BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Secara umum pengenalan bentuk atau pola digunakan untuk mengenali objek dalam citra. Untuk menyelesaikan permasalahan pengenalan bentuk bisa dilakukan dengan berbagai metode. Salah satunya dengan menggunakan metode *approximate string matching.*

Metode *Approximate String Matching* merupakan metode pencocokkan string. Terdapat dua objek pada metode ini yaitu teks dan pola, tujuan pendekatan ini adalah untuk mencari string didalam sebuah teks atau pola yang mirip dengan string yang ada pada target.

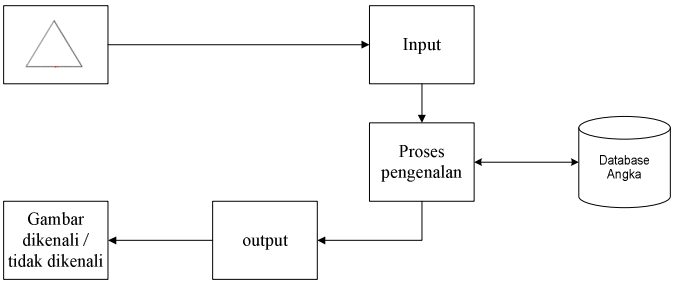
3.2 Analisis Sistem

Tujuan analisis sistem adalah mendapatkan suatu hasil analisis yang sesuai dengan tingkat kebutuhan *user* dan sistem baik dari segi masukan yang dibutuhkan dan keluaran sistem yang diinginkan. Hasil analisis ini nantinya akan menjadi masukan bagi proses perancangan.

Mekanisme global dalam sistem ini adalah:

a. Input citra sumber
b. Proses pengenalan
c. output
d. Hasil keluaran, dikenali atau tidak dikenali.

Bisa dilihat pada gambar berikut



Gambar 3.1 Sistem pengenalan bentuk geometri

3.2.1 Analisis Masukan

Setelah sistem menerima masukan dari pemakai berupa citra sumber, sistem melakukan pengecekan ukuran citra sumber yang dimasukan pemakai, jika ukuran kurang dari 64 *pixel* maka citra sumber ditolak. Citra sumber dalam format bitmap dan jpg yang merupakan citra geometri 2 dimensi yang akan dikenai sistem.

3.2.2 Analisis Pengenalan Pola

Analisis pengenalan pola dilakukan dengan metode A*pproximate String Matching,* proses yang akan dilakukan adalah:

1. Proses Binerisasi

Pada proses ini dilakukan pengubahan citra sumber menjadi citra biner. Untuk menghasilkan citra biner harus melalui dua tahap, yaitu:

a. Pengabuan(*Gray Scaling*)

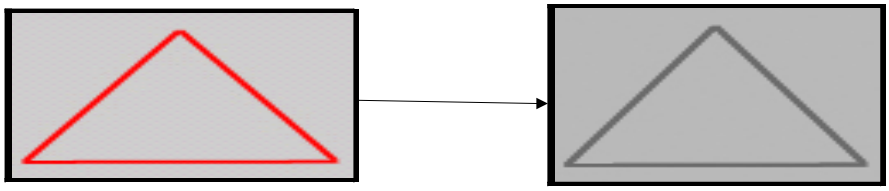
Untuk mengubah citra berwarna ke dalam citra keabuan dilakukan proses *gray scaling*, yaitu dengan cara menentukan nilai (R+G+B) / 3

R= Unsur warna merah pada basis warna RGB

G= Unsur warna hijau pada basis warna RGB

B= Unsur warna biru pada basis warna RGB

Berikut proses konversi citra berwarna menjadi citra *grayscale*



Gambar 3.2 citra berwarna menjadi citra *grayscale*

```
Procedure grayscaling;
kamus
iX,iY : integer
algoritma
   For iX ← 0 to -1 do
   For iY ← 0 to -1 do
 begin
warna= (iX, iY)
R = warna And RGB(255, 0, 0)
G = Int((warna And RGB(0, 255, 0)) /
256)
B = Int(Int((warna And RGB(0, 0, 255))
/ 256) / 256)
 iX, iY = (R + G + B) / 3
    Next iX
    Next iY
end for
```

b. *Thresholding*

Tresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra, untuk proses pengubahan citra keabuan menjadi citra biner dalam sistem yang akan dibangun digunakan metoda *global thresholding* yang bisa mendapatkan nilai *threshold* T yang tepat. *Threshold* dimodifikasi secara berulang-ulang sampai ditemukan nilai yang cocok. Langkah-langkah dalam menentukan *threshold* T adalah sebagai berikut:

1. Pilih nilai T awal, yaitu nilai rata-rata dari intensitas citra.
2. Bagi citra menjadi dua daeah, misalnya R_1 dan R_2, menggunakan nilat T awal yang telah ditentukan.

3. Hitung nilai rata-rata intensitas μ_1 dan μ_2 masing-masing untuk daerah R_1 dan daerah R_2.
4. Hitung nilai *threshold* yang baru dengan rumus $T = \frac{\mu 1 + \mu 2}{2}$
5. Ulangi langkah 2 sampai 4 hingga nilai μ_1 dan μ_2 tidak berubah lagi. Saat itulah nilai T merupakan nilai yang dicari.

T= nilai threshold

μ_1= nilai pixel rata-rata obyek

μ_2= nilai pixel rata-rata background

Berikut proses *grayscalling* menjadi *threshold*



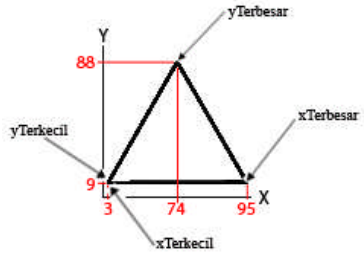
Gambar 3.3 proses *grayscalling* menjadi *threshold*

```
Procedure threshold;
kamus
iX,iY : integer
algoritma
   For iX ← 0 to -1 do
   For iY ← 0 to -1 do
 begin
get pixel (iX,iY)
If  pixel>level
    Next iX
    Next iY
end for
```

c. Proses Pencarian sudut terluar

Pada proses ini dilakukan penulusuran setiap pixel pada citra sumber untuk mencari titik terluar objek, yaitu titik y terbesar, titik y terkecil, titik x terbesar dan titik x terkecil. Sejak citra sumber sudah dalam bentuk citra biner maka kita sudah

bisa membedakan mana objek dan mana background. Objek disini adalah pixel yang berwarna hitam



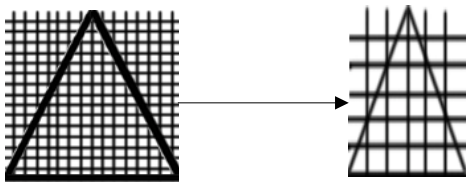
Gambar 3.4 Pencarian sudut terluar

```
procedure  pencarian  sudut
terluar;
kamus
  i,j,n:integer

xMax,Xmin,yMax,yMax:integer
algoritma
  for i→1 to P do
  for i→1 to L do
      N=GetColourpixel (i,j)
      if N>0 then
      if xMin>i then
         xMin→i
      else
      if xMax<i then
         xMax→i
      else
      if yMin>j then
         YMin→j
      else
         yMax→j
  endif
 endif
endif
```

d. Proses Transformasi

Pada proses ini sistem akan menggunakan metode scaling yang dimaksudkan untuk memperkecil (*zoom-out)* citra. Metode scaling digunakan untuk melakukan transformasi agar objek yang dimaksud dapat dimuat dalam wadah berukuran 64x64 pixel sebagai ukuran wadah untuk proses pengenal objek.



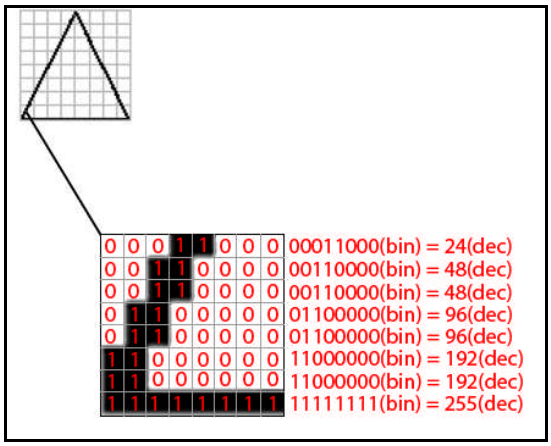
Gambar 3.5 Proses zoom out

Algoritma nya:

```
procedure zoom out;
kamus
 i,j,k,m,n:integer
algoritma
m→0;
n→0;
for i→0 to i>= n-1 do
  for j→0 to j<=m-1 do
     B[m] [n]→ A [i] [j]
     B[m] [n+1]→ A [i] [j]
     B[m+1] [n]→ A [i] [j]
     B[m+1] [n+1]→ A[i] [j]
     n→ n+64
   endfor
  m→ m+64
  n→0
  endfor
```

e. Proses Segmentasi

Hingga tahap ini citra objek yang akan dikenali sudah dalam bentuk citra biner dan berukuran 64x64 pixel, sehingga siap untuk dilakukan proses segementasi menjadi 64 buah segment berukuran 8x8 pixel.



Gambar 3.6 Proses segmentasi

Setiap segment pada citra sumber akan di proses menjadi deret decimal dengan melakukan konversi setiap baris pada segment yang merupakan deret biner menjadi sebuah nilai desimal.

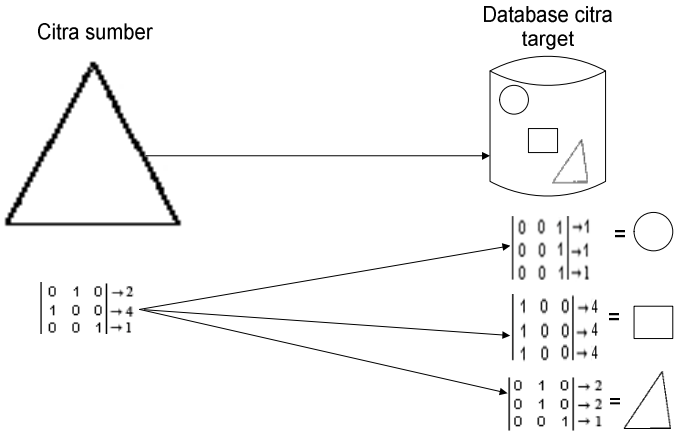
Algoritmanya:

```
procedure segmentasi;
kamus
  i,j,a,b,N:integer
  strBin:string
  arDec:array
algoritma
  for i→0 to 7 do
   for j→0 to 7 do
    for b→0 to 7 do
     strBin→""
     for a→ 0 to 7 do

N=GetColourPixel((I*8)+a,(J*8)+b)
       if N>0 then
         strBin&"1"
        else
         strBin→strBin&"0"
       endif
      endfor
     P→P+1
arDec(P)→strBinToDecimal(strBin)
    endfor
  endfor
 endfor
```

f. Proses Pengenalan Menggunakan Algoritma *levenshtein distance*

Setelah proses segmentasi maka sistem sudah mendapatkan deret desimal dari citra sumber, langkah selanjutnya adalah melakukan proses pencocokkan dengan setiap deret desimal dari citra target yang sudah dipersiapkan didalam *database system* menggunakan algoritma *levenshtein distance*, database deret desimal citra target adalah *knowledge* yang dimiliki oleh sistem. Berikut gambar 3.7 proses pencocokan setiap deret citra sumber dan citra target.

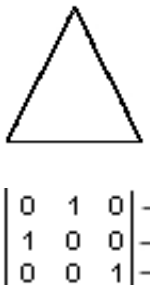


Gambar 3.7 Proses pencocokan deret desimal

Dalam hal ini penulis akan melakukan beberapa analisis bagaimana penerapan algoritma *levenshtein distance* dalam pengenalan gambar.

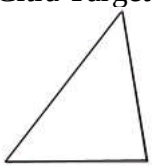
Misalkan terdapat 2 citra yang sudah dalam bentuk biner yang akan dibandingkan:

Citra Sumber A



$$\begin{vmatrix} 0 & 1 & 0 & \rightarrow 2 \\ 1 & 0 & 0 & \rightarrow 4 \\ 0 & 0 & 1 & \rightarrow 1 \end{vmatrix}$$

Citra Target X



$$\begin{vmatrix} 0 & 1 & 0 & \rightarrow 2 \\ 0 & 1 & 0 & \rightarrow 2 \\ 0 & 0 & 1 & \rightarrow 1 \end{vmatrix}$$

Kemudian dibuat representasi deret desimal dimulai dari baris pertama dari citra hingga ke baris terakhir, untuk citra sumber deret desimal yang dapat kita buat adalah 2 4 1 dan deret desimal untuk citra target adalah 2 2 1.

Selanjutnya untuk mengetahui jarak citra sumber dengan citra target di lakukan proses algoritma *levenshtein distance* terhadap deret desimal dari citra target dengan deret desimal dari citra sumber.

string sumber: 241 string target: 221

Proses algoritma *levenshtein distance* adalah sebagai berikut:

1. Jika panjang sumber adalah 0 maka jarak sumber dengan target adalah panjang target yaitu 3.
2. Jika panjang target adalah 0 maka jarak sumber dengan target adalah panjang sumber yaitu 3.
3. Buat matrix dengan ukuran (sumber+1) x (target+1).

s\t		2	2	1
	0	1	2	3
2	1			
4	2			
1	3			

4. Melakukan proses pencocokkan dengan membandingkan setiap karakter target dengan setiap karakter sumber.

Iterasi 1,1

s/t		2	2	1
	0	1	2	3
2	1	0		
4	2			
1	3			

Jika karakter 1 target dengan karakter 1 sumber sama maka masukkan nilai jarak 0 pada cell 1,1.

Iterasi 1,2

s/t		2	2	1
	0	1	2	3
2	1	0		
4	2	1		
1	3			

Jika karakter 1 target dengan karakter 2 sumber berbeda maka masukkan nilai terkecil dari :

a. nilai cell (x-1, y-1) +1, pada matrix diatas adalah 2

b. nilai cell (x-1, y) + 1, pada matrix diatas adalah 3

c. nilai cell (x, y-1) + 1, pada matrix diatas adalah 1

maka nilai jarak antara karakter 1 target dengan karakter 2 sumber adalah 1.

Iterasi 1,3

s/t		2	2	1
	0	1	2	3
2	1	0		
4	2	1		
1	3	2		

Iterasi 2,1

s/t		2	2	1
	0	1	2	3
2	1	0	0	
4	2	1		
1	3	2		

Iterasi 2,2

s/t		2	2	1
	0	1	2	3
2	1	0	0	
4	2	1	1	
1	3	2		

Iterasi 2,3

s/t		2	2	1
	0	1	2	3
2	1	0	0	
4	2	1	1	
1	3	2	2	

Iterasi 3,1

s/t		2	2	1
	0	1	2	3
2	1	0	0	1
4	2	1	1	
1	3	2	2	

Iterasi 3,2

s/t		2	2	1
	0	1	2	3
2	1	0	0	1
4	2	1	1	1
1	3	2	2	

Iterasi 3,3

s/t		2	2	1
	0	1	2	3
2	1	0	0	1
4	2	1	1	1
1	3	2	2	1

Setelah selesai dilakukan proses iterasi maka kita bisa mendapatkan total jarak string sumber dengan string target di cell 3,3 atau panjang string sumber, panjang string target yaitu 1.

s/t		2	2	1
	0	1	2	3
2	1	0	0	1
4	2	1	1	1
1	3	2	2	1

Hingga tahap ini dapat disimpulkan bahwa jarak dari citra sumber A dengan citra target X mendekati dengan jarak adalah 1.

