BAB III

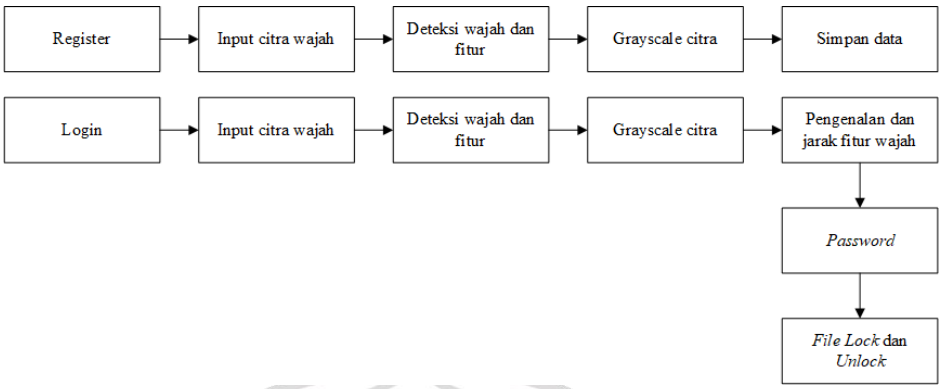
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Sistem keamanan file merupakan sistem keamanan yang mengamankan sebuah file atau data penting bersifat pribadi yang terdapat pada ruang penyimpanan komputer. Pengamanan yang digunakan saat ini masih menggunakan kata sandi atau password, yang dimana penggunaan kata sandi atau *password* tidak aman dan dapat dirusak (*hack*) oleh orang yang tidak bertanggung jawab. Sehingga diperlukan pengamanan yang lebih aman yaitu mengamankan dengan menggunakan pengenalan wajah.

Data diperoleh berupa citra wajah yang didapatkan langsung dari seseorang. Data yang dikumpulkan berupa citra wajah manusia dengan resolusi citra asli yang kemudian di *resize* menjadi 290 x 290 piksel. Karakteristik dari citra yang digunakan adalah citra berwarna dengan posisi wajah menghadap ke depan *webcam* dengan warna kulit wajah berbeda-beda dan pencahayaan yang merata. Data untuk deteksi wajah diperoleh dari metode *triangle face* dengan menentukan titik koordinat fitur wajah atau posisi berdasarkan mata, hidung, mulut dengan *haar cascade classifier*. Sedangkan untuk pengenalan wajah diperoleh dengan *eigenface* dan hasilnya akan digunakan pada perhitungan jarak *minkowski distance* yang didapatkan dari piksel citra. Dengan metode ini akan dihasilkan jarak antar fitur wajah yang nantinya dapat digunakan untuk proses pengenalan wajah.

Tahap pertama pada perancangan ini merupakan blok diagram dari keseluruhan sistem yang dirancang. Blok diagram merupakan salah satu bagian terpenting dalam perencanaan suatu alat, dari blok diagram inilah dapat diketahui cara kerja dari rangkaian keseluruhan yang digunakan. Sehingga keseluruhan blok diagram rangkaian tersebut akan menghasilkan suatu sistem yang dapat difungsikan atau dapat bekerja sesuai dengan perancangan. Berikut adalah blok diagram keseluruhan sistem.



Gambar 3.1 Blok Diagram Sistem

Keterangan :

1. Register

   Pengguna mendaftar membuat akun data diri beserta wajah yang terdiri dari *input* nama, *input username, input password*, dan *capture* wajah.

2. Login

   Pada tahap ini pengguna terlebih dahulu akan melakukan proses *login* menggunakan wajah dan berikutknya *login* menggunakan kata sandi (*password)*.

3. Pengambilan Citra Wajah (*Webcam*)

   Mempersiapkan pengguna yang akan diambil citra wajahnya untuk disimpan ke dalam sistem.

4. Grayscale Citra

   Pada tahap ini citra pengguna berupa *rgb*, diubah menjadi gray atau abu-abu.

5. Deteksi Wajah dan Fitur

   Pada tahap ini wajah setiap pengguna akan dideteksi menggunakan *haar cascade classifier* untuk deteksi wajah, jika wajah pengguna tidak terdeteksi maka tidak akan bisa mencapture wajah dan disimpan ke database.

6. Pengenalan dan Jarak Fitur Wajah

   Proses ini untuk pencocokan antar fitur wajah dari citra masukan dan citra acuan dengan menentukan nilai koordinat atau posisi dari fitur wajah secara *real time* yaitu mata, hidung, mulut (*triangle face*) dengan *haar cascade classifier* dan akan dilakukan perhitungan menggunakan jarak *minkowski distance*.

7. Password

   Pada proses ini pengguna diminta untuk *login* menggunakan kata sandi atau *password,* dimana penggunaan *login* menggunakan *password* ini sebagai keamanan berlapis agar keamanan lebih terjamin.

8. File Lock dan Unlock

   Pengguna menyiapkan *file* yang akan diamankan atau tidak diamankan.

3.2 Hasil Analisis

Berdasarkan hasil analisis dari penelitian yang dilakukan adalah menghasilkan keamanan sebuah file atau data penting yang bersifat pribadi dan diharapkan mampu dalam mengamankan sebuah file dari orang yang tidak bertanggung jawab dengan menggunakan pengenalan wajah.

Secara umum sistem yang akan dibuat dalam penilitian ini adalah sebagai berikut :

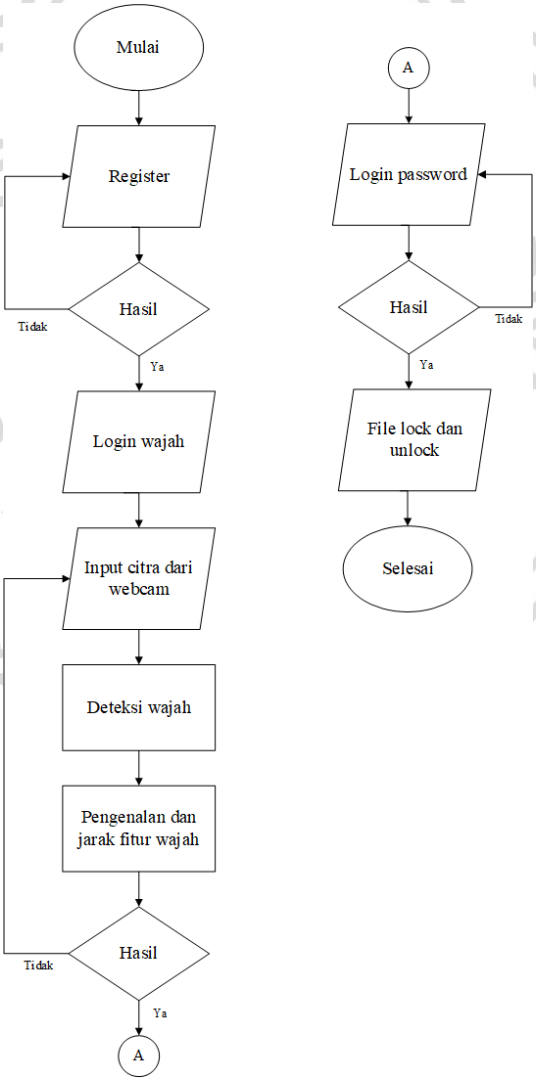
1. Pengguna mendaftar dan memasukan data data diri beserta citra wajah untuk disimpan ke database dan login kedalam sistem.
2. Pengguna melakukan pecocokan saat login kedalam sistem menggunakan pengenalan wajah, yang dimana citra wajah pengguna sebelumnya telah disimpan di database.
3. Sistem akan memberikan akses *lock* & *unlock file* sesuai keinginan pengguna.

