

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Oleh :

Kelompok 3

Teuku Bintang Hadi Negara : 2311103139

Muhammad Rafi Awallaisal : 2311103134

Muhammad Raflan Kemal : 2311103132

S1SI07C

PROGRAM STUDI S1 SISTEM INFORMASI

FAKULTAS REKAYASA INDUSTRI

UNIVERSITAS TELKOM PURWOKERTO

2024

Calorie Counter

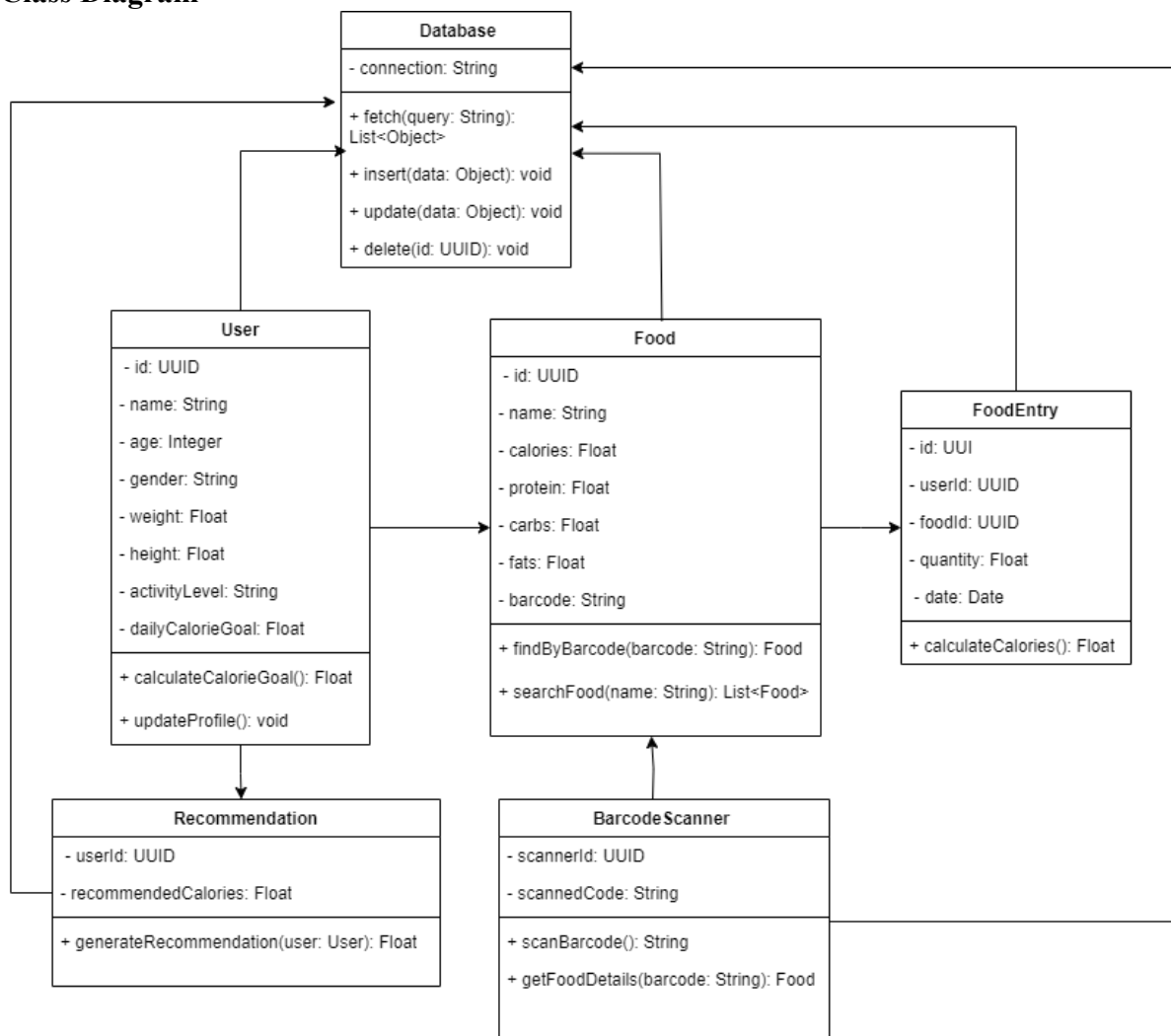
1. Deskripsi:

Aplikasi untuk menghitung asupan kalori harian berdasarkan makanan yang dikonsumsi.

Teknologi:

- Frontend: React.js
- Backend: Express.js
- Fitur:
 - Database makanan dengan kandungan kalori
 - Rekomendasi kalori harian berdasarkan profil pengguna
 - Fitur scan barcode makanan

2. Class Diagram



3. Penjelasan Class Diagram

1. Database
 - Kelas ini menyediakan fungsi-fungsi untuk operasi database seperti fetch, insert, update, dan delete. Semua kelas yang membutuhkan akses data bergantung pada kelas ini.
2. User
 - Merepresentasikan pengguna sistem.
 - Atribut utama: id, name, age, gender, weight, height, activityLevel, dan dailyCalorieGoal.
 - Fungsi utama: menghitung kebutuhan kalori (calculateCalorieGoal) dan memperbarui profil (updateProfile).
3. Food
 - Merepresentasikan data makanan.
 - Atribut utama: id, name, calories, protein, carbs, fats, dan barcode.
 - Fungsi utama: mencari makanan berdasarkan barcode (findByBarcode) atau nama (searchFood).
4. FoodEntry
 - Menghubungkan *User* dengan *Food*.
 - Atribut utama: id, userId, foodId, quantity.
 - Fungsi utama: menghitung jumlah kalori berdasarkan jumlah makanan (calculateCalories).
5. BarcodeScanner
 - Kelas untuk memindai barcode makanan.
 - Atribut utama: scannerId dan scannedCode.
 - Fungsi utama: memindai barcode (scanBarcode) dan mengambil detail makanan berdasarkan barcode (getFoodDetails).
6. Recommendation
 - Memberikan rekomendasi kebutuhan kalori untuk pengguna.
 - Atribut utama: userId dan recommendedCalories.
 - Fungsi utama: menghasilkan rekomendasi kalori berdasarkan profil pengguna (generateRecommendation).

Relasi Antar Kelas:

1. User
 - Hubungan One-to-Many: Seorang pengguna (*User*) dapat memiliki banyak entri makanan (*FoodEntry*).
2. Food
 - Hubungan One-to-Many: Beberapa entri makanan (*FoodEntry*) dapat merujuk ke satu item makanan (*Food*).
3. FoodEntry
 - Asosiasi: *FoodEntry* berfungsi sebagai penghubung antara pengguna (*User*) dan makanan (*Food*).
4. BarcodeScanner
 - Agregasi: *BarcodeScanner* berinteraksi dengan *Food* untuk mengambil detail berdasarkan kode batang (*barcode*).
5. Recommendation

- Asosiasi: *Recommendation* (rekomendasi) terkait langsung dengan pengguna (*User*) tertentu.

6. Database

- Komponen Bersama: *Database* diakses oleh semua kelas untuk operasi CRUD (*Create, Read, Update, Delete*).