

# PERANCANGAN DAN PEMBUATAN SOFTWARE PHOTO-TO-PATTERN PADA CROSS STITCH

**Yulia, Kartika Gunadi**

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra  
e-mail : [yulia@petra.ac.id](mailto:yulia@petra.ac.id) , [kgunadi@petra.ac.id](mailto:kgunadi@petra.ac.id)

**Jenny Jaury**

Alumni Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

**ABSTRAK:** *Cross Stitch* adalah salah satu bentuk penyulaman pada kain dengan menyulam benang membentuk tanda silang (X). Untuk memulai suatu *cross stitch*, diperlukan sebuah *pattern* yang digunakan sebagai patokan dalam menyulam. Meskipun telah tersedia aneka macam *pattern cross stitch*, tetapi kadang-kadang konsumen menginginkan *pattern* lainnya yang tidak tersedia, misalnya konsumen ingin menyulam wajah seseorang atau binatang kesayangan.

Tujuan dari penelitian ini adalah untuk menghasilkan sebuah *software* yang dapat membentuk *pattern cross stitch* dengan *input-an* berupa sebuah *file* citra. Algoritma *Bicubic Interpolation scaling* digunakan dalam proses *scaling* untuk mengubah ukuran citra sesuai ukuran *pattern* yang diinginkan. *Input-an file* citra dibatasi pada format citra .bmp, .pcx, .gif, dan .jpg. *Output software* berupa *pattern* berwarna, dan *pattern* dengan simbol beserta keterangannya. Bahasa pemrograman yang digunakan adalah Borland Delphi 6.0.

Hasil pengujian *software* menunjukkan bahwa *pattern cross stitch* dengan rasio perbandingan ukuran *pattern* dengan ukuran gambar aslinya lebih atau sama dengan 100 %, akan mirip dengan gambar aslinya meskipun warna benang yang digunakan sedikit. Akan tetapi, jika rasio perbandingannya kurang dari 100 %, maka kemiripan antara *pattern cross stitch* dengan gambar aslinya bergantung pada variasi warna benang yang dimiliki *pattern*. Semakin banyak jumlah warna benang yang digunakan sebuah *pattern*, maka semakin mirip *pattern* dengan gambar aslinya.

**Kata kunci:** *Pattern Cross Stitch, File Citra, Scaling.*

**ABSTRACT:** *Cross Stitch* is one kind of embroidery that makes stitching forming a cross sign (X). To begin a cross stitch, we need a pattern. Although cross stitch stores have already provided so many cross stitch patterns, sometimes consumer wants another pattern which is not provided, for example a pattern that describes someone's face or lovely pets.

The aim of this research is to create a software that can make a cross stitch pattern with an image file input. The Bicubic Interpolation scaling algorithm is used in scaling process to change the size of the original image file to the size of the wanted pattern. Image file input is restricted to some image file formats, those are .bmp, .pcx, .gif, and .jpg. The outputs of the software are color pattern, and symbol pattern along with its symbol explanation. Borland Delphi 6.0 is used for the programming language.

The software testing results show that a cross stitch pattern with the comparison ratio of the pattern size and the original picture size is more or equal to 100 %, will look like the original picture even though the variation color is low. But if the comparison size ratio is less than 100 %, then the similarity between the cross stitch pattern and the original picture depends on the color variation of the pattern. If there are many colors used on the pattern, the pattern will look more like the original picture.

**Keywords:** *Cross Stitch Pattern, Image File, Scaling..*

## 1. PENDAHULUAN

*Cross Stitch* adalah salah satu bentuk penyulaman pada kain dengan menyulam benang membentuk tanda *cross* (X). Gabungan dari hasil sulaman X tersebut

akan membentuk sebuah gambar. Untuk memulai suatu *cross stitch*, diperlukan sebuah *pattern* yang digunakan sebagai patokan dalam menyulam. *Pattern* ini dapat diperoleh dari katalog-katalog yang

disediakan di toko-toko sulaman. Akan tetapi, kadang-kadang ada konsumen yang menginginkan *pattern* tertentu yang tidak tersedia di katalog, seperti gambar wajah seseorang atau binatang kesayangan, atau gambar-gambar lain yang unik atau jarang ditemukan.

