



KERJA PRAKTIK - EF234603

Refaktorisasi Frontend 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:

Muhammad Ferdian Iqbal

5025201020

Aryan Shafa Wardana

5025211031

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

Refaktorisasi Frontend 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:

Muhammad Ferdian Iqbal

5025201020

Aryan Shafa Wardana

5025211031

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**

**Refaktorisasi Frontend Aplikasi Pencatatan dan
Pemantauan Pasien Continuous Ambulatory Peritoneal
Dialysis (CAPD)**

Oleh:

Muhammad Ferdian Iqbal

5025201020

Aryan Shafa Wardana

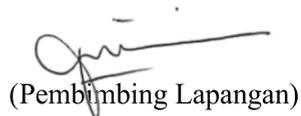
5025211031

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]

Refaktorisasi Frontend Aplikasi Pencatatan dan Pemantauan Pasien Continuous Ambulatory Peritoneal Dialysis (CAPD)

Nama Mahasiswa : Muhammad Ferdian Iqbal
NRP : 5025201020
Nama Mahasiswa : Aryan Shafa Wardana
NRP : 5025211031
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

Penyakit Ginjal Kronis (PGK) adalah masalah kesehatan global serius. Continuous Ambulatory Peritoneal Dialysis (CAPD) memungkinkan pasien menjalani dialisis mandiri, meningkatkan kualitas hidup hingga 90%. Namun, risiko komplikasi CAPD tetap tinggi akibat kelalaian dan kurangnya monitoring. Aplikasi SahabatCAPD dikembangkan untuk mencatat penggantian cairan dan memonitor kesehatan pasien melalui Android, serta menyediakan platform monitoring bagi rumah sakit. Karena hanya bisa diakses pengguna Android, aplikasi ini diubah menggunakan teknologi Progressive Web Apps (PWA) sehingga dapat diakses langsung melalui web. Didukung oleh React, Next.js, dan Tailwind CSS, aplikasi ini berfungsi optimal di berbagai perangkat. Pengembangan ini diharapkan meningkatkan efisiensi pemantauan mengurangi risiko komplikasi, dan memberikan solusi praktis bagi tenaga medis dalam merespons kondisi pasien PGK.

Kata Kunci : CAPD, Next js, Progressive Web App (PWA), React, Refaktorisasi, Tailwind CSS.

[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 Pasien 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, 26 Juni 2024

Tim Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xvi
DAFTAR KODE SEMU.....	xix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	3
1.3. Manfaat.....	3
1.4. Rumusan Masalah.....	3
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.....	4
1.6.4. Implementasi Sistem.....	5
1.6.5. Pengujian dan Evaluasi.....	5
1.7. Sistematika Laporan.....	5
BAB II PROFIL PERUSAHAAN.....	8
2.1. Sejarah Perusahaan.....	8
2.2. Visi.....	9
2.3. Misi.....	9
2.4. Struktur Organisasi.....	10
2.5. Laboratorium.....	10
BAB III TINJAUAN PUSTAKA.....	15
3.1. Continuous Ambulatory Peritoneal Dialysis (CAPD)....	15
3.2. Refaktorisasi.....	15
3.3. Progressive Web Apps.....	16
3.4. Figma.....	16

3.5. Typescript.....	16
3.6. Next Js.....	17
3.7. Tailwind CSS	18
3.8. Radix UI dan Shadcn.....	18
3.9. Axios	19
BAB IV ANALISIS DAN PERANCANGAN SISTEM	21
4.1. Analisis Sistem	21
4.2. Definisi Umum Aplikasi.....	21
4.3. Analisis Kebutuhan	21
4.3.1. Kebutuhan Fungsional	21
4.3.2. Kebutuhan Non-Fungsional	22
4.4. Diagram <i>Use Case</i>	23
4.5. Use Case Specification	24
4.6. Diagram Aktivitas	41
4.7. Conceptual Data Model.....	51
4.8. Physical Data Model	52
4.9. Deskripsi Data	52
BAB V IMPLEMENTASI SISTEM.....	61
5.1. Implementasi Design System	61
5.1.1. Mobile App (PWA)	61
5.1.2. Dashboard Admin	69
5.2. Struktur Folder Aplikasi	71
5.2.1. Mobile App (PWA)	71
5.2.2. Dashboard Admin	73
5.3. Implementasi Pengambilan Data API	76
5.3.1. Mobile App (PWA)	76
5.3.2. Dashboard Admin	78
5.4. Implementasi Antarmuka Pengguna.....	82
5.4.1. Mobile App (PWA)	82
5.4.2. Dashboard Admin	93
BAB VI PENGUJIAN DAN EVALUASI.....	99

6.1. Tujuan Pengujian.....	99
6.2. Kriteria Pengujian.....	99
6.3. Skenario Pengujian.....	100
6.3.1. Admin	100
6.3.2. Pasien	100
6.3.3. Evaluasi Pengujian.....	101
BAB VII KESIMPULAN DAN SARAN	105
7.1. Kesimpulan.....	105
7.2. Saran.....	105
DAFTAR PUSTAKA.....	106
BIODATA PENULIS.....	110

[Halaman ini sengaja dikosongkan]

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>	23
<i>Gambar 4.2 Diagram Use Case Admin Aplikasi CAPD</i>	24
<i>Gambar 4.3 Diagram Use Case Superadmin Aplikasi CAPD</i>	24
<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
<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 Design System Dashboard Admin</i>	69
<i>Gambar 5.2 Struktur Folder Mobile App (PWA)</i>	72
<i>Gambar 5.3 Isi Folder src pada Mobile App (PWA)</i>	73
<i>Gambar 5.4 Struktur Folder Dashboard Admin</i>	74
<i>Gambar 5.5 Isi folder src Dashboard Admin</i>	75
<i>Gambar 5.5 Isi folder services pada Dashboard Admin</i>	80
<i>Gambar 5.6 Halaman Register Mobile App (PWA)</i>	82
<i>Gambar 5.7 Halaman Login Mobile App (PWA)</i>	83
<i>Gambar 5.8 Halaman lupa password Mobile App (Kiri). Halaman reset password Mobile App (Kanan)</i>	84
<i>Gambar 5.9 Halaman Beranda Mobile App (PWA)</i>	85
<i>Gambar 5.10 Halaman Logbook Mobile App (PWA)</i>	86

<i>Gambar 5.11 Halaman Mencatat Penggantian Cairan Mobile App (PWA)</i>	87
<i>Gambar 5.12 Halaman Resep Penggantian Mobile App (PWA)</i>	88
<i>Gambar 5.13 Halaman Riwayat Kesehatan Mobile App (PWA)</i>	89
<i>Gambar 5.14 Halaman Statistik Mobile App (PWA)</i>	90
<i>Gambar 5.15 Halaman Profil Mobile App (PWA)</i>	91
<i>Gambar 5.16 Halaman Chatbot Mobile App (PWA)</i>	92
<i>Gambar 5.17 Halaman Pengaturan Mobile App (PWA)</i>	93
<i>Gambar 5.18 Halaman Login Dashboard Admin</i>	94
<i>Gambar 5.19 Halaman Beranda Dashboard Admin</i>	94
<i>Gambar 5.20 Halaman Pengguna Aktif Dashboard Admin</i>	95
<i>Gambar 5.21 Halaman Profil Pasien Dashboard Admin</i>	96
<i>Gambar 5.22 Halaman Verifikasi Pengguna Dashboard Admin</i>	96
<i>Gambar 5.23 Halaman Daftar Dokter Dashboard Admin</i>	97
<i>Gambar 5.24 Halaman Daftar Cairan Dashboard Admin</i>	97

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

<i>Tabel 4.1 Kebutuhan Fungsional Aplikasi CAPD.....</i>	<i>22</i>
<i>Tabel 4.2 Kebutuhan Fungsional Admin Aplikasi CAPD</i>	<i>22</i>
<i>Tabel 4.3 Kebutuhan Fungsional Superadmin Aplikasi CAPD ..</i>	<i>22</i>
<i>Tabel 4.4 Kebutuhan Non-Fungsional Aplikasi CAPD.....</i>	<i>23</i>
<i>Tabel 4.5 Tabel Use Case Aplikasi CAPD Mendaftarkan Akun ..</i>	<i>25</i>
<i>Tabel 4.6 Tabel Use Case Sahabat CAPD Melakukan Login.....</i>	<i>26</i>
<i>Tabel 4.7 Tabel Use Case Sahabat CAPD Menambahkan Resep Penggantian</i>	<i>27</i>
<i>Tabel 4.8 Tabel Use Case Sahabat CAPD Mencatat Pemasukan Cairan.....</i>	<i>29</i>
<i>Tabel 4.9 Tabel Use Case Sahabat CAPD Mencatat Pengeluaran Cairan.....</i>	<i>30</i>
<i>Tabel 4.10 Tabel Use Case Sahabat CAPD Memverifikasi Pengguna.....</i>	<i>32</i>
<i>Tabel 4.11 Tabel Use Case Sahabat CAPD Melihat Detail Pencatatan Pengguna Aktif.....</i>	<i>33</i>
<i>Tabel 4.12 Tabel Use Case Sahabat CAPD Mengelola Informasi Dokter.....</i>	<i>34</i>
<i>Tabel 4.13 Tabel Use Case Sahabat CAPD Mengelola Informasi Cairan Penggantian.....</i>	<i>37</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>54</i>
<i>Tabel 4.19 Deskripsi Data pada Tabel Riwayat Kesehatan Aplikasi CAPD</i>	<i>55</i>

<i>Tabel 4.20 Deskripsi Data pada Tabel Cairan Aplikasi CAPD..</i>	<i>55</i>
<i>Tabel 4.21 Deskripsi Data pada Tabel Resep Aplikasi CAPD....</i>	<i>56</i>
<i>Tabel 4.22 Deskripsi Data pada Tabel Penggantian Aplikasi CAPD</i>	<i>56</i>
<i>Tabel 4.23 Deskripsi Data pada Tabel Detail Penggantian Aplikasi CAPD</i>	<i>57</i>
<i>Tabel 4.24 Deskripsi Data pada Tabel Pertanyaan Keamanan Aplikasi CAPD</i>	<i>58</i>
<i>Tabel 4.25 Deskripsi Data pada Tabel Provinsi Aplikasi CAPD</i>	<i>58</i>
<i>Tabel 4.26 Deskripsi Data pada Tabel Kabupaten/Kota Aplikasi CAPD</i>	<i>59</i>
<i>Tabel 4.27 Deskripsi Data pada Tabel Kecamatan Aplikasi CAPD</i>	<i>59</i>
<i>Tabel 6.1 Hasil Evaluasi Pengujian Aplikasi Dashboard Admin</i>	<i>101</i>
<i>Tabel 6.2 Hasil Evaluasi Pengujian Aplikasi Mobile PWA Pasien</i>	<i>102</i>

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SEMU

<i>Kode Semu 5.1 Konfigurasi file tailwind.config.ts (Bagian 1)</i>	62
<i>Kode Semu 5.1 Konfigurasi file tailwind.config.ts (Bagian 2)</i>	63
<i>Kode Semu 5.2 Konfigurasi warna di file tailwind.config.ts</i>	64
<i>Kode Semu 5.3 Konfigurasi kelas utilitas di file tailwind.config.ts</i>	65
<i>Kode Semu 5.4 Konfigurasi animasi di file tailwind.config.ts</i>	66
<i>Kode Semu 5.5 Konfigurasi directive tailwind dan font-face di file</i> <i>src/app/globals.css</i>	67
<i>Kode Semu 5.6 Konfigurasi directive layer base di file</i> <i>src/app/globals.css (Bagian 1)</i>	67
<i>Kode Semu 5.6 Konfigurasi directive layer base di file</i> <i>src/app/globals.css (Bagian 2)</i>	68
<i>Kode Semu 5.7 Konfigurasi directive layer utilities di file</i> <i>src/app/globals.css</i>	68
<i>Kode Semu 5.9 Konfigurasi directive layer utilities di file</i> <i>tailwind.config.css (Bagian 1)</i>	69
<i>Kode Semu 5.9 Konfigurasi directive layer utilities di file</i> <i>tailwind.config.css (Bagian 2)</i>	70
<i>Kode Semu 5.9 Konfigurasi directive layer utilities di file</i> <i>tailwind.config.css (Bagian 3)</i>	71
<i>Kode Semu 5.10 Konfigurasi directive layer utilities di file</i> <i>src/app/globals.css</i>	71
<i>Kode Semu 5.11 Fungsi callApi di file src/lib/api/index.ts (Bagian</i> <i>1)</i>	76
<i>Kode Semu 5.11 Fungsi callApi di file src/lib/api/index.ts (Bagian</i> <i>2)</i>	77
<i>Kode Semu 5.12 Fungsi callApi di file src/services/detail-</i> <i>penggantian/index.ts</i>	78
<i>Kode Semu 5.13 Konfigurasi utama API di file src/lib/api/index.ts</i> <i>(Bagian 1)</i>	78

*Kode Semu 5.13 Konfigurasi utama API di file src/lib/api/index.ts
(Bagian 2)..... 79*

*Kode Semu 5.15 Penerapan API pada cairan di file
src/services/cairan/index.ts 81*

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Penyakit ginjal kronis (PGK) merupakan masalah kesehatan global yang memerlukan perhatian serius dalam manajemennya. Salah satu metode penanganan PGK yang menonjol adalah Continuous Ambulatory Peritoneal Dialysis (CAPD). CAPD merupakan metode dialisis mandiri yang memanfaatkan membran rongga perut (peritoneum) sebagai penyaring, dan cairan dialisat (dextrose) steril sebagai pengganti fungsi ginjal yang dilakukan 3-5 kali sehari [1]. Keunggulan utama CAPD bagi pasien PGK adalah kemampuannya untuk memungkinkan dialisis dilakukan secara mandiri tanpa harus ke rumah sakit, yang secara signifikan meningkatkan kualitas hidup pasien hingga 90% dibandingkan dengan metode terapi ginjal lainnya [2]. Namun, tingkat risiko kematian pasien CAPD akibat komplikasi cukup tinggi, seringkali disebabkan oleh kelalaian, kesalahan teknis, dan kurangnya monitoring, dengan tingkat kelalaian mencapai 74% [3]. Selain itu, minimnya follow-up data oleh pasien CAPD kepada tenaga medis membuat komplikasi semakin sulit terdeteksi, dan keberhasilan terapi sangat bergantung pada perilaku normatif pasien, pelatihan, serta pemantauan teratur oleh tenaga medis [4].

Sifat dari CAPD yang memungkinkan pasien melakukannya secara mandiri telah mendorong pengembangan berbagai sistem pemantauan jarak jauh. Salah satu inovasi sebelumnya adalah aplikasi pencatatan CAPD yang dapat mendeteksi abnormalitas pada cairan pengeluaran pasien [5]-[7]. Sistem aplikasi ini dikembangkan untuk berjalan pada handphone pasien menggunakan sistem Android, sehingga pasien dapat

melakukan pencatatan setiap kali mengganti cairan dan dengan mudah memantau perkembangan kondisi kesehatannya. Selain itu, aplikasi ini juga menyediakan platform monitoring bagi pihak rumah sakit dalam bentuk website, memungkinkan tenaga medis untuk memantau perkembangan pasien secara lebih efektif dan memberikan pengobatan lanjutan yang sesuai dengan kondisi kesehatan pasien. Namun, aplikasi yang saat ini hanya dapat diakses menggunakan handphone Android, membuat pengguna yang memiliki handphone dengan sistem IOS tidak dapat menggunakannya, yang tentunya menyulitkan pasien yang tidak memiliki perangkat Android.

Untuk mengatasi keterbatasan ini, dilakukan refaktorisasi aplikasi pencatatan CAPD dengan menggunakan teknologi progressive web apps (PWA). PWA dipilih karena kemampuannya untuk membangun aplikasi multi-platform yang cepat dan mudah diakses. Pedoman utama pengembangan ini adalah aplikasi yang telah ada, yaitu SahabatCAPD. Dengan PWA, pengguna tidak harus mengunduh aplikasi terlebih dahulu untuk menggunakan layanan ini karena dapat diakses langsung melalui alamat situs yang tersedia. Selain itu, pengguna juga memiliki opsi untuk mengunduh aplikasi dari situs tersebut untuk akses yang lebih cepat. Pilihan fleksibel ini diharapkan dapat membantu pasien dalam menggunakan aplikasi lebih cepat dan efisien, tanpa tergantung pada jenis perangkat yang mereka miliki [8].

Selain itu, aplikasi dashboard juga akan dikembangkan untuk memantau setiap pasien oleh pihak rumah sakit. Dashboard ini akan memberikan kemudahan bagi tenaga medis dalam mengetahui perkembangan kesehatan pasien secara real-time dan memberikan pengobatan lanjutan yang sesuai dengan kondisi pasien. Pengembangan kedua aplikasi ini menggunakan teknologi modern seperti React, Next.js, dan Tailwind CSS. Teknologi

tersebut dipilih karena kemudahannya dalam pengembangan serta komunitas pendukung yang besar, sehingga memastikan aplikasi yang dikembangkan dapat berfungsi dengan optimal dan mendapatkan dukungan yang diperlukan jika ada masalah teknis.

Dengan pengembangan ini, diharapkan bahwa aplikasi multi-platform untuk CAPD dapat meningkatkan efisiensi dan efektivitas pemantauan serta pengobatan bagi pasien PGK. Peningkatan awareness sistem terhadap abnormalitas melalui real-time monitoring akan sangat membantu dalam deteksi dini komplikasi, sehingga intervensi medis dapat dilakukan lebih cepat dan tepat. Secara keseluruhan, inovasi ini bertujuan untuk meningkatkan kualitas hidup pasien CAPD, mengurangi risiko komplikasi, dan memberikan solusi praktis bagi tenaga medis dalam memantau dan merespons kondisi pasien dengan lebih baik.

1.2. Tujuan

Tujuan dari kerja praktik ini adalah untuk mengembangkan refaktorisasi aplikasi pencatatan dan pemantauan pasien CAPD.

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 cara mengimplementasikan frontend aplikasi pencatatan CAPD dalam bentuk Progressive Web App

(PWA)?

2. Bagaimana cara mengimplementasikan frontend aplikasi pemantauan CAPD?

1.5. Lokasi dan Waktu Kerja Praktik

Pengerjaan kerja praktik ini lakukan 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 Pembimbing Lapangan 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 SQL Server (Microsoft). Selain itu, dikembangkan aplikasi Dashboard untuk admin rumah sakit.

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 II

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) [9].

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 [10].

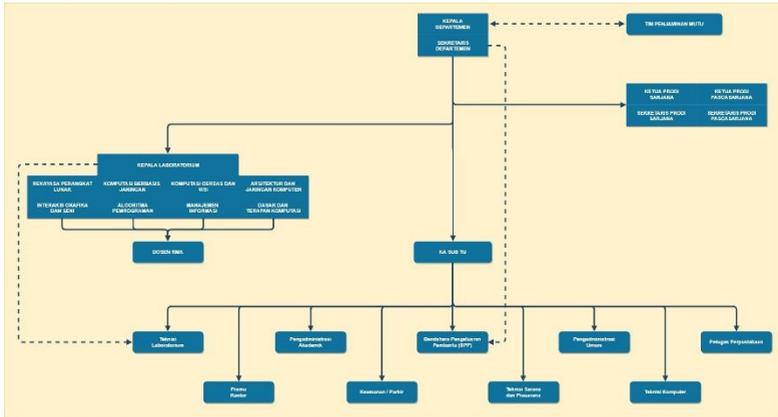
2.3. Misi

Berikut adalah misi-misi yang dimiliki oleh Departemen Teknik Informatika, ITS [10].

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

Gambar 2.1 merupakan gambar struktur organisasi yang terdapat di Departemen Teknik Informatika, ITS [11].



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 [12].

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 III

TINJAUAN PUSTAKA

3.1. Continuous Ambulatory Peritoneal Dialysis (CAPD)

Continuous Ambulatory Peritoneal Dialysis (CAPD) adalah salah satu jenis perawatan cuci darah untuk gagal ginjal yang menggunakan lapisan rongga perut (*peritoneum*) sebagai penyaring alami untuk menghilangkan zat-zat sisa metabolisme, elektrolit, mineral, dan cairan berlebihan akibat penurunan fungsi ginjal [13]. Proses ini dapat dilakukan secara mandiri oleh pasien hingga 4 kali sehari di rumah masing-masing dan tidak membutuhkan mesin [14]. Sebelum prosedur CAPD bisa dimulai, sebuah kateter harus dimasukkan di rongga perut pasien. Langkah-langkah untuk melakukan CAPD adalah sebagai berikut.

1. Pasien meletakkan kantong medis yang berisi cairan dialisis pada ketinggian se-bahu
2. Dengan bantuan gaya gravitasi, cairan dialisis dari kantong tersebut akan disalurkan ke dalam rongga perut
3. Jika semua cairan dialisis sudah masuk ke dalam rongga perut maka kateter perlu ditutup dengan rapat
4. Setelah 4 hingga 6 jam, cairan dialisis yang mengandung zat sisa dialirkan keluar dari rongga perut dan bisa dibuang ke toilet.

3.2. Refaktorisasi

Refaktorisasi atau lebih dikenal dengan *code refactoring* merupakan proses untuk memperbaiki kode tanpa menghilangkan atau mengubah fungsionalitas-fungsionalitas awal. Tujuan dilakukan refaktorisasi adalah untuk meningkatkan ekstensibilitas, pemeliharaan, dan keterbacaan dari kode suatu program. Beberapa manfaat dari refaktorisasi seperti ukuran kode menurun dan kode yang membingungkan dapat distruktur ulang menjadi kode yang

bisa dibaca [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 [8].

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.

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 [16].

3.5. Typescript

Typescript adalah bahasa pemrograman open source yang dikembangkan oleh Microsoft dan menggunakan Javascript sebagai bahasa utamanya dengan fitur – fitur yang lebih baik. Beberapa fitur yang dimiliki Typescript adalah static typing,

interface, type definition, class, dan lainnya. Fitur – fitur tersebut sangat berguna bagi developer terutama saat pengembangan aplikasi skala besar. Typescript dikembangkan karena sangat sulit mengembangkan aplikasi skala besar menggunakan Javascript. Javascript memiliki keterbatasan dalam menghubungkan berbagai unit kode yang besar. Selain itu, penulisan kode Javascript terlalu bebas sehingga menjadikan developer sulit untuk mengelola kode aplikasi berskala besar. Diperlukan melakukan konfigurasi lebih ketika menggunakan Javascript jika digunakan untuk mengembangkan aplikasi besar. Jenis kesalahan umum yang dilakukan developer saat menggunakan Javascript adalah type error, yaitu tipe data atau jenis nilai tertentu dapat digunakan di tempat di mana nilai yang berbeda diharapkan. Kesalahan tersebut dapat terjadi karena kesalahan penulisan sederhana, kegagalan memahami API, atau lainnya. Typescript dikembangkan di atas Javascript sehingga seluruh aturan penulisan Javascript dapat digunakan pada Typescript. Hal tersebut berarti seluruh kode Javascript adalah Typescript, tetapi tidak berlaku sebaliknya karena terdapat fitur Typescript yang tidak valid di Javascript [17].

3.6. Next Js

Next.js adalah framework atau kerangka kerja yang dikembangkan oleh Vercel, perusahaan teknologi ternama yang berfokus pada pengembangan produk untuk developer dan desainer. Next.js digunakan untuk membuat aplikasi web dengan berbagai kelebihannya. Disebut framework karena Next.js merupakan pengembangan dari React.js dengan tambahan berbagai fitur. Pada penggunaannya, React digunakan untuk pembuatan antarmuka pengguna dan Next.js digunakan untuk mempercepat proses bundling, compiling, dan optimisasi. Hal tersebut membuat pengguna lebih fokus dalam pengembangan aplikasi daripada melakukan konfigurasi yang dapat memakan

banyak waktu. Dengan menggunakan Next.js, aplikasi yang dibuat akan dinamis, interaktif, dan cepat [18].

3.7. Tailwind CSS

Tailwind CSS adalah salah satu framework CSS yang bersifat utility-first, yaitu penerapan styling dengan menggunakan class – class yang disediakan tanpa harus meninggalkan file HTML. Hal ini membuat pembuatan antarmuka menjadi lebih cepat dan mudah dikustom. Keunggulan lain yang dimiliki Tailwind CSS adalah sederhana, responsif, dan component-friendly. Tailwind CSS sangat membebaskan penggunaannya untuk menggunakan class – class yang disediakan sesuai kebutuhan. Selain itu, class – class yang sudah disediakan juga dapat dikustom lagi karena Tailwind CSS ditulis dalam bentuk PostCSS dan dikonfigurasi dengan Javascript sehingga pengembang dapat membuat sistem desainnya sendiri. Tailwind CSS memiliki dokumentasi yang sangat lengkap dan mudah dipahami. Framework ini memungkinkan untuk digabungkan dengan teknologi pengembangan aplikasi lainnya, seperti Next.js, Laravel, dan Nuxt.js. Di dalam dokumentasi resmi Tailwind CSS, sudah disediakan cara penggunaannya. Pada saat pertama kali mengunduh Tailwind CSS, akan disediakan file `globals.css` dan `tailwind.config.js`. Kedua file tersebut digunakan untuk melakukan konfigurasi Tailwind CSS yang dapat disesuaikan dengan kebutuhan [19].

3.8. Radix UI dan Shadcn

Radix UI adalah pustaka komponen UI open-source yang menyediakan fondasi yang kuat bagi pengembangan antarmuka pengguna. Dengan beragam komponen tingkat rendah seperti tombol, accordion, dropdown, dan lainnya, Radix UI membantu pengembang menghemat waktu dan upaya yang biasanya dibutuhkan untuk membangun komponen-komponen ini secara

mandiri. Keuntungan utamanya adalah kinerja yang optimal dan kepatuhan terhadap standar aksesibilitas, serta kemampuan untuk menciptakan antarmuka pengguna yang konsisten dan estetis. Dengan Radix UI, pengembang dapat fokus pada fungsionalitas aplikasi mereka sambil memastikan desain yang berkualitas tinggi dan mudah diakses oleh semua pengguna [20].

Shadcn adalah sekumpulan komponen reusable yang dikembangkan di atas Radix UI. Komponen – komponen yang disediakan dapat langsung digunakan atau disesuaikan dengan kebutuhan aplikasi. Dengan menggunakan komponen – komponen tersebut pembuatan aplikasi akan semakin mudah [21].

3.9. Axios

Axios adalah sebuah pustaka JavaScript yang digunakan untuk melakukan permintaan HTTP dari browser atau Node.js. Pustaka ini memungkinkan Anda untuk berinteraksi dengan API dan sumber daya eksternal lainnya dari aplikasi JavaScript Anda dengan mudah. Axios mendukung berbagai metode HTTP seperti GET, POST, PUT, DELETE, dan sebagainya, serta menyediakan berbagai fitur termasuk dukungan untuk permintaan asinkron, penanganan respons dan kesalahan, pembatalan permintaan, dan lainnya [22].

[Halaman ini sengaja dikosongkan]

BAB IV

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

Kode Kebutuhan	Deskripsi Kebutuhan
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

Kode Kebutuhan	Deskripsi Kebutuhan
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

Kode Kebutuhan	Deskripsi Kebutuhan
F-201	Mengelola Informasi Admin

4.3.2. Kebutuhan Non-Fungsional

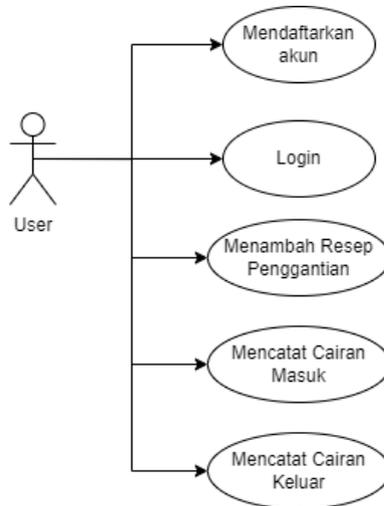
Kebutuhan non-fungsional adalah batasan dan karakteristik dari sebuah sistem. Kebutuhan non-fungsional dari aplikasi CAPD terdapat pada Tabel 4.4.

Tabel 4.4 Kebutuhan Non-Fungsional Aplikasi CAPD

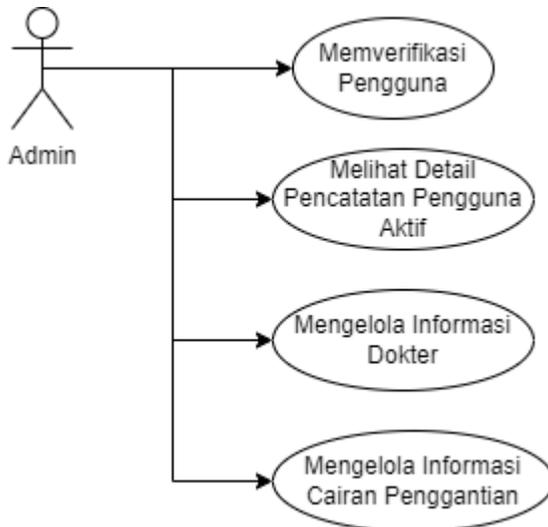
Kode Kebutuhan	Deskripsi Kebutuhan
NF-001	Sistem dapat diakses oleh pengguna

4.4. Diagram Use Case

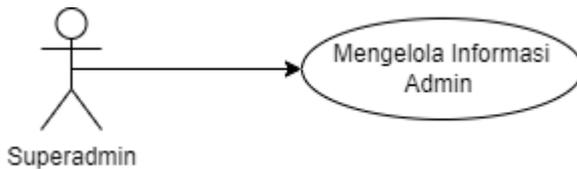
Kebutuhan fungsional pengguna digambarkan dalam bentuk *Use Case Diagram* untuk memudahkan pemahaman fungsional aplikasi CAPD. Diagram tersebut dapat dilihat pada Gambar 4.1, 4.2, dan 4.3.



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*. Berikut adalah tabel masing-masing *use case* tersebut.

a. Mendaftarkan Akun

Tabel 4.5 merupakan 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"> 1. Aktor mengisi form daftar 2. Aktor menekan tombol 'Daftar' 3. Sistem memasukkan akun baru ke daftar untuk diverifikasi
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor tidak mengisi formulir dengan lengkap <ol style="list-style-type: none"> a. Sistem memberi peringatan bahwa kolom harus diisi.

	b. Kembali ke Alur Normal nomor 1
Pengecualian	-

b. Melakukan Login

Tabel 4.6 merupakan 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	1. Aktor mengisi form login 2. Aktor menekan tombol ‘Masuk’

Secara Normal	<ul style="list-style-type: none"> 3. Sistem mencocokkan data login dengan database 4. Sistem menampilkan halaman beranda
Alur Kejadian Alternatif	<ul style="list-style-type: none"> 1. Aktor tidak mengisi formulir dengan lengkap <ul style="list-style-type: none"> a. Sistem memberi peringatan bahwa kolom harus diisi. b. Kembali ke Alur Normal nomor 1. 3. Data yang diinputkan tidak cocok dengan basis data <ul style="list-style-type: none"> a. Sistem memberi peringatan bahwa email atau password salah. b. Kembali ke Alur Normal nomor 1.
Pengecualian	-

c. Menambahkan Resep Penggantian

Tabel 4.7 merupakan 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"> 1. Aktor menekan tombol 'Resep' pada halaman beranda 2. Sistem akan menampilkan halaman berisi resep pengguna yang terdaftar 3. Aktor menekan tombol 'Tambah' 4. Sistem menampilkan halaman formulir resep 5. Aktor mengisi informasi resep 6. Aktor menekan tombol 'Simpan' 7. Sistem menyimpan resep baru 8. Sistem menampilkan kembali daftar resep
Alur Kejadian Alternatif	-
Pengecualian	-

d. Mencatat Pemasukan Cairan

Tabel 4.8 merupakan 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">1. Aktor menekan ‘Catat’ di logbook pada jam pemasukan yang ingin dicatat2. Sistem menampilkan formulir cairan masuk3. Aktor mengisi formulir4. Aktor menekan ‘Simpan’

	5. Sistem menyimpan catatan 6. Sistem menampilkan halaman logbook pada hari tersebut
Alur Kejadian Alternatif	-
Pengecualian	-

e. Mencatat Pengeluaran Cairan

Tabel 4.9 merupakan 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"> 1. Aktor menekan ‘Catat’ di logbook pada jam pengeluaran yang ingin dicatat 2. Sistem menampilkan formulir cairan keluar 3. Aktor mengisi formulir 4. Aktor menekan ‘Simpan’ 5. Sistem menyimpan catatan 6. Sistem menampilkan halaman logbook pada hari tersebut
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 3. Aktor menekan ‘Ambil Foto Cairan Buangan’ <ol style="list-style-type: none"> a. Sistem menampilkan aplikasi kamera b. Aktor mengambil foto cairan c. Sistem menyimpan dan melampirkan foto pada pengeluaran tersebut d. Kembali ke langkah 3
Pengecualian	-

f. Memverifikasi Pengguna

Tabel 4.10 merupakan 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"> 1. Aktor menekan ‘Verifikasi Pengguna’ di sidebar 2. Sistem menampilkan daftar pengguna yang menunggu verifikasi 3. Aktor menekan tombol ‘Setujui’ untuk memverifikasi akun

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

g. Melihat Detail Pencatatan Pengguna Aktif

Tabel 4.11 merupakan 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"> 1. Aktor menekan ‘Pengguna Aktif’ di sidebar. 2. Sistem menampilkan daftar pengguna. 3. Aktor menekan tombol ‘detail’ pada pengguna yang ingin dilihat. 4. Sistem menampilkan detail pengguna tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

h. Mengelola Informasi Dokter

Tabel 4.12 merupakan 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"> 1. Aktor menekan 'Dokter' di sidebar. 2. Sistem menampilkan daftar dokter. 3. Aktor melakukan operasi pada informasi dokter. 4. Sistem menyimpan perubahan.
Alur Kejadian Alternatif	<p>3a. Aktor menambah data dokter</p> <ol style="list-style-type: none"> i. Aktor menekan tombol 'Tambah Data Dokter'. ii. Sistem menampilkan form informasi dokter. iii. Aktor mengisi form informasi dokter.

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

i. Mengelola Informasi Cairan Penggantian

Tabel 4.13 merupakan 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"> 1. Aktor menekan 'Cairan Penggantian' di sidebar. 2. Sistem menampilkan daftar cairan penggantian. 3. Aktor melakukan operasi pada informasi cairan penggantian. 4. Sistem menyimpan perubahan.

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

	<p>yang ingin diubah.</p> <p>ii. Sistem menampilkan form informasi cairan penggantian.</p> <p>iii. Aktor mengisi form informasi cairan penggantian.</p> <p>iv. Aktor menekan tombol ‘Simpan’.</p> <p>v. Sistem menyimpan informasi cairan penggantian.</p>
Pengecualian	-

j. Mengelola Informasi Admin

Tabel 4.14 merupakan 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

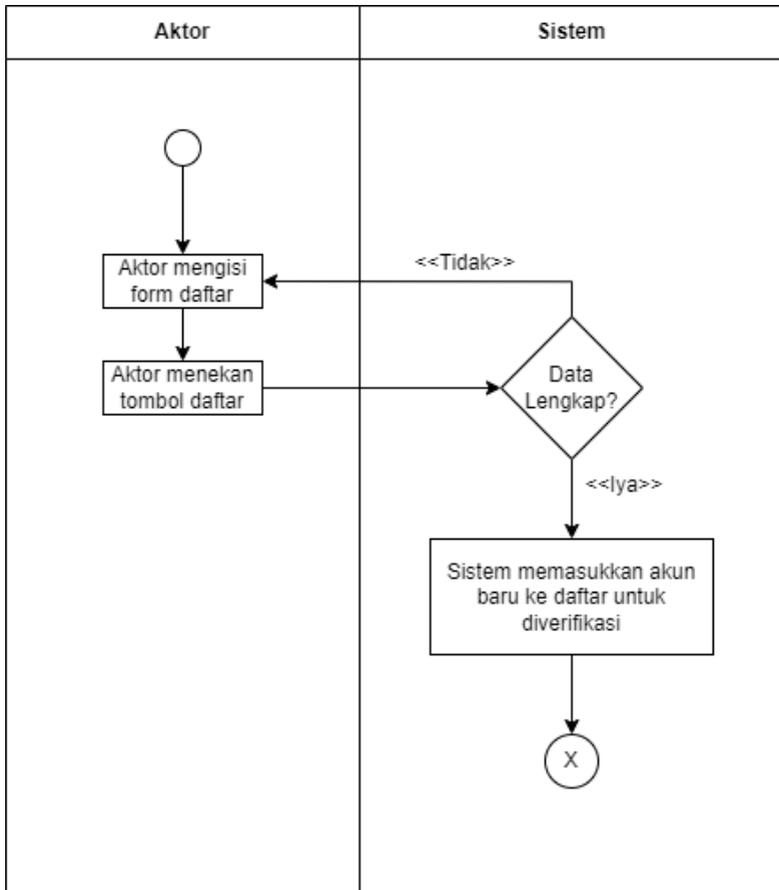
Pemicu	Aktor menekan tombol 'Admin' di sidebar.
Aktor	Superadmin
Kondisi Awal	Superadmin masuk dashboard superadmin.
Kondisi Akhir	Informasi daftar admin diperbarui.
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Aktor menekan 'Cairan Penggantian' di sidebar. 2. Sistem menampilkan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

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

4.6. Diagram Aktivitas

a. Mendaftarkan Akun

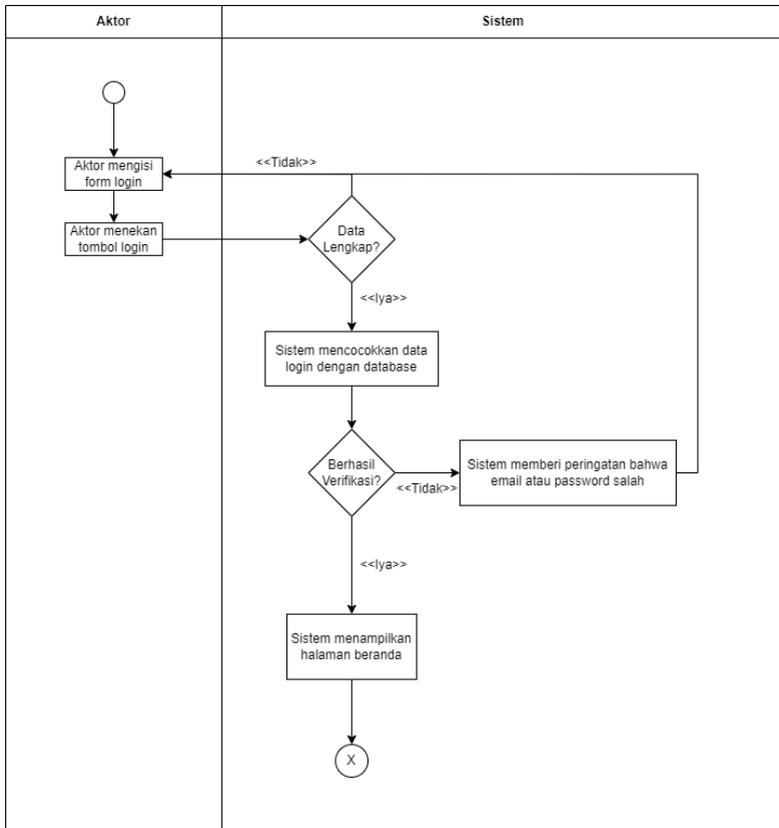
Gambar 4.4 merupakan diagram aktivitas untuk *use case* Mendaftarkan Akun (UC001).



Gambar 4.4 Diagram Aktivitas Mendaftarkan Akun

b. Masuk ke Aplikasi (Login)

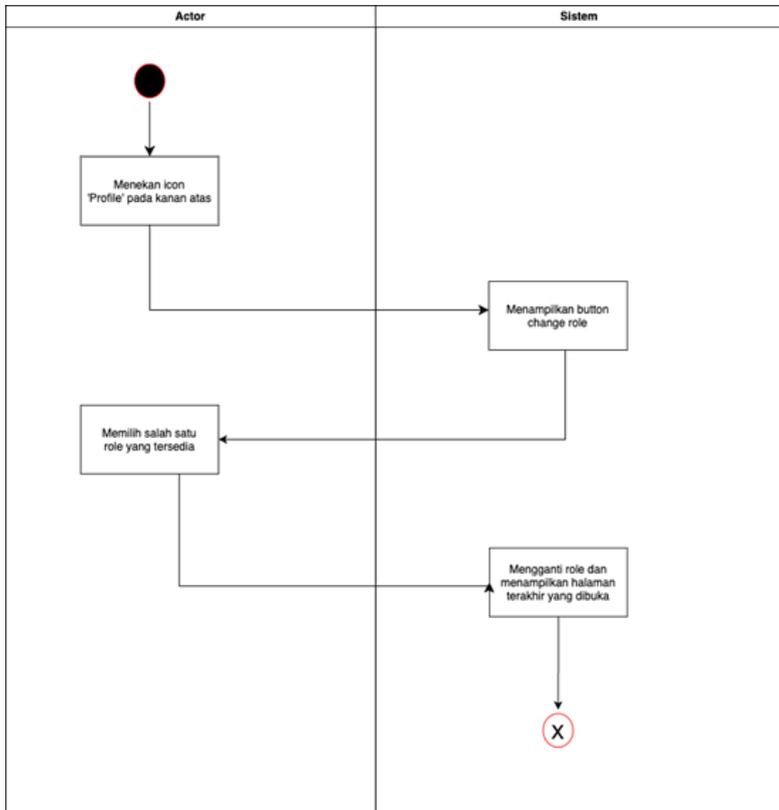
Gambar 4.5 merupakan diagram aktivitas dari *use case* Masuk ke Aplikasi (Login) (UC002).



Gambar 4.5 Diagram Aktivitas Login

c. Menambah Resep

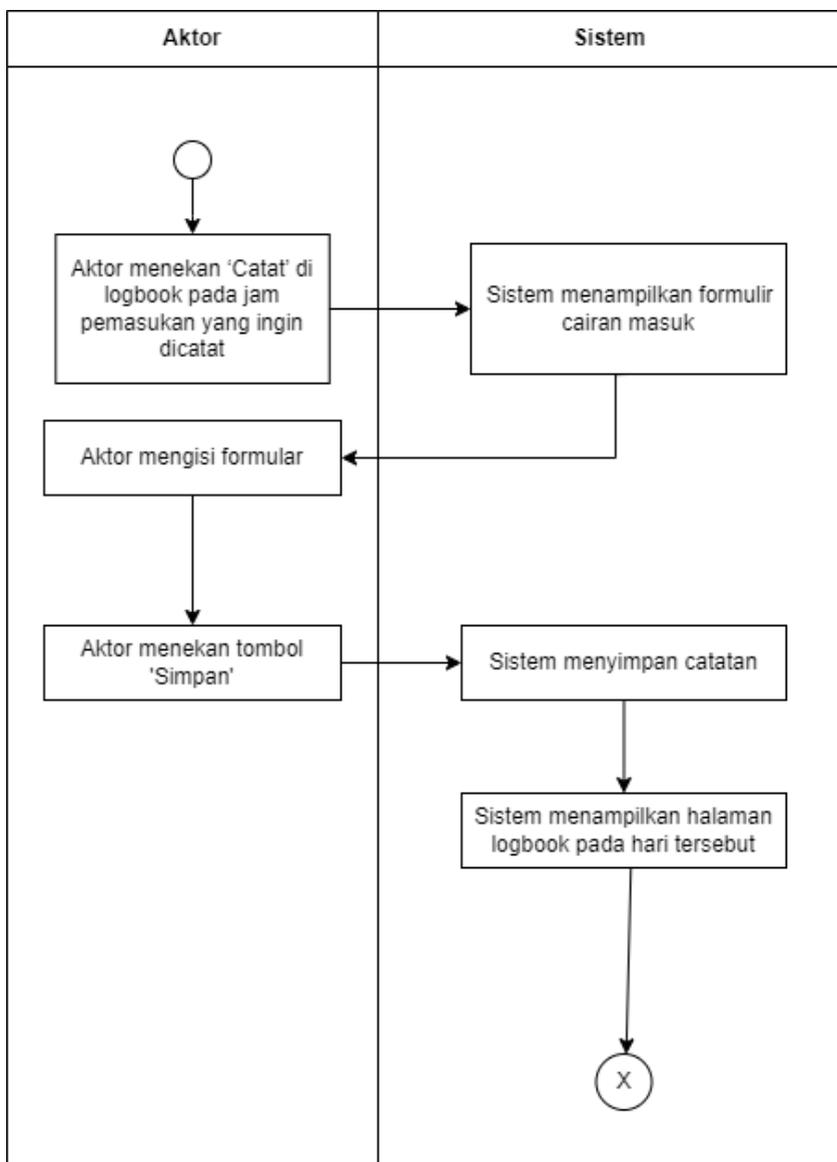
Gambar 4.6 merupakan diagram aktivitas dari *use case* Menambah Resep (UC003).



Gambar 4.6 Diagram Aktivitas Menambah Resep

d. Mencatat Pemasukan Cairan

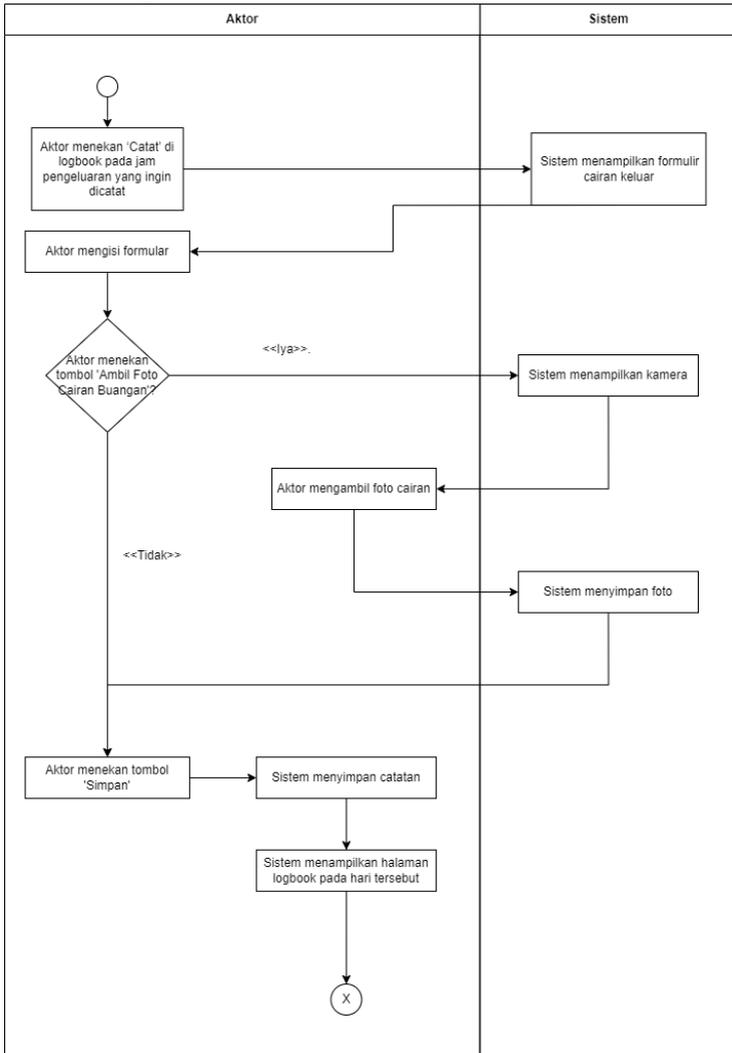
Gambar 4.7 merupakan diagram aktivitas dari *use case* Mencatat Pemasukan Cairan (UC004).



Gambar 4.7 Diagram Aktivitas Mencatat Pemasukan Cairan

e. Mencatat Pengeluaran Cairan

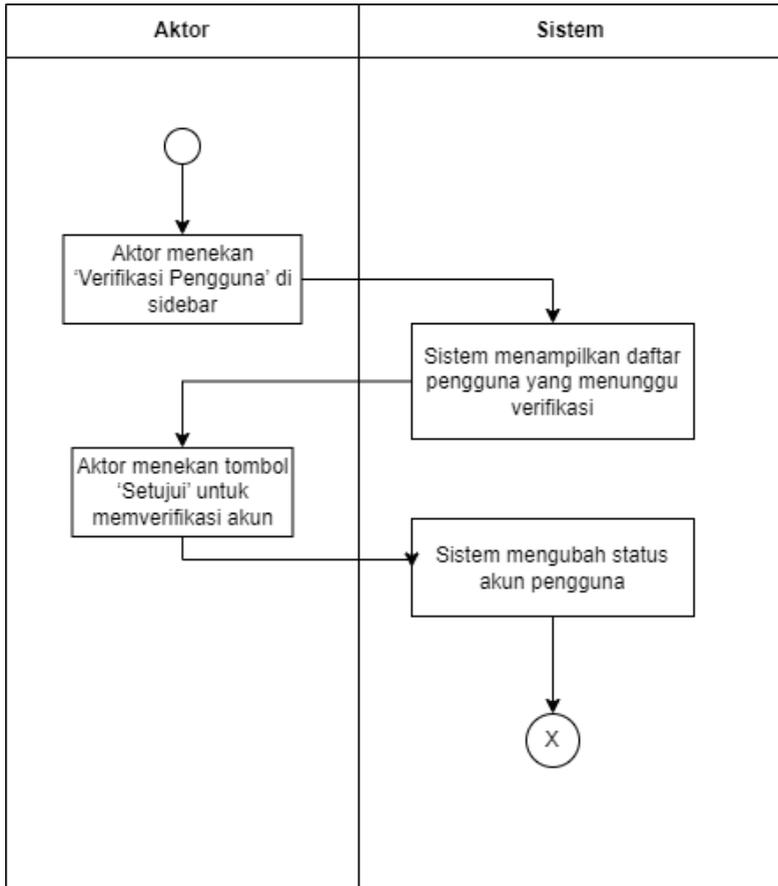
Gambar 4.8 merupakan diagram aktivitas dari *use case* Mencatat Pengeluaran Cairan (UC005).



Gambar 4.8 Diagram Aktivitas Mencatat Pengeluaran Cairan

f. Memverifikasi Pengguna

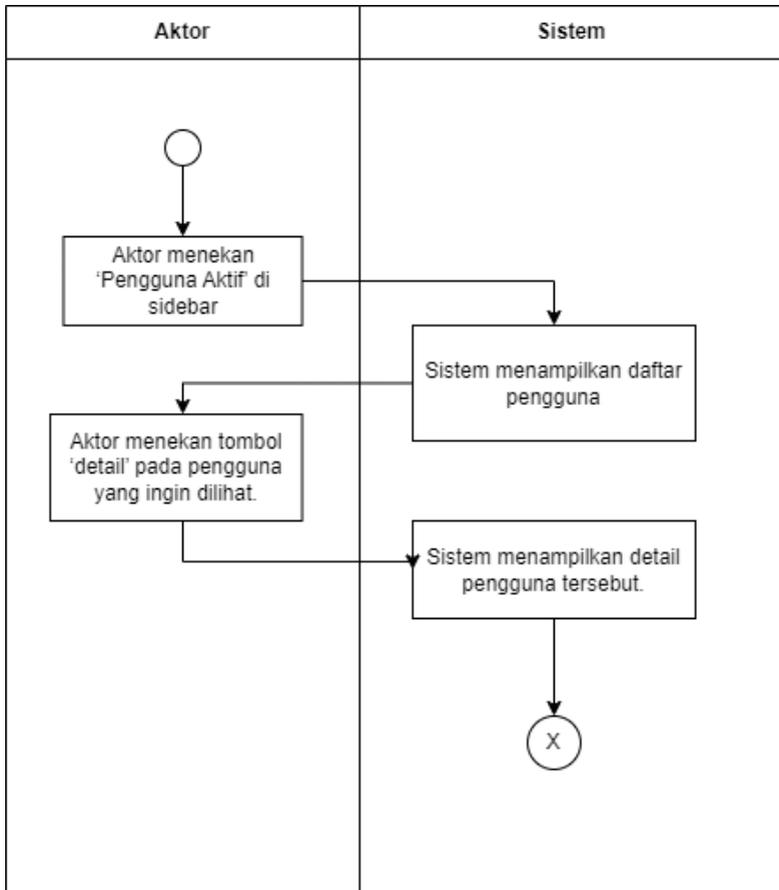
Gambar 4.9 merupakan diagram aktivitas dari *use case* Memverifikasi Pengguna (UC101).



Gambar 4.9 Diagram Aktivitas Memverifikasi Pengguna

g. Melihat Detail Pengguna

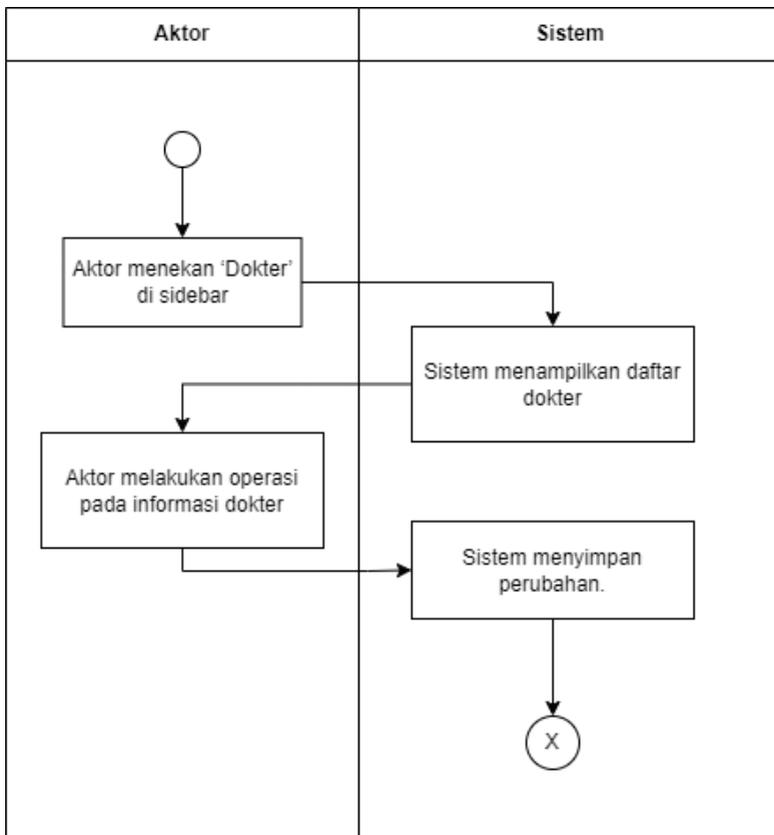
Gambar 4.10 merupakan diagram aktivitas dari *use case* Melihat Detail Pengguna (UC102).



Gambar 4.10 Diagram Aktivitas Melihat Detail Pengguna

h. Mengelola Informasi Dokter

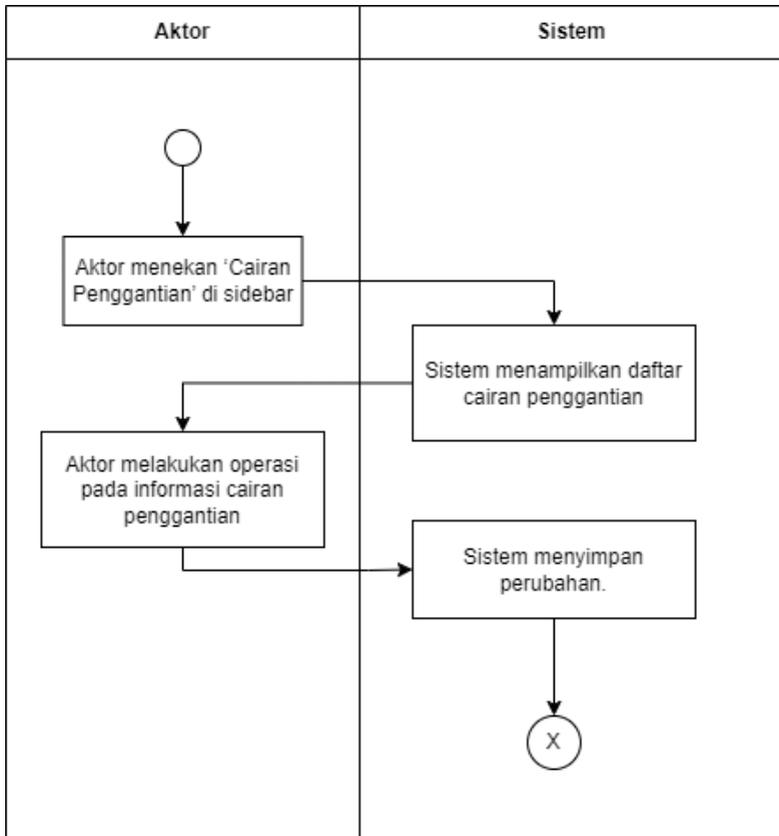
Gambar 4.11 merupakan diagram aktivitas dari *use case* Mengelola Informasi Dokter (UC103).



Gambar 4.11 Diagram Aktivitas Mengelola Informasi Dokter

i. Mengelola Informasi Cairan Penggantian

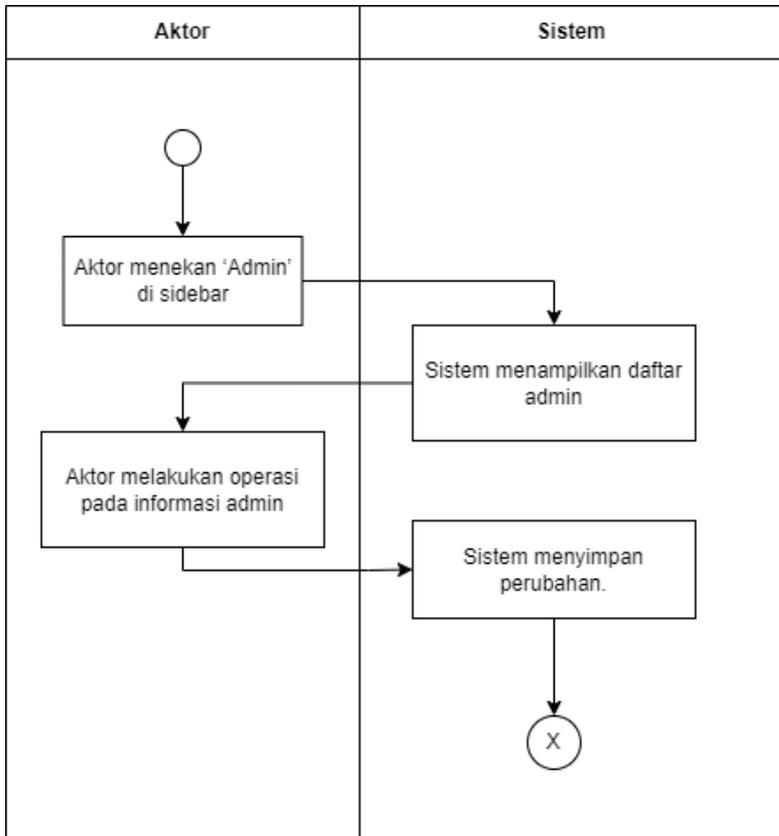
Gambar 4.12 merupakan diagram aktivitas dari *use case* Mengelola Informasi Cairan Penggantian (UC104).



Gambar 4.12 Diagram Aktivitas Mengelola Informasi Cairan Penggantian

j. Mengelola Informasi Admin

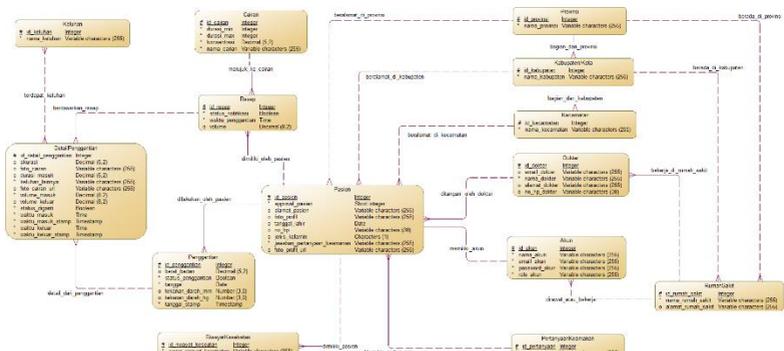
Gambar 4.13 merupakan diagram aktivitas dari *use case* Mengelola Informasi Admin (UC201).



Gambar 4.13 Diagram Aktivitas Mengelola Informasi Admin

4.7. Conceptual Data Model

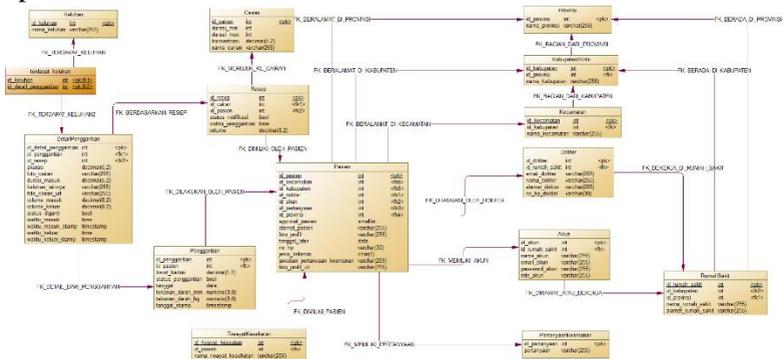
Gambar 4.14 adalah *conceptual data model* (CDM) dari aplikasi Sahabat CAPD.



Gambar 4.14 CDM Aplikasi Sahabat CAPD

4.8. Physical Data Model

Gambar 4.15 adalah *physical data model* (PDM) dari aplikasi Sahabat CAPD.



Gambar 4.15 PDM Aplikasi Sahabat CAPD

4.9. Deskripsi Data

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

a. Akun Pengguna

Tabel 4.15 adalah deskripsi data untuk tabel Akun 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 4.16 adalah deskripsi data untuk tabel Pasien.

Tabel 4.16 Deskripsi Data pada Tabel Pasien Aplikasi CAPD

Pasien		
No	Nama Atribut	Tipe Data
1.	id_pasien	Integer
2.	approval_pasien	Boolean
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 4.17 adalah deskripsi data untuk tabel Rumah Sakit.

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 4.18 adalah deskripsi data untuk tabel Dokter.

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 4.19 adalah deskripsi data untuk tabel Riwayat Kesehatan.

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 4.20 adalah deskripsi data untuk tabel Cairan.

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 4.21 adalah deskripsi data untuk tabel Resep.

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 4.22 adalah deskripsi data untuk tabel Penggantian.

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 4.23 adalah deskripsi data untuk tabel Detail Penggantian.

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	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 4.24 adalah deskripsi data untuk tabel Pertanyaan Keamanan.

Tabel 4.24 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 4.25 adalah deskripsi data untuk tabel Provinsi.

Tabel 4.25 Deskripsi Data pada Tabel Provinsi Aplikasi CAPD

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

l. Kabupaten/Kota

Tabel 4.26 adalah deskripsi data untuk tabel Kabupaten/Kota.

Tabel 4.26 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 4.27 adalah deskripsi data untuk tabel Kecamatan.

Tabel 4.27 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 V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari perancangan sistem dan pengaplikasian sistem dalam bentuk situs web.

5.1. Implementasi Design System

Sistem yang dibuat merupakan sistem untuk mencatat penggantian cairan CAPD yang ditujukan untuk pasien rumah sakit. Adapun fitur-fitur dari aplikasi Sahabat CAPD adalah mencatat penggantian cairan CAPD (termasuk cairan masuk dan cairan keluar), mencatat tekanan darah dan berat badan, menambahkan resep penggantian, serta mencatat riwayat kesehatan. Di sisi admin terdapat fitur-fitur untuk melakukan monitoring setiap pasien, seperti melihat detail perkembangan pasien, menambahkan dokter dan pasien, dan menambahkan cairan.

Aplikasi ini dibuat dengan menggunakan Next Js, Tailwind CSS, dan beberapa library pendukung lainnya seperti Radix UI dan Shadcn. Dalam proses pengambilan data digunakan Axios.

Selanjutnya, akan dijelaskan tentang penerapan design system yang telah dibuat oleh tim UI/UX.

5.1.1. Mobile App (PWA)

Mobile App untuk aplikasi Sahabat CAPD ditujukan kepada pasien rumah sakit. Aplikasi ini berbentuk *progressive web app* (PWA) sehingga aplikasi ini berbasis *website* tetapi bisa diakses sebagai aplikasi *mobile*. Aplikasi ini memiliki antarmuka yang sederhana dan mudah untuk digunakan. Desain dari antarmuka ini dibentuk menggunakan aplikasi Figma lalu diimplementasikan menggunakan *framework* Next Js dengan *styling* kustom menggunakan Tailwind CSS dan Radix UI. Berikut

adalah konfigurasi Tailwind yang menerapkan *design system* dari desain Figma.

Konfigurasi tailwind pada aplikasi Sahabat CAPD ini mengatur desain yang digunakan. Konfigurasi ini terletak pada dua file, yaitu *tailwind.config.ts* dan *src/app/globals.css*. Penjelasan konfigurasi yang terdapat pada kedua file tersebut sebagai berikut.

a. File *tailwind.config.ts*

Pada file ini terdapat konfigurasi untuk membuat kelas-kelas utilitas kustom yang menyesuaikan desain Figma. Kelas-kelas utilitas ini digunakan untuk properti tertentu seperti font, warna border, bayangan, transisi, dan animasi. Pada potongan Kode Semu 5.1, dapat dilihat bahwa konfigurasi disimpan pada variabel konstan bernama *config*. File-file dan direktori yang akan diproses oleh tailwind dapat dikonfigurasi pada atribut *content*. Kelas-kelas utilitas ditambahkan pada bagian *extend* di atribut *theme*.

```
3  const config = {
4    darkMode: ["class"],
5    content: [
6      "./pages/**/*.{ts,tsx}",
7      "./components/**/*.{ts,tsx}",
8      "./app/**/*.{ts,tsx}",
9      "./src/**/*.{ts,tsx}"
10   ],
11   prefix: "",
12   theme: {
13     container: {
...     ...
20     extend: {
```

Kode Semu 5.1 Konfigurasi file *tailwind.config.ts* (Bagian 1)

```

21     fontFamily: {
22         manrope: ["Manrope", "sans-serif"]
23     },
24     colors: {
...         ...
98     },
99     borderRadius: {
...         ...
103    },
104    boxShadow: {
...         ...
109    },
110    lineHeight: {
...         ...
112    },
113    keyframes: {
...         ...
140    },
141    animation: {
...         ...
147    }
148    }
149    },
150    plugins: [require("tailwindcss-animate")]
151 } satisfies Config;
152
153 export default config;

```

Kode Semu 5.1 Konfigurasi file tailwind.config.ts (Bagian 2)

Dalam mengatur sistem desain, salah satu komponen

penting adalah pemilihan warna yang digunakan. Pada file *tailwind.config.ts* terdapat salah satu konfigurasi *colors* pada baris 24 untuk mengatur warna tersebut. Sebagai contoh, pada desain Figma diatur bahwa terdapat pewarnaan yang bernama “primary” dengan beberapa tingkat. Untuk menambah warna tersebut, kami menambahkan objek *primary* dengan tingkat-tingkatnya beserta kode heks dari masing-masing warna. Hasil dari konfigurasi tersebut adalah kelas-kelas seperti “text-primary-300”, “bg-primary-200”, dsb. Contoh tersebut dapat dilihat pada Kode Semu 5.2 berikut.

```
24     colors: {
...     ...
30     primary: {
31         DEFAULT: "hsl(var(--primary))",
32         foreground: "hsl(var(--primary-
33         foreground))",
33         50: "#FBEBED",
34         75: "#EEABB4",
35         100: "#E78895",
36         200: "#DD5567",
37         300: "#D63248",
38         400: "#962332",
39         500: "#831F2C",
40         600: "#57383C"
41     },
...     ...
```

Kode Semu 5.2 Konfigurasi warna di file tailwind.config.ts

Untuk membuat kelas utilitas properti lain seperti *borderRadius* dan *boxShadow* dilakukan hal serupa dengan penambahan warna, yaitu menambahkan nama dari tipe

propertinya dengan konfigurasinya. Selain itu, untuk mengatur ukuran pada ukuran-ukuran layar tertentu bisa menggunakan atribut “lg”, “md”, dan “sm” yang diikuti dengan ukurannya. Kode Semu 5.3 menunjukkan konfigurasi `borderRadius` dan `boxShadow` di file `tailwind.config.ts` yang terdapat pada baris 99 hingga 109.

```
99     borderRadius: {
100         lg: "var(--radius)",
101         md: "calc(var(--radius) - 2px)",
102         sm: "calc(var(--radius) - 4px)"
103     },
104     boxShadow: {
105         clickable:
106             "0px 1px 3px 0px rgba(0, 0, 0, 0.30), 0px
107             4px 8px 0px rgba(0, 0, 0, 0.15)",
108         button:
109             "0px 1px 3px 0px rgba(0, 0, 0, 0.30), 0px
110             4px 8px 3px rgba(0, 0, 0, 0.15);"
111     },
```

Kode Semu 5.3 Konfigurasi kelas utilitas di file `tailwind.config.ts`

Pada file ini terdapat juga tambahan atau *plugin* yang memungkinkan penambahan konfigurasi berupa animasi. *Plugin* ini bernama “`tailwindcss-animate`” yang diaktifkan pada baris 150. Untuk membuat konfigurasi animasi terdapat *keyframes* dan *animation*. Atribut *keyframes* mengatur tahap animasi menggunakan *style* di titik-titik tertentu (seperti awal dan akhir) dari rangkaian animasi. Sedangkan, atribut *animation* mengatur durasi dan fungsi waktu untuk *keyframe* yang digunakan. Konfigurasi animasi dapat dilihat pada Kode Semu 5.4.

```

113     keyframes: {
114         "accordion-down": {
115             from: { height: "0" },
116             to: { height: "var(--radix-accordion-
content-height)" }
117         },
118         "accordion-up": {
119             from: { height: "var(--radix-accordion-
content-height)" },
120             to: { height: "0" }
121         },
...     ...
140     },
141     animation: {
142         "accordion-down": "accordion-down 0.2s
ease-out",
143         "accordion-up": "accordion-up 0.2s ease-
out",
...     ...
147     }

```

Kode Semu 5.4 Konfigurasi animasi di file tailwind.config.ts

b. File `src/app/globals.css`

File ini mengatur konfigurasi dasar seperti menambahkan font, variabel, *style* khusus bagi elemen-elemen tertentu, dan penambahan kelas yang hanya bisa dikonfigurasi pada file css. File ini dibagi dalam beberapa *directive*, yaitu perintah-perintah yang dimulai dengan simbol “@”. *Directive* pada baris-baris pertama digunakan oleh tailwind untuk menambahkan kelas-kelas utilitas. *Directive* selanjutnya adalah “font-face” yang digunakan untuk menambahkan font baru. Konfigurasi tersebut dapat dilihat pada potongan Kode Semu 5.5.

```

1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  @font-face {
6      font-family: "Manrope";
7      src: url("../assets/fonts/manrope/manrope.ttf")
      format("truetype");
8  }

```

Kode Semu 5.5 Konfigurasi directive tailwind dan font-face di file src/app/globals.css

Directive yang digunakan selanjutnya adalah “layer” atau lapisan untuk membagikan kustomisasi *style* menjadi lapisan “base” dan “utilities”. Lapisan “base” digunakan untuk mendefinisikan *style* dan variabel untuk elemen-elemen yang ditentukan. Contohnya yaitu pada elemen “input” yang bertipe “time” atau “date”. Semua kemunculan elemen tersebut akan diberikan *style* kustom yang dapat dilihat pada Kode Semu 5.6.

```

74 @layer base {
...   ...
89   input[type="time"],
90   input[type="date"] {
91       position: relative;
92       z-index: 1;
93   }
94
95   input[type="time"]::-webkit-calendar-picker-
indicator,

```

Kode Semu 5.6 Konfigurasi directive layer base di file src/app/globals.css (Bagian 1)

```

96     input[type="date"]::-webkit-calendar-picker-
      indicator {
97         display: block;
98         top: 0;
99         right: 0;
100        height: 100%;
101        width: 100%;
102        position: absolute;
103        background: transparent;
104    }
105 }

```

Kode Semu 5.6 Konfigurasi directive layer base di file src/app/globals.css (Bagian 2)

Lapisan “utilities” digunakan untuk menambahkan kelas utilitas kustom melalui file ini. Lapisan ini digunakan untuk menambahkan kelas utilitas yang tidak dapat ditambahkan melalui file *tailwind.config.ts*. Contohnya adalah kelas “no-scrollbar” yang menghilangkan *scroll bar* pada suatu elemen. Potongan kode tersebut dapat dilihat di Kode Semu 5.7 baris 107 s.d. 116.

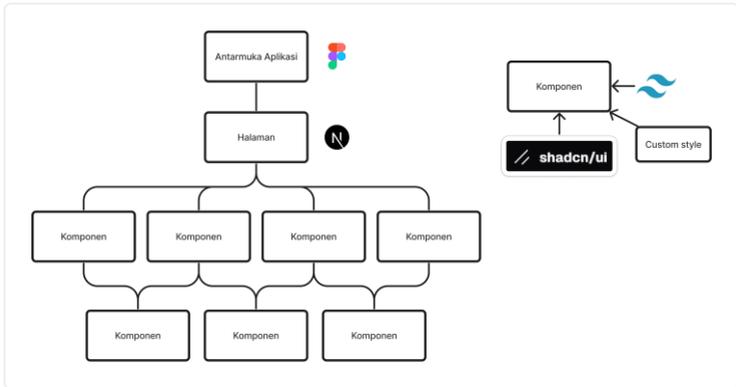
```

107 @layer utilities {
108     .no-scrollbar::-webkit-scrollbar {
109         display: none;
110     }
111
112     .no-scrollbar {
113         -ms-overflow-style: none;
114         scrollbar-width: none;
115     }
116 }

```

Kode Semu 5.7 Konfigurasi directive layer utilities di file src/app/globals.css

5.1.2. Dashboard Admin



Gambar 5.1 Design System Dashboard Admin

Gambar di atas adalah gambaran umum mengenai *design system* yang akan digunakan. Aplikasi dipecah menjadi komponen – komponen terkecil yang saling berhubungan membentuk komponen yang lebih besar. Di dalam setiap komponen, terdapat *syling* menggunakan gabungan dari *framework* tailwind css, shadcn, dan *styling* yang sudah dikustomisasi sesuai komponen tersebut.

```
1 ...
2     primary: {
3     ...
4         50: "#FBEBED",
5         75: "#EEABB4",
6         100: "#E78895",
7         200: "#DD5567",
8         300: "#D63248",
```

Kode Semu 5.9 Konfigurasi directive layer utilities di file *tailwind.config.css* (Bagian 1)

```

9           400: "#962332",
10          500: "#831F2C",
11          600: "#57383C",
12          },
13          secondary: {
14      ...
15          50: "#e9faff",
16          75: "#a5e9ff",
17          100: "#7fe0ff",
18          200: "#48d2ff",
19          300: "#23c9ff",
20          400: "#198db3",
21          500: "#157b9c",
22          },
23      ...
24          neutrals: {
25          20: "#F7F5F5",
26          50: "#A9999B",
27          60: "#BFB4B5",
28          70: "#B5A8A9",
29          100: "#907C7E",
30          200: "#846D70",
31          300: "#785F62",
32          400: "#6D5256",
33          500: "#57383C",
34          900: "#320C11",
35          },
36          success: {
37          300: "#1B998B",

```

Kode Semu 5.9 Konfigurasi directive layer utilities di file tailwind.config.css (Bagian 2)

```

38     },
39     danger: {
40         300: "#931621",
41         500: "#614448",
42         600: "#BFB4B5",
43         900: "#320C11",
44     },
45     ...

```

Kode Semu 5.9 Konfigurasi directive layer utilities di file tailwind.config.css (Bagian 3)

```

1     ...
2     .PopoverContent {
3         width: var(--radix-popover-trigger-width);
4         max-height: var(--radix-popover-content-available-
5         height);
6     }

```

Kode Semu 5.10 Konfigurasi directive layer utilities di file src/app/globals.css

File tailwind.config.ts dan globals.css merupakan tempat untuk menyimpan konfigurasi *design system* aplikasi. Pada Kode Semu 5.9 ditambahkan konfigurasi warna menyesuaikan kode warna pada *design system* aplikasi dan kode *default* dari Radix UI dan Shadcn, yaitu kode program titik tiga (...). Pada Kode Semu 5.10 ditambahkan kode *custom* untuk menyesuaikan lebar *popup* data *dropdown*.

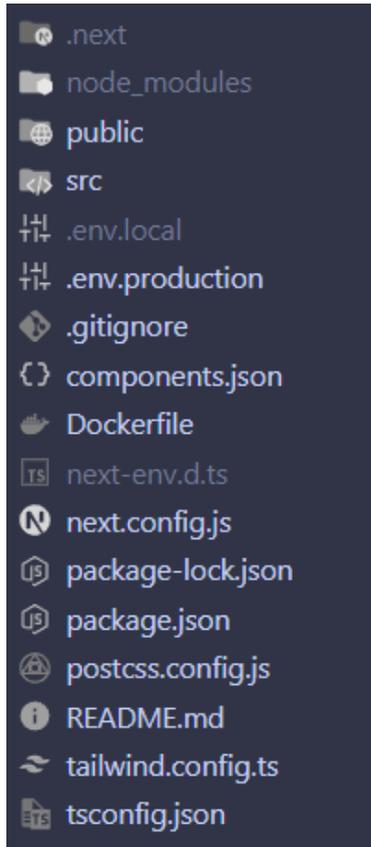
5.2. Struktur Folder Aplikasi

Berikut adalah struktur folder yang digunakan dalam implementasi sistem.

5.2.1. Mobile App (PWA)

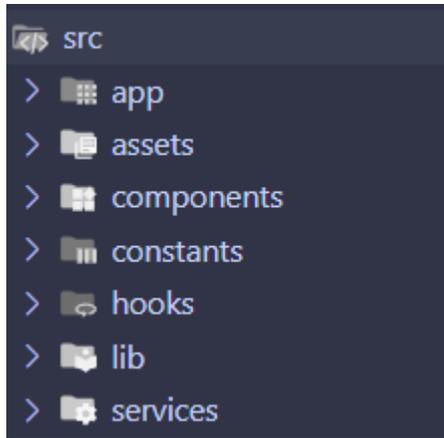
Pada *mobile app*, struktur folder mengikuti struktur folder yang ditetapkan oleh *framework* Next Js, khususnya versi Next Js

13. Struktur file dan folder pada bagian luar dapat dilihat pada gambar berikut.



Gambar 5.2 Struktur Folder Mobile App (PWA)

Halaman, komponen, asset, dan file-file yang digunakan tersimpan pada folder utama, yaitu folder *src*. Gambaran dari struktur folder *src* sebagai berikut.

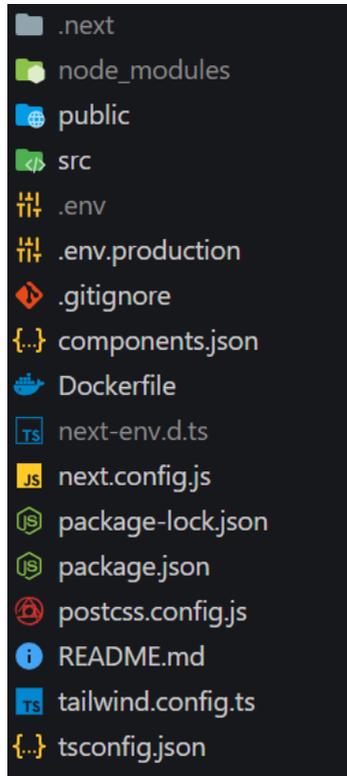


Gambar 5.3 Isi Folder src pada Mobile App (PWA)

Penjelasan masing-masing folder pada folder *src* sebagai berikut.

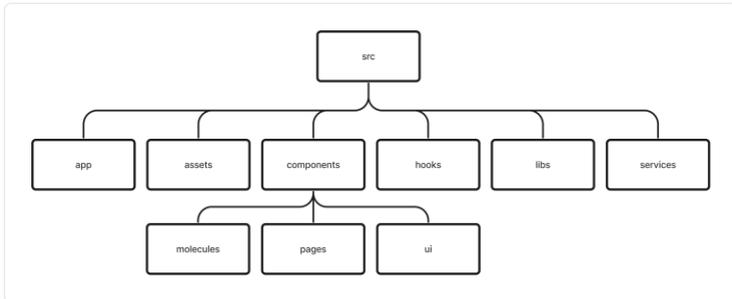
- Folder *app* digunakan untuk halaman dan *routing* aplikasi
- Folder *assets* berisi gambar dan font yang digunakan untuk aplikasi
- Folder *components* berisi komponen-komponen yang digunakan dalam halaman aplikasi
- Folder *constants* berisi data yang tidak akan berubah
- Folder *hooks* berisi *react hooks*, yaitu fungsi yang digunakan untuk menyimpan *state* suatu data
- Folder *lib* berisi file-file untuk menyimpan konfigurasi API, token, validasi form, tipe data, dan fungsi-fungsi kustom yang diperlukan
- Folder *services* berisi fungsi-fungsi untuk memanggil API yang dapat digunakan pada halaman aplikasi

5.2.2. *Dashboard Admin*



Gambar 5.4 Struktur Folder Dashboard Admin

Gambar di atas merupakan struktur folder aplikasi pada bagian paling luar. Seluruh konfigurasi aplikasi yang lebih detail berada di dalam folder src.



Gambar 5.5 Isi folder src Dashboard Admin

Di dalam folder src terdapat beberapa folder yang dikelompokkan berdasarkan kegunaannya.

- App digunakan untuk membuat halaman dan *routing* aplikasi
- Assets digunakan untuk menyimpan gambar dan *font* aplikasi
- Components untuk menyimpan komponen – komponen yang digunakan di setiap halaman aplikasi. Di dalam folder ini terdapat tiga komponen utama yaitu ui untuk menyimpan bagian terkecil aplikasi seperti button, card, dialog, dan lainnya, molecules untuk menyimpan gabungan dari beberapa ui dengan *styling* sesuai kebutuhan, dan pages untuk menyimpan gabungan molecules dan ui di dalam halaman tersebut.
- Hooks digunakan untuk menyimpan fungsi aplikasi untuk menyimpan data yang dapat digunakan di semua halaman.
- Libs digunakan untuk menyimpan konfigurasi API, token, validasi form, dan fungsi – fungsi kustom yang diperlukan.
- Services digunakan untuk menyimpan fungsi – fungsi pemanggilan API sesuai dengan halaman aplikasi.

5.3. Implementasi Pengambilan Data API

Dalam pengambilan data, aplikasi CAPD memanfaatkan pemanggil API dari *backend*. Berikut adalah penjelasan cara pengambilan data dari aplikasi *Mobile App* dan aplikasi Dashboard Admin.

5.3.1. *Mobile App (PWA)*

Pengambilan data API pada *Mobile App* memanfaatkan fungsi *callAPI* sebagai fungsi dasar untuk mengirim permintaan API. Fungsi ini mengambil beberapa argumen sebagai berikut.

1. *method*, untuk menentukan jenis metode pemanggilan yang ingin digunakan. Opsinya yaitu “GET”, “POST”, “DELETE”, “PATCH”, dan “PUT”
2. *url*, untuk menentukan URL dari API yang dipanggil
3. *data*, untuk menentukan data yang dikirim bersama dengan pemanggilan API
4. *token*, untuk otorisasi pemanggilan API
5. *config*, yaitu konfigurasi tambahan yang diperlukan

Fungsi ini terdapat pada *src/lib/api/index.ts* pada baris 11 s.d. 38. Potongan kode fungsi *callAPI* terdapat pada Kode Semu 5.11.

```
11  const callAPI = async ({ method, url, data, token,
12    config }: Props) => {
13    try {
14      const headers: any = {
15        "Content-Type": "application/json",
16        ...config?.headers,
17      };
18      if (token) {
```

Kode Semu 5.11 Fungsi callApi di file src/lib/api/index.ts (Bagian 1)

```

18     headers["Authorization"] = `Bearer ${token}`;
19   }
20   const response = await axios({
21     method,
22     url,
23     data,
24     ...config,
25     headers,
26   });
27   const res = await response.data;
28   return res;
29 } catch (error: any) {
30   return {
31     error: 1,
32     status: error.status,
33     message: error.response?.data.message ||
34     "Error",
35     errorsData: error.response?.data.errors ||
36     null,
37     data: null,
38   };

```

Kode Semu 5.11 Fungsi callApi di file src/lib/api/index.ts (Bagian 2)

Fungsi *callAPI* ini digunakan dengan membuat suatu *service* baru di direktori *src/services*. *Service* tersebut merupakan fungsi yang mengirim permintaan pada API dan mengembalikan data atau hasil dari pemanggilan API tersebut. Salah satu contoh *service* yang digunakan adalah fungsi untuk mengambil data semua detail penggantian. Argumen-argumen yang diperlukan

dimasukkan ke dalam fungsi *callAPI*, lalu hasilnya disimpan dalam variabel *res* dan dikembalikan. *Service* tersebut terletak pada *src/services/detail-penggantian/index.ts*. Potongan kode *service* tersebut sebagai berikut.

```
6 export async function
  getAllDetailPenggantian(token: string) {
7   const url = `${URL}/api/v1/detail-penggantian`;
8   const res = await callAPI({
9     url,
10    method: "GET",
11    token,
12  });
13
14  return res as
  PaginatedApiResponse<DetailPenggantianType>;
15 }
```

Kode Semu 5.12 Fungsi callApi di file src/services/detail-penggantian/index.ts

5.3.2. *Dashboard Admin*

```
1 ...
2 const api = axios.create({
3   baseURL:
4     `${process.env.NEXT_PUBLIC_API_URL}/api/v1` ||
5     "http://localhost:8000/api/v1",
6   timeout: 10000,
7   headers: { "Content-Type": "application/json" },
8   withCredentials: false,
9 });
10 ...
```

Kode Semu 5.13 Konfigurasi utama API di file src/lib/api/index.ts (Bagian 1)

```

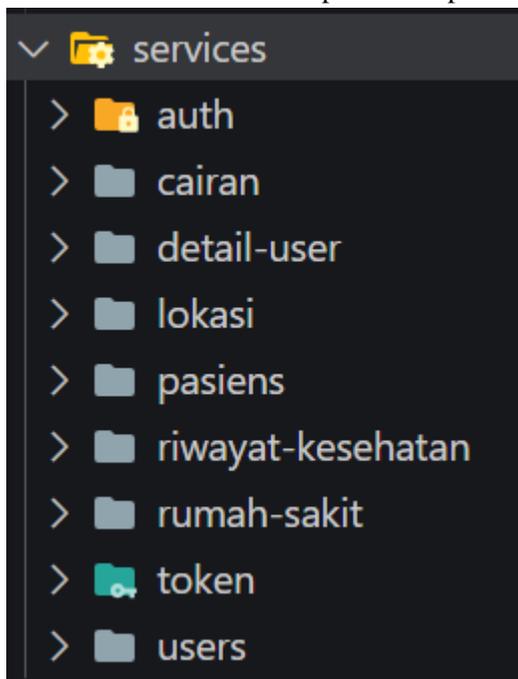
11  const callAPI = async ({ method, url, data, token,
    config }: Props) => {
12    try {
13      const headers: any = {
14        "Content-Type": "application/json",
15        ...config?.headers,
16      };
17      if (token) {
18        headers["Authorization"] = `Bearer ${token}`;
19      }
20      const response = await axios({
21        method,
22        url,
23        data,
24        ...config,
25        headers,
26      });
27      const res = await response.data;
28      return res;
29    } catch (error: any) {
30      return {
31        error: 1,
32        status: error.status,
33        message: error.response?.data.message ||
34        "Error",
35        errorsData: error.response?.data.errors ||
36        null,
37        data: null,
38      };
39    }
40  };

```

Kode Semu 5.13 Konfigurasi utama API di file src/lib/api/index.ts

(Bagian 2)

Pada Kode Semu 5.13 ini terdapat dua fungsi utama yaitu api dan callAPI. Api adalah fungsi untuk pemanggilan api dengan menuliskan langsung url yang diperlukan. Fungsi ini berfungsi untuk mengambil data tabel. Selanjutnya, fungsi callAPI digunakan untuk melakukan pengambilan data yang lebih detail seperti pengambilan data detail pasien dan detail penggantian cairan. Fungsi callAPI juga digunakan untuk melakukan pengambilan data sesuai kebutuhan yang terdapat di folder src/services. Isi dari folder tersebut dapat dilihat pada Gambar 5.5.



Gambar 5.5 Isi folder services pada Dashboard Admin

Di dalam folder services, terdapat beberapa folder yang dikelompokkan berdasarkan data yang dibutuhkan setiap halaman, contohnya folder auth digunakan untuk pengambilan API untuk

proses login aplikasi, cairan digunakan untuk proses CRUD cairan, dan pasien untuk proses CRUD data pasien. Kode Semu 5.14 adalah contoh penerapan API pada cairan di file *src/services/cairan/index.ts*.

```
1  export async function getAllCairan({
2    token,
3    queryParams,
4  }): {
5    token: string;
6    queryParams?: QueryParamsType;
7  } {
8    let params = "";
9    if (queryParams) params =
10     convertQueryParamsToURL(queryParams);
11
12     const url = `${URL}/api/v1/cairan${params &&
13     `?${queryParams}`}`;
14     const res = await callAPI({
15       url,
16       method: "GET",
17       token,
18     });
19     return res;
20   }
```

*Kode Semu 5.15 Penerapan API pada cairan di file
src/services/cairan/index.ts*

Pada file *index.ts* terdapat beberapa fungsi yang menerapkan fungsi *callAPI* sesuai dengan kebutuhannya. Salah satu fungsi dalam file ini adalah *GetAllCairan* untuk mengambil data semua cairan yang tersedia. Data tersebut kemudian ditampilkan pada halaman cairan aplikasi.

5.4. Implementasi Antarmuka Pengguna

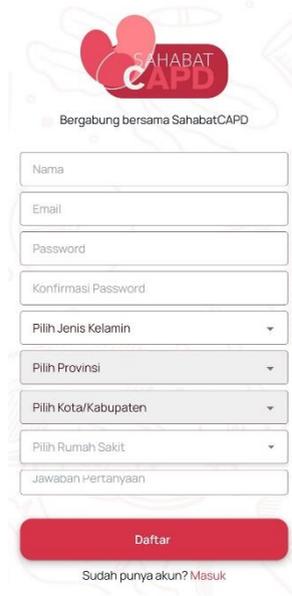
Desain aplikasi yang dibuat menggunakan Figma diimplementasikan pada aplikasi frontend. Berikut adalah implementasi desain pada *Mobile App* dan Dashboard Admin.

5.4.1. *Mobile App* (PWA)

Halaman-halaman web pada *mobile app* dapat diakses melalui *link* sesuai dengan *routing* Next js. Halaman yang terdapat pada *mobile app* memiliki antarmuka-antarmuka sebagai berikut.

a. Register

Halaman register digunakan untuk mendaftarkan akun baru. Tampilan dari halaman ini dapat dilihat pada Gambar 5.6.



SAHABAT
CAPD

Bergabung bersama SahabatCAPD

Nama

Email

Password

Konfirmasi Password

Pilih Jenis Kelamin

Pilih Provinsi

Pilih Kota/Kabupaten

Pilih Rumah Sakit

Jawaban Pertanyaan

Daftar

Sudah punya akun? [Masuk](#)

Gambar 5.6 Halaman Register Mobile App (PWA)

b. Login

Halaman login berisi form untuk melakukan login ke dalam aplikasi. Tampilan dari halaman login bisa dilihat pada

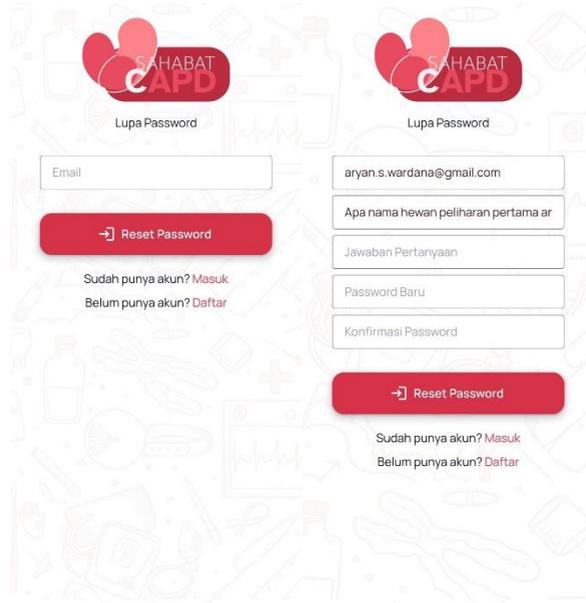
Gambar 5.7.



Gambar 5.7 Halaman Login Mobile App (PWA)

c. Lupa dan Reset Password

Halaman lupa dan reset password berisi form untuk mencari mengubah password yang telah dilupakan. Halaman ini bisa dilihat pada Gambar 5.8



*Gambar 5.8 Halaman lupa password Mobile App (Kiri).
Halaman reset password Mobile App (Kanan)*

d. Beranda

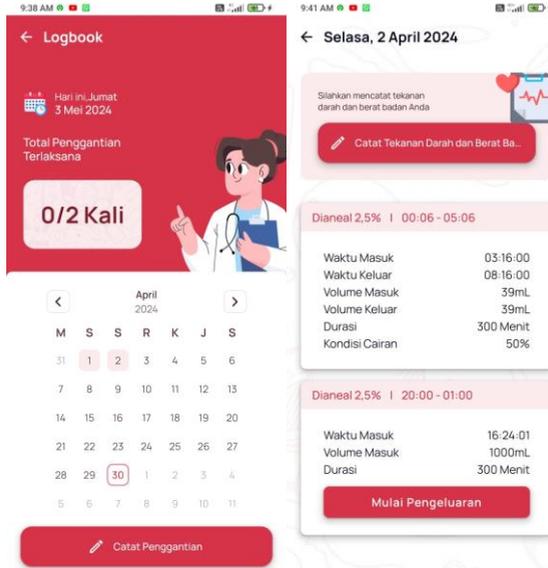
Halaman beranda aplikasi berisi menu-menu yang dapat dipilih, informasi terkait penggantian carian berikutnya, serta informasi dokter. Halaman ini dapat dilihat pada Gambar 5.9.



Gambar 5.9 Halaman Beranda Mobile App (PWA)

e. Logbook

Halaman yang berisi pencatatan penggantian cairan setiap hari. Halaman logbook dapat dilihat pada Gambar 5.10.



Gambar 5.10 Halaman Logbook Mobile App (PWA)

f. Pencatatan

Halaman untuk mencatat penggantian cairan yang dilakukan oleh pasien. Halaman ini dapat dilihat pada Gambar 5.11.

Catat Cairan Masuk

Jenis Cairan
Dianeal 2,5%

Volume Masuk (ml)
39

Waktu Masuk

Durasi Masuk (Menit)
Durasi Masuk

Catat Cairan Keluar

Jenis Cairan
Dianeal 2,5%

Volume Keluar (ml)
39

Waktu Keluar
8:16:00 AM

Ambil Foto Cairan Buangan

Keluhan

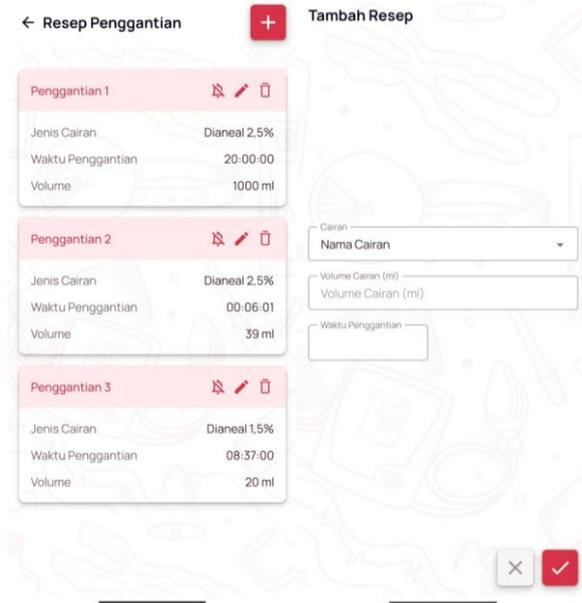
- Bengkak kaki/wajah
- Aliran masuk keluar tidak lancar
- Exit site kemerahan/bernanah/nyeri
- Lemas
- Nyeri perut hebat
- Diare

Keluhan Lain
Keluhan Lain

Gambar 5.11 Halaman Mencatat Penggantian Cairan Mobile App (PWA)

g. Resep Penggantian

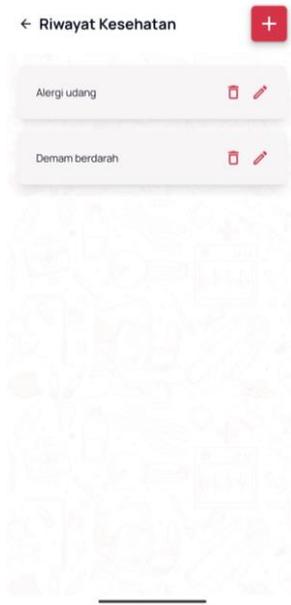
Halaman untuk menambahkan, mengubah, dan menghapus resep penggantian. Halaman ini dapat dilihat pada Gambar 5.12.



Gambar 5.12 Halaman Resep Penggantian Mobile App (PWA)

h. Riwayat Kesehatan

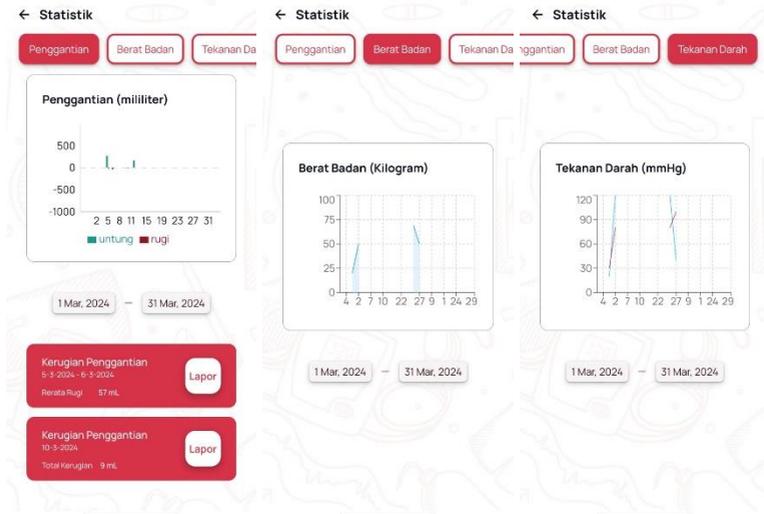
Halaman untuk mencatat riwayat kesehatan yang pernah dialami pasien. Halaman riwayat kesehatan bisa dilihat pada Gambar 5.13.



Gambar 5.13 Halaman Riwayat Kesehatan Mobile App (PWA)

i. Statistik

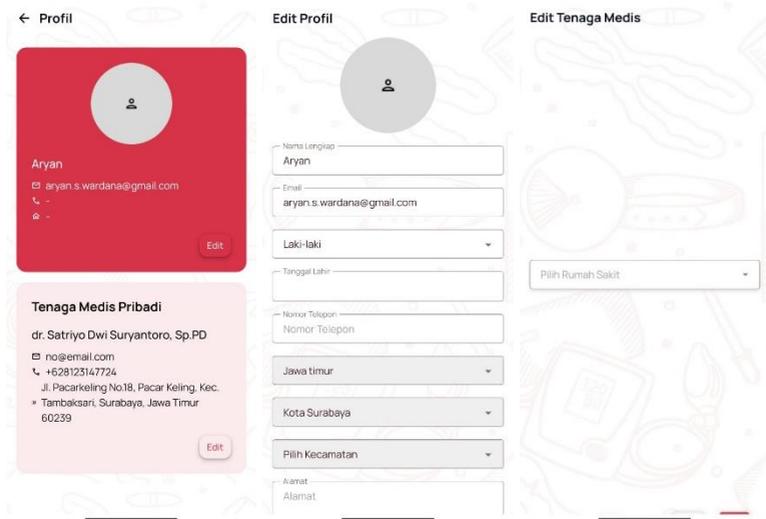
Halaman yang menyajikan informasi mengenai penggantian cairan, berat badan, dan tekanan darah. Halaman ini dapat dilihat pada Gambar 5.14.



Gambar 5.14 Halaman Statistik Mobile App (PWA)

j. Profil

Halaman yang berisi informasi terkait user dan dokter, serta form untuk mengubah informasi terkait user dan dokter tersebut. Halaman ini terdapat pada Gambar 5.15.



Gambar 5.15 Halaman Profil Mobile App (PWA)

k. Chatbot

Halaman untuk melayani pengguna jika terdapat pertanyaan mengenai kesehatan. Gambar 5.16 berisi tampilan dari halaman ini.



Gambar 5.16 Halaman Chatbot Mobile App (PWA)

1. Pengaturan

Halaman berisi pengaturan untuk mengaktifkan notifikasi dan mengubah password. Gambar 5.17 adalah tampilan dari halaman pengaturan.



Gambar 5.17 Halaman Pengaturan Mobile App (PWA)

5.4.2. Dashboard Admin

a. Login Aplikasi

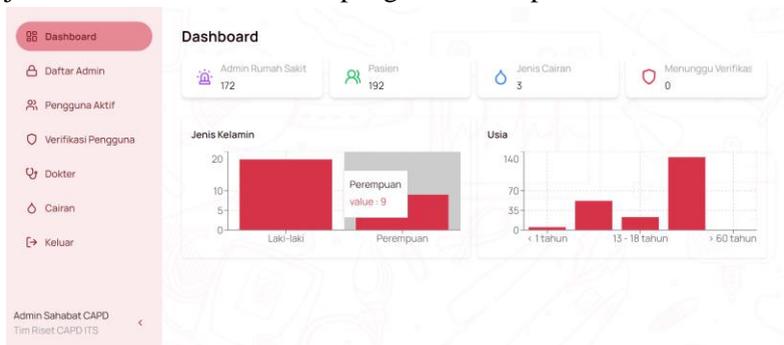
Halaman ini berisi form untuk melakukan login ke dalam aplikasi.



Gambar 5.18 Halaman Login Dashboard Admin

b. Dashboard

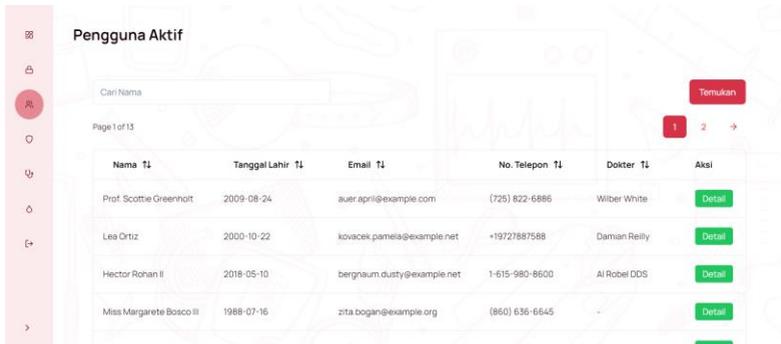
Halaman ini berisi tentang informasi umum mengenai aplikasi seperti jumlah admin rumah sakit, pasien yang terdaftar, jenis cairan. Selain itu, terdapat grafik data aplikasi.



Gambar 5.19 Halaman Beranda Dashboard Admin

c. Pengguna Aktif

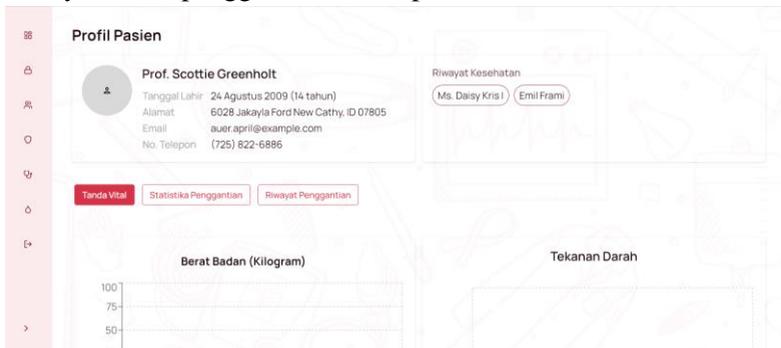
Halaman ini berisi informasi pengguna atau pasien yang sudah disetujui

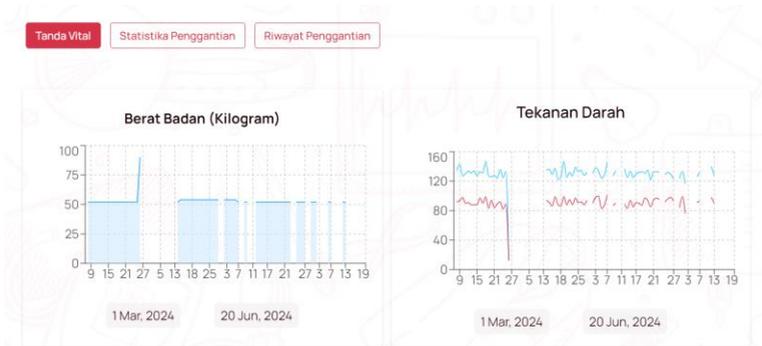


Gambar 5.20 Halaman Pengguna Aktif Dashboard Admin

d. Detail Pengguna

Halaman ini berisi detail pengguna atau pasien. Terdapat profil dan riwayat kesehatan pasien. Halaman ini juga menampilkan grafik berat badan, tekanan darah, dan informasi lainnya terkait penggantian cairan pasien.





Gambar 5.21 Halaman Profil Pasien Dashboard Admin

e. Verifikasi Pengguna

Halaman ini berisi pengguna atau pasien yang mendaftar di aplikasi mobile dan memerlukan persetujuan admin.

Nama TI	Tanggal Lahir TI	Email TI	No. Telepon TI	Dokter TI	Aksi
Mr. Sydney Williamson	1977-10-06	sandy59@example.net	1-650-290-2995	-	Tolak Setujui
Abe Rippin	2004-07-13	flatley.keely@example.com	975.239.3436	Zion Sipes	Tolak Setujui
Audrey Rofson	1975-08-08	nitzsche.izalah@example.org	1-432-620-8658	Jake Denesik PhD	Tolak Setujui
Dorothea Kemmer	2021-02-21	paul.olson@example.org	+1-463-624-3110	-	Tolak Setujui

Gambar 5.22 Halaman Verifikasi Pengguna Dashboard Admin

f. Dokter

Halaman ini berisi informasi dokter yang terdaftar di dalam aplikasi. Halaman ini juga dapat melakukan penambahan, penghapusan, dan penggantian data dokter.

Daftar Dokter

Cari Nama + Tambah Dokter

Page 1 of 4/71 1 2 →

Nama TI	Email TI	No. Telepon TI	Alamat TI	Aksi
Samanta Bernier V	farrell.demario@wolff.info	(571) 927-0122	295 Antone Shoal Apt. 721 North Kathleenfurt, AL 96649-4857	Hapus Edit
Miss Nayeli Mills	toymae@gmail.com	+1-602-479-2622	2959 Al Point Lake Marion, SC 68007-2873	Hapus Edit
Mrs. Leanna Daeh	eleazar63@yahoo.com	+1-209-629-1453	9547 Miller Ramp Christianamouth, OH 20350-1020	Hapus Edit
Ms. Halile Wiegand	anabel.nyatt@gmail.com	1-205-232-0159	170 Vivian Trafficway Suite 395 New Websterside, CT 66233	Hapus Edit

Gambar 5.23 Halaman Daftar Dokter Dashboard Admin

g. Cairan

Halaman ini berisi informasi cairan yang tersedia. Halaman ini juga dapat melakukan penambahan, pengubahan, dan penghapusan cairan.

Daftar Cairan

Cari Nama + Tambah Cairan

Page 1 of 1 1

Nama TI	Konsentrasi TI	Durasi Minimal TI	Durasi Maksimal TI	Aksi
Extraneal 7.5%	7.5	300	360	Hapus Edit
Dianeal 2.5%	2.5	300	360	Hapus Edit
Dianeal 1.5%	1.5	300	360	Hapus Edit

Page 1 of 1 1

Gambar 5.24 Halaman Daftar Cairan Dashboard Admin

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba dilakukan terhadap Aplikasi Mobile PWA pasien dan Dashboard Admin. Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dibangun dan kesesuaian hasil eksekusi perangkat lunak dengan analisis dan perancangan perangkat lunak.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap Aplikasi Mobile PWA pasien dan Dashboard Admin guna menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem aplikasi

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memerhatikan beberapa hasil yang diharapkan berikut ini:

- a. Kemampuan aplikasi menampilkan berbagai form.
- b. Kemampuan aplikasi menampilkan berbagai tabel.
- c. Kemampuan aplikasi menampilkan detail pasien.
- d. Kemampuan aplikasi untuk menyimpan data pengisian form.
- e. Kemampuan aplikasi untuk menampilkan grafik sesuai data yang diperlukan.
- f. Kemampuan aplikasi memenuhi kebutuhan lainnya, yaitu mencari data dengan filter pada tabel dan mengurutkan data pada kolom tabel tertentu.
- g. Kesesuaian dalam memenuhi kebutuhan non-fungsional aplikasi, yaitu:
 - Pengaksesan sistem dari area yang terhubung internet
 - Sistem memiliki tampilan (antarmuka) yang mudah dipahami

6.3. Skenario Pengujian

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 perubahan data dokter.
- i. Admin melihat informasi cairan.
- j. Admin melakukan penambahan, penghapusan, dan perubahan 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, perubahan, dan penghapusan resep penggantian
- e. Pasien melakukan penambahan dan perubahan 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

6.3.3. *Evaluasi Pengujian*

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem Aplikasi Mobile PWA pasien dan Dashboard Admin terhadap kasus skenario uji coba. Pengujian dilakukan oleh pihak pengembang, pengguna, dan pembimbing lapangan.

a. Admin

Berikut adalah hasil evaluasi pengujian dari aplikasi dashboard admin.

Tabel 6.1 Hasil Evaluasi Pengujian Aplikasi Dashboard Admin

Kriteria Pengujian	Hasil Pengujian
Melakukan login	Terpenuhi
Melihat halaman dashboard	Terpenuhi
Melihat halaman pengguna aktif	Terpenuhi
Melihat detail pasien atau pengguna	Terpenuhi
Melihat halaman verifikasi pengguna dan melakukan persetujuan atau penolakan pengguna	Terpenuhi
Melihat halaman dokter dan melakukan CRUD	Terpenuhi

Melihat halaman cairan dan melakukan CRUD	Terpenuhi
---	-----------

b. Pasien

Berikut adalah hasil evaluasi pengujian dari *Aplikasi Mobile* pasien.

Tabel 6.2 Hasil Evaluasi Pengujian Aplikasi Mobile PWA Pasien

Kriteria Pengujian	Hasil Pengujian
Pasien melakukan registrasi	Terpenuhi
Pasien melakukan login	Terpenuhi
Pasien dapat melihat halaman resep penggantian	Terpenuhi
Pasien melakukan penambahan, pengubahan, dan penghapusan resep penggantian	Terpenuhi
Pasien dapat melihat halaman logbook	Terpenuhi
Pasien melakukan penambahan dan pengubahan penggantian cairan logbook	Terpenuhi
Pasien dapat melihat halaman riwayat kesehatan	Terpenuhi
Pasien melakukan penambahan, pengubahan, dan penghapusan riwayat kesehatan	Terpenuhi
Pasien dapat melihat halaman statistik	Terpenuhi

Pasien dapat melihat data statistik penggantian cairan, berat badan, dan tekanan darah	Terpenuhi
Pasien dapat melihat halaman berisi data profil dan dokter	Terpenuhi
Pasien dapat melakukan perubahan data profil dan dokter	Terpenuhi
Pasien dapat melihat halaman pengaturan	Terpenuhi
Pasien dapat melakukan penggantian password	Terpenuhi

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pengembangan aplikasi pada kegiatan KP di Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember adalah sebagai berikut:

- a. Aplikasi yang dibangun berguna untuk membantu pasien CAPD untuk melakukan pencatatan penggantian cairan
- b. Dengan adanya website admin, admin dapat memantau detail dari penggantian cairan pasien seperti cairan masuk, cairan keluar, berat badan, serta tekanan darah. Admin juga dapat dengan mudah menambahkan, mengubah, dan menghapus data cairan dan dokter.

7.2. Saran

Berikut ini adalah saran yang penulis berikan untuk arah perkembangan selanjutnya, yaitu sebaiknya dalam penerapan *view* untuk lebih meningkatkan *user friendliness* untuk *mobile view*.

DAFTAR PUSTAKA

- [1] M. Goldstein, M. Carrillo, and S. Ghai, “Continuous ambulatory peritoneal dialysis—a guide to imaging appearances and complications,” *Insights into Imaging*, vol. 4, no. 1, pp. 85–92, Dec. 2012, doi: <https://doi.org/10.1007/s13244-012-0203-y>.
- [2] N. Wearne *et al.*, “Continuous ambulatory peritoneal dialysis: perspectives on patient selection in low- to middle-income countries,” *International Journal of Nephrology and Renovascular Disease*, vol. 10, pp. 1–9, Jan. 2017, doi: <https://doi.org/10.2147/IJNRD.S104208>.
- [3] Wu J, J. Z, Pi M, and Y. T, “Real-Time Monitoring and Reminding of Remote Peritoneal Dialysis System Based on the Principle of Least Squares,” *International journal of nephrology and kidney failure*, vol. 7, no. 3, Jan. 2021, doi: <https://doi.org/10.16966/2380-5498.214>.
- [4] E. J. Cartwright, Z. ZS Goh, M. Foo, C. M. Chan, H. Htay, and K. Griva, “eHealth interventions to support patients in delivering and managing peritoneal dialysis at home: A systematic review,” *Peritoneal Dialysis International: Journal of the International Society for Peritoneal Dialysis*, vol. 41, no. 1, p. 089686082091813, Apr. 2020, doi: <https://doi.org/10.1177/0896860820918135>.
- [5] F. I. Eka Sari, N. A. Nurwardani, M. A. Fandiaz, S. Ulwiya, M. M. A. Jalil, and D. A. Navastara, “Mobile Virtual Assistant for Continuous Ambulatory Peritoneal Dialysis Complication Detection,” *CSRID (Computer Science Research and Its Development Journal)*, vol. 14, no. 2, p. 136, Sep. 2022, doi: <https://doi.org/10.22303/csrid.14.2.2022.136-143>.
- [6] Dini Adni Navastara *et al.*, “Abnormality Detection of Effluent Dialysate Images on Continuous Ambulatory Peritoneal Dialysis Using Deep Learning,” *2023 6th International Seminar on Research of Information*

- Technology and Intelligent Systems (ISRITI)*, Dec. 2023, doi: <https://doi.org/10.1109/isriti60336.2023.10467296>.
- [7] M. M. A. Jalil, “Rancang Bangun Sistem Pendeteksi dan Pemantauan Risiko Komplikasi Continuous Ambulatory Peritoneal Dialysis (CAPD) bagi Pasien dengan Penyakit Ginjal Kronis,” Institut Teknologi Sepuluh Nopember, 2023.
- [8] Mozilla, “Progressive web apps,” *MDN Web Docs*, Jun. 14, 2019. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps (accessed Jun. 24, 2024).
- [9] “Sejarah,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/tentang-kami/sejarah/> (accessed Feb. 01, 2024).
- [10] “Visi dan Misi,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/tentang-kami/visi-dan-misi/> (accessed Feb. 01, 2024).
- [11] “Struktur Organisasi,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/dosen-staff/struktur-organisasi/> (accessed Jun. 17, 2024).
- [12] “Laboratorium,” *Departemen Teknik Informatika*. <https://www.its.ac.id/informatika/id/fasilitas/laboratorium/> (accessed Jun. 17, 2024).
- [13] “Mengenal CAPD, Metode Cuci Darah Lewat Perut,” *Alodokter*, Oct. 23, 2019. <https://www.alodokter.com/mengenal-capd-metode-cuci-darah-lewat-perut> (accessed Jun. 19, 2024).
- [14] L. Vale *et al.*, “Continuous ambulatory peritoneal dialysis (CAPD) versus hospital or home haemodialysis for end-stage renal disease in adults,” *Cochrane Database of Systematic Reviews*, Oct. 2004, doi: <https://doi.org/10.1002/14651858.cd003963.pub2>.
- [15] “Refactoring - Introduction and Its Techniques,” *GeeksforGeeks*, Aug. 08, 2020.

- <https://www.geeksforgeeks.org/refactoring-introduction-and-its-techniques/> (accessed Jun. 19, 2024).
- [16] “What is Figma?,” *Figma Learn*.
<https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma> (accessed Feb. 03, 2024).
- [17] “What is TypeScript?,” *TypeScript Tutorial*.
<https://www.typescripttutorial.net/typescript-tutorial/what-is-typescript/> (accessed May 14, 2024).
- [18] Vercel, “What is Next.js?,” *nextjs.org*, 2024.
<https://nextjs.org/docs> (accessed May 14, 2024).
- [19] “Get started with Tailwind CSS,” *tailwindcss.com*.
<https://tailwindcss.com/docs/installation> (accessed May 14, 2024).
- [20] “Radix UI Introduction,” *radix-ui.com*. <https://www.radix-ui.com/primitives/docs/overview/introduction> (accessed May 14, 2024).
- [21] shadcn, “Introduction,” *ui.shadcn.com*.
<https://ui.shadcn.com/docs> (accessed May 14, 2024).
- [22] Axios, “Getting Started | Axios Docs,” *axios-http.com*, 2023. <https://axios-http.com/docs/intro> (accessed May 14, 2024).

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Muhammad Ferdian Iqbal, lahir pada tanggal 17 Januari 2002 di Magelang. Penulis merupakan mahasiswa yang sedang menempuh studi di Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS). Penulis aktif dalam kepanitiaan departemen dan beberapa program magang web development.



Aryan Shafa Wardana, lahir pada tanggal 13 April 2005 di Surabaya. 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 organisasi di kampus serta kepanitiaan dalam bidang web development.