



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

METODE DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Penerapan aplikasi Face Tracking dan Filter berdasarkan raut wajah menggunakan algoritma Fisher-Yates, dijelaskan pada tahapan-tahapan sebagai berikut:

a. Studi Pustaka

Penelitian dimulai dengan mencari berbagai sumber yang relevan serta konkret dengan penelitian ini. Sumber yang dicari dapat berupa *e-book*, *e-journal*, website dan lain sebagainya. Pada tahapan ini, konsep-konsep yang dibutuhkan dalam penelitian seperti algoritma Fisher-Yates, Computer Vision, Face Tracking, dan Classifier akan dimatangkan.

b. Pengumpulan Data

Data penelitian yang digunakan adalah foto-foto raut wajah dengan berbagai ekspresi, yaitu marah, senang, sedih, dan terkejut. Data yang digunakan adalah *dataset* dari Cohn-Kanade AU-Coded Facial Expression Database (CK) dan Karolinska Directed Emotional Faces (KDEF).

c. Proses Pembersihan Data

Proses pembersihan data merupakan proses pembersihan data seperti memeriksa kondisi data, menghilangkan redundansi data, memperbaiki kesalahan pada data, dan menghilangkan beberapa data yang tidak diperlukan dan tidak layak uji. Dalam penelitian, data yang telah

dilakukan proses pembersihan akan menghasilkan foto-foto raut wajah dengan cahaya yang cukup, tidak *blur*, dan tidak *noisy*.

d. Proses Transformasi Data

Proses transformasi data merupakan pengelompokan atau penyederhanaan data. Data berupa foto-foto raut wajah akan di unggah ke <https://customvision.ai> untuk melatih *classifier* (*Training the Classifier*). Setelah melakukan *Training Classifier*, data dapat dilakukan *evaluation* menggunakan proses *k-fold cross validation* yang akan menampilkan *Performance Per Tag* berupa *precision* dan *recall*. *Classifier* ini dapat dibuat menjadi sebuah model dan di unduh sebagai file berekstensi *.mlmodel* untuk digunakan dalam penelitian ini. Menurut CustomVision (2018), 50 gambar per kelas sudah cukup untuk memulai prototipe.

e. Perancangan dan Pembangunan Aplikasi

Setelah data dikumpulkan, dibersihkan, dan ditransformasikan, selanjutnya dimulai tahap perancangan dan pembangunan aplikasi. Pada tahap ini aplikasi akan dirancang dan dibangun sesuai dengan Data Flow Diagram (DFD), Flowchart dan rancang antar muka sebagai landasan pemrograman aplikasi secara keseluruhan. Data Flow Diagram digunakan untuk menggambarkan arus data pada aplikasi sebagai pedoman untuk implementasi. Flowchart dapat membantu untuk mengembangkan pemahaman tentang bagaimana tiap proses saling berinteraksi. Rancang antar muka memainkan peran penting dalam sebuah *software*. Rancang antar muka berfokus untuk mengantisipasi apa yang mungkin perlu dilakukan oleh pengguna dan memastikan bahwa antar muka memiliki

elemen yang mudah diakses, dipahami, dan digunakan untuk memfasilitasi tindakan tersebut. Rancang antar muka dari segi visibilitas, baik desain dan ketepatannya memegang arti penting yang utama untuk menggambarkan jumlah yang tepat dari informasi untuk pengguna yang dituju (Guntupa, 2008).

f. Uji Coba Aplikasi

Uji coba aplikasi berfungsi untuk mendapatkan data faktual berupa nilai *confidence* dari raut wajah pengguna berdasarkan model yang telah di *training* oleh Custom Vision. Data *confidence* memberikan nilai akurasi observasi dari raut wajah pengguna terhadap model yang telah di *training*. Nilai *confidence* yang telah di normalkan adalah 0.0 hingga 1.0, dimana nilai 0.0 mengindikasikan tidak ada *confidence*, sedangkan nilai 1.0 mengindikasikan *confidence* tertinggi. Pada tahap pengujian, dilakukan pula pengukuran tingkat *rasa kesenangan (joy)* dengan menggunakan Hedonic Motivation System Adoption Model (HMSAM). Data yang diperlukan dalam pengujian tingkat *rasa kesenangan (joy)* didapat dari kuesioner yang diberikan kepada pengguna.

g. Evaluasi Hasil Aplikasi

Evaluasi bertujuan untuk menilai hasil dari data uji coba dengan hasil *training performance* pada Custom Vision dan menarik kesimpulan dari penelitian yang telah dibuat. Selain menilai hasil *training performance* pada Custom Vision, pada tahap evaluasi akan mengukur kegunaan dan tingkat kepuasan penggunaan untuk tingkat *rasa kesenangan* melalui penggunaan skala Likert lima tingkat.

h. Penulisan dan Konsultasi

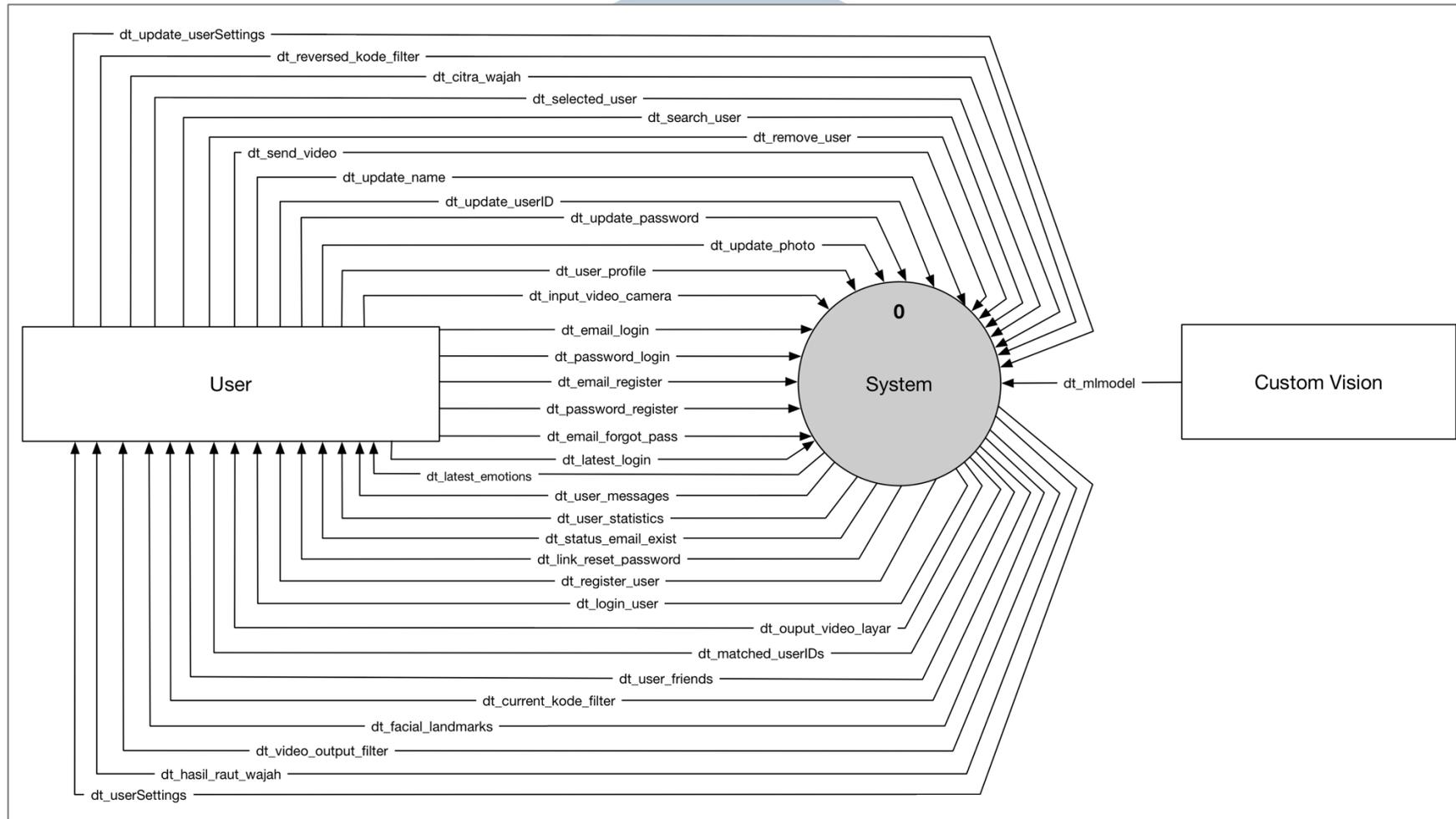
Penulisan merupakan tahapan penyusunan laporan sebagai dokumentasi disertai dengan konsultasi sebagai evaluasi dari penulisan laporan.

3.2 Perancangan Aplikasi

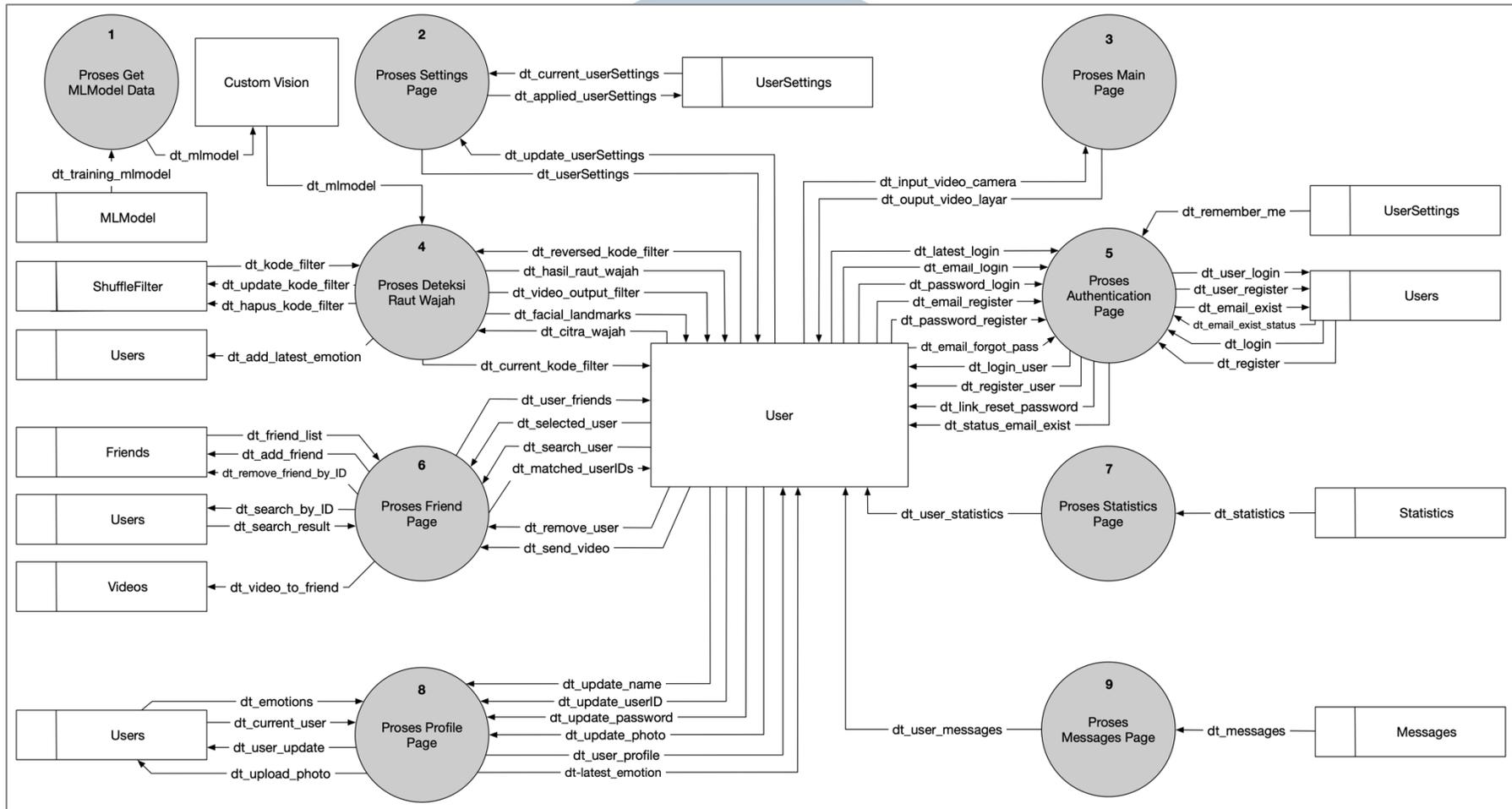
Dalam perancangan aplikasi pada penelitian ini, dibuatlah Data Flow Diagram, Flowchart, dan Rancangan Antarmuka.

3.2.1 Data Flow Diagram

Data Flow Diagram (DFD) dirancang untuk menggambarkan sistem sebagai suatu jaringan proses yang terhubung satu sama lain dengan alur data. Pada bagian *context diagram* seperti yang ditunjukkan pada Gambar 3.1, entitas User dapat memberikan data *update userSettings*, data *reversed* kode *filter*, data citra wajah, data *selected user*, data *search user*, data *remove user*, data *send video*, data *update name*, data *update userID*, data *update password*, data *update photo*, data *user profile*, data *input video camera*, data *email login*, data *password login*, data *email register*, data *password register*, data *email forgot pass* dan data *latest login*. Entitas User dapat menerima data *latest emotions*, data *user messages*, data *user statistics*, data *status email exist*, data *link reset password*, data *register user*, data *login user*, data *output video* layar, data *matched userID*, data *user friend*, data *current* kode *filter*, data *facial landmarks*, data *video output filter*, data hasil raut wajah dan data *userSettings*. Entitas Custom Vision memberikan data *output mlmodel* yang merupakan data hasil *training*.

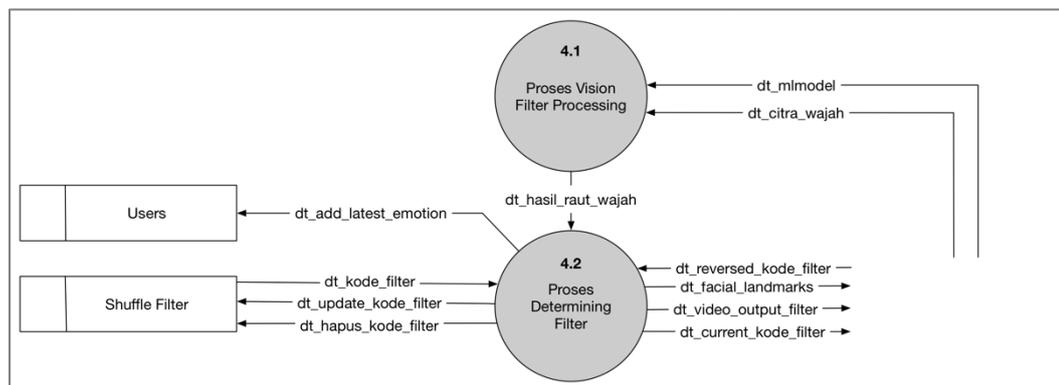


Gambar 3.1 Data Flow Diagram Context



Gambar 3.2 Data Flow Diagram Level 0

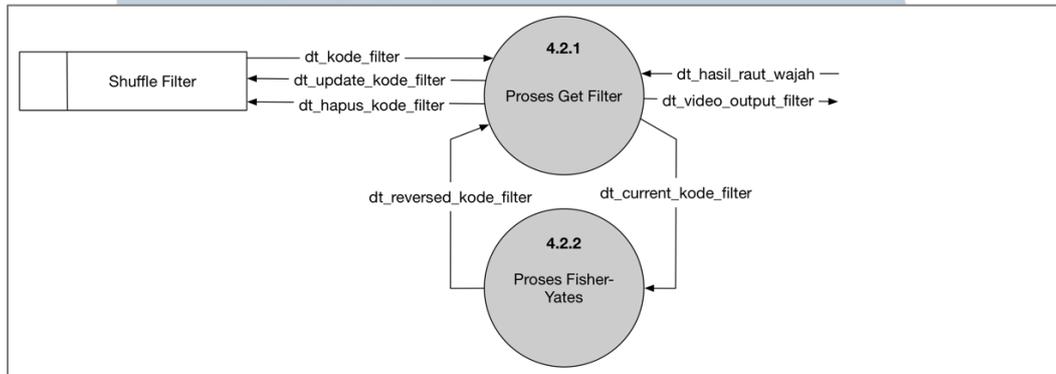
Gambar 3.2 merupakan gambar *data flow diagram level 0* yang merupakan perincian dari *diagram context* pada gambar 3.1. Pada *data flow diagram level 0*, semua proses pada sistem akan dijabarkan secara terperinci. Proses yang terdapat pada *data flow diagram level 0* diantaranya, Proses Get MLModel Data untuk mendapatkan data hasil *training* dari file *mlmodel*, Proses Settings Page untuk mendapatkan dan *update* data pada UserSettings, Proses Main Page untuk proses *input* dan *output* video, Proses Deteksi Raut Wajah untuk mendeteksi raut wajah dan menampilkan *filter* berdasarkan raut wajah menggunakan Fisher-Yates Algorithm, Proses Authentication Page untuk dapat melakukan *login*, *register*, dan *forgot password*, Proses Friend Page untuk melakukan manajemen *friend* seperti melihat, *remove* dan mengirimkan *video*, Proses Statistics Page untuk melihat *statistics* dari pengguna, Proses Profile Page untuk melakukan manajemen *profile* pengguna yaitu melihat dan *update* informasi pengguna seperti *email*, nama, *password* dan *profile picture*, dan Proses Messages Page untuk mendapatkan data *messages* yang dikirimkan oleh *friend* pengguna kepada pengguna.



Gambar 3.3 Data Flow Diagram Level 1 Proses Deteksi Raut Wajah

Gambar 3.3 merupakan gambar *data flow diagram level 1* Proses Deteksi Raut Wajah. Proses Deteksi Raut Wajah mempunyai dua subproses yaitu Proses Vision Filter Processing yang menerima input data *mlmodel* dan citra wajah.

Kemudian Proses Vision Filter Processing memberikan output data hasil raut wajah sebagai input untuk Proses Determining Filter. Proses Determining Filter berfungsi untuk menentukan filter yang digunakan berdasarkan input data hasil raut wajah pada Proses Vision Filter Processing. Pada *data flow diagram level 1* Proses Deteksi Raut Wajah terdapat 2 *datastore* yaitu, Users dan Shuffle Filter.

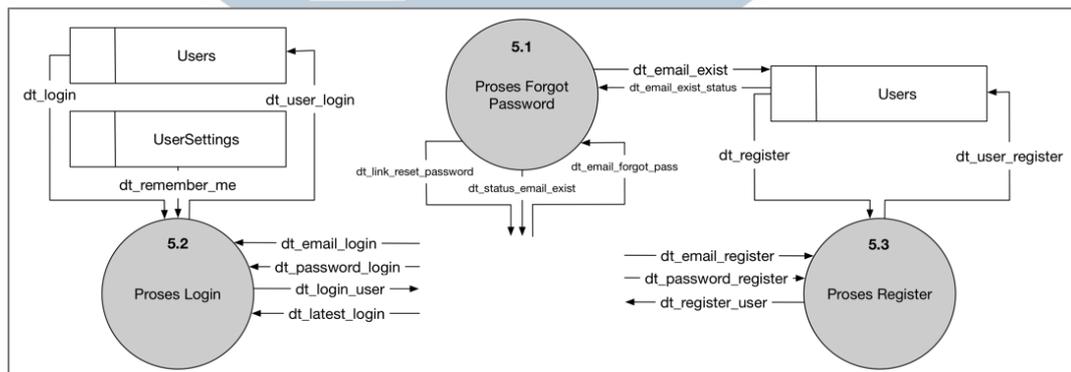


Gambar 3.4 Data Flow Diagram Level 1 Proses Determining Filter

Gambar 3.4 merupakan gambar *data flow diagram level 2* Proses Determining Filter. Proses Determining Filter mempunyai dua subproses yaitu, Proses Get Filter dan Proses Fisher-Yates. Proses Get Filter berfungsi untuk mendapatkan *filter* berdasarkan data hasil raut wajah. Proses Get Filter akan memberikan input data *current kode filter* ke Proses Fisher-Yates untuk *shuffle* data kode *filter* jika *flag isShuffled* pada data kode *filter* bernilai *false*. Pada Proses *data flow diagram level 2*, Proses Determining Filter terdapat satu *datastore* yaitu Shuffle Filter yang merupakan turunan dari proses sebelumnya.

Gambar 3.5 merupakan gambar *data flow diagram level 1* Proses Authentication Page yang merupakan perincian dari *diagram context*. Pada *data flow diagram level 1* Proses Authentication Page, terdapat subproses yaitu proses Forgot Password, Proses Login, dan Proses Register. Proses Forgot Password berfungsi untuk memproses *input* dan *output* data ketika pengguna lupa dari

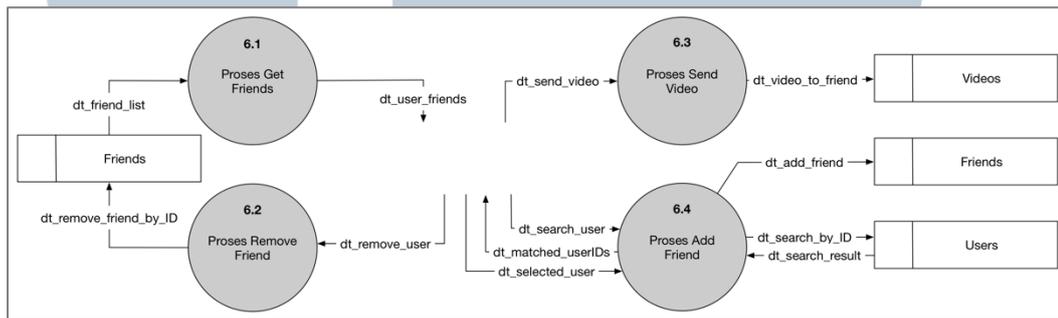
password accountnya. Proses Forgot Password menerima input data *email forgot pass* dan mengembalikan output data *status email exist* dan data *link reset password*. Proses Forgot Password menggunakan *datastore* Users untuk mengecek apakah terdapat *user* dengan *email* yang telah diberikan. Proses Login menerima input data *email login*, data *password login* dan data *latest login*. Pada Proses Login, *datastore* yang digunakan adalah *datastore* Users dan *datastore* UserSettings. *Datastore* Users pada Proses Login digunakan untuk mengecek data yang diberikan oleh pengguna. Sedangkan, *datastore* UserSettings pada Proses Login digunakan untuk mendapatkan data *remember me* jika pengguna telah mengatur sebelumnya. Proses Register menerima input data *email register* dan data *password register*. Proses Register menggunakan *datastore* Users untuk mengupdate *database* saat pengguna berhasil melakukan *register* pada aplikasi.



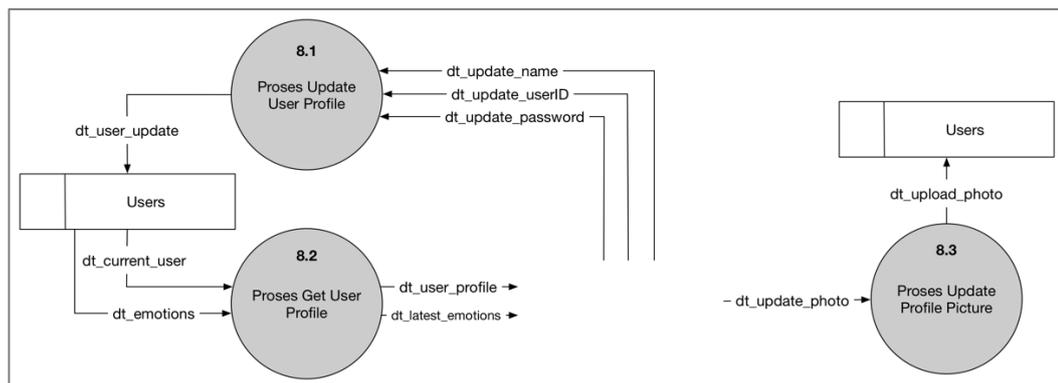
Gambar 3.5 Data Flow Diagram Level 1 Proses Authentication Page

Gambar 3.6 merupakan gambar *data flow diagram level 1* Proses Friend Page yang merupakan perincian dari *diagram context*. Pada *data flow diagram level 1* Proses Friend Page, terdapat empat subproses yaitu proses Get Friends, Proses Remove Friend, Proses Send Video dan Proses Add Friend. Proses Get Friends berfungsi untuk mendapatkan *friend list* dari pengguna dengan data *user friends* sebagai output dari *datastore* Friends. Proses Remove Friend berfungsi untuk

menghapus *friend* yang telah dipilih oleh pengguna dan mengupdate *database* melalui *datastore* Friends. Proses Send Video berfungsi untuk mengirim video kepada *friend* yang telah dipilih dengan input data *send video* dan mengupdate *database* dengan *datastore* Videos. Proses Add Friend berfungsi untuk menambahkan *friend* pada pengguna. Proses Add Friend mendapat input berupa data *search user*, dan data *selected user*. Proses Add Friend mengembalikan output kepada pengguna berupa data *matched* userIDs. Pada Proses Add Friend terdapat dua *datastore* yang digunakan, yaitu *datastore* Friends dan Users.



Gambar 3.6 Data Flow Diagram Level 1 Proses Friend Page



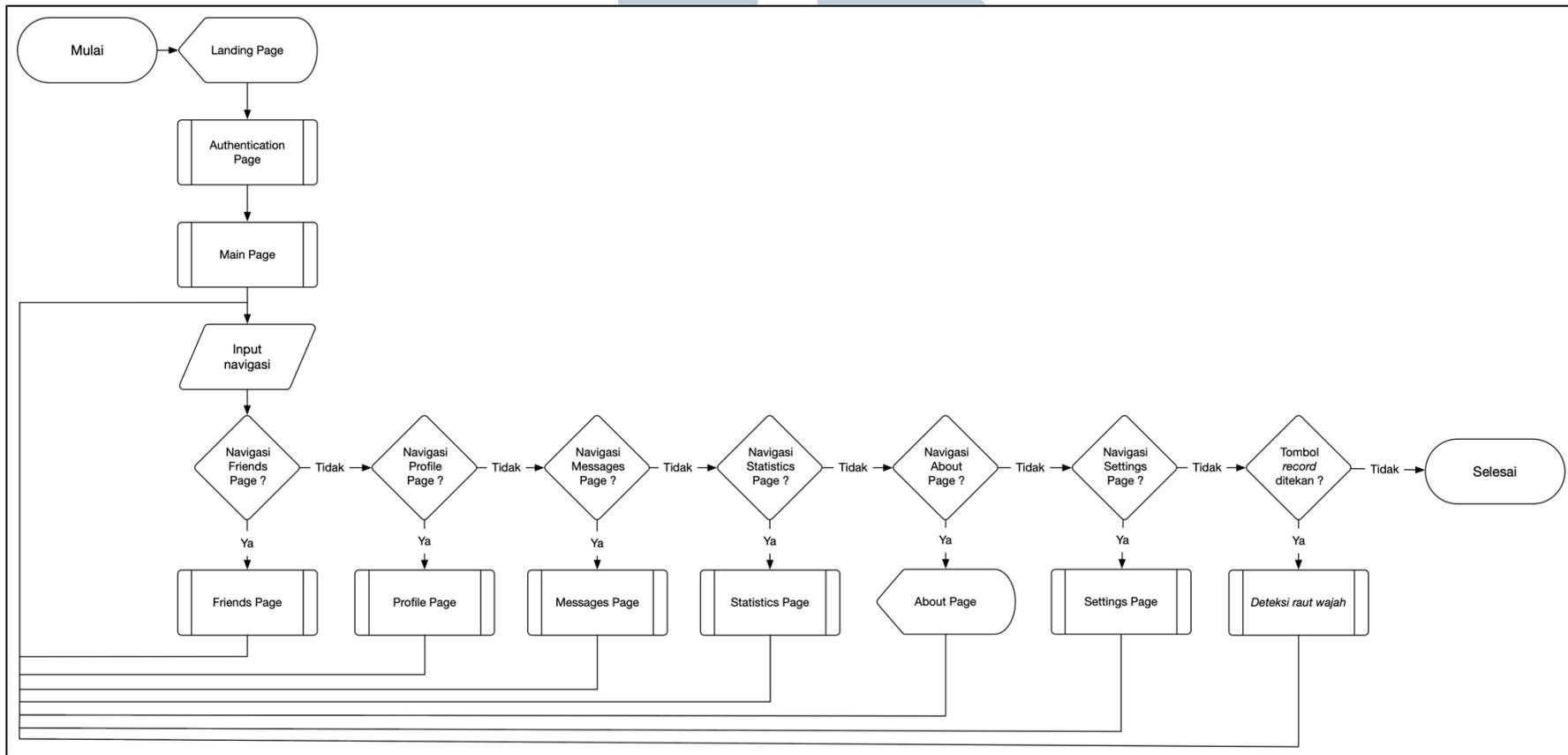
Gambar 3.7 Data Flow Diagram Level 1 Proses Profile Page

Gambar 3.7 merupakan gambar *data flow diagram level 1* Proses Profile Page yang merupakan perincian dari diagram context. Pada Proses Profile Page, terdapat tiga subproses yaitu Proses Update User Profile, Proses Get User Profile dan Proses Update Profile Picture. Proses Update User Profile berfungsi untuk

mengupdate informasi pengguna berupa nama, userID dan *password*. Proses Get User berfungsi untuk mendapatkan data *current user* dan data *emotions* sebagai informasi pengguna dan *emotions* yang terakhir digunakan. Proses Update Profile Picture berfungsi untuk mengganti foto dari pengguna.

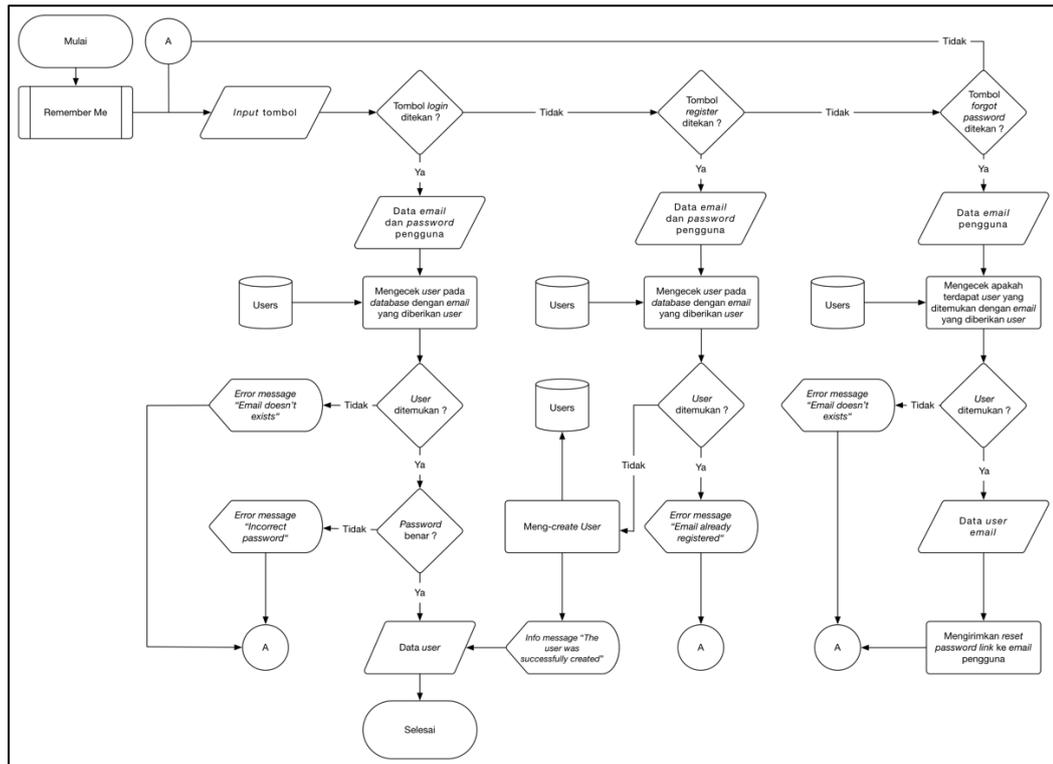
3.2.2 Flowchart

Penggambaran *flowchart* dapat membantu untuk mengembangkan pemahaman tentang bagaimana pada tiap proses saling berinteraksi. Dalam penggunaannya, aplikasi pada pertama kali akan menampilkan Landing Page kemudian melakukan subproses Authentication Page. Kemudian, aplikasi akan melakukan subproses Main Page dan aplikasi akan mendapatkan input navigasi dari pengguna, jika pengguna melakukan navigasi ke Friends Page, maka aplikasi akan melakukan subproses Friends Page. Jika pengguna melakukan navigasi ke Profile Page, maka aplikasi akan melakukan subproses Profile Page. Jika pengguna melakukan navigasi ke Messages Page, maka aplikasi akan melakukan subproses Messages Page. Jika pengguna melakukan navigasi ke Statistics Page, maka aplikasi akan melakukan subproses Statistics Page. Jika pengguna melakukan navigasi ke About, maka aplikasi akan menampilkan About Page. Jika pengguna melakukan navigasi ke Settings Page, maka aplikasi akan melakukan subproses Settings Page. Jika pengguna menekan tombol *record* pada Main Page, maka aplikasi akan melakukan subproses Deteksi Raut Wajah. Flow aplikasi akan berakhir saat pengguna tidak melakukan navigasi ataupun pengguna tidak menekan tombol *record* pada Main Page. Flowchart aplikasi secara umum seperti tertera pada gambar 3.8.



Gambar 3.8 Flowchart Aplikasi Secara Umum

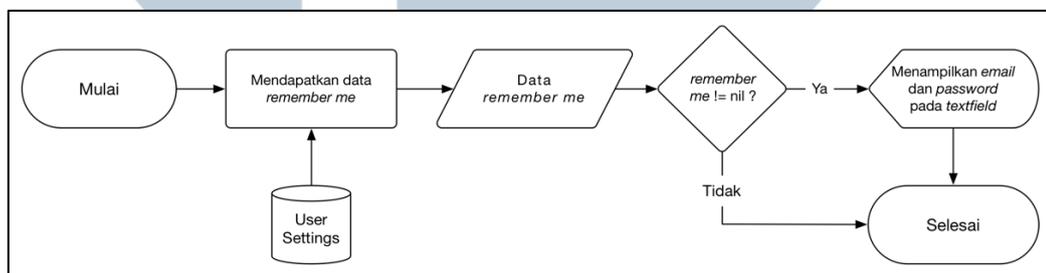
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.9 Flowchart Subproses Authentication Page

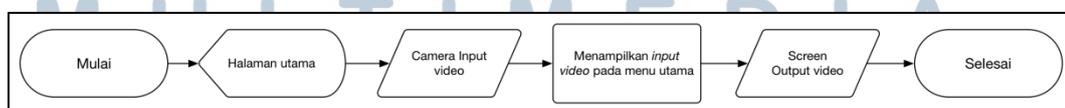
Gambar 3.9 merupakan *flowchart* untuk subproses Authentication Page. Subproses Authentication Page diawali dengan melakukan subproses Remember Me untuk mengecek apakah terdapat data Remember Me pada aplikasi. Kemudian, aplikasi akan mendapatkan input dari pengguna. Jika pengguna menekan tombol *login*, maka aplikasi akan mendapatkan data *email* dan *password* pengguna, kemudian aplikasi akan melakukan proses untuk mendapatkan data *user* dari *database*. Jika *user* tidak ditemukan, aplikasi akan menampilkan *error message* “*Email doesn't exist*”. Sebaliknya, jika *user* ditemukan aplikasi akan mengecek apakah *password* yang di input oleh pengguna cocok dengan *password* pada *user* yang didapat dari *database*. Jika tidak cocok, maka aplikasi akan menampilkan *error message* “*Incorrect password*”. Jika *password* cocok, maka aplikasi akan mendapatkan output data *user* dan flow untuk subproses Authentication telah selesai. Jika pengguna menekan tombol *register*, maka aplikasi akan mendapatkan

data *email* dan *password* pengguna. Selanjutnya, aplikasi akan mencoba terlebih dahulu untuk mendapatkan *user* dengan email yang telah di input, jika *user* ditemukan maka aplikasi menampilkan *error message* “*Email already registered*”. Jika *user* tidak ditemukan, maka aplikasi akan melakukan proses *mencreate user* dan menyimpannya pada *database* lalu menampilkan *info message* “*The user was successfully created*” dan flow untuk subproses Authentication Page telah selesai. Jika pengguna menekan tombol *forgot password*, maka aplikasi akan mendapatkan data *email* pengguna. Aplikasi akan melakukan proses untuk mendapatkan *user* dari *database* berdasarkan *input email* pengguna. Jika *user* tidak ditemukan, aplikasi akan menampilkan *error message* “*Email doesn’t exist*”. Jika *user* ditemukan, maka aplikasi akan mengirimkan *reset password link* ke *email* pengguna.



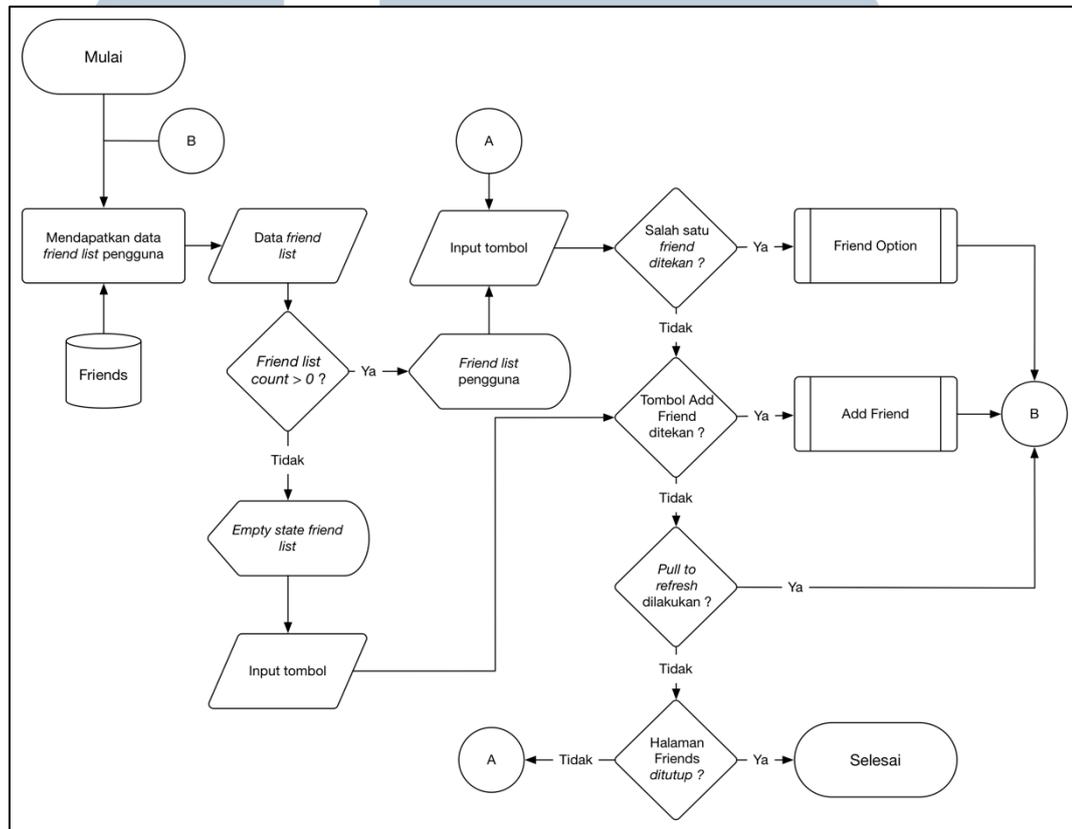
Gambar 3.10 Flowchart Subproses Remember Me

Gambar 3.10 merupakan *flowchart* subproses Remember Me. Subproses Remember Me diawali dengan aplikasi melakukan proses mendapatkan data *remember me* dari *offline database*. Kemudian, aplikasi akan mengecek apakah data *remember me* bernilai *nil*. Jika terdapat data *remember me*, maka aplikasi akan menampilkan *email* dan *password* pengguna pada *textfield*. Sebaliknya, jika data *remember me* adalah *nil* maka flow subproses Remember Me berakhir.



Gambar 3.11 Flowchart Subproses Main Page

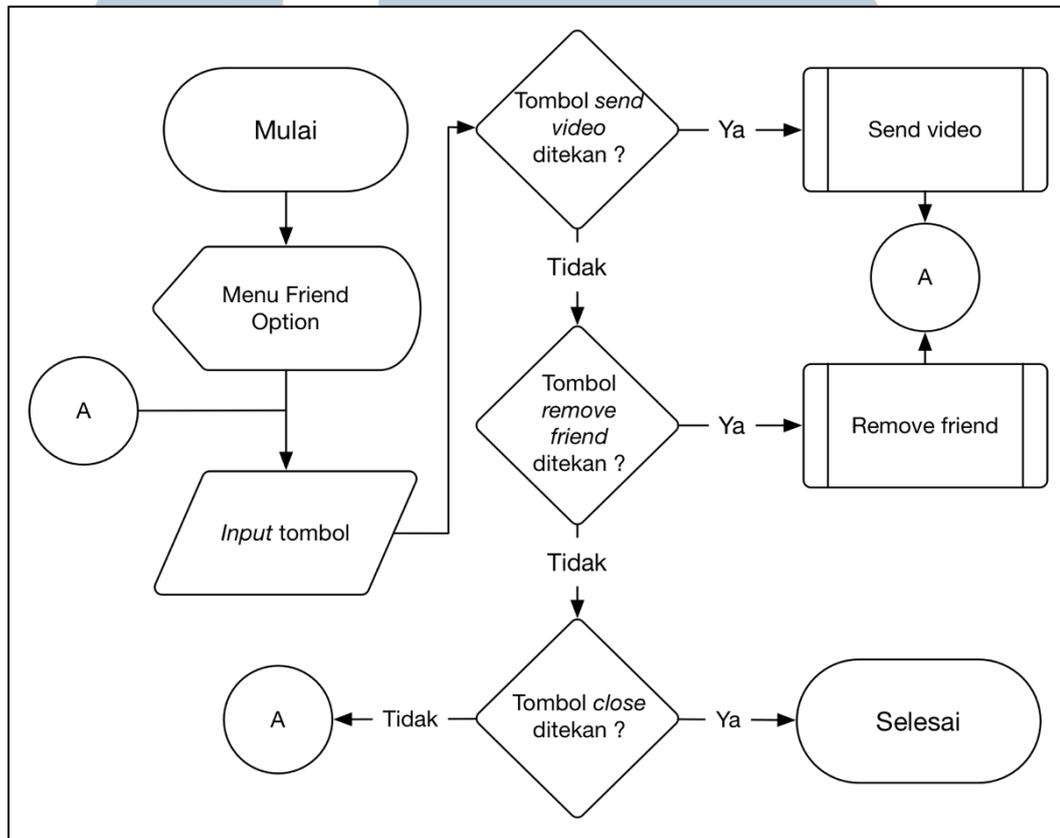
Gambar 3.11 merupakan flowchart untuk subproses Main Page. Subproses Main Page diawali dengan menampilkan halaman utama. Kemudian, aplikasi akan mendapatkan *input* berupa video yang akan di proses menjadi *output* video yang ditampilkan pada Main Page.



