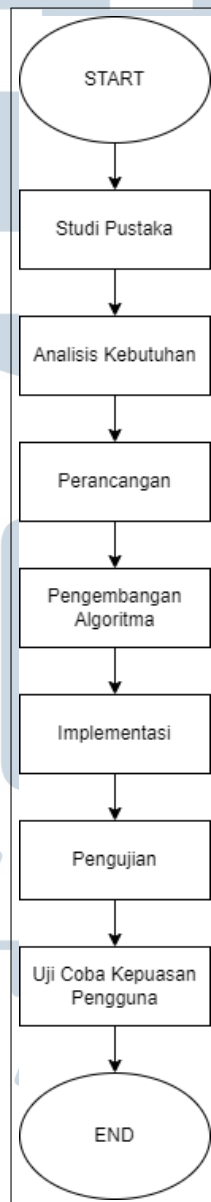


BAB 3 METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Pada bab ini akan dijelaskan langkah-langkah sistematis yang akan dilakukan dalam penelitian ini. Langkah-langkah ini akan digunakan untuk menyelesaikan masalah yang telah dibahas sebelumnya. Metode atau langkah-langkah yang digunakan adalah sebagai berikut:



Gambar 3.1. Flowchart tahapan penelitian

1. Studi Pustaka

Pada tahap ini dilakukan studi pustaka dengan mempelajari buku-buku referensi atau sumber-sumber yang berkaitan dengan penelitian ini, baik dari buku, jurnal, maupun internet yang membahas penggunaan metode *triangle face* dan *minkowski distance* untuk pengenalan wajah.

Selain itu, studi pustaka juga dilakukan untuk mengetahui metode-metode lain yang dapat digunakan untuk melakukan pengenalan wajah. Metode-metode lainnya dapat digunakan sebagai acuan dan perbandingan dalam penelitian ini.

2. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan yang diperlukan dalam penelitian. Analisis kebutuhan dibuat sehingga mempermudah pemahaman terhadap aplikasi yang akan dibangun.

(a) Analisis Data

Dalam penelitian ini sumber data yang dibutuhkan diambil dari *website* Kaggle. Data yang diambil berupa sekumpulan gambar/citra wajah, terdiri dari 1.680 orang dengan 2 - 50 gambar/citra per orang. Setiap gambar/citra memiliki ukuran 250×250 dan dikodekan dalam RGB.

(b) Analisis Proses

Pada tahap ini akan dilakukan analisis alur dari proses pengenalan wajah yang akan dilakukan. Tahap awal ketika citra/gambar telah didapatkan, maka akan dilakukan proses pada citra menggunakan *triangle face* untuk mendapatkan jarak antar fitur wajah. Selain citra masukan, citra/gambar wajah yang tersimpan dalam *database* juga diterapkan *triangle face* sebagai acuan. Kemudian pada tahap selanjutnya, nilai jarak antar fitur wajah yang telah didapatkan dari citra masukan dan citra acuan dihitung menggunakan *minkowski distance*. Nilai *minkowski distance* terkecil maka menandakan citra masukan dan citra acuan adalah sama.

3. Perancangan Sistem

Pada tahap ini, diberikan sebuah gambaran secara lengkap mengenai konsep yang diterapkan dalam pembuatan sistem. Untuk menggambarkan setiap

proses yang ada pada sistem, maka akan disertakan *use case diagram*, *activity diagram*, dan *database schema*.

4. Pengembangan Algoritma

Tahap ini meliputi pembuatan sistem keamanan menggunakan pengenalan wajah. Metode yang telah ada dijelaskan pada tahap penelitian sebelumnya akan dibangun berdasarkan rancangan yang telah dibuat sebelumnya. Spesifikasi perangkat keras dan perangkat lunak yang digunakan adalah sebagai berikut:

(a) Perangkat keras (*Hardware*)

- i. *Processor* : Intel(R) Core(TM) i7-8565U
- ii. Memori (RAM) : 8 GB
- iii. Kamera : USB2.0 HD UVC WebCam

(b) Perangkat lunak (*Software*)

- i. Sistem Operasi : OS Name Microsoft Windows 10

5. Implementasi

Sistem yang telah dikembangkan akan diimplementasikan pada aplikasi Android untuk melakukan login. Aplikasi yang telah berhasil dibuat akan menjadi hasil akhirnya yang dapat digunakan pengguna untuk mengakses dan mengubah data-data pribadi yang disimpan dalam *database*.

