang ke -1 - Copy.xlsx')
print(df_ts)
from scipy import signal
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from scipy.signal import butter, lfilter
t = np.linspace(0, 40, 240001)
fig, axs = plt.subplots(nrows=4, ncols=1, figsize=(12.5, 8))

# plotting time series pertama pada Axes pertama
axs[0].plot(t, df_ts['Netral'])
axs[0].set_xlabel('Waktu')
axs[0].set_ylabel('Beda tegangan')
axs[0].set_title('Plot Sinyal Netral', fontsize=11)
axs[0].legend(loc='upper left')

# plotting time series kedua pada Axes kedua
axs[1].plot(t, df_ts['Rileks'])
axs[1].set_xlabel('Waktu')
axs[1].set_ylabel('Beda tegangan')
axs[1].set_title('Plot Sinyal Rileks', fontsize=11)
axs[1].legend(loc='upper left')

# plotting time series ke3 pada Axes ke3
axs[2].plot(t, df_ts['Konsentrasi'])
axs[2].set_xlabel('Waktu')
axs[2].set_ylabel('Beda tegangan')
axs[2].set_title('Plot Sinyal Konsentrasi', fontsize=11)
axs[2].legend(loc='upper left')

# plotting time series ke4 pada Axes ke4
axs[3].plot(t, df_ts['Netral'], label='Netral')
axs[3].plot(t, df_ts['Konsentrasi'], label='Rileks')
axs[3].plot(t, df_ts['Rileks'], label='Konsentrasi')
```

Lampiran 7 Syntax Visualisasi Data (Lanjutan)

```
axs[3].set_xlabel('Time')
axs[3].set_ylabel('Beda Tegangan')
axs[3].set_title('Plot Sinyal Gabungan', fontsize=11)
axs[3].legend(loc='upper right', fontsize=8)

# menampilkan plot
plt.tight_layout()
plt.savefig('TS_plot.png', dpi=1000)
plt.show()
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Membuat DataFrame baru yang menyimpan kondisi dan nilai tegangan
df_boxplot = pd.DataFrame({
    'Netral': df_ts['Netral'],
    'Rileks': df_ts['Rileks'],
    'Konsentrasi': df_ts['Konsentrasi']
})

# Mengubah DataFrame ke format long untuk seaborn
df_melted = df_boxplot.melt(var_name='Kondisi', value_name='Beda
Tegangan')

# Membuat boxplot dengan warna yang disesuaikan
plt.figure(figsize=(10, 6))
sns.boxplot(x='Kondisi', y='Beda Tegangan', data=df_melted,
palette={'Netral': 'blue', 'Konsentrasi': 'orange', 'Rileks':
'green'})
plt.title('Boxplot dari Kondisi Mental (Netral, Rileks, Konsentrasi)')
plt.xlabel('Kondisi')
plt.ylabel('Beda Tegangan')
plt.show()
```

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis bernama lengkap Nuris Farandhini Hidayat yang kerap disapa Nuris ini lahir di Gresik, 28 Desember 1998. Penulis adalah anak ke satu dari tiga bersaudara anak dari pasangan bapak Alm.Nurul Hidayat dan Lilik Isnawati. Penulis telah menyelesaikan pendidikan formal di TK Aisyiyah Busthanul Athfal lulus tahun 2004, SD Muhammadiyah 1 Gresik lulus pada tahun 2010, SMPN 2 Gresik lulus pada tahun 2013, SMA Muhammadiyah 1 Gresik lulus pada tahun 2016, Diploma III Statistika Bisnis ITS lulus tahun 2019, dan menjalankan perkuliahan di Departemen S1 Statistika ITS melalui program RPL di tahun 2020 dan sekarang masih menempuh program Magiester. Alamat asal penulis Kecamatan Kebomas, Kabupaten Gresik. Selama masa perkuliahan, penulis aktif dalam organisasi sebagai staff Departemen Hubungan Luar HIMADATA-ITS periode 2017/2018, Sekretaris Departemen Hubungan Luar HIMADATA-ITS periode 2018/2019. Selain itu, penulis aktif mengikuti berbagai kepanitiaan maupun kegiatan pelatihan. Segala kritik dan saran akan diterima penulis untuk perbaikan kedepannya. Untuk informasi dan komunikasi lebih lanjut dengan penulis dapat menghubungi melalui email nurisfarandhini@gmail.com.
Terimakasih

