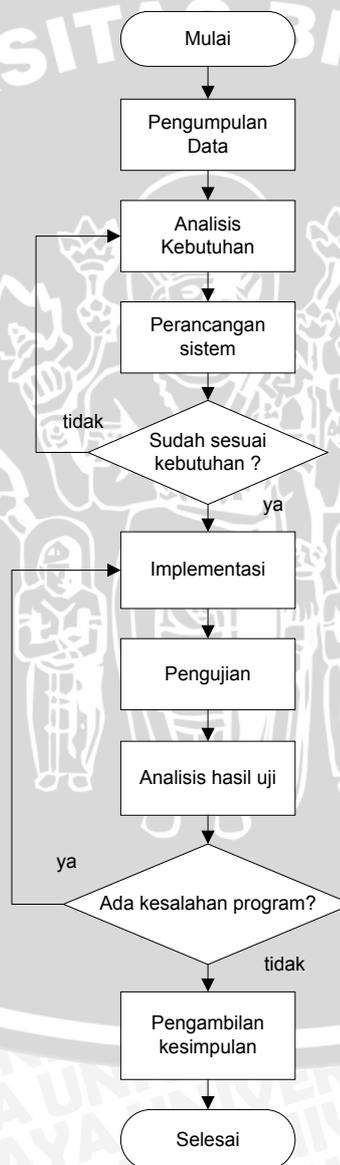


### BAB III METODE PENELITIAN

Bab ini menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi, diantaranya adalah studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian aplikasi, analisis hasil uji aplikasi dan pengambilan kesimpulan. Penulis menyertakan juga kesimpulan dan saran sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya.



**Gambar 3.1** Diagram Alir Metode Penelitian  
Sumber : Perancangan

### 3.1 Studi Literatur

Pada tahap ini penulis mempelajari berbagai referensi dan mengumpulkan data yang berhubungan dengan masalah yang akan dibahas, diantaranya adalah sebagai berikut:

- Interaksi Manusia dan Komputer
- Analisis Pengolahan Citra Digital
- Algoritma *K-Curvature* untuk mendeteksi *corner*

Referensi yang didapatkan oleh penulis berupa buku dan jurnal-jurnal penelitian pada kurun waktu tujuh tahun terakhir. Namun penulis juga mencantumkan referensi dengan tahun penerbitan lebih dari tujuh tahun terakhir disebabkan jurnal atau buku tersebut sudah tidak dicetak lagi dan metode yang diuraikan pada sumber tersebut masih digunakan pada penelitian-penelitian tujuh tahun terakhir ini.

### 3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mengetahui semua kebutuhan yang diperlukan oleh aplikasi yang akan dibangun sehingga dapat mencapai tujuan yang dimaksud. Analisis kebutuhan didapatkan dengan mengidentifikasi semua kebutuhan sistem yang direpresentasikan dalam pemodelan *Class Diagram* serta *Use Case Diagram*. Sistem atau aplikasi yang dibangun akan berorientasikan objek, sehingga pemodelan *Class Diagram* sangat cocok untuk menjelaskan kebutuhan dari sistem. Pemodelan *Use Case Diagram* digunakan untuk menggambarkan fungsi-fungsi apa saja yang disediakan oleh sistem.

### 3.3 Perancangan Sistem

Perancangan sistem dilakukan setelah semua data dan kebutuhan sistem terpenuhi. Data dan kebutuhan sistem tersebut didapatkan melalui tahap sebelumnya yaitu analisis kebutuhan. Dari hasil analisis kebutuhan, penulis dapat menentukan kelas apa saja yang dibutuhkan oleh sistem. Berikut adalah rancangan kelas yang ada di aplikasi beserta penjelasan relasi antar kelas:

1. Kelas Main

Kelas Main merupakan kelas utama atau kelas yang dijalankan pertama kali. Kelas main akan memanggil kelas-kelas lain yang digunakan untuk membantu *preprocessing* dan menangkap citra dari *webcam*. Di dalam kelas ini juga dideklarasikan kelas-kelas yang mendukung proses mendeteksi jumlah ujung jari yang direntangkan.

2. Kelas Capture

Merupakan kelas abstrak yang di dalamnya dideklarasikan beberapa konfigurasi dari aplikasi. Konfigurasi ini dimaksudkan untuk menghubungkan aplikasi dengan kamera lokal komputer sehingga nantinya aplikasi dapat menangkap citra dari kamera.

3. Kelas BinaryRegion

Kelas yang nantinya digunakan sebagai tempat penyimpanan pixel-pixel yang tersusun dalam kesatuan daerah atau *region*.

4. Kelas Contour

Kelas yang nantinya digunakan sebagai tempat penyimpanan kontour yang berhasil dideteksi oleh aplikasi.

5. Kelas ContourTracer

Merupakan kelas yang digunakan untuk mendeteksi atau mendapatkan kontour dari citra biner.

6. Kelas Test Render Callback

Kelas yang berada di dalam kelas main, kelas utama. Selain tempat pendeklarasian fungsi pengolahan citra digital, kelas ini juga difungsikan untuk menangkap citra dari kamera dan disimpan dalam variabel image. Dari data yang tersimpan tadi, dapat diproses lebih lanjut atau langsung ditampilkan pada form aplikasi.

### 3.4 Implementasi

Implementasi aplikasi dilakukan berdasarkan pada hasil perancangan aplikasi. Pada bab ini akan dijelaskan tentang lingkungan dari implementasi dan proses implementasi dari proses ekstraksi ciri citra telapak tangan oleh perangkat lunak. Pada bagian ini juga akan dijelaskan fungsi-fungsi spesifik dari tiap kelas.

### 3.4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat keras dan lingkungan perangkat lunak yang akan digunakan. Untuk pengembangan perangkat lunak yang mengekstraksi ciri telapak tangan dengan menggunakan algoritma *k-curvature*.

Perangkat keras yang digunakan dalam pengembangan perangkat lunak yang mengekstraksi ciri telapak tangan menggunakan algoritma *k-curvature* adalah:

1. Prosesor Intel i5 @ 2,4 Ghz
2. Memori 4 GB
3. Hardisk dengan kapasitas 500 GB
4. Monitor 13"
5. Keyboard
6. Webcam 2 Mega pixel

Perangkat lunak yang digunakan dalam pengembangan perangkat lunak untuk mengekstraksi ciri telapak tangan dengan menggunakan algoritma *k-curvature* adalah:

1. Sistem Operasi Mac OSX Lion
2. Netbean IDE 7.1
3. Java SE 6

### 3.4.2 Implementasi Perangkat Lunak

Berdasarkan perancangan dan analisis kebutuhan yang telah dijelaskan maka berikut ini diberikan penjelasan tentang implementasi dari setiap tahapan proses.

- a. Menangkap Citra dari *Webcam*

Pengguna memperlihatkan isyarat tangan di depan kamera komputer tanpa terdapat objek lain yang berwarna menyerupai warna kulit. Kamera akan menangkap setiap citra yang tertangkap baik didalamnya terdapat objek tangan maupun tidak. Kemudian citra yang ditangkap oleh kamera komputer digunakan sebagai data masukan aplikasi. Aplikasi selanjutnya melakukan penskalaan citra untuk menghasilkan ukuran citra yang diinginkan. Hasil citra yang ditangkap akan ditampilkan kepada pengguna melalui *form user interface* aplikasi.

b. Segmentasi Warna Kulit

Citra yang ditangkap melalui kamera memiliki format warna RGB. Untuk menghasilkan segmentasi warna kulit yang stabil, maka citra RGB diubah menjadi format YCrCb menggunakan Persamaan 2-1, 2-2 dan 2-3. Citra berwarna tersebut kemudian dijadikan citra biner, warna putih untuk *foreground* dan hitam untuk *background*. Untuk range warna tiap elemen Y, Cr dan Cb yang sesuai range warna kulit diberikan warna putih dan hitam untuk sebaliknya.

c. Mendapatkan Kontur Area

Citra biner yang telah didapatkan dari proses sebelumnya akan dicari konturnya melalui algoritma yang telah dijelaskan pada Bab II sub bab 2.24. Salah satu cara yang sederhana untuk menentukan kontur adalah memprosesnya dengan cara mengidentifikasi area yang terhubung dalam citra dan untuk tiap area, diproses sekelilingnya, dimulai dari pixel yang dipilih dari tepiannya.

