

# CONTENT-BASED IMAGE RETRIEVAL UNTUK PENCARIAN PRODUK PONSEL

Nickolas Cornelius Siantar<sup>1</sup>, Janson Hendryli<sup>2</sup>, Dyah Erny Herwindiati<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknik Informatika, Fakultas Teknik Informasi Universitas Tarumanagara  
Jl. Let. Jend. S. Parman No. 1, Jakarta 11440 Indonesia

Email: <sup>1</sup>nikolas.535150077@stu.untar.ac.id, <sup>2</sup>jansonh@fti.untar.ac.id, <sup>3</sup>dyahh@fti.untar.ac.id

## Abstrak

Pada saat ini, ponsel dan toko online merupakan sesuatu yang tidak dapat dipisahkan dari manusia. Begitu banyak jenis ponsel bermunculan setiap tahunnya sehingga menyebabkan manusia bingung dalam mengenali ponsel tersebut. Pada program pengenalan ponsel ini digunakan Histogram of Oriented Gradient untuk mengambil fitur berupa bentuk ponsel, Color Quantization untuk mengambil fitur warna, dan Local Binary Pattern untuk mengambil fitur tekstur ponsel. Hasil dari pengambilan fitur berupa fitur vektor yang digunakan pada Learning Vector Quantization untuk proses pengenalan dengan mencari nilai terkecil Euclidean Distance antara vektor fitur dengan vektor bobot terlatih. Hasil dari program pengenalan ini yaitu program dapat melakukan pengenalan terhadap 16 jenis ponsel dengan akurasi sebesar 9.6%.

**Kata kunci**—Color Quantization, Histogram of Oriented Gradient, Learning Vector Quantization, Local Binary Pattern, Ponsel.

## Abstract

Phone or smartphone and online shop, there is something that cannot be separated with human. There are so many type of smartphones show up in the market that people are confused on which one to get on the online stores. Smartphones recognition is done by using the Histogram of Oriented Gradient to recognize shapes of phones, Color Quantization to recognize the color, and Local Binary Pattern to recognize texture of the phones. The output of the Feature Extractor is a feature vector which is used on the LVQ to process recognize through finding the smallest Euclidean Distance between the trained vectors. The result of this paper is an application that can recognize 16 phone types using the image with the accuracy of 9.6%.

**Keywords**—Color Quantization, Histogram of Oriented Gradient, Learning Vector Quantization, Local Binary Pattern, Smartphone.

## 1. PENDAHULUAN

Seiring dengan perkembangan teknologi informasi (TI) yang sangat pesat, kebutuhan manusia pun terus meningkat. Manusia selalu mencari sesuatu untuk dapat menjalankan kehidupannya dengan lebih mudah. Banyak penemuan pada bidang TI yang memberi berbagai kemudahan bagi manusia. Dengan demikian, manusia dapat menggunakan waktunya secara maksimal dan efisien.

Salah satu penemuan teknologi informasi yang memberikan kemudahan bagi manusia adalah mesin pencari. Teknologi ini sangat mempermudah manusia untuk mencari informasi yang dibutuhkannya dalam memenuhi kebutuhan sehari-hari. Mesin pencari berpengaruh besar dalam berbagai bidang kehidupan manusia, contohnya dalam bidang edukasi, ekonomi, sosial

budaya, dan berbagai bidang lainnya. Namun, manusia selalu menginginkan hal lebih dari yang sudah ada, sehingga mesin pencari ini terus dikembangkan.

Mesin pencari dikembangkan untuk dapat digunakan dengan semudah mungkin, salah satunya yang sedang terus dikembangkan adalah mesin pencari menggunakan gambar atau foto atau dikenal juga dengan mesin pencari visual. Mesin pencari visual pertama kali diluncurkan oleh TinEye pada tahun 2008 dan pada tahun 2010 Google meluncurkan Google Goggles yang menjadikan pencarian dapat dilakukan langsung menggunakan kamera telepon seluler (ponsel). Mesin pencarian visual yang dimiliki TinEye pada saat itu hanya mampu melakukan pencarian mengenai landmark suatu negara namun tidak dapat mendeteksi garis tepi dari objek [1].

*Content-Based Image Retrieval* (CBIR) adalah sistem untuk melakukan pencarian dengan melakukan deteksi pada objek dalam citra digital yang merupakan suatu kemajuan dalam perangkat lunak mesin pencari. Awalnya CBIR ditujukan untuk mengatasi permasalahan munculnya koleksi gambar dalam skala besar. Pada CBIR, citra secara otomatis akan diindeks dengan melakukan peringkasan konten visual berdasarkan fitur yang diekstrak seperti warna, bentuk, dan tekstur [2].

Pada penulisan makalah ini akan dibahas mengenai CBIR untuk melakukan pencarian produk ponsel dengan menggunakan metode *Histogram of Oriented Gradient* (HOG), *Color Quantization*, dan *Local Binary Pattern* (LBP) sebagai *feature extraction* dan metode *Learning Vector Quantization* (LVQ) sebagai metode pembelajaran dan pengklasifikasian. Masukkan perangkat lunak ini berupa citra digital RGB dan keluarannya berupa hasil pengenalan atau hasil deteksi dari produk yang dikenali.

Metode HOG digunakan untuk melakukan pengambilan fitur bentuk dari citra. Langkah yang dilakukan pada metode HOG yaitu *preprocessing*, pada tahap ini citra disiapkan untuk diolah seperti pengaturan ukuran citra dengan rasio 1:2. Selanjutnya citra akan dihitung nilai besaran dan nilai arahnya. Citra yang sudah memiliki kedua nilai tersebut dibagi menjadi beberapa sel 8x8 untuk dilakukan proses pembentukan histogram. Pembentukan histogram dilakukan dengan melakukan perhitungan berdasarkan nilai besaran gradien dan arah gradien. Setelah histogram terbentuk maka dilakukan normalisasi lalu seluruh histogram yang membentuk vektor digabungkan membentuk vektor fitur raksasa. Pada metode *Color Quantization*, warna tertentu akan dijadikan acuan untuk pengelompokan warna. Nilai warna pada citra akan dihitung menggunakan *Euclidean Distance* untuk mencari jarak terdekat antara nilai warna dengan nilai warna acuan. Nilai warna yang memiliki jarak terdekat dengan nilai acuan, akan diambil dan dimasukkan ke dalam kelompok nilai acuan tersebut. Hasil dari proses tersebut akan membentuk histogram sehingga dapat diketahui komposisi warna pada citra tersebut. Model warna yang digunakan pada umumnya adalah model warna RGB. Pada metode LBP, fitur yang akan diambil merupakan tekstur dari citra. Pertama citra dikonversikan menjadi citra grayscale, lalu citra akan diproses dengan operator LBP. Pada operator LBP, 8 nilai piksel pada sel berukuran 3x3 akan dibandingkan dengan nilai piksel pusat lalu akan dilakukan proses konversi kode biner. Kode biner ini memiliki nilai desimal yang akan disimpan pada matriks baru berukuran sesuai citra semula dan histogram yang akan membentuk vektor fitur. Hasil dari ketiga metode tersebut akan digabungkan menjadi suatu vektor fitur yang akan digunakan menjadi vektor masukkan pada metode LVQ.

Pada metode LVQ, akan dilakukan pelatihan terlebih dahulu secara terawasi dengan menggunakan vektor fitur yang ada. Setelah melakukan pelatihan, jaringan LVQ mampu untuk menentukan atau mengenali suatu objek dengan menggunakan jarak *Euclidean Distance* terkecil. Pelatihan berupa penyesuaian bobot untuk setiap koneksi dan juga melatih jaringan agar dapat mengenali suatu pola pada citra yang telah dikumpulkan dalam vektor fitur. Bobot

yang telah dilatih akan disimpan pada *file* csv untuk digunakan pada saat pengujian. Hasil dari pengenalan program ini berupa nama atau jenis dari ponsel. Pengujian dilakukan dengan memberi masukan berupa citra yang sudah dikumpulkan sebelumnya untuk data uji. Citra akan diproses melalui *feature extraction* sehingga menghasilkan suatu vektor fitur dan akan dicari nilai *Euclidean Distance* terkecil antara vektor fitur dengan bobot yang sudah dilatih. Bobot yang memiliki jarak terkecil dengan vektor masukan akan dipilih sebagai hasil pengenalan.

## 2. METODE PENELITIAN

Program yang dirancang menggunakan metode *Histogram of Oriented Gradient* (HOG), *Color Quantization* dan *Local Binary Pattern* (LBP) sebagai *feature extraction* dan metode *Learning Vector Quantization* (LVQ) untuk proses pengenalan produk. Berikut adalah dasar teori yang digunakan pada perancangan program ini :

### 2.1 Citra Digital

Citra digital merupakan representasi dari gambar nyata sebagai serangkaian angka yang dapat disimpan dan ditangani oleh komputer digital. Untuk menerjemahkan gambar menjadi angka, citra dibagi menjadi area kecil yang disebut piksel (elemen gambar). Untuk setiap piksel, alat perekam citra mencatat angka atau sekumpulan angka kecil yang mendeskripsikan beberapa properti piksel ini, seperti kecerahan (intensitas cahaya) atau warnanya. Angka-angka disusun dalam array baris dan kolom yang sesuai dengan posisi vertikal dan horizontal dari piksel dalam gambar [3]. Kualitas suatu citra dipengaruhi oleh jumlah piksel dan variasi nilai piksel yang disebut dengan resolusi. Resolusi dinyatakan dalam jumlah piksel per inci (ppi). Semakin tinggi resolusi suatu citra maka semakin tinggi pula ketajaman atau detail citra tersebut.

### 2.2 Content-Based Image Retrieval

*Content-Based Image Retrieval* (CBIR) adalah sistem pencarian citra berdasarkan konten atau fitur visual seperti warna, bentuk, dan tekstur atau struktur gambar [4]. Pada metode lama, pengindeksan citra dimulai dari menyimpan citra pada database dan mengaitkannya dengan kata kunci, mengaitkannya dengan deskripsi yang dikategorikan ini memakan waktu. Dalam CBIR, setiap citra yang disimpan dalam basis data memiliki fitur yang dibandingkan dengan fitur citra lainnya. Hal ini melibatkan dua langkah:

1. *Feature Extraction*, yaitu proses mengekstraksi fitur pada citra ketingkat yang dapat dibedakan.
2. *Matching*, merupakan proses pencocokan fitur-fitur untuk menghasilkan hasil yang serupa secara visual.

### 2.3 Histogram of Oriented Gradient

*Histogram of Oriented Gradient* (HOG) merupakan suatu metode yang digunakan untuk mengambil fitur berupa distribusi arah gradien dari suatu gambar. Dalam HOG, gradien dari suatu gambar sangat berguna sebagai fitur untuk mengenali objek karena nilai gradien yang sangat besar terdapat di sekitar sudut atau tepi objek, sehingga banyak informasi yang dapat diambil untuk diproses. Fitur tersebut dapat digunakan untuk mengetahui bentuk objek. Berikut proses yang dilakukan dalam HOG [5]:

1. Tahapan pertama ada preprocessing, yaitu melakukan resize pada citra menjadi ukuran 64 x 128. Pada tahap ini terdapat kendala berupa aspek rasio pada citra yang beragam. Untuk menghasilkan hasil deteksi yang baik, maka dilakukan cropping agar citra memiliki aspek rasio 1:2, misalnya 100:200, 128:256, atau 1000:2000.
2. Setelah citra berukuran 64 x 128, maka akan dilakukan proses perhitungan gradien vektor pada setiap piksel. Untuk menghitung gradien vektor, perlu terlebih dahulu menghitung

gradien horizontal dan gradien vertikal. Perhitungan dapat dilakukan dengan mudah menggunakan filter dengan Sobel operator atau kernel berikut

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

G<sub>x</sub>

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

G<sub>y</sub>

Gambar 1 Sobel operator

Setelah mendapatkan nilai gradien horizontal dan vertikal, maka selanjutnya dapat menghitung nilai gradien vektor dan arah menggunakan rumus berikut.

$$g = \sqrt{g_x^2 + g_y^2} \quad (1)$$

$$\theta = \arctan \frac{g_y}{g_x} \quad (2)$$

Keterangan:

- g = Nilai gradien
- x, y = koordinat horizontal dan vertikal citra
- θ = arah dari gradien

Hasil dari perhitungan tersebut menghasilkan citra gradien yang menghapus banyak informasi non-esensial seperti latar belakang yang berwarna konstan.

3. Selanjutnya menghitung *Histogram of Oriented Gradients* dengan memasukkan citra ke sel 8x8. Citra yang dimasukkan ke sel 8x8 akan memberikan representasi yang ringkas. Representasi ini selain lebih ringkas, representasi ini juga lebih kuat terhadap gangguan atau noise. Setiap gradien sel mengandung 2 nilai yaitu besaran dan arah per piksel sehingga terdapat 8x8x2 = 128 angka. Pada akhir proses ini, 128 angka tersebut akan disimpan pada histogram 9 wadah. Histogram yang digunakan merupakan sebuah vektor atau array 9 wadah yang berisikan sudut 0, 20, 40, 80, 100, 120, 140, dan 160. Setiap sel 8x8 mengandung 2 nilai tersebut akan dimasukkan ke histogram 9 wadah dengan menyesuaikan nilai gradien arah.
4. Tahap berikutnya yaitu normalisasi blok 16x16, hal ini ditujukan agar histogram tidak terpengaruh dengan variasi pencahayaan karena pada tahap sebelumnya gradien citra cenderung sensitif terhadap efek pencahayaan secara keseluruhan. Jika citra dibuat lebih gelap dengan cara membagi 2 setiap nilai pada citra, maka nilai besaran gradien akan terbagi 2 juga dan nilai histogram akan terbagi pula. Maka normalisasi vektor ini dilakukan untuk menghilangkan skala. Blok 16x16 memiliki 4 histogram yang dapat digabungkan untuk membentuk vektor elemen 36x1 dan dapat dinormalisasikan seperti normalisasi vektor 3x1. Lalu blok 16x16 dipindahkan 8 piksel berikutnya dan proses akan dilakukan berulang.
5. Proses terakhir pada HOG adalah menghitung nilai vektor fitur HOG. Untuk menghitung vektor fitur akhir seluruh bagian citra, seluruh vektor 36x1 digabungkan menjadi vektor raksasa. Besar dari vektor tersebut dapat dihitung sebagai berikut. Banyak posisi dari blok 16x16 yang terdapat pada citra yaitu 7 pada horizontal dan 15 pada vertikal sehingga totalnya ada 105 posisi. Setiap blok 16x16 diwakili oleh 36x1 vektor, maka ketika digabungkan akan didapatkan panjang vektor 3780.

#### 2.4 Color Quantization

Metode *Color Quantization* digunakan untuk mengambil fitur warna dan memperkecil jumlah dari kemungkinan kombinasi warna. Kombinasi warna pada RGB dengan rentang nilai dari 0 sampai 255 adalah 16.777.216. Hal tersebut akan memakan waktu untuk diproses. *Color Quantization* untuk proses ini menggunakan *K-Means* untuk melakukan pengelompokan warna. Berikut tahapan untuk proses *Color Quantization* dengan menggunakan *K-Means* [6]:

1. Melakukan identifikasi jumlah klaster yang diperlukan. Jumlah klaster akan menentukan nilai dari K.
2. Tentukan nilai dari K yang merepresentasikan sebuah warna yang akan digunakan.
3. Menghitung jarak seluruh nilai warna dengan nilai K. Proses perhitungan jarak dapat menggunakan *Euclidean Distance*.

$$D = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (3)$$

Keterangan:

D = nilai *Euclidean Distance*

R<sub>1,2</sub> = nilai warna merah pada pixel citra dan K

G<sub>1,2</sub> = nilai warna hijau pada pixel citra dan K

B<sub>1,2</sub> = nilai warna biru pada pixel citra dan K

4. Mengklasifikasikan setiap nilai warna dengan jarak terdekat terhadap nilai K.

Dengan menggunakan *Color Quantization*, fitur warna dapat dikelompokkan secara efektif tanpa mengganggu kinerja sistem secara keseluruhan.

#### 2.5 Local Binary Pattern

*Local Binary Pattern* (LBP) digunakan untuk mengambil fitur berupa tekstur objek. Sebelum melakukan proses LBP, warna citra digital dikonversikan menjadi citra grayscale dan dilakukan penambahan padding. Selanjutnya proses yang dilakukan dalam LBP yaitu membandingkan nilai piksel dengan 8 piksel tetangganya pada sel berukuran 3x3. Nilai piksel kedelapan tetangganya akan dikurangi dengan nilai piksel pusat lalu akan diubah menjadi nilai 0 jika hasilnya negatif dan 1 jika hasil pengurangan positif. Selanjutnya akan diperoleh nilai biner dengan menggabungkan seluruh kode biner searah jarum jam mulai dari yang paling kiri atas dan ini berkaitan dengan nilai desimal yang nantinya akan dihasilkan. Kode biner yang diperoleh akan dikonversikan menjadi nilai desimal. Hasil dari nilai desimal akan disimpan dalam histogram dan disimpan pada matriks baru dengan ukuran yang sesuai dengan citra semula.

#### 2.6 Learning Vector Quantization

*Learning Vector Quantization* (LVQ) adalah jaringan saraf tiruan yang menggunakan pembelajaran *supervised* atau pembelajaran yang diawasi. LVQ menggunakan pembelajaran yang diawasi sehingga jaringannya akan diberikan pola pelatihan dengan pengklasifikasian yang sudah dikenali bersama dengan hasil dari outputnya. Setelah melakukan proses pelatihan, LVQ akan mengklasifikasikan vektor masukan dengan menugaskannya ke kelas yang sama dengan unit keluaran. Klasifikasi dilakukan dengan cara menemukan unit keluaran yang memiliki jarak *Euclidean Distance* paling dekat dengan vektor masukan.

Secara arsitektur, LVQ mempunyai kemiripan dengan arsitektur *Kohonen Self-Organizing Maps* (KSOM). Parameter yang digunakan pada proses pelatihan yaitu:

X = vektor pelatihan (x<sub>1</sub>, x<sub>2</sub>, ..., x<sub>n</sub>)

T = kelas untuk pelatihan vektor x

$w_J$  = bobot vektor untuk keluaran unit ke J

CJ = kelas untuk hasil keluaran ke J

$\|x-w_j\|$  = *Euclidean Distance* antara vektor masukan dengan bobot vektor untuk unit keluaran ke j

Berikut adalah algoritma yang dilakukan pada LVQ[7]:

1. Inisialisasi vektor masukan, inisialisasi laju pembelajaran  $\alpha(0)$ .
2. Selama kondisi berhenti tidak terpenuhi, lakukan langkah 3-7
3. Untuk setiap pelatihan vektor masukan  $x$ , lakukan langkah 4-5.
4. Cari nilai J sehingga  $\|x-w_j\|$  bernilai minimum.
5. Perbaharui nilai  $w_J$ , dengan aturan sebagai berikut:  
Jika  $T = CJ$ , lakukan  
 $w_J(\text{baru}) = w_J(\text{lama}) + \alpha[x - w_J(\text{lama})]$ ;  
Jika  $T \neq CJ$ , lakukan  
 $w_J(\text{baru}) = w_J(\text{lama}) - \alpha[x - w_J(\text{lama})]$ ;
6. Kurangi laju pembelajaran.
7. Uji kondisi berhenti:  
Kondisi dapat berupa jumlah pasti iterasi yang dilakukan atau tingkat pembelajaran yang mencapai nilai tertentu.

Bobot hasil dari pelatihan pada metode LVQ ini akan disimpan pada file csv dan digunakan untuk proses pengujian. Proses pelatihan bobot ini ditujukan agar proses pengenalan pada objek dapat dilakukan dengan baik. Pada proses pelatihan, tingkat pembelajaran dan jumlah epoch atau jumlah banyaknya pelatihan dilakukan ini berpengaruh juga pada tingkat akurasi dari jaringan syaraf tiruan LVQ.

Pada pengujian LQV, fungsi aktivasi yang digunakan adalah fungsi aktivasi linear. Tujuan menggunakan fungsi aktivasi linear adalah memperoleh keluaran yang sama dengan masukan, sesuai dengan rumus fungsi linear yaitu  $y = x$ . Fungsi aktivasi F1 akan memetakan  $y_{in1}$  ke  $Y1 = 1$  apabila  $|x - W1| < |x - W2|$ , dan  $y1 = 0$  jika sebaliknya. Demikian pula fungsi aktivasi F2 akan memetakan  $y_{in2}$  ke  $Y2 = 1$  apabila  $|x - W2| < |x - W1|$ , dan  $y2 = 0$  jika sebaliknya.

Setelah dilakukan pelatihan, maka akan diperoleh jaringan LVQ yang memiliki bobot-bobot antar neuron yang sudah mampu melakukan pengklasifikasian. Berikut adalah algoritma untuk pengujian pada metode LVQ:

1. Masukkan data yang akan diuji beserta bobot yang telah dilatih.
2. Tentukan nilai J sehingga  $\|x-w_j\|$  bernilai minimum.
3. Tentukan neuron dengan jarak minimum.
4. Objek akan dikenali dengan nilai minimum vektor masukan dengan kelas keluaran.

### 3. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan memberikan masukan berupa citra ponsel sebanyak 250 citra yang terdiri dari 10 citra setiap kelasnya. Pengujian dilakukan dengan menggunakan bobot yang dilatih dengan beberapa perbedaan epoch dan tahap pembelajaran. Proses pengujian dilakukan dengan memproses setiap citra melalui *feature extraction* yang akan menghasilkan fitur vektor. Fitur vektor tersebut akan digunakan untuk proses pengenalan pada metode LVQ oleh bobot terlatih.

Pengujian ditujukan untuk mencari bobot terlatih yang mampu mengenali setiap jenis ponsel dengan baik. Model latih dibedakan berdasarkan jumlah epoch dan nilai tahap pembelajaran. Perbedaan tersebut dapat menghasilkan bobot terlatih yang mampu mengenali

jenis ponsel dengan baik. Pengujian dilakukan dengan menggunakan model latih 250 epoh dan tahap pembelajaran 0,002, model latih 250 epoh dan tahap pembelajaran 0,001, model latih 500 epoh dengan tahap pembelajaran 0,001, model latih 500 epoh dengan tahap pembelajaran 0,0001, model latih 750 epoh dengan tahap pembelajaran 0,00001. Berikut hasil dari pengujian terhadap model latih tersebut:

**Tabel 1** Nilai rata-rata hasil pengujian

Model Latih	Tahap Pembelajaran	Precision	Recall	F1-score
250 epoh	0.002	0.082	0.092	0.084
250 epoh	0.001	0.1584	0.092	0.08612
500 epoh	0.001	0.109	0.076	0.076
500 epoh	0.0001	0.131	0.096	0.093
750 epoh	0.00001	0.075	0.148	0.0728

Dari hasil pengujian yang telah dilakukan, didapatkan hasil pengujian sebagai berikut:

1. Penambahan jumlah epoh dan pengecilan nilai tahap pembelajaran tidak selalu menghasilkan bobot terlatih terbaik. Hal ini dapat dilihat pada model latih 500 epoh dengan tahap pembelajaran 0,0001 menghasilkan nilai precision, recall, dan f1-score lebih baik disbanding model latih 750 epoh dengan tahap pembelajaran 0,00001.
2. Hasil terbaik yang didapatkan dari model latih yaitu model latih dengan 500 epoh dan tahap pembelajaran 0,0001. Model latih mampu menghasilkan nilai rata-rata precision sebesar 0,131, recall sebesar 0,096, dan f1-score 0,093.
3. Model latih 500 epoh dengan tahap pembelajaran 0,0001 mampu mengenali sebanyak 16 jenis ponsel dari total 25 jenis. Hal tersebut merupakan nilai tertinggi dari model latih lainnya.

Hasil pengujian memiliki akurasi yang kurang baik dikarenakan terdapat beberapa kendala seperti fitur warna dan tekstur tidak dapat memberikan kontribusi yang baik untuk program pengelanaan karena warna dan tekstur setiap ponsel memiliki kemiripan yang sangat tinggi. Sementara pada metode *Histogram of Oriented Gradient* (HOG) untuk pengambilan fitur bentuk mengalami kendala berupa detil ponsel yang merupakan bagian dari karakteristik ponsel tidak dapat diambil dengan baik.. Contohnya yaitu bentuk kamera dan merk dari ponsel tidak dapat terdeteksi dengan baik.

Permasalahan ini menyebabkan proses pengenalan menjadi tidak baik. Selain itu terdapat permasalahan berupa data yang memiliki banyak gangguan seperti pencahayaan, latar belakang pengambilan citra ponsel, latar belakang dari layar ponsel yang beragam, benda-benda kecil disekitarnya seperti stiker IMEI pada belakang ponsel, pantulan pada permukaan ponsel terhadap benda dihadapannya, bagian dari ponsel yang tertutup benda lain seperti tertutup jari atau tangan dan juga perbedaan sudut pada saat pengambilan ponsel. Gangguan tersebut menyebabkan proses pengambilan fitur terganggu. Dalam proses pengujian juga terjadi hal serupa. Gangguan tersebut mempersulit proses pengenalan sehingga cukup banyak ponsel yang dideteksi sebagai ponsel dengan jenis lainnya.

Pada proses pelatihan bobot oleh metode *Learning Vector Quantization* (LVQ) juga terdapat masalah berupa tingkat kemiripan dari fitur yang tinggi. LVQ melakukan pelatihan untuk menghasilkan *feature space* untuk setiap kelasnya dengan baik. Namun dengan kondisi fitur yang memiliki karakteristik yang sulit dibedakan, *feature space* yang terbentuk menjadi sangat padat, sehingga perbedaan jarak antara bobot satu dengan yang lain tidak jauh. Hal ini menyebabkan setiap hasil pengujian dapat menghasilkan pengenalan yang berbeda. Perbedaan sudut pengambilan citra, pencahayaan, latar belakang pengambilan citra, latar belakang dari

layar ponsel, dan benda yang menutupi bagian ponsel seperti stiker IMEI atau jari tangan dapat mempengaruhi hasil pengenalan produk ponsel.

Selain permasalahan berupa gangguan data latih dan data uji, terdapat juga permasalahan berupa jumlah fitur yang lebih banyak dibanding jumlah data latih dengan anggapan jumlah fitur adalah jumlah variabel dan jumlah data adalah jumlah observasi. Pada makalah ini, program pengenalan memiliki fitur sebanyak 3813 dan data latih sebanyak 1500. Jumlah data latih yang lebih sedikit menyebabkan beberapa variabel tidak terobservasi dengan baik.

#### 4. KESIMPULAN

Kesimpulan yang diperoleh setelah melakukan pengujian pada program yang dirancang adalah berikut:

1. Seluruh modul dapat berfungsi dengan baik.
2. Hasil dari pengenalan oleh program aplikasi memiliki akurasi yang kurang baik.
3. Metode untuk pengambilan fitur harus dapat mengambil fitur dengan karakteristik yang kuat atau detail sehingga dapat digunakan untuk membedakan jenis-jenis ponsel.
4. Pengaruh dari pencahayaan, latar belakang layar ponsel dan latar belakang pengambilan citra ponsel, benda disekitar ponsel, bagian dari ponsel yang tertutup jari atau tangan, pantulan pada permukaan ponsel dan sudut pengambilan citra ponsel dapat mempengaruhi hasil pengenalan.
5. Model latih terbaik mampu mendeteksi sebanyak 16 jenis ponsel dan 9 kelas tidak terdeteksi.

#### DAFTAR PUSTAKA

- [1] Rebecca Sentence, 2016, The new wave of visual search: what it can do, and what might be possible, <https://searchenginewatch.com/2016/10/13/the-new-wave-of-visual-search-what-it-can-do-and-what-might-be-possible/>, diakses tanggal 27 Agustus 2018.
- [2] Ying Li, C.C. Jay Kuo, and X. Wan, 2002, *Image Databases: Search and Retrieval of Digital Imagery*, John Wiley & Sons, Inc., Hoboken.
- [3] Digital Image, Computer Sciences, 2002, <https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/digital-images>, diakses tanggal 4 September 2018.
- [4] K. Kranti Kumar dan T. Venu Gopal, 2010, CBIR: Content Based Image Retrieval, <https://www.researchgate.net/publication/235634738>, diakses tanggal 1 September 2018.
- [5] Satya Mallick, Histogram of Oriented Gradients, 2016, <https://www.learnopencv.com/histogram-of-oriented-gradients/>, diakses tanggal 30 Agustus 2018.
- [6] Perfect Makanju, K Means and Image Quantization [Part 2], 2018, <https://medium.com/consonance/k-means-and-image-quantization-part-2-be0a62c50c11>, diakses tanggal 10 September 2018.
- [7] Laurene Fausett, Fundamental of Neural Networks Architectures, 1994, *Fundamental of Neural Network: Algorithms, and Applications*, Prentice Hall, Upper Saddle River.