

**Tugas Besar II3160 Teknologi Sistem Terintegrasi  
Health Based Dietary Catering**



**Disusun Oleh:**

**Harry Truman Suhalim**

**18222081**

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

## Daftar Isi

|      |                               |    |
|------|-------------------------------|----|
| 1.   | Deskripsi Sistem .....        | 3  |
| 2.   | Link Penting.....             | 3  |
| 3.   | Business Capability.....      | 3  |
| 3.1. | BCM.....                      | 3  |
| 4.   | Subdomain .....               | 4  |
| 4.1. | Domain yang dipilih .....     | 4  |
| 4.2. | Subdomain .....               | 4  |
| 5.   | Proposed Architecture.....    | 5  |
| 5.1. | Sequence Diagram .....        | 5  |
| 5.2. | Teknologi yang digunakan..... | 5  |
| 5.3. | Layered Architecture .....    | 6  |
| 6.   | Implementation Plan .....     | 7  |
| 6.1. | Development Environment ..... | 7  |
| 6.2. | Timeline .....                | 9  |
| 7.   | Implementation .....          | 11 |
| 7.1. | <i>Authentication</i> .....   | 11 |
| 7.2. | <i>Database</i> .....         | 11 |
| 7.3. | <i>Backend</i> .....          | 13 |
| 7.4. | <i>Frontend</i> .....         | 14 |
| 7.5. | Integrasi API .....           | 20 |
| 7.6. | <i>Docker</i> .....           | 22 |

# 1. Deskripsi Sistem

*Health Based Dietary Catering* adalah sistem *catering* berbasis teknologi yang mengintegrasikan kecerdasan buatan (AI) untuk memberikan rekomendasi dan perencanaan menu makanan yang dipersonalisasi sesuai dengan kondisi kesehatan pelanggan. Sistem ini memiliki tiga komponen utama: *AI-Driven Dietary Recommendation* untuk memberikan saran makanan personal dan menganalisis nilai gizi, *Menu Customization Engine* untuk menghasilkan menu dinamis berdasarkan rekomendasi AI, dan *Health Outcome Tracking* untuk memantau serta melaporkan kemajuan kesehatan pelanggan.

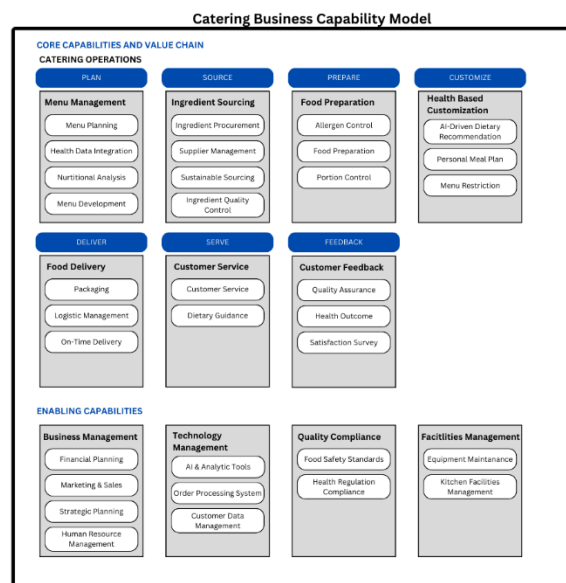
Dari sisi teknis, sistem dibangun dengan arsitektur berlapis menggunakan FastAPI untuk *backend*, MongoDB sebagai *database*, dan HTML/CSS/JavaScript untuk *frontend*, serta memanfaatkan Groq AI API untuk memberikan rekomendasi diet yang akurat. Sistem menyediakan fitur-fitur seperti autentikasi pengguna menggunakan Auth0, profil pengguna untuk menyimpan data kesehatan, *dashboard* pemantauan perkembangan diet, halaman resep dengan rekomendasi AI, serta *chatbot* untuk membantu pengguna, yang semuanya diimplementasikan menggunakan *Docker* untuk memastikan *deployment* yang efisien dan handal.

## 2. Link Penting

- Website : <https://ii3160-production.up.railway.app/>
- Github : <https://github.com/Harryts1/II3160>

## 3. Business Capability

### 3.1. BCM



Ini adalah BCM untuk sebuah bisnis *catering* yang menggunakan teknologi AI dalam membantu pembuatan menu dan rekomendasi makanan diet yang menyesuaikan dengan kondisi fisik dan kesehatan dari *customer*.

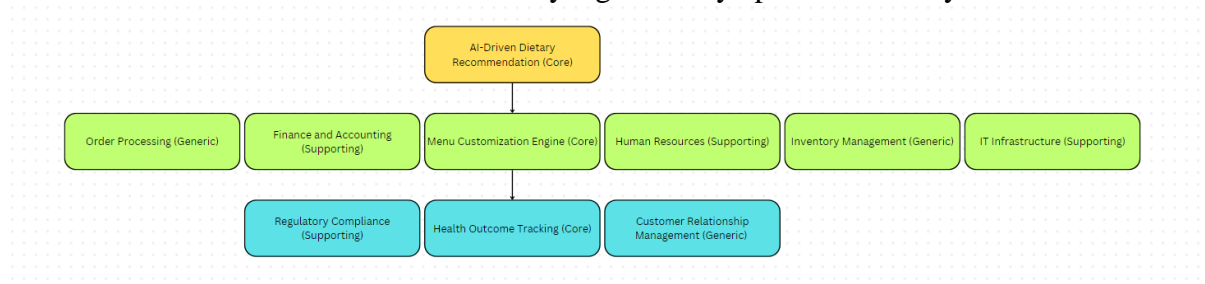
## 4. Subdomain

### 4.1. Domain yang dipilih

Dari BCM yang telah dibuat, saya memilih core berupa **AI-Driven Dietary Recommendation** karena menurut saya domain ini sifatnya paling unik dibanding domain yang lainnya.

### 4.2. Subdomain

Berikut ini adalah subdomain dari domain yang telah saya pilih sebelumnya.



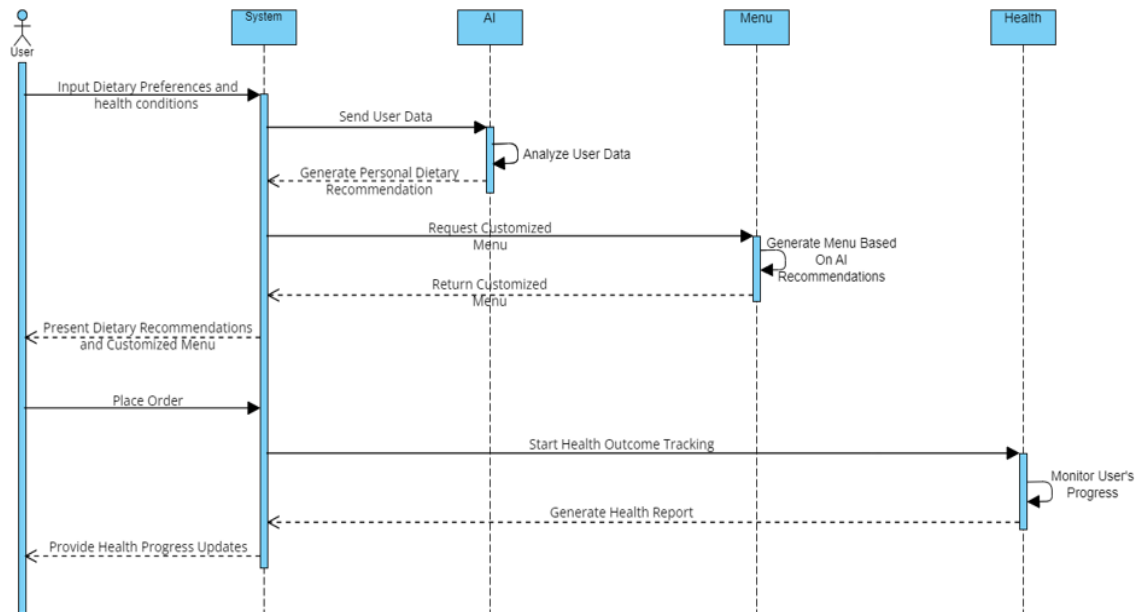
Fungsi :

- Core
  - a. *AI-Driven Dietary Recommendation* : Memberikan saran makanan personal menggunakan teknologi AI, Menganalisis dan mengoptimalkan nilai gizi makanan, serta memberikan rekomendasi makanan untuk mengatasi kondisi medis.
  - b. *Menu Customization Engine* : Menghasilkan menu secara dinamis berdasarkan rekomendasi AI.
  - c. *Health Outcome Tracking* : Memantau dan melaporkan kemajuan kesehatan pelanggan.
- Generic
  - a. *Order Processing* : Mengelola penerimaan dan manajemen pesanan.
  - b. *Inventory Management* : Melacak stok bahan makanan.
  - c. *Customer Relationship Management* : Mengelola sistem umpan balik dan ulasan pelanggan.
- Supporting
  - a. *Finance and Accounting* : Membuat anggaran dan perkiraan keuangan.
  - b. *Human Resources* : Mengelola staf.
  - c. *IT Infrastructure* : Menjaga keamanan dan privasi data.
  - d. *Regulatory Compliance* : Memastikan kepatuhan terhadap standar keamanan pangan.

## 5. Proposed Architecture

### 5.1. Sequence Diagram

Berikut ini adalah sequence diagram untuk domain yang telah dipilih



### 5.2. Teknologi yang digunakan

#### 1. Backend:

- FastAPI (Python Framework) : FastAPI dibangun di atas Starlette dan Pydantic, memberikan performa yang sangat cepat, mendukung asynchronous programming secara native, validasi otomatis menggunakan Pydantic, *generate* dokumentasi API secara otomatis (Swagger UI & ReDoc), serta banyak kelebihan lainnya jika dibanding dengan teknologi *backend* lainnya.

#### 2. Middleware dan Utilitas:

- Body-parser: Middleware untuk parsing body request HTTP.
- CORS (Cross-Origin Resource Sharing): Digunakan untuk menangani permintaan lintas domain.

#### 3. Database:

- MongoDB : Ini cocok dengan stack MERN dan ideal untuk aplikasi web modern karena fleksibilitasnya.

#### 4. Frontend :

- HTML, CSS, Java Script : untuk mendesain web interaktif dan menarik.

#### 5. API:

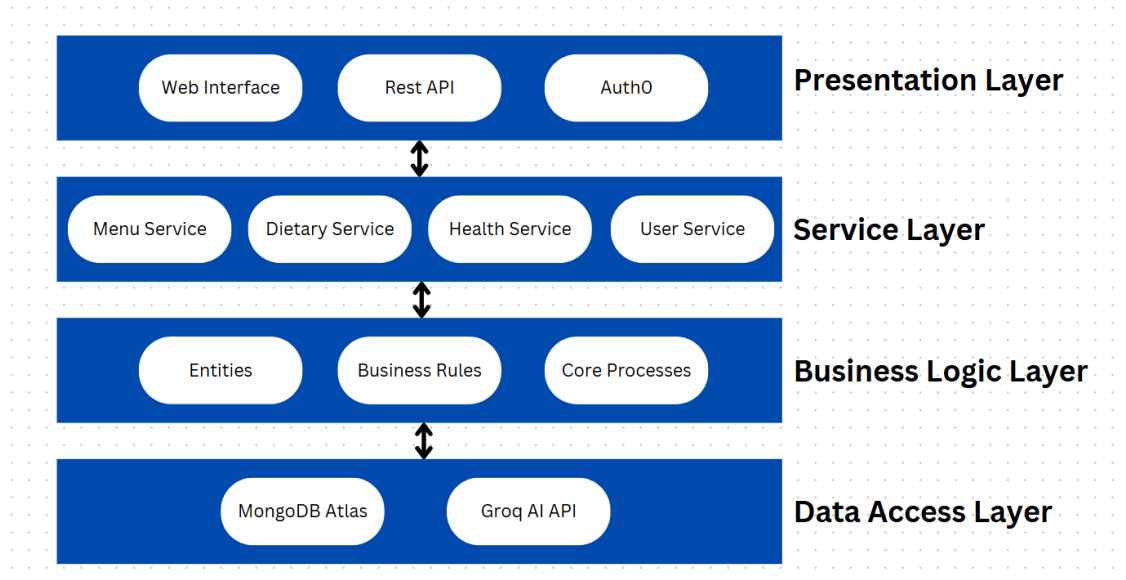
- **RESTful API:** Server menyediakan endpoint-endpoint REST seperti /api/register, /api/login, dan /api/recommendation.
- **Groq AI API:** Untuk teknologi AI yang digunakan, di luar memberikan fitur yang gratis, Groq AI dapat memberikan hasil yang memuaskan dalam waktu yang tergolong cepat.

#### 6. Authentication :

- **Auth0 :** open source dan terpercaya

### 5.3. Layered Architecture

Berikut ini adalah gambaran *layered architecture* dari aplikasi ini.



#### • **Presentation Layer (UI)**

- **Web Interface:** Antarmuka pengguna berbasis HTML, CSS, Java Script
- **REST API:** *Endpoints* untuk komunikasi *client-server*
- **Auth0 Interface:** Manajemen autentikasi dan otorisasi

#### • **Service Layer (Application)**

- **Menu Service:** Mengelola pembuatan dan kustomisasi menu
- **Dietary Service:** Mengintegrasikan AI untuk rekomendasi diet
- **User Service:** Manajemen pengguna dan profil
- **Health Service:** *Tracking* dan analisis kesehatan pelanggan

#### • **Business Logic Layer (Domain)**

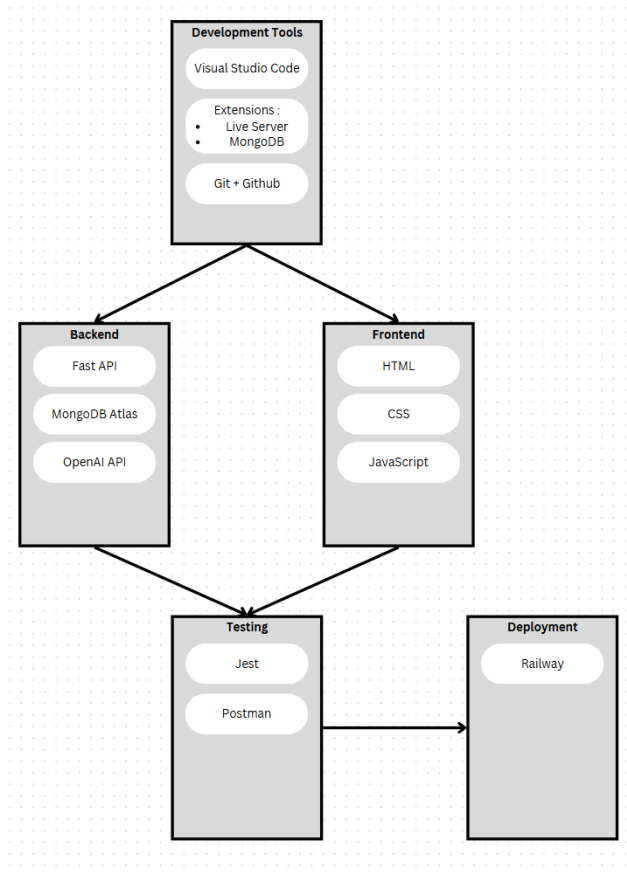
- **Entities:**

- *User*: Data pelanggan dan preferensi
- *Menu*: *Item menu* dan kategori
- *Order*: Pesanan dan statusnya
- *Diet Plan*: Rencana diet personal
- *Health Profile*: Profil kesehatan pelanggan
- *Business Rules*:
  - *Diet Restrictions*: Batasan makanan berdasarkan kondisi medis
  - *Nutritional Guidelines*: Standar gizi
  - *Pricing Rules*: Kalkulasi harga
  - *Health Metrics*: Parameter pengukuran kesehatan
- *Core Processes*:
  - *Menu Generation*: Algoritma pembuatan menu
  - *Diet Analysis*: Analisis kebutuhan nutrisi
  - *Order Fulfillment*: Proses pemenuhan pesanan
  - *Health Monitoring*: Pemantauan kesehatan
- *Data Access Layer (Infrastructure)*
- MongoDB Atlas: *Database* utama untuk menyimpan data
- Groq AI API: Integrasi AI untuk rekomendasi

## 6. Implementation Plan

### 6.1. Development Environment

Berdasarkan teknologi tersebut, berikut ini adalah *development environment* yang akan diterapkan.



- *Development Tools:*

- Visual Studio Code: Penggunaan Visual Studio Code sebagai IDE karena memiliki banyak *extensions* yang dapat membantu membuat *website*
- *Extensions:*
  - *Live Server:* Memungkinkan *auto-reload* saat *development*
  - *MongoDB:* Ekstensi untuk memudahkan interaksi dengan *database* MongoDB
- Git + Github: Sistem *version control* dan *platform hosting repository*

- *Backend:*

- FastAPI (Python *Framework*) : Layanan *backend* yang mudah untuk digunakan dan dapat melihat dokumentasi langsung pada */docs*
- MongoDB Atlas: Layanan *cloud database* MongoDB yang terkelola
- OpenAI API: API untuk mengintegrasikan kemampuan AI/ML dari OpenAI

- *Frontend:*

- HTML: Bahasa *markup* untuk struktur halaman *website*
- CSS: Bahasa *styling* untuk desain visual *website*
- JavaScript: Bahasa pemrograman untuk membuat *website* interaktif

- *Testing:*

- Jest: *Framework testing* JavaScript yang fokus pada kesederhanaan



- Postman: *Tool* untuk *testing* API dan dokumentasi
  - *Deployment*:
- Railway: *Platform hosting* yang juga mampu dihubungkan dengan *docker*

## 6.2. Timeline

Berikut ini adalah alur pengembangan untuk *website* yang telah direncanakan.



- ***Setup & Foundation:***

Tahap ini adalah dasar dari keseluruhan proyek, termasuk menyiapkan lingkungan pengembangan, struktur proyek, dan versi kontrol. Melakukan ini terlebih dahulu memungkinkan untuk bekerja di lingkungan yang lebih terstruktur.

- ***Backend Development:***

Backend menjadi prioritas berikutnya karena API dan layanan dasar perlu dibangun terlebih dahulu sebelum *front-end* dapat diintegrasikan. Tahap ini mencakup *endpoint* inti, model data, dan autentikasi, yang semuanya mendasari fitur aplikasi.

- ***Frontend Development:***

Setelah backend selesai dibuat, *front-end* dapat dikembangkan dengan mengintegrasikan API dan layanan yang sudah ada. Ini memungkinkan untuk membangun UI sesuai dengan logika *backend*.

- ***Testing & Integration:***

Tahap ini diperlukan untuk memastikan semua komponen (*backend* dan *frontend*) berfungsi dengan baik secara terintegrasi. Tes dan optimasi dilakukan untuk mendeteksi dan memperbaiki masalah sebelum peluncuran.

- ***Deployment & Launch:***

Tahap terakhir adalah deployment, di mana aplikasi siap diluncurkan dan digunakan oleh pengguna. Pada tahap ini, infrastruktur produksi disiapkan, dan sistem dipantau untuk memastikan performa yang optimal.

Berikut ini adalah *timeline* yang direncanakan untuk membuat *software* ini.

| November 2024 - Desember 2024 |        |        |        |        |        |        |        |        |  |
|-------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--|
| Task                          | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 |  |
| Setup & Foundation            |        |        |        |        |        |        |        |        |  |
| Backend Development           |        |        |        |        |        |        |        |        |  |
| Frontend Development          |        |        |        |        |        |        |        |        |  |
| Testing & Integration         |        |        |        |        |        |        |        |        |  |
| Deployment & Launch           |        |        |        |        |        |        |        |        |  |

### Sprint 1: Setup & Foundation

**Durasi:** 4 November 2024 - 17 November 2024

**Aktivitas:**

- Mengatur lingkungan pengembangan
- Mendesain skema basis data
- Menyusun struktur proyek dan mengonfigurasi git

### Sprint 2: Backend Development

**Durasi:** 18 November 2024 - 8 Desember 2024

**Aktivitas:**

- Mengembangkan endpoint API inti
- Integrasi dengan Groq AI API
- Menyiapkan autentikasi dengan Auth0
- Mengimplementasikan model data dan penanganan kesalahan

### Sprint 3: Frontend Development

**Durasi:** 25 November 2024 - 15 Desember 2024

**Aktivitas:**

- Membuat komponen UI responsif
- Mengimplementasikan alur autentikasi
- Membangun antarmuka rekomendasi
- Menambahkan fitur kustomisasi menu dan dashboard pelacakan kesehatan

### Sprint 4: Testing & Integration

**Durasi:** 9 Desember 2024 - 20 Desember 2024

**Aktivitas:**

- Menulis dan menjalankan pengujian unit

- Melakukan pengujian integrasi dan keamanan
- Menyelesaikan bug, mengoptimalkan performa
- Mendokumentasikan endpoint API

## Sprint 5: Deployment & Launch

**Durasi:** 13 Desember 2024 - 20 Desember 2024

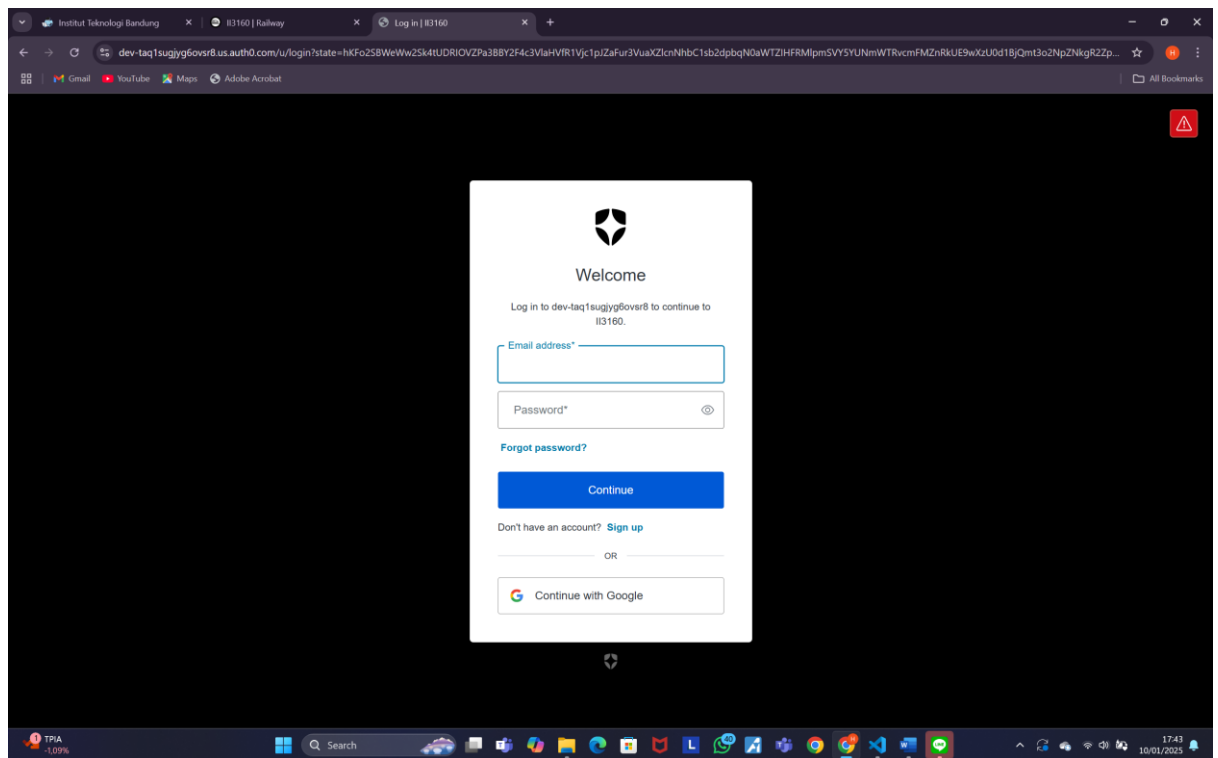
**Aktivitas:**

- Menghubungkan proyek ke Railway dan mengatur variabel lingkungan
- Menguji deployment pratinjau dan memastikan semua fungsi berjalan
- Menerapkan ke produksi di Railway dan memantau performa

## 7. Implementation

### 7.1. Authentication

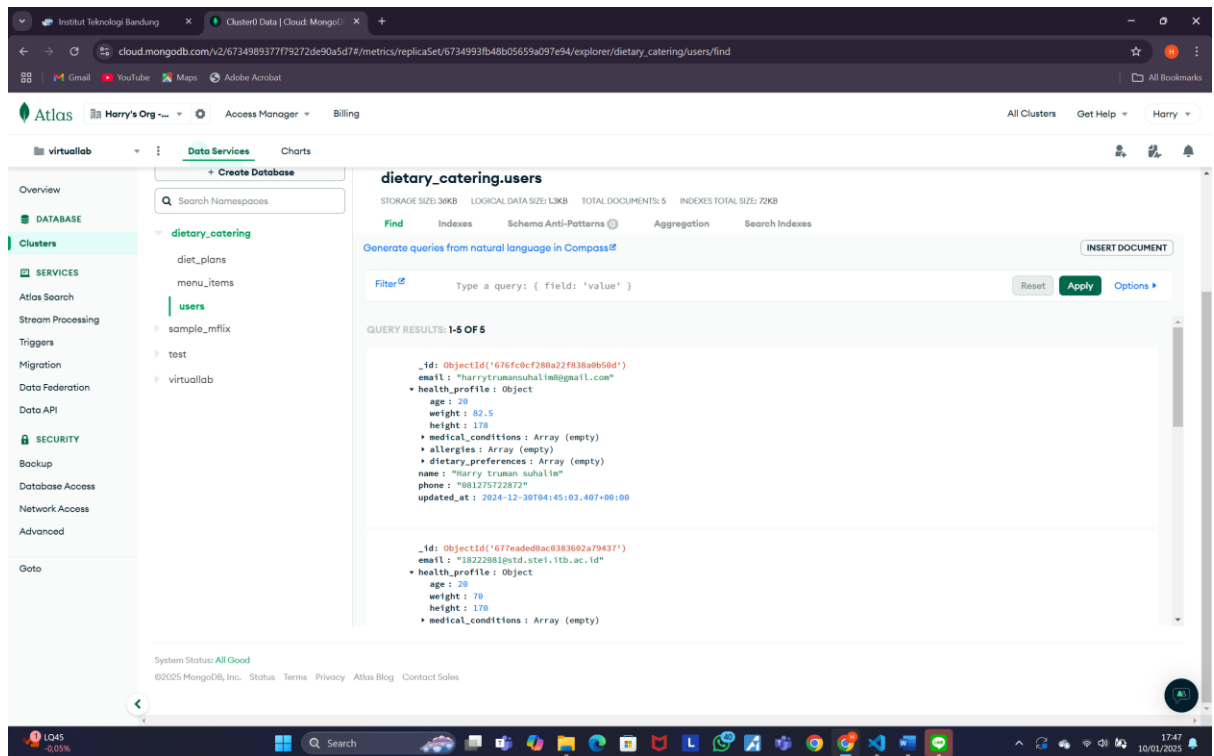
*Authentication* dibuat menggunakan Auth0 dan dideploy dengan Railway.



### 7.2. Database

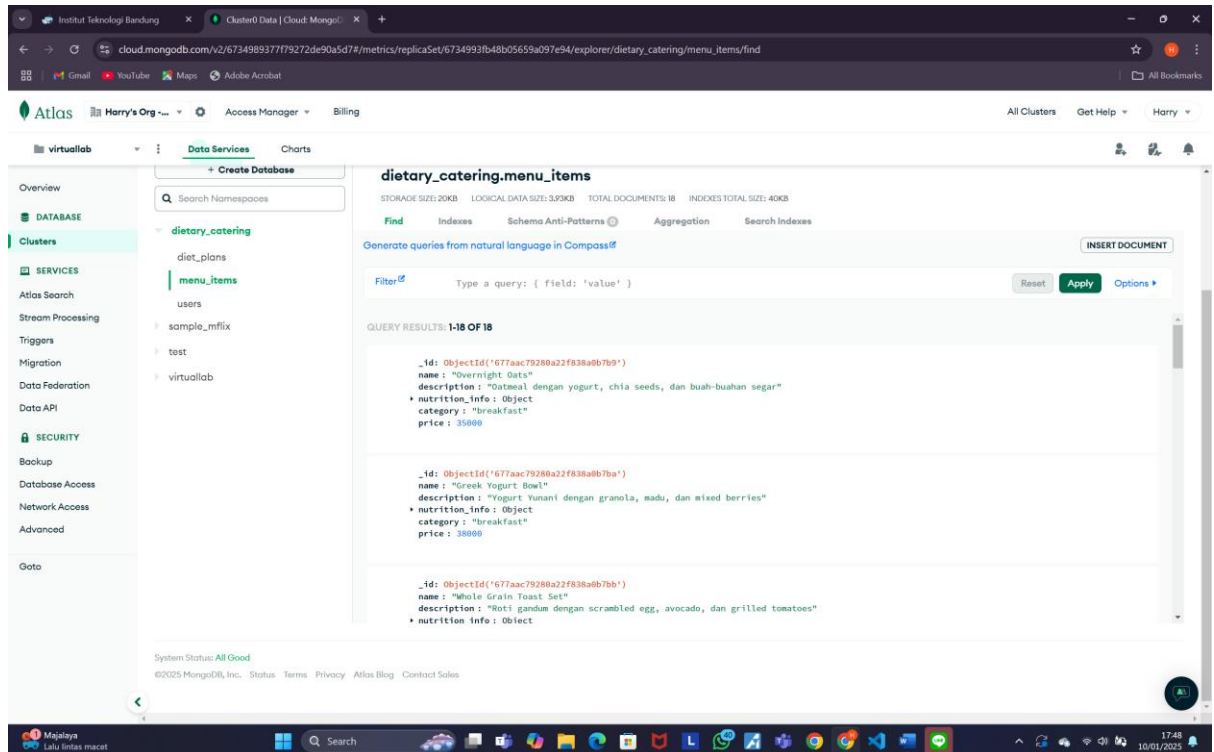
Berikut ini adalah beberapa hal yang disimpan dalam database.

- Data User  
Data User dicatat untuk kebutuhan *generate menu*.



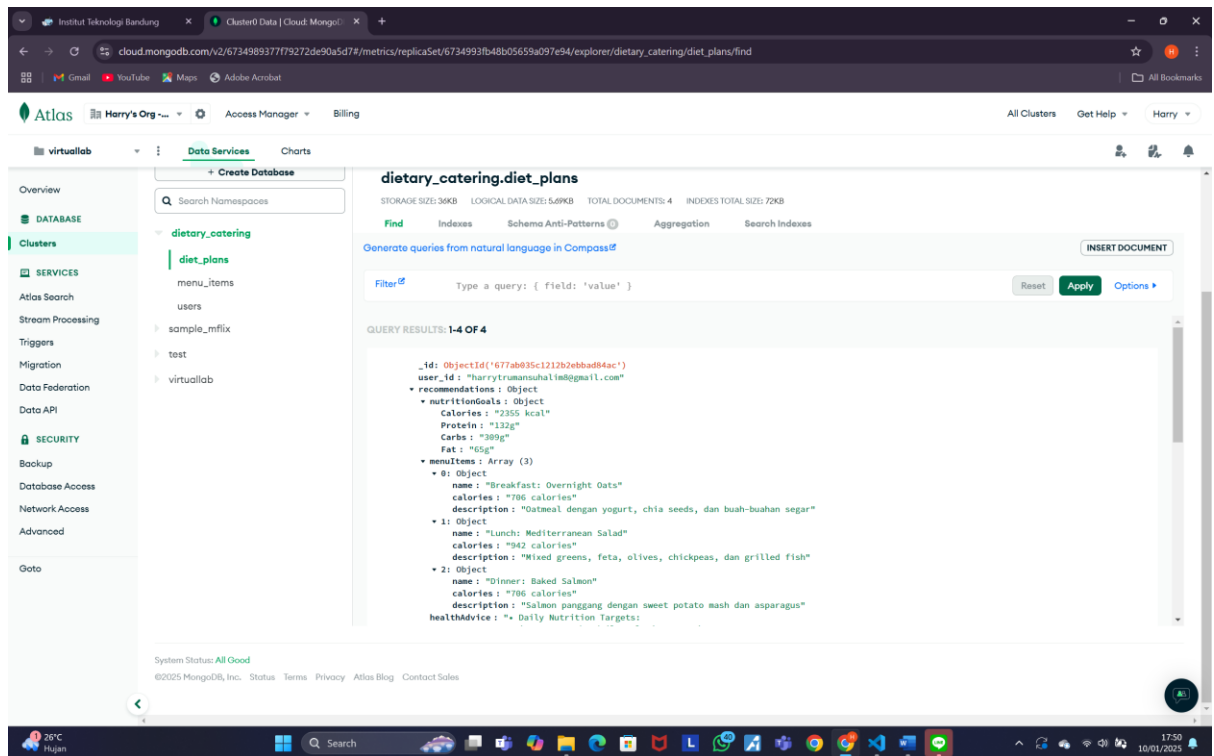
#### - Data Menu

Data Menu berisikan berbagai menu yang tersedia untuk direkomendasikan berdasarkan kondisi Kesehatan dari *user*.



#### - Data diet\_plans

Diet plans berisi data terkait hasil *generate* AI yang terakhir dilakukan oleh *user*.



### 7.3. Backend

Berikut ini adlaha beberapa API *endpoints* yang digunakan dalam *backend*.

## Health Based Dietary Catering API 1.0.0 QAS 3.1

/openapi.json

API for managing dietary plans with Auth0 authentication

### default ^

|      |                       |                     |   |
|------|-----------------------|---------------------|---|
| GET  | /health               | Health Check        | ▼ |
| GET  | /                     | Serve Home          | ▼ |
| GET  | /login                | Login               | ▼ |
| GET  | /callback             | Callback            | ▼ |
| POST | /update-profile       | Update Profile      | ▼ |
| GET  | /dashboard            | Dashboard           | ▼ |
| GET  | /logout               | Logout              | ▼ |
| GET  | /complete-profile     | Complete Profile    | ▼ |
| GET  | /menu-items           | Get Menu Items      | ▼ |
| POST | /menu-items           | Create Menu Item    | ▼ |
| POST | /diet-plans           | Create Diet Plan    | ▼ |
| GET  | /diet-plans/{user_id} | Get User Diet Plans | ▼ |
| POST | /recommendations      | Get Recommendations | ▼ |

### users ^

|      |        |             |   |
|------|--------|-------------|---|
| GET  | /users | Get Users   | ▼ |
| POST | /users | Create User | ▼ |

### Schemas ^

DietPlan > Expand all object

HTTPValidationError > Expand all object

MenuItem > Expand all object

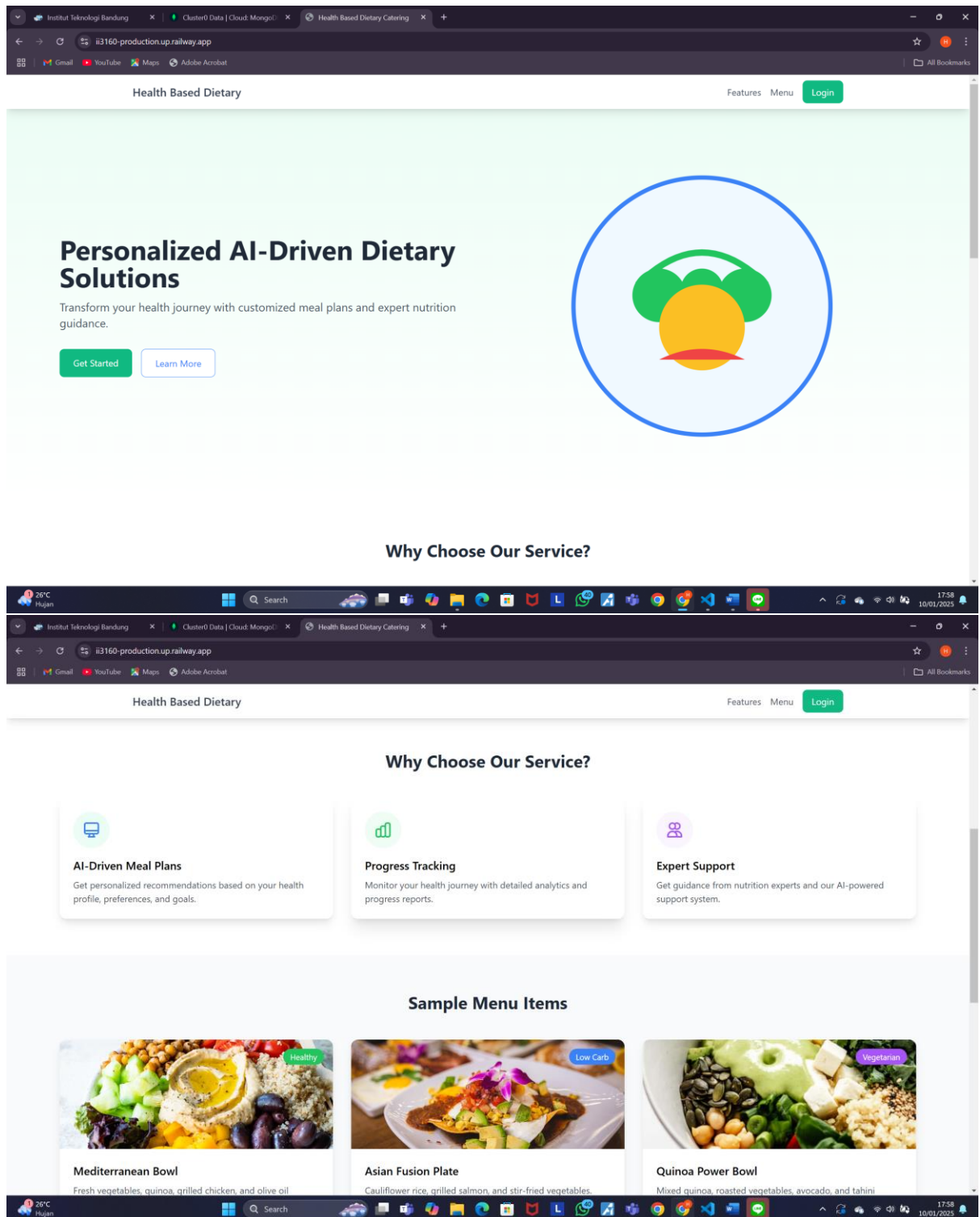
User > Expand all object

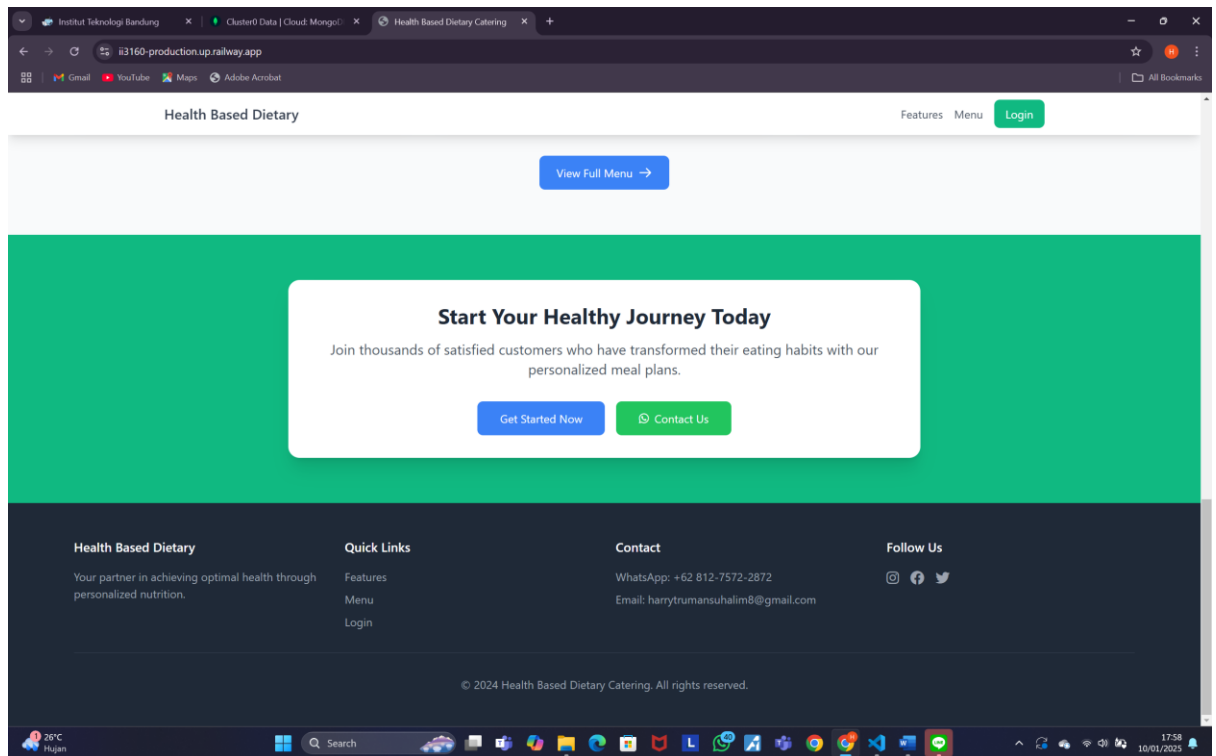
ValidationError > Expand all object

## 7.4. Frontend

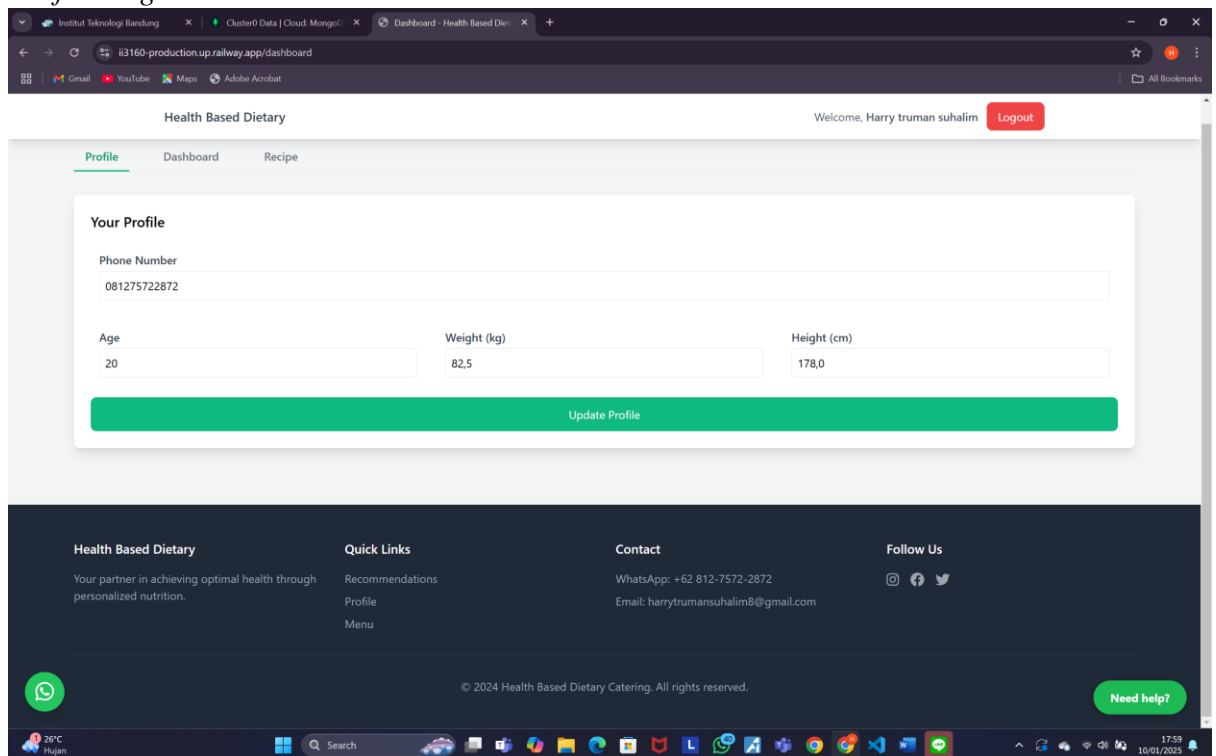
Berikut ini adalah beberapa tampilan pada *frontend* yang dibuat dengan HTML, CSS, dan Java Script

- *Hero Page*



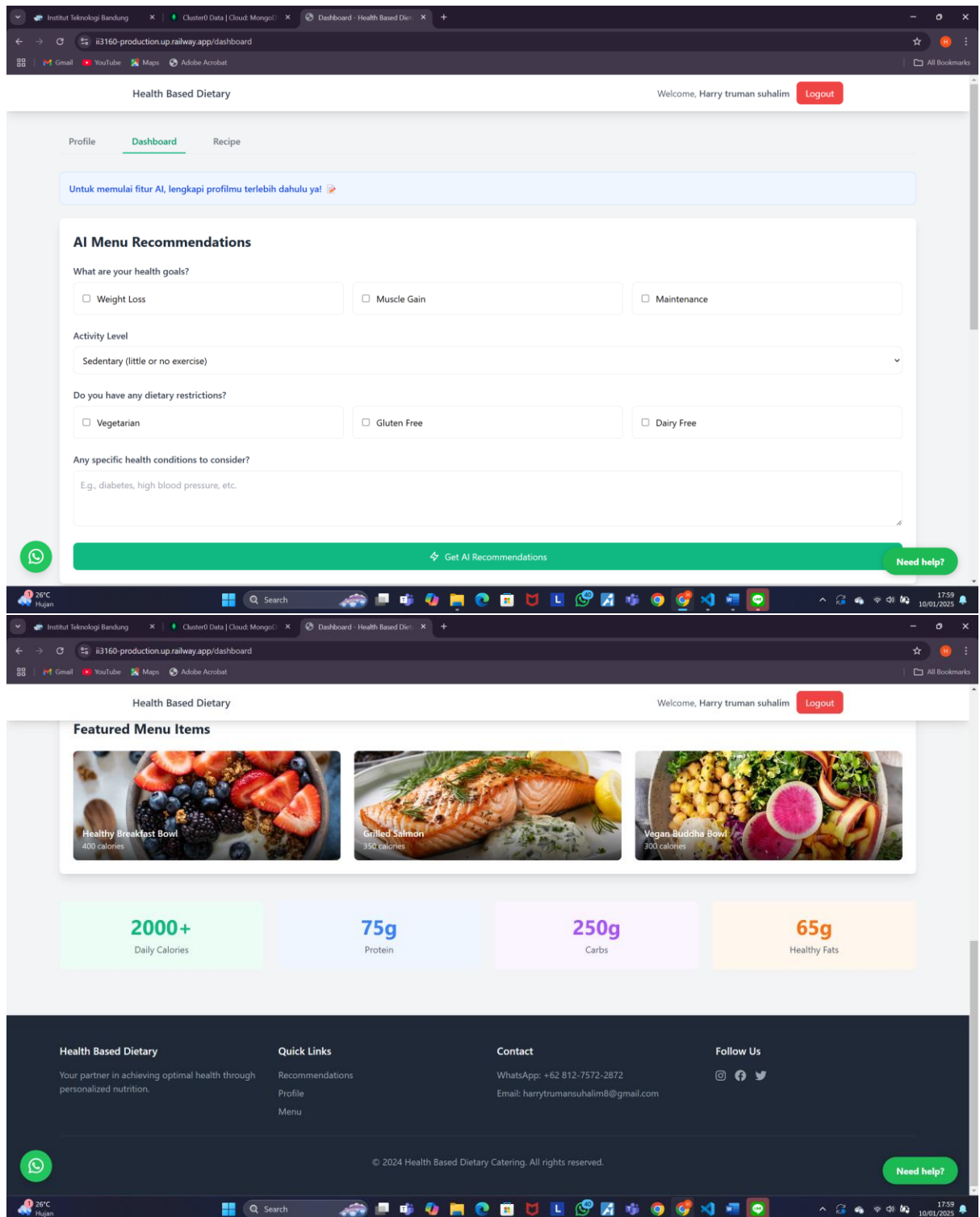


- *Profile Page*

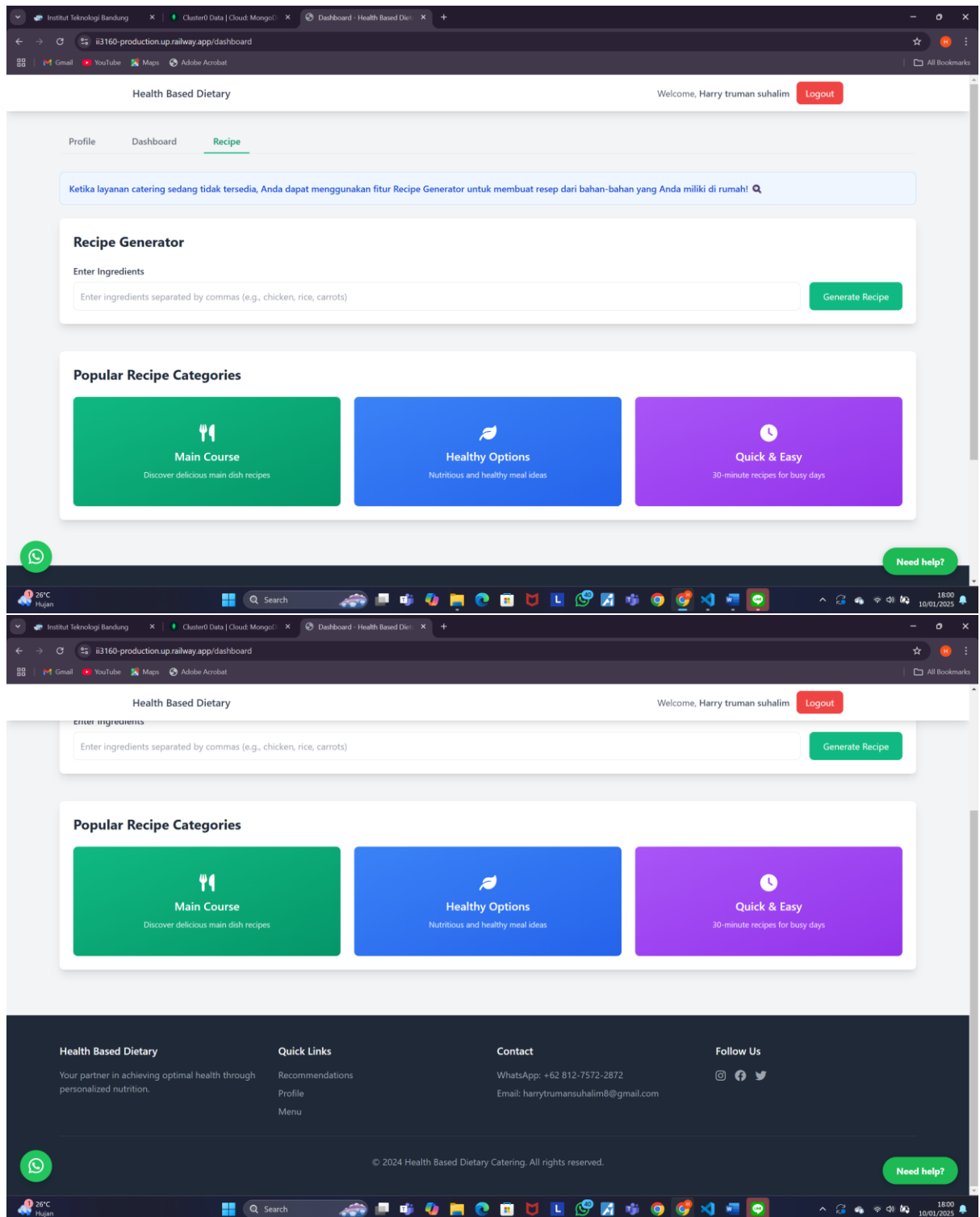


- *Dashboard Page*

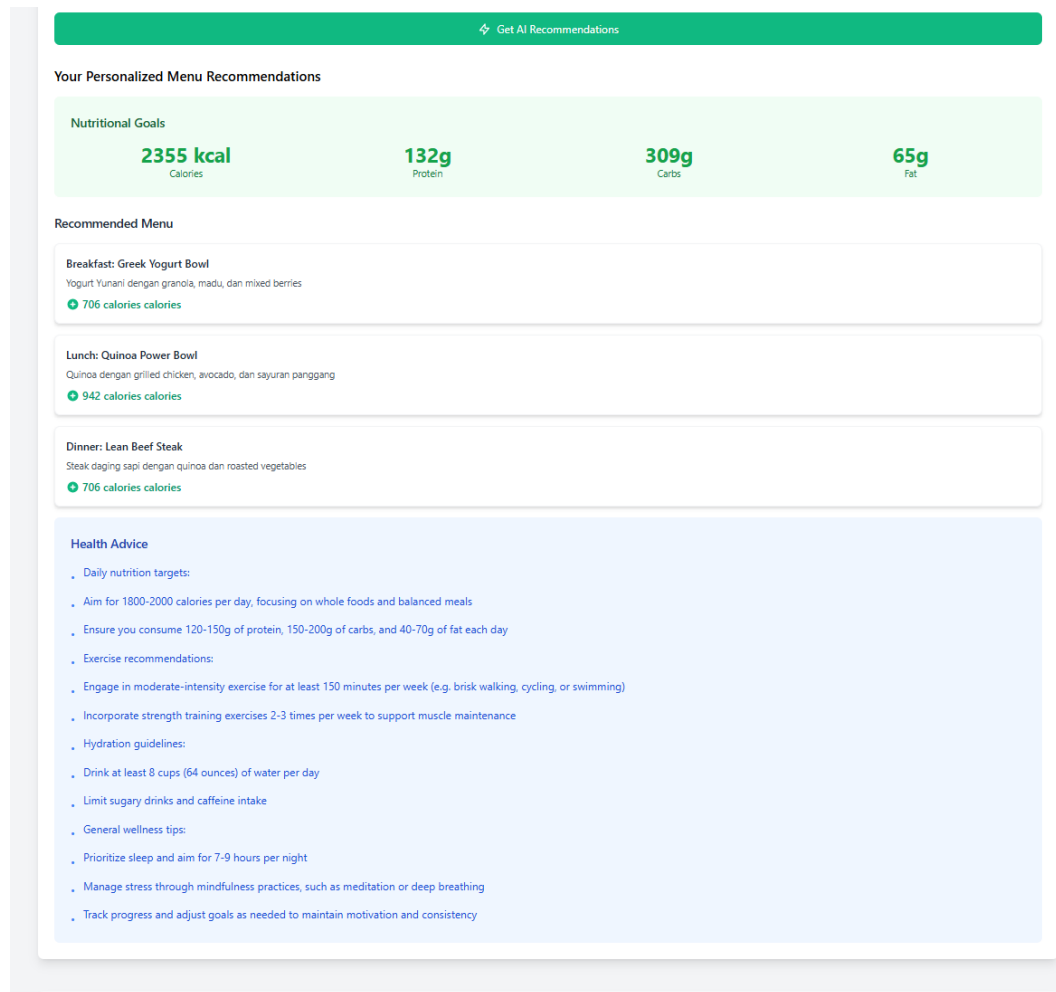




- Recipe Page



- Hasil AI



- Hasil Resep (Integrasi API)

### Recipe Generator

Enter Ingredients

ayam

Generate Recipe

#### Sate Ayam

Sate ayam goreng dengan bumbu ketumbar

##### Ingredients:

- Ayam
- Cabai merah
- Bawang Putih
- Ketumbar

##### Instructions:

1. Haluskan cabai merah, bawang putih, dan ketumbar
2. Celup ayam ke bumbu, lalu satue

#### Ayam Bakar BBQ

Ayam bakar dengan rasa BBQ

##### Ingredients:

- Ayam
- Jinten
- Cabe Rawit
- Gula Merah
- Kecap Manis

##### Instructions:

1. Campur jinten, cabe rawit, gula merah, dan kecap manis
2. Marinasi ayam dengan bumbu
3. Bakar ayam

#### Ayam Goreng Crispy

Ayam goreng krispi dan renyah

##### Ingredients:

- Ayam
- Tepung Terigu
- Garam
- Merica

