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.

Perpustakaan Univeristas Multimedia Nusantara

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam uji pengaruh tingkat variasi iluminasi pada pengulangan pendeteksian fitur dengan algoritma *DCT* mengikuti standar penelitian metodologi *System Development Life Cycle (SDLC)* dengan beberapa tahapannya yaitu planning, analisis, perancangan sistem, pengembangan sistem, serta pengujian sistem. Adapun penjabaran secara mendetail mengenai metode penelitian yang akan digunakan adalah sebagai berikut.

a. Planning

   Tujuan *planning* adalah untuk mengatasi permasalahan yang terdapat dalam bidang pendeteksian fitur wajah dengan merujuk pada penelitian sebelumnya. Pada penelitian ini permasalahan yang dirumuskan adalah bagaimana pengaruh tingkat variasi iluminasi pada kualitas pengulangan pendeteksi fitur. Masalah yang ada dapat diselesaikan melalui pengembangan sistem yang memanfaatkan algoritma *image processing* untuk meningkatkan hasil akurasi pendeteksian melalui aplikasi *Matlab.*

b. Analisis

   Pada tahap analisis akan dikumpulkan semua informasi mengenai teori berdasarkan kasus yang akan ditangani oleh sistem. Solusi atas permasalahan variasi iluminasi pada gambar wajah akan diselesaikan dengan berdasarkan atas teori mengenai algoritma *DCT*, algoritma *CLAHE*, teori *Epipolar Geometry*, Iluminasi, *Keypoints*, *Feature Detector*, dan teori pada

perhitungan *Recall*, *Precision* serta *F-score*. Studi literatur didapatkan dari artikel, jurnal ilmiah yang ada, serta penelitian sebelumnya dan penelitian terkait lainnya.

c. Perancangan Sistem

Pada tahap ini, dilakukan perancangan alur kerja dari sistem. Perancangan dibuat dalam bentuk *flowchart* yang berfungsi menyederhanakan rangkaian proses yang akan diimplementasikan pada sistem dapat lebih mudah dipahami.

d. Pengembangan Sistem

Pada tahap ini kode program akan dituliskan sesuai dengan rancangan yang sudah ada pada tahap sebelumnya.

e. Pengujian Sistem

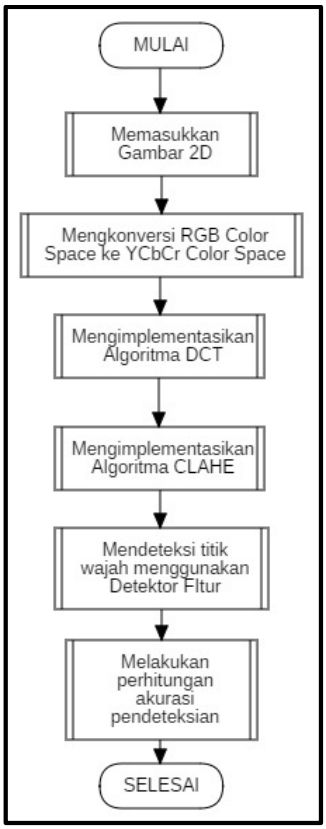
Pada tahap ini penelitian akan dilanjutkan dengan pengujian terhadap sistem. Pengujian dilakukan dengan menjalankan sistem, dan dengan menggunakan hasil *output* dari sistem akan dilakukan perhitungan *keypoints* wajah yang berhasil terdeteksi dari program yang dirancang pada tahap sebelumnya. Pada tahap pengujian juga akan dilakukan proses evaluasi dengan menghitung *recall*, *precision* dan juga *F-score* dengan tujuan mendapatkan hasil akurasi dari penelitian yang telah dilakukan.

