



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 DAN PERANCANGAN APLIKASI

3.1 Metodologi Penelitian

Pada penelitian ini, tahapan pengerjaan dilakukan mengikuti tahapan *Software Development Lifecycle* (SDLC):

1. Telaah Literatur

Tahapan dilakukan untuk mempelajari teori-teori serta cara kerja yang terkait dengan penelitian ini, yaitu: *Histogram Equalization*, *DCT*, *Feature Detectors*, *F-Measure*, *Key Points*, Pengulangan (*Repeatability*), dan Geometri Epipolar. Pembelajaran dilakukan dengan membaca *e-journal*, halaman web, dan *e-book* yang terkait.

2. Perancangan & Pengumpulan Data

Pada tahapan ini, dibuat *Flowchart* yang digunakan untuk membantu menjelaskan tahapan-tahapan kerja dalam sistem serta membuat perancangan antarmuka untuk pembuatan tampilan dari hasil yang telah didapat dari sistem.

Kemudian, data berupa gambar dikumpulkan dari dua dataset yang sudah ditentukan, yaitu: *Head Pose Image Dataset* dan *The Database of Faces*. Setelah itu, dilakukan proses penyortiran untuk memilih gambar subjek yang sesuai dengan batasan masalah dan terakhir, melakukan *cropping* pada gambar yang dipilih. Tujuan dari *cropping* adalah untuk mengurangi bagian daerah yang tidak akan menghasilkan *keypoint*.

3. Pembangunan Sistem

Pada tahapan ini, sistem untuk uji coba beserta tampilan antar muka dibangun sesuai dengan rancangan yang telah dibuat sebelumnya. Pembangunan sistem dilakukan dengan menggunakan bahasa pemrograman Python. Sistem yang dibuat akan mengaplikasikan DCT untuk menghilangkan komponen *low frequency*, *Histogram Equalization*, dan lima *feature detector* untuk melakukan deteksi *keypoint*.

4. Uji Coba

Gambar kemudian dimasukkan ke dalam sistem dan sistem akan mengeluarkan hasil berupa gambar yang telah ditandai *keypoints*-nya. Kemudian, perhitungan atas jumlah *keypoint* yang telah didapat dilakukan dan melakukan evaluasi nilai *F-score* berdasarkan perhitungan yang dilakukan.

5. Analisis Hasil

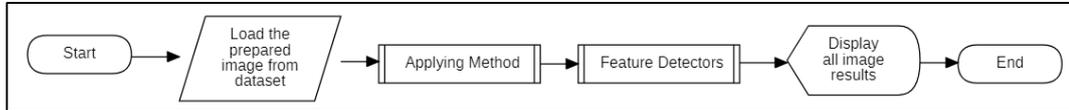
Setelah hasil didapatkan, maka dilakukan analisis atas faktor-faktor yang menyebabkan sistem dapat mengeluarkan hasil tersebut serta menarik kesimpulan dari penelitian yang telah dilakukan.

3.2 Perancangan Sistem

Dalam tahapan ini, dibuat *Flowchart* untuk mendeskripsikan tahapan pengerjaan di dalam sistem dan rancangan antarmuka yang digunakan sebagai acuan dalam membuat tampilan dari hasil yang didapat.

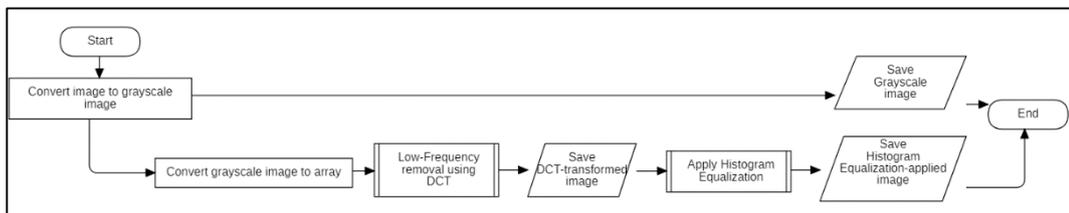
3.2.1 Flowchart

Alur dari sistem adalah sebagai berikut ini:



Gambar 3.1 *Flowchart Sistem*

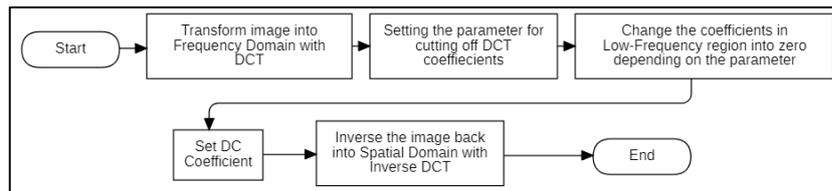
Gambar 3.1 mendeskripsikan alur secara garis besar dari sistem yang dibuat pada penelitian ini. Gambar yang telah dipilih dari Head Pose Image Dataset dan The Database of Faces akan dimasukkan ke dalam sistem dan langkah pertama adalah mengaplikasikan algoritma yang dipakai yang dimana terdiri dari mentransformasikan gambar menjadi *grayscale* dan memasukkan gambar ke dalam *function* DCT dan *Histogram Equalization*. Kemudian, gambar tersebut akan kembali dijalankan di dalam *function* *Feature Detectors*. Langkah terakhir adalah menampilkan semua gambar yang didapatkan melalui sistem ini.



Gambar 3.2 *Flowchart Applying Method*

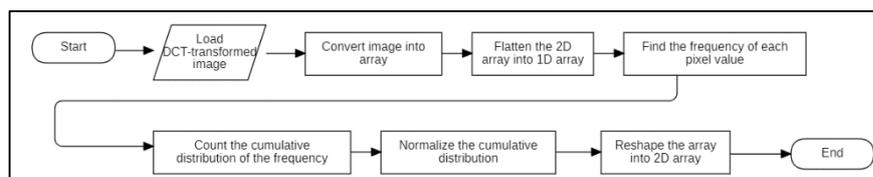
Gambar 3.2 menjelaskan alur dari cara pengaplikasian metode. Dalam langkah ini terdapat dua alur, yaitu: mengaplikasikan DCT & *Histogram Equalization* dan tidak mengaplikasikan metode apapun (original). Jika tidak diaplikasikan apapun, maka gambar yang sudah diubah menjadi gambar *grayscale* akan langsung disimpan. Di alur yang lain, gambar yang telah dikonversi menjadi *grayscale* akan menjadi *array* yang berisi intensitas dari setiap piksel di rentang 0-255. *Array* ini akan dimasukkan ke *function* DCT untuk dilakukan penghilangan

koefisien di *low frequency*. Proses ini akan menghasilkan gambar baru yang kemudian akan disimpan. Lalu, gambar akan kembali diproses dengan *Histogram Equalization* untuk ditingkatkan kontrasnya dan langkah ini juga menghasilkan gambar yang kemudian juga akan disimpan.



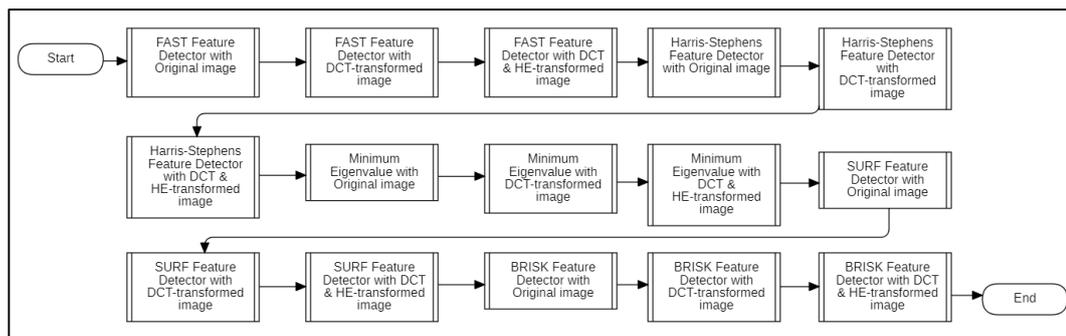
Gambar 3.3 *Flowchart Low Frequency Removal Using DCT*

