

BAB 2

DASAR TEORI

Pada bagian ini, akan dibahas mengenai referensi-referensi terdahulu yang relevan dengan topik penelitian tepatnya pada bagian kajian pustaka. Pada bagian dasar teori, akan dibahas mengenai teori-teori yang relevan dengan topik penelitian. Teori meliputi anatomi mata manusia, pemaparan mengenai fundus mata, *Diabetic Retinopathy* (DR), pengolahan citra, *3D Gray-Level Co-occurrence Matrix* (3D GLCM), Jaringan Saraf Tiruan (JST)propagasi balik, *confusion matrix*, dan *hold-out validation*.

2.1 KAJIAN PUSTAKA

Penelitian mengenai deteksi penyakit DR serupa telah dilakukan oleh beberapa peneliti. Pada penelitian [3], deteksi dini penyakit DR dilakukan dengan pengolahan citra berbasis algoritma deteksi *microaneurysm* dengan morfologi matematika. Algoritma terbagi menjadi tiga tahap, yaitu *preprocessing*, menentukan kandidat *microaneurysm*, dan *postprocessing*. Sistem dibuat menggunakan *raspberrypi* sebagai medianya. Dari hasil pengujian didapatkan tingkat akurasi deteksi sistem mencapai 70.5%. Pada penelitian tersebut, tingkat akurasi sistem masih tergolong rendah. Hal ini mengindikasikan bahwa sistem belum mencapai performansi yang optimal.

Pada penelitian [4], pengklasifikasian penderita DR dilakukan dengan menggunakan metode JST *Multi-Layered Perceptron*. Untuk klasifikasi, digunakan juga metode *Levenberg-Marquardt* (LM) dan *Bayesian Regularization* (BR). Hasil penelitian tersebut mendapatkan nilai tingkat akurasi sebesar 67.47% untuk proses pengujian data. Tetapi, tingkat akurasi tersebut masih tergolong rendah. Selain itu, pada penelitian tersebut tidak digunakan metode evaluasi performansi sistem klasifikasi.

Pada penelitian [5], deteksi penyakit DR dilakukan dengan model komputasi untuk mengolah piksel citra retina. Digunakan metode *Local Binary*

Patern (LBP) untuk ekstraksi ciri dan *Learning Vector Quantization* (LVQ) untuk proses klasifikasi. Dari hasil pengujian didapatkan tingkat akurasi sebesar 85%.

Selain itu, metode klasifikasi LVQ juga digunakan oleh penelitian [6]. Pada hasil pengujian, didapatkan tingkat akurasi deteksi sebesar 43.75%. Tingkat akurasi pada penelitian tersebut sangat kecil. Selain itu, penelitian tersebut tidak menggunakan metode evaluasi performansi untuk sistem yang telah dibuat.

Pada penelitian [7], dilakukan pengenalan biometrik berbasis citra iris dengan metode 3D GLCM. Pada penelitian tersebut juga dilakukan analisis komparasi performansi antara metode 3D GLCM dengan metode 2D GLCM. Hasil analisis diperoleh tingkat akurasi sistem sebesar 87.22% untuk metode 3D GLCM. Tetapi, pada penelitian tersebut tidak digunakan metode evaluasi performansi sistem klasifikasi.

Pada penelitian[8], deteksi DR dilakukan dengan bantuan *Graphical User Interface* (GUI). Pada sistem tersebut, GUI akan menampilkan tingkat keparahan penderita beserta dengan tindakan yang harus dilakukan oleh penderita. Sementara itu, pada [9] telah dibuat sistem untuk mengetahui tingkat keparahan penyakit DR dengan metode *Support Vector Machine* (SVM). Tingkat akurasi yang didapat sebesar 95.93%. Tetapi, pada penelitian tersebut masih digunakan metode ekstraksi ciri GLCM terdahulu, yaitu 2D GLCM.

Pada penelitian [10], dilakukan deteksi katarak dengan metode ekstraksi ciri 3D GLCM dan metode klasifikasi JST Propagasi Balik. Pada sistem tersebut, diklasifikasikan tingkat keparahan katarak ke dalam tiga kelas, yaitu normal, *immature*, dan *mature*. Tingkat akurasi yang didapatkan yaitu 91.11% pada sudut orientasi 0° dan 45° dengan jarak piksel bernilai 1. Tetapi, pada penelitian ini hanya menggunakan 9 arah *offset* untuk ekstraksi ciri 3D GLCM. Padahal, arah *offset* pada 3D GLCM berjumlah 13 arah *offset*.

Berikut merupakan ringkasan perbandingan penelitian terdahulu dengan penelitian kali ini.

