Rancang Bangun Aplikasi Pendeteksi Penyakit *Lumpy Skin Disease* Pada Sapi Berbasis *Android* Dengan Metode *Agile*

1st Muhammad Rifgi Alghifari
Direktorat Universitas Telkom Purwokerto
Universitas Telkom Purwokerto
Purwokerto, Indonesia
20102068@ittelkom-pwt.ac.id

2nd Wahyu Andi Saputra, S.Pd., M.Eng
Direktorat Universitas Telkom Purwokerto
Universitas Telkom Purwokerto
Purwokerto, Indonesia
andiwahyu@telkomuniversity.ac.id

***Abstrak* — Lumpy Skin Disease (LSD) merupakan penyakit baru yang menyerang sapi di Indonesia, ditandai dengan gejala bentol-bentol pada kulit. Kurangnya informasi tentang LSD menyebabkan penanganan yang terlambat, sehingga menimbulkan kerugian bagi peternak. Deteksi dini penyakit ini sangat penting untuk mencegah penyebaran virus secara luas. Untuk membantu peternak, dikembangkan sistem deteksi LSD berbasis Android yang mampu memindai gejala dan memberikan informasi lengkap tentang penyakit tersebut. Sistem ini dirancang menggunakan metode Agile, yang memungkinkan pengembangan fitur secara fleksibel dan berkelanjutan sesuai kebutuhan pengguna. Hasil pengujian aplikasi meliputi Blackbox Testing, Compatibility Test, dan Performance Efficiency Test menunjukkan bahwa aplikasi berfungsi dengan baik. Blackbox Testing mencatat 97,14% keberhasilan (105 dari 108 test case), Compatibility Test membuktikan aplikasi stabil pada Android dengan API Level 26 hingga 34, dan Performance Efficiency Test menunjukkan penggunaan CPU yang stabil serta penggunaan memori yang optimal tanpa memory leak. Dengan demikian, aplikasi ini diharapkan dapat menjadi solusi efektif untuk deteksi dini LSD dan mengurangi dampak kerugian pada peternak.**

***Kata kunci*— Android, *Lumpy Skin Disease*, Metode *Agile*, Rancang Bangun**

I. PENDAHULUAN

Sapi merupakan salah satu hewan yang sering dikonsumsi oleh masyarakat Indonesia sebagai pemenuhan kebutuhan protein. Tantangan yang sering kali dihadapi para peternak sapi dalam memelihara sapi yaitu kesehatan ternak yang disebabkan oleh penyakit dapat menurunkan produktivitas dan pertumbuhan ternak karena gangguan penyerapan nutrisi [1].

Salah satu penyakit yang menyerang peternak sapi di Indonesia pada tahun 2022 yaitu Lumpy Skin Disease (LSD), merupakan penyakit yang disebabkan oleh Lumpy Skin Disease Virus (LSDV). Virus tersebut menyebabkan salah satu gejala unik yaitu munculnya nodul-noodul pada bagian kulit dengan besar sekitar 2-7 cm. Secara keseluruhan penyakit ini memengaruhi produksi daging dan susu, kualitas kulit, daya tarik hewan dan efisiensi produksi (aborsi dan kemandulan) [2].

Karena penyakit Lumpy Skin Disease(LSD) ini terbilang cukup baru di Indonesia, permasalahan yang dihadapi peternak sapi salah satunya adalah kurangnya informasi mengenai LSD sehingga menyebabkan terlambatnya penanganan yang tepat bagi sapi yang terinfeksi. Oleh Karena itu, Deteksi awal penyakit dianggap sangat penting untuk menentukan kebijakan sehingga dapat menurunkan kemungkinan penyebaran virus secara luas [3]. Untuk membantu peternak mengambil tindakan yang cepat dan tepat maka, diperlukan sebuah pengembangan sistem yang dapat membantu peternak mendeteksi gejala awal pada penyakit LSD serta memberikan informasi mengenai cara pencegahan agar mengurangi penyebaran virus pada ternak sapi ke ternak sapi lainnya maupun ke daerah lain. Dengan menggunakan implementasi dari *machine learning* yang dapat mendeteksi gejala fisik dari penyakit LSD, bisa menjadi sistem penghubung antara dokter hewan dan peternak. Aplikasi yang akan dibangun memanfaatkan model *machine learning* yang telah diubah ke *tensorflow lite* agar bisa di implementasikan pada aplikasi android.

Dalam penelitian ini, pengembangan aplikasi deteksi pneyakit LSD pada sapi menggunakan metode *Agile*. *Agile Development* berupa iterasi atau perulangan pada setiap fasenya, tujuannya untuk merespon dan mengatasi setiap perubahan secara fleksibel, sehingga mengurangi waktu pengerjaan proyek dan mencapai kepuasan pelanggan. Fokus pada metode ini adalah pengerjaan aplikasi dengan meminimalisir dokumentasi, sehingga menghasilkan aplikasi dengan kualitas yang teruji [4].

Berdasarkan latar belakang tersebut, diangkatlah sebuah penelitian dengan topik pembangunan perangkat lunak dengan menggunakan metode *agile* yang berjudul "Rancang Bangun Aplikasi Pendeteksi Penyakit *Lumpy Skin Disease* pada Sapi Berbasis Android Dengan Metode *Agile*". Diharapkan dengan pelaksanaan penelitian ini dapat menghasilkan aplikasi yang dapat mendeteksi gejala awal penyakit LSD dan memberikan informasi tindakan yang benar apabila sapi terjangkit LSD agar dapat mengurangi kerugian yang diakibatkan penyakit LSD.

II. KAJIAN TEORI

a) Lumpy Skin Disease (LSD)

Lumpy Skin Disease (LSD) adalah penyakit virus pada sapi dan kerbau, yang disebabkan oleh *Lumpy Skin Disease virus* (LSDV) yang termasuk dalam genus *Capripoxvirus*, famili Poxviridae. Genus *Capripoxvirus* juga terdiri dari *goatpox virus* (GTPV) dan *sheepox virus* (SPPV). Penyakit ini termasuk virus lintas batas yang memiliki dampak ekonomi yang signifikan terhadap industri peternakan, termasuk kerugian finansial [5].

b) Rancang Bangun Aplikasi

Perancangan atau desain adalah proses menerjemahkan hasil analisis sebuah sistem ke dalam bahasa pemrograman untuk menjelaskan cara mengimplementasikan komponen-komponen sistem. Bangun atau Pengembangan adalah sebuah proses pembuatan sistem baru atau peningkatan seluruh atau sebagian sistem [6]. Oleh karena itu, rancang bangun dapat didefinisikan sebagai proses mengintegrasikan hasil analisis sistem ke dalam perangkat lunak, yang selanjutnya akan digunakan untuk membuat sistem secara keseluruhan atau untuk memperbaiki beberapa bagian darinya

c) Tensorflow lite

Tensorflow lite adalah kumpulan alat yang memungkinkan pembelajaran mesin di perangkat ponsel, perangkat tersemat, dan *edge devices*. *Tensorflow lite* dioptimalkan untuk menjalankan model di perangkat yang memiliki sumber daya komputasi dan memori yang terbatas, serta menawarkan peningkatan kinerja dengan akselerasi perangkat keras dan optimisasi model [7]. Prosesnya melibatkan pemilihan model, konversi model *TensorFlow* ke format *flat buffer* yang dikompresi menggunakan *Tensorflow lite Converter* [7]. *FlatBuffers* memiliki dua keuntungan utama yaitu dapat dipetakan dalam memori dan digunakan langsung dari *disk* tanpa dimuat dan diuraikan. Ini meningkatkan waktu *startup* dan memungkinkan sistem operasi untuk memuat dan membongkar halaman yang dibutuhkan dari *file model* untuk menghindari aplikasi dari kehabisan memori [8].

d) Metode *Agile*

Metodologi *agile* merupakan metodologi pengembangan perangkat lunak yang mempunyai prinsip yang sama ata sebuah metodologi untuk pengembangan suatu sistem dengan waktu yang pendek dan memerlukan adaptasi yang cepat dari pengembang terhadap perubahan dalam bentuk apapun [9]. Pendekatan *Agile* memungkinkan fleksibilitas terhadap perubahan yang terjadi selama siklus pengembangan. *Manifesto Agile* menekankan empat nilai utama dalam pengembangan perangkat lunak:

1. Mengutamakan individu dan interaksi daripada proses dan alat.
2. Menyediakan perangkat lunak yang berfungsi daripada dokumentasi yang lengkap.
3. Berkolaborasi dengan pelanggan daripada hanya bernegosiasi kontrak.
4. Merespons perubahan lebih penting daripada sekadar mengikuti rencana

e) Pengujian Perangkat Lunak

Pengujian Perangkat lunak merupakan proses eksekusi suatu program atau sistem dengan maksud menemukan atau, melibatkan setiap kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan suatu program atau sistem dan menentukan bahwa itu memenuhi hasil yang dibutuhkan Perusahaan [10]. Pengujian yang akan dilakukan menggunakan *blackbox testing*, *performance efficiency* dan *compatibility test*.

Blackbox testing merupakan salah satu pengujian perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian black box bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi [11]. Penelitian ini menggunakan standar pengujian metrik dengan rumus sebagai berikut:

$$Test\ Case\ Pass = \frac{Test\ Case\ Passed}{Total\ Test\ Case} \times 100 \quad (1)$$

$$Test\ Case\ \text{Failed} = \frac{Find\ Test\ Case\ Failed}{Total\ Test\ Case} \times 100 \quad (2)$$

Performance efficiency merupakan pengujian yang dilakukan untuk mengetahui kinerja dari sistem terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu, sedangkan *compatibility* merupakan pengujian non-fungsional yang dimaksudkan untuk memastikan bahwa perangkat lunak berjalan lancar pada di lingkungan yang berbeda-beda, kompatibel dalam perangkat dan versi android yang berbeda [12].

III. METODE

a) Subjek dan Objek Penelitian

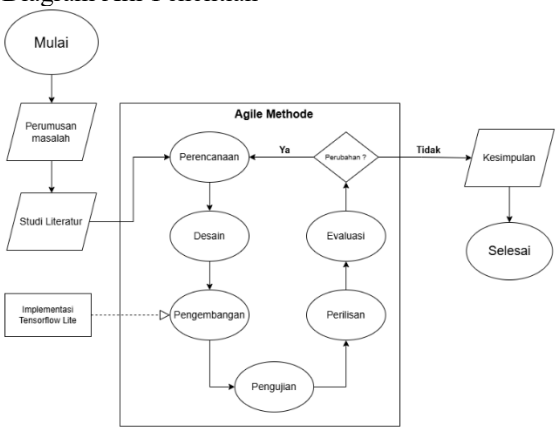
Subjek yang akan diteliti pada penelitian ini adalah peternak sapi dari Amanah Farm dan pakar dibidang aplikasi mobile. Sedangkan Objek yang diteliti adalah rancang bangun aplikasi pendeteksi penyakit *Lumpy Skin Disease* pada sapi berbasis android dengan metode Agile sebagai metode pengemabangan perangkat lunak.

b) Bahan Penelitian

Berikut merupakan bahan penelitian yang digunakan penelitian ini :

1. Studi Literatur
   Studi literatur dilakukan dengan mengumpulkan kerangka teoritis mengenai pengembangan perangkat lunak, aplikasi android, penyakit LSD sapi, dan pengujian aplikasi ini. Dengan berbagai referensi seperti buku, jurnal, artikel dan penelitian sebelumnya guna mendukung dan melandasi penelitian yang akan dilakukan. Dan juga dataset foto-foto penyakit LSD pada sapi dengan ciri-ciri berupa benjolan dan bintik-bintik pada kulit.
2. Wawancara
   Wawancara dilakukan dengan peternak sapi sebagai calon pengguna aplikasi. Dalam wawancara terebut menanyakan secara langung pengalaman sebagai peternak dan kendala yang dialami selama menjadi peternak sapi, sekaligus mengumpulkan kebutuhan sistem yang diperlukan aplikasi.

c) Diagram Alir Penelitian



GAMBAR 3.1
(DIAGRAM ALIR PENELITIAN)

1. Perumusan Masalah
   Merumuskan masalah yang akan diangkat untuk dijadikan pembahasan yang nantinya akan menghasilkan Solusi dari permsalahan tersebut.
2. Studi Literatur
   Pada tahap ini peneliti mencari kerangka teoritis yang akan menjadi acuan penelitian ini seperti pengembangan aplikasi, penyakit *Lumpy Skin Disease, agile methode,* pengujian *blackbox testing*.
3. Perencanaan
   Menyusun daftar fitur apa saja yang akan dibuat sesuai dengan hasil dari studi literatur dan wawancara.
4. Desain
   Membuat desain sistem dari daftar fitur yang telah dirumuskan ke dalam *Use Case Diagram, Activity Diagram,* dan *Sequence Diagram*
5. Pengembangan
   Pada tahap ini desain sistem yang telah dibuat akan di implementasikan ke dalam kodingan menggunakan bahasa Kotlin dengan IDE Android Studio. Selanjutnya akan mengintegrasikan model deteksi *tensorflow lite* ke dalam aplikasi android.
6. Pengujian
   Setelah dalam pengembangan aplikasi yang telah jadi akan diujikan agar berjalan sesuai dengan harapan. Pengujian yang dilakukan adalah *Blackbox testing, Compatibility Test* dan *Performance Efficiency.*
7. Perilisan
   Perilisan dapat dilakukan dengan menyimpan aplikasi kedalam penyimpanan seperti *hardisk*, *flasdisk*, penyimpanan *cloud*, ataupun rilis di *playstore*.
8. Evaluasi
   Pada tahap ini aplikasi akan di evaluasi untuk diidentifikasi kekurangan, permasalahan, dan efektifitasnya. Data tersebut dikumpulkan untuk keperluan pengembangan aplikasi kedepannya.
9. Kesimpulan
   Pada tahap ini akan diambil kesimpulan dari hasil penelitian yang nantinya akan menjawab rumusan masalah yang dibahas.

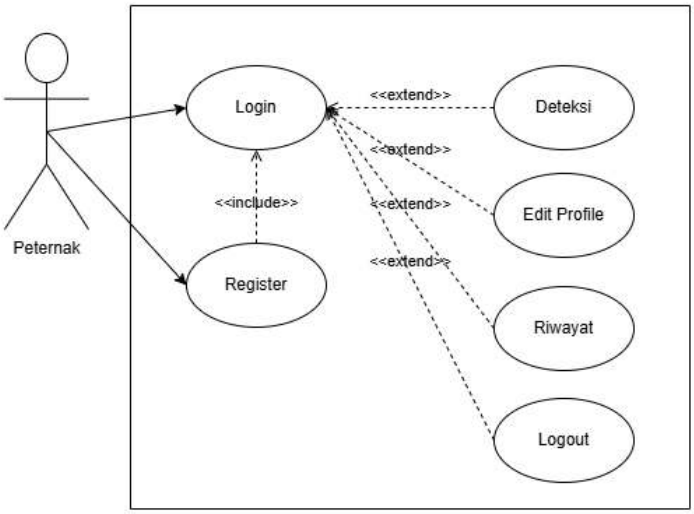
IV. HASIL DAN PEMBAHASAN

Berikut adalah hasil dari proses penerapan metode *agile* yang digunakan pada aplikasi deteksi penyakit ini.

a) Perencanaan
   Setelah melakukan pengumpulan data dengan studi literatur dan wawancara dengan peternak sapi di dapatkan kebutuhan sistem deteksi ini sebagai berikut :
   1. Sistem dapat melakukan deteksi gejala fisik pada penyakit LSD melalui gambar atau foto.
   2. Sistem dapat memberikan informasi tentang penyakit LSD.
   3. Sistem dapat memberikan informasi tentang perawatan dari penyakit LSD.
   4. Sistem dapat memberikan informasi penyebaran virus LSD dan bagaimana melakukan pertolongan pertama terhadap sapi yang terindikasi LSD.
   5. Sistem dapat menyimpan riwayat pemeriksaan sebelumnya

b) Desain
   Pada tahap ini akan membuat desain sistem dari perencanaan yang telah dirumuskan yang menjelaskan alur kerja sistem. Berikut adalah uraian dari desain sistem yang telah dibuat:

1. Use Case Diagram

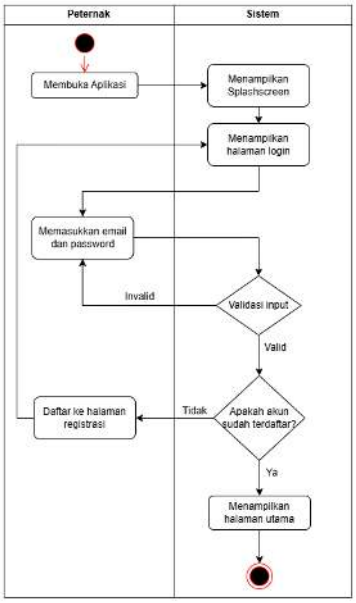


GAMBAR 4 1 (USE CASE DIAGRAM APLIKASI DETEKSI PENYAKIT LSD)

Pada Gambar 4.1 menjelaskan sistem akan memiliki satu aktor yaitu Peternak. Peternak harus Login sebelum ke mengakses aplikasi, apabila belum memiliki akun bisa membuatnya pada halaman register. Setelah login peternak dapat mengakses informasi yang berkaitan dengan penyakit *Lumpy Skin Disease*, dapat melakukan deteksi pada halaman deteksi dan dapat mengubah informasi akun pada halaman profil, serta dapat melihat riwayat deteksi sebelumnya yang telah tersimpan.

2. Activity Diagram

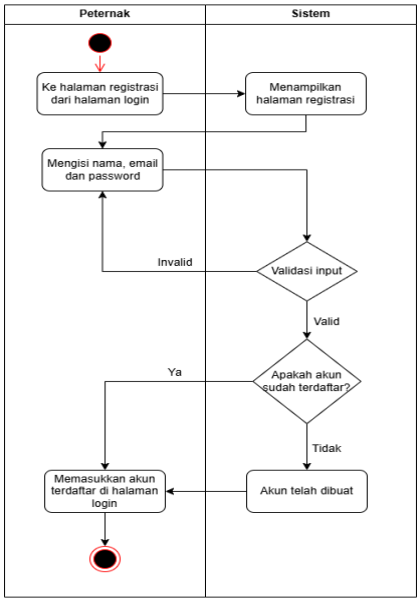
a. Activity Diagram Halaman Login



GAMBAR 4.2
(ACTIVITY DIAGRAM HALAMAN LOGIN)

Gambar 4.2 menampilkan activity diagram halaman Login. Saat aplikasi dibuka, peternak akan melihat splashscreen sebelum diarahkan ke halaman login jika belum pernah login. Di halaman login, peternak diminta memasukkan email dan password. Sistem akan memvalidasi inputan terlebih dahulu, kemudian memverifikasi email dan password. Jika data terdaftar, sistem akan mengarahkan peternak ke halaman utama.

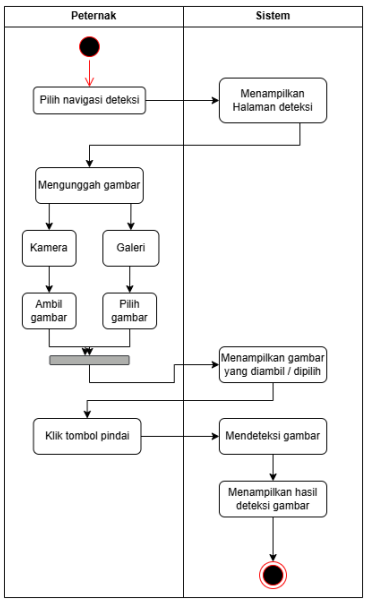
b. Activity Diagram Halaman Register



GAMBAR 4.3
(ACTIVITY DIAGRAM HALAMAN REGISTER)

Gambar 4.3 menunjukkan activity diagram halaman Register. Jika belum memiliki akun, peternak dapat membuatnya melalui halaman register yang diakses dari halaman login. Di halaman register, peternak diminta mengisi data seperti nama, email, dan password. Sistem akan memvalidasi inputan dan memverifikasi apakah email sudah terdaftar. Jika email belum terdaftar, akun berhasil dibuat, dan peternak dapat login menggunakan akun baru tersebut untuk masuk ke halaman utama.

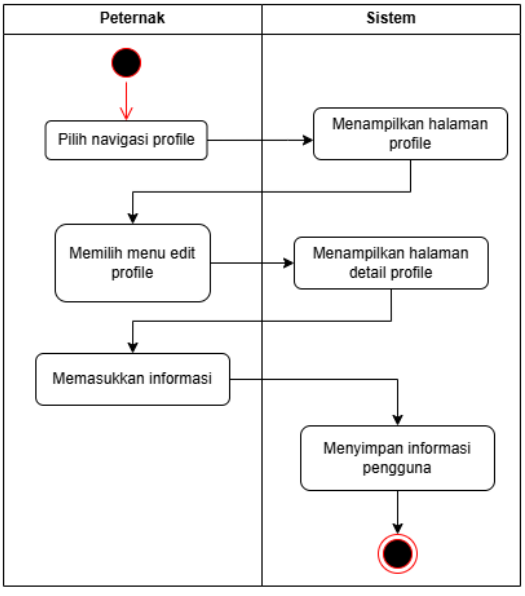
c. Activity Diagram Halaman Deteksi



GAMBAR 4.4
(ACTIVITY DIAGRAM HALAMAN DETEKSI)

Gambar 4.4 menampilkan activity diagram halaman Deteksi. Peternak dapat mendeteksi penyakit LSD dengan mengunggah gambar dari kamera atau galeri. Setelah gambar dipilih, sistem akan menampilkannya di halaman deteksi dan menganalisis apakah gambar tersebut menunjukkan sapi sehat atau sapi yang terinfeksi LSD.

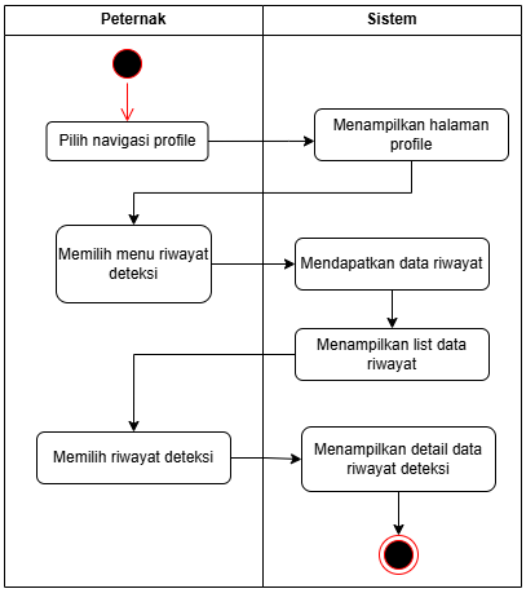
d. Activity Diagram Halaman Edit Profile



GAMBAR 4.5
(ACTIVITY DIAGRAM HALAMAN EDIT PROFILE)

Gambar 4.5 menampilkan *activity diagram* dari halaman *edit profile*. Peternak dapat merubah informasi akun seperti nama, pekerjaan, deskripsi dan foto *profile*. Lalu sistem akan menyimpan perubahan yang dilakukan peternak.

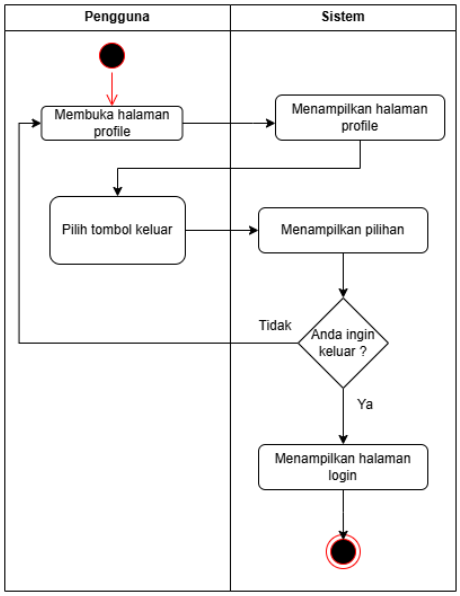
e. Activity Diagram Halaman Riwayat



GAMBAR 4.6
(ACTIVITY DIAGRAM HALAMAN RIWAYAT)

Gambar 4.6 menampilkan *activity diagram* dari halaman riwayat deteksi. Peternak dapat melihat hasil deteksi sebelumnya yang telah disimpan sistem pada halaman riwayat deteksi. Peternak juga bisa menghapus riwayat dan melihat *detail* dari data dari riwayat deteksi ketika memilih salah satu data riwayat.

f. Activity Diagram Logout



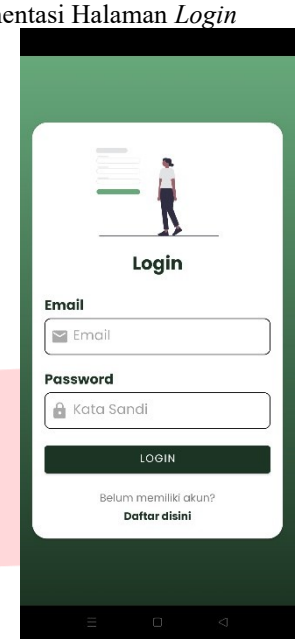
GAMBAR 4.7
(ACTIVITY DIAGRAM LOGOUT)

Gambar 4.7 menampilkan *activity diagram* yang menjelaskan proses peternak logout dari aplikasi. Peternak menuju ke halaman *profile* dimana ada tombol logout, dengan menekan tombol tersebut peternak akan di arahkan ke halaman login.

c) Pengembangan

Selanjutnya adalah tahap pengembangan dimana desain sistem yang telah dibuat akan diimplementasikan ke dalam pemrograman dengan bahasa kotlin. Berikut adalah hasil aplikasi yang sudah jadi.

1. Implementasi Halaman *Login*



GAMBAR 4.8
(IMPLEMENTASI HALAMAN LOGIN)

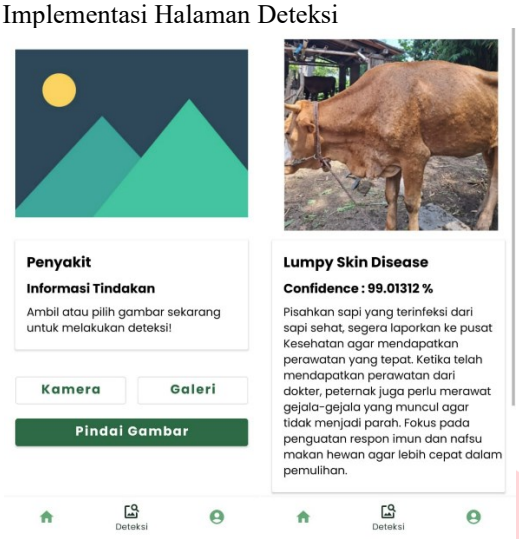
Gambar 4.8 menampilkan hasil implementasi dari halaman login dimana pengguna harus memasukkan email dan password yang telah terdaftar di database. Apabila pengguna belum memiliki akun bisa membuat di halaman register dengan mengklik “Daftar Disini”.

2. Implementasi Halaman *Register*

GAMBAR 4.9
(IMPLEMENTASI HALAMAN REGISTER)

Gambar 4.9 menampilkan hasil implementasi dari halam register. Pengguna perlu memasukkan nama, email dan password untuk mendaftarkan akun. Apabila sudah mendaftar pengguna bisa mengakses halaman login dengan mengklik “Masuk”.

3. Implementasi Halaman Deteksi



GAMBAR 4.10
(IMPLEMENTASI HALAMAN DETEKSI SEBELUM DAN SESUDAH)

Gambar 4.10 menampilkan halaman deteksi ketika sebelum melakukan deteksi dan sesudha melakukan deteksi. Pada halaman sebelum di deteksi akan menamilkan tombol untuk mengambil gambar dari kamera atau galeri. Lalu setelah halaman melakukan deteksi akan menampilkan hasil deteksi serta penjelasan tindakan yang perlu dilakukan. Pada halaman deteksi ini juga terdapat implementasi *tensorflow lite* yang ditunjukan pada gambar berikut.

```
var tensorImage = TensorImage(DataType.FLOAT32)
tensorImage.load(bitmap)

tensorImage = imageProcessor.process(tensorImage)

val model = EfficientNetB0.newInstance(safeContext)

// Creates inputs for reference.
val inputFeature0 = TensorBuffer.createFixedSize(intArrayOf(1, 224, 224, 3), DataType.FLOAT32)
inputFeature0.loadBuffer(tensorImage.buffer)

// Runs model inference and gets result.
val outputs = model.process(inputFeature0)
val outputFeature0 = outputs.outputFeature0AsTensorBuffer.floatArray

// Find the index with the maximum probability
var maxIdx = 0
for (i in outputFeature0.indices) {
    if (outputFeature0[i] > outputFeature0[maxIdx]) {
        maxIdx = i
    }
}
```

GAMBAR 4.11
(IMPELEMNTASI TENSORFLOW LITE PADA KODINGAN)

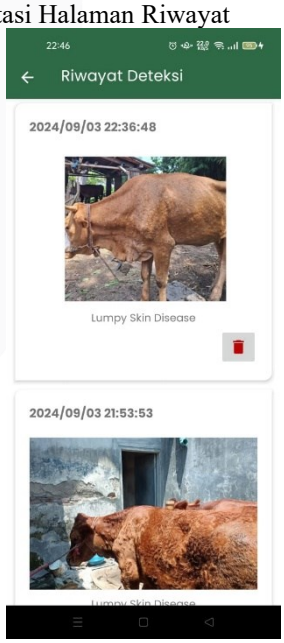
Gambar 4.11 menunjukkan implementasi Kode Model TensorFlow Lite yang dikembangkan oleh Cendana Harry Kristianto dan diubah ke format TensorFlow Lite untuk digunakan di perangkat Android. Proses integrasi ke Android melibatkan input berupa bitmap yang disesuaikan dengan aturan model. File model dengan ekstensi .tflite dipanggil, menghasilkan output berupa indeks dengan probabilitas tertinggi. Output juga mencakup confidence score (skor kepercayaan) yang menunjukkan tingkat akurasi deteksi, apakah gambar menunjukkan sapi sehat atau sapi terinfeksi LSD. Skor ini direpresentasikan dalam rentang 0 hingga 100 atau 0 hingga 1.

4. Implementasi Halaman *Edit Profile*

GAMBAR 4.12
(IMPLEMENTASI HALAMAN EDIT PROFILE)

Gambar 4.12 menunjukan hasil akhir implementasi dari halaman edit *profile*, dimana pengguna dapat meengubah informasi yang akan disimpan dan ditampilkan oleh sistem. Klik tombol “Ubah Foto *Profile*” untuk memunculkan galeri dan memilih gambar, lalu gambar yang dipilih akan di tampilkan pada halaman edit *profile*.

5. Implementasi Halaman Riwayat



GAMBAR 4.13
(IMPELENTASI HALAMAN RIWAYAT)

Pada Gambar 4.13 merupakan hasil akhir implementasi dari halaman utama riwayat deteksi dan detail riwayat deteksi. Halaman riwayat deteksi akan memunculkan *list* riwayat yang telah di deteksi sebelumnya dan tersimpan pada *database*.

6. Implementasi *Logout*

GAMBAR 4. 14
(IMPELEMNTASI LOGOUT)

Gambar 4.14 menunjukan hasil implementasi dari logout dimana tombol keluar terdapat pada halaman profile. Pengguna bisa keluar akun dengan menekan tombol keluar, lalu sistem akan memunculkan informasi apakah yakin keluar apabila iya akan diarahkan ke halaman login dan apabila tidak akan tetap berada pada halaman *profile*.

d) Pengujian

Tahap selanjutnya adalah pengujian dimana aplikasi akan diujikan terlebih dahulu agar mengetahui error-error yang dapat terjadi sebelum digunakan pengguna. Penelitian ini mengguna beberapa pengujian seperti *blackbox test, compatibility test dan performance efficiency test.* Berikut adalah urauian dari pengujian yang dilakukan.

1. Blackbox Test

Blackbox Testing digunakan untuk menguji fungsionalitas sistem yang berfokus pada input dan output dari perangkat lunak serta perilaku sistem terhadap berbagai skenario uji, sesuai dengan spesifikasi yang telah di tentukan. Berikut adalah data penguji aplikasi yaitu tiga orang dosen Telkom University dan spesifikasi dari blackbox testing pada aplikasi deteksi ini:

TABEL 4.1
(PENGUJI APLIKASI DAN DEVICE)

Nama Penguji	Device
Arif Amrullah, S.Kom., M.Kom	Samsung A15
Novian Adi Prasetyo, S.Kom., M.Kom	Xiaomi Redmi Note 11
Alon Jala Tirta Segara, S.Kom., M.Kom	Realme

Berdasarkan hasil pengujian fitur-fitur yang telah dilakukan, kesimpulan hasil ditunjukan pada Tabel 4.2. Total fitur yang diujikan adalah 8 fitur dengan total 36 *test case* yang telah diujikan kepada tiga penguji dosen.

TABEL 4.2
(HASIL PENGUJIAN BLACKBOX)

Fitur	Test Case	Berhasil	Gagal
Login	15	13	2
Register	12	12	0
Beranda/Home	18	18	0
Deteksi	12	12	0
Kamera	15	15	0
Profile	12	12	0
Edit *Profile*	12	11	1
Riwayat	12	12	0
Total	108	105	3

2. Compatibility Test

Compatibility merupakan pengujian android yang bertujuan untuk memastikan aplikasi dapat berjalan baik pada berbagai perangkat android yang memiliki ekosistem yang berbeda. Berikut adalah hasil dari pengujian *compatibility test* aplikasi dengan menggunakan *firebase test lab* :

TABEL 4.3
(HASIL PENGUJIAN PADA API YANG BERBEDA)

API	Hasil
Level 26	Passed
Level 27	Passed
Level 28	Passed
Level 29	Passed
Level 30	Passed
Level 31	Passed
Level 32	Passed
Level 33	Passed
Level 34	Passed

Berdasarkan hasil pengujian compatibility test yang dilakukan secara otomatis dengan *firebase lab*, didapatkan hasil aplikasi deteksi penyakit *Lumpy Skin Disease* ini dapat berjalan pada android dengan *API Level 26* atau *codename Oreo* sampai dengan *Android API Level 34* atau *codename Upside Down Cake*.

3. Performance Efficiency

Performance Efficiency test merupakan pengujian aplikasi untuk mengetahui kinerja dari sistem terhadap sumber daya yang digunakan, aspek yang diamati pengujian ini adalah penggunaan memori dan cpu pada saat sistem berjalan. Pengujian berfokus pada device saat sedang menjalankan kamera, mengambil gambar, dan melakukan deteksi gambar dengan menjalankan *model tensorflow lite*. Pengujian akan dilakukan secara otomatis dengan *firebase lab*, dengan device Galaxy S9 dengan Level API 26 dengan waktu pengujian selama 2 menit 39 detik, melakukan 2 kali pengambilan gambar dengan kamera dan melakukan deteksi dengan dua gambar yang diambil. Hasil dari penggunaan memori dan cpu pada setiap device yang diujikan ditunjukan pada penjelasan berikut ini:

a) CPU

Penggunaan CPU ditampilkan pada gambar grafik berikut:

GAMBAR 4. 15
(GRAFIK PENGGUNAAN CPU)

Pada grafik CPU, sumbu x (horizontal) menunjukkan waktu pengujian, sedangkan sumbu y (vertikal) menunjukkan persentase penggunaan CPU (%). Garis naik turun menggambarkan total penggunaan CPU dalam rentang waktu tertentu. Saat aplikasi membuka kamera pada 00:49, penggunaan CPU naik 15%. Pada 00:59, saat kamera mengambil gambar, CPU meningkat 19%. Pada 01:05, aplikasi memindai gambar dan menampilkan hasil dengan penggunaan CPU 13%. Selanjutnya, pada 01:36, aplikasi mengambil gambar kedua dengan CPU naik 20%, dan pada 01:43, memindai gambar serta menampilkan hasil dengan penggunaan CPU 14%.

b) Memori

Penggunaan Memori ditampilkan pada gambar grafik berikut:

GAMBAR 4. 16
(GRAFIK PENGGUNAAN MEMORI)

Pada grafik, sumbu x (horizontal) menunjukkan waktu pengujian, sedangkan sumbu y (vertikal) menampilkan jumlah memori dalam satuan KiB. Garis naik turun menggambarkan total penggunaan memori dalam rentang waktu tertentu. Saat aplikasi membuka kamera pada 00:49, penggunaan memori stabil di 340.000 KiB. Pada 00:59, saat mengambil gambar, memori naik menjadi 470.000 KiB. Pada 01:07, saat gambar dipindai dan hasil ditampilkan, memori meningkat lagi menjadi 505.000 KiB. Saat menggunakan kamera kedua pada 01:27, memori turun ke 280.000 KiB. Pengambilan gambar kedua menyebabkan kenaikan memori menjadi 372.000 KiB, dan saat gambar dipindai, memori naik stabil ke 403.000 KiB.

Berdasarkan hasil grafik penggunaan cpu dan memori pada aplikasi deteksi ini di saat menggunakan kamera, mengambil gambar dan memindai gambar terdapat kenaikan dari penggunaan cpu dan kamera karena aplikasi sedang melakukan operasi yang cukup besar, tetapi kaenaikan tetapi stabil tidak tiba-tiba naik atau tiba-tiba turun. Walaupun penggunaan memori cukup besar dengan penggunaan maksimal sebesar 505.000 KiB tetapi aplikasi tidak menyebabkan *memory leak* yang dapat membuat aplikasi *crash*.

V. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan, pengembangan aplikasi pendeteksi penyakit *Lumpy Skin Disease* (LSD) telah berhasil dibangun menggunakan metode pengembangan *Agile* dan menghasilkan alikasi yang dapat mendeteksi sekaligus memberikan informasi terkait dengan penyakit *lumpy skin disease.*

Hasil pengujian *blackbox test* aplikasi pendeteksi penyakit LSD ini menunjukan performa yang cukup baik dengan tingkat keberhasilan 97,2% dengan total *test case* berhasil sebesar 105 *test case*. Hasil pengujian *compatibility test* menunjukan aplikasi dapat berjalan baik pada *device* dengan *API Level* 26 sampai dengan *API Level* 34. Untuk pengujian *performance efficiency* penggunaan cpu cukup stabil dan tidak memuat aplikasi *crash*, sedangkan penggunaan memori walaupun terbilang tinggi tetapi tidak menyebabkan *memory leak* yang dapat membuat aplikasi *crash*.

REFERENSI

[1] D. M. Nuraini, N. W. Sunarto, A. Pramono and S. Prastowo, "Peningkatan Kapasitas Tata Laksana Kesehatan Ternak Sapi Potong di Pelemrejo, Andong, Boyolali," *PRIMA: Journal of Community Empowering and Services,* vol. Vol 4 (2), no. 102, pp. 102-108, 2020.

[2] T. G. Chahota, V. Patial, D. Bali, S. Angaria, M. Sharma and Rajesh, "A review: Lumpy skin disease and its emergence in India," *Veterinary Research Communications,* p. 111–118, 2020.

[3] N. L. P. I. Dharmayanti and D. Nurjanah, "Ulasan Lumpy Skin Disease: Penyakit Infeksius Berpotensi Mengancam Kesehatan Sapi Di Indonesia," *Berita Biologi,* vol. 20, no. 3, pp. -, 2020.

[4] N. Hikmah, A. Suradika and R. A. A. Gunadi, "Metode Agile Untuk Meningkatkan Kreativitas Guru Melalui Berbagi Pengetahuan (Knowledge Sharing)," *Jurnal Instruksional,* vol. 3, no. 1, pp. -, 2021.

[5] P. Koirala, I. Meki, M. Maharjan, B. Settypalli, S. Manandhar, S. Yadav, G. Cattoli and C. Lamien, "Molecular Characterization of the 2020 Outbreak of Lumpy Skin Disease in Nepal," *Microorganisms,* vol. 10, no. -, p. 539, 2022.

[6] R. S. Pressman, Software Engineering A Practitioner's Approach 7th Ed, New York: McGraw-Hill, 2009.

[7] C. C. A. 4. License, "TensorFlow Lite for Android," Tensorflow, 3 September 2022. [Online]. Available: https://www.tensorflow.org/lite/android. [Accessed 20 November 2023].

[8] B. a. N. I. Syamsuri, "Plant Disease Classification using Lite Pretrained Deep Convolutional Neural

Network on Android Mobile Device," *International Journal of Innovative Technology and Exploring Engineering* , Vols. -, no. -, pp. -, 2019.

[9] H. R. Suharno, N. Gunantara and M. Sudarma, "Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri & Organisasi Digital," *Majalah Ilmiah Teknologi Elektro,* vol. 19, no. 2, pp. 203-210, 2020.

[10] W. E. Perry, A Standard for Testing Application Software, -: Auerbach Publishers, 1990.

[11] F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," *Jurnal Informatika Universitas Pamulang,* vol. 4, no. 4, pp. 125-130, 2019.

[12] ISO 25000 software and data quality, "ISO/IEC 25010," ISO25000, - - -. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010/. [Accessed 2 Desember 2024].