**Tugas Besar II3160 Teknologi Sistem Terintegrasi**

**Smart health Online Shop**

**A blue and white logo with a white elephant and text

Description automatically generated**

**Disusun Oleh:**

**Steven Adrian Corne - 18222101**

**Program Studi Sistem dan Teknologi Informasi**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

# **DAFTAR ISI**

[**DAFTAR ISI** 2](#_Toc187439611)

[**1.** **Webapp** 3](#_Toc187439612)

[**2.** **BMC** 3](#_Toc187439613)

[**3.** **Subdomain** 5](#_Toc187439614)

[**4.** **Proposed Architecture (Sequence Diagram)** 7](#_Toc187439615)

[**a.** **Autentikasi:** 7](#_Toc187439616)

[**b.** **Shopping** 7](#_Toc187439617)

[**c.** **Recipe Recommendation:** 8](#_Toc187439618)

[**d.** **Forum:** 9](#_Toc187439619)

[**e.** **Point Service:** 10](#_Toc187439620)

[**f.** **Review:** 10](#_Toc187439621)

[**5.** **SOFTWARE ARCHITECTURE:** 11](#_Toc187439622)

[**6.** **Rencana Implementasi** 13](#_Toc187439623)

[**7.** **Hasil Implementasi:** 15](#_Toc187439624)

[**a.** **Deploy docker:** 15](#_Toc187439625)

[**b.** **Database:** 18](#_Toc187439626)

[**c.** **Sertifikat:** 21](#_Toc187439627)

[**d.** **Frontend:** 21](#_Toc187439628)

# **Webapp**

Webapp kami adalah platform belanja online yang dirancang untuk memberikan pengalaman belanja yang lebih baik dan lebih mudah bagi pelanggan. Dengan menggunakan teknologi canggih seperti AI dan geolocation, kami dapat membantu pelanggan menemukan produk yang mereka cari dengan lebih cepat dan akurat. Fitur-fitur seperti rekomendasi produk, autocomplete alamat, dan integrasi dengan chatbot AI membuat pengalaman belanja online menjadi lebih menyenangkan dan interaktif.

# **BMC**



**Core Capability:**

1. Order Management

* Mengelola seluruh proses pemesanan, mulai dari penempatan order dan checkout, pelacakan status pesanan, hingga pemenuhan pesanan untuk memastikan pengalaman belanja yang lancar.

1. Product Management

* Mengatur katalog produk dan penetapan harga untuk memastikan informasi produk yang akurat dan strategi harga yang kompetitif.

1. Delivery & Logistic Management

* Menangani proses pengiriman dan logistik, termasuk reverse logistics, manajemen mitra pengiriman, dan pelacakan real-time untuk memastikan pengiriman yang efisien.

1. Partner Management

* Mengatur hubungan dengan mitra bisnis, termasuk onboarding agregator, pelacakan performa, pengelolaan komisi, dan penyelesaian sengketa.

1. Customer Experience

* Fokus pada pengalaman pelanggan melalui shopping experience yang optimal, program poin rewards, layanan resep, dan forum komunitas untuk meningkatkan engagement.

**Enabling Capability:**

1. IT Infrastructure & Data Management

* Menyediakan fondasi teknologi melalui platform web app, layanan integrasi, AI services, dan perlindungan data.

1. Business Analytic & Development

* Menganalisis perilaku pelanggan, performa penjualan, dan mengembangkan inovasi melalui predictive analytics dan prototyping.

1. Financial Management

* Mengelola aspek keuangan bisnis, termasuk pendapatan dan pengeluaran, pelaporan keuangan, dan manajemen perpajakan.

1. HR Management

* Menangani manajemen kinerja dan pengelolaan karyawan untuk memastikan operasional yang efektif.

1. Marketing & Branding

* Mengatur strategi pemasaran melalui kampanye, digital marketing, program afiliasi, dan pengembangan strategi brand.

1. Regulatory & Compliance

* Memastikan kepatuhan terhadap regulasi dan standar yang berlaku dalam industri kesehatan dan e-commerce.

# **Subdomain**

A screenshot of a web site

Description automatically generated

Domain yang dipilih yaitu Customer Experience, dengan breakdown subdomain sebagai berikut:

**Core:**

1. Shopping Experience

* Checkout Process: Proses pembayaran yang lancar dan aman
* Cart Management: Pengelolaan keranjang belanja untuk menyimpan item yang akan dibeli
* User Friendly: Antarmuka yang mudah digunakan dan navigasi yang intuitif
* Rating and Review Product: Sistem ulasan dan penilaian produk dari pelanggan

1. Point Management

* Points Rewarding: Sistem pemberian poin reward untuk setiap transaksi
* Points Payment Management: Pengelolaan penggunaan poin untuk pembayaran

1. Recipe Management

* AI LLM Integration: Integrasi AI untuk memberikan rekomendasi resep
* Save Recipe Feature: Fitur untuk menyimpan resep favorit
* Print Recipe: Kemampuan untuk mencetak resep
* Save Recipe: Sistem penyimpanan resep untuk diakses kemudian

1. Community Forum

* Like and Comment System: Fitur interaksi dalam forum komunitas
* Forum Post: Pengelolaan posting dan diskusi dalam forum

**Generic:**

1. Authentication & Authorization

* Email Authentication: Verifikasi melalui email
* Google Authentication: Opsi login menggunakan akun Google
* Authorization Management: Pengelolaan hak akses pengguna

1. Payment

* Bank Transfer: Metode pembayaran melalui transfer bank
* E-Wallet Transfer: Metode pembayaran menggunakan dompet digital

