BAB III
PERANCANGAN SISTEM

3.1 Analisis Masalah

Permasalahan yang diangkat pada penelitian ini ialah menganalisis kondisi suatu gambar teks yang dapat diolah dengan baik oleh program aplikasi sehingga menghasilkan hasil pengenalan teks Arab yang optimal. Analisis tersebut dilakukan berdasarkan pengambilan gambar yang mungkin terjadi pada saat pengguna aplikasi melakukan pengambilan gambar teks Arab secara langsung di lokasi keberadaan teks tersebut.

3.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional terdiri dari perangkat keras, perangkat lunak dan pengguna.

3.2.1 Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras untuk menjalankan aplikasi ini ialah *smartphone* Android yang memiliki beberapa spesifikasi utama sebagai berikut:

1. Sistem operasi Android v2.2 (Froyo).
2. Kamera 3.15MP (2048x1536 pixels), untuk keperluan pengambilan gambar teks yang dilakukan secara langsung.
3. Tampilan layar sentuh, untuk keperluan pemotongan gambar teks yang dilakukan secara manual.
4. Media penyimpanan eksternal, untuk keperluan menyimpan berbagai data yang dihasilkan oleh program aplikasi.
5. Jaringan 2G/3G, untuk menghubungkan program aplikasi dengan layanan OCR dan layanan penerjemah.

3.2.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dipakai dalam pengembangan program aplikasi, diantaranya:

1. Sistem Operasi Ubuntu 12.04.
2. Java Development Kit versi 6, untuk keperluan kompilasi kode-kode program.
3. Eclipse IDE 3.7, sebagai perangkat lunak yang digunakan untuk membangun aplikasi android.
4. Android SDK, untuk pengembangan aplikasi Android
5. Android Development Tool, sebagai plugin untuk mengintegrasikan Android SDK pada eclipse.
6. ABBYY Clud OCR SDK for Android, sebagai antarmuka pemrograman aplikasi yang menyediakan berbagai fungsi untuk melakukan konfigurasi parameter ke layanan ABBYY Cloud OCR serta prosedur pengiriman dan pengunduhan dokumen ke layanan tersebut.
7. Microsoft translator java API, sebagai antarmuka pemrograman aplikasi yang menyediakan berbagai fungsi untuk menentukan konfigurasi parameter penerjemahan serta menghubungkan program aplikasi dengan layanan Microsoft Translator.
8. Better Arabic Resharper, merupakan sebuah pustaka yang berfungsi untuk merapihkan penulisan teks Arab pada *smartphone* Android, khususnya untuk *smartphone* Android dengan versi di bawah 4.0.

3.2.3 Analisis Pengguna

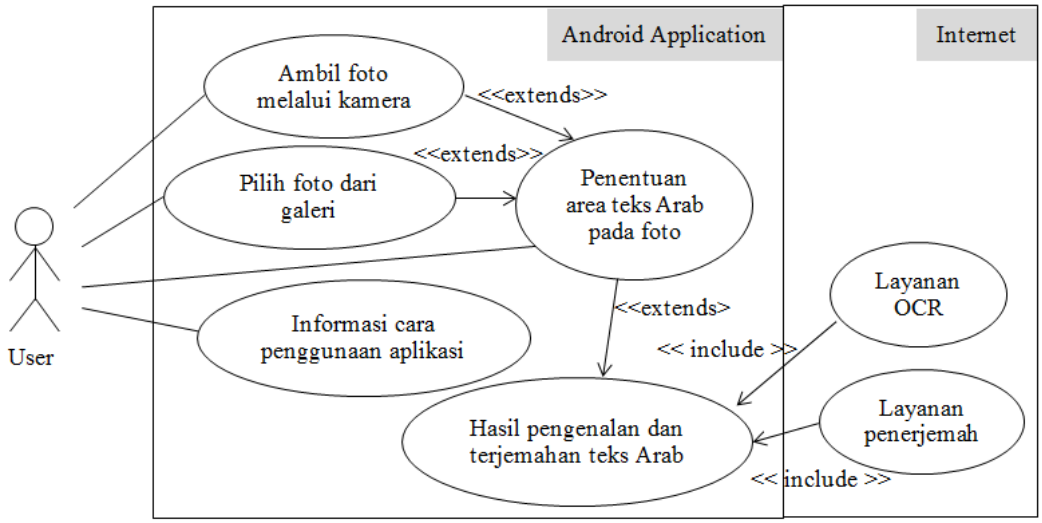
Aplikasi yang akan dibangun ditujukan bagi pengguna *smartphone* Android di Indonesia, khususnya bagi warga Indonesia yang sedang berkunjung ke negara Arab dan kurang begitu memahami bahasa Arab. Aplikasi ini juga dapat difungsikan sebagai kamus untuk mengenal berbagai macam kata yang ditulis menggunakan bahasa Arab, sehingga tidak menutup kemungkinan dapat dipakai oleh berbagai kalangan.

3.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional meliputi beberapa diagram UML, diantaranya: diagram *use case*, diagram *activity*, diagram *class* dan diagram *sequence*. Diagram tersebut dipergunakan sebagai desain rancangan untuk mempermudah proses pengembangan aplikasi khususnya untuk aplikasi yang dikembangkan menggunakan pemrograman berorientasi objek.

3.3.1 Diagram Use Case

Pada sistem aplikasi yang dirancang terdapat lima buah *uses case* dan *user* sebagai aktor tunggal yang berperan dalam sistem. Kata kunci *extends* menjelaskan bahwa *use case* tersebut merupakan perluasan dari *use case* sebelumnya sedangkan kata kunci *include* menjelaskan bahwa *use case* tersebut memerlukan fungsionalitas dari *use case* lain di luar sistem.



Gambar 3.1. Diagram *Use Case*

Penjelasan diagram *use case* diatas menggunakan tabel spesifikasi naratif sebagai berikut:

Tabel 3.1. Spesifikasi naratif untuk *use case* ambil foto melalui kamera

Nama *Use Case*	**Ambil foto melalui kamera**
Deskripsi Singkat	*Use case* ini memungkinkan pengguna untuk melakukan pengambilan foto secara langsung menggunakan kamera.
Aktor	Pengguna aplikasi.
Kondisi awal	Aplikasi telah dijalankan.
Tindakan Utama	Pengguna mengambil foto dengan cara memilih menu Ambil Foto yang terdapat di aplikasi.
Pasca Kondisi	Foto ditampilkan di layar.

Tabel 3.2. Spesifikasi naratif untuk *use case* pilih foto dari galeri

Nama *Use Case*	**Pilih foto dari galeri**
Deskripsi Singkat	*Use case* ini memungkinkan pengguna untuk memilih foto yang sudah tersimpan di *external memory* melalui galeri.
Aktor	Pengguna aplikasi.
Pra Kondisi	Aplikasi telah dijalankan.
Tindakan Utama	Pengguna memilih foto yang sudah tersimpan di *external memory* dengan cara memilih menu Pilih Foto dari Gallery yang terdapat di aplikasi.
Pasca Kondisi	Foto ditampilkan di layar.

Tabel 3.3. Spesifikasi naratif untuk *use case* penentuan area teks Arab pada foto

Nama *Use Case*	**Penentuan area teks Arab pada foto**
Deskripsi Singkat	*Use case* ini memungkinkan pengguna untuk melakukan penyeleksian area teks pada foto untuk menghasilkan hasil potongan gambar teks yang lebih kecil.
Aktor	Pengguna aplikasi.
Pra Kondisi	Salah satu *use case* sebelumnya telah dilakukan.
Tindakan Utama	Pengguna menyeleksi area teks pada citra yang ingin diproses oleh aplikasi dengan memanfaatkan fitur layar sentuh pada perangkat Android.
Pasca Kondisi	Dialog ditampilkan di layar.

Tabel 3.4. Spesifikasi naratif untuk *use case* hasil pengenalan dan penerjemahan teks Arab

Nama *Use Case*	**Hasil pengenalan dan penerjemahan teks Arab**
Deskripsi Singkat	*Use case* ini merupakan perluasan dari *use* case penentuan area teks Arab pada foto. *Use case* akan menampilkan hasil pengenalan dan penerjemahan teks Arab dengan menyertakan fungsionalitas dua buah *use case* dari luar sistem
Aktor	-
Pra Kondisi	*Use case* penentuan teks Arab pada foto telah dilakukan
Tindakan Utama	-
Pasca Kondisi	Hasil pengenalan dan penerjemahan teks Arab ditampilkan di layar

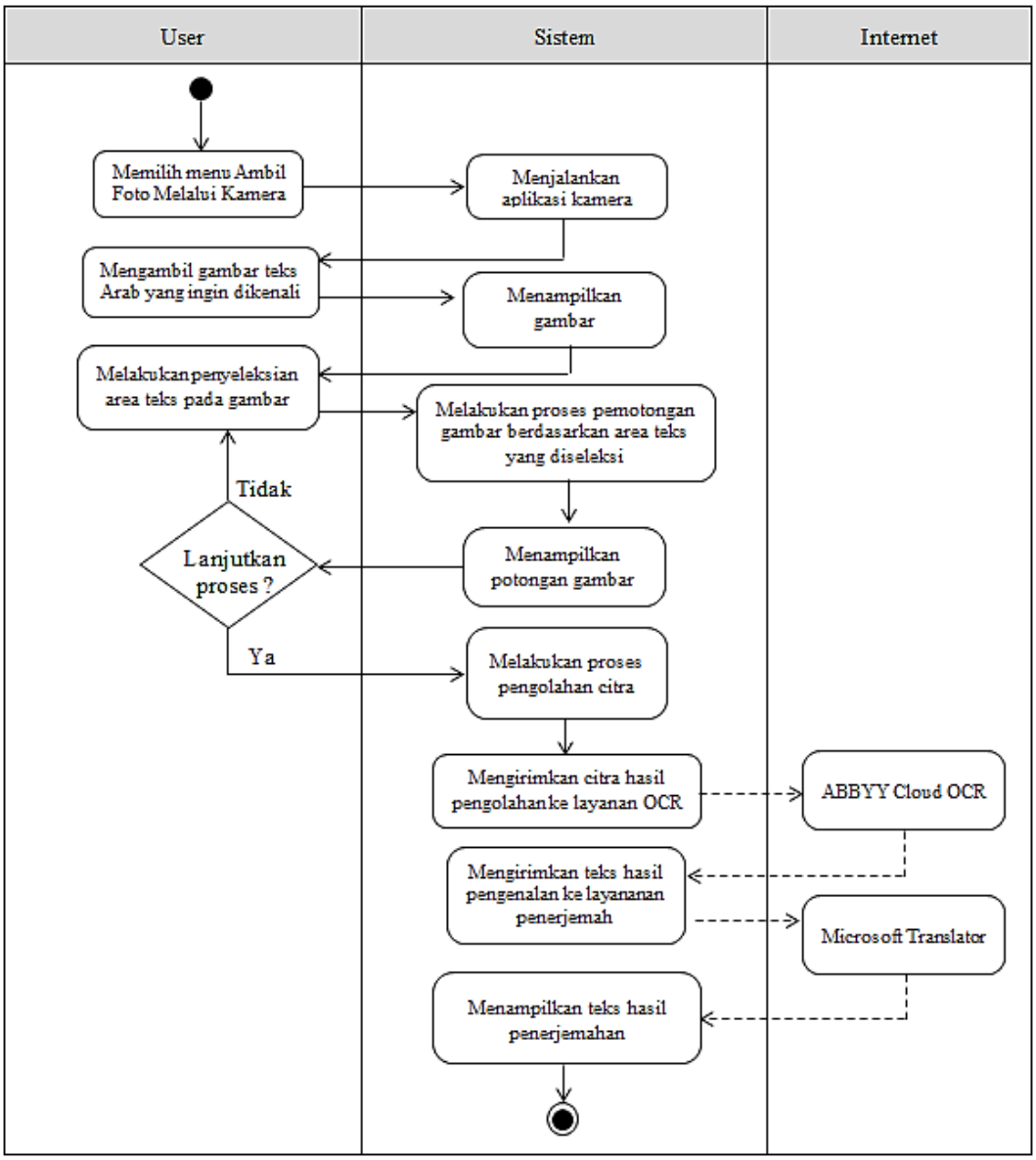
Tabel 3.5. Spesifikasi naratif untuk *use case* informasi cara pengguaan aplikasi

Nama *Use Case*	**Informasi cara penggunaan Aplikasi**
Deskripsi Singkat	*Use case* ini memungkinkan pengguna untuk mendapatkan informasi mengenai cara penggunaan aplikasi.
Aktor	Pengguna aplikasi.
Pra Kondisi	Aplikasi telah dijalankan.
Tindakan Utama	Pengguna memilih menu Bantuan yang terdapat di aplikasi.
Pasca Kondisi	Informasi mengenai cara penggunaan aplikasi ditampilkan di layar.

3.3.2 Diagram Activity

Diagram *activity* digunakan untuk menggambarkan alur rangkaian proses yang terjadi di dalam sistem aplikasi serta menjelaskan bagaimana sebuah *use case* terjadi atau dipicu dari *use case* yang lain. Masing-masing diagram aktifitas dapat dilihat pada gambar 3.2 sampai dengan 3.4.

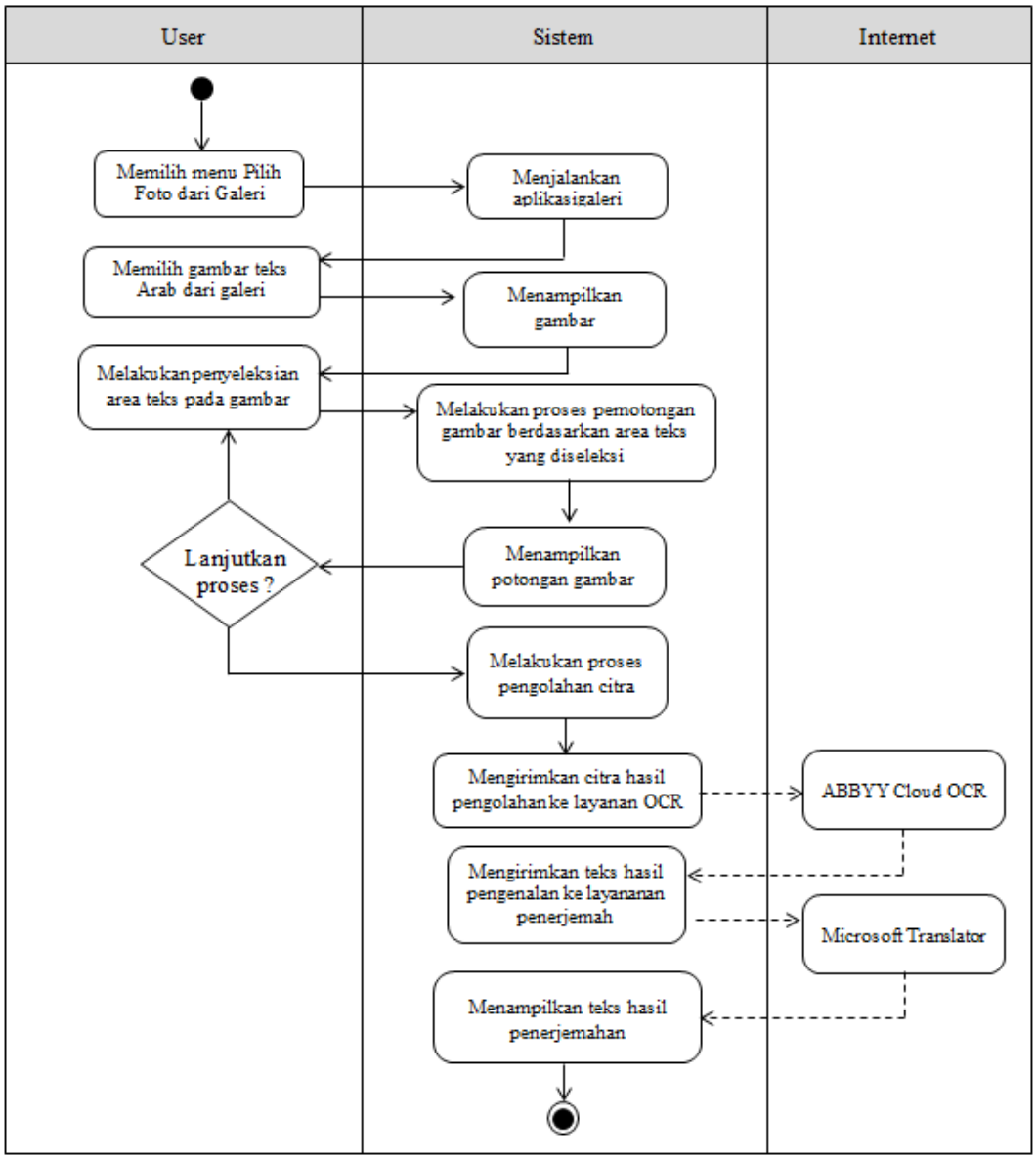
Gambar 3.2 memperlihatkan aktifitas yang terjadi antara *user* dan sistem pada saat *user* memilih menu ambil foto melalui kamera. Gambar 3.3 memperlihatkan aktifitas yang terjadi antara *user* dan sistem pada saat *user* memilih menu pilih foto dari galeri. Kedua diagram tersebut memiliki banyak kesamaan aktifitas, baik yang dilakukan *user* maupun sistem, yang membedakan keduanya hanya pada saat penentuan gambar teks Arab yang ingin dikenali. Untuk itu, penjelasan mengenai gambar 3.3 akan mengacu pada penjelasan gambar 3.2.



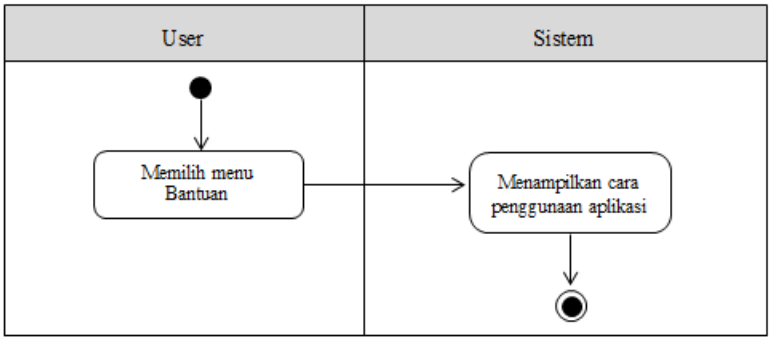
Gambar 3.2. Diagram *Activity* ambil foto melalui kamera

Diagram *activity* di atas menggambarkan aktifitas yang dilakukan *user* pada saat memilih menu ambil foto melalui kamera sampai dengan sistem menampilkan teks hasil penerjemahan. Penjelasan mengenali aktifitas yang terjadi:

1. Aktifitas dimulai dari *user* memilih menu ambil foto melalui kamera;
2. Sistem aplikasi selanjutnya akan memanggil aplikasi kamera yang kemudian memicu *use case* ambil foto melalui kamera.
3. Foto hasil pengambilan tersebut akan ditampilkan oleh sistem yang kemudian memicu *use case* berikutnya yaitu penyeleksian area teks pada foto.
4. Sistem kemudian melakukan proses pemotongan gambar berdasarkan area teks yang diseleksi.
5. Pada saat sistem melakukan aktifitas menampilkan potongan gambar, sistem melakukan pengkondisian agar user dapat melakukan kembali aktifitas penyeleksian area teks pada gambar jika hasil potongan gambar tidak sesuai dengan yang diinginkan.
6. Jika user memilih untuk melanjutkan proses, maka sistem akan melakukan prosedur pengolahan citra.
7. Gambar hasil pengolahan selanjutnya akan dikirimkan ke layanan ABBYY Cloud OCR untuk dilakukan proses pengenalan teks.
8. Proses selanjutnya, hasil pengenalan tersebut akan dikirimkan ke layanan Microsoft Translator.
9. Respon yang diterima dari layanan Microsoft Translator selanjutnya akan ditampilkan di layar.



Gambar 3.3. Diagram *Activity* memilih foto dari galeri

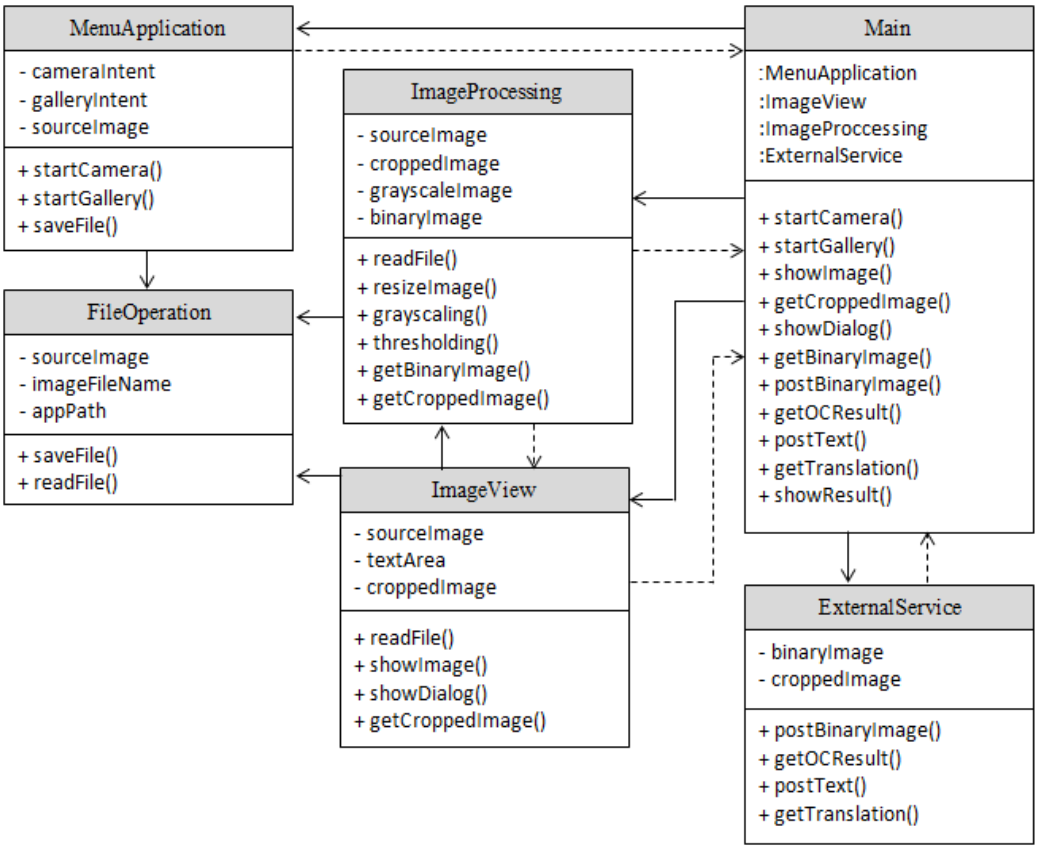


Gambar 3.4. Diagram *Activity* lihat cara penggunaan aplikasi

Diagram *activity* di atas menggambarkan aktifitas yang dilakukan *user* untuk memperoleh informasi mengenai cara penggunaan aplikasi.

3.3.3 Diagram *Class*

Diagram *Class* digunakan untuk menggambarkan secara abstrak struktur dari aplikasi yang akan dibangun, *class-class* yang terlibat, serta hubungan antar *class* untuk saling berkomunikasi satu sama lain. Struktur *class* dari aplikasi yang akan dibangun dapat dilihat pada gambar 3.5.



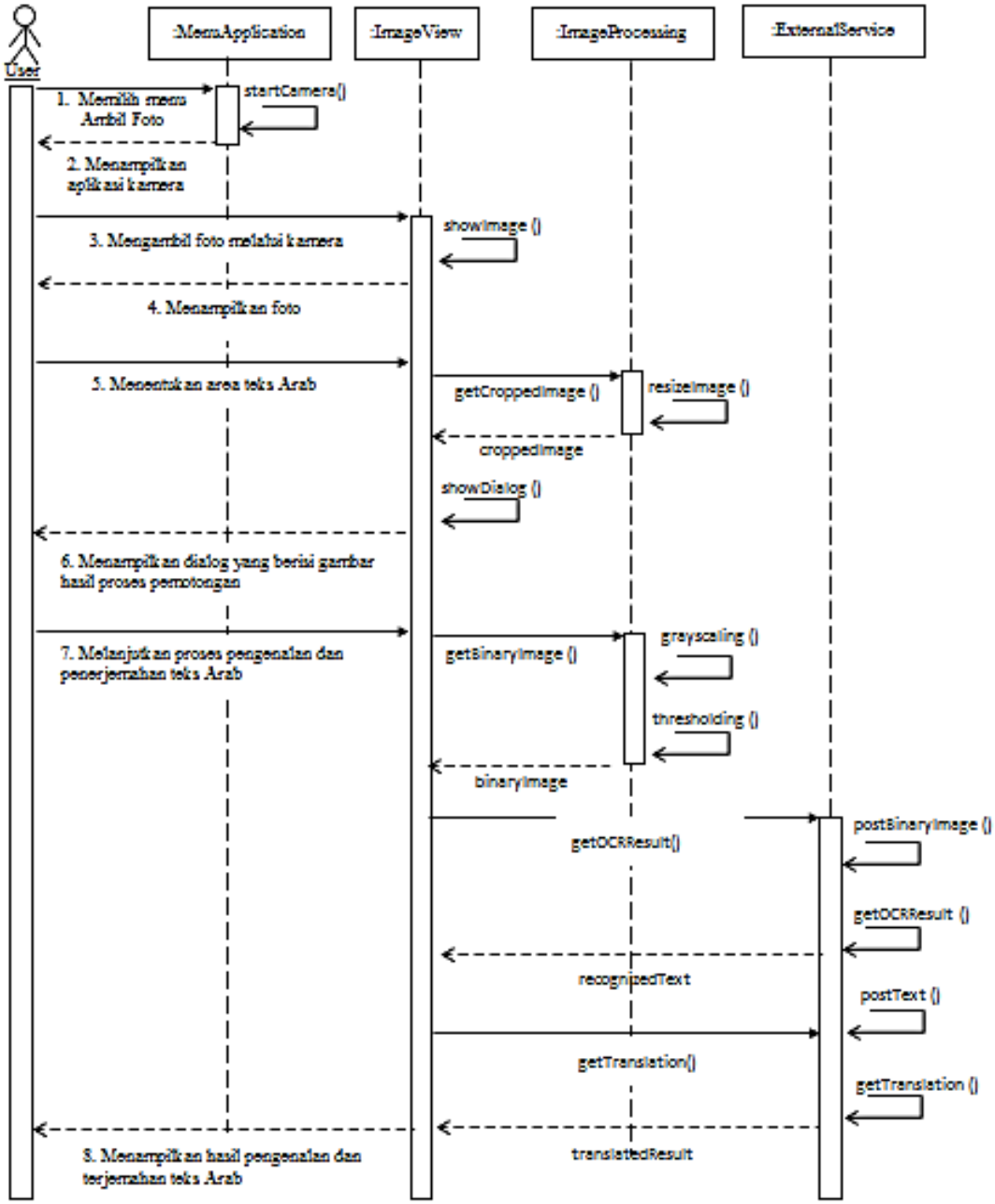
Gambar 3.5. Diagram *Class*

Aplikasi Android yang akan dibangun terdiri dari enam buah *class*, dengan *class* Main sebagai *class* utamanya. Pada *class* Main terdapat empat objek yang merupakan interpretasi dari *class* lain yang terlibat. Pada diagram *class* diatas terlihat dua jenis garis yang menghubungkan antara *class* yang satu dengan *class*

lain. Garis penuh menunjukkan pertukaran pesan antar objek, sedangkan garis putus-putus menunjukkan respon dari objek yang dipanggil.

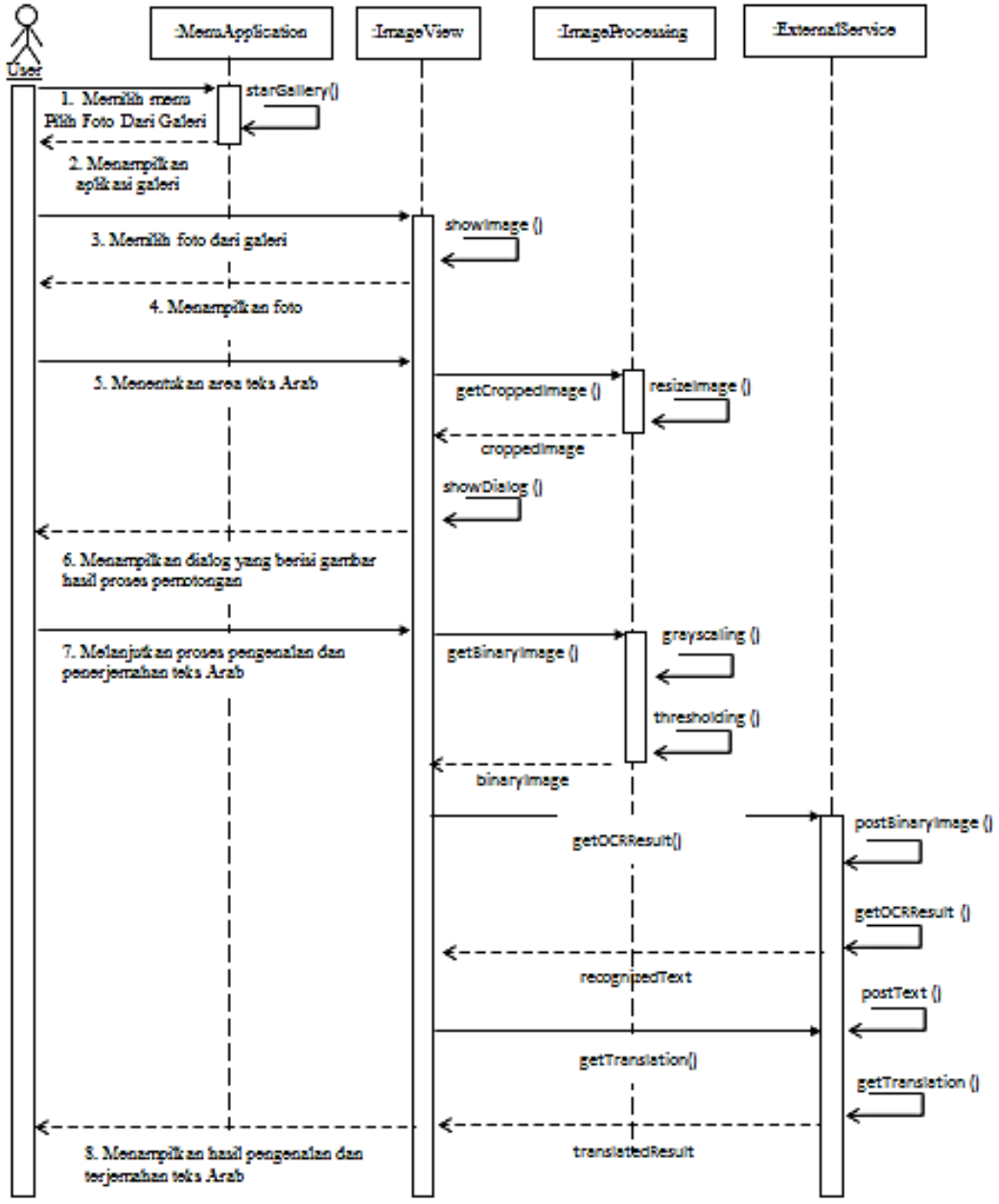
3.3.4 Diagram *Sequence*

Diagram *sequence* menjabarkan urutan aktifitas yang terjadi di dalam sistem. Diagram ini menunjukkan pengiriman pesan-pesan yang melewati objek yang terlibat di dalam sistem serta tahapan langkah yang harus dilalui untuk mencapai *use case* tertentu. Diagram *sequence* dari aplikasi yang akan dibuat, dapat dilihat pada gambar 3.6. sampai dengan gambar 3.8



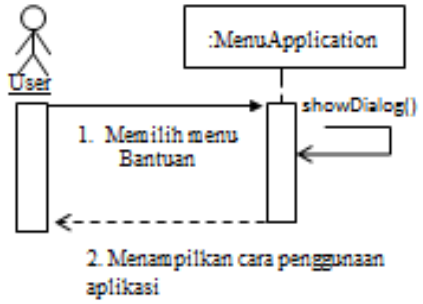
Gambar 3.6. Diagram *Sequence* ambil foto dari kamera

Pada diagram *sequence* diatas, terdapat delapan aktifitas yang terjadi hingga *user* memperoleh hasil pengenalan citra teks Arab. *User* berinteraksi secara langsung dengan objek MenuAplication dan ImageView, sedangkan dua objek lain melakukan proses secara tersembunyi dari *user*. Pengiriman pesan antar objek lebih banyak melibatkan objek ImageView yang merupakan antarmuka *user* dengan program aplikasi.



Gambar 3.7. Diagram *Sequence* pilih foto dari galeri

Pada diagram *sequence* diatas, sebagian besar aktifitas yang terjadi sama dengan diagram *sequence* pada gambar 3.6. Perbedaan hanya pada aktifitas ke-1 sampai dengan aktifitas ke-3. Untuk itu penjelasan mengenai diagram ini akan mengacu pada penjelasan gambar 3.6.



Gambar 3.8. Diagram *Sequence* cara penggunaan aplikasi

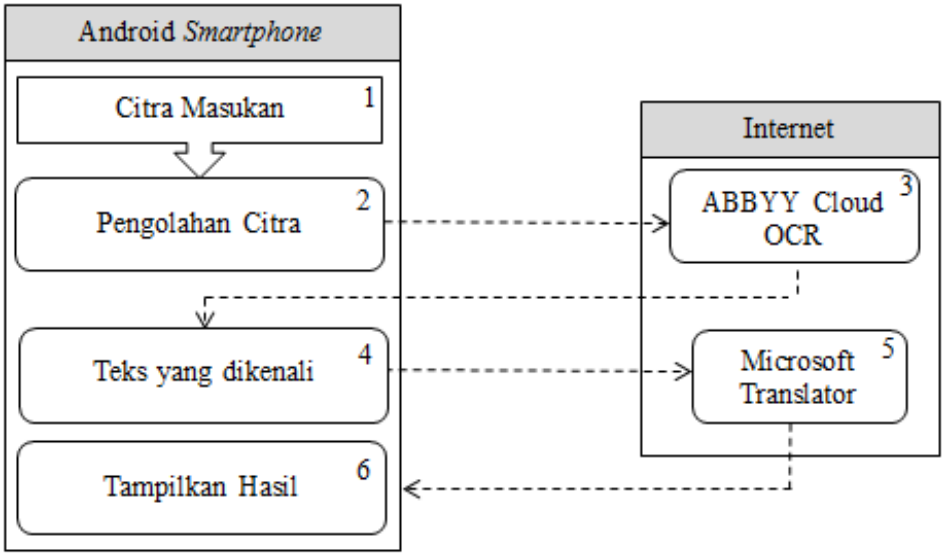
Pada diagram *sequence* di atas terdapat dua aktifitas yang terjadi hingga *user* memperoleh informasi mengenai cara penggunaan aplikasi. Objek yang terlibat ialah MenuAplication yang merupakan antarmuka *user* dengan program aplikasi pada saat pertamakali aplikasi dijalankan.

3.4 Perancangan Sistem

Sistem aplikasi android yang akan dibangun memiliki fungsi utama untuk melakukan proses pengolahan citra. Sedangkan untuk proses pengenalan teks dan penerjemahan akan memanfaatkan dua layanan web yang sudah ada, yaitu ABBYY Cloud OCR dan Microsoft Translator. *Input* citra teks yang dapat dikenali oleh layanan ABBYY Cloud OCR harus berupa citra teks yang berwarna hitam dengan latar belakang berwarna putih. Oleh karena itu, tujuan dari pengolahan citra yang dikerjakan oleh program aplikasi ialah untuk menghasilkan citra teks yang bisa dikenali oleh layanan OCR tersebut.

3.4.1 Rancangan Sistem Aplikasi Android

Secara umum, rancangan sistem aplikasi yang akan dibangun dapat dilihat pada gambar 3.9.



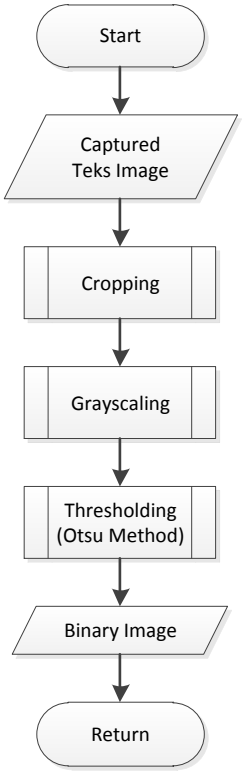
Gambar 3.9. Diagram blok rancangan sistem aplikasi Android

Penjelasan rancangan sistem aplikasi Android:

1. Citra masukan merupakan citra teks Arab yang akan diolah oleh program aplikasi. Citra ini diperoleh dari hasil pengambilan foto secara langsung atau bisa juga berasal dari *external memory* pada *smartphone* Android.
2. Pengolahan citra pada program aplikasi Android akan mengubah citra masukan menjadi sebuah citra teks berwarna hitam dengan latar belakang berwarna putih. Citra hasil pengolahan tersebut kemudian akan dikirimkan ke layanan ABBYY Cloud OCR untuk dilakukan proses pengenalan teks.
3. Setelah teks pada citra berhasil dikenali, hasil pengenalan tersebut kemudian akan dikirimkan kembali ke program aplikasi.
4. Hasil pengenalan dari layanan OCR berupa teks Arab selanjutnya akan dikirimkan ke layanan Microsoft Translator untuk diterjemahkan ke dalam bahasa Indonesia.
5. Hasil penerjemahan dari Microsoft Translator selanjutnya akan dikirimkan kembali ke program aplikasi.
6. Tampilan akhir pada program aplikasi berupa hasil pengenalan teks Arab dan terjemahannya dalam bahasa Indonesia.

3.4.2 Rancangan Struktur Pengolahan Citra *(Image Processing)*

Struktur Pengolahan Citra pada program aplikasi terdiri dari tiga buah proses secara berurut yaitu *cropping*, *grayscaling* dan *thresholding*. Tujuan dari modul ini ialah untuk menghasilkan citra teks yang dapat dikenali oleh layanan OCR. Urutan proses ditunjukan oleh *flowchart* pada gambar 3.10.



