



### **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

Metode yang digunakan dalam tahap penelitian ini terdiri dari telaah literatur, menganalisa kebutuhan, pengerjaan program, uji coba dan evaluasi hasil yang didapatkan.

##### 1. Telaah Literatur

Tahap telaah literatur merupakan tahapan awal yang dilakukan untuk mengetahui permasalahan yang ada dan materi yang perlu dipelajari. Pada tahap ini, dilakukan analisis terhadap jurnal rujukan yang meneliti tentang citra yang diambil menggunakan metode geometri epipolar untuk mengetahui cara kerja dari metode yang digunakan. Kemudian dari hasil analisis tersebut dilakukan pembelajaran mengenai metode geometri epipolar, *ORL images database*, *Head Pose image database*, DCT, algoritma Wiener, *feature detectors*, dan F-Score. Materi pembelajaran didapatkan dari berbagai sumber seperti jurnal, buku, ataupun situs *online* yang menyediakan materi yang terkait dengan penelitian yang akan dilakukan.

##### 2. Pemilihan objek penelitian

Berdasarkan hasil pembelajaran telaah literatur yang telah dilakukan, maka akan dilakukan tahap seleksi citra yang akan digunakan sebagai objek penelitian dari *dataset* ORL dan Head Pose dengan kriteria tertentu. Tahap ini dilakukan agar hasil uji coba yang dihasilkan dapat memberikan hasil yang *valid* atau relevan.

##### 3. Analisis dan Perancangan

Pada tahap ini, dilakukan analisis terhadap jenis metode, objek penelitian, dan cara evaluasi yang digunakan dalam penelitian ini. Setelah mengidentifikasi segala jenis kebutuhan dalam penelitian ini, dilakukan perancangan sistem yang akan digunakan untuk menguji coba objek penelitian, kemudian hasil uji coba tersebut akan dievaluasi untuk dilakukan analisis kembali agar dapat dibuat sebuah kesimpulan.

#### 4. Pembangunan Program

Pada tahap ini dilakukan realisasi dari rancangan sistem yang telah dibuat dengan menggunakan bahasa pemrograman Python. Kemudian dibuat juga sebuah tampilan *interface* untuk mempermudah tahap ujicoba dan evaluasi berupa GUI yang dibangun dengan menggunakan *library* tkinter.

#### 5. Uji Coba dan Evaluasi

Program yang telah dibangun diuji coba dengan memberikan *input* 2 citra geometri epipolar sebagai objek penelitian, kemudian dari hasil uji coba tersebut akan dilakukan pengumpulan informasi berupa jumlah *keypoints* yang dideteksi, jumlah *facial keypoints*, jumlah *corresponding points*, dan jumlah *correct matches*. Data dari hasil uji coba tersebut kemudian dievaluasikan dengan menggunakan perhitungan dari F-Score, dan hasil dari F-Score tersebut akan dibandingkan dengan penelitian yang dilakukan sebelumnya untuk diketahui perubahan nilai yang terjadi.

#### 6. Penulisan Laporan dan Konsultasi

Tahap ini dilakukan untuk kebutuhan dokumentasi atas segala jenis proses yang telah dilakukan dalam penelitian kali ini.

### 3.2 Perancangan Sistem

#### A. Alur Kerja

Untuk melakukan uji coba dalam penelitian kali ini, dilakukan observasi terhadap nilai F-Score yang dihasilkan dari masing-masing subjek dalam citra yang berjumlah 10 orang (5 subjek dari ORL dan 5 subjek dari Head Pose) antara metode original, metode DCT dan metode DCT yang ditambahkan dengan algoritma Wiener. Sebelum uji coba dilakukan, dilakukan beberapa tahapan sebagai berikut:

1. Menyeleksi dan memilih 5 subjek dari masing-masing *dataset* ORL dan Head Pose berdasarkan kriteria yang telah ditentukan. Kemudian dari tiap subjek akan digunakan 2 citra yang diambil dari sisi kiri dan kanan subjek tersebut.
2. Melakukan *cropping* pada citra yang telah diseleksi menjadi ukuran  $m \times m$  ( $92 \times 92$  untuk citra dari *dataset* ORL dan  $165 \times 165$  untuk citra dari *dataset* Head Pose yang diambil dari sudut  $-30^\circ$  dan  $+30^\circ$ ) dan difokuskan ke bagian wajah dari tiap subjek citra yang digunakan.
3. Melakukan tahap *preprocessing* dengan menerapkan *grayscale* untuk menghilangkan warna pada tiap pasangan citra.
4. Menerapkan DCT pada citra yang telah *digrayscale* untuk dilakukan penghilangan komponen *high frequency* dari citra tersebut.
5. Mengaplikasikan Wiener Filtering pada citra yang telah diproses dengan DCT.
6. Melakukan pendeteksian *keypoints* untuk tiap pasangan citra yang telah diproses melalui metode DCT dan DCT-Wiener dengan menggunakan *feature detector*. Dalam penelitian kali ini digunakan 5 jenis *feature detector* yang berbeda dengan *threshold* atau ambang batas dalam mendeteksi *keypoints* seperti tabel berikut:

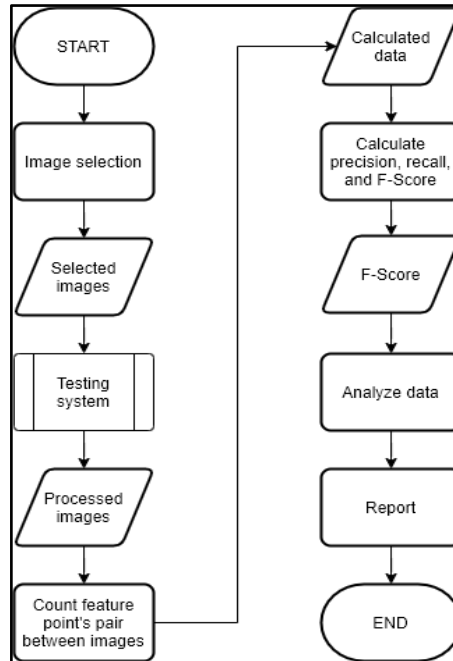
Tabel 3.1 Tabel Threshold Tiap Feature Detectors

Feature Detectors	Threshold
SURF	400
FAST	30
Harris-Stephens	0.5
BRISK	30
Minimum Eigenvalue	20

Penetapan nilai *threshold* tersebut diberlakukan untuk meminimalisir tingkat *human-error* atau kesalahan manusia dalam proses perhitungan *facial keypoints*, *correct matches*, dan *false pair* secara manual pada tiap citra yang diteliti.

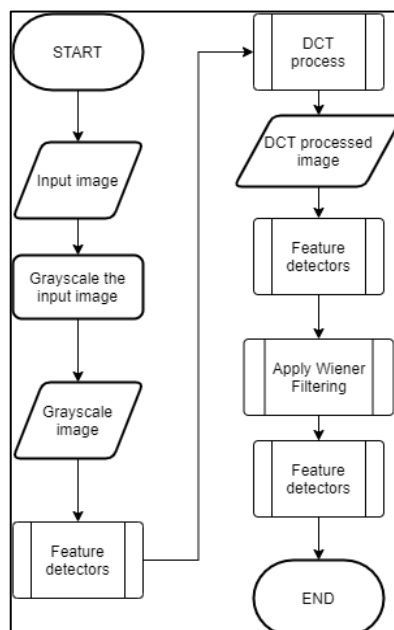
7. Data dari hasil pendeteksian tiap *feature detector* di tiap metodenya akan dilakukan penghitungan *facial keypoint*, *correct matches*, dan *false pair* secara manual, kemudian data-data tersebut disimpan ke dalam tabel dengan bantuan perangkat lunak berupa Microsoft Excel 2016 untuk melakukan perhitungan *precision*, *recall*, dan F-Score. Dalam penelitian kali ini, digunakan rumus *precision* dan *recall* yang telah diterapkan oleh Kusnadi, dkk (2019) seperti pada Rumus (2.19) dan (2.21). Berdasarkan rumus tersebut, *correct matches* merupakan total *facial keypoints* yang berhasil berkorespondensi dari citra kiri dan kanan dan *Total of All Matches* berupa total semua *keypoints* yang berhasil korespondensi dari citra kiri dan kanan. Kemudian nilai F-Score akan dikalkulasikan menggunakan rumus (2.18) yang merepresentasikan kualitas pengulangan (*repeatability*) pada penelitian ini.

