ALAM BAHTERA MASSIVE TEAM

RAKA AGI SAPUTRA ROLE AS A HACKER

Saya saat ini sedang mengikuti Program Studi Independen Kampus Merdeka di Infinite Learning dengan fokus pada Mobile Development dan UI/UX Design menggunakan metode pengembangan berbasis Scrum. Dalam program ini, saya berperan sebagai Hacker di proyek besar (Massive Project), di mana saya bertugas mengembangkan aplikasi mobile. . .

Learn More

#AgsriSynergiPlatform

Agrisynergi Platform

Platform ini Memberdayakan petani jagung skala kecil dan menengah di Indonesia dengan menghubungkan mereka secara langsung ke pasar, memberikan edukasi tentang teknologi pertanian modern, dan memperkuat jaringan komunitas.

Learn More

Project Overview

#AgsriSynergiPlatform

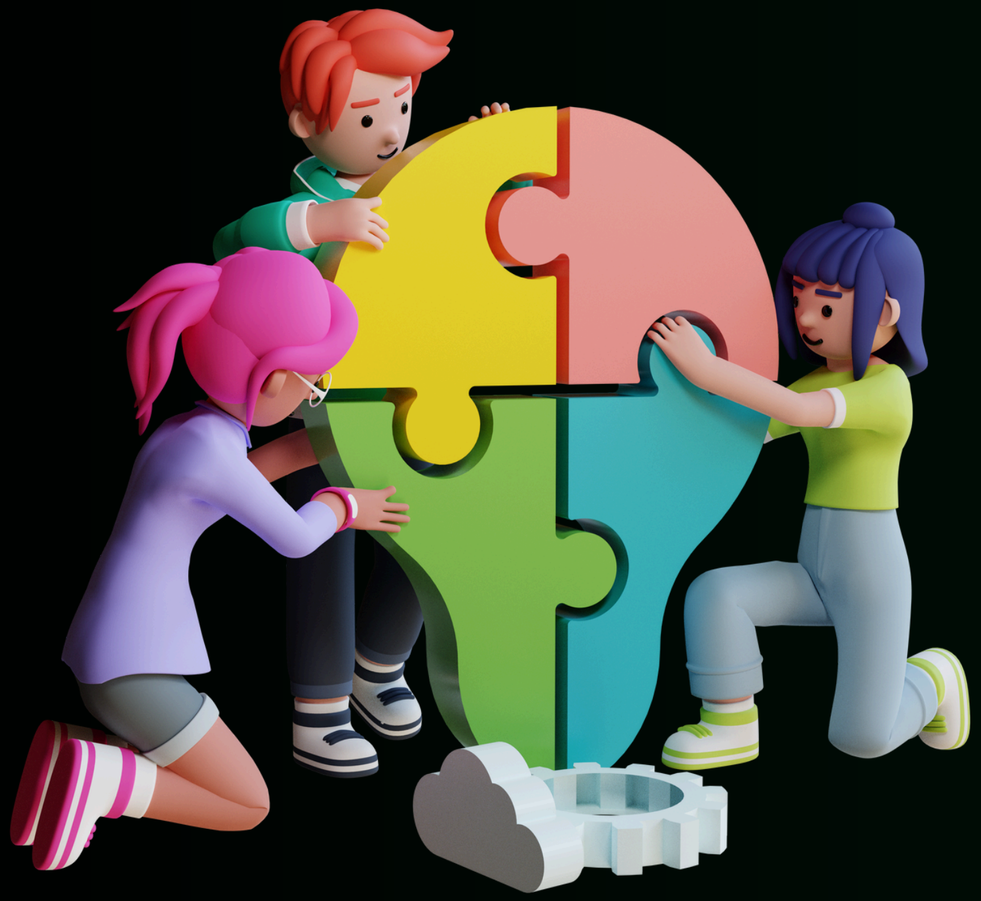
Peran Saya Sebagai Hacker

Sebagai pemrograman aplikasi bagian mobile, saya bertanggung jawab dalam merancang dan mengimplementasikan fitur-fitur kunci, memastikan performa aplikasi yang optimal, serta mengintegrasikan solusi teknologi modern.

Learn More

#Scrum_role

#AgsriSynergiPlatform



My Contribution In the team

1. **Consultation Screen,** Mengintegrasikan Gemini API menggunakan Retrofit, MVVM, dan Coroutines.
2. **Login dan Register dengan Google dan Firebase**, Menerapkan Firebase Authentication dengan Google Login dan SharedPreferences.
3. **Retrofit, MVVM, dan Coroutines,** Mengelola pengambilan data API modular menggunakan Retrofit dan Coroutines.
4. **Room SQLite dan SharedPreferences,** Menyimpan data offline dengan Room dan token menggunakan SharedPreferences
5. Membuat Fitur **Checkout, Beli Sekarang** dan **Keranjang**

1. Use Gemini AI API

Consultation Screen, di mana saya menerapkan Gemini API yang berfungsi sebagai penghubung ke para ahli pertanian, memberikan solusi berbasis data kepada pengguna.

```
package com.example.agrisynergi_mobile.consultant

import android.content.Context
import androidx.compose.runtime.State
import androidx.compose.runtime.mutableStateOf
import androidx.compose.ui.platform.LocalContext
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.google.ai.client.generativeai.GenerativeModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.asStateFlow
import kotlinx.coroutines.launch
import org.osmdroid.library.BuildConfig

class ChatViewModel : ViewModel() {
    private val _userMessage = mutableStateOf<List<UserMessage>>(emptyList())
    private val _textGenerationResult = MutableStateFlow<String?>(null)
    val userMessage: State<List<UserMessage>> = _userMessage
    val textGenerationResult = _textGenerationResult.asStateFlow()

    private val generativeModel = GenerativeModel(
        modelName = "gemini-1.5-flash",
        apiKey = "AIzaSyDrU3CHomozODCEBBrlxdgNiR47_JVdb7o"
//        apiKey = BuildConfig.GEMINI_API_KEY
    )

    var gabungan: String = """ini adalah bagian penting prompt dasar "nama adalah
        |Budi, konsultan pertanian dan jangan sebutkan nama lalu langsung jawab saja pertanyaan :"
    """.trimMargin()
    init {
        gabungan
    }


    fun generateStoryFromApi(prompt: String) {
        viewModelScope.launch(Dispatchers.IO) {
            gabungan += prompt
            try {
                val result = generativeModel.generateContent(gabungan)
                _textGenerationResult.value = result.text
                val textReply = _textGenerationResult.value
                addMessage(UserMessage("Consultant", "consult", textReply.toString(), true))
            } catch (e: Exception) {
                _textGenerationResult.value = "Error: ${e.message}"
            }
        }
    }

    // Fungsi untuk menambah pesan
    fun addMessage(message: UserMessage) {
        _userMessage.value = _userMessage.value.toMutableList().apply { add(message) }
    }

//    fun addMessage(message: UserMessage) {
//        _userMessage.value += message // Membuat daftar baru dengan elemen yang ditambahkan
//    }
}
```

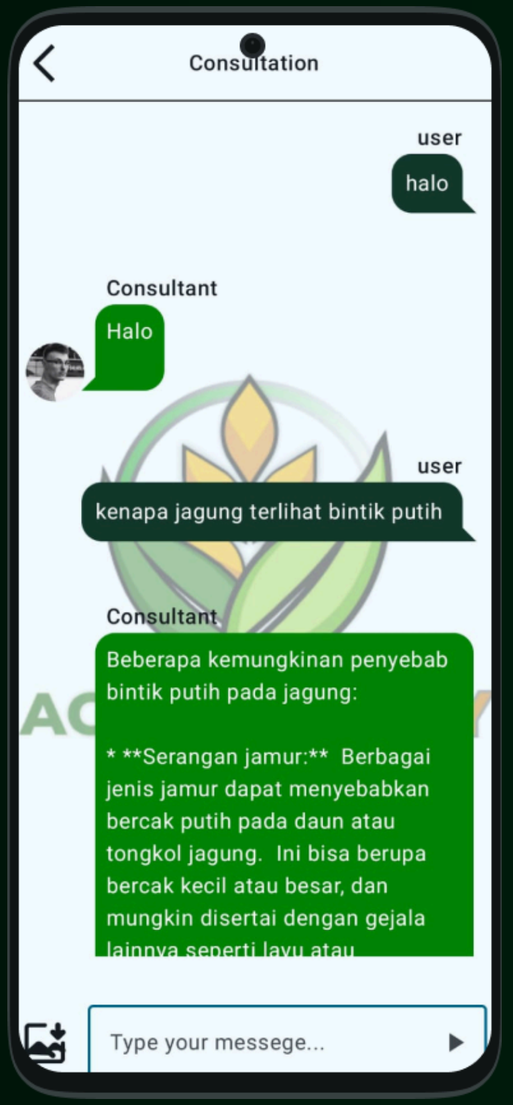
codesnap.de

```
package com.example.agrisynergi_mobile.consultant

import android.net.Uri

data class UserMessage(
    val user: String,
    val type: String, // "text" atau "image"
    val message: String,
    val isUser: Boolean,
    val imageUri: Uri? = null // URI untuk gambar (opsional)
)
```

codesnap.de

```
package com.example.agrisynergi_mobile.consultant

import android.net.Uri

data class UserMessage(package com.example.agrisynergi_mobile.consultant

import androidx.compose.ui.geometry.Size
import androidx.compose.ui.graphics.Outline
import androidx.compose.ui.graphics.Path
import androidx.compose.ui.graphics.Shape
import androidx.compose.ui.layout.BeyondBoundsLayout
import androidx.compose.ui.unit.Density
import androidx.compose.ui.unit.LayoutDirection

class TriangleEdgeShape(val offset: Int) : Shape {

    override fun createOutline(
        size: Size,
        layoutDirection: LayoutDirection,
        density: Density
    ): Outline {
        val trianglePath = Path().apply {
            moveTo(x = 0f, y = size.height-offset)
            lineTo(x = 0f, y = size.height)
            lineTo(x = 0f + offset, y = size.height)
        }
        return Outline.Generic(path = trianglePath)
    }
}


class ReverseTriangleEdgeShape(val offset: Int) : Shape {
    override fun createOutline(
        size: Size,
        layoutDirection: LayoutDirection,
        density: Density
    ): Outline {
        val trianglePath = Path().apply {
            moveTo(x = size.width, y = size.height - offset) // Move to the right side
            lineTo(x = size.width, y = size.height) // Draw to the bottom-right corner
            lineTo(x = size.width - offset, y = size.height) // Draw to the left side
        }
        return Outline.Generic(path = trianglePath)
    }
}

    val user: String,
    val type: String, // "text" atau "image"
    val message: String,
    val isUser: Boolean,
    val imageUri: Uri? = null // URI untuk gambar (opsional)
)
```

codesnap



#MyContribution

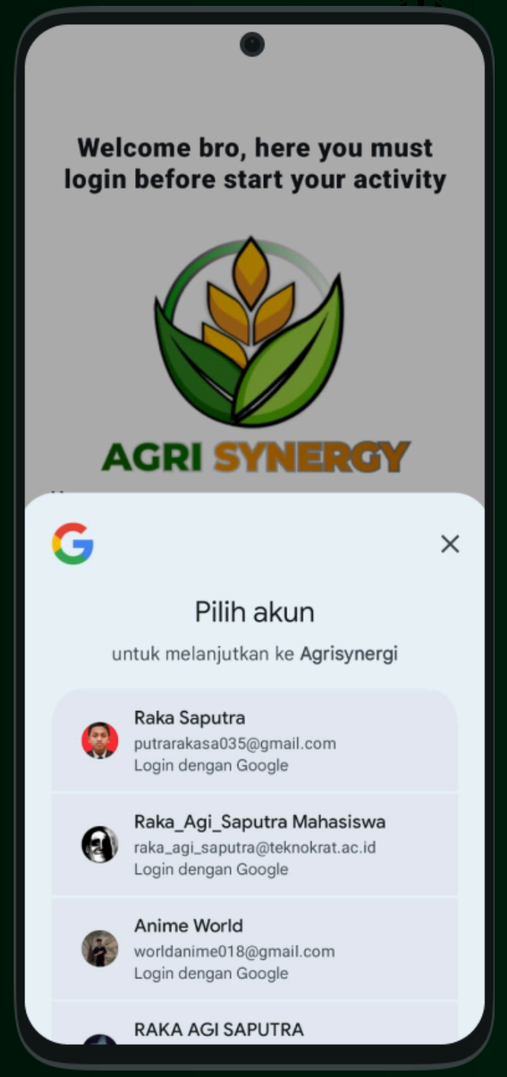
#AgsriSynergiPlatform

2. Use Auth Google With Firebase

Login or Register With Google Screen, mengembangkan fitur Authentication with Google menggunakan Firebase untuk Login dan Register Screen, memastikan proses autentikasi yang aman, cepat, dan mudah digunakan.

```
package com.example.agrisynergi_mobile.auth

import android.content.Context
import android.widget.Toast
import androidx.credentials.CredentialManager
import androidx.credentials.GetCredentialRequest
import androidx.navigation.NavHostController
import com.example.agrisynergi_mobile.R
import com.example.agrisynergi_mobile.navigation.Screen
import com.google.android.gms.auth.GoogleAuthException
import com.google.android.libraries.identity.googleid.GetGoogleIdOption
import com.google.android.libraries.identity.googleid.GoogleIdTokenCredential
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.GoogleAuthProvider
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.launch
import java.io.IOException

private val WEB_ID_CLIENT = "79716735029-f5ok3m4ihfpc65gagbi48p3ndlv8mlkn.apps.googleuserconten

fun AuthWithGoogle(scope: CoroutineScope,context: Context,credentialManager: CredentialManager,
    val googleIdOption: GetGoogleIdOption = GetGoogleIdOption.Builder()
        .setFilterByAuthorizedAccounts(isUserAgri) // Filter akun yang udah terotorisasi aja
        .setServerClientId(com.example.agrisynergi_mobile.auth.WEB_ID_CLIENT)  //  client ID
        .build()

    // Request untuk ambil kredensial dari Credential Manager
    val request = GetCredentialRequest.Builder()
        .addCredentialOption(googleIdOption)
        .build()

    scope.launch { // Jalankan di coroutine biar nggak ngeblok thread utama
        try {
            // Ambil kredensial dari Credential Manager
            val result = credentialManager.getCredential(
                context = context,
                request = request
            )

            // Ambil Google ID Token dari kredensial yang didapet
            val credential = result.credential
            val googleIdTokenCredential =
                GoogleIdTokenCredential.createFrom(credential.data)

            val googleIdToken = googleIdTokenCredential.idToken

            // Login ke Firebase pake Google ID Token
            val firebaseCredential =
                GoogleAuthProvider.getCredential(googleIdToken, null)

            auth.signInWithCredential(firebaseCredential)
                .addOnCompleteListener { task ->
                    if (task.isSuccessful) {
                        // Kalau login sukses, langsung ke halaman utama (Beranda)
                        navController.popBackStack() // Balik ke screen sebelumnya
                        navController.navigate(Screen.Beranda.route) // Pindah ke Beranda
```

```
composable(Screen.Regist.route) {
                RegisterScreen(
                    navController = navController,
                    registerViewModel = registerViewModel,
                    registerWithGoogle = {
                        AuthWithGoogle(
                            scope = scope,
                            context = context,
                            credentialManager = credentialManager,
                            auth = auth,
                            navController = navController,
                            isUserAgri = false
                        )
                    }
                )
            }
```



#MyContribution

#AgsriSynergiPlatform

```
package com.example.agrisynergi_mobile.auth

import android.content.Context
import android.widget.Toast
import androidx.credentials.CredentialManager
import androidx.credentials.GetCredentialRequest
import androidx.navigation.NavHostController
import com.example.agrisynergi_mobile.R
import com.example.agrisynergi_mobile.navigation.Screen
import com.google.android.gms.auth.GoogleAuthException
import com.google.android.libraries.identity.googleid.GetGoogleIdOption
import com.google.android.libraries.identity.googleid.GoogleIdTokenCredential
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.GoogleAuthProvider
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.launch
import java.io.IOException

private val WEB_ID_CLIENT = "79716735029-f5ok3m4ihfpc65gagbi48p3ndlv8mlkn.apps.googleuserconten

fun AuthWithGoogle(scope: CoroutineScope,context: Context,credentialManager: CredentialManager,
    val googleIdOption: GetGoogleIdOption = GetGoogleIdOption.Builder()
        .setFilterByAuthorizedAccounts(isUserAgri) // Filter akun yang udah terotorisasi aja
        .setServerClientId(com.example.agrisynergi_mobile.auth.WEB_ID_CLIENT)  //  client ID
        .build()

    // Request untuk ambil kredensial dari Credential Manager
    val request = GetCredentialRequest.Builder()
        .addCredentialOption(googleIdOption)
        .build()

    scope.launch { // Jalankan di coroutine biar nggak ngeblok thread utama
        try {
            // Ambil kredensial dari Credential Manager
            val result = credentialManager.getCredential(
                context = context,
                request = request
            )

            // Ambil Google ID Token dari kredensial yang didapet
            val credential = result.credential
            val googleIdTokenCredential =
                GoogleIdTokenCredential.createFrom(credential.data)

            val googleIdToken = googleIdTokenCredential.idToken

            // Login ke Firebase pake Google ID Token
            val firebaseCredential =
                GoogleAuthProvider.getCredential(googleIdToken, null)

            auth.signInWithCredential(firebaseCredential)
                .addOnCompleteListener { task →
                    if (task.isSuccessful) {
                        // Kalau login sukses, langsung ke halaman utama (Beranda)
                        navController.popBackStack() // Balik ke screen sebelumnya
                        navController.navigate(Screen.Beranda.route) // Pindah ke Beranda
```

3. Use Retrofit, MVVM, and Couroutine

Login or Register With Username and Paswword, menggunakan Retrofit, Kotlin Coroutines, dan menerapkan arsitektur MVVM (Model-View-ViewModel). Retrofit digunakan untuk berkomunikasi dengan API melalui metode HTTP seperti POST, memungkinkan pengiriman data pengguna seperti nama, email, dan password untuk proses registrasi, serta email dan password untuk login. Arsitektur MVVM memastikan pemisahan tanggung jawab antara logika bisnis dan tampilan, menjadikan kode lebih modular, mudah dikelola, dan efisie

#MyContribution

#AgsriSynergiPlatform

4. Use Room SQLite, SharedPreferences

SharedPreferences, Library Room, Runtime menyediakan API utama untuk operasi database seperti menyimpan, membaca, memperbarui, dan menghapus data secara lokal.

```
package com.example.agrisynergi_mobile.s

import androidx.room.*

@Dao
interface FarmerDao {
    @Insert(onConflict = OnConflictStrat
    suspend fun insertFarmer(farmer: Far

    @Query("SELECT * FROM farmers")
    suspend fun getAllFarmers(): List<Fa

    @Query("SELECT * FROM farmers WHERE
    suspend fun getFarmerById(id: Int):

    @Update
    suspend fun updateFarmer(farmer: Far

    @Delete
    suspend fun deleteFarmer(farmer: Far
}
```

```
package com.example.package com.example.agrisynergi_mobile.storage

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "farmers")
data class FarmerEntity(
    @PrimaryKey(autoGenerate = true) val id: Int = 0,
    val name: String,
    val location: String,
    val cropType: String,
    val farmSize: Double // luas lahan dalam hektar
)
.storage
```

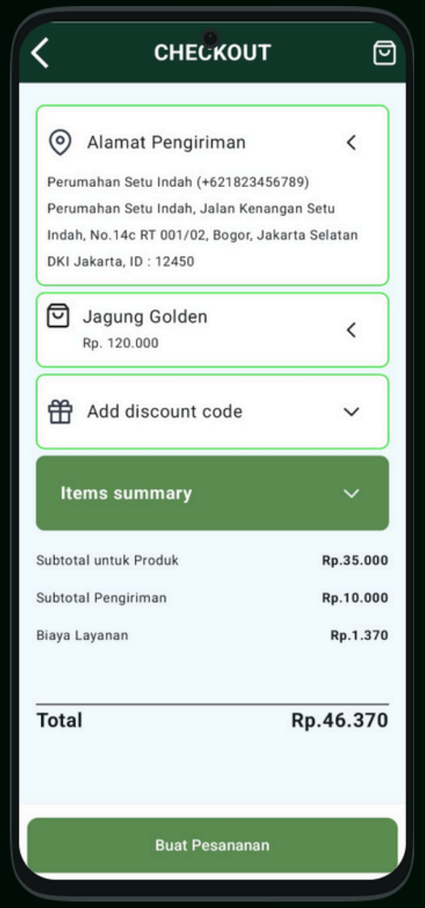
```
package com.example.package com.example.agrisynerg

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "farmers")
data class FarmerEntity(
    @PrimaryKey(autoGenerate = true) val id: Int =
    val name: String,
    val location: String,
    val cropType: String,
    val farmSize: Double // luas lahan dalam hekta
)
.storage

import androidx.room.*

@Dao
interface FarmerDao {
    @Insert(onConflict = OnConflictStrategy.REPLAC
    suspend fun insertFarmer(farmer: FarmerEntity)

    @Query("SELECT * FROM farmers")
    suspend fun getAllFarmers(): List<FarmerEntity

    @Query("SELECT * FROM farmers WHERE id = :id")
    suspend fun getFarmerById(id: Int): FarmerEnti

    @Update
    suspend fun updateFarmer(farmer: FarmerEntity)

    @Delete
    suspend fun deleteFarmer(farmer: FarmerEntity)
}
```

