

**Pengembangan *B-Care*: Aplikasi *Helpdesk* Instansi
Pemerintah di Kota Batam Berbasis *Mobile* Dengan
Integrasi *API***

PROYEK AKHIR

Disusun Oleh:
NADILLA NAGITA
3312311055



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan laporan ini dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dari pelaksanaan proyek/penelitian yang telah dilakukan. Dalam penyusunan laporan ini, penulis banyak mendapat dukungan, bimbingan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. **Ibu Mir'atul Khusna Mufida** selaku dosen pembimbing yang telah memberikan arahan, bimbingan, dan motivasi dalam proses penyusunan laporan ini.
2. **Politeknik Negeri Batam**, atas fasilitas belajar, ilmu pengetahuan dan kesempatan untuk mengembangkan diri
3. **Keluarga**, atas dukungan moril dan materi yang telah diberikan. Doa dan semangat dari kalian menjadi penyemangat penulis untuk terus berjuang.
4. **Teman-teman**, atas semangat dan diskusi yang bermanfaat selama proses pengerjaan Tugas Akhir
5. **Pihak Instansi**, yang telah memberikan kesempatan, fasilitas, serta dukungan dalam pelaksanaan penelitian ini.

Seluruh pihak yang telah berkontribusi dalam penyelesaian TA ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis dengan tulus menerima segala bentuk kritik dan saran yang membangun demi perbaikan di masa mendatang. Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca serta pihak-pihak yang berkepentingan.

Batam,

Nadilla Nagita

3312311055

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI	3
DAFTAR TABEL	5
DAFTAR GAMBAR	6
DAFTAR LAMPIRAN	8
BAB I PENDAHULUAN	10
1.1. Latar Belakang	10
1.2. Rumusan Masalah	12
1.3. Tujuan.....	12
1.4. Batasan Masalah.....	12
1.5. Manfaat.....	12
1.6 Kolaborasi dan Sinergi dalam Pelaksanaan Proyek	13
BAB II TINJAUAN PUSTAKA.....	14
2.1 Tinjauan Solusi dan Sistem Terkait	14
2.2 Landasan Teori	19
2.2.1 Definisi dan Konsep Utama.....	19
2.2.3 Dart.....	20
2.2.4 Rest API.....	20
2.2.7 Figma	21
2.2.9 Instansi pemerintah di kota batam	21
2.3 Metode Pengembangan Produk.....	22
2.3.1 Analisis Kebutuhan	23
2.3.2 Desain Sistem	23
2.3.3 Implementasi	24
2.3.4 Pengujian	24
2.3.5 Pemeliharaan	24
BAB III ANALISIS DAN PERANCANGAN.....	25
4.1.1 Proses Bisnis Berjalan	26
4.1.2 Gambaran Umum Sistem	29

3.2. Perancangan.....	33
3.2.1 Kebutuhan Fungsional.....	33
3.2.2 Kebutuhan Non-Fungsional.....	36
3.2.4 Skenario Use Case.....	37
3.2.5 <i>Activity</i> diagram	51
3.2.6 <i>ER</i> Diagram.....	63
3.2.7. Perancangan Antarmuka <i>Mock up</i>	64
3.2.8 Perancangan Integrasi <i>REST API</i>	87
3.2.9 Rencana Pengujian dan Indikator Keberhasilan Sistem.....	88
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	90
4.1. Hasil Implementasi.....	90
4.1.1 Spesifikasi Pengembangan.....	90
4.1.2 Implementasi Basis Data.....	92
4.1.3 Implementasi API (Backend Laravel).....	95
4.1.4 Implementasi Antarmuka	104
4.2 Pengujian Sistem.....	120
4.2.1 Pengujian <i>Black Box</i>	120
4.2.2 Pengujian Pengguna (<i>User Acceptance Testing</i>)	123
4.3 Pemeliharaan Sistem	124
BAB V KESIMPULAN.....	126
5.1 Kesimpulan	126
5.2 Keterbatasan Penelitian.....	126
5.3 Saran.....	127
DAFTAR PUSTAKA.....	128
DAFTAR LAMPIRAN	130
Lampiran A. Dokumen Proses Pengumpulan Requirement	130
Lampiran B. Link Produk.....	130

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Sebelumnya.....	17
Tabel 3. 1 Tabel Jenis Pengguna	34
Tabel 3. 2 Kebutuhan Fungsional	34
Tabel 3. 3 Kebutuhan NonFungsional	36
Tabel 3. 4 Skenario Use Case Login.....	38
Tabel 3. 5 Skenario Use case Registrasi Akun	39
Tabel 3. 6 Skenario Use case Melihat Info Layanan	40
Tabel 3. 7 Skenario Use case Membuat Tiket Layanan	40
Tabel 3. 8 Skenario Use case Melihat Status Tiket	41
Tabel 3. 9 Skenario Use Case Melihat Riwayat Tiket.....	42
Tabel 3. 10 Skenario Use Case Memberikan Ulasan.....	42
Tabel 3. 11 Skenario Use Case Lupa Password.....	43
Tabel 3. 12 Skenario Use Case Search Nomor Tiket	44
Tabel 3. 13 Skenario Use Case Edit Profil	45
Tabel 3. 14 Skenario Use Case Notifikasi	45
Tabel 3. 15 Skenario Use Case Logout.....	46
Tabel 3. 16 Skenario Use Case Login pelaksana atau supervisor	47
Tabel 3. 17 Skenario Use Case supervisor Melihat Daftar Tiket	47
Tabel 3. 18 Skenario Use Case pelaksana atau supervisor melihat detail tiket	48
Tabel 3. 19 Skenario Use Case pelaksana atau supervisor update status tiket	49
Tabel 3. 20 Skenario Use Case pelaksana atau supervisor konfirmasi tiket.....	50
Tabel 3. 21 Endpoint API	87
Tabel 4. 1 Perangkat Keras	90
Tabel 4. 2 Perangkat Lunak	91
Tabel 4. 3 Hasil Pengujian Endpoint API Register.....	95
Tabel 4. 4 Hasil Pengujian Endpoint API Login.....	97
Tabel 4. 5 Hasil Pengujian Endpoint API Manajemen Tiket.....	99
Tabel 4. 6 Hasil Pengujian Endpoint API Kelola Tiket	101
Tabel 4. 7 Blackbox Testing Sisi User	121
Tabel 4. 8 Blackbox Testing Sisi Supervisor	121
Tabel 4. 9 Blackbox Testing Sisi Pelaksana	122
Tabel 4. 10 Hasil Pengujian UAT	123
Tabel 4. 11 Peeliharaan Aplikasi	125

DAFTAR GAMBAR

Gambar 2. 1 Proses Komunikasi	20
Gambar 2. 2 Metode waterfall	23
Gambar 3. 1 Proses Bisnis	27
Gambar 3. 2 Gambaran Umum Sistem	30
Gambar 3. 3 Use Case Diagram	37
Gambar 3. 4 Activity Diagram Registrasi Akun	51
Gambar 3. 5 Activity Diagram Login	52
Gambar 3. 6 Activity Diagram Melihat Layanan pengaduan	53
Gambar 3. 7 Activity Diagram Status Tiket	54
Gambar 3. 8 Activity Diagram Riwayat Tiket	55
Gambar 3. 9 Activity Diagram Lupa Password	56
Gambar 3. 10 Activity Diagram Search	57
Gambar 3. 11 Activity Diagram Buat Tiket Sebagai Guest	58
Gambar 3. 12 Activity Diagram Edit Profile	59
Gambar 3. 13 Activity Diagram Login pelaksana dan supervisor	60
Gambar 3. 14 Activity Diagram Pelaksana Kelola Tiket	61
Gambar 3. 15 Activity Diagram Supervisor Kelola Tiket	62
Gambar 3. 16 ER Diagram	63
Gambar 3. 17 Splash page	64
Gambar 3. 18 Welcome Page	65
Gambar 3. 19 Login Page	66
Gambar 3. 20 Register Page 1	67
Gambar 3. 21 Register Page 2	68
Gambar 3. 22 Dashboard Page	69
Gambar 3. 23 Kategori Layanan Page	70
Gambar 3. 24 Layanan Page	71
Gambar 3. 25 Form Pengaduan Page	72
Gambar 3. 26 Riwayat Tiket Page	73
Gambar 3. 27 Detail Tiket riwayat Page	74
Gambar 3. 28 FAQ Page	75
Gambar 3. 29 Detail FAQ Page	76
Gambar 3. 30 Profile Page	77
Gambar 3. 31 Notifikasi Page	78
Gambar 3. 32 Guest Page	79
Gambar 3. 33 login pelaksana dan supervisor	80
Gambar 3. 34 Dashboard Pelaksana Page	81
Gambar 3. 35 Management tiket page	82
Gambar 3. 36 Detail Tiket Page	83
Gambar 3. 37 Tambah Tiket Supervisor dan Pelaksana Page	84
Gambar 3. 38 Notifikasi Page	85
Gambar 3. 39 Profile Page pelaksana	86

Gambar 4. 1 Tabel Database	92
Gambar 4. 2 Tabel akun	92
Gambar 4. 3 Tabel tiket	93
Gambar 4. 4 Tabel Riwayat status.....	93
Gambar 4. 5 Tabel Notifikasi	94
Gambar 4. 6 Tabel Layanan	94
Gambar 4. 7 Respon Error API Register Saat Email Kosong	96
Gambar 4. 8 Respon Berhasil API Register	96
Gambar 4. 9 Database Create akun baru	96
Gambar 4. 10 Respon Berhasil API Login	97
Gambar 4. 11 Respon API jika Password Salah.....	98
Gambar 4. 12 Respon API Login Jika Email Salah	98
Gambar 4. 13 Respon API Tiket User.....	99
Gambar 4. 14 Verifikasi Perbedaan Guest dan User pada Database.....	100
Gambar 4. 15 Respon API Jika Judul kosong	100
Gambar 4. 16 Respon API Search Tiket	101
Gambar 4. 17 Respon API Assign Tiket	102
Gambar 4. 18 Database Riwayat Status	103
Gambar 4. 19 Respon API Tiket Selesai Diproses.....	103
Gambar 4. 20 Implementasi Welcome Page	104
Gambar 4. 21 Implementasi Layanan Guest	105
Gambar 4. 22 Implementasi Pop Up Kode Tiket	105
Gambar 4. 23 Implementasi Tracking Kode Tiket Guest.....	106
Gambar 4. 24 Hasil Implementasi Detail Tiket.....	106
Gambar 4. 25 Implementasi Registrasi Akun 1.....	107
Gambar 4. 26 Implementasi Registrasi Akun 2.....	107
Gambar 4. 27 Implementasi Halaman Login	108
Gambar 4. 28 Implementasi Dashboard User	108
Gambar 4. 29 Implementasi Form Tiket	109
Gambar 4. 30 Implementasi Riwayat Tiket.....	109
Gambar 4. 31 Implementasi Detail riwayat Tiket	110
Gambar 4. 32 Implementasi Notifikasi	110
Gambar 4. 33 Implementasi Login Pengelola	111
Gambar 4. 34 Implementasi Dashboard Supervisor	112
Gambar 4. 35 Implementasi Page Tiket masuk.....	112
Gambar 4. 36 Implementasi Detail Tiket SPV	113
Gambar 4. 37 Implementasi Assign Pelaksana	114
Gambar 4. 38 Implementasi Monitoring Tiket.....	114
Gambar 4. 39 Implementasi Detail Tiket Monitoring	115
Gambar 4. 40 Implementasi Notifikasi SPV	115
Gambar 4. 41 Implementasi Dashboard Pelaksana	116
Gambar 4. 42 Implementasi List Tiket Pelaksana.....	117
Gambar 4. 43 Implementasi Detail Tiket Status Konfirmasi	117
Gambar 4. 44 Implementasi Detail Tiket Status Proses	118
Gambar 4. 45 Implementasi Riwayat Tiket Selesai	118
Gambar 4. 46 Implementasi Detail Tiket selesai Pelaksana.....	119
Gambar 4. 47 Implementasi FCM Notifikasi SmartPhone	119

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement	126
Lampiran B. Link Produk	126
Lampiran C. Dokumen Pengujian UAT	126

ABSTRAK

B-Care merupakan sistem layanan pengaduan yang digunakan di lingkungan instansi pemerintah di kota batam untuk mempermudah masyarakat maupun pegawai internal dalam menyampaikan laporan, kritik, dan saran. Selama ini sistem hanya tersedia dalam bentuk *website*, sehingga pengguna harus mengakses melalui *browser* cara ini dinilai kurang praktis untuk pengguna yang ingin membuat laporan pengaduan dengan cepat. Untuk meningkatkan kemudahan akses, proyek akhir ini mengembangkan aplikasi *mobile B-Care* dikembangkan menggunakan *framework flutter* dengan integrasi *API* dummy. Metode pengembangan perangkat lunak yang digunakan adalah metode Waterfall, yang dilaksanakan secara sistematis dan berurutan melalui tahapan analisis kebutuhan, perancangan sistem, implementasi, integrasi *API*, serta pengujian aplikasi. Produk yang dihasilkan berupa aplikasi *mobile B-Care* dengan fitur utama yang selaras dengan versi *website* yang sudah ada seperti registrasi dan *login* pengguna, pembuatan tiket pengaduan, pemantauan status tiket, serta riwayat pengaduan. Hasil pengujian menunjukkan bahwa aplikasi dapat berjalan dengan baik di perangkat *mobile*, mendukung proses pengaduan secara cepat dan praktis, serta meningkatkan pengalaman pengguna. Kesimpulannya, pengembangan *B-Care* berbasis *mobile* dengan Integrasi *API*.

Kata Kunci: Layanan Pengaduan, *Flutter*, Aplikasi *Mobile*, *API*, *Waterfall*

BAB I PENDAHULUAN

1.1. Latar Belakang

Instansi pemerintah di kota batam merupakan salah satu instansi pemerintah yang memiliki peran penting dalam memberikan pelayanan publik di wilayah Batam. Untuk meningkatkan upaya keterbukaan, kualitas layanan, instansi pemerintah di kota batam sejak tahun 2024 telah menghadirkan sistem layanan pengaduan Masyarakat yaitu *B-Care* dalam versi *website*. Seiring dengan perkembangan teknologi informasi yang berkembang pesat dan kebutuhan masyarakat yang semakin tinggi versi *website* saja tidak memadai. Banyak masyarakat yang lebih nyaman menggunakan *smartphone* untuk berbagai aktivitas termasuk mengakses layanan publik.

Namun, dalam penerapannya versi *website B-Care* masih memiliki keterbatasan. Berdasarkan hasil wawancara dengan staff IT instansi pemerintah di kota batam, pelaksana dan *supervisor* kesulitan memperbarui serta memantau tiket ketika berada di lapangan karena sistem hanya dapat diakses melalui komputer atau laptop. Hal ini membuat pengelola harus selalu membawa laptop, sehingga proses penanganan laporan menjadi kurang efisien dan lambat. Dari sisi masyarakat sebagai pengguna utama, kendala juga muncul karena jarang ada yang menggunakan laptop, sehingga akses melalui *website* menjadi tidak praktis. Meskipun dapat dibuka melalui browser di *smartphone*, tampilan yang kurang responsif dan navigasi yang terbatas membuat masyarakat merasa kurang nyaman saat membuat laporan maupun memantau status pengaduan secara *real-time*. Kondisi ini menegaskan adanya kesenjangan antara kebutuhan pengguna dan kemampuan sistem saat ini, baik bagi pengelola maupun masyarakat.

Kondisi tersebut menunjukkan bahwa sistem *B-Care* versi *website* belum sepenuhnya mampu memenuhi kebutuhan pengguna, baik dari sisi masyarakat maupun petugas internal di lapangan. Oleh karena itu, diperlukan pengembangan dalam bentuk aplikasi *mobile* agar pengguna dapat mengakses layanan dengan lebih cepat, efisien, dan sesuai kebutuhan di masyarakat. Melalui pengembangan ini, diharapkan hadir sistem pengaduan berbasis *mobile*, yang menjadi inovasi dan

wadah masyarakat dalam menyampaikan laporan, kritik, dan saran terkait pelayanan publik.

Menurut (Agustina et al., 2024) penerapan teknologi informasi pada pelayanan publik terbukti dapat memperbaiki kualitas pelayanan agar lebih cepat, tepat dan transparan. Penggabungan layanan online, sistem informasi dapat memperluas akses Masyarakat terhadap pelayanan pemerintah,

Asosiasi Penyelenggara Jasa Internet Indonesia (2024, February 7) Hasil survei APJII tahun 2024 menunjukkan bahwa jumlah pengguna internet di Indonesia sudah menembus lebih dari 221 juta jiwa, dengan tingkat penggunaan hampir 80%. Tren ini terus naik setiap tahun dan sebagian besar penggunanya berasal dari generasi muda seperti Gen Z dan Milenial, yang sehari-harinya selalu menggunakan *smartphone* untuk belajar, bekerja, maupun mengakses layanan digital.

Hal ini menggambarkan bahwa masyarakat semakin terbiasa menggunakan *smartphone* sebagai media utama untuk mencari informasi dan berinteraksi. Hal ini sama dengan kebutuhan akan layanan publik yang lebih praktis dan mudah dijangkau. Oleh karena itu, meskipun instansi pemerintah di kota Batam telah menghadirkan *B-Care* dalam bentuk *website*, pengembangan ke versi *mobile* menjadi sangat penting agar layanan pengaduan dapat lebih dekat, nyaman, dan sesuai dengan pola hidup masyarakat saat ini.

Berdasarkan kajian tersebut, proyek akhir ini dilakukan untuk mengembangkan aplikasi *mobile B-Care* berbasis *Flutter*, dengan integrasi *API* lokal/*dummy*, sehingga mendukung pengelolaan pengaduan secara terstruktur, mempermudah alur kerja pengguna, dan memfasilitasi keterbukaan informasi pelayanan.

1.2. Rumusan Masalah

1. Bagaimana merancang aplikasi *mobile B-Care* yang dapat digunakan oleh pelaksana, supervisor, dan masyarakat dalam melakukan pelaporan serta pemantauan pengaduan secara lebih fleksibel dibandingkan versi *website*?
2. Bagaimana implementasi *API* dalam mendukung proses simulasi pengelolaan data pengaduan pada aplikasi *mobile B-Care*?

1.3. Tujuan

1. Mengembangkan aplikasi *B-Care* berbasis *mobile* dengan integrasi *REST API* sebagai media simulasi dalam pengelolaan pengaduan masyarakat.
2. Merancang dan membangun fitur pelaporan serta pemantauan status pengaduan pada aplikasi *B-Care* lebih efektif melalui perangkat *mobile*.

1.4. Batasan Masalah

1. Aplikasi *mobile* hanya mengimplementasikan sebagian fitur dari *B-care* versi *website* sesuai kebutuhan pengelolaan tiket dan pemantauan pengaduan
2. Pengujian aplikasi dilakukan hanya pada perangkat berbasis *Android*.
3. Penelitian ini hanya berfokus pada pengembangan sisi aplikasi *mobile* dengan menggunakan *Rest Api* sebagai media simulasi dalam pengelolaan data pengaduan.
4. Aplikasi *B-Care Mobile* dikembangkan khusus untuk mendukung layanan pengaduan di BP Batam sehingga apabila diterapkan pada instansi lain diperlukan penyesuaian sesuai kebutuhan dan alur kerja masing-masing instansi.

1.5. Manfaat

Penelitian ini diharapkan dapat menambah wawasan dalam bidang pengembangan perangkat lunak, khususnya terkait penerapan *framework Flutter* dalam pembuatan aplikasi *mobile* serta penggunaan metode *waterfall* sebagai pendekatan pengembangan. Selain itu, penelitian ini juga dapat menjadi referensi bagi pengembangan aplikasi serupa di masa mendatang serta selain manfaat bagi instansi dan masyarakat, proyek akhir ini berkontribusi terhadap pencapaian *Sustainable Development Goals* (SDGs), khususnya pada Goal 16: Perdamaian,

Keadilan, dan Kelembagaan yang Tangguh. Melalui aplikasi B-Care, tercipta sistem transparansi dan akuntabilitas dalam pelayanan publik, di mana masyarakat dapat memantau proses pengaduan secara terbuka, sehingga memperkuat tata kelola lembaga pemerintah yang lebih efektif dan responsif.

1.6 Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Dalam pelaksanaan proyek ini, penulis melakukan kolaborasi aktif dengan berbagai pihak. Sinergi dilakukan bersama staff IT instansi terkait dalam proses analisis kebutuhan sistem dan struktur data, serta berkoordinasi dengan masyarakat pengguna selama tahap pengujian UAT. Selain itu, bimbingan intensif dengan dosen pembimbing memastikan bahwa solusi yang dibangun selaras dengan standar teknis dan akademis yang ditetapkan.

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Solusi dan Sistem Terkait

Sebelum pengembangan versi *mobile*, instansi pemerintah di kota batam telah menghadirkan *B- Care* versi *website* sebagai sistem layanan pengaduan masyarakat. Sistem ini memudahkan masyarakat dalam menyampaikan laporan, kritik, dan saran terkait pelayanan publik, serta telah digunakan secara aktif oleh pelaksana dan *supervisor* internal instansi pemerintah di kota batam sebagai saluran resmi untuk pengelolaan pengaduan. Kesuksesan awal *B-Care* versi *website* terlihat dari meningkatnya penggunaan oleh staff internal, dengan tingginya jumlah laporan yang masuk setiap harinya, meskipun data pengguna masyarakat secara umum belum tersedia secara resmi.

Namun, versi *website* juga memiliki beberapa keterbatasan signifikan. Sistem hanya dapat diakses secara optimal melalui komputer atau laptop, sehingga pelaksana dan *supervisor* yang sering berada di lapangan mengalami kesulitan dalam memantau dan memperbarui tiket. Selain itu, tampilan *website* kurang responsif pada perangkat *smartphone*, sehingga masyarakat kesulitan membuat laporan dan memantau status pengaduan secara praktis. Tidak adanya dukungan notifikasi *real-time* juga membuat proses pemantauan laporan menjadi kurang efisien bagi pengguna internal maupun masyarakat.

Menurut (De Paula Vieiralves et al., 2022) jurnal ini membahas tentang aplikasi *mobile service* desk yang dirancang untuk mempermudah proses pembukaan dan penanganan *call* atau permintaan layanan *industry* TI terkhusus pada Perusahaan minuman, sistem ini memungkinkan pengguna untuk mengirimkan masalah atau permintaan layanan, yang diteruskan ke profesional TI yang bertanggung jawab untuk menyelesaikan.

(Musa, 2023) Penelitian ini membahas pembangunan sistem informasi *Helpdesk* berbasis web di Kantor Desa Tanjung Sari. Sebelumnya, masyarakat

hanya bisa menyampaikan keluhan atau permintaan administrasi lewat telepon, *WhatsApp*, atau datang langsung ke kantor. Cara ini menimbulkan masalah, seperti laporan yang tidak tercatat, pencatatan manual yang rawan duplikasi, dan lambatnya penyampaian laporan ke kepala desa.

(Saragih & Azis, 2023) Penelitian ini bertujuan membangun aplikasi *service desk* berbasis Android yang dinamakan STELA (Sistem Tiket dan Eskalasi Layanan), untuk Pusat Teknologi dan Informasi (Pustekinfo) di Sekretariat Jenderal DPR-RI. Masalah yang dihadapi adalah bahwa pengelolaan keluhan/permintaan layanan TI (*hardware, software, jaringan*) masih dilakukan secara manual dan belum memiliki SOP (prosedur operasional standar) yang jelas. Aplikasi ini dirancang agar pengguna (pegawai & staf di DPR) dapat mengajukan tiket layanan kapan saja dan dari mana saja, sehingga layanan TI lebih fleksibel, terstruktur, dan sesuai prosedur.

(Sumarno & Mubarak, 2021) Penelitian ini membahas perancangan aplikasi pengaduan masyarakat berbasis *Android* untuk Desa Sindangsari. Sebelumnya, pengaduan dilakukan secara manual dengan tatap muka, sehingga masyarakat harus datang langsung ke kantor desa. Hal ini menyebabkan keterbatasan pelayanan karena hanya bisa dilakukan pada jam kerja, dan banyak warga yang tidak memahami proses administrasi desa. Aplikasi yang dikembangkan memungkinkan masyarakat menyampaikan pengaduan infrastruktur kapan saja dan di mana saja, sesuai alur administrasi yang benar. Selain itu, aplikasi menyediakan fitur berita desa agar informasi dapat tersebar lebih cepat dan menyeluruh.

Berdasarkan hasil tinjauan solusi dan sistem terkait, sebagian besar sistem helpdesk yang dikembangkan masih berbasis web dan belum sepenuhnya mendukung mobilitas pengguna dalam pengelolaan pengaduan. Selain itu, beberapa sistem belum mengakomodasi pembagian peran pengguna secara spesifik seperti pelaksana dan supervisor dalam proses penanganan tiket.

Oleh karena itu, pada penelitian ini dikembangkan aplikasi *B-Care Mobile* berbasis *Flutter* yang memungkinkan akses sistem secara lebih fleksibel dan real-time melalui perangkat *mobile*. Selain itu, aplikasi ini dilengkapi dengan fitur pembagian peran pengguna yang lebih terstruktur sehingga proses pengelolaan

dan monitoring pengaduan dapat dilakukan dengan lebih efektif dan terorganisir.

Dari keempat penelitian tersebut dapat disimpulkan bahwa meskipun berbagai sistem pengaduan dan tiket sudah berhasil dikembangkan dengan baik, masih terdapat keterbatasan, seperti cakupan penggunaan yang sempit, kurangnya fitur tambahan untuk kenyamanan pengguna, serta integrasi dengan sistem *backend* yang belum optimal. Berdasarkan berbagai solusi dan sistem yang telah dikembangkan sebelumnya, aplikasi B-Care Mobile dibangun sebagai pengembangan sistem layanan pengaduan berbasis mobile yang terintegrasi dengan API lokal berbentuk prototype serta notifikasi mobile untuk mendukung kenyamanan pengguna dan meningkatkan efektivitas layanan pengaduan.

pengaduan. Berikut tabel perbandingan penelitian yang sebelumnya dapat dilihat pada Tabel 2.1

Tabel 2. 1 Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan/ Celah
(De Paula Vieiralves et al., 2022)	Mobile Service Desk: Creating a Mobile Service Desk Application To Open Calls in The IT Industry In a Beverage Company	<i>Prototype, Java, Firebase</i>	Kemudahan akses pengguna menggunakan perangkat mobile, permissan layanan yang langsung diteruskan ke profesional yang tepat, dapat mengurangi waktu dalam menunggu	Tidak menyebutkan adanya perbandingan Tingkat dampak atau prioritas tiket.
(Musa, 2023)	Perancangan Helpdesk Tracking Sistem Pada Kantor Desa Tanjung	observasi & analisis sistem berjalan PHP, MySQL, XAMPP	Pencatatan sistematis, keluhan dan permintaan terdokumentasi dalam database	Skala penggunaan terbatas, berbasis website
(Saragih & Azis, 2023)	Rancang Bangun Aplikasi Service Desk Berbasis Android Sistem Tiket Dan Eskalasi Layanan	<i>Figma, API, PostMan, Black box testing</i>	Pengelolaan Data yang Lebih Baik, Data keluhan dan permohonan dapat dikelola dengan lebih sistematis.	Keterbatasan Fitur, Fitur yang tersedia belum mencakup semua kebutuhan pengguna.
(Sumarno &	Aplikasi Pengaduan Masyarakat	<i>Waterfall, Java</i>	Akses mudah, masyarakat	Tidak ada fitur notifikasi,

Mubarok, 2021)	terhadap Infrastruktur Berbasis Android pada Desa Sindangsari		bisa melaporkan saja tanpa harus datang ke kantor desa. Transparansi & dokumentasi, setiap pengaduan tercatat dalam sistem dan sesuai jalur birokrasi.	metode pengembangan kurang adaptif, skala terbatas di desa
(Nadilla Nagita, 2025)	Pengembangan B-Care: Aplikasi Helpdesk instansi pemerintah di kota batam berbasis flutter dengan Integrasi API	<i>Waterfall, Waterfall, Rest API, PostMan, Firebase Cloud Messaging</i>	Memfasilitasi pelaporan dan pemantauan status pengaduan melalui perangkat mobile, serta mendukung pengelolaan tiket secara terstruktur dan efisien menggunakan database lokal	Belum tersedia dalam platform iOS

2.2 Landasan Teori

2.2.1 Definisi dan Konsep Utama

Pada awalnya, *B-Care* dikembangkan dalam bentuk website sebagai sistem layanan pengaduan masyarakat berbasis digital yang memudahkan pengguna dalam menyampaikan laporan, kritik, dan saran terkait pelayanan publik instansi pemerintah di kota batam. melalui versi *website* ini, pengguna dapat mengakses layanan menggunakan perangkat komputer maupun *browser* pada *smartphone*. Namun, seiring dengan perkembangan teknologi mobile dan kebutuhan akan akses layanan yang lebih praktis, fleksibel, cepat, dan efisien di lapangan lebih dibutuhkan, maka dilakukan pengembangan *B-Care Mobile*.

Aplikasi *mobile* sendiri merupakan perangkat lunak yang berjalan pada *smartphone*, yang memungkinkan pengguna mengakses layanan secara *real-time* dan lebih praktis. Menurut Franko (2012), aplikasi *mobile* mendukung akses informasi yang cepat. Pada proyek *B-Care Mobile*, aplikasi ini dirancang untuk menyediakan fitur dalam membuat laporan pengaduan, memantau status tiket, penyampaian informasi layanan melalui perangkat pintar.

2.2.2 Flutter

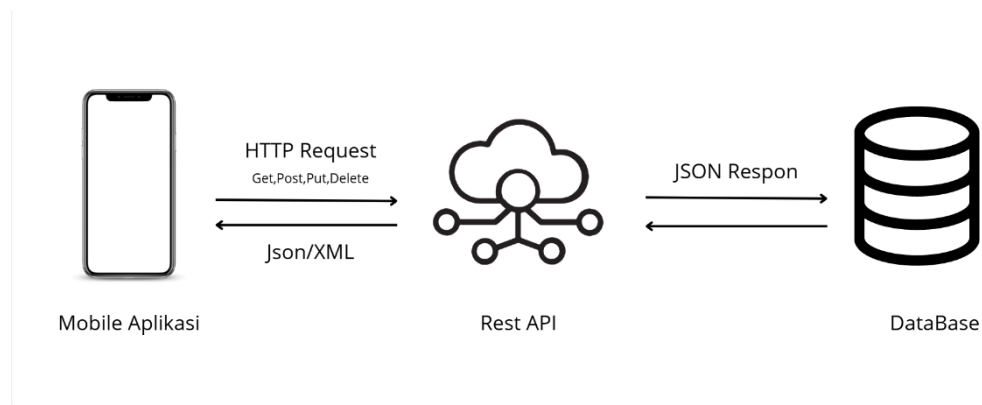
(Japara & Arifin, 2023) *Flutter* adalah *framework open-source* yang dikembangkan oleh *Google* untuk membangun aplikasi *mobile* berkinerja tinggi di berbagai *platform* dari satu basis kode. *Flutter* memungkinkan pengembangan aplikasi untuk *IOS*, *Android*, web, desktop secara bersamaan tanpa mengorbankan kualitas atau performa. Keunggulan utama *Flutter* termasuk *hot reload*, *widget* yang dapat disesuaikan, dan integrasi yang kuat dengan *Dart*.

2.2.3 Dart

(Urathal Alias Sri Swathiga et al., 2021) Dart adalah bahasa pemrograman yang dikembangkan oleh *Google*, dirancang untuk membangun aplikasi berkualitas tinggi di berbagai *platform*. Dart mendukung fitur-fitur modern seperti *null safety* dan *pattern matching*, serta kompatibel dengan *Flutter* untuk pengembangan aplikasi *mobile*.

2.2.4 Rest API

(Belkhir et al., 2019) *REST (Representational State Transfer)* adalah arsitektur yang digunakan untuk membangun layanan web yang dapat diakses melalui *HTTP*. *REST API* memungkinkan aplikasi *mobile* berkomunikasi dengan server untuk mengambil atau mengirim data, mendukung integrasi dengan berbagai sistem dan layanan.



Gambar 2. 1 Proses Komunikasi

Gambar diatas menunjukkan proses komunikasi antara aplikasi mobile dan server melalui *Rest API*. Aplikasi mengirimkan permintaan menggunakan metode *HTTP (Get, Post, Put, Delete)* dalam format *JSON/XML*, kemudian *Rest API* akan mengolah permintaan tersebut, mengambil atau memperbarui data pada *database* dan mengembalikan respon dalam format *JSON* pada aplikasi *mobile*.

2.2.5 Firebase Cloud Messaging

(Putro et al., 2022) *Firebase Cloud Messaging (FCM)* adalah salah satu layanan andalan dari *Google* yang dirancang khusus untuk mengirimkan pesan dan notifikasi ke berbagai perangkat pengguna, baik itu *Android, iOS*, maupun aplikasi berbasis web. Dengan *FCM*, pengembang dapat menyampaikan informasi penting secara *real-time* tanpa perlu membangun infrastruktur pengiriman pesan sendiri.

2.2.6 Visual Studio Code

(Tan et al., 2024) *Visual Studio Code (VS Code)* adalah editor kode sumber yang ringan namun kuat, dikembangkan oleh *Microsoft*. *VS Code* mendukung berbagai bahasa pemrograman dan dilengkapi dengan fitur seperti *IntelliSense*, *debugging*, dan kontrol versi, serta dapat diperluas dengan berbagai ekstensi untuk mendukung pengembangan aplikasi *mobile* dan web.

2.2.7 Figma

(Kimseng et al., 2024) Figma adalah alat desain antarmuka pengguna berbasis web yang memungkinkan kolaborasi *real-time*. Figma digunakan untuk merancang *prototipe*, *wireframe*, dan elemen UI lainnya, serta mendukung alur kerja desain yang efisien dengan fitur berbagi dan umpan balik langsung.

2.2.8 Postman

(Kore et al., 2022) *Postman* adalah alat yang digunakan untuk mengembangkan, menguji, dan mendokumentasikan *API*. Dengan *Postman*, pengembang dapat membuat permintaan HTTP, memeriksa respons, dan mengotomatisasi pengujian *API* untuk memastikan fungsionalitas dan kinerja yang diinginkan.

2.2.9 Instansi pemerintah di kota batam

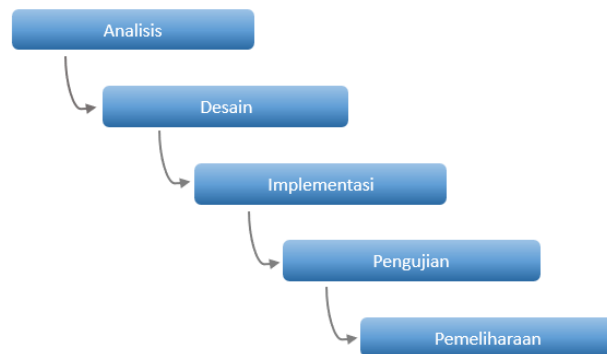
(Batam, 2025) instansi pemerintah di kota batam merupakan salah satu instansi pemerintah yang menyelenggarakan pelayanan publik dan pengelolaan informasi kepada masyarakat. Dalam upaya meningkatkan kualitas layanan dan transparansi informasi publik, instansi pemerintah di kota batam telah mengembangkan sistem layanan pengaduan berbasis web yang terintegrasi dengan portal layanan informasi publik internal.

Sistem tersebut memungkinkan masyarakat untuk mengajukan permohonan informasi serta menyampaikan pengaduan secara daring. Fitur yang tersedia meliputi pembuatan tiket layanan, pelacakan (*tracking*) status permohonan, serta komunikasi antara pengguna dan petugas.

Dalam konteks pengembangan sistem pengaduan publik, sistem berbasis web ini telah digunakan untuk memfasilitasi masyarakat dan pegawai dalam menyampaikan pengaduan dan permohonan informasi. Namun, berdasarkan hasil observasi dan wawancara dengan pihak internal, masih terdapat tantangan dalam hal aksesibilitas dan fleksibilitas penggunaan, terutama bagi petugas yang bekerja di lapangan atau memiliki mobilitas tinggi. Kondisi ini menunjukkan perlunya pengembangan sistem dalam bentuk aplikasi mobile sebagai pendukung layanan berbasis web yang telah ada.

2.3 Metode Pengembangan Produk

Metode pengembangan sistem merupakan suatu pendekatan yang digunakan untuk merancang, membangun, dan memelihara perangkat lunak melalui tahapan yang terstruktur dan sistematis. Dalam penelitian ini, pengembangan aplikasi dilakukan dengan menggunakan metode *Waterfall*. Menurut (Ind, Karambir, 2015) yang bersifat bertahap, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini terdiri atas lima fase utama, yaitu analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Pendekatan ini banyak digunakan karena memberikan alur yang jelas serta dokumentasi yang lengkap, sehingga memudahkan peneliti maupun pengembang dalam melaksanakan pembangunan sistem secara terarah dan sistematis. Berikut adalah metode yang digunakan untuk mengembangkan produk.



Gambar 2. 2 Metode waterfall

2.3.1 Analisis Kebutuhan

Pada tahap ini, peneliti berfokus untuk memahami apa saja yang dibutuhkan pengguna dari aplikasi. Proses dimulai dengan melihat kembali fitur-fitur yang sudah ada pada *B-Care Web*, lalu menentukan fitur utama yang wajib ada di aplikasi *mobile*, seperti *login*, manajemen tiket, pembaruan status, dan notifikasi. Selain itu, diperhatikan juga kebutuhan non fungsional, misalnya keamanan, kecepatan aplikasi, serta tampilan yang ramah pengguna.

2.3.2 Desain Sistem

Tahap desain bertujuan menggambarkan ide menjadi gambaran nyata tentang bagaimana aplikasi akan terlihat dan bekerja. Kegiatan di tahap ini meliputi pembuatan *mockup* dengan Figma, menyusun alur navigasi di *Flutter* agar mudah digunakan, serta menyiapkan rancangan integrasi *API* dari prototype.

2.3.3 Implementasi

Setelah rancangan siap, tahap berikutnya adalah mengubah desain menjadi aplikasi nyata. Proses implementasi dilakukan dengan membangun aplikasi menggunakan *Flutter* dan *Dart*, mengembangkan fitur-fitur utama seperti login, tiket, *update* status, dan notifikasi, serta menghubungkannya dengan *API* lokal agar data bisa terintegrasi dengan baik.

2.3.4 Pengujian

Tahap ini dilakukan untuk memastikan aplikasi berjalan sesuai harapan. Pengujian mencakup pengecekan *API* dengan Postman, pengujian fungsional pada setiap modul, serta mencoba aplikasi langsung di perangkat *Android* untuk melihat apakah performanya sudah stabil dan nyaman digunakan.

2.3.5 Pemeliharaan

Setelah aplikasi digunakan, akan selalu ada kebutuhan untuk melakukan perbaikan dan peningkatan. Pada tahap ini, peneliti melakukan perbaikan bug berdasarkan laporan pengguna, meningkatkan performa aplikasi.

BAB III ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan

Analisis sistem dilakukan untuk memahami bagaimana sistem *B-Care* versi *website* yang telah berjalan digunakan oleh instansi pemerintah di kota batam dalam menangani laporan pengaduan masyarakat maupun pegawai internal. Analisis ini meliputi identifikasi pengguna, alur proses bisnis, serta kendala yang dihadapi pada sistem yang berjalan.

Berdasarkan hasil wawancara dengan staff IT instansi pemerintah di kota batam, sistem *B-Care* web telah berjalan efektif dalam proses pengelolaan laporan melalui 6 peran utama yaitu pelaksana, *supervisor*, superadmin, *customer service* dan pengguna yaitu *user* dan *guest*. Namun, sistem tersebut hanya dapat diakses melalui perangkat komputer, dan dapat diakses melalui browser mobile, namun tampilan belum responsif serta navigasi kurang *user-friendly*, sehingga pelaksana dan supervisor yang beraktivitas di lapangan mengalami kesulitan dalam memperbarui dan memantau tiket secara langsung. Selain itu, masyarakat juga mengalami kendala dalam membuat dan memantau tiket layanan melalui perangkat *mobile*. Dari keenam aktor yang diidentifikasi, pengembangan aplikasi mobile dalam penelitian ini difokuskan pada 4 peran (User, Guest, Pelaksana, dan Supervisor) untuk kebutuhan operasional di lapangan. Adapun peran Superadmin dan Customer Service diimplementasikan pada platform web untuk manajemen data pusat.

Dari hasil analisis tersebut, disimpulkan bahwa diperlukan pengembangan sistem dalam bentuk *B-Care* versi *mobile* yang diharapkan dapat mempermudah ke-4 peran yaitu pelaksana, supervisor, user, dan guest dalam melakukan pembuatan tiket, pengelolaan tiket, pembaruan status, serta monitoring laporan secara *real-time* melalui perangkat *smartphone*.

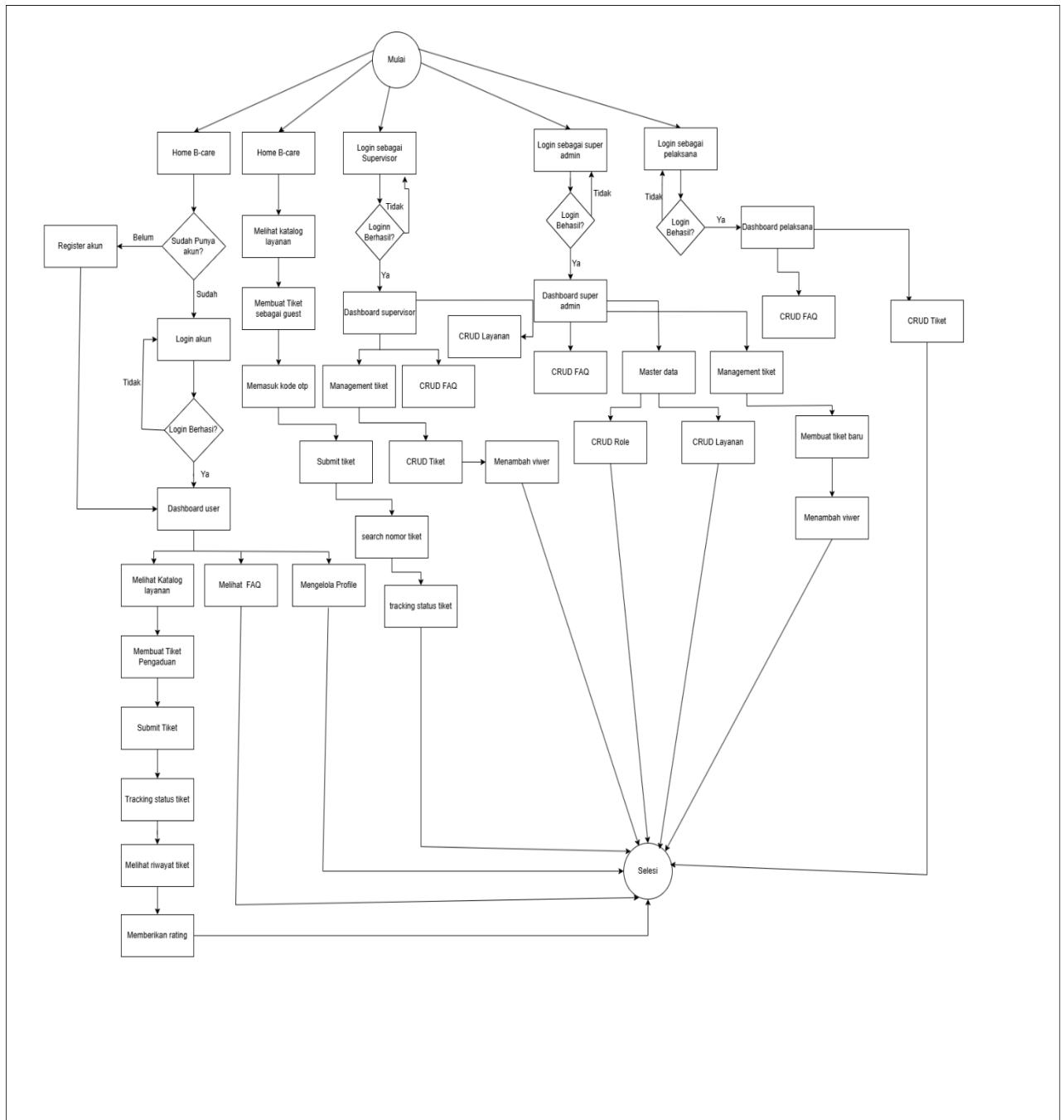
Cakupan keahlian Madya Komputer dalam riset ini ditunjukkan melalui kemampuan implementasi perangkat lunak secara utuh, mulai dari membangun struktur *database* yang efisien, pengembangan *Backend* berbasis *API* menggunakan *Framework Laravel*, hingga transformasi desain antarmuka menjadi

aplikasi *mobile* fungsional menggunakan flutter. Fokus utama riset ini adalah pada ketersediaan solusi praktis yang dapat langsung digunakan untuk menangani keluhan di unit kerja saat ini.

Meskipun dikembangkan untuk instansi pemerintah di Kota Batam, konsep riset ini telah mengestimasi fleksibilitas penggunaan di luar unit kerja tersebut. Dengan melakukan konfigurasi pada kategori kendala dan struktur organisasi di sisi *database*, sistem *Helpdesk* ini dapat diadaptasi untuk berbagai instansi layanan publik lainnya, seperti layanan pengaduan sarana prasarana kampus atau rumah sakit, karena alur kerja (*workflow*) tiket yang dirancang bersifat universal. Hal ini membuktikan bahwa arsitektur sistem yang dibangun memiliki skalabilitas dan potensi penerapan yang luas pada berbagai studi kasus manajemen layanan.

3.1.1 Proses Bisnis Berjalan

Proses Flow atau proses bisnis adalah langkah dan aktivitas yang dilakukan untuk merancang, mengembangkan, dan menerapkan aplikasi yang memenuhi kebutuhan pengguna dan tujuan bisnis. Berikut ini adalah proses bisnis pembuatan tiket pengaduan di aplikasi *B-Care* yang ditampilkan pada gambar 3.1



Gambar 3. 1 Proses Bisnis

Gambar 3.1 adalah gambar Proses bisnis yang ditunjukkan merupakan alur utama sistem *B-Care* yang berjalan pada versi web. Aplikasi *B-Care Mobile* yang dikembangkan dalam penelitian ini mengikuti alur bisnis yang sama, namun disesuaikan dengan kebutuhan pengguna di perangkat *mobile*. Beberapa fitur yang sebelumnya dikelola melalui antarmuka web, seperti manajemen tiket oleh

supervisor dan pelaksana, kini dioptimalkan agar dapat diakses secara langsung melalui aplikasi *mobile*.

1. *Guest*

Bagi pengguna yang belum memiliki akun, sistem menyediakan opsi untuk mengakses aplikasi sebagai *Guest*. *Guest* dapat melihat informasi layanan yang tersedia di sistem dan juga membuat tiket pengaduan tanpa harus melakukan *login*. Setelah tiket dibuat, sistem akan memberikan kode tiket unik yang bisa digunakan untuk melacak status laporan di kemudian hari. Selain itu, *guest* juga bisa menerima notifikasi pembaruan status tiket dan mencari tiket berdasarkan kode tersebut.

2. *User*

Jika pengguna sudah memiliki akun, maka mereka bisa *login* ke sistem sebagai *User*. Apabila belum memiliki akun, pengguna dapat melakukan registrasi terlebih dahulu. Setelah berhasil *login*, user akan diarahkan ke *dashboard* utama, di mana mereka dapat:

1. Melihat katalog layanan dan *FAQ*,
 2. Membuat laporan pengaduan baru melalui form yang tersedia,
 3. Melihat riwayat tiket yang pernah dibuat,
 4. Mengelola profil pribadi, dan
 5. Memberikan rating atau ulasan setelah tiket selesai diproses.
- Dengan begitu, user dapat berinteraksi penuh dengan sistem dan memantau progres pengaduan mereka secara langsung.

3. *Supervisor*

Aktor *supervisor* memiliki peran untuk memantau dan mengelola tiket yang masuk dari masyarakat. Setelah *login*, *supervisor* akan diarahkan ke *dashboard supervisor*, di mana mereka bisa melakukan:

1. Pengelolaan tiket (melihat daftar laporan yang masuk),
2. Menugaskan pelaksana yang akan menangani tiket tertentu,
3. Melakukan pembaruan status tiket, dan

4. Mengelola konten seperti FAQ atau daftar layanan. Peran *supervisor* ini memastikan setiap laporan masyarakat bisa segera ditindaklanjuti oleh pelaksana yang tepat.

4. Super Admin

Super Admin memiliki akses tertinggi dalam sistem *B-Care*. Setelah *login*, super admin diarahkan ke *dashboard* super admin yang berisi berbagai fitur pengelolaan data dan sistem, seperti:

1. Master Data, untuk mengatur data layanan, kategori, role pengguna, dan SLA,
2. Manajemen Tiket, untuk melihat semua laporan yang masuk dan status penanganannya,
3. CRUD FAQ dan CRUD Layanan, untuk memperbarui informasi yang muncul di aplikasi. Peran super admin ini berfungsi sebagai pengelola utama yang memastikan sistem tetap berjalan dengan baik dan data tetap terorganisir.

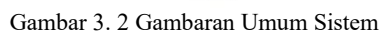
5. Pelaksana

Pelaksana adalah pihak yang menangani langsung laporan dari masyarakat. Setelah *login*, pelaksana diarahkan ke *dashboard* pelaksana, di mana mereka dapat:

1. Melihat daftar tiket yang telah ditugaskan oleh *supervisor*,
2. Menambahkan catatan hasil penanganan,
3. Memperbarui status pekerjaan, dan
4. Mengakhiri tiket setelah laporan selesai diproses.
5. Mengambil tiket yang ingin di proses

3.1.2 Gambaran Umum Sistem

Gambaran umum adalah sebuah gambaran mengenai bagaimana suatu sistem bekerja, komponen yang ada pada sistem, dan bagaimana komponen-komponen tersebut saling berinteraksi untuk mencapai tujuan tertentu. Berikut ini adalah gambaran umum sistem :



30

memantau status tiket. Pada sistem *B-Care Mobile*, bagian berwarna abu-abu menunjukkan Tim IT Pusat Data dan Sistem Informasi Instansi yang bertanggung jawab terhadap pengelolaan *server* dan *database* sistem *B-Care*. Aktivitas yang dilakukan meliputi pemeliharaan *server*, pembaruan konfigurasi, serta pengelolaan data yang tidak dilakukan melalui aplikasi *mobile*, tetapi secara internal melalui sistem web *B-Care*. Aplikasi *mobile* hanya berfungsi untuk menampilkan, memperbarui, dan mengirim data melalui *API* lokal, serta membangun *API* untuk kepentingan pengujian aplikasi. penjelasan mengenai apa saja yang bisa dilakukan oleh keempat peran tersebut adalah sebagai berikut:

a. User

User Adalah pegawai atau Masyarakat yang memiliki akun untuk membuat tiket pengaduan. *User* memiliki akses penuh terhadap fitur pengaduan layanan, mulai dari registrasi akun hingga riwayat tiket.

Berikut Adalah hal-hal yang dapat dilakukan oleh user dalam aplikasi:

1. *Login* menggunakan akun yang sudah terdaftar.
2. Mengelola tiket layanan, meliputi:
 - a. Membuat tiket layanan baru sesuai kategori.
 - b. Melakukan submit tiket.
 - c. Melihat status tiket secara *real-time*.
 - d. Melihat riwayat seluruh tiket yang pernah dibuat.
 - e. Memberikan ulasan/*feedback* terhadap penyelesaian tiket.
 - f. Menerima notifikasi terkait perkembangan tiket.
3. Mengelola akun, seperti memperbarui profil dan melakukan reset/lupa *password*.
4. *Logout* setelah selesai menggunakan aplikasi.

b. Guest

Guest adalah pengguna yang tidak memiliki akun, namun tetap dapat membuat pengaduan dengan akses terbatas. *Guest* hanya bisa membuat tiket dan melacak status tiket menggunakan nomor tiket.

1. Hal-hal yang dapat dilakukan oleh *guest* di dalam aplikasi:

2. Membuat tiket layanan sebagai guest dengan mengisi data dasar.
3. Mendapatkan nomor tiket atau kode referensi setelah submit.
4. Mendapatkan notifikasi setelah pembuatan tiket
5. Melakukan pelacakan status tiket menggunakan nomor tiket tersebut.
6. *Logout* setelah selesai menggunakan aplikasi.

c. Supervisor

Adalah pengelola yang memiliki peran dalam memantau tiket yang masuk, menunjuk pelaksana untuk menangani tiket tertentu, memverifikasi hasil pekerjaan, memantau progres penyelesaian tiket, membuat tiket, mengkonfirmasi menugaskan pelaksana mengelola tiket, menambah *viewer* memberikan komentar, serta *Re-open* tiket.

d. Pelaksana

Adalah pengelola yang memiliki fungsi untuk mengambil tiket yang tersedia, memproses laporan sesuai bidang kerja, membuat tiket, mengkonfirmasi, menambah *viewer*, mengelola tiket, memberikan komentar, serta *Re-open* tiket, memperbarui status pengerjaan, serta menandai laporan sebagai selesai.

e. API, Server, dan Database dalam Sistem B-Care Mobile

Untuk mendukung semua aktivitas *user*, *guest*, *supervisor*, dan pelaksana, *B-Care Mobile* menggunakan arsitektur *Client* → *API* → *Server* → *Database*, sebagai berikut:

1. Aplikasi (*Client*) ke *API*
 - a. Aplikasi mengirim *request* ke *API* ketika user melakukan aksi, misal membuat tiket, melihat status, atau menambah komentar.
 - b. *Request* berisi data dan instruksi yang diperlukan, misal informasi tiket, *user ID*, atau token autentikasi.
 - c. *API* memberikan response berisi hasil *request*, seperti konfirmasi pembuatan tiket, data tiket, atau status error.
2. *API* ke *Server*
 - a. *API* meneruskan request ke *server backend* yang menjalankan logika bisnis:

Validasi data - Cek hak akses - Proses pengaturan tiket
(misal supervisor menugaskan pelaksana)

- b. *Server* mengembalikan hasil proses ke *API* sebagai *feedback*, yang kemudian diteruskan ke aplikasi.

5. *Server* ke *Database*

- a. *Server* mengirim perintah ke *database* untuk menyimpan, mengubah, atau mengambil data tiket, user, atau status.
- b. *Database* mengeksekusi perintah dan mengirim umpan balik ke *server*:
Berhasil → data tersimpan / ID tiket baru
Gagal → *error message*
- c. *Server* meneruskan hasil tersebut ke *API*, lalu *API* mengirim ke aplikasi untuk ditampilkan ke *user*.

3.2. Perancangan

Perancangan sistem dilakukan untuk memberikan gambaran jelas mengenai bentuk dan mekanisme kerja aplikasi *B-Care Mobile* yang akan dibangun. Tahap perancangan ini bertujuan untuk menjabarkan kebutuhan fungsional maupun non-fungsional ke dalam rancangan teknis yang dapat digunakan sebagai acuan pada tahap implementasi.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan fitur atau fungsi yang harus ada dalam sistem, serta batasan atau keterbatasan dalam penggunaannya. Berikut ini adalah tabel jenis pengguna kebutuhan fungsional dari aplikasi yang tertera dalam tabel 3.1

Tabel 3. 1 Tabel Jenis Pengguna

User	User yang memakai aplikasi untuk membuat tiket pengaduan, melihat status laporan, dan memantau hasil penyelesaian. User bisa login, menggunakan fitur yang tersedia
Guest	Merupakan pengguna yang dapat mengakses aplikasi tanpa melakukan proses login.
Pelaksana	Pengguna internal yang bertugas mengambil, memproses, dan menyelesaikan tiket pengaduan dari masyarakat.
Supervisor	Pengguna internal yang bertugas memantau dan mengelola tiket pengaduan. Supervisor dapat menugaskan pelaksana, memeriksa progres pekerjaan, serta memastikan tiket diselesaikan dengan baik

Tabel 3. 2 Kebutuhan Fungsional

Kode	Kebutuhan Fungsional	Aktor
KF-01	Dapat Melakukan <i>Login</i>	<i>User</i> , supervisor, dan pelaksana
KF-02	Registrasi Akun	<i>User</i>
KF-03	Dapat Melihat Info Layanan	<i>User</i>
KF-04	Dapat Membuat Tiket Layanan	<i>User</i> , <i>Guest</i>
KF-05	Dapat Melihat Riwayat tiket	<i>User</i>
KF-06	Melihat detail tiket	<i>User</i>

KF-07	Dapat Melihat Status Tiket	<i>User</i>
KF-08	Dapat melihat katalog layanan	<i>User</i>
KF-09	Mendapatkan Notifikasi otomatis dari sistem	<i>User, supervisor, pelaksana,</i>
KF-10	Memberikan Ulasan kepada sistem	<i>User</i>
KF-11	Dapat mencari nomor Tiket yang sudah dibuat	<i>User dan Guest</i>
KF-12	Pengaturan <i>Profile</i>	<i>User</i>
KF-13	Lupa <i>Password</i>	<i>User</i>
KF-14	Dapat <i>Memperbarui</i> Status Tiket	<i>Supervisor dan pelaksana</i>
KF-15	Dapat Melihat Detail Tiket yang masuk	<i>Supervisor dan pelaksana</i>
KF-16	Dapat Melihat Daftar Tiket yang sedang diproses	<i>Supervisor dan pelaksana</i>
KF-17	Dapat memberikan catatan pada tiket	<i>Supervisor dan pelaksana</i>
KF-18	Re-Open Tiket	<i>Supervisor dan pelaksana</i>
KF-19	Menugaskan Pelaksana	<i>Supervisor</i>
KF-20	Konfirmasi Tiket	<i>Supervisor dan pelaksana</i>
KF-21	Dapat Melakukan <i>Logout</i>	<i>User, supervisor, pelaksana, Guest</i>

3.2.2 Kebutuhan Non-Fungsional

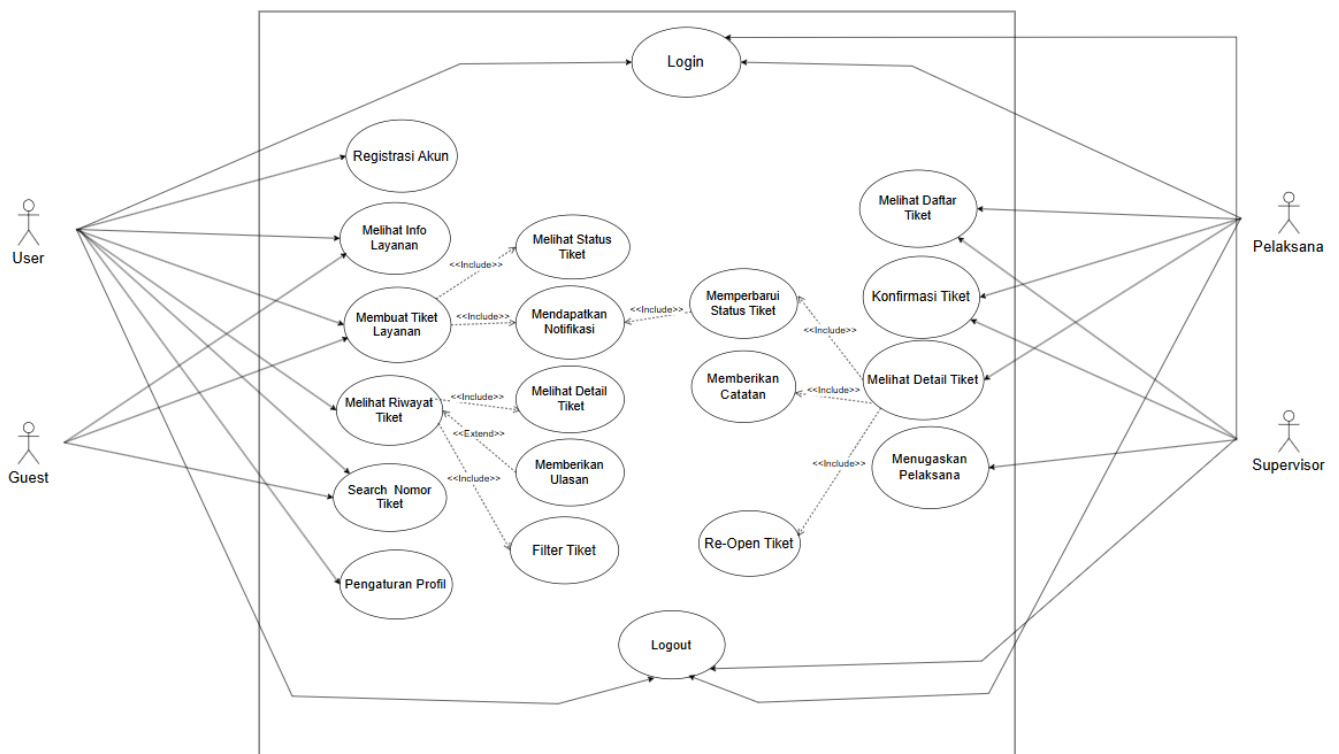
Kebutuhan Non-Fungsional atau *Non-functional* requirements dalam pengembangan perangkat lunak adalah persyaratan yang tidak berkaitan langsung dengan fungsionalitas sistem atau aplikasi, melainkan lebih berfokus pada aspek teknis, kinerja dan pengalaman pengguna. Berikut ini adalah kebutuhan non-fungsional dari aplikasi yang dijelaskan di dalam tabel 3.0.3 sebagai berikut:

Tabel 3. 3 Kebutuhan NonFungsional

Kode	Kebutuhan Fungsional
KNF-01	Antarmuka pada aplikasi ini dibuat dengan baik, mudah dipahami dan <i>responsive</i>
KNF-02	Navigasi sederhana, teks terbaca, ikon jelas, dan mudah digunakan oleh pengguna.
KNF-03	Notifikasi push diterima ≤ 3 detik setelah dikirim via <i>Firebase Cloud Messaging</i> .
KNF-04	Sistem menggunakan login dan hak akses pengguna untuk menjaga keamanan data sesuai peran pengguna.
KNF-05	Sistem dikembangkan dan digunakan sesuai dengan etika serta ketentuan penggunaan teknologi informasi yang berlaku.
KNF-06	Penggunaan sistem membantu mengurangi penggunaan dokumen fisik sehingga lebih efisien dan ramah lingkungan.

3.2.3 Diagram Use Case

Use case diagram adalah alat bantu visual yang digunakan untuk menggambarkan bagaimana pengguna berinteraksi dengan suatu sistem. Diagram ini menyederhanakan informasi sehingga lebih mudah dipahami oleh semua pihak yang terlibat dalam proyek pengembangan sistem. Berikut ini adalah *use case* diagram untuk aplikasi



Gambar 3. 3 Use Case Diagram

Gambar 3.3 diatas adalah gambar yang menunjukkan *use case* diagram dari aplikasi *B-care*. Dalam *use case* tersebut, dijelaskan bahwa terdapat 4 peran penting yang ada pada aplikasi yaitu *User*, *Guest*, *supervisor* dan pelaksana. Untuk penjelasan lebih lanjut mengenai ketiga aktor, akan dijelaskan dengan lebih rinci pada *use case* skenario di bawah ini.

3.2.4 Skenario Use Case

3.2.4.1 Skenario Use case Login

Tabel 3.4 di bawah menjelaskan skenario *use case Login* pada aplikasi *B-Care Mobile*, Dimana aktor utama adalah *User*. *Use case* ini menggambarkan bagaimana pengguna dapat masuk ke dalam sistem melalui beberapa metode *login* yang disediakan, yaitu manual (*email* dan *password*).

Tabel 3. 4 Skenario *Use Case Login*

Kode Use Case	UC 001
Nama Use Case	<i>Login</i>
Aktor	<i>User</i>
Deskripsi	Pengguna dapat masukan error ke <i>Home Screen</i> apabila kredensial/email terdaftar; jika tidak, muncul pesan
Kondisi Awal	<i>User</i> membuka halaman <i>login</i>
Kondisi Akhir	<i>User</i> berhasil masuk kedalam aplikasi
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> membuka halaman <i>Login</i>	1. Sistem menampilkan <i>form Login</i>
2. <i>User</i> memilih metode <i>login</i> , Manual (<i>Email & Password</i>).	
Cabang A: Manual	
3. <i>User</i> memasukkan <i>email & password</i> , lalu klik “ <i>Login</i> ”	3. Sistem melakukan validasi kredensial:
	• Jika valid → navigasi ke <i>Home Screen</i> dan tampil notifikasi “<i>Login Berhasil</i>”. • Jika tidak → tampil pesan “<i>Password salah</i>”.
Alur Kejadian Alternatif	
- Jika email atau <i>password</i> salah → tampilkan pesan error. - Jika akun belum terdaftar → arahkan ke halaman registrasi.	

3.2.4.2 Skenario *Use case* Registrasi akun

Tabel 3.5 dibawah merupakan skenario *use case* Registrasi Akun, fitur ini digunakan oleh *guest* yang ingin membuat akun di aplikasi *B-Care*, mulai dari mengisi form pendaftaran dan *email* sudah dipakai didalam aplikasi maka akan muncul pesan *error*.

Tabel 3. 5 Skenario *Use case* Registrasi Akun

Kode Use Case	UC 002
Nama Use Case	Registrasi Akun
Aktor	<i>Guest</i>
Deskripsi	<i>Guest</i> mendaftar akun baru agar dapat menggunakan sistem
Kondisi Awal	<i>Guest</i> belum memiliki akun
Kondisi Akhir	Akun berhasil dibuat dan dapat <i>login</i>
Alur Kejadian Normal	
Aktor	Sistem
1. <i>Guest</i> membuka halaman registrasi akun	
	2. Sistem menampilkan <i>form</i> pendaftaran.
3. Pengguna mengisi <i>form</i> registrasi	
	4. Sistem menyimpan data akun baru
	5. sistem menampilkan pesan berhasil registrasi
Alur Kejadian Alternatif	
- Jika email sudah terdaftar → tampilkan pesan <i>error</i> .	

3.2.4.3 Skenario *Use case* Melihat Info Layanan

Tabel 3.6 dibawah menunjukkan skenario *use case* Melihat Info Layanan, di mana *user* dan *guest* dapat membuka halaman layanan untuk melihat daftar layanan yang tersedia. Jika data layanan kosong, sistem akan menampilkan pesan “Layanan belum tersedia”.

Tabel 3. 6 Skenario *Use case* Melihat Info Layanan

Kode Use Case	UC 003
Nama Use Case	Melihat Info Layanan
Aktor	<i>User</i> dan <i>guest</i>
Deskripsi	<i>User</i> atau <i>guest</i> melihat informasi layanan yang tersedia
Kondisi Awal	Halaman layanan dapat diakses
Kondisi Akhir	<i>User</i> dapat melihat info layanan
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> membuka halaman layanan	
	2. Sistem menampilkan daftar layanan
Alur Kejadian Alternatif	
- Jika data layanan kosong → tampilkan pesan 'Layanan belum tersedia'.	

3.2.4.4 Skenario *Use case* Membuat Tiket Layanan

Tabel 3.7 dibawah merupakan skenario *use case* membuat tiket layanan, di mana *user* dan *guest* dapat membuat tiket pengaduan melalui form layanan. Sistem akan menyimpan data pengaduan dan menampilkan pesan berhasil. Jika data yang diisi tidak lengkap, sistem menampilkan pesan *error*.

Tabel 3. 7 Skenario *Use case* Membuat Tiket Layanan

Kode Use Case	UC 004
Nama Use Case	<i>Membuat Tiket Layanan</i>
Aktor	<i>Guest, user, supervisor, pelaksana</i>
Deskripsi	<i>Guest</i> dan <i>User</i> membuat tiket layanan untuk pengaduan
Kondisi Awal	<i>User</i> sudah <i>login</i> dan <i>guest</i> membuat tiket sebagai <i>guest</i>
Kondisi Akhir	Tiket berhasil dibuat

Alur Kejadian Normal	
Aktor	Sistem
1. <i>Actor</i> membuat tiket layanan	
	2. Sistem menampilkan <i>form</i> pendaftaran.
3. Pengguna mengisi form layanan	
	4. Sistem menyimpan data pengaduan
	5. sistem menampilkan pesan berhasil membuat tiket
Alur Kejadian Alternatif	
- Jika data tidak lengkap → tampilkan pesan <i>error</i> .	

3.2.4.5 Skenario *Use case* Melihat Status Tiket

Tabel 3.8 dibawah merupakan skenario *use case* melihat status tiket di mana *user* dapat melihat daftar tiket beserta status pengaduan yang telah dibuat. Sistem akan menampilkan informasi tiket yang dicari. Jika tiket tidak ditemukan, sistem menampilkan pesan *error*.

Tabel 3. 8 Skenario *Use case* Melihat Status Tiket

Kode Use Case	UC 005
Nama Use Case	Melihat status tiket
Aktor	<i>User</i>
Deskripsi	<i>User</i> melihat status tiket yang sudah dibuat
Kondisi Awal	<i>User</i> dan <i>Guest</i> memiliki tiket
Kondisi Akhir	<i>User</i> mengetahui status tiket
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> membuka menu status tiket	
	2. Sistem menampilkan daftar tiket dengan statusnya
3. <i>User</i> mencari tiket yang ingin dilihat statusnya	
	4. Sistem menampilkan tiket yang dicari
Alur Kejadian Alternatif	
- Jika tiket tidak ditemukan → tampilkan pesan <i>error</i> .	

3.2.4.6 Skenario *Use case* Melihat Riwayat Tiket

Tabel 3.9 dibawah merupakan skenario *use case* melihat Riwayat tiket, *user*, pada awalnya *user* membuka halaman Riwayat tiket lalu sistem menampilkan halaman Riwayat tiket dan *user search* tiket yang ingin dilihat progressnya.

Tabel 3. 9 Skenario *Use Case* Melihat Riwayat Tiket

Kode Use Case	UC 006
Nama Use Case	Melihat Riwayat Tiket
Aktor	User
Deskripsi	User melihat riwayat tiket yang pernah dibuat
Kondisi Awal	User sudah login
Kondisi Akhir	User dapat melihat riwayat tiket
Alur Kejadian Normal	
Aktor	Sistem
1. User membuka menu riwayat tiket	
	2. Sistem menampilkan daftar tiket tiket yang sudah selesai prosesnya
3. User mencari tiket yang ingin dilihat	
	4. Sistem menampilkan riwayat tiket yang dicari
Alur Kejadian Alternatif	
- Jika tiket tidak ditemukan → tampilkan pesan tiket tidak ditemukan.	

3.2.4.7 Skenario *Use case* Memberikan Ulasan

Tabel 3.10 dibawah merupakan skenario *use case* memberikan ulasan, *user* dapat memberikan ulasan terhadap aplikasi dari layanan yang disediakan

Tabel 3. 10 Skenario *Use Case* Memberikan Ulasan

Kode Use Case	UC 007
Nama Use Case	Memberikan ulasan
Aktor	<i>User</i>
Deskripsi	<i>User</i> memberikan rating terhadap layanan

Kondisi Awal	Tiket sudah selesai di proses
Kondisi Akhir	Ulasan berhasil disimpan
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> membuka menu riwayat tiket	
	2. Sistem menampilkan daftar tiket tiket yang sudah selesai prosesnya
3. <i>User</i> mencari tiket yang ingin dilihat	
	4. Sistem menampilkan riwayat tiket yang dicari
Alur Kejadian Alternatif	
- Jika user mencoba memberikan rating di tiket yang belum selesai diproses, sistem menampilkan peringatan “Hanya tiket yang sudah selesai yang bisa diulas.”	

3.2.4.8 Skenario *Use case* Lupa Password

Tabel 3.11 dibawah merupakan skenario *use case* lupa password, user dapat memilih fitur lupa password lalu sistem akan menampilkan form input email dan user memasukkan email yang sudah terdaftar, jika emailnya sama dengan yang ada di aplikasi maka sistem akan mengirimkan password user ke email user.

Tabel 3. 11 Skenario *Use Case* Lupa Password

Kode Use Case	UC 008
Nama Use Case	Lupa Password
Aktor	User
Deskripsi	User melakukan reset password
Kondisi Awal	User memiliki akun dengan email valid
Kondisi Akhir	Password berhasil diganti
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> memilih menu lupa password	
	2. Sistem menampilkan halaman input email
3. User memasukkan email yang sudah terdaftar	
	4. Sistem mengirim link reset password ke email

5. <i>User</i> membuka link dan mengganti <i>password</i>	
Alur Kejadian Alternatif	
- Jika email tidak salah → tampilkan pesan <i>error</i> .	

3.2.4.9 Skenario *Use case Search Nomor Tiket*

Tabel 3.12 dibawah merupakan skenario *use case search* nomor tiket, user dan guest dapat mencari tiket berdasarkan nomor tiket,

Tabel 3. 12 Skenario *Use Case Search* Nomor Tiket

Kode Use Case	UC 009
Nama Use Case	Search nomor tiket
Aktor	User dan guest
Deskripsi	User dan Guest mencari tiket berdasarkan nomor tiket
Kondisi Awal	Nomor tiket tersedia
Kondisi Akhir	Tiket berhasil ditemukan
Alur Kejadian Normal	
Aktor	Sistem
1. User /Guest membuka menu search tiket	
	2. Sistem membuka halaman search
3. User/Guest memasukkan nomor tiket	
	4. Sistem mencari dan menampilkan tiket
Alur Kejadian Alternatif	
- Jika tiket tidak ditemukan → tampilkan pesan tiket tidak ada	

3.2.4.10 Skenario Use case Edit Profile

Tabel 3.13 dibawah merupakan skenario *use case* edit *profile*, *user* dapat mengedit *profile* nya seperti mengganti *email* atau yang lainnya.

Tabel 3. 13 Skenario Use Case Edit Profil

Kode Use Case	UC 010
Nama Use Case	Edit Profile
Aktor	<i>User</i>
Deskripsi	<i>User</i> mengelola profil akun
Kondisi Awal	<i>User login</i>
Kondisi Akhir	Profil berhasil diperbarui
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> membuka menu profil	
	2. Sistem menampilkan halaman profil
3. <i>User</i> dapat mengubah data diri pribadi	
	4. Sistem menyimpan perubahan
Alur Kejadian Alternatif	
- Pesan error data tidak berhasil disimpan	

3.2.4.11 Skenario Use case Notifikasi

Tabel 3.14 dibawah merupakan skenario *use case notifikasi user* dan *guest*, kedua *actor* ini akan mendapatkan notifikasi mengenai tiket layanan yang sudah dibuat.

Tabel 3. 14 Skenario Use Case Notifikasi

Kode Use Case	UC 011
Nama Use Case	Notifikasi
Aktor	<i>User</i> dan <i>guest</i>
Deskripsi	<i>User</i> menerima notifikasi terkait tiket
Kondisi Awal	<i>User</i> dan <i>guest</i> memiliki tiket
Kondisi Akhir	<i>User</i> mendapatkan informasi terbaru
Alur Kejadian Normal	
Aktor	Sistem

	1. Sistem mendeteksi perubahan status tiket
	2. Sistem mengirimkan notifikasi ke <i>user</i>
2. <i>User</i> membaca notifikasi	
Alur Kejadian Alternatif	
- Jika gagal mengirim notifikasi → <i>User</i> tetap bisa melihat status tiket di menu riwayat.	

3.2.4.12 Skenario *Use case Logout*

Tabel 3.15 dibawah merupakan skenario *use case logout*, *user* memilih menu *logout* dan sistem mengakhiri sesi *login*.

Tabel 3. 15 Skenario *Use Case Logout*

Kode Use Case	UC 012
Nama Use Case	<i>Logout</i>
Aktor	<i>User</i> ,Pelaksana,Supervisor
Deskripsi	<i>User</i> atau Pelaksana,Supervisor keluar dari sistem
Kondisi Awal	<i>User</i> atau Pelaksana,Supervisor <i>login</i>
Kondisi Akhir	<i>User</i> atau Pelaksana,Supervisor keluar dari sistem
Alur Kejadian Normal	
Aktor	Sistem
1. <i>User</i> atau Pelaksana,Supervisor memilih menu <i>logout</i>	
	2. Sistem mengakhiri sesi <i>login</i>
Alur Kejadian Alternatif	
Sistem menampilkan pesan “Terjadi kesalahan saat keluar dari akun. Silakan coba lagi.” tanpa menghapus sesi pengguna.	

3.2.4.13 Skenario Use case Login pelaksana dan supervisor

Tabel 3.16 dibawah merupakan skenario *use case login* pelaksana dan *supervisor*, Dimana pelaksana atau *supervisor* dapat memasukan *email* dan *password* yang sudah ada, jika berhasil akan diarahkan kehalaman *dashboard*.

Tabel 3. 16 Skenario Use Case Login pelaksana atau supervisor

Kode Use Case	UC 013
Nama Use Case	Login pelaksana dan supervisor
Aktor	Pelaksana dan supervisor
Deskripsi	Pelaksana dan supervisor masuk kedalam sistem
Kondisi Awal	Pelaksana dan supervisor membuka halaman login
Kondisi Akhir	Pelaksana dan supervisor berhasil masuk kedalam aplikasi
Alur Kejadian Normal	
Aktor	Sistem
1. Pelaksana atau supervisor membuka halaman Login	1. Sistem menampilkan form Login
2. Pelaksana atau supervisor memasukkan email dan password	
	3. Sistem validasi akun
4. Pelaksana dan supervisor masuk kedalam dashboard	
Alur Kejadian Alternatif	
- Jika Pelaksana atau supervisor memasukkan email atau password salah, sistem menampilkan pesan error “Email atau password salah” dan meminta Pelaksana atau supervisor mencoba lagi.	

3.2.4.14 Skenario Use case Pelaksana dan supervisor Melihat Daftar Tiket

Tabel 3.17 di bawah merupakan skenario *use case* Melihat Daftar Tiket. Pada skenario ini, pelaksana dapat melihat seluruh tiket yang masuk. Saat pelaksana membuka halaman daftar tiket, sistem akan menampilkan data tiket secara lengkap.

Tabel 3. 17 Skenario Use Case supervisor Melihat Daftar Tiket

Kode Use Case	UC 014
Nama Use Case	Pelaksana dan supervisor Melihat Daftar Tiket
Aktor	Pelaksana dan supervisor
Deskripsi	Pelaksana dan supervisor melihat tiket yang di proses
Kondisi Awal	Pelaksana dan supervisor login

Kondisi Akhir	Pelaksana dan <i>supervisor</i> memilih tiket yang ingin di <i>update</i>
Alur Kejadian Normal	
Aktor	Sistem
1. Pelaksana atau <i>supervisor</i> membuka menu daftar tiket	
	2. sistem menampilkan halaman Daftar tiket
3. Pelaksana atau <i>supervisor</i> memilih tiket yang ingin di <i>update</i>	
	4. Sistem menampilkan halaman tiket
Alur Kejadian Alternatif	
- Jika terjadi gagal memuat daftar tiket , sistem menampilkan notifikasi “Gagal memuat daftar tiket, silakan coba lagi” dan tetap menampilkan daftar tiket	

3.2.4.15 Skenario *Use case* Pelaksana atau *Supervisor* melihat detail tiket

Tabel 3.18 dibawah merupakan skenario *use case* melihat detail tiket, di mana pelaksana atau *supervisor* yang sudah *login* dapat membuka daftar tiket, memilih salah satu tiket, dan sistem akan menampilkan detail tiket tersebut.

Tabel 3. 18 Skenario *Use Case* pelaksana atau *supervisor* melihat detail tiket

Kode Use Case	UC 015
Nama Use Case	Pelaksana atau <i>supervisor</i> Melihat detail tiket
Aktor	Pelaksana atau <i>supervisor</i>
Deskripsi	Pelaksana atau <i>supervisor</i> mengecek detail tiket
Kondisi Awal	Pelaksana atau <i>supervisor login</i>
Kondisi Akhir	Pelaksana atau <i>supervisor</i> dapat melihat detail tiket
Alur Kejadian Normal	
Aktor	Sistem
1. Pelaksana atau <i>supervisor</i> membuka daftar tiket	
	2. sistem menampilkan halaman daftar tiket
3. Pelaksana atau <i>supervisor</i> memilih tiket	
	4.Sistem menampilkan detail tiket
Alur Kejadian Alternatif	

- Jika terjadi gagal memuat detail tiket , sistem menampilkan notifikasi “Gagal memuat detail tiket, silakan coba lagi” dan tetap menampilkan daftar tiket.

3.2.4.16 Skenario *Use Case* Pelaksana atau *Supervisor* update status tiket

Tabel 3.19 di bawah merupakan skenario *use case* Pelaksana atau *supervisor Update* Status Tiket, di mana pelaksana yang telah *login* dapat memilih tiket dari daftar, mengubah status tiket sesuai progres penanganan, dan sistem akan menyimpan serta memperbarui data status tiket tersebut.

Tabel 3. 19 Skenario *Use Case* pelaksana atau *supervisor update* status tiket

Kode Use Case	UC 016
Nama Use Case	Pelaksana atau <i>supervisor Update</i> Status Tiket
Aktor	Pelaksana atau <i>supervisor</i>
Deskripsi	Pelaksana atau <i>supervisor</i> memperbarui status tiket
Kondisi Awal	Pelaksana atau <i>supervisor login</i> dan tiket tersedia
Kondisi Akhir	Status Tiket berhasil diperbarui
Alur Kejadian Normal	
Aktor	Sistem
1. Pelaksana atau <i>supervisor</i> membuka daftar tiket	
	2. sistem menampilkan halaman daftar tiket
3. Pelaksana atau <i>supervisor</i> memilih tiket	
4. Pelaksana atau <i>supervisor</i> merubah status tiket	
	5.Sistem menyimpan perubahan
6. Pelaksana atau <i>supervisor</i> menambahkan komentar	
	7. Sistem menyimpan dan mengirim komentar ke user
8.Pelaksana atau <i>supervisor</i> menambahkan <i>viewer</i>	
	9.Sistem menampilkan nama <i>viewer</i>
Alur Kejadian Alternatif	
- Jika terjadi kegagalan penyimpanan (misal koneksi server gagal), sistem menampilkan notifikasi “Gagal memperbarui status” dan menyimpan perubahan sementara untuk dicoba ulang.	

3.2.4.17 Skenario *Use Case* Pelaksana atau *Supervisor* konfirmasi tiket

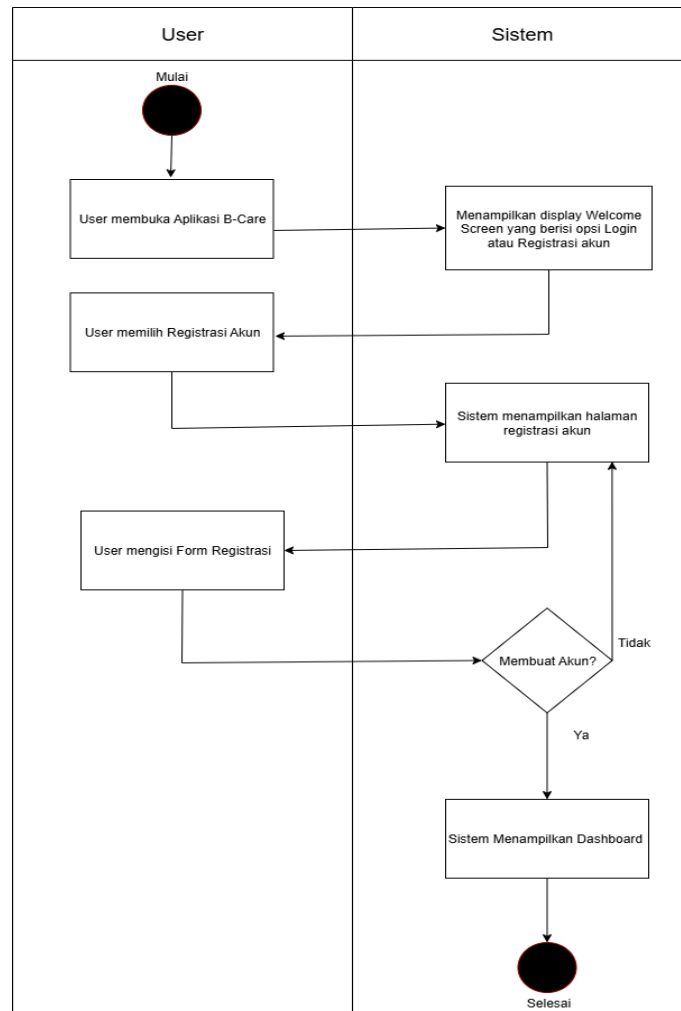
Tabel 3.20 di bawah merupakan skenario *use case* Pelaksana atau *supervisor Update Status Tiket*, di mana pelaksana yang telah *login* dapat memilih tiket dari daftar, mengubah status tiket sesuai progres penanganan, dan sistem akan menyimpan serta memperbarui data status tiket tersebut.

Tabel 3. 20 Skenario *Use Case* pelaksana atau *supervisor* konfirmasi tiket

Kode Use Case	UC 016
Nama Use Case	Pelaksana atau <i>supervisor</i> konfirmasi penanganan Tiket
Aktor	Pelaksana atau <i>supervisor</i>
Deskripsi	Pelaksana atau <i>supervisor</i> konfirmasi penanganan tiket
Kondisi Awal	Pelaksana atau <i>supervisor login</i> dan tiket tersedia
Kondisi Akhir	Status Tiket sedang ditangani atau selesai
Alur Kejadian Normal	
Aktor	Sistem
1. Pelaksana atau <i>supervisor</i> membuka daftar tiket	
	2. sistem menampilkan halaman daftar tiket
3. Pelaksana atau <i>supervisor</i> memilih tiket	
4. Pelaksana atau <i>supervisor</i> konfirmasi penanganan tiket	
	5.Sistem menyimpan perubahan
Alur Kejadian Alternatif	
- Jika terjadi kegagalan penyimpanan (misal koneksi server gagal), sistem menampilkan notifikasi “Gagal memperbarui status” dan menyimpan perubahan sementara untuk dicoba ulang.	

3.2.5 Activity diagram

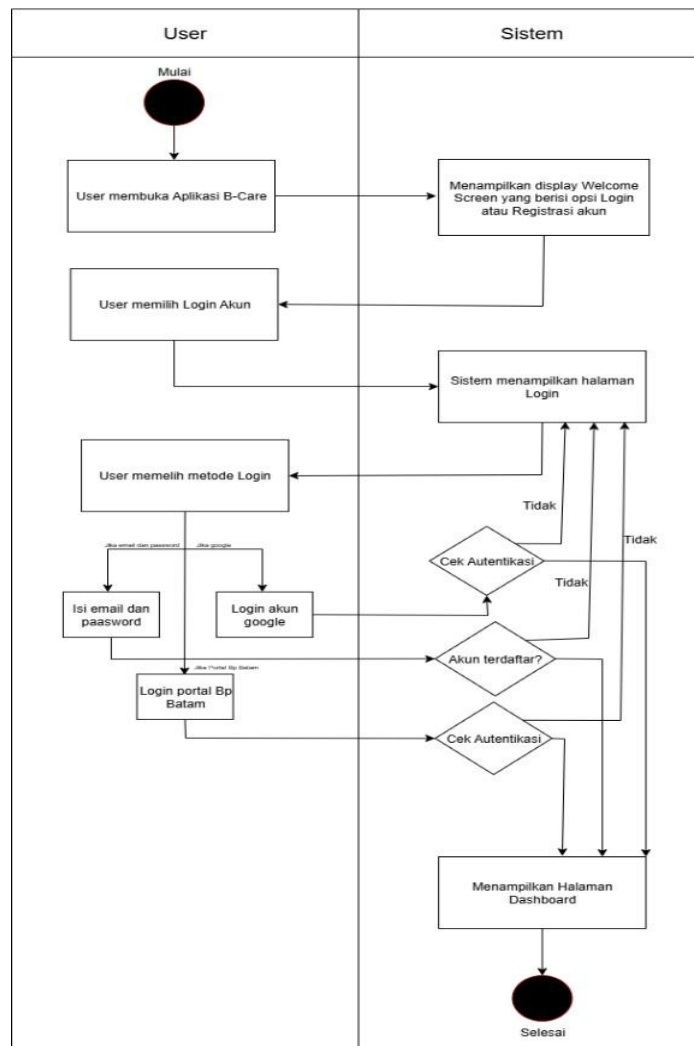
3.2.5.1 Activity Diagram Registrasi Akun



Gambar 3. 4 Activity Diagram Registrasi Akun

Gambar 3.4 di atas merupakan *Activity* Diagram proses registrasi akun. Proses ini digunakan oleh *guest* yang ingin membuat akun pada *aplikasi B-Care Mobile*. *Guest* mengisi form registrasi berisi data diri seperti nama, email, Nik dan *password*. Setelah data divalidasi, sistem akan menyimpan informasi pengguna baru ke dalam basis data dan menampilkan bahwa registrasi berhasil.

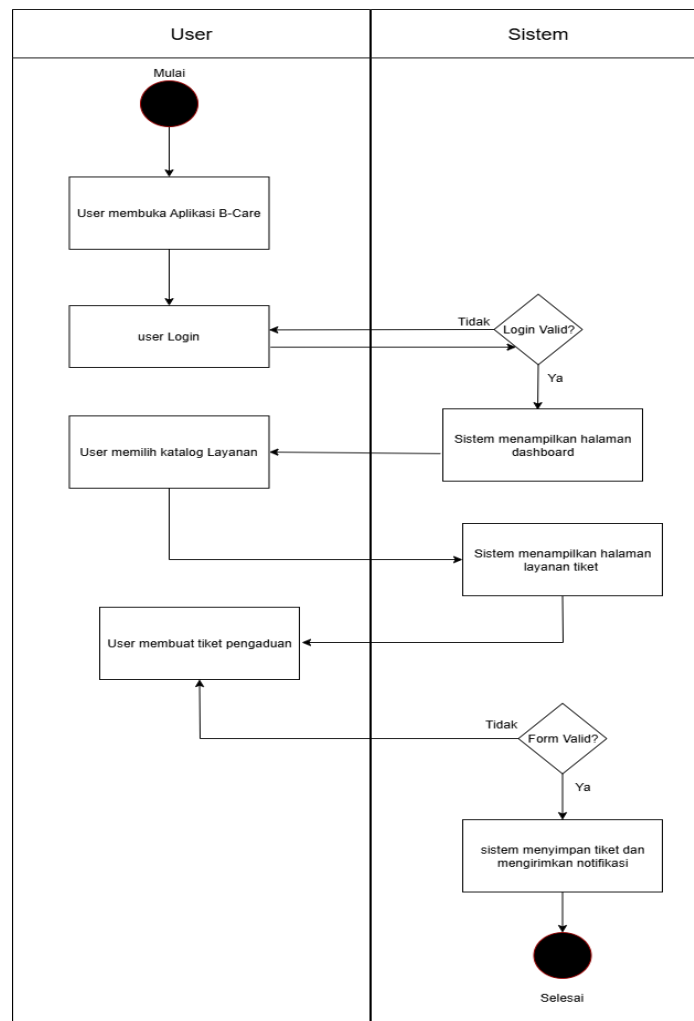
3.2.5.2 Activity Diagram Login



Gambar 3. 5 Activity Diagram Login

Gambar 3.5 merupakan Activity Diagram proses Login. Proses ini digunakan oleh *user* untuk masuk ke aplikasi melalui beberapa metode login seperti *email*, akun *Google*, atau portal BP Batam. Setelah sistem melakukan autentikasi dan data dinyatakan valid, pengguna diarahkan ke halaman *dashboard* sesuai perannya.

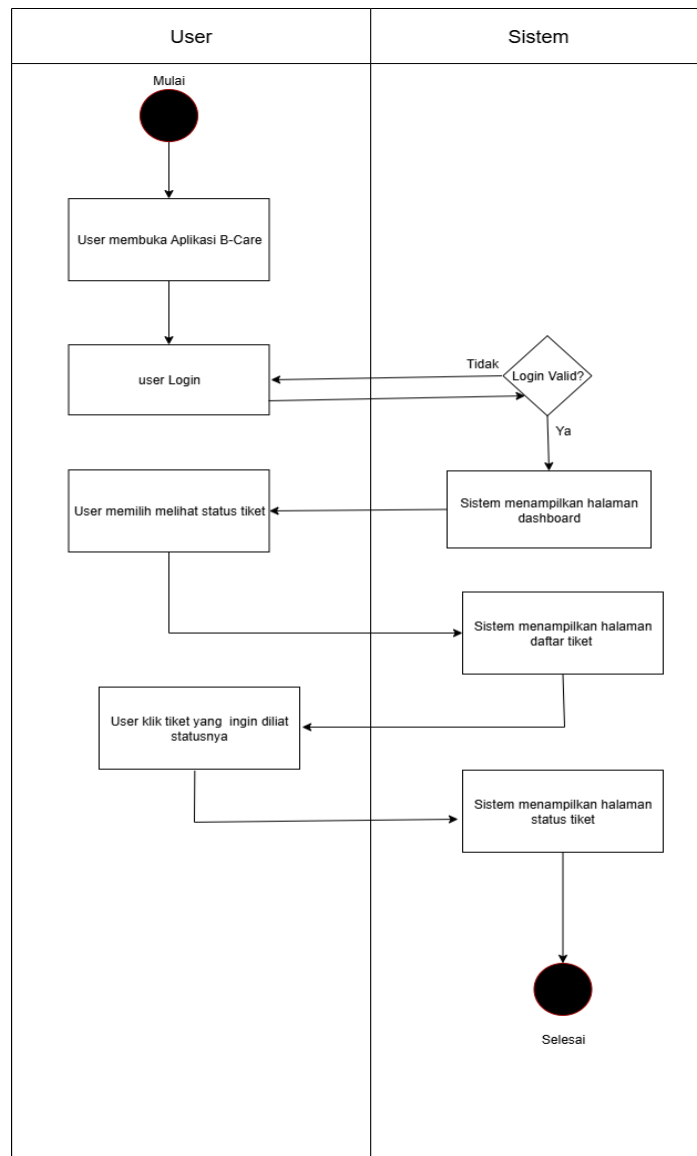
3.2.5.3 Activity Diagram Melihat Layanan pengaduan



Gambar 3. 6 Activity Diagram Melihat Layanan pengaduan

Gambar 3.6 di atas merupakan *Activity* Diagram proses melihat layanan pengaduan. Pada proses ini, *user* dapat mengakses halaman Layanan Pengaduan untuk melihat daftar layanan yang tersedia. Sistem akan menampilkan berbagai kategori layanan beserta deskripsinya, sehingga pengguna dapat memilih layanan yang sesuai dengan jenis pengaduan yang ingin diajukan.

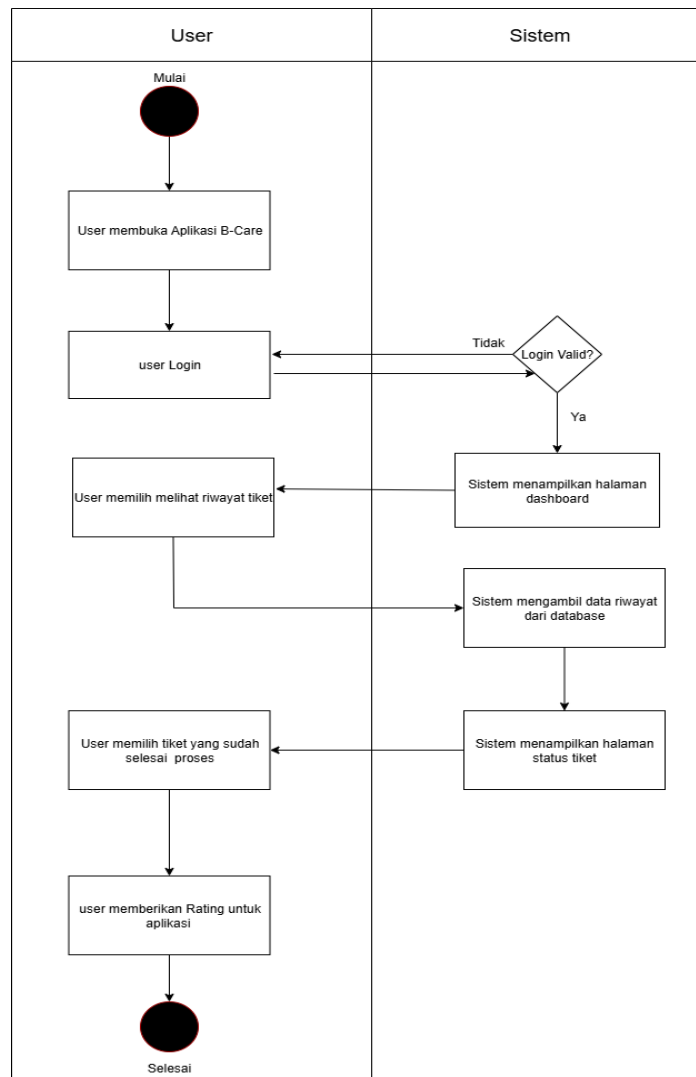
3.2.5.4 Activity Diagram Status Tiket



Gambar 3. 7 Activity Diagram Status Tiket

Gambar 3.7 di atas merupakan *Activity Diagram* proses melihat status tiket. Pada proses ini, *user* dapat memantau perkembangan laporan atau pengaduan yang telah dibuat melalui halaman Status. Sistem akan menampilkan daftar tiket beserta detail statusnya, seperti tiket baru, dalam proses, atau telah selesai, sehingga pengguna dapat mengetahui sejauh mana tindak lanjut dari laporan yang telah dikirimkan.

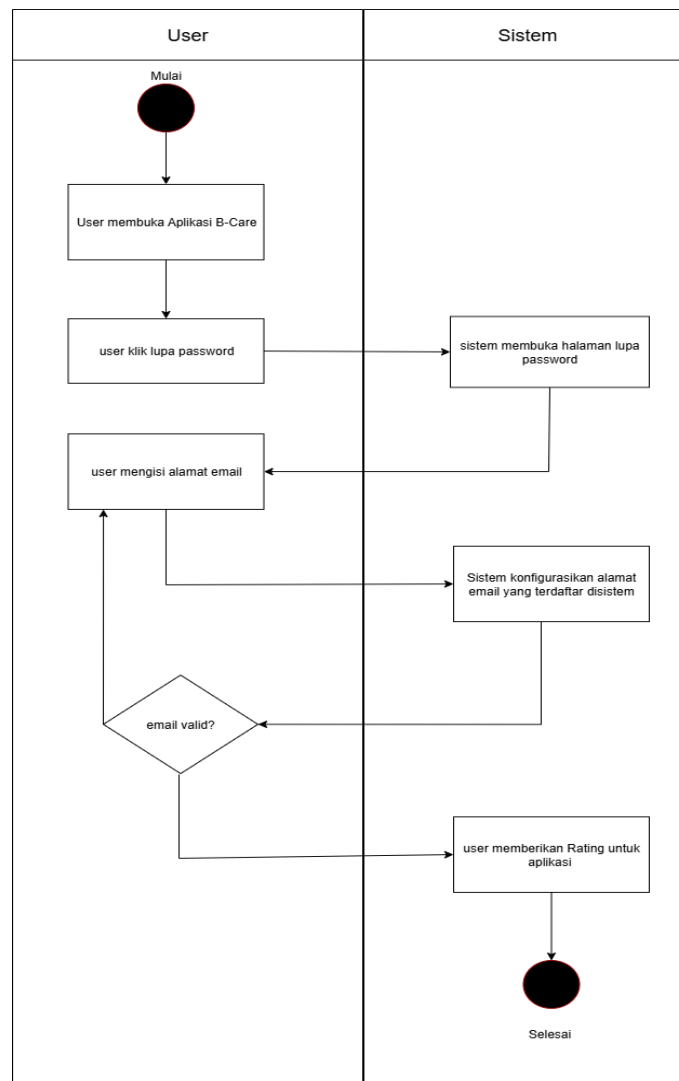
3.2.5.5 Activity Diagram Riwayat Tiket



Gambar 3. 8 Activity Diagram Riwayat Tiket

Gambar 3.8 merupakan *Activity Diagram* proses Riwayat Tiket. Fitur ini *user* melihat Riwayat tiket yang pernah dibuat pada aplikasi. Melalui halaman ini, pengguna dapat memeriksa tiket yang sudah selesai diproses dan memberikan *rating* sebagai bentuk penilaian terhadap layanan yang diterima.

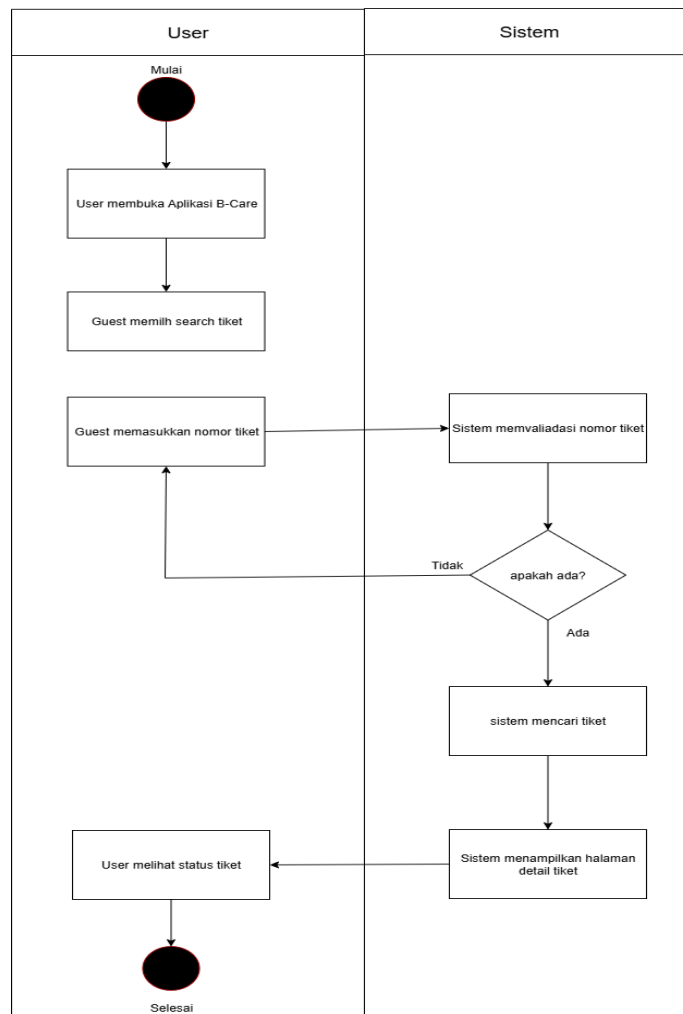
3.2.5.6 Activity Diagram Lupa Password



Gambar 3. 9 Activity Diagram Lupa Password

Gambar 3.9 diatas merupakan gambar *activity* diagram lupa *password* lupa *password* dapat digunakan jika user lupa dengan *password* lama,maka sistem akan mengirimkan *password* tersebut ke *email user*.

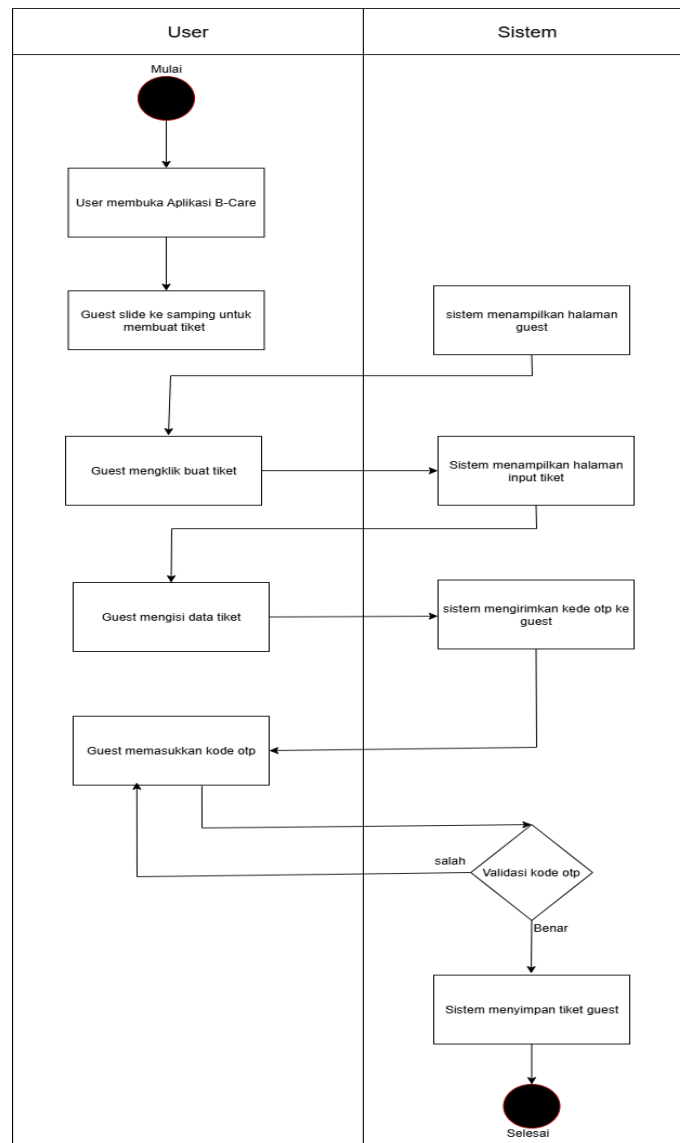
3.2.5.7 Activity Diagram Search



Gambar 3. 10 Activity Diagram Search

Gambar 3.10 diatas merupakan gambar *Activity diagram Search*, *activity* jika *guest* sudah membuat tiket dan ingin mengetahui progress.

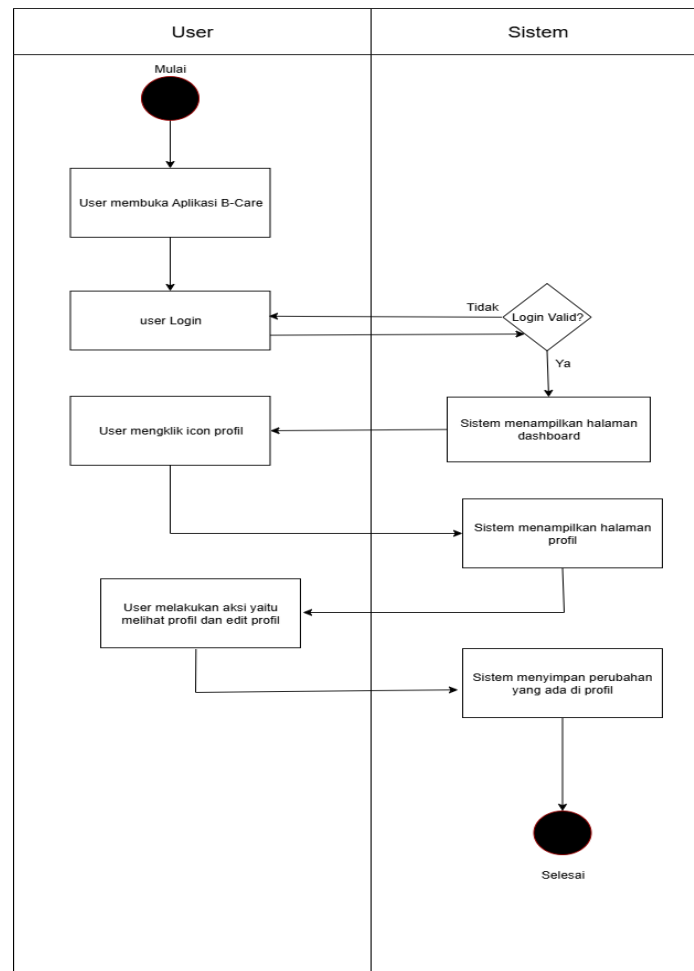
3.2.5.8 Activity Diagram Buat Tiket Sebagai Guest



Gambar 3. 11 Activity Diagram Buat Tiket Sebagai Guest

Gambar 3.11 merupakan *activity* diagram guest membuat tiket, mulai dari *welcome page* guest slide ke samping maka sistem akan menampilkan katalog kategori layanan dan guest membuat tiket layanan.

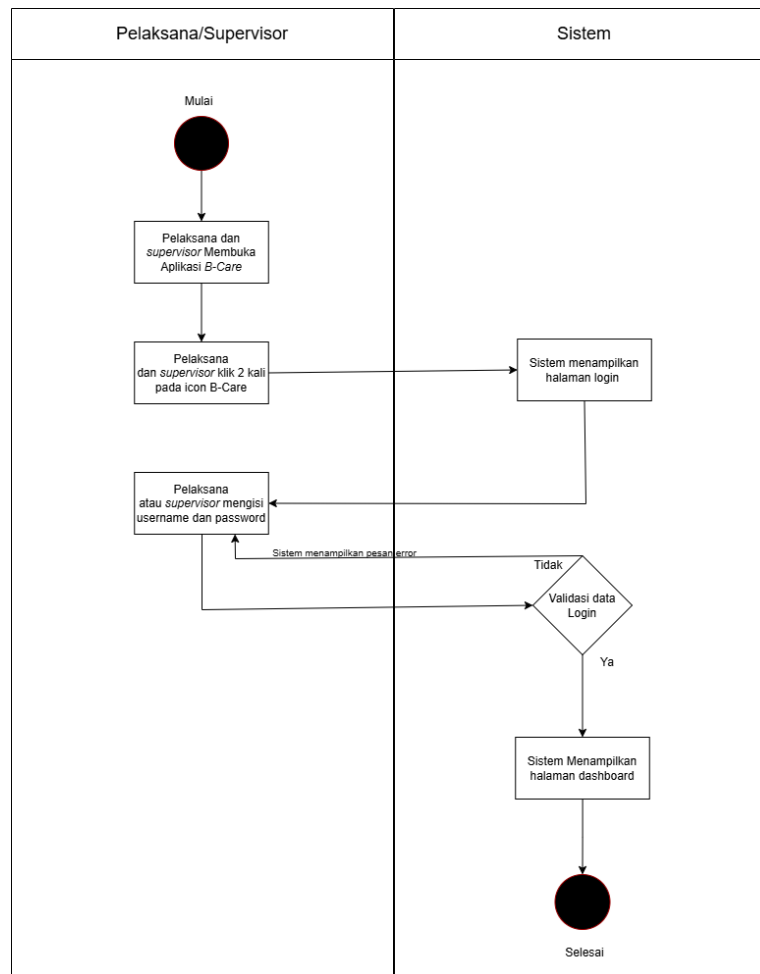
3.2.5.9 Activity Diagram Edit Profile



Gambar 3. 12 Activity Diagram Edit Profile

Gambar 3.12 di atas merupakan *Activity Diagram* proses *Edit Profile*. Pada fitur ini, *user* dapat memperbarui informasi pribadinya seperti nama, alamat *email*, atau kata sandi melalui halaman *Edit Profile*. Setelah melakukan perubahan data, sistem akan menyimpan pembaruan tersebut ke dalam *database* dan menampilkan notifikasi bahwa *profile* berhasil diperbarui.

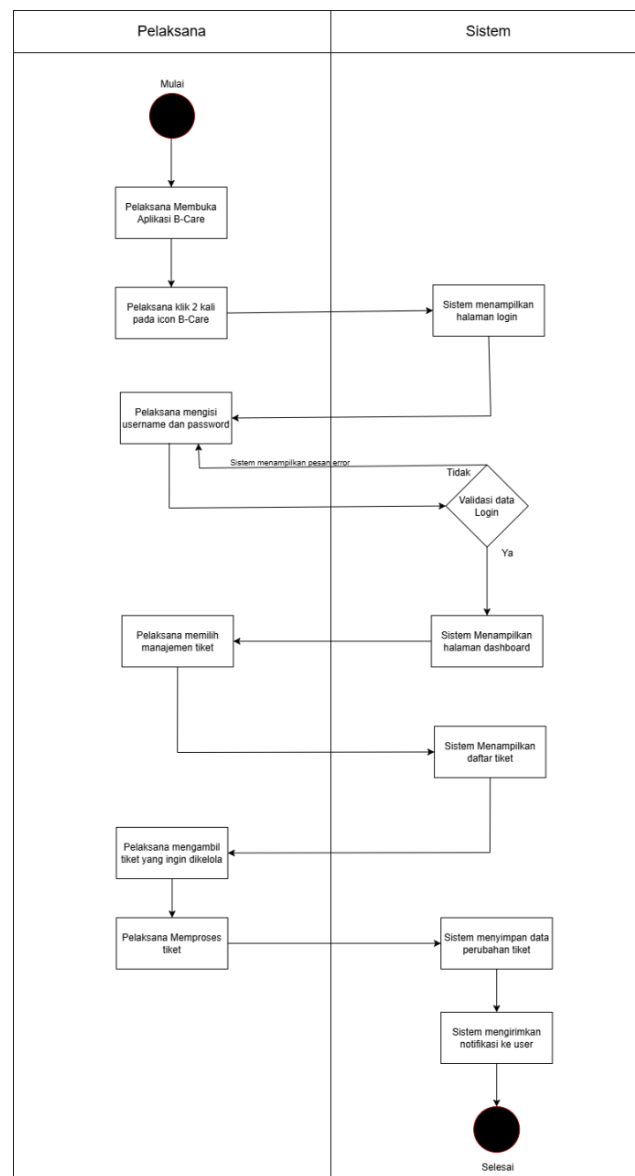
3.2.5.10 Activity Diagram Login



Gambar 3. 13 Activity Diagram Login pelaksana dan supervisor

Gambar 3.13 merupakan Activity Diagram proses Login. Proses ini digunakan oleh untuk masuk ke dalam sistem *B-Care Mobile*. Aktivitas dimulai ketika pelaksana membuka aplikasi dan memilih opsi login. Selanjutnya, sistem menampilkan halaman tempat pelaksana atau supervisor memasukkan *username* dan *password*.

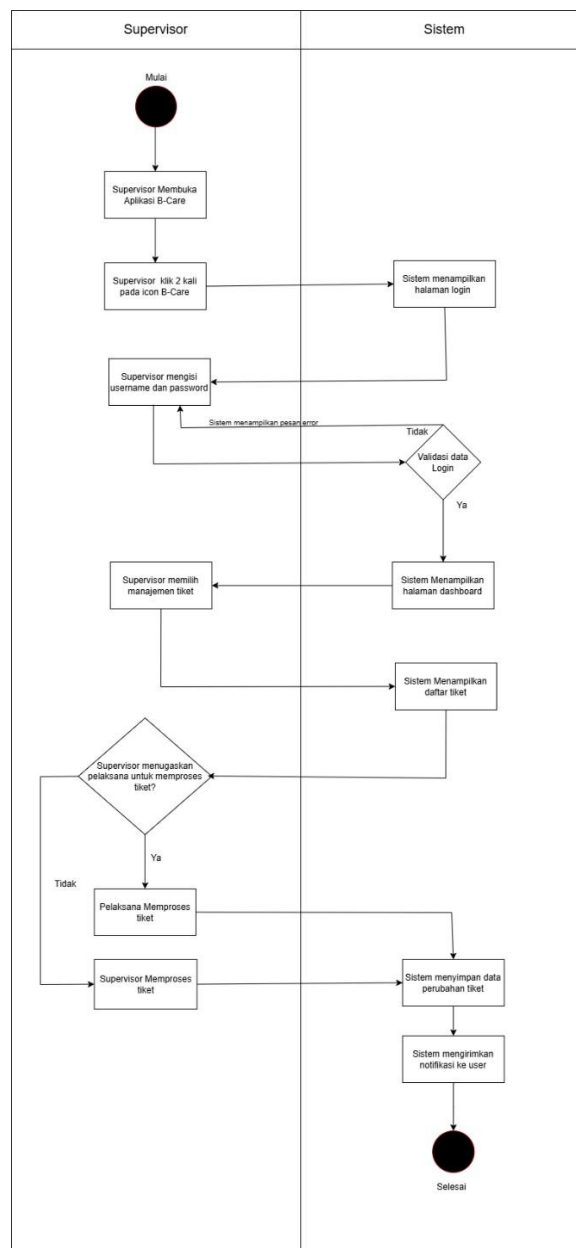
3.2.5.11 Activity Diagram Pelaksana Kelola Tiket



Gambar 3. 14 Activity Diagram Pelaksana Kelola Tiket

Gambar 3.14 merupakan Activity Diagram Kelola Tiket. Diagram ini menjelaskan proses dalam mengelola tiket laporan pada sistem *B-Care Mobile*. Pelaksana atau *supervisor* dapat melihat daftar tiket, memeriksa detail laporan, memperbarui status tiket, serta menambahkan keterangan tindak lanjut. Fitur ini membantu pelaksana menangani laporan secara cepat dan terstruktur.

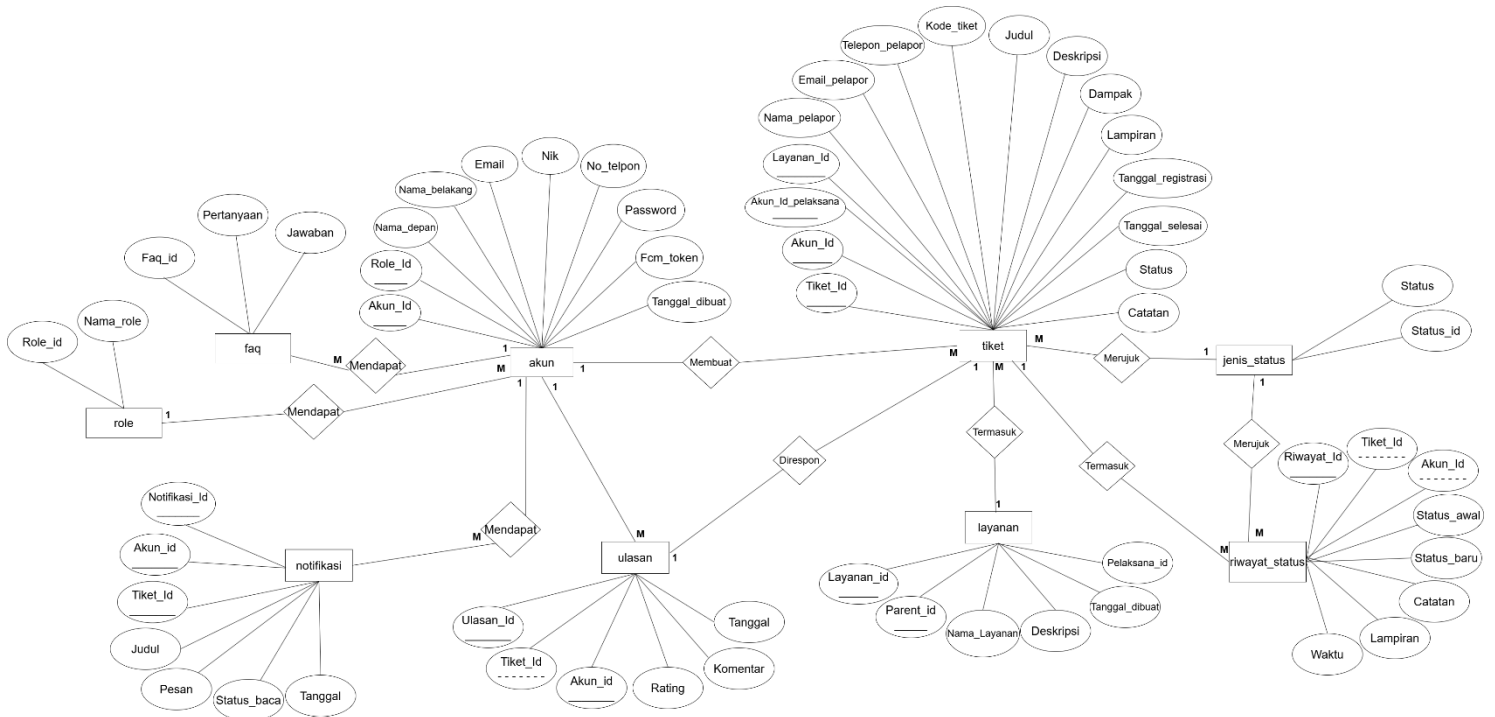
3.2.5.12 Activity Diagram Supervisor Kelola Tiket



Gambar 3. 15 Activity Diagram Supervisor Kelola Tiket

Gambar 3.15 merupakan *Activity Diagram Supervisor* Kelola tiket. Diagram ini menjelaskan proses *supervisor* dalam mengelola tiket laporan pada sistem *B-Care Mobile*. *Supervisor* dapat melihat daftar tiket, memeriksa detail laporan, memperbarui status tiket, serta menambahkan keterangan tindak lanjut. Fitur ini membantu *supervisor* menangani laporan secara cepat dan terstruktur.

3.2.6 ER Diagram

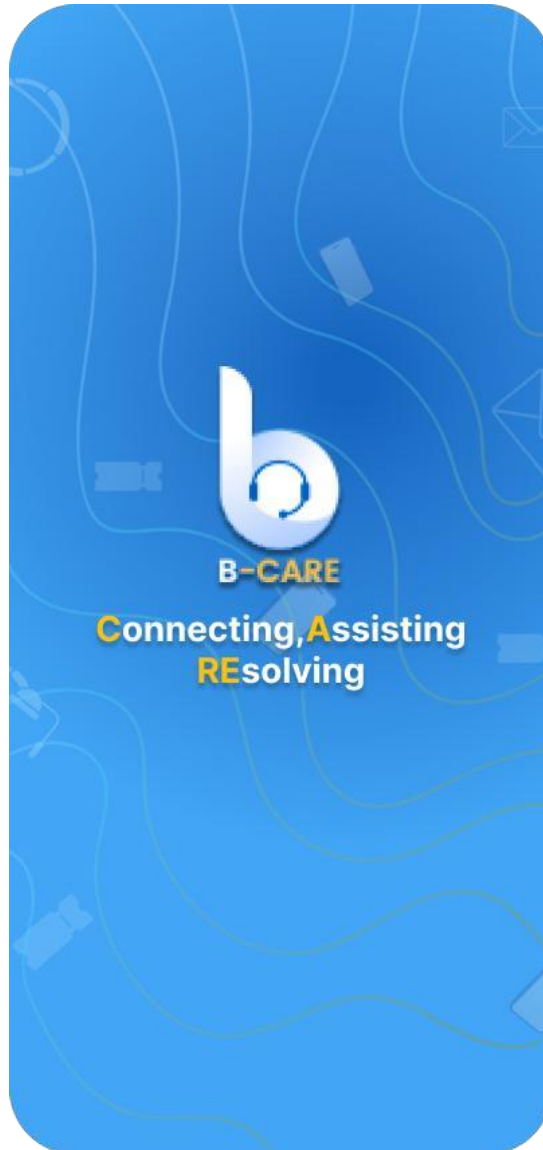


Gambar 3. 16 ER Diagram

Gambar 3.16 merupakan *ER Diagram*. Diagram ini menggambarkan hubungan antar entitas dalam sistem *B-Care Mobile*. Entitas utama terdiri dari Akun, Tiket, Kategori, dan Riwayat tiket, notifikasi dan ulasan. Masing-masing entitas saling terhubung melalui relasi yang menggambarkan proses pembuatan laporan, pengelolaan tiket, serta pemantauan status oleh pengguna dan pelaksana. *ER Diagram* ini menjadi dasar perancangan struktur *database* pada sistem. *ER Diagram* ini bersifat konseptual, karena pengembangan aplikasi mobile dilakukan tanpa mengakses database internal *B-Care*. Diagram ini hanya digunakan sebagai acuan perancangan struktur data pada aplikasi mobile agar sesuai dengan logika dan alur sistem yang diakses melalui *API*.

3.2.7. Perancangan Antarmuka *Mock up*

3.2.7.1 *Splash page*



Gambar 3. 17 *Splash page*

Gambar 3.17 diatas merupakan gambar *Splash page* aplikasi *B-Care*, yang menampilkan logo dan nama aplikasi saat pertama kali dijalankan. Halaman ini berfungsi sebagai layar pembuka sebelum pengguna diarahkan ke halaman berikutnya.

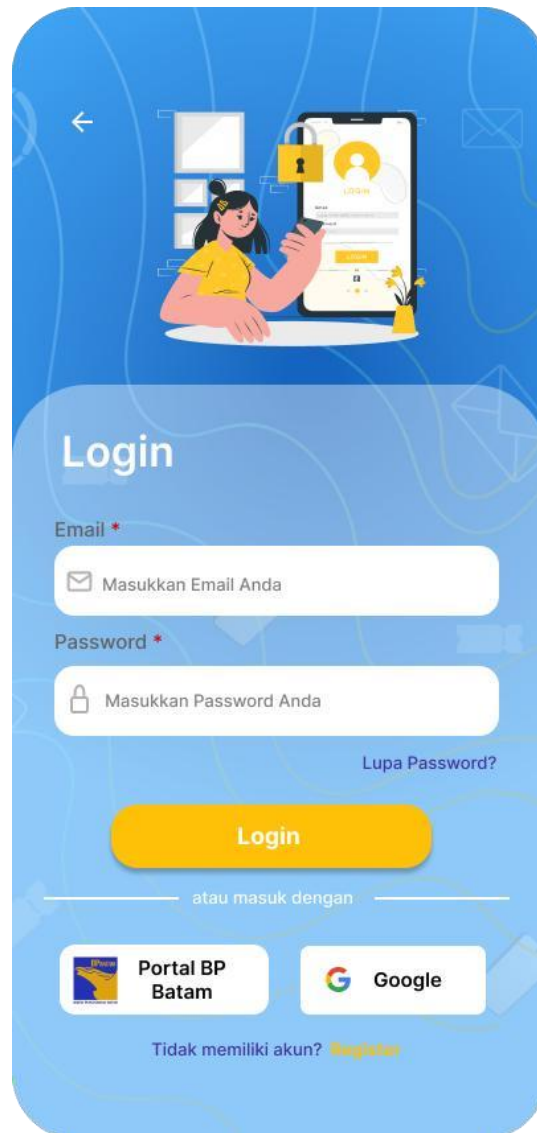
3.2.7.2 Welcome Page



Gambar 3. 18 Welcome Page

Gambar 3.18 di atas merupakan tampilan *Welcome Page* aplikasi *B-Care*. Halaman ini berfungsi untuk memperkenalkan aplikasi kepada pengguna dengan menampilkan informasi singkat serta memberikan pilihan untuk melanjutkan ke halaman *login* atau menjelajahi aplikasi sebagai *guest*.

3.2.7.3 Login Page



Gambar 3. 19 Login Page

Gambar 3.19 di atas merupakan tampilan *Login Page* aplikasi *B-Care*. Halaman ini digunakan oleh pengguna untuk masuk ke dalam sistem dengan memasukkan *email* dan *password* yang terdaftar, *login* dengan *google* atau portal Bp Batam Setelah berhasil *login*, pengguna akan diarahkan ke halaman utama sesuai dengan perannya.

3.2.7.4 Registrasi Page



The image shows a mobile application registration screen with a blue background and rounded corners. At the top, there is a white back arrow icon. Below it, an illustration depicts a person in a yellow shirt and black pants interacting with a large smartphone. The phone screen displays a registration form with fields for 'Nama Depan', 'Nama Belakang', 'Email', and 'NIK'. A yellow padlock icon is positioned above the phone, and a yellow thumbs-up icon is to its left. The background also features faint icons of a mail envelope and a document. The title 'Registrasi Akun' is centered in white text. Below the title, there are four white input fields: the first is labeled 'Nama Depan' with a person icon, the second is labeled 'Nama Belakang' with a person icon, the third is labeled 'Email' with an envelope icon, and the fourth is labeled 'NIK'. At the bottom center, there is a large yellow circular button with a white right-pointing arrow.

Registrasi Akun

Nama Depan

Nama Belakang

Email

NIK

Gambar 3. 20 Register Page 1



Gambar 3. 21 *Register Page 2*

Gambar 3.20 dan 3.21 di atas merupakan tampilan *Register Page 1* dan *2* pada aplikasi *B-Care*. Halaman ini digunakan untuk proses pendaftaran akun baru. Pada *Register Page 1*, pengguna mengisi data dasar seperti nama dan *email*, sedangkan pada *Register Page 2*, pengguna melengkapi informasi tambahan seperti kata sandi dan konfirmasi data sebelum akun berhasil dibuat.

3.2.7.5 Dashboard Page



Gambar 3. 22 Dashboard Page

Gambar 3.22 di atas merupakan tampilan *Dashboard Page* aplikasi *B-Care*. Halaman ini menjadi beranda utama setelah pengguna *login*, menampilkan ringkasan fitur utama seperti daftar layanan, status tiket, dan notifikasi. *Dashboard* memudahkan pengguna untuk mengakses berbagai menu dan informasi penting secara cepat.

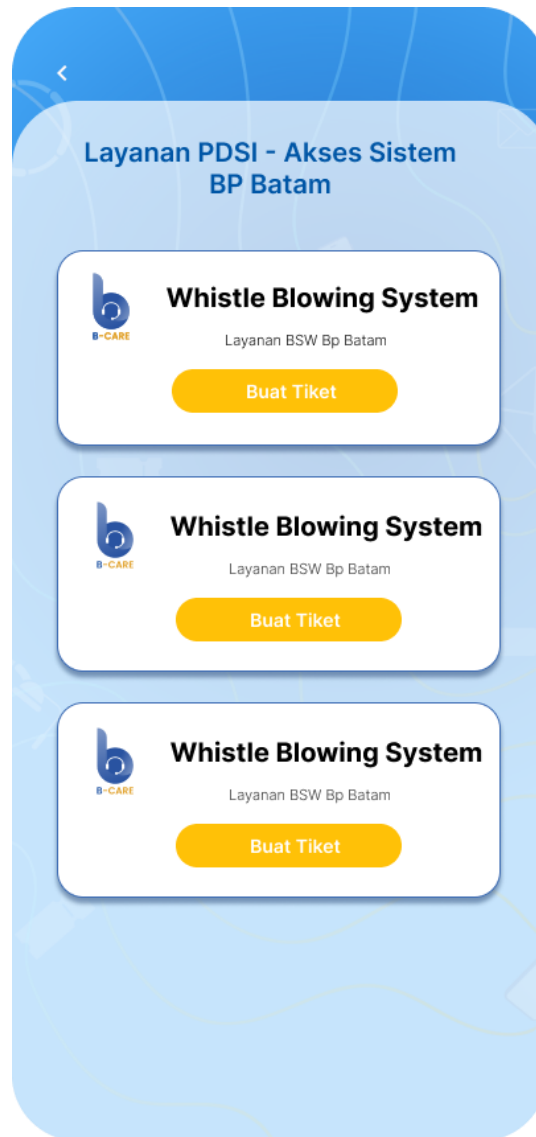
3.2.7.6 Kategori Layanan *Page*



Gambar 3. 23 Kategori Layanan *Page*

Gambar 3.23 di atas merupakan tampilan Kategori Layanan *Page* aplikasi *B-Care*. Halaman ini menampilkan berbagai kategori layanan yang tersedia di BP Batam. Pengguna dapat memilih kategori sesuai kebutuhan untuk melihat daftar layanan yang lebih spesifik sebelum membuat laporan atau pengaduan.

3.2.7.7 Layanan Page



Gambar 3. 24 Layanan Page

Gambar 3.24 di atas merupakan tampilan Layanan Page aplikasi *B-Care*. Halaman ini menampilkan daftar layanan berdasarkan kategori yang telah dipilih pengguna. Setiap layanan dilengkapi dengan nama dan deskripsi singkat agar pengguna dapat memahami fungsi layanan sebelum melanjutkan ke proses pengaduan atau permintaan.

3.2.7.8 Form Pengaduan *Page*

The image shows a mobile application interface for reporting a complaint. The form is divided into two main sections: 'Informasi Pemohon' (Applicant Information) and 'Informasi Aduan' (Complaint Information). The 'Informasi Pemohon' section includes a text input field for a name, followed by 'Email*' and 'Nomor Telp*' (Phone Number*) with corresponding input fields. The 'Informasi Aduan' section starts with a 'Layanan*' (Service*) label, a checkbox for 'Tampilkan Detail' (Show Details), and three dropdown menus for 'Pilih Unit' (Select Unit), 'Pilih Kategori' (Select Category), and 'Pilih Layanan' (Select Service). Below these are 'Tingkat Dampak*' (Impact Level*) with a dropdown, 'Judul Permasalahan*' (Problem Title*) with a text input, and 'Deskripsi*' (Description*) with a larger text input. A link 'Klik disini untuk lampiran dokumen pendukung' (Click here for supporting document attachment) is located below the description field. At the bottom of the form are two buttons: 'Reset' in red and 'Submit' in blue.

Gambar 3. 25 Form Pengaduan *Page*

Gambar 3.25 di atas merupakan tampilan Form Pengaduan *Page* aplikasi *B-Care*. Halaman ini digunakan oleh pengguna untuk membuat laporan atau pengaduan baru dengan mengisi data seperti judul laporan, deskripsi, kategori layanan, dan lampiran pendukung. Setelah formulir diisi lengkap, pengguna dapat mengirim pengaduan untuk diproses oleh pelaksana.

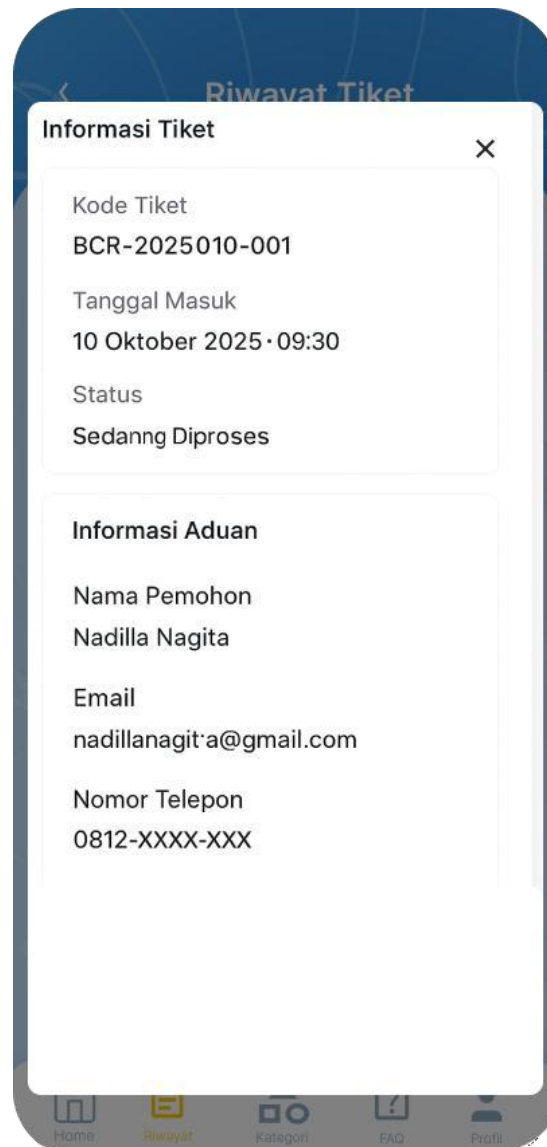
3.2.7.9 Riwayat Tiket Page



Gambar 3. 26 Riwayat Tiket Page

Gambar 3.26 di atas merupakan tampilan Riwayat Tiket Page aplikasi *B-Care*. Halaman ini menampilkan daftar seluruh laporan atau tiket yang telah dibuat oleh pengguna, lengkap dengan status prosesnya seperti “Menunggu”, “Diproses”, atau “Selesai”. Pengguna dapat memantau perkembangan setiap tiket dan membuka detailnya jika diperlukan.

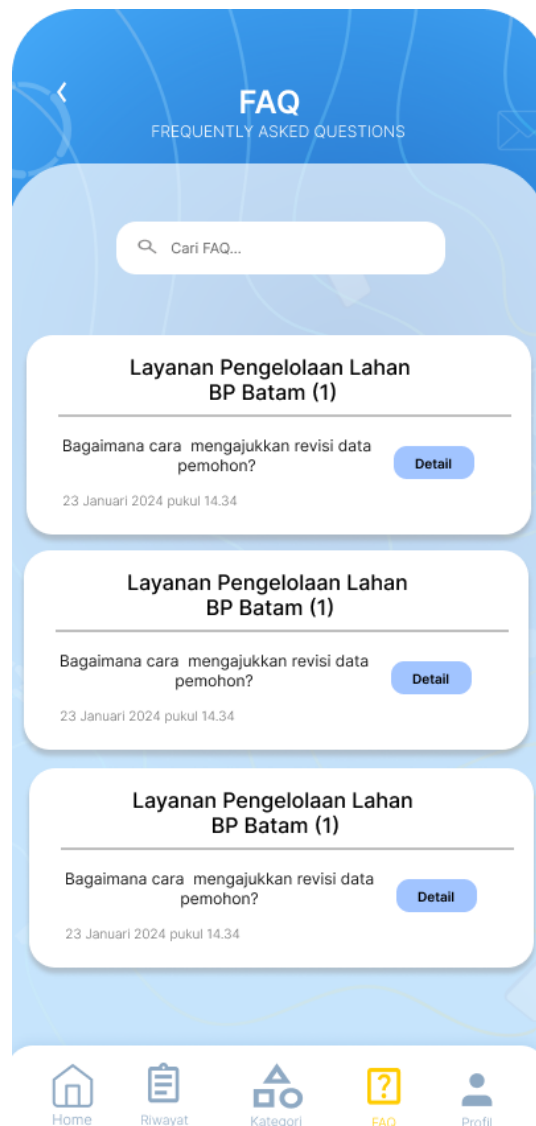
3.2.7.10 Detail tiket Riwayat *Page*



Gambar 3. 27 Detail Tiket riwayat *Page*

Gambar 3.27 di atas merupakan tampilan Detail Tiket *Page* aplikasi *B-Care*. Halaman ini menampilkan informasi lengkap mengenai laporan yang telah dibuat, seperti nomor tiket, tanggal pengaduan, kategori layanan, deskripsi, serta status terkini. Pengguna juga dapat melihat tanggapan dari pelaksana terkait laporan tersebut.

3.2.7.11 FAQ page



Gambar 3. 28 *FAQ Page*

Gambar 3.28 diatas merupakan gambar *FAQ Page*, yang menampilkan pertanyaan umum dan jawabannya agar pengguna dapat dengan cepat menemukan informasi tanpa perlu menghubungi pelaksana.

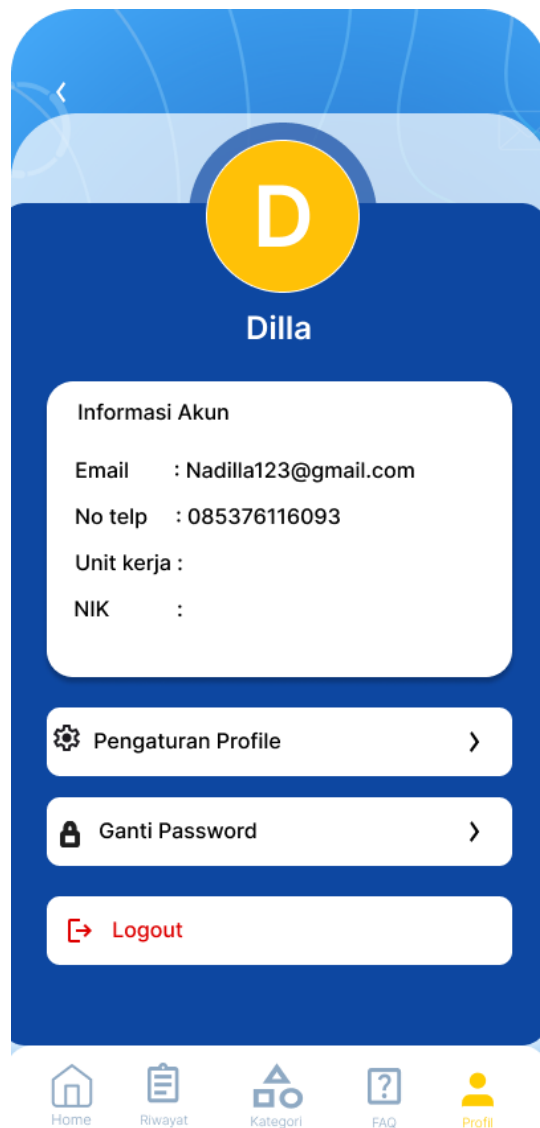
3.2.7.12 Detail *FAQ* Page



Gambar 3. 29 Detail *FAQ* Page

Gambar 3.29 di atas merupakan *FAQ* Detail Page, yang menampilkan jawaban lengkap dari pertanyaan yang dipilih pengguna. Halaman ini memudahkan pengguna memahami informasi secara lebih rinci dengan tampilan yang jelas dan terstruktur.

3.2.7.13 Profile Page



Gambar 3. 30 Profile Page

Gambar 3.30 di atas merupakan *Profile Page*, yang menampilkan informasi akun pengguna seperti nama, *email*, dan foto *profile*. Halaman ini memungkinkan pengguna untuk melihat dan mengelola data pribadinya dengan mudah.

3.2.7.14 Notifikasi *Page*



Gambar 3. 31 Notifikasi *Page*

Gambar 3.31 di atas merupakan Notifikasi *Page*, yang menampilkan pemberitahuan terbaru terkait aktivitas atau status laporan pengguna. Halaman ini memudahkan pengguna untuk tetap *update* secara *real-time* mengenai informasi penting dalam aplikasi.

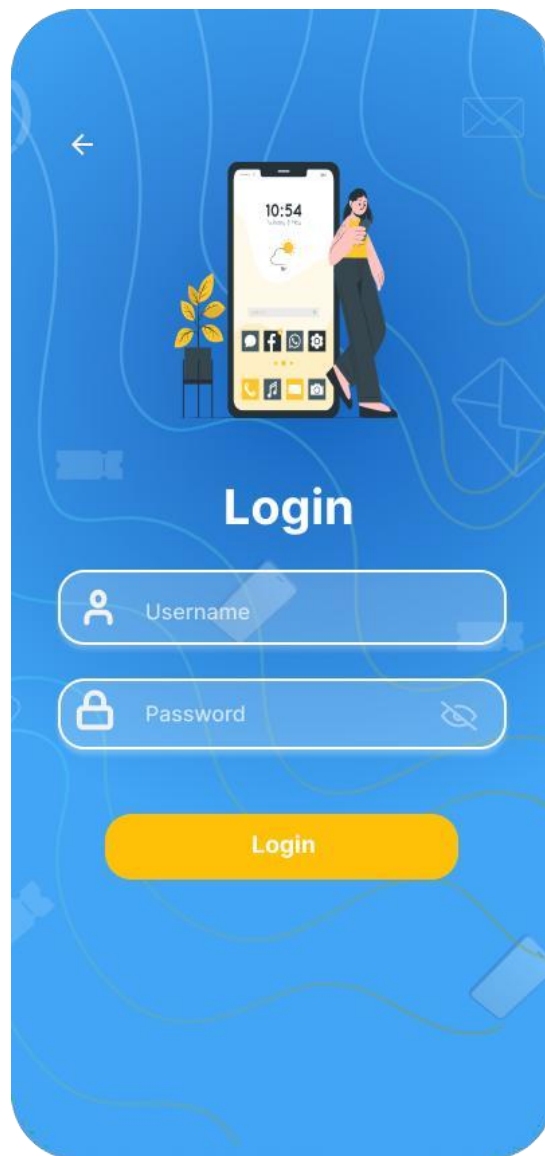
3.2.7.15 Guest Page



Gambar 3. 32 Guest Page

Gambar 3.32 di atas merupakan *Guest Page*, yang menampilkan informasi umum aplikasi seperti katalog layanan bagi pengguna yang belum melakukan *login*. Halaman ini dirancang agar pengunjung dapat mengakses informasi dan membuat tiket layanan penting tanpa harus mendaftar terlebih dahulu.

3.2.7.16 Pelaksana dan *Supervisor Login Page*



Gambar 3. 33 login pelaksana dan supervisor

Gambar 3.33 di atas merupakan Pelaksana dan *supervisor Login Page*, yang menampilkan form untuk masuk ke akun pelaksana atau supervisor. Halaman ini digunakan oleh pelaksana atau supervisor untuk mengakses fitur pengelolaan aplikasi secara aman dan terkontrol.

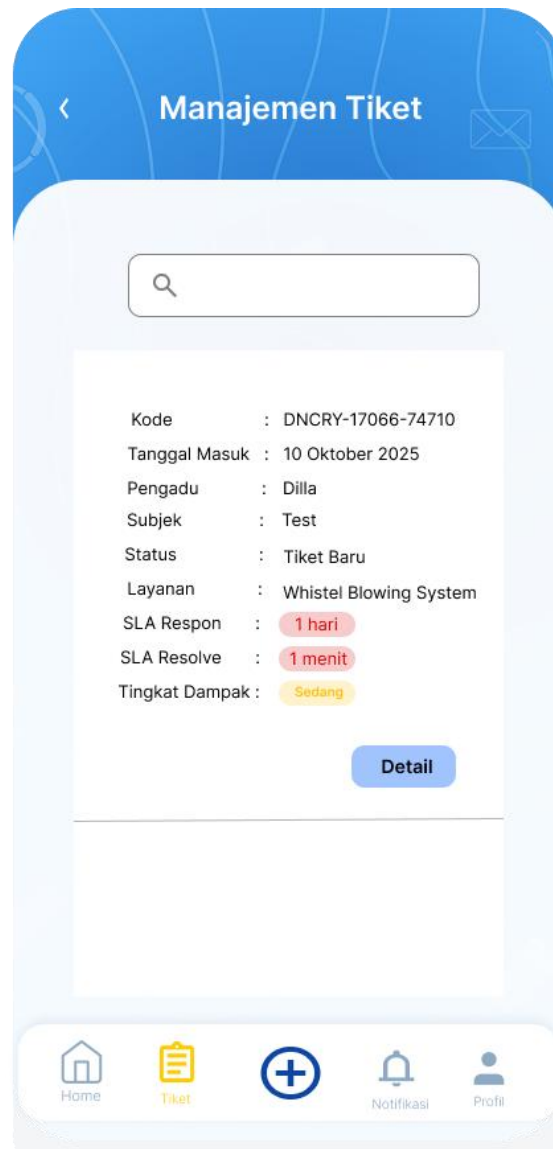
3.2.7.17 Dashboard Pelaksana Page



Gambar 3. 34 Dashboard Pelaksana Page

Gambar 3.34 di atas merupakan *Dashboard Page*, yang menampilkan ringkasan informasi penting seperti jumlah tiket, status laporan, dan aktivitas pengguna. Halaman ini memudahkan pelaksana dalam memantau dan mengelola seluruh aktivitas aplikasi.

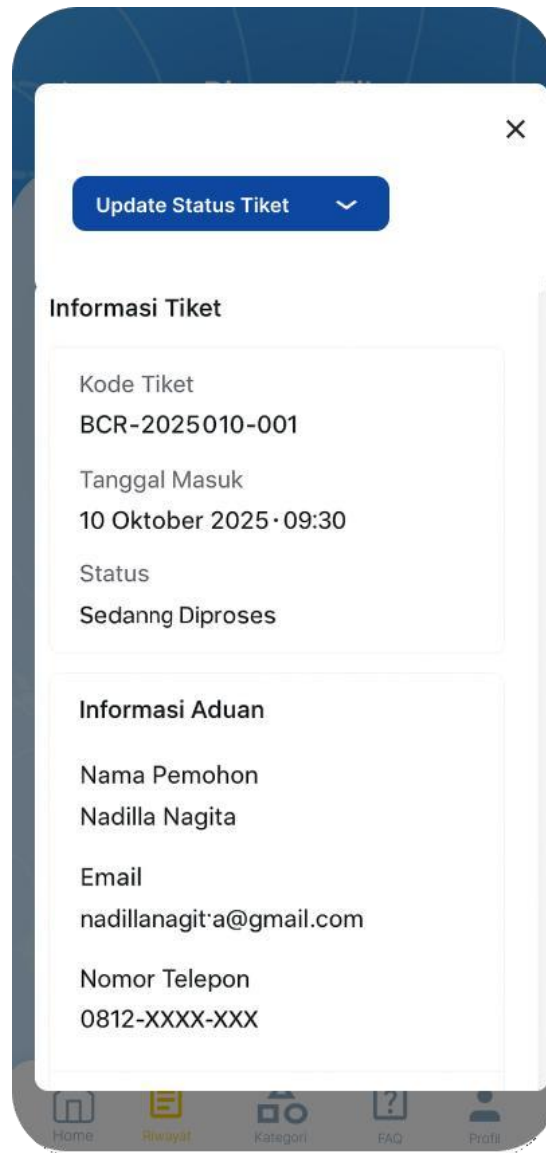
3.2.7.18 Management Tiket Page



Gambar 3. 35 Management tiket page

Gambar 3.35 di atas merupakan *Management Tiket page*, yang menampilkan daftar semua tiket atau laporan yang masuk beserta statusnya. Halaman ini digunakan pelaksana untuk memantau, memproses, dan menindaklanjuti tiket secara terstruktur dan efisien.

3.2.7.19 Detail Tiket *Page*



Gambar 3. 36 Detail Tiket *Page*

Gambar 3.36 di atas merupakan Detail Tiket *Page*, yang menampilkan informasi lengkap dari setiap tiket, termasuk deskripsi, lampiran, status. Halaman ini memudahkan pelaksana atau supervisor untuk memproses tiket layanan yang masuk.

3.2.7.20 Tambah Tiket *supervisor* dan pelaksana *Page*

×

Tambah Tiket

Tanggal Tiket Masuk*

Jenis Tiket*

Tiket Eksternal

Sumber Laporan*

Email

Nama Pengadu*

Dilla

☐ Anonim (Nama Anda tidak akan terpublikasi pada laporan)

Email*

nadilla123@gmail.com

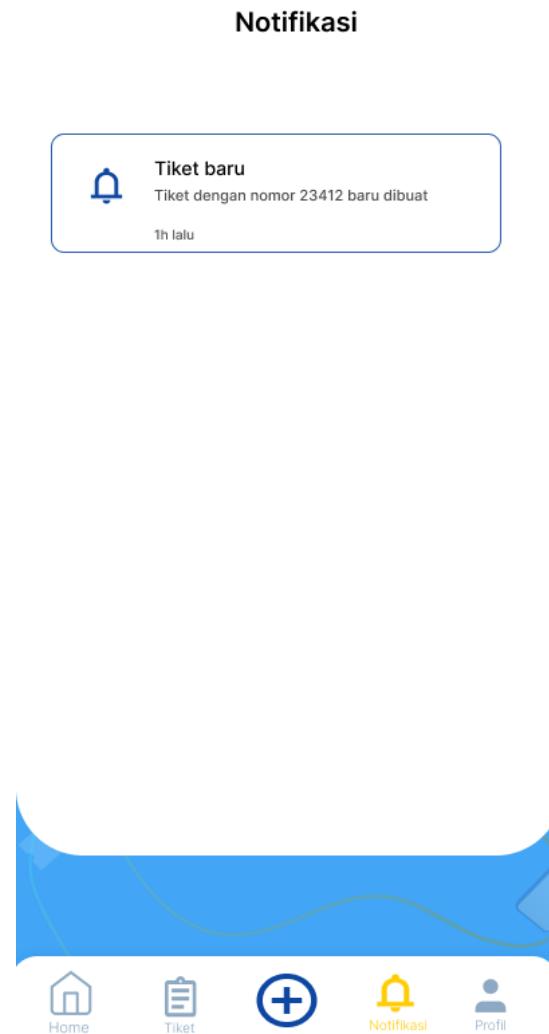
Submit

Home Tiket + Notifikasi Profil

Gambar 3. 37 Tambah Tiket *Supervisor* dan Pelaksana *Page*

Gambar 3.37 di atas merupakan Tambah Tiket *Supervisor* dan pelaksana *Page*, yang menampilkan form untuk membuat tiket atau laporan baru. Halaman ini memungkinkan *supervisor* untuk memasukkan detail informasi, kategori, dan lampiran sehingga tiket dapat langsung tercatat dan diproses dalam sistem.

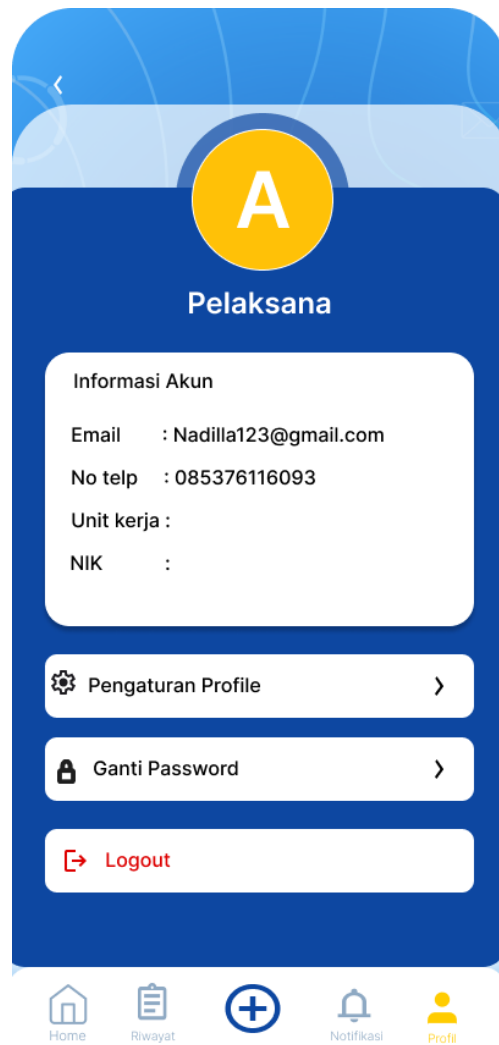
3.2.7.21 Notifikasi *Page*



Gambar 3. 38 Notifikasi *Page*

Gambar 3.38 di atas merupakan Notifikasi pelaksana *Page*, yang menampilkan pemberitahuan terkait aktivitas aplikasi, seperti tiket baru, *update* status, atau pesan penting. Halaman ini membantu pelaksana tetap terinformasi secara *real-time* untuk mengambil tindakan yang diperlukan.

3.2.7.22 Profile Page



Gambar 3. 39 Profile Page pelaksana

Gambar 3.0.39 di atas merupakan pelaksana *Profile Page*, yang menampilkan informasi akun pelaksana seperti nama, *email*, dan foto profil.

3.2.8 Perancangan Integrasi *REST API*

Berdasarkan hasil korespondensi dengan pihak instansi, akses terhadap *API* internal tidak dapat diberikan karena kebijakan sistem yang bersifat tertutup. Oleh karena itu, dalam penelitian ini dikembangkan *API backend* secara mandiri untuk mendukung proses integrasi data antara aplikasi *mobile* dan *database*.

Aplikasi *B-Care Mobile* dirancang menggunakan arsitektur *client-server*, di mana aplikasi *mobile* berperan sebagai *client* dan sistem backend dibangun menggunakan *framework Laravel* sebagai *REST API*.

Komunikasi antara aplikasi *mobile* dan *server* dilakukan melalui protokol *HTTP* dengan pertukaran data dalam format *JSON*. *REST API* berfungsi sebagai penghubung antara aplikasi *mobile* dan *database* untuk melakukan proses autentikasi, pengambilan data, serta pembaruan data. Namun, pada tahap implementasi dalam penelitian ini, *REST API* yang digunakan masih bersifat *dummy* dan belum terhubung dengan *server* produksi secara penuh.

Tabel 3. 21 *Endpoint API*

Endpoint	Method	Keterangan
/api/login	POST	Login pengguna
/api/logout	POST	Logout pengguna
/api/me	GET	Mengambil data user login
/api/tiket	POST	Membuat tiket
/api/tiket	GET	Menampilkan daftar tiket
/api/tiket/{kode_tiket}	GET	Melihat detail tiket
/api/tiket/{kode_tiket}	PUT	Update status tiket
/api/tiket/{kode_tiket}/assign	PUT	Assign pelaksana (Supervisor)
/api/tiket/{kode_tiket}/re-open	POST	Membuka kembali tiket

/api/notifikasi	GET	Mengambil notifikasi
/api/notifikasi/{id}	PUT	Menandai notifikasi dibaca
/api/tiket/ulasan/{kode_tiket}	POST	Mengirim ulasan
/api/ulasan	GET	Menampilkan daftar ulasan
/api/dashboard/stats	GET	Menampilkan statistik dashboard

Setiap permintaan yang dikirim oleh aplikasi *mobile* akan diproses oleh *controller* pada *Laravel*, kemudian sistem akan mengakses *database* untuk mengambil atau menyimpan data sesuai dengan kebutuhan. Respon dari *server* dikembalikan dalam bentuk *JSON* dan ditampilkan pada aplikasi *mobile*.

Proses autentikasi dilakukan melalui *endpoint* */api/login* menggunakan metode *POST*. Pada tahap ini, pengguna mengirimkan kredensial berupa email dan password ke *server* untuk dilakukan proses validasi. Sistem menggunakan *Laravel Sanctum* sebagai mekanisme *autentikasi* berbasis token. Setelah kredensial dinyatakan valid, sistem akan menghasilkan token autentikasi yang bersifat unik untuk setiap pengguna. Token tersebut digunakan untuk mengakses *endpoint* lain yang memerlukan otorisasi.

3.2.9 Rencana Pengujian dan Indikator Keberhasilan Sistem

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi yang dikembangkan berjalan sesuai dengan kebutuhan fungsional yang telah dirancang. Metode pengujian yang digunakan adalah *Black-box Testing* dan *User Acceptance Testing* (UAT).

Black-box Testing digunakan untuk menguji fungsi sistem tanpa melihat struktur kode program. Pengujian dilakukan dengan cara memberikan input pada setiap fitur seperti *login*, pengelolaan tiket, pembaruan status, dan integrasi *REST API*, kemudian mengamati apakah output yang dihasilkan sesuai dengan yang diharapkan.

Selain itu, *User Acceptance Testing (UAT)* dilakukan untuk mengetahui tingkat penerimaan pengguna terhadap sistem. *UAT* dilakukan dengan melibatkan beberapa responden untuk mencoba fitur utama sistem dan memberikan penilaian terhadap kemudahan penggunaan dan kesesuaian sistem dengan kebutuhan pengguna.

Keberhasilan aplikasi diukur berdasarkan dua indikator, yaitu:

1. Seluruh fitur sistem berjalan dengan baik sesuai kebutuhan (berdasarkan hasil *Black-box Testing*).
2. Tingkat kepuasan pengguna berada pada kategori baik berdasarkan hasil *UAT*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

Bagian ini menjelaskan hasil implementasi sistem *B-Care* yang telah dibangun berdasarkan rancangan yang telah ditetapkan. Implementasi ini mencakup pembangunan *backend* sebagai penyedia layanan data dan *frontend* sebagai antarmuka pengguna.

4.1.1 Spesifikasi Pengembangan

Implementasi sistem dilakukan menggunakan perangkat keras dan perangkat lunak tertentu guna memastikan proses pengembangan dan pengujian berjalan dengan optimal. Berikut adalah rincian spesifikasi yang digunakan:

1. Perangkat Keras (*Hardware*)

Tabel 4. 1 Perangkat Keras

No.	Komponen	Spesifikasi
1.	Perangkat Utama	Laptop HP
2.	Prosesor	AMD Ryzen 5 5500U
3.	Memori (RAM)	8 GB
4.	Penyimpanan	SSD 512 GB
5.	Perangkat Uji Fisik	<i>Smartphone</i> Samsung Galaxy A13

Tabel 4.1 menyajikan rincian spesifikasi perangkat keras yang digunakan peneliti, mulai dari perangkat utama berupa laptop dengan prosesor AMD Ryzen 5 dan RAM 8 GB, hingga perangkat uji fisik berupa *smartphone* untuk memastikan aplikasi berjalan dengan baik.

2. Perangkat Lunak (*Software*)

Tabel 4. 2 Perangkat Lunak

No.	Komponen	Spesifikasi
1.	Sistem Operasi	Windows 11 64-bit
2.	Bahasa Pemrograman	Dart (Frontend) dan PHP (Backend)
3.	Framework Frontend	Flutter SDK 3.35.7
4.	Framework Backend	Laravel Framework 12.38.1
5.	Database Management	MySQL (via XAMPP)
6.	Alat Pengembangan (IDE)	Visual Studio Code
7.	Pengujian API	Postman
8.	Emulator	Android Emulator pixel 5
9.	Layanan Push Notification	Firebase Cloud Messaging (FCM)

Kebutuhan perangkat lunak dalam pengembangan aplikasi *B-Care* disajikan pada Tabel 4.2. Peneliti menggunakan kombinasi *framework* Flutter untuk sisi *frontend* dan *Laravel* untuk sisi *backend*, serta *MySQL* sebagai sistem manajemen basis data.

4.1.2 Implementasi Basis Data

Implementasi basis data dilakukan menggunakan *MySQL*. Struktur tabel dirancang untuk mendukung manajemen data pengguna, kategori hingga riwayat status penanganan tiket. Berikut adalah tabel utama yang digunakan dalam sistem

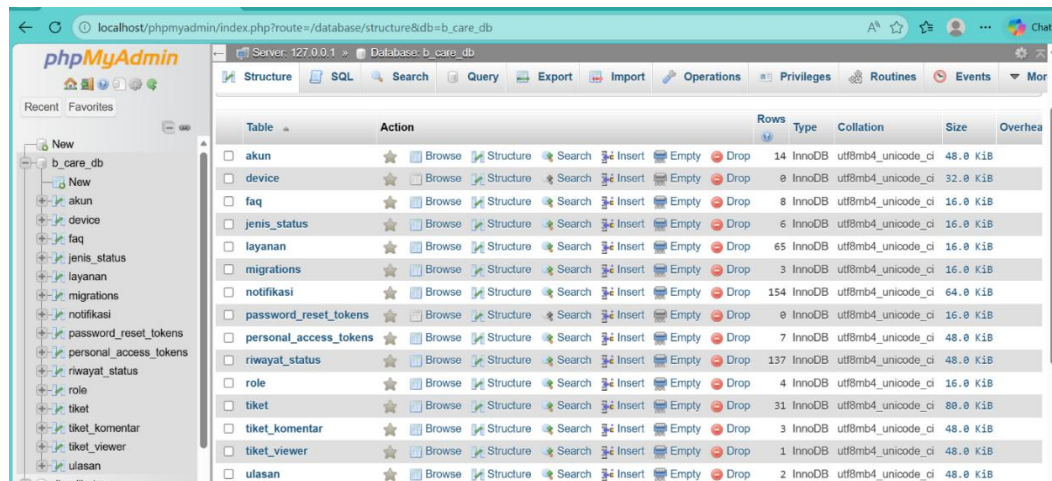
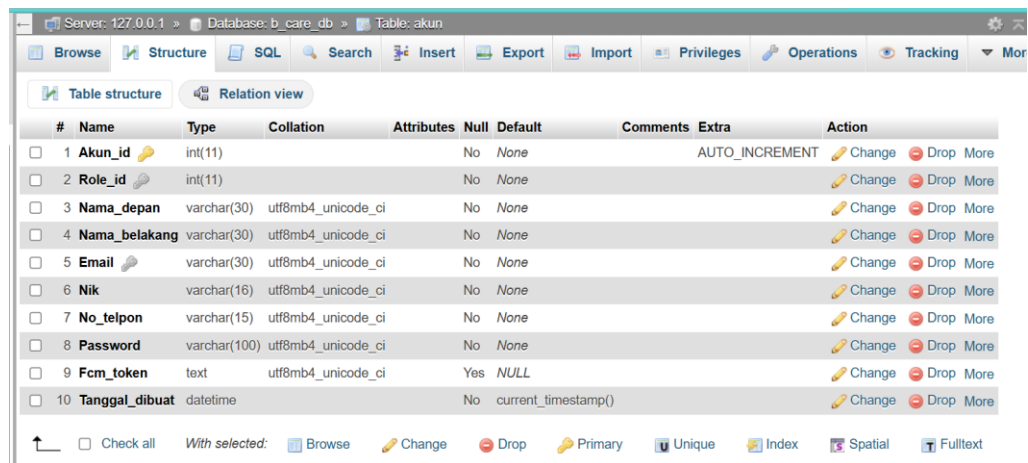


Table	Action	Rows	Type	Collation	Size	Overhead
akun	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	14	InnoDB	utf8mb4_unicode_ci	48.0 K	
device	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	0	InnoDB	utf8mb4_unicode_ci	32.0 K	
faq	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	0	InnoDB	utf8mb4_unicode_ci	16.0 K	
jenis_status	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	6	InnoDB	utf8mb4_unicode_ci	16.0 K	
layanan	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	65	InnoDB	utf8mb4_unicode_ci	16.0 K	
migrations	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	3	InnoDB	utf8mb4_unicode_ci	16.0 K	
notifikasi	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	154	InnoDB	utf8mb4_unicode_ci	64.0 K	
password_reset_tokens	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	0	InnoDB	utf8mb4_unicode_ci	16.0 K	
personal_access_tokens	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	7	InnoDB	utf8mb4_unicode_ci	48.0 K	
riwayat_status	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	137	InnoDB	utf8mb4_unicode_ci	48.0 K	
role	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	4	InnoDB	utf8mb4_unicode_ci	16.0 K	
tiket	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	31	InnoDB	utf8mb4_unicode_ci	80.0 K	
tiket_komentar	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	3	InnoDB	utf8mb4_unicode_ci	48.0 K	
tiket_viewer	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	1	InnoDB	utf8mb4_unicode_ci	48.0 K	
ulasan	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	2	InnoDB	utf8mb4_unicode_ci	48.0 K	

Gambar 4. 1 Tabel Database

Berdasarkan Gambar 4.1 merupakan tabel database terdapat beberapa tabel utama yang memiliki peran penting dalam fungsionalitas sistem *B-Care*, yaitu:

1.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Akun_id	int(11)			No	None		AUTO_INCREMENT	[Change] [Drop] More
2	Role_id	int(11)			No	None			[Change] [Drop] More
3	Nama_depan	varchar(30)	utf8mb4_unicode_ci		No	None			[Change] [Drop] More
4	Nama_belakang	varchar(30)	utf8mb4_unicode_ci		No	None			[Change] [Drop] More
5	Email	varchar(30)	utf8mb4_unicode_ci		No	None			[Change] [Drop] More
6	Nik	varchar(16)	utf8mb4_unicode_ci		No	None			[Change] [Drop] More
7	No_telpun	varchar(15)	utf8mb4_unicode_ci		No	None			[Change] [Drop] More
8	Password	varchar(100)	utf8mb4_unicode_ci		No	None			[Change] [Drop] More
9	Fcm_token	text	utf8mb4_unicode_ci		Yes	NULL			[Change] [Drop] More
10	Tanggal_dibuat	datetime			No	current_timestamp()			[Change] [Drop] More

Gambar 4. 2 Tabel akun

Gambar 4.2 Merupakan tabel akun yang ada di *database b-care*, tabel ini Menyimpan data informasi pengguna yang mencakup data diri pengguna berdasarkan *role* yang sudah ditentukan yang terbagi menjadi tiga hak akses, yaitu *User*, *Supervisor*, dan Pelaksana.

2.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Tiket_id	int(11)			No	None		AUTO_INCREMENT	Change Drop
2	Akun_id	int(11)			Yes	NULL			Change Drop
3	Akun_id_pelaksana	int(11)			Yes	NULL			Change Drop
4	Layanan_id	int(11)			No	None			Change Drop
5	Nama_pelapor	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop
6	Email_pelapor	varchar(30)	utf8mb4_unicode_ci		Yes	NULL			Change Drop
7	Telepon_pelapor	varchar(20)	utf8mb4_unicode_ci		Yes	NULL			Change Drop
8	Kode_tiket	varchar(30)	utf8mb4_unicode_ci		No	None			Change Drop
9	Judul	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop
10	Deskripsi	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop
11	Dampak	varchar(30)	utf8mb4_unicode_ci		No	None			Change Drop
12	Lampiran	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop
13	Tanggal_registrasi	datetime			No	current_timestamp()			Change Drop
14	Tanggal_selesai	timestamp			Yes	NULL			Change Drop
15	Status	enum('Registrasi', 'Proses', 'Kembalikan', 'Tolak')	utf8mb4_unicode_ci		No	Registrasi			Change Drop
16	Catatan	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop

Gambar 4. 3 Tabel tiket

Gambar 4.3 Merupakan tabel tiket yang berfungsi sebagai penyimpanan utama data pengaduan yang masuk, baik dari pengguna yang telah *login* maupun *guest*. Tabel ini juga mencatat perbedaan ID akun untuk membedakan laporan ini dibuat oleh *user* atau *guest*. Di tabel ini juga menyimpan semua tiket-tiket yang dikelola mulai dari tiket yang berstatus register hingga tiket yang sudah selesai semuanya ada didalam tabel ini berserta catatan, tanggal register dan tanggal selesai tiket. Pada tabel ini juga terdapat beberapa relasi dari tabel-tabel lainnya seperti tabel Akun, tabel Riwayat status dan tabel layanan tabel ini memiliki relasi satu dengan lainnya.

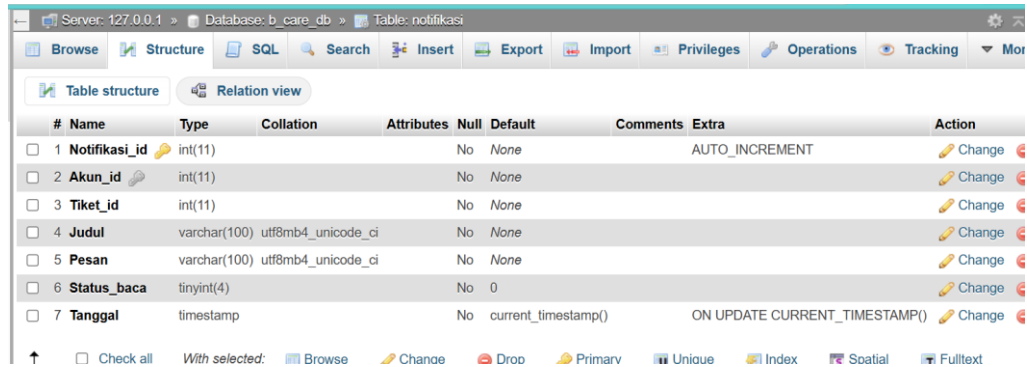
3.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Riwayat_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Tiket_id	int(11)			No	None			Change Drop More
3	Akun_id	int(11)			Yes	NULL			Change Drop More
4	Status_awal	enum('Registrasi', 'Proses', 'Kembalikan', 'Tolak')	utf8mb4_unicode_ci		No	None			Change Drop More
5	Status_baru	enum('Registrasi', 'Proses', 'Kembalikan', 'Tolak')	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
6	Catatan	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
7	Lampiran	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
8	Waktu	datetime			No	current_timestamp()			Change Drop More

Gambar 4. 4 Tabel Riwayat status

Gambar 4.4 Diatas merupakan Tabel Riwayat tabel digunakan untuk mencatat setiap tahapan perubahan status laporan (*log*) guna keperluan *tracking* status tiket yang akurat, Dimana pada tabel ini setiap status tiket yang berubah maka akan tercatat ID akun yang merubah status dan catatan yang dibuat pada status tersebut.

4.

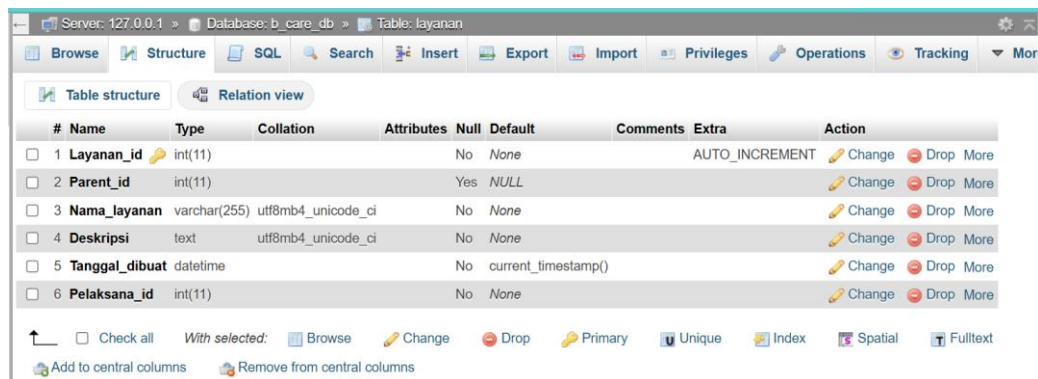


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Notifikasi_id	int(11)			No	None		AUTO_INCREMENT	Change
2	Akun_id	int(11)			No	None			Change
3	Tiket_id	int(11)			No	None			Change
4	Judul	varchar(100)	utf8mb4_unicode_ci		No	None			Change
5	Pesan	varchar(100)	utf8mb4_unicode_ci		No	None			Change
6	Status_baca	tinyint(4)			No	0			Change
7	Tanggal	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change

Gambar 4. 5 Tabel Notifikasi

Gambar 4.5 merupakan Tabel notifikasi, Digunakan untuk mengelola data pemberitahuan yang terintegrasi dengan *Firestore Cloud Messaging (FCM)*, notifikasi yang terkirim akan tercatat di tabel ini dan dipush oleh *FCM* ke *smartphone* pengguna dan pengelola.

5.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Layanan_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Parent_id	int(11)			Yes	NULL			Change Drop More
3	Nama_layanan	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	Deskripsi	text	utf8mb4_unicode_ci		No	None			Change Drop More
5	Tanggal_dibuat	datetime			No	current_timestamp()			Change Drop More
6	Pelaksana_id	int(11)			No	None			Change Drop More

Gambar 4. 6 Tabel Layanan

Gambar 4.6 merupakan tabel layanan, Menyimpan kategori layanan atau jenis kendala yang dapat dipilih oleh pelapor, tabel ini fungsi sebagai tempat penyimpanan data layanan yang tersedia di dalam sistem b-care

4.1.3 Implementasi API (Backend Laravel)

Implementasi *backend* dilakukan menggunakan *framework Laravel* untuk menyediakan layanan *API* sebagai penghubung antara basis data dengan aplikasi *mobile*. Karena adanya keterbatasan akses pada *API* internal instansi (sebagaimana bukti korespondensi yang dapat diakses melalui tautan pada halaman Lampiran), maka dikembangkan *API* mandiri yang mencakup manajemen autentikasi, pengolahan tiket, hingga integrasi notifikasi. Data yang digunakan pada *API* mandiri ini merupakan data simulasi yang telah disesuaikan dengan skema kebutuhan fungsional aplikasi B-Care

Berikut adalah beberapa hasil pengujian *endpoint* utama menggunakan aplikasi *Postman*:

4.1.3.1 Pengujian *API Register*

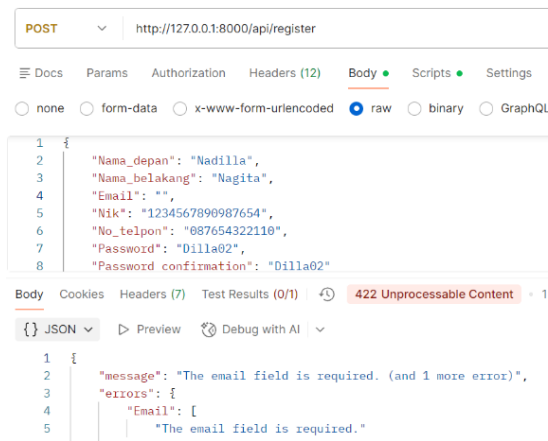
Pengujian pada *endpoint* register dilakukan untuk memastikan sistem dapat menangani berbagai skenario input dari pengguna. Tabel 4.3 menunjukkan skenario pengujian yang dilakukan:

Tabel 4. 3 Hasil Pengujian *Endpoint API Register*

No	Endpoint	Method	Data Uji	Hasil yang Diharapkan	Hasil
1	/api/register	POST	Data valid	Status 201, akun berhasil dibuat	Berhasil
2	/api/register	POST	Email kosong	The email field is required	Berhasil

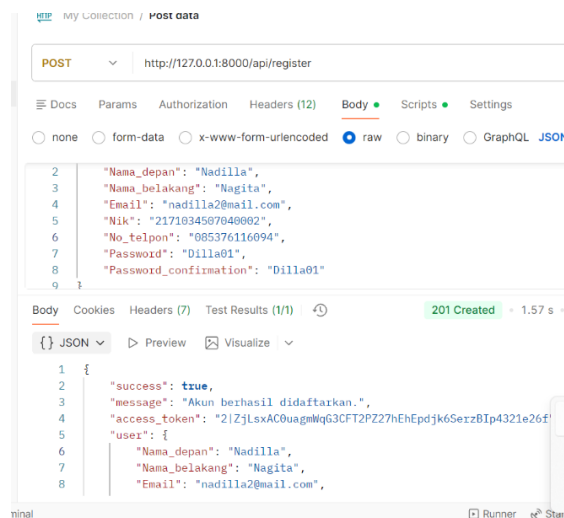
Hasil Pengujian *API Register*: Untuk membuktikan hasil pada Tabel 4.3, berikut adalah beberapa hasil pengujian menggunakan *Postman*:

1. Gambar 4.7 Skenario Email Kosong, Sistem berhasil menolak permintaan dengan respons 422 Unprocessable content dan memberikan pesan validasi "*The email field is required*".



Gambar 4. 7 Respon *Error API* Register Saat *Email* Kosong

2. Gambar 4. 8 respon Skenario Data Valid, Sistem berhasil membuat akun baru dengan status 201 Created.



Gambar 4. 8 Respon Berhasil *API Register*

3. Gambar 4.9 Verifikasi Database, Setelah proses registrasi berhasil, data tersimpan dengan benar pada tabel akun.

☐	Edit	Copy	Delete	5	4	Nadilla	Nagita	nadilla1@mail.com	1234567890987654	087654322110	\$2y\$12\$
☐	Edit	Copy	Delete	6	4	Nadilla	Nagita	nadilla2@mail.com	2171034507040002	085376116094	\$2y\$12\$

Gambar 4. 9 *Database Create* akun baru

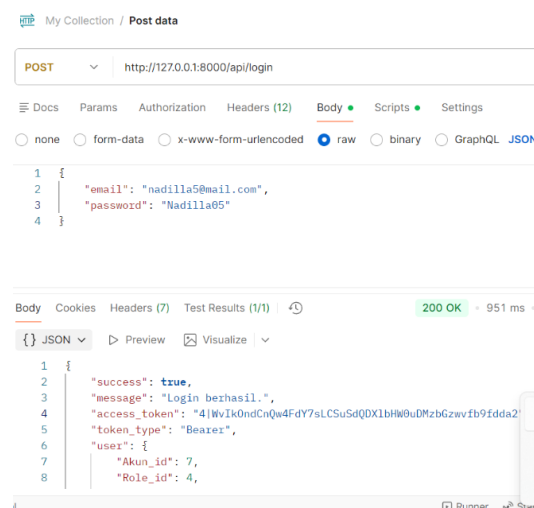
4.1.3.2 Pengujian *API Login*

Pengujian pada *endpoint login* dilakukan untuk memastikan proses autentikasi pengguna berjalan dengan aman dan sesuai dengan kebutuhan sistem. Pengujian ini mencakup skenario input data yang valid maupun penanganan kesalahan saat kredensial tidak sesuai. Tabel 4.4 menunjukkan skenario pengujian yang dilakukan:

Tabel 4. 4 Hasil Pengujian *Endpoint API Login*

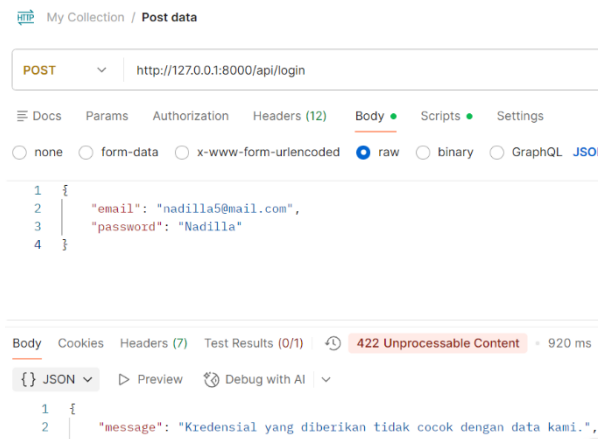
No	Endpoint	Method	Data Uji	Hasil yang Diharapkan	Hasil
1	/api/login	POST	Data valid	Status 201, berhasil login	Berhasil
2	/api/login	POST	Password salah	Kredensial yang diberikan tidak cocok dengan data kami.	Berhasil
3	/api/login	POST	Email salah	Kredensial tidak cocok dengan data kami.	Berhasil

1. Gambar 4.10 merupakan Skenario Data Valid (Login Berhasil), Sistem berhasil mengenali kredensial pengguna dan memberikan respons status 200 OK beserta sebuah *Access Token*. Token ini nantinya digunakan oleh aplikasi Flutter sebagai kunci autentikasi untuk mengakses fitur-fitur lainnya.



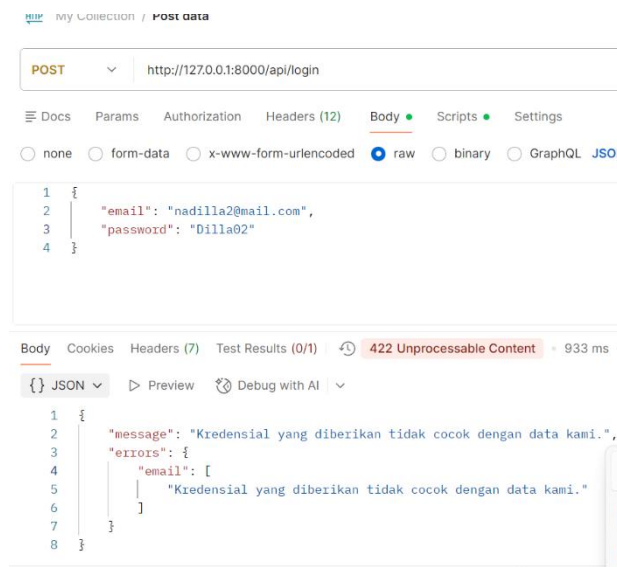
Gambar 4. 10 Respon Berhasil *API Login*

2. Gambar 4.11 merupakan Skenario *Password* Salah, Sistem memberikan proteksi dengan menolak permintaan jika *password* tidak sesuai. Respons yang diberikan adalah status 422 *Unprocessable Content* dengan pesan kesalahan "*Kredensial yang diberikan tidak cocok dengan data kami.*"



Gambar 4. 11 Respon *API* jika *Password* Salah

2. Gambar 4.12 merupakan Skenario Email Salah/Tidak Terdaftar, Sistem juga melakukan validasi terhadap email pengguna. Jika email tidak ditemukan di database, sistem memberikan respons yang serupa untuk menjaga keamanan informasi akun.



Gambar 4. 12 Respon *API Login* Jika *Email* Salah

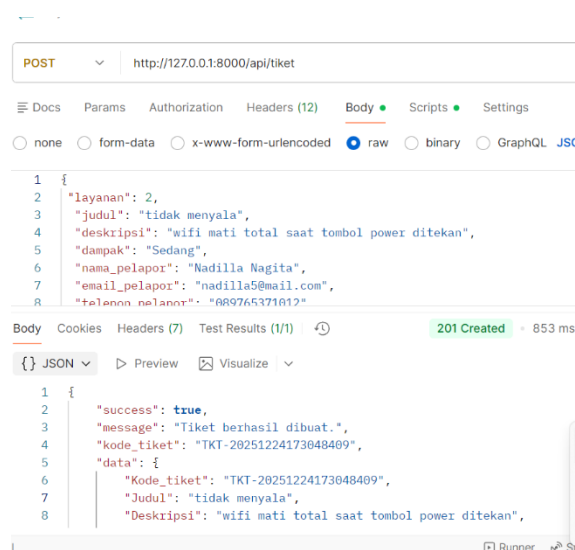
4.1.3.3 Pengujian API Manajemen Tiket

Pengujian pada *endpoint* tiket dilakukan untuk memastikan fungsionalitas utama aplikasi dalam menampung pengaduan berjalan dengan baik. Sistem dirancang untuk dapat menerima laporan baik dari pengguna yang terautentikasi (*User*) maupun pengguna tanpa akun (*Guest*). Tabel 4.5 menunjukkan skenario pengujian tersebut:

Tabel 4. 5 Hasil Pengujian *Endpoint API* Manajemen Tiket

No	Skenario Pengujian	Kondisi Pengguna	Data Input	Hasil yang Diharapkan	Hasil Pengujian
1	Membuat tiket sebagai guest	Tidak login	Semua field wajib diisi	Tiket berhasil dibuat dan disimpan sebagai guest	Berhasil
2	Membuat tiket sebagai user login	Login	Semua field wajib diisi	Tiket berhasil dibuat dan tercatat dengan akun pengguna	Berhasil
3	Membuat tiket dengan data tidak lengkap	Guest / Login	Salah satu field kosong	Sistem menolak dan menampilkan pesan kesalahan	Berhasil
4	Melihat detail tiket berdasarkan kode tiket	Guest / Login	Kode tiket valid	Detail tiket berhasil ditampilkan	Berhasil

1. Skenario Pembuatan Tiket (Login): Sistem berhasil memproses pembuatan tiket saat pengguna mengirimkan data lengkap. Respons status *201 Created* menunjukkan tiket berhasil disimpan beserta kode unik tiket untuk pelacakan.



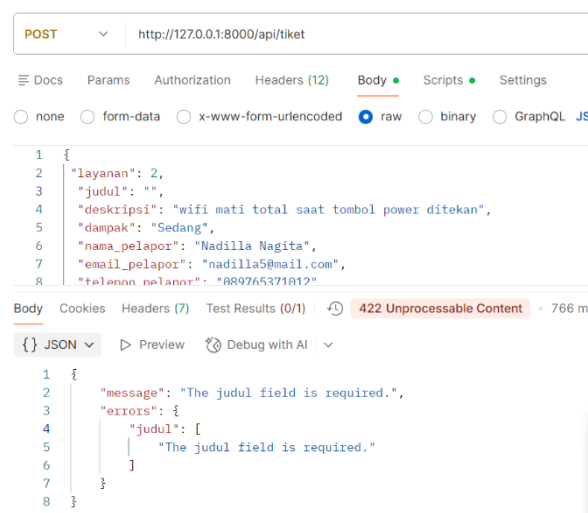
Gambar 4. 13 Respon *API* Tiket User

2. Gambar 4.14 merupakan Verifikasi Perbedaan *Guest* dan *User* pada Database: terlihat hasil pada tabel basis data. Sistem secara otomatis memberikan nilai NULL pada kolom akun_id untuk tiket yang dibuat oleh *Guest*, sedangkan untuk *User* yang login, kolom akun_id terisi sesuai dengan ID pengguna. Ini membuktikan relasi antar tabel berjalan dengan benar.

Tiket_id	Akun_id	Akun_id_pelaksana	Layanan_id	Nama_pelapor	Email_pelapor	Telepon_pelapor	Kode_tiket	Judul	Deskripsi
ete	3	NULL	NULL	1 Nadilla Nagita	nadilla5@mail.com	089765371012	TKT-20251224171128631	Laptop tidak bisa menyala	Laptop mati total saat tombol power ditekan
ete	4	NULL	NULL	3 Nadilla Nagita	nadilla5@mail.com	089765371012	TKT-20251224171735973	wifi tidak menyala	wifi mati total saat tombol power ditekan
ete	5	7	NULL	2 Nadilla Nagita	nadilla5@mail.com	089765371012	TKT-	tidak yala	wifi mati total saat tombol power ditekan

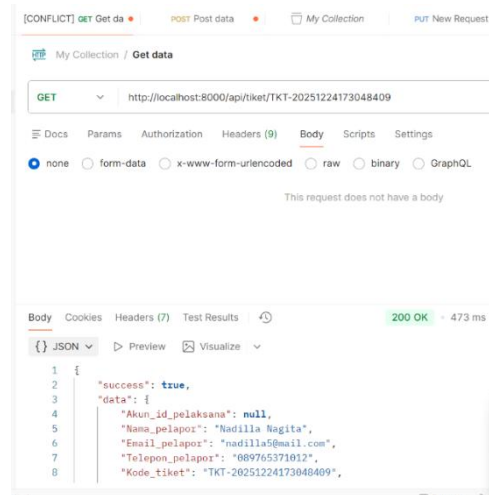
Gambar 4. 14 Verifikasi Perbedaan *Guest* dan *User* pada Database

3. Gambar 4.15 merupakan Skenario Validasi (*Field Kosong*), Sistem memiliki validasi yang kuat. Saat pengguna lupa mengisi judul laporan, sistem menolak permintaan dan memberikan pesan error 422 *Unprocessable Content* berupa "*The judul field is required*".



Gambar 4. 15 Respon *API* Jika Judul kosong

4. Gambar 4.16 merupakan Skenario Pencarian Tiket (Get Tiket): Fungsi pencarian berdasarkan nomor tiket berhasil dijalankan. Sistem menampilkan detail informasi laporan secara lengkap hanya dengan memasukkan nomor tiket yang valid.



Gambar 4. 16 Respon API Search Tiket

4.1.3.4 Pengujian API Kelola Tiket

Pengujian ini difokuskan pada alur kerja (*workflow*) penanganan laporan, mulai dari proses disposisi oleh *Supervisor* hingga penyelesaian laporan oleh Pelaksana. Pengujian ini memastikan bahwa setiap perubahan status tercatat dengan benar dalam log sistem. Tabel 4.6 menunjukkan skenario pengujian tersebut:

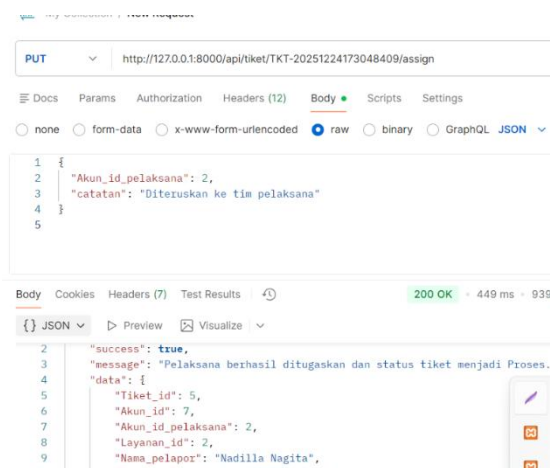
Tabel 4. 6 Hasil Pengujian *Endpoint API* Kelola Tiket

No	Skenario Pengujian	Aktor	Langkah Pengujian	Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status
1	Supervisor melihat daftar tiket	Supervisor	Login → menu daftar tiket	Akun supervisor	Sistem menampilkan seluruh tiket dengan status Registrasi	Sesuai	Berhasil
2	Supervisor melakukan disposisi tiket	Supervisor	Klik tombol Tangani → isi catatan → pilih	Catatan & Akun Pelaksana	Status tiket berubah menjadi Dikonfirmasi , pelaksana	Sesuai	Berhasil

			pelaksana		terisi		
3	Pelaksana melihat tiket yang ditugaskan	Pelaksana	Login → menu daftar tiket	Akun pelaksana	Sistem menampilkan tiket yang sudah ditugaskan	Sesuai	Berhasil
4	Pelaksana memproses tiket	Pelaksana	Klik tombol Proses	Tiket Dikonfirmasi	Status tiket berubah menjadi Proses	Sesuai	Berhasil
5	Pelaksana menyelesaikan tiket	Pelaksana	Klik tombol Selesai → isi catatan	Catatan penyelesaian	Status tiket berubah menjadi Selesai	Sesuai	Berhasil

Berikut adalah bukti teknis bahwa alur disposisi dan penanganan tiket telah berjalan sesuai rancangan:

1. Gambar 4.17 merupakan Skenario Disposisi oleh *Supervisor*, *Supervisor* berhasil melakukan tugas "*Assign*" atau penunjukan Pelaksana melalui *endpoint* /assign. Respons 200 OK menunjukkan bahwa tiket tersebut kini telah memiliki penanggung jawab (Pelaksana).



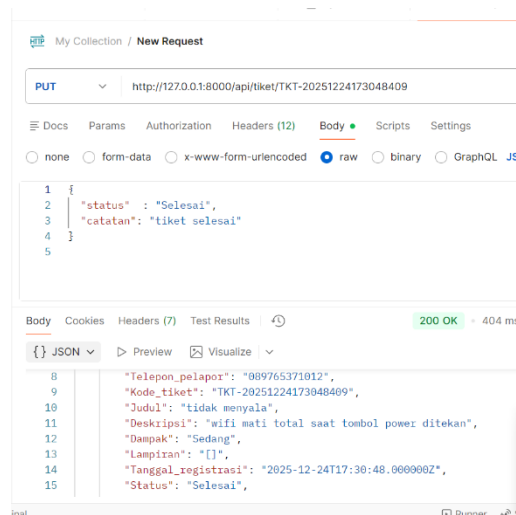
Gambar 4. 17 Respon *API Assign* Tiket

2. Verifikasi Riwayat Status (*Log*): Seperti yang terlihat pada Gambar 4.18 , setiap aksi yang dilakukan oleh aktor (*Supervisor*/*Pelaksana*) terekam secara otomatis pada tabel riwayat_status. Hal ini membuktikan bahwa sistem mampu melakukan *tracking* laporan secara mendetail mulai dari status '*Registrasi*' hingga '*Selesai*'.

Extra options									
		Riwayat_id	Tiket_id	Akun_id	Status_awal	Status_baru	Catatan	Lampiran	Waktu
☐	Edit Copy Delete		1	3	NULL	Registrasi	Registrasi	Tiket dibuat oleh Guest	NULL 2025-12-25 00:11:28
☐	Edit Copy Delete		2	4	NULL	Registrasi	Registrasi	Tiket dibuat oleh Guest	NULL 2025-12-25 00:17:35
☐	Edit Copy Delete		3	5	7	Registrasi	Registrasi	Tiket dibuat oleh Pelapor	NULL 2025-12-25 00:30:48
☐	Edit Copy Delete		4	5	1	Registrasi	Proses	Diteruskan ke tim pelaksana	NULL 2025-12-24 18:22:01

Gambar 4. 18 Database Riwayat Status

- Gambar 4. 19 merupakan Skenario Penyelesaian oleh Pelaksana: Pelaksana dapat memperbarui status tiket menjadi 'Selesai' setelah kendala di lapangan teratasi. Sistem menerima catatan penyelesaian dan memperbarui status akhir tiket di *database* utama.



Gambar 4. 19 Respon API Tiket Selesai Diproses

4.1.4 Implementasi Antarmuka

Bagian ini menyajikan hasil akhir tampilan antarmuka (*User Interface*) aplikasi *B-Care* setelah melalui proses pengembangan dengan *Flutter*. Penjelasan pada bagian ini disusun berdasarkan alur akses masing-masing pengguna (*role*) untuk memberikan gambaran yang jelas mengenai fungsi dan navigasi yang tersedia di dalam sistem

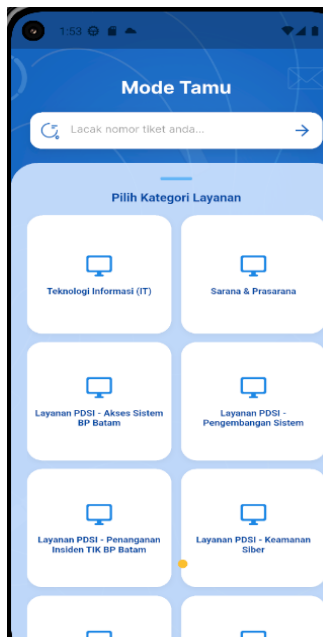
4.1.4.1 Antarmuka Umum dan *Role Guest*

Halaman ini merupakan tampilan awal yang dapat diakses oleh masyarakat umum tanpa harus melakukan proses *login*.



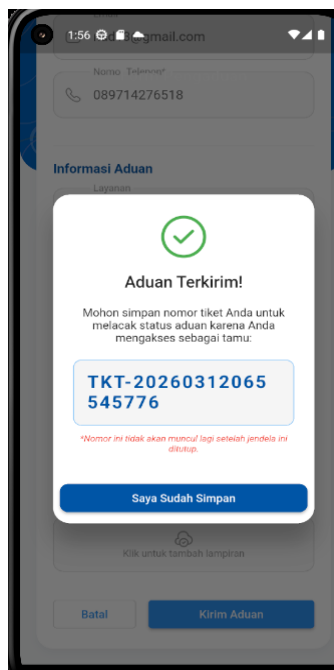
Gambar 4. 20 Implementasi *Welcome Page*

Berdasarkan Gambar 4.20, merupakan hasil implementasi halaman *welcome page* merupakan tampilan utama saat aplikasi pertama kali dijalankan. Pada halaman ini, pengguna diberikan tiga opsi navigasi utama, yaitu melakukan *Login* bagi pengguna terdaftar, *Register* untuk pembuatan akun baru, atau melakukan geser (*swipe*) pada bagian bawah untuk masuk sebagai *Guest* (Tamu) guna mengakses layanan pengaduan tanpa akun



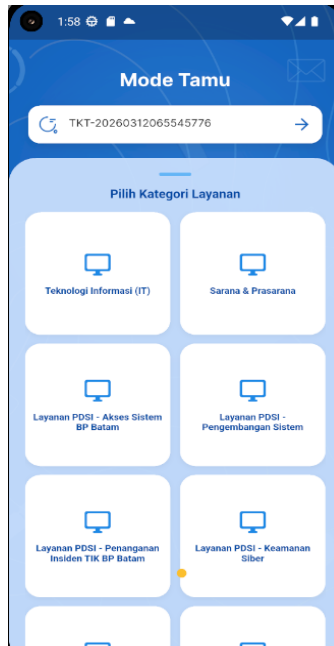
Gambar 4. 21 Implementasi Layanan *Guest*

Setelah pengguna memilih masuk sebagai tamu, sistem akan mengarahkan ke halaman layanan *Guest* seperti yang terlihat pada Gambar 4.21 diatas lalu *guest* memilih layanan yang ingin dibuat laporan.



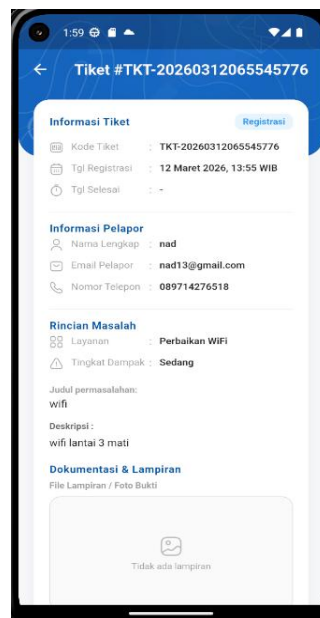
Gambar 4. 22 Implementasi *Pop Up* Kode Tiket

Setelah *guest* selesai mengisi *form* pengaduan dan mengirimkan tiket pengaduannya setelah itu *guest* akan mendapatkan *pop up* seperti gambar 4. 22 Diatas yang menyuruh *guest* menyimpan nomor tiket pengaduannya.



Gambar 4. 23 Implementasi *Tracking* Kode Tiket *Guest*

Gambar 4.23 Diatas merupakan hasil implementasi cara *guest tracking* status tiketnya dimana *guest* mencari tiektnya menggunakan nomor tiket yang sudah diberikan saat selesai membuat aduan tiket.

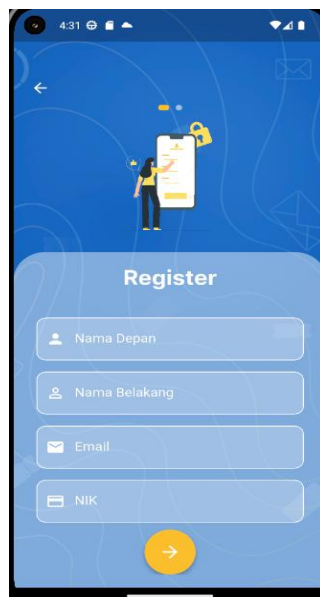


Gambar 4. 24 Hasil Implementasi Detail Tiket

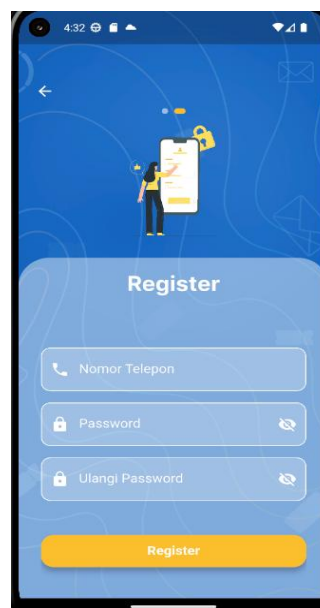
Pada gambar 4.24 Diatas merupakan hasil implementasi detail tiket yang berhasil ditampilkan sistem setelah guest melakukan tracking tiket tadi menggunakan nomor tiket.

4.1.4.2 Antarmuka Role *User (Login)*

Halaman ini merupakan antarmuka khusus bagi pengguna yang telah melakukan otentikasi melalui proses *login*. Berbeda dengan akses tamu, pengguna pada *role* ini memiliki akses ke fitur manajemen tiket pribadi, notifikasi, dan riwayat pengaduan yang lebih mendetail.

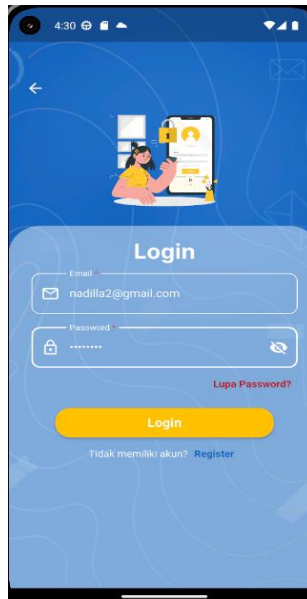
A mobile application screenshot showing a registration form titled "Register". The form has a blue background with a subtle pattern. At the top, there is an illustration of a person holding a document. Below the title, there are four input fields: "Nama Depan", "Nama Belakang", "Email", and "NIK". Each field has a corresponding icon (person, person, envelope, and ID card respectively). At the bottom of the form, there is a yellow circular button with a right-pointing arrow.

Gambar 4. 25 Implementasi Registrasi Akun 1

A mobile application screenshot showing a registration form titled "Register". The form has a blue background with a subtle pattern. At the top, there is an illustration of a person holding a document. Below the title, there are three input fields: "Nomor Telepon", "Password", and "Ulangi Password". Each field has a corresponding icon (phone, lock, and lock respectively). At the bottom of the form, there is a yellow rectangular button labeled "Register".

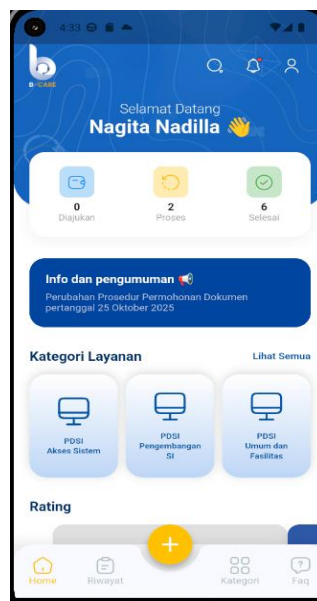
Gambar 4. 26 Implementasi Registrasi Akun 2

Gambar 4.25 Dan 4.26 Merupakan hasil implementasi halaman register, halaman ini tempat untuk *user* mendaftarkan akun yang pertama kali dibuat.



Gambar 4. 27 Implementasi Halaman *Login*

Gambar 4. 27 Merupakan hasil implementasi halaman *login user*, halaman ini digunakan untuk autentikasi akun *user* yang sudah registrasi akun.

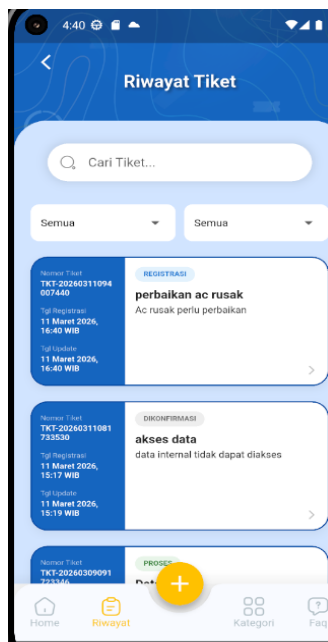


Gambar 4. 28 Implementasi *Dahboard User*

Gambar 4. 28 Merupakan hasil implementasi *dashboard user*, halaman ini digunakan sebagai halaman pertama *user* ketika *login* akunya berhasil, *Dashboard* dirancang untuk memberikan kemudahan akses bagi pengguna dalam memantau status pengaduan dan memilih layanan yang tersedia.

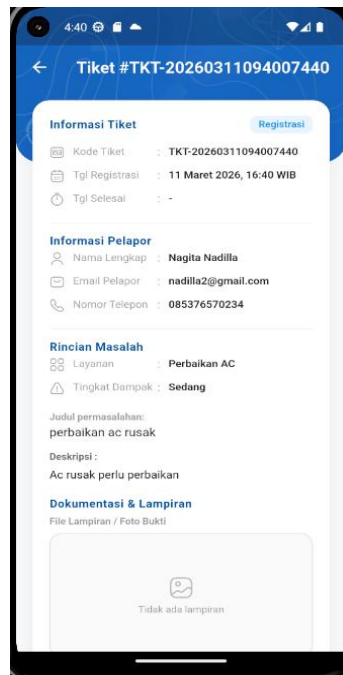
Gambar 4. 29 Implementasi *Form* Tiket

Gambar 4. 29 Merupakan hasil implementasi *form* pengaduan tiket, di halaman ini user dapat memilih layanan aduan dan menambahkan judul serta deskripsi masalah yang diadukan.



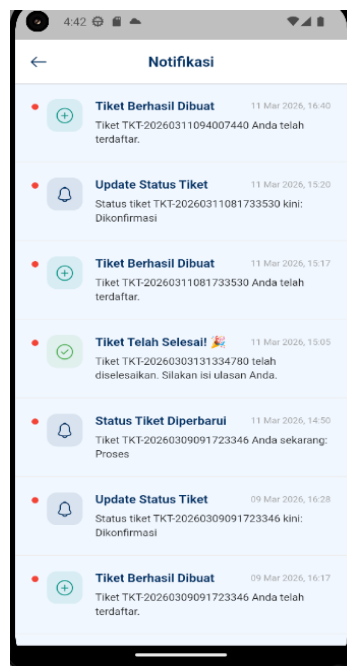
Gambar 4. 30 Implementasi Riwayat Tiket

Gambar 4.30 Merupakan hasil implementasi halaman riwayat tiket, di halaman ini *user* dapat melihat semua tiket yang pernah dibuat dan melihat tanggal dibuat serta tanggal *update* tiketnya.



Gambar 4. 31 Implementasi Detail riwayat Tiket

Gambar 4.31 Merupakan hasil implementasi detail tiket didalam riwayat tiket, halaman ini berisi semua informasi tiket yang telah dibuat.

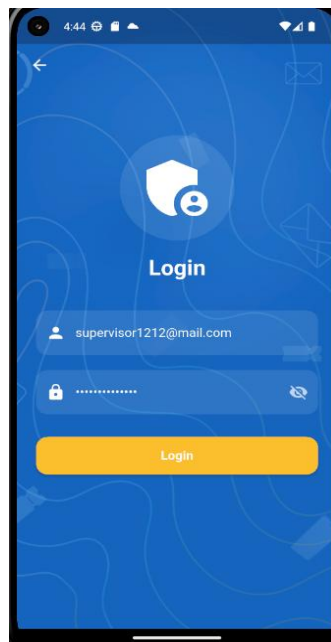


Gambar 4. 32 Implementasi Notifikasi

Gambar 4. 32 Merupakan hasil implementasi halaman notifikasi yang ada didalam aplikasi, halaman ini merupakan halaman tercatatnya semua notifikasi yang masuk ke user.

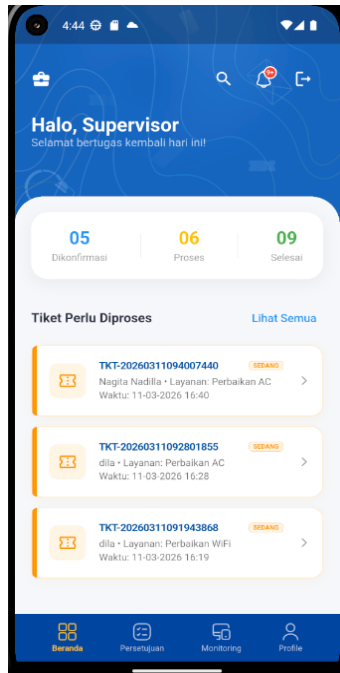
4.1.4.3 Antar Muka Role Supervisor

Implementasi sistem pada sisi Supervisor ditampilkan pada bagian ini. Antarmuka khusus pengelola ini memberikan akses penuh terhadap fungsi manajemen tiket pengaduan. Hal ini mencakup proses masuk ke sistem, pemantauan status laporan secara menyeluruh (*monitoring*), serta fitur alokasi tugas (*assign pelaksana*) untuk mempercepat respon terhadap kendala yang dilaporkan oleh pengguna.



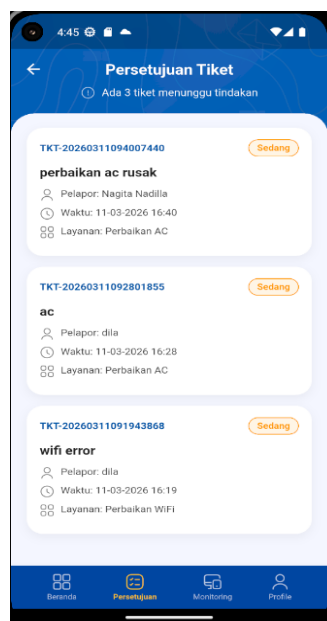
Gambar 4. 33 Implementasi *Login* Pengelola

Gambar 4.33 merupakan antarmuka halaman masuk (*login*) yang diperuntukkan khusus bagi staf pengelola atau *Supervisor*. Sistem keamanan pada halaman ini memastikan bahwa hanya akun dengan hak akses (*role*) Supervisor yang dapat masuk ke panel kendali utama untuk mengelola data pengaduan yang masuk dari masyarakat maupun pegawai.



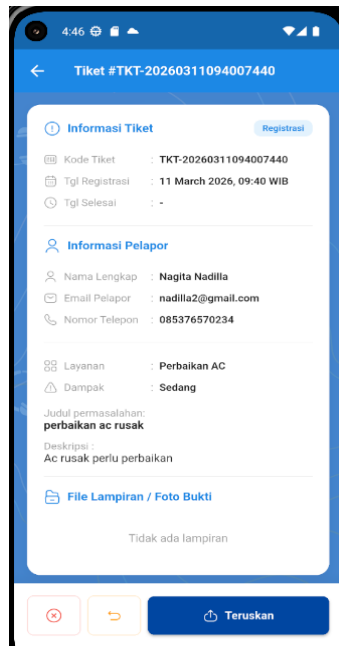
Gambar 4. 34 Implementasi *Dashboard Supervisor*

Gambar 4.34 Merupakan hasil implementasi *dashboard supervisor*, *dashboard* ini menyajikan daftar tiket yang baru masuk dan segera di *assign* dan data tiket yang sedang di proses atau sudah selesai, *dashboard* ini di rancang untuk *supervisor* dapat mengetahui tiket-tiket yang perlu segera di proses.



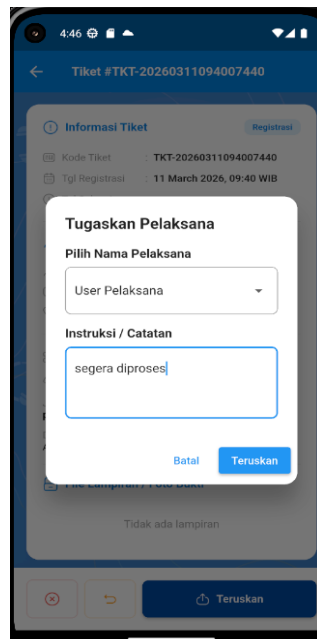
Gambar 4. 35 Implementasi *Page Tiket masuk*

Gambar 4.35 Merupakan hasil implemenatasi halaman persetujuan tiket, atau tiket yang yang masuk tetapi belum memiliki pelaksana yang memprosesnya, maka di halaman ini *supervisor* meninjau informasi dasar dari setiap tiket tersebut sebagai manajemen tiket.



Gambar 4. 36 Implementasi Detail Tiket SPV

Gambar 4.36 Merupakan hasil implementasi detail tiket yang diajukan oleh *user*, dalam halaman ini *supervisor* dapat *assign* pelaksana, jika didalam tiket masih ada yang kurang maka *supervisor* dapat tolak atau mengembalikan tiket ke *user*.



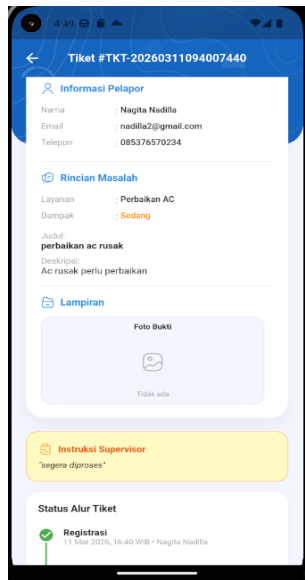
Gambar 4. 37 Implementasi *Assign* Pelaksana

Gambar 4. 37 Merupakan hasil implementasi *supervisor assign* pelaksana dimana supervisor dapat menunjuk pelaksana dan memberikan instruksi.



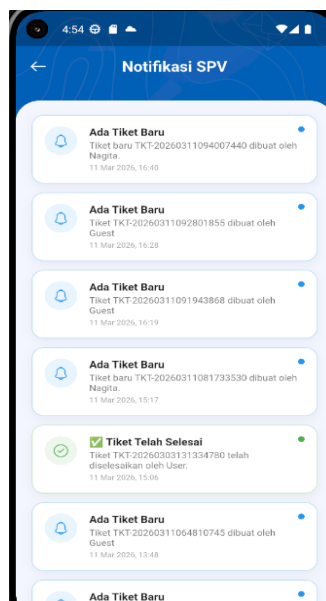
Gambar 4. 38 Implementasi Monitoring Tiket

Gambar 4.38 Merupakan hasil implementasi monitoring tiket semua tiket berdasarkan status tiketnya, didalam halman ini *supervisor* dapat melihat semua tiket yang sedang dikelola.



Gambar 4. 39 Implementasi Detail Tiket Monitoring

Gambar 4.39 Merupakan hasil implementasi detail tiket didalam monitoring tiket, didalam halaman ini *supervisor* dapat melihat detail tiket berserta instruksi dan catatan dari pelaksana jika tiket sudah selesai di proses serta *time line* tiket.

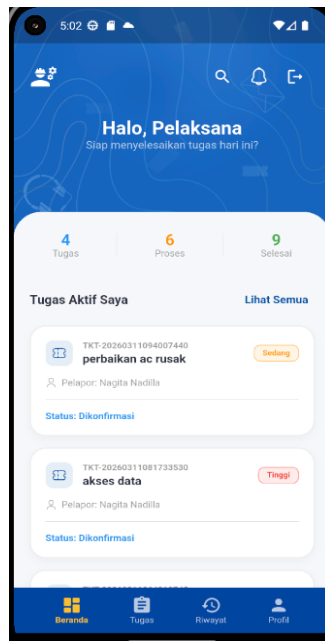


Gambar 4. 40 Implementasi Notifikasi SPV

Gambar 4.40 Merupakan hasil implementasi notifikasi supervisor pada page ini, supervisor diberikan akses untuk meninjau isi pengaduan secara mendalam.

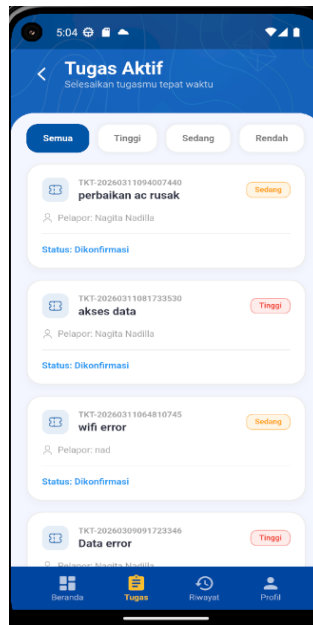
4.1.4.4 Antar Muka Role Pelaksana

Implementasi sistem pada sisi Staf Pelaksana ditampilkan pada bagian ini. Berbeda dengan peran manajerial, antarmuka ini difokuskan untuk memfasilitasi kebutuhan teknis di lapangan. Hal ini mencakup penerimaan instruksi kerja (*assignment*), akses ke rincian kendala teknis yang harus diperbaiki, serta fitur pelaporan progres kerja guna memastikan transparansi penyelesaian masalah di lingkungan *B-Care*



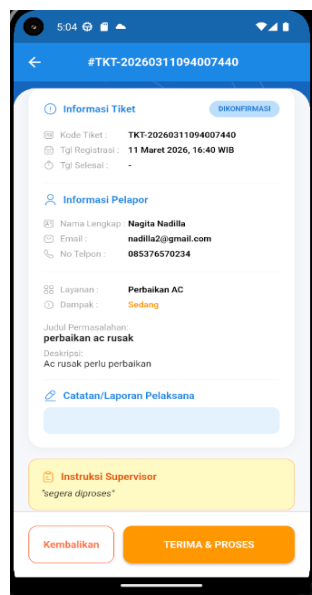
Gambar 4. 41 Implementasi *Dashboard* Pelaksana

Gambar 4.41 Merupakan hasil implementasi *dashboard* pelaksana yang fungsinya sama dengan *dashaboard supervisor* tadi yaitu untuk memantau tiket yang baru masuk ke pelaksana dan statistik status tiket yang sedang di proses atau sudah selesai dikerjakan.



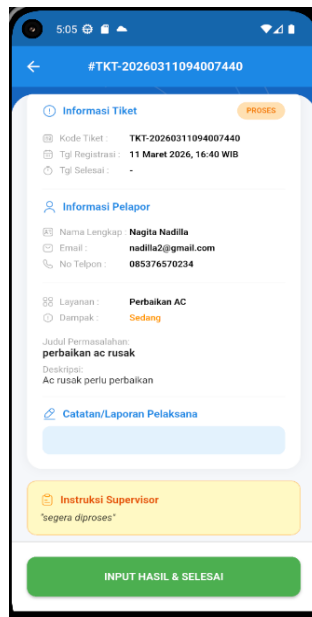
Gambar 4. 42 Implementasi List Tiket Pelaksana

Gambar 4. 42 Merupakan hasil implementasi list tugas yang ada di role pelaksana, didalam halaman ini pelaksana dapat melihat semua tiket yang ditugaskan, dengan status dikonfirmasi atau tiket yang sedang di proses.



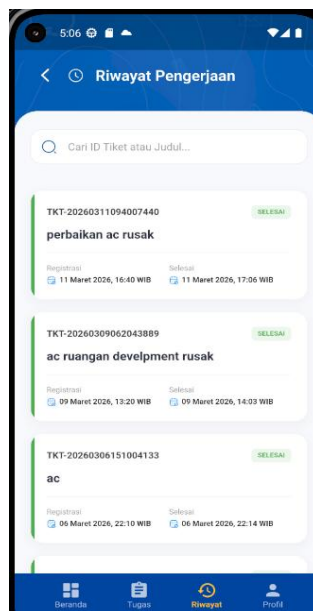
Gambar 4. 43 Implementasi Detail Tiket Status Konfirmasi

Gambar 4.43 Merupakan hasil implementasi detail tiket yang berstatus dikonfirmasi, pada halaman ini pelaksana dapat melihat detail tiket jika tiket tersebut benar harus ia yang mnegelola maka pelaksana klik *button* terima.



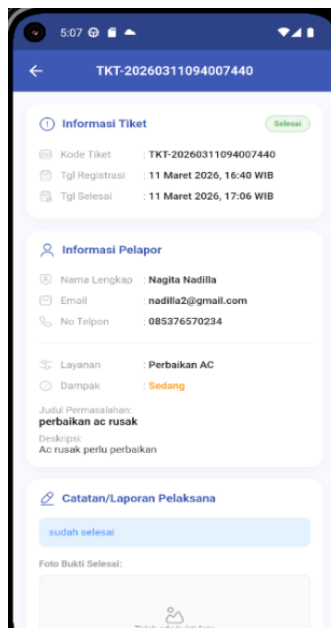
Gambar 4. 44 Implementasi Detail Tiket Status Proses

Gambar 4.44 Merupakan gambar hasil implementasi detail tiket yang berstatus proses didalam halaman ini jika pelaksana sudah menyelesaikan tugasnya maka pelaksana dapat klik *button* input hasil dan selesai, maka akan muncul *pop up* yang harus diisi catatan hasil pengerjaan.



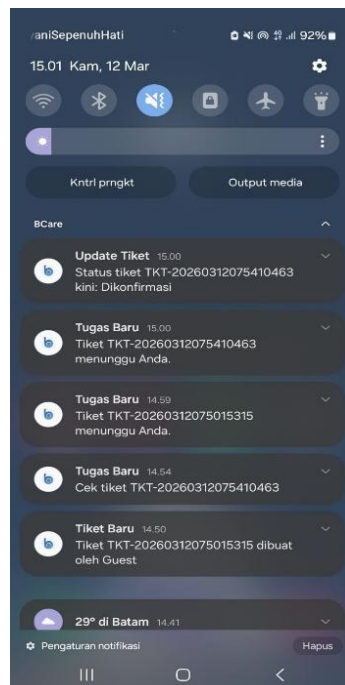
Gambar 4. 45 Implementasi Riwayat Tiket Selesai

Gambar 4. 45 Merupakan hasil implementasi riwayat tiket yang selesai di sisi pelaksana, didalam halaman ini akan tercatat waktu registrasi dan waktu selesai tiket.



Gambar 4. 46 Implementasi Detail Tiket selesai Pelaksana

Gambar 4.46 Merupakan hasil implementasi detail tiket selesai di riwayat tiket pelaksana, di halaman ini, semua data mengenai laporan yang telah selesai dikerjakan tersimpan secara permanen.



Gambar 4. 47 Implementasi *FCM* Notifikasi *SmartPhone*

Implementasi notifikasi berbasis *FCM* pada perangkat pengguna ditunjukkan pada Gambar 4.47 Melalui integrasi ini, sistem secara otomatis mengirimkan pemberitahuan instan terkait pembaruan status laporan.

4.2 Pengujian Sistem

Setelah tahap implementasi selesai dilakukan, langkah selanjutnya adalah pengujian sistem. Pengujian ini bertujuan untuk memastikan bahwa seluruh fungsi dan fitur yang telah dibangun pada aplikasi *B-Care* dapat berjalan dengan baik dan sesuai dengan kebutuhan pengguna yang telah didefinisikan sebelumnya. Pengujian User Acceptance Test (UAT) dilakukan secara langsung kepada pengguna/client untuk mengetahui tingkat kesesuaian sistem terhadap kebutuhan pengguna. Pengujian dilakukan kepada 17 responden dengan menggunakan beberapa skenario penggunaan aplikasi.

Selain pengujian UAT, pengujian blackbox juga dilakukan dari sisi pengelola sistem untuk memastikan seluruh fungsi aplikasi berjalan sesuai kebutuhan. Pengujian API dilakukan menggunakan Postman guna memastikan proses pertukaran data dapat berjalan dengan baik.

4.2.1 Pengujian *Black Box*

Metode pengujian yang digunakan dalam penelitian ini adalah *Black Box Testing*. Pengujian ini berfokus pada fungsionalitas aplikasi tanpa melihat struktur internal kode programnya. Fokus utama dari pengujian ini adalah untuk memvalidasi masukan (*input*) dan memeriksa apakah keluaran (*output*) yang dihasilkan oleh sistem sudah sesuai dengan skenario yang diharapkan. Adapun rangkuman hasil pengujian fitur pada aplikasi B-Care yang mencakup peran Pengguna, *Supervisor*, dan Staf Pelaksana dapat dilihat pada tabel di bawah ini:

1. Pengujian Sisi Pengguna (*User & Guest*)

Pengujian pada sisi pengguna mencakup fungsi autentikasi, manajemen profil, pengajuan laporan kendala, hingga fitur akses tamu (*guest*). Hasil pengujiannya adalah sebagai berikut:

Tabel 4. 7 *Blackbox Testing* Sisi User

No	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Registrasi	Data lengkap & valid	Akun berhasil dibuat	Berhasil
2.	Registrasi	Email sudah terdaftar	Muncul pesan email sudah digunakan	Berhasil
3.	Login	Login dengan email & password valid	Sistem berhasil mengarahkan ke halaman Dashboard	Berhasil
4.	Login	Password atau Email salah	Muncul pesan "Kredensial tidak valid"	Berhasil
5.	Login	Mengosongkan form login lalu klik tombol Login	Muncul pesan peringatan "Email dan Password wajib diisi"	Berhasil
6.	Logout	Klik tombol logout	Kembali ke halaman welcome screen	Berhasil
7.	Buat Tiket	Input data lengkap	Tiket tersimpan dengan status Registrasi	Berhasil
8.	Buat Tiket	Field judul kosong	Muncul validasi	Berhasil
9.	Riwayat Tiket	Membuka menu riwayat	Daftar riwayat tiket tampil sesuai data	Berhasil
10.	Detail Tiket	Klik salah satu tiket	Menampilkan data tiket	Berhasil
11.	Melihat Faq	Klik Faq	Menampilkan pertanyaan dan jawaban faq	Berhasil
12.	Notifikasi User	Setiap membuat tiket maka akan mendapatkan notifikasi hingga tiket selesai	User menerima notifikasi	Berhasil
13.	Akses layanan tanpa login	Guest mencoba akses layanan	Sistem mengarahkan ke halaman list layanan	Berhasil
14.	Buat Tiket	Guest memilih layanan dan membuat tiket	Tiket berhasil dibuat	Berhasil
15.	Search nomor tiket	Guest melihat status tiket dengan serach nomor tiket	Menampilkan detail tiket yang dicari	Berhasil

2. Pengujian Sisi *Supervisor*

Tahap pengujian selanjutnya dilakukan pada hak akses *Supervisor* yang berfokus pada manajemen tiket masuk dan penugasan staf. Detail pengujiannya dapat dilihat pada tabel di bawah ini:

Tabel 4. 8 *Blackbox Testing* Sisi *Supervisor*

1.	Login	Login dengan email & password valid	Sistem berhasil mengarahkan ke halaman Dashboard	Berhasil
2.			Menampilkan jumlah tiket	Berhasil

	Dashboard Supervisor	Membuka dashboard	sesuai status dan shortcut tiket yang perlu di assign	
2.	List tiket masuk	Membuka page persetujuan tiket	Menampilkan list data tiket yang perlu dikonfirmasi	Berhasil
3.	Konfirmasi Tiket dan assign pelaksana	Assign Pelaksana dan Klik konfirmasi + tambahkan catatan	Status berubah menjadi "Dikonfirmasi" dan diteruskan ke pelaksana yang bertugas	Berhasil
4.	Tolak Tiket	Klik tolak + masukkan alasan	Status berubah menjadi "Tolak"	Berhasil
5.	Kembalikan Tiket	Klik kembalikan + masukkan alasan	Status berubah menjadi "Dikembalikan"	Berhasil
6.	Monitoring tiket	Membuka monitoring tiket	Melihat informasi semua tiket yang masuk	
7.	Filter Tiket	Menggunakan fitur filter	Data tampil sesuai filter	Berhasil
8.	Notifikasi Supervisor	Setia proses pada tiket yang masuk	SPV menerima notifikasi	Berhasil

3. Pengujian Sisi Pelaksana

Pengujian terakhir dilakukan pada sisi Staf Pelaksana untuk memvalidasi alur kerja teknis di lapangan, mulai dari penerimaan tugas, pembaruan status pengerjaan, hingga pelaporan hasil akhir tugas yang telah selesai. Hasil pengujiannya adalah sebagai berikut

Tabel 4. 9 *Blackbox Testing* Sisi Pelaksana

1.	Login	Login dengan email & password valid	Sistem berhasil mengarahkan ke halaman Dashboard	Berhasil
2.	Dashboard Pelaksana	Membuka dashboard	Menampilkan jumlah tiket sesuai status dan shortcut tiket yang perlu diproses	Berhasil
3.	Daftar Tiket	Membuka tiket ditugaskan	Hanya tiket yang assign dan ditugaskan yang tampil	Berhasil
4.	Detail Tiket	Membuka detail tiket	Informasi tampil lengkap	Berhasil
5.	Proses Tiket	Klik tombol proses	Status tiket berubah menjadi Proses	Berhasil
6.	Kembalikan Tiket	Klik tombol kembalikan + masukkan alasan	Tiket Kembali ke supervisor dengan status yang sama Dikonfirmasi	Berhasil
7.	Selesaikan Tiket	Klik selesai + masukkan catatan	Status berubah menjadi "Selesai"	Berhasil
8.	Riwayat Tiket pelaksanaan	Membuka list tiket yang sudah selesai	Melihat detail tiket yang sudah selesai	Berhasil

9.	Notifikasi Pelaksana	Jika ada tiket yang ditujukan ke pelaksana	Pelaksana menerima notifikasi	Berhasil
----	----------------------	--	-------------------------------	----------

4.2.2 Pengujian Pengguna (*User Acceptance Testing*)

Pengujian pengguna ini dilakukan untuk mengukur Tingkat penerimaan dan kepuasan terhadap aplikasi yang sudah dikembangkan yaitu aplikasi *B-care*. Pengujian ini disebarkan menggunakan kuisioner melibatkan 17 responden, yang terdiri dari kategori masyarakat umum (58,8%) dan mahasiswa (41,2%). Pengujian menggunakan kuisioner dengan skala likert (1-5) yang berisi 10 pertanyaan fungsional dan antarmuka aplikasi.

Perhitungan hasil kuesioner dilakukan dengan menggunakan rumus persentase skor aktual terhadap skor ideal sebagai berikut:

$$P = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100\%$$

Tabel 4. 10 Hasil Pengujian UAT

No	Pernyataan Kuesioner	Skor Total	Persentase
1.	Tampilan antarmuka B-Care menarik dan profesional	73	85,9%
2.	Penggunaan ikon dan tombol dalam aplikasi sudah jelas fungsinya	78	91,8%
3.	Proses masuk sebagai Guest (Tamu) mudah dilakukan	75	88,2%
4.	Fitur Tracking Tiket memudahkan pencarian status laporan tanpa harus login	74	87,1%
5.	Fitur-fitur yang tersedia dalam aplikasi berjalan dengan baik	73	85,9%
6.	Aplikasi membantu pengguna dalam melakukan pengelolaan tiket atau laporan	75	88,2%
7.	Aplikasi mempermudah pengguna dalam mengelola atau memantau tiket.	75	88,2%
8.	Fitur Notifikasi membantu mengetahui perubahan status tiket secara cepat.	79	92,9%

9.	Penggunaan aplikasi B-Care pada <i>smartphone</i> membantu pengelola merespon kendala secara lebih cepat dan fleksibel saat berada di luar ruangan.	77	90,6%
10.	tampilan manajemen laporan pada <i>smartphone</i> terasa ringkas dan mudah dioperasikan (user-friendly)	79	92,9%
Total			89,17%

Berdasarkan Tabel 4.10 di atas, hasil pengujian pengguna (*User Acceptance Testing*) terhadap 10 pernyataan menunjukkan tingkat kepuasan yang sangat baik. Skor tertinggi didapatkan pada poin pernyataan nomor 8 dan 10 mengenai respon aplikasi yang cepat serta kebermanfaatan aplikasi secara keseluruhan dengan persentase masing-masing sebesar 92,9%.

Secara akumulatif, nilai rata-rata persentase kelayakan dari seluruh aspek adalah sebesar 89,17%. Merujuk pada skala interpretasi skor, nilai tersebut berada pada rentang 81% - 100% yang dikategorikan sebagai 'Sangat Layak'. Dengan demikian, dapat disimpulkan bahwa aplikasi *B-Care* telah memenuhi ekspektasi pengguna dari sisi fungsionalitas, kemudahan penggunaan (*usability*), dan tampilan antarmuka (*interface*).

4.3 Pemeliharaan Sistem

Setelah proses pengujian sistem selesai dilakukan, tahap pemeliharaan dilakukan untuk mengevaluasi aplikasi dan melakukan beberapa penyesuaian terhadap fitur yang telah dikembangkan. Tahap ini bertujuan untuk memperbaiki kesalahan yang ditemukan selama pengujian serta meningkatkan kenyamanan dan efektivitas penggunaan aplikasi.

Berdasarkan hasil evaluasi dan pengujian, dilakukan beberapa perbaikan dan penyesuaian pada sistem agar aplikasi dapat berjalan lebih optimal sesuai kebutuhan pengguna. Dapat dilihat pada tabel berikut ini:

Tabel 4. 11 Peeliharaan Aplikasi

No	Perbaikan Sistem	Tujuan
1.	Perbaikan tampilan halaman login	Meningkatkan kenyamanan pengguna
2.	Penyesuaian Status Tiket	Mempermudah monitoring tiket
3.	Perbaiki validasi input	Mengurangi kesalahan input data
4.	Optimalisasi API simulasi	Memastikan pertukaran data berjalan dengan baik

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada aplikasi *B-Care* berbasis *mobile*, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Keberhasilan Pengembangan: Aplikasi *B-Care* berhasil dibangun dengan mengintegrasikan *framework Flutter* untuk sisi *client (mobile)* dan *Laravel* sebagai penyedia layanan API (*backend*). Sistem ini mampu menangani proses pengiriman keluhan hingga pemantauan status tiket secara mandiri.
2. Solusi Teknis: Penggunaan *API* mandiri dan *dummy data* terbukti efektif sebagai solusi sementara dalam menjalankan fungsionalitas aplikasi di tengah keterbatasan akses ke *API* resmi instansi, sehingga alur proses bisnis tetap dapat disimulasikan dengan baik.
3. Kualitas Aplikasi: Berdasarkan pengujian *Black Box*, seluruh fitur utama telah berfungsi sesuai harapan. Selain itu, hasil pengujian *User Acceptance Testing (UAT)* menunjukkan tingkat kepuasan sebesar 89,17%, yang mengindikasikan bahwa aplikasi ini layak dan mudah digunakan oleh masyarakat atau pengguna.

5.2 Keterbatasan Penelitian

¹Penelitian ini masih memiliki beberapa keterbatasan yang perlu diperhatikan.

1. pengujian sistem belum sepenuhnya melibatkan pihak instansi secara langsung, sehingga validasi dari sisi pengguna internal masih terbatas.
2. Pengujian yang dilakukan masih berfokus pada fungsi sistem menggunakan metode *blackbox* dan uji coba kepada pengguna umum, sehingga belum mencerminkan seluruh kondisi operasional di lingkungan kerja sebenarnya.

3. Sistem yang dikembangkan masih memiliki ruang untuk pengembangan lebih lanjut, baik dari sisi fitur maupun performa agar dapat lebih optimal dalam mendukung kebutuhan pengguna.

Oleh karena itu, penelitian selanjutnya diharapkan dapat melakukan implementasi dan pengujian yang lebih luas sesuai kebutuhan instansi terkait.

5.3 Saran

Untuk pengembangan lebih lanjut agar aplikasi B-Care menjadi lebih sempurna di masa mendatang, penulis menyarankan beberapa hal sebagai berikut:

1. Integrasi *API* Instansi: Diharapkan adanya koordinasi lebih lanjut untuk integrasi dengan database atau *API* resmi agar data yang disajikan lebih akurat dan *real-time*.
2. Fitur Notifikasi Aktif: Menambahkan fitur *Push Notification* agar pengguna mendapatkan informasi pembaruan status keluhan secara instan tanpa perlu melakukan pengecekan secara manual di dalam aplikasi.
3. Peningkatan Keamanan: Menerapkan sistem enkripsi data yang lebih kuat pada proses pengiriman berkas lampiran guna menjaga kerahasiaan data pengguna.

DAFTAR PUSTAKA

- Agustina, D. S., Purwanti, I., Baity, N. L., Oktaviani, L., & Jannah, R. (2024). Peran Teknologi Informasi dalam Meningkatkan Efisiensi Pelayanan Publik. *Manajemen Bisnis Syariah*, 4(II), 141–147. <https://ejournal.an-nadwah.ac.id/index.php/Al-amal/article/view/744/556>
- Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends Blockchain technologies View project Social Media Networks View project. *Article in International Journal of Interactive Mobile Technologies*, 246–270. <https://doi.org/10.3991/ijim.v14i11.13269>
- Belkhir, A., Abdellatif, M., Tighilt, R., Moha, N., Gueheneuc, Y. G., & Beaudry, E. (2019). An observational study on the state of REST API uses in android mobile applications. *Proceedings - 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2019, February 2021*, 66–75. <https://doi.org/10.1109/MOBILESoft.2019.00020>
- De Paula Vieiralves, G., Carla Silva, R., Mark Lobo De Oliveira, J., & Mary Moreira De Melo, A. (2022). Mobile Service Desk: Creating a Mobile Service Desk Application To Open Calls in the It Industry in a Beverage Company. *International Journal of Advanced Research*, 10(10), 1489–1496. <https://doi.org/10.21474/ijar01/15627>
- Japara, E. M., & Arifin, S. (2023). Android application development using flutter framework: Creation of geolocation system module to validate user location coordinates. *AIP Conference Proceedings*, 2734(1). <https://doi.org/10.1063/5.0155337>
- Kimseng, N., Kurnia, D. A., Vuthy, I., Arifin, R. W., & Setiyadi, D. (2024). UI/UX Development Using Figma based on Inclusive Design. *JINAV: Journal of Information and Visualization*, 4(2), 227–234. <https://doi.org/10.35877/454ri.jinav2257>
- Kore, P. P., Lohar, M. J., Surve, M. T., & Jadhav, S. (2022). API Testing Using Postman Tool. *International Journal for Research in Applied Science and Engineering Technology*, 10(12), 841–843. <https://doi.org/10.22214/ijraset.2022.48030>
- Musa, T. (2023). Perancangan helpdesk tracking sistem pada Kantor Desa Tanjung Sari. *Jurnal Persegi Bulat*, 2(1), 1–6. <https://doi.org/10.36490/jurnalpersegibulat.v2i1.909>
- Saragih, Y. Y. M., & Azis, M. A. (2023). Rancang Bangun Aplikasi Service Desk Berbasis Android Sistem Tiket dan Eskalasi Layanan. *Jurnal Komputer Antartika*, 1(2), 54–61. <https://doi.org/10.70052/jka.v1i2.13>
- Sumarno, T., & Mubarak, A. (2021). Aplikasi Pengaduan Masyarakat Terhadap Infrastruktur Berbasis Android Pada Desa Sindangsari. *Competitive*, 16(2), 80–86. <https://doi.org/10.36618/competitive.v16i2.1487>
- Tan, J., Chen, Y., & Jiao, S. (2024). Visual Studio Code in Introductory Computer Science Course: An Experience Report. In *ASEE Annual Conference and Exposition, Conference Proceedings* (Vol. 1, Issue 1). Association for

- Computing Machinery. <https://doi.org/10.18260/1-2--48259>
- Urathal Alias Sri Swathiga, U., Vinodhini, P., & Sasikala, V. (2021). an Interpretation of Dart Programming Language. *Pharmacognosy Magazine*, 75(17), 399–405.
- Asosiasi Penyelenggara Jasa Internet Indonesia. (2024, February 7). *Asosiasi Penyelenggara Jasa Internet Indonesia*. Asosiasi Penyelenggara Jasa Internet Indonesia. Retrieved September 23, 2025, from <https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>
- ind, Karambir, S. T. (2015). A Simulation Model for the Spiral Software Development Life Cycle. *International Journal of Innovative Research in Computer and Communication Engineering*, 03(05), 3823–3830. <https://doi.org/10.15680/ijircce.2015.0305013>
- Putro, D. P., Gunawan, I., & Suryani, P. E. (2022). Software Push Notification Disposisi Persuratan Berbasis Website Menggunakan Firebase Cloud Messaging. *Journal of Information Technology Ampera*, 3(3), 370–381. <https://journal-computing.org/index.php/journal-ita/article/view/330>
- Batam, B. (2025). *BP Batam Kenalkan Inovasi Permohonan Informasi Melalui Aplikasi B- Care*. <https://bpbatam.go.id/bimtek-pengelolaan-website-bp-batam-kenalkan-inovasi-permohonan-informasi-melalui-aplikasi-b-care/>

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

1. Form Interview Analisis Kebutuhan Perangkat Lunak

https://drive.google.com/drive/folders/1QKS7JIaRL-kQ3zvg4MaJhcNXsYqUHb_S?usp=sharing

2. Figma

<https://www.figma.com/design/BLMgBh8VkbxOSwEqhRyfMd/B-care-APP?node-id=0-1&p=f&t=3mOKilGU4MwwUB2h-0>

Lampiran B. Link Produk

1. Link aplikasi

https://github.com/NadillaNagita/App_B_care

2. Link API

<https://github.com/NadillaNagita/B-care-API>

Lampiran C. Dokumen Pengujian UAT

1. Bukti Korespondensi Permintaan API

<https://drive.google.com/file/d/1u32yESZXeIM7OAO0TglbkogSchw6m8HZ/view?usp=sharing>

2. Pertanyaan Kuisisioner

<https://drive.google.com/drive/folders/1bNEGIXb5JolsQlqlIPOMdEj8jYeDaR-b?usp=sharing>

3. Hasil Rekepan Jawaban Kuisisioner

<https://drive.google.com/drive/folders/11gSXLbn2J17SrIDGKdiSDb3hcJYdBmj7?usp=sharing>