# **BAB III METODOLOGI PENELITIAN**

**MACAM METODE PENELITIAN**

**-- METODE PENELITIAN KUALITATIF**

**\* Metode Etnografi**

**\* Metode Studi Kasus**

**\* Metode Fenomenologi**

**\* Metode Grounded Theory**

**-- METODE PENELITIAN KUANTITATIF**

**\* Metode Deskriptif-Kuantitatif**

**\* Metode Korelasional**

**\* Metode Kuasi Eksperimental (\*)**

**\* Metode Ekperimental (\*)**

**\* Metode Meta-Analisis**

**-- METODE PENELITIAN CAMPURAN (MIXED METHOD)**

BAB III Tugas Akhir nya mengunakan salah satu Metode Penelitian.... Kalau penelitiannya membuat aplikasi (Software) maka Metodologi penelitian dan metode pengembangannya di buat juga.

Metode Pengembangan Software

-- Metode Pengembangan Agile

-- Metode Pengembangan Waterfall

-- Metode Pengembangan Lean

-- Metode Prototipe

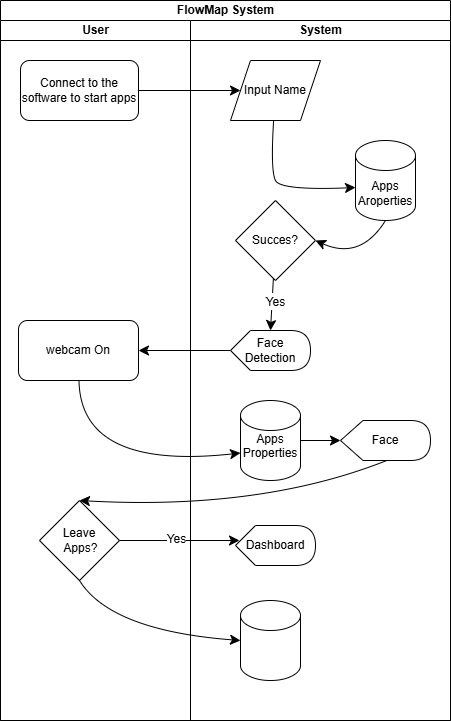
-- Metode Pengembangan Scrum

(Lihat buku Rekayasa Perangkat Lunak Roger S. Pressman).

## ***Analysis* (Analisis Kebutuhan)**

Dalam upaya untuk mengembangkan aplikasi pendeteksi *mood* mahasiswa tingkat akhir berbasis *android*, penelitian ini menggunakan metode pengembangan perangkat lunak yang terstruktur dan sistematis. Aplikasi ini dirancang untuk memanfaatkan teknologi terkini seperti Dart, Flutter, dan TensorFlow, serta metode klasifikasi *Haar* *Cascade* untuk pengenalan wajah. Dart dan Flutter dipilih karena kemampuannya untuk membangun antarmuka pengguna yang responsif dan performa tinggi pada perangkat *mobile*. TensorFlow, dengan dukungan TensorFlow Lite, memungkinkan integrasi model *machine* *learning* yang efisien untuk deteksi dan analisis ekspresi wajah secara *real*-*time*. Metode *Haar* *Cascade*, yang dikenal karena kecepatan dan efisiensinya dalam deteksi wajah, akan digunakan sebagai fondasi untuk proses pengenalan wajah.

Pendekatan ini bertujuan untuk menghasilkan aplikasi yang tidak hanya akurat dalam mendeteksi *mood* tetapi juga cepat dan *user*-*friendly*, sehingga dapat digunakan secara efektif oleh mahasiswa. Metodologi penelitian ini mencakup tahap-tahap yang jelas, mulai dari studi literatur, desain sistem, pengembangan, hingga pengujian dan evaluasi, untuk memastikan hasil yang optimal dan aplikasi yang fungsional. Dengan melakukan suatu analisis dapat mengetahui apa saja kebutuhan dalam membangun suatu aplikasi.



Gambar 6. Flowmap System

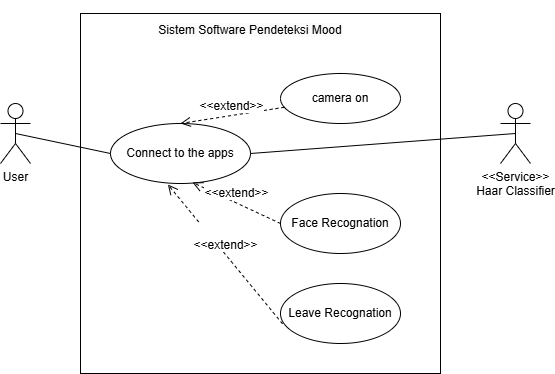
## ***System and Software Design***

Pada tahap ini, dilakukan sebuah perancangan desain atau antarmuka program yang akan dibuat. Perancangan ini dapat membantu dalam medefinisikan arsitektur sistem yang akan dibuat secara keseluruhan. Tahapan perancangan desain ini meliputi perancangan sistem dan perancangan *interface* (antarmuka).

1. Perancangan Sistem

Dalam perancangan sistem, UML (*Unified* *Modeling* *Language*) digunakan sebagai bahasa pemodelan dan analisis dalam perancangan ini. Perancangan aplikasi ini menggunakan UML jenis *use* *case* diagram, *activity* *diagram*, *sequence* *diagram*, dan *class* *diagram*. Berikut penjelasannya :

* + 1. *Use Case Diagram*

Pada sistem perancangan aplikasi ini, use case diagram menggambarkan suatu sistem aplikasi pendeteksi *mood* menggunakan algoritma *Haar Cascade*.

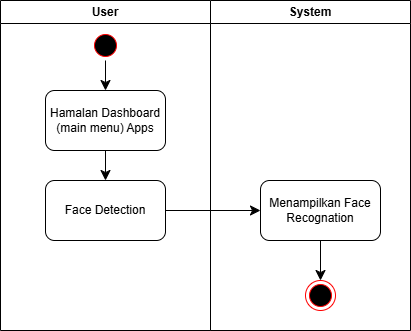
Gambar 7. Use Case Diagram software pendeteksi mood

Dapat dilihat pada gambar 7, Bagaiamana interaksi user dengan sistem dapat bekerja. User adalah pengguna aktif yang akan menjalankan suatu sistem dalam aplikasi. Untuk memulai *software*, *user* terlebih dahulu harus *connect (start the apps)* ke sistem *face* *recognation*, untuk dapat melakukan *face* *detection* dan melakukan keluar atau *leave recognation.*

* + 1. *Activity Diagram*

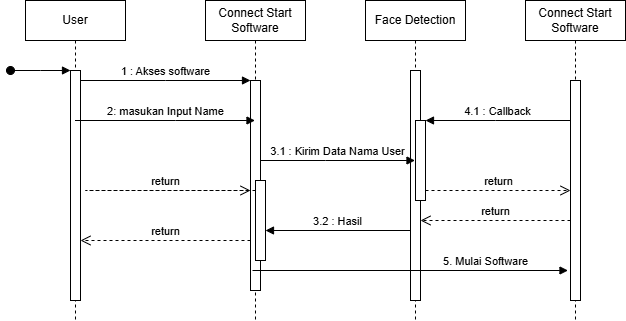
*Activity diagram* adalah gambaran alur kerja yang terjadi dalam sistem yang mengacu pada use case diagram yang ada. Pada aplikasi ada 2 *activity diagram* yaitu:

1. *Activity* *Diagram* *Face* *Detection*

*Activity* diagram *face detection* dimulai setelah *user* memilih menu *dashboard* pada halaman *main* *menu*, kemudian sistem menampilkan kamera untuk melakukan *record expression.*

Gambar 9. Activity diagram software pendeteksi mood

* + 1. *Sequence* Diagram

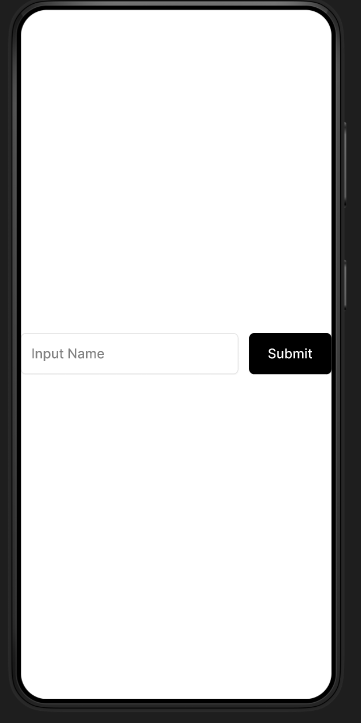
Sequence diagram menggambarkan interaksi obyek yang ada di dalam sistem pada software. Berikut perancangannya: 

Gambar 10. Sequence diagram software pendeteksi mood

* 1. *Interface* (Perancangan Antarmuka)

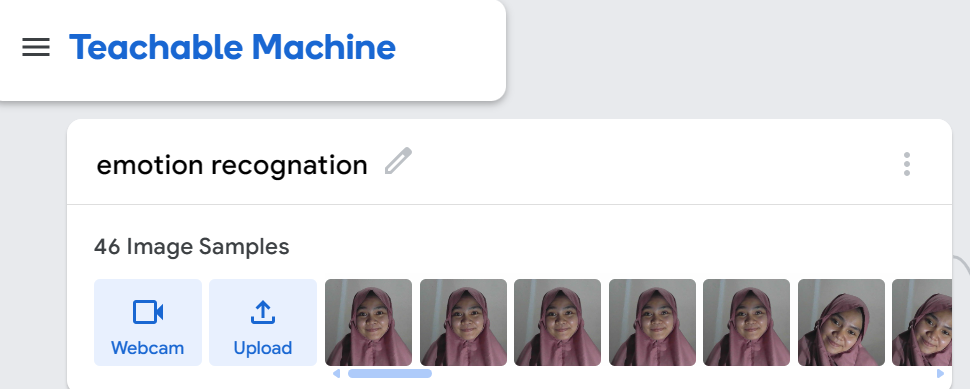
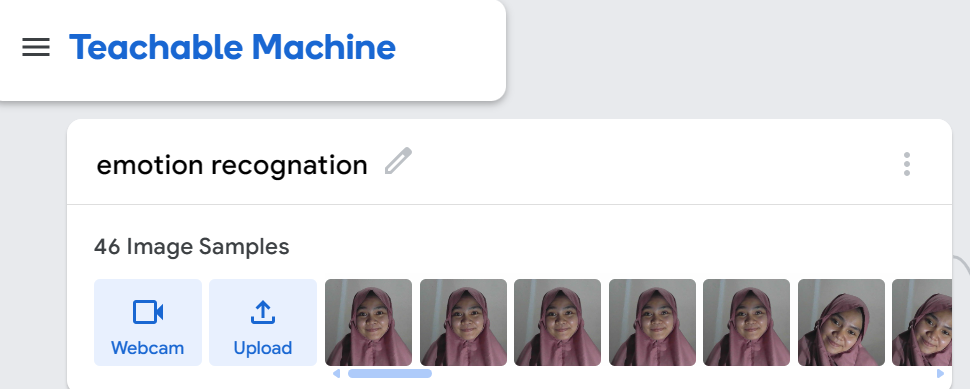
Perancangan antarmuka (interface) merupakan proses krusial dalam pembuatan perangkat lunak dan aplikasi, bertujuan untuk memastikan kemudahan dan kenyamanan bagi pengguna. Dalam tahap ini, peneliti mempertimbangkan tampilan dan fungsionalitas antarmuka agar intuitif dan user-friendly, dengan memperhatikan elemen seperti tata letak, warna, font, dan navigasi. Fokus pada kebutuhan pengguna adalah kunci, sehingga desain antarmuka bertujuan untuk meningkatkan kepuasan pengguna dan memungkinkan penggunaan produk secara lebih efisien.

1. Halaman *dashboard* (*main menu*)

*Halaman dashboard adalah halaman awal yang muncul saat aplikasi* pertama kali dijalankan.

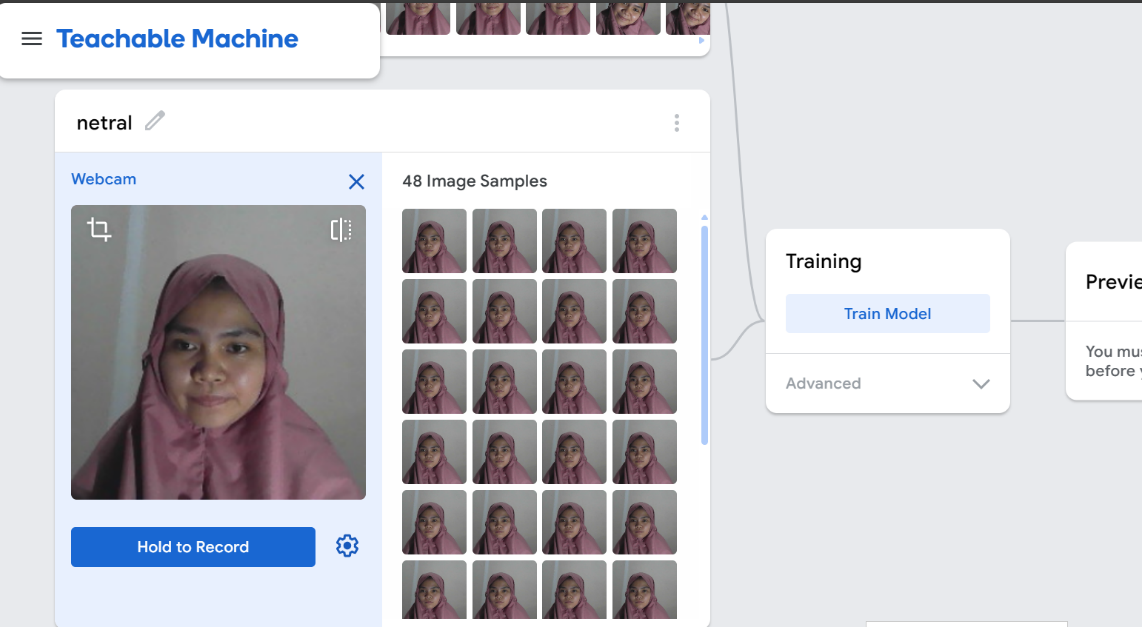
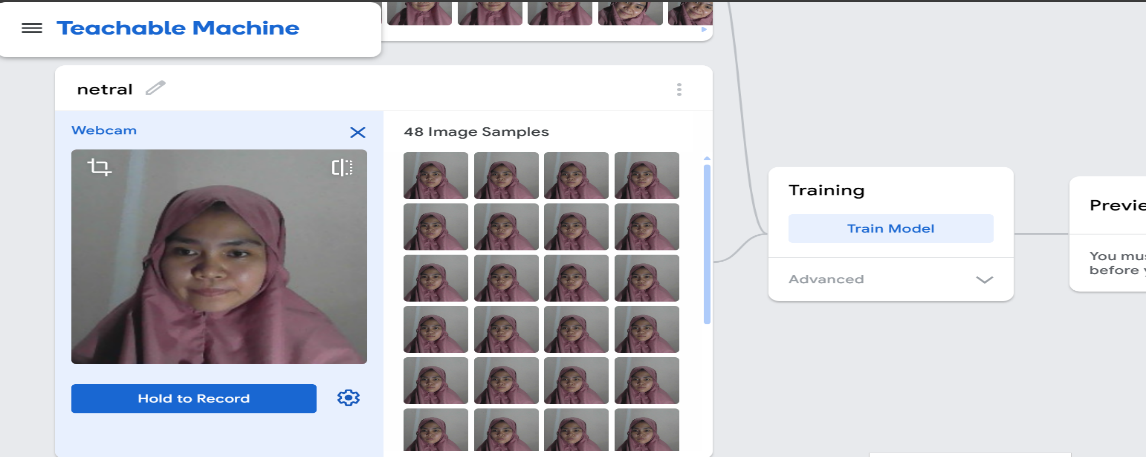
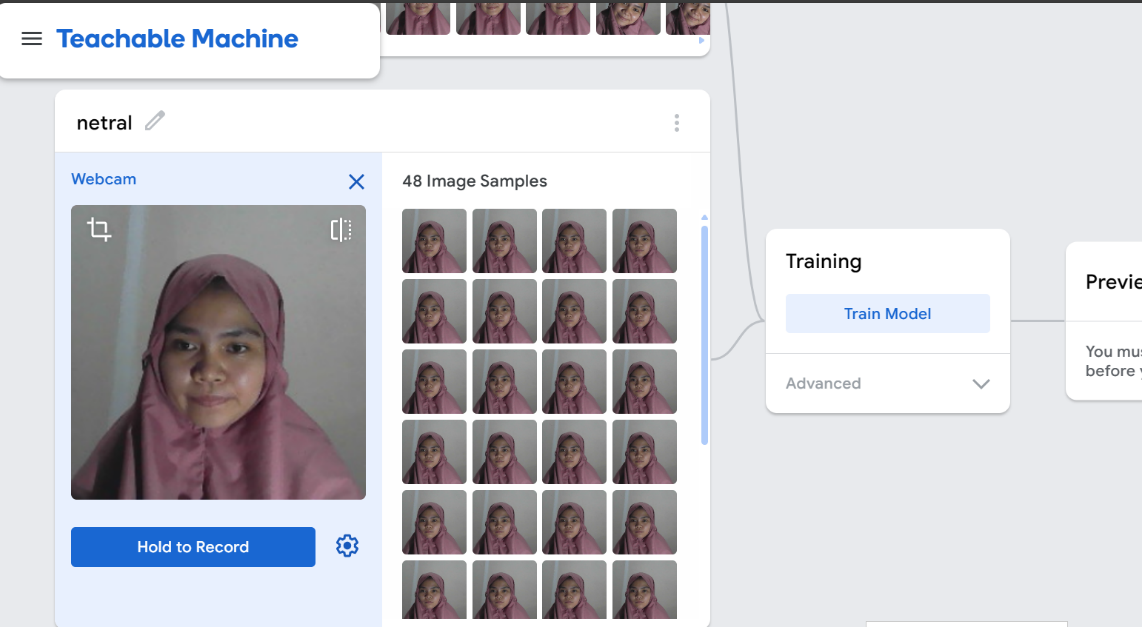
Gambar 11. dashboard software

1. Halaman *face detection*

Halaman face detection merupakan halaman setelah user melakukan input name. Selanjutnya user harus melakukan face detection secara jujur. Ini akan menampilkan tampilan yang berisikan record foto pergerakan dari si user yang akan menjadi dataset.

Gambar 12. Halaman face detection

1. Halaman Hasil *face recognation*

Setelah diproses oleh AI maka *software* ini akan menampilkan hasil rekognisi wajah *user* seperti perancangan dibawah ini. Sebagai contoh wajah *user* saatini dalam keadaan netral.

Gambar 13. Halaman hasil face recognation

## **Implementasidan *Unit Testing***

Untuk dipahami oleh komputer, desain yang telah dirancang sebelumnya harus diubah ke dalam format yang dapat dimengerti oleh bahasa mesin, yaitu dalam bentuk kode pemrograman. Tahap ini merupakan implementasi desain yang kemudian dilaksanakan secara teknis oleh programmer. Pada tahap ini, penulis mengembangkan software berdasarkan desain yang telah dibuat sebelumnya. Dalam pembuatan sofrware ini, digunakan bahasa pemrograman Dart, dengan pengembangan dilakukan dari awal hingga aplikasi siap dijalankan.

## ***Integration and Sytem Testing***

Pada tahap pengujian aplikasi yang telah dikembangkan, penulis menggunakan metode blackbox testing. Metode ini berfokus pada pengujian fungsionalitas perangkat lunak, terutama pada aspek masukan (input) sistem, untuk memastikan apakah semuanya sudah berfungsi sesuai dengan yang diharapkan. Metode blackbox testing menguji perangkat lunak tanpa memahami detail teknisnya. Penguji dapat melakukan tes fungsional, pengujian ulang, pengujian beban, dan pengujian keamanan. Metode ini menilai aplikasi dari perspektif pengguna, memastikan semua fitur berfungsi, dan menemukan masalah tanpa perlu memahami kode program.

## ***Operation and Mantainance***

Tahap pemeliharaan mencakup pembaruan dan pemeliharaan sistem untuk pengguna. Pemeliharaan aplikasi yang telah dikembangkan melibatkan perbaikan Corrective Maintenance untuk mengatasi bug dan kesalahan teknis, serta pemeliharaan adaptif untuk menyesuaikan aplikasi dengan perubahan kebutuhan sistem di masa mendatang melalui pemeliharaan dan pembaruan.