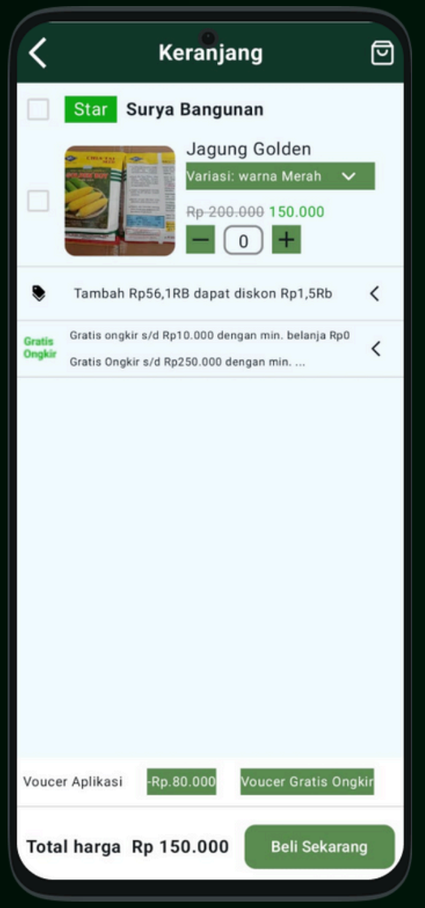
#MyContribution

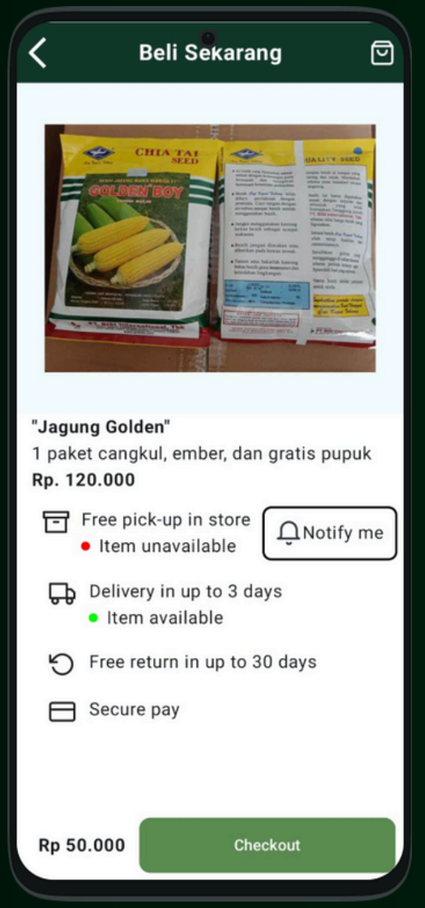
#MyContribution

4. Membuat Fitur Checkout, Beli Sekarang dan Keranjang

Keranjang, menggunakan Room SQLite untuk menyimpan data lokal, memungkinkan pengguna menambah, mengubah, atau menghapus produk, dengan pembaruan otomatis di UI berkat arsitektur MVVM. **Beli Sekarang,** memungkinkan pembelian langsung tanpa keranjang, dengan data dikirim ke API menggunakan Retrofit dan Coroutines untuk memastikan proses berjalan asinkron dan responsif. **Fitur Checkout** membantu pengguna mengonfirmasi pembelian dengan menyimpan data sementara di SharedPreferences dan mengirim detail transaksi melalui API.







#MyContribution

#MyContribution

Challange and Solution

Challange, Menerapkan arsitektur yang modular dan terpisah antara logika bisnis dan UI menjadi tantangan besar dalam pengembangan aplikasi yang kompleks. Salah satu tantangan utamanya adalah bagaimana mengintegrasikan data dari API menggunakan Retrofit ke dalam aplikasi dengan cara yang terstruktur dan mudah dikelola, tanpa membebani UI atau membuat kode sulit dipahami.

Challange, Saya menerapkan arsitektur MVVM (Model-View-ViewModel) untuk membagi tanggung jawab aplikasi dengan jelas. Pendekatan ini mempermudah pengelolaan data, meningkatkan skalabilitas aplikasi, dan menjadikan kode lebih modular, terstruktur, serta mudah dipelihara. Saya juga terus belajar melalui dokumentasi resmi dan studi kasus untuk mengoptimalkan penggunaan MVV**M.**

Firebase Kotlin MySQL C++

#Scrum_role #AgsriSynergiPlatform

Thank You

FOR YOUR ATTENTION

LET’S CONNECT!

@rakasa_22

@rakasa_22