

## **BAB II**

### **LANDASAN TEORI**

#### **A. Beras**

##### **1. Pengertian Beras**

Definisi secara umum beras sesuai Peraturan Menteri Perdagangan RI Nomor 19/M-DAG/PER/3/2014 menjelaskan beras adalah biji-bijian baik berkulit, tidak berkulit, diolah atau tidak diolah yang berasal dari *Oriza Sativa*. Pada definisi ini beras mencakup gabah, beras giling, dan beras pecah kulit. Sedangkan definisi umum, beras merupakan bagian bulir padi (gabah) yang telah dipisah dari sekam dan dedak atau bekatul (Kementan, 2015). Definisi ini menunjukkan beras sebagai produk akhir dari gabah. Gabah tidak didefinisikan sebagai beras. Bentuk gabah dan beras ditunjukkan pada Gambar 1.



Gambar 1. Gabah Dan Beras

Gubah dan beras merupakan biji-bijian yang dihasilkan oleh tanaman *Oriza Sativa*. Bentuk tanaman *Oriza Sativa* ditunjukkan pada Gambar 2. Penemuan padi tertua di Indonesia berada di wilayah Papua pada 7000 tahun silam (Levetin, 2008:183). Beras diperoleh melalui beberapa tahapan penanaman padi, meliputi tahap panen dan pasca panen padi atau *Oriza Sativa*.



Gambar 2. Tanaman *Oriza Sativa*

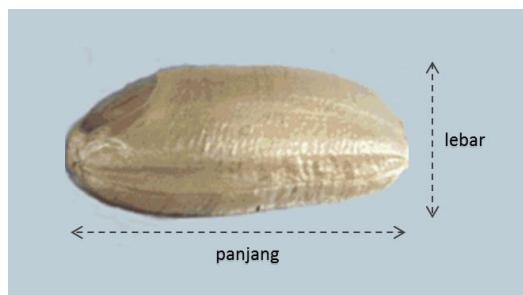
Pada tahap panen padi menghasilkan jenis gabah yang beragam sesuai ciri fisik bulir padi, salah satunya adalah jenis gabah konsumsi. Gabah konsumsi memiliki ciri-ciri: seluruh tanaman tampak kuning, bulu-bulu berwarna hijau hanya pada bagian atas dari semua bagian tanaman, dan isi gabah sudah keras tetapi mudah pecah dengan kuku (kementan, 2015).

Pada tahap pasca panen, gabah konsumsi menghasilkan beras pecah kulit. Beras pecah kulit merupakan beras yang telah mengalami proses pemisahan isi dengan kandung lembaga. Beras pecah kulit baru bisa dikonsumsi setelah disosoh menjadi beras giling. Beras giling merupakan beras yang telah mengalami proses penghilangan sekam, lapisan aleuron (dedak) dan kotiledon. Beras giling juga disebut sebagai beras sosoh (Kementan, 2015).

Beras giling atau beras sosoh digunakan sebagai olahan nasi untuk konsumsi. Beras sosoh ini lebih dikenal masyarakat sebagai beras pada umumnya. Beras telah dikonsumsi oleh tidak kurang dari 26 negara padat penduduk (China, India, Indonesia, Pakistan, Bangladesh, Malaysia, Thailand, Vietnam), atau lebih separuh penduduk dunia (Koswara, 2009:2).

## 2. Ukuran dan Bentuk Beras

Ukuran beras dipengaruhi oleh panjang butir beras, dan lebar butir beras. Panjang butir beras diukur antara dua ujung butir beras utuh (Badan Standarisasi Nasional, 2008). Bentuk beras ditentukan oleh nilai rasio panjang (P) terhadap lebar (L) butir beras. Lebar butir beras diukur antara punggung dan perut beras utuh menggunakan alat *micrometer* (Badan Standarisasi Nasional, 2008). Ilustrasi pengukuran beras ditunjukkan pada Gambar 3 (Rost, 1997),



Gambar 3. Ilustrasi Panjang dan Lebar Beras

Berdasarkan panjangnya, beras dibagi dalam empat tipe, yaitu: “sangat panjang” dengan panjang lebih dari 7 mm, “panjang” dengan panjang 6 mm sampai dengan 7 mm, “sedang” dengan panjang antara 5,0 mm sampai 5,9 mm, dan “pendek” dengan panjang kurang dari 5 mm (Belsnio, 1992). Tabel 1 menunjukkan besaran panjang beras antara penelitian FAO dan Balai Penelitian.

Tabel 1. Panjang Beras

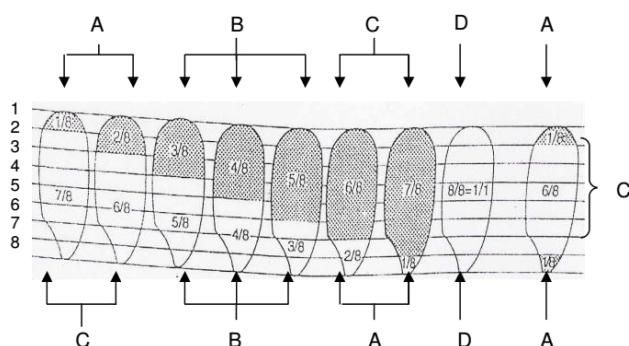
SUBTIPE BERAS	FAO	BALAI PENELITIAN
Sangat panjang	> 7,0 mm	> 7,5 mm
Panjang	6,01 - 7 mm	6,61 – 7,50 mm
Sedang	5,01 – 6 mm	5,51 – 6,60 mm
Pendek	< 5,01 mm	< 5,50 mm

Berdasarkan bentuknya, beras dapat dibagi menjadi empat tipe, yaitu: “lonjong” dengan rasio lebih dari 3, “sedang” dengan rasio 2,4 sampai dengan 3,0, “agak bulat” dengan rasio antara 2,0 sampai 2,39, dan “bulat” dengan rasio kurang dari 2 (McKill, 1996:160). Tabel 2 menunjukkan besaran bentuk beras antara penelitian IRRI dan Balai Penelitian.

Tabel 2. Besaran Bentuk Beras (Rasio P/L)

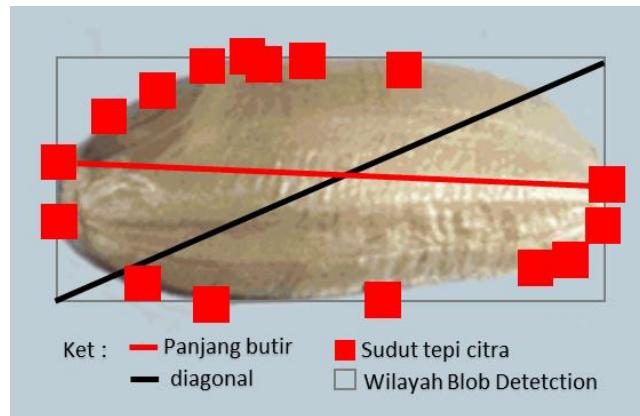
<b>SUBTIPE BERAS</b>	<b>IRRI</b>	<b>BALAI PENELITIAN</b>
Lonjong / ramping (slender)	> 3,0	>3,0
Medium	2,1 - 3,0	2,1 – 3,0
Agak bulat	1,1 - 2,0	-
Bulat (bold)	< 1,1	≤ 2,0

Sebuah butir beras yang telah terpisah, dapat dianalisis 2 tipe beras yang berbeda. Potongan kecil dari sebuah butir beras utuh menghasilkan beras kepala dan butir menir. Gambar 4 (Badan Standarisasi Nasional, 2008) menunjukkan bagian-bagian dari beras, dengan tipe A merupakan butir menir, tipe B merupakan butir patah, tipe C merupakan Beras kepala, dan tipe D merupakan Beras utuh.



Gambar 4. Bagian-Bagian Beras : (A) Patahan Kecil, (B) Patahan Besar, (C) Beras Kepala, (D) Beras Utuh

Pengambilan citra pada sebuah butir beras dapat dilakukan pendeksiyan pada sudut tepi citra, dan hasil pengujian diagonal citra pada sudut tepi citra. Ilustrasi diagonal antar sudut tepi citra beras ditunjukkan Gambar 5



Gambar 5. Illustrasi Diagonal Antar Sudut Tepi Citra Beras.

### 3. Kebijakan Perberasan Nasional

Masalah perberasan telah menyangkut masalah bangsa dan antar bangsa yang terjadi selama bertahun-tahun. Pada masa berlakunya Repelita I, kualitas beras produksi Indonesia dibedakan sesuai jenis, karakter, dan daerah produksi beras, seperti: rojolele, saigon bandung putih, DN II, LIMO, BGA, BGS Krawang, dan sebagainya. Patokan kualitas beras secara empiris terjadi pada masa Repelita II (Hutabarat, 1974:113). Tabel 3 menunjukkan perbedaan sistem standarisasi beras pada masa Repelita I, dan Repelita II.

Tabel 3. Standarisasi Beras Nasional Tahun 1973

Kualitas	Sebelum – 1973	1974 – setelahnya
Bagus	rojolele, saigon bandung putih	Broken < 15%
Biasa	DN II, LIMO, BGA,	Broken 15% - 25%
Buruk	BGS Krawang	Broken >25%

Pada tahun 1998, bergulir program beras bersubsidi untuk masyarakat miskin dalam mengatasi masalah krisis yang terjadi akibat runtuhnya orde baru. Program beras bersubsidi merupakan implementasi dari Instruksi Presiden tentang

kebijakan perberasan nasional (Tim Koordinasi Raskin Pusat,2014). Komponen kualitas beras mengacu Inpres No 3/2007 (Kemedagri, 2007) dan diperbaharui kembali melalui Inpres No.5 tahun 2015 (Kemedagri, 2015).

#### 4. Standar Kualitas Beras

Menurut Soerjandoko (2010:44-47), dalam penelitiannya mengenai teknik pengujian mutu beras skala laboratorium, kualitas beras sangat bergantung pada mutu gabah yang akan digiling dan sarana mekanis yang digunakan dalam penggilingan. Kualitas gabah juga dipengaruhi oleh genetik tanaman, cuaca, waktu pemanenan, dan penanganan pascapanen.

Tabel 4. Spesifikasi Mutu Beras Giling, SNI: 6128-2008

Komponen Mutu	Satuan	Mutu 1	Mutu 2	Mutu 3	Mutu 4	Mutu 5
Derajat Sosoh	% min	100	100	95	95	85
Kadar air	% mak	14	14	14	14	15
Beras kepala	% min	95	89	78	73	60
butir Patah	% mak	5	10	20	25	35
butir menir	% mak	0	1	2	2	5
Butir merah	% mak	0	1	1	3	3
Butir kuning	% mak	0	1	1	3	5
Butir mengapur	% mak	0	1	1	3	5
Benda Asing	% mak	0	0,02	0,02	0,05	0,2
Butir Gabah	btr/100g	0	1	1	2	3

Tabel 4 menunjukkan spesifikasi antar mutu beras giling. Spesifikasi kualitas beras berdasarkan hasil penelitian yang dilakukan oleh Litbang Kementerian Pertanian. Hasil penelitian tersebut kemudian dijadikan standar nasional SNI.

Standar beras bersubsidi yang diberlakukan oleh Bulog sesuai mutu 4 dalam standar SNI dengan butir patah maksimal 20%.

## 5. Komponen Kualitas Beras

Kualitas beras tergantung dari penilaian terhadap preferensi maupun produk akhir yang diinginkan oleh konsumen (BPTP, 2006). Teknik pengujian komponen kualitas beras sesuai ketetapan SNI: 6128-2015 yang disesuaikan dengan inpres No 3/2007, yakni:

a. Butir patah

Beras patah yaitu butir beras sehat maupun cacat dengan ukuran lebih besar atau sama dengan 0,2 sampai dengan lebih kecil 0,8 bagian dari butir beras utuh. Penentuan butir patah dimulai dengan mengambil sampel sebanyak 100 gram beras. Butir patah dipisahkan dengan beras kepala menggunakan *rice grader*. Butir patah dipisahkan dengan menggunakan ayakan 2,0 mm atau menggunakan pinset dan kaca pembesar secara visual. Prosentase beras patah ditunjukkan pada persamaan 1.

$$\text{Prosentase beras patah} = \frac{\text{berat beras patah}}{\text{Berat sampel}} \times 100\% \quad (1)$$

b. Butir menir

Butir menir yaitu butir beras sehat maupun cacat yang mempunyai ukuran lebih kecil dari 0,2 bagian butir beras utuh. Penentuan butir menir diawali dengan pengambilan sampel sebanyak 100 gram beras. Butir menir dipisahkan dengan beras kepala menggunakan *rice grader*. Butir menir dipisahkan menggunakan

pinset dan kaca pembesar secara visual. Prosentase butir menir ditunjukkan pada persamaan 2.

$$\text{Prosentase beras menir} = \frac{\text{berat butir menir}}{\text{Berat sampel}} \times 100\% \quad (2)$$

c. Butir kuning

Butir kuning yaitu butir beras utuh, beras kepala, beras patah, dan menir yang berwarna kuning, kuning kecoklatan, atau kuning semu akibat proses fisik atau aktifitas mikroorganisme. Penentuan butir kuning atau rusak dilakukan dengan cara mengambil sampel sebanyak 100 gram beras, kemudian dipisahkan secara visual menggunakan pinset dan kaca pembesar. Persamaan 3 menunjukkan perhitungan kadar butir kuning.

$$\text{Kadar butir kuning} = \frac{\text{berat butir kuning}}{\text{Berat sampel}} \times 100\% \quad (3)$$

d. Derajat Sosoh

Derajat sosoh yaitu tingkat terlepasnya lapisan perikarp, testa, dan aleuron serta lembaga dari tiap butir beras. Derajat sosoh 95% merupakan tingkat terlepasnya lapisan perikarp, testa, dan aleuron dari butir beras sehingga sisa yang belum terlepas sebesar 5%. Derajat sosoh dapat ditentukan secara kualitatif dan kuantitatif.

Penentuan derajat sosoh secara kuantitatif dikenal dengan metode konversi, yakni menggunakan alat pengukur derajat putih. Nilai derajat sosoh ditentukan dari hasil prosentase derajat putih yang dikonversikan berdasarkan Tabel 5.

