APLIKASI DAILY CALORIE BERBASIS WEB MENGGUNAKAN RESTFUL API GOLANG

Haryoko[1)] Aditya Himawan[2)] Andriyan Dwi Putra[3)]
Program Studi Informatika[1)2)] Program Studi Sistem Informasi[3)]
Fakultas Ilmu Komputer, Univesitas Amikom Yogyakarta, Jawa Tengah
haryoko@amikom.ac.id[1)] aditya.himawan@students.amikom.ac.id[2)] andriyan.putra@amikom.ac.id[3)]

ABSTRACT

The problem which is still unresolved is nutrition, from year to year, the problem of nutrition becomes one of the crucial aspects of obesity rates in Indonesia. The number of obesity cases increased by 35.4 percent in 2018 from 2007 which was only 19.1 percent. It is feared that this problem will continue to increase every year. Efforts to solve these problems can be overcome with self-awareness. In this problem, it was sparked to create a Daily Calorie Application that functions as a food diary and can track the calorie needs to be needed of each individual with an application that is integrated with the Open API, so it is hoped that the food menu will always be updated.
Keywords*: RESTful API, Golang, OPEN API, WDLC, calorie tracker*

ABSTRAK

Problematika yang sampai saat ini masih belum terselesaikan adalah masalah gizi, dari tahun ke tahun masalah gizi menjadi salah satu aspek yang krusial pada angka obesitas di Indonesia. Jumlah kasus obesitas mengalami peningkatan sebesar 35,4 persen pada tahun 2018 dari tahun 2007 yang hanya 19.1 persen. Masalah tersebut dikhawatirkan dapat terus naik tiap tahunnya. Upaya untuk menyelesaikan masalah tersebut bisa diatasi dengan kesadaran diri. Pada permasalahan tersebut maka tercetuslah untuk membuat Aplikasi *Daily Calorie* yang berfungsi sebagai buku harian makanan dan mampu melacak kebutuhan kalori yang dibutuhkan pada masing – masing individu dengan aplikasi yang terintegrasi dengan *Open API* maka diharapkan menu makanan akan selalu diperbaharui.
Kata Kunci: RESTfulAPI, Golang, OPEN API, WDLC, Pelacak Kalori.

PENDAHULUAN

Obesitas adalah suatu kondisi yang mengacu pada berat badan yang lebih besar dari apa yang dianggap normal atau dimana terdapat penumpukan lemak yang sangat tinggi di dalam tubuh. Obesitas tersebut terjadi karena asupan kalori lebih banyak dari pada aktivitas pembakaran kalori, sehingga kelebihan kalori tersebut menumpuk sebagai lemak. Jika kondisi ini terjadi dalam waktu yang lama, maka berat badan akan mengalami penambahan berat badan yang signifikan dan dapat mengakibatkan obesitas [1]. Aktivitas fisik yang rendah juga berpengaruh terhadap peningkatan prevalensi obesitas [2]. Menurut national institute of diabetes and digestive and kidney diseases obesitas menjadi salah satu penyebab dari faktor munculnya beberapa penyakit antara lain penyakit jantung, stroke, tekanan darah tinggi dan diabetes [3]. Tentunya hal tersebut dapat menimbulkan dampak negatif bagi tubuh. Dikutip dari *WHO* atau *World Health Organization* pada tahun 2020 penyakit tersebut menjadi salah satu dari sepuluh penyebab kematian tertinggi [4].

Masalah obesitas di indonesia sangat mengkhawatirkan. Menurut Riset Kesehatan Dasar (Riskesdas) 2018 menunjukkan bahwa proporsi obesitas di tahun 2018 mencapai 21.8% hal itu mengalami peningkatan dua kali lipat dalam kurun waktu 11 tahun terakhir pada tahun 2007 yang hanya mencapai 10.5% [5]. Tentunya untuk mengatasi dan mencegah masalah berat badan berlebih atau obesitas bisa dilakukan dengan cara asupan makan yang dimodifikasi, meningkatkan aktivitas fisik, olahraga rata – rata satu jam perhari, dan menurunkan berat badan dengan bantuan penurunan berat badan salah satu contohnya adalah program diet [6]. Adapun salah satu keberhasilan untuk menurunkan berat badan menurut *The 2013 White Paper from the American Nurse Practitioners Foundation on the Prevention and Treatment of Obesity* beranggapan bahwa buku harian makanan sebagai intervensi nutrisi berbasis bukti penting dalam faktor penurunan berat badan [7]. Perkembangan teknologi yang semakin maju membuat kita harus selalu berinovasi

dalam mengembangkan suatu aplikasi. Dan salah satu teknologi yang populer di dunia industri saat ini adalah penggunaan *API* (*Application Programming Interface*). Perkembangan teknologi yang semakin maju membuat kita harus selalu berinovasi dalam mengembangkan suatu aplikasi. Dan salah satu teknologi yang populer di dunia industri saat ini adalah penggunaan *API* (*Application Programming Interface*). Perkembangan teknologi yang selalu berkembang diperlukan aplikasi yang mampu dikembangkan seiring dengan kebutuhan masyarakat dan memiliki performa yang baik. Sehingga aplikasi daily calorie berbasis *RESTful API* menjadi solusinya. Dengan mengimplementasikan *RESTful API* maka aplikasi mampu menangani permintaan *user* secara baik dan mampu menyelesaikan masalah berat badan dan obesitas tersebut.

Penelitian yang berjudul Aplikasi Perhitungan Kebutuhan Kalori Dan Perhitungan Kalori Dari Makanan Yang Dikonsumsi. Aplikasi ini dibuat untuk mempermudah masyarakat menghitung kebutuhan kalorinya berdasarkan *basal metabolic rate* dengan menggunakan metode *Harris Benedict*. Aplikasi ini dibuat dengan bahasa pemrograman HTML, CSS, PHP dan MySQL sebagai databasenya [8]. Implementasi Metode *Harris Benedict* Pada Sistem Informasi Penghitungan Gizi Remaja Berbasis Website. Sistem informasi ini dibuat bertujuan untuk memberikan pengetahuan kepada remaja terhadap kebutuhan gizi. Aplikasi tersebut dibangun dengan PHP dan MySQL. Dalam aplikasi tersebut belum memiliki database makanan untuk menghitung asupan kalori makanan. Untuk menghitung kebutuhan kalori peneliti menggunakan metode *Harris Benedict* namun belum cukup informatif karena tidak terdapat jumlah kalori dari riwayat makanan yang dikonsumsi [9]. Sistem Informasi Panduan Diet Bagi Penderita Obesitas Berbasis Website. Tujuan peneliti adalah mengembangkan sistem informasi berbasis web yang dapat membantu penderita obesitas dalam menjalankan program dietnya dengan baik dan benar. Pada aplikasi tersebut teknologi yang digunakan menggunakan teknologi HTML, CSS, PHP, dan MySQL dengan menggunakan metode Waterfall. Pada aplikasi tersebut untuk menginputkan makanan dan detail informasi dari makanan tersebut masih bersifat manual [10]. Penelitian yang berjudul Rancang bangun Identifikasi Kebutuhan kalori dengan Aplikasi *Go Healthy Life*. Aplikasi ini dibuat untuk memudahkan masyarakat umum dalam mencari informasi gizi seperti status gizi, berat badan ideal, ukuran ideal, kebutuhan kalori total, dan pola makan. Pada penelitian tersebut aplikasi dibangun dengan bahasa pemrograman PHP dan MySql. Untuk menghitung status gizi peneliti menggunakan rumus *BMI* dan berdasarkan tabel berat badan ideal [11]. Aplikasi Penghitung Kebutuhan Gizi dalam Satuan Kalori Berbasis Web. Pada penelitian sebelumnya aplikasi tersebut dibangun dengan bahasa pemrograman PHP dan MySQL sebagai databasenya. Aplikasi tersebut dibangun dengan metode *Prototype*. Tujuan peneliti membangun aplikasi tersebut untuk memudahkan masyarakat dalam memperoleh informasi kebutuhan gizi kalori dan menu asupan [12].

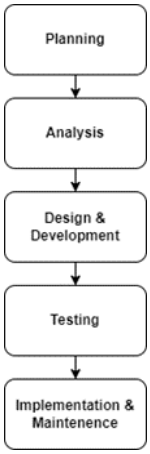
METODOLOGI PENELITIAN

Tahapan Penelitian

Web Development Life Cycle atau bisa disebut juga *WDLC* adalah sebuah metode yang bertujuan untuk membangun suatu website dimulai dari perencanaan sampai website tersebut di *deploy* atau dapat diakses secara publik. Suatu website dibangun dengan tujuan yang berbeda dan bervariasi dari mulai website komersial sampai Pendidikan dan masih banyak lagi. Walaupun setiap website memiliki tujuan dan fungsionalitas yang berbeda namun pengembangan *website* mengikuti pola desain dasar untuk memastikan konsistensi dan kelengkapan saat membangun sebuah *website*. Untuk itulah metode *WDLC* digunakan untuk membangun sampai memelihara website [13].

Tahapan WDLC

WDLC mempunyai lima tahapan yang harus diikuti untuk membangun sebuah website, antara lain yaitu *planning, analysis, design & development, testing, dan implementation & maintenance*. Pada *Gambar 1. Tahapan WDLC* merupakan tahapan dari *WDLC*.



Gambar 1. Tahapan WDLC

a. *Planning*

Tahapan pertama dari WDLC adalah planning atau perencanaan. Tahap perencanaan adalah tahapan yang penting karena jika suatu perencanaan berjalan salah maka tahapan selanjutnya juga akan berjalan dengan salah. Berikut ini adalah Langkah – Langkah yang harus dilakukan pada tahapan perencanaan.

1. Identifikasi tujuan dari website yang akan dibangun.
2. Identifikasi siapa yang akan menggunakan website.
3. Identifikasi teknologi web apa yang digunakan.
4. Identifikasi pemilik dan penulis konten dari sebuah website.
5. Tentukan dimana dan apa informasi yang dikirim di sebuah website.

b. *Analysis*

na sesuai dengan kondisi saat ini.

Tahapan *Analysis* adalah tahapan untuk mengumpulkan kebutuhan informasi dari pengguna, lalu menganalisanya secara sistematis dalam bentuk fungsionalitas sistem informasi, dan menganalisis inputan dari suatu kebutuhan website hingga menghasilkan *output* yang akan diterima oleh pengguna.

c. *Design & Development*

Tahapan *design & development* adalah tahapan untuk mempersiapkan cetak biru dari sebuah website. Dalam tahapan ini mempersiapkan berbagai diagram termasuk model data yang menjadi panduan dalam membangun sebuah website. Dalam tahapan ini melibatkan pemrograman dan pengujian program berdasarkan cetak biru yang sudah didesain.

d. *Testing*

Tahapan *testing* adalah tahapan untuk pengujian suatu website yang sudah dibangun dan menunjukan bahwa website sudah berjalan sesuai apa yang sudah direncanakan dari tahap perencanaan. Dalam tahapan ini website harus diuji dengan beberapa tahap seperti isi dari konten, fungsionalitas, kegunaan, dan ketepatan dari suatu website.

e. *Implementation & Maintenance*

Tahapan terakhir dari *WDLC* adalah *Implementation dan Maintenance*, dalam tahapan ini melibatkan persiapan data mulai dari server, *DBMS*, dan lainnya. Dalam tahap inilah pengguna dapat mengakses suatu website. Lalu dalam tahap ini juga melibatkan pemeliharaan website agar website selalu *up to date* dan memastikan bahwa informasi yang diberikan penggu

HASIL DAN PEMBAHASAN

Basis Data dan Relasi

Hasil dari rancangan database dapat diuraikan sebagai berikut.

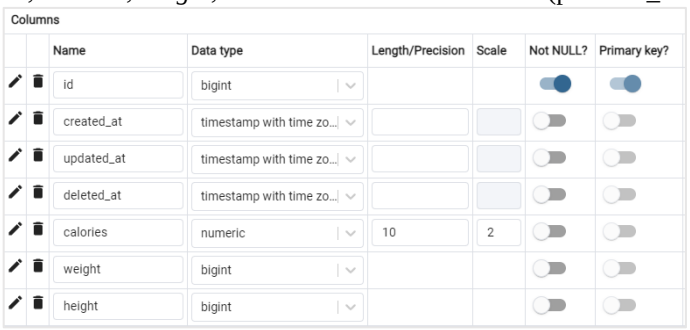
a. Tabel *User*

Hasil dari implementasi tabel *users* menghasilkan beberapa kolom utama diantaranya yaitu *id*, *name*, *email, password*, *avatar_url, gender, personal_data_id*, *created_at, updated_at,* dan *deleted_at*. Untuk lebih lengkapnya dapat dilihat pada Gambar 2 Implementasi Tabel User.

Gambar 2 Implementasi Tabel *User*

b. Tabel *Personal Data*

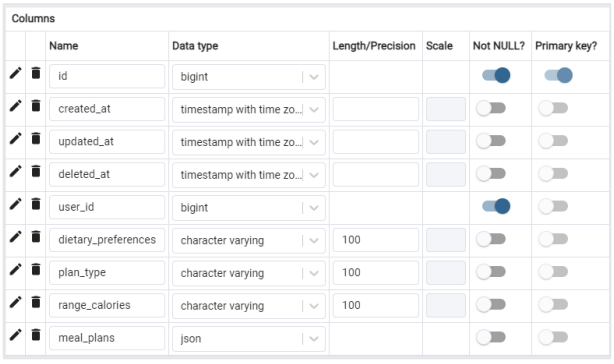
Hasil dari implementasi tabel personal data menghasilkan beberapa kolom utama diantaranya *yaitu id, calories, weight, height, created_at, updated_at,* dan *deleted_at*. Untuk lebih lengkapnya dapat dilihat pada Gambar 3 Implementasi Tabel Personal Data (personal_data).



Gambar 3 Implementasi Tabel *Personal Data (personal_data)*

c. Tabel *Meal Plans*

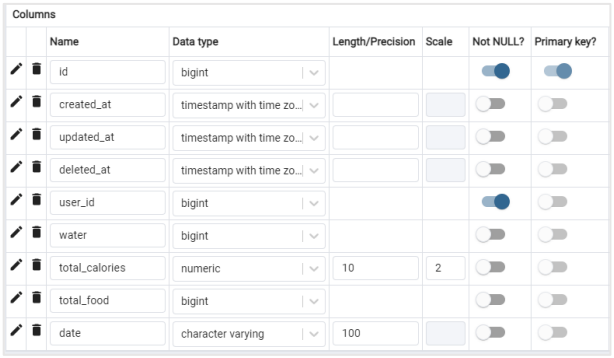
Hasil dari implementasi tabel *meal plans* menghasilkan beberapa kolom diantaranya yaitu id, user_id, dietary_preferences, plan_type, range_calories, meal_plans, created_at, updated_at, dan deleted_at. Untuk lebih lengkapnya dapat dilihat pada Gambar 4 Implementasi Tabel Meal Plans (meal_plans).



Gambar 4 Implementasi Tabel *Meal Plans (meal_plans)*

d. Tabel *Histories*

Hasil dari implementasi tabel *histories* menghasilkan beberapa kolom utama diantaranya *yaitu id, user_id, water, total_calories, total_food, date, created_at, updated_at,* dan *deleted_at*. Untuk lebih lengkapnya dapat dilihat pada Gambar 5 Implementasi Tabel Histories (histories).

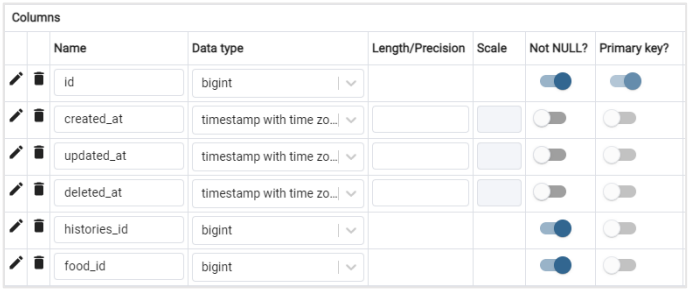

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	id	bigint			✓	✓
	created_at	timestamp with time zo...				
	updated_at	timestamp with time zo...				
	deleted_at	timestamp with time zo...				
	user_id	bigint			✓	
	water	bigint				
	total_calories	numeric	10	2		
	total_food	bigint				
	date	character varying	100			

Gambar 5 Implementasi Tabel *Histories (histories)*

e. Tabel *Histories Detail*

Hasil dari implementasi *histories* detail menghasilkan beberapa kolom utama diantaranya yaitu *id, histories_id, food_id, created_at, updated_at,* dan *deleted_at.* Untuk lebih lengkapnya dapat dilihat pada Gambar 6 Implementasi Tabel Histories Detail (histories_details).

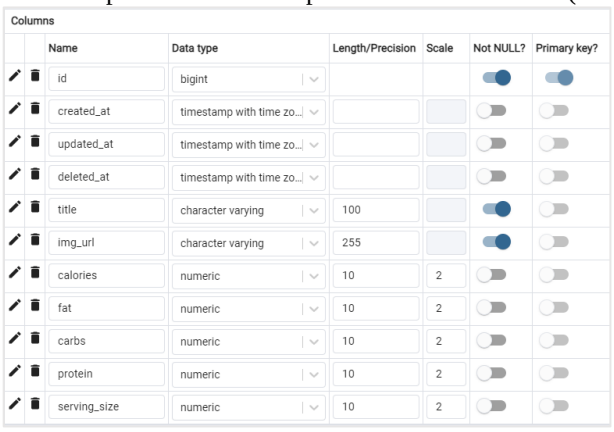

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	id	bigint			✓	✓
	created_at	timestamp with time zo...				
	updated_at	timestamp with time zo...				
	deleted_at	timestamp with time zo...				
	histories_id	bigint			✓	
	food_id	bigint			✓	

Gambar 6 Implementasi Tabel *Histories Detail (histories_details)*

f. Tabel Foods

Hasil dari implementasi tabel foods menghasilkan beberapa kolom utama diantaranya yaitu *id, title, img_url, calories, fat, carbs, protein, serving_size, created_at, updated_at,* dan *deleted_at*. Untuk lebih lengkapnya dapat dilihat pada Gambar 7 Implementasi Tabel Foods (foods).

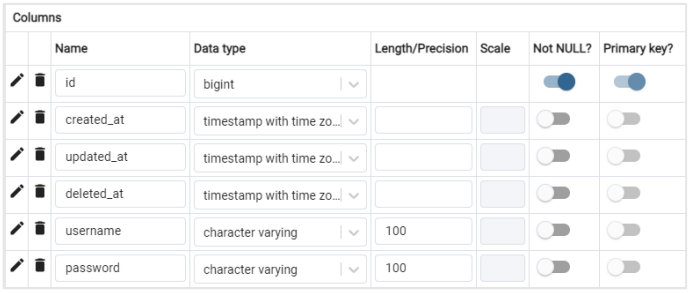

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	id	bigint			✓	✓
	created_at	timestamp with time zo...				
	updated_at	timestamp with time zo...				
	deleted_at	timestamp with time zo...				
	title	character varying	100		✓	
	img_url	character varying	255		✓	
	calories	numeric	10	2		
	fat	numeric	10	2		
	carbs	numeric	10	2		
	protein	numeric	10	2		
	serving_size	numeric	10	2		

Gambar 7 Implementasi Tabel *Foods (foods)*

g. Tabel Admin

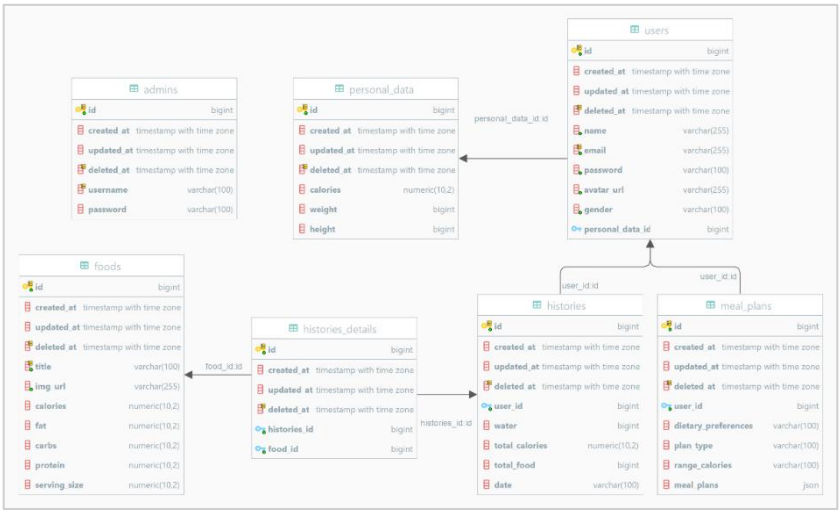
Hasil dari implementasi tabel admin menghasilkan beberapa kolom utama diantaranya yaitu id, username, password, created_at, updated_at, dan deleted_at. Untuk lebih lengkapnya dapat dilihat pada Gambar 8 Implementasi Tabel Admin (admins).



Gambar 8 Implementasi Tabel Admin (admins)

Relasi Database

Setelah migrasi tabel berhasil maka terbentuklah relasi antar tabel seperti pada **Gambar 9** Relasi Tabel. Visualisasi relasi tersebut dihasilkan otomatis menggunakan DataGrip.



Gambar 9 Relasi Tabel

RESTful API

Hasil dari RESTful API tersebut nantinya berbentuk JSON (javascript object notation). Data tersebut yang akan diolah oleh frontend untuk ditampilkan kepada client. Untuk mendapatkan data dari rest tersebut diperlukan HTTP Request, selanjutnya server akan memproses request tersebut dan menampilkannya melalui HTTP response yang datanya berbentuk JSON.

a. *Endpoint Users*

Endpoint user digunakan untuk menangani segala proses yang diperlukan *user* seperti *login*, mendaftarkan akun baru, merubah data *user*, melihat data *user*, menghitung kalori, dan menghapus *user*.

Tabel 1 *Endpoint User*

Method	URL	Status Code	Status	Auth	Fungsi
POST	/users/login	200	Berhasil		*Login user*
POST	/users/register	201	Berhasil		Menambahkan *user*
PUT	/users/:id	200	Berhasil	*	Merubah *user*
GET	/users	200	Berhasil	*	Melihat semua user
GET	/users/:id	200	Berhasil	*	Melihat *user* berdasarkan *id*
POST	/users/count-calories	200	Berhasil		Menghitung kalori

DELETE	/users/:id	200	Berhasil	*	Menghapus *user*

b. *Endpoint Admin*

Endpoint admin digunakan untuk menangani proses yang diperlukan *admin* seperti *login admin*, membuat *akun admin*, dan melihat data semua *user*.

Tabel 2 *Endpoint Admin*

Method	URL	Status Code	Status	Auth	Fungsi
POST	/admin/login	200	Berhasil		*Login admin*
POST	/admin/register	200	Berhasil		Menambahkan *admin*

c. *Endpoint Foods*

Endpoint foods digunakan untuk menangani segala proses yang berkaitan dengan *foods* seperti melihat data *food* berdasarkan id, melihat semua data *food*, menambahkan *food*, merubah data *food*, dan menghapus *food*.

Tabel 3 *Endpoint Foods*

Method	URL	Status Code	Status	Auth	Fungsi
GET	/foods/:id	200	Berhasil	*	Melihat food berdasarkan id
GET	/foods	200	Berhasil	*	Melihat semua food
GET	/foods/?name	200	Berhasil	*	Mencari food berdasarkan nama
POST	/foods	201	Berhasil	*	Menambahkan food
PUT	/foods/:id	200	Berhasil	*	Merubah food
DELETE	/foods/:id	200	Berhasil	*	Menghapus food

d. *Endpoint Histories*

Endpoint histories digunakan untuk menangani proses *histories*, seperti menambah *histories* secara manual atau otomatis melalui *open api*, menambahkan atau merubah data air minum, melihat *histories* terakhir, melihat semua data *histories*, melihat *histories* berdasarkan id, dan menghapus *histories detail*.

Tabel 4 *Endpoint Histories*

Method	URL	Status Code	Status	Auth	Fungsi
POST	/histories	201	Berhasil	*	Menambahkan histories manual
POST	/histories/water	200	Berhasil	*	Menambahkan atau merubah air minum
POST	/histories/automatic	201	Berhasil	*	Menambahkan histories melalui open api
GET	/histories/last	200	Berhasil	*	Melihat histories terakhir
GET	/histories/list	200	Berhasil	*	Melihat semua histories
GET	/histories/:id	200	Berhasil	*	Melihat histories berdasarkan id
DELETE	/histories/:id	200	Berhasil	*	Menghapus histories detail

e. *Endpoint Histories Detail*

Endpoint histories detail digunakan untuk menangani segala keperluan yang dibutuhkan *histories detail* seperti melihat semua *histories detail* berdasarkan *histories id*.

Tabel 5 Endpoint *Histories Detail*

Method	URL	Status Code	Status	Auth	Fungsi
GET	/histories_detail/all/:id	200	Berhasil	*	Melihat histories detail berdasarkan histories id
DELETE	/histories_detail/:id	200	Berhasil	*	Menghapus histories detail

f. *Endpoint Meal Plans*

Endpoint meal plan digunakan untuk menangani segala keperluan yang dibutuhkan meal plans seperti menambahkan *meal plan* dan melihat *meal plan* terakhir yang dihasilkan oleh *user*.

Tabel 6 *Meal Plans*

Method	URL	Status Code	Status	Auth	Fungsi
POST	/meal-plan	201	Berhasil	*	Menambahkan meal plan
GET	/meal-plan/last	200	Berhasil	*	Melihat meal plan terakhir

g. *Endpoint Open API*

Endpoint open api digunakan untuk menangani proses yang diperlukan *open api* seperti mendapatkan list data makanan dari proses pencarian dan mendapatkan data *meal plan*.

Tabel 7 *Open API*

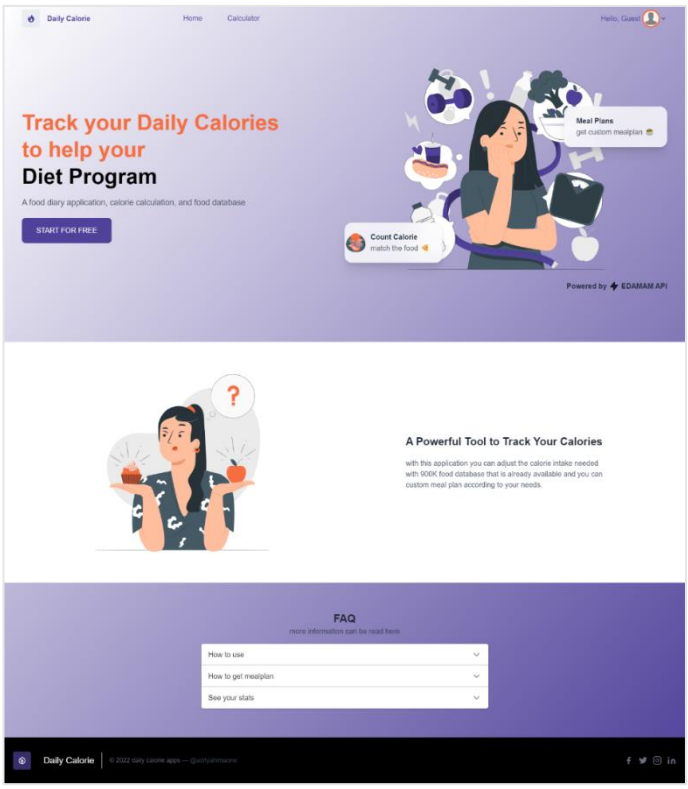
Method	URL	Status Code	Status	Auth	Fungsi
GET	/open-api/food/?name	200	Berhasil	*	Mendapatkan food
POST	/open-api/meal-plan	200	Berhasil	*	Mendapatkan meal plan

Implementasi Aplikasi

Antarmuka *Home*

a. Halaman *Landing Page*

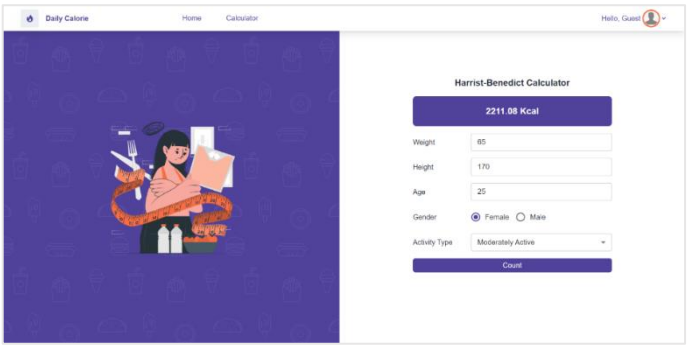
Halaman landing page merupakan halaman awal dari aplikasi ini. Pada halaman ini dibagi 3 *section* yaitu *hero section* untuk menarik pengunjung pada saat pertama kali pengunjung membuka aplikasi, *section about* berfungsi menjelaskan inti dari aplikasi, dan *section faq (frequently asked question)* untuk menjelaskan beberapa informasi yang dibutuhkan *user*.



Gambar 10 Landing Page

b. Halaman *Calculator*

Halaman *calculator* digunakan untuk menghitung kebutuhan kalori berdasarkan *harris benedict equation*, untuk mendapatkan nilai dari perhitungan tersebut diperlukan beberapa inputan seperti berat badan, tinggi badan, umur, gender, dan tipe aktivitas. Dalam halaman ini pengunjung website atau *user* dapat menghitung kebutuhan kalori mereka berdasarkan data diri.

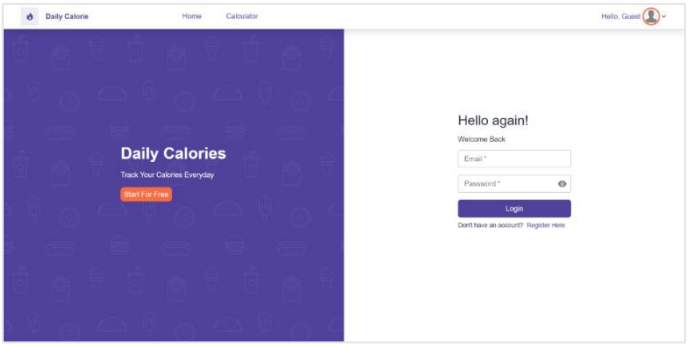


Gambar 11 *Calculator*

Antarmuka *Login*

1. Halaman *Login User*

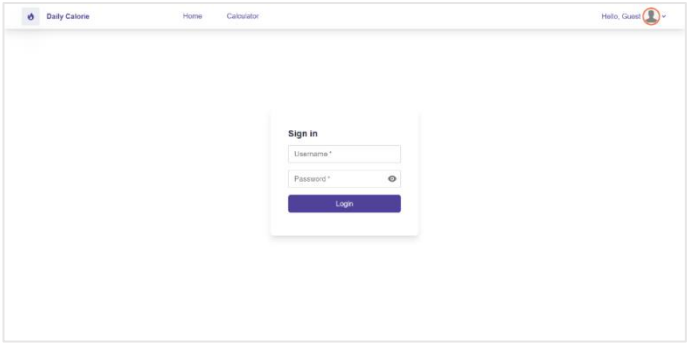
Halaman *login user* digunakan sebelum masuk ke halaman *dashboard user*, disini user dapat *show hide password* dan terdapat pesan jika *user* gagal *login*. Terdapat *link* ke halaman *register* jika sebelumnya pengunjung belum mempunyai akun.



Gambar 12 *Login User*

2. Halaman *Login Admin*

Halaman *login admin* digunakan sebelum masuk ke *dashboard admin*, terdapat fitur *show hide password* jika password yang dimasukan tidak yakin.



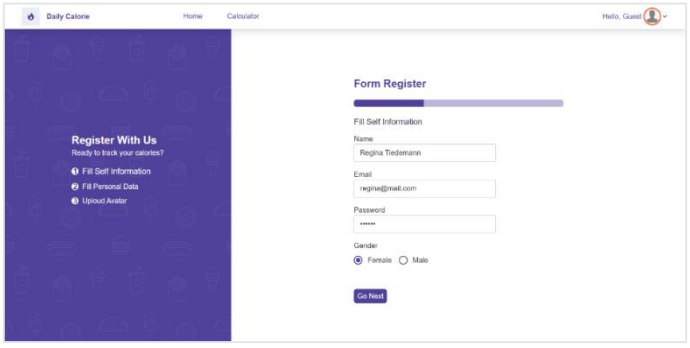
Gambar 13 *Login Admin*

Antarmuka *Register*

1. Halaman *Register User*

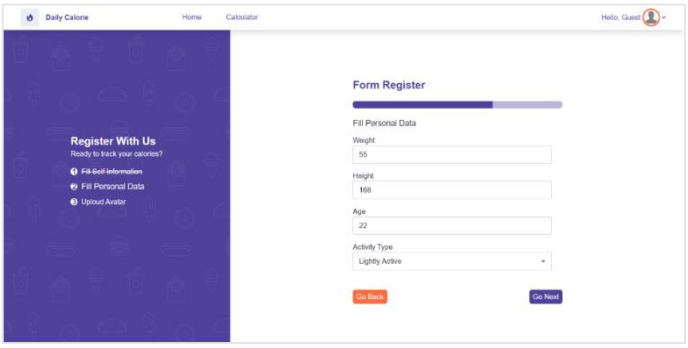
Halaman *register user* digunakan untuk mendaftarkan akun baru *user*, pada halaman ini dibagi menjadi 3 langkah. Langkah pertama untuk registrasi akun *user* adalah mengisi informasi diri dengan menginputkan nama, *email*, *password*, dan jenis kelamin untuk tampilannya bisa dilihat pada Gambar 14 Register *Step 1*, Langkah kedua adalah menginputkan data pribadi seperti berat badan, tinggi badan, umur, dan jenis aktivitas untuk tampilannya bisa dilihat pada gambar Gambar 15 Register *Step 2*, dan Langkah terakhir adalah mengunggah avatar yang dapat dilihat pada Gambar 16 Register Step 3. Setelah *form* pendaftaran tersubmit maka akan dialihkan ke halaman pendaftaran berhasil, *user* dapat memilih untuk *login* atau kembali ke halaman *register* untuk tampilannya dapat dilihat pada Gambar 17 Register *Completed*.

a. *Step 1*



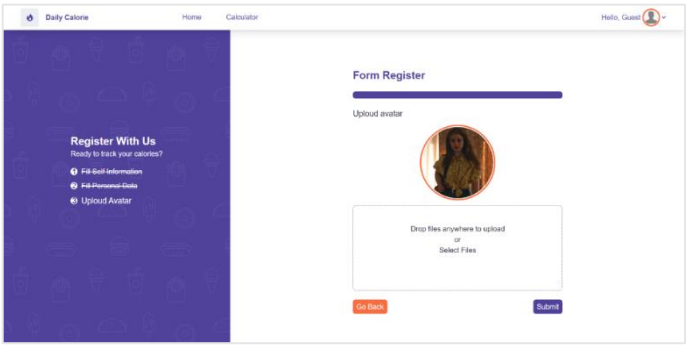
Gambar 14 *Register Step 1*

b. *Step 2*



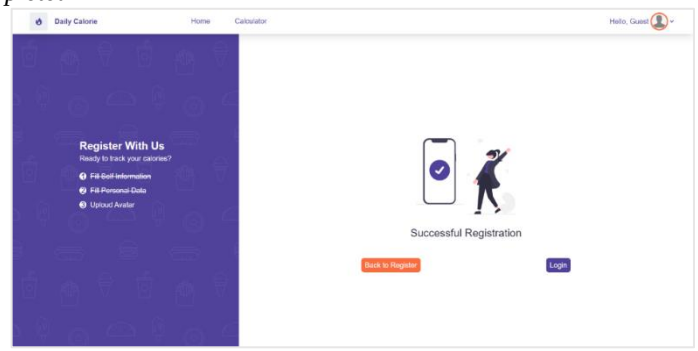
Gambar 15 *Register Step 2*

c. *Step 3*



Gambar 16 Register Step 3

d. *Step Completed*

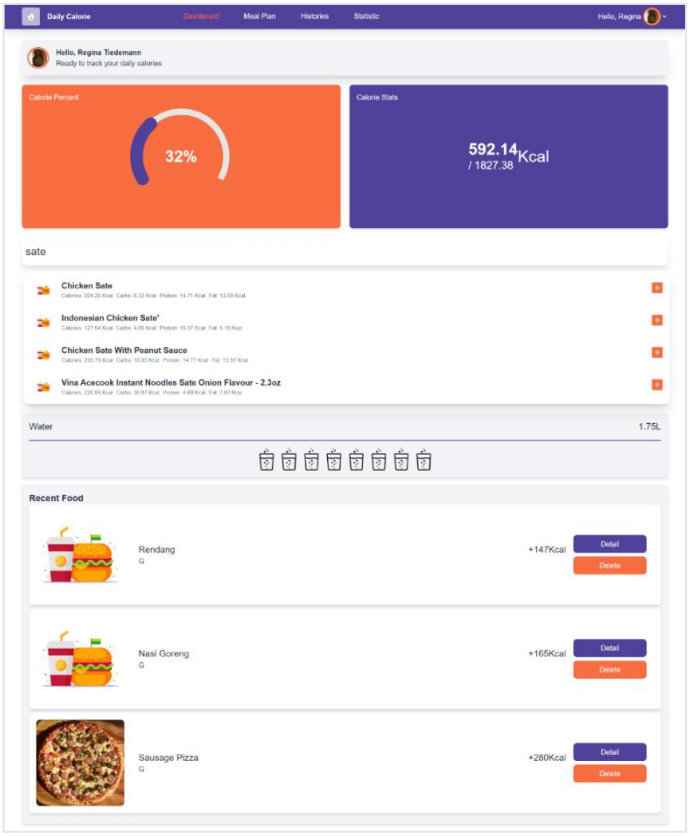


Gambar 17 *Register Completed*

Antarmuka Dashboard User

a. Halaman *Dashboard User*

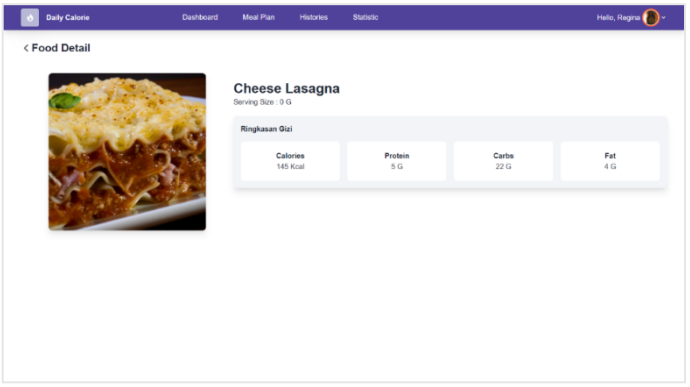
Halaman dashboard user merupakan halaman utama dari *user*, pada halaman ini *user* dapat melihat informasi jumlah kalori dari *user* yang sudah login dalam satu hari. Pada halaman ini *user* dapat menambahkan riwayat air minum dan menambahkan riwayat catatan makanan yang telah dikonsumsi dan tersambung ke *open api*, lalu *user* juga dapat melihat detail dari informasi gizi yang tampilannya dapat dilihat pada Gambar 19 *Food Detail* dan pada halaman ini *user* juga dapat menghapus riwayat makanan pada hari itu juga. Untuk tampilan *dashboard user* dapat dilihat pada Gambar 18 *Dashboard User*.



Gambar 18 *Dashboard User*

b. Halaman *Food Detail*

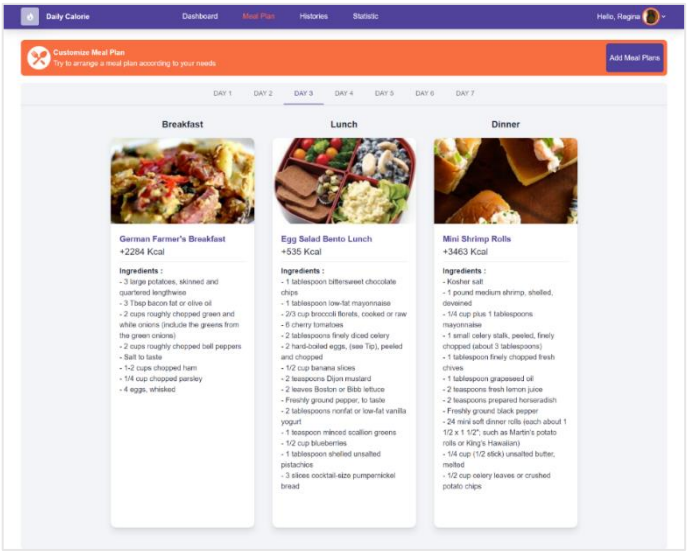
Halaman *food detail* digunakan untuk menampilkan detail makanan berupa nama makanan dan juga informasi gizi seperti jumlah kalori, lemak, karbohidrat dan juga protein.



Gambar 19 *Food Detail*

c. Halaman *Meal Plan*

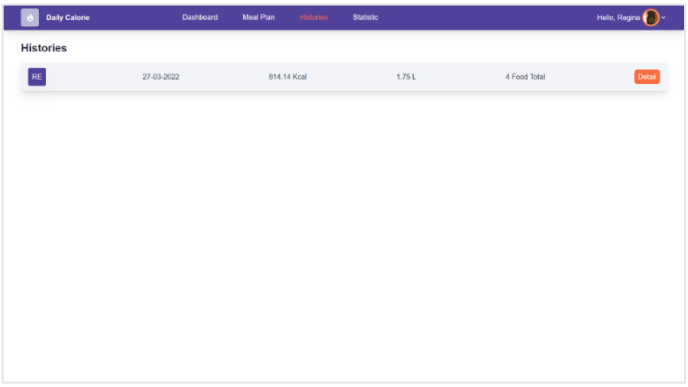
Halaman *meal plan* digunakan untuk menambahkan daftar rencana makanan yang berisi resep dan link yang menuju website resep yang bersangkutan. *Meal plan* dapat disesuaikan dengan kebutuhan *user* seperti harian atau mingguan, lalu *user* juga dapat memilih tipe diet, dan kisaran kalori. Untuk tampilannya dapat dilihat pada **Gambar 20** Halaman *Meal Plan*.



Gambar 20 Halaman *Meal Plan*

d. Halaman *Histories*

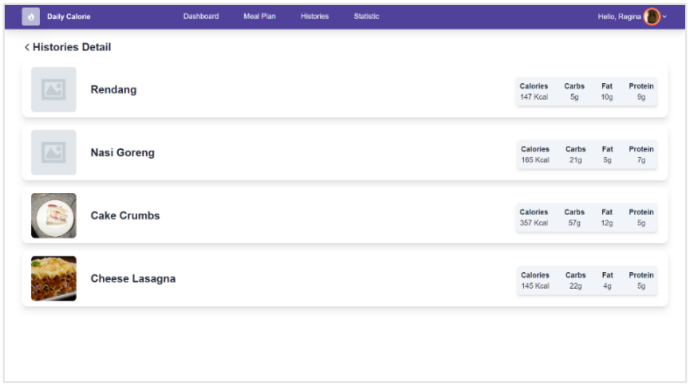
Halaman histories digunakan untuk melihat seluruh daftar riwayat yang dikelompokan dalam satu hari berdasarkan tanggal, pada halaman ini terdapat informasi seperti tanggal, jumlah kalori, jumlah air minum, dan total makanan. Terdapat tombol detail yang mengarahkan ke halaman histories detail yang dapat dapat dilihat pada Gambar 22 Halaman *Histories Detail*. Untuk tampilan halaman *histories* dapat dilihat pada Gambar 21 Halaman Histories.



Gambar 21 Halaman Histories

e. Halaman *Histories Detail*

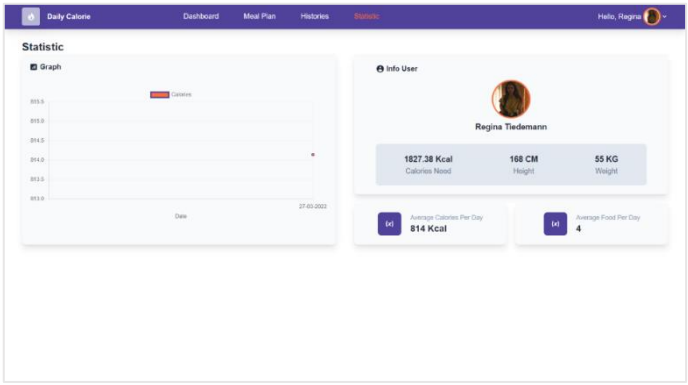
Halaman *histories detail* digunakan untuk menampilkan *detail histories*, pada halaman ini menampilkan daftar makanan yang telah dikonsumsi, terdapat informasi dari nama makanan, gambar makanan, dan informasi gizi. Untuk tampilannya dapat dilihat pada Gambar 22 Halaman Histories *Detail*.



Gambar 22 Halaman *Histories Detail*

f. Halaman *Statistic*

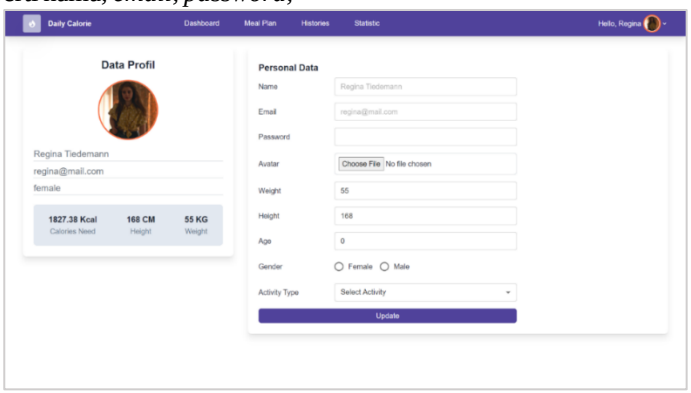
Halaman *statistic* digunakan untuk menampilkan informasi *user* mulai dari kebutuhan kalori, tinggi badan, berat badan. Pada halaman ini terdapat *card* yang menampilkan jumlah dan rata – rata makanan yang telah dikonsumsi, lalu terdapat juga bagan dari kalori harian user. Untuk tampilannya dapat dilihat Gambar 23 Halaman *Statistic*.



Gambar 23 Halaman *Statistic*

g. Halaman *Edit Profile*

Halaman *edit profile* digunakan untuk merubah data *user* yang sudah login atau masuk, pada halaman ini seorang *user* dapat merubah data seperti nama, *email*, *password*, *avatar*, berat badan, tinggi, dan kebutuhan kalori, untuk lebih lengkapnya dapat dilihat pada Gambar 24 Halaman *Edit Profile.*

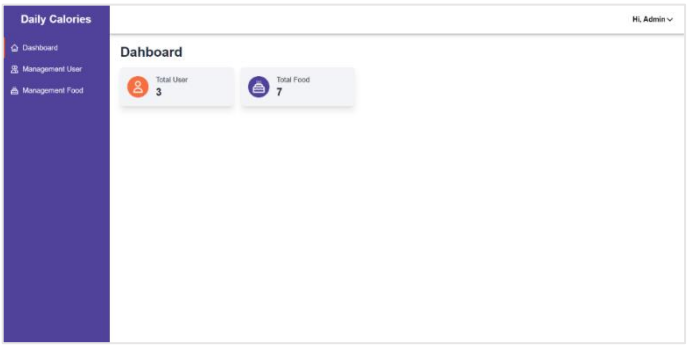


Gambar 24 Halaman *Edit Profile*

Antarmuka Dashboard Admin

a. Halaman *dashboard admin*

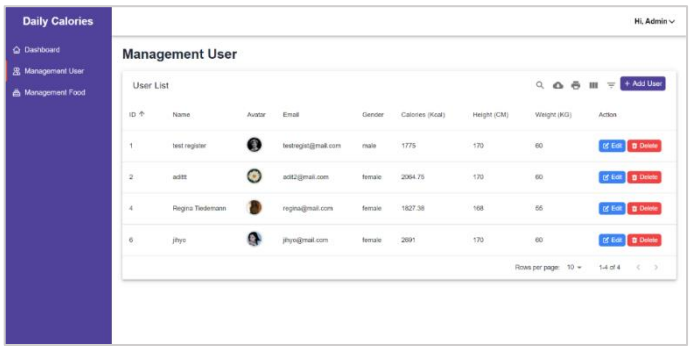
Halaman *dashboard admin* digunakan untuk menampilkan informasi dari total *user* yang sudah mendaftar dan menampilkan jumlah makanan yang tersimpan dalam *database*. Untuk tampilannya dapat dilihat pada Gambar 25 Halaman *Dashboard Admin*.



Gambar 25 Halaman *Dashboard Admin*

b. Halaman *Management User*

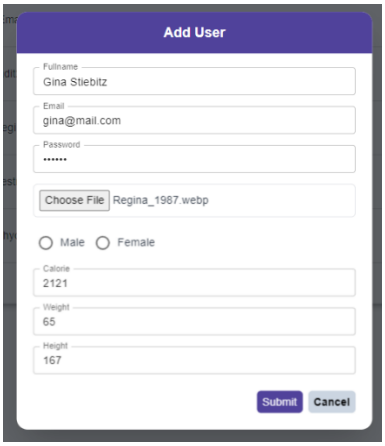
Halaman *management user* digunakan untuk mengelola akun yang terdaftar pada aplikasi, *admin* dapat melihat daftar *user* beserta informasinya seperti nama, *avatar*, *email*, jenis kelamin, kalori, tinggi badan, dan berat badan. Pada halaman ini *user* juga dapat melakukan menambahkan *user* baru, merubah data *user* serta menghapus *user*, untuk tampilannya dapat dilihat pada Gambar 26 Halaman *Management User*.



Gambar 26 Halaman *Management User*

c. *Modal Add User*

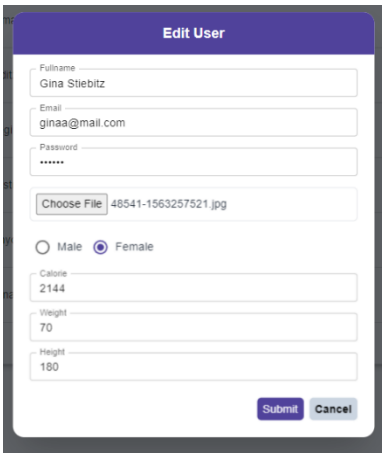
Modal add user digunakan untuk menambahkan *user* baru, terdapat inputan berupa nama lengkap, *email*, *password*, *upload avatar*, jenis kelamin, kebutuhan kalori, tinggi badan dan berat badan.



Gambar 27*Modal Add User*

d. *Modal Edit User*

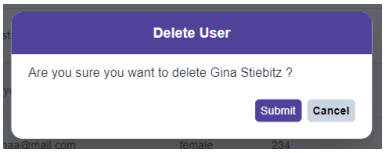
Modal edit user digunakan untuk merubah data *user* yang telah terdaftar, *admin* dapat merubah data *user* seperti nama lengkap, *email*, *password*, *avatar*, jenis kelamin, kebutuhan kalori, berat badan, dan tinggi badan.



Gambar 28 *Modal Edit User*

e. *Modal Delete User*

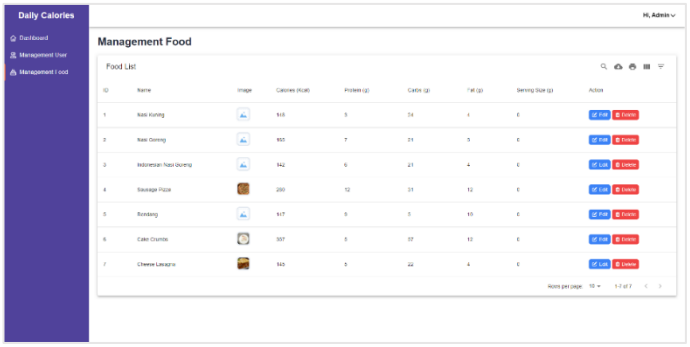
Modal delete user digunakan untuk menghapus *user* yang telah terdaftar, sebelum *user* dihapus admin dapat mengkonfirmasi apakah *user* jadi dihapus atau tidak.



Gambar 29 *Modal Delete User*

f. Halaman *Management Food*

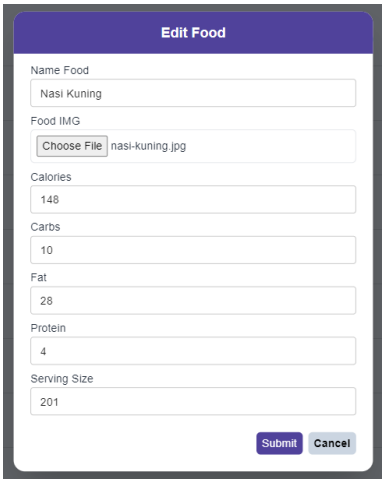
Halaman *management food* digunakan untuk mengelola makanan. Admin dapat melihat daftar makanan yang tersimpan secara otomatis ketika *user* menambahkan makanan melalui pencarian dan menambahkannya pada *histories*. Pada halaman ini *admin* dapat merubah data makanan serta menghapus makanan yang telah tersimpan. Untuk tampilannya dapat dilihat pada Gambar 30 Halaman *Management Food*.



Gambar 30 Halaman *Management Food*

g. *Modal Edit Food*

Modal edit food digunakan untuk merubah data makanan yang telah tersimpan di *database*, *admin* dapat merubah data makanan seperti nama makanan, gambar makanan, kalori, karbohidrat, lemak, protein dan ukuran porsi makanan.

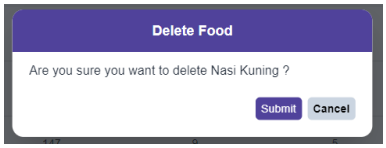


Gambar 31 *Modal Edit Food*

h. *Modal Delete Food*

Modal delete food digunakan untuk menghapus makanan yang telah tersimpan nya.

dalam *database*. *Admin* dapat mengkonfirmasi apakah ingin melanjutkan menghapus makanan atau membatalkan



Gambar 32 *Modal Delete Food*

SIMPULAN

Pembuatan Aplikasi Daily Calorie berbasis web menggunakan RESTful API Golang dapat dilakukan dengan langkah – langkah sebagai berikut:

a. Pembuatan aplikasi daily calorie dilakukan dengan cara mengindentifikasi dan menganalisis masalah yang ada.
b. Perancangan aplikasi daily calorie dilakukan dengan metode WDLC.
c. Pembuatan backend menggunakan bahasa pemrogaman golang dengan desain perangkat lunak Clean Architecture.
d. Pembuatan frontend menggunakan ReactJS.
e. Pengujian aplikasi menggunakan unit test dan black box testing.

DAFTAR PUSTAKA

[1] NIDDK, “Definition & Facts for Adult Overweight & Obesity,” 2018. https://www.niddk.nih.gov/health-information/weight-management/adult-overweight-obesity (accessed Oct. 13, 2021).

[2] S. Singh, R. Issac, A. I. Benjamin, and S. Kaushal, “Prevalence and association of physical activity with obesity: an urban, community-based, cross-sectional study,” *Indian J. Community Med.*, vol. 40, no. 2, pp. 103–107, 2015, doi: 10.4103/0970-0218.153873.

[3] NIDDK, “Health Risks of Overweight & Obesity,” 2018. https://www.niddk.nih.gov/health-information/weight-management/adult-overweight-obesity/health-risks?dkrd=/health-information/weight-management/health-risks-overweight (accessed Oct. 13, 2021).

[4] WHO, “The top 10 causes of death,” 2020. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (accessed Oct. 13, 2021).

[5] Kementerian Kesehatan Republik Indonesia, “Laporan Nasional RISKESDAS 2018,” 2019. [Online]. Available: https://www.kemkes.go.id/article/view/19093000001/penyakit-jantung-penyebab-kematian-terbanyak-ke-2-di-indonesia.html

[6] S. M. Fruh, “Obesity: Risk factors, complications, and strategies for sustainable long-term weight management,” *J. Am. Assoc. Nurse Pract.*, vol. 29, no. S1, pp. S3–S14, Oct. 2017, doi: 10.1002/2327-6924.12510.

[7] American Nurse Practitioner Foundation, “Nurse practitioners and theprevention and treatment of adult obesity—A White Paper of the American NursePractitioner Foundation (electronic version),” 2013. [Online]. Available: https://international.aanp.org/Content/docs/ObesityWhitePaper.pdf

[8] Mufid Ajidarma, “Aplikasi Perhitungan Kebutuhan Kalori Dan Perhitungan Kalori Dari Makanan Yang Dikonsumsi,” 2019.

[9] Arya Guntara, “Implementasi Metode Harris Benedict Pada Sistem Informasi Penghitungan Gizi Remaja Berbasis Website,” vol. Vol.1; No:, 2021, Accessed: Nov. 02, 2021. [Online]. Available: https://publikasi.hawari.id/index.php/jnastek/article/view/3

[10] R. Rizal and A. Rahmatulloh, “RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan.”

[11] A. Dwi Suryani, Q. Jafar Ardian,

and Rusliyawati, "RANCANG BANGUN IDENTIFIKASI KEBUTUHAN KALORI DENGAN APLIKASI GO HEALTHY LIFE," 2020. [Online]. Available: http://jim.teknokrat.ac.id/index.php/sisteminformasi

[12] H. Muhtarom, K. F. Liyana, K. Larasati, and A. R. Supriyatna, "Aplikasi Penghitung Kebutuhan Gizi dalam Satuan Kalori Berbasis Web," 2019.

[13] P. Iyer and P. Singh, "Software Engineering:Web Development Life Cycle." [Online]. Available: www.ijert.org