Pembangunan Model *Clustering* Untuk Pengelompokan Citra *Near Duplicate*

Trian Annas Thoriq Sumarjadi[1], Dr. Ir. Rinaldi Munir[2], M.T, Fariska Zakhralativa Ruskanda, S.T, M.T[3]
Program Studi Teknik Informatika,
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung
40132, Indonesia
Email: tatstrian@gmail.com[1], rinaldi@informatika.org[2], fariska@informatika.org[3]

***Abstract*—Dalam mesin pencarian gambar pada *e-commerecei* seperti Bukalapak.com, gambar yang dihasilkan merupakan gambar yang *near-duplicate*. Untuk mengatasi hal tersebut, diperlukan sebuah model *clustering* yang mampu mengelompokan gambar berdasarkan kemiripan gambar. Untuk dapat melakukan *clustering* diperlukan metode ekstraksi fitur yang tepat. Terdapat beberapa metode ekstraksi fitur seperti SIFT, ORB, PCA-SIFT, dan SURF. Hasil Ekstraksi fitur yang didapatkan dari metode tersebut kemudian dilakukan *clustering* dengan metode DBSCAN.**

Keywords—DBSCAN; near-duplicate; ekstraksi fitur; keypoints;

I. PENDAHULUAN

Dalam era yang serba digital ini, *online marketplace* seperti Bukalapak.com membutuhkan teknologi untuk pengguna agar bisa melakukan pencarian produk-produk di Bukalapak.com dengan menggunakan gambar yang diterima dari pengguna. Namun, dalam melakukan pencarian berdasarkan gambar, hasil yang didapat adalah produk-produk yang hanya mirip dengan gambar tersebut.

Dalam dunia teknologi, gambar-gambar yang mirip tersebut biasa disebut dengan *near-duplicate* citra. Berbeda halnya dengan istilah *duplicate* citra. Pada *duplicate* citra yang memiliki istilah dua gambar atau lebih yang memiliki ukuran dimensi, komposisi warna, dan ukuran *file* yang sama. Sedangkan *Near-duplicate* citra adalah dua gambar atau lebih yang memiliki tingkat kemiripan dekat tetapi tidak sepenuhnya mirip. Kemiripan antara dua gambar tersebut dipengaruhi oleh beberapa faktor seperti perbedaan sudut pengambilan gambar, pencahayaan, dan pengaturan kamera [1].

Salah satu solusi untuk menyaring gambar yang *near-duplicate* diperlukan adanya proses untuk melakukan pengelompokan *near-duplicate* citra terlebih dahulu pada *database* yang sudah ada. Pengelompokan *near-duplicate* citra ini bertujuan untuk menyaring hasil dari pencarian, sehingga hanya salah satu anggota dari *cluster* saja yang akan ditampilkan. Oleh karena itu hasil pencarian menjadi lebih beragam dan tidak dipenuhi oleh gambar yang *near-duplicate.*

Untuk dapat melakukan pengelompokan gambar berdasarkan kemiripan gambarnya, sebelum melakukan *clustering* diperlkukan satu proses yaitu ekstraksi fitur pada gambar. Terdapat beberapa metode ekstraksi fitur seperti SIFT, ORB, PCA-SIFT, dan SURF. Hasil dari proses tersebut kemudian dilakukan *clustering* dengan metode DBSCAN.

II. DASAR TEORI

A. Ekstraksi Fitur SIFT

SIFT atau *Scale Infariant Feature Transform* adalah algoritma untuk mencari fitur pada sebuah gambar. Fitur yang diperoleh dari algoritma SIFT termasuk *scale invariant.* Hasil yang didapat dari algoritma SIFT ini tidak sebatas pada *scale variant* saja, namun meskipun gambarnya telah dilakukan operasi rotasi, perubahan skala, perubahan pencahayaan, fitur yang didapat akan tetap sama. Oleh karena itu, algoritma SIFT ini mampu mengidentifikasi *near-duplicate* citra.

Terdapat 6 langkah utama dalam melakukan ekstraksi fitur SIFT, yaitu:

1. *Scale Space Speak Selection*. Pada tahap ini citra akan melakukan proses *bluring* menggunakan *Gaussian blur* hingga dicapai 4 gambar yang sudah dilakukan *bluring*. Total terdapat 5 gambar dengan gambar awal ikut ke dalamnya. Setiap gambar memiliki tingkat *blur* yang berbeda. Tahap tersebut disebut dengan *first octave*. Kemudian dibuat *octave* selanjutnya dengan mengubah ukuran awal gambar menjadi lebih kecil dan melakukan *bluring* kembali sampai memiliki empat *octave.*
2. *LoG Approximations.* Pada tahap ini setiap *octave* dilakukan proses operasi *Laplacian of Gaussians (LoG),* sehingga didapati 4 gambar dari setiap *octave.*
3. *Finding Keypoints.* Pada tahap ini dilakukan proses *local maxima/minima* dengan melakukan iterasi pada setiap *pixel* dengan *pixel* tetangganya dengan menggunakan 3 gambar, sehingga sekali proses pengecekan dilakukan 26 kali perbandingan.
4. *Removing Low Contrast Feature.* Pada tahap ini yang memiliki nilai intensitas di bawah nilai tertentu akan dihapus.
5. *Keypoints Orientation.* Proses ini melakukan pencarian *gradient direction* dengan skala 36. Kemudian dipilih *gradient direction* yang memiliki jumlah *80%* dari jumlah maksimum *pixel.*

6. *Generatin feature*. Pada tahap ini setiap *keypoints* yang telah didapat, akan dibuat *window* berukuran 16×16, kemudian dibagi empat menjadi ukuran 4×4. Setiap *window* 4×4 tersebut dilakukan proses *gradient magnitude* dan *orientation*. Hasil tersebut dibuat 8 bin histogram. Setiap *keypoints* akan memiliki vector dengan ukuran 128.

B. Ekstraksi Fitur PCA-SIFT

Ekstraksi fitur PCA-SIFT merupakan gabungan metode PCA dan SIFT dimana hasil dari ekstraksi fitur SIFT kemudian direduksi dimensi *descriptor*nya menggunakan PCA. PCA atau *Principal Component Analysis* adalah algoritma untuk melakukan reduksi dimensi. Yang pertama adalah hitung *S* pada

$$S = (X - \bar{X})^T (X - \bar{X}) \quad (1)$$

Dimana *S* adalah matriks *covariance*, *X* adalah matriks masukan dan *X bar* adalah *mean* dari matriks *X*. Kemudian dekomposisikan matriks *covariance* tersebut ke *eigenvactor* dan *eigenvalue*. Jika dimensi yang dikehendaki adalah sebesar *p* dimensi, maka ambil sebanyak *p eigenvactor* dengan nilai *eigenvalue p*-terbesar.

Untuk mendapatkan matriks yang telah di reduksi, hitung *Z* pada

$$Z = XW \quad (2)$$

Dimana *Z* adalah matriks yang sudah direduksi dengan cara mengkalikan matriks *X* dengan matriks *eigenvactor W*.

C. Ekstraksi Fitur ORB

ORB atau *Oriented* FAST *and Rotated* BRIEF adalah algoritma untuk mencari fitur pada sebuah gambar sama halnya dengan SIFT, tetapi yang membedakan dengan SIFT adalah bahwa pada ORB ini merupakan *rotation invariant* dan *resistant to noise* [2]. Oleh karena itu ORB dapat mengenali objek pada dua gambar. Objek tersebut sudah mengalami perubahan posisi seperti melakukan rotasi dan juga tahan terhadap gambar yang memiliki *noise*.

ORB merupakan algoritma ekstraksi fitur yang mengadopsi dua teknik yaitu FAST (*Features from Accelerated Segment Test*) dan BRIEF (*Binary Robust Independent Elementary Features*). FAST digunakan untuk mencari keypoints dan BRIEF digunakan sebagai descriptor keypoint.

Dalam melakukan deteksi *keypoints* dilakukan modifikasi dengan menghitung momen gambar untuk *x* dan *y* pada area yang melingkar sebesar radius *r*. Momen tersebut digunakan untuk mendapatkan orientasi dari *keypoints*. Sudut yang didapatkan digunakan pada *descriptor* sehingga *descriptor* yang dihaslikan peka terhapat rotasi.

Bentuk *descriptor* yang didapatkan berupa *binary* dengan ukuran 256 bit. Berbeda dari SIFT dan SURF, yang merupakan vector. Sehingga dalam melakukan pencocokan fitur dilakukan dengan menggunakan pengukuran jarak dengan metode *hamming distance*.

D. Ekstarksi Fitur SURF

SURF atau *Speeded Up Robust Features* adalah algoritma untuk mencari fitur pada sebuah gambar sama halnya dengan SIFT, tetapi yang membedakan dengan SIFT adalah pada pendekatan dalam melakukan *keypoints detector* dan *descriptor*nya. Dalam melakukan deteksi *keypoints*, Peneliti [3] menggunakan *Fast-Hessian Detector* atau *hessian* matriks. Determinan dari *hessian* matriks digunakan untuk memilih lokasi dan skala dari pada menggunakan *hessian laplace detector*.

Untuk menghitung nilai determinan dari *hessian* matriks, yang pertama adalah menerapkan konvolusi dengan *gaussian kernel* lalu menghitung turunan kedua. Untuk melakukan kedua hal tersebut peneliti [3] menggunakan pendekatan dari *box* filter.

SURF memiliki kecepatan yang sama untuk berapun *size* dari filter. Oleh karena itu untuk menganalisis *scale-space* dilakukan dengan menaikkan *scale* dari filter. Jadi setiap octave yang baru akan memiliki ukuran filter yang lebih besar. Untuk mendapatkan lokasi *keypoints* dilakukan *nonmaximum suppresions* pada 3x3x3 *neighborhood* diterapkan.

Untuk mendapatkan *descriptor*, pertama membuat *window* berukuran 20×20 untuk setiap *keypoints*. Kemudian buat menjadi empat bagian *sub-window* yang berukuran 5×5. Dari setiap sub-*window* hitung jumlah respons *wavelet* pada arah *vertical* dan arah *horizontal*. Untuk mendapatkan informasi yang peka terhadap intensitas cahaya, respons *wavelet vertical* dan horizontal dibuat menjadi nilai mutlak. Sehingga setiap sub-*window* terdapat empat nilai *wavelet*. Oleh karena itu karena terdapat empat sub-*window* maka pada satu *keypoints* terdapat 64 nilai *wavelet*, sehingga dimensi yang dihasilkan dari proses *descriptor* adalah *vector* dengan panjang sebesar 64 dimensi.

E. Fitur Matching SIFT, SURF dan PCA-SIFT

Fitur *matching* adalah sebuah proses untuk melakukan pencocokan antara *keypoints* pada gambar pertama dengan *keypoints* yang ada pada gambar kedua. Pada proses pencocokan, setiap *keypoint* pada gambar pertama dipasangkan dua titik *keypoints* terdekat pada gambar kedua. Sehingga setiap *keypoints* pada gambar pertama akan memiliki dua *keypoints* dari gambar kedua.

Jumlah *keypoints* yang dihasilkan dari ekstraksi fitur memiliki jumlah yang berbeda untuk setiap gambar. Hal ini mengakibatkan jumlah *keypoints* dari hasil pencocokan fitur akan mengakibatkan jumlah yang berbeda. Contohnya adalah apabila gambar A dibandingkan dengan gambar B, akan memiliki pasangan 15 *keypoints* yang berbeda dengan gambar B yang dibandingkan dengan gambar A. Oleh karena itu, untuk membandingkan 2 buah gambar perlu dilakukan dua kali perbandingan, yaitu gambar A dengan gambar B begitu pula sebaliknya.

Namun hasil dari pencocokan fitur ini belum menghasilkan pasangan *keypoints* yang baik. Oleh karena itu perlu adanya *filtering* lebih lanjut dengan menggunakan *ratio test* yang diusulkan pada penelitian [4].

Untuk menyaring *keypoint* yang sudah berpasangan digunakan metode *ratio test*, dimana setiap *keypoint* pada gambar pertama akan memiliki dua *keypoint* terbaik pada gambar kedua. *Test* dilakukan dengan cara membandingkan

nilai jarak pada pasangan *keypoint* pertama terbaik dengan pasangan *keypoint* kedua terbaik.

Pasangan *keypoint* pertama merupakan pasangan yang bagus, sedangkan yang kedua merupakan *noise*. Oleh karena itu semakin jauh nilai perbedaan antara jarak antara pasangan *keypoint* pertama dengan kedua adalah semakin baik. Seberapa jauh nilai tersebut kitalah yang menentukan, seperti pasangan *keypoint* kedua memiliki nilai 0.45 lebih kecil dari pasangan *keypoint* pertama.

F. Clustering DBSCAN

DBSCAN atau *Density-based Spatial Clustering of Applications with Noise* adalah salah satu algoritma *clustering*. Algoritma ini melakukan pencarian titik-titik yang memiliki jumlah tetangga sebesar minimum *point* dan berada pada jangkauan *epsilon.* Setiap titik yang memenuhi syarat ini akan dijadikan sebagai *core point*. setiap *core point* yang terhubung akan dijadikan sebagai satu *cluster*. Untuk titik 17 yang tidak memenuhi minimum *point*, titik ini akan dijadikan sebagai *border point*, sedangkan titik yang tidak memenuhi syarat minimum *point* dan jarak *epsilon* maka titik ini akan dijadikan *noise point*. *Noise point* merupakan titik yang berada di outlier.

Dimana *Z* adalah matriks yang sudah direduksi dengan cara mengalikan matriks *X* dengan matriks *eigenvactor W*.

III. RANCANGAN SOLUSI

Dikembangkan alur proses dari *preprocessing* data, ekstraksi fitur, desain parameter, pembangunan model dan evaluasi model. Model *clustering* yang dibangun menggunakan DBSCAN dengan berbagai metode ekstraksi fitur yang akan dibandingkan seperti SIFT, ORB, PCA-SIFT dan SURF. Modifikasi DBSCAN juga diperlukan agar dapat melakukan perhitungan jarak antara dua gambar dan dapat melakukan prediksi terhadap data uji.

A. Data Preprocessing

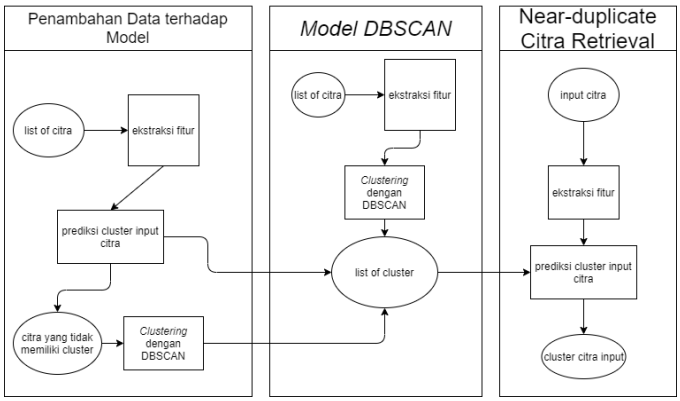
Dataset yang digunakan dari bukalapak.com tidak semuanya bisa digunakan. Diperlukan suatu proses manual untuk menyaring data. Data yang bukan merupakan *near-duplicate* akan dihilangkan. Kemudian setelah melakukan seleksi tersebut seluruh data diubah formatnya menjadi *grayscale.* Ukuran gambar tidak diubah, hal ini dilakukan semata-mata untuk menjaga kualitas informasi yang terdapat pada *dataset.*

B. Pembangunan Model

Pembangunan model *clustering* menggunakan metode DBSCAN yang dimodifikasi. Modifikasi yang dilakukan adalah penambahan fungsi prediksi dan juga penyesuaian fungsi perhitungan jarak antar gambar. Model DBSCAN dipilih karena bila dibandingkan dengan model *clustering* k-*means*, algoritma k-*means* tidak bisa melakukan *clustering* untuk data yang sudah diekstraksi fiturnya dengan metode SIFT, ORB, PCA-SIFT, dan SURF. Hal ini disebabkan karena jumlah *keypoints* yang dihasilkan berbeda-beda untuk setiap gambar, lalu kemudian perhitungan jarak yang dilakukan tidak bisa menggunakan cara yang biasa dilakukan. Perhitungan jarak yang dilakukan untuk dua gambar adalah dengan menghitung jumlah *keypoints* yang berpasangan antara gambar satu dengan yang lainnya. Semakin banyak *keypoint* yang dihasilkan semakin dekat jarak kedua gambar tersebut. Tetapi cara ini memiliki kelemahan yaitu tidak semua pasangan *keypoints* yang terbentuk merupakan pasangan yang benar. Oleh karena itu perlu dilakukan penyaringan pada pasangan *keypoints* yang terbentuk agar menyisakan pasangan-pasangan yang benar saja.

Untuk dapat melakukan prediksi, model akan mencari salah satu perwakilan objek pada salah satu cluster, dimana objek tersebut merupakan objek yang memiliki tetangga terbanyak. Selain itu dicatat juga jarak terjauh antar objek untuk setiap *cluster* yang ada. Langkah pertama dalam melakukan prediksi adalah dengan mencari kandidat *cluster* untuk data uji dengan cara membandingkan jarak datauji dengan perwakilan objek dari setiap *cluster*. Apabila jaraknya lebih kecil dari jarak terjauh pada *cluster* tersebut, maka *cluster* tersebut merupakan kandidat *cluster* untuk data uji. Kemudian dari kandidat-kandidat *cluster* yang ada, data uji akan dibandingkan dengan seluruh data yang ada pada kandidat *cluster*. Apabila ditemukan lebih dari sekian nilai yang ditentukan, maka *cluster* tersebut merupakan *cluster* untuk data uji dan pencarian langsung diberhentikan.

Arsitektur yang dikembangkan terdiri dari tiga komponen, yaitu komponen untuk penambahan dataset, komponen model clustering, dan komponen prediksi datauji. Arsitektur tersebut dapat dilihat pada gambar 1.



Gambar 1. Arsitektur yang dikembangkan

IV. IMPLEMENTASI

Implementasi dilakukan dengan meggunakan *google colab.* Program dan dataset disimpan pada *google drive.* Program dikembangakan menggunakan bahasa *pyhton* dan menggunakan library *OpenCV*. Implementasi terdiri dari pembersihan dataset, pembangunan model, pelatihan model, dan pemilihan model terbaik.

A. Dataset

Dataset yang digukan berjumlah 461 gambar dan sudah dilakukan penyaringan data. Jumlah *cluster* pada gambar tersebut adalah sebesar 25 *cluster*. Jumlah data tersebut tidak bisa lebih besar lagi karena keterbatasan *resource* karena implementasi DBSCAN tidak bisa menggunakan VGA, sehingga perlu waktu yang lama untuk melakukan *clustering* dan juga *google colab* hanya memberikan batas maksimal 12 jam untuk sekali *run.*

Dataset tersebut dilakukan *split* untuk setiap *cluster*. Setiap cluster dibagi menjadi data latih dan data uji dengan perbandingan 80:20. Hal ini dilakukan agar proses evaluasi bisa menilai semua cluster yang ada. Rata-rata setiap cluster memiliki jumlah kisaran antara 10 sampai 20 gambar.

B. Desain Hyperparameter

Setiap metode ekstraksi fitur memiliki Desain *hyperparameter* untuk melakukan *ratio test*, tetapi pada metode ekstrasi fitur ORB tidak dapat melakukan *ratio test* dikarenakan bentuk *descriptor* yang berbeda, sehingga untuk melakukan fitur *matching*, metode ORB menggunakan *hamming distance.*

Setiap metode memiliki konfigurasi *hyperparameter default* untuk *minimum points* dan *epsilon* yang sama, kecuali pada metode SIFT. Konfigurasi tersebut dapat dilihat pada Tabel I.

Tabel I Hyperparameter default untuk setiap metode

Metode	*Lowe_ratio*	*Min_pts*	*epsilon*
SIFT	0.65	50	3
ORB	-	70	3
PCA-SIFT	0.85	70	3
SURF	0.75	70	3

Pada tabel I, terlihat bahwa metode SIFT berbeda sendiri dengan metode yang lain bila dilihat dari *min_pts* dan *epsilon*, hal ini karena metode yang lain menggunakan *hyperparameter default* berdasarkan hasil terbaik dari metode SIFT. Pada metode PCA-SIFT, dimensi *descriptor* direduksi menjadi 32 dari 128.

Setelah melakukan pelatihan model untuk setiap metode yang ada, didapati hasil terbaik untuk masing masing model dengan konfigurasi terbaiknya masing-masing, namun metode ORB dan PCA-SIFT gagal dalam melakukan *clustering*, sehingga hanya dua metode saja yang berhasil melakukan *clustering* yaitu SIFT dan SURF dengan konfigurasi *hyperparameter* terbaik dapat dilihat pada tabel II.

Tabel II Hyperparameter terbaik

Metode	*Lowe_ratio*	*Min_pts*	*epsilon*
SIFT	0.65	70	3
SURF	0.75	75	3

Score yang didapat dari dua metode tersebut dapat dilihat pada tabel III. Pada tabel III dapat dilihat bahwa, SIFT lebih unggul dari SURF dengan nilai akurasi sebesar 0.9029 sedangkan SURF sebesar 0.8058.

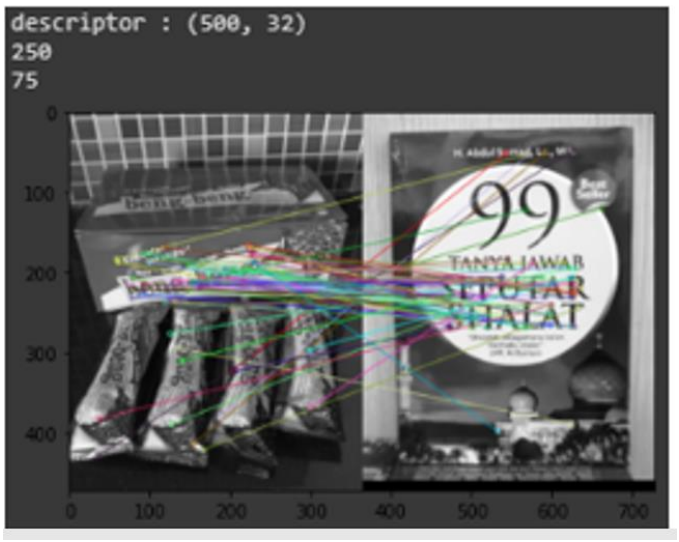
Tabel III Score terbaik dari metode SIFT dan SURF

Metode	Penilaian	
	Purity	Akurasi
SIFT	1	0.9029
SURF	1	0.8058

V. EVALUASI

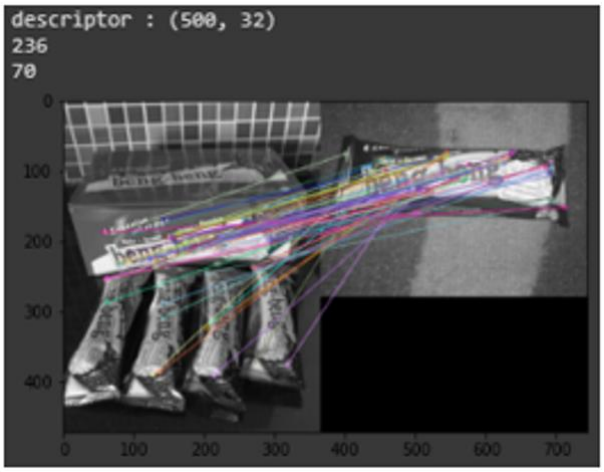
Pada percobaan yang telah dilakukan diketahui bahwa metode ekstrasi fitur ORB dan PCA-SIFT gagal dalam melakukan *clustering*. Penyebap kegagaln *clustering* bukan dikarenakan nilai konfigurasi *hyperparameter* yang salah melainkan dalam melakukan *clustering*, model tidak bisa membeda gambar yang *near-duplicate* atau bukan.

Pada metode ekstraksi fitur ORB, diketahui bahwa bentuk *descriptor*nya berbeda dari yang lain, oleh sebab itu dalam melakukan fitur *matching*, pasangan *keypoints* yang dihasilkan tidak bisa disaring menggunakan *ratio test*, hal ini menyebapkan banyaknya pasangan *keypoints* yang *false positive*. Sehingga dalam membandingkan dua gambar yang benar-benar berbeda, metode ORB tidak dapat membedakannya, seperti yang terlihat pada gambar 2. Terlihat pada gambar 2 bahwa psangan *keypoints* yang terbentuk sebanyak 75 pasang.



Gambar 2 Hasil perbandingan dua gambar yang berbeda

Sedangkan pada gambar 3, ditunjukan bahwa untuk dua gambar yang *near-duplicate*, jumlah *keypoints* yang terbentuk sebesar 72. Hal ini membuktikan bahwa metode ekstraksi fitur ORB tidak dapat membedakan dua buah gambar yang bukan *near-duplicate.* Hal ini bisa terjadi karena disebabkan metode ekstraksi ORB memiliki *descriptor* berbenuk *binary* yang membuat perhitungan jarak antara dua *keypoints* menggunakan *hamming distance* yang berakibat tidak dapat melakukan *ratio test* untuk menyaring pasangan *keypoints* yang *false positive.*



Gambar 3 hasil perbandingan dua gamba yang near-duplicate

Pada metode ekstraksi fitur PCA-SIFT, didapati bahwa model tidak bisa melakukan *clustering*, jumlah *cluster* yang terbentuk hanya dua *cluster* yang terbentuk. Hal ini dikarenakan pengaruh dari penurunan dimensi pada *descriptor* yang membuat *descriptor* mengalami penurunan kualitas dari informasi yang dikandungnya. Sehingga sulit untuk melakukan *clustering.*

Dari Tabel III dapat dilihat bahwa metode ekstraksi fitur SIFT memiliki nilai performa yang lebih baik dari SURF. Pada metode ekstraksi fitur SURF dengan mengubah nilai *hyperparamater epsilon* dari 70 menjadi 75 membuat nilai performa yang lebih baik. Hal ini menunjukkan bahwa setiap metode memiliki nilai *hyperparameter* terbaiknya sendiri.

Data uji yang mengalami kegagalan dalam melakukan prediksi diketahui merupakan objek gambar yang memiliki sudut pengambilan gambar yang sangat berbeda yang mengakibatkan posisi objek pada gambar menjadi miring tidak tegak lurus seperti pada gambar 4. Pada gambar 4 (kiri) dapat dilihat bahwa dengan menggunakan metode ekstraksi fitur SIFT jumlah pasangan *keypoints* yang didapatkan sebesar 33 dan pada gambar 4 (kanan) dapat dilihat jumlah pasangan *keypoints* yang dihasilkan dari metode SURF sebesar 38. Kedua nilai tersebut lebih kecil dari nilai *hyperparameter epsilon* sebesar 70 untuk SIFT dan 75 untuk SURF.

Dari gambar 4 diketahui bahwa metode SURF lebih peka terhadap perbedaan kemiringan objek pada gambar dengan ditunjukkan jumlah pasangan *keypoints* yang dihasilkan lebih banyak dari metode SIFT, meskipun tidak terlalu besar. Hal ini juga telah dibuktikan berdasarkan penelitian [5], Kedua metode SIFT dan SURF bagus untuk *invariant* rotasi, ukuran, *noise*, *blur*, dan pencahayaan. Tetapi Ektrasksi fitur SIFT baik untuk objek pada gambar yang memiliki perbedaan ukuran sedangkan SURF baik untuk gambar yang memiliki perbedaan rotasi gambar dan *noise*. Hal ini dibuktikan dengan hasil performa yang baik karena *dataset* yang digunakan pada percobaan ini setiap gambar memiliki ukuran yang bervariasi, tetapi setiap objek pada gambar tidak memliki perbedaan rotasi yang besar.

Gambar 4 Pencocokan Fitur pada dataset yang gagal melakukan prediksi untuk metode SIFT (kiri) dan metode SURF (kanan)

VI. KESIMPULAN

Berdasarkan hasil eksperimen, diperoleh bahwa model *clustering* DBSCAN mampu melakukan pengelompokan citra berdasarkan kemiripan citra berhasil dibangun dengan menggunakan metode ekstraksi fitur SIFT dan SURF. Nilai *purity* untuk SIFT dan SURF adalah 1. Nilai akurasi untuk SIFT dan SURF berturut-turut adalah 0.9029 dan 0.8058 dengan *hyperparameter* terbaik untuk masing-masing metode.

Penggunaan metode ekstraksi fitur ORB dan PCA-SIFT dalam membangun model *clustering* DBSCAN tidak dapat melakukan pengelompokan citra berdasarkan kemiripan citra karena ekstraksi fitur ORB tidak mampu membedakan gambar yang bukan *near-duplicate* citra sedangkan metode PCA-SIFT, *keypoints* yang dihasilkan tidak *distinctive* lagi.

Model *clustering* DBSCAN dengan metode ekstraksi fitur SIFT memiliki kinerja yang lebih baik dibandingkan metode ekstraksi fitur SURF. Hal tersebut terbukti dengan perbandingan hasil evaluasi berdasarkan *purity* dan akurasi.

Akhir kata meskipun memiliki hasil performa yang bagus, metode yang digunakan masih harus dikembangkan lebih lanjut. Dari segi kecepatan dalam melakukan *clustering* dan prediksi memerlukan waktu yang lama. Hal itu akan berpengaruh kepada efisiensi waktu apabila diterapkan pada sistem yang menggunakan data yang sangat banyak dan memiliki pengguna yang sangat banyak. Ke depannya dalam melakukan *clustering* DBSCAN diperlukan pendekatan secara *parallel* sehingga mempersingkat waktu dalam melakukan *clustering*. Sehingga data yang digunakan bisa lebih banyak dan dapat menghemat waktu yang digunakan.

REFERENCES

[1] Zhang, Y., Zhang, Y., Sun, J., Li, H., Zhu, Y. Learning Near Duplicate Image Pairs using Convolutional Neural Networks: 2018

[2] Rublee, E., Rabaud, V., Konolige, K., Bradski, G. ORB: an efficient alternative to SIFT or SURF, 2011

[3] Bay, H., Tuytelaars, T., Gool, L. V. SURF: Speeded Up Robust Features, 2006

[4] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints, 2004

[5] Mistry, D., Banerjee, A. (2017). Comparison of Feature Detection and Matching Approaches: SIFT and SURF

[6] Ester, M., Kriegel, H.P., Sander, J., XuA, X. Density-Based Algorithm for Discovering *Cluster*s in Large Spatial Databases with Noise. 1996