

BAB 2

LANDASAN TEORI

Bab ini membahas tentang teori penunjang dan penelitian sebelumnya yang berhubungan dengan penerapan metode *probabilistic neural network* untuk mengidentifikasi *hypertensive retinopathy*.

2.1. *Hypertensive Retinopathy*

Hypertensive retinopathy adalah penyakit yang merusak retina mata hingga dapat mengakibatkan hilangnya penglihatan dan erat terkait dengan tekanan darah tinggi atau sering disebut dengan *hypertensive* (Narasimhan et al., 2012).

2.1.1. Gejala *Hypertensive Retinopathy*

Hypertensive retinopathy tidak memiliki gejala yang terlalu signifikan hingga terjadinya kerusakan yang parah pada retina. Penyakit ini pun sering disebut dengan *silent killer*, dikarenakan tidak memiliki gejala namun perlahan akan mengakibatkan kematian apabila tidak segera diobati (Badii, 2016). Tanda-tanda dari gejala penyakit ini diantaranya adalah :

- a. Penglihatan mulai berkurang.
- b. Terjadinya pembengkakan pada mata.
- c. Meledaknya pembuluh darah.
- d. Penglihatan ganda dengan disertai sakit kepala.

2.1.2. Penyebab *Hypertensive Retinopathy*

Penyebab dari *hypertensive retinopathy* dikarenakan tekanan darah tinggi yang berkepanjangan. Tekanan darah tinggi / *hypertensive* didefinisikan tekanan darah

sistolik ≥ 140 mmHg (*milimeter hydrargyrum*), tekanan darah diastolik ≥ 90 mmHg (*milimeter hydrargyrum*) (Ong et al., 2013). Sedangkan tekanan darah normal yaitu untuk tekanan darah sistolik < 120 mmHg dan diastolik < 80 mmHg. *Hypertensive* merupakan masalah kronis dimana kekuatan darah yang terlalu tinggi terhadap pembuluh darah. Ketika terjadinya *hypertensive*, maka pembuluh darah akan mengalami penyempitan kemudian penebalan dikarenakan suplai darah yang terlalu banyak menuju retina. Sehingga apabila tidak cepat diberi pengobatan, pembuluh darah akan meledak dan akan menyebabkan kehilangan penglihatan.

2.1.3. Faktor Resiko *Hypertensive Retinopathy*

Resiko terjadinya *hypertensive retinopathy* lebih sering terjadi pada perempuan dibandingkan laki-laki. Beberapa kondisi yang menempatkan pada resiko yang lebih tinggi untuk penyakit ini diantaranya adalah (Badii, 2016) :

- a. Tekanan darah tinggi / *hypertensive* yang berkepanjangan
- b. Penyakit jantung
- c. *Atherosclerosis* (penyakit arteri dimana dinding arteri menjadi lebih tebal dan kurang lentur).
- d. Diabetes
- e. Merokok
- f. Kolesterol tinggi
- g. Kelebihan berat badan
- h. Makan diet yang tidak sehat
- i. Konsumsi alkohol berat

2.1.4. Pemeriksaan *Hypertensive Retinopathy*

Pemeriksaan *hypertensive retinopathy* biasanya dilakukan dengan anamnesis, pemeriksaan fisik, pemeriksaan dengan alat funduskopi. Pemeriksaan dengan anamnesis merupakan pemeriksaan yang dilakukan melewati percakapan antara dosen dan pasien. Dokter akan menanyakan keluhan-keluhan yang diderita oleh pasien. Kemudian pemeriksaan fisik merupakan pemeriksaan yang biasanya dilakukan oleh dokter setelah pemeriksaan anamnesis. Pemeriksaan fisik yang dilakukan salah

satunya adalah mengukur tekanan darah dengan bantuan alat pengukur tekanan darah yang disebut tensimeter. Kemudian pada pemeriksaan funduskopi akan terlihat penyempitan *arterioles* retina. Peningkatan tekanan darah akan mengakibatkan penyempitan dan penebalan pada dinding pembuluh darah.

Selain itu juga digunakan alat *ophthalmoscope* dan *fluorescein angiography*. Pemeriksaan dengan *ophthalmoscope* yang bertujuan untuk melihat penyempitan pembuluh darah dan tanda kebocoran yang terjadi pada pembuluh darah tersebut (NIH, 2014). Kemudian pemeriksaan *fluorescein angiography*, citra hasil dari pemeriksaan tersebut terbentuk dari sejumlah foton yang dipancarkan oleh zat pewarna *fluorescein*. Zat tersebut disuntikkan kepada penderita dan zat akan beredar keseluruh tubuh termasuk retina. Ketika zat mulai masuk kebagian retina, maka proses *angiography* dilakukan. Citra yang dihasilkan dari proses ini dapat melihat pembuluh darah, mikroaneurisma, makula, dan pendarahan yang terjadi secara jelas (Febriani, 2014). Contoh hasil dari pemeriksaan tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1. Citra *fluorescein angiography* (Lim et al., 2015)

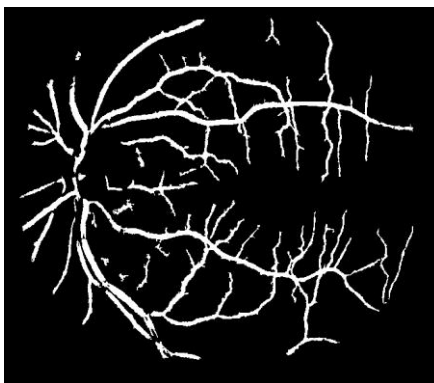
2.1.5. Klasifikasi *Hypertensive Retinopathy*

Pengelompokkan penyakit ini sudah banyak dilakukan. Downie et al., 2013 melakukan perbandingan hasil penelitian sebelumnya yang dilakukan oleh Keith, Wagener Barker pada tahun 1993. Klasifikasi yang dibuat berupa hubungan antara klasifikasi sederhana dari *hypertensive retinopathy* dan *cardiovascular disease*.

2.2.1. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Hitam direpresentasikan dengan nilai intensitas 0 sedangkan putih direpresentasikan dengan nilai intensitas 1. Citra biner juga disebut dengan citra B&W (*black and white*) atau citra monokrom. Karena hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner.

Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, penhambangan, morfologi, ataupun *dithering*. Contoh citra biner dapat dilihat pada Gambar 2.6.



Gambar 2.6 Citra Biner

2.2.2. Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih.

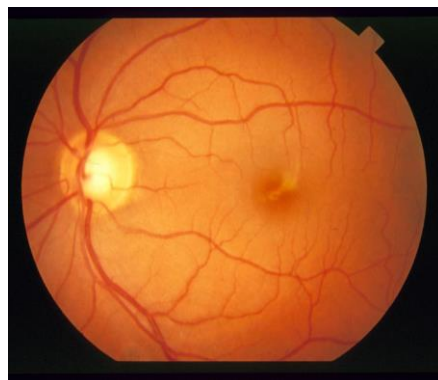
Tingkat keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Jika citra skala keabuan memiliki jumlah 8 bit, maka jumlah warna pada citra adalah 2^8 atau 256., dimana nilai intensitas berkisar antara 0-255. Nilai 0 merupakan warna hitam, nilai 255 merupakan warna putih dan nilai diantara itu adalah warna keabuan (Fatimah, 2016). Contoh citra *grayscale* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Citra *Grayscale*

2.2.3. Citra Warna

Citra warna merupakan jenis citra yang menyediakan warna dalam bentuk RGB (red, green, dan blue). Setiap komponen warna menggunakan 8 bit, nilainya terletak antara 0-255. Warna yang disediakan yaitu $255 \times 255 \times 255$. Warna ini disebut juga dengan *true color* dikarenakan memiliki jumlah warna yang cukup besar (Chairani, 2016). Contoh citra warna dapat dilihat pada Gambar 2.8.



Gambar 2.8 Citra Warna

2.3. Pengolahan Citra Digital

Pengolahan citra digital merupakan suatu teknologi yang menerapkan sejumlah algoritma komputer untuk dapat memproses citra digital. Hasil keluaran dari proses tersebut dapat berupa citra atau karakteristik yang merepresentasikan citra. Tujuan utama pengolahan citra ini yaitu untuk mendapatkan citra yang berkualitas tinggi atau deskriptif dari citra asli sehingga dapat meningkatkan informasi tentang citra tersebut

(Zhou et al., 2010). Beberapa teknik pengolahan citra yang diterapkan pada penelitian ini diantaranya sebagai berikut.

2.3.1. Green Channel

Green channel merupakan salah satu jenis dari *grayscale* yang mengganti nilai setiap piksel pada citra hanya dengan nilai *green* dari piksel citra tersebut (Febriani, 2014). *Green channel* mampu mendeteksi pembuluh darah dikarenakan dapat meningkatkan kontras yang lebih tinggi sehingga pembuluh darah dan *background* dari retina lebih terlihat jelas apabila dibandingkan dengan *channel* yang lainnya yaitu *red channel* dan *blue channel* yang kontrasnya lebih rendah (Raja, 2015). *Green channel* dilakukan dengan persamaan 2.1.

$$I(x, y) = 0.R + 1.G + 0.B = G \quad (2.1)$$

Dimana : $I(x, y)$ = piksel citra hasil *green channel*
 R = nilai *red* dari sebuah piksel
 G = nilai *green* dari sebuah piksel
 B = nilai *blue* dari sebuah piksel

2.3.2. Peningkatan Kontras Citra

Peningkatan kontras citra bertujuan untuk dapat meningkatkan kualitas citra dan dapat memperoleh citra yang dapat memberikan informasi yang sesuai dengan tujuan pengolahan citra. Peningkatan kontras citra dalam penelitian ini menggunakan metode *contrast limited adaptive histogram equalization* (CLAHE).

CLAHE merupakan metode untuk mengatasi keterbatasan standar pemerataan histogram pada suatu citra. CLAHE merupakan metode kelanjutan dari metode *adaptive histogram equalization* (AHE). Metode AHE cenderung masih banyak mengalami masalah *noise* di daerah yang relatif homogen dari suatu citra dan dengan CLAHE dapat mengatasi masalah tersebut dengan membatasi peningkatan kontras khususnya di daerah yang homogen (Sharma, 2013). Untuk mengontrol kualitas citra, CLAHE mempunyai dua parameter yaitu *block size* dan *clip limit* yang memiliki

beberapa nilai *default* dan juga bisa ditentukan oleh pengguna (Singh et al., 2015). Algoritma CLAHE dapat dijelaskan sebagai berikut (Ramya, 2012)

Langkah 1 : Citra asli dibagi menjadi beberapa bagian citra yang tiap bagian citra berukuran $M \times N$.

Langkah 2 : Setiap bagian citra dihitung histogramnya.

Langkah 3 : *Clipped histogram* setiap bagian citra. Jumlah piksel dari tiap bagian citra didistribusi pada masing-masing derajat keabuan. Rata-rata jumlah piksel tersebut dilakukan dengan persamaan 2.2.

$$N_{avg} = \frac{N_{SI-XP} \cdot N_{SI-YP}}{N_{graylevel}} \quad (2.2)$$

Dimana : N_{avg} = rata-rata jumlah piksel
 N_{SI-XP} = jumlah piksel dalam dimensi X dari bagian citra
 N_{SI-YP} = jumlah piksel dalam dimensi Y dari bagian citra
 $N_{graylevel}$ = jumlah nilai derajat keabuan dari bagian citra

Berdasarkan persamaan 2.2, *clip limit* dapat dihitung menggunakan persamaan 2.3.

$$N_{C-L} = N_c \cdot N_{avg} \quad (2.3)$$

Dimana : N_{C-L} = *clip limit*
 N_c = nilai maksimum rata-rata piksel setiap nilai derajat keabuan dari bagian citra

Pada histogram yang asli, piksel akan di *clipped* jika jumlah piksel lebih besar dari N_c . Jumlah piksel didistribusikan secara merata kedalam masing-masing derajat keabuan (N_d) yang didefinisikan dengan total jumlah piksel yang di *clipped* (N_{Tc}) dalam persamaan 2.4.

$$N_d = \frac{N_{Tc}}{N_{graylevel}} \quad (2.4)$$

$N_{SI}(i)$ merupakan jumlah piksel dalam setiap derajat keabuan bagian citra dan ‘i’ adalah jumlah derajat keabuan. Dengan menggunakan persamaan 2.4. *contrast limited histogram* bagian citra dapat dihitung dengan menggunakan persamaan 2.5.

$$\begin{aligned} & \text{if } H_{SI} > N_{C-L}, H_{NSI}(i) = N_{C-L} \\ & \text{else if } H_{SI}(i) + N_d \geq N_{C-L}, H_{NSI}(i) = N_{C-L} \\ & \text{else } H_{NSI}(i) = H_{SI}(i) + N_d \end{aligned} \quad (2.5)$$

Akhir dari distribusi pada persamaan 2.5, sisa jumlah piksel yang di *clipped* dinyatakan sebagai H_{RP} , tahap distribusi piksel dirumuskan dalam persamaan 2.6.

$$S = \frac{N_{gray}}{N_{RP}} \quad (2.6)$$

Metode ini memindai semua piksel dari yang minimum hingga yang maksimum dari nilai *graylevel*. Jika frekuensi piksel *graylevel* adalah N_{C-L} , metode ini akan mendistribusikan satu piksel nilai *graylevel*. Jika pencarian berakhir sebelum distribusi semua piksel, maka akan dihitung ulang sesuai dengan persamaan 2.6 hingga semua piksel terdistribusi. Dengan demikian akan diperoleh *histogram* yang baru.

Langkah 4 : Membatasi *contrast histogram* setiap bagian citra diproses dengan HE kemudian piksel dari bagian citra dipetakan dengan menggunakan interpolasi linear.

2.3.3. Morphological Operator

Morphological operator merupakan suatu teknik pengolahan citra yang berdasarkan pada pengolahan bentuk. Teknik ini menerapkan *structuring element* (SE) pada citra yang diolah dan menghasilkan citra dengan ukuran yang sama. SE merupakan sebuah operator yang dapat mempengaruhi kinerja pengolahan *morphological*. Nilai setiap piksel pada citra yang dimasukkan berdasarkan pada perbandingan antara piksel yang bersesuaian dari citra masukkan dengan nilai piksel tetangganya. Dengan memilih ukuran dan bentuk tetangga tersebut, maka dapat membangun sebuah *morphological*

operator untuk mengolah citra yang dimasukkan agar lebih spesifik (Kaur et al., 2013).

Morphological operator mempunyai dua operasi dasar yaitu *dilation* dan *erosion*. *Dilation* merupakan suatu proses untuk meningkatkan batas piksel *foreground* sehingga pada daerah tersebut ukurannya akan bertambah dan menebal. *Dilation* dilakukan dengan persamaan 2.7.

$$D(A, B) = A \oplus B = \{x : B_x \cap A \neq \emptyset\} \quad (2.7)$$

Dimana : D = citra hasil dari *dilation*
 A = citra masukan
 B = *structure element*
 B_x = translasi B

Sedangkan *erosion* kebalikan dari *dilation* yang akan mengurangi batas piksel *foreground* sehingga pada daerah tersebut ukurannya akan berkurang dan menipis (Chudasama et al, 2015). *Erosion* dilakukan dengan persamaan 2.8.

$$E(A, B) = A \ominus B = \{x : B_x \cap X\} \quad (2.8)$$

Dimana : E = citra hasil dari *erosion*
 A = citra masukan
 B = *structure element*
 B_x = translasi B

Operasi dasar tersebut telah banyak dikombinasikan, sehingga terdapat operasi-operasi lainnya salah satunya yaitu *morphological close*. Operasi tersebut merupakan kombinasi dimana suatu citra diterapkan *dilation* terlebih dahulu kemudian diterapkan operasi *erosion*. *Morphological close* bertujuan untuk memperhalus objek pada citra dengan cara menyambungkan pecahan-pecahan dan menghilangkan lubang-lubang kecil pada citra. Operasi *morphological close* diterapkan pada penelitian ini bertujuan untuk menghaluskan pembuluh darah retina beserta

struktur retina yang lain sehingga *background* dari retina lebih kelihatan lebih menonjol.

2.3.4. Thresholding

Thresholding merupakan suatu proses untuk mengubah citra menjadi citra biner atau sering disebut dengan proses binerisasi. Proses ini menggunakan nilai batas (*threshold*) untuk dapat mengubah nilai piksel menjadi warna hitam atau putih. Jika nilai piksel pada citra lebih besar dari nilai *threshold* yang ditentukan maka nilai piksel tersebut akan diubah menjadi warna putih dan diinisialkan dengan angka biner 1. Sementara apabila nilai piksel lebih kecil dari nilai *threshold* maka akan diubah menjadi warna hitam dan diinisialkan dengan angka biner 0 (Febriani, 2014). Proses tersebut dilakukan dengan persamaan 2.9.

$$g(x, y) \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (2.9)$$

Dimana : $g(x, y)$ = piksel citra hasil biner
 $f(x, y)$ = piksel citra masukan
 T = nilai *threshold*

2.3.5. Conneted Component Analysis

Conneted Component Analysis merupakan suatu teknik untuk mengekstrak daerah-daerah yang hanya terhubung dan menghilangkan daerah yang tidak terhubung setelah batas-batas minimum terdeteksi. Teknik ini biasanya juga sering digunakan dalam tahap segmentasi citra (Bouman, 2015). Terdapat dua konektivitas yang dapat digunakan yaitu 4-konektivitas (*4-connected neighbors*) dan 8-konektivitas (*8-connected neighbors*) (Chairani, 2016). 4-konektivitas apabila piksel-piksel yang berdekatan tersebut terletak berdampingan secara horizontal dan vertikal. Model dari 4-konektivitas ini dapat dilihat pada Tabel 2.2.

Tabel 2.2. Model 4-konektivitas

	$P(x,y-1)$	
$P(x-1,y)$	$P(x,y)$	$P(x+1,y)$
	$P(x,y+1)$	

Pada konsep 8-konektivitas apabila terdapat 2 piksel yang bersinggungan baik secara diagonal maupun secara horizontal dan vertikal maka akan dianggap satu objek. Model konektivitas tersebut dapat dilihat pada Tabel 2.3.

Tabel 2.3. Model 8-konektivitas

$P(x-1,y-1)$	$P(x,y-1)$	$P(x+1,y-1)$
$P(x-1,y)$	$P(x,y)$	$P(x+1,y)$
$P(x+1,y+1)$	$P(x,y+1)$	$P(x+1,y+1)$

2.4. Ekstraksi Ciri Citra

Ekstraksi ciri bertujuan untuk mengekstraksi ciri yang unik pada citra yang akan dijadikan input pada tahap klasifikasi. Ekstraksi ciri yang dilakukan pada penelitian ini menggunakan dua metode yaitu menggunakan *fractal dimension* yang menggunakan algoritma *box counting* dan menggunakan *invariant moments*.

2.4.1. Fractal Dimension

Konsep *fractal* berasal dari bahasa latin yaitu *fractus* yang berarti pecah atau tidak teratur. Objek *fractal* dapat ditemukan dimana-diaman, seperti garis pantai, pohon pakis, awan, gunung, bakteri, dll. *Fractal* memiliki karakteristik utama yaitu kemiripan dengan diri sendiri (*self-similarity*). Karakteristik tersebut yang membuat *fractal* memiliki kemampuan memodelkan objek alam yang rumit, tidak teratur, dan juga dapat menentukan dimensi suatu objek. Sehingga *fractal* mampu menghasilkan dimensi pecahan (*fractional dimension*). *Fractal* sangat serupa dengan dirinya, yang dapat dikatakan bahwa struktur dapat diulang dengan skala yang ukuran berbeda-beda. (Kisan et al., 2016).

Pada geometri *euclidean*, garis memiliki dimensi 1 karena hanya memiliki panjang. Bidang memiliki dimensi 2 karena memiliki panjang dan lebar. Ruang berdimensi 3 karena memiliki panjang, lebar, dan tinggi. Sementara titik tidak memiliki dimensi karena tidak memiliki panjang, lebar, ataupun tinggi. Namun, apabila bidang tersebut bukan bidang yang sempurna maka akan susah untuk menghitung dimensinya. Sehingga *fractal dimension* mampu berperan dalam menghitung dimensi dari bidang tersebut.

Salah satu metode dalam menghitung *fractal dimension* yaitu menggunakan metode *box counting*. *Fractal dimension* suatu citra dengan menggunakan metode ini dihitung dengan persamaan 2.10.

$$D(s) = \frac{\log(N(s))}{\log(s)} \quad (2.10)$$

Dimana : $N(s)$ = banyaknya kotak berukuran s
 $D(s)$ = *fractal dimension* dari suatu objek

Berikut adalah langkah-langkah metode penghitungan kotak adalah (Putra, 2010).

- Citra dibagi kedalam kotak-kotak dengan ukuran s . Nilai s berubah dari 1 hingga 2^k , dengan $k = 0, 1, 2, \dots$ dan seterusnya dan 2^k tidak boleh lebih besar dari ukuran citra. Bila citra berukuran $2^m \times 2^m$ maka nilai k akan berhenti sampai m .
- Hitung banyaknya kotak $N(s)$ yang melingkup satu objek. Nilai $N(s)$ akan sangat bergantung pada s
- Hitung *fractal dimension* dengan menggunakan persamaan 2.10.
- Langkah terakhir adalah membuat garis lurus berdasarkan nilai-nilai $\log(N(s))$ sebagai sumbu y , dan nilai-nilai $\log(s)$ sebagai sumbu x untuk setiap nilai s , kemudian hitung kemiringan (*slope*) dari garis lurus. Nilai dari *slope* inilah yang merupakan *fractal dimension* dari suatu citra. *Slope* suatu garis lurus dapat dihitung dengan metode *least square*. Suatu garis lurus dapat dinyatakan dengan persamaan 2.11.

$$Y = a_0 + a_1X \quad (2.11)$$

Kemiringan dari persamaan garis lurus diatas dinyatakan sebagai a_1 , dapat dihitung dengan metode *least square*. Metode tersebut dapat dilakukan dengan persamaan 2.12.

$$a_1 = \frac{N(\sum_{k=1}^N XY) - (\sum_{k=1}^N X)(\sum_{k=1}^N Y)}{N(\sum_{k=1}^N X^2) - (\sum_{k=1}^N X)^2} \quad (2.12)$$

Dengan N menyatakan banyak data yang digunakan untuk membentuk garis lurus.

2.4.2. Invariant Moments

Invariant moments merupakan salah satu metode yang sering digunakan untuk ekstraksi ciri bentuk dalam bidang pengolahan citra. Metode ini pertama kali dipublikasikan oleh Hu pada tahun 1961 (Chairani, 2016). Hasil dari metode ini terdapat tujuh nilai pada setiap objek citra. Nilai-nilai tersebut bersifat *independen* terhadap translasi, rotasi, dan perskalaan. Momen yang mentransformasikan fungsi citra $f(i,j)$ pada system diskrit dinyatakan pada persamaan 2.13.

$$m_{pq} = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} x^p y^q f(x, y) \quad (2.13)$$

Dimana : m_{pq} = momen
 H = tinggi citra
 W = lebar citra
 x dan y = baris dan kolom
 $f(x, y)$ = nilai intensitas citra

Momen pusat (*central moments*) μ adalah momen yang bersesuaian dengan pusat area didefinisikan pada persamaan 2.14.

$$\mu_{pq} = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (2.14)$$

Dimana : $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$

Dan momen pusat tersebut dilakukan normalisasi dengan menggunakan persamaan 2.15.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (2.15)$$

Dimana : $\gamma = \frac{p+q}{2} + 1$
 $\mu_{00} = m_{00}$

Kemudian momen yang telah ternormalisasi tersebut dapat didefinisikan sekumpulan momen-momen invarian (*invariant moments*). Persamaan dari momen tersebut dilakukan dengan persamaan 2.16.

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (2.16)$$

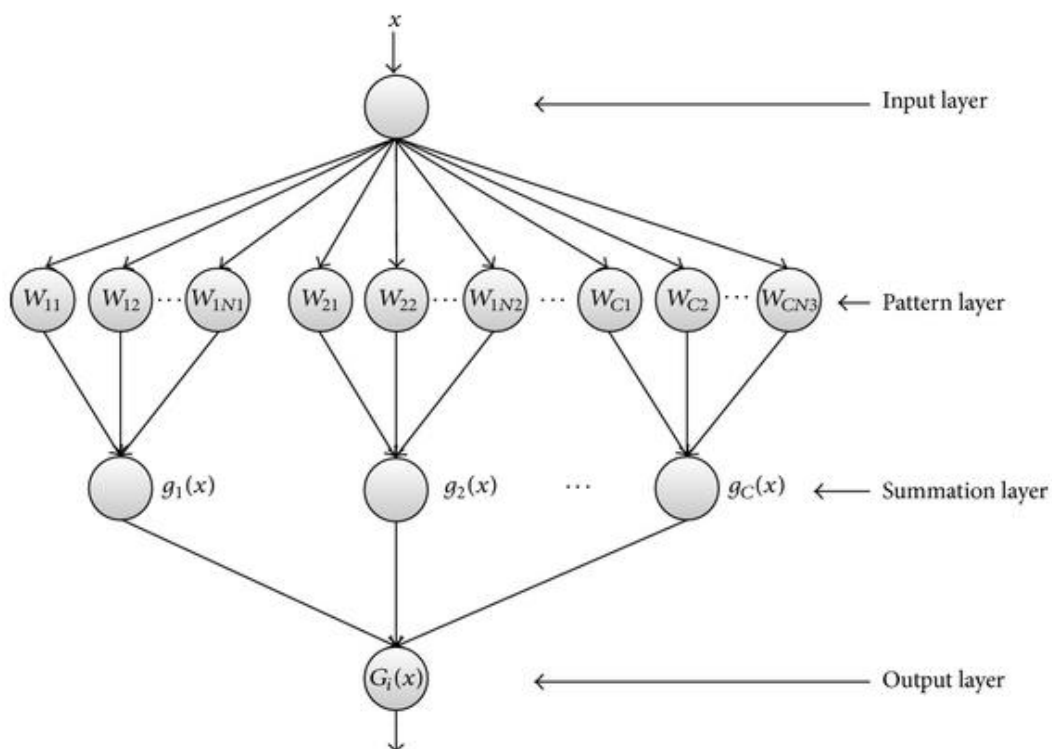
2.5. Probabilistic Neural Network

Probabilistic Neural Network (PNN) berdasarkan pada metode teorema Bayes untuk probabilitas bersyarat dan metode Parzen untuk memperkirakan fungsi kepadatan probabilitas variabel acak. PNN pertama kali diperkenalkan oleh Specht pada tahun 1990 yang menunjukkan bagaimana *Bayes Parzen Classifier* bisa dipecah menjadi sejumlah besar dari proses sederhana dan diimplementasikan kedalam jaringan saraf *multilayer* (Shahana et al, 2016).

PNN dapat didefinisikan sebagai implementasi dari algoritma statistik yang biasa disebut dengan kernel diskriminasi analisis dimana operasi tersebut akan disusun kedalam *multilayered feedforward network* dengan empat lapisan yaitu *input layer*, *pattern layer*, *summation layer*, dan *output layer*. Ada keuntungan utama yang membedakan PNN adalah proses pelatihan yang cepat, struktur paralel yang tidak dapat dipisahkan, dijamin dalam menemukan klasifikasi optimal sesuai dengan peningkatan perwakilan data pelatihan, dan pelatihan dapat ditambahkan atau dihapus tanpa melakukan pelatihan ulang. Dengan demikian, PNN belajar lebih cepat dari pada banyak model jaringan saraf tiruan dan telah sukses di beberapa aplikasi. Berdasarkan fakta tersebut, PNN dapat dilihat sebagai *supervised neural network* yang mampu digunakan dalam masalah klasifikasi dan pengenalan pola (Mishra, 2013).

PNN merupakan tipe khusus dari *radial basis neural network* terutama dalam masalah klasifikasi. Seperti *radial basis neural network*, PNN menggunakan fungsi aktivasi dilapisan kedua yaitu *hidden layer* yang bertujuan untuk membuat *local decision function* yang berpusat pada sampel dari *input layer*. Setelah pelatihan, fungsi tersebut dijumlahkan pada *summation layer*. Hasil dari jumlah fungsi tersebut itu merupakan probabilitas. Sehingga probabilitas yang paling *maximum* masuk kedalam sebuah kelas yang spesifik. *Neural network* ini biasanya digunakan untuk masalah dengan dataset pelatihan berukuran kecil (Lotfi, 2014).

PNN memiliki beberapa *layer*, diantaranya yaitu *input layer*, *radial basis layer*, *summation layer*, dan *output layer*. Struktur dari jaringan PNN ini dapat dilihat pada Gambar 2.9.



Gambar 2.9 Arsitektur *Probabilistic Neural Network* (Palomino et al, 2014)

- *Input Layer*

Pada lapisan ini terdapat variabel vektor input yang akan dijadikan input kedalam jaringan. Nilai dari variabel ini merupakan hasil dari ekstraksi ciri dari setiap data yang diuji.

- *Pattern Layer*

Pada lapisan ini dilakukan perhitungan kedekatan jarak antara vektor bobot dengan vektor input. Vektor bobot merupakan nilai dari data latih setiap kelas nya sedangkan vektor input merupakan nilai dari ekstraksi ciri data yang akan diuji. Proses yang terjadi pada lapisan ini menggunakan persamaan 2.17.

$$W_{ij}(x) = \frac{1}{2\pi^{d/2} \sigma^d} \exp \left[-\frac{\| (x - x_{ij}) \|^2}{2\sigma^2} \right] \quad (2.17)$$

Dimana : $W_{ij}(x)$ = gaussian kernel
 D = dimensi vector x
 σ = spread / smoothing parameter
 x = vektor pengujian
 x_{ij} = vektor pelatihan ke j dari kelas i

Tidak terdapat metode untuk menentukan nilai dari *smoothing parameter* sehingga digunakan teknik *trial and error*.

- *Summation Layer*

Pada lapisan ini menghitung penjumlahan kemungkinan maksimum dari setiap *i*-neuron pada lapisan *pattern layer* dengan kelas yang sama dan dirata-ratakan dengan jumlah data uji masing-masing kelas. Proses yang terjadi dengan menggunakan persamaan 2.18.

$$g_i(x) = \frac{1}{2\pi^{d/2} \sigma^d} \frac{1}{N_i} \sum_{j=1}^{N_i} \exp \left[-\frac{\|(x - x_{ij})\|^2}{2\sigma^2} \right] \quad (2.18)$$

Dimana : $g_i(x)$ = fungsi kepadatan probabilitas
 D = dimensi vector x
 σ = spread / smoothing parameter
 N = jumlah data latih pada kelas i
 x = vektor pengujian
 x_{ij} = vektor pelatihan ke j dari kelas i

- *Output Layer*

Pada lapisan terakhir ini dibandingkan nilai antara hasil dari dua kelas. Nilai probabilitas yang tertinggi maka akan dikelompokkan menjadi kelas tersebut. Proses yang dilakukan pada lapisan ini dengan menggunakan persamaan 2.19.

$$G_i(x) = \operatorname{argmax}\{g_i(x)\}, \quad i = 1, 2, \dots, m \quad (2.19)$$

Dimana : $G_i(x)$ = bayes's decision

a. Proses Pelatihan (*training*)

Proses pelatihan terdiri langkah yang unik, yaitu bobot masing-masing neuron pada *pattern layer* terbentuk oleh vektor karakteristik dari masing-masing data pelatihan.

b. Proses Pengujian (*testing*)

Pada tahap pengujian, data input / data uji yang akan akan diklasifikasikan akan mengalami proses yang seperti pada arsitektur Gambar 2.9. Dimana data uji akan masuk kedalam *pattern layer*, proses yang terjadi yaitu dengan menerapkan fungsi *gaussian kernel*. Kemudian, selanjutnya masuk kelapisan *summation layer* dimana dilakukan penjumlahan hasil dari fungsi *gaussian kernel* yang dikelompokkan dari kelas yang sama kemudian dirata-ratakan dengan jumlah data uji dari masing-masing kelas. Pada tahap ini menggunakan fungsi kepadatan probabilitas. Tahap terakhir yaitu mengambil nilai probabilitas yang tertinggi akan masuk kedalam kelas tersebut. Tahap ini menggunakan *bayes's decision*.

2.6. Penelitian Terdahulu

Beberapa penelitian sebelumnya telah dilakukan, diantaranya pernah dilakukan untuk mengidentifikasi *hypertensive* melalui analisis *fundus images* dengan menghitung rasio vena arteri. Pada tahap *preprocessing*, menggunakan metode *Adaptive Histogram Equalization* (AHE) yang digunakan untuk menyamakan kecerahan dan kontras yang berbeda setelah melewati tahap *green channel*. Metode yang lain digunakan yaitu *Radon Transform* untuk segmentasi pembuluh darah dan menggunakan *Hough Transform* untuk mendeteksi *optic disk*. Deteksi dan identifikasi *optic disk* sangat signifikan dalam menemukan *region of interest* (ROI) yang merupakan wilayah standar untuk menghitung rasio vena arteri. Hasil perhitungan dari rasio vena arteri ini yang akan dimanfaatkan oleh dokter untuk identifikasi penyakit *hypertensive retinopathy* dan akurasi yang dicapai 92% (Noronha et al., 2012).

Penelitian selanjutnya yaitu bertujuan untuk segmentasi pembuluh darah pada pasien penyakit *diabetic retinopathy*. Tahapan yang dilakukan untuk segmentasi pembuluh darah yang dilakukan adalah mengubah citra RGB menjadi *green channel*, meningkatkan kontras dengan menggunakan *contrast limited adaptive histogram*

equalization, eliminasi *optical disk* dengan menggunakan *morphological close, background exclusion*, dan pembentukan citra biner dengan menggunakan *thresholding* (Joshi et al., 2012).

Penelitian selanjutnya yaitu dilakukan pada penyakit *hypertensive retinopathy* dan *Cerebral Autosomal Dominant Arteriopathy with Subcortical Infarcts and Leukoencephalopathy* (CADASIL). Penilaian kuantitatif rasio vena arteri, indeks tortuositas dan *fractal dimension* dilakukan setelah ekstraksi pembuluh darah. *Fractal dimension* menggunakan algoritma *Box Counting*. Salah satu hasil dari penelitian ini mengatakan bahwa *fractal dimension* yang didapatkan dari penyakit *hypertensive retinopathy* lebih rendah dibandingkan dengan mata sehat (Cavallari et al., 2015).

Penelitian berikutnya dilakukan pada 3 kondisi yaitu pada pasien mata sehat, *diabetic retinopathy*, dan *glaucoma*. *Fractal analysis* dan *invariant moments* merupakan metode yang dipilih untuk tahap ekstraksi ciri setelah dilakukan ekstraksi pembuluh darah kemudian diubah menjadi citra biner yang menggunakan *kirsch's templates* dan metode *linear discriminant analysis* untuk tahap klasifikasi. Pada penelitian ini akurasi yang dicapai sangat tinggi sebesar 99,2% (Hutson et al., 2016).

Penelitian berikutnya yaitu meningkatkan teknik segmentasi pembuluh darah citra retina agar dalam mendeteksi ataupun mengidentifikasi suatu penyakit didapatkan hasil yang akurat. Tahapan yang dilakukan yaitu pembentukan *green channel* dari citra retina RGB, *morphological operation*, menghilangkan *noise* dengan menggunakan *rician denoise*. Kemudian tahap segmentasi menggunakan *thresholding* dan tahap terakhir yaitu *post processing* dengan menggunakan *length filtering* dan *connected component analysis*. Akurasi yang dicapai sebesar 94,35% pada data DRIVE & 94,49% pada data STARE (Mehta et al., 2016).

Penelitian berikutnya yaitu mendiagnosis penyakit diabetes tipe II. Metode yang digunakan pada penelitian ini yaitu *probabilistic neural network*. *Neural network* ini memiliki kemampuan untuk mendiagnosa dengan kesalahan minimum. Penelitian ini diterapkan dalam bidang data mining dan dataset yang digunakan dari Pima Indians Diabetes dan akurasi yang dicapai sebesar 81,49% (Soltani et al., 2016).

Penelitian selanjutnya dilakukan untuk mengidentifikasi penyakit pada daun teh. Penyakit yang akan diidentifikasi diantaranya yaitu *helopeltis* dan *blister blight*. Sebelum masuk ketahap ekstraksi ciri, citra daun teh dilakukan pengolahan citra yaitu dilakukannya *resizing* dan *grayscale*. Kemudian pada tahap segmentasi dilakukan

deteksi tepi menggunakan sobel. Metode *nvariant moments* dipilih untuk tahap ekstraksi ciri dan metode PNN untuk tahap klasifikasi dan akurasi yang dicapai sebesar 90% (Oktariani, 2016).

Penelitian terdahulu yang telah dipaparkan akan diuraikan secara singkat pada Tabel 2.4.

Tabel 2.4. Penelitian Terdahulu

No	Peneliti	Tahun	Metode	Accuracy
1	Noronha et al	2012	<i>Adaptive Histogram Equalization & Radon Transform</i>	92%
2	Joshi et al	2012	<i>Contrast Limited Adaptive Histogram Equalization & Morphological Operation</i>	-
3	Cavallari et al	2015	<i>BRetina Algorithm & Box Counting</i>	-
4	Hutson et al	2016	<i>Box Counting , Invariant Moments, Linear Discriminat Analysis</i>	99.2%
5	Mehta et al	2016	<i>Morphological Operation, Rician Denoise Method, Connected Component Analysis</i>	94,35% & 94,49%
6	Soltani et al	2016	<i>Probabilistic Neural Network</i>	81,49%
7	Oktariani	2016	<i>Probabilistic Neural Network</i>	90%