Sistem keamanan file menggunakan metode Triangle Face.merupakan salah satu metode metode yang dapat digunakan untuk mengenali wajah seseorang pada suatu citra digital. Metode ini dapat mengenali seseorang dengan mendeteksi fitur-fitur wajah yang terdapat pada citra masukan. Fitur-

fitur wajah yang dimaksud antara lain seperti mata, hidung, mulut, serta lebar dan tinggi wajah.

3.2.1 Flowchart Sistem

Flowchart adalah bentuk alir dari diagram blok yang merupakan salah satu bagian penting dalam perancangan suatu sistem. Cara kerja keseluruhan alat yang akan dibuat dapat dilihat pada *flowchart* yang akan menghasilkan suatu sistem yang dapat difungsikan atau dapat dijalankan.

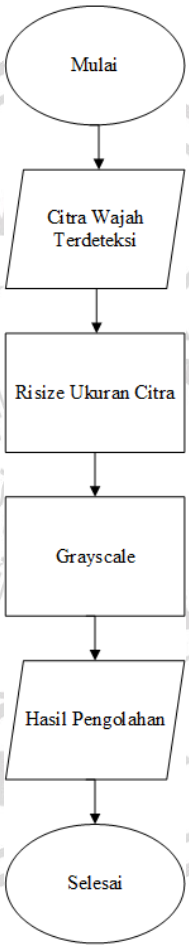


Gambar 3.2 Flowchart Keseluruhan Sistem

3.2.2 Pengolahan Citra Terdeteksi

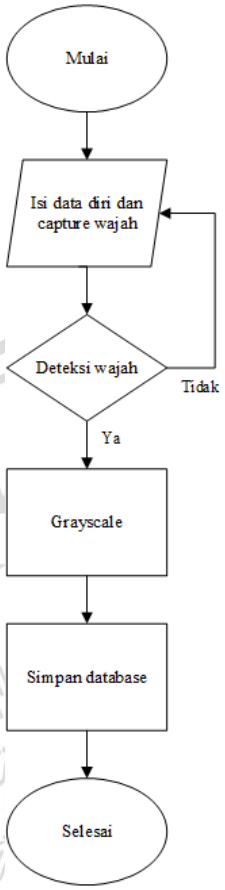
Proses pengolahan wajah dapat dilihat pada gambar 3.3 berikut :

1. Inputan berupa citra wajah yang telah terdeteksi di lakukan proses resize atau penyamaan ukuran menjadi 290 x 290.
2. Hasil dari *resize* ini kemudian diproses untuk didapatkan nilai keabuannya (*Grayscale*).



Gambar 3.3 Proses Pengolahan Citra Terdeteksi

3.2.3 Proses Peyimpanan Data *Tamplate*



Gambar 3.4 Proses Penyimpanan Data *Template*

Keterangan :

1. Isi data dan *capture* wajah

   Pengguna mendaftar akun dengan mengisi data diri ke dalam sistem sesuai yang telah disediakan dan pengguna juga memasukkan data berupa citra wajah sebagai citra acuan.

2. Deteksi wajah

   Jika wajah telah terdeteksi oleh sistem, maka sistem akan melanjutkan ke tahap proses *grayscale* citra dan jika wajah tidak terdeteksi maka akan kembali ke proses awal.

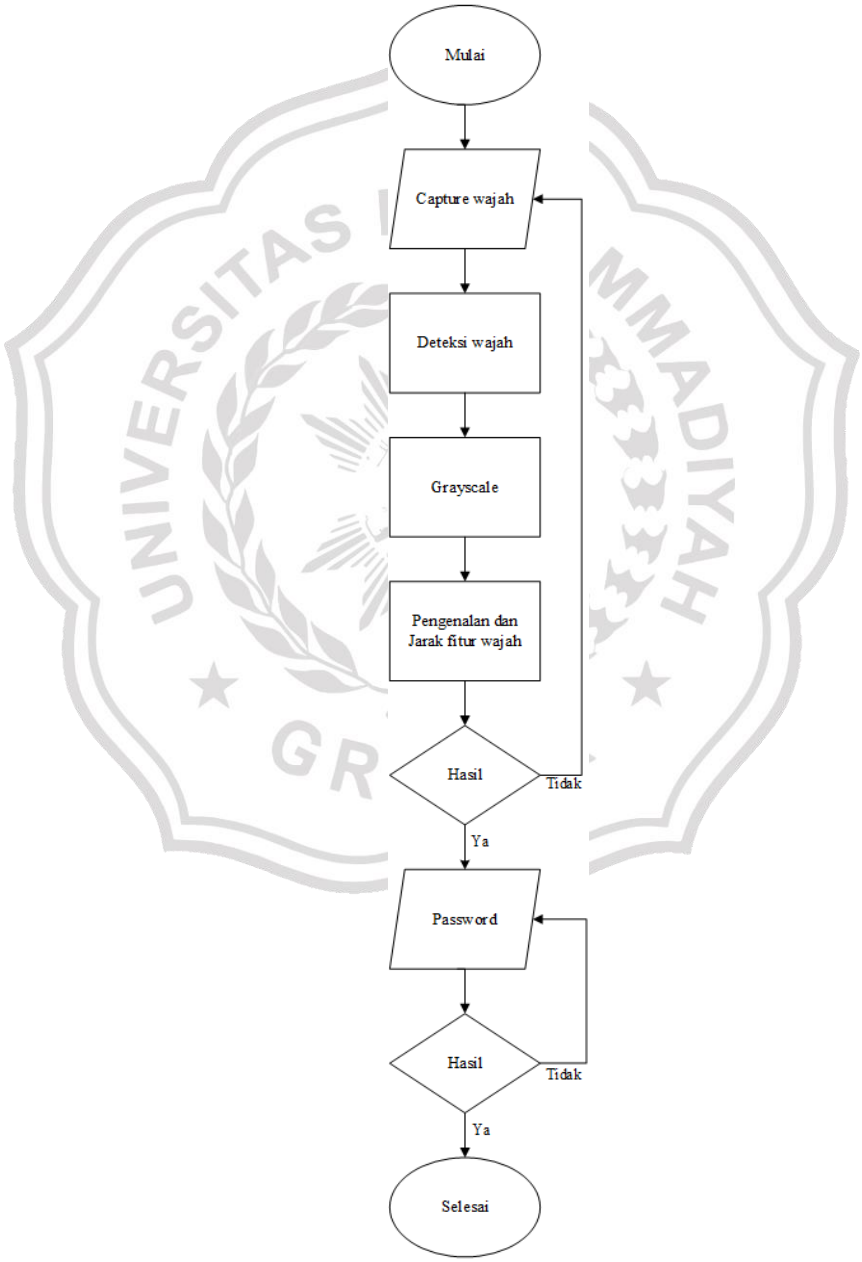
3. Grayscale

   Proses konversi citra dari *rgb* menjadi *gray* atau keabuan.

4. Simpan database

   Citra hasil grayscale akan disimpan ke dalam database.

3.2.4 Proses Pencocokan Data *Tamplate*



Gambar 3.5 Proses Pencocokan Data *Template*

Keterangan :

1. Capture wajah

   Proses ini adalah proses dimana pengguna login menggunakan wajah. Pada tahap ini wajah pengguna akan dipersiapkan untuk pengambilan citra wajah masing-masing yang meliputi posisi menghadap lurus ke arah webcam dan tidak boleh miring atau membelakangi webcam.

2. Deteksi wajah

   Pada tahap ini wajah pengguna harus terdeteksi jika tidak wajah pengguna tidak akan bisa dicapture dan disimpan ke database. Untuk deteksi wajah ini menggunakan *haar cascade classifier*.

3. Grayscale

   Pada tahap ini proses konversi citra dari *rgb* menjadi *gray* atau keabuan.

4. Pengenalan wajah

   Pada tahap ini adalah proses pencocokan antar fitur wajah pengguna pada citra masukan dengan citra acuan, untuk pengenalan wajah menggunakan *eigenface*. Untuk menentukan nilai koordinat atau posisi dari fitur wajah yaitu mata, hidung, mulut (*triangle face*) dengan *haar cascade classifier* yang diambil secara *real time* dan akan dilakukan perhitungan menggunakan jarak *minkowski distance*.

5. Password

   Pada proses ini pengguna diminta untuk login menggunakan kata sandi atau *password,* dimana penggunaan *login* menggunakan *password* ini sebagai keamanan berlapis agar keamanan lebih terjamin.

3.3 Representasi Model

Pada tahap representasi model ini penulis menggunakan data yang di dapatkan langsung dari sistem yang dibuat. Data yang akan digunakan adalah data citra wajah seseorang dengan berformat *JPG*.

3.3.1 Register

Pada tahap ini pengguna diminta registrasi identitas data diri, yang terdiri dari :

1. *Input* Nama
2. *Input Username*
3. *Input Password*
4. *Capture* Wajah

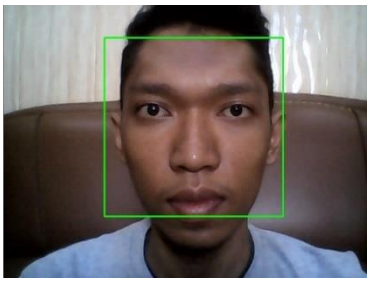
3.3.2 Tahap Pengambilan Citra

Pada tahap ini citra masukan berupa citra *rgb* yang diperoleh dari webcam. Citra yang digunakan merupakan citra wajah seseorang. Citra *RGB* (*Red, Green, Blue*) merupakan citra yang nilai intensitas pikselnya tersusun oleh tiga kanal warna yaitu merah, hijau, dan biru. Untuk citra masukan dapat dilihat pada gambar dibawah ini.

Gambar 3.6 Citra Masukan

3.3.3 Deteksi Wajah

Pada tahap ini deteksi wajah pengguna menggunakan *Haar Cascade Classifier*. *Haar Cascade Classifier* atau *Haar-like features* merupakan *rectangular features* (fungsi persegi), yang memberikan indikasi secara spesifik pada sebuah citra. Fungsi *Haar-like features* adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image obyek tersebut. Tujuan dari penggunaan *Haar Cascade Classifier* penelitian ini adalah untuk mendeteksi objek berupa wajah. Untuk citra deteksi wajah dapat dilihat pada gambar dibawah ini.



Gambar 3.7 Citra Deteksi Wajah

Sedangkan pengenalan wajah pengguna menggunakan *eigenface*. *eigenface* adalah sebuah algoritma pengenalan wajah (*face detection*). Dalam menghasilkan *eigenface*, sekumpulan citra wajah manusia diambil pada kondisi pencahayaan yang sama kemudian dinormalisasikan dan diproses pada resolusi yang sama (misal m x n), kemudian citra tadi diperlakukan sebagai vektor dimensi m x n dimana komponennya diambil dari nilai piksel citra. Tujuan dari penggunaan *eigenface* penelitian ini adalah untuk mengetahui identitas wajah pengguna yang terdeteksi.

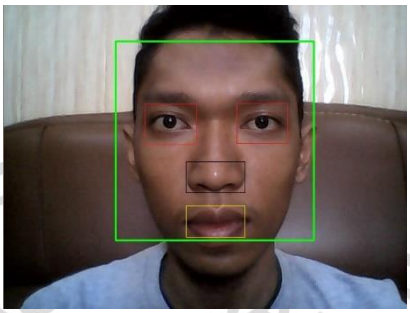
3.3.4 Prepocessing

Pada tahap ini dilakukan untuk mengubah citra dari citra berwarna menjadi keabuan (*rgb to grayscale*) yang berguna untuk mempermudah dan mempercepat pengenalan pola wajah. Citra *RGB* (*Red, Green, Blue*) merupakan citra yang nilai intensitas pikselnya tersusun oleh tiga kanal warna yaitu merah, hijau, dan biru. Sedangkan citra *grayscale* adalah citra yang nilai intensitas pikselnya berdasarkan derajat keabuan.

Gambar 3.8 Citra *Grayscale*

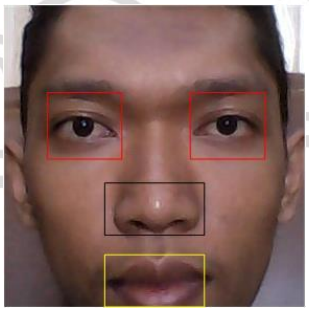
3.3.5 Menghitung Koordinat Fitur Wajah

Pada tahap ini pencarian koordinat posisi fitur wajah yaitu mata, hidung, dan mulut (*triangle face*) dengan menggunakan *haar cascade classifier*. Untuk proses *haar cascade classifier* bisa dilihat pada gambar dibawah ini :

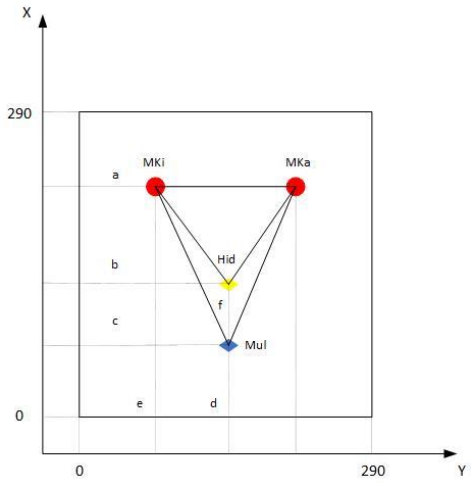


Gambar 3.9 Citra *Haar Cascade Classifier*

Setelah didapatkan fitur – fitur wajah, maka langkah selanjutnya adalah proses perhitungan jarak antar fitur wajah akan digunakan *minkowski distance* yang meliputi *euclidean distance*. Pada perhitungan ini nilai citra diambil secara *real time* dengan *haar cascade*. Untuk proses *minkowski distance* bisa dilihat pada gambar di bawah ini.



Gambar 3.10 Citra *Real Time* dengan *Haar Cascade*



Gambar 3.11 Jarak Antar Fitur Wajah

Jarak fitur wajah yang dicari antara lain :

1. Jarak mata kiri – mata kanan (MKi – MKa)
2. Jarak mata kanan – mulut (MKa – Mul)
3. Jarak mata kiri – mulut (MKi – Mul)
4. Jarak mata kanan – hidung (MKa – Hid)
5. Jarak mata kiri – hidung (MKi – Hid)

Keterangan :

MKa = Mata Kanan

MKi = Mata Kiri

Mul = Mulut

Hid = Hidung

Koordinat titik fitur wajah :

MKa (215,113)

MKi (77,115)

Mul (148,269)

Hid (148,192)

1. Menghitung antar fitur wajah

   $d(x,y) = \sqrt{|dx^2 + dy^2|}$
   dimana,
   dx $= x_2 - x_1$
   dy $= y_2 - y_1$
2. Menghitung jarak antara MKa-MKi

   MKa (215,113)

   MKi (77,115)

   MKa-Ki $= \sqrt{|218 - 77|^2 + |113 - 115|^2}$

   $= \sqrt{|138|^2 + |-2|^2}$

   $= \sqrt{19{,}044 + 4}$

   $= \sqrt{19{,}048}$

   $= 138$

   Berdasarkan perhitungan rumus menggunakan *minkowski distance* yang meliputi *euclidean distance* antara jarak matak kanan – mata kiri menghasilkan nilai 138 perpiksel. Untuk lebih jelasnya bisa dilihat pada tabel 3.1.

   Tabel 3.1 Perhitungan jarak Mka-Mki

√|218-80|^2 + |113-113|^2	
138	-2
19,044	4
19,048	
138	

   Selanjutnya untuk perhitungan mata kanan – mulut sebagai berikut.

3. Menghitung jarak antara MKa-Mul

   MKa (215,113)

   Mul (148,269)

   MKa-Mul $= \sqrt{|215-148|^2 + |113-269|^2}$

   $= \sqrt{|67|^2 + |-156|^2}$

   $= \sqrt{4{,}489 + 24{,}336}$

   $= \sqrt{28{,}825}$

   = 169.779

   Berdasarkan perhitungan rumus menggunakan *minkowski distance* yang meliputi *euclidean distance* antara e jarak matak kanan – mulut menghasilkan nilai 169,779 perpiksel. Untuk lebih jelasnya bisa dilihat pada tabel 3.2.

   Tabel 3.2 Perhitungan jarak Mka-Mul

√|215-148|^2 + |113-269|^2	
67	-156
4,489	24,336
28,825	
169.779	

   Selanjutnya untuk perhitungan mata kiri – mulut sebagai berikut

4. Menghitung jarak antara MKi-Mul

   MKi (77,115)

   Mul (148,269)

   MKi-Mul $= \sqrt{|77-148|^2 + |115-269|^2}$

   $= \sqrt{|-71|^2 + |-156|^2}$

   $= \sqrt{5{,}041 + 24{,}336}$

   $= \sqrt{29{,}377}$

   = 171.397

   Berdasarkan perhitungan rumus menggunakan *minkowski distance* yang meliputi *euclidean distance* antara jarak matak kanan – mulut

menghasilkan nilai 171,397 perpiksel. Untuk lebih jelasnya bisa dilihat pada tabel 3.3.

Tabel 3.3 Perhitungan jarak Mki-Mul

√|77-148|^2 + |115-269|^2	
-71	-156
5,041	24,336
29,377	
171.397	

Selanjutnya untuk perhitungan mata kanan – hidung sebagai berikut

5. Menghitung jarak antara MKa-Hid

   MKa (215,113)

   Hid (148,192)

   MKa-Hid $= \sqrt{|215-148|^2 + |113-192|^2}$

   $= \sqrt{|67|^2 + |-79|^2}$

   $= \sqrt{4{,}489 + 6{,}241}$

   $= \sqrt{10{,}730}$

   $= 104$

   Berdasarkan perhitungan rumus menggunakan *minkowski distance* yang meliputi *euclidean distance* antara jarak matak kanan – mulut menghasilkan nilai 104 perpiksel. Untuk lebih jelasnya bisa dilihat pada tabel 3.4.

Tabel 3.4 Perhitungan jarak Mka-Hid

√|215-148|^2 + |113-192|^2	
67	-79
4,489	6,241
10,730	
104	

Selanjutnya untuk perhitungan mata kanan – hidung sebagai berikut

6. Menghitung jarak antara MKi-Hid

   MKi (77,115)

   Hid (148,192)

   MKi-Hid $= \sqrt{|77-148|^2 + |115-192|^2}$

   $= \sqrt{|-71|^2 + |-77|^2}$

   $= \sqrt{5{,}041 + 5{,}929}$

   $= \sqrt{10{,}970}$

   = 104.737

Berdasarkan perhitungan rumus menggunakan minkowski distance jarak matak kanan – mulut menghasilkan nilai 104,737 perpiksel. Untuk lebih jelasnya bisa dilihat pada tabel 3.5.

Tabel 3.5 Perhitungan jarak Mki-Hid

√|77-148|^2 + |115-192|^2	
-71	-77
5,041	5,929
10,970	
104.737	

3.3.6 Pencocokan Citra

Pada tahap ini akan dilakukan pencocokan antar fitu wajah yaitu mata, hidung dan mulut (*triangle face*) pada tiap citra. Sistem memliki pengguna dengan username “Anul”, “Ilham” dan “Tri”. Masing-masing pengguna telah memasukkan citra acuan pada sistem.

Citra acuan yang ada pada *database* :

Gambar 3.12 Citra (A)

Gambar 3.13 Citra (B)

Gambar 3.14 Citra (C)

Masing-masing citra mempuyai nilai jarak antar fitur wajah yang berbeda.

Citra A memiliki nilai jarak antar fitur wajah yaitu :

1. Mata kanan-mata kiri : 122
2. Mata kanan-hidung : 88
3. Mata kanan-mulut : 139
4. Mata kiri-hidung : 88
5. Mata kiri-mulut : 138

Citra B memiliki nilai jarak antar fitur wajah yaitu :

1. Mata kanan-mata kiri : 113
2. Mata kanan-hidung : 76
3. Mata kanan-mulut : 115
4. Mata kiri-hidung : 72
5. Mata kiri-mulut : 111

Citra C memiliki nilai jarak antar fitur wajah yaitu :

1. Mata kanan-mata kiri : 102
2. Mata kanan-hidung : 79
3. Mata kanan-mulut : 122
4. Mata kiri-hidung : 74
5. Mata kiri-mulut : 117

Setiap citra disimpan pada direktori sesuai dengan nama saat pendaftaran akun pengguna. Citra A disimpan dalam direktori dengan nama “Anul”, Citra B dengan nama “Ilham” dan Citra C disimpan dengan nama “Tri”. Kemudian salah satu pengguna ingin *login* ke dalam sistem dengan memasukkan *password* dan citra masukan ke dalam sistem.

Citra input dengan *username* “Anul” :

Gambar 3.15 Citra (D)

Citra D mempuyai nilai jarak antar fitur wajah yaitu :

1. Mata kanan-mata kiri : 138
2. Mata kanan-hidung : 104
3. Mata kanan-mulut : 105
4. Mata kiri-hidung : 170
5. Mata kiri-mulut : 169

Kemudian sistem akan menghitung nilai jarak kemiripan antar fitur wajah citra input dan citra acuan yang ada pada database. Sehingga akan disimpulkan citra acuan mana yang lebih mirip citra dengan citra input. Berikut perhitungan untuk menghitung kemiripan dengan *minkowski distance* :

$$\text{Citra (A,D)} = \sqrt[3]{\begin{matrix}|122-138|^3+|88-104|^3+|139-105|^3+\\ |88-170|^3+|138-169|^3\end{matrix}}$$

$$= \sqrt[3]{\begin{matrix}|-16|^3+|-16|^3+|34|^3+|-82|^3+\\ |-31|^3\end{matrix}}$$

$$= \sqrt[3]{4{,}096+4{,}096+39{,}753+546{,}922+29{,}791}$$

$$= \sqrt[3]{624{,}658}$$

$$= 85{,}483$$

$$\text{Citra (B,D)} = \sqrt[3]{\begin{matrix}|113-138|^3+|76-104|^3+|115-105|^3+\\ |72-170|^3+|111-169|^3\end{matrix}}$$

$$= \sqrt[3]{\begin{matrix}|-25|^3+|-28|^3+|10|^3+|-98|^3+\\ |-58.000|^3\end{matrix}}$$

$$= \sqrt[3]{15{,}625+21{,}952+1{,}032+934{,}839+195{,}112}$$

$$= \sqrt[3]{1{,}168{,}559}$$

$$= 105{,}329$$

$$\text{Citra (C,D)} = \sqrt[3]{\begin{matrix}|102-138|^3+|79-104|^3+|122-105|^3+\\ |74-170|^3+|117-169|^3\end{matrix}}$$

$$= \sqrt[3]{\begin{matrix}|-36|^3+|-25|^3+|17|^3+|-96|^3+\\ |-52.000|^3\end{matrix}}$$

$$= \sqrt[3]{46{,}656+15{,}625+4{,}610+878{,}640+140{,}608}$$

$$= \sqrt[3]{1{,}086{,}139}$$

$$= 102{,}793$$

Dari hasil perhitungan diatas, maka dapat disimpulkan citra acuan yang paling mirip dengan citra input adalah citra A. Sistem kemudian mencocokkan kesamaan dengan *username* dan *password* yang pengguna masukkan saat login. Citra A berada pada direktori "Anul" dengan *username* dan *password* yang dimasukkan adalah "Anul", sehingga keduanya cocok. Kemudian akan dilihat nilai kemiripan antar fitur wajahnya. Nilai kemiripan antar fitur wajah antara citra A dan citra D atau (A,D) adalah 85,483 yaitu lebih kecil dari nilai ambang citra yang laiinya.

3.3.7 *Login*

Pada tahap ini pengguna diminta *login* dengan *password* yang bertujuan sebagai keamanan berlapis, data masukan yang dibutuhkan *input password.*

3.3.8 *File Lock* dan *Unlock*

Pada tahap ini pengguna dapat mengakses *file* dan mengamankan *file* sesuai yang diinginkan.

3.4 Perancangan Sistem

Perancangan adalah penggambaran perencanaan sistem agar lebih terstruktur dan memudahkan dalam implementasi sistem. Perancangan sistem ini menggunakan *use case* diagram sistem yang menggambarkan setiap proses yang ada pada sistem. *use case* diagram sistem terdiri dari gambaran input, proses, dan output dari sistem yang dirancang. pada *use case* diagram sistem pada bab ini bertujuan untuk menguraikan gambaran umum dari sistem yang akan dibangun.

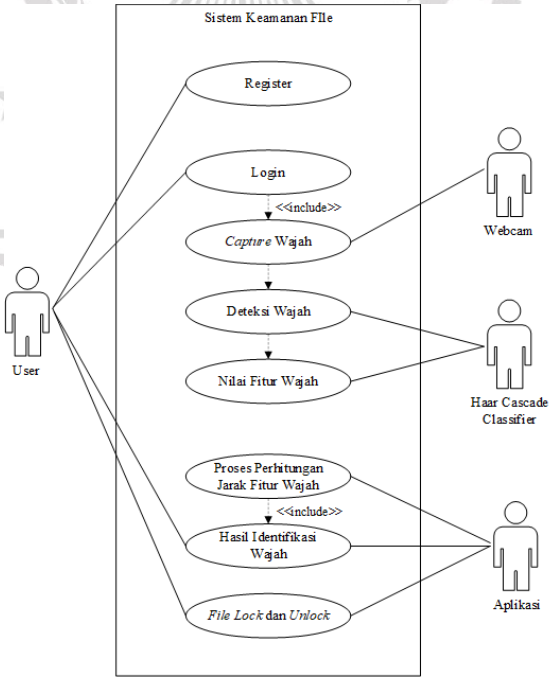
Perancangan sistem keamanan *file* ini menggunakan pengenalan wajah melalui *webcam* sebagai media pengambil objek yang akan diproses dan dibatasi dengan jarak tertentu. Pada saat pengguna ingin mengamankan serta mengakses *file*, maka terlebih dahulu *login* menggunakan wajah dengan cara

mendekatkan wajah ke arah laptop yang terdapat *webcam* untuk mengambil citra wajah orang tersebut. Citra wajah yang telah diambil nantinya akan dijadikan pembanding dan dicocokkan dengan data citra wajah yang telah disimpan. Jika citra wajah yang diambil ternyata tidak cocok dengan beberapa database yang telah ditentukan, maka pengguna tidak diizinkan untuk mengakses *file* tersebut, begitu juga sebaliknya jika citra wajah yang diambil ternyata cocok dengan salah satu data citra wajah yang telah ditentukan maka pengguna tersebut berhak mengakses *file* tersebut.

Adapun spesifikasi alat yang direncanakan antara lain unit pengambil citra wajah pengguna yang meliputi *webcam* dalam pengambilan wajah serta unit sentral yang merupakan sebuah laptop yang bertugas melakukan pengendalian secara keseluruhan yang didalamnya terdapat perangkat lunak yang dirancang untuk mengenali wajah pengguna.

3.4.1 *Use Case Diagram*

Use case diagram merupakan interaksi yang saling berkaitan antara sistem dengan actor atau pengguna. Berikut merupakan *use case* diagram dalam sistem.



Gambar 3.16 *Use Case Diagram*

Deskripsi pendefinisian *use case* dan deskripsi *use case* pada sistem dapat dilihat pada tabel tabel berikut :

Tabel 3.6 Definis Aktor

No	Skenario	Deskripsi
1	User	User harus terdaftar terlebih dahulu, setelah itu user login menggunakan pengenalan wajah dengan jarak antar fitur wajah, jarak fitur wajah tersebut sebagai identitas dari user tersebut dan user berhak mengakses file jika wajah telah terverifikasi dengan benar.
2	Webcam	Pengambilan citra atau gambar dari objek bergerak.
3	Haar Cascade Clasiffier	Mendeteksi objek begerak berupa wajah dan menentukan nilai dari antar fitur wajah yang dicari.
4	Aplikasi	Sistem yang akan melakukan proses peritungan dan menampilkannya sebagai informasi kepada user.

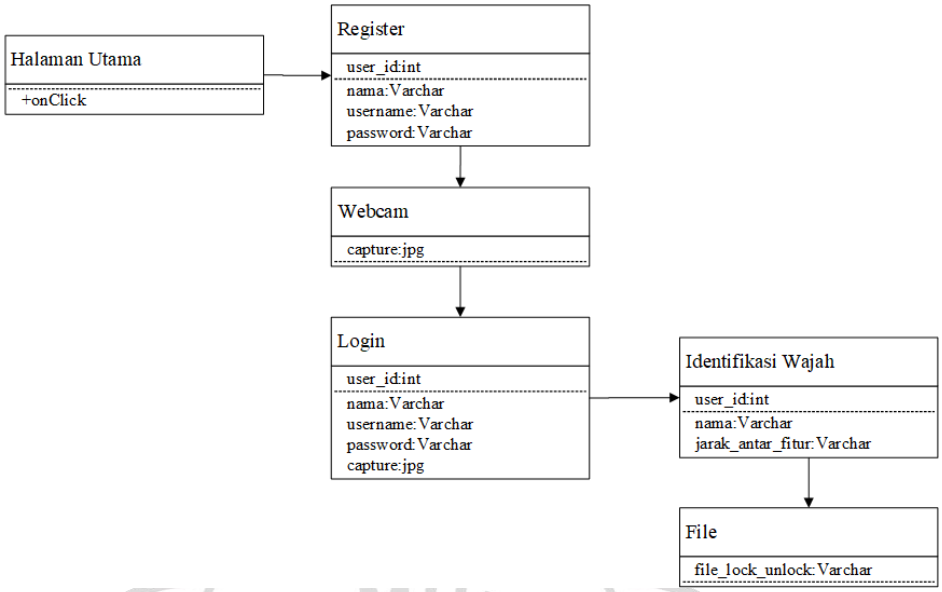
Tabel 3.7 Definisi *Use Case*

No.	Skenario	Deskripsi
1	*Register*	Pengguna melakukan pendaftaran akun baru.
2	*Login*	Pengguna *login* menggunakan wajah dan *password* sebagai keamanan berlapis agar lebih terjamin..
3	*Capture* wajah	Mengambil citra wajah pengguna dari webcam kemudian citra dikonversi dari *rgb* ke *gray* dan disimpan ke database.

4	Deteksi wajah	Pengguna akan dilakukan pendeteksian wajah, jika wajah tidak terdeteksi maka pengguna tidak akan bisa mencapture wajah.
5	Nilai Fitur Wajah	Untuk mencari nilai fitur wajah yang didapatkan dari haar cascade classifier yang nantinya digunakan sebagi pembeda antar user lainnya.
6	Proses perhitungan	Setelah nilai fitur wajah ditentukan, kemudian dilakukan pencocokan antar fitur wajah dengan menampilkan nilai koordinat fitur wajah yaitu mata, hidung, mulut (*triangle face*) kemudian fitur tersebut dihitung menggunakan jarak *minkowski distance* sebagai pengenalan wajah.
7	*File lock* dan *unlock*	Pengguna berhak mengakses, mengunci dan membuka *file.*

3.4.2 *Class Diagram*

Class diagram merupakan diagram yang sering dijumpai pada permodelan diagram digunakan untuk menunjukan interaksi antar class di dalam sistem.

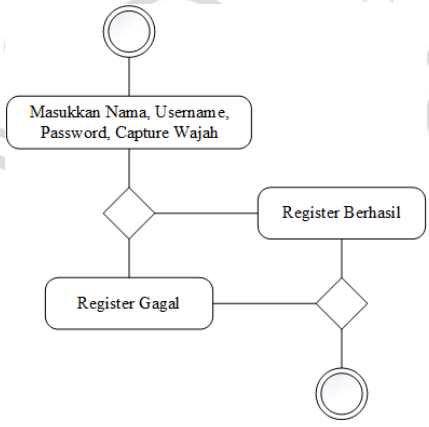


Gambar 3.17 *Class Diagram*

3.4.3 *Activity Diagram*

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem, tetapi bukan aktivitas aktor. Diagram aktivitas juga menggambarkan bagaimana alur sistem berawal, pilihan (decission) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut. Berikut ini diagram aktivitas pada sistem informasi yang dikembangkan.

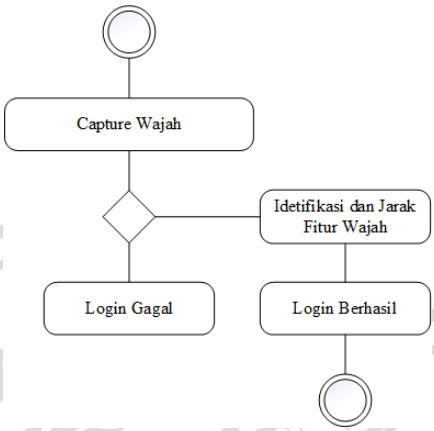
1. *Activity Diagram* : *Register*



Gambar 3.16 *Activity Diagram Register*

Diagram diatas menerangkan alur proses register pada pengguna, dimulai dari memasukan nama, username, password dan capture wajah, jika nama, username dan password benar login berhasil, jika nama, username dan password salah login gagal.

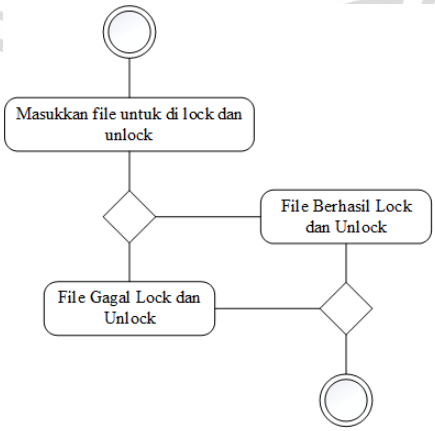
2. *Activity Diagram* : *Login*



Gambar 3.19 *Activity Diagram Login*

Diagram diatas menerangkan alur proses login pada pengguna, dimulai dari *capture* wajah, jika wajah sudah teridentifikasi dengan nilai antar fitur wajah maka otomatis *username* akan muncul. Selanjutnya pengguan memasukkan *password* manual sesuai denga *password* yang telah terdaftar. Jika wajah gagal teridentifikasi maka pengguna dinyatakan gagal login.

3. *Activity Daigram* : *File*



Gambar 3.20 *Activity Diagram File*

Diagram diatas menerangkan alur proses *file* pada pengguna, dimulai dari memasukan *file* yang ingin di *lock* dan di *unlock*. Berikut tahapan penguncian *file* dalam sistem :

1. Pilih *file* yang akan dikunci.
2. *File* dikunci dengan user masing-masing.

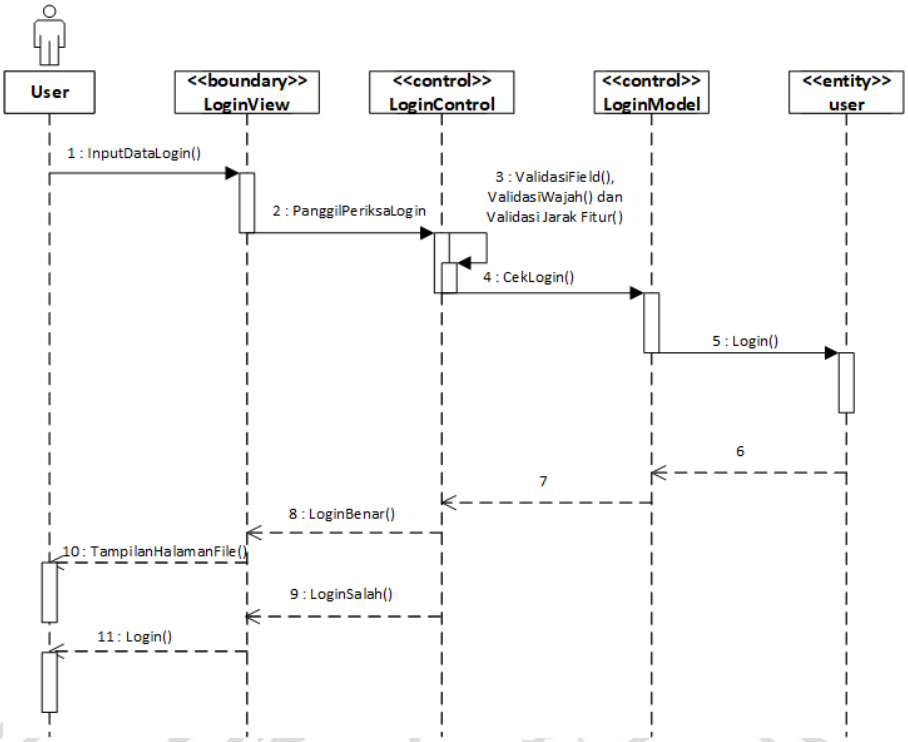
Sebaliknya jika pengguna ingin membuka *file* yang sudah terkunci. Berikut tahapan membuka file dalam sistem :

1. Pilih *file* yang akan dibuka.
2. *File* dibuka dengan user masing-masing.

3.4.4 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Proses menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta *methode* yang dimiliki kelas yang diinisialisasi menjadi objek yang sudah tergambar dalam *class* diagram.

Berdasarkan desain *use case*, terdapat beberapa *use case* yang prosesnya hampir sama satu sama lain. Untuk mempermudah pembahasan proses dalam pembuatan *sequence* diagram, berikut ini ringkasan diagram sekuen pada sistem yang dikembangkan:



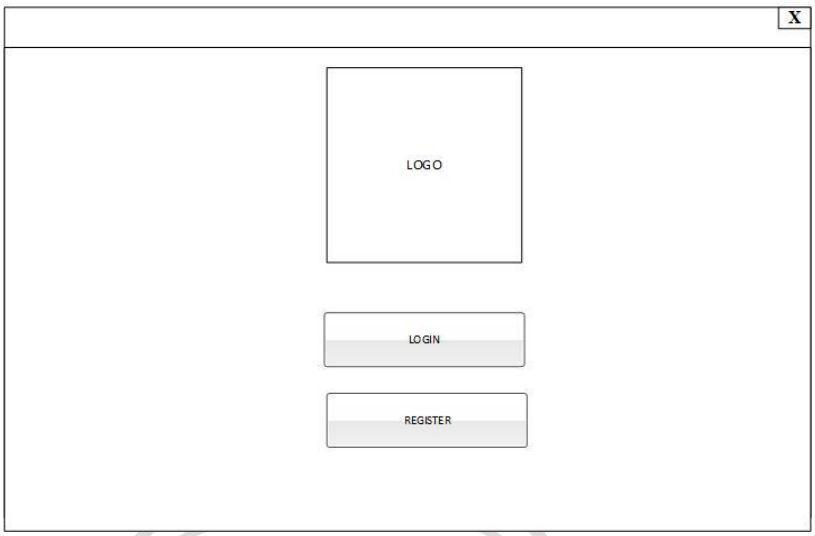
Gambar 3.21 *Squence* Diagram

3.5 Desain Interface

Desain interface merupakan penggambaran perencanaan tampilan sistem,tampilan antarmuka ini akan digunakan sebagai sarana informasi yang telah dikelola oleh sistem. Adapun perancangan antarmuka sistem yang dibuat adalah sebagai berikut :

3.5.1 Tampilan Halaman Utama

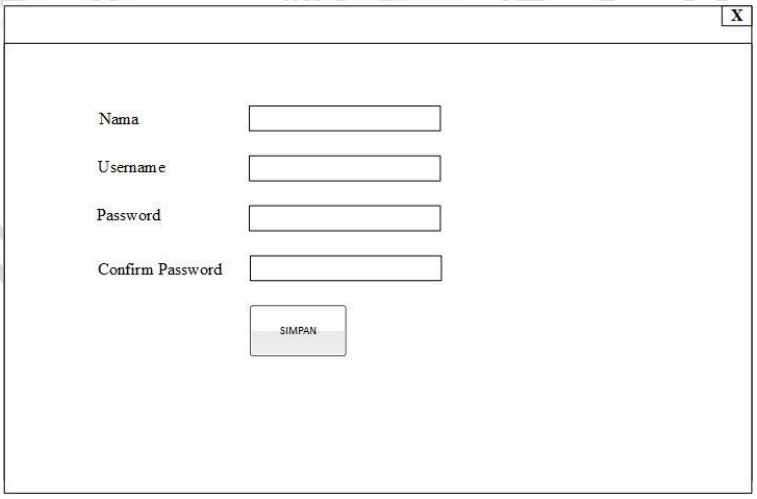
Halaman ini akan memuat halaman utama dari sistem yang berisi *login* dan *register*.



Gambar 3.22 Halaman Utama

3.5.2 Tampilan Halaman *Register*

Halaman ini akan memuat halaman pendaftaran data diri berserta wajah pengguna.



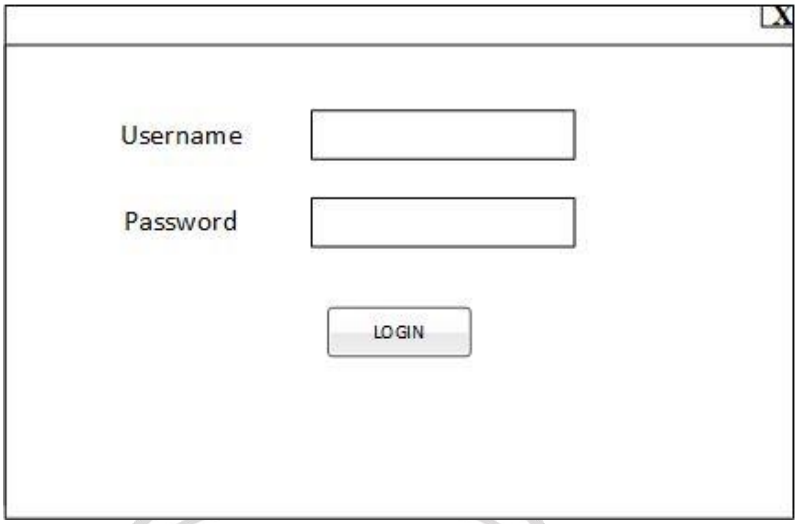
Gambar 3.23 Halaman *Register*

Gambar 3.24 Halaman *Capture* Wajah

3.5.3 Tampilan Halaman *Login*

Halaman ini akan memuat halaman *login* menggunakan wajah, setelah login wajah berhasil maka pengguna akan diminta untuk login kembali menggunakan *password* agar kemanan lebih terjamin, jika wajah tidak dikenali maka *button login* tidak akan aktif atau pengguna tidak dizinkan ke tahap selanjutnya.

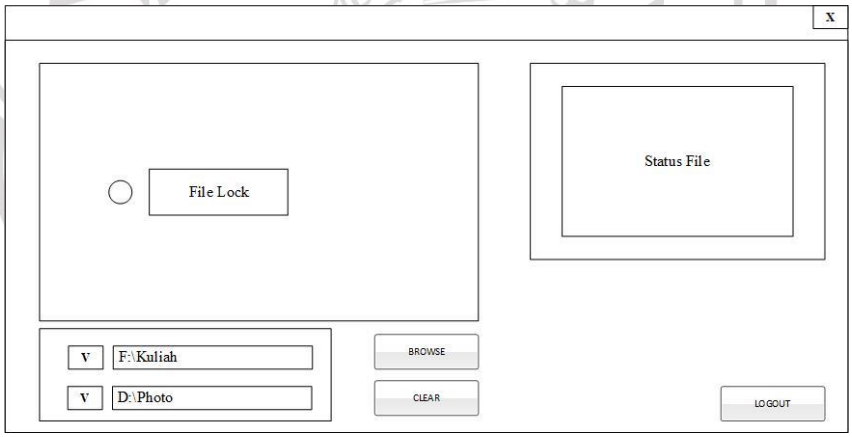
Gambar 3.25 Halaman *Login* menggunakan wajah



Gambar 3.26 Halaman *Login* menggunakan *password*

3.5.4 Tampilan Halaman *File Lock* dan *Unlock*

Halaman *file lock* dan *unlock* akan memuat halaman untuk mengunci dan membuka keamanan *folder*.



Gambar 3.27 Halaman *File Lock* dan *Unlock*

3.6 Skenario Pengujian

Adapun tahapan dalam pengujian sistem keamanan file, sebagai berikut :

a. Tahap pertama
    1. Sistem akan diuji oleh 20 sampel citra manusia untuk diambil bagian wajahnya.

2. Pengujian akan dilakukan mengambil wajah pengguna untuk disimpan ke database dan pencocokan wajah.
3. Pada pengujian ini dilakukan 5 nilai jarak fitur wajah yaitu mata kiri-mata kanan, mata kiri-hidung, mata kanan-hidung, mata kiri-mulut dan mata kanan-mulut.
4. Sistem akan memberikan notifikasi dapat mengakses *lock* & *unlock file* setelah wajah dikenali.

b. Tahap kedua

Pada penelitian ini, untuk mengukur kerja *accuracy* sistem yang dibangun dilakukan dengan *confusion matrix*.

3.7 Confusion Matrix

Confusion matrix dilakukan untuk menguji data set yang telah diklasifikasikan, hal ini dilakukan untuk menentukan seberapa baik sistem dalam mengklasifikasikan data. *Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Presetyo, 2012).

Untuk mengukur nilai akurasi dan dari hasil pengujian, dilakukan perhitungan dengan menggunakan metode *confusion matrix* sebagai berikut :

$$Akurasi = \frac{Hasil\ Benar}{Banyaknya\ Data} x100\%$$

$$Kesalahan = 100\% - Akurasi$$