



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODE DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

a. **Studi Literatur**

Melakukan studi yang berkaitan sistem ini dengan mengumpulkan teori-teori dan konsep yang bersifat relevan dari jurnal ilmiah, artikel dan buku yang berkaitan dengan metode *template matching* dan materi pendukung lainnya.

b. **Perancangan Aplikasi**

Melakukan perancangan awal dan analisa terhadap aplikasi yang akan dibangun, meliputi perancangan *Flowchart Diagram*, *Data Flow Diagram* dan *Entity Relationship Diagram*.

c. **Pembangunan aplikasi**

Melakukan pembangunan sistem dengan mengimplementasikan rancangan dan metode *Template Matching* yang telah dibahas pada landasan teori.

d. **Uji Coba**

Melakukan uji coba sistem terhadap objek citra, dan mengevaluasi hasil yang didapat.

3.2 Perancangan Aplikasi

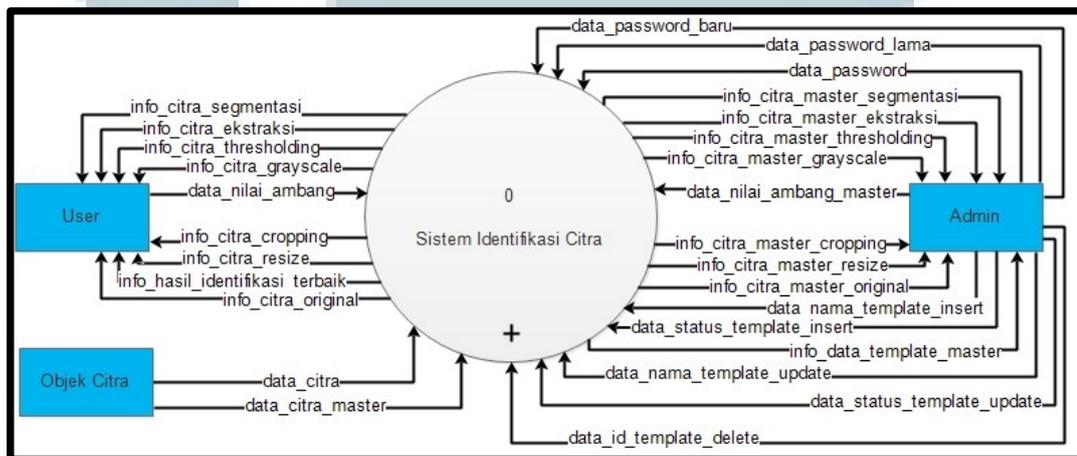
Perancangan aplikasi ini menggunakan beberapa diagram, antara lain *Flowchart Diagram*, *Data Flow Diagram* dan *Entity Relationship Diagram*.

Flowchart Diagram bertujuan untuk menggambarkan alur proses aplikasi,

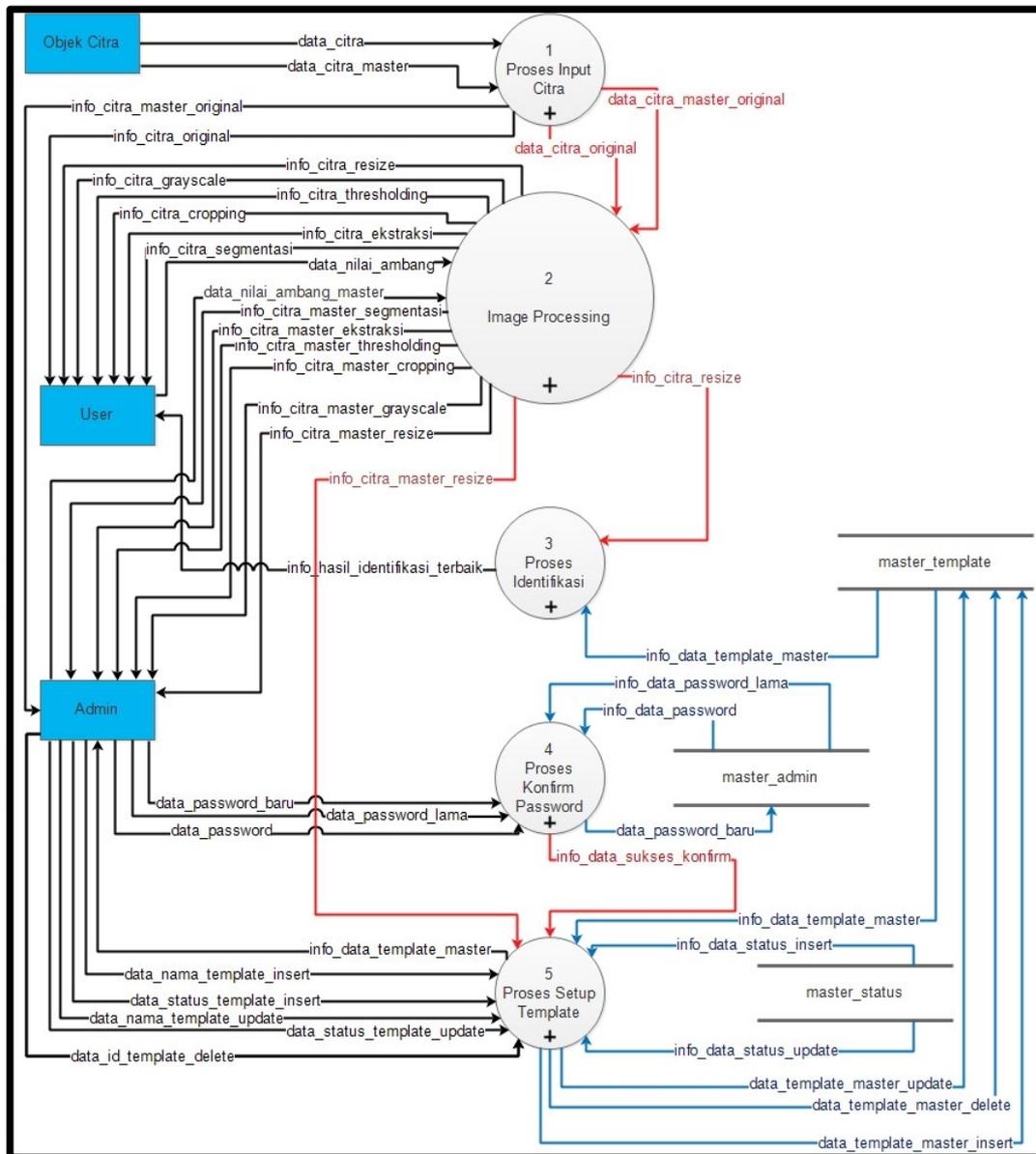
sedangkan *Data Flow Diagram* menjelaskan arus dari data sistem yang bertujuan untuk membantu memahami sistem secara terstruktur dan jelas. Kemudian *Entity Relationship Diagram* bertujuan untuk menggambarkan hubungan antar tabel dalam *database*.

3.2.1 Data Flow Diagram

Context Diagram sistem ini dapat dilihat pada Gambar 3.1. Sistem memiliki dua pengguna yaitu *user* dan *admin*, dan satu objek yaitu Objek Citra. *User* adalah pengguna yang hanya dapat menggunakan fitur identifikasi, sedangkan *admin* dapat menggunakan fitur *setup template*, sedangkan objek citra merupakan objek yang akan dikirim kedalam sistem oleh *user* dan *admin* kemudian objek tersebut mengirimkan data objeknya ke dalam sistem.

