

Pengenalan Gambar Tempat Wisata Dengan *Deep Local Feature* Dan *Support Vector Machine*

Angelika Dibijo, Agustinus Noertjahyana, Alvin Nathaniel Tjondrowiguno

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: angelikadibijo@gmail.com, agust@petra.ac.id, alvin.nathaniel@petra.ac.id

ABSTRAK

Pengabdian momen pada suatu tempat biasa dilakukan dengan foto. Akan tetapi, karena banyaknya foto, pengorganisasian foto menjadi sulit. Wisatawan mungkin tidak mengetahui nama tempat wisata yang ia kunjungi, tidak sempat memberi nama foto atau lupa nama tempat foto itu diambil. Pencarian tempat melalui foto secara manual akan membutuhkan waktu yang cenderung lama.

Penelitian ini melakukan uji coba dengan implementasi metode *Deep Local Feature* (DELFF) serta *Support Vector Machine* (SVM) untuk mengenali foto tempat wisata secara otomatis. Metode DELFF merupakan metode yang efektif dalam pengambilan fitur gambar, khususnya gambar tempat. Setelah pengambilan fitur gambar dilakukan, gambar akan dikelompokkan berdasarkan fitur dengan SVM.

Pengujian dilakukan untuk mendapatkan nilai parameter pengambilan fitur dengan DELFF dan klasifikasi dengan SVM agar pengenalan gambar tempat wisata memiliki tingkat akurasi yang tinggi. Untuk 153 *class* gambar, dilakukan DELFF dengan *image threshold* 50 dan *max feature* 1000. Sedangkan klasifikasi menggunakan SVM ber-*kernel* rbf dengan *cost* 10 dan *gamma* 0,01. Dengan menggunakan DELFF dan SVM tersebut didapatkan akurasi dengan data *test* sebesar 0.6178.

Kata Kunci: Pengenalan Gambar, *Deep Local Feature*, *Support Vector Machine*

ABSTRACT

Saving moment at a place usually done with photos. However, due to the large number of photos, organizing photos becomes difficult. A tourist may not know the name of the tourist spot he visited, do not have time to name the photo or forget the name of the place the photo was taken. Manually searching for places names through photos will take a long time.

This research will conduct a trial with the implementation of the Deep Local Feature (DELFF) method and Support Vector Machine (SVM) to recognize photos of tourist attractions automatically. The DELFF method is an effective method for capturing image features, especially place pictures. After capturing image features, the images will be grouped based on features with SVM.

The test is carried out to get the value of the parameter taking features with DELFF and classification with SVM so that the recognition of tourist attractions has a high level of accuracy. For 153 image classes, DELFF is performed with an image threshold of 50 and a max feature of 1000. While the classification uses SVM with kernel rbf with cost 10 and gamma 0.01. By using the DELFF and SVM obtained accuracy with a test data of 0.6178.

Keywords: *Image Recognition, Deep Local Feature, Support Vector Machine*

1. PENDAHULUAN

Pergi mengunjungi tempat wisata adalah salah satu pilihan untuk menghabiskan waktu luang ataupun liburan. Berbagai macam tempat wisata tersebar diseluruh dunia dan tidak jarang wisatawan mengambil foto untuk mengabadikan momen di tempat itu. Akan tetapi, karena banyaknya foto, wisatawan mungkin tidak sempat memberi nama foto dan lupa di mana foto itu diambil. Wisatawan juga mungkin tidak mengetahui nama tempat wisata yang ia kunjungi dan ia foto.

Pencarian tempat melalui foto secara manual akan membutuhkan waktu yang cenderung lama dan wisatawan harus mengingat-ingat kembali tempat yang telah dikunjunginya. Kondisi ini akan tambah membingungkan apabila koleksi foto yang dimiliki banyak. Penelitian ini diharapkan mampu mengatasi masalah tersebut. Pengenalan gambar tempat wisata akan diotomatiskan menggunakan *Deep Local Feature* dan *Support Vector Machine*. Pengenalan gambar ini dilakukan untuk mengetahui dimana tempat wisata itu sehingga membantu wisatawan dalam mengetahui serta mengorganisasikan fotonya.

