



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 1 (2026) pp: 13328-13337

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Penerapan Metode Forward Chaining dalam Sistem Pakar untuk Diagnosis Penyakit Kulit pada Kucing Berbasis Android

M. Nur Rois Abid, Tri Susilowati

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Cendekia Abditama

nurrois@uca.ac.id, susilawatitri13@gmail.com*

Abstrak

Penyakit kulit pada kucing merupakan salah satu permasalahan kesehatan yang sering terjadi dan kerap membingungkan pemilik hewan peliharaan akibat keterbatasan pengetahuan dalam mengenali gejala serta menentukan penanganan yang tepat. Kondisi ini dapat menyebabkan keterlambatan diagnosis yang berpotensi memperburuk kondisi kesehatan kucing. Permasalahan utama yang sering dihadapi pemilik kucing adalah kesulitan dalam mengenali jenis penyakit kulit dan menentukan langkah penanganan yang tepat. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah aplikasi sistem pakar berbasis Android yang mampu membantu proses diagnosis awal penyakit kulit pada kucing secara mandiri oleh pengguna. Sistem yang dikembangkan menggunakan metode forward chaining sebagai teknik penalaran, yaitu dengan menelusuri fakta berupa gejala yang diinputkan pengguna untuk menghasilkan kesimpulan berupa jenis penyakit yang kemungkinan diderita. Aplikasi ini dirancang dalam bentuk interaktif melalui serangkaian pertanyaan yang disusun berdasarkan gejala klinis yang umum ditemukan pada penyakit kulit kucing. Hasil dari sistem ini berupa hasil diagnosis awal serta informasi penanganan yang dapat dijadikan panduan oleh pemilik kucing. Dengan adanya aplikasi ini, diharapkan dapat meningkatkan kemudahan akses informasi, mempercepat proses identifikasi penyakit, serta membantu pemilik dalam mengambil langkah penanganan awal secara tepat. Selain itu, sistem ini juga diharapkan dapat meminimalkan kesalahan diagnosis awal dan meningkatkan kesadaran masyarakat terhadap pentingnya menjaga kesehatan hewan peliharaan.

Kata kunci: Diagnosa Penyakit Kucing, Penyakit Kulit Kucing, Sistem Pakar, Android, Forward Chaining.

1. Pendahuluan

Kucing merupakan salah satu hewan peliharaan yang banyak diminati masyarakat karena memiliki sifat yang jinak, lucu, dan mudah beradaptasi dengan manusia. Berbagai jenis kucing seperti kucing ras (Angora, Persia) maupun kucing domestik banyak dijadikan hewan kesayangan. Namun, dalam pemeliharaan kucing, kesehatan merupakan faktor penting yang harus diperhatikan agar terhindar dari berbagai penyakit. Salah satu penyakit yang sering dijumpai adalah penyakit kulit. Penyakit kulit pada kucing dapat menyebabkan rasa gatal, kerontokan bulu, dan infeksi serius apabila tidak segera ditangani [1].

Permasalahan utama yang sering dihadapi pemilik kucing adalah kesulitan dalam mengenali jenis penyakit kulit dan menentukan langkah penanganan yang tepat. Hal ini disebabkan oleh kurangnya pengetahuan mengenai gejala dan jenis penyakit kulit pada kucing. Akibatnya, banyak kasus penyakit kulit yang terlambat ditangani sehingga menurunkan kualitas hidup hewan peliharaan. Untuk mengatasi hal tersebut, diperlukan sistem yang mampu membantu pemilik kucing dalam melakukan diagnosis awal penyakit kulit dengan cepat dan akurat.

Pemanfaatan teknologi informasi, khususnya dalam bentuk sistem pakar berbasis *Android*, dapat menjadi solusi yang efektif untuk membantu pemilik kucing melakukan diagnosis mandiri. Sistem pakar ini menggunakan metode *forward chaining*, yaitu metode penalaran yang bekerja dengan menelusuri fakta atau gejala yang diketahui untuk memperoleh kesimpulan diagnosis. Dengan pendekatan ini, sistem akan menanyakan serangkaian pertanyaan berdasarkan gejala yang diamati, kemudian memberikan hasil diagnosis serta rekomendasi penanganan yang sesuai. Melalui pengembangan sistem pakar berbasis *Android*, pengguna dapat dengan mudah mengakses informasi diagnosis penyakit kulit pada kucing kapan pun dan di mana pun, sehingga mampu meningkatkan efisiensi serta kemudahan dalam memperoleh informasi, memberikan pengalaman penggunaan yang lebih praktis dan interaktif, serta membantu pengguna dalam memahami hasil diagnosis secara lebih cepat dan tepat [2]. Sistem

ini diharapkan dapat membantu pemilik kucing dalam mengenali gejala penyakit sejak dini, memberikan saran penanganan awal, serta meningkatkan kesadaran masyarakat terhadap pentingnya kesehatan hewan peliharaan.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pakar diagnosis penyakit kulit pada kucing berbasis *Android* menggunakan metode *forward chaining*. Sistem ini diharapkan dapat menjadi alat bantu yang efektif bagi pemilik kucing dalam melakukan diagnosis awal secara mandiri serta mendukung penerapan kecerdasan buatan dalam bidang kesehatan hewan. Selain itu, pengembangan sistem ini juga diharapkan dapat meningkatkan efisiensi waktu dalam proses identifikasi penyakit meminimalkan kesalahan diagnosis.

Salah satu penelitian yang relevan dilakukan pada tahun 2020 oleh Aditya Sugih Pangestu dan Rinabi Tanamal dengan judul Rancang Bangun Aplikasi Sistem Pakar Berbasis Mobile untuk Mendiagnosis Penyakit Kulit pada Kucing Persia, dengan fokus utama mengembangkan sistem pakar diagnosis penyakit kulit pada kucing Persia dengan memanfaatkan hasil wawancara dan observasi pakar yang diolah menjadi pohon keputusan. Sistem ini dibangun menggunakan *Expert System Builder (McGoo)* dan *Thunkable*, serta menerapkan metode *forward chaining* sebagai dasar penalaran untuk menghasilkan diagnosis secara cepat dan akurat [3]. Penelitian yang dilakukan pada tahun 2023 oleh Bayu Candra Saputra dengan judul Perancangan Sistem Pakar Diagnosa Penyakit pada Hewan Peliharaan Menggunakan Metode Forward Chaining ini berfokus membantu pemilik hewan memahami kondisi peliharaannya melalui sistem pakar yang menyediakan informasi dan diagnosis penyakit berdasarkan gejala yang diinputkan. Sistem pakar berbasis web ini menggunakan metode *forward chaining* untuk mendiagnosis penyakit pada kucing dan anjing, mencakup 22 penyakit dengan 110 gejala untuk kucing serta 21 penyakit dengan 105 gejala untuk anjing. Sistem dikembangkan menggunakan *PHP, HTML5, CSS, MySQL*, dengan *flowchart* dan *Certainty Factor* sebagai dasar perancangannya [4].

Berdasarkan hasil wawancara dengan dr. Muhammad Fadli Shalih dan studi pustaka, terdapat tiga jenis penyakit kulit yang umum dialami kucing, yaitu *Otodectes cynotis*, *Abses*, dan *Scabies*. *Otodectes cynotis* disebabkan oleh tungau yang menginfeksi saluran telinga dan menyebabkan gatal serta peradangan. *Abses* merupakan infeksi bakteri yang menimbulkan benjolan berisi nanah di bawah kulit, umumnya pada bagian kepala, kaki, pinggang, atau ekor. Sedangkan *Scabies* disebabkan oleh tungau *Sarcoptes scabiei* atau *Notoedrescati* yang menyerang lapisan kulit luar dan menimbulkan rasa gatal, ruam merah, serta kerontokan bulu pada kucing.

Sistem pakar merupakan salah satu cabang dari kecerdasan buatan yang bekerja dengan menggabungkan pengetahuan dan pencarian dalam basis pengetahuan untuk memecahkan suatu permasalahan. Sistem ini dirancang menyerupai kemampuan seorang pakar sehingga dapat digunakan oleh banyak orang dalam pengambilan keputusan. Secara umum, sistem pakar memiliki beberapa karakteristik utama, antara lain memiliki kinerja tinggi dengan hasil yang setara atau lebih baik dari seorang ahli, waktu respon yang cepat, tingkat keandalan yang tinggi, serta kemampuan untuk menjelaskan proses penalarannya agar mudah dipahami pengguna. Adapun manfaat sistem pakar antara lain digunakan dalam berbagai bidang, seperti sistem untuk membantu pengambilan keputusan berdasarkan hasil observasi, sistem prediksi untuk memperkirakan kemungkinan kejadian di masa depan, sistem diagnosis untuk menentukan penyebab gangguan berdasarkan gejala yang muncul, sistem desain untuk merancang konfigurasi komponen sesuai kebutuhan, serta sistem perencanaan yang membantu menyusun langkah-langkah strategis untuk mencapai tujuan tertentu [5].