Beranjak dari masalah ini, maka akan dibuat sebuah *software* yang dapat membentuk *pattern cross stitch* dari sebuah gambar.

Pada penelitian ini format citra yang dapat digunakan adalah:

- BMP (\*.BMP) dengan jumlah *bit* per *pixel*-nya 8 atau 24 *bit* dan tanpa terkompresi.
- PCX (\*.PCX) dengan versi *file PC Paintbrush 3.0 (version 5)*.
- CompuServe GIF (\*.GIF).
- JPEG (\*.JPG)

Sedangkan panjang dan lebar citra tidak lebih dari 65535 *pixel*, kecuali untuk citra .gif dibatasi hanya sampai 200x200 *pixel*. Untuk warna benang yang digunakan meliputi 38 warna benang DMC dan 77 warna benang wol.

*Output* dari *software* ini meliputi *Pattern* berwarna, *Pattern* dengan simbol-simbol untuk memudahkan penjahitan, keterangan dari simbol-simbol yang digunakan, meliputi: warna, kode (khusus untuk benang DMC), dan nama benang. Perangkat lunak yang digunakan yaitu *Borland Delphi 6*.

### Algoritma Bicubic Interpolation

Pendekatan yang digunakan dalam algoritma *bicubic interpolation* untuk *image scaling* adalah memperkirakan warna dari setiap *pixel* pada *final image* dengan mengambil nilai rata-rata dari 16 *pixel* terdekat di sekitar *pixel* yang bersesuaian pada *original image*.

Contoh gambar menjelaskan prinsip dari *scaling*, untuk menentukan warna dari setiap titik (*i',j'*) pada *final image*.

Terdapat hubungan linear *scaling* di antara dua *image*, yaitu titik (*i',j'*) menunjuk pada posisi *non integer* pada *original image*. Posisi ini didapat dengan:

$$x = i' \cdot w / w' \quad (1)$$

$$y = j' \cdot h / h' \quad (2)$$

Keterangan :

*w* = *width* dari *original image*

*h* = *height* dari *original image*

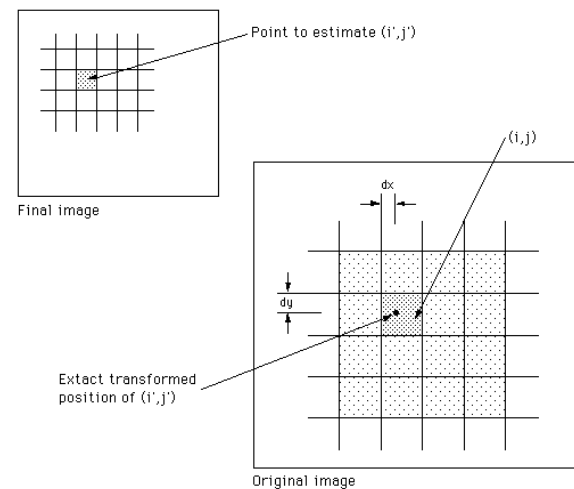
*w'* = *width* dari *final image*

*h'* = *height* dari *final image*

Koordinat *pixel* terdekat (*i,j*) adalah nilai *integer* dari *x* dan *y*. Nilai *dx* dan *dy* dalam gambar diperoleh melalui:

$$dx = x - i \quad (3)$$

$$dy = y - j \quad (4)$$



Sumber: <http://astronomy.swin.edu.au/~pbourke/colour/bicubic/>

**Gambar 1. Ilustrasi *scaling***

Persamaan berikut memberikan nilai interpolasi yang diaplikasikan pada setiap komponen *red*, *green* dan *blue* dari suatu *pixel*.

$$F(i', j') = \sum_{m=-1}^2 \sum_{n=-1}^2 F(i + m, j + n) R(m - dx) R(n - dy) \quad (5)$$

Keterangan:

$F(i', j')$  = koordinat *pixel* pada *final image*

$F(i, j)$  = koordinat *pixel* pada *original image* dengan fungsi  $R(x)$

$$R(x) = \frac{1}{6} [P(x+2)^3 - 4P(x+1)^3 + 6P(x)^3 - 4P(x-1)^3] \quad (6)$$

$$P(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (7)$$

Sebagai contoh pada sebuah *image* berukuran 200x100 *pixel*. Pada *image* ini

akan dilakukan *scaling bicubic interpolation* menjadi *image* berukuran 300x150 *pixel*.

$$w = 200 \quad h = 100 \quad w' = 300 \quad h' = 150$$

Kemudian untuk setiap *pixel* pada *final image*, akan dilakukan fungsi interpolasi sesuai persamaan di atas. Misalnya *pixel* (151,83), berarti  $i' = 151$  dan  $j' = 83$ .

Selanjutnya, dengan persamaan (1) dan (2), dapat diperoleh:

$$x = 100,67$$

$$y = 55,33$$

Kemudian dengan persamaan (3) dan (4) diperoleh nilai  $dx$  dan  $dy$ .

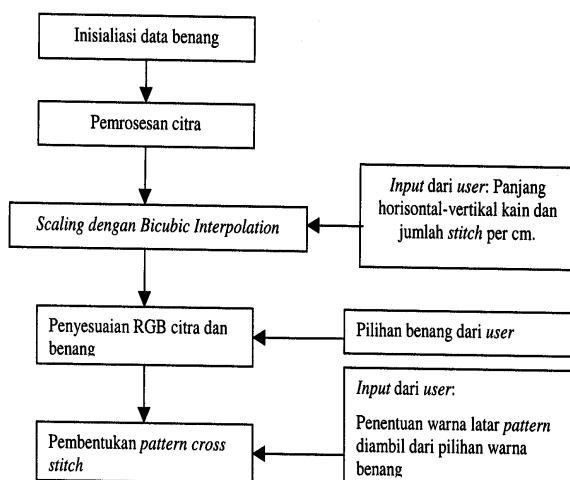
$$dx = x - i = 100,67 - 100 = 0,67$$

$$dy = y - j = 55,33 - 55 = 0,33$$

Dengan menggunakan persamaan yang ditunjukkan pada persamaan (5), maka akan diperoleh nilai interpolasi dari *pixel* (151,83) pada *final image* dengan melibatkan 16 *pixel* di sekitar *pixel* (100,55) pada *original image*. Hal ini dilakukan untuk masing-masing nilai *red*, *green*, dan *blue* dari *pixel* (151,83).

## 2. ANALISIS DAN DESAIN

Perencanaan *software* terbagi dalam lima tahap, yaitu: inisialisasi data benang, pemrosesan citra, *scaling*, penyesuaian RGB citra dan benang, dan pembentukan *pattern cross stitch*.



**Gambar 2. Bagan Garis Besar Perencanaan Software**

### 2.1 Tahap Inisialisasi Data Benang

Inisialisasi data benang meliputi nilai *red*, *green*, dan *blue* dari tiap benang. Juga kode (khusus untuk DMC saja) dan nama benang. Data benang meliputi 38 benang DMC dan 77 benang wol dan disimpan dalam variabel *b\_wol* (untuk benang wol) dan *b\_dmc* (untuk benang DMC).

### 2.2 Tahap Pemrosesan Citra

Data citra yang dibaca diletakkan dalam variabel *iminput* dalam urutan mulai dari *top-left* terus ke kanan dan ke bawah sampai *right-bottom*. Variabel *iminput* ini memiliki tipe variabel *timagedata*. Lebar dan tinggi dari citra disimpan dalam variabel *width\_in* dan *height\_in*. Sedangkan untuk palet warna disimpan dalam variabel *paletteClr*.

```

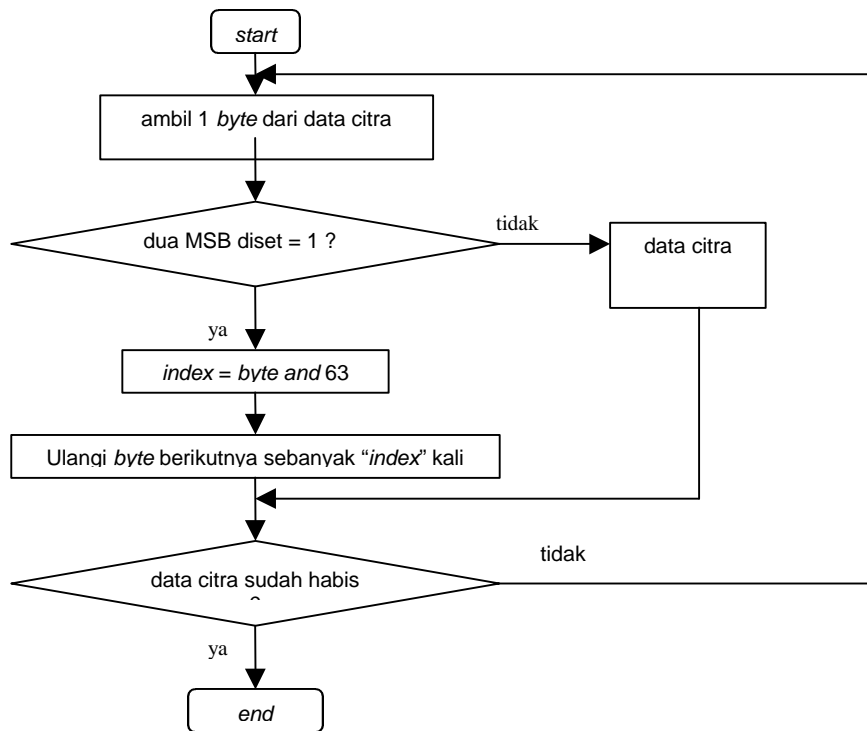
Type
    timagedata = record
        r,g,b : pbytearray;
    end;
    tPaletteClr = Record
        Red, Green, Blue : Byte;
    end;
Var
    paletteClr : array [0..255] of
    tPaletteClr;
    iminput : timagedata;
    width_in,height_in : longword;
  
```

#### 2.2.1 Pemrosesan Citra BMP

Untuk BMP 8 *bit*, data citra merupakan rangkaian indeks dari *color map entry* BMP. Sedangkan untuk 24 *bit*, data citra disusun dalam pola *blue-green-red* untuk setiap *pixel*-nya. Citra BMP menyimpan data citranya dalam urutan terbalik yaitu mulai dari *bottom-left* terus ke kanan sebanyak *width* dari citra, lalu ke atas sebanyak *height* dari citra. Jadi data citra pertama yang dibaca adalah data citra pada posisi *bottom-left*.

#### 2.2. Pemrosesan Citra PCX

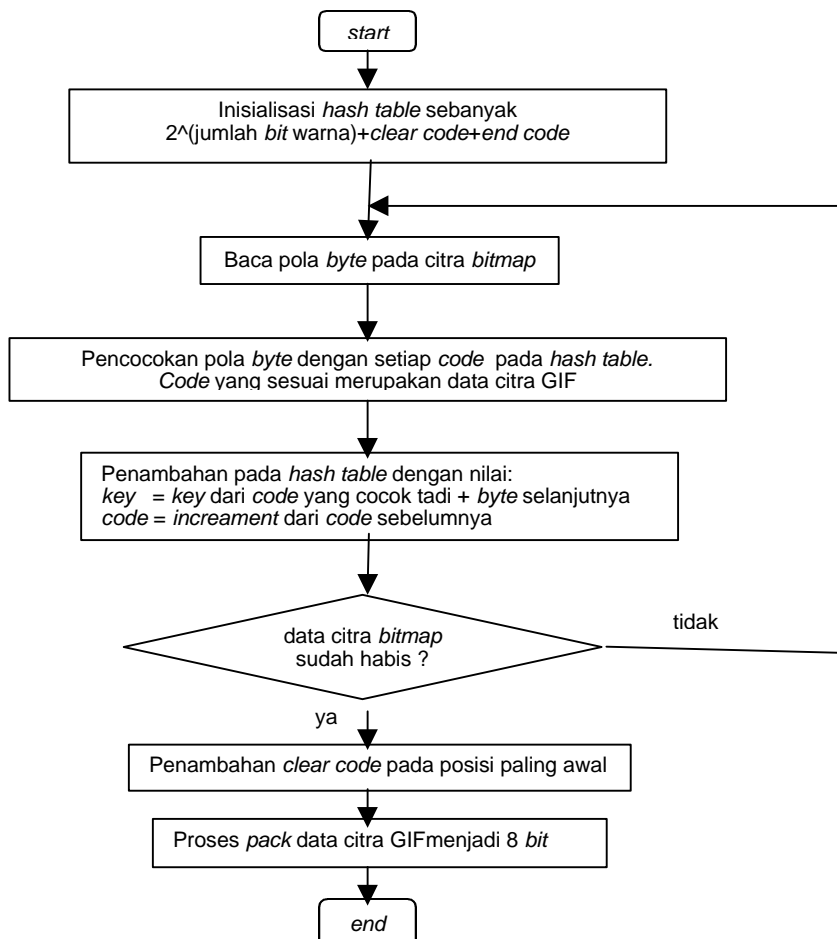
Data citra PCX menggunakan algoritma kompresi *run-length encoding* (RLE).



**Gambar 3. Flowchart Algoritma RLE Citra PCX**

### 2.2.3 Pemrosesan Citra GIF

Data citra GIF menggunakan pengembangan algoritma kompresi LZW.



**Gambar 4. Flowchart Proses Kompresi Data Citra Bitmap Menjadi Data Citra GIF**

### 2.2.4 Pemrosesan Citra JPG

Untuk pemrosesan citra JPG, digunakan bantuan komponen *TJpegImage* untuk membaca citra yang menggunakan kompresi JPEG. Karena citra .jpg menggunakan kompresi JPEG, maka citra.jpg dapat langsung dimanipulasi dengan menggunakan komponen *TJpegImage* tersebut. Kemudian objek citra *jpeg* tersebut diubah menjadi objek citra *bitmap* dengan bantuan komponen *TBitmap*.

### 2.3 Tahap *Scaling*

Pada tahap ini akan dilakukan proses *scaling* terhadap data citra yang sudah disimpan dalam variabel *iminput.r*, *iminput.g*, dan *iminput.b* tadi. Proses *scaling* menggunakan metode *bicubic interpolation*, yaitu dengan memperkirakan warna dari setiap *pixel* pada *final image* dengan mengambil nilai rata-rata dari 16 *pixel* terdekat di sekitar *pixel* yang bersesuaian pada *original image*. Karena membutuhkan nilai dari 16 *pixel* di sekitar *pixel* yang bersangkutan, maka terlebih dahulu dilakukan proses *padding*.

Proses *padding* adalah proses penambahan dua *pixel* pada masing-masing sisi kiri, kanan, atas dan bawah dari data citra. Hal ini dilakukan pada masing-masing nilai *red*, *green*, dan *blue*. Nilai dari dua *pixel* tambahan tersebut sama dengan nilai dari *pixel* paling ujung dari tiap sisi.

Setelah dilakukan proses *padding*, kemudian dihitung nilai interpolasi dari tiap komponen *red*, *green*, dan *blue* dari setiap *pixel* pada *final image* dengan menggunakan algoritma *bicubic interpolation*. Ukuran dari *final image* diperoleh dari masukan *user* mengenai panjang horisontal kain dan jumlah *stitch* per cm. Panjang vertikal kain akan dihitung secara otomatis disesuaikan dengan ukuran citra.

### 2.4 Tahap Penyesuaian RGB Citra Dan Benang

Pada tahap ini dilakukan pencarian terhadap warna dari pilihan benang dari *user*. Warna yang dicari adalah warna yang nilai RGB-nya paling kecil selisihnya dengan RGB setiap *pixel* dari *final image*.

*User* dapat memilih benang yang diinginkan apakah itu wol atau DMC, dan juga *user* dapat membatasi jumlah warna dengan memilih warna benang yang diinginkan saja. Hasil pilihan benang dari *user* akan disimpan dalam variabel *selected* sesuai indeks benangnya pada saat inisialisasi.

Jika terdapat nilai total selisih yang sama, maka yang terpilih adalah yang perbedaan nilai antara selisih nilai RGB nya yang paling kecil. Misalkan terdapat dua data sebagai berikut:

Data #1	Selisih Nilai Red	40
	Selisih Nilai Green	30
	Selisih Nilai Blue	95
	Total	165

Data #2	Selisih Nilai Red	5
	Selisih Nilai Green	20
	Selisih Nilai Blue	140
	Total	165

Kedua data di atas mempunyai total selisih nilai yang sama yaitu 165. Tetapi yang terpilih adalah data pertama karena perbedaan nilai antara selisih nilai RGB nya lebih kecil daripada data kedua.

Data #1	Selisih Nilai Red dan Green	10
	Selisih Nilai Green dan Blue	65
	Selisih Nilai Blue dan Red	55
	Total	130

Data #2	Selisih Nilai Red dan Green	15
	Selisih Nilai Green dan Blue	120
	Selisih Nilai Blue dan Red	135
	Total	270

### 2.5 Tahap Pembentukan *Pattern Cross Stitch*

Pembentukan *pattern cross stitch* terdiri atas tiga bagian, yaitu *pattern* berwarna, *pattern* dengan simbol, dan keterangan dari simbol-simbol yang digunakan. Untuk *pattern* berwarna dan *pattern* dengan simbol menggunakan komponen *TDrawGrid*. Pengisian tiap *cell* dilakukan pada event *OnDrawCell*. Khusus untuk *pattern* dengan simbol, *pattern* akan dibagi menjadi beberapa bagian jika panjang horisontal dari

*pattern* lebih dari 70 *stitch*, atau panjang vertikal dari *pattern* lebih dari 42 *stitch*.

Simbol yang melambangkan warna *background* tidak akan digambarkan (user dapat menentukan salah satu warna dari warna-warna yang dipilih sebagai warna *background*). Hal ini dimaksudkan supaya *user* tidak bingung dalam melihat *pattern* simbol.

### 3. HASIL DAN PENGUJIAN

Berikut ini contoh pembentukan *pattern cross stitch* dengan data citra:

*Width* = 284 *pixel*.

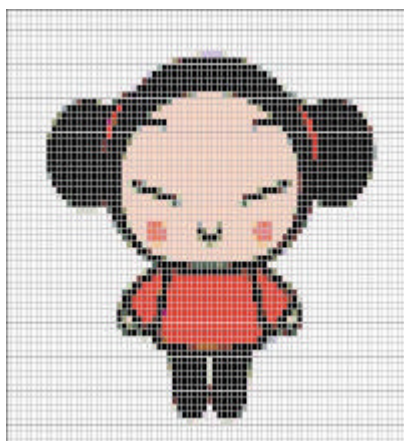
*Height* = 229 *pixel*.

Data Pattern	
Panjang Horisontal	40
Panjang Vertikal	32
Jumlah Stitch per cm	2

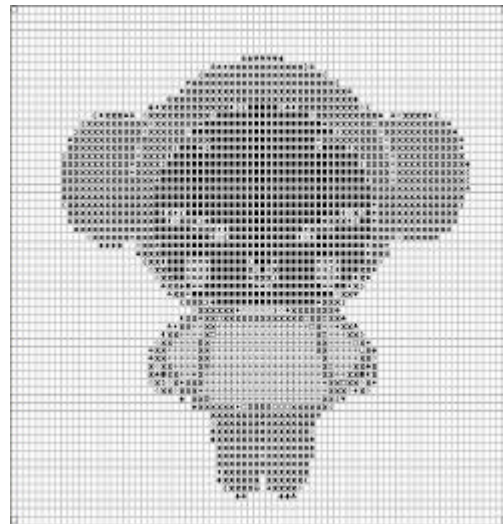
Data Benang	
Tipe Benang	Wol
Warna Benang	77
Warna Latar	Putih



Gambar 5. Citra Sebenarnya (Pucca)



Gambar 6. *Pattern Cross Stitch* Berwarna (Pucca) - Wol



Gambar 7. *Pattern Cross Stitch* dengan Simbol (Pucca) – Wol

Simbol	Warna	Kode	Nama
Φ			putih
◆			pink pucat
◆			coklat.mocca
◆			ungu.muda
◆			hijau.amy - muda
◆			coklat.tua
◆			hitam
◆			hijau.amy
◆			abu.abu
◆			ungu.pucat
◆			blewah
◆			hijau.gelap
◆			merah.keoranyean - terang
◆			coklat.kemerahan - gelap
◆			merah.tua
◆			coklat.kemerahan
◆			hijau.krem
◆			merah.bata - muda
◆			orange - terang
◆			kuning.keoranyean - muda
◆			pink.oranye

Gambar 8. Keterangan Simbol-simbol yang Digunakan (Pucca) - Wol

Contoh *pattern cross stitch* di atas (pucca), menggunakan tipe benang wol. Berikut adalah *pattern cross stitch* yang sama, tetapi dengan menggunakan tipe benang DMC.

Data benang:

Tipe benang = DMC

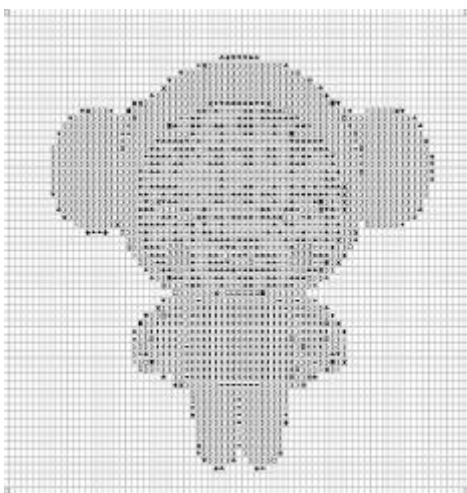
Pilihan warna benang = *all* (38 warna)

Penentuan warna latar = *White*





**Gambar 9. Pattern Cross Stitch Berwarna (Pucca) – DMC**



**Gambar 10. Pattern Cross Stitch dengan Simbol (Pucca) – DMC**

	Simbol	Warna	Kode	Nama
<b>SIZE</b>	Φ		0	White
Horizontal : 80	♣		747	Sky Blue - vy lt
Vertical : 64	♥		772	Yellow Green - vy lt
	♦		554	Violet - lt
40 x 32 cm	♠		776	Pink - med
2 stitch per cm	•		552	Violet - med
	✕		820	Royal Blue - vy dk
	⌘		310	Black
	⌚		911	Emerald Green - med
Jenis benang : dmc	⊙		340	Blue Violet - med
Total warna : 22	■		319	Pistachio Green - vy dk
Background : Φ	◇		936	Avocado Green - dk
	◆		3078	Golden Yellow - vy lt
	□		550	Violet - vy dk
	▣		666	Christmas Red
	▤		915	Phm - med
	◀		606	Bright Orange-Red
	▶		731	Olive Green - dk
	▲		955	Nile Green - lt
	▼		815	Garnet - med
	⊖		3341	Apricot
	⊕		3825	Pale Pumpkin

**Gambar 11. Keterangan Simbol-simbol yang Digunakan (Pucca) – DMC**

**Tabel 1. Hasil Pengujian *Scaling* dan Variasi Warna (sambungan)**

Warna Ukuran	Hitam dan putih	38 warna DMC	77 warna wol
10x10 (10 %)			
20x20 (20 %)			
30x30 (30 %)			
40x40 (40 %)			
50x50 (50 %)			
60x60 (60 %)			
70x70 (70 %)			
80x80 (80 %)			
90x90 (90 %)			

Pengujian yang dilakukan adalah sebanyak 9 kali, yaitu dengan rasio 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%. Dari 9 pengujian tersebut, diperoleh hasil sebagai berikut:

- Untuk pembentukan *pattern cross stitch* dengan dua warna benang (hitam dan putih), *pattern* yang berhasil membentuk kotak adalah sebanyak satu (11.11 %).
- Untuk pembentukan *pattern cross stitch* dengan 38 warna benang DMC, *pattern*

yang berhasil membentuk kotak adalah sebanyak 7 (77,78 %).

Untuk pembentukan *pattern cross stitch* dengan 77 warna benang wol, *pattern* yang berhasil membentuk kotak adalah sebanyak 7 (77,78 %).

#### 4. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

- Algoritma *bicubic interpolation scaling* terbukti dapat digunakan untuk melakukan *scaling* dengan baik dalam membentuk *pattern cross stitch*.
- Semakin banyak variasi warna yang dimiliki sebuah *pattern* maka semakin mirip *pattern* dengan gambar aslinya.

##### 4.2 Saran

Beberapa saran pengembangan terhadap *software photo-to-pattern* ini adalah:

- Format citra yang dapat diproses oleh *software* ini hanya terbatas pada format .bmp (8 atau 24 *bit* dan tanpa terkompresi), .pcx (*version* 5), .gif, dan .jpg. Jika cakupan format citra lebih luas, maka akan jadi lebih baik karena tidak dibutuhkan lagi perangkat lunak lain (seperti *ACDSee* atau *Adobe Photoshop*) untuk mengubah format citra sebagai *input* (dalam kasus: citra yang dimiliki *user* tidak termasuk dalam keempat format tersebut).
- Sebaiknya ukuran citra tidak dibatasi agar pemakaian *software* bisa lebih fleksibel dan dapat menerima input langsung dari scanner.
- Sebaiknya warna benang yang disediakan untuk tipe benang DMC lebih diperbanyak supaya variasi warna *pattern* nantinya juga akan lebih banyak.

#### DAFTAR PUSTAKA

1. Kay, D.C., & Levine, J.R, *Graphics File Formats 2<sup>nd</sup> Edition* , USA: Windcrest, 1995
2. Bourke, P, *Bicubic Interpolation for Image Scaling*, <http://astronomy.swin.edu.au/~pbourke/colour/bicubic/> (2003, April 11), 2001, May
3. Cheok, Y.C, *Introduction On Text Compression Using Lempel, Ziv, Welch (LZW) Method*, <http://www.geocities.com/yccheok/lzw/lzw.html> (2003, Juni 18).
4. Harper, J., *GIF Graphics Interchange Format Version 89a*, <http://www.dcs.ed.ac.uk/home/mxr/gfx/2d/GIF89a.txt> (2003, January 10), 1990, July
5. Walch, V., *Color Wheel in DMC Floss*, <http://www.jccniowa.org/~pgega/workshops/> (2003, January 10), 1999