

PERANCANGAN DAN PEMBUATAN APLIKASI UNTUK MENDESAIN KARTU UCAPAN

Rudy Adipranata¹, Liliana², Gunawan Iteh
Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya
e-mail : rudya@peter.petra.ac.id¹, lilian@peter.petra.ac.id²

ABSTRAK

Kartu ucapan adalah salah satu media yang digunakan untuk menunjukkan perhatian kepada teman, kerabat serta orang-orang terdekat. Setiap orang memiliki caranya masing-masing dalam mengekspresikan dirinya. Peluang inilah yang ditangkap oleh para produsen kartu ucapan dengan berlomba-lomba menciptakan berbagai desain kartu ucapan. Akan tetapi, tetap saja belum semua keinginan *user* terjawab oleh kartu ucapan konvensional. Ada kalanya *user* ingin membuat desain kartu ucapan yang sesuai dengan *style* dan seleranya. Di sinilah letak keterbatasan kartu ucapan konvensional. Selain itu kita juga tidak dapat memasukkan gambar maupun foto yang kita kehendaki pada kartu ucapan ini.

Untuk mengatasi hal tersebut maka dikembangkan suatu aplikasi yang memungkinkan *user* untuk mendesain kartu ucapan sesuai dengan keinginannya. Dengan demikian *user* memiliki keleluasaan untuk mengekspresikan dirinya melalui kartu ucapan yang dikirimkannya. Adapun fasilitas pada aplikasi yang dikembangkan ini adalah fasilitas untuk memasukkan teks atau gambar bebas, fasilitas pengolahan citra digital yaitu pengaturan *brightness*, *contrast* serta *color balance*. Untuk memperbaiki kualitas gambar yang kurang baik terdapat fasilitas *histogram equalization*, *sharpening* serta *smoothing*. Selain itu juga terdapat fasilitas untuk memberikan efek pada citra digital yaitu negatif, *grayscale* serta *sepia*.

Untuk mempermudah *user*, aplikasi ini juga menyediakan beberapa *template* yang siap untuk dipakai. Hasil desain kartu ucapan dapat dicetak di segala ukuran kertas sesuai keinginan *user*.

Kata kunci: kartu ucapan, desain kartu, pengolahan citra digital.

1. PENDAHULUAN

Untuk menyampaikan pesan di hari-hari istimewa, salah satu cara yang sering dilakukan oleh kebanyakan orang adalah dengan mengirimkan kartu ucapan. Melalui kartu ucapan kita dapat tetap menjaga hubungan dengan teman, saudara, keluarga, maupun relasi bisnis. Desain kartu yang dipilih dapat menggambarkan pribadi pengirimnya, demikian pula dengan kata-kata ucapan di dalamnya. Peluang inilah yang ditangkap oleh para produsen kartu ucapan dengan berlomba-lomba merancang berbagai macam desain kartu ucapan. Hasilnya dapat kita lihat begitu banyak kartu ucapan yang beredar di pasaran, baik itu kartu Natal, kartu Idul Fitri, kartu Tahun Baru Imlek, dll.

Tetapi ada kalanya *user* ingin membuat desain kartu ucapan yang sesuai dengan *style* dan selernya. Di sinilah letak keterbatasan kartu ucapan konvensional. Selain itu kita juga tidak dapat memasukkan gambar maupun foto yang kita kehendaki pada kartu ucapan ini. Untuk mengatasi hal ini maka diperlukan suatu aplikasi yang memungkinkan *user* untuk mendesain kartunya sendiri. Dengan demikian *user* memiliki keleluasaan untuk mengekspresikan dirinya melalui kartu ucapan yang dikirimkannya.

Pada makalah ini akan dibahas pembuatan aplikasi untuk mendesain kartu ucapan. Adapun fasilitas pada aplikasi yang dikembangkan ini adalah fasilitas untuk memasukkan teks atau gambar bebas, fasilitas pengolahan citra digital yaitu pengaturan *brightness*, *contrast* serta *color balance*. Untuk memperbaiki kualitas gambar yang kurang baik terdapat fasilitas *histogram equalization*, *sharpening* serta *smoothing*. Selain itu juga terdapat fasilitas untuk memberikan efek pada citra digital yaitu negatif, *grayscale* serta *sepia*.