Gambar 3.12 Flowchart Subproses Friends Page

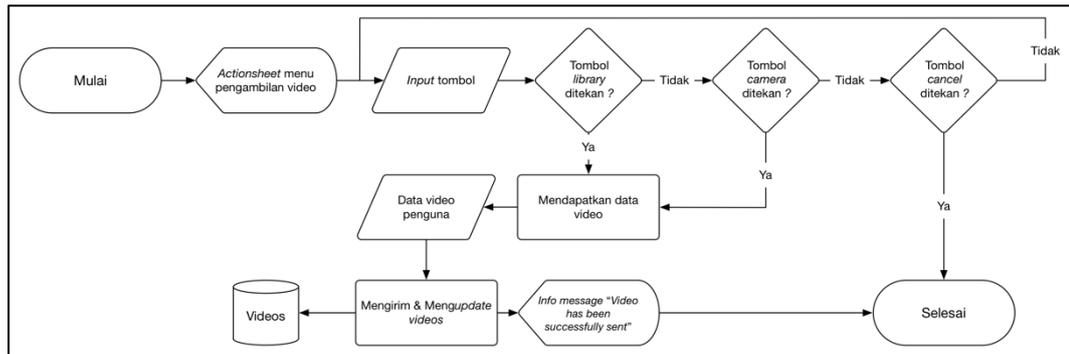
Gambar 3.12 merupakan *flowchart* untuk subproses Friends Page. Subproses Friends dimulai dengan aplikasi melakukan proses mendapatkan data *friend list* pengguna dari *database*. Kemudian, aplikasi akan mengecek apakah jumlah data *friend list* lebih dari nol. Jika lebih dari nol, maka aplikasi akan menampilkan *friend list* pengguna dan selanjutnya akan menerima input dari pengguna. Jika pengguna menekan salah satu *friend*, maka aplikasi akan melakukan subproses Friend Option. Jika pengguna menekan tombol *add friend*, maka aplikasi akan melakukan subproses Add Friend. Jika pengguna melakukan *pull to refresh*,

maka aplikasi akan kembali untuk mendapatkan data *friend list* pengguna dari *database*. Jika data *friend list* adalah nol atau kosong, maka aplikasi akan menampilkan *empty state* untuk *friend list*. Flow subproses Friends Page berakhir saat pengguna menutup Friends Page.



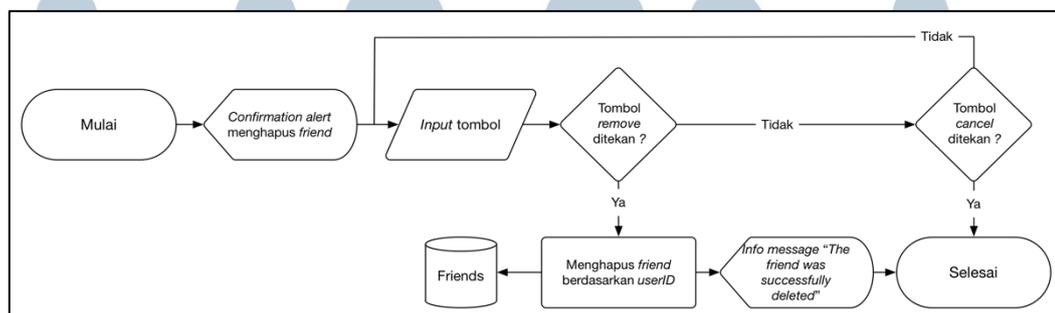
Gambar 3.13 Flowchart Subproses Friend Option

Gambar 3.13 merupakan *flowchart* untuk subproses Friend Option. Subproses Friend Option dimulai dengan menampilkan menu *friend option*. Kemudian, aplikasi akan menerima input dari pengguna. Jika pengguna menekan tombol *send video*, maka aplikasi akan melakukan subproses Send Video. Jika pengguna menekan tombol Remove Friend, maka aplikasi akan melakukan subproses Remove Friend. Flow subproses Friend Option berakhir saat pengguna menekan tombol *close*.



Gambar 3.14 Flowchart Subproses Send Video

Gambar 3.14 merupakan *flowchart* untuk subproses Send Video. Subproses Send Video dimulai dengan aplikasi akan menampilkan sebuah *actionsheet menu* pengambilan video. Kemudian, aplikasi akan mendapatkan input dari pengguna. Jika pengguna menekan tombol *library* atau menekan tombol *camera*, aplikasi akan melakukan proses mendapatkan data video. Kemudian, aplikasi akan melakukan proses mengirim dan *update* video tersebut kepada *friend* yang telah dipilih dan menyimpannya pada *database*. Setelah itu, aplikasi akan menampilkan *info message* “*Video has been successfully sent*”. Subproses Send Video berakhir saat pengguna berhasil mengirim video atau pengguna menekan tombol *cancel*.



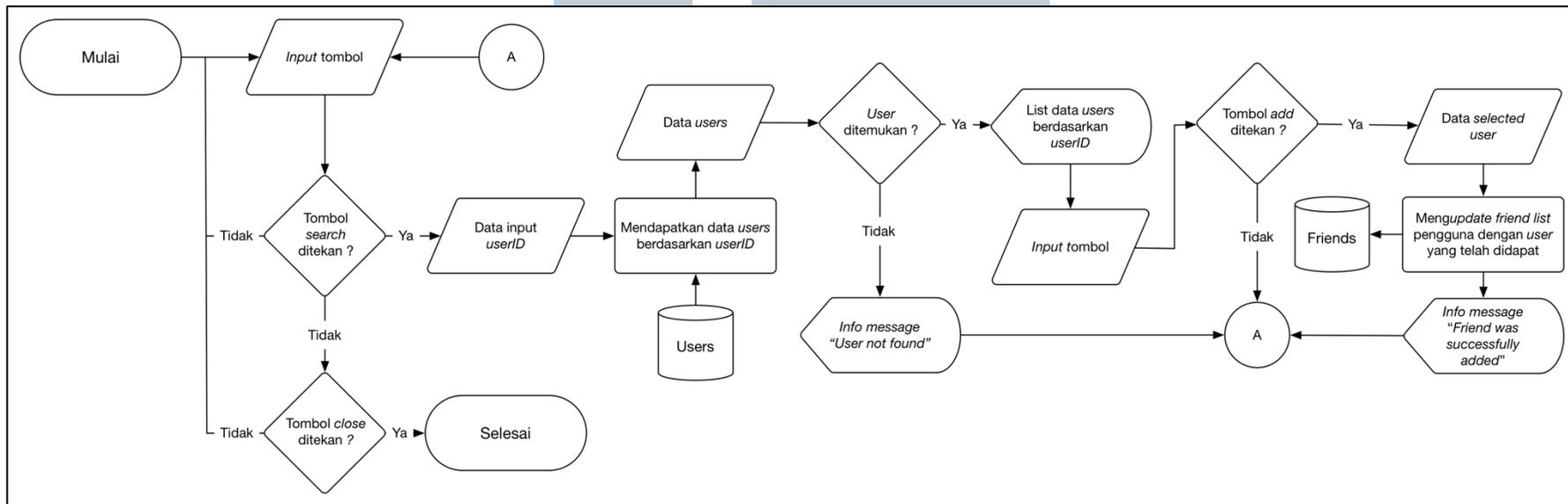
Gambar 3.15 Flowchart Subproses Remove Friend

Gambar 3.15 merupakan *flowchart* untuk subproses Remove Friend. Subproses Remove Friend dimulai dengan aplikasi akan menampilkan *confirmation alert* untuk menghapus *friend*. Aplikasi akan mendapatkan input dari pengguna, jika pengguna menekan tombol *remove* maka aplikasi akan melakukan

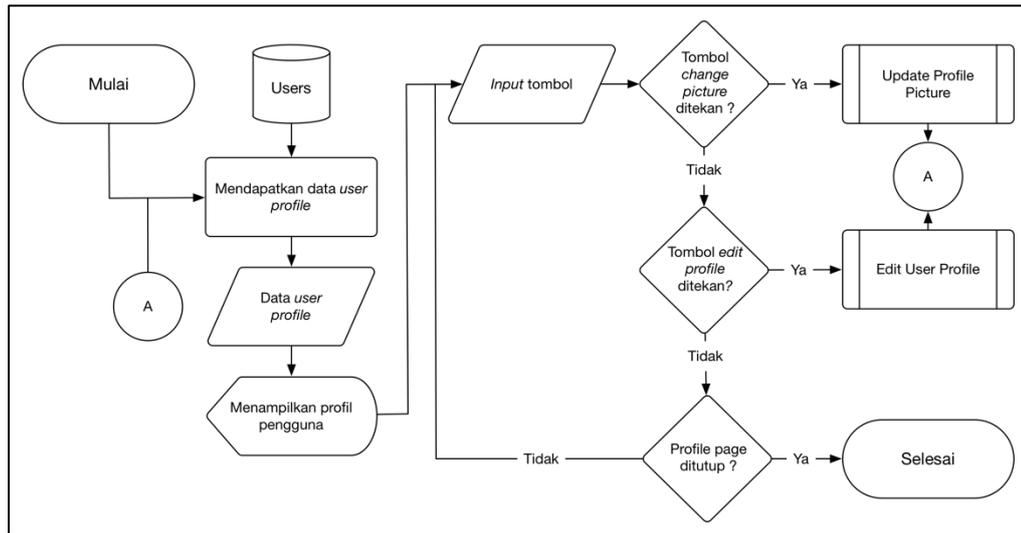
proses menghapus *friend* berdasarkan *userID* yang ada pada *database*. Kemudian, aplikasi akan menampilkan *info message* “*The friend was successfully removed*”. Flow subproses Remove Friend berakhir saat pengguna selesai menghapus *friend* atau pengguna menekan tombol *cancel*.

Gambar 3.16 merupakan *flowchart* untuk subproses Add Friend yang dimulai dengan aplikasi akan mendapatkan input dari pengguna. Jika pengguna menekan tombol *search*, maka aplikasi akan mendapatkan data input *userID*. Kemudian aplikasi akan melakukan proses untuk mendapatkan data *user* berdasarkan *userID* pada *database*. Selanjutnya aplikasi akan mengecek, jika *user* ditemukan maka aplikasi akan menampilkan data *users* berdasarkan *userID*. Jika *user* tidak ditemukan maka aplikasi akan menampilkan *info message* “*User not found*”. Kemudian, jika pengguna menekan tombol *add* pada salah satu *user* yang telah di cari, maka aplikasi akan mendapatkan data *user* yang telah dipilih kemudian aplikasi melakukan proses *update friend list* pengguna dengan data *friend* tersebut kedalam *database*. Kemudian, aplikasi akan menampilkan *info message* “*Friend was successfully added*”. Flow subproses Add Friend akan berakhir saat pengguna menekan tombol *close*.





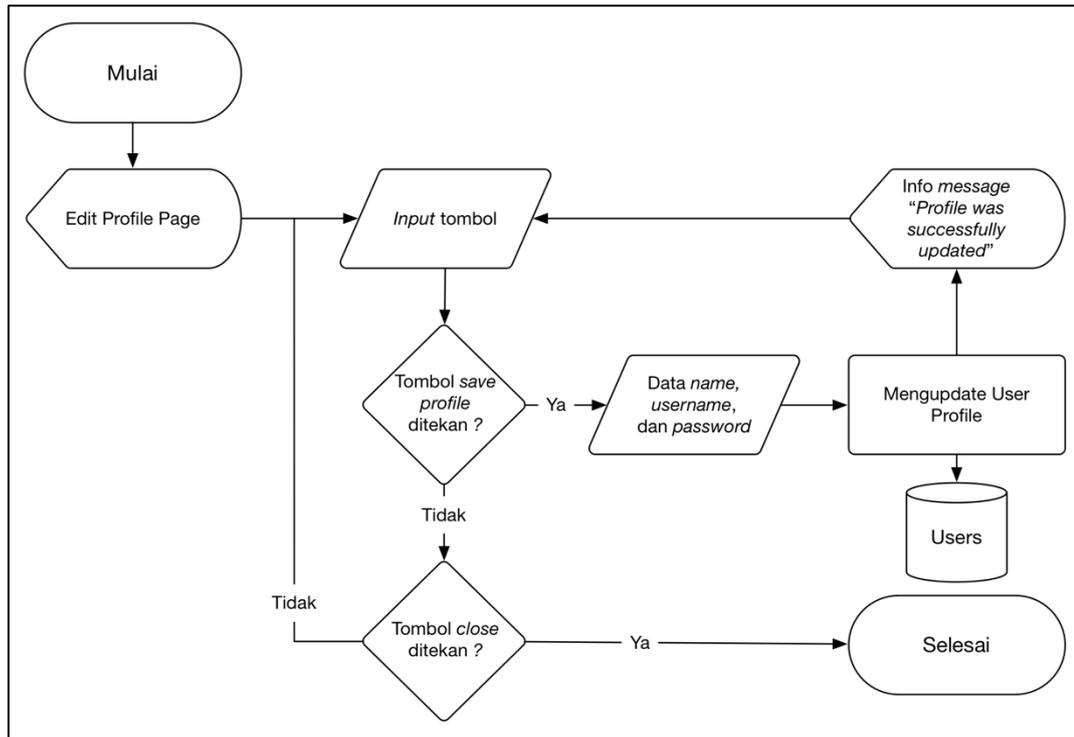
Gambar 3.16 Flowchart Subproses Add Friend



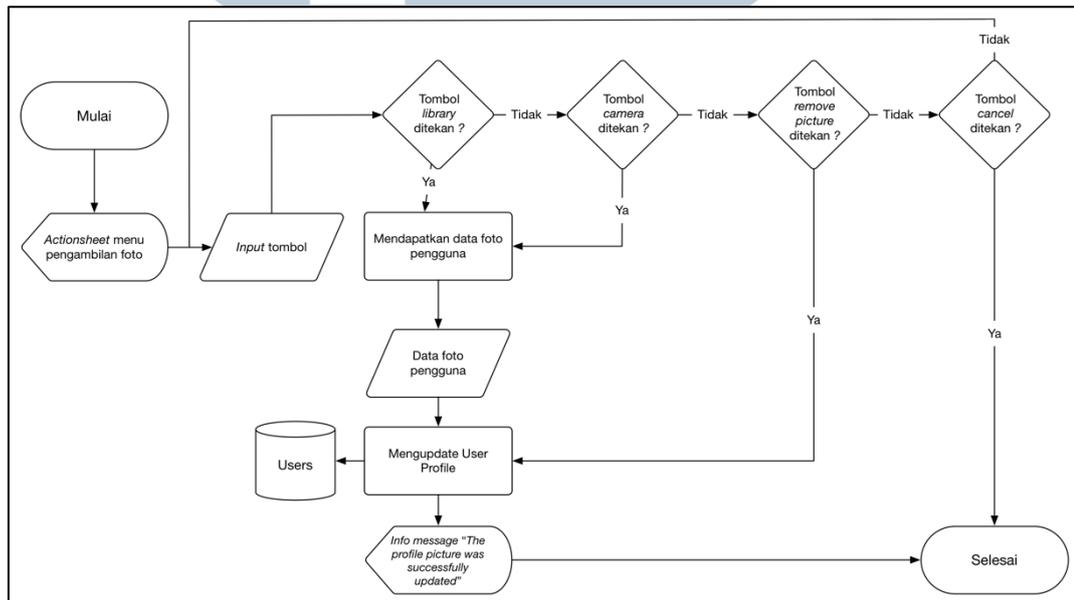
Gambar 3.17 Flowchart Subproses Profile Page

Gambar 3.17 merupakan *flowchart* untuk subproses Profile Page. Subproses Profile Page dimulai dengan aplikasi melakukan proses untuk mendapatkan data *user profile* dari *database* dan kemudian menampilkan *profile* pengguna. Selanjutnya, aplikasi akan menerima input pengguna, jika pengguna menekan tombol *change picture*, maka aplikasi akan melakukan subproses Update Profile Picture. Jika pengguna menekan tombol *edit profile*, maka aplikasi akan melakukan subproses Edit User Profile. Flow subproses Profile berakhir saat pengguna menutup Profile Page.

Gambar 3.18 merupakan *flowchart* untuk subproses Update User Profile. Subproses Update User Profile dimulai dengan aplikasi akan menampilkan Edit Profile Page, kemudian aplikasi akan menerima input dari pengguna. Jika pengguna menekan tombol *save profile*, maka aplikasi akan mendapatkan data *name*, *username*, dan *password* yang telah di *update* oleh pengguna. Kemudian, aplikasi akan melakukan proses mengupdate User Profile pada *database* dan menampilkan *info message* "Profile was successfully updated". Flow subproses Update User Profile akan berakhir saat pengguna menekan tombol *close*.



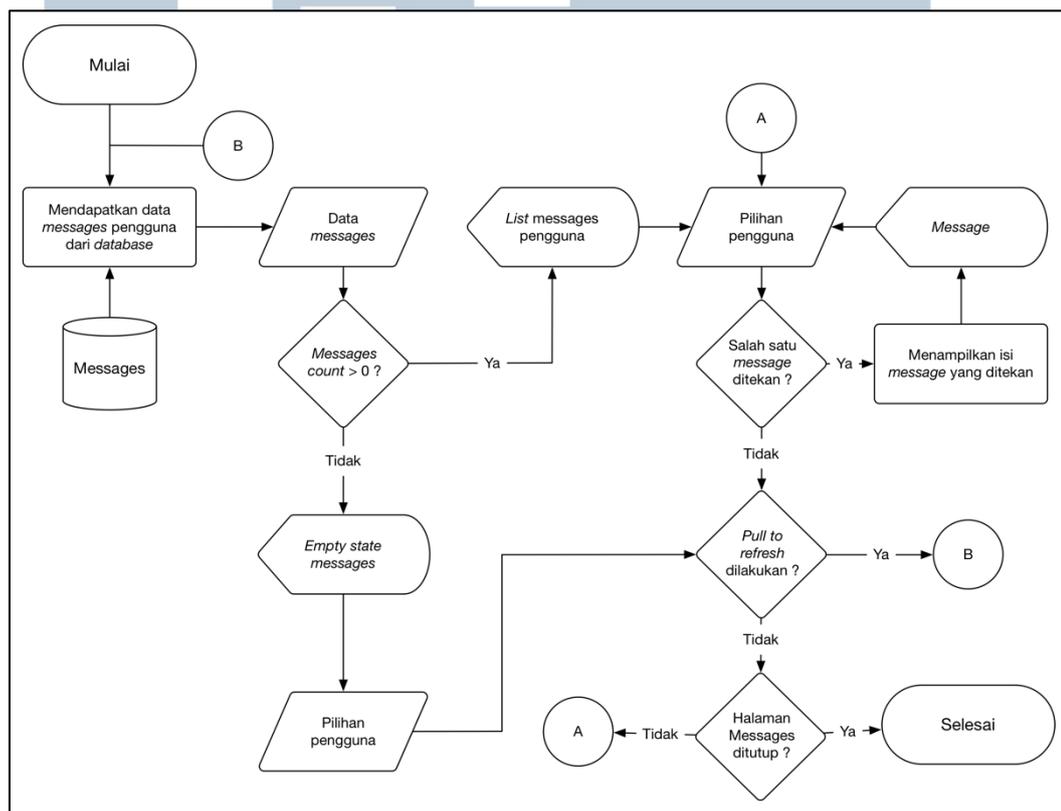
Gambar 3.18 Flowchart Subproses Update User Profile



Gambar 3.19 Flowchart Subproses Update Profile Picture

Gambar 3.19 merupakan *flowchart* untuk subproses Update Profile Picture. Subproses Update Profile Picture dimulai dengan aplikasi menampilkan *actionsheet menu* pengambilan foto. Kemudian, aplikasi akan menerima input dari

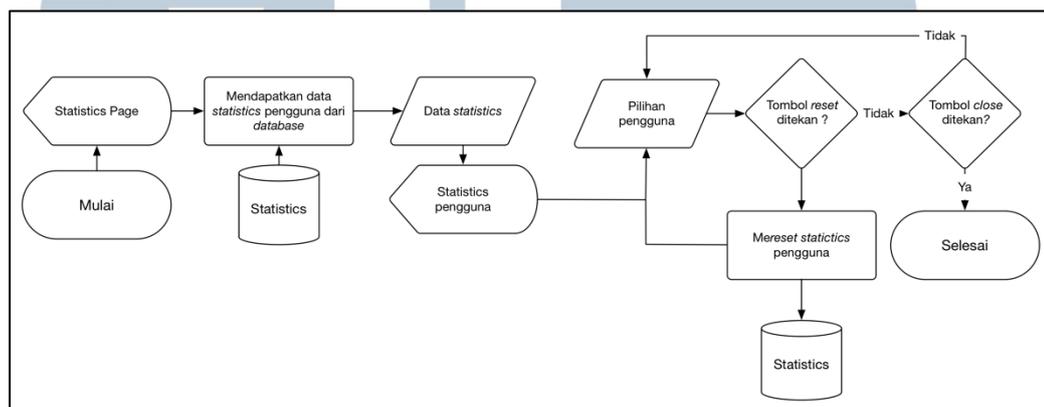
pengguna. Jika pengguna menekan tombol *library* atau tombol *camera*, maka aplikasi akan melakukan proses mendapatkan data foto pengguna kemudian melakukan proses *update user profile* ke *database* dan menampilkan *info message* “*The profile picture was successfully updated*”. Jika pengguna menekan tombol *remove picture*, aplikasi akan melakukan *update user profile* dengan foto profil *default*. Flow subproses Update Profile Picture berakhir saat pengguna menekan tombol *cancel*.



Gambar 3.20 Flowchart Subproses Messages Page

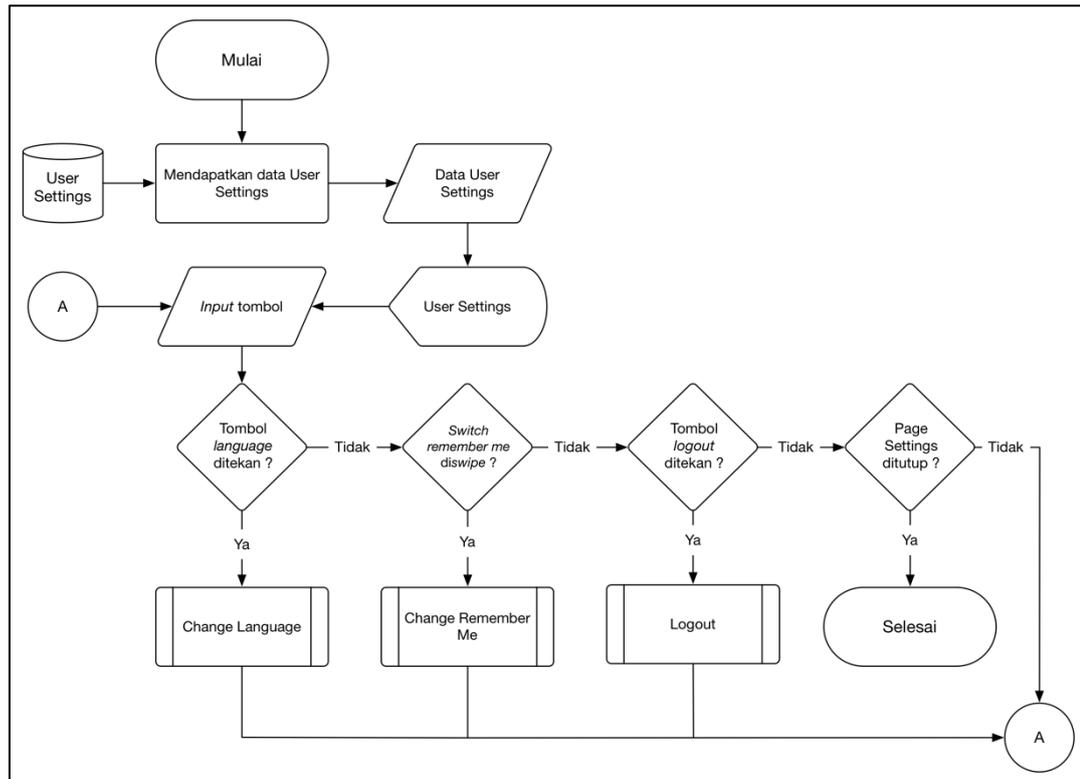
Gambar 3.20 merupakan *flowchart* untuk subproses Messages Page. Subproses Messages Page dimulai dengan aplikasi melakukan proses mendapatkan data *messages* pengguna dari *database*. Kemudian, dilakukan pengecekan pada jumlah *messages*. Jika jumlah *messages* lebih dari nol, maka aplikasi akan menampilkan *list messages* pengguna. Kemudian, aplikasi akan mendapatkan

pilihan selanjutnya dari pengguna. Jika salah satu *messages* ditekan, maka aplikasi akan melakukan proses menampilkan isi dari *message* yang telah ditekan. Jika pengguna melakukan *pull to refresh* maka, aplikasi akan mendapatkan data *messages* pengguna dari *database* kembali. Jika Jumlah *messages* pengguna adalah nol, maka aplikasi akan menampilkan *empty state messages*. Flow subproses Messages Page berakhir saat pengguna menutup Messages Page.



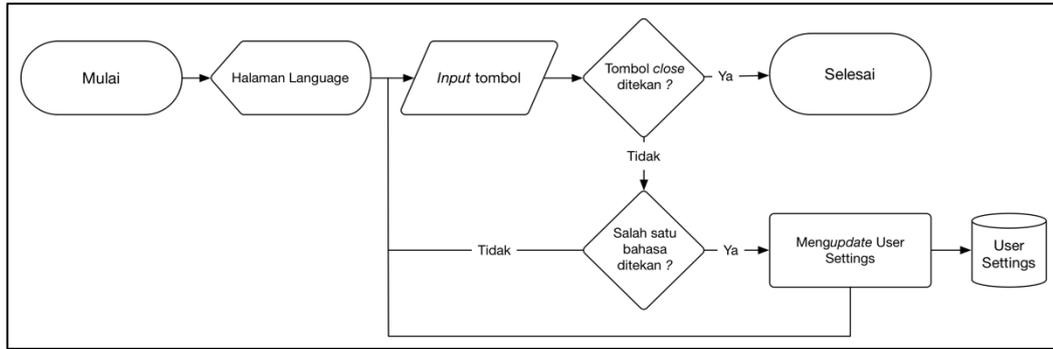
Gambar 3.21 Flowchart Subproses Statistics Page

Gambar 3.21 merupakan *flowchart* untuk subproses Statistics Page. Subproses Statistics Page dimulai dengan menampilkan Statistics Page kemudian aplikasi melakukan proses mendapatkan data *statistics* pengguna dari *database*. Selanjutnya, aplikasi akan menampilkan data *statistics* pada pengguna dan aplikasi akan mendapatkan input pilihan pengguna. Jika pengguna menekan tombol *reset*, maka aplikasi akan melakukan proses *mereset statistics* pengguna dan *update* kembali pada *database*. Flow subproses Statistics Page berakhir saat pengguna menekan tombol *close*.



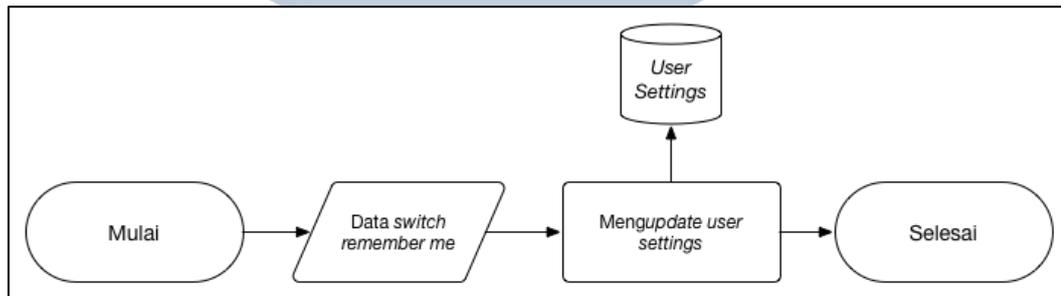
Gambar 3.22 Flowchart Subproses Settings Page

Gambar 3.22 merupakan *flowchart* untuk subproses Settings Page. Subproses Settings Page dimulai dengan aplikasi melakukan proses mendapatkan data User Settings dari *offline database* dan kemudian menampilkannya. Selanjutnya, aplikasi akan menerima input dari pengguna. Jika pengguna menekan tombol *language*, maka aplikasi akan melakukan subproses Change Language. Jika pengguna menekan tombol Logout, maka aplikasi akan melakukan subproses Logout. Jika pengguna men-*swipe switch remember me*, aplikasi akan melakukan subproses Change Remember Me. Flow subproses Settings Page berakhir saat pengguna menutup Settings Page.



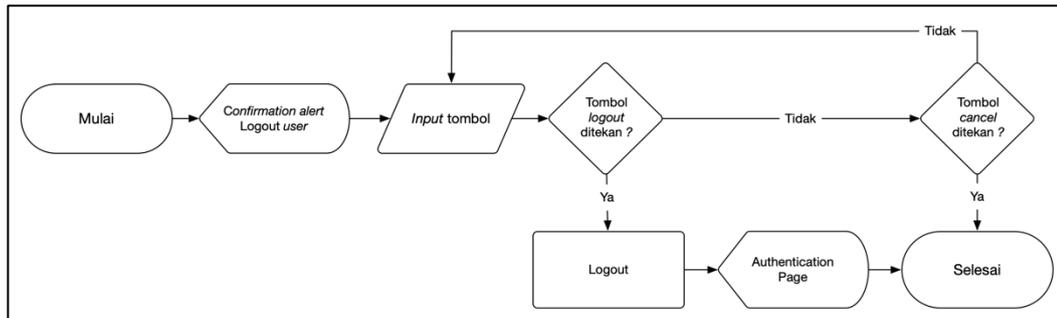
Gambar 3.23 Flowchart Subproses Change Language

Gambar 3.23 merupakan *flowchart* untuk subproses Change Language. Subproses Change Language dimulai dengan aplikasi menampilkan halaman Language kemudian aplikasi akan mendapatkan input dari pengguna. Jika pengguna menekan salah satu bahasa, maka aplikasi akan melakukan proses *update user settings* ke *offline database*. Flow subproses Change Language berakhir saat pengguna menekan tombol *close*.



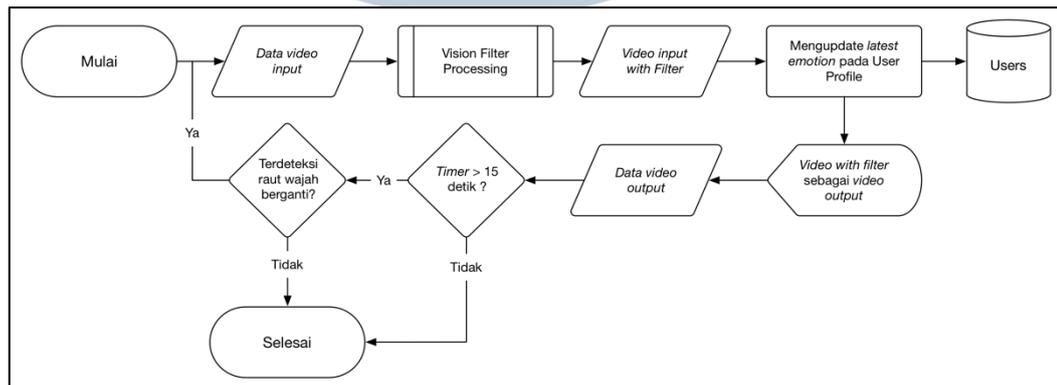
Gambar 3.24 Flowchart Subproses Change Remember Me

Gambar 3.24 merupakan *flowchart* subproses Change Remember Me. Subproses Change Remember Me dimulai dengan aplikasi menerima input data *switch remember me* yang telah di *swipe* kemudian aplikasi akan melakukan proses *update user settings* ke *offline database*.



Gambar 3.25 Flowchart Subproses Logout

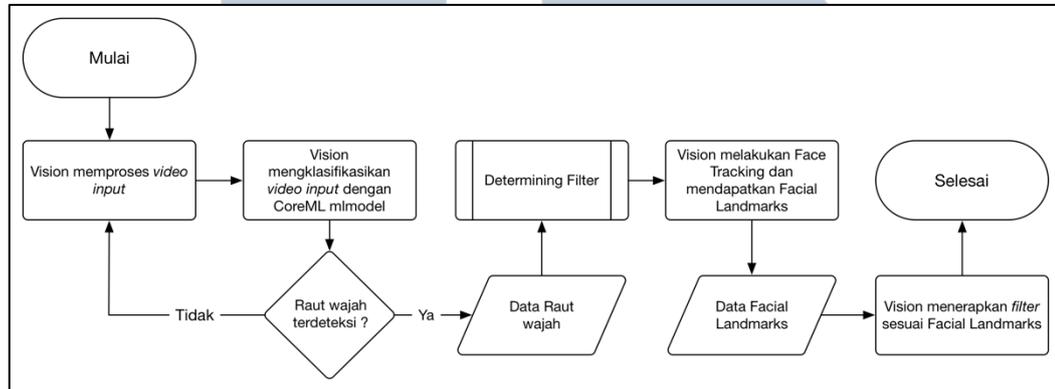
Gambar 3.25 merupakan *flowchart* subproses Logout. Subproses Logout dimulai dengan aplikasi menampilkan *confirmation alert* sebagai konfirmasi untuk melakukan *logout*. Kemudian aplikasi akan menerima input dari pengguna, jika pengguna menekan tombol *logout*, maka aplikasi akan melakukan proses Logout dan pengguna diarahkan kembali pada Authentication Page. Flow subproses Logout akan berakhir saat pengguna telah *logout* atau pengguna menekan tombol *cancel*.



Gambar 3.26 Flowchart Subproses Deteksi Raut Wajah

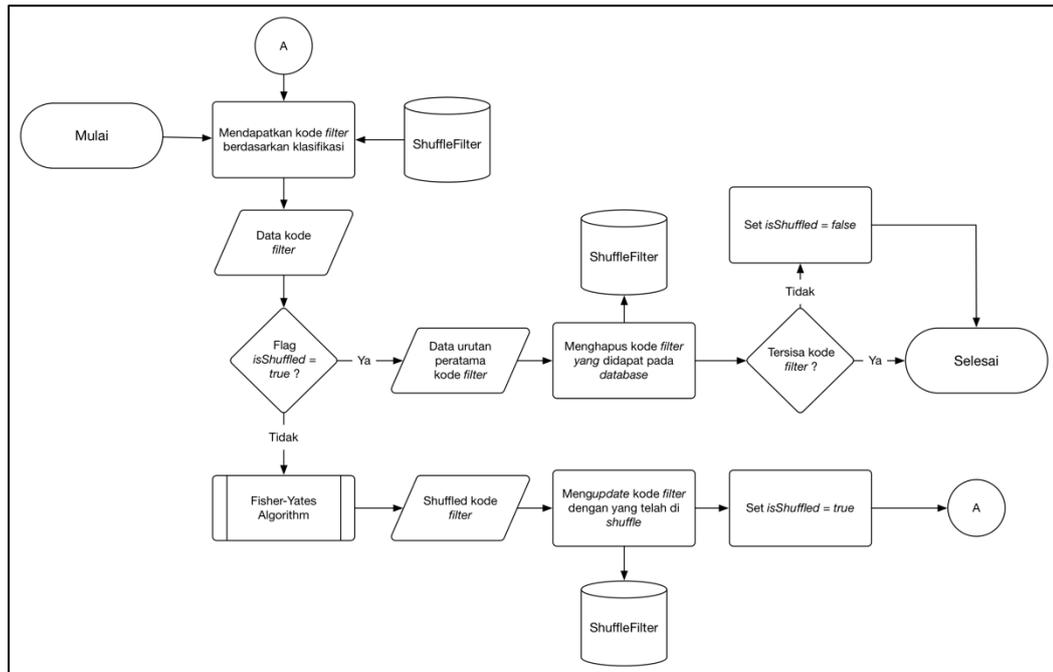
Gambar 3.26 merupakan *flowchart* subproses Deteksi Raut Wajah. Subproses Deteksi Raut Wajah dimulai dengan aplikasi menerima input data video. Kemudian, aplikasi akan melakukan subproses Vision Filter Processing untuk mendapatkan video input dengan *filter* dan menampilkannya. Jika terdeteksi raut wajah yang berganti, maka aplikasi akan memulai kembali untuk mendapatkan

input video dan melakukan subproses Vision Filter Processing seperti sebelumnya. Flow subproses Deteksi Raut Wajah berakhir saat timer telah mencapai 15 detik atau tidak terdeteksi raut wajah berganti.



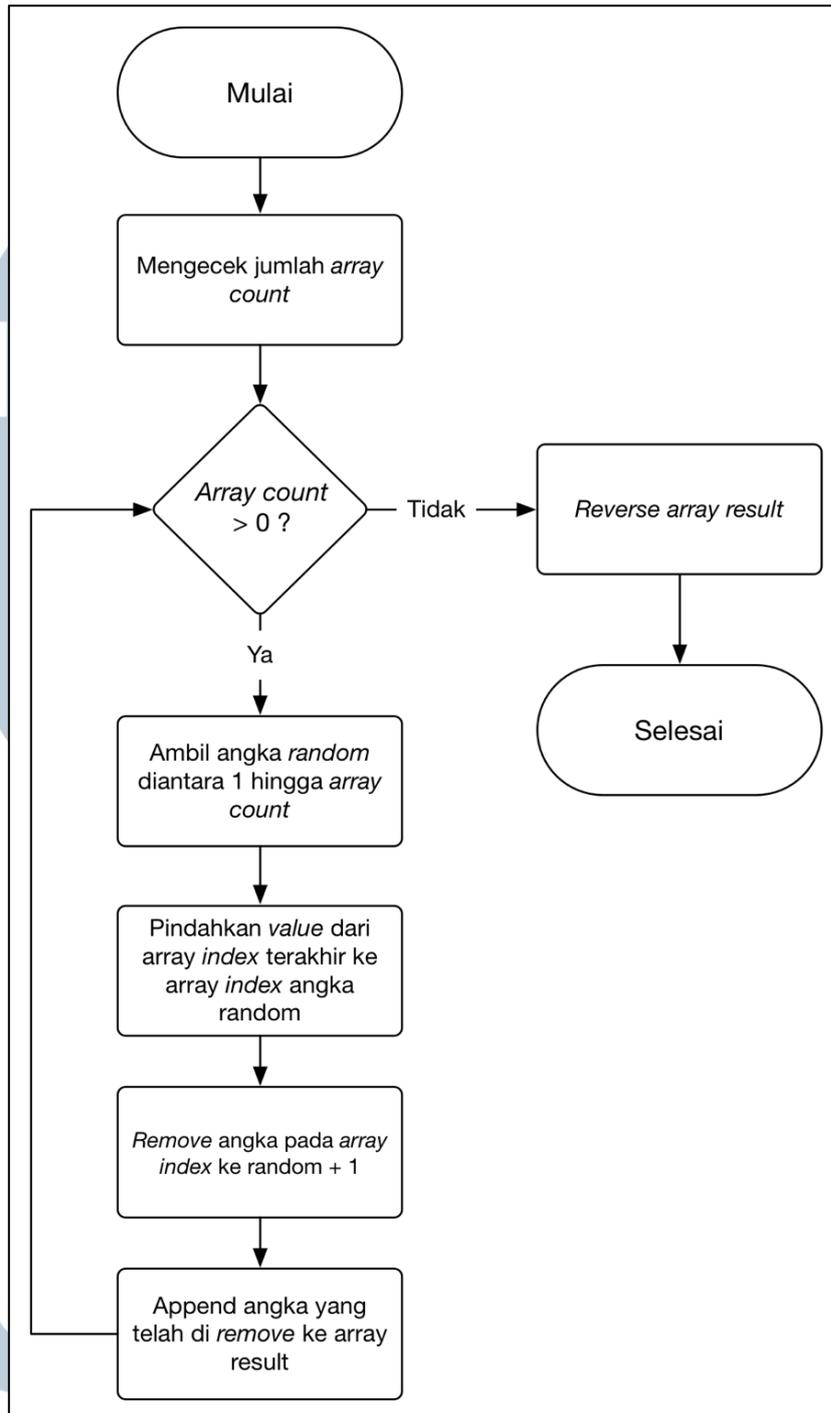
Gambar 3.27 Flowchart Subproses Vision Filter Processing

Gambar 3.27 merupakan *flowchart* subproses Vision Filter Processing. Subproses Vision Filter Processing dimulai dengan aplikasi melakukan proses input video. Kemudian, Vision melakukan proses untuk mengklasifikasi input video berupa raut wajah tersebut dengan *mlmodel* yang merupakan hasil *training*. Selanjutnya, apabila hasil dari klasifikasi tersebut tidak terdapat raut wajah yang terdeteksi, maka aplikasi akan kembali memproses input video untuk mengklasifikasikan ulang. Jika hasil dari klasifikasi terdapat raut wajah yang terdeteksi, maka aplikasi akan mendapatkan output data raut wajah. Selanjutnya, output data raut wajah tersebut digunakan untuk mendapatkan *filter* sesuai dengan raut wajah yang terdeteksi pada subproses Determining Filter. Setelah mendapatkan *filter*, Vision akan melakukan Face Tracking untuk mendapatkan data *facial landmarks* dan kemudian Vision akan menerapkan *filter* sesuai dengan raut wajah pada *facial landmarks* pengguna.



Gambar 3.28 Flowchart Subproses Determining Filter

Gambar 3.28 merupakan *flowchart* subproses Get Filter. Subproses Get Filter dimulai dengan aplikasi melakukan proses mendapatkan kode *filter* berdasarkan klasifikasi raut wajah dari *offline database*. Setelah mendapatkan data kode *filter*, aplikasi akan mengecek nilai *flag isShuffled* dari kode *filter* tersebut. Jika *flag isShuffled* bernilai *true*, aplikasi akan mendapatkan data urutan pertama dari kode *filter*. Selanjutnya, aplikasi akan menghapus kode *filter* tersebut pada *offline database*. Kemudian, aplikasi mengecek kembali apakah tersisa kode *filter* setelah dilakukan proses penghapusan tersebut. Jika tidak terdapat kode *filter* setelah dilakukan proses penghapusan, aplikasi akan men-*set value* dari *flag isShuffled* dengan *false*. Jika setelah mendapatkan kode *filter* dan *value* dari *flag isShuffled* adalah *false*, maka aplikasi akan melakukan subproses Fisher-Yates Algorithm untuk mendapatkan kode *filter* yang telah *dishuffle* dan mengupdatenya kedalam *offline database* serta men-*set value* dari *flag isShuffled* dengan nilai *true*.



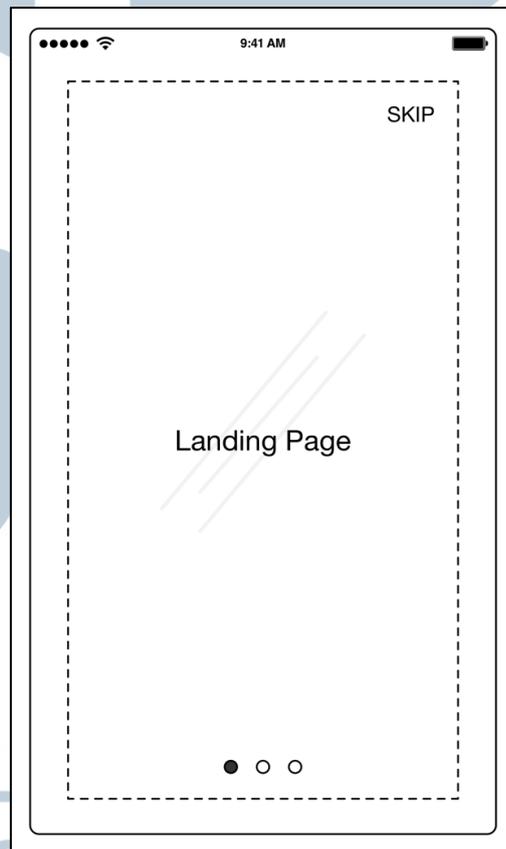
Gambar 3.29 Flowchart Subproses Fisher-Yates Algorithm

Gambar 3.29 merupakan *flowchart* subproses Fisher-Yates Algorithm. Subproses Fisher-Yates dimulai dengan mengecek apakah jumlah pada *array* lebih dari 0. Jika lebih dari 0, maka dilakukan pengambilan angka *random* diantara angka 1 hingga jumlah dari *array*. Kemudian, dilakukan pemindahan *value* dari *array*

index terakhir ke *array index* angka yang telah di *random*. Selanjutnya, dilakukan *remove value* dari *array index* ke *random + 1*. Kemudian, angka yang telah di *remove* tersebut di *append* kedalam *array result*. Setelah tidak tersisa angka pada *array*, maka dilakukan proses *reverse array* untuk membalikan hasil dari *array result*.

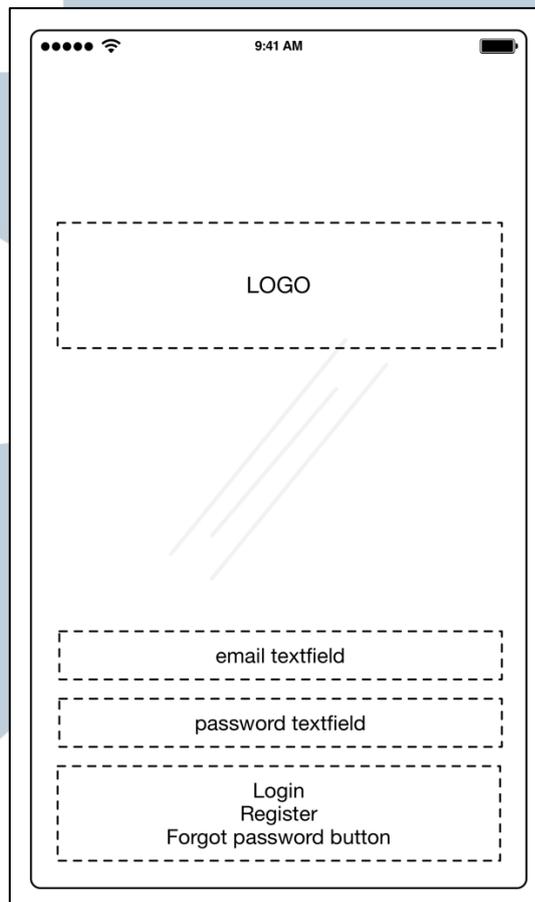
3.3 Rancangan Antarmuka

Pada rancangan antarmuka Landing Page, terdapat 3 slide sebagai *intro* atau pengenalan dari aplikasi. Untuk menuju Authentication Page, pengguna dapat melakukan *swipe* kearah kanan hingga *intro* dari aplikasi habis atau dengan menekan tombol *skip*. Rancangan antarmuka untuk Landing Page dapat dilihat pada gambar 3.30.



Gambar 3.30 Rancangan Antarmuka Landing Page

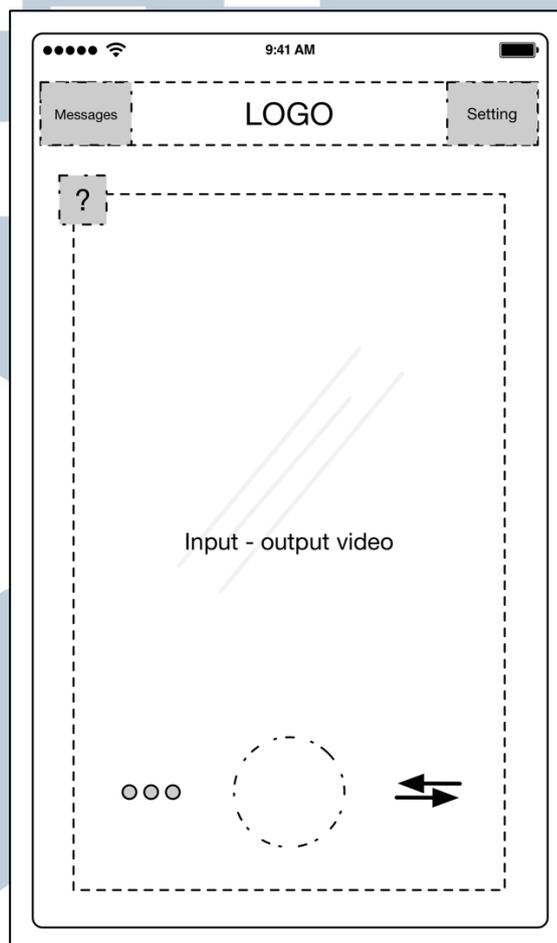
Pada rancangan antarmuka Authentication Page, terdapat *email textfield*, *password textfield*, *login button* dan pilihan *button* untuk melakukan *register* atau *forgot password*. Jika pengguna menekan *register button* maka data dari *email textfield* dan *password textfield* akan dijadikan data yang akan dikirim sebagai proses *register*. Jika pengguna menekan *forgot password button*, maka data yang dikirim hanyalah data dari *email textfield*, sedangkan *password textfield* akan di-*disable* karena tidak dibutuhkan dalam proses *forgot password*. Pada posisi tengah Authentication Page, terdapat logo dari aplikasi. Rancangan antarmuka untuk Authentication Page dapat dilihat pada gambar 3.31.



Gambar 3.31 Rancangan Antarmuka Authentication Page

Pada rancangan antarmuka Main Page, terdapat *messages button* pada kiri atas Main Page yang berfungsi untuk navigasi ke Messages Page. Pada kanan atas

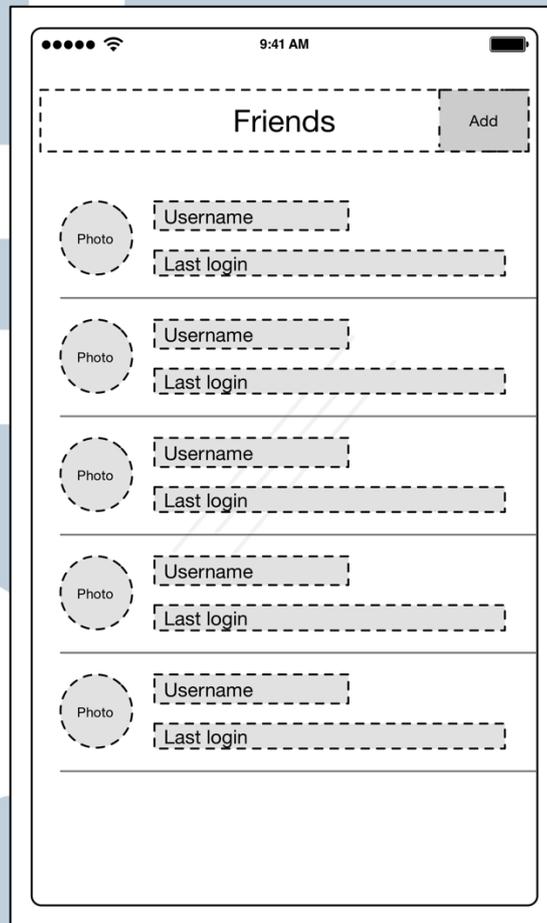
Main Page terdapat *settings button* untuk mengarahkan pengguna ke Settings Page. *About button* yang berada dibawah *messages button* berfungsi untuk navigasi ke About Page. Secara keseluruhan, tampilan Main Page adalah untuk menunjukkan output dari video pengguna baik yang belum terdapat *filter* maupun sudah terdapat *filter* berdasarkan raut wajah. Terdapat *record button* untuk merekam citra wajah pengguna yang dibatasi selama 15 detik. Rancangan antarmuka untuk Main Page dapat dilihat pada gambar 3.32.



Gambar 3.32 Rancangan Antarmuka Main Page

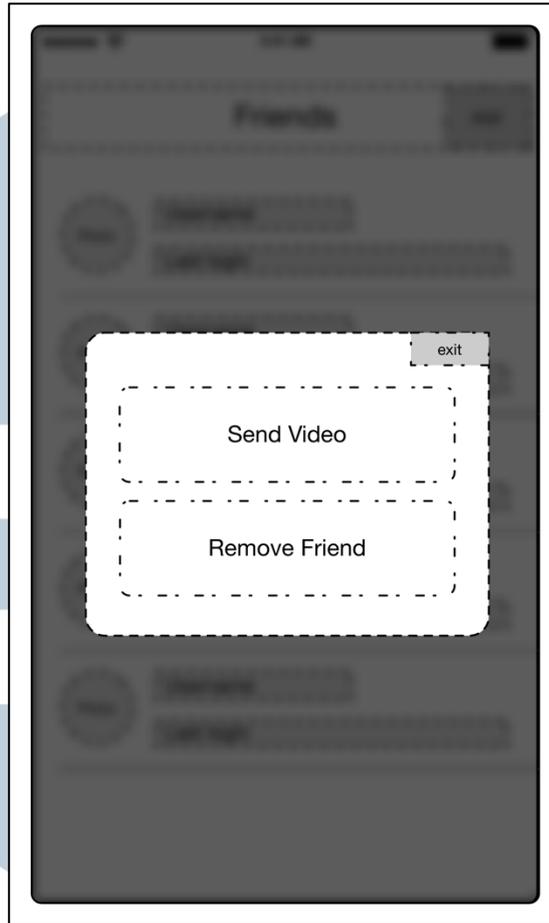
Pada rancangan antarmuka Friends Page, terdapat *add friend button* pada ujung kanan atas yang berfungsi untuk melakukan pencarian dan penambahan *friend*. Secara keseluruhan, tampilan Friends Page adalah untuk menunjukkan *friend*

list dari pengguna. *Friend list* dari pengguna menampilkan informasi seperti foto profil, nama *username* dan data *last login* dari *friends*. Pengguna dapat melakukan aksi Friend Option dengan cara men-*tap* salah satu *friend* pada *friend list*. Rancangan antarmuka untuk Friend Page dapat dilihat pada gambar 3.33.



Gambar 3.33 Rancangan Antarmuka Friends Page

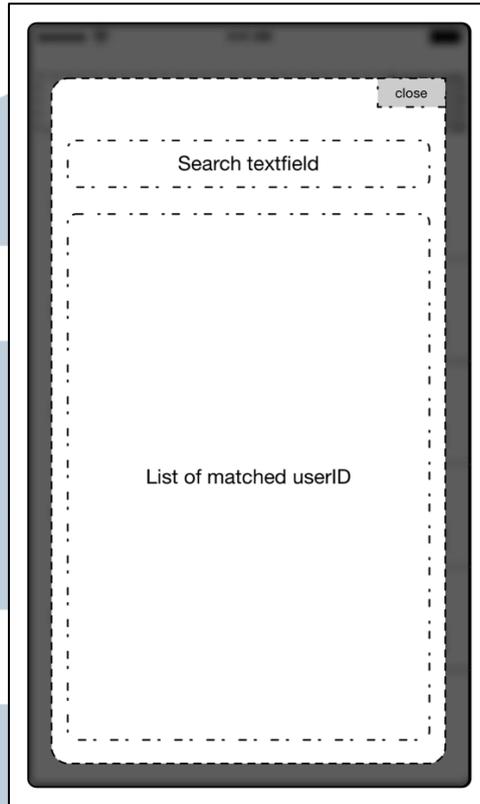
Pada rancangan antarmuka Friend Options Page, terdapat tombol *send video* untuk mengirimkan *video* kepada teman yang telah dipilih. Tombol *remove friend* berfungsi untuk menghapus teman yang dipilih oleh pengguna. Untuk keluar dari Friend Options Page, pengguna dapat menekan tombol *exit* pada ujung kanan atas. Rancangan antarmuka untuk Friend Options Page dapat dilihat pada gambar 3.34.



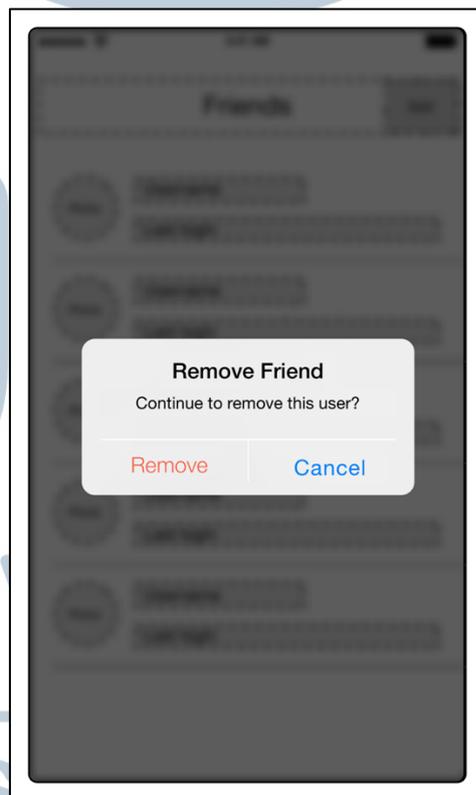
Gambar 3.34 Rancangan Antarmuka Friend Options Page

Pada rancangan antarmuka Add Friend, terdapat *textfield* untuk memasukan nama dari userID yang ingin dicari. Hasil pencarian teman dapat terlihat pada *tableview* dibawah *textfield* pencarian teman. Rancangan antarmuka untuk Add Friend Page dapat dilihat pada gambar 3.35.

Pada rancangan antarmuka Remove Friend Page, akan menampilkan *confirmation alert* sebagai tanda konfirmasi pada pengguna untuk menghapus *friend* yang telah dipilih. Jika pengguna menekan tombol *remove*, maka *friend* akan di hapus, jika pengguna menekan tombol *cancel*, maka proses menghapus *friend* akan dibatalkan. Rancangan antarmuka untuk Remove Friend Page dapat dilihat pada gambar 3.36.

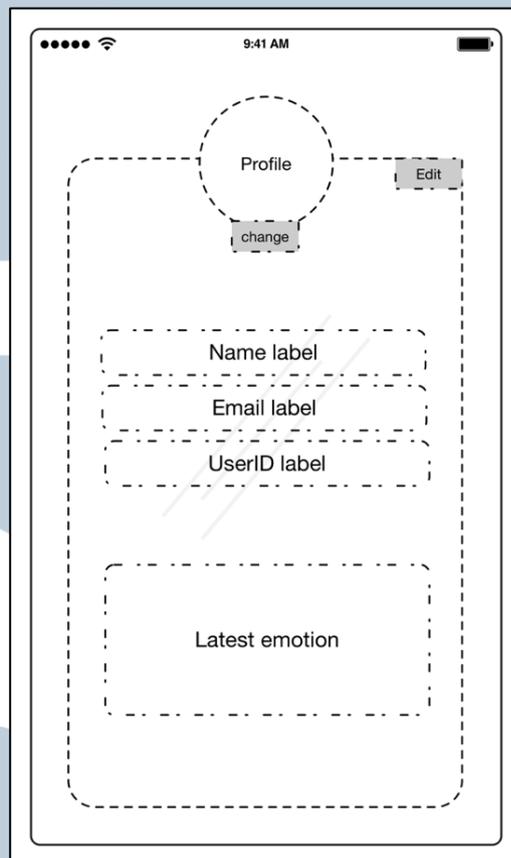


Gambar 3.35 Rancangan Antarmuka Add Friend Page



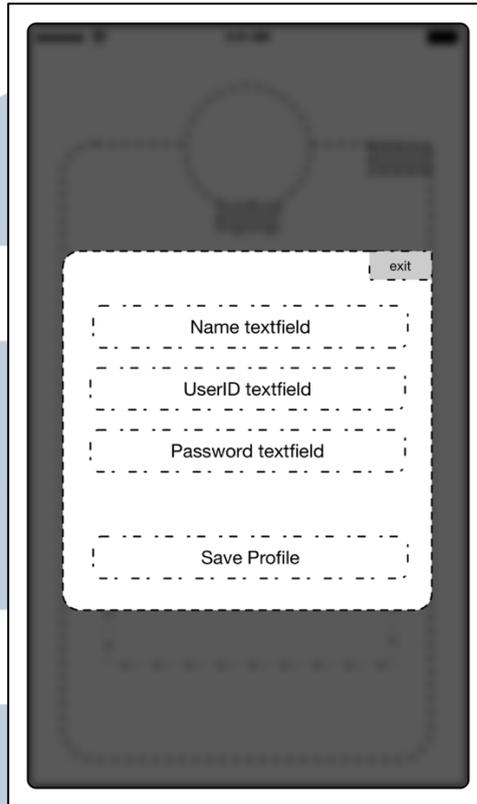
Gambar 3.36 Rancangan Antarmuka Remove Friend Page

Pada rancangan antarmuka Profile Page, terdapat *name label*, *email label*, *userID label* dan *lateset emotion*. Secara keseluruhan, Profile Page menampilkan informasi pengguna dan menampilkan list dari raut wajah terakhir yang pengguna rekam. Pengguna dapat menutup Profile Page dengan cara mengusap layar kearah kiri. Rancangan antarmuka untuk Profile Page dapat dilihat pada gambar 3.37.



Gambar 3.37 Rancangan Antarmuka Profile Page

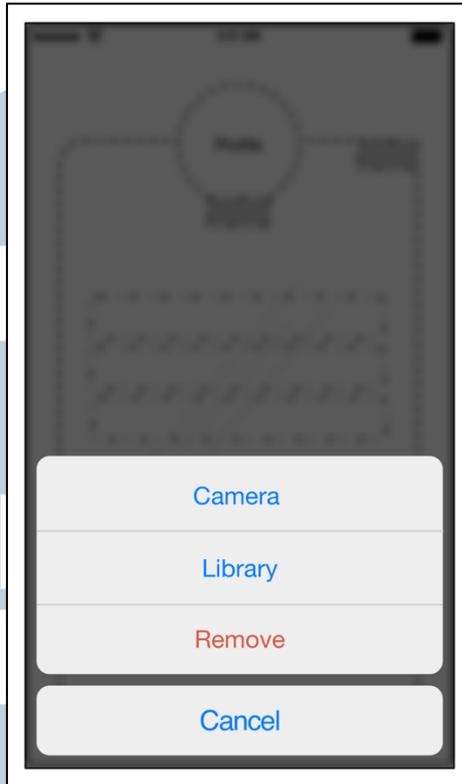
Pada rancangan antarmuka Edit Profile Page, terdapat *name textfield*, *userID textfield* dan *password textfield*. Secara keseluruhan, Edit Profile Page bertugas untuk mengupdate informasi pengguna. Pengguna dapat menutup Edit Profile Page dengan cara menekan tombol *exit* pada ujung kanan atas. Rancangan antarmuka untuk Edit Profile dapat dilihat pada gambar 3.38.



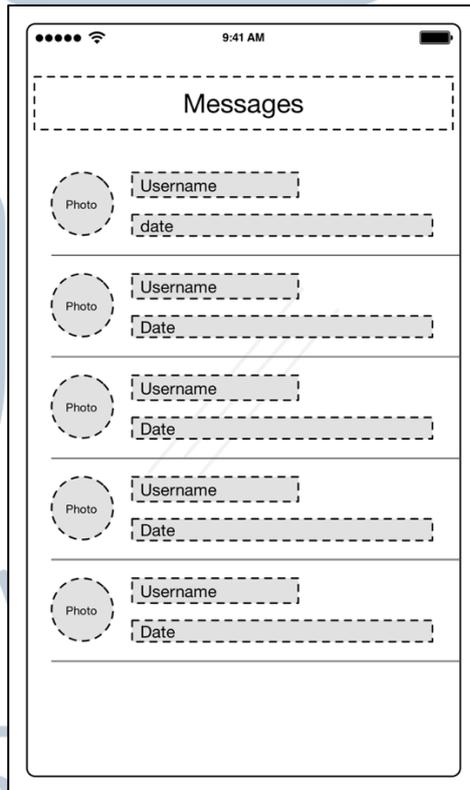
Gambar 3.38 Rancangan Antarmuka Edit Profile Page

Pada rancangan antarmuka Edit Photo Profile Page, terdapat *actionsheet* untuk memilih *input photo* yang ingin digunakan dalam mengupdate *foto profile*. Jika pengguna menekan tombol *camera*, maka aplikasi akan membuka *camera* dan mendapatkan foto dari pengguna. Jika pengguna menekan tombol *library*, maka aplikasi akan membuka *photo library* dari *device* pengguna. Jika pengguna menekan tombol *remove* maka aplikasi akan mengganti foto *profile* pengguna dengan foto *default*. Rancangan antarmuka untuk Edit Profile Page dapat dilihat pada gambar 3.39.

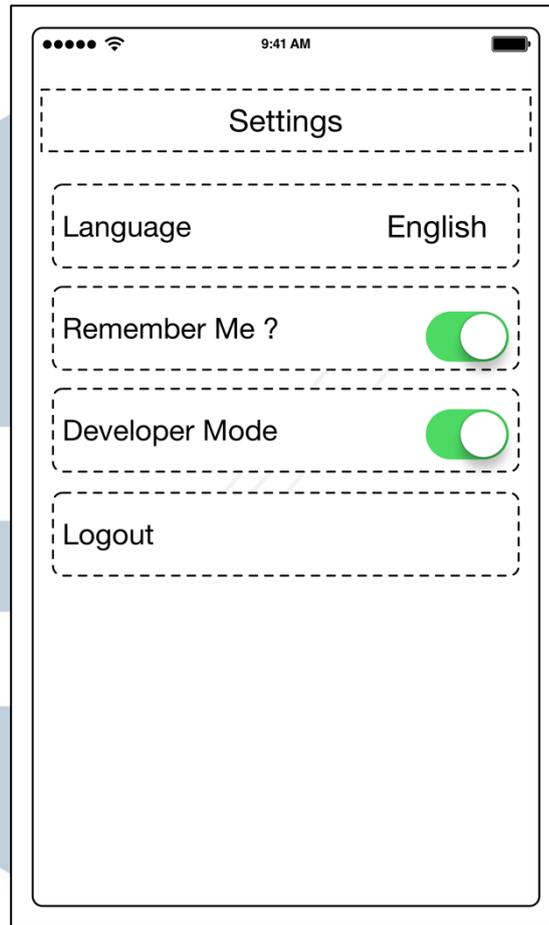
Pada rancangan antarmuka Messages Page, aplikasi akan menampilkan *list* dari *messages* yang dikirimkan kepada pengguna. Messages yang diterima adalah notifikasi teman baru dan *video* yang dikirimkan oleh teman pengguna. Rancangan antarmuka untuk Messages Page dapat dilihat pada gambar 3.40.



Gambar 3.39 Rancangan Antarmuka Edit Photo Profile



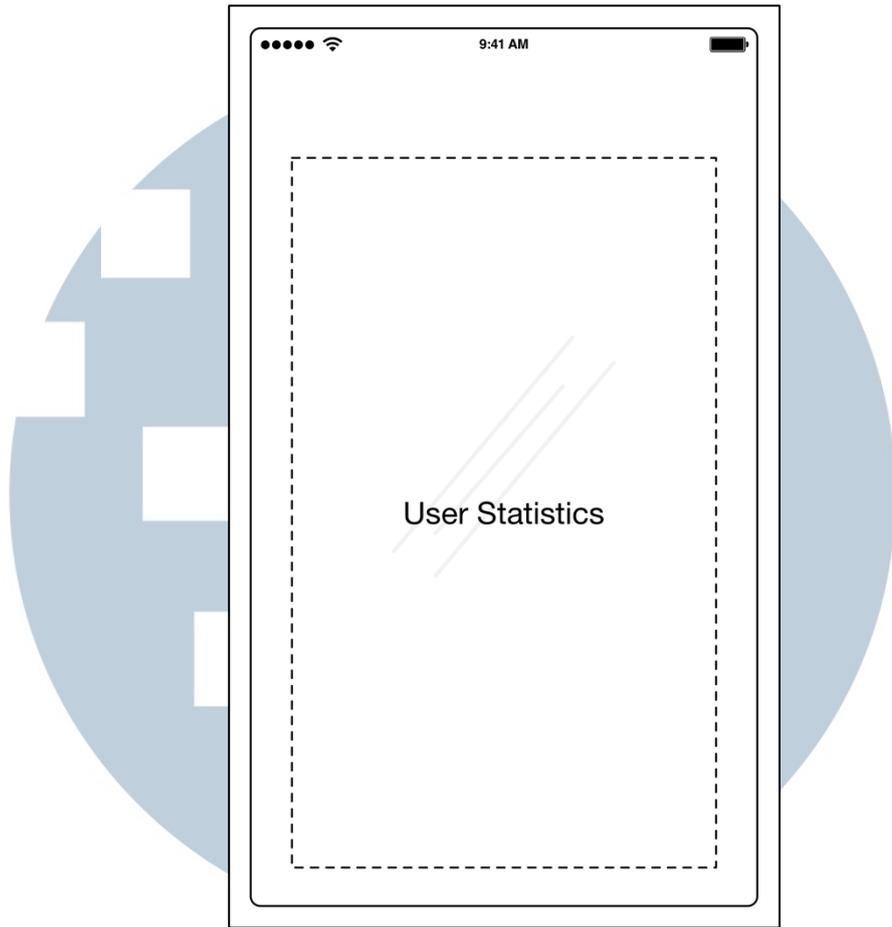
Gambar 3.40 Rancangan Antarmuka Messages Page



Gambar 3.41 Rancangan Antarmuka Settings Page

Pada rancangan antarmuka Settings Page, aplikasi akan menampilkan sebuah *tableview* yang berisi pilihan *setting language* untuk mengubah bahasa dari aplikasi, *switch remember me* untuk menyimpan data *email* dan *password* sehingga tidak perlu mengetik ulang *email* dan *password* pada Authentication Page, *switch developer mode* dan tombol Logout untuk melakukan proses Logout dan mengarahkan pengguna ke Authentication Page kembali. Rancangan antarmuka untuk Settings Page dapat dilihat pada gambar 3.41.

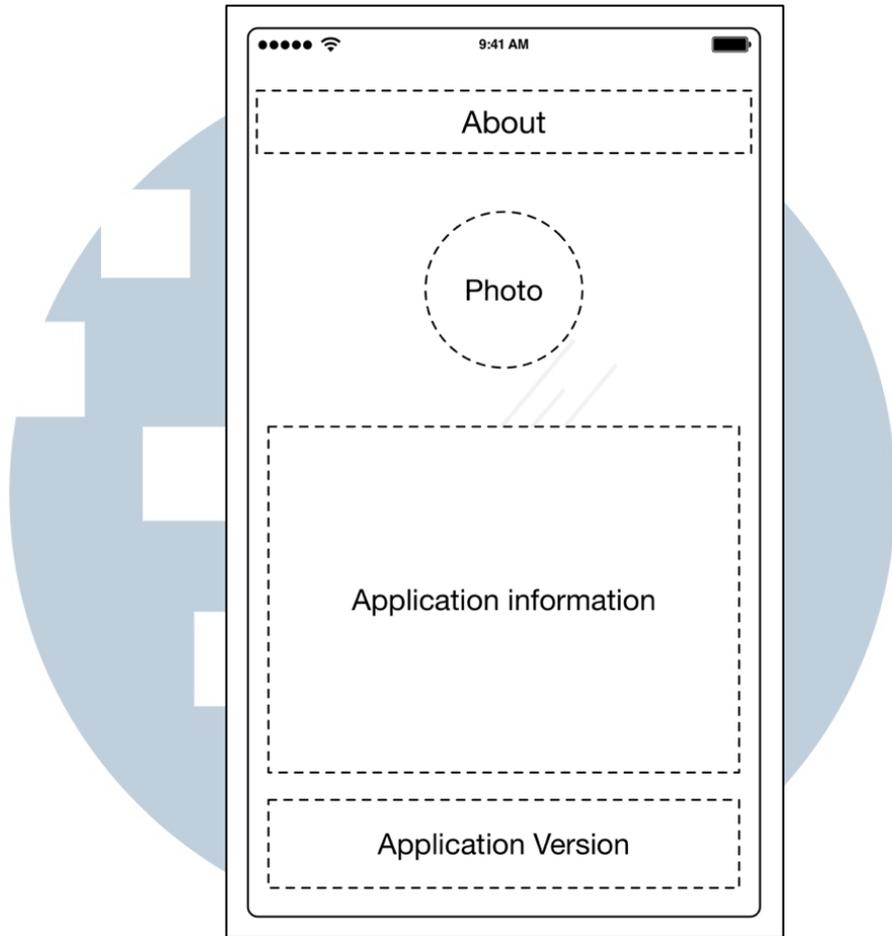
Pada rancangan antarmuka Statistics Page, aplikasi akan menampilkan sebuah grafik yang memperlihatkan data raut wajah yang pengguna telah lakukan. Rancangan antarmuka untuk Statistics Page dapat dilihat pada gambar 3.42.



Gambar 3.42 Rancangan Antarmuka Statistics Page

Pada rancangan antarmuka About Page, aplikasi akan menampilkan foto dari pembuat aplikasi, list dari *library* yang digunakan dan versi dari aplikasi. Rancangan antarmuka untuk About Page dapat dilihat pada gambar 3.43.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.43 Rancangan Antarmuka About Page

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA