

Analisis Jumlah Video Latih yang Berbeda pada Setiap Kelas untuk Aplikasi *Near Duplicate Video Retrieval* (NDVR)

Ruben Stefanus¹, Riko Arlando Saragih²

Fakultas Teknik, Program Studi Teknik Elektro, Universitas Kristen Maranatha

Jl. Surya Sumantri No.65, Bandung 40164, Indonesia

rstefanus66@gmail.com

riko.as@eng.maranatha.edu

Diterima : 20 Februari 2022

Disetujui 27 Maret 2022

Abstract— *Near Duplicate Video Retrieval* (NDVR) adalah salah satu jenis teknik *retrieval* berbasis video yang sering dilakukan karena banyaknya video yang hampir duplikat dari video asli. Biasanya aktivitas *video retrieval* dilakukan untuk memperoleh manfaat (umumnya komersial) tanpa seijin pemilik hak cipta yang sah. Salah satu isu dalam sistem NDVR adalah ketidakseimbangan jumlah video latih untuk setiap kelas, sehingga di dalam tulisan ini akan dikaji pengaruh jumlah video latih yang berbeda untuk setiap kelas. Metode t-USMVH merupakan rancangan yang akan digunakan dalam sistem NDVR pada penelitian ini. Sementara itu, sistem NDVR yang akan direalisasikan terdiri atas 4 tahapan, yaitu proses ekstraksi keyframe, ekstraksi ciri, pelatihan sistem, serta pencocokan kode hash antarvideo. Pada tahap ekstraksi keyframe, pemilihan keyframe dilakukan berdasarkan jumlah sampel per detik tanpa mengabaikan perubahan citra antardetik. Di tahap kedua, yaitu proses ekstraksi ciri, ada 2 pendekatan yang dipakai, yaitu ekstraksi ciri global (*Hue Saturation Value/HSV*) serta ekstraksi ciri lokal (*Local Directional Pattern/LDP*). Selanjutnya, teknik *backpropagation* merupakan metode untuk pelatihan sistem, dan diakhiri dengan pencocokan kode hash antarvideo menggunakan metode *Hamming Distance* (HD). Variabel yang diuji adalah kombinasi variasi ciri, variasi jumlah iterasi, dan variasi jumlah video latih untuk setiap kelas. Selain itu, ada 2 variasi jumlah video latih untuk setiap kelas yaitu, *imbalance query video* dan *balance query video*. Hasil simulasi menunjukkan bahwa nilai *Mean Average Precision* (MAP) untuk kondisi *balance query video* lebih besar daripada *imbalance query video* pada semua variasi iterasi dan semua variasi ciri. Selain itu, penggabungan ekstraksi ciri lokal dan ciri global pada sistem *balance query video* menghasilkan nilai MAP yang lebih baik daripada hanya menggunakan satu jenis ekstraksi ciri.

Keywords— *Keyframe*, NDVR, t-USMVH, HSV, LDP, *balance query video*, *imbalance query video*, MAP

I. PENDAHULUAN

Keberadaan informasi berupa video di dunia maya semakin berkembang seiring dengan kemajuan teknologi internet. Beberapa aktivitas secara *online* yang menggunakan video seperti pengunggahan, pengunduhan, dan menonton video secara *online* menunjukkan kenaikan yang pesat dalam satu dekade terakhir ini. Hal ini memicu masifnya keberadaan video di internet

sekaligus memunculkan kebutuhan yang meningkat untuk melakukan pengenalan obyek, klasifikasi video, deteksi keaslian video, dan lain-lain [1] [2][3].

Salah satu masalah yang muncul akibat meningkatnya keberadaan video dalam dunia maya adalah masalah video-video yang hampir duplikat dari video asli (*Near Duplicate Video/NDV*). Video yang hampir duplikat dari

video asli ini dapat diperoleh dengan mengubah frame tertentu pada video asli. Pihak yang akan dirugikan akibat aktivitas ini adalah pemilik hak cipta dari video asli tersebut karena video tersebut dapat dimanfaatkan dan dikomersialkan tanpa seijin (sepengetahuan) dari pemilik hak cipta.

Untuk mengatasi masalah ini, *Near Duplicate Video Retrieval* (NDVR) diusulkan untuk mencari dan memeriksa video yang hampir duplikat dari video asli. Secara umum, proses NDVR diawali dengan ekstraksi *keyframe* dari video, dimana *keyframe* adalah merupakan informasi frame citra yang dapat merepresentasikan isi dan informasi yang penting dari suatu video [3][4]. Selanjutnya dilakukan ekstraksi ciri terhadap setiap *keyframe* dan diakhiri dengan melakukan pencarian kemiripan antara video menggunakan informasi ciri tersebut.

Studi terkini tentang NDVR, terutama *large-scale near-duplicate video retrieval*, dilakukan oleh Yanbin Hao dan kawan-kawan [1]. Algoritma yang diusulkan dalam penelitian mereka memiliki akurasi yang tinggi, efisien, dan memiliki skalabilitas (kemampuan algoritma menangani komputasi untuk jumlah video yang besar) yang baik untuk menemukan *near duplicate video*. Selain itu para peneliti juga tidak hanya menggunakan sebuah vektor ciri tunggal (*single feature vector*) dalam pendekatan yang mereka usulkan.

Jenis algoritma yang dipakai dalam penelitian tersebut adalah berbasis *Stochastic Multi-View Hashing* (SMVH). Algoritma ini memanfaatkan beberapa jenis informasi dari video, yaitu ciri global, ciri lokal, dan *keyframe*. Hal ini mengakibatkan kinerja NDVR meningkat tetapi berkonsekuensi dimensi data yang dihasilkan menjadi tinggi, yang berujung kepada waktu komputasi yang lama dan kompleksitas yang tinggi, sehingga memerlukan proses untuk mereduksi dimensi data.

Untuk mempercepat pencarian kemiripan antara video, SMVH menerapkan teknik *hashing*. *Hash* merupakan suatu metode untuk menyandikan (*encode*) obyek seperti video, gambar, atau dokumen menjadi suatu deretan biner yang memiliki ukuran tetap dengan

menggunakan proses pemetaan [2]. Untuk proses pencarian *near duplicate video*, metode *hash* ini memiliki kelebihan karena waktu komputasi yang cepat dan mengurangi penggunaan memori saat komputasi.

Selain masalah dimensi data yang tinggi, ketidakseimbangan jumlah video latih (*imbalance data*) juga merupakan masalah yang dapat mempengaruhi kinerja sistem NDVR[5]. Jumlah kelas video yang satu lebih sedikit (kelas minoritas) atau lebih banyak (kelas mayoritas) dibandingkan dengan jumlah kelas video lainnya menjadi penyebab ketidakseimbangan jumlah video latih, sehingga distribusi kelas video menjadi tidak seimbang. Hal ini berakibat akurasi klasifikasi video untuk kelas mayoritas jauh lebih tinggi daripada kelas minoritas, sementara untuk kasus NDVR, biasanya jumlah *near duplicate video* lebih sedikit (kelas minoritas) dibandingkan jumlah video yang berbeda (kelas mayoritas), tetapi informasi *near duplicate video* sangat penting untuk berhasil diidentifikasi.

Sistem NDVR yang akan direalisasikan dalam penelitian ini adalah berdasarkan metode pendekatan dari Yanbin Hao dan kawan-kawan [2], dimana aspek yang akan dievaluasi adalah pengaruh jumlah video latih untuk setiap kelas pada kinerja sistem. Hal yang berbeda dalam penelitian ini adalah pemilihan jenis deskriptor lokal yang akan digunakan, yaitu *Local Directional Pattern* (LDP), karena deskriptor lokal *Local Binary Pattern* (LBP) yang digunakan Yanbin Hao dan kawan-kawan [2] tidak dapat mengakomodir keberadaan *noise* dan variasi iluminasi yang non-monotonik [6]. Selain itu, pada bagian ekstraksi *keyframe* akan digunakan metode berdasarkan perbedaan tepi dan bentuk poligon dari citra.

II. TEORI PENDUKUNG

Misalnya terdapat sejumlah V video, lalu didapatkan sejumlah *keyframe* yang diekstraksi dari setiap video menggunakan metode *uniform time sampling*. Asumsikan sejumlah n *keyframe* diekstraksi dari V video. Untuk setiap *keyframe*, diekstraksi beberapa ciri untuk mencirikan informasi dari *keyframe* tersebut. Kemudian

informasi dari setiap ciri menjadi masukan untuk metode t-USMVH.

A. Fungsi Hash

Fungsi *hash* adalah fungsi matematis yang mengubah nilai masukan yang memiliki ukuran tertentu menjadi suatu nilai yang memiliki ukuran yang tetap [2]. Berdasarkan ide ini, maka fungsi *hash* mulai digunakan untuk memetakan data yang berukuran besar, seperti gambar, video atau dokumen digital lainnya menjadi kode biner dengan panjang bit tertentu [2]. Persamaan matematis untuk fungsi *hash* dapat dituliskan sebagai berikut:

$$\mathbf{z}_{il} = \text{sigmoid} \left(\sum_{g=1}^m \sum_{j=1}^{d_g} \mathbf{x}_{ij}^{(g)} \mathbf{w}_{jl}^{(g)} + \mathbf{b}_l \right) \quad (1)$$

Keterangan :

\mathbf{z}_{il} : kode hash

sigmoid : fungsi aktivasi

m : tipe ciri (HSV, LDP)

d_g : dimensi dari histogram ciri

$\mathbf{x}_{ij}^{(g)}$: matriks histogram ciri dari *keyframe*

$\mathbf{w}_{ij}^{(g)}$: matriks nilai bobot

\mathbf{b}_l : matriks nilai bias

i, j : indeks *keyframe*

l : panjang kode *hash*

B. Peluang Multi-view Data [2] [14]

Multi-view data adalah data yang memiliki banyak informasi ciri. Peluang *multi-view data* (\mathbf{P}) disebut sebagai peluang aktual. Tahapan perhitungan peluang untuk *multi-view data* (\mathbf{P}) adalah sebagai berikut :

