Digital Repository Universitas Jember

SISTEM PENGENALAN POLA GERAK TANGAN SEDERHANA BERBASIS EEG (*ELECTROENCHELPALOGRAPH*) UNTUK KONTROL ROBOT TANGAN

SKRIPSI

Oleh

Rijal Al Kautsar Muluk

NIM. 151910201068

PROGRAM STUDI TEKNIK ELEKTRO STRATA SATU
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019

Digital Repository Universitas Jember

SISTEM PENGENALAN POLA GERAK TANGAN SEDERHANA BERBASIS EEG (*ELECTROENCHELPALOGRAPH*) UNTUK KONTROL ROBOT TANGAN

SKRIPSI

Diajukan guna melengkapai tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh

Rijal Al Kautsar Muluk

NIM. 151910201068

PROGRAM STUDI TEKNIK ELEKTRO STRATA SATU
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019

Digital Repository Universitas Jember

DAFTAR ISI

Digital Repository Universitas Jember

Digital Repository Universitas Jember

DAFTAR GAMBAR

Digital Repository Universitas Jember

Digital Repository Universitas Jember

Digital Repository Universitas Jember

DAFTAR TABEL

Digital Repository Universitas Jember

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Otak manusia memiliki aktivitas kimia dan listrik intensif yang disebut bio-sinyal yang terjadi pada waktu tertentu dan di lokasi otak yang terlokalisasi. Semua itu dapat diamati dengan tingkat pengulangan tertentu di bawah kondisi lingkungan yang terdefinisi dengan baik. Masalah fisiologis sederhana ini dapat mengarah pada pengembangan sistem komunikasi baru. Mendeteksi dan menganalisis bio-sinyal otak manusia dapat bermanfaat bagi kita untuk mengetahui konstruksi otak dan fungsi operasional. Pada perkembangannya sinyal otak atau yang biasa disebut *electroencephalograph* (EEG) dapat dimanfaatkan pada pengendalian suatu perangkat keras, termasuk robot.

Robot dapat menjadi salah satu alat bantu terapi, contohnya bagi penyandang disabilitas terutama gangguan pada bagian motorik. Di Indonesia angka disabilitas yang terganggu pada bagian motoriknya mencapai 10,26% (Kementrian Kesehatan RI, 2014) dengan sebaran gangguan yang berbeda – beda. Melihat kondisi yang ada tentunya berbagai upaya telah dilakukan oleh pemerintah dalam mengatasi permasalahan tersebut, seperti upaya pemenuhan hak – hak penyandang disabilitas dalam berbagai aspek kehidupan baik dalam upaya pemberian terapi serta pemberian alat bantu untuk penyandang disabilitas tersebut (Kementrian Kesehatan RI, 2014).

Robot, dalam perkembangnya kini tidak sekedar dijadikan sebagai alat bantu pada dunia industri, namun juga mulai dikembangkan secara bertahap untuk memasuki dan berinteraksi dengan kehidupan manusia, salah satunya penggunaan robot bantu untuk penyandang disabilitas atau dikenal dengan sebutan *assistive robots* (Bi, Fan, & Liu, 2013).

Pada umumnya, untuk dapat mengoperasikan *assistive robots* masih dioperasikan secara konvesional seperti menggunakan *joystick, remote control, keyboard* dan *mouse*. Hal ini tentu saja akan menimbulkan masalah apabila penyandang disabilitas yang menjadi pengguna robot ini memiliki umur yang tua

Digital Repository Universitas Jember

dan memiliki tenaga yang sangat lemah untuk dapat mengoperasikan robot yang menggunakan kendali konvesional ini. Selain itu, beberapa penyakit juga memiliki dampak pada terganggunya bagian anggota motorik, seperti pada orang – orang yang memiliki penyakit *lumpuh sebagian***,** *parkinson* dan *stroke* (Bi et al., 2013) yang tentunya akan memberikan kesulitan dalam pengoperasian robot. Selain itu, pada beberapa kasus penggunaan robot ini sebagai robot terapi juga memiliki kendala, yaitu pada ketidaktahuan batasan tenaga tambahan yang dibutuhkan pasien pada saat proses terapi, sehingga apabila tenaga yang diberikan melampaui batasan yang dibutuhkan, cenderung dapat menimbulkan cidera tambahan bagi pasien. Oleh karena itu proses penyembuhan masih belum bisa bersifat alami (Veneman et al., 2007).

Dengan berkembangnya teknologi biomedis, permasalahan penggunaan kendali robot yang masih menerapkan kendali konvesional tersebut sebenarnya telah diselesaikan. Permasalahan tersebut telah dipecahkan dengan cara menggunakan sinyal – sinyal yang terdapat pada anggota tubuh manusia (bio-sinyal) untuk mengedalikan suatu perangkat (Rechy-ramirez & Hu, 2015). Solusi alternatif yang dapat digunakan adalah mengambil sinyal yang terdapat pada sinyal kendali utama pada tubuh, yaitu sinyal pada otak menggunakan EEG.

Dalam berbagai penelitian yang telah dipublikasikan dan erat kaitannya dengan penggunaan sinyal EEG untuk mengendalikan sebuah perangkat, telah banyak sekali perkembangannya dan variasi penelitiannya, terutama penggunaan EEG untuk mengendalikan robot tangan tiruan (*prosthetic hand robot*) sebagai alat pengganti tangan ataupun sebagai alat bantu terapi. Dalam topik tersebut, EEG difungsikan untuk dapat mengendalikan *prosthetic hand robot* dengan tugas tertentu, misalnya untuk dapat menentukan perbedaan gerakan tangan kiri dan kanan (Bhattacharyya, Khasnobish, & et al, 2011) menggerakan tangan terbuka dan menggengam untuk terapi *stroke* (Fok et al., 2011) menggerakan tangan berdasarkan arah tertentu (Loza, Philips, & et al, 2014), menggerakan bagian jari tertentu (Hayashi, Yokoyama, & et al, 2017) dan lain sebagainya. Selain itu, fokus penelitian tentang pengaplikasian dari EEG untuk diterapkan pada *assitive robot* pada umumnya dan robot tangan tiruan pada khususnya juga banyak berfokus

Digital Repository Universitas Jember

pada penggunaan algoritma ataupun sistem kecerdasaan buatan untuk dapat mengklasifikasi atau mengenali sinyal yang dikeluarkan oleh EEG (Bi et al., 2013; Rechy-ramirez & Hu, 2015). Beberapa algoritma klasifikasi yang telah digunakan diantaranya adalah *Bayesian Classifier* (BC), *Linear Discriminant Analysis* (LDA), *Support Vector Machines* (SVM), *Statistical Classifier*, *Fuzzy Logic* dan *Artificial Neural Network* (ANN) (Bi et al., 2013; Rechy-ramirez & Hu, 2015). Tujuan dan urgensitas dari penelitian yang berfokus pada penggunaan metode klasifikasi yang berbeda – beda ini adalah untuk mencari tingkat performa yang terbaik dari algoritma yang digunakan dalam menterjemahkan sinyal dari EEG (Bi et al., 2013; Rechy-ramirez & Hu, 2015) dengan kriteria : 1) penggunaan metode yang mudah; 2) *low computational complexity;* dan 3) tingkat akurasi yang tinggi (Bi et al., 2013).

Salah satu penelitian yang berfokus pada tiga kriteria tersebut adalah penelitian yang dilakukan oleh Andreas Schwarz dkk pada tahun 2017. Penelitian tersebut berjudul *decoding natural reach-and-grasp actions from human EEG* (Schwarz, Ofner, Pereira, Sburlea, & Gernot R. Müller-Putz, 2017). Pada penelitian ini, peneliti melakukan klasifikasi pada gerakan tangan yang berbeda. Gerakan tangan yang diklasifikasi adalah *grasp* dan *no movement* (tanpa gerakan). Untuk mempermudah dalam pengklasifikasian gerakan tangan tersebut, dalam penelitian ini metode yang dipakai adalah analisa pada sinyal *movement-related cortical potentials* (MRCPs). Metode ini berfokus pada pembacaan pola amplitudo pada sinyal EEG yang telah di *filter* pada frekuensi tertentu, sehingga mudah dibaca dan diklasifikasi. Dengan menggunakan analisa pada sinyal tersebut, kedua gerakan tangan diklasifikasi dengan *binary classification* yang memiliki hasil akurasi 72.4%. Hasil tersebut adalah hasil terbaik yang telah didapatkan dengan menggunakan metode ini yaitu dengan mengujinya pada parameter *windows size* dan jumlah *channel* yang berbeda. Melihat hasil pada penelitian tersebut perlu adanya perbaikan pada proses metode pengklasifikasian agar mendapatkan tingkat akurasi yang tinggi.

Salah satu alternatif yang bisa digunakan pada klasifikasi *hand movement task* adalah penggunaan metode *Extreme Learning Machine* (ELM). *Extreme*

Digital Repository Universitas Jember

Learning Machine (ELM) merupakan pengembangan dari *Artificial Neural Network* yang menggunakan *single-hidden layer feed- forward neural networks* (SLFNs) (Cambria et al., 2013). Pada beberapa kajian yang membahas tentang metode ini, *extreme learning machine* sering diunggulkan karena memiliki tingkat *error* yang kecil pada saat proses *training, extremely fast learning speed,* serta memberikan performa yang terbaik dalam setiap aplikasi implementasinya (Cambria et al., 2013). Hal ini dapat dibuktikan pada beberapa penelitian yang menggunakan metode ini dan membandingkan dengan metode lain dalam berbagai aplikasi, seperti penggunaan ELM pada proses diagnosis penyakit kanker (Zhang, Huang, Sundararajan, & Saratchandran, 2007) yang membandingkan metode ELM dengan metode *Support Vector Machine* (SVM), *fault detection* untuk *line transmission protection* (Malathi, Marimuthu, & Baskar, 2010) yang membandingkan ELM dengan SVM, serta untuk pendeteksian penyakit epilepsi berdasarkan sinyal *interictal* dan *ictal* EEG (Yuan, Zhou, Li, & Cai, 2011) dengan membandingkan metode ELM dengan BP-ANN (*backpropagation*) serta dengan SVM

Dengan melihat kelebihan pada beberapa penelitian yang telah dilakukan terkait penggunaan metode ELM dalam beberapa aplikasi serta urgensitas pengembangan *assistive robots* untuk robot tangan tiruan terutama dalam pengklasifikasian *hand movement task*, maka metode ini perlu digunakan. Dengan menggunakan metode ini dalam pengklasifikasian sinyal EEG pada *hand movement task*, diharapkan mampu memberikan kontribusi dalam menyelesaikan permasalahan pada tingkat akurasi serta *fast computational complexity* penelitian sebelumnya. Oleh karena itu pada penelitian ini, peneliti akan berfokus pada penggunaan metode klasifikasi *Extreme Learning Machine* untuk mengklasifikasi sinyal dari EEG yang kemudian implementasikan pada robot tangan tiruan (*prosthetic hand robot*).

Digital Repository Universitas Jember

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang diatas, dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Bagaimana rancangan dan pembuatan robot tangan tiruan yang dapat digunakan sebagai *asistive robot* ?
2. Bagaimana cara mengklasifikasi sinyal EEG dengan menggunakan metode *Extreme Learning Machine* untuk *mengenal gerakan tangan sederhana* pada metode klasisfikasi biner?
3. Bagaimana performa metode *Extreme Learning Machine* (ELM) dalam pengklasifikasian sinyal EEG untuk *mengenal gerakan tangan sederhana* pada metode klasisfikasi biner?

1.3. Batasan Masalah

Berdasarkan rumusan masalah diatas, batasan masalah yang dapat diambil adalah:

1. Robot tangan buatan menggunakan model *Flexy-Hand* yang di design oleh Gyrobot yang kemudian dikembangkan oleh peneliti.
2. Metode klasifikasi sinyal EEG menggunakan *Extreme Learning Machine single-hidden layer* yang diberikan bobot secara acak.
3. Klasifikasi sinyal EEG menggunakan metode klasisfikasi biner pada gerakan menggenggam yang bernilai biner 1 dan tangan terbuka yang bernilai biner 0.

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Mendapatkan model robot tangan tiruan yang dapat digunakan sebagai *asistive robot* untuk disabilitas.
2. Mengetahui cara pengklasifikasian sinyal EEG dengan menggunakan metode *Extreme Learning Machine* (ELM).
3. Mengetahui performa *Extreme Learning Machine* (ELM) untuk pengklasifikasian sinyal EEG pada *gerakan tangan sederhana* dengan metode klasisfikasi biner.

Digital Repository Universitas Jember

1.5. Manfaat Penelitian

Manfaat dari penelitian ini dalam dunia biomedis tentunya akan memberikan kontribusi pada peningkatan performa pada proses pengklasifikasian bio-signal untuk sinyal EEG dengan kriteria metode yang mudah diaplikasikan, *low computational complexity* dan tentunya memiliki tingkat akurasi yang tinggi. Selain itu manfaat yang dapat diambil oleh mahasiswa adalah sebagai tambahan informasi terkait penggunaan *Extreme Learning Machine* (ELM) dalam pengklasifikasian sinyal EEG khususnya untuk menggerakan robot tangan tiruan.

