



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.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Pada bagian ini akan dijelaskan metode yang digunakan dalam penelitian ini. Metode-metode yang digunakan adalah studi literatur, analisis sistem, perancangan sistem, pemograman sistem, testing.

3.1.1 Studi Literatur

Tahap studi literatur dilakukan untuk mempelajari tahap-tahap yang dibutuhkan dalam melakukan pemodelan objek tiga dimensi dan mempelajari algoritma *Direct Linear Transformation*. Pembelajaran tersebut dilakukan dengan membaca dari beberapa sumber, di antaranya buku, makalah, jurnal, dan berbagai sumber online.

3.1.2 Analisis Sistem

Pada tahap analisis sistem ini akan dilakukan analisis pada algoritma *Direct Linear Transformation* yang diimplementasikan pada pemodelan tiga dimensi serta melakukan analisis terhadap tahap-tahap dalam melakukan pemodelan tiga dimensi.

3.1.3 Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem mencakup pembuatan *flowchart* dan desain antarmuka.

3.1.4 Pemrograman Sistem

Pada tahap ini sistem dibuat dengan menggunakan bahasa pemrograman C# dan mengimplementasikan *library face detection*, *face recognition* serta algoritma *Direct Linear Transformation*.

3.1.5 Pengujian

Pada tahap ini sistem akan diuji coba tingkat keberhasilan dengan cara melakukan percobaan dalam sisi ketepatan dan kecepatan.

A. Ketepatan

Pengujian ketepatan dalam penelitian ini dilakukan dengan menggunakan *False Acceptance Rate (FAR)* dan *False Reject Rate (FRR)*. Pengujian tersebut didapatkan dari 20 kali percobaan, yang didapatkan dari 2 kali percobaan dari setiap 10 wajah yang berbeda.

B. Kecepatan

Pada proses ini akan dilakukan perhitungan waktu yang dibutuhkan untuk membaca wajah dengan menggunakan pendekatan *face recognition 3D* dengan penelitian sebelumnya.

3.2 Analisis Sistem

Aplikasi ini memiliki 2 menu utama yaitu menu *Face Recognition* dan menu *Add New Face*. Menu *face recognition* merupakan proses melakukan proses pengenalan wajah dengan menggunakan *library neural network*. Proses *face recognition* tersebut memiliki tiga menu yaitu menu *training* atau pelatihan, menu *testing* atau pengujian, dan menu *back* yang berfungsi untuk kembali ke menu sebelumnya. Menu pelatihan berfungsi untuk melatih *neural network* pada sistem sehingga dapat mengenal wajah masukan dengan lebih akurat dan menu pengujian

berfungsi untuk melakukan pengujian terhadap data masukan berupa citra wajah dari kamera kiri dan kamera kanan dengan melakukan kalibrasi kamera terlebih dahulu. Menu kedua pada aplikasi ini adalah menu *Add New Face* merupakan menu yang bertujuan untuk menambahkan wajah baru kedalam *database*. Terdapat 3(tiga) langkah utama yang terdapat pada menu tersebut. Langkah pertama adalah kalibrasi kamera yang bertujuan untuk mendapatkan parameter intrinsik dan ekstrinsik yang dimiliki oleh kamera, untuk mendapatkan parameter-parameter tersebut gambar papan catur dengan ukuran yang telah ditentukan (7x6 adalah ukuran yang digunakan pada penelitian ini) dibutuhkan sebagai citra masukan. Langkah selanjutnya yaitu kedua adalah pemasukan citra wajah pada kamera kiri dan kamera kanan, sistem akan mendeteksi wajah pada citra wajah masukan tersebut serta melakukan pendeteksian titik-titik dua dimensi wajah pada citra tersebut dengan menggunakan *library FaceLuxand SDK*. Langkah terakhir yaitu langkah ketiga merupakan proses perhitungan titik-titik dua dimensi wajah pada citra wajah dengan menggunakan algoritma *Direct Linear Transformation*. Hasil perhitungan *Direct Linear Transformation* yang dijalankan oleh sistem berupa titik-titik tiga dimensi dari wajah yang akan disimpan dalam *database* dan akan digunakan pada proses pengenalan wajah dengan menggunakan *library neural network* yang bertujuan sebagai *biometric security* pada aplikasi login.

3.3 Perancangan Sistem

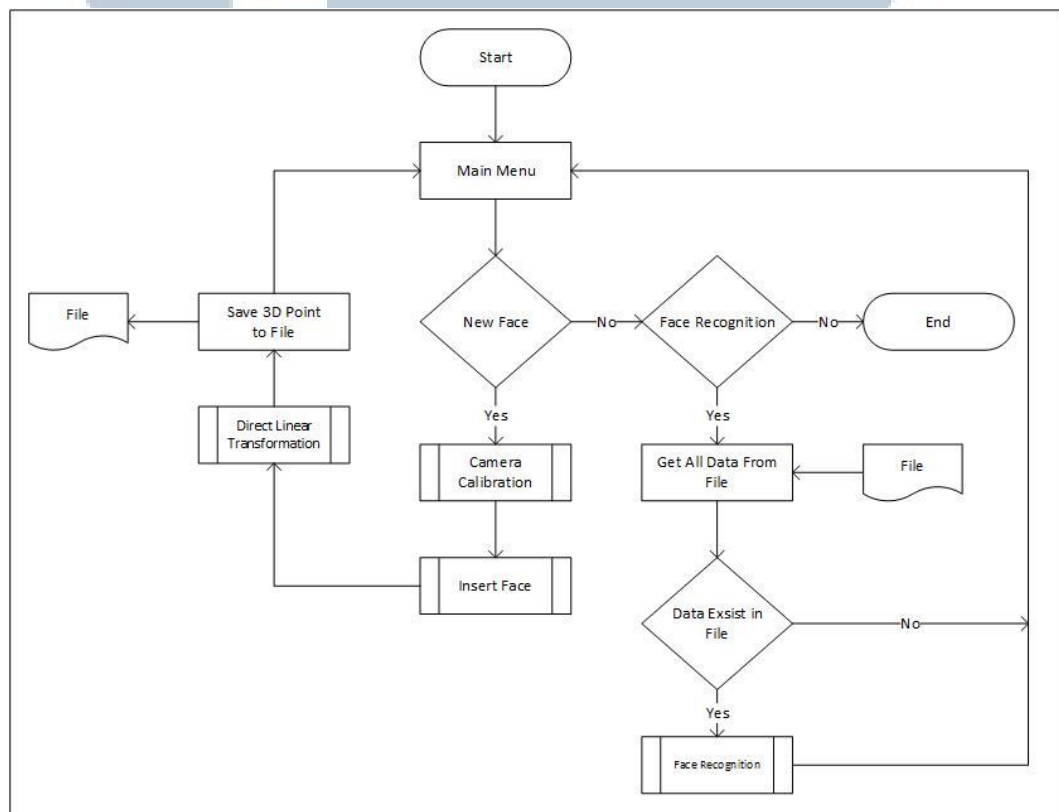
Pada proses perancangan sistem akan dijelaskan mengenai beberapa rancangan yang dibagi menjadi dua bagian yaitu perancangan *flowchart* dan perancangan antarmuka.

