LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama	:	Fadila Rizka Nur Aminah
NIM	:	21090037
Program Studi	:	Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama	:	Muhammad Fikri Hidayattullah, S.T., M.Kom.
Status	:	Dosen
NIDN	:	0623108801
Jabatan Fungsional	:	Lektor
Pangkat/Golongan	:	Penata Tk. I/III-d

Pada hari ini Rabu tanggal 05 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama wajib menyelesaikan Tugas Akhir maksimal pada bulan Juni 2025, jika melebihi batas waktu maka Pihak Kedua berhak untuk tidak melanjutkan proses bimbingan Tugas Akhir. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

Tegal, 05 Maret 2025

Pihak Pertama	Pihak Kedua
Fadila Rizka Nur Aminah	Muhammad Fikri Hidayattullah, S.T., M.Kom.

Mengetahui,
Ketua Program Studi Sarjana Terapan Teknik Informatika

Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.225

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama	:	Fadila Rizka Nur Aminah
NIM	:	21090037
Program Studi	:	Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama	:	Mirza Alim Mutasodirin, S.Kom., M.Kom
Status	:	Dosen
NIDN	:	0605049004
Jabatan Fungsional	:	Asisten Ahli
Pangkat/Golongan	:	III/B

Pada hari ini Rabu tanggal 05 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Tugas Akhir minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

Tegal, 05 Maret 2025

Pihak Pertama	Pihak Kedua
Fadila Rizka Nur Aminah	Mirza Alim Mutasodirin, S.Kom., M.Kom

Mengetahui,
Ketua Program Studi Sarjana Terapan Teknik Informatika

Dyah Apriliani, S.T., M.Kom
NIPY. 09.015.225

Lampiran 2 Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1	Nama	:	Fadila Rizka Nur Aminah
	Kewarganegaraan	:	Indonesia
	Alamat	:	Desa Suradadi Rt 02/ Rw 02, Kecamatan Suradadi, Kabupaten Tegal, Provinsi Jawa Tengah, 52182
2	Nama	:	Muhammad Fikri Hidayattullah, S.T., M.Kom.
	Kewarganegaraan	:	Indonesia
	Alamat	:	Jl. Glatik No. 68 Randugunting, Kota Tegal, Jawa Tengah 52131
3	Nama	:	Mirza Alim Mutasodirin, S.Kom., M.Kom.
	Kewarganegaraan	:	Indonesia
	Alamat	:	Griya Santika blok J no. 11, Desa Pengabean, Kec. Dukuhturi, Kab. Tegal, Jawa Tengah, Indonesia 52192

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:

   Berupa : Program Komputer

   Berjudul : HealthCat: Sistem Deteksi Penyakit Kulit Pada Kucing Menggunakan *Convolutional Neural Network* Berbasis *Website*

   - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
   - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
   - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
   - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;

- Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
- Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
- Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
- Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:

   a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
   b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
   c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagimana mestinya.

Tegal, 23 Juni 2023

Fadila Rizka Nur Aminah
Pemegang Hak Cipta*

Muhammad Fikri Hidayattullah, S.T., M.Kom
Pemegang Hak Cipta*

Mirza Alim Mutasodirin, S.Kom., M.Kom.
Pemegang Hak Cipta*

Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3 Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1	Nama	: Fadila Rizka Nur Aminah
	Kewarganegaraan	: Indonesia
	Alamat	: Desa Suradadi Rt 02/ Rw 02, Kecamatan Suradadi, Kabupaten Tegal, Provinsi Jawa Tengah, 52182
2	Nama	: Muhammad Fikri Hidayattullah, S.T., M.Kom.
	Kewarganegaraan	: Indonesia
	Alamat	: Jl. Glatik No. 68 Randugunting, Kota Tegal, Jawa Tengah 52131
3	Nama	: Mirza Alim Mutasodirin, S.Kom., M.Kom.
	Kewarganegaraan	: Indonesia
	Alamat	: Griya Santika blok J no. 11, Desa Pengabean, Kec. Dukuhturi, Kab. Tegal, Jawa Tengah, Indonesia 52192

Adalah Pihak I selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama	:	Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	:	Jl. Mataram No.9, Pesurungan Lor, Kec. Margadana, Kota Tegal, Provinsi Jawa Tengah, 52147

Adalah Pihak II selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "HealthCat: Sistem Deteksi Penyakit Kulit Pada Kucing Menggunakan *Convolutional Neural Network* Berbasis *Website*" untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 23 Juni 2025

Pemegang Hak Cipta

Kepala P3M

(Muhammad Fikri Hidayattullah, S.T., M.Kom.)

Pencipta

(Fadila Rizka Nur Aminah)

(Muhammad Fikri Hidayattullah, S.T., M.Kom.)

(Mirza Alim Mutasodirin, S.Kom., M.Kom.)

Lampiran 4 Manual Book dan Dokumen Teknikal

DAFTAR ISI

DAFTAR GAMBAR

1. PENDAHULUAN

1.1. Tujuan Pembuatan Dokumen

Manual Book ini dirancang untuk memberikan panduan lengkap mengenai cara menggunakan *website* HealthCat, sebuah sistem berbasis web yang dirancang untuk mendeteksi penyakit kulit pada kucing secara cepat dan akurat menggunakan teknologi kecerdasan buatan (AI). Manual ini ditujukan bagi seluruh pengguna, baik *User* (pemilik kucing), Admin, maupun Dokter, agar dapat memahami cara kerja sistem, navigasi menu, serta fungsi-fungsi yang tersedia sesuai dengan peran (*role*) masing-masing. Dengan adanya dokumen ini, diharapkan pengguna dapat mengoperasikan sistem secara optimal, memanfaatkan fitur-fitur yang disediakan, serta membantu dalam peningkatan kualitas perawatan kucing secara menyeluruh.

1.2. Deskripsi Umum Sistem

HealthCat adalah platform digital berbasis web yang dirancang untuk membantu pengguna dalam mendeteksi dini penyakit kulit pada kucing serta memberikan edukasi seputar perawatan yang tepat. Sistem ini mampu mengidentifikasi empat jenis penyakit kulit beserta tingkat keparahannya, kemudian menampilkan hasil deteksi secara otomatis dan akurat. Selain fitur utama tersebut, HealthCat juga menyediakan berbagai layanan pendukung seperti artikel edukatif, forum diskusi antar pengguna, rekomendasi produk pengobatan yang disesuaikan dengan hasil deteksi, serta chatbot interaktif bernama CatGPT yang siap membantu menjawab pertanyaan seputar kesehatan kucing. Platform ini bertujuan untuk memberikan kemudahan baik bagi pengguna umum dalam merawat kucing peliharaan, maupun bagi admin dan dokter dalam melakukan monitoring, pengelolaan data, serta pemberian saran medis yang tepat.

Filosofi dan tujuan dari sistem ini direpresentasikan melalui logo HealthCat, yang menampilkan siluet tapak kaki kucing berwarna cokelat dengan ikon kucing putih dan simbol tambah (+) di dalamnya. Tapak kaki menggambarkan fokus utama pada hewan peliharaan, sedangkan simbol tambah menandakan peran layanan kesehatan yang menjadi inti dari platform. Warna cokelat memberi kesan hangat dan alami, sementara warna putih melambangkan kepercayaan dan keakuratan informasi. Logo ini tidak hanya menjadi identitas visual, tetapi juga mencerminkan komitmen HealthCat dalam menghadirkan solusi teknologi yang peduli, informatif, dan ramah bagi kesehatan kucing.

Gambar 1. 1 Logo Website HealthCat

1.3. Deskripsi Dokumen

Dokumen ini berisi panduan penggunaan *website* HealthCat yang disusun secara sederhana dan mudah dipahami. Di dalamnya mencakup informasi tentang perangkat yang dibutuhkan, struktur menu pada *website*, serta cara mengakses dan menggunakan fitur-fitur utama sesuai dengan peran pengguna, seperti user, admin, dan dokter. Dokumen ini bertujuan membantu pengguna menjalankan sistem dengan baik dan memahami fungsi dasar dari setiap menu yang tersedia.

2. PERANGKAT YANG DIBUTUHKAN

Untuk menjalankan dan mengembangkan *website* HealthCat dengan baik, terdapat dua jenis perangkat yang dibutuhkan, yaitu perangkat keras dan perangkat lunak.

2.1. Perangkat Lunak

Berikut adalah perangkat lunak yang diperlukan untuk menjalankan *website* HealthCat:

1. Visual Studio Code (VS Code), kode editor untuk pengembangan aplikasi web dan API, mendukung debugging dan kontrol versi yang efisien.
2. MySQL, digunakan sebagai sistem manajemen basis data untuk menyimpan data pengguna, hasil deteksi, artikel, produk, dan lainnya.
3. Web Browser, untuk mengakses sistem dari sisi pengguna, disarankan menggunakan browser modern seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge versi terbaru guna memastikan seluruh tampilan dan fitur *frontend* berfungsi dengan baik.

2.2. Perangkat Keras

Berikut adalah perangkat keras yang diperlukan untuk menjalankan *website* HealthCat:

1. Laptop, perangkat dengan spesifikasi minimal RAM 4GB dan sistem operasi Windows, Linux, atau MacOS, untuk mendukung akses *website* HealthCat secara optimal melalui browser, termasuk eksplorasi semua fitur yang tersedia.
2. Smarthphone, perangkat dengan RAM minimal 2GB untuk mengakses *website* Healthcat dengan lancar, termasuk fitur seperti deteksi penyakit kulit, rekomendasi procuk, CatGPT.
3. Koneksi internet yang cepat dan stabil diperlukan untuk mendukung pemesanan tiket, pengunduhan konten, dan interaksi aplikasi dengan *backend* secara real-time.

3. MENGAKSES SISTEM & TAMPILAN AWAL

3.1. Akses Website

Website HealthCat dapat diakses melalui browser modern seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge dengan koneksi internet yang stabil. Pengguna cukup mengetikkan alamat URL *website* pada kolom pencarian browser, kemudian halaman utama (beranda) akan ditampilkan. Tidak diperlukan instalasi tambahan untuk menggunakan layanan ini dari sisi pengguna. Halaman utama website saat pertama kali diakses ditunjukkan pada Gambar 3.1.

Gambar 3. 1 Halaman utama Website

3.2. Login dan Registrasi

Untuk dapat menggunakan fitur utama pada *website* HealthCat, setiap pengguna wajib memiliki akun. Terdapat dua jalur akses, yaitu melalui Daftar akun baru dan Masuk akun yang sudah terdaftar. Pengguna dapat mengklik tombol Masuk atau Daftar yang tersedia di pojok kanan atas halaman. Saat melakukan registrasi, pengguna perlu mengisi informasi dasar seperti nama, email, dan password. Sistem akan menyimpan data tersebut dan memberikan akses sesuai dengan role user. Saat ini role dokter dan admin ditetapkan dari admin. Setelah proses registrasi selesai, pengguna dapat langsung login menggunakan email

dan password yang telah terdaftar. Jika berhasil login, sistem akan secara otomatis mengarahkan pengguna ke tampilan sesuai dengan role-nya: user ke halaman beranda, admin ke dashboard admin, dan dokter ke dashboard admin dengan akses yang terbatas. Halaman masuk dan daftar dapat dilihat Gambar 3.2.

Gambar 3. 2 Tampilan Form Login dan Register

3.3. Tampilan Sebelum Login

Sebelum login, pengguna hanya dapat mengakses fitur-fitur umum seperti halaman Beranda, Artikel, Forum Diskusi, Tentang. Menu navigasi atas akan menampilkan tombol Masuk dan Daftar. Jika pengguna mencoba mengakses fitur yang membutuhkan autentikasi seperti Deteksi, komentar pada forum, dan profil. Maka sistem akan otomatis mengarahkan pengguna ke halaman login dan menampilkan notifikasi bahwa login diperlukan. Tampilan menu dan halaman website sebelum login ditunjukkan pada Gambar 3.3.

