

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan acuan utama pada penelitian ini berupa beberapa studi yang pernah dilakukan berkaitan dengan penelitian.

Penelitian yang dilakukan Agustin dan Prasetyo (2011) untuk klasifikasi jenis pohon mangga gadung dan mangga curut berdasarkan tekstur daun menggunakan daun mangga hijau dan fitur warna tekstur yang digunakan adalah warna green dari bagian warna RGB sedangkan untuk fitur tekstur yang digunakan adalah rata-rata intensitas, *smoothness*, *entropy*, *5 moment invariant*, *energy* dan *kontras*. Data yang digunakan dengan 30 daun mangga gadung dan 30 mangga curut dengan hasil akurasi pada K-NN sebesar 54.24% sedangkan menggunakan *Backpropagation* dengan akurasi sebesar 65.19%.

Pada penelitian oleh Astuti dkk (2015) mengenai diagnosis gangguan ginjal melalui citra iris mata menggunakan 16 citra iris mata yang diuji dengan deteksi tepi *canny* berdasarkan metode *backpropagation* diagnosa gangguan ginjal terbaik terjadi pada iterasi ke-2 dengan nilai *hiddenlayer* 30, konstanta belajar 0.7, tampilan per iterasi 10000, galat 0.01 dan rasio kesalahan sebesar 0.7 sehingga pada penelitian tersebut menunjukkan 100% data uji yang dikenali sesuai target.

Pada penelitian oleh Rabersyah dkk (2016) dengan memanfaatkan *electronic nose* untuk membedakan jenis kopi. Karakteristik kopi dapat ditentukan berdasarkan kandungan gas pada kopi yang terdiri dari 5 sensor gas yaitu TGS 2610, TGS 2611,

TGS 2602, TGS 2620 dan TGS 822. Pola data diperoleh dari perubahan resistansi masing – masing sensor apabila mendeteksi aroma kopi yang mengakibatkan perubahan tegangan. Pola data tersebut akan diolah menggunakan jaringan syaraf tiruan (JST) backpropagation. Arsitektur JST yang digunakan dibentuk dari 5 node input, 6 node hidden dan 2 node output. Hasil pengujian memperlihatkan JST backpropagation mampu melakukan identifikasi dengan tingkat keberhasilan 40 % untuk arabika, 100 % untuk robusta dan 100 % untuk udara bebas (tanpa kopi).

Pada penelitian oleh Sebatubun dan Nugroho (2017) mengenai pengenalan varietas kopi arabika dengan segmentasi menggunakan metode otsu untuk ekstraksi fitur bentuk menggunakan *circularity* dan, klasifikasi *multilayer perceptron*. dari 30 citra kopi mandailing MLP mampu mengenali 25 citra sedangkan 5 citra dikenali sebagai citra kopi lini S-795 dan dari 30 citra kopi lini S-795 dikenali 23 citra sedangkan 7 citra dikenali sebagai citra kopi sigarar utang sehingga nilai akurasi sebesar 80% dengan sensitivitas sebesar 83.33% dan spesifisitas sebesar 76.7%.

Pada penelitian yang dilakukan oleh Qoriah (2017) untuk mengenali jenis beras menggunakan metode *backpropagation* dengan deteksi tepi *canny* dari 4 jenis beras nilai akurasi ketepatan data identifikasi pada saat pengujian sebesar 70%.

Pada penelitian yang akan dilakukan penulis yaitu identifikasi jenis biji kopi *specialty arabica gayo* dengan menggunakan deteksi tepi *canny* dan metode yang digunakan *backpropagation* dengan data *training* dan data *testing* sebanyak 120 data dari 6 jenis biji kopi *specialty arabica gayo* untuk hasil identifikasi dan akurasi ketepatan sistem belum diketahui karena masih dalam tahap penelitian.

Untuk mempermudah perbandingan tinjauan pustaka dengan penelitian, maka dibuat tabel 2.1 sebagai berikut :

Tabel 2.1 Ringkasan Penelitian

Penulis	Objek	Metode Ekstraksi ciri	Metode pengenalan Pola	Hasil
Agustin dan Prasetyo (2011)	Daun mangga gadung dan daun mangga curut	Ekstraksi fitur : (rata-rata intensitas, smoothness, entropy, 5 moment invariant, energy, dan kontras)	K-NN dan <i>Backpropagation</i>	Klasifikasi menggunakan 30 daun mangga gadung dan 30 daun mangga curut. Hasil K-NN memberikan hasil akurasi keseluruhan 54.24% sedangkan JST <i>backpropagation</i> memberikan akurasi 65.19%.
Astuti, Edkk (2015)	Image iris mata	Deteksi tepi	<i>Backpropagation</i>	Dari 16 citra iris mata yang diuji, program ini dapat mengenali 16 citra sehingga pengenalannya 100%.
Rabersyah, Edkk (2016)	Bubuk Kopi	Electronic nose	<i>backpropagation</i>	Dengan <i>backpropagation</i> sistem mampu melakukan identifikasi jenis bubuk kopi dengan tingkat keberhasilan untuk udara netral (tanpa kopi), kopi arabika dan kopi robusta secara berturut-turut adalah 100%, 40% dan 100%.
Sebatubun dan Nugroho (2017)	Biji kopi	metode Otsu	<i>Circularity</i>	Hasil klasifikasi menunjukkan tingkat akurasi yang diperoleh sebesar 80%, sensitivitas 83,33% dan spesifisitas 76,7%.
Rizki Qoriah (2017)	Gambar beras	Segmentasi tepi <i>canny</i>	Jaringan syaraf tiruan <i>backpropagation</i>	Jenis beras dan akurasi keberhasilan pengenalan dengan akurasi sistem sebesar 70%
Yang diusulkan	Gambar Biji kopi	Segmentasi tepi <i>canny</i>	<i>backpropagation</i>	Jenis kopi <i>specialty</i>

2.2 Dasar Teori

2.2.1 Kopi

Kopi adalah spesies tanaman berbentuk pohon yang termasuk dalam famili *Rubiaceae* dan genus *Coffea*. Tanaman ini tumbuh tegak, bercabang, dan tingginya dapat mencapai 12 m. Daunnya bulat telur dengan ujung agak meruncing. Daunnya tumbuh berhadapan pada batang, cabang, dan ranting (Najiyati, 2008).

Kopi arabika berasal dari Ethiopia dan masih bisa ditemukan tumbuh di alam liar di hutan tropis dataran tinggi. Penanaman awal dan pemilihan spesies dibawa oleh Arab yang mengenalkannya pada Yaman, kemungkinan pada abad ke-13 atau ke-14. Varietas lokal yang dipilih di Yaman sudah merupakan basis dari budidaya kopi di seluruh wilayah yang menanam kopi dengan kemungkinan pengecualian Ethiopia. Sekitar tahun 1700 tanaman kopi dari Yaman dikenalkan oleh Belanda ke Indonesia dan dari sana beberapa puluh tahun kemudian ke Amerika Tengah dan Selatan. Material kopi tersebut diambil dari satu pohon yang ditanam oleh Belanda di Taman Botani di Amsterdam, kopi itu diberi nama "*Typica*" atau "*Arabica*" dan sudah secara luas dikembangkan di benua Amerika selama sekitar dua abad, sekarang masih tumbuh di beberapa Negara termasuk Indonesia (Wintgens, 2004).

Kopi jenis Arabika sangat baik dibudidayakan di daerah yang berketinggian 1.000 – 2.100 mdpl. Semakin tinggi lokasi perkebunan kopi, citarasa yang dihasilkan oleh biji kopi akan semakin baik. Karena itu, perkebunan Kopi Arabika hanya

terdapat di beberapa daerah tertentu (di daerah yang memiliki ketinggian di atas 1.000m) (Panggabean, 2011).

Metode pengolahan biji arabica dilakukan dengan 6 cara yang berbeda, yaitu pengolahan basah (*fullwashed*), pengolahan kering *honey*, premium, Luwak, *Wine* dan pengolahan kering-natural. Metode pengolahan basah dilakukan dengan cara buah kopi hasil panen dikupas secara mekanis menggunakan mesin pulper untuk memisahkan kulit buah dari bijinya, kemudian difermentasi selama 24 jam, setelah itu dicuci sampai bersih dan langsung dijemur di bawah sinar matahari. Pengolahan kering *honey* dilakukan dengan cara buah kopi hasil panen dikupas secara mekanis menggunakan mesin pulper untuk memisahkan kulit buah dari bijinya. Biji yang sudah terkupas kemudian dijemur langsung di bawah sinar matahari. Sedangkan pengolahan kering-natural, buah kopi hasil panen langsung dijemur di bawah sinar matahari, hingga kadar airnya mencapai $\pm 12\%$. Biji kering yang masih berkulit tanduk dikupas secara manual menggunakan alat tumbuk yang terbuat dari kayu. Biji beras hasil dari ketiga metode pengolahan tersebut kemudian disortasi untuk memisahkan biji yang pecah, bolong, tidak utuh, hitam, dan hampa (Sherfey, 2016 ;Turp, 2016).

Gambar biji kopi ditunjukkan pada bab 2.1 – 2.6 :



Gambar 2.1 Biji kopi Luwak



Gambar 2.2 Biji kopi Natural



Gambar 2.3 Biji kopi Wine



Gambar 2.4 Biji kopi *Honey*



Gambar 2.5 Biji kopi *Fullwashed*



Gambar 2.6 Biji kopi *Wash*

2.2.2 Definisi Jaringan Syaraf Tiruan

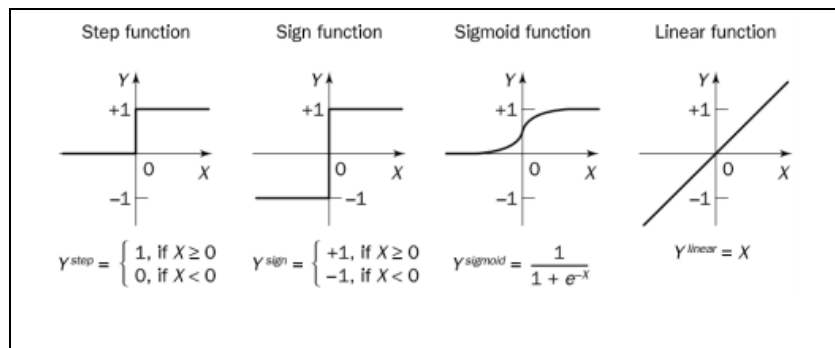
Jaringan Syaraf Tiruan adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut (Andrijasa, 2010). Jaringan Syaraf Tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi sebagai berikut :

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron.
2. Isyarat mengalir diantara sel syaraf (*neuron*) melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian.
4. Setiap sel syaraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya (Jumarwanto , 2009).

Komponen utama pada jaringan syaraf tiruan adalah sebagai berikut :

1. *Neuron* atau *node* merupakan tempat untuk memproses informasi. Setiap *neuron* akan menerima input, memproses input lalu menghasilkan sebuah output (Purnamasari, 2013).
2. Bobot atau *weight* merupakan nilai yang merepresentasikan koneksi antar *neuron* (Purnamasari, 2013). Pada setiap penghubung akan dilakukan operasi perkalian bobot dengan sinyal yang melewati penghubung tersebut.
3. Fungsi Aktivasi adalah fungsi yang menentukan output dari suatu neuron berdasarkan sinyal masukan yang diterima. Setiap *neuron* akan

menetapkan fungsi aktivasi (Wicaksono, 2008). Ada empat jenis fungsi aktivasi yang secara umum digunakan yaitu : *step function*, *sign function*, *sigmoid function* dan *linear function*. Gambar ke empat fungsi aktivasi dapat dilihat pada Gambar 2.7.



Gambar 2.7. Fungsi Aktivasi *Neuron* (Negnevitsky, 2005)

Step function dan *sign function* disebut sebagai fungsi pembatasan kasar yang digunakan secara umum pada permasalahan klasifikasi dan pengenalan pola. *Sigmoid function* digunakan pada jaringan propagasi balik dan dapat mengubah input yang memiliki jangkauan nilai $[-\infty, \infty]$ menjadi output dengan jangkauan nilai $[0.0, 1.0]$. *Linear activation function* digunakan pada pendekatan linear dan dapat menghasilkan output yang sama dengan input yang diterima oleh *neuron*.

4. Layer merupakan lapisan pada JST. Pada arsitektur JST terdapat dua tipe layer yaitu jaringan layer tunggal dan jaringan layer jamak. Jaringan layer tunggal terdiri dari lapisan input dan output. Sementara jaringan layer jamak terdiri dari lapisan input, output dan lapisan tersembunyi yang terletak di antara lapisan input dan output.

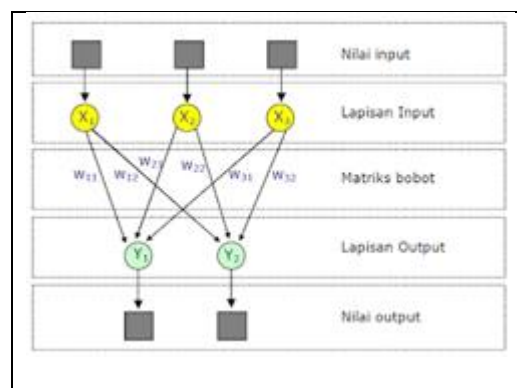
JST memiliki metode pembelajaran untuk memproses input yaitu *supervised learning* (memiliki target) dan *unsupervised learning* (tidak memiliki target). Pada pembelajaran *supervised learning*, target akan ditentukan dan kemudian nilai input dan output akan dilatih hingga nilai error (selisih output dan target) dapat seminimal mungkin (Wicaksono, 2008).

2.2.3 Arsitektur Jaringan Syaraf Tiruan (JST)

Sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain (Sukma Puspitorini, 2012) :

1. Jaringan Layer Tunggal (*Single Layer Network*)

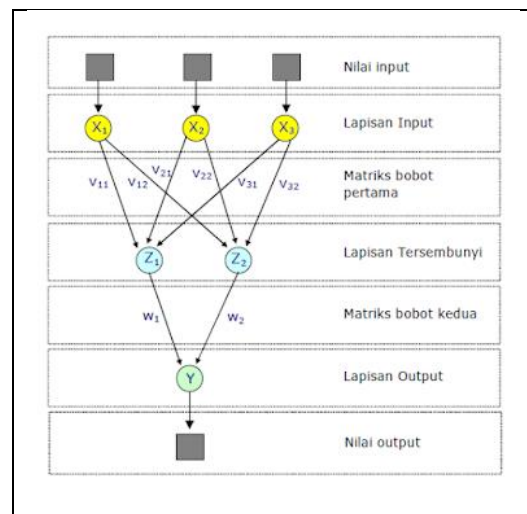
Jaringan dengan lapisan tunggal terdiri dari 1 layer input dan 1 layer output. Setiap neuron/unit yang terdapat di dalam lapisan/layer input selalu terhubung dengan setiap *neuron* yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi. Contoh algoritma JST yang menggunakan metode ini yaitu : *ADALINE*, *Hopfield*, *Perceptron*.



Gambar 2.8 Arsitektur Jaringan Syaraf Tiruan Layer Tunggal

2. Jaringan Banyak Lapisan (*Multilayer Net*)

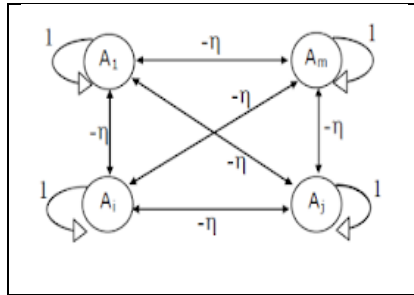
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis layer yakni layer input, layer output, dan juga layer tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma Jaringan Syaraf Tiruan yang menggunakan metode ini yaitu : *MADALINE*, *backpropagation*, *Neocognitron*.



Gambar 2.9 Arsitektur Jaringan Syaraf Tiruan *Multi Layer*

3. Jaringan Lapisan Kompetitif (*Competitive Layer*)

Pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah LVQ.

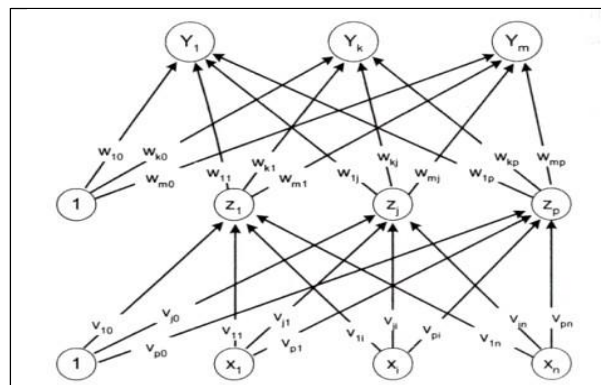


Gambar 2.10 Arsitektur Lapisan Kompetitif

2.2.4 Backpropagation

Backpropagation terdiri atas tiga lapisan atau lebih unit pengolah yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Ketiga lapisan ini saling berhubungan secara penuh (Astuti, dkk 2015).

Model *neuron* jaringan syaraf perambatan galat mundur diperlihatkan pada gambar 2.11 berikut ini:



Gambar 2.11 Arsitektur *Backpropagation* (Haryoko, 2016)

Fungsi Aktivasi

Karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan *backpropagation* antara lain harus kontinu, terdiferensialkan, dan tidak menurun secara monotonis (*monotonically non-decreasing*). Lebih lanjut, untuk

efisiensi komputasi, turunan fungsi tersebut mudah didapatkan dan nilai turunannya dapat dinyatakan dengan fungsi aktivasi itu sendiri (Suhardi, 2007).

Fungsi aktivasi pertama adalah *sigmoid* biner atau *sigmoid logistic*, yang memiliki jangkauan nilai [0,1]

$$f(x) = \frac{1}{1+exp^{-\sigma x}} \dots\dots\dots (2.1)$$

Dengan

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)] \dots\dots\dots (2.2)$$

Algoritma *Backpropagation*

Sedangkan algoritma pelatihan pada jaringan syaraf tiruan propagasi balik adalah sebagai berikut:

- 1) Langkah 0 : Inisialisasi bobot-bobot dengan bilangan nilai acak kecil.
- 2) Langkah 1 : Selama kondisi berhenti salah, kerjakan langkah 2 sampai langkah 9.
- 3) Langkah 2 : Untuk setiap pasangan pelatihan, kerjakan langkah 3 sampai langkah 8.

Umpan maju (*Feedforward*)

- 4) Langkah 3 : Tiap-tiap unit masukan ($X_i, i=1, \dots, n$) menerima isyarat masukan X_i dan dilanjutkan ke unit-unit tersembunyi.
- 5) Langkah 4 : Tiap unit tersembunyi ($Z_j, j = 1, \dots, p$) menjumlahkan bobot sinyal *input*.

$$z_{in_{jk}} = v_{oj} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2.3)$$

Dengan menerapkan fungsi aktivasi hitung :

$$z_j = f(z - in_j) \dots\dots\dots (2.4)$$

Dan mengirimkan isyarat ini kesemua unit pada unit keluaran.

- 6) Langkah 5 : Tiap unit keluaran (y_k , $k = 1, \dots, m$) menjumlahkan isyarat masukan berbobot.

$$y_in_k = w_{ok} + \sum_{k=1}^p z_j w_{jk} \dots\dots\dots (2.5)$$

Dengan menetapkan fungsi aktivasi hitung,

$$y_j = f(y - in_k) \dots\dots\dots (2.6)$$

- 7) Langkah 6 : Tiap unit keluaran (y_k , $k = 1, \dots, m$) menerima pola pelatihan masukannya. Hitung galat informasi :

$$\delta_k = (t_k - y_k) f'(y(in_k)) \dots\dots\dots (2.7)$$

Hitung koreksi bobot dan biasnya :

$$\Delta w_{jk} = \sigma \delta_k x_j$$

$$\Delta w_{ok} = \alpha \delta_k (\dots\dots\dots (2.8)$$

- 8) Langkah 7 : Tiap unit tersembunyi ($z_{j,j} = 1, \dots, p$) menjumlahkan delta masukannya (dari unit-unit yang berada pada lapisan di atasnya).

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk} \dots\dots\dots (2.9)$$

Hitung galat informasinya :

$$\delta_j = \delta_in_j f(x_in_j) \dots\dots\dots (2.10)$$

Hitung koreksi bobot dan biasnya :

$$\Delta v_{ij} = \alpha \delta_j x_i \dots\dots\dots (2.11)$$

Perbaiki bobot dan bias

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

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots (2.12)$$

Tiap unit tersembunyi (z_j , $j = 1, \dots, p$) memperbarui bobot dan bias ($i = 0, 1, \dots, n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots\dots\dots (2.13)$$

- 10) Langkah 9 : Uji syarat berhenti.

2.2.5 Pendeteksian Tepi (*Edge Detection*)

Pada proses deteksi tepi metode *Canny* adalah salah satu algoritma deteksi tepi *modern*. deteksi tepi *Canny* ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia.

Ada beberapa kriteria pendeteksian tepian paling optimum yang dapat dipenuhi oleh algoritma *Canny* (*Canny, 1986*):

- a. Mendeteksi dengan baik (kriteria deteksi)

Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.

b. Melokalisasi dengan baik (kriteria lokalisasi)

Dengan *Canny* dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

c. Respon yang jelas (kriteria respon)

Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya.

Pemilihan parameter deteksi tepi *Canny* sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut antara lain :

1. Nilai Standart Deviasi Gaussian
2. Nilai Ambang

Langkah-langkah dalam proses deteksi tepi dengan metode *canny* sebagai berikut :

1. pertama-tama dilakukan penghalusan (*smoothing*) citra untuk menghilangkan *noise*. contohnya menggunakan filtering dengan *Gaussian Filter*.
2. Selanjutnya dicari *gradient magnitude* citra untuk melihat daerah-daerah yang memiliki turunan spasial yang tinggi.
3. Ditentukan arah dari tepi dengan menggunakan *invers* tangen dari gradient magnitude Y (Gy) dibagi *gradient magnitude* X (Gx). Arah yang diperoleh dari perhitungan ini kemudian dipetakan ke 0, 45, 90 atau 135 derajat berdasarkan kedekatannya dengan keempat derajat arah tadi.
4. kemudian dilakukan *Non Maximum Suppression*. yaitu, penghilangan nilai-nilai yang tidak maksimum.

5. Selanjutnya dilakukan *Hysteresis*. *Hysteresis* menggunakan dua threshold T1 (*threshold* bawah) dan T2 (*threshold* atas). Bila *magnitude* ada dibawah T1, titik tersebut di-set nol (dijadikan non-tepi). Bila *magnitude* ada diantara T1 dan T2, di-set nol kecuali jika ada jalan (*path*) dari titik tersebut ke titik yang memiliki *magnitude* di atas T2.

2.2.6 Confusion matrix

Berdasarkan *confusion matrix* dapat dilakukan analisis menggunakan 4 parameter yaitu *true positive* (TP), *false positive* (FP), *true negative* (TN) dan *false negative* (FN). Yang digunakan untuk menghitung sensitivitas (*sensitivity*), spesifisitas (*specificity*), *positive predicate value* (PPV) dan *negative predicate value* (NPV). Perhitungan nilai-nilai tersebut menggunakan persamaan (2.14), (2.15), (2.16), (2.17) (Tomar dkk, 2012).

$$Sensitivity = \left[\frac{TP}{TP+FN} \right] * 100\% \dots\dots\dots (2.14)$$

$$Specificity = \left[\frac{TN}{TN+FP} \right] * 100\% \dots\dots\dots (2.15)$$

$$PPV = \left[\frac{TP}{TP+FP} \right] * 100\% \dots\dots\dots (2.16)$$

$$NPV = \left[\frac{TN}{TN+FN} \right] * 100\% \dots\dots\dots (2.17)$$

Dari Han dan Kamber (2006), *confusion matrix* merupakan cara yang berguna untuk menganalisis seberapa baik sistem mengenali *tuple* dari kelas yang berbeda. TP dan TN digunakan ketika sistem benar, sedangkan FP dan FN memberitahu ketika sistem salah. Nilai Sensitivitas dan spesifisitas yang nantinya

digunakan untuk mengklasifikasikan akurasi. sensitivitas sebagai *true positives (recognition) rate* (proporsi dari *tuple* positif yang diidentifikasi dengan benar). Sedangkan spesifisitas adalah *true negatives rate* (proporsi *tuple* negatif yang diidentifikasi secara benar). Dengan menggunakan sensitivitas dan spesifisitas dapat menganalisis tingkat akurasi menggunakan persamaan (2.18) dan ukuran tingkat kesalahan sistem menggunakan persamaan (2.19).

$$Accuracy = SV \left(\frac{P}{(P+N)} \right) + SF \left(\frac{N}{(P+N)} \right) \dots\dots\dots (2.18)$$

$$Error Rate = \frac{(FP+FN)}{(P+N)} * 100\% \dots\dots\dots (2.19)$$

dimana:

SV : Nilai sensitivitas

SF : Nilai spesifisitas

P : Jumlah total kasus dengan hasil positif

N : Jumlah total kasus dengan hasil negatif

Dengan menggunakan perhitungan pada *confusion matrix* sistem dinilai benar dengan *true positive* (TP) dan *false positive* (FP) jika sistem memberi nilai akurasi hasil pengujian dengan minimal *threshold*. Nilai yang digunakan dianggap positif jika mencapai 0,8 atau 80% (Faizal dan Hamdani,2018).