Digital Repository Universitas Jember

BAB 2. TINJAUAN PUSTAKA

2.1. *Electroencephalograph* (EEG)

Menurut Kamus Oxford, *electroencephalograph* (EEG) adalah suatu teknik untuk merekam aktifitas listrik di bagian yang berbeda di otak dan mengubah informasi ini menjadi suatu pola atau gambaran baik secara digital atau dicatat di atas kertas yang disebut sebagai *electroencephalograph*. Alat yang merekam aktifitas listrik di otak ini disebut sebagai *encephalograph*. *Encephalograph* akan membanding tegangan volt yang direkam pada 2 bagian yang berlainan di otak. Pada EEG, susunan elektroda logam akan diletakan pada kulit kepala pasien dan aktifitas listrik akan direkam selama 30 menit. Aktivitas listrik otak ini dibaca di bagian yang berlainan pada korteks otak pada masa yang sama. Dulu, encephalograph adalah dalam bentuk kertas, sekarang sudah ada dalam bentuk digital.

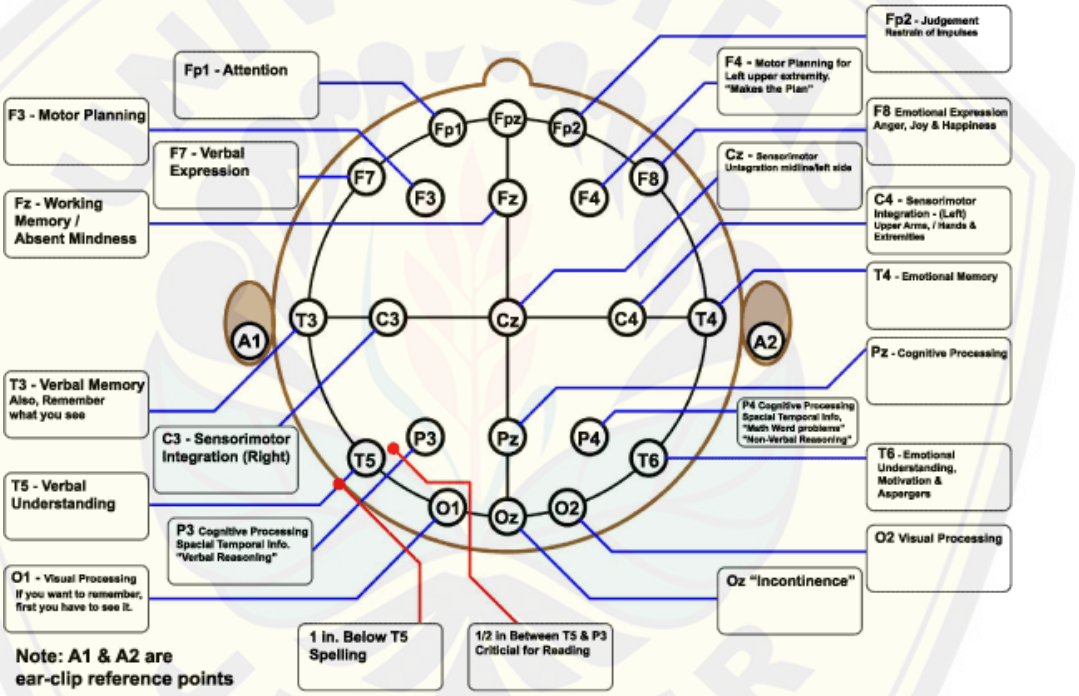
Pada orang yang normal, gambaran EEG akan menunjukkan beberapa jenis gelombang yang spesifik mengikut dengan keadaan seseorang itu. Terdapat 4 jenis gelombang di otak normal yaitu :

1. *Beta* merupakan gelombang otak dengan rentang frekuensi antara 13 - 30 Hz. *Beta* adalah gelombang otak yang biasanya terjadi pada saat seseorang sedang aktif berpikir, aktif konsentrasi atau sedang fokus dalam memecahkan suatu permasalahan. Beta terbagi lagi ke dalarn 2 kelompok berdasarkan rentang frekuensi yaitu *Beta Low* (13-20 Hz) dan *Beta High* (20 - 30 Hz)
2. *Alpha* merupakan gelombang otak dengan rentang frekuensi antara 8 - 13 Hz. *Alpha* mengindikasikan seseorang dalam kcadaan kondisi pikiran relaks. Gelombang ini paling besar muncul pada bagian otak daerah *occipital cortex* dan juga pada bagian *frontal cortex*.
3. *Theta* merupakan gelombang otak dengan rentang fiekuensi antara 4 - 8 Hz. *Theta* muncul pada seseoarang yang mengalami stress secara emosional terutama frustasi atau kekecewaan.

Digital Repository Universitas Jember

4. *Delta* merupakan gelombang otak dengan rentang frekuensi antara 0.5 - 4 Hz. *Delta* muncul pada saat seseorang dalam keadaan tidur lelap. Sinyal *delta* dapat pula mengindikasikan adanya cacat fisik di otak (Guyton dan Hall, 2006).

Dalam EEG terdapat beberapa titik pengambilan sinyal yang memiliki informasi dominan tertentu. Oleh karena itu, pemasangan elektroda yang merupakan perangkat untuk mengambil sinyal pada otak perlu diperhatikan. Posisi dari elektro dapat dilihat pada Gambar 2.1 tentang informasi titik sinyal dan posisi elektroda pada EEG.



Gambar 2.1. Titik sinyal informasi elektroda
(Sumber : https://www.quora.com/)

Untuk dapat memanfaat sinyal EEG dalam merepresentasikan sinyal gerakan tubuh manusia, maka dapat memanfaatkan salah satu sinyal pada EEG, yaitu sinyal *Event—Related Synchronization / Desynchronization* (ERS/ERD). Sinyal ini adalah jenis sinyal pada EEG yang dihasilkan akibat adanya aktivitas pergerakan pada bagian motorik tubuh manusia sehingga terjadi kenaikan dan penurun pada amplitudo sinyal dalam rentan frekuensi yang spesifik. Untuk dapat mengambil sinyal dari ERS/ERD, maka diperlukan perlakukan khusus agar sinyal

Digital Repository Universitas Jember

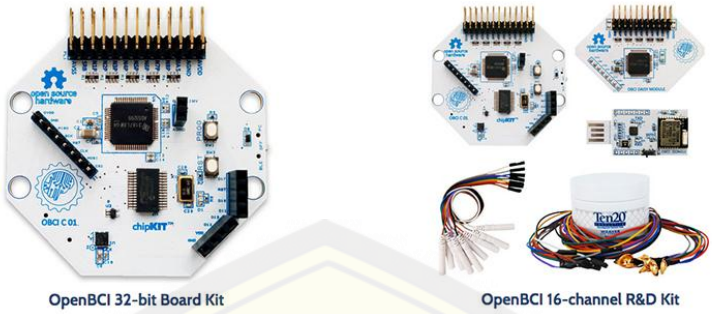
dapat diolah dengan tepat. diperlukan suatu proses pengolahan sinyal yang tepat. Hal ini disebabkan sinyal yang terdapat pada tubuh tercampur dengan sinyal lainnya seperti sinyal pergerakan jantung, pergerakan tangan yang juga menghasilkan tegangan listrik. Oleh karena itu perlu adanya tahap segmentasi khusus pada data yang diinginkan agar sinyal ERS/ERD dapat diperoleh dan dapat menghasilkan sinyal yang optimal. (Yulianto et al. 2013; Majkowski et al. 2017).

2.2. Divais EEG

Untuk dapat dapat menangkap sinyal EEG, pada penelitian ini peneliti akan menggunakan divais EEG yang merupakan produk keluaran dari *openBCI*. Divais tersebut bernama Ultracortex Mark IV *EEG Headset*. Ultracortex Mark IV adalah edisi terbaru Ultracortex — *headset EEG* yang nyaman, dapat disesuaikan dengan ukuran kepala subjek, dan dapat dicetak ulang 3D desainnya serta kompatibel dengan semua papan dari produk *OpenBCI*. Desain revolusioner Ultracortex menggunakan *dry sensor* EEG dan membutuhkan waktu kurang dari 30 detik untuk menyala dan beroperasi. Bentuk dari *headset* EEG ini dapat dilihat pada Gambar 2.2 dan Gambar 2.3.

Gambar 2.2. EEG *headset* Ultracortex Mark IV
(Sumber : https://docs.openbci.com/Headware/01-Ultracortex-Mark-IV)

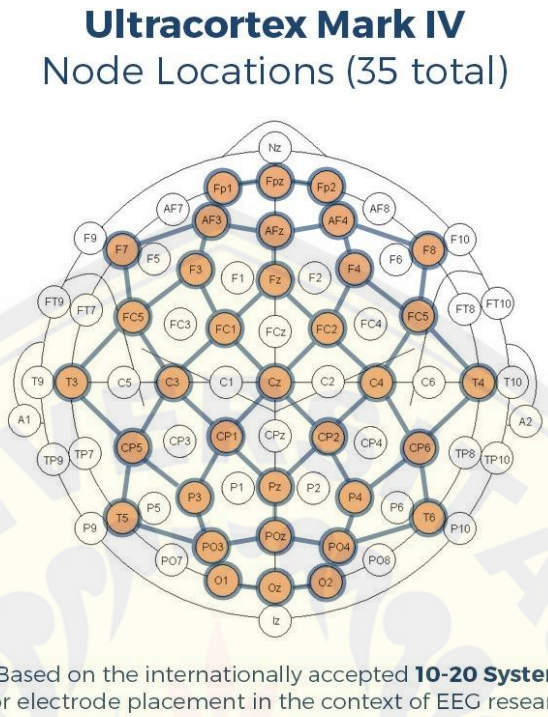
Digital Repository Universitas Jember



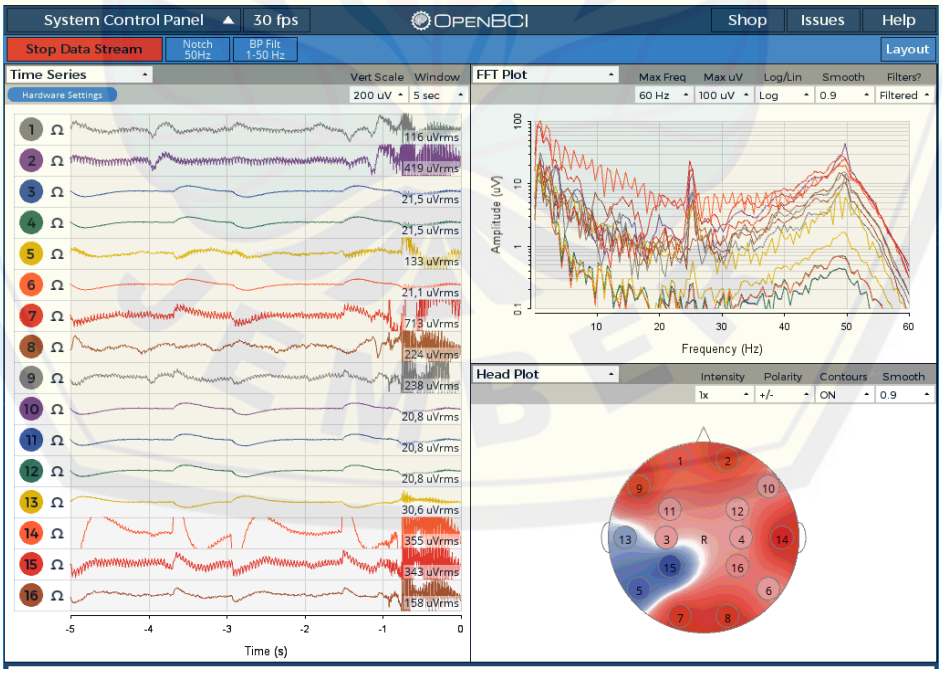
Gambar 2.3. EEG *headset* dengan *board* OpenBCI
(Sumber : https://docs.openbci.com/Headware/01-Ultracortex-Mark-IV)

Ultracortex Mark IV ini memiliki *channel* pengambilan data mencapai 35 titik pada standar *channel* 10-20 *system international*. *Channel* yang dimaksud dapat dilihat pada Gambar 2.4. Selain itu, Ultracortex Mark IV ini juga dilengkapi dengan dua *board* dari *open*BCI untuk dapat menyimpan data dan mengkomunikasikan data kepada aplikasi dari *open*BCI sendiri melalui komunikasi *bluetooth*. Dua *board* tersebut adalah *Cyton* dan *Ganglion* yang masing – masing *board* memiliki fungsi tersendiri. *Board* ini membutuhkan daya suplai dari baterai AA 6V untuk menyala. Disamping itu *board* ini juga disediakan *slot* untuk SD *Card* apabila pengguna ingin menyimpan data EEG pada SD *Card*. Namun apabila pengguna menginginkan menyimpan data kedalam PC secara langsung, maka dapat menggunakan aplikasi *open*BCI yang sudah tersedia pada laman dari *open*BCI. Aplikasi yang dimaksud dapat dilihat pada Gambar 2.5.