Metode *forward chaining* merupakan strategi inferensi yang bekerja dengan memulai penalaran dari sejumlah fakta yang telah diketahui. Proses pencarian dilakukan dengan menggunakan aturan yang premisnya sesuai dengan fakta yang ada untuk menghasilkan fakta baru. Proses ini berlangsung secara berurutan hingga mencapai suatu kesimpulan atau ketika tidak ada lagi aturan yang sesuai dengan fakta yang diketahui maupun fakta yang diperoleh selama penelusuran [6].

Awalnya, *Android* dikembangkan oleh *Android Inc.*, sebuah perusahaan rintisan yang fokus pada perangkat lunak ponsel. Sistem ini kemudian diakuisisi oleh *Google*. Untuk mendukung pengembangannya, dibentuklah *Open Handset Alliance (OHA)*, sebuah konsorsium yang terdiri dari 34 perusahaan ternama di bidang perangkat keras, perangkat lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile*, dan *Nvidia* [7].

Android Studio pertama kali dikenalkan dalam versi 0.1 pada Mei 2013. Kemudian, versi beta 0.8 dirilis pada Juni 2014, disusul oleh peluncuran versi stabil pertamanya, yaitu versi 1.0, yang diluncurkan pada Desember 2014. *Android Studio* dikembangkan berdasarkan *IntelliJ IDEA* dari *JetBrains* dan dirancang khusus untuk

pengembangan aplikasi *Android* [8].

Kotlin merupakan bahasa pemrograman modern yang bersifat pragmatis dan dirancang untuk pengembangan aplikasi *Android* dengan mengombinasikan paradigma *object-oriented* dan fungsional. Bahasa ini memiliki tingkat

interoperabilitas tinggi, sehingga dapat digunakan bersamaan dengan *Java* dalam satu proyek. Selain untuk *Android*, *Kotlin* juga mendukung pengembangan aplikasi berbasis *desktop*, *web*, dan *backend*. Bahasa ini dikembangkan oleh *JetBrains* dan dirilis secara open source. Saat ini, *Google* memberikan dukungan penuh terhadap *Kotlin* sebagai bahasa utama dalam pengembangan aplikasi *Android*. Kelebihan *Kotlin* antara lain bersifat ekspresif dan ringkas, dapat digunakan bersama *Java*, serta memiliki struktur kode yang terorganisasi dengan baik [9].

Java merupakan bahasa pemrograman multiplatform yang dapat dijalankan di berbagai sistem operasi tanpa bergantung pada perangkat keras tertentu. Bahasa ini tidak memiliki *Integrated Development Environment (IDE)* khusus, namun didukung oleh berbagai *IDE* seperti *NetBeans*, *Eclipse*, dan *IntelliJ IDEA*. *Java* memiliki elemen dasar pemrograman berupa himpunan karakter, pengenalan, kata kunci, serta tipe data *primitif* seperti *byte*, *short*, *int*, *long*, *float*, *double*, *boolean*, dan *char*. Dengan sifatnya yang stabil, portabel, dan aman, *Java* banyak digunakan untuk pengembangan aplikasi berskala besar, termasuk aplikasi *Android* [10].

PHP (Hypertext) merupakan bahasa pemrograman *server-side* yang digunakan untuk memproses dan menerjemahkan kode program agar dapat dijalankan oleh komputer. Bahasa ini umumnya dikombinasikan dengan *HTML* untuk membangun situs *web dinamis* yang mampu mengelola *database* dan konten secara interaktif. Dengan kemampuannya dalam memproses data di sisi *server*, *PHP* banyak digunakan dalam pengembangan aplikasi berbasis *web* karena *fleksibel*, mudah dipelajari, dan memiliki dukungan luas dari berbagai *platform* serta *database* [11].

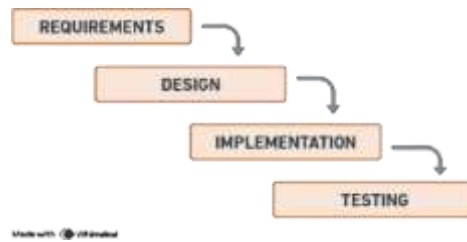
Jetpack Compose merupakan *framework* modern untuk pengembangan antarmuka pengguna (*user interface*) pada aplikasi *android*. *Framework* ini memanfaatkan komponen yang dapat digunakan kembali (*reusable components*) serta mendukung penerapan tema gelap, animasi, dan desain responsif secara efisien. *Jetpack Compose* dibangun di atas arsitektur *Android Jetpack*, sehingga tetap mendukung seluruh fungsionalitas *android* asli. *Jetpack Compose* juga mendukung berbagai perangkat lain seperti tablet, perangkat lipat (*foldables*), *ChromeOS*, dan *Wear OS*, memungkinkan pengembang membuat tampilan aplikasi yang konsisten dan optimal di berbagai ukuran serta jenis layar perangkat [12].

2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif, merupakan Pendekatan ini berfokus pada pengumpulan dan analisis data numerik untuk memahami fenomena tertentu. Metode ini diterapkan untuk mengumpulkan, menganalisis, dan menginterpretasikan data sehingga menghasilkan informasi yang akurat.

Pengumpulan data dalam penelitian ini dilakukan melalui tiga tahapan utama, yaitu wawancara, *observasi*, dan studi pustaka. Wawancara dilakukan secara langsung dengan dr. Muhammad Fadli Shalih di Klinik Hewan *City Petcare* untuk memperoleh informasi mengenai jenis penyakit kulit yang sering dialami oleh kucing, gejala yang muncul, serta langkah penanganan awal yang tepat. *Observasi* dilakukan dengan mengamati secara langsung proses pemeriksaan dan perawatan kucing yang mengalami penyakit kulit di klinik tersebut untuk mendapatkan data yang akurat dan sesuai dengan kondisi di lapangan. Sementara itu, studi pustaka dilakukan dengan mengumpulkan berbagai sumber referensi seperti jurnal ilmiah, artikel, dan buku yang relevan dengan topik penelitian guna memperkuat dasar teori dan mendukung proses pengembangan sistem pakar diagnosis penyakit kulit pada kucing berbasis *Android*.

Pada pengembangan sistem ini, metode yang digunakan adalah metode *waterfall*, Metode *waterfall* merupakan pendekatan *SDLC (Software Development Life Cycle)* yang merupakan tahap paling awal yang digunakan untuk pengembangan perangkat lunak, model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara terurut dimulai dari analisis, desain, pengkodean, dan pengujian. [13]



Gambar 1. Metode *Waterfall*

Requirements, pada tahap ini dilakukan analisis terhadap kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi fitur utama seperti halaman home untuk input data kucing, halaman *dashboard* untuk memulai proses diagnosis, halaman diagnosis untuk pemilihan gejala, serta halaman hasil diagnosis yang menampilkan hasil analisis dan solusi penanganan penyakit kulit pada kucing. Sementara itu, kebutuhan non-fungsional mencakup spesifikasi perangkat keras dan perangkat lunak yang digunakan, antara lain *prosesor Intel Celeron N4020*, RAM 4 GB, dan penyimpanan 118 GB, dengan dukungan *Android Studio* sebagai lingkungan pengembangan, bahasa pemrograman *Kotlin*, serta *framework Jetpack Compose* untuk membangun antarmuka aplikasi yang interaktif dan responsif.

Design, Tahapan ini meliputi proses perancangan sistem menggunakan *United Modeling Language (UML)* yang berfungsi untuk menggambarkan struktur dan alur kerja aplikasi. Salah satu diagram yang digunakan pada tahap ini adalah *use case diagram*. Pada gambar 2, Alur kerja sistem dimulai ketika pengguna masuk ke aplikasi dan memasukkan data berupa nama serta umur kucing. Setelah itu, pengguna diarahkan ke halaman dashboard untuk memulai proses diagnosis dengan memilih gejala yang dialami oleh kucing. Hasil diagnosis kemudian ditampilkan pada halaman hasil diagnosis, yang berisi informasi mengenai jenis penyakit kulit beserta solusi penanganannya. Selain pengguna, *developer* juga memiliki akses terhadap sistem untuk melakukan pembaruan, perbaikan, dan pemeliharaan secara berkala agar kinerja aplikasi tetap optimal.



Gambar 2. *Use Case Diagram*

Implementasi, pada tahap implementasi ini adalah proses penerapan rancangan sistem ke dalam bentuk aplikasi yang siap dijalankan dan diuji secara fungsional. Pada tahap ini, setiap komponen yang telah dirancang diimplementasikan menggunakan *Android Studio* dengan bahasa pemrograman *Kotlin* serta *framework Jetpack Compose* untuk membangun antarmuka pengguna yang interaktif. Selanjutnya, dilakukan pengujian unit guna memastikan bahwa setiap fungsi dalam aplikasi, seperti input data, pemilihan gejala, proses diagnosis, dan tampilan hasil, berjalan sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini bertujuan untuk mendeteksi kesalahan sedini mungkin agar sistem dapat berfungsi dengan baik sebelum masuk ke tahap pengujian lanjutan atau integrasi penuh.

Testing, Pada tahap ini dilakukan proses pengujian aplikasi menggunakan metode *blackbox testing*. Metode ini berfokus pada pengujian fungsi perangkat lunak berdasarkan spesifikasi yang telah ditentukan tanpa memeriksa struktur internal atau kode program. Pengujian dilakukan dengan memberikan berbagai skenario masukan untuk memastikan bahwa setiap fitur, proses, dan keluaran aplikasi berjalan sesuai kebutuhan dan spesifikasi fungsional yang telah ditetapkan. Melalui tahap ini, dapat diketahui apakah sistem mampu merespons masukan pengguna dengan benar, menampilkan hasil diagnosis sesuai alur, serta memastikan tidak terjadi kesalahan proses yang dapat mengganggu penggunaan aplikasi.

Perancangan sistem diawali dengan identifikasi gejala penyakit kulit pada kucing yang akan digunakan sebagai dasar penentuan aturan dalam sistem. Data gejala diperoleh melalui hasil wawancara dengan dr. Muhammad Fadli Shalih di Klinik Hewan *City Petcare*, observasi langsung terhadap kasus penyakit kulit, serta studi pustaka dari jurnal dan literatur terkait. Setelah data terkumpul, dilakukan analisis untuk menentukan gejala yang paling umum dan signifikan sehingga dapat dijadikan acuan dalam pembentukan aturan sistem pakar. Selanjutnya disusun basis pengetahuan yang berisi kumpulan data penyakit, gejala, serta aturan dalam bentuk *if-then* sebagai representasi hubungan antara gejala dan diagnosis penyakit. Basis pengetahuan ini menjadi komponen utama dalam proses pengambilan keputusan sistem.

Tabel 1. Data Penyakit

No	Kode Penyakit	Nama Penyakit
1	P01	Otodectes Cynotis
2	P02	Scabies
3	P03	Abses

Tabel 2. Data Gejala

No	Kode Penyakit	Nama Penyakit
1	G01	Telinga Berkerak
2	G02	Sering Menggaruk Telinga
3	G03	Telinga Berbau
4	G04	Bulu Rontok
5	G05	Kulit Bersisik
6	G06	Kulit Berkerak
7	G07	Sering Menggaruk Badan
8	G08	Bengkak di Kulit
9	G09	Luka Bulat Merah
10	G10	Lemas
11	G11	Kehilangan Nafsu Makan

Tabel 3. Data Gejala

No	Kode	Aturan
1	R01	IF (G01) Telinga berkerak AND (G02) sering menggaruk telinga AND (G03) Telinga berbau AND (G05) Kulit bersisik THEN (P01) Otodectes Cynotis
2	R03	IF (G04) Bulu rontok AND (G05) Kulit bersisik AND (G06) Kulit Berkerak AND (G07) Sering menggaruk badan THEN (P03) Scabies
3	R05	IF (G07) Sering menggaruk badan AND (G08) Bengkak dikulit AND (G10) Lemas AND (G11) Kehilangan nafsu makan THEN (P05) Abses

3. Hasil dan Diskusi

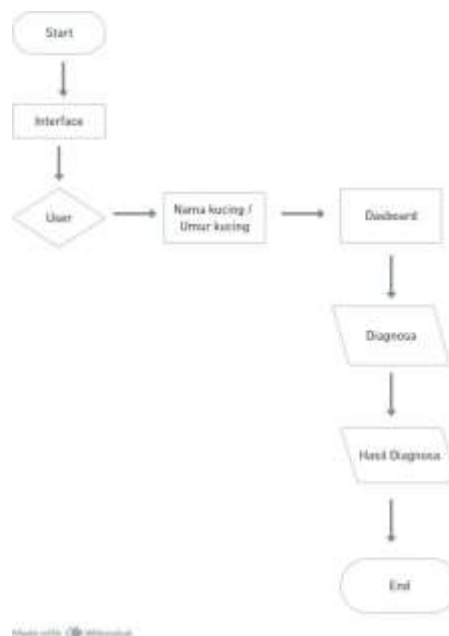
3.1 Implementasi Sistem

Aplikasi sistem pakar diagnosis penyakit kulit pada kucing ini dirancang untuk memberikan pengalaman penggunaan yang informatif dan mudah dipahami. Aplikasi ini memanfaatkan pendekatan teknologi modern berupa sistem pakar untuk membantu pemilik kucing dalam mengidentifikasi dan menangani permasalahan kulit berdasarkan gejala yang terlihat. Dengan adanya aplikasi ini, pengguna tidak hanya memperoleh hasil diagnosis secara cepat, tetapi juga mendapatkan pengetahuan awal mengenai gejala dan solusi penanganannya.

Secara umum, aplikasi ini terdiri dari beberapa halaman utama, halaman dashboard, halaman diagnosis gejala, dan halaman hasil diagnosis. Setiap halaman memiliki peran yang saling terintegrasi, Sehingga alur penggunaan aplikasi lebih sistematis, mulai dari input gejala hingga proses analisis dan penyajian hasil diagnosis.

3.2 Flowchart Alur Aplikasi

Flowchart merupakan teknik yang digunakan untuk menggambarkan urutan logika dari suatu prosedur dalam menyelesaikan masalah secara sistematis. Diagram ini menyajikan langkah-langkah penyelesaian dalam bentuk simbol-simbol tertentu yang mudah dipahami. Selain berfungsi sebagai alat bantu komunikasi antara perancang dan pengguna sistem, *Flowchart* juga berperan sebagai dokumentasi yang menjelaskan alur kerja program. Tujuan utama dari penggunaan *Flowchart* adalah untuk menggambarkan tahapan proses penyelesaian masalah secara sederhana, terstruktur, dan jelas, sehingga memudahkan analisis serta pengembangan sistem [14]. *Flowchart* pada penelitian ini menggambarkan alur kerja sistem pakar diagnosis penyakit kulit pada kucing berbasis *Android*, mulai dari tahap awal hingga akhir proses. Proses diawali dengan pengguna membuka aplikasi (*start*) dan masuk ke tampilan antarmuka (*interface*) untuk melakukan interaksi dengan sistem [15]. Selanjutnya, pengguna (*user*) memasukkan data berupa nama dan umur kucing yang kemudian diarahkan ke halaman utama (*dashboard*). Pada tahap ini, pengguna dapat memulai proses diagnosis dengan memilih gejala-gejala yang dialami kucing pada menu diagnosis. Setelah sistem memproses data tersebut, hasil diagnosis akan ditampilkan pada halaman hasil diagnosis yang berisi informasi jenis penyakit kulit beserta solusi penanganannya. Tahapan diakhiri dengan proses *end*, yang menandakan bahwa seluruh prosedur diagnosis telah selesai dilakukan.

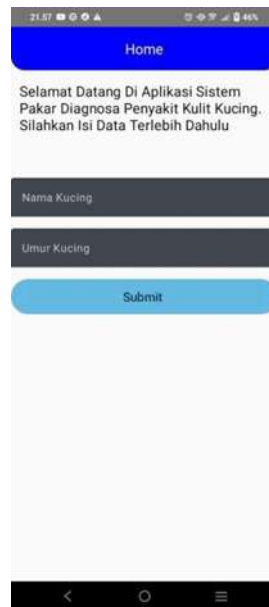


Gambar 3. Flowchart Alur Aplikasi

3.3 Tampilan Aplikasi

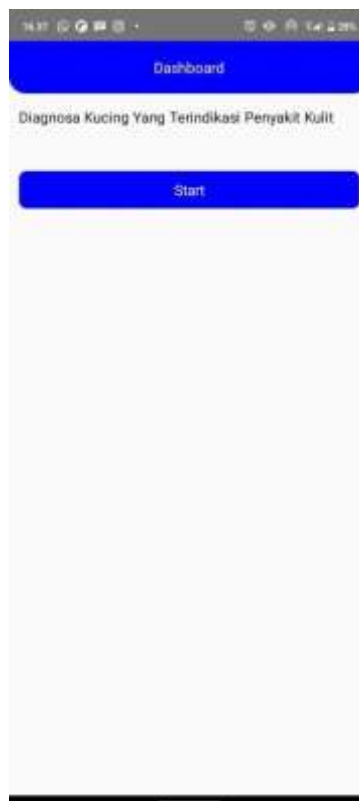
Tampilan aplikasi sistem pakar diagnosis penyakit kulit pada kucing berbasis *Android* dirancang dengan antarmuka yang sederhana dan mudah digunakan. Aplikasi ini terdiri dari beberapa halaman utama, yaitu halaman *home*, halaman diagnosis, dan halaman hasil diagnosis.

Halaman *Home* , Halaman ini merupakan tampilan awal aplikasi sistem pakar diagnosis penyakit kulit pada kucing. Pengguna memasukkan data identitas kucing berupa nama dan umur sebelum mengakses fitur utama. Data tersebut digunakan sebagai inisialisasi sehingga proses diagnosis dapat berjalan lebih terarah dan sesuai dengan kondisi kucing yang diperiksa.



Gambar 4. Tampilan Halaman *Home*

Halaman *Dashboard*, Pada halaman ini terdapat fitur “start” yang berfungsi untuk memulai proses diagnosis. Ketika tombol ditekan, pengguna akan diarahkan ke halaman pemilihan gejala. Pada tahap ini, pengguna menentukan gejala-gejala yang dialami kucing, yang kemudian akan diproses oleh sistem untuk menghasilkan diagnosis penyakit kulit secara otomatis.



Gambar 5. Tampilan Halaman *Dashboard*

Halaman Diagnosis Gejala, Pada halaman ini, pengguna memilih gejala-gejala yang sesuai dengan kondisi kulit kucing. Data gejala yang dipilih akan menjadi dasar proses penalaran sistem dalam menentukan penyakit. Setelah konfirmasi pilihan, pengguna akan diarahkan ke halaman hasil diagnosis untuk melihat informasi penyakit yang terdeteksi.



Gambar 6. Tampilan Halaman Diagnosis Gejala

Halaman Hasil Diagnosis, Halaman ini menampilkan hasil analisis berdasarkan gejala yang telah dipilih pengguna pada tahap sebelumnya. Informasi yang ditampilkan meliputi nama penyakit, dampak yang ditimbulkan pada kulit kucing, serta rekomendasi penanganan yang dapat dilakukan. Dengan adanya informasi tersebut, pengguna dapat memahami kondisi kucing secara lebih jelas dan memperoleh langkah penanganan yang sesuai.



Gambar 7. Tampilan Halaman Hasil Diagnosis

3.4 Pengujian Aplikasi

Pengujian sistem dilakukan menggunakan metode *Blackbox Testing*, yaitu metode pengujian yang berfokus pada

fungsionalitas sistem tanpa memperhatikan struktur internal atau kode program yang digunakan. Pengujian ini dilakukan dengan memberikan berbagai skenario masukan untuk memastikan bahwa setiap fitur dalam aplikasi dapat berjalan sesuai dengan yang diharapkan serta menghasilkan keluaran yang tepat. Dengan metode ini, kesalahan pada sistem dapat diketahui dari sisi pengguna sehingga aplikasi dapat diperbaiki dan dioptimalkan sebelum digunakan secara lebih luas [16]. Metode ini bertujuan untuk memastikan bahwa setiap fitur pada aplikasi dapat merespons input dengan benar serta menghasilkan keluaran yang sesuai dengan skenario penggunaan yang telah ditetapkan [17]. Dengan demikian, pengujian ini membantu memberi verifikasi bahwa sistem telah berjalan sesuai kebutuhan fungsional dan dapat digunakan oleh pengguna tanpa mengalami kesalahan proses.

Tabel 4. *Blackbox Testing*

No	Fungsi	Tampilan yang Diharapkan	Hasil Pengujian
1	Pengecekan halaman <i>home</i> , ketika pengguna tidak mengisi data nama kucing dan umur kucing maka tidak bisa berpindah ke halaman <i>dashboard</i>	Sistem akan menampilkan pemberitahuan agar mengisi nama dan umur kucing terlebih dahulu	Berhasil
2	Pengecekan halaman <i>dashboard</i> , ketika pengguna mengklik <i>button start</i> maka akan berpindah ke halaman diagnosis gejala	Sistem akan menampilkan halaman diagnosis gejala ketika pengguna mengklik <i>button start</i>	Berhasil
3	Pengecekan halaman diagnosis gejala ketika pengguna sudah memilih gejala yang dialami kucing dan mengklik <i>button lihat hasil</i> , maka akan berpindah ke halaman hasil diagnosis	Sistem akan menampilkan hasil diagnosis gejala	Berhasil

Hasil pengujian menggunakan metode *blackbox* yang ditampilkan pada Tabel 4 menunjukkan bahwa aplikasi dapat berfungsi dengan baik dalam menampilkan halaman *home*, *dashboard*, hingga halaman diagnosis. Sistem mampu memberikan peringatan ketika data belum diisi, menampilkan navigasi halaman sesuai tombol yang dipilih, serta menghasilkan output diagnosis berdasarkan gejala yang dimasukkan oleh pengguna. Seluruh skenario pengujian fungsional dinyatakan berhasil karena menghasilkan keluaran sesuai dengan yang diharapkan.

4. Kesimpulan

Berdasarkan penelitian ini, dapat disimpulkan bahwa Aplikasi Sistem Pakar Diagnosis Penyakit Kulit pada Kucing Berbasis *Android* dengan Metode *Forward Chaining* mampu mempermudah pemilik kucing dalam melakukan diagnosis awal secara mandiri, dengan tampilan antarmuka yang sederhana, mudah dipahami, serta mampu memberikan rekomendasi tindakan perawatan berdasarkan gejala yang dipilih. Meskipun demikian, pengembangan sistem ini masih memiliki keterbatasan, seperti cakupan penyakit yang masih terbatas pada penyakit kulit dan penggunaan yang hanya tersedia pada perangkat *Android*. Oleh karena itu, pada penelitian selanjutnya disarankan agar sistem dikembangkan dengan metode analisis yang lebih akurat, penambahan cakupan penyakit, serta fitur pendukung lainnya, sehingga aplikasi dapat menjadi referensi yang lebih komprehensif dan bermanfaat bagi pemilik kucing dalam menangani masalah kesehatan hewan peliharaan.

Referensi

1. S. Firmansyah, *Klasifikasi jenis kucing berdasarkan bentuk telinga dan bentuk wajah dengan metode K-Nearest Neighbor (KNN)*, Ph.D. dissertation, Universitas Islam Lamongan, 2021.
2. M. N. R. Abid and A. Hidayat, "Implementasi Virtual Reality PT. PLN (Persero) Gardu Induk Tegangan Ekstra Tinggi Balaraja," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 30, no. 1, pp. 34–41, 2025.
3. A. S. Pangestu and R. Tanamal, "Rancang bangun aplikasi sistem pakar berbasis mobile untuk mendiagnosis penyakit kulit pada kucing persia," *Teknika*, vol. 9, no. 2, pp. 81–87, 2020.
4. B. C. Saputra, S. B. Nauli, and W. Kurniawan, "Perancangan sistem pakar diagnosa penyakit pada hewan peliharaan menggunakan metode forward chaining," *SENTRI: Jurnal Riset Ilmiah*, vol. 2, no. 10, pp. 4193–4205, 2023.
5. I. Y. Panessai, *Arsitektur sistem pakar: Pengenalan sistem pakar*. 2021.
6. B. Pitaloka, B. Pitaloka, A. Putra, and S. Lestari, "Implementasi metode forward chaining dan backward chaining dalam mendeteksi kerusakan pada prasarana lalu lintas," in *Prosiding Seminar Nasional Darmajaya*, vol. 1, pp. 182–190, 2023.
7. A. S. Karim, S. Sutedi, and M. Agarina, "Prototype development of android-based thesis information system at Institute Informatics and Business (IIB) Darmajaya Bandar Lampung," in *Proceeding International Conference on Information Technology and Business*, pp. 122–132, 2019.

8. E. M. Rianof, B. P. Adhi, and Z. F. F. Putra, "Pengembangan aplikasi m-commerce pada toko optik menggunakan android studio," *PINTER: Jurnal Pendidikan Teknik Informatika dan Komputer*, vol. 4, no. 2, pp. 15–18, 2020.
9. Jubilee Enterprise, *Pengantar Pemrograman Kotlin*. Jakarta: Elex Media Komputindo, 2021.
10. F. Alpiansyah *et al.*, "Rancang bangun sistem administrasi perkantoran berbasis bahasa pemrograman java," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 2, no. 1, pp. 1–7, 2023.
11. T. Yuliano, "Pengenalan PHP," *IlmuKomputer.com*, 2007.
12. V. Soininen, "Jetpack Compose vs React Native: Differences in UI development," 2021.
13. Y. E. Achyani and S. Saumi, "Penerapan metode waterfall pada sistem informasi manajemen buku perpustakaan berbasis web," *Jurnal Saintekom*, vol. 9, no. 1, pp. 83–94, 2019.
14. Z. Tuasamu *et al.*, "Analisis sistem informasi akuntansi siklus pendapatan menggunakan DFD dan flowchart pada bisnis Porobico," *Jurnal Bisnis dan Manajemen (JURBISMAN)*, vol. 1, no. 2, pp. 495–510, 2023.
15. R. B. D. R. Nugraha and A. Wibowo, "Sistem pakar diagnosis penyakit kulit pada hewan menggunakan metode forward chaining," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 3, pp. 601–608, 2020, doi:10.25126/jtiik.2020732456.
16. M. N. R. Abid, S. C. Artanti, N. Adiyati, and E. Junaedi, "Android-Based Garbage Management Application Using K-Means Algorithmon RT 03/02 Kel. Karawaci Baru," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 4, pp. 735–743, 2024.
17. M. A. Hossain, M. R. Islam, and S. Ahmed, "An expert system for diagnosing animal diseases using forward chaining method," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, pp. 45–52, 2019, doi:10.14569/IJACSA.2019.0100506.