Gambar 3.3 menjelaskan bagaimana koefisien *low frequency* dihilangkan dengan menggunakan DCT. Gambar yang sebelumnya telah ditransformasi menjadi bentuk *array* adalah gambar yang masih berada di *Spatial Domain* dan langkah pertama dari bagian ini adalah dengan mentransformsikan gambar tersebut ke dalam *Frequency Domain* dengan DCT. Kemudian, dipasang sebuah parameter yang akan berfungsi di langkah selanjutnya. Langkah selanjutnya adalah mengganti koefisien yang berada di *low-frequency* dengan nol (0) yang merupakan langkah untuk menghilangkan *low-frequency* dari gambar dan mengatur nilai di koefisien DC. Parameter yang dipasang dilangkah sebelumnya berfungsi untuk menentukan seberapa banyak koefisien *low-frequency* akan dihilangkan di daerah yang merupakan bagian dari *low-frequency*. Setelah komponen *low-frequency* dihilangkan, maka gambar akan dikembalikan ke *Spatial Domain* dengan menjalankan *Inverse DCT*.



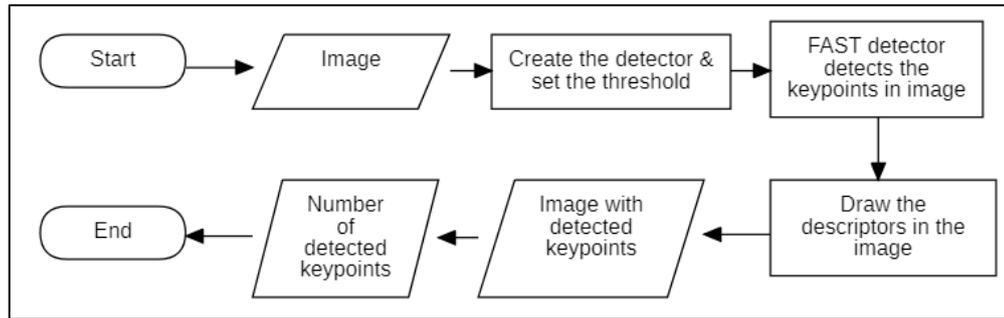
Gambar 3.4 *Flowchart Histogram Equalization*

Gambar 3.4 menjelaskan alur dari *Histogram Equalization*. Pada langkah ini, gambar yang telah dihilangkan komponen *low frequency* akan dimasukkan kembali untuk dijalankan *Histogram Equalization*. Gambar ini kemudian akan ditransformasikan ke dalam bentuk *array*. Lalu, *array* yang memiliki bentuk 2D ini akan ditransformasikan ke dalam bentuk 1D. Setelah itu, *function* akan melakukan perhitungan frekuensi atas kemunculan dari nilai intensitas di setiap *piksel*. Perhitungan dilanjutkan dengan menghitung distribusi kumulatif dari frekuensi yang didapatkan dan melakukan normalisasi. Kemudian, *array* ini akan dikembalikan ke dalam bentuk 2D agar dapat kembali dibaca dalam bentuk gambar.



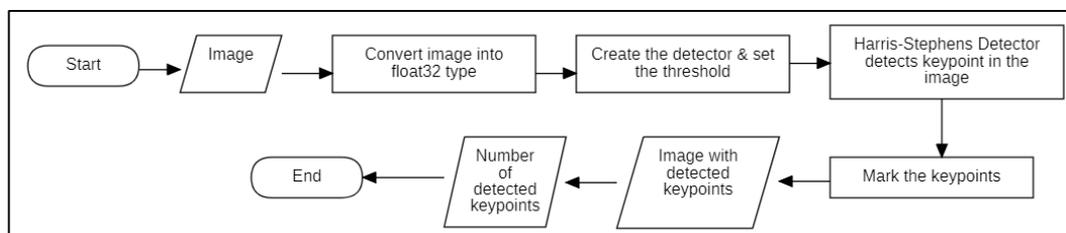
Gambar 3.5 *Flowchart Feature Detector*

Gambar 3.5 menjelaskan bagaimana *Feature Detector* dijalankan. Secara garis besar, setiap *feature detector* dijalankan minimal tiga kali karena terdapat gambar yang tidak diaplikasikan metode apapun, hanya ditransformasikan dengan DCT dan gambar yang ditransformasikan dengan DCT serta *Histogram Equalization*. Dalam penelitian ini, *feature detector* FAST, Harris-Stephens, Minimum Eigenvalue (Shi-Tomasi), SURF, dan BRISK digunakan untuk melakukan deteksi *keypoints* di gambar yang telah diproses. Semua isi dari *function* adalah sama, yang membedakan hanyalah gambar yang dimasukkan ke dalam *function* tersebut.



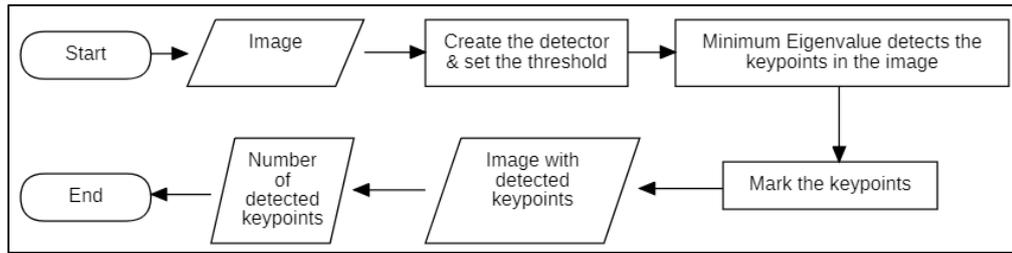
Gambar 3.6 *Flowchart FAST Feature Detector*

Gambar 3.6 menjelaskan bagaimana FAST dijalankan. Setelah gambar dimasukkan ke dalam *function*, maka *detector* FAST dibuat dan dimasukkan besaran nilai ambang batas (*threshold*) yang digunakan untuk *detector* ini. Lalu, FAST akan melakukan deteksi *keypoint* dan kemudian menandai bagian yang merupakan *keypoint*. *Function* akan menghasilkan gambar yang ditandai *keypoint*-nya dan jumlah *keypoint* yang berhasil dideteksi.



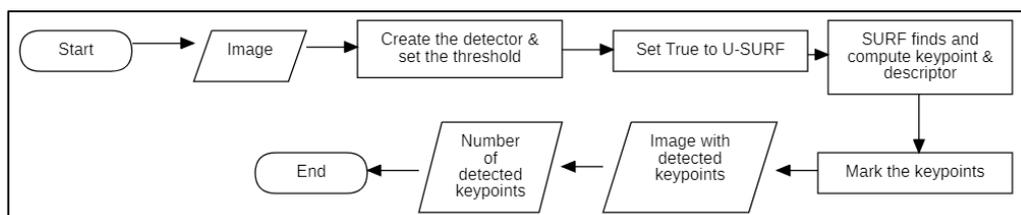
Gambar 3.7 *Flowchart Harris-Stephens Feature Detectors*

Gambar 3.7 menjelaskan bagaimana Harris-Stephens dijalankan pada gambar. Gambar yang telah dimasukkan kemudian dikonversi ke dalam bentuk Float32. Setelah itu, *detector* Harris-Stephens dibuat dan dimasukkan nilai yang akan menjadi ambang batas (*threshold*) untuk *detector* ini. Kemudian, *detector* Harris-Stephens akan melakukan deteksi *keypoint*. Lalu, bagian yang merupakan *keypoint* akan ditandai dan pada akhirnya akan menghasilkan gambar yang telah ditandai *keypoint*-nya serta jumlah *keypoint* yang berhasil dideteksi.



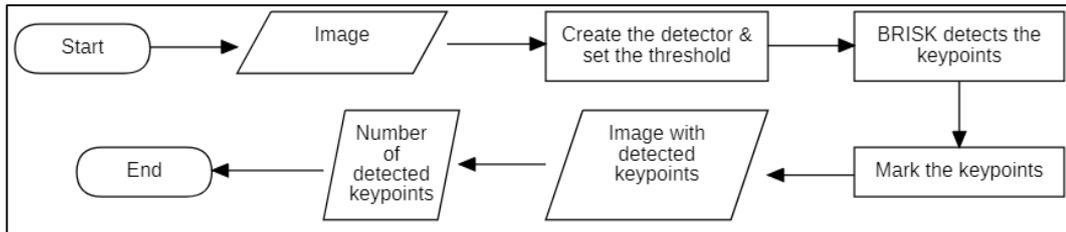
Gambar 3.8 *Flowchart Minimum Eigenvalue Feature Detectors*

Gambar 3.8 menjelaskan bagaimana Minimum Eigenvalue dijalankan pada gambar. Setelah gambar dimasukkan, *detector* Minimum Eigenvalue dibuat dan ditentukan nilai ambang batas (*threshold*) yang akan digunakan. Lalu, Minimum Eigenvalue akan mendeteksi *keypoint* di gambar. Setelah itu, *keypoint* akan ditandai dan akan dihasilkan gambar yang telah ditandai bagian *keypoint*-nya dan jumlah *keypoint* yang berhasil dideteksi.



Gambar 3.9 *Flowchart SURF Feature Detector*

Gambar 3.9 menjelaskan bagaimana SURF dijalankan pada gambar. Setelah gambar dimasukkan, *detector* SURF akan dibuat dan dimasukkan nilai ambang batas (*threshold*) yang akan digunakan. Langkah selanjutnya adalah mengaktifkan U-SURF (*Upright SURF*) karena *keypoint* yang diinginkan tidak perlu diketahui arahnya. Jika tidak diaktifkan, maka SURF akan memunculkan hasil deteksi *keypoint* beserta arahnya. Setelah U-SURF diaktifkan, maka SURF akan melakukan deteksi *keypoint* dan dilanjutkan dengan membuat *descriptor*-nya. Kemudian, *keypoint* akan ditandai. Pada akhirnya, akan dikembalikan berupa gambar yang telah ditandai *keypoint*-nya dan jumlah *keypoint* yang berhasil dideteksi.

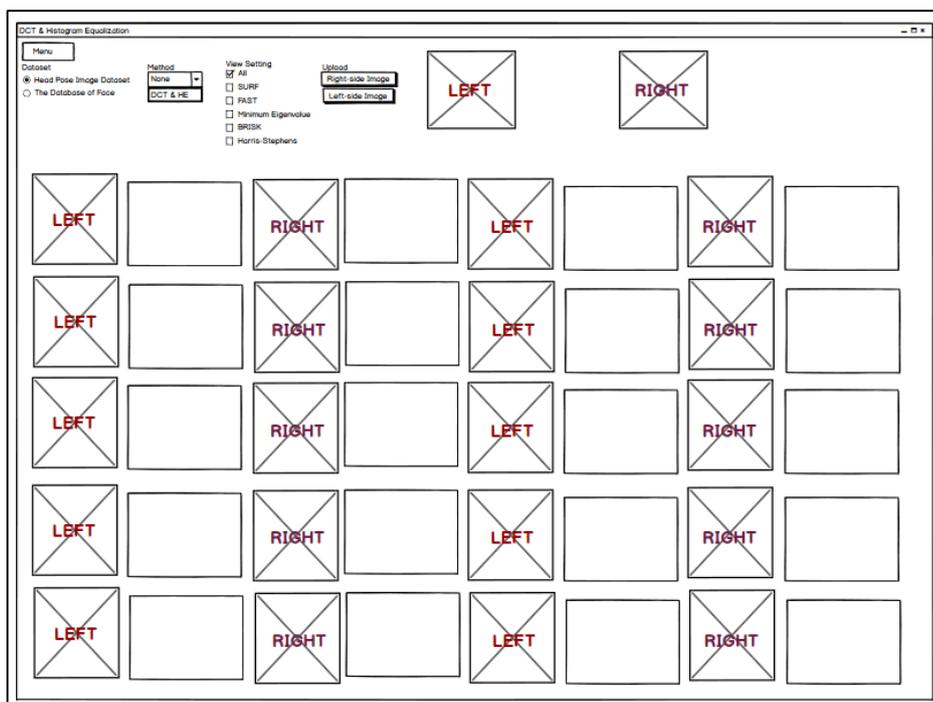


Gambar 3.10 *Flowchart BRISK Feature Detector*

Gambar 3.10 menjelaskan bagaimana BRISK diaplikasikan pada gambar. Setelah gambar dimasukkan, *detector* BRISK dibuat dan dimasukkan nilai ambang batas (*threshold*) yang akan digunakan. Lalu, BRISK akan melakukan deteksi *keypoint* dan kemudian *keypoint* yang telah dideteksi akan ditandai. Gambar yang telah ditandai *keypoint*-nya akan dikembalikan di akhir beserta jumlah *keypoint* yang berhasil dideteksi.

3.2.2 Rancangan Antarmuka

Gambar 3.11 menunjukkan rancangan antarmuka untuk menampilkan perbandingan hasil dari metode dan hasil dari *feature detectors*.

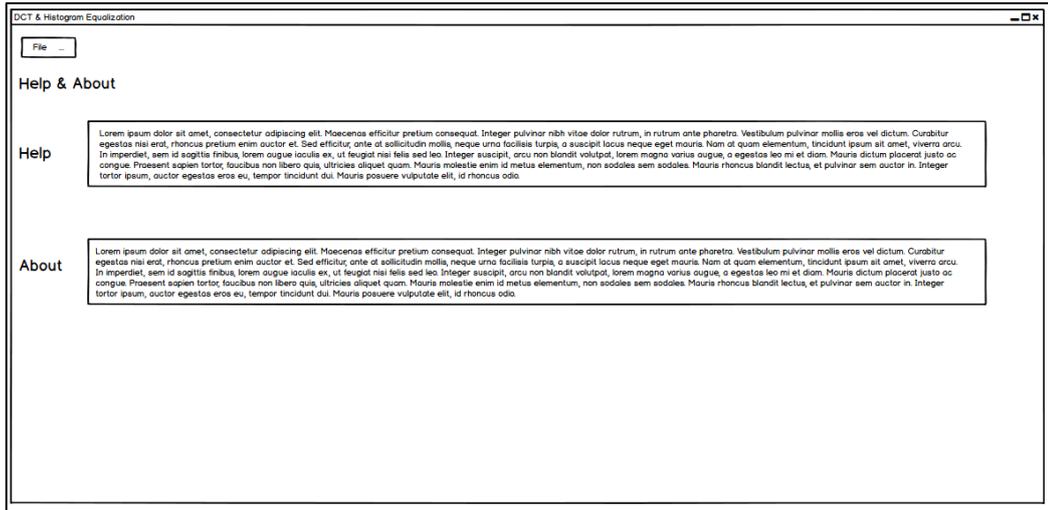


Gambar 3.11 Rancangan Antarmuka 1

Pada Gambar 3.11, terdapat empat menu yaitu: *Dataset*, *Method*, *View Setting*, dan *Upload*. *Dataset* digunakan untuk menyatakan asal dari *dataset* yang dipakai. *Method* digunakan untuk menentukan metode yang akan diaplikasikan dari dua pilihan, yaitu: *None* (tidak mengaplikasikan metode apapun) dan *DCT & HE* (mengaplikasikan *DCT* dan *Histogram Equalization*). *View Setting* digunakan untuk memilih banyaknya gambar yang harus ditampilkan serta *Upload* digunakan untuk memasukkan gambar ke dalam aplikasi. Bagian bawah dari grup menu digunakan untuk menampilkan hasil gambar beserta deskripsi metode dan jumlah *keypoint* yang di dapat dari gambar tersebut.

Pada Gambar 3.11 juga, tombol Menu yang berada di ujung kiri atas berisi beberapa *submenu*, yaitu: *Main Menu*, *Help & About*, dan *Reset*. *Main Menu* digunakan untuk kembali ke halaman awal yang ditunjukkan pada Gambar 3.11. *Help & About* digunakan untuk menuju halaman *Help & About* yang ditunjukkan di Gambar 3.12 sementara *Reset* digunakan untuk menghapus semua gambar baik yang berada dimasukkan melalui tombol di grup menu *Uplaod* maupun yang gambar hasil proses yang berada di sebelah bawah.

Gambar 3.12 menunjukkan rancangan antarmuka pada halaman *Help & About*. Pada halaman ini, penjelasan mengenai *Help & About* dipisahkan dalam dua *text box*. Bagian *Help* mendeskripsikan cara pemakaian aplikasi dan *About* mendeskripsikan informasi seputar aplikasi. Dalam halaman ini juga terdapat tombol Menu yang berada di ujung kiri atas yang dapat digunakan untuk kembali ke halaman utama.



Gambar 3.12 Rancangan Antarmuka 2