3.3.1 Flowchart

Flowchart digunakan pada perancangan sistem dengan tujuan untuk menjelaskan aliran sistem atau algoritma, dan pada perancangan *flowchart* pada sistem dibagi menjadi beberapa *flowchart* yang terdiri dari *flowchart* sistem secara keseluruhan, *flowchart* deteksi wajah, *flowchart* kalibrasi kamera, *flowchart direct linear transformation*, dan *flowchart* pengenalan wajah.

A. Flowchart Sistem

Pada bagian ini akan dijelaskan alur proses yang terdapat pada sistem yang akan dibuat seperti pada Gambar 3.1.



Gambar 3.1 Flowchart Sistem

Sistem memiliki pilihan menu yang akan ditampilkan pada saat aplikasi dibuka. Pilihan menu tersebut meliputi pemasukan data wajah dan pengenalan

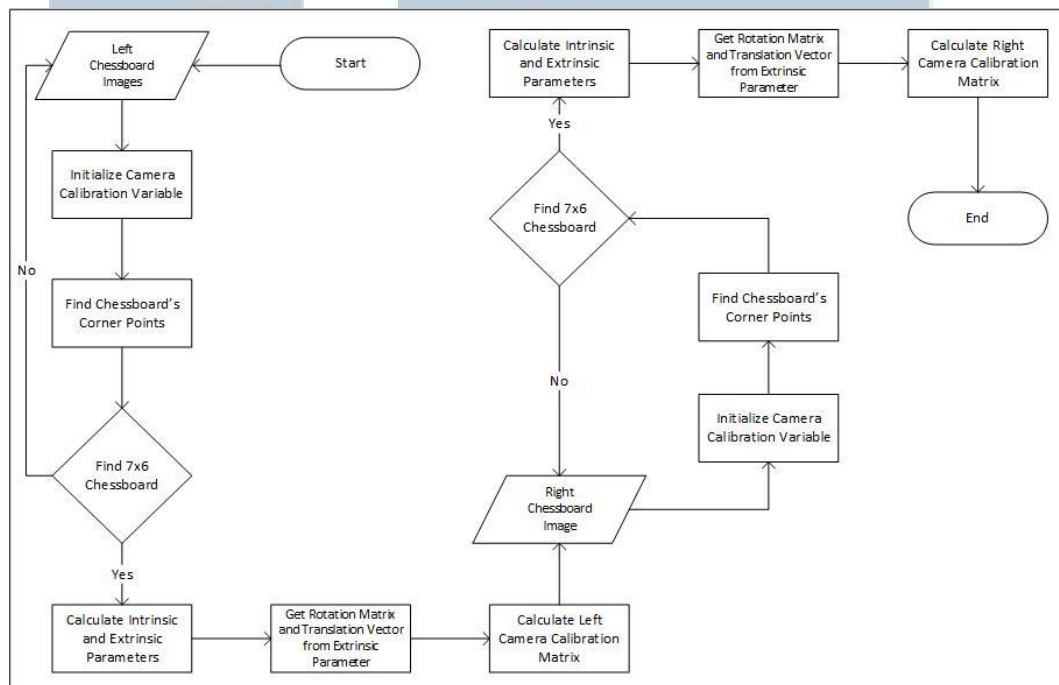
wajah. Pada bagian pengenalan wajah, langkah pertama yang dilakukan oleh sistem adalah mendapatkan semua data yang di peroleh dari *file* penyimpanan dan memasukan semua data tersebut ke dalam sebuah *array*. Setelah semua data dimasukkan ke dalam sebuah *array*, selanjutnya sistem akan melakukan pengecekan terhadap isi dari *array* tersebut. Jika *array* tersebut tidak terdapat data maka sistem akan kembali ke menu utama, tetapi jika *array* tersebut terdapat data maka sistem akan melakukan proses pengenalan wajah dengan menggunakan *library neural network*.

Pada bagian pemasukan data wajah sistem akan meminta user untuk melakukan tahap kalibrasi kamera terlebih dahulu. Tahap tersebut membutuhkan input berupa gambar papan catur dengan ukuran yang telah ditentukan (7x6 adalah ukuran yang digunakan pada penelitian ini) dan diambil dengan menggunakan 2 (dua) buah kamera yang akan digunakan pada pengambilan citra wajah. Proses tersebut bertujuan menentukan parameter *intrinsic* dan *extrinsic* pada kedua kamera yang digunakan. Langkah selanjutnya adalah memasukan data 2 (dua) buah citra wajah yang telah diambil dengan menggunakan kamera, yang telah diatur jarak antara kamera ke wajah. Data dari gambar tersebut di terima oleh sistem dan selanjutnya sistem akan melakukan proses pendeteksian wajah dengan menggunakan *library FaceLuxand SDK*, jika wajah terdeteksi pada citra masukan maka sistem akan melakukan pendeteksian titik-titik wajah pada kedua citra tersebut dengan menggunakan *library FaceLuxandSDK*. Tepat setelah sistem dapat mendeteksi titik-titik wajah, sistem akan melakukan proses perhitungan titik-titik tiga dimensi dengan memanfaatkan parameter *intrinsic* dan *extrinsic* serta titik-titik dua dimensi dari citra wajah masukan. Proses perhitungan titik-titik tersebut

dijalankan sistem dengan menggunakan algoritma *Direct Linear Transformation* dan selanjutnya data hasil perhitungan tersebut akan dimasukkan ke dalam *file* penyimpanan yang akan digunakan pada proses pengenalan wajah.

B. Flowchart Camera Calibration

Pada *flowchart camera calibration* menjelaskan alur dari sistem untuk mendapatkan matriks kalibrasi kamera yang akan digunakan pada proses algoritma *Direct Linear Transformation (DLT)*. Proses *flowchart* tertera pada Gambar 3.2.



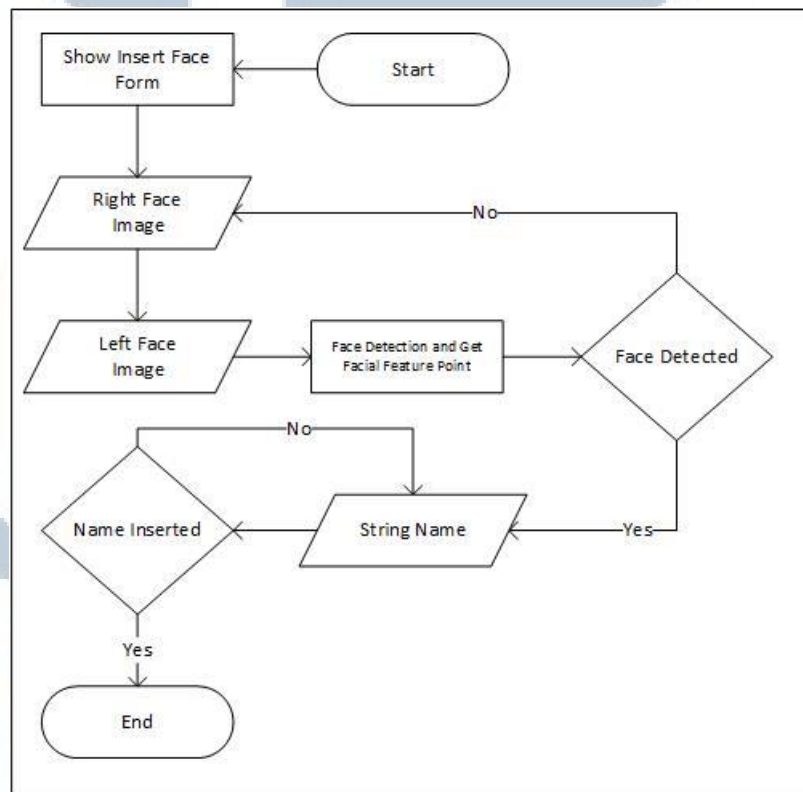
Gambar 3.2 Flowchart Camera Calibration

Langkah awal dari sistem yaitu menerima input berupa citra papan catur dengan ukuran yang telah ditentukan (7x6 adalah ukuran yang digunakan dalam penelitian ini). Gambar tersebut akan diproses oleh sistem untuk mencari titik-titik sudut pada papan catur tersebut. Jika sistem tidak menemukan data berupa citra papan catur berukuran 7x6, maka sebuah modal pemberitahuan akan keluar dan memberitahukan user bahwa tidak terdapat gambar papan catur berukuran 7x6 pada

citra yang dimasukkan dan meminta untuk memasukan kembali citra papan catur dengan ukuran 7x6. Jika ukuran titik-titik sudut pada citra papan catur tersebut terdapat ukuran 7x6, maka sistem akan melakukan proses perhitungan parameter-parameter pada kamera yang akan digunakan sebagai masukan dari perhitungan algoritma *Direct Linear Transformation (DLT)*. Parameter-parameter yang terdapat pada kamera adalah parameter intrinsik dan parameter ekstrinsik. Parameter ekstrinsik memiliki matriks rotasi dan matriks translasi yang akan digunakan untuk melakukan perhitungan matriks kamera kalibrasi.

C. Flowchart Insert Face

Gambar berikut merupakan *flowchart insert face* yang menjelaskan oalur algoritma dari proses pemasukan wajah ke dalam sistem.



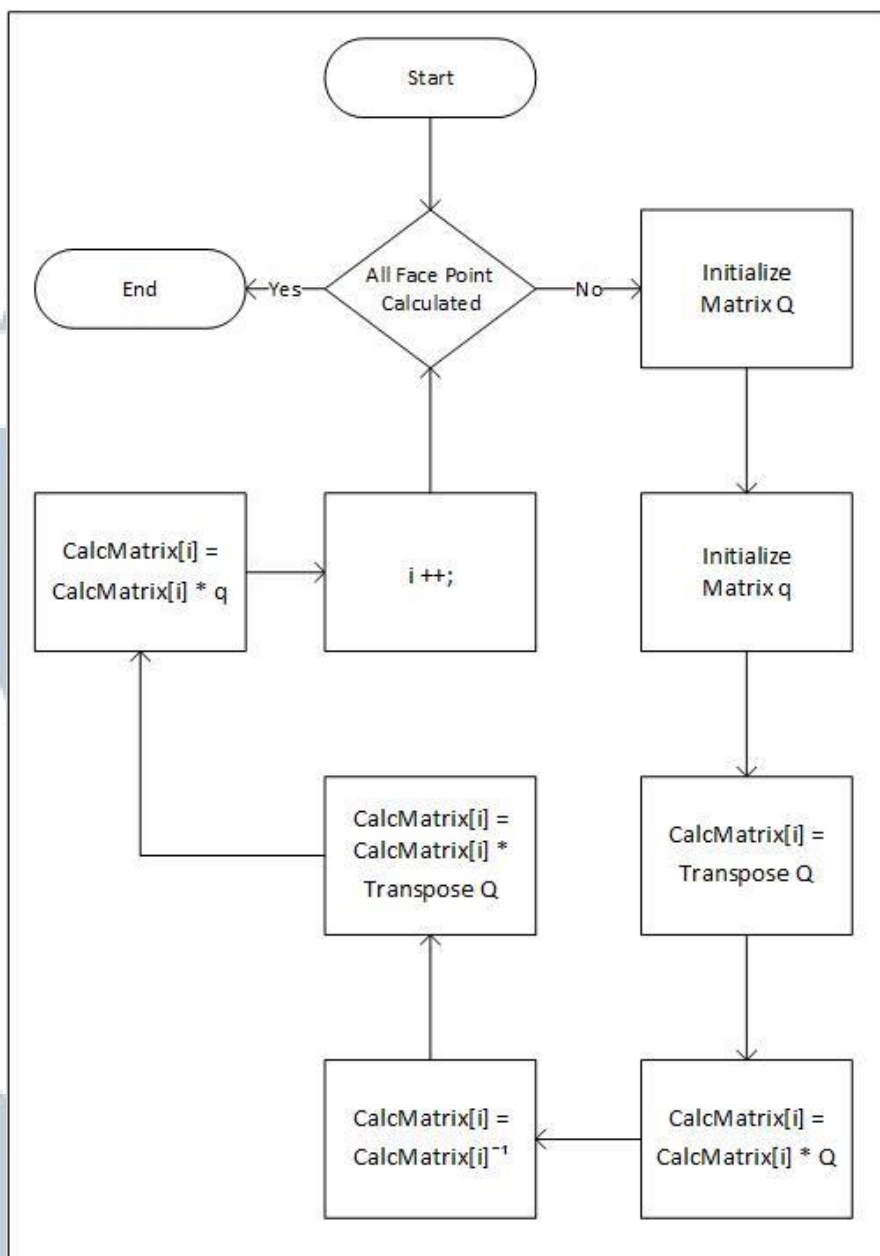
Gambar 3.3 Flowchart Insert Face

Data masukan berupa citra wajah yang diambil menggunakan dua buah kamera yang telah di kalibrasi sebelumnya dibutuhkan oleh sistem untuk dapat melakukan perhitungan titik-titik tiga dimensi dengan algoritma *Direct Linear Transformation*. Tepat setelah sistem menerima data berupa citra wajah sistem akan melakukan pendeteksian wajah terhadap citra tersebut, jika terdapat wajah yang terdeteksi oleh sistem pada citra tersebut maka sistem akan melakukan pendeteksian titik-titik wajah dengan menggunakan *library FaceLuxand SDK* yang menghasilkan titik-titik dua dimensi sebanyak 70(tujuh puluh) buah titik. Langkah selanjutnya adalah memasukan data berupa nama dari citra wajah yang dimasukkan.

D. Flowchart Direct Linear Transformation

Flowchart Direct Linear Transformation menjelaskan alur dari algoritma *Direct Linear Transformation* (DLT), yang memanfaatkan input berupa 2(dua) buah titik 2(dua) dimensi dan mengubah kedua titik dua dimensi tersebut menjadi sebuah titik tiga dimensi.