1. Hitung matriks peluang kemiripan antara ciri (\mathbf{p}). Matriks ini mengukur seberapa dekat nilai x_j dari x_i dengan memperhitungkan distribusi *Gaussian* menggunakan Persamaan (2).

$$\mathbf{p}_{j|i}^{(g)} = \frac{\exp \left(- \frac{\| \mathbf{x}_i^{(g)} - \mathbf{x}_j^{(g)} \|_2^2}{2\sigma_{ig}^2} \right)}{\sum_{l \neq i} \exp \left(- \frac{\| \mathbf{x}_i^{(g)} - \mathbf{x}_l^{(g)} \|_2^2}{2\sigma_{ig}^2} \right)} \quad (2)$$

2. Hitung matriks koefisien *Jaccard* (\mathbf{J}). Koefisien *Jaccard* berguna untuk menghitung

seberapa besar kemiripan antara dua ciri *keyframe* menggunakan Persamaan (3).

$$\mathbf{J}_{ij}^{(g)} = \frac{|\mathbf{x}_i^{(g)} \cap \mathbf{x}_j^{(g)}|}{|\mathbf{x}_i^{(g)} \cup \mathbf{x}_j^{(g)}|} \quad (3)$$

3. Hitung matriks total kemiripan (\mathbf{P}_c) dengan persamaan (4).

$$\mathbf{P}_c = \frac{1}{2n} \sum_{g=1}^m \mathbf{J}_{ij}^{(g)} (\mathbf{p}_{j|i}^{(g)} + \mathbf{p}_{i|j}^{(g)}) \quad (4)$$

4. Hitung matriks *with-in* video ($\mathbf{P}_w = \mathbf{p}_{ij}^{(w)}$) dengan Persamaan (5). Matriks *with-in* video ini memberikan informasi tentang *keyframe* yang berasal dari suatu video yang sama.

$$\mathbf{p}_{ij}^{(w)} = \begin{cases} 1, & \text{jika } x_i \text{ dan } x_j \text{ (} i \neq j \text{) diekstraksi} \\ & \text{dari video yang sama} \\ 0, & \text{dalam keadaan lain} \end{cases} \quad (5)$$

5. Hitung matriks peluang *multi-view* (\mathbf{P}) dengan Persamaan (6).

$$\mathbf{P} = \mathbf{0.7 N}(\mathbf{P}_c) + \mathbf{0.3 N}(\mathbf{P}_w) \quad (6)$$

Keterangan :

i, j : indeks *keyframe*

m : tipe ciri (HSV, LDP)

σ_i^2 : varians dari *keyframe* dengan indeks i

n : jumlah ciri yang digunakan

$\mathbf{p}_{i|j}^{(g)}$: transpose dari $\mathbf{p}_{j|i}^{(g)}$

$|\cdot|$: kardinalitas

$\|\cdot\|_2$: *Euclidian Norm*

\mathbf{z}_i : kode *hash* untuk *keyframe* dengan indeks i

C. Peluang Single-view Data [2] [14]

Peluang *single-view data* (\mathbf{Q}) disebut sebagai peluang yang diprediksi, karena setiap kali iterasi nilainya akan berubah. Matriks peluang ($\mathbf{Q} = \mathbf{q}_{ij}$) dibuat berdasarkan kode *hash* yang didapatkan dari fungsi *hash*. Peluang ($\mathbf{Q} = \mathbf{q}_{ij}$) dapat dilihat pada Persamaan (7).

$$\mathbf{q}_{ij} = \frac{(1 + \|\mathbf{z}_i - \mathbf{z}_j\|_2^2)^{-1}}{\sum_{k \neq i} (1 + \|\mathbf{z}_k - \mathbf{z}_i\|_2^2)^{-1}} \quad (7)$$

D. Perbedaan Kullback-Leibler [2]

Perbedaan *Kullback Leibler* adalah ukuran seberapa besar suatu distribusi peluang berbeda dari distribusi peluang pembandingnya.

Perbedaan *Kullback Leibler* adalah salah satu jenis *cost function*. *Cost Function* adalah fungsi yang berguna untuk mengukur seberapa besar kesalahan dalam memodelkan \mathbf{Q} dari \mathbf{P} . Karena *Kullback Leibler* tidak simetris maka dihitunglah fungsi objektif berdasarkan KL_1 dan KL_2 dengan persamaan sebagai berikut :

$$KL_1 = \sum_{t \neq i} p_{it} \log \frac{p_{it}}{q_{it}} \quad (8)$$

$$KL_2 = \sum_{t \neq i} q_{it} \log \frac{q_{it}}{p_{it}} \quad (9)$$

$$O = \lambda KL_1 + (1 - \lambda) KL_2 \quad (10)$$

Keterangan :

KL_1 : *Kullback Leibler 1*

KL_2 : *Kullback Leibler 2*

i, t : *indeks keyframe*

p_{it} : *matriks peluang multi-view data*

q_{it} : *matriks peluang single-view data*

O : *fungsi objektif*

λ : *parameter balancing antara KL_1 dan KL_2*

E. Gradien Descent [2] [14]

Gradient Descent adalah algoritma pengoptimalan yang mencoba menemukan nilai minimum dari suatu fungsi pada setiap iterasi. *Gradient Descent* adalah metode iterasi yang berarti prosesnya dilakukan secara berulang sampai sejumlah iterasi yang ditetapkan atau sampai mencapai suatu kondisi tertentu. Pada penelitian ini, *Gradient Descent* digunakan untuk meminimalkan perbedaan *Kullback Leibler* terhadap parameter model bobot (\mathbf{w}) dan bias (\mathbf{b}).

Tahapan proses *Gradient Descent* adalah sebagai berikut :

1. Inisialisasi nilai bobot (\mathbf{w}) dan bias (\mathbf{b}) secara acak di awal.
2. Hitung nilai gradien untuk fungsi objektif terhadap bobot dan terhadap bias menggunakan Persamaan (11) dan (12).

$$\frac{\partial O}{\partial \mathbf{w}} = \left[\lambda \frac{\partial KL_1}{\partial \mathbf{w}} + (1 - \lambda) \frac{\partial KL_2}{\partial \mathbf{w}} \right] \frac{\partial z}{\partial \mathbf{w}} \quad (11)$$

$$\frac{\partial O}{\partial \mathbf{b}} = \left[\lambda \frac{\partial KL_1}{\partial \mathbf{b}} + (1 - \lambda) \frac{\partial KL_2}{\partial \mathbf{b}} \right] \frac{\partial z}{\partial \mathbf{b}} \quad (12)$$

$$\frac{\partial KL_1}{\partial z} = 4 \sum_t (p_{it} - q_{it}) (1 + \|z_i - z_t\|_2^2)^{-1} (z_i - z_t) \quad (13)$$

$$\frac{\partial KL_2}{\partial z} = 4 \sum_t q_{it} \left(\sum_{k \neq i} q_{ki} \log \frac{q_{ki}}{p_{ki}} - \log \frac{q_{it}}{p_{it}} \right) (z_i - z_t) (1 + \|z_i - z_t\|_2^2)^{-1} \quad (14)$$

$$\frac{\partial z}{\partial \mathbf{w}} = z [1 - z] x_{ij}^{(g)} \quad (15)$$

$$\frac{\partial z}{\partial \mathbf{b}} = z [1 - z] \quad (16)$$

Keterangan :

$\frac{\partial O}{\partial \mathbf{w}}$: nilai gradien fungsi objektif terhadap parameter bobot

$\frac{\partial O}{\partial \mathbf{b}}$: nilai gradien fungsi objektif terhadap parameter bias

$\frac{\partial KL_1}{\partial z}$: nilai gradien *Kullback Leibler 1* terhadap kode hash

$\frac{\partial KL_2}{\partial z}$: nilai gradien *Kullback Leibler 2* terhadap kode hash

$\frac{\partial z}{\partial \mathbf{w}}$: nilai gradien kode hash terhadap parameter bobot

$\frac{\partial z}{\partial \mathbf{b}}$: nilai gradien kode hash terhadap parameter bias

λ : parameter *balancing* antara KL_1 dan KL_2

3. Perbarui nilai bobot (\mathbf{w}) dan bias (\mathbf{b}) dengan Persamaan (17) dan (18)

$$w(t+1) = w(t) + \eta \frac{\partial O}{\partial \mathbf{w}} + \xi(t)(w(t) - w(t-1)) \quad (17)$$

$$b(t+1) = b(t) + \eta \frac{\partial O}{\partial \mathbf{b}} + \xi(t)(b(t) - b(t-1)) \quad (18)$$

Keterangan :

$w(t+1)$: nilai bobot terbaru

$b(t+1)$: nilai bias terbaru

$w(t)$: nilai bobot saat ini

$b(t)$: nilai bias saat ini

$w(t-1)$: nilai bobot sebelumnya

$b(t-1)$: nilai bobot sebelumnya

η : *learning rate*

$\xi(t)$: momentum

4. Ulangi tahap (2) dan (3) sampai mencapai sejumlah iterasi yang ditetapkan.

F. Pembuatan Kode Biner [2]

Proses ini bertujuan untuk membuat kode biner untuk setiap video berdasarkan kode *hash keyframe*. Masukan untuk fungsi ini berupa kode *hash keyframe*. Variabel h_{il} menunjukkan kode biner *keyframe* ke- i dengan panjang kode sebesar l . Variabel Ind_i menunjukkan indeks *keyframe* pada video dan $|\cdot|$ menunjukkan kardinalitas. Untuk membuat kode biner digunakan fungsi ambang $T(x)$ yang akan bernilai 1 jika $x > 0.5$ dan untuk nilai lainnya bernilai 0. Persamaan untuk membuat kode biner dapat dilihat pada persamaan (19).

$$h_{il} = T\left(\frac{1}{|\text{Ind}_i|} \sum_{j \in \text{Ind}_i} z_{jl}\right) \quad (19)$$

Matriks kode biner yang didapatkan akan berukuran $k \times 100$ dengan nilai setiap elemennya antara 0 atau 1. Bagian baris berukuran k menunjukkan jumlah video, sedangkan bagian kolom berukuran 100 menunjukkan panjang kode biner 100 bit.

III. TEKNIK EKSTRAKSI YANG DIUSULKAN

Pada penelitian ini diusulkan metode ekstraksi *keyframe* menggunakan gabungan dari *uniform time sampling* dan *shot boundary detection*. Metode ekstraksi *keyframe* yang diusulkan menggunakan algoritma jarak *hausdorff* dan deteksi tepi *canny*. Kemudian digunakan ciri LDP sebagai pengganti ciri LBP yang digunakan pada penelitian sebelumnya.

A. Jarak Hausdorff

Jarak *Hausdorff* adalah metode untuk menghitung kemiripan antara dua bentuk polygon [9]. Metode ini merupakan pengembangan dari metode jarak *euclidian*. Jarak *euclidian* dikembangkan karena metode ini memiliki beberapa kekurangan dalam menghitung kemiripan dua bentuk poligon. Kekurangannya adalah jarak *euclidian* terpendek tidak mempertimbangkan keseluruhan bentuk poligon, dan tidak memperhitungkan posisi rotasi dari poligon. Padahal dalam pencocokan dua bentuk citra, keseluruhan bentuk poligon dan posisi rotasi

dari poligon merupakan informasi yang sangat menentukan kemiripan dua bentuk citra.

Hausdorff Distance dapat dihitung dengan persamaan berikut :

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (20)$$

Dengan:

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (21)$$

Keterangan :

A, B : hasil deteksi tepi *canny* terhadap suatu *frame* citra.

B. Deteksi Tepi Canny

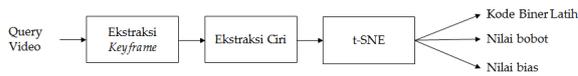
Deteksi tepi pada citra merupakan proses untuk mencari perbedaan intensitas yang menyatakan batas-batas suatu objek dalam citra. Tujuan pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu objek di dalam citra. Proses deteksi tepi citra dilakukan dengan mencari piksel-piksel yang memiliki nilai intensitas yang sangat berbeda dengan piksel-piksel yang berdekatan. Deteksi tepi *canny* adalah salah satu operator untuk proses deteksi tepi pada citra [8] [18].

C. Local Directional Pattern (LDP)

Local Directional Pattern (LDP) adalah deskriptor lokal yang merepresentasikan tekstur pada citra [6]. Pada deskriptor lokal, citra yang menjadi masukan akan dibagi menjadi beberapa bagian citra dengan ukuran yang lebih kecil dari ukuran citra masukan. Kemudian untuk setiap blok ukuran 3x3 piksel pada citra bagian akan dikonvolusikan dengan delapan arah kirsch *mask*.

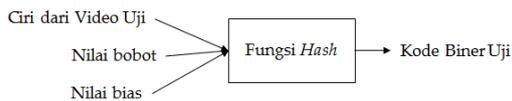
IV. PERANCANGAN SISTEM

Secara garis besar sistem NDVR dapat dibagi menjadi 3 diagram blok. Pada diagram blok pertama terdiri atas proses ekstraksi *keyframe* dari setiap video yang akan dilatih. Kemudian proses kedua adalah ekstraksi ciri dari setiap *keyframe* yang didapatkan menggunakan deskriptor lokal dan deskriptor global. Dan yang terakhir adalah proses pelatihan sistem menggunakan algoritma t-SNE. Proses ini dapat digambarkan dengan diagram blok pada Gambar 1.



Gambar 1. Diagram Blok Ekstraksi Ciri dan Pelatihan

Pada diagram blok kedua dilakukan proses untuk mendapatkan kode biner uji. Masukannya berupa ciri yang didapat dari 2 buah video uji serta nilai bobot dan bias yang didapatkan dari hasil diagram blok pertama.



Gambar 2. Diagram Blok Pembuatan Kode Biner Uji

Pada diagram blok ketiga dilakukan proses pengujian antara kode biner latih dan kode biner uji menggunakan metode jarak *hamming*. Keluaran dari jarak *hamming* adalah jumlah perbedaan bit biner dan rank kemiripan video.



Gambar 3. Diagram Blok Pengujian

V. DATA PENGAMATAN DAN ANALISIS

Pada penelitian ini pengambilan data dilakukan terhadap 25 video pada query video 3 dan 25 video pada query video 4. Pengambilan data dilakukan setelah proses pengujian antara video uji dengan 25 video pada query video 3 selesai diuji dan pengujian antara video uji dengan 25 video pada query video 4 selesai diuji. Pengambilan data dilakukan dengan tiga variasi ciri, tiga variasi jumlah iterasi, dan dua variasi jumlah video latih untuk setiap kelas. Variasi ciri terdiri dari ciri HSV, ciri LDP, dan gabungan ciri HSV dan LDP. Variasi jumlah iterasi terdiri dari 50 iterasi, 150 iterasi, dan 200 iterasi.

Variasi jumlah video latih untuk setiap kelas terdiri dari tipe *imbalance* query video dan tipe *balance* query video. Untuk tipe *imbalance* query video terdiri dari 5 *near duplicate video* (NDV) dan 20 *dissimilar video* (DV), sedangkan untuk tipe *balance* query video terdiri dari 12 *near duplicate video* (NDV) dan 13 *dissimilar video* (DV). Sehingga untuk masing – masing tipe

imbalance dan *balance* terdapat 9 kombinasi pengambilan data sebagai berikut :

Tabel 1. Kombinasi Pengambilan Data

Tipe <i>imbalance</i> query video	Tipe <i>balance</i> query video
Ciri HSV dengan 50 iterasi	Ciri HSV dengan 50 iterasi
Ciri HSV dengan 150 iterasi	Ciri HSV dengan 150 iterasi
Ciri HSV dengan 200 iterasi	Ciri HSV dengan 200 iterasi
Ciri LDP dengan 50 iterasi	Ciri LDP dengan 50 iterasi
Ciri LDP dengan 150 iterasi	Ciri LDP dengan 150 iterasi
Ciri LDP dengan 200 iterasi	Ciri LDP dengan 200 iterasi
Ciri (HSV+LDP) dengan 50 iterasi	Ciri (HSV+LDP) dengan 50 iterasi
Ciri (HSV+LDP) dengan 150 iterasi	Ciri (HSV+LDP) dengan 150 iterasi
Ciri (HSV+LDP) dengan 200 iterasi	Ciri (HSV+LDP) dengan 200 iterasi

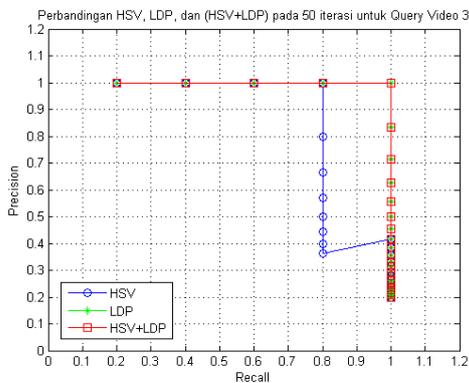
Data pengamatan yang diambil berupa tabel hasil deteksi duplikasi video dengan variasi yang sesuai dengan kombinasi pengambilan data pada Tabel 1. Kolom ke-1 pada tabel artinya urutan rank kemiripan query video terhadap video uji dari query tersebut. Kolom ke-2 pada tabel menunjukkan jumlah perbedaan bit *Hamming*. Kolom ke-3 menunjukkan indeks dari video latih. Kolom ke-4 terdapat istilah *Prec* yang artinya *Precision*. Kolom ke-5 terdapat istilah *Rec* yang artinya *Recall*.

Berdasarkan tabel 2, pada query video 3 dan 4 terlihat bahwa pada rank ke-1 sampai 4 memiliki nilai *precision* sebesar 1. Hal ini menunjukkan sistem berhasil mendeteksi NDV berturut-turut pada rank tersebut. Nilai *recall* pada query video 3 baru bernilai 1 pada rank ke-12 ini menunjukkan sistem baru berhasil mendeteksi semua NDV pada rank ke-12. Sedangkan nilai *recall* pada query video 4 baru bernilai 1 pada rank ke-7 ini menunjukkan sistem baru berhasil mendeteksi semua NDV pada rank ke-7.

Tabel 2. Hasil Deteksi Duplikasi Video untuk *Imbalance* Query Video pada Ciri HSV dengan 50 Iterasi Berdasarkan nilai *precision* dan *recall* yang

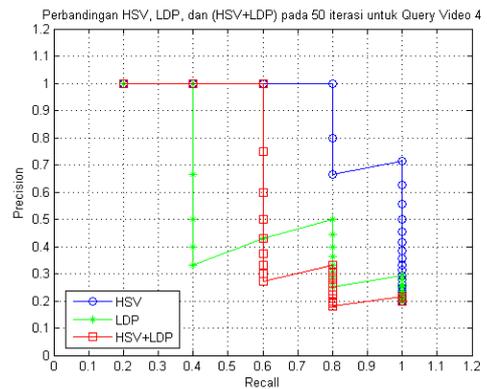
Rank	Query video 3				Query video 4			
	Jarak Hamming (bit)	Indeks video	Prec	Rec	Jarak Hamming (bit)	Indeks Video	Prec	Rec
1	22	5	1	0.200	11	4	1	0.200
2	25	3	1	0.400	16	5	1	0.400
3	26	1	1	0.600	19	2	1	0.600
4	26	2	1	0.800	28	3	1	0.800
5	39	8	0.800	0.800	28	16	0.800	0.800
6	39	11	0.667	0.800	34	25	0.667	0.800
7	40	10	0.571	0.800	35	1	0.714	1
8	40	21	0.500	0.800	37	24	0.625	1
9	42	23	0.444	0.800	38	19	0.556	1
10	43	25	0.400	0.800	39	17	0.500	1
11	46	12	0.364	0.800	40	8	0.455	1
12	48	4	0.417	1	41	15	0.417	1
13	49	7	0.385	1	42	6	0.385	1
14	49	20	0.357	1	43	18	0.357	1
15	50	6	0.333	1	44	21	0.333	1
16	50	9	0.313	1	44	22	0.313	1
17	50	13	0.294	1	46	10	0.294	1
18	50	14	0.278	1	46	23	0.278	1
19	50	15	0.263	1	47	7	0.263	1
20	50	17	0.250	1	47	9	0.250	1
21	50	18	0.238	1	47	13	0.238	1
22	50	19	0.227	1	49	14	0.227	1
23	50	22	0.217	1	52	11	0.217	1
24	50	24	0.208	1	52	12	0.208	1
25	52	16	0.200	1	52	20	0.200	1

didapat dari tabel deteksi duplikasi video kemudian dibuat grafik *precision-recall* untuk semua kombinasi pengambilan data seperti pada contoh Gambar 4 dan Gambar 5.



Gambar 4. Grafik *Precision-Recall* untuk *Imbalance* Query Video 3 pada 50 Iterasi

Pada aplikasi *Near Duplicate Video Retrieval* (NDVR) ini, kinerja sistem diukur dari 2 parameter yaitu *precision* dan *recall*. Pada penelitian ini diinginkan nilai *precision* dan *recall* yang sebaik mungkin, hal ini digambarkan dengan grafik *precision-recall* yang semakin mendekati titik (1,1). Kemudian dihitung nilai *Average Precision* (AP) pada *imbalance* query video dan *balance* query video serta nilai *Mean Average Precision* (MAP). Berikut ini akan ditampilkan rangkuman dari hasil percobaan dan hasil rangkumannya dapat dilihat pada Tabel 3, Tabel 4, Tabel 5, dan Tabel 6.



Gambar 5. Grafik *Precision-Recall* untuk *Imbalance* Query Video 4 pada 50 Iterasi

Tabel 3. Rangkuman Nilai *Average Precision* (AP) untuk setiap *Imbalance* Query Video

Iterasi	HSV		LDP		HSV+LDP	
	Query video 3	Query video 4	Query video 3	Query video 4	Query video 3	Query video 4
50	0.883	0.943	1	0.645	1	0.710
150	0.856	0.925	1	0.648	0.967	0.756
200	0.967	0.848	1	0.648	1	0.777

Tabel 4. Rangkuman Nilai *Average Precision* (AP) untuk setiap *Balance* Query Video

Iterasi	HSV		LDP		HSV+LDP	
	Query video 3	Query video 4	Query video 3	Query video 4	Query video 3	Query video 4
50	0.895	0.962	0.987	0.875	0.971	0.927
150	0.895	0.967	1	0.779	1	0.918
200	0.890	0.960	0.987	0.783	1	0.913

Tabel 5. Rangkuman Nilai *Mean Average Precision* (MAP) untuk setiap *Imbalance* Query Video

Iterasi	HSV	LDP	HSV+LDP
50	0.913	0.822	0.855
150	0.890	0.824	0.861
200	0.907	0.824	0.888

Tabel 6. Rangkuman Nilai *Mean Average Precision* (MAP) untuk setiap *Balance* Query Video

Iterasi	HSV	LDP	HSV+LDP
50	0.928	0.931	0.949
150	0.931	0.890	0.959
200	0.925	0.885	0.956

Jika dilihat dari data Tabel 5 pada ciri HSV didapatkan nilai MAP yang tertinggi untuk semua variasi iterasi. Selain itu jika dilihat pada ciri LDP didapatkan nilai MAP yang terendah untuk semua variasi iterasi. Sedangkan pada penggunaan gabungan ciri HSV dan LDP didapatkan nilai MAP yang lebih baik dibandingkan penggunaan ciri LDP saja namun nilai MAP nya masih belum lebih baik daripada penggunaan ciri HSV saja. Kemudian jika dilihat dari data Tabel 6 pada ciri HSV terdapat sedikit kenaikan nilai MAP dibandingkan pada tabel 5 dengan kenaikan 0.015 untuk 50 iterasi, 0.041 untuk 150 iterasi dan 0.018 untuk 200 iterasi. Namun pada ciri LDP terdapat kenaikan nilai MAP yang cukup besar yakni 0.109 untuk 50 iterasi, 0.066 untuk 150 iterasi dan 0.061 untuk 200 iterasi.

Dan pada penggabungan ciri HSV dan LDP terjadi kenaikan nilai MAP yang sangat besar yakni 0.094 untuk 50 iterasi, 0.098 untuk 150 iterasi dan 0.068 untuk 200 iterasi. Selain itu penggunaan gabungan ciri HSV dan LDP pada kasus *balance* query video menghasilkan nilai MAP yang lebih baik dibandingkan penggunaan ciri HSV saja atau ciri LDP saja.