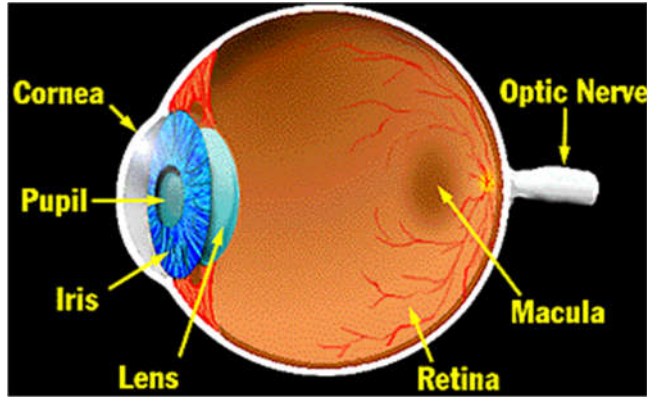
Tabel 2. 1 Perbandingan Penelitian

Pengarang	Metode	Tahun	<i>Purpose Method</i>
Lukman Heryawan [3].	Morfologi matematika dengan penggunaan <i>rasbery pi</i> .	2017	Hanya menggunakan <i>software</i> dan tidak menggunakan <i>hardware</i> .
Nor Hazlyna Harun, Yuhanis Yusof, Faridah Hassan, dan Zunaina Embong [4].	<i>JST Multi-Layered Perceptron</i> , <i>Levenberg-Marquardt (LM)</i> , dan <i>Bayesian Regularization (BR)</i> .	2019	Ekstraksi ciri 3D GLCM, klasifikasi JST propagasi balik, <i>confusion matrix</i> , dan <i>hold-out validation</i> .
Wen-Shiung Chen, Ren-Hung Huang, dan Lili Hsieh [7].	Ekstraksi ciri 3D GLCM.	2009	Ekstraksi ciri 3D GLCM, klasifikasi JST propagasi balik, <i>confusion matrix</i> , dan <i>hold-out validation</i> .
Taufiq Galang Adi Putranto dan Ika Candradewi [9].	Ekstraksi ciri 2D GLCM dan klasifikasi SVM.	2018	Ekstraksi ciri 3D GLCM dan klasifikasi JST propagasi balik.
Ichvanul Yulizar Putra, Rita Magdalena, dan Yunendah Nur Fu'adah [10].	Ekstraksi ciri 3D GLCM dan klasifikasi JST propagasi balik.	2019	Ekstraksi ciri 3D GLCM dan klasifikasi JST propagasi balik, <i>confusion matrix</i> , dan <i>hold-out</i>

			<i>validation.</i>
--	--	--	--------------------

2.2 DASAR TEORI

2.2.1 Mata



Gambar 2. 1AnatomiMata Manusia[11]

Mata adalah organ utama penglihatan yang merupakan salah satu sistem sensorik luar biasa. Manusia mengumpulkan sebagian besar informasi tentang lingkungan eksternal melalui mata mereka[12]. Dengan demikian, manusia lebih mengandalkan penglihatan dari pada indra lain. Selain berfungsi sebagai jendela dunia, mata berfungsi sebagai jendela identitas seseorang. Hal ini menawarkan fitur unik untuk penerapan teknologi identifikasi. Baik teksturnya yang sangat detail maupun pola pembuluh darah fundus yang unik setiap orangnya, fakta-fakta ini yang kemudian cocok untuk pengenalan secara biometrik. Kedua bola mata terletak di orbit dan menempati satu per lima bagian orbit tersebut. Sisa dari volume orbit ditempati oleh otot-otot ekstraokular, fascia, lemak, pembuluh darah, saraf, dan kelenjar lakrimal. Mata secara embriologis merupakan perpanjangan dari sistem saraf pusat. Sistem ini berbagi banyak sifat anatomi dan fisiologis yang umum dengan otak. Mata dan otak dilindungi oleh dinding tulang, memiliki penutup fibrosa yang kuat, dan suplai darah ganda ke lapisan saraf esensial di retina. Mata dan otak memiliki rongga internal yang diperfusi oleh cairan dengan komposisi yang sama dan di bawah tekanan yang setara[13].

Mata manusia terdiri dari beberapa bagian. Bagian-bagian tersebut diantaranya adalah sebagai berikut[14].

1) Kornea

Kornea adalah bagian luar yang jernih dari sistem pemfokusan mata yang terletak di bagian depan mata.

2) Iris

Iris adalah bagian berwarna dari mata yang mengelilingi pupil. Iris berperan dalam mengatur jumlah cahaya yang masuk ke mata.

3) Lensa

Lensa adalah bagian mata yang bening, terletak di belakang iris, dan berperan dalam membantu memfokuskan cahaya dan gambar pada retina.

4) Makula

Makula adalah area retina kecil dan sensitif yang memberikan penglihatan sentral. Makula terletak di pusat retina.

5) Saraf optik

Saraf optik adalah saraf sensorik terbesar pada mata. Saraf optik berperan dalam membawa impuls untuk dilihat dari retina menuju otak.

6) Pupil

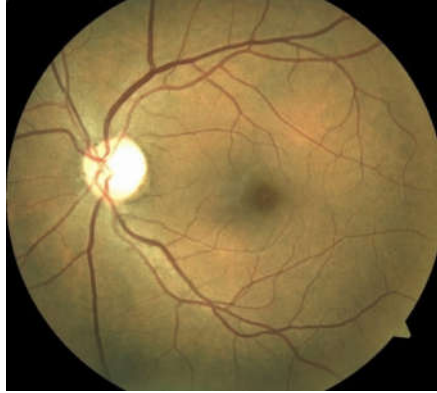
Pupil adalah lubang di tengah iris yang dilalui cahaya. Iris menyesuaikan ukuran pupil untuk mengontrol jumlah cahaya yang masuk ke mata.

7) Retina

Retina adalah jaringan peka cahaya di bagian belakang mata. Retina mengubah cahaya menjadi impuls listrik yang dikirim ke otak melalui saraf optik.

8) Cairan *vitreous*

Cairan *vitreous* adalah massa transparan dan tidak berwarna yang mengisi dua per tiga bagian belakang bola mata, antara lensa dan retina.

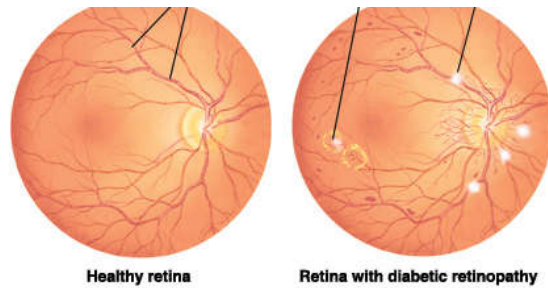


Gambar 2. 2 Fundus Mata[15]

Fundus adalah bagian dalam, permukaan belakang mata dan terdiri dari retina, makula, *optic disc*, fovea, dan pembuluh darah. “Fundus” adalah kata Latin untuk “bagian bawah”. Dalam kedokteran, fundus mengacu pada bagian bawah atau pangkal organ [16]. Dengan fotografi fundus, kamera fundus khusus diarahkan ke bagian belakang mata dan mengambil gambar. Gambar-gambar ini akan membantu dokter mata untuk menemukan dan mengobati penyakit.