Pada Gambar 3.4 menjelaskan alur sistem pada perhitungan algoritma *Direct Linear Transformation*. Sistem akan melakukan perhitungan terhadap semua titik yang telah terdeteksi sebelumnya oleh sistem dengan menggunakan *library FaceLuxand SDK*. Data-data berupa matriks kalibrasi dari kamera kiri dan matriks kalibrasi dari kamera kanan serta titik-titik dua dimensi dari wajah pada citra masukan dibutuhkan oleh sistem untuk dapat melakukan perhitungan algoritma *Direct Linear Transformation*.



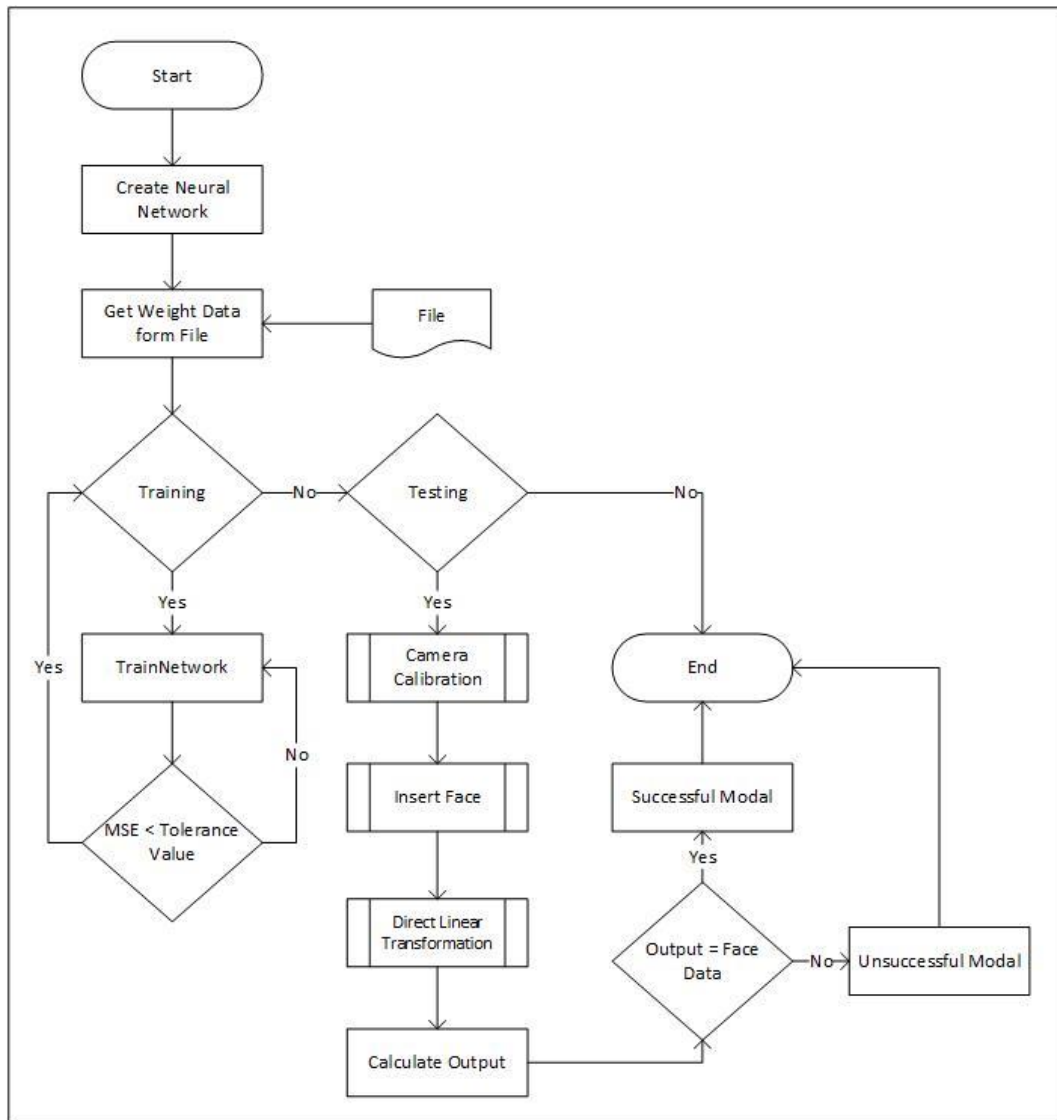
Gambar 3.4 Flowchart Direct Linear Transformation

Data-data tersebut digunakan oleh sistem untuk dapat membuat matriks Q beserta matriks q yang digunakan pada proses perhitungan. Hasil perhitungan pada Gambar 3.4. di atas merupakan titik tiga dimensi pada wajah dan akan disimpan ke dalam *file* s dan akan digunakan pada proses pengenalan wajah.

E. Flowchart Face Recognition

Flowchart face recognition menjelaskan alur pada sistem dalam mengenali wajah masukan. Pada proses pengenalan wajah menggunakan *library neural network* memiliki dua menu yaitu menu *training* atau pelatihan dan menu *testing* atau pengujian. Menu pelatihan dijalankan oleh sistem yang tujuan membuat proses pengujian lebih akurat, proses pelatihan akan terus dijalankan hingga nilai MSE (*Mean Squared Error*) lebih kecil dari nilai toleransi yang telah ditetapkan. Menu pengujian bertujuan untuk melakukan pengujian terhadap wajah pada citra masukan dari kedua kamera yaitu kamera kiri dan kamera kanan. Proses pengujian memiliki proses yang sama dengan proses pemasukan wajah dengan terlebih dahulu melakukan proses kalibrasi kamera pada kamera kanan dan kamera kiri sehingga mendapatkan matriks kamera kalibrasi pada kedua kamera tersebut. Setelah matriks didapatkan maka langkah selanjutnya merupakan pemasukan citra wajah yang ditangkap menggunakan kamera yang telah dikalibrasi sebelumnya. Sistem akan mendeteksi wajah pada citra masukan tersebut dan akan melakukan pendeteksian titik-titik wajah jika sistem mendeteksi wajah pada citra masukan tersebut.





Gambar 3.5 Flowchart Face Recognition

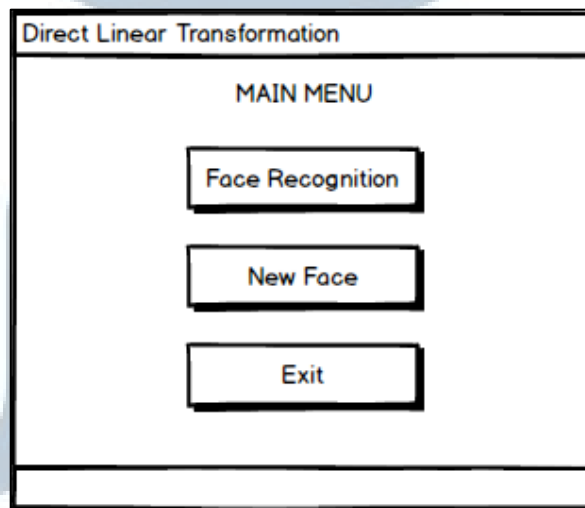
Data-data berupa matriks kalibrasi kamera dan titik-titik wajah tersebut akan digunakan sebagai parameter perhitungan algoritma *Direct Linear Transformation* dan akan menghasilkan data berupa titik-titik tiga dimensi. Hasil perhitungan tersebut akan dimasukkan ke dalam *neural network* untuk dilakukan perhitungan mendapatkan nilai *output* yang digunakan sebagai pembandingan pada nilai *output* yang berada pada database.

3.3.2 Desain Antarmuka

Berikut ini merupakan rancangan antarmuka yang digunakan pada sistem. Sistem memiliki beberapa desain antarmuka yang terdiri dari desain antarmuka *form* menu utama, desain antarmuka *form* pengenalan wajah, desain antarmuka *form* kalibrasi kamera, desain antarmuka *form* penambahan wajah, desain antarmuka *form* pemilihan titik, desain antarmuka *modal* keluar program, desain antarmuka *login success modal*, desain antarmuka *login failed modal*, desain antarmuka *insert chess board modal*, desain antarmuka *insert face modal*, dan desain antarmuka *undetected face modal*.

A. Form Menu Utama

Gambar berikut merupakan gambar rancangan desain antar muka pada *form* menu utama pada sistem.



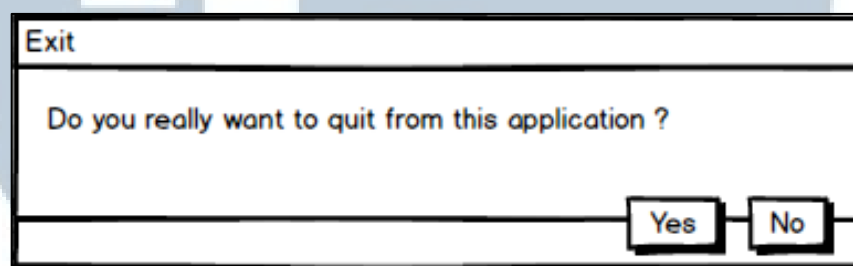
Gambar 3.6 Desain Antarmuka Form Menu Utama

Form menu utama merupakan *form* pertama yang dijalankan oleh sistem, *form* ini memiliki 3(tiga) tombol utama yaitu tombol *face recognition* yang digunakan untuk membuka *form* pengenalan wajah ketika tombol tersebut ditekan

oleh *user*, tombol *new face* digunakan untuk membuka *form* penambahan wajah ketika tombol tersebut ditekan, dan tombol *exit* digunakan untuk keluar dari aplikasi. Tombol *exit* tidak menutup sistem secara langsung melainkan menampilkan *modal* keluar program untuk memberikan pilihan terakhir kepada *user*.

B. Modal Keluar Program

Gambar berikut merupakan rancangan desain antar muka pada *modal* keluar program.

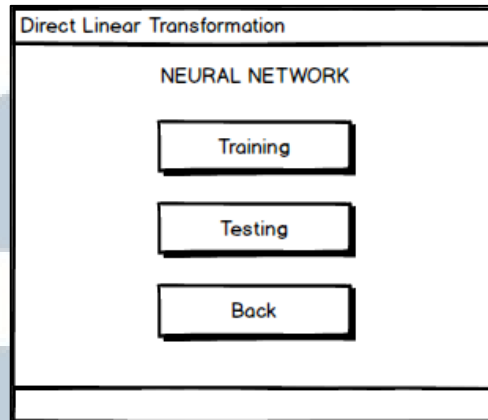


Gambar 3.7 Desain Antarmuka Modal Keluar Program

Modal tersebut dijalankan ketika *user* menekan tombol *exit* yang berada pada *form* menu utama. Pada *modal* ini *user* diberikan pilihan untuk dapat keluar dari program atau tidak keluar dari program, dengan melakukan pemilihan pada tombol *Yes* atau tombol *No*.

C. Form Pengenalan Wajah

Gambar 3.8 merupakan rancangan desain antar muka pada *form* pengenalan wajah. Pada *form* pengenalan wajah memiliki 3(tiga) tombol, yaitu tombol *Training* yang digunakan untuk menampilkan *form* Pembelajaran.

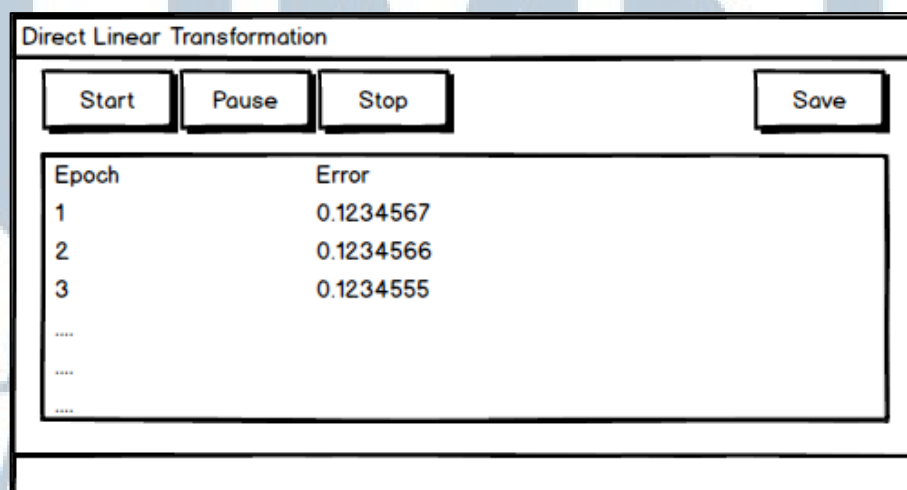


Gambar 3.8 Desain Antarmuka Form Pengenalan Wajah

Tombol *Testing* yang digunakan untuk melakukan pengujian data wajah. Pengujian dilakukan dengan mengkalibrasi kamera terlebih dahulu dan selanjutnya memasukan data wajah yang diambil dari 2(dua) buah kamera, sehingga *form* kalibrasi kamera akan ditampilkan ketika tombol tersebut ditekan oleh *user*. Tombol *back* berfungsi untuk kembali ke halaman utama yaitu *form* menu utama.

D. Form Detail Pelatihan

Gambar 3.9 merupakan gambar rancangan desain antarmuka pada *form* Detail Pelatihan.

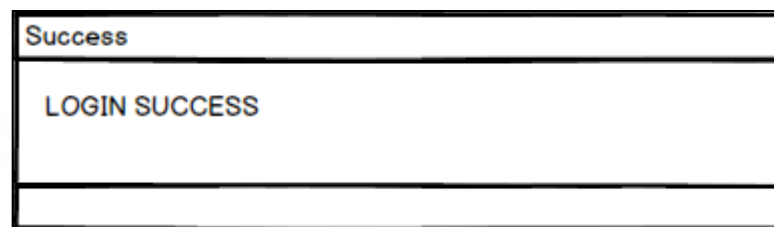


Gambar 3.9 Desain Antarmuka Form Detail Pelatihan

Pada *form* tersebut sistem akan menjalankan pelatihan pada *neural network* dengan turut menampilkan data error pada *ListView* yang tersedia. *ListView* tersebut menampilkan data berupa *epoch* dan *error* pada pelatihan *neural network*. Terdapat 4(empat) tombol pada *form* tersebut. Tombol *start* berfungsi untuk memulai perhitungan dari pelatihan *neural network* serta menampilkan data perhitungan di *ListView*, tombol *pause* berfungsi untuk memberikan sementara perhitungan pelatihan *neural network* dan dapat dilanjutkan kembali dengan menekan tombol *start*. Tombol *stop* berfungsi untuk memberhentikan proses perhitungan pelatihan *neural network* dan tombol *save* berfungsi untuk menyimpan data hasil perhitungan *neural network* tersebut.

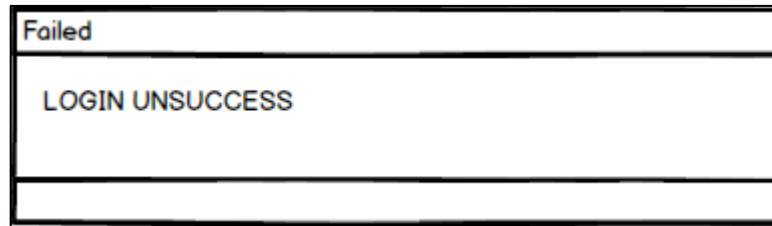
E. Modal Success dan Failed

Gambar di atas merupakan rancangan desain antarmuka pada *modal success*. *Modal*



Gambar 3.10 Desain Antarmuka Login Success Modal

Modal tersebut bertujuan untuk memberikan informasi bahwa data citra 2(dua) dimensi memiliki kesamaan dengan data yang berada pada database kepada *user*. Data tersebut akan ditampilkan pada *modal success* berupa data nama dari *user*.

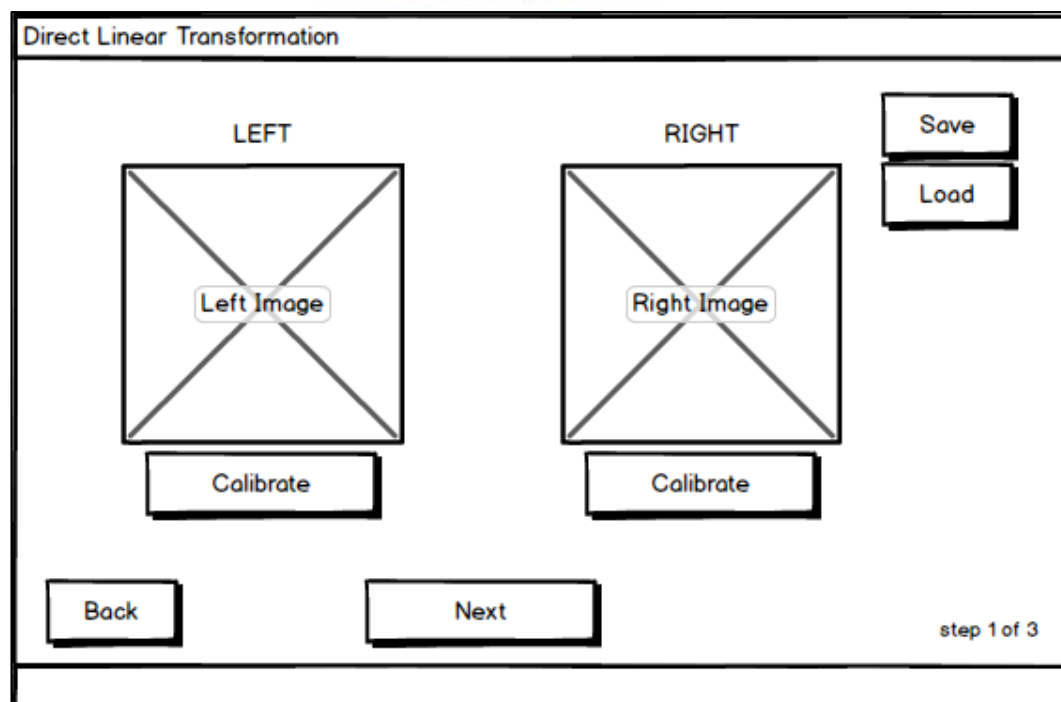


Gambar 3.11 Desain Antarmuka Login Failed Modal

Gambar di atas merupakan rancangan desain antar muka pada *modal success*. *Modal* tersebut bertujuan untuk memberikan informasi bahwa data citra 2(dua) dimensi tidak memiliki kesamaan dengan semua data yang berada pada database kepada *user*. Kedua *modal* tersebut yaitu *modal success* dan *modal unsuccess* dijalankan dalam *form* pengenalan wajah.

F. Form Kalibrasi Kamera

Gambar 3.12 merupakan gambar desain antar muka dari *form* kalibrasi kamera.

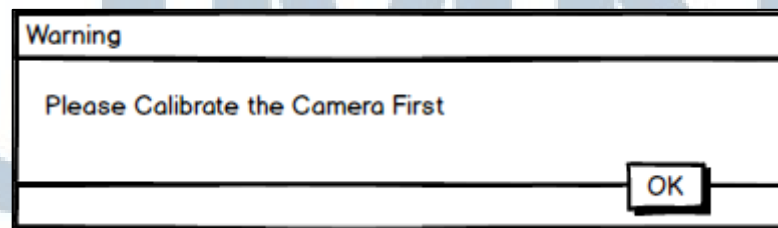


Gambar 3.12 Desain Antarmuka Form Kalibrasi Kamera

Terdapat beberapa tahap sebelum algoritma *Direct Linear Transformation* dijalankan antara lain adalah tahap perhitungan matriks kamera kalibrasi dan tahap memasukan citra wajah. Pada tahap perhitungan matriks kamera kalibrasi terdiri dari perhitungan matriks kalibrasi kamera kiri dan kamera kanan. Pada form ini terdapat beberapa tombol antara lain adalah tombol *calibrate* yang digunakan untuk menampilkan *form* detail kalibrasi kamera. Tombol *save* digunakan untuk melakukan penyimpanan data matriks kalibrasi kamera ke dalam file .txt dengan memanfaatkan fitur *openFileDialog* dan tombol *load* digunakan untuk melakukan pemuatan data matriks kalibrasi kamera ke dalam sistem dengan memanfaatkan fitur *openFileDialog*. Tombol *back* berfungsi untuk kembali ke *form* sebelumnya yaitu *form* Menu Utama.

G. Modal Kalibrasi Kamera

Gambar 3.13 merupakan desain antarmuka dari *modal* kalibrasi kamera, yang akan dijalankan ketika tombol *next* pada *form* kalibrasi kamera ditekan dan sistem tidak memiliki matriks kalibrasi kamera pada kedua kamera maupun salah satu kamera.

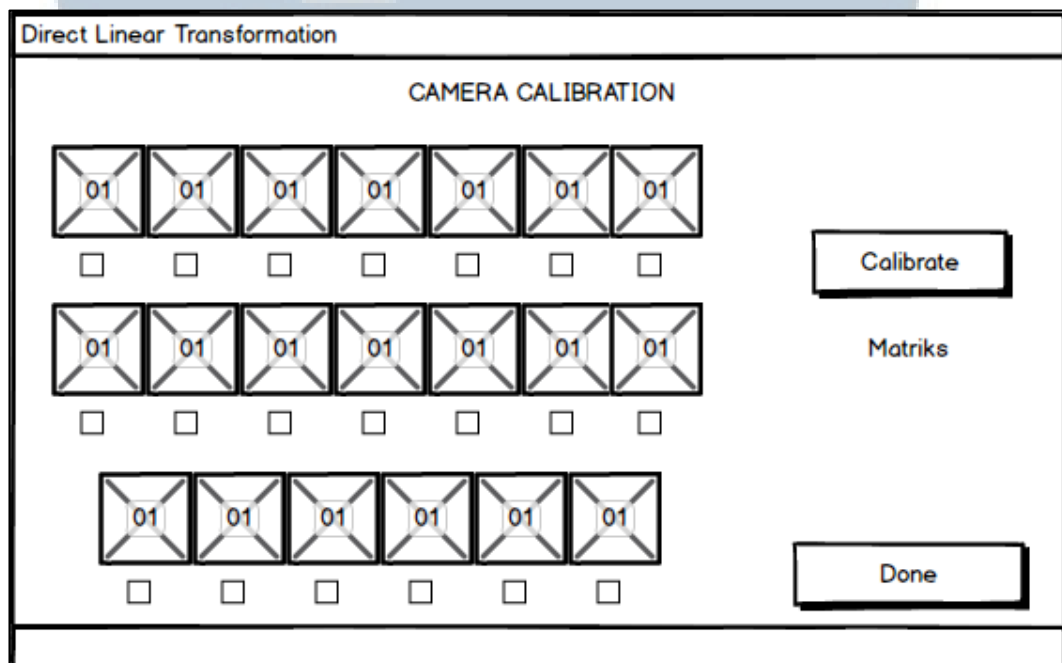


Gambar 3.13 Desain Antarmuka Modal Kalibrasi Kamera

Modal tersebut bertujuan untuk memberi peringatan terhadap *user* untuk terlebih dahulu melakukan kalibrasi kamera sebelum melanjutkan ke tahap selanjutnya yaitu memasukan citra wajah dua dimensi.

H. Form Detail Kalibrasi Kamera

Gambar 3.14 merupakan gambar dari rancangan desain antarmuka *form* detail kalibrasi kamera. Pada *form* tersebut terdapat 20(dua puluh) *ImageBox* yang digunakan untuk memasukan citra dengan menggunakan fitur *OpenFileDialog* dan selanjutnya sistem akan mendeteksi citra papan catur dengan ukuran yang telah ditentukan (7x6 adalah ukuran yang digunakan pada penelitian ini), serta 20(dua puluh) *CheckBox* yang digunakan untuk memilih citra yang akan diproses untuk mendapatkan matriks kamera kalibrasi



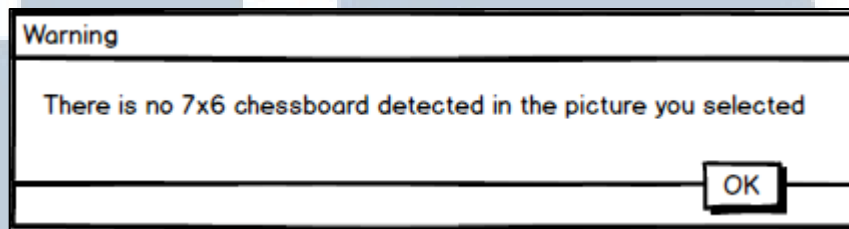
Gambar 3.14 Desain Antarmuka Form Detail Kalibrasi Kamera

Tombol *Calibrate* digunakan untuk melakukan perhitungan matriks kamera kalibrasi yang bertujuan untuk mendapatkan parameter-parameter dari kamera yang digunakan untuk menangkap citra papan catur tersebut. Parameter tersebut berupa parameter *intrinsic* dan *extrinsic*, yang akan digunakan dalam perhitungan mendapatkan matriks kalibrasi kamera. Matriks kalibrasi kamera tersebut akan

digunakan sebagai masukan dari algoritma *Direct Linear Transformation*. Tombol *Done* digunakan untuk kembali ke *form* sebelumnya yaitu *form* Kalibrasi Kamera.

I. Insert Chessboard Modal

Gambar 3.15 merupakan desain antarmuka dari *insert chessboard modal*, yang akan dijalankan ketika sistem mendapatkan 2(dua) kejadian.

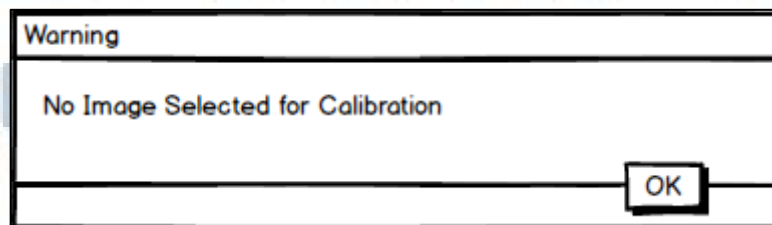


Gambar 3.15 Desain Antarmuka Insert Chessboard Modal

Kejadian pertama adalah kejadian dimana sistem tidak menemukan papan catur dengan ukuran yang telah ditentukan (7x6 adalah ukuran yang digunakan pada penelitian ini). Kejadian kedua adalah kejadian ketika tidak terdapat citra papan catur pada *ImageBox*.

J. Calibration Image Modal

Gambar 3.16 merupakan desain antarmuka *calibration image modal*. Sistem akan menjalankan *modal* tersebut ketika *user* menekan tombol *Calibrate* pada *form* detail kalibrasi kamera dan tidak terdapat citra papan catur yang terdeteksi pada *form* tersebut.

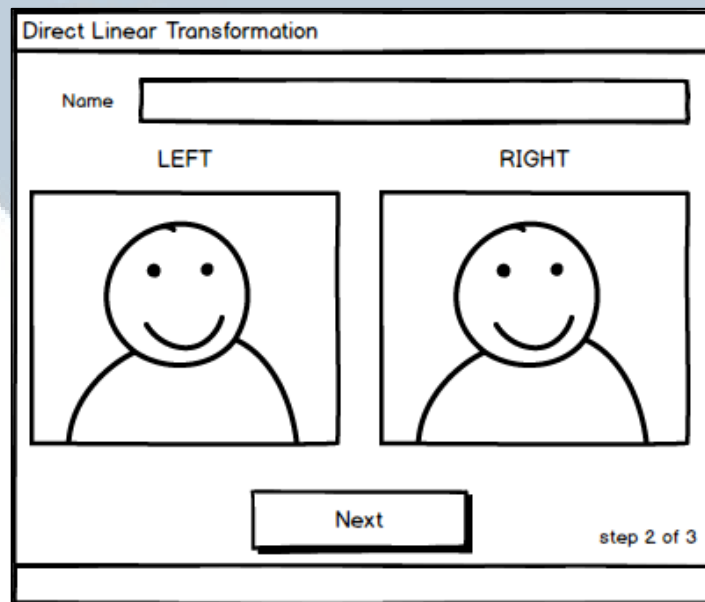


Gambar 3.16 Desain Antarmuka Calibration Image Modal

Modal tersebut bertujuan untuk memberi peringatan kepada *user* untuk terlebih dahulu memasukan citra papan catur sebelum melakukan perhitungan kalibrasi kamera.

K. Form Penambahan Wajah

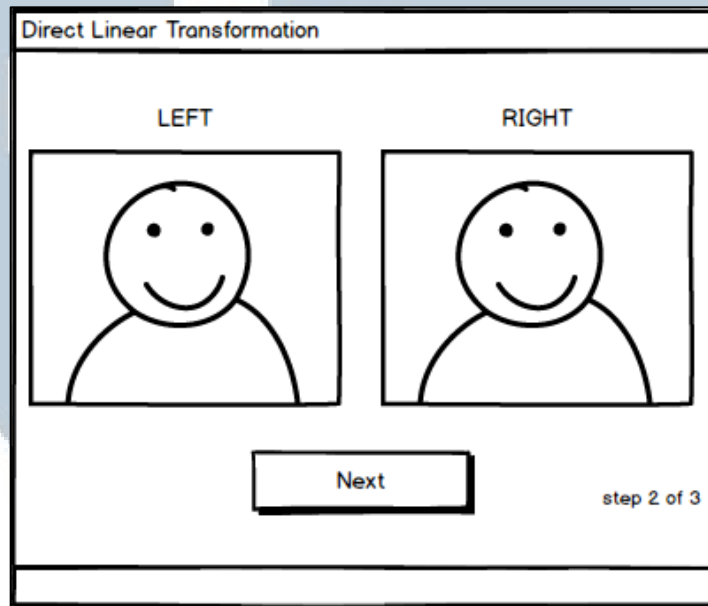
Gambar 3.17 merupakan rancangan desain antarmuka pada *form* penambahan wajah. *Form* tersebut bertujuan untuk memasukan 2(dua) citra 2(dua) dimensi yang digunakan sebagai data masukan pada proses *Direct Linear Transformation* untuk mendapatkan koordinat-koordinat 3(tiga) dimensi.



Gambar 3.17 Desain Antarmuka Form Penambahan Wajah

Pada *form* ini terdapat 2(dua) *ImageBox* yang dapat ditekan dan akan menampilkan *openFileDialog* yang bertujuan untuk mengambil citra 2(dua) dimensi dan menampilkan citra tersebut pada *ImageBox*. Citra wajah masukan akan di-*check* dengan memanfaatkan *library Luxand FaceSDK* untuk mendeteksi wajah serta mendapatkan titik-titik wajah dari citra tersebut. Tombol *Next* berperan sebagai tombol konfirmasi bahwa citra yang terdapat pada *ImageBox* merupakan

citra yang tepat. Setelah citra merupakan citra yang tepat langkah selanjutnya sistem akan melakukan perhitungan titik-titik tiga dimensi dengan *Direct Linear Transformation*. *Form* tersebut juga memiliki *textbox* yang berfungsi untuk memasukan data berupa nama dari citra wajah yang dimasukkan ke dalam kedua *PictureBox* yang telah tersedia sebelumnya.

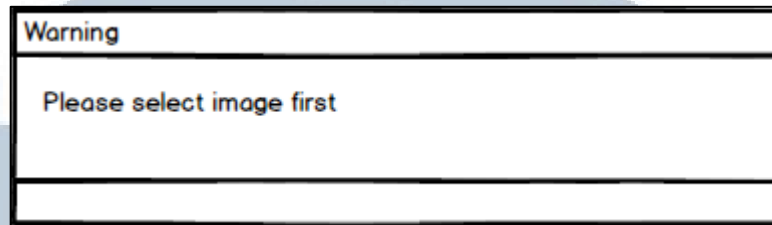
The image shows a software interface window titled "Direct Linear Transformation". Inside the window, there are two rectangular boxes labeled "LEFT" and "RIGHT". Each box contains a simple line drawing of a person's face with a smile. Below these two boxes is a rectangular button labeled "Next". In the bottom right corner of the window, it says "step 2 of 3".

Gambar 3.18 Desain Antarmuka Form Pemasukan Wajah Pengenalan

Form Penambahan Wajah memiliki tampilan yang sama dengan *Form* Pemasukan Wajah Pengenalan, tetapi dengan fungsi yang berbeda. *Form* Penambahan Wajah berfungsi untuk memasukan data wajah baru kedalam sistem dan disimpan di *database*, sedangkan pada *Form* Pemasukan Wajah Pengenalan berfungsi untuk melakukan pengujian wajah masukan dengan membandingkan data wajah yang terdapat pada *database*. Perbedaan diantara kedua *form* terlihat jelas dengan hilangnya *textbox* pemasukan nama wajah pada sisi atas dari *form*.

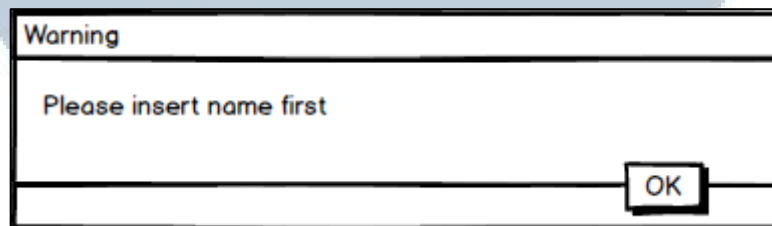
L. Insert Face Modal dan Insert Name Modal

Gambar 3.19 merupakan gambar desain antarmuka pada *insert face modal*.



Gambar 3.19 Desain Antarmuka Insert Face Modal

Modal tersebut merupakan *modal* pemberitahuan kepada *user* untuk memasukan data berupa citra wajah terlebih dahulu sebelum melanjutkan ke tahap perhitungan titik-titik tiga dimensi dengan algoritma *Direct Linear Transformation*.

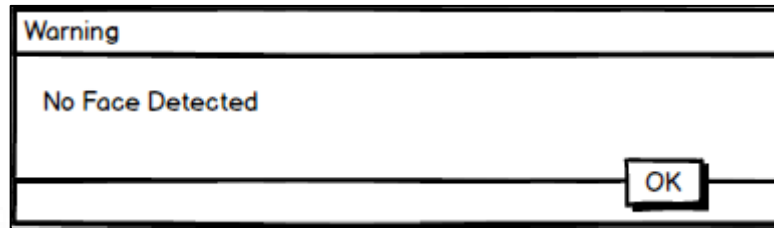


Gambar 3.20 Desain Antarmuka Insert Name Modal

Gambar di atas merupakan desain antarmuka pada *insert name modal*. *Modal* tersebut merupakan *modal* pemberitahuan kepada *user* untuk memasukan data berupa nama dari citra wajah terlebih dahulu sebelum melanjutkan ke tahap perhitungan titik-titik tiga dimensi dengan algoritma *Direct Linear Transformation*.

M. Undetected Face Modal

Desain antarmuka di atas merupakan desain antarmuka *undetected face modal*. *Modal* tersebut akan dijalankan ketika tidak terdeteksi data wajah dari citra wajah masukan.



Gambar 3.21 Desain Antarmuka Undetected Face Modal

Proses deteksi wajah serta mendapatkan titik-titik dari wajah menggunakan *library Luxand FaceSDK*. Modal tersebut bertujuan memberikan peringatan kepada *user* untuk dapat memasukan citra wajah masukkkan yang tepat.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA