

**PEMBENTUKAN VIRTUAL FOTOGRAFI BERBASIS IMAGE
MOSAICING DENGAN METODE TRANSFORMASI GEOMETRIK**

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Pada Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia



Disusun Oleh:

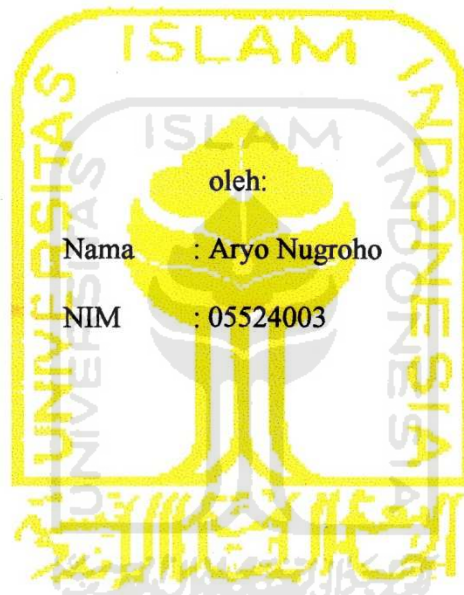
Nama : Aryo Nugroho

No. Mahasiswa : 05524003

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

LEMBAR PENGESAHAN PEMBIMBING
PEMBENTUKAN VIRTUAL FOTOGRAFI BERBASIS IMAGE
MOSAICING DENGAN METODE TRANSFORMASI
GEOMETRIK
TUGAS AKHIR



Pembimbing I

Dwi Ana Ratna Wati S.T., M.Eng.

Pembimbing II

Tito Yuwono ,ST.,MSc.

LEMBAR PENGESAHAN PENGUJI
PEMBENTUKAN VIRTUAL FOTOGRAFI BERBASIS IMAGE
MOSAICING DENGAN METODE TRANSFORMASI
GEOMETRIK

TUGAS AKHIR

oleh:

Nama : Aryo Nugroho

NIM : 05524003

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Teknik Elektro Fakultas Teknologi Industri Universitas Islam
Indonesia

Yogyakarta, _____

Tim Penguji,

Dwi Ana Ratna Wati, ST., M.Eng

Ketua

Wahyudi Budi Pramono, ST., M.Eng

Anggota I

Ir. Hj. Budi Astuti, MT.

Anggota II



Mengetahui,

Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri
Universitas Islam Indonesia



Tito Yuwono ,ST.,MSc.

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

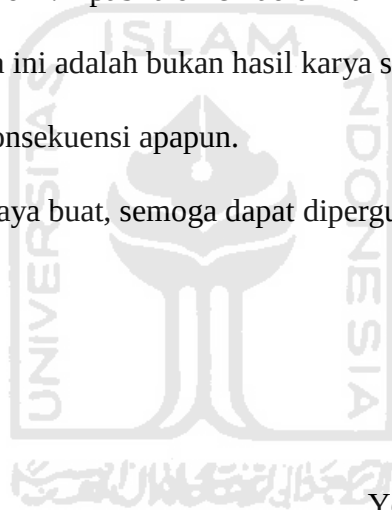
Saya yang bertandatangan di bawah ini,

Nama : Aryo Nugroho

NIM : 05524003

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.



Yogyakarta, 8 januari 2011

Aryo Nugroho

HALAMAN PERSEMBAHAN

Kupersembahkan tulisan ini untuk Bunda



MOTTO

”Dan, orang-orang yang mencurahkan kemampuannya semata-mata karena

Kami, niscaya Kami tunjukkan jalan Kami”

(Terjemah Qur'an Surat Al Ankabut [29]:69)



ABSTRAK

Image mosaic merupakan teknik untuk menggabungkan sekumpulan sekuensial image yang diambil dari kamera dengan pusat proyeksi yang sama. Salah satu kontribusi penting dari implementasi teknik ini banyak dimanfaatkan dalam bidang pemetaan tematik, peta udara dan pengambilan objek geografis lainnya terutama ketika pengambilan suatu objek tidak bisa dilakukan dalam satu kali capture. Selain menggunakan lensa khusus, panorama penuh 360⁰ juga dapat dibuat dengan menggunakan kamera digital lensa biasa. Proses yang dilakukan adalah dengan mengambil 8 gambar panorama di setiap sudut tertentu dan menggabungkannya menjadi sebuah panorama utuh. Adapun cara penggabungan 2 image yang adalah dengan cara pergeseran antar image sehingga dirasa sudah tidak ada lagi bagian dari image yang berulang dengan bantuan Transformasi Geometri. Pengambilan gambar benda bergerak didapatkan hasil yang tidak beraturan, karena pada setiap pengambilan gambar, benda akan berpindah tempat. Sehingga tidak mendapatkan kecocokan gambar. Metode pengambilan gambar yang tepat untuk image mosaic ini adalah gambar diam dan hanya dalam satu sudut saja.



KATA PENGANTAR



Segala puji bagi Allah SWT, pemberi rahmat atas ciptaanNya, pemilik hidayah hingga ditunjukinya orang-orang yang menempuh jalan kebenaran sekaligus pemilik kesesatan hingga disesatkan orang-orang yang menempuh jalan aniaya. Shalawat dan salam atas junjungan Nabi Muhammad SAW, keluarga sahabat dan pengikut-pengikutnya, semoga kita termasuk orang-orang yang beruntung dengan mendapatkan syafaatnya karena ittiba' kita terhadap beliau. Amin.

Alhamdulillah berkat dukungan, doa, dan segala sesuatu yang diberikan oleh semua pihak, telah selesai penyusunan Tugas Akhir yang ditujukan untuk memperoleh gelar akademik Strata Satu di Jurusan Teknik Elektro Universitas Islam Indonesia setelah mengalami penantian yang cukup panjang. Semoga Tugas Akhir ini dapat berkontribusi positif bagi siapa saja yang membutuhkan, khususnya bagi Jurusan Teknik Elektro UII.

Ucapan terimakasih kami ucapkan kepada segenap pihak yang secara langsung maupun tidak langsung telah membantu serta memudahkan proses terselesaikannya Tugas Akhir ini. Semoga Allah SWT memberikan balasan yang sebaik-baiknya untuk semua. Ucapan terimakasih saya tujukan kepada:

1. Kedua orang tua Ibu Praptiningsih Idi Wuryani, kangmas Ari Sujarwo ,ST, C.Msc . atas segala limpahan doa dan dukungan.

2. Rekan-rekan di Jurusan Teknik elektro 2005, teman-teman Angkatan Muhammadiyah Kotagede.
3. Special thanks to Lintang
4. Yang saya hormati Bpk Ir. Gumbolo Hadi Susanto, M.Sc.selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Ketua Jurusan Teknik Elektro FTI UII Bpk Tito Yuwono , ST.,Msc.
6. Dwi Ana Ratna Wati S.T., M.Eng._. selaku dosen pembimbing I, serta Bapak Tito Yuwono ,ST., MSc. selaku dosen pembimbing II dalam penyusunan Tugas Akhir ini.
7. Serta semua pihak yang tidak dapat saya sebutkan satu persatu, yang telah membantu memberikan dukungan hingga terselesaikannya Tugas Akhir ini.

Wassalamualaykum Warahmatullah Wabarakatuh

Yogyakarta , 8 Januari 2011

Hormat Saya

Aryo Nugroho

DAFTAR ISI

	Halaman
Halaman Judul	i
Lembar Pengesahan Pembimbing	ii
Lembar Pengesahan Penguji	iii
Lembar Pernyataan	iv
Ucapan Terima Kasih	v
Abstrak	vii
Kata Pengantar	viii
Daftar Isi	x
Daftar Gambar	xiii
Daftar Tabel	xvi
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat Penelitian	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	5

BAB 2 STUDI PUSTAKA

2.1 Pengantar Teknik <i>Mosaicing</i>	7
2.2 Image <i>Processing</i>	9
2.3 Ciri Suatu Gambar	10
2.4 Feature <i>Matching</i>	13
2.5 Image <i>Mosaicing</i>	14
2.6 Transformasi Geometrik	16
2.7 <i>Blending image</i>	19

BAB 3 METODOLOGI

3.1 Gambaran Umum	20
3.2 Deskripsi sistem	20
3.3 Perancangan Sistem <i>Virtual</i> Fotografi	24
3.3 Perancangan Antar muka Sistem	27

BAB 4 DATA DAN ANALISA

4.1 <i>Preprocessing</i>	46
4.2 Hasil pengujian <i>Virtual</i> Fotografi Berbasis <i>Image Mosaicing</i> Dengan Metode Transformasi Geometrik	57
4.3 <i>Cropping</i> Gambar	67
4.4 Hasil Akhir	68

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan	81
5.2 Saran	82

DAFTAR PUSTAKA	83
-----------------------	----

LAMPIRAN	84
-----------------	----



DAFTAR GAMBAR

Gambar 2.1 Blok Diagram Pengolahan Citra	9
Gambar 2.2 RGB 24-bit <i>color cubs</i>	11
Gambar 2.3 Diagram Kromasitas	12
Gambar 2.4 Penerapan transformasi geometrik secara bertahap pada daerah tumpang tindih kedua citra yang digabungkan	17
Gambar 2.5 Konstruksi <i>Image Dinamis</i>	19
Gambar 3.1 Lenovo 3000 G230	21
Gambar 3.2 Kamear dSLR Nikon D300s	22
Gambar 3.3 Kamera Digital Panasonic Lumix F3	23
Gambar 3.4 Tripod Velbon	23
Gambar 3.5 Blok diagram virtual fotografi berbasis <i>image mosaicing</i>	26
Gambar 3.6 Interface menu utama virtual fotografi	27
Gambar 3.7 Interface masukan image 1	28
Gambar 3.8 Interface masukan image 2	28
Gambar 3.9 Interface masukan image 3	29
Gambar 3.10 Interface masukan image 4	29
Gambar 3.11 Interface masukan image 5	30
Gambar 3.12 Interface masukan image 6	30
Gambar 3.13 Interface masukan image 7	31
Gambar 3.14 Interface masukan image 8	31
Gambar 3.15 Figure proses	32

Gambar 3.16 Figure Help	32
Gambar 4.1 Diafragma	47
Gambar 4.2 Focal Length atau jarak fokus	51
Gambar 4.3 <i>Depth of field</i>	54
Gambar 4.4 Proses pengecilan gambar	55
Gambar 4.5 Proses pemberian kualitas gambar	56
Gambar 4.6 Gambar sebelum melalui proses <i>image rezising</i>	56
Gambar 4.7 Gambar setelah melalui proses <i>image rezising</i>	57
Gambar 4.8 Figure image mosaicing figure 1 step 1	58
Gambar 4.9 Figure hasil image mosaicing figure 1 step 1	59
Gambar 4.10 Figure image mosaicing figure 3 step 1	60
Gambar 4.11 Figure hasil image mosaicing figure 4 step 1	60
Gambar 4.12 Figure image mosaicing figure 5 step 1	61
Gambar 4.13 Figure hasil image mosaicing figure 6 step 1	61
Gambar 4.14 Figure image mosaicing figure 7 step 1	63
Gambar 4.15 Figure hasil image mosaicing figure 8 step 1	63
Gambar 4.16 Figure image mosaicing figure 9 step 2	64
Gambar 4.17 Figure hasil image mosaicing figure 10 step 2	64
Gambar 4.18 Figure image mosaicing figure 11 step 2	65
Gambar 4.19 Figure hasil image mosaicing figure 12 step 2	65
Gambar 4.20 Figure image mosaicing figure 13 step 3	66
Gambar 4.21 Figure hasil image mosaicing figure 14 step 3	66
Gambar 4.22 contoh hasil cropping	67

Gambar 4.22 Contoh hasil akhir image mosaicing di perumahan prumpung	68
Gambar 4.23 Contoh hasil akhir image mosaicing di Kampus FTI menggunakan lensa 18 mm	69
Gambar 4.24 Contoh hasil akhir image mosaicing di Masjid Ulil Albab Bagian Dalam	70
Gambar 4.25 Contoh hasil akhir image mosaicing di Masjid Ulil Albab Bagian Luar	71
Gambar 4.26 Contoh hasil akhir image mosaicing di Gedung FIAI	72
Gambar 4.27 Contoh hasil akhir image mosaicing di Gedung FTI	73
Gambar 4.28 Contoh hasil akhir image mosaicing di Gedung Sarjito	74
Gambar 4.29 Contoh hasil akhir image mosaicing di Gedung GOR UII	75
Gambar 4.30 Contoh hasil akhir image mosaicing di Gedung MIPA	76
Gambar 4.31 Contoh hasil Image Mosaicing menggunakan lensa nikkor 50mm di masjid mataram Kotagede	77
Gambar 4.32 Contoh hasil Image Mosaicing menggunakan lensa nikkor 50mm di candi Boko Prambanan	78
Gambar 4.33 Contoh hasil gambar image mosaicing dengan gambar bergerak	79
Gambar 4.34 Contoh gambar image mosaicing dengan pengambilan bergerak	80

Daftar tabel

Tabel 2.1	Tabel Perbedaan Image Processing, Computer Vision, Computer Graphics	10
-----------	---	----



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Panorama merupakan pemandangan yang dapat dilihat dengan sudut pandang luas. Panorama ini telah dimanfaatkan oleh banyak kalangan baik untuk penelitian maupun sebagai bagian dari fotografi.

Dalam dunia fotografi, panorama yang diperoleh merupakan hasil dari pengambilan gambar menggunakan kamera tunggal, dimana sudut pandang yang terekam hanya sebatas lebar lensa kamera. Perkembangan fotografi telah menghasilkan sebuah lensa yang dapat digunakan untuk mengambil gambar panorama 360^0 , sehingga diperoleh panorama penuh. Lensa ini secara otomatis dapat merekam seluruh sisi panorama sehingga mampu menampilkan panorama penuh yang ada di hadapannya.

Selain menggunakan lensa khusus, panorama penuh 360^0 juga dapat dibuat dengan menggunakan kamera digital lensa biasa. Proses yang dilakukan adalah dengan mengambil gambar panorama di setiap sudut tertentu dan menggabungkannya menjadi sebuah panorama utuh. Proses ini dapat dilakukan secara manual, dimana besar perubahan sudut pengambilan gambar harus konstan.

Virtual photography yang terbentuk dari penggabungan beberapa citra ini dapat ditampilkan secara lebih menarik agar terlihat seperti aslinya. Proses yang dilakukan adalah dengan membangun sebuah aplikasi, dimana aplikasi

ini akan merepresentasikan keadaan yang sesungguhnya. Sudut pandang virtual fotografi akan ditampilkan sesuai dengan keinginan pengamat, sehingga pengamat seolah sedang berada di tempat panorama yang sebenarnya.

Tampilan dari virtual photography sangat dipengaruhi oleh kualitas pengambilan gambar. Dalam proses ini, dibutuhkan ketelitian dalam pergeseran sudut dan kemiringan kamera serta ketepatan besar sudut yang digunakan agar dalam implementasinya nanti, gambar yang dihasilkan mendekati aslinya, dalam arti tidak terlihat cembung akibat dari efek pengambilan sudut yang terlalu besar atau sudut yang terlalu kecil.

Pada tugas akhir ini akan dibahas mengenai pembangunan perangkat lunak untuk menampilkan panorama 90° , 180° atau lebih agar terlihat seperti tampilan aslinya serta menentukan besar sudut yang ideal untuk membuat sebuah virtual photography.

1.2 Rumusan Masalah

Berdasarkan pada latar belakang di atas, permasalahan yang akan diuraikan dan diteliti adalah:

1. Bagaimana cara menangkap citra dalam setiap sudut pandang tertentu agar diperoleh image yang dapat digabungkan?
2. Bagaimana cara menggabungkan citra yang telah ditangkap di setiap sudut pandang agar panorama 180° atau lebih.

3. Bagaimana membangun aplikasi yang dapat menampilkan panorama penuh 180^0 atau lebih berdasarkan virtual photography yang ada?

1.3 Batasan Masalah

Adapun batasan masalah tugas akhir ini adalah sebagai berikut :

1. Gambar yang diambil adalah panorama diam, tidak ada objek yang bergerak.
2. Kamera yang digunakan untuk pengambilan gambar adalah kamera digital manual tanpa lensa khusus untuk panorama.

1.4 Tujuan Penelitian

Tujuan dari proyek akhir ini adalah:

- Membangun sebuah aplikasi virtual gambar yang dibangun dari kumpulan potongan image dengan memanfaatkan teknik mosaik dan mencari kongruensi atau kesesuaian antar potongan dengan transformasi geometrik.

1.5 Manfaat Penelitian

Manfaat dari tugas akhir ini adalah untuk memberikan solusi kepada fotografer untuk mendapatkan sudut lebar dengan biaya murah, karena sudut lebar harus menggunakan lensa sudut pandang lebar yang harganya sering tidak terjangkau oleh fotografer.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam menyelesaikan masalah di atas adalah dengan menggunakan langkah-langkah berikut:

a. Studi Literatur

Teknik yang digunakan untuk mendukung pembentukan virtual fotografi adalah:

- Mempelajari referensi dan sumber-sumber yang berhubungan dengan fotografi, dalam hal ini adalah cara pengambilan gambar agar didapat kualitas gambar yang maksimal.
- Mempelajari referensi yang berhubungan dengan cara penentuan sudut pandang objek dan algoritma penggabungan image.
- Mempelajari referensi yang berhubungan dengan cara pembentukan panorama digital dengan tampilan 360⁰.

b. Pengembangan Perangkat Lunak

Pengembangan perangkat lunak dilakukan secara bertahap, yaitu :

- Mengidentifikasi kebutuhan

Pada Tugas Akhir ini identifikasi kebutuhan yang dilakukan adalah persiapan alat untuk pengambilan gambar, serta tools yang akan digunakan untuk penggabungan gambar dan pembuatan program. Pada bagian ini juga dilakukan identifikasi fasilitas yang akan disediakan pada aplikasi virtual photography.

- Perancangan sistem, meliputi perancangan workflow dan arsitektur perangkat lunak dengan menggunakan konsep analisis dan desain berorientasi object.
- Implementasi, yaitu membangun perangkat lunak yang dapat menampilkan panorama 360⁰.
- Testing, dengan mencobakan beberapa gambar sesuai dengan format virtual photography untuk memastikan bahwa perangkat lunak yang dibangun telah berjalan sesuai dengan yang diharapkan.

1.7 Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut :

Bab I Pendahuluan

Memuat latar belakang penelitian, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penelitian.

Bab II Landasan Teori

Membahas dasar-dasar teori yang digunakan dalam merancang dan membangun Virtual Photography menggunakan Matlab R2009a beserta tinjauan pustaka.

Bab III Metodologi

Memuat uraian tentang metode analisis, analisis masalah dan hasil analisis yang mencakup kebutuhan perangkat lunak, gambaran umum perangkat lunak.

Bab IV Hasil dan Pembahasan

Memuat dokumentasi hasil pengujian terhadap aplikasi serta analisa terhadap aplikasi itu sendiri

Bab V Penutup

Memuat kesimpulan-kesimpulan dari seluruh rangkaian proses implementasi perangkat lunak, baik pada tahap analisis, perancangan, implementasi, terutama pada analisis kinerja aplikasi. Bab ini juga membahas saran yang dapat digunakan oleh pihak yang berkepentingan maupun untuk peneliti terhadap kekurangan serta keterbatasan dalam penelitian ini.

BAB II

STUDI PUSTAKA

2.1 Pengantar Teknik Image Mosaicing

Pembentukan Virtual Photography Berbasis Image Mozaicing merupakan sebuah perangkat lunak komputer yang berfungsi untuk memberikan tampilan berbeda dari sebuah gambar. Konsep dari perangkat lunak ini adalah menggabungkan beberapa gambar yang kemudian divirtualisasikan.

Ada beberapa metode image mosaicing yang pernah ada. Metode pertama yaitu Alexander, Paul. *PEMBENTUKAN CITRA PANORAMA 360 DENGAN IMAGE MOZAICING*. Fakultas Teknologi Industri, Universitas Kristen Petra. Kelemahan yang kami temukan pada metoda yang sudah diterapkan pada metode ini adalah di dalam transformasi yang dilakukan terhadap seluruh wilayah citra untuk menyamakan sudut dan posisi perspektif dua buah citra. Jika hal ini dilakukan untuk jumlah citra yang besar maka citra yang terakhir akan mengalami distorsi yang sangat besar bahkan mungkin mencapai keadaan degenerasi.

Kemudian metode kedua oleh Adipranata, Rudy...., 2006: *IMPLEMENTASI PANORAMIC IMAGE MOSAIC DENGAN METODE 8 PARAMETER PERSPECTIVE TRANSFORMATION*. Dalam Image Mosaic ini ada beberapa metode seperti Cylindrical and Spherical Panoramas, Perspective (8-Parameter) Panoramas, Rotational (3-Parameter) Panoramas, dan lain-lain. Metode Cylindrical and Spherical Panoramas hanya dapat menggabungkan

gambar-gambar yang diambil dengan rotasi pengambilan terkontrol dengan baik. Sedangkan metode Perspective (8-Parameter) Panoramas dan Rotational (3-Parameter) Panoramas dapat menggabungkan gambar-gambar yang rotasi pengambilannya tidak terkontrol dengan baik. Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra Jl. Siwalankerto 121-131, Surabaya

Metode ketiga oleh Harmokho, Harris.....,2006: PERANCANGAN APLIKASI PENGGABUNGAN IMAGE DENGAN METODE IMAGE MOSAICING. Permasalahan yang ada dalam penggabungan image adalah waktu yang cukup lama dalam mencocokkan gambar, karena masih membutuhkan persepsi visual dari orang itu sendiri. Masalah muncul ketika image yang ingin digabungkan lebih dari 3, karena otomatis waktu yang digunakan sudah pasti akan berlipat ganda karena masih bersifat manual dengan cara penggunaan program – program pengedit image.

Adapun cara penggabungan 2 image yang adalah dengan cara pergeseran antar image sehingga dirasa sudah tidak ada lagi bagian dari image yang berulang dengan bantuan metode Geometri Hashing dan Random Adaptive serta penggabungan image menggunakan Algoritma Harris Stephen.

Dari perancangan ini memperlihatkan hasil yang cukup baik dengan error yang cukup kecil dan dengan waktu yang relatif cepat

Hasil dari pembacaan `imread('filename')` bisa berupa matriks dua dimensi jika gambar yang dibaca adalah gambar grayscale dan matrik 3 dimensi jika berupa gambar 3 dimensi.

2.2 Image Proccesing

Proyek akhir ini merupakan bentuk implementasi aplikasi yang memanfaatkan image processing. Image processing adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra dengan kualitas yang lebih baik. Dengan kata lain image processing atau pengolahan citra digital merupakan suatu proses dari gambar asli menjadi gambar lain yang sesuai dengan keinginan kita. Misal suatu gambar yang kita dapatkan terlalu gelap maka dengan image processing gambar tersebut bisa kita proses sehingga mendapatkan gambar yang jelas. Secara garis besar bisa kita gambarkan seperti blok diagram pada gambar 2.1 berikut ini:



Gambar 2.1 Blok Diagram Pengolahan Citra

Proses Filter yang digambarkan diatas tentu saja bukan satu- satunya operasi dalam image processing akan tetapi hanya salah satu contoh implementasi perbaikan citra. Pada umumnya operasi-operasi pengolahan citra dilakukan apabila:

- Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung dalam citra.

- Elemen citra perlu dicocokkan, dikelompokkan atau diukur.
- Sebagian citra perlu digabung dengan bagian citra yang lain

Terminologi yang berkaitan dengan image processing adalah computer vision. Pada hakikatnya computer vision mencoba meniru cara kerja sistem visual manusia (human vision). Computer vision (machine vision) membutuhkan pemahaman dari informasi visual yang diterima kemudian mendefinisikan informasi tertentu dalam tema model parametric. Seringkali bentuk dasar dari inputan mengimplementasikan sistem pengenalan pola(pattern recognition).

Berikut ini tabel yang menjelaskan perbedaan antara image processing, computer vision dan computer graphics.

Tabel 2.1 Tabel Perbedaan Image Processing, Computer Vision, Computer Graphics

VISUAL INTEPRETATION	INPUT	OUTPUT
Image Processing	Image	Image
Computer Vision	Image	Deskripsi
Computer Graphics	Deskripsi	Image

2.3 Ciri Suatu Gambar

Dalam hal ini yang dimaksud dengan ciri merupakan suatu tanda tertentu / tanda khusus yang khas dan membedakan antara suatu image dengan image yang lainnya. Adapun yang termasuk ciri-ciri dasar dari suatu image meliputi:

- Warna

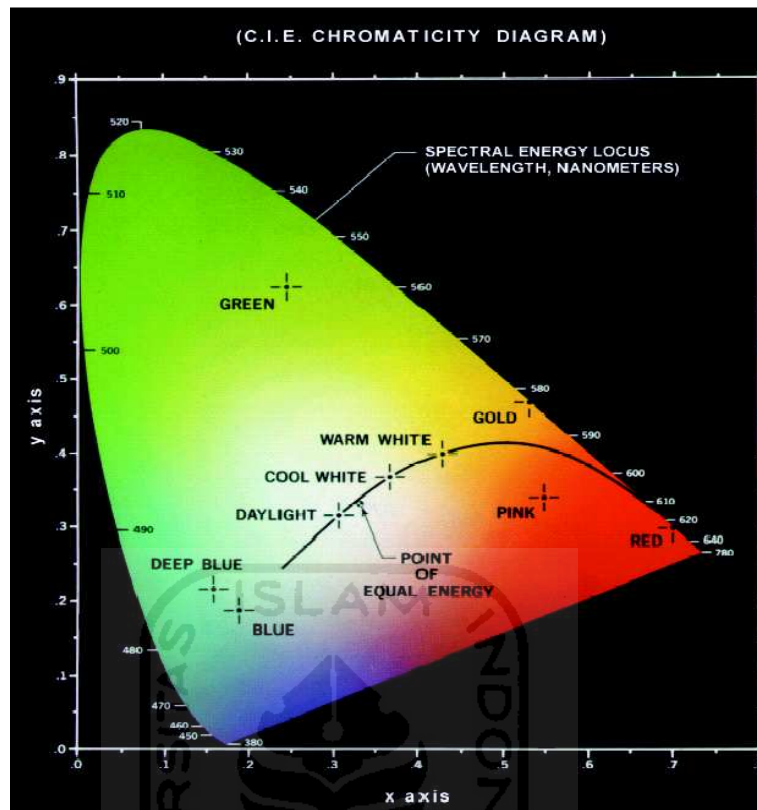
Warna yang diterima oleh mata kita dapat didefinisikan menjadi 3 warna utama yakni merah(r), hijau (g), dan blue (b).

Ciri suatu image dapat dinyatakan dalam bentuk histogram dari image tersebut yang dituliskan dengan $H(r,g,b)$, dimana $H(r,g,b)$ adalah jumlah munculnya pasangan warna r (red), g (green), dan b (blue) tertentu. Seperti dijelaskan tadi warna pokok dalam pengelolaan citra terdiri dari 3(tiga) unsur, yaitu merah (r), hijau (g), dan biru (b). Jika warna – warna pokok tersebut digabungkan, maka akan menghasilkan warna lain.



Gambar 2.2 RGB 24-bit color cubs

Warna yang dideskripsikan dengan RGB adalah pemetaan yang mengacu pada panjang gelombang dari RGB[5]. Pemetaan menghasilkan nuansa warna untuk masing-masing R, G, dan B. Masing-masing R, G, dan B didiskritkan dalam skala 256, sehingga RGB akan memiliki indeks antara 0 sampai 255. Berikut adalah gambar perbedaan panjang gelombang masing-masing warna



Gambar 2.3 Diagram Kromasitas

Dalam pengerjaan proyek akhir ini sebagian besar menggunakan ciri warna sebagai identitas suatu image.

- Bentuk

Ciri bentuk suatu gambar dapat ditentukan oleh tepi (sketsa), atau besaran moment dari suatu gambar. Pemakaian besaran moment pada ciri bentuk ini banyak digunakan orang dengan memanfaatkan nilai nilai transformasi fourier dari gambar. Proses yang dapat digunakan untuk menentukan ciri bentuk adalah deteksi tepi, threshold, segmentasi dan perhitungan moment seperti (mean, median dan standard deviasi dari setiap lokal gambar).

- Tekstur

Ciri tekstur suatu gambar dapat ditentukan dengan menggunakan filter Gabor . Ciri tekstur ini ampuh dalam menentukan informasi suatu gambar bila digabungkan dengan ciri warna suatu gambar.

2.4 Feature Matching

Penentuan kolom korespondensi nantinya diperlukan untuk menentukan bagian mana dari image yang perlu pengolahan transformasi lebih lanjut. Perlu diingat kembali bahwa teknik mosaic yang digunakan pada sistem merupakan syntetic wide-angle camera atau dengan kata lain merupakan tipuan kamera yang seolah-olah menampilkan kamera yang mempunyai sudut tangkap lebar dan teknik ini berhasil selama pengambilan image memiliki pusat proyeksi yang sama.

Sebelum sistem memberlakukan transformasi geometric pada image yang sudah tergabung, akan dicari terlebih dahulu titik pada kolom korepondensi yang menjadi acuan sumbu putar image 2. Metode pencarian ini dikenal dengan feature matching. Adapun didalam metode feature matching ada 2 algoritma dasar yakni:

- a. Direct (pixel – based) alignment

Yang termasuk dalam algoritma ini adalah metode

1. Brute Force Search

Metode ini akan mencari titik korespondensi dengan mengandalkan kesamaan ciri warna pada pixel tetangga dari titik yang dijadikan acuan

pencarian. Besarnya keruangan tetangga dan range yang dijadikan sebagai area pencarian tergantung pada input sistem.

Sistem proyek akhir ini mengimplementasikan brute force search untuk mencari letak titik korespondesinya. Keuntungan menggunakan metode ini adalah metode ini merupakan metode dengan implementasi yang paling sederhana dan praktis serta relatif akurat untuk mencari korespondensi titik akan tetapi kerugiannya adalah permasalahan yang akan timbul ketika kita salah menentukan besar x_0, y_0 dan step terutama yang berhubungan dengan waktu komputasi yang akan semakin lama.

2.5 Image Mosaicing

Metode khusus diperlukan untuk membentuk citra dengan sudut pandang yang lebar & distorsi yang kecil. Hal ini akan menjelaskan sebuah metode untuk memproses sekelompok citra dengan sudut pandang normal (kecil) menjadi sebuah citra dengan sudut pandang 360° .

Citra-citra normal tersebut memiliki bagian tumpang-tindih & proses ini akan membentuk penyambungan yang halus antara setiap pasangan citra normal tersebut. Metode-metode lain yang sudah umum, bekerja berdasarkan transformasi geometri pada seluruh bagian citra-citra normal. Dalam metode-metode tersebut, ada citra acuan dan citra yang akan digabungkan. Citra yang akan digabungkan ditransformasi sehingga obyek-obyek dalam citra tampak dilihat dari sudut pandang yang sama dengan citra acuan. Metode-metode ini akan menghasilkan distorsi yang semakin besar untuk penggabungan sudut

pandang yang makin besar sehingga tidak dapat diterapkan untuk pembentukan citra panorama 360°.

Prinsip dasar yang membedakan pendekatan *image mosaicing* yang diterapkan pada artikel ini dengan penerapan *image mosaicing* lainnya adalah,

- ✓ Transformasi geometrik hanya dilakukan pada daerah tumpang tindih antara setiap pasang citra normal yang digabungkan.
- ✓ Penyambungan hasil transformasi dilakukan dengan pencampuran (*image blending*) dengan pembobotan untuk menghindari terjadinya efek patahan pada citra hasil.

Kelebihan metode ini adalah dapat membentuk citra panorama 360° dengan perubahan sudut pandang bebas antar citra normal sehingga tidak diperlukan peralatan khusus untuk mengambil citra-citra normal untuk pembentukan panorama.

Kekurangan metode ini adalah belum dapat mengakomodasi pergerakan atau guncangan vertikal yang besar terhadap kamera dengan baik.

Tujuan yang ingin dicapai dalam penelitian ini adalah diperolehnya cara untuk membentuk panorama 360° dengan menggabungkan sejumlah citra tunggal yang diperoleh dengan menggunakan kamera biasa. Berbeda dengan cara yang dilakukan oleh perangkat lunak QuickTime VR dimana digunakan alat khusus untuk memutar kamera dengan sudut yang tepat, cara yang kami gunakan di sini tidak memerlukan alat khusus dan posisi sudut pengambilan setiap citra tidak perlu diketahui. Lebih dari itu, posisi horisontal

citra satu dengan yanglainnya pada pada saat pengambilan citra dengan kamera dimungkinkan untuk berbeda tanpa mengorbankan kualitas gambar terlalu banyak.

Dalam percobaan yang telah dilakukan, setiap gambar panorama dibentuk dari 8 gambar tunggal yang memiliki wilayah tumpang-tindih sekitar 20-30%. Wilayah tumpang tindih yang semakin besar akan memberikan gambar panorama yang lebih baik namun memerlukan jumlah gambar tunggal yang lebih banyak.

2.6 Transformasi Geometrik

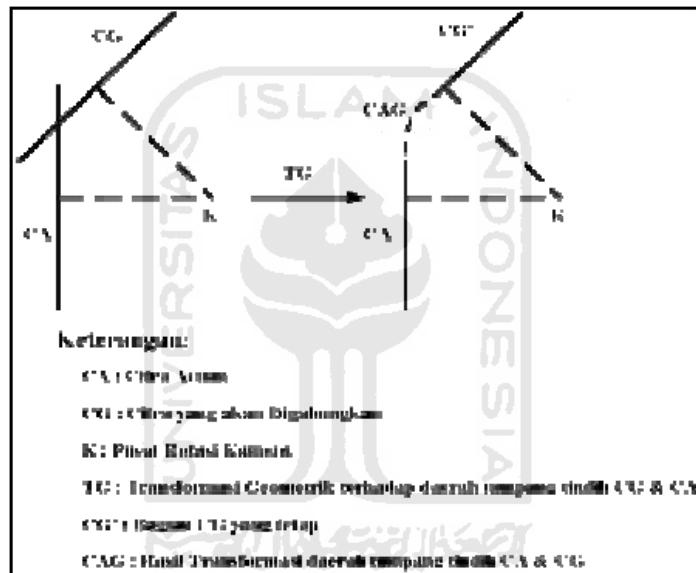
Untuk mengatasi kelemahan ini, transformasi geometrik dilakukan hanya terhadap wilayah gambar yang tumpang tindih saja. Dengan konsep ini, sebuah citra tunggal I_k di dalam rangkaian citra $I_1, \dots, I_{k-1}, I_k, I_{k+1}, \dots, I_n$ akan memiliki tiga bagian: kiri (<), tengah (-), dan kanan (>).

1. $I_{k,<}$ bagian kiri yang tumpang tindih dengan bagian kanan $I_{k-1,>}$
2. $I_{k,-}$ bagian tengah yang tidak tumpang tindih dengan citra lain
3. $I_{k,>}$ bagian kanan yang tumpang tindih dengan bagian kiri $I_{k+1,<}$

Transformasi geometrik dilakukan hanya untuk kasus (1) dan (3) di atas. Dengan demikian dari n citra tunggal yang akan dibentuk menjadi citra panorama, akan terdapat n pasang wilayah tumpang tindih yang dibentuk dari pasangan

$$[I_{1,>}, I_{2,<}] [I_{2,>}, I_{3,<}] [I_{3,>}, I_{4,<}] , \dots, [I_{n,>}, I_{1,<}].$$

Untuk setiap pasangan $[I_{k,>}, I_{k+1,<}]$ ditentukan dua fungsi pemetaan $f_k(x, y)$ dan $g_k(x, y)$. Fungsi $f_k(x, y)$ memetakan piksel-piksel dari I_k ke $I_{k+1,<}$, sedangkan fungsi $g_k(x, y)$ memetakan piksel-piksel dari $I_{k+1,<}$ ke $I_{k,>}$. Dalam implementasi, pemetaan dilakukan secara bertahap untuk membentuk kolom-kolom piksel pada daerah tumpang tindih menjadi daerah transisi perspektif antara I_k dan I_{k+1} .



Gambar 2.4 Penerapan transformasi geometrik secara bertahap pada daerah tumpang tindih kedua citra yang digabungkan

Fungsi $f_k(x, y)$ dan $g_k(x, y)$ adalah fungsi bernilai vektor 2D yang memiliki bentuk:

$$f(x, y) = \begin{bmatrix} p1 & p2 & p3 & p4 & p5 & 0 \\ p6 & p7 & p8 & 0 & p4 & p5 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \\ x^2 \\ xy \\ y^2 \end{bmatrix} \quad (2.1)$$

Fungsi (2.1) diatas , diturunkan dari konsep *moving planar surface* .
 Nilai ke delapan cvb parameter ($p_1, p_2, \dots, p_8$) pada fungsi tersebut
 ditentukan berdasarkan korespondensi empat pasang titik yang ditemukan
 pada wilayah tumpang tindih.

Untuk mendapatkan sambungan yang halus dari I_k ke I_{k+1} dilakukan
 pencampuran (dengan pembobotan dari hasil pemetaan $f_k(x, y)$ dan $g_k(x, y)$).

2.7 Blending Image

Tujuan tahapan blending image merupakan proses pencampuran image,
 menghapus garis potongan agar potongan antar image menjadi lebih halus
 dan tidak tampak. Idealnya masing-masing sample image yang mengoverlap
 adalah pixel dari image yang sama akan tetapi tidak pada kenyataannya, hal
 ini dikarenakan perubahan waktu aperture/eksposurennya (titik bidik
 kamera), peredupan (intensitas cahaya berkurang ke arah edge (tepi) image,
 efek parallax dikarenakan pergerakan yang tidak diharapkan dari pusat optik
 kamera. Secara teknis proses ini akan mengambil kedua image dengan
 prosentasi tertentu untuk dicampur dan dijadikan dalam satu image.

$$Z = a X + (1-a) Y \quad (2.2)$$

dengan:

Z = pixel image hasil blending

a = prosentasi pixel

X = komponen warna pixel image 1

Y = komponen warna pixel image 2

Blending dilakukan terhadap kolom dari image yang termasuk dalam kolom korespondensi.

Berikut adalah contoh perkiraan hasil yang dihasilkan sistem:



Gambar 2.5 Konstruksi Image Dinamis

BAB III

METODOLOGI

3.1 Gambaran umum

Perancangan sistem yang akan dibuat adalah virtual fotografi berbasis image mosaicing tujuan utama dari program simulasi ini adadlah merancang suatu sistem yang mampu menmpilkan sebuah gambar secara luas yang belum mampu dilakukan oleh kamera.gambar yang diambil menggunakan kamera dSLR atau kamera digital biasa diolah sedemikina rupa menggunakan metode pengolahan citra hingga akhirnya diperoleh hasil gambar dengan wide angle. Simulasi ini dirancang dengan tidak menutupi kemungkinan untuk dikembangkan lebih lanjut.

3.2 Deskripsi Sistem

3.2.1 Perangkat Keras (Hard ware)

Perangkat keras yang digunakan untuk merancang simulasi dalam tugas akhir ini adalah:

- sebuah notebook dengan spesifikasi sebagai berikut:

Notebook Lenovo 3000 G230 T6600 @ 2.20GHz 2.20 GHz

Sistem notebook : Intel(R) Core(TM)2 Duo CPU

Sistem operasi : Microsoft windows seven ultimate

media tampilan : 12"

Media Masukan : Papan ketik (keyboard) dan mouse

Memori : 2 GB RAM



Gambar 3.1 Lenovo 3000 G230

- Kamera dSLR Nikon d300s dengan spesifikasi:
 - ✓ Image Sensor Format : DX
 - ✓ Sensor Size : 23.6 x 15.8mm
 - ✓ Total Pixels : 13.1 million
 - ✓ LCD Monitor Size : 3 inch diagonal
 - ✓ LCD Monitor Resolution : 920,000 Dots
 - ✓ Slowest Shutter Speed : 30 sec
 - ✓ Fastest Shutter Speed : 1/8000 sec
 - ✓ Mirror Lock Up : Yes
 - ✓ Lowest ISO Sensitivity : ISO 200
 - ✓ Highest ISO Sensitivity : ISO 3200
 - ✓ Highest Expanded ISO Sensitivity : ISO 6400 equivalent

- ✓ Storage Media : CompactFlash© (Type I/II, compliant with UDMA), SD, SDHC
- ✓ Movie Modes : Movie with sound (HD1280×720 /24fps, VGA640×424 /24fps, QVGA320×216 /24fps)
- ✓ Autofocus Sistem : Autofocus TTL phase detection, 51 focus points
- ✓ Built-in Flash : Yes
- ✓ Flash Sync Speed : Up to 1/250
- ✓ Battery : Rechargeable EN-EL3e Lithium-ion Battery
- ✓ Battery Life (shots per charge) : 950 shots (CIPA), 3000 shots (Nikon Standard)
- ✓ Weight : 30oz (840 grams)



Gambar 3.2 Kamera dSLR Nikon d300s

- Kamera digital Panasonic Lumix F3

Dengan spesifikasi:

12.1 megapixel

4x optical zoom

28mm wide-angle lens

Digital image stabilizer

Face Detection

HD Movie for PC

battery Lithium



Gambar 3.3 Kamera Digital Panasonic Lumix F3

- Tripod Reebon dengan spesifikasi
 - ❖ 3-section, Weight 1 kg, Height 145 cm, Load 2 kg, include Pan and Tilt Head with Quick Release



Gambar 3.4 Tripod Velbon

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam tugas akhir ini adalah

1. Matlab R2009a : Toolbox matlab yang digunakan adalah image processing toolbox, image processing toolbox digunakan untuk menangani pengolahan citra. GUI matlab juga digunakan untuk membuat tampilan secara visual sebagai antarmuka bagi pengguna
2. Adobe Photoshop CS3 : Digunakan sebagai program untuk melakukan proses rezising gambar. Gambar ini akan digunakan sebagai database.

3.3 Perancangan sistem virtual fotografi

3.3.1 Perancangan Data

Perancangan data meliputi penentuan format data image masukan sistem juga termasuk pula ukuran file yang akan diproses. Sistem proyek akhir ini dirancang untuk bisa menerima masukan dari semua format gambar termasuk di dalamnya bitmap(*.bmp), jpeg(*.jpg), gif(*.gif), PNG(*.png) ataupun TIFF(*.tiff). Sedangkan ukuran file sangat mempengaruhi waktu komputasi yang dibutuhkan dan ukuran file yang diterima oleh sistem adalah 200 X 150 sehingga gambar yang diambil dari kamera perlu melalui proses kompresi terlebih dahulu agar tidak membutuhkan waktu proses yang terlalu lama. Bahasa pemrograman yang digunakan adalah Visual Basic.NET. Perlu diperhatikan pula bahwa data masukan adalah gambar dalam keadaan terurut yang diambil dari kamera dengan pusat proyeksi yang sama dalam artian kita tidak berpindah tempat dalam mengambil gambar masukan.

3.3.2 Perancangan Form Perangkat Lunak

Telah dijelaskan pada bab sebelumnya metode/prosedur yang digunakan dalam pembuatan sistem meliputi :

- Masukan Gambar
- Image Matching

Dalam hal ini perancangan form mengacu pada sisi tampilan GUI metode-metode diatas.

- Konstruksi Form Masukan Gambar

Masukan Image dirancang dengan sistem explorer tree, splitter, dan listview control yang memungkinkan user untuk memasukkan banyak gambar sekaligus ke dalam project dengan sistem drag and drop.

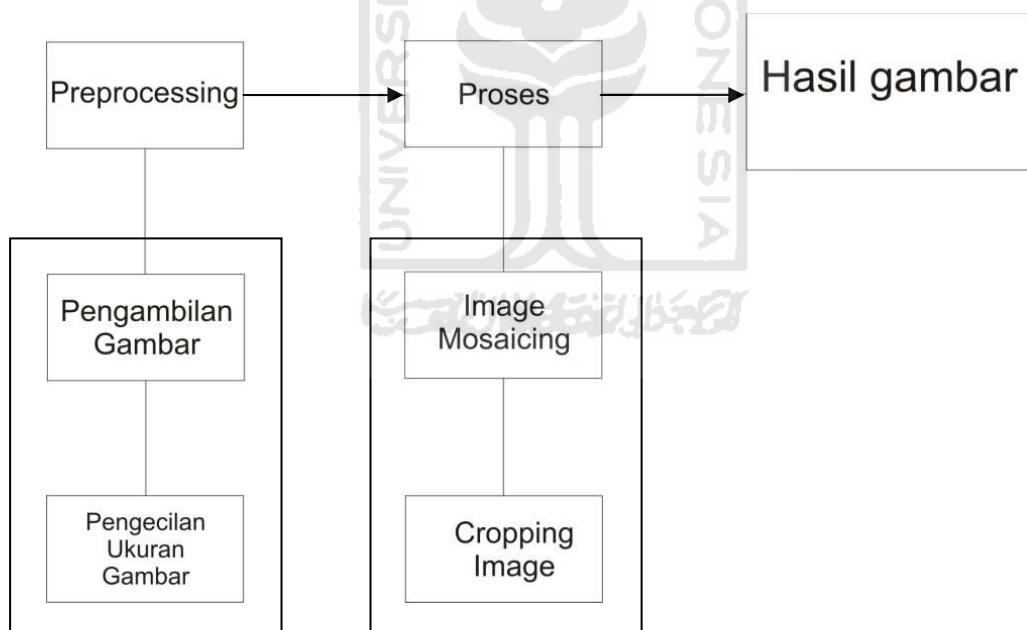
- Konstruksi Form Image Matching

Pada form perancangan ini terdapat tiga buah histogram yang masing-masing menampilkan(urut mulai dari atas ke bawah) komponen greyscale per kolom pada image 1, kemudian komponen greyscale pada range pertama(dalam hal ini range yang digunakan 10 kolom) pada image2 dan terakhir adalah histogram yang menunjukkan korespondensi greyscale image 1 dan image 2. Pada histogram yang terakhir inilah dapat diketahui kolom korespondensi yaitu letak titik yang menunjukkan tingkat kemiripan paling tinggi diantara keduanya dan di kolom ini pula image 2 akan mengoverlay pada image1.

Hasil perhitungan matematis pembentukan histogram tersebut ditampilkan dalam listview sehingga mempermudah user untuk mengetahui dengan pasti perbedaan antara kedua image. Selain itu nilai perbandingan yang paling minimal menunjukkan kolom korespondensi.

Secara garis besar perancangan sistem virtual fotografi ini dibagi menjadi beberapa proses:

1. Preprocessing
2. Image Reziseing
3. Proses image mosaicing



Gambar 3.5 Blok diagram sistem Virtual fotografi berbasis image mosaicing

3.4 Perancangan antar muka sistem

1. Interface Menu Utama

Merupakan menu utama sistem ini. Semua tahapan sistem ada di menu utama ini. Bentuk rancangan menu utama adalah sbagai berikut:



Gambar 3.6 Interface menu utama virtual fotografi

Interface menu utama dari program terdiri dari:

1. Menu “masukan nama image” digunakan untuk memasukan nama image yang akan diproses dalam sistem.
2. Tombol “Read Image 1” digunakan untuk memasukan nama image 1 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



Gambar 3.7 Interface masukan image 1

3. Tombol “Read Image 2” digunakan untuk memasukan nama image 2 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



Gambar 3.8 Interface masukan image 2

4. Tombol “Read Image 3” digunakan untuk memasukan nama image 3 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



Gambar 3.9 Interface masukan image 3

5. Tombol “Read Image 4” digunakan untuk memasukan nama image 4 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



Gambar 3.10 Interface masukan image 4

6. Tombol “Read Image 5” digunakan untuk memasukan nama image 5 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



Gambar 3.11 Interface masukan image 5

7. Tombol “Read Image 6” digunakan untuk memasukan nama image 6 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



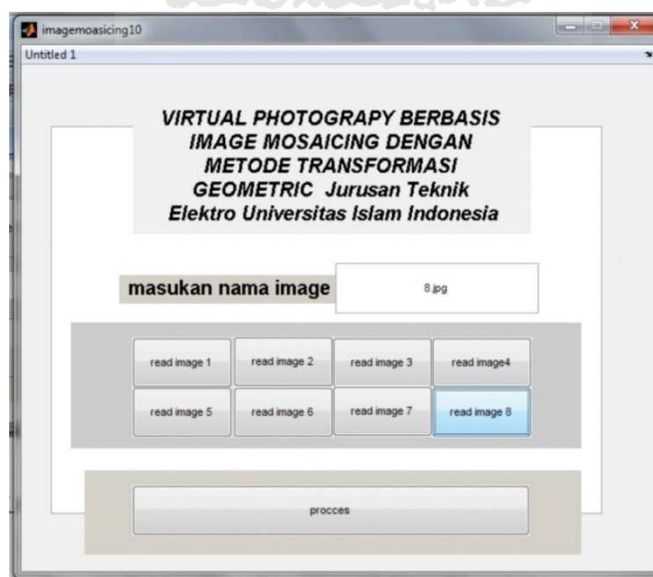
Gambar 3.12 Interface masukan image 6

8. Tombol “Read Image 7” digunakan untuk memasukan nama image 7 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



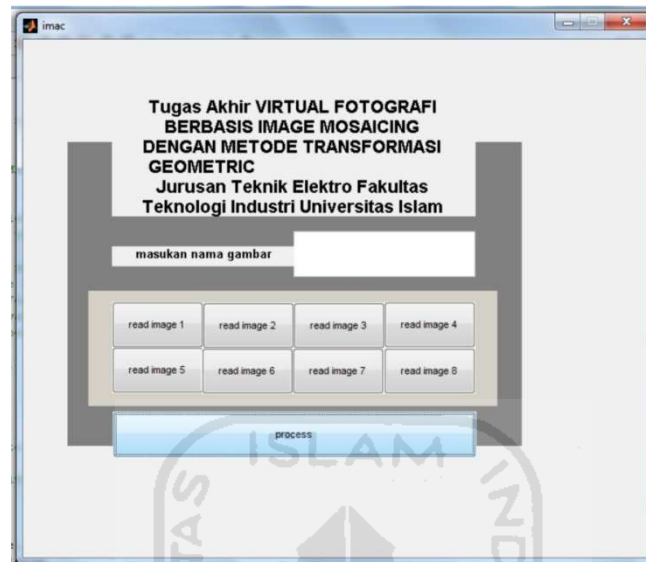
Gambar 3.13 Interface masukan image 7

9. Tombol “Read Image 8” digunakan untuk memasukan nama image 8 yang sudah ada dalam tabel untuk diproses dalam image mosaicing.



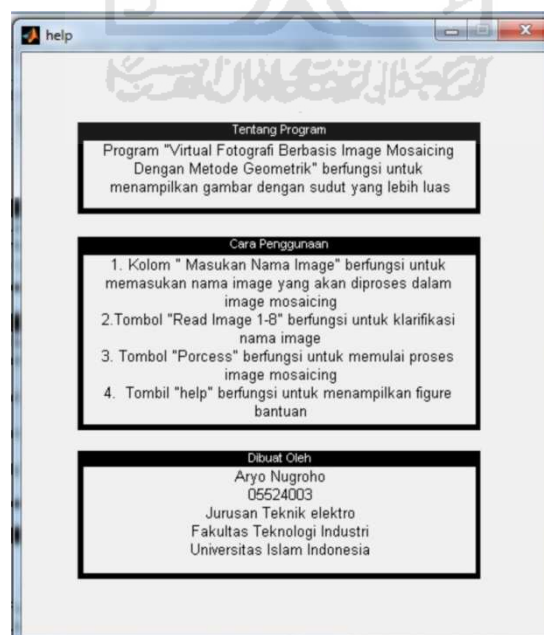
Gambar 3. 14Interface masukan image 8

10. Tombol “proses” digunakan untuk memulai proses image mosaicing setelah 8 image telah dmasukan.



Gambar 3.15 Figure proses

11. Tombol “Help” berfungsi untuk menampilkan figure bantuan dan cara kerja program.



Gambar 3.16 Figure Help

Berikut adalah beberapa tahap yang dilakukan dalam proses virtual fotografi dengan image mosaicing.

1. Masukan 8 gambar

```
G1=double(imread('1.jpg'));  
G2=double(imread('2.jpg'));  
G3=double(imread('3.jpg'));  
G4=double(imread('4.jpg'));  
G5=double(imread('5.jpg'));  
G6=double(imread('6.jpg'));  
G7=double(imread('7.jpg'));  
G8=double(imread('8.jpg'));
```

G1-G8 merupakan inisialisasi gambar. G1 merupakan inisial dari gambar 1, G2 merupakan inisial dari gambar 2, G3 merupakan inisial dari gambar 3, G4 merupakan inisial dari gambar 4, G5 merupakan inisial dari gambar 5, G6 merupakan inisial dari gambar 6, G7 merupakan inisial dari gambar 7 dan G8 merupakan inisial dari gambar 8. Gambar 1-8 adalah objek yang akan diproses di image mosaicing. Pada matlab fungsi untuk melakukan pembacaan image standar yaitu:

imread('filename')

Perintah ini digunakan untuk membaca beberapa format file diantaranya: Hasil dari pembacaan *imread('filename')* bisa berupa matriks dua dimensi jika gambar

yang dibaca adalah gambar grayscale dan matrik 3 dimensi jika berupa gambar 3 dimensi.

2. Penggabungan gambar tahap I

```
for k=1:4
    if k==1
        I1=G1;
        I2=G2;
    elseif k==2
        I1=G3;
        I2=G4;
    elseif k==3
        I1=G5;
        I2=G6;
    elseif k==4
        I1=G7;
        I2=G8;
    end
```



Program di atas adalah perintah penggabungan tahap pertama yaitu untuk k1-k4. Untuk K=1 yaitu penggabungan gambar G1 dan G2 dinamakan sebagai I1 dan I2. Jika K==1 maka I1 adalah G1 atau gambar 1 dan I2 adalah G2 atau gambar 2. Jika K==2 maka I1 adalah G3 atau gambar 3 dan I2 adalah G4 atau gambar 4.

Jika $K=3$ maka I1 adalah G5 atau gambar 5 dan I2 adalah G6 atau gambar 6.

Jika $K=4$ maka I1 adalah G7 atau gambar 7 dan I2 adalah G8 atau gambar 8.

a. memuat masukan gambar

```
[h1 w1 d1] = size(I1);  
[h2 w2 d2] = size(I2);
```

b. Menampilkan 2 gambar dalam satu figure

```
figure; subplot(1,2,1); image(I1/255); axis image; hold on;  
title('first input image');  
[X1 Y1] = ginput2(2); % get two points from the user  
subplot(1,2,2); image(I2/255); axis image; hold on;  
title('second input image');  
[X2 Y2] = ginput2(2); % get two points from the user
```

Pemberian titik persamaan [X1 Y1] untuk [X2 Y2] dilakukan dalam proses ini. 2 gambar ditampilkan dalam 1 figure agar dapat diberikan titik persamaan antara gambar 1 dan gambar 2.

Figure menciptakan grafis gambar objek. Figur objek adalah jendela individu pada layar di mana perangkat lunak MATLAB menampilkan output grafis.

```
figure
figure('PropertyName',propertyvalue,...)
figure(h)
h = figure(...)
```

Subplot membagi figure ke panel persegi panjang yang diberi nomor rowwise.

Setiap panel berisi sumbu objek. Plot berikutnya adalah output ke pane saat ini.

```
h = subplot(m,n,p) or subplot(mnp)
subplot(m,n,p,'replace')
subplot(m,n,P)
subplot(h)
subplot('Position',[left bottom width height])
subplot(..., prop1, value1, prop2, value2, ...)
h = subplot(...)
subplot(m,n,p,'v6')
```

Ginput digunakan untuk memilih poin dari angka menggunakan mouse untuk posisi kursor. Angka tersebut harus memiliki fokus sebelum ginput menerima input. estimasi parameter vektor (t)

```
[x,y] = ginput(n)
[x,y] = ginput
[x,y,button] = ginput(...)
```

c. estimasi parameter vektor (t)

```
Z = [ X2' Y2' ; Y2' -X2' ; 1 1 0 0 ; 0 0 1 1 ]';
xp = [ X1 ; Y1 ];
```

```

t = Z \ xp; % solve the linear sistem
a = t(1); % = s cos(alpha)
b = t(2); % = s sin(alpha)
tx = t(3);
ty = t(4);

```

Proses diatas merupakan pembentukan matrix. Z digunakan untuk mencari persamaan linier. xp merupakan output dari G1.

d. membangun matriks transformasi (T)

```

T = [a b tx ; -b a ty ; 0 0 1];

```

e. sudut lengkung masuk untuk menentukan ukuran gambar output

```

cp = T*[ 1 1 w2 w2 ; 1 h2 1 h2 ; 1 1 1 1 ];
Xpr = min( [ cp(1,:) 0 ] ) : max( [cp(1,:) w1] ); % min x : max x
Ypr = min( [ cp(2,:) 0 ] ) : max( [cp(2,:) h1] ); % min y : max y
[Xp,Yp] = ndgrid(Xpr,Ypr);
[wp hp] = size(Xp); % = size(Yp)

```

Perintah di atas merupakan perintah untuk penyesuaian gambar dengan parameter aljabar.

f. melakukan transformasi mundur (dari luar ke dalam)

```
X = T \ [ Xp(:) Yp(:) ones(wp*hp,1) ]'; % warp
```

g. Penyesuaian warna Red Green Blue

```
clear Ip;
xI = reshape( X(1,:),wp,hp)';
yI = reshape( X(2,:),wp,hp)';
Ip(:,:,1) = interp2(I2(:,:,1), xI, yI, '*bilinear'); % red
Ip(:,:,2) = interp2(I2(:,:,2), xI, yI, '*bilinear'); % green
Ip(:,:,3) = interp2(I2(:,:,3), xI, yI, '*bilinear'); % blue
```

h. Hasil Penggabungan

```
offset = -round( [ min( [ cp(1,:) 0 ] ) min( [ cp(2,:) 0 ] ) ] );
Ip(1+offset(2):h1+offset(2),1+offset(1):w1+offset(1),:) =
double(I1(1:h1,1:w1,:));
```

i. Menampilkan hasil

```
figure; image(Ip/255); axis image;
title('mosaic image');
```

3. Penggabungan Gambar Tahap II

```
if k==1
    Ip1=Ip;
elseif k==2
    Ip2=Ip;
elseif k==3
    Ip3=Ip;
elseif k==4
    Ip4=Ip;
end
end
for k=1:2
    if k==1
        I1=Ip1;
        I2=Ip2;
    elseif k==2
        I1=Ip3;
        I2=Ip4;
    end
end
```



Program di atas merupakan perintah penggabungan tahap kedua. Jika $K=1$ maka $Ip1$ sama dengan Ip . Dan jika $K==2$ maka $Ip2$ sama dengan Ip . Untuk $K==3$ maka $Ip3$ sama dengan Ip . Untuk $K==4$ maka $Ip4$ sama dengan Ip juga.

Untuk $K1$ sampai $K2$, jika $K == 1$ maka $I1$ sama dengan $Ip1$ yang merupakan hasil penggabungan dari $G1$ dan $G2$. $I2$ sama dengan $Ip2$ yang

merupakan hasil penggabungan dari G3 dan G4. Jika $K=2$ maka I1 sama dengan Ip3 yang merupakan hasil penggabungan dari G5 dan G6. Sedangkan I2 sama dengan Ip4 yang merupakan hasil penggabungan dari G7 dan G8

a. Memuat gambar masukan tahap II

```
[h1 w1 d1] = size(I1);
[h2 w2 d2] = size(I2);
```

b. Menampilkan masukan gambar dan untuk korespondensi cepat tahap II

```
figure; subplot(1,2,1); image(I1/255); axis image; hold on;
title('first input image');
[X1 Y1] = ginput2(2); % get two points from the user
subplot(1,2,2); image(I2/255); axis image; hold on;
title('second input image');
[X2 Y2] = ginput2(2); % get two points from the user
```

c. Estimasi parameter vektor (t) tahap II

```
Z = [ X2' Y2' ; Y2' -X2' ; 1 1 0 0 ; 0 0 1 1 ]';
xp = [ X1 ; Y1 ];
t = Z \ xp; % solve the linear sistem
a = t(1); % = s cos(alpha)
b = t(2); % = s sin(alpha)
tx = t(3);
```

```
ty = t(4);
```

d . Membangun matriks transformasi (T) tahap II

```
T = [a b tx ; -b a ty ; 0 0 1];
```

T merupakan matrik transformasi

e . Sudut lengkung masuk untuk menentukan ukuran gambar output tahap II

```
cp = T*[ 1 1 w2 w2 ; 1 h2 1 h2 ; 1 1 1 1 ];  
Xpr = min( [ cp(1,:) 0 ] ) : max( [cp(1,:) w1] ); % min x : max x  
Ypr = min( [ cp(2,:) 0 ] ) : max( [cp(2,:) h1] ); % min y : max y  
[Xp,Yp] = ndgrid(Xpr,Ypr);  
[wp hp] = size(Xp); % = size(Yp)
```

f . Melakukan transformasi mundur (dari luar ke dalam) tahap II

```
X = T \ [ Xp(:) Yp(:) ones(wp*hp,1) ]'; % warp
```

g . Penyesuaian warna red, green, blue

```
clear Ip;  
xI = reshape( X(1,:),wp,hp)';
```

```

yI = reshape( X(2,:),wp,hp)';
Ip(:,:,1) = interp2(I2(:,:,1), xI, yI, '*bilinear'); % red
Ip(:,:,2) = interp2(I2(:,:,2), xI, yI, '*bilinear'); % green
Ip(:,:,3) = interp2(I2(:,:,3), xI, yI, '*bilinear'); % blue

```

h. Hasil penggabungan tahap II

```

offset = -round( [ min( [ cp(1,:) 0 ] ) min( [ cp(2,:) 0 ] ) ] );
Ip(1+offset(2):h1+offset(2),1+offset(1):w1+offset(1),:) =
double(I1(1:h1,1:w1,:));

```

i. Menampilkan hasil tahap II

```

figure; image(Ip/255); axis image;
title('mosaic image');

```

4. Penggabungan gambar Tahap III

```

if k==1
    Ipp1=Ip;
elseif k==2
    Ipp2=Ip;
end
end

```

a . Memuat gambar masukan tahap III

```
I1 = Ipp1;  
I2 = Ipp2;  
  
[h1 w1 d1] = size(I1);  
[h2 w2 d2] = size(I2);
```

b . Menampilkan masukan gambar dan untuk korespondensi cepat tahap III

```
figure; subplot(1,2,1); image(I1/255); axis image; hold on;  
title('first input image');  
[X1 Y1] = ginput2(2); % get two points from the user  
subplot(1,2,2); image(I2/255); axis image; hold on;  
title('second input image');  
[X2 Y2] = ginput2(2); % get two points from the user
```

c . Estimasi parameter vektor (t) tahap III

```
Z = [ X2' Y2' ; Y2' -X2' ; 1 1 0 0 ; 0 0 1 1 ]';  
xp = [ X1 ; Y1 ];  
t = Z \ xp; % solve the linear sistem  
a = t(1); % = s cos(alpha)  
b = t(2); % = s sin(alpha)
```

```
tx = t(3);
ty = t(4);
```

d . Membangun matriks transformasi (T) tahap III

```
T = [a b tx ; -b a ty ; 0 0 1];
```

e . Sudut lengkung masuk untuk menentukan ukuran gambar output tahap III

```
cp = T*[ 1 1 w2 w2 ; 1 h2 1 h2 ; 1 1 1 1 ];
Xpr = min( [ cp(1,:) 0 ] ) : max( [cp(1,:) w1] ); % min x : max x
Ypr = min( [ cp(2,:) 0 ] ) : max( [cp(2,:) h1] ); % min y : max y
[Xp,Yp] = ndgrid(Xpr,Ypr);
[wp hp] = size(Xp); % = size(Yp)
```

f . Melakukan transformasi mundur (dari luar ke dalam) tahap III

```
X = T \ [ Xp(:) Yp(:) ones(wp*hp,1) ]'; % warp
```

j . Penyesuaian warna red, green, blue

```
clear Ip;
xI = reshape( X(1,:),wp,hp)';
yI = reshape( X(2,:),wp,hp)';
Ip(:,:,1) = interp2(I2(:,:,1), xI, yI, '*bilinear'); % red
```

```
Ip(:,:,2) = interp2(I2(:,:,2), xI, yI, '*bilinear'); % green
Ip(:,:,3) = interp2(I2(:,:,3), xI, yI, '*bilinear'); % blue
```

g. Hasil penggabungan

```
offset = -round( [ min( [ cp(1,:) 0 ] ) min( [ cp(2,:) 0 ] ) ] );
Ip(1+offset(2):h1+offset(2),1+offset(1):w1+offset(1),:) =
double(I1(1:h1,1:w1,:));
```

h. Menampilkan hasil tahap III

```
figure; image(Ip/255); axis image;
title('mosaic image');
```

BAB IV

DATA DAN ANALISA

Pembahasan pada bab ini lebih difokuskan terhadap pengujian dan analisa sistem kerja dari perangkat lunak yang telah dibuat. Hal yang banyak diuji dan dianalisa disini diantaranya algoritma atau metode yang digunakan dan masukan data yang didapatkan. Beberapa algoritma telah dicoba untuk mendapatkan hasil yang paling optimal terutama pada proses image matching dengan histogram proyeksi yaitu proses untuk menentukan kolom korespondensi (kolom dengan tingkat kemiripan paling tinggi diantara 2 gambar dan juga kolom tempat image kedua mengoverlay image pertama) antara 2 image.

4.1 Preprocessing

1. Pengambilan Gambar

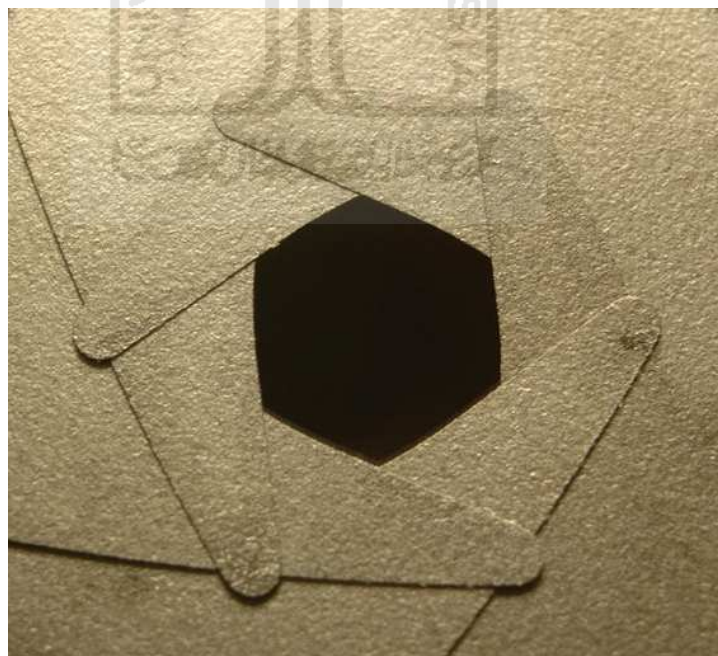
Pada saat pengambilan gambar untuk proses virtual fotografi, banyak hal yang harus diperhatikan. Contohnya adalah luas area yang akan diambil gambarnya, pencahayaan, dll. Pencahayaan merupakan hal yang paling berpengaruh dalam proses image mosaicing nantinya. Ada beberapa komponen dari kamera yang harus diatur untuk mendapatkan pencahayaan yang tepat, yaitu:

- Apperture atau diafragma

Diafragma adalah komponen dari lensa yang berfungsi mengatur intensitas cahaya yang masuk ke kamera. Diafragma lensa

biasanya membentuk lubang mirip lingkaran atau segi tertentu. Ia terbentuk dari sejumlah lembaran logam (umumnya 5, 7, atau 8 lembar) yang dapat diatur untuk mengubah ukuran lubang dimana cahaya akan lewat. Tingkap akan mengembang dan menyempit persis pupil di mata manusia.

Karena fungsinya untuk menghentikan cahaya yang akan masuk kebidang fokal, diafragma juga disebut sebagai stop, blind, field stop dan flare stop. Dan untuk itu, diafragma selalu diletakkan pada jalan masuk antara subyek, lensa dan bidang fokal. Titik tengah tingkap pada diafragma merupakan sumbu optis dari sebuah lensa.



Gambar 4.1 Diafragma

- Shutter Speed atau kecepatan rana

Dalam istilah fotografi, Kecepatan rana atau Shutter Speed adalah ukuran kecepatan rana membakar medium penangkap cahaya (lebih umum disebut film atau sensor *digital*).

Umumnya Kecepatan rana terdiri dari urutan angka 8000, 4000, 2000, 1000, 500, 250, 125, 60, 30, 15, 8, 4, 2, dan 1. Angka ini merupakan angka kebalikan dari lama pajanan dalam detik. Misalnya angka 30 berarti 1/30 detik, dan seterusnya.

Untuk kecepatan rana lebih lama dari 1 detik menggunakan tanda ". Sementara kecepatan rana bebas sesuai dengan pemencetan tombol rana oleh fotografer diberi tanda B(Bulb).

Namun angka tersebut tidaklah mutlak. Banyak produsen kamera menggunakan kecepatan rana yang hanya mendekati angka tersebut.

Kecepatan rana mempengaruhi eksposur cahaya yang membakar film. Semakin cepat pembukaan rana, semakin sedikit cahaya membakar medium, dan sebaliknya. Hal ini akan mempengaruhi pajanan.

- ISO atau kecepatan film

Kecepatan film adalah istilah dalam fotografi untuk mengukur tingkat kesensitivitas atau kepekaan film foto terhadap cahaya. Film

dengan kepekaan rendah (memiliki angka ISO rendah) membutuhkan sorotan (Inggris: exposure) yang lebih lama sehingga disebut slow film, sedangkan film dengan kepekaan tinggi (memiliki angka ISO tinggi) membutuhkan exposure yang singkat.

Standarnya dikenal dengan ISO 5800:1987 dari *International Organization for Standardization (ISO)* yang menetapkan skala linear dan skala logaritmik untuk mengukur kecepatan film. Skala linear ISO dikenal dengan ASA.

- White Balance

White Balance adalah istilah dalam fotografi untuk kalibrasi titik berwarna putih. Sebagaimana dijelaskan pada bagian suhu warna / color temperature, warna yang dianggap putih dapat bervariasi tergantung pada kondisi pencahayaan.

- Rana

Rana atau penutup dalam istilah fotografi adalah tirai pada kamera yang menutupi permukaan atau sensor foto. Jika tirai ini terbuka maka akan terjadi pajanan pada permukaan film atau sensor foto tadi.

- Lensa

PERBEDAAN penggunaan lensa memberikan perbedaan perspektif. Perspektif adalah ukuran dan kedalaman relatif subjek dalam gambar. Perspektif juga bisa berarti perubahan bentuk, ukuran, dan kedalaman bidang yang relatif akibat perbedaan cara pandang antara objek dengan kamera. Perbedaan tersebut terjadi karena ada pergeseran posisi dalam melihat sesuatu dari sudut pandang, jarak, dan ketinggian yang tidak sama maupun penggunaan lensa dengan focal length yang berbeda.

Penggeseran posisi maupun penggunaan focal length yang berbeda memberikan perspektif yang berbeda pula.

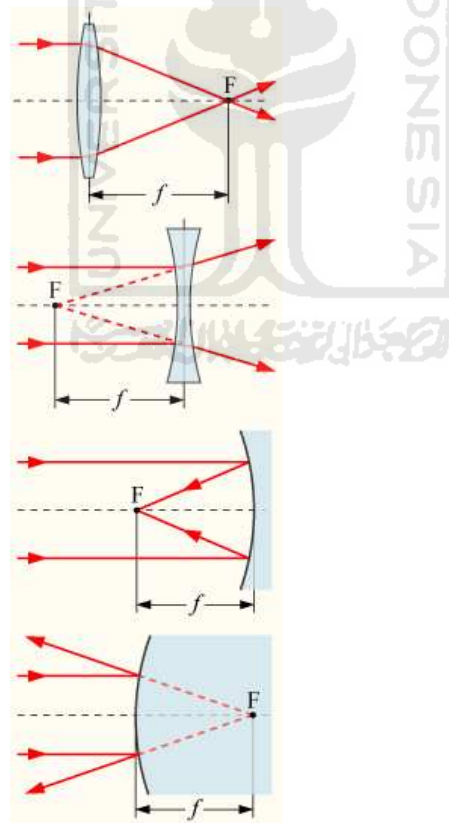
Disebut lensa standar atau lensa normal karena memiliki panjang fokus sekitar 50 mm, sama dengan mata manusia saat melihat. Perbedaan mata dengan lensa normal itu hanya terletak pada sudut pandang. Mata manusia hampir 180 derajat dalam melihat sesuatu dari depan. Sedangkan penglihatan lensa standar dibatasi jendela bidik kamera yang punya sudut pandang 46 derajat. Namun, sekarang lensa normal tidak hanya memiliki focal length pada nilai 50 mm, melainkan berkembang mulai 46 mm hingga 55 mm.

Sedangkan lensa wide memiliki focal length kurang dari 50 mm, lebih tepatnya focal length 35 mm pada kamera full frame. Sifat lensa sudut lebar adalah meluaskan pandangan atau memberikan

kesan menjauhkan sesuatu yang dekat. Tetapi, lensa lebar memberikan efek distorsi yang mencembungkan daerah sekitar lingkaran tengah dan memipihkan bagian pinggir foto. Sehingga, lensa wide (lebar) lingkungan foreground tampak lebih lebar daripada background.

- Fokal Length atau Jarak Fokus

Jarak focus adalah ukuran jarak antara elemen lensa dengan permukaan film (atau sensor digital) pada kamera.



Gambar 4.2 Focal Length atau jarak fokus

- DOF (depth of field) atau kedalaman ruang

Pada optika, terutama yang berhubungan dengan film dan fotografi, kedalaman ruang adalah istilah khusus untuk menunjukkan ruang tertentu di dalam citra yang nampak relatif tajam karena adanya perbedaan ketajaman (fokus). Walaupun sebuah lensa hanya mempunyai satu fokus yang presisi, yaitu pada bidang fokus yang berada pada satu jarak fokus (*focal distance*) tertentu, penurunan ketajaman citra pada kedua sisi di samping bidang fokus bersifat gradual, sehingga di dalam kedalaman ruang, ketidaktajaman citra dapat tidak nampak pada kondisi perspektif normal.

Pada kondisi tertentu, seluruh komponen dalam foto diharapkan untuk nampak tajam, karenanya kedalaman ruang akan dibuat besar. Pada kondisi yang lain, kedalaman ruang yang lebih kecil menjadi efektif untuk penekanan subyek fotografi pada latar depan (*foreground*) atau latar belakang (*background*). Pada sinematografi, kedalaman ruang yang besar sering disebut *deep focus* dan kedalaman ruang yang kecil disebut *shallow focus*.

Perubahan kedalaman ruang dipengaruhi oleh tiga faktor:

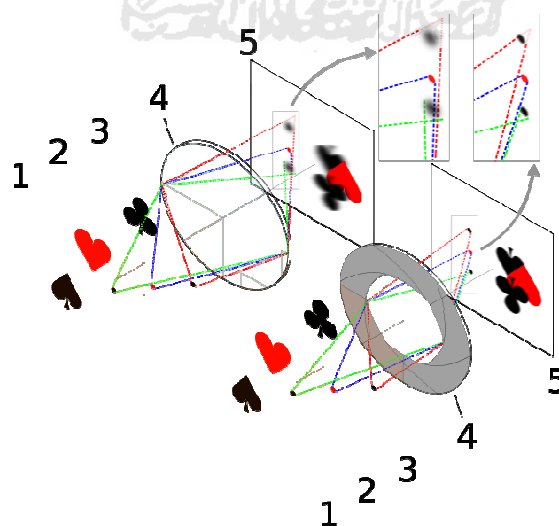
- ❖ Jarak fokus dari kamera, lebar ruang tajam berbanding lurus dengan kuadrat jarak obyek. Jika kita mengubah jarak antara

kamera dengan objek sebesar 3x (lebih jauh - dengan menggeser kamera mundur dari posisi semula) maka lebar ruang tajam akan menjadi 9x lebar semula. Dengan kriteria lingkaran gamang (circle of confusion).

- Saat fokus ditetapkan pada jarak hiperfokal (hyperfocal distance), kedalaman ruang akan berkisar antara setengah dari jarak hiperfokal hingga tak berhingga, dan merupakan kedalaman ruang terbesar untuk sebuah nilai bukaan.
- ❖ bukaan (f-number) dari rana, lebar ruang tajam berbanding lurus dengan rana. Contoh: jika rana dinaikkan 2 stop dari $f/8$ ke $f/16$, maka lebar ruang tajam akan menjadi 2x lebar semula.
- ❖ panjang fokus (focal length) dari lensa yang digunakan, lebar ruang tajam berbanding terbalik dari kuadrat panjang fokus. Dengan kata lain, lebar ruang tajam akan menjadi 4x lebar semula jika kita mengubah lensa dari 100mm ke 50mm (panjang fokus lensa setengah dari semula).
- ❖ Besar format (format size) bidang fokal, dengan subyek pada jarak moderat, kedalaman ruang ditentukan oleh pembesaran (magnification ratio) subyek dan bukaan rana (f-number). Pada bukaan tertentu, penambahan pembesaran, baik dengan bergerak mendekati subyek maupun dengan penggunaan lensa dengan panjang fokus yang lebih besar, akan menurunkan kedalaman ruang; sebaliknya menurunkan pembesaran akan membesarkan

kedalaman ruang. Pada pembesaran tertentu, menaikkan nilai bukaan (mengecilkan diameter rana) akan membesarkan kedalaman ruang; menurunkan nilai bukaan akan membesarkan kedalaman ruang.

- ❖ Ketika sebuah foto dibuat dengan dua besar format yang berbeda pada bukaan dan lensa yang memberikan sudut pandang yang sama, besar format yang lebih kecil akan memberikan kedalaman ruang yang lebih besar.
- ❖ Ketika sebuah foto dibuat dengan dua besar format yang berbeda pada bukaan, panjang fokus lensa dan jarak fokus (focal distance), jarak antara subyek dengan bidang fokal) yang sama, besar format yang lebih kecil akan menghasilkan kedalaman ruang yang lebih dangkal

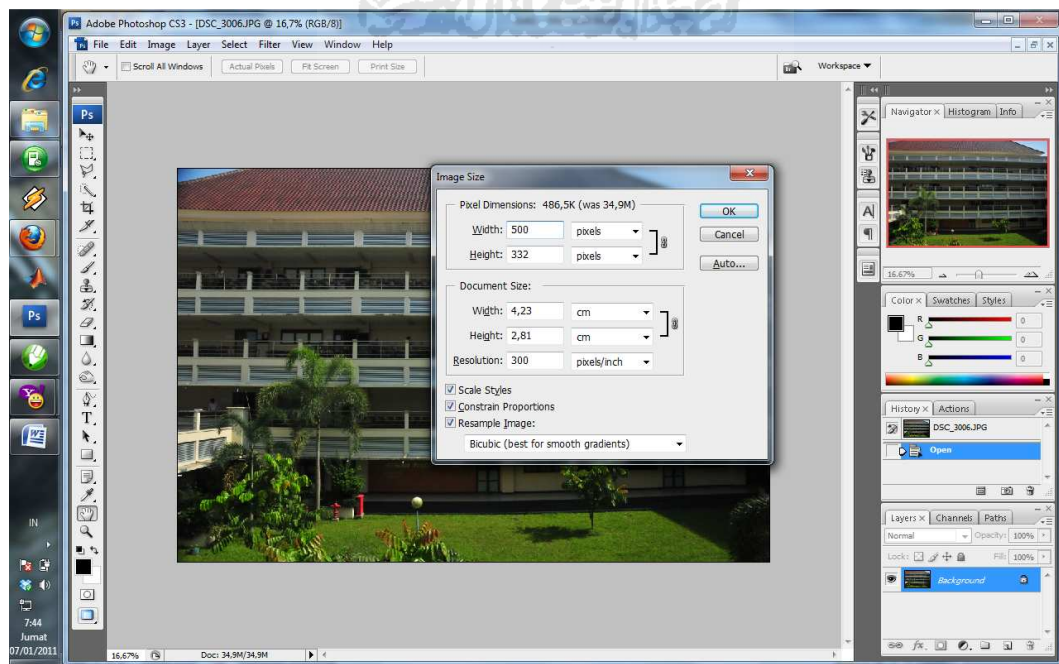


Gambar 4.3 Depth of field

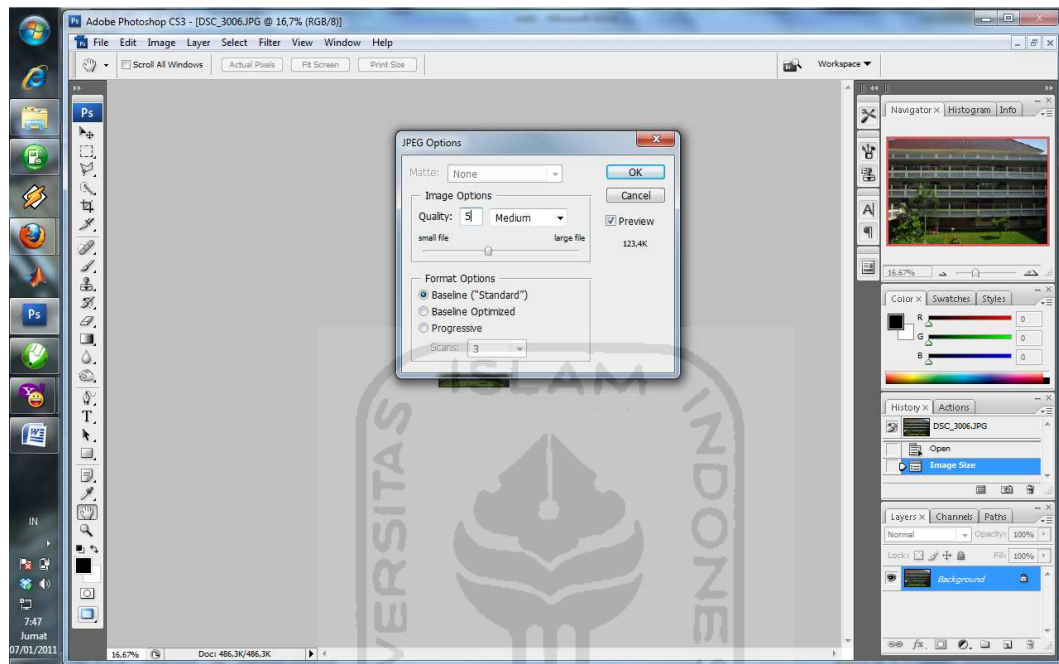
2. Pengecilan Ukuran Gambar

Pengecilan ukuran gambar harus dilakukan karena batas ukuran gambar yang diperbolehkan hanya dibawah 80kb saja. Pengecilan gambar dilakukan dengan cara manual. Yaitu menggunakan Adobe Photoshop CS3. Ada banyak cara untuk mengecilkan gambar, salah satunya seperti diterangkan diatas, yaitu menggunakan Adobe Photoshop CS3. Adobe Photoshop CS3 sendiri merupakan program untuk mengolah image atau foto untuk mendapatkan hasil yang lebih baik.

Pengecilan ukuran gambar secara manual dengan photoshop cs3 dapat ditentukan ukuran gambarnya. Sedangkan untuk kualitas gambar yang berkurang dapat diatasi dengan menaikkan kualitas gambar tanpa menambah beban ukuran image atau gambar.



Gambar 4.4 Proses pengecilan gambar



Gambar 4.5 Proses pemberian kualitas gambar



Gambar 4.6 Gambar sebelum melalui proses image rezising



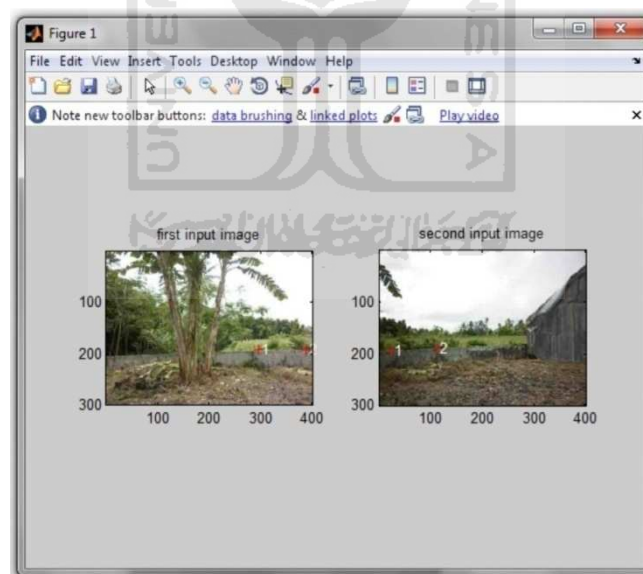
Gambar 4.7 Gambar setelah melalui proses image rezising

4.2 Hasil pengujian Virtual Fotografi Berbasis Image Mosaicing Dengan Metode Transformasi Geometric

Pada prinsipnya untuk mendeteksi kesamaan suatu objek, ditentukan dengan sama tidaknya komponen warna yang dikandung. Metode ini akan menjumlahkan nilai greyscale per kolom untuk tiap periode kolom range yang ditentukan. Luasan kolom range yang pernah diuji adalah 30 dan 50 kolom. Secara teknis sistem akan menjumlahkan nilai greyscale per kolom pada image pertama mulai dari kolom ke 0 sampai kolom ke range kemudian dibandingkan dengan jumlah greyscale per kolom dari image kedua mulai dari kolom ke 0 sampai kolom ke range. Perbandingan selanjutnya dilakukan terhadap kolom 1 sampai dengan kolom ke $\text{range}+1$ image pertama dengan kolom ke 0 sampai kolom ke range image kedua. Begitu seterusnya sampai dengan lebar image pertama. Dari keseluruhan perbandingan tersebut dicari selisih paling minimal untuk menentukan letak kolom korepondensi.

Algoritma ini bisa dikatakan belum optimal untuk dijadikan acuan dalam menentukan kolom korespondensi dengan histogram proyeksi, yang bisa kita simpulkan dari algoritma ini adalah:

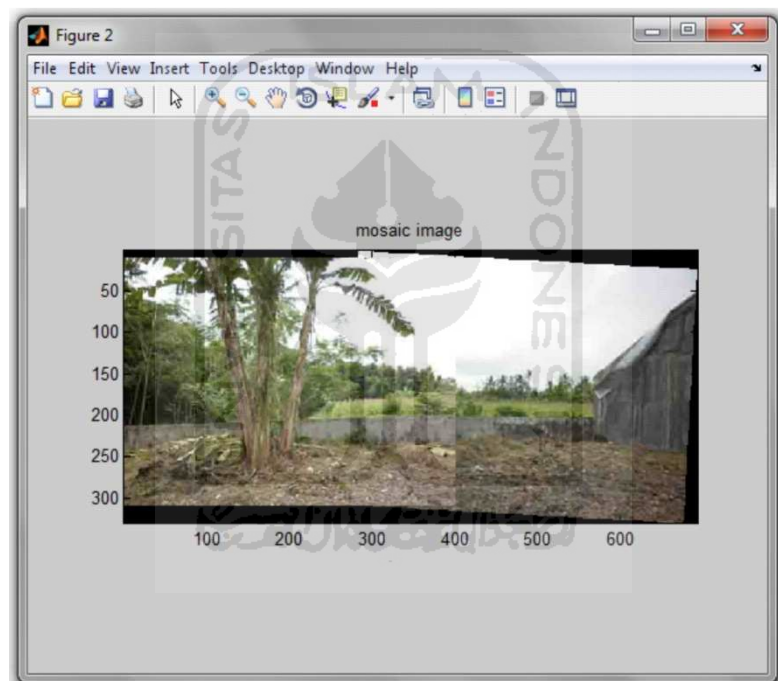
1. Algoritma hanya berfungsi dengan optimal pada beberapa masukan data saja, seperti untuk gambar masukan 1 dan 2 juga gambar masukan 2 dan 3.
2. Letak kolom korespondensi yang benar sulit didapat karena dengan membandingkan penjumlahan nilai greyscale per luasan kolom range secara khusus tidak bisa dijadikan acuan sama tidaknya suatu gambar walaupun memang dalam studi kasus ini menempatkan warna sebagai acuan utama untuk menentukan kesamaan obyek.
3. Perbesaran luasan range yang dipilih relatif lebih bagus tetapi perlu juga dipertimbangkan waktu komputasi yang dibutuhkan.



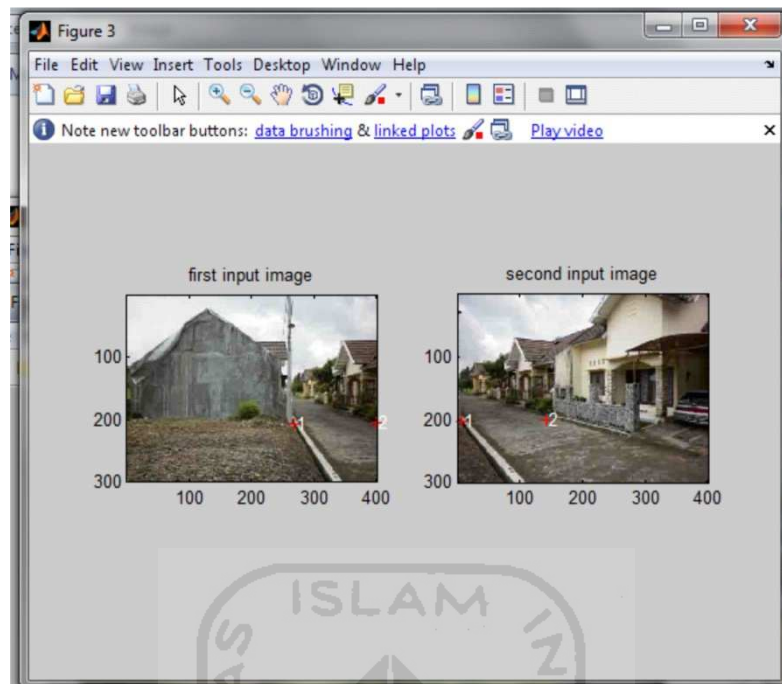
Gambar 4.8 Figure image mosaicing figure 1 step 1

Tanda merah menunjukkan perbandingan minimal yang menjadi acuan letak kolom korepondensi. Algoritma ini bisa dikatakan paling optimal apabila dibandingkan dengan algoritma- algoritma lain yang pernah diterapkan.

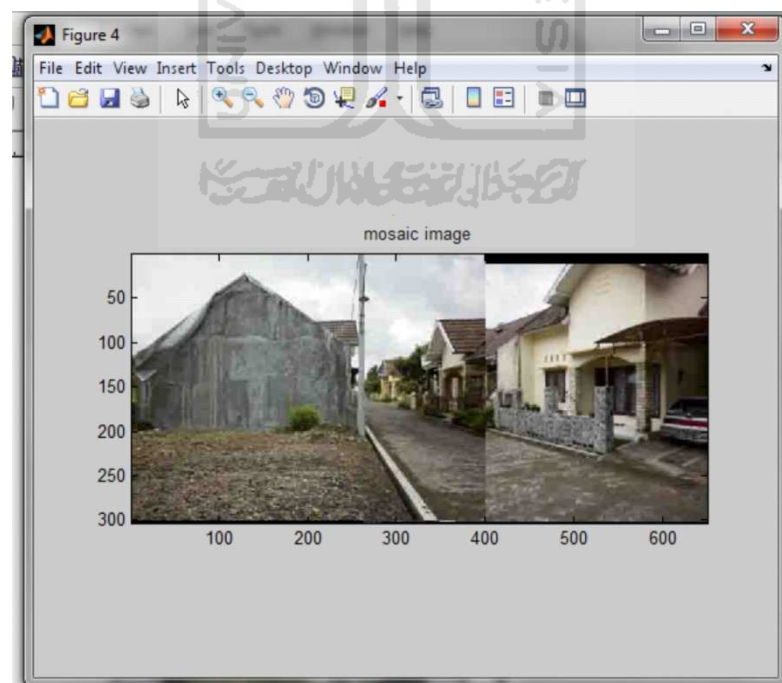
Mampu menentukan dengan benar letak titik korespondensi pada semua masukan data yang diberikan. Salah satu kelemahan dari metode ini adalah kurang rentan terhadap perubahan intensitas cahaya pada image sekuensial(masukan image). Masalah ini adalah masalah krusial pada saat pengambilan data karena perubahan intensitas cahaya otomatis akan merubah pula pada intensitas warna yang dikandung suatu pixel.



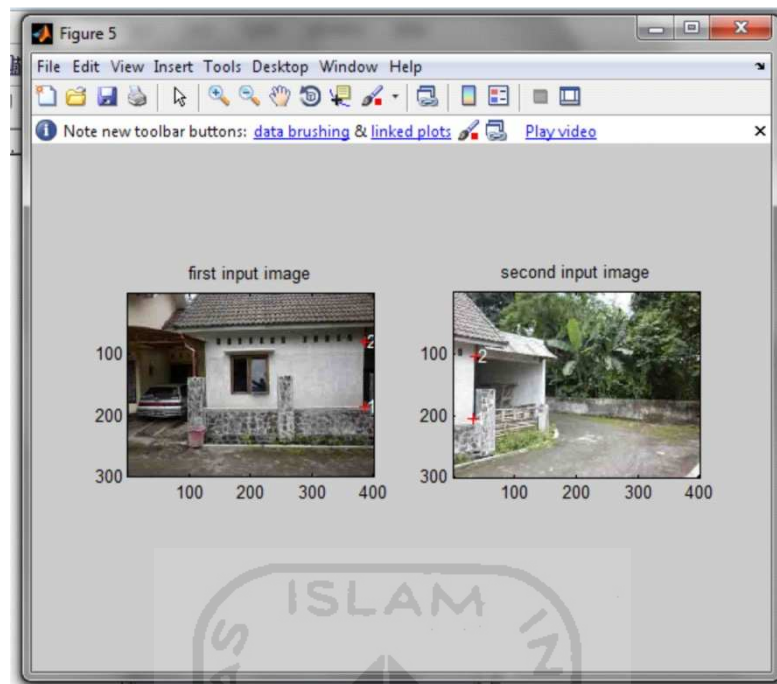
Gambar 4.9 Figure hasil image mosaicing figure 1 step 1



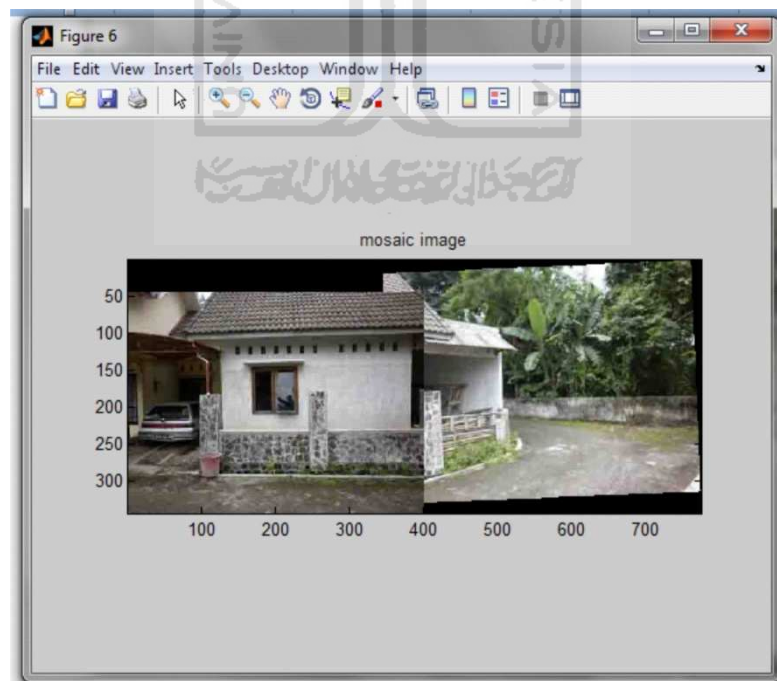
Gambar 4.10 Figure image mosaicing figure 3 step 1



Gambar 4.11 Figure hasil image mosaicing figure 4 step 1



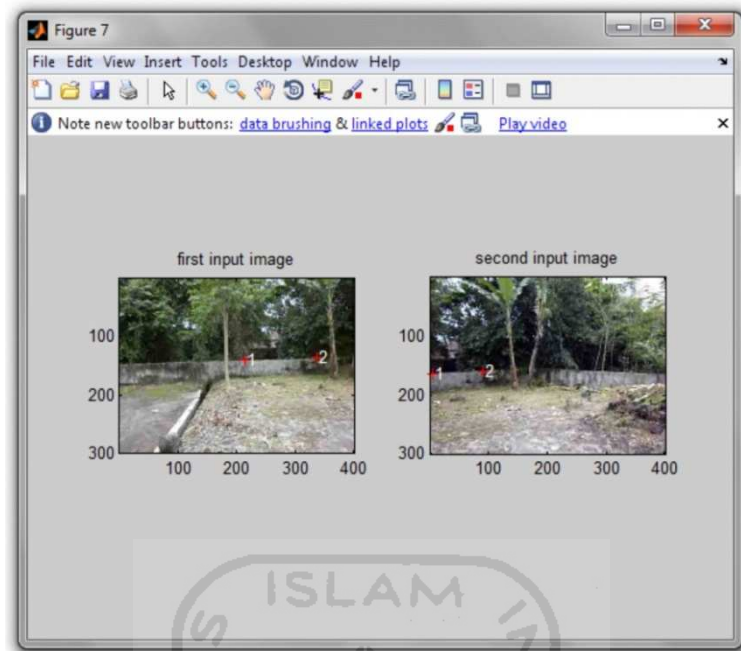
Gambar 4.12 Figure image mosaicing figure 5 step 1



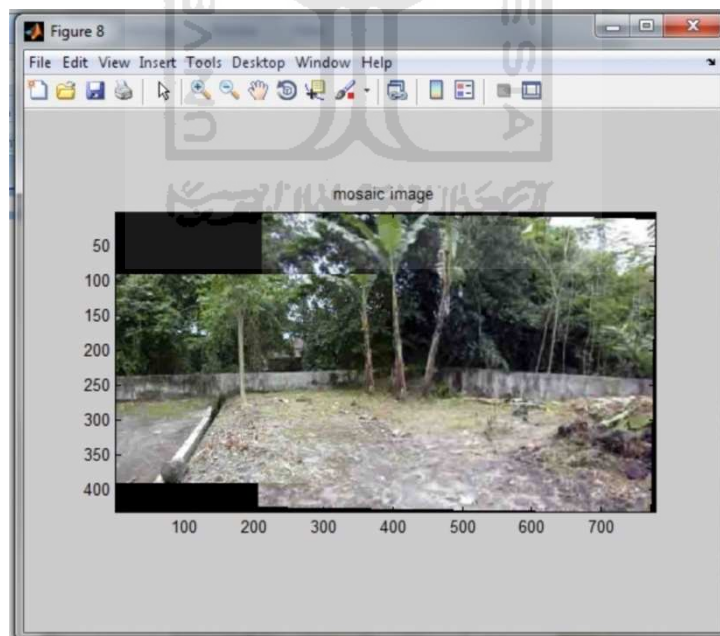
Gambar 4.13 Figure hasil image mosaicing figure 6 step 1

Terjadi kesalahan dalam menentukan letak kolom korespondensi, seperti yang telah dijelaskan sebelumnya pada bab II, dasar teori, ciri suatu objek dapat dikenali melalui 3 komponen yang dimiliki yakni warna, bentuk dan tekstur. Metode ini hanya menjadikan warna sebagai acuan untuk menentukan sama atau tidaknya suatu objek tanpa melibatkan ciri image yang lainnya. Ketika suatu objek atau gambar berubah nilai kecerahan atau intensitas warnanya maka perubahan ini juga berdampak pada perubahan komponen warna (merah, hijau, dan biru) yang dimiliki oleh sebuah image. Pada akhirnya perubahan itu berdampak pada kesalahan dalam menentukan letak kolom korespondensi. Kondisi ini pada umumnya terjadi pada saat pengambilan objek alam dengan cuaca yang sering berubah secara drastis.

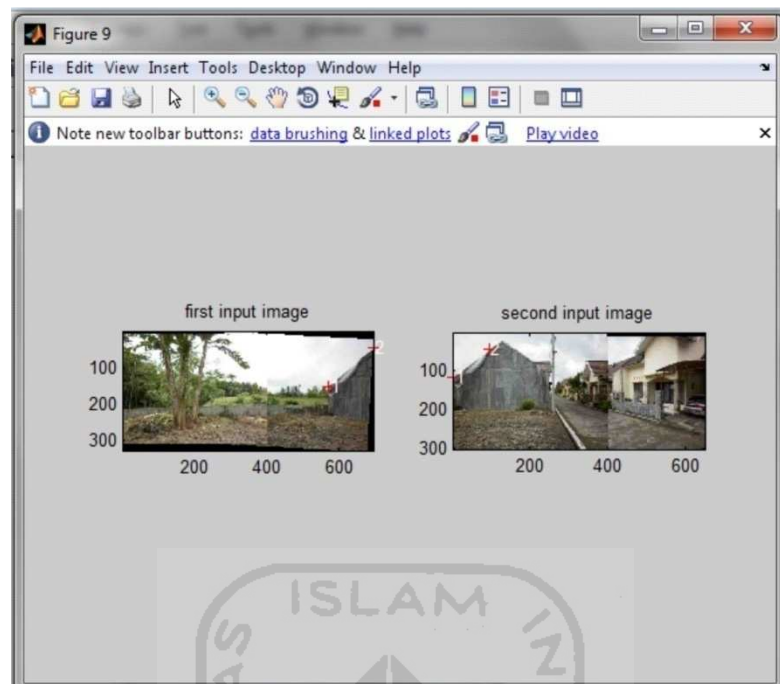
Selain rentan terhadap perubahan intensitas warna metode ini juga rentan terhadap perubahan titik aperture (titik proyeksi pengambilan gambar yang dihasilkan ketika mengambil gambar tidak dengan penompang yang stabil seperti tripod melainkan menggunakan tangan secara manual) terlalu besar sehingga tidak bisa dikenali sebagai kolom yang sama.



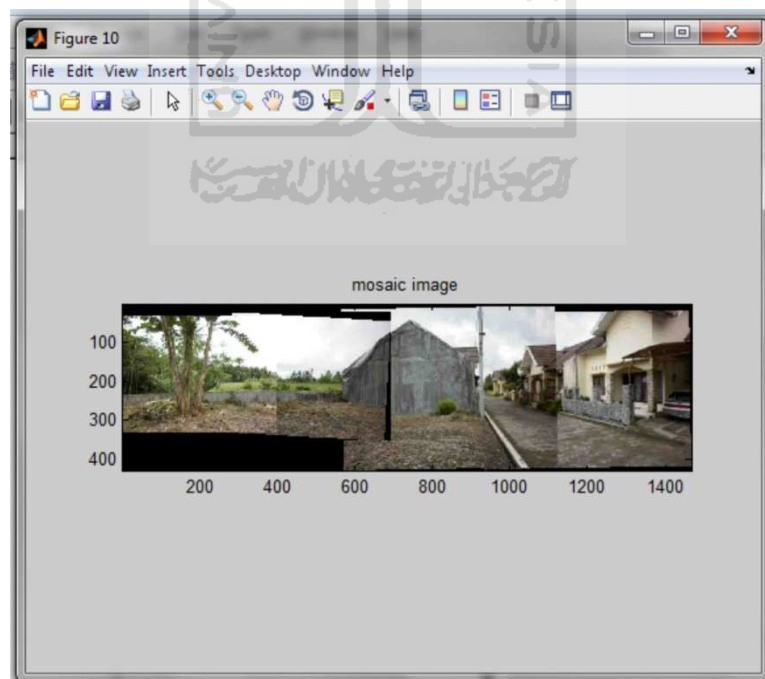
Gambar 4.14 Figure image mosaicing figure 7 step 1



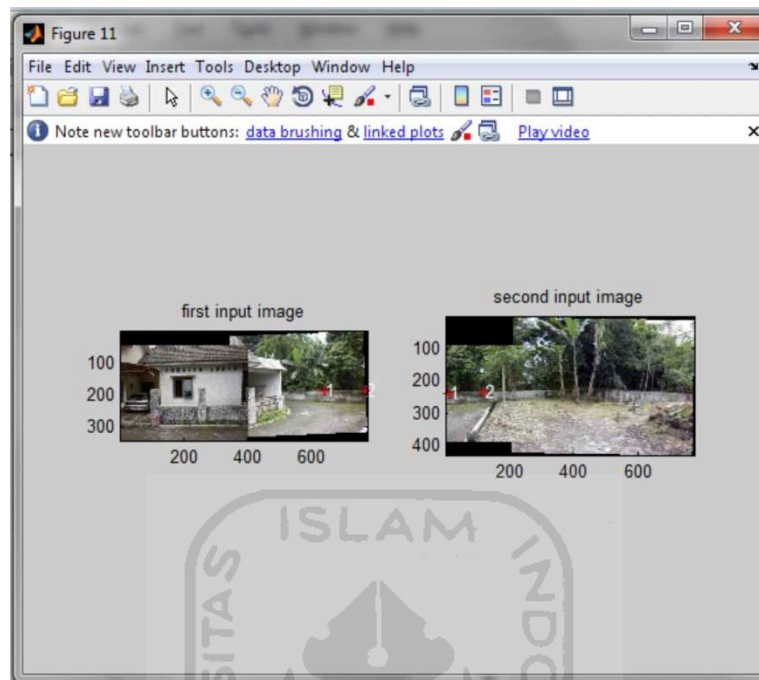
Gambar 4.15 Figure hasil image mosaicing figure 8 step 1



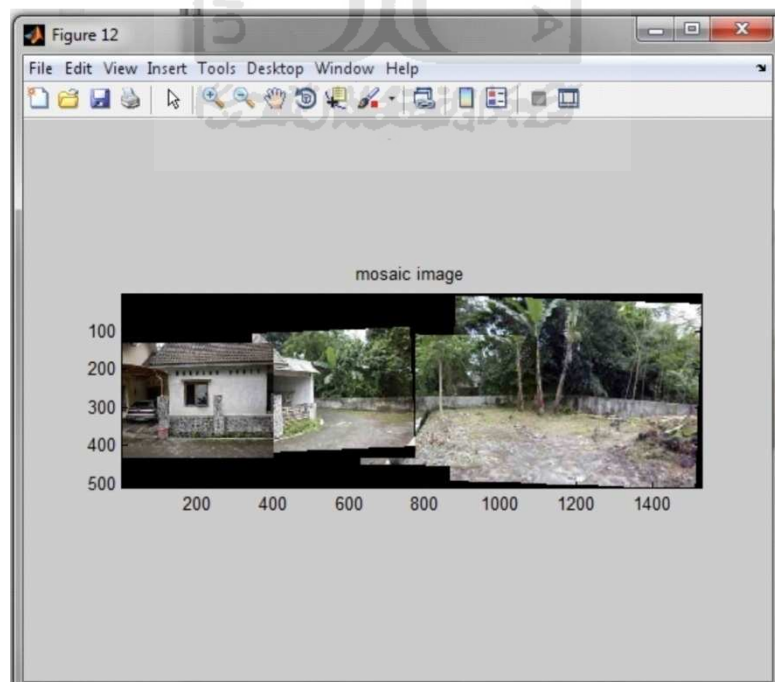
Gambar 4.16 Figure image mosaicing figure 9 step 2



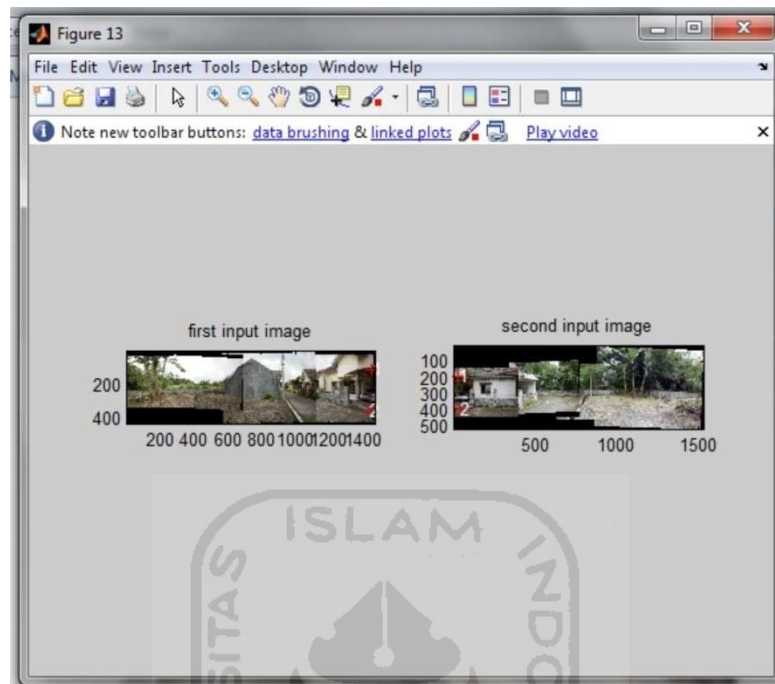
Gambar 4.17 Figure hasil image mosaicing figure 10 step 2



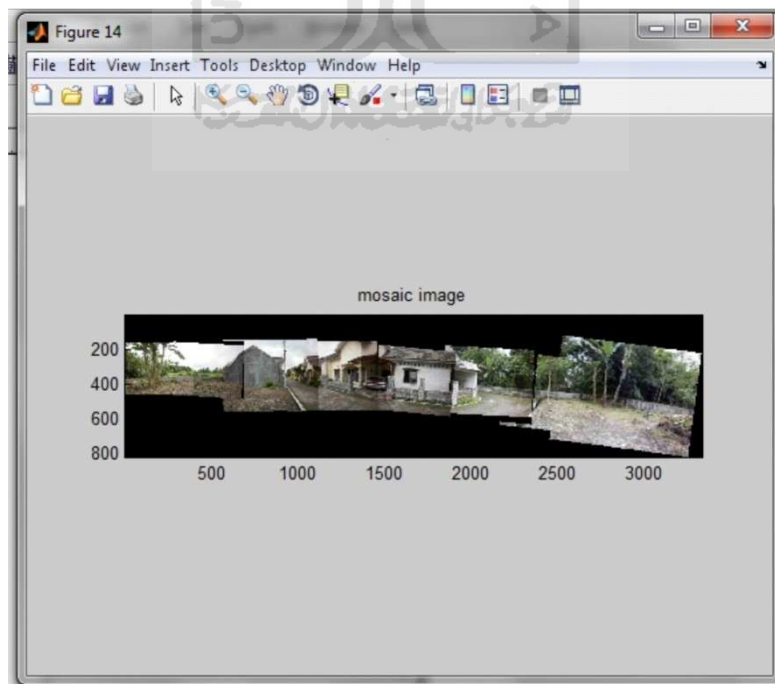
Gambar 4.18 Figure image mosaicing figure 11 step 2



Gambar 4.19 Figure hasil image mosaicing figure 12 step 2



Gambar 4.20 Figure image mosaicing figure 13 step 3



Gambar 4.21 Figure hasil image mosaicing figure 14 step 3

Hasil yang tidak bagus untuk objek taman antara lain dipengaruhi oleh penentuan letak kolom korepondensi yang kurang tepat dan analisa mengenai kesalahan tersebut sudah dijelaskan sebelumnya antara lain karena titik apperture yangterlalu besar, objek yang yang seragam (sebagian besar objek yang mirip), dan perbedaan intensitas cahaya). Alur kerja dari sistem mosaic panorama ini adalah suatu bentuk sistem yang sekuensial artinya berkelanjutan dari proses pertama diikuti oleh proses selanjutnya. Ketika kita sudah salah dari awal dalam menentukan kolom korepondensi maka akan berakibat fatal pada hasil overlay image 2 terhadap image 1.

4.3 Cropping Gambar

Cropping gambar berfungsi untuk memotong gambar yang kurang rata. Cropping gambar dilakukan menggunakan Adobe Photoshop cs3 secara manual. Dapat juga dilakukan menggunakan Program olah gambar Corel Draw x3.

Cropping gambar dapat juga dilakukan dengan PHP atau ACDsee. Namun cropping gambar lebih mudah dilaukan dengan adaobe photoshop atau corel draw, karena merupakan program umum yang mudah digunakan oleh orang awam.



Gambar 4.22 contoh hasil cropping

4.4 Hasil Akhir

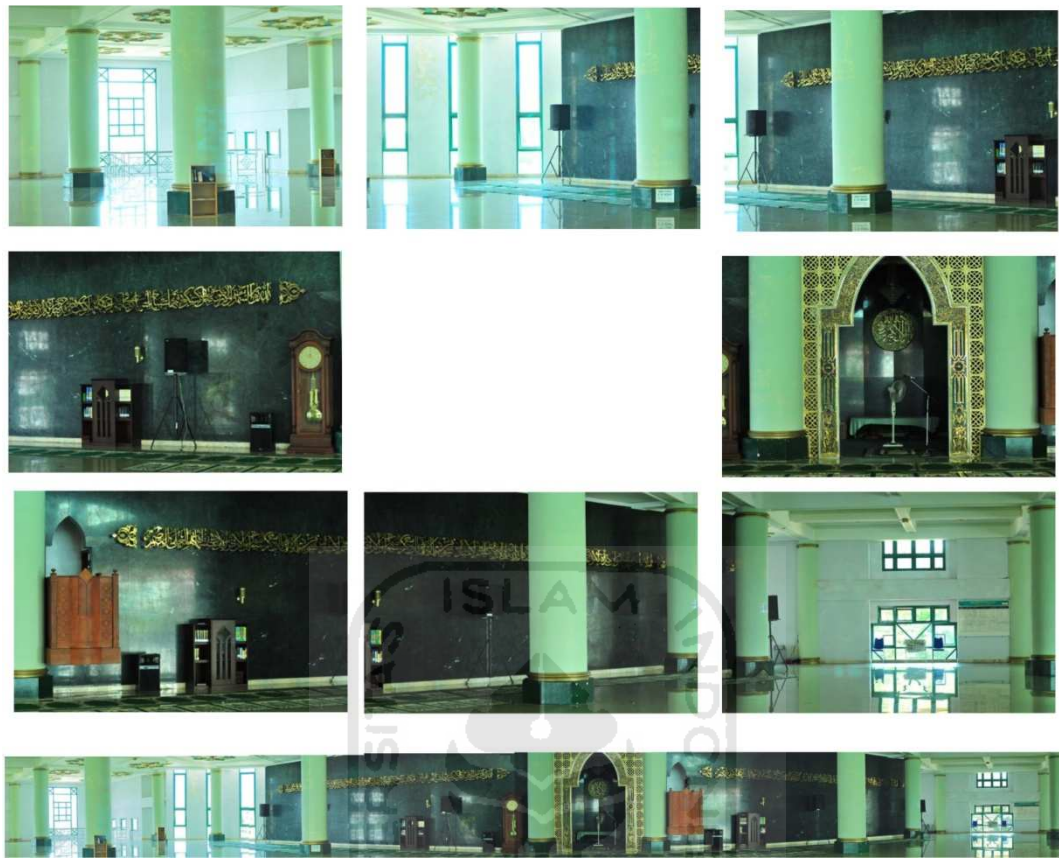


Gambar 4.22 Contoh hasil akhir image mosaicing di perumahan prumpung



Gambar 4.23 Contoh hasil akhir image mosaicing di Kampus FTI
menggunakan lensa 18mm

Gambar 4.22 dan 4.23 merupakan hasil image mosaicing dengan pengambilan gambar menggunakan lensa 18mm. Karena memiliki sudut perspektif tinggi maka mengakibatkan melengkungnya sudut gambar, sehingga sulit menemukan sudut yang sama untuk meletakkan titik point.



Gambar 4.24 Contoh hasil akhir image mosaicing di Masjid Ulil Albab



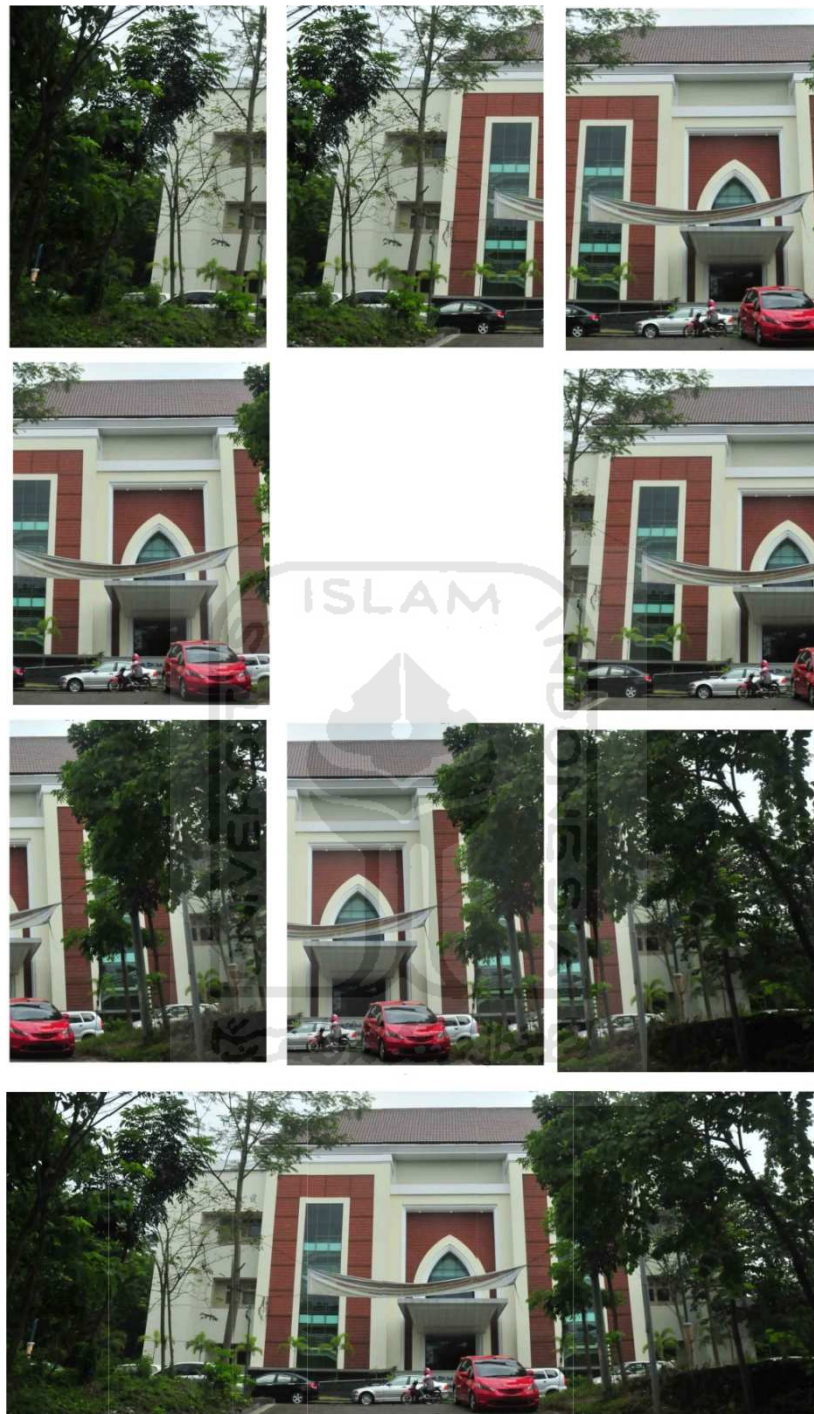
Gambar 4.25 Contoh hasil akhir image mosaicing di Masjid Ulil Albab dari luar



Gambar 4.26 Contoh hasil akhir image mosaicing di Gedung FIAI



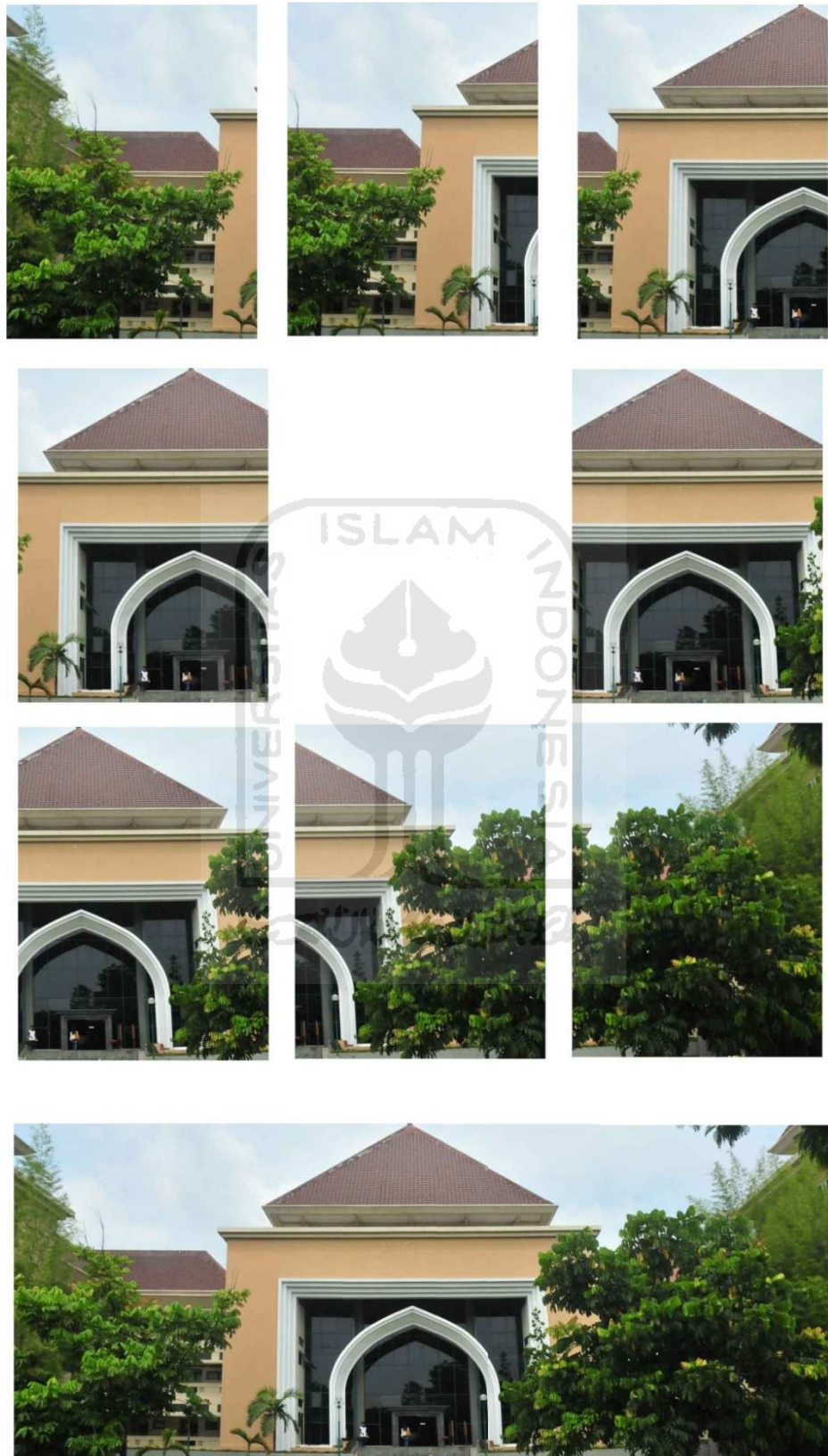
Gambar 4.27 Contoh hasil akhir image mosaicing di Gedung FTI



Gambar 4.28 Contoh hasil akhir image mosaicing di Gedung Sarjito



Gambar 4.29 Contoh hasil akhir image mosaicing di Gedung GOR UII



Gambar 4.30 Contoh hasil akhir image mosaicing di Gedung MIPA



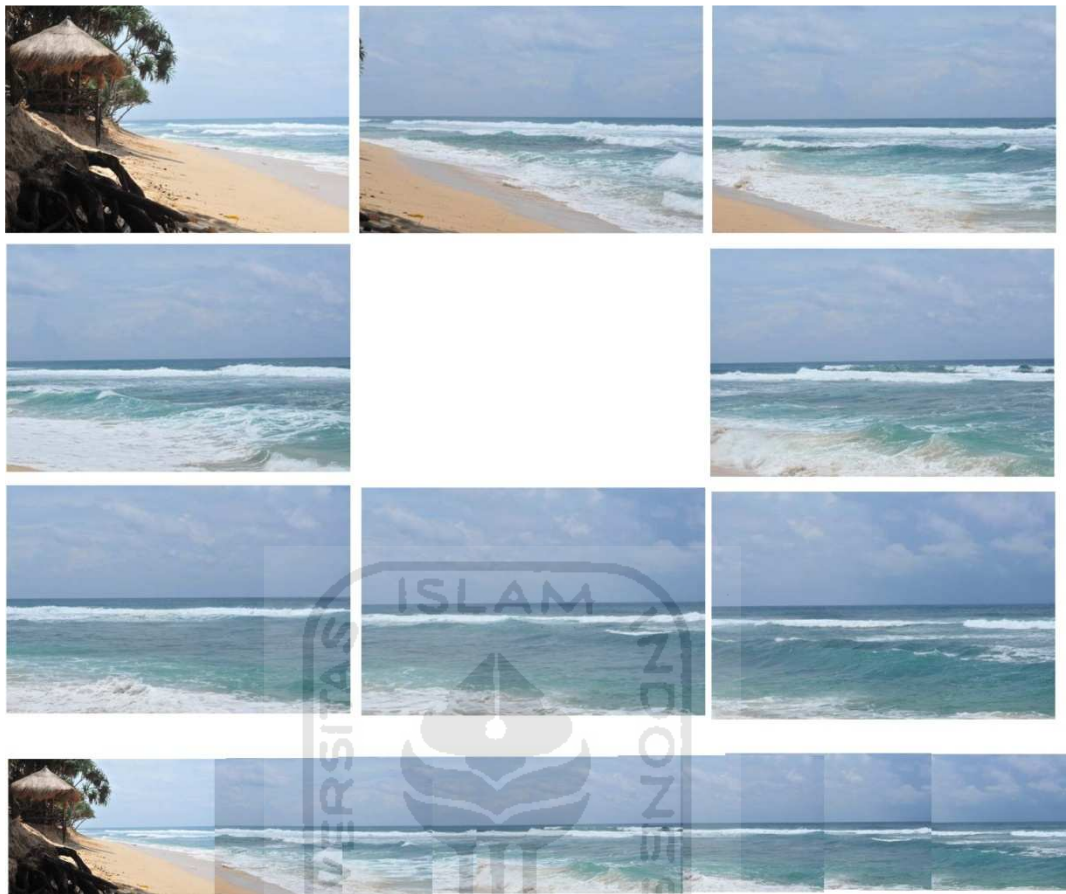
Gambar 4.31 Hasil Image Mosaicing menggunakan lensa nikkor 50mm di Masjid Mataram Kotagede



Gambar 4.32 Hasil Image Mosaicing menggunakan lensa nikkor 50mm di candi

Boko Prambanan

Gambar 4.24 sampai dengan gambar 4.32 merupakan hasil dari image mosaicing dengan pengambilan gambar menggunakan lensa 50mm yang memiliki sudut perspektif mendekati lensa mata manusi. Sehingga mampu mengurangi sudut lengkung gambar atau distorsi gambar. Sehingga mudah untuk meletakan titik poin image mosaicing.



Gambar 4.33 Gambar image mosaicing dengan gambar bergerak

Gambar 4.33 merupakan hasil dari image mosaicing dengan objek benda bergerak. Benda bergerak memiliki letak atau bentuk berbeda pada setiap gambarnya. Sehingga sangat sulit menentukan kesamaan sudut pada setiap gambarnya.



Gambar 4.34 Gambar image mosaicing dengan pengambilan bergerak

Gambar 4.34 merupakan hasil penggabungan gambar dengan metode pengambilan gambar bergerak secara konstan. Sehingga didapatkan hasil gambar terkesan seperti apa yang kita lihat. Tetapi karena ada distorsi gambar pada lensa sehingga sulit menggabungkan gambar secara virtual.

BAB V

PENUTUP

5.1 Kesimpulan

Pada bagian ini akan diulas tentang kesimpulan dari seluruh percobaan dan pengujian sistem perangkat lunak yang telah dibuat. Berikut beberapa kesimpulan yang dapat diambil dari percobaan dan pengujian konstruksi image mosaic menggunakan transformasi geometrik

- Optimal tidaknya image mosaic yang dihasilkan selain dipengaruhi oleh algoritma yang digunakan juga dipengaruhi oleh masukan datanya. Masukan data yang masih bisa ditoleransi untuk menghasilkan image mosaic yang optimal adalah masukan data yang antara framenya tidak begitu banyak terdapat perubahan intensitas warna dan juga perubahan besar sudut aperture yang terlalu besar.
- Pengambilan gambar benda bergerak didapatkan hasil yang tidak beraturan, karena pada setiap pengambilan gambar, benda akan berpindah tempat. Sehingga tidak mendapatkan kecocokan gambar.
- Metode pengambilan gambar secara bergerak memberikan gambar terkesan bergerak. Tetapi karena adanya distorsi lensa, sehingga merubah sudut pandang kamera.
- Penggunaan ukuran lensa sangat berpengaruh dengan hasil image mosaicing. lensa paling cocok untuk image mosaicing ini adalah lensa

nikkor 50 mm, karena lensa ini memiliki sudut pandang paling mirip dengan mata manusia.

- Penggunaan lensa wide angle kurang cocok digunakan karena adanya sudut perspektif gambar, sehingga sulit mencari kesamaan sudut dari setiap gambarnya.

5.2 Saran

Untuk pengembangan lebih lanjut, diharapkan dapat menggunakan metode yang lain yang secara umum bisa :

1. Mengatasi variasi data seperti perubahan intensitas warna dan titik aperture. sehingga hasil image mosaiknya lebih bagus lagi.
2. Mampu mengenali objek panorama sehingga masukan datanya tidak perlu harus image yang sekuensial.

DAFTAR PUSTAKA

- [1] Aaland, Mikkell. Pro Tips for Taking Great Pictures with Your Digital Camera.
Available at :
<http://www.vrphotography.com/data/pages/pdfs/ShootingDigitalCh9.pdf>
- [2] Alexander,Paul. *Pembentukan Citra Panotama 360 Dengan Image Mozaicing*.
Fakultas Teknologi Industri, Universitas Kristen Petra
- [3] Bachelder,Chris. Lessons in Virtual Tour Photography. Available at
[:www.mcsweeneys.net/bachelder/bachelder.pdf](http://www.mcsweeneys.net/bachelder/bachelder.pdf).
- [4] Geometra - 3D from Images for Geeks. Available at :
<http://www.sandyressler.com/about/library/weekly/aa092999.htm>. Diakses
tanggal 24 September 2009
- [5] Immersive Imaging Panoramas. Available at :
<http://www.arjayonline.com/pano.htm>.Diakses tanggal 24 September 2009.
- [6] T. L. Editors, *Life Library of Photography: The Camera*, 1970.
- [7] Wikipedia. Panoramic Photopraphy. Available at :http://en.wikipedia.org/wiki/Panoramic_photography. Diakses tanggal 4 Oktober 2009.
- [8] Wikipedia. VR Photopraphy. Available at :http://en.wikipedia.org/wiki/vr_photography. Diakses Tanggal 4 Oktober 2009.

LAMPIRAN

Listing Program

```
function varargout = imac(varargin)
% IMAC M-file for imac.fig
%
%   IMAC, by itself, creates a new IMAC or raises the existing
%   singleton*.
%
%
%   H = IMAC returns the handle to a new IMAC or the handle to
%   the existing singleton*.
%
%
%   IMAC('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in IMAC.M with the given input
arguments.
%
%
%   IMAC('Property','Value',...) creates a new IMAC or raises
the
%   existing singleton*. Starting from the left, property
value pairs are
%   applied to the GUI before imac_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to imac_OpeningFcn via
varargin.
%
```

```

%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help imac

% Last Modified by GUIDE v2.5 07-Jan-2011 08:06:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @imac_OpeningFcn, ...
                  'gui_OutputFcn',  @imac_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before imac is made visible.

function imac_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of
MATLAB

% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to imac (see VARARGIN)

% Choose default command line output for imac
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes imac wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = imac_OutputFcn(hObject, eventdata, handles)

% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```
% handles      structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
```

```
varargout{1} = handles.output;
```

```
function namaimagetext_Callback(hObject, eventdata, handles)
```

```
% hObject      handle to namaimagetext (see GCBO)
```

```
% eventdata    reserved - to be defined in a future version of  
MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of namaimagetext  
as text
```

```
%             str2double(get(hObject,'String')) returns contents of  
namaimagetext as a double
```

```
% --- Executes during object creation, after setting all  
properties.
```

```
function namaimagetext_CreateFcn(hObject, eventdata, handles)
```

```
% hObject      handle to namaimagetext (see GCBO)
```

```
% eventdata    reserved - to be defined in a future version of  
MATLAB
```

```
% handles      empty - handles not created until after all  
CreateFcns called
```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in image1.
function image1_Callback(hObject, eventdata, handles)
% hObject    handle to image1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.G1=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);
% --- Executes on button press in image2.
function image2_Callback(hObject, eventdata, handles)
% hObject    handle to image2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.G2=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);
% --- Executes on button press in image5.
function image5_Callback(hObject, eventdata, handles)
% hObject    handle to image5 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of
MATLAB

% handles structure with handles and user data (see GUIDATA)
handles.G5=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);

% --- Executes on button press in image4.
function image4_Callback(hObject, eventdata, handles)
% hObject handle to image4 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.G4=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);
% --- Executes on button press in image3.
function image3_Callback(hObject, eventdata, handles)
% hObject handle to image3 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.G3=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);

% --- Executes on button press in image6.
function image6_Callback(hObject, eventdata, handles)
% hObject handle to image6 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

handles.G6=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);

% --- Executes on button press in image7.
function image7_Callback(hObject, eventdata, handles)
% hObject    handle to image7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.G7=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);

% --- Executes on button press in image8.
function image8_Callback(hObject, eventdata, handles)
% hObject    handle to image8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.G8=double(imread(get(handles.namaimagetext,'string')));
Guidata(hObject,handles);

% --- Executes on button press in imageproses.
function imageproses_Callback(hObject, eventdata, handles)
% hObject    handle to imageproses (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
G1=handles.G1

```

```

G2=handles.G2
G3=handles.G3
G4=handles.G4
G5=handles.G5
G6=handles.G6
G7=handles.G7
G8=handles.G8
for k=1:4
    if k==1
        I1=G1;
        I2=G2;
    elseif k==2
        I1=G3;
        I2=G4;
    elseif k==3
        I1=G5;
        I2=G6;
    elseif k==4
        I1=G7;
        I2=G8;
    end
end

```



```

% load input images
[h1 w1 d1] = size(I1);
[h2 w2 d2] = size(I2);

```

```

% show input images and prompt for correspondences
figure; subplot(1,2,1); image(I1/255); axis image; hold on;
title('first input image');
[X1 Y1] = ginput(2); % get two points from the user
subplot(1,2,2);image(I2/255); axis image; hold on;
title('second input image');
[X2 Y2] = ginput(2); % get two points from the user

% estimate parameter vector (t)
Z = [ X2' Y2' ; Y2' -X2' ; 1 1 0 0 ; 0 0 1 1 ]';
xp = [ X1 ; Y1 ];
t = Z \ xp; % solve the linear sistem
a = t(1); % = s cos(alpha)
b = t(2); % = s sin(alpha)
tx = t(3);
ty = t(4);

% construct transformation matrix (T)
T = [a b tx ; -b a ty ; 0 0 1];

% warp incoming corners to determine the size of the output image
(in to out)
cp = T*[ 1 1 w2 w2 ; 1 h2 1 h2 ; 1 1 1 1 ];
Xpr = min( [ cp(1,:) 0 ] ) : max( [cp(1,:) w1] ); % min x : max x
Ypr = min( [ cp(2,:) 0 ] ) : max( [cp(2,:) h1] ); % min y : max y
[Xp,Yp] = ndgrid(Xpr,Ypr);
[wp hp] = size(Xp); % = size(Yp)

```

```

% do backwards transform (from out to in)
X = T \ [ Xp(:) Yp(:) ones(wp*hp,1) ]'; % warp

% re-sample pixel values with bilinear interpolation
clear Ip;
xI = reshape( X(1,:),wp,hp)';
yI = reshape( X(2,:),wp,hp)';
Ip(:, :,1) = interp2(I2(:, :,1), xI, yI, '*bilinear'); % red
Ip(:, :,2) = interp2(I2(:, :,2), xI, yI, '*bilinear'); % green
Ip(:, :,3) = interp2(I2(:, :,3), xI, yI, '*bilinear'); % blue

% offset and copy original image into the warped image
offset = -round( [ min( [ cp(1,:) 0 ] ) min( [ cp(2,:) 0 ] ) ] );
Ip(1+offset(2):h1+offset(2),1+offset(1):w1+offset(1),:) =
double(I1(1:h1,1:w1,:));

% show the result
figure; image(Ip/255); axis image;
title('mosaic image');
if k==1
    Ip1=Ip;
elseif k==2
    Ip2=Ip;
elseif k==3
    Ip3=Ip;
elseif k==4
    Ip4=Ip;

```

```

end

end

for k=1:2

    if k==1

        I1=Ip1;

        I2=Ip2;

    elseif k==2

        I1=Ip3;

        I2=Ip4;

    end

    % load input images
[h1 w1 d1] = size(I1);
[h2 w2 d2] = size(I2);

% show input images and prompt for correspondences
figure; subplot(1,2,1); image(I1/255); axis image; hold on;
title('first input image');

[X1 Y1] = ginput2(2); % get two points from the user
subplot(1,2,2); image(I2/255); axis image; hold on;
title('second input image');

[X2 Y2] = ginput2(2); % get two points from the user

% estimate parameter vector (t)
Z = [ X2' Y2' ; Y2' -X2' ; 1 1 0 0 ; 0 0 1 1 ]';
xp = [ X1 ; Y1 ];
t = Z \ xp; % solve the linear sistem
a = t(1); % = s cos(alpha)

```

```

b = t(2); % = s sin(alpha)
tx = t(3);
ty = t(4);

% construct transformation matrix (T)
T = [a b tx ; -b a ty ; 0 0 1];

% warp incoming corners to determine the size of the output image
(in to out)
cp = T*[ 1 1 w2 w2 ; 1 h2 1 h2 ; 1 1 1 1 ];
Xpr = min( [ cp(1,:) 0 ] ) : max( [cp(1,:) w1] ); % min x : max x
Ypr = min( [ cp(2,:) 0 ] ) : max( [cp(2,:) h1] ); % min y : max y
[Xp,Yp] = ndgrid(Xpr,Ypr);
[wp hp] = size(Xp); % = size(Yp)

% do backwards transform (from out to in)
X = T \ [ Xp(:) Yp(:) ones(wp*hp,1) ]'; % warp

% re-sample pixel values with bilinear interpolation
clear Ip;
xI = reshape( X(1,:),wp,hp)';
yI = reshape( X(2,:),wp,hp)';
Ip(:,:,1) = interp2(I2(:,:,1), xI, yI, '*bilinear'); % red
Ip(:,:,2) = interp2(I2(:,:,2), xI, yI, '*bilinear'); % green
Ip(:,:,3) = interp2(I2(:,:,3), xI, yI, '*bilinear'); % blue

% offset and copy original image into the warped image

```

```
offset = -round( [ min( [ cp(1,:) 0 ] ) min( [ cp(2,:) 0 ] ) ] );
Ip(1+offset(2):h1+offset(2),1+offset(1):w1+offset(1),:) =
double(I1(1:h1,1:w1,:));
```

```
% show the result
```

```
figure; image(Ip/255); axis image;
title('mosaic image');
```

```
if k==1
```

```
    Ipp1=Ip;
```

```
elseif k==2
```

```
    Ipp2=Ip;
```

```
end
```

```
end
```

```
    I1 = Ipp1;
```

```
    I2 = Ipp2;
```

```
[h1 w1 d1] = size(I1);
```

```
[h2 w2 d2] = size(I2);
```

```
% show input images and prompt for correspondences
```

```
figure; subplot(1,2,1); image(I1/255); axis image; hold on;
```

```
title('first input image');
```

```
[X1 Y1] = ginput2(2); % get two points from the user
```

```
subplot(1,2,2); image(I2/255); axis image; hold on;
```

```
title('second input image');
```



```

[X2 Y2] = ginput2(2); % get two points from the user

% estimate parameter vector (t)
Z = [ X2' Y2' ; Y2' -X2' ; 1 1 0 0 ; 0 0 1 1 ]';
xp = [ X1 ; Y1 ];
t = Z \ xp; % solve the linear sistem
a = t(1); % = s cos(alpha)
b = t(2); % = s sin(alpha)
tx = t(3);
ty = t(4);

% construct transformation matrix (T)
T = [a b tx ; -b a ty ; 0 0 1];

% warp incoming corners to determine the size of the output image
(in to out)
cp = T*[ 1 1 w2 w2 ; 1 h2 1 h2 ; 1 1 1 1 ];
Xpr = min( [ cp(1,:) 0 ] ) : max( [cp(1,:) w1] ); % min x : max x
Ypr = min( [ cp(2,:) 0 ] ) : max( [cp(2,:) h1] ); % min y : max y
[Xp,Yp] = ndgrid(Xpr,Ypr);
[wp hp] = size(Xp); % = size(Yp)

% do backwards transform (from out to in)
X = T \ [ Xp(:) Yp(:) ones(wp*hp,1) ]'; % warp

% re-sample pixel values with bilinear interpolation
clear Ip;

```

```

xI = reshape( X(1,:),wp,hp)';
yI = reshape( X(2,:),wp,hp)';

Ip(:,:,1) = interp2(I2(:,:,1), xI, yI, '*bilinear'); % red
Ip(:,:,2) = interp2(I2(:,:,2), xI, yI, '*bilinear'); % green
Ip(:,:,3) = interp2(I2(:,:,3), xI, yI, '*bilinear'); % blue

% offset and copy original image into the warped image
offset = -round( [ min( [ cp(1,:) 0 ] ) min( [ cp(2,:) 0 ] ) ] );
Ip(1+offset(2):h1+offset(2),1+offset(1):w1+offset(1),:) =
double(I1(1:h1,1:w1,:));

% show the result
figure; image(Ip/255); axis image;
title('mosaic image');

% --- Executes on button press in figurehelp.
function figurehelp_Callback(hObject, eventdata, handles)

% hObject    handle to figurehelp (see GCBO)

% eventdata  reserved - to be defined in a future version of
MATLAB

% handles     structure with handles and user data (see GUIDATA)

```

Program Help

```

function varargout = help(varargin)

% HELP M-file for help.fig

%     HELP, by itself, creates a new HELP or raises the existing

```

```

%     singleton*.
%
%     H = HELP returns the handle to a new HELP or the handle to
%     the existing singleton*.
%
%     HELP('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in HELP.M with the given input
arguments.
%
%     HELP('Property','Value',...) creates a new HELP or raises
the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before help_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to help_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help help

```

```

% Last Modified by GUIDE v2.5 06-Jan-2011 06:56:41

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @help_OpeningFcn, ...
                  'gui_OutputFcn',    @help_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before help is made visible.
function help_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to help (see VARARGIN)


% Choose default command line output for help
handles.output = hObject;


% Update handles structure
guidata(hObject, handles);


% UIWAIT makes help wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = help_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)


% Get default command line output from handles structure
varargout{1} = handles.output;


% -----
-----

```

```
function uitable1_ButtonDownFcn(hObject, eventdata, handles)

% hObject    handle to uitable1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