Algoritma *levenshtein distance*:

```
Procedure levenshtein distance;
kamus
i, j, s1 , s2, m, cost : integer
algoritma
for i ← 0 to s1.length do
for j ← 0 to s2.length do
if i = 0 then
m[i,j] ← j {perbandingan dengan kosong}
else if j = 0 then
m[i,j] ← i {perbandingan dengan kosong}
else {implementeasi pemrograman dinamis}
if s1[i] = s2[j] then
```

```
cost ← 0
else
cost ← 1
m[i,j] = minimum (
m[i-1, j-1] + cost, {substitusi}
m[i-1,j] + 1, {penghapusan}
m[i ,j-1] + 1, {penambahan}
)
return m[s1.length, s2.length]
```

6. Proses pengolahan citra target

Hal-hal yang harus dilakukan yaitu:

a. Merubah citra target menjadi citra biner.
b. Normalisasi citra target menggunakan proses tranformasi agar dapat menjadi citra yang siap untuk dibandingkan, citra yang siap dibandingkan adalah memiliki ukuran yang sama dengan citra sumber yang akan dibandingkan dalam hal ini berukuran 64 pixel x 64 pixel.
c. Segmentasi pada citra target dalam bentuk ukuran 8 pixel x 8 pixel.
d. Translasi setiap baris pada setiap segment citra target ditransalasikan menjadi deret angka atau karakter string dengan mengubah setiap baris pixel pada segment menjadi rangkaian angka biner dengan merubah pixel berwarna hitam menjadi angka 1 dan pixel berwarna putih menjadi 0, kemudian rangkaian biner tersebut ditransalasikan kembali menjadi angka desimal atau karakter.

3.2.3 Analisis keluaran

Dalam pembangunan perangkat lunak ini diharapkan perangkat lunak berbasis algoritma *levenshtein distance* tersebut dapat mengenali gambar yang utuh dan gambar yang tidak utuh. Setelah dilakukan proses pencarian terhadap

gambar tersebut akan dihasilkan keluaran berupa gambar yang telah dibinerisasi, selain itu perangkat lunak ini juga akan menampilkan jarak gambar.

3.4 Speifikasi kebutuhan non fungsional

Spesifikasi kebutuhan non fungsional menggambarkan keadaan sistem yang akan digunakan untuk mengolah aplikasi ini, diantaranya perangkat keras dan perangkat lunak.

3.4.1 Analisis Perangkat Keras Pendukung

Perangkat keras yang digunakan selama pengembangan sistem ini memiliki spesifikasi sebagai berikut :

1. Procesor 1,6 GHz
2. monitor 14 “digital”
3. Hard disk 40 GB
4. Memory 1 GB
5. Keyboard
6. Mouse

3.4.2 Analisis Perangkat Lunak Pendukung

Kebutuhan perangkat lunak merupakan faktor-faktor yang harus dipenuhi untuk merancang sebuah perangkat lunak sehingga perangkat lunak tersebut sesuai dengan maksud dan tujuan perangkat lunak tersebut di buat. Beberapa perangkat lunak pendukung sebagai berikut :

a. Sistem Operasi Windows XP
b. Visual basic 6.0
c. MySQL

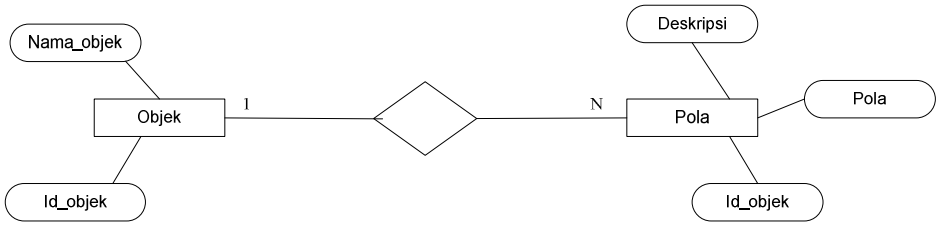
d. Adobe Photoshop

3.4.3 Analisis Pengguna

Untuk menggunakan program ini, tidak terbatas hanya untuk pengguna IT saja, namun program ini diharapkan dapat digunakan oleh semua kalangan yang sudah terbiasa menggunakan komputer berbasis windows.

3.5 Analisis Basis Data

Dalam memodelkan data dan menggambarkan hubungan antara data yang ada pada sistem digunakan alat bantu yaitu *Entity Relationship Diagram* (ERD). Berikut adalah ERD yang ada pada sistem, yang dapat pada gambar 3.9



Gambar 3.8 ERD pada sistem pengenalan bentuk geometri

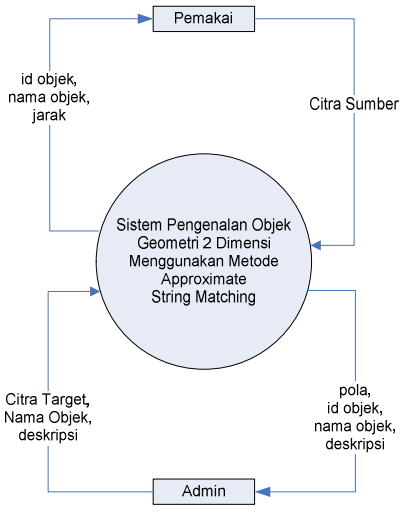
3.6 Analisis Kebutuhan Fungsional

Merupakan hal-hal yang dibutuhkan untuk merancang suatu program aplikasi atau perangkat lunak.

3.6.1 Diagram Konteks

Diagram konteks merupakan suatu kasus data flow diagram yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Perancangan ini melibatkan dua entiti luar, yaitu Admin dan Pemakai. Admin

adalah entiti luar yang melakukan proses masukan data objek geometri 2 dimensi untuk menambah database pengetahuan dari sistem. Data yang dimasukkan admin adalah data nama dan gambar berukuran 64x64 pixel yang siap untuk dilakukan proses ekstraksi pola tanpa proses transformasi. Pemakai adalah entiti luar yang memberikan masukan data gambar yang ingin dikenali oleh sistem. Untuk lebih jelasnya akan digambarkan dalam diagram konteks sebagai berikut :

