

MAKALAH SEMINAR TUGAS AKHIR  
**ANALISIS TEKSTUR MENGGUNAKAN  
 METODE *RUN LENGTH***  
 Mita Indriani\*, Imam Santoso\*\*, Yuli Christyono \*\*

**Abstrak** - Seiring dengan berkembangnya teknologi pengolahan citra digital, manusia menginginkan agar mesin, dalam hal ini adalah komputer, dapat mengenali citra seperti layaknya penglihatan manusia. Salah satu cara untuk mengenali citra adalah dengan membedakan tekstur citra tersebut, sehingga citra rumput dan citra jerami dapat dibedakan dengan mudah. Tekstur citra dapat dibedakan oleh kerapatan, keseragaman, keteraturan, kekasaran, dan lain-lain. Karena komputer tidak dapat membedakan tekstur seperti halnya penglihatan manusia, maka digunakan analisis tekstur untuk mengetahui pola suatu citra digital. Analisis tekstur akan menghasilkan nilai dari ciri atau karakteristik tekstur yang kemudian dapat dioleh oleh komputer untuk proses klasifikasi.

Salah satu metode analisis tekstur yang dapat digunakan adalah metode *run length*. Metode ini mengenali banyaknya *run* pada suatu pixel dengan level intensitas sama yang berurutan dalam satu arah tertentu. Ciri tekstur dapat diperoleh dari SRE (*Short Run Emphasis*), LRE (*Long Run Emphasis*), GRU (*Grey Level Uniformity*), RLU (*Run Length Uniformity*), dan RPC (*Run Percentage*). Pada program simulasi dengan menggunakan bahasa pemrograman Matlab, hasil analisis tekstur yang berupa lima ciri tersebut kemudian digunakan untuk klasifikasi dengan menggunakan metode LDA (*Linear Discriminant Analysis*) dan k-NN (*k-Nearest Neighbor*).

Dari hasil penelitian diketahui bahwa metode *run length* ini digunakan untuk membedakan tekstur halus dan tekstur kasar. Tekstur halus akan menghasilkan nilai SRE, RLU, dan RPC yang besar dan nilai LRE kecil. Sebaliknya tekstur kasar akan menghasilkan nilai LRE besar namun nilai SRE, RLU, dan RPC yang kecil. Setelah dilakukan klasifikasi dengan metode LDA dan k-NN dengan  $k=1, 3, 5,$  dan  $7$ , diperoleh hasil klasifikasi terbaik menggunakan metode k-NN untuk  $k=1$  dengan tingkat kesalahan klasifikasi sebesar  $0\%$ .

**Kata kunci:** tekstur, analisis tekstur, *run length*, klasifikasi, LDA, k-NN.

## I. PENDAHULUAN

### 1.1 Latar Belakang

Analisis tekstur memegang peranan penting dalam pengolahan citra digital karena analisis tekstur dikembangkan dengan tujuan agar komputer dapat memahami, membuat model, serta memproses tekstur

untuk dapat menirukan proses pembelajaran mata atau penglihatan manusia.

Tekstur dapat dianggap sebagai pengelompokan kesamaan di dalam suatu citra. Sifat-sifat subpola lokal tersebut menimbulkan cahaya yang diterima, keseragaman, kerapatan, kekasaran, keteraturan, kelinieran, frekuensi, fase, keterarahan, ketidakteraturan, kehalusan, dan lain-lain. Karena komputer tidak memiliki indra penglihatan, maka komputer hanya mengetahui pola suatu citra digital dari ciri atau karakteristik teksturnya. Ciri atau karakteristik tekstur diperoleh melalui proses ekstraksi ciri dengan berbagai metode seperti metode *co-occurrence*, autokorelasi, *wavelet*, frekuensi tepi, *run length*, dan lain sebagainya.

Dalam Tugas Akhir ini, metode yang digunakan untuk ekstraksi ciri adalah metode *run length*. Kemudian diteruskan dengan klasifikasi menggunakan metode klasifikasi linier LDA (*Linear Discriminant Analysis*) dan k-NN (*k-Nearest Neighbor*).

### 1.2 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat program simulasi analisis tekstur dari suatu citra menggunakan metode *run length* dengan klasifikasi LDA dan k-NN.

### 1.3 Pembatasan Masalah

Untuk menyederhanakan pembahasan pada Tugas Akhir ini, masalah dibatasi sebagai berikut.

1. Data citra untuk pembelajaran diambil dari *Vision Texture (VisTex) Database*. Data VisTex yang digunakan hanya 5 kelas dari 19 kelas yang ada dengan jumlah data sebanyak 38 citra.
2. Analisis tekstur menggunakan metode ekstraksi ciri *grey level run length* dengan klasifikasi LDA dan k-NN.
3. Program bantu yang digunakan dalam membuat tugas akhir ini adalah Matlab R14.

## II. LANDASAN TEORI

### 2.1 Analisis Tekstur

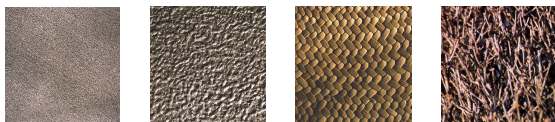
Secara umum tekstur mengacu pada repetisi elemen-elemen tekstur dasar yang sering disebut primitif atau *texel (texture element)*. Suatu *texel* terdiri dari beberapa *pixel* dengan aturan posisi bersifat periodik, kuasiperiodik, atau acak.

\* Mahasiswa Jurusan Teknik Elektro UNDIP

\*\* Staf Pengajar Jurusan Teknik Elektro UNDIP

Syarat-syarat terbentuknya tekstur setidaknya ada dua, yaitu:

1. Adanya pola-pola primitif yang terdiri dari satu atau lebih pixel. Bentuk-bentuk pola primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
2. Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulangannya.



(a) Halus (b) Kasar (c) Teratur (d) Tak Teratur  
Gambar 1. Contoh tekstur dari VisTex Database

Ada dua pendekatan yang digunakan untuk menganalisis tekstur dari suatu area, yaitu statistis dan struktural. Pendekatan statistis mempertimbangkan bahwa intensitas dibangkitkan oleh medan acak dua dimensi. Metode ini berdasar pada frekuensi-frekuensi ruang (*spatial*) dan menghasilkan karakterisasi tekstur seperti halus, kasar, dan lain-lain. Contoh metode statistik adalah *run length*, autokorelasi, *co-occurrence*, transformasi Fourier, frekuensi tepi, dan metode Law (pengukuran energi tekstur). Sedangkan teknik struktural berkaitan dengan penyusunan bagian-bagian terkecil (primitif) suatu citra, contoh metode ini adalah model fractal.

## 2.2 Metode Run Length

Metode *run length* menggunakan distribusi suatu *pixel* dengan intensitas yang sama secara berurutan dalam satu arah tertentu sebagai primitifnya. Masing-masing primitif didefinisikan atas panjang, arah, dan level keabuan. Panjang dari primitif tekstur pada arah yang berbeda dapat digunakan untuk menggambarkan suatu tekstur.

Analisis tekstur metode *run length* ini digunakan untuk membedakan citra halus dan citra kasar. Tekstur kasar menunjukkan banyak *pixel* tetangga yang memiliki intensitas yang sama sedangkan tekstur halus menunjukkan sedikit *pixel* tetangga yang menunjukkan intensitas yang sama.

Untuk melakukan ekstraksi ciri dengan menggunakan metode *run length*, citra aras keabuan dengan matriks  $f(x,y)$  harus ditransformasikan terlebih dahulu kedalam matriks *grey level run length* (GLRL),  $B(a,r)$ .

$$f(x,y) \xrightarrow{GLRL} B(a,r) \quad (1)$$

Elemen matriks dari GLRL  $B(a,r)$  menghitung banyaknya primitif dengan panjang  $r$  dan level keabuan  $a$ . Jumlah dari primitif dapat diperoleh dengan persamaan (2).

$$K = \sum_{a=1}^L \sum_{r=1}^{Nr} B(a,r) \quad (2)$$

dengan  $L$  : banyaknya level keabuan dari citra  
 $Nr$  : panjang maksimal dari primitif  
 $K$  : jumlah *run*  
 $M,N$  : dimensi citra

Adapun ciri dari tekstur dapat diperoleh dari persamaan-persamaan berikut ini:

### 1. Short Run Emphasis (SRE)

SRE mengukur distribusi *short run*. SRE sangat tergantung pada banyaknya *short run* dan diharapkan bernilai besar pada tekstur halus.

$$SRE = \frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{Nr} \frac{B(a,r)}{r^2} \quad (3)$$

### 2. Long Run Emphasis (LRE)

LRE mengukur distribusi *long run*. LRE sangat tergantung pada banyaknya *long run* dan diharapkan bernilai besar pada tekstur kasar.

$$LRE = \frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{Nr} r^2 B(a,r) \quad (4)$$

### 3. Grey Level Uniformity (GLU)

GLU mengukur persamaan nilai derajat keabuan di seluruh citra dan diharapkan bernilai kecil jika nilai derajat keabuan serupa di seluruh citra.

$$GLU = \frac{1}{K} \sum_{a=1}^L \left( \sum_{r=1}^{Nr} B(a,r) \right)^2 \quad (5)$$

### 4. Run Length Uniformity (RLU)

RLU mengukur persamaan panjangnya *run* di seluruh citra dan diharapkan bernilai kecil jika panjangnya *run* serupa di seluruh citra.

$$RLU = \frac{1}{K} \sum_{r=1}^{Nr} \left( \sum_{a=1}^L B(a,r) \right)^2 \quad (6)$$

### 5. Run Percentage (RPC)

RPC mengukur keserbasamaan dan distribusi *run* dari sebuah citra pada arah tertentu. RPC bernilai paling besar jika panjangnya *run* adalah 1 untuk semua derajat keabuan pada arah tertentu.

$$RPC = \frac{K}{\sum_{a=1}^L \sum_{r=1}^{Nr} r B(a,r)} = \frac{K}{MN} \quad (7)$$

Citra aras keabuan  $f(x,y)$

Run Length  $B(a,r)$

0	1	2	3	0°	1	2	3	4
0	2	3	3	0	4	0	0	0
1	1	1	1	1	1	0	1	0
2	3	0	0	2	3	0	0	0
3	0	3	0	3	3	1	0	0

Gambar 2. Contoh citra dan matriks *run length* pada arah 0°

## 2.3 Pengenalan Pola

Pengenalan pola merupakan ilmu mengenai diskripsi atau klasifikasi dari hasil ekstraksi ciri yang

memetakan suatu fitur, yang merupakan ciri utama suatu obyek (yang dinyatakan dalam sekumpulan bilangan-bilangan) ke suatu kelas yang sesuai. Ada 3 metode pengenalan pola, yaitu:

1. Metode statistik

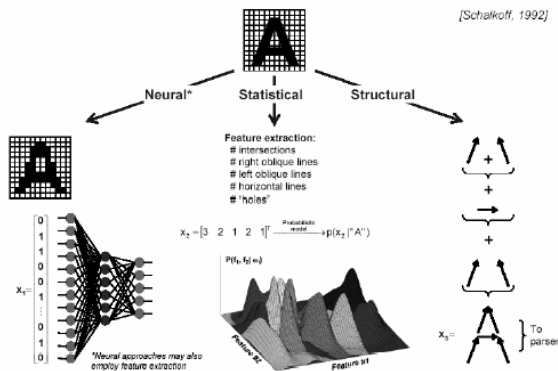
Model statistik didefinisikan sebagai sebuah keluarga dari fungsi kerapatan peluang bersyarat kelas  $Pr(x | c_i)$ , yakni peluang vektor fitur  $x$  jika diberikan kelas  $c_i$ . Metode ini dapat dilakukan dengan pendekatan *supervised* maupun pendekatan *unsupervised*. Pendekatan *supervised* (dengan pengarah) menyediakan *training set* untuk mengarahkan atau memberi informasi atau pengetahuan tentang kelas-kelas yang ada. Contoh pendekatan *supervised* adalah metode k-NN, Bayes, Linear Discriminant Analysis, dan lain-lain. Pendekatan *unsupervised* (tanpa pengarah) tidak menyediakan *training set*. Informasi yang disediakan adalah jumlah kluster yang ada. Sehingga pengelompokan dilakukan sepenuhnya berdasarkan karakteristik data. Contoh dari pendekatan ini adalah metode k-Mean.

2. Metode sintaktik

Dalam metoda sintaktik atau metoda struktural, pola dipilah berdasarkan keserupaan ukuran struktural, 'pengetahuan' direpresentasikan secara formal grammar atau deskripsi relasional (*graf*). Metode ini dipakai tidak hanya untuk pemilahan, tetapi juga untuk deskripsi. Biasanya, metode ini memformulasikan deskripsi hirarkis dari pola kompleks yang tersusun dari pola bagian yang lebih sederhana.

3. Metode jaringan syaraf

Dalam metode jaringan syaraf, pemilahan dilakukan berdasarkan tanggapan suatu neuron jaringan pengolah sinyal (*neuron*) terhadap stimulus masukan (pola). 'Pengetahuan' disimpan dalam sambungan antar neuron dan kekuatan pembobot sinaptik

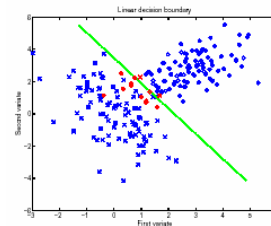


Gambar 3. Perbandingan teknik pengenalan pola

2.4 Linear Discriminant Analysis (LDA)

Tujuan utama dari analisis diskriminan adalah untuk memperoleh kaidah matematis, yang dikenal dengan fungsi diskriminan, yang dapat digunakan untuk

memisahkan kelompok obyek yang berbeda, seperti kelompok air dan pasir. Fungsi diskriminan ditentukan oleh parameter statistik yang tergambar dari populasi ciri obyek pada kelas yang telah diketahui. Vektor ciri yang telah diperoleh dari obyek yang akan diklasifikasikan dipergunakan sebagai masukan. Keluarannya biasanya bernilai skalar yang dapat digunakan untuk menentukan kelas yang paling memungkinkan. Fungsi diskriminan menetapkan permukaan keputusan dari n-dimensi yang memisahkan kelas-kelas distribusi ciri pada n-dimensi ruang ciri.



Gambar 4. Contoh batas keputusan LDA pada ruang ciri bivarian

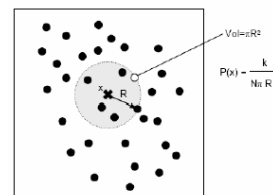
2.5 k-Nearest Neighbor (k-NN)

Metode k-NN merupakan pengembangan dari estimasi kerapatan non-parametrik. Ekspresi umum dari estimasi fungsi kerapatan peluang non-parametrik dapat dituliskan dengan rumus berikut ini.

$$p(\underline{x}) \cong \frac{k}{NV} \tag{8}$$

dengan  $V$  volume yang melingkupi  $\underline{x}$   
 $N$  jumlah total sampel  
 $k$  jumlah sampel di dalam  $V$

Estimasi kerapatan non-parametrik bisa dihitung dengan dua cara. Cara pertama adalah dengan memilih nilai tetap volume  $V$  dan menentukan  $k$  dari data. Hal ini dilakukan dalam metoda yang disebut estimasi kerapatan kernel (KDE atau *Kernel Density Estimation*). Cara yang kedua yaitu dengan memilih jumlah tetap  $k$  dan menentukan volume  $V$  yang sesuai dari data. Hal ini menghasilkan metoda k-buah tetangga terdekat (k-NN atau *k-Nearest Neighbor*).



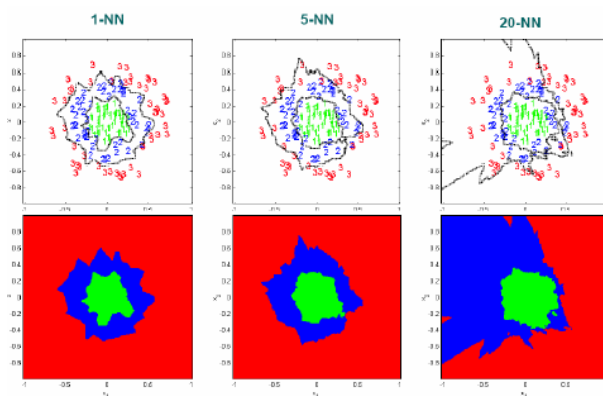
Gambar 5. Ilustrasi konsep tetangga terdekat pada ruang 2 dimensi

Keuntungan utama dari metoda k-NN adalah metoda ini memberikan pendekatan yang sangat sederhana untuk pemilahan optimal Bayes. Apabila terdapat suatu data masukan yang jumlahnya  $N$  buah dimana  $N_i$  buah dari kelas  $\omega_i$  dan dilakukan pemilahan sampel  $\underline{x}$  yang tak diketahui, maka dapat digambarkan suatu bola (*hyper*) dengan volume  $V$  disekitar  $\underline{x}$ . Bila

volume ini mengandung  $k$ -buah data, dimana  $k_i$ -buah diantaranya dari kelas  $\omega_i$ , maka fungsi kebolehjadian dapat didekati dengan metoda  $k$ -NN dengan  $p(\underline{x} | \omega_i) = k_i / N_i V$ . Dengan cara yang sama, fungsi kerapatan tak-kondisional diestimasi oleh  $p(\underline{x}) = k / NV$ . Sedangkan prior-nya didekati dengan rumus  $p(\omega_i) = N_i / N$ . Setelah dikumpulkan, pemilahan Bayes menjadi

$$p(\omega_i | \underline{x}) = \frac{p(\underline{x} | \omega_i) p(\omega_i)}{p(\underline{x})} = \frac{(k_i / N_i V)(N_i / N)}{(k / NV)} = \frac{k_i}{k} \quad (9)$$

Pada metode  $k$ -NN, penggunaan nilai  $k$  yang besar memiliki keuntungan, yaitu menyediakan informasi probabilistik. Akan tetapi, pengambilan  $k$  yang terlalu besar akan merusak lokalitas estimasi dan juga meningkatkan beban komputasi. Berikut ini diperlihatkan pengaruh pemilihan harga  $k$ .



Gambar 6. Ilustrasi pengaruh pemilihan  $k$

Pemilahan  $k$ -NN dilakukan dengan mencari  $k$ -buah tetangga terdekat dan memilih kelas dengan  $k_i$  terbanyak pada kelas  $\omega_i$ . Adapun langkah pemilahan  $k$ -NN adalah sebagai berikut:

1. Jika sekumpulan vektor data *training* memiliki  $N$  titik data secara keseluruhan, maka kenallah  $k$ -buah tetangga terdekat dari sampel  $\underline{x}$  dimana  $k$  merupakan bilangan ganjil.
2. Dari  $k$ -buah tetangga terdekat tersebut, identifikasikan jumlah vektor  $k_i$  pada kelas  $\omega_i$ ,  $i=1,2,\dots,M$  dengan  $\sum_i k_i = k$ .
3. Masukkan  $\underline{x}$  pada kelas  $\omega_i$  dengan nilai  $k_i$  maksimum.
4. Jika terdapat dua atau lebih kelas  $\omega_i$  yang memiliki  $E$  tetangga terdekat, maka terjadilah kondisi seimbang (konflik) dan digunakan strategi pemecahan konflik.
5. Untuk masing-masing kelas yang terlibat dalam konflik, tentukan jarak  $d_i$  antara  $\underline{x}$  dengan kelas  $\omega_i$  berdasarkan  $E$  tetangga terdekat yang ditemukan pada kelas  $\omega_i$ .
6. Pola *training* dengan sejumlah  $m$ -buah dari kelas  $\omega_i$ , dituliskan sebagai  $y^{im} = \{y_1^{im}, \dots, y_N^{im}\}$ , kemudian jarak antara  $\underline{x}_u$  dengan kelas  $\omega_i$  adalah

$$d_i = \frac{1}{E} \sum_{j=1}^E |(x_j - y_j^{im})| \quad (10)$$

7. Masukkan  $\underline{x}$  ke dalam kelas dengan jarak di paling kecil. Contohnya,  $\underline{x} \in \omega_C$ , jika  $d_C < d_i$ , untuk  $i, C \in [1, \dots, M]$  dan  $i \neq C$ .

### III. PERANCANGAN DAN IMPLEMENTASI SISTEM

#### 3.1 Perangkat Keras

Perangkat keras yang digunakan untuk membuat dan menjalankan program adalah satu set komputer pribadi (PC) dengan spesifikasi sebagai berikut.

1. Sistem Komputer : Pentium IV 1,8 GHz
2. Sistem Operasi : Microsoft Windows XP Profesional
3. Media Tampilan : VGA, true color 32 bit  
800 × 600 piksel
4. Media masukan : papan ketik dan mouse
5. Memori : 256 MB RAM

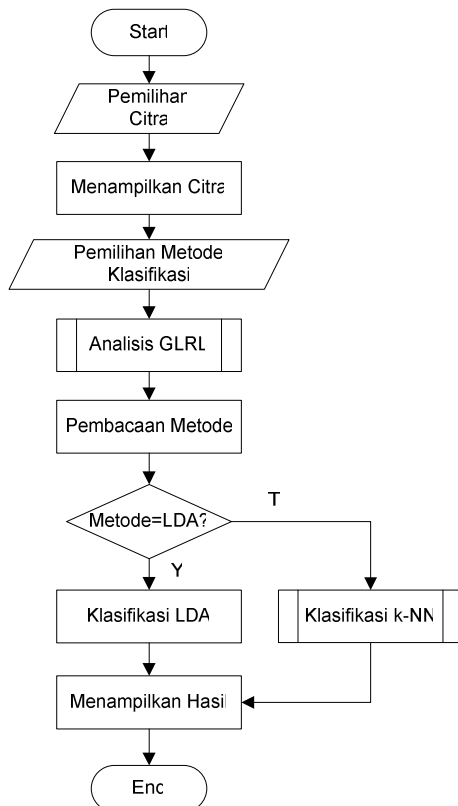
#### 3.2 Perangkat Lunak

Perangkat lunak yang digunakan adalah Matlab R14. Matlab R14 merupakan bahasa pemrograman yang bekerja dalam sistem operasi Windows. Agar dalam pembuatan program dapat dikompilasi, persyaratan minimal pembuatan program menggunakan perangkat lunak Matlab R14 dan sistem operasi Microsoft Windows XP Profesional harus dipenuhi. Hal ini untuk menghindari jika dalam perangkat lunak versi sebelumnya tidak terdapat beberapa fungsi yang ada pada versi sesudahnya.

#### 3.3 Diagram Alir Program

Roses perancangan program secara garis besar dibagi dalam 3 tahapan utama. Tahapan pertama adalah pemilihan citra, menampilkan citra, serta pemilihan metode klasifikasi. Tahap kedua dilanjutkan dengan analisis tekstur metode grey level run length untuk memperoleh ciri tekstur. Tahap terakhir adalah klasifikasi ciri yang telah diperoleh dengan menggunakan metode yang telah dipilih sebelumnya pada tahap pertama.

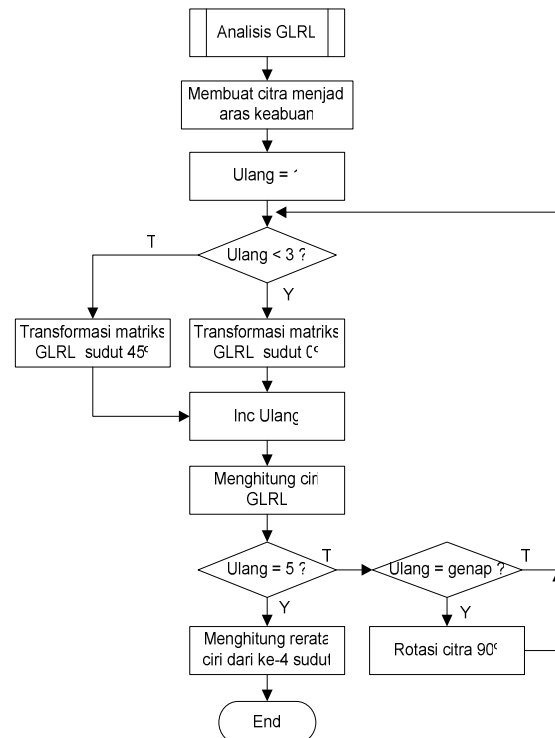
Diagram alir pembuatan program untuk melakukan analisis tekstur menggunakan metode *run length* ditunjukkan pada Gambar 7.



Gambar 7. Diagram alir program analisis tekstur

Pada tahap pertama mula-mula dilakukan pemilihan citra. Citra latih yang digunakan dalam program ini diambil dari basis data VisTex dengan ukuran  $512 \times 512$  pixel yang disimpan dalam format \*.ppm. Setelah citra dipilih citra langsung ditampilkan. Dan kemudian dilanjutkan dengan memilih metode klasifikasi yang akan digunakan.

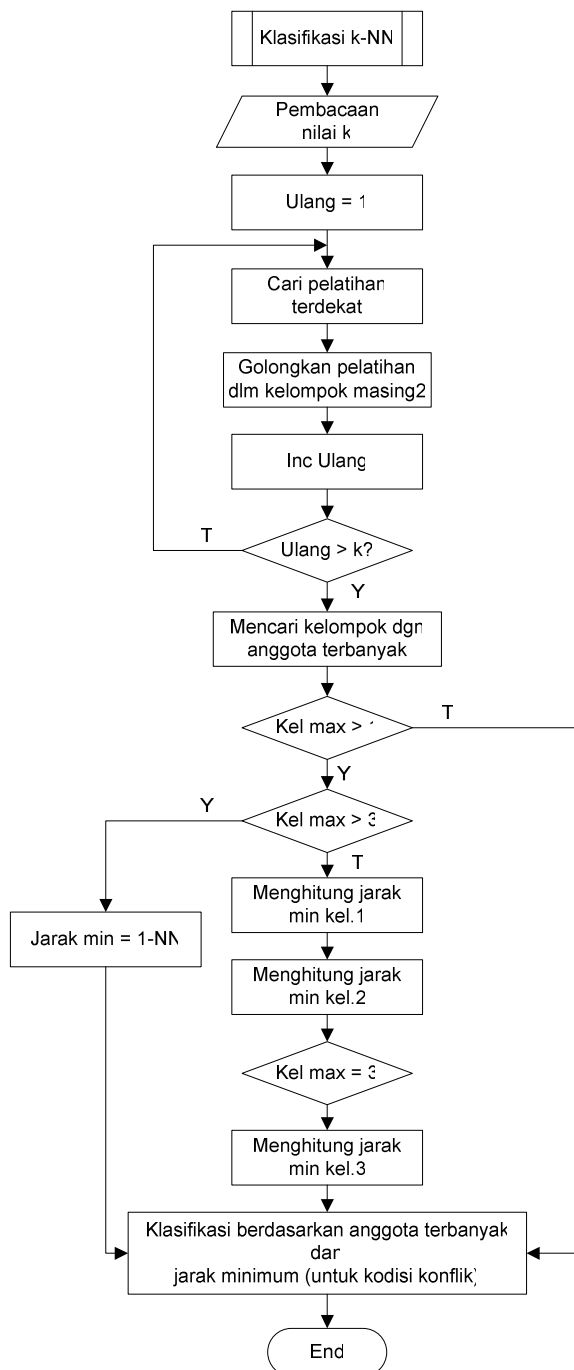
Tahap kedua dilanjutkan dengan melakukan analisis tekstur metode run length pada citra yang telah dipilih. Analisis run length ini dilakukan pada citra aras keabuan. Jika citra merupakan citra warna maka terlebih dahulu diubah menjadi citra aras keabuan. Citra aras keabuan akan diubah menjadi matriks  $B(a,r)$  untuk dapat dihitung cirinya seperti pada rumus (3) hingga rumus (7). Analisis dilakukan pada arah  $0^\circ$ ,  $90^\circ$ ,  $45^\circ$ , dan  $135^\circ$ . Dari keempat arah tersebut masing-masing dihasilkan 5 ciri, namun tidak semuanya digunakan untuk proses klasifikasi. Ciri yang digunakan untuk klasifikasi adalah rerata ciri dari keempat sudut pencarian.



Gambar 8. Diagram alir analisis GLRL

Tahap terakhir adalah melakukan klasifikasi. Ada 2 metode klasifikasi yang digunakan yaitu LDA dan k-NN. Jika metode yang dipilih adalah metode LDA, maka langsung dilakukan klasifikasi dengan bantuan *statistic toolbox* yang terdapat dalam Matlab sehingga langsung dapat diketahui hasil kelompoknya. Namun jika yang dipilih adalah metode k-NN, maka perlu dilakukan klasifikasi sesuai dengan prosedur k-NN sebagai berikut:

1. Nilai k yang dapat digunakan adalah 1, 3, 5, dan 7.
2. Setelah ditentukan nilai k, maka dilakukan pencarian tetangga terdekat sebanyak k buah. Kemudian dari k-tetangga terdekat tersebut, dilakukan identifikasi pelatihan  $k_i$  yang masuk dalam kelompok  $\omega_i$ .
3. Hasil klasifikasi ditentukan dengan nilai k terbanyak yang ada dalam kelompok.
4. Jika terdapat lebih dari 1 kelompok yang memiliki k maksimal, maka terjadilah konflik. Konflik ini dipecahkan dengan cara menghitung jarak minimal untuk masing-masing kelompok yang terlibat konflik.
5. Namun jika terdapat lebih dari 3 kelompok yang terlibat dalam konflik, maka jarak terdekat ditentukan oleh kelompok 1-NN (kelompok pelatihan pertama yang paling dekat). Hal ini dikarenakan bila ada lebih dari 3 kelompok yang terlibat konflik, maka masing-masing kelompok hanya terdiri dari 1 anggota saja sehingga dapat diputuskan kelompok terdekat adalah kelompok 1-NN



Gambar 9. Diagram alir klasifikasi k-NN

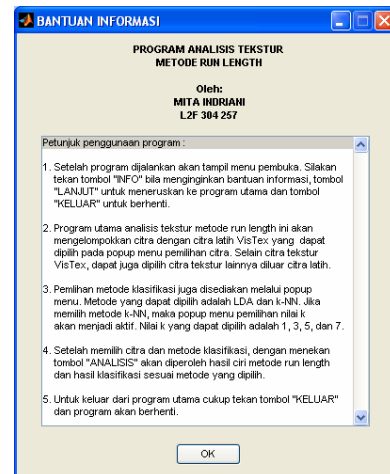
## IV. PENGUJIAN DAN ANALISIS

### 4.1 Pengujian Perangkat Lunak

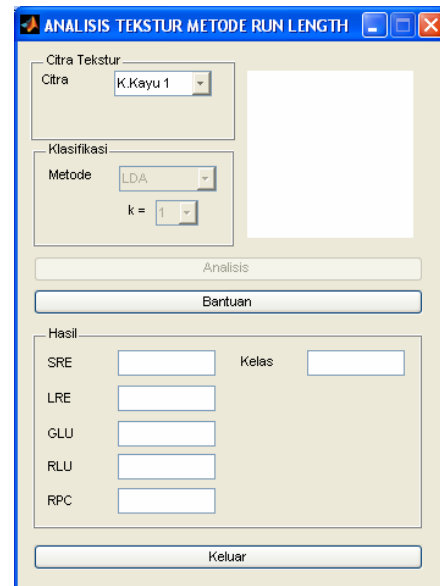
Pengujian perangkat lunak memiliki langkah yang telah ditentukan secara urut. Pengurutan langkah tersebut dimaksudkan selain untuk mempermudah pengoperasian program juga untuk menghindari kesalahan dalam pengoperasian.



Gambar 10. Tampilan awal program



Gambar 11. Tampilan informasi



Gambar 12. Tampilan menu utama

Gambar 12 menunjukkan tampilan menu utama program. Jendela utama program menampilkan hasil seluruh proses pengolahan untuk menganalisis tekstur menggunakan metode *run length*.

## 4.2 Analisis Metode Run Length

Hasil analisis tekstur pada citra latih VisTex yang dibagi dalam 5 kelompok dengan jumlah total 38 citra ditunjukkan pada Tabel 1 dibawah ini.

Tabel 1. Tabel ciri citra latih VisTex

Nama Citra	Hasil ciri Grey Level Run Length				
	SRE	LRE	GLU	RLU	RPC
Kulit kayu 1	0,964133	1,18122	1176,95	226649	0,948968
Kulit kayu 2	0,976267	1,11141	1726,87	238171	0,966908
Kulit kayu 3	0,954469	1,21115	2867,46	218328	0,938437
Kulit kayu 4	0,961082	1,21367	1798,06	223459	0,942543
Kulit kayu 5	0,985438	1,06106	1923,84	247293	0,980479
Kulit kayu 6	0,981150	1,08545	1590,62	242948	0,973979
Kulit kayu 7	0,981081	1,08452	1481,30	242917	0,974059
Kulit kayu 8	0,987504	1,05464	1328,19	249284	0,982862
Kulit kayu 9	0,986405	1,06106	1366,78	248164	0,981252
Logam 1	0,992537	1,03182	1804,96	254402	0,989831
Logam 2	0,983382	1,08469	2579,87	244957	0,975657
Logam 3	0,981590	1,09819	2398,41	243029	0,972685
Logam 4	0,981664	1,08849	2456,45	243306	0,973865
Logam 5	0,967631	1,23679	2277,12	228229	0,946009
Logam 6	0,958151	1,31121	2738,66	219139	0,930688
Pasir 1	0,980979	1,07943	3309,42	242954	0,974708
Pasir 2	0,970992	1,12586	3203,19	233432	0,961218
Pasir 3	0,967078	1,14361	3697,58	229793	0,956070
Pasir 4	0,976619	1,09875	3112,04	238753	0,968906
Pasir 5	0,970945	1,12561	3139,04	233372	0,961196
Pasir 6	0,964112	1,17776	2033,46	226584	0,949168
Pasir 7	0,953653	1,24034	1825,25	217007	0,933816
Ubin 1	0,935731	1,32471	3291,52	202042	0,911476
Ubin 2	0,867139	1,88131	3874,09	151367	0,813046
Ubin 3	0,856454	2,00209	3788,69	144138	0,796882
Ubin 4	0,958304	1,21416	1190,82	221223	0,940669
Ubin 5	0,965823	1,17078	3267,46	228115	0,951539
Ubin 6	0,918291	1,42550	11072,80	188269	0,888744
Ubin 7	0,952365	1,27551	4772,96	215357	0,929684
Ubin 8	0,949531	1,26937	5808,63	213290	0,928043
Air 1	0,979106	1,09905	1312,85	241101	0,970737
Air 2	0,928034	1,35431	4420,35	196358	0,903380
Air 3	0,924058	1,39656	5786,68	193745	0,897371
Air 4	0,959444	1,17945	4515,14	223014	0,946149
Air 5	0,962000	1,17831	1578,46	225084	0,947990
Air 6	0,950624	1,22261	6286,64	215405	0,936810
Air 7	0,952271	1,21485	5218,17	216975	0,936934
Air 8	0,970865	1,14487	1857,15	232977	0,959268

Keterangan: Yang dicetak biru merupakan nilai maksimum dan yang dicetak merah merupakan nilai minimum

### 4.2.1 SRE (Short Run Emphasis)

Nilai SRE sangat tergantung pada banyaknya sort run dan diharapkan bernilai besar pada tekstur halus. Karena terdapat sedikit pixel tetangga yang memiliki intensitas yang sama untuk tekstur halus, maka dapat dikatakan tekstur halus memiliki run yang pendek-pendek. Sedangkan tekstur kasar memiliki run

yang lebih panjang karena banyak pixel tetangga yang memiliki intensitas yang sama. Karena nilai SRE berbanding terbalik dengan run, maka semakin kecil run semakin besar nilai SRE.

Dari Tabel 4.1 dapat dilihat bahwa nilai SRE yang paling besar diperoleh untuk citra tekstur “Logam 1”, dan nilai SRE minimum pada citra “Ubin 3”. Hal ini menunjukkan bahwa citra tekstur yang paling halus adalah citra “Logam 1”. Kebalikan tekstur halus adalah kasar, maka tekstur dengan SRE minimum merupakan citra tekstur paling kasar, yaitu citra “Ubin 3”.

### 4.2.2 LRE (Long Run Emphasis)

Nilai LRE sangat tergantung pada banyaknya long run dan diharapkan bernilai besar pada tekstur kasar. Nilai LRE yang dihasilkan oleh tekstur kasar akan lebih besar bila dibandingkan dengan tekstur halus karena tekstur kasar memiliki run yang lebih panjang dan nilai run berbanding lurus dengan besarnya LRE sehingga semakin panjang run semakin besar nilai LRE.

Dari Tabel 4.1 dapat dilihat bahwa nilai LRE maksimum diperoleh untuk citra tekstur “Ubin 3”, dan nilai LRE minimum pada citra “Logam 1”. Hal ini menunjukkan bahwa Nilai LRE ini berkebalikan dengan nilai SRE. Citra tekstur yang paling halus adalah citra menghasilkan nilai SRE maksimum dan nilai LRE minimum. Kebalikannya, citra tekstur kasar menghasilkan SRE minimum dan LRE maksimum.

### 4.2.3 GLU (Grey Level Uniformity)

GLU mengukur persamaan nilai derajat keabuan di seluruh citra dan diharapkan bernilai kecil jika nilai derajat keabuan serupa di seluruh citra. Dari Tabel 4.1 dapat dilihat bahwa nilai GLU maksimum diperoleh untuk citra tekstur “Ubin 6”, dan nilai LRE minimum pada citra “Kulit kayu 1”.

### 4.2.4 RLU (Run Length Uniformity)

RLU mengukur persamaan panjangnya run di seluruh citra dan diharapkan bernilai kecil jika panjangnya run serupa di seluruh citra. Dari Tabel 4.1 dapat dilihat bahwa nilai RLU maksimum diperoleh untuk citra tekstur “Logam 1”, dan nilai RLU minimum pada citra “Ubin 3”. Karena citra “Logam 1” merupakan citra dengan tekstur paling halus, maka dapat disimpulkan citra halus cenderung memiliki distribusi run yang mengumpul pada run pendek untuk semua derajat keabuannya. Kebalikannya, citra “Ubin 3” yang merupakan citra dengan tekstur paling kasar, cenderung memiliki distribusi run yang lebih menyebar untuk semua derajat keabuannya.

### 4.2.5 RPC (Run Percentage)

RPC mengukur keserbasamaan dan distribusi run dari sebuah citra pada arah tertentu. RPC bernilai

paling besar jika panjangnya run adalah 1 untuk semua derajat keabuan pada arah tertentu. Dari Tabel 4.1 dapat dilihat bahwa tidak ada citra yang memiliki nilai RPC sama dengan satu. Semua citra memiliki nilai RPC kurang dari satu. RPC maksimum diperoleh untuk citra tekstur “Logam 1”, dan nilai RPC minimum pada citra “Ubin 3”.

Dari rumus (3) dan (7) dapat dilihat bahwa nilai RPC berbanding terbalik dengan nilai LRE. Karena nilai LRE juga berbanding terbalik dengan nilai SRE, maka dapat dikatakan nilai RPC berbanding lurus dengan nilai SRE. Hal ini dibuktikan dengan citra “Logam 1” memiliki nilai RPC dan SRE tertinggi dengan nilai LRE terendah. Sedangkan citra “Ubin 3” memiliki nilai RPC dan SRE terendah namun nilai LRE-nya tertinggi.

Tekstur halus memiliki nilai RPC yang lebih tinggi karena distribusi run untuk semua derajat keabuan lebih mengumpul pada run pendek. Sedangkan RPC bernilai maksimal atau sama dengan satu jika semua derajat keabuan memiliki run dengan panjang 1.

### 4.3 Rotasi

Jika citra tekstur dirotasi dengan kelipatan 90o, akan diperoleh hasil ciri yang sama dengan citra tanpa rotasi. Hal ini disebabkan karena proses ekstraksi ciri untuk metode run length dilakukan pada 4 sudut pencarian, yaitu sudut 0°, 90°, 45°, dan 135° untuk kemudian diambil reratanya. Sehingga walau citra dirotasi dengan kelipatan 90° sekalipun tetap menghasilkan ciri yang sama.

Tabel 2. Tabel hasil ciri untuk rotasi citra Logam 1

Rotasi	Hasil ciri Grey Level Run Length				
	SRE	LRE	GLU	RLU	RPC
0°	0,99253	1,03182	1804,96	254402	0,98983
90°	0,99253	1,03182	1804,96	254402	0,98983
180°	0,99253	1,03182	1804,96	254402	0,98983
270°	0,99253	1,03182	1804,96	254402	0,98983
360°	0,99253	1,03182	1804,96	254402	0,98983

### 4.4 Klasifikasi Ciri

Metode klasifikasi yang digunakan adalah metode LDA dan k-NN. Citra latih VisTex yang telah diekstraksi ciri seperti yang ditunjukkan pada Tabel 4.1 dipergunakan sebagai basis data pelatihan. Adapun hasil dari kedua metode klasifikasi LDA dan k-NN ditunjukkan pada Tabel 3.

Tabel 3. Tabel hasil klasifikasi LDA dan k-NN

Nama Citra	Hasil Klasifikasi				
	LDA	k-NN			
		k = 1	k = 3	k = 5	k = 7
Kulit kayu 1	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu
Kulit kayu 2	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu
Kulit kayu 3	Air	Kulit kayu	Kulit kayu	Kulit kayu	Air
Kulit kayu 4	Kulit kayu	Kulit kayu	Kulit kayu	Air	Kulit kayu
Kulit kayu 5	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu
Kulit kayu 6	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Logam
Kulit kayu 7	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Logam
Kulit kayu 8	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu
Kulit kayu 9	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu	Kulit kayu
Logam 1	Logam	Logam	Kulit kayu	Kulit kayu	Logam
Logam 2	Logam	Logam	Logam	Logam	Logam
Logam 3	Logam	Logam	Logam	Logam	Logam
Logam 4	Kulit kayu	Logam	Logam	Logam	Logam
Logam 5	Logam	Logam	Logam	Pasir	Pasir
Logam 6	Logam	Logam	Logam	Logam	Ubin
Pasir 1	Pasir	Pasir	Logam	Logam	Logam
Pasir 2	Pasir	Pasir	Pasir	Pasir	Pasir
Pasir 3	Air	Pasir	Pasir	Pasir	Pasir
Pasir 4	Pasir	Pasir	Pasir	Pasir	Kulit Kayu
Pasir 5	Pasir	Pasir	Pasir	Pasir	Pasir
Pasir 6	Kulit kayu	Pasir	Pasir	Pasir	Pasir
Pasir 7	Kulit kayu	Pasir	Pasir	Pasir	Air
Ubin 1	Air	Ubin	Air	Ubin	Ubin
Ubin 2	Ubin	Ubin	Ubin	Ubin	Ubin
Ubin 3	Ubin	Ubin	Ubin	Ubin	Ubin
Ubin 4	Kulit kayu	Ubin	Ubin	Kulit kayu	Kulit kayu
Ubin 5	Pasir	Ubin	Ubin	Pasir	Pasir
Ubin 6	Ubin	Ubin	Air	Ubin	Ubin
Ubin 7	Ubin	Ubin	Air	Ubin	Ubin
Ubin 8	Ubin	Ubin	Ubin	Ubin	Ubin
Air 1	Kulit kayu	Air	Kulit kayu	Kulit kayu	Pasir
Air 2	Air	Air	Air	Ubin	Ubin
Air 3	Air	Air	Air	Ubin	Ubin
Air 4	Air	Air	Air	Air	Air
Air 5	Air	Air	Air	Kulit kayu	Air
Air 6	Air	Air	Air	Air	Air
Air 7	Air	Air	Air	Air	Air
Air 8	Pasir	Air	Pasir	Pasir	Pasir



Tabel 4. Tabel kesalahan klasifikasi LDA dan k-NN

Kelompok Citra	Σ Error					Σ Sampel
	LDA	k-NN				
		k = 1	k = 3	k = 5	k = 7	
Kulit kayu	1	0	0	1	3	9
Logam	1	0	1	2	2	6
Pasir	3	0	1	1	3	7
Ubin	3	0	3	2	2	8
Air	2	0	2	5	4	8
Jumlah	10	0	7	11	14	38

Untuk mengetahui unjuk kerja dari suatu pelatih (*clasifier*), maka perlu dihitung rasio kesalahan dari suatu pelatih dengan rumus sebagai berikut:

$$Error\ rate = \frac{\sum error}{\sum sample} \times 100\% \quad (11)$$

Sehingga dapat diperoleh rasio kesalahan dari masing-masing metode, yaitu:

1. LDA

$$Rasio\ kesalahan = \frac{10}{38} \times 100\% = 26,316\%$$

2. k-NN

- Untuk k = 1

$$Rasio\ kesalahan = \frac{0}{38} \times 100\% = 0\%$$

- Untuk k = 3

$$Rasio\ kesalahan = \frac{7}{38} \times 100\% = 18,421\%$$

- Untuk k = 5

$$Rasio\ kesalahan = \frac{11}{38} \times 100\% = 28,947\%$$

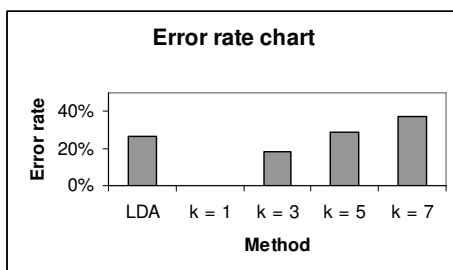
- Untuk k = 7

$$Rasio\ kesalahan = \frac{14}{38} \times 100\% = 36,842\%$$

Kemudian hasil perhitungan rasio kesalahan tersebut dibuat tabel seperti yang ditunjukkan oleh Tabel 5.

Tabel 5. Tabel rasio kesalahan

LDA	k-NN			
	k = 1	k = 3	k = 5	k = 7
26,316 %	0 %	18,421 %	28,947 %	36,842 %



Gambar 13. Grafik rasio kesalahan klasifikasi

Dari Tabel 4.3, dapat dilihat bahwa LDA mempunyai rasio kesalahan yang lebih besar dibandingkan dengan k-NN untuk nilai k=1 dan k=3. Pada klasifikasi k-NN, semakin besar nilai k yang diambil, maka semakin besar pula kesalahan klasifikasi yang terjadi. Hal ini dikarenakan semakin besar nilai k, maka daerah batas keputusan juga semakin lebar. Semakin lebar daerah batas keputusan, maka kemungkinan terjadi kesalahan semakin tinggi. Jika nilai k=1, maka yang diambil adalah sampel latihan terdekat. Karena semua sampel latihan juga digunakan sebagai citra yang akan diklasifikasi, maka pasti terdapat 1 sampel latihan yang cocok untuk masing-masing ke-38 citra *VisTex*. Hal ini akan menyebabkan tidak adanya kesalahan klasifikasi k-NN untuk citra *VisTex* dengan k=1. Saat k dinaikkan lebih dari 1, daerah pencarian sampel latihan akan lebih besar sehingga merusak lokalitas estimasi yang menyebabkan kesalahan klasifikasi yang semakin besar. Hal ini sesuai seperti yang tertulis dalam landasan teori.

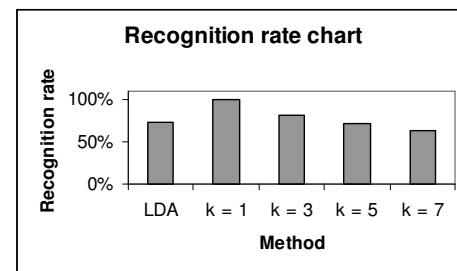
Untuk menghitung unjuk kerja pelatihan dapat juga dihitung rasio pengenalan (recognition rate) dengan rumus sebagai berikut:

$$Recognition\ rate = \frac{\sum correct}{\sum sample} \times 100\% \quad (4.2)$$

$$= 1 - Error\ rate \quad (4.3)$$

Tabel 6. Tabel rasio pengenalan

LDA	k-NN			
	k = 1	k = 3	k = 5	k = 7
73,684 %	100 %	81,579 %	71,053 %	63,158 %



Gambar 14. Grafik rasio pengenalan klasifikasi

Sehingga dapat disimpulkan bahwa pada analisis tekstur metode run length, metode klasifikasi yang paling bagus adalah metode k-NN dengan nilai k=1.

## V. PENUTUP

### 5.1 Kesimpulan

Setelah melakukan uji coba pada perangkat lunak yang telah dibuat, dapat diambil beberapa kesimpulan sebagai berikut:

1. Analisis tekstur metode *run length* ini dapat digunakan untuk membedakan tekstur halus dan tekstur kasar.
2. Citra dengan tekstur yang halus memiliki *run* yang pendek di seluruh citra sehingga menghasilkan nilai SRE yang tinggi namun nilai LRE-nya rendah.
3. Citra dengan tekstur yang kasar memiliki *run* yang lebih panjang di seluruh citra sehingga menghasilkan nilai SRE yang rendah dan nilai LRE yang tinggi.
4. Pada citra dengan tekstur yang halus, distribusi *run* untuk semua derajat keabuan cenderung mengumpul pada *run* pendek sehingga memiliki nilai RLU dan RPC yang tinggi.
5. Pada citra dengan tekstur yang kasar, distribusi *run* untuk semua derajat keabuan cenderung lebih menyebar sehingga memiliki nilai RLU dan RPC yang tinggi.
6. Rotasi citra dengan kelipatan  $90^\circ$  tidak akan mengubah nilai ciri yang diperoleh.
7. Ketika analisis tekstur metode *run length* digunakan untuk proses klasifikasi, diperoleh rasio kesalahan sebesar 26,32 % untuk metode LDA.
8. Ketika analisis tekstur metode *run length* digunakan untuk proses klasifikasi k-NN, diperoleh rasio kesalahan sebesar 0 % untuk k=1, 18,42 % untuk k=3, 28,95 % untuk k=5, dan 36,84 % untuk k=7.
9. Pada metode k-NN, semakin besar nilai k semakin besar pula kesalahan klasifikasinya.

### 5.2 Saran

Dalam pembuatan Tugas Akhir ini, masih terdapat banyak kekurangan yang dapat diperbaiki untuk pengembangan berikutnya diantaranya adalah :

1. Untuk dapat membandingkan kinerja metode *run length* ini, dapat dibuat analisis tekstur dengan metode yang berbeda, seperti metode autokorelasi, *co-occurrence*, frekuensi tepi, dan lain-lain.
2. Analisis tekstur metode *run length* ini dapat dibuat dengan metode klasifikasi yang berbeda seperti misalnya metode *k-mean*.

## DAFTAR PUSTAKA

- [1] Kitaguchi, S., dkk, Suitability of Texture Analysis Method for Perceptual Texture, Congress of the International Colour Association, 2005.
- [2] Munir, Rinaldi, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Informatika, Bandung, 2004.
- [3] Nadler, M. dan E. P. Smith, Pattern Recognition Engineering, John Wiley & Sons, New York, 1993.
- [4] Schalkoff, Robert, Pattern Recognition Statistical, Structural and Neural Approaches, John Wiley & Sons, New York, 1992.
- [5] Schalkoff, Robert, Digital Image Processing and Computer Vision, John Wiley & Sons, New York, 1989.
- [6] Shingh, Sameer, dkk, Nearest Neighbor Strategies for Image Understanding, British Crown Copyright, 1999.
- [7] Suksmono, Andrian B., Dasar Pengolahan Citra dan Pengenalan Pola, Institut Teknologi Bandung, 2006.
- [8] van Rikxoort, E. M., Texture Analysis, Graduate Research Proposal in AI, 15 April 2004.
- [9] Walker, Ross F., Adaptive Multi Scale Texture Analysis with Application to Automated Cytology, Doctor of Philosophy Thesis, University of Queensland, 1997.
- [10] VisTex Database  
[www.vismod.media.mit.edu/pub/VisTex/Images/Reference](http://www.vismod.media.mit.edu/pub/VisTex/Images/Reference)



**Mita Indriani** [L2F304257]  
Lahir di Denpasar, 27 Januari 1982  
Mahasiswa Teknik Elektro Ekstensi 2004,  
Bidang Konsentrasi Elektronika dan  
Telekomunikasi,  
Universitas Diponegoro Semarang  
Email : rheina\_2000@yahoo.com

Semarang,.....Mei 2007

Menyetujui dan Mengesahkan

Pembimbing I,

Imam Santoso, S.T., M.T.  
NIP. 132 162 546

Pembimbing II,

Yuli Christyono, S.T., M.T.  
NIP. 132 163 660