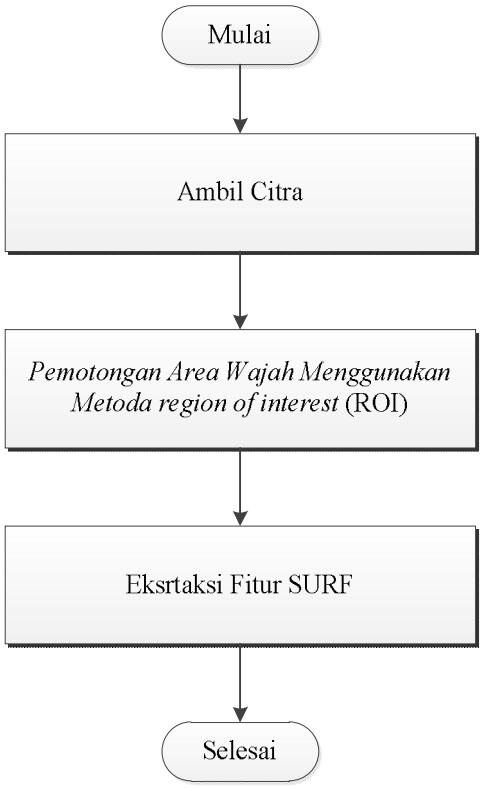
# BAB III

# PERANCANGAN SISTEM

## 3.1 Diagram Blok

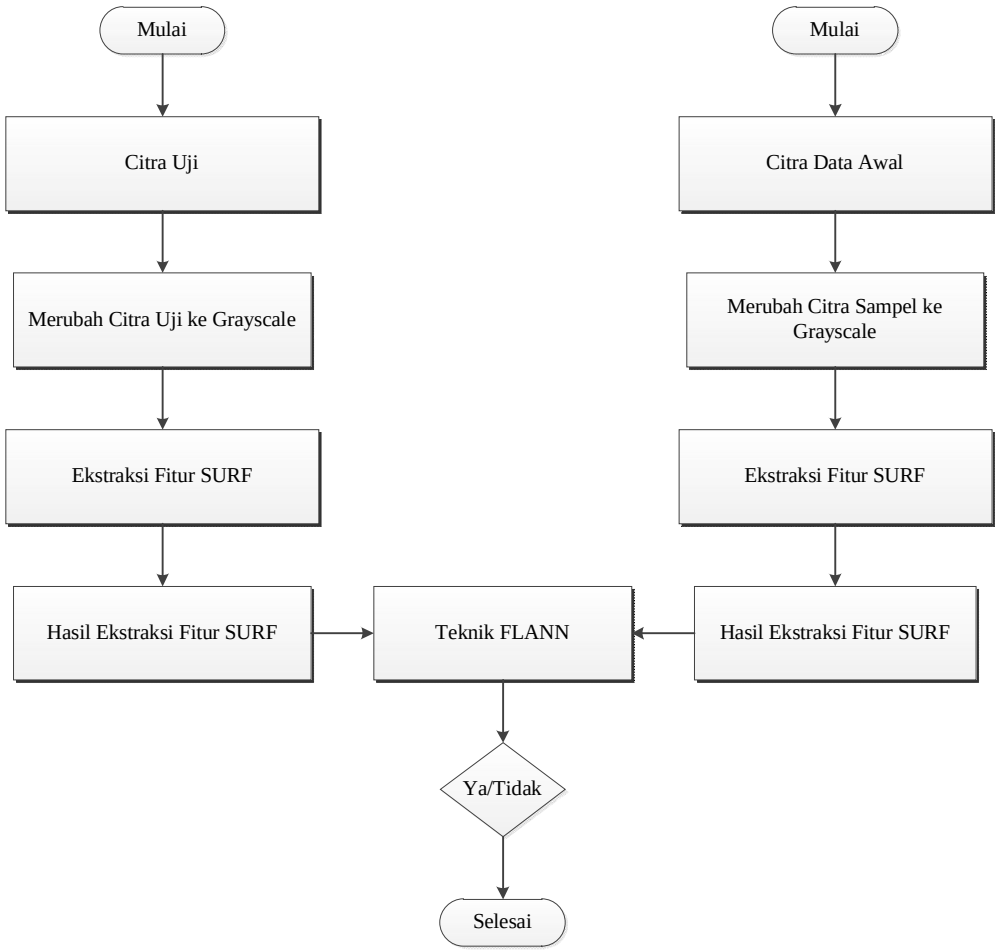
Sistem yang di rancang terdiri dari dua blok utama yaitu blok proses pemotongan area wajah dan proses ekstraksi fitur menggunakan metode SURF. Berikut ini merupakan diagram blok sistem pemotongan area wajah :




Gambar 3.1 Diagram Blok Keseluruhan Sistem

Blok pemotongan area wajah menggunakan metode *region of interest* (ROI), hal ini dilakukan untuk memudahkan proses ekstraksi fitur SURF hanya pada area wajah saja. Blok proses ekstraksi fitur citra dilakukan dengan memanfaatkan metoda *Speed-Up Robust Features* (SURF). Dimana citra yang diekstraksi adalah citra uji maupun citra data awal. Untuk selanjutnya hasil

ekstraksi citra uji dan citra data awal akan di bandingkan nilainya dengan metode FLANN. Berikut ini merupakan diagram blok secara keseluruhan sistem yang akan dirancang :




Gambar 3.2 Diagram Blok Sistem Deteksi Wajah Menggunakan Metode SURF

Fitur SURF ini digunakan untuk menentukan *keypoint* pada objek dalam citra. Setiap objek citra uji atau citra sdata awal akan di ekstrak oleh fitur SURF yang dimiliknya. Pada proses pencocokan (*matching)* fitur SURF dengan menggunakan teknik FLANN, yang akan di cocokan hanyalah fitur *keypoint* yang berada pada wajah saja dari bagian gambar yang menjadi latar belakangnya. Dengan demikian fitur yang ekstrak hanya *keypoint* yang ada pada area wajah saja

## 3.2 Fitur SURF

Fitur SURF memiliki nilai yang konstan atau stabil (*invariant*) terhadap perubahan skala, rotasi, *blurring*, pencahayaan dan transformasi 3 dimensi. Oleh karena itu dapat digunakan untuk proses pendeteksian wajah. Deteksi wajah menggunakan metoda SURF pada intinya adalah mencari *interest point* pada suatu lokasi objek wajah yang terdeteksi. Terdapat dua langkah yang dapat kita lakukan untuk mendeteksi wajah menggunakan metoda SURF :

1. Langkah pertama : Membandingkan ada atau tidaknya *interst point* pada keadaan tersebut yang dapat dijadikan parameter yang cukup kuat untuk mendefinisikan suatu wajah. Pada langkah ini dapat diperkuat dengan membuat nilai minimal *hessian* yang bisa dikalibrasi sebelum mengakftifasi proses deteksi.

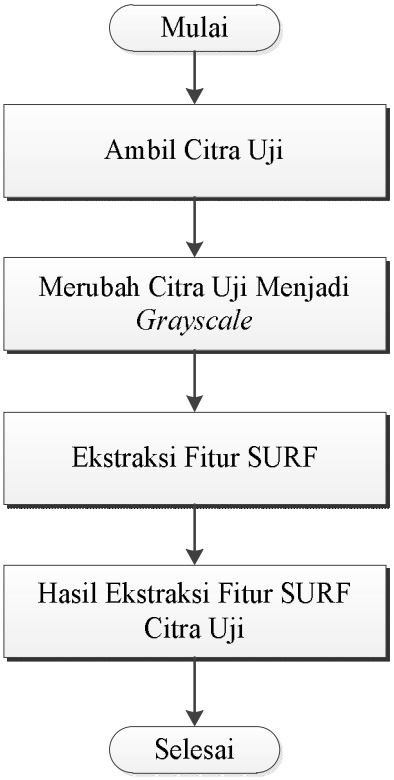
2. Langkah kedua : Jika parameter yang didapatkan dari langkah pertama cukup kuat, maka tahap selanjutnya adalah proses pencocokan (*matching)* setiap *interest point* pada keadaan yang diketahui sebagai suatu objek wajah. Proses pencocokan dapat dilakukan dengan menggunakan metoda *Flann matcher* dan diperbaiki unuk mengeliminasi pasangan-pasangan *interest point* yang kurang tepat dengan menggunakan jarak *encludian.* Untuk melakukan pencocokan, maka diperlukan citra sampel wajah yang diambil sebelum proses deteksi wajah.

## 3.3 Diagram Alir

Berikut ini adalah diagram alir yang akan digunakan dalam membuat algoritma program untuk mendukung sistem pendeteksian wajah menggunakan metoda SURF ini :

### 3.3.1 Ekstraksi Citra Uji Menggunakan Metoda SURF

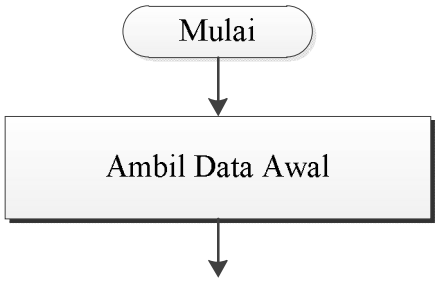
Pada proses ini dilakukan ekstraksi SURF pada data uji dengan berbagai posisi. Ekstraksi ini dilakukan untuk mempermudah pada tahap perbandingan. Berikut ini merupakan tahap ekstraksi citra uji :

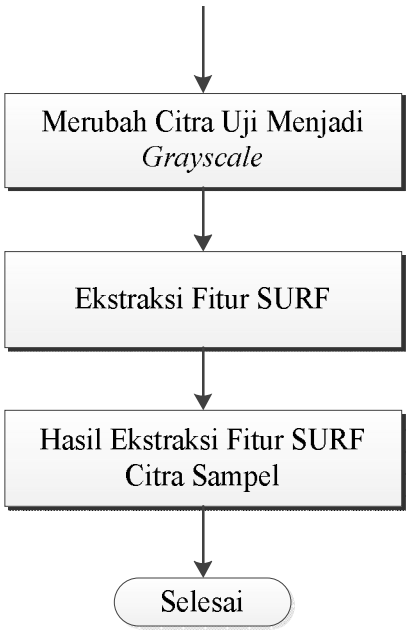



Gambar 3.3 Diagram Ekstraksi Citra Uji Menggunakan Metoda SURF

### 3.3.2 Ekstraksi Citra Data Uji Menggunakan Metoda SURF

Ekstraksi citra data awal tidak berbeda jauh dengan ekstraksi citra uji. Ekstraksi ini dilakukan agar mendapatkan hasil ekstraksi SURF yang selanjutnya akan digunakan sebagai citra pembanding pada proses *matching*. Berikut ini diagram blok ekstraksi citra sampel :

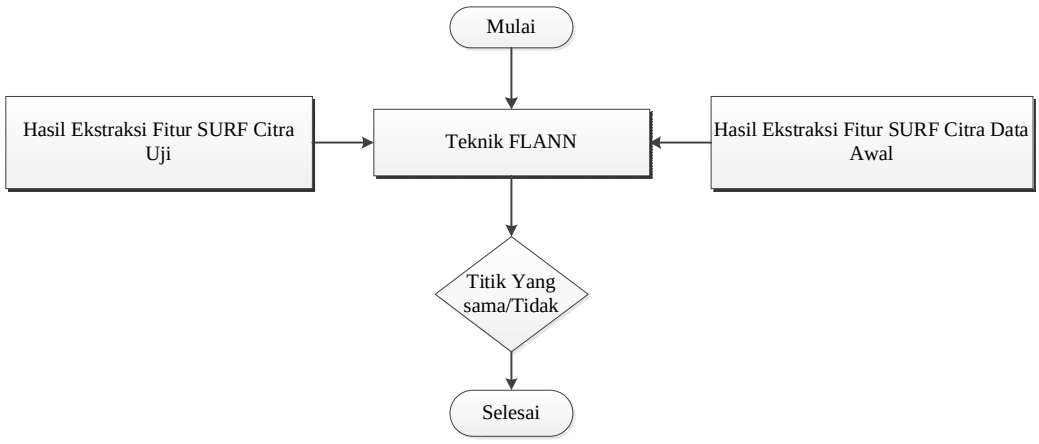





Gambar 3.4 Diagram Ekstraksi Citra Data Awal Menggunakan Metoda SURF

### 3.3.3 *Matching Interest Point* Menggunakan Teknik FLANN

Perbandingan (*matching)* dengan teknik FLANN dilakukan untuk mencari nilai *interest point* yang dianggap sama pada masing-masing citra, baik itu *interest point* citra uji ataupun *interest point* citra data awal. Berikut ini merupakan diagram blok proses perbandingan :




Gambar 3.5 Diagram Perbandingan (*Matching*) dengan Teknik FLANN

## 3.4 Perancangan Sistem

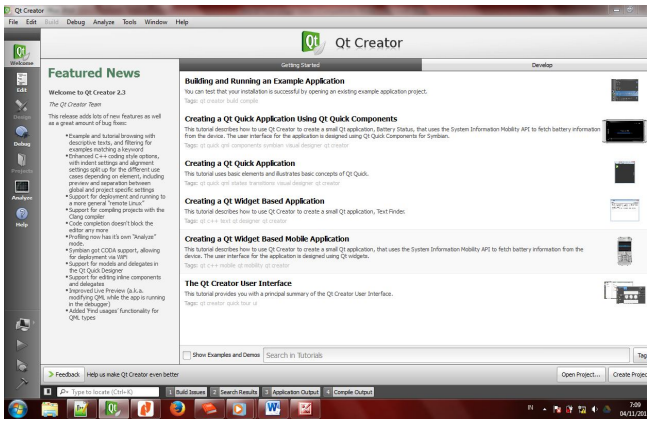
Pada proses perancangan sistem ini menjelaskan tentang bagaimana cara konfigurasi *library* terhadap *IDE* yang digunakan dan antarmuka sistem yang akan dilakukan.

### 3.4.1 Konfigurasi OpenCV

OpenCV adalah *library* tambahan yang digunakan pada penelitian ini, sehingga tidak otomatis terhubung dengan IDE yang digunakan. Maka dari itu perlu dilakukan konfigurasi tertentu untuk menghubungkan *library* dengan IDE yang digunakan agar sistem pendeteksian wajah ini dapat berjalan. Pada penelitian ini *library* yang digunakan adalah OpenCV. OpenCV dianggap cukup mendukung pada proses pendeteksian wajah menggunakan metoda SURF ini, karena *library* yang digunakan sudah tersedia. Sementara untuk IDE menggunakan QT Creator

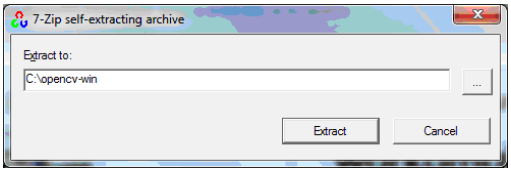
Berikut ini merupakan tahapan konfigurasi *library* OpenCVdengan *IDE* QT Creator:

1. Install *IDE* Qt Creator secara *default*, setalah itu kita akan dapat menggunakannya. Berikut ini merupakan tampilan *IDE* QT Creator setelah sukses di *install* :



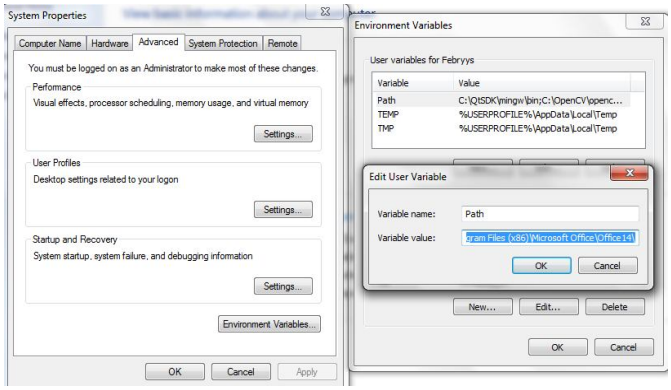

Gambar 3.6 Tampilan *IDE* QT Creator

2. Selanjutnya jalankan dan ekstrak OpenCV 2.3.1 pada folder yang berada pada *directori* C:/



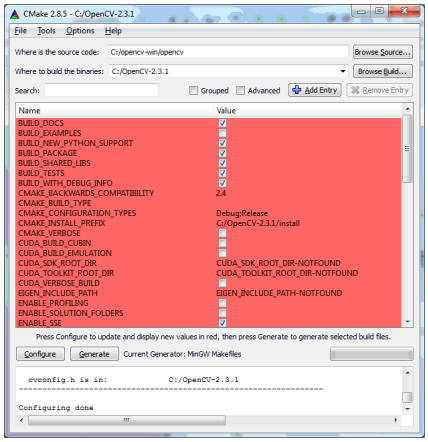

Gambar 3.7 Ekstraksi OpenCV

3. Setelah di ekstrak, selanjutnya adalah *compile library* menggunakan *compiler* dasar *mingw/g++* yang dimiliki *default* oleh Qt Creator. Akan tetapi sebelum itu dilakukan, kita harus membuat *path* pada *environment variable* pada komputer kita. Berikut ini tampilan konfigurasi pada *environment variable* :




Gambar 3.8 Konfigurasi *Environment Variable*

4. Setelah tahap konfigurasi *environment variable* selesai, selanjunya kita akan membangun *library* yang dimiliki OpenCV menggunakan *sofware* Cmake. Cmake ini merupakan *cross-platform* dan *open source* untuk membangun *library packages.* Ini merupakan gambar dimana sedang dilakukan *build library* menggunakan *Cmake* :




Gambar 3.9 *Build Library* OpenCV Menggunakan Cmake

Setelah prose *build* ini selesai, QT Creator sudah dapat menggunakan *library* yang dimiliki oleh OpenCV.

5. Penyertaan *library* OpenCV pada sistem yang akan dirancang dengan cara menambahkan *library file* kedalam *linker input properties* dari *project prorerties* yang akan dirancang :

   *INCLUDEPATH+=C:\\OpenCV\\opencv\\binary\\opencv2.3.1\\install\\include*

   *LIBS += -LC:\\OpenCV\\opencv\\binary\\opencv2.3.1\\install\\lib*

   *-lopencv_core231.dll *

   *-lopencv_highgui231.dll *

   *-lopencv_imgproc231.dll *

   *-lopencv_features2d231.dll *

   *-lopencv_calib3d231.dll*

   *-lopencv_contrib231.dll*

   *-lopencv_flann231.dll*

*-lopencv_gpu231.dll*

*-lopencv_legacy231.dll*

*-lopencv_ml231.dll*

*-lopencv_objdetect231.dll*

*-lopencv_video231.dll*

### 3.4.2 Antarmuka Sistem

Antarmuka sistem yang akan dibangun berupa *frame command prompt* dan *frame window* yang terpisah, hal ini didasari karena keterbatasan *library* yang tidak mendukung *manage code*, yang menyebabkan terjadi *error* pada perancangan dengan antarmuka berbasis *windows form application*. Rancangan antarmuka yang akan dirancang terdiri dari :

1. Hasil ekstraksi fitur SURF pada data uji.
2. Hasil ekstraksi fitur SURF pada data sampel.
3. Hasil perbandingan (*matcher*) *interest point* menggunakan teknik FLANN.
4. Hasil jumlah *interest point* yang terdeteksi dari citra uji.
5. Hasil jumlah *interest point* yang terdeteksi dari citra sampel.
6. Hasil jumlah *interest point* yang terdeteksi pada hasil perbandingan dengan teknik FLANN.