

Klasifikasi Citra Makanan Menggunakan Algoritme *Learning Vector Quantization* Berdasarkan Ekstraksi Fitur *Color Histogram* dan *Gray Level Co-occurrence Matrix*

Sarah Yuli Evangelista Simarmata¹, Yuita Arum Sari², Sigit Adinugroho³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: ¹sarahyulies@gmail.com, ²yuita@ub.ac.id, ³sigit.adinu@ub.ac.id

Abstrak

Banyak foto makanan yang tersebar di media sosial namun terkadang kita lupa bahkan tidak mengetahui nama makanan tersebut. Kemampuan manusia untuk mengenali dan mengidentifikasi juga subjektif terhadap pengaruh eksternal, seperti kelelahan, prasangka dan sebagainya. Komputer dapat membantu dengan membuat sebuah sistem yang dapat mengenali dan mengidentifikasi makanan melalui gambar. Penelitian telah dilakukan bahwa proses identifikasi dan klasifikasi secara otomatis menggunakan komputer dapat menghemat waktu dibandingkan harus mengidentifikasi secara manual. Citra pada makanan memiliki warna dan tekstur yang berbeda-beda. Metode ekstraksi fitur warna yang digunakan pada penelitian ini adalah *Color Histogram* dan ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GLCM). Algoritme klasifikasi yang digunakan adalah algoritme *Learning Vector Quantization* (LVQ), dengan parameter terbaik yang dapat digunakan yaitu nilai *learning rate* (α) sebesar 0,1, penurunan *learning rate* sebesar 0,1, maksimum *epoch* sebesar 2, nilai minimum *learning rate* sebesar 0,01 menghasilkan akurasi sebesar 53,33%. Pengujian yang dilakukan menggunakan ekstraksi fitur warna dan tekstur menghasilkan akurasi sebesar 53,33%. Penggunaan metode ekstraksi fitur warna saja menghasilkan akurasi tertinggi yaitu sebesar 67,00%, dan penggunaan metode ekstraksi fitur tekstur saja menghasilkan akurasi sebesar 53,33%. Berdasarkan hasil tersebut dapat disimpulkan bahwa algoritme LVQ berdasarkan ekstraksi fitur *Color Histogram* dan GLCM dapat digunakan untuk klasifikasi citra makanan namun tidak menghasilkan akurasi yang sempurna.

Kata kunci: makanan, warna, tekstur, color histogram, GLCM, LVQ

Abstract

Many photos of food we see on social media, but we forget and don't even know the name of the food. Humans ability to recognize and identify is also subjective to external such as fatigue, prejudice and etc. Computers can help by build a system that can recognize and identify food through images. Researches have been conducted that the process of automatically identifying and classifying using computer can save more time compared to identify manually. Food image has different colors and textures. The color feature extraction method used in this research is *Color Histogram* and for the texture feature extraction is *Gray level co-occurrence matrix* (GLCM). The classification algorithm used is *Learning Vector Quantization* (LVQ) with the best parameters that can be used are *learning rate* (α) 0.1, decreament *learning rate* 0.1, maximum *epoch* 2, minimum *learning rate* 0.01 and gives accuracy that is equal to 53,33%. The test gives 53.33% accuracy for using color and texture extraction. The use of color feature extraction only gives the highest accuracy that is equal to 67.00%, and the use of texture feature extraction only gives accuracy that is equal to 53.33%. From the results, concluded that LVQ algorithm based on *Color Histogram* feature extraction and GLCM can be used to classify food image but can not give a perfect accuracy.

Keywords: foods, color, texture, color histogram, GLCM, LVQ

1. PENDAHULUAN

Citra digital adalah salah satu pengetahuan

yang populer untuk dikembangkan di zaman yang serba digital ini. Salah satu contohnya adalah mengambil gambar makanan

menggunakan ponsel pintar kemudian di unggah di media sosial. Banyak foto makanan yang tersebar di media sosial namun terkadang kita lupa bahkan tidak mengetahui nama makanan tersebut. Kemampuan manusia untuk mengenali dan mengidentifikasi juga subjektif terhadap pengaruh eksternal, seperti kelelahan, prasangka dan sebagainya (Patil, Malemath dan Yadahalli, 2011). Teknologi dapat membantu manusia untuk memantau makanan yang dikonsumsi agar dapat mengontrol kesehatan mereka. Komputer dapat membantu dengan membuat sebuah sistem yang dapat mengenali dan mengidentifikasi makanan melalui gambar (Ciocca, Napoletano, dan Schettini, 2018). Penelitian telah dilakukan bahwa proses identifikasi dan klasifikasi secara otomatis menggunakan komputer dapat menghemat waktu dibandingkan harus mengidentifikasi secara manual (Karegowda, Samreen dan Pushpalatha, 2015).

Citra pada makanan memiliki warna yang berbeda-beda. Sejauh ini, karakteristik warna merupakan salah satu informasi penting yang dapat diekstrak dari sebuah citra untuk dijadikan pembandingan dengan citra lainnya (Hussain, Rao dan Praveen, 2013). Salah satu metode untuk ekstraksi fitur warna adalah menggunakan *color histogram*, yaitu menghitung jumlah kemunculan setiap warna pada sebuah citra sehingga membentuk suatu histogram. Ekstraksi fitur *color histogram* merupakan metode yang paling umum dan mudah digunakan untuk mengekstraksi informasi fitur warna dari sebuah gambar (Mathai dan Soni, 2015).

Citra-citra yang memiliki warna berbeda dapat diidentifikasi dengan mudah, namun citra yang memiliki warna sama membutuhkan fitur tekstur untuk diidentifikasi (Karegowda, Samreen dan Pushpalatha, 2015). Salah satu metode untuk ekstraksi fitur tekstur adalah GLCM, yaitu melihat jumlah intensitas *pixel* yang berpasangan dan dibentuk menjadi *matrix co-occurrence* (K, Santoso dan Isnanto, 2011). GLCM memiliki kelebihan yaitu terbukti merupakan metode ekstraksi fitur tekstur yang sangat baik untuk digunakan pada analisis citra (Schwartz, Siqueira, dan Pedrini, 2013).

