

BAB 2

DASAR TEORI

1.1 KAJIAN PUSTAKA

Data berdimensi tinggi merupakan tantangan tersendiri. Dalam persoalan klasifikasi, tantangan tersebut muncul dari kenyataan bahwa dimensi yang lebih tinggi secara drastis meningkatkan ruang vektor sehingga data menjadi terlihat sangat jarang. Hal ini menjadi masalah karena data statistik membutuhkan sampel yang banyak untuk menghasilkan hasil data yang baik secara statistik baik. Diberikan sebuah himpunan untuk pelatihan dengan ukuran tertentu, ada suatu ukuran maksimum untuk vektor ciri dimana ukuran yang lebih besar akan memperburuk klasifikasi daripada memperbaikinya. Reduksi dimensi bertujuan untuk mengatasi persoalan ini dengan melakukan pemetaan data dari ruang berdimensi tinggi ke ruang berdimensi rendah. Pemetaan tersebut dilakukan agar informasi yang relevan tetap muncul pada hasil pemetaan dan di saat yang sama metode-metode pembelajaran dapat diaplikasikan dengan sukses[11].

Principal component analysis (PCA) adalah pelopor metode subspace linear pada reduksi dimensi. *Principal Component Analysis* (PCA) berusaha menemukan transformasi ortogonal yang memproyeksikan data ke suatu *subspace* yang meminimalkan korelasi hasil proyeksi. *Subspace* ini disebut sebagai *principal subspace*. *Principal Component Analysis* (PCA) merupakan suatu metode unsupervised dimana label pada data pelatihan tidak digunakan untuk melakukan pembelajaran. Proyeksi *Principal Component Analysis* (PCA) selalu berusaha mempertahankan semua jenis variasi secara maksimal tanpa menghiraukan faktor-faktor yang memunculkan variasi tersebut. *Linear Discriminant Analysis* (LDA) sendiri merupakan pengklasifikasi generatif yang mengasumsikan bahwa data di setiap kelas terdistribusi secara normal dengan struktur kovarian intra-kelas yang identik. Dalam *reduced rank Linear Discriminant Analysis* (LDA), model generatif tersebut dibatasi agar semua mean dari kelas-kelas yang ada terletak pada suatu subspace linear berdimensi rendah yang sama.

Beberapa penelitian yang terkait dengan penelitian ini diantaranya adalah penelitian yang dilakukan oleh Khairani Dewi (2018) dengan judul “Perancangan Sistem Otomatis Klasifikasi Mutu Teh Hitam Menggunakan Image Processing dan *Artificial Neural Network* (ANN)”. Ekstraksi fitur dilakukan dengan perhitungan rata-rata layer red, green, blue dari layer RGB dan fitur *hue, saturation, value* dari layer HSV. Dengan menggunakan trial dan error, maka layer red, blue, dan hue digunakan sebagai input dari *Artificial Neural Network* (ANN). Klasifikasi menggunakan pengujian *offline* menghasilkan akurasi 91,67% dan menggunakan real time menghasilkan akurasi 86,67% dengan rata-rata waktu proses 0.79 detik [2].

Pada penelitian yang berjudul “Deteksi Cacat Daun *Teh Camellia Sinensis* dengan Pengolahan Citra Digital dan *JST Learning Vector Quantization*” oleh Puji Lestari menggunakan *Principal Component Analysis* (PCA) sebagai metode ekstraksi ciri dan *JST Learning Vector Quantization* (LVQ) sebagai metode klasifikasi. pengujian sistem dilakukan dengan menggunakan 100 citra latih dan 160 citra uji. Parameter terbaik yang digunakan adalah jumlah PC (*Principal Component*) 1-55, epoch 200, learning rate 0,0075, dan *hidden layer* 16. Hasil akurasi dengan parameter tersebut diperoleh ketika menggunakan metode kedua (PCA dan analisis warna) yaitu 78,46% untuk pengklasifikasian empat kelas dan 85,38% untuk pengklasifikasian dua kelas dengan waktu komputasi sebesar 4,447 detik [4].

Penelitian selanjutnya oleh Raden Roro Ayuni Purbo Okta Briliani (2022) dengan judul “Klasifikasi Penyakit Kulit Menggunakan *Image Processing* dan *Artificial Neural Network* (ANN)”. Pada penelitian ini penulis melakukan klasifikasi penyakit kulit menggunakan *Artificial Neural Network* (ANN) dengan memanfaatkan *Image Preprocessing* untuk mendapatkan fitur yang ada pada citra. Fitur yang digunakan pada penelitian ini adalah fitur tekstur dan fitur warna. Fitur tekstur yang digunakan pada penelitian ini adalah *grey-level co-occurrence matrix* (GLCM) sedangkan fitur warna yang di ekstrak adalah warna YCbCr. Dari hasil *cross-validation* diperoleh K ideal yaitu sebesar 10. Dengan menggunakan metode yang diajukan penulis memperoleh akurasi sekitar $(70 \pm 3)\%$.

Pada Penelitian yang dilakukan oleh Christina Natalia (2022) dengan judul “Klasifikasi Jumlah kasus bulanan COVID-19 menggunakan Metode *Artificial*

Neural Network (ANN)". Kinerja model klasifikasi dengan menggunakan metode *Artificial Neural Network (ANN)* dan pengaruh *cross validation* mampu menghasilkan kinerja yang lebih baik. Dengan menerapkan K-Fold pada evaluasi model yang dimana pada evaluasi terhadap 3 skenario yaitu K=3, K=5, dan K=8 menghasilkan model klasifikasi terbaik dengan model KFold = 8 menghasilkan akurasi yang tinggi tepatnya pada bulan Juli 2021 dengan akurasi 98%.

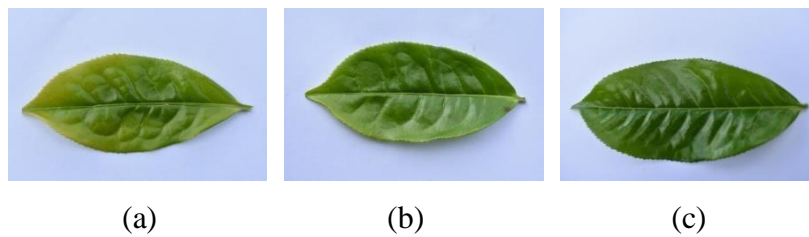
Adapun penelitian serupa yang dilakukan oleh Rahardian Perwira Putra (2022) yang berjudul "RoBERTa sebagai *Semantic Approach* untuk Prediksi *Big Five Personality* menggunakan *Artificial Neural Network (ANN)*". Fokus penelitian ini adalah menyajikan analisis perilaku pengguna Twitter yang dapat digunakan untuk *Big Five Personality Prediction* dari pengguna Twitter menggunakan metode *Artificial Neural Network (ANN)*. Penggunaan metode ini memiliki aspek yang harus diperhatikan yaitu ketepatan klasifikasi sifat *Openness, Conscientiousness, Extraversion, Agreeableness, dan Neuroticism (OCEAN)*. Hasil eksperimen menunjukkan bahwa metode *Artificial Neural Network (ANN)* mendapatkan akurasi 87,09%.

1.2 DASAR TEORI

1.2.1 Dau Teh

Tanaman teh memiliki nama latin yaitu *Camellia Sinensis*. Tanaman teh merupakan salah satu tanaman perdu yang memiliki daun berwarna hijau, tumbuh rendah dekat dengan permukaan tanah, dan tidak mempunyai batang yang tegak. Kematangan daun teh dipengaruhi beberapa faktor seperti iklim yang meliputi suhu, curah hujan, kecepatan angin, dan kelembaban serta kondisi tanah seperti jenis tanah, pH, dan mikroorganisme tanah. Tanaman teh memiliki dua *varietas* yaitu *varietas Assamica* dan *varietas Sinensis*. Pada proposal penelitian tugas akhir ini digunakan jenis teh dari klon *Varietas Assamica*, yakni Teh Gambung. *Varietas Assamica* memiliki keunggulan diantaranya tumbuh cepat, daunnya lebar, dan menghasilkan daun berwarna hijau mengkilat. Teh Gambung yang peneliti ambil sampelnya berada pada musim kemarau menjelang musim hujan. Dengan intensitas curah hujan yang rendah. Teh gambung yang ada pada PT. Perkebunan Nusantara

VIII Unit Sinumbra ini termasuk dalam dataran tinggi [2]. Berikut gambar tingkat kematangan daun Teh Gambung dengan mengambil blok panen yang berbeda.



Gambar 2. 1 Daun Teh (a) belum (b) setengah matang (c) matang [8].

Tanaman teh dapat diperbanyak secara generatif atau vegetatif. Secara generatif tanaman teh dapat diperbanyak dengan menggunakan biji sedangkan secara vegetatif dapat menggunakan stek atau klon. Teknik klon memperbanyak tanaman dengan cara menggabungkan dua atau lebih organ tanaman tertentu seperti rantingnya. Masing-masing tanaman yang akan digabungkan memiliki sifat dan ciri yang berbeda. Tumbuhan yang diperbanyak dengan cara vegetatif maka akan menghasilkan klon yang sama dengan induknya [9]. Tabel 2.1 berikut merupakan klasifikasi tanaman teh.

Tabel 2. 1 Klasifikasi Tanaman Teh

Kingdom	Plantae
<i>Divisi</i>	<i>Spermatophyta</i>
<i>Sub Divisi</i>	<i>Angiospermae</i>
<i>Class</i>	<i>Dicotyledoneae</i>
<i>Ordo</i>	<i>Guttiferales</i>
<i>Family</i>	<i>Tehaceae</i>
<i>Genus</i>	<i>Camellia</i>
<i>Spesies</i>	<i>Camellia Sinensis L</i>
<i>Varietas</i>	<i>Sinensis dan Assamica</i>

1.2.2 Ekstrasi Fitur

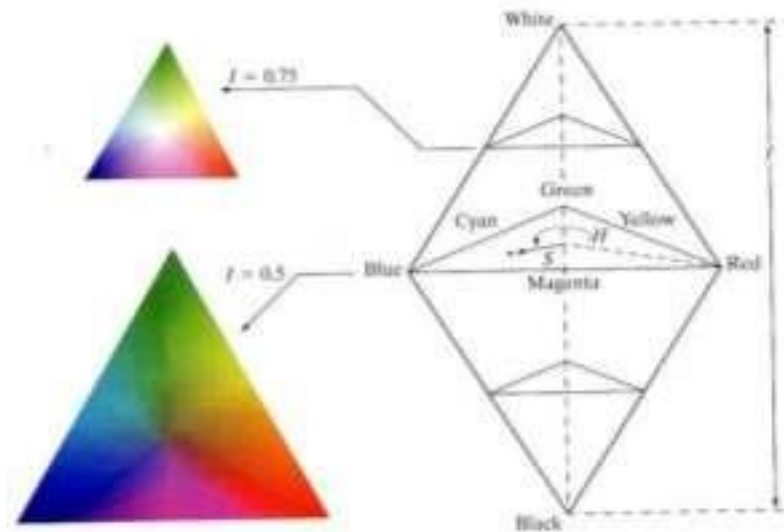
Fitur merupakan karakteristik yang unik dari suatu objek. Fitur dibedakan menjadi dua yaitu fitur alami yang merupakan bagian dari gambar dan fitur buatan yang merupakan perolehan dari operasi suatu gambar. Fitur-fitur suatu objek mempunyai peran yang penting untuk berbagai aplikasi seperti pencarian citra, penyederhanaan dan hampiran bentuk, serta pengenalan dan klasifikasi. Ekstraksi fitur merupakan tahap pengambilan ciri objek yang bisa menjadi bahan pembeda suatu objek dengan objek lainnya [9]. Ekstraksi fitur terbagi menjadi tiga macam yakni ekstraksi fitur bentuk, ekstraksi fitur tekstur, dan ekstraksi fitur warna.

Pada penelitian identifikasi kematangan daun teh kali ini, ciri pembedanya adalah dari warna daun teh. Warna daun teh yang memiliki umur lebih muda akan berwarna hijau terang daripada warna daun teh yang memiliki umur yang lebih tua. Karena ciri pembedanya adalah warna daun teh, ekstraksi fitur yang digunakan adalah ekstraksi fitur warna. Pada penelitian kali ini digunakan fitur warna HSI (*Hue, Saturation, Intensity*) serta HSV (*Hue, Saturation, Value*) [10].

1.2.3 Fitur Warna *Hue, Saturation, Intensity* (HSI)

Selama ini kita telah mengetahui ruang warna RGB (*Red, Green, Blue*) memiliki basis yang bagus dalam mendeskripsikan citra, akan tetapi ruang RGB tidak cocok untuk beberapa aplikasi pemrosesan citra. Dalam memproses citra akan lebih baik jika ruang warna RGB dikonversi ke ruang warna lainnya agar lebih mudah untuk memberikan nilai ambang pada warna asli (*hue*) citra dan akan menyebabkan masalah gamut [10]. Fitur yang digunakan yang pertama adalah HSI (*Hue, Saturation, Intensity*).

Fitur warna HSI mendefinisikan warna dalam terminologi *Hue, Saturation, Intensity*. Hue mendeskripsikan warna mewakili warna dasar, dan ditentukan oleh yang dominan panjang gelombang dalam distribusi spektrum panjang gelombang cahaya. Gambar 2.2 merepresentasikan model warna HSI.



Gambar 2. 2 Model Warna HSI

Dalam mengkonversi dari ruang warna RGB ke ruang warna lain seperti HSI, nilai-nilai RGB pada citra masih diperlukan untuk sebagai dasar perhitungannya. Berikut persamaan-persamaan konversi dari RGB ke HSI [12]:

$$R = 0.8706,$$

$$G = 0.8745,$$

$$B = 0.7216,$$

karena $G \geq B$ maka,

$$H = \begin{cases} \alpha, & \text{jika } G \geq B \\ 360^\circ, & \text{selain itu} \end{cases} \quad (2.1)$$

$$\alpha = \cos^{-1} \left(0,5 \times \frac{R-G+R-B}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \right) \quad (2.2)$$

$$\alpha = \cos^{-1} \left(0,5 \times \frac{0.8706-0.8745+0.8706-0.8745}{\sqrt{(0.8706-0.8745)^2+(0.8706-0.7216)(0.8706-0.7216)}} \right) \quad (2.3)$$

$$H = \alpha = 61.2688 \text{ radian} = 1.0697 \text{ degree}, \quad (2.4)$$

$$S = 1 - \left(3 \frac{\min R,G,B}{R+G+B} \right) = 1 - \left(3 \frac{0.7216}{2.4667} \right) = 1.2244 \quad (2.5)$$

$$I = \frac{R+G+B}{3} = \frac{2.4667}{3} = 0.8222 \quad (2.6)$$

dimana R adalah lapisan pertama RGB (*red*), G adalah lapisan kedua RGB (*green*), dan B adalah lapisan kedua RGB (*blue*).

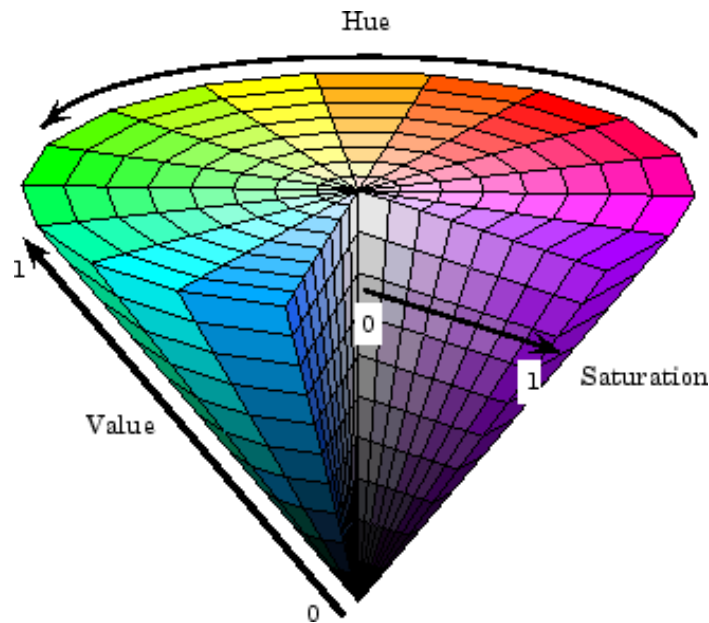
Nilai HSI yang telah diperoleh akan dijadikan suatu nilai fitur. Nilai fitur inilah yang nantinya akan digunakan sebagai ciri tiap citra. Nilai fitur mengkombinasikan komponen warna *Hue*, *Saturation*, dan *Intensity* yang telah didefinisikan pada ruang warna HSI. Berikut persamaan fitur warna HSI [4].

$$feature = 9 \times H' + 3 \times S' + I', \quad (2.7)$$

dimana H adalah nilai *hue*, S adalah nilai *saturation*, dan I adalah nilai *intensity*.

1.2.4 Fitur Warna Hue, Saturation, Value (HSV)

Fitur warna HSV mendefinisikan warna dalam terminologi *Hue*, *Saturation*, *Value*. *Hue* Warna mewakili warna dasar, dan ditentukan oleh yang dominan panjang gelombang dalam distribusi spektrum panjang gelombang cahaya, representasinya dalam bentuk derajat yakni merah pada 0 derajat, hijau pada 120 derajat, dan biru pada 240 derajat. Saturasi (S) adalah keberwarnaan suatu warna, semakin berwarna sebuah warna maka semakin besar nilai saturasinya. *Value* (V) adalah nilai kecerahan sebuah warna. Warna cerah memiliki nilai *value* yang tinggi dan begitupun sebaliknya [13]. Gambar 2.3 mempresentasikan model warna HSV.



Gambar 2. 3 Model Warna HSV

Nilai *hue* tergantung dengan *value* yang dimiliki. Apabila *value* bernilai rendah maka nilai yang dimiliki *hue-hue* sangat sulit dibedakan. Jika nilai *value* = 0 maka, nilai *hue* nya akan sama dengan warna hitam. Jadi berapapun nilai *hue* nya jika nilai *value* = 0 maka semua nilai *hue* akan berwarna hitam. Untuk mendapatkan warna H , S , V berdasarkan R , G , B terdapat cara yang tersederhana menurut Acharya dan Ray yakni seperti persamaan berikut ini [13]:

$$V = \max(R, G, B) = 0.8745,$$

$$Vm = V - (\min R, G, B) = 0.8745 - 0.7216 = 0.1529, \quad (2.8)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ \frac{Vm}{v}, & \text{jika } V > 0 \end{cases} = \frac{0.1529}{0.8745} = 0.1709$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ 60^\circ \times \left(\frac{G - B}{Vm} \text{ mod } 6 \right), & \text{jika } V = R \\ 60^\circ \times \left(2 + \frac{B - R}{Vm} \right), & \text{jika } V = G \\ 60^\circ \times \left(4 + \frac{R - G}{Vm} \right), & \text{jika } V = B \end{cases}$$

Karena $V = G$ maka,

$$H = 60^\circ \times \left(2 + \frac{B-R}{Vm} \right) = 60^\circ \times \left(2 + \frac{0.7216-0.8706}{0.1529} \right) = 0.1709 \quad (2.9)$$

dimana R adalah lapisan pertama RGB (*red*), G adalah lapisan kedua RGB (*green*), dan B adalah lapisan ketiga RGB (*blue*).

Nilai HSV yang telah diperoleh akan dijadikan suatu nilai fitur. Nilai fitur inilah yang nantinya akan digunakan sebagai ciri tiap citra. Nilai fitur mengkombinasikan komponen warna *Hue*, *Saturation*, dan *Value* yang telah didefinisikan pada ruang warna HSV. Berikut persamaan fitur warna HSV [4]

$feature = 9 \times H' + 3 \times S' + V'$ [4], (2.12) dimana H adalah nilai *hue*, S adalah nilai *saturation*, dan V adalah nilai *value*.

1.2.5 Artificial Intelligence (AI)

Artificial Intelligence (AI) atau kecerdasan buatan adalah salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin atau computer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Dalam topik ilmu computer, *Artificial Intelligence* (AI) masih merupakan area penelitian yang pas dan dinamis. Dan sampai sekarang, penelitian menggunakan perkembangan *Artificial Intelligence* (AI) banyak dipergunakan, diantaranya *Neural Network*, *evolutionary computing*, *machine learning*, *natural language processing*, dan *object oriented programming*. Ada beberapa hal, untuk membuat model berpikir manusia yang bisa dituangkan dalam mendesain mesin atau komputer agar dapat persis menirukan cara berpikir manusia. Contohnya seperti cerdas, yang berarti mempunyai pengetahuan dan pengalaman [10].

Berikutnya ada juga penalaran (cara membuat keputusan dan pengambilan Tindakan) dan moral yang baik. Mesin atau computer tidak bisa cerdas dengan sendirinya, tetapi mesin atau computer bisa cerdas dengan di beri arahan atau bekal pengetahuan agar mempunyai kemampuan tersebut. Dalam membuat aplikasi kecerdasan buatan, dibutuhkan dua hal yang menjadi bagian utamanya yaitu:

1. Basis Pengetahuan (*Knowledge Base*), dalam bagian ini bersifat fakta-fakta, terori, pemikiran dan hubungan antar satu dengan yang lainnya.
2. Motor Inferensi (*Inference Enginer*), bagian ini memiliki kemampuan menarik kesimpulan berdasarkan pengalaman.

Untuk penerapan konsep Artificial Intelligence pada mesin atau computer dapat dilihat di Gambar:



Gambar 2. 4 Konsep Artificial Intelligence pada mesin atau computer.

1.2.6 *Machine Learning*

Machine Learning merupakan ilmu dari *computer science* yang membuat komputer atau mesin dapat mempelajari sebuah kecerdasan. Dalam membuat kecerdasan itu, computer atau mesin diharuskan untuk belajar. Dengan begitu machine learning bisa disebut juga dengan suatu bidang keilmuan yang isinya membuat pembelajaran untuk komputer atau mesin agar menjadi cerdas. Fungsi dari *machine learning* adalah untuk mengoptimalkan kriteria atau mengelompokan dengan menggunakan contoh data yang sudah disediakan. Jika sistem dapat belajar dan beradaptasi dengan perubahan tersebut atau dengan cepat, maka pembuat sistem tidak perlu lagi untuk memberikan solusi pada semua kemungkinan situasi [11].

Pada machine learning terdapat tiga bagian learning process, yaitu:

1. *Supervised Learning*

Pembelajaran menggunakan masukan data yang di beri label. Setelah itu membuat prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

Pembelajaran menggunakan masukan data yang tidak diberi label. Setelah itu mencoba untuk mengelompokkan data-data berdasarkan karakteristik yang ditemui.

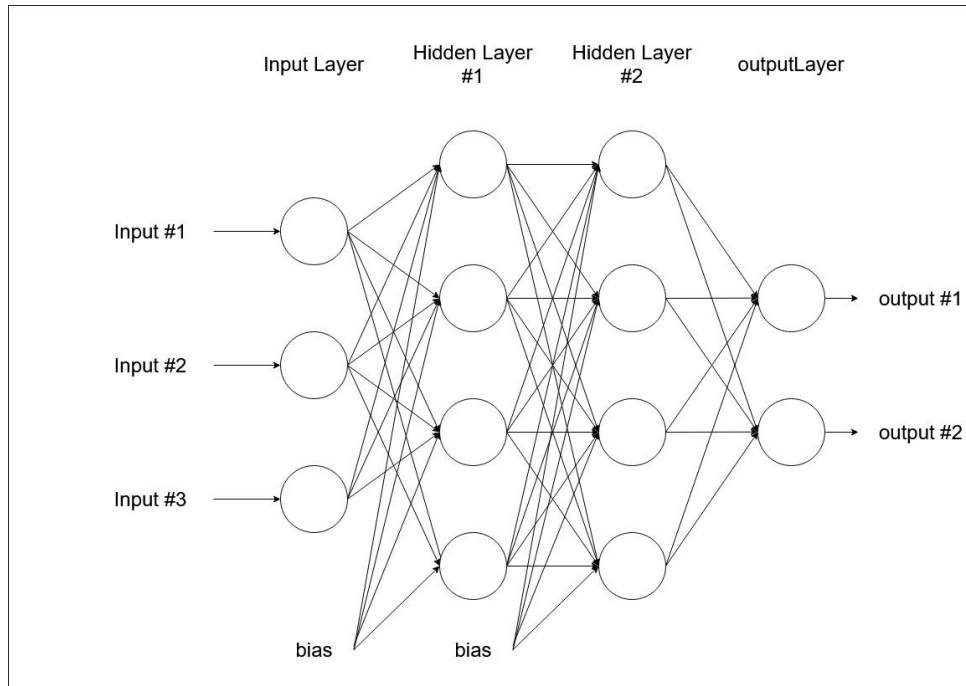
3. *Reinforcement Learning*

Pembelajaran dan *test* saling di campur untuk mengumpulkan informasi pembelajaran secara aktif, dengan berinteraksi ke lingkungan sehingga mendapatkan balasan dari setiap aksi pembelajaran.

Dalam membangun sebuah model matematikanya, *machine learning* menggunakan teori statistic, dikarenakan tugas inti dari *machine learning* untuk membuat kesimpulan dari suatu sampel [19].

1.2.7 *Artificial Neural Network (ANN)*

Artificial Neural Network (ANN) adalah sebuah tiruan dari bagaimana otak bekerja. Otak manusia memiliki kemampuan untuk mempelajari hal baru dan beradaptasi pada lingkungan yang baru dan dinamis. Jaringan syaraf tiruan (JST) atau *Artificial Neural Network (ANN)* merupakan representasi tiruan dari otak manusia dalam melakukan proses pembelajaran pada otak manusia. *Neural network (NN)* terdiri dari sebuah elemen yang disebut saraf (*neuron*). *Neuron* bertujuan untuk mengolah sebuah informasi yang diterima lalu kemudian diteruskan kepada neuron lainnya, cara kerja ini terinspirasi dari sistem saraf secara biologi yang bekerja pada otak manusia. *Artificial neural network (ANN)* merupakan pemodelan data yang baik di mana dapat mengambil serta merepresentasikan hubungan tiap masukan (Input) dan keluaran (output) [8].



Gambar 2. 5 Jaringan Syaraf Tiruan

Terdapat berbagai macam tipe neural network, akan tetapi secara umum menggunakan neural network berjenis multi-layer perceptron (MLP). Multi-layer perceptron (MLP) merupakan jenis neural network yang memanfaatkan banyak lapisan neuron atau biasa disebut hidden layer di mana tiap *neuron* pada *hidden layer* memperoleh nilai masukan dari *neuron* pada pada lapisan sebelumnya. Contoh MLP seperti pada Gambar 2.5, menggunakan dua lapis tersembunyi (*hidden layer*). Tiap-tiap neuron pada masing-masing lapisan (*layer*) dilakukan perhitungan untuk memperoleh keluaran yang akan diteruskan kepada *neuron* pada layer selanjutnya [22].

Proses pada *Artificial Neural Network* (ANN) dimulai dari input yang diterima oleh neuron beserta dengan nilai bobot dari tiap-tiap input yang ada. Setelah masuk ke dalam *neuron*, nilai input yang ada akan dijumlahkan oleh suatu fungsi perambatan (*summing function*), yang bisa dilihat seperti pada di gambar dengan lambang sigma (Σ). Hasil penjumlahan akan diproses oleh fungsi aktivasi setiap *neuron*, disini akan dibandingkan hasil penjumlahan dengan *threshold* (nilai ambang) tertentu. Jika nilai melebihi *threshold*, maka aktivasi *neuron* akan dibatalkan, sebaliknya, jika masih dibawah nilai *threshold*, *neuron* akan diaktifkan. Setelah aktif, neuron akan mengirimkan nilai *output* melalui bobot-bobot outputnya

ke semua *neuron* yang berhubungan dengannya. Proses ini akan terus berulang pada input-input selanjutnya [23].

1.2.8 *Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) merupakan algoritma yang digunakan pada pengolahan sinyal digital untuk pengurangan dimensi data dengan cara mengambil ciri penting dari suatu data atau untuk dekomposisi variabel data. *Principal Component Analysis (PCA)* dapat digunakan dalam teori deteksi, pengenalan pola dan image coding. Keunggulan *Principal Component Analysis (PCA)* adalah dapat mereduksi data dengan cara mencari pola dan melakukan kompresi data tanpa harus kehilangan informasi yang penting. Dengan dimensi data yang rendah ini, waktu komputasi dapat dikurangi. *Principal Component Analysis (PCA)* menghasilkan menghasilkan vektor-vektor *Eigen* atau vektor-vektor karakteristik yang digunakan untuk membuat ruang *Eigen*. Penentuan vektor *eigen* diperlukan dalam mereduksi dimensi data. Urutan Vektor *Eigen* dimulai dari nilai *eigen* paling besar hingga nilai *eigen* paling kecil. Vektor *Eigen* yang dapat direduksi adalah vektor *Eigen* yang memiliki nilai terkecil. Hal ini mengindikasikan bahwa vektor *Eigen* yang direduksi dianggap tidak memiliki informasi terlalu penting sehingga data dapat direduksi tanpa kehilangan pola yang penting [12].

Terdapat lima langkah penting dalam pembentukan *Principal Component Analysis (PCA)*, berikut adalah langkah – langkahnya:

1. Normalisasi

Normalisasi dilakukan untuk menyamakan skala pada data. Cara melakukan normalisasi adalah dengan menghitung rata-rata semua dimensi kumpulan data, kecuali labelnya. Skala data sehingga setiap variabel memberikan kontribusi yang sama untuk analisis. Berikut merupakan persamaan dari normalisasi.

$$z = \frac{x - \mu}{\sigma} \quad (2.10)$$

Dimana:

- Z = nilai skala
- X = inisial
- μ = mean
- σ = standar deviasi

2. Perhitungan *Matriks Kovarian*

Matrix kovarian merupakan *matrix* yang menunjukkan penyebaran data ukuran dimensi lebih besar dari satu. *Matrix kovarian* dapat menentukan ciri sebuah pola dengan cara mengukur penyebaran pola dari kumpulan data. Untuk menghitung nilai kovarian, dapat digunakan persamaan sebagai berikut:

$$\text{Cov}(x, y) = \frac{\sum_{i=1}^n \Sigma(x_i - \bar{x})(y_i - \bar{y})}{(n - 1)} \quad (2.11)$$

Dimana:

x_i = elemen ke- i dari matrix x

y_i = elemen ke- i dari matrix y

\bar{x} = rata rata dari matrix x

\bar{y} = rata rata dari matrix y

3. Hitung *Vector Eigen* dan nilai *Eigen* yang sesuai

Vektor *eigen* atau vektor karakteristik dari transformasi *linier*, adalah vektor bukan nol yang paling banyak berubah oleh faktor skalar ketika transformasi linier tersebut diterapkan padanya. Nilai *eigen* yang sesuai adalah faktor yang digunakan untuk menskalakan vektor *eigen*. Secara umum, vektor *eigen* dari matriks A adalah vektor yang berlaku sebagai berikut:

$$A\vec{v} = \lambda\vec{v} \quad (2.12)$$

$$A\vec{v} - \lambda\vec{v} = 0 \quad (2.13)$$

$$\approx \vec{v}(A - \lambda I) = 0 \quad (2.14)$$

Dimana:

A = matrix A

\vec{v} = vektor eigen

λ = nilai eigen

I = matrix identitas

4. Pilih *Vector Eigen* dengan nilai *Eigen* terbesar

Untuk mendapatkan komponen utama dalam urutan signifikansinya, kita perlu memberi peringkat dalam urutan nilai *eigennya*. Urutkan 11 vektor eigen sesuai dengan urutan penurunan nilai eigennya. Pilih k , dimana k adalah jumlah dimensi yang diinginkan pada dataset baru. Setelah memilih beberapa komponen utama, matriks vektor baru dibuat dan disebut feature vector.

5. Menyusun Kembali data di sepanjang sumbu komponen utama

Untuk Pada langkah terakhir, kita perlu mengubah sampel kita ke sub ruang baru dengan mengarahkan kembali data dari sumbu asli ke sumbu yang sekarang diwakili oleh komponen utama.

$$Data\ Akhir = Feature\ Vector * Transpose(Scaled(Data))$$

1.2.9 *Linear Discriminant Analysis (LDA)*

Linear Discriminant Analysis (LDA) ialah metode pengenalan wajah yang sering dikenal sebagai *Fisher's Linear Discriminant*. *Linear Discriminant Analysis (LDA)* dikenal masyarakat setelah Ronald A. Fisher sebagai penemu metode ini mempublikasikan melalui *paper Tea Use of Multiple Measures in Taxonomic Problems* pada tahun 1936 [13].

Linear Discriminant Analysis (LDA) adalah metode ekstraksi fitur yang merupakan perpaduan dari perhitungan statistika dan matematika yang memberlakukan properti statistik terpisah untuk tiap objek. Tujuan metode *Linear Discriminant Analysis (LDA)* adalah mencari proyeksi linier (yang biasa disebut '*fisherimage*') untuk memaksimalkan matriks kovarian antar-kelas (*between-class covariance matrix*) sekaligus meminimumkan matriks kovarian dalam-kelas (*within-class covariance matrix*), agar anggota didalam 12 kelas lebih terkumpul penyebarannya dan pada akhirnya dapat meningkatkan keberhasilan pengenalan.

Linear Discriminant Analysis (LDA) adalah salah satu metode yang dipakai untuk pengenalan pola pada perhitungan statistika dengan cara menemukan proyeksi linear dari data yang akan memaksimalkan jarak antar kelas dan meminimalkan jarak data yang memiliki kesamaan. Metode ini akan menjadi dasar klasifikasi dari data yang ada. Contohnya pengaplikasian dari kombinasi linear ini adalah data buah mangga. Mangga ini dibedakan menjadi mangga manalagi,

mangga arumanis, dan mangga kweni. Masing-masing mangga ini dibedakan dari kelasnya dan memiliki nilai jumlah N_i , sebanyak n dimensi (dalam contoh ini karena terdiri dari 3 kelas maka $n = 3$). Dari data yang ada tersebut, maka sampel dari data tersebut dapat dinotasikan $\{x_1, x_2, \dots, x_{N_i}\}$, merujuk kepada kelas yang ada. Kemudian data tersebut dikumpulkan menjadi matriks X yang tiap kolom dari matriks ini merepresentasikan dari satu sample data. Setelah itu dicari hasil transformasi dari X ke Y dengan menggunakan proyeksi sample dalam matriks X ke dalam hyperplane dengan dimensi sebesar $K - 1$ [22].

Rumus yang digunakan Dalam melakukan perhitungan *Linear Discriminant Analysis* (LDA) ditunjukkan pada Persamaan 3:

$$f_i = \mu_i C^{-1} x_k^T - \frac{1}{2} \mu_i C^{-1} \mu_i^T + \ln(P_i) \quad (2.15)$$

Dimana:

f_i = Fungsi diskriminan kelas ke- i

μ_i = Rata-rata nilai setiap kelas dari masing-masing matriks x_i

C^{-1} = Inverse dari grup *matriks kovarian*

x_k^T = Transpos dari *matriks kovarian*

P_i = Peluang munculnya kelas ke- i

μ_i^T = *Transpos* rata-rata kelas ke- i

P_i = Peluang munculnya kelas ke- i

1.2.10 Confusion Matrix

Confusion matrix sebagai indikator analisis performa *classifier* dalam mengidentifikasi tupel (sampel) dari kelas yang berbeda. *Confusion matrix* juga dikenal dengan istilah *true positive* (tupel positif) dengan label benar, sedangkan *true negative* (tupel negatif) dengan label benar. Ada juga *false positive* yang merupakan tupel negatif dengan label salah, sedangkan *false negative* (tupel positif) dengan label salah [16].

Tabel 2. 2 Confussion Matrix 3 kelas

Confusion Matrix		Predicted		
		Class 1	Class 2	Class 3
Actual	Class 1	A	B	C
	Class 2	D	E	F
	Class 3	G	H	I

True positives
 True Negatives
 Misclassified cases.

Berikut ini merupakan formulasi perhitungan nilai akurasi, presisi dan *recall* [18]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP \text{ Total}}{Dataset \text{ Total}} \quad (2.16)$$

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{Prediction \text{ Total}} \quad (2.17)$$

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{Actual \text{ Total}} \quad (2.18)$$

Keterangan:

1. *True Positive* (TP), label milik kelas dan diprediksi dengan benar.
2. *False Positive* (FP), label bukan milik kelas tetapi *classifier* diprediksi sebagai positif.
3. *False Negative* (FN), label bukan milik kelas dan diprediksi dengan benar.
4. *True Negative* (TN), label memang milik kelas tetapi diprediksi sebagai negatif.

