

## MODUL 7



Disusun Oleh :  
**Nama : Ganes Gemi Putra**  
**Kelas : SE-07-02**  
**NIM : (2311104075)**

**Dosen : YUDHA ISLAMI SULISTYA S.Kom. M.Cs**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2025**

## 1. Pendahuluan

Tugas praktikum ini bertujuan untuk membuat satu proyek aplikasi e-commerce sederhana menggunakan Flutter yang menampilkan beberapa produk dan halaman, dengan menerapkan class model serta navigasi halaman. Aplikasi ini dibangun menggunakan bahasa pemrograman Dart dan framework Flutter, yang memungkinkan pengembangan aplikasi mobile lintas platform (Android dan iOS).

## 2. Deskripsi Proyek

Proyek ini adalah aplikasi e-commerce bernama "Toko Kanesuuu\_" yang menampilkan produk-produk Apple seperti iPhone, MacBook, dan iPad. Aplikasi ini memiliki fitur-fitur dasar e-commerce seperti:

- Daftar produk dengan gambar, harga, rating, dan stok.
- Pencarian produk.
- Filter berdasarkan kategori.
- Halaman detail produk.
- Favorit produk.
- Keranjang belanja.
- Halaman akun.

## 3. Struktur Proyek

Struktur proyek Flutter ini terdiri dari:

- **lib/main.dart**: Entry point aplikasi, mengatur tema dan navigasi utama.
- **lib/product.dart**: Class model Product yang mendefinisikan struktur data produk.
- **lib/home\_page.dart**: Halaman utama yang menampilkan daftar produk, slider promo, pencarian, dan filter kategori.
- **lib/product\_detail\_page.dart**: Halaman detail produk.
- **lib/favorites\_page.dart**: Halaman produk favorit.
- **lib/cart\_page.dart**: Halaman keranjang belanja.
- **lib/account\_page.dart**: Halaman akun pengguna.
- **assets/**: Folder untuk gambar produk (iPhone, MacBook, iPad).

## 4. Implementasi Class Model

Class Product di lib/product.dart didefinisikan sebagai berikut:

```
class Product {  
    final int id;  
    final String name;  
    final double price;  
    final String description;  
    final String imageUrl;  
    final double rating;  
    final int stock;  
    final String category;  
    final bool isNew;  
    final bool isHot;  
    final bool isOnSale;  
    final double? originalPrice;  
  
    const Product({  
        required this.id,  
        required this.name,  
        required this.price,  
        required this.description,  
        required this.imageUrl,  
        required this.rating,  
        required this.stock,  
        required this.category,  
    });
```

```
this.isNew = false,  
this.isHot = false,  
this.isOnSale = false,  
this.originalPrice,  
});  
  
factory Product.fromJson(Map<String, dynamic> json) {  
    return Product(  
        id: json['id'],  
        name: json['name'],  
        price: json['price'].toDouble(),  
        description: json['description'],  
        imageUrl: json['imageUrl'],  
        rating: json['rating'].toDouble(),  
        stock: json['stock'],  
        category: json['category'] ?? 'General',  
        isNew: json['isNew'] ?? false,  
        isHot: json['isHot'] ?? false,  
        isOnSale: json['isOnSale'] ?? false,  
        originalPrice: json['originalPrice']?.toDouble(),  
    );  
}  
  
bool get isFavorite => false; // This will be managed by state management  
}
```

Class ini menggunakan constructor const untuk efisiensi, factory method untuk parsing JSON, dan properti boolean untuk status produk (baru, panas, diskon).

## 5. Navigasi Halaman

Navigasi halaman diimplementasikan menggunakan Navigator.push dan BottomNavigationBar. Dari halaman utama (HomePage), pengguna dapat:

- Menekan kartu produk untuk masuk ke ProductDetailPage.
- Menggunakan bottom navigation untuk berpindah ke halaman Favorit, Keranjang, atau Akun.
- Navigasi ini memungkinkan perpindahan halus antar halaman dengan passing data seperti objek Product.

## 6. Fitur Utama

- **Slider Promo:** Menggunakan PageView dengan auto-slide setiap 3 detik.
- **Pencarian dan Filter:** TextField untuk pencarian, FilterChip untuk kategori.
- **Grid Produk:** GridView.builder untuk menampilkan produk dalam grid 2 kolom.
- **Interaksi:** Tombol favorit, tambah ke keranjang, dengan Snackbar feedback.
- **Tema:** Gradient background, custom TextTheme, dan CardTheme untuk konsistensi UI.

## 7. Kesimpulan

Proyek ini berhasil memenuhi instruksi tugas dengan menerapkan class model Product untuk struktur data dan navigasi halaman menggunakan Flutter. Aplikasi ini dapat diperluas dengan menambahkan state management (seperti Provider atau Bloc) untuk persistensi data favorit dan keranjang, serta integrasi API untuk dinamis.



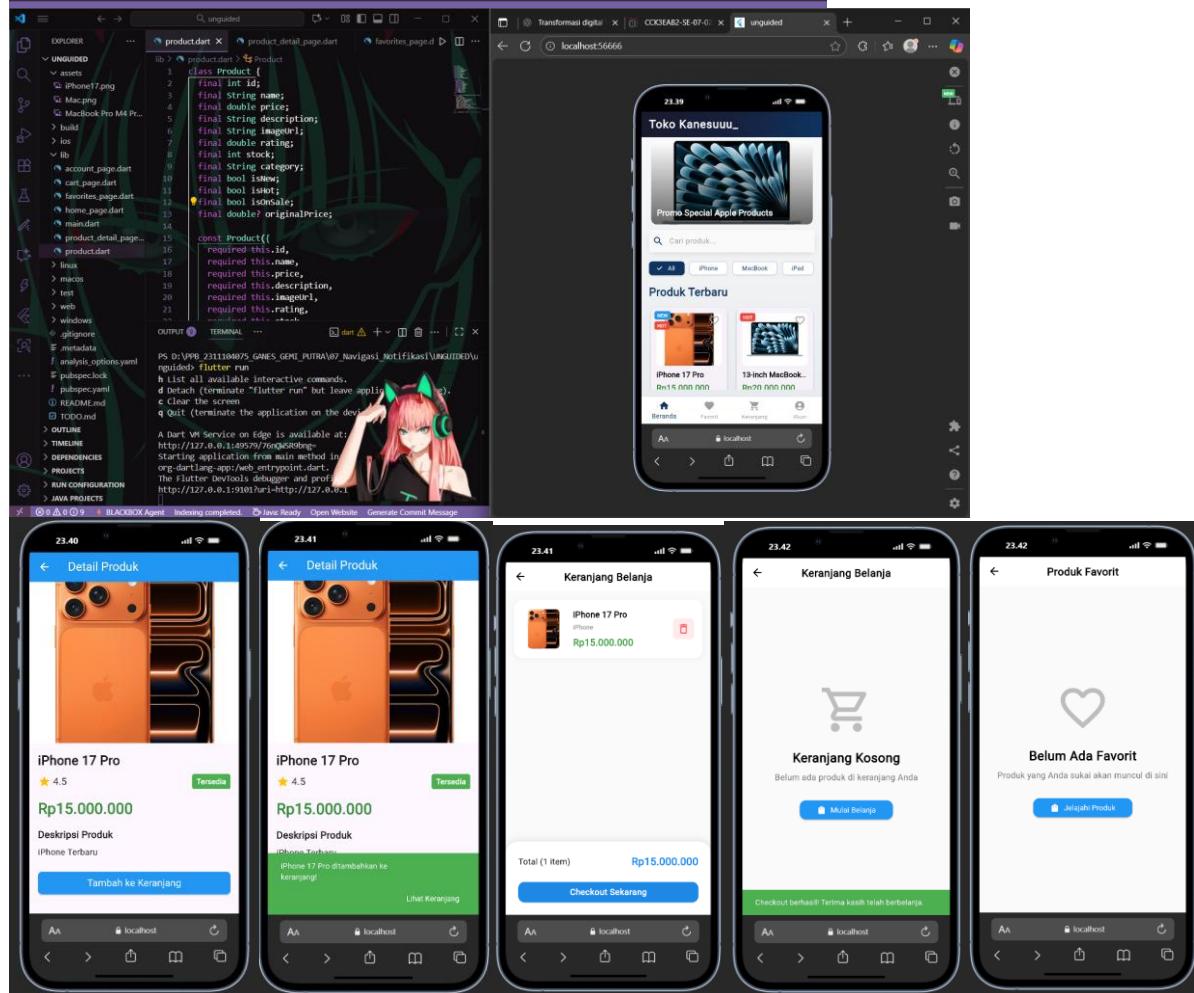
## 8. Screenshot dan Demo

```

OUTPUT 9 TERMINAL ...
PS D:\PPB_2311104075_GANES_GEMI_PUTRA\07_Na
nguided> flutter run
h List all available interactive commands.
d Detach (terminate "flutter run" but leave
c Clear the screen
q Quit (terminate the application on the de

A Dart VM Service on Edge is available at:
http://127.0.0.1:49579/76nQWSR9bng=
Starting application from main method in
org-dartlang-app:/web_entrypoint.dart.
The Flutter DevTools debugger and profi
http://127.0.0.1:9101?uri=http://127.0.0.1