Alur kerja dari keseluruhan tersebut dilakukan seperti pada Gambar 3.1.



Gambar 3.1 *Flowchart Alur Kerja Umum*

Kemudian terdapat alur kerja dari sistem uji coba yang digunakan dalam penelitian ini seperti pada Gambar 3.2.

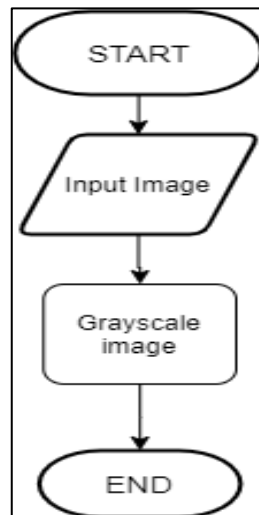


Gambar 3.2 *Flowchart Testing System*

Dalam penelitian ini citra akan diproses dengan tiga cara yang berbeda, yaitu citra hanya diproses dengan *grayscale* atau original, citra diproses dengan DCT,

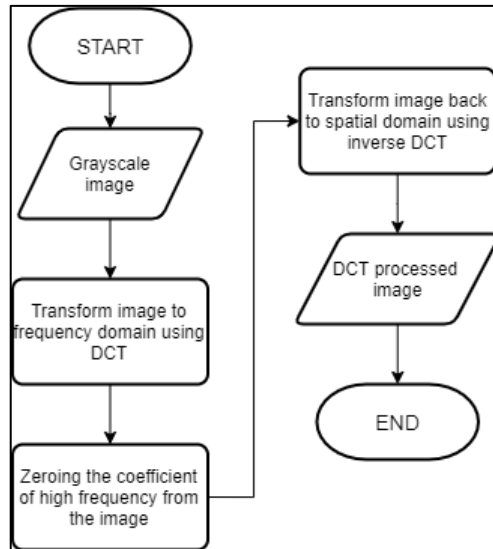
dan citra diproses dengan DCT dan diaplikasikan dengan Wiener Filtering. Proses kerja dari tiap metode dijelaskan dalam *flowchart* berikut ini:

1. Pada citra original, sebelum dilakukan proses pendeteksian fitur wajah citra hanya diproses menggunakan *grayscale* seperti pada Gambar 3.3.



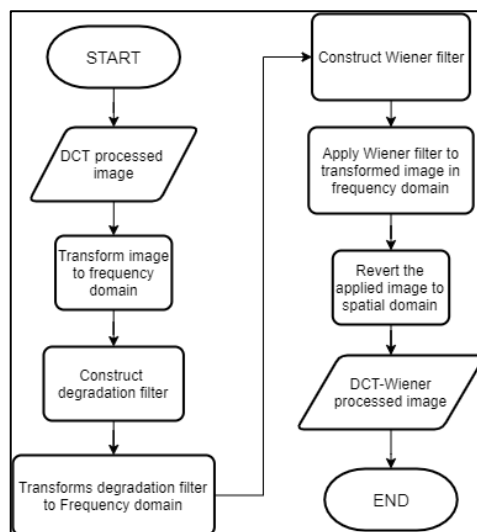
Gambar 3.3 *Flowchart* Citra Original

2. Ketika diproses dengan metode DCT, citra yang telah di *grayscale* diubah menjadi bentuk matriks untuk ditransformasikan ke domain frekuensi dengan metode DCT. Karena dalam penelitian kali ini hal yang diteliti adalah penghilangan komponen *high-frequency* dengan metode DCT, maka koefisien yang terletak di daerah komponen *high-frequency* diberikan nilai 0 sehingga hanya tersisa komponen *low* dan *mid frequency* dari citra tersebut. Matriks yang sudah dihilangkan komponen *high-frequency* ditransformasikan kembali ke domain *spatial* untuk digunakan sebagai objek penelitian.



Gambar 3.4 Flowchart Dengan Metode DCT

3. Pada metode DCT-Wiener dilakukan proses pengaplikasian filter Wiener yang telah dibentuk pada citra yang telah diproses dengan DCT. Pembentukan filter Wiener dibentuk berdasarkan rumus (2.3) dengan menggunakan *degradation filter* yang sudah dibentuk dan ditransformasikan ke dalam domain frekuensi. Filter Wiener yang telah dibentuk akan diaplikasikan ke citra yang sudah diproses dengan DCT dalam domain frekuensi menggunakan rumus (2.4). Citra yang telah diaplikasikan dengan Filter Wiener kemudian dikembalikan ke dalam domain spasial.



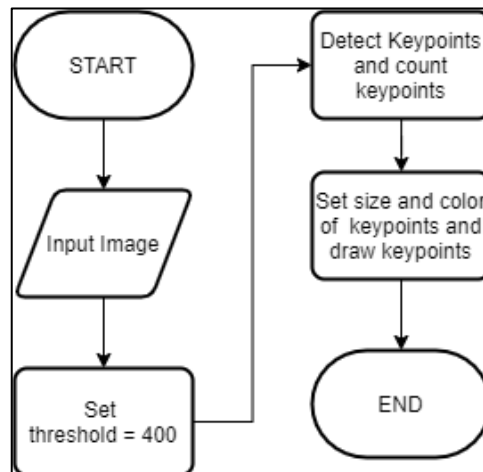
Gambar 3.5 Flowchart Dengan Metode DCT-Wiener



Hasil citra dari tiap proses kemudian akan diuji cobakan dengan 5 jenis *feature detector* untuk mendeteksi *keypoints* dari tiap citra. Adapun cara kerja dari kelima *feature detector* yang digunakan seperti pada *flowcharts* berikut:

## 1. SURF

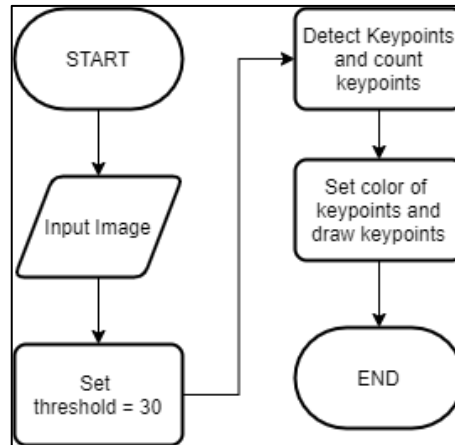
Pada *feature detector* SURF yang diimplementasikan dalam sistem pengujian ini dilakukan input citra kemudian dilakukan pendeteksian *keypoints* berdasarkan nilai *threshold* yang sudah ditentukan. Setelah itu, *keypoints* yang sudah terdeteksi akan digambarkan pada citra sesuai dengan ukuran dan warna yang telah ditentukan.



Gambar 3.6 Flowchart Feature Detector SURF

## 2. FAST

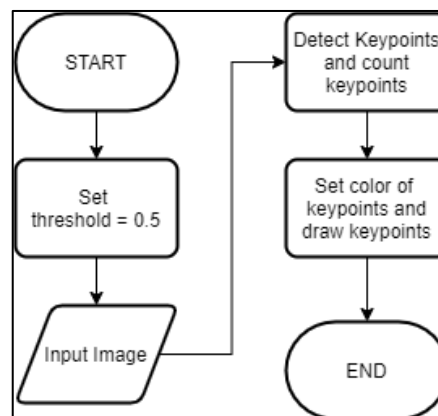
Pada *feature detector* FAST yang diimplementasikan dalam sistem pengujian ini dilakukan input citra kemudian dilakukan pendeteksian *keypoints* berdasarkan nilai *threshold* yang sudah ditentukan. Setelah itu, *keypoints* yang sudah terdeteksi akan ditampilkan pada citra sesuai dengan warna yang telah ditentukan.



Gambar 3.7 Flowchart Feature Detector FAST

### 3. Harris-Stephens

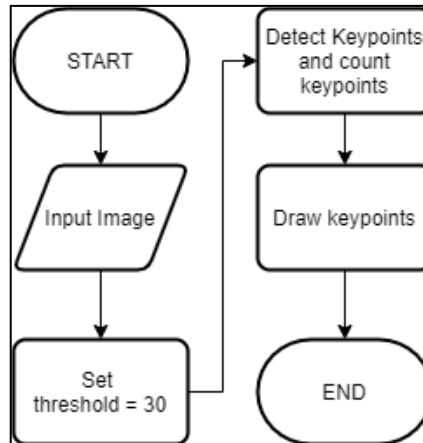
Pada *feature detector* Harris-Stephens yang diimplementasikan dalam sistem pengujian ini dilakukan input citra kemudian dilakukan pendeteksian *keypoints* berdasarkan nilai *threshold* yang sudah ditentukan. Setelah itu, *keypoints* yang sudah terdeteksi akan ditampilkan pada citra sesuai dengan warna yang telah ditentukan.



Gambar 3.8 Flowchart Feature Detector Harris-Stephens

### 4. BRISK

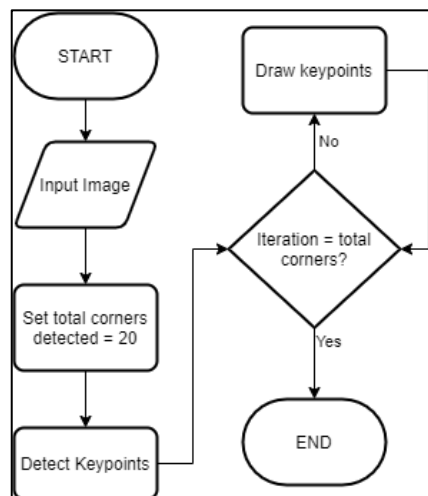
Pada *feature detector* BRISK yang diimplementasikan dalam sistem pengujian ini dilakukan input citra kemudian dilakukan pendeteksian *keypoints* berdasarkan nilai *threshold* yang sudah ditentukan. Setelah itu, *keypoints* yang sudah terdeteksi akan ditampilkan pada citra.



Gambar 3.9 Flowchart Feature Detector BRISK

## 5. Minimum Eigenvalue

Pada *feature detector* Minimum Eigenvalue yang diimplementasikan dalam sistem pengujian ini dilakukan input citra kemudian dilakukan pendeteksian *keypoints* sebanyak jumlah yang telah ditentukan. Setelah itu, *keypoints* yang sudah terdeteksi akan ditampilkan pada citra sesuai dengan warna yang telah ditentukan.



Gambar 3.10 Flowchart Feature Detector Minimum Eigenvalue

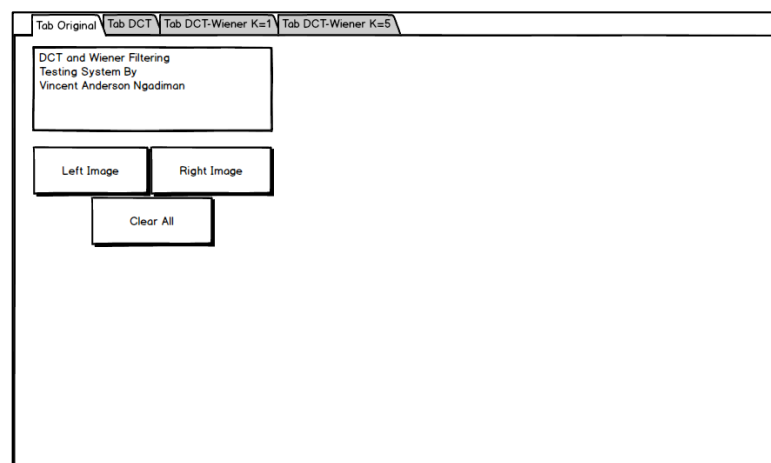
Setelah itu, dari pasangan citra dari seorang subjek akan dihitung jumlah *facial keypoints* yang terdeteksi dari total *keypoints* yang terdeteksi dari masing-masing citra secara manual. Kemudian, dari kedua citra tersebut dihitung jumlah *corresponding points* dan *correct pair* secara manual pada tiap jenis *feature detector*. Dari data yang didapatkan tersebut akan dievaluasi dengan menggunakan

perhitungan F-Score yang merepresentasikan kualitas pengulangan (*repeatability*) dari tiap pasangan citra yang telah melewati proses berbeda tersebut.

#### B. Rancangan Tampilan Antarmuka

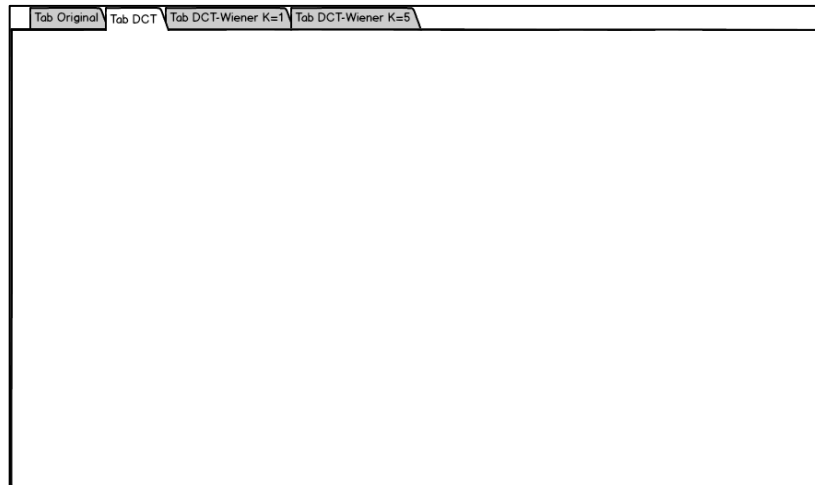
Pada tampilan antarmuka dari sistem uji coba ini, terdapat sebuah tombol untuk *user* menginput citra yang akan diuji coba. Kemudian setelah *user* menginput citra tersebut, akan ditampilkan hasil citra yang sudah diproses dengan tiap metodenya dan hasil citra setelah diaplikasikan dengan 5 jenis *feature detector* sehingga *user* dapat melakukan kalkulasi terhadap hasil yang didapat dan melakukan perhitungan F-Score untuk mengevaluasi hasil uji coba ini.

Gambar 3.11 merupakan rancangan tampilan antarmuka untuk halaman utama atau *tab original*, dimana *user* dapat melakukan input citra dan melakukan penghapusan hasil citra yang ada.



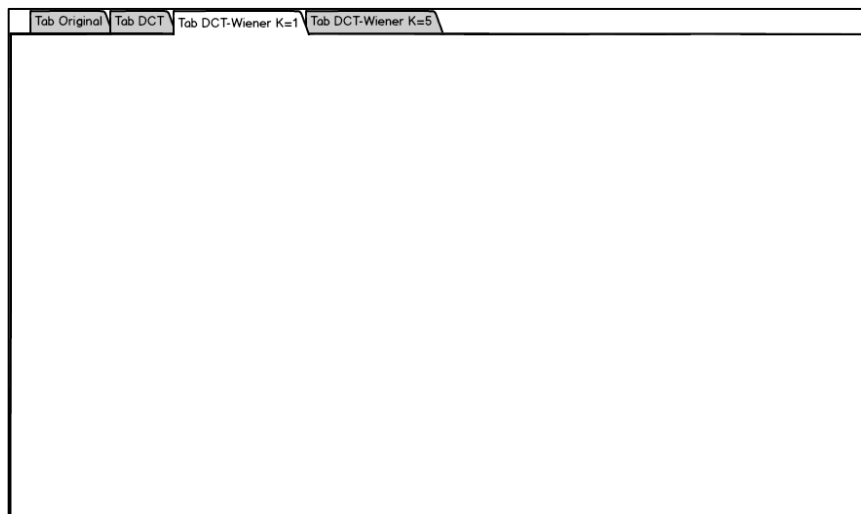
Gambar 3.11 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab Original* Sebelum Diberikan Input

Gambar 3.12 merupakan rancangan tampilan antarmuka pada halaman Tab DCT sebelum diberikan input, yang nantinya akan digunakan untuk menampung hasil citra yang telah diproses dengan DCT.



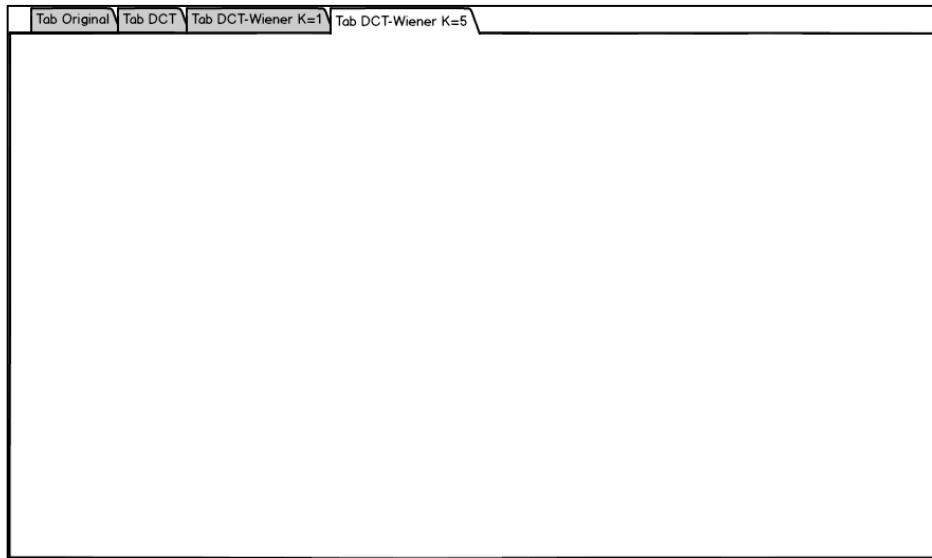
Gambar 3.12 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab DCT* Sebelum Diberikan Input

Gambar 3.13 merupakan rancangan tampilan antarmuka pada halaman Tab DCT-Wiener K=1 sebelum diberikan input, yang nantinya akan digunakan untuk menampung hasil citra yang telah diproses dengan DCT-Wiener K=1.



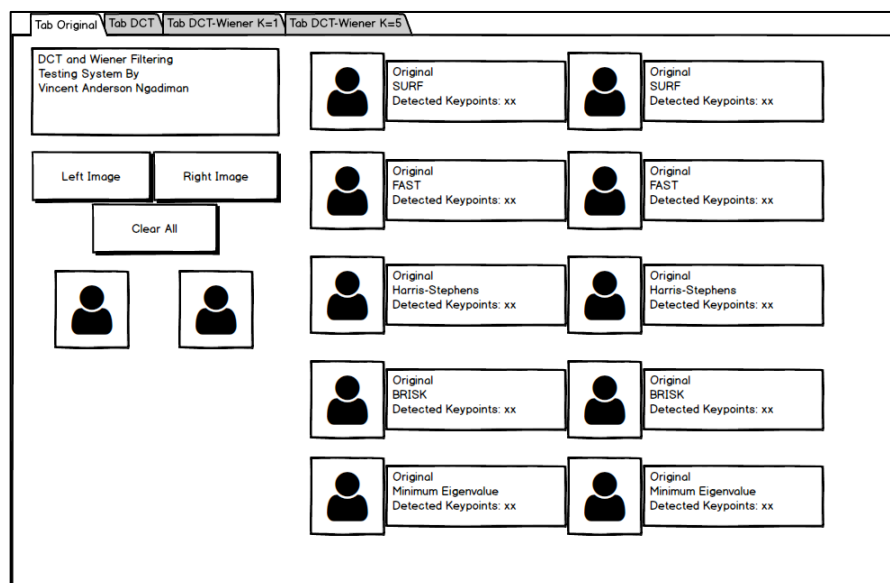
Gambar 3.13 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab DCT-Wiener K=1* Sebelum Diberikan Input

Gambar 3.14 merupakan rancangan tampilan antarmuka pada halaman Tab DCT-Wiener K=5 sebelum diberikan input, yang nantinya akan digunakan untuk menampung hasil citra yang telah diproses dengan DCT-Wiener K=5.



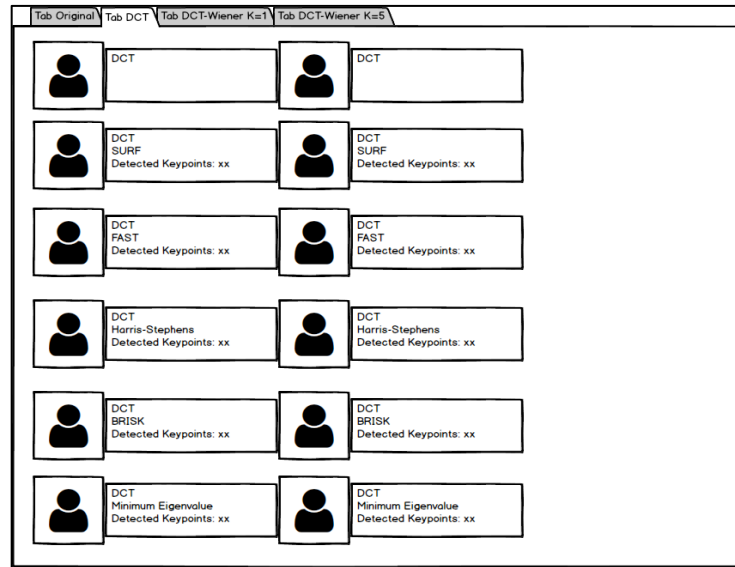
Gambar 3.14 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab DCT-Wiener K=5* Sebelum Diberikan Input

Gambar 3.15 merupakan rancangan tampilan antarmuka pada halaman utama atau *tab original* setelah diberikan input oleh *user*. Hasil citra yang ditampilkan di halaman ini adalah citra yang hanya diproses dengan *grayscale*.



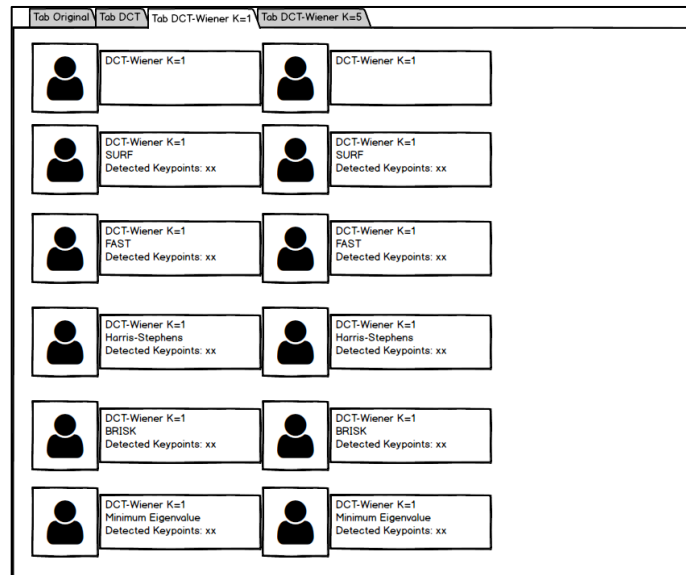
Gambar 3.15 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab Original* Sesudah Diberikan Input

Gambar 3.16 merupakan rancangan tampilan antarmuka pada halaman *tab DCT* yang digunakan untuk menampilkan hasil citra yang diproses dengan metode DCT.



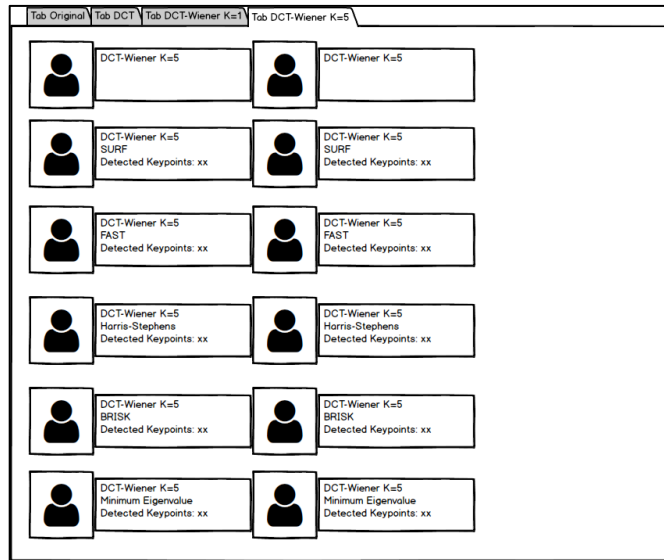
Gambar 3.16 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab* DCT Setelah Diberikan Input

Gambar 3.17 merupakan rancangan tampilan antarmuka pada halaman *tab* DCT-Wiener K=1 yang digunakan untuk menampung hasil citra yang diproses dengan metode DCT-Wiener K=1.



Gambar 3.17 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab* DCT-Wiener K=1 Setelah Diberikan Input

Gambar 3.18 merupakan rancangan tampilan antarmuka pada halaman *tab* DCT-Wiener K=5 yang digunakan untuk menampung hasil citra yang diproses dengan metode DCT-Wiener K=5.



Gambar 3.18 *Mockup* dari Tampilan Antarmuka Pada Halaman *Tab* DCT-Wiener K=5 Setelah Diberikan Input