

**RANCANG BANGUN KLASIFIKASI PENDETEKSIAN
JENIS KLIK PADA *POINTING DEVICE*
MENGUNAKAN ELECTROMYOGRAPH**

TUGAS AKHIR



DEVINA TRIxie

311510010

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG**

MALANG

2021

**CLICK-TYPE DETECTION CLASSIFICATION
DESIGN ON POINTING DEVICE USING
ELECTROMYOGRAPH**

ESSAY



DEVINA TRIxie

311510010

**INFORMATICS ENGINEERING STUDY PROGRAM
FACULTY OF SCIENCE AND TECHNOLOGY
UNIVERSITAS MA CHUNG
MALANG
2021**

**PERNYATAAN KEASLIAN
TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “RANCANG BANGUN KLASIFIKASI PENDETEKSIAN JENIS KLIK PADA POINTING DEVICE MENGGUNAKAN ELECTROMYOGRAPH” adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, Februari 2021

Devina Trixie
311510010

LEMBAR PENGESAHAN

RANCANG BANGUN KLASIFIKASI PENDETEKSI JENIS KLIK PADA
POINTING DEVICE MENGGUNAKAN ELECTROMYOGRAPH

Oleh:

DEVINA TRIxie

311510010

dari

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas
Akhir sebagai syarat kelulusan dan berhak
mendapatkan gelar Sarjana Komputer (S.Kom)

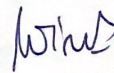
Dosen Pembimbing 1



Dr. Eng. Romy Budhi Widodo

NIP. 20070035

Dosen Pembimbing 2



Windra Swastika, Ph. D

NIP. 20070039

Dekan Fakultas Sains dan Teknologi



Kesatria Rega Priati, S.Si., M.Si.

NIP. 20120035

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas rahmat dan penyertaan-Nya sehingga tugas akhir dapat berjalan dengan baik. Laporan ini berisi hasil dari tugas akhir yang telah dilaksanakan di Universitas Ma Chung. Atas dukungan moral dan materil yang diberikan dalam pembuatan laporan ini, penulis ingin mengucapkan terima kasih kepada:

1. Orang tua dan keluarga yang telah mendukung penulis dalam menyelesaikan Tugas Akhir,
2. Bapak Hendry Setiawan, ST, M.Kom selaku Kepala Program Studi Teknik Informatika,
3. Bapak Dr.Eng. Romy Budhi Widodo selaku dosen pembimbing utama Tugas Akhir Universitas Ma Chung,
4. Windra Swastika, Ph.D selaku dosen pembimbing kedua Tugas Akhir Universitas Ma Chung,
5. Marcell, Daniel, Rendy, Ko Andra, Vanno, Sovi, Ekky, Natha, Yuan, Bima, dan teman-teman Teknik Informatika angkatan 2015 lainnya yang telah mendukung selama pengerjaan Tugas Akhir.

Laporan Tugas Akhir ini disusun berdasarkan materi yang diberikan yaitu “Rancang Bangun Klasifikasi Pendeteksian Jenis Klik Pada Pointing Device Menggunakan Electromyograph”. Tugas akhir ini adalah sebagai mata kuliah yang wajib ditempuh mahasiswa Teknik Informatika Universitas Ma Chung Malang. Diharapkan dengan penulisan laporan tugas akhir ini, dapat bermanfaat bagi seluruh pembaca.

Malang, Februari 2021

Devina Trixie

Rancang Bangun Klasifikasi Pendeteksian Jenis Klik Pada Pointing Device Menggunakan Electromyograph

Devina Trixie
311510010

Abstrak

Pengoperasian komputer mengharuskan manusia menggunakan beberapa bagian tubuhnya. Namun, terdapat beberapa kondisi dimana manusia tidak dapat mengoperasikan komputer dengan baik atau dengan posisi normal, contoh dari kondisi tersebut adalah korban kecelakaan dan penyandang disabilitas. Maka dari itu dibutuhkan sistem yang dapat membantu agar memudahkan orang-orang tersebut dalam mengoperasikan komputer. Dalam penelitian ini akan dibuat sebuah sistem yang dapat mengklasifikasikan jenis klik menggunakan sensor EMG, metode K-NN, dan metode SVM. Sensor EMG berguna untuk mengambil data berupa sinyal dari kontraksi otot manusia yang nantinya akan diklasifikasikan menjadi klik kiri dan klik kanan, sedangkan untuk metode K-NN dan SVM berguna untuk mengklasifikasikan jenis-jenis klik tersebut. Data dari sensor EMG akan dilatih menggunakan metode K-NN dan SVM menggunakan 54 data set pada masing-masing kelasnya yaitu kelas klik kiri dan klik kanan. Pada metode K-NN akan dilatih menggunakan $k = 3, 5, 7, 9$, dan 11 dan metode SVM menggunakan kernel linear, *Radial Basis Function* (RBF), polinomial, dan sigmoid. Setelah itu nilai akurasi dari kedua metode itu akan dibandingkan satu sama lainnya. Sistem telah berhasil mengembangkan metode klasifikasi yang dapat mengklasifikasikan jenis-jenis klik berdasarkan masukan dari sensor EMG menggunakan metode K-NN dengan hasil akurasi tertinggi adalah menggunakan $k = 3$ yaitu sebesar 81,81% dan metode SVM adalah menggunakan kernel polynomial yaitu sebesar 84,84%. Dengan membandingkan kedua metode didapatkan nilai akurasi tertinggi yaitu menggunakan metode SVM kernel polinomial. Penambahan dataset dan melakukan percobaan menggunakan metode lain sebagai perbandingan lebih lanjut dapat digunakan untuk meningkatkan akurasi sistem.

Kata kunci : Elektromyography (EMG), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Klasifikasi Jenis Klik pada Perangkat Penunjuk

Design of Click Type Detection Classification on Pointing Device Using Electromyograph

Devina Trixie
311510010

Abstract

Computer operation requires humans to use several parts of their bodies. However, there are several conditions where humans cannot operate computers properly or in a normal position, examples of these conditions are accident victims and people with disabilities. Therefore we need a system that can help make it easier for these people to operate the computer. In this research, a system that can classify the types of clicks using the EMG sensor, the K-NN method and the SVM method will be built. The EMG sensor is useful for retrieving data in the form of signals from human muscle contraction which will later be classified into left click and right click, while for the K-NN and SVM methods it is useful for classifying the types of clicks. Data from the EMG sensor will be trained using the K-NN and SVM methods using 54 data sets in each class, namely the left click and right click classes. The K-NN method will be trained using $k = 3, 5, 7, 9$, and 11 and the SVM method uses a linear kernel, Radial Basis Function (RBF), polynomials, and sigmoid. After that the accuracy values of the two methods will be compared with each other. The system has succeeded in developing a classification method that can classify the types of clicks based on input from the EMG sensor using the K-NN method with the highest accuracy result is using $k = 3$ which is equal to 81.81% and the SVM method is using the polynomial kernel which is 84.84% . By comparing the two methods, the highest accuracy value is obtained, namely using the SVM method of the polynomial kernel. Adding datasets and conducting experiments using other methods as further comparison can be used to increase system accuracy.

Keywords : Eletromyography (EMG), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Mouse Click Classification

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vii
Bab I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Rumusan Masalah	3
1.5 Tujuan	3
1.6 Manfaat	3
1.7 Sistematika Penulisan	3
Bab II Tinjauan Pustaka	5
2.1 Otot	5
2.2 Elektromiografi (EMG)	8
2.2.1 Biosignalplux	10
2.2.2 OpenSignals (r)evolution	14
2.3 Perangkat Penunjuk	14
2.4 Penyandang Disabilitas	15
2.5 Python	16
2.6 <i>Support Vector Machine</i> (SVM)	16
2.6.1 Linier	19
2.6.2 <i>Radial Basis Function</i> (RBF)	19
2.6.3 Sigmoid	19
2.6.4 Polinomial	20
2.7 <i>K-Nearest Neighbor</i> (K-NN)	20
2.8 <i>Confussion Matrix</i>	21
2.9 Ekstraksi Fitur	23
2.9.1 <i>Max EMG Value</i>	23
2.9.2 <i>Min EMG Value</i>	23

2.9.3	<i>Wave Length</i>	23
2.9.4	<i>Mean Absolute Deviation</i>	24
2.10	ISO 9241-411	24
2.11	Analisis Statistika	25
2.11.1	Analisis Kualitatif	25
2.11.2	Analisis Kuantitatif	26
2.12	Penelitian Terdahulu	27
Bab III	Analisis dan Perancangan Sistem	30
3.1	Tahapan Penelitian	30
3.2	Analisis Kebutuhan	31
3.3	Desain Sistem	32
3.3.1	Kontraksi Otot Lengan Atas Manusia	33
3.3.2	Pembacaan Data Oleh Sensor	34
3.3.3	Sensor EMG Biosignalsplux	35
3.3.4	Feature Extraction Data	35
3.3.5	Pembelajaran pada K-NN	36
3.3.6	Pembelajaran pada SVM	37
3.4	Implementasi Sistem	38
3.5	Rancangan Pengujian	39
Bab IV	Hasil dan Pembahasan	41
4.1	Hasil Pengumpulan Data	41
4.1.1	Pemasangan Perangkat Sensor Pada Responden	41
4.1.2	Proses Pengambilan Data Responden	42
4.1.3	Proses Pengelolaan <i>Dataset</i>	43
4.1.4	Proses <i>Splitting Data Train</i> dan <i>Data Test</i>	45
4.2	Hasil Rancangan Aplikasi	45
4.2.1	Antarmuka Halaman <i>Home</i>	45
4.2.2	Antarmuka Halaman <i>Accuracy</i>	47
4.2.2	Antarmuka Halaman <i>Information</i>	48
4.3	Hasil Pengujian Sistem	49
4.3.1	Hasil Pengujian K-NN	49
4.3.2	Hasil Pengujian SVM	51

4.4 Hasil Kuesioner Penggunaan Sistem	52
4.5 Diskusi Hasil	69
Bab V Penutup	71
5.1 Simpulan	71
5.2 Saran	71
DAFTAR PUSTAKA	72
LAMPIRAN	74

DAFTAR TABEL

Tabel 2.1 Spesifikasi Wireless 4-channel Hub	11
Tabel 2.2 Tabel <i>Confusion Matrix</i>	21
Tabel 2.3 Format Instrumentasi kuesioner <i>Borg Scale</i> ISO 9241-411	25
Tabel 2.4 Format Instrumentasi Penilaian Bebas ISO 9241-411	26
Tabel 3.1 Spesifikasi Komputer	32
Tabel 4.1 Data Responden	42
Tabel 4.2 Hasil Rekaman Kontraksi Salah Satu Responden	44
Tabel 4.3 Hasil Pengujian KNN dengan Parameter $k = 3$	49
Tabel 4.4 Hasil Pengujian KNN dengan Parameter $k = 5$	49
Tabel 4.5 Hasil Pengujian KNN dengan Parameter $k = 7$	50
Tabel 4.6 Hasil Pengujian KNN dengan Parameter $k = 9$	50
Tabel 4.7 Hasil Pengujian KNN dengan Parameter $k = 11$	50
Tabel 4.8 Hasil Pengujian SVM dengan Kernel Linear	51
Tabel 4.9 Hasil Pengujian SVM dengan Kernel Polinomial	51
Tabel 4.10 Hasil Pengujian SVM dengan Kernel RBF	51
Tabel 4.11 Hasil Pengujian SVM dengan Kernel Sigmoid	52
Tabel 4.12 Tabel Hasil Kuesioner Usaha pada Bagian Tangan (<i>Arm</i>)	53
Tabel 4.13 Tabel Hasil Kuesioner Usaha pada Bagian Pundak (<i>Shoulder</i>)	54
Tabel 4.14 Tabel Hasil Kuesioner Usaha pada Bagian Leher (<i>Neck</i>)	55
Tabel 4.15 Tabel Hasil Kuesioner Usaha yang Diperlukan untuk Membuat Alat Beroperasi (<i>Force Required for Actuation</i>)	56
Tabel 4.16 Tabel Hasil Kuesioner Kelancaran Saat Menjalankan Program (<i>Smoothness During Operation</i>)	57
Tabel 4.17 Tabel Hasil Kuesioner Usaha yang Diperlukan untuk Melakukan Saat Uji Coba (<i>Effort Required for Operation</i>)	58
Tabel 4.18 Tabel Hasil Kuesioner Akurasi (<i>Accuracy</i>)	59
Tabel 4.19 Tabel Hasil Kuesioner Kecepatan Sistem dalam Beroperasi (<i>Operation Speed</i>)	60
Tabel 4.20 Tabel Hasil Kuesioner Kenyamanan Pemakaian Alat (<i>General Comfort</i>)	61

Tabel 4.21 Tabel Hasil Kuesioner Tingkat Kesulitan Pemakaian Sensor (<i>Overall Operation of Input Device</i>)	62
Tabel 4.22 Tabel Hasil Kuesioner Kelelahan pada Jari Saat Menggunakan Alat (<i>Finger Fatigue</i>)	63
Tabel 4.23 Tabel Hasil Kuesioner Kelelahan pada Pergelangan Tangan Saat Menggunakan Alat (<i>Wrist Fatigue</i>)	64
Tabel 4.24 Tabel Hasil Kuesioner Kelelahan pada Tangan Saat Menggunakan Alat (<i>Arm Fatigue</i>)	65
Tabel 4.25 Tabel Hasil Kuesioner Kelelahan pada Pundak Saat Menggunakan Alat (<i>Shoulder Fatigue</i>)	66
Tabel 4.26 Tabel Hasil Kuesioner Kelelahan pada Leher Saat Menggunakan Alat (<i>Neck Fatigue</i>)	67
Tabel 4.27 Perbandingan Nilai Akurasi Metode KNN dan SVM	70

DAFTAR GAMBAR

Gambar 2.1 Struktur Jaringan Otot	7
Gambar 2.2 Jenis-jenis otot	7
Gambar 2.3 Kontraksi Otot Bisep dan Trisep	8
Gambar 2.4 <i>Intramuscular</i> EMG	9
Gambar 2.5 <i>Surface</i> EMG	9
Gambar 2.6 4- <i>Channel</i> Biosignalsplux Hub	12
Gambar 2.7 Biosignalsplux EMG	13
Gambar 2.8 Tampilan awal Opensignals (r)evolution	14
Gambar 2.9 Contoh <i>Hyperplane</i> pada SVM	17
Gambar 2.10 Pola <i>One-directional tapping test</i>	24
Gambar 2.11 Prototipe Perangkat Penunjuk Terdahulu	28
Gambar 3.1 Tahapan Perancangan Kecerdasan Buatan	30
Gambar 3.2 Alur Desain Kinerja Sistem Perangkat	33
Gambar 3.3 Grafik tampilan hasil pembacaan data oleh sensor EMG pada aplikasi OpenSignals(r)evolution	34
Gambar 3.4 Grafik hasil pembacaan sensor pada excel	35
Gambar 3.5 Ilustrasi pemasangan sensor EMG dan Hub	35
Gambar 3.6 Diagram alur K-NN	36
Gambar 3.7 Diagram alur SVM	37
Gambar 3.7 <i>Mockup</i> tampilan aplikasi pengujian	39
Gambar 4.1 Proses Akuisisi Data	41
Gambar 4.2 Sinyal Kontraksi Pengambilan Data	42
Gambar 4.3 Grafik Fluktuasi Pengambilan Data	43
Gambar 4.4 Tampilan Antarmuka Halaman <i>Home</i>	46
Gambar 4.5 Tampilan Halaman Ketika Status <i>Idle</i>	46
Gambar 4.6 Tampilan Ketika Status <i>Clicked</i> dan Result <i>Left Click</i>	47
Gambar 4.7 Tampilan Ketika Status <i>Clicked</i> dan Result <i>Right Click</i>	47
Gambar 4.8 Tampilan Antarmuka Halaman <i>Accuracy</i>	48
Gambar 4.9 Tampilan Antarmuka Halaman <i>Information</i>	48
Gambar 4.10 Pilihan Jawaban Responden terhadap Penilaian <i>Effort</i> pada <i>Arm</i>	53

Gambar 4.11 Pilihan Jawaban Responden terhadap Penilaian <i>Effort</i> pada <i>Shoulder</i>	54
Gambar 4.12 Pilihan Jawaban Responden terhadap Penilaian <i>Effort</i> pada <i>Neck</i>	55
Gambar 4.13 Rata-rata Kuesioner Borg Scale	56
Gambar 4.14 Pilihan Jawaban Responden terhadap Penilaian <i>force required for actuation</i>	57
Gambar 4.15 Pilihan Jawaban Responden terhadap Penilaian <i>Smoothness</i> <i>During Operation</i>	58
Gambar 4.16 Pilihan Jawaban Responden terhadap Penilaian <i>Effort</i> <i>Required for Operation</i>	59
Gambar 4.17 Pilihan Jawaban Responden terhadap Penilaian <i>Accuracy</i>	60
Gambar 4.18 Pilihan Jawaban Responden terhadap Penilaian <i>Operation Speed</i>	61
Gambar 4.19 Pilihan Jawaban Responden terhadap Penilaian <i>General Comfort</i>	62
Gambar 4.20 Pilihan Jawaban Responden terhadap Penilaian <i>Overall Operation</i> <i>of Input Device</i>	63
Gambar 4.21 Pilihan Jawaban Responden terhadap Penilaian <i>Finger Fatigue</i>	64
Gambar 4.22 Pilihan Jawaban Responden terhadap Penilaian <i>Wrist Fatigue</i>	65
Gambar 4.23 Pilihan Jawaban Responden terhadap Penilaian <i>Arm Fatigue</i>	66
Gambar 4.24 Pilihan Jawaban Responden terhadap Penilaian <i>Shoulder Fatigue</i>	67
Gambar 4.25 Pilihan Jawaban Responden terhadap Penilaian <i>Neck Fatigue</i>	68
Gambar 4.25 Rata-rata Hasil Kuesioner Instrumentasi Penilaian Bebas	68

Bab I

Pendahuluan

1.1 Latar Belakang

Kecelakaan lalu lintas merupakan hal yang tidak dapat diperkirakan dan sangat dihindari oleh semua orang. Kejadian ini melibatkan kendaraan dengan atau tanpa pengendara lain yang dapat mengakibatkan korban manusia dari hanya luka ringan, luka berat bahkan sampai kematian. Terdapat empat faktor utama terjadinya kecelakaan lalu lintas, yaitu faktor kelalaian pengguna jalan, faktor kendaraan yang bermasalah, faktor jalan seperti jalan berlubang yang membahayakan pengguna jalan terutama pengguna sepeda motor, dan faktor kondisi lingkungan, contohnya hujan deras yang menyebabkan jalan menjadi licin dan jarak pandang menjadi lebih dekat. Menurut berita dari Kompas, Kepolisian Negara Republik Indonesia mencatat jumlah kecelakaan lalu lintas pada 2019 meningkat sebesar 3% bila dibandingkan dengan tahun 2018 lalu. Beberapa korban kecelakaan ini memiliki pekerjaan yang tidak dapat mereka tinggalkan, salah satu contohnya adalah mereka yang harus mengoperasikan komputer. Korban kecelakaan yang mengalami luka berat seperti patah tulang permanen atau tidak dapat duduk dengan posisi normal kesulitan untuk mengoperasikan komputer secara maksimal. Selain korban kecelakaan lalu lintas, ada juga orang yang kesulitan dalam mengoperasikan komputer, yaitu penyandang disabilitas.

Penyandang disabilitas adalah setiap orang yang mengalami keterbatasan fisik, intelektual, mental, dan/atau sensorik yang dalam berinteraksi dengan lingkungan dapat mengalami hambatan dan kesulitan untuk berpartisipasi secara penuh dan efektif dengan orang lainnya berdasarkan kesamaan hak. Daripada bekerja secara fisik akan lebih baik apabila mereka bekerja di depan komputer karena tidak memerlukan tenaga fisik yang besar. Untuk mengoperasikan komputer, penggunaan kursor sangatlah penting untuk berbagai keperluan, contohnya menggerakkan kursor ke sebuah *icon/shortcut* tertentu, kemudian klik kiri untuk mengakses sebuah file, klik kanan untuk membuka menu tambahan dan masih banyak yang lainnya. Namun, tidak semua penyandang disabilitas mampu mengoperasikan *mouse* dengan maksimal karena keterbatasannya.

Tidak kalah dengan berkembangnya *smartphone* yang sedang marak saat ini, teknologi komputer juga terus berkembang. Baik dalam hal pendidikan, pekerjaan maupun hanya untuk bersenang-senang mengisi waktu seperti main *game*, belanja secara daring, ataupun membaca berita. Pada perangkat komputer sudah mulai ada beberapa alat ataupun aplikasi untuk membantu para penyandang disabilitas seperti *speech recognition*, robot pengetik, tetikus yang berbentuk seperti sarung tangan yang dipasang untuk penggerak perangkat penunjuk, dan lain sebagainya.

Pada penelitian sebelumnya (Agustinus, 2018) telah diciptakan sebuah prototipe perangkat penunjuk menggunakan *Electromyography* atau biasa disebut dengan EMG (sensor otot) sebagai prototipe 1 dan *bend sensor* (sensor tekuk) sebagai prototipe 2, yang keduanya merupakan perangkat masukan untuk pendeteksi klik. Pendeteksi klik yang telah diselesaikan adalah untuk mendeteksi klik kiri saja. Berdasarkan saran dari peneliti sebelumnya, disarankan untuk mengembangkan prototipe 1 atau sensor EMG karena sensor EMG lebih nyaman dipakai dan mengeluarkan upaya yang lebih sedikit dibandingkan menggunakan sensor tekuk.

Oleh karena itu, pada tugas akhir ini akan dikembangkan pada fitur klik. Pada penelitian ini akan mencoba menggunakan satu sensor yaitu sensor *Electromyograph* (EMG) dengan alat khusus dari biosignalplux. Hasil dari sensor EMG tersebut akan diklasifikasikan menjadi dua kelas yaitu, klik kiri, dan klik kanan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah disebutkan, maka identifikasi masalahnya adalah ketidakmampuan seseorang untuk mengoperasikan komputer pada posisi normal yang disebabkan karna berbagai hal.

