

## BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab 2 berisikan penjelasan dan uraian kajian pustaka, dasar teori, konsep ataupun metode dari literatur penelitian lainnya yang berkaitan dengan penelitian ini. Pada landasan kepustakaan terdapat dasar teori dari berbagai sumber pustaka penelitian lainnya yang menunjang penelitian ini.

### 2.1 Kajian Pustaka

Terdapat beberapa penelitian sebelumnya yang memiliki keterkaitan dengan permasalahan penelitian ini. Penelitian-penelitian tersebut berisi teori dan metode yang digunakan untuk studi kasus dengan objek yang sama. Tabel Kajian Pustaka dapat dilihat pada Tabel 2.1.

**Tabel 0.1 Kajian pustaka**

No.	Judul dan Objek	Metode	Hasil
1.	<p><b>Judul:</b> <i>Pothole Detection in Asphalt Pavement Images</i></p> <p><b>Objek:</b> Input berupa citra perkerasan jalan berlubang. Disebutkan bahwa lubang memiliki bentuk yang cenderung elips sebagai akibat dari distorsi perspektif, secara visual memiliki ciri nampak lebih gelap jika dibandingkan dengan area sekitarnya dan teksturnya cenderung lebih kasar dibandingkan tekstur jalan di sekeliling lubang. (Koch &amp; Brilakis 2011)</p>	<ol style="list-style-type: none"> <li>1. Citra masukan RGB rubah menjadi citra keabuan (<i>grayscale</i>)</li> <li>2. Dari hasil citra keabuan dilakukan penghilangan derau menggunakan <i>Median Filter</i></li> <li>3. Dilakukan segmentasi citra menggunakan <i>Histogram Shape-Based Thresholding Algorithm</i> yang nilai ambangnya ditentukan menggunakan histogram</li> <li>4. Dilakukan ekstraksi bentuk menggunakan Morfologi <i>Thinning</i></li> <li>5. Dilakukan ekstraksi fitur tekstur menggunakan pendekatan statistik berupa standar deviasi dari intensitas keabuan dan kemudian</li> </ol>	<p>Hasil yang ditunjukkan dalam percobaan penentuan <i>threshold</i> untuk ekstraksi bentuk dan perbandingan tekstur yaitu akurasi 86%, presisi 82% dan recall 86%. Hal tersebut menunjukkan bahwa sebagian besar lubang pada gambar perkerasan aspal dapat dideteksi dengan benar.</p>

		dilakukan perbandingan	
2.	<p><b>Judul:</b> <i>GLCM Textural Feature for brain Tumor Classification</i></p> <p><b>Objek:</b> <i>Input berupa citra Magnetic resonance imaging (MRI) atau pencitraan resonansi magnetic penyakit tumor otak. Hasil pengolahan citra diklasifikasikan dalam empat kelas tumor yaitu Astrocytoma meningioma, Metastatic bronchogenic carcinoma atau Sarcoma</i> (Zulpe &amp; Pawar 2012)</p>	<ol style="list-style-type: none"> <li>1. <i>Pre-processing</i> penghilangan derau menggunakan <i>Gaussian Filter</i></li> <li>2. Dilakukan ekstraksi fitur tekstur menggunakan GLCM</li> <li>3. Hasil ekstraksi fitur diklasifikasi menggunakan metode <i>Support Vector Model (SVM)</i></li> </ol>	Hasil menunjukkan bahwa penerapan ekstraksi fitur tekstur GLCM mampu memberikan tingkat pengenalan lebih baik sebesar 97,5%.
3.	<p><b>Judul:</b> <i>Evaluation of Euclidean and Manhattan Metrics In Content Based Image Retrieval System</i></p> <p><b>Objek:</b> <i>Input citra query dipilih secara acak sebanyak 5 dari 100 citra pada tiap kategori. Kategori citra terbagi menjadi 10 yang terdiri dari orang, pantai, monumen, bus, dinosaurus, gajah, bunga, kuda, gunung dan masakan</i> (Khosla et al. 2014)</p>	<ol style="list-style-type: none"> <li>1. Dilakukan ekstraksi fitur warna dengan menggunakan metode <i>color correlogram</i> dan <i>color coherence</i></li> <li>2. Dilakukan ekstraksi fitur tekstur menggunakan <i>Gabor Wavelet</i></li> <li>3. Dilakukan perhitungan jarak terdekat antara citra <i>query</i> dengan korpus citra menggunakan <i>manhattan distance</i> dan <i>euclidean distance</i>.</li> <li>4. Evaluasi sistem menggunakan <i>precision</i></li> </ol>	Hasil evaluasi <i>precision</i> dalam sistem temu kembali citra menunjukkan bahwa metode penghitungan jarak menggunakan <i>manhattan distance</i> menghasilkan nilai berkisar 0,6, sedangkan <i>Euclidean distance</i> berkisar 0,47. Dari evaluasi yang telah dilakukan pada penelitian ini menunjukkan bahwa penggunaan metode <i>manhattan distance</i> lebih baik dibanding <i>euclidean distance</i> .

## 2.2 Dasar Teori

### 2.2.1 Kerusakan Jalan Raya

Kerusakan jalan merupakan suatu kondisi perkerasan jalan yang mengalami perubahan struktur dan bentuk semula menjadi tidak sesuai dengan perkerasan jalan yang sebenarnya sebagai akibat dari suatu kejadian, sehingga menyebabkan jalan tersebut menjadi rusak (Sukirman 1999). Kerusakan dari konstruksi perkerasan jalan kemungkinan diakibatkan oleh:

1. Lalu lintas, berupa peningkatan repetisi dan beban muatan kendaraan di atas permukaan perkerasan.
2. Air, dapat berasal dari air hujan, sistem pembuangan air yang kurang baik sehingga perkerasan jalan menampung air yang berlebih atau kapilaritas yang menyebabkan naiknya air.
3. Material konstruksi perkerasan, diakibatkan oleh pengolahan bahan konstruksi perkerasan yang kurang tepat atau sifat dari material itu sendiri.
4. Iklim, kondisi iklim tropis di Indonesia dengan suhu udara dan curah hujan yang umumnya tinggi merupakan salah satu factor kerusakan jalan.
5. Kondisi tanah yang kurang baik, akibat dari sifat tanah dasar dengan kondisi yang buruk.
6. Proses pemadatan lapisan di atas tanah dasar yang masih buruk.