d. Mendeteksi Lengkungan

Dari kontur yang telah didapatkan maka dari kumpulan titik yang terdapat dalam kontur dapat diketahui titik mana yang termasuk dalam *corner* yang ingin kita cari. Deteksi *corner* yang akan penulis gunakan adalah menggunakan algoritma *k-curvature*. Untuk mendeteksi *corner* diperlukan variabel inisialisasi yaitu nilai *k* dan nilai ambang *theta*. Nilai *k* digunakan untuk menentukan panjang kaki-kaki di sebelah samping kanan dan kiri dari titik yang akan dideteksi. Dan nilai *theta* digunakan untuk sudut yang dihasilkan antara kedua kaki. Untuk mendapatkan sudut dari ujung jari maka sudut yang dipakai adalah sudut lancip.

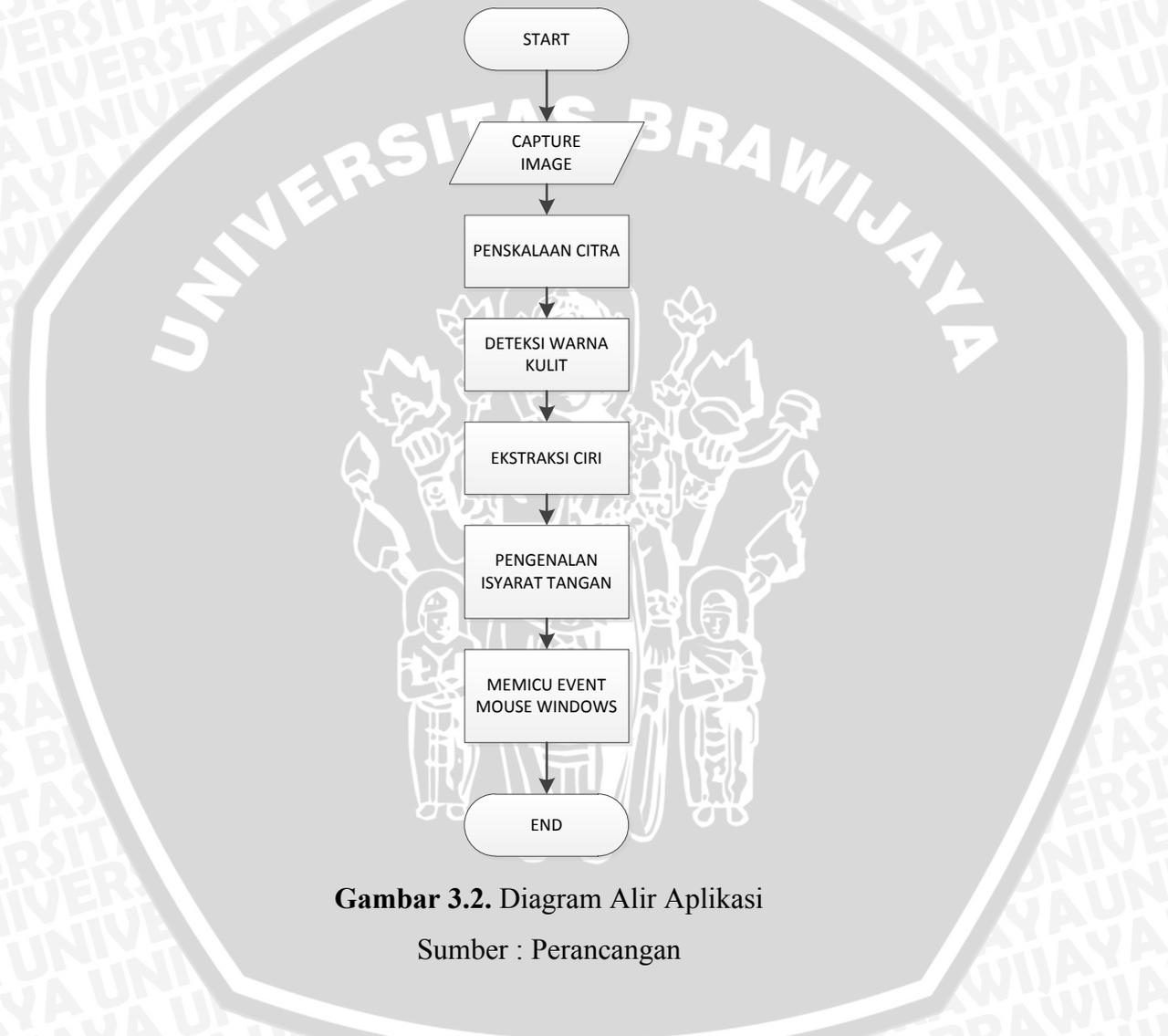
e. Pengenalan Isyarat

Dari proses sebelumnya dapat diketahui titik ujung dari jari. Dengan kata lain aplikasi dapat mengetahui berapa jumlah jari yang direntangkan. Jumlah *corner* yang berhasil untuk mendeteksi ujung jari adalah sama dengan jumlah jari yang sedang direntangkan. Dari jumlah jari yang direntangkan dijadikan sebagai isyarat dalam menggerakkan *mouse pointer komputer*. Misalnya satu jari untuk menggerakkan *pointer* dan dua jari untuk klik *drag pointer*.

f. Memicu Event Mouse

Aplikasi ini nantinya akan memanggil fungsi dari sistem operasi yaitu fungsi operasi *mouse pointer*. Agar aplikasi dapat memanggil fungsi yang dimiliki

oleh sistem operasi maka aplikasi memanfaatkan kelas Robot yang telah disediakan oleh java. Pada kelas robot terdapat berbagai macam method yang dapat digunakan untuk memicu event yang sering digunakan pada mouse. Selain dapat memicu *event mouse*, robot dapat difungsikan untuk memicu event lainnya seperti memicu tombol yang ditekan pada keyboard.



Gambar 3.2. Diagram Alir Aplikasi

Sumber : Perancangan

### 3.5 Pengujian Aplikasi

Penulis melakukan pengujian tentang tingkat akurasi algoritma *k-curvature* dalam mendeteksi ujung jari untuk tiap nilai  $K$  yang berbeda-beda. Algoritma *k-curvature* memerlukan inisialisasi awal yaitu nilai  $K$  dan *threshold* sudut kelengkungan. Besar resolusi diduga berpengaruh pada proses pengolahan

citra dalam mengesktraksi ciri telapak tangan. Selain waktu proses pengolahan citra yang diduga lebih lama dikarenakan ukuran gambar yang lebih besar, ukuran gambar juga diduga mempengaruhi keberhasilan pendeteksian kelengkungan. Oleh karena itu pada penelitian ini terdapat tiga buah variabel yang akan diuji keterkaitanya satu sama lainnya yaitu nilai  $K$ , nilai *threshold* sudut kelengkungan dan ukuran gambar. Selain itu, pada pengujian ini akan dihitung rata-rata waktu yang diperlukan aplikasi dalam memproses gambar untuk tiap ukuran gambar yang berbeda. Pengujian dilakukan dengan membandingkan jumlah hasil pendeteksian benar dari aplikasi dengan jumlah citra asli yang diinputkan ke aplikasi. Untuk mendapatkan tingkat keakurasian penulis menggunakan metode *error rate* dan *mean square error*.

### 3.6 Analisis Aplikasi

Analisis dilakukan untuk mendapatkan pola karakteristik dari algoritma *K-curvature* atau *K-cosine* dalam mendeteksi lengkungan. Variabel uji dalam algoritma yang penulis analisis adalah ukuran gambar masukan, sudut *threshold* dan nilai  $K$ . Perolehan data hasil pengujian untuk tiap perlakuan berbeda dari variabel uji akan dianalisis, sehingga penulis mendapatkan nilai variabel yang optimal dengan tingkat kesalahan yang seminimal mungkin dan mengetahui keterkaitan variabel satu sama lainnya. Tingkat kesalahan algoritma dalam mengekstraksi ciri citra telapak tangan didapatkan dari perhitungan *error rate* dan *mean square error*.

### 3.7 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, pengujian dan analisis sistem aplikasi telah selesai dilakukan. Kesimpulan penelitian akan menjawab dari tujuan penelitan. Hasil kesimpulan diberikan berdasarkan dari analisis hasil pengujian yang telah dilakukan sebelumnya. Selain kesimpulan, disertakan pula saran untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya, baik di bidang sama maupun di bidang lain.