

## **BAB 2**

### **DASAR TEORI**

#### **2.1 KAJIAN PUSTAKA**

Penelitian I Putu Sutawinaya, I Nyoman Gede Arya Astawa, dan Ni Kadek Dessy Hariyanti yang dilakukan pada tahun 2017 dengan judul “Perbandingan Metode Jaringan Saraf Tiruan Pada Peramalan Curah Hujan” melakukan perbandingan menggunakan dua metode yaitu metode *Backpropagation* dan metode *Adaline* untuk peramalan curah hujan. Penelitian dilakukan menggunakan data rata-rata curah hujan setiap bulan dari tahun 2006 sampai dengan tahun 2016 di kota Denpasar, Bali yang mana selanjutnya data tersebut di normalisasikan min-max. Pada penelitian ini menggunakan 5 neuron *input layer* dan 5 neuron layer tersembunyi dengan diperoleh hasil terbaik pada iterasi 1000. Pada iterasi 1000 yang dihasilkan *Root Mean Square Error* (RMSE) dengan metode *Backpropagation* sebesar 0,0435, sedangkan metode *Adaline* sebesar 0,0674. Berdasarkan perbandingan dari data nilai RMSE tersebut dimana metode *Backpropagation* lebih baik dibandingkan dengan metode *Adaline* [1].

Penelitian yang dilakukan oleh Ghufron Zaida Muflih, Sunardi, dan Anton Yudhana pada tahun 2019 dengan judul “Jaringan Saraf Tiruan *Backpropagation* Untuk Prediksi Curah Hujan Di Wilayah Kabupaten Wonosobo” melakukan penelitian prediksi curah hujan yang berada di wilayah Kabupaten Wonosobo dengan menggunakan jaringan saraf tiruan *Backpropagation* untuk mengukur tingkat curah hujan yang turun dalam waktu tertentu. Penelitian ini menggunakan data sekunder dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG) pada stasiun 24 Wanganaji pada tahun 2009 sampai dengan tahun 2011 dimana data tersebut sebagai masukan. Data dalam tiga tahun tersebut digunakan sebagai pelatihan dan pengujian. Data yang digunakan untuk pelatihan jaringan menggunakan data curah hujan di bulan ke 1 sampai dengan bulan ke 24 dan untuk pengujian jaringan menggunakan data curah hujan di bulan ke 13 sampai dengan bulan ke 36. Penelitian ini menggunakan arsitektur jaringan saraf tiruan 12-10-1 yang mana terdiri dari 12 nilai masukan data curah hujan selama 1 tahun atau 12 bulan, 10 neuron pada *hidden layer*, dan 1 nilai keluaran data curah hujan pada

bulan berikutnya. Data yang diperoleh MSE pada pelatihan sebesar 0,00099899 yang dicapai pada epoch yang ke 161 dimana koefisien korelasi R yang dihasilkan sebesar 0,99205. Untuk MSE pada pengujian jaringan diperoleh dengan nilai 0,17042. Pada proses pengujian menunjukkan dengan menggunakan jaringan saraf tiruan *Backpropagation* cukup baik untuk memprediksi curah hujan. Di penelitian ini menggunakan 10 *hidden layer* dan mempunyai akurasi yang cukup baik. Jumlah *hidden layer* yang berbeda jumlah iterasi juga berbeda dimana jumlah *hidden layer* lebih besar belum tentu menyebabkan iterasi meningkat. Untuk mempengaruhi performansi adalah dengan memperbanyak data latih dan mengubah parameter seperti error goal, jumlah epoch, arsitektur jaringan dan jenis fungsi aktivasi [4].

Pada penelitian Diera Desmonda, Tursina, dan M. Azhar Irwansyah pada tahun 2018 dengan judul “Prediksi Besaran Curah Hujan Menggunakan Metode *Fuzzy Time Series*” merancang prediksi besaran curah hujan untuk melihat dan menganalisa pola hujan untuk kedepannya dimana aplikasi yang akan dibangun menggunakan *forecasting* atau peramalan dengan menggunakan metode *Fuzzy Time Series*. Pada penelitian ini menggunakan logika Fuzzy karena dapat memetakan suatu input ke dalam suatu output dan memiliki toleransi terhadap data-data yang tersedia. Pengujian oleh program menggunakan data *time series* dengan 15 periode waktu yang berbeda yaitu 1 tahun, 2 tahun, 3 tahun sampai dengan 15 tahun dimana masing-masing data terdiri dari 12 bulan setiap periodenya, sehingga untuk 1 tahun terdapat 12 data, 2 tahun memiliki 24 data, 3 tahun memiliki 36 data hingga 15 tahun memiliki 180 data. Dari periode tersebut ditemukan nilai interval optimal di interval 401 yang mana menghasilkan nilai MAPE (*Mean Average Percentage Error*) terbaik. Nilai MAPE terbaik berada pada periode 2015-2017 dengan jumlah interval 401 sebesar 0,151%. Pada penelitian ini aplikasi yang dibangun dapat memproses olah data curah hujan harian menjadi curah hujan bulanan dan hasil perhitungan prediksi *Fuzzy Time Series* dipengaruhi dari jumlah data aktual yang digunakan pada acuan perhitungan pola data dan interval pada perhitungan pola data [3].

Penelitian I Putu Bagus Arya Pradnyana, Arief Andy Soebroto, Dan Rizal Setya Perdana pada tahun 2018 dengan judul “Peramalan Curah Hujan Menggunakan Metode Jaringan Saraf Tiruan Dengan Optimasi Algoritma *Bee*

*Colony*” melakukan suatu peramalan dengan menggunakan jaringan saraf tiruan (*Backpropagation*) dengan *Artificial Bee Colony* untuk mengetahui tingkat kesalahan pada peramalan curah hujan. Penelitian ini menggunakan data curah hujan dari BMKG Denpasar pada tahun 2006 sampai dengan 2016. Pada penelitian ini bertujuan untuk mengetahui perbandingan antara metode *Backpropagation* dengan metode *Backpropagation-Bee colony*. Untuk hasil pengujian algoritma dan akurasi pada metode *Backpropagation-Bee colony* lebih baik dibandingkan dengan metode *Backpropagation* dimana pada nilai rata-rata MSE memiliki hasil yang terbaik pada iterasi 1000 yang mana memperoleh hasil rata-rata MSE sebesar 0,0329. Untuk pengujian akurasi dengan iterasi 1000 memperoleh hasil rata-rata MSE sebesar 0,030 dibulan Desember dengan nilai akurasinya 95%. Namun metode *Bee colony* menggunakan proses yang cukup panjang dan dapat memakan waktu lebih lama daripada metode *Backpropagation* karena untuk menggunakan *Bee colony* harus melakukan proses *Backpropagation* untuk mendapatkan nilai MSE dan digunakan sebagai masukan pada proses *Bee colony* untuk dioptimasi dan dari nilai parameter *Backpropagation* yang sudah dioptimasi akan kembali lagi ke proses *Backpropagation* untuk mendapatkan nilai MSE kembali [5].

Sedangkan untuk penelitian Yoga Alif Kurnia Utama, Adang Hamdani, dan Arief Budijanto pada tahun 2017 dengan judul “Desain dan Pembuatan Sensor Curah Hujan Tanpa Penakar Berbasis Sensor *Load Cell*” melakukan pembuatan alat curah hujan tanpa penakar dengan sensor *load cell* dengan mengukur massa air hujan yang mengenai alat tersebut dan nilai massa dikonversi menjadi nilai curah hujan. Pengujian dilakukan untuk mengetahui hubungan antara massa air hujan dan tegangan keluaran *load cell*, dan pengujian untuk mengukur *error* dari pengukuran alat tersebut. Dari pengujian tersebut mendapatkan nilai rata-rata *error* sebesar 0,6657% dimana alat tersebut cukup baik untuk melakukan pengukuran curah hujan disuatu tempat [6].

## **2.2 DASAR TEORI**

### **2.2.1 Hujan**

Hujan merupakan proses fisis yang berasal dari hasil fenomena cuaca, sedangkan cuaca merupakan suatu sistem yang kompleks [7]. Hujan adalah bagian

dari siklus air yang berfungsi untuk menjaga keseimbangan air di alam. Hujan merupakan salah satu anugerah dari sang pencipta yang mana memiliki banyak manfaat dan sangat penting bagi kehidupan makhluk hidup di bumi. Namun, pada sisi lain hujan merupakan salah satu penyebab bencana alam jika jumlah hujan tidak terkendali dan bencana alam sangat sulit untuk dikendalikan oleh manusia. Sekarang ini usaha maksimal yang dapat dilakukan oleh manusia memahami pola hujan dengan mengamati formulasi hujan tersebut yang terdiri dari durasi hujan ( $t$ ) dan ketebalan hujan ( $R$ ). Dua variabel utama tersebut selalu diamati untuk analisa, prediksi dan perencanaan dimana dengan menggunakan variabel ini akan dapat menurunkan intensitas curah hujan [8].

Peristiwa jatuhnya air dengan bentuk cairan ataupun padat yang dicurahkan dari atmosfer ke permukaan bumi disebut dengan hujan. Sesuai dengan ukuran dari butiran, hujan dibedakan menjadi:

- a. Hujan gerimis (*drizzle*), mempunyai diameter butiran kurang dari 0,5mm.
- b. Hujan batu es, merupakan curahan batu es yang turun di cuaca panas awan dengan temperatur dibawah titik beku ( $0^{\circ}\text{C}$ ).
- c. Hujan salju (*snow*), merupakan kristal-kristal es dengan temperatur udara dibawah titik beku ( $0^{\circ}\text{C}$ ).
- d. Hujan deras (*rain*), merupakan curah hujan yang turun dari awan dan memiliki nilai temperatur diatas titik beku dengan diameter butiran sebesar  $\pm 7$  mm [9].

Probabilistik curah hujan selama 24 jam di wilayah Indonesia menggunakan data model prediksi numerik. Satu milimeter hujan dapat diartikan dengan air hujan yang turun di wilayah tertentu seluas satu meter persegi dan memiliki ketinggian satu milimeter pada saat air hujan tidak mengalir, meresap ataupun menguap. Untuk batas nilai yang digunakan intensitas hujan, diantaranya:

- a. 0mm/hari: berawan.
- b. 0,5mm-20mm/hari: hujan ringan.
- c. 20mm-50mm/hari: hujan sedang.
- d. 50mm-100mm/hari: hujan lebat
- e. 100mm-150mm/hari: hujan sangat lebat
- f. >150mm/hari: hujan ekstrem [10].

### 2.2.2 Curah Hujan

Curah hujan merupakan suatu ketinggian air hujan yang terkumpul pada tempat yang datar, tidak menguap, tidak meresap dan tidak mengalir. Curah hujan di Indonesia menggunakan satuan milimeter (mm) yang berarti pada luasan satu meter persegi di tempat datar yang tertampung air setinggi satu milimeter atau sebanyak satu liter. Negara Indonesia merupakan Negara yang termasuk dalam curah hujan yang cukup tinggi karena Indonesia merupakan wilayah tropis. Faktor-faktor yang memengaruhi curah hujan antara lain faktor garis lintang, faktor ketinggian tempat, jarak dari sumber air (penguapan), deretan pegunungan, perbedaan suhu daratan dan lautan, dan luas daratan [11]. Namun faktor utama adalah adanya intensitas curah hujan yang sangat tinggi yang menyebabkan kapasitas sungai tidak mampu menampung kapasitas air yang banyak yang akan menyebabkan limpahan air sungai berada pada daerah sekitarnya bisa dikatakan sebagai bencana banjir [5]. Curah hujan dapat dihitung dengan menggunakan rumus utama pada persamaan 2.1 [6].

$$C = \frac{V}{L} \quad (2.1)$$

Keterangan:

C : Curah hujan (mm)

V : Volume air hujan (mm<sup>3</sup>)

L : Luas penampang air hujan (mm<sup>2</sup>)

Untuk mencari luas penampang air hujan sesuai dengan bentuk penampang yang digunakan. Jika bentuk penampang air hujan berbentuk lingkaran maka luas yang dicari menggunakan persamaan 2.2. Dan untuk mencari volume air dapat menggunakan persamaan 2.3 [6].

$$L = \pi r^2 \quad (2.2)$$

$$V = \frac{m}{\rho} \quad (2.3)$$

Keterangan:

L : Luas penampang air hujan (mm<sup>2</sup>) dengan luas penampang bagian atas.

$\pi$  : 3.14

r : Jari-jari penampang air hujan (mm)

V : Volume air hujan (mm<sup>3</sup>)

m : massa air hujan (g)

$\rho$  : massa jenis air ( $\text{g/mm}^3$ ) ( $0.001 \text{ g/mm}^3$ ) [6].

### 2.2.3 Alat Penakar Hujan

Alat penakar hujan merupakan alat yang berfungsi untuk mengukur jumlah dari curah hujan yang turun ke permukaan tanah dengan per satuan luas. Dengan berdasarkan mekanisme, alat penakar hujan dibagi menjadi dua tipe yaitu penakar hujan tipe manual dan penakar hujan tipe otomatis (perekam) [12].

#### 2.2.3.1 Penakar Hujan Tipe Manual

Penakar hujan tipe manual hanya mempunyai satu tipe yaitu penakar hujan tipe ombrometer observatorium yang cara mengukur air hujan dengan menggunakan gelas ukur. Penakar hujan ini banyak digunakan di Indonesia [12].



Gambar 2.1 Penakar Hujan Tipe Ombrometer Observatorium (OBS)

Spesifikasi penakar hujan ombrometer observatorium (OBS):

1. Corong penakar berbentuk lingkaran dan dapat dilepas.
2. Luas corong  $100 \text{ cm}^2$ .
3. Menggunakan tabung penampung air hujan.
4. Kran untuk mengeluarkan air.
5. Terdapat penyangga.
6. Gelas ukur skala 0-25 mm [12].

Prinsip kerja dari penakar hujan ombrometer observatorium yaitu pada saat terjadi hujan, air hujan yang tercurah masuk dalam corong penakar. Air yang masuk dalam penakar kemudian dialirkan dan terkumpul di dalam tabung penampung. Pada jam-jam pengamatan air hujan yang tertampung diukur dengan menggunakan gelas ukur. Apabila jumlah curah hujan yang tertampung jumlahnya melebihi kapasitas ukur pada gelas ukur, maka pengukuran dilakukan beberapa kali hingga

air hujan yang tertampung dapat terukur semua [12]. Penakar ombrometer observatorium mempunyai kelebihan dan kekurangan, dimana kelebihannya adalah mudah untuk dipasang, dioperasikan, dan mudah untuk pemeliharannya. Sedangkan untuk kekurangannya adalah data curah hujan hanya didapat dengan periode selama 24 jam, mempunyai resiko kesalahan saat membaca permukaan dari tinggi air pada gelas ukur sehingga hasilnya dapat berbeda, mempunyai resiko kerusakan pada gelas ukur dan dalam pengambilan data masih manual tidak bisa menyimpan data dalam bentuk file [12].

### 2.2.3.2 Penakar Hujan Tipe Otomatis (Perekam)

Penakar hujan dengan tipe otomatis merupakan alat ukur dengan mekanisme pencatatan hujan secara otomatis atau perekam yang berfungsi untuk mengukur tinggi rendahnya curah hujan dan dapat mencatat lamanya waktu hujan dengan periode waktu tertentu. Penakar hujan otomatis memiliki kesamaan dengan penakar hujan manual dimana menggunakan tiga komponen yaitu corong, bejana pengumpul dan alat ukur. Namun perbedaannya berada di komponen bejana dan alat ukur dimana komponen tersebut dibuat secara khusus. Penakar hujan tipe otomatis biasanya digunakan oleh Badan Meteorologi Klimatologi dan Geofisika (BMKG) dan memiliki beberapa macam, yaitu:

#### a. Penakar Hujan Hellman

Penakar hujan Hellman merupakan peralatan meteorologi yang berfungsi untuk mengukur curah hujan. Untuk alat penakar hujan Hellman dapat dilihat pada gambar 2.2 [12].



Gambar 2.2 Penakar Hujan Hellman

Spesifikasi penakar hujan Hellman:

1. Bahan plat besi dan cat anti karat.
2. Luas corong 100 cm<sup>2</sup>.

3. Ketinggian alat dari tanah 120 cm.
4. Jam Hellman berputar selama 24 jam.
5. Pias diganti setiap jam 00.00 UTC (07.00 WIB).
6. Kapasitas pelampung sebesar 100 mm [12].

Prinsip kerja dari penakar hujan Hellman pada setiap hujan turun, air masuk ke corong dan disalurkan ke pelampung sehingga pena naik dan membuat grafik pada pias dimana ketinggian dari grafik menunjukkan jumlah curah hujan yang turun. Jika curah hujan bernilai 10 mm atau lebih, maka pena menunjukkan nilai 10 mm yang merupakan nilai maksimal dan air akan tumpah dari pelampung ke pipa hevel dan pena turun menjadi angka 0. Namun, jika masih terjadi hujan maka pena akan mencatat lagi, proses ini berlangsung terus menerus. Alat penakar jenis ini dapat mengetahui kapan terjadi hujan, durasi hujan, dan intensitas hujan dalam jangka waktu tertentu. Untuk kapasitas pada pengukuran tidak terbatas, pena yang digunakan jenis pena cartridge, dan jam Hellman menggunakan pegas sehingga diputar setiap jangka waktu tertentu [12].

b. Penakar Hujan *Tipping Bucket*

Penakar hujan *tipping bucket* adalah penakar hujan otomatis menggunakan prinsip menampung air hujan pada bejana yang berjungkit. Bila air mengisi bejana penampung yang setara dengan tinggi hujan 0,5 mm akan berjungkit dan air dikeluarkan. Terdapat dua buah bejana yang saling bergantian menampung air hujan. Tiap gerakan bejana berjungkit secara mekanis tercapat pada pias atau menggerakkan counter (penghitung). Jumlah hitungan dikalikan dengan 0,5 mm adalah tinggi hujan yang terjadi. Untuk penakar hujan *tipping bucket* dapat dilihat pada gambar 2.3 [13].



Gambar 2.3 Penakar Hujan *Tipping Bucket*

Penakar jenis ini mempunyai kekurangan dimana akurasi kurang baik dan eror nonlinieritasnya besar jika dalam keadaan hujan besar. Namun ada juga kelebihan yaitu operasi alatnya sangat simpel, dimana *bucket* atau wadah mempunyai dua bagian terpisah namun seimbang [14].

#### **2.2.4 Prediksi**

Prediksi merupakan proses untuk memperkirakan atau meramalkan sesuatu yang dapat terjadi di masa depan berdasarkan informasi relevan dan data pada masa lalu ataupun masa sekarang melalui metode ilmiah. Tujuan diadakannya prediksi dimana untuk memperkecil suatu kesalahan yang ada dan mengambil keputusan yang tepat dengan pertimbangan pada masa lampau ataupun pada saat pengambilan keputusan sehingga dapat mengurangi kesalahan dan tidak mengalami kesalahan yang sama [11].

Prediksi di dunia industri berfungsi untuk memperkirakan produksi, anggaran biaya, kebutuhan bahan baku, dan pemasaran. Pada prediksi atau peramalan ini menggunakan beberapa metode yang dapat digunakan diantaranya statistik dan *Artificial Intelligence* (AI). Metode statistik dibagi menjadi dua yaitu deskripsi statistik dan akurasi statistik. Sedangkan untuk metode *Artificial Intelligence* (AI) merupakan metode kecerdasan buatan atau sering disebut juga sebagai metode jaringan saraf tiruan (JST) dan ada beberapa metode lainnya seperti metode fuzzy, algoritma genetika, dan lain-lain [1].

#### **2.2.5 Jaringan Saraf Tiruan**

##### **2.2.5.1 Pengertian Jaringan Saraf Tiruan (JST)**

Jaringan saraf tiruan (JST) dapat disebut dengan *Artificial Neural Network* (ANN) merupakan pengolahan informasi yang terinspirasi oleh sistem saraf biologis manusia. Elemen kunci pada sistem ini dimana struktur dari sistem pengolahan informasi dalam jumlah besar pemrosesan yang saling terhubung (neuron) yang mana neuron dapat bekerja untuk menyelesaikan suatu masalah. Untuk prinsip kerja dari jaringan saraf tiruan (JST) seperti kerja otak manusia sendiri dimana pengenalan pola ataupun aplikasi data melalui proses dari pembelajaran. Pembelajaran tersebut berupa sistem biologi pada JST yang mana melibatkan penyusunan pada koneksi *dynamic* (yang terdapat antara neuron) [1].

Jaringan saraf tiruan (JST) mempunyai salah satu struktur yang ditiru dari otak manusia yaitu bentuk dari neuron atau sel saraf. Pada JST setiap neuron terdiri dari beberapa masukan dan satu keluaran, dimana jalur masukan berupa data mentahan atau data yang sudah diolah dari neuron sebelumnya. Sedangkan untuk hasil keluaran neuron merupakan hasil akhir atau bahan masukan bagi neuron berikutnya. Jaringan neuron terdiri dari lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan juga lapisan keluaran (*output layer*) [15].

#### **2.2.5.2 Kelebihan dan Kekurangan Jaringan Saraf Tiruan (JST)**

Kelebihan pada Jaringan Saraf Tiruan (JST) diantaranya:

1. *Self Organization*, sistem jaringan saraf tiruan (JST) dapat merepresentasikan lagi informasi yang telah diterima pada saat pembelajaran.
2. *Adaptive learning*, untuk menyelesaikan pekerjaan berdasarkan data sebelumnya.
3. *Real time operation*, membutuhkan waktu sedikit untuk komputasi atau perhitungan pada JST yang dilakukan dengan sistem paralel [11].
4. Dapat diimplementasikan tanpa ada masalah yang berarti dan dapat diimplementasikan pada berbagai aplikasi atau *software*.
5. Jaringan saraf tiruan (JST) dapat menjalankan tugas dimana program linier tidak dapat menjalankannya.
6. Pada saat jaringan gagal, JST tetap dapat menjalankan proses tanpa masalah pada paralel [15].

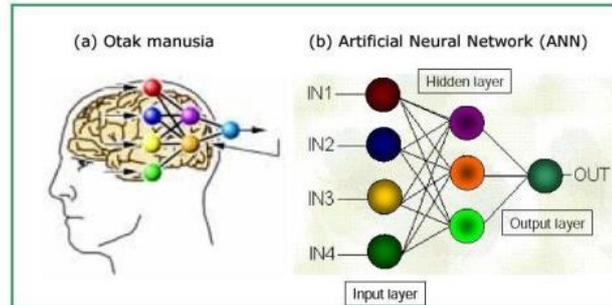
Kekurangan pada jaringan saraf tiruan (JST) diantaranya:

1. Membutuhkan adanya pelatihan sebelum menggunakan JST.
2. Membutuhkan adanya proses emulasi karena mempunyai perbedaan pada arsitektur di kebanyakan mikroprosesor.
3. Membutuhkan banyak waktu untuk menjalankan *software* pada saat menggunakan JST dengan ukuran besar [15].

#### **2.2.5.3 Arsitektur Jaringan Saraf Tiruan (JST)**

Pada dasarnya neuron mempunyai pola pembobot dan fungsi aktivasi dimana dua hal tersebut merupakan faktor terpenting untuk menentukan kelakuan pada neuron. Jika setiap lapisan sama, maka neuron-neuron mempunyai fungsi aktivasi yang sama. Namun, jika setiap lapisannya berbeda dimana neuron-neuron dalam

suatu lapisan atau lapisan tersembunyi dihubungkan ke neuron-neuron di lapisan lain atau lapisan *output*, maka setiap neuron pada lapisan tersembunyi harus terhubung ke lapisan *output* juga [15].

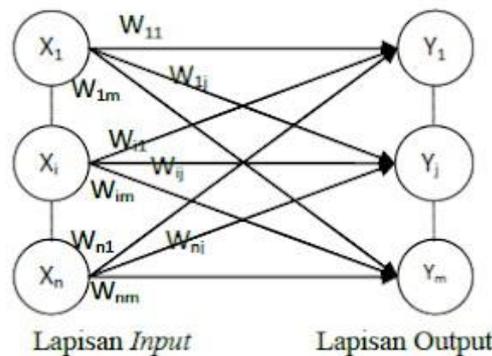


Gambar 2.4 Arsitektur Jaringan Saraf Tiruan

Terdapat beberapa bentuk arsitektur pada jaringan saraf tiruan, diantaranya:

a. Jaringan dengan lapisan tunggal (*single layer net*)

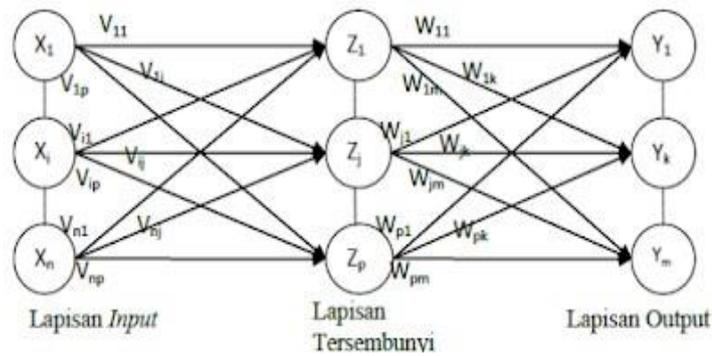
Pada bentuk lapisan tunggal mempunyai satu lapisan dengan bobot-bobot terhubung, dimana jaringan menerima *input* atau masukkan kemudian akan diolah secara langsung menjadi *output* atau keluaran tanpa melalui lapisan tersembunyi [15].



Gambar 2.5 Jaringan Lapisan Tunggal (*Single Layer Net*)

b. Jaringan dengan banyak lapisan (*multilayer*)

Pada bentuk *multilayer* atau banyak lapisan mempunyai satu atau lebih lapisan yang terdapat diantara lapisan *input* dan *output* yang mempunyai satu ataupun lebih lapisan tersembunyi. Jaringan bentuk ini mempunyai banyak lapisan yang mana dapat lebih sulit dalam menyelesaikan masalah dan pembelajarannya lebih kompleks. Namun, jaringan ini memiliki banyak kasus yang lebih berhasil untuk menyelesaikan suatu masalah [15].



Gambar 2.6 Jaringan Banyak Lapisan (Multilayer)

#### 2.2.5.4 Konsep Dasar Pemodelan Jaringan Saraf Tiruan (JST)

Tiruan dapat disebut sebagai buatan dari sebuah neuron. Nilai yang berasal dari masukan (*input*) dan keluaran (*output*) diproses di neuron kemudian neuron berkumpul di dalam suatu lapisan yang mana disebut sebagai *neuron layers*. Lapisan neuron sendiri mempunyai 3 macam lapisan, diantaranya:

1. *Input layer*, merupakan lapisan berupa unit-unit masukan (*input*) untuk menggambarkan permasalahan dari luar.
2. *Hidden layer*, merupakan lapisan yang berfungsi untuk memproses suatu unit masukan (*input*) sebelum ke lapisan keluaran (*output*).
3. *Output layer*, merupakan lapisan yang berfungsi untuk memberikan solusi ataupun hasil dari pemrosesan suatu permasalahan [11].

#### 2.2.5.5 Model dan Metode Pembelajaran Jaringan Saraf Tiruan (JST)

Perbedaan dari setiap jenis jaringan saraf tiruan (JST) dilihat berdasarkan algoritma, *activation function*, *type*, dan *learning method* yang digunakan. Model-model jaringan saraf tiruan (JST) yang memiliki kaitan dengan koneksi antar neuron yaitu:

1. *Feedforward*, merupakan tipe jaringan saraf tiruan (JST) yang mana neuron di suatu layer hanya dapat terkoneksi dengan neuron pada layer yang berbeda.
2. *Feedback*, merupakan tipe jaringan saraf tiruan (JST) yang mana neuron di suatu layer dapat terkoneksi dengan neuron layer lain dan neuron layer yang sama [15].

Untuk metode pembelajaran pada jaringan saraf tiruan (JST) memiliki tiga metode antara lain:

1. *Supervised Learning*, merupakan *error* atau selisih antara keluaran aktual dengan keluaran target untuk mengubah bobot yang bertujuan untuk mendapatkan nilai keluaran aktual yang tidak jauh beda dengan nilai keluaran

target. Untuk algoritma yang menggunakan metode ini yaitu *ADALINE*, *Backpropagation*, dan *Perceptron*.

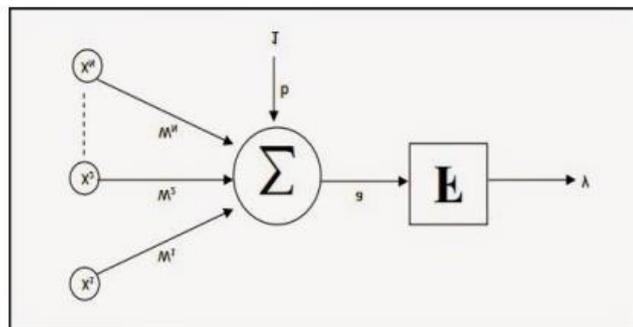
2. *Unsupervised Learning*, metode ini hanya dapat menyusun nilai bobot dengan range tertentu dimana tergantung pada masukkan yang diberikan, tidak dapat menentukan keluaran target. Metode ini bertujuan untuk mengelompokkan unit-unit yang hampir sama di area tertentu. Untuk algoritma yang menggunakan metode ini yaitu *Competitive*, *Neocognitron*, dan *LVQ*.
3. *Hybrid Learning*, metode ini merupakan suatu gabungan dari *Supervised Learning* dan *Unsupervised Learning* dimana untuk menentukan bobot sebagian dilakukan di *Supervised Learning* dan sebagian lagi di *Unsupervised Learning* [11].

#### 2.2.5.6 Fungsi Aktivasi Jaringan Saraf Tiruan (JST)

Fungsi aktivasi berfungsi untuk mengaktifkan dan menonaktifkan neuron. Untuk karakteristik JST ditentukan oleh input output, pembobot, dan fungsi aktivasi yang telah ditetapkan. Pada gambar 2.7 merupakan proses fungsi aktivasi pada JST, dimana keluaran ( $y$ ) didapatkan dari masukkan nilai  $X_1$ ,  $X_2$ , dan  $X_N$  yang memiliki pembobot masing-masing  $W_1$ ,  $W_2$ , dan  $W_N$ . Masing-masing impuls neuron dijumlahkan sebagai berikut:

$$\text{Net} = X_1W_1 + X_2W_2 + X_NW_N \quad (2.4)$$

Nilai impuls yang diterima oleh keluaran ( $y$ ) mengikuti dengan fungsi aktivasi dimana  $y = f(\text{net})$ . Jika nilai fungsi aktivasi cukup kuat, maka nilai sinyal diteruskan dan nilai keluaran akan digunakan untuk mengubah bobot [11].

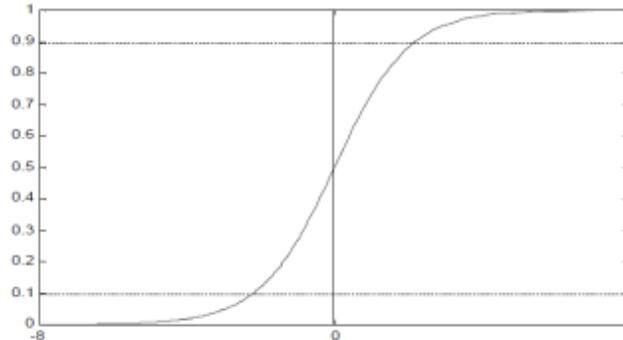


Gambar 2.7 Fungsi Aktivasi Jaringan Saraf Tiruan

Untuk jaringan saraf tiruan dengan metode *backpropagation* ada beberapa fungsi aktivasi yang digunakan, seperti fungsi sigmoid biner dan bipolar [16].

1. Fungsi aktivasi sigmoid biner, fungsi aktivasi ini menggunakan *range* dari 0 sampai 1 dirumuskan sebagai berikut [16]:

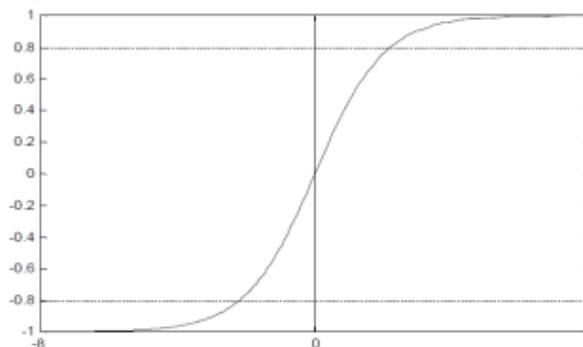
$$y = f(x) = \frac{1}{(1+e^{-x})} \quad (2.5)$$



Gambar 2.8 Fungsi Aktivasi Sigmoid Biner

2. Fungsi aktivasi sigmoid bipolar, fungsi aktivasi ini mempunyai bentuk fungsi yang mirip dengan fungsi sigmoid biner, namun pada *range* berbeda. Fungsi sigmoid bipolar menggunakan *range* dari -1 sampai 1, dirumuskan sebagai berikut [16]:

$$y = f(x) = \frac{1-e^{-x}}{(1+e^{-x})} \quad (2.6)$$



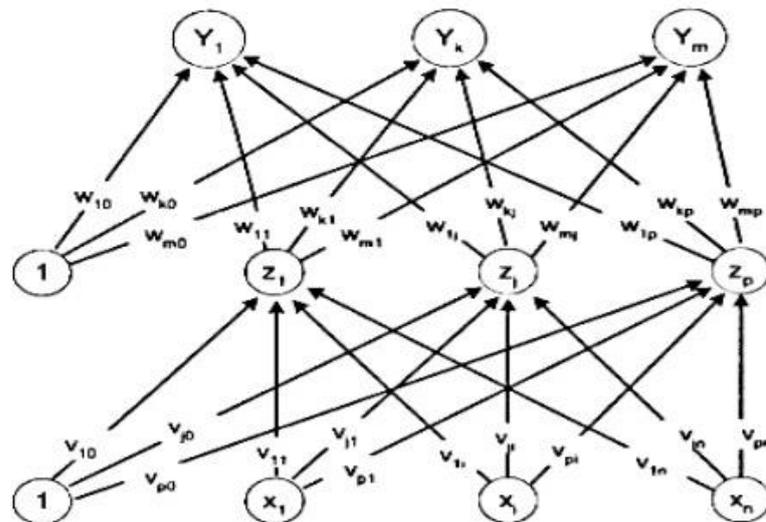
Gambar 2.9 Fungsi Aktivasi Sigmoid Bipolar

### 2.2.5.7 Jaringan Saraf Tiruan (*Backpropagation*)

Jaringan Saraf Tiruan (JST) dengan menggunakan metode *Backpropagation* yang mana merupakan algoritma pembelajaran yang berfungsi untuk mengurangi nilai *error* dengan menyesuaikan bobot berdasarkan perbedaan target dan *output*. *Backpropagation* menggunakan *multilayer network* dan arsitektur yang digunakan untuk proses pembelajaran yang terdiri dari tiga lapisan (*layer*) diantaranya *input layer*, *hidden layer*, dan *output layer* [5].

Metode *backpropagation* sudah banyak digunakan salah satunya diterapkan dalam bidang finansial, pengolahan citra medika, pengenalan pola suara sistem kendali, prediksi-prediksi, pengenalan pola tulisan tangan, dan lainnya. Untuk prinsip kerjanya jaringan mempunyai pola masukkan dimana pola tersebut akan diteruskan menuju unit-unit di lapisan tersembunyi kemudian diteruskan lagi ke lapisan keluaran. Dari lapisan keluaran diberikan tanggapan yang disebut sebagai keluaran jaringan. Jika pada keluaran jaringan memiliki hasil tidak sama dengan keluaran yang diinginkan, maka keluaran disebar mundur di lapisan tersembunyi kemudian ke unit-unit lapisan masukkan. Meskipun pada metode ini menggunakan lebih dari satu lapisan tersembunyi dan memiliki banyak manfaat di beberapa kasus tetapi memang pelatihannya memerlukan waktu yang cukup lama [15].

Arsitektur *backpropagation* pada gambar 2.8 terdiri dari jumlah  $n$  masukkan yang ditambah bias, sebuah lapisan tersembunyi berupa  $p$  unit yang ditambah bias, dan  $m$  merupakan unit keluaran. Untuk  $V_{ji}$  adalah bobot garis dari unit masukkan  $X_i$  ke unit lapisan tersembunyi ( $V_{j0}$  adalah bobot garis yang mana menghubungkan bias pada unit masukkan ke unit lapisan tersembunyi  $Z_j$ ). Untuk  $W_{kj}$  adalah bobot dari unit lapisan tersembunyi  $Z_j$  ke unit keluaran  $Y_k$  ( $W_{k0}$  adalah bobot dari bias di lapisan tersembunyi ke unit keluaran  $Y_k$ ) [15].



Gambar 2.10 Arsitektur *Backpropagation*

Berikut merupakan langkah-langkah algoritma *Backpropagation* adalah:

Langkah 1: Inisiasi bobot menggunakan nilai acak kecil.

Langkah 2: Selama syarat kondisi berhenti salah, kerjakan langkah 3-8.

Umpan maju (*feedforward*)

Langkah 3: Setiap unit masukan ( $x_i, i=1, \dots, n$ ) menerima sinyal masukan  $x_i$  dan diteruskan ke unit-unit tersembunyi (*hidden layer*).

Langkah 4: Setiap unit tersembunyi ( $z_j, z = 1, \dots, p$ ) menjumlahkan bobot sinyal input dengan persamaan berikut:

$$z\_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (2.7)$$

Dengan menerapkan fungsi aktivasi hitung:

$$z_j = (z\_in_j) \quad (2.8)$$

Jika menggunakan fungsi aktivasi sigmoid, maka rumus persamaan 2.5 dan mengirimkan isyarat ke semua unit pada unit keluaran.

Keterangan:

$z\_in_j$  = sinyal *input* pada *hidden layer* ke-j

$v_{oj}$  = bias ke *hidden layer* ke-j

$v_{ij}$  = bobot antara unit *input* layer ke-I dan *hidden layer* ke-j

$x_i$  = unit *input layer* ke-i

$z_j$  = unit *input layer* ke-j

$i$  = urutan unit *input layer*

$j$  = urutan unit *hidden layer*

$p$  = jumlah maksimum unit pada *hidden layer*

Langkah 5: Setiap unit keluaran ( $y_k, k=1, 2, \dots, m$ ) menjumlahkan isyarat masukan terbobotnya.

$$y\_in_k = w_{ok} + \sum_{j=1}^p z_j v_{jk} \quad (2.9)$$

Dengan menerapkan fungsi aktivasi hitung:

$$y_j = f(y\_in_k) \quad (2.10)$$

Keterangan:

$y\_in_k$  = sinyal masukan *output* ke-k

$w_{ok}$  = bias ke *hidden layer* ke-k

$v_{jk}$  = *output* ke-j dan *hidden layer* ke-k

$y_j$  = aktivasi *hidden layer* ke-j

Perambatan galat mundur (*backpropagation*)

Langkah 6: Setiap unit keluaran ( $y_k$ ,  $k=1,2,\dots,m$ ) menerima pola pelatihan masukannya. Hitung galat (*error*) dengan persamaan:

$$\delta_k = (tk - y_k) f' (y_{-ink}) \quad (2.11)$$

Hitung koreksi bobot dan biasnya:

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.12)$$

$$\Delta w_{ok} = \alpha \delta_k \quad (2.13)$$

Keterangan:

$\delta_k$  = faktor koreksi *error* bobot  $w_{jk}$

$tk$  = target *output* ke- $k$

$y_k$  = aktivasi output ke- $k$

$\Delta w_{jk}$  = nilai koreksi error bobot  $w_{jk}$

$\Delta w_{ok}$  = nilai koreksi error bias  $w_{ok}$

$z_j$  = aktivasi hidden layer ke- $j$

Langkah 7: Setiap unit tersembunyi ( $z_j$ ,  $j=1,\dots,p$ ) menjumlahkan delta masukannya (dari unit-unit yang ada pada lapisan atasnya).

$$\delta_{-inj} = \sum_{k=1}^m \delta_k w_{kj} \quad (2.14)$$

Daan hitung galat *error* dengan persamaan:

$$\delta_j = \delta_{-nj} f' (z_{-inj}) \quad (2.15)$$

Hitung koreksi bobot dan biasnya:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.16)$$

$$\Delta v_{oj} = \alpha \delta_j \quad (2.17)$$

Keterangan:

$\delta_{-inj}$  = jumlah delta bobot hidden layer ke- $j$

$\delta_k$  = faktor koreksi error bobot  $w_{kj}$

$w_{kj}$  = bobot antara output ke- $k$  dan hidden layer ke- $j$

$\delta_j$  = faktor koreksi bobot  $v_{ij}$

$z_{-inj}$  = aktivasi hidden layer ke- $j$

$\Delta v_{ij}$  = pembaruan  $v_{ij}$

$\Delta v_{oj}$  = pembaruan  $v_{oj}$

$\alpha$  = laju percepatan (*learning rate*)

$x_i$  = unit input ke- $i$

Perbaiki bobot dan bias

Langkah 8: Setiap unit keluaran ( $y_k, k=1, \dots, m$ ) memperbaharui bobot dan bias ( $j = 0, 1, \dots, p$ ).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.18)$$

Setiap unit tersembunyi ( $z_j, j=1, \dots, p$ ) memperbaharui bobot dan bias ( $i = 0, 1, \dots, n$ ).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.19)$$

Keterangan:

$w_{jk}(\text{baru})$  = bobot baru dari unit *hidden layer* menuju unit *output layer*

$w_{jk}(\text{lama})$  = bobot lama dari unit *hidden layer* menuju unit *output layer*

$v_{ij}(\text{baru})$  = bobot baru dari unit *hidden layer* menuju unit *output layer*

$v_{ij}(\text{lama})$  = bobot lama dari unit *hidden layer* menuju unit *output layer*

Langkah 9: Uji syarat berhenti [17].

Metode *backpropagation* menggunakan proses normalisasi dimana proses normalisasi merupakan proses penskalaan nilai yang diperoleh dari data dengan range tertentu. Normalisasi menggunakan salah satu metodenya yaitu min-max dimana mengubah data ke *range* baru lain antara 0 sampai 1 [5]. Untuk rumus normalisasi sebagai berikut:

$$x' = \frac{(x-x_{\min})}{x_{\max}-x_{\min}} \quad (2.20)$$

Keterangan:

$x'$  : hasil normalisasi

$x$  : data asli

$x_{\max}$  : nilai maksimum dari semua data asli

$x_{\min}$  : nilai minimum dari semua data asli [18].

Jika menggunakan fungsi aktivasi sigmoid biner dengan interval 0 dan 1 yang mana nilai tidak tepat 0 atau 1, maka data dinormalisasi dengan interval 0,1 sampai 0,9, rumusnya sebagai berikut [19]:

$$x' = \frac{(0,9-0,1)(x-x_{\min})}{x_{\max}-x_{\min}} + 0,1 \quad (2.21)$$

$$x' = \frac{0,8(x-x_{\min})}{x_{\max}-x_{\min}} + 0,1 \quad (2.22)$$

Untuk *Mean Square Error* (MSE) merupakan rata-rata kesalahan atau *error* dari meramal yang dikuadratkan. Jika terdapat sebanyak  $m$  pada training data, maka menghitung mean square error menggunakan persamaan berikut:

$$MSE = \sum \frac{(t-y_k)^2}{N} \quad (2.23)$$

Keterangan:

$t$  : nilai *output* atau keluaran pada target

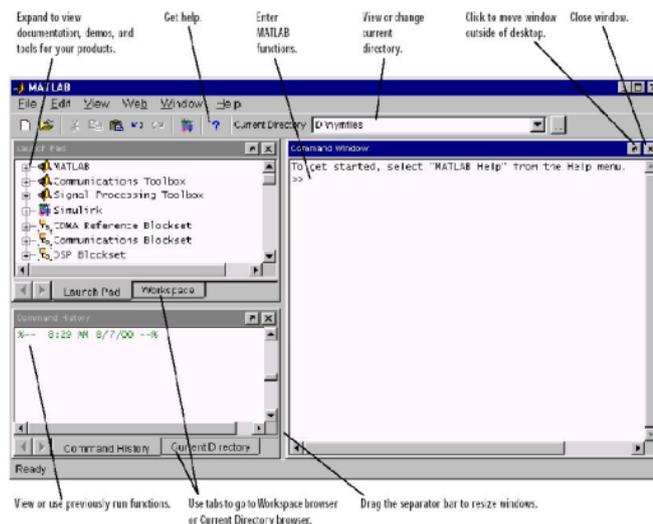
$y_k$  : nilai *output* atau keluaran pada jaringan

$N$  : banyaknya data [18].

## 2.2.6 MATLAB

Matlab sendiri singkatan dari *matrix laboratory* yang mana merupakan perangkat lunak yang digunakan untuk komputasi teknis dan saintifik. Matlab juga disebut sebagai integrasi, komputasi, visualisasi, dan pemrograman yang mudah untuk digunakan. Matlab dapat berfungsi sebagai kalkulator dan bahasa pemrograman. Untuk operasi yang dapat digunakan matlab yaitu sebagai saklar yang mana berupa bilangan kompleks atau *real*, matriks dan vektor dengan elemen bilangan kompleks atau *real*, dan sebagai teks untuk pengolahan kata [20].

Pada saat matlab dijalankan muncul *tools (graphical user interfaces)* yang berfungsi untuk mengatur variabel, *file* dan aplikasi matlab. Gambar 2.11 menunjukkan tampilan desktop pada matlab.



Gambar 2.11 Desktop Matlab

Jendela-jendela yang terdapat pada matlab sebagai berikut:

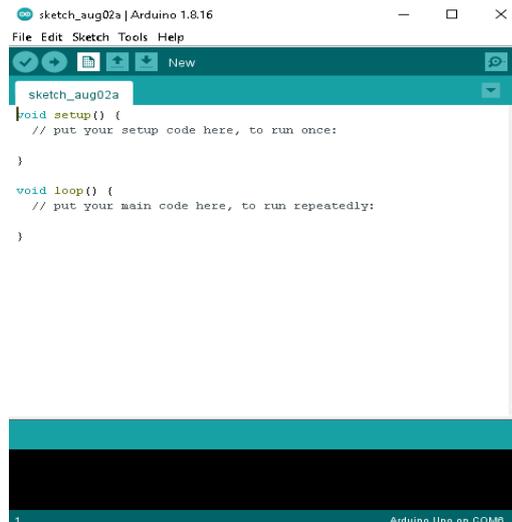
- Command window*, untuk menjalankan fungsi-fungsi pada matlab.
- Command history*, untuk menyimpan baris-baris perintah yang terdapat pada *command window* dan dapat melihat fungsi-fungsi yang telah digunakan sebelumnya, menyalin dan menjalankan kembali dari *command window*.

- c. *Launch pad*, digunakan untuk menyediakan akses yang mudah ke demo, dokumentasi, dan *tools*.
- d. *Help browser*, digunakan untuk menampilkan dan mencari dokumentasi pada semua produk *Mathwork*.
- e. *Current directory browser*, operasi pada file matlab menggunakan *current directory* dan *search path* sebagai referensi. File yang akan dijalankan harus ada pada *current directory* ataupun *search path*.
- f. *Workspace browser*, merupakan kumpulan variabel yang terbentuk pada sepanjang sesi matlab dan akan disimpan di *memory*.
- g. *Array editor*, digunakan untuk menampilkan isi pada *array* dengan format tabel dan digunakan juga untuk mengedit isi *array*.
- h. *Editor/debugger*, untuk membuat, mengedit, dan men-*debug* M-file yang mana merupakan program untuk menjalankan fungsi-fungsi pada matlab.
- i. *Figure*, merupakan jendela yang berfungsi untuk menampilkan gambar grafik seperti perintah plot dan lainnya. Gambar dapat disimpan di sebuah file yang mana dapat dipanggil kembali sewaktu-waktu.
- j. *Help*, digunakan untuk menghafalkan semua perintah pada matlab yang mana mungkin mendapati banyak kesulitan. Untuk itu, matlab memberikan menu *help online* untuk bantuan dan untuk memperoleh informasi cepat pada *command window* atau *help browser* [20].

Matlab merupakan software yang memiliki *control toolbox*, dimana *toolbox* digunakan untuk berbagai macam fungsi untuk menganalisis sistem kontrol. Fungsi yang digunakan untuk menganalisis sistem kontrol tersebut antara lain *series*, *step*, *rlocus*, *feedback*, dan lainnya [21].

### 2.2.7 Arduino IDE

*Software* arduino IDE digunakan untuk mengembangkan dan memprogram ke dalam board arduino. Arduino IDE merupakan *tool* yang memiliki manfaat untuk menulis program yang mana pada arduino disebut sketsa atau *sketch*, mengompilasi dan mengunggahnya ke papan arduino. Gambar 2.12 menunjukkan tampilan *sketch* atau sketsa arduino [22].



Gambar 2.12 Tampilan Sketsa atau *Sketch* Arduino

Ada beberapa tombol menu yang terdapat paada software arduino IDE diantaranya *File*, *Sketch*, *Tools* dan *Help*. Dan pada tombol *tool bar* terdiri dari:

- a. *Verify*, untuk mengkompilasi program atau mengknversi program pada arduino menjadi informasi/data yang dapat dibaca oleh mikrokontroler.
- b. *Upload*, untuk mengunggah program ke dalam board arduino.
- c. *New*, untuk membuat *file sketch* baru.
- d. *Open*, untuk membuka *file sketch* yang sudah disimpan/dibuat.
- e. *Save*, untuk menyimpan *sketch* program yang sedang dibuat.
- f. *Serial Monitor*, untuk mengaktifkan jendela komunikasi serial dan transfer data (kirim/terima) antara board arduino dan komputer [22].

### 2.2.8 Mikrokontroler

Mikrokontroler merupakan sistem komputer yang berada pada *chip* yang mana didalamnya terdapat memori (RAM, program atau dua-duanya), masukan dan keluaran, dan prosesor. Mikrokontroler dapat disebut sebagai alat elektronika digital yang memiliki *input* dan *output* serta sistem kendali dengan memiliki program yang dapat dihapus dan ditulis secara khusus. Mikrokontroler juga memiliki fungsi kerja dengan menulis dan membaca data. Pada tugas akhir ini menggunakan salah satu jenis mikrokontroler yaitu Arduino ATmega 328 [23]. Mikrokontroler ATmega328 merupakan Arduino UNO yang menggunakan IC (*Integrated Circuit*) ATmega 328 yang terdiri dari 14 input/output digital atau 6 output untuk PWM, memiliki 6 analog *input*, koneksi USB, pin *header* ICPS, tombol riset, dan resonator kristal keramik sebesar 16MHz. Hal tersebut dibutuhkan

untuk menjadikan mikrokontroler mudah terhubung ke kabel *power* USB atau kabel *power supply* adaptor AC ke DC ataupun baterai [24].

Spesifikasi dari mikrokontroler ATmega328 sebagai *hardware* sistem pengendali sebagai berikut:

- a. *Operating Voltage*: 5V
- b. *Input Voltage*: 7-12V
- c. *Input Voltage (Limits)*: 6-20V
- d. *Digital I/O Pins*: 14(Of Which 6 Provide PWM Output)
- e. *Analog Input Pins*: 6
- f. *DC Current Per I/O Pin*: 40Ma
- g. *DC Current For 3.3V Pin*: 50ma
- h. *Flash Memory*: 32KB
- i. *SRAM*: 2KB
- j. *EEPROM*: 1KB
- k. *Clock Speed*: 16mhz
- l. *Length*: 68.6mm
- m. *Width*: 53.4mm
- n. *Weight*: 25g [24].

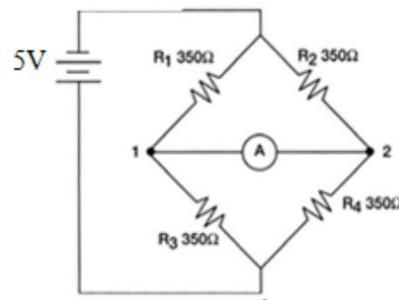
Mikrokontroler merupakan arduino yang dapat melakukan suatu proses masukan dan keluaran data dengan cara analog dan digital. Mikrokontroler ini memiliki kelebihan diantaranya mudah diimplementasikan, memiliki banyak modul yang dapat diintegrasikan dengan modul utama, dan untuk harganya terbilang murah. Mikrokontroler ATmega328 digunakan untuk meminimalisir penggunaan daya, biaya, dan peralatan [25].



Gambar 2.13 Mikrokontroler (Atmega328)

### 2.2.9 Sensor Load Cell

Pada penelitian ini sensor curah hujan memanfaatkan sensor *load cell* sebagai komponen sensornya untuk mengukur curah hujan atau mengukur air hujan yang mengenai sensor *load cell*. Sensor *load cell* merupakan sensor yang berfungsi untuk mendeteksi berat maupun tekanan suatu benda. Pada umumnya sensor *load cell* sebagai komponen utama pada timbangan digital dan diaplikasikan pada jembatan timbangan untuk menimbang truk pengangkut bahan baku. Pengukuran pada sensor *load cell* menggunakan prinsip tekanan [26]. Sensor *load cell* terdiri dari *strain gauge*, jembatan *wheatstone*, dan konduktor. Untuk prinsip kerja pada sensor *load cell* berdasarkan rangkaian jembatan *wheatstone* seperti pada gambar 2.14, dimana proses penimbangan dengan elemen logam pada *load cell* menyebabkan reaksi dan menimbulkan gaya elastis. Pada regangan menimbulkan gaya yang mana akan diubah oleh *strain gauge* yang terdapat pada *load cell* ke dalam sinyal elektrik [27].



Gambar 2.14 Rangkaian Jembatan *Wheatstone*

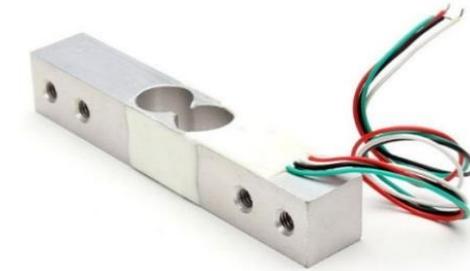
$$V_o = \left( V_s \times \left( \frac{R_1}{R_1 + R_4} \right) \right) - \left( V_s \times \left( \frac{R_2}{R_2 + R_3} \right) \right) \quad (2.11)$$

Pada teori yang ada, jika rangkaian jembatan *wheatstone* memiliki beban, maka pada nilai resistansi mengalami perubahan. Pada saat dalam keadaan posisi seimbang,  $V_{out}$  *load cell* memiliki tegangan sebesar 0 volt. Namun jika nilai pada resistansi  $R_1$  dan  $R_3$  naik atau turun maka akan mengalami perubahan pada  $V_{out}$  *load cell* [27].

Spesifikasi dari sensor *load cell* sebagai berikut:

1. Menggunakan material *Aluminium Alloy*.
2. Ukuran: 60 x 12,8 x 12,8 mm dan berat 23 gram.
3. Memiliki kapasitas sebesar 5 kg atau 5000 gram.
4. Rentang tegangan keluaran sebesar 0,1 mV – 1 mV/V (skala 1:1000 terhadap tegangan masukan,  $error\ margin \leq 1,5\%$ ).

5. Impedansi masukan (*input impedance*):  $1066 \Omega \pm 20\%$ .
6. Impedansi keluaran (*output impedance*):  $1000 \Omega \pm 10\%$ .
7. Tegangan maksimum 10 VDC.
8. Temperatur kerja dari  $-20^{\circ}\text{C}$  sampai dengan  $65^{\circ}\text{C}$  [28].



Gambar 2.15 Sensor *Load Cell*

Keterangan:

- a. Kabel merah merupakan masukan (*input*) dari tegangan sensor.
- b. Kabel hitam merupakan masukan (*input*) dari *ground* sensor.
- c. Kabel hijau merupakan keluaran (*output*) dari positif sensor.
- d. Kabel putih merupakan keluaran (*output*) dari *ground* sensor [26].

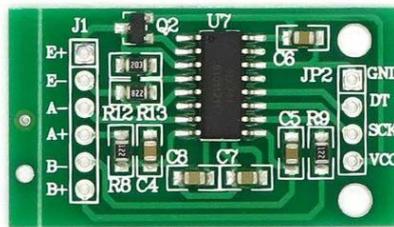
Prinsip kerja sensor *load cell*, pada saat mendapat tekanan beban dimana bagian lain yang lebih elastik mendapat tekanan, maka sisi lain mengalami perubahan regangan seperti yang dihasilkan oleh *strain gauge*, karena terdapat gaya yang seperti melawan pada sisi lainnya. Perubahan nilai resistansi yang diakibatkan dari perubahan gaya diubah menjadi nilai tegangan oleh IC HX711. Berat yang diukur dari objek dapat diketahui dengan mengukur besar nilai tegangan yang timbul [28].

#### **2.2.10 Modul HX711**

Modul HX711 merupakan modul timbangan yang digunakan untuk sensor timbangan dan industrial kontrol aplikasi yang dikoneksi dengan sensor jembatan [26]. Modul ini berfungsi untuk menguatkan sinyal keluaran dari sensor yang kemudian dikonversi dari data analog menjadi data digital dan hasil data pengukuran diolah pada mikrokontroler dan selanjutnya dikonversi kembali menjadi satuan berat [27]. Modul HX711 dapat digunakan di bidang elektrik, konstruksi, mekanik, aerospace, farmasi, kimia dan lain-lain dimana berfungsi untuk mengukur gaya, perpindahan, torsi, gaya tekanan, gaya tarikan, dan percepatan [26].

Spesifikasi dari modul HX711 sebagai berikut:

1. *Differential input voltage* sebesar  $\pm 40$  mV (*full-scale differential input voltage*  $\pm 40$  mV).
2. Akurasi data 24 bit (24 bit A/D converter chip).
3. *Refresh Frequency* 80Hz.
4. *Operating Voltage* 5 VDC.
5. *Operating Current* <10 mA.
6. Memiliki ukuran sebesar 38mm x 21mm x 10mm [26].



Gambar 2.16 Modul HX711

### 2.2.11 LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) merupakan media tampil dengan menggunakan Kristal cair untuk penampil utamanya. Fitur-fitur yang terdapat pada LCD antara lain:

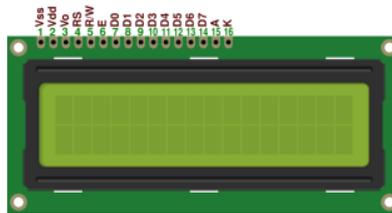
- a. Memiliki 16 karakter dan 2 baris.
- b. Memiliki 192 karakter yang tersimpan.
- c. Dapat dialamati menggunakan mode 4-bit dan 8-bit.
- d. Dilengkapi dengan *backlight*.
- e. Mempunyai karakter generator yang sudah diprogram.

Untuk proses inisialisasi pada pin arduino yang terhubung di pin LCD RS, *Enable*, D4, D5, D6, dan D7 yang dilakukan di baris LiquidCrystal (2,3,4,5,6,7) yang mana LCD adalah variabel yang akan dipanggil pada saat instruksi LCD akan digunakan. Untuk pin LCD 16x2 dapat dilihat pada tabel 2.1 dan gambar 2.17 [29].

Tabel 2.1 Spesifikasi *Liquid Crystal Display* (LCD) 16X2

Pin	Deskripsi
1	<i>Ground</i>
2	Vcc
3	Pengatur Kontras

Pin	Deskripsi
4	Register <i>Select</i>
5	<i>Read/Write</i> LCD Register
6	<i>Enable</i>
7-14	Data I/O Pins
15	Vcc + LED
16	<i>Ground</i> - LED



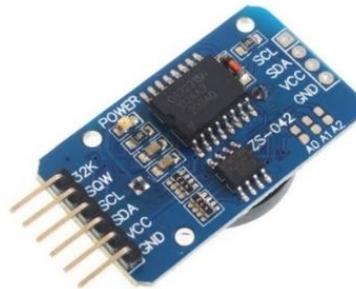
Gambar 2.17 LCD (*Liquid Crystal Display*) 16x2

### 2.2.12 RTC (*Real Time Clock*)

Modul DS3231 merupakan salah satu jenis modul untuk pewaktu digital atau *Real Time Clock* (RTC). Cara mengakses modul DS3231 dengan menggunakan i2c atau *two wire* terdiri dari SDA dan SCL. Untuk mengakses modul DS3231 menggunakan mikrokontroler yang mana seperti Arduino Uno membutuhkan 2 pin SDA dan SCL dan juga 2 pin power. Modul ini sangat akurat untuk mengelola semua fitur pada pengatur waktu dan dapat menghitung waktu dalam detik, menit, jam, hari dalam seminggu, tanggal, bulan, dan tahun namun dengan batas hingga tahun 2100. Untuk spesifikasi dan fitur dari *Real Time Clock* (RTC) diantaranya: [30].

- a. Menggunakan tegangan sebesar 3.3-5.5 Volt.
- b. *Clock chip* DS3231.
- c. *Memory chip* AT24C32 dengan kapasitas 32K.
- d. *Error* penghitungan waktu  $\pm 1$  menit.
- e. Akurasi  $\pm 2$  ppm dari  $0^{\circ}\text{C}$  sampai  $+40^{\circ}\text{C}$
- f. Akurasi 3.5ppm dari  $-40^{\circ}\text{C}$  sampai  $+85^{\circ}\text{C}$
- g. Mendaftar untuk *aging trim*.
- h. Digital temp sensor *output*  $3^{\circ}\text{C}$  akurasi.
- i. *Two time-of-day alarms*.

- j. *Active-low RST output/push button reset debounce input.*
- k. *Output-programmable square-wave output.*
- l. Kecepatan data transfer *i2c interface* sebesar 400 kHz.
- m. Antarmuka serial sederhana kebanyakan dapat menghubungkan ke mikrokontroler.
- n. Membutuhkan masukkan cadangan baterai untuk pencatatan waktu terus menerus. Baterai isi ulang CR2032 untuk memastikan sistem setelah power.
- o. Kecepatan jam 0-40.
- p. Dapat mengalir menggunakan perangkat *i2c* lainnya.
- q. Ukuran 38x22x14mm.
- r. Berat sebesar 8 gram [29].



Gambar 2.18 *Real Time Clock (RTC)*