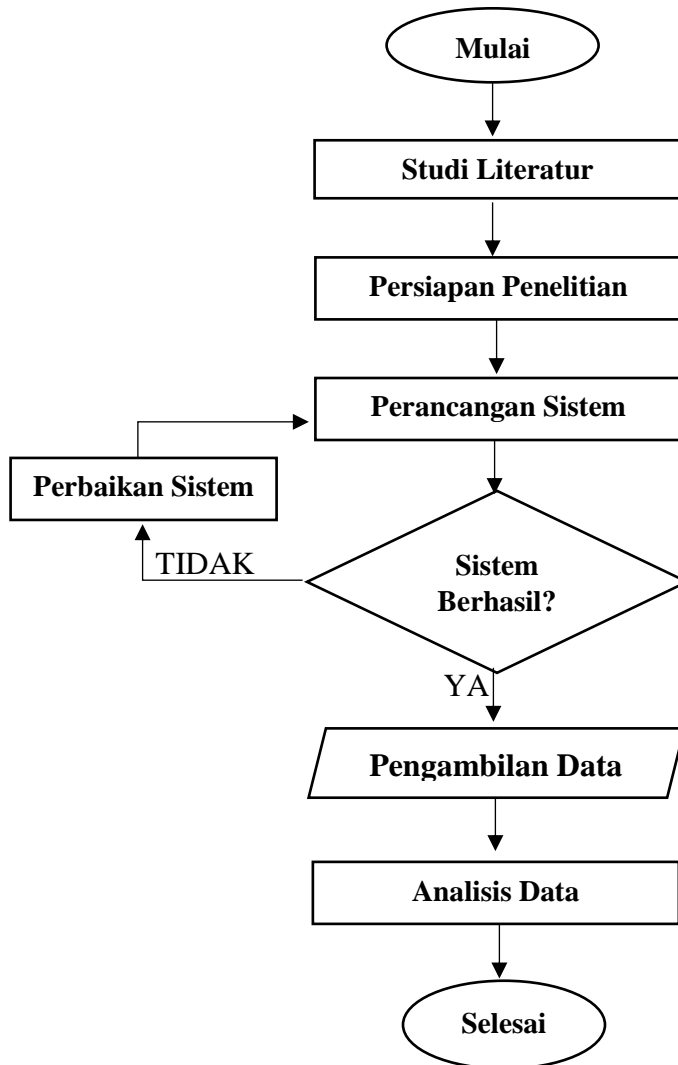


BAB III METODE PENELITIAN

1.1. Diagram Alir Penelitian

Dalam diagram alir penelitian ini merupakan suatu runtutan lajur yang akan dilakukan dalam menyelesaikan sebuah aplikasi *self check-in* yang dapat digambarkan seperti ditunjukkan pada Gambar 3.1. :



Gambar 3.1. Diagram Alir Penelitian

Penjelas diagram alir penelitian diatas adalah :

a. Mulai

Pada proses ini penulis memulai untuk mengerjakan penelitian tentang alat.

b. Studi Literatur

Pada tahap ini penulis melakukan proses mencari informasi mengenai aplikasi yang dibuat. Informasi yang berkaitan dengan pembuatan aplikasi yaitu seperti bahan, komponen, sistem aplikasi. Informasi ini bisa didapatkan dari jurnal, karya tulis ilmiah, buku, *website* pemerintahan dan media online.

c. Persiapan Penelitian

Setelah melakukan studi literature tahap selanjutnya yaitu persiapan penelitian hal ini harus dilakukan sebelum memulai suatu penelitian. Pada penelitian ini mempersiapkan seperangkat computer yang dilengkapi dengan webcam. Hal yang tidak kalah penting adalah memasang software pendukung ke dalam komputer. Software pendukung tersebut yaitu python dan *library OpenCV* serta memasang SQL sebagai tempat penyimpanan data.

d. Perancangan Sistem

Dalam proses ini penulis melakukan perancangan sistem untuk perancangan aplikasi yang akan dibuat yaitu *Self Check-In*. Diharapkan sistem yang dibuat dapat sesuai dengan tujuan penelitian.

e. Pengambilan Data

Setelah dilakukannya proses perancangan sistem dan sistem berhasil maka hal yang dilkaukan selanjutnya adalah pengambilan data.

f. Pengujian Aplikasi

Setelah proses pengambilan data maka proses yang akan dilakukan adalah analisis data. Analisis data ini bertujuan untuk menganalisis

atau mengidentifikasi data yang didapat, sesuai atau tidak dengan tujuan dilakukannya penelitian ini.

g. Analisis Data / Pembuatan Laporan

Berisikan tentang analisis data pembuatan program aplikasi yang telah dilakukan perbaikan sehingga alat sudah dapat diaplikasikan.

1.2. Objek Penelitian

Objek yang akan di teliti dalam penelitian ini yaitu sebuah citra wajah pengguna akun yang digunakan sebagai akses login di bandara pada bagian imigrasi bandara. Objek tersebut berupa citra gambar yang diambil dengan menggunakan camera/webcam dari wajah pengguna pada saat awal mendaftarkan atau membuat passport dan pada saat melakukan login. Dari citra wajah tersebut akan dilakukan proses grayscaling kemudian diproses menggunakan *library OpenCV* sehingga menghasilkan pendeteksian wajah.

1.3. Sumber dan Jenis Data

Data yang akan digunakan pada penelitian ini adalah *capture* wajah dari pengguna akun atau pemilik *passport*. Data didapat melalui proses pendaftaran ataupun pembuatan *passport*. Didalam uji coba alat ini dilakukan dengan menyimpan *capture* wajah pengguna. Data berupa *capture* wajah dari pengguna tersebut akan di simpan ke dalam *database* yang akan dijadikan sebuah acuan ketika sistem akan mencocokkan wajah pada saat pengguna melakukan *login / check-in*.

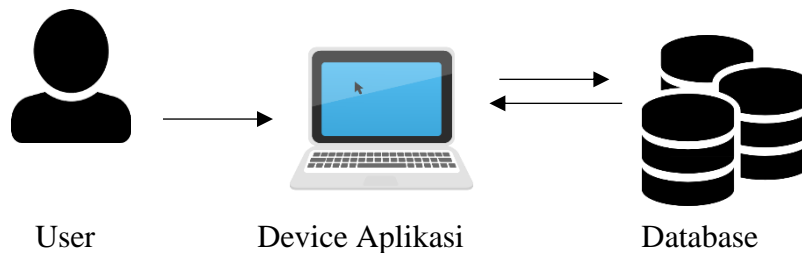
1.4. Metode Analisis

Data yang di dapat yaitu gambar wajah pengguna akan disimpan dalam database sebagai acuan. Dalam hal ini pengaturan posisi, ekspresi wajah dan jarak antara kamera dan wajah harus diatur dengan baik. Hal ini dimaksudkan agar gambar yang di dapat atau dihasilkan lebih baik dengan *noise* yang sedikit.

Sehingga pada saat dilakukan proses pengenalan wajah sistem bisa mengenali wajah pengguna .

Hasil akan ditampilkan oleh sistem dari proses pencocokan wajah apabila sesuai dengan wajah yang telah tersimpan di database atau wajah yang menjadi acuan. Tapi apabila sistem tidak menemukan wajah yang tepat pada database, maka pengguna tidak dapat masuk ke dalam sistem. Nilai jarak kemiripan yang dihasilkan sistem dalam mengenali wajah tergantung pada beberapa factor seperti pose wajah, jarak antar wajah dan kamera, intensitas cahaya, dll, yang dapat mempengaruhi sistem dalam menentukan keputusan pencocokan data.

1.5. Arsitektur Sistem



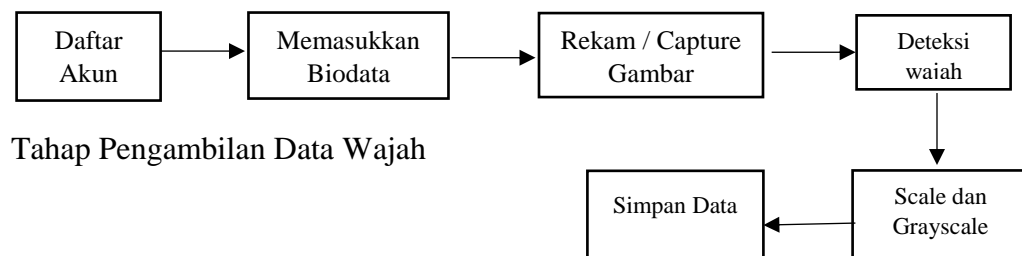
Gambar 3.2. Arsitektur Sistem

Penjelasan tentang gambar :

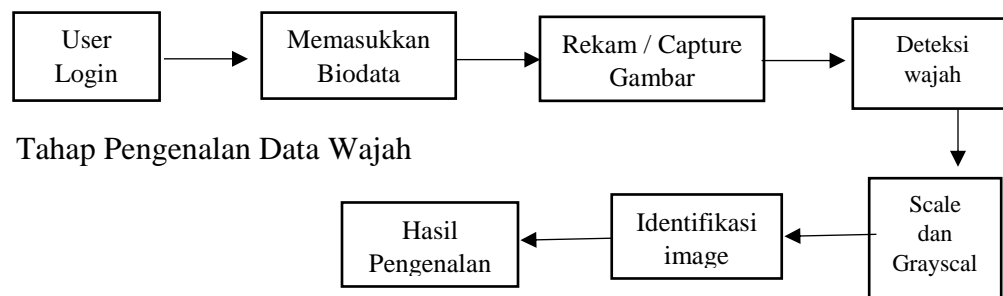
1. Pada gambar diatas *user* (pengguna) akan melakukan *login* pada aplikasi, yang akan memberi perintah pada aplikasi untuk mencocokkan dan menyimpan data pada *database* sehingga dapat melakukan *check-in*.
2. Aplikasi akan mengambil data dari *database*.
3. Aplikasi akan meminta *user* untuk memasukkan beberapa identitas yang akan dicocokkan dengan *database*.
4. Kemudian aplikasi akan melakukan *check-in* dan pencocokan pada database terlebih dahulu. Apabila data telah sesuai dengan yang ada pada *database* maka *login / check-in* berhasil tetapi jika tidak sesuai maka harus mengulang.

1.6. Kerangka konsep

Pada pendeteksian wajah (*face detection*) pengguna pada perangkat lunak ini digunakan sebuah metode yaitu metode *viola-jones*. Serta menerapkan algoritma *Local Binary Pattern Histogram (LBPH)* untuk mengidentifikasi dan pelabelan wajah (*face recognition*). Pada proses pendeteksian ini menggunakan *library* dalam *OpenCV*. Konsep disini juga meminta memasukkan data-data untuk melengkapi keakuratan dalam proses *check-in*. Berikut ini adalah gambaran blok diagram dari sistem *self check-in*:



Gambar 3.3. Tahap Pengambilan pada Pendeteksian Wajah



Gambar 3.4. Tahap Pengenalan Data Wajah

Keterangan:

1. *Daftar Akun*: Ketika pengguna membuat passport pada bagian imigrasi maka akan otomatis terdaftar pada database.
2. *User Login* : Suatu kondisi dimana pengguna akan melakukan proses login untuk check-in di bandara.
3. *Memasukkan Biodata* : Pengguna memasukkan biodata lengkap sesuai dengan persyaratan yang akan dibuat sistem.

4. Rekam / *Capture* Gambar : Pengguna merekam citra wajah untuk dimasukkan ke dalam sistem.
5. *Grayscale* : Gambar hasil capture dari wajah pengguna yang berupa matrik RGB kemudian akan diubah ke dalam matrik grayscale atau keabuan.
6. Simpan Database : Hasil masukan dari pengguna yang berupa biodata, citra wajah yang telah melalui grayscale akan disimpan di database yang telah dibuat.

1.7. Proses Deteksi Wajah

Proses deteksi wajah disini menggunakan algoritma Viola-jones yaitu menggunakan haar cascade. Untuk mendeteksi wajah diperlukan data training setelah itu akan dilakukan pengimplementasian algoritma haar cascade pada program.

1.7.1. Training Data

Pada proses haar cascade ini membutuhkan data training berupa citra positif (berisi wajah) dan citra negatif. Hasil training ini merupakan acuan dari bentuk objek yang akan dideteksi, misalnya mata, hidung, mulut, wajah dan tubuh, yang telah disediakan oleh OpenCV dalam bentuk file *.xml, jadi sudah dapat langsung di impor ke dalam sistem.

1.7.2. Algoritma Haar Cascade

Setelah proses sebelumnya yaitu proses training data telah dilakukan, proses yang akan dilakukan selanjutnya yaitu memanggil algoritma Haar Cascade dalam program. Pada proses ini dilakukan pelacakan wajah pada penelitian ini khususnya untuk pelacakan wajah dengan posisi lurus ke depan kamera (frontalface). Maka berikut ini adalah proses-proses yang akan dilakukan pada metode haar cascade ini.

1. Proses menentukan Haar

Haarcascade merupakan nilai haar feature pada OpenCV. Proses ini dilakukan untuk menempatkan sebuah sampel dari sebuah gambar yang di dapat dari kamera dan akan dibandingkan dengan nilai yang ada pada OpenCV. Kemudian hasil nilai frame akan digunakan untuk mendeteksi apakah objek gambar yang masuk wajah atau bukan wajah.

Dengan mengurangi nilai pixel diwilayah gelap dengan nilai pixel diwilayah terang dapat menentukan keberadaan ada atau tidaknya fitur wajah. Oleh karena itu setiap gambar dirubah ke dalam warna hitam dan putih. Jadi jika nilai dari hasil memiliki perbedaan lebih dari ambang batas selama proses masa pembelajaran citra maka fitur tersebut dapat dikatakan ada.

2. Proses Membuat Cascade Classifier

Untuk mendapatkan hasil deteksi dari haar yang lebih akurat maka harus dilakukan proses Haar yang dibuat banyak, semakin banyak proses Haar yang dilakukan maka hasil yang akan didapat semakin akurat. Akan tetapi, waktu yang akan dibutuhkan untuk memproses gambar juga akan menjadi lebih lama. Pemrosesan tersebut diatur di dalam classifier cascade.

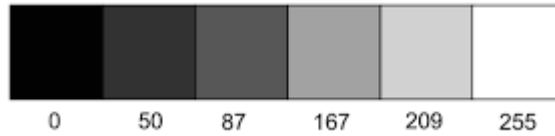
1.7.3. Tahap Scaling

Tahap scaling merupakan proses yang merubah ukuran gambar dalam hal ini memperbesar dan memperkecil gambar, agar semua gambar memiliki ukuran yang sama. Biasanya ukuran pada proses pendeteksian wajah yang efektif menggunakan 20x20.

1.7.4. Tahap Grayscale

Pada tahap ini citra digital yang telah melalui proses scaling / penskalaan kemudian diubah menjadi citra dua warna dengan proses grayscale. Grayscale merupakan warna-warna piksel yang memiliki rentang gradasi warna hitam dan putih. Grayscale dilakukan oleh sistem perangkat lunak. Warna yang

digunakan pada umumnya adalah warna hitam sebagai warna minimal (0), warna putih sebagai warna maksimal (255) dan warna antara kedua warna itu adalah warna abu-abu.

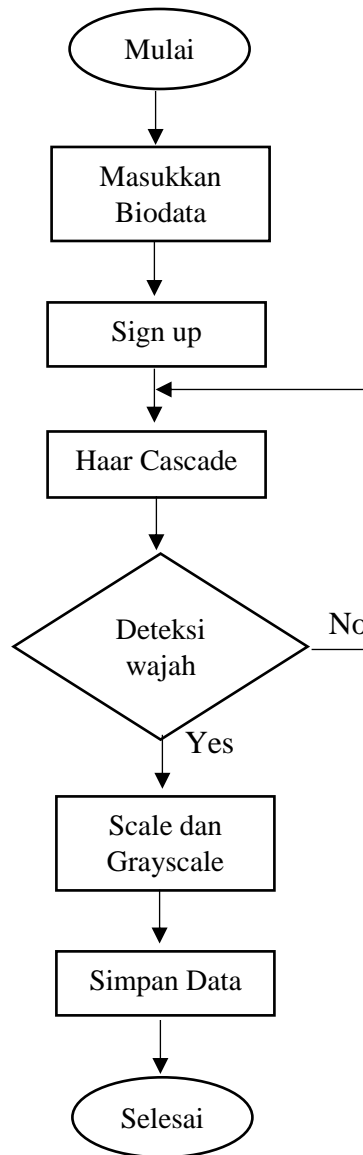


Gambar 3.5 Gradiasi citra skala keabuan (grayscale)

Pada penentuan derajat keabuan sebenarnya memiliki beberapa nilai tidak hanya pada skala 0 – 255 saja. Hal ini tergantung pada berapa nilai kedalaman pixel yang dimiliki. Pada program untuk mengubah kamera menjadi grayscale menggunakan `gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)`.

1.8. Proses Sign Up

Hal yang dilakukan sebelum gambar diberi label dan diidentifikasi adalah menyimpan gambar dan data pemilik passport dalam database. Berikut adalah tahap-tahap proses pengambilan data dan gambar yang ditunjukkan pada Gambar 3.5. berikut ini :



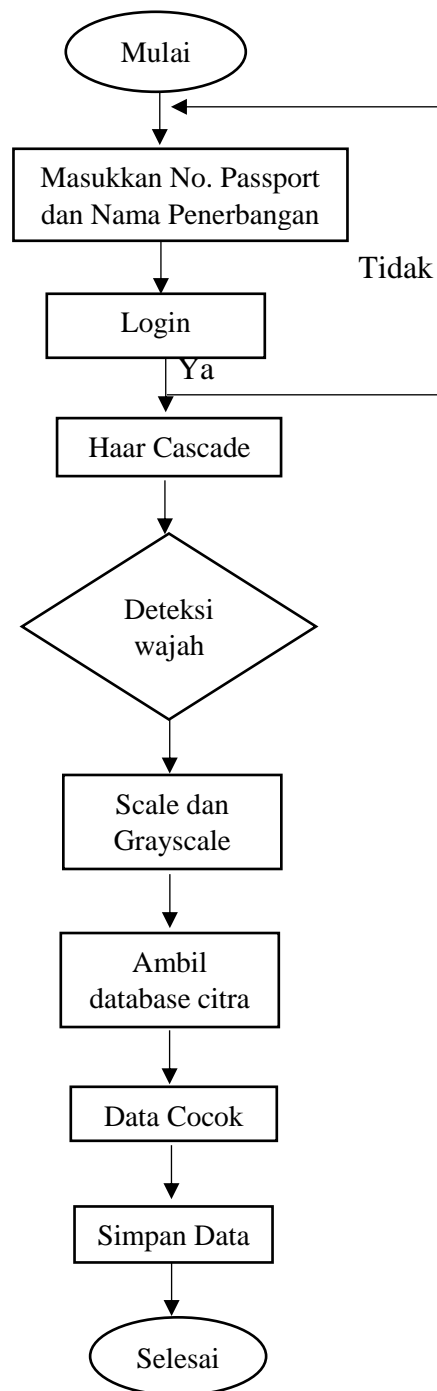
Gambar 3.6. Flowchart Proses Sign-up

Pada flowchat diatas merupakan langkah-langkah dalam melakukan proses pendaftaran passport yang akan langkah awal yang akan digunakan pada proses Self Check-In yang dapat dijelaskan sebagai berikut:

1. Masukkan Biodata : Hal ini dilakukan pada saat pemilik passport melakukan pembuatan passport dan memasukkan data biodata ke dalam sistem sesuai dengan isian yang telah disediakan. Seperti nama, nomor passport, nomor KTP, alamat, dll.
2. *Sign-Up* : ketika proses mengisi data biodata telah selesai langkah selanjutnya yaitu klik tombol *sign-up* yang akan mengaktifkan kamera.
3. *Haar Cascade* : Setelah menekan tombol *sign-up* maka otomatis kamera aktif yang akan mendeteksi wajah, apabila wajah terdeteksi oleh sistem maka sistem melanjutkan dengan melakukan proses *grayscale* (*scale* dan *gray scale*) dan mengcapture gambar wajah 10 kali. Tetapi jika wajah tidak terdeteksi maka kamera tidak mencapture.
4. *Scale* dan *grayscale* : Proses konversi citra yang berupa matrik RGB menjadi citra keabuan.
5. Simpan Database : citra hasil *grayscale* kemudian kemudian disimpan pada database.

1.9. Proses Login

Pada proses ini kamera akan diarahkan ke wajah pengguna, wajah akan dikenali apabila wajah sama dengan gambar yang telah disimpan. Pada pendeteksian menggunakan metode *Viola Jones*. Seperti flowchart yang ditunjukkan pada Gambar 3.6. berikut ini:



Gambar 3.7. Flowchart Proses Login

Pada gambar flowchart 3.7. diatas merupakan flowchart proses login pada aplikasi Self Check-In, yang dapat dijelaskan sebagai berikut:

1. Masukkan nomor passport dan nama penerbangan : Pengguna memasukkan nomor passport dan nama penerbangan untuk melakukan proses login.
2. Login : Setelah selesai mengisi nomor passport dan nama penerbangan maka langkah selanjutnya klik tombol login. Jika data sesuai pada database maka akan masuk langkah selanjutnya yaitu deteksi wajah, akan tetapi apabila tidak sesuai maka akan diulang untuk mengisi data nomor passport.
3. *Haar Cascade* : Setelah menekan tombol login maka otomatis kamera aktif yang akan mendeteksi wajah, apabila wajah terdeteksi oleh sistem maka sistem melanjutkan dengan melakukan proses grayscale (scale dan gray scale).
4. *Grayscale* : Proses konversi citra yang berupa matrik RGB menjadi citra keabuan.
5. Ambil Database Citra Acuan : Mengambil data citra acuan dalam database.
6. *Local Binary Pattern Histogram (LBPH)* : Proses pengenalan wajah menggunakan algoritma *Local Binary Pattern Histogram (LBPH)*.
7. Data Cocok : pada tahap ini koncisi apabila pencocokan citra dianggap cocok atau mirip, maka akan dilanjutkan proses menyimpan data login.

1.10. Rancangan Database

Database pada penelitian ini digunakan sebagai penyimpan data-data pemilik passport sebagai calon penumpang pesawat. Pada database yang dibuat didalamnya berisi biodata pemilik *passport*. Disini ada 2 database yaitu database sign-up dan database login. Pada database sign up tabel yang berisi biodata tersebut seperti : nomor passport, nama, nomor ktp, tanggal lahir, alamat, foto. Database login berisi nomor passport, nama penerbangan dan foto.

1.11. Analisa Kebutuhan Non-Fungsional

Pada Analisis Kebutuhan non-fungsional ini berfungsi untuk menjelaskan hal pendukung sistem yang akan dijalankan. Adapun Kebutuhan yang akan diperlukan untuk mendukung kinerja dari aplikasi ini yaitu:

1) Analisis Kebutuhan Perangkat Lunak

Komputer adalah suatu gabungan dari perangkat lunak dan perangkat keras yang satu sama lainnya saling mendukung pada proses kerjanya. Perangkat lunak akan memerintahkan perangkat keras untuk melakukan tugas-tugas tertentu. Pada Aplikasi Self Check-in ini digunakan sebuah komputer berbasis desktop. Adapun kriteria dari komputer yang digunakan saat pembangunan maupun saat menjalankan aplikasi yaitu:

Tabel 3.1. Data Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1.	Processor	Intel Core i3 2330M 2.20Ghz
2.	Monitor	Asus 14 inci
3.	VGA	AMD Radeon HD 6730M 2GB DDR3 Dedicated
4.	Memory	DDR3 4GB
5.	Kamera	External Webcam Logitech C270

2) Analisis Kebutuhan Perangkat Lunak

Perangkat lunak pada proses ini digunakan untuk memberikan perintah-perintah kepada perangkat keras untuk melakukan perintah-perintah. Berikut ini perangkat lunak yang akan digunakan untuk mendukung perangkat lunak aplikasi self check-in ini:

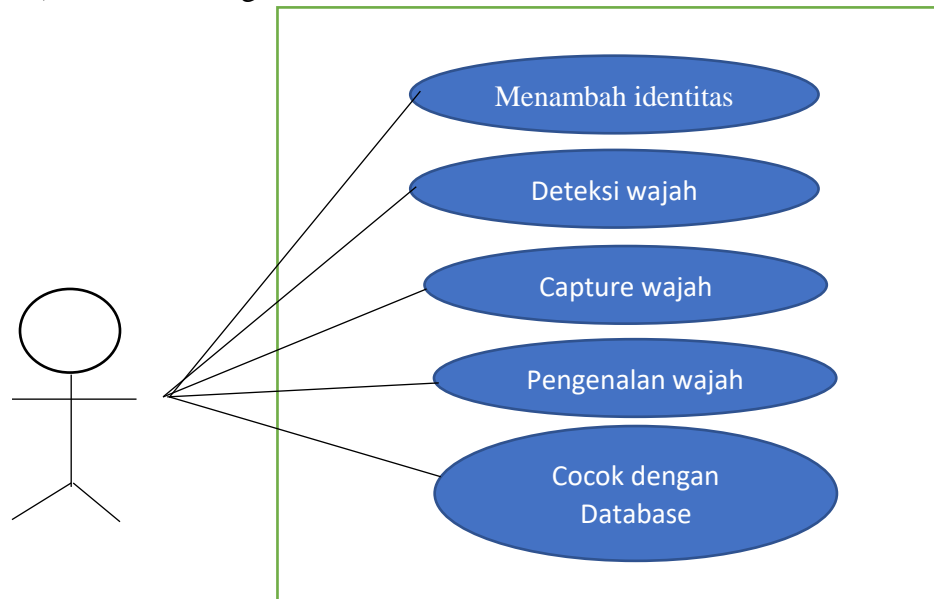
- a. Sistem Operasi : Windows 10 ultimate 64-bit Operating System
- b. Bahasa Pemrograman : Python
- c. Editor/compiler : IDLE Python 2.7
- d. Prototyping : IDLE Python 2.7

e. Perancangan : IDLE Python 2.7 dan PyQt4 (Qt Designer)

1.12. Analisis Kebutuhan Fungsional

Pada analisis kebutuhan fungsional ini merupakan sebuah alur dan pekerjaan dari perangkat lunak yang akan dibuat. Perangkat lunak pendeteksian dan pengenalan wajah ini menggunakan pendeteksian berorientasi pada objek. Di penelitian ini penulis merancang analisis kebutuhan fungsional berupa : use case diagram, use case scenario, dan activity diagram. Berikut ini adalah penjelasan detail mengenai masing-masing perancangan berdasarkan pendekatan berorientasi objek:

1) Use case Diagram



Gambar 3.8. Use Case Diagram

Berdasarkan pada gambar 3.8. diatas terdapat aktor yang merupakan pengguna yang berhubungan langsung dengan perangkat lunak ini. dan setiap use case akan dijelaskan dalam use case scenario.

2) Use case skenario

Skenario proses-proses yang terjadi pada aplikasi, akan dipaparkan dengan tujuan agar lebih memudahkan dalam membangun aplikasi dan lebih konsisten.

A. Use Case Skenario Penambahan Data

Dibawah ini merupakan tabel *use case* skenario proses penambahan data. Pada *use case* penambahan data ini dilakukan agar memudahkan dalam memahami proses skenario pada penambahan data yang telah dipaparkan pada tabel berikut ini:

Tabel 3.2. *Use Case* Skenario Penambahan Data

Nama Use Case	Menambah Identitas calon pengguna
Tujuan	Menambah data akun
Syarat	Telah masuk ke dalam aplikasi
Kondisi sukses	Data pengguna tersimpan
Kondisi gagal	Data pengguna tidak tersimpan
Aktor	Pengguna dan Pihak imigrasi
Alur Utama	<ol style="list-style-type: none"> 1. Mengisi biodata calon pemilik akun, seperti nama, no.ktp, alamat, tanggal lahir, tempat lahir 2. Menekan tombol sign-up 3. Sistem akan menyimpan data dalam databse
Alur alternatif	<ol style="list-style-type: none"> 1. Salah memasukkan identitas pengguna 2. Sistem gagal menyimpn data

B. Use Case Skenario Deteksi Wajah

Dibawah ini merupakan tabel *use case* skenario proses deteksi wajah. Pada *use case* deteksi wajah ini dilakukan agar memudahkan dalam memahami proses skenario pada deteksi wajah yang dipaparkan pada sebuah tabel berikut ini:

Tabel 3.3. *Use Case* Skenario Deteksi Wajah

Nama Use Case	Deteksi Wajah
Tujuan	Mendeteksi wajah
Syarat	Berada didepan kamera
Kondisi sukses	Wajah terdeteksi
Kondisi gagal	Wajah tidak terdeteksi
Aktor	Pengguna (user) dan pihak imigrasi (admin)
Alur Utama	1. Pada saat melakukan sign-up dan selesai melakukan pengisian biodata maka kamera akan aktif 2. Sistem mendeteksi wajah
Alur alternatif	Wajah tidak terdeteksi

C. *Use Case* Skenario *Capture Image*

Di bawah ini merupakan tabel *use case* skenario proses *capture image*. Pada *use case capture image* ini dilakukan agar memudahkan dalam memahami proses skenario pada *capture image* yang telah dipaparkan pada tabel berikut ini:

Tabel 3.4. *Use Case* Skenario *Capture Image*

Nama Use Case	Capture Image
Tujuan	Mengami sample citra dari kamera
Syarat	Berada didepan kamera dan telah mengisi biodata
Kondisi sukses	Citra hasil capture tersimpan
Kondisi gagal	Citra hasil capture tidak tersimpan
Aktor	Pengguna (user) dan pihak imigrasi (admin)

Alur Utama	<ol style="list-style-type: none"> 1. Melakukan capture wajah 2. Sistem mendeteksi wajah 3. Sistem mencapture dan menyimpan hasil capture
Alur alternatif	Sistem tidak dapat menyimpan hasil capture

D. Use Case Skenario Pengenalan Wajah

Di bawah ini merupakan tabel *use case* skenario proses pengenalan wajah. Pada *use case* pengenalan wajah ini dilakukan agar memudahkan dalam memahami proses skenario pada pengenalan wajah yang telah dipaparkan pada tabel berikut ini:

Tabel 3.5. Use Case Skenario Pengenalan Wajah

Nama Use Case	Pengenalan Wajah
Tujuan	Mengenalai wajah yang telah terdeteksi
Syarat	Berada didepan kamera dan telah mengisi data untuk login
Kondisi sukses	Berhasil melakukan pengenalan wajah atau wajah dikenali
Kondisi gagal	Tidak berhasil melakukan pengenalan wajah atau wajah tidak dikenali
Aktor	Pengguna (User)
Alur Utama	<ol style="list-style-type: none"> 1. User mengisi biodata sebelum login 2. User menekan tombol login 3. Sistem mendeteksi wajah 4. Sistem mencocokkan wajah pengguna dengan database pengguna (user) 5. Sistem menampilkan data pengguna sesuai yang ada didatabase.

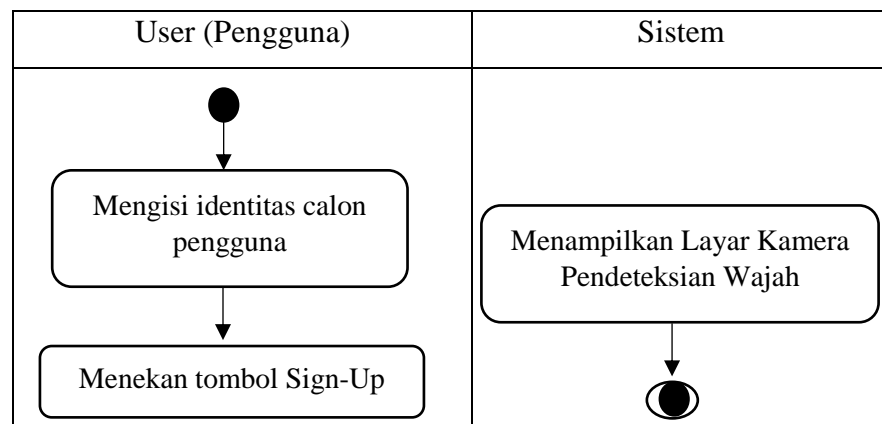
Alur alternatif	<ol style="list-style-type: none"> 1. Sistem tidak dapat menemukan data wajah pengguna 2. Sistem tidak menampilkan data kucing
-----------------	--

3. Activity Diagram

Activity diagram merupakan cara untuk menjelaskan aktifitas use case yang digunakan *tool* yang bernama *activity diagram*. Berikut ini merupakan gambaran aktifitas pada setiap use case:

A. Activity Diagram Menambahkan Identitas Calon Pengguna

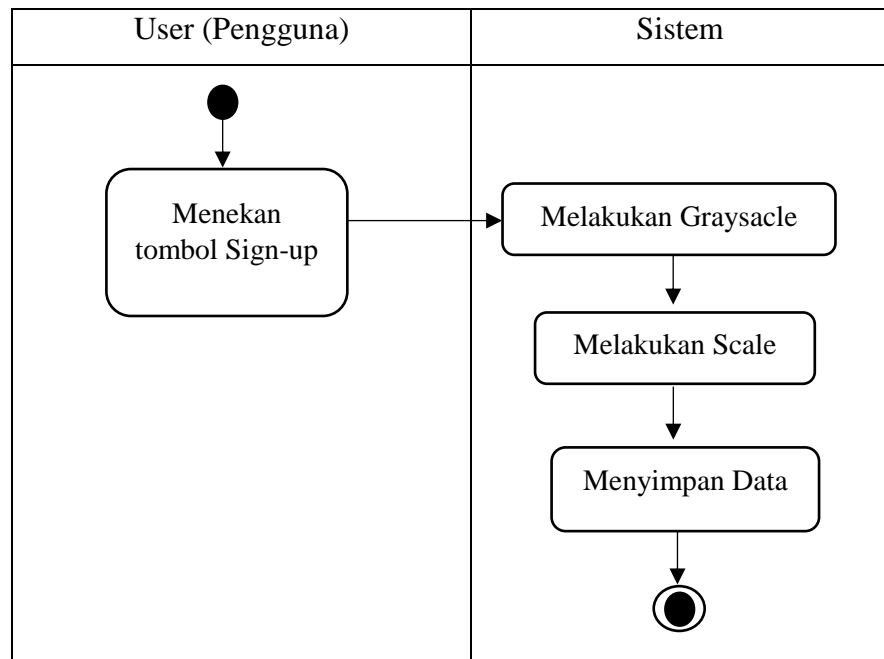
Di bawah ini merupakan diagram untuk menambahkan identitas calon pengguna. Pada proses ini hal yang pertama dilakukan adalah mengisi identitas calon penumpang, kemudian menekan tombol *sign-up* yang akan menampilkan layer kamera dan melakukan pendeteksian wajah.



Gambar 3.9. Activity Diagram Menambahkan Identitas Calon Pengguna

B. Activity Diagram Deteksi Wajah

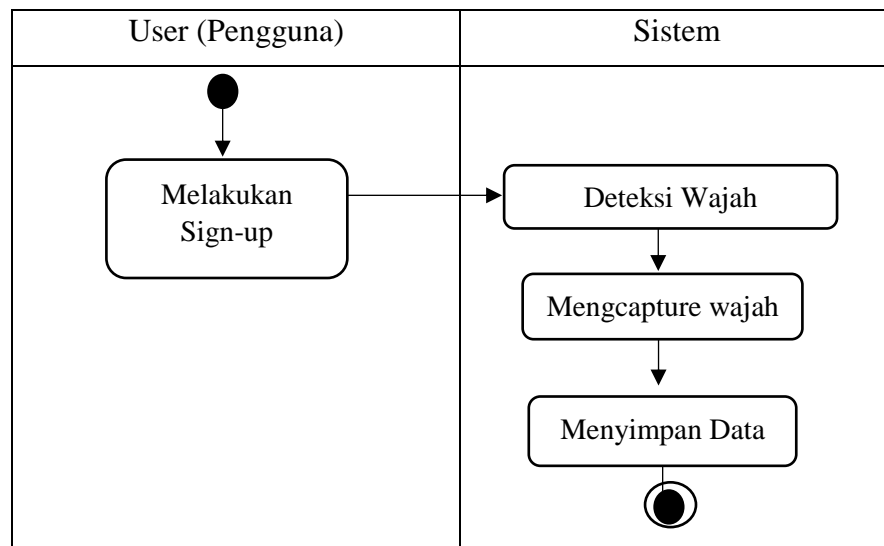
Di bawah ini merupakan proses *activity diagram* pada proses deteksi wajah. Yang mana pada proses ini ketika kita menekan tombol *sign-up* akan terjadi proses deteksi wajah yang kemudian di proses dengan *grayscale* dan dilakukan proses *scale*, sehingga gambar yang tersimpan dalam warna keabuan.



Gambar 3.10. Activity Deteksi Wajah

C. Activity Diagram Capture Image

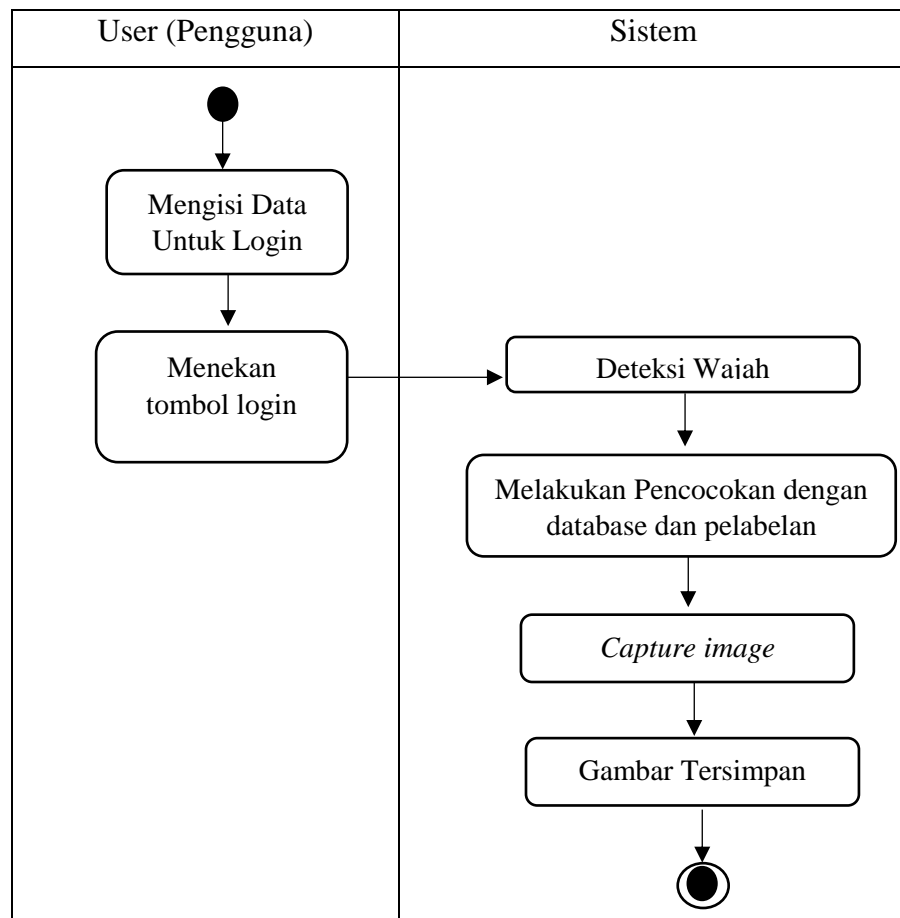
Di bawah ini merupakan proses *activity* diagram pada proses *capture image*. Pada proses ini setelah melakukan sign-up maka kamera aktif dan akan mendeteksi wajah. Setelah wajah terdeteksi maka kamera akan *capture* gambar yang kemudian di simpan pada sebuah folder yang telah ditentukan.



Gambar 3.11. Activity Diagram Capture Image

D. Activity Diagram Pengenalan Wajah

Di bawah ini merupakan proses *activity* diagram pada proses pengenalan wajah. Hal yang pertama yang harus dilakukan adalah mengisi data untuk *login* kemudian menekan tombol *login* yang akan mengaktifkan kamera dan melakukan pendeteksian wajah. Wajah akan dicocokkan dengan database dan dilakukan pelabelan kemudian *capture* dan gambar tersimpan pada folder yang telah ditentukan.



Gambar 3.12. Activity Diagram Pengenalan Wajah