1. Address Management

* Pengelolaan alamat pengiriman pelanggan

**Supporting:**

1. Customer Management

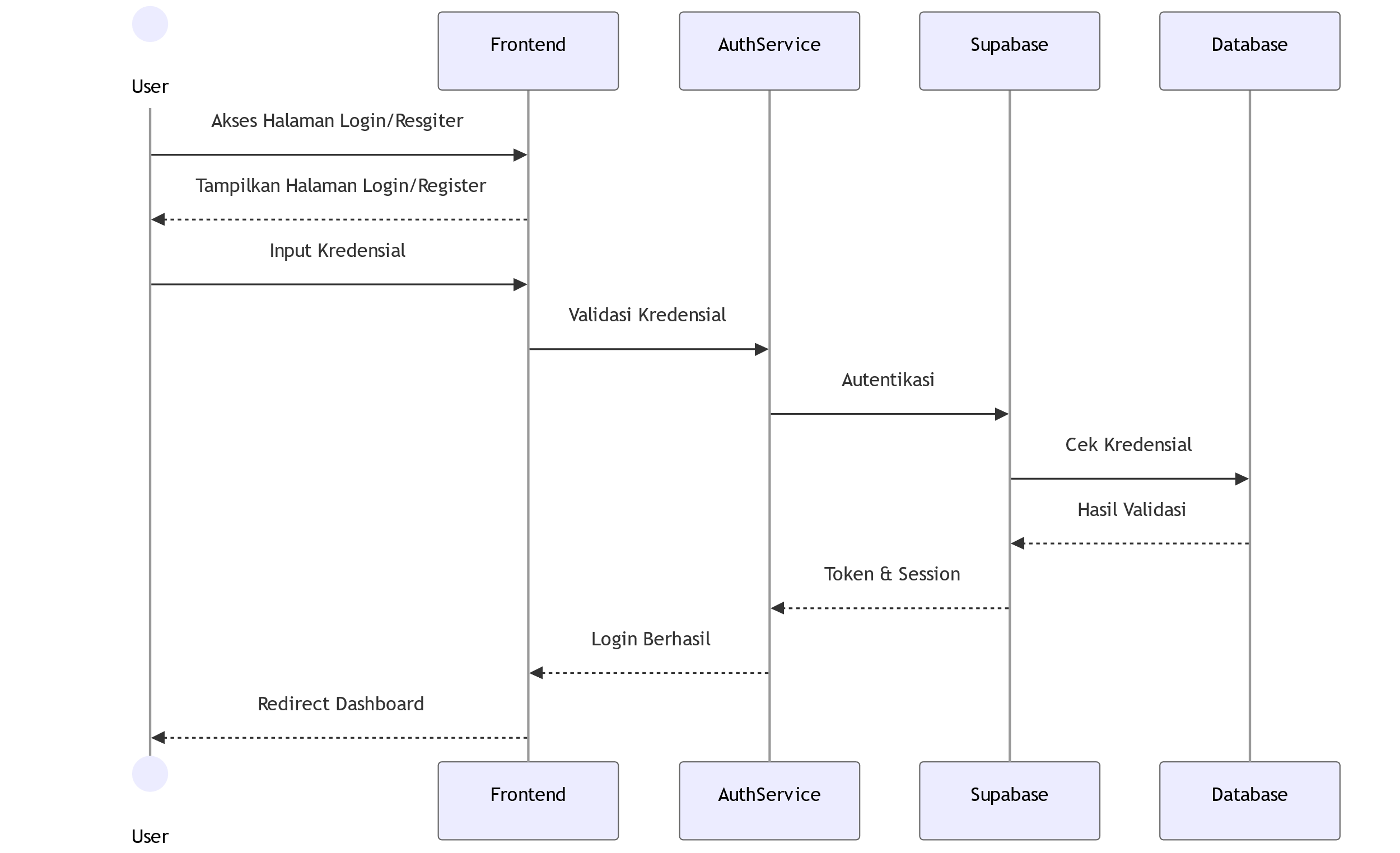
* Customer Support & Care: Layanan dukungan pelanggan
* Customer Feedback Management: Pengelolaan umpan balik pelanggan
* Account & Profile Management: Pengelolaan profil dan akun pengguna

1. Forum Management

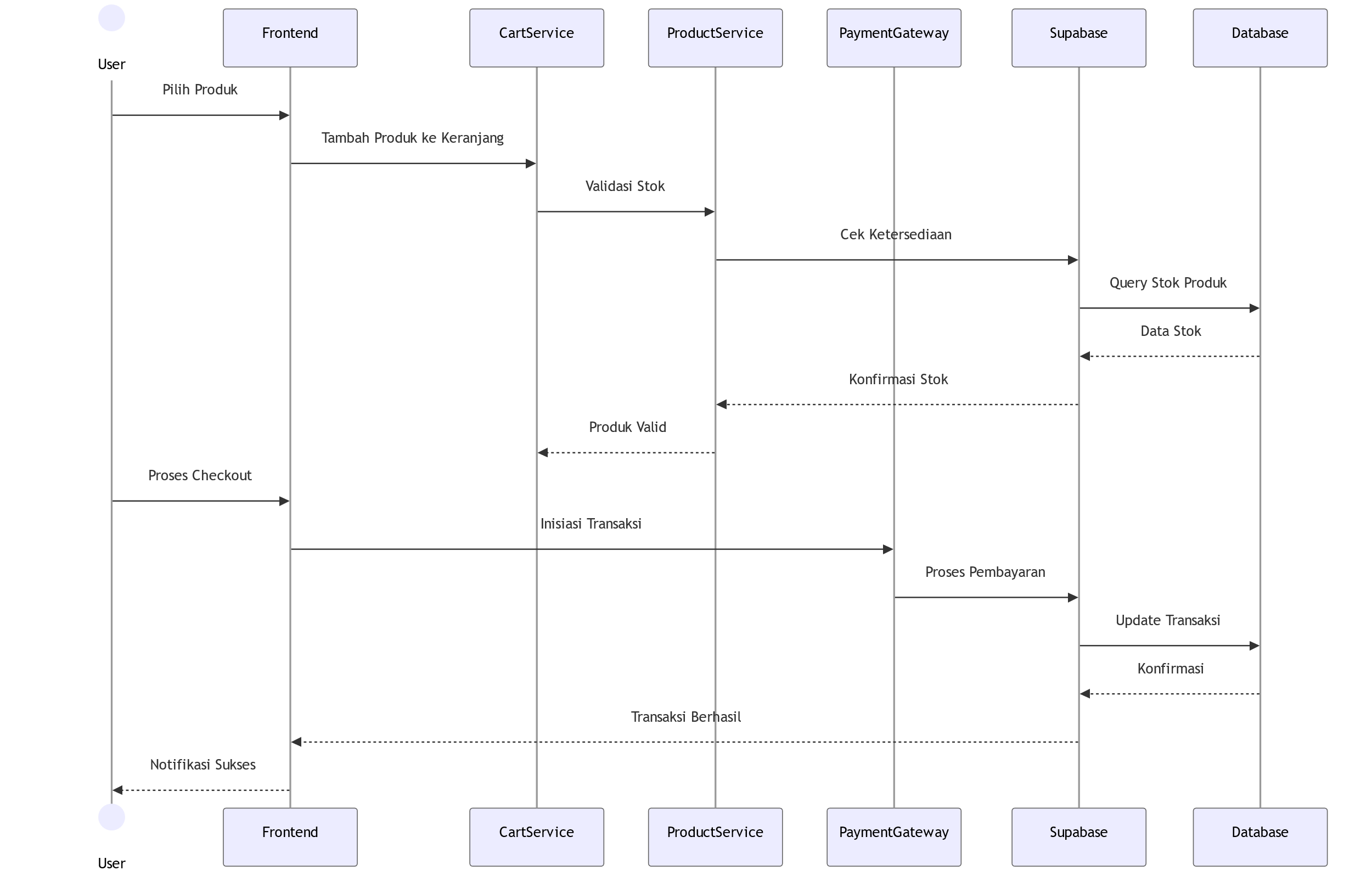
* Latest and Popular Post: Pengelolaan posting terbaru dan populer
* Likes and Comment: Pengelolaan sistem like dan komentar forum

