

BAB II KAJIAN TEORI

2.1. Teori Dasar

2.1.1. Kecerdasan Buatan

Kecerdasan buatan (*Artificial intelligence*) merupakan ilmu yang mempelajari tentang bagaimana komputer melakukan penalaran berdasarkan pengetahuan atau pengalaman dan menarik kesimpulan sesuai dengan masukan yang diberikan. Kusumadewi mengemukakan bahwa kecerdasan buatan sebagai salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia (Kusumadewi, 2003). Menurut Nagy, Kecerdasan buatan adalah ilmu yang digunakan untuk membangun kecerdasan menggunakan solusi perangkat keras dan perangkat lunak (Nagy, 2018). Sedangkan Haenlein & Kaplan mendefinisikan kecerdasan buatan sebagai kemampuan sistem untuk menafsirkan data eksternal dengan benar, belajar dari data tersebut, dan menggunakan pembelajaran tersebut untuk mencapai tujuan dan tugas tertentu melalui adaptasi yang fleksibel (Haenlein & Kaplan, 2019).

2.1.2. Jaringan Saraf Tiruan

Kusumadewi dalam bukunya mendefinisikan Jaringan Saraf Tiruan (JST) sebagai salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi non-linear, klasifikasi data *cluster* dan regresi non-parametrik atau sebuah simulasi dari koleksi model jaringan saraf biologis (Kusumadewi, 2003).

Menurut Bell JST pada dasarnya dimodelkan pada arsitektur paralel otak hewan, yang belum tentu manusia. Jaringan didasarkan pada bentuk input dan *output* yang sederhana (Bell, 2015). Sedangkan Gurney mendefinisikan JST sebagai kumpulan elemen, unit atau *node* pemrosesan yang saling berhubungan, yang fungsinya didasarkan pada neuron hewan. Kemampuan pemrosesan jaringan disimpan dalam kekuatan koneksi interunit, atau bobot, yang diperoleh dengan proses adaptasi atau belajar dari serangkaian pola pelatihan (Gurney, 2004).

2.2. Teori Khusus

2.2.1. Disabilitas

Disabilitas merupakan keadaan yang membatasi kemampuan mental dan atau fisik seseorang (“Kamus Besar Bahasa Indonesia (KBBI) Daring,” n.d.). Disabilitas meliputi meliputi keterbatasan aktivitas, gangguan (masalah pada fungsi tubuh), dan pembatasan partisipasi. Penyandang Disabilitas adalah setiap orang yang

mengalami keterbatasan fisik, intelektual, mental, dan atau sensorik dalam jangka waktu lama yang dalam berinteraksi dengan lingkungan dapat mengalami hambatan dan kesulitan untuk berpartisipasi secara penuh dan efektif dengan warga negara lainnya berdasarkan kesamaan hak (“Undang-Undang Republik Indonesia Nomor 8 Tahun 2016 Tentang Penyandang Disabilitas,” 2016) .

Ragam Penyandang Disabilitas sesuai dengan yang tercantum dalam UU No. 8 Tahun 2016 diantaranya Penyandang Disabilitas fisik, Penyandang Disabilitas intelektual, Penyandang Disabilitas mental, Penyandang Disabilitas sensorik. Penyandang Disabilitas fisik yaitu terganggunya fungsi gerak, antara lain amputasi, lumpuh layuh atau kaku, *paraplegi*, *celebral palsy* (CP), akibat stroke, akibat kusta, dan orang kecil. Penyandang Disabilitas intelektual yaitu terganggunya fungsi pikir karena tingkat kecerdasan di bawah rata-rata, antara lain lambat belajar, disabilitas grahita dan *down syndrom*. Penyandang Disabilitas mental yaitu terganggunya fungsi pikir, emosi, dan perilaku, antara lain: psikososial di antaranya *skizofrenia*, bipolar, depresi, *anxietas*, dan gangguan kepribadian; dan disabilitas perkembangan yang berpengaruh pada kemampuan interaksi sosial di antaranya autisme dan hiperaktif. Penyandang Disabilitas sensorik yaitu terganggunya salah satu fungsi dari panca indera, antara lain disabilitas netra, disabilitas rungu, dan atau disabilitas wicara.

2.2.2. Arduino Uno Rev3

Arduino adalah papan tunggal mikrokontroler dengan perangkat keras, yang terdiri dari papan perangkat keras sumber terbuka (*open source*) yang dirancang

dengan Atmel AVR Mikrocontroller (Schwartz & Manickum, 2015). Pajankar mengemukakan bahwa Arduino adalah platform dan ekosistem prototipe elektronik sumber terbuka, yang didasarkan pada lingkungan perangkat keras dan perangkat lunak yang mudah digunakan (Pajankar, 2018).



Gambar 2.1 Arduino Uno Rev3
(Sumber : <http://bit.ly/2wjVbvd>)

Uno adalah bahasa Italia untuk nomor satu, Uno digunakan sebagai nama untuk rilis versi 1.0 dari perangkat lunak Arduino. R3 yang diartikan sebagai revisi ketiga, berkaitan dengan revisi fitur di papan Arduino itu sendiri, yang mencakup pembaruan, penyempurnaan, dan perbaikan.

Tabel 2.1 di menunjukkan ringkasan spesifikasi dari Arduino Uno Rev 3.

Tabel 2.1 Spesifikasi Arduino Uno Rev3

Spesifikasi	Nilai
Mikrokontroler	ATmega328P
Tegangan operasi	5V
Tegangan input (direkomendasikan)	7-12V
Tegangan input (batas)	6-20V
Pin I/O digital	14 (6 pin dapat digunakan sebagai output PWM)
PWM digital I/O pin	6

Tabel 2.1 (Lanjutan) Spesifikasi Arduino Uno Rev3

Spesifikasi	Nilai
Pin input analog	6
Arus DC per pin I/O	20 mA
Arus DC untuk 3.3V pin	50 mA
Memori <i>flash</i>	32 KB (ATmega328P) 0.5 KB digunakan untuk <i>bootloader</i>
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Kecepatan <i>clock</i>	16 MHz
LED_BUILTIN	13
Panjang	68.6 mm
Lebar	53.4 mm
Berat	25 g

(Sumber : <http://bit.ly/2wjVbvd>)

Berikut merupakan bagian-bagian dari Arduino Uno Rev3.

1. *Power*

Papan Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Daya eksternal dapat berasal dari adaptor AC ke DC atau baterai. Papan Arduino Uno dapat beroperasi dengan suplai eksternal dari 6 hingga 20V. Namun, jika diberikan kurang dari 7V, pin 5V dapat memasok kurang dari 5V dan papan mungkin menjadi tidak stabil. Jika menggunakan lebih dari 12V, pengatur tegangan bisa menjadi terlalu panas dan merusak papan. Kisaran yang disarankan adalah 7V hingga 12V. Pin *power* diantaranya adalah pin Vin yang digunakan untuk tegangan input ke papan Arduino ketika menggunakan sumber daya eksternal, pin 5V menghasilkan tegangan 5V, pin 3.3V menghasilkan tegangan 3,3V, pin GND digunakan sebagai *ground*, dan pin IOREF memberikan referensi tegangan yang digunakan mikrokontroler.

2. Memori

ATmega328 memiliki memori 32 KB (dengan 0,5 KB digunakan oleh *bootloader*). Selain itu juga memiliki 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan *library* EEPROM).

3. Input dan *output*

Masing-masing dari 14 pin pada Arduino Uno dapat digunakan sebagai input atau *output*. Beberapa pin memiliki fungsi khusus. Pin 3, 5, 6, 9, 10, dan 11 memiliki fungsi khusus yaitu menyediakan *output* PWM 8-bit. Pin 0 (RX) dan pin 1 (TX) digunakan untuk menerima (RX) dan mengirimkan (TX) data serial *Transistor-Transistor Logic* (TTL).

2.2.3. *MindWave Mobile 2*

MindWave Mobile 2 merupakan perangkat yang dengan aman dapat mengukur dan mengeluarkan spektrum daya EEG (gelombang alfa, gelombang beta, dll), NeuroSky *eSense meter* (*Attention* dan *Meditation*) dan kedipan mata ("Mindwave," n.d.). Perangkat terdiri dari *headset*, klip telinga, dan lengan sensor. Referensi *headset* dan *ground* elektroda ada di klip telinga dan elektroda EEG ada di lengan sensor, ditempatkan di dahi di atas mata. Perangkat ini menggunakan baterai AAA tunggal dengan masa pakai baterai selama 8 jam.



Gambar 2.2 Neurosky *MindWave Mobile 2* EEG Headset
(Sumber : <http://bit.ly/2SBd1m8>)

MindWave Mobile 2 dirancang untuk aplikasi pendidikan dan hiburan dengan mengubah perubahan kondisi mental menjadi tindakan. *MindWave Mobile 2* melaporkan kondisi mental pemakai dalam bentuk algoritma kepemilikan NeuroSky, termasuk algoritma *Attention* dan *Meditation eSense*. NeuroSky *MindWave Mobile 2* dapat digunakan dengan video game yang didukung, perangkat lunak pendidikan, dan banyak aplikasi lainnya (*MindWave Mobile 2: User Guide*, 2018).

Electroencephalogram (EEG) adalah tes yang merekam aktivitas listrik di otak menggunakan cakram logam kecil (elektroda) yang ditempelkan di kulit kepala (“EEG (electroencephalogram),” n.d.). Sel-sel otak yang juga dikenal sebagai neuron, saling berkomunikasi dengan memancarkan impuls listrik dan aktif setiap saat bahkan ketika sedang tidur, mengelola cara berpikir dan cara berinteraksi. Aktivitas ini ditampilkan sebagai garis bergelombang pada rekaman EEG. Hasil EEG menunjukkan perubahan aktivitas otak yang mungkin berguna

dalam mendiagnosis kondisi otak (“Reading Your Brainwaves: Understanding the Basics of EEG,” n.d.).

Beberapa jenis gelombang otak yang umum adalah sebagai berikut (Demos, 2019):

1. Gelombang *delta*

Gelombang *delta* dikaitkan dengan tidur, tidur tanpa mimpi, tidur *Non-Rapid Eye Movement* (N-REM), dan tidak sadar. Gelombang *delta* dominan muncul pada bayi. Gelombang *delta* yang terlalu tinggi mengindikasikan ketidakmampuan untuk meremajakan tubuh, ketidakmampuan untuk merevitalisasi otak, atau kurang tidur. Gelombang *delta* yang terlalu rendah mengindikasikan cedera otak atau *traumatic brain injury* (TBI), masalah belajar, ketidakmampuan untuk berpikir, *attention deficit hyperactivity disorder* (ADHD) berat.

2. Gelombang *theta*

Gelombang *theta* dikaitkan dengan kreativitas, koneksi emosional, intuisi, relaksasi. Anak-anak memiliki amplitudo *theta* yang lebih tinggi daripada orang dewasa. Gelombang *theta* yang terlalu tinggi mengindikasikan ADHD, depresi, hiperaktif, impulsif, dan kurang perhatian. Gelombang *theta* yang terlalu rendah mengindikasikan kecemasan, kesadaran emosional yang buruk, dan stres.

3. Gelombang *alpha*

Gelombang *alpha* dikaitkan dengan santai, tetapi tidak mengantuk, tenang, sadar. Gelombang *alpha* yang terlalu tinggi mengindikasikan melamun, ketidakmampuan untuk fokus, terlalu santai. Gelombang *alpha* yang terlalu rendah

mengindikasikan kecemasan, stres tinggi, susah tidur, *Obsessive compulsive disorder* (OCD).

4. Gelombang *beta*

Gelombang *beta* dengan nilai yang tepat dikaitkan dengan fokus sadar, ingatan, pemecahan masalah. Gelombang *beta* yang terlalu tinggi mengindikasikan kecemasan yang berlebihan, gairah tinggi, ketidakmampuan untuk rileks, dan stres. Gelombang *beta* yang terlalu rendah mengindikasikan ADHD, melamun, depresi, kognisi buruk.

5. Gelombang *gamma*

Gelombang *gamma* dengan nilai yang tepat dikaitkan dengan pengikatan indera, kognisi, pemrosesan informasi, pembelajaran (mempelajari materi atau hal baru), persepsi, tidur REM. Gelombang *gamma* yang terlalu tinggi mengindikasikan kecemasan, gairah tinggi, stres. Individu yang mengalami gangguan mental dan memiliki ketidakmampuan belajar cenderung memiliki aktivitas *gamma* yang lebih rendah daripada rata-rata. Gelombang *gamma* yang terlalu rendah mengindikasikan ADHD, depresi, dan ketidakmampuan belajar.

2.2.3.1. *ThinkGear*

ThinkGear adalah teknologi di dalam setiap produk atau mitra NeuroSky yang memungkinkan perangkat untuk berinteraksi dengan gelombang otak pemakai. Ini termasuk sensor yang menyentuh dahi, titik kontak dan referensi yang terletak di klip telinga, dan *chip on-board* yang memproses semua data. Baik

gelombang otak mentah atau *raw brainwaves* dan *eSense meter* (*Attention* dan *Meditasi*) dihitung pada *chip ThinkGear*.

2.2.3.2. *eSense*

eSense adalah algoritma kepemilikan NeuroSky untuk mengkarakterisasi keadaan mental. Untuk menghitung *eSense*, teknologi Neurosky *Thinkgear* memperkuat sinyal *raw brainwave* dan menghilangkan kebisingan sekitar dan gerakan otot. Algoritma *eSense* kemudian diterapkan pada sinyal yang tersisa, menghasilkan nilai meteran *eSense meter* yang dapat diinterpretasikan. Nilai-nilai *eSense meter* tidak menggambarkan angka yang pasti, melainkan menggambarkan rentang aktivitas (*MindWave Mobile 2: User Guide*, 2018).

eSense meter adalah cara untuk menunjukkan seberapa efektif pengguna melakukan perhatian (mirip dengan konsentrasi) atau meditasi (mirip dengan relaksasi). Untuk setiap jenis *eSense* (*Attention*, dan *Meditation*), nilai meter dilaporkan pada skala *eSense* relatif dari 1 hingga 100. Pada skala ini, nilai antara 40 hingga 60 pada saat tertentu dianggap "netral" dan serupa dengan "*baselines*" yang ditetapkan dalam teknik pengukuran gelombang otak konvensional (meskipun metode untuk menentukan *ThinkGear baselines* adalah hak milik dan mungkin berbeda dari metode lain). Nilai dari 60 hingga 80 dianggap "sedikit lebih tinggi", dan dapat ditafsirkan sebagai tingkat yang cenderung lebih tinggi dari normal (tingkat *Attention* atau *Meditation* yang mungkin lebih tinggi dari normal untuk orang tertentu). Nilai dari 80 hingga 100 dianggap "tinggi". Demikian pula, nilai antara 20 hingga 40 menunjukkan tingkat "rendah" dari *eSense*, sementara nilai

antara 1 hingga 20 menunjukkan tingkat "sangat rendah" tingkat *eSense*. Tingkat dapat menunjukkan keadaan gangguan, agitasi, atau kelainan, sesuai dengan kebalikan dari masing-masing *eSense* (*MindWave Mobile 2: User Guide*, 2018).

2.2.3.3. *Attention eSense*

Indikator *Attention eSense* menunjukkan intensitas tingkat "fokus" atau "perhatian" pengguna, seperti yang terjadi selama konsentrasi intens dan aktivitas mental terarah (tetapi stabil). Nilainya berkisar dari 0 hingga 100. Gangguan, pikiran yang berkeliaran, kurang fokus, atau kecemasan dapat menurunkan tingkat meteran *Attention* (*MindWave Mobile 2: User Guide*, 2018).

2.2.3.4. *Meditation eSense*

Indikator *Meditation eSense* menunjukkan tingkat "ketenangan" atau "relaksasi" mental pengguna. Nilainya berkisar dari 0 hingga 100. Nilai *Meditation* meningkat ketika ketika pengguna merilekskan pikiran dan berkurang ketika gelisah atau stres. Meditasi adalah ukuran keadaan mental seseorang, bukan tingkat fisik, jadi sekadar merelaksasikan semua otot tubuh mungkin tidak akan segera menghasilkan tingkat Meditasi yang tinggi. Namun, bagi kebanyakan orang dalam kebanyakan situasi normal, rileks tubuh sering membantu pikiran untuk rileks juga. Meditasi berhubungan dengan berkurangnya aktivitas oleh proses mental aktif di otak. Menutup mata dapat mematikan aktivitas mental yang memproses gambar dari mata. Jadi menutup mata sering kali merupakan metode yang efektif untuk

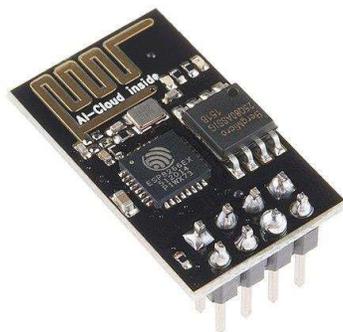
meningkatkan level meteran *Meditation*. Gangguan, pikiran yang berkeliaran, kecemasan, agitasi, dan rangsangan sensorik dapat menurunkan tingkat meteran *Meditation* (*MindWave Mobile 2: User Guide*, 2018).

2.2.3.5. *Blink Detection*

Blink Detection menunjukkan tingkat kedipan mata pengguna. Nilai yang lebih tinggi menunjukkan kedipan "kuat", sedangkan nilai kecil menunjukkan kedipan "lebih ringan" atau "lebih lemah". Nilai ini berkisar dari 0 hingga 255 ("EEG Algorithms," n.d.).

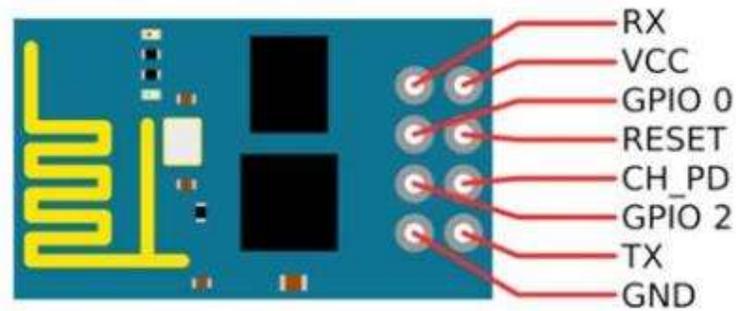
2.2.4. ESP8266

Pada dasarnya, ESP8266 merupakan modul Wifi (Towaha, 2018). ESP8266 memiliki dukungan untuk 802.11 b/g/n dan beroperasi pada frekuensi 2,4 GHz. Modul ini mendukung *open networks*, WPA dan WPA2 dan dapat digunakan sebagai *Station* atau *Access Point* atau gabungan keduanya.



Gambar 2.3 ESP8266-01
(Sumber : <http://bit.ly/39NcfZg>)

ESP8266 biasa digunakan untuk pembuatan IoT. Perangkat ini terintegrasi dengan prosesor 32-bit yang berjalan pada kecepatan 80 MHz (dapat di-*overclock* hingga 160 MHz).



Gambar 2.4 ESP8266 *pinout*
Sumber: (Towaha, 2018)

Tabel 2.2 berikut berisi keterangan *pinout* untuk ESP8266.

Tabel 2.2 ESP8266 *pinout*

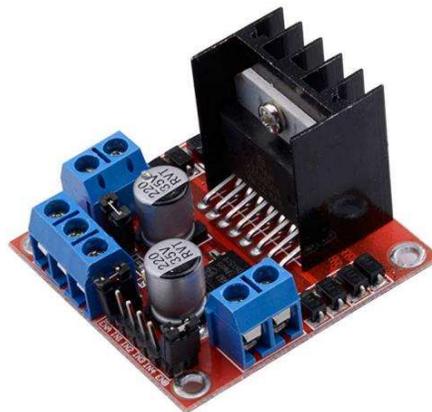
Nomor	Nama	Nama alternatif	Deskripsi/Kegunaan
1	GND	-	Sebagai <i>ground</i>
2	TX	GPIO-1	Mentransmisikan data
3	GPIO 2	-	<i>Input/output</i> pin untuk tujuan umum
4	CH_EN	-	<i>Chip enable</i>
5	GPIO 0	Flash	<i>Input/output</i> pin untuk tujuan umum
6	Reset	-	Mengatur ulang modul.
7	RX	GPIO-3	Menerima data
8	VCC	-	<i>Pin power</i> (3.3V)

(Sumber : (Towaha, 2018))

ESP8266 juga dapat diprogram menggunakan Arduino IDE atau AT-Command dan mendukung komunikasi *serial* sehingga kompatibel dengan banyak platform pengembangan seperti Arduino.

2.2.5. L298N

L298N adalah *motor driver H-Bridge* ganda yang memungkinkan pengendalian arah dan kecepatan sekaligus untuk dua motor DC. (Bi, Kapoor, & Bhatia, 2018). *Motor driver* digunakan untuk mengendalikan motor DC. Alasan utama diperlukannya *motor driver* yaitu karena komponen seperti motor tidak dapat dihubungkan langsung ke Arduino karena tidak akan mendapatkan arus yang cukup dari Arduino. Ketika motor berjalan maka motor akan mengkonsumsi arus tinggi. *Motor driver* memberikan keamanan pada mikrokontroler sehingga aliran tidak akan mengalir ke mikrokontroler.



Gambar 2.5 L298N
(Sumber : <https://amzn.to/3bN4zaW>)

2.2.6. Android

Android pada dasarnya adalah sistem operasi komputasi seluler, misalnya untuk perangkat seluler seperti ponsel pintar dan tablet (Vandome, 2017). Menurut Rollins & Sandberg Android adalah sistem operasi berbasis Linux yang

mengandalkan Java untuk menyederhanakan pengembangan perangkat lunak dan menggunakan *Extensible Markup Language* (XML) untuk menyederhanakan tugas pengkodean (Rollins & Sandberg, 2013). Android adalah sistem operasi *open source*, yang berarti pengembang dan produsen dapat bekerja dengan kode sumber untuk menyesuaikannya dengan kebutuhan dan perangkat mereka sendiri (selama memenuhi persyaratan dan standar tertentu) (Vandome, 2017).



Gambar 2.6 Logo Android
(Sumber : <http://bit.ly/323Hpsu>)

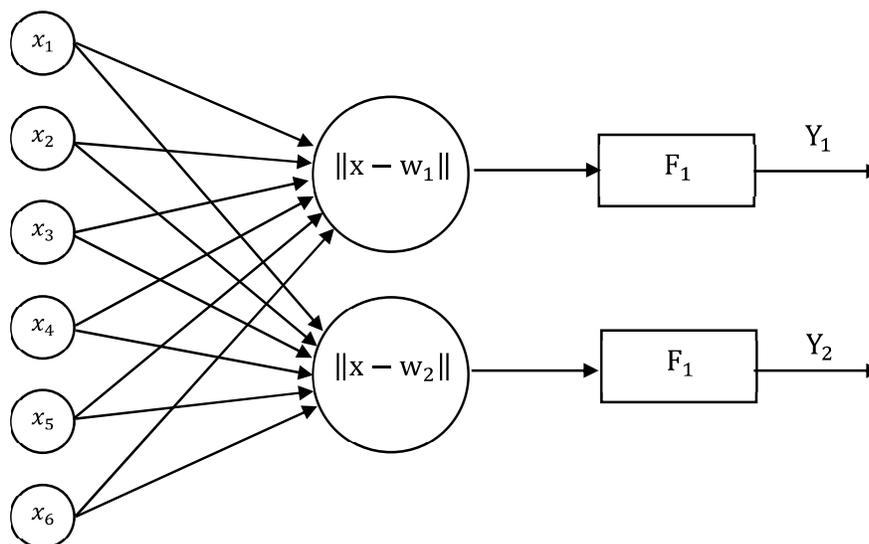
2.3. Model JST

Model JST yang digunakan dalam penelitian ini adalah *Learning Vector Quantization* (LVQ). LVQ adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi (Kusumadewi, 2003). LVQ merupakan pengembangan dari metode kuantisasi vektor dimana setiap kelas sampel input vektor diwakili oleh set vektor *codebook*. Satu-satunya tujuan LVQ adalah untuk menggambarkan batas kelas dengan aturan tetangga terdekat (Kohonen, 1995).

Lapisan kompetitif akan belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika 2 vektor input mendekati

sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama.

LVQ menggunakan arsitektur jaringan dengan lapisan kompetitif (*competitive layer net*). Pada JST dengan LVQ terdiri dari $x_1..x_n$ input yang terhubung dengan lapisan kompetitif dan menuju lapisan keluaran. Lapisan keluaran merupakan kelas atau target yang ditentukan selama proses pelatihan. Arsitektur JST dengan algoritma LVQ dapat dilihat pada gambar 2.7.



Gambar 2.7 Arsitektur JST dengan LVQ
(Sumber : (Kusumadewi, 2003))

Berikut merupakan algoritma dari LVQ (Kusumadewi, 2003):

1. Tetapkan nilai untuk bobot (w), maksimum *epoch* (MaxEpoch), *error* minimum yang diharapkan (ϵ) dan *learning rate* (α).
2. Masukkan Input : $x(m,n)$ dan Target : $(T1,n)$ dimana m merupakan jumlah input, n merupakan jumlah data.
3. Inisialisasi kondisi awal $epoch = 0$, $\epsilon = 1$
4. Lakukan jika: $(epoch < MaxEpoch)$ atau $(\alpha < \epsilon)$

1. Epoch = Epoch + 1
2. Kerjakan untuk $i = 1$ sampai n
 1. Tentukan J sedemikian hingga $\|x - w_j\|$ minimum (sebagai C_j)
 2. Perbaiki w_j dengan ketentuan:
 1. Jika $T = C_j$ maka: $w_j(\text{baru}) = w_j(\text{lama}) + \alpha (x - w_j(\text{lama}))$
 2. Jika $T \neq C_j$ maka: $w_j(\text{baru}) = w_j(\text{lama}) - \alpha (x - w_j(\text{lama}))$
3. Kurangi nilai α :

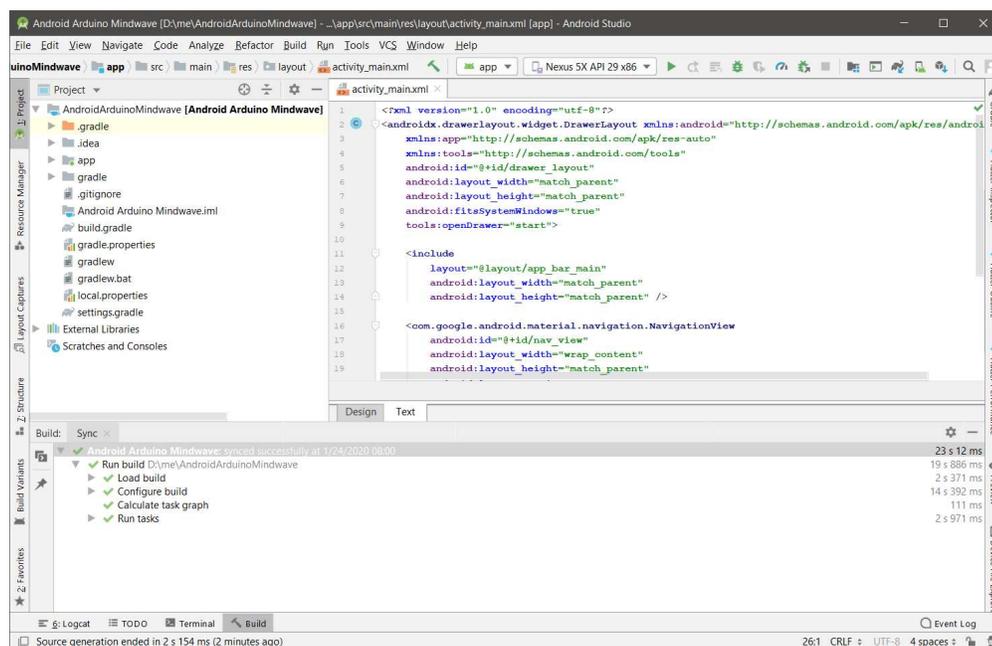
2.4. Software Pendukung dan Perangkat Pemodelan

2.4.1. Java

Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan. Java berjalan di atas sebuah mesin penerjemah (*interpreter*) yang diberi nama *Java Virtual Machine* (JVM). JVM akan membaca kode bit (*bytecode*) dalam *file* .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang portabel karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (*platform independent*) (S & Shalahuddin, 2015).

2.4.2. Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi/*integrated development environment* (IDE), mirip dengan ADT *Bundle* atau *Eclipse* dengan ADT *Plugin*, tetapi didasarkan pada IntelliJ IDEA, yang merupakan alat yang sangat populer untuk pengembangan Java (Jr., Darcey, & Conder, 2014).

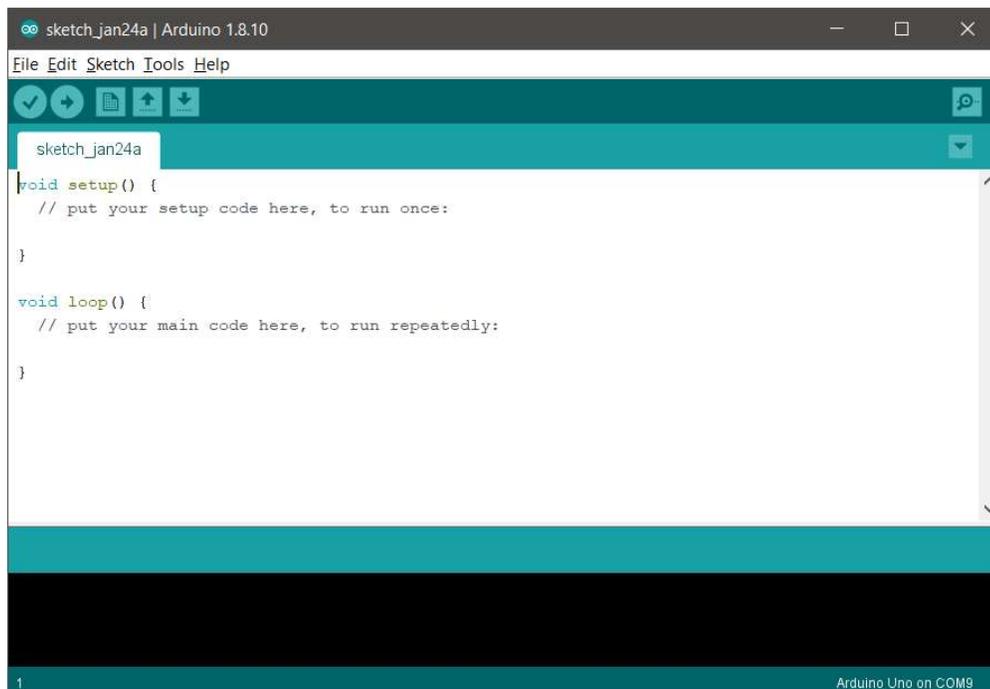


Gambar 2.8 Antarmuka pengguna Android Studio
(Sumber : Data Penelitian, 2020)

Menurut Drongelen Android Studio adalah IDE yang disarankan untuk mengembangkan aplikasi Android dan tersedia secara gratis bagi siapa saja yang mengembangkan aplikasi Android profesional (Drongelen, 2015). Horton mendefinisikan Android Studio sebagai alat yang menyederhanakan pengembangan Android dan menggunakan JDK untuk memungkinkan pengguna menulis dan membangun program Java (Horton, 2015).

2.4.3. Arduino IDE

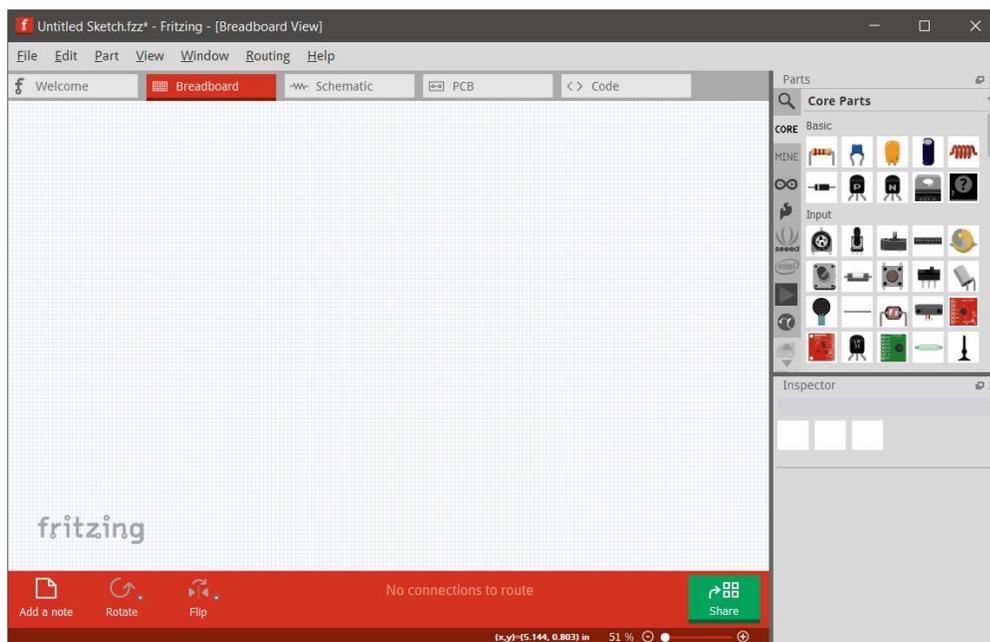
Arduino IDE adalah aplikasi *cross-platform* yang dikembangkan dengan Java yang dapat digunakan untuk mengembangkan, meng-*compile*, dan mengunggah program ke papan Arduino (Desai, 2015).



Gambar 2. 9 Antarmuka pengguna Arduino IDE
(Sumber : Data Penelitian, 2020)

2.4.4. Fritzing

Fritzing adalah program yang memungkinkan kita mengubah desain sirkuit menjadi representasi grafis langsung di komputer. Manfaat utamanya adalah sirkuit yang dirancang dalam *Fritzing* terlihat seperti aslinya. Ini akan memberikan representasi yang intuitif, langsung, dan segera dapat dikenali dari *breadboard* dan komponen yang dihubungkan (Allan & Bradford, 2013).



Gambar 2.10 Antarmuka pengguna *Fritzing*
(Sumber : Data Penelitian, 2020)

2.4.5. *StarUML*

StarUML merupakan alat pemodelan sumber terbuka (Lee, 2008) . *StarUML* mendukung pemodelan UML yang diadopsi sebagai standar oleh *Object Management Group* (OMG). Selain itu *StarUML* juga mendukung pemodelan diagram ER, *flowchart* diagram dan *Data Flow Diagram* (DFD).

2.4.6. *Blender*

Blender adalah perangkat lunak sumber terbuka yang menyediakan paket pembuatan grafik 3D paling lengkap, termasuk alat untuk pemodelan, penteksturan, *shading*, animasi, komposisi, *rendering*, pengeditan video, dan banyak lagi (Villar, 2014). Salah satu hal yang membuat *Blender* berbeda dibandingkan dengan

perangkat lunak 3D lain adalah karena *Blender* tersedia secara gratis dan merupakan perangkat lunak sumber terbuka. Menurut Gumster *Blender* adalah program grafis komputer yang mampu menghasilkan gambar diam dan animasi berkualitas tinggi menggunakan geometri tiga dimensi (Gumster, 2015). *Blender* juga merupakan perangkat lunak yang *cross-platform* dan dapat berjalan dengan baik di Linux, Windows, dan Macintosh.



Gambar 2.11 Logo *blender*
(Sumber : <http://bit.ly/37DhXvh>)

2.4.7. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah notasi dan semantik untuk visualisasi, konstruksi dan dokumentasi model untuk pengembangan perangkat lunak berorientasi objek yang telah distandarisasi oleh *Object Management Group (OMG)* (Maier, 2004). UML bukan metodologi atau proses melainkan notasi (Hunt, 2003). UML tidak terikat dengan alat pengembangan spesifik, bahasa pemrograman khusus, atau platform target spesifik tempat sistem yang akan dikembangkan harus digunakan (Seidl et al., 2015).

Dalam UML, model direpresentasikan secara grafis dalam bentuk diagram. Diagram memberikan pandangan tentang bagian realitas yang dijelaskan oleh model. Ada diagram yang menyatakan pengguna mana yang menggunakan

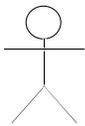
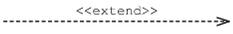
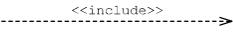
fungsionalitas dan diagram yang menunjukkan struktur sistem tetapi tanpa menentukan implementasi konkret.

Diagram UML yang digunakan dalam penelitian ini diantaranya use case diagram, class diagram, dan deployment diagram. Berikut merupakan ringkasan beberapa diagram uml tersebut.

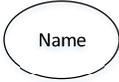
1. *Use case diagram*

Use case menggambarkan fungsionalitas yang diharapkan dari sistem yang akan dikembangkan tetapi tidak membahas rincian spesifik implementasi (Seidl et al., 2015). *Use case* mencakup sejumlah fungsi yang dieksekusi ketika menggunakan sistem. Unit fungsionalitas yang disediakan sistem untuk penggunaannya disebut *use case*. Tabel 2.3 menunjukkan *node* dan *path* dalam *use case diagram*.

Tabel 2.3 *node* dan *path* dalam *class diagram*

Tipe <i>node/path</i>	Notasi	Deskripsi singkat
<i>Actor (default)</i>		Menentukan peran yang dimainkan oleh pengguna atau sistem lain yang berinteraksi dengan subjek. Aktor dapat mewakili peran yang dimainkan oleh pengguna manusia, perangkat keras eksternal, atau subjek lainnya.
<i>Extend</i>		Hubungan ini menentukan bahwa perilaku <i>use case</i> dapat diperluas oleh perilaku <i>use case</i> lain (biasanya tambahan).
<i>Include</i>		Hubungan <i>include</i> mendefinisikan bahwa <i>use case</i> berisi perilaku yang didefinisikan dalam <i>use case</i> lain.

Tabel 2.3 (Lanjutan) *node* dan *path* dalam *class diagram*

Tipe <i>node/path</i>	Notasi	Deskripsi singkat
<i>UseCase</i>		Spesifikasi dari serangkaian tindakan yang dilakukan oleh suatu sistem, yang menghasilkan hasil yang dapat diamati, yang biasanya, bernilai bagi satu atau lebih aktor atau pemangku kepentingan lain dari sistem.

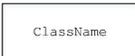
(Sumber : (“OMG Unified Modeling Language (OMG UML), Superstructure Version 2.4.1,” 2011))

2. *Class diagram*

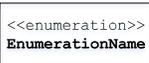
Class diagram digunakan untuk memodelkan struktur statis suatu sistem, sehingga menggambarkan elemen-elemen sistem dan hubungan di antara elemen-elemen tersebut yang tidak berubah seiring waktu (Seidl et al., 2015).

Tabel 2.4 menunjukkan *node* dan *edge* yang biasa digunakan dalam *class diagram*.

Tabel 2.4 *node* dan *edge class diagram*

Tipe <i>node/path</i>	Notasi	Deskripsi singkat
<i>Association</i>		Hubungan semantik yang dapat terjadi di antara instansi.
<i>Aggregation</i>		Bentuk asosiasi khusus yang digunakan untuk menyatakan bahwa instansi dari satu kelas adalah bagian dari instansi dari kelas lain.
<i>Class</i>		Serangkaian objek yang memiliki spesifikasi fitur, batasan, dan semantik yang sama. <i>Class</i> memiliki fitur atribut dan operasi.
<i>Composition</i>		Menunjukkan bahwa properti diagregasi secara komposit.
<i>Dependency</i>		Hubungan yang menandakan bahwa satu atau serangkaian elemen model memerlukan elemen model lain untuk spesifikasi atau implementasinya.

Tabel 2.4 (Lanjutan) *node* dan *edge class diagram*

Tipe <i>node/path</i>	Notasi	Deskripsi singkat
<i>Generalization</i>		Proses mengekstraksi karakteristik bersama dari dua kelas atau lebih, dan menggabungkannya menjadi <i>superclass</i> umum.
<i>Interface</i>		Pengelompokan yang mewakili deklarasi serangkaian fitur dan kewajiban publik yang koheren. Antarmuka menentukan kontrak, setiap instansi dari <i>classifier</i> yang menerapkan antarmuka harus memenuhi kontrak tersebut.
<i>InterfaceRealization</i>		InterfaceRealization adalah hubungan realisasi khusus antara <i>Classifier</i> dan <i>Interface</i> .
<i>Enumeration</i>		Tipe data yang nilainya disebutkan dalam model sebagai literasi enumerasi.

(Sumber : (“OMG Unified Modeling Language (OMG UML), Superstructure Version 2.4.1,” 2011))

Selain notasi diatas dalam dari *class diagram* juga terdapat *VisibilityKind*.

VisibilityKind adalah tipe enumerasi yang mendefinisikan literal untuk menentukan visibilitas elemen dalam suatu model. Tabel 2.5 berikut merupakan ringkasan deskripsi untuk *VisibilityKind*.

Tabel 2.5 *VisibilityKind class diagram*

Nama	Notasi	Deskripsi singkat
<i>public</i>	+	Terlihat oleh semua elemen yang dapat mengakses konten <i>namespace</i> yang memilikinya
<i>private</i>	-	Hanya terlihat di dalam <i>namespace</i> yang memilikinya.
<i>protected</i>	#	Terlihat oleh elemen yang memiliki hubungan generalisasi dengan <i>namespace</i> yang memilikinya
<i>package</i>	~	Terlihat oleh elemen-elemen yang ada dalam paket yang sama dengan <i>namespace</i> yang dimilikinya.

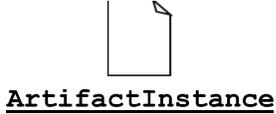
(Sumber : (“OMG Unified Modeling Language (OMG UML), Superstructure Version 2.4.1,” 2011))

3. *Deployment diagram*

Deployment diagram menunjukkan topologi perangkat keras yang digunakan dan sistem runtime. Perangkat keras memuat unit yang membentuk simpul serta hubungan komunikasi antar *node*. Sistem *runtime* berisi artefak atau *artifact* yang digunakan untuk node (Seidl et al., 2015).

Tabel 2.6 berikut menunjukkan *node* dan *edge* yang biasanya digunakan dalam *deployment diagram*:

Tabel 2.6 *Node dan edge deployment diagram*

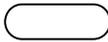
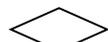
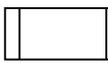
Tipe node/edge	Notasi	Deskripsi singkat
<i>Artifact</i>		Spesifikasi bagian informasi fisik yang digunakan atau diproduksi oleh proses pengembangan perangkat lunak, atau dengan <i>deployment</i> dan pengoperasian sistem.
<i>Node</i>		Node adalah sumber daya komputasi di mana artefak dapat digunakan untuk eksekusi.

(Sumber : (“OMG Unified Modeling Language (OMG UML), Superstructure Version 2.4.1,” 2011))

2.4.8. *Flowchart*

Flowchart adalah representasi grafis yang mewakili langkah-langkah yang spesifik, atau kegiatan dari suatu proses (Fryman, 2001). Tabel 2.7 menunjukkan simbol-simbol *flowchart* yang umum digunakan. *Flowchart* dapat digunakan sebagai komunikasi dan menunjukkan alur suatu algoritma atau proses serta mendokumentasikannya.

Tabel 2.7 Simbol-simbol *flowchart*

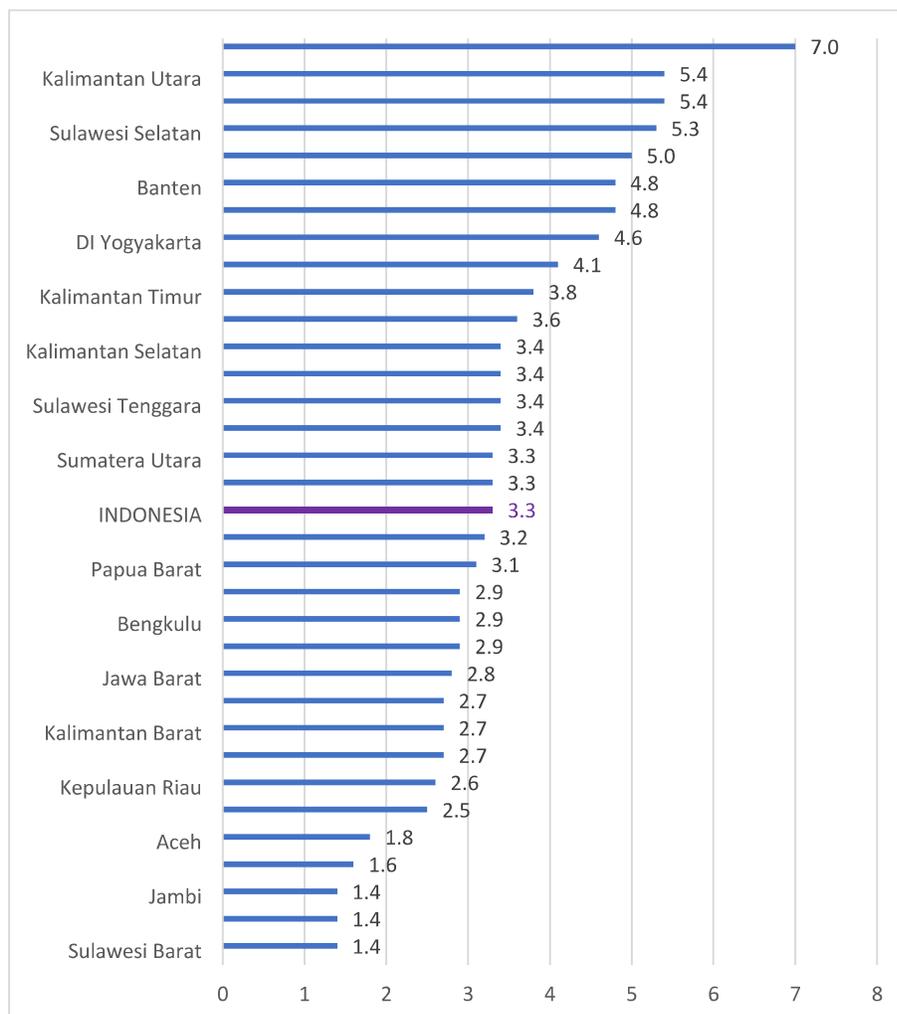
Simbol	Nama	Deskripsi
	<i>Flow</i>	Menunjukkan arah aliran proses.
	<i>Terminator</i>	Merepresentasikan awal dan akhir suatu proses.
	<i>Process</i>	Merepresentasikan langkah dalam proses
	<i>Decision</i>	Merepresentasikan percabangan atau pertanyaan dalam aliran proses
	<i>Predefined Process</i>	Merepresentasikan proses lain atau serangkaian langkah aliran proses yang didefinisikan di tempat lain.
	<i>Delay</i>	Mendeskripsikan masa tunggu yang merupakan bagian dari suatu proses.

2.5. Penelitian Terdahulu

Sebelum menetapkan judul dalam penelitian ini, terlebih dahulu dilakukan penelusuran terhadap penelitian-penelitian sebelumnya yang terkait dengan penelitian ini. Berikut beberapa penelitian terdahulu yang terkait dengan penelitian ini:

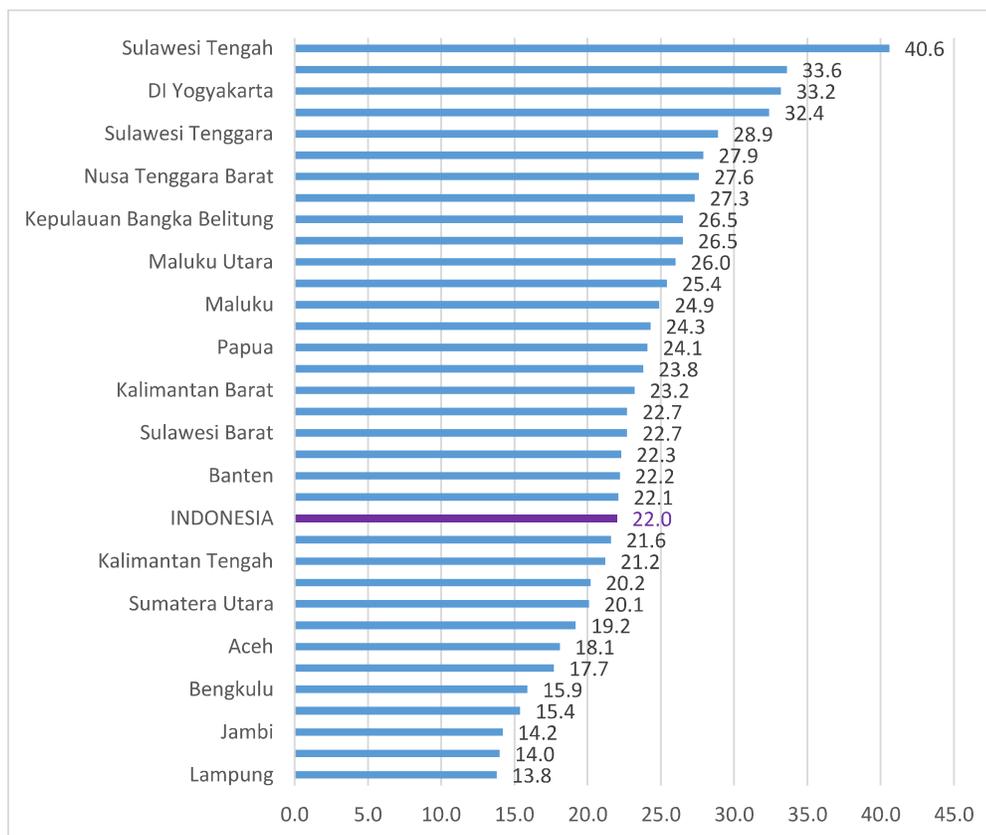
1. Disabilitas (“Disabilitas,” 2019).

Jurnal ini memberikan ringkasan data penyandang disabilitas yang ada di Indonesia berdasarkan Survei Penduduk Antar Sensus (SUPAS) tahun 2015 yang dikumpulkan oleh Badan Pusat Statistik (BPS) dan hasil Riset Kesehatan Dasar (Riskesdas) tahun 2018 yang telah dikumpulkan oleh Kementerian Kesehatan. Berikut ringkasan data Penyandang Disabilitas dalam Riskesdas.



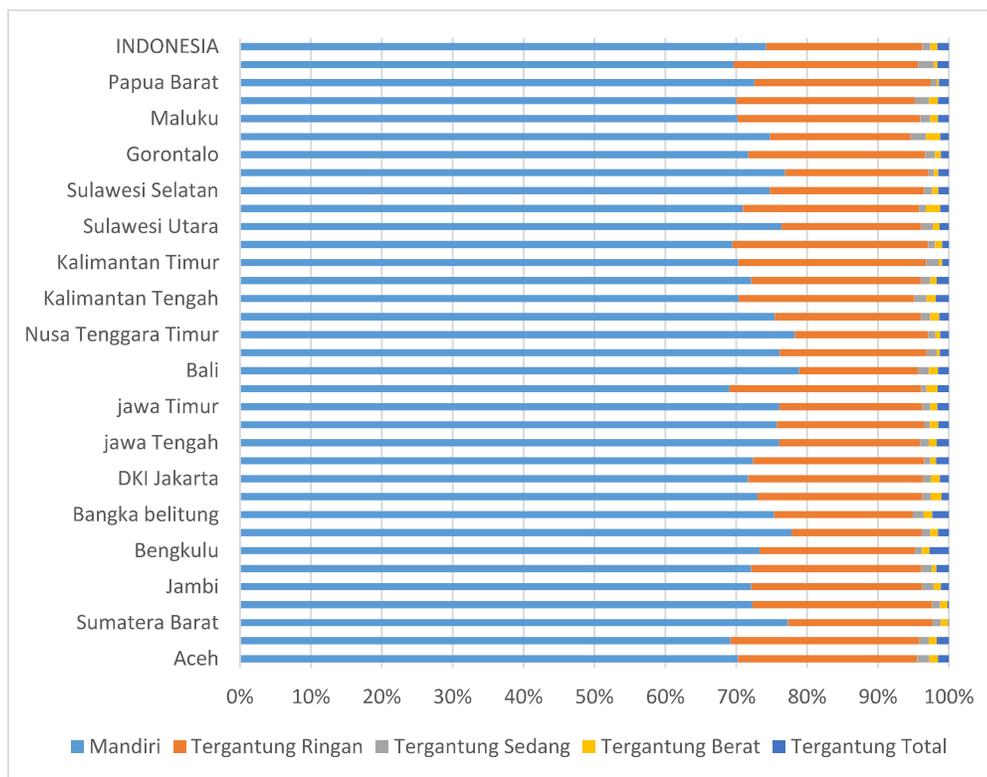
Gambar 2.12 Disabilitas pada Anak Umur 5-17 Tahun menurut Provinsi Tahun 2018
(Sumber :("Disabilitas," 2019))

Dari hasil tersebut anak umur 5-17 tahun yang mengalami disabilitas di Indonesia adalah sebesar 3,3%. Provinsi dengan proporsi disabilitas anak tertinggi adalah Sulawesi Tengah (7,0%), Kalimantan Utara, dan Gorontalo (masing-masing 5,4%), sedangkan proporsi terendah di Provinsi Sulawesi Barat, Lampung dan Jambi (masing-masing 1,4%).



Gambar 2.13 Proporsi Disabilitas Dewasa umur 18-59 Tahun menurut Provinsi di Indonesia Tahun 2018
(Sumber :("Disabilitas," 2019))

Proporsi disabilitas pada umur 18-59 tahun di Indonesia adalah sebesar 22,0%, proporsi terendah terdapat di Provinsi Lampung yaitu sebesar (13,8%), Kepulauan Riau (14,0%) dan Jambi (14,2%), proporsi tertinggi di Provinsi Sulawesi Tengah (40,6%), Sulawesi Selatan (33,6%), dan DI Yogyakarta (33,2%).



Gambar 2.14 Proporsi Disabilitas Lansia (umur ≥ 60 tahun) Menurut Provinsi di Indonesia Tahun 2018
(Sumber :("Disabilitas," 2019))

Didapatkan sebanyak 74,3% lansia (umur ≥ 60 tahun) dapat beraktifitas sehari-hari secara mandiri, 22,0% mengalami ketergantungan ringan, 1,1% mengalami ketergantungan sedang, 1% mengalami ketergantungan berat, dan 1,6% mengalami ketergantungan total. Provinsi dengan persentase lansia mandiri tertinggi adalah Bali (78,9%) dan terendah Banten (69,1%). Provinsi dengan proporsi lansia yang mengalami ketergantungan sedang, berat dan ketergantungan total tertinggi masing-masing adalah Sulawesi Barat (5,4%), Kepulauan Bangka Belitung (5,0%) dan Maluku Utara (4,9%), sedangkan provinsi dengan proporsi terendah adalah Papua Barat (2,6%) dan Kalimantan Utara, Sulawesi Tenggara, serta Nusa Tenggara Timur (masing-masing 2,9%).

2. *Application of Artificial Neural Networks in Modeling Direction Wheelchairs Using Neurosky Mindset Mobile (EEG) Device* (Siswoyo et al., 2017)

Penelitian ini dilakukan dengan latar belakang untuk membantu disabilitas dengan gangguan mobilitas. Penelitian ini menerapkan JST dengan algoritma *backpropagation* untuk memprediksi arah kursi roda listrik dengan masukan gelombang otak untuk disabilitas dengan gangguan mobilitas fisik. Perangkat yang digunakan dalam penelitian ini adalah Arduino Uno, *Neurosky Mindset sensor*, dan dengan perangkat lunak Matlab. Hasil percobaan didapatkan *Mean square Error* (MSE) pada akhir pelatihan sebesar 0,92495 pada epoch 9958, nilai koefisien korelasi sebesar 0,92804 menunjukkan bahwa akurasi hasil proses pelatihan sudah baik.

3. Identifikasi Kematangan Buah Tomat Berdasarkan Warna Menggunakan Metode Jaringan Saraf Tiruan (JST) (Kusumaningtyas & Asmara, 2016)

Penelitian ini dilatar belakangi oleh identifikasi kematangan buah dengan pengamatan manusia yang memiliki beberapa kelemahan diantaranya adalah waktu yang dibutuhkan relatif lama serta menghasilkan produk yang beragam karena adanya keterbatasan visual manusia, tingkat kelelahan dan perbedaan persepsi tentang mutu buah. Dalam penelitian ini JST dengan metode *perceptron* dan pengolahan citra digital diterapkan untuk mengidentifikasi kematangan buah tomat. Tampak visual buah didapatkan menggunakan webcam, yang kemudian kematangan buah tersebut diidentifikasi menggunakan JST. Tingkat keberhasilan identifikasi kematangan buah tomat dalam penelitian ini adalah sebesar 43,33%.

Dari hasil identifikasi yang diperoleh menghasilkan 3 output yaitu Mentah 10%, Setengah Matang 6,66%, dan Matang 26,66%.

4. Aplikasi Pengenalan Pola Daun Menggunakan Jaringan Saraf *Learning Vector Quantification* Untuk Penentuan Tanaman Obat (Indrawan, 2010)

Penelitian ini berangkat dari banyaknya jenis tanaman obat sehingga membuat sulit untuk mengidentifikasinya. Untuk menghindari pengonsumsi tanaman obat yang salah, penelitian ini kemudian dilakukan. Metode jaringan saraf yang digunakan dalam penelitian ini adalah LVQ. Pada penelitian ini JST digunakan untuk identifikasi pola daun tanaman obat, khususnya tanaman obat di daerah tropis. Pola yang akan diuji kemudian dibandingkan dengan pola hasil pelatihan. Hasil dari pengujian berupa informasi tanaman obat diantaranya nama tanaman, jenis, khasiat serta gambar daun dan pohon dari tanaman yang dicek.

5. Jaringan Saraf Tiruan *Learning Vector Quantization* untuk Aplikasi Pengenalan Tanda Tangan (Qur'ani & Rosmalinda, 2010)

Pada penelitian ini, digunakan metode deteksi tepi (*edge detection*) untuk mengolah citra yang digunakan sebagai masukan JST. Ini dilakukan untuk memperoleh garis tepi dari citra tanda tangan. Hasil pengujian menunjukkan bahwa JST dengan algoritma LVQ dapat mengenali pola tanda tangan dengan ketepatan 98% pada data yang di uji.

6. Identifikasi Telapak Tangan Menggunakan Jaringan Syaraf Tiruan *Learning Vector Quantization* (LVQ) (Afriandi & Sutikno, 2016).

Latar belakang penelitian ini adalah sistem pengenalan diri seperti kata sandi (*password*), ID card, atau PIN memiliki beberapa kelemahan yaitu dapat dicuri dan

mudah diduplikasi, memiliki kemungkinan seseorang untuk lupa dan beberapa password dapat diperkirakan sehingga dapat dimanfaatkan oleh orang-orang yang tidak bertanggung jawab. Penelitian ini menerapkan JST dengan metode LVQ untuk mengenali identitas seseorang melalui telapak tangan. Penelitian ini menggunakan model proses pengembangan perangkat lunak *Waterfall*. Perangkat lunak yang digunakan dalam penelitian ini diantaranya Matlab dan Microsoft Access sebagai basis datanya. Penelitian ini menggunakan JST dengan algoritma LVQ. Tingkat akurasi aplikasi hasil penelitian ini adalah sebesar 74,66%.

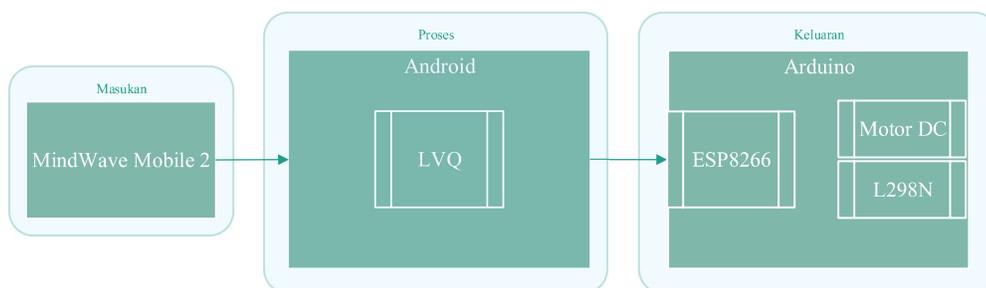
7. *Prediction of the Number of Airport Passengers Using Fuzzy C-Means and Adaptive Neuro Fuzzy Inference System* (Sitohang, Girsang, & Suharjito, 2017)

Penelitian yang dilakukan oleh Sitohang, Girsang, & Suharjito ini di latar belakang oleh masalah transportasi yang umum terjadi hampir di semua kota besar di dunia. Masalah transportasi tersebut diantaranya fasilitas transportasi yang terbatas, infrastruktur yang tidak memadai, tingkat disiplin yang rendah, urbanisasi yang cepat, dan perencanaan yang buruk. Sehubungan dengan masalah transportasi tersebut, diperlukan prediksi sebagai referensi untuk perencanaan. Prediksi jumlah penumpang merupakan hal yang penting untuk mempersiapkan fasilitas dalam mengantisipasi meningkatnya jumlah penumpang, seperti menyiapkan jadwal penerbangan tambahan, fasilitas ruang, tempat parkir yang lebih luas dan sebagainya. Dalam jurnal ini *Fuzzy C-Means* dan *Adaptif Neuro Fuzzy Inference System* digunakan untuk memprediksi jumlah penumpang bandara. Hasilnya menunjukkan bahwa “*Mean Absolute Percentage Errors (MAPE)* yang mewakili

kesalahan selama 4 tahun adalah 4,20%, 5,70%, 5,36% dan 4,47% dengan rata-rata 4,93%. Berdasarkan hasil ini, FCM dan ANFIS dapat dikombinasikan untuk memprediksi deret waktu data.

2.6. Kerangka Pemikiran

Berikut kerangka pemikiran yang penulis susun dalam penelitian ini.



Gambar 2.15 Kerangka pemikiran
(Sumber : Data Penelitian, 2020)

MindWave Mobile 2 akan bertindak sebagai masukan (sumber data) dengan mengirimkan data berupa gelombang otak ke Android melalui *bluetooth*. Data tersebut kemudian di proses di Android menggunakan algoritma LVQ untuk menghasilkan data keluaran berupa kelas yang menentukan arah kursi roda yang dapat menggerakkan motor. Setelah melakukan klasifikasi Arduino kemudian mengirimkan data hasil proses ke ESP8266 melalui Wifi, kemudian Arduino akan menerima dari ESP8266 dan menentukan arah pergerakan dari kursi roda dan menggerakkan motor DC melalui L298N.