##### Instructions:

1. Campur tepung terigu, garam, dan merica
2. Balur ayam dengan adonan tepung
3. Goreng ayam hingga matang

## 7.5. Integrasi API

Berikut ini adalah Integrasi API yang saya lakukan dengan API teman saya.

- *Generate Resep* (Steven Adrian Corne – 18222101)

## Recipe Generator

Enter Ingredients

ayam

Generate Recipe

### Sate Ayam

Sate ayam goreng dengan bumbu ketumbar

Ingredients:

- Ayam
- Cabai merah
- Bawang Putih
- Ketumbar

Instructions:

1. Haluskan cabai merah, bawang putih, dan ketumbar
2. Celup ayam ke bumbu, lalu satue

### Ayam Bakar BBQ

Ayam bakar dengan rasa BBQ

Ingredients:

- Ayam
- Jinten
- Cabe Rawit
- Gula Merah
- Kecap Manis

Instructions:

1. Campur jinten, cabe rawit, gula merah, dan kecap manis
2. Marinasi ayam dengan bumbu
3. Bakar ayam

### Ayam Goreng Crispy

Ayam goreng krispi dan renyah

Ingredients:

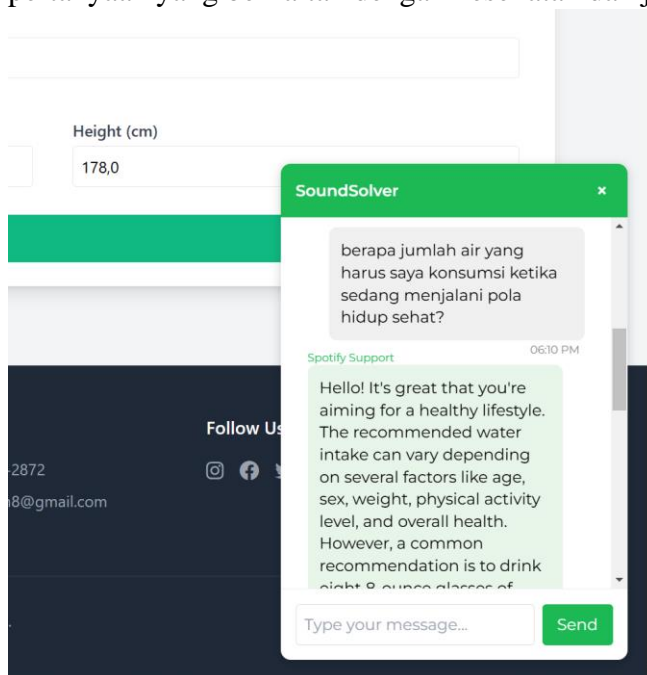
- Ayam
- Tepung Terigu
- Garam
- Merica

Instructions:

1. Campur tepung terigu, garam, dan merica
2. Balur ayam dengan adonan tepung
3. Goreng ayam hingga matang

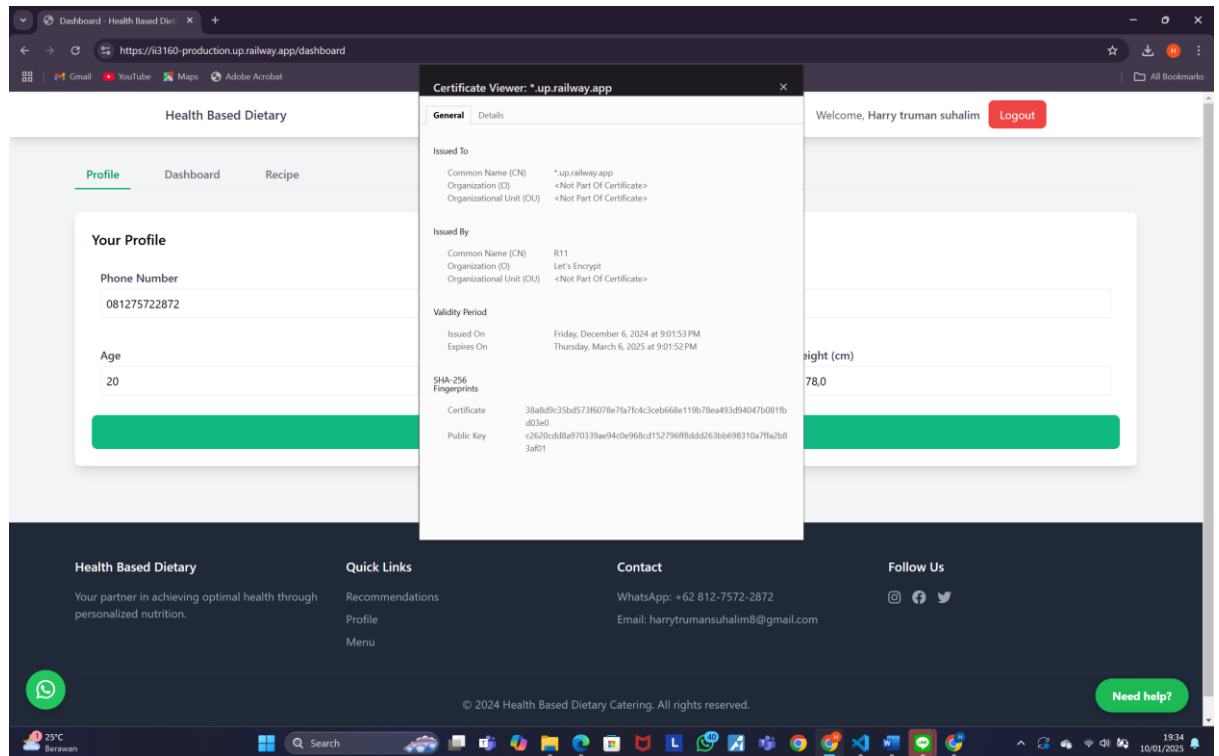
- Chatbot (Jonathan Wiguna – 18222019)

Chatbot awalnya ditujukan untuk musik, tetapi tetap dapat menjawab pertanyaan-pertanyaan yang berkaitan dengan kesehatan dan juga *catering*.



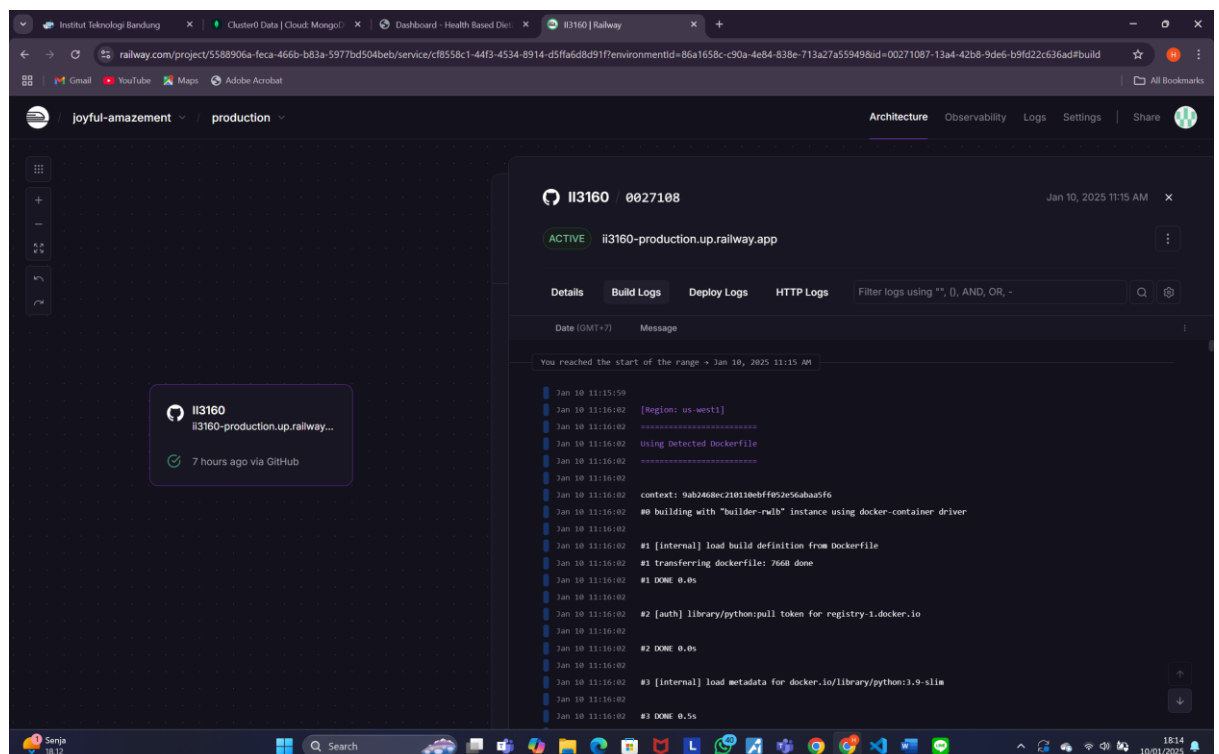
## 7.6. HTTPS

Berikut ini adalah HTTPS dari *website* saya.



## 7.7. Docker

Berikut ini adalah implementasi *docker* yang telah dilakukan dan di *deploy* pada Railway.



joyful-amazement / production

Architecture Observability Logs Settings Share

ii3160 0027108 Jan 10, 2025 11:15 AM

ACTIVE ii3160-production.up.railway.app

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

| Date (GMT+7)    | Message  |
|-----------------|--|
| Jan 10 11:16:29 | #12 exporting layers 2.1s done   |
| Jan 10 11:16:29 | #12 ...  |
| Jan 10 11:16:29 | #13 [auth] sharing credentials for us-west1.registry.rly.net   |
| Jan 10 11:16:29 | #13 DONE 0.0s  |
| Jan 10 11:16:29 | #12 exporting to image   |
| Jan 10 11:16:29 | #12 exporting manifest sha256:3cfc8c640972a53fata02f37aa78be742bdf2d2bde19ee0b9d6a1c64ae1d7b done  |
| Jan 10 11:16:29 | #12 exporting config sha256:da4046fe38ecae3770f8c691680b1a2c7b3bfe5c4c31080ddcde9a9ebc0 done   |
| Jan 10 11:16:29 | #12 exporting attestation manifest sha256:cb0ebc8b18c322b8427129991a4db0b4fbff5692209f60d31b3ab74484fe done  |
| Jan 10 11:16:29 | #12 exporting manifest list sha256:758f83241b8f7d3440da750611067a82ac3553956dfe1767e27f1bdfdee48872 done   |
| Jan 10 11:16:29 | #12 pushing layers   |
| Jan 10 11:16:31 | #12 pushing layers 2.3s done   |
| Jan 10 11:16:31 | #12 pushing manifest for us-west1.registry.rly.net/cf858c1-44f3-4534-8914-d5ffa6dd91f:00271087-13a4-42b8-9de6-b9fd22c36a@sha256:758f83241b8f7d3440da750611067a82ac3553956dfe1767e27f1bdfdee48872           |
| Jan 10 11:16:31 | #12 pushing manifest for us-west1.registry.rly.net/cf858c1-44f3-4534-8914-d5ffa6dd91f:00271087-13a4-42b8-9de6-b9fd22c36a@sha256:758f83241b8f7d3440da750611067a82ac3553956dfe1767e27f1bdfdee48872 0.7s done |
| Jan 10 11:16:32 | #12 DONE 5.2s  |
| Jan 10 11:16:32 | Build time: 30.59 seconds  |

Senja 18.12 1814 10/01/2025