3.2 Perancangan Sistem

A. Flowchart Utama

Flowchart utama menggambarkan keseluruhan alur proses penelitian yang dapat dilihat pada gambar 3.1. Alur proses pendeteksian *keypoints* wajah terdiri dari beberapa tahapan yaitu memasukkan gambar 2D, mengonversi *RGB Color Space*

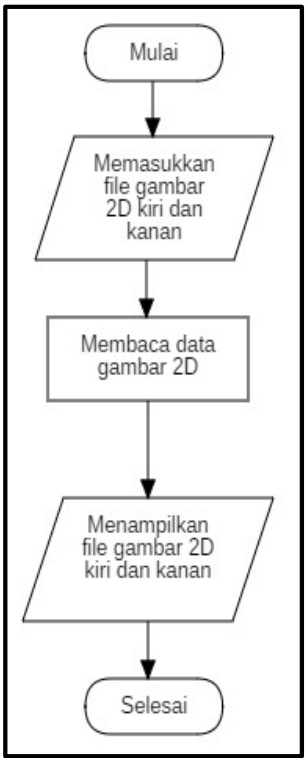
ke *YCbCr Color Space*, Mengimplementasikan Algoritma *DCT*, Mengimplementasikan Algoritma *CLAHE*, Mendeteksi *keypoints* dengan menggunakan detektor fitur, dan diakhiri dengan perhitungan akurasi hasil pendeteksian.



Gambar 3.1 Flowchart Utama

B. Flowchart Memasukkan Gambar 2D

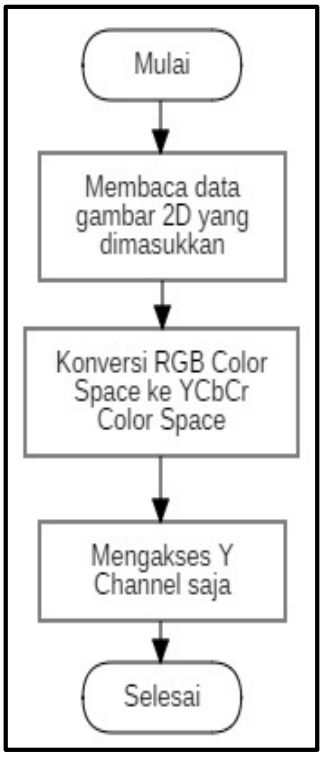
Flowchart pada gambar 3.2 menggambarkan alur proses memasukkan gambar 2D yang akan digunakan dalam proses penelitian. Gambar 2D tersebut diakses dari dua *database* yaitu database *Head-Pose* dan *ORL*. *Dataset* gambar diambil dari sisi kiri dan kanan dengan sudut 15 derajat. Setiap *database* akan dipilih 3 pasang *dataset* gambar wajah 2D untuk dilakukan pengujian *keypoints*.



