

KINERJA PENGENALAN CITRA TEKSTUR MENGGUNAKAN ANALISIS TEKSTUR METODE *RUN LENGTH*

Imam Santoso, Yuli Christyono, Mita Indriani

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro

Jln. Prof. H. Soedarto, S.H., Tembalang, Semarang, 50275

e-mail: imamstso@elektro.ft.undip.ac.id

ABSTRAKSI

Salah satu cara untuk mengenali suatu citra adalah dengan membedakan tekstur yang merupakan komponen dasar pembentuk citra. Tekstur citra dapat dibedakan dengan 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 berdasarkan ciri yang diperoleh secara matematis.

Makalah ini membahas hasil penelitian salah satu metode analisis tekstur yaitu metode *run length*. Metode ini mengenali jumlah *run* pada suatu piksel dengan level intensitas sama dan berurutan dalam satu arah tertentu. Kemudian ciri tekstur dapat diperoleh dari nilai *SRE* (*Short Run Emphasis*), *LRE* (*Long Run Emphasis*), *GRU* (*Grey Level Uniformity*), *RLU* (*Run Length Uniformity*), dan *RPC* (*Run Percentage*). Kelima ciri tersebut kemudian digunakan untuk klasifikasi dengan menggunakan metode *LDA* (*Linear Discriminant Analysis*) dan *k-NN* (*k-Nearest Neighbor*). Sedangkan citra tekstur yang dianalisis berasal dari *VisTex Database*.

Dari hasil penelitian diketahui bahwa metode *run length* ini dapat 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*.

1. PENDAHULUAN

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 indera 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.[1]

Dalam penelitian ini, dibahas hasil ekstraksi ciri menggunakan analisis tekstur metode *gray level run length*. Kemudian dilanjutkan dengan klasifikasi menggunakan metode klasifikasi linier *LDA* (*Linear Discriminant Analysis*) dan *k-NN* (*k-Nearest Neighbor*).

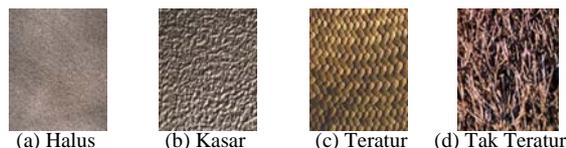
Adapun data citra tekstur yang digunakan adalah dari *Vision Texture (VisTex) Database* menggunakan 5 kelas dari 19 kelas yang ada dan jumlah data sebanyak 38 citra.[7]

2. 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.

Syarat-syarat terbentuknya tekstur setidaknya ada dua, yaitu [3]:

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.



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.[1]

2.1 Metode Run Length [2,3]

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.

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

0	1	2	3
0	2	3	3
2	1	1	1
3	0	3	0

Citra aras keabuan $f(x,y)$

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

Run Length $B(a,r)$

Gambar 2. Contoh citra dan matriks *run length* pada arah 0° [1]

2.2 Pengenalan Pola [4,6]

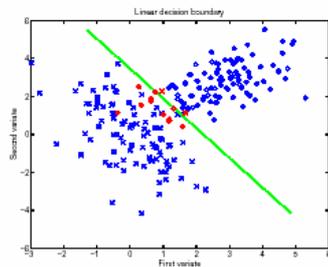
Pengenalan pola merupakan ilmu mengenai diskripsi atau klasifikasi dari hasil ekstraksi ciri yang memetakan suatu fitur, yang merupakan ciri utama suatu objek (yang dinyatakan dalam sekumpulan bilangan-bilangan) ke suatu kelas yang sesuai. Salah satu metode pengenalan pola, yaitu metode statistik.

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 pengarahan) 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 pengarahan) 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.

2.3 Linear Discriminant Analysis (LDA)^[4]

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 3. Contoh batas keputusan LDA pada ruang ciri bivarian

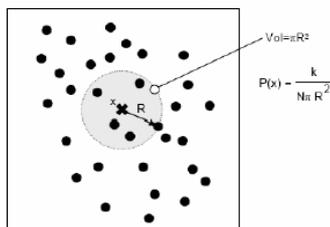
2.4 k-Nearest Neighbor (k-NN)^[4,5]

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.

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

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*).

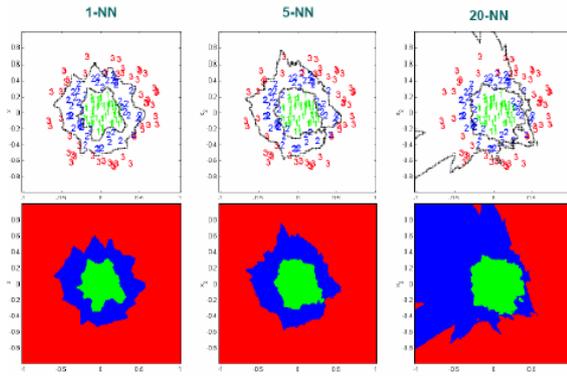


Gambar 4. Ilustrasi konsep tetangga terdekat pada ruang 2 dimensi [4]

Keuntungan utama dari metoda k-NN adalah metoda ini memberikan pendekatan yang sangat sederhana untuk pemilah optimal Bayes. Apabila terdapat suatu data masukan yang jumlahnya N buah dimana N_i buah dari kelas ω_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 ω_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, pemilah Bayes menjadi

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

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 .

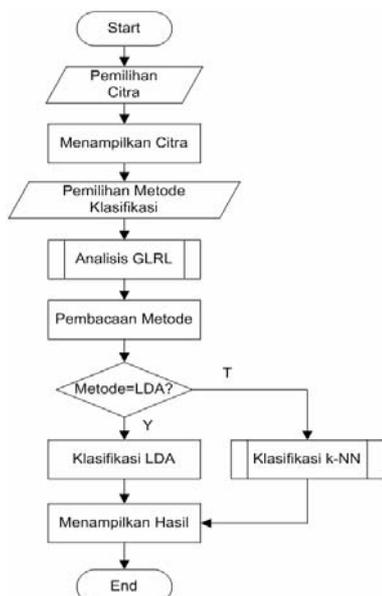


Gambar 5. Ilustrasi pengaruh pemilihan k [4]

Pemilahan k-NN dilakukan dengan mencari k -buah tetangga terdekat dan memilih kelas dengan k_i terbanyak pada kelas ω_i .

3. PERANCANGAN DAN IMPLEMENTASI

Proses 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.



Gambar 6. Diagram alir program analisis tekstur

Pada tahap pertama mula-mula dilakukan pemilihan citra. Citra latihan yang digunakan dalam program ini diambil dari basis data VisTex dengan ukuran 512x512 *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°, 90°, 45°, dan 135°. 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.

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.

4. PENGUJIAN DAN ANALISIS

Hasil analisis tekstur pada citra latihan VisTex [7] yang terbagi dalam 5 kelompok dengan jumlah total 38 citra ditunjukkan pada Tabel 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 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".

Tabel 1. Tabel ciri citra latihan VisTex

Nama Citra	Hasil ciri Grey Level Run Length				
	SRE	LRE	GLU	RLU	RPC
Kulit kayu 1	0,9641	1,181	1176,95	226649	0,9489
Kulit kayu 2	0,9763	1,1114	1726,87	238171	0,9669
Kulit kayu 3	0,9545	1,2112	2867,46	218328	0,9384
Kulit kayu 4	0,961082	1,2137	1798,06	223459	0,9425
Kulit kayu 5	0,985438	1,0610	1923,84	247293	0,9805
Kulit kayu 6	0,981150	1,0855	1590,62	242948	0,9739
Kulit kayu 7	0,981081	1,0845	1481,30	242917	0,9740
Kulit kayu 8	0,987504	1,0546	1328,19	249284	0,9828
Kulit kayu 9	0,986405	1,0610	1366,78	248164	0,9812
Logam 1	0,992537	1,0318	1804,96	254402	0,9898
Logam 2	0,983382	1,0847	2579,87	244957	0,9757
Logam 3	0,981590	1,0982	2398,41	243029	0,9727
Logam 4	0,981664	1,0885	2456,45	243306	0,9739
Logam 5	0,967631	1,2368	2277,12	228229	0,9460
Logam 6	0,958151	1,3112	2738,66	219139	0,9307
Pasir 1	0,980979	1,0794	3309,42	242954	0,9747
Pasir 2	0,970992	1,1259	3203,19	233432	0,9612
Pasir 3	0,967078	1,1436	3697,58	229793	0,9560
Pasir 4	0,976619	1,0988	3112,04	238753	0,9689
Pasir 5	0,970945	1,1256	3139,04	233372	0,9612
Pasir 6	0,964112	1,1778	2033,46	226584	0,9492
Pasir 7	0,953653	1,2403	1825,25	217007	0,9338
Ubin 1	0,935731	1,3247	3291,52	202042	0,9115
Ubin 2	0,867139	1,8813	3874,09	151367	0,8130
Ubin 3	0,856454	2,0020	3788,69	144138	0,7969
Ubin 4	0,958304	1,2142	1190,82	221223	0,9407
Ubin 5	0,965823	1,1708	3267,46	228115	0,9515
Ubin 6	0,918291	1,4255	11072,80	188269	0,8887
Ubin 7	0,952365	1,2755	4772,96	215357	0,9297
Ubin 8	0,949531	1,2694	5808,63	213290	0,9280
Air 1	0,979106	1,0991	1312,85	241101	0,9707
Air 2	0,928034	1,3543	4420,35	196358	0,9034
Air 3	0,924058	1,3966	5786,68	193745	0,8974
Air 4	0,959444	1,1795	4515,14	223014	0,9461
Air 5	0,962000	1,1783	1578,46	225084	0,9479
Air 6	0,950624	1,2226	6286,64	215405	0,9368
Air 7	0,952271	1,2148	5218,17	216975	0,9369
Air 8	0,970865	1,1449	1857,15	232977	0,9593

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 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.

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 1 dapat dilihat bahwa nilai GLU maksimum diperoleh untuk citra tekstur “Ubin 6”, dan nilai LRE minimum pada citra “Kulit kayu 1”.

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 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.

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.

Pengaruh Rotasi

Jika citra tekstur dirotasi dengan kelipatan 90°, 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°	0,9925	1,0318	1804,96	254402	0,989
90°	0,9925	1,0318	1804,96	254402	0,989
180°	0,9925	1,0318	1804,96	254402	0,989
270°	0,9925	1,0318	1804,96	254402	0,989
360°	0,9925	1,0318	1804,96	254402	0,989

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 C Citra	ΣError					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

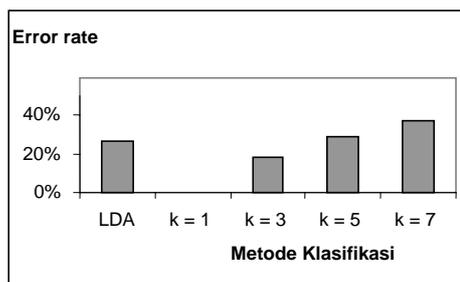
Untuk mengetahui unjuk kerja suatu pelatih (*classifier*), maka perlu dihitung rasio kesalahan dari suatu pelatih dengan rumus sebagai berikut:

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

Sehingga dapat diperoleh rasio kesalahan dari masing-masing metode, yaitu seperti yang ditunjukkan pada Tabel 5.

Tabel 5. Tabel rasio kesalahan

LDA	k-NN			
	k = 1	k = 3	k = 5	k = 7
26,32 %	0 %	18,421 %	28,95%	36,84 %



Gambar 7. Grafik rasio kesalahan klasifikasi

Dari Tabel 4 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 latih terdekat. Karena semua sampel latih juga digunakan sebagai citra yang akan diklasifikasi, maka pasti terdapat 1 sampel latih 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 latih akan lebih besar sehingga merusak lokalitas estimasi yang menyebabkan kesalahan klasifikasi yang semakin besar. Hal ini sesuai seperti yang tertulis dalam landasan teori.

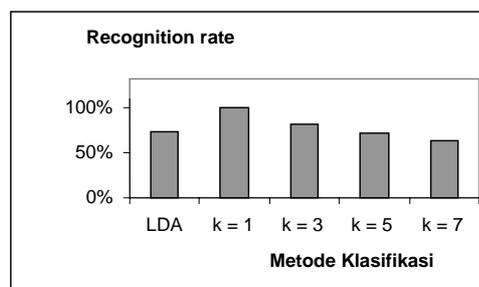
Untuk menghitung unjuk kerja pelatihan dapat juga dihitung rasio pengenalan (*recognition rate*) dengan rumus sebagai berikut:

$$Recognition\ rate = \frac{\sum\ correct}{\sum\ sample} \times 100\ \%$$

$$= 1 - Error\ rate$$

Tabel 6. Tabel rasio pengenalan

LDA	k-NN			
	k = 1	k = 3	k = 5	k = 7
73,68 %	100 %	81,58%	71,05%	63,16%



Gambar 8. 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.

5. PENUTUP

Setelah dilakukan pengujian dan analisis dapat diketahui:

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° 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.

Penelitian ini dapat dikembangkan untuk berbagai aplikasi misalnya untuk identifikasi atau klasifikasi objek tertentu misalnya biji-bijian, batubatuan, tanah, bahan makanan dan lain-lain. Bisa juga untuk penentuan kualitas suatu objek misalnya kain atau adonan.

PUSTAKA

- [1] Kitaguchi, S., dkk, Suitability of Texture Analysis Method for Perceptual Texture, *Congress of the International Colour Assosiation*, 2005.
- [2] Dong-Hui Xu et. al. *Run Length Encoding for Volumetric Texture, Intelligent Multimedia Proc. Lab. School of Computer Science, Telecomm. & Information Sys., De Paul Univ., Chicago, U.S.A., 2004.*
- [3] van Rikxoort, E. M., Texture Analysis, *Graduate Research Proposal in AI*, 15 April 2004.
- [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] VisTex Database
(www.vismod.media.mit.edu/pub/VisTex/Images/Reference)