# **Proposed Architecture (Sequence Diagram)**

## **Autentikasi:**

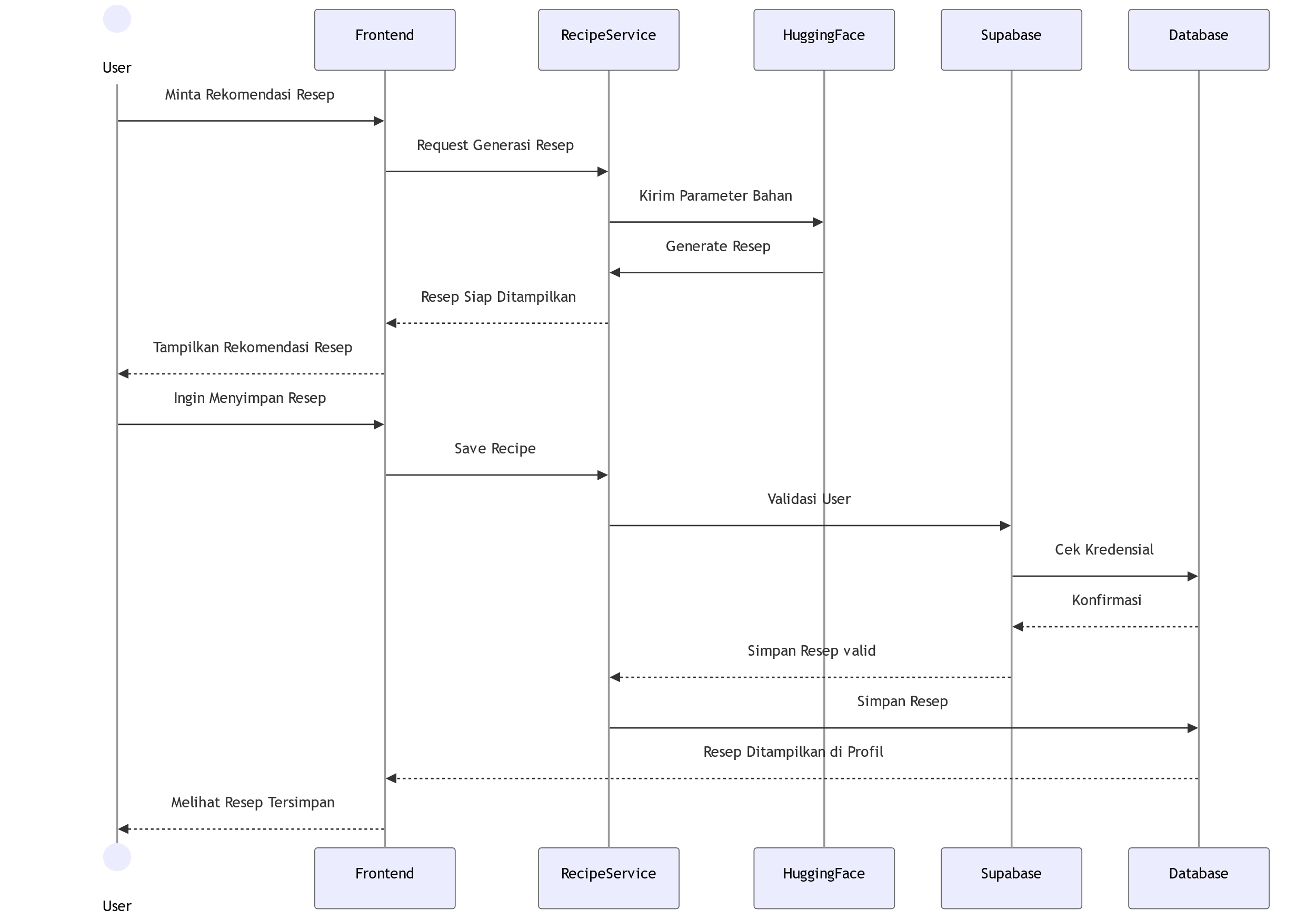


* Diagram ini menunjukkan proses autentikasi user dari login hingga masuk ke dashboard
* Alur dimulai saat user mengakses halaman login/register
* User memasukkan kredensial yang kemudian divalidasi melalui AuthService
* Proses validasi melibatkan Supabase untuk autentikasi dan database untuk cek kredensial
* Setelah validasi berhasil, sistem memberikan token & session
* User diredirect ke dashboard setelah login berhasil
  1. **Shopping:**



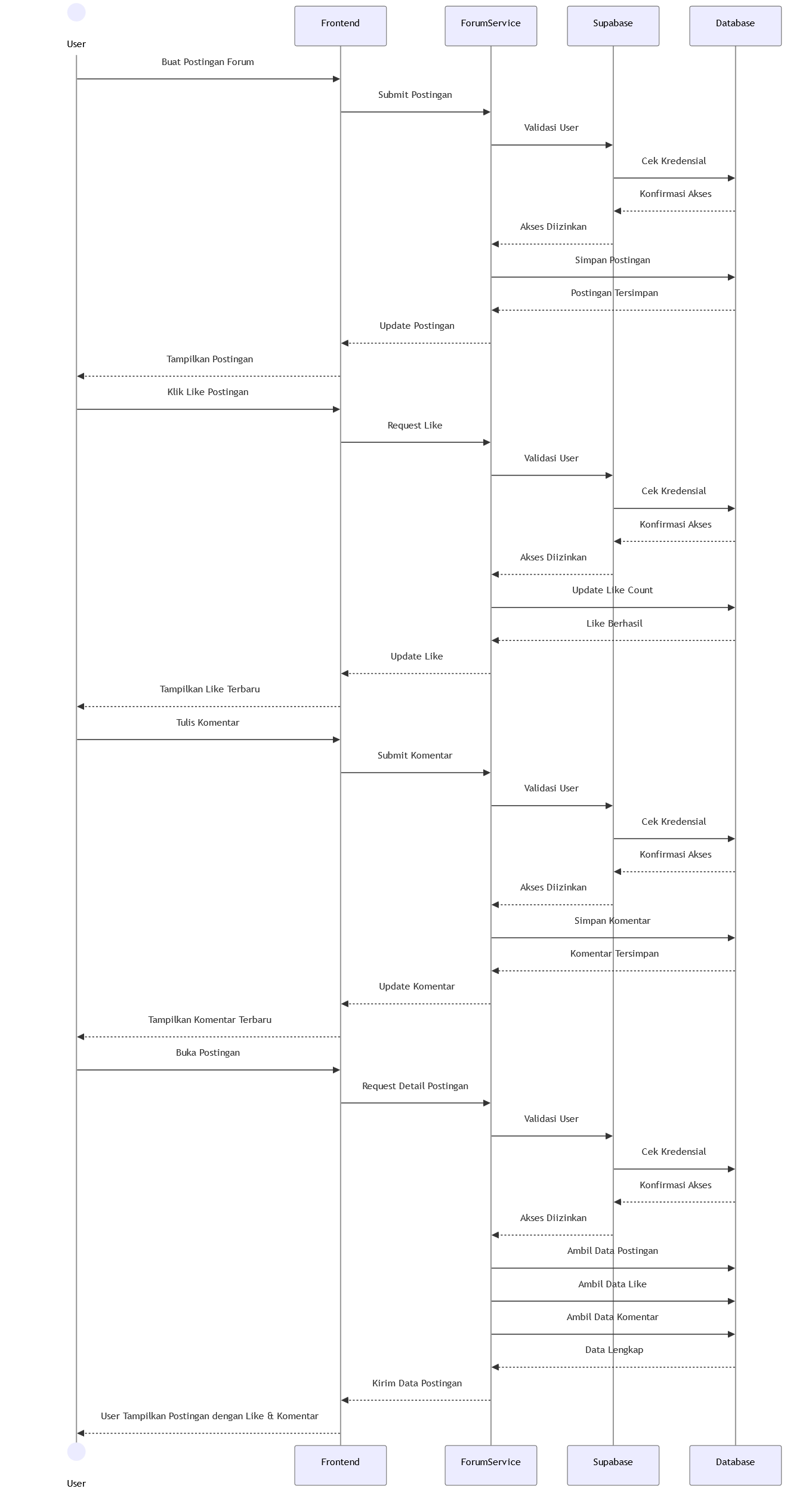
* Menggambarkan proses dari pemilihan produk hingga checkout
* User memilih produk yang ditambahkan ke keranjang melalui CartService
* Sistem melakukan validasi stok melalui ProductService dengan mengecek ketersediaan di database
* Setelah validasi stok, user dapat melakukan checkout
* Proses pembayaran dihandle oleh PaymentGateway
* Transaksi diupdate di database dan user menerima notifikasi sukses

## **Recipe Recommendation:**



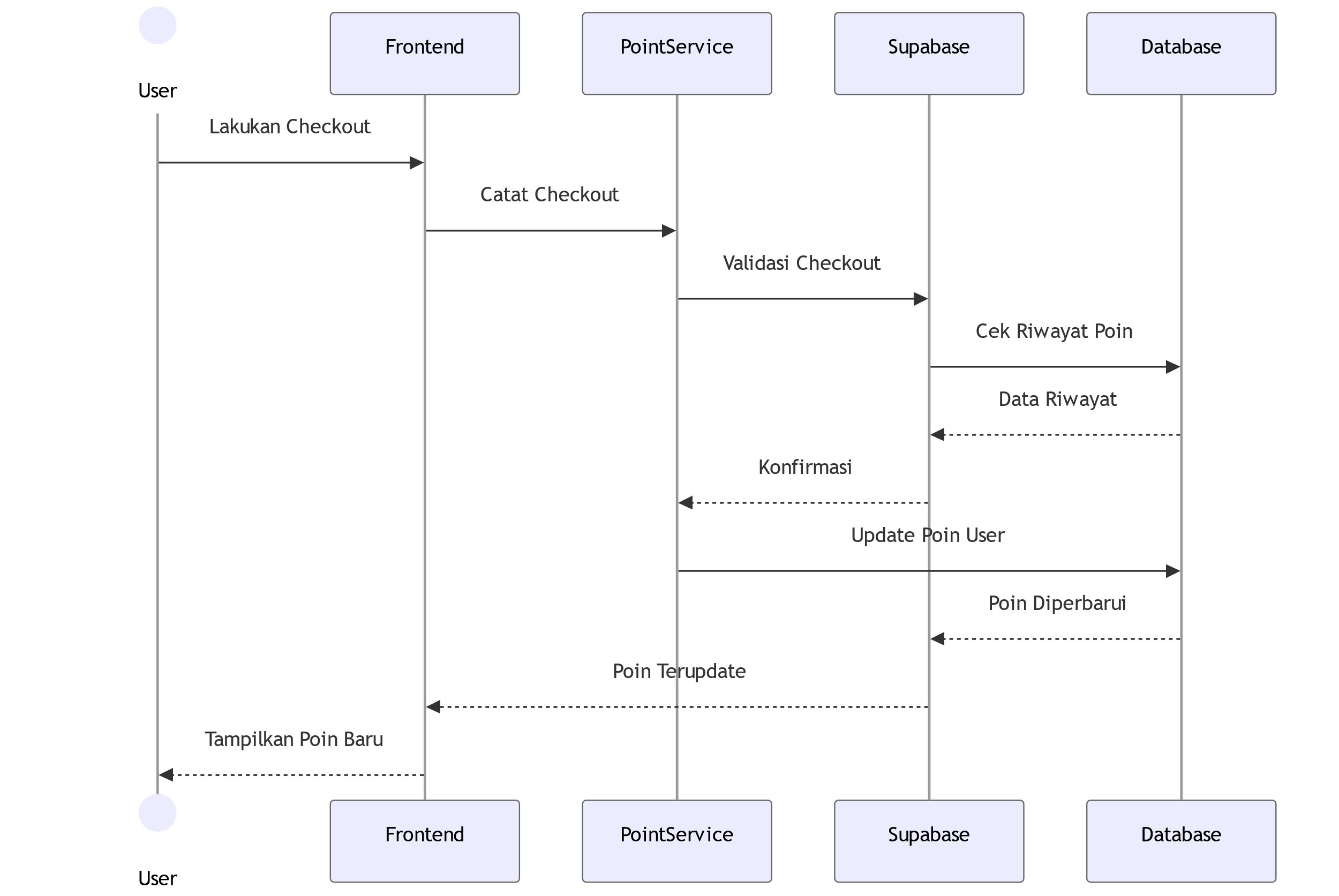
* Menunjukkan proses rekomendasi dan penyimpanan resep
* User meminta rekomendasi resep yang diproses oleh RecipeService
* HuggingFace digunakan sebagai AI service untuk generate resep
* User dapat menyimpan resep yang disukai
* Sistem memvalidasi user dan menyimpan resep ke database
* Resep tersimpan dapat dilihat di profil user

## **Forum:**



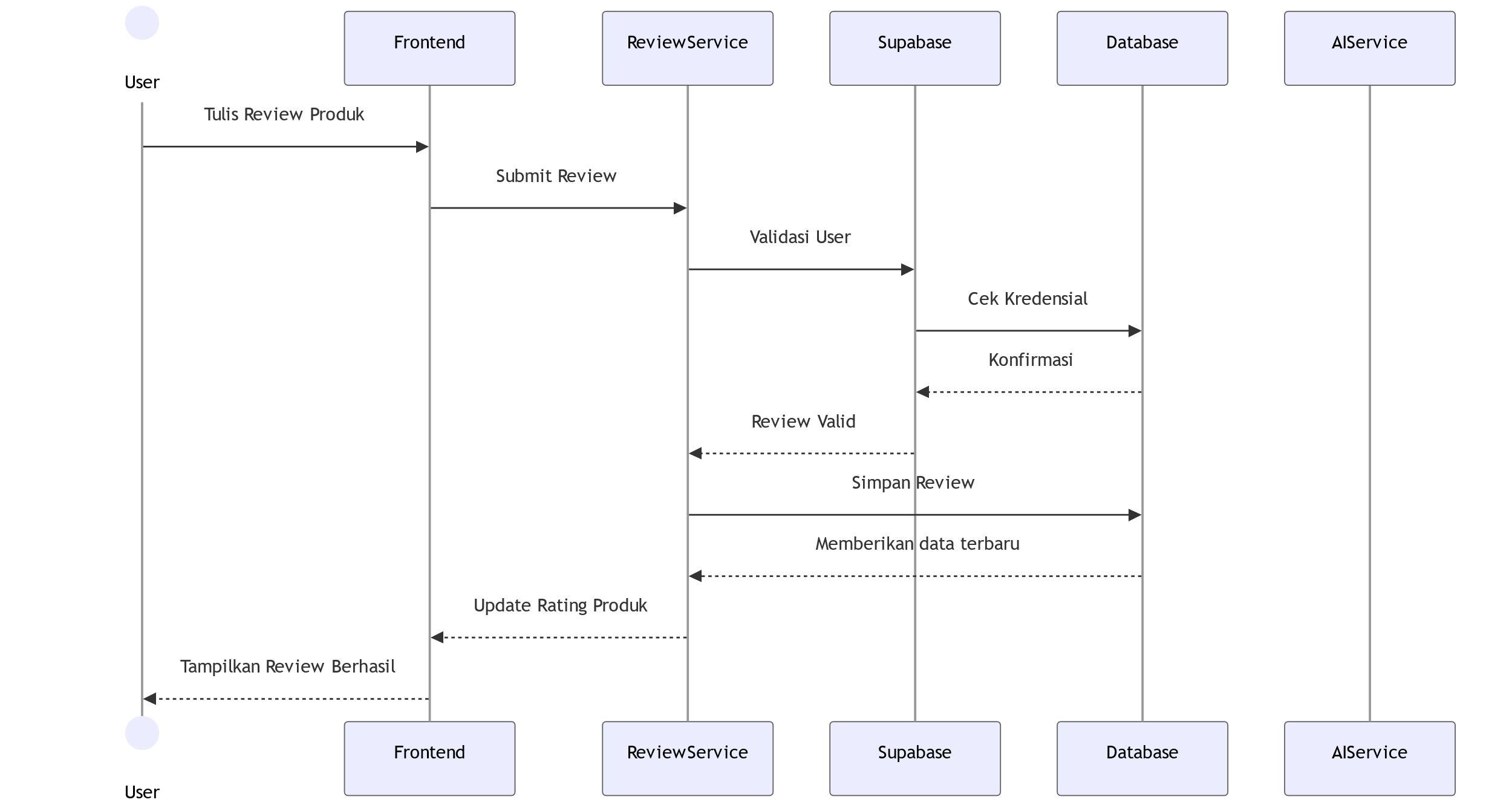
* Menjelaskan interaksi user dalam forum komunitas
* User dapat membuat postingan yang diproses ForumService
* Sistem validasi user sebelum menyimpan postingan
* User dapat memberikan like dan komentar pada postingan
* Setiap interaksi (like/komentar) melalui proses validasi
* Data lengkap postingan termasuk like dan komentar ditampilkan

## **Point Service:**



* Menggambarkan sistem pemberian poin saat checkout
* User melakukan checkout yang dicatat oleh PointService
* Sistem memvalidasi checkout dan mengecek riwayat poin
* Poin user diupdate berdasarkan transaksi
* User dapat melihat poin baru yang diperoleh

## **Review:**



* Menunjukkan proses pemberian review produk
* User menulis review yang diproses ReviewService
* Sistem melakukan validasi user
* Review valid disimpan di database
* Rating produk diupdate berdasarkan review baru
* User menerima konfirmasi review berhasil

# **SOFTWARE ARCHITECTURE:**

A diagram of a flowchart

Description automatically generated

**Tech Stack Web App Anda**

Berikut adalah teknologi yang digunakan dalam setiap layer:

**1. Client Layer (Frontend)**

* **Next.js**: Framework React untuk membangun aplikasi web dengan server-side rendering (SSR), static site generation (SSG), dan routing otomatis.
* **React**: Library JavaScript untuk membangun antarmuka pengguna (UI) yang dinamis dan interaktif.
* **State Management**: Kemungkinan menggunakan **Redux**, **Context API**, atau **Recoil** untuk mengelola state aplikasi.
* **Responsive Design**: Menggunakan **CSS Modules**, **Tailwind CSS**, atau **Styled Components** untuk desain responsif.

**2. Service Layer (Backend)**

* **Authentication Service**: Menggunakan **JWT (JSON Web Tokens)** untuk autentikasi dan **Google OAuth** untuk login pihak ketiga.
* **Product Management Service**: Mengelola CRUD (Create, Read, Update, Delete) produk.
* **AI Recommendation Service**: Mengintegrasikan model AI dari **Hugging Face** untuk rekomendasi produk atau resep.
* **Community Forum Service**: Mengelola postingan, komentar, dan likes di forum.
* **Payment Gateway Service**: Mengintegrasikan **Payment Processors** (seperti Stripe, PayPal, atau Midtrans) untuk transaksi pembayaran.
* **Points & Rewards Service**: Mengelola sistem poin dan hadiah pengguna.
* **Address Service**: Menggunakan **Geolocation API** untuk validasi dan manajemen alamat.

**3. Data Layer (Database)**

* **Supabase PostgreSQL**: Database relasional yang menyimpan semua data aplikasi, termasuk tabel-tabel seperti (data legkap kolom ada dibagian bawah):
  + Users
  + Products
  + Reviews
  + Recipes
  + Cart
  + Forum Posts
  + Forum Likes
  + Forum Comments
  + Points
  + Addresses

**4. External Integrations**

* **Hugging Face AI**: Untuk model AI dan rekomendasi.
* **Google OAuth**: Untuk autentikasi pengguna.
* **Payment Processors**: Untuk integrasi pembayaran.
* **Geolocation API**: Untuk validasi alamat dan lokasi.

**5. Security Layer**

* **JWT Authentication**: Untuk autentikasi pengguna.
* **Role-Based Access Control (RBAC)**: Untuk membatasi akses berdasarkan peran pengguna.
* **Data Encryption**: Mengenkripsi data sensitif seperti password dan informasi pembayaran.
* **Input Validation**: Memvalidasi input pengguna untuk mencegah serangan seperti SQL Injection atau XSS.

# **Rencana Implementasi**

Berikut adalah rencana implementasi aplikasi Anda dalam bentuk **tahapan pengembangan**:

**Tahap 1: Persiapan dan Perencanaan**

* **Tujuan**: Membuat rencana proyek, menentukan fitur utama, dan menyiapkan lingkungan pengembangan.
* **Aktivitas**:
  + Menyiapkan repositori Git (GitHub/GitLab).
  + Menyiapkan lingkungan pengembangan (Node.js, npm/yarn, database lokal).
  + Membuat dokumentasi arsitektur dan API.

**Tahap 2: Pengembangan Frontend**

* **Tujuan**: Membangun antarmuka pengguna (UI) dan logika aplikasi di sisi klien.
* **Aktivitas**:
  + Membuat komponen UI menggunakan React dan Next.js.
  + Mengimplementasikan routing dengan Next.js.
  + Mengintegrasikan state management (Redux/Context API).
  + Membuat desain responsif dengan CSS/Tailwind.

**Tahap 3: Pengembangan Backend**

* **Tujuan**: Membangun layanan backend dan mengintegrasikannya dengan database.
* **Aktivitas**:
  + Membuat API endpoints untuk setiap layanan (autentikasi, produk, forum, dll.).
  + Mengimplementasikan JWT untuk autentikasi.
  + Mengintegrasikan Google OAuth.
  + Membuat logika bisnis untuk setiap layanan (misalnya, manajemen produk, sistem poin).

**Tahap 4: Integrasi Database**

* **Tujuan**: Menyimpan dan mengelola data aplikasi.
* **Aktivitas**:
  + Membuat skema database di Supabase PostgreSQL.
  + Menghubungkan backend dengan database.
  + Mengimplement asikan operasi CRUD untuk tabel-tabel yang diperlukan.

**Tahap 5: Integrasi Layanan Eksternal**

* **Tujuan**: Menghubungkan aplikasi dengan layanan eksternal untuk fungsionalitas tambahan.
* **Aktivitas**:
  + Mengintegrasikan Hugging Face AI untuk rekomendasi.
  + Menghubungkan Payment Processors untuk transaksi.
  + Mengimplementasikan Geolocation API untuk manajemen alamat.

**Tahap 6: Pengujian**

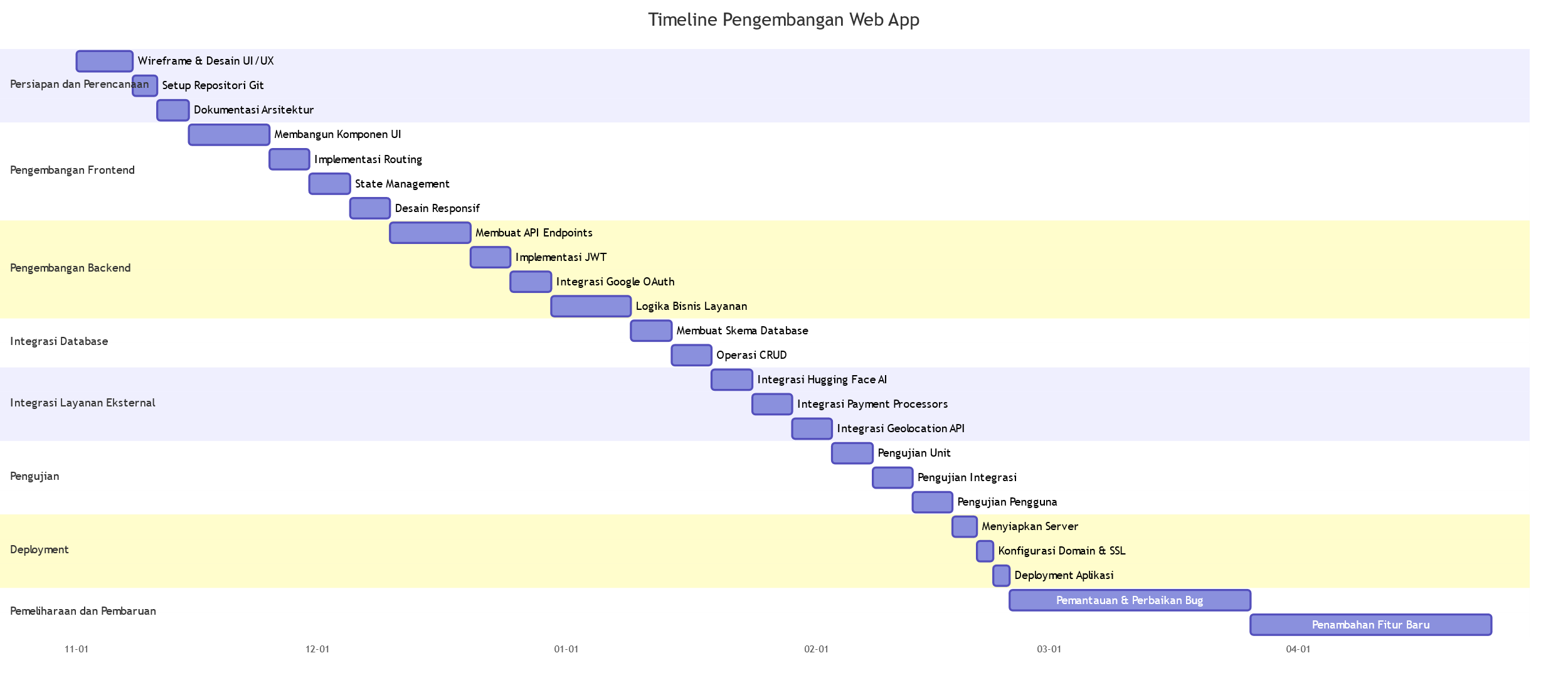
* **Tujuan**: Memastikan aplikasi berfungsi dengan baik dan bebas dari bug.
* **Aktivitas**:
  + Melakukan pengujian unit untuk komponen frontend dan backend.
  + Melakukan pengujian integrasi untuk memastikan semua layanan berfungsi bersama.
  + Melakukan pengujian pengguna untuk mendapatkan umpan balik.

**Tahap 7: Deployment**

* **Tujuan**: Meluncurkan aplikasi ke lingkungan produksi.
* **Aktivitas**:
  + Menyiapkan server untuk hosting (misalnya, Vercel untuk frontend dan Heroku/AWS untuk backend).
  + Mengonfigurasi domain dan SSL.
  + Melakukan deployment aplikasi dan memantau performa.

**Tahap 8: Pemeliharaan dan Pembaruan**

* **Tujuan**: Memastikan aplikasi tetap berjalan dengan baik dan melakukan pembaruan sesuai kebutuhan.
* **Aktivitas**:
  + Memantau log dan performa aplikasi.
  + Mengatasi bug dan masalah yang muncul.
  + Menambahkan fitur baru berdasarkan umpan balik pengguna.



# **Hasil Implementasi:**

## **Deploy docker:**

1. # Base image
2. FROM node:18-alpine AS base
3. # Install dependencies only when needed
4. FROM base AS deps
5. RUN apk add --no-cache libc6-compat
6. WORKDIR /smart-health-tst
7. # Install dependencies
8. COPY package.json package-lock.json\* ./
9. RUN npm ci
10. # Build stage
11. FROM base AS builder
12. WORKDIR /smart-health-tst
13. COPY --from=deps /smart-health-tst/node\_modules ./node\_modules
14. COPY . .
15. # Set environment variables during build
16. ARG NEXT\_PUBLIC\_SUPABASE\_URL
17. ARG NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY
18. ARG NEXT\_PUBLIC\_HUGGING\_FACE\_API\_KEY
19. ARG NEXT\_PUBLIC\_HUGGING\_FACE\_MODEL\_NAME
20. ARG MUSICMATE\_API\_KEY
21. ARG RECIPE\_API\_KEY
22. ARG CHATBOT\_BASE\_URL
23. ARG FORUM\_API\_KEY
24. ENV NEXT\_PUBLIC\_SUPABASE\_URL=$NEXT\_PUBLIC\_SUPABASE\_URL
25. ENV NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=$NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY
26. ENV NEXT\_PUBLIC\_HUGGING\_FACE\_API\_KEY=$NEXT\_PUBLIC\_HUGGING\_FACE\_API\_KEY
27. ENV NEXT\_PUBLIC\_HUGGING\_FACE\_MODEL\_NAME=$NEXT\_PUBLIC\_HUGGING\_FACE\_MODEL\_NAME
28. ENV MUSICMATE\_API\_KEY=$MUSICMATE\_API\_KEY
29. ENV RECIPE\_API\_KEY=$RECIPE\_API\_KEY
30. ENV CHATBOT\_BASE\_URL=$CHATBOT\_BASE\_URL
31. ENV FORUM\_API\_KEY=$FORUM\_API\_KEY
32. RUN npm run build
33. # Production image
34. FROM base AS runner
35. WORKDIR /smart-health-tst
36. ENV NODE\_ENV production
37. RUN addgroup --system --gid 1001 nodejs
38. RUN adduser --system --uid 1001 nextjs
39. # Copy build artifacts
40. COPY --from=builder /smart-health-tst/public ./public
41. COPY --from=builder --chown=nextjs:nodejs /smart-health-tst/.next/standalone ./
42. COPY --from=builder --chown=nextjs:nodejs /smart-health-tst/.next/static ./.next/static
43. COPY --from=builder --chown=nextjs:nodejs /smart-health-tst/ ./
44. COPY --chown=nextjs:nodejs public/ ./public/
45. # Set environment variables in runtime
46. ENV NEXT\_PUBLIC\_SUPABASE\_URL=$NEXT\_PUBLIC\_SUPABASE\_URL
47. ENV NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=$NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY
48. ENV NEXT\_PUBLIC\_HUGGING\_FACE\_API\_KEY=$NEXT\_PUBLIC\_HUGGING\_FACE\_API\_KEY
49. ENV NEXT\_PUBLIC\_HUGGING\_FACE\_MODEL\_NAME=$NEXT\_PUBLIC\_HUGGING\_FACE\_MODEL\_NAME
50. ENV MUSICMATE\_API\_KEY=$MUSICMATE\_API\_KEY
51. ENV RECIPE\_API\_KEY=$RECIPE\_API\_KEY
52. ENV CHATBOT\_BASE\_URL=$CHATBOT\_BASE\_URL
53. ENV FORUM\_API\_KEY=$FORUM\_API\_KEY
54. USER nextjs
55. EXPOSE 3000
56. CMD ["node", "server.js"]

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## **Database:**

1. Addresses:  
   A screenshot of a computer

   Description automatically generated
2. Cart:

A screenshot of a computer

Description automatically generated

1. Category:

A screenshot of a computer

Description automatically generated

1. Forum comment:

A screenshot of a computer

Description automatically generated

1. Forum like:

A screenshot of a computer

Description automatically generated

1. Forum post:

A screenshot of a computer

Description automatically generated

1. Product:

A screenshot of a computer

Description automatically generated

1. Recipe:

A screenshot of a computer

Description automatically generated

1. Review:

A screenshot of a computer

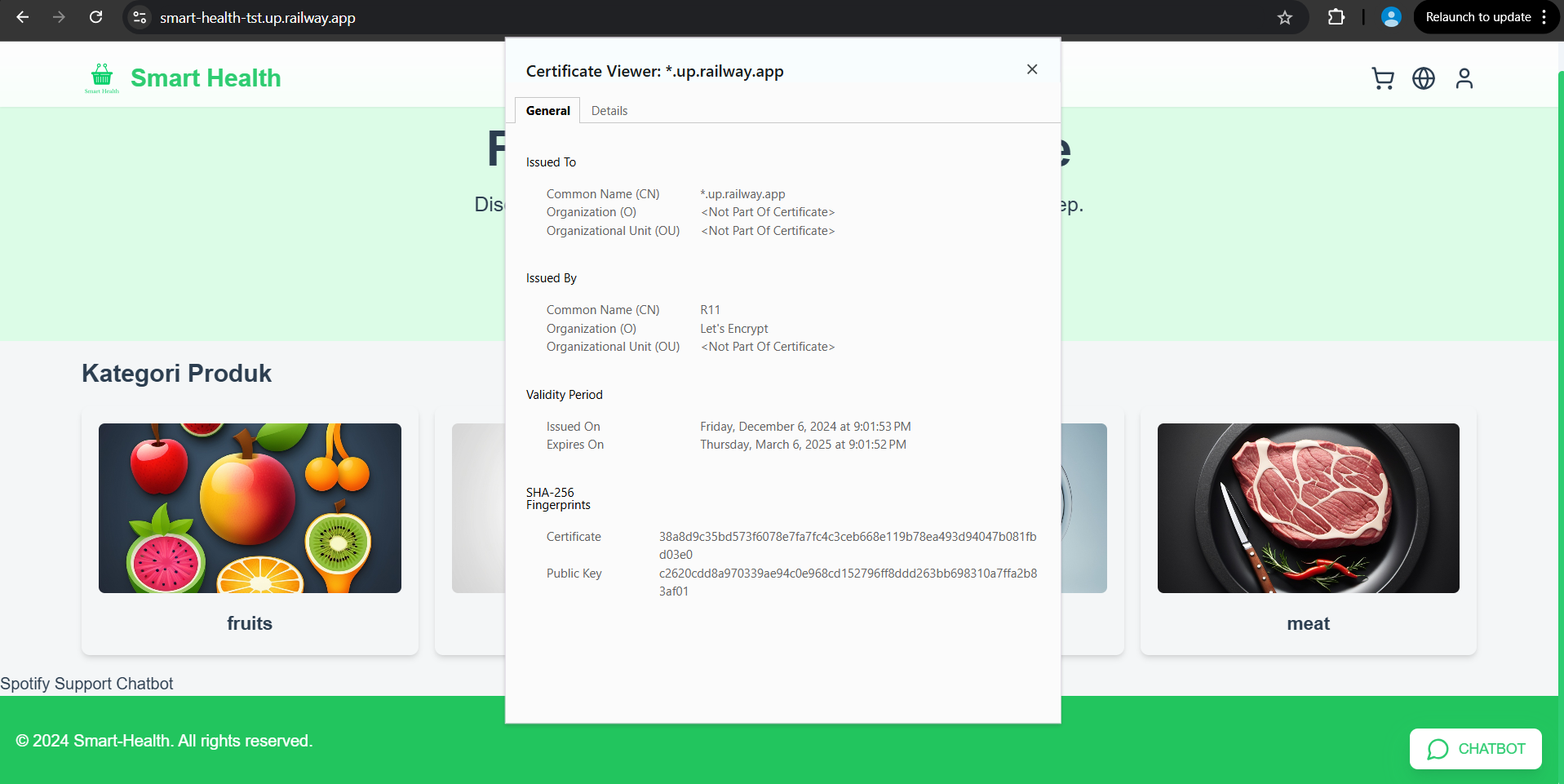
Description automatically generated

1. User:

A screenshot of a computer

Description automatically generated

## **Sertifikat:**



## **Frontend:**

* 1. Home page:

A screenshot of a phone

Description automatically generated

* 1. Login:

A screenshot of a computer

Description automatically generated

* 1. Register:

A screenshot of a computer

Description automatically generated

* 1. Chatbot:

A screenshot of a phone

Description automatically generated

* 1. Shop:

A screenshot of a computer

Description automatically generated

* 1. Cart:

A white rectangular object with a black border

Description automatically generated

A screenshot of a chat

Description automatically generated

* 1. Resep rekomendasi:

A screenshot of a computer

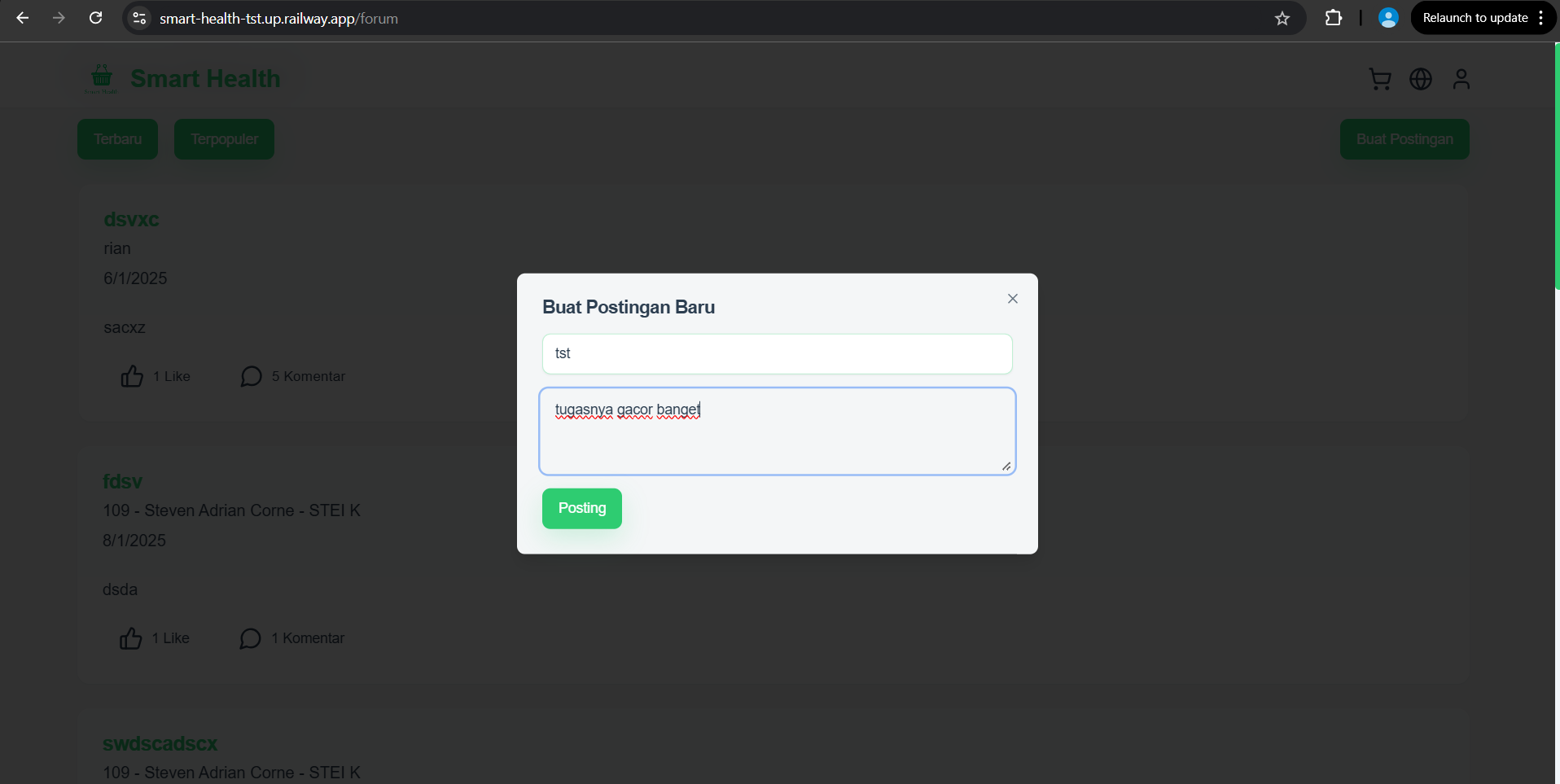
Description automatically generated

* 1. Forum:

A screenshot of a computer

Description automatically generated

* 1. Forum (buat postingan):



* 1. Forum (like dan comment):

A screenshot of a computer

Description automatically generated

* 1. Profile (dashboard):

A screenshot of a computer

Description automatically generated

* 1. Profile (resep tersimpan):

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# **Link:**

* <https://github.com/steven-adrnn/smart-health-tst> (Dokumentasi API ada di README)
* <https://smart-health-tst.up.railway.app> (Link deployment)