Dari data nilai MAP pada Tabel 6 terlihat bahwa nilai MAP tertinggi didapatkan untuk 150 iterasi, sedangkan pada 200 iterasi nilai MAP mengalami penurunan nilai yang sangat kecil berkisar 0.005. Hal ini dikarenakan pada awal pelatihan dilakukan inisialisasi nilai bobot dan bias secara acak yang mengakibatkan setiap kali pelatihan nilai MAP yang didapatkan dapat berubah-ubah namun perubahannya sangat kecil.

Berdasarkan data nilai MAP di Tabel 5 dan Tabel 6 terlihat bahwa *balance* query video mempunyai nilai MAP yang lebih baik dibandingkan dengan *imbalance* query video. Hal ini berarti bahwa jumlah video latih untuk setiap kelas sangat mempengaruhi kinerja sistem *Near Duplicate Video Retrieval* (NDVR).

VI. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil pengujian dan pembahasan yang sudah disampaikan, maka dapat ditarik kesimpulan sebagai berikut :

1. Untuk semua variasi iterasi dan variasi ciri, maka nilai MAP pada *balance* query video lebih besar daripada *imbalance* query video.
2. Penggunaan gabungan ekstraksi ciri lokal dan global akan menghasilkan nilai MAP yang lebih baik daripada hanya menggunakan satu ekstraksi ciri (lokal saja atau global saja) untuk jumlah video latih yang sama dari tiap kelas.

B. Saran

Ada beberapa hal yang masih bisa dilakukan untuk menyempurnakan hasil penelitian ini, yaitu:

1. Untuk memperoleh hasil yang lebih optimal, maka pelatihan sistem sebaiknya berdasarkan nilai fungsi objektif yang diinginkan daripada menggunakan jumlah iterasi.
2. Untuk mengurangi kesalahan dalam klasifikasi, maka dapat dicoba dengan menggunakan metode *cost-sensitive learning* agar dapat mengatasi masalah *imbalance* query video.

DAFTAR PUSTAKA

- [1] Hao, Y., T. Mu., R. Hong., M. Wang., N. An. and J.Y. Goulermas.2017. Stochastic Multiview Hashing for Large-Scale Near-Duplicate Video Retrieval. IEEE Trans. Multimedia. Vol 19 No 1 : 1–14.
- [2] Hao, Y., T. Mu., R. Hong, M. Wang, N. An. and J. Y. Goulermas.2017. Unsupervised t-Distributed Video Hashing and Its Deep Hashing Extension. IEEE Trans. Image Processing. Vol 26 No 11 : 5531–5544.
- [3] Patel, B.V. and B.B. Meshram.2012. Content Based Video Retrieval. The International Journal of Multimedia & Its Application (IJMA). Vol 4 No 5 : 77-98.
- [4] Khurana, M.K. and Dr.M.B. Chandak. 2013. Key Frame Extraction Methodology for Video Annotation. IJCET. Vol 4 No. 2 : 221-228.
- [5] Vimalraj, S.S. and R. Porkodi. 2018. A Review on Handling Imbalanced Data. 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT). pp 1-11.
- [6] Jabid, T., Md.H. Kabir. and O. Chae. 2010. Local Directional Pattern (LDP) for Face Recognition. 2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE). pp 329-330.
- [7] Wu, X., A.G. Hauptmann. and C.W. Ngo. 2007. Practical Elimination of Near-Duplicates from Web Video Search. Proceedings of the 15th ACM International Conference on Multimedia. pp 218-227.

- [8] Al-amri, S.S, N.V. Kalyankar. and S.D. Khamitkar. 2010. Image Segmentation By Using Edge Detection. *International Journal on Computer Science and Engineering (IJCSE)*. Vol 2 No 3 : 804-807.
- [9] Huttenlocher, D.P., G.A. Klanderman. and W.J. Rucklidge. 1993. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol 15 No 9 : 850-863.
- [10] Kabir, Md.H., T. Jabid. and O. Chae. 2010. A Local Directional Pattern Variance (LDPv) Based Face Descriptor for Human Facial Expression Recognition. 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance. pp 526-532.
- [11] Plataniotis, K.N. and A.N. Venetsanopoulos. 2000. *Color Image Processing and Applications*. Springer : Verlag.
- [12] Gonzalez, R.C., R.E. Woods. 2008. *Digital Image Processing*. 3rd Edition. New Jersey : Prentice Hall.
- [13] Stallings, W. 2014. *Cryptography and Network Security Principles and Practice*. 6th Edition. Pearson Education, Inc.
- [14] Maaten, L. and G. Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*. pp 2579-2605.
- [15] Hamming, R.W., 1950. Error Detecting and Error Correcting Codes. *Bell Labs Technical Journal*, Vol 29 No 2 : 147-160.
- [16] Powers, D.M.W. 2007. Evaluation : From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001. School of Informatics and Engineering. Flinders University of South Australia.
- [17] Manning, C.D., P. Raghavan., H.Schutze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- [18] Richard Szeliski. 2022. *Computer Vision : Algorithms and Applications*. 2nd Edition. Springer.