

**Tugas Besar II3160 Teknologi Sistem Terintegrasi**  
**Smart Health Online Shop**



**Disusun Oleh:**  
**Steven Adrian Corne - 18222101**

**Program Studi Sistem dan Teknologi Informasi**  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**

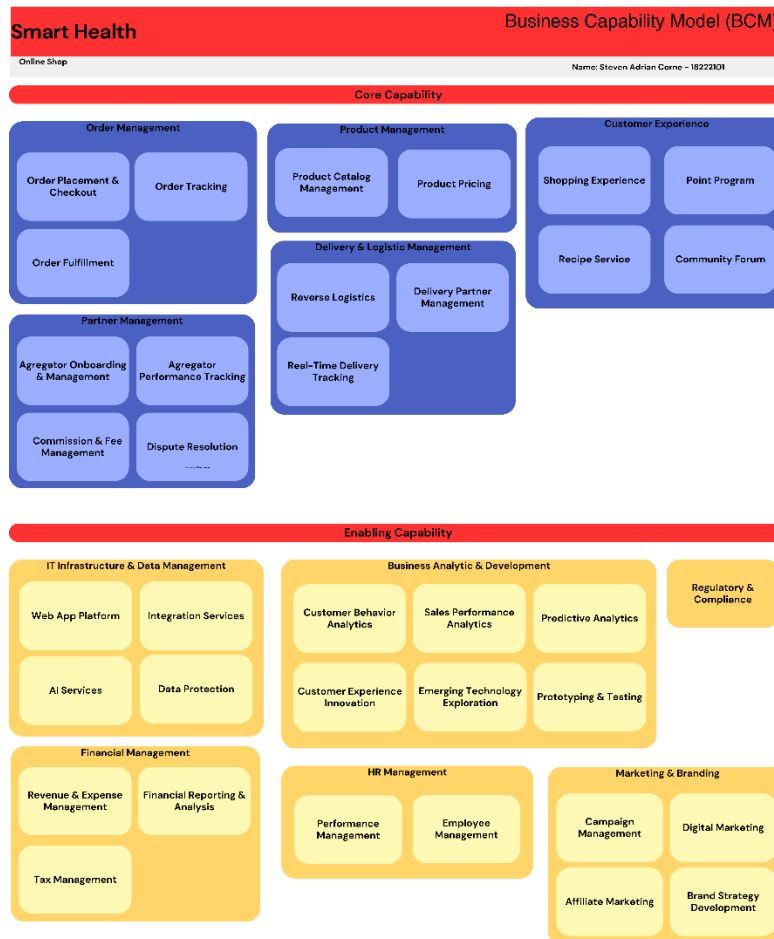
## DAFTAR ISI

<b>DAFTAR ISI</b> .....	<b>2</b>
<b>1. Webapp</b> .....	<b>3</b>
<b>2. BMC</b> .....	<b>3</b>
<b>3. Subdomain</b> .....	<b>5</b>
<b>4. Proposed Architecture (Sequence Diagram)</b> .....	<b>7</b>
<b>a. Autentikasi:</b> .....	<b>7</b>
<b>b. Shopping</b> .....	<b>7</b>
<b>c. Recipe Recommendation:</b> .....	<b>8</b>
<b>d. Forum:</b> .....	<b>9</b>
<b>e. Point Service:</b> .....	<b>10</b>
<b>f. Review:</b> .....	<b>10</b>
<b>5. SOFTWARE ARCHITECTURE:</b> .....	<b>11</b>
<b>6. Rencana Implementasi</b> .....	<b>13</b>
<b>7. Hasil Implementasi:</b> .....	<b>15</b>
<b>a. Deploy docker:</b> .....	<b>15</b>
<b>b. Database:</b> .....	<b>18</b>
<b>c. Sertifikat:</b> .....	<b>21</b>
<b>d. Frontend:</b> .....	<b>21</b>
<b>Link:</b> .....	<b>26</b>

## 1. 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.

## 2. 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.

#### 2. Product Management

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

### 3. Delivery & Logistic Management

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

### 4. Partner Management

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

### 5. 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.

### 2. Business Analytic & Development

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

### 3. Financial Management

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

### 4. HR Management

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

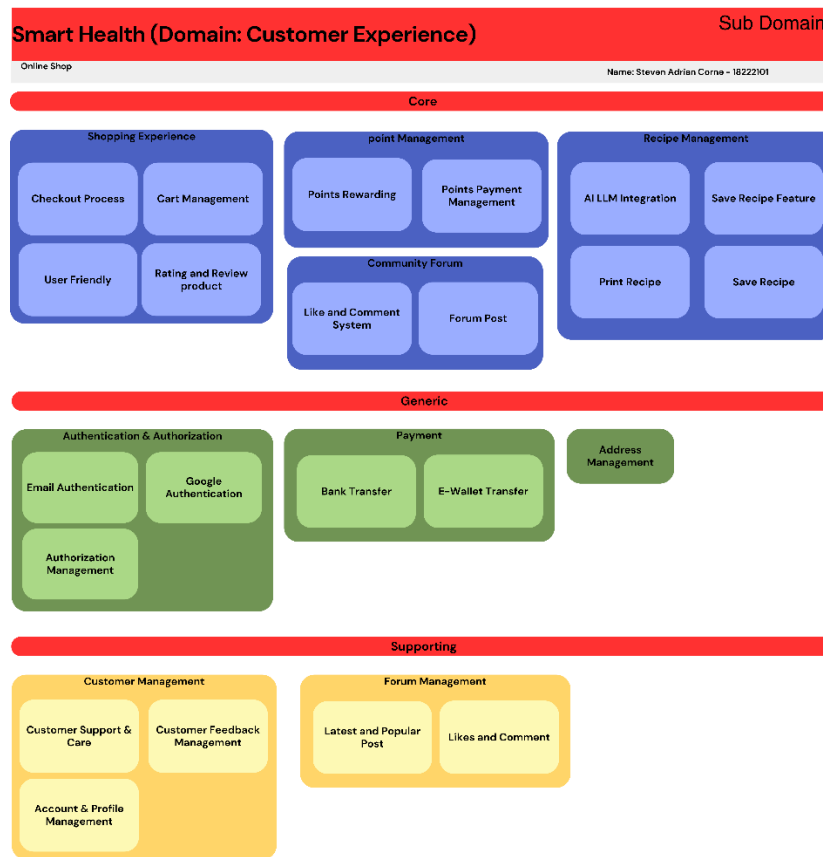
### 5. Marketing & Branding

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

### 6. Regulatory & Compliance

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

### 3. Subdomain



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
2. Point Management
  - Points Rewarding: Sistem pemberian poin reward untuk setiap transaksi
  - Points Payment Management: Pengelolaan penggunaan poin untuk pembayaran
3. 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
4. Community Forum
- Like and Comment System: Fitur interaksi dalam forum komunitas
  - Forum Post: Pengelolaan posting dan diskusi dalam forum

**Generic:**

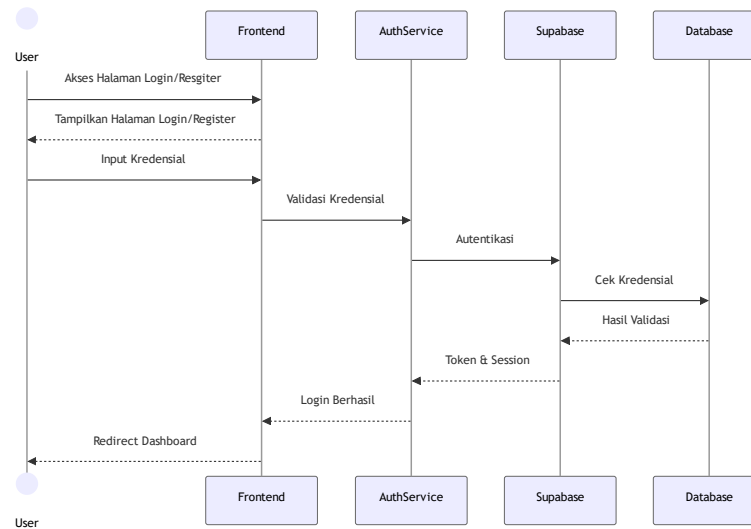
1. Authentication & Authorization
  - Email Authentication: Verifikasi melalui email
  - Google Authentication: Opsi login menggunakan akun Google
  - Authorization Management: Pengelolaan hak akses pengguna
2. Payment
  - Bank Transfer: Metode pembayaran melalui transfer bank
  - E-Wallet Transfer: Metode pembayaran menggunakan dompet digital
3. 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
2. Forum Management
  - Latest and Popular Post: Pengelolaan posting terbaru dan populer
  - Likes and Comment: Pengelolaan sistem like dan komentar forum

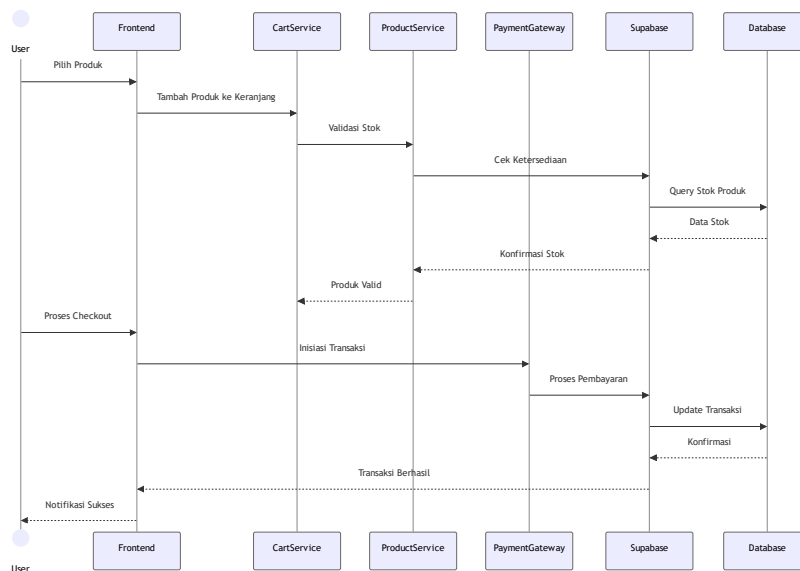
## 4. Proposed Architecture (Sequence Diagram)

### a. 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

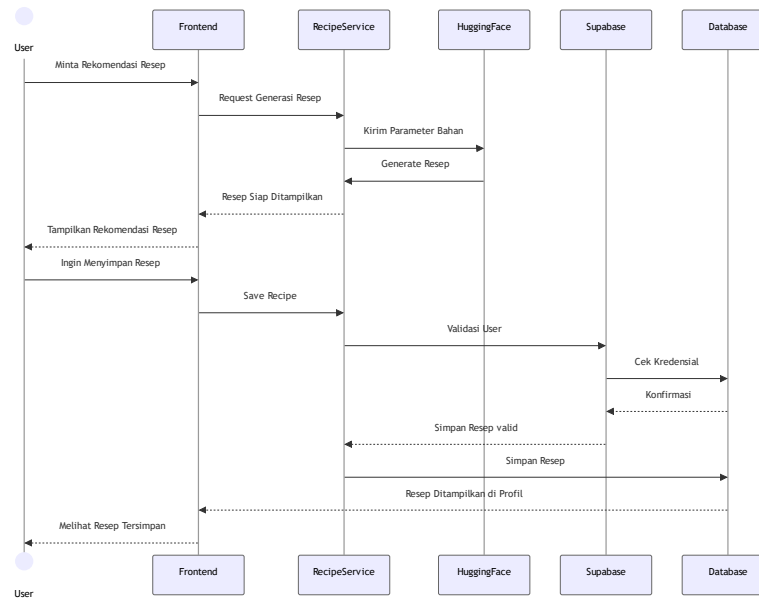
### b. Shopping:



- 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

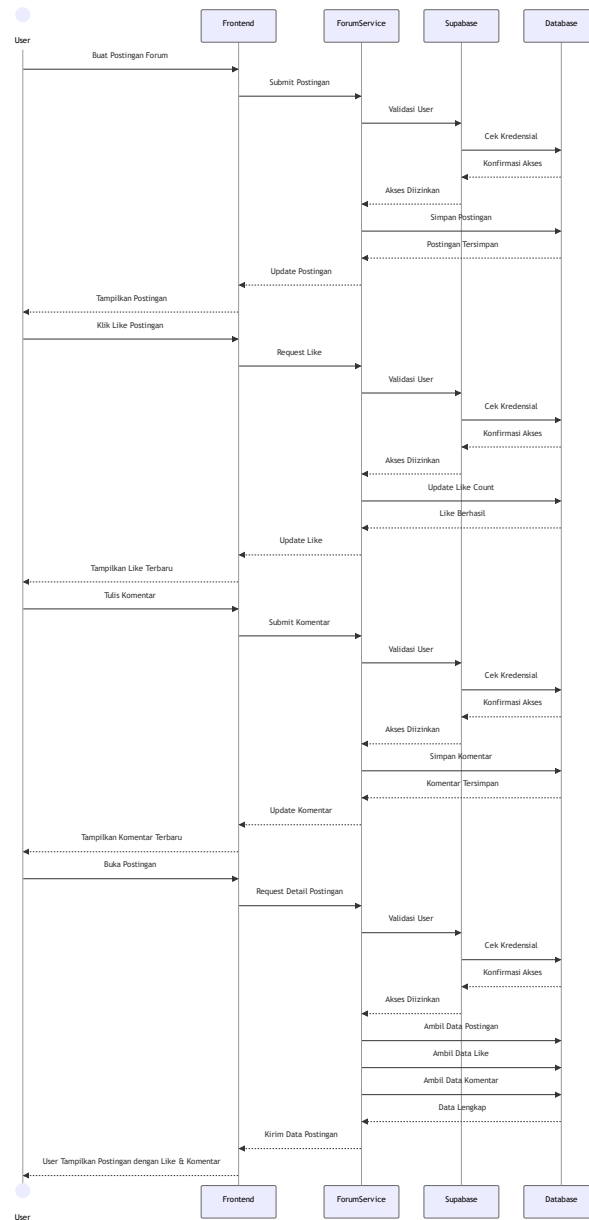
### c. Recipe Recommendation:



- 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

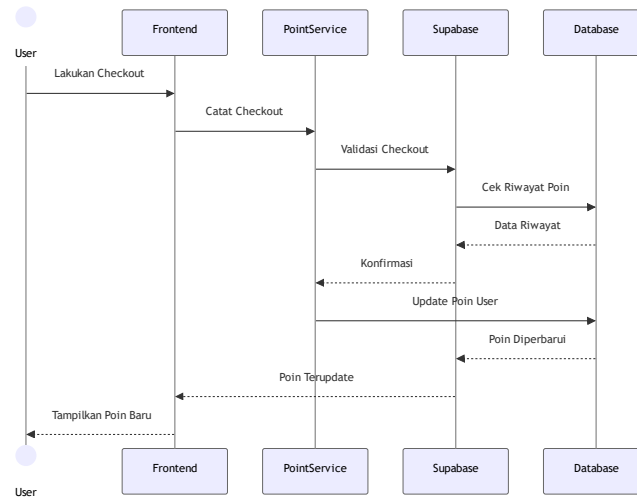


#### d. Forum:



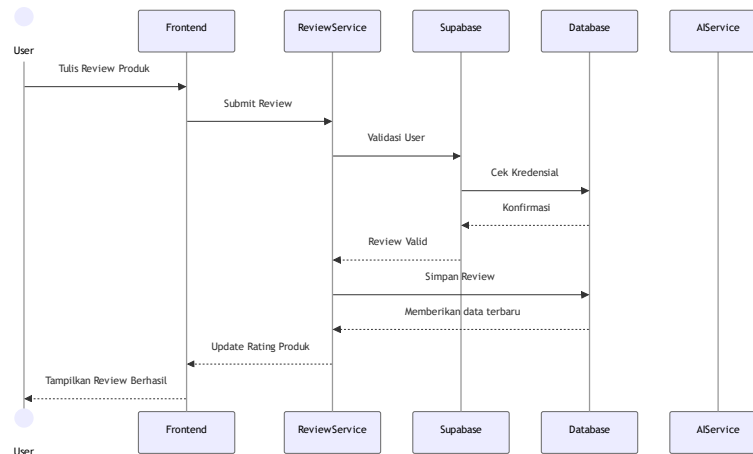
- Menjelaskan interaksi user dalam forum komunitas
- User dapat membuat postingan yang diproses ForumService
- Sistem validasi user sebelum menyimpan postingan
- User dapat memberikan like dan komentar pada postingan
- Setiap interaksi (like/komentar) melalui proses validasi
- Data lengkap postingan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

### e. Point Service:



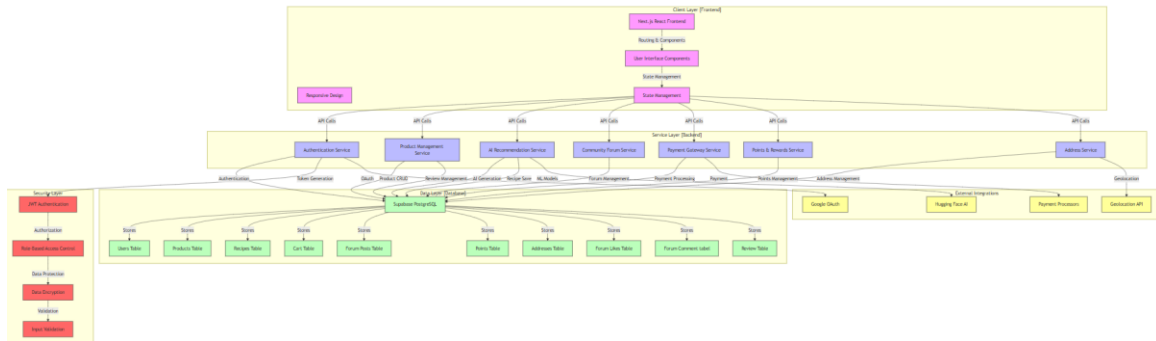
- Menggambarkan sistem pemberian poin saat checkout
- User melakukan checkout yang dicatat oleh PointService
- Sistem memvalidasi checkout dan mengecek riwayat poin
- Poin user diupdate berdasarkan transaksi
- User dapat melihat poin baru yang diperoleh

### f. 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

## 5. SOFTWARE ARCHITECTURE:



### 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 lengkap 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.

## 6. 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.
  - Mengimplementasikan 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**

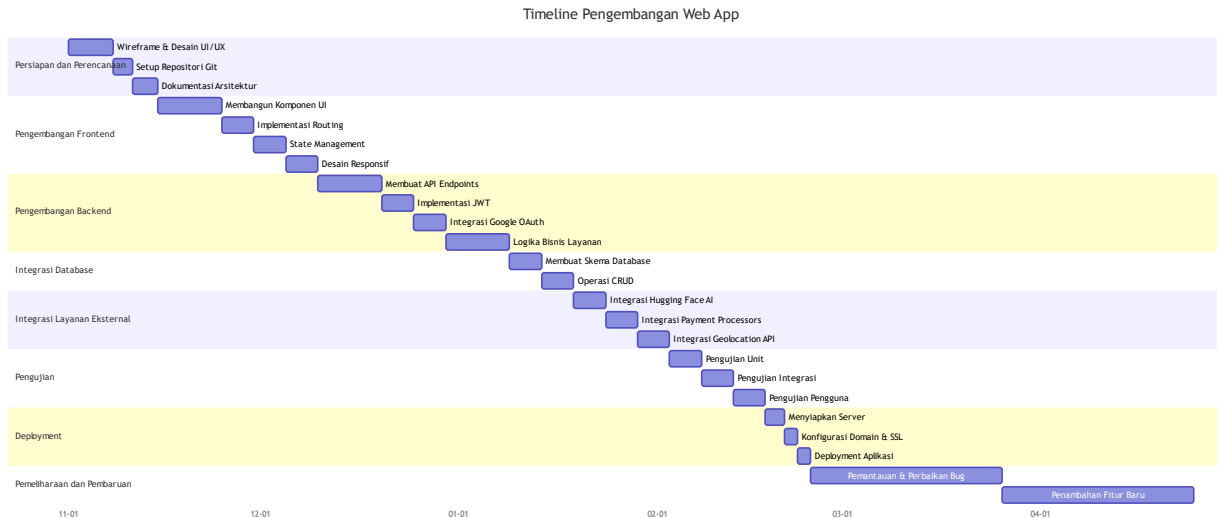
- **Tujuan:** Memastikan aplikasi berfungsi dengan baik dan bebas dari bug.
- **Aktivitas:**
  - Melakukan pengujian unit untuk komponen frontend dan backend.
  - Melakukan pengujian integrasi untuk memastikan semua layanan berfungsi bersama.
  - Melakukan pengujian pengguna untuk mendapatkan umpan balik.

#### **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.



## 7. Hasil Implementasi:

### a. Deploy docker:

```


1. # Base image
2. FROM node:18-alpine AS base
3.
4. # Install dependencies only when needed
5. FROM base AS deps
6. RUN apk add --no-cache libc6-compat
7. WORKDIR /smart-health-tst
8.
9. # Install dependencies
10. COPY package.json package-lock.json* ./
11. RUN npm ci
12.
13. # Build stage
14. FROM base AS builder
15. WORKDIR /smart-health-tst
16. COPY --from=deps /smart-health-tst/node_modules ./node_modules
17. COPY . .
18.
19. # Set environment variables during build
20. ARG NEXT_PUBLIC_SUPABASE_URL
21. ARG NEXT_PUBLIC_SUPABASE_ANON_KEY
22. ARG NEXT_PUBLIC_HUGGING_FACE_API_KEY
23. ARG NEXT_PUBLIC_HUGGING_FACE_MODEL_NAME
24. ARG MUSICMATE_API_KEY
25. ARG RECIPE_API_KEY

```

```
26. ARG CHATBOT_BASE_URL
27. ARG FORUM_API_KEY
28. ENV NEXT_PUBLIC_SUPABASE_URL=$NEXT_PUBLIC_SUPABASE_URL
29. ENV NEXT_PUBLIC_SUPABASE_ANON_KEY=$NEXT_PUBLIC_SUPABASE_ANON_KEY
30. ENV NEXT_PUBLIC_HUGGING_FACE_API_KEY=$NEXT_PUBLIC_HUGGING_FACE_API_KEY
31. ENV
    NEXT_PUBLIC_HUGGING_FACE_MODEL_NAME=$NEXT_PUBLIC_HUGGING_FACE_MODEL_NAME
32. ENV MUSICMATE_API_KEY=$MUSICMATE_API_KEY
33. ENV RECIPE_API_KEY=$RECIPE_API_KEY
34. ENV CHATBOT_BASE_URL=$CHATBOT_BASE_URL
35. ENV FORUM_API_KEY=$FORUM_API_KEY
36.
37. RUN npm run build
38.
39. # Production image
40. FROM base AS runner
41. WORKDIR /smart-health-tst
42.
43. ENV NODE_ENV production
44.
45. RUN addgroup --system --gid 1001 nodejs
46. RUN adduser --system --uid 1001 nextjs
47.
48. # Copy build artifacts
49. COPY --from=builder /smart-health-tst/public ./public
50. COPY --from=builder --chown=nextjs:nodejs /smart-health-tst/.next/standalone ./
51. COPY --from=builder --chown=nextjs:nodejs /smart-health-tst/.next/static ./next/static
52. COPY --from=builder --chown=nextjs:nodejs /smart-health-tst/ ./
53.
54. COPY --chown=nextjs:nodejs public/ ./public/
55.
56. # Set environment variables in runtime
57. ENV NEXT_PUBLIC_SUPABASE_URL=$NEXT_PUBLIC_SUPABASE_URL
58. ENV NEXT_PUBLIC_SUPABASE_ANON_KEY=$NEXT_PUBLIC_SUPABASE_ANON_KEY
59. ENV NEXT_PUBLIC_HUGGING_FACE_API_KEY=$NEXT_PUBLIC_HUGGING_FACE_API_KEY
60. ENV
    NEXT_PUBLIC_HUGGING_FACE_MODEL_NAME=$NEXT_PUBLIC_HUGGING_FACE_MODEL_NAME
61. ENV MUSICMATE_API_KEY=$MUSICMATE_API_KEY
62. ENV RECIPE_API_KEY=$RECIPE_API_KEY
63. ENV CHATBOT_BASE_URL=$CHATBOT_BASE_URL
64. ENV FORUM_API_KEY=$FORUM_API_KEY
```




```
65.  
66.USER nextjs  
67.  
68.EXPOSE 3000  
69.  
70.CMD ["node", "server.js"]
```

 smart-health-tst e65a236 Jan 10, 2025 7:36 PM X

ACTIVE smart-health-tst.up.railway.app

Details Build Logs Deploy Logs HTTP Logs Filter logs using \*\*, (), AND, OR, -

Date (GMT+7)	Message
You reached the start of the range → Jan 10, 2025 7:36 PM	
Jan 10 19:36:24	
Jan 10 19:36:24	[Region: us-west1]
Jan 10 19:36:24	-----
Jan 10 19:36:24	Using Detected Dockerfile
Jan 10 19:36:24	-----
Jan 10 19:36:24	context: ea97d5b8ce629889c1532e9458974508
Jan 10 19:36:25	#0 building with "builder-Xqd-" instance using docker-container driver
Jan 10 19:36:25	
Jan 10 19:36:25	#1 [internal] load build definition from Dockerfile
Jan 10 19:36:25	#1 transferring dockerfile: 2.14kB done
Jan 10 19:36:25	#1 DONE 0.0s
Jan 10 19:36:25	

 smart-health-tst e65a236 Jan 10, 2025 7:36 PM X

ACTIVE smart-health-tst.up.railway.app

Details Build Logs Deploy Logs HTTP Logs Filter logs using \*\*, (), AND, OR, -

Date (GMT+7)	Message
Jan 10 19:37:55	#21 exporting config sha256:f4dc7ce4851b9556773ea9c8ba180acb8c1a78f1221c9286ab081dd882badb done
Jan 10 19:37:55	#21 exporting attestation manifest sha256:3ffe5e46f8cc4bda1170616f6e43fad03dbe9cc9d464cc746f85e949c730f6c 0.0s done
Jan 10 19:37:55	#21 exporting manifest list sha256:e3bab3ed006a50332438ef02dc79a8eb2d04ba3177959f348ede8cd4cd6ad629 done
Jan 10 19:37:55	
Jan 10 19:37:55	#21 pushing layers
Jan 10 19:37:55	
Jan 10 19:37:59	#21 pushing layers 4.1s done
Jan 10 19:37:59	#21 pushing manifest for us-west1.registry.rly.net/8790e2a6-906f-4471-8286-6ef1e4a098f6:e65a236e-2d50-4ca1-8605-1a6078acd5959sha256:e3bab3ed006a50332438ef02dc79a8eb2d04ba3177959f348ede8cd4cd6ad629
Jan 10 19:37:59	
Jan 10 19:38:00	#21 pushing manifest for us-west1.registry.rly.net/8790e2a6-906f-4471-8286-6ef1e4a098f6:e65a236e-2d50-4ca1-8605-1a6078acd5959sha256:e3bab3ed006a50332438ef02dc79a8eb2d04ba3177959f348ede8cd4cd6ad629 0.8s done
Jan 10 19:38:00	#21 DONE 12.8s
Jan 10 19:38:00	
Jan 10 19:38:00	Build time: 95.60 seconds

## b. Database:

### a. Addresses:

Update table addresses

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
user_id	uuid	auth.uid()	<input type="checkbox"/>
address	T text	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>
latitude	# numeric	NULL	<input type="checkbox"/>
longitude	# numeric	NULL	<input type="checkbox"/>
street	T text	NULL	<input type="checkbox"/>
city	T text	NULL	<input type="checkbox"/>
province	T text	NULL	<input type="checkbox"/>
postal_code	T text	NULL	<input type="checkbox"/>

### b. Cart:

Update table cart

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
user_id	uuid	auth.uid()	<input type="checkbox"/>
product_id	uuid	NULL	<input type="checkbox"/>
quantity	# int4	1	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>
address_id	uuid	NULL	<input type="checkbox"/>

### c. Category:

Update table categories

☐ Enable Realtime  
Broadcast changes on this table to authorized subscribers

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	NULL	<input checked="" type="checkbox"/>
name	T varchar	NULL	<input type="checkbox"/>
images	T varchar	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>

### d. Forum comment:

Update table forum\_comments

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	uuid_generate_v4()	<input checked="" type="checkbox"/>
post_id	uuid	NULL	<input type="checkbox"/>
user_id	uuid	NULL	<input type="checkbox"/>
content	T text	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>

e. Forum like:

Update table forum\_likes

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	uuid_generate_v4()	<input checked="" type="checkbox"/>
post_id	uuid	NULL	<input type="checkbox"/>
user_id	uuid	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>

f. Forum post:

Update table forum\_posts

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	uuid_generate_v4()	<input checked="" type="checkbox"/>
user_id	uuid	NULL	<input type="checkbox"/>
title	T text	NULL	<input type="checkbox"/>
content	T text	NULL	<input type="checkbox"/>
category	T text	NULL	<input type="checkbox"/>
likes_count	# int4	0	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>

g. Product:

Update table products

Columns [About data types](#)

Name	Type	Default Value	Primary
id	T uuid	NULL	<input checked="" type="checkbox"/>
name	T varchar	NULL	<input type="checkbox"/>
price	# numeric	NULL	<input type="checkbox"/>
image	T text	NULL	<input type="checkbox"/>
category	varchar	NULL	<input type="checkbox"/>
description	T text	NULL	<input type="checkbox"/>
farm	T text	NULL	<input type="checkbox"/>
rating	# int4	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>
quantity	# int4	0	<input type="checkbox"/>

h. Recipe:

Update table recipes

Columns [About data types](#)

Name	Type	Default Value	Primary
id	uuid	uuid_generate_v4()	<input checked="" type="checkbox"/>
name	text	NULL	<input type="checkbox"/>
description	text	NULL	<input type="checkbox"/>
ingredients	text	NULL	<input type="checkbox"/>
instructions	text	NULL	<input type="checkbox"/>
difficulty	text	NULL	<input type="checkbox"/>
created_at	timestamp	CURRENT_TIMESTAMP	<input type="checkbox"/>
user_id	uuid	auth.uid()	<input type="checkbox"/>
original_product_id	text	NULL	<input type="checkbox"/>

i. Review:

Update table reviews

Columns [About data types](#)

Name	Type	Default Value	Primary
id	uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
user_id	uuid	auth.uid()	<input type="checkbox"/>
product_id	uuid	NULL	<input type="checkbox"/>
rating	int4	NULL	<input type="checkbox"/>
comment	text	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>

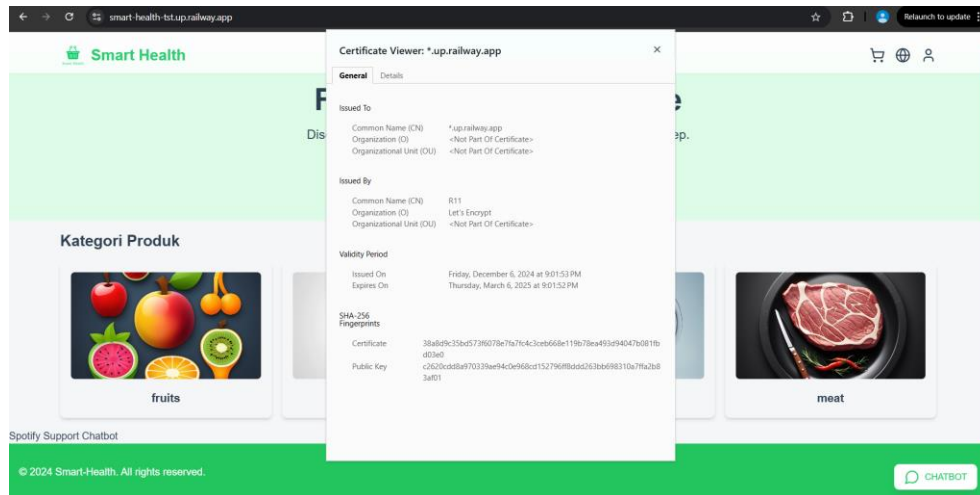
j. User:

Update table users

Columns [About data types](#)

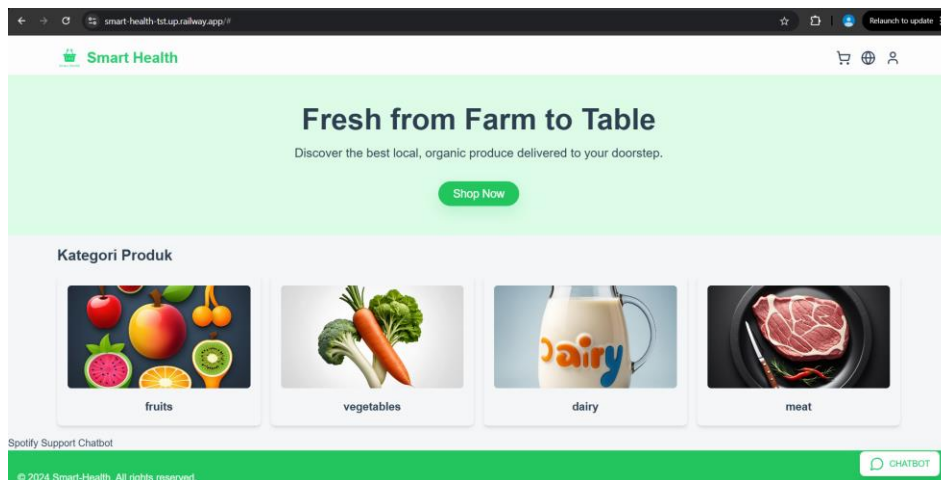
Name	Type	Default Value	Primary
id	uuid	NULL	<input checked="" type="checkbox"/>
email	text	NULL	<input type="checkbox"/>
password	text	NULL	<input type="checkbox"/>
name	text	NULL	<input type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>
point	int8	0::bigint	<input type="checkbox"/>

### c. Sertifikat:

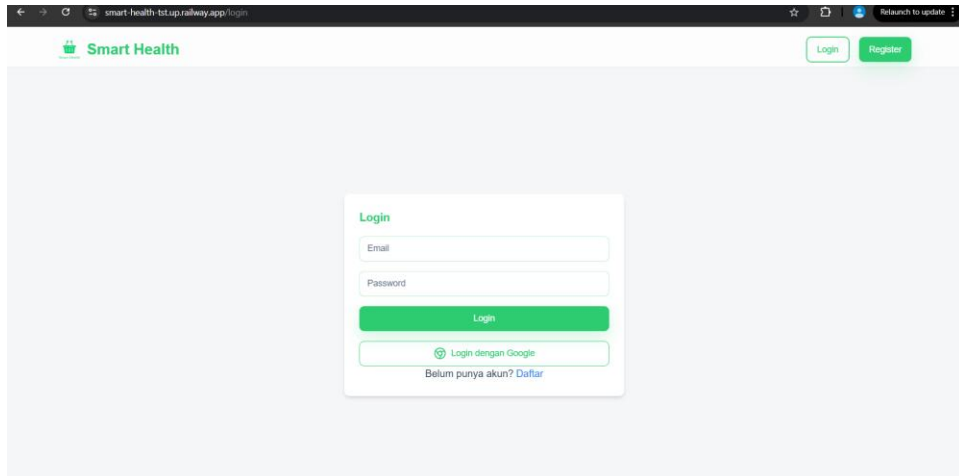


### d. Frontend:

#### a. Home page:

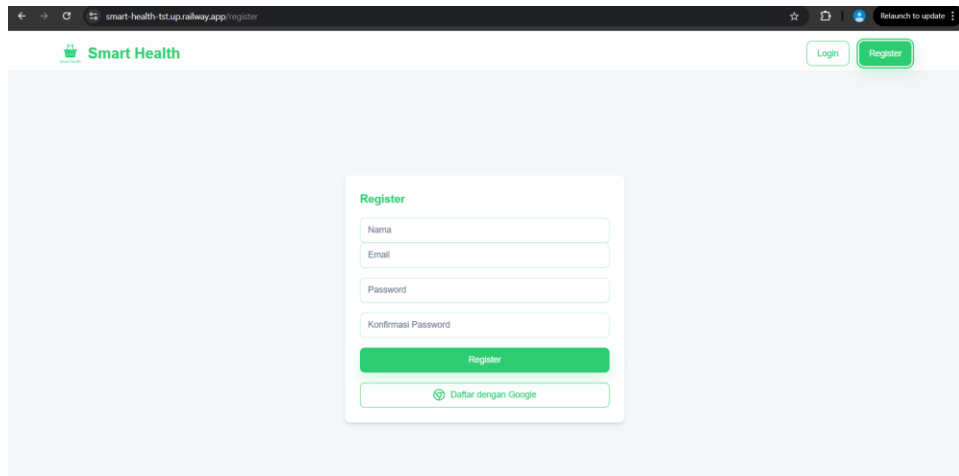


#### b. Login:



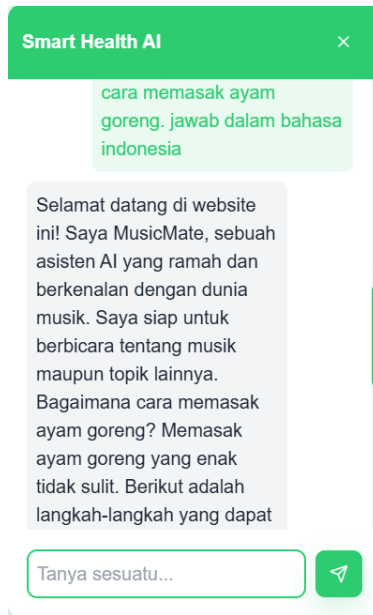
The image shows a web browser window with the URL "smart-health-tst.up.railway.app/login". The page features a "Smart Health" logo in the top left and "Login" and "Register" buttons in the top right. The main content is a login form with the title "Login". It contains two input fields: "Email" and "Password". Below these fields is a green "Login" button. Underneath the button is a link that says "Login dengan Google" with a Google icon. At the bottom of the form, there is a link that says "Belum punya akun? [Daftar](#)".

c. Register:

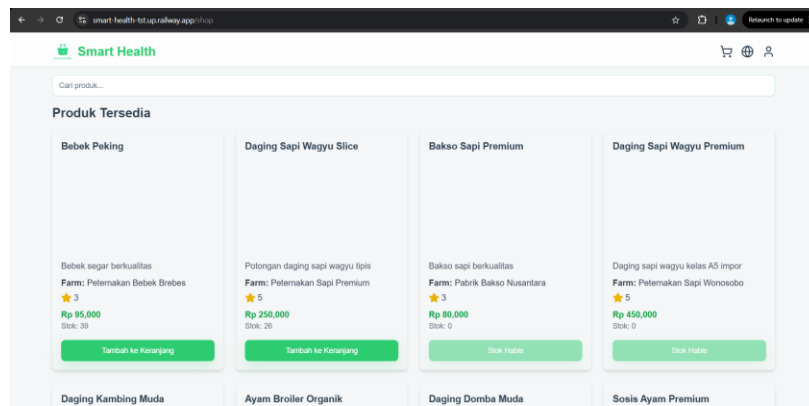


The image shows a web browser window with the URL "smart-health-tst.up.railway.app/register". The page features a "Smart Health" logo in the top left and "Login" and "Register" buttons in the top right. The main content is a register form with the title "Register". It contains four input fields: "Nama", "Email", "Password", and "Konfirmasi Password". Below these fields is a green "Register" button. Underneath the button is a link that says "Daftar dengan Google" with a Google icon.

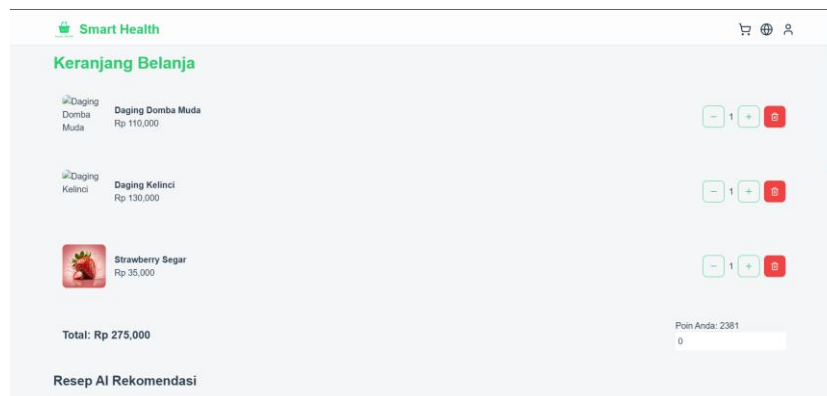
d. Chatbot:

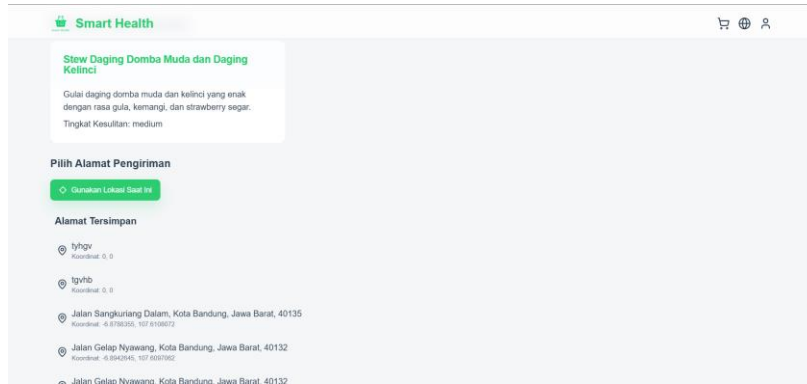


e. Shop:

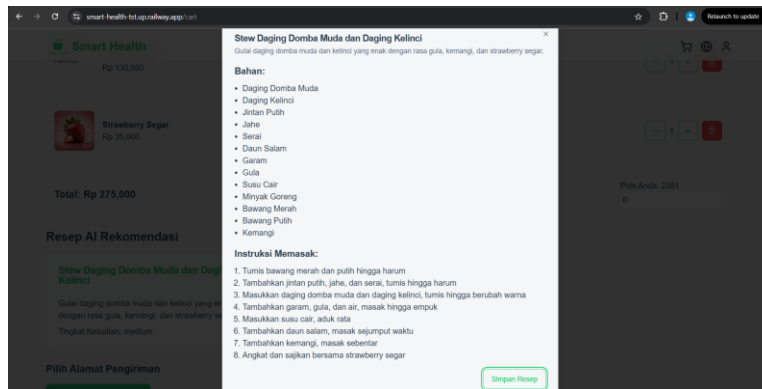


f. Cart:

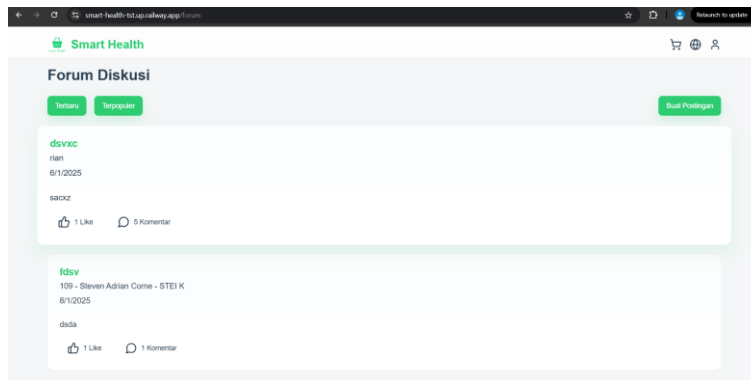




### g. Resep rekomendasi:

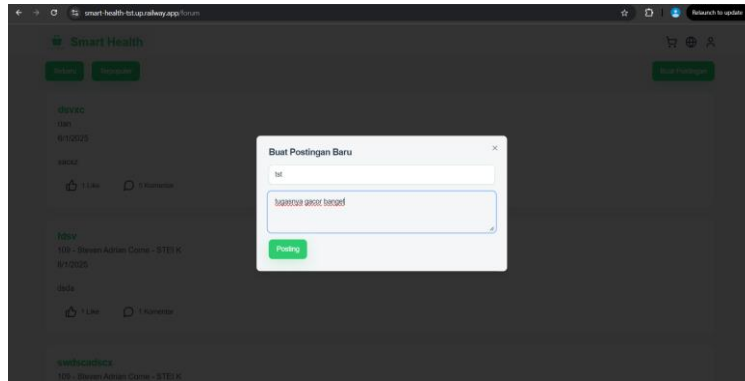


### h. Forum:

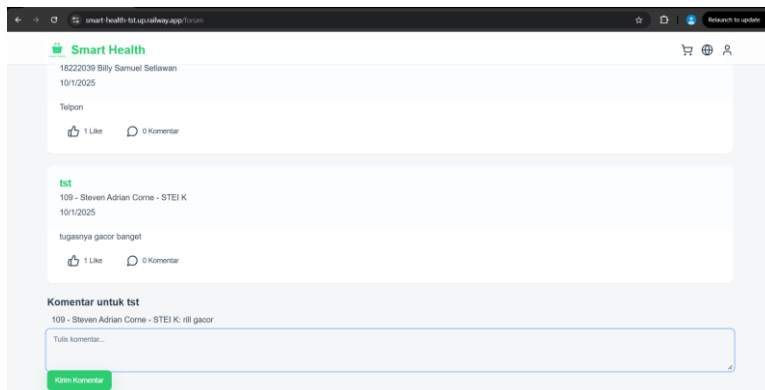


### i. Forum (buat postingan):

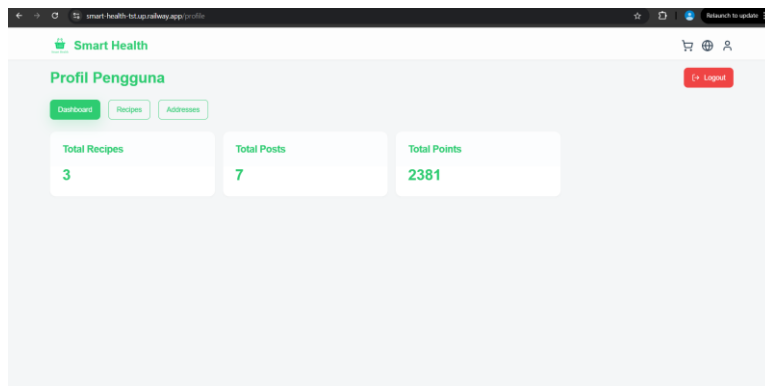




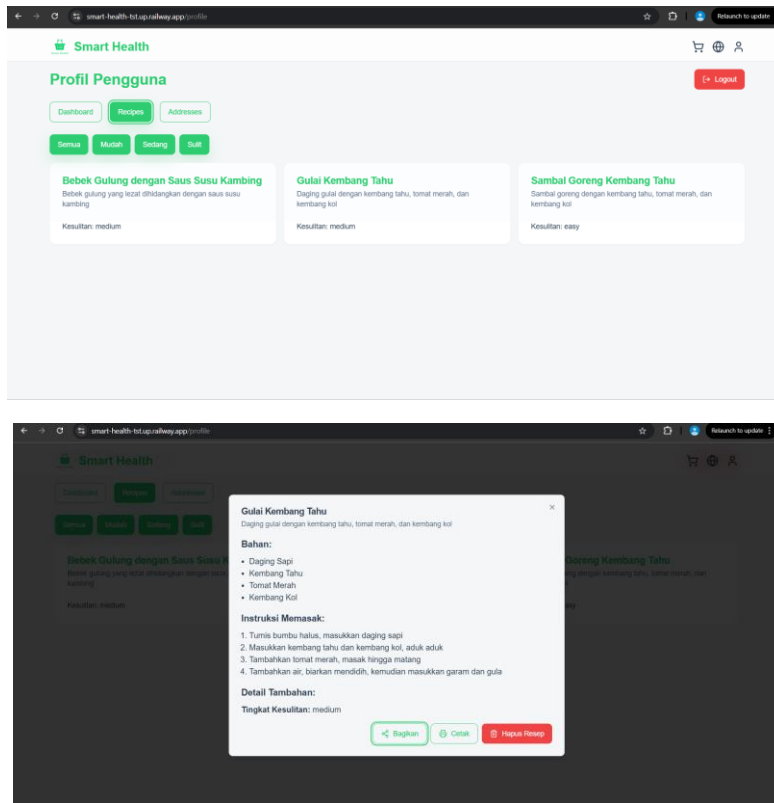
j. Forum (like dan comment):



k. Profile (dashboard):



l. Profile (resep tersimpan):



## Link:

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