



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, 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

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi Penelitian yang digunakan dalam tahap penelitian uji pengaruh tingkat variasi frekuensi tinggi pada kualitas pendeteksian fitur menggunakan algoritma *discrete cosine transform* (DCT) terdiri atas rangkaian proses *Systems Development Life Cycle* (SDLC) yang terdiri dari perencanaan, analisis, perancangan sistem, implementasi, uji coba dan analisis hasil, konsultasi dan penulisan skripsi.

A. Perencanaan

Tahapan ini dilakukan dengan merumuskan masalah yang secara spesifik berkaitan dengan pendeteksian fitur pada citra wajah 2D. Rumusan masalah tersebut diambil berdasarkan pada rujukan penelitian terkait yang pernah diuji dan diteliti. Pada penelitian ini permasalahan yang dirumuskan adalah tentang pengaruh tingkat variasi frekuensi tinggi pada kualitas pendeteksian fitur menggunakan algoritma DCT.

B. Analisis

Pada tahap ini dilakukan pencarian dan pembelajaran mengenai teori-teori yang berkaitan dengan penelitian seperti *Discrete Cosine Transform*(DCT), *Epipolar Geometry*, *Recall*, *Precision*, *F-Score*, *Image Processing*, dan *Feature Detector*. Materi terkait teori-teori tersebut didapatkan dari berbagai sumber, seperti jurnal ilmiah, buku, situs-situs *online* dan lainnya. Selain itu, pada tahapan ini juga

akan dicari perangkat lunak dan keras apa yang cocok sehingga pekerjaan menjadi lebih efektif.

C. Perancangan Sistem

Pada tahapan ini alur proses dari sistem akan digambarkan dalam bentuk *flowchart*. *Flowchart* digunakan untuk menggambarkan urutan proses secara mendetail dan sederhana sehingga sistem dapat dipahami dengan mudah.

D. Implementasi

Berdasarkan pembelajaran dan rancangan yang dibuat maka pada tahap ini diimplementasikan kode program pada MATLAB untuk menguji pengaruh tingkat variasi frekuensi tinggi pada kualitas pendeteksian fitur menggunakan algoritma DCT.

E. Uji Coba dan Analisa Hasil

Setelah selesai tahapan implementasi, akan dilanjutkan dengan pengujian hasil penelitian. Berdasarkan *keypoints* yang terdeteksi maka uji coba yang dilakukan adalah dengan menghitung jumlah *keypoints* yang ada. Kemudian hasil uji coba tersebut dinilai dengan menggunakan perhitungan *recall*, *precision*, dan *F-score* sehingga didapatkan hasil akurasi dari uji coba yang dilakukan.

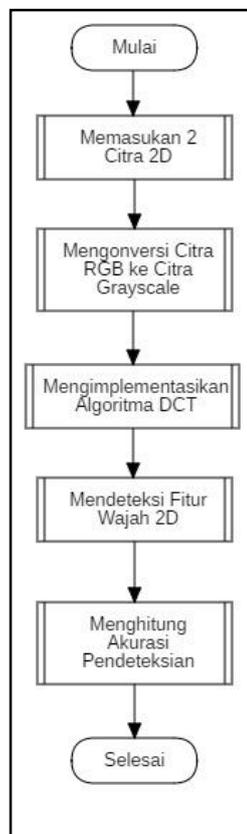
F. Konsultasi dan Penulisan Skripsi

Pada tahap ini hasil evaluasi yang telah ada dikonsultasikan kepada pembimbing untuk mendapatkan hasil uji yang lebih baik. Kemudian, dilakukan penulisan laporan dengan tujuan mendokumentasikan proses penelitian yang dilakukan berdasarkan implementasi dan uji coba penelitian. Selain itu, dokumentasi ini dapat dijadikan bukti bahwa penelitian telah selesai.

3.2 Perancangan Sistem

A. *Flowchart* Utama

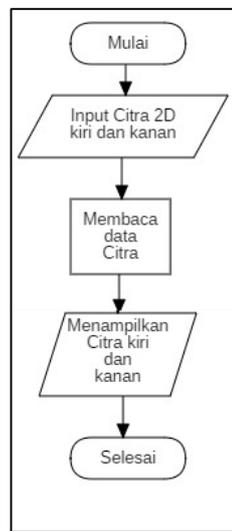
Flowchart utama digunakan untuk menjelaskan rangkaian secara umum dari alur proses yang dipakai dalam pengujian pengaruh tingkat variasi frekuensi tinggi pada kualitas pendeteksian fitur menggunakan algoritma *discrete cosine transform* (DCT). Alur proses dimulai dari memasukkan 2 citra dua dimensi berwarna ke dalam MATLAB yang kemudian dikonversi menjadi suatu citra *grayscale* atau keabuan. Langkah berikutnya adalah mengimplementasikan algoritma DCT untuk menghilangkan derau yang ada pada citra sehingga didapatkan citra yang baik untuk mendeteksi fitur wajah pada citra. Setelah fitur wajah dideteksi maka akan dihitung tingkat akurasi yang didapat dari hasil pengujian. Adapun *flowchart* yang dimaksud dapat dilihat pada gambar 3.1.



Gambar 3. 1 Flowchart Utama

B. *Flowchart* Memasukan 2 Citra 2D

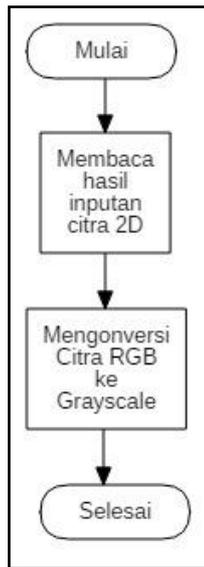
Flowchart pada gambar 3.2 merupakan langkah pertama yang dilakukan pada proses pengujian citra. Pada proses ini akan dimasukan 2 buah citra dua dimensi yang diambil dari arah yang berbeda yaitu, kiri dan kanan dengan sudut tertentu. Kedua citra yang digunakan adalah citra yang menampilkan wajah dari seseorang. Sepasang citra yang digunakan adalah citra yang sudah diubah ke ukuran 128 x 128. Ukuran lain sempat dicoba namun hasil yang lebih memuaskan terdapat pada ukuran 128 x 128.



Gambar 3. 2 *Flowchart* Memasukan 2 Citra 2D

C. *Flowchart* Mengonversi Citra RGB ke Citra Grayscale

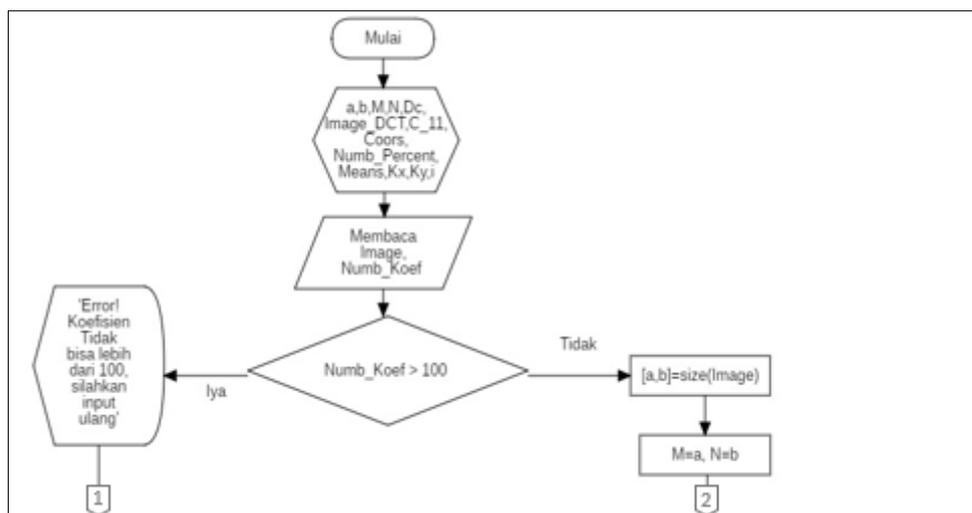
Pada *flowchart* yang ada pada gambar 3.3 dilakukan proses konversi warna citra yang awalnya RGB kemudian diubah menjadi *grayscale* atau keabu-abuan. Citra yang dikonversi menjadi *grayscale* mempunyai informasi yang cukup untuk pengenalan wajah sehingga menjadikan pengujian lebih spesifik fokus pada informasi yang dibutuhkan. Selain itu, ukuran citra yang didapat menjadi relatif lebih kecil dibandingkan dengan citra yang memiliki warna RGB. Citra dikonversi menggunakan *library* yang sudah disediakan MATLAB.



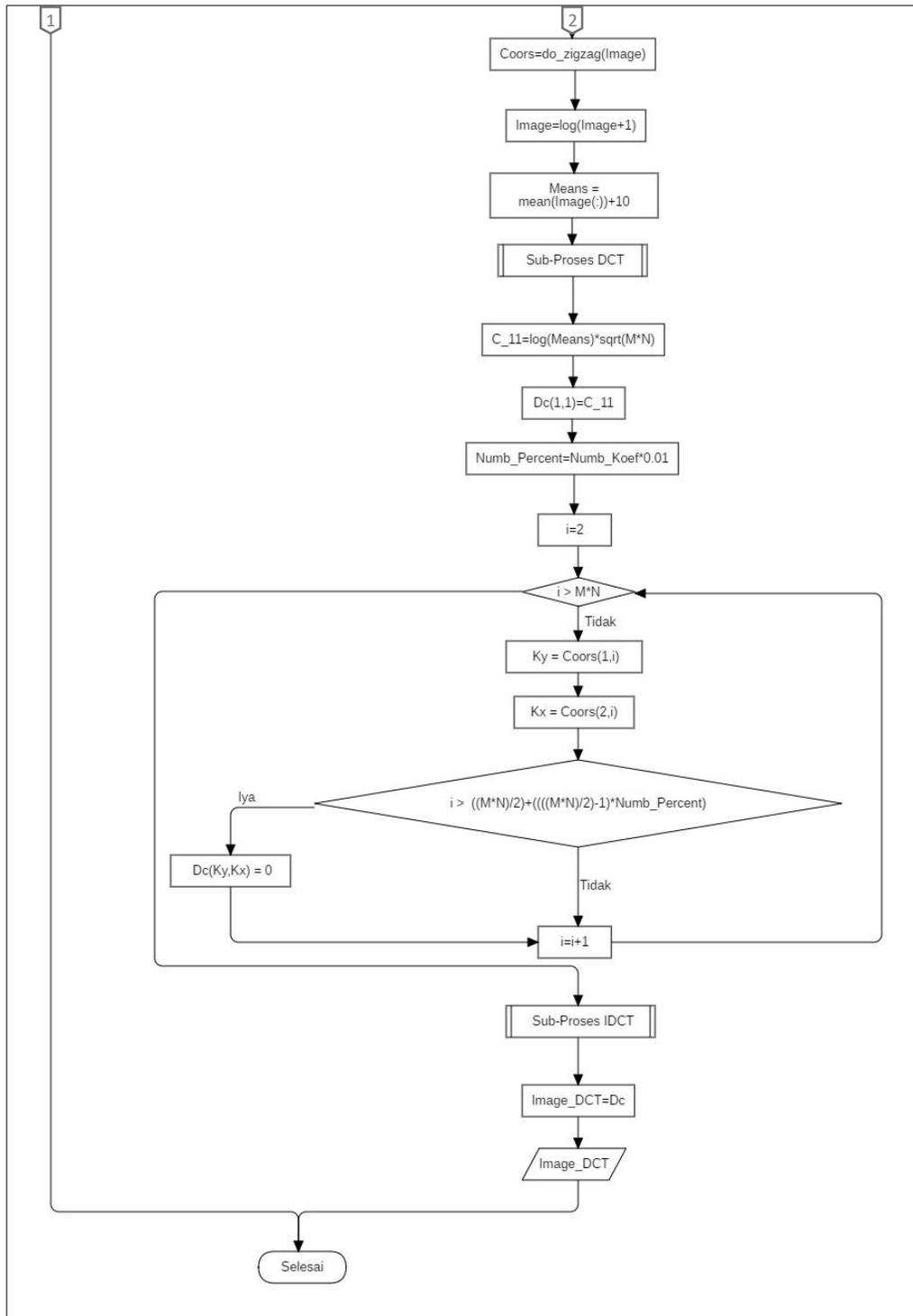
Gambar 3. 3 Flowchart Mengonversi Citra RGB ke Citra Grayscale

D. Flowchart Mengimplementasikan Algoritma DCT

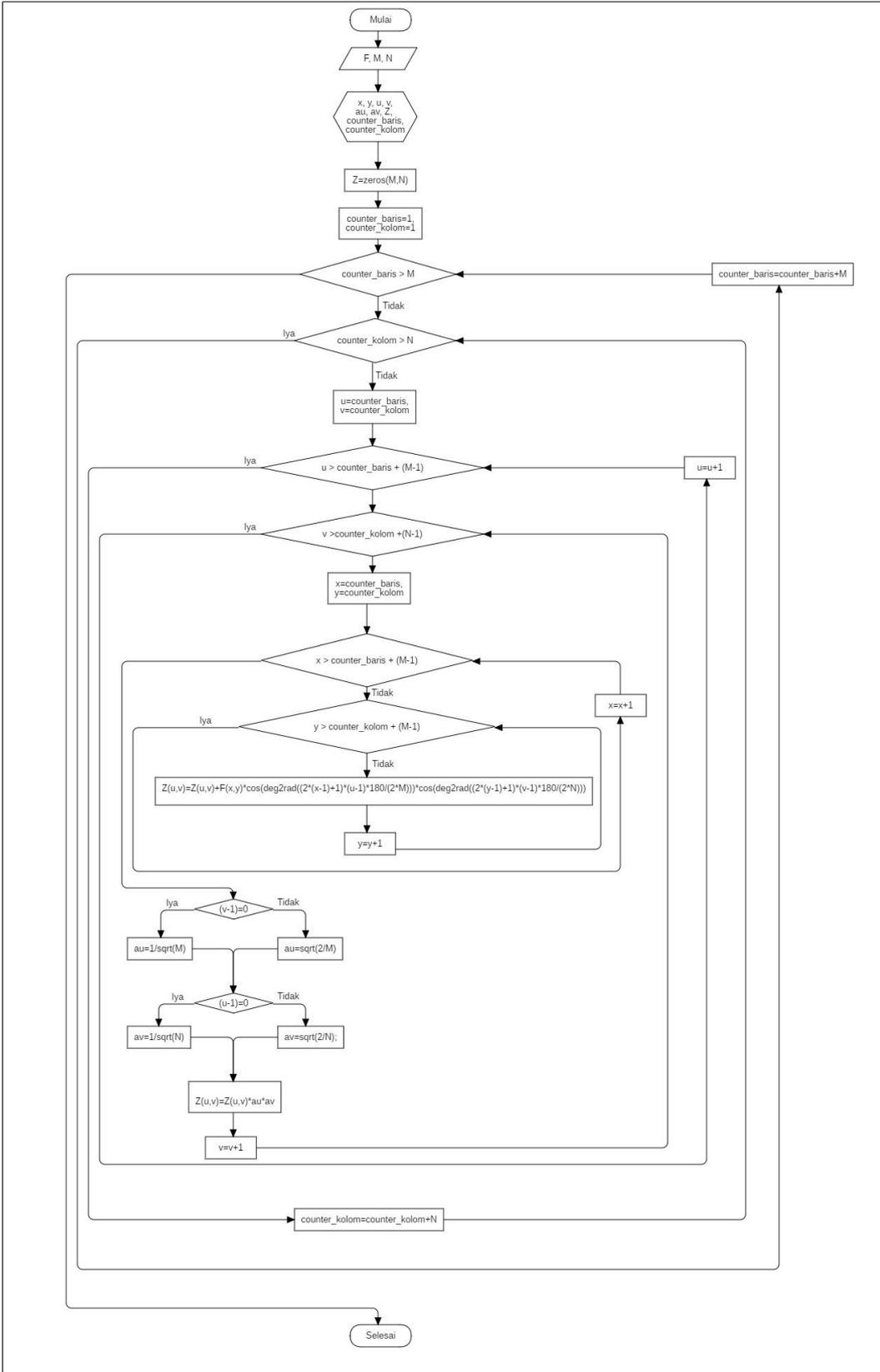
Flowchart pada gambar 3.4 menjelaskan tentang proses implementasi algoritma DCT secara umum. DCT yang digunakan pada proses ini spesifik dipakai untuk menghilangkan derau yang direpresentasikan pada pita frekuensi tinggi. Pada gambar a terdapat sub-proses yang menjelaskan lebih spesifik tentang alur kerja algoritma DCT sedangkan pada gambar b adalah *flowchart* yang menjelaskan sub-proses dari inverse DCT.



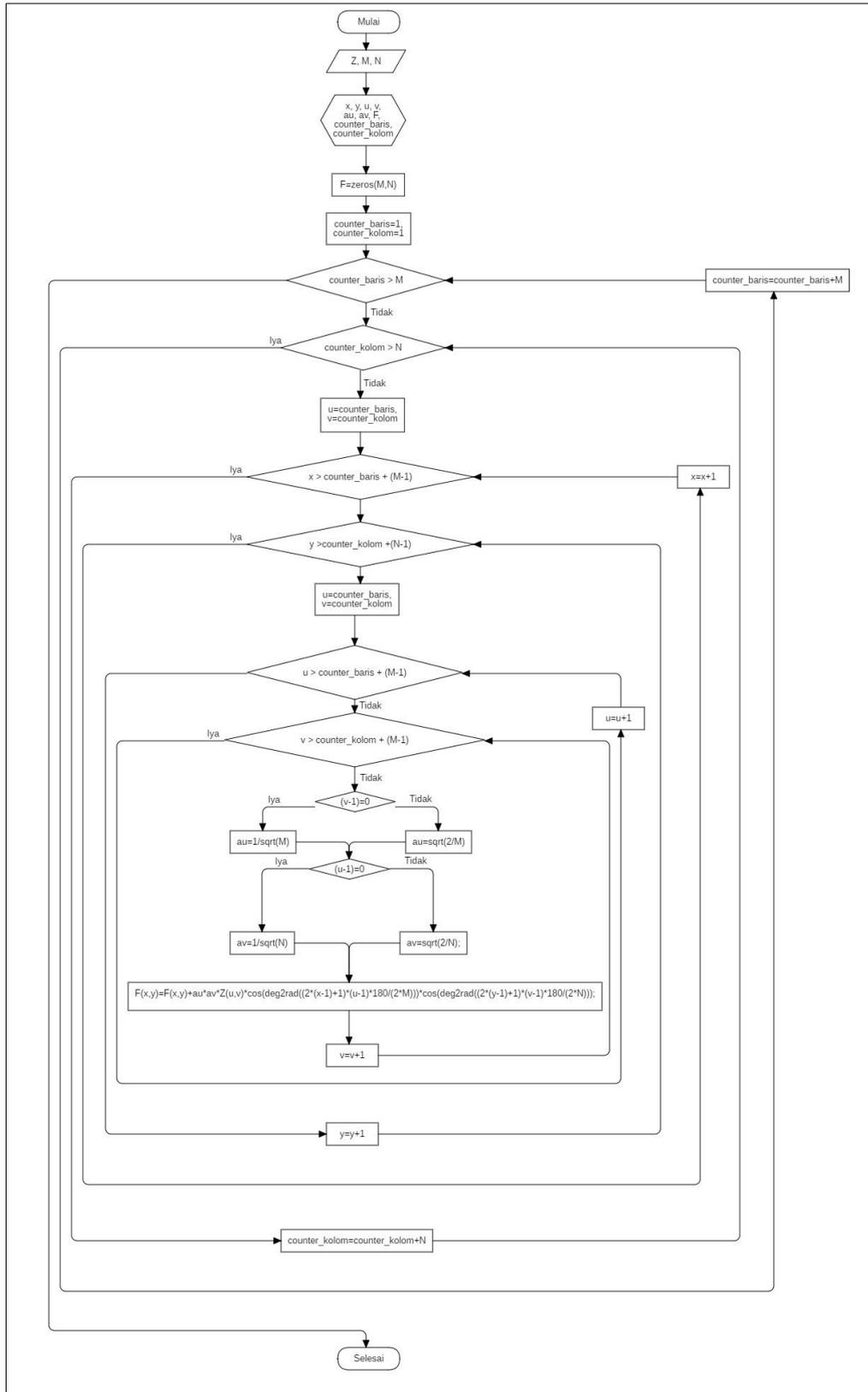
Gambar 3. 4 Flowchart Mengimplementasikan Algoritma DCT



Gambar 3. 5 Flowchart Lanjutan Mengimplementasikan Algoritma DCT



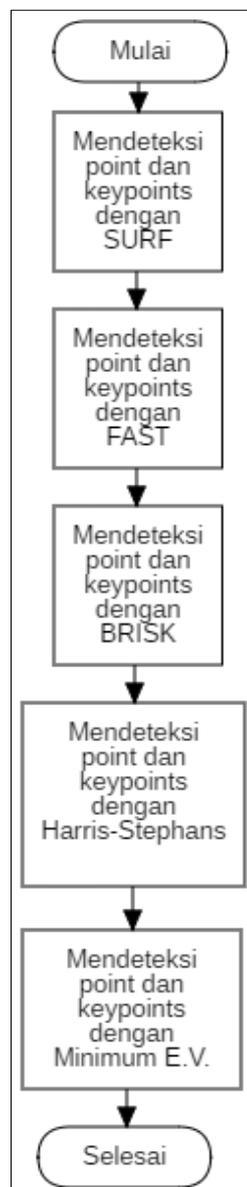
Gambar a. Flowchart Sub-Proses DCT



Gambar b. Flowchart Sub-Proses IDCT

E. Flowchart Mendeteksi Fitur Wajah 2D

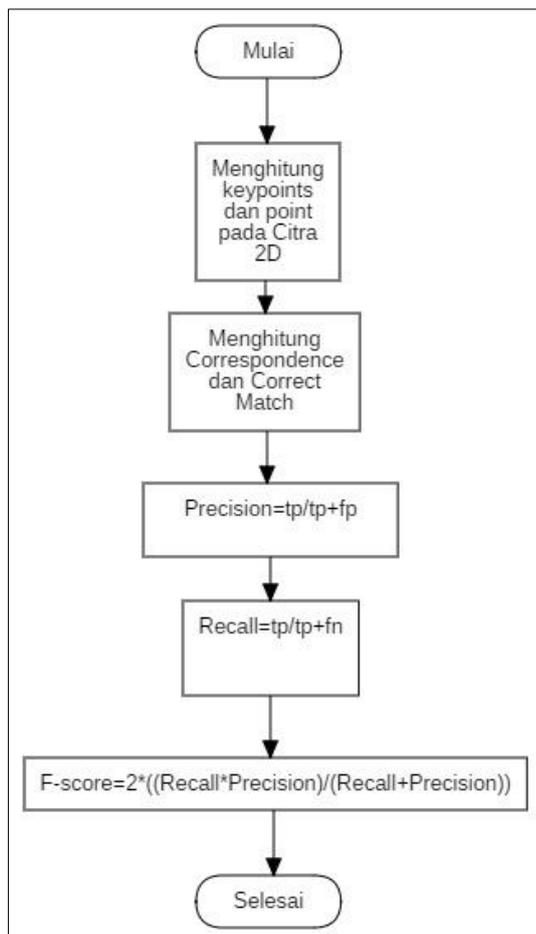
Flowchart pada gambar 3.5 menggambarkan langkah-langkah dalam mendeteksi *point-point* dan *keypoints* yang ada pada citra wajah kiri dan kanan. Setiap citra akan dideteksi menggunakan 4 detektor fitur yaitu, *Speeded Up Robust Features* (SURF), *Features From Accelerated Segment Test* (FAST), *Binary Robust Invariant Scalable Keypoints* (BRISK), *Harris-Stephans*, dan *Minimum Eigen Value*.



Gambar 3. 5 Mendeteksi Fitur Wajah 2D

F. Flowchart Menghitung Akurasi Pendeteksian

Flowchart Menghitung Akurasi Pendeteksian merupakan proses menghitung *point*, *keypoints*, *correct match*, *correspondences*, *recall*, *precision*, dan *f-score*. Berdasarkan hasil dari proses pendeteksian fitur pada *flowchart* sebelumnya maka *flowchart* pada gambar 3.6 ini menghitung jumlah *point* dan *keypoints* yang terdeteksi pada citra. Kemudian jumlah dari keduanya digunakan untuk melakukan perhitungan *correct match*, *correspondences*, *recall*, dan *precision*. Hasil dari *recall* dan *precision* tersebut digunakan untuk menghitung nilai *f-score* sehingga dapat dilihat seberapa besar akurasi yang dihasilkan pada keseluruhan proses pengujian.



Gambar 3. 6 Flowchart Menghitung Akurasi Pendeteksian