



**KERJA PRAKTIK - EF234603**

**Perancangan Antarmuka dan Implementasi Backend  
Refaktorisasi Aplikasi Pencatatan dan Pemantauan  
Pasien Continuous Ambulatory Peritoneal Dialysis (CAPD)**

Departemen Teknik Informatika - ITS

Jalan Teknik Kimia, Sukolilo, Kota Surabaya, Jawa Timur,  
60111

Periode: 18 Oktober 2024 - 31 Mei 2024

**Oleh:**

Naufal Adli Purnama

5025201195

Putu Ravindra Wiguna

5025201237

**Pembimbing Departemen**

Dr. Baskoro Adi Pratomo., S.Kom., M.Kom.

**Pembimbing Lapangan**

Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



**KERJA PRAKTIK - EF234603**

**Perancangan Antarmuka dan Implementasi Backend Refaktorisasi Aplikasi Pencatatan dan Pemantauan Pasien Continuous Ambulatory Peritoneal Dialysis (CAPD)**

Departemen Teknik Informatika - ITS

Jalan Teknik Kimia, Sukolilo, Kota Surabaya, Jawa Timur,  
60111

Periode: 18 Oktober 2024 - 31 Mei 2024

Oleh:

Naufal Adli Purnama

5025201195

Putu Ravindra Wiguna

5025201237

**Pembimbing Departemen**

Dr. Baskoro Adi Pratomo., S.Kom., M.Kom.

**Pembimbing Lapangan**

Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

**LEMBAR PENGESAHAN  
KERJA PRAKTIK**

**Perancangan Antarmuka dan Implementasi Backend  
Refaktorisasi Aplikasi Pencatatan dan Pemantauan  
Pasien Continuous Ambulatory Peritoneal Dialysis  
(CAPD)**

Oleh:

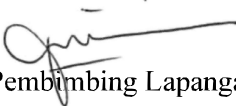
Naufal Adli Purnama	5025201195
Putu Ravindra Wiguna	5025201237

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Baskoro Adi Pratomo.,  
S.Kom., M.Kom.  
NIP. 198702182014041001

  
(Pembimbing Departemen)

2. Dini Adni Navastara,  
S.Kom., M.Sc.  
NIP. 198510172015042001

  
(Pembimbing Lapangan)

**Surabaya, Juni 2024**

*[Halaman ini sengaja dikosongkan]*

# **Perancangan Antarmuka dan Implementasi Backend Refaktorisasi Aplikasi Pencatatan dan Pemantauan Continuous Ambulatory Peritoneal Dialysis (CAPD)**

Nama Mahasiswa : Naufal Adli Purnama  
NRP : 5025201195  
Nama Mahasiswa : Putu Ravindra Wiguna  
NRP : 5025201237  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Departemen : Dr. Baskoro Adi Pratomo., S.Kom.,  
M.Kom.  
Pembimbing Lapangan : Dini Adni Navastara, S.Kom., M.Sc.

## **ABSTRAK**

Continuous Ambulatory Peritoneal Dialysis (CAPD) adalah pengobatan jangka panjang untuk pasien penyakit ginjal stadium akhir yang memerlukan pemantauan rutin. Pasien di RS Universitas Airlangga sebelumnya mencatat perkembangan CAPD dalam buku catatan khusus, yang memiliki beberapa kekurangan seperti inkonsistensi informasi dan kesulitan akses monitoring. Untuk mengatasi masalah ini, tim mahasiswa ITS membuat aplikasi SahabatCAPD untuk membantu pencatatan, tetapi aplikasi tersebut hanya tersedia di platform Android. Oleh karena itu, aplikasi ini direfaktorisasi menjadi Progressive Web App (PWA) menggunakan framework React untuk frontend dan Laravel untuk backend, agar dapat digunakan di berbagai platform, memudahkan pencatatan dan pemantauan penggantian cairan untuk pasien CAPD dan tenaga medis. Hasil pengujian UI/UX yang dilakukan melalui kuesioner menunjukkan bahwa responden menilai kemudahan penggunaan tiga fungsionalitas utama aplikasi dengan rata-rata 4,5 dari skala lima. Ini menunjukkan bahwa aplikasi

SahabatCAPD memiliki tingkat kemudahan penggunaan yang tinggi, sehingga diharapkan dapat lebih efektif dalam membantu pencatatan dan pemantauan bagi pasien dan tenaga medis.

**Kata Kunci : CAPD, Refaktorisasi, Progressive Web App (PWA), Laravel**

*[Halaman ini sengaja dikosongkan]*

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Refaktorisasi Frontend Aplikasi Pencatatan dan Pemantauan Continuous Ambulatory Peritoneal Dialysis.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, segala kritik, saran, dan masukan yang bersifat membangun sangat penulis harapkan guna perbaikan di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada orang-orang yang telah memungkinkan penyusunan buku laporan kerja praktik ini baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Dr. Baskoro Adi Pratomo., S.Kom., M.Kom. selaku dosen pembimbing kerja praktik
3. Ibu Dini Navastara, S.Kom., M.Kom. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., selaku koordinator KP.
5. Dokter dan pasien di Poli CAPD Rumah Sakit UNAIR
6. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 30 Juni 2024

Tim Penulis



*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

<i>LEMBAR PENGESAHAN</i> .....	<i>iii</i>
<i>ABSTRAK</i> .....	<i>v</i>
<i>KATA PENGANTAR</i> .....	<i>viii</i>
<i>DAFTAR ISI</i> .....	<i>x</i>
<i>DAFTAR GAMBAR</i> .....	<i>xiv</i>
<i>DAFTAR TABEL</i> .....	<i>xvi</i>
<i>DAFTAR KODE SEMU</i> .....	<i>xviii</i>
<i>BAB 1 PENDAHULUAN</i> .....	<i>1</i>
1.1. Latar Belakang .....	1
1.2. Tujuan .....	3
1.3. Manfaat .....	3
1.4. Rumusan Masalah .....	4
1.5. Lokasi dan Waktu Kerja Praktik .....	4
1.6. Metodologi Kerja Praktik .....	4
1.6.1. Perumusan Masalah .....	4
1.6.2. Studi Literatur .....	4
1.6.3. Analisis dan Perancangan Sistem .....	5
1.6.4. Implementasi Sistem .....	5
1.6.5. Pengujian dan Evaluasi .....	5
1.7. Sistematika Laporan .....	5
<i>BAB 2 PROFIL PERUSAHAAN</i> .....	<i>8</i>

2.1. Sejarah Perusahaan.....	8
2.2. Visi .....	9
2.3. Misi.....	9
2.4. Struktur Organisasi.....	10
2.5. Laboratorium.....	10
<b>BAB 3 TINJAUAN PUSTAKA.....</b>	<b>15</b>
3.1. Continuous Ambulatory Peritoneal Dialysis (CAPD)....	15
3.2. Refaktorisasi.....	15
3.3. Progressive Web Apps .....	16
3.4. Figma.....	16
3.5. Laravel.....	17
3.6. SQL Server .....	17
3.7. RESTful API .....	17
<b>BAB 4 ANALISIS DAN PERANCANGAN SISTEM.....</b>	<b>20</b>
4.1. Analisis Sistem .....	20
4.2. Definisi Umum Aplikasi.....	20
4.3. Analisis Kebutuhan .....	20
4.3.1. Kebutuhan Fungsional .....	20
4.3.2. Kebutuhan Non-Fungsional.....	22
4.4. Diagram <i>Use Case</i> .....	22
4.5. Use Case Specification .....	23
4.6. Diagram Aktivitas .....	41

4.7. Conceptual Data Model.....	51
4.8. Physical Data Model .....	52
4.9. Deskripsi Data .....	52
<b>BAB 5 IMPLEMENTASI SISTEM .....</b>	<b>63</b>
5.1. Implementasi Design System .....	63
5.2. Desain Tampilan Antarmuka Pengguna .....	63
5.2.1. Antarmuka Pengguna.....	63
5.2.2. Antarmuka Admin .....	71
5.3. Implementasi Source Code Backend.....	78
5.3.1. Migrations.....	79
5.3.2. Seeders .....	80
5.3.3. Eloquent models .....	81
5.3.4. Controllers .....	82
5.3.5. Filters .....	108
<b>BAB 6 PENGUJIAN DAN EVALUASI.....</b>	<b>111</b>
6.1. Tujuan Pengujian.....	111
6.2. Kriteria Pengujian.....	111
6.2.1. UI / UX .....	111
6.2.2. Backend .....	112
6.3. Skenario Pengujian.....	113
6.3.1. Admin .....	113
6.3.2. Pasien.....	114

6.4. Evaluasi Pengujian .....	114
6.4.1.    UI / UX .....	115
6.4.2.    Backend .....	116
<i>BAB 7 KESIMPULAN DAN SARAN</i> .....	<i>120</i>
7.1. Kesimpulan.....	120
7.2. Saran.....	120
<i>DAFTAR PUSTAKA</i> .....	<i>121</i>
<i>BIODATA PENULIS</i> .....	<i>125</i>

## DAFTAR GAMBAR

<i>Gambar 2.1 Struktur organisasi Departemen Teknik Informatika, ITS</i> .....	10
<i>Gambar 4.1 Diagram Use Case Pengguna Aplikasi CAPD</i> .....	22
<i>Gambar 4.2 Diagram Use Case Admin Aplikasi CAPD</i> .....	23
<i>Gambar 4.3 Diagram Use Case Superadmin Aplikasi CAPD</i> ...	23
<i>Gambar 4.4 Diagram Aktivitas Mendaftarkan Akun</i> .....	42
<i>Gambar 4.5 Diagram Aktivitas Login</i> .....	43
<i>Gambar 4.6 Diagram Aktivitas Menambah Resep</i> .....	44
.....	45
<i>Gambar 4.7 Diagram Aktivitas Mencatat Pemasukan Cairan</i> ....	45
<i>Gambar 4.8 Diagram Aktivitas Mencatat Pengeluaran Cairan</i> ..	46
<i>Gambar 4.9 Diagram Aktivitas Memverifikasi Pengguna</i> .....	47
<i>Gambar 4.10 Diagram Aktivitas Melihat Detail Pengguna</i> .....	48
<i>Gambar 4.11 Diagram Aktivitas Mengelola Informasi Dokter</i> ...49	
<i>Gambar 4.12 Diagram Aktivitas Mengelola Informasi Cairan Penggantian</i> .....	50
<i>Gambar 4.13 Diagram Aktivitas Mengelola Informasi Admin</i> ...	51
<i>Gambar 4.14 CDM Aplikasi Sahabat CAPD</i> .....	52
<i>Gambar 4.15 PDM Aplikasi Sahabat CAPD</i> .....	52
<i>Gambar 5.1 Tampilan Halaman Pendaftaran</i> .....	64
<i>Gambar 5.2 Tampilan Halaman Login</i> .....	65
<i>Gambar 5.13 Tampilan Halaman Beranda</i> .....	66
<i>Gambar 5.4 Tampilan Daftar Resep</i> .....	67
<i>Gambar 5.5 Tampilan Daftar Riwayat Kesehatan</i> .....	68
<i>Gambar 5.6 Tampilan Logbook</i> .....	69
<i>Gambar 5.7 Tampilan Formulir Pencatatan Penggantian</i> .....	70
<i>Gambar 5.8 Tampilan Login Admin</i> .....	71
<i>Gambar 5.9 Tampilan Dashboard Admin</i> .....	72
<i>Gambar 5.10 Tampilan Halaman Verifikasi Pengguna</i> .....	73
<i>Gambar 5.11 Tampilan Daftar Pengguna Aktif</i> .....	74

<i>Gambar 5.12 Tampilan Detail Pengguna Aktif.....</i>	<i>75</i>
<i>Gambar 5.13 Tampilan Daftar Dokter.....</i>	<i>76</i>
<i>Gambar 5.14 Tampilan Daftar Jenis Cairan Penggantian.....</i>	<i>77</i>
<i>Gambar 5.15 Tampilan Daftar Admin .....</i>	<i>78</i>
<i>Gambar 5.16 Daftar Migrasi pada aplikasi.....</i>	<i>80</i>
<i>Gambar 5.17 Contoh Seeder untuk Tabel Cairan.....</i>	<i>81</i>
<i>Gambar 5.18 Contoh Model untuk Tabel Pasien.....</i>	<i>82</i>

## DAFTAR TABEL

<i>Tabel 4.1 Kebutuhan Fungsional Aplikasi CAPD.....</i>	<i>21</i>
<i>Tabel 4.2 Kebutuhan Fungsional Admin Aplikasi CAPD .....</i>	<i>21</i>
<i>Tabel 4.3 Kebutuhan Fungsional Superadmin Aplikasi CAPD ..</i>	<i>21</i>
<i>Tabel 4.4 Kebutuhan Non-Fungsional Aplikasi CAPD.....</i>	<i>22</i>
<i>Tabel 4.5 Tabel Use Case Aplikasi CAPD Mendaftarkan Akun .</i>	<i>24</i>
<i>Tabel 4.6 Tabel Use Case Sahabat CAPD Melakukan Login.....</i>	<i>25</i>
<i>Tabel 4.7 Tabel Use Case Sahabat CAPD Menambahkan Resep Penggantian .....</i>	<i>26</i>
<i>Tabel 4.8 Tabel Use Case Sahabat CAPD Mencatat Pemasukan Cairan.....</i>	<i>28</i>
<i>Tabel 4.9 Tabel Use Case Sahabat CAPD Mencatat Pengeluaran Cairan.....</i>	<i>29</i>
<i>Tabel 4.10 Tabel Use Case Sahabat CAPD Memverifikasi Pengguna.....</i>	<i>31</i>
<i>Tabel 4.11 Tabel Use Case Sahabat CAPD Melihat Detail Pencatatan Pengguna Aktif.....</i>	<i>32</i>
<i>Tabel 4.12 Tabel Use Case Sahabat CAPD Mengelola Informasi Dokter.....</i>	<i>33</i>
<i>Tabel 4.13 Tabel Use Case Sahabat CAPD Mengelola Informasi Cairan Penggantian.....</i>	<i>36</i>
<i>Tabel 4.14 Tabel Use Case Sahabat CAPD Mengelola Informasi Admin .....</i>	<i>39</i>
<i>Tabel 4.15 Deskripsi Data pada Tabel Akun Pengguna Aplikasi CAPD .....</i>	<i>53</i>
<i>Tabel 4.16 Deskripsi Data pada Tabel Pasien Aplikasi CAPD ..</i>	<i>53</i>
<i>Tabel 4.17 Deskripsi Data pada Tabel Rumah Sakit Aplikasi CAPD .....</i>	<i>54</i>
<i>Tabel 4.18 Deskripsi Data pada Tabel Dokter Aplikasi CAPD ..</i>	<i>55</i>
<i>Tabel 4.19 Deskripsi Data pada Tabel Riwayat Kesehatan Aplikasi CAPD .....</i>	<i>56</i>



<i>Tabel 4.20 Deskripsi Data pada Tabel Cairan Aplikasi CAPD..</i>	<i>56</i>
<i>Tabel 4.21 Deskripsi Data pada Tabel Resep Aplikasi CAPD....</i>	<i>57</i>
<i>Tabel 4.22 Deskripsi Data pada Tabel Penggantian Aplikasi CAPD .....</i>	<i>57</i>
<i>Tabel 4.23 Deskripsi Data pada Tabel Detail Penggantian Aplikasi CAPD .....</i>	<i>58</i>
<i>Tabel 4.25 Deskripsi Data pada Tabel Pertanyaan Keamanan Aplikasi CAPD .....</i>	<i>59</i>
<i>Tabel 4.26 Deskripsi Data pada Tabel Provinsi Aplikasi CAPD</i>	<i>60</i>
<i>Tabel 4.27 Deskripsi Data pada Tabel Kabupaten/Kota Aplikasi CAPD .....</i>	<i>60</i>
<i>Tabel 4.28 Deskripsi Data pada Tabel Kecamatan Aplikasi CAPD .....</i>	<i>61</i>

## DAFTAR KODE SEMU

<i>Kode Semu 5.1 RegisteredUserController Bagian 1</i> .....	83
<i>Kode Semu 5.2 RegisteredUserController Bagian 2</i> .....	84
<i>Kode Semu 5.3 AuthenticatedSessionController Bagian 1</i> .....	85
<i>Kode Semu 5. 4 AuthenticatedSessionController Bagian 2</i> .....	86
<i>Kode Semu 5.5 UserController</i> .....	87
<i>Kode Semu 5.6 PasienController</i> .....	89
<i>Kode Semu 5.7 PertanyaanKeamananController</i> .....	90
<i>Kode Semu 5.8 DokterController</i> .....	91
<i>Kode Semu 5.9 Rumah Sakit Controller Bagian 1</i> .....	92
<i>Kode Semu 5.10 RumahSakitController Bagian 2</i> .....	93
<i>Kode Semu 5.11 ProvinsiController</i> .....	93
<i>Kode Semu 5.12 KabupatenController</i> .....	94
<i>Kode Semu 5.13 KecamatanController</i> .....	94
<i>Kode Semu 5.14 CairanController</i> .....	95
<i>Kode Semu 5.15 ResepController Bagian 1</i> .....	96
<i>Kode Semu 5.16 ResepController Bagian 2</i> .....	97
<i>Kode Semu 5.17 PenggantianController Bagian 1</i> .....	98
<i>Kode Semu 5.18 PenggantianController Bagian 2</i> .....	99
<i>Kode Semu 5.19 KeluhanController</i> .....	99
<i>Kode Semu 5.20 RiwayatKesehatanController Bagian 1</i> .....	100
<i>Kode Semu 5.21 RiwayatKesehatanController Bagian 2</i> .....	101
<i>Kode Semu 5.22 DetailPenggantianController Bagian 1</i> .....	102
<i>Kode Semu 5. 23 DetailPenggantianController Bagian 2</i> .....	103
<i>Kode Semu 5.24 DetailPenggantianController Bagian 3</i> .....	104
<i>Kode Semu 5.25 StatController</i> .....	105
<i>Kode Semu 5.26 ReportController</i> .....	106
<i>Kode Semu 5.27 AdminController</i> .....	107
<i>Kode Semu 5.28 ApiFilter</i> .....	109

# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Continuous Ambulatory Peritoneal Dialysis (CAPD) merupakan pengobatan yang dilakukan untuk pasien penyakit ginjal stadium akhir. Prosedur CAPD melibatkan pemasukan dan pengeluaran cairan dialisis. Proses pemasukan dan pengeluaran cairan ini dilakukan secara rutin setiap hari [1]. Meskipun CAPD merupakan salah satu pengobatan efektif untuk pasien penyakit ginjal, CAPD tidak terlepas dari risiko komplikasi seperti penyakit peritonitis, infeksi exit site, dan komplikasi dengan kateter [2]. Penyakit seperti peritonitis merupakan penyebab utama kegagalan dalam pengobatan CAPD yang dapat menyebabkan kematian [3].

CAPD merupakan pengobatan jangka panjang dengan risiko komplikasi, sehingga pasien CAPD perlu dipantau secara rutin. Pasien RS Universitas Airlangga yang menjalani CAPD sebelumnya mencatat perkembangannya dalam sebuah buku catatan khusus. Metode ini memiliki beberapa kekurangan. Catatan fisik berarti proses pencatatan yang cukup memakan waktu dan kurang konsisten dalam informasi yang dicatat. Selain itu, buku fisik tidak dapat diintegrasikan ke dalam sistem, sehingga catatan pasien hanya dapat ditinjau ketika pasien sedang berobat langsung di rumah sakit. Hal ini menurunkan kemampuan upaya preventif dan penanganan sigap karena tidak semua pasien berkesempatan untuk ke rumah sakit secara rutin.

Beberapa pasien tinggal di luar kota sehingga hanya datang ke rumah sakit satu bulan sekali ketika perlu menambah persediaan cairan dialisis. Selain itu, pasien tidak dapat diandalkan 100% untuk melaporkan komplikasi yang mereka alami. Dilaporkan bahwa pasien secara global mengalami kesulitan dalam

mengenali komplikasi tersebut [4]. Kombinasi dari kendala ini mengurangi efektivitas pemantauan kesehatan pasien CAPD secara keseluruhan.

Untuk mengatasi masalah tersebut, tim mahasiswa ITS yang diketuai oleh Fiqey Indriati Eka Sari dari Departemen Teknik Informatika dan beranggotakan Muchamad Maroqi Abdul Jalil dari Departemen Teknik Informatika, Nabilla Alvania Nurwardani dari Departemen Biologi, Shinta Ulwiya dari Departemen Perencanaan Wilayah dan Kota (PWK), dan Millads Anwary Fandiaz dari Departemen Teknik Fisika. Dengan bimbingan dari dosen pendamping Dini Adni Navastara S.Kom., M.Sc., tim tersebut berhasil membuat sebuah aplikasi pencatatan digital bernama SahabatCAPD. Aplikasi SahabatCAPD memiliki tiga fungsionalitas utama, yaitu logbook, chatbot, dan model deteksi dini [4]. Setiap fungsionalitas ditujukan untuk memudahkan pencatatan dari sisi pasien dan pemantauan dari sisi tenaga medis. Aplikasi berhasil mendapatkan skor 80 berdasarkan standar System Usability Scale (SUS) sehingga terbukti dapat memudahkan pasien dan tenaga medis.

Kesuksesan dari tim pengembang ingin dikembangkan lebih lanjut agar dapat digunakan oleh lebih banyak pasien dengan perangkat yang berbeda. Aplikasi versi satu dibuat khusus untuk platform Android. Oleh karena itu, diadakan refaktorisasi untuk menghadirkan fungsionalitas SahabatCPD di berbagai jenis perangkat dengan sistem operasi yang beragam. Refaktorisasi ini dilakukan dengan memanfaatkan konsep Progressive Web App (PWA). PWA merupakan tipe aplikasi yang berbasis web, tetapi dapat diunduh secara lokal. Hal ini menggabungkan kemudahan akses multiplatform yang didapatkan dari laman web dengan kemampuan offline yang didapatkan dari aplikasi yang diunduh dari App Store/Playstore.

Untuk pengembangan PWA diperlukan rancangan tampilan serta API. Tugas backend adalah untuk memastikan kelancaran fungsionalitas aplikasi pada berbagai jenis perangkat dengan sistem operasi yang berbeda. Untuk itu, backend perlu mengembangkan dan menyediakan API yang dapat diakses oleh frontend untuk mengirim dan menerima data, serta memastikan keamanan dan kinerja aplikasi. Sementara itu, perancang UI/UX bertanggung jawab atas perancangan antarmuka pengguna yang responsif dan mudah digunakan. Perancang perlu memastikan bahwa tampilan memiliki tata letak yang intuitif, warna dan font yang sesuai, serta memastikan konsistensi pengalaman pengguna di seluruh platform. Selain itu, desain baru harus mempertahankan bahasa desain aplikasi versi pertama.

Kolaborasi antara tim pengembang backend, UI/UX designer, dan anggota tim lainnya, kami diharapkan dapat menghadirkan SahabatCAPD dalam format PWA yang dapat diakses oleh lebih banyak pasien dengan perangkat yang berbeda. Hal ini diharapkan dapat meningkatkan efektivitas dan kenyamanan pemantauan kesehatan pasien CAPD secara keseluruhan.

## **1.2. Tujuan**

Tujuan kerja praktik ini adalah berkontribusi dalam refaktorisasi aplikasi pencatatan dan pemantauan CAPD melalui perancangan UI/UX serta implementasi Backend.

## **1.3. Manfaat**

Manfaat yang diperoleh dengan adanya sistem pencatatan CAPD antara lain adalah:

1. Mempermudah pasien dan keluarga dalam mencatat penggantian cairan CAPD.
2. Mempermudah tenaga medis dalam memantau kondisi

pasien.

#### **1.4. Rumusan Masalah**

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana mendesain antarmuka dan pengalaman pengguna frontend aplikasi pencatatan CAPD menggunakan aplikasi Figma?
2. Bagaimana mengimplementasi backend aplikasi pencatatan CAPD dengan Laravel?

#### **1.5. Lokasi dan Waktu Kerja Praktik**

Pengerjaan kerja praktik ini dilakukan secara jarak jauh dengan periode kerja praktik dimulai pada tanggal 18 Oktober 2023 hingga 31 Mei 2024.

#### **1.6. Metodologi Kerja Praktik**

Metodologi dalam pelaksanaan buku kerja praktik meliputi:

##### **1.6.1. Perumusan Masalah**

Kebutuhan dari website diambil dari rancangan aplikasi sebelumnya. Untuk mengetahui kebutuhan refaktorisasi, kami mengikuti rapat bersama tim developer dengan Ibu Dini dan tim pengembang sistem sebelumnya. Pada saat rapat kami dijelaskan bagaimana alur penggantian cairan CAPD dan penerapannya dalam aplikasi. Selain itu, dijelaskan kebutuhan refaktorisasi aplikasi ke dalam bentuk PWA. Diputuskan untuk membuat aplikasi PWA dengan frontend Javascript, backend Laravel, serta basis data Microsoft SQL Server.

##### **1.6.2. Studi Literatur**

Setelah menentukan *development stack* yang akan digunakan, dilakukan studi literatur mengenai cara implementasi sistem. Pada tahap ini dilakukan proses pencarian dan

pembelajaran *framework* yang akan digunakan untuk mengembangkan aplikasi ini.

### **1.6.3. Analisis dan Perancangan Sistem**

Pada website ini tim developer setuju menggunakan framework Progressive Web App (PWA) dan Next.Js untuk pengembangan sisi *frontend* serta Laravel untuk sisi *backend*. Selain itu, dikembangkan sistem *dashboard* admin untuk memonitor kegiatan pasien.

### **1.6.4. Implementasi Sistem**

Pada tahap ini, desain web dan back-end dirancang terlebih dahulu untuk mempermudah pengembangan aplikasi. Dengan tampilan yang ada, tim front-end mulai implementasi aplikasi. Rapat progres dilakukan setiap minggu secara rutin.

### **1.6.5. Pengujian dan Evaluasi**

Pengujian dilakukan dengan menguji performa aplikasi yang telah dibuat terhadap kebutuhan fungsional dan non-fungsional yang telah ditetapkan. Hal ini akan menjadi indikator kesesuaian implementasi dengan kebutuhan sistem.

## **1.7. Sistematika Laporan**

Laporan KP ini terdiri dari tujuh bab dengan rincian sebagai berikut:

### **1. Bab I Pendahuluan**

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

### **2. Bab II Profil Perusahaan**

Bab ini berisi gambaran umum Departemen Teknik Informatika ITS.

### **3. Bab III Tinjauan Pustaka**

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

#### **4. Bab IV Analisis dan Perancangan Sistem**

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

#### **5. Bab V Implementasi Sistem**

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi aplikasi.

#### **6. Bab VI Pengujian dan Evaluasi**

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

#### **7. Bab VII Kesimpulan dan Saran**

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.



*[Halaman ini sengaja dikosongkan]*

## **BAB 2**

### **PROFIL PERUSAHAAN**

#### **2.1. Sejarah Perusahaan**

Sebagai perkiraan untuk saat ini dan masa yang akan datang, teknologi informasi menjadi tulang punggung pertumbuhan ekonomi bangsa. Saat ini pun invasi teknologi informasi sudah terasa di berbagai bidang kehidupan manusia. Hal ini sepenuhnya disadari oleh pemerintah, sehingga sejak Repelita V yang lalu, pemerintah telah mencanangkan bahwa pengembangan pendidikan tinggi dalam bidang komputer dan informatika merupakan salah satu program prioritas, bersama-sama dengan disiplin ilmu lainnya seperti rekayasa, perilaku, manajemen, akuntansi, dan kesenian.

Pendidikan tinggi diarahkan untuk mempersiapkan bangsa Indonesia dalam menghadapi era pembangunan industri dan informasi. Untuk itu pemerintah melalui Direktorat Jendral Pendidikan Tinggi pada tahun 1985 menginstruksikan untuk membuka Program Studi S1 baru untuk bidang ilmu teknologi komputer di empat universitas atau institut di mana ITS termasuk di dalamnya. Di ITS, program ini awalnya diberi nama Program Studi Teknik Komputer. Namun sejak tahun 1993, nama Program Studi Teknik Komputer diubah menjadi Jurusan Teknik Komputer. Akhirnya, pada tahun 1996 secara resmi jurusan ini berganti nama menjadi Jurusan Teknik Informatika berdasarkan Surat Keputusan Direktur Jendral Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996, tanggal 11 Juli 1996. Pada saat ini, Jurusan Teknik Informatika memperoleh nilai akreditasi A berdasarkan Surat Keputusan Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) Nomor 003/BAN-PT/Ak-X/S1/V/2006, tanggal 18 Mei 2006.

Selain program Sarjana (S1), Jurusan Teknik Informatika

juga menyelenggarakan program Pasca Sarjana (S2) yang dirintis sejak tahun 1994, dengan surat keputusan Direktur Jendral Pendidikan Tinggi No. 2851/D/T/2001, perihal ijin penyelenggaraan Program-Program Studi Jenjang Program Strata-2 (S2) pada Institut Teknologi Sepuluh Nopember Surabaya. Dan pada tahun 2011, Jurusan Teknik Informatika mulai menyelenggarakan program Doktor (S3) [5].

## **2.2. Visi**

Sejalan dengan visi ITS yaitu menjadi perguruan tinggi dengan reputasi internasional dalam ilmu pengetahuan, teknologi, dan seni, terutama yang menunjang industri dan kelautan yang berwawasan lingkungan, maka visi Departemen Informatika adalah menjadi inovator bidang informatika yang unggul di tingkat nasional dengan reputasi internasional, serta berperan aktif dalam upaya memajukan dan mensejahterakan bangsa.

Visi PSTI adalah menjadi lembaga pendidikan dan penelitian di bidang informatika yang unggul di tingkat nasional dan memiliki reputasi internasional [6].

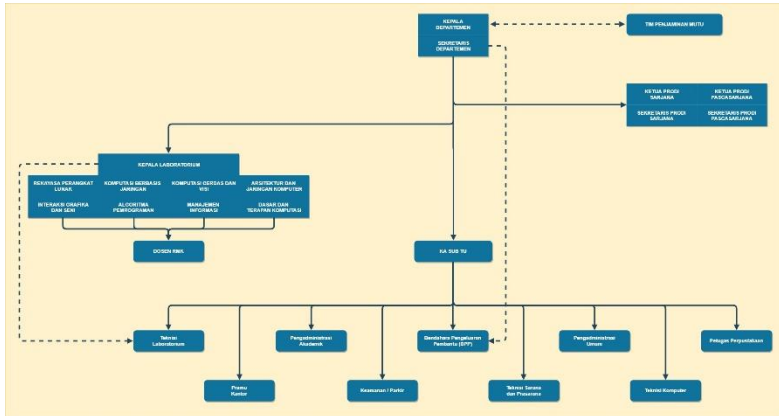
## **2.3. Misi**

Berikut adalah misi-misi yang dimiliki oleh Departemen Teknik Informatika, ITS [6].

1. Menyelenggarakan proses pembelajaran yang berkualitas, dan memenuhi standar nasional maupun internasional.
2. Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
3. Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
4. Menjalin kemitraan dengan berbagai lembaga, baik di dalam maupun di luar negeri.

## 2.4. Struktur Organisasi

Berikut adalah gambar struktur organisasi yang terdapat di Departemen Teknik Informatika, ITS [7].



Gambar 2.1 Struktur organisasi Departemen Teknik Informatika, ITS

## 2.5. Laboratorium

Dalam Departemen Teknik Informatika ITS tersedia fasilitas berupa laboratorium yang dibagi menjadi laboratorium bidang minat dan laboratorium workshop. Jenis-jenis laboratorium bidang minat meliputi rumpun-rumpun mata kuliah yang dapat ditekuni oleh mahasiswa Departemen Teknik Informatika, yaitu sebagai berikut [8].

### 1. Laboratorium Rekayasa Perangkat Lunak (RPL)

Di laboratorium ini ditawarkan bidang minat yang berfokus pada keahlian melakukan pengujian perangkat lunak, kemampuan mengelola proyek perangkat lunak, kemampuan mengurangi resiko kesalahan perangkat lunak, dan kemampuan membuat perangkat lunak game.

### 2. Laboratorium Komputasi Cerdas dan Visi

Di Laboratorium ini ditawarkan bidang keahlian yang

ditekankan pada kemampuan lulusan dalam memanipulasi dan menganalisis data citra pada berbagai bidang aplikasi (A.I., biomedika, industri), kemampuan menerapkan metode sistem cerdas pada berbagai bidang aplikasi dan kemampuan memodelkan dan mengoptimasikan sistem nyata.

### 3. Laboratorium Grafika, Interaksi, dan Game

Laboratorium ini di bidang minat ini menawarkan bidang keahlian yang ditekankan pada kemampuan lulusan dalam mendesain, mengembangkan dan mendokumentasikan proses pembuatan game sesuai dengan standar. Serta membuat model 3 dimensi dan pemrograman di dalam realitas virtual serta aplikasi realitas virtual 3 dimensi dengan menggunakan game engine.

### 4. Laboratorium Manajemen Cerdas Informasi

Laboratorium di bidang minat ini menawarkan bidang keahlian yang ditekankan pada kemampuan lulusan dalam menganalisis, mensintesa dan mengevaluasi proses bisnis dan sistem informasi pada sistem Enterprise, mengimplementasikan rekayasa pengetahuan ke dalam suatu aplikasi, melakukan investigasi, pengujian, evaluasi kematangan dan kepatutan terhadap prosedur standard dan tata kelola teknologi informasi, melakukan tata kelola proyek dan sumber daya manusia dan merancang dan mengimplementasikan solusi basis data terdistribusi dan teknologi Big Data.

### 5. Laboratorium Komputasi Berbasis Jaringan

Di Laboratorium ini ditawarkan bidang keahlian yang ditekankan pada Kemampuan lulusan sarjana/magister/doktor dalam membangun infrastruktur jaringan yang aman, kemampuan membangun sistem grid, Kemampuan membangun aplikasi jaringan sesuai Standard dan Kemampuan membangun aplikasi multimedia berbasis jaringan.

### 6. Laboratorium Teknologi Jaringan dan Keamanan Siber

Cerdas

Laboratorium di bidang minat ini menawarkan bidang keahlian yang ditekankan pada Kemampuan lulusan dalam membangun berbagai macam arsitektur jaringan sesuai standar teknologi terkini dan menerapkan keamanan jaringan.

#### 7. Laboratorium Algoritma dan Pemrograman

Laboratorium ini menawarkan bidang keahlian yang ditekankan pada kemampuan mampu untuk merancang dan menganalisa algoritma dalam menyelesaikan permasalahan secara efektif dan efisien berdasarkan kaidah pemrograman yang kuat, mampu mengaplikasikan model pemrograman yang mendasari berbagai bahasa pemrograman yang ada, serta mampu memilih bahasa pemrograman untuk menghasilkan aplikasi yang sesuai, seperti mengembangkan sistem/aplikasi berbasis kerangka kerja dan pada perangkat bergerak.

#### 8. Laboratorium Pemodelan dan Komputasi Terapan

Laboratorium ini mewadahi riset dan kerjasama industri di bidang pemodelan & simulasi, peramalan sains, optimasi, serta komputasional saintifik.

Laboratorium workshop yang disediakan oleh Departemen Teknik Informatika sebagai berikut.

##### 1. Workshop Pemrograman 1

Laboratorium ini memiliki fasilitas unggulan baik di bidang teknologi management PC dan fasilitas fisik representatif (PC, Video Audio System, Network, dll) serta dapat menampung 54 orang dengan space yang nyaman.

##### 2. Workshop Pemrograman 2

Workshop Pemrograman 2 memiliki fasilitas unggulan baik di bidang teknologi management PC dan fasilitas fisik representatif (PC, Video Audio System, Network, dll) serta dapat menampung 54 orang dengan space yang nyaman.

### 3. Laboratorium Pascasarjana S2

Laboratorium ini merupakan fasilitas mahasiswa program master untuk menyelesaikan tugas-tugas kuliah dan tesis seperti studi literatur, ujicoba aplikasi/data, dan penulisan tesis. Selain laboratorium ini mahasiswa program master dapat juga menggunakan laboratorium bidang minat sesuai dengan bidang penelitian masing- masing.

### 4. Laboratorium Pascasarjana S3

Laboratorium ini merupakan fasilitas mahasiswa program doktor untuk menyelesaikan tugas-tugas kuliah dan disertasi seperti studi literatur, ujicoba aplikasi/data, dan penulisan disertasi. Laboratorium Pascasarjana S3 terdiri dari 3 laboratorium:

- Laboratorium S3 (lantai 1) digunakan oleh mahasiswa S3 yang belum kandidasi.
- Laboratorium S3 (lantai 3) digunakan oleh mahasiswa S3 yang sudah kandidasi.
- Laboratorium S3 Kerjasama (lantai 3) digunakan oleh mahasiswa S3 kerjasama.

Selain laboratorium ini, mahasiswa program doktor dapat juga menggunakan laboratorium bidang minat sesuai dengan bidang penelitian masing- masing.

*[Halaman ini sengaja dikosongkan]*



## **BAB 3**

### **TINJAUAN PUSTAKA**

#### **3.1. Continuous Ambulatory Peritoneal Dialysis (CAPD)**

Continuous Ambulatory Peritoneal Dialysis (CAPD) merupakan dialisis peritoneal (membran abdomen) yang diperkenalkan pada tahun 1976 untuk pengobatan penyakit ginjal stadium akhir [9]. Dalam CAPD, cairan dialisis dimasukkan (infusi) dan dikeluarkan (drainase) beberapa kali setiap hari ke dalam rongga peritoneal [10]. Pengobatan ini cocok untuk pasien dengan profesi yang aktif serta pasien yang masih muda [11].

Meskipun efektif dalam mengobati pasien penyakit ginjal, CAPD memiliki beberapa risiko komplikasi seperti penyakit peritonitis, infeksi *exit site*, dan malfungsi kateter. Komplikasi ini dapat menggagalkan pengobatan CAPD atau bahkan menyebabkan kematian, khususnya untuk peritonitis [10]. Peritonitis yang disebabkan oleh CAPD merupakan salah satu kontributor utama morbiditas dan mortalitas pasien CAPD. CAPD merupakan opsi pengobatan yang mampu meningkatkan kualitas hidup pasien penyakit ginjal. Namun, pasien perlu dipantau secara rutin untuk mitigasi risiko komplikasi yang dapat timbul.

#### **3.2. Refaktorisasi**

Refaktorisasi perangkat lunak merupakan proses yang merestrukturisasi sebuah perangkat lunak tanpa mengubah fungsionalitasnya. Tujuan utama refaktorisasi adalah untuk meningkatkan kualitas desain perangkat lunak dan memudahkan pemeliharaan [12]. Selain itu, refaktorisasi juga merupakan salah satu proses yang memudahkan evolusi dari sebuah perangkat lunak [13].

Penelitian menemukan bahwa refaktorisasi berdampak positif pada kualitas perangkat lunak dan meningkatkan

kemampuan pemeliharaan jangka panjang [14]. Peningkatan kualitas terjadi karena desain yang diperbaiki dan kode sumber dirancang agar semakin mudah dibaca. Selain itu, proses refaktorisasi dapat mengurangi jumlah *bug* sehingga kualitas keseluruhan perangkat lunak mengalami peningkatan dari iterasi terakhir [15].

### **3.3. Progressive Web Apps**

*Progressive Web Apps* (PWA) adalah jenis aplikasi web yang menyediakan pengalaman pengguna serupa dengan aplikasi mobile native, tetapi dapat diakses melalui browser web. Salah satu fitur utama PWA adalah kemampuannya untuk diakses secara offline dengan menyimpan data di cache, sehingga pengguna dapat tetap berinteraksi dengan aplikasi dalam kondisi tidak terhubung jaringan [16].

Selain itu, PWA juga menawarkan pengalaman pengguna yang responsif dan cepat melalui teknologi seperti Service Workers. Service Workers memungkinkan aplikasi menyimpan dan mengelola data di latar belakang, meningkatkan kinerja dan responsivitas aplikasi. Dengan cara ini, PWA memberikan solusi yang lebih ringan dan mudah diakses daripada aplikasi native, tanpa memerlukan pengunduhan atau instalasi melalui *App Store*, sehingga lebih mudah diakses oleh pengguna melalui browser web pada berbagai perangkat [17].

### **3.4. Figma**

Figma merupakan aplikasi desain yang digunakan untuk merancang tampilan dan pengalaman laman web, aplikasi mobile, dan produk digital lainnya. Aplikasi ini makin lazim digunakan dalam proses perancangan dan pengembangan produk digital karena membantu desainer, manajer, dan pengembang berkolaborasi untuk meningkatkan kinerja tim [18].

### **3.5. Laravel**

Laravel adalah kerangka kerja (framework) pengembangan web yang ditulis dalam bahasa pemrograman PHP. Dikembangkan oleh Taylor Otwell, Laravel dirancang untuk membuat pengembangan aplikasi web lebih cepat dan lebih mudah dengan menyediakan berbagai alat dan fitur yang powerful.

Selain itu, Laravel menyediakan berbagai macam pustaka dan modul yang dapat diintegrasikan untuk mempermudah pengembangan aplikasi web dengan fitur-fitur seperti otentikasi pengguna, manajemen sesi, dan routing yang efisien. Dengan dokumentasi yang kuat, Laravel menjadi pilihan populer di kalangan pengembang web untuk membangun aplikasi web modern dan skala besar [19].

### **3.6. SQL Server**

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) milik Microsoft. Ada beberapa edisi SQL Server yang didukung oleh Microsoft yaitu SQL Server 2022, SQL Server 2019, SQL Server 2017, dan SQL Server 2016. Mesin basis data dari SQL Server ini merupakan layanan utama dari beberapa teknologi yang tersedia di SQL Server untuk menyimpan, memproses, dan mengamankan data. Mesin basis data ini menyediakan akses terkendali dan pemrosesan transaksi untuk memenuhi kebutuhan aplikasi yang ada, serta menyediakan dukungan untuk mempertahankan kelangsungan bisnis seperti *database recovery* [20].

### **3.7. RESTful API**

RESTful API adalah suatu antarmuka yang memungkinkan dua sistem komputer untuk berkukar informasi dengan aman melalui internet. API ini mengikuti prinsip-prinsip REST seperti *statelessness*, antarmuka yang seragam, kemampuan *cache*, dan sistem berlapis, menggunakan metode HTTP seperti

GET, POST, PUT, DELETE, dan umumnya bertukar data dalam format JSON. RESTful API ini bersifat fleksibel dan independen dari teknologi, sehingga ideal untuk komunikasi dan integrasi antar sistem dalam aplikasi modern [21].

*[Halaman ini sengaja dikosongkan]*

## **BAB 4**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **4.1. Analisis Sistem**

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun infrastruktur aplikasi sistem penerimaan peserta didik baru yaitu analisis dari infrastruktur sistem yang akan dibangun. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum aplikasi dan analisis kebutuhan.

#### **4.2. Definisi Umum Aplikasi**

Aplikasi CAPD merupakan sistem pencatatan penggantian cairan CAPD. Sistem ini memiliki dua *view* yang dipisahkan berdasarkan tipe pengguna seperti berikut:

1. Pengguna, pasien atau keluarga yang dapat memasukkan resep penggantian cairan CAPD serta mencatat pemasukan dan pengeluaran cairan.
2. Admin, petugas rumah sakit yang dapat memverifikasi pengguna, memasukkan informasi cairan dan dokter, serta memantau pasien.

#### **4.3. Analisis Kebutuhan**

Dalam aplikasi ini, terdapat fungsi-fungsi yang harus dipenuhi oleh sistem. Kebutuhan ini terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

##### **4.3.1. Kebutuhan Fungsional**

Kebutuhan fungsional pada aplikasi ini menjelaskan apa saja yang dapat dilakukan pengguna sistem. Kebutuhan fungsional dari aplikasi CAPD dijelaskan pada Tabel 4.1, 4.2, dan 4.3.

*Tabel 4.1 Kebutuhan Fungsional Aplikasi CAPD*

<b>Kode Kebutuhan</b>	<b>Deskripsi Kebutuhan</b>
F-001	Mendaftarkan Akun
F-002	Masuk ke dalam Aplikasi (Login)
F-003	Menambahkan Resep Penggantian
F-004	Mencatat Pemasukan Cairan
F-005	Mencatat Pengeluaran Cairan

*Tabel 4.2 Kebutuhan Fungsional Admin Aplikasi CAPD*

<b>Kode Kebutuhan</b>	<b>Deskripsi Kebutuhan</b>
F-101	Memverifikasi Pengguna
F-102	Melihat Detail Pencatatan Pengguna Aktif
F-103	Mengelola Informasi Dokter
F-104	Mengelola Informasi Cairan Penggantian

*Tabel 4.3 Kebutuhan Fungsional Superadmin Aplikasi CAPD*

<b>Kode Kebutuhan</b>	<b>Deskripsi Kebutuhan</b>
F-201	Mengelola Informasi Admin

### 4.3.2. *Kebutuhan Non-Fungsional*

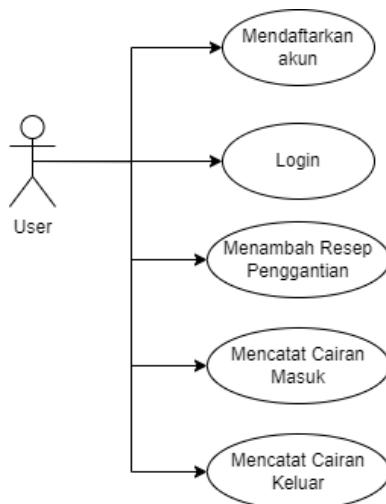
Kebutuhan non-fungsional adalah batasan dan karakteristik dari sebuah sistem. Kebutuhan non-fungsional dari aplikasi CAPD terdapat pada Tabel 4.4.

*Tabel 4.4 Kebutuhan Non-Fungsional Aplikasi CAPD*

Kode Kebutuhan	Deskripsi Kebutuhan
NF-001	Sistem dapat diakses oleh pengguna

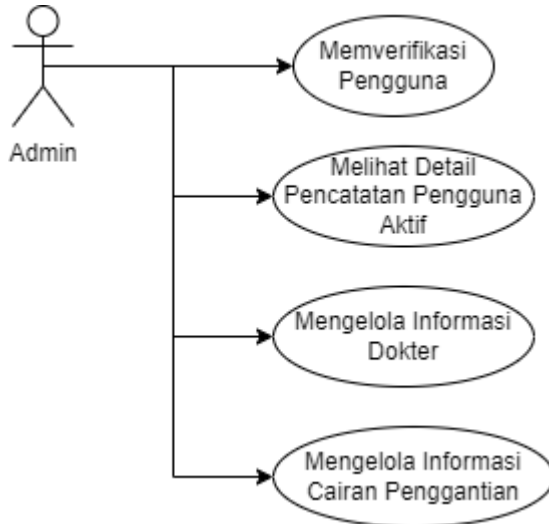
### 4.4. *Diagram Use Case*

Kebutuhan fungsional pengguna digambarkan dalam bentuk *Use Case Diagram* untuk memudahkan pemahaman fungsional aplikasi CAPD.

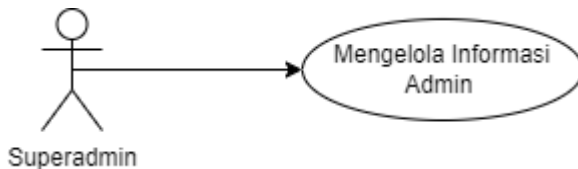


*Gambar 4.1 Diagram Use Case Pengguna Aplikasi CAPD*





Gambar 4.2 Diagram Use Case Admin Aplikasi CAPD



Gambar 4.3 Diagram Use Case Superadmin Aplikasi CAPD

#### 4.5. Use Case Specification

*Use Case Specification* menggambarkan interaksi antara pengguna dengan sistem serta langkah-langkah yang diambil pengguna untuk menjalankan sebuah *use case*.

a. Mendaftarkan Akun

Berikut adalah tabel *use case* mendaftar akun.

Tabel 4.5 Tabel Use Case Aplikasi CAPD Mendaftarkan Akun

Nama	Mendaftarkan Akun
Kode	UC001
Deskripsi	Aktor dapat mendaftar akun baru
Tipe	Fungsional
Pemicu	Aktor mengisi formulir data diri, lalu menekan tombol 'Daftar' pada halaman daftar.
Aktor	User (pasien/keluarga)
Kondisi Awal	Form daftar ditampilkan
Kondisi Akhir	Akun aktor muncul di daftar verifikasi pada dashboard admin.
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor mengisi form daftar</li> <li>2. Aktor menekan tombol 'Daftar'</li> <li>3. Sistem memasukkan akun baru ke daftar untuk diverifikasi</li> </ol>
Alur Kejadian Alternatif	<ol style="list-style-type: none"> <li>1. Aktor tidak mengisi formulir dengan lengkap             <ol style="list-style-type: none"> <li>a. Sistem memberi peringatan bahwa kolom harus diisi.</li> <li>b. Kembali ke Alur Normal nomor 1</li> </ol> </li> </ol>

Pengecualian	-
--------------	---

b. Melakukan Login

Berikut adalah tabel *use case* melakukan login.

*Tabel 4.6 Tabel Use Case Sahabat CAPD Melakukan Login*

Nama	Melakukan login
Kode	UC002
Deskripsi	Aktor dapat masuk ke akun sesuai role
Tipe	Fungsional
Pemicu	Aktor menekan tombol ‘Masuk’ setelah mengisi username dan password pada halaman login
Aktor	Seluruh tipe akun (pasien/keluarga dan admin)
Kondisi Awal	Form login ditampilkan
Kondisi Akhir	Aktor dapat menggunakan sistem sesuai kewenangannya
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor mengisi form login</li> <li>2. Aktor menekan tombol ‘Masuk’</li> <li>3. Sistem mencocokkan data login dengan database</li> <li>4. Sistem menampilkan halaman beranda</li> </ol>

Alur Kejadian Alternatif	<ol style="list-style-type: none"> <li>1. Aktor tidak mengisi formulir dengan lengkap <ol style="list-style-type: none"> <li>a. Sistem memberi peringatan bahwa kolom harus diisi.</li> <li>b. Kembali ke Alur Normal nomor 1.</li> </ol> </li> <li>3. Data yang diinputkan tidak cocok dengan basis data <ol style="list-style-type: none"> <li>a. Sistem memberi peringatan bahwa email atau password salah.</li> <li>b. Kembali ke Alur Normal nomor 1.</li> </ol> </li> </ol>
Pengecualian	-

c. Menambahkan Resep Penggantian

Berikut adalah tabel *use case* menambahkan resep penggantian.

*Tabel 4.7 Tabel Use Case Sahabat CAPD Menambahkan Resep Penggantian*

Nama	Menambahkan Resep Penggantian
Kode	UC003
Deskripsi	Aktor dapat menambahkan resep penggantian
Tipe	Fungsional

Pemicu	Aktor menekan tombol 'Tambah' pada halaman daftar resep
Aktor	User
Kondisi Awal	Aktor ingin menambahkan resep penggantian
Kondisi Akhir	Resep baru masuk ke dalam daftar resep
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol 'Resep' pada halaman beranda</li> <li>2. Sistem akan menampilkan halaman berisi resep pengguna yang terdaftar</li> <li>3. Aktor menekan tombol 'Tambah'</li> <li>4. Sistem menampilkan halaman formulir resep</li> <li>5. Aktor mengisi informasi resep</li> <li>6. Aktor menekan tombol 'Simpan'</li> <li>7. Sistem menyimpan resep baru</li> <li>8. Sistem menampilkan kembali daftar resep</li> </ol>
Alur Kejadian Alternatif	-
Pengecualian	-

d. Mencatat Pemasukan Cairan

Berikut adalah tabel *use case* mencatat pemasukan cairan.

*Tabel 4.8 Tabel Use Case Sahabat CAPD Mencatat Pemasukan Cairan*

Nama	Mencatat Pemasukan Cairan
Kode	UC004
Deskripsi	Aktor dapat mencatat pemasukan cairan
Tipe	Fungsional
Pemicu	Aktor menekan tombol ‘Catat’ di logbook pada jam pemasukan yang ingin dicatat
Aktor	User (pasien/keluarga)
Kondisi Awal	Aktor ingin mencatat cairan masuk
Kondisi Akhir	Catatan pemasukan cairan tersimpan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan ‘Catat’ di logbook pada jam pemasukan yang ingin dicatat</li> <li>2. Sistem menampilkan formulir cairan masuk</li> <li>3. Aktor mengisi formulir</li> <li>4. Aktor menekan ‘Simpan’</li> <li>5. Sistem menyimpan catatan</li> <li>6. Sistem menampilkan halaman logbook pada hari tersebut</li> </ol>

Alur Kejadian Alternatif	-
Pengecualian	-

e. Mencatat Pengeluaran Cairan

Berikut adalah tabel *use case* pengeluaran cairan.

*Tabel 4.9 Tabel Use Case Sahabat CAPD Mencatat Pengeluaran Cairan*

Nama	Mencatat Pengeluaran Cairan
Kode	UC005
Deskripsi	Aktor dapat mencatat pengeluaran cairan
Tipe	Fungsional
Pemicu	Aktor menekan tombol 'Catat' di logbook pada jam pengeluaran yang ingin dicatat
Aktor	User (pasien/keluarga)
Kondisi Awal	Aktor ingin mencatat cairan keluar

Kondisi Akhir	Catatan pengeluaran cairan tersimpan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan ‘Catat’ di logbook pada jam pengeluaran yang ingin dicatat</li> <li>2. Sistem menampilkan formulir cairan keluar</li> <li>3. Aktor mengisi formulir</li> <li>4. Aktor menekan ‘Simpan’</li> <li>5. Sistem menyimpan catatan</li> <li>6. Sistem menampilkan halaman logbook pada hari tersebut</li> </ol>
Alur Kejadian Alternatif	<ol style="list-style-type: none"> <li>3. Aktor menekan ‘Ambil Foto Cairan Buangan’ <ol style="list-style-type: none"> <li>a. Sistem menampilkan aplikasi kamera</li> <li>b. Aktor mengambil foto cairan</li> <li>c. Sistem menyimpan dan melampirkan foto pada pengeluaran tersebut</li> <li>d. Kembali ke langkah 3</li> </ol> </li> </ol>
Pengecualian	-

f. Memverifikasi Pengguna

Berikut adalah tabel *use case* memverifikasi pengguna.



*Tabel 4.10 Tabel Use Case Sahabat CAPD Memverifikasi Pengguna*

Nama	Memverifikasi Pengguna
Kode	UC101
Deskripsi	Admin dapat memverifikasi pengguna yang telah mendaftarkan akun agar dapat masuk aplikasi dan melakukan pencatatan
Tipe	Fungsional
Pemicu	Aktor menekan tombol ‘Verifikasi Pengguna’ di sidebar.
Aktor	Admin
Kondisi Awal	Admin masuk dashboard admin.
Kondisi Akhir	Pengguna terverifikasi dan dapat menggunakan aplikasi
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan ‘Verifikasi Pengguna’ di sidebar</li> <li>2. Sistem menampilkan daftar pengguna yang menunggu verifikasi</li> <li>3. Aktor menekan tombol ‘Setujui’ untuk memverifikasi akun</li> </ol>

	4. Sistem mengubah status akun pengguna
Alur Kejadian Alternatif	-
Pengecualian	-

g. Melihat Detail Pencatatan Pengguna Aktif

Berikut adalah tabel *use case* melihat detail pencatatan pengguna aktif.

*Tabel 4.11 Tabel Use Case Sahabat CAPD Melihat Detail Pencatatan Pengguna Aktif*

Nama	Melihat Detail Pencatatan Pengguna Aktif
Kode	UC102
Deskripsi	Aktor dapat melihat detail pengguna.
Tipe	Fungsional
Pemicu	Aktor menekan tombol 'Pengguna Aktif' di sidebar.
Aktor	Admin

Kondisi Awal	Admin masuk dashboard admin.
Kondisi Akhir	Catatan pengeluaran cairan tersimpan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan ‘Pengguna Aktif’ di sidebar.</li> <li>2. Sistem menampilkan daftar pengguna.</li> <li>3. Aktor menekan tombol ‘detail’ pada pengguna yang ingin dilihat.</li> <li>4. Sistem menampilkan detail pengguna tersebut.</li> </ol>
Alur Kejadian Alternatif	-
Pengecualian	-

#### h. Mengelola Informasi Dokter

Berikut adalah tabel *use case* mengelola informasi dokter.

*Tabel 4.12 Tabel Use Case Sahabat CAPD Mengelola Informasi Dokter*

Nama	Mengelola Informasi Dokter
Kode	UC103
Deskripsi	Aktor dapat mengelola informasi dokter.

Tipe	Fungsional
Pemicu	Aktor menekan tombol 'Dokter' di sidebar.
Aktor	Admin
Kondisi Awal	Admin masuk dashboard admin.
Kondisi Akhir	Informasi daftar dokter diperbarui.
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan 'Dokter' di sidebar.</li> <li>2. Sistem menampilkan daftar dokter.</li> <li>3. Aktor melakukan operasi pada informasi dokter.</li> <li>4. Sistem menyimpan perubahan.</li> </ol>
Alur Kejadian Alternatif	<p>3a. Aktor menambah data dokter</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol 'Tambah Data Dokter'.</li> <li>ii. Sistem menampilkan form informasi dokter.</li> <li>iii. Aktor mengisi form informasi dokter.</li> <li>iv. Aktor menekan tombol 'Simpan'.</li> </ol>

	<p>3b. Aktor menghapus data dokter</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol 'Hapus' pada baris informasi dokter yang ingin dihapus.</li> <li>ii. Sistem menghapus informasi dokter.</li> </ol> <p>3b. Aktor mengedit informasi dokter</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol 'Edit' pada baris informasi dokter yang ingin diubah.</li> <li>ii. Sistem menampilkan form informasi dokter.</li> <li>iii. Aktor mengisi form informasi dokter.</li> <li>iv. Aktor menekan tombol 'Simpan'.</li> </ol>
Pengecualian	-

i. Mengelola Informasi Cairan Penggantian

Berikut adalah tabel *use case* mengelola informasi cairan penggantian.

*Tabel 4.13 Tabel Use Case Sahabat CAPD Mengelola Informasi Cairan Penggantian*

Nama	Mengelola Informasi Cairan Penggantian
Kode	UC104
Deskripsi	Aktor dapat mengelola informasi cairan penggantian.
Tipe	Fungsional
Pemicu	Aktor menekan tombol 'Cairan Penggantian' di sidebar.
Aktor	Admin
Kondisi Awal	Admin masuk dashboard admin.
Kondisi Akhir	Informasi daftar cairan penggantian diperbarui.
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan 'Cairan Penggantian' di sidebar.</li> <li>2. Sistem menampilkan daftar cairan penggantian.</li> <li>3. Aktor melakukan operasi pada informasi cairan penggantian.</li> </ol>

	4. Sistem menyimpan perubahan.
Alur Kejadian Alternatif	<p>3a. Aktor menambah data cairan penggantian</p> <ul style="list-style-type: none"> <li>i. Aktor menekan tombol ‘Tambah Data Cairan’.</li> <li>ii. Sistem menampilkan form informasi cairan penggantian.</li> <li>iii. Aktor mengisi form informasi cairan penggantian.</li> <li>iv. Aktor menekan tombol ‘Simpan’.</li> <li>v. Sistem menyimpan informasi cairan penggantian.</li> </ul> <p>3b. Aktor menghapus data cairan penggantian</p> <ul style="list-style-type: none"> <li>i. Aktor menekan tombol ‘Hapus’ pada baris informasi cairan penggantian yang ingin dihapus.</li> <li>ii. Sistem menghapus informasi cairan penggantian.</li> </ul>

	<p>3b. Aktor mengedit informasi cairan penggantian</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol 'Edit' pada baris informasi cairan penggantian yang ingin diubah.</li> <li>ii. Sistem menampilkan form informasi cairan penggantian.</li> <li>iii. Aktor mengisi form informasi cairan penggantian.</li> <li>iv. Aktor menekan tombol 'Simpan'.</li> <li>v. Sistem menyimpan informasi cairan penggantian.</li> </ol>
Pengecualian	-

j. Mengelola Informasi Admin

Berikut adalah tabel *use case* mengelola informasi admin.



*Tabel 4.14 Tabel Use Case Sahabat CAPD Mengelola Informasi Admin*

Nama	Mengelola Informasi Admin
Kode	UC103
Deskripsi	Aktor dapat mengelola informasi admin.
Tipe	Fungsional
Pemicu	Aktor menekan tombol 'Admin' di sidebar.
Aktor	Superadmin
Kondisi Awal	Superadmin masuk dashboard superadmin.
Kondisi Akhir	Informasi daftar admin diperbarui.
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan 'Cairan Penggantian' di sidebar.</li> <li>2. Sistem menampilkan daftar cairan penggantian.</li> <li>3. Aktor melakukan operasi pada informasi cairan penggantian.</li> </ol>

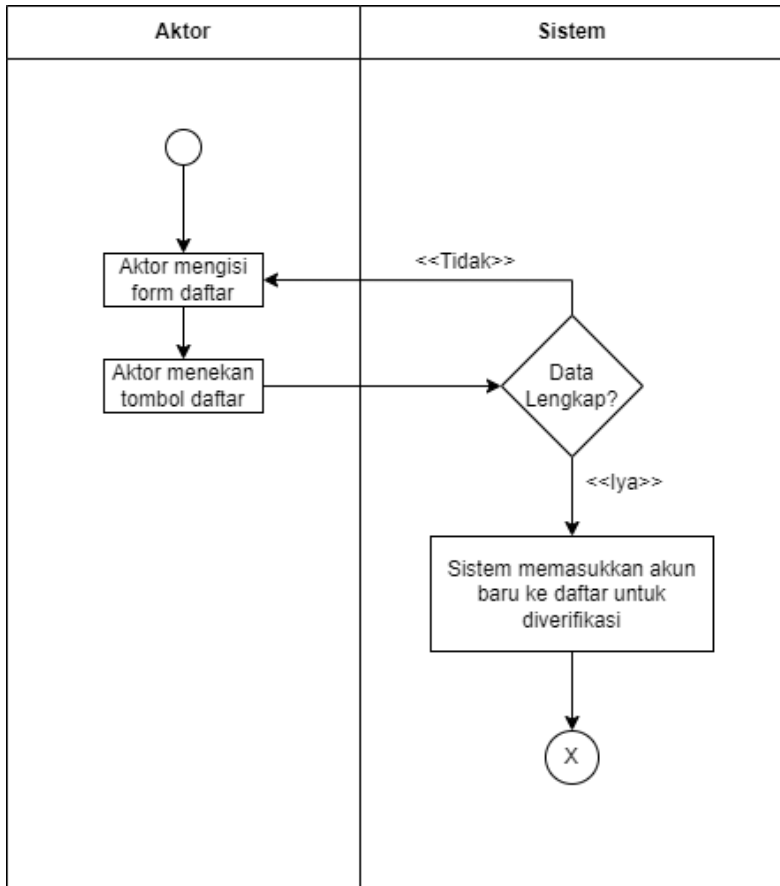
	4. Sistem menyimpan perubahan.
Alur Kejadian Alternatif	<p>3a. Aktor menambah data admin</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol ‘Tambah Data Admin’.</li> <li>ii. Sistem menampilkan form informasi admin.</li> <li>iii. Aktor mengisi form informasi admin.</li> <li>iv. Aktor menekan tombol ‘Simpan’.</li> <li>v. Sistem menyimpan informasi admin.</li> </ol> <p>3b. Aktor menghapus data admin</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol ‘Hapus’ pada baris informasi admin yang ingin dihapus.</li> <li>ii. Sistem menghapus informasi admin.</li> </ol> <p>3b. Aktor mengedit informasi admin</p> <ol style="list-style-type: none"> <li>i. Aktor menekan tombol ‘Edit’ pada baris informasi admin yang ingin diubah.</li> <li>ii. Sistem menampilkan form informasi admin.</li> <li>iii. Aktor mengisi form</li> </ol>

	<p>informasi admin.</p> <p>iv. Aktor menekan tombol 'Simpan'.</p> <p>v. Sistem menyimpan informasi admin.</p>
Pengecualian	-

#### 4.6. Diagram Aktivitas

##### a. Mendaftarkan Akun

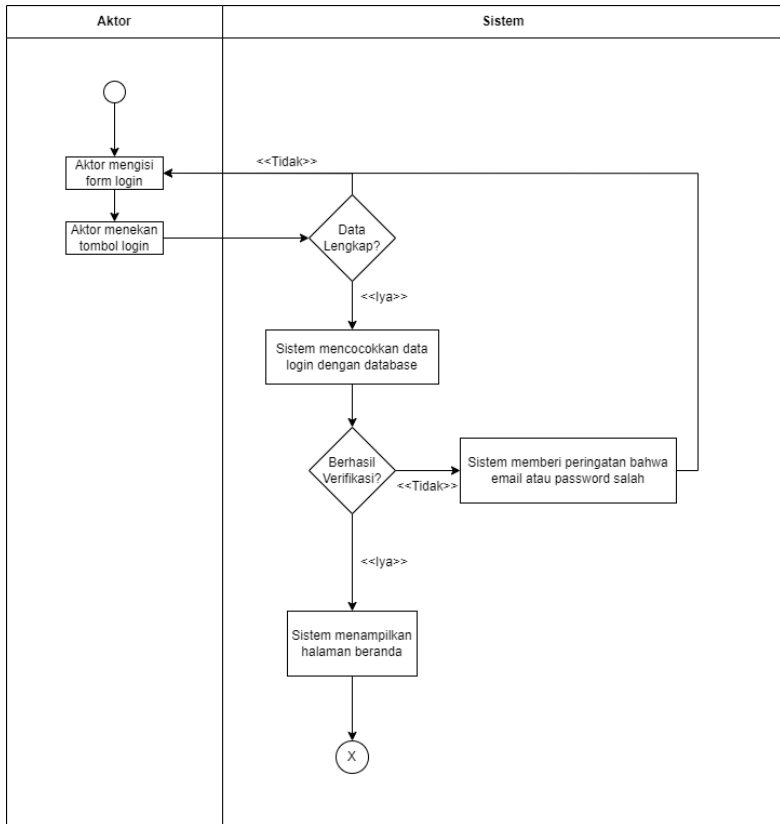
Gambar berikut merupakan diagram aktivitas untuk *use case* Mendaftarkan Akun (UC001).



Gambar 4.4 Diagram Aktivitas Mendaftarkan Akun

b. Masuk ke Aplikasi (Login)

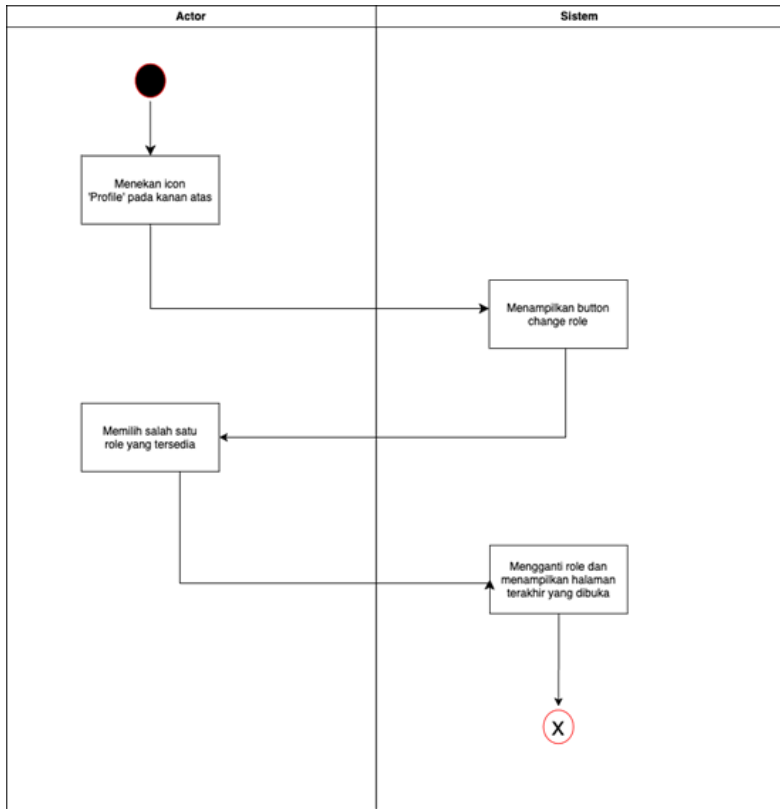
Gambar berikut merupakan diagram aktivitas dari *use case* Masuk ke Aplikasi (Login) (UC002).



Gambar 4.5 Diagram Aktivitas Login

c. Menambah Resep

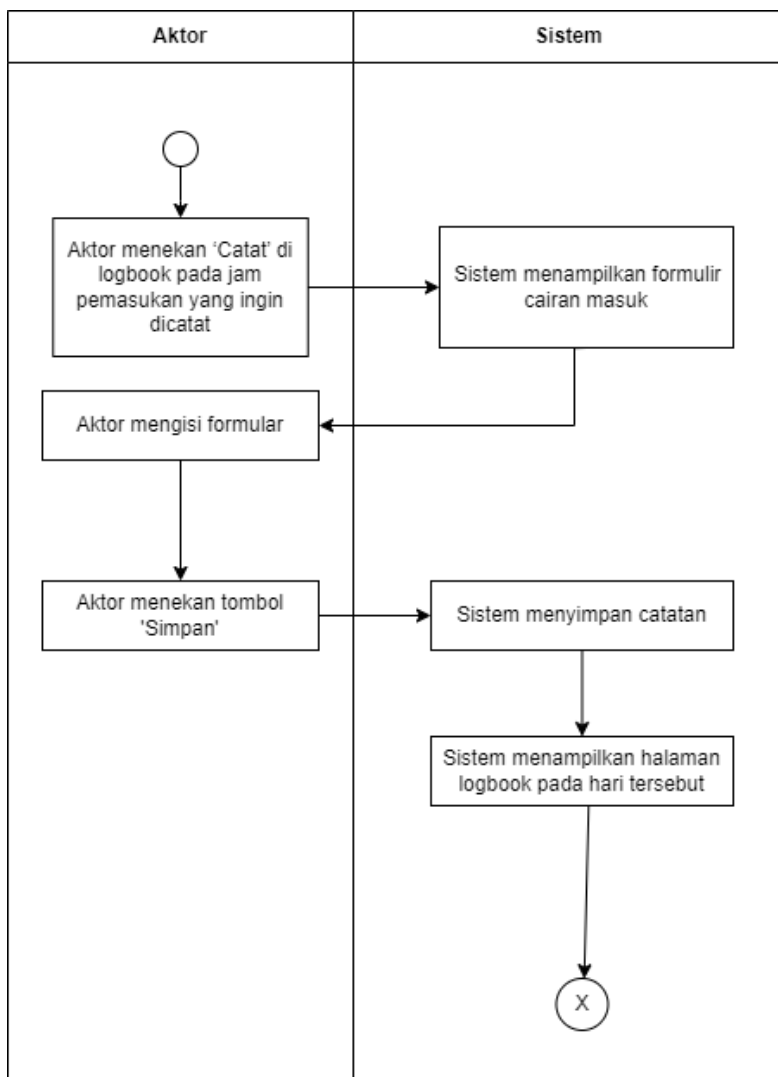
Gambar berikut merupakan diagram aktivitas dari *use case* Menambah Resep (UC003).



*Gambar 4.6 Diagram Aktivitas Menambah Resep*

d. Mencatat Pemasukan Cairan

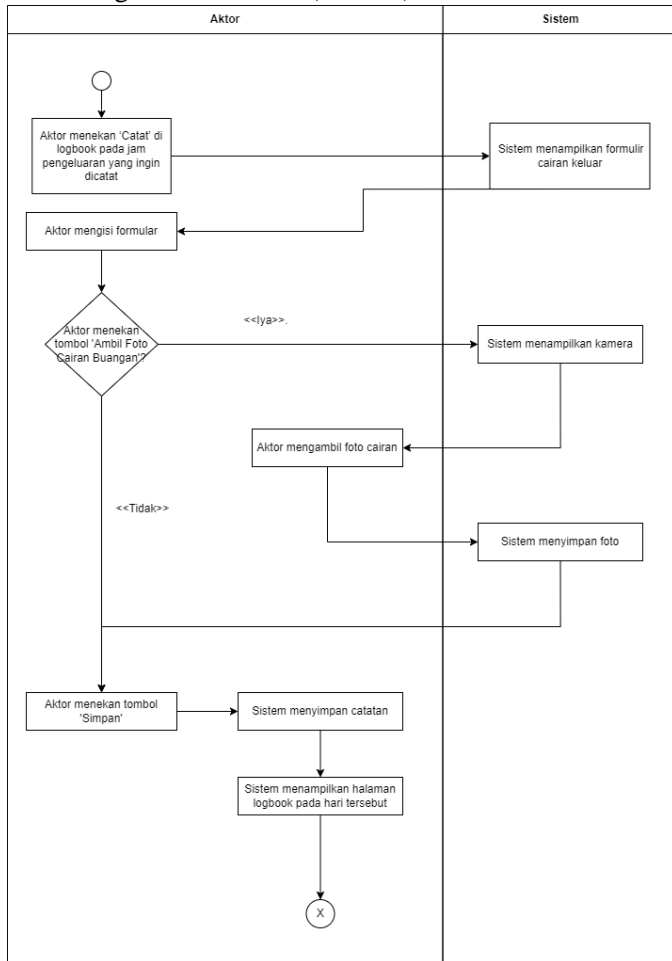
Gambar berikut merupakan diagram aktivitas dari *use case* Mencatat Pemasukan Cairan (UC004).



*Gambar 4.7 Diagram Aktivitas Mencatat Pemasukan Cairan*

e. Mencatat Pengeluaran Cairan

Gambar berikut merupakan diagram aktivitas dari *use case* Mencatat Pengeluaran Cairan (UC005).

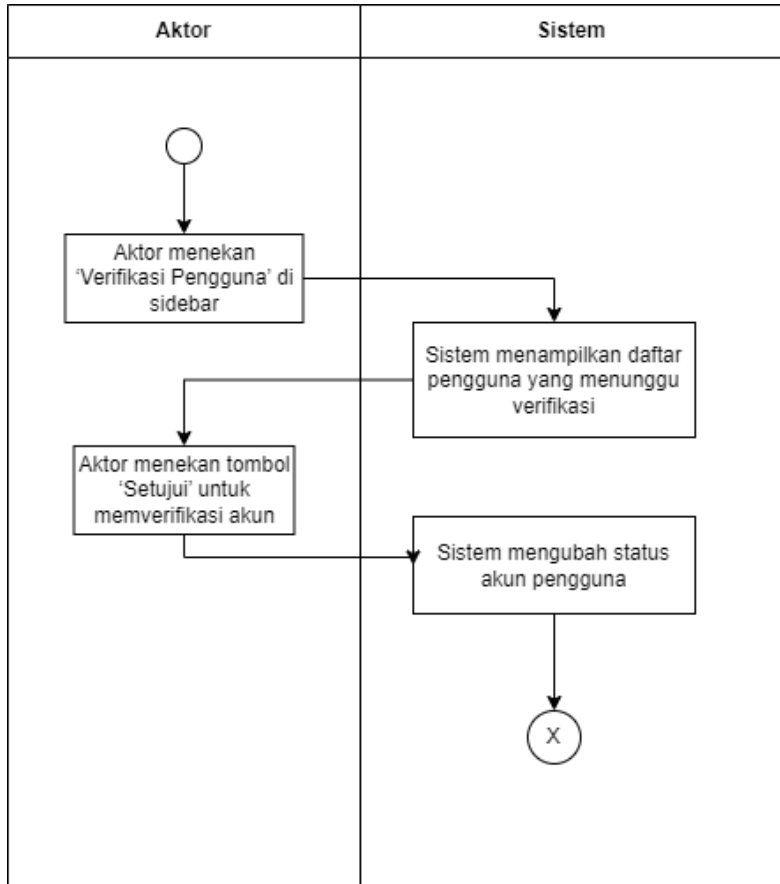


Gambar 4.8 Diagram Aktivitas Mencatat Pengeluaran Cairan



f. Memverifikasi Pengguna

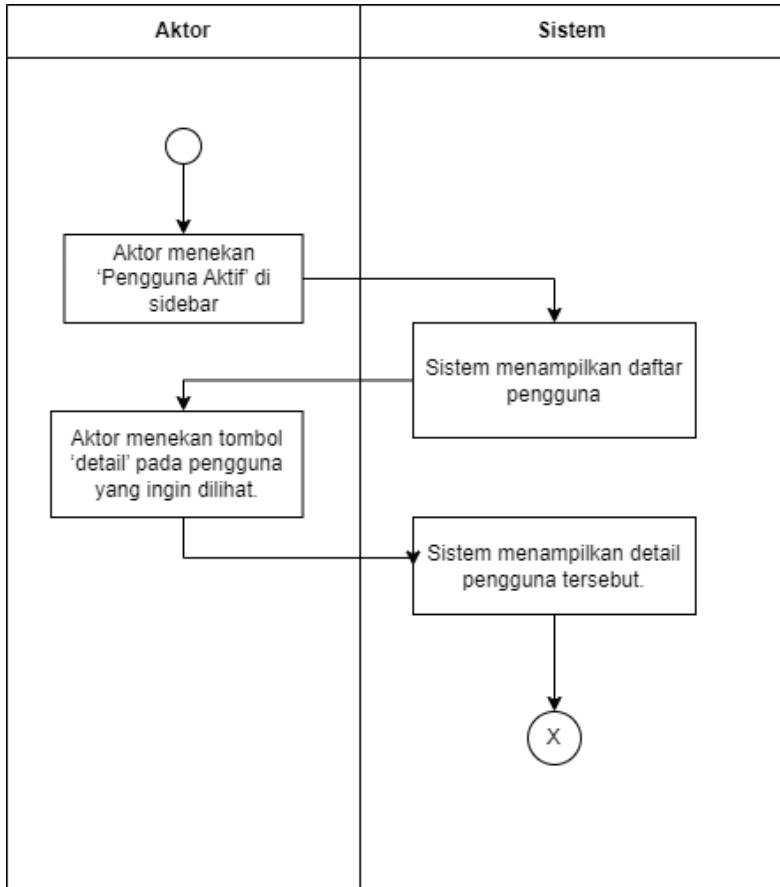
Gambar berikut merupakan diagram aktivitas dari *use case* Memverifikasi Pengguna (UC101).



Gambar 4.9 Diagram Aktivitas Memverifikasi Pengguna

g. Melihat Detail Pengguna

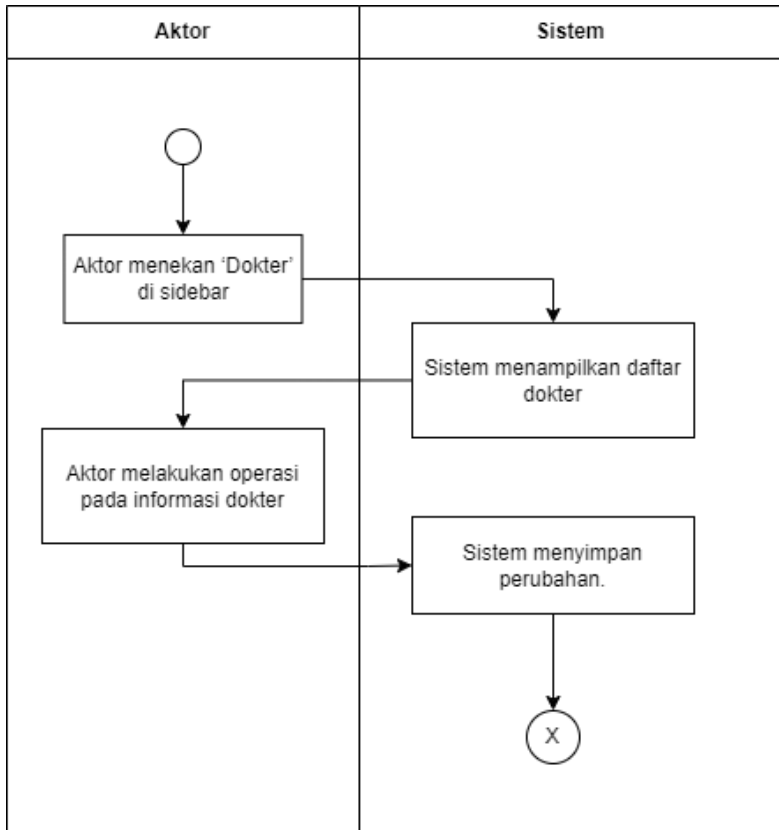
Gambar berikut merupakan diagram aktivitas dari *use case* Melihat Detail Pengguna (UC102).



Gambar 4.10 Diagram Aktivitas Melihat Detail Pengguna

h. Mengelola Informasi Dokter

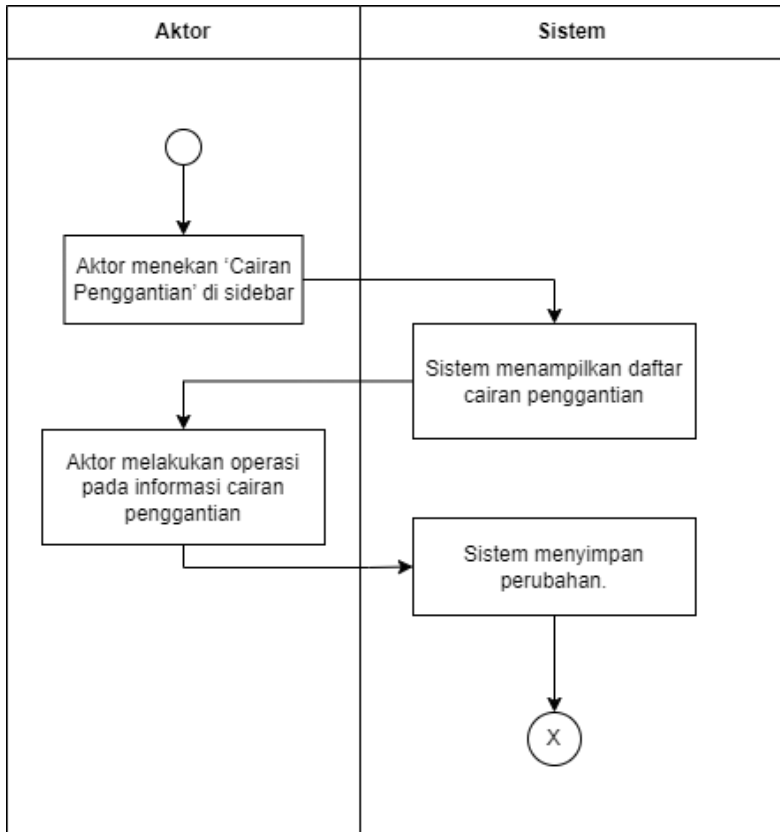
Gambar berikut merupakan diagram aktivitas dari *use case* Mengelola Informasi Dokter (UC103).



Gambar 4.11 Diagram Aktivitas Mengelola Informasi Dokter

i. Mengelola Informasi Cairan Penggantian

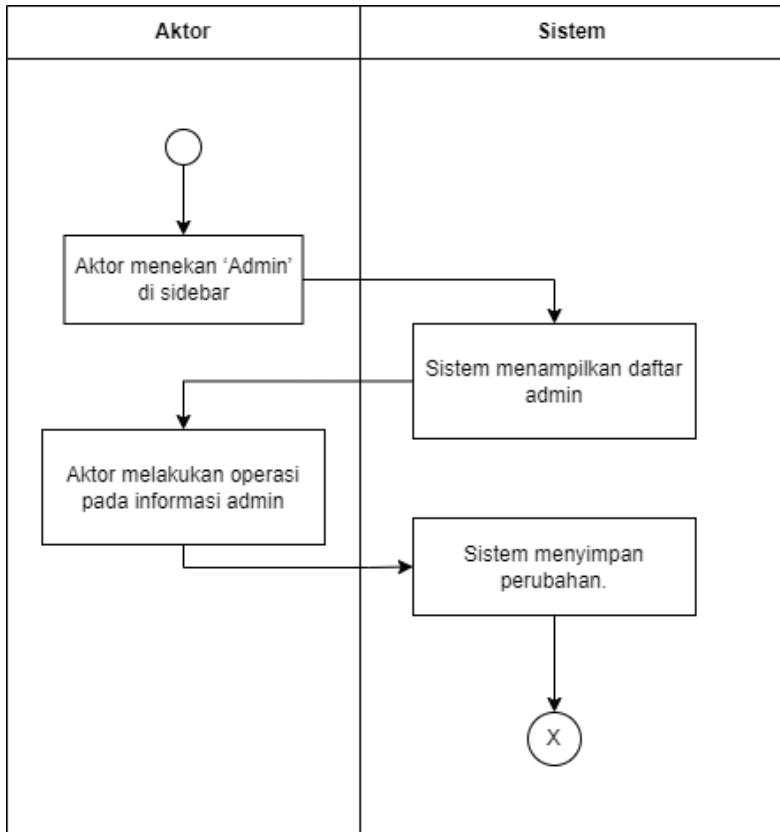
Gambar berikut merupakan diagram aktivitas dari *use case* Mengelola Informasi Cairan Penggantian (UC104).



Gambar 4.12 Diagram Aktivitas Mengelola Informasi Cairan Penggantian

j. Mengelola Informasi Admin

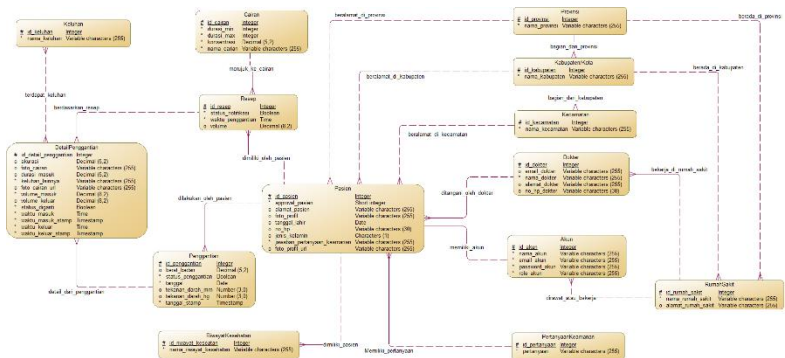
Gambar berikut merupakan diagram aktivitas dari *use case* Mengelola Informasi Admin (UC201).



Gambar 4.13 Diagram Aktivitas Mengelola Informasi Admin

#### 4.7. Conceptual Data Model

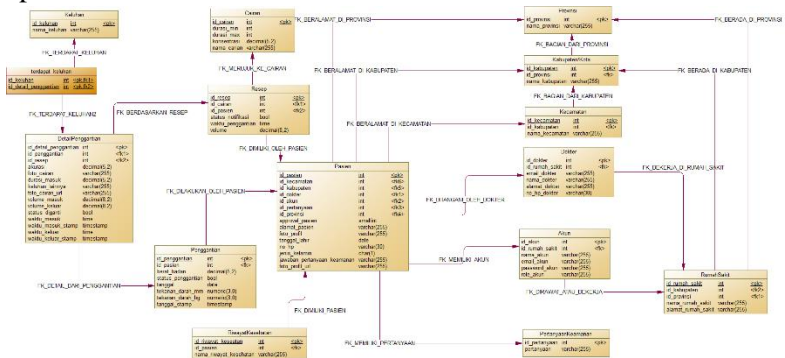
Gambar berikut adalah *conceptual data model* (CDM) dari aplikasi Sahabat CAPD.



Gambar 4.14 CDM Aplikasi Sahabat CAPD

4.8. Physical Data Model

Gambar berikut adalah physical data model (PDM) dari aplikasi Sahabat CAPD.



Gambar 4.15 PDM Aplikasi Sahabat CAPD

4.9. Deskripsi Data

Berikut adalah deskripsi dari masing-masing data yang digunakan pada aplikasi CAPD.

a. Akun Pengguna

Tabel ini adalah deskripsi data untuk tabel Akun Pengguna. Atribut role digunakan untuk menentukan tipe akun seperti pasien,

admin, atau super admin. Tabel ini menyimpan informasi tentang semua pengguna aplikasi, termasuk detail login, nama, email, dan peran pengguna.

*Tabel 4.15 Deskripsi Data pada Tabel Akun Pengguna Aplikasi CAPD*

Akun Pengguna		
No	Nama Atribut	Tipe Data
1.	id_akun	Integer
2.	nama_akun	Variable characters (255)
3.	email_akun	Variable characters (255)
4.	password_akun	Variable characters (255)
5.	role_akun	Variable characters (255)

b. Pasien

Tabel ini adalah deskripsi data untuk tabel Pasien. Pasien yang baru mendaftar akan memiliki nilai approval\_pasien 0 yang berarti menunggu verifikasi, jika terverifikasi oleh admin atau super admin, nilai berganti menjadi 1, jika ditolak, nilai diganti menjadi -1.

*Tabel 4.16 Deskripsi Data pada Tabel Pasien Aplikasi CAPD*

Pasien		
No	Nama Atribut	Tipe Data

1.	id_pasien	Integer
2.	approval_pasien	ShortInteger
3.	alamat_pasien	Variable characters (255)
4.	foto_profil	Variable characters (255)
5.	Tanggal_lahir	Date
6.	No_hp	Variable characters (30)
7.	Jenis_kelamin	Characters(1)
8.	Jawaban_pertanyaan_keamanan	Variable characters (255)
9.	Foto_profil_url	Variable characters (255)

c. Rumah Sakit

Tabel ini adalah deskripsi data untuk tabel Rumah Sakit. Tabel rumah sakit menyimpan informasi tentang rumah sakit yang bekerja sama, termasuk id, nama rumah sakit, dan alamat.

*Tabel 4.17 Deskripsi Data pada Tabel Rumah Sakit Aplikasi CAPD*

Rumah Sakit		
No	Nama Atribut	Tipe Data
1.	Id_rumah_sakit	Integer



2.	Nama_rumah_sakit	Variable characters (255)
3.	Alamat_rumah_sakit	Variable characters (255)

d. Dokter

Tabel Dokter adalah deskripsi data untuk tabel Dokter yang digunakan oleh pasien untuk memilih tenaga medis. Tabel ini berisi data tentang dokter yang bekerja di rumah sakit, termasuk nama, spesialisasi, dan informasi kontak.

*Tabel 4.18 Deskripsi Data pada Tabel Dokter Aplikasi CAPD*

Dokter		
No	Nama Atribut	Tipe Data
1.	Id_dokter	Integer
2.	Email_dokter	Variable characters (255)
3.	Nama_dokter	Variable characters (255)
4.	Alamat_dokter	Variable characters (255)
5.	No_hp_dokter	Variable characters (30)

e. Riwayat Kesehatan

Tabel ini adalah deskripsi data untuk tabel Riwayat Kesehatan. Pasien dapat mencatat riwayat kesehatan yang pernah dimilikinya.

*Tabel 4.19 Deskripsi Data pada Tabel Riwayat Kesehatan Aplikasi CAPD*

Riwayat Kesehatan		
No	Nama Atribut	Tipe Data
1.	Id_riwayat_kesehatan	Integer
2.	Nama_riwayat_kesehatan	Variable characters (255)

f. Cairan

Tabel ini adalah deskripsi data untuk tabel Cairan. Isi dari tabel ini hanya dapat dimodifikasi oleh super admin. Isi dari tabel ini adalah nama, durasi cairan dimasukkan, dan konsentrasi.

*Tabel 4.20 Deskripsi Data pada Tabel Cairan Aplikasi CAPD*

Cairan		
No	Nama Atribut	Tipe Data
1.	Id_cairan	Integer
2.	Durasi_min	Integer
3.	Durasi_max	Integer
4.	Konsentrasi	Decimal(5,2)
5.	Nama_cairan	Variable characters (255)

g. Resep

Tabel ini adalah deskripsi data untuk tabel Resep. Setiap pasien memiliki resep untuk melakukan penggantian cairan.

*Tabel 4.21 Deskripsi Data pada Tabel Resep Aplikasi CAPD*

Resep		
No	Nama Atribut	Tipe Data
1.	Id_resep	Integer
2.	Status_notifikasi	Boolean
3.	Waktu_penggantian	Time
4.	Volume	Decimal (8,2)

h. Penggantian

Tabel ini adalah deskripsi data untuk tabel Penggantian. Setiap penggantian, pasien dapat mencatat berat badan dan tekanan darah mereka.

*Tabel 4.22 Deskripsi Data pada Tabel Penggantian Aplikasi CAPD*

Penggantian		
No	Nama Atribut	Tipe Data
1.	Id_penggantian	Integer
2.	Berat_badan	Decimal (5,2)

3.	Status_penggantian	Boolean
4.	Tanggal	Date
5.	Tanggal_stamp	Timestamp
6.	Tekanan_darah_mm	Number (3,0)
7.	Tekanan_darah_hg	Number (3,0)

i. Detail Penggantian

Tabel ini adalah deskripsi data untuk tabel Detail Penggantian. Berisikan data-data yang lebih rinci untuk penggantian cairan yang dilakukan pasien.

*Tabel 4.23 Deskripsi Data pada Tabel Detail Penggantian Aplikasi CAPD*

Detail Penggantian		
No	Nama Atribut	Tipe Data
1.	Id_detail_penggantian	Integer
2.	Akurasi	Decimal (5,2)
3.	Foto_cairan	Variable characters (255)
4.	Durasi_masuk	Decimal (5,2)
5.	Keluhan_lainnya	Variable characters (255)
6.	Foto_cairan_url	Variable characters (255)
7.	Volume_masuk	Decimal (8, 2)

8.	Volume_keluar	Decimal (8, 2)
9.	Status_diganti	Boolean
10.	Waktu_masuk	Time
11.	Waktu_masuk_stamp	Timestamp
12.	Waktu_keluar	Time
13.	Waktu_keluar_stamp	Timestamp

j. Pertanyaan Keamanan

Tabel ini adalah deskripsi data untuk tabel Pertanyaan Keamanan. Tabel ini berisikan data pertanyaan yang sebelumnya sudah disepakati bersama oleh pihak pengembang sebagai pertanyaan keamanan untuk pengguna yang ingin mengganti *password* mereka jika pengguna lupa akan *password* yang dimiliki sebelumnya.

*Tabel 4.25 Deskripsi Data pada Tabel Pertanyaan Keamanan Aplikasi CAPD*

Pertanyaan Keamanan		
No	Nama Atribut	Tipe Data
1.	Id_pertanyaan	Integer
2.	Pertanyaan	Variable characters (255)

k. Provinsi

Tabel ini adalah deskripsi data untuk tabel Provinsi. Berisikan data nama-nama provinsi di Indonesia. Tabel ini menyimpan daftar semua provinsi yang ada dalam sistem dengan kolom id dan nama.

*Tabel 4.26 Deskripsi Data pada Tabel Provinsi Aplikasi CAPD*

Provinsi		
No	Nama Atribut	Tipe Data
1.	Id_provinsi	Integer
2.	Nama_provinsi	Variable characters (255)

1. Kabupaten/Kota

Tabel ini adalah deskripsi data untuk tabel Kabupaten/Kota. Tabel ini mencatat semua kota di dalam setiap provinsi, termasuk id dan nama.

*Tabel 4.27 Deskripsi Data pada Tabel Kabupaten/Kota Aplikasi CAPD*

Kabupaten/Kota		
No	Nama Atribut	Tipe Data
1.	Id_kabupaten	Integer
2.	Nama_kabupaten	Variable characters (255)

m. Kecamatan

Tabel ini adalah deskripsi data untuk tabel Kecamatan. Tabel ini berisi informasi tentang semua kecamatan dalam setiap kota yang mencakup informasi id dan nama.

*Tabel 4.28 Deskripsi Data pada Tabel Kecamatan Aplikasi CAPD*

Kecamatan		
No	Nama Atribut	Tipe Data
1.	Id_kecamatan	Integer
2.	Nama_kecamatan	Variable characters (255)

*[Halaman ini sengaja dikosongkan]*



## **BAB 5**

### **IMPLEMENTASI SISTEM**

Bab ini membahas tentang hasil desain UI/UX serta implementasi Backend dari aplikasi Sahabat CAPD.

#### **5.1. Implementasi Design System**

Sistem yang dibuat merupakan sistem untuk melakukan pencatatan penggantian cairan CAPD yang ditujukan untuk pasien dan keluarga pasien CAPD. Selain itu, admin yang ditunjuk dari pihak rumah sakit akan memiliki akses kepada dashboard admin. Dashboard ini akan memungkinkan pengelolaan data pencatatan pasien sehingga dapat dipantau secara jarak jauh. Fungsionalitas aplikasi ini dari sisi pengguna secara umum mencakupi penginputan resep dan pencatatan penggantian. Dari sisi admin, fungsionalitas bersifat manajerial seperti verifikasi pengguna, pengelolaan data dokter dan cairan, serta pengelolaan admin rumah sakit (khusus untuk Superadmin).

Aplikasi ini dibuat dengan backend Laravel dan DBMS MySQL. Desain tampilan dirancang dengan memanfaatkan perangkat lunak Figma.

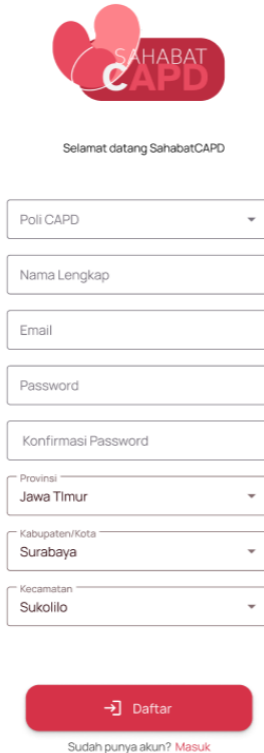
#### **5.2. Desain Tampilan Antarmuka Pengguna**

Berikut adalah desain antarmuka pengguna baik untuk pasien maupun admin untuk aplikasi Sahabat CAPD.

##### **5.2.1. Antarmuka Pengguna**

Berikut adalah desain antarmuka aplikasi Sahabat CAPD untuk pasien.

a. Register



Selamat datang SahabatCAPD

Poli CAPD

Nama Lengkap

Email

Password

Konfirmasi Password

Provinsi  
Jawa Timur

Kabupaten/Kota  
Surabaya

Kecamatan  
Sukolilo

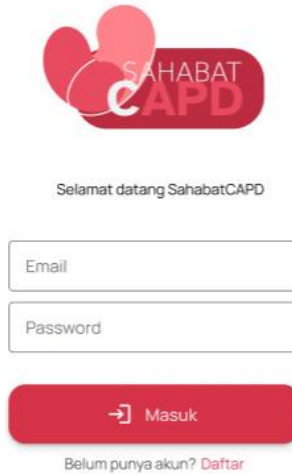
Daftar

Sudah punya akun? [Masuk](#)

*Gambar 5.1 Tampilan Halaman Pendaftaran*

Gambar 5.1 menggambarkan tampilan formulir untuk pengguna yang ingin mendaftarkan akun pada sistem Sahabat CAPD. Formulir pendaftaran memuat informasi yang diperlukan untuk menginisialisasi akun pengguna. Informasi tersebut meliputi poli CAPD, nama, surel, kata sandi, dan informasi alamat pengguna.

b. Login



*Gambar 5.2 Tampilan Halaman Login*

Gambar 5.2 menggambarkan desain halaman login bagi pengguna. Pengguna yang telah mendaftar dan diverifikasi admin akan masuk ke aplikasi lewat halaman tersebut.

c. Beranda



*Gambar 5.13 Tampilan Halaman Beranda*

Gambar 5.3 menggambarkan tampilan halaman beranda yang menjadi halaman utama bagi pengguna aplikasi Sahabat CAPD. Pada halaman ini tersedia berbagai opsi navigasi, antara lain ke halaman Logbook, Resep, Riwayat, Kesehatan, Statistik, Profil, dan Pengaturan. Selain itu, terdapat tombol yang melayang untuk mengakses Chatbot.

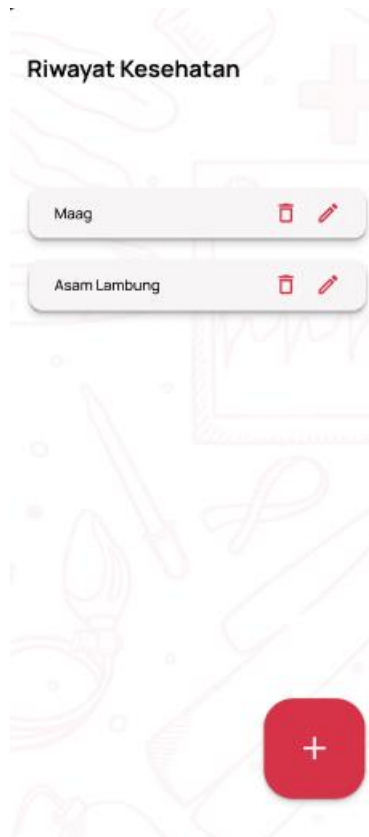
#### d. Resep Penggantian



*Gambar 5.4 Tampilan Daftar Resep*

Gambar 5.4 menggambarkan tampilan halaman yang menampilkan informasi resep pengguna. Setiap penggantian memuat informasi tentang jenis cairan, waktu penggantian, dan durasi cairan tersebut sebelum dikeluarkan. Pengguna dapat menghapus dan mengedit informasi penggantian melalui tombol yang berada di bagian atas kartu. Selain itu, pengguna dapat menekan tombol tambah untuk menambahkan resep baru.

e. Riwayat Kesehatan



*Gambar 5.5 Tampilan Daftar Riwayat Kesehatan*

Gambar 5.5 menggambarkan tampilan halaman Riwayat Kesehatan yang menampilkan riwayat kesehatan pengguna. Pengguna dapat menghapus dan mengedit informasi penggantian melalui tombol yang berada di bagian kanan kartu. Selain itu, pengguna dapat menekan tombol tambah untuk menambahkan resep baru.

f. Logbook



Gambar 5.6 Tampilan Logbook

Gambar 5.6 menggambarkan tampilan Logbook. Halaman pertama merupakan sebuah kalender. Setiap tanggal diberi pewarnaan sesuai dengan status pengisian. Sebuah tanggal akan berwarna putih jika belum terisi, pink jika sudah terisi, dan ber-outline merah jika merupakan tanggal yang sedang berjalan.

Halaman kedua merupakan logbook harian yang berisi penggantian cairan setiap hari sesuai dengan data yang dimasukkan

pada halaman Resep Penggantian.

g. Pencatatan

**Catatan Penggantian**

**Cairan Masuk**

Jenis Cairan  
Dextrose 7,5

Volume Masuk (mL)  
1000

Waktu Masuk  
16:00

Durasi Masuk (Menit)  
05:00

**Cairan Keluar**

Volume Keluar (mL)  
1000

Waktu Keluar  
21:00

Ambil Foto Cairan Buangan

**Keluhan**

Bengkak Kaki

Exit Site Nyeri

Checkbox

Keluhan Lain

*Gambar 5.7 Tampilan Formulir Pencatatan Penggantian*

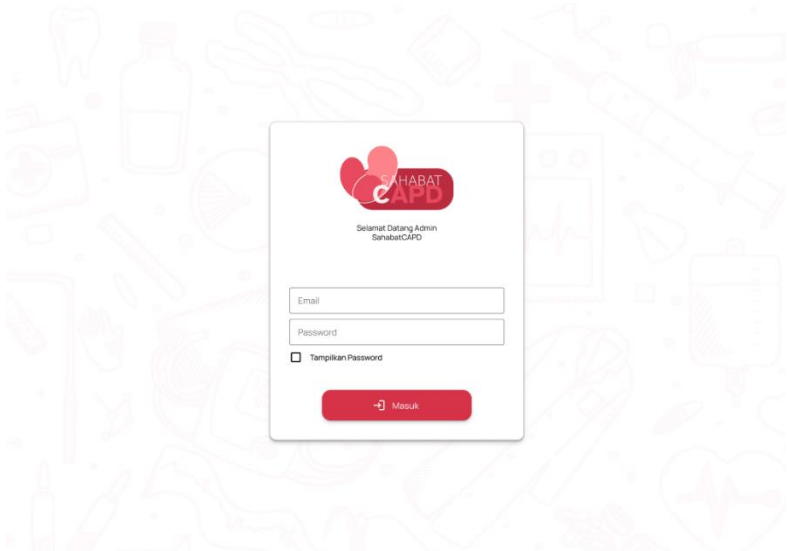
Gambar 5.7 menggambarkan tampilan formulir pencatatan penggantian. Pada halaman ini, pengguna dapat memasukkan informasi mengenai penggantian cairan seperti volume, waktu, dan durasi. Selain itu, pengguna dapat melampirkan foto cairan. Foto cairan ini akan dikirim untuk dinilai oleh model kecerdasan buatan yang dapat mengklasifikasi kualitas cairan CAPD



### 5.2.2. Antarmuka Admin

Berikut adalah desain antarmuka untuk admin pada aplikasi Sahabat CAPD.

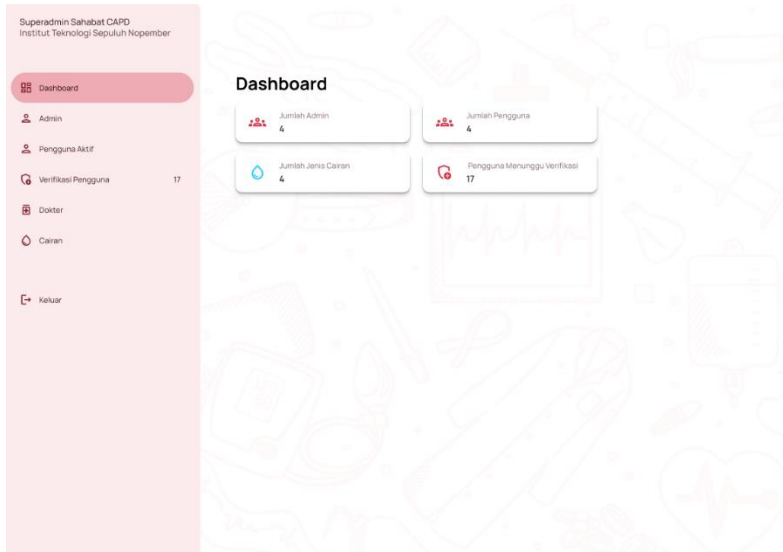
#### a. Login



*Gambar 5.8 Tampilan Login Admin*

Gambar 5.8 menggambarkan tampilan login bagi admin. Tidak terdapat halaman pendaftaran untuk admin karena akun admin disediakan dari sisi superadmin. Akun admin yang telah dibuatkan akan masuk ke dashboard admin melalui halaman ini.

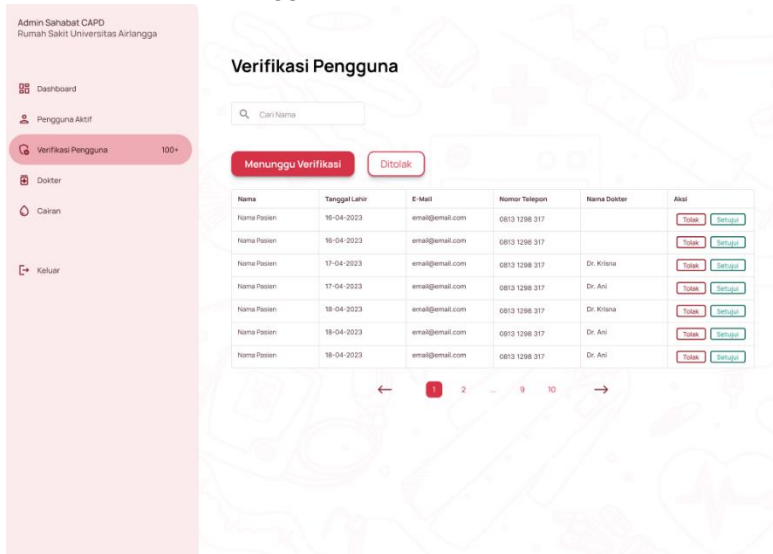
## b. Beranda



*Gambar 5.9 Tampilan Dashboard Admin*

Gambar 5.9 menggambarkan tampilan dashboard admin. Pada halaman ini, admin dapat melihat informasi umum, antara lain jumlah pengguna yang menunggu verifikasi, jumlah pengguna aktif, jumlah jenis cairan, dan jumlah admin (khusus untuk user superadmin). Pada sisi dashboard terdapat sidebar yang digunakan untuk navigasi antar halaman di aplikasi

### c. Verifikasi Pengguna



Gambar 5.10 Tampilan Halaman Verifikasi Pengguna

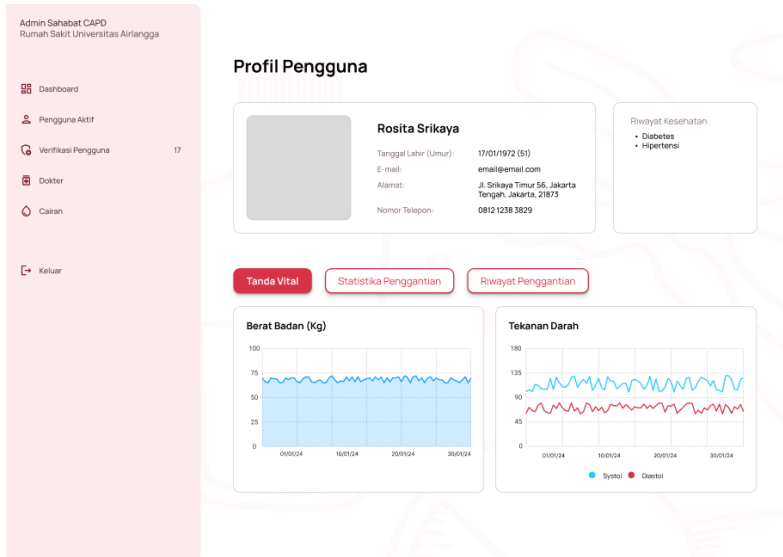
Gambar 5.10 menggambarkan tampilan Verifikasi Pengguna. Pada halaman ini, admin dapat melihat pengguna yang telah melakukan pendaftaran akun. Halaman ini penting untuk memastikan tidak sembarang pengguna yang dapat menggunakan aplikasi. Admin dapat menyetujui atau menolak pendaftaran seorang pengguna dengan menekan tombol pada kolom terakhir di grid.

#### d. Data Pengguna Aktif

Nama	Tanggal Lahir	E-Mail	Nomor Telepon	Rumah Sakit	Aksi
Nama Pasien	19-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	19-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	17-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	17-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	18-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	18-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	18-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>
Nama Pasien	18-04-2023	email@email.com	0813 1298 317	RSU	<a href="#">Detail</a>

*Gambar 5.11 Tampilan Daftar Pengguna Aktif*

Gambar 5.11 menggambarkan tampilan daftar pengguna aktif. Pada halaman ini, admin dapat melakukan pencarian untuk mencari seorang pengguna. Lalu, admin dapat menekan tombol detail untuk melihat biodata dan detail pencatatan pengguna.



*Gambar 5.12 Tampilan Detail Pengguna Aktif*

Gambar 5.12 menggambarkan tampilan detail pengguna. Pada halaman ini, biodata pengguna serta informasi hasil pencatatan dapat dijangkau. Informasi seperti tanda vital, statistika penggantian, dan riwayat penggantian tersedia dengan tujuan memantau kondisi pengguna.

## e. Data Dokter

Admin Sahabat CAPD  
Rumah Sakit Universitas Airlangga

Dashboard  
Pengguna Aktif  
Verifikasi Pengguna 17  
**Dokter**  
Cairan  
Keluar

### Daftar Dokter

Cari Nama + Tambah Data Dokter

Nama	E-Mail	Nomor Telepon	Alamat	Aksi
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>
Nama Dokter	email@gmail.com	0813 1298 317	Jalan Ruman No. 1	<span>Hapus</span> <span>Edit</span>

← 1 2 ... 9 10 →

*Gambar 5.13 Tampilan Daftar Dokter*

Gambar 5.13 menggambarkan tampilan daftar dokter. Setiap baris memberikan informasi dokter berupa nama, surel, nomor telepon, dan Alamat. Admin juga dapat mengelola informasi melalui 76olo maksi di kanan grid. Tersedia tombol untuk menghapus dan mengedit informasi dokter. Pilihan tenaga medis yang tersedia di sisi pengguna akan bergantung pada daftar dokter di halaman ini.

## f. Data Cairan

Nama	Konsentrasi	Durasi Minimal	Durasi Maksimal	Aksi
Destrope	1.5	10	100	<a href="#">Hapus</a> <a href="#">Edit</a>
Destrope	2.5	10	100	<a href="#">Hapus</a> <a href="#">Edit</a>
Destrope	7.5	10	100	<a href="#">Hapus</a> <a href="#">Edit</a>

*Gambar 5.14 Tampilan Daftar Jenis Cairan Penggantian*

Gambar 5.14 menggambarkan tampilan daftar jenis cairan penggantian. Setiap baris memberikan informasi cairan penggantian berupa nama, konsentrasi, durasi minimal, dan durasi maksimal. Admin juga dapat mengelola informasi melalui kolom aksi di kanan grid. Pengguna aplikasi hanya dapat mencatat resep yang telah tercatat oleh admin dalam halaman ini.

## g. Data Admin

The screenshot displays the 'Daftar Admin' interface. On the left is a sidebar with a pink background containing navigation links: Dashboard, Admin (highlighted), Pengguna Aktif, Verifikasi Pengguna (17), Dokter, Cairan, and Keluar. The main content area is titled 'Daftar Admin' and features a search bar labeled 'Cari Nama' and a '+ Tambah Admin' button. Below this is a table with the following data:

Nama	E-Mail	Rumah Sakit	Password	Aksi
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>
Nama Admin	email@email.com	RSU	*****	<a href="#">Hapus</a> <a href="#">Edit</a>

At the bottom of the table, there is a pagination control showing a left arrow, a red square with '1', a right arrow, and numbers '2', '9', and '10'.

Gambar 5.15 Tampilan Daftar Admin

Gambar 5.15 menggambarkan tampilan daftar admin. Setiap baris memberikan informasi admin berupa nama, surel, rumah sakit, dan password. Halaman ini hanya tersedia bagi superadmin. Superadmin dapat mengelola informasi melalui kolom aksi di kanan grid. Tersedia tombol untuk menghapus dan mengedit informasi admin. Pengguna hanya dapat mendaftar ke dalam sebuah poli CAPD apabila terdaftar seorang admin untuk poli tersebut.

### 5.3. Implementasi Source Code Backend

Backend dari aplikasi ini dibuat dengan menggunakan Laravel. Laravel pada aplikasi ini digunakan untuk berkomunikasi dengan database SQL Server dan menyediakan API untuk frontend melakukan proses bisnis aplikasi. Dalam implementasinya, dibuat *migrations* untuk mendefinisikan skema basis data aplikasi,



*seeders* untuk menambah data ke dalam tabel pada basis data, *Eloquent models* untuk berinteraksi dengan tabel pada basis data, termasuk membuat, membaca, menyunting, dan menghapus data pada tabel dalam basis data, *controllers* untuk mengatur logika pada proses bisnis aplikasi, dan *filters* untuk membantu proses filter dalam pengambilan data pada basis data.

Digunakan pula beberapa *package* dari Laravel untuk membantu proses bisnis aplikasi seperti Laravel Sanctum untuk autentikasi pengguna dan penggunaan token, DPTSI Laravel Storage untuk akses penyimpanan file milik ITS menggunakan Laravel, dan Laravel Excel untuk mengolah data excel yang digunakan untuk membuat laporan penggantian.

Selanjutnya, akan dijelaskan lebih rinci mengenai *migrations*, *seeders*, *Eloquent models*, *controllers*, dan *filters* yang dibuat menggunakan Laravel.

### **5.3.1. *Migrations***

*Migration* pada Laravel digunakan untuk mendefinisikan skema basis data aplikasi. Pada aplikasi ini dibuat beberapa *migrations* untuk mendefinisikan tabel pada basis data. Selain itu terdapat *migrations* untuk melakukan perubahan tipe data maupun penambahan kolom pada tabel dalam basis data seiring perkembangan aplikasi. Keseluruhan *migrations* yang dibuat untuk aplikasi ini dapat dilihat pada gambar 5.1.

```
▼ migrations
  2013_10_11_070640_create_rumah_sakits_table.php
  2014_10_12_000000_create_users_table.php
  2014_10_12_100000_create_password_reset_tokens_table.php
  2019_08_19_000000_create_failed_jobs_table.php
  2019_12_14_000001_create_personal_access_tokens_table.php
  2023_01_17_093153_create_pertanyaan_keamanans_table.php
  2023_10_25_070644_create_dokters_table.php
  2023_10_25_070644_create_provinsis_table.php
  2023_10_25_070645_create_kabupatens_table.php
  2023_10_25_070646_create_kecamatanans_table.php
  2023_10_25_070713_create_pasienans_table.php
  2023_10_25_070807_create_penggantians_table.php
  2023_10_25_071330_create_cairans_table.php
  2023_10_25_071428_create_reseps_table.php
  2023_10_25_072356_create_riwat_kesehatans_table.php
  2023_10_25_072519_create_detail_penggantians_table.php
  2024_01_16_093541_create_keluhans_table.php
  2024_01_17_072141_add_tanggal_stamp_to_penggantian_table.php
  2024_01_17_093153_create_keluhan_detail_penggantian_table.php
  2024_02_05_082932_add_kabupaten_id_to_rumah_sakits_table.php
  2024_02_05_082932_add_provinsi_id_to_rumah_sakits_table.php
  2024_02_16_132624_change_approval_data_type_in_pasienans_table.php
  2024_02_29_033435_add_column_foto_profil_url_to_table.php
```

Gambar 5.16 Daftar Migrasi pada aplikasi

Tabel `user`, `personal_access_tokens` dibuat dari Laravel dan *package* Laravel Sanctum. Tabel `user` ini adalah implementasi tabel akun pengguna. Kemudian untuk tabel lainnya merupakan implementasi dari tabel itu sendiri sesuai dengan namanya. Beberapa tabel mengalami perubahan hingga menjadi versi terakhir, sesuai dengan apa yang tertera pada bagian 4.8 mengenai *Physical Data Model*.

### 5.3.2. Seeders

*Seeder* pada aplikasi ini digunakan untuk menambahkan data ke dalam tabel pada basis data yang bukan berasal dari pengguna, seperti data provinsi, kabupaten, dan kecamatan, data pertanyaan keamanan, data rumah sakit, dan data cairan.

```

1  <?php
2  namespace Database\Seeders;
3  use DB;
4  use Illuminate\Database\Seeder;
5
6  class CairanSeeder extends Seeder
7  {
8      public function run(): void
9      {
10         // Cairan are:
11         // Dextrose, 7.5%
12         // Dextrose 2,5%
13         // Dextrose 1,5%
14
15         // insert manually to table cairans
16         DB::table('cairans')->insert([
17             ['nama_cairan' => 'Dextrose',
18              'konsentrasi' => 7.5,
19              'durasi_min' => 5*60,
20              'durasi_max' => 6*60],
21             ['nama_cairan' => 'Dextrose',
22              'konsentrasi' => 2.5,
23              'durasi_min' => 5*60,
24              'durasi_max' => 6*60],
25             ['nama_cairan' => 'Dextrose',
26              'konsentrasi' => 1.5,
27              'durasi_min' => 5*60,
28              'durasi_max' => 6*60]
29             // Add more data as needed
30         ]);
31     }
32 }
33

```

Gambar 5.17 Contoh Seeder untuk Tabel Cairan

### 5.3.3. Eloquent models

Dibentuk model untuk tabel cairan, detail penggantian, dokter, provinsi, kabupaten, kecamatan, keluhan, pasien, penggantian, pertanyaan keamanan, resep, riwayat kesehatan dan rumah sakit untuk berinteraksi dengan tabel masing-masing. Pada tiap model didefinisikan juga relasi yang dimiliki sesuai dengan yang tertera pada *Physical Data Model*.

```

1  <?php
2  namespace App\Models;
3  use Illuminate\Database\Eloquent\Factories\HasFactory;
4  use Illuminate\Database\Eloquent\Model;
5
6  class Pasien extends Model
7  {
8      use HasFactory;
9  > protected $fillable = [ ...
24 > ];
25 // tiap Pasien punya dokter
26 public function dokter()
27 > { ...
29 > }
30 // tiap Pasien punya banyak riwayat kesehatan
31 public function riwayatKesehatans()
32 > { ...
34 > }
35 // tiap Pasien punya banyak resep
36 public function reseps()
37 > { ...
39 > }
40 // tiap Pasien punya banyak penggantian
41 public function penggantian()
42 > { ...
44 > }
45 // tiap Pasien punya 1 akun/user
46 public function user()
47 > { ...
49 > }
50 }
51

```

*Gambar 5.18 Contoh Model untuk Tabel Pasien*

### 5.3.4. *Controllers*

*Controller* pada Laravel digunakan untuk mengatur logika pada proses bisnis aplikasi. Pada aplikasi ini dibentuk beberapa *controller* diantaranya:

- a. RegisteredUserController

*Controller* ini digunakan untuk mengatur logika pendaftaran akun pasien pada aplikasi Sahabat CAPD. Berisikan satu fungsi *store* yang menerima request data pendaftaran, yang kemudian akan divalidasi berdasarkan beberapa aturan sederhana untuk setiap fields pendaftaran seperti nama, email, password, id rumah

sakit, jenis kelamin, id provinsi, id kabupaten, id kecamatan, id pertanyaan keamanan, dan jawaban pertanyaan keamanan. Aturan validasi yang diterapkan seperti untuk field nama berupa string dengan panjang maksimal sebanyak 255 karakter, email berupa email yang valid dan unik, yang berarti belum pernah digunakan sebelumnya, password minimal delapan karakter dan harus mengandung setidaknya huruf dan angka, id rumah sakit harus merupakan id rumah sakit yang ada pada basis data, jenis kelamin harus berupa string, id provinsi merupakan id provinsi yang ada pada basis data, begitu pula seterusnya untuk id kabupaten dan kecamatan, tetapi untuk id kecamatan bersifat opsional, kemudian id pertanyaan keamanan harus merupakan id yang valid pada basis data, dan terakhir jawaban pertanyaan keamanan berupa string dengan panjang maksimal sebanyak 255 karakter.

Jika data yang diterima valid, maka pada basis data akan ditambahkan 1 baris baru pada tabel user dan pasien dengan data sesuai tabel. Untuk password dan jawaban pertanyaan keamanan akan dihash sebelum dimasukkan ke dalam basis data. Setelah menambahkan data ke dalam basis data, aplikasi akan membuat token untuk user, yang akan digunakan untuk melakukan request api pada aplikasi Sahabat CAPD. Jika proses pendaftaran berhasil, maka sistem akan mengembalikan data user dan token yang baru saja dibuat. Berikut adalah kode semu untuk `RegisteredUserController`.

```
1 class RegisteredUserController extends Controller{
2     public function store(Request $request){
3         $request->validate([
...     ...
```

*Kode Semu 5.1 RegisteredUserController Bagian 1*

```

21         ]);
22         $user = User::create([
...     ...
29         ]);
30         $jawabanPertanyaanKeamanan = $request-
>jawabanPertanyaanKeamanan;
31         $jawabanPertanyaanKeamanan =
strtolower($jawabanPertanyaanKeamanan);
32         $jawabanPertanyaanKeamanan = str_replace('
', '', $jawabanPertanyaanKeamanan);

33         $pasien = Pasien::create([
...     ...
45         ]);
46         $apiToken = $user->createToken('apiToken',
['pasien']);

...     ...
51         return response()->json(
...     ...
57         );
58     }
59 }

```

*Kode Semu 5.2 RegisteredUserController Bagian 2*

#### b. AuthenticatedSessionController

*Controller* ini digunakan untuk mengatur logika login dan logout pada Sahabat CAPD. Dimana untuk login akan dilakukan pada fungsi store, *controller* akan memvalidasi email dan password yang diterima, jika pasangan email dan password cocok dengan yang ada pada basis data maka proses login dilanjutkan dengan pengecekan *approval* untuk user yang merupakan pasien, dan jika pasien belum mendapatkan *approval* maka proses login gagal, dan

jika berhasil, sistem akan memberi respon yang berisi token kepada user dengan kode status 201. Untuk logout, dilakukan pada fungsi `destroy`, *controller* akan menghapus data token user yang meminta logout dan memberikan respon status 200. Berikut merupakan kode semu untuk `AuthenticatedSessionController`.

```
1 class AuthenticatedSessionController
  extends Controller{
2   public function store(Request $request){
3     $data = $request->validate([
...   ...
6     ]);
7     $user = User::where('email',
  $data['email']->first());
8     if (!$user ||
  !Hash::check($data['password'], $user->password)) {
...   ...
14    }
15    if($user->role == 'pasien'){
...   ...
21    }
22    $apiToken = $user->createToken('apiToken',
  [$user->role]);
23    $user->load('pasien')
24    return response()->json(
...   ...
33    );
34  }
```

*Kode Semu 5.3 AuthenticatedSessionController Bagian 1*

```

... ..
38     public function destroy(Request $request){
39         auth()->user()->tokens()->delete();
40         return response()->json(
... ..
46         );
47     }
48 }

```

*Kode Semu 5. 4 AuthenticatedSessionController Bagian 2*

### c. UserController

*Controller* ini berisi logika untuk proses-proses yang berkaitan dengan model User. Diantaranya seperti fungsi index untuk mengquery data user pada basis data sesuai dengan kapasitas role user. Data akan difilter terlebih dahulu pada fungsi prepareIndex sesuai dengan role user, seperti role pasien hanya bisa melihat dirinya sendiri, super admin dapat melihat semua user, dan admin hanya dapat melihat user yang ada pada rumah sakit admin tersebut. Query dapat dilakukan dengan filter nama, email, role, dan id rumah sakit. Hasil query dapat diurutkan berdasarkan nama dan dilakukan paging (penomoran halaman). Fungsi show digunakan untuk melihat data satu user beserta relasi yang dimiliki user jika diperlukan seperti data pasien dan rumah sakit. *Controller* ini juga dapat memperbaharui data user untuk field nama, email, dan id rumah sakit menggunakan fungsi update, yang mana data pada request telah divalidasi sebelumnya, sesuai pada saat proses pendaftaran. Fungsi gantiPassword dan lupaPassword digunakan mengubah password user, untuk gantiPassword, diperlukan data password lama user, dan lupaPassword digunakan untuk mengganti password jika user lupa dengan password yang



dimilikinya, dimana untuk memvalidasi bahwa user tersebut yang meminta penggantian password, user perlu menjawab pertanyaan keamanan yang sebelumnya telah diisi pada saat proses pendaftaran. Berikut adalah kode semu untuk UserController.

```
1  class UserController extends Controller{
2      private function prepareIndex(User $user){
...  ...
19  }
20  public function index(Request $request){
...  ...
45  }
46  public function show(User $user){
...  ...
55  }
56  public function update(UpdateUserRequest
    $request, User $user){
...  ...
83  }
...  ...
95  public function gantiPassword(Request $request,
    User $userInQuestion){
...  ...
120  }
121  public function lupaPassword(Request $request){
...  ...
159  }
160  }
```

*Kode Semu 5.5 UserController*

#### d. PasienController

*Controller* ini berisi logika untuk proses-proses yang berkaitan dengan model Pasien. Diantaranya seperti fungsi `index` untuk mengquery data pasien pada basis data sesuai dengan kapasitas role user. Data akan difilter terlebih dahulu pada fungsi `prepareIndex` sesuai dengan role user, seperti role pasien hanya bisa melihat dirinya sendiri, super admin dapat melihat semua data pasien, dan admin hanya dapat melihat pasien yang berkaitan pada rumah sakit admin tersebut. Hasil query dapat diurutkan berdasarkan nama dan dilakukan paging (penomoran halaman). *Controller* ini juga mengatur proses untuk memperbaharui data pasien menggunakan fungsi `update` dan `updateFotoProfil`. Fungsi `update` digunakan untuk menyunting data pasien berupa alamat, tanggal lahir, nomor hp, id dokter, jenis kelamin, id provinsi, id kabupaten, id kecamatan, dan id rumah sakit. *Approval* pasien juga dapat diubah menggunakan fungsi ini, hanya saja user yang melakukan request harus memiliki role sebagai admin ataupun super admin. Data yang diterima pada fungsi `update` telah divalidasi sebelumnya saat pendaftaran seperti alamat berupa string dengan panjang maksimal 255 karakter, tanggal lahir harus dalam format “yyyy-mm-dd” yaitu tahun, kemudian diikuti oleh bulan dan kemudian hari, nomor hp memenuhi regex “/^(\\+\\d{1,5}[- ]?)?\\d+\$/” dan minimal delapan karakter dan maksimal 30 karakter, kemudian untuk id dokter, id rumah sakit, id provinsi, id kabupaten, dan id kecamatan harus berupa id yang valid pada basis data, dan jenis kelamin berupa string. Fungsi `updateFotoProfil` digunakan untuk mengubah foto profil, dimana untuk penyimpanan foto menggunakan Laravel Storage milik DPTSI ITS pada fungsi `saveImage`. Selain itu *controller ini* menjadi penghubung api untuk chatbot pada fungsi `permisiChatBot` yang meneruskan request dari user ke api chatbot aplikasi sebelumnya dan mengembalikan hasil dari api tersebut ke

user. Berikut adalah kode semu untuk PasienController.

```
1 class PasienController extends Controller{
2     private function prepareIndex(User $user){
... ..
20     }
21     public function saveImage($image){
... ..
49     }
50     public function index(Request $request){
... ..
106    }
107    public function show(Pasien $pasien){
... ..
114    }
115    public function updateFotoProfil(Request
    $request, Pasien $pasien){
... ..
168    }
169    public function update(UpdatePasienRequest
    $request, Pasien $pasien){
... ..
254    }
255    public function permisiChatBot(Request
    $request){
... ..
260    }
261 }
```

*Kode Semu 5.6 PasienController*

e. *PertanyaanKeamananController*

*Controller* ini hanya digunakan untuk melakukan query pertanyaan keamanan yang sebelumnya telah diisi ke dalam basis

data. Fungsi `index` digunakan untuk mengembalikan seluruh pertanyaan keamanan pada tabel, dan fungsi `show` digunakan untuk menampilkan satu data pertanyaan keamanan. Berikut ini merupakan kode semu untuk `PertanyaanKeamananController`.

```
1  class PertanyaanKeamanan extends Controller{
2      public function index(){
...  ...
6      }
7      public function show(PertanyaanKeamanan
      $pertanyaanKeamanan){
...  ...
10     }
11 }
```

*Kode Semu 5.7 PertanyaanKeamananController*

f. `DokterController`

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan model `Dokter`. Diantaranya seperti fungsi `index` untuk mengquery data dokter pada basis data. Data akan difilter terlebih dahulu pada fungsi `prepareIndex` sesuai dengan role user, seperti role pasien dapat melihat data seluruh dokter, super admin dapat melihat semua data dokter, dan admin hanya dapat melihat dokter yang berkaitan pada rumah sakit admin tersebut. Hasil query dapat diurutkan berdasarkan nama dan dilakukan paging (penomoran halaman). *Controller* ini juga mengatur proses untuk memperbaharui data dokter pada fungsi `update`, dimana hanya admin dan super admin yang dapat mengganti data dokter. Fungsi `store` digunakan untuk menambahkan data dokter, dimana admin hanya dapat menambah dokter untuk rumah sakitnya, dan super admin dapat menambah dokter untuk semua rumah sakit. Data yang diterima telah

divalidasi juga sebelumnya. Fungsi destroy digunakan untuk penghapusan data dokter, admin hanya dapat menghapus dokter pada rumah sakitnya dan super admin dapat menghapus semua dokter. Berikut adalah kode semu untuk DokterController.

```
1 class DokterController extends Controller{
2     private function prepareIndex(User $user){
... ..
20     }
22     public function index(Request $request){
... ..
63     }
64     public function store(StoreDokterRequest
    $request){
... ..
99     }
100    public function show(Dokter $dokter){
... ..
109    }
110    public function update(UpdateDokterRequest
    $request, Dokter $dokter){
... ..
124    }
125    public function destroy(Dokter $dokter){
... ..
130    }
131 }
```

*Kode Semu 5.8 DokterController*

g. RumahSakitController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan model RumahSakit. Diantaranya seperti fungsi index untuk mengquery data rumah sakit pada basis

data. Data akan difilter terlebih dahulu pada fungsi `prepareIndex`, yaitu menghilangkan satu data *dummy* rumah sakit super admin, karena super admin memerlukan rumah sakit, jadi dibuatkan data *dummy* sebagai rumah sakit dari super admin. Hasil query dapat diurutkan berdasarkan nama dan dilakukan paging (penomoran halaman). *Controller* ini juga mengatur proses untuk memperbaharui data rumah sakit menggunakan fungsi `update`. Fungsi `store` digunakan untuk menambahkan data rumah sakit. Fungsi `destroy` digunakan untuk penghapusan data rumah sakit. Semua fungsi penambahan, penyuntingan, dan penghapusan rumah sakit hanya dapat dilakukan oleh super admin. Berikut adalah kode semu `RumahSakitController`.

```
1 class RumahSakitController extends Controller{
2     private function prepareIndex(){
... ..
8     }
9     public function index(Request $request){
... ..
35    }
36    public function store(StoreRumahSakitRequest
    $request){
... ..
46    }
47    public function show(RumahSakit $rumahSakit){
... ..
54    }
55    public function update(UpdateRumahSakitRequest
    $request, RumahSakit $rumahSakit){
... ..
```

*Kode Semu 5.9 Rumah Sakit Controller Bagian 1*

```

60     }
61     public function destroy(RumahSakit
        $rumahSakit){
...     ...
66     }
67 }

```

*Kode Semu 5.10 RumahSakitController Bagian 2*

#### h. ProvinsiController

*Controller* ini hanya digunakan untuk melakukan query data provinsi yang sebelumnya telah ditambahkan ke dalam basis data. Fungsi `index` digunakan untuk mengembalikan seluruh data provinsi pada tabel, data yang diambil dapat diurutkan berdasarkan nama. Fungsi `show` digunakan untuk menampilkan satu data provinsi. Berikut merupakan kode semu untuk `ProvinsiController`.

```

1     class ProvinsiController extends Controller{
2         public function index(Request $request){
...     ...
17     }
18     public function show(Provinsi $provinsi){
...     ...
22     }
23 }

```

*Kode Semu 5.11 ProvinsiController*

#### i. KabupatenController

*Controller* ini hanya digunakan untuk melakukan query data kabupaten/kota yang sebelumnya telah ditambahkan ke dalam basis data. Fungsi `index` digunakan untuk mengembalikan seluruh data kabupaten/kota pada tabel dan dapat diurutkan berdasarkan nama. Fungsi `show` digunakan untuk menampilkan satu data

kabupaten/kota. Berikut adalah kode semu untuk KabupatenController.

```
1 class KabupatenController extends Controller{
2     public function index(Request $request){
... ..
21     }
22     public function show(Kabupaten $kabupaten){
... ..
26     }
27 }
```

*Kode Semu 5.12 KabupatenController*

j. KecamatanController

*Controller* ini hanya digunakan untuk melakukan query data kecamatan yang sebelumnya telah ditambahkan ke dalam basis data. Fungsi *index* digunakan untuk mengembalikan seluruh data kecamatan pada tabel dan dapat diurutkan berdasarkan nama. Fungsi *show* digunakan untuk menampilkan satu data kecamatan. Berikut merupakan kode semu untuk KecamatanController.

```
1 class KecamatanController extends Controller{
2     public function index(Request $request){
... ..
21     }
22     public function show(Kecamatan $kecamatan){
... ..
26     }
27 }
```

*Kode Semu 5.13 KecamatanController*



#### k. CairanController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan model Cairan. Diantaranya seperti fungsi index untuk mengquery data cairan pada basis data. *Controller* ini juga mengatur proses untuk memperbaharui data cairan pada fungsi update, menambahkan data cairan pada fungsi store, dan penghapusan data cairan pada fungsi destroy. Seluruh kegiatan penambahan, penyuntingan, dan penghapusan data cairan hanya dapat dilakukan oleh super admin. Berikut adalah kode semu untuk CairanController.

```
1  Class CairanController extends Controller{
2      public function index(Request $request){
...  ...
28  }
29  public function store(StoreCairanRequest
    $request){
...  ...
37  }
38  public function show(Cairan $cairan){
...  ...
47  }
48  public function update(UpdateCairanRequest
    $request, Cairan $cairan){
...  ...
54  }
55  public function destroy(Cairan $cairan){
...  ...
60  }
61  }
```

*Kode Semu 5.14 CairanController*

## 1. ResepController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan model Resep. Terdapat fungsi `manualAuthorize` untuk mengatur tindakan mana saja yang dapat dilakukan oleh suatu user berdasarkan role user tersebut. Super admin dapat melakukan seluruh tindakan penambahan, penyuntingan dan penghapusan resep jika diperlukan, admin hanya dapat melakukan tindakan tersebut untuk resep milik pasien rumah sakitnya jika diperlukan, dan pasien hanya dapat melakukan tindakan tersebut khusus untuk resep milik dirinya sendiri. Terdapat fungsi `index` untuk mengquery data resep pada basis data sesuai kapasitas role user. Data akan difilter terlebih dahulu pada fungsi `prepareIndex` sesuai dengan role user, seperti role pasien hanya mengakses resepnya sendiri, super admin dapat mengakses seluruh data resep, dan admin hanya dapat mengakses data resep milik pasien dari pada rumah sakit admin tersebut. *Controller* ini juga mengatur proses untuk memperbaharui data resep pada fungsi `update`, menambahkan data resep pada fungsi `store`, dan penghapusan data resep pada fungsi `destroy`. Berikut adalah potongan kode `ResepController.php`.

```
1 class ResepController extends Controller{
2     private function manualAuthorize(User $user,
    Resep $potentialResep){
... ..
18     }
19     private function prepareIndex(User $user){
... ..
40     }
```

*Kode Semu 5.15 ResepController Bagian 1*

```

41     public function index(Request $request){
...   ...
79     }
80     public function store(StoreResepRequest
      $request){
...   ...
108    }
109    public function show(Resep $resep){
...   ...
126    }
127    public function update(UpdateResepRequest
      $request, Resep $resep){
...   ...
154    }
155    public function destroy(Resep $resep){
...   ...
175    }
176  }

```

*Kode Semu 5.16 ResepController Bagian 2*

#### m. PenggantianController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan dengan model Penggantian. Terdapat fungsi `manualAuthorize` untuk mengatur tindakan mana saja yang dapat dilakukan oleh suatu user berdasarkan role user tersebut. Super admin dapat melakukan seluruh tindakan penambahan, penyuntingan dan penghapusan data penggantian jika diperlukan, admin hanya dapat melakukan tindakan tersebut untuk penggantian milik pasien rumah sakitnya, dan pasien hanya dapat melakukan tindakan tersebut khusus untuk penggantian milik dirinya sendiri. Fungsi `index` digunakan untuk mengquery data penggantian pada basis data sesuai kapasitas role user. Data akan difilter terlebih

dahulu pada fungsi `prepareIndex` sesuai dengan role user, seperti role pasien hanya dapat mengakses data penggantian milik dirinya sendiri, super admin dapat mengakses seluruh data penggantian, dan admin hanya dapat mengakses data penggantian milik pasien yang dari rumah sakit admin tersebut. *Controller* ini juga mengatur proses untuk memperbaharui data penggantian pada fungsi `update`, menambahkan data penggantian pada fungsi `store`, dan penghapusan data penggantian pada fungsi `destroy`. Berikut adalah kode semu untuk `PenggantianController`.

```
1  class PenggantianController extends Controller{
2      private function manualAuthorize(User $user,
    Penggantian $potentialPenggantian){
...  ...
21  }
22  private function prepareIndex(User $user){
...  ...
43  }
44  public function index(Request $request){
...  ...
82  }
83  public function store(StorePenggantianRequest
    $request){
...  ...
139 }
140 public function show(Penggantian $penggantian){
...  ...
156 }
157 public function update(UpdatePenggantianRequest
    $request, Penggantian $penggantian){
```

*Kode Semu 5.17 PenggantianController Bagian 1*

```

...     ...
222     }
223     public function destroy(Penggantian
        $penggantian){
...     ...
235     }
236     }

```

*Kode Semu 5.18 PenggantianController Bagian 2*

n. KeluhanController

*Controller* ini hanya digunakan untuk melakukan query data keluhan yang sebelumnya telah ditambahkan ke dalam basis data. Fungsi index digunakan untuk mengembalikan seluruh data keluhan pada tabel. Fungsi show digunakan untuk menampilkan satu data keluhan. Berikut adalah kode semu untuk KeluhanController.

```

1   class KeluhanController extends Controller{
2       public function index(Request $request){
...   ...
7       }
8       public function show(Keluhan $keluhan){
...   ...
12      }
13     }

```

*Kode Semu 5.19 KeluhanController*

o. RiwayatKesehatanController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan model RiwayatKesehatan. Terdapat fungsi manualAuthorize untuk mengatur tindakan mana saja yang dapat dilakukan oleh suatu user berdasarkan role user tersebut.

Super admin dapat melakukan seluruh tindakan penambahan, penyuntingan dan penghapusan data riwayat kesehatan jika diperlukan, admin hanya dapat melakukan tindakan tersebut untuk riwayat kesehatan milik pasien rumah sakitnya, dan pasien hanya dapat melakukan tindakan tersebut khusus untuk riwayat kesehatan milik dirinya sendiri. Terdapat juga fungsi index untuk mengquery data riwayat kesehatan pada basis data sesuai kapasitas role user. Data akan difilter terlebih dahulu pada fungsi prepareIndex sesuai dengan role user, seperti role pasien hanya dapat mengakses data riwayat kesehatan milik dirinya sendiri, super admin dapat mengakses seluruh data riwayat kesehatan, dan admin hanya dapat mengakses data riwayat kesehatan milik pasien yang dari rumah sakit admin tersebut. *Controller* ini juga mengatur proses untuk memperbaharui data riwayat kesehatan pada fungsi update, menambahkan data riwayat kesehatan pada fungsi store, dan penghapusan data riwayat kesehatan pada fungsi destroy. Berikut adalah kode semu untuk *RiwayatKesehatanController*.

```
1 class RiwayatKesehatanController extends
  Controller{
2     private function manualAuthorize(User $user,
  RiwayatKesehatan $potentialRiwayatKesehatan){
...     ...
18     }
19     private function prepareIndex(User $user){
...     ...
37     }
38     public function index(Request $request){
...     ...
72     }
```

*Kode Semu 5.20 RiwayatKesehatanController Bagian 1*

```

73     public function
      store(StoreRiwayatKesehatanRequest $request){
...     ...
102     }
103     public function show(RiwayatKesehatan
      $riwayatKesehatan){
...     ...
118     }
119     public function
      update(UpdateRiwayatKesehatanRequest $request,
      RiwayatKesehatan $riwayatKesehatan){
...     ...
143     }
144     public function destroy(RiwayatKesehatan
      $riwayatKesehatan){
...     ...
155     }
156     }

```

*Kode Semu 5.21 RiwayatKesehatanController Bagian 2*

p. DetailPenggantianController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan dengan model DetailPenggantian. Terdapat fungsi manualAuthorize untuk mengatur tindakan mana saja yang dapat dilakukan oleh suatu user berdasarkan role user tersebut. Super admin dapat melakukan seluruh tindakan penambahan, penyuntingan dan penghapusan data detail penggantian jika diperlukan, admin hanya dapat melakukan tindakan tersebut untuk detail penggantian milik pasien rumah sakitnya, dan pasien hanya dapat melakukan tindakan tersebut khusus untuk detail penggantian milik dirinya sendiri. Terdapat juga fungsi index untuk mengquery data detail penggantian pada basis data sesuai kapasitas role user. Data akan difilter terlebih dahulu pada fungsi

prepareIndex sesuai dengan role user, seperti role pasien hanya dapat mengakses data detail penggantian milik dirinya sendiri, super admin dapat mengakses seluruh data detail penggantian, dan admin hanya dapat mengakses data detail penggantian milik pasien yang dari rumah sakit admin tersebut. *Controller* ini juga mengatur proses untuk memperbaharui data detail penggantian menggunakan fungsi update, menambahkan data detail penggantian menggunakan fungsi store, dan penghapusan data detail penggantian menggunakan fungsi destroy. Selain itu, *controller* ini juga menjadi penghubung/wrapper untuk api klasifikasi foto cairan menggunakan fungsi permissiDetection yang meneruskan request dari user ke api deteksi gambar pada aplikasi sebelumnya dan mengembalikan hasil dari api tersebut kembali ke user. Terdapat pula beberapa fungsi pembantu seperti saveImage untuk menyimpan foto penggantian pada Laravel Storage milik DPTSI ITS, dan fungsi extraCheckSanity untuk memastikan data resep dan detail penggantian dimiliki oleh pengguna yang sama. Berikut adalah kode semu untuk DetailPenggantianController.

```
1 class DetailPenggantianController extends
  Controller{
2     private function manualAuthorize(User $user,
      DetailPenggantian $potentialDetailPenggantian){
... ..
18    }
19    private function prepareIndex(User $user){
... ..
45    }
46    private function saveImage($image){
```

*Kode Semu 5.22 DetailPenggantianController Bagian 1*



```

...    ...
76      }
77      private function
      extraCheckSanity($pemilikPenggantianId,
      $pemilikResepId){
...    ...
84      }
85      public function index(Request $request){
...    ...
103     }
104     public function
store(StoreDetailPenggantianRequest $request){
...    ...
189     }
190     public function show(DetailPenggantian
      $detailPenggantian){
...    ...
206     }
207     public function
updateFotoDetailPenggantian(Request $request,
      DetailPenggantian $detailPenggantian){
...    ...
267     }
268     public function
update(UpdateDetailPenggantianRequest $request,
      DetailPenggantian $detailPenggantian){
...    ...
387     }
388     public function destroy(DetailPenggantian
      $detailPenggantian){
...    ...

```

*Kode Semu 5. 23 DetailPenggantianController Bagian 2*

```
401     }
402     public function permisiDetection(Request
      $request){
...     ...
422     }
423 }
```

*Kode Semu 5.24 DetailPenggantianController Bagian 3*

q. StatController

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan statistik pasien. Diantaranya seperti fungsi stat untuk mendapatkan data pasien dan jumlahnya untuk suatu kriteria berdasarkan kombinasi usia, jenis kelamin, provinsi, kabupaten, rumah sakit, dan *approval*. Kemudian terdapat fungsi *chartWilayah*, digunakan untuk mendapatkan total pasien untuk setiap wilayah yang diminta, berdasarkan provinsi ataupun kabupaten. Selain itu terdapat fungsi *chartUmur*, digunakan untuk mendapatkan total pasien untuk setiap rentang usia yang diminta. Berikut adalah kode semu untuk StatController.

```

1  class StatController extends Controller{
2      public function stat(Request $request){
...  ...
132  }
133  public function chartWilayah(Request $request){
...  ...
223  }
224  public function charUmur(Request $request){
...  ...
271  }
272  }

```

*Kode Semu 5.25 StatController*

r. ReportController

*Controller* ini digunakan untuk mengotor logika untuk proses pembuatan laporan penggantian pasien. Laporan dibentuk berdasarkan *request* yang berisikan tanggal mulai dan tanggal selesai laporan. Kemudian akan dilakukan *query* pada data penggantian dan detail penggantian yang berada pada rentang waktu tersebut. Selanjutnya akan dibuat file excel yang berisikan data yang telah diambil sebelumnya, terakhir akan dikirimkan email kepada dokter pasien yang berisikan laporan penggantian pasien tersebut. Di dalam *controller* ini terdapat beberapa fungsi pembantu untuk melakukan *parsing* dan *formatting* data penggantian dan detail penggantian pasien seperti fungsi *getDateFromTimeStamp* untuk mendapatkan tanggal yang diminta berdasarkan data timestamp dan *offset* waktu UTC, fungsi *getTimeFromTimeStamp* untuk mendapatkan waktu yang diminta berdasarkan timestamp dan *offset* waktu UTC, fungsi *processPenggantian* untuk mengubah data penggantian ke dalam bentuk *array* dan menambahkan nama kolom untuk setiap atribut

pada tabel penggantian, dan fungsi `processDetailPenggantian` untuk mengubah data detail penggantian ke dalam bentuk *array* dan menambahkan nama kolom untuk setiap atribut tabel detail penggantian. Berikut adalah kode semu untuk `ReportController`.

```
1  class ReportController extends Controller{
2      public function generateReport(Request
   $request){
...  ...
53  }
54  private function
   getDateFromTimeStamp($timestampDB, $utcoffset){
...  ...
60  }
61  private function
   getTimeFromTimeStamp($timestampDB, $utcoffset){
...  ...
67  }
68  private function
   processPenggantian($penggantian){
...  ...
99  }
100 private function
   processDetailPenggantian($detailPenggantian){
...  ...
142 }
143 }
```

*Kode Semu 5.26 ReportController*

s. `AdminController`

*Controller* ini digunakan untuk mengatur logika untuk proses-proses yang berkaitan dengan model `User` dengan role sebagai admin. Diantaranya seperti fungsi `index` untuk mengquery data user dengan role admin. *Controller* ini juga mengatur proses untuk

memperbaharui data admin menggunakan fungsi update, menambahkan data admin menggunakan fungsi store, dan penghapusan data admin menggunakan fungsi destroy. Admin di sini hanyalah data user / akun pengguna seperti pasien, tetapi pada atribut role diisi dengan kata admin. Tindakan penambahan, penyuntingan, dan penghapusan data admin hanya dapat dilakukan oleh super admin. Berikut adalah kode semu untuk AdminController.

```
1  class AdminController extends Controller{
2      public function index(Request $request){
...  ...
16     }
17     public function store(StoreAdminRequest
    $request){
...  ...
27     }
28     public function show($id){
...  ...
46     }
47     public function update(UpdateAdminRequest
    $request, User $admin){
...  ...
61     }
62     public function destroy(User $admin){
...  ...
76     }
77 }
```

*Kode Semu 5.27 AdminController*

### 5.3.5. *Filters*

*Filter* di sini merupakan *custom class* yang dibentuk untuk membantu proses filter pada saat mengambil data pada tabel dalam basis data. Terdapat kelas dasar yaitu *ApiFilter* yang akan digunakan untuk membentuk kelas filter untuk setiap tabel. Di kelas ini terdapat empat variabel yang digunakan untuk proses filter yaitu *safeParams*, *columnMap*, *safeWith*, dan *withMap*. Variabel *safeParams* adalah sebuah array yang akan diisi dengan parameter yang dapat digunakan untuk melakukan filter sesuai dengan kolom tabel. Variabel *columnMap* digunakan untuk menyesuaikan nama parameter dan nama kolom pada tabel jika diperlukan. Variabel *safeWith* digunakan untuk mendefinisikan parameter yang dapat digunakan untuk melakukan *query* data tambahan dari tabel lain yang berhubungan dengan tabel yang sedang difilter, sebagai contoh pengguna ingin mendapatkan data cairan dari suatu resep saat sedang melakukan *query* terhadap tabel resep. Terakhir, *withMap*, sama seperti *columnMap*, digunakan untuk menyesuaikan nama parameter dengan nama relasi sebenarnya dari tabel pada basis data. Kemudian terdapat dua fungsi yaitu *transform* dan *transformWith*. Fungsi *transform* digunakan untuk mengubah *request* filter yang diterima ke dalam bentuk *eloquent query* Laravel, fungsi *transformWith*, juga melakukan hal yang sama, yaitu mengubah *request* relasi yang diterima agar dapat digunakan untuk memanggil relasi dari *Eloquent model* Laravel. Berikut adalah kode semu untuk *ApiFilter*.

```
1  class ApiFilter{
2      protected $safeParams = [];
3      protected $columnMap = [];
4      protected $safewith = [];
5      protected $withMap = [];
6      Protected $operatorMap = [
...  ...
18  ];
19  public function transform(Request $request){
...  ...
47  }
48  public function transformWith(Request
    $request){
...  ...
64  }
65  }
```

*Kode Semu 5.28 ApiFilter*

*[Halaman ini sengaja dikosongkan]*



## **BAB 6**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba dilakukan terhadap Aplikasi Sahabat CAPD, baik dari segi Backend maupun UI/UX. Pengujian dilakukan untuk memastikan kualitas desain dan rancangan perangkat lunak yang dibangun. Selain itu, penilaian ini menjadi umpan balik untuk menilai kesesuaian hasil perangkat lunak dengan analisis kebutuhan pengguna.

#### **6.1. Tujuan Pengujian**

Pengujian dilakukan terhadap Aplikasi Sahabat CAPD untuk menilai ketepatan desain dan fungsionalitas backend dan UI/UX aplikasi. Selain itu, pengujian ini bertujuan untuk mendapatkan umpan balik langsung dari pasien yang akan menggunakan aplikasi ini.

#### **6.2. Kriteria Pengujian**

Kriteria pengujian aplikasi Sahabat CAPD terdiri dari dua bagian, yaitu pengujian UI/UX dan pengujian Backend. Pengujian UI/UX diselenggarakan melalui kuesioner yang disebarakan setelah sosialisasi, sedangkan pengujian backend aplikasi diuji berdasarkan fungsionalitas yang dapat dijalankan.

##### **6.2.1. UI / UX**

Penilaian UI/UX didasarkan rating pengguna mengenai kemudahan penggunaan aplikasi. Kuesioner terdiri dari tiga pertanyaan. Setiap pertanyaan dapat dijawab dengan rating dari rentang 1-5 untuk menandakan seberapa mudah fungsionalitas dapat dijalankan. Nilai satu menandakan fungsionalitas tersebut sangat sulit dijalankan sedangkan nilai lima menandakan

fungsionalitas sudah sangat mudah dijalankan. Untuk menilai ditargetkan rata-rata rating untuk setiap halaman mencapai angka empat.

### **6.2.2. Backend**

Pengujian backend dilakukan dengan menguji fungsionalitas backend yang diperlukan oleh aplikasi diantaranya meliputi:

- a. Kemampuan untuk melakukan autentikasi berupa pendaftaran untuk user serta login dan logout aplikasi untuk pasien dan admin.
- b. Kemampuan untuk melihat, menambah, menyunting, dan menghapus resep penggantian dari sisi pasien.
- c. Kemampuan untuk melihat dan memilih tenaga medis / dokter untuk pasien.
- d. Kemampuan untuk mengubah data diri dan foto profil untuk pasien.
- e. Kemampuan untuk mengganti password untuk pasien.
- f. Kemampuan untuk melihat dan mencatat tekanan darah dan berat badan pasien.
- g. Kemampuan untuk melihat dan mencatat penggantian dan pengeluaran cairan pasien.
- h. Kemampuan untuk melihat statistik berat badan dan tekanan darah pasien.
- i. Kemampuan untuk melihat statistik jenis kelamin dan usia pasien untuk admin.
- j. Kemampuan untuk melihat, menambah, dan menyunting dan menghapus dokter oleh admin.
- k. Kemampuan untuk memverifikasi terhadap pengguna yang baru mendaftar oleh admin
- l. Kemampuan untuk melihat data pengguna aktif beserta

detail data pasien.

- m. Kemampuan untuk melihat, menambah, menyunting, dan menghapus data cairan oleh admin.

### **6.3. Skenario Pengujian**

Pengujian dilakukan di RS Universitas Airlangga dengan fokus pada fungsionalitas frontend, backend, serta desain UI/UX aplikasi. Mengingat batasan waktu interaksi di RS Universitas Airlangga, pengujian difokuskan pada sisi pasien untuk menguji tiga fungsionalitas utama yaitu mendaftar dan masuk aplikasi, penambahan resep, penggantian dan pengeluaran cairan. Peserta sosialisasi dibagikan formulir yang dapat diisi untuk menilai desain dan pengalaman pengguna berdasarkan tiga fungsionalitas utama tersebut.

Untuk keseluruhan aplikasi dan sisi admin, skenario pengujian dilakukan dengan melakukan peran sebagai admin dan pasien yang akan menjalankan fitur-fitur dan seluruh kebutuhan fungsional dari sistem. Langkah-langkah untuk setiap kebutuhan fungsional yaitu sebagai berikut:

#### **6.3.1. Admin**

Berikut adalah kebutuhan fungsional dengan role akses sebagai admin.

- a. Admin melakukan login.
- b. Admin melihat data dashboard.
- c. Admin melihat data pasien di halaman pengguna aktif.
- d. Admin melihat detail data pasien.
- e. Admin melihat data pasien yang belum disetujui di halaman verifikasi pengguna.
- f. Admin melakukan persetujuan atau penolakan pasien.
- g. Admin melihat daftar dokter.

- h. Admin melakukan penambahan, penghapusan, dan pengubahan data dokter.
- i. Admin melihat informasi cairan.
- j. Admin melakukan penambahan, penghapusan, dan pengubahan data cairan.

### **6.3.2. Pasien**

Berikut adalah kebutuhan fungsional dengan role akses pasien.

- a. Pasien melakukan registrasi
- b. Pasien melakukan login.
- c. Pasien melihat data di beranda
- d. Pasien melakukan penambahan, pengubahan, dan penghapusan resep penggantian
- e. Pasien melakukan penambahan dan pengubahan penggantian cairan logbook
- f. Pasien melakukan penambahan, pengubahan, dan penghapusan riwayat kesehatan
- g. Pasien dapat melihat data statistik penggantian cairan, berat badan, dan tekanan darah
- h. Pasien dapat melihat data profil dan dokter
- i. Pasien dapat melakukan pengubahan data profil dan dokter
- j. Pasien dapat melihat pengaturan
- k. Pasien dapat melakukan penggantian password

### **6.4. Evaluasi Pengujian**

Berikut adalah hasil pengujian aplikasi Sahabat CAPD berdasarkan data formulir pada saat sosialisasi dengan pengguna dan juga pengujian yang dilakukan oleh pihak pengembang dan pembimbing lapangan.

#### 6.4.1. UI / UX

Hasil pengujian UI/UX didapatkan dari formulir yang disebarakan kepada peserta untuk menilai kemudahan menjalankan fungsionalitas utama. Sosialisasi aplikasi dihadiri empat peserta. Tabel 6.1 mendeskripsi demografi peserta.

*Table 6.1 Demografi Responden*

Kode	Nama	Jenis Kelamin	Usia
R01	Windy Gerda Wulandari	Perempuan	33
R02	Ika Yuniar Cahyanti	Perempuan	46
R03	Benar	Perempuan	33
R04	Retnoningsih	Perempuan	29

Tabel 6.2 menjelaskan hasil survei penilaian UI/UX aplikasi Sahabat CAPD. Responden menilai tingkat kemudahan menjalankan sebuah fungsionalitas.

*Table 6.2 Hasil Survei Penilaian Aplikasi Sahabat CAPD*

<b>Kriteria Pengujian</b>	<b>Rating Responden</b>		<b>Hasil Pengujian</b>
Login dan Daftar	R01	4	Terpenuhi
	R02	5	
	R03	5	
	R04	5	
	Mean	4,75	

Penambahan Resep	R01	4	Terpenuhi
	R02	4	
	R03	5	
	R04	4	
	Mean	4,25	
Pencatatan Penggantian Cairan	R01	4	Terpenuhi
	R02	4	
	R03	5	
	R04	5	
	Mean	4,5	
Mean		4,5	Terpenuhi

Tabel 6.2 merangkum penilaian responden terhadap tiga fungsionalitas utama aplikasi. Berdasarkan kuesioner, fungsionalitas utama dinilai cukup mudah untuk dijalankan. Setiap fungsionalitas memiliki rating minimal empat dengan rata-rata rating di atas empat.

#### **6.4.2. Backend**

Berikut adalah hasil pengujian kebutuhan fungsional aplikasi Sahabat CAPD dari sisi Backend.

*Table 6.3 Hasil Uji Coba Aplikasi*

<b>Kriteria Pengujian</b>	<b>Hasil Pengujian</b>
Melakukan pendaftaran	Terpenuhi
Melakukan login (Pasien dan Admin)	Terpenuhi
Melihat, menambah, menyunting, dan menghapus Resep Penggantian	Terpenuhi
Melihat , menambah, menyunting, dan menghapus Riwayat Kesehatan	Terpenuhi
Melihat dan memilih tenaga medis/dokter (Pasien)	Terpenuhi
Mengubah data diri dan foto profil	Terpenuhi
Mengubah password	Terpenuhi
Mencatat tekanan darah dan berat Badan	Terpenuhi
Mencatat dan melihat penggantian dan pengeluaran cairan	Terpenuhi
Melihat statistik berat badan dan tekanan darah	Terpenuhi
Melakukan penggantian password (pasien)	Terpenuhi
Melihat statistik jenis kelamin dan usia (Admin)	Terpenuhi
Menambah, menyunting, dan menghapus dokter (Admin)	Terpenuhi
Memverifikasi / approve pengguna (Admin)	Terpenuhi

Melihat data pengguna aktif dan detail pasien (Admin)	Terpenuhi
Melihat, menambah, menyunting, dan menghapus data cairan (Admin)	Terpenuhi



*[Halaman ini sengaja dikosongkan]*

## **BAB 7**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan pengembangan aplikasi Sahabat CAPD pada kegiatan KP di Departemen Teknik Informatika ITS adalah sebagai berikut:

- a. Aplikasi yang dibangun dapat membantu pasien dan keluarga untuk melakukan dan mencatat penggantian cairan CAPD tepat waktu.
- b. Admin dapat memantau pasien dengan mengakses catatan penggantian dan keadaan pasien.

#### **7.2. Saran**

Saran yang dapat penulis berikan untuk pengembangan aplikasi yang lebih baik selanjutnya adalah seperti berikut:

- a. Berikan opsi untuk memasukkan dan mengelola berbagai jenis data yang relevan dengan CAPD, seperti jadwal dialisis, catatan medis, dan hasil tes laboratorium.
- b. Perluas integrasi dengan sistem EHR (Electronic Health Record) agar data dapat dengan mudah diakses dan dipertukarkan antara aplikasi CAPD dan sistem lainnya. Hal ini dapat memudahkan pengguna dengan mengurangi jumlah aplikasi yang diperlukan.
- c. Sertakan fitur notifikasi dan pengingat untuk jadwal penting, seperti jadwal dialisis, pengambilan obat, dan kunjungan ke dokter.
- d. Sediakan layanan dukungan dan pelatihan untuk pengguna agar mereka dapat menggunakan aplikasi dengan optimal dan memahami fitur-fitur yang tersedia

## DAFTAR PUSTAKA

- [1] Reynaga-Ornelas, L. (2019). Impact of sleep and dialysis mode on quality of life in a Mexican population. *Southwest Journal of Pulmonary and Critical Care*, 18(5), 122–134. <https://doi.org/10.13175/swjpc017-19>
- [2] Nomoto, Y., Suga, T., Nakajima, K., Sakai, H., Osawa, G., Ota, K., Kawaguchi, Y., Sakai, T., Sakai, S., Shibata, M., & Takahashi, S. (1989). Acute Hydrothorax in Continuous Ambulatory Peritoneal Dialysis; A Collaborative Study of 161 Centers. *American Journal of Nephrology*, 9(5), 363–367. <https://doi.org/10.1159/000167997>
- [3] El-Shahawy, M. A., Kim, D., & Gadallah, M. F. (1998). Peritoneal Dialysis-Associated Peritonitis Caused by *Alcaligenes xylosoxidans*; *American Journal of Nephrology*, 18(5), 452–455. <https://doi.org/10.1159/000013370>
- [4] Eka Sari, F. I., Nurwardani, N. A., Fandiaz, M. A., Ulwiya, S., Jalil, M. M. A., & Navastara, D. A. (2022). Mobile Virtual Assistant for Continuous Ambulatory Peritoneal Dialysis Complication Detection. *CSRID (Computer Science Research and Its Development Journal)*, 14(2), 136. <https://doi.org/10.22303/csrid.14.2.2022.136-143>
- [5] “Sejarah,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/tentang-kami/sejarah/> (accessed Feb. 01, 2024).
- [6] “Visi dan Misi,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/tentang-kami/visi-dan-misi/> (accessed Feb. 01, 2024).
- [7] “Struktur Organisasi,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/dosen-staff/struktur-organisasi/> (accessed Jun. 17, 2024).
- [8] “Laboratorium,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/fasilitas/laboratorium/> (accessed Jun. 17, 2024).
- [9] Yu, A. W.-Y., Chau, K.-F., Ho, Y.-W., & Li, P. K.-T.

- (2007). Development of the “Peritoneal Dialysis First” Model in Hong Kong. *Peritoneal Dialysis International: Journal of the International Society for Peritoneal Dialysis*, 27(2\_suppl), 53–55.  
<https://doi.org/10.1177/089686080702702s09>
- [10] Oh, T. G., Baek, J. H., Jeong, S. J., Chae, Y. T., Ku, N. S., Chin, B. S., Han, S. H., Kim, C. O., Song, Y. G., Lee, H. Y., Kim, J. M., & Choi, J. Y. (2011). Continuous Ambulatory Peritoneal Dialysis-Associated Peritonitis Caused by *Achromobacter xylosoxidans* : A Case Report and Comprehensive Literature Review. *Infection and Chemotherapy*, 43(3), 275.  
<https://doi.org/10.3947/ic.2011.43.3.275>
- [11] Laudański, K., Nowak, Z., & Niemczyk, S. (2013). Age-related differences in the quality of life in end-stage renal disease in patients enrolled in hemodialysis or continuous peritoneal dialysis. *Medical Science Monitor*, 19, 378–385.  
<https://doi.org/10.12659/MSM.883916>
- [12] Almogahed, A., Omar, M., Zakaria, N. H., Muhammad, G., & AlQahtani, S. A. (2023). Revisiting Scenarios of Using Refactoring Techniques to Improve Software Systems Quality. *IEEE Access*, 11, 28800–28819.  
<https://doi.org/10.1109/ACCESS.2022.3218007>
- [13] Almogahed, A., Mahdin, H., Omar, M., Zakaria, N. H., Muhammad, G., & Ali, Z. (2023). Optimized Refactoring Mechanisms to Improve Quality Characteristics in Object-Oriented Systems. *IEEE Access*, 11, 99143–99158.  
<https://doi.org/10.1109/ACCESS.2023.3313186>
- [14] Gatrell, M., & Counsell, S. (2015). The effect of refactoring on change and fault-proneness in commercial C# software. *Science of Computer Programming*, 102, 44–56. <https://doi.org/10.1016/j.scico.2014.12.002>
- [15] Elish, K. O., & Alshayeb, M. (2011). A Classification of Refactoring Methods Based on Software Quality Attributes. *Arabian Journal for Science and Engineering*,

- 36(7), 1253–1267. <https://doi.org/10.1007/s13369-011-0117-x>
- [16] Mozilla, “Progressive Web Apps,” MDN Web Docs, Jun. 14, 2019. [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps) (accessed Feb. 03, 2024).
- [17] “Making PWAs Work Offline with Service Workers - Progressive Web Apps (PWAs) | MDN,” [developer.mozilla.org](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Offline_Service_workers), Nov. 29, 2023. [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps/Tutorials/js13kGames/Offline\\_Service\\_workers](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Offline_Service_workers) (accessed Feb. 03, 2024).
- [18] “What is Figma?,” Figma Learn. <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma> (accessed Feb. 03, 2024).
- [19] Laravel, “Laravel - The PHP Framework for Web Artisans,” [laravel.com](https://laravel.com/docs/10.x). <https://laravel.com/docs/10.x> (accessed Feb. 03, 2024).
- [20] R. West and M. Ray, “What Is SQL Server? - SQL Server,” [learn.microsoft.com](https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16), Sep. 06, 2023. <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16> (accessed Jun. 14, 2024).
- [21] Amazon Web Services, “What Is RESTful API? - RESTful API Beginner’s Guide - AWS,” Amazon Web Services, Inc., 2024. <https://aws.amazon.com/what-is/restful-api/> (accessed Jun. 14, 2024).

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Naufal Adli Purnama, lahir pada tanggal 19 Januari 2003 di Canberra. Penulis merupakan mahasiswa yang sedang menempuh studi di Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS). .



Putu Ravindra Wiguna, lahir pada tanggal 15 April 2002 di Singaraja. Penulis adalah mahasiswa yang sedang menempuh studi di Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember (ITS). Penulis aktif dalam kegiatan UKM Robotika di kampus.