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah sebagai berikut:

- a. orang yang tidak dapat mengoperasikan komputer pada posisi normal atau duduk,

- b. tempat peletakan EMG adalah pada lengan kiri atas,
- c. biosignals explorer adalah perangkat keras dan open signals merupakan perangkat lunak yang digunakan untuk pengambilan data dari sensor EMG,
- d. data latih, data uji, dan responden diperoleh dari orang normal bukan olahragawan

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah di atas, maka dapat dibuat rumusan masalah yaitu bagaimana mengembangkan metode klasifikasi yang dapat mengklasifikasikan jenis-jenis klik berdasarkan input dari EMG.

1.5 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah dapat mengembangkan metode klasifikasi yang dapat mengklasifikasikan jenis-jenis klik.

1.6 Manfaat

Manfaat yang didapatkan dari tugas akhir ini adalah memberikan media alternatif untuk orang yang tidak dapat mengoperasikan komputer pada posisi normal contohnya pada posisi tidur agar dapat mengoperasikan komputer dengan baik.

1.7 Sistematika Penulisan

Penulisan laporan tugas akhir ini ditulis dengan sistematika penulisan sebagai berikut.

1. Bab I Pendahuluan

Berisi latar belakang, permasalahan, dan tujuan penelitian yang berkaitan dengan perancangan aplikasi kecerdasan buatan untuk mendeteksi jenis klik.

2. Bab II Tinjauan Pustaka

Berisi teori-teori mengenai alat yang akan digunakan dalam melakukan penelitian. Alat yang digunakan adalah sensor EMG beserta hub dan juga aplikasi OpenSignals (r)evolution untuk pengambilan data. Menggunakan

metode K-NN dan SVM untuk proses klasifikasi, dan bahasa pemrograman yang digunakan adalah Python.

3. Bab III Analisis dan Perancangan Sistem

Berisi perancangan sistem mulai dari tahap penelitian, analisis kebutuhan, desain sistem yang berisi kontraksi otot lengan atas manusia, pembacaan data oleh sensor, sensor EMG Biosignalplux, *feature extraction*, pembelajaran pada K-NN dan SVM, implementasi sistem serta rancangan pengujian.

4. Bab IV Hasil dan Pembahasan

Berisi tentang pembahasan dan hasil yang telah dilakukan dalam penelitian ini, seperti hasil pengumpulan data yang berisi pemasangan perangkat sensor pada responden, proses pengambilan data responden, proses pengolahan *dataset*, proses *splitting data train* dan *data test*. Selanjutnya terdapat hasil rancangan aplikasi yang berisikan antarmuka halaman *home*, antarmuka halaman *accuracy*, antarmuka halaman *information*. Terdapat pula dua hasil pengujian sistem yaitu pengujian sistem K-NN dan SVM. Setelah hasil rancangan aplikasi, terdapat hasil kuesioner pengguna sistem, dan diskusi hasil.

5. Bab V Penutup

Berisi simpulan menyeluruh berdasarkan hasil dan uji coba yang sudah dilakukan, serta saran–saran perbaikan atau berbagai aspek lain yang perlu dikaji lebih lanjut untuk penelitian selanjutnya.

Bab II

Tinjauan Pustaka

2.1 Otot

Tubuh manusia dapat digerakan karena adanya otot. Otot adalah kumpulan sel otot yang membentuk jaringan yang berfungsi menggerakkan organ tubuh (Sari, 2019). Sel-sel otot mempunyai kemampuan berkontraksi (bentuk sel otot memendek) dan melakukan relaksasi (kembali ke ukuran semula). Oleh karena itu jaringan otot disebut sebagai alat gerak aktif, sedangkan rangka tubuh merupakan alat gerak pasif. Fungsi otot pada manusia antara lain:

- a. Menghasilkan gerakan rangka, seperti kontraksi dan relaksasi otot yang menempel pada rangka dapat menggerakkan rangka
- b. Mempertahankan postur dan posisi tubuh, misalnya mempertahankan posisi kepala saat membaca buku, berjalan dengan posisi tegak dan lain sebagainya.
- c. Mengatur pintu masuk dan keluar saluran dalam sistem tubuh, misalnya menelan, buang air besar maupun kecil semua hal tersebut dipengaruhi oleh otot rangka yang menyelaputinya.
- d. Menyokong jaringan lunak, menggerakkan organ-organ dalam tubuh seperti usus, jantung dan sistem tubuh lainnya.
- e. Mempertahankan suhu tubuh, kontraksi rangka memerlukan energidan menghasilkan panas untuk mempertahankan suhu normal bagi tubuh

Struktur pada otot memiliki beberapa bagian dan fungsinya masing-masing.

Penjelasan struktur penyusun jaringan otot adalah sebagai berikut:

a. Tendon

Tendon adalah penghubung otot dengan tulang yang mempunyai serabut berwarna putih dan tidak elastis disebut aponeurosis. Aponeurosis berfungsi untuk melekatkan satu otot dengan yang lain. Setiap otot punya tendon di ujung-ujungnya. Tendon memiliki kemampuan meregang yang sangat kecil. Tugas tendon adalah untuk mengirimkan daya di antara tulang dan otot. Pada dasarnya tendonlah

yang memungkinkan kita bergerak karena tendon adalah perantara ketika otot menggerakkan tulang.

b. Fascia

Fascia adalah jaringan penghubung dan penyusun kolagen yang letaknya berada di bawah kulit. Bagian ini adalah salah satu bagian yang cukup penting dalam membentuk otot badan. Fungsi utama fascia adalah sebagai pembungkus, pelekat, serta pemisah antara otot-otot serta organ-organ internal. Untuk lebih mudah, banyak orang menyebutnya sebagai connective tissue (selaput otot). Fascia sendiri terletak di mana-mana, mulai dari tendon, ligament, dan organ. Fascia membantu tubuh untuk menyelimuti otot sehingga gerakan tubuh kita menjadi lebih luwes dan tidak patah-patah layaknya robot.

c. Sarkolema

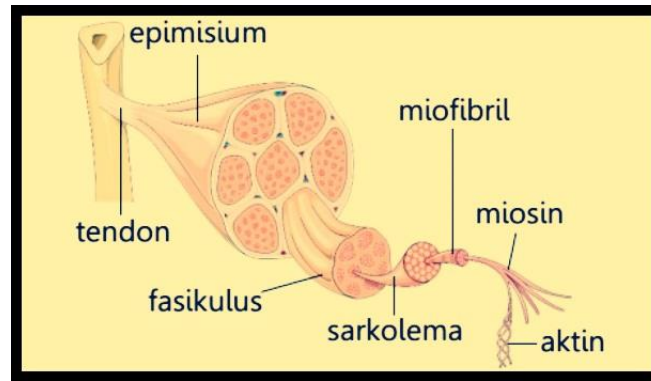
Sarkolema adalah membrane yang melapisi suatu sel otot yang fungsinya sebagai pelindung otot. Besar dan jumlah jaringan akan meningkat sejalan dengan penambahan usia.

d. Miofibril

Miofibril adalah suatu struktur berbentuk silindris pada serabut otot yang terdiri dari filamen-filamen (miofilamen). Ada 3 jenis miofilamen pada otot, yaitu : 1. Actin (filamen tipis), 2. miosin (filamen tebal) dan 3. titin (filamen elastis). Pada setiap ujung serat otot, miofibril berikatan dengan permukaan dalam sarkolemma. Bila miofibril memendek, maka otot akan memendek dan terjadi kontraksi.

e. Miofilamen

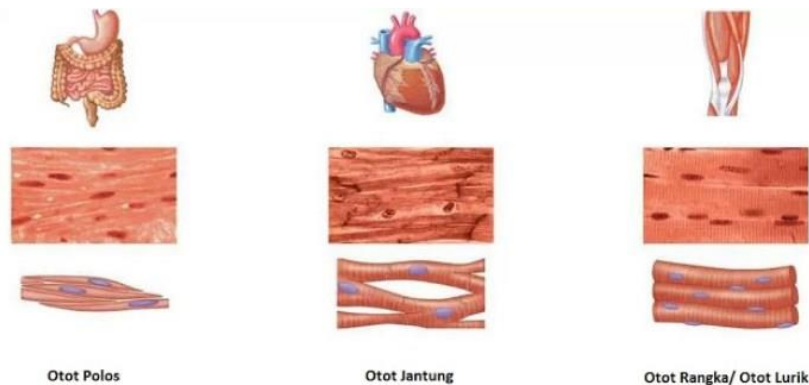
Miofilamen merupakan jaringan berbentuk benang-benang atau filament halus yang berasal dari myofibril. Jaringan ini terdapat dua macam yaitu miofilamen homogeny dan miofilamen heterogen yang terdapat pada otot jantung dan otot lurik.



Gambar 2.1 Struktur Jaringan Otot

(Sumber: <https://materibelajar.co.id/wp-content/uploads/2019/05/Struktur-jaringan-otot.jpg>)

Otot pada manusia terbagi menjadi tiga jenis, yaitu otot polos, otot jantung, dan otot rangka. Otot rangka atau otot lurik merupakan otot yang melekat pada tulang dan berperan sebagai sistem perototan yaitu menggerakkan tubuh (Madri, 2017). Otot ini bekerja secara sadar atas perintah manusia yang berasal dari otak. Ciri-ciri otot rangka adalah memiliki bentuk silindris dengan warna gelap dan terang, melekat pada rangka manusia, cepat dan mudah lelah, berbentuk panjang, dan memiliki banyak inti sel yang berada di tepi. Jaringan otot rangka bersifat volunter karena berkontraksi dan berelaksasi di bawah kontrol kesadaran.



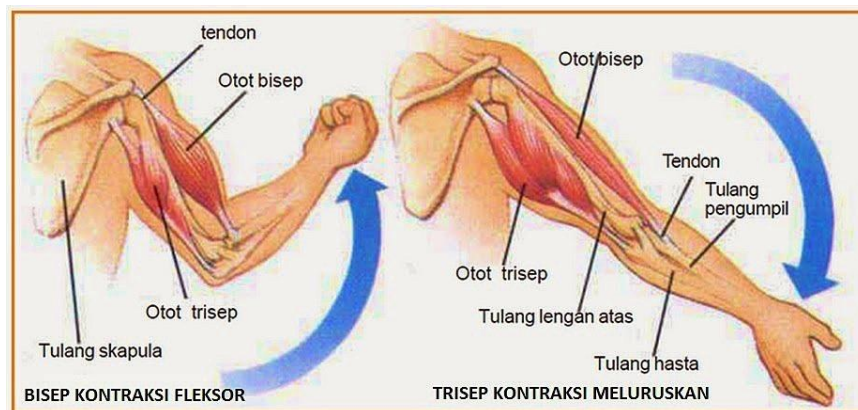
Gambar 2.2 Jenis-jenis otot

(Sumber: <https://seputarilmu.com/wp-content/uploads/2020/02/Jaringan-Otot.jpg>)

Mekanisme pembangkitan sinyal listrik di dalam otot sangat terkait dengan konsep motor unit atau unit motorik. Sistem saraf pusat tersusun dalam tingkatan-

tingkatan. Bagian korteks mengirimkan sinyal ke *spinal cord* (medula spinalis). *Spinal cord* kemudian meneruskan sinyal tersebut ke motorneuron (sel saraf motorik) yang ada di otot. Perlu diketahui bahwa motorneuron bersama-sama dengan serat-serat otot membentuk unit motorik. Oleh karena itu, ketika motorneuron diaktifkan oleh sinyal dari *spinal cord*, serat-serat otot akan diaktifkan pula untuk menghasilkan gerakan yang sesuai dengan perintah otak.

Otot pada bagian lengan atas manusia merupakan otot lurik. Ada dua jenis otot pada bagian lengan atas manusia, yaitu otot bicep dan otot trisep. Otot bicep terletak pada lengan atas bagian depan, sedangkan otot trisep terletak pada lengan atas bagian belakang. Posisi kedua otot tersebut tertera pada gambar 2.3. Kedua otot ini memiliki cara kerja yang saling berlawanan atau biasa disebut dengan cara kerja otot antagonis. Saat lengan bawah melakukan gerakan ke atas (mengangkat lengan bawah), otot bicep akan berkontraksi. Berlaku juga sebaliknya, saat lengan bawah dalam keadaan lurus, maka otot trisep akan berkontraksi.



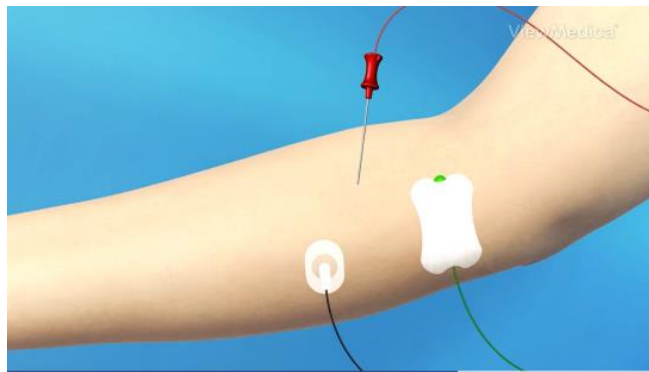
Gambar 2.3 Kontraksi Otot Bicep dan Trisep

(Gambar: <https://sel.co.id/wp-content/uploads/2018/03/Kontraksi-otot.jpg>)

2.2 Elektromiografi (EMG)

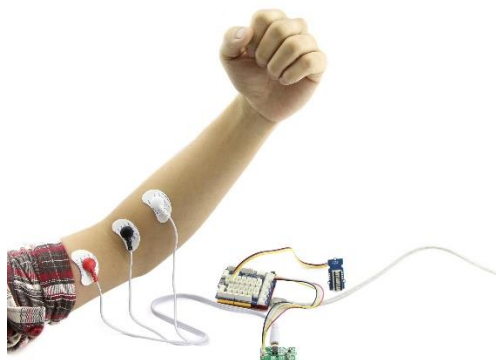
Elektromiografi atau biasa disebut dengan EMG adalah proses deteksi, analisis, dan pemanfaatan sinyal listrik yang berasal dari kontraksi otot (Anam, 2016 *cit.* C. De Luca, 2006). Alat yang digunakan dalam proses ini disebut dengan Elektromiograph, sedangkan sinyal yang dihasilkan disebut dengan elektromiogram atau sinyal mioelektrik. EMG banyak dimanfaatkan untuk proses rehabilitas penyandang disabilitas, baik karena amputasi ataupun karena sebab lain seperti stroke (Anam, 2016). Sinyal EMG bisa diperoleh dengan dua cara, melalui

penanaman elektroda (*Intramuscular* EMG) dan tanpa penanaman elektrode di dalam tubuh pasien (*Surface* EMG). *Intramuscular* EMG menggunakan jarum yang berbentuk kawat halus yang diletakkan dalam otot seperti pada gambar 2.4. Elektrode yang ditanam memberikan sinyal yang lebih baik dan langsung dari sumber otot yang diinginkan. Namun, proses pemasangannya harus melalui operasi bedah sehingga kurang disukai dan dihindari. Elektrode yang tidak ditanam atau diletakkan di permukaan lebih banyak digunakan. Hanya saja, sinyal yang diperoleh tidak sebagus yang ditanam serta sering kali dipengaruhi oleh sinyal dari otot-otot yang di sekitarnya. Metode elektrode yang diletakkan dipermukaan biasa disebut dengan *Surface* EMG adalah sebuah teknik di mana elektroda ditempatkan pada kulit diatasnya otot untuk mendeteksi aktivitas listrik pada otot. Pemasangan *Surface* EMG dapat dilihat pada gambar 2.5.



Gambar 2.4 *Intramuscular* EMG

(Sumber: <https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electromyography-emg>)



Gambar 2.5 *Surface* EMG

(Sumber: https://raw.githubusercontent.com/SeedDocument/Grove-EMG_Detector/master/img/Emg_connect.jpg)

Untuk mendapatkan sinyal EMG, perlu dilakukan desain sistem yang tepat yang mempertimbangkan *noise* yang mungkin terjadi. *Noise* bisa terjadi pada semua tahapan dari proses akuisisi. Padahal, proses akuisisi data diupayakan semaksimal mungkin untuk mendapatkan sinyal EMG yang mengandung sebanyak mungkin informasi dengan seminimal mungkin *noise*. Satu hal yang bisa dilakukan untuk mengatasi *noise*, khususnya dari radiasi listrik pada frekuensi 50 atau 60 Hz, adalah menggunakan penguat beda (differential amplifier). Hal ini dilakukan dengan menggunakan dua elektrode dari dua lokasi yang berbeda. Ide dari penguat beda ini adalah membuang sinyal yang sama dari dua elektrode dan menguatkan beda sinyal dari keduanya. Sinyal yang sama mewakili sinyal yang berasal dari lokasi yang jauh dari pendeteksian sinyal, sedangkan beda sinyal mewakili area langsung dari permukaan yang dideteksi. Jadi, sinyal EMG lokal akan dikuatkan dan *noise* karena daya listrik yang jauh akan dibuang.

2.2.1 Biosignalplux

Menurut Ilham (2019), biosignal adalah sinyal pada makhluk hidup yang dapat diukur dan dimonitor secara konstan. Istilah biosignal sering digunakan untuk merujuk pada sinyal bioelektrik, tetapi dapat juga merujuk pada sinyal listrik dan non-listrik. Sinyal bioelektrik adalah sinyal yang dihasilkan oleh sel-sel saraf dan sel-sel otot. Berdasarkan bentuk fisik dari klasifikasi biosignal, otot digolongkan dalam bentuk listrik karena adanya perubahan tegangan pada otot yang dapat dilihat pada *Electroencephalogram* (EGG), *Electrocardiography* (ECG), dan *Electromyograph* (EMG).

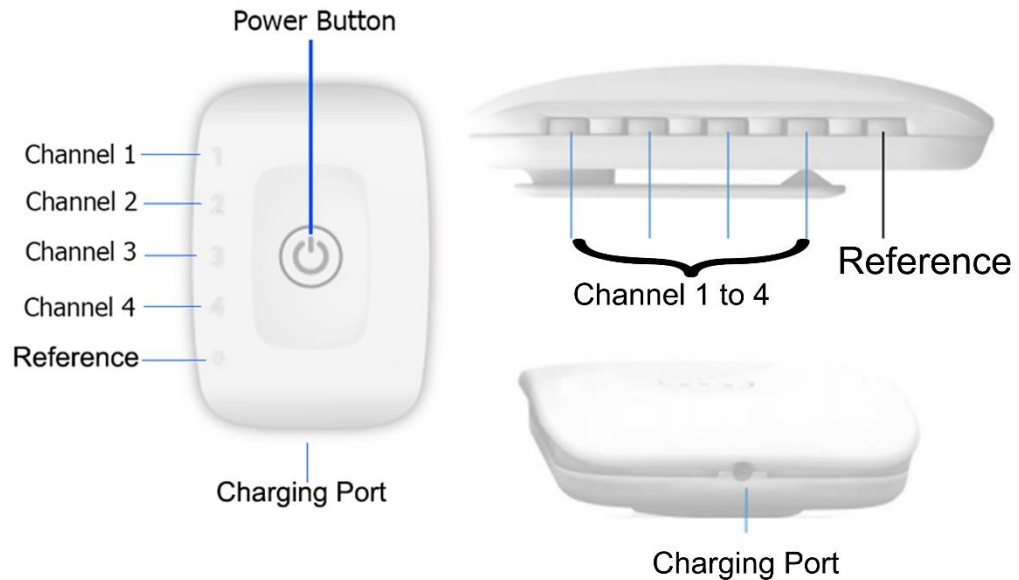
Biosignalsplux merupakan salah satu produk dari perusahaan yang bernama Plux. Sejak tahun 2007, Perusahaan Plux membuat berbagai produk yang inovatif untuk fisioterapi dan penelitian contohnya EMG, ECG, *respiration*, dan *accelerometers*. Biosignalsplux adalah set sensor yang dapat mengukur data biosignal berupa kontraksi otot yang mampu digunakan secara nirkabel (*wireless*) yang didesain untuk memudahkan peneliti mendapatkan data biosignal yang memiliki reliabilitas tinggi pada berbagai macam aplikasi dan kasus penggunaan. Terdapat sembilan *research kits* pada biosignalsplux, salah satunya adalah Biosignals Explorer. Biosignal Explorer meliputi:

- a. 1 x Wireless 4-channel hub, dengan spesifikasi:

Tabel 2.1 Spesifikasi *Wireless* 4-channel Hub

Komponen	Keterangan
<i>Analog Ports</i>	4 <i>generic inputs</i>
<i>Auxiliary Ports</i>	1 <i>ground</i>
<i>Resolution</i>	<i>up to 16-bit (per channel)</i>
<i>Sampling Rate</i>	<i>up to 4000 Hz (per channel)</i>
<i>Communications</i>	<i>Bluetooth Class II</i>
<i>Range</i>	<i>up to ~10 m (extendable)</i>
<i>Internal Memory</i>	<i>(optional extra)</i>
<i>Battery Life</i>	<i>~10 h streaming</i>
<i>Size</i>	85 x 54 x 10 mm

- b. 4 x Professional sensors. Dapat memilih dari pilihan berikut ini:
- 1) EMG
 - 2) *Electrodermal Activity* (EDA)
 - 3) ECG
 - 4) EEG
 - 5) *Accelerometer* (ACC)
 - 6) *Temperature* (TMP)
 - 7) *Respiration* (PZT)
 - 8) *Force* (FSR)
 - 9) *Light* (LUX)
 - 10) *Synchronization Kit*
 - 11) *Foot Switch*
 - 12) *Push Button*
- c. 24 x *Pre-gelled Electrodes*
- d. 1 x *Bluetooth dongle*
- e. 1 x *Medical-grade charger*
- f. 1x *Portable and rugged storage case with foam cushioning to house all the parts*



Gambar 2.6 4-Channel Biosignalsplux Hub

Pada Biosignalsplux Hub 4-channel terdapat *power button* yang berfungsi untuk menghidupkan dan mematikan perangkat dan menampilkan status perangkat dengan LED seperti pada gambar 2.6. Jika LED menyala maka perangkat tersebut sedang aktif, dan jika LED mati maka perangkat tersebut sedang tidak aktif. Empat *channel* dalam Hub ini yang dapat menghubungkan sampai dengan empat sensor secara bersamaan tanpa menginterupsi sensor yang lain. Referensi berguna untuk menghubungkan kabel ground agar sensor dapat berjalan dengan baik. *Charging Port* berfungsi untuk pengisian daya ketika baterai pada Hub habis. Letak *Charging Port* berada pada bagian bawah Hub. Baterai pada Hub dapat bertahan sampai dengan dua belas jam dengan penggunaan secara terus menerus. LED pada Hub akan berwarna merah jika baterai sudah mulai habis. Untuk pengisian daya, dapat dilakukan dengan cara seperti pengisian daya pada umumnya yaitu dengan menghubungkan *charger* ke Hub dan ke stopkontak. Proses pengisian daya akan memakan waktu sekitar dua jam setengah agar baterai dapat terisi dengan penuh. LED pada tombol *power* akan berwarna merah ketika sedang pengisian daya dan akan mati ketika baterai sudah penuh. Pada saat pengisian daya disarankan agar Hub tidak dipakai terlebih dahulu.