Gambar 3. 3 Tampilan Sebelum Login

3.4. Tampilan Setelah Login

Setelah berhasil masuk, tampilan halaman dan menu akan disesuaikan berdasarkan role pengguna yang terdaftar dalam sistem, yaitu sebagai *User*, Admin, atau Dokter. Masing-masing role akan mendapatkan akses dan tampilan antarmuka yang berbeda sesuai fungsinya.

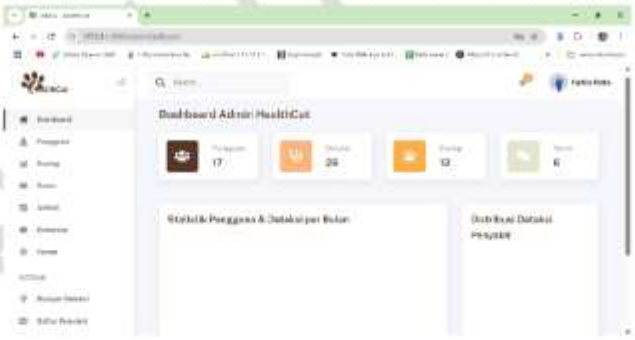
3.4.1. Tampilan User

Setelah masuk sebagai *user*, pengguna akan melihat tampilan beranda yang lebih lengkap dengan akses ke fitur utama yaitu Deteksi. Pada bagian kanan atas, terdapat menu dropdown profil yang berisi akses ke Edit Profil, Data Kucing, Riwayat Deteksi, dan Logout. Struktur menu dan tampilan halaman setelah login sebagai user dapat dilihat pada Gambar 3.4.

Gambar 3. 4 Tampilan Setelah Login (*User*)

3.4.2. Tampilan Admin

Admin akan diarahkan ke Dashboard Backend setelah login. Menu sidebar pada dashboard admin mencakup fitur manajemen pengguna, artikel, forum, penyakit, produk, deteksi, feedback, dan statistik. Admin memiliki kontrol penuh terhadap seluruh data sistem. Tampilan dashboard khusus untuk admin ditunjukkan pada Gambar 3.5.



Gambar 3. 5 Tampilan Setelah *Login* (Admin)

3.4.3. Tampilan Dokter

Setelah login, dokter akan melihat dashboard dengan tampilan serupa admin namun dengan akses terbatas. Dokter hanya dapat mengakses fitur Riwayat Deteksi, Detail Kucing, serta memberikan Catatan atau

Saran Medis. Menu seperti manajemen pengguna dan artikel tidak tersedia bagi dokter. Tampilan untuk role dokter dapat dilihat pada Gambar 3.6.

Gambar 3. 6 Tampilan Setelah Login (Dokter)

4. MENU DAN CARA PENGGUNAAN

Setelah berhasil melakukan login melalui halaman utama *website*, pengguna akan diarahkan ke tampilan dan menu yang berbeda sesuai dengan peran atau role yang dimilikinya dalam sistem. Role dalam *website* HealthCat terbagi menjadi tiga, yaitu *User*, Admin, dan Dokter. Masing-masing role memiliki akses dan fungsi tertentu yang telah disesuaikan dengan kebutuhannya. Berikut adalah penjelasan struktur menu dan cara penggunaan fitur pada masing-masing role:

4.1. Penggunaan *User*

Role *User* ditujukan bagi pemilik kucing yang ingin memanfaatkan *website* HealthCat untuk mendeteksi penyakit kulit, membaca artikel edukatif, berdiskusi di forum, mendapatkan rekomendasi produk, serta mencatat riwayat deteksi. *User* merupakan *role* utama dalam sistem karena menjadi pusat interaksi dari fitur-fitur berbasis layanan.

4.1.1. Struktur Menu *User*

Setelah *login*, *user* akan diarahkan ke halaman beranda yang menampilkan ringkasan fitur utama dan akses cepat ke berbagai menu. Adapun struktur menu yang dapat diakses oleh *user* antara lain:

1. Halaman Beranda

   Halaman Beranda merupakan tampilan utama yang pertama kali dilihat oleh pengguna saat mengakses website. Pada halaman ini, pengguna diperkenalkan dengan tampilan visual sistem HealthCat yang bersih, informatif, dan intuitif. Halaman ini berfungsi sebagai pusat informasi awal, yang memuat navigasi ke fitur-fitur utama seperti Deteksi, Artikel, Forum Diskusi, dan CatGPT.

   Selain itu, pada bagian bawah halaman terdapat formulir Feedback yang memungkinkan pengguna untuk menyampaikan kritik, saran, atau masukan yang konstruktif. Jika pengguna telah login ke dalam sistem, maka form Feedback akan otomatis mengisi kolom

nama dan email berdasarkan data akun. Sebaliknya, jika pengguna belum login, kolom tersebut akan kosong dan dapat diisi secara manual. Formulir ini menjadi jembatan komunikasi antara pengguna dan pengembang, guna meningkatkan kualitas sistem ke depannya. Tampilan formulir feedback dihalaman beranda dapat dilihat di gambar 4.1

Gambar 4. 1 Tampilan Formulir Feedback di Halaman Beranda

2. Halaman Deteksi

Halaman Deteksi merupakan fitur inti dari sistem HealthCat yang digunakan oleh pengguna untuk menganalisis kondisi kulit kucing berdasarkan gambar yang diunggah. Dengan bantuan teknologi Artificial Intelligence (AI) berbasis deep learning, sistem mampu mengidentifikasi jenis penyakit kulit, menentukan tingkat keparahan, serta memberikan rekomendasi produk dan penanganan.

Untuk dapat mengakses halaman deteksi, pengguna diwajibkan untuk login terlebih dahulu. Jika pengguna belum login, maka sistem akan secara otomatis mengarahkan ke halaman login dan menampilkan notifikasi bahwa autentikasi diperlukan. Setelah berhasil login, pengguna dapat mengikuti langkah-langkah berikut:

a. Mengakses Menu Deteksi

pengguna dapat mengklik tombol Deteksi untuk menuju halaman deteksi penyakit. Menu ini tersedia baik di halaman beranda maupun pada navigasi atas seluruh halaman. Tampilan awal halaman deteksi berisi tutorial penggunaan kemudian klik mulai deteksi untuk melanjutkan. Dapat dilihat pada gambar 4.2.

Gambar 4. 2 Tutorial Penggunaan Deteksi

b. Memilih Kucing yang Akan Didiagnosis

Sebelum melakukan deteksi, pengguna diminta untuk memilih kucing dari daftar kucing yang telah didaftarkan sebelumnya. Jika pengguna belum memiliki data kucing, atau ingin menambahkan kucing baru, tersedia tombol "Daftar Kucing" yang terletak di sebelah komponen pemilihan kucing. Dan isi semua formulirnya. Fungsi menu ini sangat penting untuk memastikan bahwa proses deteksi dapat dikaitkan dengan identitas kucing yang spesifik, sehingga hasil deteksi dapat disimpan ke dalam riwayat kucing tersebut secara terstruktur. Tampilan pemilihan kucing dan Daftarkan Kucing Baru dapat dilihat pada Gambar 4.3.

Gambar 4. 3 Tampilan pemilihan kucing

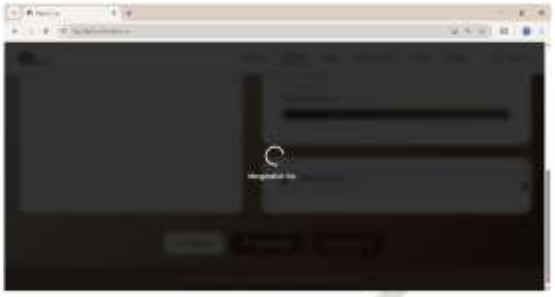
c. Mengunggah Gambar Kulit Kucing

Setelah memilih kucing, pengguna akan diminta untuk mengunggah atau memotret gambar kulit kucing yang ingin dideteksi, dan klik analisis foto untuk melanjutkan proses deteksi penyakit kulit seperti pada Gambar 4.4.

Gambar 4. 4 Tampilan Upload Gambar

d. Memulai Proses Deteksi

Setelah pengguna klik tombol analisis foto maka sistem akan menampilkan indikator loading yang menunjukkan bahwa proses deteksi sedang berlangsung, Waktu pemrosesan tergantung pada ukuran gambar dan koneksi internet, namun biasanya tidak lebih dari beberapa detik. Seperti pada Gambar 4.5.



Gambar 4. 5 Tampilan Indikator Proses Deteksi

e. Melihat Hasil Deteksi

Setelah proses deteksi selesai, sistem akan menampilkan hasil secara lengkap, meliputi:

- Gambar yang Didiagnosis: Ditampilkan ulang untuk konfirmasi
- Informasi kucing yang dideteksi: menampilkan nama kucing, ras, umur
- Nama Penyakit: Scabies, Ringworm, Feline Acne, Flea allergy, dan Sehat
- Tingkat Keparahan: Ringan atau Parah
- Rekomendasi Penanganan: Rekomendasi tindakan lanjutan, seperti penggunaan obat, perawatan di rumah, atau saran untuk konsultasi ke dokter hewan jika parah.

Setelah hasil deteksi ditampilkan, sistem juga menyediakan tiga tombol aksi utama di bagian bawah tampilan hasil deteksi, masing-masing memiliki fungsi sebagai berikut:

- Tombol "Kembali" : Digunakan untuk kembali ke halaman utama deteksi, misalnya jika pengguna ingin melihat ulang halaman sebelumnya atau mengubah pilihan kucing.
- Tombol "Deteksi Baru" : Digunakan untuk memulai ulang proses deteksi dengan gambar yang berbeda. Tombol ini

akan mereset formulir deteksi dan memungkinkan pengguna untuk memilih ulang gambar atau kucing lainnya.

- Tombol "Lihat Produk" : Tombol ini akan mengarahkan pengguna ke halaman Rekomendasi Produk berdasarkan hasil deteksi yang diperoleh. Sistem akan secara otomatis menyaring produk sesuai jenis penyakit dan tingkat keparahan yang telah teridentifikasi. Hasil deteksi dapat dilihat seperti Gambar 4.6.

Gambar 4. 6 Tampilan Hasil Deteksi

3. Halaman Rekomendasi

   Halaman Rekomendasi Produk berfungsi untuk menampilkan berbagai pilihan obat atau perawatan yang sesuai dengan hasil deteksi penyakit kulit pada kucing. Produk yang ditampilkan telah dikategorikan berdasarkan jenis penyakit dan tingkat keparahan, seperti scabies ringan, scabies parah, ringworm, acne, atau flea allergy. Pengguna dapat membuka halaman ini melalui dua cara:

   a. Menu Navigasi Atas (Navbar):

      Klik menu "Rekomendasi Produk" di bagian atas halaman *website*.

   b. Dari Hasil Deteksi:

      Setelah deteksi selesai, klik tombol "Lihat Produk" di bagian bawah hasil. Sistem akan langsung menampilkan produk yang sesuai dengan hasil diagnosa. Seperti dapat dilihat pada Gambar 4.7.

Gambar 4. 7 Tampilan Halaman Rekomendasi Produk

Jika klik salah satu produk akan ditampilkan nama produk, deskripsi, kategori penyakit, harga, jumlah terjual, rating, dan tombol beli sekarang di Shopee, seperti pada Gambar 4.8.

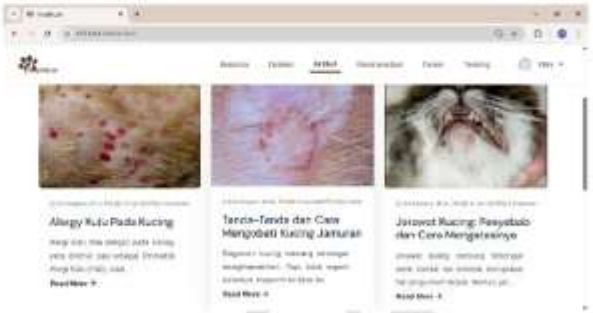
Gambar 4. 8 Tampilan Detail Produk

4. Halaman Artikel

Halaman Artikel menyajikan berbagai informasi dan edukasi seputar kesehatan kulit kucing, tips perawatan, dan pencegahan penyakit. Artikel ditulis dengan bahasa yang ringan namun informatif, agar mudah dipahami oleh semua kalangan pengguna. Berikut langkah-langkah penggunaan halaman Artikel:

a. Klik menu "Artikel" di navbar untuk membuka daftar artikel.

b. Scroll ke bawah untuk melihat seluruh artikel yang tersedia dalam bentuk kartu berisi judul, konten, dan gambar. Seperti Gambar 4.9.



Gambar 4. 9 Tampilan Halaman Artikel

c. Klik judul artikel untuk membaca isi lengkap.
d. Jika sudah login, Anda dapat memberi komentar di bagian bawah artikel.
e. Komentar yang sudah dikirim dapat diedit dan dihapus menggunakan tombol edit di samping komentar. Seperti pada Gambar 4.10.

Gambar 4. 10 Tampilan Komentar Pada Artikel

5. Halaman Forum

Halaman Forum berfungsi sebagai ruang diskusi antar pengguna seputar kesehatan kucing, pengalaman pribadi, serta saling

bertukar informasi atau tips perawatan. Berikut langkah-langkah penggunaannya:

a. Akses halaman Forum melalui menu "Forum" di bagian *navbar website*. Tampilan daftar diskusi awal dapat dilihat pada Gambar 4.12.

Gambar 4. 11 Tampilan Halaman Forum

b. Klik judul diskusi untuk membaca isi lengkap dan komentar di dalamnya, seperti pada Gambar 4.12.

Gambar 4. 12 Tampilan Halaman Detail Forum Diskusi

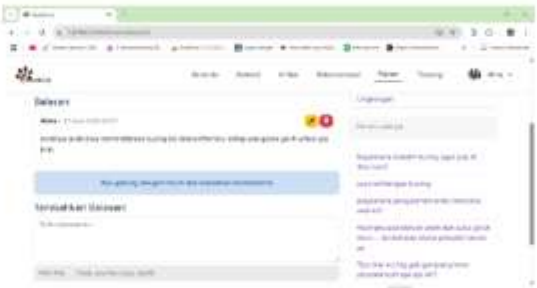
c. Untuk membuat diskusi baru, klik tombol "Buat Diskusi" di bagian atas, lalu isi formulir judul dan isi diskusi, seperti pada Gambar 4.13.

Gambar 4. 13 Tampilan Buat Forum

d. Pengguna dapat mengedit atau menghapus diskusinya sendiri melalui tombol yang tersedia, seperti pada Gambar 4.14.

Gambar 4. 14 Tampilan Tombol Hapus dan Edit Forum

e. Pengguna lain juga dapat memberikan komentar, mengedit atau menghapus melalui tombol yang tersedia, seperti pada Gambar 4.15.



Gambar 4. 15 Tampilan Balasan Forum Diskusi

f. Tersedia juga like/dislike disediakan untuk menanggapi isi diskusi tanpa memberikan komentar. Seperti pada gambar 4.16.

Gambar 4. 16 Tampilan Like Dislike Forum Diskusi

6. Halaman Tentang

Halaman Tentang menyajikan informasi singkat mengenai tujuan, latar belakang, dan manfaat dari pengembangan website HealthCat sebagai platform deteksi penyakit kulit pada kucing berbasis AI. Halaman ini juga memperkenalkan visi, misi, serta tim pengembang. Tampilan halaman dapat dilihat pada Gambar 4.17.

Gambar 4. 17 Tampilan Halaman Tentang

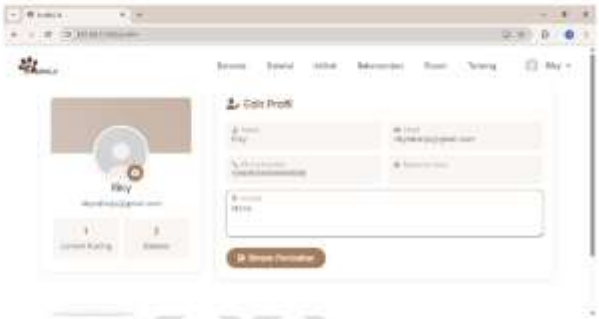
7. Fitur CatGPT

   Halaman CatGPT merupakan fitur chatbot berbasis kecerdasan buatan yang dirancang untuk membantu pengguna mendapatkan informasi dan menjawab pertanyaan seputar kesehatan kulit kucing secara cepat dan interaktif. Pengguna dapat mengetik pertanyaan di kolom chat, seperti gejala penyakit, perawatan, atau saran umum mengenai kondisi kucing mereka. Chatbot ini bekerja secara responsif dan tersedia selama 24 jam, memberikan pengalaman konsultasi awal sebelum pengguna memutuskan langkah lanjutan. Fitur ini dapat diakses melalui menu utama, dan tampilan halaman CatGPT dapat dilihat pada Gambar 4.18.

Gambar 4. 18 Tampilan Fitur CatGPT

8. Halaman Edit Profil

Halaman edit profil berada di navbar profil. Pengguna dapat memperbarui informasi akun seperti nama, email, dan foto profil dan informasi lain. Perubahan yang dilakukan akan langsung tersimpan dalam sistem setelah menekan tombol "Simpan Perubahan". Tampilan halaman Edit Profil dapat dilihat pada Gambar 4.19.



Gambar 4. 19 Tampilan Halaman Edit Profile

9. Halaman Data Kucing

Pada halaman ini, pengguna dapat mengelola informasi kucing peliharaan mereka. Di sisi kiri terdapat daftar semua kucing yang telah ditambahkan, lengkap dengan tombol "Tambah Kucing". Di sisi kanan, detail kucing yang dipilih akan ditampilkan dan dapat diedit atau dihapus. Jika pengguna menekan tombol hapus, sistem akan menampilkan pop-up konfirmasi. Tampilan halaman Data Kucing ditunjukkan pada Gambar X.25.

Gambar 4. 20 Tampilan Halaman Kucing

10. Halaman Riwayat Deteksi

Halaman ini menampilkan riwayat hasil deteksi penyakit kulit pada setiap kucing yang telah diperiksa. Pengguna dapat menyaring data berdasarkan nama kucing, tanggal, jenis penyakit, dan tingkat keparahan. Tersedia juga tombol "Download PDF" untuk menyimpan hasil riwayat. Tampilan halaman Riwayat Deteksi dapat dilihat pada Gambar 4.21.

Gambar 4. 21 Tampilan Halaman Riwayat Deteksi

11. Logout

Fitur Logout digunakan untuk keluar dari akun pengguna secara aman. Menu ini tersedia melalui dropdown pada profil pengguna di pojok kanan atas. Setelah logout, pengguna akan diarahkan

kembali ke halaman login. Tampilan opsi logout terlihat pada Gambar 4.22.

Gambar 4. 22 Tampilan Pop up Logout

4.2. Penggunaan Admin

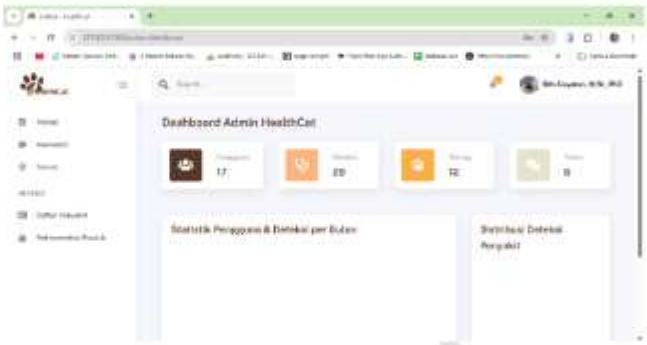
Role Admin memiliki hak akses penuh terhadap seluruh data dan pengelolaan konten pada sistem. Admin bertanggung jawab atas manajemen pengguna, artikel, forum, produk, penyakit, riwayat deteksi, dan aktivitas lainnya yang berkaitan dengan sistem.

4.2.1. Stuktur Menu Admin

Setelah login, admin akan diarahkan ke dashboard Admin, dengan struktur menu sidebar sebagai berikut:

1. Dashboard

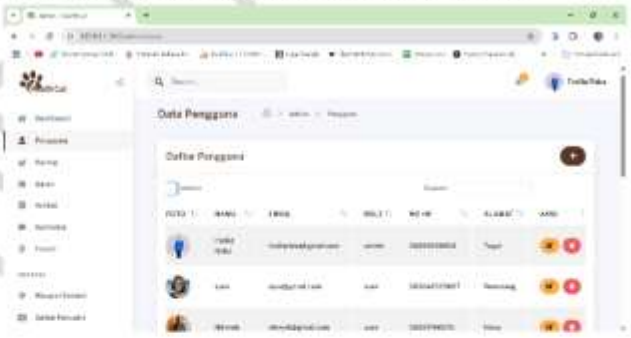
   Menampilkan ringkasan statistik utama sistem seperti jumlah total pengguna, jumlah kucing yang terdaftar, total deteksi yang dilakukan, jumlah artikel, serta data aktivitas terkini. Tampilan ini berfungsi sebagai overview atau pusat kontrol utama. Dapat dilihat pada Gambar 4.25.



Gambar 4. 23 Tampilan Dashboard Admin

2. Manajemen Pengguna

   Berfungsi untuk mengelola seluruh akun dalam sistem, baik itu user biasa, dokter, maupun admin. Admin dapat melihat daftar pengguna, mengedit profil pengguna, mengubah peran (role), mengaktifkan atau menonaktifkan akun, serta menghapus pengguna jika diperlukan. Dapat dilihat pada Gambar 4.24.



Gambar 4. 24 Tampilan Halaman Manajemen Pengguna

3. Manajemen Kucing

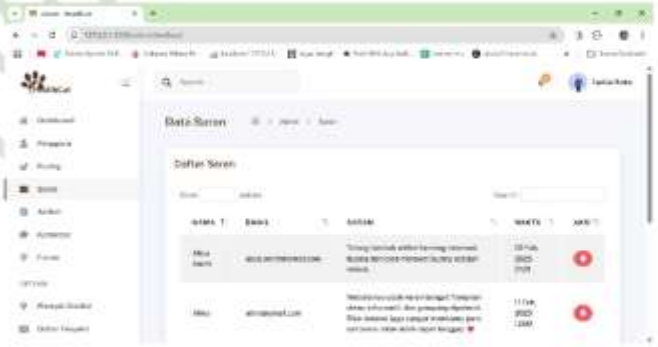
   Menampilkan seluruh data kucing yang telah ditambahkan oleh pengguna. Admin dapat melihat informasi detail setiap kucing, serta melakukan pengeditan atau penghapusan data jika

diperlukan. Fitur ini penting untuk moderasi dan validasi data hewan peliharaan yang terdaftar dalam sistem. Dapat dilihat pada Gambar 4.25.

Gambar 4. 25 Tampilan Halaman Manajemen Kucing

4. Manajemen Saran

Berisi daftar masukan, kritik, dan saran dari pengguna yang dikirim melalui formulir feedback. Admin hanya dapat membaca, menghapus, atau menindaklanjuti saran tersebut sebagai bahan evaluasi dan pengembangan fitur lebih lanjut. Dapat dilihat pada Gambar 4.26.



Gambar 4. 26 Tampilan Halaman Manajemen Saran / Masukan

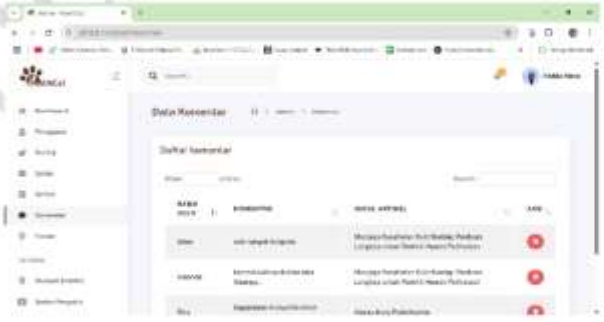
5. Manajemen Artikel

Digunakan untuk membuat, menyunting, dan menghapus artikel edukatif seputar kesehatan kucing. Artikel yang dipublikasikan melalui menu ini akan muncul pada halaman frontend (user). Admin juga dapat menambahkan gambar. Dapat dilihat pada Gambar 4.27.

Gambar 4. 27 Tampilan Halaman Manajemen Artikel

6. Manajemen Komentar Artikel

Menampilkan seluruh komentar yang diberikan pengguna pada artikel. Admin dapat memoderasi komentar, menyunting konten, atau menghapus komentar yang mengandung unsur tidak sesuai (spam, tidak sopan, dsb). Dapat dilihat pada Gambar 4.28.



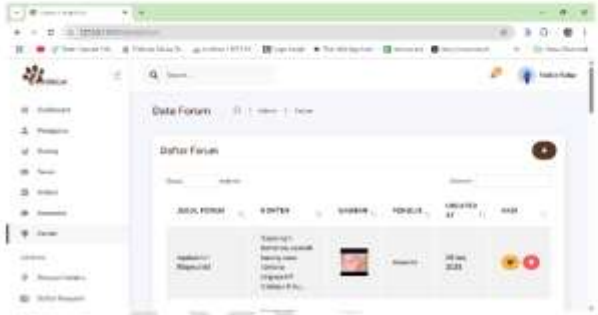
Gambar 4. 28 Tampilan Manajemen Komentar Artikel

7. Manajemen Forum

Halaman Forum terbagi menjadi tiga bagian:

a. Forum Diskusi

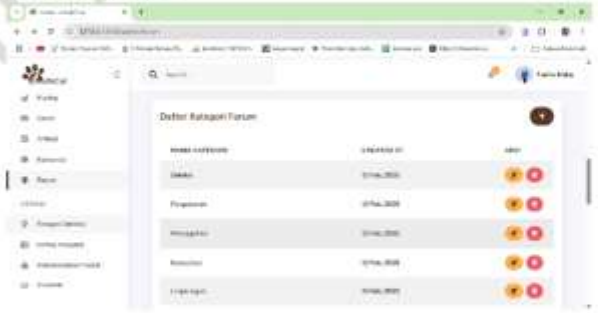
Menampilkan seluruh topik diskusi yang dibuat pengguna. Admin dapat membaca, menyembunyikan, atau menghapus topik. Seperti pada Gambar 4.29.



Gambar 4. 29 Tampilan Manajemen Forum

b. Kategori Forum

Digunakan untuk menambah, mengedit, atau menghapus kategori diskusi, seperti "Perawatan", "Penyakit", "Umum", dan sebagainya. Tampilan manajemen kategori dapat dilihat pada Gambar 4.30.



Gambar 4. 30 Tampilan Manajemen Kategori Forum

c. Balasan Diskusi

Berisi seluruh komentar/balasan dalam setiap forum. Admin dapat memfilter, mengedit, atau menghapus balasan dari pengguna. Dapat dilihat pada Gambar 4.31.

Gambar 4. 31 Tampilan Manajemen Balasan Forum

8. Riwayat Deteksi

Menampilkan seluruh data hasil deteksi yang dilakukan oleh pengguna. Admin dapat melihat detail riwayat berdasarkan waktu, pengguna, jenis penyakit, dan nama kucing. Halaman ini dapat digunakan untuk keperluan rekap dan analisis. Admin hanya dapat menghapus dan melihat saja. Seperti pada Gambar 4.32.

Gambar 4. 32 Tampilan Halaman Riwayat Deteksi

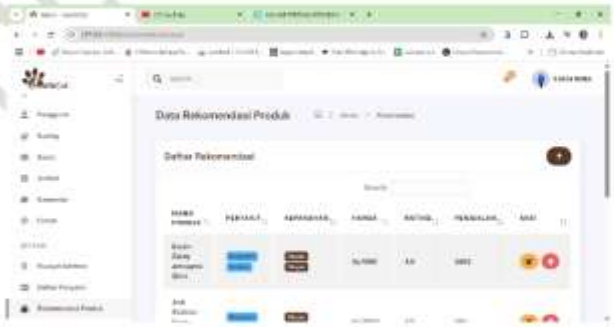
9. Daftar Penyakit

Berfungsi untuk mengelola jenis-jenis penyakit kulit pada kucing yang dikenali oleh sistem. Admin hanya dapat mengedit deskripsi penyakit, rekomendasi penanganan, karena sudah disesuaikan dengan dataset dari model AI. Seperti dilihat pada Gambar 4.33.

Gambar 4. 33 Tampilan Halaman Daftar Penyakit

10. Daftar Produk Rekomendasi

Menampilkan produk-produk pengobatan yang direkomendasikan berdasarkan hasil deteksi. Admin dapat mengelola nama produk, gambar, harga, deskripsi, jenis obat, serta menghubungkan produk dengan penyakit tertentu. Admin juga dapat menghapus mengedit produk. Dapat dilihat pada Gambar 4.34.



Gambar 4. 34 Tampilan Halaman Daftar Produk

4.3. Penggunaan Dokter

Role Dokter memiliki akses terbatas dalam dashboard. Dokter berfungsi untuk melihat riwayat deteksi dari pengguna, memberikan catatan atau saran medis, dan menganalisis hasil deteksi sebagai bahan pertimbangan lebih lanjut jika diperlukan tindakan lanjutan.

4.3.1. Stuktur Menu Dokter

Setelah berhasil login, pengguna dengan role Dokter akan diarahkan ke halaman Dashboard yang secara tampilan mirip dengan admin. Namun, akses fitur yang tersedia telah dibatasi hanya pada menu yang relevan dengan tugas dan tanggung jawab dokter dalam sistem. Dokter tidak memiliki akses untuk mengelola pengguna atau data internal lainnya yang bersifat administratif. Berikut adalah struktur menu yang tersedia untuk role Dokter:

1. Dashboard

   Menu ini menampilkan ringkasan aktivitas sistem seperti jumlah artikel, forum, dan data penyakit. Fungsinya serupa dengan menu dashboard pada admin. Dapat dilihat pada Gambar 4.35.

Gambar 4. 35 Dashboard Dokter

2. Artikel

   Menu ini memiliki fungsi yang sama seperti pada admin, yaitu untuk menulis, mengedit, atau menghapus artikel yang berkaitan

dengan kesehatan kucing. Dokter dapat membagikan informasi medis yang relevan untuk diedukasi kepada pengguna.

3. Komentar Artikel

   Seperti pada admin, menu ini memungkinkan dokter melihat dan merespons komentar pengguna terhadap artikel. Namun, dokter hanya memiliki kewenangan terbatas dalam menghapus komentar.

4. Forum

   Dokter dapat mengakses forum diskusi, kategori, dan balasan komentar seperti admin, namun tidak memiliki akses untuk memoderasi secara penuh atau mengatur kategori secara langsung. Fitur ini disediakan untuk mendorong partisipasi dokter dalam diskusi pengguna.

5. Daftar Penyakit

   Menu ini memiliki tampilan dan data yang sama dengan versi admin. Namun, peran dokter difokuskan untuk meninjau atau menyarankan perubahan pada deskripsi penyakit, bukan untuk menambah atau menghapus data.

6. Rekomendasi Produk

   Menu ini menampilkan daftar produk yang direkomendasikan berdasarkan penyakit. Sama seperti pada admin, dokter dapat melihat produk, memahami relasi produk dengan penyakit tertentu, namun tidak memiliki akses untuk menambahkan atau menghapus produk.

Catatan : Semua menu yang dapat diakses oleh dokter pada dasarnya memiliki tampilan dan fungsi serupa dengan versi admin, namun dengan batasan hak akses sesuai dengan peran dokter dalam sistem. Hal ini bertujuan untuk menjaga validitas data dan menjaga batas kewenangan antar peran pengguna.

DOKUMEN TEKNIKAL

HEALTHCAT

Platform berbasis web yang memudahkan pemilik kucing untuk mendeteksi penyakit kulit, menyimpan riwayat kesehatan, berbagi di forum, membaca artikel informatif, dan mendapatkan rekomendasi produk pengobatan yang sesuai..

Fadila Rizka Nur Aminah
Muhammad Fikri Hidayattullah, S.T., M.Kom.
Mirza Alim Mutasodirin, M.Kom.

DAFTAR ISI

DAFTAR GAMBAR

1. PENDAHULUAN

1.1. Deskripsi Sistem

HealthCat adalah platform berbasis web yang dikembangkan untuk membantu pengguna dalam mendeteksi penyakit kulit pada kucing secara otomatis dengan memanfaatkan teknologi kecerdasan buatan. Sistem ini mampu mengenali empat jenis penyakit kulit pada kucing yang diklasifikasikan berdasarkan dua tingkat keparahan, yaitu ringan dan parah. Setelah pengguna mengunggah gambar kondisi kulit kucing, sistem akan menampilkan hasil deteksi secara langsung. Meskipun menyediakan fitur deteksi awal, HealthCat tidak dirancang untuk menggantikan peran dokter hewan. Sebaliknya, sistem ini mendukung tindakan lanjutan yang lebih profesional—khususnya pada hasil dengan kategori parah, sistem akan memberikan rekomendasi kepada pengguna untuk membawa kucing ke klinik atau dokter hewan terdekat.

Selain fitur utama berupa deteksi, HealthCat juga dilengkapi dengan layanan pendukung seperti artikel edukatif, forum diskusi antar pengguna, sistem rekomendasi produk perawatan sesuai hasil deteksi, serta chatbot interaktif bernama CatGPT. Sistem ini dikembangkan tidak hanya untuk memberikan kemudahan dalam pemantauan kesehatan kucing peliharaan oleh pengguna umum, tetapi juga sebagai alat bantu bagi admin dan dokter untuk memantau data deteksi, memberikan saran medis, dan mengelola informasi secara efisien dan terstruktur.

1.2. Tujuan dan Manfaat

Tujuan utama dari pengembangan sistem HealthCat adalah untuk menyediakan solusi deteksi penyakit kulit kucing secara cepat, akurat, dan mudah diakses oleh masyarakat umum. Dengan memanfaatkan teknologi computer vision dan klasifikasi citra, pengguna dapat memperoleh hasil diagnosa awal tanpa harus langsung mengunjungi dokter hewan. Selain itu, sistem ini juga memberikan manfaat sebagai

media edukasi melalui artikel dan forum diskusi, serta menawarkan rekomendasi produk perawatan yang relevan. Bagi pihak admin dan dokter, HealthCat mempermudah proses monitoring, evaluasi data deteksi, serta memberikan catatan medis kepada pemilik kucing secara terstruktur dan terpusat.

Manfaat yang dihadirkan antara lain:

- Mempermudah pengguna dalam mengidentifikasi jenis penyakit kulit pada kucing melalui fitur deteksi berbasis gambar.
- Memberikan rekomendasi perawatan dan produk yang relevan berdasarkan hasil deteksi dan tingkat keparahan penyakit.
- Menyediakan ruang edukasi melalui artikel dan forum diskusi.
- Mendukung peran dokter hewan melalui fitur pencatatan dan analisis data deteksi, tanpa menggantikan pemeriksaan klinis langsung.
- Memberikan pengalaman interaktif melalui chatbot CatGPT untuk menjawab pertanyaan umum seputar kesehatan dan perawatan kucing.

1.3. Deskripsi Dokumen

Dokumen ini disusun sebagai dokumentasi teknis dari sistem HealthCat, yang bertujuan memberikan penjelasan menyeluruh terkait arsitektur, teknologi, dan proses pengembangan yang digunakan. Di dalamnya mencakup deskripsi lingkungan pengembangan (perangkat lunak dan perangkat keras), implementasi model klasifikasi penyakit kulit berbasis deep learning, struktur pengembangan frontend dan backend berbasis Flask dengan pendekatan Blueprint, serta integrasi berbagai fitur utama seperti deteksi penyakit, manajemen data kucing, forum, rekomendasi produk, dan chatbot berbasis API Gemini.

2. LINGKUNGAN PENGEMBANGAN

Pengembangan sistem HealthCat dilakukan dalam lingkungan yang dirancang untuk mendukung pengembangan aplikasi web berbasis *Python* serta integrasi dengan model kecerdasan buatan. Seluruh proses pengembangan menggunakan *framework*, pustaka (*library*), dan layanan pihak ketiga yang saling terintegrasi untuk membangun, menguji, dan mengoperasikan sistem baik dari sisi frontend, backend, maupun model AI. Pada bagian ini dijelaskan perangkat lunak dan perangkat keras yang digunakan dalam proses pengembangan sistem secara menyeluruh.

2.1. Perangkat Lunak

Berikut adalah perangkat lunak yang diperlukan untuk menjalankan *website* HealthCat:

1. Visual Studio Code (VS Code), digunakan sebagai *code editor* utama dalam menulis dan mengelola seluruh berkas program. Editor ini mendukung ekstensi *Python*, *Jinja2*, *Flask*, serta integrasi *Git* untuk mempermudah proses pengembangan secara modular dan efisien.
2. MySQL, Sistem manajemen basis data yang digunakan adalah *MySQL*, yang menyimpan data pengguna, data kucing, hasil deteksi, artikel, komentar, dan aktivitas forum. Interaksi dengan basis data dikelola menggunakan *ORM* (*Object Relational Mapping*) melalui pustaka *SQLAlchemy*.
3. Python, Bahasa pemrograman utama yang digunakan adalah Python 3.x. Seluruh logika backend, integrasi model AI, dan interaksi dengan API eksternal ditulis dalam Python.
4. Flask Framework, Flask digunakan sebagai *framework* utama untuk pengembangan backend. Struktur sistem dibagi ke dalam beberapa *Blueprint* untuk modularitas, seperti *frontend*, *auth*, *admin*, dan *chatbot*.

5. TensorFlow dan Keras, Untuk membangun model klasifikasi penyakit kulit kucing, digunakan *TensorFlow* dan *Keras*. Model ini dilatih menggunakan dataset citra penyakit kulit, dan diintegrasikan ke dalam sistem backend untuk menghasilkan prediksi secara real-time.
6. Google Generative AI API (Gemini), API ini digunakan untuk mengintegrasikan fitur chatbot yang disebut *CatGPT*. Chatbot ini menerima pertanyaan dari pengguna dan merespons berdasarkan model bahasa generatif dari *Gemini*, melalui pustaka *google.generativeai*.
7. Google OAuth 2.0, sistem menyediakan fitur login dan registrasi menggunakan akun Google melalui layanan *OAuth 2.0*. Dengan fitur ini, pengguna dapat masuk ke dalam sistem tanpa perlu membuat akun manual, dan informasi seperti nama serta email akan otomatis diambil dari akun Google yang digunakan.
8. Bootstrap 5, HTML, CSS, dan JavaScript, untuk antarmuka pengguna (UI), sistem menggunakan *Bootstrap 5* dengan kombinasi *HTML*, *CSS*, dan *JavaScript* untuk memastikan tampilan yang responsif dan mudah diakses di berbagai perangkat.
9. Git dan Github, Sistem kontrol versi dilakukan dengan *Git* dan seluruh repositori pengembangan disimpan di *GitHub*, yang memungkinkan kolaborasi pengembangan dan pencatatan histori perubahan kode.
10. *Google Colab*, digunakan sebagai platform pelatihan model deteksi berbasis CNN. Meskipun berbasis web, Colab menyediakan akses ke perangkat keras seperti GPU *Tesla T4* atau *P100*, sehingga dianggap sebagai bagian dari lingkungan komputasi sistem ini.
11. Web Browser, Untuk menjalankan proyek dan melihat hasil antarmuka pengguna, digunakan web browser modern seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge. Browser digunakan untuk mengakses sistem secara lokal melalui alamat seperti http://localhost:5000 atau secara online jika telah dideploy.

2.2. Perangkat Keras

Berikut adalah perangkat keras yang diperlukan untuk menjalankan *website* HealthCat:

1. Laptop, dengan spesifikasi :
   - Prosesor: *Intel Core i5* atau setara
   - RAM: Minimum 8 GB
   - Penyimpanan: SSD 521 GB
   - Sistem Operasi: *Windows 10/11* atau *Linux Ubuntu*

3. PENGEMBANGAN MODEL DETEKSI PENYAKIT KULIT KUCING

Pengembangan model deteksi merupakan inti dari sistem HealthCat, di mana sistem ini mampu mengidentifikasi jenis penyakit kulit pada kucing dari citra yang diunggah oleh pengguna. Proses pengembangan model melibatkan pemilihan arsitektur *Convolutional Neural Network (CNN)*, preprocessing data, augmentasi, pelatihan, dan evaluasi performa model terhadap data uji. Hasil dari model ini digunakan secara langsung dalam sistem untuk memberikan prediksi penyakit dan tingkat keparahan.

3.1. Pendekatan Model

Sistem HealthCat menggunakan pendekatan *Convolutional Neural Network (CNN)* berbasis *transfer learning* untuk mendeteksi penyakit kulit pada kucing. Pendekatan ini memungkinkan sistem mengenali pola visual seperti tekstur dan warna dari gambar kulit kucing, yang sangat penting dalam proses klasifikasi. Dalam implementasinya, sistem menggunakan metode *feature extraction*, di mana bagian bawah model pretrained digunakan sebagai ekstraktor fitur tanpa dilatih ulang, sementara lapisan klasifikasi di atasnya dibangun dan dilatih khusus untuk kasus ini.

Model CNN yang digunakan dalam sistem *HealthCat* adalah *DenseNet121*, sebuah arsitektur *Convolutional Neural Network (CNN)* yang dikenal unggul dalam mengekstraksi fitur visual secara mendalam. DenseNet121 dipilih karena kemampuannya dalam menghubungkan setiap layer dengan layer sebelumnya melalui koneksi langsung (*dense connectivity*), sehingga memungkinkan aliran informasi dan gradien yang lebih baik selama pelatihan. Keunggulan ini menjadikannya sangat efektif untuk tugas klasifikasi gambar medis, termasuk deteksi penyakit kulit pada kucing. Model dilatih menggunakan *categorical_crossentropy* dan *optimizer Adam*, lalu disimpan dalam format .h5.

3.2. Dataset

3.2.1. Deskripsi Dataset

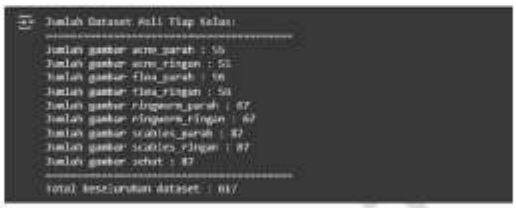
Dataset yang digunakan dalam sistem *HealthCat* terdiri dari kumpulan gambar yang merepresentasikan berbagai kondisi kulit pada kucing. Sumber data diperoleh dari berbagai referensi terbuka di internet, seperti *Kaggle*, *Google Images*, *Roboflow*, dan pengambilan secara langsung menggunakan kamera *smartphone*. Tujuan utama dari penyusunan dataset ini adalah untuk membangun sistem klasifikasi visual yang mampu mengenali penyakit kulit kucing secara otomatis dan akurat.

Setiap gambar dikelompokkan ke dalam sembilan kelas berdasarkan kombinasi antara jenis penyakit dan tingkat keparahannya. Penamaan kelas dilakukan secara manual, di mana nama folder disesuaikan langsung dengan label kelas. Dan sudah melewati tahap validasi d puskeswan. Kelas-kelas tersebut meliputi:

- scabies_ringan
- scabies_parah
- ringworm_ringan
- ringworm_parah
- acne_ringan
- acne_parah
- flea_ringan
- flea_parah
- sehat

Secara keseluruhan, dataset awal (*dataset_ori*) terdiri dari 617 gambar. Namun, terdapat ketidakseimbangan dalam jumlah gambar antar kelas. Distribusi yang tidak seimbang ini menjadi salah satu pertimbangan utama dalam proses *preprocessing*, yaitu dengan melakukan *upsampling* terhadap kelas minoritas untuk menghasilkan

dataset yang lebih proporsional sebelum pelatihan model dilakukan. Distribusi gambar untuk masing-masing kelas dapat di lihat pada Gambar 3.1.



Gambar 3. 1. Jumlah Dataset asli

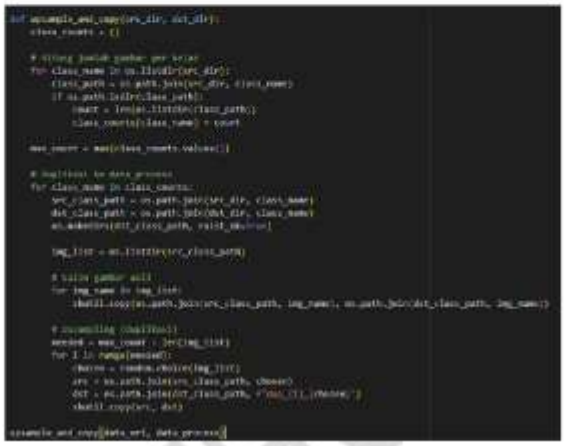
3.2.2. Penyeimbangan Data (*Upsampling*)

Berdasarkan distribusi awal pada dataset (*dataset_ori*), ditemukan adanya ketidakseimbangan jumlah gambar antar kelas. Beberapa kelas seperti sehat dan *scabies* memiliki jumlah gambar yang jauh lebih banyak dibandingkan kelas seperti *flea* dan *acne*. Ketidakseimbangan ini dapat menyebabkan bias pada model, di mana prediksi lebih condong ke kelas dengan frekuensi tinggi, sementara kelas minoritas cenderung diabaikan.

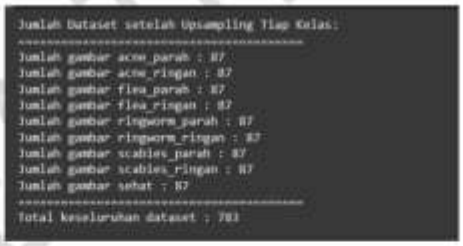
Untuk mengatasi hal ini, dilakukan proses *upsampling* pada kelas-kelas dengan jumlah data yang rendah. Teknik ini bertujuan untuk menyeimbangkan distribusi data antar kelas, agar model dapat belajar secara merata dari semua kelas yang ada. Proses dilakukan dengan menyalin gambar-gambar dari kelas minoritas secara acak hingga jumlahnya setara dengan kelas mayoritas. Dataset hasil upsampling disimpan dalam direktori baru bernama *dataset_process/* untuk memisahkannya dari dataset asli dan menjaga integritas data mentah.

Proses upsampling ini diotomatisasi menggunakan skrip Python. Skrip tersebut menyalin gambar dari folder kelas minoritas ke folder baru hingga memenuhi target jumlah yang ditentukan. Cuplikan dari kode upsampling dapat dilihat pada Gambar 3.2. sedangkan hasil

visualisasi distribusi dataset setelah proses upsampling disajikan pada Gambar 3.3.



Gambar 3. 2. Code *Upsampling*



Gambar 3. 3. Hasil Proses Upsampling

Meskipun proses *upsampling* telah meningkatkan jumlah data secara signifikan dan meratakan distribusi antar kelas, total data per kelas yang dihasilkan (87 gambar per kelas) masih tergolong terbatas untuk proses pelatihan model yang optimal. Oleh karena itu, pada tahap selanjutnya dilakukan proses *data augmentation* secara *online* dan disimpan ke dalam direktori *dataset_aug/* untuk menghasilkan dataset akhir yang lebih kaya secara variasi dan volume.

3.2.3. Augmentasi

Setelah proses *upsampling* dilakukan untuk menyeimbangkan jumlah gambar antar kelas, tahap berikutnya dalam persiapan data adalah melakukan augmentasi data. Teknik augmentasi ini diterapkan secara *offline*, yaitu dengan menghasilkan salinan baru gambar yang telah dimodifikasi menggunakan transformasi acak, lalu disimpan secara permanen dalam direktori terstruktur bernama *dataset_aug/*. Tujuan utama dari proses augmentasi ini adalah untuk memperbanyak variasi data pelatihan, meningkatkan generalisasi model, serta mengurangi risiko *overfitting* terhadap dataset yang relatif terbatas.

Proses augmentasi dilakukan menggunakan skrip Python yang menggabungkan pustaka *TensorFlow* dan *PIL (Python Imaging Library)*. Untuk setiap gambar asli, sistem menghasilkan sebanyak delapan gambar augmentasi dengan variasi transformasi, seperti rotasi acak sebesar ±10%, flipping horizontal, zoom sebesar 10%, dan resizing ke ukuran target 224×224 piksel. Selain itu, gambar asli juga disalin ulang dengan penamaan sistematis agar tidak tercampur dengan data mentah. Setiap hasil augmentasi disimpan ke dalam subfolder yang sesuai dengan nama kelasnya, seperti *scabies_ringan*, *ringworm_parah*, dan sebagainya.

Proses ini diotomatisasi melalui fungsi *augment_and_save()*, yang akan menelusuri seluruh direktori sumber, melakukan augmentasi untuk setiap gambar, dan menyimpan hasilnya ke direktori tujuan. Cuplikan dari kode program yang digunakan untuk proses augmentasi offline ini dapat dilihat pada Gambar 3.4 berikut.

Melalu tahap augmentasi ini setiap gambar menjadi 8 gambar augmentasi. Dengan demikian, total gambar untuk setiap kelas menjadi 783 gambar (87 asli + 696 hasil augmentasi). Jika dikalikan dengan total 9 kelas, maka jumlah keseluruhan dataset hasil augmentasi (dataset_aug) mencapai 7.047 gambar.

3.3. Pelatihan Model

Setelah proses augmentasi selesai dilakukan, data dilanjutkan ke tahap pembagian untuk keperluan pelatihan, validasi, dan pengujian. Pemisahan ini penting untuk memastikan bahwa model dapat diuji secara adil terhadap data yang belum pernah dilihat sebelumnya. Proses pelatihan dilakukan menggunakan *DenseNet121* yang telah dikonfigurasi dengan parameter dan teknik pelatihan yang sesuai. Seluruh proses

eksperimen dan pelatihan model dilaksanakan pada platform *Google Colab*, yang menyediakan akses GPU untuk mempercepat komputasi.

3.3.1. Pembagian Data

Setelah proses augmentasi offline selesai, seluruh data yang berada pada direktori dataset_aug/ disusun kembali dan disimpan dalam direktori baru bernama dataset_final/. Direktori ini berisi data akhir yang telah dibagi menjadi tiga subset utama, yaitu train, validation, dan test, yang masing-masing digunakan untuk proses pelatihan model, validasi selama pelatihan, serta pengujian performa model secara keseluruhan.

Proporsi pembagian data dilakukan dengan komposisi umum 70% untuk training set, 15% untuk validation set, dan 15% untuk testing set. Pembagian dilakukan secara terstratifikasi berdasarkan label kelas untuk memastikan distribusi kelas yang seimbang di semua subset. Distribusi jumlah data pada masing-masing subset dapat dilihat pada Gambar 3.4.

Gambar 3. 4. Visualisasi Pembagian Data

3.3.2. Augmentasi Online (Selama Pelatihan)

Meskipun proses augmentasi offline telah dilakukan sebelumnya, sistem tetap menerapkan *augmentasi online* saat pelatihan sebagai strategi untuk meningkatkan keragaman data dan memperkuat generalisasi model. knik ini memungkinkan setiap batch data yang dikirimkan ke model memiliki variasi transformasi yang berbeda-beda, sehingga model tidak hanya belajar dari satu bentuk gambar saja, tetapi juga dari modifikasi acak yang menyerupai variasi dunia nyata.

Transformasi dilakukan dengan menggunakan pipeline augmentasi berbasis *tf.keras.Sequential*, yang diterapkan secara langsung pada objek *tf.data.Dataset*. Beberapa transformasi yang digunakan meliputi:

- *Rescaling*: Normalisasi nilai piksel gambar ke rentang [0, 1].
- *Random Flip*: Membalik gambar secara horizontal secara acak.
- *Random Zoom*: Memperbesar gambar secara acak hingga 10%.
- *Random Rotation*: Memutar gambar secara acak hingga 10 derajat.
- *Random Translation*: Menggeser gambar secara acak baik secara horizontal maupun vertikal hingga 10%.

Pipeline augmentasi tersebut hanya diterapkan pada data pelatihan *(train_ds)* menggunakan *map()* dengan *training=True*, sedangkan untuk data validasi *(val_ds)* dan data pengujian *(test_ds)*, hanya dilakukan proses normalisasi tanpa transformasi tambahan, untuk menjaga konsistensi evaluasi performa model.

Selain itu, ketiga dataset *(train_ds, val_ds, dan test_ds)* juga dilengkapi dengan metode *prefetch()* yang berguna untuk mempercepat proses loading data saat pelatihan berlangsung. Implementasi dari proses ini dapat dilihat pada Gambar 3.5, yang menampilkan cuplikan kode program augmentasi online.

Gambar 3. 5. Kode Augmentasi Online dan Preprocessing Dataset

3.3.3. Konfigurasi Pelatihan

Sebelum proses pelatihan dimulai, ditentukan sejumlah parameter penting untuk mendefinisikan perilaku pelatihan. Parameter-parameter tersebut disusun dalam sebuah blok konfigurasi yang diimplementasikan dalam kode, seperti ukuran gambar input *(IMG_SIZE)*, jumlah batch *(BATCH_SIZE)*, jumlah *epoch*, nilai *learning rate*, dan *random seed* untuk memastikan reproduktifitas eksperimen. Nilai-nilai parameter tersebut dapat dilihat pada Gambar 3.6.

```
# Set Parameter
IMG_SIZE = (224, 224)
BATCH_SIZE = 32
SEED = 42
LR = 2E-05
EPOCHS = 20
```

Gambar 3. 6. Set Parameter

Arsitektur DenseNet121 digunakan sebagai *base model* tanpa bagian klasifikasinya (include_top=False), dan bobot awalnya dimuat dari *ImageNet*. Selanjutnya ditambahkan beberapa lapisan untuk menyesuaikan dengan klasifikasi multi-kelas pada dataset penyakit kulit kucing. Model dikompilasi menggunakan *Adam Optimizer* dan

fungsi loss categorical_crossentropy, dengan metrik akurasi. Rangkaian arsitektur lengkap dapat dilihat pada Gambar 3.8.

Gambar 3. 7. Model Arsitektur DenseNet121

Selama proses pelatihan, beberapa *callback* digunakan untuk mengoptimalkan hasil pelatihan, termasuk:

- *EarlyStopping* untuk menghentikan pelatihan jika *val_loss* tidak membaik selama 10 *epoch*.
- *ModelCheckpoint* untuk menyimpan model terbaik berdasarkan akurasi validasi tertinggi.
- *ReduceLROnPlateau* untuk menurunkan *learning rate* secara adaptif jika tidak ada perbaikan.

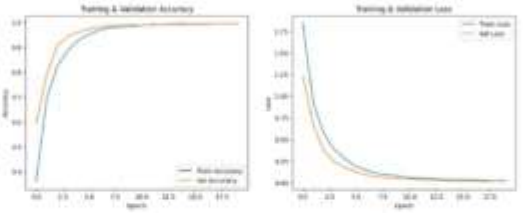
Proses pelatihan dilakukan selama 20 *epoch* dan dieksekusi menggunakan GPU di platform *Google Colab*. Lama waktu pelatihan dicatat dan disajikan sebagai bagian dari evaluasi performa model. Berikut code pelatihannya dapat dilihat pada Gambar 3.8.

Gambar 3. 8. Code Pelatihan

3.3.4. Evaluasi Performa Model

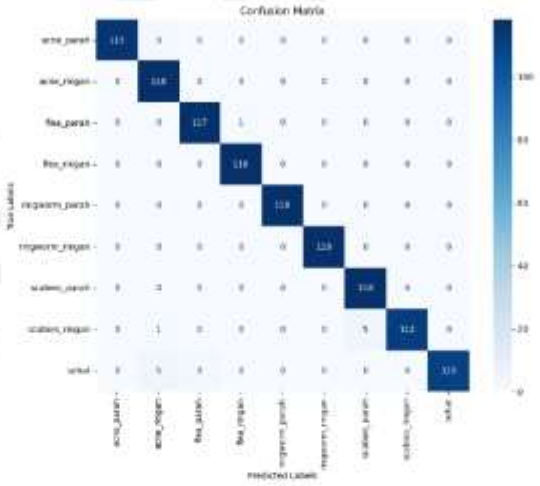
Setelah proses pelatihan model selesai dilakukan menggunakan arsitektur *DenseNet121*, evaluasi performa dilakukan untuk menilai sejauh mana model mampu mengklasifikasikan gambar secara akurat dan konsisten. Evaluasi dilakukan melalui dua pendekatan, yaitu analisis kurva pembelajaran (*learning curves*) dan pengujian langsung menggunakan data uji (*test set*) yang belum pernah dilihat model sebelumnya.

Pertama, performa selama pelatihan dianalisis melalui visualisasi *akurasi* dan *loss* pada data pelatihan dan validasi. Grafik *Training Accuracy* dan *Validation Accuracy* menunjukkan tren peningkatan yang stabil pada tiap epoch, sementara *Training Loss* dan *Validation Loss* menurun secara konsisten hingga mendekati titik konvergensi. Tidak ditemukan indikasi *overfitting* atau *underfitting* yang signifikan, menandakan bahwa model berhasil belajar secara optimal dari data yang diberikan. Visualisasi kurva ini ditampilkan pada Gambar 3.9.



Gambar 3. 9. Grafik akurasi dan loss pada data pelatihan dan validasi

Selanjutnya, performa model dievaluasi secara kuantitatif menggunakan data pengujian. Metode evaluasi yang digunakan meliputi *confusion matrix* dan *classification report*. *Confusion matrix* digunakan untuk mengidentifikasi akurasi prediksi antar kelas, memperlihatkan jumlah prediksi benar dan salah untuk masing-masing kelas. Dari Gambar 3.10, terlihat bahwa sebagian besar kelas berhasil diprediksi dengan cukup baik



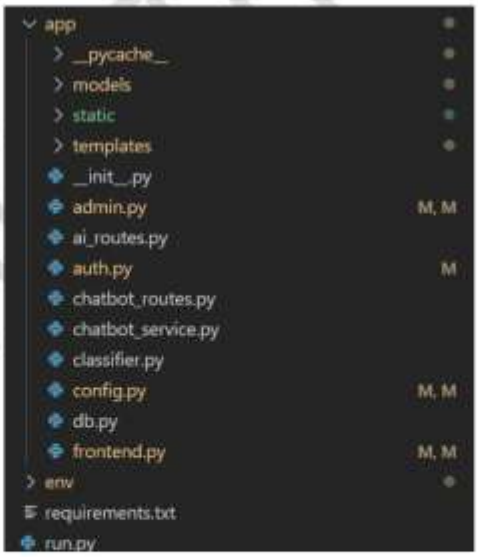
Gambar 3. 10. Confusion Matrix Hasil Data Test

3.4. Implementasi Model

Setelah model klasifikasi penyakit kulit kucing berhasil dilatih dan dievaluasi, langkah selanjutnya adalah mengintegrasikan model tersebut ke dalam aplikasi berbasis web menggunakan framework *Flask*. Implementasi ini memungkinkan pengguna akhir untuk mengakses layanan deteksi secara langsung melalui antarmuka web yang interaktif. Sistem akan memproses gambar yang diunggah pengguna, menjalankan prediksi menggunakan model *DenseNet121*, dan menampilkan hasil deteksi secara real-time, termasuk rekomendasi produk pengobatan dan riwayat deteksi.

3.4.1. Struktur Proyek Flask

Struktur proyek Flask dirancang secara modular agar mudah dikembangkan, dipelihara, dan diskalakan. Gambar 3. berikut menggambarkan struktur direktori aplikasi HealthCat:



Gambar 3. 11. Struktur Project

3.4.2. Memuat Model CNN

Setelah proses pelatihan selesai, model CNN yang telah disimpan dalam format .h5 dimuat ke dalam sistem menggunakan file classifier.py. Untuk efisiensi, pemanggilan dilakukan dengan pendekatan *lazy loading*, yaitu model hanya dimuat satu kali saat pertama kali dibutuhkan. Gambar 3.12 berikut menampilkan potongan kode untuk pemanggilan dan loading model CNN.

Gambar 3. 12 Kode Pemanggilan Model CNN

3.4.3. Preprocessing Gambar

Setiap gambar yang dikirim oleh pengguna akan diproses melalui fungsi *preprocess_image()*. Fungsi ini akan dilakukan *resize* gambar ke dimensi (224, 224) piksel, Mengubah gambar ke array numerik, Menyesuaikan dimensi input model, Melakukan normalisasi sesuai skema preprocessing dari DenseNet121. Fungsi predict() kemudian dipanggil untuk memproses input dan mengembalikan prediksi. Berikut kode *preprocessing* dapat dilihat pada Gambar 3.13.

Gambar 3. 13. Kode Preprocesing

3.4.4. Proses Deteksi Penyakit *(Endpoint /detect)*

Setelah model *DenseNet121* berhasil dimuat ke dalam sistem, proses deteksi gambar penyakit kulit kucing diakses melalui *endpoint /detect*. Endpoint ini menerima file gambar dari pengguna yang telah login, kemudian melakukan serangkaian proses seperti:

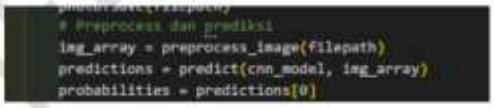
a. Validasi dan Penyimpanan Gambar

Saat permintaan dikirim, sistem pertama-tama memverifikasi keberadaan file gambar dan ID kucing. Jika valid, gambar disimpan di folder tertentu menggunakan penamaan berbasis waktu untuk menghindari duplikasi. Seperti pada Gambar 3.14.

Gambar 3. 14. Kode Route Deteksi

b. Preprocessing dan Prediksi

Gambar yang sudah disimpan akan diproses melalui *preprocess_image()* untuk resize ke ukuran (224, 224), normalisasi nilai pixel *(preprocessing DenseNet)*, konversi ke format tensor. Selanjutnya, prediksi dilakukan menggunakan model CNN yang telah dimuat. Dapat dilihat pada Gambar 3.15.



Gambar 3. 15. Preprocesing Route Deteksi

c. Menentukan Kelas Penyakit

Probabilitas tertinggi diambil dari output model dan dibandingkan dengan ambang batas (threshold = 0.33). Jika lebih besar dari threshold, hasil dianggap valid. Seperti pada Gambar 3.16.

```
    # Preprocess dan prediksi
    img_array = preprocess_image(filepath)
    predictions = predict(cnn_model, img_array)
    probabilities = predictions[0]
```

Gambar 3. 16. Kode Menentukan Kelas Penyakit

d. Simpan Deteksi ke Database

Jika hasil prediksi valid, sistem akan mencari data penyakit dari database (Penyakit.query.get), selanjutnya menyimpan metadata deteksi ke tabel Deteksi dan menyimpan hasil klasifikasi ke tabel Hasil. Dapat dilihat di Gambar 3.17.

```
if disease:# Simpan deteksi & hasil ke database
    detection = Deteksi(
        img_url=filename,
        id_user=current_user.id_user,
        id_kucing=cat_id
    )
    db.session.add(detection)
    db.session.flush()
    hasil = Hasil(
        id_deteksi=detection.id_deteksi,
        id_penyakit=disease.id_penyakit,
        probabilitas=max_probability
    )
    db.session.add(hasil)
    db.session.commit()
```

Gambar 3. 17. Kode Simpan Deteksi

e. Mengembalikan Respons ke Pengguna

Setelah data disimpan, sistem mengembalikan hasil deteksi dalam bentuk JSON, termasuk Nama penyakit, Kategori (ringan/parah), Probabilitas, Rekomendasi pengobatan. Jika prediksi **tidak valid**, sistem akan memberikan pesan bahwa gambar tidak dikenali, lalu menghapus gambar tersebut dari direktori. Dapat dilihat pada Gambar 3.18.

Gambar 3. 18. Kode Respons JSON (valid & tidak valid)

4. Pengembangan Sistem

4.1. Deteksi Penyakit Kulit Pada Kucing

Fitur utama dari sistem HealthCat adalah modul deteksi penyakit kulit kucing, yang diimplementasikan dalam bentuk 4 buah slide interaktif untuk memandu pengguna secara bertahap. Keempat slide ini disusun secara terstruktur pada halaman templates/frontend/detection.html, dan dikendalikan oleh skrip interaktif yang tertulis pada static/fomtend/js/deteksi.js. berikut penjelasan tiap slidenya:

a. Slide pertama

Pada slide pertama, pengguna diberikan penjelasan visual mengenai langkah-langkah deteksi. Konten ini penjelasan bagaimana cara menggunakan fitur deteksi. Adapun tombol mulai deteksi untuk melanjutkan ke tahap selanjutnya. Berikut kode html dapat dilihat pada Gambar 4.1.

Gambar 4. 1. Kode Slide Tutorial pada halaman Deteksi

b. Slide Kedua

Slide kedua menampilkan daftar dropdown atau kartu kucing yang telah didaftarkan oleh pengguna. Pengguna wajib memilih salah satu kucing untuk dikaitkan dengan hasil deteksi. Jika pegguna belum mempunyai kucing yang terdaftar di sistem, maka penggunaka wajib

daftarkan terlebih dahulu. Berikut kode html dapat dilihat pada Gambar 4.2. dan kode Js di Gambar 4.3.

Gambar 4. 2. Kode HTML slide pilih kucing

Gambar 4. 3. Js Pemilihan Kucing

c. Slide Ketiga

Pada slide *3*, pengguna diminta mengambil gambar langsung via kamera atau mengunggah gambar dari galeri, yang akan digunakan untuk proses deteksi yang dikirimkan kepada rute /detect. Berikut kode Html dapat dilihat pada Gambar 4.4. dan js pada Gambar 4.5.

Gambar 4. 4. Slide upload foto kulit kucing

Gambar 4. 5. Js Slide proses deteksi

d. Slide Keempat

Setelah pengguna mengunggah gambar kulit kucing dan sistem selesai melakukan analisis menggunakan model CNN, sistem akan menampilkan slide keempat, yaitu hasil deteksi. Slide ini merupakan bagian akhir dari proses diagnosis, yang menyajikan informasi berisi gambar yang dideteksi, detail kucing yg dideteksi, Hasil deteksi, daran rekomendasi penanganan, dan tombol lihat produk, deteksi baru, serta kembali. Berikut kode bagian hasil dapat dilihat pada Gambar 4.6.

Gambar 4. 6. Kode Slide Hasil Deteksi

Adapun tombol lihat produk akan mengarahkan ke halaman rekomendasi produk sesuai hasil penyakit dan tingkat keparahannya. Berikut kode js untuk tombol lihat produk dapat dilihat pada Gambar 4.7.

Gambar 4. 7. kode Tombol lihat produk

4.2. Chatbot menggunakan API GEMINI

Chatbot dalam sistem HealthCat menggunakan API dari *Gemini* (Google AI) untuk menjawab pertanyaan seputar kesehatan dan perawatan kucing.

Tujuan utama integrasi ini adalah memberikan respon berbasis natural language yang informatif, relevan, dan terfokus hanya pada topik kesehatan kucing. Pengembangan chatbot ini terdiri dari beberapa komponen utama:

File	Fungsi
config.py	Menyimpan *API Key* dan *Endpoint URL* dari Gemini
chatbot_service.py	Kelas utama GeminiChatbot yang menangani request ke API
chatbot_routes.py	Route Flask (/chatbot) untuk menerima input user dan mengirim response
Frontend JS & HTML	Komponen UI untuk input pertanyaan dan menampilkan balasan

Pada file config.py, terdapat dua konfigurasi penting, bisa dilihat pada Gambar 4.8.

Gambar 4. 8. Gemini API

File ini mendefinisikan kelas GeminiChatbot, dengan fungsi utama get_response(user_message). Berikut penjelasan alur kerjanya:

Gambar 4. 9. Kelas GeminiChatbot

- **system_prompt** diatur untuk *membatasi ruang lingkup jawaban hanya pada topik seputar kucing.*
- Ini mencegah chatbot menjawab topik di luar konteks dan menjaga relevansi.

Fungsi get_response, dapat dilihat pada Gambar 4.10.

Gambar 4. 10. Kode Fungsi get_response

- API Gemini menerima input dalam format JSON.
- Prompt akhir digabung dari **system prompt + pertanyaan user**.
- Balasan dari Gemini akan diambil dari response JSON bagian:

Setelah layanan chatbot berhasil dikonfigurasi melalui kelas GeminiChatbot, sistem HealthCat menyediakan endpoint khusus berbasis Flask untuk menghubungkan permintaan dari pengguna ke layanan Gemini API. Rute tersebut diatur dalam file chatbot_routes.py dan menggunakan blueprint dengan nama chatbot_bp untuk menjaga struktur modular aplikasi. Endpoint utama yang digunakan adalah POST /chatbot, yang berfungsi untuk menerima pesan dari pengguna dalam bentuk JSON dan mengembalikan respons dari chatbot berdasarkan hasil interaksi dengan Gemini API. Seperti dalam Gambar 4.11.

```python
from flask import Blueprint, request, jsonify, current_app
from .chatbot_service import get_chatbot_response
import logging

chatbot_bp = Blueprint('chatbot', __name__)

@chatbot_bp.route('/chatbot', methods=['POST'])
def chatbot_response():
    try:
        data = request.get_json()
        if not data or 'message' not in data:
            return jsonify({
                'status': 'error',
                'message': 'Pesan tidak boleh kosong'
            }), 400

        user_message = data['message'].strip()
        if not user_message:
            return jsonify({
                'status': 'error',
                'message': 'Pesan tidak boleh kosong'
            }), 400
        # Dapatkan response dari Gemini
        bot_response = get_chatbot_response(user_message)
        return jsonify({
            'status': 'success',
            'response': bot_response
        })
    except Exception as e:
        current_app.logger.error(f"Chatbot error: {str(e)}")
        return jsonify({
            'status': 'error',
            'message': 'Terjadi kesalahan pada server. Silakan coba lagi.'
        }), 500
```

Gambar 4. 11. Kode Rute Chatbot

Dan rute akan dipanggil dalam tampilan frontend, berikut potongan kode chatbot dapat dilihat pada Gambar 4.12. dan js untuk memanggil rute /chatbot pada Gambar 4.13.

Gambar 4. 12. Kode Tampilan Chatbot

```
autoResizeTextarea();

// Set status processing
setProcessingState(true);
showTypingIndicator();

// Kirim ke backend
fetch("/chatbot", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    message: message,
  }),
})
```

Gambar 4. 13. Kode Js Chatbot

4.3. Login & Registrasi

Sistem autentikasi di-*implementasi* menggunakan Blueprint auth.py. Proses login dan registrasi manual memanfaatkan form HTML dan route Flask seperti pada Gambar 4.14.

```
@auth_bp.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        nama = request.form.get('nama')
        email = request.form.get('email')
        password = request.form.get('password')

        # Hash password dengan bcrypt
        hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')

        if Users.query.filter_by(email=email).first():
            flash('Email sudah terdaftar', 'error')
        else:
            new_user = Users(nama=nama, email=email, password=hashed_password)
            db.session.add(new_user)
            db.session.commit()
            flash('Registrasi berhasil! Silakan login.', 'success')
            return redirect(url_for('auth.login'))

    return render_template('register.html')
```

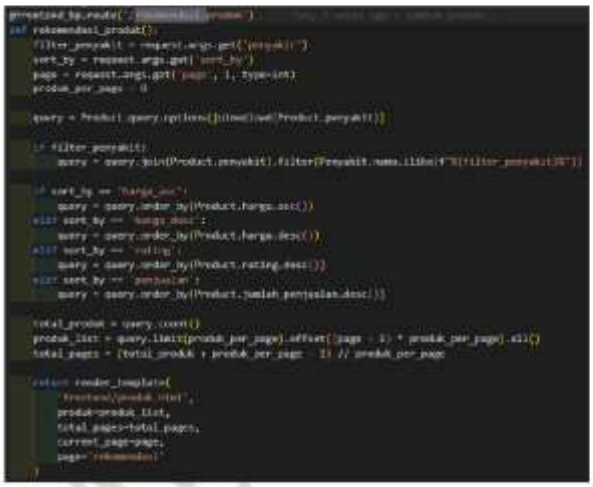
Gambar 4. 14. Kode Rute Register

Sedangkan untuk **Google OAuth 2.0**, sistem memanfaatkan library OAuth seperti Authlib, Pendekatan ini memungkinkan pengguna melakukan autentikasi baik melalui email/password ataupun akun Google, dengan sesi login yang dikelola oleh *Flask-Login*.

Gambar 4. 15. Kode Rute Login dengan Google

4.4. Rekomendasi Produk

Setelah selesai deteksi, pengguna diarahkan ke halaman /rekomendasi-produk yang menghasilkan daftar produk berdasarkan penyakit dan tingkat keparahan. Data produk diambil dari tabel Produk dan difilter:



Gambar 4. 16. Kode Rute Rekomendasi Produk

4.5. Riwayat Deteksi

Riwayat hasil deteksi disimpan pada tabel Deteksi dan Hasil setelah setiap permintaan berhasil. Pada frontend, setiap baris riwayat mencantumkan thumbnail gambar, tanggal deteksi, hasil diagnosis, dan tombol unduh PDF atau detail.

```python
@frontend_bp.route('/riwayat')
@login_required
def riwayat_deteksi():
    kucing = request.args.get('kucing')
    penyakit = request.args.get('penyakit')
    kategori = request.args.get('kategori')
    tanggal_dari = request.args.get('tanggal_dari')
    tanggal_sampai = request.args.get('tanggal_sampai')

    query = db.session.query(
        Deteksi.id_deteksi,
        Kucing.nama_kucing.label("kucing_nama"),
        Deteksi.img_url,
        Penyakit.nama.label("penyakit_nama"),
        Penyakit.kategori,
        Penyakit.rekomendasi,
        Hasil.probabilitas,
        Deteksi.created_at
    ).join(Users, Deteksi.id_user == Users.id_user)\
     .join(Kucing, Deteksi.id_kucing == Kucing.id_kucing)\
     .join(Hasil, Deteksi.id_deteksi == Hasil.id_deteksi)\
     .join(Penyakit, Hasil.id_penyakit == Penyakit.id_penyakit)\
     .filter(Deteksi.id_user == current_user.id_user)
```

Gambar 4. 17. Kode Rute Riwayat

Lampiran 5 Sertifikat HKI

REPUBLIK INDONESIA
KEMENTERIAN HUKUM

SURAT PENCATATAN CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC00202507619G, 26 Juni 2025
Pencipta		
Nama	:	**Fadila Rizka Nur Aminah, Muhammad Fikri Hidayattullah, S.T., M.Kom. dkk**
Alamat	:	Jl. Kerupuk, RT 02/RW 02, Desa Suradadi., Suradadi, Kab. Tegal, Jawa Tengah, 52180
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**
Alamat	:	Jalan Mataram No. 9, Pesuningan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**HealthCat: Sistem Deteksi Penyakit Kulit Pada Kucing Menggunakan Convolutional Neural Network Berbasis Website"**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	26 Juni 2025, di Kota Tegal
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	:	000916457

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Agung Damarsasongko,SH.,MH.
NIP. 196912261994031001

Balai Besar Sertifikasi Elektronik

Disclaimer:
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah ditanda tangani secara elektronik menggunakan sigel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

Lampiran 6 Lembar Bimbingan

D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama	:	**Fadila Rizka Nur Aminah**
NIM	:	**21090037**
No. Ponsel	:	**08895638802**
Judul TA	:	**Pengembangan Sistem Deteksi Penyakit Kulit pada Kucing Peliharaan Menggunakan *Convolutional Neural Network* Berbasis *Website***
Dosen Pembimbing I	:	**Muhammad Fikri Hidayatullah, S.T., M.Kom.**

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	28/4 2025	Konsep	- Komponen model arsitektur - Dokumen fitur hasil uji coba	
2.	28/5	Progress	- Kenapa epoch 60? - Kenapa setting hyper parameter nilainya segini dan 7.	
7.	18/6	Aplikasi HKI	- Siapkan manual book + dokumen	

4.	23/6	HKI	Pokumen	
5.	24/6	Lapom	Lengkapi	
6	26/6	Lapor	Lengkapi	
7.	7/7	Lapom	Administrasi	
8.	8/7	Lapor	ACC	

Tegal,
Dosen Pembimbing I

Muhammad Fikri Hidayattullah, S.T., M.Kom.
NIPY. 09.017.337

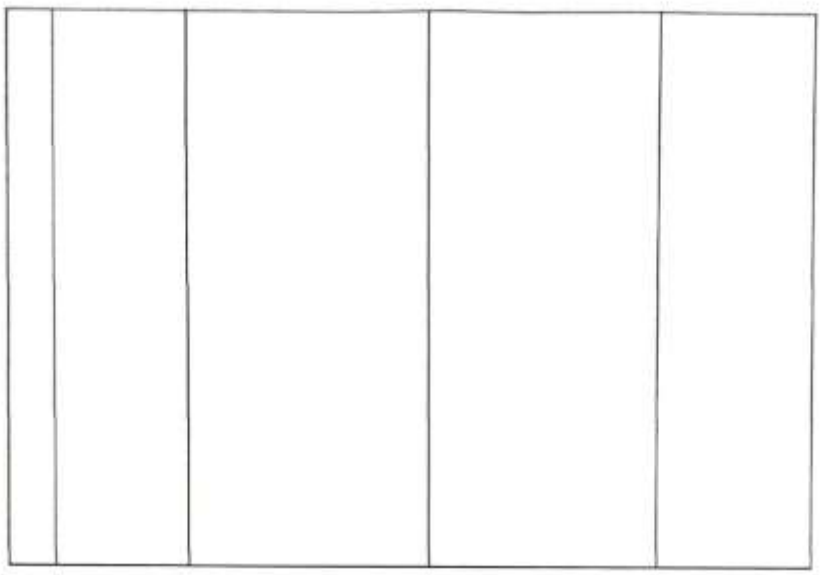
D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : **Fadila Rizka Nur Aminah**
NIM : **21090037**
No. Ponsel : **08895638802**
Judul TA : **PENGEMBANGAN SISTEM DETEKSI PENYAKIT KULIT PADA KUCING PELIHARAAN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK* BERBASIS *WEBSITE***
Dosen Pembimbing II : **Mirza Alim Mutasodirin, S.Kom., M.Kom.**

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	18/03/2025	Problem Statement	Perbaikan Problem statement	
2.	21/03/2025	Tujuan dan Manfaat	Perbaikan Tujuan dan Manfaat	
3.	21/04/2025	Data Penelitian	Perbaikan Data	
4.	23/04/2025	literature Review	Perbaikan literatur Review	

5.	01/05/2025	website	perbaikan web	
6	09/05/2025	HPO	Review HPO	
7	23/05/2025	unify the code	jadikan dalam 1File 1code	
8	31/05/2025	HPO	melanjutkan running project gantt epoch 20	
9	11/06/2025	Test case	buat Dokumen test case	
10.	23/06/2025	Final Demo Website	lanjutkan HKI dan laporan	



Tegal,
Dosen Pembimbing II

Mirza Alim Mutasodirin, S.Kom , M.Kom
NIPY. 03.023.534

Lampiran 7 Surat Permohonan Observasi dan Wawancara

POLITEKNIK HARAPAN BERSAMA

Sarjana Terapan Teknik Informatika

Nomor : 48.03/TI.PHB/IV/2025
Lampiran : -
Hal : Permohonan Observasi
Kepada Yth. : **Kepala UPTD PUSKESWAN Tegal**
Di Tempat

Dengan hormat, mahasiswa dengan identitas berikut ini:

nama : Fadila Rizka Nur Aminah
NIM : 21090037
prodi : Sarjana Terapan Teknik Informatika

Bermaksud melakukan penelitian untuk keperluan Skripsi dengan judul "**Sistem Deteksi Penyakit Kulit pada Kucing Peliharaan Menggunakan Convolutional Neural Network Berbasis Website**". Kami memohon Bapak/Ibu memberikan izin kepada mahasiswa yang bersangkutan agar dapat memperoleh data, keterangan, dan bahan yang diperlukan.

Demikian permohonan ini disampaikan, atas perhatian kami ucapkan terima kasih.

Tegal, 14 April 2025
Ka. Prodi S.Tr. Teknik Inormatika,

Dyah Apriliani, S.T., M.Kom
NIPY : 09.01.[illegible]25

Jl. Mataram No. 9 Kota Tegal 52143, Jawa Tengah, Indonesia.
(0283)352000
informatika@politegal.ac.id
politegal.ac.id