

# **LAPORAN KERJA PRAKTIK**

## **RANCANG BANGUN APLIKASI PENDAFTARAN PELAYANAN BERBASIS WEB MENGGUNAKAN *NODE.JS* DAN *POSTGRESQL* DI PUSKESWAN PEMBANTU ARJASARI**

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan pada  
Mata Kuliah TIF335 Kerja Praktik

Oleh:

**RISKA NURHAYAN/ 301210009**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG  
2025**

**LEMBAR PENGESAHAN**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**RANCANG BANGUN APLIKASI PENDAFTARAN PELAYANAN  
BERBASIS WEB MENGGUNAKAN *NODE.JS* DAN *POSTGRESQL*  
DI PUSKESWAN PEMBANTU ARJASARI**

oleh:  
RISKA NURHAYAN/ 301210009

disetujui dan disahkan sebagai  
**LAPORAN KERJA PRAKTIK**

Bandung, Februari 2025  
Koordinator Kerja Praktik  
Program Studi Teknik Informatika

Yusuf Muharam, S.Kom., M.Kom.  
NIK: 04104820003

**LEMBAR PENGESAHAN**

**PUSKESWAN PEMBANTU ARJASARI**

**RANCANG BANGUN APLIKASI PENDAFTARAN PELAYANAN  
BERBASIS WEB MENGGUNAKAN *NODE.JS* DAN *POSTGRESQL*  
DI PUSKESWAN PEMBANTU ARJASARI**

oleh:  
RISKA NURHAYAN/ 301210009

disetujui dan disahkan sebagai  
**LAPORAN KERJA PRAKTIK**

Bandung, Februari 2025  
Koordinator Puskesmas

Drh. Nur Azizah Rahman  
NIP. 199412182023212020

## ABSTRAKSI

Sistem informasi pendaftaran berbasis web dirancang untuk meningkatkan efisiensi layanan Puskesmas Pembantu (Pustu) Arjasari. Sistem ini menggantikan metode manual yang lambat dan rentan kesalahan, dengan fitur pendaftaran online, pengelolaan data pasien, dan pencatatan riwayat medis hewan. Menggunakan metode *waterfall*, sistem ini mendukung pelayanan kesehatan hewan lebih cepat, akurat, dan terstruktur.

Pengembangan ini bertujuan untuk merancang aplikasi pendaftaran berbasis web untuk mendukung operasional Puskesmas Pembantu (Pustu) Arjasari. Dengan menyederhanakan proses administrasi manual seperti pendaftaran pasien, pengelolaan data hewan, dan pelacakan riwayat konsultasi. Fitur utama sistem mencakup layanan pendaftaran pasien, pengelolaan nomor antrian, serta manajemen data pasien dan konsultasi. Selain itu, sistem juga dirancang untuk admin dapat mengakses sistem dan bertugas mengelola data secara efisien. Dikembangkan dengan menggunakan metodologi *Waterfall*, pengembangan ini mencakup analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Perancangan dibuat dengan menggunakan *Unified Modeling Language* (UML) untuk memvisualisasikan sistem, termasuk *use case*, *activity*, *class diagram*, serta *wireframe* untuk antarmuka yang *user-friendly*. Teknologi yang digunakan dalam pengembangan adalah *HTML*, *CSS*, *JavaScript*, *framework Express.js*, dan *database postgresSQL*, memastikan aplikasi berfungsi secara optimal, aman, dan sesuai kebutuhan pengguna.

Aplikasi pendaftaran berbasis web untuk Puskesmas Pembantu (Pustu) Arjasari berhasil dirancang dan dikembangkan untuk meningkatkan efisiensi administrasi dengan fitur utama meliputi pendaftaran pasien, pengelolaan data hewan, riwayat konsultasi, dan otorisasi admin. Hasil pengembangan menunjukkan aplikasi ini mampu meningkatkan efisiensi operasional dan memberikan pengalaman pengguna yang lebih baik dalam

melakukan pendaftaran. Aplikasi ini juga memiliki potensi untuk dikembangkan lebih lanjut dengan fitur tambahan dimasa mendatang, seperti integrasi pembayaran online dan rekam medis dari laporan statistik mendalam.

**Kata kunci:** Aplikasi Pendaftaran, Efisiensi Administrasi, Pelayanan Kesehatan Hewan, Puskesmas Pembantu, Sistem Informasi.

## KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat ALLAH SWT yang selalu menyertai setiap langkah penulis. Atas segala berkah dan Rahmat-Nya kepada penulis sehingga mampu menyelesaikan laporan kerja praktik ini. Laporan ini disusun sebagai salah satu syarat untuk memenuhi kelulusan untuk mata kuliah Kerja Peraktik yang bertempat di Pusat Kesehatan Hewan Pembantu Arjasari yang bertempat di kantor BPPPK Kp. Cihonje Rt.02/Rw.01, Desa Wargaluyu, Kec. Arjasari. Kerja Praktik ini merupakan salah satu mata kuliah wajib yang harus diselesaikan oleh Mahasiswa program studi Teknik Informatika di Universitas Bale Bandung. Laporan kerja praktik ini berjudul “Rancang Bangun Aplikasi Pendaftaran Pelayanan Berbasis Web Menggunakan *Node.js* Dan *postgreSQL* Di Puskesmas Pembantu Arjasari”.

Akhir kata penulis ucapkan banyak terima kasih kepada Dosen, Pembimbing yang telah membantu dan juga telah membimbing dalam setiap proses pembuatan proposal ini. Tak lupa juga penulis ucapkan terima kasih kepada orang tua yang telah memberikan doa dan dukungan, serta teman-teman di kampus yang telah memberikan motivasi kepada penulis.

Bandung, 07 Januari 2024  
Penyusun

Riska Nurhayan

## DAFTAR ISI

LEMBAR PENGESAHAN PROGRAM STUDI TEKNIK INFORMATIKA...	i
LEMBAR PENGESAHAN PUSKESWAN PEMBANTU ARJASARI .....	ii
ABSTRAKSI.....	iii
KATA PENGANTAR .....	v
DAFTAR ISI .....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR .....	ix
DAFTAR LAMPIRAN .....	xii
BAB I PENDAHULUAN .....	1
I.1 Latar Belakang.....	1
I.2 Lingkup .....	3
I.3 Tujuan.....	4
BAB II LINGKUNGAN KERJA PRAKTIK .....	5
II.1 Struktur Organisasi .....	5
II.2 Lingkup Pekerjaan .....	6
II.3 Deskripsi Pekerjaan .....	7
II.4 Jadwal Kerja .....	10
BAB III TEORI PENUNJANG KERJA PRAKTIK .....	13
III.1 Teori Penunjang .....	13
III.2 Peralatan Pembangunan .....	15
BAB IV PELAKSANAAN KERJA PRAKTIK .....	36
IV.1 Input.....	36
IV.2 Proses.....	38
IV.2.1. Analisis Kebutuhan.....	38
IV.2.2. Perancangan Perangkat Lunak .....	40
IV.2.3. Implementasi Antarmuka .....	70
IV.2.4. Pengujian .....	77
IV.2.5. Pelaporan Hasil Kerja Praktik.....	83
IV.3 Pencapaian Hasil.....	84
BAB V PENUTUP .....	85

V.1	Kesimpulan Dan Saran Mengenai Pelaksanaan .....	85
V.1.1.	Kesimpulan Pelaksanaan Kerja Praktik .....	85
V.1.2.	Saran Pelaksanaan Kerja Praktik.....	87
V.2	Kesimpulan Dan Saran Mengenai Substansi .....	87
V.2.1.	Kesimpulan Mengenai Substansi .....	87
V.2.2.	Saran Mengenai Substansi .....	88
DAFTAR PUSTAKA .....		xiii



## DAFTAR TABEL

Tabel II. 1 Jadwal Kerja Praktik .....	10
Tabel III. 1 Atribut HTML.....	23
Tabel III. 2 Atribut <i>Use case Diagram</i> .....	30
Tabel III. 3 Atribut <i>Activity diagram</i> .....	31
Tabel III. 4 Atribut <i>Class Diagram</i> .....	32
Tabel IV. 1 Kebutuhan Perangkat Keras .....	39
Tabel IV. 2 Minimum Kebutuhan Perangkat Keras .....	39
Tabel IV. 3 Kebutuhan Perangkat Lunak .....	40
Tabel IV. 4 Struktur Tabel Admin .....	59
Tabel IV. 5 Struktur Tabel <i>Consultation</i> .....	60
Tabel IV. 6 Struktur Tabel <i>Users</i> .....	61
Tabel IV. 7 Pengujian <i>Black Box User</i> .....	78
Tabel IV. 8 Pengujian <i>Black Box Admin</i> .....	79

## DAFTAR GAMBAR

Gambar II. 1 Struktur Organisasi Puskesmas Pembantu (Pustu) Arjasari..	5
Gambar III. 1 Tampilan <i>draw.io</i> .....	16
Gambar III. 2 Tampilan <i>Figma</i> .....	17
Gambar III. 3 Tampilan <i>PgAdmin</i> .....	18
Gambar III. 4 Tampilan <i>Visual Studio Code</i> .....	19
Gambar III. 5 Tampilan <i>Microsoft Edge</i> .....	22
Gambar III. 6 Tampilan <i>Google Chrome</i> .....	22
Gambar III. 7 Arsitektur <i>CSS</i> .....	24
Gambar III. 8 Arsitektur <i>Morgan</i> .....	25
Gambar III. 9 Arsitektur <i>Bcrypt</i> .....	25
Gambar III. 10 Arsitektur <i>postgreSQL</i> .....	26
Gambar III. 11 Arsitektur <i>MySQL</i> .....	27
Gambar III. 12 Arsitektur <i>Node.js</i> .....	28
Gambar III. 13 Arsitektur <i>Express.js</i> .....	28
Gambar III. 14 <i>Bootstrep</i> .....	29
Gambar III. 15 Arsitektur Metode <i>Waterfall</i> .....	34
Gambar IV. 1 <i>Use case User</i> .....	41
Gambar IV. 2 <i>Use case Admin</i> .....	42
Gambar IV. 3 <i>Activity diagram Login</i> .....	43
Gambar IV. 4 <i>Activity diagram</i> Buat Akun Baru .....	44
Gambar IV. 5 <i>Activity diagram</i> Pendaftaran .....	45
Gambar IV. 6 <i>Activity diagram Home</i> .....	46
Gambar IV. 7 <i>Activity diagram Profile</i> .....	47
Gambar IV. 8 <i>Activity diagram Update Profile</i> .....	48
Gambar IV. 9 <i>Activity diagram Riwayat</i> .....	49
Gambar IV. 10 <i>Activity diagram Logout</i> .....	50
Gambar IV. 11 <i>Activity diagram Login Admin</i> .....	51
Gambar IV. 12 <i>Activity diagram Home Admin</i> .....	52
Gambar IV. 13 <i>Activity diagram Approved Rejected Consultation</i> .....	53
Gambar IV. 14 <i>Activity diagram Manage Admin</i> .....	54

Gambar IV. 15 <i>Activity diagram Add Admin</i> .....	55
Gambar IV. 16 <i>Activity diagram Manage Consultation</i> .....	56
Gambar IV. 17 <i>Activity diagram Log Out Admin</i> .....	57
Gambar IV. 18 <i>Class Diagram</i> .....	58
Gambar IV. 19 Tabel Relasi.....	62
Gambar IV. 20 <i>Wireframe Login</i> .....	63
Gambar IV. 21 <i>Wireframe Buat Akun</i> .....	63
Gambar IV. 22 <i>Wireframe Home</i> .....	64
Gambar IV. 23 <i>Wireframe Pendaftaran</i> .....	64
Gambar IV. 24 <i>Wireframe Profile</i> .....	65
Gambar IV. 25 <i>Wireframe Riwayat</i> .....	66
Gambar IV. 26 <i>Wireframe Login Admin</i> .....	66
Gambar IV. 27 <i>Wireframe Home Admin</i> .....	67
Gambar IV. 28 <i>Wireframe Detail Konsultasi</i> .....	67
Gambar IV. 29 <i>Wireframe Manage Admin</i> .....	68
Gambar IV. 30 <i>Wireframe Add Admin</i> .....	69
Gambar IV. 31 <i>Wireframe Manage Consultation</i> .....	69
Gambar IV. 32 Tampilan <i>Login User</i> .....	70
Gambar IV. 33 Tampilan <i>Create an Account</i> .....	71
Gambar IV. 34 Tampilan <i>Home</i> .....	72
Gambar IV. 35 Tampilan <i>Pendaftaran</i> .....	72
Gambar IV. 36 Tampilan <i>Profile</i> .....	73
Gambar IV. 37 Tampilan <i>Riwayat</i> .....	74
Gambar IV. 38 Tampilan <i>Login Admin</i> .....	74
Gambar IV. 39 Tampilan <i>Home Admin</i> .....	75
Gambar IV. 40 Tampilan <i>Detail Konsultasi</i> .....	75
Gambar IV. 41 Tampilan <i>Manage Admin</i> .....	76
Gambar IV. 42 Tampilan <i>Add Admin</i> .....	76
Gambar IV. 43 Tampilan <i>Manage Consultation</i> .....	77
Gambar IV. 44 Grafik Uji Fungsionalitas 1 .....	81
Gambar IV. 45 Grafik Uji Fungsionalitas 2 .....	81
Gambar IV. 46 Grafik Uji Fungsionalitas 3 .....	82

Gambar IV. 47 Grafik Uji Fungsionalitas 4 .....	82
Gambar IV. 48 Grafik Uji Fungsionalitas 5 .....	82
Gambar IV. 49 Grafik Ui Responsif 1 .....	83
Gambar IV. 50 Grafik Uji Responsif 2 .....	83

## DAFTAR LAMPIRAN

LAMPIRAN A. TOR .....	LAMPIRAN A - 1
LAMPIRAN B. LEMBAR WAWANCARA.....	LAMPIRAN B - 1
LAMPIRAN C. <i>LOG ACTIVITY</i> .....	LAMPIRAN C - 1
LAMPIRAN D. DOKUMENTASI FOTO .....	LAMPIRAN D - 1

# **BAB I**

## **PENDAHULUAN**

### **I.1 Latar Belakang**

Sistem informasi pendaftaran Puskesmas adalah layanan berbasis teknologi yang dirancang untuk mempermudah masyarakat dalam mendaftar layanan kesehatan hewan. Sistem ini meliputi pengumpulan data pemilik dan hewan, penjadwalan layanan, serta pencatatan riwayat medis hewan secara digital, menggantikan metode manual yang lebih lambat dan rentan kesalahan. Pendaftaran yang efisien sangat penting untuk mendukung operasional Puskesmas, khususnya bagi lembaga yang ingin memberikan pelayanan cepat, akurat, dan ramah pengguna. Dengan teknologi berbasis web, sistem ini memungkinkan pengelolaan data yang lebih terorganisir dan efisien, mempercepat pendaftaran, serta mengurangi potensi kesalahan. Akses informasi yang cepat dan akurat membantu petugas Puskesmas memberikan layanan kesehatan hewan yang lebih baik dan terstruktur.

Puskesmas Arjasari merupakan Pusat Kesehatan Hewan Pembantu yang berada di Arjasari, Kabupaten Bandung, yang menjadi salah satu fasilitas pelayanan kesehatan hewan yang disediakan oleh pemerintah, puskesmas ini memiliki peran penting dalam mendukung kesehatan hewan peliharaan dan ternak di daerah sekitarnya. Fasilitas ini bertujuan untuk memberikan layanan kesehatan yang terjangkau dan berkualitas, sehingga dapat membantu masyarakat dalam menjaga kesehatan hewan. Puskesmas ini juga menjadi tempat edukasi bagi masyarakat tentang pentingnya menjaga kesehatan hewan sebagai bagian dari upaya meningkatkan kesejahteraan hewan dan produktivitas peternakan. Sebagai unit pelayanan yang berada di bawah pengelolaan pemerintah, menjadikan meningkatnya jumlah masyarakat yang memanfaatkan layanan puskesmas dari waktu ke waktu.

Pelayanan kesehatan hewan di Puskesmas Pembantu Arjasari menghadapi tantangan karena proses pendaftaran yang masih manual, memakan waktu, rentan kesalahan, dan sulit dalam pengelolaan data pasien. Hal ini menghambat peningkatan kualitas pelayanan. Pencatatan data yang tidak terstruktur menyulitkan petugas dalam mengakses informasi riwayat medis, sementara kesalahan dalam pencatatan dan penjadwalan layanan menurunkan kepuasan masyarakat. Waktu pendaftaran yang lama dan tidak adanya sistem terpusat juga menyebabkan akses data menjadi lambat dan tidak efisien.

Perancangan sistem informasi pendaftaran berbasis web untuk Puskesmas Pembantu Arjasari menjadi solusi efektif untuk mengatasi masalah dalam proses pendaftaran manual, seperti tidak efisien, kesalahan pencatatan, dan sulitnya pengelolaan data. Sistem digital berbasis web ini dirancang untuk menggantikan metode manual yang selama ini digunakan, sehingga mampu meningkatkan akurasi pencatatan data dan mempermudah akses informasi dalam pengimplementasian sistem pendaftaran yang modern yang kemudian dapat membantu petugas dalam mengelola data pasien hewan. Dalam pengembangannya, metode *waterfall* digunakan sebagai pendekatan sistematis yang melibatkan tahapan analisis, desain, implementasi, pengujian, dan pemeliharaan. Hasil dari proyek ini adalah sebuah aplikasi pendaftaran pelayanan puskesmas pembantu Arjasari berbasis web yang mudah diakses bagi petugas dan masyarakat. Aplikasi ini dilengkapi dengan fitur-fitur seperti pendaftaran online, dan pengelolaan data pasien. Sistem ini diharapkan mampu meningkatkan kualitas pelayanan, pengelolaan data yang lebih efisien, serta kemudahan akses informasi yang mendukung operasional Puskesmas secara keseluruhan. Dengan adanya sistem ini, diharapkan agar meningkatnya kualitas pelayanan kesehatan hewan dapat berjalan dengan lebih optimal dan memberikan manfaat signifikan bagi masyarakat, serta efisiensi pengelolaan untuk bagi staf administrasi.

## I.2 Lingkup

Lingkup perancangan sistem informasi pendaftaran berbasis web untuk Puskesmas Pembantu Arjasari mencakup pembuatan aplikasi digital yang mempermudah pendaftaran dan pengelolaan data pasien hewan. Sistem ini dirancang untuk dua jenis pengguna, yaitu *user* dan *admin*, dengan fitur utama seperti menu *login*, pendaftaran pasien baru, dan halaman *home* yang berisi informasi antrian. Pengembangan sistem menggunakan UML, termasuk *use case diagram*, *activity diagram*, *class diagram*, struktur tabel, dan tabel relasi untuk memastikan desain yang terstruktur. Metode *waterfall* diterapkan melalui tahapan analisis, desain, implementasi, pengujian, dan pemeliharaan, dengan *Express.js* digunakan sebagai framework *backend* utama untuk mendukung aplikasi berbasis web yang cepat, aman dan efisien.

Dalam pelaksanaan kerja praktik ini, lingkup pekerjaan dibatasi untuk memastikan fokus dan keberhasilan pencapaian tujuan yang mana diantara lingkungannya meliputi:

1. Lokasi : Sistem ini dirancang untuk digunakan di Puskesmas Pembantu Pustu Arjasari, sebuah fasilitas kesehatan hewan yang melayani masyarakat yang berlokasi di Kp. Cihonje Rt.03/Rw.01 Ds. Wargaluyu, Kec. Arjasari, Kabupaten Bandung.
2. Fokus Utama: Permasalahan di puskesmas yaitu penyebaran informasi dan pendaftaran pelayanan yang masih di lakukan secara manual, kerja praktik ini di fokuskan pada rancang bangun aplikasi pendaftaran pelayanan puskesmas berbasis web untuk mempermudah dan mempercepat proses pendaftaran serta pengelolaan data pasien hewan secara digital, menggantikan metode manual yang selama ini digunakan.
3. Perancangan : Perancangan sistem melibatkan pembuatan antarmuka pengguna yang sederhana dan mudah digunakan oleh dua jenis pengguna, yaitu *user* (masyarakat) dan *admin* (petugas Puskesmas), dengan memperhatikan navigasi dan efisiensi proses. Perancangan ini dilakukan dengan menggunakan UML yang terdiri dari *use case*



*diagram, activity diagram, class diagram*, struktur tabel, dan tabel relasi untuk mendeskripsikan struktur dan alur sistem, serta perancangan *UI/UX* yang mencakup pembuatan *wireframe* untuk memastikan tampilan yang intuitif dan *user-friendly*, sehingga memudahkan pengguna dalam berinteraksi dengan sistem.

4. Pengembangan : Sistem ini dikembangkan dengan metode *waterfall*, yang meliputi tahapan analisis, desain, implementasi, pengujian, dan pemeliharaan. Teknologi yang digunakan mencakup *Express.js* untuk backend berbasis web, serta *HTML*, *CSS* dengan *Bootstrap*, dan *JavaScript (Node.js)*.
5. Fungsionalitas dan Fitur : Sistem ini memiliki berbagai fitur, di antaranya untuk *user*: menu *login* untuk autentikasi, pendaftaran online pasien hewan, *home* untuk informasi antrian, riwayat untuk pengelolaan data konsultasi, dan *Profile* untuk data pengguna. Untuk admin: menu *login* admin, *home* untuk mengelola data pasien, *Manage Admin* untuk mengelola daftar admin, dan *Manage Consultation* untuk melihat riwayat pasien. Fitur-fitur ini dirancang untuk meningkatkan efisiensi dan akurasi data.
6. Pelaporan : Pelaporan hasil kerja praktik dilakukan pada tahap akhir kerja praktik di puskesmas Arjasari, yang mencakup perancangan dan pembangunan sistem informasi pendaftaran pelayanan puskesmas.

### **I.3 Tujuan**

Tujuan dilaksanakannya Kerja Praktik di Puskesmas Pembantu Arjasari adalah sebagai berikut :

1. Membangun sistem aplikasi Pendaftaran Pelayanan bagi instansi Pusat Kesehatan Hewan Pembantu Arjasari berbasis web.
2. Meningkatkan efektifitas dan efisiensi dalam kemudahan melakukan proses pendaftaran.
3. Mengelola data pasien secara lebih efisien dan meningkatkan kualitas pelayanan kepada masyarakat.

## **BAB II**

### **LINGKUNGAN KERJA PRAKTIK**

#### **II.1 Struktur Organisasi**

Struktur organisasi Puskesmas Pembantu Pustu Arjasari terdiri dari beberapa tingkat jabatan yang memiliki peran dan tanggung jawab masing-masing dalam menjalankan operasional pelayanan kesehatan hewan.



Gambar II. 1 Struktur Organisasi Puskesmas Pembantu (Pustu) Arjasari

*Sumber : Pustu Arjasari*

Struktur organisasi Puskesmas Pembantu (Pustu) Arjasari dirancang untuk memastikan kelancaran operasional dan pelayanan kesehatan hewan yang efektif. Setiap anggota dalam organisasi ini memiliki peran dan tanggung jawab yang jelas, mulai dari pengelolaan administrasi hingga pelaksanaan layanan medis, yang semuanya berkontribusi dalam mendukung tujuan

Puskesmas untuk memberikan pelayanan terbaik kepada masyarakat dan hewan yang membutuhkan.

1. UPTD Puskesmas Arjasari : sebagai lembaga utama yang menjadi pusat koordinasi Puskesmas Pembantu (Pustu) Arjasari dalam pelaksanaan berbagai kegiatan terkait pelayanan kesehatan hewan dalam organisasi.
2. Koordinator di Puskesmas Pembantu (Pustu) Arjasari, yang dijabat oleh Drh. Azizah Rahman, bertugas untuk mengawasi pelayanan kesehatan hewan yang dilakukan oleh tim, termasuk paramedik veteriner, serta memastikan semua kegiatan berjalan sesuai dengan prosedur yang telah ditetapkan. Koordinator juga mengoordinasikan tim untuk membagi tugas kepada staf, seperti paramedik veteriner, dan menyusun rencana kegiatan atau program kerja terkait pelayanan kesehatan hewan. Selain itu, koordinator menentukan prioritas kegiatan untuk memastikan efektivitas dan efisiensi kerja, melakukan evaluasi hasil kerja tim, serta memastikan target yang ditetapkan tercapai. Koordinator juga bertanggung jawab untuk menyusun laporan kegiatan, termasuk data kesehatan hewan dan pelaksanaan program, serta melaporkan hasil kerja kepada pimpinan atau atasan.
3. Staf Teknis (Paramedik Veteriner) berperan dalam mendukung kegiatan teknis yang berkaitan dengan pelayanan kesehatan hewan, baik dalam pencegahan, perawatan, maupun pengobatan penyakit. Paramedik veteriner ini bertindak sebagai perpanjangan tangan dokter hewan untuk memastikan hewan tetap sehat dan produktif. Dan Paramedik Veteriner di Puskesmas Pembantu (Pustu) Arjasari dijalankan oleh Bapak Aep Tatang Rohmana dan Bapak Fiter Faisal.

## **II.2 Lingkup Pekerjaan**

Lingkup pekerjaan divisi di Puskesmas Pembantu (Pustu) Arjasari mencakup berbagai tugas yang berfokus pada pelayanan kesehatan hewan dan administrasi terkait. Divisi ini bertanggung jawab dalam memberikan perawatan medis kepada hewan yang membutuhkan, termasuk

pemeriksaan kesehatan, pengobatan, vaksinasi, dan tindakan pencegahan penyakit. Selain itu, divisi ini juga mengelola data pasien hewan, melakukan pencatatan medis, serta memastikan kelancaran administrasi pendaftaran. Divisi ini bekerja sama dengan seluruh tim, termasuk paramedik veteriner, untuk memastikan pelayanan yang efektif, efisien, dan sesuai dengan prosedur yang telah ditetapkan.

Selama proses kerja praktik, Puskesmas Pembantu Pustu Arjasari berencana untuk mengembangkan sistem informasi pendaftaran berbasis web untuk meningkatkan efisiensi dan kualitas pelayanan kesehatan hewan. Fokus utama dari kerja praktik ini adalah merancang dan membangun sistem pendaftaran online yang dapat mempermudah proses pendaftaran pasien hewan serta pengelolaan data kesehatan hewan. Sistem ini dirancang untuk memenuhi kebutuhan Puskesmas dalam mempercepat administrasi pendaftaran, meningkatkan akurasi data, dan memberikan kemudahan akses bagi masyarakat. Lingkup pekerjaan meliputi analisis kebutuhan, desain antarmuka, pengembangan web menggunakan framework Express.js, hingga implementasi awal untuk memastikan sistem dapat digunakan secara efektif oleh petugas Puskesmas dan masyarakat.

### **II.3 Deskripsi Pekerjaan**

Pembuatan aplikasi pendaftaran pelayanan Puskesmas Pembantu (Pustu) Arjasari berbasis web melibatkan beberapa tahapan terstruktur untuk menghasilkan sistem yang sesuai dengan kebutuhan operasional. Dimulai dengan analisis kebutuhan untuk mengidentifikasi fitur yang diperlukan, kemudian dilanjutkan dengan perancangan sistem menggunakan *Unified Modeling Language*, termasuk *use case diagram*, *activity diagram*, dan *class diagram* untuk menggambarkan alur dan struktur sistem. Setelah perancangan, tahap pengembangan dilakukan menggunakan *framework Express.js*, diikuti dengan implementasi dan pengujian untuk memastikan

sistem berfungsi dengan baik dan efektif digunakan oleh petugas dan masyarakat.

1. Analisis Kebutuhan :

Melakukan Pengumpulan Data dengan melakukan wawancara terhadap petugas puskesmas untuk memahami alur kerja pendaftaran yang berjalan saat ini, serta mengidentifikasi permasalahan utama. Melakukan identifikasi kebutuhan fungsional dan non-fungsional terhadap fitur utama utama yang harus dimiliki aplikasi, seperti pendaftaran pasien, nomor antrian, dan pengelolaan data pasien, dan persyaratan teknis seperti keamanan data kecepatan akses, dan kompatibilitas perangkat. Melakukan identifikasi analisis permasalahan untuk kendala sistem manual, seperti waktu pendaftaran yang lama, kesalahan pencatatan, dan kesulitan pencarian data.