2.2.2 *Diabetic Retinopathy (DR)*

Diabetic Retinopathy (DR) merupakan salah satu komplikasi penyakit diabetes melitus. Penyakit ini diindikasikan dengan kadar gula darah yang tinggi sehingga menyebabkan kerusakan pada pembuluh darah retina mata. Penyakit DR dapat diderita oleh penderita diabetes melitus tipe 1 maupun tipe 2. Kadar gula darah yang terlalu banyak menyebabkan penyumbatan dan terganggunya mikrosirkulasi pembuluh darah retina, sehingga retina menjadi kekurangan oksigen dan nutrisi dalam darah. Akibatnya retina membentuk pembuluh darah baru untuk memenuhi kebutuhan darah. Namun dalam proses pembentukkannya tidak dapat berkembang dengan sempurna, pembuluh darah tersebut rentan pecah dan rusak. Hal ini mengakibatkan penumpukan cairan yang mengandung lemak protein hingga pendarahan pada retina. Terjadinya pendarahan secara terus-menerus dapat mengakibatkan kebutaan [17].



Gambar 2. 3 Perbandingan Retina Sehat dan Penderita DR [18]

Secara umum DR terbagi dalam dua jenis, yaitu *Non-Proliferative Diabetic Retinopathy* (NPDR) dan *Proliferative Diabetic Retinopathy* (PDR). Pada NPDR, gejala awal ditandai dengan adanya kapiler yang membentuk kantung-kantung kecil menonjol seperti titik-titik berwarna merah yang disebut mikroaneurisma dan munculnya eksudat berupa kebocoran kapiler disekitar retina, terlihat seperti bercak-bercak kuning atau putih yang didalamnya terdapat protein dan lipid. NPDR terbagi menjadi *Mild* NPDR, *Moderate* NPDR, dan *Severe* NPDR. Sementara itu, PDR merupakan komplikasi mata yang paling parah pada penderita diabetes melitus. PDR ditandai dengan rusaknya sebagian besar pembuluh darah retina dan terjadinya iskemia retina. Hal ini nantinya akan merangsang pembentukan pembuluh darah baru yang tidak normal dan rapuh disertai dengan pendarahan.

2.2.3 Citra dan Pengolahan Citra

Citra atau *image* adalah fungsi dari intensitas cahaya yang direpresentasikan pada bidang dua dimensi $f(x,y)$ [10]. Pada fungsi tersebut, x dan y merupakan koordinat spasial dan f merupakan amplitudo (misalnya kecerahan). Jika x dan y berhingga (*finite*) dan diskret, maka fungsi tersebut merepresentasikan citra digital. Citra digital terdiri dari sejumlah elemen berhingga yang masing-masing mempunyai lokasi dan nilai, setiap elemen x dan y disebut sebagai elemen citra atau *peel* (piksel). Citra digital digambarkan oleh sekumpulan *array* dua dimensi dan setiap *array* menyajikan satu kanal warna. Setiap elemen *array* disebut dengan piksel. Dimensi citra umumnya dinyatakan dalam format panjang x tinggi, misalnya 415x239 piksel. Secara matematis,

definisi citra dapat dituliskan kedalam bentuk matriks dengan x menyatakan baris dan y menyatakan kolom seperti pada Persamaan 2.1 berikut ini[10].

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N - 1) \\ f(1,0) & f(1,1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M - 1,0) & f(M - 1,1) & \cdots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Pada persamaan di atas, M merupakan jumlah baris dan N merupakan jumlah kolom. Indeks baris x dan indeks kolom y menyatakan suatu koordinat piksel citra.

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga citra tersebut adalah sebagai berikut.

1) Citra warna



Gambar 2. 4 Representasi Citra RGB[19]

Citra warna atau sering disebut dengan citra RGB merupakan jenis citra yang menyajikan warna dalam bentuk komponen *Red*/Merah (R), *Green*/Hijau (G), dan *Blue*/Biru (B) pada setiap pikselnya [19]. Setiap komponen warna terdiri dari 8bit dengan nilai berkisar antara 0-255.

2) Citra skala keabuan



Gambar 2. 5 Representasi Citra *Grayscale*[19]

Citra skala keabuan atau sering disebut dengan citra *grayscale* merupakan jenis citra yang merepresentasikan gradasi warna hitam dan putih sehingga menghasilkan efek warna abu-abu[19]. Intensitas warna pada citra ini berkisar antara 0-255, dengan nilai 0 menyatakan warna hitam dan nilai 255 menyatakan warna putih. Citra ini memiliki kedalaman piksel sebesar 8bit [0, 255], namun ada juga yang bernilai 16bit [0, 65535] untuk penggunaan yang memerlukan ketelitian lebih tinggi.

3) Citra biner



Gambar 2. 6 Representasi Citra Biner[19]

Citra biner merupakan jenis citra yang hanya mempunyai satu nilai pada setiap pikselnya, yaitu 0 (hitam) atau 1 (putih)[19]. Citra jenis ini hanya terdiri

dari 1bit dan biasanya digunakan untuk mendeteksi tepian objek seperti teks, sidik jari, maupun benda.

Pengolahan citra merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra dengan bantuan komputer. Pengolahan citra adalah suatu upaya untuk memperbaiki kualitas citra agar didapatkan kualitas citra yang lebih baik. Nantinya, citra tersebut akan lebih mudah diinterpretasikan oleh manusia. Beberapa alasan mengapa perlu dilakukan pengolahan citra, antara lain[10]:

- 1) Untuk mendapatkan citra asli dari suatu citra yang mengalami degradasi akibat pengaruh derau.
- 2) Untuk memperoleh citra dengan karakteristik tertentu dan cocok secara visual yang dibutuhkan untuk tahap lebih lanjut dalam pemrosesan citra.

Secara umum, tahapan pengolahan citra meliputi:

- 1) Akuisisi citra menggunakan kamera analog, kamera digital, *handycam*, *scanner*, *optical reader*, dan sebagainya.
- 2) Peningkatan kualitas citra (pra-pemrosesan).
- 3) Segmentasi citra.
- 4) Representasi dan uraian.
- 5) Pengenalan dan interpretasi.

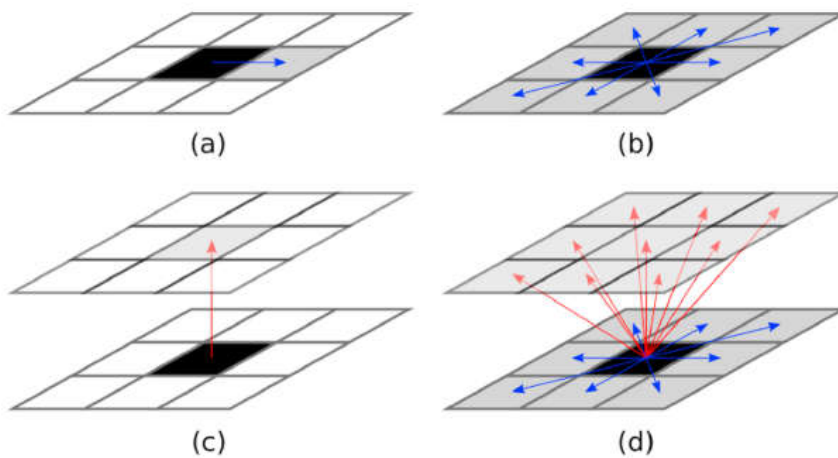
2.2.4 Ekstraksi Ciri 3D Gray-Level Co-occurrence Matrix (3D GLCM)

Ekstraksi ciri merupakan cara yang ditempuh untuk mengetahui ciri atau informasi dari suatu citra yang akan diidentifikasi. Ciri atau informasi tersebut nantinya akan digunakan sebagai parameter pembeda antara objek yang satu dengan yang lainnya pada tahapan klasifikasi.

Ekstraksi ciri 3D GLCM pertama kali dikenalkan oleh Haralick pada tahun 1973. Ekstraksi ciri 3D GLCM merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi. Metode ini melakukan ekstraksi ciri berbasis statistikal menggunakan distribusi derajat keabuan dari hubungan ketetanggaan antarpiksel di dalam citra.

Misalkan sekumpulan citra N_z (layer) yang akan dianalisis berbentuk persegi panjang (*rectangular*), dengan piksel N_x pada arah horizontal dan piksel

N_y pada arah vertikal. Selain itu, nilai digital setiap piksel dikuantisasi menjadi tingkat abu-abu N_g . Misalkan $L_x = \{0, 1, \dots, N_x - 1\}$, $L_y = \{0, 1, \dots, N_y - 1\}$, dan $L_z = \{0, 1, \dots, N_z - 1\}$ menjadi domain X, Y , dan Z . Selanjutnya, misalkan $G = \{0, 1, \dots, N_g - 1\}$ menjadi himpunan tingkat abu-abu N_g . Himpunan $L_x \times L_y \times L_z$ adalah himpunan citra yang diurutkan berdasarkan penunjukkan kolom-baris-lapisannya. Citra I dapat direpresentasikan sebagai fungsi yang memberikan beberapa nilai digital dalam G untuk setiap piksel di $L_x \times L_y \times L_z$ ($I: L_x \times L_y \times L_z \rightarrow G$)[20].



Gambar 2. 7 Relasi Spasial Piksel (a) $d=1$ dan Sudut 0 ; (b) Piksel Tetangga 2D GLCM Untuk $d=1$; (c) $d=[0, 0, 1]$; (d) Piksel Tetangga 3D GLCM Untuk $d=1$ [20]

Dalam definisi asli ko-okurensi menurut Haralick, sepasang piksel diidentifikasi oleh posisi relatif satu piksel, yang selanjutnya disebut piksel tetangga, terhadap piksel lainnya, yang selanjutnya disebut piksel referensi[20]. Jarak relatif ditentukan oleh skalar d dan sudut orientasi α . Pada Gambar 2.7, piksel referensi ditentukan dengan warna hitam dan piksel tetangga berwarna abu-abu. Gambar 2.7(a) menunjukkan turunan dari sepasang piksel dengan $d = 1$ dan $\alpha = 0$. Sementara itu, Gambar 2.7(b) menunjukkan semua kemungkinan piksel tetangga (0, 45, 90, 135) pada 2D GLCM. Pada 3D GLCM, jarak relatif ditentukan oleh sebuah vektor $d = [d_1, d_2, d_3]$ yang menunjuk dari piksel referensi menuju piksel tetangga. Gambar 2.7(c) menggambarkan sepasang piksel yang dipisahkan oleh $d = [0, 0, 1]$ dan Gambar 2.7(d) menunjukkan semua piksel tetangga terdekat untuk ruang yang terdiri dari dua gambar.

Konsep dasar perhitungan 3D GLCM hampir sama dengan 2D GLCM. Pada 3D GLCM, matriks ko-okurensi berisi kumpulan jumlah piksel pasangan dengan indeks i dan j . Matriks ko-okurensi didefinisikan dengan menentukan perpindahan $d = (dx, dy, dz)$, dimana dx , dy , dan dz adalah piksel yang bergerak pada sumbu x , y , dan z pada citra tiga dimensi. Dengan kata lain, arah *offset* menjadi 13 arah dengan nilai jarak (D) yaitu 1 [21][20]. Perolehan ciri diperoleh dari nilai piksel matriks yang mempunyai nilai tertentu dan membentuk suatu sudut pola/*offset*. Sudut yang dibentuk dari nilai piksel citra menggunakan 3D GLCM adalah seperti yang tersaji pada Tabel 2.1 di bawah ini.

Tabel 2. 2 Sudut Pola/Offset Pada 3D GLCM

Arah Offset	Horizontal	Vertikal
[0 1 0]	0°	0°
[-1 1 0]	45°	0°
[-1 0 0]	90°	0°
[-1 -1 0]	135°	0°
[0 1 -1]	0°	45°
[0 0 -1]	-	90°
[0 -1 -1]	0°	135°
[-1 0 -1]	90°	45°
[1 0 -1]	90°	135°
[-1 1 -1]	45°	45°
[1 -1 -1]	45°	135°
[-1 -1 -1]	135°	45°
[1 1 -1]	135°	135°

Ciri statistik yang umum digunakan pada 3D GLCM adalah[7]:

1) *Energy*

Energy atau sering disebut juga *Angular Second Moment* (ASM) adalah kekonstanan tingkat keabuan citra atau dengan kata lain kondisi pada saat piksel citra sangat mirip. Untuk menghitung *energy*, digunakan rumus:

$$Energy = \sum_i \sum_j \sum_k [p(i, j, k)]^2 \quad (2.2)$$

Pada persamaan di atas, $p(i, j, k)$ menunjukkan masukan dalam matriks ko-okurensi.

2) *Image Pixel Correlation*

Image Pixel Correlation merepresentasikan kawasan linier dari derajat keabuan. *Image Pixel Correlation* menunjukkan bagaimana piksel yang menjadi referensi dihubungkan dengan tetangganya. Untuk menghitung *correlation*, digunakan rumus:

$$Correlation = \sum_i \sum_j \sum_k p(i, j, k) \left[\frac{(i-\mu_i)(j-\mu_j)(k-\mu_k)}{\sigma_i \sigma_j \sigma_k} \right] \quad (2.3)$$

3) *Contrast*

Contrast adalah perbedaan tingkat warna atau skala keabuan pada sebuah citra. Untuk menghitung *contrast*, digunakan rumus:

$$Contrast = \sum_i \sum_j \sum_k p(i, j, k) [(i-j)^2 + (i-k)^2 + (j-k)^2] \quad (2.4)$$

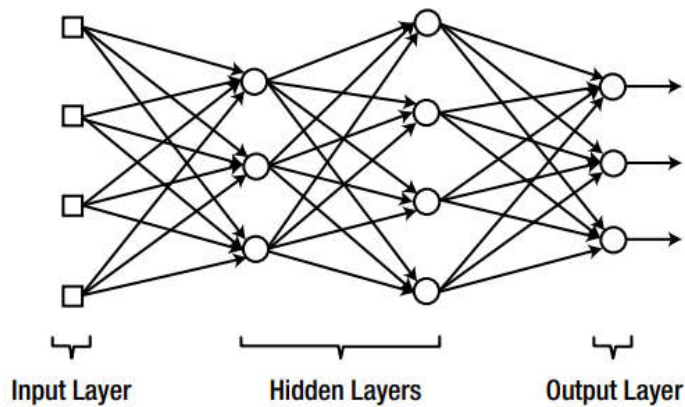
4) Homogeneity

Homogeneity atau sering disebut dengan *Inverse Difference Momentum* (IDM) merupakan ciri statistik yang merepresentasikan homogenitas atau ukuran kesamaan pada sebuah citra digital. Nilai *homogeneity* akan tinggi ketika semua elemen pada sebuah citra sama. Untuk menghitung *homogeneity*, digunakan rumus:

$$Homogeneity = \sum_i \sum_j \sum_k \frac{p(i,j,k)}{1+[(i-j)^2+(i-k)^2+(j-k)^2]} \quad (2.5)$$

Pada persamaan di atas, i menunjukkan baris ke- i , j menunjukkan kolom ke- j , dan k menunjukkan arah piksel tetangga kedua pada 3D GLCM. Notasi μ_i menunjukkan nilai rata-rata elemen baris pada matriks $p(i,j,k)$, notasi μ_j menunjukkan nilai rata-rata elemen kolom pada matriks $p(i,j,k)$, dan notasi μ_k menunjukkan nilai rata-rata elemen pada arah piksel tetangga kedua pada matriks $p(i,j,k)$. Notasi σ_i menunjukkan nilai standar deviasi elemen baris pada matriks $p(i,j,k)$, notasi σ_j menunjukkan nilai standar deviasi elemen kolom pada matriks $p(i,j,k)$, dan notasi σ_k menunjukkan nilai standar deviasi elemen pada arah piksel tetangga kedua pada matriks $p(i,j,k)$.

2.2.5 Klasifikasi Jaringan Saraf Tiruan (JST) Propagasi Balik



Gambar 2. 8 Ilustrasi JST[22]

Jaringan Saraf Tiruan (JST) pertama kali dikenalkan oleh McCulloch dan Pitts pada tahun 1943. Kemudian pada tahun 1958, Rosenblatt memperkenalkan model jaringan yang disebut perceptron. Model jaringan ini dikembangkan untuk memperoleh hasil optimal pada iterasinya. Namun, perceptron ini merupakan model jaringan lapisan tunggal atau *single layer*. Oleh karena itu, pada tahun 1986 Rumelhart mengembangkan perceptron menjadi propagasi balik/*backpropagation* yang memungkinkan pemrosesan menjadi beberapa lapisan/*layer* [10].

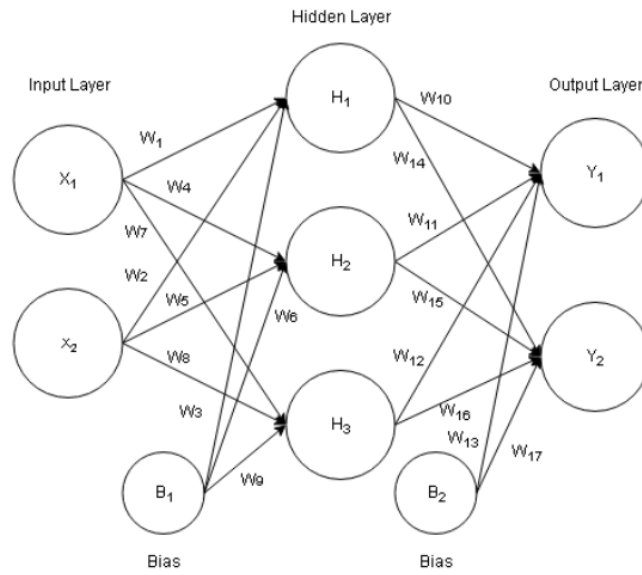
JST merupakan salah satu representasi buatan dari otak manusia dengan menggunakan sistem komputer yang tersusun atas *neuron* untuk saling terhubung. Setiap pola informasi *input* dan *output* yang diberikan kedalam JST diproses dalam *neuron*. Setiap *neuron* yang terkumpul di dalam lapisan disebut *neuron layers*. JST yang merupakan imitasi dari jaringan saraf biologis ditentukan oleh tiga hal, yaitu:

- 1) Pola hubungan antar-*neuron* (arsitektur jaringan).
- 2) Metode untuk menentukan bobot penghubung (menguatkan/melemahkan sinyal).
- 3) Fungsi aktivasi oleh *neuron* (untuk menentukan keluaran yang nantinya akan dibandingkan dengan suatu nilai ambang batas).

Lapisan-lapisan penyusun JST dibagi menjadi tiga, yaitu:

- 1) Lapisan masukan/*input layer*; lapisan yang menerima pola masukan data dari luar yang merepresentasikan suatu permasalahan.
- 2) Lapisan tersembunyi/*hidden layer*; lapisan yang keluarannya tidak bisa diamati secara langsung.
- 3) Lapisan keluaran/*output layer*; lapisan yang menjadi solusi JST terhadap permasalahan yang dimuat pada lapisan masukan.

Propagasi balik/*backpropagation* adalah salah satu metode pembelajaran yang dapat diimplementasikan pada JST. Propagasi balik merupakan sebuah metode sistematis untuk pelatihan lapisan jamak/*multilayer* JST.



Gambar 2. 9 Ilustrasi JST PropagasiBalik [10]

Gambar 2.9 merupakan ilustrasi dari arsitektur JST propagasi balik. Di dalam jaringan propagasi balik di atas terdiri dari lima bagian, yaitu lapisan masukan/*input layer*, lapisan tersembunyi/*hidden layer*, lapisan keluaran/*output layer*, bias, dan nilai bobot/*weight*. Lapisan masukan disimbolkan dengan X . Banyaknya lapisan ini tergantung dari banyaknya ciri statistik yang diproduksi dari hasil ekstraksi ciri. Lapisan tersembunyi disimbolkan dengan H . Sementara itu, lapisan keluaran disimbolkan dengan Y . Banyaknya lapisan keluaran ditentukan oleh banyaknya hasil klasifikasi yang diinginkan. Bias dilambangkan dengan B . Bias merupakan *node* khusus pada jaringan propagasi balik yang nilainya selalu satu. Bias digunakan untuk mempercepat proses pelatihan data karena bias merupakan faktor koreksi terhadap kecukupan-kecukupan variabel masukan yang telah ditetapkan sebelumnya. Nilai bobot disimbolkan dengan W . Nilai bobot inilah yang akan diperbarui terhadap kesalahan atau *error* yang terjadi karena selisih dari target dan keluaran pada jaringan propagasi balik.

Sinyal input dari luar akan dikalikan dengan bobot sebelum mencapai *node*. Setelah sinyal tertimbang dikumpulkan di *node*, nilai-nilai ini ditambahkan menjadi jumlah tertimbang. Jumlah tertimbang yang dimaksud dapat dihitung sebagai berikut[22].

$$V = WX + B \quad (2.6)$$

dengan W dan X dapat didefinisikan sebagai berikut.

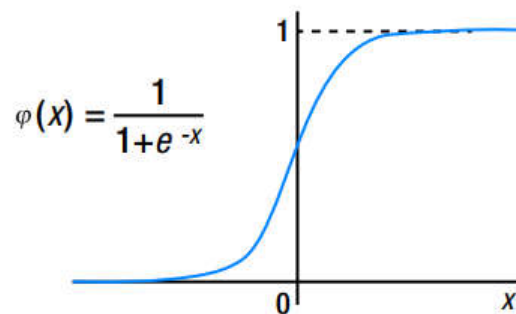
$$W = [W_1 \quad W_2 \quad \cdots \quad W_n] \quad (2.7)$$

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \quad (2.8)$$

Akhirnya, *node* memasukkan jumlah tertimbang ke dalam fungsi aktivasi dan menghasilkan keluaran. Fungsi aktivasi menentukan perilaku *node*. Keluaran didapatkan dengan memasukkan nilai jumlah tertimbang ke dalam fungsi aktivasi sebagai berikut[22].

$$Y = \varphi(V) \quad (2.9)$$

Simbol $\varphi(\cdot)$ pada persamaan di atas merupakan fungsi aktivasi. Ada berbagai jenis fungsi aktivasi yang sering digunakan pada JST, salah satunya adalah fungsi sigmoid. Fungsi sigmoid didefinisikan melalui Gambar 2.10 di bawah ini.

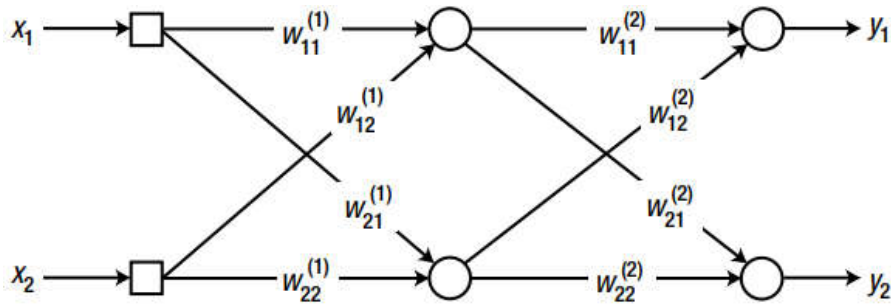


Gambar 2. 10 Fungsi Aktivasi Sigmoid[22]

Pada Gambar 2.10, terlihat bahwa fungsi sigmoid menghasilkan keluaran dengan nilai nol sampai satu. Karakteristik fungsi sigmoid seperti ini sangat berguna ketika JST yang dibuat mempunyai keluaran yang bersifat probabilitas.

Setiap unit yang berada di lapisan masukan terhubung dengan setiap unit yang ada di lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan keluaran. Jaringan ini terdiri dari banyak lapisan (*multilayer network*). Ketika jaringan diberi pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi yang selanjutnya diteruskan ke unit-unit lapisan keluaran. Kemudian unit-unit lapisan keluaran merespon sebagai keluaran JST. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (*backward*) pada lapisan tersembunyi kemudian menuju lapisan masukan. Proses *backward* tersebut akan terus dilakukan sampai memenuhi target *output* atau iterasi telah mencapai nilai maksimum.

Misalkan sebuah JST yang terdiri dari dua *node* masukan dan keluaran serta satu buah *hidden layer* yang juga terdiri dari dua *node*. Contoh JST yang dimaksud tersaji pada Gambar 2.11 dengan aksens *superscript* yang merupakan indikator *layer*.



Gambar 2. 11 Contoh Struktur JST[22]

Untuk mendapatkan *error* keluaran, dibutuhkan keluaran JST dari data masukan. Karena jaringan contoh memiliki satu *hidden layer*, maka diperlukan dua manipulasi data masukan sebelum penghitungan keluaran diproses. Pertama, jumlah tertimbang dari *hidden node* dihitung sebagai berikut[22].

$$\begin{bmatrix} V_1^{(1)} \\ V_2^{(1)} \end{bmatrix} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \triangleq W_1 X \quad (2.10)$$

Ketika nilai jumlah tertimbang dimasukkan ke dalam fungsi aktivasi, maka akan menghasilkan keluaran sebagai berikut.

$$\begin{bmatrix} Y_1^{(1)} \\ Y_2^{(1)} \end{bmatrix} = \begin{bmatrix} \varphi(V_1^{(1)}) \\ \varphi(V_2^{(1)}) \end{bmatrix} \quad (2.11)$$

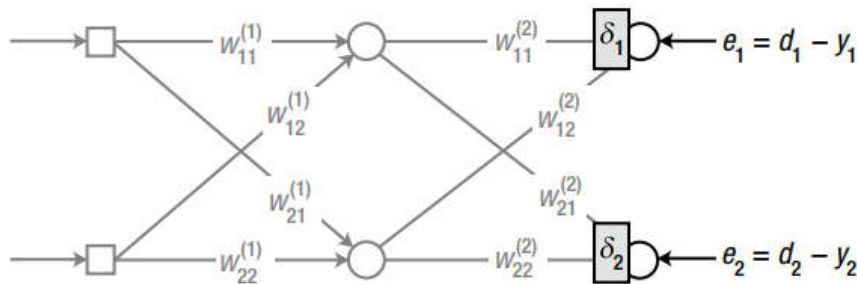
Simbol $Y_1^{(1)}$ dan $Y_2^{(1)}$ merupakan keluaran dari masing-masing *hidden node*. Menggunakan cara yang sama, maka nilai jumlah tertimbang dari *node* keluaran adalah sebagai berikut[22].

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} W_{11}^{(2)} & W_{12}^{(2)} \\ W_{21}^{(2)} & W_{22}^{(2)} \end{bmatrix} \begin{bmatrix} Y_1^{(1)} \\ Y_2^{(1)} \end{bmatrix} \triangleq W_2 Y^{(1)} \quad (2.12)$$

Ketika nilai jumlah tertimbang dimasukkan ke dalam fungsi aktivasi, maka akan menghasilkan keluaran sebagai berikut.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \varphi(V_1) \\ \varphi(V_2) \end{bmatrix} \quad (2.13)$$

Selanjutnya, akan dilakukan pelatihan JST dengan menggunakan algoritma propagasi balik. Hal pertama yang harus dilakukan adalah menghitung nilai delta (δ) setiap *node*. Delta ini merupakan pendekatan sistematis untuk memodifikasi bobot menurut informasi yang diberikan[22].



Gambar 2. 12 Nilai Delta Pada *Node* Keluaran[22]

Pada propagasi balik, δ dirumuskan sebagai berikut.

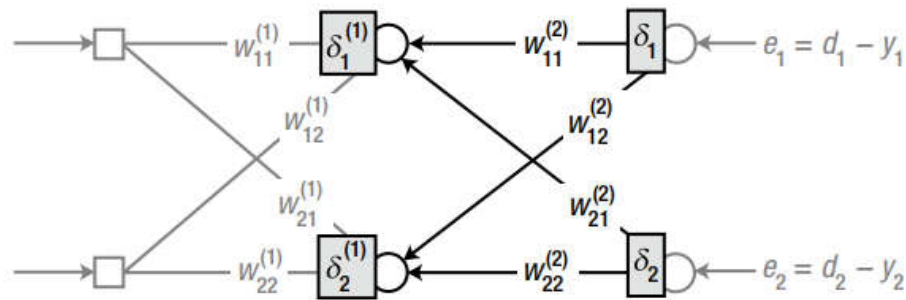
$$e_1 = d_1 - Y_1 \quad (2.14)$$

$$\delta_1 = \varphi'(V_1)e_1 \quad (2.15)$$

$$e_2 = d_2 - Y_2 \quad (2.16)$$

$$\delta_2 = \varphi'(V_2)e_2 \quad (2.17)$$

Simbol $\varphi'(\cdot)$ merupakan turunan fungsi aktivasi pada *node* keluaran, Y_i merupakan keluaran dari *node* keluaran, d_i merupakan keluaran yang benar dari hasil pelatihan data, e_i merupakan nilai *error*, dan V_i merupakan nilai jumlah tertimbang dari masing-masing *node*.



Gambar 2. 13 Nilai Delta Pada *Hidden Node*[22]

Sementara itu, delta untuk *hidden node* diilustrasikan dengan Gambar 2.13 dan persamaan seperti di bawah ini[22].

$$e_1^{(1)} = W_{11}^{(2)} \delta_1 + W_{21}^{(2)} \delta_2 \quad (2.18)$$

$$\delta_1^{(1)} = \varphi'(V_1^{(1)})e_1^{(1)} \quad (2.19)$$

$$e_2^{(1)} = W_{12}^{(2)} \delta_1 + W_{22}^{(2)} \delta_2 \quad (2.20)$$

$$\delta_2^{(1)} = \varphi'(V_2^{(1)})e_2^{(1)} \quad (2.21)$$

Dalam bentuk matriks, persamaan (2.18)sampai persamaan (2.21) di atas akan menjadi sebagai berikut.

$$\begin{bmatrix} e_1^{(1)} \\ e_2^{(1)} \end{bmatrix} = \begin{bmatrix} W_{11}^{(2)} & W_{21}^{(2)} \\ W_{12}^{(2)} & W_{22}^{(2)} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \quad (2.22)$$

