

SCENE COMPLETION MENGGUNAKAN TEMPLATE MATCHING DAN POISSON BLENDING

Erick Alfons Lisangan

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar

Alamat email : erick_lisangan@lecturer.uajm.ac.id

ABSTRACT

Image completion, inpainting or hole-filling, is the process to fill or replace an area of image with another image so the replacement can not be detected. The replacement area is the area from the image that is similar than image that has the removing area. The feature of color image that can be used is statistic feature, such as mean, median, and standard deviation. Template matching is used to find the area from the similar image. Then, image blending, like Poisson Blending, is used to equalize gradient of the area from the similar image. The image dataset is obtained from LabelMe and the resolution of images is 256x256. The number of training image is 10 images and 3 images as testing images or query images. The accuracy to produce the scene completion is 66.67%.

Keywords: *scene completion, statistic feature, template matching, image blending*

1. PENDAHULUAN

Image completion merupakan proses mengisi atau mengganti daerah suatu citra dengan citra lain sehingga perubahan pada citra tidak dapat dideteksi. Image completion juga biasa disebut inpainting atau hole-filling. Ide awal dari image completion adalah keinginan untuk menghapus beberapa bagian dari foto-foto lama. Salah satu cara untuk mengisi atau mengganti daerah pada citra dilakukan dengan mencoba menemukan bagian pengganti yang masuk akal untuk mengisi pixel yang hilang dengan menganggap bagian pengganti benar-benar berada di daerah yang digantikan[2].

Bagian pengganti pada daerah yang dihapus membutuhkan citra lain yang mirip dengan bagian yang dihapus sehingga perlu dilakukan pencarian citra yang memiliki kemiripan dengan citra yang memiliki bagian yang dihapus. Proses pencarian citra yang memiliki kemiripan dapat dilakukan dengan membandingkan ciri yang dimiliki oleh citra.

Chary et al (2007) mengatakan bahwa salah satu ciri citra yang dapat digunakan pada citra berwarna dengan menggunakan pendekatan matematika atau statistika. Pendekatan tersebut berupa nilai mean, median, dan standar deviasi untuk masing-masing komponen warna.

Setelah ditemukan citra yang mirip maka kemudian dilakukan proses pencarian bagian citra yang akan digunakan untuk mengisi bagian yang dihapus. Salah satu metode yang dapat dilakukan dengan menggunakan template matching. Template matching melakukan pengecekan setiap bagian dari citra dengan template atau bagian yang dihapus. Bagian citra yang paling mirip dengan template kemudian mengisi bagian yang dihapus dengan terlebih dahulu dilakukan blending untuk menyamakan gradient antara bagian yang diisi dengan daerah sekitarnya.

Pada penelitian scene completion yang dilakukan oleh Hays dan Efros (2007), citra terlebih dahulu dikonversi ke dalam ruang warna L^*a^*b . Setelah diperoleh citra yang mirip kemudian dilakukan proses pencarian bagian citra yang akan diganti dengan menggunakan template matching[2].

Pada penelitian ini, perbandingan antara citra query dengan citra training menggunakan statistic feature untuk masing-masing komponen warna RGB. Setiap citra nantinya akan diwakilkan dengan 9 (ciri) statistik, yaitu mean R,G,B; median R,G,B; dan standar deviasi R,G,B.

2. TINJAUAN PUSTAKA

2.1 Template Matching

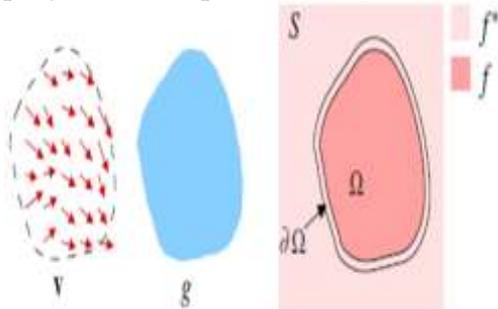
Template matching adalah proses mencari suatu objek pada keseluruhan objek yang berada dalam suatu citra. *Template* dibandingkan dengan keseluruhan objek tersebut dan apabila *template* mirip dengan suatu objek yang belum diketahui maka objek yang ditemukan dapat ditandai sebagai objek yang sama dalam *template*. Pada umumnya ukuran *template* jauh lebih kecil dari ukuran citra^[5].

