

**PENILAIAN KOMPOSISI *RULE OF THIRDS* PADA FOTOGRAFI  
MENGUNAKAN BANTUAN PENGOLAHAN CITRA DIGITAL**

**SKRIPSI**

**Konsentrasi Rekayasa Komputer**

Diajukan untuk memenuhi sebagian persyaratan

Memperoleh gelar Sarjana Teknik

*ajukan Untuk Memenuhi Sebagian Persyaratan Teknik*



**Disusun oleh:**

**TOMI PUTRO UTOMO**

**NIM. 0710633057 - 63**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK ELEKTRO**

**MALANG**

**2014**

**LEMBAR PERSETUJUAN**

**PENILAIAN KOMPOSISI *RULE OF THIRDS* PADA FOTOGRAFI  
MENGUNAKAN BANTUAN PENGOLAHAN CITRA DIGITAL**

**SKRIPSI**

**JURUSAN TEKNIK ELEKTRO**

Diajukan untuk memenuhi sebagian persyaratan

Memperoleh gelar Sarjana Teknik



Disusun oleh:

**TOMI PUTRO UTOMO**

**NIM. 0710633057 - 63**

Telah diperiksa dan disetujui oleh

Dosen Pembimbing:

**Pembimbing I**

**Pembimbing II**

**Adharul Muttaqin, S.T., M.T.**  
**NIP. 19760121 200501 1 001**

**Ir. Muhammad Aswin, M.T.**  
**NIP 19640626 199002 1 001**

**LEMBAR PENGESAHAN**

**PENILAIAN KOMPOSISI *RULE OF THIRDS* PADA FOTOGRAFI  
MENGUNAKAN BANTUAN PENGOLAHAN CITRA DIGITAL**

Disusun oleh:

**TOMI PUTRO UTOMO**  
**NIM. 0710633057 – 63**

Skripsi ini telah diuji dan dinyatakan lulus  
pada tanggal 12 Agustus 2014

Dosen penguji:

**Waru Djuriatno, S.T., M.T.**  
**NIP. 19690725 199702 1 001**

**Raden Arief Setyawan, S.T., M.T.**  
**NIP. 19750819 199903 1 001**

**Rahmadwati, S.T., M.T., Ph.D.**  
**NIP. 19771102 200604 2 003**

Mengetahui  
Ketua Jurusan Teknik Elektro

**Muhammad Aziz Muslim, S.T., M.T., Ph.D.**  
**NIP. 19741203 200012 1 001**



## PENGANTAR

Dengan mengucapkan Alhamdulillah Rabbil'alamin, puji syukur kepada Allah Subhānahu Wa Ta'Ala yang telah melimpahkan rahmat, ridho, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Penilaian Komposisi *Rule of Thirds* Pada Fotografi Menggunakan Bantuan Pengolahan Citra Digital” sebagai salah satu syarat yang diajukan untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya Malang.

Penulis menyadari bahwa penulisan skripsi ini tidak akan terselesaikan tanpa adanya bantuan dari berbagai pihak karena itu ucapan terima kasih penulis sampaikan kepada semua pihak yang telah dengan rela meluangkan waktunya untuk membantu penulis dalam proses penulisan skripsi dan juga dengan ikhlas memberikan bimbingan, arahan, masukan, dan semangat sehingga penulis dapat menyelesaikan skripsi ini. Oleh karena itu pada kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Bapak Muhammad Aziz Muslim, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya;
2. Bapak Hadi Suyono, S.T., M.T., Ph.D. selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya;
3. Bapak Mochammad Rif'an, S.T., M.T. selaku Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya;
4. Bapak Waru Djuriatno, S.T., M.T. selaku KKDK Rekayasa Komputer;
5. Bapak Adharul Muttaqin, S.T., M.T. selaku dosen pembimbing I serta Bapak Ir. Muhammad Aswin, M.T. selaku dosen pembimbing II yang telah memberikan bimbingan, masukan, dan arahan dalam penelitian skripsi ini;
6. Ibu Rusmi Ambarwati, S.T., M.T. selaku dosen penasihat akademik yang telah memberikan nasihat dan arahan dalam proses akademik penulis;
7. Bapak Sudiat Makno, Mama Suchaeni, Adik Reza Dwi Utomo, Adik Rizky Tri Utomo, dan Adik Tasya Rahma Fadillah atas segala dukungan, doa, serta dorongan semangat hingga terselesaikannya skripsi ini;
8. Rekan-rekan Laboratorium Informatika dan Komputer;

- repository.ub.ac.id
9. Rekan-rekan Riset Teknologi Informasi Elektro (RisTIE) periode kepengurusan 2009/2010 dan 2010/2011;
  10. Rekan-rekan Himpunan Mahasiswa Teknik Elektro periode kepengurusan 2009/2010 dan 2010/2011;
  11. Teman-teman Indekos Kertosentono 119;
  12. Teman-teman angkatan 2007 (CORE '07);
  13. Mas Norman Valentine Alpharian atas bantuan penilaian fotonya;
  14. Semua pihak yang telah membantu penyelesaian skripsi ini baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu;

Penulis berharap semoga skripsi ini bermanfaat bagi pembaca dan dapat memberikan sumbangan pemikiran bagi pihak-pihak lain khususnya mahasiswa Jurusan Teknik Elektro. Kritik dan saran yang membangun penulis terima untuk perbaikan skripsi ini di masa mendatang, karena penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan masih banyak kekurangan.

Malang, Agustus 2014

Penyusun



## DAFTAR ISI

|  |      |
|--|------|
| <b>PENGANTAR</b> .....                           | i    |
| <b>DAFTAR ISI</b> .....                          | iii  |
| <b>DAFTAR GAMBAR</b> .....                       | vi   |
| <b>DAFTAR TABEL</b> .....                        | vii  |
| <b>ABSTRAK</b> .....                             | viii |
| <b>BAB I PENDAHULUAN</b> .....                   | 1    |
| 1.1 Latar Belakang .....                         | 1    |
| 1.2 Rumusan Masalah .....                        | 2    |
| 1.3 Batasan Masalah .....                        | 2    |
| 1.4 Tujuan .....                                 | 3    |
| 1.5 Manfaat .....                                | 3    |
| 1.6 Sistematika Penulisan .....                  | 3    |
| <b>BAB II TINJAUAN PUSTAKA</b> .....             | 5    |
| 2.1 Fotografi.....                               | 5    |
| 2.2 Komposisi Fotografi .....                    | 5    |
| 2.3 Aturan Komposisi <i>Rule of Thirds</i> ..... | 6    |
| 2.4 Sampling .....                               | 8    |
| 2.4.1 Penarikan Sampel Acak Sederhana .....      | 8    |
| 2.5 <i>Centroid</i> .....                        | 9    |
| 2.5.1 Metode Garis Tegak Lurus .....             | 10   |
| 2.5.2 Metode Keseimbangan.....                   | 10   |
| 2.5.3 Himpunan Titik-Titik Tentu .....           | 11   |
| 2.5.4 Dekomposisi Geometri .....                 | 11   |
| 2.5.5 Rumus Integral.....                        | 12   |
| 2.5.6 <i>Centroid</i> Poligon.....               | 13   |
| 2.5.7 Metode Monte Carlo .....                   | 14   |



|  |  |           |
|--|--|-----------|
| 2.6  | Citra Digital.....                                     | 14        |
| 2.6.1  | Piksel dan Resolusi Citra .....                        | 15        |
| 2.6.2  | Citra <i>Bitmap</i> .....                              | 15        |
| 2.6.3  | Citra RGB .....  | 16        |
| 2.6.4  | Citra Grayscale.....                                   | 17        |
| 2.6.5  | Citra Biner.....                                       | 17        |
| 2.7  | Segmentasi Citra .....                                 | 18        |
| 2.7.1  | <i>Grabcut</i> .....                                   | 19        |
| 2.7.2  | Metode Otsu.....                                       | 21        |
| <b>BAB III METODE PENELITIAN .....</b>           |  | <b>22</b> |
| 3.1  | Studi Literatur .....                                  | 22        |
| 3.2  | Penentuan Spesifikasi Alat.....                        | 22        |
| 3.3  | Konfigurasi Sistem.....                                | 23        |
| 3.4  | Pengujian dan Analisis Sistem.....                     | 23        |
| 3.5  | Kesimpulan dan Saran .....                             | 24        |
| <b>BAB IV PERANCANGAN DAN IMPLEMENTASI .....</b> |  | <b>25</b> |
| 4.1  | Perancangan Secara Umum .....                          | 25        |
| 4.1.1  | Blok Diagram Sistem.....                               | 25        |
| 4.1.2  | Prinsip Kerja Aplikasi.....                            | 26        |
| 4.2  | Perancangan Perangkat Lunak .....                      | 27        |
| 4.2.1  | Subproses Segmentasi.....                              | 27        |
| 4.2.2  | Subproses <i>Grayscale</i> dan Pengembangan Otsu ..... | 28        |
| 4.2.3  | Subproses Pelukisan Garis <i>Rule Of Thirds</i> .....  | 29        |
| 4.2.4  | Subproses Pengecekan <i>Rule Of Thirds</i> .....       | 29        |
| 4.3  | Implementasi Sistem .....                              | 32        |
| 4.3.1  | Lingkungan Implementasi .....                          | 32        |
| 4.4  | Implementasi Antarmuka.....                            | 32        |

|   |           |
|---|-----------|
| 4.4.1 Implementasi Antarmuka Pemuatan Citra .....             | 33        |
| 4.4.2 Implementasi Antarmuka Pengolahan Citra .....           | 33        |
| <b>BAB V PENGUJIAN .....</b>                                  | <b>35</b> |
| 5.1 Pengujian Penilaian Foto Oleh Komputer .....              | 35        |
| 5.2 Pengujian Penilaian Foto Oleh Fotografer Profesional..... | 42        |
| 5.3 Persentase Keberhasilan Pengujian.....                    | 48        |
| 5.4 Analisis Faktor Kegagalan.....                            | 48        |
| <b>BAB VI PENUTUP .....</b>                                   | <b>50</b> |
| 6.1 Kesimpulan .....  | 50        |
| 6.2 Saran.....  | 50        |
| <b>DAFTAR PUSTAKA.....</b>                                    | <b>51</b> |
| <b>LAMPIRAN.....</b>  | <b>56</b> |





## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Komposisi yang tidak sesuai dengan <i>rule of thirds</i> dan komposisi yang sesuai.....                            | 6  |
| Gambar 2.2 Foto kayaker tanpa garis bantu <i>rule of thirds</i> dan foto kayaker dengan garis bantu.....                      | 7  |
| Gambar 2.3 Contoh komposisi <i>rule of thirds</i> .....   | 8  |
| Gambar 2.4 Penggunaan metode tegak lurus.....   | 10 |
| Gambar 2.5 Penggunaan metode keseimbangan. ....   | 10 |
| Gambar 2.6 Contoh penggunaan dekomposisi geometri .....   | 12 |
| Gambar 2.7 Contoh sebuah poligon sederhana. ....  | 13 |
| Gambar 2.8 Representasi Citra Digital. ....   | 14 |
| Gambar 2.9 Aturan koordinat representasi citra digital .....  | 15 |
| Gambar 2.10 Contoh citra bitmap.....  | 16 |
| Gambar 2.11 Diagram kanal RGB.....  | 16 |
| Gambar 2.12 Contoh citra <i>grayscale</i> . ....  | 17 |
| Gambar 2.13 Contoh Citra Biner. ....  | 18 |
| Gambar 2.14 Contoh segmentasi citra. ....   | 18 |
| Gambar 2.15 Ilustrasi proses <i>grabcut</i> yang diterapkan pada citra.....   | 20 |
| Gambar 2.16 Contoh hasil segmentasi <i>grabcut</i> .....  | 21 |
| Gambar 2.17 Contoh hasil metode Otsu.....   | 21 |
| Gambar 3.1 Konfigurasi Sistem.....  | 23 |
| Gambar 4.1 Blok Diagram Sistem.....   | 25 |
| Gambar 4.2 Diagram Alir Aplikasi.....   | 26 |
| Gambar 4.3 Diagram Alir Subproses Segmentasi. ....  | 28 |
| Gambar 4.4 Diagram Alir Subproses <i>Grayscale</i> dan Pengembangan Otsu .....  | 28 |
| Gambar 4.5 Diagram Alir Subproses Pelukisan Garis <i>Rule Of Thirds</i> .....   | 29 |
| Gambar 4.6 Diagram Alir Subproses Pengecekan <i>Rule Of Thirds</i> .....  | 32 |
| Gambar 4.7 Antarmuka pemuatan citra. ....   | 33 |
| Gambar 4.8 Antarmuka pengolahan citra sebelum dan sesudah segmentasi .....  | 34 |
| Gambar 4.9 Antarmuka pengolahan citra subproses pengembangan dan pelukisan garis <i>rule of thirds</i> .....                  | 34 |
| Gambar 4.10 Antarmuka pengolahan citra setelah selesai subproses pengecekan <i>rule of thirds</i> dan menampilkan nilai. .... | 34 |

## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 4.1 Toleransi dan skor yang didefinisikan untuk program. ....      | 31 |
| Tabel 5.1 Hasil pengujian 35 sampel foto dilakukan oleh komputer. ....   | 41 |
| Tabel 5.2 Hasil penilaian 35 sampel foto dilakukan oleh fotografer. .... | 48 |





## ABSTRAK

**TOMI PUTRO UTOMO**, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Agustus, 2014. *Penilaian Komposisi Rule Of Thirds Pada Fotografi Menggunakan Bantuan Pengolahan Citra Digital*, Dosen Pembimbing: Adharul Muttaqin, S.T., M.T. dan Ir. Muhammad Aswin, M.T.

*Rule of thirds* adalah salah satu panduan komposisi fotografi yang paling dikenal oleh fotografer untuk menciptakan foto yang berkualitas (Mai, Le, Niu, & Liu, 2011). Komposisi ini didapatkan dengan membagi bidang gambar dalam tiga bagian yang sama besar dan proporsional baik horizontal maupun vertikal. Maka terbentuklah garis-garis imajiner dan empat titik perpotongan garis imajiner tersebut. Menurut aturan ini, sebaiknya bagian foto yang paling menarik ditempatkan di salah satu titik tersebut (Tjin, 2011). Berdasarkan kajian psikologis, kemampuan manusia untuk menilai dan mengklasifikasi estetika lebih bersifat kualitatif dan subjektif dan banyak mengacu pada persepsi visual dan ketertarikan pada jenis fotografi yang berbeda-beda. (Mahlke, 2008)

Pada tugas akhir ini dibangun perangkat lunak untuk menilai komposisi *rule of thirds* pada sebuah foto. Foto dinilai dengan cara menghitung jarak antara *centroid* (pusat massa suatu bangun) subjek foto dengan salah satu titik perpotongan *rule of thirds* yang terdekat. *Centroid* ditentukan dengan menggunakan metode Monte Carlo, yaitu dengan cara menyebar sampel acak pada wilayah yang telah digunakan saat segmentasi subjek, kemudian sampel-sampel yang berada di area subjek akan dirata-rata untuk mendapatkan *centroid*-nya. Jarak dari *centroid* ke titik perpotongan terdekat dinilai dengan cara menghitung hipotenusanya (sisi miring segitiga), setelah itu dibandingkan dengan hipotenusa dari keseluruhan bidang foto yang telah dikalikan dengan toleransi dan rentang nilai yang telah ditentukan.

Hasil pengujian menunjukkan bahwa semakin dekat jarak *centroid* subjek dengan salah satu titik perpotongan maka nilainya akan semakin tinggi, begitu pula jika jarak *centroid* dengan salah satu titik perpotongan semakin jauh, maka nilainya akan semakin rendah. Namun, jika jarak antara *centroid* subjek dengan salah satu titik perpotongan terlalu jauh, maka foto dianggap tidak sesuai dengan *rule of thirds* dan nilainya 0 (nol). Hal ini terjadi karena posisi subjek berada di tengah-tengah bidang foto.

**Kata Kunci:** Aturan Sepertiga, *Centroid*, Citra Fotografi, Estetika, Fotografi, Komposisi Fotografi, Metode Monte Carlo, Otsu *Threshold*, *Rule of Thirds*.



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Rule of thirds adalah salah satu aturan komposisi yang paling dikenal oleh fotografer untuk menciptakan foto yang berkualitas (Mai, Le, Niu, & Liu, 2011). Komposisi ini didapatkan dengan membagi bidang gambar dalam tiga bagian yang sama besar dan proporsional baik horizontal maupun vertikal. Maka terbentuklah garis-garis imajiner dan empat titik perpotongan garis imajiner tersebut. Menurut aturan ini, sebaiknya bagian foto yang paling menarik ditempatkan di salah satu titik tersebut (Tjin, 2011). Sedangkan komposisi dalam fotografi bertujuan untuk menata elemen-elemen visual yang akan dimasukkan ke dalam foto agar menjadi lebih menarik (Tjin, 2011). Penerapan *rule of thirds* pada sebuah foto dapat membuat foto menjadi lebih baik secara kualitas visual. Fotografer amatir biasanya lebih sering menempatkan subjek foto pada tengah-tengah bidang foto, hal ini membuat foto tampak statis dan membosankan, sedangkan fotografer profesional menempatkan subjek foto pada sepertiga bagian bidang foto, baik di sepertiga kiri atas dan bawah maupun sepertiga kanan atas dan bawah bidang foto, penempatan seperti ini membuat foto tampak dinamis dan lebih indah dilihat (Waterhouse, 2011).

Berdasarkan kajian psikologis yang telah dilakukan oleh Mahlke (Mahlke, 2008), kemampuan yang dimiliki seorang manusia untuk menilai dan mengklasifikasi estetika bersifat kualitatif dan subjektif dan mengacu pada persepsi visual dan ketertarikan pada jenis fotografi yang berbeda. Sementara persepsi tersebut cenderung tidak pasti dan tidak presisi. Metode dalam menilai sesuatu dalam lingkungan yang tidak presisi dan tidak pasti dalam dunia komputer dilakukan dengan pendekatan *soft computing*, yaitu suatu metode yang berusaha menggabungkan berbagai macam pendekatan komputasi yang paralel dengan kemampuan otak manusia dalam memikirkan dan belajar dari lingkungan yang tidak pasti dan tidak presisi tersebut (Jang, Sun, & Mizutani, 1997).

Menurut Luo (Luo & Tang, 2008) estetika dalam fotografi berhubungan dengan empat poin yaitu, (1) komposisi, yaitu penataan elemen-elemen visual di dalam foto, (2) pencahayaan, agar dapat menghasilkan kesan tiga dimensi dan kontras yang baik pada foto, (3) pengendalian fokus, pemisahan subjek foto dengan latar belakangnya, (4) warna, aspek harmonisasi warna pada foto. Beberapa metode telah dikembangkan sebelumnya oleh para peneliti yang bertujuan untuk menilai dan mengklasifikasi sebuah foto (Tong,

Li, Zhang, He, & Zhang, 2004), (Datta, Joshi, Li, & Wang, 2006), (Ke, Tang, & Jing, 2006), (Luo & Tang, 2008), (Kurnianto, Sumpeno, & Hariadi, 2010). Namun penelitian-penelitian tersebut meneliti estetika pada citra fotografi dari berbagai parameter, dan tidak berfokus hanya kepada aturan komposisi *rule of thirds* saja. Selain algoritma perbaikan komposisi foto sebelumnya juga telah dikembangkan oleh Liu (Liu, Chen, Wolf, & Cohen-Or, 2010) dan perangkat lunak seperti *Adobe Photoshop CS6* dengan cara memotong foto agar sesuai dengan komposisi *rule of thirds*. Sedangkan mengacu pada penelitian yang telah dilakukan oleh Mai (Mai, Le, Niu, & Liu, 2011) dan Maleš (Maleš, Hedi, & Grgić, 2012), yang berfokus pada pendeteksian aturan komposisi *rule of thirds* pada citra fotografi, maka dapat disimpulkan bahwa aturan komposisi, terutama *rule of thirds*, adalah salah satu hal yang penting dalam menentukan kualitas estetika sebuah foto.

Dengan banyaknya penelitian mengenai perangkat lunak yang dapat menilai kualitas citra fotografi diharapkan dapat menjadi masukan bagi pengguna, baik awam maupun profesional dalam memilih mana foto yang terbaik dari sekian banyak foto berdasarkan penilaian komputer. Namun pemilihan terakhir tetap dilakukan oleh manusia karena komputer hanya digunakan untuk memberikan masukan nilai.

Berdasarkan uraian di atas, penelitian skripsi ini akan merancang sebuah perangkat lunak yang dapat menilai apakah komposisi komposisi dalam sebuah foto mengikuti aturan komposisi *rule of thirds* atau tidak.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka rumusan masalah ditekankan pada:

1. Bagaimana cara mengenali parameter yang akan dinilai sesuai dengan *rule of thirds*.
2. Bagaimana cara menilai parameter yang telah dikenali sesuai dengan *rule of thirds*.

## 1.3 Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam pembuatan program ini antara lain:

1. Citra yang dapat diproses adalah citra bitmap yang berformat \*.jpeg, \*.jpg, \*.bmp, dan \*.png;
2. Citra yang akan diproses adalah citra RGB;
3. Resolusi citra dibatasi pada resolusi VGA yaitu 640x480 piksel pada orientasi *landscape* atau 480x640 piksel pada orientasi *portrait*;



4. Bahasa pemrograman yang digunakan adalah C++ dengan pustaka OpenCV;
5. Aturan komposisi yang digunakan hanya *rule of thirds*;

#### 1.4 Tujuan

Tujuan penelitian dalam skripsi ini adalah:

Merancang dan membangun aplikasi yang dapat menilai kualitas foto berdasarkan aturan komposisi *rule of thirds*.

#### 1.5 Manfaat

Diharapkan manfaat yang dapat diperoleh melalui penelitian skripsi ini adalah:

##### a) Bagi Penyusun

1. Diharapkan dapat merancang dan membangun aplikasi penilai komposisi *rule of thirds* pada fotografi dengan baik.
2. Dapat mempelajari lebih dalam tentang fotografi, khususnya tentang komposisi *rule of thirds*.
3. Memahami bahwa tidak mudah bagi komputer untuk menilai sebuah karya seni seperti yang manusia lakukan.
4. Mengaplikasikan ilmu yang telah didapat dalam perkuliahan.

##### b) Bagi Pengguna

1. Diharapkan dapat memberikan pengetahuan bagi pengguna awam bahwa komposisi fotografi, terutama *rule of thirds* adalah salah aspek yang menentukan kualitas sebuah foto.

#### 1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini adalah sebagai berikut:

##### **BAB I           Pendahuluan**

Menjelaskan latar belakang, rumusan masalah, ruang lingkup, tujuan dan sistematika penulisan skripsi;

##### **BAB II           Tinjauan Pustaka**

Menjelaskan kajian pustaka dan dasar teori yang digunakan;

##### **BAB III          Metode Penelitian**

Menjelaskan metode yang digunakan dalam pengerjaan skripsi;

##### **BAB IV          Perancangan dan Implementasi**

Menjelaskan langkah langkah perancangan aplikasi penilai komposisi *rule of thirds* pada fotografi beserta hasil implementasinya;



**BAB V Pengujian**

Menjelaskan langkah-langkah pengujian dari sistem yang telah dibuat dan membahas hasil pengujiannya;

**BAB VI Penutup**

Berisi Kesimpulan dan Saran.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Fotografi

Fotografi pertama kali ditemukan oleh Nicéphore Niépce pada tahun 1825 (Batz & Willinghofer, 1997). Definisi fotografi menurut Spancer (Spancer, 1973) adalah seni, ilmu, dan praktik menciptakan gambar yang dapat bertahan lama dengan merekam cahaya atau dengan radiasi elektromagnetik lainnya, baik secara kimiawi dengan menggunakan material peka cahaya seperti film fotografi—biasa dikenal dengan sebutan rol film, atau secara elektronik menggunakan sebuah *image sensor*. Biasanya, sebuah lensa digunakan untuk memfokuskan cahaya yang dipantulkan atau dipancarkan dari benda-benda ke dalam gambar nyata pada permukaan peka cahaya di dalam kamera selama waktu yang telah ditentukan. Hasil yang didapatkan dalam sebuah *image sensor* adalah muatan listrik pada setiap diode peka cahaya, yang secara elektronik diproses dan disimpan dalam berkas citra digital untuk ditampilkan atau pengolahan selanjutnya.

Sedangkan secara etimologi, kata fotografi berasal dari bahasa Inggris: *photography*, yang berasal dari dua kata dalam bahasa Yunani yaitu "photos": cahaya dan "grafo": melukis/menulis (Collins, 2011), jadi jika kedua kata tersebut digabungkan, fotografi berarti melukis dengan cahaya.

#### 2.2 Komposisi Fotografi

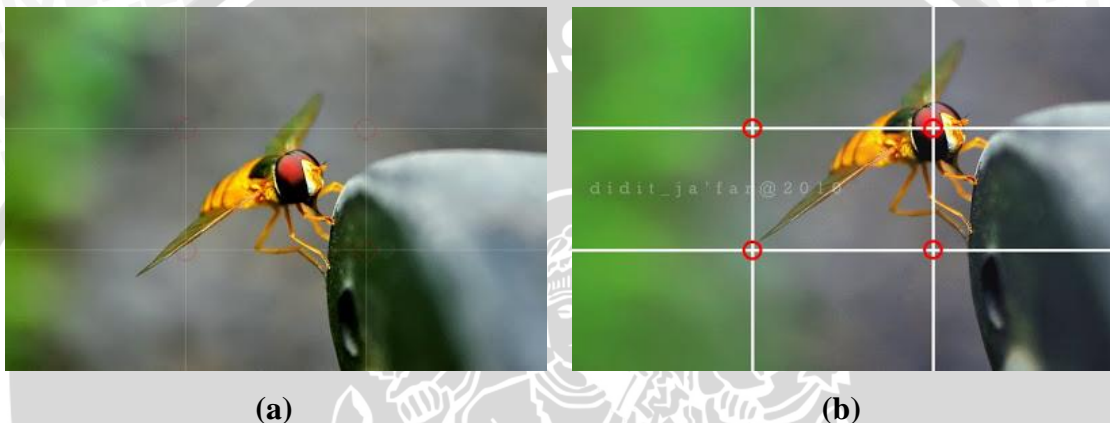
Komposisi fotografi adalah penataan elemen-elemen visual dengan sedemikian rupa sehingga foto tidak hanya menjadi lebih menarik tetapi juga mampu mengungkapkan maksud fotografer dengan jelas. Apa yang dikomunikasikan, tergantung dari apa yang dipilih untuk dimasukkan dalam foto dan bagaimana cara menyusunnya (Tjin, 2011).

Menurut Peterson (Peterson, 18 Composition Rules For Photos That Shine, 2013) terdapat 18 aturan komposisi yang digunakan di dunia fotografi, yaitu *rule of thirds*, *golden ratio*, *golden triangles*, *golden spirals*, *rule of odds*, *leaving space*, *fill the frame*, *simplification*, *balance*, *lines*, *patterns*, *color*, *texture*, *symmetry*, *viewpoint*, *background*, *depth*, dan *framing*. Dari ke-18 aturan komposisi tersebut, hanya aturan komposisi *rule of thirds* saja yang akan dibahas dalam skripsi ini.



### 2.3 Aturan Komposisi *Rule of Thirds*

*Rule of thirds*, atau aturan sepertiga merupakan rumus komposisi yang paling populer. Komposisi ini didapatkan dengan membagi bidang gambar dalam tiga bagian yang sama besar dan proporsional baik horizontal maupun vertikal. Dengan pembagian tersebut, terbentuklah garis-garis imajiner dan empat titik perpotongan garis imajiner tersebut. Menurut aturan ini, sebaiknya bagian foto yang paling menarik ditempatkan di salah satu titik tersebut. Titik yang sebelah mana tergantung dengan konteks, selera, dan apa yang ingin ditonjolkan. Aturan ini berlaku untuk sebagian besar jenis fotografi, dari pemandangan, *portrait*, *still life*, foto jurnalisisme, dan lain-lain (Tjin, 2011).



**Gambar 2.1** Komposisi yang tidak sesuai dengan *rule of thirds* (a) dan komposisi yang sesuai dengan *rule of thirds* (b).

Sumber: [adietdigitalizing.blogspot.com/2012/11/rule-of-thirds-komposisi-dasar-dalam.html](http://adietdigitalizing.blogspot.com/2012/11/rule-of-thirds-komposisi-dasar-dalam.html) (2012).

Foto pada Gambar 2.1 (a) peletakan subjek utama tidak sesuai dengan prinsip *rule of thirds* karena subjek utama yang menjadi pusat perhatian tepat berada di tengah-tengah foto dan tidak ada satu pun bagian objek yang berada di salah satu atau lebih titik perpotongan antara dua garis. Berbeda dengan posisi subjek pada Gambar 2.1 (b), di sini serangga diposisikan di salah satu titik perpotongan yang berada pada sebelah kanan atas gambar. Salah satu titik perpotongan berada tepat pada kepala serangga tersebut—dan ini berarti sesuai dengan prinsip *rule of thirds*.

Untuk makhluk hidup yang memiliki mata atau benda yang menyerupainya—contohnya patung binatang sebaiknya salah satu titik perpotongan berada pada bagian kepala seperti pada Gambar 2.1 (b), karena bagian kepala terutama mata adalah cerminan ekspresi subjek tersebut (Young, 2010). Tetapi itu adalah sebuah anjuran, bukan menjadi hal yang wajib dipatuhi jika ingin memotret makhluk hidup yang bermata. Contoh lain



terdapat pada Gambar 2.2 (b), di foto ini titik perpotongan sebelah kanan atas tidak berada pada bagian kepala, tetapi berada di bahu kiri si kayaker—dan kondisi ini tetap sesuai dengan *rule of thirds*. Karena *rule of thirds* pada prinsipnya hanya memperhitungkan apakah sebuah bagian dari subjek foto berada pada salah satu titik atau lebih perpotongan garis atau tidak.

Menurut Peterson (Peterson, *Why Does the Rule of Thirds Work?*, 2012) fotografer dan seniman menggunakan *rule of thirds* saat menciptakan karyanya karena suatu teori yang telah terbukti. Yaitu, jika subjek yang menjadi pusat perhatian berada di tengah bidang foto maka objek akan dianggap statis. Mata langsung diarahkan ke subjek dan tidak akan mengarah ke mana-mana lagi karena jarak subjek ke semua sisi foto sama. Untuk itu subjek ditempatkan mendekati salah satu tepi foto, hal ini akan memaksa mata mengikuti dan mencari objek foto. Dengan cara seperti ini foto menjadi lebih menarik karena terkesan interaktif. Seperti sebuah percakapan yang terjadi antara yang melihat dan foto tersebut.

Foto kayaker di Gambar 2.2 adalah salah satu contohnya, di foto itu subjek diposisikan di sepertiga bagian kanan gambar—mata manusia secara alamiah akan mengikuti condongnya subjek ke air, hal ini mungkin membuat pengamat merasa seperti sedang berada di dalam suasana tersebut. Jika subjek diposisikan di tengah-tengah gambar, mata tidak akan melirik ke mana-mana lagi. Pandangan akan “menabrak” tepi sebelah kiri gambar dan foto tersebut terasa aneh saat dilihat. Dengan posisi kayaker yang berada di kanan gambar, pengamat masih memiliki ruang untuk pergerakan mata.



(a)

(b)

**Gambar 2.2** Foto kayaker tanpa garis bantu *rule of thirds* (a) dan foto kayaker dengan garis bantu *rule of thirds* (b).

Sumber: [www.digital-photo-secrets.com/tip/2742/why-does-the-rule-of-thirds-work/](http://www.digital-photo-secrets.com/tip/2742/why-does-the-rule-of-thirds-work/) (2012).



**Gambar 2.3** Contoh komposisi *rule of thirds* (a) dan (b).

Sumber: [rumorkamera.com/catatan-kami/komposisi-fotografi-oleh-arbain-rambey/](http://rumorkamera.com/catatan-kami/komposisi-fotografi-oleh-arbain-rambey/) (2013).

## 2.4 Sampling

Sampling adalah cara pengumpulan data atau penelitian dengan hanya meneliti sebagian dari elemen populasi. Hasil dari proses sampling adalah data perkiraan. Berbeda dengan sensus yang mengumpulkan data dengan cara meneliti seluruh elemen populasi dan hasilnya merupakan data sebenarnya. Kekurangan dari metode sampling adalah adanya kesalahan sampling, yaitu suatu kesalahan yang terjadi karena tidak seluruh elemen diteliti. Makin kecil kesalahan sampling suatu perkiraan, makin teliti perkiraan tersebut, maka nilainya akan semakin mendekati nilai yang sebenarnya (Supranto, 2007).

### 2.4.1 Penarikan Sampel Acak Sederhana

Penarikan sampel acak sederhana menurut Cochran (Cochran, 2010) adalah sebuah metode untuk memilih  $n$  unit dari  $N$  sehingga setiap elemen dari  $N C_n$  sampel yang berbeda mempunyai kesempatan yang sama untuk dipilih. Dalam praktiknya, penarikan sampel acak sederhana dipilih unit per unit. Unit-unit dalam populasi diberi nomor dari 1 sampai  $N$  kemudian dipilih, dengan cara menggunakan sebuah tabel bilangan acak atau dengan program komputer yang menghasilkan tabel bilangan acak. Pada setiap penarikan, proses yang digunakan harus memberikan kesempatan terpilih yang sama untuk setiap bilangan dalam populasi. Unit-unit yang terpilih ini sebanyak  $n$  merupakan sampel. Untuk menentukan ukuran sampel dapat menggunakan rumus



$$n = \frac{\frac{t^2 PQ}{d^2}}{1 + \frac{1}{N} \left( \frac{t^2 PQ}{d^2} - 1 \right)} \quad (2-1)$$

dengan:

$n$  = jumlah sampel minimum

$t$  = tingkat kepercayaan

$P$  = proporsi dari karakteristik tertentu

$Q = 1 - P$

$d$  = taraf kekeliruan

$N$  = ukuran populasi

$I$  = bilangan konstanta

## 2.5 Centroid

*Centroid* berasal dua kata, yaitu dari bahasa Latin *centrum* yang berarti *center* dalam bahasa Inggris dan *oid* yang berarti *like* (Page, 2009). Suatu titik di mana seluruh berat atau massa pada sebuah bangun terkonsentrasi diketahui sebagai pusat gravitasi dari bangun tersebut. Pusat gravitasi secara umum dilambangkan dengan “G”.

*Centroid* adalah istilah lain yang berhubungan dengan pusat gravitasi. *Centroid* dari sebuah bangun geometri seperti persegi panjang, segi tiga, trapesium, lingkaran, dan lain-lain, kata *centroid* digunakan saat hanya ada bangun geometri daripada berat atau massa. Oleh karena itu, pusat gravitasi bangun geometri dapat diistilahkan sebagai *centroid* atau pusat dari area tersebut. (Afsar, 2013)

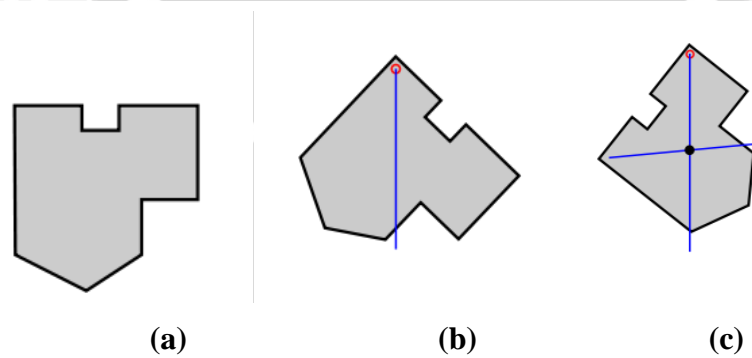
Beberapa metode yang dapat digunakan untuk menentukan *centroid* yaitu: metode garis tegak lurus, metode keseimbangan, himpunan dari titik-titik tentu, metode integral, metode dekomposisi geometris, dan metode poligon.

Namun pada penelitian ini akan digunakan metode Monte Carlo untuk menentukan *centroid* sebuah foto, metode ini dipilih karena subjek foto adalah objek yang bentuknya tidak selalu sama, dan diharapkan dapat menentukan *centroid* lebih cepat daripada menggunakan *vertex* poligon. Metode ini bekerja dengan cara menyebar sampel acak yang terdistribusi secara merata di bidang yang sudah disegmentasi, kemudian sampel-sampel tersebut dijumlahkan dan dirata-rata, lalu hasil rata-rata tersebut dipilih sebagai *centroid* subjek. Tetapi karena metode Monte Carlo menggunakan sampel acak, maka terdapat kemungkinan *centroid* bergeser dari lokasi sebelumnya, namun tidak akan terlalu jauh jaraknya.



### 2.5.1 Metode Garis Tegak Lurus

*Centroid* dari planar lamina seragam, seperti pada Gambar 2.4 (a), dapat ditentukan dengan menggunakan garis tegak lurus dan pin untuk menentukan *centroid* bangun tipis dengan kepadatan yang seragam. Keseluruhan bidang ditahan dengan pin yang diletakkan dekat pinggir, sehingga bangun tersebut dapat diputar dengan bebas. Garis tegak lurus dapat dilukis dari awal pin. Percobaan ini diulangi dengan meletakkan pin di titik yang lain sampai ditemukan perpotngan garis yang menandakan *centroid* seperti pada Gambar 2.4 (c). (Protter & Morrey, 1970)

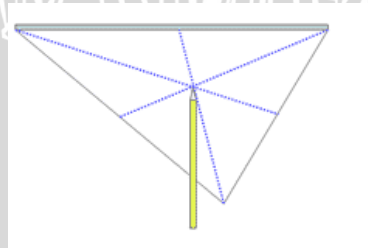


**Gambar 2.4** Penggunaan metode tegak lurus.

Sumber: <http://en.wikipedia.org/wiki/Centroid> (2014).

### 2.5.2 Metode Keseimbangan

Untuk bangun dengan lintasan tertutup dua dimensi, *centroid* dapat ditentukan dengan menyeimbangkan bangun pada bangun yang lebih kecil, seperti di ujung pensil yang runcing. *Centroid* berada di sekitar wilayah yang bersentuhan dengan bangun yang lebih kecil. (Protter & Morrey, 1970)



**Gambar 2.5** Penggunaan metode keseimbangan.

Sumber: <http://www.mathopenref.com/trianglecentroid.html> (2014).

### 2.5.3 Himpunan Titik-Titik Tentu

*Centroid* dari himpunan titik-titik yang telah ditentukan pada  $x_1, x_2, \dots, x_k$  dalam  $R^n$  adalah

$$C = \frac{x_1 + x_2 + \dots + x_k}{k} \quad (2-2)$$

dengan:

$C$  = *centroid*

$x_k$  = titik  $x$  ke- $k$

(Protter & Morrey, 1970)

### 2.5.4 Dekomposisi Geometri

*Centroid* pada bidang  $X$  dapat ditentukan dengan membaginya menjadi beberapa bidang yang lebih sederhana  $X_1, X_2, \dots, X_n$ , lalu menentukan *centroid*  $C_i$  dan area  $A_i$  dari setiap bagian, lalu hitung

$$C_x = \frac{\sum C_{ix} A_i}{\sum A_i} \quad (2-3)$$

$$C_y = \frac{\sum C_{iy} A_i}{\sum A_i} \quad (2-4)$$

dengan:

$C_x$  = *centroid* pada koordinat  $x$

$C_y$  = *centroid* pada koordinat  $y$

$\sum C_{ix}$  = Jumlah *centroid* ke- $i$  pada koordinat  $x$

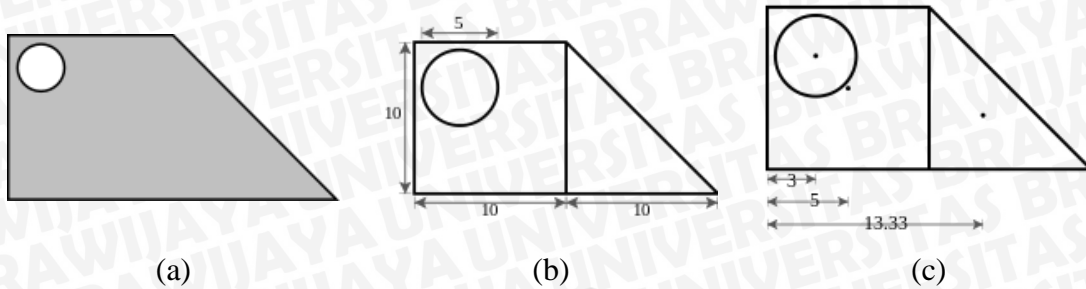
$\sum C_{iy}$  = Jumlah *centroid* ke- $i$  pada koordinat  $y$

$A_i$  = luas area bagian ke- $i$

$\sum A_i$  = Jumlah luas area bagian ke- $i$

Lubang pada bidang  $X$ , bertumpuk di antara bagian-bagian, atau bagian yang memanjang hingga luar bangun dapat dihitung dengan menggunakan area negatif  $A_i$ . Yaitu, mengukur  $A_i$  harus menggunakan tanda positif dan negatif sedemikian rupa sehingga jumlah dari tanda-tanda  $A_i$  untuk semua bagian yang menyertakan titik  $p$  tertentu adalah 1 jika  $p$  termasuk  $X$ , dan 0 untuk sebaliknya. (Protter & Morrey, 1970)

Misalnya, Gambar 2.6 (a) dapat dibagi menjadi persegi dan segitiga, keduanya dengan area positif; dan lubang lingkaran dengan luas negatif pada Gambar 2.6 (b).



**Gambar 2.6** Contoh penggunaan dekomposisi geometri

Sumber: <http://en.wikipedia.org/wiki/Centroid> (2014).

*Centroid* dari setiap bagian dapat ditemukan pada Gambar 2.6 (c). Kemudian *centroid* dari bidang dapat ditentukan dengan menghitung rata-rata dari ketiga *centroid* bangun sebelumnya. Posisi horizontal dari *centroid* dari sisi kiri bidang adalah

$$x = \frac{5 \times 10^2 + 13,33 \times \frac{1}{2} 10^2 - 3 \times \pi 2,5^2}{10^2 + \frac{1}{2} 10^2 - \pi 2,5^2} \approx 8,5 \text{ units.}$$

Posisi vertikal dari *centroid* dapat ditentukan dengan cara yang sama.

### 2.5.5 Rumus Integral

*Centroid* himpunan bagian  $X$  dari  $R^n$  juga dapat dihitung menggunakan integral

$$C = \frac{\int xg(x)dx}{\int g(x)dx} \tag{2-5}$$

dengan:

$g(x)$  = fungsi karakteristik dari himpunan bagian, yaitu 1 di dalam  $X$  dan 0 di luarnya

(Protter & Morrey, 1970). Rumus lain untuk menentukan *centroid* adalah

$$C_k = \frac{\int zS_k(z)dz}{\int S_k(z)dz} \tag{2-6}$$

dengan:

$C_k$  = koordinat ke- $k$  dari  $C$

$S_k(z)$  = ukuran dari perotongan  $X$  dengan *hyperplane* yang ditentukan oleh persamaan  $x_k = z$



### 2.5.6 Centroid Poligon

Poligon adalah sebuah bangun geometri yang dibentuk dari beberapa garis lurus yang saling terhubung dan akhirnya membentuk suatu lintasan tertutup. Garis lurus dari sebuah poligon disebut sisi atau tepi sedangkan titik pertemuan dari dua garis lurus yang saling terhubung disebut *vertex*. Dalam sebuah poligon sederhana tidak terdapat dua garis lurus yang saling berpotongan pada satu titik.

Luas dari sebuah poligon adalah daerah dua dimensi yang dibatasi oleh lintasan tertutup dari sebuah poligon. Untuk sebuah poligon sederhana dengan  $n$  *vertex*, luas dari poligon dan *centroid*-nya dapat dirumuskan sebagai berikut:

$$A = \frac{1}{2} \left| \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i) \right| \quad (2-7)$$

$$C_x = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (2-8)$$

$$C_y = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (2-9)$$

dengan:

$A$  = area poligon

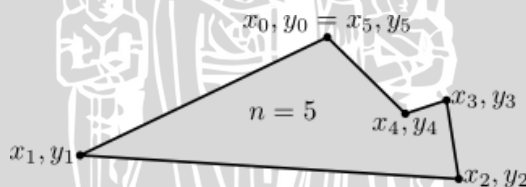
$C_x$  = *centroid* koordinat  $x$

$C_y$  = *centroid* koordinat  $y$

$n$  = jumlah *vertex*

$x_i$  = koordinat *vertex*  $x$  ke- $i$

$y_i$  = koordinat *vertex*  $y$  ke- $i$



**Gambar 2.7** Contoh sebuah poligon sederhana.

Sumber: [en.wikipedia.org/wiki/Polygon](http://en.wikipedia.org/wiki/Polygon) (2014).

Untuk membentuk sebuah lintasan tertutup pada poligon, maka *vertex* pertama dan *vertex* terakhir harus sama—atau dapat dikatakan sisi pertama dan sisi terakhir harus saling terhubung pada sebuah *vertex* yang sama, contohnya seperti  $x_0, y_0 = x_5, y_5$  pada Gambar 2.7. *Vertex* pada poligon harus diurutkan secara positif atau secara negatif (berlawanan arah putaran poros jarum jam atau searah putaran poros jarum jam). Jika *vertex* diurutkan secara negatif maka, hasil yang diberikan oleh rumus luas poligon akan

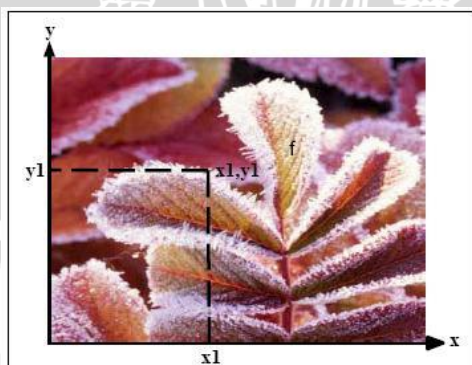
menjadi negatif, tetapi akan dikoreksi dengan nilai mutlak. Tetapi saat menghitung  $C_x$  dan  $C_y$ , hasil negatif yang didapatkan  $A$  harus tetap digunakan. (Bourke, 1988)

### 2.5.7 Metode Monte Carlo

Metode Monte Carlo adalah salah satu algoritma komputasi yang mengandalkan sampel acak yang berulang untuk mendapatkan hasil numerik; biasanya menggunakan sampel acak yang dibangkitkan dari pembangkit bilangan acak semu, sebuah simulasi dapat berjalan berkali-kali untuk mendapatkan distribusi entitas probabilistik yang tidak diketahui. Nama metode ini berasal dari kemiripan teknik yang digunakan untuk bermain dan mengingat hasil dalam perjudian kasino. Metode Monte Carlo sering digunakan dalam dunia fisika dan matematika, serta sangat berguna saat sulit atau tidak memungkinkan untuk mendapatkan ekspresi bentuk tertutup, atau tidak dapat menerapkan algoritma deterministik. Metode Monte Carlo terutama digunakan dalam tiga masalah yang berbeda yaitu: optimasi, integrasi numerik, dan pembangkitan seri dari distribusi probabilitas. Metode ini digunakan untuk mengevaluasi nilai integral tentu, biasanya integral multidimensi. (Nakano, 2013)

## 2.6 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel,  $f(x,y)$ , di mana  $x$  dan  $y$  adalah koordinat spasial dan nilai  $f(x,y)$  adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada Gambar 2.8. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue* – RGB). (Gonzalez & Woods, 2002)



**Gambar 2.8** Representasi Citra Digital

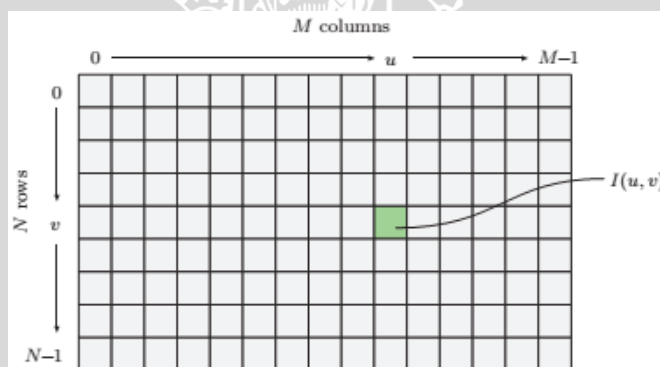
Sumber: <http://goo.gl/eiKSDP> (2008).



### 2.6.1 Piksel dan Resolusi Citra

Satuan atau bagian terkecil dari suatu citra disebut piksel (*picture element*) yang berarti elemen citra. Umumnya citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal antar piksel adalah sama pada seluruh bagian citra. Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (*integer*) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat (*integer*) untuk menunjukkan cahaya atau keadaan terang gelap piksel tersebut.

Untuk menunjukkan lokasi suatu piksel, koordinat  $(0,0)$  digunakan untuk posisi kanan atas dalam bidang citra, dan koordinat  $(m-1,n-1)$  digunakan untuk posisi kanan bawah dalam citra berukuran  $m \times n$  piksel. Untuk menunjukkan tingkat pencahayaan suatu piksel, sering kali digunakan bilangan bulat yang besarnya 8 bit, dengan lebar selang nilai 0-255, di mana 0 untuk warna hitam, 255 untuk warna putih dan tingkat abu-abu berada di antara nilai 0 dan 255. seperti yang diilustrasikan pada Gambar 2.9. (Burger & Burge, 2009)



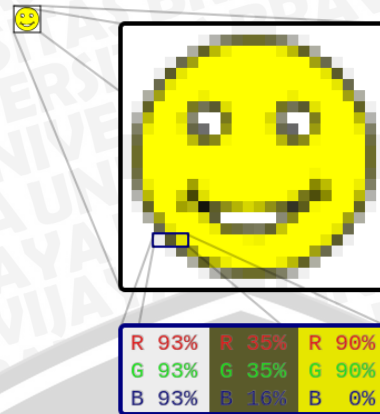
**Gambar 2.9** Aturan koordinat representasi citra digital.

Sumber: Burger & Burge (2009).

### 2.6.2 Citra Bitmap

Citra *bitmap* direpresentasikan dalam bentuk matriks atau dipetakan dengan menggunakan bilangan biner ataupun bilangan lain. Citra ini memiliki kelebihan untuk memanipulasi warna dari sebuah gambar. Citra bitmap atau disebut juga citra *raster* adalah citra yang menyimpan data kode citra secara digital dan lengkap (cara penyimpanan per-piksel). Citra bitmap biasanya diperoleh dari hasil pemindaian, kamera digital, dan *webcam*. (Sutoyo, Mulyanto, Suhartono, Nurhayati, & Wijanarto, 2009)





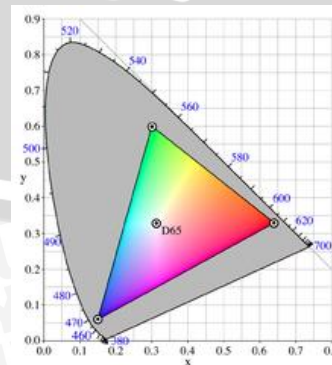
**Gambar 2.10** Contoh citra bitmap.

Sumber: [http://en.wikipedia.org/wiki/Raster\\_graphics](http://en.wikipedia.org/wiki/Raster_graphics) (2014).

### 2.6.3 Citra RGB

Skema warna RGB mengubah warna asli menjadi kombinasi tiga warna utama, yaitu *red* (R), *green* (G), dan *blue* (B). Skema ini digunakan secara luas pada pengiriman, representasi, dan penyimpanan citra pada perangkat analog seperti televisi analog atau perangkat digital seperti komputer, kamera digital, dan pemindai. Dengan alasan ini, banyak pengolahan citra dan program pengolah grafik menggunakan skema RGB sebagai representasi citra berwarna, dan banyak pustaka pengolahan citra, seperti Java Imaging API menggunakan RGB sebagai standar representasi citranya. (Burger & Burge, 2009)

Pada citra RGB, setiap komponen warna memiliki intensitas sendiri dengan nilai minimum 0 dan nilai maksimum 255 (8-bit). Hal ini menyebabkan setiap piksel pada citra RGB membutuhkan media penyimpanan 3 *byte*. Jumlah kemungkinan warna pada citra RGB adalah  $2^{24} = 16.777.216$  warna. (Gonzalez & Woods, 2002)



**Gambar 2.11** Diagram kanal RGB.

Sumber: [http://en.wikipedia.org/wiki/RGB\\_color\\_model](http://en.wikipedia.org/wiki/RGB_color_model), 2011.

#### 2.6.4 Citra Grayscale

Dalam fotografi dan komputasi, *grayscale* adalah gambar di mana nilai setiap piksel merupakan sampel tunggal, yaitu hanya membawa informasi intensitas. Gambar semacam ini, juga dikenal sebagai hitam dan putih, terdiri eksklusif dari warna abu-abu, bervariasi dari hitam pada intensitas terlemah putih di terkuat .

Gambar *grayscale* berbeda dari satu bit gambar hitam putih bi-tonal, yang dalam konteks pencitraan komputer adalah gambar yang terdiri dari dua warna, hitam dan putih (juga disebut bilevel atau gambar biner). Gambar *Grayscale* memiliki banyak nuansa abu-abu di antaranya. Gambar *grayscale* juga disebut monokromatik, yang menunjukkan kehadiran hanya satu (*mono*) warna (*chrome*). Gambar *grayscale* sering kali adalah hasil pengukuran intensitas cahaya pada setiap piksel dalam sebuah pita tunggal dari spektrum elektromagnetik (misalnya inframerah, cahaya tampak, ultraviolet dll), dan dalam kasus seperti ini, citra *grayscale* menjadi benar-benar monokromatik jika hanya frekuensi yang diberikan sajalah yang ditangkap. Tetapi citra *grayscale* juga dapat dikonversi dari gambar berwarna. (Burger & Burge, 2009)



**Gambar 2.12** Contoh citra *grayscale*.

Sumber: [en.wikipedia.org/wiki/Grayscale](http://en.wikipedia.org/wiki/Grayscale) (2013).

#### 2.6.5 Citra Biner

Citra biner atau *monokrom* adalah tipe gambar yang setiap pikselnya hanya berisi oleh dua kondisi nilai, yaitu hitam atau putih. Nilai ini biasanya di-*encode* dengan 1 bit (0 atau 1) per pikselnya. Citra biner biasanya digunakan untuk merepresentasikan garis gambar, pengkodean untuk transmisi fax serta *printer monokrom*. (Gonzalez & Woods, 2002)



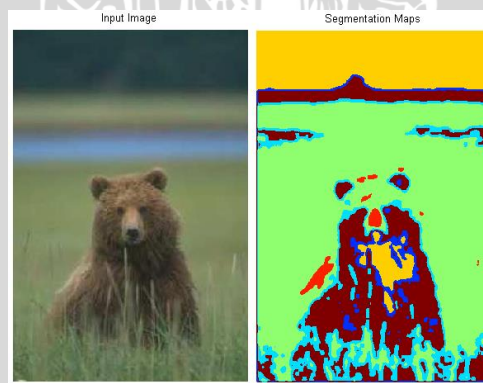


**Gambar 2.13** Contoh Citra Biner.

Sumber: [http://en.wikipedia.org/wiki/Binary\\_image](http://en.wikipedia.org/wiki/Binary_image) (2014).

## 2.7 Segmentasi Citra

Segmentasi citra bertujuan untuk membagi wilayah-wilayah yang homogen. Segmentasi adalah salah satu metode penting yang digunakan untuk mengubah citra input ke dalam citra *output* berdasarkan atribut yang diambil dari citra tersebut. Segmentasi membagi citra ke dalam daerah intensitasnya masing-masing sehingga bisa membedakan antara objek dan latar belakangnya. Pembagian ini tergantung pada masalah yang akan diselesaikan. Segmentasi harus dihentikan apabila masing-masing objek telah terisolasi atau terlihat dengan jelas. Tingkat keakurasian segmentasi bergantung pada tingkat keberhasilan prosedur analisis yang dilakukan. (Sutoyo, Mulyanto, Suhartono, Nurhayati, & Wijanarto, 2009)



**Gambar 2.14** Contoh segmentasi citra.

Sumber: <http://www.mathworks.com/matlabcentral/fileexchange/25257-color-image-segmentation> (2009).



### 2.7.1 Segmentasi *Grabcut*

*Grabcut* adalah salah satu metode segmentasi citra. Metode ini dapat mensegmentasi objek citra atau latar belakang secara interaktif. Biasanya, proses segmentasi citra menggunakan informasi tekstur yang berhubungan dengan warna seperti Magic Wand, atau informasi tepi citra yang berhubungan dengan kontras seperti Intelligent Scissors. Metode *grabcut* berdasar pada metode lainnya yaitu *graph-cut*. *Graph-cut* dapat mengombinasikan kedua informasi yang dapat digunakan untuk segmentasi citra yaitu tekstur dan dan tepi. Sedangkan *grabcut* sendiri adalah pengembangan dari *graph-cut*, tetapi dengan optimasi yang lebih baik, penggunaan algoritma yang teriterasi, dan dapat secara simultan memperkirakan hal lain di sekitar batas objek serta warna dari objek.

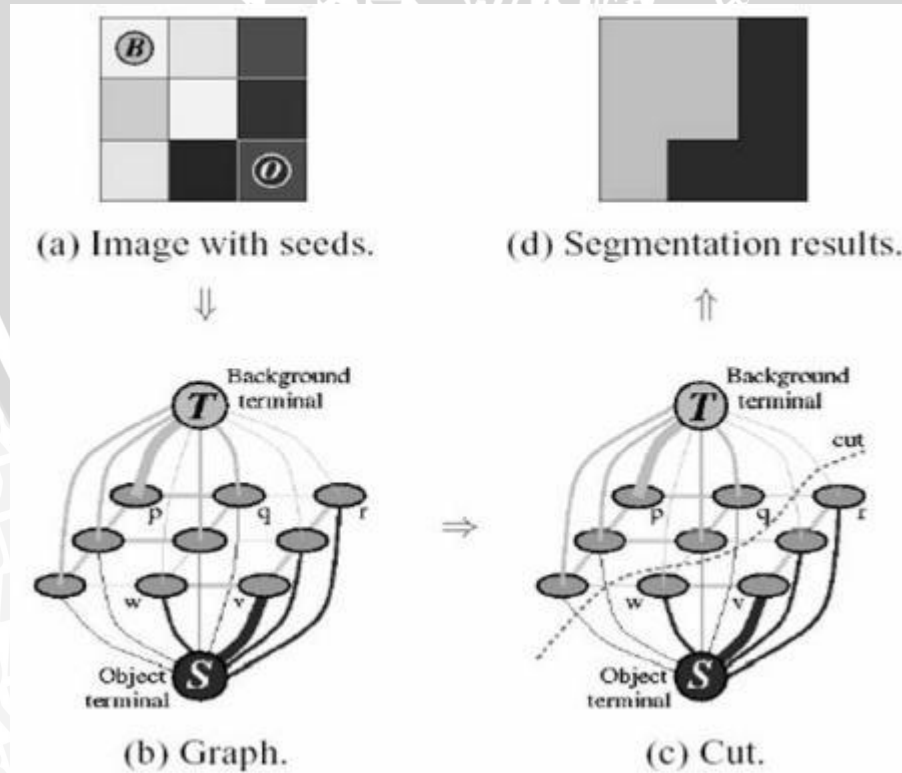
*Grabcut* digunakan untuk mengekstraksi subjek dari sebuah citra yang memiliki latar belakang yang kompleks di mana latar belakang tersebut tidak dapat dengan mudah dihilangkan. Cara kerja *grabcut* adalah dengan menganggap semua yang berada di luar subjek yang telah ditandai dengan sebuah bangun persegi panjang adalah latar belakang, kemudian subjek yang telah ditentukan dipisahkan dari latar belakang dengan cara memoloskan area di luar subjek yang telah ditentukan. (Rother, Kolmogorov, & Blake, 2004) *Grabcut* bekerja dengan cara pengguna melukis persegi panjang di sekitar daerah subjek (subjek harus benar-benar berada dalam area persegi panjang). Lalu *grabcut* mensegmentasi subjek tersebut secara berulang kali untuk mendapatkan hasil terbaik.

Berikut ini adalah penjelasan lebih lanjut tentang proses segmentasi *grabcut* yang dikutip dari website dokumentasi OpenCV di alamat [http://docs.opencv.org/trunk/doc/py\\_tutorials/py\\_imgproc/py\\_grabcut/py\\_grabcut.html](http://docs.opencv.org/trunk/doc/py_tutorials/py_imgproc/py_grabcut/py_grabcut.html):

1. Pengguna menentukan persegi panjang, semua yang berada di luar persegi panjang akan dianggap sebagai latar belakang. Area citra yang berada di dalam persegi panjang dianggap belum diketahui;
2. Komputer melakukan labeling awal mengacu pada citra di dalam persegi panjang yang telah diberikan. Komputer melabeli piksel di area subjek dan latar belakang (atau *hard label*-nya);
3. *Gaussian Mixture Model*(GMM) digunakan untuk memodelkan subjek dan latar belakang;
4. Mengacu pada citra yang telah diberikan, GMM mempelajari dan membuat distribusi piksel yang baru. Oleh karena itu, piksel-piksel yang tidak diketahui dilabeli sebagai kemungkinan subjek atau kemungkinan latar belakang

bergantung pada hubungannya dengan piksel *hard label* yang lain pada statistik warna seperti melakukan pengelompokan;

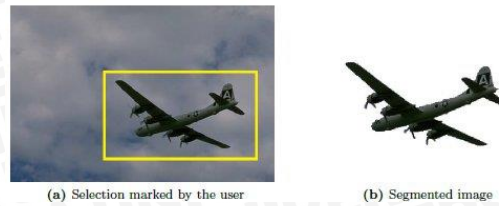
5. Graf dibuat berdasarkan distribusi pikselnya. Simpul pada graf adalah piksel. Dua simpul tambahan ditambahkan, *source node* dan *sink node*. Setiap piksel subjek terhubung ke *source node* dan setiap piksel latar belakang terhubung dengan *sink node*;
6. Berat dari piksel tepi yang terhubung dengan *source node* atau *end node* didefinisikan oleh probabilitas piksel dipilih sebagai subjek atau latar belakang. Berat di antara piksel didefinisikan oleh informasi tepi atau kesamaan piksel. Jika terdapat perbedaan yang besar dalam warna piksel, tepinya akan mendapatkan berat yang rendah;
7. Lalu algoritma *minicut* digunakan untuk mensegmentasi graf. Algoritma ini memotong Graf menjadi dua bagian terpisah *source node* and *sink node* dengan *cost function* yang minimum. *Cost function* adalah penjumlahan dari seluruh berat dari tepi yang dipotong. Setelah dipotong, semua piksel yang terhubung ke *source node* menjadi subjek dan yang terhubung ke *sink node* menjadi latar belakang;
8. Proses berlangsung terus sampai klasifikasi seluruh area yang dicakup selesai.



**Gambar 2.15** Ilustrasi proses *grabcut* yang diterapkan pada citra

Sumber: <http://www.cs.ru.ac.za/research/g02m1682/> (2014).



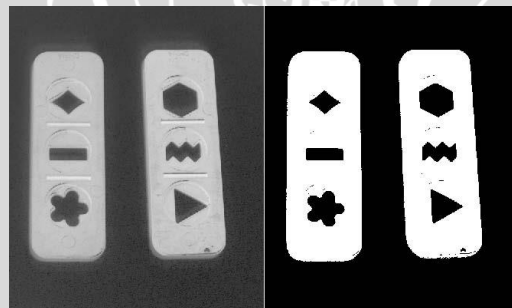


**Gambar 2.16** Contoh hasil segmentasi grabcut

Sumber: [http://www.scielo.edu.uy/scielo.php?pid=S0717-50002013000200004&script=sci\\_arttext](http://www.scielo.edu.uy/scielo.php?pid=S0717-50002013000200004&script=sci_arttext) (2014).

### 2.7.2 Metode Otsu

Metode Otsu merupakan metode yang digunakan untuk menghitung nilai ambang (T) secara otomatis berdasarkan citra masukan. Tujuan dari metode Otsu adalah membagi histogram citra gray level ke dalam dua daerah yang berbeda secara otomatis tanpa membutuhkan bantuan pengguna untuk memasukkan nilai ambang. Pendekatan yang dilakukan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimumkan variabel tersebut agar dapat membagi objek dengan latar belakang. Nilai ambang yang dicari pada suatu citra *grayscale* dilambangkan dengan  $k$ . Nilai  $k$  ini berkisar antara 1 sampai dengan  $L$  sedangkan  $L$  memiliki nilai maksimum 255. (Otsu, 1979)



**Gambar 2.17** Contoh hasil metode Otsu

Sumber: <http://www.walrusvision.com/wordpress/category/letters/> (2014).

## BAB III

### METODE PENELITIAN

Pada bab ini akan dijelaskan metodologi penelitian yang digunakan untuk mengimplementasikan perangkat lunak yang akan dibangun dalam penelitian skripsi perancangan dan pembuatan aplikasi penilai komposisi *rule of thirds* pada fotografi.

Langkah-langkah yang akan dikerjakan adalah sebagai berikut:

1. Studi Literatur;
2. Penentuan Spesifikasi Alat;
3. Konfigurasi Sistem;
4. Pengujian Dan Analisis Sistem;
5. Kesimpulan Dan Saran;

#### 3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

1. Mempelajari buku dan jurnal ilmiah yang berhubungan dengan pengolahan citra digital dan fotografi.
2. Mempelajari buku tentang pustaka OpenCV.

#### 3.2 Penentuan Spesifikasi Alat

Dalam hal ini menentukan perangkat keras dan perangkat lunak yang akan digunakan untuk menunjang proses perancangan dan pembangunan aplikasi. Oleh karena itu, di bawah ini akan dijelaskan spesifikasi alat yang digunakan pada skripsi ini.

- a. Perangkat keras:
  - 1 unit Notebook PC dengan spesifikasi Prosesor Intel Core i5 4200M 2,5 GHz, RAM 4 GB DDR3 SDRAM, VGA NVIDIA GEFORCE GT 740M 2 GB;
- b. Perangkat lunak:
  - Microsoft Windows Embedded 8.1 Industry Pro 64-bit;
  - Qt Creator 5.3.0;
  - Pustaka OpenCV 2.4.9;



### 3.3 Konfigurasi Sistem



Gambar 3.1 Konfigurasi Sistem.

Keterangan:

1. Citra Uji : Citra berskema warna RGB yang akan dijadikan citra uji pada sistem ini.
2. Komputer : Proses pengolahan citra dilakukan pada bagian ini. Seperti segmentasi, pengambangan, pelukisan garis-garis *rule of thirds*.
3. Citra Keluaran : Setelah citra diolah, *centroid* subjek foto ditentukan dengan menggunakan metode Monte Carlo, kemudian diukur jarak dari *centroid* ke salah satu titik perpotongan terdekat, lalu jarak tersebut dibandingkan toleransi dan rentang nilai yang telah ditentukan.
4. Hasil Proses : Nilai, status *rule of thirds*, posisi *centroid*, dan jarak dari titik perpotongan terdekat ke *centroid* serta informasi-informasi lainnya akan ditampilkan pada sebuah *messagebox*.

### 3.4 Pengujian dan Analisis Sistem

Pengujian dan analisis dilakukan untuk menguji kinerja sistem, dan agar dapat mengetahui dan memperbaiki jika terdapat cacat pada program. Pada penelitian skripsi ini dilakukan pengujian program. Pengujian aplikasi dilakukan terhadap beberapa jenis foto yang telah dipilih sebagai sampel dan juga dilakukan penilaian oleh fotografer profesional agar dapat membandingkan nilai yang diberikan oleh komputer dan seorang ahli. Foto yang dijadikan sampel uji adalah foto dokumentasi pribadi penulis dan foto dari beberapa situs komunitas fotografi. Tahapan rangkaian pengujian adalah sebagai berikut:

1. Pengujian penilaian foto oleh komputer;
2. Pengujian penilaian foto oleh fotografer profesional;

3. Analisis persentase keberhasilan dapat dihitung dengan persamaan

$$\text{persentase} = \frac{\text{total nilai}}{\text{jumlah sampel}} \times 100\% \quad (3-1)$$

dengan:

- Persentase = nilai yang menunjukkan akurasi sistem
- Total nilai = jumlah nilai dari masing-masing sampel foto
- Jumlah sampel = seluruh jumlah foto yang digunakan sebagai sampel

4. Analisis faktor kegagalan;

### 3.5 Kesimpulan dan Saran

Pada bagian ini akan dilakukan penarikan kesimpulan yang dihasilkan dari hasil pengujian dan analisis terhadap aplikasi penilai komposisi *rule of thirds* pada fotografi. Tahapan selanjutnya adalah memberikan saran untuk penyempurnaan sistem ke depannya.





## BAB IV

### PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan tentang perancangan dan implementasi aplikasi penilai komposisi *rule of thirds* pada fotografi dalam beberapa tahap pengerjaan. Tahapan-tahapan tersebut yaitu: inisialisasi citra, penentuan subjek dengan menggunakan segmentasi *grabcut*, pengambangan citra dengan menggunakan metode Otsu, melukis garis-garis *rule of thirds*, menentukan *centroid* dengan menggunakan metode Monte Carlo dari subjek yang telah diambangkan, mengukur jarak dari *centroid* ke salah satu titik perpotongan terdekat, dan yang terakhir menilai jarak sesuai dengan toleransi yang telah ditentukan. Pembuatan secara bertahap tersebut agar memudahkan penganalisaan aplikasi pada setiap bagian maupun secara keseluruhan.

#### 4.1 Perancangan Secara Umum

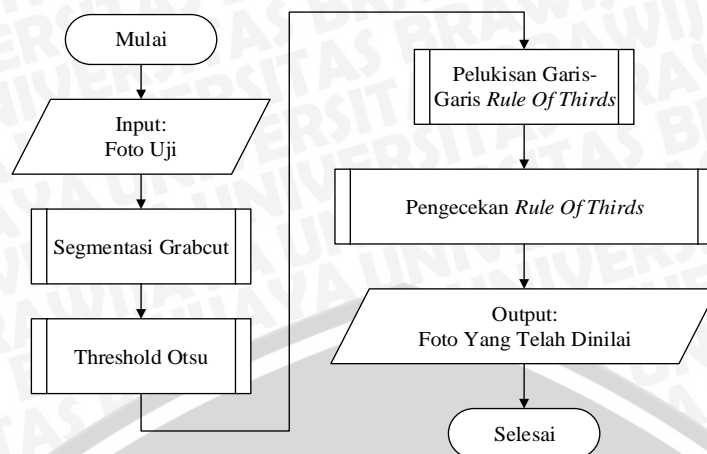
Tahap awal yang dilaksanakan adalah perancangan aplikasi secara global, tahap ini berfungsi sebagai acuan dalam proses pembangunan aplikasi. Tahap ini diawali dengan mendefinisikan perilaku penggunaan aplikasi penilai komposisi *rule of thirds* pada fotografi, dan menjelaskan tentang blok diagram sistem serta prinsip kerja aplikasi.

##### 4.1.1 Blok Diagram Sistem

Garis besar desain aplikasi penilai komposisi *rule of thirds* pada fotografi terdiri dari beberapa langkah. Blok diagram sistem ditunjukkan pada Gambar 4.1. Sedangkan untuk diagram alir aplikasi secara umum ditunjukkan pada Gambar 4.2.



Gambar 4.1 Blok Diagram Sistem



**Gambar 4.2** Diagram Alir Aplikasi.

Fungsi masing-masing bagian dari blok diagram sistem pada Gambar 4.2 dijelaskan sebagai berikut:

1. Citra RGB sebagai input sistem dengan resolusi maksimal sesuai dengan ketentuan pada batasan masalah;
2. Subjek pada foto disegmentasi dengan metode *grabcut* sehingga menghasilkan foto yang subjeknya terpisah dari latar belakangnya;
3. Dilakukan operasi pengambangan pada foto yang sudah terpisah subjeknya dengan metode Otsu;
4. Melukis garis-garis *rule of thirds* di atas bidang foto. Kemudian menentukan *centroid* subjek foto yang telah diambangkan dengan menggunakan metode Monte Carlo;
5. Mengukur jarak dari *centroid* ke salah satu titik perpotongan yang terdekat lalu membandingkannya dengan toleransi dan rentang nilai yang telah ditentukan;

#### 4.1.2 Prinsip Kerja Aplikasi

Pembangunan aplikasi ini menggunakan bahasa pemrograman pustaka *Open Source Computer Vision Library* (OpenCV). Prinsip kerja aplikasi penilai komposisi *rule of thirds* pada fotografi dimulai dengan memuat citra uji yang berskema warna RGB, kemudian citra disegmentasi dengan metode *grabcut* yang bertujuan untuk memisahkan subjek foto dengan latar belakangnya. Setelah subjek dipisahkan, maka dapat dilakukan operasi pengambangan dengan metode Otsu agar memudahkan penentuan *centroid* dengan metode Monte Carlo. Selanjutnya garis-garis *rule of thirds* dilukis di atas bidang foto agar memudahkan pengguna mengetahuinya. Kemudian sampel acak yang terdistribusi secara seragam disebar di wilayah segmentasi *grabcut* sebelumnya, hal ini bertujuan untuk menentukan titik-titik yang akan dirata-rata oleh metode Monte Carlo,



titik hasil rata-rata inilah yang ditentukan sebagai *centroid* subjek foto. Kemudian jarak dari *centroid* subjek ke salah satu titik perpotongan ditentukan dengan menghitung hipotenusa (sisi miring segitiga) dari jarak tersebut. Setelah jarak didapatkan, akan dibandingkan dengan toleransi jarak dan rentang nilai yang telah ditentukan. Terakhir, nilai, lokasi *centroid*, jarak dari *centroid* ke salah satu titik perpotongan, dan informasi lainnya akan ditampilkan pada kotak pesan di layar.

## 4.2 Perancangan Perangkat Lunak

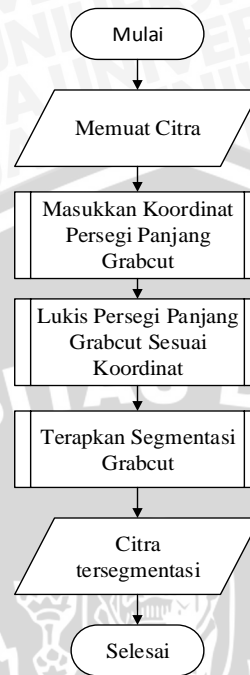
Pada bagian ini perangkat lunak akan dibangun menggunakan bahasa pemrograman C++ dengan menggunakan IDE (*Integrated Development Environment*) Qt Creator dan pustaka OpenCV. Sistem ini dirancang mampu melakukan hal-hal sebagai berikut:

1. Mengakses citra yang tersimpan di dalam media penyimpanan yang terpasang di komputer;
2. Melukis bidang persegi panjang sesuai dengan ukuran yang ditentukan oleh pengguna;
3. Melakukan segmentasi *grabcut* pada bidang persegi panjang yang telah ditentukan;
4. Melakukan proses konversi dari citra RGB ke *grayscale*;
5. Melakukan operasi pengambangan dengan metode Otsu;
6. Melukis garis-garis *rule of thirds* pada bidang foto;
7. Menentukan *centroid* dari subjek foto dengan menggunakan metode Monte Carlo;
8. Mengukur jarak dari *centroid* ke salah satu titik perpotongan terdekat dengan menghitung hipotenusa (sisi miring segitiga) dari jarak tersebut;
9. Membandingkan jarak yang telah didapat dengan toleransi jarak dan rentang nilai yang telah ditentukan;
10. Menampilkan nilai, posisi *centroid*, jarak *centroid* ke arah titik perpotongan terdekat, dan beberapa informasi lainnya;

### 4.2.1 Subproses Segmentasi

Pada subproses segmentasi, foto yang dimuat akan dipisahkan subjeknya. Caranya seperti yang terlihat pada Gambar 4.3 adalah dengan melukis bidang persegi panjang di sekitar subjek yang ingin dipisahkan sesuai dengan koordinat yang telah ditentukan oleh pengguna. Setelah ditentukan daerah yang akan disegmentasi, barulah segmentasi

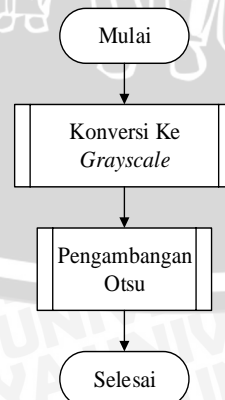
*grabcut* diterapkan sehingga menghasilkan subjek foto yang terpisah dari latar belakangnya—latar belakang diganti dengan warna hitam agar memudahkan saat proses pengambangan.



**Gambar 4.3** Diagram Alir Subproses Segmentasi.

#### 4.2.2 Subproses *Grayscale* dan Pengambangan Otsu

Pada subproses konversi ke *grayscale* dan pengambangan Otsu seperti yang terlihat pada Gambar 4.4, foto yang telah dipisahkan subjeknya akan diubah menjadi citra *grayscale* untuk menjadi masukan saat proses pengambangan Otsu. Proses ini bertujuan untuk mengubah citra RGB menjadi citra biner yang nantinya akan digunakan saat proses pengecekan *rule of thirds*.

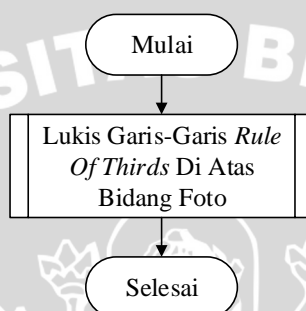


**Gambar 4.4** Diagram Alir Subproses Grayscale dan Pengambangan Otsu



#### 4.2.3 Subproses Pelukisan Garis *Rule Of Thirds*

Pada subproses ini garis-garis *rule of thirds* dilukiskan di atas bidang foto dengan tujuan memudahkan pengguna mengetahui bagian mana saja yang berpotongan dengan garis-garis *rule of thirds* seperti terlihat pada Gambar 4.5. Garis pertama dilukis dengan cara membagi bidang foto menjadi sepertiga bagian horizontal, kemudian garis kedua dilukis dengan cara membagi bidang foto menjadi dua kali sepertiga bagian horizontal. Garis ketiga dilukis dengan cara membagi bidang foto menjadi sepertiga bagian vertikal, kemudian garis keempat dilukis dengan cara membagi bidang foto menjadi dua kali sepertiga bagian vertikal.



**Gambar 4.5** Diagram Alir Subproses Pelukisan Garis *Rule Of Thirds*

#### 4.2.4 Subproses Pengecekan *Rule Of Thirds*

Pada subproses ini seperti pada Gambar 4.6 dilakukan beberapa langkah pengerjaan yaitu: 1. metode Monte Carlo; 2. mengukur jarak dari *centroid* ke titik perpotongan terdekat; 3. membandingkan jarak dengan toleransi dan rentang nilai yang telah ditentukan; 4. setelah itu akan ditampilkan nilainya di layar. Penjelasan lebih detail adalah:

1. Pada tahap metode Monte Carlo, *centroid* subjek akan ditentukan dengan cara menyebar sampel secara acak di seluruh bidang yang digunakan saat segmentasi *grabcut*. Kemudian dicari sampel-sampel mana saja yang memiliki intensitas warna 255 atau warna putih. Lalu sampel-sampel berwarna putih tersebut akan dijumlahkan kemudian dirata-rata. Hasil rata-rata tersebut kemudian ditetapkan sebagai *centroid* subjek. Posisi *centroid* yang ditentukan kemungkinan dapat berpindah dari lokasi sebelumnya karena sampel yang digunakan bersifat acak;
2. Pada tahap ini dilakukan pengukuran jarak dengan menghitung hipotenusa (sisi miring segitiga) dari *centroid* ke empat titik perpotongan, kemudian dicari yang terdekat;

3. Pada tahap ini akan dibandingkan jarak yang didapatkan pada tahap 2 dengan toleransi dikalikan dengan hipotenusa dari keseluruhan bidang foto. Toleransi dan rentang nilai telah didefinisikan sebelumnya seperti terlihat pada **Tabel 4.1**. Jika jarak dari *centroid* ke titik perpotongan lebih kecil daripada hipotenusa foto dikalikan dengan toleransi, maka skor yang telah terdefinisi dan status *rule of thirds* akan ditampilkan di layar. Namun jika jarak *centroid* ke titik perpotongan lebih besar dari jarak toleransi dikalikan dengan hipotenusa foto maka dianggap tidak terdeteksi *rule of thirds* dan skornya nol. Formula untuk menentukan centroid adalah

$$x_t = G_{xb} + \text{random mod} (G_{xe} - G_{xb}) \quad (4-1)$$

$$y_t = G_{yb} + \text{random mod} (G_{ye} - G_{yb}) \quad (4-2)$$

$$C_x = \frac{x_{acc}}{\text{sample}} \quad (4-3)$$

$$C_y = \frac{y_{acc}}{\text{sample}} \quad (4-4)$$

dengan:

$x_t$  = variabel untuk menyimpan nilai  $x$  sementara

$x_t$  = variabel untuk menyimpan nilai  $x$  sementara

$G_{xb}$  = koordinat awal *grabcut* pada koordinat  $x$

$G_{yb}$  = koordinat awal *grabcut* pada koordinat  $y$

$G_{xe}$  = koordinat akhir *grabcut* pada koordinat  $x$

$G_{ye}$  = koordinat akhir *grabcut* pada koordinat  $y$

$C_x$  = posisi *centroid* pada koordinat  $x$

$C_y$  = posisi *centroid* pada koordinat  $y$

$x_{acc}$  = nilai akumulatif  $x$  ( $x_{acc}$ ) +  $x_t$

$y_{acc}$  = nilai akumulatif  $y$  ( $y_{acc}$ ) +  $y_t$

Formula untuk mengukur jarak centroid ke perpotongan adalah

$$\text{dist} = \sqrt{(C_x - I_{xi})^2 + (C_y - I_{yi})^2} \quad (4-5)$$

dengan:

$\text{dist}$  = jarak dari *centroid* ke titik perpotongan

$C_x$  = posisi *centroid* pada koordinat  $x$

$C_y$  = posisi *centroid* pada koordinat  $y$

$I_{xi}$  = perpotongan garis ke- $i$  pada koordinat  $x$



$I_{yi}$  = perpotongan garis ke- $i$  pada koordinat  $y$

$i = 1,2,3,4$

Sedangkan formula untuk mengukur hipotenusa keseluruhan bidang foto adalah

$$hyp = \sqrt{(I_w)^2 + (I_h)^2} \quad (4-6)$$

dengan:

$hyp$  = hipotenusa keseluruhan bidang foto

$I_w$  = lebar bidang foto

$I_h$  = tinggi bidang foto.

Untuk pengecekan *rule of thirds* dan penilaian digunakan formula

$$if(dist < tolerance * hyp) \quad (4-7)$$

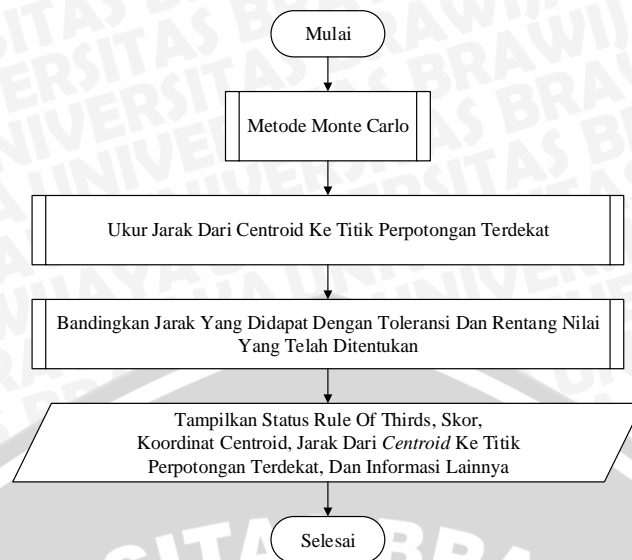
dengan:

$tolerance$  = rentang toleransi pada Tabel 4.1;

| Toleransi | Skor |
|-----------|------|
| 0,001     | 100  |
| 0,008     | 95   |
| 0,020     | 90   |
| 0,030     | 85   |
| 0,045     | 80   |
| 0,060     | 75   |
| 0,090     | 60   |
| 0,100     | 50   |
| > 0,100   | 0    |

**Tabel 4.1** Toleransi dan skor yang didefinisikan untuk program.

- Menampilkan status *rule of thirds*, skor, jarak dari *centroid* ke titik perpotongan terdekat, koordinat *centroid*, dan beberapa informasi lainnya;



**Gambar 4.6** Diagram Alir Subproses Pengecekan Rule Of Thirds

### 4.3 Implementasi Sistem

Setelah tahap perancangan selesai, tahapan selanjutnya adalah implementasi. Tahap ini merupakan proses penerapan dari proses sebelumnya, hasil perancangan diterapkan pada program sesuai dengan bahasa pemrograman yang digunakan dalam penelitian ini yaitu bahasa pemrograman C++.

#### 4.3.1 Lingkungan Implementasi

Aplikasi dibangun menggunakan bahasa pemrograman C++ pada IDE Qt Creator serta menggunakan pustaka OpenCV. Sistem diimplementasikan pada perangkat keras dengan spesifikasi sebagai berikut:

Notebook PC

Spesifikasi:

Prosesor : Intel Core i5 4200M 2,5GHz;

Memori : 4GB DDR3-SDRAM;

OS : Microsoft Windows Embedded 8.1 Industry Pro 64-bit;

VGA : NVIDIA GeForce GT740M 2GB;

### 4.4 Implementasi Antarmuka

Program penilai komposisi *rule of thirds* pada fotografi ini mempunyai 3 tampilan utama berikut ini tampilan:

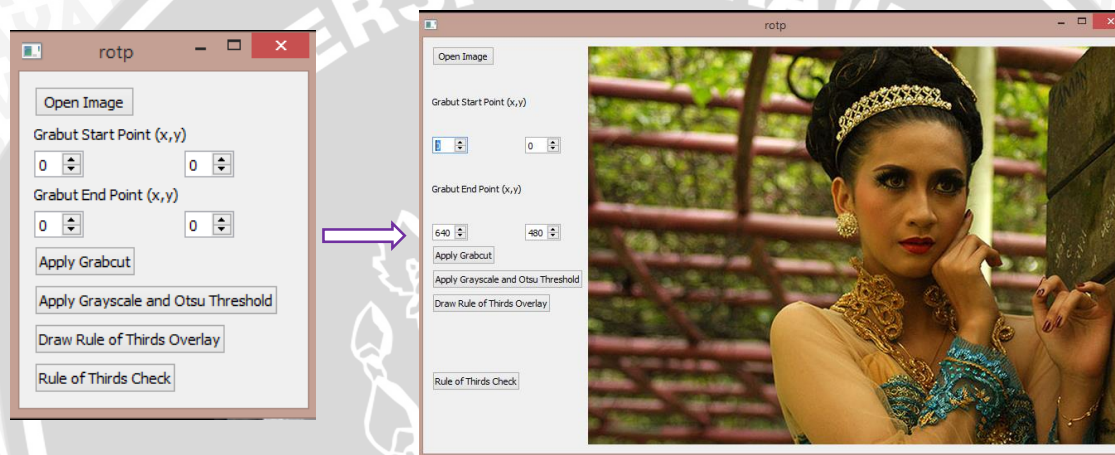
1. Pemuatan Citra;
2. Proses Pengolahan Citra;



Bentuk antarmuka program ini dibuat dengan desain sesederhana mungkin dan sesuai dengan langkah-langkah yang berurutan, hal ini bertujuan untuk memudahkan pemakai dalam menggunakan program ini.

#### 4.4.1 Implementasi Antarmuka Pemuatan Citra

Pada tampilan antarmuka pemuatan citra yang terlihat pada Gambar 4.7 terdapat beberapa tombol yang berfungsi untuk mengambil citra dari *harddisk*, mengaplikasikan segmentasi *grabcut*, mengaplikasikan pengambangan Otsu, melukis garis-garis *rule of thirds*, dan mengecek apakah citra yang dimuat sesuai dengan *rule of thirds*. Serta terdapat empat spinbox untuk mengatur ukuran bidang persegi panjang yang akan digunakan oleh segmentasi *grabcut*.

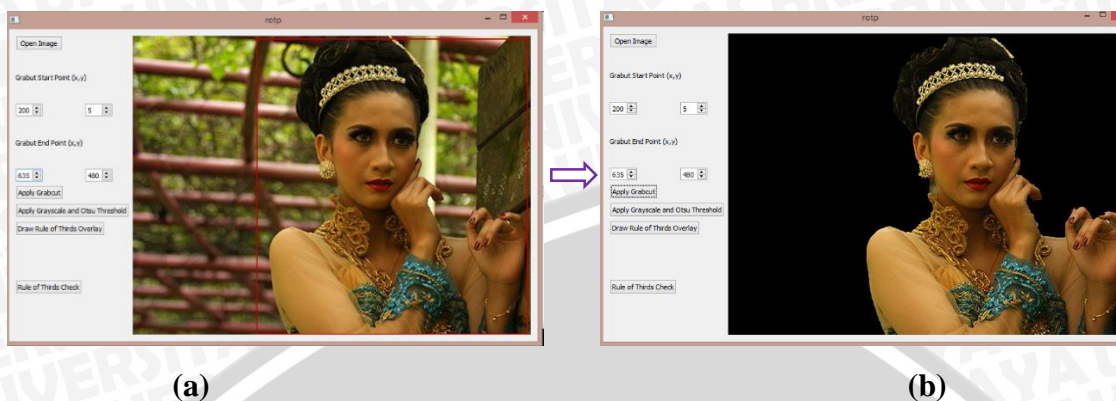


Gambar 4.7 Antarmuka pemuatan citra.

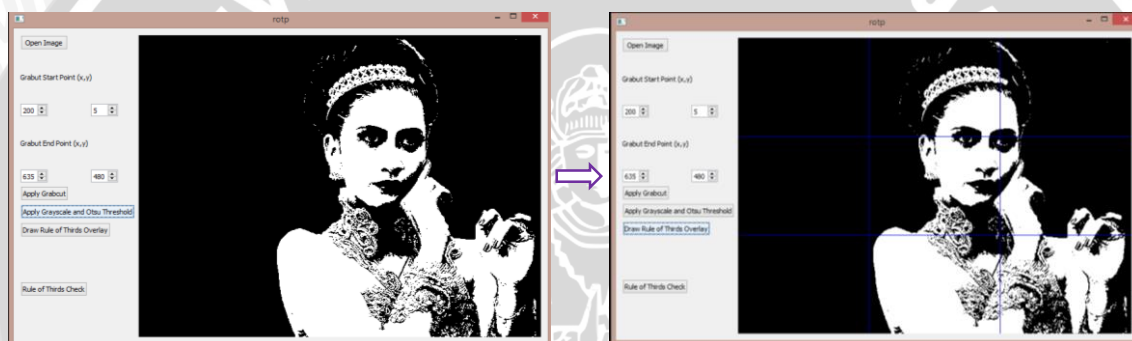
#### 4.4.2 Implementasi Antarmuka Pengolahan Citra

Pada antarmuka proses pengolahan citra terdapat empat langkah. Langkah pertama adalah segmentasi dengan metode *grabcut*, segmentasi ini menghasilkan citra yang subjeknya terpisah dari latar belakangnya—dan latar belakangnya diganti dengan warna hitam. Seperti yang terlihat pada Gambar 4.8 (a) dan Gambar 4.8 (b). Pada langkah kedua dilakukan proses perubahan ke dalam citra *grayscale* dan proses pengambangan citra seperti pada Gambar 4.9 (a). Pada langkah ketiga garis-garis *rule of thirds* akan dilukis di atas bidang citra seperti pada Gambar 4.9. (b) Pada langkah keempat akan dilakukan penentuan *centroid* dengan menggunakan metode Monte Carlo. Setelah *centroid* ditentukan, diukur jaraknya dari titik perpotongan terdekat dengan cara menghitung hipotenusa dari jarak tersebut. Setelah jarak didapatkan, maka jarak tersebut akan dibandingkan dengan toleransi dan rentang nilai yang telah ditentukan. Lalu nilai,

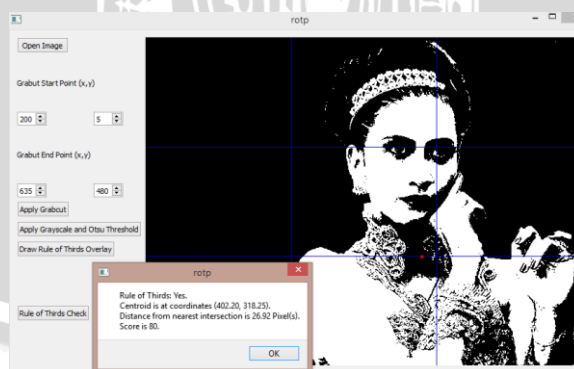
status *rule of thirds*, dan informasi lainnya akan ditampilkan pada *messagebox* seperti pada Gambar 4.10.



**Gambar 4.8** (a) Antarmuka pengolahan citra sebelum subproses segmentasi dan (b) sesudah subproses segmentasi.



**Gambar 4.9** (a) Antarmuka pengolahan citra saat subproses pengambangan dan (b) pelukisan garis *rule of thirds*.



**Gambar 4.10** Antarmuka pengolahan citra setelah selesai subproses pengecekan *rule of thirds* dan menampilkan nilai.




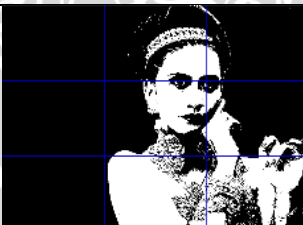


## BAB V PENGUJIAN

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan, maka diperlukan pengujian, yaitu menilai beberapa citra uji sesuai dengan aturan *rule of thirds*. Pengujian ini dilakukan pada 35 sampel foto. Pengujian yang dilakukan dalam bab ini adalah:


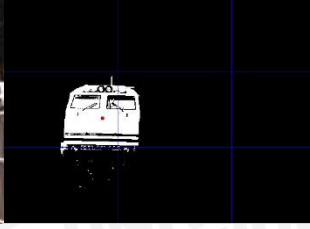

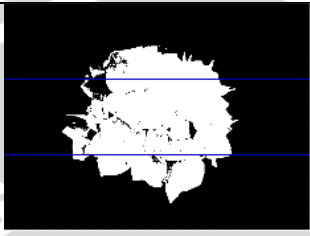

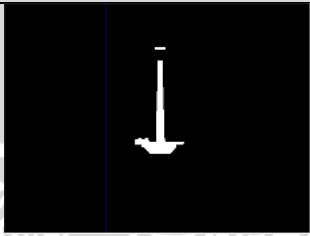

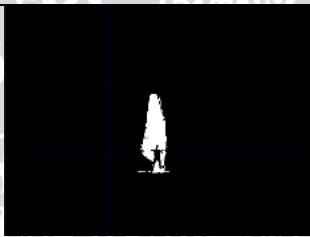

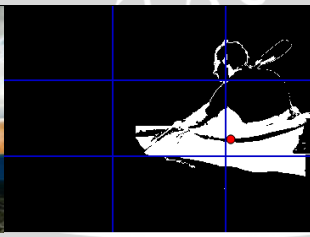
1. Pengujian penilaian foto oleh komputer;
2. Pengujian penilaian foto oleh fotografer profesional;
3. Analisis faktor kegagalan;

### 5.1 Pengujian Penilaian Foto Oleh Komputer


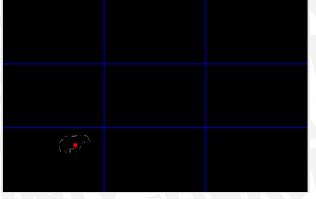

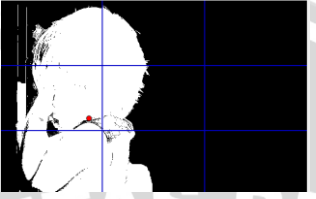

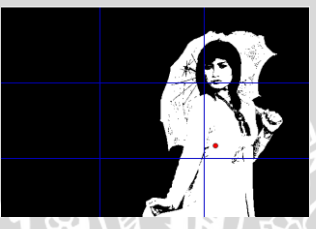

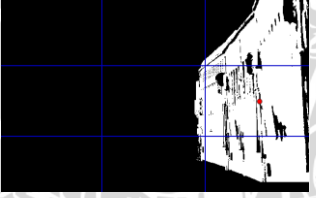

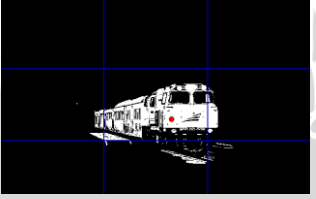

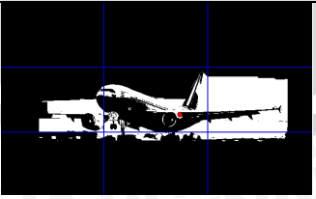
Berikut adalah hasil pengujian terhadap 35 sampel foto yang dilakukan oleh komputer.

| No. | Citra Masukan  | Citra Keluaran  | Status Rule Of Thirds | Skor | Jarak Centroid Ke Titik Perpotongan Terdekat (piksel) |
|-----|--|---|-----------------------|------|---|
| 1   | <br>(Utomo, Kebaya Menatap, 2013) |  | Ya                    | 85   | 20,56   |
| 2   | <br>(Utomo, Kunci Ganda, 2012)    |  | Ya                    | 60   | 51,70   |




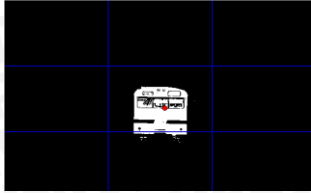

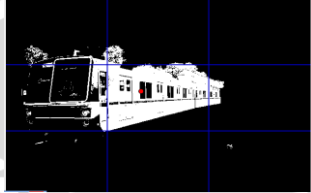

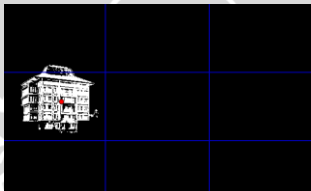

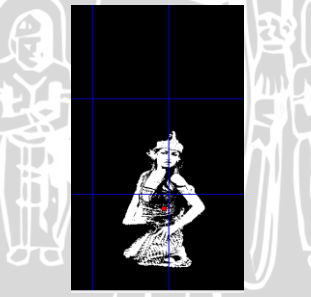

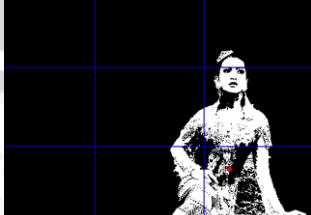
|          |  |   |              |           |               |
|----------|--|---|--------------|-----------|---------------|
| <p>3</p> |  <p>(Utomo, Lokomotif Bekasi, 2012)</p> |    | <p>Ya</p>    | <p>60</p> | <p>61,91</p>  |
| <p>4</p> |  <p>(Microsoft, 2013)</p>               |    | <p>Tidak</p> | <p>0</p>  | <p>126,80</p> |
| <p>5</p> |  <p>(Midori, 2007)</p>                 |   | <p>Tidak</p> | <p>0</p>  | <p>113,01</p> |
| <p>6</p> |  <p>(Unknown, Surfer_01, 2008)</p>    |  | <p>Tidak</p> | <p>0</p>  | <p>122,43</p> |
| <p>7</p> |  <p>(Peterson, Kayaking, 2013)</p>    |  | <p>Ya</p>    | <p>75</p> | <p>26,49</p>  |


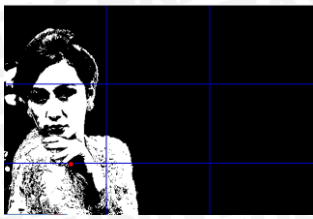

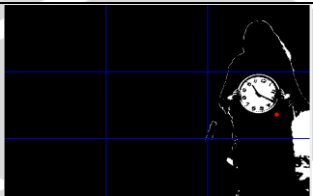

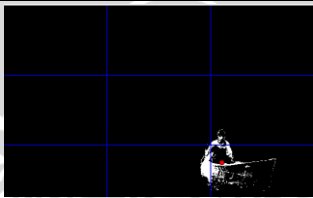

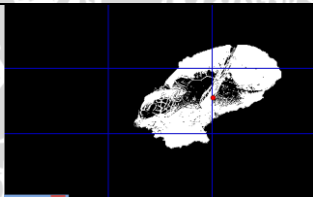

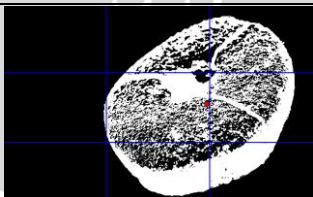


|    |   |  |       |    |        |
|----|---|--|-------|----|--------|
| 8  | <br>(Unknown, Brown Bear Alaska, 2011) |    | Ya    | 50 | 72,52  |
| 9  | <br>(McGowan, 2012)                    |    | Ya    | 75 | 41,91  |
| 10 | <br>(Ferryzal, 2009)                  |   | Ya    | 75 | 36,49  |
| 11 | <br>(Alichwan, 2013)                 |  | Tidak | 0  | 130,75 |
| 12 | <br>(Rananggono, 2014)               |  | Tidak | 0  | 91,66  |
| 13 | <br>(Unknown, Untitled)              |  | Ya    | 50 | 73,43  |


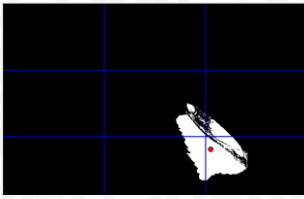

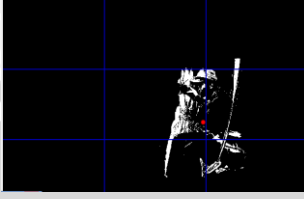

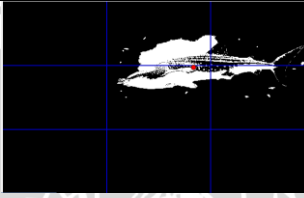

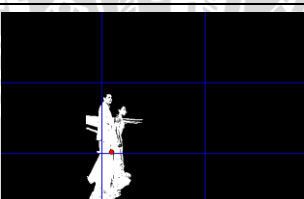

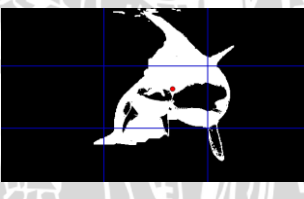

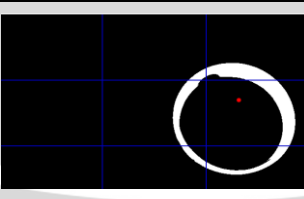
|    |   |   |       |    |        |
|----|---|---|-------|----|--------|
| 14 |  <p>(Wibhawa, 2010)</p>  |    | Ya    | 85 | 15,91  |
| 15 |  <p>(Unknown, flowers<br/>birds animals plants<br/>hummingbirds, 2014)</p> |    | Ya    | 75 | 40,15  |
| 16 |  <p>(Strathdee, 2014)</p>   |   | Tidak | 0  | 167,94 |
| 17 |  <p>(Clarke, 2014)</p>   |  | Tidak | 0  | 282,93 |
| 18 |  <p>(Nugroho, 2009)</p>  |  | Ya    | 60 | 51,33  |
| 19 |  <p>(Utomo, Melaju di<br/>Jalan Tol, 2012)</p>                           |  | Tidak | 0  | 104,13 |



|    |   |   |       |    |        |
|----|---|---|-------|----|--------|
| 20 |  <p>(Utomo, Welcome 103 To Jakarta Kota, 2012)</p> |    | Tidak | 0  | 115,32 |
| 21 |  <p>(Utomo, TM 7000 at Jakarta Kota, 2012)</p>     |    | Tidak | 0  | 171,67 |
| 22 |  <p>(Utomo, Gedung FISIP, 2012)</p>               |   | Tidak | 0  | 319,19 |
| 23 |  <p>(Utomo, Mengalahkan Musuhku, 2012)</p>       |  | Ya    | 80 | 31,12  |
| 24 |  <p>(Utomo, Kebaya 2, 2013)</p>                  |  | Ya    | 60 | 59,26  |

|           |   |   |              |           |               |
|-----------|---|---|--------------|-----------|---------------|
| <p>25</p> |  <p>(Utomo, Kebaya 3, 2013)</p>              |    | <p>Ya</p>    | <p>50</p> | <p>77,53</p>  |
| <p>26</p> |  <p>(Aviali, Untitled, 2014)</p>             |    | <p>Tidak</p> | <p>0</p>  | <p>150,96</p> |
| <p>27</p> |  <p>(Mamun, 2014)</p>                       |   | <p>Ya</p>    | <p>75</p> | <p>36,55</p>  |
| <p>28</p> |  <p>(Unknown, Loggerhead Turtle, 2014)</p> |  | <p>Ya</p>    | <p>85</p> | <p>19,83</p>  |
| <p>29</p> |  <p>(Zuhri, 2012)</p>                      |  | <p>Ya</p>    | <p>60</p> | <p>57,91</p>  |







|    |  |   |       |    |        |
|----|--|---|-------|----|--------|
| 30 | <br>(Irsan, 2010)                 |    | Ya    | 85 | 20,95  |
| 31 | <br>(S., 2011)                    |    | Ya    | 75 | 41,26  |
| 32 | <br>(Wolf, 2006)                  |    | Ya    | 75 | 39,17  |
| 33 | <br>(Setiawan, 2009)             |   | Ya    | 85 | 17,55  |
| 34 | <br>(Crawford, 2013)            |  | Tidak | 0  | 117,49 |
| 35 | <br>(Aviali, Untitled #2, 2014) |  | Ya    | 50 | 73,47  |






Tabel 5.1 Hasil pengujian 35 sampel foto dilakukan oleh komputer.







## 5.2 Pengujian Penilaian Foto Oleh Fotografer Profesional

Berikut adalah hasil penilaian terhadap 35 sampel foto yang dilakukan oleh fotografer. Namun perlu diingat bahwa penilaian bersifat subjektif karena setiap orang memiliki ketertarikan pada jenis fotografi yang berbeda.







| No. | Citra Uji  | Status <i>Rule Of Thirds</i> | Skor |
|-----|--|------------------------------|------|
| 1   |  <p>(Utomo, Kebaya Menatap, 2013)</p>     | Ya                           | 60   |
| 2   |  <p>(Utomo, Kunci Ganda, 2012)</p>       | Ya                           | 75   |
| 3   |  <p>(Utomo, Lokomotif Bekasi, 2012)</p> | Ya                           | 80   |
| 4   |  <p>(Microsoft, 2013)</p>               | Tidak                        | 0    |







|   |   |       |    |
|---|---|-------|----|
| 5 |  <p>(Midori, 2007)</p>                       | Tidak | 0  |
| 6 |  <p>(Unknown, Surfer_01, 2008)</p>           | Tidak | 0  |
| 7 |  <p>(Peterson, Kayaking, 2013)</p>          | Ya    | 90 |
| 8 |  <p>(Unknown, Brown Bear Alaska, 2011)</p> | Ya    | 90 |
| 9 |  <p>(McGowan, 2012)</p>                    | Ya    | 80 |

|           |   |              |            |
|-----------|---|--------------|------------|
| <p>10</p> |  <p>(Ferryzal, 2009)</p>   | <p>Ya</p>    | <p>70</p>  |
| <p>11</p> |  <p>(Alichwan, 2013)</p>   | <p>Tidak</p> | <p>0</p>   |
| <p>12</p> |  <p>(Rananggono, 2014)</p>   | <p>Tidak</p> | <p>0</p>   |
| <p>13</p> |  <p>(Unknown, Untitled)</p>  | <p>Tidak</p> | <p>0</p>   |
| <p>14</p> |  <p>(Wibhawa, 2010)</p>  | <p>Ya</p>    | <p>100</p> |
| <p>15</p> |  <p>(Unknown, flowers<br/>birds animals plants<br/>hummingbirds, 2014)</p> | <p>Tidak</p> | <p>0</p>   |






|    |   |       |    |
|----|---|-------|----|
| 16 | <br>(Strathdee, 2014)                            | Ya    | 50 |
| 17 | <br>(Clarke, 2014)                               | Ya    | 85 |
| 18 | <br>(Nugroho, 2009)                              | Ya    | 85 |
| 19 | <br>(Utomo, Melaju di Jalan Tol, 2012)         | Tidak | 0  |
| 20 | <br>(Utomo, Welcome 103 To Jakarta Kota, 2012) | Tidak | 0  |
| 21 | <br>(Utomo, TM 7000 at Jakarta Kota, 2012)     | Ya    | 80 |

|    |   |       |    |
|----|---|-------|----|
| 22 |  <p>(Utomo, Gedung FISIP, 2012)</p>        | Ya    | 50 |
| 23 |  <p>(Utomo, Mengalahkan Musuhku, 2012)</p> | Tidak | 0  |
| 24 |  <p>(Utomo, Kebaya 2, 2013)</p>           | Ya    | 50 |
| 25 |  <p>(Utomo, Kebaya 3, 2013)</p>          | Ya    | 50 |
| 26 |  <p>(Aviali, Untitled, 2014)</p>         | Ya    | 85 |



|    |   |    |    |
|----|---|----|----|
| 27 |  <p>(Mamun, 2014)</p>                      | Ya | 85 |
| 28 |  <p>(Unknown, Loggerhead Turtle, 2014)</p> | Ya | 70 |
| 29 |  <p>(Zuhri, 2012)</p>                     | Ya | 85 |
| 30 |  <p>(Irsan, 2010)</p>                    | Ya | 75 |
| 31 |  <p>(S., 2011)</p>                       | Ya | 85 |
| 32 |  <p>(Wolf, 2006)</p>                     | Ya | 80 |

|    |  |       |    |
|----|--|-------|----|
| 33 | <br>(Setiawan, 2009)            | Tidak | 0  |
| 34 | <br>(Crawford, 2013)            | Tidak | 0  |
| 35 | <br>(Aviali, Untitled #2, 2014) | Ya    | 85 |

Tabel 5.2 Hasil penilaian 35 sampel foto dilakukan oleh fotografer.

### 5.3 Persentase Keberhasilan Pengujian

Untuk persentase keberhasilan sistem secara keseluruhan diperoleh dari rata-rata nilai pada masing-masing foto sampel. Perhitungan persentase keberhasilan sistem dengan menyertakan foto sampel bernilai 0 adalah:

$$\begin{aligned}
 \text{persentase} &= \frac{\text{total nilai}}{\text{jumlah sampel}} \times 100\% \\
 &= \frac{1350}{35} \times 100\% \\
 &= 43,71\%
 \end{aligned}$$

### 5.4 Analisis Faktor Kegagalan

Pada pengekseskuan program penilai komposisi *rule of thirds* pada fotografi ini tidak selalu didapat hasil yang optimal. Hal ini disebabkan oleh beberapa faktor antara lain:

1. Segmentasi *grabcut* tidak selalu optimal saat memisahkan subjek dengan latar belakang karena faktor pencahayaan pada foto yang terlalu gelap atau terlalu terang. Serta jika ukuran subjek terlalu besar seperti pada foto nomor 13, maka



hasil segmentasi *grabcut* juga akan membesar. Hal ini tentunya berpengaruh saat proses penentuan *centroid* karena bidang yang akan dievaluasi menjadi lebih luas daripada seharusnya.

2. Segmentasi *grabcut* mengalami kesulitan saat memisahkan subjek dengan latar belakangnya, kasus ini terjadi pada foto nomor 8. Hal ini terjadi karena kontras subjek kurang baik jadi *grabcut* menganggap subjek sama dengan latar belakang.
3. Posisi *centroid* begitu pula subjek sangat berpengaruh pada hasil penilaian, begitu pula bagi subjek yang terletak pada kiri atau kanan bidang foto akan memiliki nilai lebih besar daripada subjek yang terletak pada tengah-tengah bidang foto karena jarak dari *centroid* ke titik perpotngan akan semakin dekat jika subjek berada di kiri atau kanan foto daripada di tengah-tengah foto.
4. Skor tertinggi yang didapat dari pengujian 35 sampel adalah 85, hal ini terjadi karena tidak ada *centroid* subjek yang terletak tepat di salah satu titik perpotongan.



## BAB VI PENUTUP

### 6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

1. Posisi subjek dalam foto sangat berpengaruh pada hasil penilaian.
2. Jarak dari *centroid* ke salah satu perpotongan berpengaruh pada nilai yang diberikan.
3. Jika *centroid* subjek berada jauh dari salah satu titik perpotongan tetapi ada bagian objek yang berada di salah satu titik perpotongan, maka nilai yang ditampilkan adalah 0 (nol) karena jarak *centroid* jauh dari titik perpotongan.
4. Semakin dekat *centroid* subjek dengan salah satu titik perpotongan semakin besar nilai yang ditampilkan.
5. Jika terdapat lebih dari satu subjek dalam foto, maka hanya satu subjek yang dapat dinilai dikarenakan keterbatasan segmentasi *grabcut*.
6. Posisi *centroid* sangat bergantung pada hasil segmentasi *grabcut*, jika segmentasi tidak bekerja optimal, maka subjek akan menjadi lebih besar dari seharusnya karena area sekitar subjek juga ikut tersegmentasi. Jika kontras subjek kurang juga akan mengakibatkan segmentasi tidak optimal.

### 6.2 Saran

Terdapat beberapa hal yang perlu dikembangkan dalam program yang telah dibangun oleh penulis. Oleh karena itu jika akan dikembangkan pada penelitian berikutnya, maka saran yang dapat penulis berikan adalah sebagai berikut:

1. Penambahan beberapa fitur penting seperti, penilaian warna, penilaian pencahayaan, penilaian fokus foto;
2. Pengembangan versi *mobile* untuk diaplikasikan di ponsel pintar;
3. Penambahan fitur pemotongan otomatis agar foto yang telah dinilai dapat dipotong sehingga komposisinya sesuai dengan *rule of thirds*;



## DAFTAR PUSTAKA

- Afsar, J. (2013). *Center of Gravity (Centroid)*. Dipetik Desember 26, 2013, dari Engineering Intro: <http://www.engineeringintro.com/2013/01/center-of-gravity-centroid/>
- Alichwan, R. (2013). *KA Serayu silang KA Gajayana*. Jakarta. Dipetik Agustus 15, 2014, dari [https://www.flickr.com/photos/rofiq\\_alichwan\\_photo/8451105337/](https://www.flickr.com/photos/rofiq_alichwan_photo/8451105337/)
- Aviali, E. (2014). *Untitled*. Athens. Dipetik Agustus 15, 2014, dari [http://static.tumblr.com/qyxkvlf/3XFmcxdnb/tumblr\\_mcx7cvt74c1rks6ueo1\\_1280.jpg](http://static.tumblr.com/qyxkvlf/3XFmcxdnb/tumblr_mcx7cvt74c1rks6ueo1_1280.jpg)
- Aviali, E. (2014). *Untitled #2*. Athens. Dipetik Agustus 15, 2014, dari <http://dustoncrustphoto.tumblr.com/image/66008128784>
- Baatz, W., & Willinghamofer, H. (1997). *Photography: An Illustrated Historical Overview*. New York: Barron's Educational Series.
- Beer, F. P., E. Russell Johnston, J., Mazurek, D. F., & Eisenber, E. R. (2010). *Vector mechanics for engineers. Statics and dynamics* (9th ed.). New York: McGraw-Hill.
- Bourke, P. (1988). *Calculating the area and centroid of a polygon*. Dipetik Februari 15, 2014, dari Polygons and meshes: <http://paulbourke.net/geometry/polygonmesh>
- Burger, W., & Burge, M. (2009). *Principles of Digital ImageProcessing Fundamental Techniques*. London: Springer.
- Clarke, R. (2014). *Longboard At Sunset*. Dipetik Agustus 15, 2014, dari <http://abackgrounds.com/surfing-wallpaper-longboard/>
- Cochran, W. G. (2010). *Teknik Penarikan Sampel* (3rd ed.). Jakarta: UI-Press.
- Collins. (2011). *Collins English Dictionary* (11th ed.). London: HarperCollins.
- Crawford, L. (2013). *Killer Whales Are Non-Human Persons*. Dipetik Agustus 15, 2014, dari <http://greymattersjournal.com/killer-whales-are-non-human-persons/>
- Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying Aesthetics in Photographic images Using a Computational Approach. *Computer Vision – ECCV 2006 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III*. 3953, hal. 288-301. Graz: Springer Berlin Heidelberg.
- Ferryzal. (2009). *Bersama Payung #2*. Dipetik Agustus 15, 2014, dari <http://ferryexpose.blogspot.com/2009/11/anunya-kami.html>

- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing* (2nd ed.). Upper Saddle River: Prentice Hall.
- Irsan, M. (2010). *It's Papua*. Fakfak. Dipetik Agustus 15, 2014, dari <http://www.fotografer.net/galeri/view.php?id=1170737>
- Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River: Prentice Hall.
- Jensen, B. F. (2005). *Monte Carlo Integration*. Dipetik Juni 16, 2014, dari <http://owww.phys.au.dk/~bfj/numeric/MonteCarloInt.pdf>
- Ke, Y., Tang, X., & Jing, F. (2006). The Design of High-Level Features for Photo Quality Assessment. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume:1*, hal. 419-426. New York: IEEE Xplore Digital Library.
- Kurnianto, A., Sumpeno, S., & Hariadi, M. (2010). *Klasifikasi Citra Berdasarkan Parameter Estetika Menggunakan Metode ANFIS*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Liu, L., Chen, R., Wolf, L., & Cohen-Or, D. (2010). Optimizing Photo Composition. *Computer Graphics Forum, Volume 29*(Issue 2), 469-478.
- Luo, Y., & Tang, X. (2008). Photo and Video Quality evaluation: Focusing on the Subject. *Computer Vision – ECCV 2008 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III. 5304*, hal. 386-399. Marseille: Springer Berlin Heidelberg.
- Mahlke, S. (2008). Visual aesthetics and the user experience. *Dagstuhl Seminar Proceedings The Study of Visual Aesthetics in Human-Computer Interaction* (hal. 1-6). Dagstuhl: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.
- Mai, L., Le, H., Niu, Y., & Liu, F. (2011). Rule of Thirds Detection from Photograph. *Multimedia (ISM), 2011 IEEE International Symposium* (hal. 91-96). Dana Point: IEEE Xplore Digital Library.
- Maleš, M., Heđi, A., & Grgić, M. (2012). Compositional rule of thirds detection. *ELMAR, 2012 Proceedings* (hal. 41-44). Zadar: IEEE Xplore Digital Library.
- Mamun, S. (2014). *Menepi Sejenak*. Cirebon. Dipetik Agustus 15, 2014, dari <http://www.fotografer.net/galeri/view.php?id=1938820>



- McGowan, E. (2012). *Hide n Seek*. Corona. Dipetik Agustus 15, 2014, dari [http://www.lomogracinha.com.br/wp-content/uploads/2012/10/inspiracao\\_crianças\\_05.jpg](http://www.lomogracinha.com.br/wp-content/uploads/2012/10/inspiracao_crianças_05.jpg)
- Microsoft. (2013). *Hydrangeas*.
- Midori. (2007). *Tugu Monas*. Jakarta. Dipetik Agustus 15, 2014, dari <http://commons.wikimedia.org/wiki/File:Monas.JPG>
- Nakano, A. (2013). *Monte Carlo Basics*. Dipetik Agustus 06, 2014, dari University Of Southern Carolina Collaboratory For Advanced Computing And Simulations: [cacs.usc.edu/education/phys516/01-1mcbasics.pdf](http://cacs.usc.edu/education/phys516/01-1mcbasics.pdf)
- Nugroho, A. (2009). *Gunung Batur*. Kintamani. Dipetik Agustus 15, 2014, dari <http://www.fotografer.net/galeri/view.php?id=904220>
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. 9(1), 62-66.
- Page, J. (2009). *Centroid of Triangle*. Dipetik Desember 24, 2013, dari Open Math Reference: <http://www.mathopenref.com/trianglecentroid.html>
- Peterson, D. (2012). *Why Does the Rule of Thirds Work?* Dipetik Januari 05, 2014, dari Digital Photo Secrets: <http://www.digital-photo-secrets.com/tip/2742/why-does-the-rule-of-thirds-work/>
- Peterson, D. (2013). *18 Composition Rules For Photos That Shine*. Dipetik Maret 03, 2014, dari Digital Photo Secrets: <http://www.digital-photo-secrets.com/tip/3372/18-composition-rules-for-photos-that-shine/>
- Peterson, D. (2013). *Kayaking*. Melbourne. Dipetik Agustus 15, 2014, dari <http://www.digital-photo-secrets.com/tip/2742/why-does-the-rule-of-thirds-work/>
- Polina, A. M., & Siang, J. J. (2009). *Cara Cepat Menyusun Skripsi Jurusan Informatika/Komputer*. Yogyakarta: ANDI.
- Protter, M. H., & Morrey, J. C. (1970). *College Calculus with Analytic Geometry* (2nd ed.). Reading: Addison-Wesley.
- Rananggono, D. (2014). *KA argo muria #1*. Batang. Dipetik Agustus 15, 2014, dari <https://www.flickr.com/photos/rananggono/11774417385/in/pool-35291953@N00/>
- Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2004*, 23(3), 309-314.

- S., H. R. (2011). *landscaper*. Kediri. Dipetik Agustus 15, 2014, dari [www\\_fotografer\\_net\\_galeri\\_view\\_php\\_id=1408079](http://www.fotografer_net_galeri_view_php_id=1408079)
- Setiawan, A. (2009). *Malioboro In Love*. Yogyakarta. Dipetik Agustus 15, 2014, dari [www\\_fotografer\\_net\\_galeri\\_view\\_php\\_id=852039](http://www.fotografer_net_galeri_view_php_id=852039)
- Spencer, D. A. (1973). *The Focal Dictionary of Photographic Technologies*. Mew Jersey: Prentice Hall.
- Strathdee, S. (2014). *Snowshoeing*. Dipetik Agustus 15, 2014, dari <http://www.free-desktop-backgrounds.net/Nature-landscapes-wallpapers/Snow-wallpapers/Winter-with-lots-of-snow.html>
- Supranto, J. (2007). *Teknik Sampling Untuk Survei dan Eksperimen*. Jakarta: Rineka Cipta.
- Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., & Wijanarto. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: ANDI.
- Tjin, E. (2011). *Kamera DSLR Itu Mudah!* Jakarta: Bukuné.
- Tong, H., Li, M., Zhang, H.-J., He, J., & Zhang, C. (2004). Classification of Digital Photos Taken by Photographers or Home Users. *Advances in Multimedia Information Processing - PCM 2004 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30 - December 3, 2004. Proceedings, Part I*. 3331, hal. 198-205. Tokyo: Springer Berlin Heidelberg.
- Unknown. (2008). *Surfer\_01*. Dipetik Agustus 15, 2014, dari <http://munjalindra.com/2008/09/26/the-rule-of-thirds-komposisi-layout-fotografi.html>
- Unknown. (2011). *Brown Bear Alaska*. Dipetik Agustus 15, 2014, dari <http://www.wallpapershdi.com/wallpaper/2816/brown-bear-alaska-widescreen.html>
- Unknown. (2014). *flowers birds animals plants hummingbirds*. Dipetik Agustus 15, 2014, dari [http://wallpaperswa.com/Animals/Birds/flowers\\_birds\\_animals\\_plants\\_hummingbirds\\_depth\\_of\\_field\\_2112x1320\\_wallpaper\\_3189/download\\_2112x1320](http://wallpaperswa.com/Animals/Birds/flowers_birds_animals_plants_hummingbirds_depth_of_field_2112x1320_wallpaper_3189/download_2112x1320)
- Unknown. (2014). *Loggerhead Turtle*. Dipetik Agustus 15, 2014, dari <http://www.wallpapersdb.org/animals/loggerhead-turtle-wallpaper-2598.htm>
- Unknown. (t.thn.). *Untitled*.
- Utomo, T. P. (2012). *Gedung FISIP*. Malang.
- Utomo, T. P. (2012). *Kunci Ganda*. Malang.



- Utomo, T. P. (2012). *Lokomotif Bekasi*. Bekasi.
- Utomo, T. P. (2012). *Melaju di Jalan Tol*. Bekasi.
- Utomo, T. P. (2012). *Mengalahkan Musuhku*.
- Utomo, T. P. (2012). *TM 7000 at Jakarta Kota*. Bekasi.
- Utomo, T. P. (2012). *Welcome 103 To Jakarta Kota*. Bekasi.
- Utomo, T. P. (2013). *Kebaya 2*. Malang.
- Utomo, T. P. (2013). *Kebaya 3*. Malang.
- Utomo, T. P. (2013). *Kebaya Menatap*. Malang.
- Waterhouse, J. (2011). *Photography Rule of Thirds*. Dipetik Maret 31, 2014, dari Ultimate Photo Tips: <http://www.ultimate-photo-tips.com/photography-rule-of-thirds.html>
- Wibhawa, A. (2010). *Bening Yah*. Jakarta. Dipetik Agustus 15, 2014, dari <http://www.fotografer.net/galeri/view.php?id=1143979>
- Wolf, Z. (2006). *One of two resident male whale sharks in the Georgia Aquarium in the United States*. Georgia. Dipetik Agustus 15, 2014, dari [http://commons.wikimedia.org/wiki/File:Male\\_whale\\_shark\\_at\\_Georgia\\_Aquarium.jpg](http://commons.wikimedia.org/wiki/File:Male_whale_shark_at_Georgia_Aquarium.jpg)
- Young, N. S. (2010). *Canon EOS 60D: From Snapshots to Great Shots*. San Francisco: Peachpit Press.
- Zuhri, S. (2012). *i love donut*. Dipetik Agustus 15, 2014, dari <http://www.fotografer.net/galeri/view.php?id=1798928>

## LAMPIRAN

**Nama berkas: rotp.pro**

```
-----  
#  
# Project created by QtCreator 2014-06-10T10:59:30  
#  
-----  
  
QT += core gui  
  
INCLUDEPATH += C://opencv-2.4.5//mybuild//install//include  
  
LIBS += C://opencv-2.4.5//mybuild//bin//*.dll  
  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  
  
TARGET = rotp  
TEMPLATE = app  
  
SOURCES += main.cpp\  
           rotimage.cpp \  
           rotspinbox.cpp  
  
HEADERS += \  
          rotimage.h \  
          rotspinbox.h
```





**Nama berkas: rotimage.h**

```

#ifndef ROTIMAGE_H
#define ROTIMAGE_H

#include <QApplication>
#include <QWidget>
#include <QImage>
#include <QPainter>
#include <QPixmap>
#include <QLabel>
#include <QMessageBox>
#include <QString>
#include <opencv2/opencv.hpp>

using namespace cv;

class ROTImage : public QLabel
{
    Q_OBJECT
public:
    explicit ROTImage(QWidget *parent = 0);
    int getHeight();
    int getWidth();

signals:
    void imageWidth(int width);
    void imageHeight(int height);
    void newlyOpen();

public slots:
    int openFilename();
    void drawOverlay();
    void drawGrabcutBoundary();
    void applyGrabcut();
    void applyGrayOtsu();
    void renderImage();
    void setGrabcut_Xbegin(int pixel);
    void setGrabcut_Ybegin(int pixel);
    void setGrabcut_Xend(int pixel);
    void setGrabcut_Yend(int pixel);
    int checkRuleofThirds();

private:
    //for tolerance and scoring
    double tolerance[8] = {0.001, 0.008, 0.020, 0.030, 0.045, 0.060,
0.090, 0.100};
    int score[9] = {100, 95, 90, 85, 80, 75, 60, 50, 0};

    // for grabcut
    Mat result; // segmentation result (4 possible values)
    Mat bgModel, fgModel; // the models (internally used)
    int grabcut_xbegin, grabcut_ybegin, grabcut_xend, grabcut_yend;
    Rect grabcut_rect;

    // centroid variables
    double intersect_x[4], intersect_y[4];
    double centroid_x, centroid_y;
    Mat image, tmp;
    QImage disp;
    QPainter painter;

```

```
double x_accumulative, y_accumulative;
double x_temp, y_temp;
double image_hypotenuse, distance_hypotenuse;
int count;
QString string;
QMessageBox messagebox;

//for sampling
double
proportion,min_sample,trust,deviation,q,tpq,grab_populate,whole_popula
te;
};

#endif // ROTIMAGE_H
```





**Nama berkas: rotspinbox.h**

```

#ifndef ROTSPINBOX_H
#define ROTSPINBOX_H

#include <QSpinBox>

class ROTSpinBox : public QSpinBox
{
    Q_OBJECT
public:
    explicit ROTSpinBox(QWidget *parent = 0);

signals:

public slots:
    void setValueMax(int val);
    void resetValue();
};

#endif // ROTSPINBOX_H
```



**Nama berkas: main.cpp**

```

#include <QApplication>
#include <QGridLayout>
#include <QPushButton>
#include <QLabel>
#include <QAction>
#include <QMenu>
#include <QMenuBar>
#include <QWidget>
#include "rotpinbox.h"
#include "rotimage.h"

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    ROTImage *rule = new ROTImage();

    ROTSpinBox *spinbox_Xstart = new ROTSpinBox();
    ROTSpinBox *spinbox_Xend = new ROTSpinBox();
    ROTSpinBox *spinbox_Ystart = new ROTSpinBox();
    ROTSpinBox *spinbox_Yend = new ROTSpinBox();
    QLabel *label_start = new QLabel("Grabut Start Point (x,y)");
    QLabel *label_end = new QLabel("Grabut End Point (x,y)");

    QPushButton *button_file = new QPushButton("Open Image");
    QPushButton *button_grabcut = new QPushButton("Apply Grabcut");
    QPushButton *button_otsu = new QPushButton("Apply Grayscale and
Otsu Threshold");
    QPushButton *button_overlay = new QPushButton("Draw Rule of Thirds
Overlay");
    QPushButton *button_rot = new QPushButton("Rule of Thirds Check");

    QGridLayout *layout = new QGridLayout();
    layout->addWidget(button_file,0,0,1,2,Qt::AlignLeft);
    layout->addWidget(label_start,1,0,1,2,Qt::AlignLeft);
    layout->addWidget(spinbox_Xstart,2,0,1,1,Qt::AlignLeft);
    layout->addWidget(spinbox_Ystart,2,1,1,1,Qt::AlignLeft);
    layout->addWidget(label_end,3,0,1,2,Qt::AlignLeft);
    layout->addWidget(spinbox_Xend,4,0,1,1,Qt::AlignLeft);
    layout->addWidget(spinbox_Yend,4,1,1,1,Qt::AlignLeft);
    layout->addWidget(button_grabcut,5,0,1,2,Qt::AlignLeft);
    layout->addWidget(button_otsu,6,0,1,2,Qt::AlignLeft);
    layout->addWidget(button_overlay,7,0,1,2,Qt::AlignLeft);
    layout->addWidget(button_rot,8,0,2,1,Qt::AlignLeft);
    layout->addWidget(rule,0,3,10,1,Qt::AlignLeft);

    QWidget *display = new QWidget();
    display->setLayout(layout);
    display->show();

    QObject::connect(rule,SIGNAL(imageWidth(int)),spinbox_Xstart,SLOT(setV
alueMax(int)));

    QObject::connect(rule,SIGNAL(imageWidth(int)),spinbox_Xend,SLOT(setVal
ueMax(int)));

    QObject::connect(rule,SIGNAL(imageHeight(int)),spinbox_Ystart,SLOT(set
ValueMax(int)));

```



```

QObject::connect(rule, SIGNAL(imageHeight(int)), spinbox_Yend, SLOT(setValueMax(int)));

QObject::connect(rule, SIGNAL(imageWidth(int)), spinbox_Xend, SLOT(setValue(int)));

QObject::connect(rule, SIGNAL(imageHeight(int)), spinbox_Yend, SLOT(setValue(int)));

QObject::connect(rule, SIGNAL(newlyOpen()), spinbox_Xstart, SLOT(resetValue()));

QObject::connect(rule, SIGNAL(newlyOpen()), spinbox_Ystart, SLOT(resetValue()));

QObject::connect(spinbox_Xstart, SIGNAL(valueChanged(int)), rule, SLOT(setGrabcut_Xbegin(int)));

QObject::connect(spinbox_Ystart, SIGNAL(valueChanged(int)), rule, SLOT(setGrabcut_Ybegin(int)));

QObject::connect(spinbox_Xend, SIGNAL(valueChanged(int)), rule, SLOT(setGrabcut_Xend(int)));

QObject::connect(spinbox_Yend, SIGNAL(valueChanged(int)), rule, SLOT(setGrabcut_Yend(int)));

QObject::connect(button_grabcut, SIGNAL(clicked()), rule, SLOT(applyGrabcut()));

QObject::connect(button_otsu, SIGNAL(clicked()), rule, SLOT(applyGrayOtsu()));

QObject::connect(button_file, SIGNAL(clicked()), rule, SLOT(openFilename()));

QObject::connect(button_overlay, SIGNAL(clicked()), rule, SLOT(drawOverlay()));

QObject::connect(button_rot, SIGNAL(clicked()), rule, SLOT(checkRuleofThirs()));
    return a.exec();
}

```

**Nama berkas: rotimage.cpp**

```
#include "rotimage.h"
#include <stdlib.h>
#include <math.h>
#include <QFileDialog>
#include <QMessageBox>
#include <QDateTime>
#include <QPen>
#include <QPainter>
#include <QString>

#define MAX_WIDTH 640
#define MAX_HEIGHT 640
#define OVERLAY_COLOR 0,0,200 // dead blue
#define GRABCUT_COLOR 200,0,0 // dead red

using namespace cv;

ROTimage::ROTimage(QWidget *parent) : QLabel(parent)
{
}

int ROTimage::openFilename() {
    image.release();
    tmp.release();

    image = imread(QFileDialog::getOpenFileName(this, tr("Open Image"),
        "./images", tr("Image Files (*.jpeg; *.jpg; *.bmp; *.png)")).toStdString());
    if(!image.data || image.cols>MAX_WIDTH || image.rows>MAX_HEIGHT) {
        messagebox.setText("Image not found or image dimension is
        greater than 640x640 pixels");
        messagebox.exec();
        return(0);
    }
    else{
        // centroid variables
        intersect_x[4]={ }, intersect_y[4]={ };
        centroid_x=0, centroid_y=0;
        x_accumulative=0, y_accumulative=0;
        x_temp=0, y_temp=0;
        count=0;

        emit newlyOpen();
        emit imageWidth(image.cols);
        emit imageHeight(image.rows);
        intersect_x[0] = image.cols/3;    intersect_y[0] =
        image.rows/3;
        intersect_x[1] = image.cols/3;    intersect_y[1] =
        2*image.rows/3;
        intersect_x[2] = 2*image.cols/3; intersect_y[2] =
        image.rows/3;
        intersect_x[3] = 2*image.cols/3; intersect_y[3] =
        2*image.rows/3;
        renderImage();
        return(0);
    }
}
```



```

void ROTImage::renderImage() {
    switch (image.type()) {
    case CV_8UC1:
        cvtColor(image, tmp, CV_GRAY2RGB);
        break;
    case CV_8UC3:
        cvtColor(image, tmp, CV_BGR2RGB);
        break;
    }
    assert(tmp.isContinuous());

    disp = QImage(tmp.data, tmp.cols, tmp.rows, tmp.cols*3,
QImage::Format_RGB888);
    setPixmap(QPixmap::fromImage(disp));
    update();
}

void ROTImage::applyGrabcut() {
    try
    {
        // GrabCut segmentation
        grabCut(image, // input image
            result, // segmentation result
            grabcut_rect, // rectangle containing foreground
            bgModel, fgModel, // models
            1, // number of iterations
            GC_INIT_WITH_RECT); // use rectangle
        // Get the pixels marked as likely foreground
        compare(result, GC_PR_FGD, result, CMP_EQ);
        // Generate output image
        Mat foreground(image.size(), CV_8UC3, Scalar(0,0,0));
        image.copyTo(foreground, result); // bg pixels not copied
        image = foreground;
        renderImage();
    }
    catch(...)
    {
        QMessageBox::setText("Open an image first before apply grabcut
segmentation \
or draw grabcut coordinates first");
        QMessageBox::exec();

        return;
    }
}

void ROTImage::applyGrayOtsu() {
    try
    {
        Mat foreground(image.size(), CV_8UC3, Scalar(0,0,0));
        image.copyTo(foreground, result); // bg pixels not copied
        Mat img_gray;
        Mat img_bw;
        cvtColor(foreground, img_gray, CV_RGB2GRAY);

        threshold(img_gray, img_bw, 0, 255, CV_THRESH_BINARY|CV_THRESH_OTSU);
        image = img_bw;
        renderImage();
    }
}

```

```
catch(...)\n{\n    messagebox.setText("Open an image first or apply the grabcut \\  
segmentation first");\n    messagebox.exec();\n    return;\n}\n}\n\nvoid ROTImage::drawOverlay(){\n    try\n    {\n        QPen pen;\n        painter.begin(&disp);\n        pen.setWidth(2);\n        pen.setColor(qRgb(OVERLAY_COLOR));\n        painter.setPen(pen);\n        painter.drawLine(0, image.rows/3, image.cols, image.rows/3);\n        //draw upper horizontal third line\n        painter.drawLine(0, 2*image.rows/3, image.cols,\n        2*image.rows/3); //draw lower horizontal third line\n        painter.drawLine(image.cols/3, 0, image.cols/3,\n        image.rows); //draw left vertical third line\n        painter.drawLine(2*image.cols/3, 0, 2*image.cols/3,\n        image.rows); //draw right vertical third line\n        painter.end();\n        setPixmap(QPixmap::fromImage(disp));\n        update();\n    }\n\n    catch(...)\n    {\n        messagebox.setText("Open an image first before use grabcut \\  
segmentation");\n        messagebox.exec();\n        return;\n    }\n}\n\nvoid ROTImage::drawGrabcutBoundary(){\n    renderImage();\n    QPen pen;\n    painter.begin(&disp);\n    pen.setColor(qRgb(GRAB CUT_COLOR));\n    pen.setWidth(4);\n    painter.setPen(pen);\n\n    painter.drawLine(grabcut_xbegin, grabcut_ybegin, grabcut_xbegin,\n    grabcut_yend); //draw left vertical line\n    painter.drawLine(grabcut_xend, grabcut_ybegin, grabcut_xend,\n    grabcut_yend); //draw right vertical line\n    painter.drawLine(grabcut_xbegin, grabcut_ybegin, grabcut_xend,\n    grabcut_ybegin); //draw upper horizontal line\n    painter.drawLine(grabcut_xbegin, grabcut_yend, grabcut_xend,\n    grabcut_yend); //draw lower horizontal line\n    painter.end();\n    setPixmap(QPixmap::fromImage(disp));\n    update();\n}\n}
```



```

void ROTimage::setGrabcut_Xbegin(int pixel){
    grabcut_xbegin = pixel;
    grabcut_rect.x = grabcut_xbegin;
    drawGrabcutBoundary();
}

void ROTimage::setGrabcut_Ybegin(int pixel){
    grabcut_ybegin = pixel;
    grabcut_rect.y = grabcut_ybegin;
    drawGrabcutBoundary();
}

void ROTimage::setGrabcut_Xend(int pixel){
    grabcut_xend = pixel;
    grabcut_rect.width = grabcut_xend-grabcut_xbegin;
    drawGrabcutBoundary();
}

void ROTimage::setGrabcut_Yend(int pixel){
    grabcut_yend = pixel;
    grabcut_rect.height = grabcut_yend-grabcut_ybegin;
    drawGrabcutBoundary();
}

int ROTimage::checkRuleofThirds(){

    //sampling size formula
    grab_populate = (grabcut_xend-grabcut_xbegin)*(grabcut_yend-
grabcut_ybegin);
    whole_populate = image.cols*image.rows;
    proportion = grab_populate/whole_populate;
    q = 1 - proportion;
    trust = 1.96;
    tpq = pow(trust,2)*proportion*q;
    deviation = pow(0.05,2);
    min_sample = (tpq/deviation)/(1+(1/whole_populate)*((q/deviation)-
1));
    qDebug("grabcut populate is %f and minimum sample is %f and tpq is
%f",grab_populate,min_sample,tpq);

    try
    {
        //centroid determine formula
        srand(QDateTime::currentMsecsSinceEpoch());
        while(count <= min_sample){

            x_temp = grabcut_xbegin + rand()%(grabcut_xend-
grabcut_xbegin);
            y_temp = grabcut_ybegin + rand()%(grabcut_yend-
grabcut_ybegin);
            //qDebug("xtemp = %f and ytemp = %f",x_temp,y_temp);
            Scalar color = image.at<uchar>(y_temp, x_temp);
            if (color.val[0]==255){ // the pixel[temp] is white!!
                x_accumulative = x_accumulative + x_temp;
                y_accumulative = y_accumulative + y_temp;
                count++;
            }
        }
        //averaging the accumulative result with the number of sample
        centroid_x = x_accumulative/min_sample;
    }
}

```

```

centroid_y = y_accumulative/min_sample;

//draw centroid mark
painter.begin(&disp);
painter.setRenderHint(QPainter::Antialiasing, true);
painter.setBrush(QBrush(Qt::red, Qt::SolidPattern));
painter.drawEllipse(centroid_x-5,centroid_y-5,10,10);
painter.end();
setPixmap(QPixmap::fromImage(disp));
update();

qDebug("Centroid (x,y): %f %f",centroid_x,centroid_y);

image_hypotenuse = sqrt(pow(image.cols,2) +
pow(image.rows,2));

// rule of thirds check, intersection rule
for (int i=0; i<4; i++){

    //check rule of thirds intersection pass thru
    Scalar color = image.at<uchar>(intersect_x[i],
intersect_y[i]);
    if (color.val[0]==255){ // the pixel[temp] is white!!
        painter.begin(&disp);
        painter.setRenderHint(QPainter::Antialiasing, true);
        painter.setBrush(QBrush(Qt::yellow,
Qt::SolidPattern));
        painter.drawEllipse(intersect_x[i]-5,intersect_y[i]-
5,10,10);

        painter.end();
        setPixmap(QPixmap::fromImage(disp));
        update();

        qDebug("Rule Of Thirds: Yes. Pass thru at (%.2f,
%.2f)",intersect_x[i],intersect_y[i]);
    }

    QPen pen;
    painter.begin(&disp);
    pen.setWidth(2);
    pen.setStyle(Qt::DashLine);
    pen.setColor(qRgb(0,255,0));
    painter.setPen(pen);
    painter.drawLine(centroid_x, centroid_y,
intersect_x[i],intersect_y[i]); //draw dashed line from centroid to
all intersection
    painter.end();
    setPixmap(QPixmap::fromImage(disp));
    update();

    for (int j=0; j<8; j++){
        distance_hypotenuse = sqrt(pow(centroid_x-
intersect_x[i],2)+pow(centroid_y-intersect_y[i],2));

        if (distance_hypotenuse <
tolerance[j]*image_hypotenuse){
            messagebox.setText(string.sprintf("Rule of Thirds:
Yes. \
\nCentroid is at
coordinates (%.2f, %.2f). \

```





```
        return(0);  
    }  
}  
  
int ROTImage::getHeight() {  
    return(image.rows);  
}  
  
int ROTImage::getWidth() {  
    return(image.cols);  
}
```





**Nama berkas: rotspinbox.cpp**

```
#include "rotspinbox.h"

ROTSpinBox::ROTSpinBox(QWidget *parent) :
    QSpinBox(parent)
{
}

void ROTSpinBox::setValueMax(int val) {
    setMaximum(val);
}

void ROTSpinBox::resetValue() {
    setValue(0);
}
```

UNIVERSITAS BRAWIJAYA

