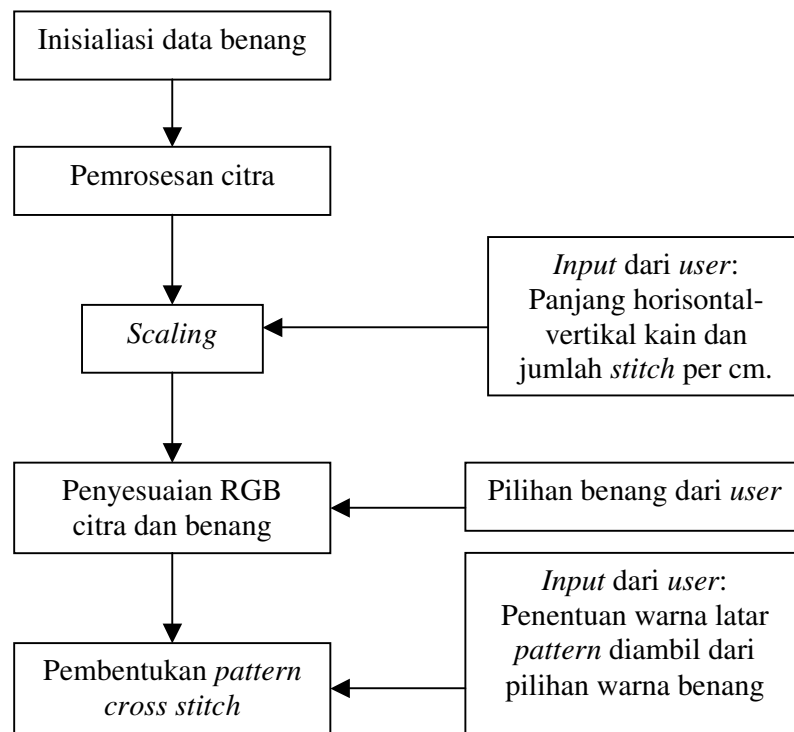


3. ANALISIS DAN DESAIN

Pada bab ini akan dibahas tentang perencanaan *software*. Secara umum, perencanaan *software* terbagi dalam lima tahap, yaitu: inisialisasi data benang, pemrosesan citra, *scaling*, penyesuaian RGB citra dan benang, dan pembentukan *pattern cross stitch*.



Gambar 3.1. Bagan Garis Besar Perencanaan *Software*

3.1. 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).

```

type
  benang = record
    er,ge,be: byte;
    kode:word;
    nama:string[30];
  end;

var
  b_dmc : array[1..38] of benang;
  b_wol : array[1..77] of benang;

```

Tabel 3.1. Tabel Data Benang DMC

<i>Red</i>	<i>Green</i>	<i>Blue</i>	Kode	Nama	Warna
255	255	255	0	<i>White</i>	
174	98	76	301	<i>Mahogany - med</i>	
254	235	52	307	<i>Lemon</i>	
0	0	0	310	<i>Black</i>	
73	89	65	319	<i>Pistachio Green - vy dk</i>	
124	114	175	333	<i>Blue Violet - vy dk</i>	
165	169	216	340	<i>Blue Violet - med</i>	
115	68	100	550	<i>Violet - vy dk</i>	
148	103	146	552	<i>Violet - med</i>	
206	165	201	554	<i>Violet - lt</i>	
236	67	78	606	<i>Bright Orange-Red</i>	
226	64	83	666	<i>Christmas Red</i>	
193	95	152	718	<i>Plum</i>	
133	115	51	731	<i>Olive Green - dk</i>	
252	173	58	741	<i>Tangerine - med</i>	
255	226	110	743	<i>Yellow - med</i>	
203	233	241	747	<i>Sky Blue - vy lt</i>	
227	238	197	772	<i>Yellow Green - vy lt</i>	
253	195	200	776	<i>Pink - med</i>	
70	69	109	791	<i>Cornflower Blue - vy dk</i>	
86	104	166	797	<i>Royal Blue</i>	
170	196	227	800	<i>Delft - pale</i>	
104	142	178	806	<i>Peacock Blue - dk</i>	
154	60	75	815	<i>Garnet - med</i>	
66	62	84	820	<i>Royal Blue - vy dk</i>	
173	198	68	907	<i>Parrot Green - lt</i>	
111	155	122	911	<i>Emerald Green - med</i>	
148	54	84	915	<i>Plum - med</i>	
195	92	82	920	<i>Copper - med</i>	
90	92	59	936	<i>Avocado Green - dk</i>	
254	90	77	947	<i>Burnt Orange</i>	
198	235	204	955	<i>Nile Green - lt</i>	
253	252	201	3078	<i>Golden Yellow - vy lt</i>	
253	149	136	3341	<i>Apricot</i>	
244	173	216	3609	<i>Plum - ul lt</i>	

Tabel 3.1. Tabel Data Benang DMC (sambungan)

<i>Red</i>	<i>Green</i>	<i>Blue</i>	Kode	Nama	Warna
69	97	124	3808	<i>Turquoise - ul vy dk</i>	
253	195	145	3825	<i>Pale Pumpkin</i>	
196	153	84	3829	<i>Old Gold - vy dk</i>	

Tabel 3.2. Tabel Data Benang Wol

<i>Red</i>	<i>Green</i>	<i>Blue</i>	Nama	Warna
248	51	64	merah	
203	35	120	merah tua - sedang	
184	27	109	merah tua	
163	22	100	merah tua - gelap	
141	46	95	merah hati	
152	30	111	merah tua keunguan	
160	40	162	merah keunguan - tua	
208	100	203	merah keunguan	
209	30	181	merah maron	
213	197	235	ungu muda	
198	159	231	ungu sedang	
174	97	224	ungu	
141	80	201	ungu tua	
64	18	143	biru sangat tua	
114	125	206	biru tua	
121	159	228	biru	
162	200	234	biru muda	
90	203	220	biru laut - muda	
81	193	223	biru laut	
88	107	90	hijau gelap	
47	122	85	hijau tua	
76	148	90	hijau	
124	187	115	hijau sedang	
139	212	122	hijau muda	
139	130	27	hijau tembaga	
128	102	37	hijau tembaga tua	
128	133	91	hijau army	
166	164	119	hijau army - muda	
152	181	80	hijau kekuningan - tua	
214	222	100	hijau kekuningan	
225	230	163	hijau kekuningan - muda	
214	218	177	hijau krem	
250	252	203	kuning muda	
247	232	160	kuning	
250	231	91	kuning tua	
246	195	18	kuning emas	
255	182	50	kuning keoranyean - tua	
255	208	153	kuning keoranyean - muda	

Tabel 3.2. Tabel Data Benang Wol (sambungan)

<i>Red</i>	<i>Green</i>	<i>Blue</i>	Nama	Warna
255	219	202	oranye muda	
236	211	148	kuning krem	
234	206	126	kuning tembaga muda	
220	184	80	kuning tembaga tua	
239	183	108	coklat keoranyean - muda	
224	136	33	coklat keoranyean	
255	160	88	Oranye	
252	116	83	Oranye - terang	
244	58	44	Merah keoranyean - terang	
247	91	93	merah keoranyean	
218	95	90	merah bata keoranyean	
212	68	92	merah bata - tua	
220	75	101	merah bata	
238	130	134	merah bata - muda	
244	131	175	pink maron	
240	128	215	pink keunguan	
240	149	206	pink keunguan - muda	
255	128	210	pink terang	
255	182	208	pink	
255	214	234	pink muda	
252	122	149	pink keoranyean	
184	146	99	coklat muda	
145	110	49	coklat	
111	57	50	coklat tua	
140	55	41	coklat kemerahan - gelap	
177	94	36	coklat kemerahan	
187	102	56	coklat kemerahan - sedang	
184	114	58	coklat kemerahan - muda	
177	129	80	coklat tanah	
0	0	0	hitam	
166	178	178	abu-abu	
255	255	255	putih	
231	226	210	coklat mocca	
250	246	211	krem muda	
252	255	230	kuning pucat	
212	240	230	hijau pucat	
205	231	250	biru pucat	
235	210	237	ungu pucat	
255	238	237	pink pucat	

Implementasi dari inisialisasi 77 data benang wol dan 38 data benang DMC dapat dilihat pada subbab 4.1.

3.2. Tahap Pemrosesan Citra

Seperti yang telah dijelaskan pada ruang lingkup pada bab 1, citra terbatas dalam format tertentu. Data citra yang dibaca kemudian 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 (jika ada) 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;
```

3.2.1. Pemrosesan Citra BMP

Seperti yang telah dijelaskan dalam ruang lingkup pada bab 1, citra BMP dibatasi pada yang tanpa kompresi dan jumlah *bit* per *pixel*-nya 8 atau 24 *bit*. Untuk BMP 8 *bit*, data citra merupakan rangkaian indeks dari *color map entry* BMP. Sedangkan untuk yang 24 *bit*, data citra disusun dalam pola *blue-green-red* untuk setiap *pixel*-nya.

Pembacaan data citra sangat sederhana, karena dibaca apa adanya. Hanya saja 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*.

Implementasi dari pemrosesan citra BMP dapat dilihat pada subbab 4.2.

Untuk citra BMP digunakan variabel *headerBMP* untuk menampung *header* citra.

```
type
  tHeaderBMP = Record
    bfType : Array[1..2]Of Char;
    bfSize : longword;
    bfReserved1,bfReserved2 : Word;
    bfOffBits,biSize,biWidth,biHeight : Longword;
    biPlanes,biBitCount : Word;
```

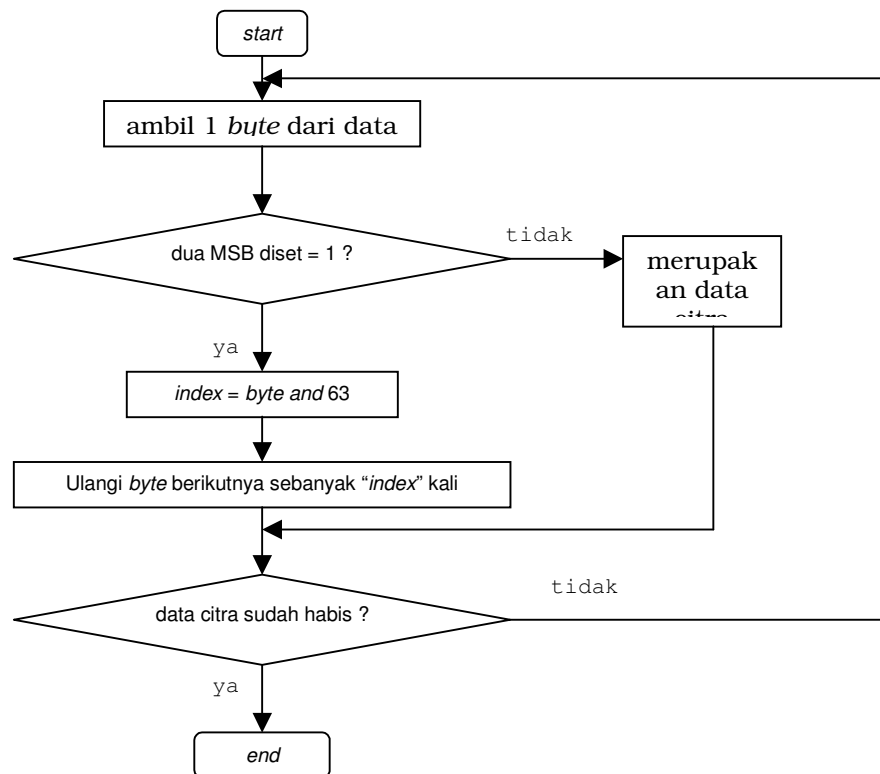
```

    biCompression, biSizeImage, biXPelsPerMeter : Longword;
    biYPelsPerMeter, biClrUsed, biClrImportant : Longword;
End;
var
    headerBMP : theaderBMP;

```

3.2.2. Pemrosesan Citra PCX

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



Gambar 3.2. *Flowchart* Algoritma RLE Citra PCX

Implementasi dari pemrosesan citra PCX dapat dilihat pada subbab 4.3.

Untuk citra PCX digunakan variabel *headerPCX* untuk menampung *header* citra.

```

type
    THeaderPCX= record
        flag, version, encoding, bps: byte;
        Dim: array[0..7] of byte;
        HRes, VRes: array[0..1] of byte;
        HeadPal: array[0..47] of byte;
        Res, plane: byte;
        bpl: array[0..1] of byte;
        HeadPalInt: array[0..1] of byte;

```

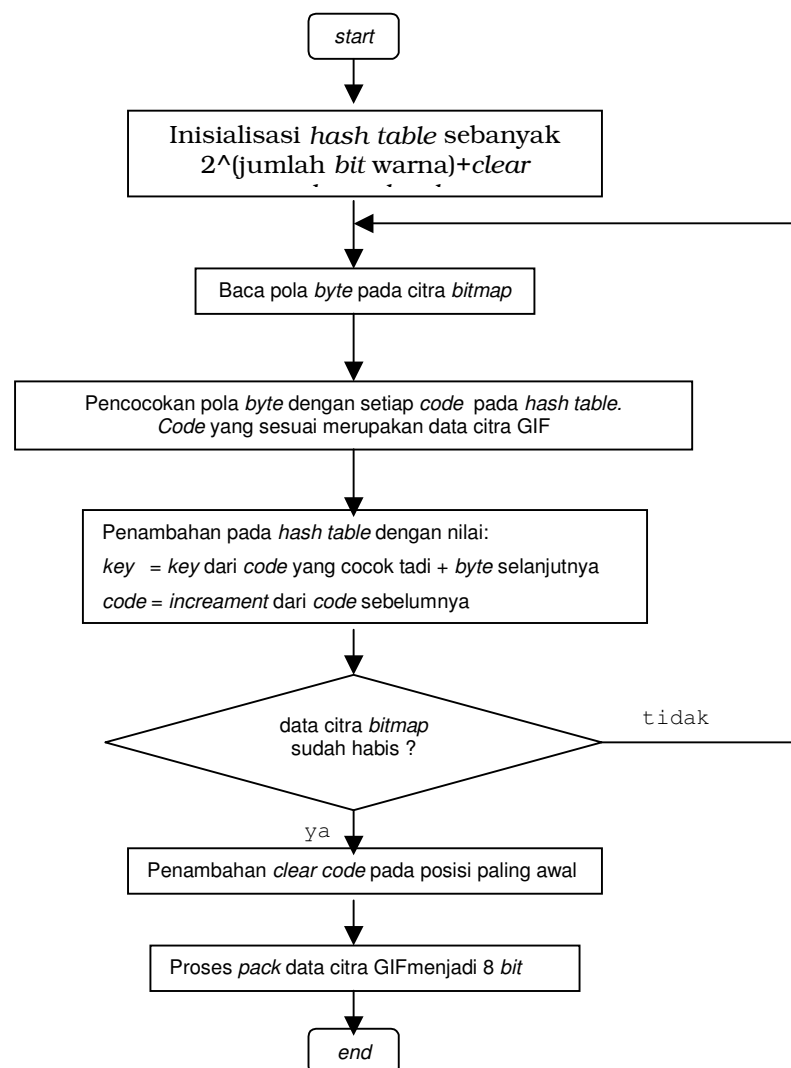
```

    SizeX, SizeY: array[0..1] of byte;
    blanks: array[0..53] of byte;
end;
var
    headerPCX : theaderpcx;

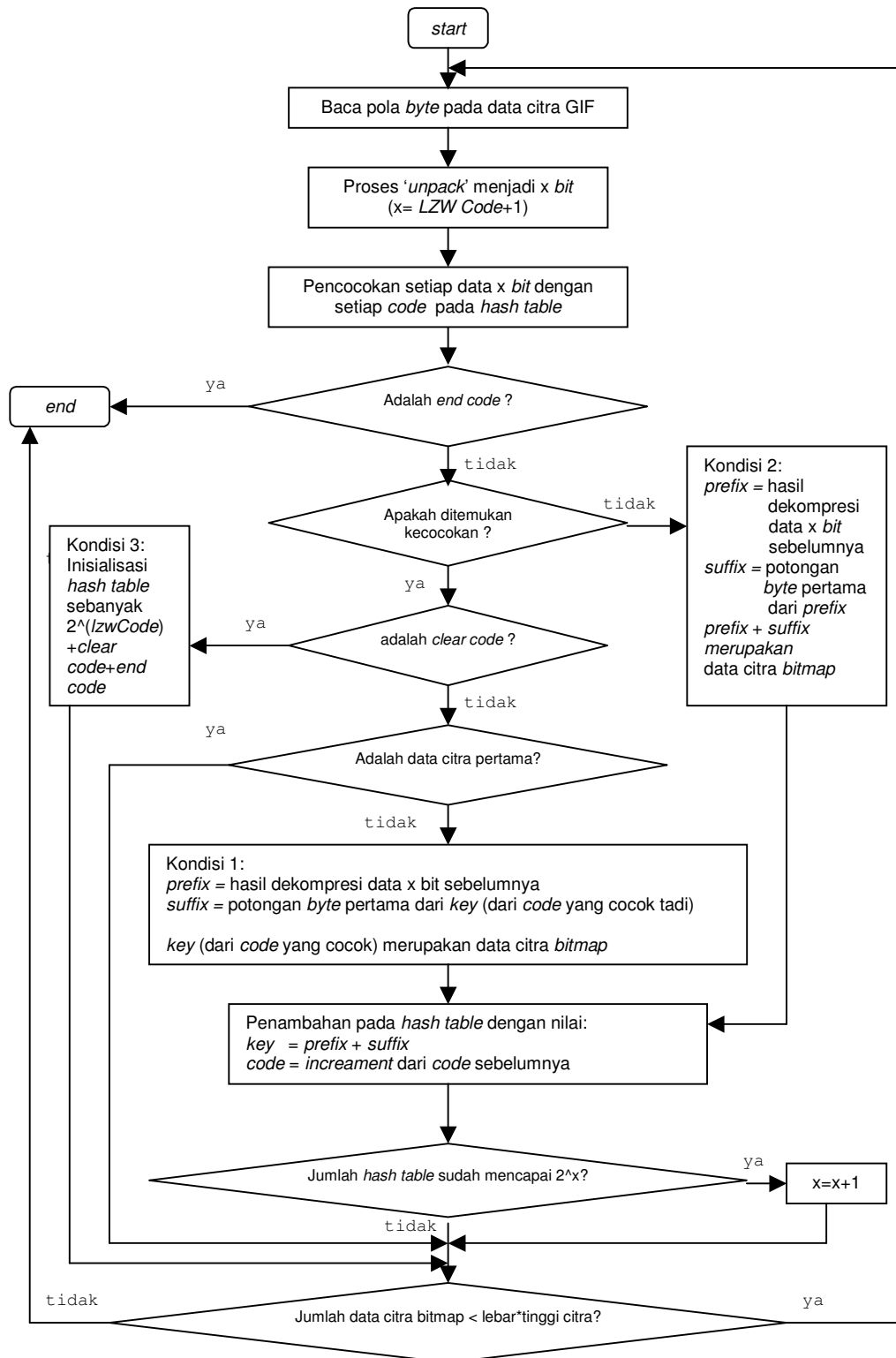
```

3.2.3. Pemrosesan Citra GIF

Data citra GIF menggunakan pengembangan algoritma kompresi LZW. Sebelum kita membahas mengenai proses dekompresi dari citra GIF, ada baiknya terlebih dahulu kita mengetahui proses kompresi karena proses dekompresi merupakan proses kebalikan dari dekompresi.



Gambar 3.3. Flowchart Proses Kompresi Data Citra Bitmap Menjadi Data Citra GIF



Gambar 3.4. Flowchart Proses Dekompresi Data Citra GIF
Menjadi Data Citra Bitmap

Implementasi dari pemrosesan citra GIF dapat dilihat pada subbab 4.5.

Untuk citra GIF digunakan variabel *headerGIF*.

```
type
  theaderGIF = record
    signature, version : array [1..3] of char;
    screenw, screenh, leftpos, toppos, width, height : word;
    flag1, bgcolor, ratio, separator, flag2, LZWCode, blocksize, termin,
    trailer, jumclrlocal, rescolor, jumbit : byte;
    globalclrtab, localclrtab : array [0..255] of tpaletteClr;
  end;

var
  headerGIF : theaderGIF;
```

3.2.4. Pemrosesan Citra JPG

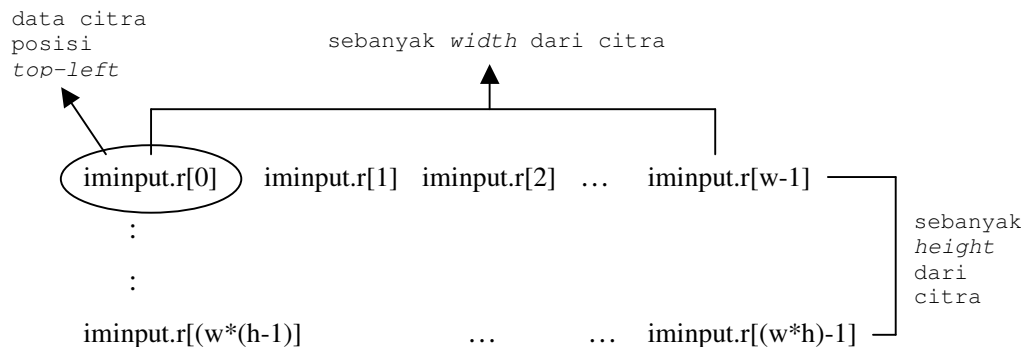
Untuk pemrosesan citra JPG, digunakan bantuan komponen *TJpegImage* untuk membaca citra yang menggunakan kompresi JPEG. Karena citra .jpg menggunakan kompresi JPEG, maka citra .jpg dapat langsung dimanipulasi dengan menggunakan komponen *TJpegImage* tersebut. Kemudian objek citra *jpeg* tersebut diubah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

Implementasi dari pemrosesan citra PCX dapat dilihat pada subbab 4.6.

```
var
  jpeg:Tjpegimage;
  bmp:Tbitmap;
```

3.2.5. Pemrosesan Data Citra

Nilai *red* dari data citra dimasukkan dalam variabel *iminput.r[i]* dengan urutan sebagai berikut:



Keterangan:

- ♣ w = lebar citra
- ♣ h = tinggi citra

Seperti halnya dengan nilai *red*, nilai *green* dan *blue* dari data citra yang telah dibaca, juga dimasukkan dalam variabel masing-masing yaitu *iminput.g[i]* dan *iminput.b[i]* dengan urutan sama seperti di atas.

3.3. 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.

Hasil *padding* akan disimpan dalam variabel *impadded* yang memiliki tipe data seperti variabel *iminput*.

```
var
    impadded : timagedata;
```

Setelah dilakukan proses *padding*, kemudian dihitunglah nilai interpolasi dari tiap komponen *red*, *green*, dan *blue* dari setiap *pixel* pada *final image* dengan menggunakan *algoritma bicubic interpolation* seperti yang sudah dijelaskan pada subbab 2.1. 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.

Hasil interpolasi akan disimpan dalam variabel *imoutput* yang juga memiliki tipe data yang sama seperti variabel *iminput* dan *impadded*.

```
var
    imoutput : timagedata;
```

Implementasi dari tahap *scaling* dapat dilihat pada subbab 4.8.

3.4. 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.

```
var
  selected : array[1..77] of byte;
```

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:

Perbedaan antar selisih nilai *red* dan *green* = 10
 Perbedaan antar selisih nilai *green* dan *blue* = 65
 Perbedaan antar selisih nilai *blue* dan *red* = 55
 Total = 130

Data 2:

Perbedaan antar selisih nilai *red* dan *green* = 15
 Perbedaan antar selisih nilai *green* dan *blue* = 120
 Perbedaan antar selisih nilai *blue* dan *red* = 135
 Total = 270

Setelah setiap *pixel* dicari kesesuaian warnanya dengan pilihan warna *user*, maka dapatlah diketahui warna-warna apa saja yang akan digunakan pada *pattern cross stitch* tersebut. Warna-warna yang terpilih tersebut disimpan dalam variabel *colorused*. Karena ada kemungkinan setiap warna yang dipilih akan digunakan, maka tipe variabel dari *colorused* harus sama dengan variabel *selected*.

```
var
  selected : array[1..77] of byte;
```

Implementasi dari tahap penyesuaian RGB citra dan benang dapat dilihat pada subbab 4.9.

3.5. 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*.

```
const
  simb_maxX = 70;
  simb_maxY = 42;
```

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.

Implementasi dari tahap pembentukan *pattern cross stitch* dapat dilihat pada subbab 4.11.

3.6. Penyimpanan (Save) *Pattern Cross Stitch*

Pattern cross stitch yang sudah dibentuk dapat disimpan dengan memilih menu *File-Save*. *Pattern cross stitch* akan disimpan dalam bentuk *untyped file* dan diberi ekstension *.csp* (*Cross Stitch Pattern*).

Tabel 3.3. Format Urutan Penyimpanan *File .csp*

Nama	Ukuran (byte)	Keterangan
TipeBenang	1	Tipe benang (0=wol, 1=DMC)
TotPil	1	Total pilihan benang <i>user</i>
BlokWarna	totpil	Indeks-indeks warna benang yang dipilih <i>user</i>
Latar	1	Indeks warna latar dari <i>pattern</i> (0=tanpa latar)
<i>Height</i>	2	Panjang vertikal dari <i>pattern</i>
<i>Width</i>	2	Panjang horisontal dari <i>pattern</i>
<i>Stitchpcm</i>	1	Jumlah <i>stitch</i> per cm dari <i>pattern</i>
<i>DataPattern</i>	3 x <i>height</i> x <i>width</i>	Data dari <i>pattern</i> mulai dari <i>stitch</i> ujung kiri atas dalam urutan <i>red-green-blue</i> r0,g0,b0,r1,g1,b1,...,rn,gn,bn (n= <i>height</i> x <i>width</i> -1)