2. PENGOLAHAN CITRA DIGITAL

Pengolahan citra adalah suatu metode yang digunakan untuk memproses atau memanipulasi gambar dalam bentuk 2 dimensi. Pengolahan citra juga dikatakan sebagai operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar. Pada umumnya, tujuan dari pengolahan citra adalah mentransformasikan atau menganalisis suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas.

Terdapat empat klasifikasi dasar dalam pengolahan citra yaitu *point*, *area*, geometri, dan *frame* [1]. Pada operasi *point*, pemrosesan nilai piksel suatu citra dilakukan berdasarkan nilai dan posisi dari piksel tersebut. Termasuk di dalam operasi *point* ini adalah pengaturan *brightness*, kontras, *color balance*, negatif, *gray scaling* serta *sepia*. Pada operasi *area*, pemrosesan nilai piksel suatu citra dilakukan berdasarkan nilai piksel tersebut beserta nilai piksel sekelilingnya. Termasuk di dalam operasi *area* ini adalah *sharpening* dan *smoothing*. Operasi geometri digunakan untuk mengubah posisi dari sebuah piksel menjadi posisi lain yang dikehendaki. Termasuk di dalam operasi geometri ini adalah translasi, *scaling*, rotasi dan *flip*.

2.1. Brightness

Di dalam ruangan, seringkali kita perlu mengatur intensitas lampu agar ruangan menjadi lebih terang atau lebih gelap. Dalam dunia pengolahan citra, hal itu disebut pengaturan *brightness* dimana dapat dilakukan dengan cara meningkatkan atau

menurunkan nilai piksel dari seluruh bagian dalam citra tersebut. [1]. Untuk melakukan hal ini digunakan formula seperti terdapat di bawah ini.

$$K_o = K_i + C \quad (1)$$

Dimana K_o adalah nilai piksel output, K_i adalah nilai piksel input dan C adalah konstanta yang bernilai positif atau negatif.

Untuk citra berwarna yang mempunyai format warna RGB (*red, green, blue*), modifikasi *brightness* dilakukan terhadap setiap elemen warna yang masing-masing nilainya dapat berbeda dengan yang lain. Misalnya citra akan dibuat lebih merah, maka nilai merah pada setiap titik ditambah dengan nilai tertentu. Sebaliknya, jika warna biru citra akan dikurangi, maka nilai elemen biru setiap titik dikurangi.

2.2. Kontras

Jika nilai-nilai warna piksel dalam sebuah citra terdiri atas nilai-nilai dengan rentang yang tidak terlalu berbeda secara keseluruhan, maka citra tersebut akan kelihatan kurang kontras. Hal ini disebabkan citra tersebut memiliki kurva histogram yang sempit dengan tepi kiri dan tepi kanan yang berdekatan, sehingga titik tergelap dalam citra tersebut tidak mencapai hitam pekat dan titik paling terang dalam citra itu tidak berwarna putih cemerlang [2]. Peningkatan kontras dapat dilakukan dengan bermacam-macam fungsi. Salah satu fungsi yang dapat digunakan adalah seperti terdapat di bawah ini.

$$\text{If } K_i > 127 \text{ then } K_o = K_i + P \text{ else } K_o = K_i - P \quad (2)$$

Dimana K_o adalah nilai piksel output, K_i adalah nilai piksel input dan P adalah konstanta pengaturan kontras.

2.3. Color Balance

Dalam citra berwarna, biasanya setiap titik memiliki nilai RGB (*red, green, blue*). Dengan menambah intensitas nilai R maka citra akan terlihat lebih merah, penambahan intensitas nilai G akan membuat citra terlihat lebih hijau, dan penambahan intensitas nilai B akan menjadikan citra terlihat lebih biru.