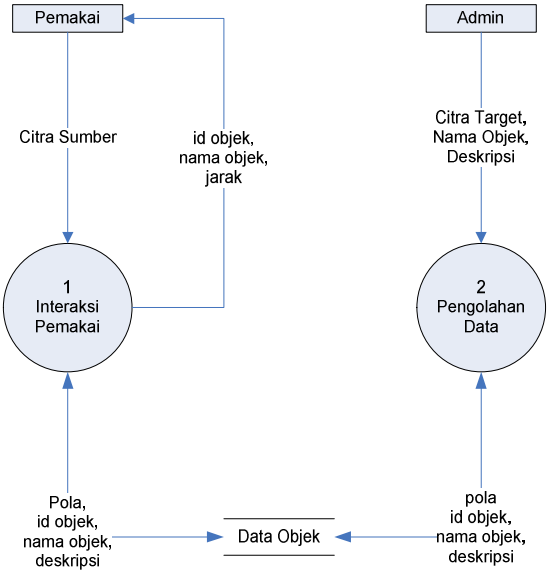


Gambar 3.9 Diagram Konteks pengenalan bentuk geometri

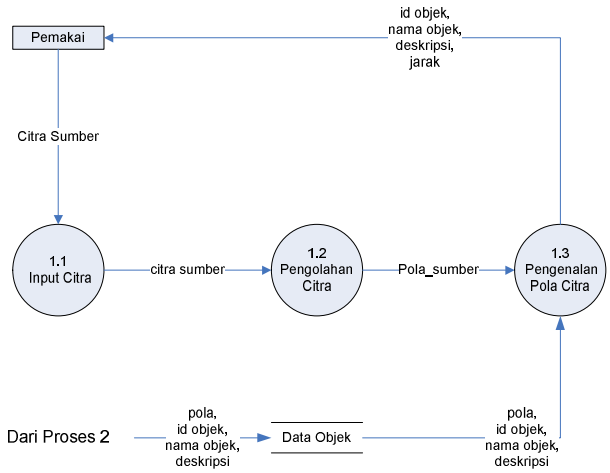
3.6.2 Data Flow Diagram (DFD)

Data flow diagram atau DFD merupakan alat yang banyak digunakan oleh para pengembang sistem perangkat lunak untuk menggambarkan suatu sistem yang telah ada ataupun sistem baru yang akan dikembangkan. Secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut akan disimpan. Data flow diagram merupakan alat yang digunakan dalam meggambarkan arus data yang mengatur dari suatu proses ke proses selanjutnya. DFD juga merupakan dokumentasi dari sistem yang baik.

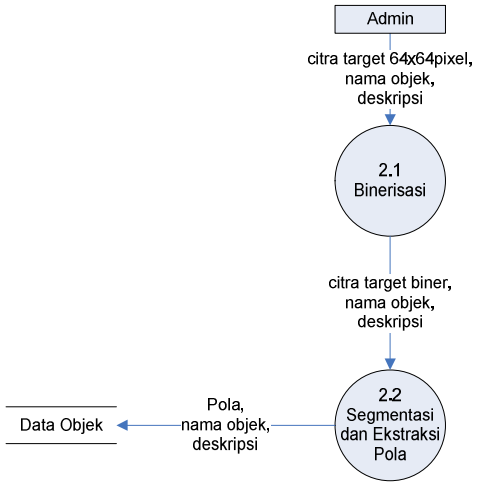
Dibawah ini merupakan DFD sistem pengenalan bentuk geometri yang akan dibuat, dimana proses interaksi pemakai terdiri dari tiga buah sub proses, yaitu proses input citra, proses pengolahan dan proses pengenalan pola citra. Pada proses pengolahan citra akan terjadi proses perubahan citra sumber menjadi citra biner berukuran 64x64 pixel hingga didapat pola, sedangkan pada proses pengenalan pola citra adalah proses pencarian pola yang paling cocok dengan pola pada data. Untuk lebih jelasnya dapat dilihat pada gambar DFD berikut ini :



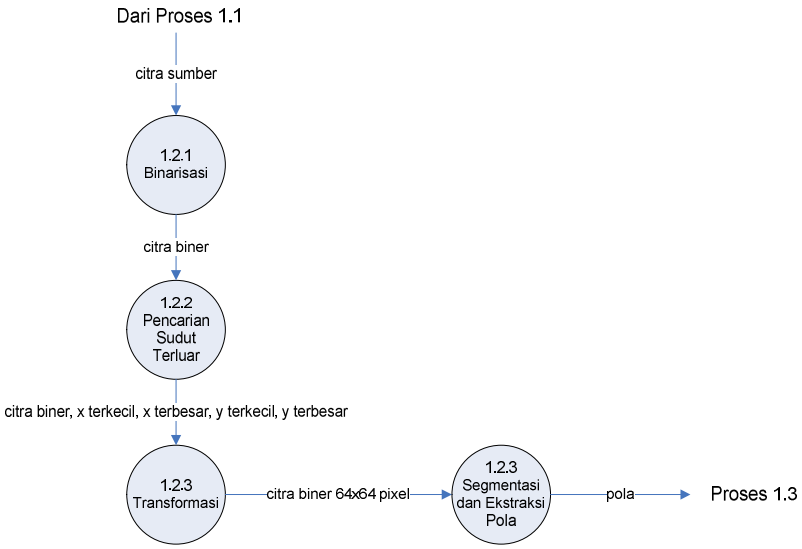
Gambar 3.10 DFD Level 1 aktivitas pengenalan bentuk geometri



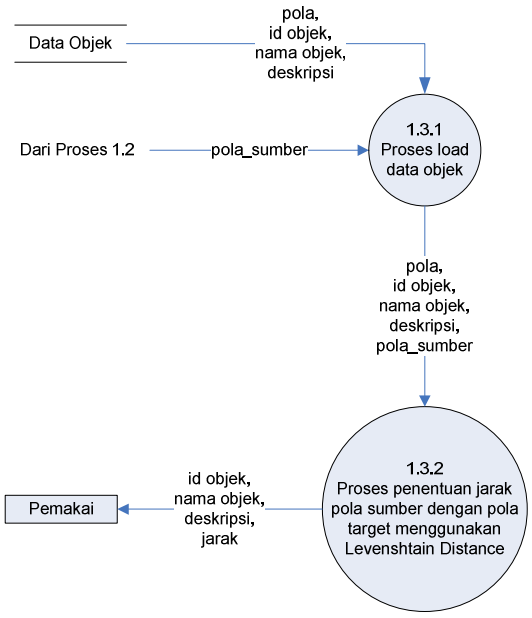
Gambar 3.11 DFD Level 2 dari proses 1 interaksi pemakai



Gambar 3.12 DFD Level 2 proses 2 interaksi admin



Gambar 3.13 DFD Level 3 dari proses 1.2 pengolahan citra sumber



Gambar 3.14 DFD Level 3 dari proses 1.3 pengolahan pola citra

C. Spesifikasi Proses

Dari Data Flow Diagram (DFD) diatas, maka proses-proses yang terdapat dalam sistem perangkat bantu ini dapat dijelaskan dengan tabel-tabel spesifikasi proses berikut :

Tabel 3.1 Spesifikasi Proses

No.	**Proses**	**Keterangan**
1	No. Proses	1.1
	Nama Proses	Aktivitas input citra
	Source (sumber)	Pemakai
	Input	Citra sumber,
	Output	Info citra sumber
	Destination (tujuan)	Proses 1.2
	Logika Proses	*Begin* { user masukan citra sumber} If input citra sumber then tampil citra sumber *End*
No.	**Proses**	**Keterangan**
2	No. Proses	1.2
	Nama Proses	Proses pengolahan citra
	Source (sumber)	Proses 1.1
	Input	Citra sumber
	Output	Pola citra sumber
	Destination (tujuan)	Proses 1.3
	Logika Proses	*Begin* {Pemakai menggunakan tombol cari} If input citra sumber then olah citra sumber *End*
No.	**Proses**	**Keterangan**
3	No. Proses	1.3
	Nama Proses	Proses pengenalan pola citra
	Source (sumber)	Proses 1.2
	Input	Pola citra sumber
	Output	Id objek, nama objek, deskripsi, jarak
	Destination (tujuan)	Pemakai
	Logika Proses	*Begin*