Gambar 3.2 Flowchart Memasukkan Gambar 2D

C. Flowchart Mengkonversi RGB Color Space ke YCbCr Color Space

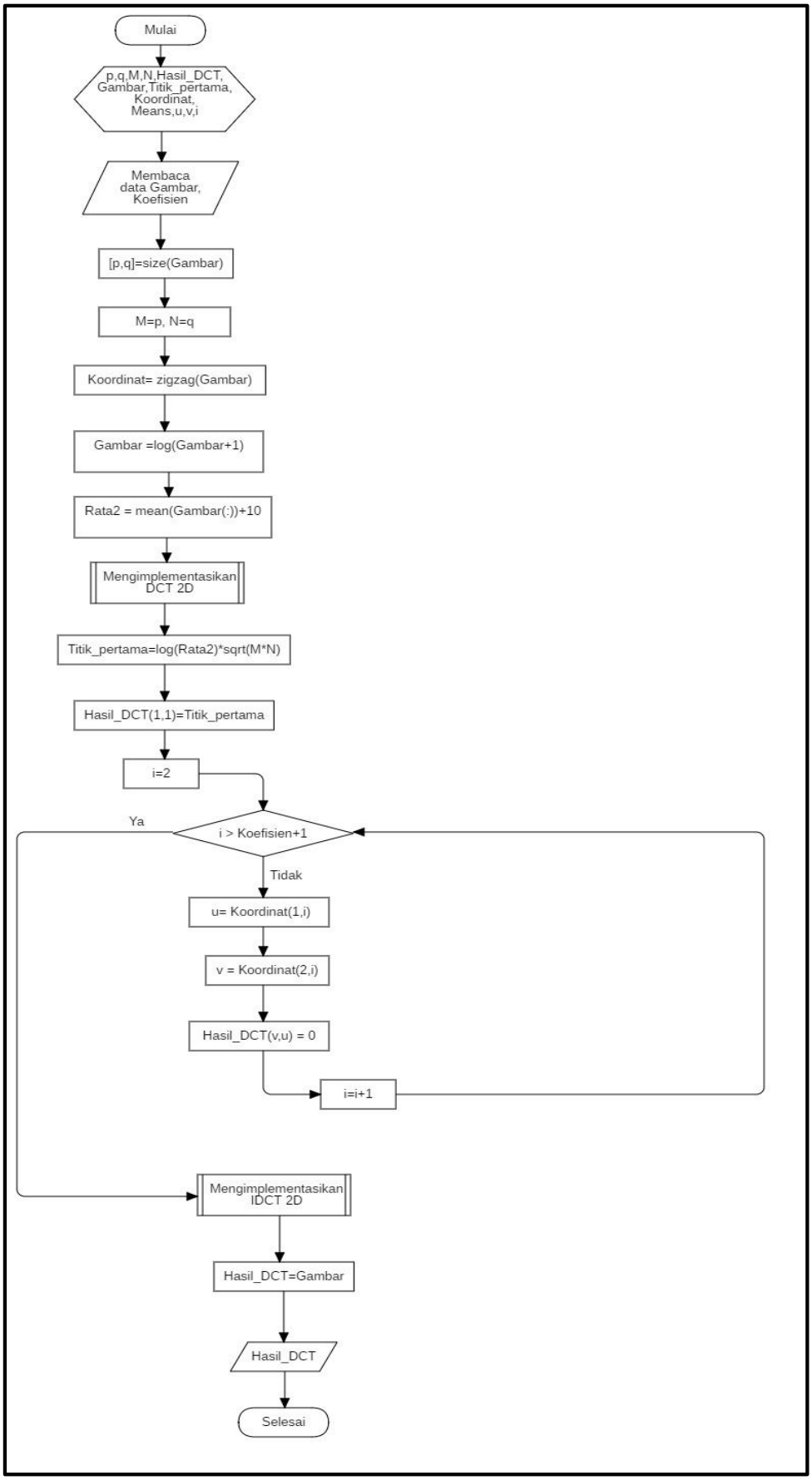
Flowchart pada gambar 3.3 menggambarkan alur proses setiap pasangan *dataset* akan dikonversi dari *RGB channel* ke *YCbCr channel.* Proses konversi dilakukan untuk mendapatkan *file* gambar yang terkompresi namun masih dapat menyimpan sebagian besar informasi dari gambar asli. Setelah proses konversi, *channel* warna yang akan digunakan dalam penelitian hanya Y *channel* saja karena sebagian besar informasi yang diterima mata manusia adalah komponen *luminance* pada Y *channel* bukan *chrominance* pada *Cb* dan *Cr Channel*. Gambar terdiri dari *pixel* digital yang direpresentasikan dalam format *RGB*, dimana 0 dan 255 masing-masing mewakili warna hitam dan putih sehingga komponen Y *channel* dapat diperoleh sesuai dengan persamaan berikut.



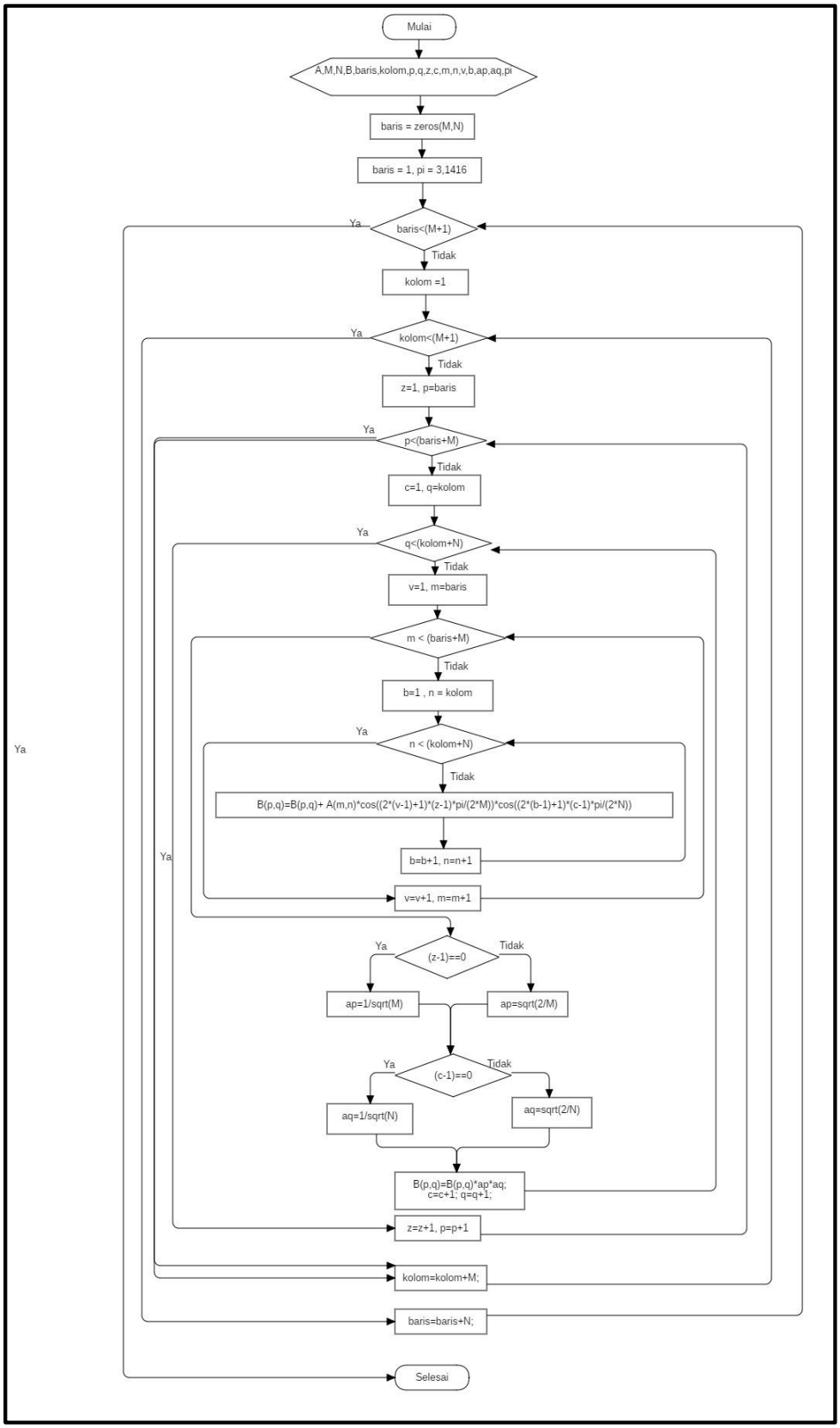
Gambar 3.3 Flowchart Mengkonversi RGB Color Space ke YCbCr Color Space