6. Pengujian

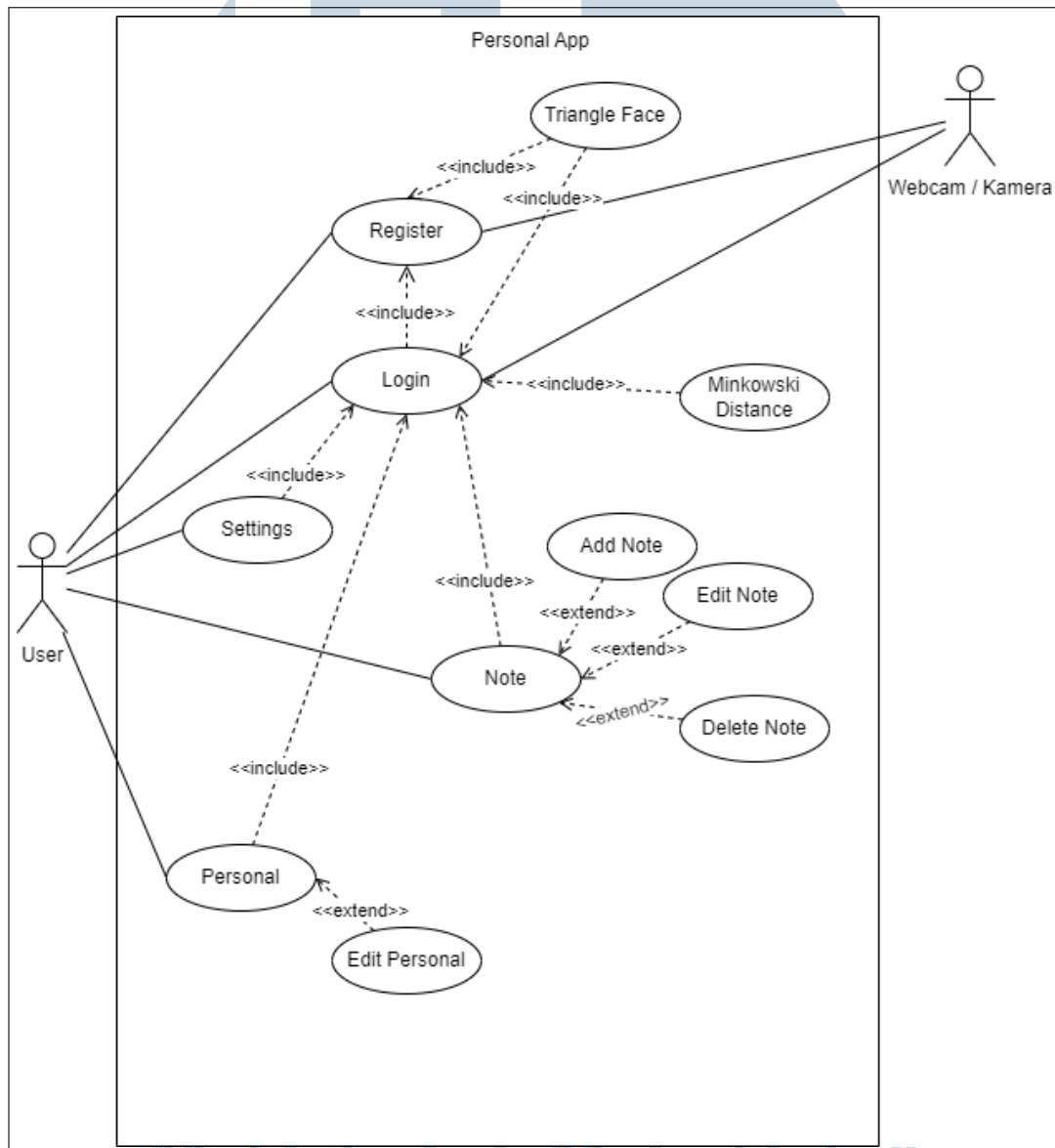
Pada tahap ini akan dilakukan pengujian terhadap sistem yang dibuat untuk mengetahui apakah sistem yang dibuat bekerja sesuai dengan yang diharapkan. Hasil pengenalan wajah menggunakan metode yang digunakan akan dihitung tingkat keberhasilannya dengan menghitung akurasi dalam melakukan pengenalan citra/gambar wajah. Hasil pengenalan wajah akan menentukan apakah seseorang dapat mengakses data yang disimpan atau tidak.

3.2 Perancangan Sistem

Perancangan sistem dibuat untuk mempermudah proses proses implementasi. Proses ini juga berguna untuk menghasilkan aplikasi yang sesuai dengan kebutuhan pengguna. Bagian ini meliputi *use case diagram*, *activity diagram*, dan *database schema*

3.2.1 Use Case Diagram

Use case diagram ini bertujuan untuk menguraikan gambaran umum dari sistem yang akan dibangun. Diagram ini akan menggambarkan keseluruhan cara kerja aplikasi secara garis besar. Use case diagram dapat dilihat pada Gambar 3.2.



Gambar 3.2. Use case diagram aplikasi Personal App

Pada diagram di atas terdapat 4 aktor utama yang berperan dalam memberikan informasi kepada sistem ataupun menerima informasi dari sistem. Aktor yang digunakan pada diagram akan dijelaskan pada Tabel 3.1 Definisi aktor.

Tabel 3.1. Definisi aktor

No	Aktor	Deskripsi
1	User	Aktor yang berperan sebagai pengguna aplikasi.
2	Webcam/Kamera	Aktor yang berperan dalam membantu pengambilan citra/gambar dari objek bergerak.

Aktor di atas akan berinteraksi dengan sistem yang ditunjukkan dengan garis dari aktor ke *use case* tertentu. *Use case* sendiri merupakan komponen gambaran fungsional yang terjadi pada sistem. Komponen ini menggambarkan fungsi yang dibangun dalam sistem. *Use case* yang digunakan pada diagram akan dijelaskan pada Tabel 3.2 Definisi *use case*.

Tabel 3.2. Definisi *use case*

No	Skenario	Deskripsi
1	<i>Register</i>	Pengguna melakukan pendaftaran baru dan data disimpan pada <i>database</i> .
2	<i>Login</i>	Pengguna <i>login</i> menggunakan pengenalan wajah.
3	<i>Triangle Face</i>	Melakukan pendeteksian wajah dari citra/gambar untuk mencari posisi fitur-fitur wajah dan menghitung jarak antar fitur (<i>triangle face</i>).
4	<i>Minkowski Distance</i>	Melakukan perbandingan nilai dari citra/gambar masukan dengan citra/gambar pada <i>database</i> untuk menentukan akses dapat diberikan atau tidak kepada pengguna.
5	<i>Note</i>	Melihat catatan-catatan yang disimpan pada <i>database</i> .
6	<i>Add Note</i>	Menambahkan catatan baru ke dalam <i>database</i> .
7	<i>Edit Note</i>	Mengubah catatan yang ada pada <i>database</i> .
8	<i>Delete Note</i>	Menghapus catatan dari <i>database</i> .
Lanjut pada halaman berikutnya		

Tabel 3.2 **Definisi use case** (lanjutan)

No	Skenario	Deskripsi
9	Personal	Melihat data-data pribadi yang disimpan pada <i>database</i> .
10	<i>Edit</i> Personal	Mengubah data pribadi yang tersimpan pada <i>database</i> .
11	<i>Settings</i>	Menampilkan atau mengubah data yang digunakan untuk <i>login</i> , yaitu <i>username</i> . Pada bagian ini juga dapat dilakukan penghapusan akun.

Diagram *use case* di atas memberikan gambaran alur kerja aplikasi Personal App yang akan dibangun. Untuk mengakses fitur-fitur pada aplikasi ini, pengguna harus terlebih dahulu melakukan *login*. Proses *login* juga harus memperhatikan apakah data pengguna telah terdaftar atau belum. Jika belum, maka pengguna harus melakukan registrasi (*register*) terlebih dahulu.

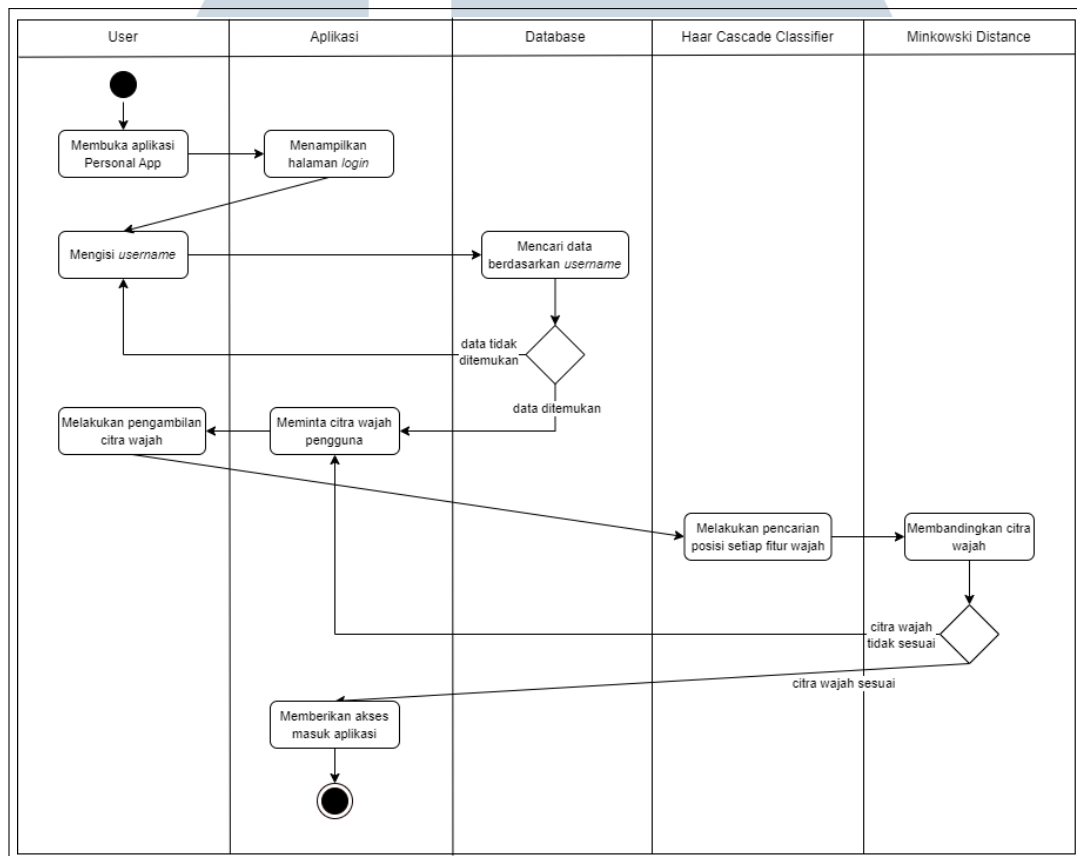
Proses *login* dan *register* akan melibatkan penggunaan kamera untuk melakukan pengambilan citra wajah. Citra wajah ini kemudian akan diproses menggunakan metode *triangle face* untuk mencari posisi fitur wajah. Pada proses *login* metode *minkowski distance* digunakan untuk mengetahui kemiripan citra wajah pengguna dengan citra wajah pada *database* sebagai pengambilan keputusan proses *login* berhasil atau tidak berhasil dilakukan.

3.2.2 Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas atau *activity diagram* merupakan diagram yang akan menggambarkan proses-proses yang terjadi pada sebuah sistem secara vertikal. *Activity diagram* merupakan pengembangan dari *use case* diagram yang memiliki alur aktivitas sehingga dapat memberikan gambaran yang lebih jelas mengenai proses yang terjadi.

A Diagram Aktivitas *Login*

Pada diagram pertama akan dijelaskan proses kerja aplikasi untuk melakukan *login* seperti yang terlihat pada Gambar 3.3. Proses untuk melakukan *login* akan melibatkan proses yang cukup panjang karena meliputi proses pengenalan wajah dan perbandingan jarak fitur wajah.



Gambar 3.3. Diagram aktivitas *login* pada aplikasi **Personal App**

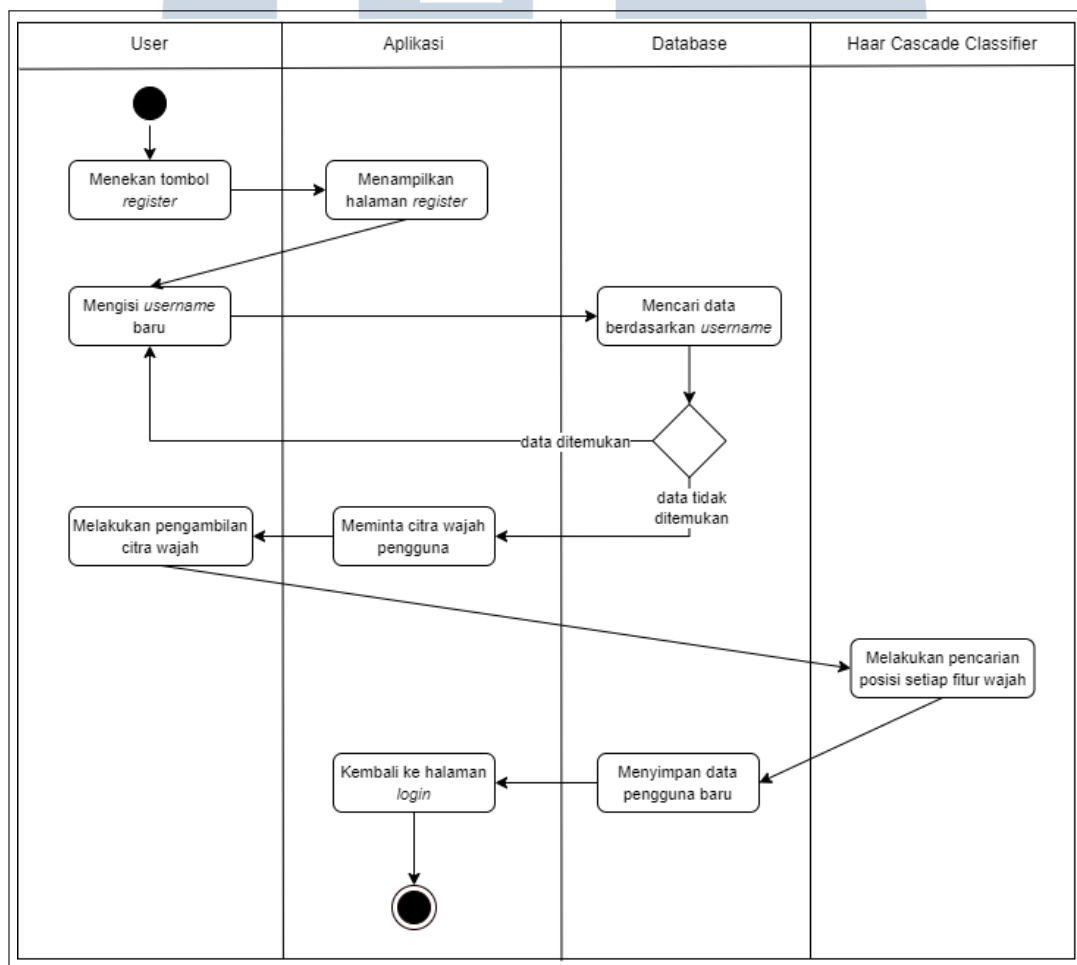
Pada diagram di atas dapat dilihat bahwa ketika pertama kali aplikasi dibuka oleh pengguna maka aplikasi akan menampilkan halaman *login*. Pada halaman ini pengguna harus mengisi data berupa *username* yang akan digunakan untuk mencari data-data pengguna pada *database*. Apabila data ditemukan, maka pengguna akan diminta untuk melakukan pengambilan citra wajah untuk dibandingkan dengan data yang telah terdaftar pada *database*.

Pada tahap ini citra wajah pengguna akan diproses dengan metode *triangle face* menggunakan *haar cascade classifier* untuk menentukan posisi setiap fitur wajah dan jaraknya. Jarak yang didapatkan ini kemudian akan dibandingkan dengan jarak yang tersimpan pada *database* menggunakan metode *minkowski distance*

untuk menentukan kemiripan kedua data. Jika citra wajah sesuai dengan data yang tersimpan, maka pengguna dapat mengakses aplikasi dan proses *login* berhasil.

B Diagram Aktivitas *Register*

Diagram aktivitas ini akan menampilkan proses yang terjadi ketika pengguna melakukan *register*. Proses ini dilakukan untuk memasukkan data pengguna baru ke dalam *database* sehingga dapat digunakan untuk *login* ke aplikasi. Diagram aktivitas *register* dapat dilihat pada Gambar 3.4.



Gambar 3.4. Diagram aktivitas *register* pada aplikasi **Personal App**

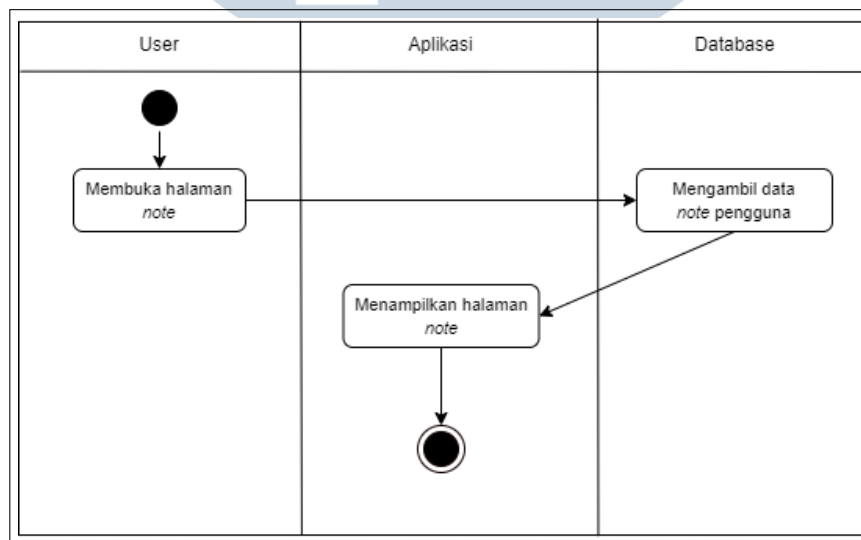
Ketika halaman *register* terbuka, pengguna akan diminta untuk mengisi *username* baru. Data *username* ini bersifat unik sehingga tidak boleh terdapat *username* yang sama dalam *database*. Maka dari itu ketika melakukan pendaftaran, akan dilakukan pengecekan terlebih dahulu ke *database* apakah *username* yang akan didaftarkan telah digunakan atau belum. Jika *username* telah terdaftar pada

database, maka pengguna akan diminta untuk mengganti *username* yang akan digunakan.

Setelah *username* berhasil digunakan, pengguna akan diminta untuk melakukan pendaftaran wajah menggunakan kamera. Citra wajah pengguna kemudian akan diproses dengan metode *triangle face* menggunakan *haar cascade classifier* untuk menentukan posisi setiap fitur wajah dan jarak antar fitur tersebut. Data jarak yang didapatkan kemudian akan disimpan dalam *database* sebagai instrumen untuk melakukan *login*. Setelah proses registrasi selesai, pengguna akan kembali ke halaman *login*.

C Diagram Aktivitas Note

Diagram aktivitas ini akan memperlihatkan proses yang terjadi ketika pengguna membuka halaman *note*. Halaman ini akan menampilkan catatan pengguna yang tersimpan dalam *database*. Diagram untuk halaman ini dapat terlihat pada Gambar 3.5 Diagram aktivitas halaman *note*.

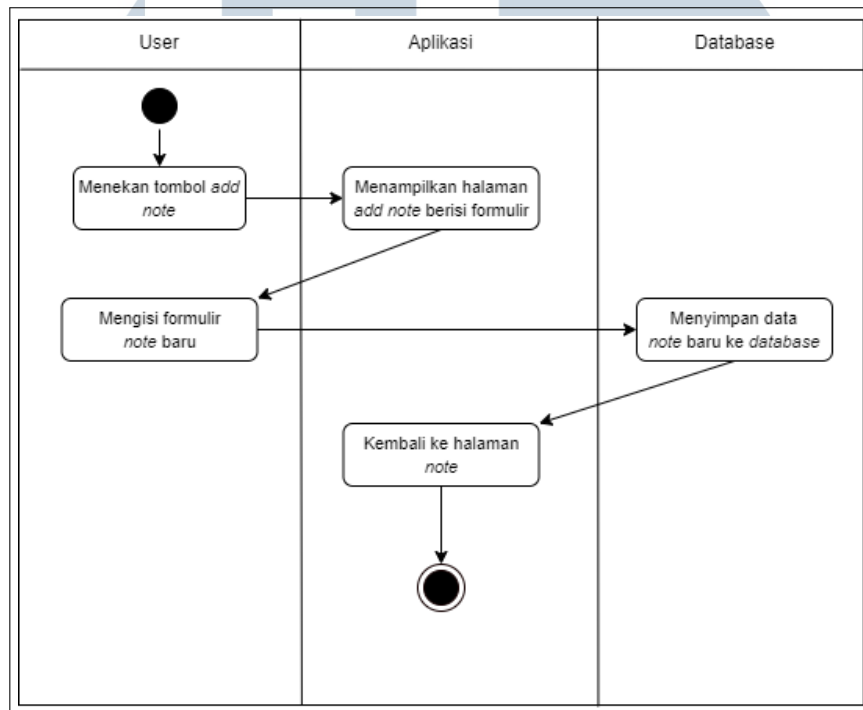


Gambar 3.5. Diagram aktivitas halaman *note*

Pada diagram dapat dilihat ketika pengguna membuka halaman *note*, maka data *note* atau catatan pengguna yang tersimpan di *database* akan diambil. Kemudian ketika halaman *note* ditampilkan pada aplikasi, data seluruh *note* yang telah diambil akan ditampilkan juga.

D Diagram Aktivitas *Add Note*

Pada diagram ini akan ditampilkan proses yang terjadi ketika pengguna menekan tombol untuk menambahkan *note* atau catatan baru. Proses ini dapat dilihat pada Gambar 3.6 Diagram aktivitas *add note*.

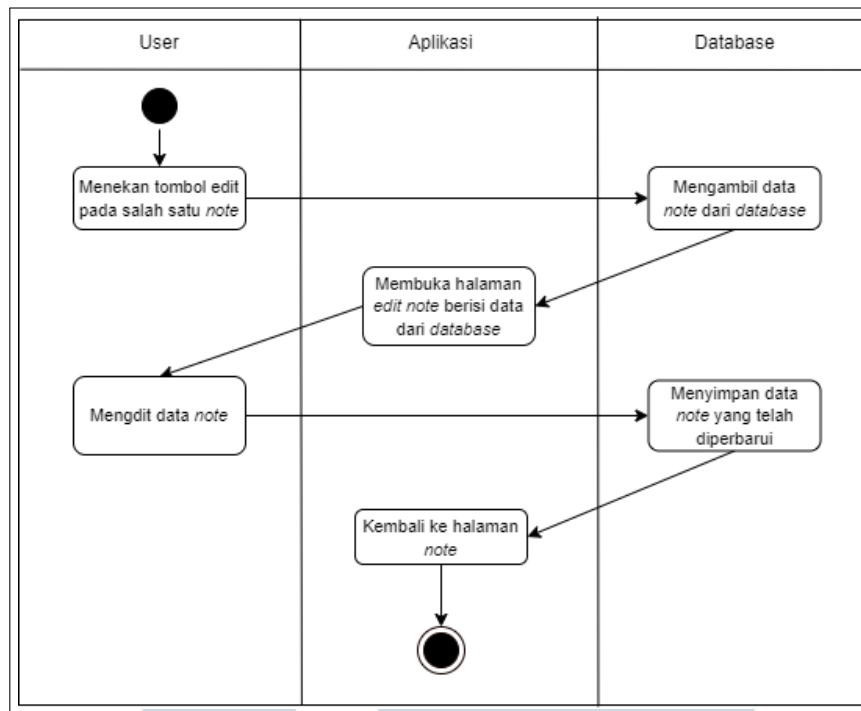


Gambar 3.6. Diagram aktivitas *add note*

Ketika pengguna menekan tombol *add note*, maka aplikasi akan menampilkan halaman baru berisi formulir catatan baru yang harus diisi. Apabila formulir telah diisi dan pengguna menekan tombol untuk menyimpan data, maka sistem akan mengirim data tersebut ke *database* dan disimpan. Setelah seluruh proses telah selesai, pengguna akan kembali ke halaman *note*.

E Diagram Aktivitas *Edit Note*

Data *note* atau catatan pengguna yang tersimpan pada *database* dapat diubah oleh pengguna. Proses ini dapat dilihat pada diagram aktivitas pada Gambar 3.7 Diagram aktivitas *edit note*.

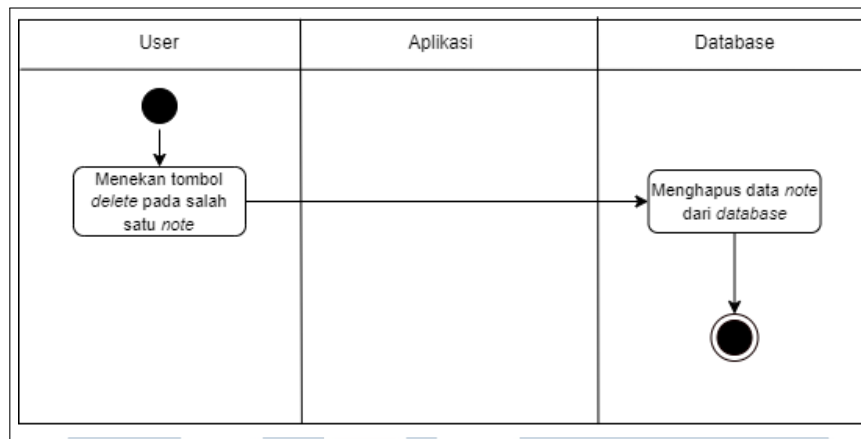


Gambar 3.7. Diagram aktivitas *edit note*

Ketika pengguna ingin mengubah salah satu *note* atau catatan, maka data catatan tersebut akan diambil dari *database*. Setelah data didapatkan, pengguna akan diarahkan ke halaman berisi data yang akan diubah. Ketika pengguna selesai mengubah data dan menekan tombol untuk menyimpan data baru, maka sistem akan mengirimkan kembali data tersebut ke *database* untuk disimpan kembali. Setelah semua proses selesai, pengguna akan kembali ke halaman *note*.

F Diagram Aktivitas *Delete Note*

Pengguna dapat menghapus catatan atau *note* yang tersimpan di *database*. Data ini akan terhapus secara permanen dan tidak dapat dikembalikan lagi. Proses untuk menghapus data ini dapat terlihat pada Gambar 3.8 Diagram aktivitas *delete note*.

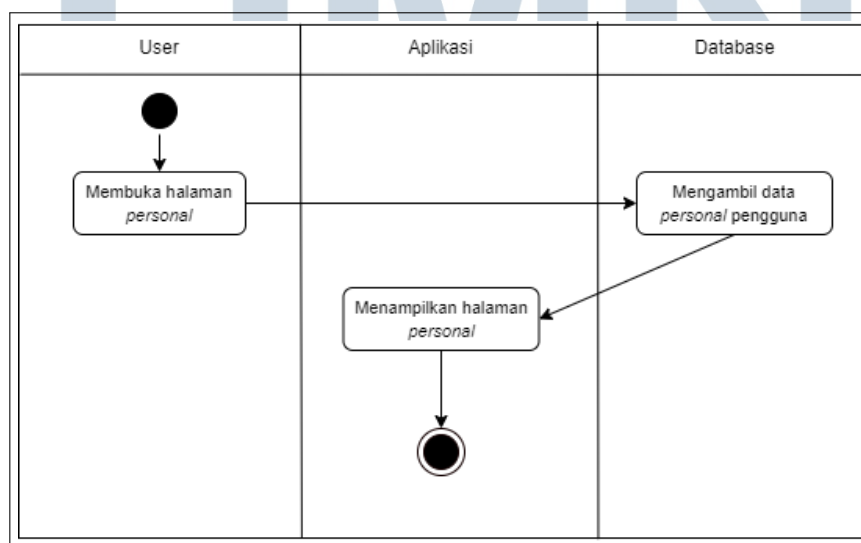


Gambar 3.8. Diagram aktivitas *delete note*

Proses penghapusan data akan dilakukan secara langsung ke *database* tanpa harus berpindah ke halaman lain. Seperti yang terlihat pada diagram, setelah pengguna memberikan perintah penghapusan data, maka perintah tersebut akan diteruskan ke *database* untuk menghapus data *note* atau catatan tertentu. Maka secara langsung data *note* atau catatan akan hilang dari *database* secara permanen.

G Diagram Aktivitas Personal

Fitur lainnya yang dapat diakses adalah halaman personal. Halaman ini akan menampilkan data-data pribadi pengguna yang tersimpan di *database*. Proses yang terjadi ketika pengguna membuka halaman personal dapat dilihat pada Gambar 3.9 Diagram aktivitas halaman personal.

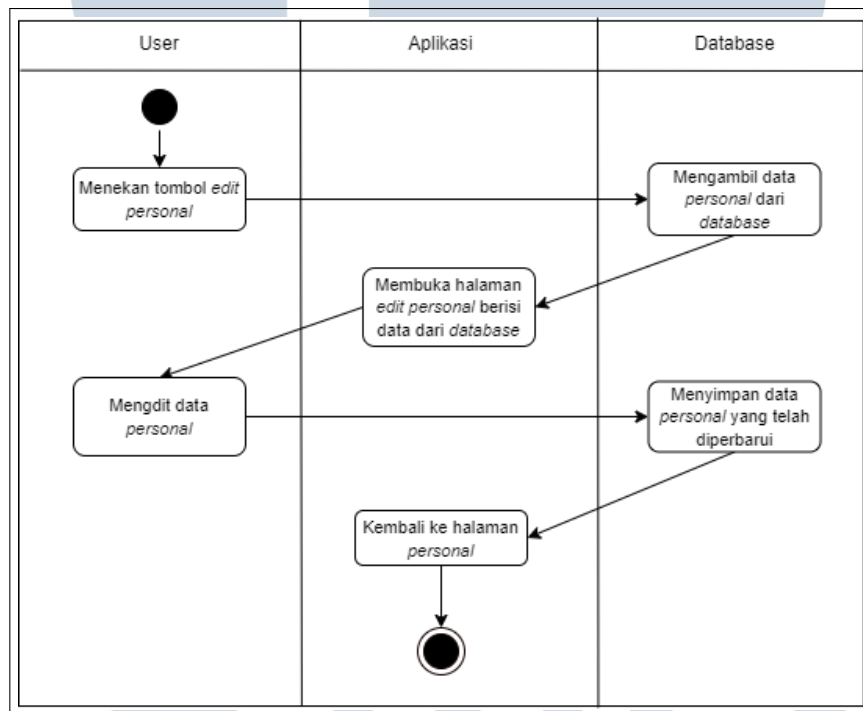


Gambar 3.9. Diagram aktivitas halaman personal

Data pribadi pengguna yang tersimpan di *database* akan diambil ketika pengguna membuka halaman personal. Data tersebut kemudian akan ditampilkan pada halaman personal. Ketika pertama kali membuat akun, data pada halaman ini akan kosong, sehingga pengguna harus terlebih dahulu mengisi data-data tersebut.