Tabel 5. Konversi Nilai Derajat Sosoh Beras

Butir Panjang ( <i>Long Grain</i> )		Butir Bulat ( <i>Short Grain</i> )	
Derajat Putih (%)	Derajat Sosoh (%)	Derajat Putih (%)	Derajat Sosoh (%)
43,57	80	48,86	80
44,12	81	49,47	81
44,66	82	50,08	82
45,21	83	50,69	83
45,75	84	51,31	84
46,30	85	51,91	85
46,84	86	52,52	86
47,38	87	53,13	87
47,93	88	53,74	88
48,47	89	54,35	89
49,02	90	54,96	90
49,56	91	55,57	91
50,11	92	56,18	92
50,65	93	56,80	93
51,20	94	57,41	94
51,74	95	58,02	95
52,29	96	58,63	96
52,83	97	59,24	97
53,38	98	59,85	98
53,92	99	60,46	99
54,47	100	61,07	100

## 6. Varietas Beras

Varietas beras yang sangat sering digunakan oleh masyarakat adalah jenis IR 64. Sampel jenis ini memiliki warna beras putih semu. Penjelasan dari varietas beras ini sesuai Tabel 6 (Suprihatno, 2009 : 5).

Tabel 6. Varietas Beras IR 64

SPESIFIKASI	KETERANGAN
Nomor seleksi	IR18348-36-3-3
Asal persilangan	IR5657/IR2061
Golongan	Cere
Umur Tanaman	110 – 120 hari
Bentuk Tanaman	Tegak
Tinggi Tanaman	115 – 126 cm
Anakan Produktif	20 – 35 batang
Warna Kaki	Hijau
Warna Batang	Hijau
Warna Telinga Daun	Tidak berwarna
Warna Lidah Daun	Tidak berwarna
Warna Daun	Hijau
Muka Daun	Kasar
Posisi Daun	Tegak
Daun Bendera	Tegak
Bentuk gabah	Ramping, Panjang
Warna Gabah	Kuning Bersih
Kerontokan	Tahan
Kerebahana	Tahan
Tekstur Nasi	Pulen
Kadar Amilosa	23%
Indeks Gilkemik	70
Bobot 1000 butir	24,1 gram
Rata-rata Hasil	5,0 t/ha
Potensi Hasil	6,0 t/ha
Ketahanan terhadap hama	Tahan Wereng Coklat biotipe 1,2 dan agak tahan wereng coklat biotipe 3
Penyakit	Agak tahan hawar daun bakteri strain IV, Tahan virus kerdil rumput
Anjuran Tanam	Baik ditanam di lahan sawah irigasi dataran rendah sampai sedang
Pemulia	Inroduksi dari IRRI
Dilepas tahun	1986

## B. Pengolahan Citra Digital

Citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek dan memantulkan kembali sebagian dari berkas cahaya tersebut (Munir, 2002:2). Citra sebagai luaran suatu sistem perekaman data dapat bersifat optik, analog, atau digital. Citra yang dapat langsung disimpan pada suatu media penyimpanan disebut citra digital. Citra digital dapat diolah oleh komputer (Sutoyo, 2009:9).

Sebuah citra digital membentuk bidang dua dimensi untuk dapat dilakukan pengolahan citra (Sutoyo, 2009:10). koordinat  $(x, y)$  menyatakan posisi koordinat pada sistem Kartesius dengan sumbu mendatar dinyatakan sebagai sumbu -  $x$ , dan sumbu tegak dinyatakan sebagai sumbu -  $y$ . Fungsi intensitas cahaya pada koordinat  $(x, y)$  disimbolkan dengan  $f(x, y)$  (Munir, 2002:16). Karena cahaya merupakan bentuk energi, maka intensitas cahaya bernilai antara 0 sampai tidak berhingga, ditunjukkan pada persamaan 4.

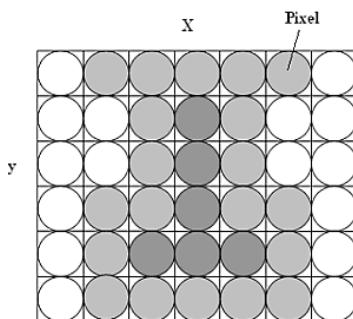
$$0 < f(x, y) < \infty \quad (4)$$

Fungsi intensitas  $f(x, y)$  merupakan hasil kali dari jumlah cahaya  $i(x, y)$  atau *illumination* dari sumbernya cahaya yang menerangi objek atau *illumination*, dengan derajat kemampuan objek  $r(x, y)$  dalam memantulkan cahaya oleh objek atau *reflection*, sesuai dengan persamaan 5. Rentang nilai  $i(x, y)$  adalah  $[0, \infty]$ , sedangkan rentang nilai  $r(x, y)$  adalah  $[0,1]$

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (5)$$

## 1. Piksel

Cara yang paling praktis dalam menyimpan sebuah citra digital adalah dengan membagi citra menjadi sekumpulan kisi-kisi persegi yang disebut piksel (Prahasty, 2008). Piksel menunjukkan elemen yang berkedudukan pada koordinat (x, y) dari sebuah citra visual (Solomon, 2011:68), seperti pada Gambar 6.



Gambar 6. Visualisasi Piksel Sumbu (X,Y) Pada Sebuah Citra Digital

Citra digital adalah sajian citra dalam nilai diskrit, baik pada koordinat ruang maupun nilai intensitas cahaya. Proses representasi citra ke dalam nilai diskrit disebut digitalisasi (Solomon, 2011:68). Setiap piksel diberi nilai dari fungsi intensitas yakni komponen warna dan tingkat intensitas cahaya (Prahasty, 2008). Piksel dianggap sebagai titik cahaya berwarna karena mengandung warna atau intensitas respon dari gambar visual. Nilai fungsi intensitas dari suatu piksel bervariasi tergantung jenis gambar (Solomon, 2011:68-69). Persamaan 6 menunjukkan himpunan piksel dalam matriks.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & f(1,1) & \cdots & f(1,M-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix} \quad (6)$$

Citra digital berbentuk persegi yang memiliki dimensi ukuran dinyatakan sebagai tinggi x lebar atau lebar x panjang yang berupa himpunan matriks.

## 2. Operasi pada Piksel

Operasi translasi adalah memindahkan elemen piksel citra *input* ke posisi baru pada citra *output* dengan ukuran piksel keduanya adalah sama (Putra, 2010:156). Translasi sebuah himpunan B oleh titik z ( $z_1, z_2$ ) dinyatakan sebagai  $(B)_z$ , yang didefinisikan dengan persamaan 7.

$$(B)_z = \{c \mid c = b + z, \text{ untuk } b \in B\} \quad (7)$$

Himpunan piksel matrik dapat ditranslasikan dengan mengubah arah translasi ke dalam bentuk matrik juga. Jika arah translasi ke bawah atau ke kiri, maka nilai translasi bernilai negatif. Begitu sebaliknya, jika arah translasi ke atas atau ke kanan, maka nilai translasi bernilai positif (Boyd, 2005).

Persamaan 8 menunjukkan contoh translasi dari suatu matriks pada segitiga. Suatu segitiga HIJ dengan titik titik sudut yang berkedudukan pada H(a,b), I(c,d), J(e,f) dikenai operasi translasi oleh matriks  $[x, y]$ . Menghasilkan segitiga baru PQR sesuai dengan persamaan tersebut (Boyd, 2005).

$$\Delta PQR = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a+x & c+x & e+x \\ b+y & d+y & f+y \end{bmatrix} \quad (8)$$

Dilatasi merupakan operasi transformasi perubahan ukuran tanpa mengubah bentuk matriks vektor (Boyd, 2005). Dilatasi memberikan efek memperbesar atau memperkecil ukuran citra *input* sesuai dengan variabel penskalaan citra (Putra, 2010: 159). Proses penskalaan dapat dinyatakan pada persamaan 9.

$$P_0 = S_p \cdot P_i \quad ; \quad L_0 = S_l \cdot L_i \quad (9)$$

$(P_i, L_i)$  menyatakan ukuran citra *input*, dan  $(P_0, L_0)$  menyatakan citra *output*, dengan  $(S_p, S_l)$  adalah variabel penskalaan yang diinginkan. Sebagai sebuah

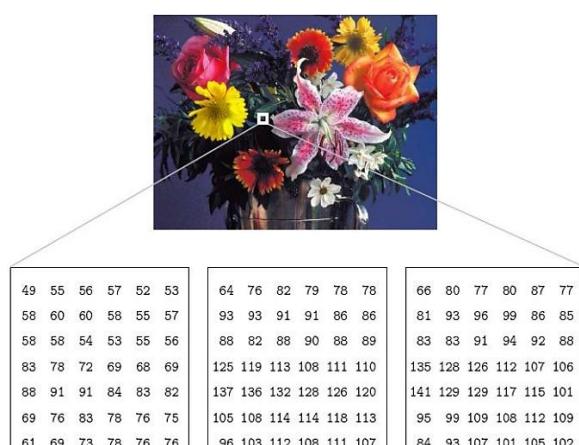
himpunan, citra dapat dikenai sebuah dilatasi matriks. Persamaan 10 menunjukkan contoh dilatasi dari suatu matriks pada segitiga. Suatu segitiga HIJ dengan titik titik sudut yang berkedudukan pada  $H(a,b)$ ,  $I(c,d)$ ,  $J(e,f)$  dikenai operasi dilatasi oleh matriks koefisien  $z$ . Menghasilkan segitiga baru PQR sesuai dengan persamaan tersebut (Boyd, 2005).

$$\Delta PQR = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix} \cdot [z] = \begin{bmatrix} za & zc & ze \\ zb & zd & zf \end{bmatrix} \quad (10)$$

### 3. Model Warna

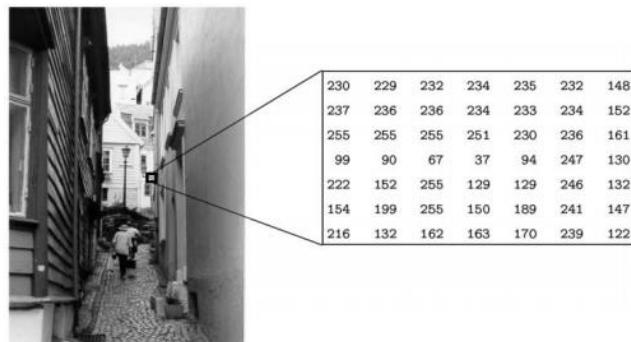
Model warna RGB sering disebut juga dengan model warna aditif (Wahana Komputer, 2005:37). Model warna ini digunakan oleh semua alat perekam visual seperti scanner, digital kamera, serta alat bantu visual elektronik lainnya seperti televisi, monitor, dan proyektor. Model warna RGB merupakan perpaduan atau kombinasi dari tiga warna yaitu merah, hijau, dan biru (Mcandraw, 2004:15 - 21).

Contoh citra RGB ditunjukkan pada Gambar 7.



Gambar 7. Citra Bertipe RGB

Setiap posisi pada citra digital bertipe RGB dapat dinyatakan dalam  $(i, j, k)$ , dengan  $i$  untuk vektor warna merah,  $j$  adalah vektor warna hijau, dan  $k$  adalah vektor warna biru. Nilai masing-masing vektor berada pada rentang 0 sampai  $2^8 - 1 = 255$ . Model warna RGB yang dihasilkan oleh alat perekaman diubah ke dalam aras keabuan atau *grayscale*, yakni citra yang hanya menggunakan tingkatan warna abu-abu. Warna abu-abu adalah warna pada ruang RGB yang memiliki intensitas merah, hijau, dan biru sama (Wahana Komputer, 2005:38).



Gambar 8. Citra Bertipe Grayscale

Gambar 8 (Mcandraw 2004:14) menunjukkan contoh dari citra bertipe *grayscale*. Intensitas citra beraras keabuan disimpan sebagai integer 8 bit sehingga memberikan sebanyak  $2^8 = 256$  tingkat keabuan dari warna hitam sampai warna putih. Citra *grayscale* menggunakan pola 8 bit atau tepatnya 1 *byte*, sehingga membutuhkan ruang memori sedikit daripada citra berwarna (RGB).

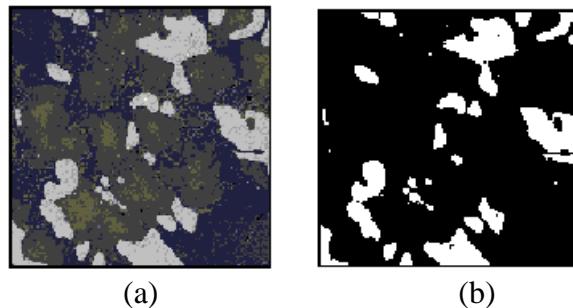
Konversi warna RGB menjadi *grayscale*, dilakukan dengan menghitung Intensitas warna dengan rata-rata komponen warna merah, hijau dan biru (Sutoyo, 2009:32) seperti ditunjukkan pada persamaan 11.

$$F_i(x, y) = \frac{f_i^R(x, y) + f_i^G(x, y) + f_i^B(x, y)}{3} \quad (11)$$

#### 4. *Thresholding*

Analisa segmentasi melalui transformasi warna citra adalah *thresholding*.

*Thresholding* membahas teknik partisi wilayah citra berdasarkan rentang warna, dan dikenal sebagai cara pemisahan object terhadap *background* citra (Gonzales, 2002 :760). Contoh *thresholding* ditunjukkan Gambar 9 (National Instrument).



Gambar 9. Thresholding (a) Citra Asli (b) Hasil Thresholding

Daerah hasil segmentasi *thresholding* didefinisikan sebagai rentang warna antara  $T_1$  dan  $T_2$  atau  $[T_1, T_2]$ . Variabel a, dan c merupakan selang warna di luar rentang warna. nilai piksel yang berada di luar wilayah segmentasi diubah menjadi gelap [0] atau terang [255]. Rentang *thresholding* ditunjukkan sebagai nilai b, yang didefinisikan sesuai persamaan 12.

$$g(x, y) = \begin{cases} a, & \text{if } f(x, y) > T_2 \\ b, & \text{if } T_1 < f(x, y) \leq T_2 \\ c, & \text{if } f(x, y) \leq T_1 \end{cases} \quad (12)$$

*Thresholding* menghasilkan citra biner berupa skala abu-abu atau skala warna dengan menetapkan nilai-nilai piksel pada rentang warna yang diinginkan. Nilai rentang tersebut tergantung pada apakah mereka di atas atau di bawah nilai ambang batas. Umumnya digunakan untuk memisahkan atau *segmentasi* daerah atau objek dalam gambar berdasarkan nilai piksel (Solomon, 2011 :56).

## 5. *Blob Detection*

*Blob detection* merupakan teknik pendekripsi suatu wilayah yang memiliki nilai puncak, baik maksimum atau minimum, baik cerah atau gelap. Blob didefinisikan sebagai wilayah yang mewakili pasangan daerah yang memiliki satu titik lembah dengan satu titik puncak (Lindeberg, 1993).

Definisi lain menggambarkan *blob detection* sebagai suatu persegi panjang yang memiliki wilayah homogen dengan nilai *contrast* tetap, dan dapat berubah menjadi maksimum berdasarkan jumlah penskalaan tertentu (Hinz, 2005). Contoh *blob detection* ditunjukkan pada Gambar 10.



Gambar 10. *Blob Detection* Pada Citra Tangan

*Blob detection* mampu memberikan representasi berupa deteksi wilayah objek hasil *thresholding*. Suatu *blob* dapat dibentuk menggunakan *connected-component labeling*. Menurut Marr dan Hildreth (1980 : 121, 122) Tujuan pensketsaan dari suatu wilayah *thresholding* menunjukkan data citra dalam hubungan terhadap deteksi tepi, *blob*, dan koneksi antar titik yang didekati menggunakan metode *laplacian* maupun *gaussian*. Wilayah *blob* dibentuk dengan memberikan nilai (vektor) naik ke tiap tepi atau sudut (Lindeberg, 2013).

Selisih turunan pada operator laplacian antar wilayah objek dapat didefinisikan sebagai Laplacian operator ditunjukkan pada persamaan 13.

$$\nabla^2 L = L_{xx} + L_{yy} = \lambda_1 + \lambda_2 \quad (13)$$

Deteksi objek merupakan penjumlahan dari matrix  $\lambda_1$  dan matrix  $\lambda_2$ . Maka secara spesifik dapat dianggap sebagai titik titik puncak dari  $\nabla^2 L$  di mana,

$$\nabla^2 L > 0 \text{ (bernilai positif)} \Rightarrow \text{dark blob}$$

$$\nabla^2 L < 0 \text{ (bernilai negatif)} \Rightarrow \text{bright blob}$$

Deteksi otomatis *blob* dari himpunan citra merupakan tahap penting dalam analisis data ilmiah skala besar. Suatu *blob* telah dimanfaatkan dalam penentuan inti suatu organisasi dalam himpunan yang berkaitan, daerah homogen dalam data geofisika, Lokasi tumor dalam data citra MRI atau CT, dll (Kaspers, 2011).

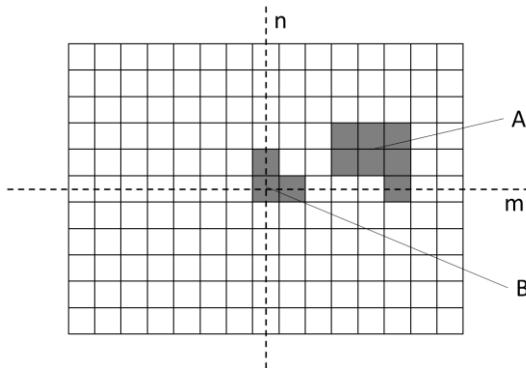
## C. Analisa Citra

Analisa citra merupakan teknik pengenalan bentuk geometri pada pola dua dimensi yang dapat dianalisis dalam diagram Kartesius. Operasi analisa citra meliputi morfologi matematika, deteksi tepi, dan Algoritma SUSAN.

### 1. Morfologi Matematika

Pengenalan bentuk objek berhubungan dengan pengklasifikasian model warna pada suatu ansumsi warna sebagai *foreground* atau yang diteliti, sedangkan yang lainnya sebagai *background*. Sebelum melakukan perbaikan citra, model warna dapat dianalisa dengan tiga pendekatan berbeda (Busch, 1995). Morfologi mempresentasikan citra objek dua dimensi sebagai suatu himpunan matematika

dalam ruang *Euclides* (Ardisasmita, 2000). Operasi morfologi mampu menemukan dan mengenali bentuk objek, sehingga dapat dilakukan teknik pelebaran, perluasan, meletakkan bentuk secara spesifik, dan sebagainya (Thyssen, 2010). Gambar 11 (Putra, 2010) menunjukkan contoh dari morfologi matematika



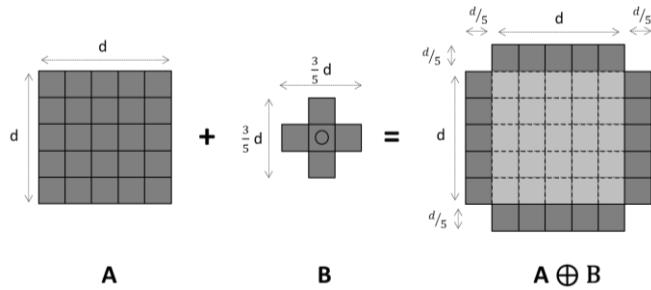
Gambar 11. Citra Biner Yang Terdiri Atas Dua Himpunan Objek A Dan B

Operasi dasar dalam morfologi matematika yaitu dilasi dan erosi (Putra, 2010:187). Dilasi merupakan operasi memperbesar atau mempertebal objek dalam suatu citra biner berdasarkan pola bentuk dilasi yang digunakan pada klasifikasi warna tertentu (Thyssen 2010). Metode dilasi mengatur tingkatan intensitas warna yang dikendalikan oleh suatu bentuk (Prasetyo, 2011:139).

Suatu citra *input* dinyatakan sebagai himpunan A, dan pola bentuk dilasi yang digunakan dinyatakan sebagai himpunan B. Himpunan  $B_x$  menyatakan hasil dilasi A dengan himpunan B. Operasi dilasi dapat dinyatakan sesuai persamaan 14 (Putra, 2010:187 dan Prasetyo, 2011:138).

$$A \oplus B = \{x \mid B_x \cap A \neq \emptyset\} = \{x \mid [B_x \cap A] \subseteq A\} \quad (14)$$

Gambar 12 menunjukkan operasi dilasi pada semua titik dalam wilayah seleksi A oleh B (Prasetyo, 2011:139).

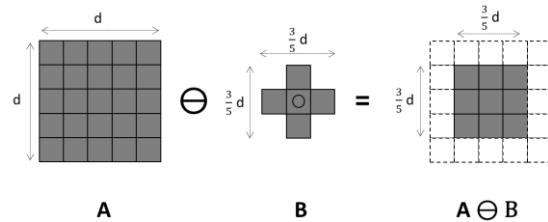


Gambar 12. Objek dan Hasil Dilasi Pada Morfologi Matematika

Operasi dasar selanjutnya adalah erosi. Erosi merupakan proses membentuk setiap piksel citra *input* pada pola bentuk erosi yang digunakan dan melapiskannya, pusat bentuk erosi tersebut tepat dengan posisi piksel citra yang diproses (Thyssen 2010). Operasi erosi dinyatakan sesuai persamaan 15 (Putra, 2010:190-191 dan Prasetyo, 2011:145).

$$A \ominus B = \{x \mid B_x \cap A^c \neq \emptyset\} = \{x \mid B_x \subseteq A\} \quad (15)$$

Gambar 13 menunjukkan operasi erosi pada semua titik dalam wilayah seleksi A oleh B (Prasetyo, 2011:139).



Gambar 13. Objek dan Hasil Erosi Pada Morfologi Matematika

## 2. Edge Detection

*Edge detection* atau deteksi garis tepi adalah pendekatan untuk mendekripsi diskontinuitas *gray level* (Prahasty, 2008). Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan intensitas warna

yang tinggi dengan tetangganya. Tepi dapat didefinisikan sebagai batas antara dua *region* yang memiliki *gray level* yang relatif berbeda (Thyssen, 2010). Diagram 1 menunjukkan ilustrasi deteksi tepi perubahan intensitas warna (Munir, 2002).

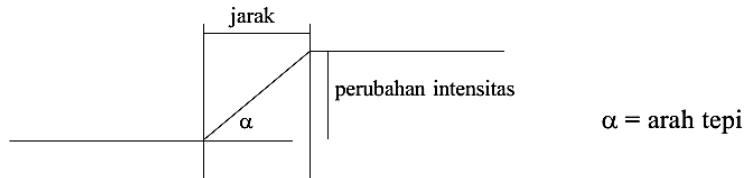


Diagram 1. Ilustrasi Pendekatan Tepi Pada Perubahan Intensitas

### 3. Susan Corner Detection

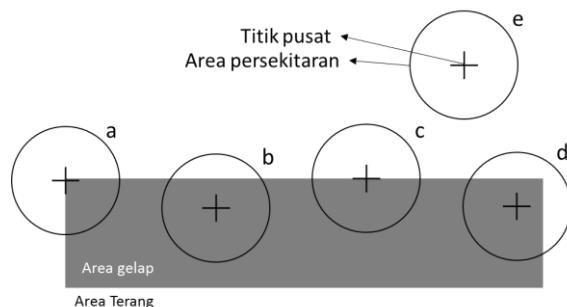
Pendekatan paling sederhana dalam menentukan sudut tepi suatu citra dapat menggunakan prinsip perbedaan area kecerahan berdasarkan jumlah kordinat ketetanggaan. Salah satu metode yang dapat digunakan adalah SUSAN (*Smallest Unvalue Assimilating Nucleus*). SUSAN *corner detector* merupakan pendekatan secara komputasi titik-titik pada area  $N_{xy}$  yang memiliki tingkat kecerahan yang sama dengan titik pusat  $(x, y)$  (Laganière, 1998), sesuai dengan persamaan 16. dengan  $t$  merupakan parameter kontrol terhadap tingkat sensitifitas *noise*.

$$n(x, y) = \sum_{(i, j) \in N_{xy}} e^{-\left(\frac{I(i, j) - I(x, y)}{t}\right)^6} \quad (16)$$

Nilai dari  $n(x, y)$  pada persamaan 16, dilakukan pengecekan terhadap nilai tengah data piksel atau  $\frac{n_{max}}{2}$ , di mana  $n_{max}$  merupakan nilai maksimum, sehingga dapat membedakan wilayah objek dengan *background*, sesuai persamaan 17.

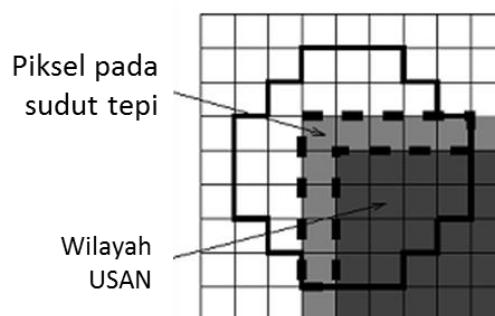
$$c_s(x, y) = \begin{cases} \frac{n_{max}}{2} - n(x, y), & \text{if } n(x, y) < \frac{n_{max}}{2} \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

Pendeteksian tepi menggunakan pola *circular mask* dengan model lingkaran ukuran  $7 \times 7$  atau sebanyak 37 piksel. Setiap circular mask ditempatkan pada area tepi citra untuk mendapatkan area USAN (*Univalue Assimilating Nucleus*) (Smith, 1995). Gambar 14 (Smith, 1995) menunjukkan pengaturan posisi *circular mask* pada area yang terdeteksi.



Gambar 14. Perbedaan Posisi Piksel Pada Area Pelapisan

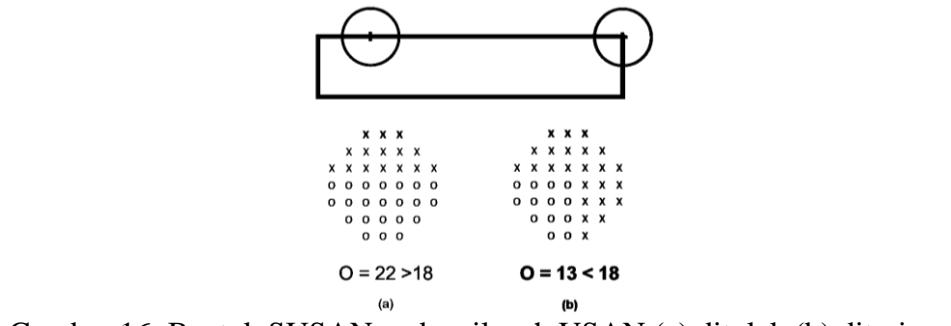
USAN merupakan keseluruhan piksel yang memiliki tingkat kecerahan sama dengan piksel pada nucleus atau titik pusat *circular mask* (Gumilang, 2011). USAN memiliki wilayah lebih kecil saat berada pada wilayah sudut dan hasil reduksi lebih kuat pada tepi sudut. SUSAN dapat digunakan pada dua garis dan deteksi tepi (Smith, 1995). Contoh aplikasi SUSAN sederhana ditunjukkan pada Gambar 15.



Gambar 15. Aplikasi Deteksi SUSAN Dalam Piksel Citra

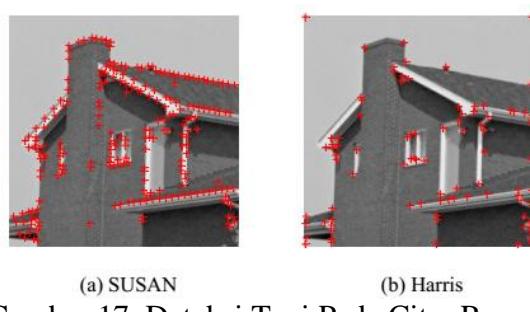
Pada contoh Gambar 16 menggunakan *circular mask* sebanyak 37 piksel.

Suatu *circular mask* yang memiliki sebanyak 37 piksel untuk dapat dideteksi sebagai sudut tepi citra, maka jumlah wilayah USAN memenuhi kurang dari setengah dari jumlah tersebut, atau kurang dari sebanyak 18 piksel.



Gambar 16. Bentuk SUSAN pada wilayah USAN (a) ditolak (b) diterima

Metode *Smallest Unvalue Assimilating Nucleus* (SUSAN) dibandingkan dengan algoritma Harris, menunjukkan nucleus yang terdeteksi tidak terpengaruh pada sudut tajam. Hal tersebut dipengaruhi teknik pengujian SUSAN yang melakukan pengujian pada seluruh area persekitaran. Gambar 17 (Chen, 2009) menunjukkan perbandingan hasil dari pendekripsi SUSAN dengan Harris. Algoritma pendekripsi yang ditunjukkan dalam Harris memiliki titik terdeteksi sebagai sudut tepi lebih sedikit dibanding SUSAN, sehingga metode SUSAN digunakan dalam pengujian, sebab dalam pengujian memerlukan analisa kemungkinan pada tiap titik sudut yang menghasilkan panjang maksimal.



Gambar 17. Deteksi Tepi Pada Citra Rumah

#### 4. Nilai *Thresholding* SUSAN

SUSAN dipengaruhi 2 jenis *thresholding*, yakni *difference Thresholding*, dan *geometrical thresholding*. *Difference thresholding* dipengaruhi oleh tingkat kecerahan atau *brightness* suatu citra. Perbedaan kecerahan mengontrol nilai masukan piksel yang temasuk dalam wilayah USAN. Jika nilai piksel pada *circular mask* tidak lebih besar dari nilai yang ditentukan terhadap piksel titik pusat, maka menjadi bagian dari wilayah USAN (Kirillov, 2006). Pada rentang warna [0, 255]. Nilai *thresholding* yang mendekati nilai 0 akan memiliki menghasilkan lebih banyak SUSAN yang terdeteksi dibanding dengan nilai thresholding mendekati 255.

*Geometrical thresholding* merupakan nilai maksimum himpunan piksel yang menjadi wilayah USAN di suatu *circular mask*. Jika suatu USAN berpotensi memiliki sudut tepi tetapi memiliki nilai lebih besar dari batas nilai yang ditentukan, maka bukan sudut tepi (Kirillov, 2006).

Pada bentuk *circular mask* 7 x 7 atau sebanyak 37 piksel, maka suatu SUSAN atau sudut tepi dapat dideteksi apabila bernilai tidak lebih dari dari setengah *circular mask*, atau maksimal sebanyak 18 piksel. Pengurangan nilai tersebut, maka menunjukkan citra yang memiliki sudut tepi tajam atau citra yang memiliki tingkat ketajaman gambar, sedangkan penambahan nilai tersebut, menyebabkan sudut tidak mudah dideteksi kecuali pada perpotongan antara 2 sisi.

*Geometrical thresholding* memiliki nilai *default* sebesar 18.

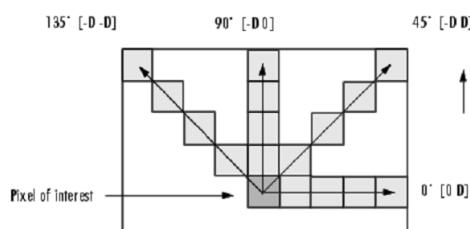
## D. Pengenalan Pola

Pola adalah suatu entitas terdefinisi berupa kumpulan informasi hasil pengukuran atau pemantauan yang dinyatakan dalam notasi vektor atau matrik. Pengenalan pola (*pattern recognition*) adalah teknik pengklasifikasian suatu pola citra berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek (Putra, 2010:303). Metode pengenalan pola yang digunakan adalah *neurofuzzy* yakni penggabungan antara logika *fuzzy*, dengan model *neural network* berdasarkan beberapa parameter statistika yang ditentukan.

### 1. Ekstraksi Citra

Ekstraksi Citra merupakan proses pengambilan informasi dari sebuah citra. Teknik pengambilan informasi menggunakan metode pembentukan matriks co-occurrence pada citra keabuan atau GLCM.

Metode GLCM diawali dengan menentukan relasi ketetanggaan antar piksel yang digunakan. Pendekatan paling sederhana dalam menentukan relasi ketetanggaan dengan cara melihat pola piksel pada sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . sesuai Gambar 18 (MathWork, 2005).



Gambar 18. Derajat Tepi Pada Penentuan Relasi Ketetanggaan

Pola piksel pada sudut tersebut dapat dipengaruhi oleh jarak relasi. Pada jarak relasi  $r$  menunjukkan pola relasi ketetanggaan sesuai dengan persamaan 18.

1.  $Pb_{0,r}(x_1, x_2) = |\{(j, k), (m, n) \in R \mid j - m = 0; |k - n| = r; f(j, k) = x_1, f(m, n) = x_2\}|$
2.  $Pb_{45,r}(x_1, x_2) = |\{(j, k), (m, n) \in R \mid (j - m = r; |k - n| = -r) \text{ or } (j - m = -r; k - n = r); f(j, k) = x_1, f(m, n) = x_2\}|$
3.  $Pb_{90,r}(x_1, x_2) = |\{(j, k), (m, n) \in R \mid |j - m| = r; k - n = 0; f(j, k) = x_1, f(m, n) = x_2\}|$
4.  $Pb_{135,r}(x_1, x_2) = |\{(j, k), (m, n) \in R \mid (j - m = r; k - n = r) \text{ or } (j - m = -r; k - n = r); f(j, k) = x_1, f(m, n) = x_2\}|$

(18)

Matrik *Co-occurrence* dibangun menggunakan histogram tingkat kedua berukuran  $L \times L$  ( $L$  menyatakan banyaknya tingkat keabuan) dengan elemen-elemen  $Pb_{\emptyset,r}(x_1, x_2)$  yang merupakan distribusi probabilitas bersama (*joint probability distribution*). dari pasangan piksel dengan tingkat keabuan piksel  $x_1$  yang berlokasi pada koordinat  $(j,k)$  dengan piksel tetangga  $x_2$  yang berlokasi pada koordinat  $(m,n)$  (Permatasari, 2012). Histogram tingkat kedua  $p(i,j)$  dihitung dengan pendekatan pada persamaan 19.

$$p(i,j) = \frac{V(i,j)}{\sum_{i,j=0}^{n-1} V(i,j)} \quad (19)$$

Tahapan pembentukan matriks *co-occurrence* (Mathwork, 2005) sebagai berikut,

1) Membentuk Frame work

Frame work yang terbentuk akan memiliki ukuran  $256 \times 256$  karena citra grayscale diatas memiliki 256 nilai intensitas. yaitu  $[0, 255]$ .

2) Membentuk Matriks *co-occurrence* dari matriks asal

Matrik baru dibuat menggunakan elemen nol dengan dimensi sebesar nilai maksimal pada matrik asal. Rentang maksimal nilai keabuan yang digunakan sebesar 255. Matriks *co-occurrence* didefinisikan dengan mencari jumlah pasaran berurutan pada relasi ketetanggaannya. Ilustrasi perhitungan matriks *co-occurrence* sesuai Gambar 19 (MathWork, 2005).

	1	2	3	4	5	6	7	8
1	1	1	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Gambar 19. Ilustrasi Pembentukan Matriks Co-Occurrence

3) Mentranspose matriks,

Matriks *co-occurrence* yang sudah didapat selanjutnya dilakukan transpose

4) Menjumlahkan matriks cooccurrence dengan matriks *transpose*.

5) Matriks baru perlu dilakukan penjumlahan untuk diubah menjadi simetris.

## 2. Parameter Statistika

Berdasarkan matriks *co-occurrence* yang telah terbentuk, pola dapat dikenali sebagai tekstur melalui perhitungan parameter statistika. Parameter statistika menggunakan pola tekstur yang dikemukakan oleh Robert M.Haralick (1973) dan dikembangkan oleh Fritz Albregtsen (2008) yakni,

a. *Energy*

*Energy* disebut juga *Angular Second Moment* (ASM) menunjukkan nilai yang tertinggi saat piksel-piksel gambar homogen (Anami, 2009). Rumus untuk mencari *energy* dari suatu gambar ditunjukkan pada persamaan 20.

$$ASM = \sum_i \sum_j (p(i,j))^2 \quad (20)$$

b. *Contrast*

*Contrast* adalah ukuran variasi antar derajat keabuan suatu daerah citra (Putra, 2013). Persamaan 21 menunjukkan parameter *contrast*.

$$Con = \sum_{n=0}^{N_g-1} n^2 \left( \sum_{\substack{i=1 \\ |i-j|=n}}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right) \quad (21)$$

c. *Correlation*

*Correlation* adalah ukuran tingkat abu-abu ketergantungan linier antara piksel padaposisi tertentu terhadap piksel lain. Rumus untuk menentukan *correlation* dari suatu gambar (Soh & Tsatsoulis, 1999: 781) ditunjukkan pada persamaan 22.

$$Cor = \frac{\sum_i \sum_j (i,j) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (22)$$

d. *Variance (Sum Of Square)*

*Variance* menunjukkan seberapa banyak tingkat keabu-abuan yang beragam dari rata-rata (Vannier, 1985). Rumus menghitung *variance* dari suatu gambar ditunjukkan pada persamaan 23.

$$Var = \sum_i \sum_j (i - \mu_{(i,j)})^2 p(i,j) \quad (23)$$

e. *Inverse Difference Moment (IDM)*

*Inverse Difference Moment (IDM)* menunjukkan kesamaan piksel. Rumus IDM suatu gambar (Yin, 2004) ditunjukkan pada persamaan 24.

$$IDM = \sum_i \sum_j \frac{p(i,j)}{1 + (i-j)^2} \quad (24)$$

f. *Sum Average*

*Sum Average* dirumuskan sesuai dengan persamaan 25.

$$SA = \sum_{i=2}^{2N_g} i \cdot p_{x+y}(i) \quad (25)$$

g. *Sum Variance*

*Sum Variance* dirumuskan sesuai dengan persamaan 26.

$$SV = \sum_{i=2}^{2N_g} \left( i + \sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i)) \right)^2 p_{x+y}(i) \quad (26)$$

h. *Sum Entropy*

*Sum Entropy* dirumuskan sesuai dengan persamaan 27,

$$SE = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i)) \quad (27)$$

i. *Entropy*

Nilai *entropy* digunakan untuk menghitung keacakan intensitas gambar (Cline, 1990:1037- 1045). Rumusan entropy ditunjukkan pada persamaan 28.

$$Ent = - \sum_i \sum_j p(i,j) \log(p(i,j)) \quad (28)$$

j. *Difference Variance*

*Difference Variance* disimbolkan sebagai DV dan dirumuskan sebagai varian dari  $p_{x-y}(k)$ .

k. *Difference Entropy*

*Difference Entropy* dirumuskan sesuai dengan persamaan 29.

$$DE = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log(p_{x-y}(i)) \quad (29)$$

l. *Information Measure of Correlation*

*Information Measure of Correlation* terdiri dari 2 teknik pengujian, yakni *First Information Measure of Correlation* dan *Second Information Measure of Correlation* (Albregtsen, 2008). *First Information Measure of Correlation*

ditunjukkan pada persamaan 30, sedangkan *Second Information Measure of Correlation* ditunjukkan pada persamaan 31.

$$MF = \frac{-\sum_i \sum_j p(i,j) \log(p(i,j)) - \sum_i \sum_j p(i,j) \log(p_x(i)p_y(j))}{\max\{H_X, H_Y\}} \quad (30)$$

$$MS = \sqrt{1 - e^{-2.0(-\sum_i \sum_j p_x(i)p_y(j) \log(p_x(i)p_y(j)) + \sum_i \sum_j p(i,j) \log(p(i,j)))}} \quad (31)$$

m. *Homogeneity*

*Homogeneity* menunjukkan nilai distribusi antara elemen. Rumus untuk mencari *homogeneity* dari suatu gambar ditunjukkan persamaan 32 (Zhu, 2006).

$$H = \sum_i \sum_j \frac{p(i,j)}{1 + |i-j|} \quad (32)$$

n. *Cluster Prominence*

*Cluster prominence* dirumuskan sesuai dengan persamaan 33.

$$CP = \sum_i \sum_j [(i - \mu_x) + (j - \mu_y)]^4 \cdot p(i,j) \quad (33)$$

o. *Cluster Shade*

*Cluster shade* dirumuskan sesuai dengan persamaan 34.

$$CS = \sum_i \sum_j [(i - \mu_x) + (j - \mu_y)]^3 \cdot p(i,j) \quad (34)$$

### 3. Jaringan Syaraf Tiruan

Pemodelan jaringan syaraf tiruan atau *artificial neural network* merupakan pemodelan dengan menggunakan pendekatan model *black box*. Prinsip kerja *neural network* didasari pada mekanisme kerja penyaluran informasi. Penyaluran informasi pada *neural network* meniru prinsip kerja dari *neuron* otak manusia

dengan keterbatasan pada jumlah jaringan yang digunakan (Putra, 2010: 246). *neural network* berkembang pada pengembangan dibidang klasifikasi, *time series*, Pendekatan *big data*, peramalan, *recognition*, pengelompokan sesuai tingkat, simulasi uang, dan lain sebagainya yang berhubungan dengan data yang berkembang.

Pembentukan *neural network* ditentukan oleh 3 hal (Fausett, 1997: 3), yaitu, a). Pola hubungan antar *neuron*, b). Bobot penghubung, dan c).Fungsi Aktivasi. Konsep *neural network* adalah terjadinya proses pembelajaran. Tujuan utama dari proses pembelajaran adalah melakukan pengaturan terhadap bobot sehingga diperoleh bobot akhir yang sesuai dengan pola data yang dilatih (Kusumadewi, 2010: 84).

### **a. Model Pembelajaran**

Pembelajaran pada *neural network* terdapat 2 macam, terawasi atau *supervised*, dan tidak terawasi atau *unsupervised*. Pembelajaran *supervised* dilakukan apabila *output* yang diharapkan telah diketahui sebelumnya. Pada proses pembelajaran, satu pola *input* akan diberikan ke suatu *neuron* pada lapisan *input*. Selanjutnya pola akan dirambatkan pada sepanjang jaringan syaraf sampai ke *neuron* pada lapisan *output*. Lapisan *output* akan membangkitkan pola *output* yang akan dicocokan dengan pola *output* targetnya (Machfudhoh, 2014).

### **b. Arsitektur *Neural network***

Arsitektur *neural network* merupakan susunan jaringan yang memiliki nilai *input – output* dan berasal dari pengambilan atau pengubahan nilai beberapa

parameter (Jang, 1997: 200). Arsitektur *neural network* berhubungan erat dengan Pola hubungan antar *neuron* yang dikembangkan. Beberapa bentuk arsitektur jaringan antara lain *single layer Perceptron*, *multilayer Perceptron*, dan *Recurrent*. Arsitektur jaringan MLP ditunjukkan Diagram 2 (Jang, 1997).

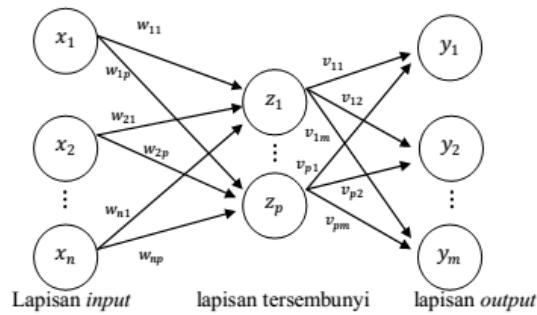


Diagram 2. Arsitektur Jaringan Multilayer *Feedforward*

Jaringan *multilayer Perceptron* atau MLP dikembangkan dari *single layer Perceptron* dengan menambah satu atau lebih lapisan tersembunyi, perhitungan hasil dapat disebut sebagai *neuron* tersembunyi. Fungsi *neuron* tersembunyi adalah penghubung antara luaran *input* dan jaringan *output* (Haykin, 1999: 43).

### c. Fungsi Aktivasi *Sigmoid*

Fungsi aktivasi merupakan fungsi yang digunakan untuk membangkitkan *neuron output* untuk dikirim sebagai *input* suatu *neuron* lain (Fausett, 1994: 17). Diagram alur fungsi aktivasi ditunjukkan pada Diagram 3 (Jang, 1997).

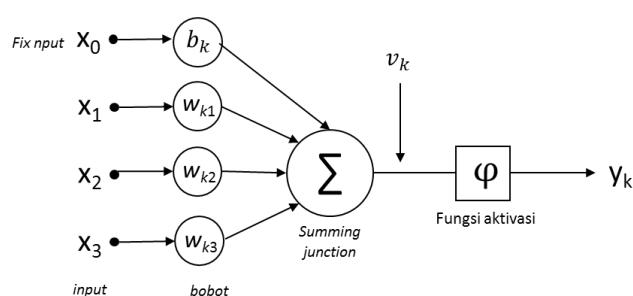


Diagram 3. Fungsi Aktivasi Pada *Nonlinear Single-Node*

Fungsi aktivasi didefinisikan sebagai  $\varphi(v)$ . Terdapat 3 macam fungsi aktivasi, yakni *threshold function*, *sigmoid function*, dan *bipolar sigmoid function* (Kirillov, 2006). Diagram 4 menunjukkan perbedaan bentuk *output* yang dihasilkan dari masing masing fungsi aktivasi (Haykin, 1999:35).

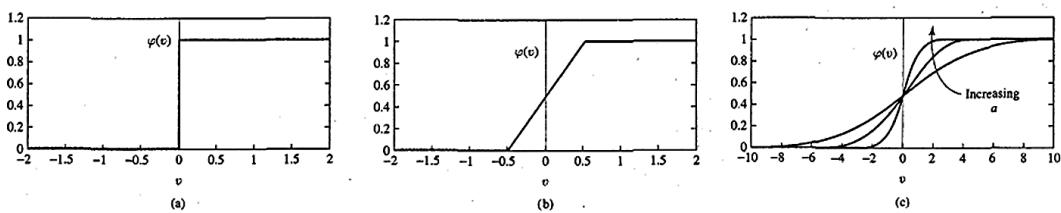


Diagram 4. Fungsi Aktivasi (a) *Threshold*, (b) *Piecewise Linear*, (c) *Sigmoid*

*Input*  $x_1, x_2, \dots, x_n$  masing-masing memiliki bobot  $(w_{k1}, w_{k2}, \dots, w_{kn})$  dan bobot bias  $b_k$  yang berasal dari *fix input*. *Neuron* akan mengolah nilai *input* sesuai persamaan 35 (Haykin, 1999:35).

$$v_k = \sum_{i=1}^n x_i w_{ki} + b_k \quad (35)$$

Fungsi aktivasi *sigmoid* akan mengaktifasi  $v_k$  menjadi *output* jaringan  $y_k$ . Fungsi *sigmoid* secara umum ditunjukkan pada persamaan 36 (Haykin, 1999:35).

$$y_k = \varphi(v) = \frac{1}{1+e^{-av}} \quad (36)$$

Perbedaan *sigmoid function* dan *bipolar sigmoid function* yakni sesuai interval dari luaran *neuron*. Luaran *neuron* pada *sigmoid function* menghasilkan nilai interval  $[0,1]$ . Fungsi *sigmoid* ini memiliki tingkat kecenderungan yang dapat dipergunakan pada bentuk *nonlinear*. Pada interval tersebut maka variabel  $a = 1$  (Fausset, 1994 :293). Nilai  $v_k$  pada persamaan 35 disubtitusikan ke dalam persamaan 36 dengan 1 layer tersembunyi pada jaringan *multilayer Perceptron*

dapat dirumuskan secara matematis fungsi aktivasi *sigmoid* sesuai persamaan 37 (Machfudhoh, 2014).

$$y_k = \frac{1}{1 + \exp\left(-\left[\sum_{j=1}^p w_{jk} \cdot \frac{1}{1 + \exp(-[v_{0j} + \sum_{i=1}^n x_i v_{ij}])} + w_{0k}\right]\right)} \quad (37)$$

#### **d. Algoritma Pembelajaran *Backpropagation***

Algoritma *backpropagation* merupakan algoritma pembelajaran pada data *training* dalam membentuk model. Model terbaik ditentukan berdasarkan banyak *neuron* terbaik pada lapisan tersembunyi dan *input* yang optimal. *Input* yang optimal adalah *input* yang menghasilkan nilai akurasi tertinggi dari proses pembelajaran menggunakan algoritma *backpropagation*.

Akurasi dihitung sesuai hasil *trial and error* (Fausset, 1994: 294-296). Nilai Error ditentukan menggunakan besar nilai RMSE. Nilai RMSE atau *Mean Square Error* merupakan tingkat kesalahan yang terjadi saat model pembelajaran tidak berhasil mendekripsi nilai *input* yang digunakan. Besar nilai RMSE dihitung berdasarkan jumlah data dalam matriks *input* yang tidak terdeteksi dibagi jumlah seluruh data *input* yang digunakan.

Misalkan  $Q$  *layer* tersembunyi,  $q = 1, 2, \dots, Q$ ,  ${}^q\text{net}_i$  dan  ${}^q\text{y}_i$  merupakan *input* dan *output neuron* ke  $i$  pada *layer* ke  $q$ , jaringan memiliki  $m$  *neuron input* dan  $n$  *neuron output*, serta  ${}^q\text{w}_{ij}$  sebagai bobot jaringan dari  ${}^{q-1}\text{y}_j$  ke  ${}^q\text{y}_i$ . Algoritma *backpropagation* untuk jaringan *feedforward* dengan fungsi aktivasi *sigmoid* biner adalah sebagai berikut (Lin, 1996 : 240 dan Fausset, 1994:294-296).

*Input* : Himpunan data *training*  $\{(x_k, d_k) \mid k = 1, 2, \dots, p\}$ , di mana *fix input* ditambahkan pada *neuron* terakhir bernilai -1, sehingga  $x_{m+1}^{MSE} = -1$ .

Langkah 0 : [inisialisasi] Memilih  $\eta > 0$  dan  $E_{max}$  (maksimum toleransi kesalahan). Inisiasi bobot bias awal menggunakan nilai random terkecil dengan mengatur *erorr* = 0 dan *MSE* = 1.

Langkah 1 : [perulangan percobaan] Menetapkan parameter pembelajaran sebagai *neuron* sinyal pada *layer input* ( $q = 1$ ),

$${}^q y_i = {}^1 y_i = x_i \quad (38)$$

Langkah 2 : [Forward propagation] Meneruskan setiap sinyal *neuron* ke seluruh jaringan *neuron* yang ada di lapisan selanjutnya. Menjumlahkan sinyal *neuron* dengan sinyal *input* berbobot  ${}^q w_{0j}$  sebagai *neuron* pada lapisan tersembunyi. ( $z_j, j=1, 2, \dots, p$ )

$${}^q net_j = w_{0j} + \sum_{j=1}^{q-1} {}^q y_j {}^q w_{ij} \quad (39)$$

menggunakan fungsi aktivasi untuk menghitung sinyal *input*

$${}^q y_j = \frac{1}{1 + \exp(-{}^q net_j)} \quad (40)$$

Langkah 3 : [Koreksi *erorr*] Menghitung nilai *erorr* dan sinyal *error*  ${}^q \delta_i$  untuk lapisan *output*.

$$E = \frac{1}{2} \sum_{i=1}^n (d_i^{MSE} - {}^q y_i)^2 + E' \quad (41)$$

$$\begin{aligned} {}^q \delta_j &= (d_j^{MSE} - {}^q y_j) f'({}^q net_j) \\ &= (d_j^{MSE} - {}^q y_j) {}^q y_j (1 - {}^q y_j) \end{aligned} \quad (42)$$

Langkah 4 : [Backpropagation]

Langkah 4a : Koreksi bobot *neuron* dan perubahan bobot menuju ke *output*.

$$\Delta^q w_{ij} = ^q\eta \cdot ^{q-1}\delta_i \cdot ^q y_j \quad (43)$$

$$^q w_{ij} (\text{baru}) = ^q w_{ij} (\text{lama}) + \Delta^q w_{ij} \quad (44)$$

Dengan koreksi bias untuk memperbaiki nilai  $w_{0j}$

$$\Delta^q w_{0j} = ^q\eta \cdot ^{q-1}\delta_i \quad (45)$$

Langkah 4b : Perhitungan sinyal error  $^{q-1}\delta_i$  pada lapisan sebelumnya.

$$^{q-1}\delta_{in\_i} = \sum_{j=1}^m ^{q-1}\delta_i \ ^q w_{ij} \quad (46)$$

Faktor  $\delta$  unit tersembunyi

$$\begin{aligned} ^{q-1}\delta_i &= f'(^{q-1}net_j) \cdot ^{q-1}\delta_{in\_i} \\ &= z_j(1 - z_j) \cdot \sum_{j=1}^m ^{q-1}\delta_i \ ^q w_{ij} \end{aligned} \quad (47)$$

Langkah 4c : Koreksi bobot jaringan untuk aktivasi  $v$

$$\Delta v_{ij} = ^q\eta \cdot ^{q-1}\delta_i \cdot ^q x_j \quad (48)$$

$$^q v_{ij} (\text{baru}) = ^q v_{ij} (\text{lama}) + \Delta^q v_{ij} \quad (49)$$

Dengan koreksi bias untuk memperbaiki nilai  $v_{0j}$

$$\Delta^q v_{0j} = ^q\eta \cdot ^{q-1}\delta_i \quad (50)$$

Langkah 5 : [Pengulangan *Epoch*] Pengecekan keseluruhan himpunan data *training* dengan pengembalian. Jika  $MSE < p$ , maka  $MSE = MSE + 1$  dan kembali ke langkah 1. Jika tidak, lanjut ke langkah berikutnya.

Langkah 6 : [Perhitungan total  $erorr$ ] Pengecekan total  $erorr$  pada setiap lapisan dapat diterima, jika  $erorr < erorr_{max}$ , kemudian hitung peningkatan nilai fungsi aktivasi dan seluruh perubahan bobot yang menuju *output*. jika tidak,  $erorr = 0, MSE = 1$ , dan inisisasi epoch data *training* baru dengan kembali ke langkah 1.

Langkah 7 : Selesai.

Pembelajaran *backpropagation* dipengaruhi beberapa faktor pembelajaran seperti *learning rate*, momentum, penggunaan *sigmoid*, jumlah iterasi atau *epoch*, jumlah *neuron*, *initial weight*, nilai *error output*, dan beberapa faktor lain.

- *Learning rate*

*Learning rate* ( $\eta$ ) merupakan faktor penting dalam menghasilkan pembelajaran yang efektif dan konvergen (Lin, 1996:245). Semakin besar *learning rate* mengakibatkan semakin besar langkah pembelajaran sehingga algoritma menjadi tidak stabil (Kusumadewi, 2013:151), seperti ditunjukkan pada persamaan 51.

$$\Delta\eta = \begin{cases} +a, & \text{if } \Delta E < 0 \\ -b\eta, & \text{if } \Delta E > 0 \\ 0, & \text{otherwise} \end{cases} \quad (51)$$

Laju pembelajaran yang tidak konvergen dapat ditentukan berdasarkan rasio pengali. Rasio pengali akan menaikkan *learning rate* apabila *learning constant* terlalu rendah untuk mencapai nilai konvergen. Rasio pengali untuk menurunkan *learning rate* apabila *learning constant* yang ada terlalu tinggi untuk mencapai nilai konvergen (Kusumadewi, 2013:151-152).

- Momentum

Momentum adalah perubahan bobot yang didasarkan atas arah *gradient* pola terakhir dan pola sebelumnya. Nilai momentum antara 0 sampai 1. Jika nilai momentum = 0, maka perubahan bobot hanya akan dipengaruhi oleh gradiennya. Jika nilai momentum = 1, maka perubahan bobot akan sama dengan perubahan bobot sebelumnya (Kusumadewi, 2013:152). Ide pemberian bobot pada tiap momentum dapat meningkatkan kecepatan (Lin, 1996:247). Perubahan bobot yang dapat diberikan ditunjukkan pada persamaan 52.

$$\Delta w(t) = -\eta \nabla E(t) + \alpha \Delta w(t-1) \quad (52)$$

- Maksimum *Epoch* / Iterasi

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi ini akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch*nya (Kusumadewi, 2013:151).

- Penentuan bobot

Penentuan bobot adalah menentukan bobot terbaik yang dapat digunakan pada jaringan *multilayer feedforward*. Pada umumnya besar bobot menggunakan nilai terendah. Kesalahan penentuan besar bobot dapat menghasilkan ketidaksamaan besar bobot pada perhitungan solusi. Besar bobot tidak boleh bernilai besar, rentang bobot dapat diberikan  $[\frac{-3}{\sqrt{MSE}}, \frac{3}{\sqrt{MSE}}]$  (Lin, 1996:245).

- Maksimum Kenaikan Kerja

Maksimum kenaikan kerja adalah nilai maksimum kenaikan *error* yang dibolehkan antara *error* saat ini dan *error* sebelumnya. Nilai *default* untuk

maksimum kenaikan kerja adalah 1,06 (Kusumadewi, 2013:151). Rumus *erorr*  $(d_i - y_i)^2$  dapat dirubah ke persamaan differensial  $F(d_i, y_i)$ . Besar kenaikan kerja sesuai dengan persamaan 53 (Lin, 1996:246).

$$E = \frac{1}{4} \sum (d_i - y_i)^p \quad \text{di mana } 1 \leq p < \infty \quad (53)$$

#### e. Peta Konsep *Neural network*

Algoritma pemrograman mengacu pada peta konsep *neural network* yang sudah disediakan dalam *library* program. seperti ditunjukkan pada Diagram 5.

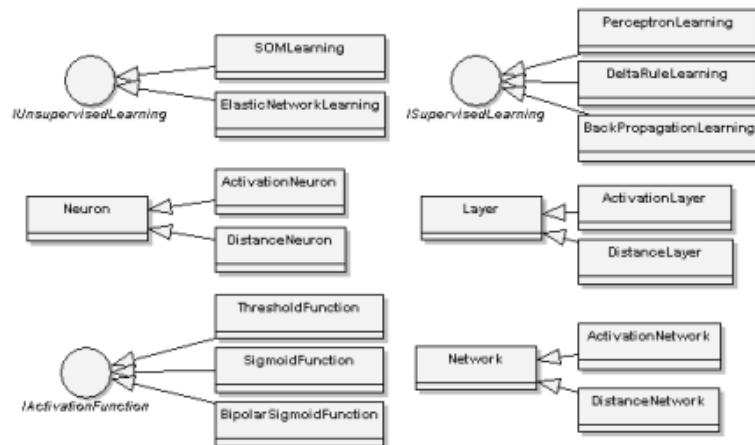


Diagram 5. Konsep *Library Neural network*

Algoritma *neural network* sangat cocok digunakan sebagai pendekatan sebaran data, peramalan atau nilai taksir, dan pengenalan objek (Kirillov, 2006) Metode *Neural network supervised learning* atau pembelajaran terawasi memiliki 3 macam jenis pembelajaran, yakni *perceptron learning*, *delta rule learning*, dan *backpropagation learning*. *Backpropagation learning* merupakan metode yang paling populer untuk algoritma pada multilayer *neural network*. Hasil dari pembelajaran berupa bobot jarak, *neuron*, dan jaringan juga dapat dikenali sebagai *distance layer*, *distance neuron*, dan *distance network*.

#### 4. Logika *Fuzzy*

Logika *fuzzy* merupakan salah satu komponen pembentuk *soft computing* yang bertujuan membuat komputer mampu berfikir seperti manusia (Kusumadewi, 2010). Logika *fuzzy* mampu menentukan ketidakjelasan sifat atau keadaan manusia melalui berbagai pendekatan. *Fuzzy* difungsikan untuk memahami sifat atau keadaan alami manusia seperti panas, dingin, besar, kecil, dan sebagainya (Ibrahim, 2004:12). Diagram 6 menunjukkan Proses Pemodelan *Fuzzy* (Mulato, 2015).

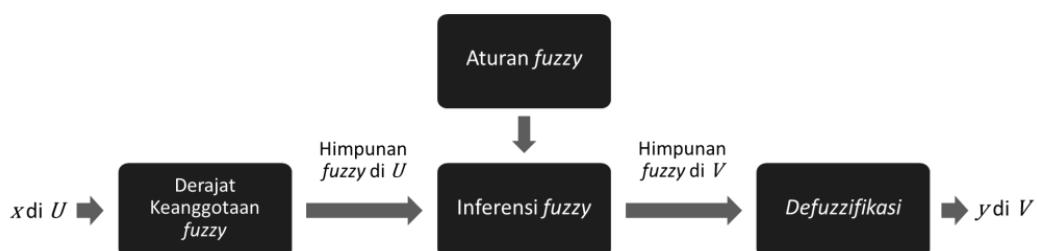


Diagram 6. Proses Pemodelan *Fuzzy*

Pengambilan keputusan dalam sistem *fuzzy* berdasarkan *input* dalam bentuk linguistik. Sifat dan keadaan yang menjadi masalah didekati berdasarkan pendekataan titik klimaks, sehingga nilai yang didapat bukan benar salah akan tetapi sebuah rentang atau selanjutnya disebut juga derajat keanggotaan. Rentang disesuaikan dengan prasyarat *linguistic* berdasarkan aturan logika *fuzzy* dan respon dari setiap aturan *fuzzy* (Lin, 1996:480).

##### a. Fungsi Keanggotaan *Fuzzy*

Logika *fuzzy* berhubungan dengan peranan fungsi keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan matriks piksel atau disebut juga teori himpunan *fuzzy* (Putra, 2010). Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*

(Kusumadewi, 2013:7). Fungsi keanggotaan adalah persamaan kurva yang menunjukkan titik-titik *input* data dalam nilai keanggotaannya atau disebut derajat keanggotaan. Fungsi keanggotaan sebuah himpunan *fuzzy* dinotasikan sebagai  $\mu_k(x)$  yang berada pada interval  $[0,1]$  (Wang, 1997:22) sesuai persamaan 54.

$$\mu_k(x): U \rightarrow [0,1] \quad (54)$$

Nilai keanggotaan pada titik klimaks suatu suku  $x$  dalam suatu logika *fuzzy* memiliki dua kemungkinan (Kusumadewi, 2013:3), yaitu:

- satu (1), yang berarti bahwa suatu suku merupakan anggota himpunan, atau
- nol (0), yang berarti bahwa suatu suku bukan merupakan anggota himpunan.

Pendekatan fungsi yang umum digunakan adalah representasi kurva *sigmoid*. Kurva *sigmoid* berhubungan dengan kenaikan dan penurunan permukaan secara tak linear. Menurut Cox (1994) fungsi keanggotaan representasi kurva *sigmoid* akan bertemu pada 50% nilai keanggotaannya yang sering disebut dengan titik infleksi. Kurva *sigmoid* didefinisikan dengan menggunakan 3 parameter, yaitu: nilai keanggotaan nol ( $\alpha$ ), nilai keanggotaan lengkap ( $\gamma$ ), dan titik infleksi atau *crossover* ( $\beta$ ) yaitu titik yang memiliki domain 50% benar.

Persebaran data derajat keanggotaan sesuai dengan representasi kurva. Terdapat tiga tipe representasi kurva, yakni kurva *sigmoid* naik, kurva *sigmoid* turun, dan kurva lonceng Gauss. Representasi kurva *sigmoid* naik pada Diagram 7.

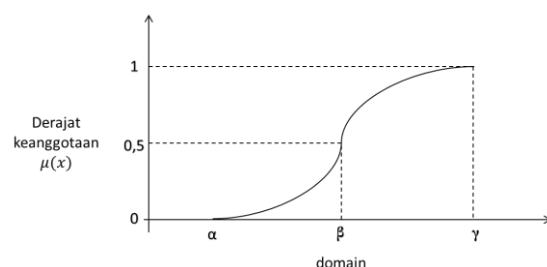


Diagram 7. Representasi Fungsi Kurva *Sigmoid* Naik

Berdasarkan Diagram 7, fungsi keanggotaan dari representasi kurva *sigmoid* naik berada pada rentang  $[\alpha, \gamma]$  dengan titik tumpu berada pada  $\beta$ . Fungsi keanggotaan dari kurva *sigmoid* naik dinyatakan sesuai persamaan 55.

$$\mu[x] = \begin{cases} 0, & x \leq \alpha \\ 2 \left( \frac{x-\alpha}{\gamma-\alpha} \right)^2, & \alpha \leq x \leq \beta \\ 1 - 2 \left( \frac{\gamma-x}{\gamma-\alpha} \right)^2, & \beta \leq x \leq \gamma \\ 1, & x \geq \gamma \end{cases} \quad (55)$$

Representasi kurva *sigmoid* juga dapat berupa fungsi menurun, yang ditunjukkan pada Diagram 8.

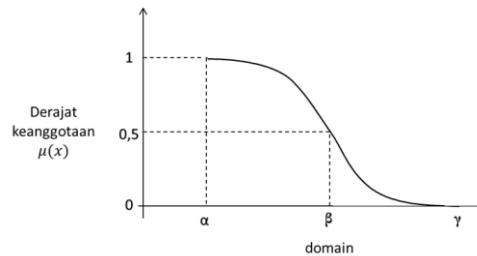


Diagram 8. Representasi Fungsi Kurva *Sigmoid* Turun

Berdasarkan Diagram 8, fungsi keanggotaan dari representasi kurva *sigmoid* turun berada pada rentang  $[\alpha, \gamma]$  dengan titik tumpu berada pada  $\beta$ . Fungsi keanggotaan dari kurva *sigmoid* turun dinyatakan sesuai persamaan 56.

$$\mu[x] = \begin{cases} 0, & x \leq \alpha \\ 2 \left( \frac{x-\alpha}{\gamma-\alpha} \right)^2, & \alpha \leq x \leq \beta \\ 1 - 2 \left( \frac{\gamma-x}{\gamma-\alpha} \right)^2, & \beta \leq x \leq \gamma \\ 1, & x \geq \gamma \end{cases} \quad (56)$$

Persebaran data derajat keanggotaan juga dapat mengalami kenaikan hanya pada titik tumpu ( $\gamma$ ). Derajat keanggotaan yang terbentuk merupakan penggabungan dari kedua kurva *sigmoid*, di mana pada rentang domain  $[\varphi_i, \gamma]$  merupakan kurva *sigmoid* naik, sedangkan pada rentang domain  $[\varphi_j, \gamma]$  merupakan

kurva *sigmoid* turun. Penggabungan dari kedua jenis kurva ini dinamakan kurva lonceng Gauss. Representasi dari kurva tersebut ditunjukkan pada Diagram 9 (Kusumadewi, 2013).

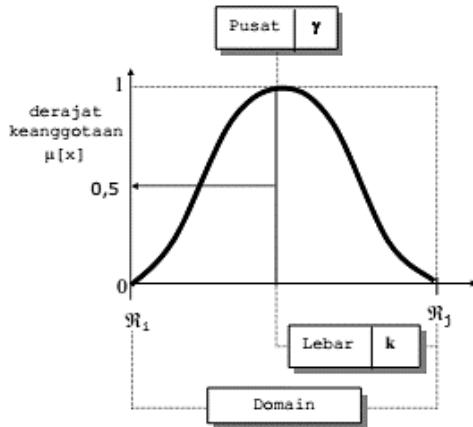


Diagram 9. Representasi Kurva L onceng Gauss

Kurva lonceng gauss menggunakan domain titik tumpu pada ( $\gamma$ ) untuk menunjukkan nilai domain pada pusat kurva, dan ( $k$ ) yang menunjukkan lebar kurva. Fungsi keanggotaan representasi kurva Gauss sesuai dengan persamaan 57.

$$G(x; k, y) = e^{-k(y-x)^2} \quad (57)$$

### b. Operator Himpunan *Fuzzy*

Dasar logika *fuzzy* berhubungan dengan peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan matriks piksel atau disebut juga teori himpunan *fuzzy* (Putra, 2010). Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy* (Kusumadewi, 2013:7). Operasi himpunan *fuzzy* pada fungsi keanggotaan terdiri dari irisan, gabungan, dan komplemen.

Fungsi keanggotaan dari operasi irisan diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan bersangkutan (Kusumadewi, 2013), yakni sesuai persamaan 58.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y)) \quad (58)$$

Representasi operasi irisan fungsi keanggotaan dari himpunan *fuzzy* dapat digambarkan sesuai diagram perpotongan pada Diagram 10 (Ibrahim, 2004).

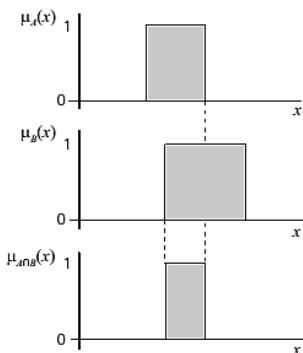


Diagram 10. Operasi Irisan Himpunan *Fuzzy*

Fungsi keanggotaan dari operasi gabungan diperoleh dengan mengambil nilai keanggotaan terbesar antarelemen pada himpunan bersangkutan (Kusumadewi, 2013), yakni sesuai persamaan 59.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y)) \quad (59)$$

Representasi operasi gabungan fungsi keanggotaan dari himpunan *fuzzy* dapat digambarkan sesuai diagram perpotongan pada Diagram 11 (Ibrahim, 2004).

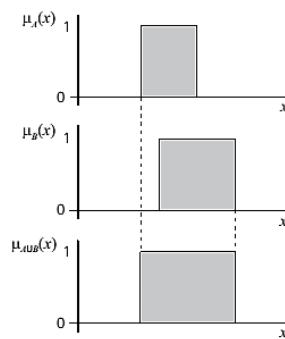


Diagram 11. Operasi Gabungan Himpunan *Fuzzy*

Fungsi keanggotaan dari komplemen A diperoleh dengan mengambil nilai keanggotaan yang tidak berada pada himpunan bersangkutan (Kusumadewi, 2013) yakni sesuai pada persamaan 60.

$$\mu_{\bar{A}} = 1 - \mu_A(x) \quad (60)$$

Representasi operasi komplemen fungsi keanggotaan dari himpunan *fuzzy* dapat digambarkan sesuai diagram perpotongan pada Diagram 12 (Ibrahim, 2004).

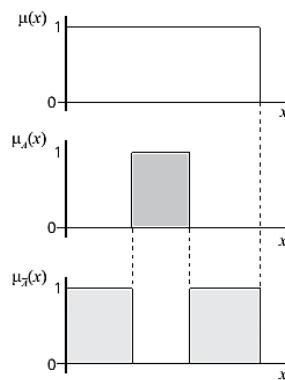


Diagram 12. Operasi Komplemen Himpunan *Fuzzy*

### c. Sistem Inferensi *Fuzzy*

Sistem inferensi *fuzzy* dikenal sebagai dasar pemrograman komputer dalam penerapan teori himpunan *fuzzy*, dan penalaran *fuzzy*. Skema inferensi *fuzzy* pada umumnya menggunakan bentuk *modus ponen* dalam menetapkan aturan terstruktur.

Pada sistem *fuzzy multi-input single-output* dapat ditunjukkan pada aturan IF-THEN dengan  $n$  menunjukkan jumlah *neuron*. Variabel  $x = (x_1, x_2, \dots, x_n)^T$  sebagai *neuron input*, dan variabel  $y$  sebagai *neuron output* ditunjukkan dalam persamaan 61.

Fakta	: $x_1$ is $A'_1$ AND $x_2$ is $A'_2$ AND ... AND $x_n$ is $A'_n$
premis 1	: IF $x_1$ is $A_1^1$ AND $x_2$ is $A_2^1$ AND ... AND $x_n$ is $A_n^1$ , THEN $y$ is $B^1$ ELSE
premis 2	: IF $x_1$ is $A_1^2$ AND $x_2$ is $A_2^2$ AND ... AND $x_n$ is $A_n^2$ , THEN $y$ is $B^2$ ELSE
.....	
premis m	: IF $x_1$ is $A_1^m$ AND $x_2$ is $A_2^m$ AND ... AND $x_n$ is $A_n^m$ , THEN $y$ is $B^m$ ELSE
Kesimpulan	: $y$ is $B'$

(61)

Dasar dari sistem inferensi *fuzzy* diambil dari masukan *fuzzy* atau crisp, dan menghasilkan *output* berupa himpunan *fuzzy*. Metode sistem inferensi *fuzzy* antara lain, Sugeno, dan Mamdani. Menurut Kusumadewi (2013), Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja *output* sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK. Model *Fuzzy* Sugeno ditunjukkan pada Diagram 13 (Jang, 1997).

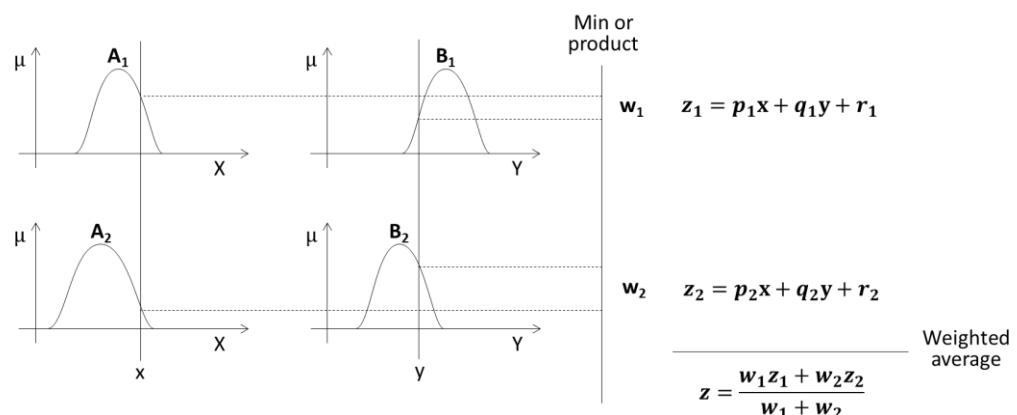


Diagram 13. Model *Fuzzy* Sugeno

Apabila komposisi aturan menggunakan metode Sugeno, maka defuzzifikasi dilakukan dengan cara mencari nilai rata ratanya. Menurut Cox (1994), metode TSK terdiri dari 2 jenis, yaitu:

1) Model *Fuzzy Sugeno* Orde-Nol secara umum dengan  $A_i$  adalah himpunan *fuzzy* ke- $i$  sebagai anteseden, dan  $k$  adalah suatu konstanta (tegas) sebagai konsekuensi ditunjukkan oleh persamaan 62.

$$\text{jika } (x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_n \text{ is } A_n) \text{ maka } z = k \quad (62)$$

2) Model *Fuzzy Sugeno* Orde-Satu secara umum dengan  $A_i$  adalah himpunan *fuzzy* ke- $i$  sebagai anteseden, dan  $p_i$  adalah suatu konstanta (tegas) ke- $i$  dan  $q$  juga merupakan konstanta dalam konsekuensi ditunjukkan oleh persamaan 63.

$$\text{jika } (x_1 \text{ is } A_1) \circ \dots \circ (x_n \text{ is } A_n) \text{ maka } z = p_1 * x_1 + \dots + p_n * x_n + q \quad (63)$$

#### d. Defuzzifikasi

Defuzzifikasi didefinisikan sebagai pemetaan suatu himpunan *fuzzy* yang dihasilkan dari proses *fuzzy inference system* ke nilai *crisp* (Wang, 1997: 108). *Input* dari defuzzifikasi adalah suatu himpunan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut (Kusumadewi, 2013:40). Terdapat 5 metode defuzzifikasi yang umum dikenali, pemilihan metode defuzzifikasi terbaik didasarkan terhadap kepekaan nilai *crisp* yang dihasilkan, kemudahan pengkodean komputer, dan keberlanjutan program (wang 1997:108 & Jang, 1997: 74).

Metode *Center Average Defuzzifier*, nilai *crisp* diperoleh dari rata-rata pembobotan suatu nilai himpunan *fuzzy* (Wang, 1997: 110). Pada metode ini solusi *crisp* diperoleh dengan cara mengambil titik pusat  $y^*$  daerah *fuzzy*, yang dirumuskan sebagai persamaan 64.

$$y^* = \frac{\sum_{l=1}^M \bar{y}^l w_l}{\sum_{l=1}^M w_l} \quad (64)$$

Himpunan *fuzzy* merupakan irisan atau penggabungan dari 2 atau lebih himpunan *fuzzy*, maka pendekatan terbaik adalah dengan menghitung rata rata pembobotan. Ilustrasi gambar tersebut menunjukkan perhitungan rata rata dari nilai *fuzzy* yang diambil dari titik tengah himpunan *fuzzy* yakni  $\bar{y}_1$  dan  $\bar{y}_2$  dengan bobot hitung berdasarkan tinggi dari himpunan *fuzzy* yakni  $w_1$  dan  $w_2$ . Penggambaran pengambilan crisp ditunjukkan pada Diagram 14 (Wang, 1997).

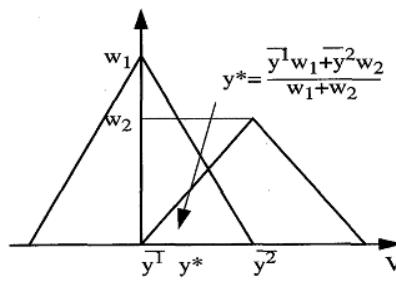


Diagram 14. Nilai crisp pada metode center average defuzzifier

#### e. Peta Konsep Logika *Fuzzy*

Algoritma pemrograman mengacu pada peta konsep logika *fuzzy* yang sudah disediakan dalam *library* program. Peta konsep dari pembangunan *library* logika *fuzzy* ditunjukkan pada Diagram 15.

Fungsi keanggotaan ditentukan oleh interpretasi bahasa verbal atau *linguistic variable* yang digunakan. Setiap bahasa verbal memuat fungsi keanggotaan yang memiliki sebaran data dalam membentuk pola trapesium atau *trapezoidal function*, dan linear atau *piecewise linear function*. Keseluruhan bahasa verbal yang digunakan himpunan *fuzzy* dikenali sebagai *database fuzzy* dan dapat digunakan sebagai rumusan pada *fuzzy inference system* (Kirillov, 2006).

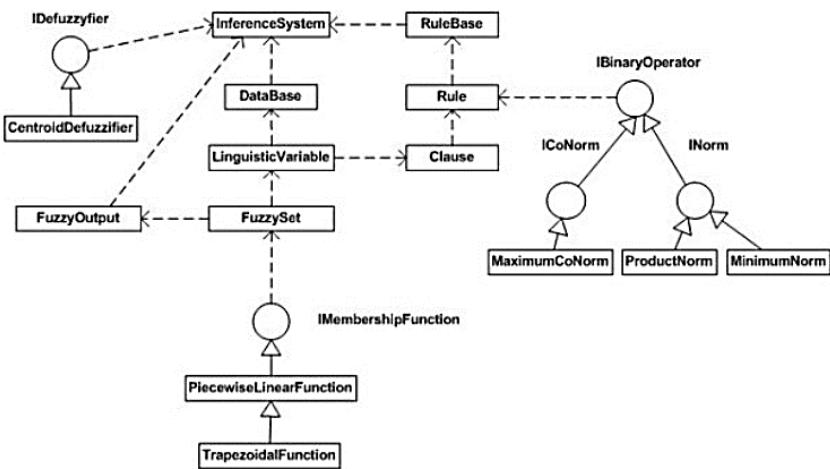


Diagram 15. Peta Konsep *Library Logika Fuzzy*

## E. Model Neurofuzzy

*Neurofuzzy* adalah gabungan dari dua sistem yaitu logika *fuzzy* dan *neural network*. Gabungan kedua sistem tersebut menciptakan bentuk pembelajaran dan adaptasi yang menciptakan asosiasi baru, parameter baru, dan gabungan fungsi baru. Terdapat 3 macam metode penggabungan sistem, seperti *Neural fuzzy systems*, *Fuzzy neural networks*, *Fuzzy-neural hybrid systems* (Lin, 1996:479-480).

Pendekatan *neural fuzzy systems* merupakan metode pembentukan sistem inferensi *fuzzy* melalui beragam macam metode yang dimiliki *neural network* dengan menggunakan fungsi seperti fuzzifikasi, inferensi *fuzzy*, logika *fuzzy* dan defuzzifikasi. Pada sistem ini, *neural network* bertugas sebagai pengolah citra berbasis angka pada himpunan *fuzzy*, seperti pemetaan derajat keanggotaan antara himpunan *fuzzy* dengan teknik pengujian *fuzzy*. Sistem ini banyak digunakan pada pengontrolan alat.

Sistem inferensi *fuzzy* dalam *neurofuzzy* dilakukan berdasarkan pembelajaran yang digunakan pada algoritma *neural network*.

## 1. Arsitektur *Neurofuzzy*

*Fuzzy Inference Networks* pada model bentuk *fuzzy neural networks* lebih dikenal sebagai *adaptive neurofuzzy inference system* (ANFIS) yang menunjukkan pembelajaran berdasarkan jaringan dengan fungsi perhitungan menggunakan inferensi *fuzzy* (Sarikaya, 2008). Metode inferensi *fuzzy* ANFIS dapat menggunakan model Sugeno dan Tsukamoto. Aturan ANFIS IF-THEN adalah sebagai berikut,

Rule 1 : IF  $x$  is  $A_1$  AND  $y$  is  $B_1$ , THEN  $f_1 = p_1x + q_1y + r_1$ ,

Rule 2 : IF  $x$  is  $A_2$  AND  $y$  is  $B_2$ , THEN  $f_2 = p_2x + q_2y + r_2$ ,

Bentuk Arsitektur dari model *neurofuzzy* ditunjukkan pada diagram 16 (Jang, 1997).

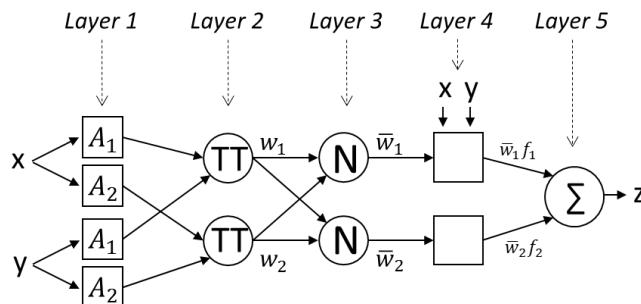


Diagram 16. Arsitektur *Neurofuzzy*

Luaran *neurofuzzy* pada tiap layer  $O_{l,k}$  (Jang, 1997: 337) sebagai berikut,

**Layer 1** : Bentuk *input* parameter yang berupa bahasa verbal diubah ke dalam derajat keanggotaan  $O_{1,k}$  pada himpunan *fuzzy*.

$$O_{1,k} = \mu_{A_k}(x), \quad \text{untuk } k = 1, 2 \text{ atau}$$

$$O_{1,k} = \mu_{A_{k-2}}(y), \quad \text{untuk } k = 3, 4$$

Fungsi keanggotaan *fuzzy* yang didekati berdasarkan parameter *input* berdasarkan fungsi tiap *neuron input* sebagai berikut

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}} \quad (65)$$

**Layer 2 :** *Neuron* pada lapisan ini merupakan hasil perkalian nilai dari setiap *input* dengan bobot tambahan yang diberikan. Luaran dari lapisan ini berupa parameter tetap.

$$O_{2,k} = w_k = \mu_{A_k}(x) \mu_{B_k}(y), \quad \text{dengan } k = 1, 2, \dots, n \quad (66)$$

**Layer 3 :** Data parameter yang dihasilkan dari layer sebelumnya, dilakukan normalisasi data untuk menghasilkan data berdistribusi normal, yaitu rasio keluaran *neuron* ke-k pada lapisan sebelumnya terhadap keseluruhan luarannya.

$$O_{3,k} = \bar{w}_k = \frac{w_k}{w_1 + w_2 + \dots + w_n}, \quad \text{dengan } k = 1, 2, \dots, n \quad (67)$$

**Layer 4 :** Fungsi *neuron* dengan bobot  $\bar{w}_i$  dan  $\{p_i, q_i, r_i\}$  merupakan himpunan parameter dilakukan pembelajaran backpropagation, menghasilkan *neuron* adaptif (parameter berubah).

$$O_{4,k} = \bar{w}_k \cdot z_k = \bar{w}_k (p_k x + q_k y + r_k), \quad k = 1, 2, \dots, n \quad (68)$$

**Layer 5 :** Hanya ada satu simpul tetap yang fungsinya untuk menjumlahkan semua masukan dalam satu sinyal.

$$O_{5,i} = \sum_{i=1}^n \bar{w}_i z_i = \frac{w_1 z_1 + w_2 z_2 + \dots + w_n z_n}{w_1 + w_2 + \dots + w_n} \quad (69)$$

## 2. Penentuan Variabel *Input* dan *Output*

Variabel *input*  $x_j, j = 1, 2, \dots, m$  menggunakan parameter statistik dari hasil ekstraksi citra. Banyaknya variabel *input* menentukan banyaknya *neuron* pada lapisan 1. Pemilihan variabel *input* berhubungan dengan variabel *output*  $y_i, i = 1, 2, \dots, n$ . Variabel *output* dari pengujian tekstur butir diukur berdasarkan karakter permukaan beras tiap butir.

Tekstur butir beras dipengaruhi oleh 2 kriteria, yakni butir kuning dan derajat sosoh. Tekstur butir kuning terdiri dari dua jenis, “butir rusak” dan “butir normal”, sedangkan tekstur derajat sosoh dalam standar SNI : 2008 terdiri dari “sosoh 100%”, “sosoh 95%”, dan “sosoh 85%”. Variabel *output* dari penggabungan kedua jenis penggolongan berupa klasifikasi kelas.

Klasifikasi kelas variabel berupa penggolongan nilai linguistik. Nilai linguistik yang digunakan didasarkan pada nilai derajat keanggotaan tertinggi pada kelas tersebut. Terdapat beberapa kriteria dalam klasifikasi butir beras, yaitu:

- a. Kelas Premium baik jika butir normal berderajat sosoh 100%.
- b. Kelas Premium buruk jika butir rusak berderajat sosoh 100%.
- c. Kelas Medium baik jika butir normal berderajat sosoh 95%.
- d. Kelas Medium buruk jika butir rusak berderajat sosoh 95%.
- e. Kelas Ekonomis jika butir rusak atau butir normal berderajat sosoh 85%.

Pengujian dengan menggunakan metode *neurofuzzy* membagi 2 bagian sampel. Menurut Hota, Shrivastava & Singhai (2013), 60% bagian sampel digunakan sebagai data *training* dan 40% bagian sampel digunakan sebagai data *testing*. Data *training* digunakan untuk pembentukan model terbaik, sedangkan data *testing*

digunakan untuk menguji ketepatan model hasil pengolahan data *training* (Machfudhoh, 2014).

Seleksi variabel *input* dilakukan dengan cara mengeliminasi variabel-variabel yang tidak diperlukan dan mempertahankan variabel-variabel yang memberikan nilai error terkecil terhadap variabel *output*  $y_i$ . Proses ini menggunakan metode eliminasi *backward* dengan fungsi biaya *Sum Square Error* atau SSE (Setiaji, 2014: 74).

### 3. Pemecahan data *Fuzzy* (*Clustering*)

*Fuzzy clustering* merupakan salah satu metode yang digunakan untuk mengelompokkan data *training* menjadi  $r$  kelompok atau *cluster*, setiap data akan dimasukkan kedalam kelas dan setiap kelas mempunyai kemiripan data antara satu dengan data yang lainnya. Penggunaan *clustering* pada *neurofuzzy* bertujuan untuk mengelompokkan data berdasarkan kemiripan data (Chiu, 1994). Salah satu metode yang digunakan untuk pemecahan data adalah *subtractive clustering*. *Subtractive clustering* dikembangkan dari algoritma *fuzzy c-mans clustering* yang sesuai dengan persamaan 70.

$$J = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \|x_k - v_i\|^2 \quad (70)$$

*Fuzzy C-Mean* adalah teknik pengelompokan data di mana setiap data dalam suatu *cluster* ditentukan oleh derajat keanggotaannya. Konsep dasar *Fuzzy C-Mean* yang pertama adalah menentukan pusat cluster yang akan menandai lokasi rata-rata untuk tiap-tiap *cluster* yang diperkenalkan oleh James C. Bezdek pada tahun 1981. *Output* dari *fuzzy c-mean* berupa data dengan *cluster* yang berhubungan

dan derajat keanggotaan tiap data. Informasi ini dapat digunakan untuk membangun suatu sistem inferensi *fuzzy* (Setiaji, 2014: 74).

Pengelompokan data secara *Subtractive clustering* mengasumsikan bahwa setiap titik data adalah pusat *cluster* potensial. Metode *Subtractive clustering* memungkinkan memecah banyak *cluster* sesuai dengan nilai fungsi penggunaannya pada ANFIS (Chiu, 1994). Estimasi *cluster* digunakan untuk menginisialisasi metode pengelompokan berbasis pengoptimalan berbasis iteratif *fuzzy c-mean* dan metode identifikasi model ANFIS.

Sistem inferensi *fuzzy* tipe Sugeno memodelkan data variabel input atau output. Konfigurasi perintah sistem inferensi *fuzzy* untuk menggunakan pengelompokan *subtraktif*. Hasil penerapan *subtractive clustering* pada model *fuzzy c-mean* ditunjukkan pada Diagram 17 (Chiu, 1994).

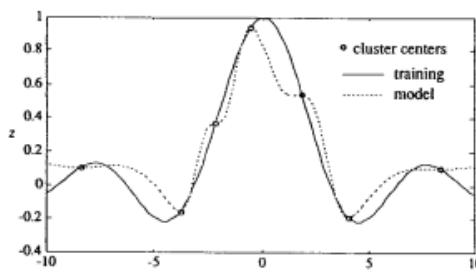
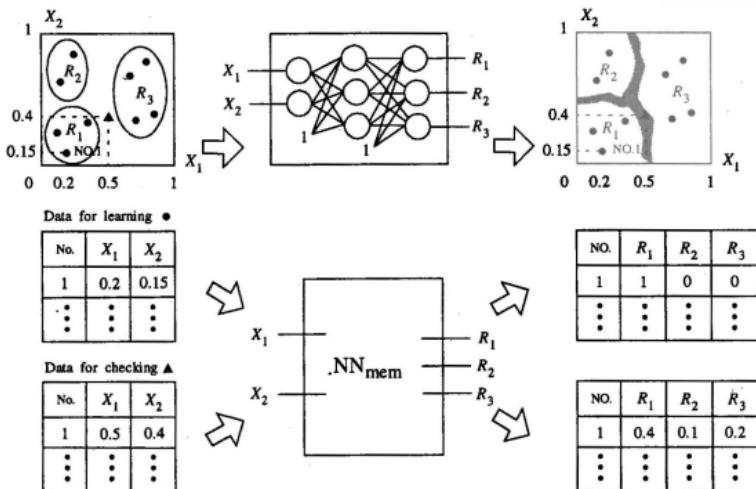


Diagram 17. Hasil Penerapan *Subtractive Cluster* pada model *fuzzy c-mean*

#### 4. Pembelajaran *Neural network* Pada Sistem Inferensi *Fuzzy*

Dasar penerapan *Neural network* pada sistem inferensi *fuzzy* model Takagi Sugeno TSK (Sugeno and Kang, 1988) merupakan pengembangan dari derajat keanggotaan *fuzzy* model TSK yang akan diterapkan dalam sistem inferensi *fuzzy* berasal dari keadaan yang dikondisikan oleh suatu *Neural network*. Pendekatan metode tersebut dapat memberi solusi pada masalah mendasar *fuzzy* seperti

kesalahan dalam menentukan nilai fungsi keanggotaan, dan kesalahan dari fungsi pembelajaran yang mengacu pada aturan yang diterapkan sistem inferensi *fuzzy*. (Lin, 1996 : 507). Penerapan *neural network* dalam sistem inferensi *fuzzy* ditunjukkan pada gambar 20.



Gambar 20. Penerapan *Neural network* dalam sistem inferensi *fuzzy* Model TSK

Pembelajaran *neural network* pada sistem inferensi *fuzzy* memiliki pola keterhubungan dari tiap tahapan. Pola pembelajaran mempengaruhi bagian anteseden dan konsekuensi. Adapun tahapan pembelajaran sesuai dengan (Lin, 1996 : 509-510) adalah sebagai berikut,

- Pembelajaran *Neural network* yang berhubungan dengan bagian IF pada sistem inferensi *fuzzy*. Setiap nilai *input* data *training* didefinisikan  $m_i = (m_i^1, m_i^2, \dots, m_i^r)$  untuk setiap  $i = 1, 2, \dots, n$  dan  $s = 1, 2, \dots, r$ , dengan aturan *fuzzy*  $m_i^s = 1$ , dan  $m_i^k = 0$  untuk  $k \neq s$ , maka derajat keanggotaan pada bagian IF dapat definisikan sebagai nilai inferensi  $\hat{m}_i^s$  sebagai *output* jaringan yang dilatih sesuai dengan persamaan 71.

$$\mu_{AS}(x_i) = \hat{m}_i^s \quad (71)$$

b. Pembelajaran *neural network* yang berhubungan dengan bagian THEN pada sistem inferensi *fuzzy*. Pada tahap ini, hasil pelatihan diujikan pada data pembelajaran dengan nilai *input* data *training*  $x_{i1}^s, x_{i2}^s, \dots, x_{im}^s$  untuk setiap  $i = 1, 2, \dots, n_c$  dan target *output*  $y_i^s$ , untuk setiap  $i = 1, 2, \dots, n_s$ . Pengujian pada data tersebut bertujuan untuk mendapatkan nilai *sum square error* yang dirumuskan sesuai dengan persamaan 72

$$E_m^s = \sum_{i=1}^{n_c} [y_i - f_s(x_i)\mu_{As}(x_i)]^2 \quad (72)$$

Estimasi nilai  $\mu_{As}(x_i)$  dapat ditambahkan sebagai bobot *output* pada pembelajaran *Neural network*, sehingga menghasilkan nilai error dengan pembobotan yang dirumuskan sesuai dengan persamaan 73

$$E_m^s = \sum_{i=1}^{n_c} \mu_{As}(x_i) [y_i - f_s(x_i)\mu_{As}(x_i)]^2 \quad (73)$$

c. Penyederhanaan bagian THEN menggunakan eliminasi dari metode *backward*. Eliminasi nilai *input* pada salah satu parameter yang didefinisikan sebagai  $x_p$  dilakukan untuk memberikan hasil terbaik. Pembelajaran *Neural network* kembali dilakukan pelatihan menggunakan data *training* seperti dalam tahapan sebelumnya, dan didapat nilai *sum square error* dengan eliminasi *input*  $E_{m-1}^{sp}$  yang didefinisikan sesuai dengan persamaan 74.

$$E_{m-1}^{sp} = \sum_{i=1}^{n_c} [y_i - \mu_s(\hat{x}_i)\mu_{As}(\hat{x}_i)]^2 \quad (74)$$

Setelah  $E_m^s > E_{m-1}^{sp}$  terpenuhi akan ditentukan parameter konsekuensi untuk tiap aturan inferensi *fuzzy* ke-s dengan menggunakan metode *Least Square Estimator* (LSE) untuk mengidentifikasi parameter linearnya. LSE merupakan pembelajaran *offline* dengan menggunakan metode *invers* (Setiaji, 2014: 80).

d. Penentuan hasil *output*. Persamaan 75 dapat memberikan hasil akhir terhadap nilai kontrol  $y_i^*$  dengan  $i = 1, 2, \dots, n$ .

$$y_i^* = \frac{\sum_{s=1}^r \mu_{As}(x_i) \mu_s(x_i)}{\sum_{s=1}^r \mu_{As}(x_i)} \quad (75)$$

## F. Penelitian yang Relevan

Penelitian yang relevan dijadikan refrensi dalam penelitian, rumusan masalah, serta materi tambahan dalam pembahasan analisa data. Berikut ini merupakan penelitian yang relevan dengan bahasan pengujian yang dilakukan.

Prihadi Wibowo, S. Dewi Indrasari, dan Jumali (2009) mengangkat identifikasi karakteristik dan hubungannya dengan tingkat standar mutu beras konsumsi di Jawa Barat, khususnya Kabupaten Indramayu, Cianjur, dan Ciamis. Karakteristik beras yang diamati meliputi karakter fisik (ukuran dan bentuk beras, kadar air, derajat sosoh, persentase beras kepala, butir pecah, butir menir, butir kuning atau rusak, dan butir mengapur), dan sifat fisiokimia (kadar amilosa dan sifat konsistensi gel), serta kadar protein. Hasil penelitian menunjukkan, beras yang banyak diperdagangkan di tiga kabupaten memiliki karakteristik dan mutu yang relatif sama, berasnya kering (kadar air <14%), bentuk butiran ramping (rasio p/l:3, 0-3,4), berukuran panjang (6,6-6,8 mm), persentase beras kepala tinggi (>70%), berwarna putih (derajat sosoh >95%), butir patah <26%, menir <2%, butir mengapur <3%, butir kuning-rusak<3%, tingkat kepulenan nasi sedang (kadar amilosa 23-25%) dengan tekstur nasi sedang (konsistensi gel 56~60 mm), dan kadar protein 8-9%.

Adnan, Suhartini, dan Bram Kusbiantoro (Adnan, 2013) menjelaskan tentang identifikasi varietas berdasarkan warna dan tekstur permukaan beras menggunakan pengolahan citra digital dan jaringan syaraf tiruan. Penelitian ini menggunakan *input* warna dan tekstur beras dari sebanyak 50 citra untuk setiap varietas. Data fitur warna dan teksur dianalisis menggunakan ANOVA dengan uji lanjut DMRT untuk mendapatkan nilai tengah, sedangkan sebaran data menggunakan metode *box plot*. Kombinasi warna dan tekstur sebagai parameter masukan dilakukan analisa menggunakan JST. Hasil penelitian menunjukkan JST dapat mengenali tiga parameter *output* berupa varietas beras Basmati, Inpari 1, dan Sintanur. Analisa tekstur dengan lima parameter masukan merupakan faktor terbaik yang digunakan untuk pemodelan JST dengan tingkat keakuratan 100%.

Agus Supriatna Somantri, Emmy Darmawati, dan I wayan Astika membahas pengolahan citra tekstur beras menggunakan pengolahan citra sederhana, untuk mengukur dan menganalisis warna pada tekstur beras. Hasil analisa digunakan untuk mengevaluasi mutu fisik beras. Pengamatan menggunakan parameter dari tingkat lingkaran, keliling dan panjang butir, serta rentang warna *Hue*, *Saturation*, dan *Intensitas* citra. Algoritma jaringan syaraf tiruan menggunakan *backpropagation*, dengan *Learning rate* 0,3 dan *logistic Const* 0,5. Keberhasilan pengujian data *training* melebihi 90%. Pada deteksi 5 jenis beras, dan 10000 butir beras. Tingkat keberhasilan tertinggi mencapai 99,6% pada butir beras Way Apo Buru. Validasi data *testing* sebesar 90,48%. Penelitian pengujian terkendala tidak adanya integrasi dengan perangkat komputasi.