Ada dua cara untuk Hub ini dapat terhubung dengan perangkat Biosignalsplux lainnya, yaitu dengan menggunakan Bluetooth dan USB. Penggunaan USB biasa digunakan hanya untuk mengunduh data secara luring,

sedangkan untuk Bluetooth digunakan untuk mengkonfigurasi perangkat, mendapatkan, serta mengirimkan sinyal sensor secara *real-time* ke komputer, dan juga untuk mengunduh data secara luring yang terdapat pada *internal memory* perangkat Biosignalsplux. Biosignalsplux menggunakan Bluetooth *Class II* untuk berkomunikasi dengan komputer serta mengirimkan data sensor yang ada. Karena sebagian besar modul Bluetooth internal tidak dirancang untuk mendukung *transfer rate* yang tinggi seperti yang diperlukan ketika memperoleh dan *streaming* sinyal menggunakan biosignalsplux, Plux menyediakan Bluetooth *dongle*. *Dongle* adalah suatu perangkat kecil yang dapat dihubungkan dan digunakan dengan komputer, terutama untuk mengakses perangkat lunak nirkabel.

Cara mengaktifkan Hub adalah dengan cara menekan tombol *power* pada tengah hub. Setelah dinyalakan, lampu status LED akan berkedip sekali per detik. Sensor yang ingin digunakan dapat dihubungkan ke *channel* satu sampai empat agar dapat bekerja dengan baik. Tidak disarankan untuk menghubungkan sensor ke *port reference* atau *ground*. *Port reference* atau *ground* dihubungkan hanya untuk *ground* itu sendiri.



Gambar 2.7 Biosignalsplux EMG

Sensor EMG pada Biosignalsplux berbentuk seperti pada gambar 2.7. Sensor EMG dapat dihubungkan ke salah satu masukan analog yang tersedia (*channel* satu sampai empat). Sensor EMG milik Biosignalsplux tidak memiliki *reference* yang menyatu dengan sensor EMG. Untuk hasil yang lebih baik, disarankan untuk menggunakan *reference* sebagai *ground* yang dipasang pada bagian tubuh yang memiliki aktivitas otot yang rendah.

2.2.2 OpenSignals (r)evolution

OpenSignals (r)evolution merupakan perangkat lunak yang digunakan untuk akuisisi, visualisasi, dan memproses data dari perangkat keras

Biosignalsplux. OpenSignals (r)evolution memungkinkan pengguna untuk memperoleh data dari satu atau lebih perangkat, mengatur parameter yang diinginkan seperti frekuensi sampling dan resolusi, menyimpan format teks ASCII atau dalam bentuk *Hierarchical Data Format* (HDF) untuk diproses lebih lanjut seperti pada Python atau MatLab dan memvisualisasikan data yang diperoleh melalui sejumlah *channels*. OpenSignals dapat menyimpan data ke dalam tiga format, yaitu .txt, .h5, dan .edf.



Gambar 2.8 Tampilan awal Opensignals (r)evolution

2.3 Perangkat Penunjuk

Menurut Lexico (n.d), perangkat penunjuk merupakan istilah umum untuk perangkat keras apa pun, seperti tablet grafis, *mouse*, *stylus*, atau *trackball* yang berfungsi untuk mengontrol pergerakan kursor pada layar komputer. Perangkat penunjuk merupakan perangkat keras yang berfungsi untuk memberi masukan pada komputer. Beberapa tipe perangkat penunjuk adalah sebagai berikut:

a. *Mouse* / Tetikus.

Mouse atau tetikus dalam Bahasa Indonesia merupakan suatu perangkat penunjuk yang biasa digunakan pada komputer atau laptop. Terdapat dua macam tetikus, yaitu tetikus mekanik dan tetikus optikal. Tetikus mekanik memiliki sebuah bola di dalamnya yang menyentuh permukaan dan berputar bersama dengan geraknya tetikus. Sedangkan tetikus optikal menggunakan tembakan laser dan chip khusus untuk menghasilkan data yang sesuai untuk pergerakan mouse dan pointer pada computer.

b. *Touchpad*

Touchpad hampir sama cara kerjanya dengan touchscreen, namun *touchpad* hanya bisa digunakan di tempat yang sudah disediakan. Pada umumnya *touchpad* berbentuk persegi atau persegi panjang. Cara kerjanya dengan menggesekkan jari pada permukaan touchpad dan menekan pada permukaannya untuk mengakses fungsi klik. *Touchpad* kebanyakan digunakan pada laptop.

2.4 Penyandang Disabilitas

Istilah disabilitas berasal dari kata bahasa inggris yaitu *disability* yang memiliki arti cacat atau ketidakmampuan. Dalam Kamus Besar Bahasa Indonesia (KBBI), “disabilitas adalah keadaan yang merusak atau membatasi kemampuan mental dan fisik seseorang, atau dapat diartikan juga keadaan seseorang yang tidak mampu melakukan hal-hal dengan cara yang biasa”. Menurut Undang-Undang Nomor 4 Tahun 1997 tentang Penyandang Cacat, Penyandang Disabilitas dikategorikan menjadi tiga jenis, yaitu penyandang disabilitas fisik, penyandang disabilitas mental, dan penyandang disabilitas ganda atau penyandang disabilitas fisik dan mental.

Penyandang disabilitas fisik adalah kelainan yang mengakibatkan gangguan pada fungsi tubuh, antara lain gerak tubuh, penglihatan, pendengaran, dan kemampuan berbicara. Disabilitas fisik terdapat empat jenis, yaitu kelainan pada tubuh (tuna daksa), kelainan pada indera penglihatan (tuna netra), kelainan pada pendengaran (tuna rungu), dan kelainan pada bicara (tuna wicara).

Tuna daksa adalah individu yang memiliki gangguan gerak yang disebabkan oleh kelainan neuro-muskular dan struktur tulang yang bersifat bawaan, sakit atau akibat kecelakaan (kehilangan organ tubuh), polio dan lumpuh. Tuna daksa berasal dari kata tuna yang berarti rusak atau cacat, sedangkan daksa berarti tubuh. Jadi tuna daksa ditujukan bagi mereka yang memiliki anggota tubuh tidak sempurna. Tuna daksa dapat digolongkan menjadi dua, yaitu pertama, menurut sebab kelainan yang berarti kelainan sejak lahir, disebabkan oleh penyakit, disebabkan kecelakaan, dan disebabkan oleh perang. Kedua, menurut jenis kelainannya, contohnya putus (amputasi) tungkai dan lengan; cacat tulang, sendi, dan otot pada tungkai dan

lengan. Dalam Pasal 12 Undang-undang nomor 8 Tahun 2016 tentang Hak kesehatan untuk Penyandang Disabilitas meliputi hak:

- a. memperoleh informasi dan komunikasi yang mudah diakses dalam pelayanan kesehatan,
- b. memperoleh kesamaan dan kesempatan akses atas sumber daya di bidang kesehatan,
- c. memperoleh kesamaan dan kesempatan pelayanan kesehatan yang aman, bermutu, dan terjangkau,
- d. memperoleh kesamaan dan kesempatan secara mandiri dan bertanggung jawab menentukan sendiri pelayanan kesehatan yang diperlukan bagi dirinya,
- e. memperoleh alat bantu kesehatan berdasarkan kebutuhannya,
- f. memperoleh obat yang bermutu dengan efek samping yang rendah,
- g. memperoleh perlindungan dari upaya percobaan medis, dan
- h. memperoleh perlindungan dalam penelitian dan pengembangan kesehatan yang mengikutsertakan manusia sebagai subjek.

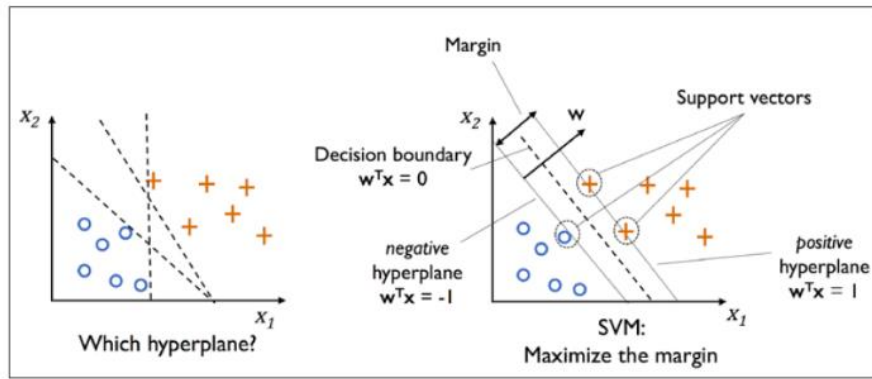
2.5 Python

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum (Muahardian, 2018). Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. Python banyak digunakan untuk membuat berbagai macam program seperti program CLI, program GUI (*desktop*), aplikasi *mobile*, *website*, IoT, untuk membuat permainan, dan lain sebagainya. Python merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan data, penyalinannya atau mendistribusikannya.

2.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory. SVM adalah metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah class pada *input space* (Nugroho

dkk., 2003). Menurut Vapnik dan Cortes (1995), *Support Vector* merupakan metode klasifikasi *machine learning* untuk dua kelompok. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. *Hyperplane* terbaik dapat ditemukan dengan mengukur margin dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* dengan *pattern* atau objek terluar terdekat dari masing-masing *class*. Dalam SVM objek data terluar yang paling dekat dengan *hyperplane* disebut *support vector*. Objek yang disebut *support vector* yang paling sulit diklasifikasikan dikarenakan posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Mengingat sifatnya yang kritis, hanya *support vector* inilah yang diperhitungkan untuk menemukan *hyperplane* yang paling optimal oleh SVM.



Gambar 2.9 Contoh *Hyperplane* pada SVM

(Sumber: https://miro.medium.com/max/731/1*ikAtK9PHxDH1xDvaXEUKTw.png)

Data yang ada dinotasikan sebagai $\vec{x}_i \in \mathcal{R}$, artinya data yang ada merupakan elemen dari angka riil. Berdasarkan jurnal milik Vapnik dan Cortes, satu data latih diberi label $(y_i, x_i), \dots, (y_l, x_l)$, dimana y_i adalah *class* yang beranggotakan -1 dan 1, sedangkan x_i merupakan *input vector*. Diasumsikan bahwa kedua *class* -1 dan 1 terpisah secara sempurna oleh *hyperplane* berdimensi d yang didefinisikan pada persamaan (2-1).

$$w_0 \cdot x + b_0 = 0 \quad (2-1)$$

Pattern x_i yang termasuk *class* -1 dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan (2-2).

$$w \cdot x_i + b \leq -1 \quad (2-2)$$

Sedangkan *pattern* x_i yang termasuk *class* 1 dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan (2-3).

$$w \cdot x_i + b \geq 1 \quad (2-3)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik terdekatnya, yaitu dengan $\frac{1}{\|w\|}$. Hal ini dapat juga disebut *Quadratic Programming (QP) problem*, yaitu mencari titik minimal persamaan (2-4), dengan memperhatikan constrain persamaan (2-5). QP adalah masalah mengoptimalkan fungsi objektif kuadrat dan merupakan salah satu bentuk paling sederhana dari pemrograman non-linear. *Problem* ini dapat dipecahkan dengan berbagai Teknik komputasi, di antaranya Lagrange Multiplier seperti pada persamaan (2-6).

$$\min_{\vec{w}} \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \quad (2-4)$$

$$y_i(x_i \cdot w + b) - 1 \geq 0, \quad \forall i \quad (2-5)$$

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\vec{x}_i \cdot \vec{w} + b) - 1) \quad (i = 1, 2, \dots, l) \quad (2-6)$$

α_i adalah Lagrange Multipliers, yang bernilai nol atau positif ($\alpha_i \geq 0$). Nilai optimal dari persamaan (2-6) dapat dihitung dengan meminimalkan L terhadap \vec{w} dan b , dan memaksimalkan L terhadap α_i . Dengan memperhatikan sifat bahwa pada titik optimal gradient $L = 0$, persamaan (2-6) dapat dimodifikasi sebagai maksimalisasi *problem* yang hanya mengandung α_i saja, sebagai mana persamaan (2-7) di bawah ini.

Maximize:

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \quad (2-7)$$

Subject to:

$$\alpha_i \geq 0 \quad (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i y_i = 0$$

Dari hasil perhitungan di atas akan diperoleh α_i yang kebanyakan bernilai positif. Data inilah yang disebut sebagai *support vector*.

Algoritma SVM menggunakan seperangkat fungsi matematika yang didefinisikan sebagai kernel. Fungsi kernel adalah untuk mengambil data sebagai input dan mengubahnya menjadi bentuk yang diperlukan. Algoritma SVM yang berbeda menggunakan berbagai jenis fungsi kernel. Contoh jenis-jenis fungsi kernel pada SVM adalah Linear, Nonlinear, Radial Basis Function (RBF), Sigmoid, dan Polynomial.

2.6.1 Linier

Linier merupakan kernel SVM yang paling sederhana. Linear kernel biasa digunakan ketika data yang dianalisis sudah terpisah secara linear. Berikut merupakan persamaan dari linear kernel SVM.

$$k(x, y) = x^T \cdot y \quad (2-8)$$

2.6.2 Radial Basis Function (RBF)

RBF kernel merupakan fungsi kernel yang biasa digunakan dalam analisis ketika data tidak terpisah secara linear. RBF kernel memiliki dua parameter yaitu Gamma dan Cost. Parameter Cost atau biasa disebut sebagai C merupakan parameter yang bekerja sebagai pengoptimalan SVM untuk menghindari misklasifikasi di setiap sampel dalam training dataset. Parameter Gamma menentukan seberapa jauh pengaruh dari satu sampel training dataset dengan nilai rendah berarti “jauh”, dan nilai tinggi berarti “dekat”. Dengan gamma yang rendah, titik yang berada jauh dari garis pemisah yang masuk akal dipertimbangkan dalam perhitungan untuk garis pemisah. Ketika gamma tinggi berarti titik – titik berada di sekitar garis yang masuk akal akan dipertimbangkan dalam perhitungan. Berikut merupakan persamaan dari RBF kernel.

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (2-9)$$

2.6.3 Sigmoid

Kernel Sigmoid atau yang dikenal juga sebagai *Hyperbolic Tangent Kernel* dan juga *Multilayer Perceptron (MLP) Kernel*. Kernel Sigmoid berasal dari Neural

Network, dimana fungsi bipolar sigmoid biasa digunakan sebagai fungsi aktivasi untuk neuron tiruan. Berikut merupakan persamaan dari Kernel Sigmoid.

$$k(x, y) = \tanh(\alpha x^T y + c) \quad (2-10)$$

2.6.4 Polynomial

Polinomial kernel merupakan fungsi kernel yang digunakan ketika data tidak terpisah secara linear. Polinomial kernel sangat cocok untuk permasalahan dimana semua training dataset dinormalisasi. Berikut merupakan persamaan dari kernel Polynomial.

$$k(x_i, x_j) = (x_i \cdot x_j + c)^b \quad (2-11)$$

2.7 *K-Nearest Neighbor* (K-NN)

K-Nearest Neighbor atau biasa disebut K-NN adalah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasi sebelumnya (Informatikalogi, 2017). K-NN termasuk dalam *supervised learning*, dimana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak k-tetangga terdekat dari kategori yang ada dalam K-NN. *Supervised learning* adalah sebuah pendekatan yang sudah terdapat data yang dilatih, dan terdapat variabel yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada. Untuk menggunakan algoritma K-NN perlu ditentukan banyaknya k tetangga terdekat yang digunakan untuk melakukan klasifikasi data baru. Banyaknya k, sebaiknya merupakan angka ganjil, misalnya k = 1, 3, 5, dan seterusnya. Penentuan nilai k dipertimbangkan berdasarkan banyaknya data yang ada dan ukuran dimensi yang dibentuk oleh data. Semakin banyak data, angka k yang dipilih sebaiknya semakin rendah. Namun, jika semakin besar ukuran dimensi data, angka k yang dipilih sebaiknya lebih tinggi. Secara sederhana, K-NN bekerja berdasarkan jarak minimum dari data baru ke data latih untuk menentukan K-tetangga terdekat. Langkah-langkah dari algoritma K-NN, yaitu

1. menentukan parameter k (jumlah banyak tetangga terdekat),
2. menghitung jarak *Euclidean* objek terhadap data training yang diberikan,

3. mengurutkan hasil dari langkah ke 2 secara *ascending* (pengurutan dari nilai terkecil ke nilai terbesar)
4. mengumpulkan kategori Y (klasifikasi nearest neighbor berdasarkan nilai k),
5. dengan menggunakan kategori nearest neighbor yang paling mayoritas maka akan dapat diprediksi kategori objek dari data uji.

Cara menghitung jarak tetangga biasanya dihitung berdasarkan jarak Euclidean atau *Euclidean Distance*. Rumus dari *Euclidean Distance* adalah sebagai berikut.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2-12)$$

keterangan:

$d(x, y)$: jarak skalar dari dua buah vektor data x dan y yang berupa matrix berukuran d dimensi

x : data uji / data *train*

y : data latih / data test

n : jumlah data latih

2.8 Confussion Matrix

Pada umumnya banyak algoritma yang digunakan pada *machine learning* menggunakan *confusion matrix* untuk perhitungan akurasi. *Confussion matrix* adalah tabel yang sering digunakan untuk menggambarkan kinerja model klasifikasi pada satu set data pengujian yang nilai sebenarnya diketahui (Sharma, n.d.). Tabel dari *confusion matrix* dapat dilihat pada tabel 2.2.

Tabel 2.2 Tabel *Confusion Matrix*

	<i>Class 1</i> <i>Predicted</i>	<i>Class 2</i> <i>Predicted</i>
<i>Class 1</i> <i>Actual</i>	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
<i>Class 2</i> <i>Actual</i>	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Berdasarkan tabel 2.2, *Class 1* adalah kelas positif dan *Class 2* adalah kelas negatif. *True Positive* atau disingkat TP merupakan suatu nilai yang menunjukkan

bahwa berdasarkan pengamatan, suatu data menunjukkan kelas positif, dan prediksi menunjukkan kelas positif juga. *False Negative* atau disingkat FN merupakan suatu nilai yang menunjukkan bahwa berdasarkan pengamatan, suatu data menunjukkan kelas positif, namun prediksi menunjukkan kelas negatif. *False Positive* atau disingkat FP merupakan suatu nilai yang menunjukkan bahwa berdasarkan pengamatan, suatu data termasuk dalam kelas negatif, namun prediksi menunjukkan kelas positif. *True Negative* atau yang disingkat menjadi TN merupakan suatu nilai yang menunjukkan bahwa berdasarkan pengamatan, suatu data termasuk dalam kelas negatif, dan juga prediksi menunjukkan kelas negatif. Cara menghitung akurasi atau *Classification Rate* dengan metode *Confusion Matrix* berdasarkan tabel 2.2 adalah menggunakan rumus sebagai berikut.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2-13)$$

Namun, dengan menghitung akurasi ini belum bisa mendapatkan nilai kinerja dari algoritma tersebut. Sebuah akurasi tidak hanya untuk akurasi keberhasilan, bisa juga akurasi kegagalan suatu program. Maka dari itu, perlu dilakukan perhitungan lebih lanjut yaitu dengan menghitung *recall* dan *precision*. *Recall* dapat didefinisikan sebagai rasio dari jumlah total yang benar terklasifikasikan dalam *class* positif dibagi dengan jumlah seluruh *class* positif, baik yang terprediksi dengan benar maupun tidak benar. Jika nilai *recall* tinggi maka menunjukkan bahwa suatu *class* diidentifikasi dengan benar. Sedangkan untuk mendapatkan nilai *precision* adalah dengan cara membagi jumlah total yang benar terklasifikasikan dalam *class* positif dengan jumlah total prediksi pada *class* positif. Semakin tinggi hasil dari *precision* menunjukkan data yang terklasifikasikan dalam *class* positif adalah benar terklasifikasikan dalam *class* positif. Berikut merupakan persamaan dari *recall* dan *precision*.

$$Recall = \frac{TP}{TP+FN} \quad (2-14)$$

$$Precision = \frac{TP}{TP+FP} \quad (2-15)$$

Jika nilai *recall* tinggi dan nilai *precision* rendah menunjukkan bahwa contoh data yang terklasifikasi dalam *class* positif dikenali dengan benar. Sebaliknya, jika nilai *recall* rendah dan nilai *precision* tinggi menunjukkan bahwa

nilai *False Negative* tinggi juga yang artinya banyak data yang seharusnya masuk *class* positif terprediksi masuk dalam *class* negatif, namun banyak juga data yang terprediksi masuk *class* positif benar seharusnya masuk dalam *class* positif. Karena ada dua perhitungan yaitu *recall* dan *precision*, maka ada perhitungan *F-measure* yang mewakili keduanya. *F-measure* dihitung menggunakan *Harmonic Mean*. Nilai dari *F-measure* akan selalu mendekati nilai terkecil dari *recall* atau *precision*. Persamaan *F-measure* dapat dilihat pada persamaan (2-5).

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \quad (2-16)$$

2.9 Ekstraksi Fitur

Ekstraksi fitur adalah nama untuk metode yang memilih dan / atau menggabungkan variabel menjadi fitur, secara efektif mengurangi jumlah data yang harus diproses, sementara masih secara akurat dan lengkap menggambarkan kumpulan data asli.

2.9.1 Max EMG Value

Mencari nilai paling tinggi (maksimal) dari keseluruhan jumlah sample yang digunakan (1000 sample/detik)

2.9.2 Min EMG Value

Mencari nilai paling rendah (minimal) dari keseluruhan jumlah sample yang digunakan (1000 sample/detik)

2.9.3 Wave Length

Digunakan untuk mencari panjang gelombang yang didapatkan oleh sensor EMG yang nantinya akan digunakan sebagai salah satu fitur klasifikasi.

$$WL = \frac{1}{N} \sum_{i=1}^{n-1} |x_{i+1} - x_i| \quad (2-17)$$

2.9.4 Mean Absolute Deviation

$$MAD = \frac{\sum_{i=1}^n |x_i - \bar{x}|}{n} \quad (2-18)$$

Keterangan:

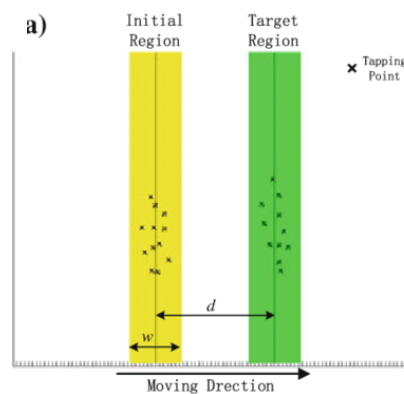
\bar{x} : mean / average

x_i : signal value

n : number of value

2.10 ISO 9241-411

ISO 9241-411 merupakan suatu pengujian untuk mengevaluasi efisiensi dan efektivitas dari suatu perangkat masukan. Terdapat empat macam jenis pengujian terdiri dari *One-directional tapping test*, *Multi-directional tapping test*, *Dragging test*, dan *Tracing test*. *One-directional tapping test* merupakan tes yang digunakan untuk mengevaluasi pergerakan penunjuk dalam satu sudut gerak pada layar komputer. Pola *One-directional tapping test* dapat dilihat pada gambar 2.10.



Gambar 2.10 Pola *One-directional tapping test*

(Sumber:

https://www.researchgate.net/profile/Guoli_Wang3/publication/286557005/figure/fig5/AS:613858926403601@1523366781504/The-GUIs-of-ISO-standard-tests-a-One-direction-tapping-test-b-Multi-direction-tapping.png)

Pengujian dilakukan dengan cara program akan menampilkan seperti pada gambar 2.9 kepada pengguna. Setiap persegi panjang memiliki lebar sebesar w dan jarak antar kedua persegi panjang sebesar d . Tujuan dari pengujian ini adalah untuk menggerakkan, mengarahkan dan juga klik berdasarkan satu sudut pada setiap persegi panjang sebanyak 25 kali klik. Setiap sesi dimulai ketika pengguna menggerakkan penunjuk ke persegi panjang dan melakukan klik. Pengujian ini memiliki beberapa tingkat kesulitan, contohnya jarak antar persegi panjang (d), dan lebar dari dua persegi panjang.

2.11 Analisis Statistika

Analisis statistika digunakan untuk mengukur tingkat keberhasilan prototipe yang terdiri dari analisis kualitatif dan analisis kuantitatif. Data yang akan dianalisis didapatkan dari pengumpulan data melalui kuesioner kepada sejumlah responden yang telah mencoba prototipe. Terdapat dua kuesioner yaitu penilaian upaya dan penilaian kenyamanan. Data yang telah diambil dari setiap responden akan dikumpulkan dan diolah untuk perbandingan performa prototipe dan alat penunjuk konvensional yang digunakan pada umumnya.

2.11.1 Analisis Kualitatif

Analisis Kualitatif adalah analisis yang dilakukan untuk mengetahui tingkat kenyamanan serta tingkat usaha dalam mengoperasikan prototipe yang dibuat menggunakan instrumentasi kuesioner standar ISO 9241-411 yang disebut *Borg Scale*. *Borg Scale* didesain untuk mengumpulkan data dari pengguna mengenai kinerja seluruh tubuh dan juga kinerja dari beberapa kelompok otot, seperti lengan, pundak, dan leher. Format instrumentasi kuesioner dari *Borg Scale* seperti pada tabel 2.3.

Tabel 2.3 Format Instrumentasi kuesioner *Borg Scale* ISO 9241-411

Effort			Effort
Arm	Shoulder	Neck	
() 10	() 10	() 10	Very, very strong (almost max.)
() 9	() 9	() 9	
() 8	() 8	() 8	
() 7	() 7	() 7	Very strong
() 6	() 6	() 6	
() 5	() 5	() 5	Strong (heavy)
() 4	() 4	() 4	Somewhat strong
() 3	() 3	() 3	Moderate
() 2	() 2	() 2	Weak (light)
() 1	() 1	() 1	Very weak
() 0.5	() 0.5	() 0.5	Very, very weak (just noticeable)
() 0	() 0	() 0	Nothing at all

2.11.2 Analisis Kuantitatif

Analisis Kuantitatif adalah analisis yang dilakukan untuk mengetahui performa prototipe yang dibuat menggunakan instrumentasi kuesioner standar ISO 9241-411 yang disebut *Rating Scale* atau penilaian bebas dan skala ketergantungan. Instrumentasi penilaian bebas mengarah pada performa, akurasi, hingga tingkat kelelahan bagian tubuh yang digunakan dalam pengoperasian prototipe. Format instrumentasi penilaian bebas dapat dilihat pada tabel 2.4.

Tabel 2.4 Format Instrumentasi Penilaian Bebas ISO 9241-411

1. Force required for actuation:						
1	2	3	4	5	6	7
Very uncomfortable				Very comfortable		
2. Smoothness during operation:						
1	2	3	4	5	6	7
Very rough				Very smooth		
3. Effort required for operation:						
1	2	3	4	5	6	7
Very high				Very low		
4. Accuracy						
1	2	3	4	5	6	7
Very inaccurate				Very accurate		
5. Operation speed:						
1	2	3	4	5	6	7
Unacceptable				Acceptable		
6. General Comfort:						
0.32.321	2	3	4	5	6	7
Very uncomfortable				Very comfortable		
7. Overall operation of input device:						
1	2	3	4	5	6	7
Very difficult (to use)				Very easy (to use)		
8. Finger fatigue:						
1	2	3	4	5	6	7
Very high				None		

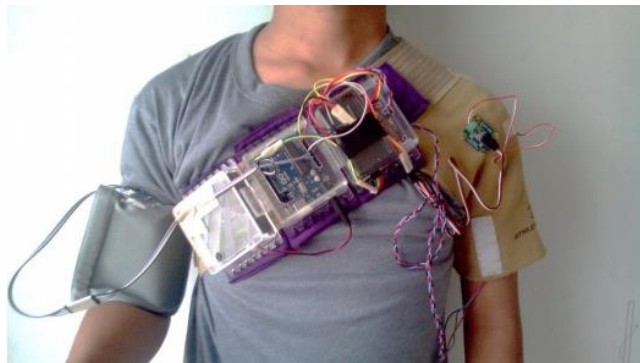
9. Wrist Fatigue:						
1	2	3	4	5	6	7
Very high				None		
10. Arm fatigue:						
1	2	3	4	5	6	7
Very high				None		
11. Shoulder Fatigue						
1	2	3	4	5	6	7
Very high				None		
12. Neck Fatigue:						
1	2	3	4	5	6	7
Very high				None		

2.12 Penelitian Terdahulu

Penelitian mengenai prototipe perangkat penunjuk menggunakan *Electromyography* sebelumnya dilakukan oleh Agustinus Bohaswara Hayrasena. Pada penelitian ini terdapat dua prototipe, prototipe 1 merupakan kombinasi sensor EMG dan sensor inersia, sedangkan prototipe 2 merupakan kombinasi sensor tekuk (*bend sensor*) dan sensor inersia. Sensor inersia digunakan untuk menggerakkan kursor dari perangkat penunjuk, sedangkan sensor tekuk dan sensor EMG berfungsi sebagai operasi klik kiri. Alat yang digunakan dalam prototipe ini antara lain, pertama adalah GY-591 yang merupakan sensor inersia dengan Sembilan derajat kebebasan. Kedua, InertiaCube4, yang merupakan sensor inersia berstandar industry yang berguna sebagai perhitungan galat pada perangkat GY-591. Ketiga, Myoware yang merupakan sensor otot atau EMG Myoware mampu mendeteksi kontraksi otot melalui *pad electrode* yang terpasang pada bisep. Keempat, sensor tekuk atau *bend sensor* merupakan sensor yang mengukur besaran yang dihasilkan oleh tekukan. Kelima, HC-05 merupakan perangkat keras yang berfungsi sebagai pengirim dan penerima sinyal Bluetooth. Keenam, Arduino Uno merupakan perangkat berupa papan mikrokontroler yang bersifat *open source* yang berdasarkan cara kerja *chip* IC ATmega328P yang berguna untuk menyatukan seluruh perangkat dan manajemen data yang terkirim melalui sinyal Bluetooth.

Pada penelitian ini menggunakan ISO 9241-411 sebagai tolak ukur efesiensi kinerja dan *Fitts' Law* sebagai tolak ukur efektifitas kinerja prototipe. Pengujian membutuhkan responden lebih dari 10 orang, yaitu pada penelitian ini berjumlah 12 orang. Setiap responden menjalankan tugas pengujian sebanyak 50 kali dengan masing-masing 25 kali *tapping* secara horizontal dan 25 kali *tapping* secara vertikal. Tugas tersebut antara lain terdapat 4 tingkat kesulitan, yaitu tinggi, sedang, rendah, dan sangat rendah yang ditentukan berdasarkan ketentuan ISO 9241-411. Masing-masing tingkat kesulitan dilakukan 3 kali.

Berdasarkan hasil dari penelitian ini, pengembangan prototipe menggunakan sensor EMG lebih dianjurkan sehingga penelitian ini menggunakan sensor EMG yang berbeda yaitu sensor EMG yang diproduksi oleh Biosignalsplux.



Gambar 2.11 Prototipe Perangkat Penunjuk Terdahulu

Terdapat beberapa penelitian sebelumnya yaitu penelitian menggunakan kombinasi *foot switch* dan *low cost Inertial Measurement Unit (IMU)* untuk tetikus yang dapat dipakai dalam interaksi manusia dan komputer, pada penelitian ini klik yang digunakan adalah menggunakan kaki. Penelitian ini memiliki *throughput* dan *movement time* yang bagus yaitu $TP = 1.37$ bps, namun *device* pada penelitian ini masih memberikan beban yang cukup berat untuk leher pengguna (Widodo, dkk. 2020^a). Penelitian kedua adalah kombinasi IMU dan sensor tekuk sebagai perangkat penunjuk dan metode klik, namun pada penelitian ini hanya menggunakan klik kiri. Penelitian ini memiliki hasil yang lebih baik dengan $TP = 1.75$ bps, namun pada *device* yang digunakan juga masih memiliki kekurangan pada kenyamanan saat penggunaan (Widodo, dkk. 2020^b). Yang terakhir adalah studi tentang tetikus menggunakan lengan atas yang berbasis android dan ANN yang menggunakan gerakan tangan sebagai penanda klik. Pada penelitian ini menggunakan dua *gesture*

untuk dibandingkan satu sama lain dan mendapatkan hasil *throughput* and *movement time* 0.19 bps and 22.18 s untuk *gesture* 1 dan 0.19 bps and 22.66 s untuk *gesture* 2. Dengan membandingkan kedua *gesture* tersebut diperoleh bahwa *gesture* 1 lebih nyaman, memiliki tingkat kelelahan yang rendah, dan lebih efektif dibandingkan *gesture* 2 (Sugihono, 2018). Dari penelitian tersebut masih belum mencakup fitur klik kanan pada mouse yang merupakan salah satu fitur yang penting dalam pengoperasian komputer.

Metode yang digunakan pada penelitian ini berdasarkan dari penelitian yang membandingkan *Non-linear Logistic Regression* (NLR), *Multi-Layer Perceptron* (MLP), *Support Vector Machine* (SVM), dan *Linear Discriminant Analysis* (LNA) menggunakan sensor EMG. Berdasarkan dari penelitian tersebut metode SVM merupakan metode yang mengeluarkan akurasi paling tinggi, lalu MLP dan terakhir adalah NLP (Bellingegni *et al*, 2017).

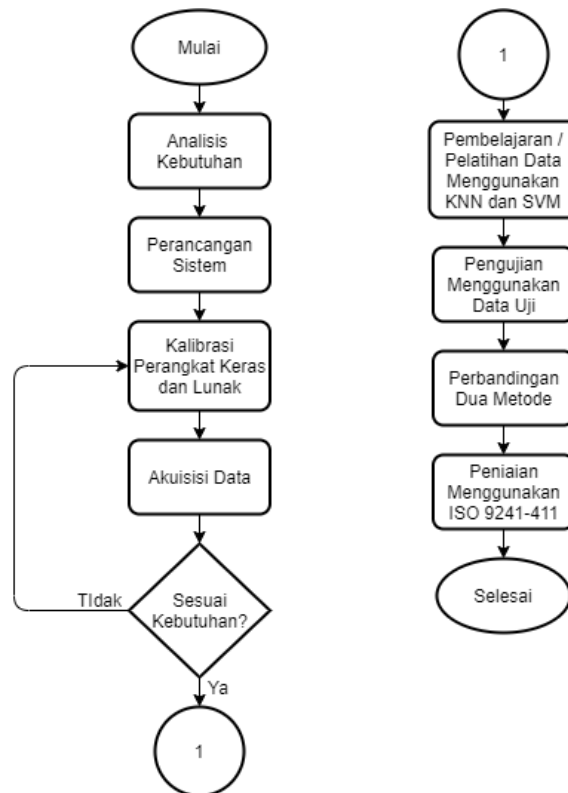
Penelitian lain yang mendasari pemilihan metode adalah berasal dari penelitian yang dilakukan Paul *et al.* (2017) yaitu analisi membandingkan klasifikasi menggunakan SVM dan K-NN menggunakan sensor EMG untuk klasifikasi sinyal EMG pada *Elementary Time Domain Features*. Pada penelitian ini juga diperoleh hasil dengan akurasi tertinggi adalah menggunakan SVM dengan kernel linear dibandingkan dengan metode K-NN.

Bab III

Analisis dan Perancangan Sistem

3.1 Tahapan Penelitian

Tahapan penelitian dalam perancangan sistem kecerdasan buatan untuk mendeteksi klik tercantum pada gambar 3.1 sebagai berikut.



Gambar 3.1 Tahapan Perancangan Kecerdasan Buatan

Tahapan awal penelitian dimulai dengan menemukan permasalahan yang ada. Dilanjutkan dengan analisis kebutuhan yang diperlukan dalam menentukan alat, bahan, serta hal-hal lainnya yang berkaitan dengan proses perancangan identifikasi ini. Berikutnya adalah tahapan perancangan sistem yang berguna untuk membuat sistem yang efektif serta efisien. Kemudian, setelah sistem berhasil dirancang, dilakukan kalibrasi perangkat keras serta perangkat lunak untuk menentukan titik awal cursor dan agar klik dapat berjalan dengan standar yang ada. Setelah proses kalibrasi, dilakukan proses akuisisi data untuk mendapatkan data

hasil pembacaan sensor EMG sebagai masukan untuk mendeteksi klik kiri dan klik kanan. Kemudian dilakukan pengecekan apakah sudah sesuai dengan kebutuhan, jika belum maka akan kembali pada proses kalibrasi perangkat keras dan perangkat lunak serta akuisisi data. Jika sudah sesuai kebutuhan, maka dilakukan pembelajaran data latih. Pembelajaran data latih pada penelitian ini terbagi menjadi 2 pembelajaran, yaitu pembelajaran menggunakan metode K-NN dan pembelajaran menggunakan metode SVM. Setelah masing-masing pembelajaran tersebut, akan dilakukan pengujian menggunakan data latih dan data uji untuk memperoleh akurasi dari hasil pembelajaran masing-masing metode. Kemudian hasil dari masing-masing metode akan dibandingkan untuk mendapatkan hasil metode pembelajaran mana yang lebih baik.

3.2 Analisis Kebutuhan

Seperti yang telah dijelaskan pada pemaparan sebelumnya, penderita tuna daksa, orang yang mengalami kecelakaan, atau kondisi lainnya yang memungkinkan seseorang tidak dapat mengoperasikan komputer terutama mengoperasikan perangkat penunjuk dalam posisi normal membutuhkan suatu alat yang dapat membantu agar dapat mengoperasikan perangkat penunjuk dengan normal. Namun, pada penelitian ini dikhususkan hanya untuk mengembangkan pada bagian klik kiri dan klik kanan saja, tidak untuk menggerakkan kursor menggunakan perangkat penunjuk.

Perangkat yang digunakan untuk mampu mendeteksi kontraksi otot adalah sensor EMG milik Biosignalsplux. Akan ada tiga elektroda yang dipasangkan pada bagian lengan atas pengguna. Satu elektroda berperan sebagai *reference* atau *ground* yang akan dipasang pada bagian bahu sebelah kiri dan dua elektroda lainnya diletakkan pada bagian otot bisep pada lengan sebelah kiri. Data akan dibaca menggunakan aplikasi dari Biosignalsplux yang bernama OpenSignals (r)evolution. Data yang dibaca secara langsung (*real-time*) pada aplikasi ini berbentuk sinyal sesuai dengan kontraksi otot, kemudian disimpan dalam bentuk .txt yang dapat diubah menjadi bentuk .csv. Data .csv itu nantinya akan dijadikan masukan pada dua metode, yaitu K-NN dan SVM yang memiliki keluaran terdeteksi klik kiri atau klik kanan. Setelah dilakukan pembelajaran dengan masing-

masing metode akan dibandingkan metode mana yang lebih baik. Keluaran akan ditampilkan dalam layar komputer dalam bentuk klik.

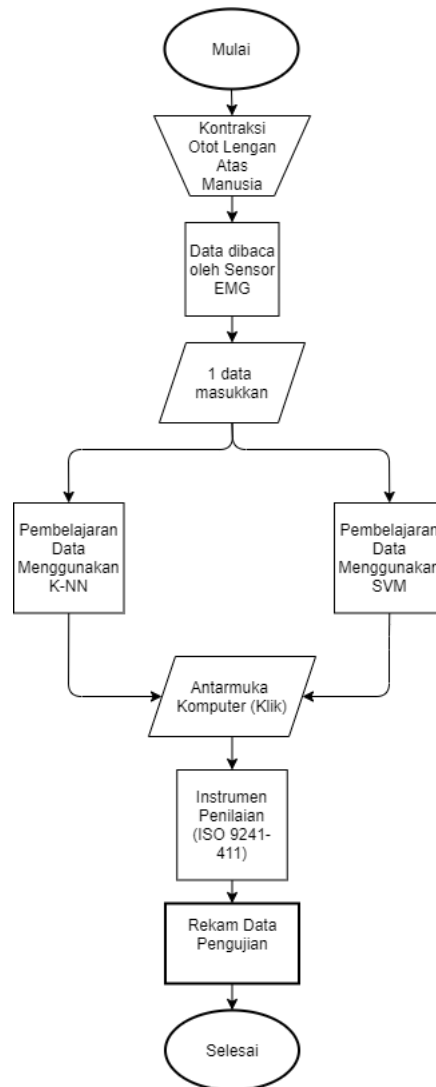
3.3 Desain Sistem

Desain kinerja sistem dapat dijabarkan pada alur yang akan dimuat pada gambar 3.2. Spesifikasi komputer yang digunakan dalam perancangan dan penggunaan perangkat dalam penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3.1 Spesifikasi Komputer

Komponen	Keterangan
Sistem operasi	Windows 10 Home 64-bit
Prosesor	Intel(R) Core(TM) i5-7200U CPU @3.50GHz 2.71GHz
Memori	8.00 GB
Resolusi layar	1366x768

Seperti yang ada pada diagram alur pada gambar 3.2, alur sistem dimulai dari kontraksi otot pada lengan atas manusia untuk menunjukkan klik kiri atau klik kanan. Kemudian tiga elektroda akan membaca data dan mengirim data ke sensor EMG yang data itu akan diterima oleh Hub dan dikirimkan ke komputer menggunakan Bluetooth. Sensor akan membaca data secara *real-time*. Data yang didapat akan diproses oleh K-NN dan SVM untuk diidentifikasi dan diklasifikasikan ke dalam klik kiri dan klik kanan. Setelah itu akan ditampilkan dalam layar dalam bentuk klik. Ketika sudah dapat melakukan klik, komputer akan menampilkan posisi penunjuk di layar yang dapat digunakan untuk menjalankan pengujian prototipe dengan instrument penilaian ISO 9241-411. Sensor EMG hanya akan berfungsi sebagai klik saja, untuk menggerakkan kursor diperlukan alat lainnya yaitu tetikus. Pengujian yang dilakukan akan menghasilkan rekam data yang berisi data akurasi, galat, dan lain sebagainya yang akan digunakan untuk pengolahan data statistic sebagai tolak ukur tingkat keberhasilan prototipe.



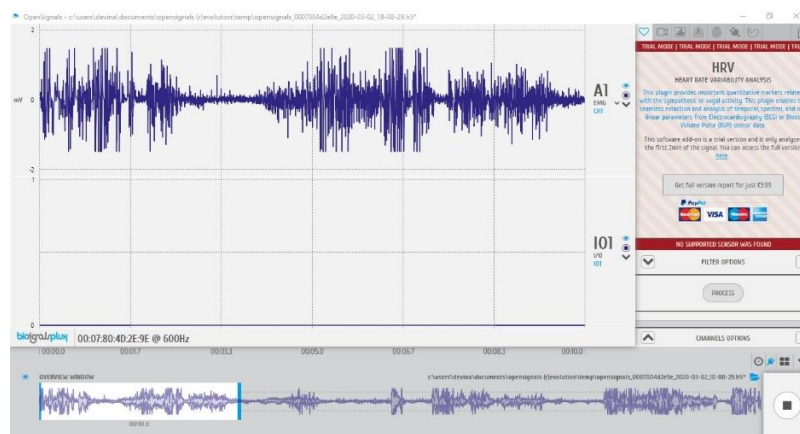
Gambar 3.2 Alur Desain Kinerja Sistem Perangkat

3.3.1 Kontraksi Otot Lengan Atas Manusia

Seperti yang sudah dipaparkan dalam bab 2, kontraksi otot pada lengan atas dibutuhkan untuk pembacaan sensor EMG agar dapat melakukan klik. Pengguna dapat melakukan klik pada saat otot bicep berkontraksi. Klik kiri dapat dilakukan ketika otot bicep berkontraksi sekali, dan klik kanan ketika otot bicep berkontraksi dua kali dalam waktu yang singkat. Pada penelitian ini lengan yang digunakan adalah lengan kiri atas saja karena keterbatasan alat, dan juga masih terbatas pada melakukan klik kiri serta klik kanan.

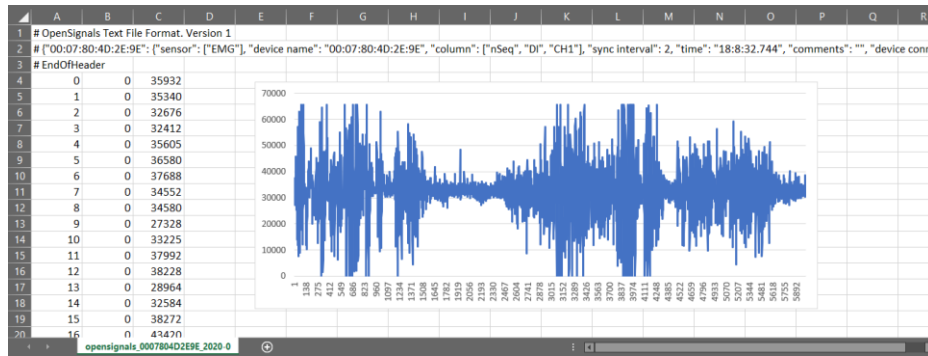
3.3.2 Pembacaan Data Oleh Sensor

Pembacaan data oleh sensor EMG dibantu dengan aplikasi OpenSignals (r)evolution. Hasil yang ditangkap oleh aplikasi OpenSignals (r)evolution berbentuk sinyal atau grafik seperti pada gambar 3.3. Seperti yang sudah dipaparkan dalam bab 2, aplikasi ini dapat menyimpan ke data dalam tiga format. Namun yang digunakan pada penelitian ini hanya format .h5 dan .txt saja. Untuk format .h5 dapat dibaca melalui aplikasi Opensignals (r)evolution seperti pada gambar 3.3.



Gambar 3.3 Grafik tampilan hasil pembacaan data oleh sensor EMG pada aplikasi OpenSignals(r)evolution

Sedangkan untuk format .txt jika dibaca dalam Microsoft Excel maka akan seperti pada gambar 3.4. Terdapat 3 kolom tabel, dari sebelah kiri yaitu pertama, nSeq yang menunjukkan berapa lama data tersebut terekam dikali oleh *sampling rate*. Kedua, *Digital IO* menunjukkan apakah dalam rekaman data ini menggunakan alat lain seperti *handheld switch* atau *foot switch* dalam merekam data. Jika memakai alat lain dan saat alat itu aktif maka *Digital IO* akan bernilai 1, namun jika tidak ada maka *Digital IO* akan bernilai 0. Ketiga, CH1 menunjukkan data yang diterima dari *channel 1*, yang pada penelitian menggunakan sensor EMG. Jika data pada CH1 dijadikan dalam bentuk grafik maka akan memiliki bentuk yang sama seperti pada aplikasi OpenSignals (r)evolution. Data inilah yang nantinya akan diolah untuk pendeteksi klik.

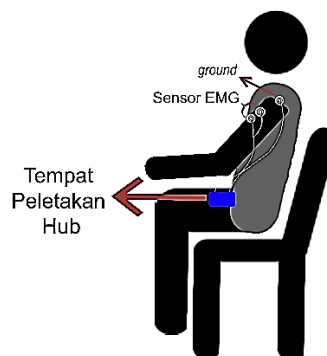


Gambar 3.4 Grafik hasil pembacaan sensor pada excel

3.3.3 Sensor EMG Biosignalsplux

Sensor EMG Biosignalsplux terdiri dari sensor EMG dan Hub yang berfungsi sebagai penghubung ke aplikasi Opensignals (r)evolution. Pada sensor EMG terdapat dua masukan yaitu positif dan negatif yang nantinya akan dihubungkan dengan elektroda. Sedangkan untuk *ground* tidak terpasang pada sensor EMG secara langsung. Secara *default*, resolusi yang biasa digunakan adalah 16 bit, dan dengan sampling rate 600 Hz.

Sensor EMG pada penelitian ini akan diletakkan pada otot bisep lengan kiri bagian atas. Hub yang menghubungkan sensor EMG dengan aplikasi akan diletakkan pada kantong yang nantinya kantong tersebut dapat dipasang pada celana pengguna. Ilustrasi pemasangan perangkat dapat dilihat pada gambar 3.5



Gambar 3.5 Ilustrasi pemasangan sensor EMG dan Hub

3.3.4 Feature Extraction Data

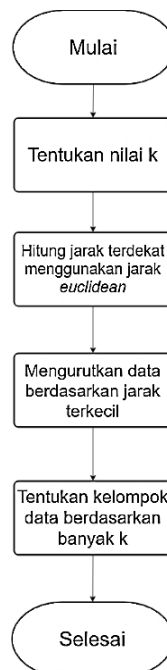
Data EMG yang terakusisi dari sensor Biosignalsplux akan diolah terlebih dahulu untuk mendapatkan fitur-fitur yang dibutuhkan demi kebutuhan *training* model. Proses awal yang dilakukan adalah mengolah hasil EMG dengan rumus

yang telah ditentukan sehingga dapat memperoleh data fitur *wave length*, *mean absolute deviation*, *min value*, dan *max value*. Keempat fitur tersebut yang akan digunakan sebagai masukan *training* model.

Setelah keempat fitur tersebut berhasil diambil, data masih memerlukan proses pengolahan agar dapat menjadi data yang lebih layak untuk *training* model. Data awal yang akan didapatkan memiliki berbagai varian *range* yang berbeda-beda, terdapat angka kecil dan besar diantara fitur tersebut dan hal tersebut dapat menyebabkan hasil dari salah satu fitur terlebih yang memiliki *range* yang terlalu kecil dari yang lain tidak akan berdampak sama dengan fitur lain yang memiliki *range* yang lebih besar. Maka dari itu dibutuhkan proses normalisasi untuk membuat data lebih terdistribusi dengan baik dan normalisasi yang digunakan untuk data tersebut adalah normalisasi *Min Max Scaler*. Normalisasi ini berguna untuk mentransformasi data pada *range* 0-1 sehingga setiap fitur bisa berkontribusi sama untuk *training* model yang akan dilakukan.

3.3.5 Pembelajaran pada K-NN

Diagram alur pembelajaran K-NN pada penelitian ini dilihat pada gambar 3.6 dibawah ini.

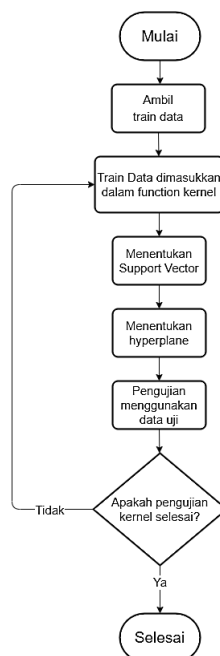


Gambar 3.6 Diagram alur K-NN

Proses pembelajaran K-NN yang digunakan pada penelitian ini dilakukan dengan pertama menentukan nilai k . Seperti yang sudah dijelaskan pada bab 2, nilai k harus bernilai ganjil. Biasanya penentuan nilai k di mulai dengan nilai 3, 5, 7, dan seterusnya. Kemudian, menghitung jarak terdekat data baru dengan data yang sudah ada menggunakan jarak *Euclidean*. Hasil dari setiap jarak tersebut diurutkan berdasarkan jarak terdekat. Setelah mengurutkan, maka akan menentukan kelompok data berdasarkan banyaknya k . Contohnya, jika $k = 3$, maka akan memilih 3 jarak terdekat. Dari ketiga jarak tersebut nantinya akan mendapatkan kesimpulan data baru tersebut masuk dalam kelompok mana. Setiap selesai mendapatkan hasil, maka dilakukan proses penghitungan akurasi menggunakan *confusion matrix*. Proses ini terus diulang dengan mengganti k hingga mendapatkan akurasi yang maksimal.

3.3.6 Pembelajaran pada SVM

Diagram alur pembelajaran SVM pada penelitian ini dilihat pada gambar 3.7 dibawah ini.



Gambar 3.7 Diagram Alur SVM

Langkah awal yang digunakan dalam proses SVM adalah *train* data. *Train* data dilakukan dengan data yang ada yaitu seperti yang sudah dijelaskan sebelumnya. Akan ada sepuluh responden dengan melakukan tes sebanyak lima

kali. Pengenalan dilakukan dengan cara jika terjadi kontraksi sekali maka akan dikelompokkan dalam klik kiri, dan jika kontraksi terjadi dua kali maka akan dikelompokkan dalam klik kanan. Setelah *train* data, maka langkah selanjutnya adalah menentukan *support vector*. Setelah mendapatkan *support vector*, maka akan dilakukan penentuan letak *hyperplane*. Kemudian akan dilakukan pengujian data dengan data baru yang tidak dijadikan data latih hingga mendapatkan hasil klasifikasi data uji. Hasil klasifikasi akan dicek apakah hasil tersebut termasuk dalam *support vector*. Jika ya, maka akan dilakukan penentuan *hyperplane* ulang, dan jika tidak, maka akan dilanjutkan dengan uji kernel SVM dan setelah itu proses pembelajaran SVM selesai.

3.4 Implementasi Sistem

Tahap implementasi sistem merupakan tahap lanjutan dari diagram alur K-NN dan SVM. Implementasi yang dilakukan adalah membuat sistem pembelajaran dengan data latih berjumlah 2 jenis klik yaitu klik kiri dan klik kanan dengan masing-masing memiliki 50 sampel. Sampel tersebut didapatkan dari 10 responden yang akan dites untuk melakukan kontraksi otot untuk klik kiri dan klik kanan. Tiap-tiap tes akan dilakukan sebanyak 5 kali. Bahasa pemrograman yang digunakan adalah Python untuk mengklasifikasikan jenis klik. Setelah dilakukan pembelajaran dengan masing-masing metode, akan dilakukan proses pengenalan dengan menggunakan data uji yang telah disiapkan yaitu 10 responden dengan masing-masing tes dilakukan 5 kali yang tidak digunakan pada data latih pembelajaran. Sehingga total data yang digunakan sebanyak 100 data.

Pengambilan data akan dilakukan cara memasang sensor EMG pada lengan kiri bagian atas, lebih tepatnya pada otot bisep pengguna. Jika pengguna melakukan kontraksi sekali, maka akan terjadi klik kiri, dan jika pengguna melakukan kontraksi dua kali, maka akan terjadi klik kanan. Nilai-nilai yang didapat dari sensor EMG melalui aplikasi OpenSignals (r)evolution akan menjadi masukkan dalam pembelajaran metode K-NN dan SVM. Hasil dari pembelajaran dan pengujian akan ditampilkan dalam Tkinter. Dalam aplikasi ini akan terdapat sebuah kotak yang berwarna oranye yang jika pengguna mengarahkan penunjuk ke kotak tersebut dan melakukan kontraksi sekali yang diklasifikasikan klik kiri, maka kotak

tersebut akan berubah warna menjadi merah. Sedangkan jika pengguna melakukan kontraksi sebanyak dua kali, maka kotak tersebut akan berubah warna menjadi warna biru yang artinya klik kanan. Aplikasi ini juga menampilkan hasil klasifikasi, akurasi, serta *error* yang ada. Tampilan antar muka aplikasi dapat dilihat dalam *mockup* pada gambar 3.7.



Gambar 3.7 *Mockup* tampilan aplikasi pengujian

3.5 Rancangan Pengujian

Pengujian hasil pembelajaran metode K-NN dan SVM ini dilakukan dengan cara melakukan pengujian menggunakan *confusion matrix*. Dilihat dari hasil akurasi tiap-tiap metode. Kedua metode ini juga akan dibandingkan mana yang memiliki nilai akurasi yang lebih tinggi dan error yang lebih kecil serta lebih baik dalam melakukan klasifikasi. Kemudian hasil dari penelitian ini akan dibandingkan dengan penelitian sebelumnya yang memiliki alat yang berbeda, namun dengan metode penghitungan akurasi yang sama.

Pengujian akurasi dilakukan dengan menggunakan rumus *accuracy*, *precision*, *recall* dan *f-measure*. Berikut merupakan rumus dari *accuracy*, *precision*, *recall* dan *f-measure*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Di mana:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

$$Precision = \frac{\sum_{i=1}^I TP_i}{\sum_{i=1}^I (FP_i + TP_i)} * 100\%$$

Di mana:

$TP = True\ Positive$

$FP = False\ Positive$

$I = Jumlah$

$$Recall = \frac{\sum_{i=1}^I TP_i}{\sum_{i=1}^I (TP_i + FN_i)} * 100\%$$

Di mana:

$TP = True\ Positive$

$FN = False\ Negative$

$I = Jumlah$

$$f - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Bab IV

Hasil dan Pembahasan

4.1 Hasil Pengumpulan Data

Pada subbab ini akan membahas proses pengambilan dan pengolahan data sehingga data dapat terkumpul. Detail dari pengolahan data dari awal sampai ekstraksi fitur juga akan dijelaskan hingga data dapat digunakan menjadi data yang siap digunakan untuk *training*.

4.1.1 Pemasangan Perangkat Sensor Pada Responden

Tahap awal penelitian adalah pengambilan data. Data yang diambil berasal dari sensor EMG yang ditempelkan pada lengan kiri atas responden seperti pada gambar 4.1. Setelah sensor sudah terpasang sesuai dengan prosedur, maka langkah berikutnya adalah menguji apakah sensor sudah terpasang dengan benar menggunakan aplikasi yang sudah disediakan oleh *biosignalplux*. Pengujian dilakukan dengan cara melihat sinyal kontraksi stabil saat tidak ada kontraksi dan disaat responden melakukan kontraksi sensor dapat merespon sesuai dengan kontraksi yang diberikan. Penentuan kestabilan sinyal dapat dilihat dari gambar *output* sinyal yang diperlihatkan oleh aplikasi.



Gambar 4.1 Proses Akuisisi Data

Jumlah responden yang diambil berjumlah 9 orang dengan rentang usia yang cukup dekat dan sebagian besar gender laki-laki seperti yang ditunjukkan pada tabel 4.1. Sebanyak 88% dari responden berjenis kelamin laki-laki dan 12% sisanya

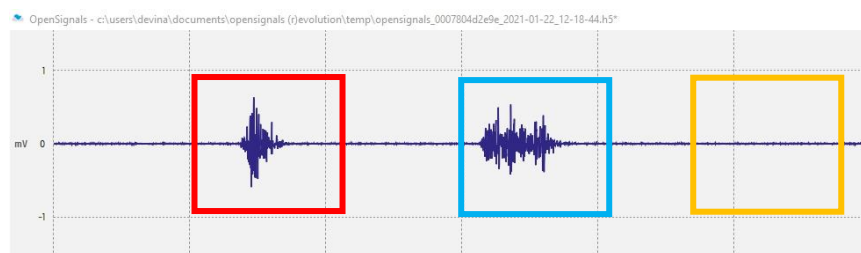
perempuan. Sementara untuk data, rentang usia dari responden ada di 23-27 tahun, berat badan di rentang 48-98 kg dan tinggi badan sekitar 166–178 cm.

Tabel 4.1 Data Responden

Jenis Kelamin	Usia	Berat Badan (kg)	Tinggi Badan (cm)
Laki-Laki	23	60	170
Laki-Laki	25	90	170
Laki-Laki	23	55	169
Laki-Laki	27	98	169
Laki-Laki	23	50	173
Laki-Laki	26	92	178
Laki-Laki	23	65	171
Laki-Laki	24	72	175
Perempuan	23	48	166

4.1.2 Proses Pengambilan Data Responden

Saat perangkat sensor sudah terpasang dengan baik, maka pengambilan data yang sebenarnya dapat dilakukan. Pengambilan *dataset* diambil menggunakan aplikasi yang sudah disediakan oleh *biosignalplex*. Aplikasi tersebut dapat merekam hasil kontraksi yang dihasilkan dari sensor dan hasil yang diperoleh bisa disimpan untuk menjadi *dataset*. Contoh proses pengambilan data dapat dilihat pada gambar 4.2, pada kotak merah adalah kontraksi saat melakukan *left click*, kotak biru kontraksi untuk *right click*, dan kotak kuning sebagai data *no click*.



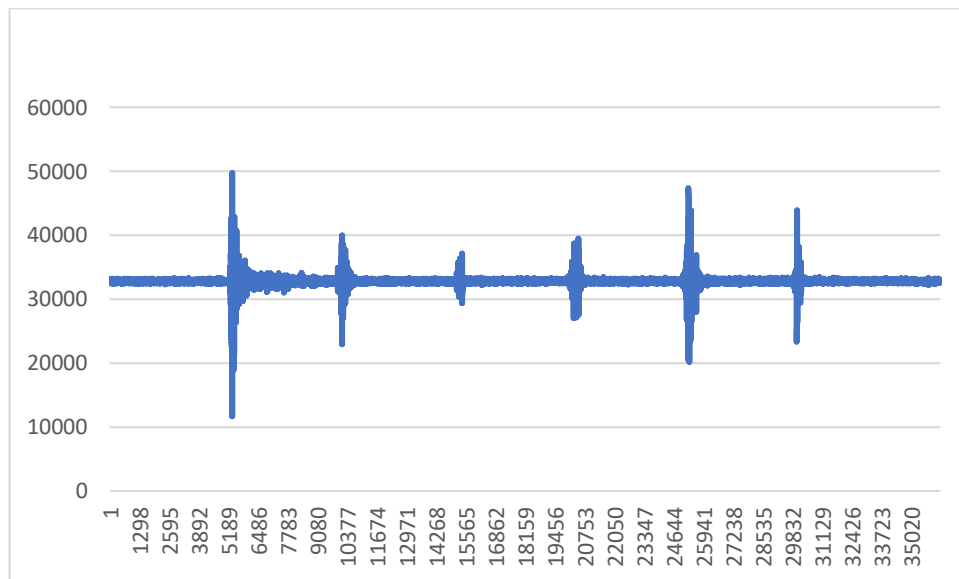
Gambar 4.2 Sinyal Kontraksi Pengambilan Data

Responden akan melakukan rekaman kontraksi sebanyak dua kali, rekaman kontraksi pertama dilakukan dengan kontraksi ringan dan cepat (0.1 – 0.2 detik)

yang akan digunakan sebagai *dataset* kelas *left click* dan rekaman kontraksi yang kedua akan dilakukan dengan kontraksi ringan dan lama (1 detik) yang akan digunakan sebagai *dataset* kelas *right click*. Setiap rekaman akan berdurasi sekitar 35 detik dan responden akan melakukan kontraksi setiap interval waktu 5 detik dimulai dari detik ke 5, sehingga untuk setiap responden akan menghasilkan data sejumlah 12 dengan detail 6 sebagai data *left click* dan 6 sebagai data *right click*. Sementara untuk kelas ketiga atau kelas *no-click* yang akan menggunakan data yang sama dengan data kontraksi *left click* dan *right click*, tetapi data yang diambil adalah disaat responden tidak melakukan kontraksi.

4.1.3 Proses Pengelolaan Dataset

Hasil yang didapatkan dari rekaman kontraksi di-*export* menjadi bentuk *file txt* dan *h5*. Pada *file txt* dapat dilihat hasil data yang dihasilkan seperti yang ditunjukkan oleh tabel 4.2. Terdapat *sequence* atau urutan dari *output* yang dihasilkan dan hasil EMG dalam satuan volt. Jumlah *sequence* yang dihasilkan per responden setiap kelas minimal 35000 dengan hasil EMG pada setiap *sequence*-nya. Total tersebut didapatkan dari lama kontraksi (35 detik) dikali dengan jumlah frekuensi sensor yang dihasilkan perdetik (1000 Hz).



Gambar 4.3 Grafik Fluktuasi Pengambilan Data

Tabel 4.2 Hasil Rekaman Kontraksi Salah Satu Responden

<i>Sequence</i>	EMG (mV)
0	32696
1	32732
2	32640
...	...
30031	42296
30032	43332
30033	43876

Berdasarkan hasil analisis dari data yang didapatkan dapat dilihat bahwa sebelum adanya kontraksi, hasil EMG yang dihasilkan stabil pada angka 32000 dan jika terdapat kontraksi maka hasil EMG akan meningkat kurang lebih diangka 37000–44000 sesuai dengan kuatnya hasil kontraksi otot yang dilakukan oleh responden. Fluktuasi hasil EMG yang disebabkan oleh kontraksi dapat divisualisasikan untuk memperjelas data ketika terjadi kontraksi disaat pengambilan data seperti yang ditunjukkan pada gambar 4.3. Satuan dari hasil EMG adalah millivolt. Angka tersebut didapatkan setelah pengkondisian sinyal yaitu terpengaruh oleh besarnya *resolution* yang terdiri dari pilihan 8 bit atau 16 bit dan *sampling rate* dengan pilihan minimal 10 Hz dan maksimal 4000 Hz pada saat pengambilan sinyal menggunakan aplikasi OpenSignals. Sementara itu setiap responden memiliki rentang angka fluktuasi yang berbeda – beda karena setiap responden juga memiliki massa otot dan lemak yang berbeda. Kedua variabel tersebut memiliki peran yang penting terhadap sensor dan hasil EMG yang diperoleh.

Sementara untuk hasil data *file* h5, *file* tersebut yang akan digunakan sebagai masukan sistem dan diolah sebagai data *training* dan data *testing*. Data h5 dapat dibaca dan diolah lebih cepat oleh komputer. Kegunaan h5 yang difokuskan pada penyimpanan dan pengolahan data oleh komputer menjadi pilihan yang tepat untuk digunakan sebagai sumber data yang akan digunakan dalam pembuatan sistem. Pada sistem ini *file* h5 berukuran rata-rata 500kb untuk tiap *filenya*.

Hasil data yang berhasil didapatkan dari file *h5*, perlu melalui proses *feature extraction* terlebih dahulu untuk mendapatkan fitur-fitur yang dibutuhkan. Fitur seperti *wavelength*, *mean absolute deviation*, *min value*, dan *max value* akan diekstrak dengan rumus masing-masing dalam proses *feature extraction*. Setelah didapatkan keempat fitur tersebut, data akan melalui proses normalisasi dengan metode *min max scaler* agar data dapat terdistribusi dengan baik. Setelah itu data akhir baru dapat digunakan untuk menjadi data *training* dan data *test* untuk model yang digunakan.

4.1.4 Proses *Splitting Data Train dan Data Test*

Jumlah dari seluruh *dataset* tidak seluruhnya akan menjadi *data training*, tetapi sebagian dari *dataset* akan dibagi untuk menjadi *data testing* yang akan digunakan untuk pengujian metode. *Dataset* akan dibagi menjadi 80% sebagai *data training* dan 20% sisanya akan dibuat menjadi *data testing*. Jumlah dari *dataset* yang awal berjumlah 54 data untuk setiap kelas, maka dibagi menjadi 43 data sebagai *data training* dan 11 data sebagai *data testing* untuk setiap kelas yang ada. Jumlah dari data juga dibuat sama untuk memaksimalkan pembelajaran sistem dan nilai akurasi.

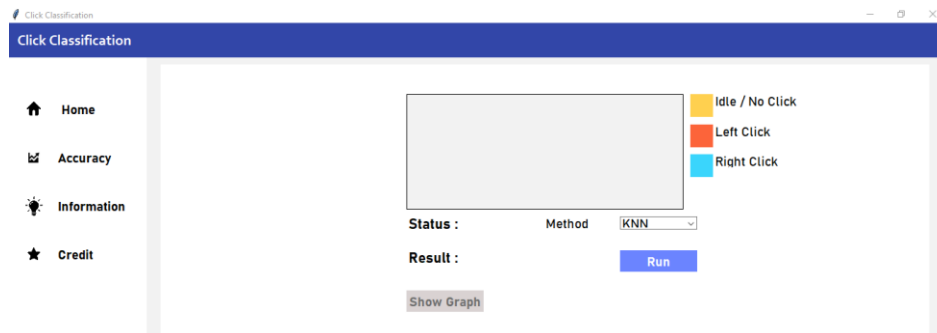
4.2 Hasil Rancangan Aplikasi

Pada subbab ini akan membahas tentang hasil rancangan aplikasi yang telah dibuat. Terdapat empat halaman pada aplikasi ini yaitu, halaman *home*, halaman *accuracy*, halaman *information*, dan halaman *credit*.

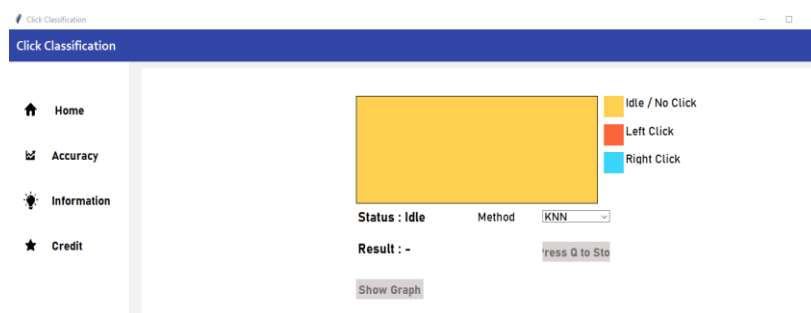
4.2.1 Antarmuka Halaman *Home*

Antarmuka halaman *home* merupakan halaman utama pada aplikasi ini. Halaman ini berfungsi untuk menampilkan hasil dari program pendeteksi klik. Hal pertama yang harus dilakukan adalah memilih metode yang akan diujikan, dengan cara memilih metode yang terdapat pada *combo box*. Terdapat dua metode yang dapat dipilih yaitu K-NN dan SVM. Setelah memilih metode, maka aplikasi dapat dijalankan dengan cara menekan tombol *Run* yang terdapat pada bagian bawah

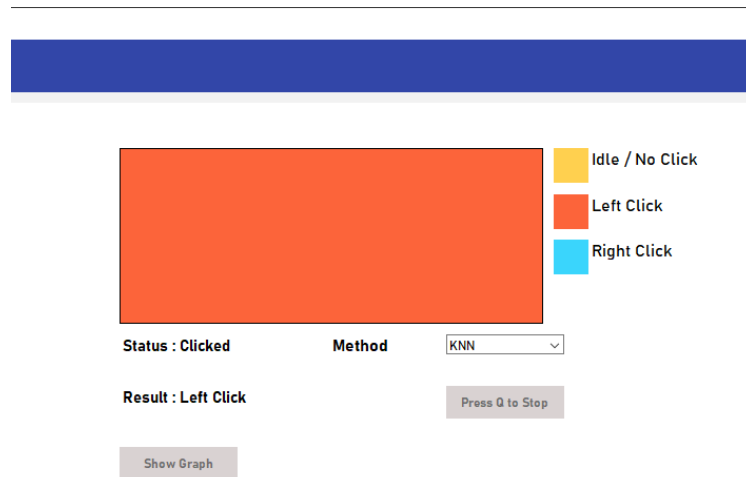
combo box metode. Terdapat satu *box* yang akan menampilkan satu warna. Warna tersebut menunjukkan status dari aktivitas klik. Pada awalnya *box* akan berwarna kuning yang menunjukkan bahwa sedang tidak ada aktivitas klik, pada *Result* akan menunjukkan tanda “-” seperti yang ditunjukkan pada gambar 4.4. Setelah program dijalankan untuk klasifikasi klik maka status awal yang diberikan adalah status *idle*. Status *Idle* merupakan status awal ketika aplikasi dijalankan seperti yang terlihat pada gambar 4.5. Sedangkan, jika *box* berwarna merah seperti pada gambar 4.6, hal tersebut menunjukkan bahwa terdapat aktivitas klik yang terklasifikasi sebagai klik kiri, pada Status akan menunjukkan *Clicked* dan *Result* akan menunjukkan *Left Click*. Warna terakhir yaitu warna biru seperti pada gambar 4.7, menunjukkan bahwa terdapat aktivitas klik yang terklasifikasikan sebagai klik kanan, maka status akan menunjukkan *Clicked* dan *Result* akan menunjukkan *Right Click*. Pada awalnya, tombol yang berlabelkan *Run* akan berganti menjadi label *press Q to stop*. Program dapat dihentikan dengan cara menekan tombol Q pada *keyboard*. Setelah program berhenti, tombol *Show Graph* dapat ditekan untuk menunjukkan rekaman data selama aplikasi dijalankan dalam bentuk grafik.



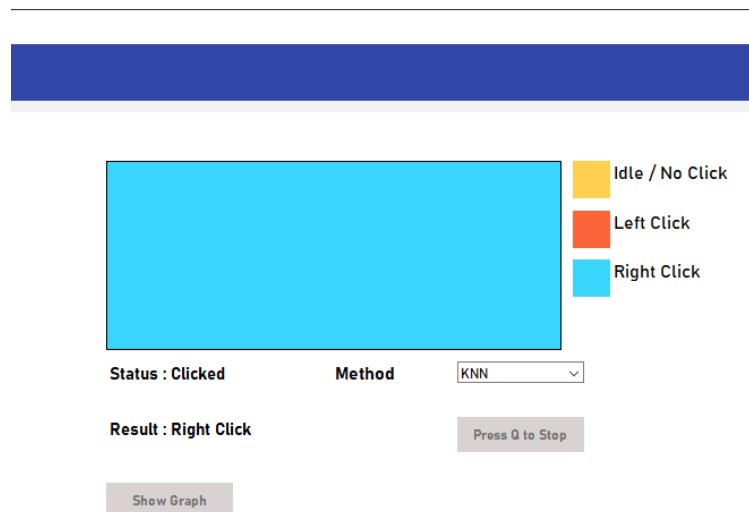
Gambar 4.4 Tampilan Antarmuka Halaman *Home*



Gambar 4.5 Tampilan Halaman Ketika Status *Idle*




Gambar 4.6 Tampilan Ketika Status *Clicked* dan *Result Left Click*




Gambar 4.7 Tampilan Ketika Status *Clicked* dan *Result Right Click*

4.2.2 Antarmuka Halaman *Accuracy*


Pada antarmuka halaman *accuracy* berfungsi untuk menunjukkan akurasi dari aplikasi yang telah dibuat. Seperti yang terlihat pada gambar 4.8 terdapat dua tabel akurasi yaitu akurasi untuk *K-NN Method* dan akurasi *SVM Method*.



Home



Accuracy



Information

Click Classification

Click Classification

KNN Method

Class	Precision	Recall	F1-Score
Idle	0.83	0.91	0.87
Left Click	0.83	0.71	0.77
Right Click	0.78	0.88	0.82

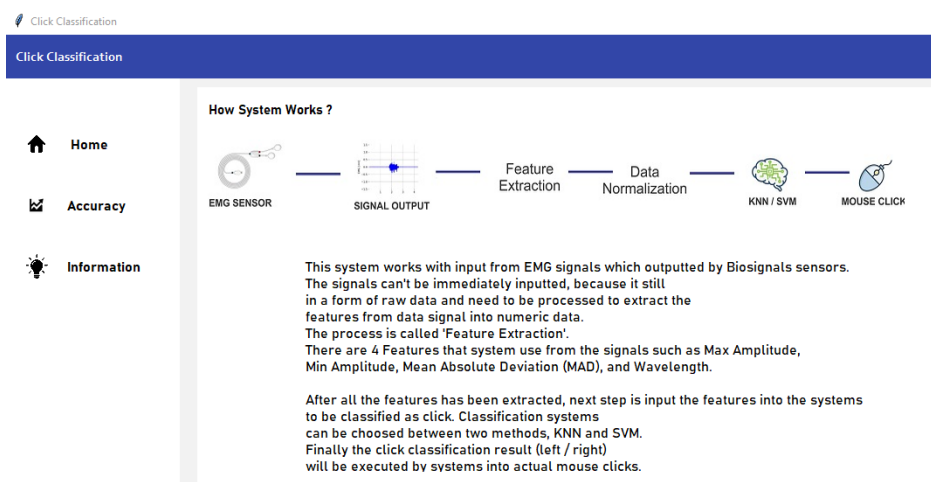
SVM Method

Class	Precision	Recall	F1-Score
Idle	0.77	0.91	0.83
Left Click	0.85	0.79	0.81
Right Click	1.0	0.88	0.93

Gambar 4.8 Tampilan Antarmuka Halaman *Accuracy*

4.2.3 Antarmuka Halaman *Information*

Antarmuka halaman *information* berisi informasi mengenai cara kerja aplikasi seperti yang ditunjukkan oleh gambar 4.9. Mulai dari sensor EMG merekam data mentah, kemudian data mentah diproses dengan fitur ekstraksi yang terdiri dari 4 fitur yaitu *Max Amplitude*, *Min Amplitude*, *Mean Absolute Deviation (MAD)*, dan *Wavelength*. Setelah semua fitur di ekstraksi, selanjutnya adalah memasukkan fitur ke sistem yang akan diklasifikasikan sebagai klik. Terdapat dua metode yang dapat dipilih untuk sistem klasifikasi yaitu K-NN dan SVM. Pada tahap akhir yaitu hasil klasifikasi klik akan dieksekusi oleh sistem menjadi klik tetikus yang sebenarnya.



Gambar 4.9 Tampilan Antarmuka Halaman *Information*

4.3 Hasil Pengujian Sistem

Pada subbab ini akan membahas mengenai hasil pengujian dari program yang telah dibuat. Terdapat dua hasil pengujian yaitu dengan metode K-NN dan metode SVM.

4.3.1 Hasil Pengujian K-NN

Pengujian menggunakan metode K-NN dilakukan dengan cara mencoba parameter k . Parameter k ini bernilai angka acak ganjil dari rentang nilai 3 sampai dengan 11. Setiap nilai parameter k diuji untuk mendapatkan nilai *precision*, *recall*, *f1-score*, dan juga akurasi. Berikut merupakan hasil pengujian dari tiap-tiap nilai parameter k .

Percobaan parameter k yang pertama bernilai 3, mendapatkan nilai yang tertera pada tabel 4.3. Nilai akurasi yang didapatkan dari $k = 3$ adalah 81,81%.

Tabel 4.3 Hasil Pengujian KNN dengan Parameter $k = 3$

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,83	0,91	0,87
<i>Left Click</i>	0,83	0,71	0,77
<i>Right Click</i>	0,78	0,88	0,82

Percobaan kedua adalah dengan parameter k bernilai 5, mendapatkan nilai yang tertera pada tabel 4.4. Nilai akurasi yang didapatkan dari $k = 5$ adalah 78,78%.

Tabel 4.4 Hasil Pengujian KNN dengan Parameter $k = 5$

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,82	0,82	0,82
<i>Left Click</i>	0,77	0,71	0,74
<i>Right Click</i>	0,78	0,88	0,82

Percobaan ketiga adalah dengan parameter k bernilai 7, mendapatkan nilai yang tertera pada tabel 4.5. Nilai akurasi yang didapatkan dari $k = 7$ adalah 81,81%.

Tabel 4.5 Hasil Pengujian KNN dengan Parameter $k = 7$

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,82	0,82	0,82
<i>Left Click</i>	0,79	0,79	0,79
<i>Right Click</i>	0,88	0,88	0,88

Percobaan keempat adalah dengan parameter k bernilai 9, mendapatkan nilai yang terta pada tabel 4.6. Nilai akurasi yang didapatkan dari $k = 9$ adalah 81,81%.

Tabel 4.6 Hasil Pengujian KNN dengan Parameter $k = 9$

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,82	0,82	0,82
<i>Left Click</i>	0,79	0,79	0,79
<i>Right Click</i>	0,88	0,88	0,88

Percobaan kelima adalah dengan parameter k bernilai 11, mendapatkan nilai yang terta pada tabel 4.7. Nilai akurasi yang didapatkan dari $k = 11$ adalah 81,81%.

Tabel 4.7 Hasil Pengujian KNN dengan Parameter $k = 11$

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,77	0,91	0,83
<i>Left Click</i>	0,83	0,71	0,77
<i>Right Click</i>	0,88	0,88	0,88

Berdasarkan dari kelima percobaan parameter k yang telah dilakukan dapat disimpulkan bahwa nilai akurasi tertinggi adalah pada saat $k = 3, 7, 9$, dan 11 yaitu 81,81%. Dikarenakan nilai akurasi tertinggi yang didapatkan ada pada empat nilai k , maka akan diambil satu nilai k yang paling optimal adalah dengan menggunakan nilai k terkecil yaitu $k = 3$.

4.3.2 Hasil Pengujian SVM

Pengujian menggunakan metode SVM dilakukan dengan cara mencoba empat jenis kernel. Kernel yang diujikan adalah Kernel Linear, Polinomial, RBF, dan Sigmoid. Berikut merupakan hasil pengujian dari tiap-tiap kernel.

Percobaan pertama adalah menggunakan kernel linear. Pada percobaan ini mendapatkan nilai seperti yang tertera pada tabel 4.8. Nilai akurasi yang didapatkan dari kernel linear adalah 78,78%.

Tabel 4.8 Hasil Pengujian SVM dengan Kernel Linear

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,71	0,91	0,80
<i>Left Click</i>	0,83	0,71	0,77
<i>Right Click</i>	0,86	0,75	0,80

Percobaan kedua adalah menggunakan kernel polynomial. Pada percobaan ini mendapatkan nilai seperti yang tertera pada tabel 4.9. Nilai akurasi yang didapatkan dari kernel polinomial adalah 84,84%.

Tabel 4.9 Hasil Pengujian SVM dengan Kernel Polinomial

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,77	0,91	0,83
<i>Left Click</i>	0,85	0,79	0,81
<i>Right Click</i>	1,00	0,88	0,93

Percobaan ketiga adalah menggunakan kernel RBF. Pada percobaan ini mendapatkan nilai seperti yang tertera pada tabel 4.10. Nilai akurasi yang didapatkan dari kernel RBF adalah 78,78%.

Tabel 4.10 Hasil Pengujian SVM dengan Kernel RBF

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,77	0,91	0,83
<i>Left Click</i>	0,82	0,64	0,72
<i>Right Click</i>	0,78	0,88	0,82

Percobaan ketiga adalah menggunakan kernel sigmoid. Pada percobaan ini mendapatkan nilai seperti yang tertera pada tabel 4.11. Nilai akurasi yang didapatkan dari kernel sigmoid adalah 42,42%.

Tabel 4.11 Hasil Pengujian SVM dengan Kernel Sigmoid

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
<i>No Click</i>	0,52	1,00	0,69
<i>Left Click</i>	0,00	0,00	0,00
<i>Right Click</i>	0,38	0,38	0,38

Berdasarkan dari percobaan keempat kernel yang telah dilakukan dapat disimpulkan bahwa nilai akurasi tertinggi adalah pada saat menggunakan kernel polynomial yaitu dengan nilai akurasi 84,84%.

4.4 Hasil Kuesioner Penggunaan Sistem

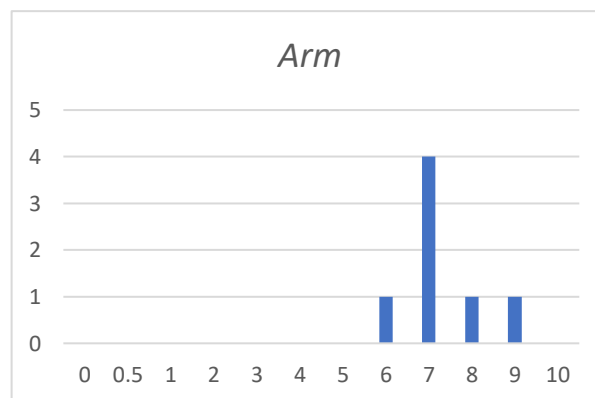
Setelah sistem selesai dirancang dan dibuat, dilakukan uji coba penggunaan sistem terhadap beberapa responden, dalam hal ini responden tersebut adalah kalangan umum yang berusia rata-rata 23 tahun. Uji coba ini dilakukan dalam 2 tahap yaitu uji coba sistem dan pengisian kuesioner sebagai sarana untuk memberi penilaian terhadap sistem dalam segi fungsionalitas. Terdapat 2 kuesioner yang perlu diisi oleh responden yaitu Kuesioner *Borg Scale* ISO 9241-411 dan Instrumentasi Penilaian Bebas ISO 9241-411.

Responden awalnya akan mencoba sistem terlebih dahulu dengan prosedur sesuai yang sudah dijelaskan pada subbab 4.1.2. Setelah itu responden akan mengisi kuesioner yang berisikan penilaian sesuai dengan Format Instrumentasi ISO 9241-411. Pertama responden akan mengisi kuesioner *Borg Scale* yang berisikan 3 kategori yang memiliki rentang penilaian 0 sampai 10. Nilai 0 yaitu *nothing at all* yang berarti tidak memerlukan usaha sama sekali, nilai 0,5 yaitu *very weak (just noticeable)* yang berarti responden merasakan sangat amat sedikit usaha yang dikeluarkan (responden hampir tidak merasakan adanya kontraksi), nilai 1 yaitu *very weak* yang berarti sangat sedikit usaha yang perlu dikeluarkan,

nilai 2 yaitu *weak (light)* yang berarti sedikit usaha yang perlu dikeluarkan, nilai 3 yaitu *moderate* yang berarti memerlukan usaha yang cukup (tidak terlalu besar atau kecil), nilai 4 yaitu *somewhat strong* yang berarti sedikit lebih kuat, nilai 5 sampai 6 yaitu *strong (heavy)* yang berarti memerlukan usaha yang kuat, nilai 7 sampai 9 yaitu *very strong* yang berarti memerlukan usaha yang sangat kuat, dan nilai 10 yaitu *very very strong (almost max.)* yang berarti memerlukan usaha yang sangat amat kuat. Berikut merupakan grafik-grafik yang menunjukkan setiap penilaian dan jawaban dari responden. Penilaian pertama yaitu mengenai usaha pada bagian tangan (*Arm*).

Tabel 4.12 Tabel Hasil Kuesioner Usaha pada Bagian Tangan (*Arm*)

Responden	<i>Arm</i>
Responden 1	6
Responden 2	7
Responden 3	7
Responden 4	9
Responden 5	7
Responden 6	8
Responden 7	7



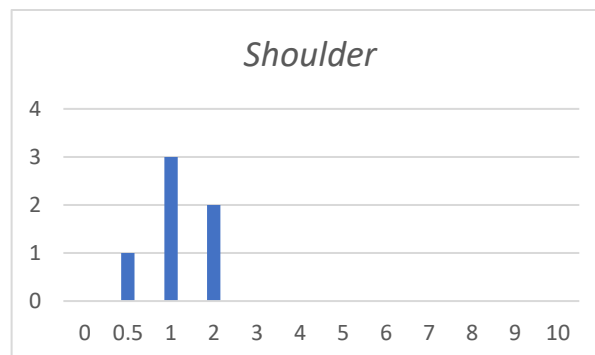
Gambar 4.10 Pilihan Jawaban Responden terhadap Penilaian *Effort* pada *Arm*

Seperti yang ditunjukkan pada tabel 4.12 dan gambar 4.10 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian pertama menunjukkan bahwa terdapat 1 orang yang memilih nilai 6, 4 orang yang memilih

nilai 7, 1 orang yang memilih nilai 8, dan 1 orang yang memilih nilai 9. Berdasarkan seluruh hasil penilaian responden mengenai usaha pada bagian tangan, didapatkan nilai rata-rata 7 yang menunjukkan bahwa pengguna memerlukan usaha yang cukup pada bagian tangan saat melakukan uji coba sistem. Penilaian kedua adalah mengenai usaha pada bagian pundak (*Shoulder*).

Tabel 4.13 Tabel Hasil Kuesioner Usaha pada Bagian Pundak (*Shoulder*)

Responden	<i>Shoulder</i>
Responden 1	2
Responden 2	2
Responden 3	1
Responden 4	2
Responden 5	2
Responden 6	1
Responden 7	0.5

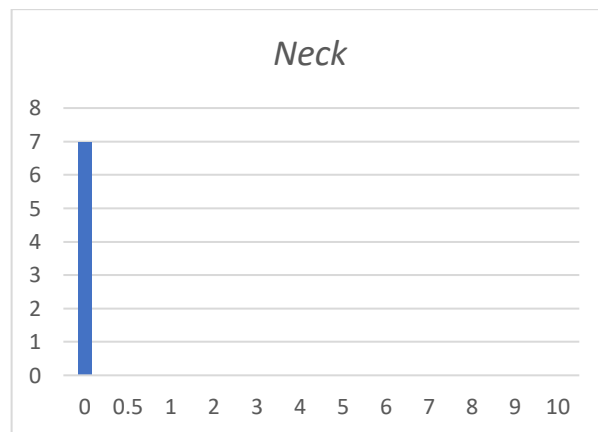


Gambar 4.11 Pilihan Jawaban Responden terhadap Penilaian *Effort* pada *Shoulder*

Seperti yang ditunjukkan pada tabel 4.13 dan gambar 4.11 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kedua menunjukkan bahwa terdapat 1 orang yang memilih nilai 0.5, 3 orang yang memilih nilai 1, dan 2 orang yang memilih nilai 2. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna memerlukan usaha yang sedikit pada bagian pundak saat melakukan uji coba sistem. Penilaian kedua adalah mengenai usaha pada bagian leher (*Neck*).

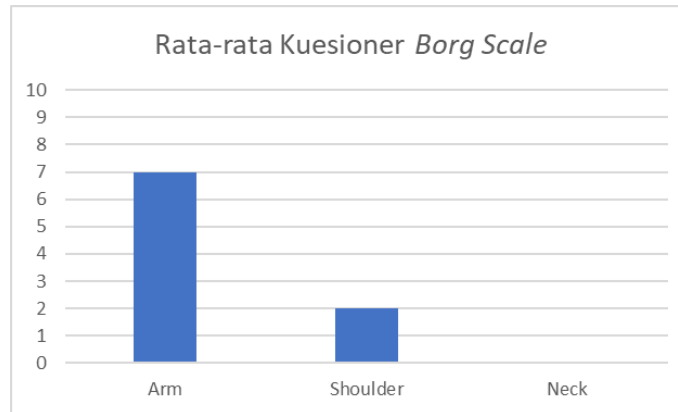
Tabel 4.14 Tabel Hasil Kuesioner Usaha pada Bagian Leher (*Neck*)

Responden	<i>Neck</i>
Responden 1	2
Responden 2	2
Responden 3	1
Responden 4	2
Responden 5	2
Responden 6	1
Responden 7	0.5



Gambar 4.12 Pilihan Jawaban Responden terhadap Penilaian *Effort* pada *Neck*

Seperti yang ditunjukkan pada tabel 4.14 dan gambar 4.12 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian ketiga menunjukkan bahwa terdapat 7 orang yang memilih nilai 0. Berdasarkan seluruh hasil penilaian responden menandakan bahwa tidak adanya usaha pengguna yang perlu dikeluarkan pada bagian leher saat melakukan uji coba sistem.



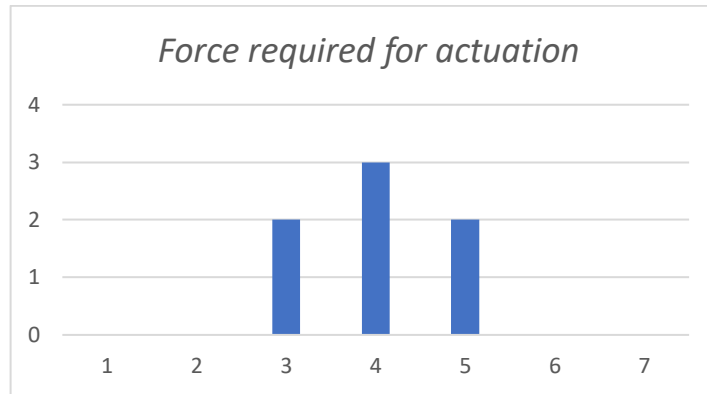
Gambar 4.13 Rata-rata Kuesioner *Borg Scale*

Seperti yang ditunjukkan pada gambar 4.13 yang merupakan rata-rata dari kuesioner *Borg Scale*. Dari 7 responden diperoleh nilai rata-rata 7 pada bagian tangan (*Arm*) yang menunjukkan bahwa membutuhkan usaha yang kuat pada saat melakukan uji coba sistem, sedangkan untuk pada bagian Pundak (*Shoulder*) diperoleh nilai rata-rata 2 yang menunjukkan bahwa membutuhkan usaha yang sangat kecil dalam mengoperasikan sistem pada saat uji coba, dan yang terakhir pada bagian leher (*Neck*) diperoleh rata-rata 0 yang menunjukkan bahwa tidak membutuhkan usaha pada bagian leher pada saat uji coba sistem.

Kuesioner kedua yaitu Instrumentasi Penilaian Bebas yang berisikan 12 penilaian yang memiliki rentang penilaian 1 sampai 7. Berikut merupakan grafik-grafik yang menunjukkan setiap penilaian dan jawaban dari responden. Penilaian pertama yaitu mengenai usaha yang diperlukan untuk membuat alat beroperasi (*force required for actuation*)

Tabel 4.15 Tabel Hasil Kuesioner Usaha yang Diperlukan untuk Membuat Alat Beroperasi (*Force Required for Actuation*)

Responden	<i>Force required for actuation</i>
Responden 1	4
Responden 2	5
Responden 3	3
Responden 4	4
Responden 5	3
Responden 6	5
Responden 7	4

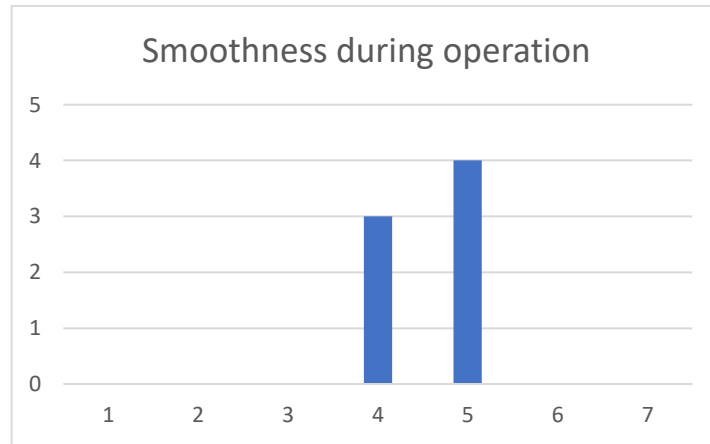


Gambar 4.14 Pilihan Jawaban Responden terhadap Penilaian *force required for actuation*

Seperti yang ditunjukkan pada tabel 4.15 dan gambar 4.14 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian pertama menunjukkan bahwa terdapat 2 orang yang memilih nilai 3, 3 orang yang memilih nilai 4, dan 2 orang yang memilih nilai 5. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa nyaman yang berarti mengeluarkan usaha yang cukup saat melakukan uji coba sistem. Penilaian kedua adalah kelancaran saat menjalankan program (*Smoothness During Operation*).

Tabel 4.16 Tabel Hasil Kuesioner Kelancaran Saat Menjalankan Program (*Smoothness During Operation*)

Responden	<i>Smoothness During Operation</i>
Responden 1	5
Responden 2	5
Responden 3	4
Responden 4	4
Responden 5	5
Responden 6	4
Responden 7	5

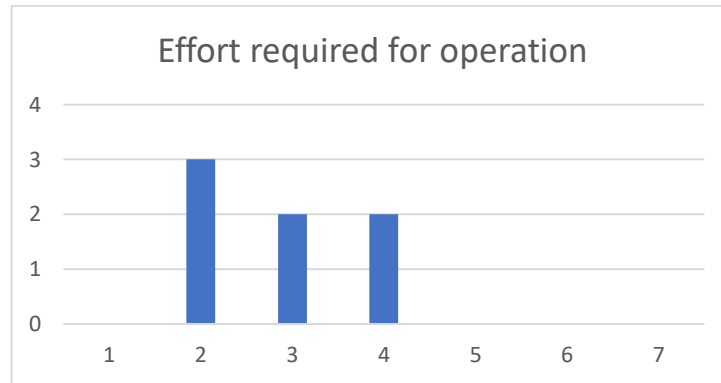


Gambar 4.15 Pilihan Jawaban Responden terhadap Penilaian *Smoothness During Operation*

Seperti yang ditunjukkan pada tabel 4.16 dan gambar 4.15 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kedua menunjukkan bahwa terdapat 3 orang yang memilih nilai 4, dan 4 orang yang memilih nilai 5. Berdasarkan seluruh hasil penilaian responden menandakan bahwa sistem dapat berjalan dengan lancar saat pengguna melakukan uji coba sistem. Penilaian ketiga adalah usaha yang diperlukan untuk melakukan saat uji coba (*Effort Required for Operation*).

Tabel 4.17 Tabel Hasil Kuesioner Usaha yang Diperlukan untuk Melakukan Saat Uji Coba (*Effort Required for Operation*)

Responden	<i>Effort Required for Operation</i>
Responden 1	4
Responden 2	4
Responden 3	2
Responden 4	3
Responden 5	3
Responden 6	2
Responden 7	2

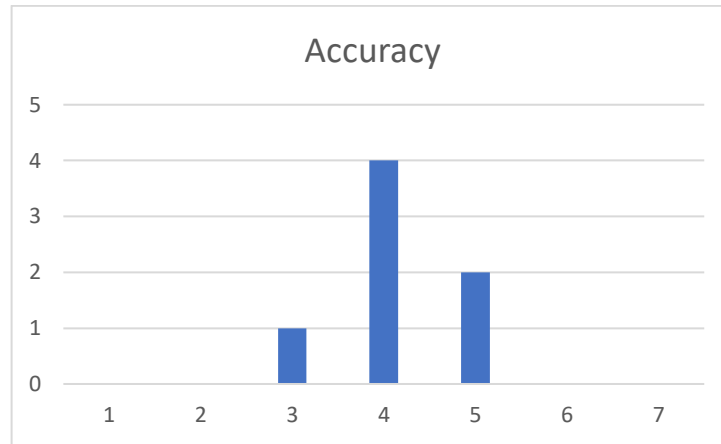


Gambar 4.16 Pilihan Jawaban Responden terhadap Penilaian *Effort Required for Operation*

Seperti yang ditunjukkan pada tabel 4.17 dan gambar 4.16 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian ketiga menunjukkan bahwa terdapat 3 orang yang memilih nilai 2, 2 orang yang memilih nilai 3, dan 2 orang yang memilih nilai 4. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna mengeluarkan usaha yang tinggi saat melakukan uji coba sistem. Penilaian keempat adalah akurasi (*Accuracy*).

Tabel 4.18 Tabel Hasil Kuesioner Akurasi (*Accuracy*)

Responden	<i>Accuracy</i>
Responden 1	5
Responden 2	4
Responden 3	5
Responden 4	3
Responden 5	4
Responden 6	4
Responden 7	4

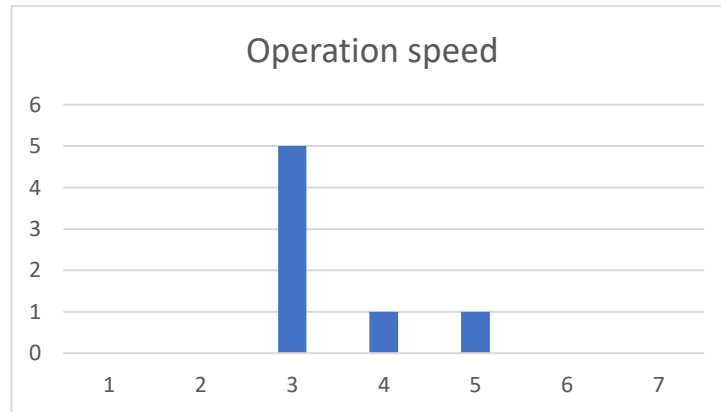


Gambar 4.17 Pilihan Jawaban Responden terhadap Penilaian *Accuracy*

Seperti yang ditunjukkan pada tabel 4.18 dan gambar 4.17 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian keempat menunjukkan bahwa terdapat 1 orang yang memilih nilai 3, 4 orang yang memilih nilai 4, dan 2 orang yang memilih nilai 5. Berdasarkan seluruh hasil penilaian responden menandakan bahwa sistem dapat mendeteksi dengan benar saat melakukan uji coba sistem. Penilaian kelima adalah kecepatan sistem dalam beroperasi (*Operation Speed*).

Tabel 4.19 Tabel Hasil Kuesioner Kecepatan Sistem dalam Beroperasi (*Operation Speed*)

Responden	<i>Operation Speed</i>
Responden 1	4
Responden 2	5
Responden 3	3
Responden 4	3
Responden 5	3
Responden 6	3
Responden 7	3

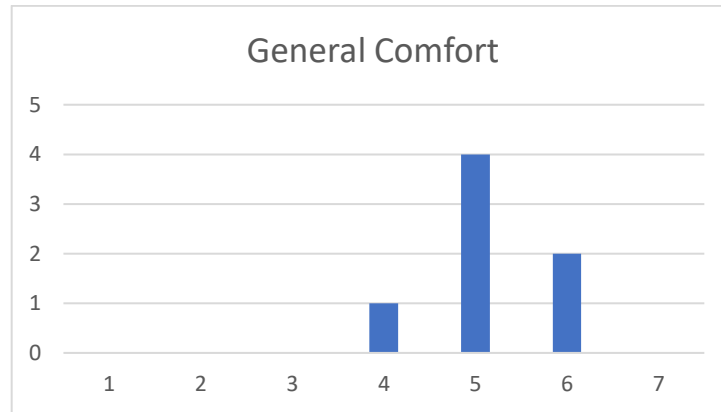


Gambar 4.18 Pilihan Jawaban Responden terhadap Penilaian *Operation Speed*

Seperti yang ditunjukkan pada tabel 4.19 dan gambar 4.18 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kelima menunjukkan bahwa terdapat 5 orang yang memilih nilai 3, 1 orang yang memilih nilai 4, dan 1 orang yang memilih nilai 5. Berdasarkan seluruh hasil penilaian responden menandakan bahwa sistem bekerja kurang cepat saat melakukan uji coba sistem. Penilaian keenam adalah kenyamanan pemakaian alat (*General Comfort*).

Tabel 4.20 Tabel Hasil Kuesioner Kenyamanan Pemakaian Alat (*General Comfort*)

Responden	<i>General Comfort</i>
Responden 1	5
Responden 2	4
Responden 3	5
Responden 4	6
Responden 5	5
Responden 6	5
Responden 7	6

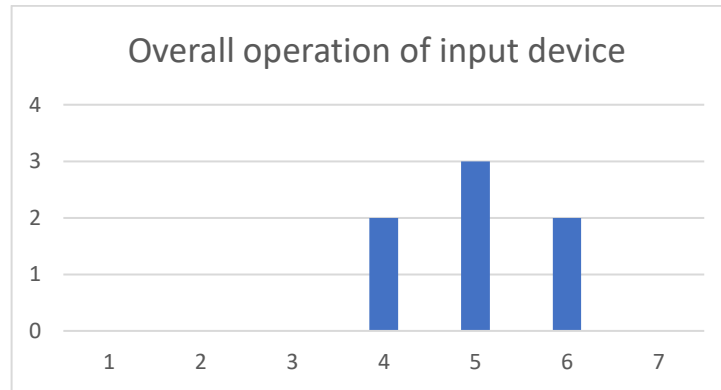


Gambar 4.19 Pilihan Jawaban Responden terhadap Penilaian *General Comfort*

Seperti yang ditunjukkan pada tabel 4.20 dan gambar 4.19 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian keenam menunjukkan bahwa terdapat 1 orang yang memilih nilai 4, 4 orang yang memilih nilai 5, dan 2 orang yang memilih nilai 6. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa nyaman secara keseluruhan saat melakukan uji coba sistem. Penilaian ketujuh adalah tingkat kesulitan pemakaian sensor (*Overall Operation of Input Device*).

Tabel 4.21 Tabel Hasil Kuesioner Tingkat Kesulitan Pemakaian Sensor (*Overall Operation of Input Device*)

Responden	Overall Operation of Input Device
Responden 1	4
Responden 2	5
Responden 3	4
Responden 4	6
Responden 5	6
Responden 6	5
Responden 7	5

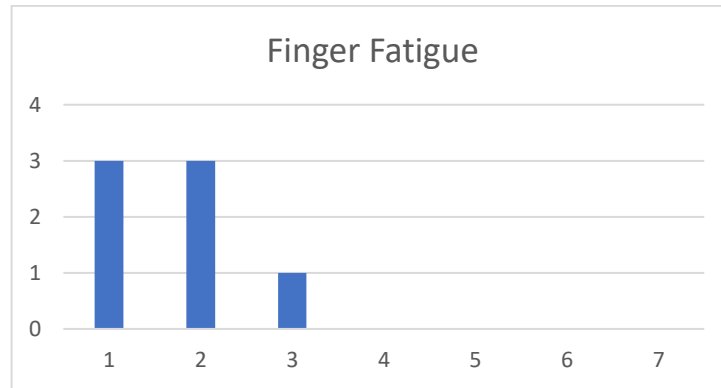


Gambar 4.20 Pilihan Jawaban Responden terhadap Penilaian *Overall Operation of Input Device*

Seperti yang ditunjukkan pada tabel 4.21 dan gambar 4.20 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian ketujuh menunjukkan bahwa terdapat 2 orang yang memilih nilai 4, 3 orang yang memilih nilai 5, dan 2 orang yang memilih nilai 6. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna tidak merasa kesulitan saat melakukan uji coba sistem. Penilaian kedelapan adalah kelelahan pada jari saat menggunakan alat (*Finger Fatigue*).

Tabel 4.22 Tabel Hasil Kuesioner Kelelahan pada Jari Saat Menggunakan Alat (*Finger Fatigue*)

Responden	<i>Finger Fatigue</i>
Responden 1	2
Responden 2	3
Responden 3	2
Responden 4	1
Responden 5	1
Responden 6	2
Responden 7	1

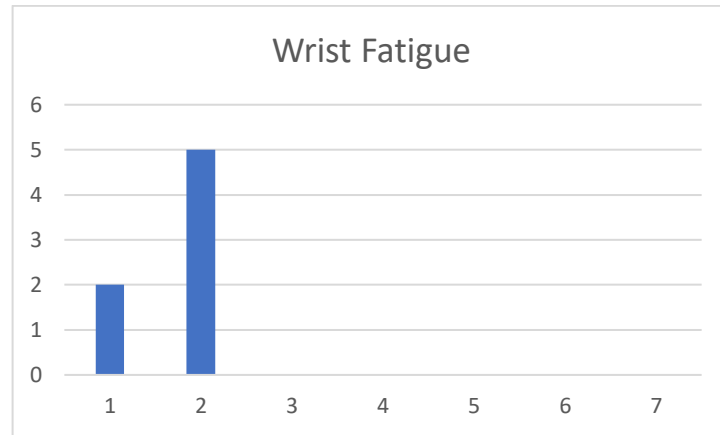


Gambar 4.21 Pilihan Jawaban Responden terhadap Penilaian *Finger Fatigue*

Seperti yang ditunjukkan pada tabel 4.22 dan gambar 4.21 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kedelapan menunjukkan bahwa terdapat 3 orang yang memilih nilai 1, 3 orang yang memilih nilai 2, dan 1 orang yang memilih nilai 3. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa sangat kelelahan pada jari saat melakukan uji coba sistem. Penilaian kesembilan adalah kelelahan pada pergelangan tangan saat menggunakan alat (*Wrist Fatigue*).

Tabel 4.23 Tabel Hasil Kuesioner Kelelahan pada Pergelangan Tangan Saat Menggunakan Alat (*Wrist Fatigue*)

Responden	<i>Wrist Fatigue</i>
Responden 1	2
Responden 2	2
Responden 3	2
Responden 4	2
Responden 5	1
Responden 6	1
Responden 7	2

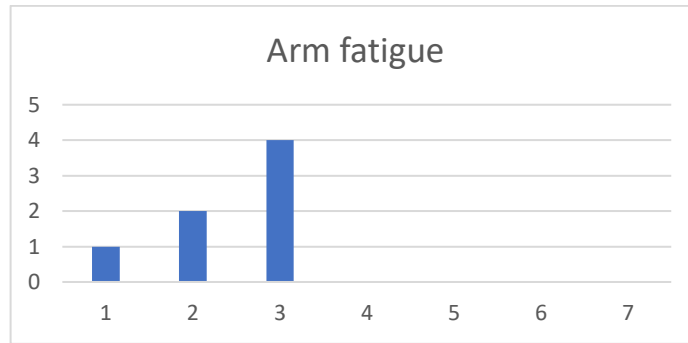


Gambar 4.22 Pilihan Jawaban Responden terhadap Penilaian *Wrist Fatigue*

Seperti yang ditunjukkan pada tabel 4.23 dan gambar 4.22 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kesembilan menunjukkan bahwa terdapat 2 orang yang memilih nilai 1, dan 2 orang yang memilih nilai 2. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa kelelahan pada pergelangan tangan saat melakukan uji coba sistem. Penilaian kesepuluh adalah kelelahan pada tangan saat menggunakan alat (*Arm Fatigue*).

Tabel 4.24 Tabel Hasil Kuesioner Kelelahan pada Tangan Saat Menggunakan Alat (*Arm Fatigue*)

Responden	<i>Arm Fatigue</i>
Responden 1	2
Responden 2	1
Responden 3	3
Responden 4	3
Responden 5	3
Responden 6	3
Responden 7	2

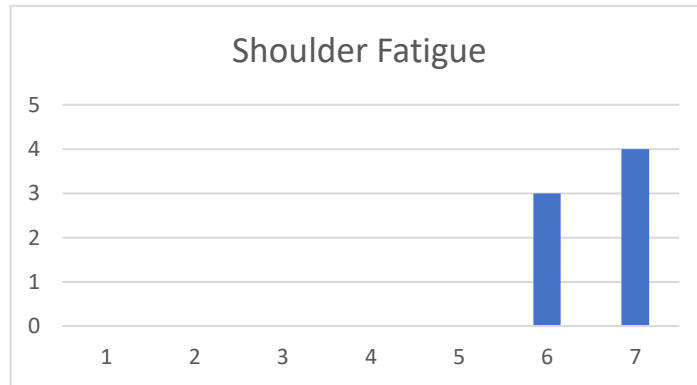


Gambar 4.23 Pilihan Jawaban Responden terhadap Penilaian *Arm Fatigue*

Seperti yang ditunjukkan pada tabel 4.24 dan gambar 4.23 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kesepuluh menunjukkan bahwa terdapat 1 orang yang memilih nilai 1, 2 orang yang memilih nilai 2, dan 4 orang yang memilih nilai 3. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa sedikit kelelahan pada tangan saat melakukan uji coba sistem. Penilaian kesebelas adalah kelelahan pada pundak saat menggunakan alat (*Shoulder Fatigue*).

Tabel 4.25 Tabel Hasil Kuesioner Kelelahan pada Pundak Saat Menggunakan Alat (*Shoulder Fatigue*)

Responden	<i>Shoulder Fatigue</i>
Responden 1	7
Responden 2	6
Responden 3	7
Responden 4	7
Responden 5	6
Responden 6	6
Responden 7	7

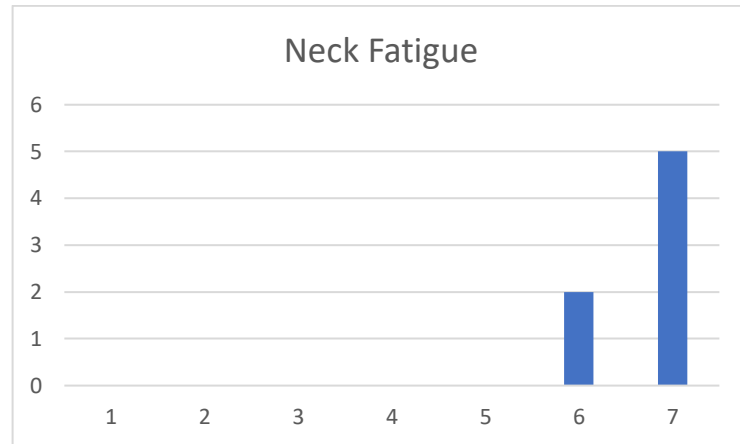


Gambar 4.24 Pilihan Jawaban Responden terhadap Penilaian *Shoulder Fatigue*

Seperti yang ditunjukkan pada tabel 4.25 dan gambar 4.24 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian kesebelas menunjukkan bahwa terdapat 3 orang yang memilih nilai 6, dan 4 orang yang memilih nilai 7. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa tidak kelelahan sama sekali pada tangan saat melakukan uji coba sistem. Penilaian keduabelas adalah kelelahan pada leher saat menggunakan alat (*Neck Fatigue*).

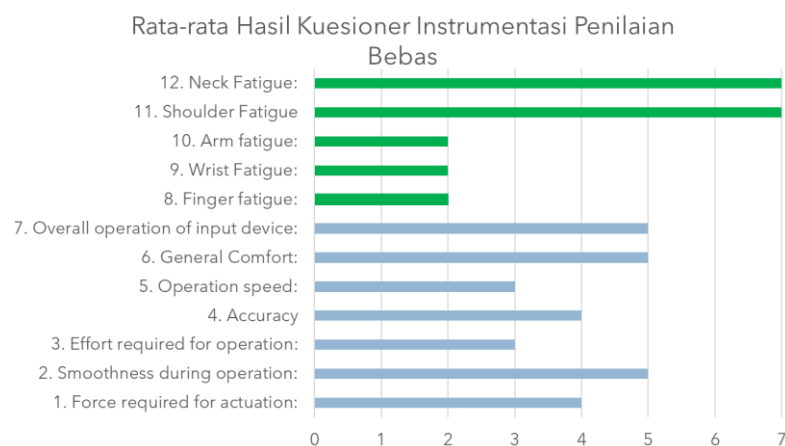
Tabel 4.26 Tabel Hasil Kuesioner Kelelahan pada Leher Saat Menggunakan Alat (*Neck Fatigue*)

Responden	<i>Neck Fatigue</i>
Responden 1	7
Responden 2	7
Responden 3	7
Responden 4	7
Responden 5	6
Responden 6	7
Responden 7	6



Gambar 4.25 Pilihan Jawaban Responden terhadap Penilaian *Neck Fatigue*

Seperti yang ditunjukkan pada tabel 4.26 dan gambar 4.25 yang diambil dari 7 responden. Hasil penilaian melalui kuesioner untuk penilaian keduabelas menunjukkan bahwa terdapat 2 orang yang memilih nilai 6, dan 5 orang yang memilih nilai 7. Berdasarkan seluruh hasil penilaian responden menandakan bahwa pengguna merasa tidak kelelahan sama sekali pada leher saat melakukan uji coba sistem.



Gambar 4.26 Rata-rata Hasil Kuesioner Instrumentasi Penilaian Bebas

Rata-rata instrumentasi penilaian bebas seperti pada gambar 4.27, pertanyaan nomor 1 sampai 7 mengenai performa sistem yang didapatkan dari 7 responden diperoleh rata-rata 4 yang menandakan bahwa sistem bekerja cukup baik, sedangkan pertanyaan nomor 8 sampai 12 yang didapatkan dari 7 responden diperoleh rata-rata nilai 4 yang menandakan bahwa tingkat kelelahan dari penilaian.

4.5 Diskusi Hasil

Setelah pengujian dilakukan terhadap kedua metode (KNN dan SVM), kedua-duanya memiliki akurasi tertinggi yang hampir sama yaitu 81,81% pada KNN dan 84,84% pada SVM. Kedua metode ini bisa dikatakan cukup seimbang dan sama baiknya dalam hal mengklasifikasikan *click* berdasarkan data yang diberikan. Namun SVM memang sedikit lebih unggul terhadap KNN dengan akurasi 84,84% dengan menggunakan kernel polinomial.

Pada pengujian nilai k pada KNN hasil yang didapatkan pada semua pengujian juga hampir sama dan hanya terdapat perbedaan pada $k=5$ yang memiliki akurasi paling rendah yaitu 78.78%. Penurunan tersebut dapat disebabkan dengan adanya satu data uji yang salah atau sulit terklasifikasi saat $k=5$ dan 2 jarak terdekat diantara data uji tersebut adalah kelas yang berbeda sehingga data satu data uji tersebut salah terklasifikasi. Sementara pada $k=3$ akurasi memiliki tinggi yang sama dengan akurasi tertinggi lain dikarenakan satu data dengan jarak terdekat adalah kelas yang benar sehingga data uji yang salah terklasifikasi salah pada $k=5$ dapat terklasifikasi dengan benar pada $k=3$.

Sementara pada pengujian SVM rata-rata akurasi sedikit bervariasi karena rumus yang digunakan pada setiap kernel untuk mengubah dan mengoptimalkan data juga berbeda-beda. Pada metode SVM dari 4 kernel yang diuji, kernel yang memiliki akurasi tertinggi yaitu kernel polinomial yang dapat diartikan bahwa kernel tersebut mampu mengoptimalkan dan mengubah data sehingga sistem dapat menemukan *hyperplane* yang lebih baik untuk mengklasifikasikan *click* daripada kernel lain yang diujikan.

Selain akurasi, *precision*, *recall*, dan *f-measure* juga merupakan angka hasil pengujian yang dilakukan. Dari angka yang didapatkan dari tabel pengujian dapat dilihat bahwa hasil *precision* dan *recall* cukup berbeda-beda setiap kelas. Namun nilai akurasi yang tinggi bisa didapatkan dari hasil *precision* dan *recall* yang tinggi dan seimbang pada setiap kelas dapat diartikan sistem dapat mengklasifikasikan kelas dengan baik dan sistem dapat menemukan klasifikasi kelas tersebut dengan baik juga. Hal tersebut berpengaruh karena hasil uji setiap kelas juga terhubung dan berelasi. Seperti contoh pada tabel 4.9 yang memiliki akurasi tertinggi punya hasil uji *f-measure* yang paling baik pada setiap kelas sehingga dia memiliki akurasi

tertinggi. Sebaliknya pada hasil uji dengan akurasi yang paling rendah dengan akurasi 42.42% yang ditunjukkan pada tabel 4.11, meskipun memiliki satu nilai recall yang tinggi dan bagus pada satu kelas tidak memastikan model tersebut memiliki akurasi yang bagus karena hasil *precision* dan *recall* dari kelas lain tidak juga memiliki hasil yang bagus. Hal tersebut juga dapat dilihat dari *f-measure* tabel tersebut bahwa semua *f-measure* yang didapatkan pada tabel tersebut memiliki nilai yang paling rendah pada setiap kelas diantara semua pengujian pada setiap pengujian.

Tabel 4.27 Perbandingan Nilai Akurasi Metode KNN dan SVM

		<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>	Akurasi
KNN	k=3	0.81	0.83	0.82	81,81%
	k=5	0.79	0.80	0.79	78,78%
	k=7	0.83	0.83	0.83	81,81%
	k=9	0.83	0.83	0.83	81,81%
	k=11	0.83	0.83	0.83	81,81%
SVM	Linear	0.80	0.79	0.79	78,78%
	RBF	0.79	0.81	0.79	78,78%
	Polinomial	0.87	0.86	0.86	84,84%
	Sigmoid	0.30	0.46	0.35	42,42%

Bab V

Penutup

5.1 Kesimpulan

Berdasarkan sistem klasifikasi pendeteksian jenis klik pada *pointing device* menggunakan *electromyograph*, diperoleh kesimpulan sebagai berikut:

1. Sistem telah berhasil mengembangkan metode klasifikasi yang dapat mengklasifikasikan jenis-jenis klik berdasarkan masukan dari sensor EMG menggunakan metode klasifikasi K-NN dengan variabel $k=3$ sebagai akurasi tertinggi yang didapatkan dari pengujian yang telah dilakukan sebesar 81,81% dan metode klasifikasi SVM dengan menggunakan kernel polinomial sebagai akurasi tertinggi yang didapatkan dari pengujian yang telah dilakukan sebesar 84,84%.
2. Dengan membandingkan kedua metode didapatkan nilai akurasi tertinggi yaitu menggunakan metode SVM memiliki nilai akurasi lebih besar dibanding metode K-NN.

5.2 Saran

Berdasarkan penelitian ini dan berbagai pengujian yang telah dilakukan, berikut merupakan beberapa saran yang dapat dibuat dan dikembangkan untuk penelitian yang selanjutnya :

1. Melakukan percobaan menggunakan metode lain sebagai perbandingan lebih lanjut untuk mencari akurasi yang lebih baik.
2. Menambahkan fitur untuk menggerakkan kursor.
3. Menambah fitur data yang dipakai untuk melatih sistem sehingga mendapat akurasi yang lebih baik.

Daftar Pustaka

- Anam, K. 2016, 'Mengenal Elektromiografi (EMG)', Komunitas Ilmuwan dan Profesional Muslim Indonesia (KIPMI), diakses tanggal 2 Februari 2020, <<https://kipmi.or.id/mengenal-elektromiografi-emg.html>>.
- Bellingegni, A., Gruppioni, E., Collazzo, G., Davalli, A., Sacchetti, R., Guglielmelli, E., et al., 2017. NLR, MLP, SVM, and LDA: a comparative analysis on EMG data from people with trans-radial amputation. *Journal of NeuroEngineering and Rehabilitation*, 14:82.
- Cortes, C., dan Vapnik, V. 1995. Support-Vector Networks, *Machine Learning*, 20(3):273-297.
- Disabilitas (Def 1 dan 2). n.d., Kamus Besar Bahasa Indonesia (KBBI) Online, diakses tanggal 13 Februari 2020, <<https://kbbi.kemdikbud.go.id/entri/disabilitas>>.
- Ilham, Anas. 2019. Biosignal – Pengertian, Klasifikasi, dan Analisa, diakses tanggal 10 Maret 2020, <<https://soalkimia.com/biosignal-pengertian-klarisifikasi-dan-analisa/>>.
- Informatikalogi. 2017, Algoritma K-Nearest Neighbor (K-NN), diakses tanggal 18 Maret 2020, <<https://informatikalogi.com/algoritma-k-nn-k-nearest-neighbor/>>.
- Lexico. n.d., *Pointing Device* – Lexico Dictionary powered by Oxford, diakses tanggal 10 Februari 2020, <https://www.lexico.com/definition/pointing_device>.
- Muahradian, A. 2018, Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal, diakses tanggal 22 Maret 2020, <<https://www.petanikode.com/python-linux/>>.
- Nugroho, A.S., Witarto, A.B., dan Handoko, D. 2003. Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika. hal 2-3.
- Paul, Y., Goyal, V., and Jaswal, R.A., 2017. Comparative Analysis between SVM & KNN Classifier for EMG Signal Classification on Elementary Time Domain Feature. 4th IEEE International Conference on Signal Processing, Computing and Control (ISPCC 2k17), pp 169-175.
- R.B. Widodo, A.B. Haryasena, H. Setiawan, M. Subianto, and P.L.T Irawan. 2020^a, "The IMU and Bend Sensor as a Pointing Device and Click Method," International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 306-309, 2019.

- R.B. Widodo, A.B. Haryasena, H. Setiawan, P.L.T. Irawan, M. Subianto, C. Wada. 2020^b, “Design and Evaluation of Upper-Arm Mouse using Inertial Sensor for Human-Computer Interaction.” *Journal of Engineering Science and Technology*, Vol.15, No.6, pp. 3673-3690.
- R.B. Widodo, E. Jahja, Y. Giovanni. 2020. The Combination of Foot Switch and Low-Cost IMU for a Wearable Mouse in Human-computer Interaction. *Proceeding of ICITEE-IEEE*.
- Sari, N. 2019, ‘Jaringan Otot – Pengertian, Gambar, Struktur, Jenis, Fungsi’, diakses tanggal 18 Februari 2020, < <https://rumus.co.id/jaringan-otot/> >.
- Sharma, A. n.d., Confusion Matrix in Machine Learning, diakses tanggal 18 Maret 2020, <<https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>>.
- Sugihono, H., Widodo, R.B., Kelana, O.H. 2018, “Study of the android and ANN-based upper-arm mouse,” *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp.718-723.

LAMPIRAN

MANUAL BOOK FOR CLICK CLASSIFICATION USING ELECTROMYOGRAPH

ELECTROMYOGRAPHY

1. Component of Electromyography

Electrodes

An electromyograph uses electrodes attached to the skin, which are able to detect electrical currents, in order to provide us with information about the contraction.



Biosignalplux HUB

The biosignalsplux hub is the 4 or 8-channel device that collects and digitizes all signals from the sensors and accessories and transmits them via Bluetooth to the computer, where they are recorded and visualized in real-time.



EMG Sensor cables & sleeves

The EMG sensor cables of this sensor can be connected to electrodes with integrated stud connectors. The colored plastic sleeves on each electrode cable indicate which electrode must be connected to the positive or negative electrode cable.



Ground cable

Cable to share a common reference point for all sensors. However, some sensors (e.g., ECG and EMG) have their own ground/reference electrodes, which can be used instead.



2. Procedur of electromyography

Connect Electrode Cable

Plug electrode cables into the first port (CH1), make sure to plug it tight.



Connect Ground Cable

After EMG sensor cables plugged, plug the ground cable into the last port and also make sure to plug it tight as well.



Attach Electrode to EMG Sensor

Next, attach electrodes to EMG sensor, attach it to both of the sensor.



Attach Electrode to Ground Cable

Attach electrode to ground cable as well, then after both sensor and ground cable attached by electrodes, remove the electrodes seal so it can be ready to use.



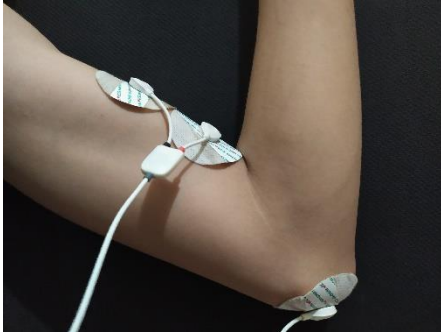
Stick EMG Sensor to Left Arm Biceps Muscle

Next, stick both EMG sensors to users left arm biceps muscle. Find the muscles properly before sticking the sensor to give proper data.



Stick Ground Cable to Left Arm Elbow

Pick the ground cable and stick it to users left arm elbow as the reference for the sensors. Make sure to stick the ground cable on the elbow which has less muscles



Turn On Device

Last but not least, turn on the device by pressing the power button on the Biosignals HUB.



3. Procedure to Run Application

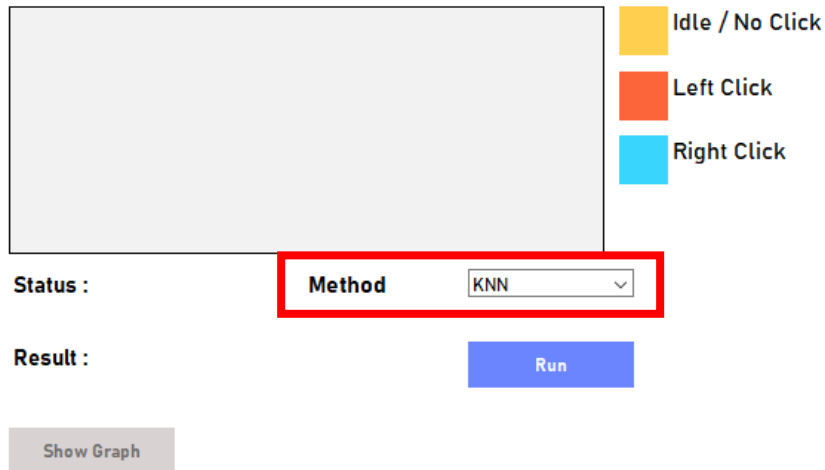
Run Application

Using python, run the application by calling the *gui.py* script. Make sure python is already installed in your computer to run this program.

```
D:\EMG-Mouse-Click-master\PLUX_API_Python2.7_v1.1>python gui.py
```

Choose Method to Classify the Click

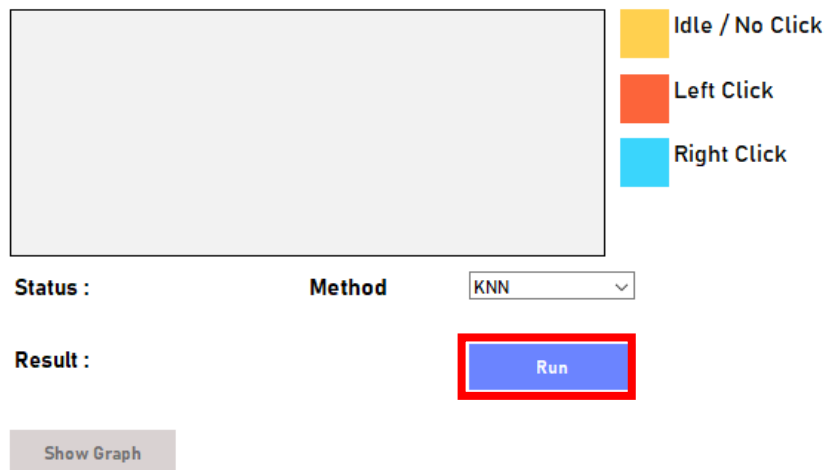
Second, choose the classification method to be used in the application. There is 2 method that can be chosen between KNN and SVM..



The screenshot shows a web application interface. At the top, there is a large gray rectangular area. To its right, there is a legend with three colored squares: yellow for 'Idle / No Click', orange for 'Left Click', and blue for 'Right Click'. Below the gray area, there are labels 'Status :' and 'Result :'. To the right of 'Status :', there is a 'Method' dropdown menu with 'KNN' selected. This dropdown menu is highlighted with a red rectangular box. Below the 'Result :' label, there is a blue 'Run' button. At the bottom left, there is a gray 'Show Graph' button.

Run Program

Click the run button to run the program. After the program running, sensors finally can be used to determine the click for your computer.



This screenshot is identical to the one above, showing the same application interface. However, in this version, the blue 'Run' button is highlighted with a red rectangular box, indicating the next step in the process.

MAIN CODE FOR CLICK CLASSIFICATION USING ELECTROMYOGRAPH

1. Import library yang dibutuhkan

```
import h5py
import joblib
import os
from FeatureExtraction import FeatureExtract as fe
from sklearn import preprocessing, svm
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

2. Membaca Dataset dan Mendapatkan Data Setiap Interval 5 Detik

```
dir_path = '../train_data'
index = 1
for dir_name in os.listdir(dir_path):
    dir_loc = dir_path + '/' + dir_name
    for filename in os.listdir(dir_loc):
        file = h5py.File(dir_loc + '/' + filename, 'r')
        data = file['00:07:80:4D:2E:9E']['raw']['channel_1'][:,0]

        for i in range(6):
            click_index = 5000 * (i + 1)
            noclick_index = click_index - 1500

            if index == 1:
                x_data.append(fe.getAllData(fe, data[click_index:click_index + 1500]))
                y_data.append(1)
                x_data.append(fe.getAllData(fe, data[noclick_index:noclick_index + 1500]))
                y_data.append(0)
            else :
                x_data.append(fe.getAllData(fe, data[click_index:click_index + 1500]))
                y_data.append(2)
            index+=1
```

3. Melakukan Proses Feature Extraction

```
def wavelength(data):
    total = 0
    for i, value in enumerate(data):
        if i == len(data) - 1:
            break
        if data[i+1] > value:
            total += data[i+1] - value
    else:
```



```

        total += value - data[i+1]
    w1 = total / len(data)

    return w1

def meanAbsoluteDeviation(data):
    avg = sum(data) / len(data)
    total = 0
    for value in data:
        total += abs(value - avg)
    mad = total / len(data)

    return mad

def maxValue(data):
    return max(data)

def minValue(data):
    return min(data)

def medianValue(data):
    return statistics.median(data)

def getAllData(self, data):
    return [
        self.wavelength(data),
        self.meanAbsoluteDeviation(data),
        self.maxValue(data),
        self.minValue(data)]

```

4. Normalisasi Data

```

min_max_scaler = preprocessing.MinMaxScaler()
min_max_scaler.fit(x_data)
x_data = min_max_scaler.transform(x_data)

```

5. Splitting Dataset

```

x_train, x_test = train_test_split(x_data, test_size=0.2, random_state=42)
y_train, y_test = train_test_split(y_data, test_size=0.2, random_state=42)

```

6. Inisialisasi Kernel

```

kernel_list = ['linear', 'poly', 'rbf', 'sigmoid']

```

7. Proses Training dan Pengujian Kernel

```
for k in kernel_list:
    clf = svm.SVC(kernel=k)
    clf.fit(x_train, y_train)

    svm_prediction = clf.predict(x_test)
    svm_accuracy = accuracy_score(y_test, svm_prediction) *100
    svm_report = classification_report(y_test, svm_prediction)
    print('SVM Accuracy (' + k + '): ', svm_accuracy, '%')
    print(svm_report)
```

8. Menyimpan Model SVM

```
svm_filename = 'svm_' + k + '_model.pkl'
joblib.dump(clf, svm_filename)
```

9. Proses Training dan Pengujian KNN

```
for n in (3, 5, 7, 9, 11):
    knn = KNeighborsClassifier(n_neighbors=n)
    knn.fit(x_train, y_train)

    knn_prediction = knn.predict(x_test)

    knn_accuracy = accuracy_score(y_test, knn_prediction) *100
    knn_report = classification_report(y_test, knn_prediction)
    print('KNN Accuracy K(' + str(n) + '): ', knn_accuracy, '%')
    print(knn_report)
```

10. Menyimpan Model KNN

```
knn_filename = 'knn_' + str(n) + '_model.pkl'
joblib.dump(knn, knn_filename)
```

11. Menyimpan Model Normalisasi

```
scaler_file = 'scaler.pkl'
joblib.dump(min_max_scaler, scaler_file)
```