Digital Repository Universitas Jember



Gambar 2.4. Titik *channel* pada Ultracortex Max IV
(Sumber : https://shop.openbci.com/products/ultracortex-mark-iv)

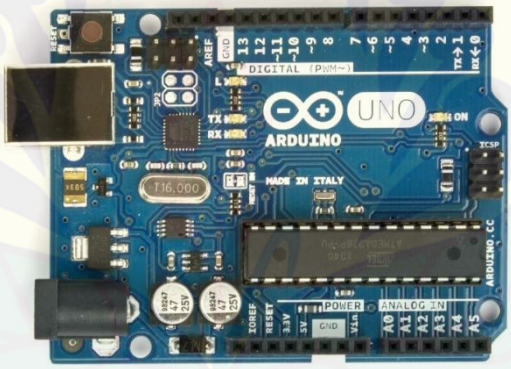


Gambar 2.5. Aplikasi dari OpenBCI
(Sumber : http://docs.openbci.com/)

Digital Repository Universitas Jember

2.3. Arduino UNO

Arduino Uno adalah board mikrokontroler berbasis ATMEGA 328. Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan sumber DC atau baterai untuk menjalankannya. Uno berbeda dari semua papan sebelumnya dalam hal itu tidak menggunakan FTDI chip driver USB-to-serial. Sebaliknya, fitur Atmega16U2 (Atmega8U2 hingga versi R2) di program sebagai konverter USB-to-serial.



Gambar 2.6. Arduino UNO

(Dayat Hidayat, 2015)

Ringkasan dari *ARDUINO UNO*

Tabel 2.1. *Datasheet* Arduino UNO

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (*of which 6 provide PWM output*)

Digital Repository Universitas Jember

Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V *Pin*	50 mA
Flash Memory	32 KB (ATmega328) *of which 0.5 KB used by bootloader*
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

2.4. *Feature Extraction*

Fitur ektraksi adalah transformasi sinyal mentah (*raw*) menjadi vektor fitur dengan mengurangi *noise* dan penguatan data pada data yang dianggap penting. Tahap ini menyiratkan "pengurangan dimensi", yaitu menghilangkan data redundan dari vektor fitur.

Dikarenakan pada sinyal EEG memiliki gelombang yang berbeda, seperti α, β, δ, γ and θ, oleh karena itu teknik dalam mengektrak sinyal EEG dapat dibagi menjadi beberapa bagian, diantaranya:

1. *Time domain*: Fitur ini diderivasi atau diturunkan secara langsung dari sinyal EEG dan termasuk (rata-rata) *time-course*.
2. *Frequency domain*: Fitur ini mendekripsikan karakteristik kekuatan sinyal otak di beberapa pita frekuensi.
3. *Time–Frequency domain*: Fitur ini menggambarkan bagaimana kekuatan spektrum bervariasi dari waktu ke waktu. Transformasi Fourier yang singkat dan transformasi Wavelet adalah yang paling banyak digunakan pada transformasi ini.
4. *Spatial filtering*: Jenis penyaringan ini menggunakan sinyal dari beberapa elektroda untuk fokus pada aktivitas di lokasi tertentu di otak.

Digital Repository Universitas Jember

5. *Bipolar montage*: saluran bipolar dihitung dengan mengurangi sinyal dari dua elektroda yang berdekatan.
6. *Common average reference*: teknik ini mengurangi nilai rata-rata seluruh montase elektroda (rata-rata umum) dari saluran yang menarik.
7. *Laplacian method*: lokasi elektrode dihitung dengan menggabungkan nilai di lokasi itu dengan nilai-nilai satu set elektroda sekitarnya. Jarak ke set elektroda sekitarnya menentukan karakteristik penyaringan spasial dari Laplacian.
8. *Common spatial patterns*: teknik untuk menganalisis data *multi-channel* berdasarkan rekaman dari dua kelas tugas.

2.5. *Power Spectral Density Feature Extraction*

Power Spectral Density atau densitas spektrum daya adalah fitur ektraksi dari sinyal yang di *filter*. Pada metode ini, analisis yang diekstrak adalah besar daya dari sinyal yang dihasilkan pada *range* frekuensi tertentu pada saat sinyal EEG berlangsung. PSD adalah metode non-parametrik dimana dihitung langsung dari sinyal yang di *filter* itu sendiri. Untuk setiap saluran, sinyal yang di *filter* dibagi menjadi 64 segmen jendela dengan 50% *overlap*. Data 5 detik yang disaring pertama dibagi menjadi 64 jendela. Jendela yang digunakan adalah jendela *hamming*. *Fast Fourier Transform* (FFT) kemudian diterapkan ke setiap segmen jendala, sehingga kepadatan spektral daya segmen kemudian dapat dihitung dengan menggunakan persamaan(Javed et al., 2017).

PSD dari setiap segmen diberikan oleh persamaan 1,

$$Pxx = \frac{|X(f)|^2}{FsLU} \quad \text{.....(2.1)}$$

dimana ‘Fs’ adalah frekuensi sampling, ‘L’ adalah panjang segmen dan X (f) adalah data setelah FFT diterapkan di dalamnya. 'U' adalah konstanta normalisasi jendela yang diberikan oleh persamaan 2.

Digital Repository Universitas Jember

$$U = \frac{1}{L}\sum_{n=0}^{N-1}|w(n)|^2 \text{.....} \quad (2.2)$$

Setelah PSD setiap segmen dihitung, nilai-nilai yang sesuai dari setiap segmen ditambahkan dan dirata-ratakan. Varian yang dihasilkan kemudian dirata-ratakan kembali dan kemudian didpatkan kekuatan band rata-rata dari saluran lebih dari 6-30 Hz *band* (karena semua frekuensi lainnya telah dihapus oleh *filter*). Proses ini diulang untuk setiap saluran, yang memberikan vektor fitur dari 14 konstituen yang menunjukkan kekuatan *band* rata-rata setiap saluran.(Javed et al., 2017).

2.6. Pengklasisfikasian

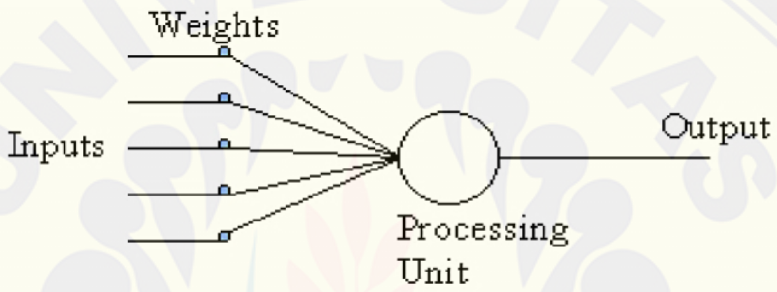
Setelah fitur telah diekstraksi dari sinyal mentah (*raw signal*) dan fitur dengan informasi redundan telah berkurang (pengurangan dimensi), maka dibutuhkan sebuah sistem yang mampu membedakan kategori yang berbeda di antara vektor fitur yang berkurang yang kemudian disebut dengan tahap pengklasifikasian. (Rechy-ramirez & Hu, 2015). Kemudian, fitur yang diperoleh dari tahap klasifikasi tersebut, akan diumpankan pada tahap berikutnya sebagai perintah kontrol, yaitu pengontrol. Pada tahap pengklasifikasian ini biasanya digunakan dua tipe pengklasifikasian, yaitu pengklasifikasian liniear yang didominasi dengan model statistik dan pengklasifikasian nonlinear yang didominasi dengan model *artificial intelegence* atau kecerdasaan buatan.(Bi et al., 2013)

2.7. *Artificial Neural Network*

Artificial Neural Network (ANN) atau yang dikenal dengan Jaringan Syaraf Tiruan (JST) adalah salah satu sistem kecerdasaan buatan / *aritifical intelegence* (AI) yang menirukan jaringan otak pada manusia. ANN adalah sistem AI yang adaptif, dimana ANN dapat merubah strukturnya jaringannya, memanipulasi nilai guna menyelesaikan permasalahan berdasarkan informasi akhir yang diinginkan / target. Dalam tipe pemodelan, ANN termasuk kedalam

Digital Repository Universitas Jember

kelompok pemodelan data statistik non-linear, sehingga dengan ANN dapat digunakan untuk menyelasaikan permasalah pemodelan yang menghubungankan informasi yang kompleks antara *input* dan *output* berdasarkan pola-pola pada data. Dalam ANN, untuk dapat mengetahui hubungan tersebut maka terdapat sistem pembelajaran yang bersifat kontinuitas dan berfungsi sebagai proses penambahan pengetahuan pada jaringan untuk mengeksploitasi informasi secara maksimal. Proses tersebut terjadi pada bagian *neuron*. (Tino, Benuskova, & Sperduti, 2015). Sedangkan struktur dari ANN, dapat dilihat pada Gambar 2.7.



Gambar 2.7. Bentuk dasar *neuron*

Pada bagian struktur ANN, dapat dibagi menjadi 4 bagian. Yang pertama adalah bagian i*nput* yang berfungsi sebagai saluran masuknya informasi yang sumbernya bisa dari informasi asli atau informasi dari struktur keluaran yang ANN yang lain. Bagian yang kedua adalah *weight* yang merupakan beban, pada struktur ini memiliki nilai yang berubah setiap terjadi perubahan nilai *input* akibat dari proses pembelajaran. Bagian yang ketiga adalah bagian *processing unit* yang merupakan bagian proses perhitungan dan pengenalan suatu informasi berdasarkan hasil pembelajaran. Bagian yang terakhir adalah bagian *output* yang merupakan keluaran dari hasil proses pengolahan informasi pada ANN.

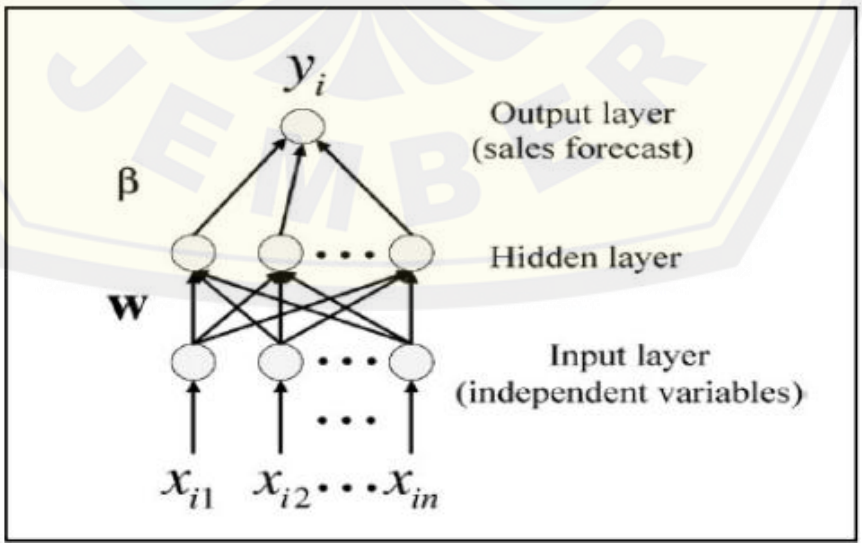
2.8. *Extreme Learning Machine*

Extreme Learning Machine merupakan pengembangan dari sistem kecerdasaan buatan *aritifical neural network* (ANN) yang diperkenalkan oleh Huang pada tahun 2004. (Huang, Zhu, & Siew, 2004). Pada dasarnya *Extreme Learning Machines* adalah *Single Hidden Layer Feedforward Neural Networks*

Digital Repository Universitas Jember

(SLFNs) yang memiliki kelebihan dibandingkan dengan jaringan syaraf tiruan biasa dengan model pembelajaran *backpropagation* (BP-ANN) (Cambria et al., 2013), terutama dalam hal *learning speed*. Disamping itu, pada metode ELM ini, memiliki *error rate* yang lebih minim apabila dibandingkan dengan metode BP-ANN apabila dihitung berdasarkan perhitungan error menggunakan parameter statistik yaitu nilai MSE (*Mean Square Error*) dan MAPE (*Mean Absolute Percentage Error*). (Huang et al., 2004).

Pada *training process* ANN metode c*onventional gradient based learning algorithm* seperti pada jaringan syaraf tiruan menggunakan *learning process backpropagration* (BP -ANN) dan pada *learning process* Lavenberg – Marquadt (LM), parameter JST (jaringan syaraf tiruan) *feedforward* –nya haruslah ditentukan secara manual. Parameter yang dimaksud adalah parameter *input weight* dan *hidden bias*. Dikarenakan pada jaringan JST paramater – parameter saling berhubungan, maka semakin banyak *hidden bias* dan *layer* yang dibangun, maka *learning speed* yang dibutuhkan akan membutuhkan waktu yang lama dan sering terjebak pada *local minima*. (Huang et al., 2004). Sedangkan pada *Extreme Learning Machine*, parameter – parameter tersebut dipilih secara *random* pada awal pelatihan, sehingga ELM cenderung memiliki kecepatan pembelajaran / *learning speed* yang cepat dan memiliki perfoma yang baik (*good generalization performance*). (Huang et al., 2004).



Gambar 2.8. Struktur ELM

Digital Repository Universitas Jember

Metode ELM mempunyai model matematis yang berbeda dari jaringan syaraf tiruan *feedforward*. Model matematis dari ELM lebih sederhana dan efektif. Berikut model matrematis dari ELM. Untuk N jumlah sample yang berbeda (Xi , ti)

$Xi = [Xi_1 , Xi_2 , ..., Xi_n]^T \in R^n$ .....(2.3)

$Xt = [Xt_1 , Xt_2 , ..., Xt_n]^T \in R^n$ .....(2.4)

Standart SLFNs dengan jumlah *hidden nodes* sebanyak N dan *activation function* g (x) dapat digambarkan secara matematis sebagai berikut:

$$\sum_{i=1}^{N} \beta i g i\,(xj) = \sum_{i=1}^{N} \beta i g\,(W_i \cdot X_j + b_i) = O_j \quad \text{.....(2.5)}$$

Dimana:

$J = 1, 2, ..., N$

$w_1 = (w_{i1}, w_{i2}, w_{i3}, \ldots, w_{in})^T$ merupakan *vektor* dari *weight* yang menghubungkan *i* th hidden nodes dan input nodes.

$\beta_1 = (\beta_{i1}, \beta_{i2}, \beta_{i3}, \ldots, \beta_{in})^T$ merupakan *weight vector* yang menghubungkan *i* th *hidden* dan *output nodes*.

bi threshold dari *i* th *hidden nodes*.

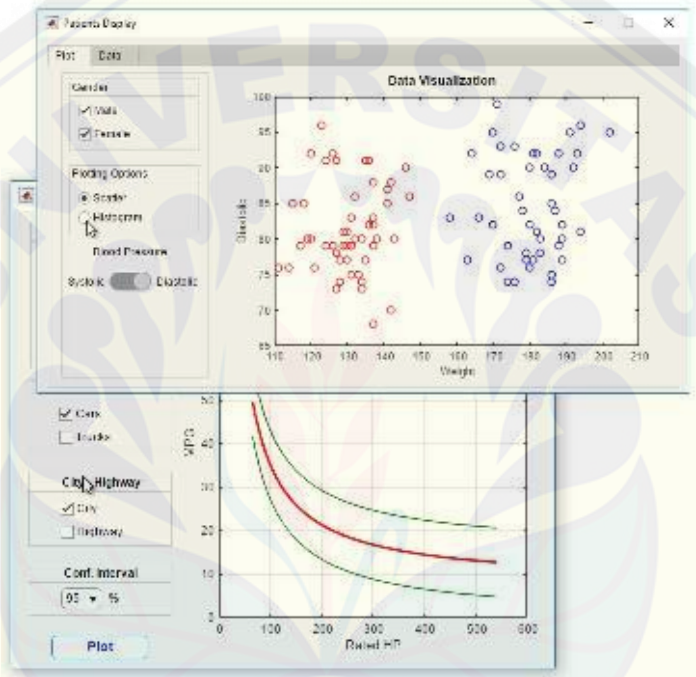
$w_i x_j$ merupakan inner produk dari w_i dan x_j

2.9. MATLAB

MATLAB adalah singkatan dari *Matrix Laboratory*, yaitu sebuah program yang digunakan untuk menganalisis dan mengkomputasi data numerik. MATLAB sendiri juga merupakan bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran sifat dan bentuk matriks. Dalam perkembangan selanjutnya, MATLAB juga mulai dikembangkan untuk dapat

Digital Repository Universitas Jember

kompitabel dengan bahasa C++ dan *assembler* (utamanya untuk fungsi-fungsi dasar). Oleh karena itu, dalam perkembangnya, MATLAB telah menjadi sebuah *environment* pemprograman yang canggih yang berisi fungsi-fungsi *built-in* untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. MATLAB juga menyediakan berbagai fungsi untuk menampilkan data, baik dalam bentuk dua dimensi maupun dalam bentuk tiga dimensi.



Gambar 2.9. Contoh tampilan pada MATLAB
(sumber: http://www.mathworks.com)

Selain itu, kelebihan dari MATLAB adalah sifatnya yang *extensible*. Maksudnya adalah bahwa seorang pengguna dapat menulis fungsi baru untuk menambahkan pada *library* apabila fungsi-fungsi *built-in* pada MATLAB tidak tersedia sehingga dapat melakukan tugas tertentu. Salah satunya adalah *library* pengolahan sinyal biomedis. Dari beberapa jurnal yang telah terpublikasi dapat diketahui bahwa MATLAB dapat digunakan sebagai pengolah sinyal biomedis yang kemudian dapat digunakan untuk mengendalikan suatu perangkat tertentu. Beberapa jurnal yang dimaksud adalah (Asanza et al. 2018; Minh Dao et al. 2017;

Digital Repository Universitas Jember

Luu, Nguyen, and Vo Van 2018) yang menggunakan fitur GUI untuk membangun *library* tersebut dan mengomunikasikan dengan perangkat lain.

2.10. *Hand Prothestic Robot*

Handprothesitc robot atau dapat disebut sebagai *artificial hand robotic* adalah robot tiruan motorik manusia bagian atas atau pada bagian tangan manusia yang difungsikan untuk menggantikan peran tangan manusia atau sebagai *assistive robot* yang membantu peran manusia untuk melakukan pemulihan gerakan atau menambah tenaga pada bagian motorik tersebut.

Pada penelitian ini, peneliti akan menggunakan *handprothestic robot* pada bagian antara pergelangan tangan dengan jari – jari tangan atau pada bagian *upperlimb* (Burke-, n.d.).

2.11. Motor Stepper 28BYJ-48

Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa.

Keunggulannya antara lain adalah :

a. Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
b. Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
c. Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
d. Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran).
e. Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC.
f. Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya.
g. Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

Digital Repository Universitas Jember

Motor stepper jenis 28BYJ-48 adalah motor stepper kecil, murah, bermesin 5 volt. Karena rasio pengurangan gigi 64: 1 ia menawarkan torsi yang layak untuk ukurannya dengan kecepatan sekitar 15 rotasi per menit (RPM). Adapun spesifikasi dari motor stepper jenis 28BYJ-48 yaitu seperti tabel berikut:

Tabel 2.2 *Datasheet* Motor Stepper

Motor Type	*Motor Stepper Unipolar*
Wire Type	*5 wire connection (to driver motor)*
Voltage	*5 – 12 Volt*
Frequency	*100 Hz*
Step mode	*Half-step mode recommended (8 step control signal sequence)*
Step angle	*Half-step mode: 8 step control signal sequence (recommended) 5.625 degrees per step / 64 steps per one revolution of the internal motor shaftFull Step mode: 4 step control signal sequence 11.25*

Digital Repository Universitas Jember

	degrees per step / 32 steps per one revolution of the internal motor shaft
Gear ratio	*Manufacturer specifies 64:1. Some patient and diligent people on the Arduino forums have disassembled the gear train of these little motors and determined that the exact gear ratio is in fact 63.68395:1. My observations confirm their findings. These means that in the recommended half-step mode we will have:64 steps per motor rotation x 63.684 gear ratio = 4076 steps per full revolution (approximately).*
Weight	*30g*

Digital Repository Universitas Jember

BAB 3. METODELOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Pelaksanaan pembuatan alat, penelitian dan analisis sistem pengenalan pola gerak tangan sederhana berbasis EEG dilakukan pada:

Tanggal : Februari 2019 sampai dengan Juni 2019

Tempat : Laboratorium Sistem Cerdas dan Robotika, Gedung CDAST (*Center for Development of Advance Science and Technology*), Universitas Jember.

3.2. **Rencana Jadwal Pelaksanaan Penelitian**

Waktu penelitian dilaksanakan dilaksanakan selama kurang lebih 5 bulan, berikut adalah tabel jadwal penelitian.

Tabel 3.1. Jadwal Kegiatan Penelitian

No	Kegiatan	Bulan			
		3	4	5	6
1	Studi Literatur				
2	Pembuatan *Hand Prosthetics Robot*				
3	Pengambilan Data EEG				
4	Pembuatan Algoritma *Filter* dan *Feature Extraction*				
5	Pembuatan dan Pengujian algoritma ELM				
6	*Training Data* pada algoritma ELM				
4	Pembuatan *embedded system* untuk pengendali robot tangan				
5	Pengujian Sistem Kendali berbasis EEG pada robot tangan				
6	Penulisan Laporan dan Jurnal				

Keterangan :

: Pengerjaan

Digital Repository Universitas Jember

3.3. **Alat dan Bahan**

Alat dan bahan yang digunakan sebagai pembuatan *hand prosthetic robot* dan *embeded system* pengenalan pola gerak tangan berbasis EEG adalah sebagai berikut:

A. *Hardware*

1.	Devais EEG Ultracortex Mark IV	1 buah
2.	OpenBCI *Cyton Board*	1 buah
3.	Baterai Li-Po	2 buah
4.	*Personal Computer*	1 buah
5.	Rangkaian *Driver* Motor	1 buah
6.	Motor Stepper 28byj-48	1 buah
7.	3D *Filament printer*	secukupnya
8.	Akrilik	secukupnya

B. *Software*

1. Arduino IDE
2. Autodesk Inventor
3. MATLAB 2015
4. OpenBCI GUI
5. Python3

C. Bahan Habis Pakai

1.	Kabel	secukupnya
2.	Tali Ukur	secukupnya
3.	Baut dan Mur	secukupnya
4.	Perekat selotip	secukupnya
5.	Perekat Lem	secukupnya

D. Alat Pendukung

1. Solder
2. Bor Listrik
3. Gerinda Listrik
4. Kabel USB

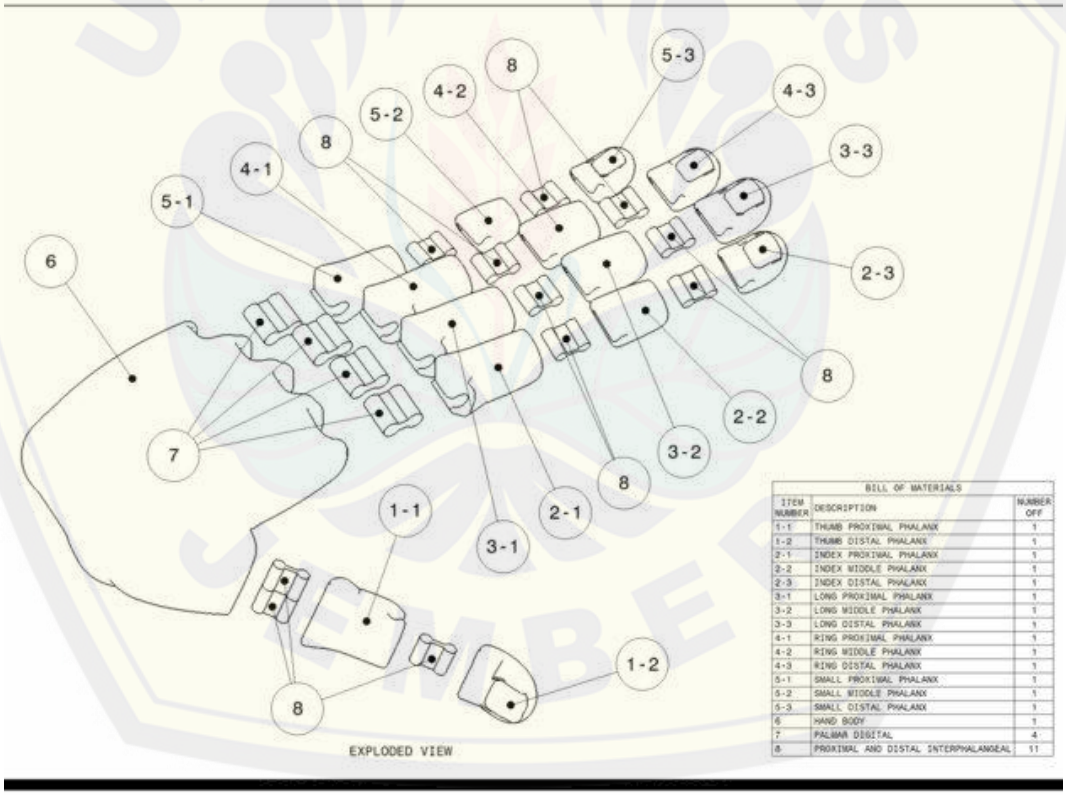
Digital Repository Universitas Jember

5. Tang Potong
6. Tang Jepit
7. Obeng Set
8. Kertas Gosok

3.4. **Perancangan *Hardware***

3.4.1. Perancangan Robot Tangan

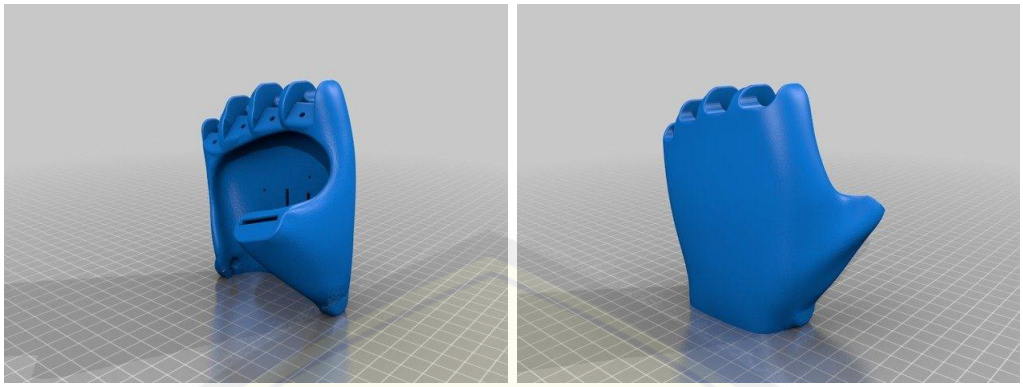
Desain *hand prosthetic robot* menggunakan model *Flexy-Hand 2* yang dibuat oleh Gyrobot, tahun 2014. Flexy-Hand 2 digunakan sebagai dasar desain hand prosthetic robot yang digunakan dengan memberikan rangkaian elektronika sebagai pengendali gerakan robot menggunakan aktuaktor motor stepper.



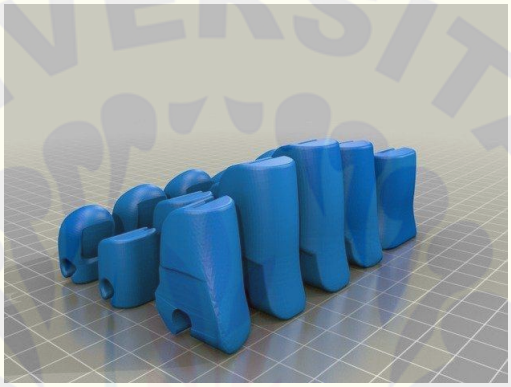
Gambar 3.1 Desain dasar model *Flexy-Hand 2* oleh Gyrobot.
(sumber: https://www.thingiverse.com/thing:380665, 2019)

Perancangan *hand prosthetic robot* model *Flexy-Hand* 2 oleh Gyrobot memiliki desain awal sebagai berikut:

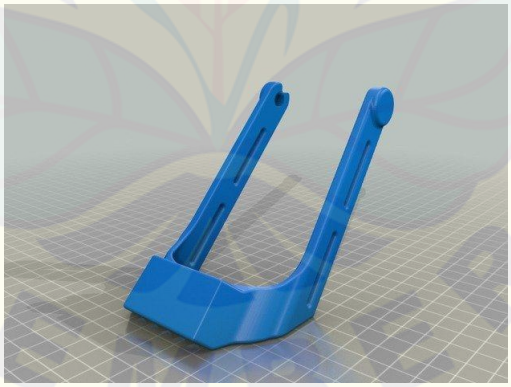
Digital Repository Universitas Jember



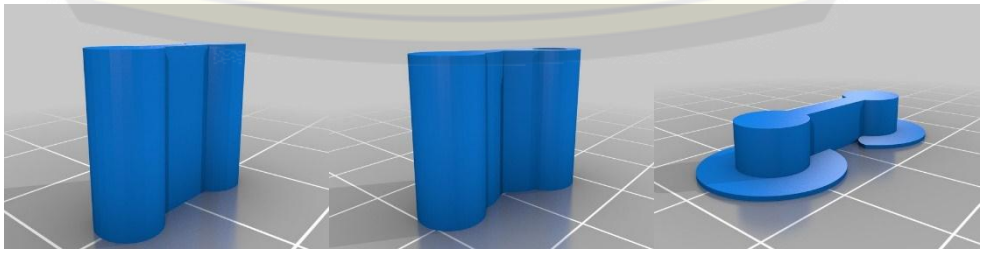
Gambar 3.2 Desain Telapak Tangan



Gambar 3.3 Desain Jari-jari tangan



Gambar 3.4 Desain Bagian Lengan Bawah



(a) (b) (c)

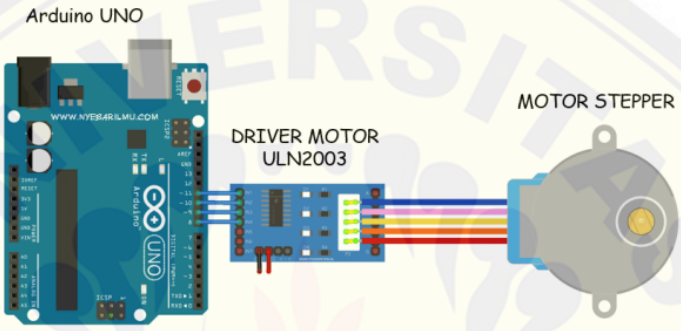
Gambar 3.5 Desain Konekor (a) 8, (b) 7 dan (c) FFX

Digital Repository Universitas Jember

3.4.2. Diagram Elektronika

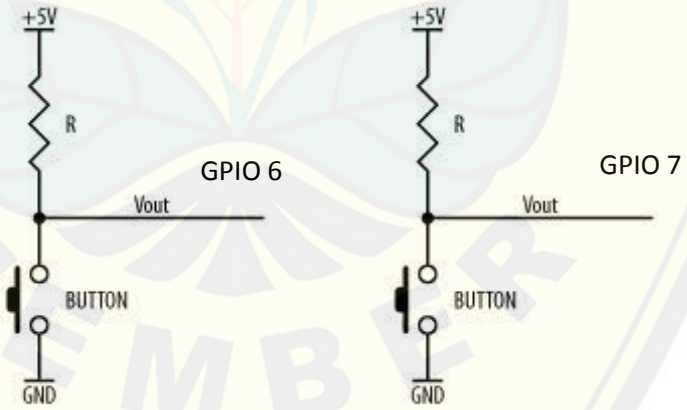
Pada rancangan elektronika robot yang dibuat menggunakan arduino UNO, driver motor ULN2003 dan motor *stepper* serta rangkaian push button sebagai pengendali manual dari robot tangan.

A. Diagram Arduino UNO dengan Motor Stepper



Gambar 3.6 Rangkaian elektronika aktuaktor

B. Diagram Ardunino UNO dengan *push button*



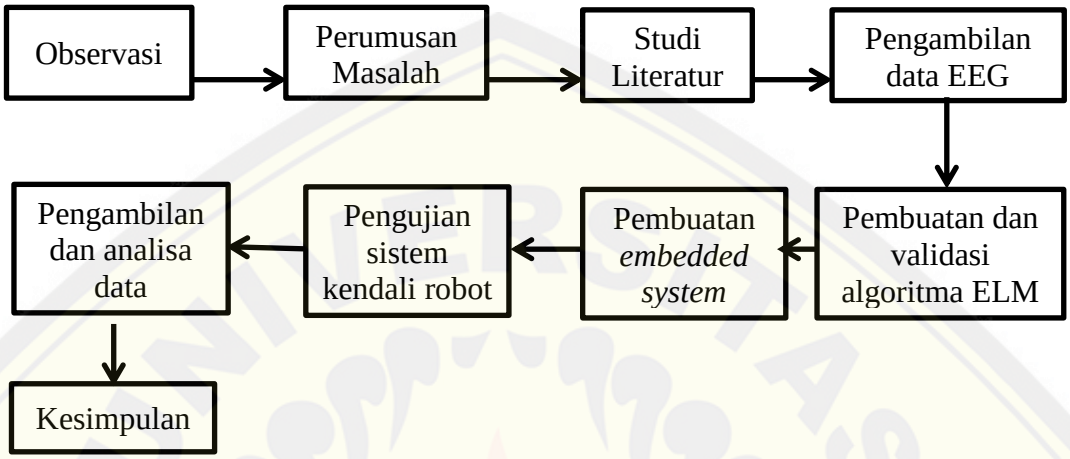
Gambar 3.7 Rangakaian elektronika kontrol

3.5. **Tahap penelitian**

Tahap penelitian di mulai dari observasi kemudian perumusan masalah, studi literatur, pengambilan data EEG, pembuatan dan validasi algoritma pendeteksi gerakan dengan ELM, pembuatan *embedded system* untuk pengendali

Digital Repository Universitas Jember

robot tangan, pengujian sistem kendali berbasis EEG pada robot tangan, pengambilan dan analisa data kemudian ditarik kesimpulan. Secara sederhana tahapan penelitian yang akan dilakukan dapat digambarkan sebagai berikut :



Gambar 3.8 Tahapan Penelitian.

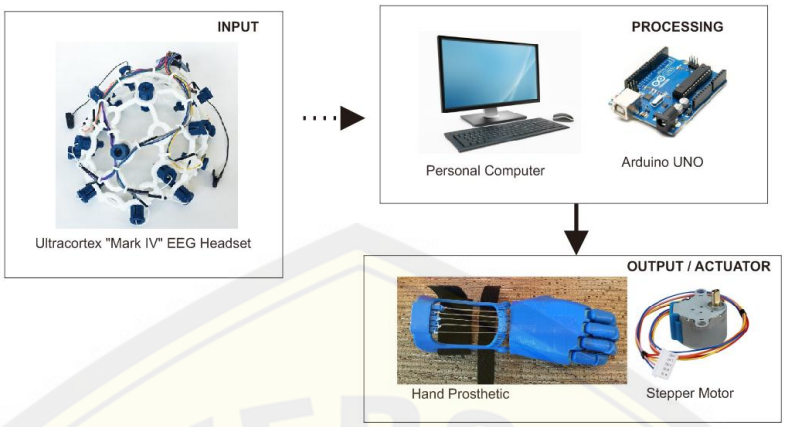
3.6. **Rancangan Sistem**

Rancangan sistem tersusun atas diagram blok sistem perangkat, diagram blok sistem kendali dan *flowchart* algoritma.

3.6.1. Diagram Blok Sistem Perangkat

Pada diagram blok ini terdapat 4 blok. Blok pertama adalah blok EEG *device* yang merupakan perangkat untuk menangkap sinyal EEG pada otak. Kemudian blok selanjutnya adalah blok PC dimana pada blok ini, sinyal yang berasal dari EEG *device* diproses dengan menggunakan algoritma yang diajukan oleh peneliti sehingga hasil keluaran dari blok ini dapat mengendalikan blok selanjutnya yaitu blok kontroller berupa Arduino. Pada blok Arduino ini, sinyal yang berasal dari PC yang berupa nilai bit diproses, kemudian Arduino akan melanjutkan kode tersebut untuk mengendalikan gerakan tangan pada blok terakhir pada sistem ini yaitu blok *handprothestic Robot.* Secara garis besar, dapat dilihat pada Gambar 3.9.

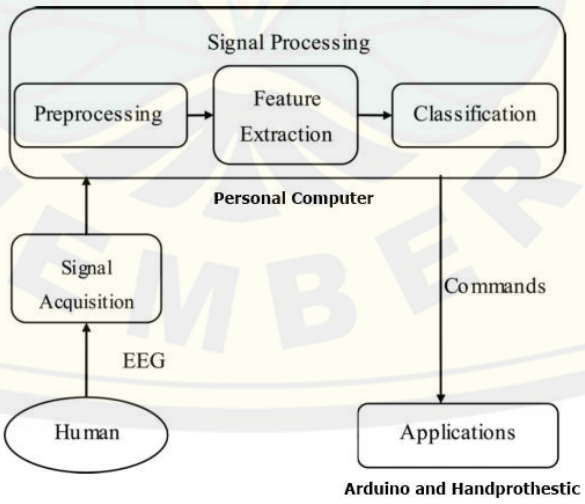
Digital Repository Universitas Jember



Gambar 3.9 Diagram Blok Sistem

3.6.2. Diagram Blok Sistem Kendali

Untuk dapat melakukan klasifikasi pada sinyal EEG, maka diperlukan rentetan algoritma tertentu agar sinyal EEG yang dikirim oleh perangkat EEG dapat diterjemahkan. Pada penelitian ini, algoritma tersebut diproses pada PC (*Personal Computer*). Oleh karena itu ada beberapa blok proses yang harus dilalui agar sinyal EEG tersebut dapat diterjemahkan yang urutan bloknya dapat dilihat pada Gambar 3.10.



Gambar 3.10. Desain Kendali Sistem

Pada Gambar 3.10, dapat dilihat bahwa pada penelitian terdiri dari 5 blok proses sebelum sinya EEG masuk kedalam pemroses algoritma. Pertama adalah proses pengambilan data pada sinyal EEG dengan menggunakan perangkat EEG.

Digital Repository Universitas Jember

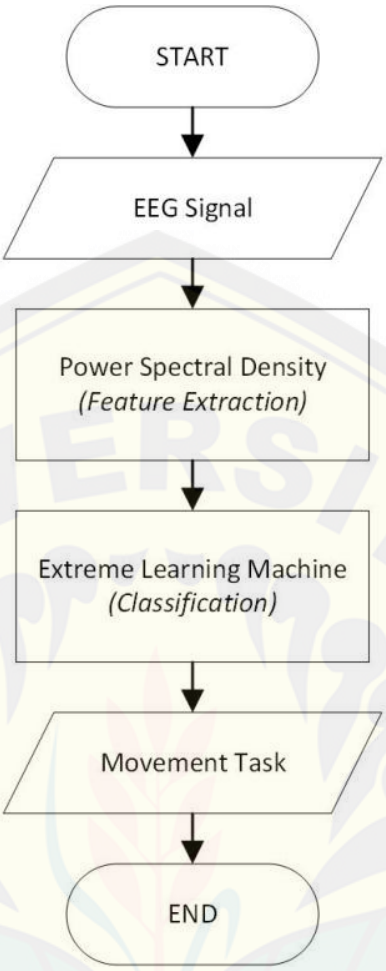
Sinyal EEG didapatkan dengan memasang elektroda yang ditempatkan pada permukaan kulit kepala pada channel C3, C4 dan dengan referensi Cz untuk mendapatkan sinyal *imagery movement tasks* (Saa, Gutierrez, Del, & Barranquilla, 2010). Setelah itu, sinyal yang didapat masuk kedalam blok *prepocessing signal* untuk menghilangkan artefak seperti gangguan saluran listrik, elektromiogram (EMG), elektrokardiogram (ECG), elektrookuramram (EOG), dan sinyal pembentukan tubuh lainnya. (Bi et al., 2013). Pada blok ini, sinyal tersebut disampling pada frekuensi 128 Hz dan di *filter* pada frekuensi antara 0.5 dan 30 Hz untuk mendapatkan sinyal yang spesifik berupa *imagery handtask movement* (Qiao, Wang, Li, & Tian, 2010). Setelah sinyal dibatasi pada frekuensi tersebut, kemudian sinyal masuk kedalam blok diagram *feature extraction* dengan tipe extractor berupa *Power Spectral Analysis*. Kemudian setelah didapatkan nilai berupa besar dari daya pada sinyal EEG yang masuk, sinyal diklasifikasi dengan menggunakan *Extreme Learning Machine* (ELM) yang kemudian keluaran dari proses klasifikasi sinyal ini adalah berupa nilai bit digital yang merupakan representasi dari sinyal EEG yang masuk kedalam blok proses. Lalu nilai bit digital tersebut diteruskan pada hardware processing berupa Arduino untuk menggerakan *handprothestic* robot sesuai dengan nilai bit yang diberikan oleh pengklasifikasi.

3.6.3. *Flowchart* Sistem

A. *Flowchart* Sistem Pengenalan Pola Sinyal EEG

Desain dari algoritma pada sistem ini dapat diamati seperti pada Gambar 7.4. Pada gambar tersebut, sinyal EEG yang dikirim akan dianalisa kerapatan daya spektralnya dengan menggunakan fitur ekstrasi PSD (*power spectral density*). Kemudian setelah pada masing – masing frekuensi diketahui besar kerapatan dayanya, data tersebut diumpankan pada ELM untuk kemudian diklasifikasi sesuai dengan pengkodean yang telah ditentukan. Pengkodean tersebut dapat dilihat pada Tabel 3.11.

Digital Repository Universitas Jember



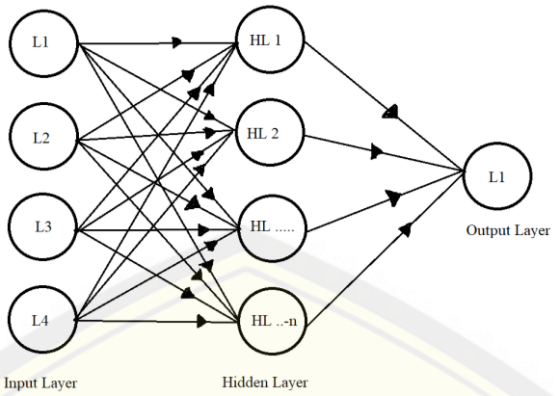
Gambar 3.11 *Flowchart* Sistem Pengenalan Pola Sinyal EEG.

Tabel 3.2 Pengkodean Sinyal EEG dengan *Classifier* ELM

No.	Kode Biner	Gerakan
1	0	Terbuka
2	1	Menggenggam

Dengan *output* layer sebanyak 2 pada ELM terdapat beberapa pergerakan robot yang dihasilkan. Robot akan menggenggam saat bit bernilai 1 dan robot tangan terbuka saat bit bernilai 0.

Digital Repository Universitas Jember



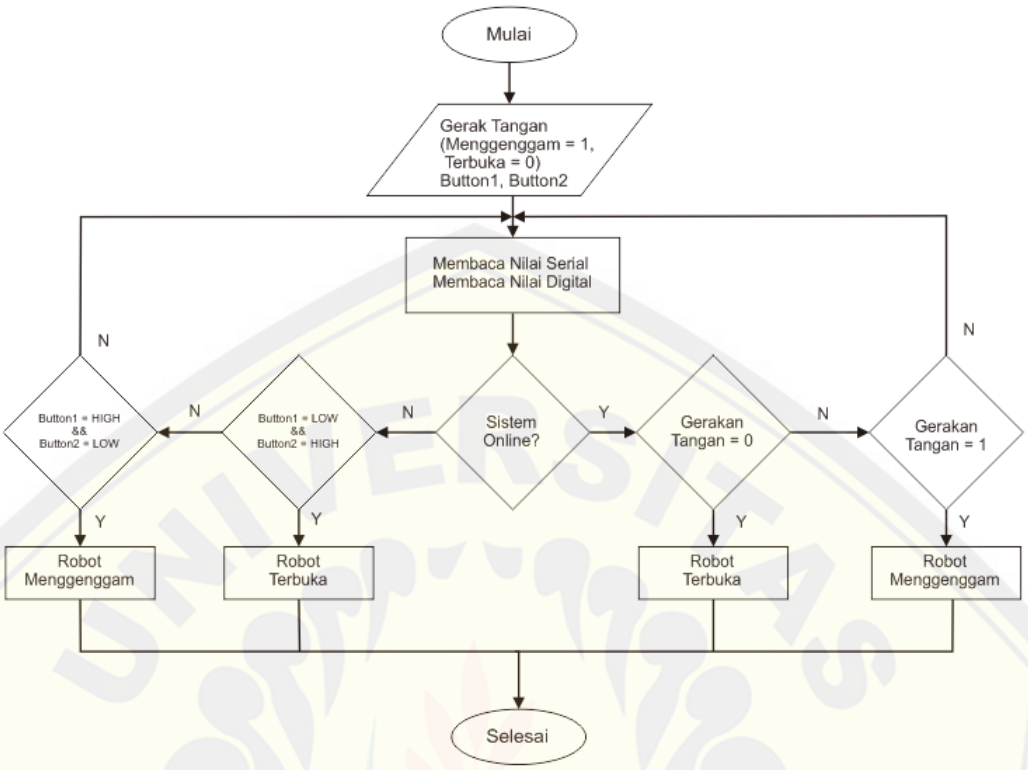
Gambar 3.12 Rancangan struktur ELM

Pada pengklasifikasian menggunakan ELM, penggunaan algoritma ELM penting sekali untuk menentukan terlebih dahulu jumlah dari *input layer, hidden layer dan output*. Pada penelitian ini, perancangan nilai masukkan atau *input* sejumlah 4, sedangkan untuk jumlah masukkan dikarenakan terdapat jenis gerakan robot sebanyak 2, maka jumlah bit gerakan adalah 2^1 atau sebanyak 1 *output* struktur. Sedangkan untuk jumlah *layer* yang digunakan sebanyak 1 *hidden layer* dengan jumlah *neuron* 10.

B. *Flowchart* Sistem Robot Tangan

Pada robot tangan menggunakan arduino sebagai komponen pengendali. Pada arduino memiliki masukan nilai serial yang diperoleh dari personal computer sebagai data online serta masukan nilai dari *push button* yaitu Button1 dan Button 2. Kerja dari robot tangan dimulai saat robot pertama kali dijalankan. Kemudian kontroler akan menginisiasikan gerak tangan, Button1 dan Button 2. Kontroler akan membaca nilai masukan yang berasal dari gerak tangan yang diperoleh dari masukan serial (*online*) dan kondisi *push button.* Saat terbaca nilai serial maka itu berarti sistem berjalan secara *online.* Sehingga apabila diberi nilai serial 0 maka robot akan terbuka. Apabila diberi nilai 1 maka robot akan menggenggam. Saat terbaca nilai digital dari *push button* maka sistem akan berjalan secara tidak *online*. Sehingga apabila nilai Button1 *high* dan Button2 *low* maka robot akan menggenggam. Apabila nilai Button1 *low* dan Button2 *high* maka robot akan terbuka.

Digital Repository Universitas Jember



Gambar 3.13 *Flowchart* Sistem Pengenalan Pola Sinyal EEG.

3.7. **Pengambilan Data**

3.7.1. Persiapan Ruangan Pengambilan Data

Pada ruangan untuk pengambilan data diperlukan kondisi yang tenang dan dapat menjaga privasi dari responden yang bersangkutan. Sehingga perlu adanya pengaturan khusus untuk proses pengambilan data.

Gambar 3.14 Ruang Pantau Peneliti

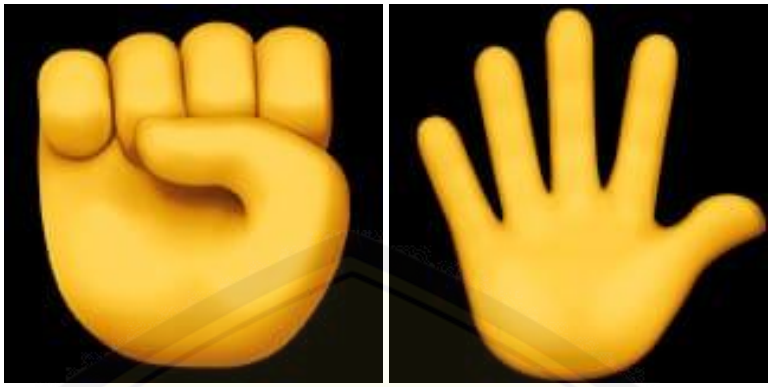
Digital Repository Universitas Jember

Gambar 3.15 Ruang Pengambilan Data pada Subyek

3.7.2. Prosedur Pelaksanaan Pengambilan Data

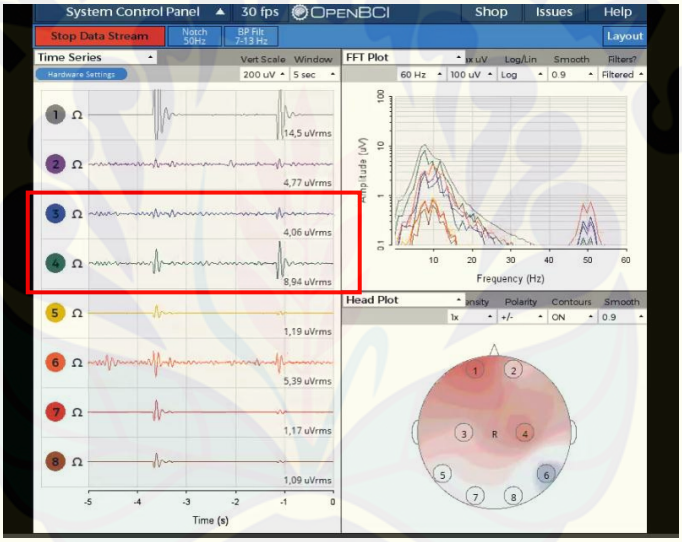
Pengambilan data *electroenchelpalograph* menggunakan devais EEG Ultracortex Mark IV *EEG Headset.* Pengambilan data dilakukan pada channel C3 dan C4 yang merupakan bagian otak yang memunculkan sinyal berdasarkan respon gerakan tubuh dalam hal ini adalah tangan. Percobaan dilakukan terhadap 4 responden yang diminta untuk menggerakkan tangan kiri mereka dengan gerakan menggenggam dan terbuka secara kala. Gerakan menggenggam dan membuka ini dilakukan secara bergantian dengan jeda waktu setiap gerakan selama 7 detik dan dilakukan sebanyak 15 kali untuk setiap gerakan. Percobaan dilakukan menggunakan *software* OpenBCI GUI yang dapat menampilkan grafik dan menyimpan data dalam bentuk *.txt*. Pada perobaan pengambilan data, responden diminta melakukan gerakan menggenggam dan membuka melalui media gambar seperti pada gambar 3.16. Hasil grafik yang ditampilkan oleh *software* OpenBCI GUI terlihat seperti gambar 3.17.

Digital Repository Universitas Jember



(a) (b)

Gambar 3.16 Media gambar untuk instruksi (a) menggenggam dan (b) membuka



Gambar 3.17 Grafik Sinyal EEG pada OpenBCI GUI.

3.8. **Variabel Pengujian Sistem**

Pada tahap pengujian sistem yang akan dilakukan oleh peneliti, terdapat tiga variabel performa yang akan diambil nilainya, antara lain yaitu:

1. Kecepatan dalam proses pelatihan data

   Pada tahap ini, penguji akan menguji kecepatan dari pengolahan data saat pelatihan pada pengklasifikasian yang digunakan, yaitu pada ELM. Pada pengklasifikasian akan diuji dengan sejumlah data, kemudian dibandingkan waktu lama proses pengolahan data pelatihan sampai mendapatkan nilai *error* minimum.

Digital Repository Universitas Jember

2. Kecepatan dalam mengolah data

   Pada tahap ini, penguji akan menguji kecepatan dari pengolahan data pada saat kedua pengklasifikasian diimplementasikan pada *plant.* Pengujian ini didasarkan dari waktu respon sinyal mulai saat subjek mulai membayangkan dan menggerakan salah satu bagian gerakan yang diperintahkan oleh peneliti kemudian dihitung waktunya sampai dengan *prothestic hand* robot memberikan respon gerakan yang sesuai.

3. Tingkat akurasi

   Pengujian pada tahap ini adalah pengujian tingkat akurasi dari pengklasifikasian. Tingkat akurasi tersebut dalam satuan persen dimana untuk dapat mengetahui besar nilai akurasi dari pengklasifikasian adalah pada saat subjek diberikan perintah untuk menggerakan motoriknya sesuai dengan perintah peneliti. Subjek akan diberikan 30 perintah yang berbeda, kemudian peneliti mencatat berapa kali keberhasilan respon dari *prothestic hand* robot dalam merepresentasikan informasi. Selain itu peneliti juga menganalisa penyebab dari gagal respon pengklasifikasi dengan melihat tingat keberhasilan pengklasifikasian menggunakan analisa statistik MSE (*Mean Square Error*) dan MAPE (*Mean Absolute Percentage Error*) yang dihitung pada saat proses *traning* selesai. (Bi et al., 2013; Cambria et al., 2013).

Digital Repository Universitas Jember

BAB 5. PENUTUP

5.1. Kesimpulan

Setelah melakukan perencanaan dan perancangan *hand prosthestic robot* dan sistem pengenalan pola gerak tangan sederhana berbasis EEG dapat diambil kesimpulan sebagai berikut:

1. *Hand prosthestic robot* menggunakan desain Flexy-Hand dengan motor stepper sebagai penggeraknya dengan menggunakan perbandingan poros yang tepat dapat melakukan gerakan menggenggam dan terbuka dengan baik.
2. Klasifikasi sinyal EEG mengunakan metode ELM dengan memberikan bobot PSD dari setiap sinyal masukan memberikan akurasi training sebesar 50% pada tangan kanan dan 62,5 % pada tangan kiri.
3. Performa klasifikasi menggunakan metode ELM memberikan hasil test akurasi sebesar 50%

5.2. Saran

Berdasarkan penelitian tentang sistem pengenalan pola gerak tangan sederhana berbasis EEG yang telah dilakukan dapat diberikan saran-saran apabila hendak melanjutkan penelitian ini yaitu :

1. Subyek penelitian harus dalam kondisi prima sehingga dapat memperoleh data yang baik.
2. Penggunaan devais EEG harus dilakukan dengan hati-hati untuk mendapatkan data yang valid.
3. Pada metode ELM semakin banyak data training maka akan semakin baik.

Digital Repository Universitas Jember

DAFTAR PUSTAKA

Asanza, Víctor, Enrique Peláez, Javier Díaz, and Francis Loayza. 2018. "EMG Signal Processing with Clustering Algorithms for Motor Gesture Tasks." IEEE Third Ecuador Technical Chapters Meeting (ETCM).

Bhattacharyya, S., Khasnobish, A., & et al. (2011). Performance Analysis of Left/Right Hand Movement Classification from EEG Signal by Intelligent Algorithms. IEEE SSCI 2011 - Symposium Series on Computational Intelligence - CCMB 2011: 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain, 1–8. https://doi.org/10.1109/CCMB.2011.5952111

Bi, L., Fan, X., & Liu, Y. (2013). EEG-Based Brain-Controlled Mobile Robots : A Survey. IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, 43(2), 161–176.

Burke-, L. A. (n.d.). Overview of the anatomy of the upper and lower limbs.

Cambria, E., Huang, G.-B., Kasun, L. L. C., Zhou, H., Vong, C. M., Lin, J., … Liu, J. (2013). Extreme Learning Machines. IEEE Intelligent Systems, 28(6), 30–59. https://doi.org/10.1109/MIS.2013.140

Fok, S., Member, S., Schwartz, R., Wronkiewicz, M., Holmes, C., Somers, T., … Leuthardt, E. (2011). An EEG-Based Brain Computer Interface for Rehabilitation and Restoration of Hand Control Following Stroke Using Ipsilateral Cortical Physiology. 33rd Annual International Conference of the IEEE EMBS Boston, Massachusetts USA, August 30 - September 3, 2011, 6277–6280. https://doi.org/10.1109/IEMBS.2011.6091549

Hayashi, T., Yokoyama, H., & et al. (2017). Prediction of Individual Finger Movements for Motor Execution and Imagery : an EEG Study. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 3020–3023. https://doi.org/10.1109/SMC.2017.8123088

Huang, G., Zhu, Q., & Siew, C. (2004). Extreme Learning Machine : A New Learning Scheme of Feedforward Neural Networks. IEEE International Joint Conference on Neural Networks, 2, 985–990. https://doi.org/10.1109/IJCNN.2004.1380068

Kementrian Kesehatan RI. (2014). Situasi Penyandang Disabilitas. Buletin Jendela Data & Informasi Kesehatan, Semester 2(1), 1–5. https://doi.org/10.1007/s13398-014-0173-7.2

Loza, C. A., Philips, G. R., & et al. (2014). Classification of hand movement direction based on EEG high-gamma activity. 2014 36th Annual International Conference of the IEEE Engineering in Medicine and

Digital Repository Universitas Jember

Biology Society, EMBC 2014, 6509–6512. https://doi.org/10.1109/EMBC.2014.6945119

Luu, Loc Gia, Nam Phuong Nguyen, and Toi Vo Van. 2018. "Design a Customizable Low-Cost, Matlab Based Wireless Data Acquisition System for Real-Time Physiological Signal Processing." IFMBE Proceedings 63: 313–17. https://doi.org/10.1007/978-981-10-4361-1_52.

Malathi, V., Marimuthu, N. S., & Baskar, S. (2010). Intelligent approaches using support vector machine and extreme learning machine for transmission line protection. Neurocomputing, 73(10–12), 2160–2167. https://doi.org/10.1016/j.neucom.2010.02.001

Minh Dao, Duc, Pham Dang Phuoc, Tran Xuan Tuy, and Tram Thuy Le. 2017. "Research on Reading Muscle Signals from the EMG Sensor during Knee Flexion — Extension Using the Arduino Uno Controller." International Conference on Advanced Technologies for Communications (ATC), 270–73. https://doi.org/10.1109/ATC.2017.8167632.

Qiao, X., Wang, Y., Li, D., & Tian, L. (2010). Feature Extraction and Classifier Evaluation of EEG for Imaginary Hand Movements. Proceedings - 2010 6th International Conference on Natural Computation, ICNC 2010, 4(Icnc), 2112–2116. https://doi.org/10.1109/ICNC.2010.5582453

Rechy-ramirez, E. J., & Hu, H. (2015). Bio-signal based control in assistive robots : a survey. Digital Communications and Networks, 1(2), 85–101. https://doi.org/10.1016/j.dcan.2015.02.004

Saa, J. F. D., Gutierrez, M. S., Del, U., & Barranquilla, N. (2010). EEG Signal Classification Using Power Spectral Features and linear Discriminant Analysis : A Brain Computer Interface Application. Eighth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2010), (2003), 1–7.

Schwarz, A., Ofner, P., Pereira, J., Sburlea, A. L., & Gernot R. Müller-Putz. (2017). Decoding natural reach -and-grasp actions from human EEG. Journal of Neural Engineering, 15(16005).

Tino, P., Benuskova, L., & Sperduti, A. (2015). Artificial Neural Network Models. Springer Handbook of Computational Intelligence, 8(3), 455–472.

Veneman, J. F., Kruidhof, R., & et al. (2007). Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 15(3), 379–386.

Yuan, Q., Zhou, W., Li, S., & Cai, D. (2011). Epileptic EEG classification based on extreme learning machine and nonlinear features. Epilepsy Research, 96(1–2), 29–38. https://doi.org/10.1016/j.eplepsyres.2011.04.013

Digital Repository Universitas Jember

Yulianto, E., Susanto, A., Widodo, T. S., & Wibowo, S. (2013). Spektrum Frekuensi Sinyal EEG Terhadap Pergerakan Motorik dan Imajinasi Pergerakan Motorik. Forum Teknik, 35, 21–32.

Zhang, R., Huang, G.-B., Sundararajan, N., & Saratchandran, P. (2007). Multi-category classification using an Extreme Learning Machine for microarray gene expression cancer diagnosis. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 4(3), 485–495. https://doi.org/10.1109/tcbb.2007.1012

Digital Repository Universitas Jember

Lampiran

1. Pembuatan Robot Tangan Tiruan

Gambar 1. Bentuk Asli Desain Flexy-Hand

Gambar 2. Bentuk Akhir Dari Hand Prosthetic Robot

Digital Repository Universitas Jember

2. Pengambilan Data

Gambar 3. Posisi Subyek Saat Pengambilan Data

3. *Listing* Program Arduino

4. *Listing* Program Filter PSD Matlab

- Pengambilan data.m

```
eeg = DataKanan((1+(2500*(a-1))):(2500*a),:)
eeg = transpose(eeg)
smoothdata = eegfilt(eeg,250,7,13)
```

- Eegfilt.m

```
function [smoothdata,filtwts] =
eegfilt(data,srate,locutoff,hicutoff,epochframes,filtorder,revfilt
,firtype,causal)

if nargin<4
    fprintf('');
    help eegfilt
    return
end

%if ~exist('firls')
%   error('*** eegfilt() requires the signal processing toolbox.
***');
```

Digital Repository Universitas Jember


```
%end

[chans frames] = size(data);
if chans > 1 & frames == 1,
    help eegfilt
    error('input data should be a row vector.');
end
nyq            = srate*0.5;  % Nyquist frequency
%MINFREQ = 0.1/nyq;
MINFREQ = 0;

minfac         = 3;     % this many (lo)cutoff-freq cycles in
filter
min_filtorder  = 15;    % minimum filter length
trans          = 0.15; % fractional width of transition zones

if locutoff>0 & hicutoff > 0 & locutoff > hicutoff,
    error('locutoff > hicutoff ???\n');
end
if locutoff < 0 | hicutoff < 0,
    error('locutoff | hicutoff < 0 ???\n');
end

if locutoff>nyq,
    error('Low cutoff frequency cannot be > srate/2');
end

if hicutoff>nyq
    error('High cutoff frequency cannot be > srate/2');
end

if nargin<6
    filtorder = 0;
end
if nargin<7
    revfilt = 0;
end
if nargin<8
    firtype = 'firls';
end
if nargin<9
    causal = 0;
end

if strcmp(firtype, 'firls')
    warning('Using firls to estimate filter coefficients. We
recommend that you use fir1 instead, which yields larger
attenuation. In future, fir1 will be used by default!');
end

if isempty(filtorder) | filtorder==0,
    if locutoff>0,
        filtorder = minfac*fix(srate/locutoff);
    elseif hicutoff>0,
        filtorder = minfac*fix(srate/hicutoff);
```

Digital Repository Universitas Jember


```
    end

    if filtorder < min_filtorder
        filtorder = min_filtorder;
    end
end

if nargin<5
    epochframes = 0;
end
if epochframes ==0,
    epochframes = frames;    % default
end
epochs = fix(frames/epochframes);
if epochs*epochframes ~= frames,
    error('epochframes does not divide frames.\n');
end

if filtorder*3 > epochframes,   % Matlab filtfilt() restriction
    fprintf('eegfilt(): filter order is %d. ',filtorder);
    error('epochframes must be at least 3 times the filtorder.');
end
if (1+trans)*hicutoff/nyq > 1
    error('high cutoff frequency too close to Nyquist frequency');
end;

if locutoff > 0 & hicutoff > 0,    % bandpass filter
    if revfilt
        fprintf('eegfilt() - performing %d-point notch
filtering.\n',filtorder);
    else
        fprintf('eegfilt() - performing %d-point bandpass
filtering.\n',filtorder);
    end;
    fprintf('            If a message, ''Matrix is close to
singular or badly scaled,'' appears,\n');
    fprintf('            then Matlab has failed to design a good
filter. As a workaround, \n');
    fprintf('            for band-pass filtering, first highpass
the data, then lowpass it.\n');
    if strcmp(firtype, 'firls')
        f=[MINFREQ (1-trans)*locutoff/nyq locutoff/nyq
hicutoff/nyq (1+trans)*hicutoff/nyq 1];
        fprintf('eegfilt() - low transition band width is %1.1g
Hz; high trans. band width, %1.1g Hz.\n',(f(3)-f(2))*srate/2,
(f(5)-f(4))*srate/2);
        m=[0       0                          1            1
0                       0];
    elseif strcmp(firtype, 'fir1')
        filtwts = fir1(filtorder, [locutoff,
hicutoff]./(srate/2));
    end
elseif locutoff > 0                % highpass filter
    if locutoff/nyq < MINFREQ
        error(sprintf('eegfilt() - highpass cutoff freq must be >
%g Hz\n\n',MINFREQ*nyq));
```

Digital Repository Universitas Jember


```
    end
    fprintf('eegfilt() - performing %d-point highpass
filtering.\n',filtorder);
    if strcmp(firtype, 'firls')
        f=[MINFREQ locutoff*(1-trans)/nyq locutoff/nyq 1];
        fprintf('eegfilt() - highpass transition band width is
%1.1g Hz.\n',(f(3)-f(2))*srate/2);
        m=[   0             0                   1      1];
    elseif strcmp(firtype, 'fir1')
        filtwts = fir1(filtorder, locutoff./(srate/2), 'high');
    end
elseif hicutoff > 0                 %  lowpass filter
    if hicutoff/nyq < MINFREQ
        error(sprintf('eegfilt() - lowpass cutoff freq must be >
%g Hz',MINFREQ*nyq));
    end
    fprintf('eegfilt() - performing %d-point lowpass
filtering.\n',filtorder);
    if strcmp(firtype, 'firls')
        f=[MINFREQ hicutoff/nyq hicutoff*(1+trans)/nyq 1];
        fprintf('eegfilt() - lowpass transition band width is
%1.1g Hz.\n',(f(3)-f(2))*srate/2);
        m=[     1           1             0                0];
    elseif strcmp(firtype, 'fir1')
        filtwts = fir1(filtorder, hicutoff./(srate/2));
    end
else
    error('You must provide a non-0 low or high cut-off
frequency');
end
if revfilt
    if strcmp(firtype, 'fir1')
        error('Cannot reverse filter using ''fir1'' option');
    else
        m = ~m;
    end;
end;

if strcmp(firtype, 'firls')
    filtwts = firls(filtorder,f,m); % get FIR filter coefficients
end

smoothdata = zeros(chans,frames);
for e = 1:epochs                % filter each epoch, channel
    for c=1:chans
        try
            if causal
                 smoothdata(c,(e-1)*epochframes+1:e*epochframes) =
filter(  filtwts,1,data(c,(e-1)*epochframes+1:e*epochframes));
            else smoothdata(c,(e-1)*epochframes+1:e*epochframes) =
filtfilt(filtwts,1,data(c,(e-1)*epochframes+1:e*epochframes));
            end;
        catch,
            if causal
```

Digital Repository Universitas Jember


```
                smoothdata(c,(e-1)*epochframes+1:e*epochframes) =
filter(  filtwts,1,double(data(c,(e-
1)*epochframes+1:e*epochframes)));
            else smoothdata(c,(e-1)*epochframes+1:e*epochframes) =
filtfilt(filtwts,1,double(data(c,(e-
1)*epochframes+1:e*epochframes)));
            end;
        end;
        if epochs == 1
            if rem(c,20) ~= 0, fprintf('.');
            else               fprintf('%d',c);
            end
        end
    end
end
fprintf('\n');
```

- FFT.m

```
function  [fy,f]=FFT(y,Fs)

if nargin<2
    error(' Sampling frequency is required to sompute (PSD,F(y)) !
');
end

% In case that the input vector is matrix :  Maping with vect{} .
y=y(:).';
L=length(y);

%  (2^N) :Number of points for computing the FFT
N=ceil(log2(length(y)));

% FFT
fy=fft(y,2^N)/(L/2);
%-------------------------------------------------
% for length<1000 one can replace fft with function :
% fy=Fast_Fourier_Transform(y,2^N)/(L/2); (line 84)
%-------------------------------------------------

% Amplitude adjustment by checking for complex input y
if isreal(y)==0
    fy=fy/2;
end

% PSD
Power=fy.*conj(fy);
%Phase Angle
phy=angle(fy);

%  Frequency axis
f=(Fs/2^N)*(0:2^(N-1)-1);

%if nargin==4
```

Digital Repository Universitas Jember


```
% Figures-----------------------------------------------------------
---------
ff1=figure;
plot(f,Power(1:2^(N-1)),'r'),  xlabel('  Frequency (Hz)'),
ylabel(' Magnitude (w)'),
title('  Power Spectral Density'), grid on;
set(ff1,'Name','PSD');

ff2=figure;
plot(f,10*log10(Power(1:2^(N-1))),'r'),  xlabel('  Frequency
(Hz)'), ylabel(' Magnitude  (dB)'),
title('  Power Spectral Density, logarithmic scale '), grid on;
set(ff2,'Name','10*log10(PSD)');

ff3=figure;
plot(f,sqrt(Power(1:2^(N-1))),'r'),  xlabel('  Frequency (Hz)'),
ylabel('|F(Y)|'),
title('  Amplitude Spectrum'), grid on;
set(ff3,'Name','|F(y)|');


ff4=figure;
plot(f,phy(1:2^(N-1)),'b'), xlabel(' Frequency (Hz)'), ylabel('
arg(F(Y))'),
title(' Phase spectrum'), grid on;
set(ff4,'Name','arg(F(Y))');

time = zeros(1,3584)
time = -2000 + (7000/3584)

for i = 2:3584
    time(:,i) = time(:,i-1) + (7000/3584)
end
```

5. *Listing* Program ELM Matlab

```
function [TrainingTime, TestingTime, TrainingAccuracy,
TestingAccuracy] = elm(TrainingData_File, TestingData_File,
Elm_Type, NumberofHiddenNeurons, ActivationFunction)
%%%%%%%%%%% Macro definition
REGRESSION=0;
CLASSIFIER=1;

%%%%%%%%%%% Load training dataset
train_data=load(TrainingData_File);
T=train_data(:,1)';
P=train_data(:,2:size(train_data,2))';
clear train_data;                                   %   Release
raw training data array

%%%%%%%%%%% Load testing dataset
test_data=load(TestingData_File);
```

Digital Repository Universitas Jember


```
TV.T=test_data(:,1)';
TV.P=test_data(:,2:size(test_data,2))';
clear test_data;                                    %   Release
raw testing data array

NumberofTrainingData=size(P,2);
NumberofTestingData=size(TV.P,2);
NumberofInputNeurons=size(P,1);

if Elm_Type~=REGRESSION
    %%%%%%%%%%%% Preprocessing the data of classification
    sorted_target=sort(cat(2,T,TV.T),2);
    label=zeros(1,1);                               %   Find and
save in 'label' class label from training and testing data sets
    label(1,1)=sorted_target(1,1);
    j=1;
    for i = 2:(NumberofTrainingData+NumberofTestingData)
        if sorted_target(1,i) ~= label(1,j)
            j=j+1;
            label(1,j) = sorted_target(1,i);
        end
    end
    number_class=j;
    NumberofOutputNeurons=number_class;

    %%%%%%%%%% Processing the targets of training
    temp_T=zeros(NumberofOutputNeurons, NumberofTrainingData);
    for i = 1:NumberofTrainingData
        for j = 1:number_class
            if label(1,j) == T(1,i)
                break;
            end
        end
        temp_T(j,i)=1;
    end
    T=temp_T*2-1;

    %%%%%%%%%% Processing the targets of testing
    temp_TV_T=zeros(NumberofOutputNeurons, NumberofTestingData);
    for i = 1:NumberofTestingData
        for j = 1:number_class
            if label(1,j) == TV.T(1,i)
                break;
            end
        end
        temp_TV_T(j,i)=1;
    end
    TV.T=temp_TV_T*2-1;

end                                                 %   end if of
Elm_Type

%%%%%%%%%%% Calculate weights & biases
start_time_train=cputime;
```

Digital Repository Universitas Jember


```
%%%%%%%%%%% Random generate input weights InputWeight (w_i) and
biases BiasofHiddenNeurons (b_i) of hidden neurons
InputWeight=rand(NumberofHiddenNeurons,NumberofInputNeurons)*2-1;
BiasofHiddenNeurons=rand(NumberofHiddenNeurons,1);
tempH=InputWeight*P;
clear P;                                                %   Release
input of training data
ind=ones(1,NumberofTrainingData);
BiasMatrix=BiasofHiddenNeurons(:,ind);                  %   Extend the
bias matrix BiasofHiddenNeurons to match the demention of H
tempH=tempH+BiasMatrix;

%%%%%%%%%%% Calculate hidden neuron output matrix H
switch lower(ActivationFunction)
    case {'sig','sigmoid'}
        %%%%%%%% Sigmoid
        H = 1 ./ (1 + exp(-tempH));
    case {'sin','sine'}
        %%%%%%%% Sine
        H = sin(tempH);
    case {'hardlim'}
        %%%%%%%% Hard Limit
        H = double(hardlim(tempH));
    case {'tribas'}
        %%%%%%%% Triangular basis function
        H = tribas(tempH);
    case {'radbas'}
        %%%%%%%% Radial basis function
        H = radbas(tempH);
        %%%%%%%% More activation functions can be added here
end
clear tempH;                                            %   Release
the temparary array for calculation of hidden neuron output matrix
H

%%%%%%%%%%% Calculate output weights OutputWeight (beta_i)
OutputWeight=pinv(H') * T';                             %
implementation without regularization factor //refer to 2006
Neurocomputing paper
%OutputWeight=inv(eye(size(H,1))/C+H * H') * H * T';   % faster
method 1 //refer to 2012 IEEE TSMC-B paper
%implementation; one can set regularizaiton factor C properly in
classification applications
%OutputWeight=(eye(size(H,1))/C+H * H') \ H * T';      % faster
method 2 //refer to 2012 IEEE TSMC-B paper
%implementation; one can set regularizaiton factor C properly in
classification applications

%If you use faster methods or kernel method, PLEASE CITE in your
paper properly:

%Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang,
"Extreme Learning Machine for Regression and Multi-Class
Classification," submitted to IEEE Transactions on Pattern
Analysis and Machine Intelligence, October 2010.
```

Digital Repository Universitas Jember


```
end_time_train=cputime;
TrainingTime=end_time_train-start_time_train        %   Calculate
CPU time (seconds) spent for training ELM

%%%%%%%%%%% Calculate the training accuracy
Y=(H' * OutputWeight)';                             %   Y: the
actual output of the training data
if Elm_Type == REGRESSION
    TrainingAccuracy=sqrt(mse(T - Y))               %   Calculate
training accuracy (RMSE) for regression case
end
clear H;

%%%%%%%%%%% Calculate the output of testing input
start_time_test=cputime;
tempH_test=InputWeight*TV.P;
clear TV.P;             %   Release input of testing data
ind=ones(1,NumberofTestingData);
BiasMatrix=BiasofHiddenNeurons(:,ind);              %   Extend the
bias matrix BiasofHiddenNeurons to match the demention of H
tempH_test=tempH_test + BiasMatrix;
switch lower(ActivationFunction)
    case {'sig','sigmoid'}
        %%%%%%%% Sigmoid
        H_test = 1 ./ (1 + exp(-tempH_test));
    case {'sin','sine'}
        %%%%%%%% Sine
        H_test = sin(tempH_test);
    case {'hardlim'}
        %%%%%%%% Hard Limit
        H_test = hardlim(tempH_test);
    case {'tribas'}
        %%%%%%%% Triangular basis function
        H_test = tribas(tempH_test);
    case {'radbas'}
        %%%%%%%% Radial basis function
        H_test = radbas(tempH_test);
        %%%%%%%% More activation functions can be added here
end
TY=(H_test' * OutputWeight)';                       %   TY: the
actual output of the testing data
end_time_test=cputime;
TestingTime=end_time_test-start_time_test           %   Calculate
CPU time (seconds) spent by ELM predicting the whole testing data

if Elm_Type == REGRESSION
    TestingAccuracy=sqrt(mse(TV.T - TY))            %   Calculate
testing accuracy (RMSE) for regression case
end

if Elm_Type == CLASSIFIER
%%%%%%%%%% Calculate training & testing classification accuracy
    MissClassificationRate_Training=0;
    MissClassificationRate_Testing=0;
```

Digital Repository Universitas Jember


```
    for i = 1 : size(T, 2)
        [x, label_index_expected]=max(T(:,i));
        [x, label_index_actual]=max(Y(:,i));
        if label_index_actual~=label_index_expected

MissClassificationRate_Training=MissClassificationRate_Training+1;
        end
    end
    TrainingAccuracy=1-MissClassificationRate_Training/size(T,2)
    for i = 1 : size(TV.T, 2)
        [x, label_index_expected]=max(TV.T(:,i));
        [x, label_index_actual]=max(TY(:,i));
        if label_index_actual~=label_index_expected

MissClassificationRate_Testing=MissClassificationRate_Testing+1;
        end
    end
    TestingAccuracy=1-MissClassificationRate_Testing/size(TV.T,2)
end
```