

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

BAB II

LANDASAN TEORI

2.1 Huruf Arab (Hijaiyah)

Huruf Hijaiyah adalah abjad Arab yang digunakan dalam penulisan Al-Qur'an dan Hadits. Penulisan Al-Qur'an sendiri telah dimulai sejak zaman Rasulullah SAW. Penulisan Al-Qur'an dilakukan secara berhati-hati, sehingga sampai sekarang Al-Qur'an tetap berada pada orisinalitasnya.

Berdasarkan buku “Cara Belajar dan Menulis Huruf Al-Quran” (Hanifah, Abu, 1981) jumlah huruf Hijaiyah yang umumnya diketahui berjumlah 29 huruf, yaitu : ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن و ه ي .
Cara penulisan antara satu huruf yang dirangkai dengan huruf lain, mempunyai beberapa bagian yaitu:

1. Dapat disambung ke kanan dan kiri. Bagian ini terdapat 22 huruf, yaitu :
ب، ت، ث، ج، ح، خ، س، ش، ص، ض، ط، ظ، ف، ق، ك، ل، م، ن، ه، ي
2. Hanya dapat disambung ke kiri. Bagian ini terdapat 6 huruf, yaitu :
ا، د، ذ، ر، ز، و
3. Tidak dapat disambung ke kanan dan kiri. Bagian ini hanya ada satu huruf, yaitu : ء

Penulisan huruf Arab ke dalam tulisan latin antara satu negara dengan negara yang lainnya bisa saja memiliki perbedaan. Namun yang terpenting adalah pengucapan lafal atau Huruf Arab itu sendiri yang harus tepat dan benar sehingga tidak merubah makna. Berikut ini adalah cara penulisan Huruf Arab ke dalam Bahasa Indonesia menurut Surat Keputusan Bersama (SKB) Menteri Agama Republik Indonesia dan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 158 Tahun 1987 dan Nomor 0543 b/U/1987 tentang Transliterasi Huruf Arab ke dalam Huruf Latin.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.1 Huruf Hijaiyah dan Latin

Bentuk penulisan huruf Hijaiyah dan huruf latinnya adalah sebagai berikut:

No	Arab	Nama	Huruf Latin
1	ا	Alif	-
2	ب	Bâ	B
3	ت	Tâ	T
4	ث	Tsâ	Ts
5	ج	Jim	J
6	ح	Hâ	H
7	خ	Khâ	Kh
8	د	Dâl	D
9	ذ	Dzâl	Dz
10	ر	Râ	R
11	ز	Zây	Z
12	س	Sin	S
13	ش	Syim	Sy
14	ص	Shôd	Sh
15	ض	Dhôd	Dh
16	ط	Thô	Th
17	ظ	Zhô	Zh
18	ع	‘Ain	‘A
19	غ	Ghoïn	Gh
20	ف	Fâ	F
21	ق	Qôf	Q
22	ك	Kâf	K
23	ل	Lâm	L
24	م	Mîm	M
25	ن	Nun	N
26	و	Wâw	W
27	ه	Hâ	H
28	ء	Hamzah	-
29	ي	Yâ	Y

2.2 Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Sebagai contoh, citra burung nuri pada Gambar 2.1 (a) tampak agak gelap, lalu dengan operasi pengolahan citra kontrasnya diperbaiki sehingga menjadi lebih terang dan tajam (b) (Munir, 2004).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.1 (a) Citra burung nuri yang agak gelap, (b) Citra burung yang telah diperbaiki kontrasnya sehingga terlihat jelas dan tajam

Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila (Jain, 1989):

1. perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra,
2. elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur,
3. sebagian citra perlu digabung dengan bagian citra yang lain.

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk ke dalam bidang ini juga adalah pemampatan citra (*image compression*). Pengubahan kontras citra seperti pada Gambar 2.1 adalah contoh operasi pengolahan citra.

Untuk mempermudah proses pengolahan citra biasanya digunakan citra biner. Citra biner (*binary image*) merupakan citra dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas, biasanya 0 dan 1, dimana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta/objek (*foreground*) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih.

2.3 Pengenalan Pola

Pola adalah entitas yang terdefinisi dan dapat didefinisikan melalui ciri-cirinya (*feature*) (Hendradjaya, 1995). Ciri-ciri tersebut digunakan untuk

membedakan suatu pola dengan pola yang lain. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi (Munir, 2004).

Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain. Terdapat dua pendekatan yang dilakukan dalam pengenalan pola: pendekatan secara statistik dan pendekatan secara sintaktik atau struktural (Hendradjaya, 1995).

1. Statistik

Semakin banyak pola yang disimpan, maka sistem akan semakin cerdas. Salah satu contoh penerapannya banyak pada pola pengenalan iris scan.

Kelemahannya: hanya bergantung pada data yang disimpan saja, tidak memiliki sesuatu struktur yang unik yang dapat menjadi kunci pengenalan pola.

2. Sintaksis (*rule*)

Dengan rule/aturan maka sistem yang lebih terstruktur sehingga memiliki sesuatu ciri yang unik. Salah satu contoh penerapannya pada pola pengenalan sidik jari (*fingerprint*).

Struktur sistem pengenalan pola ditunjukkan pada Gambar 2.2. Sistem pengenalan pola ini terdiri dari suatu sensor (misalnya kamera, dan scanner), teknik prapengolahan, suatu algoritma atau mekanisme ekstraksi ciri dan algoritma untuk klasifikasi atau pengenalan (bergantung pada pendekatan yang dilakukan). Sebagai tambahan, biasanya beberapa data yang sudah diklasifikasikan diasumsikan telah tersedia untuk melatih sistem (Putra, 2010).



Gambar 2.2 Struktur Sistem Pengenalan Pola (Putra, 2010)

Prapengolahan adalah transformasi input (masukan) data mentah untuk membantu kemampuan komputasional dan pencarian ciri serta untuk mengurangi noise (derau). Pada prapengolahan citra (sinyal) yang ditangkap oleh sensor akan

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dinormalisasi agar citra menjadi lebih siap untuk diolah pada tahap pemisahan ciri. Kualitas ciri yang dihasilkan pada proses pemisahan ciri sangat bergantung pada hasil prapengolahan.

Ekstraksi ciri merupakan proses mengambil ciri-ciri yang terdapat pada objek di dalam citra. Pada proses ini objek di dalam citra mungkin perlu dideteksi seluruh tepinya, lalu menghitung properti-properti objek yang berkaitan sebagai ciri. Beberapa proses ekstraksi ciri mungkin perlu mengubah citra masukan sebagai citra biner, melakukan penipisan pola, dan sebagainya.

Klasifikasi merupakan tahap untuk mengelompokkan input data pada satu atau beberapa kelas berdasarkan hasil pencarian beberapa ciri yang signifikan dan pemrosesan atau analisis terhadap ciri itu. Setiap kelas terdiri dari sekumpulan objek yang memiliki kedekatan (kemiripan) ciri.

2.4 Principal Components Analysis (PCA)

Principal Component Analysis (PCA) merupakan suatu perhitungan standar modern yang digunakan untuk analisis data pada beragam *field* atau multi dimensi sekumpulan data (dataset) khususnya pada bidang komputer grafik. Metode ini dinilai mudah karena tidak membutuhkan parameter khusus dalam ekstraksi informasi yang berhubungan terhadap sekumpulan data yang meragukan (Smith, 2002).

Metode PCA digunakan untuk mereduksi dimensi variable data input menjadi komponen utama yang berdimensi lebih kecil dengan kehilangan informasi minimum. Dengan usaha minimal, PCA mampu menyediakan alur bagaimana mengurangi kumpulan data yang kompleks ke dalam dimensi yang lebih kecil (Purnomo dan Muntasa, 2010).

Algoritma ini diperkenalkan pada tahun 1933 oleh H. Hotelling. Oleh karena itu sering juga disebut transformasi Hotelling. Untuk pertama kalinya, PCA dikembangkan oleh para ahli statistik dalam mengeksplorasi hubungan sejumlah variabel kualitatif yang dikembangkan oleh Karl Pearson pada tahun 1901. Karl Pearson sendiri telah menganalisis matrik korelasi yang berasal dari

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

pengukuran tujuh variabel fisik untuk tiap orang dari 3000 pelaku kriminal (Purnomo dan Muntasa, 2010).

Algoritma *Principal Component Analysis* (PCA)

Dalam penggunaan metode PCA, ada beberapa hal yang harus diperhatikan yaitu, citra harus tepat berada ditengah dan memiliki ukuran yang sama dengan data latih. Berikut ini adalah langkah-langkah dalam proses PCA atau *eigenface* untuk data latih (Turk & Pentland, 1991):

1. Membuat matriks rata-rata (*mean*) ψ

Rumus perhitungan matriks rata-rata adalah Persamaan 2.1 sebagai berikut:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \dots\dots\dots (2.1)$$

Dari rumus di atas maka akan didapatkan hasil $\Psi = [\Psi_1, \Psi_2, \Psi_3, \dots \Psi_m]$

Ψ = matriks rata-rata (*mean*)

M = banyaknya data di data set

Γ_n = data ke- n

2. Membuat matriks selisih Φ

Matriks selisih Φ dibuat dengan mengurangi setiap elemen matriks Γ pada kolom ke- i dengan ψ . Representasi matriks Φ dapat dilihat pada Persamaan 2.2 dibawah:

$$\Phi_i = \Gamma_i - \Psi \dots\dots\dots (2.2)$$

Dari rumus diatas dapat direpresentasikan ke dalam matriks selisih berikut ini:

$$\Phi = \begin{bmatrix} \Gamma_{11} - \Psi_1 & \Gamma_{12} - \Psi_2 & \Gamma_{13} - \Psi_3 & \dots & \Gamma_{1n} - \Psi_n \\ \Gamma_{21} - \Psi_1 & \Gamma_{22} - \Psi_2 & \Gamma_{23} - \Psi_3 & \dots & \Gamma_{2n} - \Psi_n \\ \Gamma_{31} - \Psi_1 & \Gamma_{32} - \Psi_2 & \Gamma_{33} - \Psi_3 & \dots & \Gamma_{3n} - \Psi_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Gamma_{N1} - \Psi_1 & \Gamma_{N2} - \Psi_2 & \Gamma_{N3} - \Psi_3 & \dots & \Gamma_{Nn} - \Psi_n \end{bmatrix}$$

Φ = matriks selisih

Γ_i = data ke- i

Ψ = data rata-rata (*mean*)

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. Membuat matriks kovarian (C)

Rumus penghitungan matriks kovarian C adalah Persamaan 2.3 sebagai berikut:

$$C = \frac{1}{M} \sum_{n=1}^M (\Phi_n)(\Phi_n)^T = AA^T \text{ atau } L = A^T A \dots\dots\dots (2.3)$$

$$A = \{\Phi_1, \Phi_2, \Phi_3, \dots \Phi_n\}$$

C = matriks kovarian berdimensi n x n.

A = berisi matriks selisih

Φ = matriks selisih

M= jumlah data set

4. Menghitung *eigenvector* (v) dan *eigenvalue* (val) dari matriks kovarian (C)

Rumus untuk menghitung *eigenvector* dan *eigenvalue* dari matriks kovarian adalah Persamaan 2.4 sebagai berikut:

$$Cv = \lambda v \dots\dots\dots (2.4)$$

Dari rumus diatas, maka didapatkan sebuah persamaan 2.5 sebagai berikut:

$$(Cv - \lambda v) = 0$$

$$(C - \lambda I) = 0 \text{ atau } \det(\lambda I - C) = 0 \dots\dots\dots (2.5)$$

Persamaan 2.6 diatas digunakan untuk mencari nilai *eigenvalue* (λ), sedangkan untuk mencari nilai *eigenvector* (v), maka kita akan mensubstitusikan nilai dari *eigenvalue* (λ) ke dalam persamaan 2.6 :

$$(\lambda I - C)v = 0 \dots\dots\dots (2.6)$$

Keterangan :

λ = *eigenvalue* atau nilai eigen

v = *eigenvector* atau vektor eigen

C = matriks kovarian

I = matriks identitas

5. Mengurutkan nilai eigen value (λ) dan eigen vektor (v) dari besar ke kecil berdasarkan urutan nilai eigen. Kemudian hitung nilai matriks eigenfaces dengan persamaan berikut ini:

$$Eigenfaces = eigenvector \times \Phi \dots\dots\dots (2.7)$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

6. Menghitung nilai ciri PCA dari citra data uji dengan mengalikan nilai *eigenfaces* yang telah didapatkan dari proses ekstraksi citra data latihan dengan matriks normalisasi dari citra data uji. Nilai *Project Image* atau nilai ciri PCA dapat dihitung dengan menggunakan persamaan berikut:

$$Project\ image = \Phi \times eigenfaces^T \dots\dots\dots (2.8)$$

7. Mencari nilai *principal component* (PC)

Pada proses 5 akan dihasilkan n *eigenvektor* dan n *eigenvalue* . Dari n *eigenvektor* diambil m *eigenvektor* v yang memiliki *eigenvalue* terbesar, karena *eigenvektor* ini memiliki dimensi yang sama dengan citra awal, maka vektor eigen ini juga bisa disebut *principal component* (PC) atau komponen utama. Nilai PC dapat juga dicari dengan Persamaan 2.9 sebagai berikut :

$$u_i = \sum_{k=1}^M v_{lk} \Phi_k \dots\dots\dots (2.9)$$

$$l = 1, 2, \dots M$$

Keterangan:

u = nilai *principal component* (PC)

v = *eigenvektor*

Φ = nilai selisih

M = jumlah data

8. Tentukan nilai PC terpilih sebanyak N. Misal N=2, maka:

$$u = \begin{bmatrix} u_{11} & \dots & u_{M-1\ m-1} & u_{M\ m} \\ \vdots & \ddots & \vdots & \vdots \\ u_{1n} & \dots & u_{M-1\ nm-1} & u_{M\ nm} \end{bmatrix} \dots\dots\dots (2.10)$$

9. Cari bobot masing-masing citra referenci dengan cara:

$$w = \Phi x uN \dots\dots\dots (2.11)$$

uN adalah matriks yang berisi nilai PC sebanyak N.

$$w = \begin{bmatrix} w_1 & \dots & w_N \\ \vdots & \ddots & \vdots \\ w_M & \dots & w_{MN} \end{bmatrix}$$

M = Jumlah baris, N = Jumlah kolom

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Setelah bobot citra latih ditemukan, langkah selanjutnya adalah menemukan bobot citra yang diujikan. Berikut adalah langkah-langkah algoritma PCA untuk ekstraksi ciri citra data uji: (Syakhala, 2015):

1. Menghitung Matriks Normalisasi dari Citra Data Uji

Pada tahap ini membutuhkan nilai mean dari citra data latih dan nilai biner dari citra data uji yang akan digunakan untuk mencari matriks normalisasi atau matriks selisih. Berikut adalah persamaan yang dapat digunakan untuk menghitung matriks normalisasi:

$$\Phi_i = \Gamma_i - \Psi \dots\dots\dots (2.12)$$

Masing-masing nilai pada setiap piksel (Γ_i) dari citra data uji kemudian dikurangkan dengan rata-ratanya (Ψ) yang merupakan hasil dari proses ekstraksi ciri citra data latih.

2. Menghitung Nilai *Project Image* dari Citra Data Uji

Langkah selanjutnya adalah menghitung nilai *Project Image* dari citra data uji dengan mengalikan nilai *eigenfaces* yang telah didapatkan dari proses ekstraksi citra data latih dengan matriks normalisasi dari citra data uji. Nilai *Project Image* dapat dihitung dengan menggunakan persamaan berikut:

$$Project\ image = \Phi \times eigenfaces^T \dots\dots\dots (2.13)$$

Setelah perhitungan algoritma PCA selesai dan nilai *Project Image* telah diperoleh, maka langkah selanjutnya mereduksi matriks *Project Image* dengan variabel k yang dapat disesuaikan dengan kebutuhan. Kemudian setelah matriks *Project Image* direduksi memasukkan nilai *Project Image* yang berupa matriks dua dimensi tersebut ke dalam proses algoritma *Backpropagation* untuk dilakukan identifikasi.

2.5 Normalisasi data

Normalisasi digunakan untuk mempersempit range data latih. Normalisasi yang digunakan adalah min-max normaization yang merupakan proses transformasi nilai dari data yang dikumpulkan pada rang value 0.0 dan 1.0,

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dimana nilai terkecil (min) adalah 0.0 dan nilai tertinggi (max) adalah 1.0 seperti yang di tunjukkan pada persamaan 2.16 (Elvianti, 2015).

$$v^T = \frac{v - \min_a}{\max_a - \min_a} (\text{new_max}_a - \text{new_min}_a) + \text{new_min}_a \dots\dots\dots (2.17)$$

Dimana:

- v^T = Data baru setelah normalisasi
- v = Data sebelum normalisasi
- new_max_a = Batas nilai max baru adalah 1
- new_min_a = Batas nilai min baru adalah 0
- max_a = Nilai maksimum pada kolom
- min_a = Nilai minimum pada kolom

2.6 Jaringan Syaraf Tiruan

Menurut (Haykin, S., 1994), mendefinisikan jaringan saraf sebagai berikut: Sebuah jaringan saraf adalah sebuah prosesor yang terdistribusi besar-besaran secara parallel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu: (1) Pengetahuan diperoleh oleh jaringan melalui suatu proses belajar; (2) Kekuatan hubungan antar sel saraf yang dikenal dengan bobot sinapsis digunakan untuk menyimpan pengetahuan.

Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian Jaringan syaraf tiruan. Neuron terdiri atas tiga elemen pembentuk (Siang, 2005):

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot / kekuatan yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawanya. Jumlah, Struktur, dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan (dan juga model jaringan yang terbentuk).
2. Suatu unit penjumlah yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain ataukah tidak.

2.6.1 Arsitektur Jaringan Syaraf Tiruan

Pada jaringan saraf tiruan, neuron-neuron akan dikumpulkan dalam sebuah lapisan yang disebut dengan lapisan neuron (*neuron layer*). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan lainnya. Informasi yang didapatkan pada sebuah neuron akan disampaikan ke semua lapisan-lapisan yang ada. mulai dari lapisan masukan sampai dengan lapisan keluaran melalui lapisan tersembunyi (*hidden layer*). Secara umum, ada tiga jenis arsitektur dari jaringan saraf tiruan yaitu (Haykin, 2009):

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

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

- b) Jaringan dengan banyak lapisan (*multilayer net*)

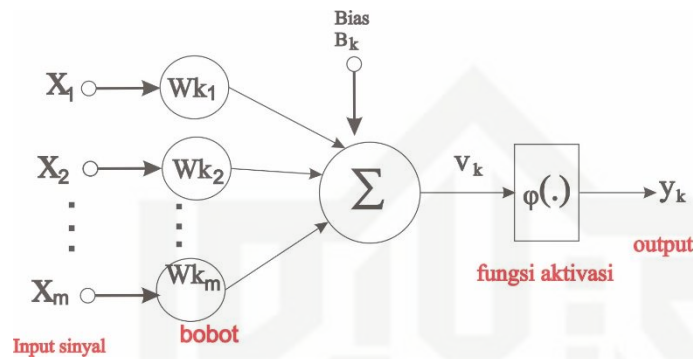
Jaringan dengan banyak lapisan memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki 1 atau lebih lapisan tersembunyi). Umumnya, ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada jaringan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, pada banyak kasus, pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.

- c) Jaringan dengan lapisan kompetitif (*competitive layer net*)

Bentuk lapisan kompetitif merupakan jaringan saraf tiruan yang sangat besar. Interkoneksi antar neuron pada lapisan ini tidak ditunjukkan pada arsitektur seperti jaringan yang lain. Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif atau sering pula disebut

dengan prinsip *winner takes all* atau yang menanglah yang mengambil semua bagian.

Perhatikan Gambar 2.3 yang merupakan gambar arsitektur jaringan saraf tiruan sederhana di bawah ini:



Gambar 2.3 Arsitektur Jaringan Saraf Tiruan Sederhana (Siang, 2004)

Pada gambar di atas, Y menerima masukan dari neuron x_1 , x_2 , dan x_3 , dengan bobot hubungan masing-masing adalah w_1 , w_2 , dan w_3 . Ketiga impuls neuron yang ada di jumlahkan menjadi $Net = x_1w_1 + x_2w_2 + x_3w_3$. Besarnya impuls yang diterima Y mengikuti fungsi aktivasi $y=f(Net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot (Siang, 2004).

2.6.2 Fungsi Aktivasi

Fungsi aktivasi bertujuan untuk menentukan nilai keluaran berdasarkan nilai total masukan pada neuron. Fungsi aktivasi suatu algoritma jaringan dapat berbeda dengan fungsi aktivasi algoritma jaringan lain. Banyak fungsi yang dapat dipakai sebagai aktivasi seperti fungsi-fungsi goniometri dan hiperboliknya, fungsi unit step, impulse sigmoid, dan lain sebagainya, tetapi yang lazim digunakan adalah fungsi sigmoid, karena dianggap lebih mendekati kinerja sinyal pada otak (Jong,2005).

Fungsi sigmoid memiliki dua jenis, yaitu fungsi sigmoid biner dan fungsi sigmoid bipolar. Berikut ini penjelasannya:

1. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode backpropagation. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1. Fungsi sigmoid biner dirumuskan sebagai :

$$y = f(x) = \frac{1}{1+e^{-\sigma x}} \dots\dots\dots (2.14)$$

Dengan $f'(x) = \sigma f(x)[1 - f(x)]$

2. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1.

Fungsi sigmoid bipolar dirumuskan sebagai :

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}} \dots\dots\dots (2.15)$$

Dengan $f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$

Fungsi ni sangat dekat dengan fungsi hyperbolic tangent. Keduanya memiliki range antara -1 sampai 1. Untuk fungsi hyperbolic tangent, dirumuskan sebagai :

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \dots\dots\dots (2.16)$$

Atau

$$y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$$

dengan $f'(x) = [1 + f(x)][1 - f(x)]$

2.7 Metode Backpropagation Neural Network (BPNN)

Algoritma pelatihan *Backpropagation* atau diterjemah menjadi propagasi balik, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada JST, dan selanjutnya algoritma ini bisa disingkat dengan BP. Algoritma ini termasuk metode pelatihan *supervised* dan didesain untuk operasi pada jaringan *feed forward* milti lapis (Jong, 2005).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

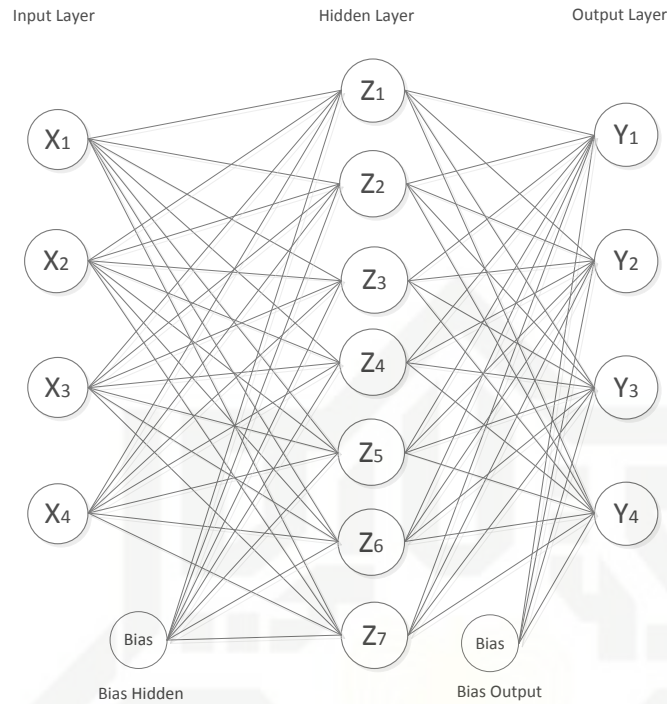
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Secara garis besar, mengapa algoritma ini disebut propagasi balik, dapat dideskripsikan sebagai berikut. Ketika jaringan diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuu ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan keluaran. Kemudian unit-unit lapisan keluaran memberikan tanggapan yang disebut sebagai keluaran jaringan. Saat keluaran jaringan tidak sama dengan keluaran yang diharapkan maka keluaran akan menyebar mundur (*backward*) pada lapisan tersembunyi diteruskan ke unit pada lapisan masukan. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan backpropagation / propagasi balik.

2.7.1 Arsitektur Backpropagation

Arsitektur *backpropagation* dibagi menjadi 3 lapisan penyusun, yaitu (Sudarsono, 2016):

1. Lapisan *Input (Input Layer)* merupakan lapisan yang terdiri dari beberapa *neuron* yang akan menerima sinyal dari luar dan akan diteruskan ke *neuron-neuron* selanjutnya.
2. Lapisan tersembunyi (*hidden layer*) lapisan yang berfungsi meningkatkan kemampuan jaringan dalam memecahkan masalah. Jumlah simpul yang ditentukan oleh pengguna dengan menggunakan persamaan $X < Hidden < X * 2$ (Rajasekaran, 2007).
3. Lapisan *output (output layer)* san yang berfungsi menyalurkan sinyal-sinyal keluaran dari hasil pemrosesan jaringan yang merupakan solusi JST terhadap suatu permasalahan. Terdiri dari 1 simpul.



Gambar 2.4 Arsitektur *Backpropagation* (Susanti, 2014)

2.7.2 Algoritma pelatihan *Backpropagation*

Algoritma pelatihan *Backpropagation* terdiri dari dua proses. *Feed forward* dan *backpropagation* dari galatnya. Untuk jelasnya dapat dijelaskan sebagai berikut (Jong, 2005):

Langkah 0:

Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil)

Langkah 1:

Ulangi langkah 2 hingga 9 sampai kondisi akhir iterasi dipenuhi

Langkah 2:

Untuk masing-masing pasangan data pelatihan (training data) lakukan langkah 3 hingga 8

Feedforward (Propagasi maju):

Langkah 3:

Tiap-tiap unit input (X_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan

sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

Langkah 4:

Tiap-tiap unit tersembunyi ($Z_j, j=1,2,3,\dots,p$) menjumlahkan sinyal-sinyal input terbobot :

$$z_{in_j} = v_{0j} \sum_{i=1}^n x_i v_{ij} \dots\dots\dots(2.18)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya :

$$z_j = f(z_{in_j}) \dots\dots\dots(2.19)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Langkah 5:

Tiap-tiap unit output ($Y_k, k=1,2,3,\dots,m$) menjumlahkan sinyal-sinyal input terbobot

$$y_{in_k} = w_{0k} + \sum_{i=1}^p z_i w_{jk} \dots\dots\dots(2.20)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya :

$$y_k = f(y_{in_k}) \dots\dots\dots(2.21)$$

Dan kirimkan sinyal tersebut ke semua unit dilapisan atasnya (unit-unit output)

Backpropagation.dan Galatnya

Langkah 6:

Tiap-tiap unit output ($Y_k, k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi *error* nya :

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \dots\dots\dots(2.22)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}) :

$$\Delta w_{jk} = \alpha \delta_k z_k \dots\dots\dots(2.23)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}) :

$$\Delta w_{0k} = \alpha \delta_k \dots\dots\dots(2.24)$$

Kirimkan δ_k ini ke unit-unit yang ada dilapisan bawahnya.

Langkah 7:

Tiap-tiap unit tersembunyi ($Z_j, j=1,2,3,\dots,p$) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya) :

$$\delta_{in_j} = \sum_{i=1}^m z_i w_{jk} \dots\dots\dots(2.25)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error :

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \dots\dots\dots(2.26)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{jk} = \alpha \delta_j x_i \dots\dots\dots(2.27)$$

Kemudian hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}):

$$\Delta v_{0j} = \alpha \delta_j \dots\dots\dots(2.28)$$

Langkah 8:

Tiap-tiap unit output ($Y_k, k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,\dots,p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots(2.29)$$

Tiap-tiap unit tersembunyi ($Z_j, j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,\dots,n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots\dots\dots(2.30)$$

Langkah 9:

Tes Kondisi berhenti

2.8 Confusion Matrix

Menurut (Han, dkk., 2012) performa dari suatu model klasifikasi dapat diukur dengan tingkat akurasi berdasarkan *Confusion Matrix*. *Confusion Matrix* merupakan alat yang berguna untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas yang berbeda. TP dan TN memberi informasi ketika *classifier* benar, sedangkan FP dan FN memberi informasi ketika *classifier* salah. Tabel 2.2 adalah contoh dari *Confusion Matrix*.

- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.2 Confusion Matrix

Classification	Predicted Class		
	Class = Yes	Class = No	
Observed Class	Class = Yes	A (TP)	B (FN)
	Class = No	C (FP)	D (TN)

Keterangan:

TP (*True Positive*) = proporsi positif dalam data set yang diklasifikasi positif.

TN (*True Negative*) = proporsi positif dalam data set yang diklasifikasi negatif

FP (*False Positive*) = proporsi negatif dalam data set yang diklasifikasi positif

FN (*False Negative*) = proporsi negatif dalam data set yang diklasifikasi negatif

Berikut adalah persamaan model *Confusion Matrix*:

- a. Nilai akurasi

$$accuracy = \frac{(a+d)}{(a+b+c+d)} \dots\dots\dots(2.31)$$

- b. *Error rate*

$$error\ rate = \frac{(b+c)}{(a+b+c+d)} \dots\dots\dots(2.32)$$