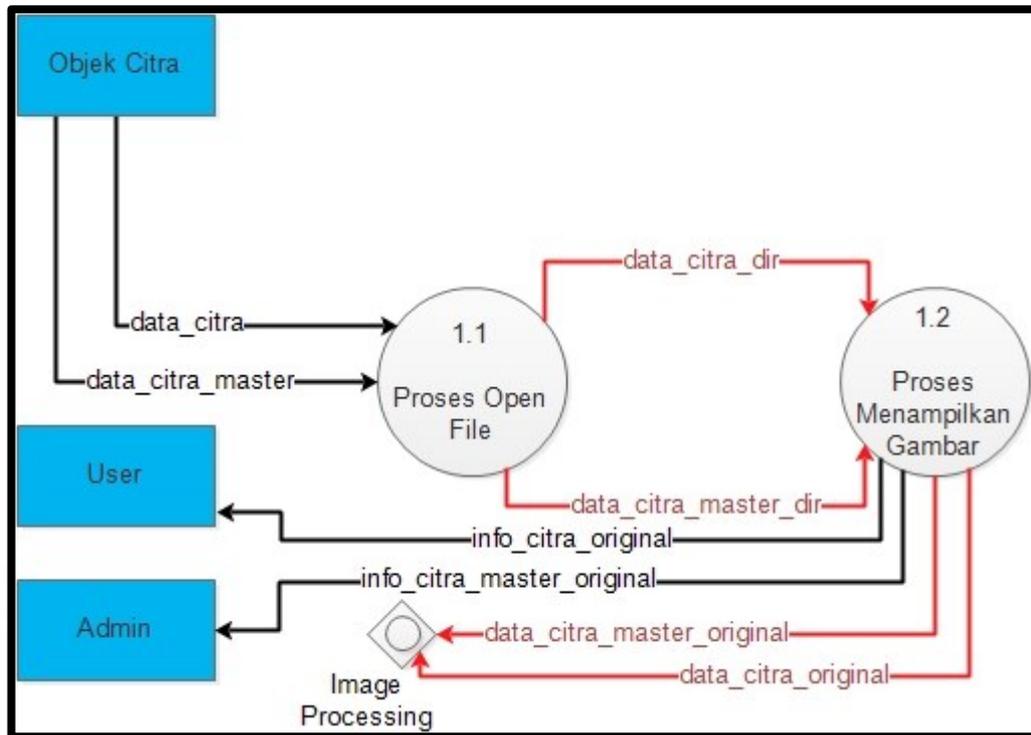


Gambar 3.1 DFD *Context Diagram* Sistem Identifikasi Citra



Gambar 3.2 DFD level 1 Sistem Identifikasi Citra

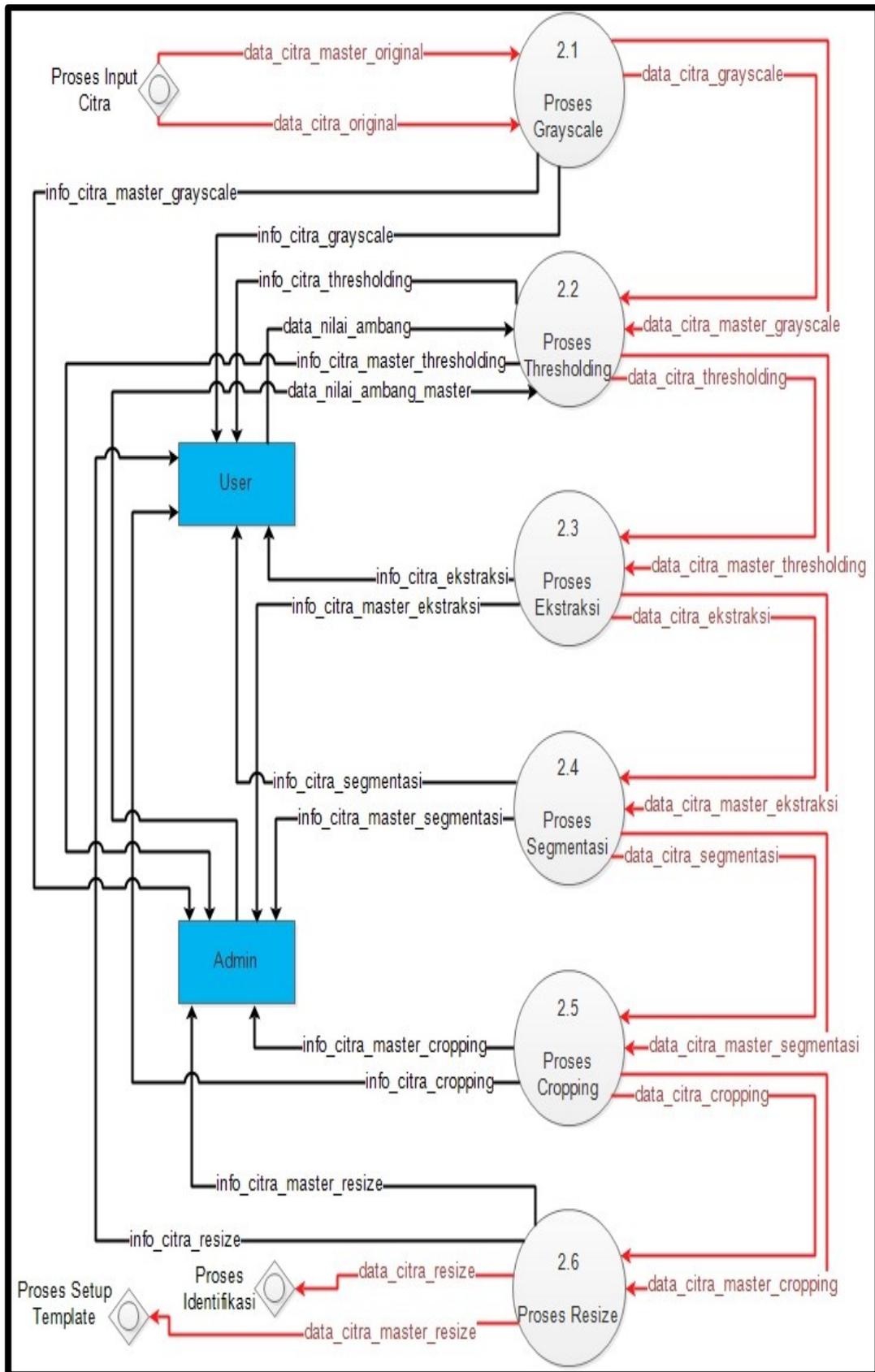
Gambar 3.2 menunjukkan aliran data pada setiap proses, perbedaan warna untuk mempermudah dalam membedakan aliran datanya. *User* dan *admin* memasukkan data citra yang akan diproses, setelah citra diproses melalui *image processing*, maka citra dapat diidentifikasi oleh *user* dan juga citra dapat disimpan ke dalam *database* oleh *admin* melalui proses *setup template*. Sebelum *admin* masuk ke dalam *setup template*, *admin* harus melalui proses konfirmasi *password* terlebih dahulu.



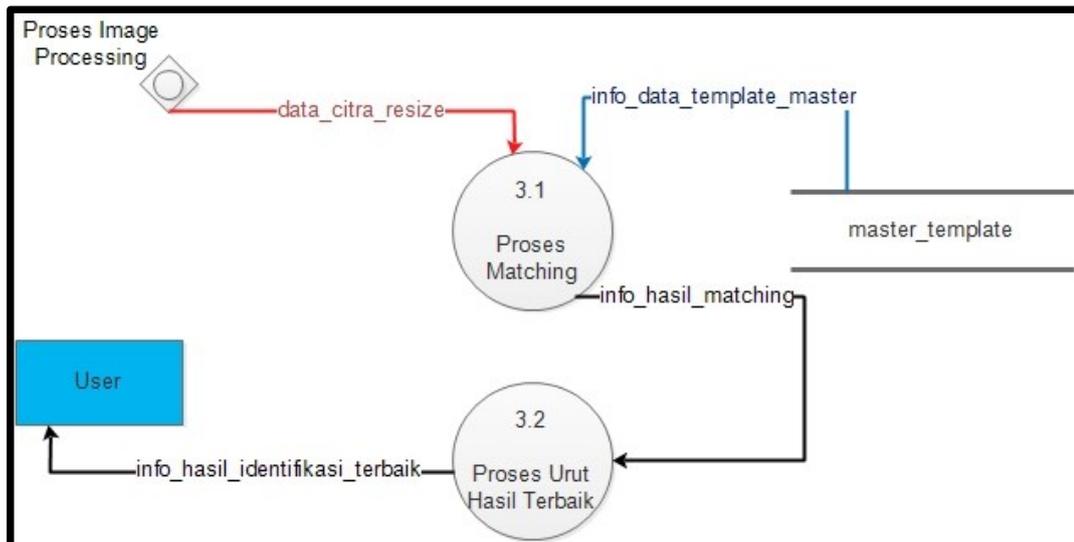
Gambar 3.3 DFD level 2 *Open File*

Gambar 3.3 menunjukkan aliran data proses *open file*. Pada proses ini *user* dan *admin* memilih data citra yang akan diproses oleh sistem, kemudian data citra yang dipilih akan ditampilkan ke *user interface* melalui proses menampilkan gambar dari data direktori citra.

Aliran data berikutnya adalah aliran data *Image Processing*. Proses ini memperoleh data citra masukan dari proses *open file*, yang kemudian akan melalui proses citra *grayscale*, proses citra *thresholding*, proses citra ekstraksi, proses citra segmentasi, proses citra *cropping*, dan proses citra *resize*. Hasil citra pada masing-masing proses akan ditampilkan ke pengguna. Gambar aliran data *Image Processing* ini dapat dilihat pada Gambar 3.4.

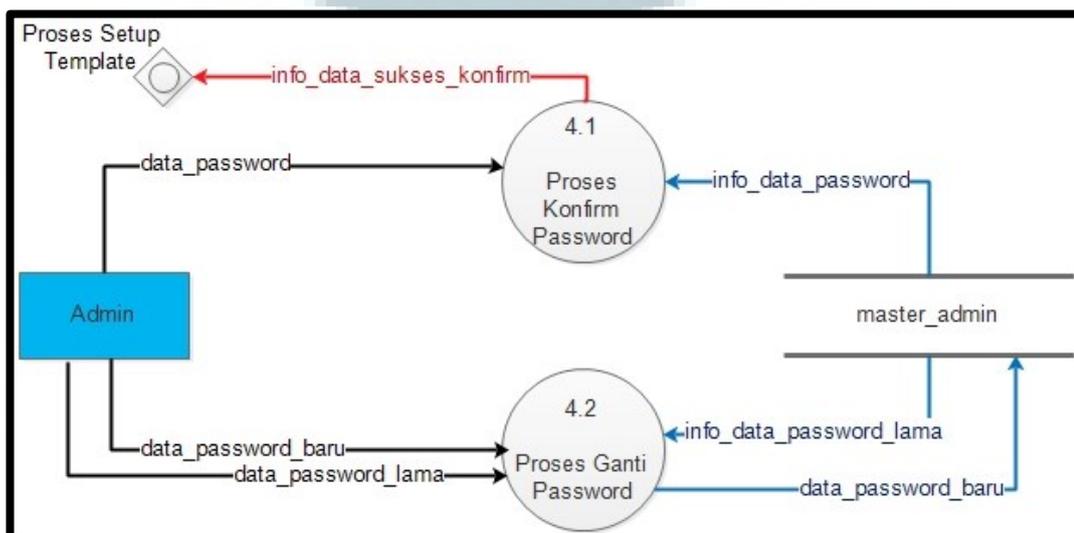


Gambar 3.4 DFD level 2 *Image Processing*



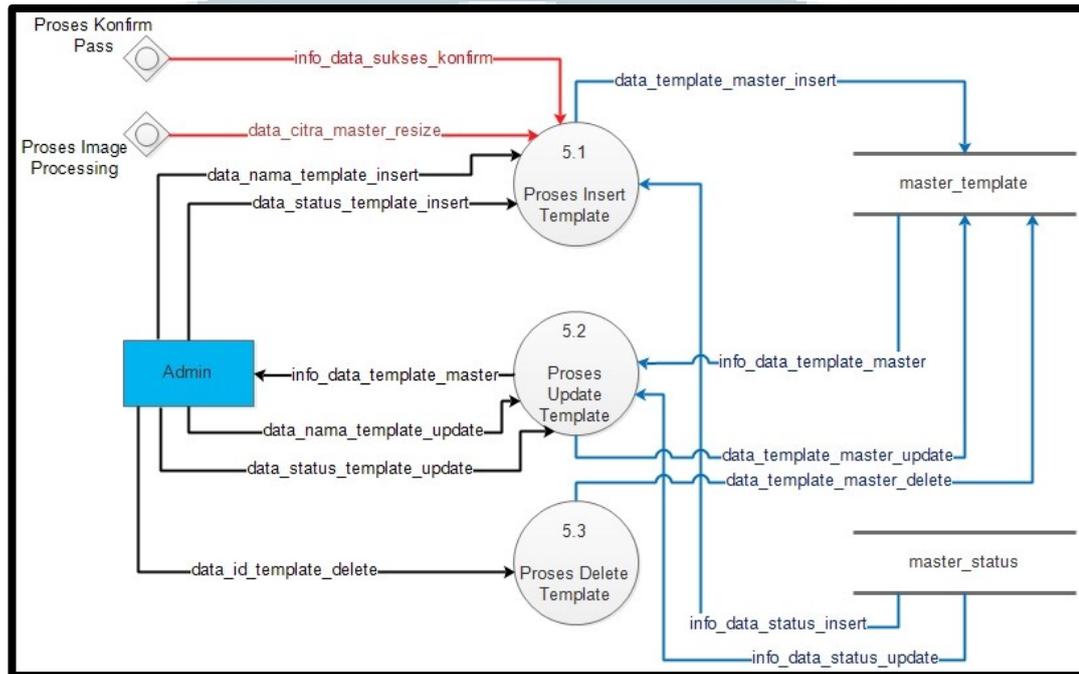
Gambar 3.5 DFD level 2 Identifikasi

Gambar 3.5 menunjukkan aliran data proses identifikasi. Proses ini melakukan identifikasi citra yang telah di-*resize*, kemudian sistem mengambil data citra *template* dari *database*. Setelah itu melakukan proses pencocokkan dan hasil dari masing-masing citra *template* akan diseleksi melalui proses urut hasil terbaik, maka data yang akan ditampilkan ke *user* adalah empat data *template* yang memiliki nilai kecocokan tertinggi.



Gambar 3.6 DFD level 2 Konfirmasi *Password*

Gambar 3.6 menunjukkan aliran data proses konfirmasi *password*. Proses ini membutuhkan data *password admin* yang akan dicocokkan dengan data *password* yang ada pada *database*. Proses ini juga dapat melakukan ganti *password* yang membutuhkan data *password* lama *admin* dan *password* baru *admin*, kemudian sistem melakukan proses *update* data ke *database*.

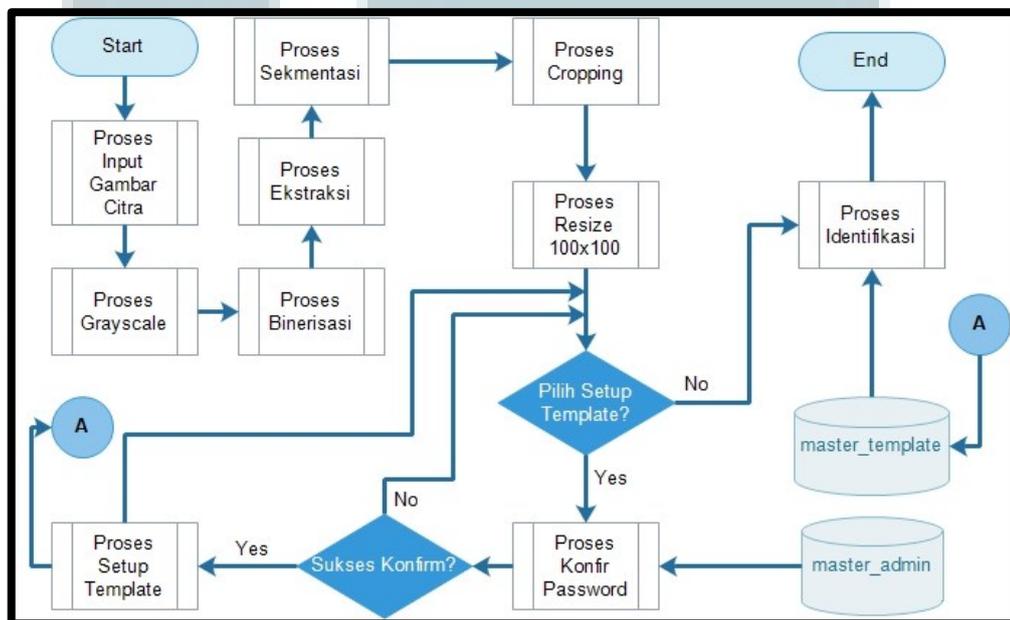


Gambar 3.7 DFD level 2 *Setup Template*

Gambar 3.7 menunjukkan aliran data *setup template*. Proses ini mendapatkan data citra dari hasil proses *resize*, kemudian *admin* memasukkan nama *template* dan status *template* yang akan disimpan ke dalam *database*. Jika *admin* ingin melakukan proses *update*, maka *admin* memasukkan nama *template* baru dan status baru. Pada proses *delete*, *admin* memilih data *template* id yang akan dihapus.

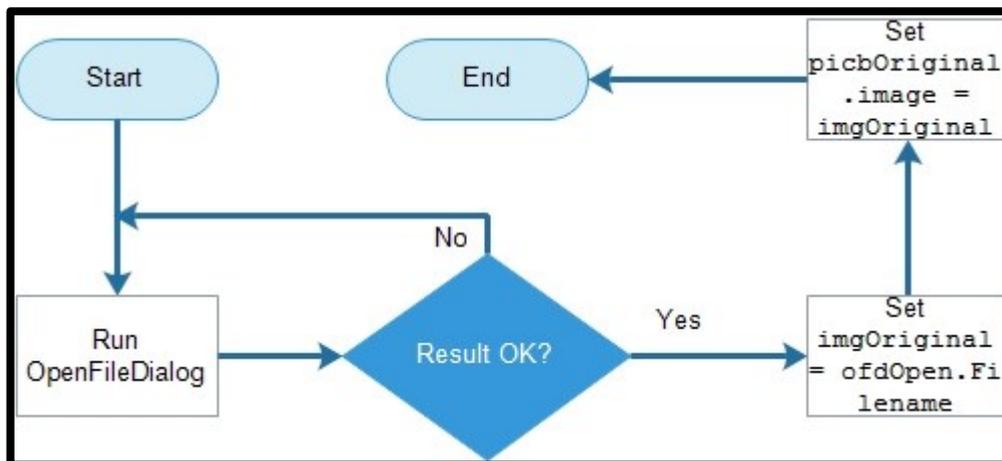
3.2.2 Flowchart Diagram

Aplikasi ini memiliki dua jenis pengguna, yaitu *user* dan *admin*. Pengguna *user* hanya dapat memasukkan gambar citra yang ingin diidentifikasi dan memperoleh hasil kecocokan bentuk citra yang tertinggi dari sejumlah *template* yang tersimpan. Sedangkan pengguna *admin* dapat memasukkan, menghapus dan meng-*edit* citra *template* baru yang akan disimpan dalam *database*. Pada Gambar 3.8 menunjukkan bagaimana alur proses pada sistem identifikasi citra.



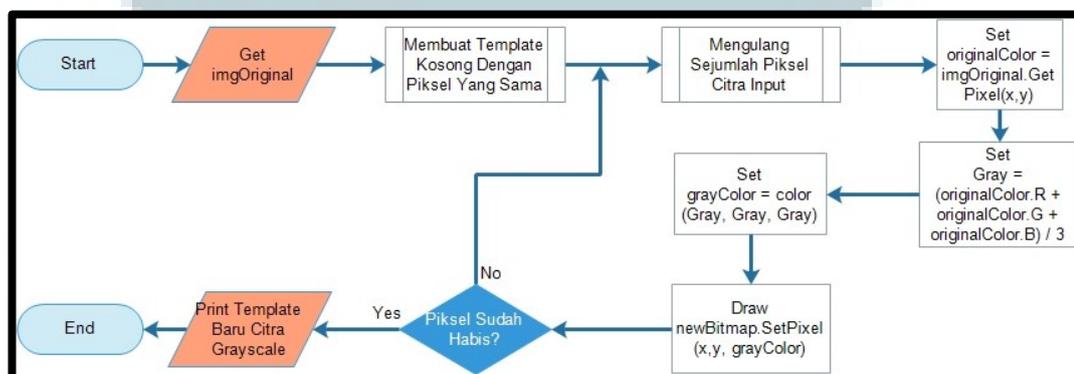
Gambar 3.8 Flowchart Diagram Sistem Identifikasi Citra

Proses awal yang dilakukan pengguna *user* maupun *admin* adalah memasukkan gambar citra ke dalam aplikasi, kemudian sistem akan melakukan proses *grayscale*, binerisasi, ekstraksi, segmentasi, *cropping* dan *resize*. Setelah proses *resize*, pengguna dapat memilih untuk langsung ke proses identifikasi atau *setup template*. Jika pengguna memilih menu *setup template*, maka sebelum masuk ke dalam proses *setup template* pengguna harus melakukan konfirmasi *password*. Jika konfirmasi *password* sudah benar maka akan masuk ke dalam proses *setup template*. Gambar 3.9 menunjukkan Flowchart Input Gambar Citra.



Gambar 3.9 *Flowchart* Input Gambar Citra

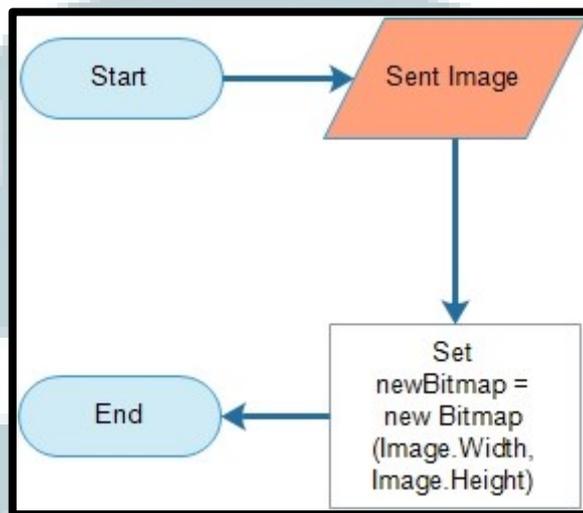
Dalam proses *input* gambar citra, pengguna *user* dan *admin* dapat memilih gambar citra yang ingin dimasukkan ke dalam sistem. Format gambar citra yang dapat dimasukkan adalah format *JPG*, *PNG*, *GIF*. Setelah pengguna memilih gambar citra, maka gambar citra yang dipilih akan ditampilkan ke dalam *interface* sistem.



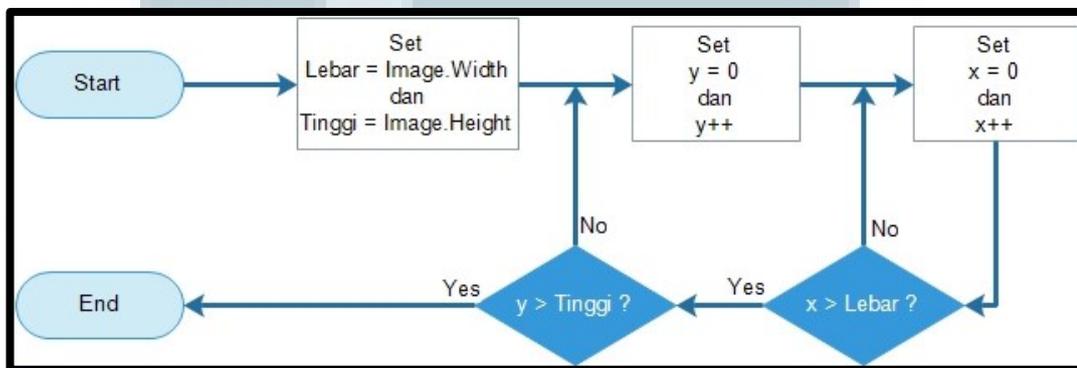
Gambar 3.10 *Flowchart* Proses *Grayscale*

Gambar 3.10 menunjukkan *Flowchart* proses *grayscale*. Proses ini bertujuan untuk memproses gambar citra masukan yang berwarna menjadi keabuan dengan cara mengambil gambar citra. Berikutnya membuat sebuah *template* kosong yang berguna sebagai *template* hasil. Selanjutnya sistem akan melakukan perhitungan nilai *grayscale* untuk warna-warna citra di setiap

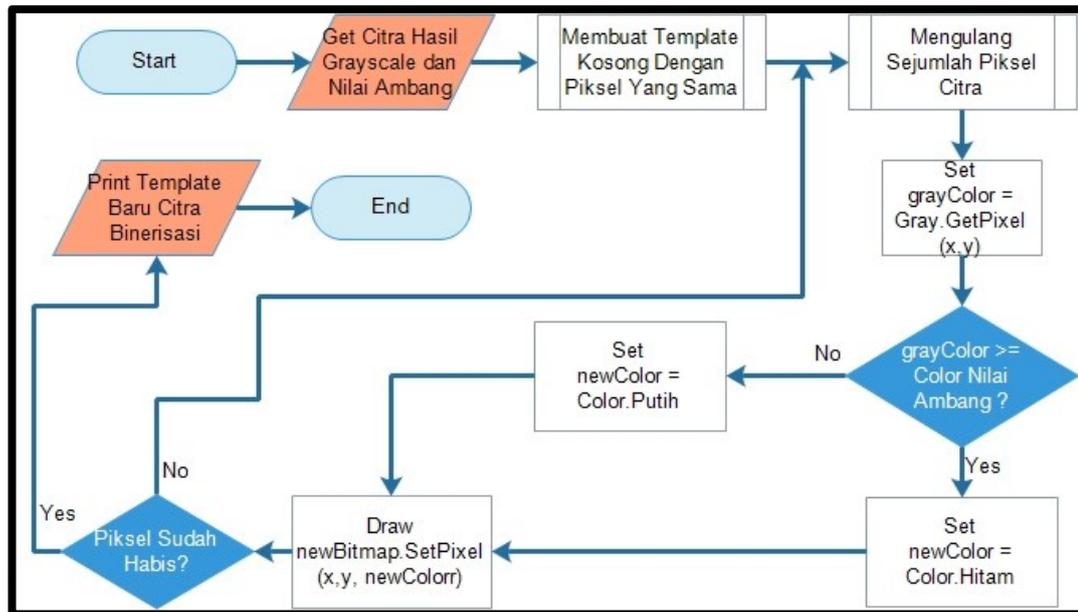
pikselnya dan langsung mencetak atau menggambar nilai tersebut pada setiap piksel dalam *template* kosong yang telah dibuat sesuai dengan ukuran citra. Jika semua piksel sudah digambar, maka sistem akan menampilkan hasil citra *grayscale* ke *user interface*.



Gambar 3.11 *Flowchart* Membuat *Template* Kosong dengan Piksel yang Sama

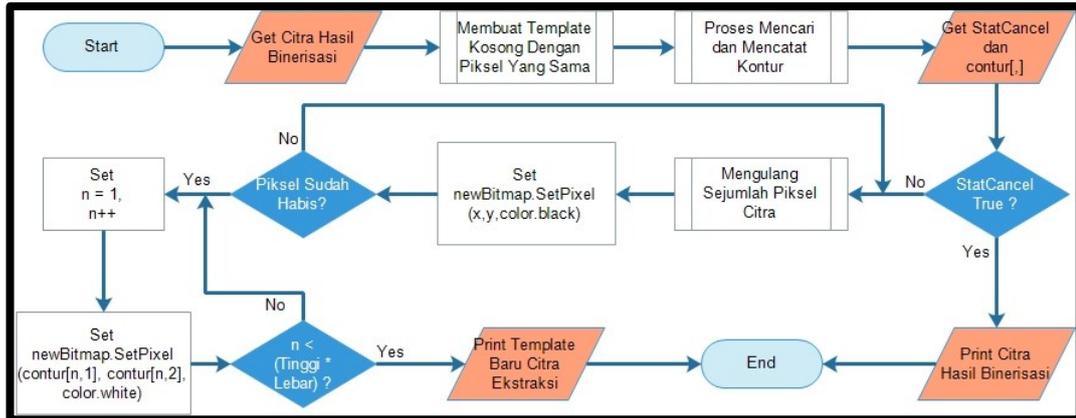


Gambar 3.12 *Flowchart* Mengulang Sejumlah Piksel Citra *Input*



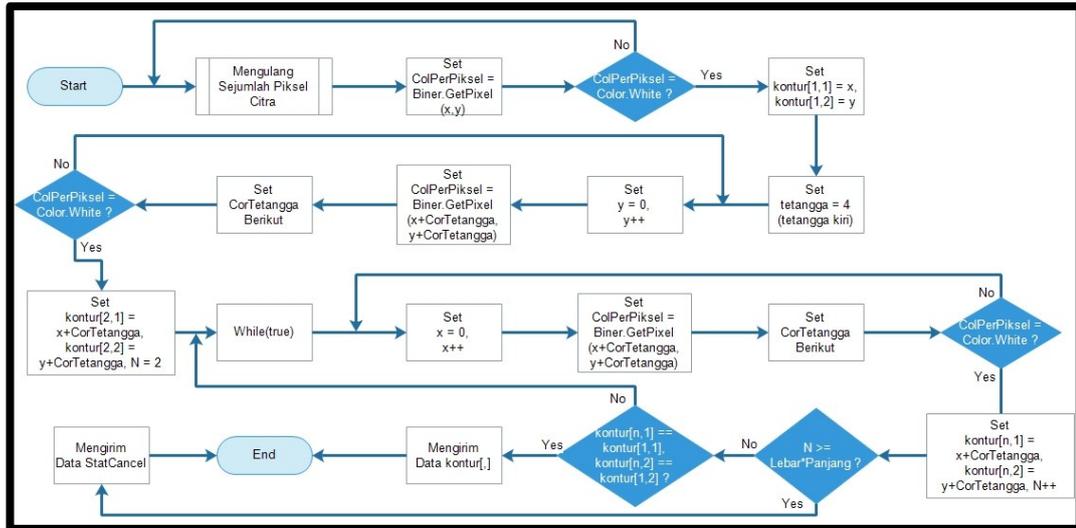
Gambar 3.13 *Flowchart* Proses Binerisasi

Gambar 3.13 menunjukkan *Flowchart* proses binerisasi, Proses ini bertujuan untuk memproses gambar citra *grayscale* yang keabuan menjadi hitam dan putih dengan cara mengambil gambar citra hasil *grayscale* dan nilai ambang yang dimasukkan pengguna. Kemudian membuat sebuah *template* kosong yang berguna sebagai *template* hasil. Selanjutnya sistem akan melakukan perhitungan nilai biner pada citra dengan membandingkan nilai *grayscale* yang ada pada setiap piksel dengan nilai ambang. Jika nilai *grayscale* lebih besar dari nilai ambang, maka nilai biner menjadi warna hitam dan jika kurang dari nilai ambang akan menjadi warna putih. Selanjutnya sistem akan langsung menggambar nilai biner tersebut pada setiap piksel dalam *template* kosong yang telah dibuat sesuai dengan ukuran citra. Jika semua piksel sudah digambar, maka sistem akan menampilkan hasil citra binerisasi ke *user interface*.

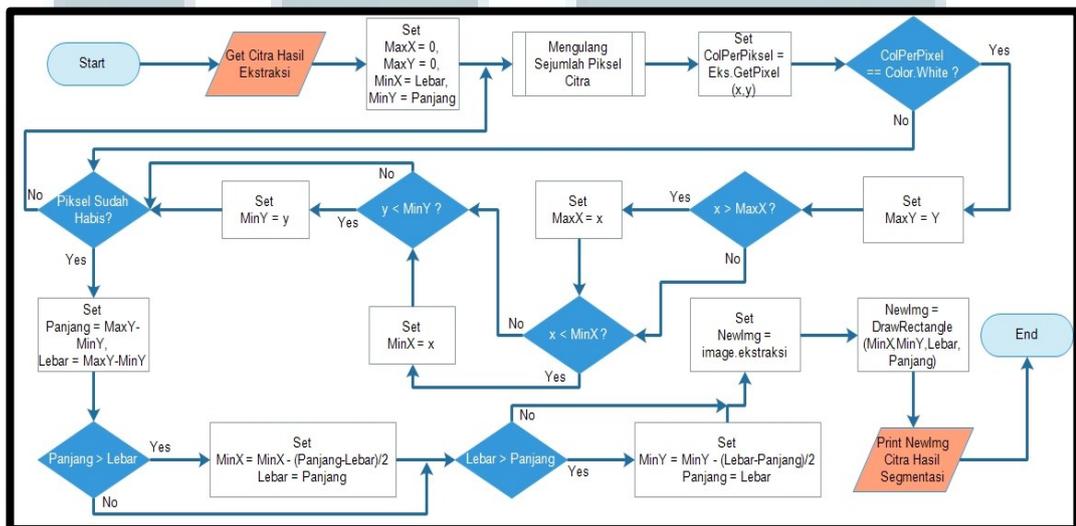


Gambar 3.14 *Flowchart* Proses Ekstraksi

Gambar 3.14 menunjukkan *Flowchart* proses ekstraksi. Proses ini bertujuan untuk memproses gambar citra binerisasi yang hitam putih dengan garis tepi yang tebal menjadi lebih tipis dengan ukuran tepi satu piksel. Pertama kali proses ini mengambil gambar citra hasil ekstraksi, setelah itu sistem akan membuat sebuah *template* kosong yang berguna sebagai *template* hasil. Kemudian pencarian kontur dengan algoritma *contour following*, yaitu dengan mencari titik yang berwarna putih pada tepi terluar objek. Selanjutnya menyimpan titik-titik kontur yang ditemukan sampai kembali ke titik awal. Jika titik kontur melebihi jumlah keliling citra maka citra tidak dapat diekstraksi, jika dapat diekstraksi maka sistem akan menggambar titik-titik kontur tersebut ke dalam *template* kosong yang telah dibuat dan ditampilkan ke *user interface*.



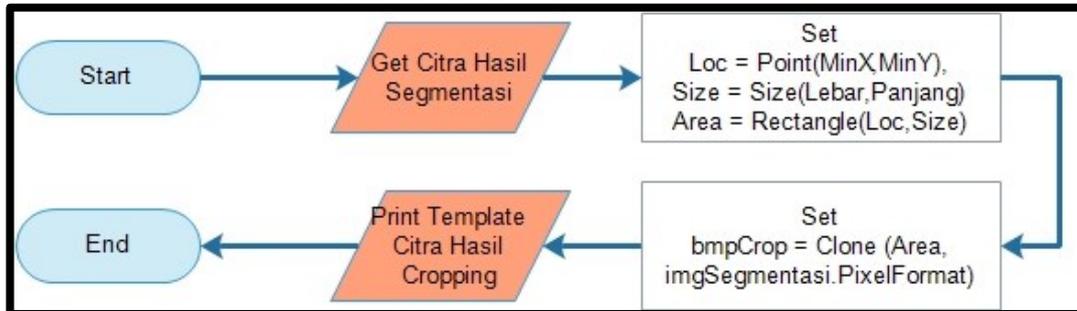
Gambar 3.15 *Flowchart* Proses Mencari dan Mencatat Kontur



Gambar 3.16 *Flowchart* Proses Segmentasi

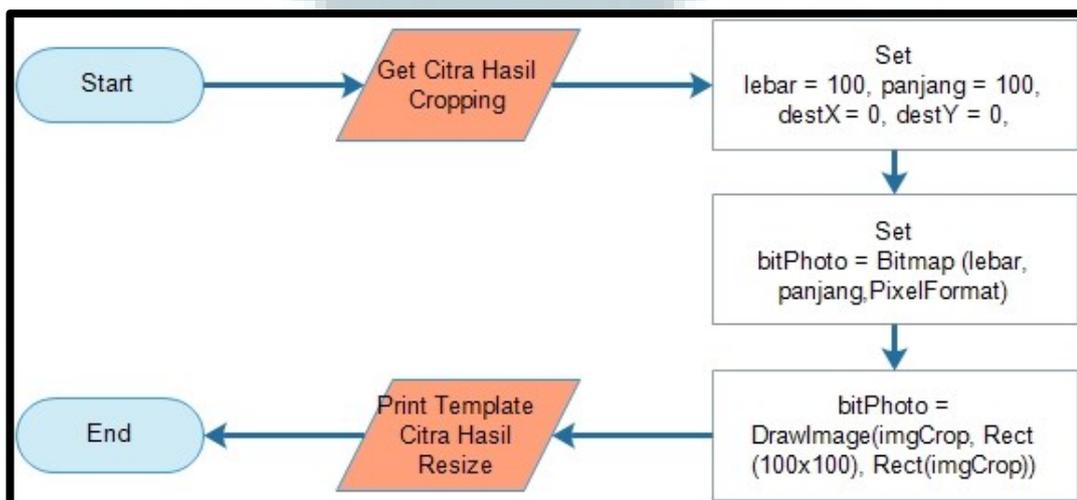
Gambar 3.16 menunjukkan *Flowchart* proses segmentasi. Proses ini bertujuan memproses gambar citra ekstraksi untuk memperoleh titik letak objek. Pertama kali, proses mengambil citra hasil ekstraksi, selanjutnya sistem akan melakukan pengulangan sebanyak jumlah piksel X dan Y. Selama pengulangan sistem mencari piksel yang berwarna putih pertama kali dan mencatat posisi awal Y, sedangkan untuk menemukan titik awal X sistem akan mencatat posisi X terkecil yang berwarna putih, sedangkan posisi akhir X dan Y dengan mencatat

titik terbesar dari X dan Y. Setelah menemukan titik awal X dan Y dan titik akhir X dan Y, sistem akan menghitung panjang dan lebar pada objek dan menggambar sebuah tanda kotak merah pada objek tersebut.



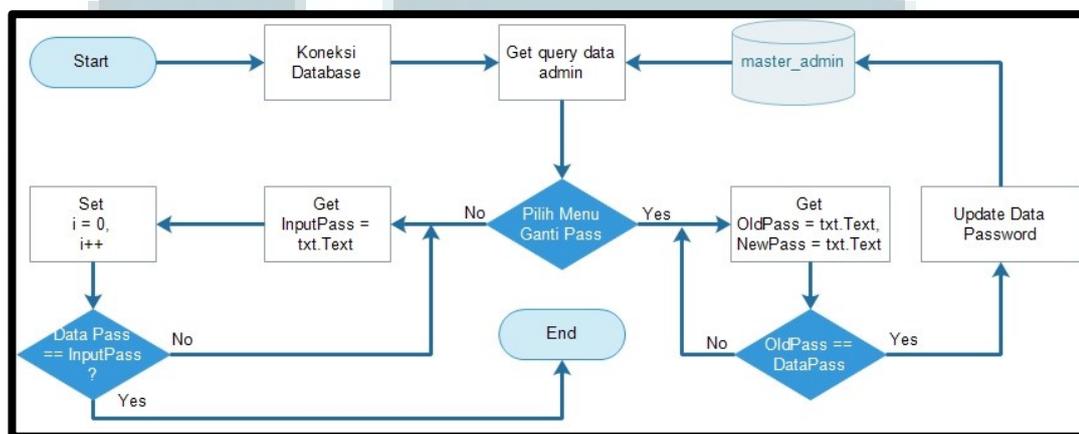
Gambar 3.17 *Flowchart* Proses *Cropping*

Gambar 3.17 menunjukkan *Flowchart* proses *cropping*, proses ini bertujuan untuk memisahkan gambar objek dengan *background*. Proses pertama sistem mengambil data gambar citra hasil segmentasi, kemudian sistem menentukan area *cropping* yang diperoleh dari hasil perhitungan proses segmentasi. Selanjutnya sistem membuat *template* kosong sesuai dengan ukuran *cropping* area dan menggambar objek citra yang sudah ditentukan areanya.



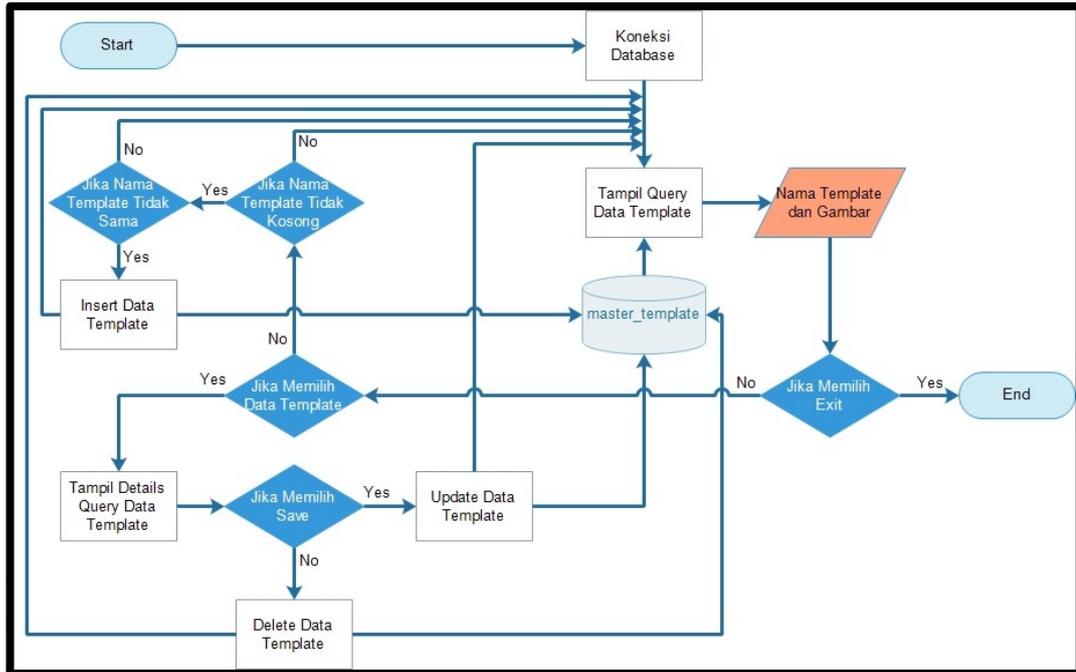
Gambar 3.18 *Flowchart* Proses *Resize* 100x100

Gambar 3.18 menunjukkan *Flowchart* proses *resize*, proses ini bertujuan untuk menyederhanakan ukuran citra yang cukup besar menjadi ukuran kecil yaitu 100 x 100 piksel, agar proses identifikasi dapat dilakukan dengan lebih cepat. Langkah awal proses ini dengan mengambil data gambar citra hasil *cropping*, kemudian sistem membuat *template* kosong berukuran 100x100 piksel, kemudian *setup* resolusi *template* sesuai dengan citra aslinya, selanjutnya menggambar citra ke dalam *template* kosong tersebut dan mencetak hasil *resize* ke *user interface*.



Gambar 3.19 *Flowchart* Konfirmasi Password

Gambar 3.19 menunjukkan *Flowchart* proses *confirm password*, proses ini bertujuan untuk menentukan hanya pengguna *admin* yang dapat melakukan proses *setup template*. Proses ini, pengguna memasukkan *password* yang sesuai dengan data pada *database*, kemudian sistem akan cek *password* apakah sudah sesuai, jika tidak maka sistem akan menampilkan pesan kesalahan, jika benar maka sistem akan berpindah ke proses *setup template*. Dalam proses ini juga pengguna dapat mengganti *password* dengan memasukkan *password* lama dan *password* baru. Jika *password* lama salah maka sistem akan menampilkan pesan kesalahan, jika benar sistem akan melakukan *update* data pada *database*.



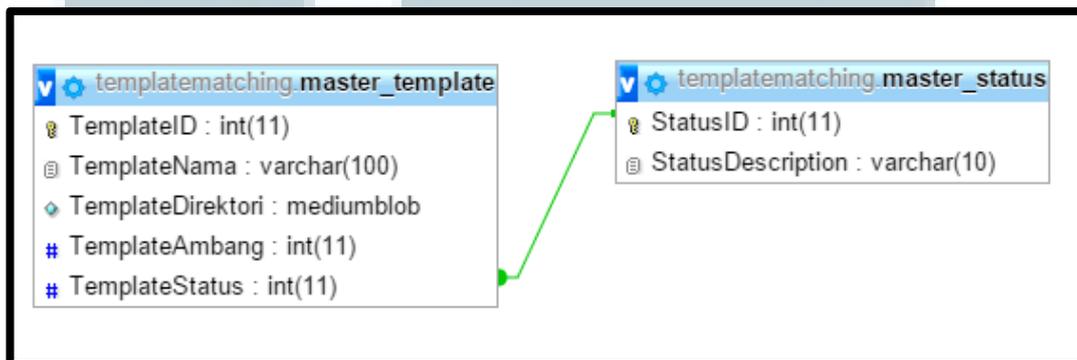
Gambar 3.20 Flowchart Setup Template

Gambar 3.20 menunjukkan *Flowchart* proses *setup template*, proses ini bertujuan untuk melakukan pengaturan data *template* di *database* dari proses *insert* data baru, *update* data dan *delete* data. Tahap awal dalam proses ini akan melakukan koneksi ke *database* lalu data ditampilkan ke *user interface*, kemudian jika *admin* tidak memilih data *template* maka *admin* akan menyimpan data *template* baru ke *database* dengan mengetikkan nama *template*. Selain itu, jika *user* memilih data *template* yang ada di *user interface*, maka sistem akan menampilkan *details* data di kolom *details* yang ada di *user interface* dan *admin* dapat melakukan *update* atau *delete* data.

Setelah menemukan nilai atas dan nilai bawah, maka proses perhitungan nilai kecocokan dengan membagi nilai atas dibagi nilai bawah kemudian dikali seratus untuk memperoleh nilai persen pada *template* yang digunakan sebagai perbandingan. Proses ini dilakukan berulang hingga jumlah data pada *database* habis. Setelah menemukan hasil nilai kecocokan pada masing-masing data *template*, maka sistem akan melakukan pencarian empat nilai kecocokan yang tertinggi dan ditampilkan ke *user interface*.

3.2.3 Entity Relationship Diagram

Entity relationship diagram pada sistem ini digambarkan pada Gambar 3.22.



Gambar 3.22 *Entity Relationship Diagram*

Tabel yang saling berhubungan atau yang memiliki relasi adalah tabel *master_template* dengan tabel *master_status*. Pada tabel *master_status* sebagai *primary key* dan salah satu atribut pada tabel *master_template* menjadi *foreign key*.

3.2.4 Struktur Tabel

Struktur tabel yang digunakan dalam sistem identifikasi ini adalah.

1. Nama Tabel : master_status
Fungsi : Tabel ini bertujuan untuk menentukan status pada data *template*

Primary key : StatusID

Foreign key : -

Tabel 3.1 Struktur Tabel Status

No	Nama Kolom	Tipe Data	Keterangan
1	StatusID	int(11)	ID Status
2	StatusDescription	varchar(10)	Keterangan Status

2. Nama Tabel : master_template
Fungsi : Tabel ini digunakan sebagai penyimpanan data *template* yang menyimpan data *template* dan gambar

Primary key : TemplateID

Foreign key : TemplateStatus

Tabel 3.2 Struktur Tabel Data *Template*

No	Nama Kolom	Tipe Data	Keterangan
1	TemplateID	int(11)	ID Template
2	TemplateNama	varchar(100)	Nama Template
3	TemplateDirektori	mediumblob	Penyimpanan Gambar
4	TemplateAmbang	int(11)	Nilai batas ambang template
5	TemplateStatus	int(11)	Status Template

3. Nama Tabel : master_admin
Fungsi : Tabel ini digunakan sebagai data *password* pada sistem
Primary key : AdminID

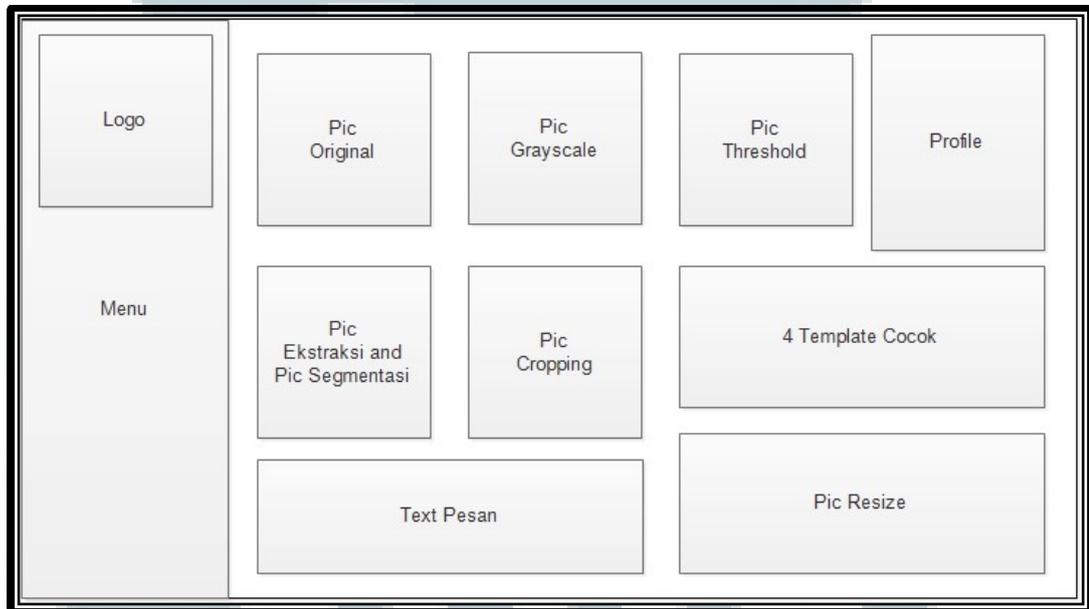
Foreign key :-

Tabel 3.3 Struktur Tabel Data *Password*

No	Nama Kolom	Tipe Data	Keterangan
1	AdminID	int(11)	ID Password
2	AdminPass	varchar(20)	Password

3.2.5 Rancangan Antar Muka

Rancangan *user interface* dari aplikasi ini dapat dilihat pada Gambar 3.23. Pada Gambar 3.23 *user interface* dirancang dengan teknik simulasi sehingga pengguna dapat mengetahui langkah dan proses apa saja yang dilakukan oleh sistem dalam teknik *Image Processing* sebelum masuk ke tahap identifikasi dengan menggunakan metode *image processing*. *User interface* ini menampilkan hasil dari proses *grayscale*, binerisasi, ekstraksi, segmentasi, *cropping* dan *resize* yang *details*-nya akan dibahas setelah Gambar 3.23.

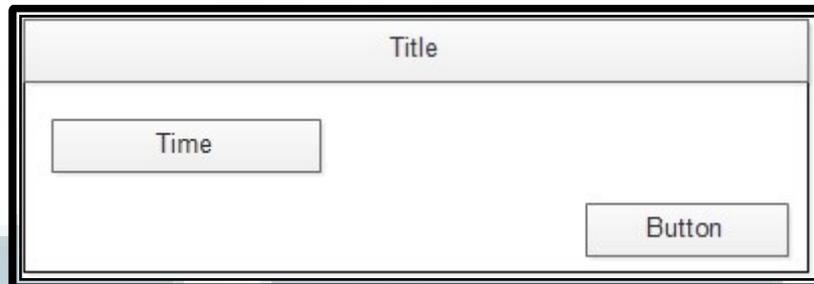


Gambar 3.23 Rancangan *User Interface*

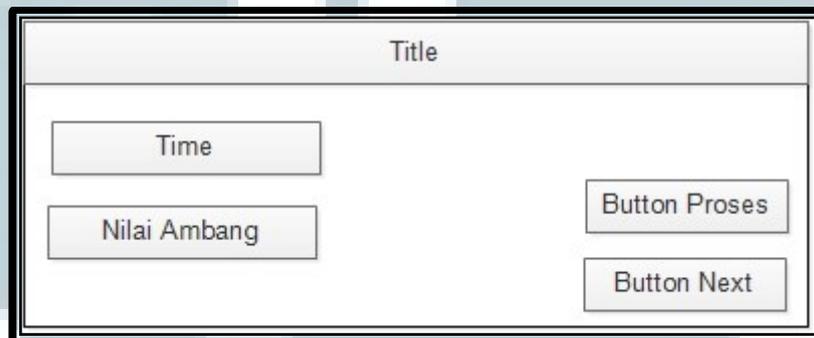
1. **Menu**
User interface ini berisi logo UMN dan juga tombol-tombol *open file*, *identifikasi*, dan *setup template*.
2. **Pic Original**
User interface ini menampilkan gambar citra asli yang dipilih oleh pengguna.
3. **Pic Grayscale**
User interface ini menampilkan gambar citra hasil proses *grayscale*.
4. **Pic Thresholding**
User interface ini menampilkan gambar citra hasil proses *thresholding*.
5. **Pic Ekstraksi dan Pic Segmentasi**
User interface ini menampilkan gambar citra hasil proses ekstraksi dan juga gambar citra hasil proses segmentasi.
6. **Pic Cropping**
User interface ini menampilkan gambar citra hasil proses *cropping*.
7. **Empat Template Cocok**
User interface ini menampilkan empat gambar citra *template* yang memiliki nilai kecocokan tertinggi dan menampilkan nilai hasil persentasi kecocokan dan nama *template*.
8. **Pic Resize**
User interface ini menampilkan gambar citra masukan hasil proses *resize*.
9. **Text Pesan**
User interface ini menampilkan pesan *text* hasil dari setiap proses.

10. Profile

User interface ini menampilkan foto dan keterangan *profile* pembuat sistem.

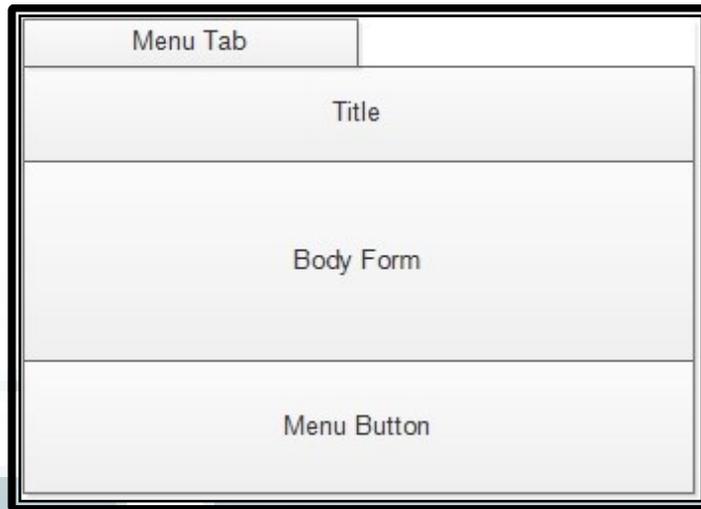


Gambar 3.24 Rancangan *User Interface Notif Proses 1*



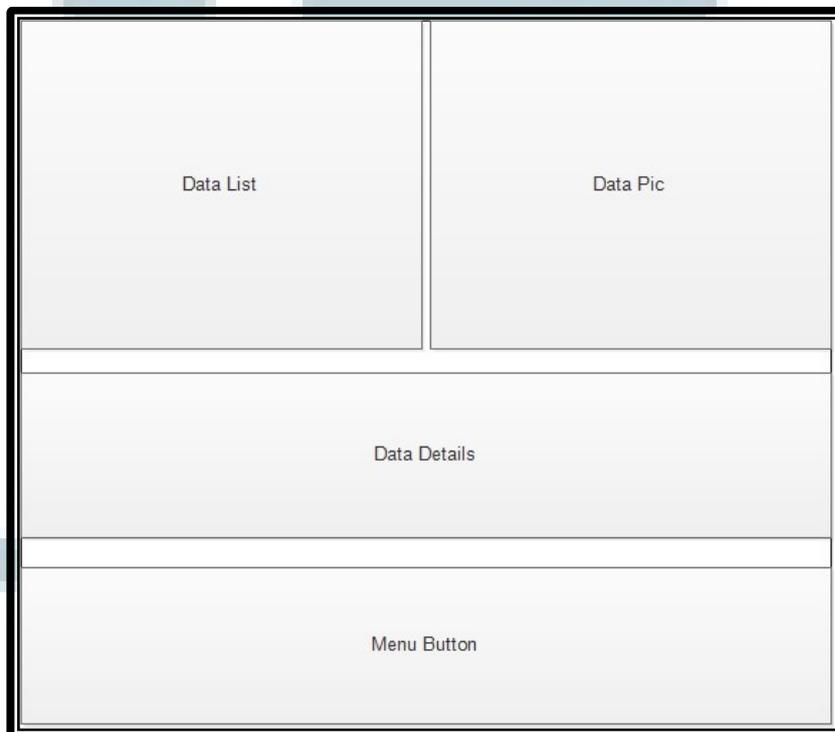
Gambar 3.25 Rancangan *User Interface Notif Proses 2*

Setiap akhir proses yang dilakukan sistem memiliki notifikasi yang akan ditampilkan kepada pengguna. Gambar 3.24 menunjukkan rancangan *user interface* notifikasi proses *grayscale*, ekstraksi, segmentasi, *cropping*, *resize* dan identifikasi, sedangkan pada Gambar 3.25 menunjukkan rancangan *user interface* notifikasi proses *thresholding*.



Gambar 3.26 Rancangan *User Interface* Konfirmasi *Password*

Gambar 3.26 menunjukkan rancangan *user interface* konfirmasi *Password* yang akan ditampilkan ke pengguna *admin*. Rancangan *user interface* ini memiliki bagian *body* yang berisikan *form*, isi *form* dapat berubah jika pengguna memilih menu *tab* di bagian atas. Sedangkan di bagian *button* terdapat tombol *ok* dan *cancel*.



Gambar 3.27 Rancangan *User Interface* *Setup Template*

Gambar 3.27 menunjukkan rancangan *user interface setup template* yang akan ditampilkan ke pengguna untuk proses *setup template*. Bagian *user interface* ini terdiri dari beberapa bagian diantaranya adalah.

1. *Data List*

User interface ini menampilkan kumpulan *list data template* berupa *list view*.

2. *Data Pic*

User interface ini menampilkan gambar citra data yang dipilih dari *list*.

3. *Data Details*

User interface ini menampilkan *details* nama, status dan ID data.

4. *Menu Button*

User interface ini menampilkan tombol *add new*, *save change* dan *delete*.

U
M
M
N