2.4. Negatif

Operasi negatif dilakukan dengan mengubah nilai piksel menjadi nilai negatifnya. Hasil operasi ini akan menghasilkan gambar seperti negatif film sebelum

dicetak menjadi foto. Fungsi yang digunakan untuk melakukan operasi ini terdapat di bawah ini.

$$K_o = K_{max} - K_i \quad (3)$$

Dimana K_o adalah nilai piksel output, K_i adalah nilai piksel input dan K_{max} adalah nilai maksimum piksel tersebut (biasanya adalah 255).

2.5. Grayscale

Grayscale adalah proses pengubahan nilai piksel dari berwarna RGB menjadi *gray-level*. Pada dasarnya proses ini dilakukan dengan meratakan nilai piksel dari 3 nilai RGB menjadi 1 nilai. Nilai piksel dapat dihitung dengan rumus:

$$K_o = (R_i + G_i + B_i) / 3 \quad (4)$$

Dimana K_o adalah nilai warna output, sedangkan R_i , G_i , dan B_i adalah nilai RGB input.

2.6. Sephia

Operasi ini menghasilkan gambar yang berwarna merah kecoklatan seperti foto-foto yang dicetak pada jaman dulu sebelum foto-foto dicetak dengan warna *true color* seperti yang sekarang ini. Untuk menghasilkan warna merah kecoklatan pada gambar, sebelumnya pada gambar dilakukan operasi *grayscale*, kemudian setelah itu dilakukan perubahan nilai pada R dan G sesuai dengan nilai *depth* (kedalaman) yang dikehendaki. Fungsi untuk menghasilkan efek ini terlihat di bawah ini.

$$\begin{aligned} K_o &= (R_i + G_i + B_i) / 3 \\ R_o &= K_o + (depth * 2) \\ G_o &= K_o + depth \\ B_o &= K_o \end{aligned} \quad (5)$$

Di mana R_i , G_i , dan B_i adalah nilai RGB input, R_o , G_o , dan B_o adalah nilai RGB output dan *depth* adalah nilai kedalaman yang diinginkan.

2.7. Smoothing dan sharpening

Proses smoothing dan sharpening dapat dilakukan pada domain spasial ataupun frekuensi. Pada dasarnya kedua proses tersebut dapat dilakukan dengan cara yang sama, hanya menggunakan parameter yang berbeda. Untuk proses pada domain spasial dapat dilakukan dengan melakukan konvolusi antara gambar dengan matriks yang biasanya berukuran 3x3 atau 5x5. Adapun perbedaan antara proses *smoothing* dan

proses *sharpening* adalah pada isi matriks yang digunakan. Berikut ini ditampilkan matriks untuk kedua operasi [3].

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1/9</td><td>1/9</td><td>1/9</td></tr> <tr><td>1/9</td><td>1/9</td><td>1/9</td></tr> <tr><td>1/9</td><td>1/9</td><td>1/9</td></tr> </table> <p>Average</p>	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1/10</td><td>1/10</td><td>1/10</td></tr> <tr><td>1/10</td><td>2/10</td><td>1/10</td></tr> <tr><td>1/10</td><td>1/10</td><td>1/10</td></tr> </table> <p>Lowpass1</p>	1/10	1/10	1/10	1/10	2/10	1/10	1/10	1/10	1/10	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1/12</td><td>1/12</td><td>1/12</td></tr> <tr><td>1/12</td><td>4/12</td><td>1/12</td></tr> <tr><td>1/12</td><td>1/12</td><td>1/12</td></tr> </table> <p>Lowpass2</p>	1/12	1/12	1/12	1/12	4/12	1/12	1/12	1/12	1/12	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1/20</td><td>1/20</td><td>1/20</td></tr> <tr><td>1/20</td><td>12/20</td><td>1/20</td></tr> <tr><td>1/20</td><td>1/20</td><td>1/20</td></tr> </table> <p>Lowpass3</p>	1/20	1/20	1/20	1/20	12/20	1/20	1/20	1/20	1/20
1/9	1/9	1/9																																					
1/9	1/9	1/9																																					
1/9	1/9	1/9																																					
1/10	1/10	1/10																																					
1/10	2/10	1/10																																					
1/10	1/10	1/10																																					
1/12	1/12	1/12																																					
1/12	4/12	1/12																																					
1/12	1/12	1/12																																					
1/20	1/20	1/20																																					
1/20	12/20	1/20																																					
1/20	1/20	1/20																																					
<table border="1" style="width: 100%; border-collapse: collapse; margin: auto;"> <tr><td>1/16</td><td>2/16</td><td>1/16</td></tr> <tr><td>2/16</td><td>4/16</td><td>2/16</td></tr> <tr><td>1/16</td><td>2/16</td><td>1/16</td></tr> </table> <p>Gaussian</p>				1/16	2/16	1/16	2/16	4/16	2/16	1/16	2/16	1/16																											
1/16	2/16	1/16																																					
2/16	4/16	2/16																																					
1/16	2/16	1/16																																					