2. Perancangan :

Pada tahap ini akan dilakukan perancangan dengan menggunakan UML (*Unified Modelling Language*)

a. *Use case*

Pada *use case diagram*, akan dibuat gambaran interaksi antara pengguna (*user* dan *admin*) dengan sistem, termasuk fitur-fitur utama seperti *login*, pendaftaran, *home*, *Profile*, riwayat, *Manage Admin*, dan manage konsultasi.

b. *Activity diagram*

*Activity diagram* akan menggambarkan alur aktivitas untuk *user* dan *admin*. Aktivitas untuk *user* mencakup pembuatan halaman *login*, halaman pendaftaran, halaman *home*, halaman *profile*, dan halaman riwayat. Sedangkan untuk *admin*, aktivitas mencakup pembuatan halaman *login admin*, halaman *home*, halaman *Manage Admin*, dan halaman *Manage Consultation*.

c. *Class Diagram*

Pada perancangan *class diagram* dirancang dengan tiga class utama untuk mendukung fungsionalitas sistem. *Class Admin* mencakup atribut seperti *id*, *email*, *password*, dan *role* untuk

mengelola data administrator. *Class User* dirancang dengan atribut *email*, *password*, nama, NIK, alamat, kelurahan, kecamatan, dan nomor WA untuk menyimpan informasi pengguna atau masyarakat yang mendaftar layanan. *Class Consultation* mencakup atribut *id*, *namaHewan*, *umurHewan*, *jenisKelamin*, *jenisHewan*, *keluhan*, *status*, *userId*, dan *queueNumber* untuk mencatat data konsultasi hewan, termasuk identitas hewan dan keluhan yang disampaikan.

d. Tabel relasi

Perancangan tabel relasi dilakukan untuk menghubungkan data antar *class* dalam sistem, termasuk relasi antara *user* dan *consultation*, serta admin dan data yang dikelola.

e. *Wireframe*

*Wireframe* dibuat untuk mendesain tata letak dan struktur antarmuka aplikasi, memastikan setiap halaman mudah digunakan dan intuitif.

f. Tampilan Antarmuka

Tampilan antarmuka dirancang dengan fokus pada kesederhanaan dan kemudahan navigasi, mencakup halaman *login*, pendaftaran, *home*, *profile*, riwayat, dan halaman pengelolaan admin dengan navigasi halaman *login admin*, *home*, *manage admin* dan *manage consultation*.

3. Implementasi

Implementasi sistem dilakukan menggunakan teknologi yang dirancang untuk memastikan kinerja yang optimal dan efisien. *HTML*, *CSS*, dan *JavaScript* digunakan untuk merancang tampilan antarmuka yang interaktif dan responsif, sehingga memudahkan pengguna dalam mengakses fitur-fitur aplikasi. Untuk pengembangan logika aplikasi dan backend, digunakan *framework Express.js* atau *Node.js*, yang memungkinkan pengelolaan data dan proses aplikasi berjalan dengan cepat dan terstruktur. Selain itu, *postgreSQL* digunakan sebagai *database* untuk mengelola dan menyimpan data pasien secara aman.

#### 4. Pengujian

Pengujian sistem dilakukan dengan menggunakan metode *black box testing* untuk memastikan bahwa semua fitur berfungsi sesuai dengan tujuannya, tanpa memeriksa kode program secara langsung. Selain itu, dilakukan juga pengujian reliabilitas menggunakan *Google Form* yang berisi pertanyaan-pertanyaan terkait fungsionalitas *website* untuk mengukur sejauh mana sistem dapat diandalkan dan memenuhi harapan pengguna.

#### 5. Pelaporan

Setelah sistem selesai diuji, laporan akan disusun untuk mendokumentasikan seluruh proses pengembangan, mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian, sebagai bagian dari hasil kerja praktik.

### II.4 Jadwal Kerja

Adapun jadwal dalam pelaksanaan kerja praktik adalah sebagai berikut:

Tabel II. 1 Jadwal Kerja Praktik

No	Kegiatan	Minggu ke-							
		I	II	III	IV	V	VI	VII	VIII
1.	Observasi dan pengenalan tempat kerja praktik								
2.	Pengumpulan Data dan Analisis kebutuhan								
3.	Perancangan aplikasi								
4.	Implementasi sistem								
5.	Pengujian								
6.	Penyusunan laporan kerja praktik								

Kerja praktik dilaksanakan mulai tanggal 25 November 2024 hingga Desember 2024, dengan durasi pelaksanaan selama 8 minggu. Setelah selesai menjalani periode kerja praktik, proses dilanjutkan dengan tahap penyusunan laporan guna mendokumentasikan hasil dan pengalaman yang diperoleh selama menjalani kerja praktik tersebut. Adapun kegiatan yang dilakukan selama kerja praktik adalah sebagai berikut :

1. Minggu Pertama :

- a. Melakukan survei dan observasi pengenalan tempat kerja praktik serta melakukan diskusi dengan pihak administrasi terkait perizinan yang diperlukan untuk pelaksanaan kegiatan kerja praktik.
- b. Melakukan diskusi dengan pihak administrasi terkait perizinan yang diperlukan untuk pelaksanaan kegiatan kerja praktik.

2. Minggu Kedua :

- a. Observasi mengenai kegiatan sehari-hari yang dilakukan di UPT Puskesmas Pembantu (Pustu) Arjasari. Serta melakukan wawancara terhadap dokter yang bertanggung jawab dalam menangani pasien.
- b. Melakukan analisis sistem serta menyusun kesepakatan terkait perancangan aplikasi yang disesuaikan dengan kebutuhan di lokasi kerja praktik.
- c. Mengumpulkan data yang diperlukan untuk merancang aplikasi yang akan dikembangkan.

3. Minggu Ketiga :

- a. Menyusun alur kerja (*workflow*) aplikasi berdasarkan hasil analisis kebutuhan.
- b. Membuat rancangan awal antarmuka dan memastikan rancangan sesuai dengan kebutuhan pengguna dan mudah digunakan.

4. Minggu Keempat :

- a. Membuat rancangan awal antarmuka dan memastikan rancangan sesuai dengan kebutuhan pengguna dan mudah digunakan.



5. Minggu Kelima :

- a. Menguji kelayakan rancangan aplikasi, dan melakukan evaluasi untuk memastikan rancangan aplikasi memenuhi standar kelayakan.
- b. Menyepakati rancangan sistem dengan pembimbing lapangan kerja praktik atau pihak terkait

6. Minggu Keenam :

- a. Menguji kelayakan rancangan aplikasi, dan melakukan evaluasi untuk memastikan rancangan aplikasi memenuhi standar kelayakan.

7. Minggu Ketujuh :

- a. Menguji akhir kelayakan rancangan aplikasi.

8. Minggu Kedelapan :

- a. Melakukan konsultasi dengan pembimbing kerja praktik serta mencatat hasil diskusi untuk disampaikan dalam laporan kepada pembimbing.
- b. Menyusun laporan yang merangkum hasil kegiatan selama kerja praktik.

## **BAB III**

### **TEORI PENUNJANG KERJA PRAKTIK**

#### **III.1 Teori Penunjang**

Selama menjalani kerja praktik di Puskesmas Pembantu Pustu Arjasari, saya memanfaatkan pengetahuan yang diperoleh selama masa perkuliahan sebagai dasar teori dalam merancang Sistem Informasi Pendaftaran Berbasis *Website* di Puskesmas tersebut. Adapun pengetahuan dan teori yang diterapkan meliputi:

##### **1. Teori Algoritma**

Algoritma merupakan salah satu cabang dari ilmu komputer dan matematika yang berfokus pada pembelajaran konsep, karakteristik, serta penerapan algoritma. Algoritma sendiri didefinisikan sebagai serangkaian langkah logis yang dirancang untuk menyelesaikan masalah dengan cara efektif dan efisien. Teori ini menjadi landasan utama dalam pengembangan perangkat lunak, analisis data, serta berbagai aplikasi teknologi lainnya. Dalam konteks pengembangan aplikasi pendaftaran untuk Puskesmas Pembantu Arjasari, teori algoritma berperan dalam memastikan aplikasi tersebut dapat beroperasi secara efisien, akurat, dan memenuhi kebutuhan pengguna. Dalam mata kuliah TIF301 Algoritma dan Pemrograman.

##### **2. Teori Teknologi Informasi**

Teknologi Informasi (TI) merupakan bidang ilmu yang berfokus pada proses pengumpulan, pengolahan, penyimpanan, dan distribusi informasi dengan memanfaatkan teknologi berbasis komputer. Dalam proyek pengembangan aplikasi pendaftaran untuk Puskesmas Pembantu (Pustu) Arjasari, teori TI menyediakan landasan untuk mengintegrasikan teknologi guna mendukung pengelolaan data, meningkatkan efisiensi operasional, serta memberikan layanan yang lebih baik kepada masyarakat. Materi ini dipelajari dalam mata kuliah UBB105 Pengantar Teknologi Informasi.

### 3. Teori Pemrograman

Teori ini membahas dasar-dasar pemrograman, meliputi sintaksis, struktur program, dan logika dalam pengembangan aplikasi pendaftaran Puskesmas Pembantu Arjasari. Praktikum ini mendukung pembuatan kode dasar, termasuk desain antarmuka pengguna menggunakan teknologi seperti *HTML*, *CSS*, dan *JavaScript* untuk validasi input dan navigasi pada aplikasi web. Fokusnya adalah pada desain *front-end* yang menarik dan fungsional untuk memastikan aplikasi berjalan lancar, memudahkan pendaftaran, dan menjaga integritas data. Pengetahuan ini diperoleh dari mata kuliah FTI205 Praktikum Pengantar Pemrograman.

### 4. Teori Struktur Data

Teori struktur data membantu pengorganisasian dan penyimpanan data secara efisien, seperti penggunaan *hash tables* untuk pencarian cepat dan *queue* untuk mengelola antrean layanan. Pemilihan struktur data yang tepat meningkatkan efisiensi pencarian dan pengambilan data, misalnya saat menampilkan daftar konsultasi hewan berdasarkan nama. Pengetahuan ini diperoleh dari mata kuliah TIF304 Struktur Data.

### 5. Teori Basis Data

Sistem basis data adalah tempat penyimpanan data yang terorganisir dan dapat diakses dengan efisien oleh aplikasi. Dalam proyek aplikasi pendaftaran Puskesmas Pembantu Arjasari, *database* digunakan untuk menyimpan data pendaftar, layanan kesehatan hewan, dan informasi terkait lainnya. Teori normalisasi, *query*, dan manajemen basis data diterapkan untuk merancang struktur *database* memastikan data tersimpan dengan baik dan mudah diakses. *Database* relasional seperti *postgreSQL* digunakan untuk memastikan keamanan dan integritas data. Diperoleh dari mata kuliah TIF311 Sistem Basis Data.

### 6. Metode Penelitian

Metode penelitian adalah proses untuk mengumpulkan dan menganalisis data dalam pengembangan proyek. Pada proyek

pembuatan aplikasi pendaftaran Puskesmas Pembantu Arjasari, metode penelitian digunakan untuk mengidentifikasi masalah yang ada, memahami kebutuhan sistem, dan menganalisis sistem administrasi pendaftaran yang sudah berjalan. Teknik pengumpulan data seperti wawancara, survei, dan observasi digunakan untuk memperoleh informasi dari petugas Puskesmas dan calon pengguna aplikasi. Data yang terkumpul kemudian dianalisis untuk menentukan fitur-fitur penting yang harus ada dalam sistem yang akan dibangun. Pengetahuan ini diperoleh melalui Mata Kuliah FTI208 Metode Penelitian

#### 7. Teori mengenai Sistem Informasi Manajemen

Teori sistem informasi manajemen memberikan kerangka kerja untuk mendukung pengambilan keputusan dan efisiensi operasional. Aplikasi dirancang dengan fitur seperti *dashboard* untuk memantau antrian dan jadwal layanan secara *real-time*, serta laporan data pendaftar untuk mendukung manajemen Puskesmas. Dengan penerapan teori-teori ini, aplikasi dapat dirancang untuk memenuhi kebutuhan pengguna, berjalan efisien, dan berkelanjutan dalam jangka panjang. Pengetahuan ini diperoleh dari mata kuliah TIF312 Sistem Informasi Manajemen yang memberikan wawasan dalam pemanfaatan teknologi meningkatkan kualitas layanan.

### III.2 Peralatan Pembangunan

Pada tahap rancang bangun aplikasi pendaftaran Puskesmas Pembantu Arjasari, berbagai peralatan dan teknologi digunakan untuk memastikan keberhasilan dan kelancaran proses pembangunan. Peralatan ini mencakup perangkat keras, perangkat lunak, serta alat bantu lainnya yang mendukung setiap aspek pengembangan sistem, mulai dari perancangan antarmuka hingga pengelolaan basis data. Pemilihan peralatan yang tepat sangat penting untuk meningkatkan efisiensi, kualitas, dan keberlanjutan aplikasi yang dibangun. Berikut adalah beberapa peralatan yang digunakan dalam pembangunan aplikasi ini:

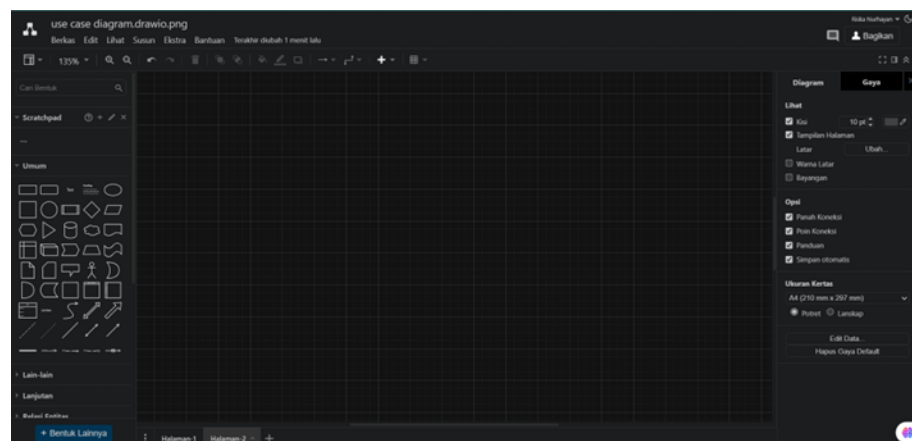
## 1. Software

*Software* adalah program yang berisi kumpulan intruksi untuk melakukan proses pengolahan data. *Software* sebagai penghubung antara manusia sebagai pengguna dengan perangkat keras komputer, berfungsi menerjemahkan bahasa manusia ke dalam bahasa mesin untuk menjalankan instruksi yang diberikan hingga memberikan hasil yang diinginkan oleh manusia tersebut. (Jan et al., 2022)

*Software* yang digunakan dalam rancang bangun aplikasi pendaftaran pelayanan puskesmas pembantu (pustu) Arjasari :

### a. *Draw.io*

*Draw.io*, dikenal sebagai *diagrams.net*, adalah alat *diagramming* berbasis web yang digunakan membuat berbagai diagram seperti *flowchart*, diagram jaringan, dan diagram UML. Alat ini menawarkan berbagai bentuk, *template*, dan fitur untuk pembuatan diagram profesional. *Draw.io* dapat digunakan secara *online* maupun *offline* dan terintegrasi dengan layanan penyimpanan *cloud* populer seperti *Google Drive* dan *OneDrive*. (Hendrawan1 et al., 2022)



Gambar III. 1 Tampilan *draw.io*

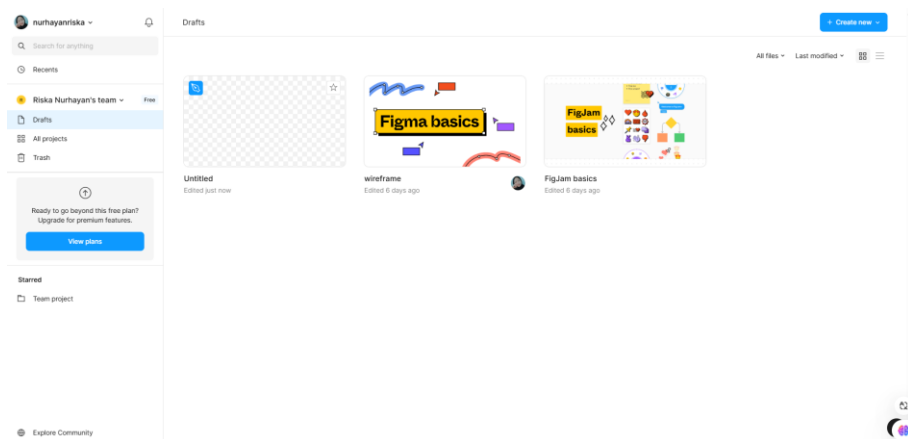
Sumber : Screenshot tampilan *darw.io*

Dalam proyek Puskesmas, *Draw.io* digunakan untuk membuat diagram alur proses aplikasi, seperti alur pendaftaran pengguna, jadwal konsultasi, dan *autentikasi admin*. Diagram ini membantu tim

pengembang memahami alur kerja aplikasi secara visual, mempermudah perencanaan dan kolaborasi dalam pengembangan fitur-fitur utama aplikasi.

b. *Figma*

*Figma* adalah sebuah alat desain yang sering dimanfaatkan untuk merancang antarmuka aplikasi *mobile*, *desktop*, *website*, dan lainnya. Alat ini dapat diakses pada berbagai sistem operasi seperti *Windows*, *Linux*, dan *macOS*, selama terhubung dengan internet. Salah satu keunggulan utama *Figma* adalah kemampuannya mendukung kolaborasi, memungkinkan beberapa orang untuk bekerja pada proyek yang sama secara bersamaan, meskipun berada di lokasi yang berbeda. (Al-Faruq et al., 2022)



Gambar III. 2 Tampilan *Figma*

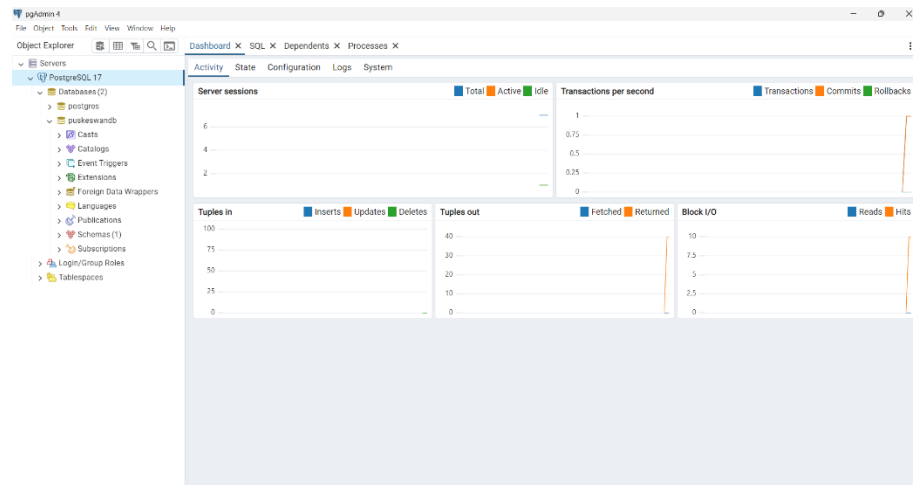
Sumber : Screenshot Tampilan *Figma*

*Figma* digunakan untuk membuat *wireframe* sebagai langkah awal dalam perancangan antarmuka aplikasi. *Wireframe* ini menggambarkan struktur dasar halaman seperti formulir pendaftaran, nomor antrian, *Profile* pengguna, riwayat konsultasi, dan *dashboard admin*.

c. *PgAdmin*

*PgAdmin* adalah alat basis data *postgreSQL* dengan kaya fitur. Sebagai antarmuka grafis, *pgAdmin* mempermudah pengguna

dalam mengelola *postgreSQL*, seperti membuat tabel, menulis *query* SQL, dan memantau kinerja basis data. Alat ini kompatibel dengan berbagai platform dan dirancang untuk memenuhi kebutuhan pengembang serta administrator dalam mengelola *postgreSQL* secara efisien (Ing. Michal Valenta, 2008).



Gambar III. 3 Tampilan *PgAdmin*

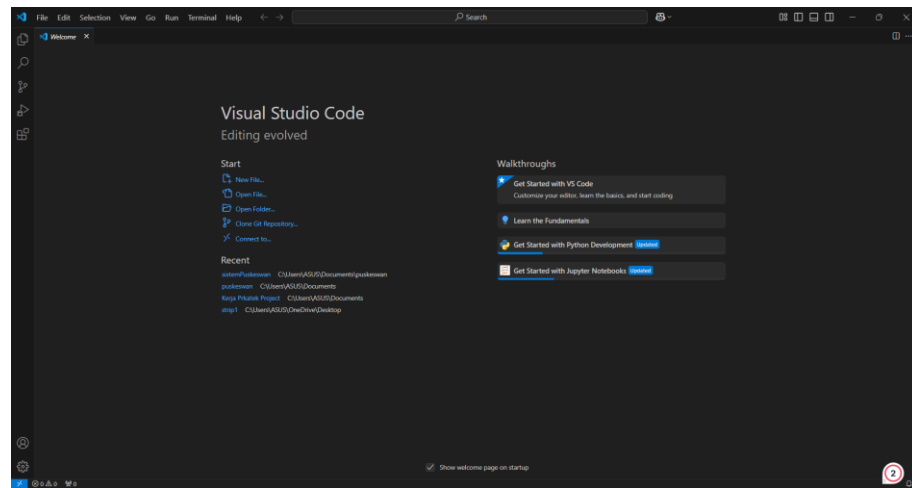
Sumber : Screenshot Tampilan *PgAdmin*

Dalam proyek Puskesmas, *pgAdmin* digunakan sebagai alat untuk mengelola dan memantau *database* yang menyimpan berbagai informasi terkait pendaftaran dan konsultasi. Alat ini memungkinkan admin untuk melakukan berbagai tugas administratif dengan lebih efisien, seperti memeriksa dan memverifikasi data pendaftar yang masuk, serta melakukan perubahan atau pembaruan jika ada kesalahan dalam data. Selain itu, jika diperlukan, admin dapat menggunakan *pgAdmin* untuk memperbaiki atau menyesuaikan struktur tabel dalam *database* agar lebih optimal dan sesuai dengan kebutuhan aplikasi.

#### d. *Visual Studio Code*

*Visual Studio Code* (VS Code) adalah editor kode sumber gratis yang ringan, cepat, dan kaya fitur, dikembangkan oleh *Microsoft*. Mendukung berbagai bahasa pemrograman, VS Code dilengkapi

dengan fitur seperti *debugging*, kontrol versi Git, integrasi terminal, serta ekstensi yang dapat disesuaikan untuk meningkatkan produktivitas. Dengan antarmuka yang intuitif dan performa tinggi, VS Code menjadi alat populer bagi pengembang di berbagai platform seperti *Windows*, *macOS*, dan *Linux*.



Gambar III. 4 Tampilan *Visual Studio Code*

Sumber : Screenshot Tampilan *Visual Studio Code*

Untuk pembuatan kode-kode program dibutuhkan sebuah aplikasi yang mumpuni. Dalam hal ini dapat menggunakan *Visual studio code*. *Visual Studio Code* adalah *Software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan *built-in* dukungan untuk *JavaScript*, naskah dan *Node.js* dan memiliki *array* beragam ekstensi yang tersedia untuk bahasa lain, termasuk *C ++*, *C #*, *Python*, dan *PHP*. (Hartati, 2020)

#### e. Koneksi Internet

Koneksi internet adalah jaringan global yang memungkinkan perangkat saling terhubung dan bertukar data melalui protokol standar seperti *TCP/IP* (Aulia et al., 2023). Berbagai media digunakan untuk koneksi internet, seperti kabel fiber-optik, jaringan seluler, dan koneksi nirkabel seperti *Wi-Fi*. Internet telah menjadi infrastruktur penting untuk mendukung berbagai aktivitas, termasuk



komunikasi, pendidikan, bisnis, dan hiburan. Koneksi *Wi-Fi*, salah satu jenis koneksi nirkabel yang populer, menggunakan gelombang radio untuk menyediakan akses internet dengan kecepatan yang tinggi (Ode Bakrim & Bina Bangsa Kendari, 2019).

Dalam proyek aplikasi pendaftaran Puskesmas, koneksi *Wi-Fi* berperan penting untuk memastikan aksesibilitas aplikasi oleh pengguna dan pengelola. *Wi-Fi* mendukung komunikasi antara server berbasis Node.js, *database postgresQL*, dan antarmuka pengguna berbasis web. Dengan koneksi internet yang stabil melalui *Wi-Fi*, pengguna dapat mendaftarkan hewan, mengakses jadwal konsultasi, dan menerima pembaruan aplikasi secara *real-time* tanpa gangguan.

## 2. **Hardware**

*Hardware* atau biasa dikenal sebagai perangkat keras adalah merupakan peralatan sistem komputer yang secara fisik dapat dilihat dan dijamah. Peralatan ini meliputi beberapa komponen yaitu komponen masukan (*input device*), komponen pemroses (*processing device*), komponen keluaran (*output*) dan komponen simpanan luar (*storage*). (Jan et al., 2022) Dalam pengembangan sistem informasi, perangkat keras (*hardware*) memegang peranan penting sebagai fondasi infrastruktur yang mendukung operasional sistem. Pemilihan perangkat keras yang tepat sangat memengaruhi kinerja, keandalan, dan efisiensi sistem yang dikembangkan. *Hardware* menjadi elemen esensial dalam memastikan bahwa aplikasi pendaftaran Puskesmas Pembantu Arjasari dapat berjalan dengan optimal, mendukung kebutuhan pengolahan data secara efektif dan efisien.

Pada bagian ini, akan dijelaskan komponen-komponen perangkat keras yang digunakan dalam proyek ini, termasuk spesifikasi teknis dan perannya masing-masing dalam mendukung keberlangsungan sistem. Dengan pemahaman yang mendalam terhadap kebutuhan hardware,

sistem dapat dirancang agar lebih efisien dan dapat diandalkan dalam jangka panjang.

a. *Komputer Client*

Komputer klien adalah perangkat yang digunakan untuk mengakses layanan, aplikasi, atau sumber daya yang disediakan oleh server dalam sebuah jaringan. (Andrew S. Tanenbaum & David J. Wetherall, 2011) Klien berfungsi sebagai antarmuka pengguna, memungkinkan pengguna untuk mengirim permintaan dan menerima data dari server. Komputer klien dapat berupa perangkat desktop, laptop, tablet, atau smartphone yang terhubung ke jaringan lokal (LAN) atau internet. (Kurose & Ross, 2017)

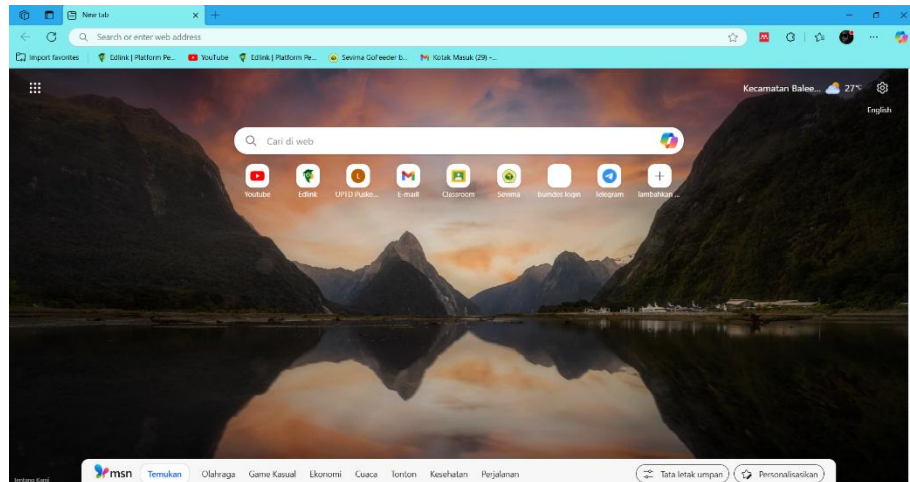
Dalam proyek aplikasi pendaftaran Puskesmas, komputer klien digunakan untuk mengakses antarmuka aplikasi berbasis web. Pengguna, seperti staf puskesmas atau pemilik hewan, dapat menggunakan komputer klien untuk mendaftarkan hewan, melihat jadwal konsultasi, atau mengelola data pendaftaran. Dengan koneksi internet melalui *Wi-Fi*, komputer klien mengirimkan permintaan ke server *Node.js*, yang kemudian memproses dan mengirimkan data kembali ke klien untuk ditampilkan di layar. Hal ini memastikan interaksi yang lancar antara pengguna dan sistem saat di jalankan.

### 3. *Tools Pengembangan Sistem*

Perangkat dan metode yang dirancang untuk mendukung perancangan, pengembangan, pengujian, dan pemeliharaan sistem secara efisien dan efektif. *Tools* yang digunakan diantaranya:

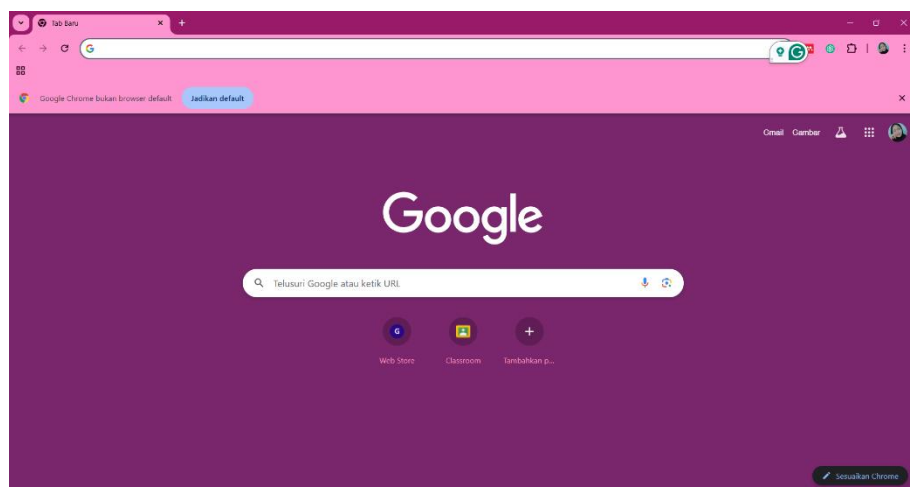
a. *Web browser*

*Browser web* adalah aplikasi yang diperlukan untuk mengakses layanan yang disediakan melalui web. Browser merupakan aplikasi yang banyak digunakan untuk mengakses informasi yang tersedia di internet. (Rasool & Jalil, 2020).



Gambar III. 5 Tampilan *Microsoft Edge*

Sumber : Screenshot Tampilan *Microsoft Edge*



Gambar III. 6 Tampilan *Google Chrome*

Sumber : Screenshot Tampilan *Google Chrome*

*Google Chrome* dan *Microsoft Edge* digunakan untuk menguji antarmuka aplikasi. Kedua browser ini dipilih untuk mendukung pengujian fitur memastikan aplikasi berjalan lancar.

#### b. HTML

HTML, singkatan dari *Hypertext Markup Language*, adalah bahasa standar web yang diatur oleh W3C (*World Wide Web Consortium*). HTML menggunakan tag-tag untuk membangun dan menyusun berbagai elemen dalam sebuah *website*..(Hidayat et al., 2019)

Tabel III. 1 Atribut HTML

Sumber : (Sholikhan et al., 2022)

Elmen	Deskripsi
<a> <i>Anchor</i>	Membuat <i>link</i> ke halaman lain atau bagian dari dokumen yang sama.
  Jeda baris	Memasuki jeda baris atau karakter kembali.
<form> <i>Form web</i>	Membuat <i>formulir web</i> untuk menerima <i>input</i> pengguna.
<h1> hingga <h6> <i>Heading</i>	Membuat wadah untuk judul, seperti teks judul.
<hr> <i>Hard rule</i>	Membuat garis horizontal.
<img> <i>Image</i>	Wadah untuk gambar.
<input> <i>Input</i>	Elemen untuk menerima <i>input</i> pengguna.
<link> <i>Link sumber</i>	Tautan ke sumber daya untuk halaman tersebut;
<p> Paragraf dalam halaman	Membuat paragraf tekstual atau area dan wadah lain untuk teks.
<script> Tak skrip	Menunjukkan skrip web atau program. Juga sering ditemukan di bagian kepala.
<span> Span	Membuat wadah untuk elemen. Sering digunakan bersama dengan informasi gaya.

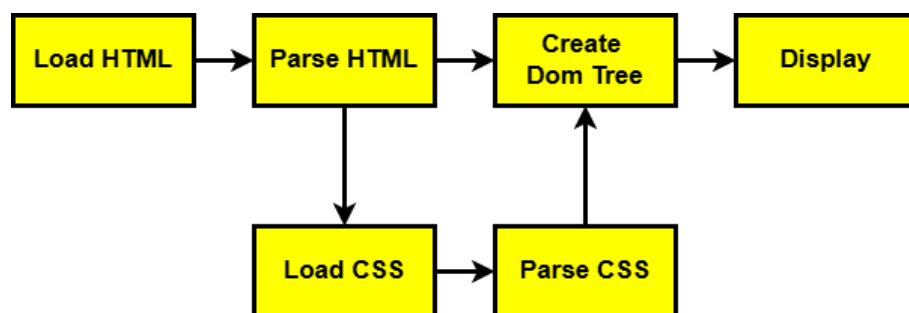
## c. NPM

NPM adalah pengelola paket untuk *JavaScript* yang digunakan untuk mengunduh, menginstal, dan mengelola library atau package yang diperlukan dalam pengembangan aplikasi. Dengan NPM, pengembang dapat mengakses ribuan *package open-source* yang tersedia di *registry* NPM, seperti *Express.js*, *Morgan*, dan *Bcrypt*. NPM juga membantu mengelola dependensi proyek secara efisien melalui file *package.json*. Dalam proyek Puskesmas, NPM digunakan untuk menginstal library yang mendukung pembuatan fitur-fitur aplikasi, seperti *Express.js* untuk *server*, *Morgan* untuk

*logging*, dan *Bcrypt* untuk pengamanan data pengguna.

d. CSS

CSS (*Cascading Style Sheets*) adalah bahasa desain yang digunakan untuk mengatur tampilan dan tata letak halaman web.(Rahmawati et al., 2018) Dengan CSS, pengembang dapat mengontrol elemen visual seperti warna, *font*, *margin*, *padding*, dan tata letak keseluruhan.(Marwanto Rahmatuloh & Muhammad Rizky Revanda, 2022). CSS juga mendukung desain responsif, memungkinkan halaman web menyesuaikan tampilannya dengan berbagai ukuran layar dan perangkat.



Gambar III. 7 Arsitektur CSS

Sumber : (Sholikhhan et al., 2022)

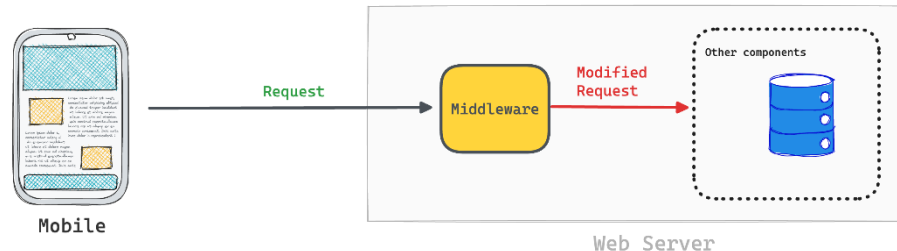
Dalam proyek aplikasi pendaftaran Puskesmas, CSS digunakan untuk mempermudah antarmuka pengguna. Dengan *framework* seperti *Bootstrap*, proyek ini memanfaatkan komponen siap pakai untuk menciptakan tampilan yang konsisten dan mudah diakses di berbagai perangkat.

e. *Dotenv*

*Dotenv* adalah *library* yang digunakan untuk mengelola variabel lingkungan (*environment variables*) melalui file *.env*. Variabel ini biasanya menyimpan informasi sensitif seperti URL *database*, API *keys*, atau konfigurasi *server*. Dalam proyek Puskesmas, *dotenv* digunakan untuk menyimpan konfigurasi seperti URL *postgreSQL* dan *secret key* untuk *token autentikasi*.

## f. Morgan

Morgan adalah *middleware logging* HTTP *Node.js*. Ia mencatat informasi penting tentang *request* dan *response*, seperti status, waktu untuk mempermudah *debugging* dan *monitoring* aplikasi.

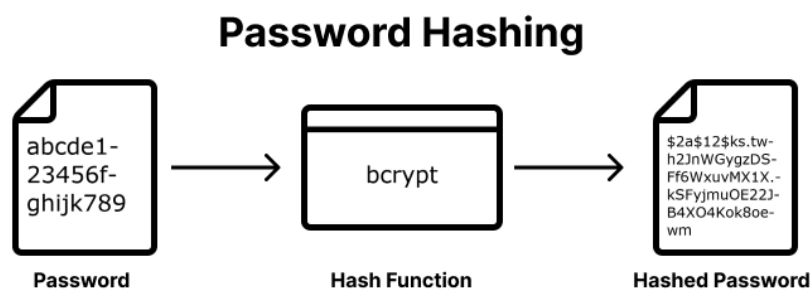


Gambar III. 8 Arsitektur Morgan

Sumber : Seyma Arslan, 2021

## g. Bcrypt

*Bcrypt* adalah *library hashing* untuk mengamankan data sensitif seperti *password*. Dengan algoritma *hashing* yang kuat, *Bcrypt* melindungi data pengguna dari ancaman seperti pencurian data.



Gambar III. 9 Arsitektur Bcrypt

Sumber : (Authgear, 2024)

Dalam proyek Puskesmas, *Bcrypt* digunakan untuk mengacak *password* pengguna sebelum disimpan di *database postgresQL*. Ini memastikan bahwa data pengguna aman.

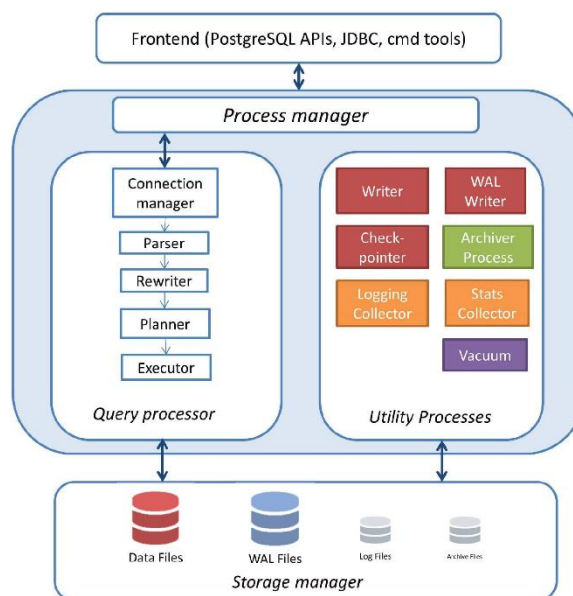
h. ORM (*Sequelize, TypeORM*)

ORM adalah teknik yang menghubungkan objek dalam kode dengan tabel di *database*, pengembang dapat bekerja dengan

*database* menggunakan sintaks pemrograman, tanpa harus menulis *query SQL* secara manual. Dalam proyek Puskesmas, digunakan untuk mempermudah pengelolaan *database*. Misalnya, untuk membuat model tabel seperti *Users* atau *Appointments*.

i. *PostgreSQL*

*PostgreSQL* adalah sistem manajemen basis data relasional open-source yang kuat dan andal. Ia mendukung fitur-fitur canggih seperti transaksi ACID (*Atomicity, Consistency, Isolation, Durability*) sistem ini memastikan bahwa setiap transaksi dilakukan secara aman dan konsisten tanpa risiko kehilangan atau kerusakan data. Selain itu, *PostgreSQL* memiliki kemampuan indexing yang efisien, memungkinkan pencarian dan pengolahan data yang lebih cepat serta optimal.



Gambar III. 10 Arsitektur *postgresql*

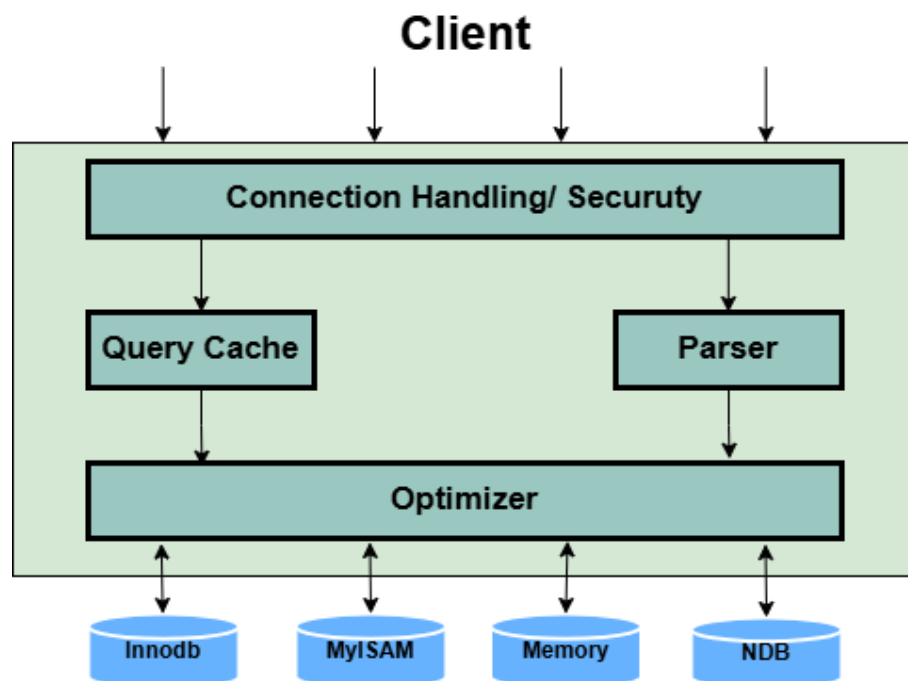
Sumber : (citusdata, 2019)

Dalam proyek Puskesmas, *postgresql* digunakan untuk menyimpan data pengguna, pendaftaran, dan jadwal konsultasi. Dengan struktur relasionalnya, *postgresql* memungkinkan

penyimpanan data yang terorganisir, seperti tabel untuk pengguna, hewan, dan jadwal.

j. *MySQL*

*MySQL* adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis.(Il-2, n.d.) Sebagai bagian dari RDBMS (*Relational Database Management System*), *MySQL* mendukung penggunaan bahasa pemrograman PHP. Selain itu, *MySQL* memiliki *query* atau bahasa SQL (*Structured Query Language*) yang sederhana serta menggunakan karakter escape yang serupa dengan PHP.(Hidayat et al., 2019)



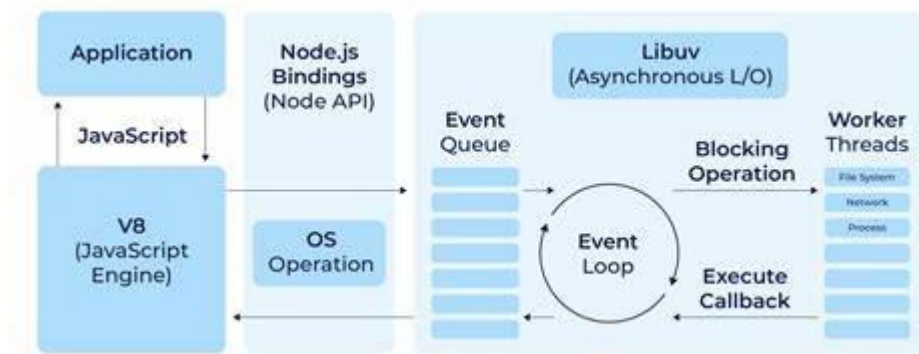
Gambar III. 11 *Arsitektur MySQL*

Sumber : accessed, 2025

k. *Node.js*

*Node.js* adalah platform runtime yang memungkinkan eksekusi kode *JavaScript* di luar browser. *Node.js* dirancang untuk membangun aplikasi server-side yang cepat dan skalabel, dengan model I/O *non-blocking* yang cocok untuk aplikasi berbasis jaringan. *Node.js* digunakan sebagai *backend* inti dalam proyek Puskesmas untuk memastikan respons cepat.



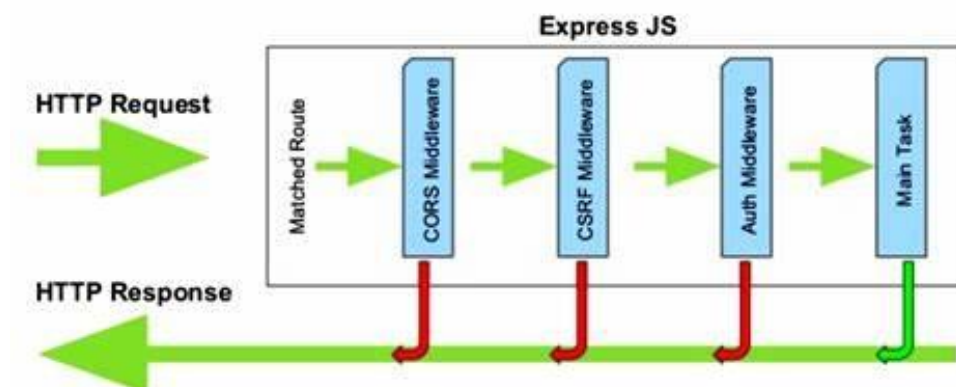


Gambar III. 12 Arsitektur Node.js

Sumber : Viacheslav Petrenko, 2024

### l. Express.js

*Express.js* adalah *framework* web untuk *Node.js* digunakan untuk membuat server dan API dengan cepat. *Express* menyediakan fitur *routing*, *middleware*, dan manajemen yang sederhana.



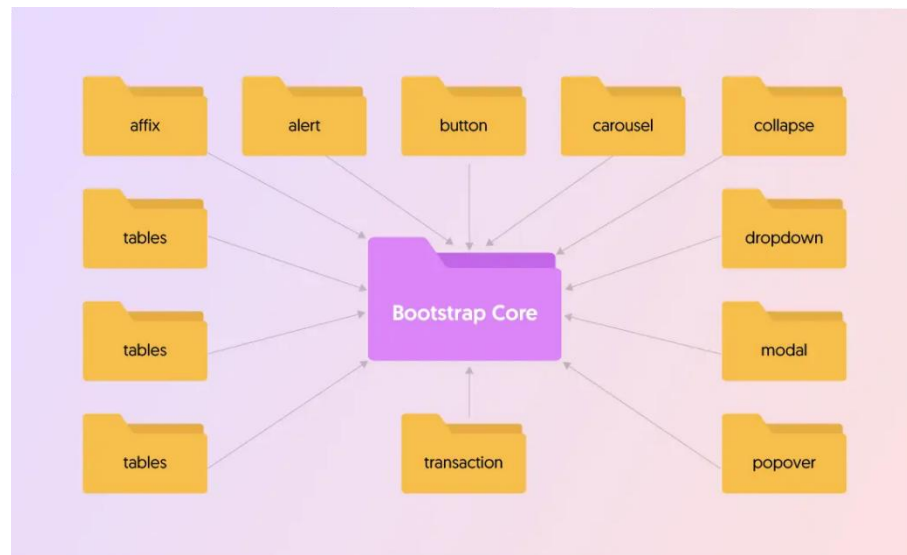
Gambar III. 13 Arsitektur Express.js

Sumber : Ankush Thakur, 2025

Express.js digunakan dalam proyek Puskesmas untuk membangun server aplikasi yang menjadi penghubung antara *frontend* dan *backend*. *Framework* ini bertugas mengelola rute-rute utama, seperti pendaftaran, pengambilan jadwal, dan *login* pengguna. Dalam proyek Puskesmas,

### m. Bootstrap

*Bootstrap* adalah *framework* CSS yang menyediakan komponen dan grid system untuk membangun antarmuka pengguna yang responsif dan modern. (Hidayat et al., 2019)



Gambar III. 14 *Arsitektur Bootstrap*

*Sumber : Ritwik Verma, 2024*

Dalam proyek Puskesmas, *Bootstrap* digunakan untuk mendesain antarmuka *form* pendaftaran, halaman jadwal konsultasi, dan *dashboard admin*. *Framework* dipilih untuk memastikan desain yang responsif.

#### 4. **Unified Modeling Language (UML)**





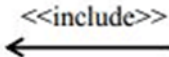
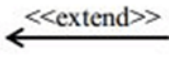
*Unified Modeling Language* (UML) adalah bahasa pemodelan standar yang digunakan untuk merancang dan menggambarkan sistem perangkat lunak secara visual. UML merupakan gabungan dari metode *Booch*, *Rumbaugh* (OMT) dan *Jacobson* (Marwanto Rahmatuloh & Muhammad Rizky Revanda, 2022). UML membantu dalam spesifikasi, konstruksi, dan dokumentasi sistem perangkat lunak, memfasilitasi komunikasi efektif antara pengembang dan pemangku kepentingan. (Il-2, n.d.) Dalam pengembangan aplikasi pendaftaran Puskesmas Pembantu Arjasari, UML digunakan untuk memodelkan berbagai aspek sistem, termasuk struktur data, alur proses, dan interaksi antar komponen. UML mencakup berbagai jenis diagram yang digunakan untuk memodelkan di antaranya:

### 1. *Use case Diagram*

Diagram ini memberikan gambaran tentang hubungan interaksi antara pengguna (aktor) dengan sistem, sekaligus menjelaskan fungsionalitas utama yang ditawarkan oleh sistem untuk mendukung kebutuhan pengguna.

Tabel III. 2 *Atribut Use case Diagram*

Sumber : (Noviantoro, 2022)

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor.
	Association : Abstaraksi dari penghubung antara aktor dengan <i>use case</i> .
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.



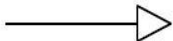


### 2. *Activity diagram*

Diagram aktivitas, atau yang dikenal sebagai *activity diagram*, digunakan untuk memvisualisasikan alur kerja (*workflow*) atau rangkaian aktivitas dalam suatu sistem, proses bisnis, atau menu

yang tersedia dalam perangkat lunak. Diagram ini berfokus pada penggambaran aktivitas yang dilakukan oleh sistem itu sendiri, bukan pada tindakan atau peran yang dilakukan oleh aktor. *Activity diagram* menyoroti aktivitas yang dapat dijalankan oleh sistem

Tabel III. 3 *Atribut Activity diagram*

Sumber : (Noviantoro, 2022)

Simbol	Keterangan
	Asosiasi/ <i>association</i> : Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertakan dengan <i>multiplicity</i> .
	Asosiasi berarah / <i>directed association</i> : Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi : Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus)
	Kebergantungan/ <i>dependency</i> : relasi antar kelas dengan makna kebergantungan antar kelas
	Agregasi/ <i>aggregation</i> : relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )






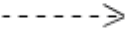
### 3. *Class Diagram*

*Class diagram* merupakan representasi visual yang menggambarkan struktur statis dari suatu sistem secara terperinci. Diagram ini mencakup berbagai kelas yang ada dalam sistem, beserta atribut-atributnya, metode-metode yang dimiliki, serta hubungan atau asosiasi antara kelas-kelas tersebut. Sebagai salah satu elemen penting dalam proses perancangan perangkat lunak,

*class* diagram berfungsi untuk membantu pengembang dalam menganalisis dan memahami kebutuhan sistem secara lebih mendalam. Selain itu, diagram ini juga berperan signifikan dalam mempermudah perancangan basis data dan struktur objek dalam aplikasi, sehingga memastikan bahwa setiap elemen dalam sistem dirancang secara terorganisir dan sesuai dengan kebutuhan.

Tabel III. 4 *Atribut Class Diagram*

Sumber : (Noviantoro, 2022)

Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descend</i> t) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
	<i>Class</i>	Himpunan dari objek- objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Realization</i>	Operasi yang benar benar dilakukan oleh suatu objek
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elmen mandiri

		( <i>independent</i> ) akan mempengaruhi elmen yang bergantung pada elmen yang tidak mandiri
—	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

## 5. *UI/UX Design*

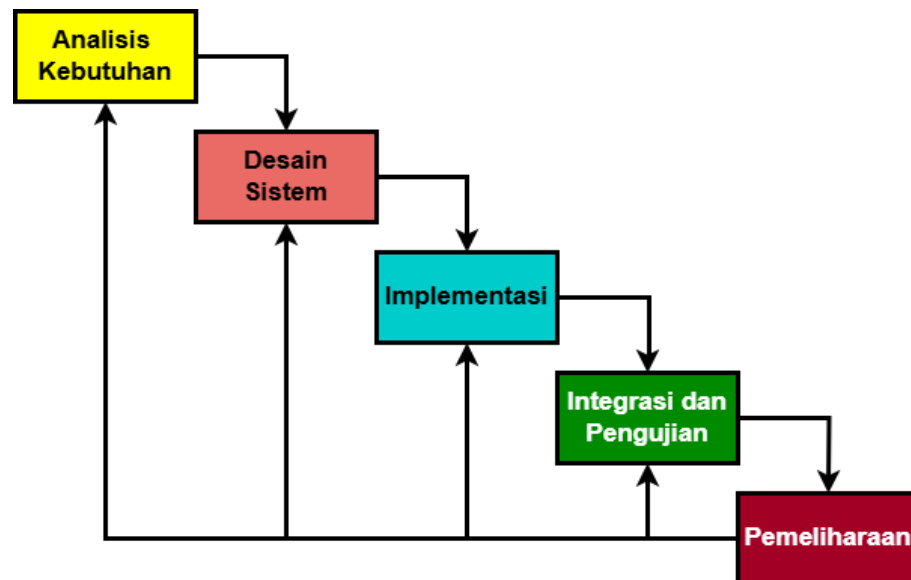
Antarmuka Pengguna (*User Interface*) dan Pengalaman Pengguna (*User Experience*) memegang peran krusial dalam pengembangan aplikasi. Desain aplikasi perlu dirancang dengan rapi dan terstruktur agar pengguna dapat dengan mudah mengakses serta memanfaatkan fitur-fitur yang tersedia.(Jamilah & Padmasari, n.d.) Selain itu, desain *UI/UX* harus dirancang sesuai dengan kebutuhan pengguna, mencakup elemen visual, fungsi-fungsi utama, dan berbagai aspek lainnya yang mendukung pengalaman pengguna secara keseluruhan.(Al-Faruq et al., 2022)

## 6. Perancangan *database*

*Database* adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data. *Database* adalah kumpulan data yang saling terhubung secara logis dirancang untuk menemukan informasi yang dibutuhkan oleh sebuah organisasi.(Pahlevi et al., 2018)

## 7. Metode

Metode pengembangan sistem yang diterapkan adalah metode *waterfall*, yang meliputi tahapan analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. (Marwanto Rahmatuloh & Muhammad Rizky Revanda, 2022)



Gambar III. 15 *Arsitektur Metode Waterfall*

*Sumber : (Abdul Wahid, 2020)*

Model *Waterfall* terdapat beberapa tahapan yang dapat diuraikan sebagai berikut (Al et al., 2022):

1) Analisis Kebutuhan

Proses pengumpulan data berdasarkan kebutuhan secara mendalam untuk memenuhi kebutuhan *user* agar dapat dipahami program yang sedang berjalan. (Farianto et al., 2014)

2) Design Sistem

Tahapan ini membuat design atau rancangan perangkat (Al et al., 2022) lunak (*Software*) termasuk struktur data, arsitektur *information Systems*, *epresentasi interface*, dan membuat perancangan model output dan input (Nuraeni, 2018).

3) Implementasi

Pada tahap ini perangkat lunak (*Software*) menggambarkan bagaimana program *system* ini bekerja, kemudian fungsi dapat dilaksanakan oleh *user* (Rahmawati et al., 2018). Teknik pemrograman yang diaplikasikan oleh *information Systems* akademik dengan menggunakan *programming language* yang dapat dipahami oleh mesin *computer* (Al et al., 2022).

#### 4) Pengujian

Tahapan ini selanjutnya menguji kode program. Tujuannya untuk memastikan agar tidak ada terjadinya *error* dan memastikan input yang dipakai menghasilkan *output* yang sesuai (Al et al., 2022).

#### 5) Pemeliharaan

Tahapan terakhir pada metode ini adalah proses ini dilakukan setelah *information Systems* telah dipakai oleh *user*. Perubahan akan dilaksanakan, jika terdapat *error* maka *information Systems* harus disesuaikan untuk kebutuhan yang diinginkan *user* (Al et al., 2022).



## **BAB IV**

### **PELAKSANAAN KERJA PRAKTIK**

#### **IV.1 *Input***

Secara keseluruhan, landasan teori yang diperoleh selama perkuliahan menjadi kontribusi berharga dalam mendukung proses pelaksanaan kerja. Pengetahuan dasar ini memainkan peran penting dalam memahami dan menguasai teknologi baru yang digunakan.

##### **IV.1.1. *User***

*User* berfungsi sebagai pihak yang memanfaatkan layanan untuk kebutuhan pendaftaran dan akses informasi.

###### **a. Halaman *Login/ Sign Up***

Halaman ini digunakan untuk mengautentikasi pengguna ke sistem. Pengguna diminta untuk memasukkan informasi *login*, seperti *email* dan *password*, untuk mendapat akses aplikasi dan bagi yang belum memiliki akun pengguna diarahkan ke halaman dimana pengguna harus mengisi data informasi lain sebagai syarat kelengkapan pasien yang di bawa seperti Nama, NIK, Alamat beserta Nomor Hp.

###### **b. Menu Pendaftaran**

Menu Pendaftaran membantu pengguna untuk memasukkan informasi terkait hewan yang akan diperiksa. Fitur ini memandu pengguna dalam melengkapi data yang diperlukan.

###### **c. Menu *Home***

Menu *home* adalah bagian dari tampilan navigasi yang menunjukkan nomor antrian setelah melakukan pendaftaran.

###### **d. Menu Riwayat**

Menu Riwayat adalah fitur yang berisi kumpulan catatan lengkap mengenai semua konsultasi yang pernah dilakukan oleh pengguna.

###### **e. Menu *Log Out***

Memungkinkan pengguna untuk keluar dari sistem dengan cara aman setelah selesai menggunakan layanan untuk melakukan pendaftaran.

#### **IV.1.2. Admin**

Admin bertugas mengelola data, memantau aktivitas sistem, serta memastikan semua fitur berjalan sesuai dengan kebutuhan pengguna.

##### **a. Menu *Login Admin***

Admin dapat mengakses halaman autentikasi sistem dengan terlebih dahulu memasukkan informasi *login* yang terdiri dari *email* dan kata sandi yang telah terdaftar. Setelah informasi tersebut dimasukkan dengan benar, mereka akan diarahkan untuk melanjutkan ke halaman berikutnya dalam sistem.

##### **b. Menu *Home***

Menu *home* ini dirancang untuk membantu admin dalam memantau permintaan konsultasi serta melihat detail konsultasi yang diajukan oleh pengguna yang telah mendaftar. Admin juga dapat memberikan persetujuan (*approved*) untuk menerima kunjungan berdasarkan permintaan tersebut.

##### **c. Menu *Manage Admin***

Fitur ini menampilkan informasi lengkap mengenai daftar admin yang memiliki izin untuk mengakses sistem. Setiap admin yang terdaftar dapat dilihat beserta data terkait, sehingga mempermudah pengelolaan dan pengawasan terhadap pihak-pihak yang memiliki otoritas dalam menjalankan dan mengelola aplikasi.

##### **d. Menu *Manage Consultation***

Menu ini dibuat untuk menampilkan catatan lengkap tentang riwayat konsultasi yang telah dilakukan oleh pengguna, dengan memberikan persetujuan (*Approved*) maupun penolakan (*Rejected*). Pada halaman ini, admin dapat mengakses detail informasi setiap konsultasi, termasuk tanggal konsultasi, keluhan yang disampaikan, serta data hewan dan pemilik yang terlibat dalam konsultasi tersebut.

##### **e. Menu *Log Out***

Memberikan kemampuan kepada admin untuk melakukan proses keluar dari sistem dengan cara yang aman setelah mereka selesai menggunakan layanan yang tersedia.

## **IV.2 Proses**

Pelaksanaan kerja mengenai perancangan aplikasi Pendaftaran Puskesmas Pembantu Arjasari diawali dengan pengenalan terhadap tempat kerja lalu diikuti beberapa tahap, yaitu tahap eksplorasi, berisikan perancangan web dan pelaporan hasil kerja eksplorasi yang dilakukan, selain pengenalan lingkungan pada tahap ini juga penulis mencari permasalahan yang ada di Puskesmas Pembantu Arjasari, tahap selanjutnya adalah tahap perancangan aplikasi, dan pelaporan hasil kerja tahap ini dilakukan oleh peserta selama kerja berlangsung.

### **IV.2.1. Analisis Kebutuhan**

Pada tahap awal, dilakukan identifikasi kebutuhan sistem melalui wawancara dan diskusi dengan pihak Puskesmas Pembantu Arjasari. Fokus utama adalah memahami kebutuhan pengguna, seperti proses pendaftaran, manajemen data pengguna, dan pembuatan laporan. Selain itu, dilakukan analisis sistem manual untuk mengidentifikasi masalah utama, seperti kurangnya efisiensi dalam pencatatan data dan risiko kehilangan informasi. Data yang dikumpulkan meliputi :

- a. Profil Pengguna: Informasi tentang siapa yang akan menggunakan sistem, seperti admin, staf, atau masyarakat umum. Data ini meliputi nama dan hak akses pengguna.
- b. Informasi tentang bagaimana proses pendaftaran layanan dilakukan secara manual atau dengan sistem yang ada.
- c. Informasi jenis data yang akan dimasukkan ke dalam sistem, seperti data pengguna, data hewan, dan data pendaftaran layanan.
- d. Informasi tentang seberapa sering layanan dilakukan dan berapa banyak pengguna yang dilayani serta informasi mengenai perangkat lunak yang sedang digunakan, seperti aplikasi lain.

Perancangan aplikasi pendaftaran di Puskesmas Pembantu Arjasari mencakup beberapa jenis data yang diolah untuk keperluan perekapan, antara lain sebagai berikut:

#### A. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk perancangan aplikasi Pendaftaran Puskesmas Pembantu Arjasari ini, penyusun menggunakan laptop maka dibutuhkan seperangkat *computer* dengan spesifikasi sebagai berikut :

Tabel IV. 1 Kebutuhan Perangkat Keras

No	Item	Spesifikasi
1	<i>Processor</i>	Intel i5-1135G7
2	Kecepatan	4.2 GHz
3	RAM	16 GB
4	<i>Storage</i>	475 GB
5	<i>System</i>	64-bit x64

#### B. Minimum kebutuhan perangkat keras

Untuk memastikan sistem informasi berbasis web dapat berfungsi dengan baik, diperlukan komputer dengan spesifikasi minimum tertentu. Spesifikasi ini meliputi perangkat keras dan perangkat lunak yang mampu mendukung performa sistem secara optimal, seperti :

Tabel IV. 2 Minimum Kebutuhan Perangkat Keras

No	Item	Spesifikasi
1	<i>Processor</i>	Intel i3
2	Kecepatan	2.40 GHz
3	RAM	4 GB
4	<i>Storage</i>	250 GB
5	<i>System</i>	32-bit x64

#### C. Kebutuhan Perangkat Lunak

Perangkat lunak atau peranti lunak adalah istilah khusus untuk data yang diformat dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak

berwujud. Perangkat lunak yang digunakan dalam perancangan aplikasi pembukuan perpustakaan berbasis web ini adalah sebagai berikut

Tabel IV. 3 Kebutuhan Perangkat Lunak

No	Item	Spesifikasi
1	Sistem Operasi	<i>Windows 11 pro 64 bit</i>
2	Koneksi Internet	<i>Wifi</i>
3	Bahasa Pemograman	<i>HTML, CSS, Bootstrap, Java script (Node Js)</i>
4	DBMS	<i>MySQL</i>
5	<i>Web Browser</i>	<i>Google Chrome, Edge</i>
6	<i>Code Editor</i>	<i>Visual Studio Code</i>
7	<i>Web Server</i>	<i>postgreSQL</i>

Data diatas merupakan perangkat yang digunakan dalam pembuatan aplikasi pendaftaran puskesmas pembantu Arjasari berbasis web akan tetapi harus digaris bawahi bahwa pada pada kerja praktek kali ini hanya dilakukan sampai tahap perancangan.

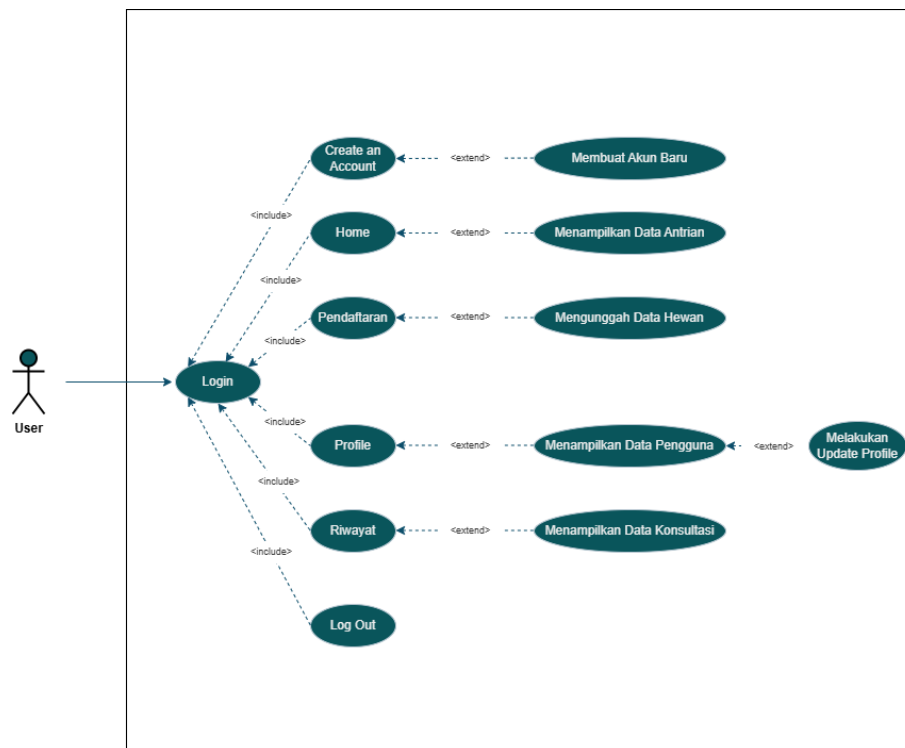
#### IV.2.2. Perancangan Perangkat Lunak

Setelah kebutuhan sistem dipahami dengan baik, tahap berikutnya adalah merancang sistem secara detail. Desain sistem meliputi perancangan arsitektur perangkat lunak, desain antarmuka pengguna, dan pemodelan proses menggunakan *Unified Modelling Language (UML)*, seperti diagram *use case*, diagram aktivitas, dan *class diagram*. Pada tahap ini juga dipilih teknologi yang akan digunakan.

##### 1) *Use case User dan Admin*

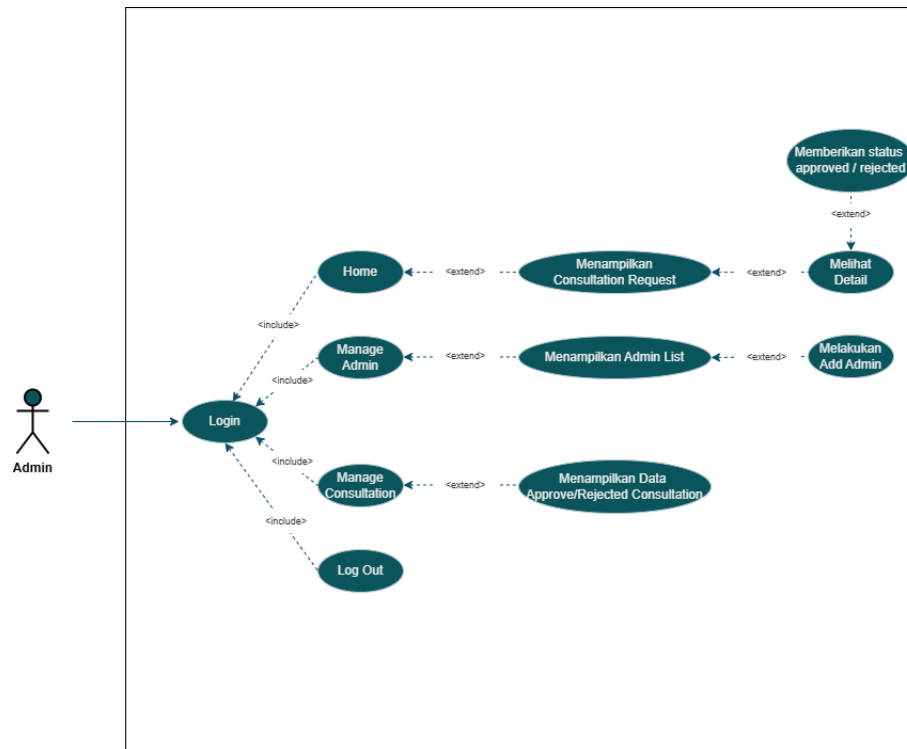
Diagram *use case* yang diterapkan dalam pengembangan aplikasi pendaftaran Puskesmas Pembantu (Pustu) Arjasari berbasis web dirancang untuk menggambarkan interaksi antara pengguna *user* dan *admin* dengan berbagai fitur dalam sistem. Diagram ini berfungsi untuk

memvisualisasikan dan menyoroti peran masing-masing aktor, baik pengguna maupun admin.



Gambar IV. 1 Use case User

Pada rancangan aplikasi pendaftaran pelayanan Pusat Kesehatan Hewan Pembantu Arjasari memiliki *actor* atau *user* yang bisa melakukan hal hal berikut. Ketika *user* berhasil melakukan *login* maka system akan menampilkan menu menu yang ada pada *dashboard* diantaranya. Dashboard aplikasi menampilkan menu-menu seperti Buat akun baru (*Create an Account*) untuk pendaftaran akun baru bagi pengguna yang ingin melakukan konsultasi, *Home* untuk melihat informasi nomor antrian, serta Pendaftaran untuk mengunggah data hewan yang akan diperiksa. Selain itu, terdapat menu *Profile* menyediakan akses untuk melihat dan memperbarui (*Update*) data pribadi pengguna, Riwayat untuk melihat catatan konsultasi yang telah dilakukan pengguna serta konsultasi yang di lakukan sebelumnya, dan *Logout* untuk keluar dari aplikasi.



Gambar IV. 2 Use case Admin

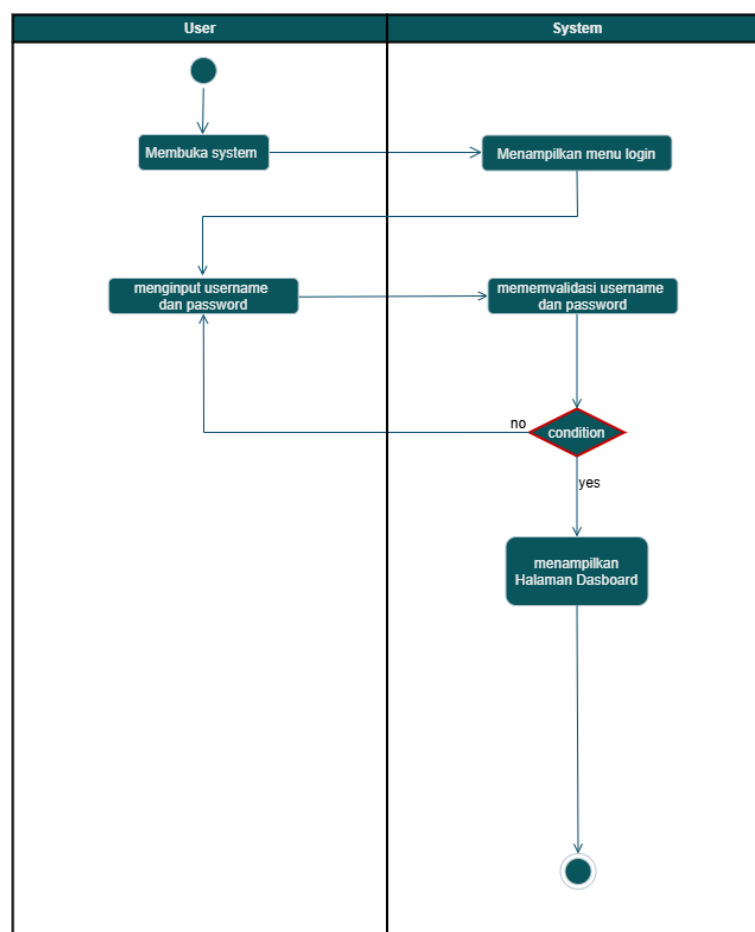
Dalam rancangan aplikasi pendaftaran pelayanan UPTD Puskesmas Pembantu (Pustu) Arjasari, terdapat aktor atau pengguna dengan peran admin yang memiliki kemampuan untuk melakukan berbagai tindakan atau pengelolaan tertentu dalam sistem. Menu *Home* menampilkan informasi permintaan konsultasi pengguna setelah pendaftaran hewan peliharaan di mana admin dapat memberikan status disetujui (*Approved*) atau ditolak (*Rejected*). Menu *Manage Admin* memberikan kemampuan untuk mengakses dan mengelola data admin lain, termasuk menambahkan data admin baru agar dapat memiliki akses ke sistem. Fitur ini mencakup pengelolaan informasi pribadi dan hak akses setiap admin, sehingga memastikan bahwa sistem dikelola secara terorganisir dan hanya pihak yang berwenang memiliki akses ke fitur-fitur tertentu. Menu *Manage Consultation* digunakan untuk melihat catatan konsultasi yang telah diproses beserta status disetujui atau ditolak. Selain itu, menu *Logout* untuk admin keluar dari aplikasi dengan aman.

## 2) Activity diagram

Berikut adalah *activity diagram* yang dibuat dalam perancangan aplikasi pembukuan perpustakaan. Diagram ini mencakup berbagai aktivitas yang terdapat dalam sistem, yang dapat diakses oleh pengguna dengan peran sebagai *user* maupun *admin*. Diagram ini menggambarkan alur proses secara rinci untuk setiap aktivitas yang dilakukan dalam sistem.

### 1. User

#### a. Activity diagram Menampilkan Halaman Login



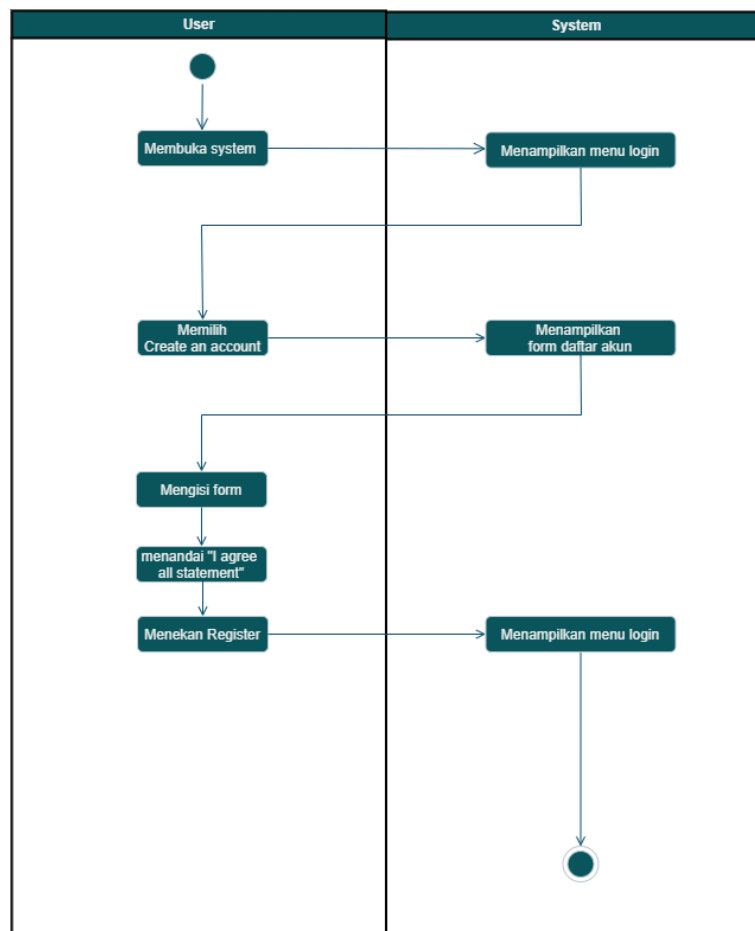
Gambar IV. 3 Activity diagram Login

Pada *activity diagram* untuk menampilkan halaman *login*, proses dimulai ketika pengguna membuka sistem. Kemudian menampilkan halaman *login*, di mana pengguna diminta untuk memasukkan *username* dan *password*. Setelah data dimasukkan, sistem akan



memvalidasi apakah *username* dan *password* yang diberikan benar. Jika terdapat kesalahan, pengguna akan diarahkan kembali memasuki ke halaman *login* hingga mencaoi validasi berhasil. Setelah itu, sistem akan menampilkan halaman *dashboard* kepada pengguna.

b. *Activity diagram* Buat Akun Baru (*Create an Account*)

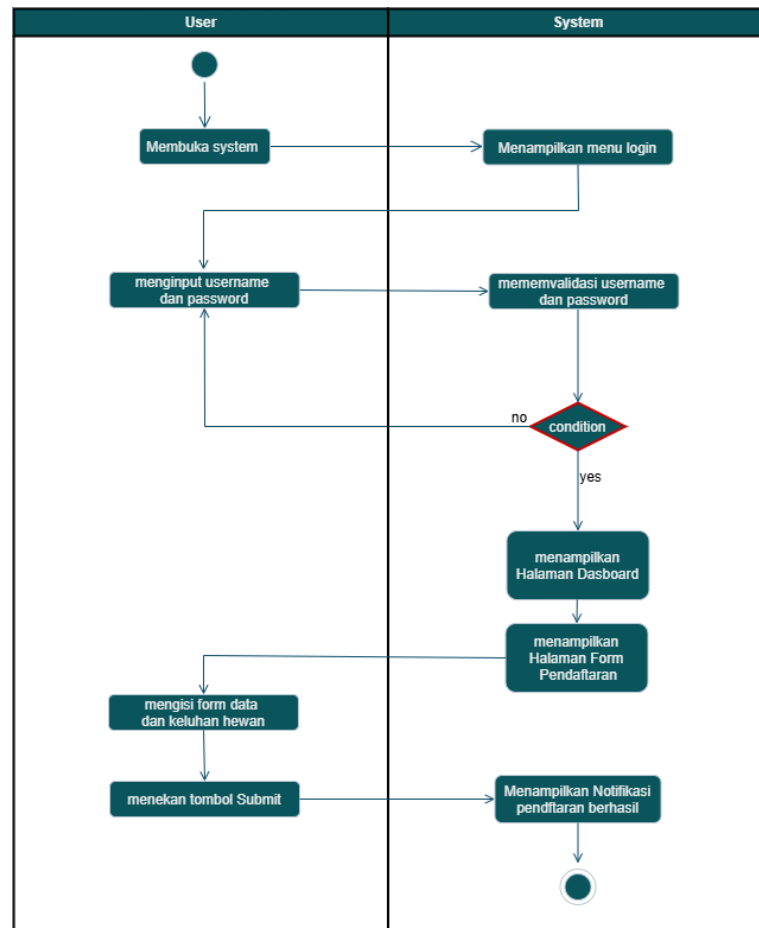


Gambar IV. 4 *Activity diagram* Buat Akun Baru

Pada *activity diagram* untuk membuat akun (*Create an Account*), proses dimulai ketika pengguna membuka sistem, dan sistem menampilkan halaman *login*. Selanjutnya, pengguna memilih menu *Create an Account*, yang mengarahkan sistem untuk menampilkan formulir pendaftaran akun. Di halaman ini, pengguna harus mengisi data pribadi yang diminta dan menyetujui bahwa semua data sudah

benar dengan menandai kotak "*I agree all statement.*" Setelah itu, pengguna menekan tombol *Register*, dan sistem kembali menampilkan halaman *login* sebagai konfirmasi bahwa proses pendaftaran telah berhasil.

c. *Activity diagram* Melakukan Pendaftaran

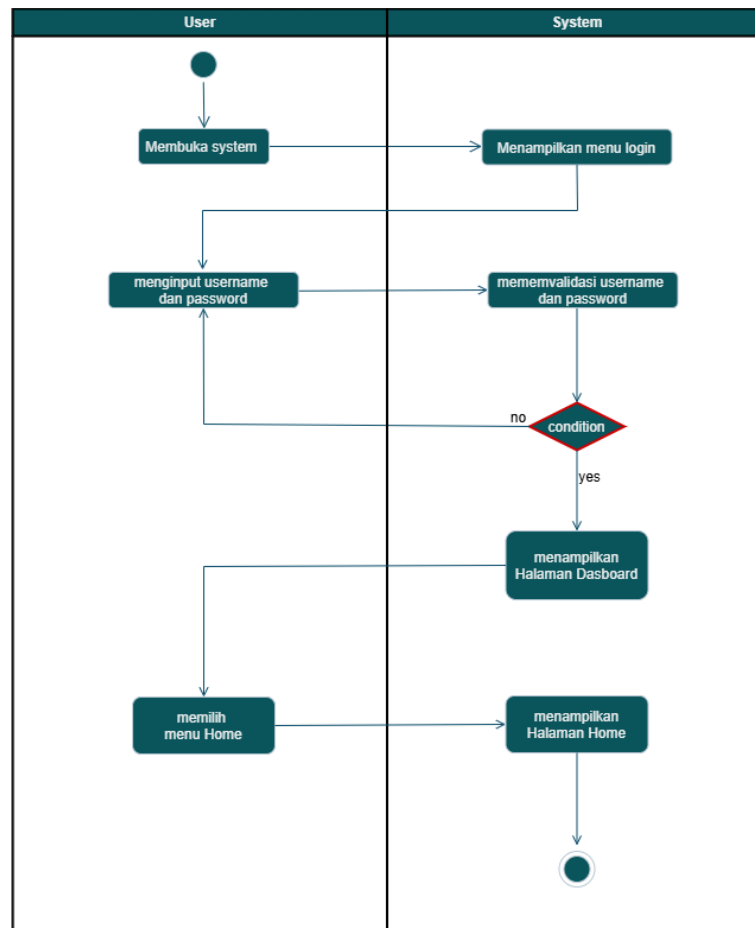


Gambar IV. 5 *Activity diagram* Pendaftaran

Pada *activity diagram* pendaftaran, proses dimulai ketika pengguna membuka sistem atau aplikasi, dan sistem menampilkan halaman *login*. Pengguna diminta memasukkan *username* dan *password*, yang akan divalidasi sistem. Jika *username* atau *password* salah, pengguna diarahkan kembali ke *login* hingga validasi berhasil. Setelah *login*, pengguna diarahkan ke dashboard yang menampilkan menu pendaftaran. Di sini, pengguna diminta mengisi data serta

keluhan terkait hewan yang akan diperiksa. Setelah data diisi, pengguna menekan tombol *Submit*, dan sistem menampilkan notifikasi "Berhasil" sebagai tanda proses pendaftaran selesai.

d. *Activity diagram* Menampilkan Halaman *Home*

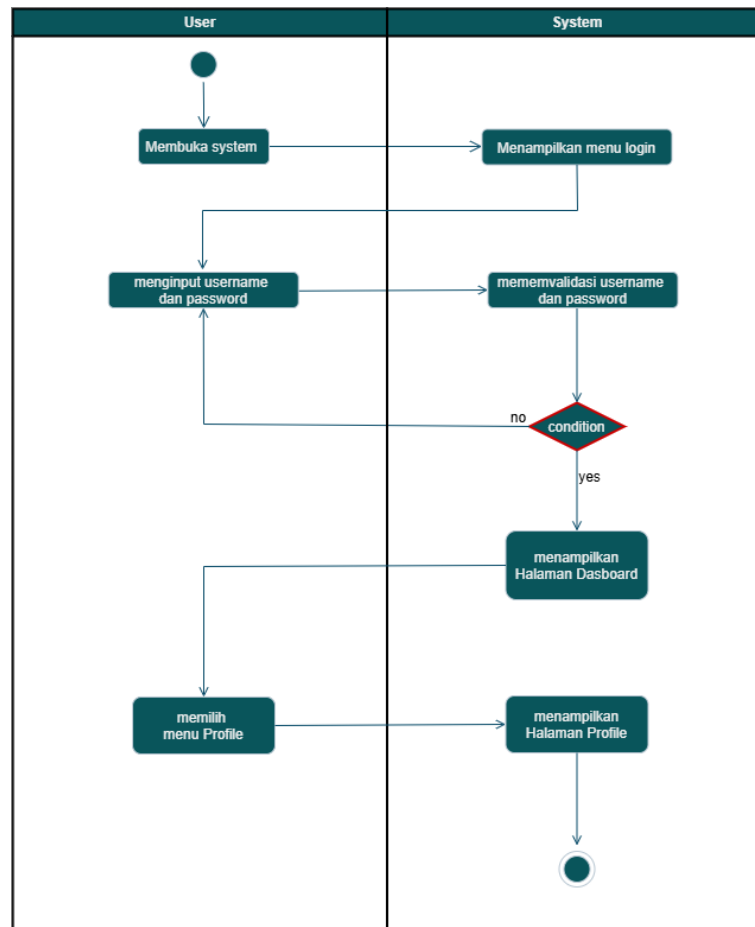


Gambar IV. 6 *Activity diagram* Home

Pada aktivitas di halaman *Home*, pengguna membuka sistem kemudian menampilkan menu *login*. Pengguna diminta untuk memasukkan *username* dan *password*. Setelah memasukkan informasi, sistem akan memeriksa validitas *username* dan *password*. Jika salah satu tidak sesuai, sistem akan mengarahkan pengguna kembali ke halaman *login* hingga validasi berhasil. Setelah validasi berhasil, sistem menampilkan *dashboard*. Di halaman *dashboard*, pengguna dapat memilih menu *Home* untuk

mengakses halaman utama. Sistem kemudian menampilkan halaman utama yang berisi daftar antrian.

e. *Activity diagram Menampilkan Halaman Profile*

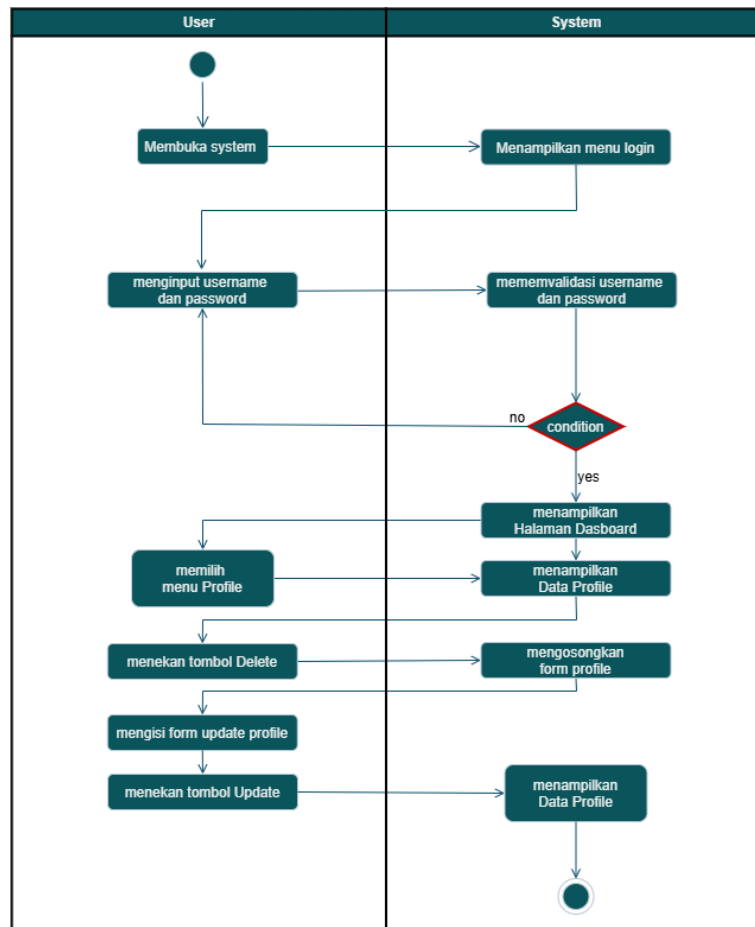


Gambar IV. 7 Activity diagram Profile

Pada aktivitas diagram, untuk langkah pertama pengguna membuka sistem, kemudian sistem menampilkan menu *login*. Pengguna diminta untuk memasukkan *username* dan *password*. Setelah data tersebut diinputkan, sistem memvalidasi *username* dan *password* yang dimasukkan adalah benar. Jika terdapat kesalahan pada *username* atau *password*, sistem mengarahkan pengguna kembali ke halaman *login* hingga validasi berhasil. Setelah berhasil melewati validasi, sistem menampilkan halaman *dashboard*. Di mana,

pengguna dapat memilih menu *Profile* dan sistem akan menampilkan halaman profil pengguna.

f. *Activity diagram Melakukan Update Profile*

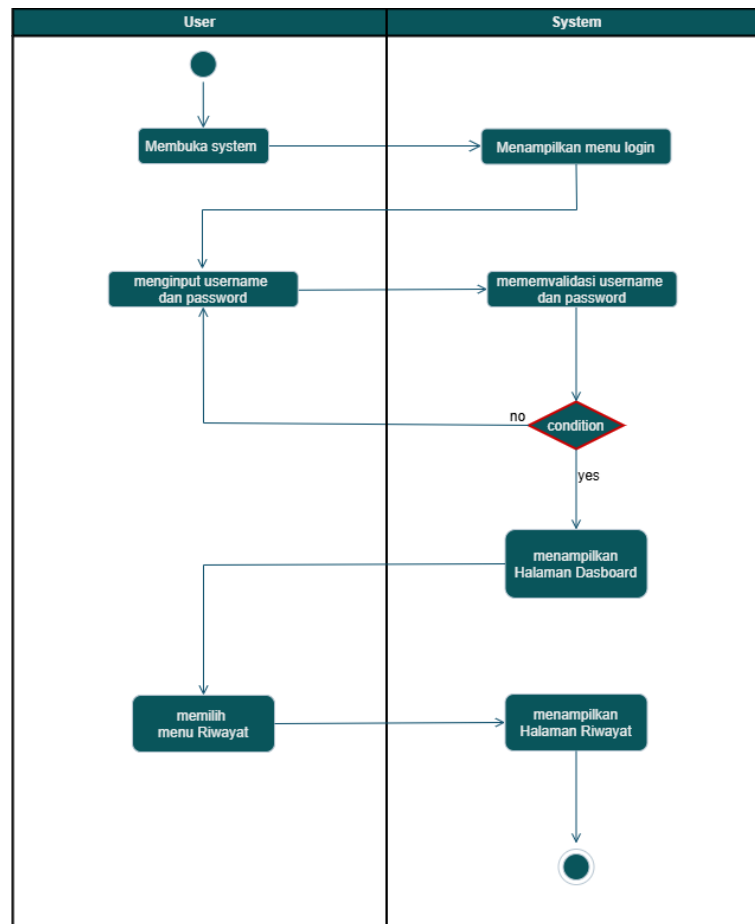


Gambar IV. 8 *Activity diagram Update Profile*

Pada *activity diagram update Profile*, pengguna membuka sistem kemudian menampilkan halaman *login*. Pengguna harus memasukkan *username* dan *password*, kemudian akan divalidasi. Jika *username* atau *password* salah, pengguna diarahkan kembali ke halaman *login* hingga validasi berhasil. Setelah berhasil *login*, sistem menampilkan halaman *dashboard*, di mana pengguna dapat memilih menu *Profile*. Sistem kemudian menampilkan halaman profil yang berisi data pribadi pengguna. Pengguna dapat menekan tombol *Delete* untuk mengosongkan formulir, lalu mengisi ulang

formulir dengan data baru. Setelah selesai, pengguna menekan tombol *Update*, dan sistem akan memperbarui data profile baru

g. *Activity diagram* Menampilkan Halaman Riwayat

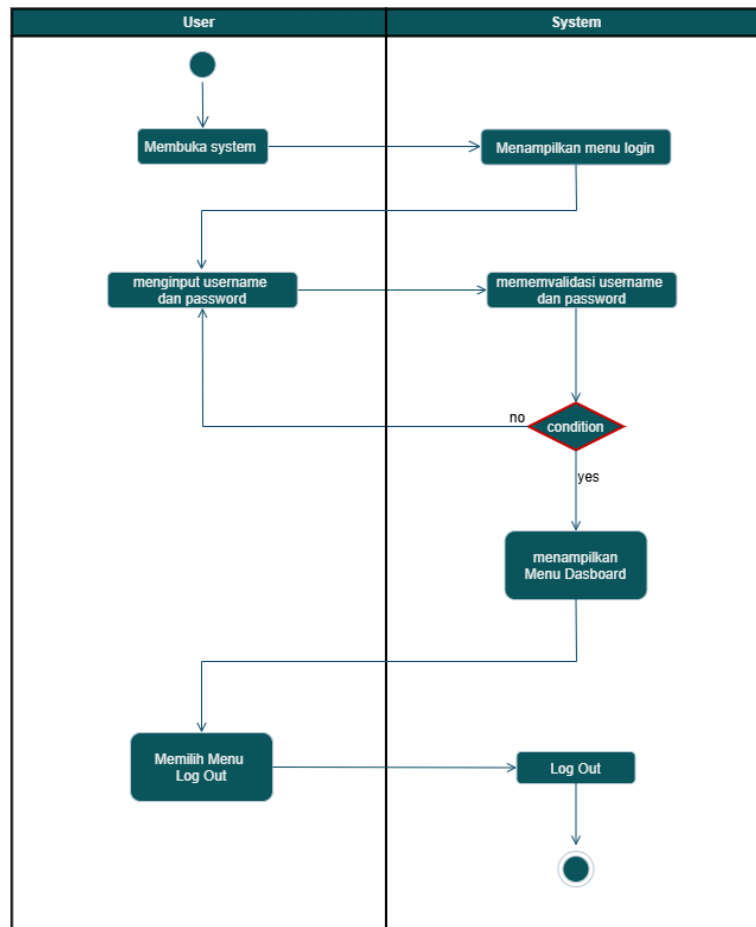


Gambar IV. 9 *Activity diagram* Riwayat

Pada *activity diagram* untuk menampilkan menu riwayat, pengguna terlebih dahulu membuka sistem atau aplikasi, yang kemudian menampilkan halaman *login*. Pengguna diminta memasukkan *username* dan *password*. Kemudian, sistem akan melakukan validasi untuk memastikan kebenaran *username* dan *password*. Jika terdapat kesalahan, pengguna akan diarahkan kembali ke halaman *login* hingga proses validasi berhasil. Setelah berhasil *login*, sistem akan menampilkan halaman dashboard. Di sana, pengguna dapat

memilih menu riwayat, dan sistem akan menampilkan halaman yang berisi informasi riwayat sesuai permintaan pengguna.

h. *Activity diagram Logout*

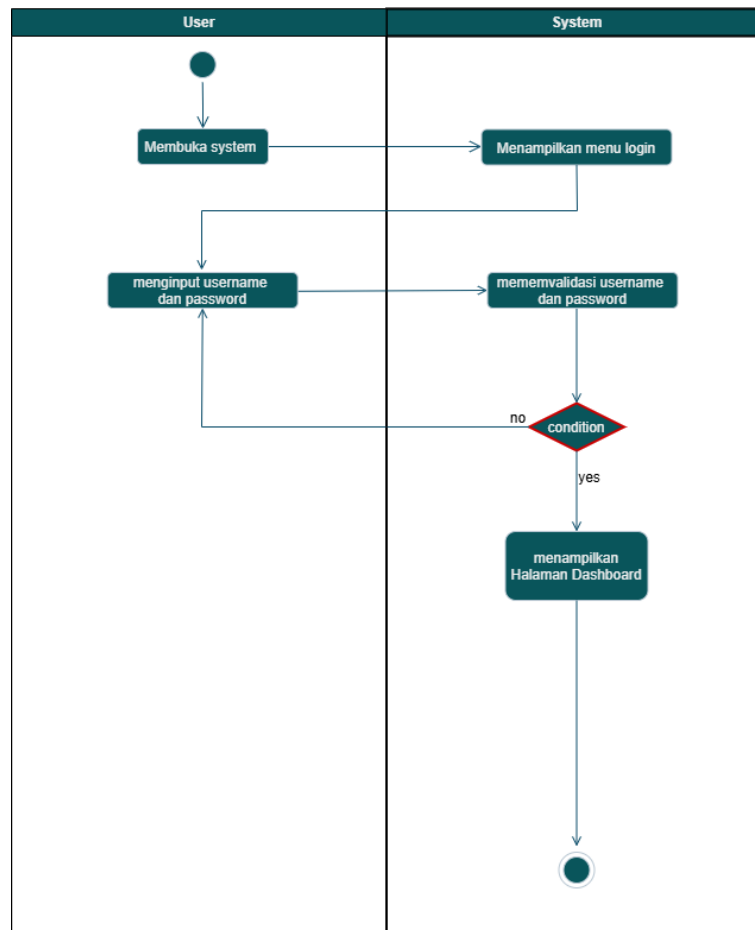


Gambar IV. 10 *Activity diagram Logout*

Pada *activity diagram logout*, pengguna membuka sistem, kemudian menampilkan halaman *login*. Pengguna diminta untuk memasukkan *username* dan *password*. Setelah itu, sistem memvalidasi *username* dan *password*. Jika terdapat kesalahan, sistem akan mengarahkan pengguna kembali ke halaman *login* hingga validasi berhasil. Setelah lolos validasi, sistem menampilkan halaman *dashboard*, di mana pengguna dapat memilih menu *logout*. Sistem kemudian memproses permintaan *logout* dan mengakhiri sesi pengguna.

## 2. Admin

### a. Activity diagram Login Admin

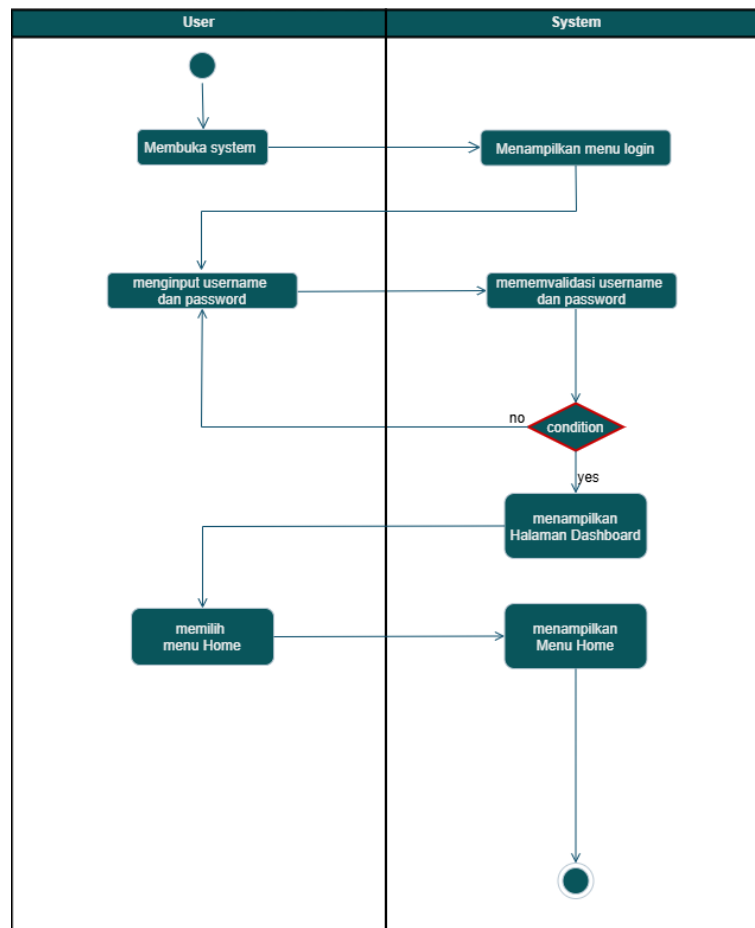


Gambar IV. 11 Activity diagram Login Admin

Pada *activity diagram login admin*, proses dimulai ketika pengguna membuka sistem atau aplikasi, yang secara otomatis menampilkan halaman *login* khusus untuk admin. Pada halaman ini, pengguna diminta untuk mengisi kolom *username* dan *password* yang sesuai. Setelah data dimasukkan, sistem akan melakukan proses validasi untuk memastikan kebenaran informasi tersebut. Jika ditemukan kesalahan pada *username* atau *password*, sistem akan mengarahkan pengguna kembali ke halaman *login* hingga data yang dimasukkan valid. Ketika validasi berhasil dilakukan, sistem akan mengalihkan pengguna ke halaman *dashboard*.



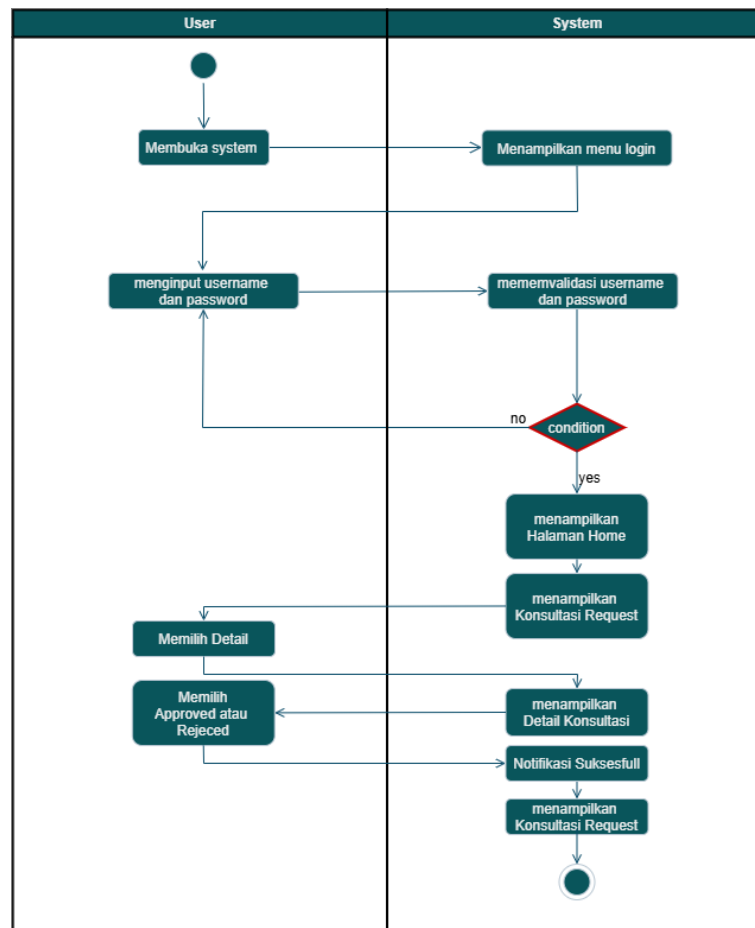
b. *Activity diagram Menampilkan Home Admin*



Gambar IV. 12 *Activity diagram Home Admin*

Pada *activity diagram* menampilkan halaman *home admin*, pengguna terlebih dahulu membuka sistem atau aplikasi, yang menampilkan halaman *login admin*. Pada halaman ini, pengguna diminta untuk memasukkan *username* dan *password* untuk dapat memasuki halaman selanjutnya. Selanjutnya, sistem akan memvalidasi data untuk memastikan keakuratan *username* dan *password* yang dimasukkan benar. Jika salah, pengguna akan diarahkan kembali ke halaman *login* hingga data yang dimasukkan berhasil validasi. Setelah validasi berhasil, sistem menampilkan halaman *dashboard*, kemudian pengguna memilih menu *home* dan sistem menampilkan menu *home admin*.

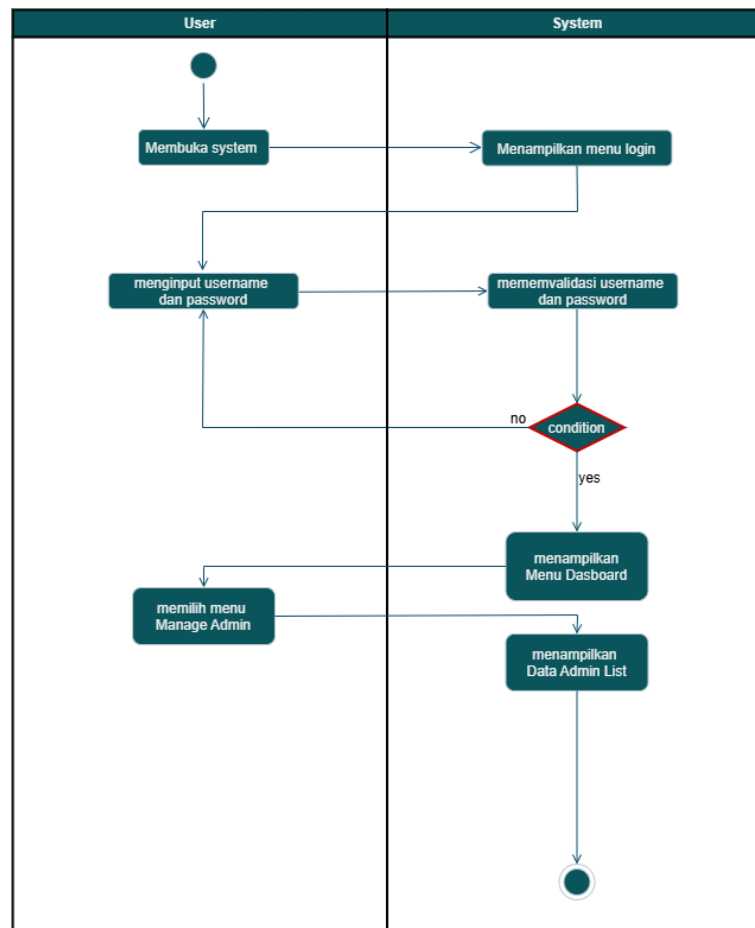
c. Activity diagram Melakukan Apprpve Rejected Consultation



Gambar IV. 13 Activity diagram Approved Rejected Consultation

Pada diagram aktivitas untuk persetujuan atau penolakan permintaan konsultasi, pertama kali membuka sistem menampilkan halaman *login admin*. Pengguna diminta memasukkan *username* dan *password*., sistem melakukan verifikasi memastikan data benar. Jika salah, pengguna diarahkan kembali ke *login* hingga validasi berhasil. Kemudian, sistem menampilkan *dashboard* dan menu *home admin*. Pada halaman ini, pengguna kemudian memilih detail untuk melihat informasi lebih lanjut tentang konsultasi. Pengguna memiliki opsi untuk menyetujui (*approve*) atau menolak (*reject*) permintaan. Setelah itu, sistem menampilkan notifikasi sukses, kemudian sistem kembali ke tampilan daftar permintaan konsultasi.

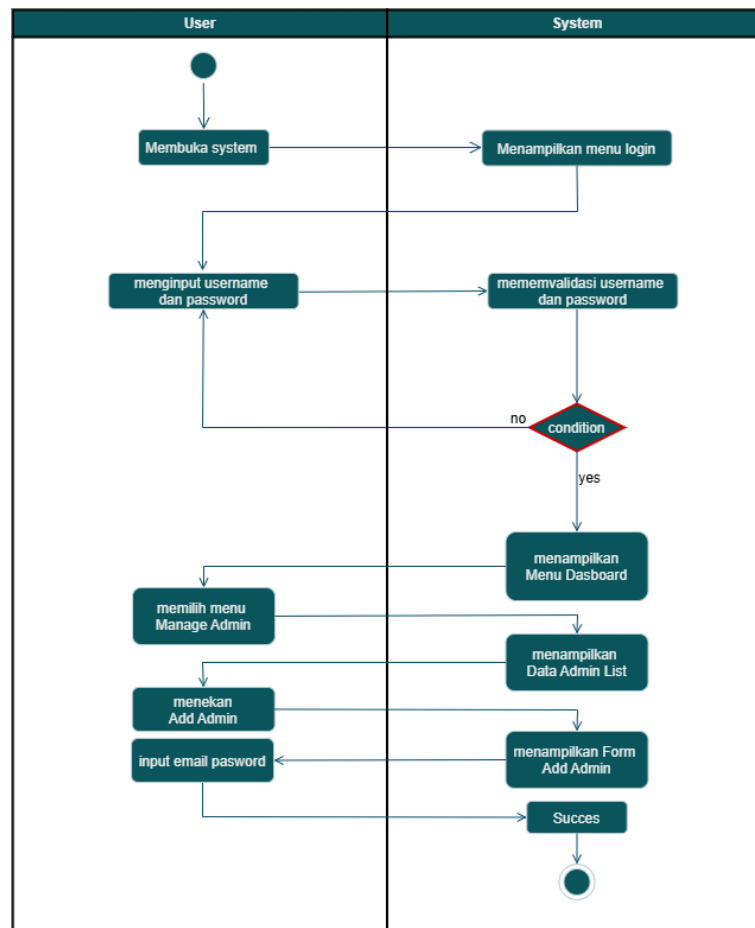
d. Activity diagram Menampilkan Manage Admin



Gambar IV. 14 Activity diagram Manage Admin

Pada diagram aktivitas untuk menampilkan pengelolaan admin, pertama kali yang dilakukan pengguna adalah membuka sistem atau aplikasi yang akan menampilkan halaman *login admin*. Pengguna diminta untuk memasukkan *username* dan *password* untuk memasuki halaman berikutnya. Setelah itu, sistem akan memverifikasi keakuratan *username* dan *password* yang dimasukkan benar. Jika terjadi kesalahan, pengguna akan diarahkan kembali ke halaman *login* hingga proses validasi berhasil dilakukan. Setelah *login* berhasil, sistem akan menampilkan halaman *dashboard*, dan pengguna dapat memilih menu pengelolaan admin. Sistem kemudian akan menampilkan halaman daftar data admin.

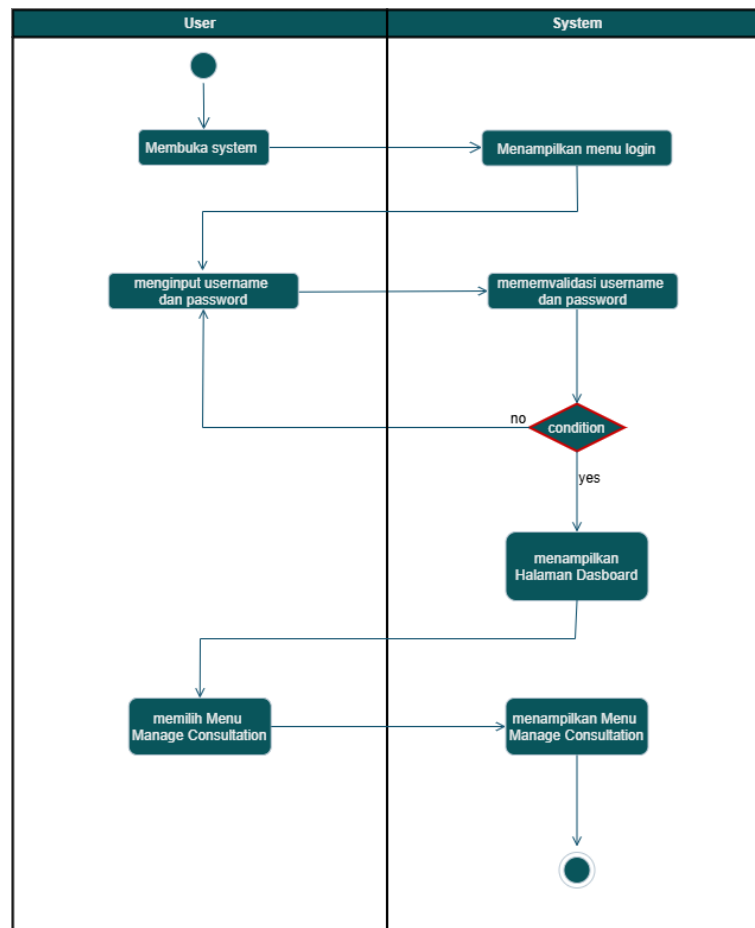
e. Activity diagram Melakukan Add admin



Gambar IV. 15 Activity diagram Add Admin

Pada diagram aktivitas *add admin*, pengguna membuka sistem yang menampilkan *login admin*. Pengguna memasukkan *username* dan *password*. Setelah itu, sistem memverifikasi *username* dan *password*. Jika terjadi kesalahan, pengguna diarahkan kembali ke halaman *login* hingga proses validasi berhasil. Setelah berhasil, sistem menampilkan halaman *dashboard*, dan pengguna memilih menu *Manage Admin*. Sistem kemudian menampilkan halaman daftar data admin. Pengguna kemudian menekan tombol *Add Admin* untuk menambah admin baru, dengan menginputkan *username* dan *password* pada form. Setelah itu, sistem menampilkan notifikasi sukses sebagai tanda admin baru telah berhasil ditambahkan.

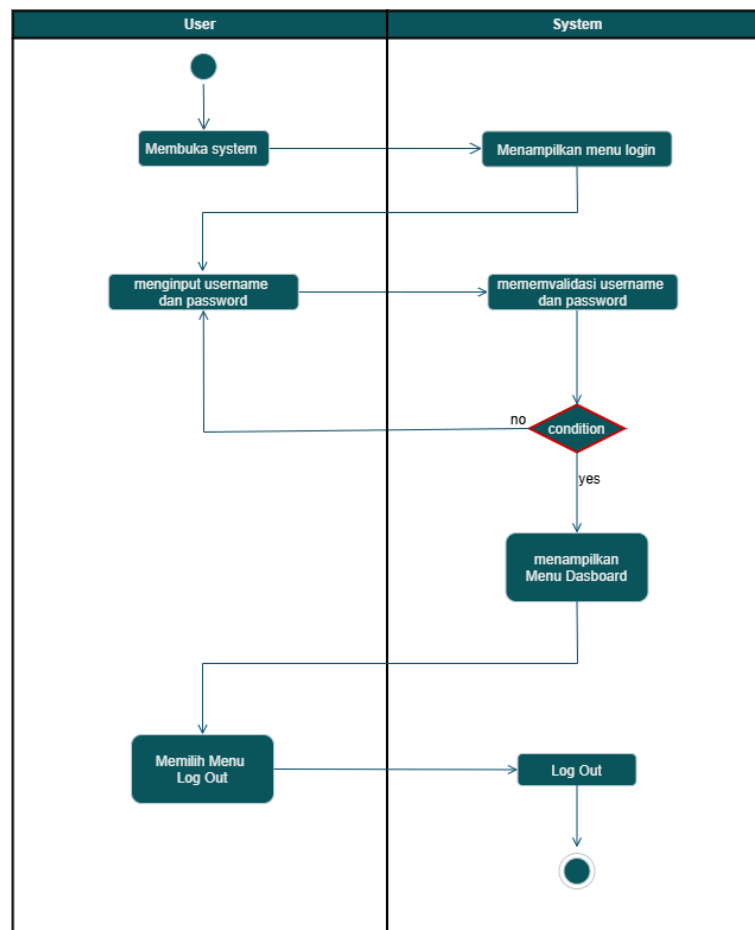
## f. Activity diagram Menampilkan Manage Consultation



Gambar IV. 16 Activity diagram Manage Consultation

Pada diagram aktivitas menampilkan halaman *Manage Consultation*, pengguna membuka sistem atau aplikasi yang menampilkan halaman *login* admin. Pengguna diminta untuk memasukkan *username* dan *password*. Setelah itu, sistem memverifikasi *username* dan *password*. Jika terjadi kesalahan, pengguna diarahkan kembali ke halaman *login* hingga proses validasi berhasil. Setelah berhasil, sistem menampilkan halaman *dashboard*, dan pengguna memilih menu *Manage Consultation*. Sistem kemudian akan menampilkan halaman *Manage Consultation* yang didalamnya terdapat daftar konsultasi yang telah dilakukan seluruh pasien.

## g. Activity diagram Melakukan Log Out

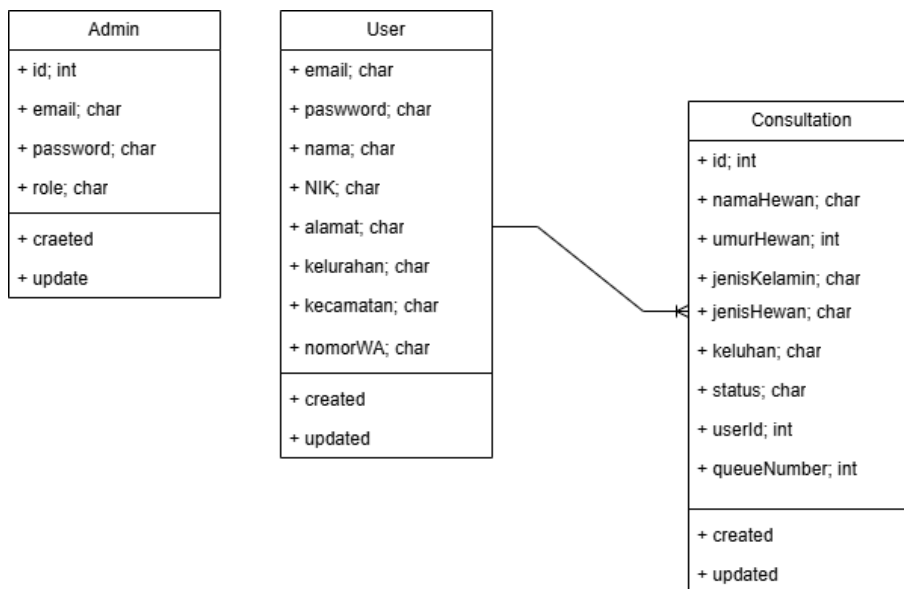


Gambar IV. 17 Activity diagram Log Out Admin

Pada diagram aktivitas untuk proses *logout*, pengguna memulai dengan membuka sistem atau aplikasi yang menampilkan halaman *login admin*. Pengguna kemudian memasukkan *username* dan *password*. Sistem memverifikasi data yang dimasukkan untuk memastikan keakuratannya. Jika terjadi kesalahan, pengguna akan diarahkan kembali ke halaman *login* hingga validasi berhasil dilakukan. Setelah berhasil *login*, sistem menampilkan halaman *dashboard*, di mana pengguna dapat memilih menu *logout*. Sistem kemudian memproses permintaan *logout* dan mengakhiri sesi pengguna dan keluar dari sistem secara aman.

### 3) Class Diagram

Dalam proses perancangan dan pembangunan aplikasi untuk pendaftaran layanan Puskesmas Pembantu Arjasari berbasis web, terdapat tiga tabel utama yang masing-masing dirancang untuk menangani berbagai aspek layanan serta memiliki atribut dan fungsi sebagai berikut.



Gambar IV. 18 Class Diagram

a. *Admin*

Entitas admin merepresentasikan administrator sistem. Admin bertanggung jawab untuk mengelola sistem, tetapi tidak memiliki relasi langsung dengan entitas lain dalam diagram ini.

b. *User*

Entitas pengguna mengacu pada informasi yang menggambarkan individu yang menggunakan aplikasi, di mana mereka memiliki kemampuan dan hak akses untuk memanfaatkan berbagai layanan yang disediakan, layanan tersebut mencakup kemampuan untuk melakukan konsultasi yang berkaitan dengan hewan.

c. *Consultation*

Entitas ini menggambarkan informasi terkait konsultasi hewan yang dilakukan oleh pengguna.

#### 4) *Data Base*

*Database* merupakan salah satu komponen utama dalam suatu sistem informasi yang memiliki peran penting dalam menyimpan, mengelola, serta mengakses data dengan cara yang lebih terstruktur dan efisien. Dengan adanya *database*, berbagai jenis informasi dapat diorganisir dengan baik, sehingga memudahkan pengguna dalam pencarian, pemrosesan, dan analisis data sesuai kebutuhan. Dalam konteks pengembangan aplikasi berbasis web untuk Puskesmas Pembantu Arjasari, *database* dirancang untuk mendukung proses pendaftaran layanan secara terstruktur dan terorganisasi.

##### a. Struktur tabel

Tabel IV. 4 Struktur Tabel *Admin*

Nama	Data Type	Keterangan
<i>id</i>	<i>Int</i>	<i>Primary Key, Auto Increment</i>
<i>email</i>	<i>Character varying (255)</i>	<i>Not Null, Unique</i>
<i>password</i>	<i>Character varying (255)</i>	<i>Not Null</i>
<i>role</i>	<i>Character varying (255)</i>	<i>Not Null, Default 'admin'</i>
<i>createdAt</i>	<i>Timestamp</i>	<i>Auto Generated</i>
<i>updatedAt</i>	<i>Timestamp</i>	<i>Auto Generated</i>

- 1) *id (int)*: *Primary key* digunakan untuk mengidentifikasi setiap pengguna secara unik dalam sistem.
- 2) *email (char)*: *Email* admin, harus unik.
- 3) *password (char)*: Kata sandi admin untuk autentikasi.
- 4) *role (char)*: Peran admin, *default*-nya adalah "*admin*".
- 5) *created (timestamp)*: Waktu ketika data admin dibuat.
- 6) *updated (timestamp)*: Waktu ketika data admin terakhir diperbarui.



Tabel IV. 5 Struktur Tabel *Consultation*

Nama	Data Type	Keterangan
id	<i>Int</i>	<i>Primary Key, Auto Increment</i>
namaHewan	<i>Character varying (255)</i>	<i>Not Null</i>
umurHewan	<i>Int</i>	<i>Not Null</i>
jenisKelamin	<i>Character varying (255)</i>	<i>Not Null</i>
JenisHewan	<i>Character varying (255)</i>	<i>Not Null</i>
Keluhan	<i>Character varying (255)</i>	<i>Not Null</i>
status	<i>Character varying (255)</i>	<i>Not Null, Default 'pending'</i>
userId	<i>Int</i>	<i>Foreign Key (User.id)</i>
queueNumber	<i>Int</i>	<i>Not Null</i>
createdAt	<i>Timestamp</i>	<i>Auto Generated</i>
updatedAt	<i>Timestamp</i>	<i>Auto Generated</i>

- 1) id (*int*): *Primary key* digunakan untuk mengidentifikasi setiap pengguna secara unik dalam sistem.
- 2) email (*char*): *Email* pengguna, email ini harus bersifat unik, artinya tidak boleh ada dua pengguna dengan alamat email yang sama dalam sistem.
- 3) password (*char*): Kata sandi yang digunakan oleh untuk proses autentikasi saat mengakses aplikasi. Data disimpan dengan keamanan tinggi untuk melindungi privasi pengguna.
- 4) nama (*char*): Nama lengkap pengguna.
- 5) NIK (*char*): Nomor Induk Kependudukan pengguna. Data ini digunakan untuk keperluan validasi dan identifikasi tambahan.
- 6) alamat (*char*): Alamat tempat tinggal pengguna.
- 7) kelurahan (*char*): Kelurahan tempat tinggal pengguna.
- 8) kecamatan (*char*): Kecamatan tempat tinggal pengguna.

- 9) nomorWA (*char*): Nomor WhatsApp pengguna untuk komunikasi.
- 10) *created* (*timestamp*): Waktu ketika data pengguna dibuat.
- 11) *updated* (*timestamp*): Waktu ketika data pengguna terakhir diperbarui.

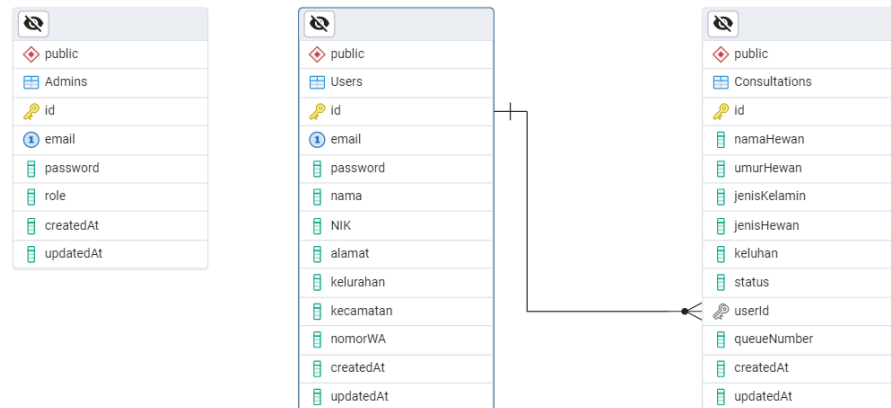
Tabel IV. 6 Struktur Tabel *Users*

Name	Data Type	Keterangan
id	<i>Int</i>	<i>Primary Key, Auto Increment</i>
<i>email</i>	<i>Character varying (255)</i>	<i>Not Null, Unique</i>
<i>password</i>	<i>Character varying (255)</i>	<i>Not Null</i>
nama	<i>Character varying (255)</i>	<i>Not Null</i>
NIK	<i>Character varying (255)</i>	<i>Not Null</i>
alamat	<i>Character varying (255)</i>	<i>Not Null</i>
kelurahan	<i>Character varying (255)</i>	<i>Not Null</i>
kecamatan	<i>Character varying (255)</i>	<i>Not Null</i>
nomorWA	<i>Character varying (255)</i>	<i>Not Null</i>
<i>createdAt</i>	<i>Timestamp</i>	<i>Auto Generated</i>
<i>updatedAt</i>	<i>Timestamp</i>	<i>Auto Generated</i>

- 1) id (*int*): *Primary key* untuk mengidentifikasi konsultasi secara unik dalam sistem.
- 2) namaHewan (*char*): Nama hewan yang dikonsultasikan.
- 3) umurHewan (*int*): Umur hewan dalam tahun.
- 4) jenisKelamin (*char*): Jenis kelamin hewan.
- 5) jenisHewan (*char*): Jenis hewan (misalnya: kucing, anjing, dll.).
- 6) keluhan (*char*): Keluhan yang dialami oleh hewan.
- 7) status (*char*): Status konsultasi, default-nya adalah "pending".
- 8) *userId* (*int*): *Foreign key* yang menghubungkan konsultasi dengan pengguna.
- 9) *queueNumber* (*int*): Nomor antrian untuk konsultasi.
- 10) *created* (*timestamp*): Waktu ketika data konsultasi dibuat.

11) updated (*timestamp*): Waktu ketika data konsultasi terakhir diperbarui.

b. Tabel Relasi



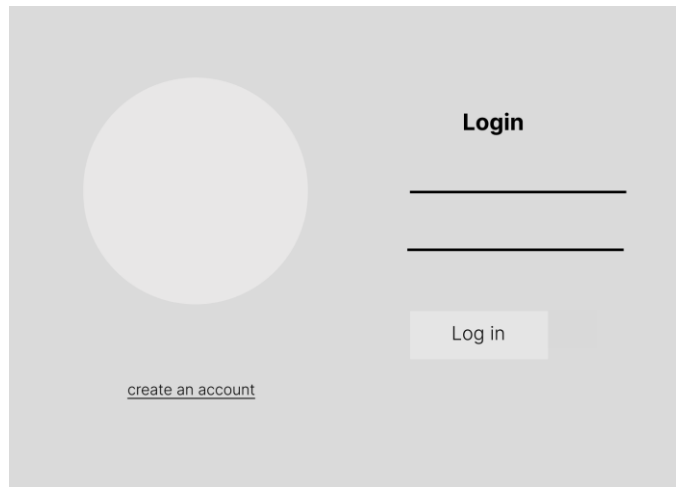
Gambar IV. 19 *Tabel Relasi*

Diagram ini menggambarkan relasi yang menunjukkan tiga tabel utama: Tabel *Admins*, tabel *Users*, dan tabel *Consultations*. Tabel *Admins* menyimpan data admin tanpa relasi langsung dengan tabel lain. Untuk tabel *Users* menyimpan data pengguna, seperti informasi pribadi dan kontak, yang memiliki relasi *one-to-many* dengan tabel *Consultations* melalui kolom `userId` sebagai *Foreign Key*. Tabel *Consultations* mencatat data konsultasi hewan, termasuk informasi hewan, keluhan, status, dan nomor antrean, yang terhubung ke pengguna tertentu. Kolom `id` pada tabel *Users* dihubungkan ke kolom `userId` pada tabel *Consultations* karena relasi *one-to-many*, di mana satu pengguna dapat memiliki banyak konsultasi. Kolom `id` sebagai *Primary Key* di tabel *Users* berfungsi sebagai identitas unik setiap pengguna, sedangkan kolom `userId` di tabel *Consultations* adalah *Foreign Key* yang merujuk pengguna terkait. Hubungan antara tabel ini memastikan setiap data konsultasi terhubung dengan pengguna yang valid, untuk menjaga integritas dan keteraturan data dalam sistem. Relasi ini memungkinkan sistem mengelola data pengguna dan konsultasi secara terstruktur.

## 5) Wireframe

### a. User

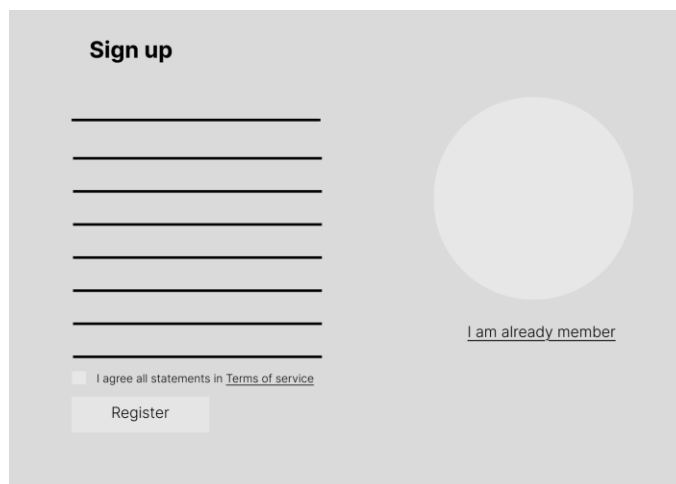
#### 1. Wireframe Halaman Login



Gambar IV. 20 Wireframe Login

Pada menu *login*, terdapat logo di sisi kiri diikuti oleh tulisan "*Login*", formulir input *username* dan *password* di kanan, serta opsi *Create an Account*, dan tombol *login* untuk masuk.

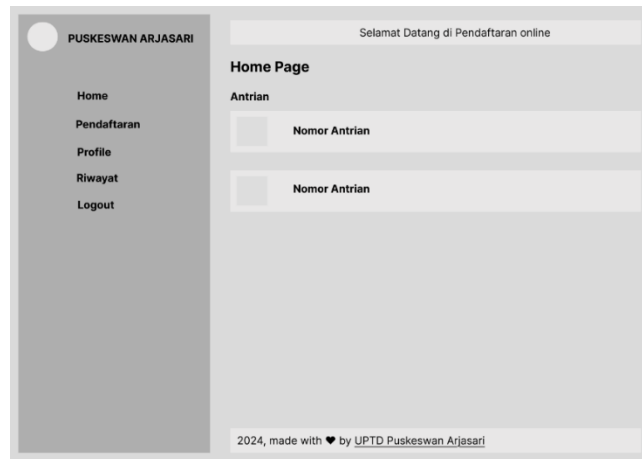
#### 2. Wireframe Halaman Buat Akun (*Create an Account*)



Gambar IV. 21 Wireframe Buat Akun

Pada *wireframe* buat akun, logo bersama teks *Sign Up*. Disertai formulir input data pengguna serta kotak persetujuan *I agree to all statements in Terms of Service* dan tombol *Register*.

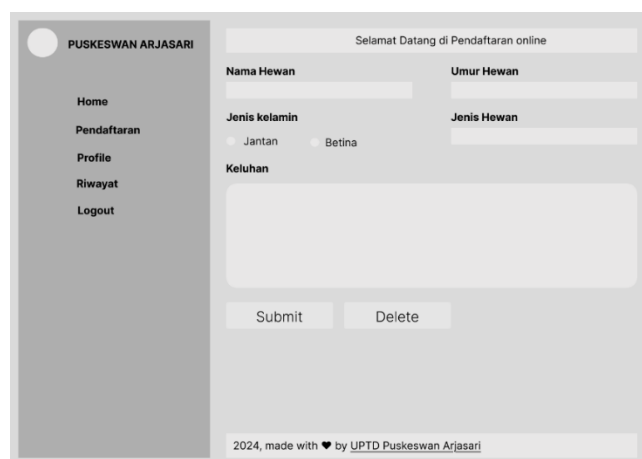
### 3. *Wireframe* Halaman *Home*



Gambar IV. 22 *Wireframe* *Home*

Pada *wireframe* halaman *Home*, ditampilkan logo dan nama instansi di bagian atas. Halaman ini juga dilengkapi dengan lima menu utama untuk navigasi. Di sisi kanan, terdapat panel daftar antrian yang menampilkan informasi secara real-time, memberikan informasi terkini mengenai status antrian.

### 4. *Wireframe* Halaman Pendaftaran



Gambar IV. 23 *Wireframe* Pendaftaran

Pada menu pendaftaran, halaman ini menampilkan logo dan nama instansi sebagai identitas utama, diikuti dengan dashboard yang terletak di sisi kiri halaman, berisi lima menu utama yang memudahkan akses ke berbagai fitur. Di sisi kanan, terdapat formulir pendaftaran hewan yang terdiri dari beberapa kolom input yang harus diisi, antara lain nama hewan, jenis kelamin, umur, jenis hewan, dan keluhan yang dialami. Formulir ini juga dilengkapi dengan dua tombol, yaitu tombol *Submit* untuk mengirimkan data yang telah diisi dan tombol *Delete* untuk menghapus formulir.

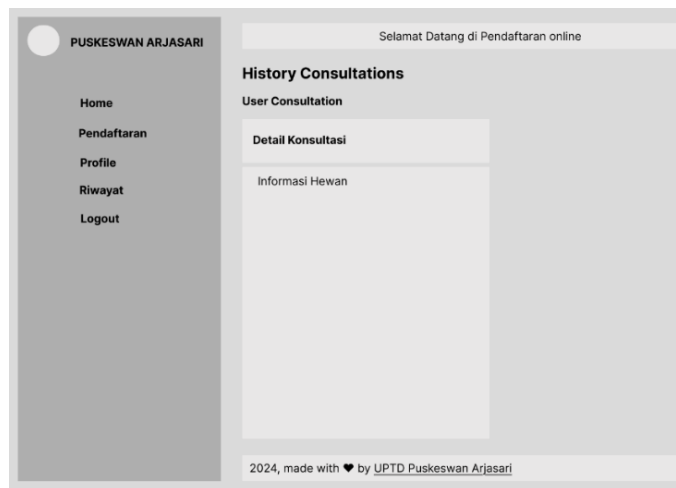
#### 5. Wireframe Halaman *Profile*

The wireframe shows a web interface for 'PUSKESWAN ARJASARI'. On the left is a sidebar with a circular logo and a list of menu items: Home, Pendaftaran, Profile, Riwayat, and Logout. The main content area on the right has a header 'Selamat Datang di Pendaftaran online' and a section titled 'Update Profile'. This section contains five input fields labeled 'Nama', 'Alamat', 'Kelurahan', 'Kecamatan', and 'Nomor WA'. Below these fields are two buttons: 'Update' and 'Delete'. At the very bottom of the page, there is a small footer text: '2024, made with ❤ by UPTD Puskesmas Arjasari'.

Gambar IV. 24 Wireframe *Profile*

Halaman ini dirancang untuk menampilkan logo dan nama instansi sebagai elemen identitas utama. Pada bagian kiri halaman, terdapat *dashboard* yang menyediakan lima menu utama yang berfungsi sebagai alat navigasi. Sementara itu, di sisi kanan halaman, tersedia informasi profil pengguna yang dapat diedit sesuai kebutuhan. Untuk mendukung pengelolaan data profil, tombol *Update* berfungsi untuk menyimpan perubahan yang dilakukan, serta tombol *Delete* yang memungkinkan pengguna menghapus formulir dengan mudah.

## 6. Wireframe Halaman Riwayat

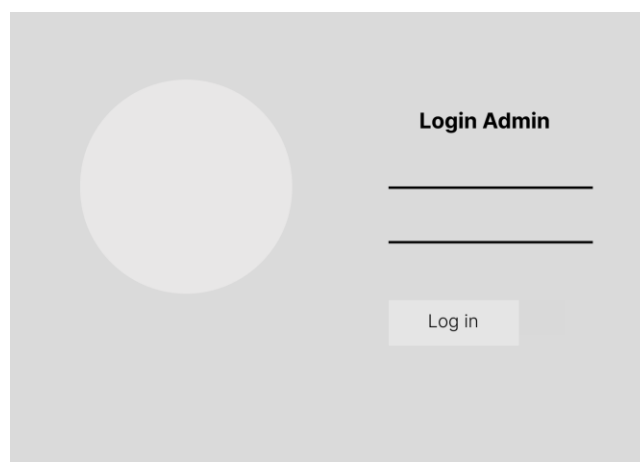


Gambar IV. 25 Wireframe Riwayat

Pada halaman riwayat, logo dan nama instansi ditempatkan di bagian atas. Halaman ini dilengkapi dengan *dashboard* di sisi kiri yang menampilkan lima menu untuk memudahkan navigasi antar fitur. Di bagian utama, tersedia tampilan data riwayat konsultasi untuk meninjau layanan yang telah dilakukan.

### b. Admin

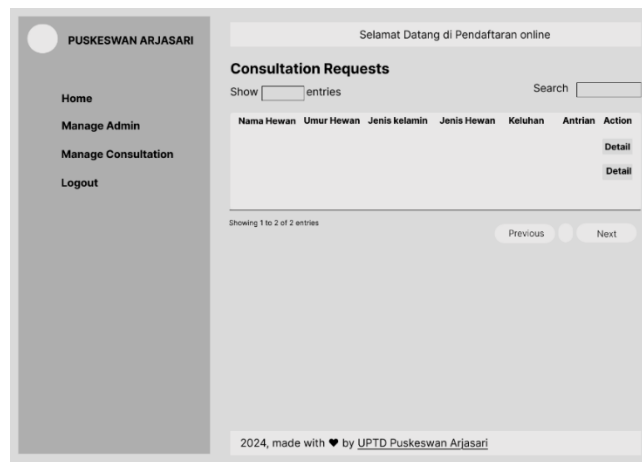
#### 1. Wireframe Login Admin



Gambar IV. 26 Wireframe Login Admin

Pada *wireframe* halaman *login admin*, halaman ini menampilkan logo, diikuti dengan *Login Admin* yang menjadi petunjuk utama di kanan halaman. Terdapat formulir input *username* dan *password*. Kemudian, admin menekan tombol *Log In* untuk mengakses halaman admin melanjutkan ke fitur berikutnya.

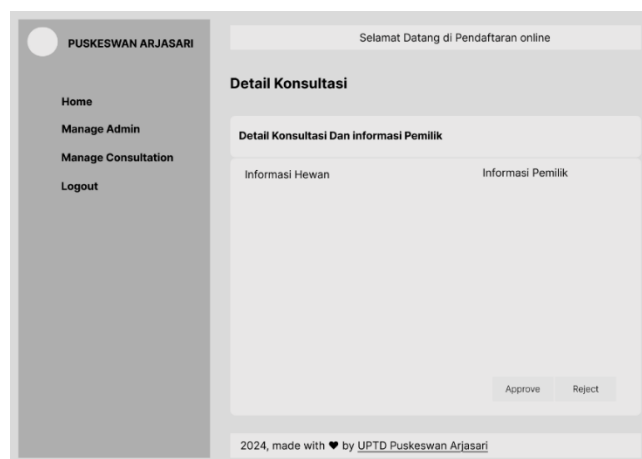
## 2. *Wireframe* Halaman *Home Admin*



Gambar IV. 27 *Wireframe* Home Admin

*Home* admin, terdapat permintaan konsultasi dengan opsi detail untuk informasi lanjutan. *Show entries* untuk melihat nomor halaman yang sedang diakses. *Search* untuk mencari data dengan kata kunci, *previous* dan *next* untuk navigasi halaman.

## 3. *Wireframe* Halaman Detail Konsultasi

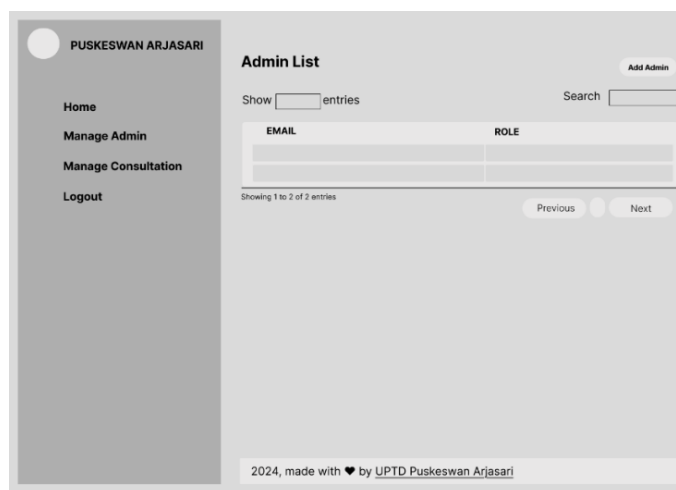


Gambar IV. 28 *Wireframe* Detail Konsultasi



Halaman ini merupakan kelanjutan dari tampilan sebelumnya, yaitu halaman detail konsultasi, yang menampilkan informasi lengkap mengenai data konsultasi terkait masalah yang dilaporkan oleh pasien. Di halaman ini, terdapat dua tombol utama, yaitu tombol *Approve* untuk menyetujui permintaan konsultasi dan tombol *Reject* untuk menolak permintaan tersebut. Desain halaman ini memudahkan admin dalam mengambil keputusan terkait permintaan konsultasi yang masuk.

#### 4. Wireframe Halaman *Manage Admin*



Gambar IV. 29 Wireframe *Manage Admin*

Pada halaman *Manage Admin*, terdapat logo dan nama instansi di sebelah kiri, dan diikuti oleh empat dashboard utama. Dimana terdapat tampilan daftar admin yang mencakup informasi *email* dan peran sebagai admin yang memiliki akses sistem disebelah kanan halaman. Halaman ini dilengkapi dengan tombol add admin untuk menambah admin dan fitur *show entries* dibuat untuk memudahkan admin dalam melihat nomor halaman yang sedang diakses, sementara menu *search* ditujukan untuk memudahkan pencarian data berdasarkan kata kunci yang dimasukan oleh admin. Selain itu, halaman ini juga dilengkapi dengan tombol *previous* dan *next* untuk navigasi antar halaman.

## 5. Wireframe Halaman Add Admin

Gambar IV. 30 Wireframe Add Admin

Pada halaman add admin, logo dan nama instansi, diikuti empat menu untuk akses fitur lainnya. Di sisi kanan, terdapat formulir untuk *email* dan *password* admin baru, dengan tombol Submit untuk menyimpan data dan *Cancel* untuk membatalkan proses.

## 6. Wireframe Halaman Manage Consultations

Gambar IV. 31 Wireframe Manage Consultation

Pada menu *Manage Consultation*, halaman menampilkan logo instansi di kiri, diikuti empat menu utama. Di kanan, terdapat data

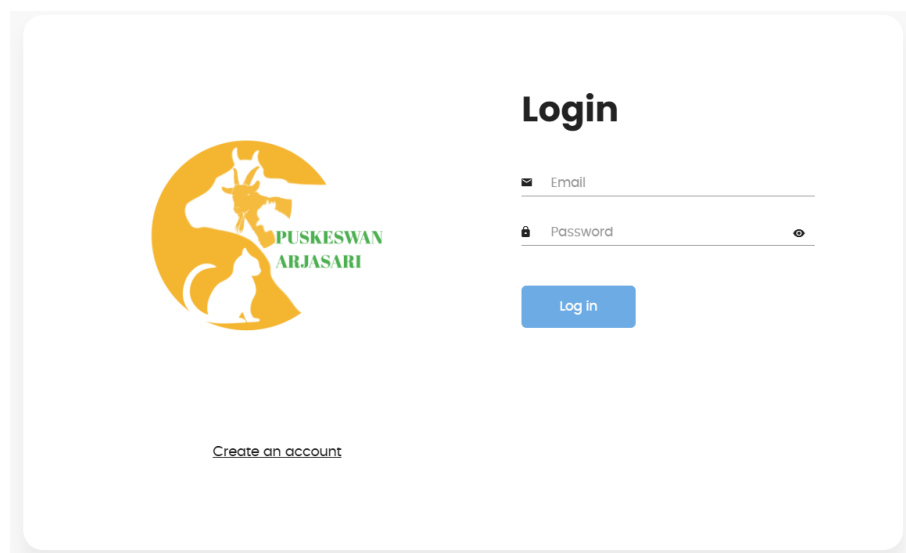
konsultasi yang telah disetujui atau ditolak, lengkap dengan informasi nama hewan, keluhan, status, dan menu detail. Fitur *show entries* untuk melihat nomor halaman, sementara menu *search* di kanan atas mempermudah pencarian data.

#### IV.2.3. Implementasi Antarmuka

Implementasi antarmuka merujuk pada penerapan desain UI ke dalam sistem yang berfungsi, memastikan elemen visual dan interaksi mendukung pengalaman pengguna yang optimal. Dalam pengembangan aplikasi pendaftaran layanan Puskesmas Pembantu (Pustu) Arjasari, terdapat berbagai implementasi antarmuka yang dirancang halaman *home*, *Profile*, dan riwayat. Sementara itu, untuk admin, terdapat antarmuka yang tersedia meliputi *login*, halaman utama (*home*), halaman detail konsultasi dari hewan yang akan dilakukan pemeriksaan, pengelolaan admin (*Manage Admin*), halaman *add admin* untuk menambah akses admin yang memiliki ijin akses, serta halaman pengelolaan konsultasi (*Manage Consultation*).

##### a. User

##### 1) Tampilan Halaman *Login*



Gambar IV. 32 Tampilan *Login User*

Pada tampilan *login*, tampilan antarmuka yang berfungsi sebagai halaman awal aplikasi yang menyediakan formulir input untuk pengguna memasukkan *username* dan *password* agar dapat

mengakses sistem dengan mudah. Halaman ini juga dilengkapi dengan menu tambahan berupa opsi buat akun baru (*Create an Account*), untuk memfasilitasi pengguna baru dalam membuat akun. Selain itu, terdapat tombol *Log In* yang digunakan untuk mengonfirmasi data yang sudah dimasukkan dan melanjutkan ke halaman utama sistem untuk mendapat pelayanan dari puskesmas.

## 2) Tampilan Halaman *Create an Account*

**Sign up**

✉ Your Email

🔒 Password

👤 Your name

🆔 NIK

📍 Alamat


📍 Kelurahan

📍 Kecamatan

☎ nomor hp

☐ I agree all statements in [Terms of service](#)

Register

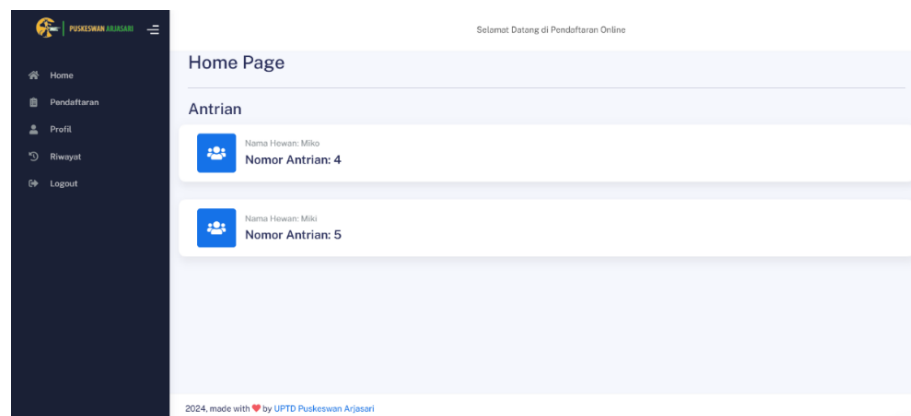


[I am already member](#)

Gambar IV. 33 Tampilan *Create an Account*

Tampilan antarmuka *Create an Account* dirancang untuk memungkinkan pengguna membuat akun baru pada sistem. Pada antarmuka ini, pengguna diminta untuk mengisi sejumlah informasi penting, termasuk email, kata sandi, nama lengkap, NIK, alamat, kelurahan, kecamatan, serta nomor *WhatsApp*. Selain itu, pengguna juga diminta untuk menyetujui pernyataan persetujuan (*I agree statement*) yang menyatakan bahwa data yang dimasukkan adalah benar dan valid, sehingga dapat digunakan sebagai akun resmi dalam aplikasi. Hal ini bertujuan untuk memastikan keabsahan data dan keamanan dalam penggunaan layanan aplikasi.

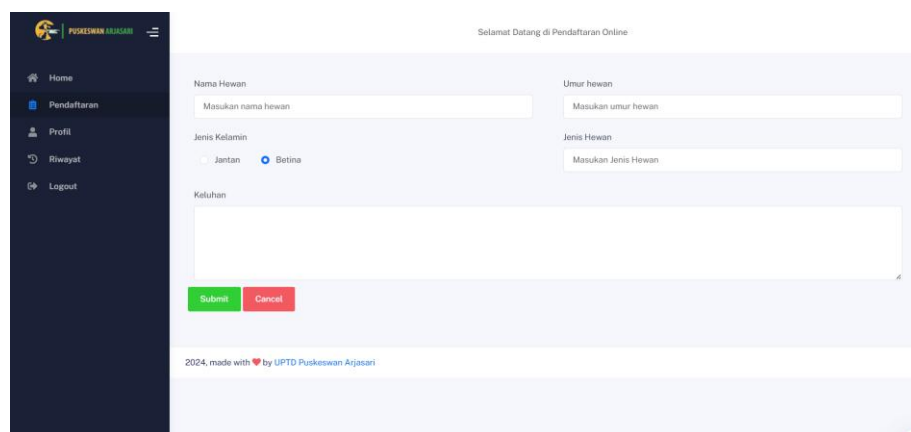
### 3) Tampilan Halaman *Home*



Gambar IV. 34 Tampilan *Home*

Pada tampilan antarmuka *home* pengguna diberikan akses untuk melihat daftar antrian yang dihasilkan setelah mereka mendaftarkan hewan sebagai pasien. Daftar antrian ini urutan antrian dan nomor antrian yang telah ditentukan untuk setiap hewan yang terdaftar. Menu ini membantu mengelola waktu dengan memberikan estimasi jadwal pemeriksaan berdasarkan nomor antrian.

### 4) Tampilan Halaman Pendaftaran



Gambar IV. 35 Tampilan *Pendaftaran*

Tampilan halaman pendaftaran dirancang untuk menampilkan formulir yang memungkinkan pengguna mengisi berbagai informasi

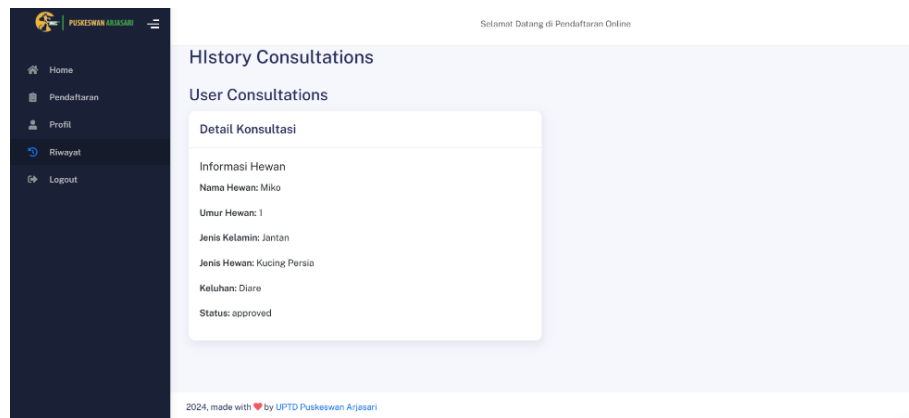
yang diperlukan untuk kelengkapan data konsultasi yang dilakukan untuk pemeriksaan. Formulir ini mencakup beberapa kolom input, seperti nama pengguna, jenis kelamin, dan usia, yang bertujuan untuk mengidentifikasi data pribadi pengguna. Selain itu, terdapat kolom tambahan untuk mencatat jenis hewan yang dimiliki beserta keluhan atau masalah yang dialami oleh hewan tersebut. Dengan pengisian data yang lengkap, halaman ini membantu memastikan bahwa informasi yang dibutuhkan dapat dikumpulkan secara akurat untuk keperluan layanan lebih lanjut.

#### 5) Tampilan Halaman *Profile*

Gambar IV. 36 Tampilan *Profile*

Dalam tampilan halaman profil ini ditampilkan data pribadi dari pada pemilik hewan dan disediakan fitur yang memungkinkan pengguna untuk mengubah dan memperbarui informasi pribadi mereka dengan mudah. Informasi yang dapat diperbarui meliputi nama lengkap, alamat tempat tinggal, serta nomor *WhatsApp*. Setelah pengguna melakukan perubahan pada data tersebut, sistem akan secara otomatis menyimpan dan memperbarui informasi yang baru dimasukkan. Fitur ini dirancang untuk memastikan bahwa data pengguna selalu terkini dan relevan, sehingga mendukung kelancaran layanan yang diberikan.

## 6) Tampilan Halaman Riwayat

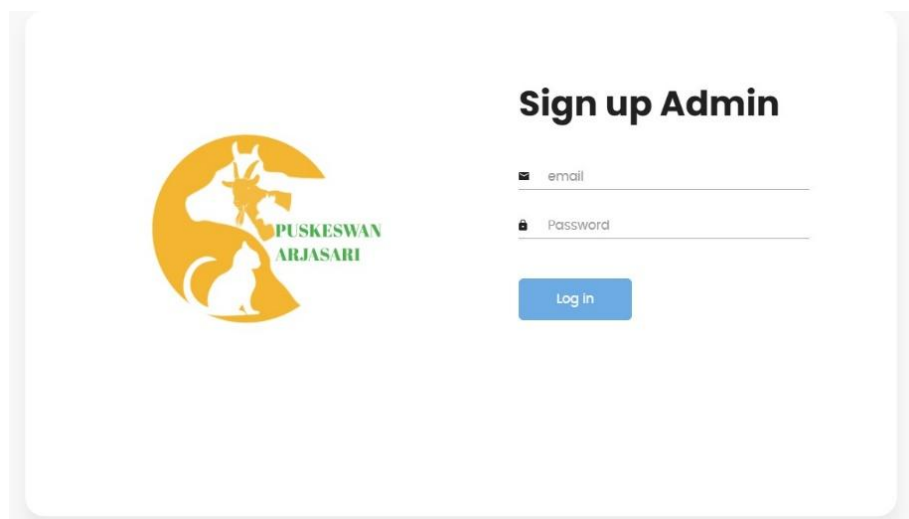


Gambar IV. 37 Tampilan Riwayat

Menu riwayat adalah salah satu menu dalam aplikasi yang berfungsi untuk menampilkan sejarah atau riwayat konsultasi yang telah dilakukan oleh pengguna. Melalui menu ini, pengguna dapat melihat daftar lengkap konsultasi sebelumnya, termasuk detail informasi mengenai tanggal, jenis layanan, dan hasil pemeriksaan atau penanganan yang diterima.

### b. Admin

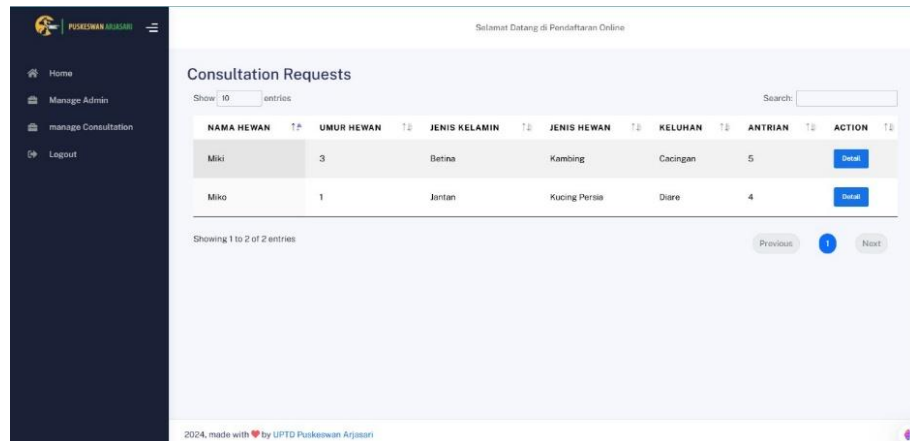
#### 1) Tampilan Login Admin



Gambar IV. 38 Tampilan Login Admin

Tampilan antarmuka *login* admin pertama yang muncul saat aplikasi dibuka, admin diminta memasukkan *username* dan *password*. Sistem memverifikasi data, jika informasi tersebut valid, admin diarahkan ke halaman *dashboard* melanjutkan ke fitur-fitur lainnya.

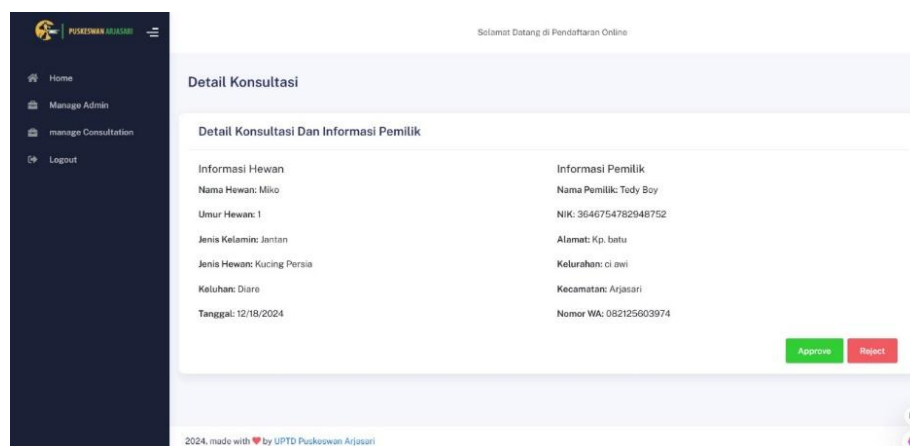
## 2) Tampilan *Home Admin*



Gambar IV. 39 Tampilan *Home Admin*

Pada halaman ini, admin dapat mengakses dan melihat semua permintaan konsultasi yang diajukan oleh pasien. Selain itu, admin memiliki hak untuk memeriksa detail konsultasi dari setiap permintaan yang masuk.

## 3) Tampilan Detail Konsultasi

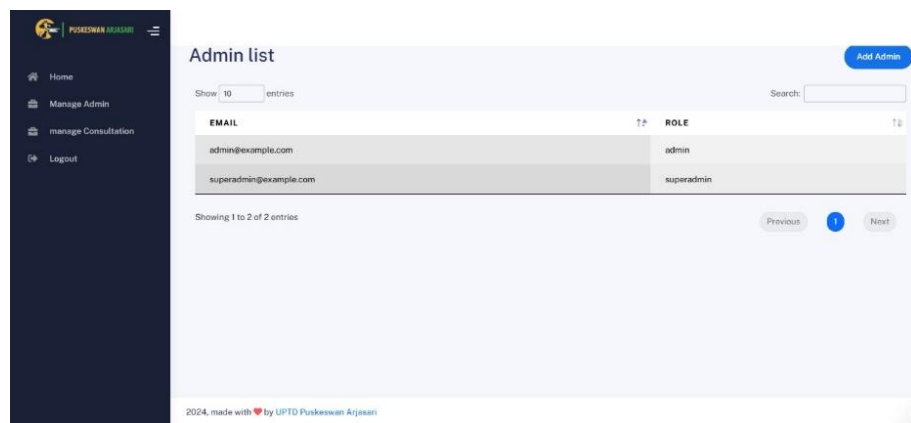


Gambar IV. 40 Tampilan Detail Konsultasi



Tampilan detail konsultasi ini lanjutan dari tampilan sebelumnya, yang menyajikan informasi lengkap mengenai konsultasi yang diajukan oleh pasien. Admin diberi opsi untuk menyetujui (*approve*) atau menolak (*reject*) penerimaan pendaftaran..

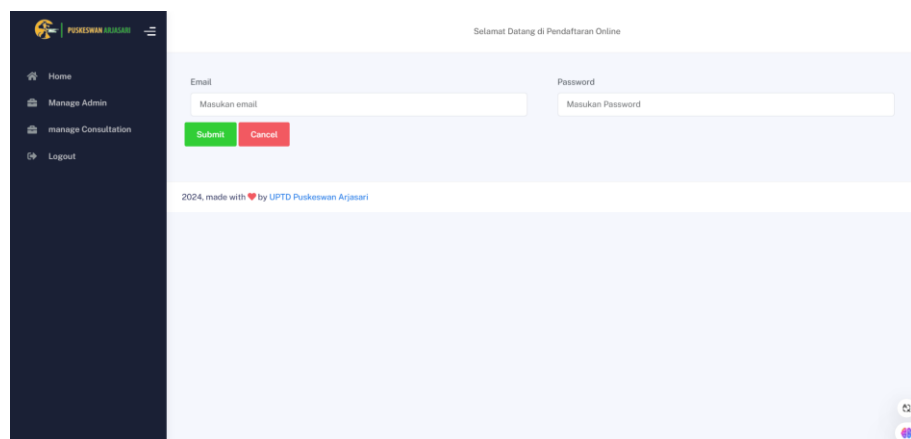
#### 4) Tampilan *Manage Admin*



Gambar IV. 41 Tampilan *Manage Admin*

Pada halaman *Manage Admin*, ditampilkan daftar lengkap admin yang memiliki akses ke sistem. Halaman ini memberikan informasi yang jelas mengenai siapa saja yang terdaftar sebagai admin.

#### 5) Tampilan *Add Admin*



Gambar IV. 42 Tampilan *Add Admin*

Antarmuka ini lanjutan dari halaman sebelumnya, untuk menampilkan formulir penambahan data admin baru. Admin diminta untuk mengisi informasi berupa *email* dan *password* sebagai syarat untuk memberikan akses ke sistem bagi admin baru.

#### 6) *Manage Consultation*



Approved or Rejected Consultations

Show 10 entries

Search:

NAMA HEWAN	KELUHAN	STATUS	DETAIL
bb	butuk	approved	<a href="#">Detail</a>
caca	sakit perut	approved	<a href="#">Detail</a>
Comot	Flu	approved	<a href="#">Detail</a>
Comot	sakit	approved	<a href="#">Detail</a>
Comot	Bulu Rontok	approved	<a href="#">Detail</a>
Miko	Situmling	approved	<a href="#">Detail</a>
Miko	Diare	approved	<a href="#">Detail</a>

2024, made with ❤️ by UPTD Puskesmas Arjasari

Gambar IV. 43 Tampilan *Manage Consultation*

Pada halaman *Manage Consultation*, ditampilkan data *Approved or Rejected Consultation* yang telah diproses oleh semua pasien, mencakup informasi tentang konsultasi yang disetujui maupun yang ditolak, lengkap dengan statusnya.

### IV.2.4. Pengujian

Pengujian pada Sistem Informasi Pendaftaran Puskesmas Pembantu Arjasari akan dilakukan dengan dua cara, yaitu pengujian menggunakan metode *Black Box* dan pengujian *Reliabilitas* melalui *Google Form* yang mencakup enam pertanyaan terkait fungsionalitas website sistem informasi pendaftaran.

#### 1. Pengujian *Black Box*

Pengujian Black Box pada Sistem Informasi Pendaftaran Puskesmas Pembantu Pustu Arjasari dilakukan untuk memastikan bahwa setiap fitur dan fungsionalitas sistem bekerja sesuai dengan spesifikasi yang telah ditentukan, tanpa memeriksa struktur internal atau kode program.

Pengujian ini berfokus pada aspek eksternal dari sistem, yaitu bagaimana sistem merespons input dari pengguna dan apakah hasil yang diharapkan sesuai dengan tujuan yang telah ditetapkan.

*Tabel IV. 7 Pengujian Black Box User*

No	Fitur Uji	Test Case	Langkah Pengujian	Hasil yang diharapkan	Ke-simpulan
1.	<i>Login</i>	Memasukkan <i>username</i> dan <i>password</i> yang valid	1.Masukkan <i>username</i> dan <i>password</i> valid 2. Klik tombol " <i>Login</i> ".	Sistem mengarahkan pengguna ke halaman <i>dashboard</i> .	<i>Valid</i>
2.	<i>Login</i>	Memasukkan <i>username</i> atau <i>password</i> yang tidak valid	1.Masukkan <i>username</i> atau <i>password</i> yang salah. 2. Klik tombol " <i>Login</i> ".	Sistem menampilkan pesan kesalahan " <i>Invalid credentials</i> ".	<i>Valid</i>
3.	Pendaftaran Pasien Hewan	Mengisi semua data pendaftaran dengan benar	Isi semua data hewan (nama hewan, jenis hewan, umur hewan, jenis kelamin dan keluhan). 2. Klik tombol " <i>Submit</i> ".	Data tersimpan dalam sistem, dan pengguna menerima notifikasi pendaftaran berhasil.	<i>Valid</i>
4.	Menu Riwayat	Pengguna melihat riwayat konsultasi.	Pilih menu riwayat	Menampilkan data history konsultasi	<i>Valid</i>
5.	Menu <i>Profile</i>	Pengguna melihat informasi profil	1. <i>Login</i> sebagai pengguna. 2. Klik menu " <i>Profile</i> ".	Sistem menampilkan data profil pengguna dengan informasi yang benar.	<i>Valid</i>

6.	Menu <i>Profile</i>	Update <i>Profile</i>	1. Tekan tombol update <i>Profile</i> 2. Masukkan data <i>Profile</i> terbaru	Menampilkan menu <i>Profile</i> dengan data yang baru dengan notifikasi "Update <i>Profile</i> Berhasil"	<i>Valid</i>
7	<i>Logout</i>	Pengguna menekan tombol logout	Klik tombol "Logout".	Sistem mengarahkan kembali ke <i>login</i> , dan sesi pengguna diakhiri.	<i>Valid</i>

Tabel IV. 8 Pengujian Black Box Admin

No	Fitur Uji	Test Case	Langkah Pengujian	Hasil yang diharapkan	Ke-simpulan
1.	<i>Login Admin</i>	Admin <i>login</i> dengan kredensial valid	Masukkan <i>username</i> dan <i>password</i> admin yang valid.	Sistem mengarahkan admin ke halaman utama <i>dashboard admin</i> .	<i>Valid</i>
2.	<i>Login Admin</i>	Admin <i>login</i> dengan kredensial tidak valid	1. Masukkan <i>username</i> atau <i>password</i> salah. 2. Klik Tombol "Login"	Sistem menampilkan pesan kesalahan "Invalid credentials"	<i>Valid</i>
3.	<i>Home (Admin)</i>	Admin dapat melihat data pasien yang terdaftar	1. <i>Login</i> sebagai admin. 2. Akses menu "Home". 3. Periksa daftar pasien yang terdaftar.	Sistem menampilkan daftar pasien yang terdaftar dengan informasi yang lengkap.	Valid

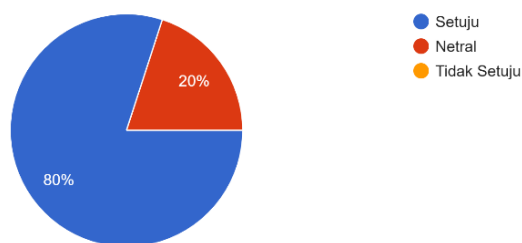
4.	<i>Manage Admin</i>	Admin dapat menambah admin baru	1. <i>Login</i> sebagai admin. 2. Akses menu " <i>Manage Admin</i> ". 3. Klik tombol " <i>add Admin</i> ". 4. Isi data admin baru.	Admin baru berhasil ditambahkan dan muncul dalam daftar admin.	<i>Valid</i>
5.	<i>Manage Consultation</i>	Admin dapat melihat riwayat konsultasi seluruh pasien	1. <i>Login</i> sebagai admin. 2. Akses menu " <i>Manage Consultation</i> ". 3. Pilih pasien untuk melihat riwayat konsultasi.	Riwayat konsultasi pasien tampil dengan data yang lengkap dan terstruktur.	<i>Valid</i>
6	<i>Manage Consultation</i>	Admin dapat mencari riwayat konsultasi pasien berdasarkan kata kunci	1. <i>Login</i> sebagai admin. 2. Akses menu " <i>Manage Consultation</i> ". 3. Masukkan kata kunci di kolom <i>search</i> .	Sistem menampilkan riwayat konsultasi sesuai dengan kata kunci yang dicari.	<i>Valid</i>
7	Akses Tidak Sah	Pengguna mencoba mengakses halaman admin tanpa <i>login</i>	1. Akses URL halaman admin tanpa <i>login</i> . 2. Tekan Enter.	Sistem mengarahkan pengguna ke halaman <i>login</i> dengan pesan " <i>Error</i> ".	<i>Valid</i>

## 2. Pengujian *Reliabilitas Google Form*

Pengujian reliabilitas sistem dilakukan untuk mengevaluasi sejauh mana sistem informasi yang telah dikembangkan dapat memenuhi kebutuhan pengguna, khususnya admin dan *user*, dalam mengakses fitur-fitur yang tersedia. Pengujian ini dilakukan menggunakan *Google Form* dengan serangkaian pertanyaan yang mencakup aspek fungsionalitas, kemudahan penggunaan, dan stabilitas sistem. Tujuan dari pengujian ini adalah untuk mengumpulkan umpan balik langsung dari pengguna sebagai dasar untuk menilai keandalan sistem serta mengidentifikasi area yang memerlukan perbaikan atau pengembangan lebih lanjut. Hasil pengujian ini akan menjadi acuan dalam memastikan bahwa sistem mampu memberikan pengalaman pengguna yang optimal.

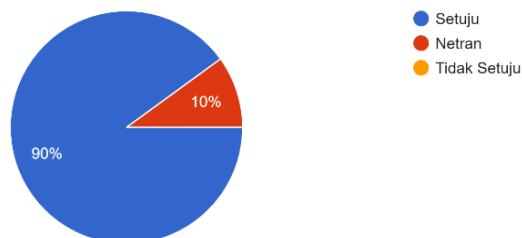
### a. Uji Fungsionalitas

1. Apakah sistem login berfungsi dengan baik?  
10 jawaban



Gambar IV. 44 *Grafik Uji Fungsionalitas 1*

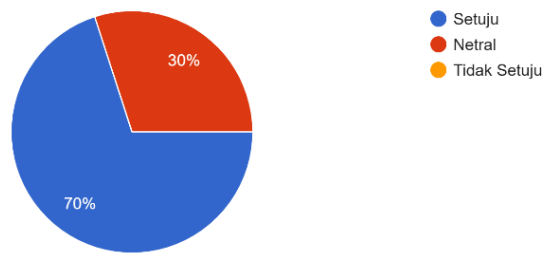
2. Apakah menu pendaftaran online mudah digunakan dan data tersimpan dengan benar?  
10 jawaban



Gambar IV. 45 *Grafik Uji Fungsionalitas 2*

3. Apakah informasi antrian pada menu "Home" ditampilkan secara jelas dan akurat?

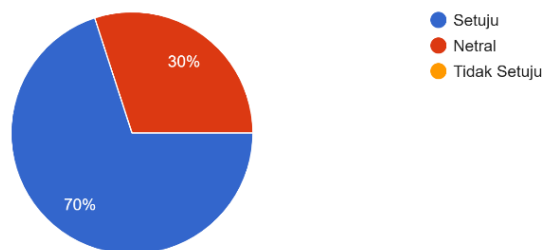
10 jawaban



Gambar IV. 46 Grafik Uji Fungsionalitas 3

4. Apakah pengelolaan data pengguna (menu profile) memudahkan Anda dalam memperbarui informasi?

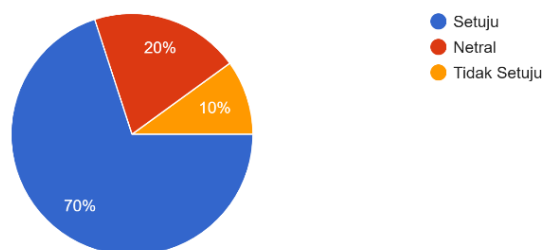
10 jawaban



Gambar IV. 47 Grafik Uji Fungsionalitas 4

5. Apakah riwayat konsultasi hewan dapat diakses dan dikelola dengan mudah?

10 jawaban

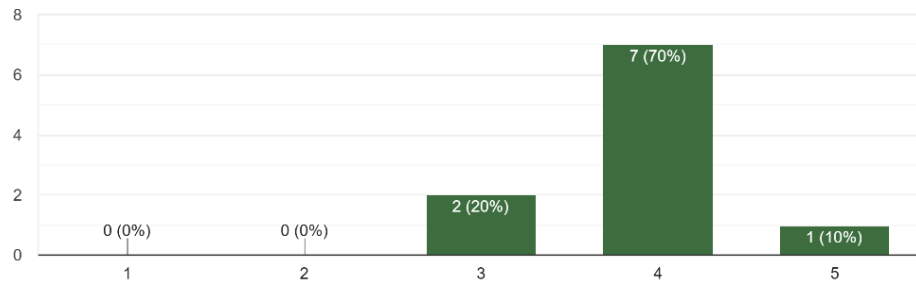


Gambar IV. 48 Grafik Uji Fungsionalitas 5

### b. Uji Responsif

6. Seberapa responsif sistem ini berjalan tanpa error atau crash?

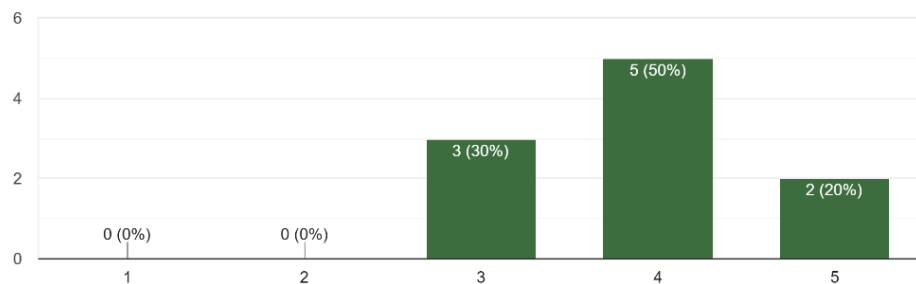
10 jawaban



Gambar IV. 49 Grafik Ui Responsif 1

7. Seberapa konsisten sistem tampil di perangkat Anda ?

10 jawaban



Gambar IV. 50 Grafik Uji Responsif 2

### IV.2.5. Pelaporan Hasil Kerja Praktik

Proses pelaporan hasil kerja praktik dilakukan pada tahap akhir kerja praktik di Puskesmas Pembantu Pustu Arjasari. Pelaporan hasil kerja praktik ini dilakukan dengan memperlihatkan hasil dari kerja praktik, yaitu berupa perencanaan pembuatan aplikasi pendaftaran untuk Puskesmas Pembantu Pustu Arjasari berbasis web. Pelaporan hasil kerja praktik juga dilakukan dengan pembuatan laporan kerja praktik yang mencakup seluruh tahapan dan hasil yang telah dicapai selama pelaksanaan kerja praktik.



### **IV.3 Pencapaian Hasil**

Adapun hasil yang dicapai dari pelaksanaan kerja praktik di Puskesmas Pembantu Pustu Arjasari mencakup pengembangan rancangan aplikasi pendaftaran berbasis web yang dirancang untuk mendukung berbagai kebutuhan operasional. Aplikasi ini dirancang untuk mengelola pendaftaran pasien, termasuk penambahan, penghapusan, dan pembaruan data pasien. Selain itu, aplikasi ini juga berfungsi untuk mengatur jadwal pelayanan kesehatan, antrian pasien, serta menghasilkan laporan terkait pendaftaran dan pelayanan kesehatan. Dalam proses pengembangan, kerja praktik ini juga menghasilkan beberapa output penting, seperti *use case diagram*, *activity diagram*, *class diagram*, *wireframe* aplikasi, dan desain antarmuka aplikasi. Hasil-hasil ini diharapkan dapat memberikan kontribusi signifikan terhadap peningkatan efisiensi dan efektivitas pengelolaan layanan di Puskesmas Pembantu Pustu Arjasari.

## **BAB V**

### **PENUTUP**

#### **V.1 Kesimpulan Dan Saran Mengenai Pelaksanaan**

Pelaksanaan kerja praktik yang telah dilakukan menunjukkan bahwa proses pembelajaran selama di bangku perkuliahan memberikan kontribusi yang sangat signifikan dalam memahami berbagai konsep yang diterapkan selama masa kerja praktik. Mata kuliah seperti pemrograman, basis data, dan rekayasa perangkat lunak menjadi fondasi yang kokoh dalam menyelesaikan berbagai tugas yang dihadapi selama kerja praktik. Pemahaman yang diperoleh dari mata kuliah tersebut memungkinkan mahasiswa untuk lebih siap dalam menghadapi tantangan yang muncul selama menjalankan proyek kerja praktik.

Selain itu, proses pelaksanaan kerja praktik memberikan pengalaman yang sangat berharga dalam mempersiapkan diri untuk memasuki dunia kerja. Dalam proses ini, mahasiswa belajar bagaimana mempersiapkan dokumen lamaran dengan baik, menghadapi wawancara secara profesional, dan berkomunikasi dengan pihak perusahaan secara efektif. Pengalaman ini tidak hanya memberikan wawasan praktis, tetapi juga melatih keterampilan *soft skill* seperti negosiasi, komunikasi interpersonal, dan pengelolaan waktu yang sangat penting dalam dunia kerja.

Lingkungan kerja yang kondusif di Puskesmas Pembantu Arjasari, ditambah dengan dukungan pembimbing lapangan yang memberikan arahan, menciptakan suasana yang ideal untuk pelaksanaan kerja praktik. Pengalaman ini tidak hanya membantu mahasiswa memahami proses kerja secara langsung juga memberikan wawasan baru tentang bagaimana teori yang dipelajari dapat diterapkan dalam dunia nyata.

##### **V.1.1. Kesimpulan Pelaksanaan Kerja Praktik**

Kesimpulan dari proses pelaksanaan kerja praktik (KP) ini mencakup seluruh tahapan yang telah dilalui, mulai dari persiapan hingga penyelesaian kegiatan. Setiap tahapan memberikan pengalaman dan

pembelajaran yang berbeda, yang secara keseluruhan memberikan kontribusi positif terhadap pengembangan kompetensi mahasiswa.

1. Tahap pertama adalah persiapan kerja praktik, yang melibatkan pencarian informasi mengenai tempat kerja praktik, pembuatan dokumen lamaran, dan persiapan wawancara. Pada tahap ini, mahasiswa belajar untuk bersikap profesional dalam mempersiapkan diri menghadapi dunia kerja. Proses ini juga memberikan wawasan tentang komunikasi yang baik dengan pihak perusahaan.
2. Tahap kedua adalah pelaksanaan kerja praktik, di mana mahasiswa terlibat langsung dalam aktivitas yang relevan dengan bidang studi mereka. Dalam hal ini, mahasiswa berpartisipasi dalam pengembangan aplikasi berbasis web untuk Puskesmas Pembantu Arjasari. Kegiatan ini meliputi analisis kebutuhan sistem, desain antarmuka, implementasi fitur, hingga pengujian aplikasi. Setiap tahap memberikan pemahaman yang lebih mendalam tentang pengembangan perangkat lunak.
3. Tahap ketiga adalah evaluasi hasil kerja praktik, di mana mahasiswa merefleksikan hasil yang telah dicapai dan mendapatkan umpan balik dari pembimbing lapangan. Evaluasi ini membantu mahasiswa untuk memahami kelebihan dan kekurangan dari pekerjaan yang telah dilakukan, sekaligus memberikan arahan untuk perbaikan di masa depan.
4. Tahap terakhir adalah penyusunan laporan kerja praktik, yang menjadi bagian penting dalam mendokumentasikan seluruh proses dan hasil kerja praktik. Penyusunan laporan ini melatih kemampuan mahasiswa dalam menulis secara sistematis dan analitis, serta memberikan kesempatan untuk merenungkan pembelajaran yang telah diperoleh selama kerja praktik.

Secara keseluruhan, proses pelaksanaan kerja praktik ini memberikan pengalaman yang sangat berharga, baik dari segi teknis maupun non-teknis. Mahasiswa tidak hanya mendapatkan pemahaman yang lebih baik tentang dunia kerja, tetapi juga mengembangkan keterampilan

interpersonal dan profesional yang penting untuk masa depan karier.

### **V.1.2. Saran Pelaksanaan Kerja Praktik**

Berdasarkan pengalaman selama pelaksanaan kerja praktik, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Mahasiswa diharapkan mempersiapkan diri dengan baik sebelum melaksanakan kerja praktik, terutama dalam menguasai dasar-dasar teori yang relevan dengan bidang yang akan digeluti. Selain itu, penting untuk proaktif dalam mencari peluang kerja praktik dan tidak ragu untuk bertanya selama pelaksanaan kerja praktik.
2. Perguruan tinggi dapat meningkatkan dukungan kepada mahasiswa dengan menyediakan lebih banyak informasi mengenai tempat kerja praktik yang relevan dan memberikan pembekalan khusus sebelum. Selain itu, penyesuaian kurikulum dengan kebutuhan industri dapat membantu mahasiswa lebih siap menghadapi tantangan di dunia kerja.
3. Lingkungan kerja di Puskesmas Arjasari sudah sangat mendukung, namun dapat ditingkatkan lebih lanjut dengan memberikan dokumentasi sistem yang lebih lengkap sehingga mempermudah mahasiswa memahami proses bisnis yang ada. Selain itu, kolaborasi dengan perguruan tinggi dapat terus diperkuat untuk menciptakan pengalaman kerja praktik yang lebih optimal.

## **V.2 Kesimpulan Dan Saran Mengenai Substansi**

Bagian ini menyajikan rangkuman utama dari pembahasan sebelumnya, yang mencakup poin-poin penting terkait substansi yang telah diuraikan. Selain itu, disampaikan pula saran-saran konstruktif yang bertujuan untuk memberikan rekomendasi atau langkah perbaikan yang dapat diambil guna meningkatkan kualitas atau efektivitas substansi yang dibahas.

### **V.2.1. Kesimpulan Mengenai Substansi**

Dalam pelaksanaan kerja praktik, substansi yang dikerjakan yaitu pengembangan aplikasi pendaftaran berbasis web untuk Puskesmas Pembantu Arjasari merupakan implementasi nyata dari berbagai teori

yang telah dipelajari selama perkuliahan. Proyek ini mencakup proses analisis kebutuhan, desain sistem, implementasi, dan pengujian, yang semuanya memberikan pengalaman praktis dalam pengembangan perangkat lunak.

Aplikasi yang dirancang berhasil memberikan solusi terhadap permasalahan administrasi yang sebelumnya dilakukan secara manual. Dengan adanya aplikasi ini, proses pendaftaran, pengelolaan data, dan pelayanan kepada masyarakat menjadi lebih efisien dan terstruktur. Penggunaan teknologi modern, seperti basis data *postgreSQL* dan framework pengembangan web, juga menunjukkan kemampuan mahasiswa dalam memanfaatkan teknologi yang relevan untuk menyelesaikan masalah nyata.

#### **V.2.2. Saran Mengenai Substansi**

1. Aplikasi ini memiliki potensi untuk terus disempurnakan dengan menambahkan fitur-fitur yang lebih kompleks, seperti integrasi dengan sistem pembayaran daring, pemberitahuan otomatis untuk mengingatkan jadwal konsultasi, serta laporan statistik yang lebih rinci guna mendukung proses pengambilan keputusan di Puskesmas.
2. Meskipun aplikasi telah memenuhi kebutuhan dasar, desain antarmuka pengguna (UI) dapat ditingkatkan agar lebih menarik secara visual dan lebih mudah digunakan. Penerapan prinsip-prinsip desain modern akan memberikan pengalaman pengguna (UX) yang lebih baik dan intuitif.
3. Penerapan keamanan data dalam aplikasi berbasis web, menjadi aspek yang sangat penting. Oleh karena itu, perlu diterapkan keamanan tambahan, seperti enkripsi data, otentikasi dua faktor, dan proteksi terhadap serangan siber, untuk melindungi informasi pasien dan admin.
4. Sebelum aplikasi diimplementasikan sepenuhnya, perlu dilakukan pengujian yang lebih luas dengan melibatkan lebih banyak pengguna. Hal ini bertujuan untuk memperoleh umpan balik yang

beragam sehingga dapat membantu mengidentifikasi dan memperbaiki bug atau kekurangan yang belum terdeteksi selama proses pengembangan.

5. Dokumentasi sistem yang lebih lengkap, termasuk panduan penggunaan dan manual teknis, akan sangat membantu dalam proses pemeliharaan dan pengembangan lebih lanjut oleh pihak lain di masa mendatang.

## DAFTAR PUSTAKA

- Abdul Wahid. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*.  
<https://www.researchgate.net/publication/346397070>
- Al, U., Mandar, A., Fauziyah, S., & Sugiarti, Y. (2022). Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web. *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer*, 8(2).  
<http://ejournal.fikom-unasman.ac.id>
- Al-Faruq, M. N. M., Nur'aini, S., & Aufan, M. H. (2022). PERANCANGAN UI/UX SEMARANG VIRTUAL TOURISM DENGAN FIGMA. *Walisongo Journal of Information Technology*, 4(1), 43–52.  
<https://doi.org/10.21580/wjit.2022.4.1.12079>
- Andrew S. Tanenbaum, & David J. Wetherall. (2011). *Andrew S. Tanenbaum - Computer Networks*.
- Aulia, B. W., Rizki, M., Prindiyana, P., & Surgana, S. (2023). Peran Krusial Jaringan Komputer dan Basis Data dalam Era Digital. *JUSTINFO | Jurnal Sistem Informasi Dan Teknologi Informasi*, 1(1), 9–20.  
<https://doi.org/10.33197/justinfo.vol1.iss1.2023.1253>
- Farianto, A., Novianto, Y., Teknik Informatika STIKOM Dinamika Bangsa Jambi Jl Jend Sudirman, M., & -Jambi, T. (2014). PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB PADA SMA N 14 MUARO JAMBI. In *Jurnal Ilmiah Media SISFO* (Vol. 8, Issue 2).
- Hartati, S. (2020). PERANCANGAN SISTEM INFORMASI INVENTARIS BARANG PADA KANTOR NOTARIS DAN PPAT R.A LIA KHOLILA, S.H MENGGUNAKAN VISUAL STUDIO CODE. *Jurnal Siskomti*, 3(2).  
<http://www.ejournal.lembahdempo.ac.id>
- Hendrawan<sup>1</sup>, E., Meisel<sup>2</sup>, M., & Sari<sup>3</sup>, D. N. (2022). ANALYSIS AND IMPLEMENTATION OF COMPUTER NETWORK SYSTEMS USING SOFTWARE DRAW.IO. In *Asia Information System Journal* (Vol. 2, Issue 1).

<http://ejournal.radenintan.ac.id/index.php/AISJ/index://creativecommons.org/licenses/by-sa/4.0/>

Hidayat, A., Yani, A., Studi Sistem Informasi, P., & Mahakarya, S. (2019). *MEMBANGUN WEBSITE SMA PGRI GUNUNG RAYA RANAU MENGGUNAKAN PHP DAN MYSQL* (Vol. 2, Issue 2).

//-2. (n.d.).

Ing. Michal Valenta, Ph. D. (2008). *slony*.

Jamilah, Y. S., & Padmasari, A. C. (n.d.). *PERANCANGAN USER INTERFACE DAN USER EXPERIENCE APLIKASI SAY.CO*.  
<https://ojs.unm.ac.id/tanra/>

Jan, I., Silaen, J., Oktavia, J. E., Sari, R., & Steven, J. (2022). *Literature Review Faktor-Faktor Yang Mempengaruhi Implementasi Si: Hardware, Software, Dan Database*. 1(1).  
<https://doi.org/10.38035/jim.v1i1>

Kurose, J. F. ., & Ross, K. W. . (2017). *Computer networking : a top-down approach*. Pearson.

Marwanto Rahmatuloh, & Muhammad Rizky Revanda. (2022). *1944-Article Text-3848-1-10-20220208*.

Noviantoro, A. S. A. F. R. (2022). 88-103+Agung+Noviantoro,+Amelia+Belinda+Silviana,+Risma+Rahmalia+Fitriani,+Hanum+Putri+Permatasari.

Nuraeni, N. (2018). PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE PADA MADRASAH TSANAWIYAH YAYASAN FISABILILLILAH BEKASI. *JURNAL SWABUMI*, 6(2).

Ode Bakrim, L., & Bina Bangsa Kendari, S. (2019). Koneksi Jaringan Internet Menggunakan Mode Ad-Hoc 802.11 Pada Tumaka Kendari. *Jurnal Sistem Informasi Dan Sistem Komputer*, 4(2). <http://ejournal.stmikbinsa.ac.id/index.php/simkom>

Pahlevi, O., Mulyani, A., & Khoir, M. (2018). SISTEM INFORMASI INVENTORI BARANG MENGGUNAKAN METODE OBJECT ORIENTED DI PT. LIVAZA TEKNOLOGI INDONESIA JAKARTA. *Jurnal PROSISKO*, 5(1). <https://livaza.com/>.



- Rahmawati, E., Nusa Mandiri Jakarta Jalan Damai No, S., Jati Barat, W., Selatan, J., Damai No, J., & Selatan Zalnika, J. (2018). Perancangan Sistem Informasi Akademik Berbasis Web (Studi Kasus SMK Insan Madani). *Jurnal Teknik Komputer*, 4(1).
- Rasool, A., & Jalil, Z. (2020). A Review of Web Browser Forensic Analysis Tools and Techniques. *Researchpedia Journal of Computing*, 1(1), 15–21. <https://doi.org/10.1111/RpJC.2020.DOI>
- Sholikhan, M., Kom, S., & Kom, M. (2022). *CSS Javascript dan HTML*. Designing a Smart Gateway for Data Fusion Implementation in a Distributed Electronic System Used in Automotive Industry - Scientific Figure on ResearchGate. Available from: [https://www.researchgate.net/figure/The-logical-architecture-of-the-ySQL-server-18-allows-connection-and-thread-handling\\_fig6\\_352146596](https://www.researchgate.net/figure/The-logical-architecture-of-the-ySQL-server-18-allows-connection-and-thread-handling_fig6_352146596) [accessed 21 Jan 2025]
- Muharam, Y., & Hanifah, W. (2024). RANCANG BANGUN APLIKASI TRACKING SAMPLE BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL DI PT. KAHATEX. In *Jurnal Informatika-COMPUTING* (Vol. 11).
- Herdiana, Y., & Zaman, F. (2024). RANCANG BANGUN APLIKASI PENDEKAR (PELAYANAN DESA KARYALAKSANA) MENGGUNAKAN FRAMEWORK NEXT.JS DALAM MENINGKATKAN PELAYANAN ADMINISTRASI DESA. In *Jurnal Informatika-COMPUTING* (Vol. 11).
- Mohammad Bayu Anggara, Aulia Hanifa Ardita, & Rudi Hermawan. (2024). RANCANG BANGUN APLIKASI SIMPAN PINJAM KOPERASI MENGGUNAKAN FRAMEWORK LARAVEL (STUDI KASUS: KOPERASI MEKAR MANDIRI SEJAHTERA). *Jurnal Informatika*, 11, 68–74.

## LAMPIRAN A. TOR

Sebelum melaksanakan kerja praktik, penulis menerapkan beberapa metode penelitian, yaitu observasi, wawancara, dan studi pustaka. Setelah melakukan pengamatan dan mempelajari lokasi kerja praktik yang telah disetujui oleh pihak Puskesmas Pembantu (Pustu) Arjasari, penulis menemukan bahwa puskesmas ini belum memiliki website instansi maupun sistem pendaftaran online. Selama ini, proses pendaftaran masih dilakukan secara manual. Oleh karena itu, tujuan utama dari kerja praktik ini adalah merancang dan membangun aplikasi pendaftaran layanan puskesmas berbasis web yang dapat diakses oleh pengguna (*user*) dan admin, dengan menyertakan beberapa menu:

1. Menu *Login* : Memuat kolom untuk input *email* serta *password* untuk autentikasi bagi pengguna (*user* atau admin) ke dalam sistem.
2. Menu buat akun : Menu ini dirancang untuk pendaftar dengan menyediakan kolom registrasi data pemilik hewan, meliputi *Email*, *Password*, Nama, NIK, Alamat, dan Kelurahan.
3. Menu *Home* : Menampilkan nomor antrian untuk pendaftar (*user*) serta rincian keluhan pendaftar pada tampilan *home* khusus admin.
4. Menu *Profile* : Berisi informasi profil pengguna, termasuk nama, NIK, alamat, kelurahan, kecamatan, dan nomor WhatsApp.
5. Menu Riwayat : Memuat halaman yang mencatat riwayat konsultasi yang pernah dilakukan oleh pengguna.
6. Menu *Manage Admin* : Menampilkan daftar admin yang memiliki akses ke sistem.
7. Menu *Manage Consultation* : Berisi data konsultasi yang dilakukan pengguna, lengkap dengan status persetujuan (*approve*) atau penolakan (*reject*).
8. Menu *logout* : Menu logout digunakan oleh *user* dan admin untuk keluar dari sistem dengan aman.

## A. Fitur dan Fungsionalitas

### 1. Fitur utama

#### a. Autentikasi pengguna :

- 1) *Login*: Memungkinkan *user* (pemilik hewan) dan admin untuk masuk ke sistem dengan *email* dan *password*.
- 2) Registrasi: *User* dapat membuat akun dengan mengisi data pribadi, seperti nama, alamat, nomor telepon, dan informasi lainnya.

#### b. Managemen *Profile*

- 1) *User*: *User* dapat melihat dan memperbarui informasi profil mereka, seperti nama, NIK, alamat, kelurahan, kecamatan, dan nomor *WhatsApp*.
- 2) Admin: Admin memiliki kemampuan untuk mengelola data profil mereka sendiri serta mendapatkan akses ke berbagai fitur tambahan yang disediakan dalam sistem.

#### c. Pendaftaran Layanan

- 1) Formulir Pendaftaran: *User* dapat mendaftarkan layanan dengan mengisi detail keluhan atau kebutuhan layanan untuk hewan mereka.
- 2) Nomor Antrian: Sistem akan secara otomatis menghasilkan nomor antrian bagi *user* yang telah mendaftar.

#### d. Manajemen Konsultasi

- 1) *Admin*: Melihat daftar pendaftaran yang masuk. Menyetujui (*approve*) atau menolak (*reject*) permohonan pendaftaran. Dan Memberikan status layanan (proses, selesai).
- 2) *User*: Melihat status pendaftaran dan riwayat konsultasi.

#### e. Riwayat Konsultasi

*User* dapat melihat riwayat konsultasi yang mencakup tanggal, keluhan, dan status layanan.

Fitur riwayat konsultasi memungkinkan pengguna untuk mengakses dan meninjau catatan lengkap dari konsultasi yang pernah mereka lakukan. Informasi yang disajikan mencakup detail

konsultasi seperti rincian informasi hewan dan keluhan yang disampaikan.

f. Manajemen Admin

*Admin superuser* dapat mengelola data admin lain, seperti menambahkan, mengedit, atau menghapus akun admin.

g. *Dashboard Admin*

Menampilkan informasi pendaftar, jumlah konsultasi, dan status layanan secara keseluruhan.

h. *Logout*

Fitur *logout* dirancang untuk memungkinkan pengguna, baik *user* maupun admin, keluar dari sistem dengan cara yang aman. Proses ini memastikan bahwa sesi pengguna ditutup sepenuhnya, sehingga data dan informasi pribadi tetap terlindungi dari akses yang tidak sah. Dengan adanya fitur ini, sistem memberikan jaminan keamanan tambahan bagi pengguna setelah mereka selesai menggunakan aplikasi.

## 2. Fungsionalitas

- a. Fungsionalitas pendukung pada aplikasi ini dirancang untuk meningkatkan kenyamanan dan keamanan pengguna. Aplikasi memiliki desain responsif yang memungkinkan akses melalui berbagai perangkat, seperti desktop, tablet, dan smartphone.
- b. Keamanan data dijamin dengan penerapan enkripsi untuk melindungi informasi pengguna serta validasi input guna mencegah masuknya data yang tidak valid atau berbahaya.
- c. Fitur pencarian dan filter mempermudah admin dalam menemukan data pendaftaran berdasarkan kata kunci nama, tanggal, atau status tertentu.
- d. Untuk memastikan keamanan dan pembagian akses yang tepat, sistem ini menggunakan kontrol akses berbasis peran (*multi-level user access*) yang membatasi fitur sesuai dengan hak akses *user* atau admin.

Aplikasi ini dirancang untuk mempermudah proses pendaftaran layanan di Puskesmas Arjasari, baik bagi pengguna maupun admin. Dengan sistem berbasis web, aplikasi ini bertujuan meningkatkan efisiensi layanan dan memberikan pengalaman yang lebih baik bagi pemilik hewan. Melalui fitur pendaftaran online yang praktis dan akses informasi yang transparan, aplikasi ini diharapkan dapat membantu pengguna dalam mendapatkan pelayanan kesehatan hewan secara lebih cepat dan terorganisir.

Bandung, 11 November 2024

Disetujui oleh :

Mahasiswa Kerja Praktik

Pembimbing Lapangan

Riska Nurhayan

NIM. 301210009

Drh. Nur Azizah Rahman

NIP. 199412182023212020

## **LAMPIRAN B. LEMBAR WAWANCARA**

### **Tujuan Wawancara:**

Mengumpulkan data untuk pengembangan aplikasi pendaftaran berbasis web untuk Puskesmas Pembantu (Pustu) Arjasari.

### **Informasi Responden:**

- **Nama:** Drh. Nur Azizah Rahman
- **Jabatan:** Koordinator Puskesmas
- **Tanggal Wawancara:** 10 Oktober 2024

---

### **Pertanyaan dan Jawaban**

- 1. Bagaimana proses pendaftaran pasien dilakukan saat ini di Puskesmas?**

**Jawaban:**

Proses pendaftaran dilakukan secara manual. Pasien datang langsung ke Puskesmas, mengisi formulir kertas, dan menunggu giliran berdasarkan urutan kedatangan. Hal ini sering menyebabkan antrean panjang dan waktu tunggu yang lama.

- 2. Apa saja kendala utama dalam proses pendaftaran yang berjalan saat ini?**

**Jawaban:**

Kendala utama meliputi kesalahan pencatatan data, sulitnya mencari data pasien sebelumnya, waktu pendaftaran yang memakan waktu, dan kurangnya sistem pengelolaan nomor antrean yang efisien.

- 3. Fitur apa saja yang menurut Anda penting untuk dimiliki oleh aplikasi pendaftaran berbasis web?**

**Jawaban:**

Fitur penting meliputi pendaftaran online, pengelolaan data pasien, pengaturan nomor antrean otomatis, pelacakan riwayat konsultasi, notifikasi jadwal konsultasi, dan laporan statistik untuk evaluasi.

**4. Bagaimana data pasien dan konsultasi saat ini disimpan dan dikelola?**

**Jawaban:**

Data disimpan dalam bentuk dokumen fisik dan sebagian di spreadsheet. Pengelolaan data manual ini menyulitkan dalam pencarian dan pembaruan data.

**5. Apakah Anda memiliki preferensi khusus terkait tampilan antarmuka aplikasi?**

**Jawaban:**

Tampilan sebaiknya sederhana, mudah digunakan oleh petugas maupun masyarakat. Warna yang digunakan harus nyaman dilihat, dan navigasi harus jelas serta intuitif.

**6. Apakah Anda memiliki kebutuhan khusus terkait keamanan data?**

**Jawaban:**

Keamanan data sangat penting. Data pasien harus terlindungi dari akses tidak sah, dan aplikasi harus memiliki sistem autentikasi untuk memastikan hanya pengguna terotorisasi yang dapat mengakses informasi.

**7. Apakah ada kendala teknis atau non-teknis yang perlu diperhatikan selama implementasi aplikasi ini?**

**Jawaban:**

Kendala teknis meliputi ketersediaan perangkat komputer dan

jaringan internet di lokasi. Kendala non-teknis mencakup pelatihan petugas untuk menggunakan aplikasi dengan baik.

**8. Bagaimana Anda melihat keberhasilan aplikasi ini di masa depan?**

**Jawaban:**

Jika aplikasi dapat mengurangi antrean, mempercepat proses pendaftaran, dan mempermudah pengelolaan data, maka itu akan sangat membantu operasional Puskesmas dan meningkatkan kepuasan masyarakat.

**9. Apakah ada saran atau masukan lain terkait pengembangan aplikasi ini?**

**Jawaban:**

Pastikan aplikasi diuji coba terlebih dahulu sebelum implementasi penuh. Selain itu, tambahkan fitur laporan untuk membantu evaluasi kinerja dan perencanaan operasional Puskesmas.

---

Bandung, 06 November 2024  
Koordinator Puskesmas,

Drh. Nur Azizah Rahman  
NIP. 199412182023212020



### LAMPIRAN C. LOG ACTIVITY

Minggu/tgl	Kegiatan
1/05 November 2024	Pada minggu pertama, kegiatan dimulai dengan survei dan observasi untuk mengenal lingkungan tempat kerja praktik. Selain itu, dilakukan diskusi dengan pihak administrasi guna memperoleh perizinan yang diperlukan untuk mendukung pelaksanaan kerja praktik.
2/16 November 2024	Minggu kedua diisi dengan observasi terhadap kegiatan sehari-hari di UPT Puskesmas Pembantu Arjasari. Wawancara juga dilakukan dengan dokter yang bertanggung jawab untuk mendapatkan wawasan lebih dalam tentang proses penanganan pasien. Selain itu, analisis sistem dilakukan untuk menyusun kesepakatan awal mengenai perancangan aplikasi, serta pengumpulan data yang relevan sebagai dasar pengembangan aplikasi.
3/23 November 2024	Pada minggu ketiga, alur kerja (workflow) aplikasi mulai disusun berdasarkan hasil analisis kebutuhan. Selanjutnya, rancangan awal antarmuka aplikasi dibuat untuk memastikan bahwa desainnya sesuai dengan kebutuhan pengguna dan mudah digunakan.
4/30 November 2024	Minggu keempat difokuskan pada finalisasi rancangan awal antarmuka aplikasi. Proses ini mencakup verifikasi untuk memastikan antarmuka yang dirancang mudah digunakan dan sesuai dengan kebutuhan pengguna.
5/03 Desember 2024	Pada minggu kelima, pengujian kelayakan rancangan aplikasi dilakukan. Evaluasi menyeluruh dilakukan untuk memastikan rancangan memenuhi standar kelayakan. Setelah itu, rancangan sistem disepakati bersama pembimbing lapangan kerja praktik atau pihak terkait.
6/14	Minggu keenam melanjutkan pengujian kelayakan

Desember 2024	rancangan aplikasi. Hasil evaluasi sebelumnya digunakan untuk melakukan perbaikan agar rancangan semakin optimal.
7/21 Desember 2024	Minggu ketujuh diisi dengan pengujian akhir kelayakan rancangan aplikasi. Tahap ini memastikan bahwa rancangan siap untuk diimplementasikan.
8/28 Desember 2024	Pada minggu terakhir, konsultasi dengan pembimbing kerja praktik dilakukan untuk mendapatkan masukan akhir. Hasil diskusi dicatat dan digunakan sebagai bahan penyusunan laporan yang merangkum seluruh kegiatan selama kerja praktik.

#### LAMPIRAN D. DOKUMENTASI FOTO



Puskesmas Pembantu (Pustu) Arjasari yaitu lokasi yang digunakan untuk terlaksananya kerja praktik.



Ini adalah pada saat melakukan wawancara dengan koordinator sekaligus dokter hewan di Pustu Arjasari dan merupakan pembimbing lapangan kerja praktik.



Dokumentasi saat melakukan konsultasi dan melaporkan perkembangan dari pembuatan Aplikasi Pendaftaran Pustu Arjasari



Foto bersama koordinator/ dokter serta petugas Puskesmas Pembantu (Pustu) Arjasari.