Terlihat bahwa matriks bobot pada persamaan (2.22) di atas merupakan hasil *transpose* dari matriks bobot dari persamaan (2.10). Maka, persamaan (2.22) dapat diubah menjadi sebagai berikut.

$$\begin{bmatrix} e_1^{(1)} \\ e_2^{(1)} \end{bmatrix} = W_2^T \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \quad (2.23)$$

Untuk menyesuaikan bobot masing-masing *layer*, maka digunakan persamaan berikut ini.

$$\Delta W_{ij} = \alpha \delta_i X_j \quad (2.24)$$

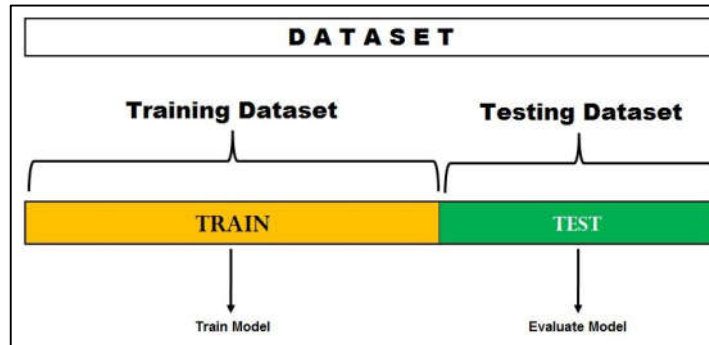
$$W_{ij} \leftarrow W_{ij} + \Delta W_{ij} \quad (2.25)$$

Terdapat beberapa algoritma pembelajaran/pelatihan jaringan yang sering digunakan pada JST. Salah satu algoritma tersebut adalah *Scaled Conjugate Gradient* (SCG). SCG merupakan algoritma pelatihan jaringan yang melakukan *update* bobot dan nilai bias sesuai dengan metode gradien konjugasi yang diskalakan [23][24]. SCG dapat melatih jaringan apapun selama bobot, *input*, dan fungsi transfernya memiliki fungsi turunan. Propagasi balik digunakan untuk menghitung turunan kinerja sehubungan dengan variabel bobot dan bias. Pada algoritma SCG, *step size* merupakan fungsi dari kuadrat perkiraan fungsi *error* yang membuatnya lebih kuat dan tidak tergantung pada parameter yang ditentukan pengguna [25]. Pelatihan berhenti jika salah satu dari kondisi berikut terjadi [25]:

1. Nilai *epoch*/repetisi maksimum tercapai
2. Waktu maksimal terlampaui
3. Performansi diminimalkan sampai mencapai tujuan
4. Gradien kinerja berada di bawah gradien minimum

5. Performansi validasi telah meningkat lebih dari max_fail kali sejak terakhir kali turun (saat menggunakan validasi)

2.2.6 *Hold-out Validation*



Gambar 2. 14 Ilustrasi *Hold-out Validation*

Hold-out validation merupakan metode validasi yang membagi data menjadi dua bagian yang tidak tumpang tindih dan kedua bagian ini masing-masing digunakan untuk pelatihan dan pengujian. Metode ini diusulkan untuk menanggulangi masalah *over-fitting* yang ditimbulkan oleh metode validasi *re-substitution*[26]. “*Hold-out*” menandakan satu bagian dari data yang akan digunakan pada proses pengujian. Sementara bagian data lainnya akan digunakan untuk proses pelatihan. Orang biasanya tidak memahami *hold-out validation* dengan sangat jelas dan menganggap bahwa *hold-out validation* membagi data menjadi dua bagian yang sama. Meskipun benar bahwa hal tersebut dapat disebut *hold-out validation*, bagaimanapun, hal tersebut adalah kasus *hold-out validation* yang sangat spesifik dengan jumlah data yang diberikan untuk pengujian sebesar 50%.

Dengan demikian, *hold-out validation* dapat memiliki persentase data yang berbeda antara data latih dan data uji. Misalnya dapat menggunakan 20% *hold-out* atau bahkan 10% *hold-out*. Tetapi pada saat yang sama, terkadang *hold-out* 10% mungkin mengalami *over-fitting* jika 90% data yang digunakan untuk mempelajari model tidak didistribusikan dengan benar dan sangat berbeda dari 10% data pengujian. Salah satu kelebihan metode *hold-out validation* yaitu waktu yang dibutuhkan untuk mempelajari model relatif lebih sedikit daripada waktu

yang dibutuhkan untuk mempelajari model dengan menggunakan *k-fold cross-validation*[26].

2.2.7 Confusion Matrix

Tabel 2.3 Confusion Matrix Untuk Klasifikasi 2 Kelas

		Terprediksi	
		Negatif	Positif
Aktual	Negatif	<i>a</i> (TN)	<i>b</i> (FP)
	Positif	<i>c</i> (FN)	<i>d</i> (TP)

Confusion matrix adalah suatu matriks yang berisi informasi tentang klasifikasi aktual dan prediksi yang dilakukan oleh sistem klasifikasi/*classifier*[27]. Performansi sistem tersebut biasanya dievaluasi menggunakan data yang terdapat pada *confusion matrix*. *Confusion matrix* berukuran $n \times n$ yang terkait dengan *classifier* menunjukkan klasifikasi yang diprediksi dan aktual. Simbol n adalah jumlah kelas yang berbeda. Tabel 2.2 menunjukkan *confusion matrix* untuk $n = 2$ yang entri-entrinya memiliki arti sebagai berikut[28].

- ***a*** adalah jumlah prediksi negatif yang benar/*True Negative* (TN)
- ***b*** adalah jumlah prediksi positif yang salah/*False Positive* (FP)
- ***c*** adalah jumlah prediksi negatif yang salah/*False Negative* (FN)
- ***d*** adalah jumlah prediksi positif yang benar/*True Positive* (TP)

Beberapa istilah standar telah didefinisikan untuk klasifikasi 2 kelas salah satunya yaitu akurasi. Akurasi adalah proporsi dari jumlah total prediksi yang benar. Akurasi ditentukan menggunakan persamaan[27]:

$$AC = \frac{a+d}{a+b+c+d} \quad (2.26)$$