Gambar 1. Matriks untuk Proses Smoothing

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>9</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table> <p>Mean removal</p>	-1	-1	-1	-1	9	-1	-1	-1	-1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>0</td><td>-1</td><td>0</td></tr> <tr><td>-1</td><td>5</td><td>-1</td></tr> <tr><td>0</td><td>-1</td><td>0</td></tr> </table> <p>Highpass1</p>	0	-1	0	-1	5	-1	0	-1	0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>-2</td><td>1</td></tr> <tr><td>-2</td><td>5</td><td>-2</td></tr> <tr><td>1</td><td>-2</td><td>1</td></tr> </table> <p>Highpass2</p>	1	-2	1	-2	5	-2	1	-2	1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>0</td><td>-1/16</td><td>0</td></tr> <tr><td>-1/16</td><td>20/16</td><td>-1/16</td></tr> <tr><td>0</td><td>-1/16</td><td>0</td></tr> </table> <p>Highpass3</p>	0	-1/16	0	-1/16	20/16	-1/16	0	-1/16	0
-1	-1	-1																																					
-1	9	-1																																					
-1	-1	-1																																					
0	-1	0																																					
-1	5	-1																																					
0	-1	0																																					
1	-2	1																																					
-2	5	-2																																					
1	-2	1																																					
0	-1/16	0																																					
-1/16	20/16	-1/16																																					
0	-1/16	0																																					

Gambar 2. Matriks untuk Proses Sharpening

2.8 Operasi Geometri

Operasi geometri ini terdiri dari translasi, *scaling*, rotasi serta *flip*. Operasi translasi adalah operasi untuk mengubah posisi gambar melakukan penambahan atau pengurangan baik pada koordinat x dan atau koordinat y suatu citra. Sedangkan operasi *scaling* adalah operasi untuk memperbesar atau memperkecil ukuran gambar. Proses ini dilakukan dengan mengalikan dengan faktor lebih dari satu untuk perbesaran dan mengalikan dengan faktor kurang dari satu untuk pengecilan.

Operasi rotasi digunakan untuk memutar gambar dengan mengalikan posisi x dan y dengan sebuah matriks. Berikut ini adalah formula yang digunakan untuk operasi rotasi.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (7)$$

x dan y adalah koordinat asal gambar, sedangkan x' dan y' adalah koordinat hasil rotasi. Formula di sebelah kiri (6) digunakan untuk memutar sejauh sudut θ berlawanan arah dengan jarum jam, sedangkan formula di sebelah kanan (7) digunakan untuk memutar searah dengan jarum jam.