		{input pola, jarak terbaik dan jumlah maksimum dari proses 1.2} If input pola, then cari pola target yang mendekati pola sumber Else Objek tidak dikenali *End*
No.	**Proses**	**Keterangan**
4	No. Proses	1.2.1
	Nama Proses	Proses Binarisasi
	Source (sumber)	Proses 1.1
	Input	Citra sumber
	Output	Citra biner
	Destination (tujuan)	User
	Logika Proses	*Begin* {Proses 1.1 input citra sumber) If input citra sumber then grayscalling and thresholding citra sumber *End*
No.	**Proses**	**Keterangan**
5	No. Proses	1.2.2
	Nama Proses	Proses pencarian titik terluar
	Source (sumber)	Proses 1.2.1
	Input	citra biner
	Output	Citra biner, x terkecil, x terbesar, y terkecil, y terbesar
	Destination (tujuan)	Proses 1.2.3
	Logika Proses	*Begin* If input citra biner then cari titik terluar *End*
No.	**Proses**	**Keterangan**
6	No. Proses	1.2.3
	Nama Proses	Proses Tranformasi
	Source (sumber)	Proses 1.2.2
	Input	Citra biner, x terkecil, x terbesar, y terkecil, y terbesar
	Output	Citra biner 64x64 pixel
	Destination (tujuan)	Proses 1.2.4
	Logika Proses	*Begin*

		If input citra biner, x terkecil, x terbesar, y terkecil, y terbesar then transformasi citra biner End
No.	**Proses**	**Keterangan**
7	No. Proses	1.2.4
	Nama Proses	Proses segmentasi dan ekstraksi pola
	Source (sumber)	Proses 1.2.3
	Input	Citra biner 64x64 pixel
	Output	Pola citra sumber
	Destination (tujuan)	Proses 1.3
	Logika Proses	*Begin* If input citra biner 64x64 pixel then segmented and extract pattern *End*
No.	**Proses**	**Keterangan**
8	No. Proses	1.3.1
	Nama Proses	Proses load data objek
	Source (sumber)	Proses 1.2
	Input	Pola citra sumber
	Output	Pola citra sumber, pola, id objek, nama objek dan deskripsi.
	Destination (tujuan)	Proses 1.3
	Logika Proses	*Begin* Load data objek and pola from databases *End*
No.	**Proses**	**Keterangan**
9	No. Proses	1.3.2
	Nama Proses	Proses penentuan jarak pola sumber dengan pola target menggunakan Levenshtain Distance
	Source (sumber)	Proses 1.3.1
	Input	Pola citra sumber, pola, id objek, nama objek dan deskripsi.
	Output	Id objek, nama objek, deskripsi dan jarak.
	Destination (tujuan)	Proses 1.3
	Logika Proses	*Begin* For each pola begin Jarak := levenshtaindistance(pola, pola_sumber)

		end *End*
No.	**Proses**	**Keterangan**
10	No. Proses	2.1
	Nama Proses	Proses Binerisasi
	Source (sumber)	Admin
	Input	Citra target 64x64 pixel
	Output	Citra target biner, Nama Objek, Deskripsi.
	Destination (tujuan)	Proses 2.2
	Logika Proses	*Begin* {admin input citra target} If inout citra target 64x64 pixel then grayscalling and thresholding citra target *End*
No.	**Proses**	**Keterangan**
11	No. Proses	2.2
	Nama Proses	Proses segmentasi dan ekstraksi pola
	Source (sumber)	Proses 2.1
	Input	Citra target biner, Nama Objek, Deskripsi.
	Output	Pola, Nama Objek, Deskripsi.
	Destination (tujuan)	Database data objek
	Logika Proses	*Begin* If input citra target biner then segmented and extract pattern *End*

Tabel 3.2 Kamus Data pada alir data

NO	Kamus	Keterangan
1	Nama	Citra Biner
	Deskripsi	Citra sumber inputan pemakai dalam mode warna hitam dan putih
	Where used / how used	User Proses 1.2.1 citra biner
2	Nama	Citra biner 64x64 pixel
	Deskripsi	Citra target inputan pemakai berukuran 64x64 pixel dalam mode warna hitam dan putih.

	Where used / how used	Admin Proses 2.1 proses binerisasi
3	Nama	Citra sumber
	Deskripsi	Citra sumber inputan pemakai dalam mode warna RGB.
	Where used / how used	User Proses 1.2 aktivitas input citra
4	Nama	Citra target 64x64 pixel
	Deskripsi	Citra target inputan admin berukuran 64x64 pixel dalam mode warna RGB.
	Where used / how used	Admin Proses 2.1 grayscalling
5	Nama	Citra target biner
	Deskripsi	Citra target inputan admin berukuran 64x64 pixel dalam mode warna hitam dan putih.
	Where used / how used	Admin - Proses 1.2.3 transformasi - Proses 1.2.4 segmentasi dan ekstraksi pola
6	Nama	Deskripsi
	Deskripsi	Penjelasan admin mengenai objek citra target
	Where used / how used	Admin - Proses 2.1 binerisasi - Proses 2.2 segmentasi dan ekstrasi pola
7	Nama	Nama Objek
	Deskripsi	Informasi nama objek dari citra target yang diinput oleh admin.
	Where used / how used	Admin - Proses 1.3 pengenalan pola - Proses 2.1 binerisasi - Proses 2.2 segmentasi dan ekstrasi pola
8	Nama	Pola
	Deskripsi	Deret pola dalam decimal setelah proses segementasi citra biner.
	Where used / how used	Database
9	Nama	x terbesar
	Deskripsi	Posisi objek paling luar sebelah kanan.
	Where used / how used	Admin - Proses 1.2.2 pencarian sudut terluar - Proses 1.2.3 transformasi
10	Nama	x terkecil
	Deskripsi	Posisi objek paling luar sebelah kiri.
	Where used / how used	Admin - Proses 1.2.2 pencarian sudut terluar - Proses 1.2.3 transformasi

11	Nama	y terbesar
	Deskripsi	Posisi objek paling luar sebelah bawah.
	Where used / how used	Admin - Proses 1.2.2 pencarian sudut terluar - Proses 1.2.3 transformasi
12	Nama	y terkecil
	Deskripsi	Posisi objek paling luar sebelah atas.
	Where used / how used	Admin - Proses 1.2.2 pencarian sudut terluar - Proses 1.2.3 transformasi

3.7 Perancangan Sistem

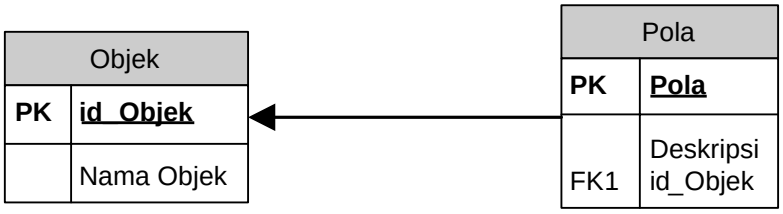
Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

3.7.1 Perancangan Basis Data

Perancangan basis data merupakan tahapan untuk memodelkan model konseptual ke model basis data yang akan dipakai. Perancangan basis data terbagi menjadi dua yaitu skema relasi dan perancangan struktur data.

3.7.2 Skema Relasi

Skema relasi berfungsi menggambarkan relasi atau keterkaitan antar tabel yang mempunyai atribut kunci utama yang sama, sehingga tabel-tabel tersebut menjadi satu kesatuan yang dihubungkan oleh atribut kunci tersebut. Pada proses ini elemen-elemen data dikelompokan menjadi satu file database beserta entitas dan hubungannya. Gambar 3.9 berikut ini merupakan skema relasi database pada pengenalan bentuk geometri.



Gambar 3.15 Skema relasi database pengenalan bentuk geometri

3.7.3 Perancangan Struktur Tabel

Perancangan struktur tabel adalah perancangan tabel-tabel yang akan digunakan pada database. Tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem ini adalah:

1. Tabel Objek

Nama Field	Type	Ukuran	Kunci
Id_objek	Integer	11	Primary Key
Nama_objek	Varchar	25	

2. Tabel Pola

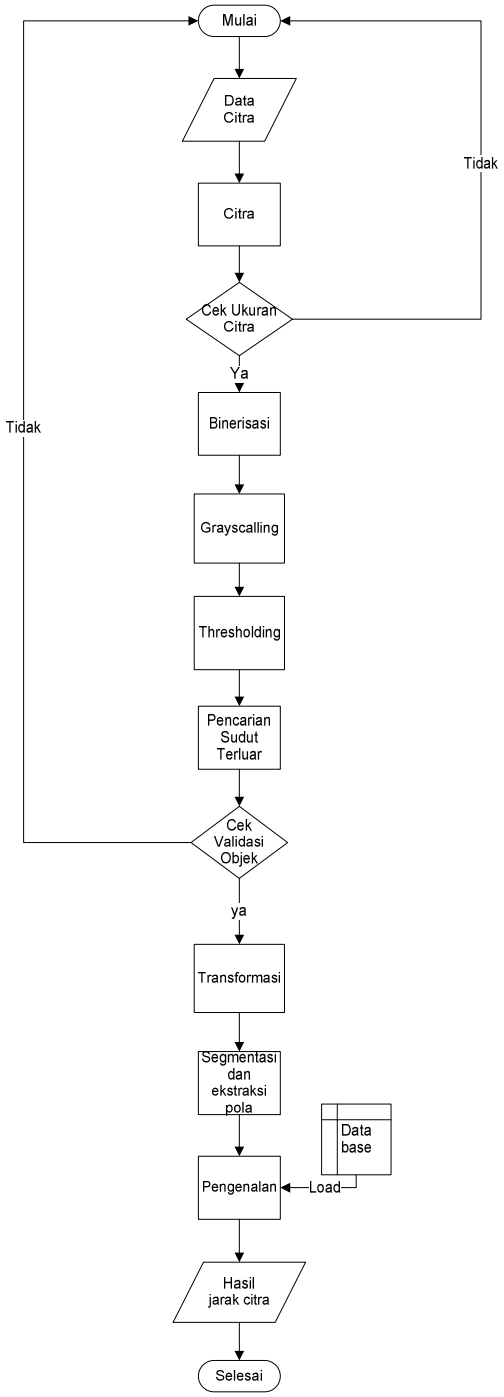
Nama Field	Type	Ukuran	Kunci
Pola	Text		Primary Key
Id_objek	Integer	11	Foreign Key
Deskripsi	Varchar	50	

3.7.4 Perancangan Prosedural

Perancangan prosedural dilakukan dengan membuat *flowchart* sistem dan algoritma untuk metode yang digunakan. *Flowchart* merupakan suatu cara untuk menggambarkan langkah-langkah kerja program yang meliputi input, proses, dan

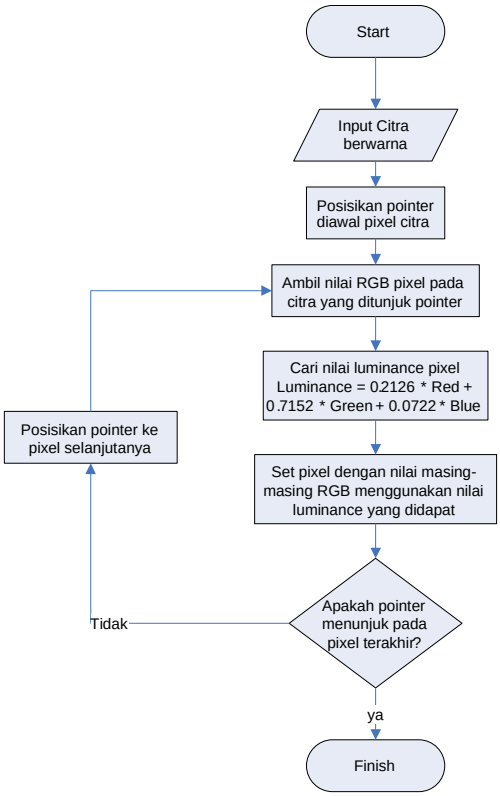
output. Sedangkan algoritma berfungsi untuk mempermudah dalam pengkodean (*coding*) kedalam bahasa pemograman.

3.7.4.1 Flowchart kerja metode Approximate String Matching



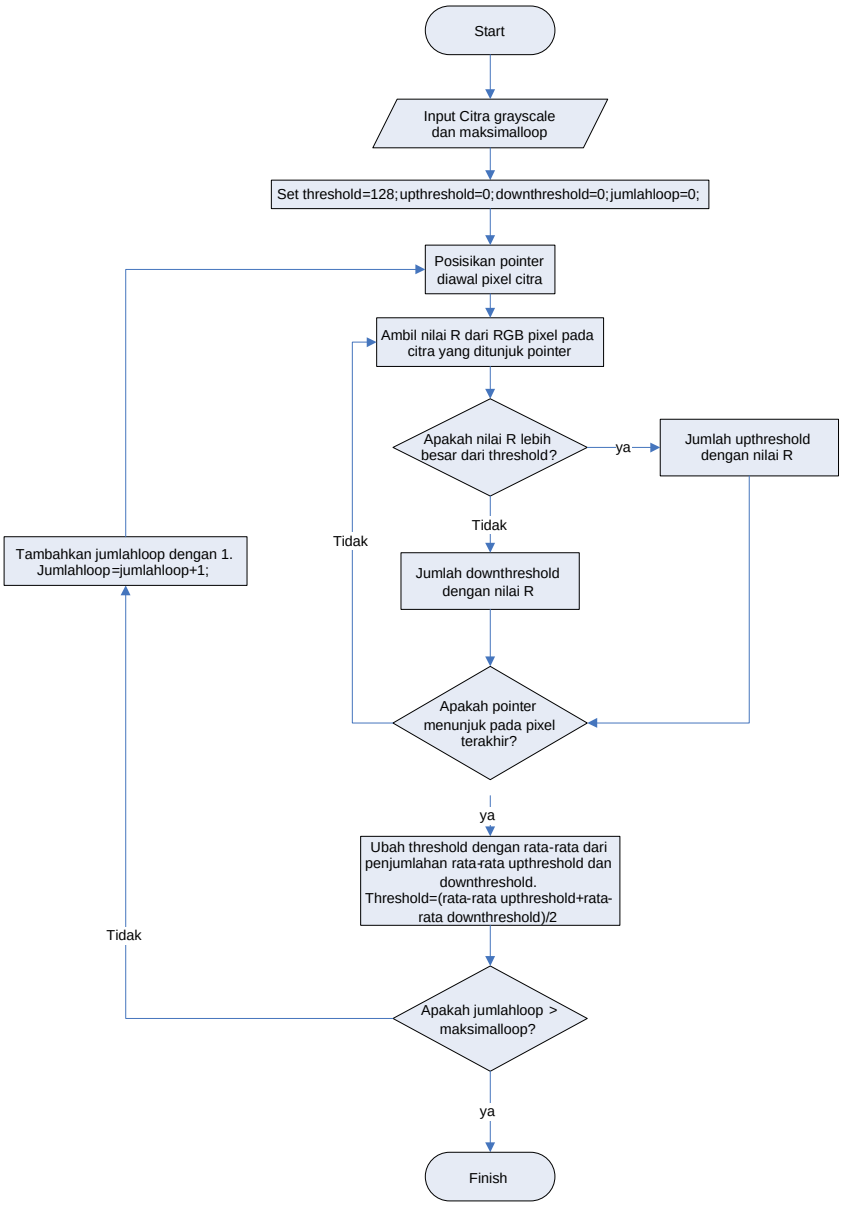
Gambar 3.17 Flowchart kerja metode Approximate String Matching

3.7.4.2 Flowchart proses grayscaling



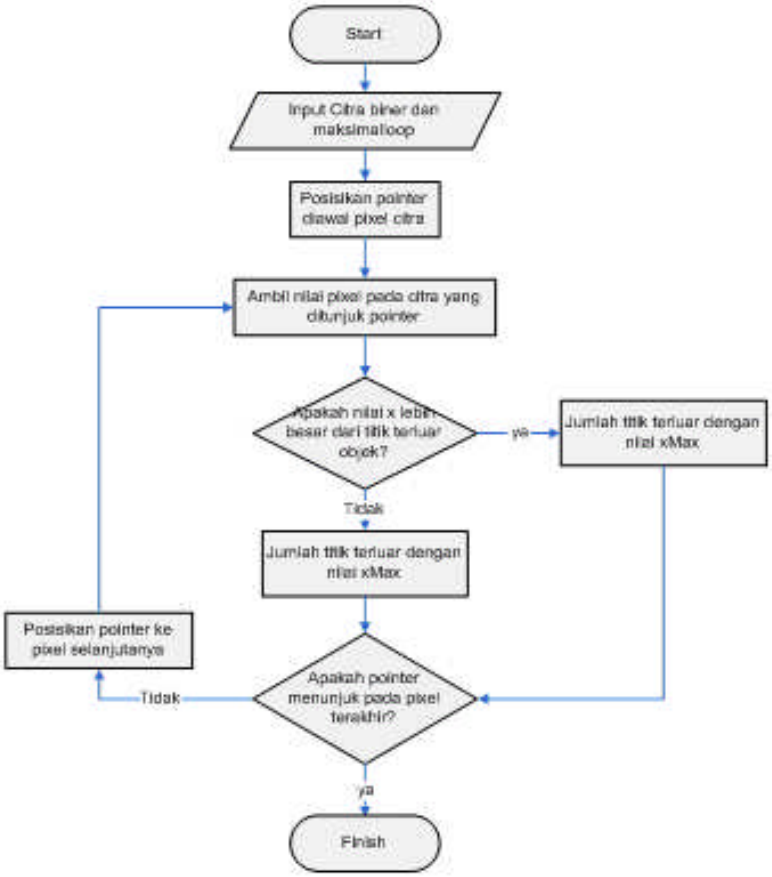
Gambar 3.18 Flowchart grayscaling

3.7.4.3 Flowchart proses thresholding



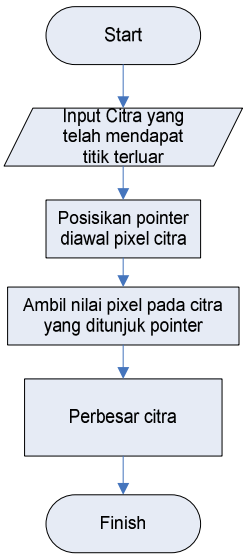
Gambar 3.19 Flowchart threholding

3.7.4.5 Flowchart proses pencarian sudut terluar



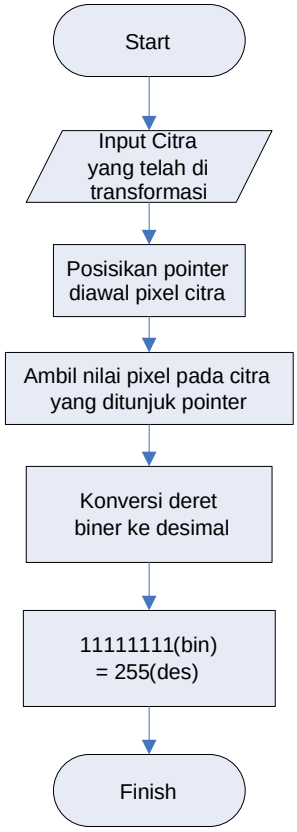
Gambar 3.20 Flowchart pencarian sudut terluar

3.7.4.6 Flowchart proses transformasi



Gambar 3.21 Flowchart transformasi

3.7.4.7 Flowchart proses segmentasi



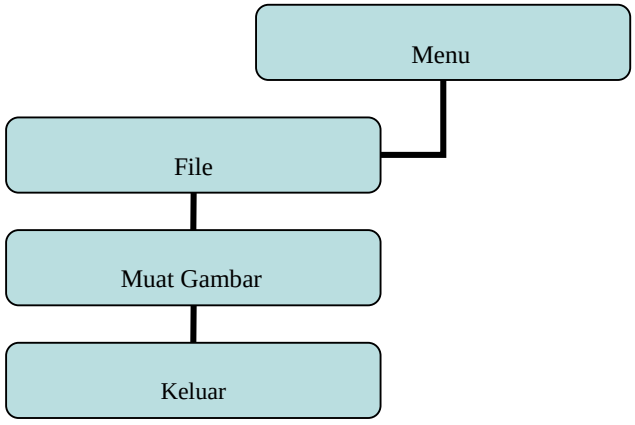
Gambar 3.21 Flowchart segmentasi

3.7.5 Perancangan Antar Muka

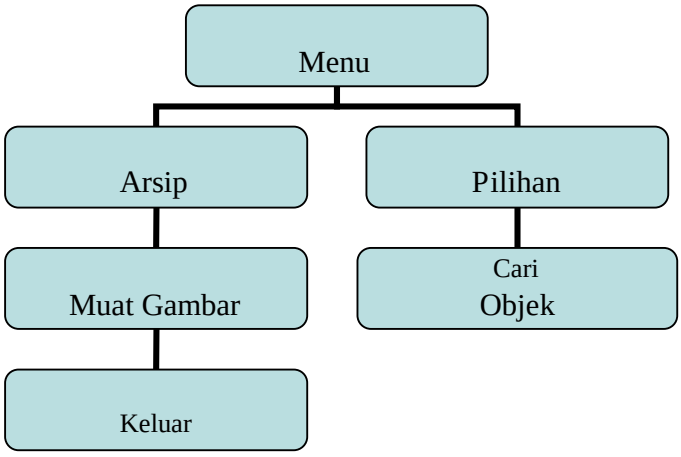
Perancangan antarmuka yang telah dibuat meliputi beberapa perancangan diantaranya perancangan struktur menu, perancangan antar muka dan jaringan semantik.

3.7.5.1 Perancangan Struktur Menu

Berikut adalah perancangan struktur menu pengenalan bentuk geometri 2 dimensi pada citra menggunakan metode *approximate string matching* :

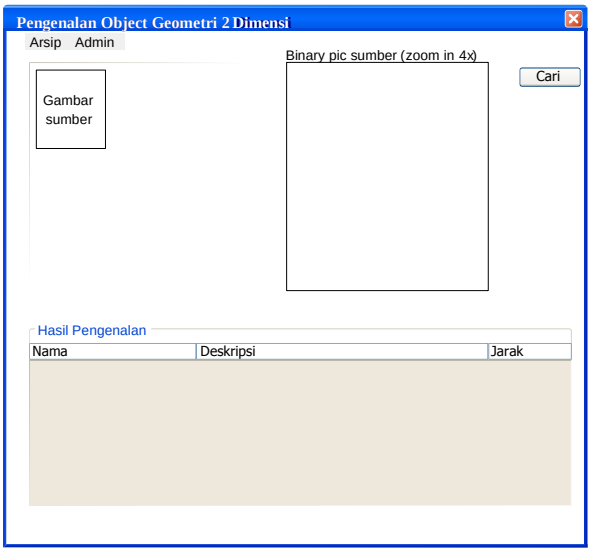


Gambar 3.22 Struktur Menu *User*

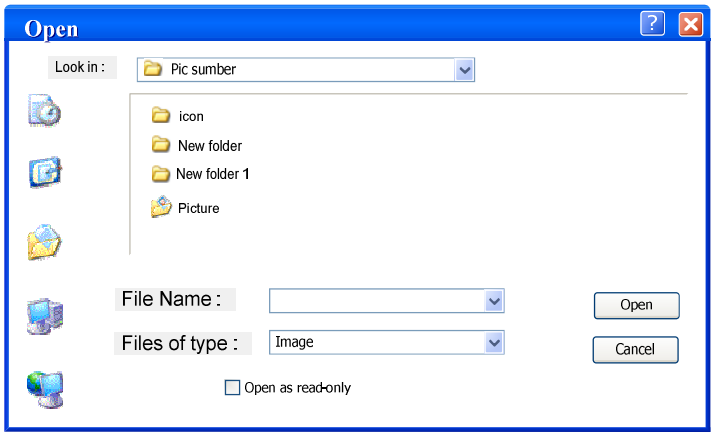


Gambar 3.23 Struktur Menu Admin

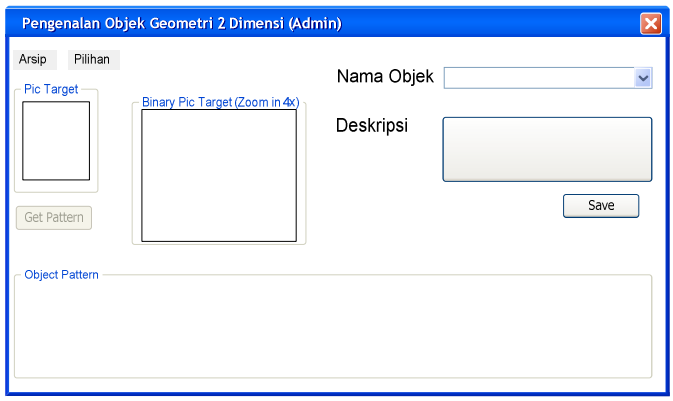
3.7.5.2 Perancangan Antar Muka

T01	Navigator
	- Klik Arsip menuju T02 - Klik Admin menuju T03

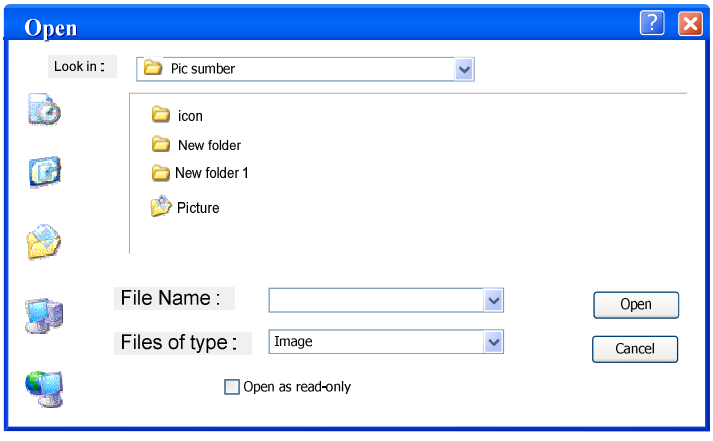
Gambar 3.24 Perancangan Antarmuka Menu Utama

T02	Navigator
	- Klik gambar menuju T01 - Ukuran gambar harus lebih dari 64x64 pixel jika tidak akan menuju P01 - Tinggi & lebar objek tidak boleh kecil dari 64x64 pixel jika tidak akan menuju P02

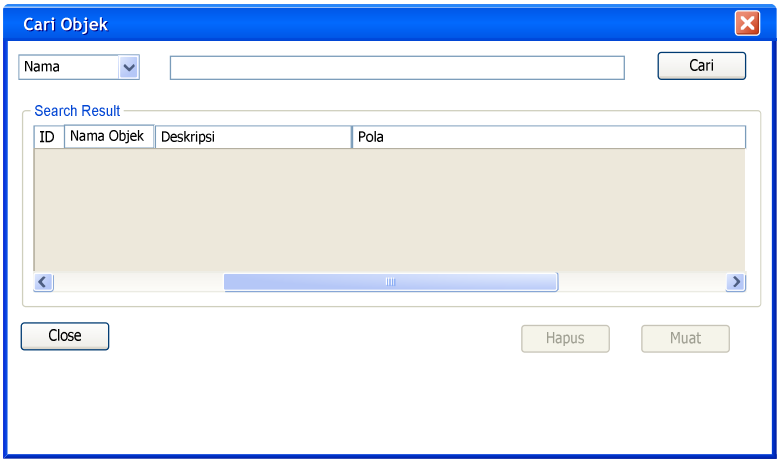
Gambar 3.25 Perancangan Antarmuka muat gambar

T03	Navigator
	- Muat gambar klik arsip menuju T04 - jika nama objek tidak ditulis maka nenuju P03 - jika panjang nama objek lebih dari 25 menuju P04 - Jika deskripsi tidak ditulis maka menuju P05 -Klik save menuju P06 - Klik pilihan menuju T05

Gambar 3.26 Perancangan Antarmuka Admin

T04	Navigator
	- Muat gambar klik open menuju T03 - Jika ukuran tidak 64x64 pixel maka menuju P07 - Jika tinggi & lebar objek tidak 64x64 pixel menuju P08

Gambar 3.27 Perancangan Antarmuka Muat Gambar

T05	Navigator
	- Admin memasukan nama objek yang akan dicari - Jika data objek tidak ditemukan akan menuju P09 - Jika dikosongkan akan menuju ke P10 - Klik hapus menuju P11, P12 - Klik muat dan close T03

Gambar 3.28 Perancangan Antarmuka cari objek

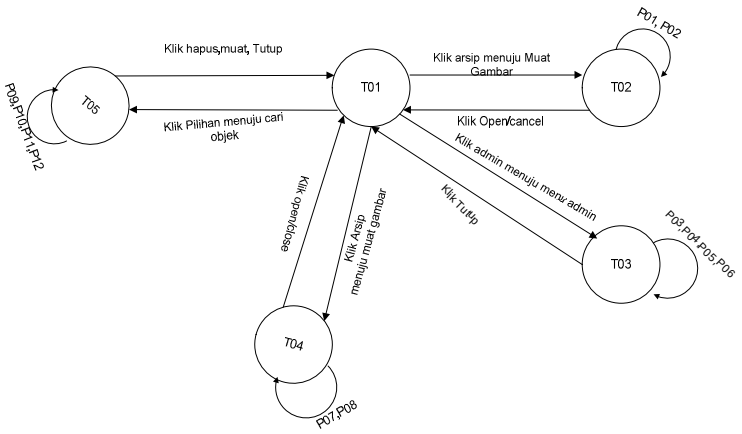
3.7.5.3 Rancangan Pesan



Gambar 3.29 Perancangan Pesan

3.7.5.4 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu halaman ke halaman lainnnya.



Gambar 3.30 Jaringan Semantik