H Diagram Aktivitas *Edit Personal*

Data pribadi pengguna yang terdapat pada halaman personal dapat diubah melalui fitur *edit* personal. Proses ini akan membawa pengguna ke halaman baru untuk mengubah data yang tersimpan. Diagram aktivitas *edit* personal dapat dilihat pada Gambar 3.10 Diagram aktivitas *edit* personal.

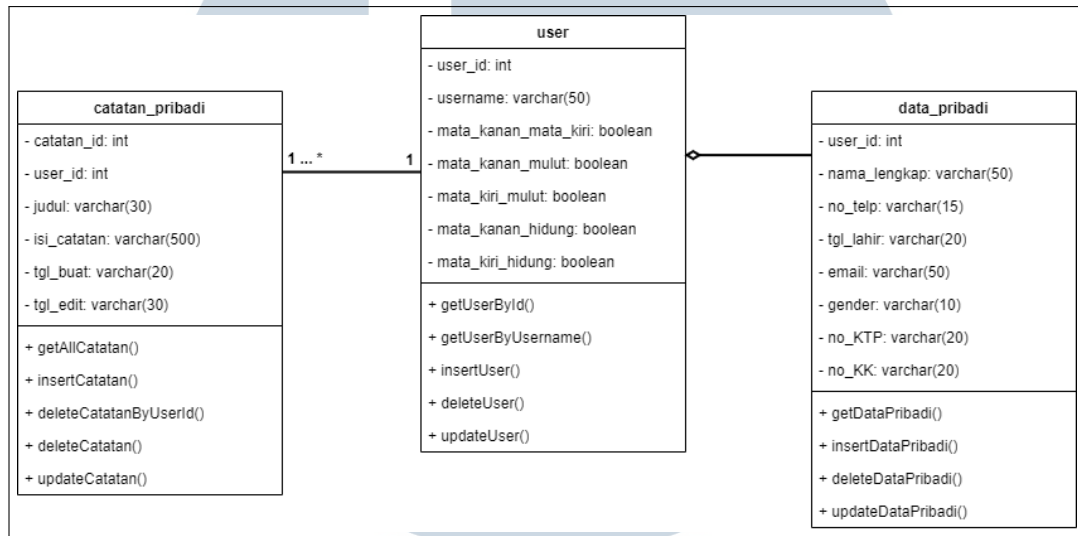


Gambar 3.10. Diagram aktivitas *edit* personal

Ketika pengguna menekan tombol *edit* personal, data pengguna akan diambil dari *database* untuk kemudian ditampilkan pada halaman *edit*. Kemudian pengguna dapat melakukan perubahan pada data dan menekan tombol simpan ketika selesai. Dengan menekan tombol simpan, data baru yang diberikan pengguna akan dikirim ke *database* untuk disimpan kembali. Setelah proses selesai, pengguna akan kembali ke halaman personal.

3.2.3 Diagram Kelas (*Class Diagram*)

Class diagram atau diagram kelas merupakan salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi kelas, atribut, metode, dan hubungan dari setiap objek. Diagram kelas yang digunakan pada aplikasi ini dapat dilihat pada Gambar 3.11 Diagram kelas aplikasi.

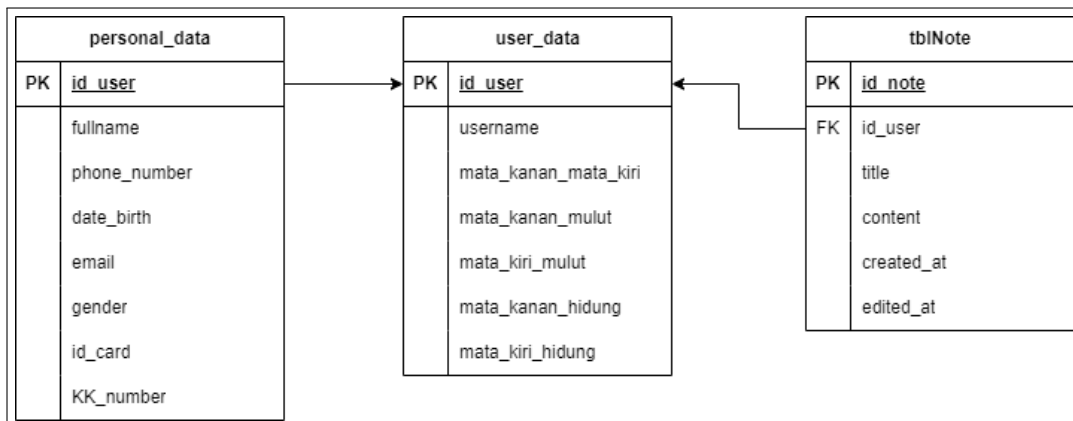


Gambar 3.11. Diagram kelas aplikasi

Pada diagram kelas di atas, dapat dilihat bahwa terdapat hubungan antara kelas `user` ke kelas `catatan_pribadi`. Hubungan ini merupakan hubungan asosiasi dua arah yang mana satu `user` dapat terhubung pada lebih dari satu data `catatan_pribadi`. Kemudian terdapat juga hubungan kelas `data_pribadi` ke kelas `user` dengan hubungan agregasi. Hubungan ini menunjukkan bahwa kedua kelas berhubungan, namun masing-masing kelas tetap dapat berdiri sendiri tanpa satu sama lain.