Gambar yang dikenali harus cukup beragam, maka dataset yang digunakan juga harus memiliki variasi yang banyak. Hal ini akan diatasi dengan penggunaan *Google Landmarks Dataset*. *Google Landmarks Dataset* memiliki tempat wisata yang bervariasi, terkenal dan kurang terkenal, dari seluruh dunia.

Suatu gambar, khususnya tempat, dapat dibedakan melalui fitur yang terdapat pada gambar tersebut. Beberapa metode untuk pencarian fitur ini telah diteliti untuk pengenalan objek ataupun tempat wisata. *Scale Invariant Feature Transform* (SIFT) dapat diterapkan untuk pengenalan tempat dengan mentransformasi data gambar menjadi koordinat berskala tetap yang relatif terhadap *local feature* [5]. Penelitian mengenai pengenalan dengan SIFT juga pernah dilakukan untuk proses pengenalan Batik dengan klasifikasi dengan SVM [1]. Sedangkan ada pula yang meneliti dengan menggunakan SPK-BoW sebagai metode ekstraksi fitur dan *extreme machine learning* [2]. Akan tetapi, seiring dengan berkembangnya dan tumbuhnya popularitas *deep learning*, metode pencarian fitur dengan *deep learning* semakin sering diteliti. Pengenalan gambar dengan SIFT kurang efektif dan menghasilkan akurasi yang cenderung lebih buruk dibandingkan pengenalan yang dilakukan dengan metode berbasis CNN [9].

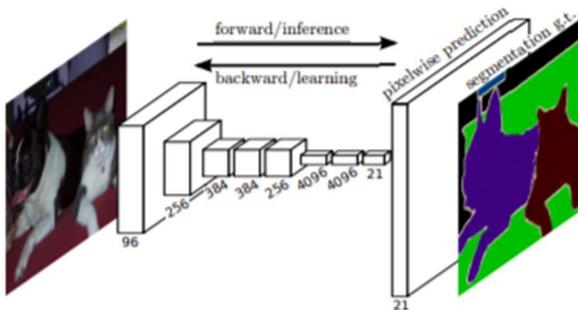
Deep Local Feature adalah metode pengambilan fitur gambar yang berbasis *Convolutional Neural Network*. *Deep Local Feature* menghasilkan pengenalan gambar dengan akurasi tinggi walaupun data gambar sangat banyak dan bervariasi, terutama untuk gambar berupa tempat. Setelah fitur telah ditemukan pada gambar, gambar-

gambar tadi akan diklasifikasikan sesuai fitur yang dimilikinya. Pengklasifikasian gambar melalui fiturnya dilakukan dengan *Support Vector Machine*. Jadi, penelitian ini akan mengenali gambar tempat wisata dengan DELF untuk *feature extraction* dan SVM untuk klasifikasi.

2. DASAR TEORI

2.1 Fully Convolutional Network (FCN)

Fully Convolutional Network (FCN) adalah salah satu metode untuk membagi gambar secara semantic atau kategorikal [4]. Jadi, setiap pixel pada gambar akan diklasifikasi berdasarkan class objeknya seperti pada Gambar 1. Hal ini biasanya dilakukan untuk membuat prediksi secara mendalam. FCN bekerja dengan menerapkan konvolusi seperti pada Convolutional Neural Network (CNN).



Gambar 1 Fully Convolutional Network (FCN) [4]

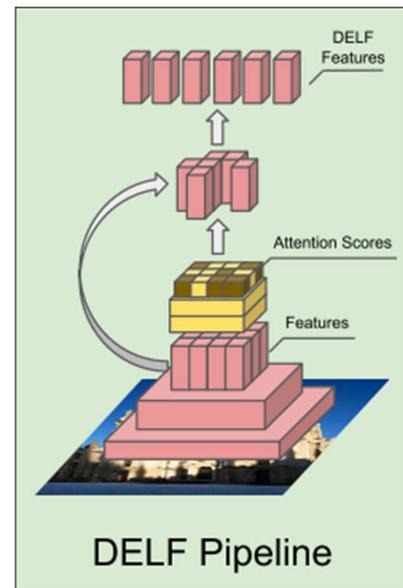
Output pada setiap *layer* adalah array 3 dimensi dari $h \times w \times d$ dimana h dan w merupakan dimensi spasial. Sedangkan d adalah fitur atau dimensi channel. *Receptive fields* adalah lokasi pada *layer* yang lebih tinggi berhubungan dengan lokasi gambar sebelumnya. Jaringan dari konvolusi terdiri dari 3 bagian dasar yaitu konvolusi, *pooling*, dan fungsi aktivasi. Ketiga bagian ini akan dikerjakan untuk bagian gambar.

Ketika proses tadi selesai *receptive fields* akan *overlap* atau bertumpukkan secara signifikan. Proses klasifikasi akan diteruskan agar dapat menghasilkan segmentasi yang sesuai. Output dari *layer-layer* tadi akan digabungkan kembali ke *pixel* semula.

FCN dapat menerapkan *skip architecture* seperti ResNet. ResNet adalah suatu metode untuk melanjutkan koneksi. Tujuannya adalah untuk mengurangi masalah dalam *gradient* dengan menggunakan jalur alternatif bagi *gradient* dan memungkinkan model untuk mempelajari fungsi identitas untuk meyakini *layer* atas akan berfungsi lebih baik daripada *layer* sebelumnya.[1]

2.2 Deep Local Feature

Deep Local Feature adalah cara untuk mengambil fitur pada gambar dengan berbasis pada *CNN*. Metode ini cocok untuk mengambil fitur gambar berskala besar atau banyak. *Deep Local Feature* dianggap cocok untuk melakukan pengambilan fitur pada gambar, khususnya pada gambar tempat. Hal ini dikarenakan DELF telah dipretrained sebelumnya dengan *Google Landmark dataset*. Langkah-langkah dalam DELF dibagi menjadi beberapa bagian seperti dapat dilihat pada Gambar 2, yaitu pengambilan fitur terlokalisasi, pemilihan *keypoint*, dan reduksi dimensional. [6]



Gambar 2 Proses DELFL Keseluruhan [6]

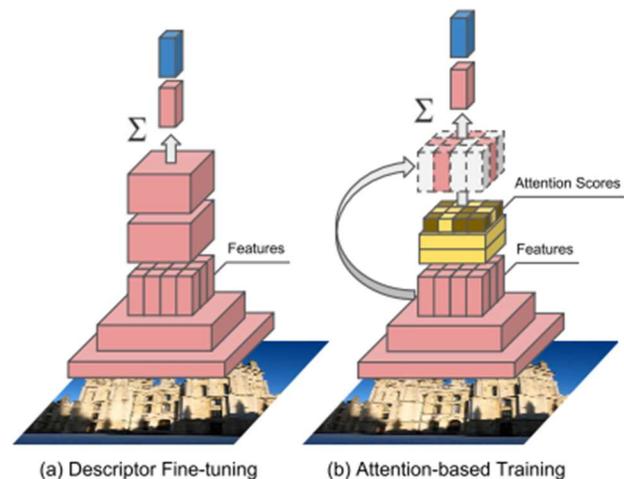
2.2.1 Pengambilan fitur terlokalisasi

Fitur akan di ekstrak dengan mengaplikasikan *fully convolutional network* (FCN) yang diambil dari ResNet50. *Fully Convolutional Network* menggunakan *Convolutional Neural Network* (CNN) untuk mengubah *pixel* gambar menjadi kategori *pixel*. Jadi FCN akan mengambil bagian dari gambar secara semantic.

Piramida gambar dibuat untuk mengatasi perubahan skala. FCN diaplikasikan pada setiap level pada piramida gambar. Hal ini akan memunculkan *feature map* berdasarkan descriptor yang banyak. Fitur kemudian akan dilokalisasi berdasar *receptive fieldsnya*. Lokasi fitur adalah bagian tengah *receptive field*. Menggunakan piramida gambar, fitur yang mendeskripsikan bagian gambar dengan berbagai ukuran bisa didapatkan.

2.2.2 Pemilihan *keypoint*

Pemilihan *keypoint* digunakan untuk menambah akurasi serta efisiensi dari pengklasifikasian gambar. Teknik ini memilih subset dari banyak fitur yang telah terekstraksi (Gambar 3).



Gambar 3 Pemilihan *Keypoint* [6]

2.2.3 Reduksi dimensional

Langkah ini dilakukan untuk mengurangi dimensi fitur yang telah dipilih agar meningkatkan akurasi. Fitur yang telah dipilih akan dilakukan normalisasi dan dimensinya akan dikurangi sampai 40 dengan Principal Component Analysis (PCA). PCA adalah prosedur statistik menggunakan transformasi orthogonal untuk merubah variabel yang berhubungan menjadi sekumpulan variable yang tidak berhubungan secara linear [8]. Setelah itu fitur akan dinormalisasi kembali.

2.3 Support Vector Machine

Support Vector Machine (SVM) akan melakukan klasifikasi dengan menemukan *hyperplane* terbaik untuk memisahkan dua *class* [7]. *Hyperplane* terbaik akan didapatkan apabila *margin* yang dihasilkan maksimum. *Margin* adalah jarak antara *support vector* atau titik terdekat pada satu *class* dengan *hyperplane*.

Permasalahan yang ditemukan untuk klasifikasi dengan SVM tidak selalu dapat dipisahkan secara linear. Maka dari itu, SVM dimodifikasi dengan menggunakan fungsi kernel untuk menyelesaikan masalah klasifikasi data non-linear. Data non linear *separable* tadi akan dipetakan ke ruang vektor dimensi yang lebih tinggi.

Akan tetapi, semakin tinggi dimensi pemetaan data, akan semakin mahal perhitungan dan komputasi *hyperplane*. Untuk mengatasi masalah ini dilakukan *kernel trick*. *Kernel trick* memberikan berbagai kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support vector*, kita hanya cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi non linear Φ . Berbagai jenis fungsi kernel dikenal, seperti pada Gambar 4.

Jenis Kernel	Definisi
Polynomial	$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^p$
Gaussian	$K(\vec{x}_i, \vec{x}_j) = \exp(-\frac{\ \vec{x}_i - \vec{x}_j\ ^2}{2\sigma^2})$
Sigmoid	$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i \cdot \vec{x}_j + \beta)$

Gambar 4 Tabel Kernel yang Umum dipakai

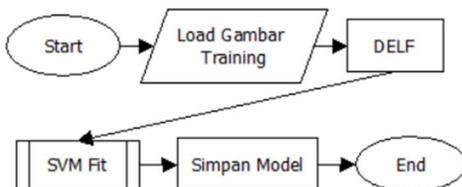
3. DESAIN SISTEM

3.1 Design Implementasi Sistem

Sistem akan meliputi pengumpulan data untuk *training*, *training* dengan fitur hasil ekstraksi DELF pada gambar *training*, pengklasifikasian *input* menggunakan SVM, dan mekanisme *testing*.

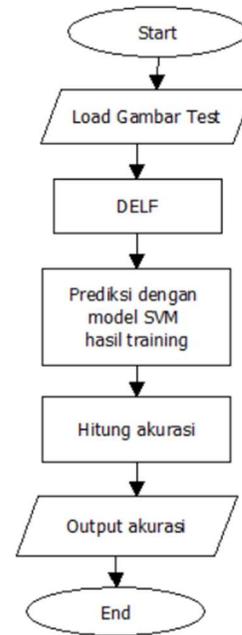
3.1.1 Flowchart Sistem Training

Training seperti pada Gambar 5 akan diawali dengan meload gambar *training*. Setelah itu, setiap gambar *training* akan diambil fiturnya dengan *Deep Local Feature* (DELFF). Tensor *descriptor* hasil dari DELFF akan ditrain dengan *Support Vector Machine* (SVM). Model hasil *training* akan disimpan untuk dipakai dalam *testing* dan pengenalan gambar.



Gambar 5 Flowchart Sistem Training

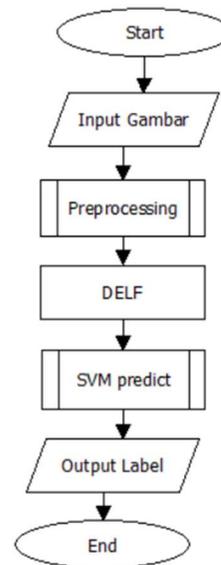
3.1.2 Flowchart Sistem Testing



Gambar 6 Flowchart Sistem Testing

Gambar *testing* akan diload dan dicari fiturnya menggunakan *Deep Local Feature* (DELFF). Setelah itu masing-masing gambar akan diprediksi *class*nya dengan model SVM yang telah dibuat (hasil *training*). Akurasi akan dihitung dengan cara membagi jumlah *class* yang diprediksi dengan benar dengan total data. Gambar sistem *testing* dapat dilihat pada Gambar 6.

3.1.3 Flowchart Sistem Klasifikasi



Gambar 7 Flowchart Sistem Klasifikasi

Sistem klasifikasi meliputi *input* gambar, *preprocessing*, DELFF, dan prediksi dengan SVM seperti pada Gambar 7. Gambar *input* akan diambil fiturnya menggunakan DELFF. Fitur yang berhasil diambil akan diklasifikasikan dengan SVM yang telah ditrain sebelumnya. Program akan menghasilkan *label* nama tempat yang merupakan hasil klasifikasi dari gambar *input*.

4. PENGUJIAN SISTEM

4.1 Dataset yang Dipakai pada Pengujian

Dataset dibagi menjadi 2 folder untuk training dan testing. Folder tersebut bernama "train50" dan "test50". Pemilihan folder tempat wisata dilakukan dengan menghitung jumlah file dalam folder tersebut. Apabila folder awal berisi lebih dari 55 gambar, maka 50 file acak akan diambil untuk dimasukkan ke folder train. Sedangkan 5 file lain akan diambil pula secara acak dari folder awal setelah proses pemilihan folder train dilakukan. Jadi file gambar yang berada pada folder train tidak sama dengan file gambar pada folder test.

Dataset yang dipakai dalam pengujian ini adalah folder train berisi 153 folder gambar tempat wisata yang bernama sesuai id tempat wisata ("0, 1, 2, 3, ..."). Masing-masing folder memiliki 50 file gambar. Folder testing akan berisi folder gambar tempat wisata yang bernama sesuai id tempat wisata ("0, 1, 2, 3, ...") yang masing-masing berisi 5 file gambar.

4.2 DELF

Parameter yang dibutuhkan untuk input pada module DELF adalah image threshold, max feature, image scale, dan image. Pengujian parameter DELF dilakukan untuk menentukan akurasi dan model terbaik pada program ini.

4.2.1 Image Threshold

Parameter *image threshold* pada DELF adalah nilai minimum suatu *descriptor* untuk dijadikan *descriptor* akhir. Apabila nilai *descriptor* tidak sampai pada nilai *image threshold*, *descriptor* akan dibuang. Hal ini dilakukan untuk memilih *descriptor* secara *attention based*. Jadi *descriptor* dengan nilai besar saja yang akan diperhatikan. Pengujian terhadap parameter ini dilakukan dengan *image threshold* 0, 10, 50, 70, 100, 200, 1000.

Tabel 1 Hasil Pengujian *Image Threshold*

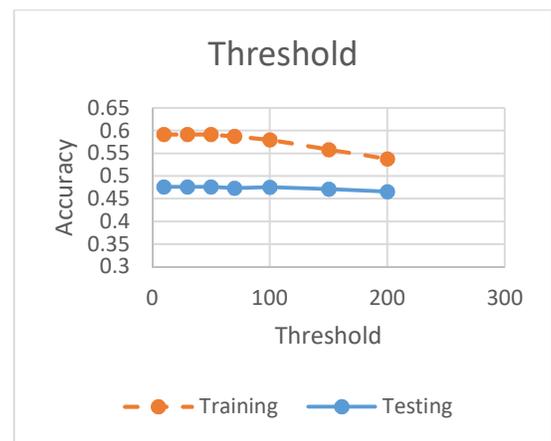
Image Threshold	Panjang Descriptor Maksimum	Gambar tidak memiliki descriptor
10	134	0
30	134	0
50	134	0
70	128	0
100	117	0
150	92	0
200	73	1
1000	2	7624

Tabel 1 merupakan hasil DELF dengan data training (7625 gambar). Ketika threshold mencapai 200, ada 1 gambar yang tidak memiliki descriptor. Sedangkan ketika threshold 1000 hanya 1 gambar memiliki descriptor dan yang lainnya tidak disimpan fiturnya. Semakin besar nilai threshold yang dipakai akan membuat panjang descriptor menurun.

Pengujian selanjutnya adalah dengan akurasi hasil ekstraksi fitur. Threshold yang dipakai adalah 0, 10, 50, 70, 100, 200. Threshold 1000 tidak dipakai karena jumlah gambar terlalu sedikit (hanya ada 1 class).

Tabel 2 Hasil Akurasi Testing dan Training menggunakan parameter DELF, *image threshold*, yang berbeda

Threshold	Testing	Training
0	0.475816993	0.591764706
10	0.475816993	0.591764706
30	0.475816993	0.591764706
50	0.475816993	0.591764706
70	0.473202614	0.58745098
100	0.47513089	0.58
150	0.471204188	0.558300654
200	0.465968586	0.537978821



Gambar 8 Hasil Pengujian Parameter *Image Threshold* terhadap Akurasi

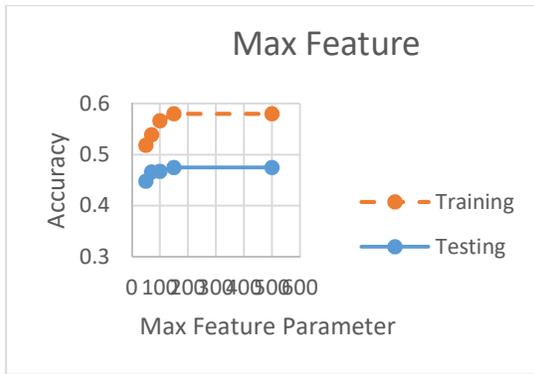
Hasil pengujian pada Tabel 2 dan Gambar 8 menunjukkan bahwa akurasi mulai turun saat threshold di atas 50. Hal ini menunjukkan bahwa semakin besar nilai threshold akan semakin pendek descriptor yang didapatkan dan semakin buruk akurasi pengenalan gambar.

4.2.2 Max Feature

Parameter ini adalah jumlah maksimum fitur yang bisa didapat dari gambar. Jadi, apabila ada fitur yang melebihi panjang *max feature*, fitur akan dipotong sesuai *max feature*. Pengujian terhadap parameter ini dilakukan dengan *max feature* 50, 70, 100, 150, 500, 1000.

Tabel 3 Hasil Akurasi Testing dan Training menggunakan parameter DELF, *max feature*, yang berbeda

Max Feature	Testing	Training
50	0.44836601	0.518562
70	0.46596859	0.539085
100	0.46727749	0.566536
150	0.47513089	0.58
500	0.47513089	0.58
1000	0.47513089	0.58



Gambar 9 Hasil Pengujian Parameter Max Feature terhadap Akurasi

Hasil pengujian pada Tabel 3 dan Gambar 9 menunjukkan bahwa hasil akurasi dengan *max feature* di bawah 150 lebih buruk dibandingkan akurasi pengenalan gambar dengan *max feature* lebih besar dari 150. Akurasi dengan *max feature* dibawah 150 mengalami penurunan ketika *max feature* dikurangi. Hal ini menunjukkan bahwa semakin kecil jumlah fitur yang didapat, semakin kecil pula akurasi pengenalan gambar. Akan tetapi, akurasi pengenalan gambar tidak berubah ketika *max feature* lebih besar dari 150. Akurasi ini tetap dikarenakan parameter ini hanya merupakan jumlah fitur maksimal. Jumlah fitur yang didapat mungkin tidak sampai 150 ataupun batasan *max feature* ini.

4.3 SVM

Pengujian parameter SVM dilakukan untuk menentukan akurasi dan model terbaik pada program ini. Pengujian dilakukan dengan menggunakan SVM *kernel* polynomial dan rbf dengan parameter yang berbeda.

4.3.1 SVM rbf

Pada SVM dengan *kernel* rbf diperlukan parameter berupa *cost* dan *gamma*. Parameter *cost* adalah parameter yang sama dengan parameter *cost* pada *kernel* lainnya, yaitu penalti apabila model salah dalam pengclusteran. Sedangkan *gamma* merupakan parameter yang menunjukkan penyebaran *kernel* dan *decision region* dari *cluster*. Pencarian parameter dilakukan dengan *grid search* menggunakan data *training* sebanyak 153 *class* yang masing-masing memiliki 50 gambar. *Grid search* dilakukan dengan *cross validation* untuk menghindari *overfitting*. Data *testing* diambil dengan pemisahan data secara acak menggunakan *StratifiedShuffleSplit* dari *open source library* Sklearn dengan *n_splits=5*, *test_size=0.2*, *random_state=None*.

Tabel 4 Hasil Grid Search dengan Cost yang berbeda

C/gamma	0.0001	0.001	0.01	0.1	1	10
0.001	0.426	0.433	0.484	0.147	0.172	0.485
0.01	0.407	0.416	0.464	0.136	0.143	0.475
0.1	0.420	0.424	0.471	0.138	0.134	0.484
1	0.419	0.483	0.584	0.365	0.072	0.46
10	0.486	0.543	0.588	0.386	0.086	0.455
100	0.512	0.514	0.584	0.376	0.079	0.451
1000	0.496	0.529	0.595	0.389	0.087	0.468

Hasil pada Tabel 4 menunjukkan bahwa *gamma* terbaik untuk *cost* 0.001 sampai *cost* 0.1 adalah 10. Sedangkan untuk *cost* 1 sampai

1000, *gamma* terbaiknya adalah 0.01. Pengujian kembali dilakukan karena hasil *gamma* terbaik adalah 10 dan 0.01. *Grid Search* kembali dilakukan dengan mengambil *cost* dari 10^{-4} sampai 10^3 dengan *gamma* 0.01.

Hasil *Grid Search* pada Tabel 5 menunjukkan *cost* terbaik yaitu 10 untuk *gamma* 0.01. *Grid Search* terakhir dilakukan kembali untuk mencari pasangan *cost* dan *gamma* terbaik. Hal ini untuk memastikan hasil mana yang lebih baik jika pasangan *cost* dan *gamma* diberikan. *Cost* yang digunakan adalah 0,1 dan 10 dengan *gamma* 0,01 dan 10.

Tabel 5 Hasil Grid Search dengan gamma 0.01

C/gamma	0.01
0.0001	0.478
0.001	0.478
0.1	0.478
1	0.576
10	0.587
100	0.587
1000	0.587

Tabel 6 Hasil Grid Search pasangan cost dan gamma

C/gamma	0.01	10
0.1	0.484	0.488
10	0.596	0.455

Hasil *Grid Search* Tabel 6 menunjukkan parameter terbaik pada data ini adalah dengan *cost* 10 dan *gamma* 0.01.

4.3.2 SVM polynomial

Pada SVM dengan *kernel* polynomial parameter yang dibutuhkan adalah *cost* dan *degree*. *Cost* adalah parameter yang menunjukkan berapa nilai penalti yang diterapkan ke model apabila model salah dalam pembuatan *cluster*. Sedangkan parameter *degree* merupakan tingkat dari dimensi yang digunakan untuk membuat model. Pada pengujian ini, *cost* yang digunakan adalah 10. *Cost* dipilih melalui proses *grid search* yang telah dilakukan pada SVM dengan *kernel* rbf.

Tabel 7 Hasil Grid Search SVM Polinomial

C/degree	3	4	5	6
10	0.46	0.424	0.39	0.356

Hasil *Grid Search* pada Tabel 7 menunjukkan *degree* yang paling baik dengan *cost* 10 adalah *degree* 3.

4.3.3 Pemilihan Kernel

Kernel yang telah dicoba sebelumnya adalah *kernel* polynomial dan rbf. Masing-masing *kernel* telah dicari parameter terbaiknya. Pada sub bab ini akan dilakukan perbandingan antara hasil SVM menggunakan *kernel* rbf yang memiliki *cost* 10 *gamma* 0.01 dengan hasil SVM menggunakan *kernel* polynomial yang memiliki *cost* 10 dan *degree* 3.

Tabel 8 Hasil Akurasi dengan kernel SVM berbeda

rbf	polinomial
0.617801047	0.613874346

Hasil pada Tabel 8 adalah tabel akurasi klasifikasi data *testing* dengan model SVM. Akurasi dengan SVM rbf lebih tinggi dibandingkan dengan SVM ber-*kernel* polynomial. Maka dari itu, parameter yang dipakai pada SVM dengan *dataset* 153 *class* adalah *kernel* rbf dengan *cost* 10 dan *gamma* 0.01.

4.4 Kesimpulan Pengujian

Setelah melakukan pengujian dalam ekstraksi fitur (DELFF) dan klasifikasi (SVM), didapatkan bahwa yang paling mempengaruhi pengenalan gambar dengan DELFF dan SVM adalah parameter pada DELFF, *threshold*. Hal ini dikarenakan *threshold* akan mempengaruhi panjang *descriptor* yang ditemukan dalam gambar. *Descriptor* yang diambil hanya *descriptor* yang lebih besar daripada batas *threshold*. Maka dari itu, ketika *threshold* sangat tinggi, panjang *descriptor* pendek dan kemungkinan suatu gambar tidak memiliki fitur lebih besar.

5. KESIMPULAN

Dari hasil perancangan dan pembuatan sistem dan aplikasi, dapat diambil kesimpulan sebagai berikut:

- Parameter terbaik pada DELFF yaitu dengan *image threshold* 50 dan *max feature* 150.
- Parameter terbaik pada SVM yaitu dengan *kernel* rbf dengan *cost* 10 dan *gamma* 0,01.
- Pengenalan gambar tempat wisata menggunakan DELFF dan SVM dengan data gambar 153 *class* dengan masing-masing *class* berisi 50 gambar serta dengan parameter terbaik menghasilkan akurasi 62%
- Parameter yang paling berpengaruh pada pengenalan tempat wisata menggunakan DELFF dan SVM adalah parameter DELFF, *image threshold*. Hal ini dikarenakan parameter *threshold* memengaruhi panjang *descriptor* yang didapat. *Descriptor* ini merupakan atribut (x) dari model SVM. Jadi semakin panjang *descriptor* gambar didapatkan, atribut pada SVM juga semakin banyak.

6. DAFTAR PUSTAKA

- [1] Azhar, R., Tuwohingide, D., & Kamudi, D. 2015. Batik Image Classification Using SIFT Feature Extraction , Bag of Features and Support Vector Machine. *Procedia - Procedia Computer Science*, 72, 24–30. DOI=<https://doi.org/10.1016/j.procs.2015.12.101>
- [2] Cao, J., Zhao, Y., Lai, X., Ong, M. E. H., Yin, C., Koh, Z. X., & Liu, N. 2015. Landmark recognition with sparse representation classification and extreme learning machine. *Journal of the Franklin Institute*, 352(10), 4528–4545. DOI=<https://doi.org/10.1016/j.jfranklin.2015.07.002>
- [3] He, K., Zhang, X., Ren, S., & Sun, J. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December, 770–778. DOI=<https://doi.org/10.1109/CVPR.2016.90>
- [4] Long, J., Shelhamer, E., & Darrell, T. (n.d.). *Fully Convolutional Networks for Semantic Segmentation*.
- [5] Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints.
- [6] Noh, H., Araujo, A., Sim, J., Weyand, T., & Han, B. 2017. Large-Scale Image Retrieval with Attentive Deep Local Features. *Proceedings of the IEEE International Conference on Computer Vision, 2017-October*, 3476–3485. DOI=<https://doi.org/10.1109/ICCV.2017.374>
- [7] Nugroho, A. S., Witarto, A. B., & Handoko, D. 2003. *Support Vector Machine*.
- [8] Umar, H. B. 2009. Principal Component Analysis (PCA) Dan Aplikasinya Dengan SPSS. *Jurnal Kesehatan Masyarakat*, Maret - September 2009, Vol. 03, No. 2.
- [9] Zheng, L., Yang, Y., & Tian, Q. 2015. *SIFT Meets CNN: A Decade Survey of Instance Retrieval*. 14(8)