Gambar 3.10. *Flowchart* Modul Pengolahan Citra

3.4.2.1 *Cropping*

Pada proses ini, dilakukan pemotongan terhadap citra *input* dengan cara menyeleksi secara langsung area teks pada citra yang ingin dikenali. Proses ini menghasilkan sebuah potongan citra dari hasil penyeleksian area teks Arab yang dilakukan oleh *user*. Potongan citra yang dihasilkan oleh program aplikasi diperoleh berdasarkan titik awal penyeleksian area teks (x_0, y_0) dan titik akhir penyeleksian (x_1, y_1), dari kedua titik tersebut dapat ditentukan panjang dan lebar potongan citra baru yang akan dibuat oleh program aplikasi dengan menggunakan persamaan berikut.

$$width = x_m - x_n \quad (3.1)$$

$$height = y_m - y_n \quad (3.2)$$

Keterangan:

x_m = koordinat sumbu x dengan nilai lebih besar

x_n = koordinat sumbu x dengan nilai lebih kecil

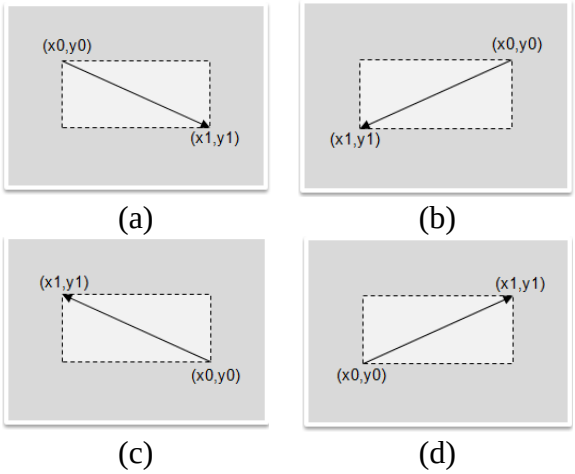
y_m = koordinat sumbu y dengan nilai lebih besar

y_n = koordinat sumbu y dengan nilai lebih kecil

$width$ = lebar gambar yang akan di-*crop*

$height$ = tinggi gambar yang akan di-*crop*

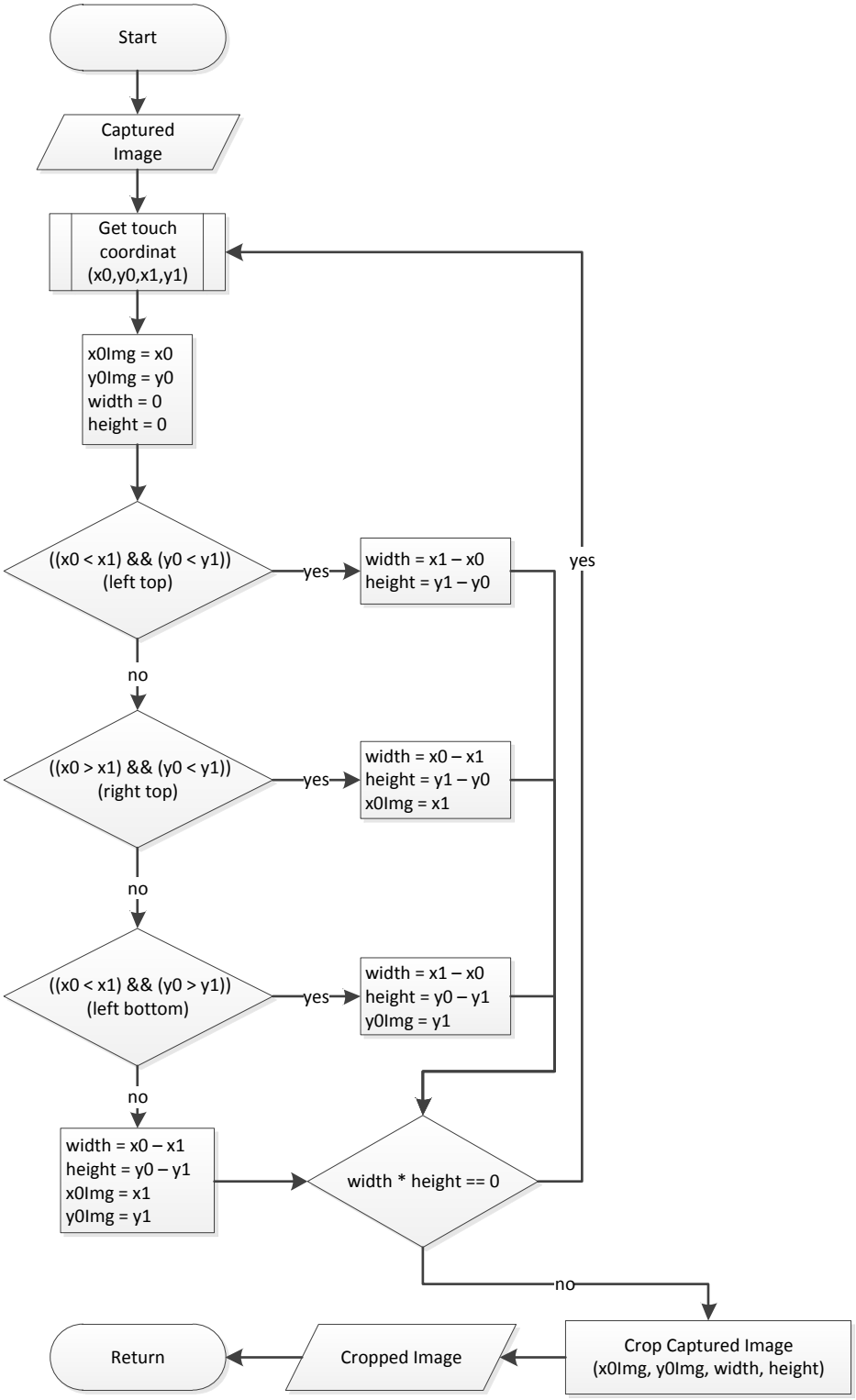
Persamaan di atas menghasilkan koordinat untuk gambar yang akan di-*crop*. Ilustrasi *cropping* dapat dilihat pada gambar 3.11 berikut.



Gambar 3.11. Ilustrasi *cropping*: (a) kiri-atas, (b) kanan-atas, (c) kiri-bawah , (d) kanan-bawah

Pada gambar diatas terdapat empat buah kemungkinan proses penyeleksian teks atau proses *cropping* yang dilakukan oleh *user*. Dari masing-

masing kemungkinan proses *cropping* tersebut akan menghasilkan koordinat (x_m, y_m) dan (x_n, y_n) yang berbeda untuk menentukan nilai *width* dan *height* serta koordinat awal dari potongan citra yang akan dihasilkan. Proses ini dapat dilihat pada gambar 3.12 berikut.



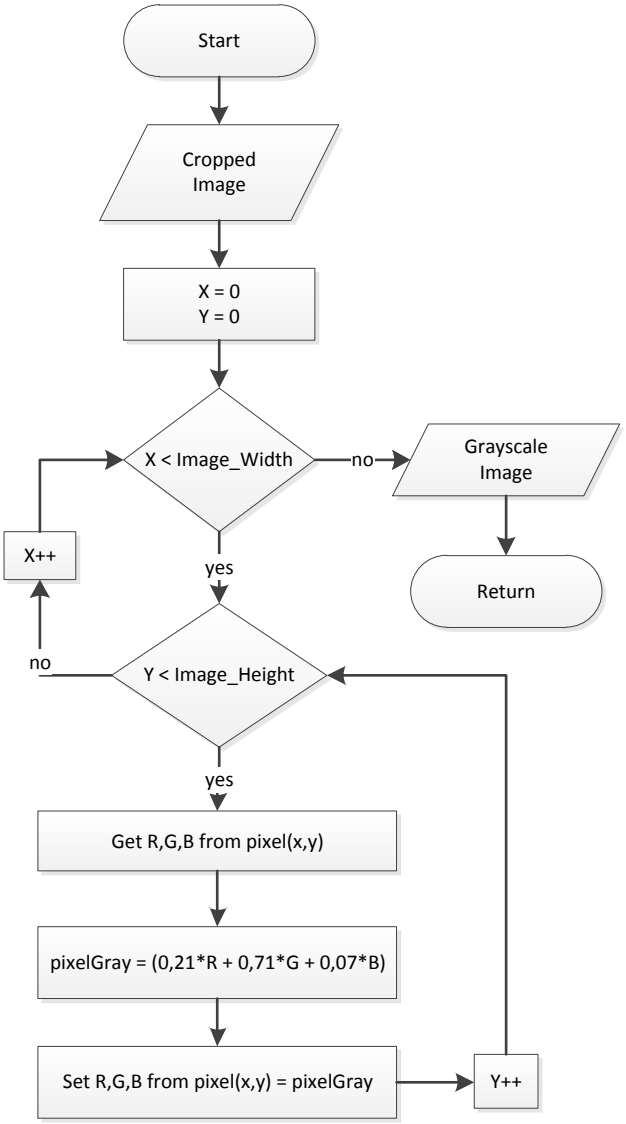
Gambar 3.12. *Flowchart cropping*

3.4.2.2 *Grayscaling*

Pada proses ini, potongan citra berwarna akan diubah menjadi citra yang citra abu-abu (*grayscale*). Transformasi citra ke bentuk *grayscale* dilakukan dengan mengambil *pixel* dari suatu citra yang kemudian dihitung dengan menggunakan persamaan berikut:

$$Grayscale = 0{,}21R + 0{,}71G + 0{,}07B \quad \text{.....(3.3)}$$

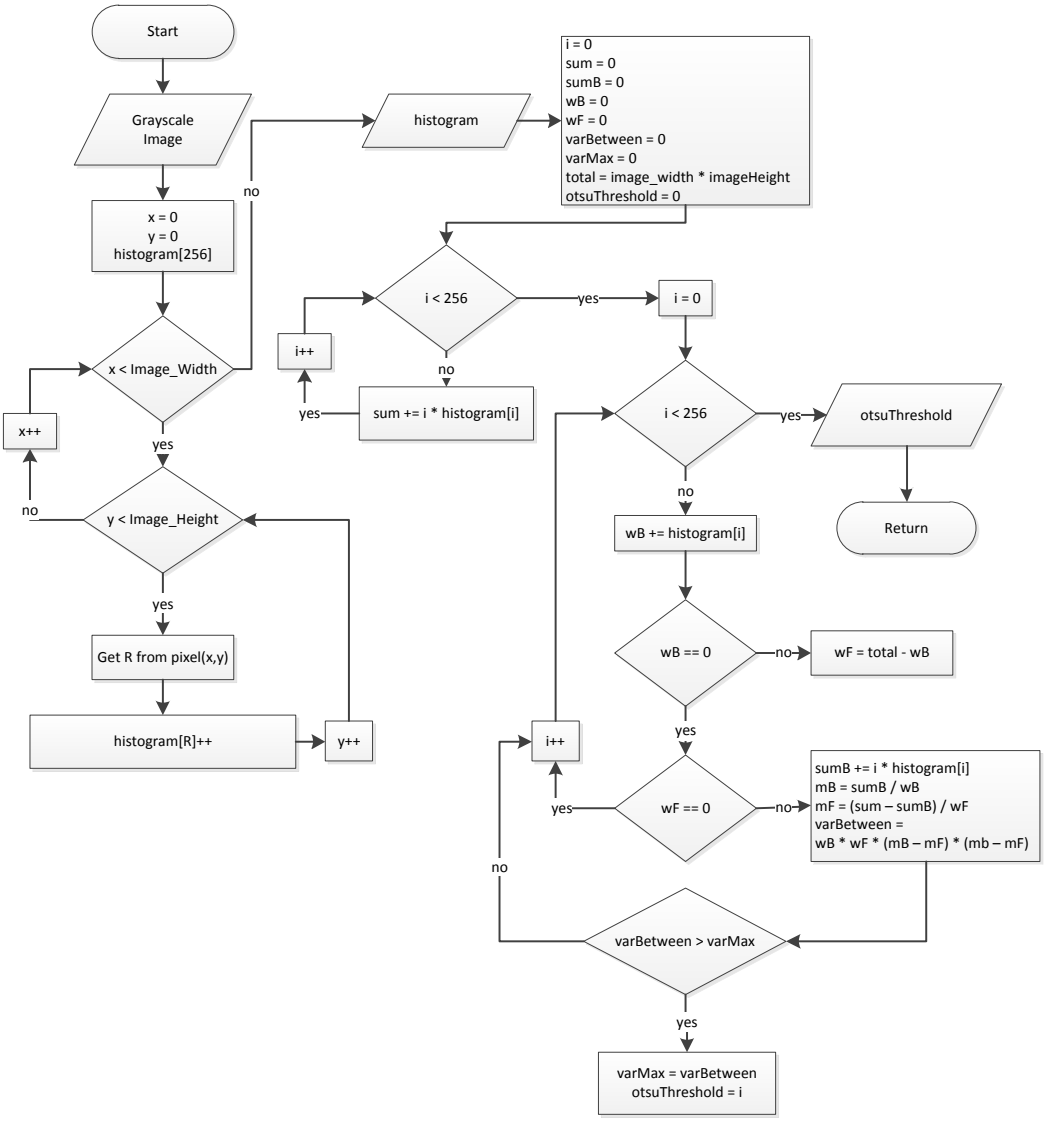
Proses pengubahan citra RGB ke dalam citra *grayscale* dapat dilihat pada gambar 3.13 berikut.



Gambar 3.13. *Flowchart grayscaling*

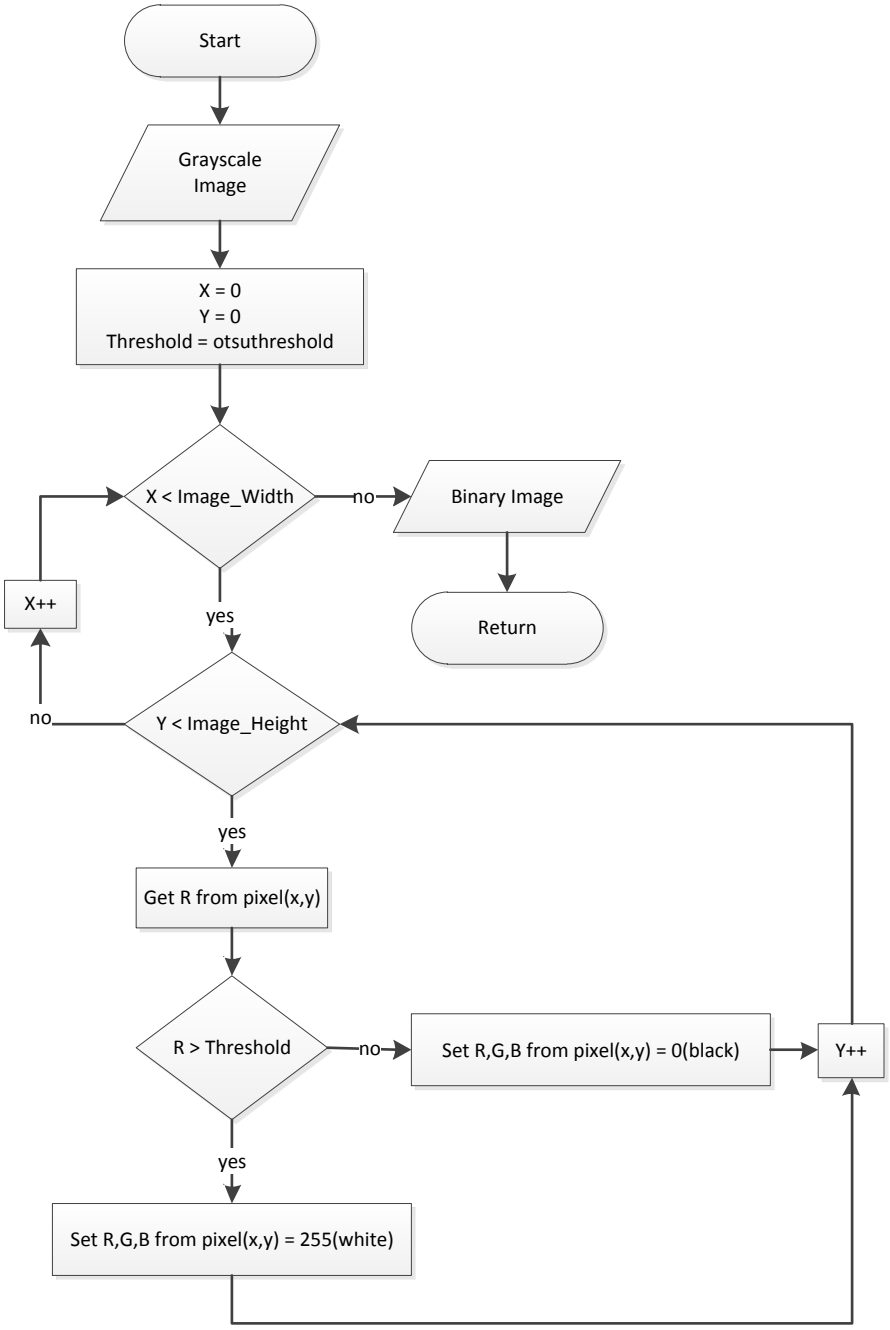
3.4.2.3 *Otsu Thresholding*

Pada proses ini, citra *grayscale* yang dihasilkan dari proses sebelumnya akan dirubah menjadi sebuah citra biner, yaitu citra yang hanya memiliki dua buah tingkat keabuan yaitu hitam dan putih. Proses penentuan pemberian tingkatan warna hitam atau putih, ditentukan berdasarkan *nilai* threshold yang diperoleh meggunakan metode Otsu yang dapat dilihat pada gambar 3.14 berikut.



Gambar 3.14. *Flowchart Otsu Method*

Proses binerisasi citra *grayscale* untuk menghasilkan citra biner dapat dilihat pada gambar 3.15 sebagai berikut:

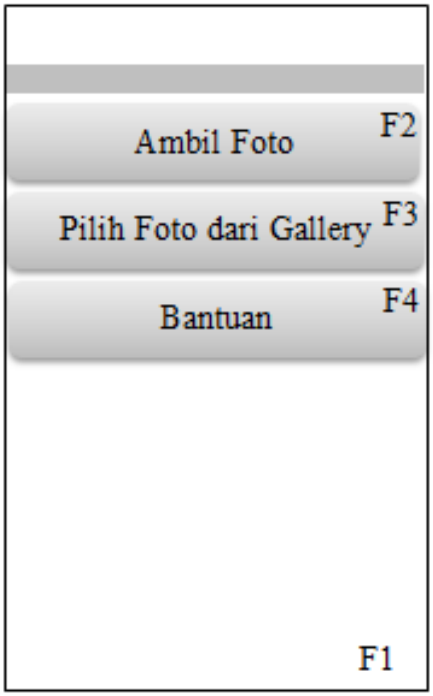


Gambar 3.15. *Flowchart Thresholding*

3.4.3 Rancangan Antarmuka

Sistem aplikasi yang akan dibangun memiliki dua buah tampilan utama, yaitu tampilan Menu Utama dan tampilan *Image View*. Tampilan Menu Utama

digunakan untuk menentukan sumber foto yang ingin diolah sedangkan *Image View* digunakan untuk menampilkan foto tersebut.



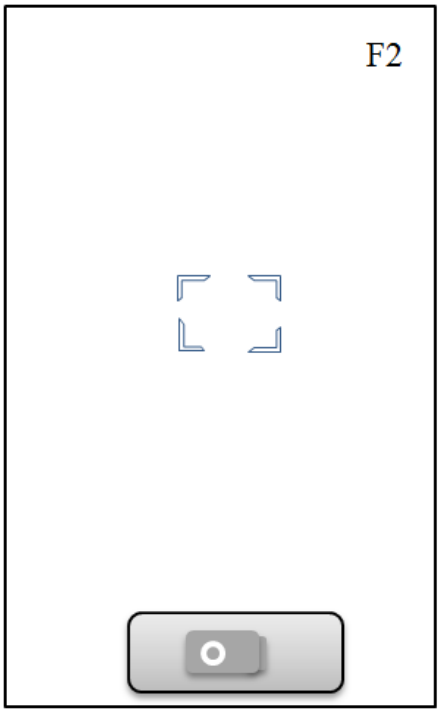
Gambar 3.16. Antarmuka aplikasi Android Form 1

Keterangan:

Pada saat tombol Ambil Foto ditekan, akan memanggil aplikasi kamera. Seperti yang terlihat pada tampilan F2 .

Pada saat tombol Pilih Foto dari Gallery di tekan, akan memanggil aplikasi gallery. Seperti yang terlihat pada tampilan F3.

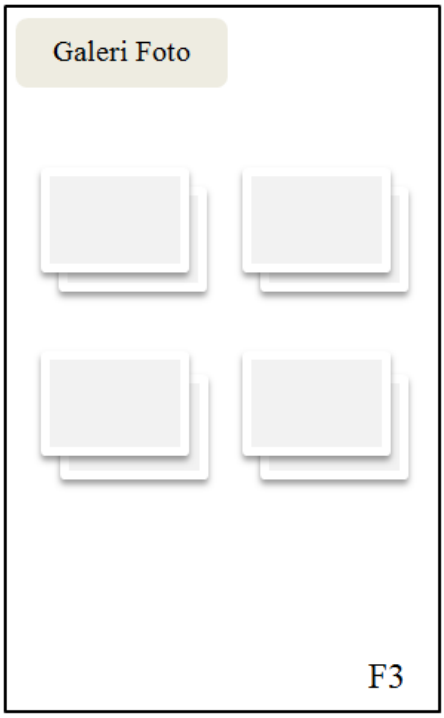
Pada saat tombol Bantuan di tekan, akan memanggil membuka dialog cara penggunaan aplikasi, seperti yang terlihat pada tampilan F4.



Keterangan:

Pada saat tombol *icon* kamera ditekan, maka foto akan disimpan di *memory external* dan foto tersebut akan ditampilkan pada tampilan F5

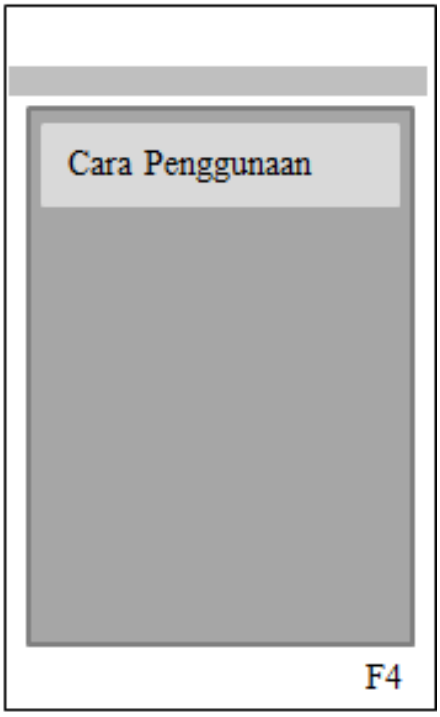
Gambar 3.17. Antarmuka aplikasi Android Form 2



Keterangan:

Ketika salah satu foto sudah dipilih, maka foto tersebut akan ditampilkan pada tampilan F5.

Gambar 3.18. Antarmuka aplikasi Android Form 3



Keterangan:

Form ini berfungsi sebagai petunjuk penggunaan aplikasi yang dibuat.

Gambar 3.19. Antarmuka aplikasi Android Form 4



Keterangan:

Pada tampilan ini user menentukan sendiri area teks Arab yang ingin dikenali, kemudian hasil pengenalan dan terjemahan teks tersebut akan terlihat seperti pada tampilan F6.

Gambar 3.20. Antarmuka aplikasi Android Form 5



Keterangan:

Ketika tombol kembali pada perangkat ditekan, maka tampilan akan kembali ke F1.

Gambar 3.21. Antarmuka aplikasi Android Form 6