3.2.4 Skema Database (*Database Schema*)

Skema *database* atau *database schema* merupakan diagram yang menjabarkan suatu rangkaian data. Diagram ini akan menunjukkan bagaimana data akan disimpan dan dihubungkan pada *database*. Pada aplikasi ini terdapat 3 tabel yang dibuat, yaitu `data_user`, `data_personal`, dan `tblNote`. Struktur dan relasi antar tabel dapat dilihat pada Gambar 3.12 Skema *database* aplikasi.



Gambar 3.12. Skema database aplikasi

Tabel data_user akan menyimpan data yang berkaitan dengan proses *login*, yakni *username* dan posisi fitur wajah. Data ini yang akan dipanggil apabila pengguna melakukan *login* untuk mencocokkan jarak antar fitur wajah yang disimpan dengan jarak antar fitur wajah yang dimasukkan pengguna. Selain memiliki *username* yang berbeda-beda, tabel ini juga memiliki kolom *id_user* yang berbeda-beda pula sebagai *primary key*. Kolom ini membantu memisahkan data-data yang akan diambil dari tabel data_personal maupun tblNote.

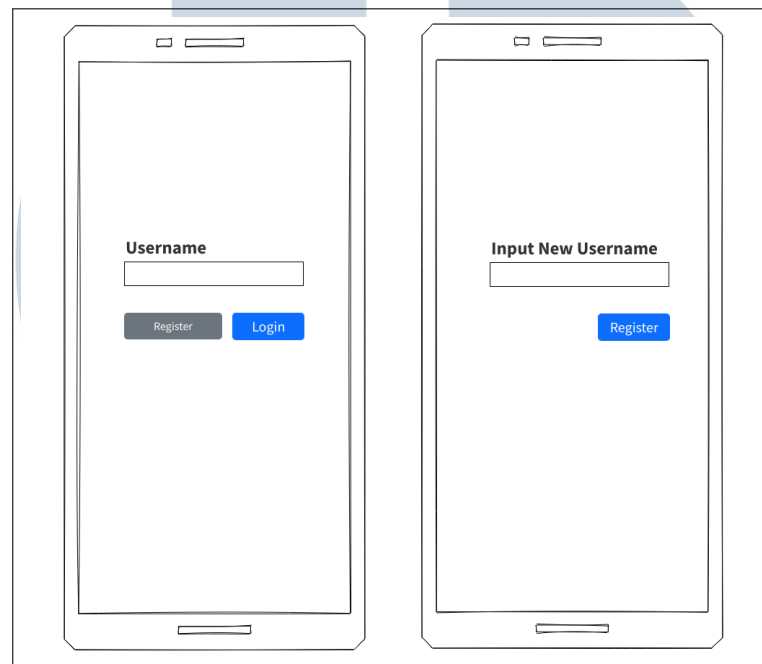
Tabel kedua adalah data_personal yang menyimpan data-data pribadi pengguna, yaitu nama lengkap pada kolom *fullname*, nomor telepon pada kolom *phone_number*, tanggal lahir pada kolom *date_birth*, email pada kolom *email*, jenis kelamin pada kolom *gender*, nomor KTP pada kolom *id_card*, dan nomor KK pada kolom *kk_number*. Tabel ini memiliki *primary key* pada kolom *id_user*. Data yang disimpan pada tabel ini harus memiliki *id_user* yang sama dengan data yang disimpan pada tabel data_user untuk pengguna yang sama.

Tabel terakhir adalah tblNote yang menyimpan data mengenai catatan pribadi setiap pengguna. Data yang disimpan berupa judul pada kolom *title*, isi catatan pada kolom *content*, waktu catatan dibuat pada kolom *created_at* dan waktu catatan diubah pada kolom *edited_at*. Selain itu tabel ini juga memiliki kolom *id_note* sebagai *primary key*, dan kolom *id_user* sebagai *foreign key* yang terhubung ke tabel data_user. Kolom *id_user* ini membantu dalam pemanggilan data berdasarkan pengguna yang melakukan *login* pada aplikasi.

3.3 Rancangan Antar Muka

Rancangan antar muka pada aplikasi Android yang dibuat akan memiliki beberapa halaman utama. Yang pertama adalah halaman *login* dan halaman

register yang akan menjadi tampilan utama pada aplikasi. Kedua halaman ini akan diimplementasikan dengan sistem pengenalan wajah Rancangan antar muka kedua halaman ini dapat dilihat pada Gambar 3.13 Rancangan antar muka halaman *login* (kiri) dan halaman *register* (kanan).



Gambar 3.13. Rancangan antar muka halaman *login* (kiri) dan halaman *register* (kanan)

Selanjutnya apabila pengguna berhasil melakukan *login*, maka akan masuk pada halaman *dashboard* yang memiliki tiga halaman utama. Halaman pertama adalah *note* yang dapat digunakan untuk menyimpan catatan pribadi pengguna. Halaman kedua adalah *personal* yang dapat digunakan untuk menyimpan data pribadi seperti nama lengkap, nomor KTP, dan nomor telepon. Halaman terakhir adalah *settings* yang dapat digunakan untuk mengubah *username* yang digunakan pengguna untuk melakukan *login*. Rancangan antar muka untuk halaman *dashboard* dapat dilihat pada Gambar 3.14 Rancangan antar muka halaman *dashboard*.

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



Gambar 3.14. Rancangan antar muka halaman *dashboard*

Halaman *dashboard* pertama adalah *note*. Pada halaman ini terdapat tombol yang dapat digunakan untuk menambahkan catatan. Catatan yang telah disimpan dalam *database* akan ditampilkan dalam bentuk *list*. Catatan yang telah disimpan dapat diubah ataupun dihapus melalui halaman ini.

Halaman kedua pada *dashboard* adalah *personal* yang berisi data pribadi pengguna. Untuk mengubah data, terdapat tombol *edit* yang dapat digunakan. Dan halaman *dashboard* terakhir adalah *settings*. Pada halaman ini juga terdapat tombol *edit* untuk mengubah data yang tersimpan pada *database*.

UMIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA