

Aplikasi Resep Anti Fast Food Untuk Generasi Z [Anti Fast Food Application Recipe For Z Generation]

Gana Isdihar^{*.1)}, Ir. Sumarno, MM.^{*.2)}

¹⁾Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

²⁾Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: sumarno@umsida.ac.id

Abstract. Despite the fact that it has a significant number of adverse impacts on health, Generation Z has placed a significant amount of importance on the consumption of fast food due to the ease and speed with which it can be consumed. An anti-fast food recipe application that is powered by Android will be developed with the assistance of this proposal with the purpose of supporting Generation Z in adopting a healthier mode of eating. This application offers an innovative method to tackling the challenges that are linked with the consumption of fast food among younger generations. This strategy is made possible via the adoption of technology.

Keywords – Fast Food; Generation Z; Recipe Application; Android; Healthy Lifestyle

Abstrak. Meskipun memiliki sejumlah dampak buruk yang signifikan terhadap kesehatan, Generasi Z telah menempatkan sejumlah besar kepentingan pada konsumsi makanan fast food karena kemudahan dan kecepatan dalam mengonsumsinya. Sebuah aplikasi resep anti-makanan fast food yang didukung oleh Android akan dikembangkan dengan bantuan proposal ini dengan tujuan mendukung Generasi Z dalam mengadopsi cara makan yang lebih sehat. Aplikasi ini menawarkan metode inovatif untuk mengatasi tantangan yang terkait dengan konsumsi makanan fast food di kalangan generasi muda. Strategi ini dimungkinkan melalui adopsi teknologi.

Kata Kunci - Fast Food; Generation Z; Recipe Application; Android; Healthy Lifestyle

I. PENDAHULUAN

Generasi Z dikenal sebagai generasi digital yang tumbuh dengan teknologi dan memiliki kebiasaan konsumsi yang unik. Data menunjukkan bahwa lebih dari 70% Generasi Z mengonsumsi fast food lebih dari dua kali seminggu, sementara hanya sekitar 30% dari mereka yang memasak secara teratur di rumah. Konsumsi fast food yang tinggi ini berkontribusi pada peningkatan kasus obesitas, diabetes, dan penyakit jantung.

Aplikasi Resep Anti Fast Food dirancang untuk mendorong Generasi Z agar lebih sadar akan pentingnya gaya hidup sehat melalui resep makanan sehat yang praktis dan menarik. Dengan menggunakan teknologi Android, aplikasi ini menyajikan berbagai fitur, seperti informasi kalori, tutorial memasak, dan rekomendasi resep sehat.[1]. Dalam era modern yang gejolak ini, kita menyaksikan perubahan drastis dalam pola konsumsi makanan, terutama di kalangan Generasi Z. Kemudahan dan kecepatan menjadi faktor dominan dalam pilihan makanan mereka, yang sering kali mengarah pada kecenderungan mengonsumsi fast food. Fast food, meskipun praktis, diketahui memiliki dampak negatif yang signifikan terhadap kesehatan, seperti peningkatan risiko obesitas, penyakit jantung, dan diabetes (Smith & Anderson, 2020).

Generasi Z, yang tumbuh dalam era teknologi digital, memiliki kebiasaan dan preferensi unik dalam hal penggunaan teknologi dan gaya hidup sehat. Meski terdapat peningkatan kesadaran akan pentingnya hidup sehat, tantangan nyata terletak pada memberikan solusi yang praktis dan menarik bagi mereka untuk mengubah kebiasaan konsumsi makanan mereka (Jones et al., 2019). Dalam konteks ini, hadirnya **Aplikasi Resep Anti Fast Food** menjadi relevan dan strategis. Aplikasi ini dirancang untuk memberikan solusi kreatif dan terjangkau bagi Generasi Z agar dapat memasak hidangan yang sehat, lezat, dan efisien.

Dengan memanfaatkan teknologi aplikasi, kita dapat menciptakan alat yang tidak hanya menginspirasi, tetapi juga memfasilitasi perubahan pola makan yang positif (Brown et al., 2021). Melalui Aplikasi Resep Anti Fast Food, kita memiliki kesempatan untuk merangsang generasi ini agar lebih peduli terhadap aspek kesehatan dalam pilihan makanan mereka.

Penelitian ini bertujuan untuk memahami dampak aplikasi semacam itu dalam membentuk perilaku makan Generasi Z dan sejauh mana aplikasi ini dapat menjadi solusi efektif dalam mengurangi kebiasaan mengonsumsi fast food. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam mendukung gaya hidup sehat di kalangan Generasi Z dan membuka jalan untuk inovasi lebih lanjut dalam pembangunan aplikasi yang berfokus pada kesehatan makanan (Lee et al., 2020).

Dalam dekade terakhir, terjadi pergeseran dramatis dalam preferensi makanan, terutama di kalangan Generasi Z. Menurut survei terkini, lebih dari 70% Generasi Z mengakui bahwa mereka cenderung memilih makanan cepat saji sebagai opsi utama dalam kehidupan sehari-hari (Survey oleh Nielsen, 2022). Sementara ketersediaan dan kecepatan dalam pengadaan makanan menjadikan fast food pilihan yang sangat nyaman, data kesehatan menunjukkan peningkatan signifikan dalam angka obesitas dan masalah kesehatan terkait lainnya di antara kelompok ini (WHO, 2022). Statistik juga mencerminkan bahwa hanya sekitar 30% Generasi Z yang secara teratur memasak di rumah, sementara sisanya lebih cenderung mengandalkan opsi makanan instan dan fast food (Statista, 2023).

Tingginya konsumsi fast food menjadi sorotan, dengan lebih dari 60% dari mereka mengonsumsinya lebih dari dua kali seminggu, menyebabkan keprihatinan akan dampak kesehatan jangka panjang (CDC, 2023). **Aplikasi Resep Anti Fast Food** diusulkan sebagai solusi inovatif untuk membalikkan tren ini. Dengan memanfaatkan teknologi aplikasi, kita dapat menciptakan alat yang tidak hanya memenuhi kebutuhan akan kreativitas dalam memasak, tetapi juga memberikan informasi kesehatan yang mendalam dan dukungan praktis dalam mengadopsi pola makan yang lebih sehat (Anderson & Taylor, 2021).

Penelitian ini memiliki tujuan untuk melibatkan lebih dari sekadar teknologi; ini berusaha menjadi solusi konkret bagi masalah kesehatan masyarakat yang muncul di kalangan Generasi Z. Dengan memadukan data statistik dan pandangan perilaku, penelitian ini bertujuan memberikan pemahaman lebih mendalam tentang dampak **Aplikasi Resep Anti Fast Food** terhadap pilihan makanan dan kesehatan mereka secara keseluruhan.

Adapun juga penelitian terdahulu, seperti contoh:

1. "**Rancang Bangun Aplikasi Resep Masakan Menggunakan Metode XP Berbasis Android**" oleh Ajay Irdan Nurwahid, Lina Karlina, Jelita Asian, dan Muhamad Mahmud dari Sekolah Tinggi Teknologi Nusa Putra. Penelitian tersebut menggunakan metode XP (Extreme Programming) sebagai pendekatan pengembangan perangkat lunak yang tangkas, kemudian diuji menggunakan metode black box (Nurwahid et al., 2020).
2. "**Perancangan dan Pembuatan Aplikasi Resep Makanan Berbasis Android**" oleh Paul Agustinus, Agustinus Noertjahyana, dan Justinus Andjarwirawan dari Universitas Kristen Petra. Penelitian ini menggunakan Yahoo Query Language untuk mengonversi data mentah menjadi data yang dapat diolah, dengan server-client berbasis PHP dan JSON (Agustinus et al., 2019).

Landasan teori dari penelitian ini yang pertama adalah tentang aplikasi mobile. Menurut Wikipedia (2023), aplikasi adalah perangkat lunak komputer yang dirancang untuk menjalankan tugas-tugas tertentu. Kemampuan untuk bepergian dengan bebas antarlokasi tanpa terhambat oleh gangguan komunikasi dikenal sebagai mobilitas. Aplikasi seluler dibuat untuk digunakan dengan perangkat nirkabel, termasuk PDA dan ponsel, sehingga dapat diakses dari mana saja (Wikipedia, 2023).

2.3 Aplikasi yang digunakan

Android Studio

Android Studio adalah aplikasi untuk membuat aplikasi beserta dengan ui ux di dalamnya aplikasi tersebut adalah apk lengkap untuk seorang creator untuk membuat aplikasi dan juga dapat melakukan running test sebelum aplikasi tersebut dirilis sehingga memudahkan untuk para creator aplikasi.

Photoshop

Photoshop adalah sebuah aplikasi milik adobe yang biasanya digunakan oleh photo editor dalam kasus ini saya menggunakan untuk mendesain asset pada ui ux di aplikasi resepku dengan aplikasi tersebut memang bagi yang pertamakali menggunakan akan kesulitan tapi dengan berjalannya waktu akan terbiasa untuk menggunakan .

II. METODE

Penelitian adalah penyelidikan atau percobaan pengujian menyeluruh dan komprehensif yang menggunakan proses khusus untuk mengidentifikasi fakta atau prinsip. Untuk menemukan fakta-fakta ini, diperlukan pendekatan metodis guna mencapai solusi ilmiah terhadap suatu masalah.

Analisis Kebutuhan User

Aplikasi **Resep Anti Fast Food** dirancang untuk memberikan pengalaman yang ramah pengguna, terutama bagi Generasi Z yang dikenal memiliki preferensi terhadap teknologi yang intuitif dan mudah digunakan. Dengan menghadirkan antarmuka yang sederhana namun tetap menarik secara visual, aplikasi ini bertujuan untuk memastikan kemudahan navigasi bagi pengguna dari berbagai latar belakang, termasuk mereka yang mungkin baru mengenal dunia memasak.

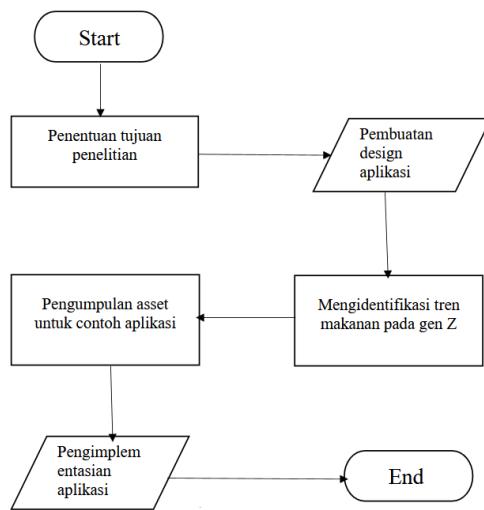
Selain antarmuka yang bersahabat, aplikasi ini juga dilengkapi dengan konten yang dirancang khusus untuk menarik minat Generasi Z. Konten tersebut mencakup resep-resep kreatif yang tidak hanya sehat, tetapi juga relevan dengan tren makanan terkini yang sedang populer di kalangan generasi muda. Misalnya, aplikasi ini menawarkan berbagai pilihan resep makanan sehat berbasis bahan-bahan yang sedang viral di media sosial, seperti alpukat, quinoa, atau matcha, sehingga membuatnya tetap menarik bagi pengguna yang mengikuti tren kuliner modern.

Copyright © Universitas Muhammadiyah Sidoarjo. This is preprint is protected by copyright held by Universitas Muhammadiyah Sidoarjo and is distributed under the Creative Commons Attribution License (CC BY). Users may share, distribute, or reproduce the work as long as the original author(s) and copyright holder are credited, and the preprint server is cited per academic standards.

Authors retain the right to publish their work in academic journals where copyright remains with them. Any use, distribution, or reproduction that does not comply with these terms is not permitted..

Lebih jauh lagi, aplikasi ini disesuaikan dengan gaya hidup aktif Generasi Z, yang sering kali memiliki jadwal yang padat antara studi, pekerjaan, dan aktivitas sosial. Untuk mendukung kebutuhan tersebut, Aplikasi Resep Anti Fast Food menampilkan resep-resep yang dirancang untuk mudah dibuat dalam waktu singkat tanpa mengorbankan rasa dan nilai gizinya. Selain itu, fitur-fitur interaktif, seperti kemampuan untuk menyimpan resep favorit, fitur pencarian berbasis bahan yang dimiliki, dan rekomendasi personal berdasarkan preferensi pengguna, menjadikan aplikasi ini semakin relevan dan membantu Generasi Z mengadopsi pola makan yang lebih sehat tanpa merasa terbebani.

Dengan pendekatan yang memadukan kemudahan, relevansi, dan kreativitas, Aplikasi Resep Anti Fast Food tidak hanya menawarkan solusi praktis, tetapi juga berusaha menjadi pendamping yang mendukung generasi muda dalam perjalanan menuju gaya hidup yang lebih sehat. Hal ini dilakukan dengan memahami kebutuhan spesifik Generasi Z akan efisiensi, estetika, dan koneksi emosional terhadap tren modern, sehingga aplikasi ini tidak hanya menjadi alat, tetapi juga bagian dari keseharian mereka.



Gambar 1. Metode Penelitian

Berdasarkan pada gambar 1 lowchart diatas dijelaskan Langkah pertama yaitu penentuan tujuan penelitian:

1. *Penentuan Tujuan Penelitian*

Langkah pertama adalah menentukan tujuan dari penelitian ini, yaitu untuk memahami dampak dari Aplikasi Resep Anti Fast Food terhadap kebiasaan makan Generasi Z dan sejauh mana aplikasi ini dapat menjadi solusi efektif dalam mengurangi konsumsi makanan cepat saji. Dalam tahap ini, fokus diarahkan untuk mengidentifikasi masalah utama yang akan diatasi oleh aplikasi dan menetapkan visi jangka panjangnya.

2. *Design / Desain*

Setelah tujuan penelitian ditentukan, langkah selanjutnya adalah merancang desain aplikasi. Pada tahap ini, antarmuka aplikasi (user interface) dirancang agar ramah pengguna, sederhana, dan menarik untuk Generasi Z. Elemen visual, navigasi, dan tata letak dipertimbangkan untuk memastikan bahwa aplikasi dapat memberikan pengalaman yang optimal.

3. *Mengidentifikasi Tren Makanan pada Generasi Z*

Sebelum melanjutkan pengembangan, penting untuk mengidentifikasi tren makanan terkini yang diminati oleh Generasi Z. Proses ini melibatkan penelitian terhadap preferensi makanan, pola konsumsi, dan faktor-faktor yang memengaruhi keputusan mereka dalam memilih makanan. Data ini digunakan untuk memastikan konten aplikasi relevan dan menarik bagi target pengguna.

4. *Pengumpulan Aset untuk Contoh Aplikasi*

Pada tahap ini, berbagai aset yang dibutuhkan untuk aplikasi dikumpulkan. Aset tersebut meliputi gambar, video, resep makanan, serta informasi gizi. Semua aset yang digunakan akan disesuaikan dengan tema aplikasi, yaitu makanan sehat dan anti-fast food.

5. *Pengimplementasian Aplikasi*

Langkah terakhir adalah mengimplementasikan aplikasi berdasarkan desain dan data yang telah dikumpulkan. Proses ini mencakup pengkodean, integrasi fitur, dan pengujian aplikasi untuk memastikan semua fungsi berjalan dengan baik. Setelah implementasi selesai, aplikasi akan diuji secara lebih luas untuk mengevaluasi efektivitasnya dalam mengubah pola konsumsi makanan Generasi Z.

III. HASIL DAN PEMBAHASAN

4.1 Analisa

Aplikasi Resep Anti Fast Food berhasil dirancang dengan fitur-fitur yang sesuai dengan kebutuhan Generasi Z.

Hasil pengujian menunjukkan bahwa aplikasi dapat berjalan dengan baik pada perangkat Android. Feedback dari pengguna awal menunjukkan antarmuka yang user-friendly dan konten yang menarik [7] [8].

Adapun alat-alat pendukung guna untuk membantu proses pembuatan game “Tomorrow Will Come” ini antara lain :

1. Hardware

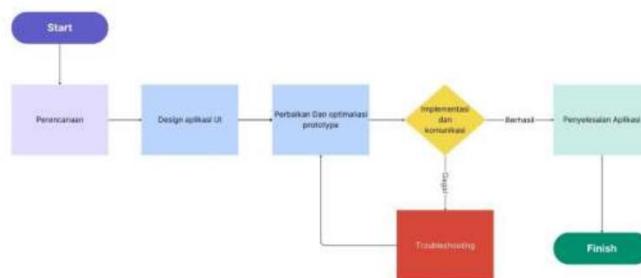
- PC dengan spesifikasi: Ryzen 1800X, RAM 16GB, GTX 1080.
- Smartphone: Infinix Note 30, RAM 8GB, ROM 256GB.

2. Software

- | | |
|-----------------------|-----------------------------|
| a. Sistem Operasi | : Windows 10 |
| b. Aplikasi / Program | : Android Studio, Photoshop |

4.2 Perancangan

Perencanaan dan definisi mekanisme, struktur, dan komponen umum permainan dikenal sebagai fase perancangan. Tahapan proses pengembangan aplikasi ini sangat penting karena menetapkan kerangka kerja untuk produk akhir.



Gambar 2. Flowchart Aplikasi Resepku

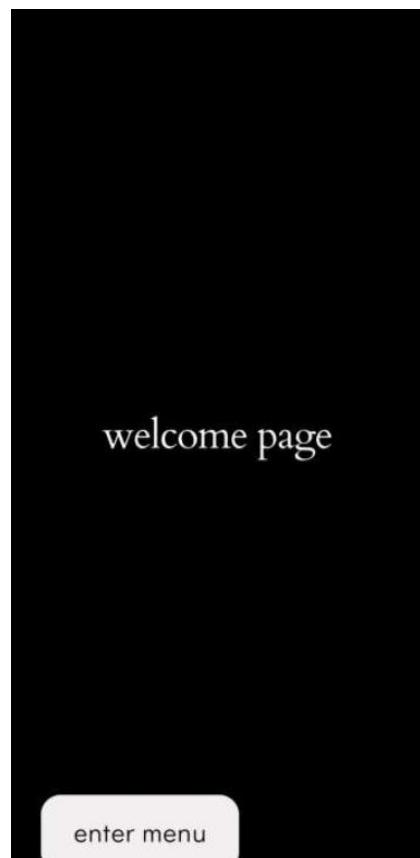
1. Alur Perencanaan

menggunakan metode prototype metode ini meliputi :

Perencanaan (planning) kemudian design aplikasi UI, kemudian perbaikan dan optimasi aplikasi (construction of prototype), kemudian implementasi aplikasi dan komunikasi dan feedback terhadap client/pengguna (deployment, delivery, and feedback), kemudian jika tidak sesuai atau adanya bug maka perlu troubleshooting dan kembali ke perbaikan dan optimisasi, tapi jika berhasil maka aplikasi sudah siap untuk digunakan

2. Desain Aplikasi

Dalam *aplikasi ini “Resepku”* user akan ditampilkan splash screen kemudian klik mulai untuk masuk ke home screen lalu user dapat langsung memilih resep yang ingin di buat setelah memilih akan ditampilkan instruksi dan resep berikut juga jumlah kalora dan porsi dalam sekali masak serta jumlah kalori sajian



Gambar 3. Blueprint desain *Splash Screen*



Gambar 4. Blueprint Home Screen

c) Blueprint desain recipe



Gambar 5. Blueprint desain recipe

d) Blueprint desain recipe



Gambar 6. Blueprint desain recipe screen

4.3 Pengembangan (*Development*)

Fase Pengembangan adalah komponen proses pengembangan aplikasi yang rumit dan memakan waktu, yang mencakup pengkodean, penyempurnaan berulang, dan penggabungan beberapa asset.

1. Pengumpulan Aset

Untuk pengembangan *aplikasi biasanya software house* memperkerjakan para profesional di bidang-bidang yang berhubungan dengan pengembangan . Para artis, para *creator*, dan *UI UX professional* mulai membuat *visual asset*, animasi, *data base*, serta *implementasi desain* yang dibutuhkan untuk pembuatan *aplikasi*, meliputi gambar background Splash Screen, Home Screen, Recipe Screen.

2. Pemrograman

Bahasa pemrograman dan Android Studio digunakan oleh pengembang aplikasi untuk membuat kode yang mengimplementasikan teknik antarmuka pengguna dan fitur lainnya. Pada fase ini, kode dikompilasi dan asset diintegrasikan ke dalam Android Studio.

Berikut adalah Sebagian Source code dan Database dari aplikasi resepku

```
package com.Codingan.Resepku.database;

import androidx.room.DatabaseConfiguration;
import androidx.room.InvalidationTracker;
import androidx.room.RoomOpenHelper;
import androidx.room.RoomOpenHelper.Delegate;
import androidx.room.RoomOpenHelper.ValidationResult;
import androidx.room.util.DBUtil;
import androidx.room.util.TableInfo;
import androidx.room.util.TableInfo.Column;
import androidx.room.util.TableInfo.ForeignKey;
import androidx.room.util.TableInfo.Index;
import androidx.sqlite.db.SupportSQLiteDatabase;
import androidx.sqlite.db.SupportSQLiteOpenHelper;
import androidx.sqlite.db.SupportSQLiteOpenHelper.Callback;
import androidx.sqlite.db.SupportSQLiteOpenHelper.Configuration;
import com.Codingan.Resepku.dao.RecipeDao;
import com.Codingan.Resepku.dao.RecipeDao_Impl;
import java.lang.Override;
import java.lang.String;
import java.lang.SuppressWarnings;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Set;

@SuppressWarnings({"unchecked", "deprecation"})
public final class RecipeDatabase_Impl extends RecipeDatabase {
    private volatile RecipeDao _recipeDao;

    @Override
    protected SupportSQLiteOpenHelper
    createOpenHelper(DatabaseConfiguration configuration) {
        final SupportSQLiteOpenHelper.Callback _openCallback = new
        RoomOpenHelper(configuration, new RoomOpenHelper.Delegate(1) {
            @Override
            public void createAllTables(SupportSQLiteDatabase _db) {
                _db.execSQL("CREATE TABLE IF NOT EXISTS `Recipes` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `dishName` TEXT NOT NULL)");
                _db.execSQL("CREATE TABLE IF NOT EXISTS `CategoryItems` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `idcategory` TEXT NOT NULL, `strcategory` TEXT NOT NULL, `strcategorythumb` TEXT NOT NULL, `strcategorydescription` TEXT NOT NULL)");
                _db.execSQL("CREATE TABLE IF NOT EXISTS `Category` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `categoryItems` TEXT)");
                _db.execSQL("CREATE TABLE IF NOT EXISTS `Meal` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `meals` TEXT)");
                _db.execSQL("CREATE TABLE IF NOT EXISTS `MealsItems` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `idMeal` TEXT NOT NULL, `categoryName` TEXT NOT NULL, `strmeal` TEXT NOT NULL, `strmealthumb` TEXT NOT NULL)");
            }
        });
    }
}
```

Gambar 7.Source Code Aplikasi

```

        _db.execSQL("CREATE TABLE IF NOT EXISTS room_master_table
(id INTEGER PRIMARY KEY,identity hash TEXT)");
        _db.execSQL("INSERT OR REPLACE INTO room_master_table
(id,identity hash) VALUES(42,
'47822e0948dbde2b09d40ef1ece027e6')");
    }

    @Override
    public void dropAllTables(SupportSQLiteDatabase _db) {
        _db.execSQL("DROP TABLE IF EXISTS `Recipes`");
        _db.execSQL("DROP TABLE IF EXISTS `CategoryItems`");
        _db.execSQL("DROP TABLE IF EXISTS `Category`");
        _db.execSQL("DROP TABLE IF EXISTS `Meal`");
        _db.execSQL("DROP TABLE IF EXISTS `MealItems`");
        if (mCallbacks != null) {
            for (int _i = 0, _size = mCallbacks.size(); _i < _size;
            i++) {
                mCallbacks.get(_i).onDestructiveMigration(_db);
            }
        }
    }

    @Override
    protected void onCreate(SupportSQLiteDatabase _db) {
        if (mCallbacks != null) {
            for (int _i = 0, _size = mCallbacks.size(); _i < _size;
            i++) {
                mCallbacks.get(_i).onCreate(_db);
            }
        }
    }

    @Override
    public void onOpen(SupportSQLiteDatabase _db) {
        mDatabase = _db;
        internalInitInvalidationTracker(_db);
        if (mCallbacks != null) {
            for (int _i = 0, _size = mCallbacks.size(); _i < _size;
            i++) {
                mCallbacks.get(_i).onOpen(_db);
            }
        }
    }

    @Override
    public void onPreMigrate(SupportSQLiteDatabase _db) {
        DBUtil.dropFtsSyncTriggers(_db);
    }

    @Override
    public void onPostMigrate(SupportSQLiteDatabase _db) {
    }

    @Override
    protected RoomOpenHelper.ValidationResult
onValidateSchema(SupportSQLiteDatabase db) {
}

```

Gambar 8. Source Code Aplikasi

```

        final HashMap<String, TableInfo.Column> _columnsRecipes =
new HashMap<String, TableInfo.Column>(2);
        _columnsRecipes.put("id", new TableInfo.Column("id",
"INTEGER", true, 1, null, TableInfo.CREATED_FROM_ENTITY));
        _columnsRecipes.put("dishName", new
TableInfo.Column("dishName", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        final HashSet<TableInfo.ForeignKey> _foreignKeysRecipes =
new HashSet<TableInfo.ForeignKey>(0);
        final HashSet<TableInfo.Index> _indicesRecipes = new
HashSet<TableInfo.Index>(0);
        final TableInfo _infoRecipes = new TableInfo("Recipes",
_columnsRecipes, _foreignKeysRecipes, _indicesRecipes);
        final TableInfo _existingRecipes = TableInfo.read(_db,
"Recipes");
        if (! _infoRecipes.equals(_existingRecipes)) {
            return new RoomOpenHelper.ValidationResult(false,
"Recipes(com.Codingan.Resepku.entities.Recipes).\n"
+ " Expected:\n" + _infoRecipes + "\n"
+ " Found:\n" + _existingRecipes);
        }
        final HashMap<String, TableInfo.Column>
_columnsCategoryItems = new HashMap<String, TableInfo.Column>(5);
        _columnsCategoryItems.put("id", new TableInfo.Column("id",
"INTEGER", true, 1, null, TableInfo.CREATED_FROM_ENTITY));
        _columnsCategoryItems.put("idcategory", new
TableInfo.Column("idcategory", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        _columnsCategoryItems.put("strcategory", new
TableInfo.Column("strcategory", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        _columnsCategoryItems.put("strcategorythumb", new
TableInfo.Column("strcategorythumb", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        _columnsCategoryItems.put("strcategorydescription", new
TableInfo.Column("strcategorydescription", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        final HashSet<TableInfo.ForeignKey>
_foreignKeysCategoryItems = new HashSet<TableInfo.ForeignKey>(0);
        final HashSet<TableInfo.Index> _indicesCategoryItems = new
HashSet<TableInfo.Index>(0);
        final TableInfo _infoCategoryItems = new
TableInfo("CategoryItems", _columnsCategoryItems,
_FOREIGN_KEYS_CATEGORYITEMS, _indicesCategoryItems);
        final TableInfo _existingCategoryItems =
TableInfo.read(_db, "CategoryItems");
        if (! _infoCategoryItems.equals(_existingCategoryItems)) {
            return new RoomOpenHelper.ValidationResult(false,
"CategoryItems(com.Codingan.Resepku.entities.CategoryItems).\n"
+ " Expected:\n" + _infoCategoryItems + "\n"
+ " Found:\n" + _existingCategoryItems);
        }
        final HashMap<String, TableInfo.Column> _columnsCategory =
new HashMap<String, TableInfo.Column>(2);
        _columnsCategory.put("id", new TableInfo.Column("id",
"INTEGER", true, 1, null, TableInfo.CREATED_FROM_ENTITY));

```

Gambar 9. Source Code Tabel

```

        _columnsCategory.put("categoryItems", new
TableInfo.Column("categoryItems", "TEXT", false, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        final HashSet<TableInfo.ForeignKey> _foreignKeysCategory =
new HashSet<TableInfo.ForeignKey>(0);
        final HashSet<TableInfo.Index> _indicesCategory = new
HashSet<TableInfo.Index>(0);
        final TableInfo _infoCategory = new TableInfo("Category",
_columnsCategory, _foreignKeysCategory, _indicesCategory);
        final TableInfo _existingCategory = TableInfo.read(_db,
"Category");
        if (! _infoCategory.equals( _existingCategory)) {
            return new RoomOpenHelper.ValidationResult(false,
"Category(com.Codingan.Resepku.entities.Category).\n"
+ " Expected:\n" + _infoCategory + "\n"
+ " Found:\n" + _existingCategory);
        }
        final HashMap<String, TableInfo.Column> _columnsMeal = new
HashMap<String, TableInfo.Column>(2);
        _columnsMeal.put("id", new TableInfo.Column("id",
"INTEGER", true, 1, null, TableInfo.CREATED_FROM_ENTITY));
        _columnsMeal.put("meals", new TableInfo.Column("meals",
"TEXT", false, 0, null, TableInfo.CREATED_FROM_ENTITY));
        final HashSet<TableInfo.ForeignKey> _foreignKeysMeal = new
HashSet<TableInfo.ForeignKey>(0);
        final HashSet<TableInfo.Index> _indicesMeal = new
HashSet<TableInfo.Index>(0);
        final TableInfo _infoMeal = new TableInfo("Meal",
_columnsMeal, _foreignKeysMeal, _indicesMeal);
        final TableInfo _existingMeal = TableInfo.read( db,
"Meal");
        if (! _infoMeal.equals( _existingMeal)) {
            return new RoomOpenHelper.ValidationResult(false,
"Meal(com.Codingan.Resepku.entities.Meal).\n"
+ " Expected:\n" + _infoMeal + "\n"
+ " Found:\n" + _existingMeal);
        }
        final HashMap<String, TableInfo.Column> _columnsMealItems
= new HashMap<String, TableInfo.Column>(5);
        _columnsMealItems.put("id", new TableInfo.Column("id",
"INTEGER", true, 1, null, TableInfo.CREATED_FROM_ENTITY));
        _columnsMealItems.put("idMeal", new
TableInfo.Column("idMeal", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        _columnsMealItems.put("categoryName", new
TableInfo.Column("categoryName", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        _columnsMealItems.put("strmeal", new
TableInfo.Column("strmeal", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        _columnsMealItems.put("strmealthumb", new
TableInfo.Column("strmealthumb", "TEXT", true, 0, null,
TableInfo.CREATED_FROM_ENTITY));
        final HashSet<TableInfo.ForeignKey> _foreignKeysMealItems
= new HashSet<TableInfo.ForeignKey>(0);
        final HashSet<TableInfo.Index> _indicesMealItems = new

```

Gambar 10. Source Code Database

```
    final TableInfo _infoMealItems = new TableInfo("MealItems", columnsMealItems, foreignKeysMealItems, indicesMealItems);
    final TableInfo _existingMealItems = TableInfo.read(_db, "MealItems");
    if (! _infoMealItems.equals( _existingMealItems)) {
        return new RoomOpenHelper.ValidationResult(false,
"MealItems(com.com.Codingan.Resepku.entities.MealsItems).\\n"
        + " Expected:\\n" + _infoMealItems + "\\n"
        + " Found:\\n" + _existingMealItems);
    }
    return new RoomOpenHelper.ValidationResult(true, null);
}
}, "47822e0948dbe2b09d40efelece027e6",
"db31e6d294b7cf3212a670982bf319a");
final SupportSQLiteOpenHelper.Configuration _sqliteConfig =
SupportSQLiteOpenHelper.Configuration.builder(configuration.context)
.name(configuration.name)
.callback(openCallback)
.build();
final SupportSQLiteOpenHelper helper =
configuration.sqliteOpenHelperFactory.create(_sqliteConfig);
return _helper;
}

@Override
protected InvalidationTracker createInvalidationTracker() {
    final HashMap<String, String> _shadowTablesMap = new
HashMap<String, String>(0);
    HashMap<String, Set<String>> _viewTables = new HashMap<String,
Set<String>>(0);
    return new InvalidationTracker(this, _shadowTablesMap,
_viewTables,
"Recipes", "CategoryItems", "Category", "Meal", "MealItems");
}

@Override
public void clearAllTables() {
    super.assertNotMainThread();
    final SupportSQLiteDatabase _db =
super.getOpenHelper().getWritableDatabase();
try {
    super.beginTransaction();
    _db.execSQL("DELETE FROM `Recipes`");
    _db.execSQL("DELETE FROM `CategoryItems`");
    _db.execSQL("DELETE FROM `Category`");
    _db.execSQL("DELETE FROM `Meal`");
    _db.execSQL("DELETE FROM `MealItems`");
    super.setTransactionSuccessful();
} finally {
    super.endTransaction();
    _db.query("PRAGMA wal_checkpoint(FULL)").close();
    if (! _db.inTransaction()) {
        _db.execSQL("VACUUM");
    }
}
}
```

Gambar 11. Source Code Database

```
        }
    }
}

@Override
public RecipeDao recipeDao() {
    if (_recipeDao != null) {
        return _recipeDao;
    } else {
        synchronized(this) {
            if(_recipeDao == null) {
                _recipeDao = new RecipeDao_Impl(this);
            }
        }
        return _recipeDao;
    }
}
```

Gambar 12. Source Code Database

4. Implementasi (*Implementation*)

Dalam pengembangan *aplikasi* dengan metode Prototype, Aplikasi dibuat dan dikembangkan selama fase implementasi menggunakan persyaratan desain yang pertama kali dibuat dan diperbaiki selama tahap desain.

a. *Splashscreen* / tampilan awal

Splashscreen biasanya adalah memberi waktu bagi aplikasi untuk memuat data sekaligus menyajikan desain yang relevan kepada pengguna. Splash screen dapat berupa gambar atau video statis atau animasi yang muncul saat aplikasi diluncurkan dan biasanya ditampilkan dalam waktu singkat sebelum beralih ke menu utama atau layar utama.

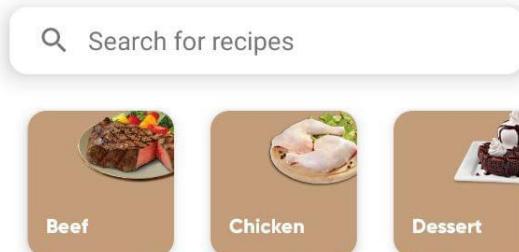


Gambar 13. *Splash Screen*

b. *Main Screen / Menu Utama*

Menu utama adalah antarmuka utama yang pertama kali di perlihatkan kepada pengguna. Menu ini berfungsi sebagai pusat navigasi utama di mana pengguna dapat mengakses berbagai fitur, seperti *memilih makanan, memilih kategori. Dan memilih jenis makanan yang ada.*

All Recipes



Beef Category



Vegetable Shepherd's Pie Traditional

Gambar 14. Main screen

c. *Recipe Screen*

Karena halaman terlalu Panjang maka akan saya bagi menjadi dua bagian atas anda aka di sajikan dengan gambar dari hasil masakannya sebagai referensi bagi pengguna untuk menjadikan gambar tersebut sebagai acuan



Steak Diane

🕒 30 min 🍽 247 cals 🧑 3 persons

Ingredients

Canola Oil 2 tbs
Beef Fillet 4
Beef Stock 1 1/2 cup
Butter 2 tbs
Garlic 2 cloves minced
Challots 1 medium finely diced
Mushrooms 4 oz
Brandy 1/4 cup
Heavy Cream 1/4 cup
Dijon Mustard 1 tbs
Worcestershire Sauce 1 tbs
Tabasco Sauce Dash
Parsley 1 tbs minced
Chives 1 tbs minced
Salt to taste
Pepper to taste

Gambar 15. Recipe Screen

d. Recipe Screen

Dalam recipe screen ini user akan di perlihatkan detail waktu memasak jumlah kalori serta porsi setiap sekali pemasakan serta ada bahan dan intruksinya



Instructions

Heat oil in a 12" skillet over medium-high heat. Season steaks with salt and pepper, and add to skillet; cook, turning once, until browned on both sides and cooked to desired doneness, about 4 to 5 minutes for medium-rare. Transfer steaks to a plate, and set aside. Return skillet to high heat, and add stock; cook until reduced until to 1/2 cup, about 10 minutes. Pour into a bowl, and set aside. Return skillet to heat, and add butter; add garlic and shallots, and cook, stirring, until soft, about 2 minutes. Add mushrooms, and cook, stirring, until they release any liquid and it evaporates and mushrooms begin to brown, about 2 minutes. Add cognac, and light with a match to flambée; cook until flame dies down. Stir in reserved stock, cream, Dijon, Worcestershire, and hot sauce, and then return steaks to skillet; cook, turning in sauce, until warmed through and sauce is thickened, about 4 minutes. Transfer steak to serving plates and stir parsley and chives into sauce; pour sauce over steaks to serve.

[YOUTUBE](#)

Gambar 16. Recipe Screen

4.4 Evaluasi

Dalam pengembangan ini didapatkan hasil aplikasi yang operasional dan relevan dengan kebutuhan Generasi Z. Aplikasi ini memiliki fitur-fitur yang dirancang secara intuitif dan sesuai dengan preferensi target pengguna, seperti antarmuka yang ramah pengguna, integrasi tutorial memasak, serta informasi gizi. Namun, terdapat beberapa area yang dapat ditingkatkan lebih lanjut, seperti penyajian analisis lebih mendalam tentang dampak aplikasi terhadap perubahan perilaku pengguna melalui survei kuantitatif. Penambahan fitur seperti integrasi media sosial dan analitik pengguna dapat meningkatkan daya tarik aplikasi. Secara keseluruhan, aplikasi ini menunjukkan potensi yang signifikan untuk memberikan kontribusi pada gaya hidup sehat di kalangan Generasi Z.

IV. KESIMPULAN

Pengembangan aplikasi ini memberikan solusi praktis untuk mengurangi konsumsi fast food di kalangan Generasi Z. Aplikasi ini menunjukkan potensi yang besar dalam mendorong perubahan pola makan di kalangan generasi muda. Dengan fitur-fitur intuitif seperti tutorial memasak, informasi gizi, dan antarmuka yang ramah pengguna, aplikasi ini mampu menjawab kebutuhan Generasi Z dalam mengadopsi gaya hidup sehat. Feedback dari pengguna awal sangat positif, menunjukkan bahwa aplikasi ini memberikan pengalaman pengguna yang menyenangkan dan membantu mereka lebih termotivasi untuk memasak di rumah. Namun, penelitian lanjutan diperlukan untuk mengevaluasi dampak jangka panjang aplikasi ini terhadap perubahan perilaku konsumsi makanan. Aplikasi ini tidak hanya berfungsi sebagai alat edukasi, tetapi juga sebagai platform yang mendorong kebiasaan makan sehat secara berkelanjutan.

Aplikasi ini sangatlah sederhana, sehingga dapat dikembangkan lebih baik lagi secara kualitas dan bisa berlanjut pada level *semi-professional*. Untuk bisa dikembangkan lebih jauh lagi hingga dapat memberikan kepuasan penuh bagi pengguna terutama pengguna yang ditargetkan yaitu generasi Z.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Sidoarjo dan pihak-pihak yang telah memberikan dukungan dalam perancangan ini.

REFERENSI

- [1] Ajay Irdan Nurwahid, dkk. (2017). "Rancang Bangun Aplikasi Resep Makanan Menggunakan Metode XP". Jurnal Rekayasa Teknologi Nusa Putra.
- [2] Paul Agustinus, dkk. (2005). "Perancangan dan Pembuatan Aplikasi Resep Makanan Berbasis Android".
- [3] Qorina Mar'atus Sholikhah, dkk. (2020). "Sistem Rekomendasi Resep Makanan dengan Collaborative Filtering". JINACS, Vol. 02 No. 02.
- [4] Wikipedia. "Aplikasi Seluler". Diakses pada Januari 2023. [Online].
- [5] Hendri, dkk. (2022). "Penerapan Metode Prototype pada Perancangan Sistem Informasi Penjualan". Jurnal STORAGE.
- [6] S. Aminah. (2018). "Implementasi Model ADDIE pada Game Edukasi". Jurnal Ilmiah Betrik, Vol. 9 No. 03.
- [7] Marzian, F., & Qamal, M. (2017). "GAME RPG Berbasis Desktop". Sisfo Jurnal Ilmiah Sistem Informasi.
- [8] Arifien, A. (2022). "Studi Eksisting Definisi Resep Masakan".

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.