Operasi flip digunakan untuk mencerminkan gambar dimana pencerminan ini dapat dilakukan horisontal atau vertikal. Untuk mencerminkan gambar secara horisontal terhadap garis vertikal di tengah gambar, digunakan rumus:

$$x' = w - 1 - x \quad (8)$$

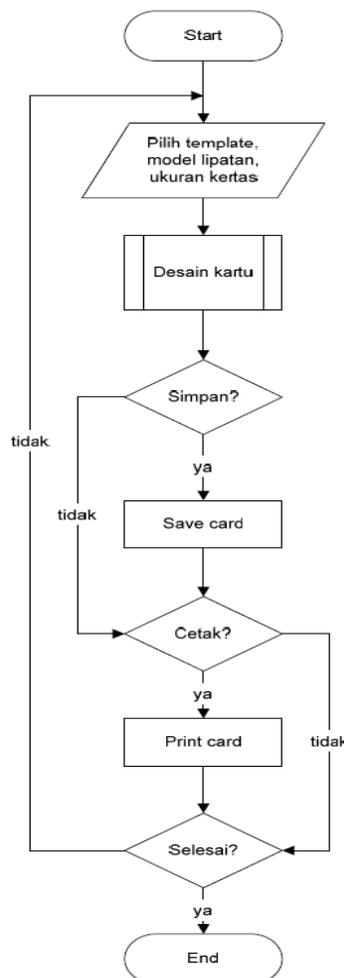
Dimana w adalah lebar gambar, x adalah posisi awal koordinat x dan x' adalah posisi hasil koordinat x . Sedangkan untuk mencerminkan gambar secara vertikal terhadap garis horisontal di tengah gambar, digunakan rumus:

$$y' = h - 1 - y \quad (9)$$

Dimana h adalah tinggi gambar, y adalah posisi awal koordinat y dan y' adalah posisi hasil koordinat y .

3. DESAIN APLIKASI

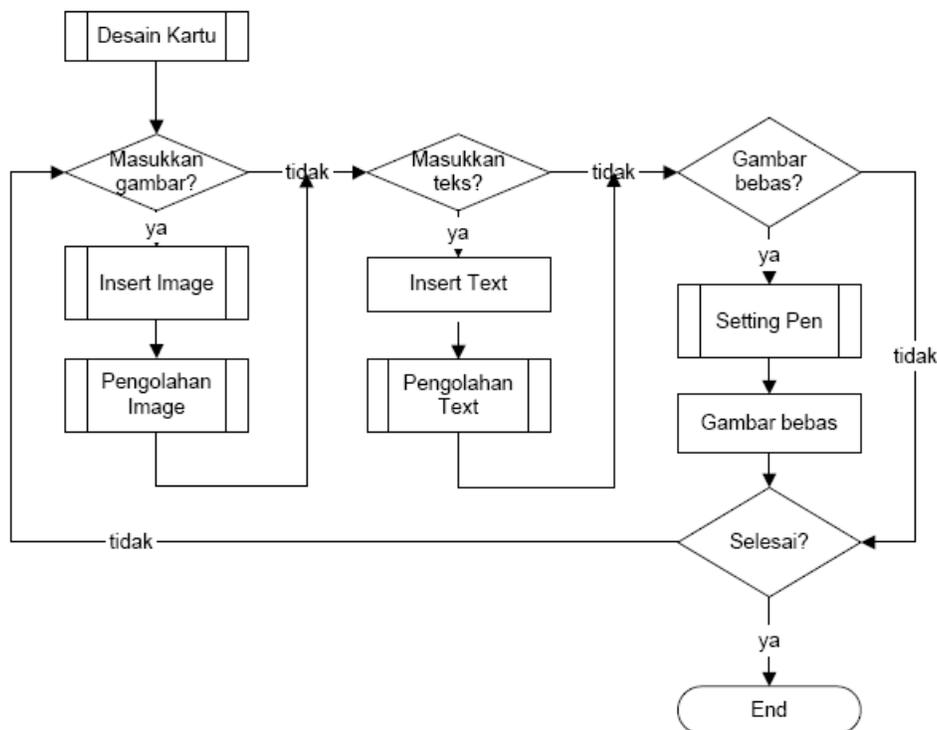
Berikut ini ditampilkan diagram alir dari aplikasi yang dikembangkan.



Gambar 3. Diagram Alir Aplikasi

Dari diagram alir tersebut terlihat bahwa input pertama kali aplikasi adalah *template* yang hendak digunakan, model lipatan serta ukuran kertas. Setelah menerima input dari *user*, aplikasi akan mempersiapkan kanvas untuk didesain sesuai keinginan *user*. Prosedur utama yang terdapat pada aplikasi ini adalah desain kartu. Pada prosedur desain kartu ini, dilakukan desain semua fitur yang dapat dimanfaatkan oleh *user* untuk mendesain kartu sesuai dengan keinginannya. Setelah selesai mendesain kartu, *user* dapat menyimpan kartu hasil desainnya untuk dibuka lagi di lain waktu.

Prosedur desain kartu ditampilkan pada diagram alir di gambar 4.



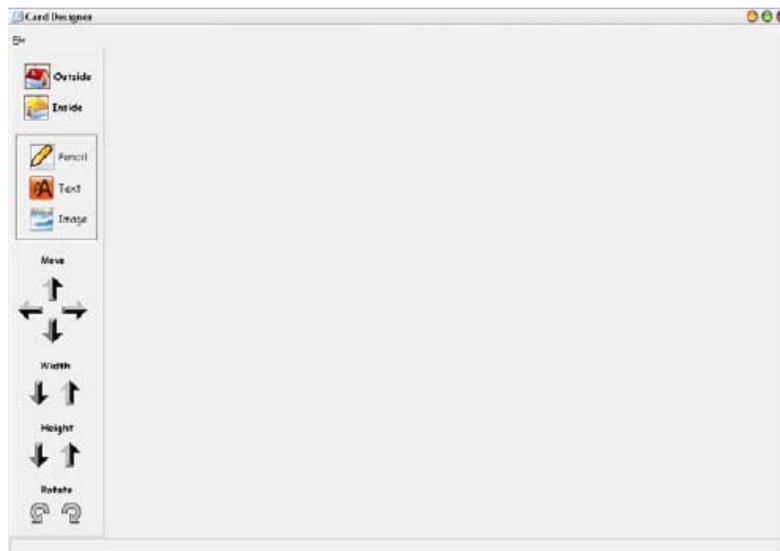
Gambar 4. Diagram Alir Desain Kartu

Pada bagian desain kartu ini terdapat fasilitas untuk memasukkan gambar, teks ataupun menggambar bebas pada kartu. Jika *user* memasukkan gambar, lebih lanjut terdapat fasilitas untuk pengolahan gambar tersebut yaitu pengaturan *brightness*, kontras, *color balance*, *histogram equalization*, *smoothing*, *sharpening*, konversi ke *grayscale*, negatif ataupun *sephia*. Sedangkan untuk pemasukan teks, terdapat fasilitas untuk memilih jenis, warna serta ukuran *font*. Gambar serta teks yang dimasukkan juga dapat diubah posisi, ukuran serta orientasinya. Di samping fasilitas untuk memasukkan gambar ataupun teks, aplikasi juga dikembangkan untuk mempunyai fasilitas menggambar

bebas dimana user dapat menggunakan *mouse* ataupun *stylus pen* untuk melakukan penggambaran bebas ini.

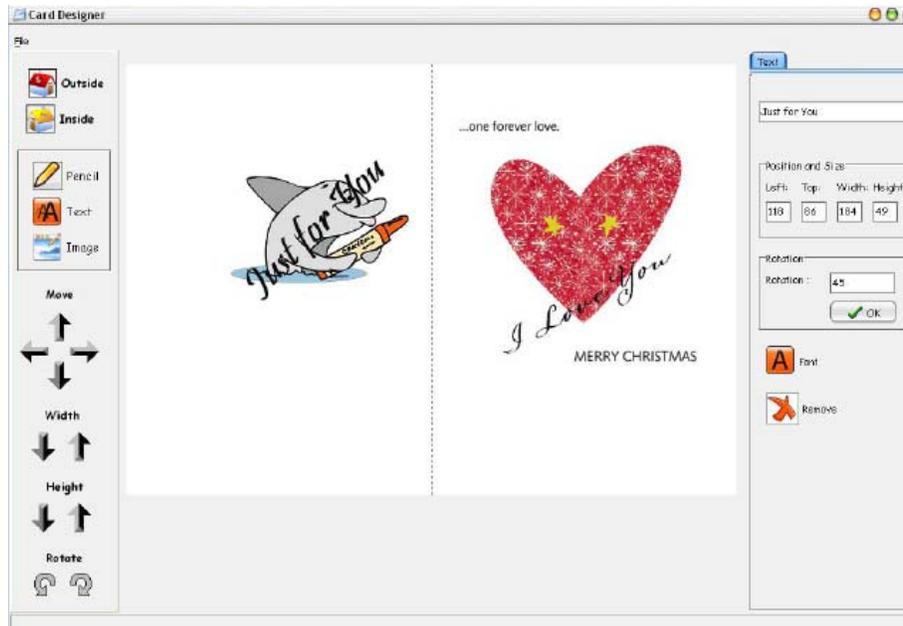
4. IMPLEMENTASI DAN HASIL

Dari desain yang telah dilakukan, pengembangan aplikasi dilanjutkan dengan melakukan pemrograman dengan menggunakan bantuan bahasa pemrograman Delphi. Pada pemrograman ini diimplementasikan semua yang berhubungan dengan fasilitas yang telah didesain. Hasil dari implementasi ini dapat dilihat pada gambar di bawah ini.



Gambar 5. Tampilan Aplikasi

Pada gambar di atas terlihat aplikasi saat pertama kali dijalankan. Fasilitas-fasilitas yang dapat dimanfaatkan oleh *user* diletakkan berupa tombol-tombol yang terletak pada *toolbar* di bagian kanan. Berikut ini ditampilkan hasil desain kartu yang telah dilakukan oleh *user*.



Gambar 6. Hasil Desain Kartu

Pada bagian kanan aplikasi terdapat fitur-fitur yang berhubungan dengan obyek yang sedang aktif digunakan oleh *user*. Sebagai contoh untuk teks, terdapat fitur yang berupa posisi, orientasi serta pemilihan font.

5. KESIMPULAN

Dari hasil pengembangan aplikasi serta penggunaannya, dapat ditarik kesimpulan sebagai berikut:

- *Brightness* dengan nilai yang terlalu besar atau kecil tidak efektif karena mengakibatkan gambar asli tidak kelihatan. Pengaturan *brightness* dimana *trackbar* digeser pada nilai maksimal akan menghasilkan gambar dengan warna putih seluruhnya, sedangkan ketika *trackbar* berada pada nilai minimal akan menghasilkan gambar dengan warna hitam seluruhnya.
- Pengaturan *color balance* pada sebuah gambar dimana setiap unsur warna berada pada suatu nilai yang sama akan menghasilkan gambar yang sama dengan pengaturan *brightness* pada nilai tersebut.
- Efek *sephia* dengan nilai *depth* 0 akan menghasilkan gambar yang berwarna keabuan.
- Proses rotasi dan *resize* pada gambar akan menurunkan kualitas gambar. Gambar yang dirotasi n derajat dan kemudian dikembalikan pada posisi semula dengan melakukan rotasi $-n$ derajat menghasilkan gambar dengan ketajaman yang

berkurang dari aslinya. Demikian juga dengan proses *resize*, setelah gambar dikembalikan pada ukuran aslinya pun ketajamannya akan berkurang. Untuk mengatasi hal ini, gambar asli harus disimpan di variabel lain sehingga setiap kali dilakukan proses rotasi maupun *resize*, proses ini dilakukan terhadap gambar yang merupakan gambar asli.

DAFTAR PUSTAKA

- [1] Achmad, Balza & Kartika Firdausy, *Teknik Pengolahan Citra Digital Menggunakan Delphi*, Yogyakarta: Ardi Publishing, 2005.
- [2] Susilo, Djoko, *Grafika Komputer dengan Delphi*, Yogyakarta: Graha Ilmu, 2005.
- [3] Gonzalez, R.C. & Woods, R.E, *Digital Image Processing*, New Jersey: Prentice Hall, 2002.
- [4] Baxes, Gregory A, *Digital Image Processing: principles and application*, New York: John Wiley & Sons Inc, 1994.