

# Aplikasi Automatic Image Stitching pada Kumpulan Gambar dalam Satu Scene

Edna Ricky Fajar Adi Putra<sup>1</sup>, Liliana<sup>2</sup>, Kartika Gunadi<sup>3</sup>

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

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

E-mail: m26410082@john.petra.ac.id<sup>1</sup>, lilian@petra.ac.id<sup>2</sup>, kgunadi@petra.ac.id<sup>3</sup>

**ABSTRAK :** Semakin populernya panorama image membuat banyak metode pembuatan panorama bermunculan seperti Sweep Panorama maupun penelitian 360° Panorama Stitching. Pada implementasinya, kedua metode ini memiliki kelemahan yaitu hanya dapat menggabungkan gambar dengan pattern horizontal maupun vertical axis saja, sehingga pengambilan hanya dapat dilakukan satu arah, oleh karena itu kemungkinan gagal dalam mendapatkan hasil yang baik maupun kegagalan dalam proses matching sehingga user harus mengulang proses dari awal. Selain itu, terdapat kelemahan terhadap pengambilan gambar pada objek yang bergerak, yang menimbulkan efek ghosting maupun kegagalan dalam stitching.

Untuk mengatasi hal tersebut, dibuatlah aplikasi yang dapat menggabungkan image secara automatic, sehingga user hanya perlu melakukan input gambar dan langsung mendapatkan output dalam bentuk panorama. Aplikasi ini pun dapat mendeteksi lebih dari satu panorama sehingga user tidak perlu memilah – milah gambar ataupun memberi urutan pada gambar terlebih dahulu. Langkah – langkah yang digunakan untuk melakukan proses Automatic Stitching yaitu Extract SIFT Feature untuk mendapatkan keypoint dari gambar input, kemudian proses Feature Matching yang digunakan untuk mencari kesamaan dalam gambar (daerah overlapped). Setelah diketahui relasi dari semua gambar input, dilakukan proses Stitching Sequence untuk mengetahui urutan penggabungan gambar. Langkah terakhir adalah Stitching, yaitu langkah untuk menggabungkan semua gambar input.

Melalui hasil pengujian, stitching image berhasil dilakukan akan tetapi kualitas hasil akhir sangat bergantung pada gambar input. Gambar input yang baik memiliki intensitas yang cukup tinggi dan memiliki banyak object di dalamnya, sehingga memiliki banyak feature yang ditemukan. Tingkat keberhasilan juga dipengaruhi oleh setting aplikasi terhadap jumlah iterasi Best Bin First Search, jumlah Match Verification threshold, dan besaran RANSAC threshold.

**Kata Kunci :** *Panorama, feature matching, pattern matching, stitching image.*

**ABSTRACT :** *The growing popularity of panoramic images, makes many method has been created like Sweep Panorama nor 360° Panorama Stitching. In the implementation, these methods has weakness that can only combine the images with horizontal or vertical axis pattern only, so the pattern just can be done in one way, therefore it is likely to fail in getting good result or failure in the matching process, so the user must repeat the*

*process from the beginning. Other than that, there are drawbacks to shooting at a moving object that can create ghosting and stitching failures.*

*To overcome the problem, an application that can combine images automatically according to the feature matching, so that the user only needs to input images and instantly get the output in the form of panorama. This application is also able to detect more than one panorama so that the user does not need to sort out or make the images sequence. The steps that are used to perform Automatic Stitching process are Extract Features to get all of the keypoint from the input images, after that process Feature Matching carried out to search for similarities in all images (on overlapped area). After having the relation between all images, the stitching sequence process performed to determine the sequence of merging images. The last step is stitching process that is a step to combine all of the input images.*

*Through test result, images stitcing successfully performed but the quality of the final result depends heavily on the input image. Good input image has high intensity and have lot of objects in it, so it has many features that are found. The success rate is also influenced by the setting of the number of Best Bin First Search iterations, the number of Match Verification threshold, and the amount of the RANSAC threshold.*

**Keywords :** *Panorama, feature matching, pattern matching, stitching image.*

## 1. LATAR BELAKANG

Gambar panorama adalah sebuah gambar yang memiliki sudut penglihatan lebih besar dari mata manusia yaitu sekitar 75 - 160 derajat. Gambar panorama memiliki rasio lebih besar atau sama dengan 2 : 1, yang disebut *wide-angle* atau yang lebih dikenal dengan *wide-resolution*. Gambar panorama, pertama dikenalkan pada tahun 1792 oleh Robert Barker dengan melukiskan sebuah panorama London yang ditampilkan dalam permukaan silinder (360 derajat). Lukisan ini memberikan ilusi bahwa orang yang berada didalam silinder akan merasa berada di London karena di kelilingi oleh lukisan panorama tersebut.

Pada abad ke-18, photographer mulai menggunakan teknik Daguerreotype dengan menyusun beberapa gambar untuk menghasilkan tingkat ketajaman tinggi dan *wide-angle*. Teknik Daguerreotype ini sangatlah mahal karena menggunakan tembaga berlapis silver untuk mencetak gambar. Teknik ini dilakukan dengan cara menggabungkan banyak gambar sehingga terbentuk satu gambar yang sangat besar dan tajam.

Terdapat juga catadioptric camera yang menggunakan lensa dan cermin yang melengkung. Cermin tersebut merefleksikan

bayangan dari pemandangan sejauh 360 derajat sudut penglihatan pada lensa sehingga permasalahan pada object yang bergerak dapat diselesaikan oleh metode ini. Kelemahan dari metode ini adalah diperlukan lensa dan cermin melengkung yang membuat tidak praktis untuk dibawa. Hasil dari kamera ini pun harus diproses terlebih dahulu secara digital.

Panoramic Photography adalah sebuah fitur yang dapat dimanfaatkan untuk pengambilan gambar dengan *wide-angle*. Panoramic Photography semakin populer seiring dengan populernya *wide-screen* atau *wide-resolution* pada banyak gadget maupun televisi. Banyak sekali fitur-fitur Panoramic Photography yang ditanamkan dalam camera, gadget, ataupun handphone, sehingga dapat dimanfaatkan oleh orang awam untuk pengambilan gambar panorama.

Fitur Panoramic Photography yang terdapat dalam alat elektronik saat ini biasanya disebut sebagai *Sweep Panorama*. Cara penggunaan *Sweep Panorama* adalah dengan me-rotasi camera sesuai dengan *pattern y-axis* atau *x-axis* (rotasi dari kiri ke kanan atau sebaliknya dan dari atas ke bawah atau sebaliknya, tergantung perintah dari fitur tersebut) dari posisi mata kamera yang harus tetap. Perintah penggunaan dari kiri ke kanan dan lain-lain tersebut digunakan untuk memberikan urutan bagi *frame-frame* yang ditangkap oleh *camera*. Urutan inilah yang nantinya akan digunakan dalam proses *stitching* untuk menggabungkan antar *overlapping image / frame* yang telah diurutkan.

*Pattern* dalam *Sweep Panorama* ini merupakan sebuah keterbatasan karena *user* hanya dapat mengambil gambar secara vertikal saja atau horizontal saja. Keterbatasan ini akan terlihat di saat *user* melakukan pengambilan gambar dari kiri ke kanan, dengan adanya sedikit rotasi ke atas maupun ke bawah (pergerakan tangan manusia), aplikasi akan meminta proses pengambilan diulang dari awal ataupun gambar yang dihasilkan menjadi *blur*.

Berdasarkan permasalahan yang telah dijabarkan diatas maka pada skripsi ini akan dibuat aplikasi untuk menggabungkan beberapa *overlapping image* menjadi sebuah panorama image secara *automatic* tanpa memerlukan *pattern*.

## 2. SIFT

SIFT (Scale Invariant Feature Transform) merupakan sebuah metode yang digunakan untuk mencari *feature* dari sebuah gambar. *Feature* yang dihasilkan oleh algoritma ini tidak terpengaruh oleh adanya *rotasi*, *scaling*, dan perubahan intensitas cahaya. Algoritma ini pertama kali dikemukakan oleh David Lowe pada tahun 1999 dan algoritma ini masih menjadi algoritma terkuat untuk sebagai *feature detection* hingga saat ini [1]. Berikut merupakan langkah – langkah dari algoritma ini :

1. Membuat *Scale Space*
2. Menghitung *Difference of Gaussian*
3. Mencari *Keypoints*
4. Eliminasi *Bad Keypoints*
5. *Orientastion Assignment*
6. *Generate feature*

## 3. FEATURE MATCHING

*Feature – feature* dari metode *SIFT* yang telah didapatkan perlu dicocokkan terhadap *feature – feature* pada gambar lain sehingga dapat ditemukan daerah *stitching* yang tepat. Penghitungan

*feature matching* menggunakan metode *n-dimensional Euclidean Distance* [3]

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (1)$$

Dengan menggunakan rumus *Euclidean Distance*, akan ditemukan nilai *distance* yang menunjukkan sejauh mana perbedaan antar *feature*, dan kemudian dengan menggunakan *threshold*, dapat ditentukan sejauh mana batas sebuah *feature* dapat dikatakan cocok. Untuk beberapa kasus, terdapat beberapa *feature* yang terdapat dalam satu titik atau saling berhimpitan, sehingga kedua *feature* tersebut memiliki *distance* yang relative kecil terhadap sebuah *feature* dari gambar lain. Oleh karena itu, kedua *feature* tersebut perlu diperhitungkan sebagai calon *feature match*. Kasus ini dapat dipecahkan dengan menggunakan *k-Nearest Neighbour*, dimana pencarian dilakukan sampai menemukan sejumlah *k* titik dengan *distance* terbaik.

*Feature matching* pada umumnya dilakukan dengan cara membandingkan semua *feature* dalam satu gambar terhadap semua *feature* dalam gambar lain (*Brute Force Search*) sehingga memakan waktu yang cukup lama. Oleh karena itu, digunakan *k-Dimensional Tree* dengan metode *Best Bin First Search*.

### 3.1 K-dimensional Tree

Setiap *feature* perlu dicocokkan terhadap *feature* lain sehingga dapat ditemukan relasi antar *image* berdasarkan kecocokan *feature-nya*. Setiap *feature* perlu dicocokkan terhadap *k-nearest neighbours* dalam *feature space* menggunakan *k-Dimensional Tree* [3]. *k-Dimensional Tree* merupakan *Binary Tree* yang memiliki 2 cabang pada setiap akarnya. Pemilihan setiap akarnya dilakukan dengan melakukan *sort* terhadap sejumlah data sesuai dengan dimensi pembagi yang digunakan, kemudian *median* dari hasil *sorting* tersebut yang terpilih menjadi akar, sedangkan nilai - nilai yang lebih kecil dari *median* akan menjadi calon *left-root* dan yang lebih besar akan menjadi calon *right-root*. Proses akan diulang kembali terhadap data – data yang tersisa yaitu nilai – nilai calon *left-root* dan calon *right-root*.

Dalam kasus *k-Dimensional Tree* yang digunakan untuk *feature SIFT*, *k* akan bernilai sebesar 128 sesuai jumlah dimensi dari *feature*. *Sorting* menggunakan metode *Quick Sort* dimana dimensi pembagi dimulai dari dimensi pertama dari *feature*. Berikut adalah *pseudo code* dari *k-Dimensional Tree*:

1. Tentukan dimensi pembagi.
2. Lakukan *sorting* terhadap data sesuai dengan dimensi pembagi.
3. Masukkan data ke-median sebagai root.
4. Lakukan 1-3 secara rekursi terhadap data yang bernilai di bawah median sebagai *left-root* dengan dimensi pembagi + 1.
5. Lakukan 1-3 secara rekursi terhadap data yang bernilai diatas media sebagai *right root* dengan dimensi pembagi + 1.

### 3.2 Best Bin First Search

*Best Bin First Search* adalah metode search yang digunakan untuk mencari *approximate solution* untuk *nearest neighbor search* dalam dimensi yang sangat tinggi [3]. Metode ini menggunakan sebuah *priority queue* dengan memasukkan hasil perhitungan terbaik pada kepala *queue* sehingga *node* dengan hasil perhitungan terbaik diperiksa terlebih dahulu. Perhitungan

dilakukan mulai dari root dalam *k-Dimensional Tree*. Iterasi pencarian dalam metode ini ditentukan oleh *user*.

### 3.3 RANSAC

RANSAC (Random Sample Consensus) merupakan metode yang digunakan untuk mencari *Homography Matrix* dan sekaligus berfungsi untuk menghilangkan outliers dari *feature - feature* yang telah ditemukan. Outliers sendiri merupakan *feature* yang memiliki nilai menyimpang dari kebanyakan *feature* lain. *Homography Matrix* merupakan *matrix* transformasi yang berguna untuk memproyeksikan gambar satu pada gambar lain sesuai dengan *feature match* yang ditemukan. Dengan mengalikan gambar dengan *Homography Matrix*, gambar akan mengalami transformasi geometri seperti translasi, rotasi, *scaling*, *skew*, *shear*, dan lain - lain.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2)$$

Dimana  $(u \ v \ 1)^T$  merupakan  $x'$ ,  $(x \ y \ 1)^T$  merupakan  $x$  dan  $H$  merupakan *Homography Matrix* yang berupa *matrix* 3 x 3. *Homography Matrix* inilah yang digunakan untuk menyatukan kedua gambar yang saling berhubungan. Output dari metode ini adalah sebuah *Homography Matrix*. Untuk menyatukan kedua gambar, gambar pertama hanya perlu dikalikan oleh *matrix* tersebut.

### 4. PROJECTIVE TRANSFORMATION

*Projection* merupakan metode yang digunakan untuk memetakan sebuah *image* pada bidang *image* lain. Banyak sekali jenis - jenis dari *projection* seperti *Projective Transformation*, *Cylindrical Projection*, *Spherical Projection* dan lain - lain. Dalam skripsi ini, *projection* akan digunakan sebagai metode *stitching*, yaitu metode untuk menggabungkan *image*.

*Projective Transformation* juga dikenal sebagai *Perspective Transformation* atau *Homography*, yang dioperasikan pada *homogeneous coordinat*. Metode ini merupakan metode yang digunakan untuk menggabungkan *image* [5]. Dalam skripsi ini akan digunakan metode *Projective Transformation* dengan memanfaatkan *Homography Matrix* yang telah ditemukan dalam langkah sebelumnya. Metode ini dilakukan dengan melakukan transformasi pada sebuah gambar sehingga terproyeksikan pada bidang gambar lain. Transformasi dilakukan dengan mengkalikan pixel - pixel pada gambar yang akan diproyeksikan dengan *Homography Matrix*[5].

$$p' = H p$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3)$$

$p'$  merupakan posisi pixel hasil transformasi,  $H$  merupakan *Homography Matrix* yang berupa *matrix* 3 x 3, kemudian  $p$  merupakan posisi pixel - pixel pada gambar original. Pada bidang 2 dimensi, nilai dari  $z = 1$ . Nilai  $x'$  dan  $y'$  dapat ditemukan dengan melakukan normalisasi perkalian *matrix* [4].

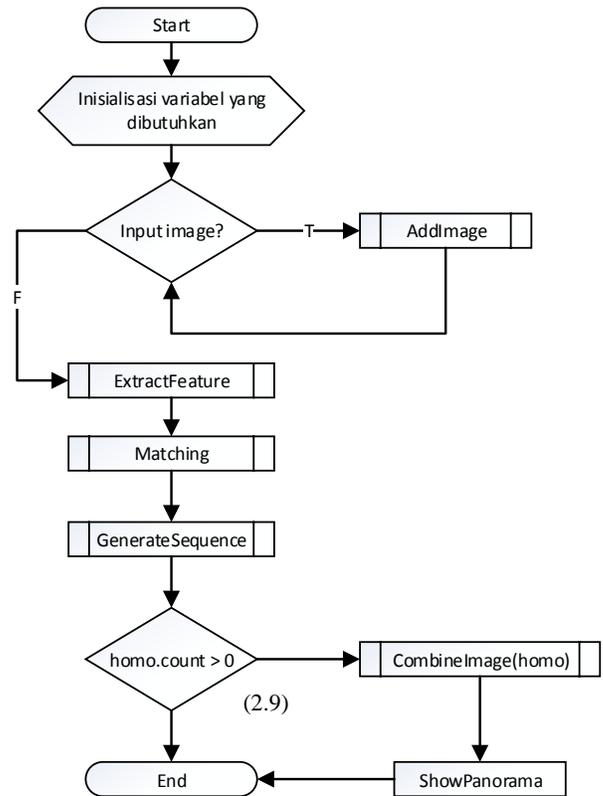
$$x' = \frac{h_{00} \cdot x + h_{01} \cdot y + h_{02}}{h_{20} \cdot x + h_{21} \cdot y + h_{22}} \quad (4)$$

$$y' = \frac{h_{01} \cdot x + h_{11} \cdot y + h_{12}}{h_{20} \cdot x + h_{21} \cdot y + h_{22}} \quad (5)$$

Smoothing menggunakan metode *Weighting Function* [6].

### 5. DESAIN SISTEM

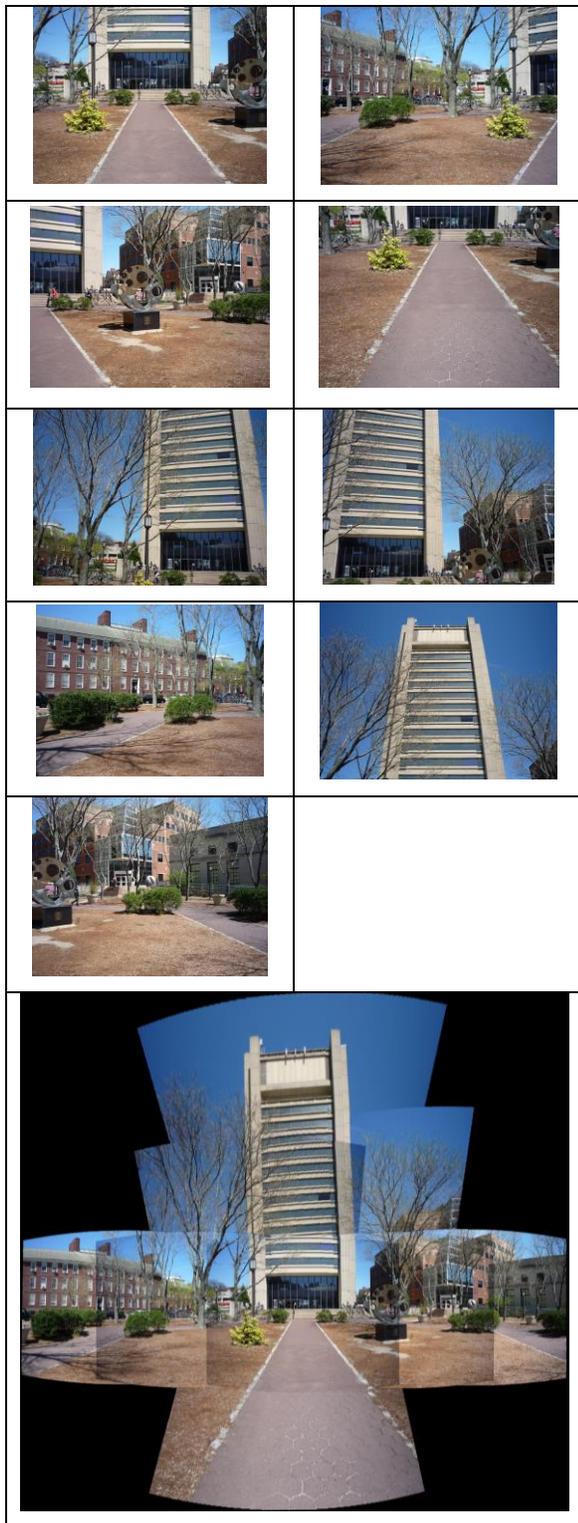
Sistem kerja perangkat lunak untuk membuat *panorama image* dari sekumpulan gambar ini terdapat beberapa tahapan yang perlu dilakukan. Pertama *user* akan diminta untuk melakukan input berupa gambar. Beberapa gambar dapat di-*input*-kan juga secara bersamaan. Selain itu *user* dapat memasukkan gambar secara berulang - ulang. *User* juga dapat melakukan proses *Extract Feature* untuk dapat mengetahui dimana saja *SIFT feature* terdeteksi. Setelah terdapat 2 gambar atau lebih, maka *user* dapat melakukan proses *Matching*. Proses *Matching* akan menghasilkan sebuah tabel yang menunjukkan relasi kecocokan antar *image*. Proses dilanjutkan dengan *Generate Sequence* untuk mengetahui berapa jumlah *panorama* yang dapat terbentuk dan urutan penyambungan gambar. Langkah terakhir adalah *Image Stitching* untuk menggabungkan gambar berdasarkan relasi yang telah ditemukan. Sampai proses *Image Stitching* ini, *user* akan mendapatkan satu (atau lebih) *panorama image* yang merupakan hasil *stitching* beberapa gambar *input*. Flowchart desain sistem dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Desain Sistem

### 6. PENGUJIAN

Pengujian dilakukan dengan menggabungkan sembilan *image* yang diambil dengan pengaruh rotasi, translasi, *skew*, dan sedikit perubahan intensitas. Gambar *input image* dan hasil pengujian dapat dilihat pada Gambar 2.



Gambar 2. Input Image dan Hasil Stitching

Hasil dari pengujian berupa gambar kasar yang perlu diolah lagi sehingga menghasilkan hasil yang lebih baik.

## 7. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut :

- *SIFT* merupakan metode pencarian *feature* dari sebuah gambar yang sangat kuat untuk digunakan dalam penggabungan gambar karena *feature* dari *SIFT* tidak terpengaruh oleh *scaling* dan *rotation*. *SIFT feature* sendiri memiliki kekurangan yaitu *SIFT* tidak kuat dalam menghadapi permasalahan intensitas yang terlalu gelap karena penentuan *feature* dari *SIFT* itu sendiri ditentukan dari nilai *gradient pixel* yang jika warna dari *pixel* tersebut semakin gelap maka *gradient* akan semakin kecil, tetapi untuk intensitas yang tidak terlalu gelap, *SIFT* masih dapat menangannya dengan baik.
- Proses dalam menemukan pasangan – pasangan gambar dapat dijalankan dengan baik. Semua pasangan dapat ditemukan dengan pengaturan *threshold* dan jumlah iterasi *Best Bin First Search* yang tepat. Semakin tinggi iterasi *Best Bin First Search* maka akan semakin tinggi pula kemungkinan untuk ditemukan pasangan *feature match* sehingga jumlah inliers dapat melewati batas *threshold*. Sebaliknya semakin tinggi iterasi *threshold*, maka semakin teliti pula aplikasi untuk melakukan pemeriksaan terhadap *feature match* sehingga kualitas gambar hasil yang didapatkan akan semakin akurat tetapi kemungkinan gambar untuk dapat digabungkan akan semakin kecil.
- Terdapat beberapa faktor yang dapat menyebabkan kesalahan dalam proses stitching. Pengambilan gambar yang kurang sesuai dengan adanya pergerakan mata camera sehingga mengakibatkan banyaknya efek – efek yang tidak bisa teratasi seperti *parallax* yang terlalu tinggi. Kemudian adanya pergerakan pada object yang diambil yang dapat menyebabkan efek *ghosting* maupun gagalnya *feature matching*. Faktor yang terdapat dalam program adalah *threshold Verification Match* dan jumlah iterasi untuk menemukan *feature match*.
- Tingkat keberhasilan dalam aplikasi ini tergantung terhadap *input* dari *user*. Gambar *input* yang semakin jelas akan menghasilkan *Extract Feature* yang baik karena memiliki banyak *SIFT feature*. Gambar yang gelap akan semakin sulit untuk digabungkan sebab *SIFT feature* tidak kuat untuk menghadapi gambar dengan intensitas rendah. Kemudian kemampuan aplikasi dalam melakukan *stitching* dipengaruhi juga oleh *setting* terhadap iterasi *Best Bin First Search*, dan jumlah *threshold*.

## 8. REFERENSI

- [1] Brown, M. & Lowe, D. G. (2007, August). *Automatic Panoramic Image Stitching using Invariant Features*. International Journal of Computer Vision, vol. 74, pp. 59-73.
- [2] Burt, P. J. & Adelson, E. H. (1983, October). *A Multiresolution Spline With Application to Image Mosaics*. ACM Transaction on Graphics, Vol 2 No. 4, pp. 217-236.
- [3] J, Beis and D. Lowe. *Shape Indexing Using Approximate Nearest Neighbour Search In High Dimensional Spaces*. In Proceedings of the International Conference on Computer Vision (ICCV03), volume 2, pages 1218-1225, Nice, October 2003.

[4] Lowe, D. G. (2004, November). *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, pp. 91-110.

[5] Szeliski, R. & Shum, H. Y. (1997, August). *Creating Full View Panoramic Image Mosaics and Environment Maps*.

Paper presented at Computer Graphics (SIGGRAPH'97 Proceedings), New York, USA.

[6] Szeliski, Richard. (2006, Desember). *Image Allignment and Stitching : A Tutorial*. One Microsoft Way, Microsoft Corporation