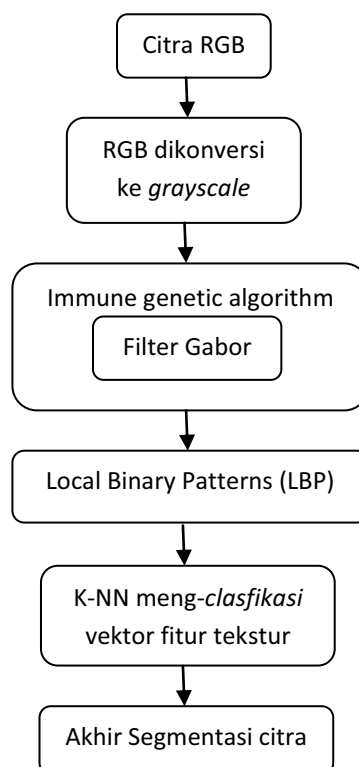


## BAB II

### TINJAUAN PUSTAKA

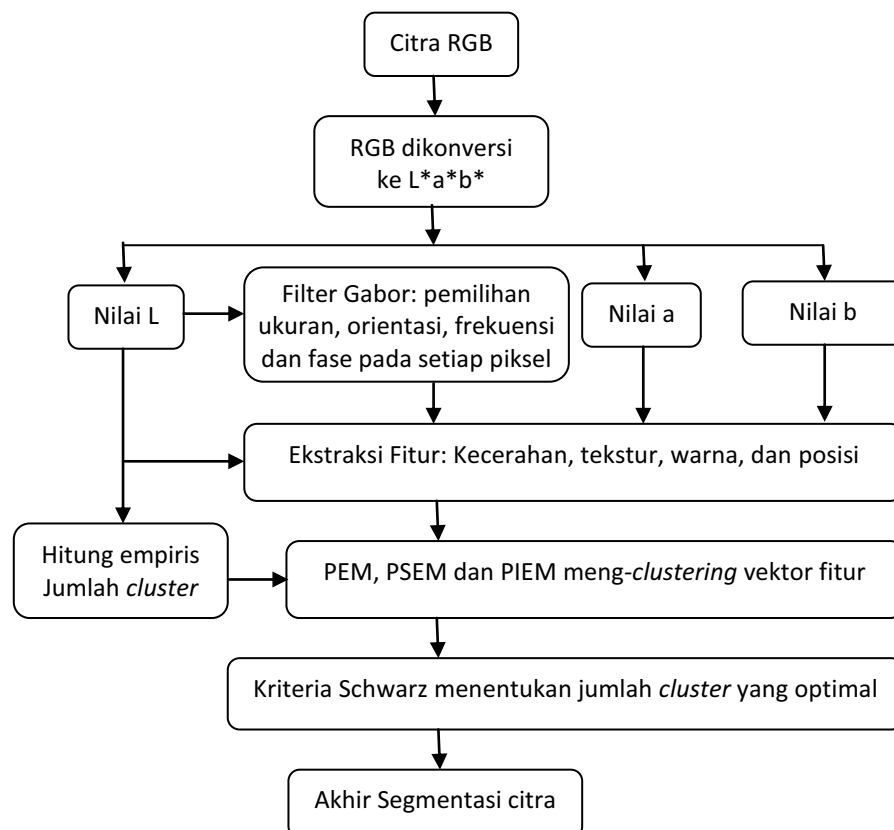
#### 2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Ma Li dan R.C. Staunton [19] sebuah pendekatan baru untuk multi-tekstur segmentasi citra berdasarkan pembentukan suatu vektor fitur tekstur yang efektif. Vektor fitur tekstur diperoleh dari integrasi antara filter gabor tunggal dengan local binary pattern (LBP). Metode yang mereka usulkan memperoleh filter gabor tunggal menjadi efisien untuk jumlah kelas bertekstur yang kecil, tetapi untuk tiga atau lebih, tidak bisa membedakan perbedaan tekstur. Persentase akurasi filter gabor tunggal secara berurutan jumlah segmentasi 2, 3, 4, 5 yaitu 96.23%, 82.45%, 75.56%, 70.34%. Ketika diintegrasikan dengan LBP persentase akurasi secara berurutan jumlah segmentasi 2, 3, 4, 5 yaitu 97.42%, 93.80%, 91.62%, 89.63%.



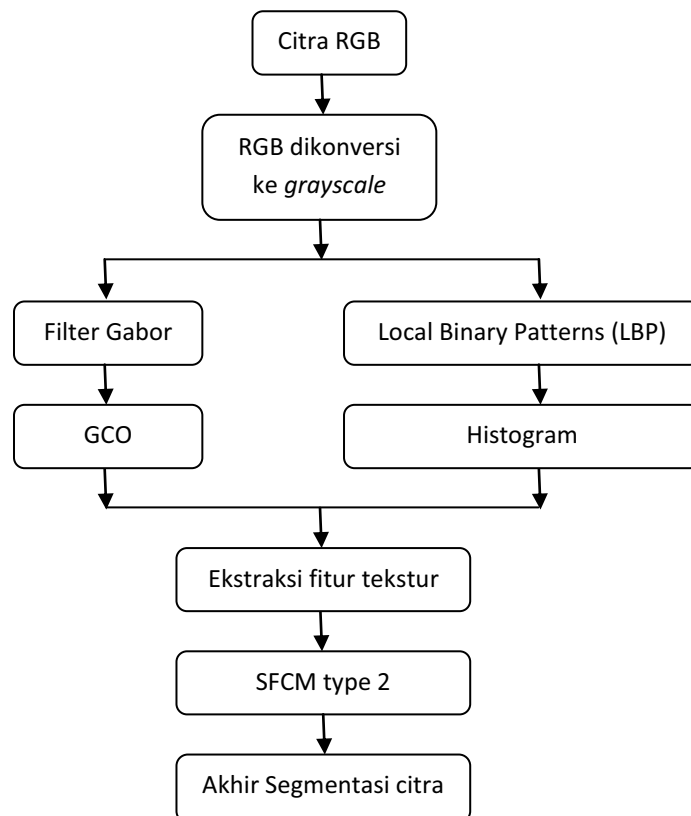
Gambar 2.1 *Flowchart* Penelitian Ma Li dan R.C. Staunton

Penelitian yang dilakukan oleh Jesmin F. Khan et al [6] sebuah pendekatan untuk metode baru seleksi skala yang didasarkan pada properti lokal citra yang terkait dengan perubahan kecerahan, warna, tekstur dan posisi yang diambil untuk setiap pixel pada ukuran yang dipilih dari filter. Fitur citra diukur dengan menggunakan sebuah filter Gabor sesuai dengan adaptif yang dipilih, orientasi frekuensi ukuran, dan fase untuk setiap piksel. Untuk cluster piksel ke daerah yang berbeda, distribusi gabungan dari fitur piksel dimodelkan dengan campuran Gaussians memanfaatkan tiga varian algoritma expectation maximization (EM). Tiga versi yang berbeda dari EM digunakan dalam penelitian dengan *clustering* adalah: (1) penalized EM, (2) penalized stochastic EM, dan (3) penalized inverse EM. Peneliti menentukan nilai dari jumlah model yang paling sesuai jumlah alami cluster dalam citra berdasarkan pada kriteria Schwarz, yang memaksimalkan probabilitas posterior dari jumlah kelompok diberi sampel pengamatan. Ruang warna yang digunakan  $L^*a^*b^*$ . Metode yang mereka usulkan memperoleh persentasi akurasi hasil segmentasi 97%.



Gambar 2.2 *Flowchart* Penelitian Jesmin F. Khan et al

Penelitian yang dilakukan oleh Lotfi Tlig et al [22] untuk analisis tekstur menggunakan filter Gabor yang secara luas telah diterapkan, tetapi filter Gabor memiliki ketergantungan yang kuat pada sejumlah parameter sehingga mempengaruhi kinerja karakterisasi tekstur. Terlebih, filter Gabor tidak dapat mengekstrak fitur tekstur mikro yang juga memiliki efek negatif pada proses *clustering*. Pendekatan yang diambil pada penelitian ini menggabungkan output grating cell operator (GCO) berasal dari filter Gabor dengan local binary pattern (LBP). Efektivitas pendekatan segmentasi yang diusulkan pada berbagai gambar sintesis dan bertekstur. Beberapa hasil eksperimen pada gambar bertekstur menunjukkan keunggulan pendekatan yang diusulkan dalam hal akurasi segmentasi sehubungan dengan perbandingan kuantitatif dan kualitatif.



Gambar 2.3 *Flowchart* Penelitian Lotfi Tlig et al

Tabel 2.1 Rangkamuan penelitian terkait segmentasi citra

Nama Peneliti, Tahun dan Judul	Masalah	Metode	Hasil
Li & Staunton, 2008, Optimum Gabor filter design and local binary patterns for texture segmentation	Tekstur analisis dengan menggunakan filter gabor hanya mampu mendeteksi tekstur makro	Warna: keabu-abuan, tekstur: filter garbor dan LBP, metode: berbasis batas	Adanya peningkatan akurasi secara berurutan untuk jumlah segmentasi 2, 3, 4, 5 dari 96.23% menjadi 97.42%, dari 82.45% menjadi 93.80%, dari 75.56% menjadi 91.62%, dari 70.34% menjadi 89.63%.
Khan et al., 2009 , A customized Gabor filter for unsupervised color image segmentation	Banyaknya fitur karakter sebagai properti lokal menjadi persyaratan pemilihan skala segementasi	Warna: L*a*b*, tekstur: filter gabor, metode: <i>clustering</i>	Hasil segmentasi memperoleh persentasi akurasi 97%
L. Tlig, M. Sayadi & F. Fnaiech, 2012, A new fuzzy segmentation approach based on S-FCM type 2 using LBP-GCO features	Filter Gabor memiliki ketergantungan parameter sehingga mempengaruhi kinerja karakterisasi tekstur, dan tidak dapat mengekstrak fitur tekstur mikro yang juga memiliki efek negatif pada proses <i>clustering</i> .	Warna: keabu-abuan, tekstur: filter garbor dan LBP, metode: <i>clustering</i>	Hasil eksperimen pada gambar bertekstur menunjukkan keunggulan pendekatan yang diusulkan dalam hal akurasi segmentasi sehubungan dengan perbandingan kuantitatif dan kualitatif.

Pada penelitian ini menerapkan ekstraksi fitur menggunakan filter gabor-LBP seperti penelitian dilakukan oleh Ma Li dan R.C. Staunton [19] dan dan Lotfi Tlig et al [22] untuk segmentasi citra berdasarkan fitur warna dan tekstur yang telah diusulkan oleh Jesmin F. Khan [6].

## 2.2 Landasan Teori

### 2.2.1 Segmentasi Citra

Citra merupakan sebuah gambar yang telah diubah dari sinyal kontinu ke format digital sehingga mudah dibaca oleh komputer. Citra merupakan fungsi intensitas cahaya  $f(x,y)$ , dimana nilai  $x$  dan  $y$  merupakan koordinat spasial dan nilai fungsi merupakan tingkat kecemerlangan citra pada setiap titik  $(x,y)$  [2]. Setiap titik sebagai elemen citra yang disebut piksel (*picture element*). Piksel  $(x,y)$  mengacu koordinat horisontal dan vertikal citra dengan ukuran  $M \times N$ , dimana  $M$  lebar citra dan  $N$  tinggi citra.



Gambar 2.4 Piksel (243,107) pada citra peppers dengan ukuran 384 x 512

Segmentasi membagi citra ke beberapa wilayah yang unik berdasarkan piksel citra yang homogen [2] [3]. Jika  $X$  adalah nilai piksel citra yang disegmentasi ke wilayah yang unik seperti  $\{C_1, C_2, C_3, \dots, C_n\}$  maka secara matematis segmentasi dapat didefinisikan sebagai berikut [3]:

$$\bigcup_{i=1}^n C_i = X, C_i \cap C_j = \emptyset, i \neq j \quad (1)$$

Setiap piksel pada citra secara keseluruhan tersegmentasi di daerah yang unik dan tidak ada irisan antara daerah tersebut. Tujuan dari segmentasi yaitu memisahkan citra ke beberapa wilayah sehingga direpresentasi secara sederhana menjadi sesuatu yang bermakna dan mudah untuk dianalisis.

### 2.2.2 Ekstraksi Fitur

Fitur citra merupakan atribut dari suatu citra yang ditentukan oleh tampilan visual atau hasil manipulasi tertentu dari piksel citra. Ekstraksi fitur adalah prosedur menghasilkan deskripsi dari suatu obyek dalam hal parameter terukur yang mewakili sifat yang relevan dari obyek, dan dapat digunakan untuk pengelompokan dengan menetapkan objek ke kelas [12]. Fitur citra yang digunakan untuk segmentasi adalah fitur warna dan tekstur, meskipun sifat dari fitur terpisah dimana tekstur menggunakan tingkat keabu-abuan citra sedangkan warna mengekstraksi semua informasi pada ruang warna.

#### 2.2.2.1 Fitur Warna $L^*a^*b^*$

Warna dilihat oleh manusia sebagai kombinasi dari R (merah), G (hijau), dan B (biru). Model warna RGB yang tersusun dari tiga kromatisitas warna merah, hijau, dan biru, dan dapat menghasilkan kromatisitas yang merupakan segitiga warna-warna primer. RGB memiliki tiga matriks dimensi berukuran  $m \times n \times 3$  yang merepresentasikan warna merah, hijau, dan biru untuk setiap pikselnya. Tiap nilai yang mewakili masing-masing intensitas kecerahan warna. Apabila ketiga digabungkan akan membentuk suatu kombinasi warna baru tergantung besarnya tingkat kecerahan warna dengan berukuran 8 bit, berarti memiliki tingkat kecerahan warna sampai 256. Artinya tiap warna dapat menyumbang tingkat kecerahan warnanya dari rentang level 0 sampai level 255. Dimana 0 merepresentasikan warna hitam dan 255 merepresentasikan warna putih. Dari representasi R, G, B dapat diperoleh jenis lain dari representasi (ruang) warna dengan menggunakan transformasi linear ataupun nonlinear.

Sistem warna CIE (Commission International de l'Eclairage) dikembangkan untuk mewakili keseragaman persepsi, dan dengan demikian memenuhi kebutuhan psikofisik untuk pengamat manusia [5]. CIE menjadi sistem warna standar sehingga ketika warna ditentukan oleh sistem ini berarti dapat digunakan dimana saja terhadap perangkat printer, scanner atau monitor dan salah satu sistem CIE paling berguna adalah CIE 1976  $L^*$ ,  $a^*$ ,  $b^*$ , dengan singkatan resmi CIELAB [23] karena memiliki efektifitas transformasi warna [2]. CIELAB memiliki tiga komponen  $L^*$  berkorelasi

dengan kecerahan,  $a^*$  dengan kemerahan-kehijauan dan  $b^*$  dengan kekuningan-kebiruan. Dengan rentang nilai  $L^*$  antara 0 sampai 100, nilai  $a^*$  dan  $b^*$  antara -128 sampai 127.

Ruang warna baru standar berdasarkan ruang RGB kolorimetri (sRGB) baru-baru ini telah diusulkan untuk menggabungkan perbedaan antara ruang RGB monitor dan ruang perangkat independen (seperti ruang HVS dan CIE LAB) yang menjadi kendala karena kompleksitas dari operasi yang diperlukan untuk mendukung transformasi dari atau ke ruang warna sesuai dengan perangkat [24]. Dalam melakukan transformasi model warna RGB ke model warna Lab terlebih dahulu dilakukan proses transformasi model warna RGB ke sRGB, sRGB ke CIE XYZ, kemudian CIE XYZ ke CIE Lab. Di bawah ini adalah rumus standar untuk konversi linier RGB ke CIE LAB [24]:

1. Nilai  $x$  adalah nilai R,G atau B yang dinormalisasi dengan nilai antara 0 – 1, sehingga nilai R,G atau B dibagi dengan nilai maksimal yaitu 255. Nilai  $f(x)$  menunjukkan nilai konversi sR, sG dan sB.

$$f(x) = \frac{x}{12.92} ; x \leq 0.04045 \quad (2)$$

$$f(x) = \frac{x+0.055}{1.055}^{2.44} ; x > 0.04045$$

$$sRGB = 100 * f(x) \quad (3)$$

2. Selanjutnya nilai sRGB ditransformasi ke model warna CIE XYZ menggunakan persamaan

$$\begin{matrix} X & 0.4125 & 0.3576 & 0.1804 & R \\ Y & 0.2127 & 0.7152 & 0.0722 & * G \\ Z & 0.0193 & 0.1192 & 0.9502 & B \end{matrix} \quad (4)$$

3. Perhitungan transformasi ruang warna dari CIE XYZ ke CIE LAB

$$L^* = 116 f \frac{Y}{Y_n} - 16 \quad (5)$$

$$a^* = 500 f \frac{X}{X_n} - f \frac{Y}{Y_n}$$

$$b^* = 200 f \frac{Y}{Y_n} - f \frac{Z}{Z_n}$$

dimana

$$f s = \sqrt[3]{s}; \quad s > 0.008856$$

$$f s = 7.787 s + 16 \quad ; \quad s \leq 0.008856$$

dan  $X_n$ ,  $Y_n$ ,  $Z_n$  adalah nilai referensi warna putih, umumnya menggunakan standar CIE penyahayaan D65  $X_n = 95.0456$ ,  $Y_n = 100.0000$ ,  $Z_n = 108.9058$ .

Matlab mengikuti ketentuan spesifikasi profil ICC (International Color Consortium) (ICC 1:2001-4) [25] untuk mewakili nilai-nilai bilangan bulat 8-bit atau 16-bit. Format profil warna  $L^*a^*b^*$  menurut ICC untuk rentang nilai  $L^*a^*b^*$ :

Tabel 2.2 Rentang nilai  $L^*$  sesuai dengan ICC

Nilai ( $L^*$ )	format uint8	format uint16
0.0	0	0
100.0	255	65280
$100.0 + (25500/65280)$	None	65535

Tabel 2.3 Rentang nilai  $a^*$  dan  $b^*$  sesuai ICC

Nilai ( $a^*$ or $b^*$ )	format uint8	format uint16
-128.0	0	0
0.0	128	32768
127.0	255	65280

$$L^* = \frac{L^* \cdot 255}{100} \quad ; \text{uint8} \quad (6)$$

$$\frac{L^* \cdot 255}{100} * 256 ; \text{uint 16}$$

$$a^* = a^* + 128 \quad ; \text{uint8}$$

$$a^* + 128 * 256 ; \text{uint16}$$

$$b^* = b^* + 128 \quad ; \text{uint8}$$

$$b^* + 128 * 256 ; \text{uint16}$$





Gambar 2.5 Citra RGB (a) dan citra CIE LAB (b)

### 2.2.2.2 Fitur Tekstur Filter Garbor

Sebuah filter Gabor adalah filter linier yang impulse response didefinisikan oleh fungsi harmonik dikalikan dengan fungsi Gaussian. Daugman [26] menemukan bahwa sel-sel yang sederhana dalam visual korteks otak mamalia dapat dimodelkan oleh fungsi Gabor, sehingga analisis citra oleh filter Gabor mirip dengan persepsi dalam sistem visual manusia. Tekstur citra dapat dilihat sebagai sinyal periodik, dan relatif energi terkonsentrasi dalam rentang frekuensi tertentu. Fungsi gabor dua dimensi (2D) memiliki bentuk umum berikut seperti yang diusulkan oleh Daugman [26]:

$$G_{\lambda\theta\phi\sigma\gamma} x, y = e^{-\frac{x^2 + \gamma^2 y^2}{2\sigma^2}} \cos 2\pi \frac{x}{\lambda} + \phi \quad (7)$$

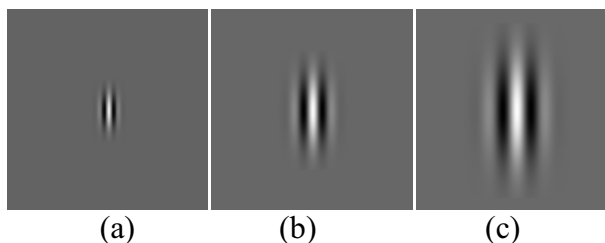
$$x = x \cos \theta + y \sin \theta$$

$$y = -x \sin \theta + y \cos \theta$$

Dari persamaan (7) memiliki parameter untuk menghasilkan filter dengan fungsi gabor:

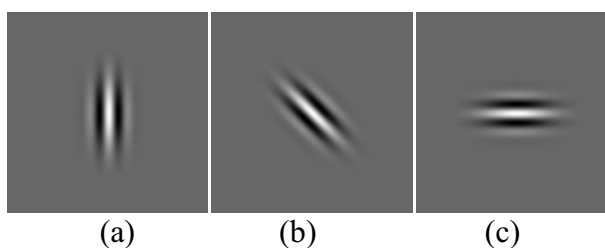
1. Lambda ( $\lambda$ ) adalah parameter panjang gelombang dari faktor sinusoidal. Nilainya ditentukan oleh piksel. Nilai yang benar adalah bilangan asli yang sama atau lebih besar dari 2. Nilai  $\lambda = 2$  tidak boleh digunakan dalam kombinasi dengan fase  $\phi = -90$  atau  $\phi = 90$  karena dalam kasus ini fungsi Gabor adalah sampel di penyeberangan nol. Untuk mencegah terjadinya efek

yang tidak diinginkan di perbatasan citra, nilai panjang gelombang harus lebih kecil dari seperlima dari ukuran citra. Lambda merupakan invers dari frekuensi gelombang dalam fungsi gabor dengan nilai  $f= 1/\lambda$ . Semakin besar nilai lambda akan memperbesar nilai batas sebuah gelombang.



Gambar 2.6 Filter gabor dengan nilai-nilai  $\lambda$  dari 5 (a), 10 (b) dan 15 (c) dengan  $\theta=0$ ,  $\varphi=0$ ,  $\gamma=0,5$ , dan  $b=1$ .

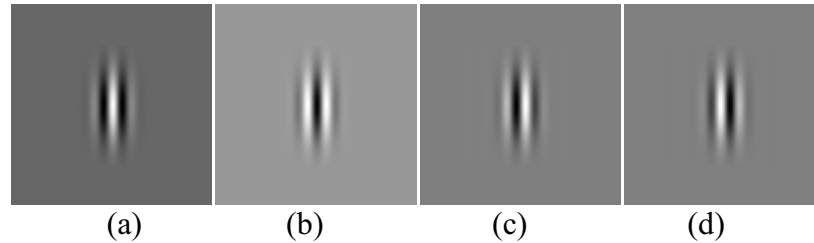
2. Tetha ( $\theta$ ) merupakan orientasi normal terhadap garis-garis paralel fungsi Gabor, nilainya ditentukan dalam derajat antara 0 dan 360. Untuk satu konvolusi tunggal, masukkan nilai satu orientasi dan ditetapkan nilai parameter dalam jumlah orientasi satu blok. Jika jumlah orientasi lebih dari satu dan disimbolkan,  $N > 1$ , maka  $N$  konvolusi akan dihitung sesuai distribusi antara 0 dan 360 derajat dengan penambahan sebesar  $360/N$ , mulai dari nilai orientasi awal yang ditentukan. Cara alternatif komputasi konvolusi ganda untuk orientasi yang berbeda membuat daftar nilai orientasi dipisahkan dengan koma (misalnya 0,45,110).



Gambar 2.7 Filter gabor dengan nilai-nilai  $\theta$  dari 0 (a), 45 (b) dan 90 (c) dengan  $\lambda =10$ ,  $\varphi=0$ ,  $\gamma=0,5$ , dan  $b=1$ .

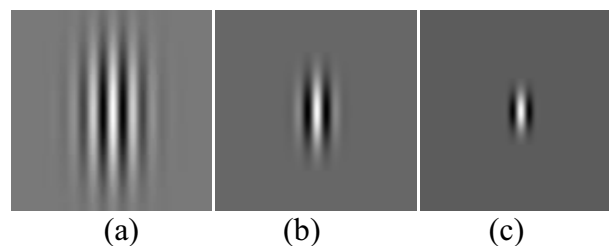
3. Phi ( $\varphi$ ) adalah fase offset sebagai faktor kosinus dalam fungsi gabor, nilainya dalam derajat antara -180 dan 180. Untuk nilai antara 0 dan 180 sesuai dengan fungsi pusat simetris sedangkan nilai antara -90 dan 90 sesuai dengan fungsi anti simetris. Jika satu nilai ditentukan, maka satu konvolusi orientasi

akan dihitung . Jika suatu daftar nilai yang diberikan (contoh : 0,90 nilai standar), maka konvolusi bertingkat orientasi akan dihitung, satu setiap nilai dari daftar dalam fase offset.



Gambar 2.8 Filter gabor dengan nilai-nilai  $\phi$  dari 0 (a), 180 (b), -90 (c) dan 90 (d) dengan  $\lambda = 10$ ,  $\theta = 0$ ,  $\gamma = 0,5$ , dan  $b = 1$ .

4. Sigma ( $\sigma$ ) standar deviasi dari Gaussian faktor menentukan ukuran (linear) dukungan dari fungsi gabor. Nilai  $\sigma$  tidak dapat ditentukan secara langsung tetapi dapat diubah hanya melalui nilai *bandwidth* ( $b$ ). Nilai *bandwidth* yang harus ditetapkan sebagai angka positif yang nyata dan standarnya adalah 1. Semakin kecil *bandwidth*, semakin besar  $\sigma$ .



Gambar 2.9 Filter gabor dengan nilai-nilai  $b$  dari 0.5 (a), 1 (b) dan 2 (c) dengan  $\lambda = 10$ ,  $\theta = 0$ ,  $\gamma = 0,5$ , dan  $\phi = 0$

*Bandwith* frekuensi spasial (dalam oktaf) dari filter Gabor berhubungan dengan rasio  $\sigma/\lambda$ , di mana persamaan *bandwith*, sebagai berikut:

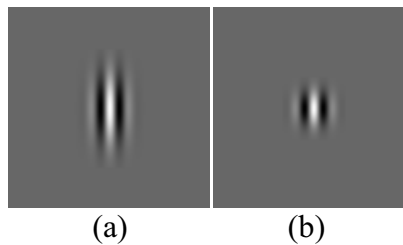
$$b = \log_2 \frac{\frac{\sigma}{\lambda} \pi + \frac{\ln 2}{2}}{\frac{\sigma}{\lambda} \pi - \frac{\ln 2}{2}} \quad (8)$$

$$\sigma = \frac{\lambda}{\pi} \frac{\ln 2}{2} * \frac{2^{b+1}}{2^b - 1}$$

Dimana untuk frekuensi  $f=1/\lambda$  maka  $\lambda=1/f$  dan

$$\sigma = \frac{1}{f\pi} \frac{\overline{\ln 2}}{2} * \frac{2^{b+1}}{2^{b-1}} \quad (9)$$

5. Gamma ( $\gamma$ ) adalah rasio aspek spasial yang menentukan elips dari bentuk fungsi Gabor. Untuk  $\gamma = 1$ , bentuknya adalah lingkaran. Untuk  $\gamma < 1$  bentuk yang terlihat memanjang dalam orientasi fungsi. Nilai standarnya adalah  $\gamma = 0.5$ .

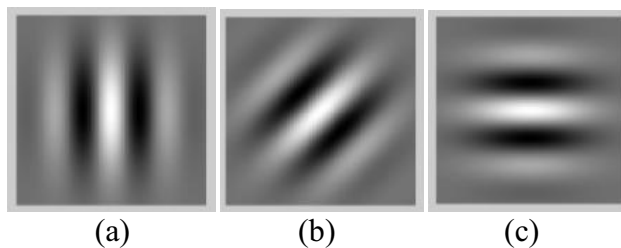


Gambar 2.10 Filter gabor dengan nilai-nilai  $\gamma$  dari 0.5 (a) dan 1 (b) dengan  $\lambda = 10$ ,  $\theta=0$ ,  $b=1$ , dan  $\varphi = 0$

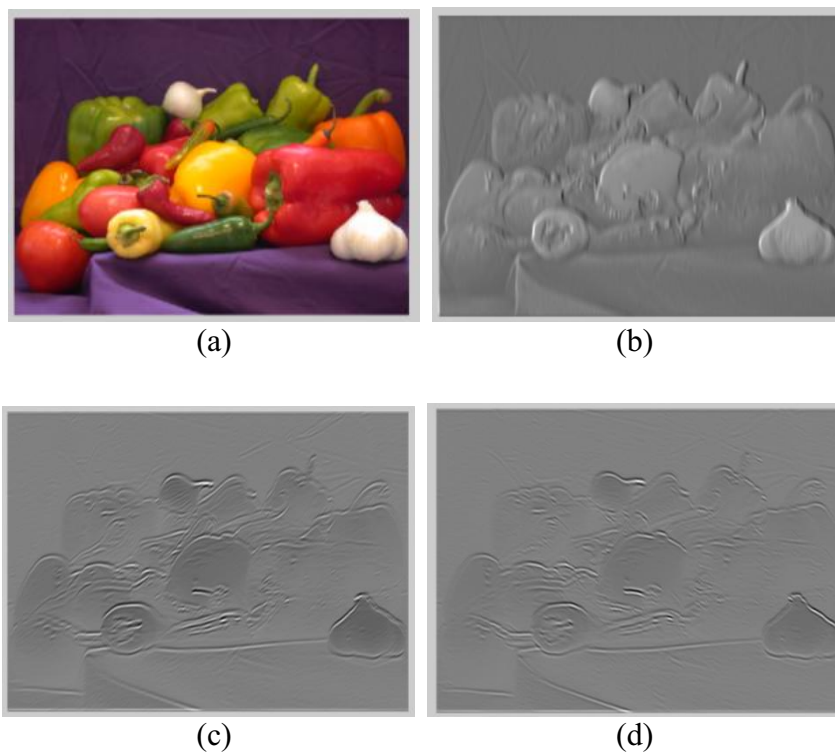
Algoritma fungsi gabor untuk membangun nilai filter dapat diperoleh melalui proses sebagai berikut:

- 1) Tentukan:
  - a. Frekuensi = Frekuensi ( $f$ );
  - b. Sudut orientasi normal = Tetha ( $\theta$ );
  - c. Faktor kosinus = Phi ( $\varphi$ )=[0,90];
  - d. nilai *bandwidth* ( $b$ ) = *bandwidth* ( $b$ )=1;
  - e. Rasio aspek spasial = Gamma ( $\gamma$ )=0.5;
- 2) Hitung fungsi gabor dengan persamaan (7).
- 3) Dari hasil nilai fungsi dikonvolusi dengan persamaan

$$c_{n1,n2} = \sum_{k1=1}^M \sum_{k2=1}^N a_{k1,k2} b_{n1-k1,n2-k2} \quad (10)$$



Gambar 2.11 Bank filter gabor dengan orientasi 0 (a), 45 (b), 90 (c) dan frekuensi 0,2



Gambar 2.12 Citra RGB (a) dan Filter gabor dengan orientasi 0 (b), 45 (c), 90 (d), frekuensi 0,2

### 2.2.2.3 Fitur Tekstur LBP

Tekstur digambarkan dalam bentuk mikro-primitif (*textons*) dan statistik penempatan aturan. *Textons* dapat digabungkan dengan ukuran pelengkap kontras citra lokal, yang mengukur kekuatan primitif. Local Binary Pattern (LBP) adalah operator tekstur pertama kali diperkenalkan sebagai ukuran pelengkap untuk kontras gambar lokal [20] dan mendeskripsikan tekstur pada citra *grayscale*. LBP bekerja mendeskripsikan tekstur secara lokal pada suatu blok tertentu di citra *grayscale*.

LBP membandingkan nilai biner piksel pada pusat citra dengan delapan nilai piksel disekelilingnya. Misal pada sebuah citra berukuran 3x3, nilai biner pada pusat citra dibandingkan dengan nilai sekelilingnya. Dengan cara mengurangkan nilai piksel pada pusat citra dengan nilai piksel disekelilingnya, jika hasilnya lebih atau sama dengan 0 maka diberi nilai 1 dan jika hasilnya kurang dari 0 maka diberi nilai. Setelah itu, menyusun delapan nilai biner searah jarum jam atau sebaliknya dan merubah delapan bit biner kedalam nilai desimal untuk menggantikan nilai piksel pada pusat citra. Piksel tetangga  $g_p$  adalah titik sampel merata atas lingkaran sejumlah P piksel dengan jari-jari R berpusat di pusat piksel  $g_c$ . Label LBP piksel pusat dengan koordinat (x,y) diperoleh dengan persamaan berikut [21]:

$$LBP_{P,R} x,y = \sum_{p=0}^{P-1} s_p (g_p - g_c) * 2^p \quad (11)$$

$$s_p = \begin{cases} 1; & x \geq 0 \\ 0; & x < 0 \end{cases}$$

Piksel 3 x 3	Threshold	Biner	Kode LBP																																				
<table border="1"> <tr><td>7</td><td>9</td><td>2</td></tr> <tr><td>5</td><td>6</td><td>12</td></tr> <tr><td>6</td><td>8</td><td>10</td></tr> </table>	7	9	2	5	6	12	6	8	10	<table border="1"> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td></td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	0	0		1	1	1	1	<table border="1"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>128</td><td></td><td>8</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	2	4	128		8	64	32	16	<table border="1"> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>123</td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>					123				
7	9	2																																					
5	6	12																																					
6	8	10																																					
1	1	0																																					
0		1																																					
1	1	1																																					
1	2	4																																					
128		8																																					
64	32	16																																					
	123																																						

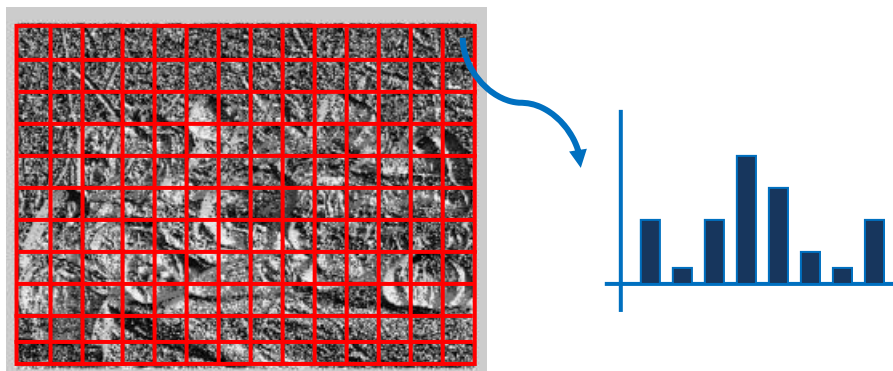
pattern = 0111011  
LBP = 1+2+8+16+32+64 = 123

Gambar 2.13 Perhitungan nilai LBP



Gambar 2.14 Citra asli (a) dan citra LBP (b)

Hasil nilai LBP terakumulasi dalam sebuah histogram dari 8 *Bins*, fitur diperoleh dengan menggabungkan LBP histogram yang diekstrak dari setiap blok.



Gambar 2.15 Histogram 8 *Bins* pada fitur LBP dari setiap blok

### 2.2.3 Fuzzy C-Means

Fuzzy C-means (FCM), atau dikenal juga sebagai Fuzzy ISODATA, merupakan salah satu metode mapping maupun clustering yang merupakan bagian dari metode Hard K-Means. FCM menggunakan model pengelompokan fuzzy sehingga data dapat menjadi anggota dari semua kelas atau cluster terbentuk dengan derajat atau tingkat keanggotaan yang berbeda antara 0 hingga 1.

Tingkat keberadaan data dalam suatu kelas atau cluster ditentukan oleh derajat keanggotaannya. Algoritma *clustering* FCM pertama kali diperkenalkan oleh Dunn 1973 dan kemudian dikembangkan oleh Bezdek 1981 [11]. FCM menghasilkan partisi yang optimal dengan meminimalkan bobot dalam kelompok fungsi objek berikut:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C U_{ij}^w \|x_i - v_j\|^2 \quad (12)$$

Solusi dari fungsi objek dapat diperoleh melalui proses iteratif, yang dilakukan sebagai berikut:

- 1) Input data yang akan diklaster  $X$ , berupa matriks berukuran  $n \times m$  ( $n$  = jumlah data,  $m$  = atribut data).  $X_{ij}$  = data ke- $i$  ( $i = 1, 2, \dots, n$ ), atribut ke- $j$  ( $j = 1, 2, \dots, m$ )
- 2) Tentukan:

- a. Jumlah kluster = c;
  - b. Pangkat/Derajat kefuzzy-an = w;
  - c. Maksimum iterasi = MaxIter;
  - d. Error terkecil yg diharapkan =  $\xi$ ;
  - e. Fungsi objektif awal =  $P_0 = 0$ ;
  - f. Iterasi awal =  $t = 1$ ;
- 3) Bangkitkan bilangan random  $\mu_{ik}$  antara 0-1 secara acak,  $i = 1,2,\dots,n$ ;  $k = 1,2,\dots,c$ ; sebagai elemen-elemen matriks partisi awal U.
- 4) Hitung pusat klaster ke-k:  $V_{kj}$ , dengan  $k = 1,2,\dots,c$ ; dan  $j = 1,2,\dots,m$ .

$$v_{kj} = \frac{\sum_{i=1}^n \mu_{ik}^w x_{ij}}{\sum_{i=1}^n \mu_{ik}^w} \quad (13)$$

- 5) Hitung fungsi obyektif pada iterasi ke-t

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \mu_{ik}^w \sum_{j=1}^m x_{ij} - v_{kj}^2 \quad (14)$$

- 6) Hitung perubahan matriks partisi U

$$\mu_{ik} = \frac{\sum_{j=1}^m x_{ij} - v_{kj}^2 \frac{-1}{w-1}}{\sum_{k=1}^c \sum_{j=1}^m x_{ij} - v_{kj}^2 \frac{-1}{w-1}} \quad (15)$$

- 7) Cek kondisi berhenti:

- a. Jika: ( $|P_t - P_{t-1}| < \xi$ ) atau ( $t > \text{MaxIter}$ ) maka berhenti;
- b. Jika tidak:  $t = t+1$ , ulangi langkah ke-4



Gambar 2.16 Citra warna ukuran 5 x 4

Untuk segmentasi citra gambar di atas membandingkan antar piksel yang homogen maka dari ukuran citra dua dimensi 5 x 4 dikonversi ke satu dimensi dan untuk vektor fitur dua dimensi merupakan atribut fitur dari tiap piksel citra. Segmentasi citra berdasarkan fitur warna dengan ruang warna RGB sebagai berikut:



1) Input data yang akan diklaster X, berupa matriks berukuran 20 x 3

Tabel 2.4 Tabel vektor fitur warna

	R	G	B
1	255	192	0
2	168	72	45
3	255	51	0
4	192	0	0
5	192	255	0
6	0	102	0
7	227	108	10
8	255	0	0
9	0	153	0
10	0	255	0
11	206	0	255
12	95	73	122
13	146	208	80
14	64	145	91
15	31	73	125
16	23	54	93
17	0	176	80
18	79	98	40
19	15	36	62
20	0	0	255

2) Tentukan:

- a. Jumlah klaster =  $c = 3$  (klaster merah, hijau, biru);
- b. Pangkat/Derajat kefuzzy-an =  $w = 2$ ;
- c. Maksimum iterasi =  $\text{MaxIter} = 100$ ;
- d. Error terkecil yg diharapkan =  $\xi = 10^{-5}$ ;
- e. Fungsi objektif awal =  $P_0 = 0$ ;
- f. Iterasi awal =  $t = 1$ ;

3) Bangkitkan bilangan random  $\mu_{ik}$  antara 0-1 secara acak

Tabel 2.5 Matrik partisi U awal

$\mu_{ik}$	k		
	i	1	2
1	0.5	0.3	0.2
2	0.6	0.3	0.1
3	0.5	0.4	0.1
4	0.7	0.2	0.1
5	0.3	0.6	0.1
6	0.3	0.5	0.2
7	0.5	0.4	0.1
8	0.5	0.2	0.3
9	0.1	0.7	0.2
10	0.3	0.6	0.1
11	0.4	0.1	0.5
12	0.2	0.1	0.7
13	0.3	0.6	0.1
14	0.2	0.5	0.3
15	0.1	0.4	0.5
16	0.1	0.3	0.6
17	0.1	0.7	0.2
18	0.3	0.6	0.1
19	0.2	0.3	0.5
20	0.1	0.2	0.7

- 4) Hitung pusat kluster ke-k:  $V_{kj}$ , dengan  $k = 1, 2, 3$ ; dan  $j = 1, 2, 3$  pada iterasi pertama

$$v_{kj} = \frac{\sum_{i=1}^{20} \mu_{ik}^2 * x_{ij}}{\sum_{i=1}^{20} \mu_{ik}^2}$$

Tabel 2.6 Pusat kluster ( $V_{kj}$ ) pada iterasi pertama

$V_{ki}$	J		
	K	1	2
1	182.7186312	80.23193916	33.01520913
2	79.63589744	150.3666667	40.6974359
3	67.46215139	52.15139442	136.1952191

- 5) Hitung fungsi obyektif pada iterasi pertama

$$P_t = \sum_{i=1}^{20} \sum_{k=1}^3 \sum_{j=1}^3 x_{ij} - v_{kj}^2 \mu_{ik}^2 = 142604.492$$

- 6) Hitung perubahan matriks partisi U, diperoleh

$$\mu_{ik} = \frac{\sum_{j=1}^3 x_{ij} - v_{kj}^2 \frac{-1}{2-1}}{\sum_{k=1}^3 \sum_{j=1}^3 x_{ij} - v_{kj}^2 \frac{-1}{2-1}}$$

Tabel 2.7 Perubahan matriks partisi U setelah iterasi pertama

$\mu_{ik}$	K		
	k1	k2	k3
1	0.553254601	0.304751831	0.141993568
2	0.949316204	0.029090905	0.021592891
3	0.767457829	0.130124189	0.102417982
4	0.707528356	0.146010474	0.14646117
5	0.373273317	0.469290044	0.157436639
6	0.174006346	0.588296782	0.237696872
7	0.828719978	0.110534809	0.060745213
8	0.686008569	0.158996386	0.154995045
9	0.139593963	0.693537623	0.166868414
10	0.183694435	0.630403461	0.185902104
11	0.309863036	0.206284552	0.483852413
12	0.074325554	0.090737418	0.834937028
13	0.267672114	0.574021316	0.15830657
14	0.093017111	0.718331794	0.188651095
15	0.050689327	0.103398944	0.845911729
16	0.093426864	0.182785908	0.723787228
17	0.122221081	0.640388972	0.237389947
18	0.166042636	0.673214383	0.160742981
19	0.156819275	0.273825353	0.569355372
20	0.157322997	0.187206311	0.655470693

- 7) Cek kondisi berhenti karena ( $|P_1 - P_0|$ ) =  $|142604.492 - 0| = 142604.492 > 10^{-5}$  dan ( $1 < 100$ ), maka ulangi langkah ke-4 dengan  $t = 2$ .
- 8) Hitung pusat kluster ke-k:  $V_{kj}$ , dengan  $k = 1, 2, 3$ ; dan  $j = 1, 2, 3$  pada iterasi kedua

Tabel 2.8 Pusat kluster ( $V_{kj}$ ) pada iterasi kedua

k	J		
	1	2	3
1	205.0842446	76.09777921	22.80756808
2	60.21904426	156.5512174	43.73602105
3	56.28831281	59.52169464	128.6465591

- 9) Hitung fungsi obyektif pada iterasi kedua

$$P_t = \sum_{i=1}^{20} \sum_{k=1}^3 \sum_{j=1}^3 x_{ij} - v_{kj}^2 \mu_{ik}^2 = 120329.671$$

- 10) Hitung perubahan matriks partisi U, diperoleh

Tabel 2.9 Perubahan matriks partisi U setelah iterasi kedua

$\mu_{ik}$	K		
	k1	k2	k3
1	0.615947692	0.246400983	0.137651325
2	0.835834115	0.083931377	0.080234508
3	0.88003546	0.062847113	0.057117427
4	0.75966824	0.112460875	0.127870885
5	0.388285033	0.438224725	0.173490242
6	0.123626519	0.627941647	0.248431834
7	0.917804812	0.048719597	0.033475591
8	0.77850248	0.106476036	0.115021483
9	0.08742079	0.76360251	0.148976699
10	0.139190965	0.681598841	0.179210194
11	0.324179289	0.214122448	0.461698263
12	0.065473748	0.100468208	0.834058044
13	0.257831699	0.550265902	0.191902399
14	0.060096819	0.739410034	0.200493148
15	0.018987339	0.053599458	0.927413202
16	0.050759669	0.136632385	0.812607945
17	0.069964289	0.727450569	0.202585143
18	0.141138862	0.620075405	0.238785733
19	0.108875522	0.252865084	0.638259395
20	0.145219318	0.203087171	0.65169351

11) Cek kondisi berhenti karena  $(|P_2 - P_1|) = |120329.671 - 142604.492| = 22274.82099 > 10^{-5}$  dan  $(2 < 100)$ , maka ulangi langkah ke-4 dengan  $t = 3$ .

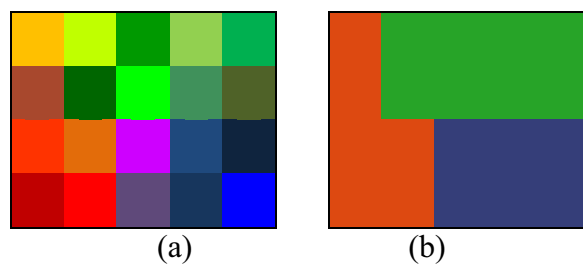
Iterasi segmentasi citra akan berhenti pada iterasi kedua puluh lima dengan kondisi berhenti  $(|P_{25} - P_{24}|) = |115645.5263612 - 115645.5263663| = 0.0000051 < 10^{-5}$ . Dan terbentuk pusat kluster pertama dengan nilai RGB [221, 73, 17], kluster kedua RGB [39, 164, 40] dan kluster ketiga RGB [53, 62, 121]. Partisi matrik U menentukan nilai keanggotaan piksel terhadap kluster untuk segmentasi citra yang mendekati satu.

Tabel 2.10 Pusat kluster ( $V_{kj}$ ) iterasi kedua puluh lima

$V_{kj}$	J		
	1	2	3
1	220.6632973	73.11344591	17.07222176
2	38.57857427	163.5394109	40.39549639
3	52.76381322	61.65766438	121.0825243

Tabel 2.11 Matriks partisi U setelah iterasi kedua puluh lima dan pembagian kluster

$\mu_{ik}$	k			Kluster
	k1	k2	k3	
1	0.652853871	0.206725643	0.140420486	1
2	0.753742658	0.106529246	0.139728096	1
3	0.936990221	0.03003196	0.032977819	1
4	0.772172379	0.096063997	0.131763625	1
5	0.40024931	0.408136079	0.191614611	2
6	0.092382666	0.666309886	0.241307448	2
7	0.941410832	0.031146947	0.027442221	1
8	0.82957116	0.07517626	0.095252579	1
9	0.049305518	0.844839596	0.105854886	2
10	0.103709772	0.741052657	0.155237572	2
11	0.334278554	0.206110639	0.459610807	3
12	0.060632473	0.090078314	0.849289213	3
13	0.26952092	0.495439447	0.235039633	2
14	0.06528984	0.646805473	0.287904687	2
15	0.012317057	0.038055881	0.949627061	3
16	0.033231495	0.100081307	0.866687198	3
17	0.041166542	0.81046411	0.148369348	2
18	0.141862469	0.507533594	0.350603937	2
19	0.084468194	0.223241935	0.692289871	3
20	0.142815377	0.212713347	0.644471276	3

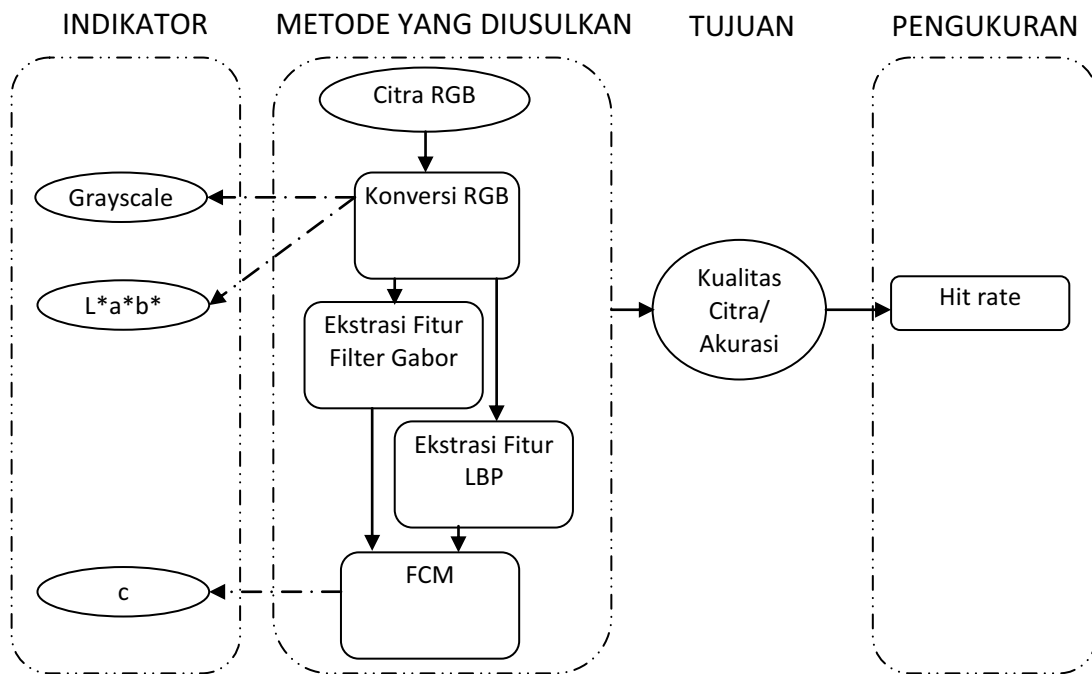


Gambar 2.17 Citra asal (a) dan telah disegmentasi (b)

### 2.3 Kerangka Penelitian

Penelitian ini didasari karena untuk menginterpretasi sebuah citra dalam memperoleh diskripsi tentang citra melalui beberapa proses antara lain *preprocessing*, segmentasi citra, analisis citra, dan interpretasi citra. Segmentasi

citra menjadi landasan untuk proses analisis dan pengenalan citra. Segmentasi membagi citra ke beberapa wilayah yang unik berdasarkan piksel citra yang homogen dan menjadi topik yang masih banyak diteliti.



Gambar 2.18 Kerangka Penelitian

Penelitian berawal dari citra warna dengan ruang warna RGB yang memiliki fitur warna dan tekstur yang akan disegmentasi. Dengan menggabungkan fitur warna dan tekstur pada citra akan bermanfaat dalam membedakan wilayah memiliki warna yang sama tetapi tekstur yang berbeda, atau sebaliknya. Berdasarkan studi literatur diketahui bahwa metode segmentasi yang mudah diterapkan untuk segmentasi citra adalah metode *clustering* dengan menggunakan algoritma fuzzy c-means, untuk mengekstasi fitur warna menggunakan transformasi ruang warna  $L^*a^*b^*$  dan tekstur menggunakan filter gabor. Filter gabor mempunyai kelebihan pada tekstur makro tetapi mempunyai kelemahan pada tekstur mikro maka akan diintegrasikan dengan menggunakan local binary patterns. Diharapkan dengan mengintegrasikan filter gabor dan local binary patterns untuk ekstraksi fitur tekstur pada tahapan segmentasi citra menjadi lebih akurat.