Kerusakan perkerasan jalan yang terjadi umumnya tidak hanya disebabkan oleh satu faktor saja, namun juga berkemungkinan gabungan dari beberapa faktor yang saling terkait (Sukirman 1999).

Dalam mengevaluasi kondisi perkerasan jalan, maka perlu diketahui:

- Jenis Kerusakan (*distress type*) dan penyebabnya
- Tingkat Kerusakan (*distress severity*)
- Jumlah kerusakan (*distress amount*)

Dengan demikian maka dapat ditentukan jenis penanganannya yang paling sesuai (Sukirman 1999).

### 2.2.2 Jenis Kerusakan Jalan

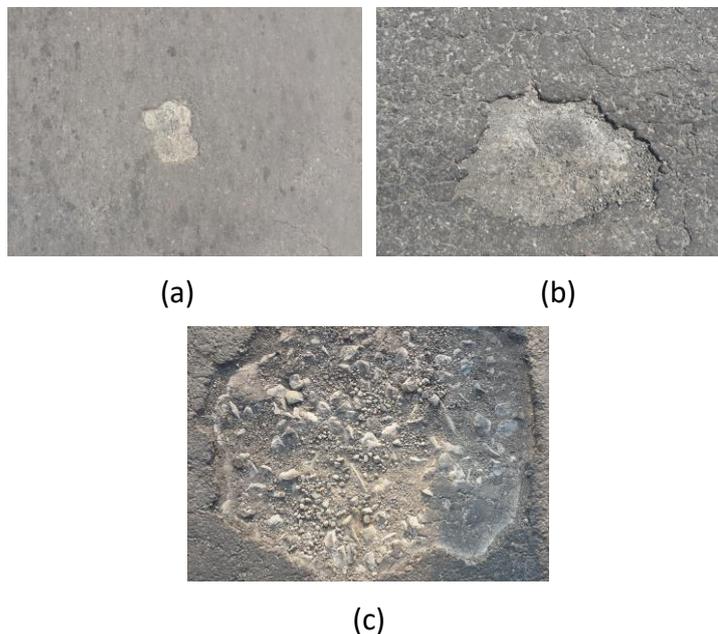
Menurut Manual Pemeliharaan Jalan Nomor : 03/MN/B/1983 yang dikeluarkan oleh Direktorat Jenderal Bina Marga, Kerusakan jalan dikategorikan menjadi:

1. Retak (*cracking*)
2. Distorsi (*distortion*)
3. Cacat permukaan (*disintegration*)
4. Pengausan (*polished aggregate*)

5. Kegemukan (*bleeding or flushing*)
6. Penurunan pada bekas penanaman utilitas

### 2.2.3 Lubang Jalan

Menurut Manual Pemeliharaan Jalan Nomor : 03/MN/B/1983 yang dikeluarkan oleh Direktorat Jenderal Bina Marga, Lubang (*potholes*) masuk kedalam kategori cacat permukaan (*disintegration*), yaitu kerusakan pada lapisan perkerasan secara kimiawi dan mekanis. Lubang pada permukaan perkerasan dapat berupa mangkuk serta memiliki ukuran yang bervariasi dari kecil hingga besar. Lubang-lubang pada permukaan perkerasan menampung air dan meresapkannya ke dalam lapis permukaan, sehingga menyebabkan semakin parahnya kerusakan jalan.



**Gambar 0.1 (a) Lubang jalan aspal rusak ringan, (b) Lubang jalan aspal rusak sedang, (c) Lubang jalan aspal rusak parah**

Ciri visual lubang jalan beraspal menurut Koch & Brilakis (2011) meliputi:

- (1) Secara visual, lubang memiliki ciri yaitu memiliki bagian bayangan yang lebih gelap dibandingkan permukaan sekitarnya.
- (2) Lubang memiliki bentuk yang cenderung atau mendekati elips sebagai akibat dari distorsi perspektif.
- (3) Lubang memiliki tekstur yang lebih kasar dibandingkan tekstur permukaan jalan sekitarnya yang tidak berlubang.

Tabel 0.2 Tingkat kerusakan lubang jalan

Kedalaman maksimum	Diameter rata-rata (mm) (inci)		
	100 – 200 mm (4 – 8 inci)	200 – 450 mm (8 – 18 inci)	450 – 750 mm (18 – 30 inci)
13 mm - ≤25 mm (1/2 – 1 inci)	L	L	M
>25 mm - ≤50 mm (1 – 2 inci)	L	M	H
>50 mm (2 inci)	M	M	H

Sumber: (Shahin, 1994)

Keterangan:

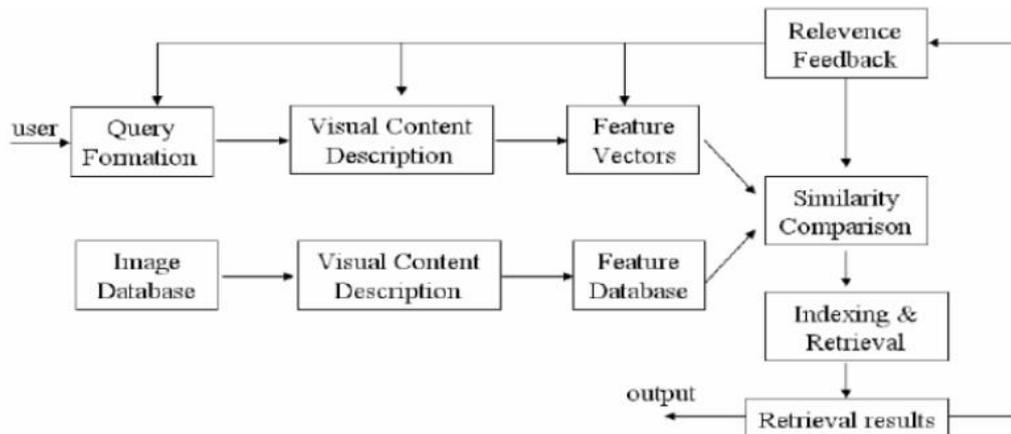
H= Rusak Parah

M= Rusak Sedang

L= Rusak Ringan

### 2.2.4 Content Base Image Retrieval (CBIR)

CBIR merupakan temu kembali citra yang membandingkan antara citra yang ada pada basis data dengan citra *query*. Cara kerja dari CBIR yaitu dilakukan dengan mengukur nilai jarak dari suatu *query* dengan citra yang ada pada basis data atau bisa disebut dengan *image distance measure*. Pengukuran jarak kemiripan dapat dilakukan dengan tiga cara yaitu menghitung kemiripan warna, kemiripan bentuk dan kemiripan tekstur (Rangkuti 2011). Konsep dari CBIR secara umum dapat dilihat pada Gambar 2.2.



Gambar 0.2 Konsep Umum CBIR

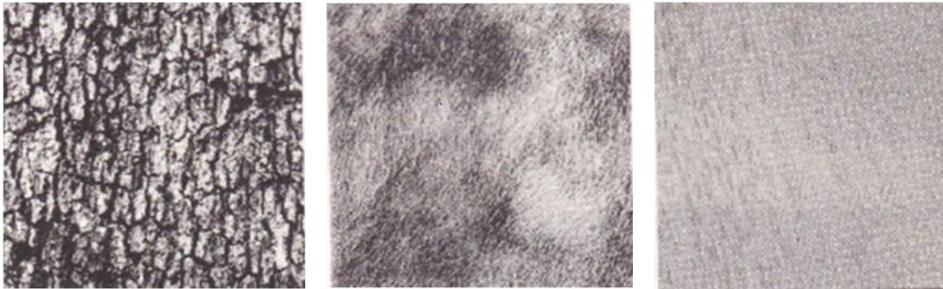
Sumber: (Rangkuti, 2011)

Kemiripan berdasarkan tekstur merupakan karakteristik yang penting untuk menganalisa permukaan dari berbagai macam objek. Istilah tekstur secara umum mengarah pada pengulangan struktur atau elemen dasar yang disebut dengan *texel* yang tersebar secara acak.

### 2.2.5 Citra *Grayscale*

aCitra *grayscale* merupakan representasi citra dengan warna keabu-abuan. Pada citra *grayscale*, intensitas warna dinyatakan seperti komponen RGB yaitu mulai dari rentang nilai 0 hingga 255. 0 menyatakan warna hitam dan 255 menyatakan warna putih. Konversi citra berwarna menjadi citra keabuan sering kali dilakukan dalam pengolahan citra mengingat banyaknya pemrosesan citra yang lebih bekerja pada citra keabuan, misalnya saja digunakan untuk mendapatkan nilai tekstur dari suatu objek (Kadir, 2013).

Berdasarkan penelitian yang dilakukan oleh Powell & Satheeshkumar (2016), dikatakan bahwa sejauh penelitian mengenai citra jalan, informasi yang lebih relevan mengarah pada nilai *pixel* yang lebih rendah. Maka dari itu, informasi berupa warna tidak penting dan tidak diperlukan dalam pendeteksian lubang jalan aspal pada penelitian ini. Di sisi lain, gambar dengan skala abu-abu juga untuk mengurangi kompleksitas pemrosesan.



**Gambar 0.3 Citra *grayscale* dengan beragam tekstur**

Sumber: (Kadir, 2013)

Untuk memperoleh citra *grayscale*, perlu dilakukan perubahan nilai representasi RGB menjadi nilai representasi *grayscale* menggunakan persamaan 2.1.

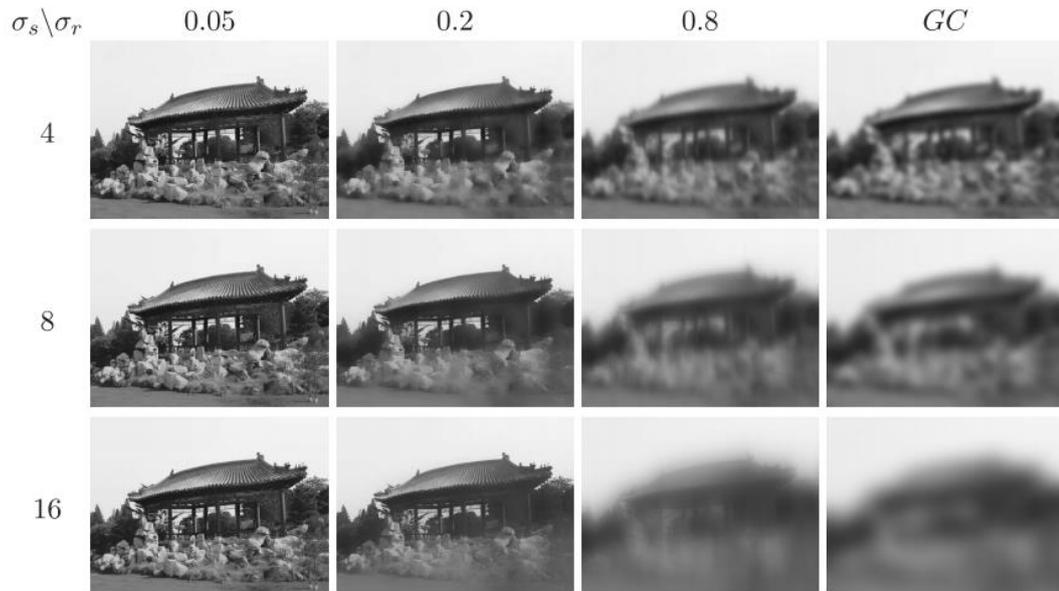
$$grayscale = \frac{R+G+B}{3} \quad (2.1)$$

Keterangan:

R, G, B = nilai representasi dari RGB (*Red, Green, Blue*)

### 2.2.6 *Bilateral Filter*

*Bilateral Filter* merupakan salah satu metode untuk menghilangkan derau atau *noise* pada citra namun tetap mempertahankan struktur (tepi) citra. *Bilateral Filter* sangat mirip dengan *Gaussian Filter*. Bedanya yaitu nilai *bilateral filter* diperoleh berdasarkan rata-rata pembobotan *pixel-pixel* tetangga melalui proses konvolusi. Kunci utama dari *bilateral filter* yaitu satu *pixel* memengaruhi *pixel* lain, tidak hanya *pixel* terdekat namun juga *pixel* yang memiliki nilai sama (S. Paris, P. Kornprobst & Durand, 2009). Penggunaan *bilateral filter* dapat menghilangkan lubang kecil sebagai *noise* dan memperkuat hasil dari *thresholding* (Jo & Ryu, 2015).



**Gambar 0.4** Citra *bilateral filter* dengan nilai  $G_{\sigma_s}$  dan  $G_{\sigma_r}$  yang berbeda-beda

Sumber: (S. Paris, P. Kornprobst & Durand, 2009)

Citra *bilateral filter* dilambangkan dengan BF yang didefinisikan dengan persamaan 2.2.

$$BF_p = \frac{1}{W_p} \sum_{q \in \Omega} G_{\sigma_s}(p - q) G_{\sigma_r}(I_p - I_q) I_q \quad (2.2)$$

$W_p$  merupakan faktor normalisasi yang digunakan untuk memastikan bahwa jumlah nilai bobot dari *pixel* yaitu 1.0.

$$W_p = \sum_{q \in \Omega} G_{\sigma_s}(p - q) G_{\sigma_r}(I_p - I_q) \quad (2.3)$$

Sedangkan  $G_{\sigma}$  merupakan perhitungan menggunakan rumus Gaussian.

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (2.4)$$

Keterangan:

$p$  = koordinat

$q$  = koordinat pusat *pixel* (titik tengah)

$\Omega$  = semua *pixel* yang berada pada lingkungan *kernel*

$I_p, I_q$  = intensitas dari  $p$  dan  $q$

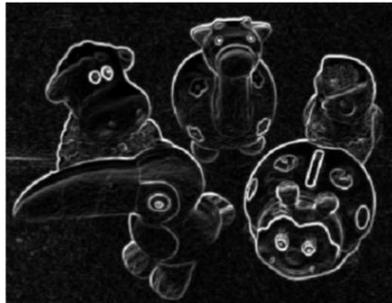
$G_{\sigma_s}(p - q)$  = perhitungan geometris jarak antara  $p$  dan  $q$

$G_{\sigma_r}(I_p - I_q)$  = perhitungan kedekatan fotometrik antara  $I_p$  dan  $I_q$

### 2.2.7 Deteksi Tepi Sobel

Metode Sobel merupakan metode deteksi tepi yang menggunakan operator sobel. Kelebihan dari metode sobel yaitu kemampuan dalam mengurangi *noise* terlebih dahulu sebelum melakukan deteksi tepi. Dalam prosesnya, operator

sobel menggunakan dua *kernel* dengan ukuran 3x3 yang digunakan untuk perhitungan *gradient* terhadap *pixel-pixel* tetangga terdekat dengan titik pusat (Sutoyo et al., 2009). Hasil *thresholding* menggunakan deteksi tepi sobel setelah proses *bilateral filtering* dapat memberikan ekstraksi lubang jalan lebih akurat dibandingkan dengan langsung hanya menggunakan metode *thresholding* (Jo & Ryu, 2015).



**Gambar 0.5 Citra hasil deteksi tepi operator sobel**

Sumber: (Kadir, 2013)

Besaran *gradient* pada titik tengah *kernel* dengan operator sobel dapat dihitung menggunakan persamaan 2.5.

$$G = \sqrt{|G_x|^2 + |G_y|^2} \quad (2.5)$$

$G_x$  dan  $G_y$  merupakan turunan parsial dengan nilai konstanta  $c = 2$  yang dapat dihitung menggunakan persamaan 2.6.

$$\begin{aligned} G_x &= (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \\ G_y &= (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \end{aligned} \quad (2.6)$$

Keterangan:

$G_x$  = *gradient* sobel horizontal

$G_y$  = *gradient* sobel vertikal

$a_0, a_1 \dots a_7$  = nilai *pixel* pada lingkungan *kernel* sobel

Berikut merupakan operator sobel  $G_x$  dan  $G_y$  apabila diimplementasikan ke dalam bentuk *kernel*.

-1	0	+1
-2	0	+2
-1	0	+1

$G_x$

+1	+2	+1
0	0	0
-1	-2	-1

$G_y$

**Gambar 0.6 Kernel operator sobel**

Sumber: (Sutoyo et al., 2009)

### 2.2.8 Otsu Thresholding

Metode Otsu merupakan metode untuk mendapatkan nilai ambang batas dengan cara membagi histogram citra *gray level* menjadi dua daerah. Metode Otsu bekerja dengan cara membedakan antara objek (*foreground*) dan latar belakang (*background*). Nilai ambang yang akan didapatkan yaitu berkisar antara 1 hingga L, nilai L adalah 255 (Mirnasari & Adi 2013). Untuk mendapatkan nilai probabilitas normalisasi setiap *pixel gray level* ke *i* dapat menggunakan persamaan 2.7.

$$P(i) = \frac{n_i}{N}, P(i) \geq 0 \text{ dan } \sum_1^{256} P(i) = 1 \quad (2.7)$$

Keterangan:

$P(i)$  = nilai probabilitas tiap *pixel* ke *i*

$n_i$  = banyaknya nilai *pixel* yang muncul pada *pixel* ke *i*

$N$  = banyaknya *pixel* yang muncul

Untuk menghasilkan satu nilai ambang batas, *pixel* pada suatu citra dibagi menjadi dua kelas yaitu  $C1 = \{0,1,\dots,t\}$  yang biasanya menunjukkan *foreground* dan  $C2 = \{t+1,t+2,\dots,L-1\}$  manunjukkan *background*,  $t$  merupakan nilai ambang batas. Probabilitas 2 kelas kumulatif ditunjukkan dengan persamaan 2.9 dan 2.10.

$$\begin{aligned} \omega_1(t) &= \sum_{i=1}^t P_i \\ \omega_2(t) &= \sum_{i=t+1}^{L-1} P_i \end{aligned} \quad (2.8)$$

Sedangkan nilai *mean* masing-masing kelas tersebut dapat dihitung dengan persamaan 2.11 dan 2.12.

$$\begin{aligned} \mu_1(t) &= \sum_{i=1}^t \frac{iP_i}{\omega_1(t)} \\ \mu_2(t) &= \sum_{i=t+1}^{L-1} \frac{iP_i}{\omega_2(t)} \end{aligned} \quad (2.9)$$

Nilai optimal ambang batas  $t^*$  ditentukan berdasarkan nilai maksimal antara kelas *variance*  $\sigma_B^2$  pada persamaan 2.10.

$$\sigma_B^2 = \sigma_1^2(t) * \omega_1(t) + \sigma_2^2(t) * \omega_2(t) \quad (2.10)$$

$$\begin{aligned} \sigma_1^2(t) &= \sum_{i=0}^t [i - \mu_1(t)]^2 \frac{P_i}{\omega_1(t)} \\ \sigma_2^2(t) &= \sum_{i=0}^{L-1} [i - \mu_2(t)]^2 \frac{P_i}{\omega_2(t)} \end{aligned} \quad (2.11)$$

$$t^* = Arg \underset{0 \leq t < L}{Max} \{ \sigma_B^2(t) \} \quad (2.12)$$

Keterangan:

$\sigma_1^2$  = *variance* kelas 1

$\sigma_2^2$  = *variance* kelas 2

Untuk mengubah nilai asli citra berdasarkan nilai ambang batas  $t$  dapat dilakukan menggunakan persamaan 2.13.

$$G(x) = \begin{cases} 1, & \text{jika } f(x,y) \geq T \\ 0, & \text{jika } f(x,y) < T \end{cases} \quad (2.13)$$

Keterangan:

$G(x)$  = citra hasil *Otsu Thresholding*

$f(x,y)$  = nilai *pixel* pada titik ke  $x, y$

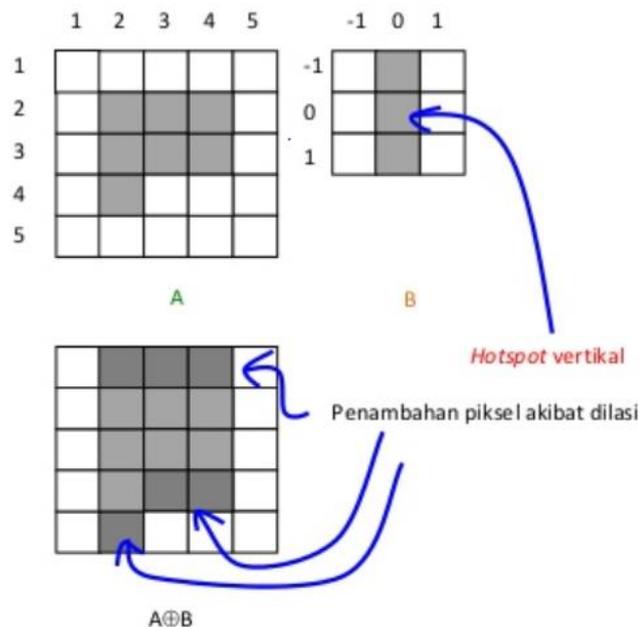
$T$  = nilai ambang batas hasil perhitungan metode otsu

### 2.2.9 Dilation

Metode *dilation* merupakan metode pengolahan citra morfologi untuk memberikan efek pelebaran terhadap *pixel* citra yang bernilai 1 menggunakan operasi dilasi. Operasi *dilation* dapat dihitung menggunakan persamaan 2.15.

$$A \oplus B = \{z | z = a + b, \text{ dengan } a \in A \text{ dan } b \in B\} \quad (2.14)$$

Hasil *dilation* didapatkan dari hasil penjumlahan dari seluruh pasangan A dan B.



Gambar 0.7 Efek dilasi dengan hotspot *vertical*

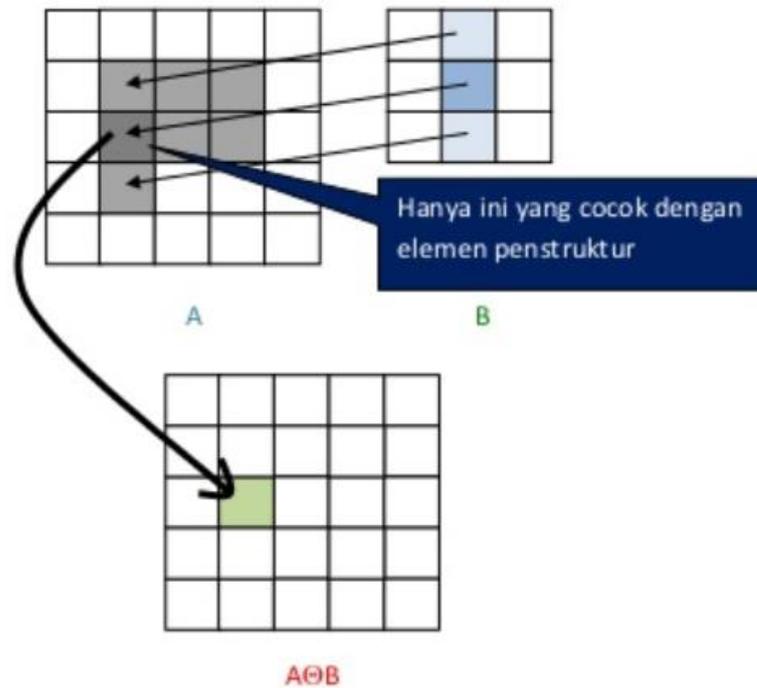
Sumber: (Kadir, 2013)

### 2.2.10 Erosion

Selain morfologi menggunakan operasi dilasi, terdapat pula operasi erosi yang memberikan efek penipisan atau memperkecil struktur citra.

$$A \ominus B = \{p \in Z^2 | (a + b) \in I, \text{ untuk setiap } b \in B\} \quad (2.15)$$

Berikut pada Gambar 2.8 merupakan contoh visualisasi operasi erosi.



**Gambar 0.8** Contoh visualisasi operasi erosi

Sumber: (Kadir, 2013)

### 2.2.11 Closing

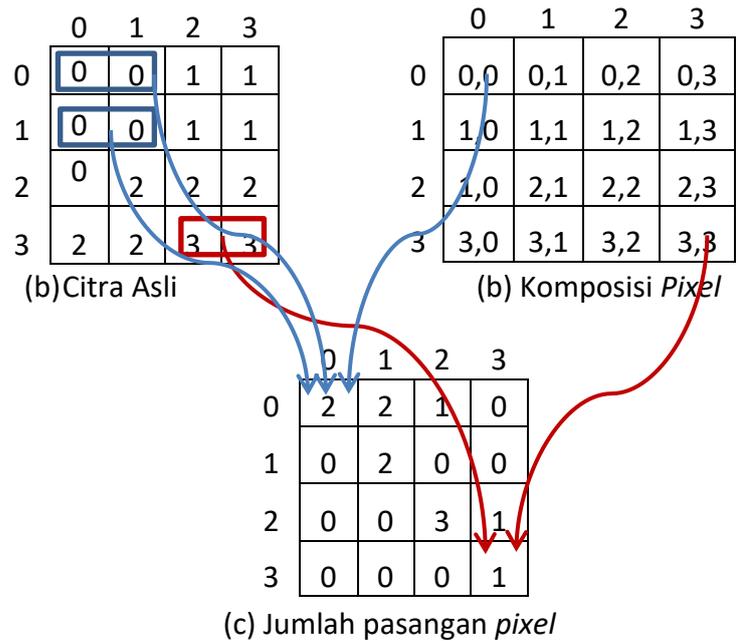
Operasi morfologi selanjutnya yaitu *closing*. Operasi *closing* berguna untuk menghilangkan lubang-lubang kecil sebagai *noise* dan menghaluskan kontur. Operasi *closing* bekerja dengan melakukan operasi dilasi yang selanjutnya dilakukan operasi erosi secara berurutan. Operasi *closing* ditunjukkan menggunakan persamaan 2.16.

$$A \bullet B = (A \oplus B) \ominus B \quad (2.16)$$

### 2.2.12 GLCM

GLCM merupakan salah satu metode yang digunakan untuk ekstraksi fitur pada citra. Untuk memperoleh informasi ekstraksi fitur pada citra, GLCM menggunakan perhitungan statistik, sehingga banyak fitur yang dapat diekstraksi. Metode GLCM dapat pula digunakan untuk antara pengklasifikasian dan membedakan objek yang berbeda (Ault, 2011).

Menurut Xie et al. (2010), Metode GLCM merupakan metode yang melakukan analisis terhadap *pixel* citra untuk mengetahui jejak tingkat keabuan yang sering muncul. Nilai awal GLCM diperoleh dari jumlah kelompok nilai *pixel* yang berkesesuaian pada dua *pixel* yang berjajar. Ilustrasi perhitungan dapat dilihat pada Gambar 2.10.



**Gambar 0.9 Ilustrasi penentuan nilai awal matriks GLCM**

Sumber: (Kadir, 2013)

Matriks pada gambar di atas merupakan matriks *framework* yang perlu diolah hingga menjadi matriks yang simetris. Cara yang dilakukan yaitu menjumlahkan nilai matriks transposenya. Pembentukan matriks GLCM yang simetris dapat dilihat pada Gambar 2.10.

$$\begin{pmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**Gambar 0.10 Hasil pembentukan GLCM simetris**

Sumber : (Kadir, 2013)

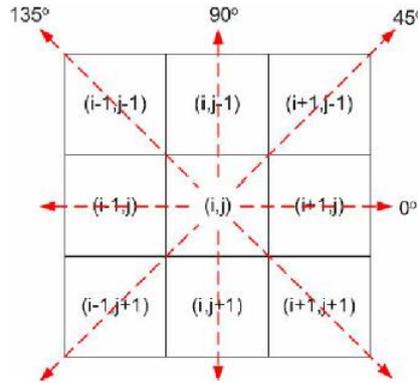
Nilai GLCM dinormalisasikan agar tidak menyebabkan ketergantungan terhadap ukuran citra, sehingga apabila dijumlahkan maka nilainya adalah 1. Matriks normalisasi dapat dilihat pada Gambar 2.11.

$$\begin{pmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{3}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{pmatrix}$$

**Gambar 0.11 Hasil GLCM matriks normalisasi**

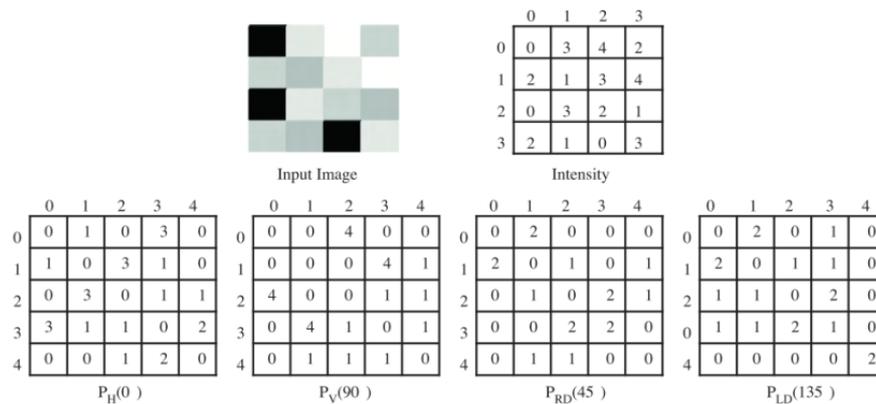
Sumber : (Kadir, 2013)

Metode GLCM melakukan ekstraksi fitur ciri berbasis statistik, fitur ciri diperoleh dari nilai *pixel* matrik yang mempunyai nilai tertentu dan membentuk suatu sudut pola (Kasim & Harjoko 2014), (Xie et al. 2010). Sudut yang dibentuk dari nilai *pixel* citra menggunakan GLCM adalah  $0^{\circ}$ ,  $45^{\circ}$ ,  $90^{\circ}$ ,  $135^{\circ}$  (Eleyan & Demirel 2011), sudut yang terbentuk terlihat pada Gambar 2.12.



**Gambar 0.12 Pixel dengan berbagai sudut**

Sumber : (Eleyan & Demirel, 2011)



**Gambar 0.13 Ilustrasi matriks co-ocurrence**

Sumber : (Eleyan & Demirel 2011)

Terdapat 14 fitur ekstraksi tekstur GLCM yang diusulkan oleh Haralick et al. (1973) yang dapat dihitung dari hasil matriks *co-occurrence*. 14 fitur tersebut yaitu:

1. *Homogeneity, Angular Second Moment (ASM)*

$$ASM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{P(i, j)\}^2 \quad (2.17)$$

ASM atau *energy* digunakan untuk mengukur konsentrasi pasangan intensitas pada matriks *co-occurrence*. Nilai *energy* semakin besar apabila pasangan *pixel* yang memenuhi kondisi matriks *co-occurrence* terkoneksi pada beberapa koordinat dan semakin kecil apabila letak koordinatnya menyebar.

## 2. Contrast

$$\text{Contrast} = \sum_{n=0}^{G-1} n^2 \{ \sum_{i=1}^G \sum_{j=1}^G P(i, j) \}, \quad |i - j| = n \quad (2.18)$$

*Contrast* digunakan untuk mengukur variasi atau perbedaan intensitas tingkat keabuan dalam citra. Nilai kontras semakin besar apabila perbedaan intensitas citra tinggi dan semakin kecil apabila perbedaan intensitas rendah.

## 3. Correlation

$$\text{Correlation} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\{i \times j\} \times P(i, j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y} \quad (2.19)$$

*Correlation* digunakan untuk mengukur korelasi dan ketergantungan antara *pixel* dengan tingkat keabuan *i* dan *pixel* dengan tingkat keabuan *j* pada citra.

## 4. Sum of Square, Variance

*Variance* digunakan untuk mengukur sebaran atau variasi nilai keabuan pada matiks kookurensi awal. Citra dengan sebaran derajat keabuan yang kecil akan menghasilkan *variance* yang kecil pula.

$$\text{Variance} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu)^2 P(i, j) \quad (2.20)$$

## 5. Local Homogeneity, Inverse Difference Moment (IDM)

$$\text{IDM} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i, j) \quad (2.21)$$

IDM atau homogenitas digunakan untuk mengukur homogenitas variasi intensitas citra dengan tingkat keabuan sejenis. Nilai homogenitas semakin besar apabila variasi intensitas citra kecil (citra yang homogen) dan semakin kecil apabila variasi intensitas citra besar (citra yang tidak homogeny).

## 6. Sum Average

*Sum Average* digunakan untuk mengukur banyaknya nilai rata-rata *pixel* pada distribusi tingkat keabuan.

$$\text{AVER} = \sum_{i=2}^{2G} iP_{x+y}(i) \quad (2.22)$$

## 7. Sum Entropy

*Sum Entropy* digunakan untuk mengukur banyaknya tingkat keabu-abuan yang acak.

$$\text{SENT} = - \sum_{i=2}^{2G} iP_{x+y}(i) \log(P_{x+y}(i)) \quad (2.23)$$

## 8. Sum Variance

*Sum Variance* digunakan untuk mengukur seberapa banyak variasi tingkat keabuan dari nilai rata-rata.

$$\text{SVAR} = \sum_{i=2}^{2G} (i - \text{SENT})^2 P_{x+y}(i) \quad (2.24)$$

### 9. Entropy

*Entropy* digunakan untuk mengukur tingkat ketidakaturan bentuk atau distribusi intensitas citra matriks *co-occurrence*. Nilai *entropy* semakin besar apabila citra tidak homogen dan semakin kecil apabila citra homogen.

$$Entropy = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \times \log(P(i, j)) \quad (2.25)$$

### 10. Difference Entropy

*Difference Entropy* digunakan untuk mengukur variasi perbedaan mikro (lokal).

$$DENT = - \sum_{i=0}^{G-1} i P_{x-y}(i) \log(P_{x-y}(i)) \quad (2.26)$$

### 11. Difference Variance

*Difference Variance* digunakan untuk me variasi *pixel* lokal.

$$DVAR = - \sum_{i=0}^{G-1} i^2 P_{x-y}(i) \quad (2.27)$$

### 12. Information Measure of Correlation 1

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad (2.28)$$

### 13. Information Measure of Correlation 2

$$f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}$$

$$HXY = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \log(P(i, j))$$

*HX* dan *HY* merupakan entropi dari  $P_x, P_y$

$$HXY1 = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \log\{P_x(i)P_y(j)\}$$

$$HXY2 = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_x(i)P_y(j) \log\{P_x(i)P_y(j)\} \quad (2.29)$$

### 14. Maxima Correlation Coefficient

$$f_{14} = (\text{Nilai eigen terbesar kedua dari } Q)^{1/2}$$

dimana

$$Q(i, j) = \sum_k \frac{P(i, k)P(j, k)}{P_x(i)P_y(k)} \quad (2.30)$$

Keterangan:

G : jumlah tingkat keabuan

$\mu$  : Nilai rata-rata dari P

$\mu_x, \mu_y$  : Rata-rata dari  $P_x, P_y$

$\sigma_x, \sigma_y$  : Standar deviasi dari  $P_x, P_y$

$P_x(i)$  : Masukan ke i dalam matriks probabilitas marjinal yang diperoleh dari penjumlahan baris  $P(i, j)$

Dari 14 fitur yang diusulkan oleh Haralick tersebut, hanya 13 fitur yang digunakan. Fitur *Maxima Correlation Coefficient* tidak digunakan karena ketidakstabilan dalam komputasi (Haralick et al., 1973).

### 2.2.13 Manhattan Distance

*Manhattan distance* atau biasa disebut  $L_1$  *distance* merupakan pengembangan dari metode  $L_p$  norm. Metode ini bekerja dengan menghitung jarak yang paling pendek antara dua poin (Sharma & Batra, 2014). Perhitungan jarak antara  $u = (x_1, y_1)$  dengan  $v = (x_2, y_2)$  menggunakan *Manhattan Distance* ditunjukkan pada persamaan 2.31.

$$MH(u, v) = |x_1 - x_2| + |y_1 - y_2| \quad (2.31)$$

Apabila memiliki variabel sepanjang  $n$ , misalnya  $a = (x_1, x_2, \dots, x_n)$  dan  $b = (y_1, y_2, \dots, y_n)$ . Maka *Manhattan distance* antara *query* ( $a$ ) dengan basis data ( $b$ ) pada persamaan 2.32.

$$\begin{aligned} MH(a, b) &= |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \\ &= \sum_{k=1}^n |x_k - y_k| \end{aligned} \quad (2.32)$$

### 2.2.14 Metode Seleksi Fitur

#### a.) *Correlation-based Feature Selection* (CFS)

Metode seleksi fitur CFS merupakan metode seleksi fitur yang memanfaatkan teknik heuristik untuk mengevaluasi nilai dari himpunan fitur. Metode CFS memilih fitur terbaik dari fitur yang ada dengan mempertimbangkan kegunaan fitur dalam melakukan prediksi terhadap label kelas atau kategori. Artinya, fitur yang terpilih merupakan fitur-fitur yang memiliki korelasi yang kuat dengan kelas. Sedangkan tiap-tiap fitur yang terbentuk tersebut tidak memiliki korelasi atau berkorelasi rendah dengan fitur yang lainnya (Hall, 1999).

#### b.) *Wrapper Subset Evaluation*

Metode seleksi fitur *wrapper subset evaluation* atau *wrapper* merupakan metode seleksi fitur yang memanfaatkan algoritme induksi atau pembelajaran untuk memperkirakan kegunaan dari fitur-fitur yang terpilih. Fitur yang terbentuk merupakan kandidat fitur yang akan memberikan perkiraan kualitas dari objek yang dibentuk dari algoritme pembelajaran. Meskipun metode seleksi fitur memberikan hasil yang *relative* lebih baik, namun waktu komputasi yang dibutuhkan jauh lebih lama karena pemanfaatan algoritme pembelajaran yang dilakukan berkali-kali (Hall, 1999).

### 2.2.15 Akurasi

Akurasi merupakan metode untuk mengetahui berapa banyak hasil citra temu kembali yang memiliki kesamaan dengan citra *query* (*relevant*) dibagi dengan jumlah temu kembali yang dimunculkan. Hasil perhitungan akurasi

biasanya ditunjukkan dalam bentuk persen. Perhitungan akurasi saja dalam sistem temu kembali masih dirasa kurang karena akurasi hanya melakukan perhitungan berapa hasil temu kembali yang mirip dengan *query* tanpa memperhatikan urutan (*Unrank Retrieval*). Sedangkan, dalam beberapa kasus hasil temu kembali yang sama dengan *query* tidak selalu berapa pada urutan atas (Manning, C., Raghavan, P., & Schütze, H., 2008).

$$Akurasi (A) = \frac{Jumlah\ dokumen\ relevant\ yang\ diterima}{Jumlah\ seluruh\ dokumen} \quad (2.33)$$

### 2.2.16 Mean Average Precision (MAP)

Metode evaluasi MAP melakukan perhitungan pada temu kembali dengan memperhatikan urutan (*ranked retrieval*). Di antara metode perhitungan evaluasi, MAP terbukti memiliki diskriminasi dan stabilitas yang sangat baik (Manning, C., Raghavan, P., & Schütze, H., 2008). MAP merupakan nilai rata-rata dari perhitungan *average precision*. Sedangkan *average precision* merupakan tiap-tiap dokumen *relevant* yang diterima berdasarkan perhitungan *precision*. Untuk dokumen *relevant* yang tidak dihasilkan oleh sistem, maka akan diberi nilai 0. Nilai *average precision* dihitung berdasarkan urutan dokumen yang diberikan oleh sistem, sehingga nilai *precision* diberikan pada tiap dokumen yang dihasilkan oleh sistem (Parwita & Winarko 2015). Perhitungan MAP dapat dilakukan berdasarkan persamaan 2.33.

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \quad (2.34)$$

Keterangan:

$Q$  = jumlah informasi yang diminta oleh pengguna (*query*)

$R$  = dokumen *relevant* yang dihasilkan sistem

$m$  = jumlah dokumen *relevant* yang dihasilkan dari *query*