D. Flowchart Mengimplementasikan Algoritma *DCT*

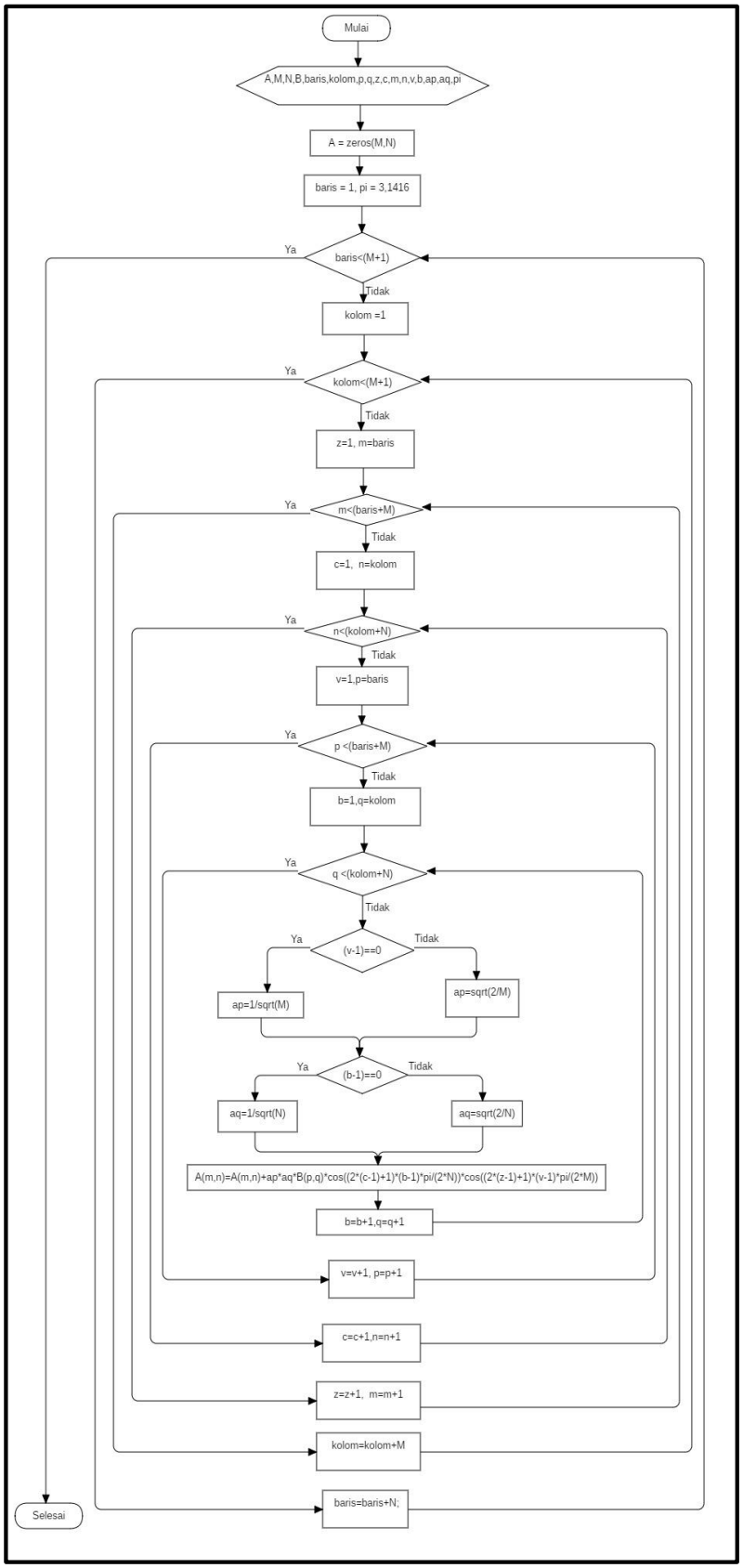
Flowchart pada gambar 3.4, 3.5 dan 3.6 menggambarkan alur implementasi algoritma *DCT* pada gambar 2D kiri dan kanan. Sebelum diimplementasikan, gambar akan terlebih dahulu ditransformasi ke domain logaritma dan frekuensi dikarenakan algoritma *DCT* bekerja pada domain logaritma suatu gambar dan transformasi ke dalam logaritma frekuensi dapat mempermudah pengecekan perubahan yang terjadi pada setiap pixel gambar. Algoritma *DCT* yang diimplementasikan terdiri dari tiga koefisien yaitu 25, 50 dan 75. Untuk mengembalikan hasil *DCT* berupa gambar dalam domain spasial, maka diakhiri dengan proses *inverse DCT*.



Gambar 3.4 Flowchart Mengimplementasikan Algoritma DCT



Gambar 3.5 Flowchart Mengimplementasikan DCT 2D



Gambar 3.6 Flowchart Mengimplementasikan IDCT 2D

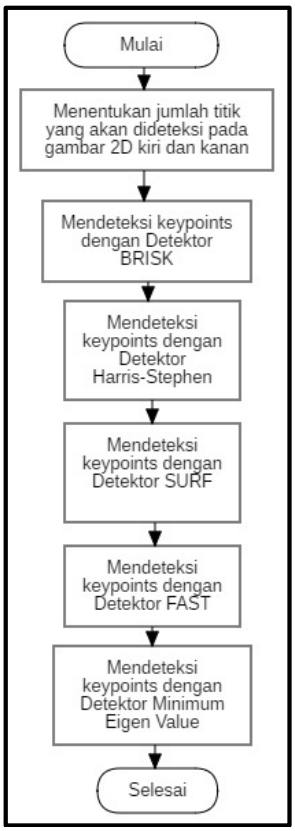
E. Flowchart Mengimplementasikan Algoritma CLAHE

Flowchart implementasi algoritma *CLAHE* menggambarkan alur tahapan pengimplementasian algoritma *CLAHE* dengan menggunakan *library* MATLAB yang tertera pada gambar 3.7. Gambar akan terlebih dahulu diakses dari gambar hasil implementasi algoritma *DCT* yang telah dilakukan pada tahap sebelumnya. Dilanjutkan proses '*clip*' pada *pixel* gambar yang dilakukan dengan penentuan value *clip limit* yang bertujuan untuk mencegah over saturasi pada gambar (pada pengujian ini digunakan nilai *clip limit default* yaitu 0.01). Kemudian *pixel* gambar yang tersisa setelah proses '*clip*' akan didistribusikan. Pengimplementasian algoritma *CLAHE* diakhiri dengan meningkatkan intensitas di setiap wilayah dengan transformasi *Rayleigh*.

Gambar 3.7 Flowchart Mengimplementasikan Algoritma CLAHE

F. Flowchart Mendeteksi Titik Wajah dengan Detektor Fitur

Flowchart pendeteksian *keypoints* wajah menggambarkan alur pendeteksian *keypoints* pada gambar 2D kiri dan kanan setelah melalui proses implementasi algoritma *DCT* dan *CLAHE* dapat dilihat pada gambar 3.8. Pendeteksian *keypoints* dilakukan dengan menggunakan lima fitur detektor yaitu *SURF*, *BRISK*, *FAST*, *Harris-Stephen* dan *Minimum Eigen-Value*.

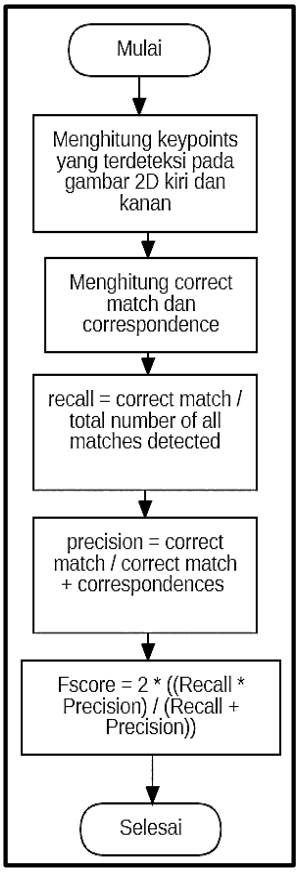


Gambar 3.8 Flowchart Mendeteksi Titik Wajah dengan Detektor Fitur

G. Flowchart Melakukan Perhitungan Akurasi Pendeteksian

Flowchart perhitungan akurasi menggambarkan alur perhitungan perhitungan recall, precision dan f-score yang akan menjadi hasil akurasi pendeteksian *keypoints* dengan menggunakan detektor fitur. *Keypoints* yang terdeteksi akan dihitung secara manual. Terdapat dua jenis posisi *keypoints* yang akan dihitung yaitu *correct match* dan *correspondents*. *Correct match* adalah

pasangan *keypoints* dari 2 sisi gambar yang terletak pada titik yang sama dan sesuai dengan tabel teori 15 *facial keypoints*. *Correspondent* adalah pasangan titik yang terletak pada posisi yang sama namun tidak termasuk dalam daftar 15 *facial keypoints* yang dapat dilihat pada Tabel 2.1.



Gambar 3.9 Flowchart Melakukan Perhitungan Akurasi Pendeteksian