Perbandingan antara *template* dengan keseluruhan objek pada citra dapat dilakukan dengan menghitung selisih jarak. Perhitungan selisih jarak dapat menggunakan Persamaan 1^[5]. Pada Persamaan 1, $D(m,n)$ menyatakan jarak *template* dengan objek citra, $f(j,k)$ menyatakan citra tempat objek yang akan dibandingkan dengan *template* $T(j,k)$. Nilai m dan n merupakan koordinat bagian citra yang sedang dibandingkan terhadap *template*.

$$D(m,n) = \sum_j \sum_k [f(j,k) - T(j-m, k-n)]^2 \quad (1)$$

2.2 Game Based Learning

Poisson blending adalah teknik untuk melakukan pencampuran (*blending*) 2 (dua) gambar secara halus dan otomatis^[6]. *Poisson blending* merupakan salah satu metode pengolahan citra pada domain *gradient*^[7].



Gambar 1. Ilustrasi *Poisson Blending*

Pada Gambar 1, v merupakan vektor *gradient* dari sebuah daerah dalam citra (nilai *gradient* diperoleh dengan menggunakan operator Laplacian), g merupakan daerah yang dipilih dari citra *source*, f^* fungsi yang diketahui yang terdapat dalam domain S atau citra target, f merupakan fungsi yang akan dicari yang terdapat dalam domain Ω , Ω merupakan daerah dari g yang disisipkan pada citra target, dan $\partial\Omega$ merupakan batas antara

daerah dalam citra *source* dan target^[4]. Tujuan dari *Poisson blending* adalah mencari nilai f yang seminimal mungkin (Persamaan 2) sehingga batas antara citra *source* dan target memiliki *gradient* yang mirip.

$$\min_{f|\Omega} \sum_{(p,q) \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2 \quad (2)$$

dengan $f_p = f^*_p$, untuk semua $p \in \partial\Omega$

Salah satu metode untuk implementasi dari *Poisson blending* adalah dengan menggunakan algoritma Jacobi. Algoritma Jacobi untuk *Poisson blending* adalah sebagai berikut^[6]:

1. Membentuk persamaan matriks $Ax = B$, dimana A merupakan matriks *gradient* dari citra target, x merupakan matriks yang akan dicari atau citra yang akan disisipkan, dan B merupakan matriks yang akan dicapai atau citra *source*
2. Inisialisasi matriks x dengan memberi nilai 0 pada seluruh elemen matriks
3. Hitung perkalian matriks Ax
4. Hitung selisih antara $b-Ax$, yang merupakan ukuran kesalahan antara nilai matriks x saat ini dengan nilai yang seharusnya dibutuhkan
5. Tambahkan selisih $b-Ax$ ke matriks x .
6. Ulangi langkah 3 sampai 5 hingga selisih antara matriks x saat ini dengan $b-Ax$ menjadi sangat kecil

2.3 Statistic Feature

Salah satu ciri citra yang dapat digunakan pada citra berwarna dengan menggunakan pendekatan matematika atau statistika, seperti nilai *mean*, *median*, dan standar deviasi untuk masing-masing komponen warna^[11].

Mean (\bar{x}) adalah nilai rata-rata dari beberapa buah data. Nilai *mean* dapat ditentukan dengan membagi jumlah data dengan banyaknya data (Persamaan 3)^[3].

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (3)$$

Median (Me) menentukan letak tengah data setelah data disusun menurut urutan nilainya. Dalam mencari *median*, dibedakan untuk jumlah data ganjil dan genap (Persamaan 4)^[3].

$$Me = \begin{cases} \frac{x_{n+1}}{2} & \text{jika } n \text{ ganjil} \\ \frac{x_n + x_{n+1}}{2} & \text{jika } n \text{ genap} \end{cases} \quad (4)$$

Standar deviasi (σ) menunjukkan seberapa dekat seluruh himpunan data dengan nilai rata-rata. Semakin kecil nilai sebarannya berarti variasi nilai data semakin sama dan apabila sebarannya bernilai 0, maka nilai semua datanya adalah sama. Semakin besar nilai sebarannya berarti data semakin bervariasi. Standar deviasi dapat dihitung dengan menggunakan Persamaan 5^[3].

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5)$$

3. METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Pada Gambar 3 dapat dilihat bahwa proses *scene completion* terbagi menjadi beberapa tahapan, yaitu *feature extraction* dan *representation*, pengukuran *similarity*, pembentukan *template*, *template matching*, dan *image blending*.

3.1.1 Feature Extraction

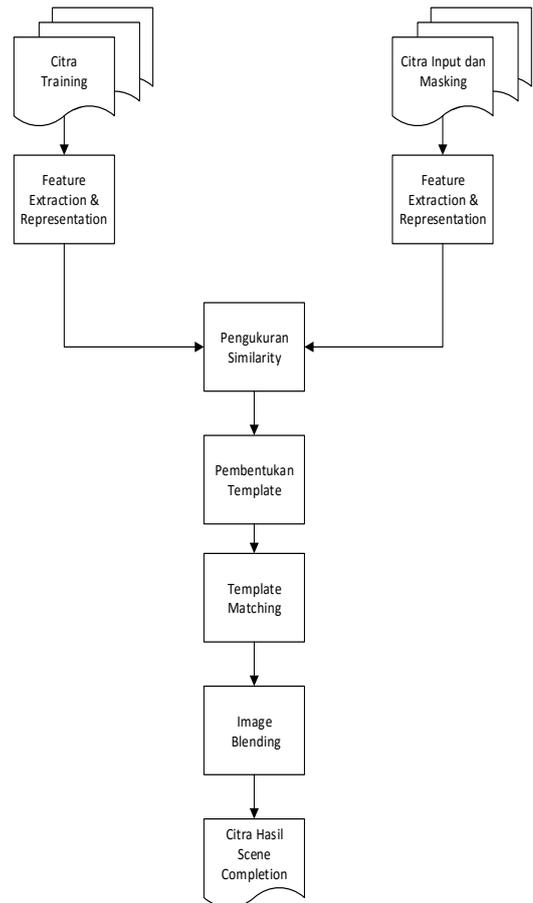
Pada tahap *feature extraction* dan *representation*, citra *training* dan citra *query* diproses untuk memperoleh ciri yang mewakili citra tersebut. Citra yang digunakan pada *scene completion* adalah citra berwarna RGB sehingga ciri yang digunakan juga merupakan ciri warna RGB. Ciri yang digunakan adalah nilai *mean*, *median*, dan standar deviasi untuk masing-masing komponen warna. Jumlah ciri yang mewakili sebuah citra adalah 9 (sembilan) ciri.

3.1.2 Representation

Seluruh ciri dari data citra *training* kemudian disimpan dalam sebuah file teks (Gambar 2). Urutan ciri yang disimpan untuk setiap citra *training* adalah nama citra, nilai *mean R*, *mean G*, *mean B*, *median R*, *median G*, *median B*, standar deviasi R, standar deviasi G, dan standar deviasi B.



Gambar 2. Representasi Ciri Citra *Training*



Gambar 3. Tahapan *Scene Completion*

3.1.3 Pengukuran Similarity

Ciri dari data citra *training* akan dilakukan pengukuran kemiripan untuk mencari citra yang paling mirip dengan citra *query*. Pengukuran kemiripan antara citra *training* (CT) dan citra *query* (CQ) menggunakan *Euclidean Distance*.

$$dist(CT, CQ) = \sqrt{\sum_{i=1}^m (CT_i - CQ_i)^2} \quad (6)$$

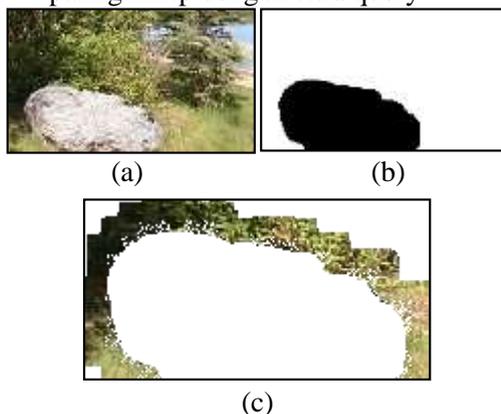
Pada Persamaan 5, i adalah ciri yang dimiliki oleh citra dan m adalah jumlah ciri yang digunakan ($m = 9$). Citra *training* yang memiliki nilai jarak paling kecil merupakan citra *training* yang paling mirip dengan citra *query*.

3.1.4 Pembentukan Template

Sebelum memasuki proses *template matching*, terlebih dahulu dibentuk *template* dari citra *query*. *Template* dibentuk dengan membandingkan antara citra *query* dan *masking*.

Proses pembentukan *template* adalah dengan cara seperti berikut:

1. Binerisasi citra masking, dimana pixel yang berwarna putih diberi nilai 1 dan pixel yang berwarna hitam diberi nilai 0.
2. Mengalikan nilai setiap pixel yang terletak pada posisi yang sama dalam citra query dan masking sehingga nantinya daerah masking pada citra query akan berwarna hitam.
3. Mencari daerah masking dalam citra query berdasarkan koordinat pixel pada citra masking.
4. Mencari nilai warna dari n pixel di sekitar titik-titik masking pada citra query.
5. Setelah diperoleh nilai warna dari pixel di sekitar masking, pembentukan bounding box dari pixel sekitar daerah masking. Bounding box yang diperoleh merupakan template dari citra yang akan di-matching-kan pada citra training yang paling mirip dengan citra query.



Gambar 4. Citra *Query* (a), Citra *Masking* (b), *Template* yang Dibentuk (c)

3.1.5 Template Matching

Setelah pembentukan *template* maka langkah selanjutnya adalah melakukan proses *template matching*. *Template matching* dilakukan dengan pencocokan antara *template* yang telah dibentuk terhadap data *training* yang termirip dengan citra *query*.

3.1.6 Image Blending

Hasil dari *template matching* adalah bagian dari citra *training* yang paling mirip dengan *template*. Bagian dari citra *training* kemudian akan ditempelkan masuk ke dalam citra *query* pada daerah *masking*. Bagian dari citra *training* ketika ditempelkan ke citra *query* dapat memiliki gradient yang berbeda dengan nilai gradient pixel di sekitar *masking*

sehingga perlu dilakukan *blending*. Proses *image blending* yang digunakan adalah *Poisson blending*. Hasil dari *image blending* merupakan citra yang telah melalui proses *scene completion*.

3.2 Data Penelitian

Data citra yang digunakan diperoleh dari *LabelMe* dan memiliki ukuran 256x256. Jumlah data citra yang diambil untuk proses *scene completion* sebanyak 13 (tiga belas) buah dimana 10 (sepuluh) untuk citra *training* dan 3 (tiga) untuk citra *query*.

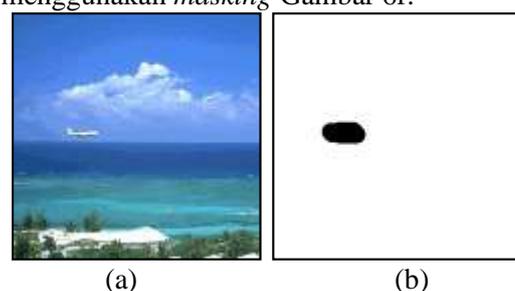


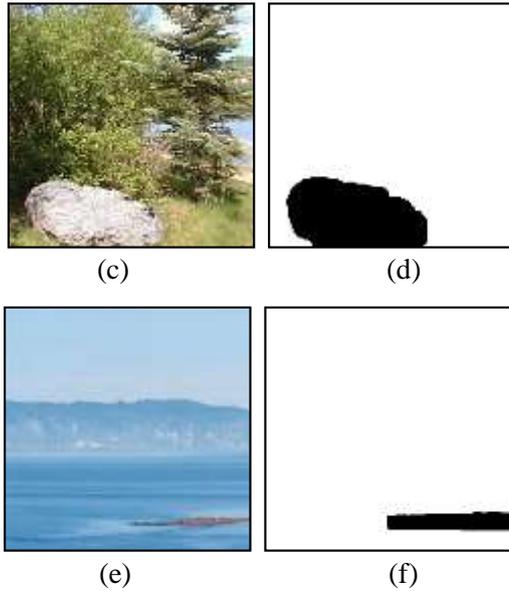
Gambar 5. Citra *Training*

4. HASIL DAN PEMBAHASAN

4.1 Data Masking Citra Query

Pengujian aplikasi *scene completion* yang dirancang dilakukan menggunakan 3 (tiga) buah citra yang berbeda (Gambar 6a, Gambar 6c, dan Gambar 6e). Percobaan I menggunakan Gambar 6a dan *masking* Gambar 6b dengan tujuan akan menghilangkan objek pesawat. Pada Percobaan II menggunakan Gambar 6c dan akan menghilangkan objek batu dengan menggunakan *masking* Gambar 6d. Percobaan III menggunakan Gambar 6e dan akan menghilangkan objek sampah dengan menggunakan *masking* Gambar 6f.





Gambar 6. Citra Query

4.2 Aplikasi Scene Completion

Aplikasi yang digunakan untuk *scene completion* dirancang dengan bahasa pemrograman PHP dan *tools* Matlab. Proses *training*, *feature extraction*, pengukuran *similarity*, pembentukan *template*, dan *template matching* diimplementasikan dengan PHP sedangkan proses *image blending* memanfaatkan Matlab yang melanjutkan hasil pemrosesan dari PHP. Hasil *image blending* dari Matlab kemudian ditampilkan kembali pada *user interface* yang telah dirancang.



Gambar 7. User Interface Aplikasi

4.3 Hasil Percobaan

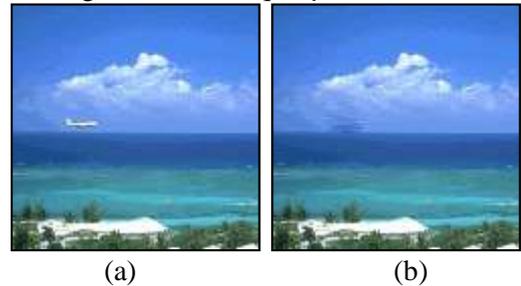
Pada percobaan ini nilai n yang diujikan adalah 10 *pixel* dalam penentuan *template* pada citra *query*. Pemilihan nilai $n=10$ dengan mempertimbangkan bahwa 10 *pixel*

telah cukup untuk membentuk *template* untuk diproses.

4.3.1 Hasil Percobaan I

Pada Percobaan I, dilakukan *scene completion* dengan menghilangkan objek pesawat pada citra *query*. Hasil *scene completion* dapat dilihat pada Gambar 8b dimana objek pesawat diganti dengan bagian laut dan awan pada citra yang memiliki tingkat kemiripan sama dengan citra *query*.

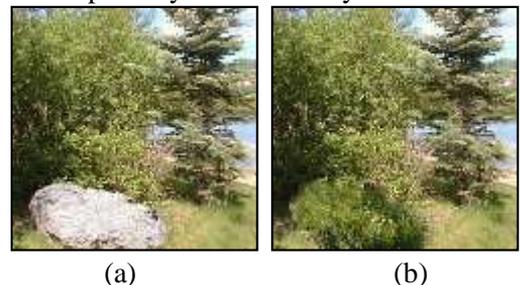
Hasil *scene completion* menunjukkan bahwa bagian citra yang mengisi objek pesawat tidak menutupi wilayah di sekitarnya dimana terdapat garis-garis yang menandakan bahwa objek pesawat dihilangkan dari citra *query*.



Gambar 8. Hasil Percobaan I

4.3.2 Hasil Percobaan II

Pada Percobaan II, dilakukan *scene completion* dengan menghilangkan objek batu pada citra *query*. Hasil *scene completion* dapat dilihat pada Gambar 9b dimana objek batu diganti dengan bagian semak. Citra hasil *scene completion* menunjukkan bahwa semak yang mengisi objek batu secara halus dapat menutupi wilayah di sekitarnya.

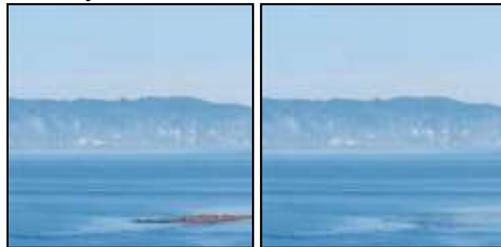


Gambar 9. Hasil Percobaan II

4.3.3 Hasil Percobaan III

Pada Percobaan III, dilakukan *scene completion* dengan menghilangkan objek sampah pada citra *query*. Hasil *scene completion* dapat dilihat pada Gambar 10b dimana objek sampah diganti dengan bagian

laut pada citra yang memiliki tingkat kemiripan sama dengan citra *query*. Hasil *scene completion* menunjukkan bahwa bagian laut yang mengisi objek sampah secara halus dapat menutupi wilayah di sekitarnya.



(a) (b)
Gambar 10. Hasil Percobaan III

5. KESIMPULAN

Berdasarkan hasil *scene completion* dengan menggunakan *template matching* dan *Poisson blending*, maka dapat diperoleh kesimpulan sebagai berikut:

1. Tingkat akurasi untuk menghasilkan citra hasil *scene completion* adalah sebesar 66,67%. Peningkatan akurasi dapat dilakukan dengan penambahan citra training.
2. *Image blending* memiliki peranan penting dalam *scene completion* karena dapat menyamakan gradient antara hasil *template matching* dengan daerah sekitar *masking* pada citra *query*.

Saran yang dapat diberikan pada penelitian selanjutnya adalah pada bagian pencarian citra yang mirip dapat dilakukan dengan membandingkan antara *template* bagian citra yang dihilangkan dengan bagian pada citra training.

6. DAFTAR PUSTAKA

- [1] Chary, R., Lakshmi, D., dan Sunitha, 2012, *Feature Extraction Methods for Color Image -Similarity*, *Advanced Computing: An International Journal (ACIJ)* Vol. 3 No. 2, pp. 147-157.
- [2] Hays, J., dan Efros, A. A., 2007., *Scene Completion Using Millions of Photographs*, *ACM Transactions on Graphics (SIGGRAPH 2007)* Vol. 26 No. 3, pp. 1-7.
- [3] MacMillan, A., dkk, 2007, *Basic statistics: mean, median, average, standard deviation, z-scores, and p-value* (https://controls.engin.umich.edu/wiki/index.php/Basic_statistics:_mean,_median,_average,_standard_deviation,_z-scores,_and_p-value, diakses 11 Desember 2014).
- [4] Perez, P., Gangnet, M., dan Blake, A., 2003, *Poisson Image Editing*, *ACM Transactions on Graphics (SIGGRAPH'03)* Vol. 22 No. 3, pp. 313-318.
- [5] Putra, D., 2010, *Pengolahan Citra Digital*, Penerbit Andi, Yogyakarta.
- [6] Tralie, C., -, *Poisson Image Editing*, (<http://www.ctralie.com/Teaching/PoissonImageEditing>, diakses 10 Desember 2014).
- [7] Yuan, E., 2013, *Poisson Blending*, (<http://eric-yuan.me/poisson-blending>, diakses 8 Desember 2014).