Klasifikasi adalah suatu teknik pengelompokan untuk menyatakan suatu objek ke dalam salah satu kategori yang sudah didefinisikan sebelumnya. Identifikasi dan klasifikasi pada makanan dengan membandingkan SVM dengan ANN juga telah dilakukan dan menghasilkan pernyataan bahwa ANN lebih akurat daripada SVM (Anami dan

Burkpalli, 2009).

Hal ini dapat dimanfaatkan untuk melakukan klasifikasi citra makanan menggunakan ANN atau jaringan saraf tiruan. Ada beberapa metode klasifikasi ANN seperti *backpropagation* dan LVQ. LVQ memiliki kelebihan yaitu merupakan algoritme yang lebih unggul dari segi akurasi yang dihasilkan dan waktu pelatihan jika dibandingkan dengan *backpropagation* (Azizi, 2013). Pada penelitian yang dilakukan oleh Sela dan Hartati tahun 2010 yaitu pengenalan jenis penyakit THT menggunakan LVQ juga menghasilkan akurasi sebesar 94%. Berdasarkan penelitian tersebut, LVQ merupakan klasifikasi berbasis jaringan saraf tiruan yang lebih unggul dibandingkan metode klasifikasi lainnya.

2. METODE PENELITIAN

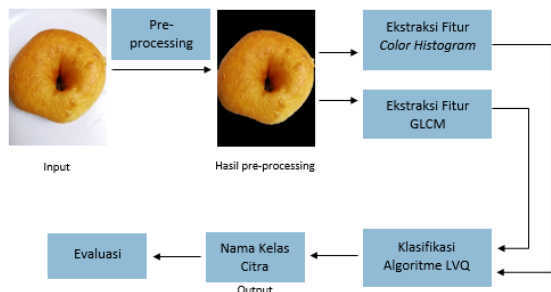
2.1. Dataset

Data yang digunakan pada penelitian ini adalah data primer yang berupa pengambilan citra makanan menggunakan Data yang digunakan untuk penelitian ini adalah data primer yang diperoleh dari pengambilan foto citra makanan yang difoto sendiri oleh penulis menggunakan kamera *smartphone* Huawei Nova 3i. Lokasi pengambilan data yaitu hari Selasa, 28 Agustus 2018 di Gedung F Fakultas Ilmu Komputer Universitas Brawijaya lantai 9 pada jam 11:00 – 12:00 siang. Data yang digunakan terdiri dari 15 kelas makanan, dengan sebanyak 17 data untuk masing-masing kelas sehingga total data yang digunakan adalah 255 data. Citra makanan yang diambil terdiri dari makanan yang utuh, $\frac{1}{2}$ bagian, dan pengambilan citra tegak lurus dan miring. Pembagian data untuk data latih dan data uji yaitu sebanyak 17 data dari masing-masing kelas digunakan sebagai data latih dan 1 data dari masing-masing kelas digunakan sebagai data uji.

2.2. Strategi Penelitian

Algoritme klasifikasi diawali dengan memasukkan citra makanan untuk dilakukan *pre-processing*. Citra hasil *pre-processing* diekstrak fitur warnanya menggunakan *color histogram*, kemudian diekstrak fitur teksturnya menggunakan GLCM. Langkah selanjutnya adalah klasifikasi menggunakan algoritme LVQ berdasarkan fitur-fitur yang telah didapatkan dari ekstraksi fitur warna dan tekstur. Kemudian hasil klasifikasi dievaluasi akurasinya untuk

diperoleh hasil akurasi penggunaan algoritme LVQ dalam mengklasifikasikan citra makanan. Rancangan algoritme pada penelitian ini ditunjukkan pada Gambar 11.



Gambar 1. Rancangan algoritme

2.3. Jaringan Saraf Tiruan

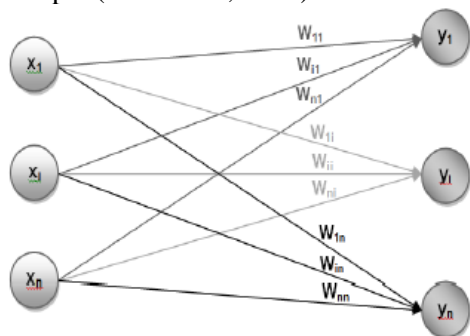
Konsep jaringan saraf tiruan adalah direpresentasikan seperti konsep biologis pada otak manusia untuk dilakukan proses pembelajaran. Jaringan saraf tiruan disusun oleh sejumlah simpul (*node*) yang adalah elemen yang memprosesnya. Sel saraf biologis (neuron) dimodelkan menjadi simpul (*node*).

2.4. LVQ

LVQ merupakan suatu metode pada lapisan kompetitif untuk melakukan proses pembelajaran pada data. LVQ digunakan untuk klasifikasi pola setiap unit sehingga setiap output merupakan representasi dari kelas atau kategori tertentu. LVQ mengklasifikasikan vektor *input* dalam kelas atau kategori yang sama dengan *output* yang memiliki vektor paling dekat dengan vektor input (Sela dan Hartati, 2010).

Arsitektur metode LVQ diilustrasikan pada Gambar 2.

Model pembelajaran LVQ dilatih secara signifikan, supaya apabila dibandingkan dengan algoritma lainnya, menghasilkan proses yang lebih cepat (Martinuva, 2015).



Gambar 2. Arsitektur jaringan LVQ (Sumber: Arifianto, Sarosa dan Setyawati, 2014)

Menurut Arifianto, Sarosa dan Setyawati, 2014 terdapat algoritme pelatihan LVQ 1, yaitu:

1. Inisialisasi bobot awal dan *learning rate* (α).
2. Sewaktu kondisi *stop* == *false* maka kerjakan langkah 3-11.
3. Untuk masing-masing *training input* (vektor x), kerjakan langkah 4-9.
4. Hitung J menggunakan rumus pada Persamaan (1), kemudian J yang terkecil merupakan neuron pemenang (C_j).

$$J = \|x - w_j\| \tag{1}$$

5. Kemudian perbarui bobot w_j dengan syarat harus memenuhi:
6. $T = C_j$ maka
7. Update bobot dengan rumus pada Persamaan (2).

$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha[x - w_j(\text{lama})] \tag{2}$$

8. Jika $T \neq C_j$ maka
9. Update bobot dengan rumus pada Persamaan (3).

$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha[x - w_j(\text{lama})] \tag{3}$$

10. Kemudian kurangi nilai α menggunakan rumus pada Persamaan (4).

$$\alpha(\text{baru}) = \alpha(\text{lama}) \times \text{deca} \tag{4}$$

11. Hentikan pada iterasi tertentu atau nilai α mendekati 0.

Keterangan:

- x = vektor *training*
- T = kelas dari vektor *training*
- w_j = bobot vektor untuk luaran ke- j
- C_j = *output* kelas yang dihasilkan
- $\|x - w_j\|$ = *Euclidean Distance* antara vektor masukan dan bobot vektor ke- j .
- α = nilai *learning rate*

2.5. Color Histogram

Color histogram adalah sebuah metode untuk mendeskripsikan konten warna dengan menghitung jumlah kemunculan setiap warna pada sebuah gambar (Meskaldji, Boucherkha dan Chikhi, 2009). Untuk mempercepat proses pembentukan histogram warna diperlukan proses kuantisasi warna, karena untuk menghitung jumlah kemunculan sejumlah variasi warna RGB membutuhkan waktu yang relatif lama. Kuantisasi warna yaitu membagi setiap komponen warna dalam rentang jarak tertentu. Kuantisasi dapat dilakukan karena terdapat sejumlah variasi warna yang sangat mirip sehingga mata manusia dapat menganggap warna-warna tersebut adalah warna yang sama atau serupa (Madenda, 2015).

Histogram dapat terdiri dari 48 bin warna, dimana setiap bin mendefinisikan sebuah rentang dari nilai *pixel*. Rentang ini merepresentasikan intensitas level yang berbeda dari setiap komponen RGB. Kemudian nilai dari setiap bin dinormalisasikan dengan membagi nilai tersebut dengan jumlah seluruh *pixel* pada gambar (Hussain, Rao dan Praveen, 2013). Penelitian ini menggunakan 8 bin warna dengan rentang 0-31, 32-63, dst. bertujuan untuk meminimalisir bin warna yang digunakan untuk dijadikan fitur pada klasifikasi citra.

2.6. GLCM

Salah satu metode statistik yang dapat dipakai untuk menganalisis tekstur adalah GLCM, yang mana *pixel* berskala keabuan yang berpasangan (bertetangga) dan memiliki intensitas tertentu pada suatu citra dapat dibentuk menjadi *matrix co-occurrence* (K, Santoso dan Isnanto, 2011). Menggunakan pendekatan statistik seperti GLCM dapat membantu untuk mendapatkan informasi yang bernilai mengenai posisi relatif suatu ketetanggaan *pixel* pada sebuah citra. Ketetanggaan *pixel* dapat dipilih berdasarkan arah sesuai sudut-sudut pada GLCM, yaitu sudut 0°, 45°, 90°, dan 135° (Eleyan dan Demirel, 2011).

Tahap untuk ekstraksi fitur GLCM yaitu:

1. Menghitung ketetanggaan *pixel* pada sudut 0°, 45°, 90°, dan 135°.
2. Mengubah matriks *framework* menjadi matriks simetris yaitu menjumlahkan dengan matriks *transpose* dari matriks tersebut.
3. Normalisasi matriks untuk menghilangkan ketergantungan terhadap ukuran citra dengan cara membagi kemunculan *pixel* pada komposisi *pixel* tertentu dengan jumlah kemunculan seluruh *pixel* pada suatu gambar seperti pada Persamaan (5).

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \tag{5}$$

Keterangan:

$P_{i,j}$ = nilai dari normalisasi (probabilitas)

$V_{i,j}$ = nilai kemunculan suatu komposisi *pixel* tertentu

N = jumlah maksimal *pixel* citra

4. Menghitung fitur-fitur tekstur.
Terdapat 9 fitur tekstur yang dapat diperoleh dari citra pada *co-occurrence matrix*, yaitu (Hall-Beyer, 2017):

1. Mean

Merupakan rata-rata dari probabilitas yang dapat dihitung berdasarkan i dan j ditunjukkan pada Persamaan (6) dan Persamaan (7).

$$\mu_i = \sum_{i,j=0}^{N-1} i(P_{i,j}) \tag{6}$$

$$\mu_j = \sum_{i,j=0}^{N-1} j(P_{i,j}) \tag{7}$$

Keterangan:

μ = rata-rata

2. Varians dan Standar Deviasi

Varians menentukan simpangan atau sebaran dari nilai rata-rata *pixel* tekstur yang dapat dihitung berdasarkan i dan j ditunjukkan pada Persamaan (8) dan Persamaan (9).

$$\sigma_i^2 = \sum_{i,j=0}^{N-1} P_{i,j} (i - \mu_i)^2 \tag{8}$$

$$\sigma_j^2 = \sum_{i,j=0}^{N-1} P_{i,j} (j - \mu_j)^2 \tag{9}$$

Keterangan:

σ^2 = varians

Standar deviasi dihitung berdasarkan i dan j dengan menggunakan rumus pada Persamaan (10) dan Persamaan (11).

$$\sigma_i = \sqrt{\sigma_i^2} \tag{10}$$

$$\sigma_j = \sqrt{\sigma_j^2} \tag{11}$$

Keterangan:

σ = standar deviasi

3. Kontras (*Contrast*)

Kontras merupakan variasi dari nilai intensitas lokal dalam matriks tekstur. Semakin rendah kontras tekstur, berarti *pixel* tetangga pada matriks memiliki nilai intensitas lokal yang berdekatan/mirip dan sebaliknya (Madenda, 2015). Nilai kontras dapat dihitung dengan menggunakan rumus pada Persamaan (12).

$$Contrast = \sum_{i,j=0}^{N-1} P_{i,j} (i - j)^2 \tag{12}$$

4. *Dissimilarity*

Dissimilarity atau ketidakmiripan merupakan ukuran yang mendefinisikan variasi dari tingkat intensitas *pixel* tetangga pada suatu citra. Nilai *dissimilarity* dapat dihitung dengan menggunakan rumus pada Persamaan (13).

$$Dissimilarity = \sum_{i,j=0}^{N-1} P_{i,j} |i - j| \tag{13}$$

5. *Homogeneity*

Homogeneity merupakan tingkat homogenitas perulangan dari tekstur. Nilai *homogeneity* dapat dihitung dengan menggunakan rumus pada Persamaan (14).

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \quad (14)$$

6. *Angular Second Moment (ASM)*

ASM merupakan ukuran homogenitas citra dan dapat dihitung menggunakan rumus pada Persamaan (15).

$$ASM = \sum_{i,j=0}^{N-1} P_{i,j}^2 \quad (15)$$

7. Energi

Energi merupakan ukuran dari tingkat keseragaman tekstur. Semakin tinggi nilai dari energi, maka semakin tinggi tingkat keseragaman tekstur suatu citra. Nilai energi dapat dihitung dengan menggunakan rumus pada Persamaan (16).

$$Energy = \sqrt{ASM} \quad (16)$$

8. Entropi

Entropi merupakan ukuran dari tingkat keacakan permukaan tekstur suatu citra. Nilai entropi memberi informasi mengenai permukaan tekstur yang halus maupun kasar. Apabila nilai entropi semakin mendekati angka 1, maka tekstur memiliki tingkat kekasaran yang tinggi, dan sebaliknya. Nilai entropi dapat dihitung dengan menggunakan rumus pada Persamaan (17).

$$Entropy = \sum_{i,j=0}^{N-1} P_{i,j} (-\ln P_{i,j}) \quad (17)$$

Keterangan:

$\ln = \log$ natural, $\ln(0)$ diasumsikan = 0

9. Korelasi (*Correlation*)

Korelasi merupakan ukuran tingkat hubungan linear keabuan satu *pixel* relatif terhadap *pixel* yang lain. Nilai korelasi dapat dihitung dengan menggunakan rumus pada Persamaan (18).

$$Correlation = \sum_{i,j=0}^{N-1} P_{i,j} \left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (18)$$

2.7. Evaluasi

Pengujian algoritme menggunakan akurasi adalah untuk mengetahui jumlah hasil klasifikasi citra makanan yang dihasilkan oleh program sesuai dengan kelas makanan asli dibagi dengan

jumlah seluruh kelas. Akurasi ditunjukkan dalam bentuk persen. Perhitungan ini dapat memberi informasi seberapa besar akurasi yang dihasilkan pada penggunaan algoritme LVQ dalam mengklasifikasikan citra makanan berdasarkan ekstraksi fitur yang telah ditentukan. Rumus untuk menghitung akurasi ditunjukkan pada Persamaan (19).

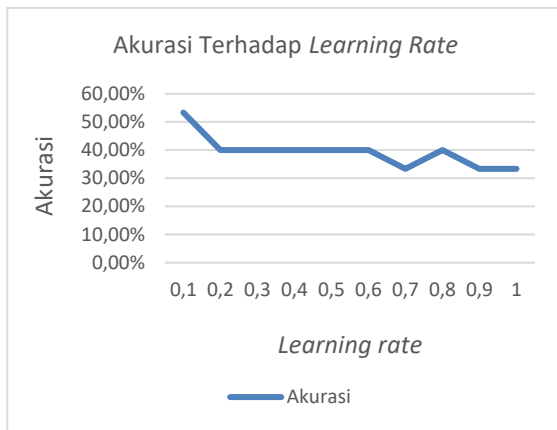
$$Akurasi = \frac{\text{jumlah kelas yang sesuai}}{\text{jumlah seluruh kelas}} \times 100\% \quad (19)$$

3. HASIL DAN PEMBAHASAN

3.1. Pengujian untuk Mengetahui Pengaruh *Learning rate* (α)

Pengujian pertama untuk algoritme LVQ yaitu untuk mengetahui pengaruh nilai *learning rate* terhadap hasil akurasi. Pengujian ini menggunakan distribusi data latih yang sama rata dengan total 240 dan data uji sebanyak 15 yang mana mewakili masing-masing kelas. Nilai *learning rate* yang digunakan adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 1. Pengujian ini menggunakan jumlah maksimum *epoch* 10, nilai penurunan *learning rate* 0,1, dan nilai minimum *learning rate* 1×10^{-4} . Pengujian terhadap *learning rate* yang digunakan menghasilkan akurasi tertinggi yaitu pada *learning rate* sebesar 0,1. Grafik akurasi yang dihasilkan dari pengujian tersebut ditunjukkan pada Gambar 3.

Berdasarkan grafik, didapatkan kesimpulan bahwa semakin besar nilai *learning rate*, maka semakin kecil akurasi yang dihasilkan. Oleh karena itu, penentuan parameter nilai *learning rate* pada data yang digunakan sangat memengaruhi hasil akurasi. Akurasi tertinggi didapatkan oleh *learning rate* 0,1 yaitu 53,33%, dan akurasi terendah didapatkan oleh *learning rate* 0,9 yaitu 33,33%. *Learning rate* yang tinggi menyebabkan perubahan bobot menjadi besar sehingga bobot yang terbentuk menjadi fluktuatif. *Learning rate* bertujuan untuk mempercepat proses iterasi, oleh karena itu semakin besar nilai *learning rate* maka algoritme akan menjadi tidak stabil sehingga akan menghasilkan penurunan hasil akurasi.



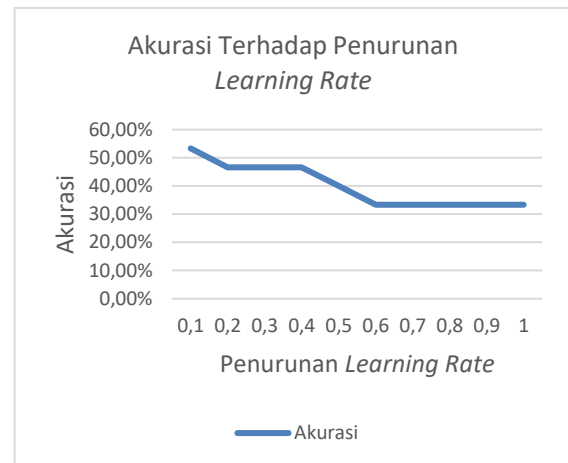
Gambar 3. Grafik akurasi pengaruh *learning rate*

3.2. Pengujian untuk Mengetahui Pengaruh Jumlah Maksimum Epoch

Jumlah maksimum *epoch* yang digunakan adalah 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 50, 500, dan 1000. Pengujian ini menggunakan jumlah nilai *learning rate* 0,1, nilai penurunan *learning rate* 0,1, dan nilai minimum *learning rate* 1×10^{-4} . Pengujian terhadap jumlah maksimum *epoch* menghasilkan akurasi yang sama yaitu 53,33% dengan jumlah maksimum *epoch* dari 2 hingga 1000. Hal ini berarti algoritme dengan *epoch* terkecil yaitu 2, akurasi yang dihasilkan sudah maksimum dan tidak berubah hingga percobaan *epoch* ke 1000. Maksimum *epoch* pada penelitian ini tidak memberi pengaruh yang signifikan terhadap akurasi yang dihasilkan karena bertambahnya *epoch* akan menurunkan nilai *learning rate* yang pada akhirnya menyebabkan perubahan bobot yang tidak signifikan.

3.3. Pengujian untuk Mengetahui Pengaruh Nilai Penurunan Learning rate

Nilai penurunan *learning rate* yang digunakan adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 1. Pengujian ini menggunakan jumlah nilai *learning rate* 0,1, jumlah maksimum *epoch* 10, dan nilai minimum *learning rate* 1×10^{-4} . Pengujian terhadap nilai penurunan *learning rate* menghasilkan akurasi tertinggi yaitu sebesar 53,33%. Grafik akurasi yang dihasilkan dari pengujian tersebut ditunjukkan pada Gambar 4. Berdasarkan grafik, semakin besar nilai penurunan *learning rate* maka semakin kecil nilai akurasi yang dihasilkan. Nilai penurunan *learning rate* digunakan sebagai pengurang *learning rate* untuk memperkecil nilai *learning rate* dalam memperoleh bobot yang ideal.



Gambar 4. Grafik akurasi pengaruh nilai penurunan *learning rate*

Sehingga semakin besar nilai penurunan *learning rate* maka semakin besar nilai *learning rate* yang dihasilkan demikian pula semakin besarnya perubahan bobot yang dihasilkan. Hasil yang diperoleh dari pengujian ini adalah penurunan *learning rate* yang ideal untuk data latih yang digunakan adalah sebesar 0,1.

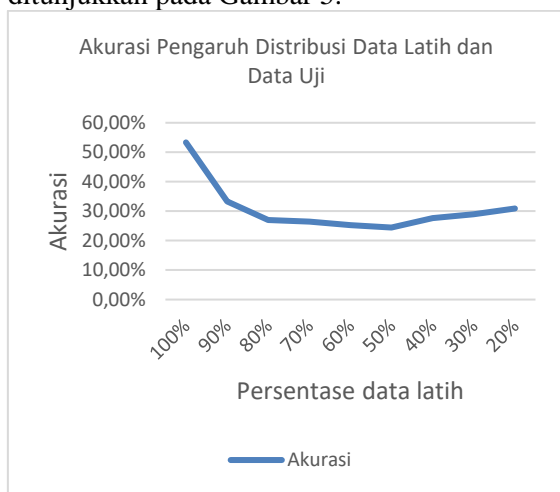
3.4. Pengujian untuk Mengetahui Pengaruh Nilai Minimum Learning rate

Nilai minimum *learning rate* yang digunakan adalah 0,01, 0,001, 0,0001, 0,00001, 0,000001, 0,0000001, 0,00000001, 0,000000001, 0,0000000001, 0,00000000001. Pengujian ini menggunakan jumlah nilai *learning rate* 0,1, jumlah maksimum *epoch* 10, dan nilai penurunan *learning rate* 0,1. Pengujian terhadap nilai penurunan *learning rate* menghasilkan akurasi yang sama besar yaitu 53,33% dan merupakan akurasi terbaik. Hal ini membuktikan bahwa nilai *learning rate* yang semakin kecil tidak memengaruhi akurasi yang dihasilkan karena tidak memberi perubahan bobot yang signifikan. Karena hasil dari pengujian pada data latih yang digunakan tidak memberikan pengaruh, yang artinya minimum *learning rate* ideal yang dapat digunakan adalah 0,01.

3.5. Pengujian untuk Mengetahui Pengaruh Distribusi Jumlah Data Latih dan Data Uji

Pengujian ini menggunakan data latih sebanyak 240, 216, 192, 168, 144, 120, 96, 72, dan 48. Pengujian ini menggunakan jumlah nilai *learning rate* 0,1, jumlah maksimum *epoch* 10, nilai minimum *learning rate* yang digunakan 1×10^{-4} , dan nilai penurunan *learning rate* 0,1.

Pengujian terhadap distribusi jumlah data latih dan data uji menghasilkan akurasi tertinggi sebesar 53,33%. Grafik akurasi yang dihasilkan ditunjukkan pada Gambar 5.



Gambar 5. Grafik akurasi pengaruh distribusi data latih dan data uji

Berdasarkan grafik, semakin sedikit data latih yang digunakan dan semakin banyak data uji yang digunakan akurasi yang dihasilkan akan semakin menurun. Pada perbandingan data latih yang semakin sedikit dan data uji semakin banyak, yaitu 90%, 80% dst. data latih akurasi menurun secara signifikan. Hal ini dikarenakan data latih yang terlalu sedikit akan menyebabkan *underfitting*, yaitu ketidakmampuan model atau mesin pembelajaran dalam mengenali *trend* pada data yang dapat menyebabkan model pelatihan data tidak dapat mewakili keseluruhan data yang digunakan.

3.6. Pengujian untuk Mengetahui Pengaruh Ekstraksi Fitur Color Histogram

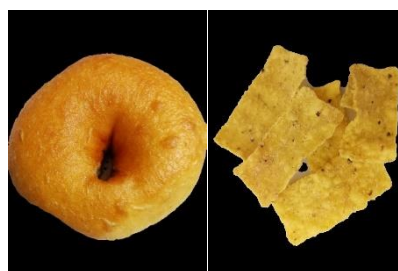
Pengujian ini menguji pengaruh ekstraksi fitur warna dengan menggunakan metode *color histogram*. Pengujian dilakukan dengan distribusi data latih yang sama rata dan data uji sebanyak 15 yang mana mewakili masing-masing kelas. Pengujian dilakukan menggunakan algoritme klasifikasi LVQ dengan nilai parameter *learning rate* 0,1, jumlah maksimum *epoch* 10, nilai penurunan *learning rate* 0,1, dan nilai minimum *learning rate* 1×10^{-4} .

Pengujian berdasarkan penggunaan ekstraksi fitur warna yaitu *color histogram* menghasilkan akurasi sebesar 67,00%. Akurasi yang didapatkan tidak terlalu tinggi dikarenakan *color model* yang digunakan pada metode *color histogram* adalah RGB.

Tabel 1. Hasil akurasi pengaruh ekstraksi fitur *color histogram*

Ekstraksi Fitur	Klasifikasi LVQ		Akurasi
	Data Latih	Data Uji	
<i>Color Histogram</i>	240	15	67,00%

Ruang warna RGB secara persepsi tidak seragam (beragam warna), misalnya pada variasi yang sama pada nilai dari komponen tidak selalu dilihat sebagai variasi warna yang sama. Hal ini berarti dengan kata lain, ukuran dari variasi yang dilihat oleh manusia berbeda dari jarak matematis (Hussain, Rao dan Praveen, 2013). Ruang warna RGB juga bergantung pada perangkat yang digunakan (Afifi dan Ashour, 2012). Pada data latih yang digunakan juga terdapat beberapa kelas makanan yang memiliki warna yang mirip. Contohnya pada Gambar 6 merupakan citra kelas makanan yang memiliki warna antara kuning, oranye dan coklat.



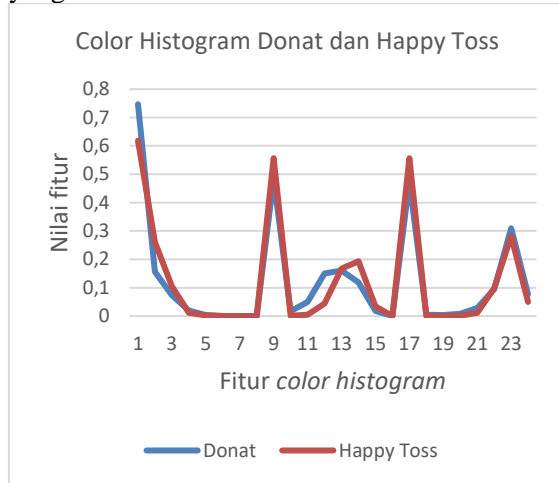
Gambar 6. Citra donat dan Happy Toss

Keberagaman warna pada data latih memengaruhi proses pelatihan algoritme klasifikasi agar menghasilkan kelas yang sesuai dengan data uji yang diberikan. Metode *color histogram* menggunakan rentang atau bin warna yang membagi setiap komponen warna dalam rentang jarak tertentu. Jika citra makanan memiliki warna yang mirip, maka bin warna yang dihasilkan pada citra-citra tersebut juga tidak jauh berbeda antar kelas.

Hal ini menyebabkan semakin sulit untuk mengklasifikasikan data uji tersebut ke kelas yang benar, sehingga memengaruhi akurasi yang dihasilkan. Pada fitur warna, dilakukan pengujian terhadap donat dan program mengklasifikasikan data tersebut sebagai Happy Toss. Hasil ekstraksi fitur warna menunjukkan bahwa *color histogram* antara donat dan Happy Toss memiliki nilai yang mendekati mirip. Hal ini ditunjukkan oleh grafik pada Gambar 7.

Karena kemiripan yang terjadi di antara kedua kelas citra, klasifikasi yang dihasilkan

menjadi tidak akurat. Oleh karena itu, ruang warna RGB yang hanya menghitung warna *red*, *green*, dan *blue*, belum cukup untuk mengklasifikasikan citra hanya dengan warna. Karena citra berwarna memiliki komponen-komponen lainnya yang lebih signifikan, misalnya *brightness*, *luminance*, *saturation* dan komponen-komponen lainnya yang dapat memberikan akurasi mengenai informasi warna yang lebih akurat.



Gambar 7. Grafik *color histogram* citra donat dan Happy Toss

3.7. Pengujian untuk Mengetahui Pengaruh Ekstraksi Fitur GLCM

Pengujian kedua yaitu menguji pengaruh ekstraksi fitur tekstur dengan menggunakan metode GLCM.

Pengujian berdasarkan penggunaan ekstraksi fitur tekstur yaitu *gray level co-occurrence matrix* menghasilkan akurasi sebesar 53,33%. Akurasi yang didapatkan lebih rendah daripada hanya menggunakan ekstraksi fitur warna.

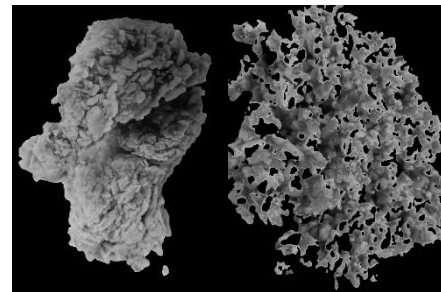
Tabel 2. Hasil akurasi pengaruh ekstraksi fitur GLCM

Ekstraksi Fitur	Klasifikasi LVQ		Akurasi
	Data Latih	Data Uji	
GLCM	240	15	53,33%

Berbeda dengan fitur warna, fitur tekstur pada citra dipengaruhi dengan bentuk citra tersebut. Pada data latih yang digunakan, data pada kelas yang sama memiliki bentuk citra yang berbeda-beda, demikian juga tekstur yang terdapat pada data latih tersebut berbeda-beda. Contohnya pada kelas ayam crispy dan nasi merah. Pada pengujian fitur tekstur, dilakukan

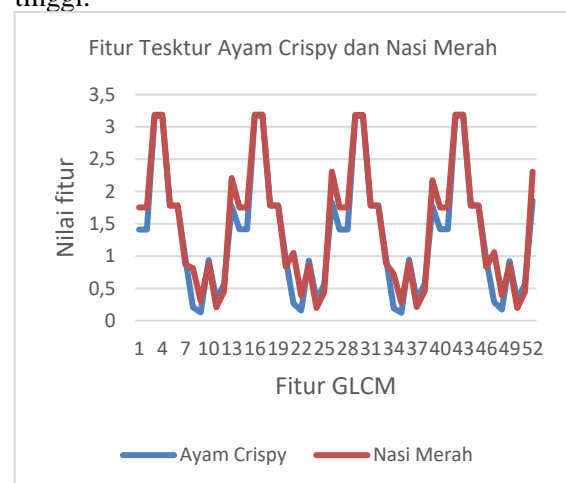
pengujian terhadap ayam crispy dan program mengklasifikasikan data tersebut sebagai nasi merah. Secara visual dapat kita lihat bahwa citra ayam crispy dan nasi merah memiliki tekstur yang hampir mirip, perbandingan citra ditunjukkan pada Gambar 8.

Ekstraksi fitur tekstur pada ayam crispy dan nasi merah menghasilkan nilai yang jika dibentuk grafik, memiliki pola dan nilai yang hampir mirip. Hal ini ditunjukkan pada Gambar 8. Berdasarkan grafik tersebut, hasil klasifikasi tentu menjadi tidak akurat karena kedua citra memiliki nilai tekstur yang hampir sama sehingga fitur-fiturnya menjadi tidak signifikan untuk memberi klasifikasi yang benar.



Gambar 8. Citra *gray level* ayam crispy dan nasi merah

Faktor yang menyebabkan terjadinya kemiripan nilai fitur tekstur karena data citra kedua kelas memiliki tekstur yang mirip, sehingga faktor pemilihan data yang memiliki tekstur yang benar-benar berbeda dapat menambah kemungkinan akurasi menjadi lebih tinggi.

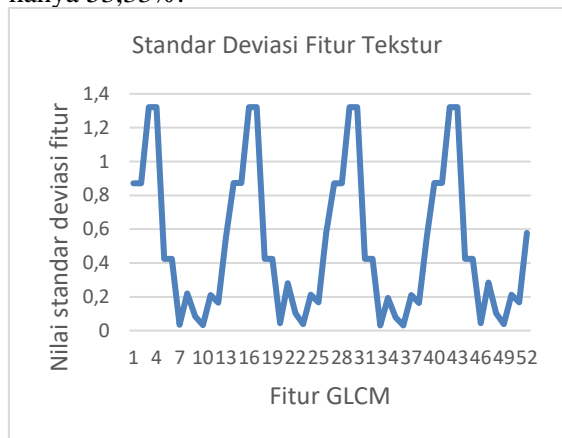


Gambar 9. Grafik GLCM citra ayam crispy dan nasi merah

Untuk mengetahui variansi pada seluruh data latih dapat dilakukan dengan menghitung standar deviasi pada masing-masing fitur. Beberapa fitur pada ekstraksi tekstur membentuk

pola yang berulang, ditunjukkan pada Gambar 10 yaitu 13 fitur pertama dari grafik adalah fitur *mean* hingga *entropy* dari sudut 0°, dan 13 fitur selanjutnya adalah fitur *mean* hingga *entropy* dari sudut 45° dan seterusnya.

Hal ini menunjukkan bahwa tidak adanya keberagaman pada fitur GLCM terhadap data latih yang digunakan. Hal ini memengaruhi proses klasifikasi sehingga hasil akurasi yang dihasilkan dengan menggunakan fitur tekstur hanya 53,33%.



Gambar 10. Grafik standar deviasi data latih untuk fitur tekstur

3.8. Pengujian untuk Mengetahui Pengaruh Ekstraksi Fitur Color Histogram dan GLCM

Pengujian ketiga yaitu menguji pengaruh ekstraksi kedua fitur yaitu warna dan tekstur dengan menggunakan metode *color histogram* dan GLCM. Pengujian berdasarkan penggunaan ekstraksi fitur warna dan tekstur yaitu *color histogram* dan GLCM menghasilkan akurasi sebesar 53,33%.

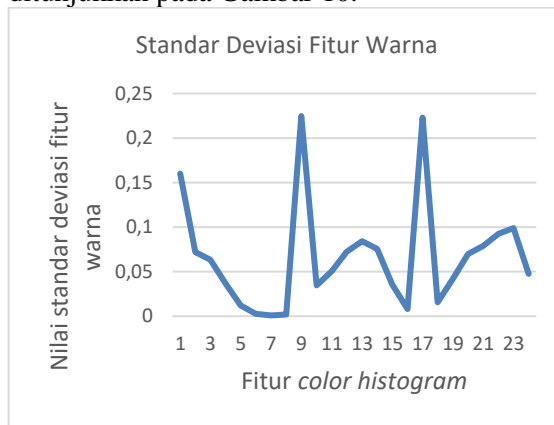
Hasil akurasi yang didapatkan sama dengan hasil akurasi yang didapatkan dengan menggunakan fitur tekstur saja.

Tabel 3. Hasil akurasi pengaruh ekstraksi fitur *color histogram* dan GLCM

Ekstraksi Fitur	Klasifikasi LVQ		Akurasi
	Data Latih	Data Uji	
Color Histogram	240	15	67,00%
GLCM	240	15	53,33%
Color Histogram dan GLCM	240	15	53,33%

Untuk mengetahui variansi pada data latih terhadap ekstraksi fitur *color histogram*, dan GLCM, dihitung standar deviasinya. Standar deviasi pada fitur *color histogram* ditunjukkan pada Gambar 11, dan pada fitur GLCM

ditunjukkan pada Gambar 10.



Gambar 11. Grafik standar deviasi fitur *color histogram*

Berdasarkan grafik pada gambar, fitur *color histogram* memiliki standar deviasi yang lebih bervariasi jika dibandingkan dengan fitur GLCM. Oleh karena itu, penggunaan klasifikasi hanya dengan fitur warna saja menghasilkan nilai akurasi yang lebih tinggi daripada dengan fitur tekstur. Hal ini terjadi juga pada penggunaan klasifikasi dengan kedua fitur, karena fitur tekstur memiliki jumlah total 52 fitur, sedangkan fitur warna hanya memiliki total 24 fitur, mengakibatkan fitur tekstur lebih dominan sehingga klasifikasi yang dihasilkan program sama dengan klasifikasi yang dihasilkan fitur tekstur saja.

4. KESIMPULAN

Berdasarkan pengujian dan analisis yang telah dilakukan, kesimpulan yang didapatkan adalah penggunaan algoritme *learning vector quantization* untuk klasifikasi citra makanan berdasarkan ekstraksi fitur *color histogram* dan GLCM menghasilkan nilai akurasi terbaik sebesar 53,33% dengan nilai parameter terbaik yaitu *learning rate* sebesar 0,1, nilai maksimum *epoch* sebesar 2, nilai penurunan *learning rate* sebesar 0,01, dan nilai minimum *learning rate* 0,01 pada data latih 240 dan data uji 15. Ekstraksi fitur *color histogram* dan GLCM berpengaruh terhadap hasil akurasi yang diperoleh. Berdasarkan pengujian, hasil akurasi tertinggi diperoleh dengan menggunakan ekstraksi fitur *color histogram* saja yaitu sebesar 67,00%, dan akurasi yang diperoleh dengan menggunakan ekstraksi fitur GLCM saja dan menggunakan ekstraksi kedua fitur menghasilkan nilai yang sama yaitu sebesar 53,33%. Ekstraksi fitur *color histogram* merupakan ekstraksi ideal karena nilai-nilai fitur

color histogram memiliki variansi yang lebih signifikan dibanding ekstraksi fitur GLCM.

Berdasarkan pengujian dan analisis, saran yang dapat diberikan adalah sebagai menggunakan ruang warna selain RGB seperti CIE*U*V*, atau HSV yang mempertimbangkan intensitas cahaya (*luminance*) atau *chrominance* dan memilih fitur-fitur yang akan digunakan berdasarkan data citra, yang mana hanya memberikan informasi yang signifikan untuk dapat membedakan data satu dengan data lainnya sehingga dapat menambah akurasi klasifikasi.

5. DAFTAR PUSTAKA

- Afifi, A.J. dan Ashour, W.M. 2012. Image Retrieval Based on Content Using Color Feature. ISRN Computer Graphics, pp.1–11.
- Anami, B.S. dan Burkpalli, V.C. 2009. Texture based Identification and Classification of Bulk Sugary Food Objects. ICGST-GVIP Journal, 9(4), pp.9–14.
- Arifianto, A.S., Sarosa, M. dan Setyawati, O. 2014. Klasifikasi Stroke Berdasarkan Kelainan Patologis dengan Learning Vector Quantization. Jurnal EECCIS, 8(2), pp.117–122.
- Ciocca, G., Napoletano, P., dan Schettini, R., 2018. CNN-based Features for Retrieval and Classification of Food Images. Computer Vision and Image Understanding.
- Eleyan, A. dan Demirel, H. 2011. Co-occurrence Matrix and Its Statistical Features as a New Approach for Face Recognition. Turkish Journal of Electrical Engineering and Computer Sciences, 19(1), pp.97–107.
- Hall-Beyer, M. 2017. Practical Guidelines for Choosing GLCM Textures to Use in Landscape Classification Tasks Over a Range of Moderate Spatial Scales. International Journal of Remote Sensing, 38(5), pp.1312–1338.
- Hussain, C.A., Rao, D.V. dan Praveen, T. 2013. Color Histogram Based Image Retrieval. International Journal of Advanced Engineering Technology, pp.63–66.
- K, Y.G., Santoso, I. dan Isnanto, R.R. 2011. Klasifikasi Citra dengan Matriks Ko-Okurensi Aras Keabuan (Gray level co-occurrence matrix-GLCM) pada Lima Kelas Biji-Bijian.
- Karegowda, A.G., Samreen, S. dan Pushpalatha, K.R. 2015. Automated Detection and Classification of Images Using Color and Texture Features-A Survey. International Journal of Innovative Research in Computer and Communication Engineering, 3(5), pp.248–253.
- Madenda, S. 2015. Pengolahan Citra dan Video Digital. Jakarta: Penerbit Erlangga.
- Martinuva, E.D. 2015. Implementasi Learning Vector Quantization (LVQ) dalam Pemilihan Keminatan (Studi Kasus: Program Studi Informatika Universitas Brawijaya. S1. Universitas Brawijaya.
- Mathai, K.J. dan Soni, D., 2015. An Efficient Content Based Image Retrieval System Based on Color Space Approach Using Color Histogram and Color Correlogram. Fifth International Conference on Communication Systems and Network Technologies, pp.488-492.
- Meskaldji, K., Boucherkha, S. dan Chikhi, S. 2009. Color Quantization and Its Impact on Color Histogram Based Image Retrieval Accuracy. 2009 1st International Conference on Networked Digital Technologies, NDT 2009, pp.515–517. 86
- Patil, N.K., Malemath, V.S. dan Yadahalli, R.M. 2011. Color and Texture Based Identification and Classification of Food Grains Using Different Color Models and Haralick Features. International Journal, 3(12), pp.3669–3680.
- Sela, E.I. dan Hartati, S. 2010. Pengenalan Jenis Penyakit THT Menggunakan Jaringan Learning Vector Quantization.