```





*source code Lengkap :*

**account\_page.dart**

```
import 'package:flutter/material.dart';
import 'home_page.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Toko Kanesuuu_',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        fontFamily: 'Roboto',
        textTheme: TextTheme(
          headlineLarge: TextStyle(
            fontSize: 32,
            fontWeight: FontWeight.w700,
            letterSpacing: -0.5,
          ),
          headlineMedium: TextStyle(
            fontSize: 28,
            fontWeight: FontWeight.w600,
            letterSpacing: -0.25,
          ),
          headlineSmall: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.w600,
          ),
        ),
      ),
    );
  }
}
```

```
letterSpacing: 0,  
),  
titleLarge: TextStyle(  
  fontSize: 22,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0,  
,  
titleMedium: TextStyle(  
  fontSize: 16,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.15,  
,  
titleSmall: TextStyle(  
  fontSize: 14,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.1,  
,  
bodyLarge: TextStyle(  
  fontSize: 16,  
  fontWeight: FontWeight.w400,  
  letterSpacing: 0.5,  
,  
bodyMedium: TextStyle(  
  fontSize: 14,  
  fontWeight: FontWeight.w400,  
  letterSpacing: 0.25,  
,  
bodySmall: TextStyle(  
  fontSize: 12,  
  fontWeight: FontWeight.w400,  
  letterSpacing: 0.4,  
,  
labelLarge: TextStyle(  
  fontSize: 14,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.1,  
,  
labelSmall: TextStyle(  
  fontSize: 11,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.5,  
,  
),  
cardTheme: CardThemeData(  
  elevation: 4,  
  shape: RoundedRectangleBorder(  
    borderRadius: BorderRadius.circular(16),  
,  
),  
elevatedButtonTheme: ElevatedButtonThemeData(  
  style: ElevatedButton.styleFrom(  
)
```

```
elevation: 2,  
shape: RoundedRectangleBorder(  
    borderRadius: BorderRadius.circular(12),  
,  
padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),  
,  
,  
inputDecorationTheme: InputDecoration(  
    border: OutlineInputBorder(  
        borderRadius: BorderRadius.circular(12),  
,  
    ),  
    filled: true,  
    fillColor: Colors.grey[50],  
,  
    chipTheme: ChipThemeData(  
        shape: RoundedRectangleBorder(  
            borderRadius: BorderRadius.circular(8),  
,  
,  
,  
    ),  
    ),  
    ),  
    home: HomePage(),  
    debugShowCheckedModeBanner: false,  
,  
,  
);  
}  
}
```

### cart\_page.dart

```
import 'package:flutter/material.dart';  
import 'package:intl/intl.dart';  
import 'product.dart';  
import 'product_detail_page.dart';  
  
class CartPage extends StatefulWidget {  
    final List<Product> cartProducts;  
    final Function(Product) onRemoveFromCart;  
    final Function(Product) onAddToCart;  
    final bool isDarkMode;  
  
    const CartPage({  
        super.key,  
        required this.cartProducts,  
        required this.onRemoveFromCart,  
        required this.onAddToCart,  
        required this.isDarkMode,  
    });  
  
    @override  
    _CartPageState createState() => _CartPageState();  
}
```

```
class _CartPageState extends State<CartPage> with TickerProviderStateMixin {
    late AnimationController _animationController;
    late Animation<double> _fadeAnimation;

    @override
    void initState() {
        super.initState();
        _animationController = AnimationController(
            duration: const Duration(milliseconds: 600),
            vsync: this,
        );
        _fadeAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(
            CurvedAnimation(parent: _animationController, curve: Curves.easeOut),
        );
        _animationController.forward();
    }

    @override
    void dispose() {
        _animationController.dispose();
        super.dispose();
    }

    double get totalPrice {
        return widget.cartProducts.fold(0.0, (sum, product) => sum + product.price);
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: widget.isDarkMode ? Colors.grey[900] : Colors.grey[50],
            appBar: AppBar(
                title: Text(
                    'Keranjang Belanja',
                    style: TextStyle(
                        fontWeight: FontWeight.w600,
                        fontSize: 20,
                        letterSpacing: 0.5,
                    ),
                ),
                backgroundColor: widget.isDarkMode ? Colors.grey[900] : Colors.white,
                foregroundColor: widget.isDarkMode ? Colors.white : Colors.black,
                elevation: 0,
                centerTitle: true,
            ),
            body: FadeTransition(
                opacity: _fadeAnimation,
                child: widget.cartProducts.isEmpty
                    ? _buildEmptyCart()
                    : _buildCartContent(),
            )
        );
    }
}
```

```
        ),  
    );  
}  
  
Widget _buildEmptyCart() {  
    return Center(  
        child: Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: [  
                Icon(  
                    Icons.shopping_cart_outlined,  
                    size: 100,  
                    color: widget.isDarkMode ? Colors.grey[600] : Colors.grey[400],  
                ),  
                SizedBox(height: 24),  
                Text(  
                    'Keranjang Kosong',  
                    style: TextStyle(  
                        fontSize: 24,  
                        fontWeight: FontWeight.w600,  
                        color: widget.isDarkMode ? Colors.white : Colors.black,  
                        letterSpacing: 0.5,  
                    ),  
                ),  
                SizedBox(height: 8),  
                Text(  
                    'Belum ada produk di keranjang Anda',  
                    style: TextStyle(  
                        fontSize: 16,  
                        color: widget.isDarkMode ? Colors.grey[400] : Colors.grey[600],  
                    ),  
                ),  
                SizedBox(height: 32),  
                ElevatedButton.icon(  
                    onPressed: () => Navigator.pop(context),  
                    icon: Icon(Icons.shopping_bag),  
                    label: Text('Mulai Belanja'),  
                    style: ElevatedButton.styleFrom(  
                        backgroundColor: Colors.blue,  
                        foregroundColor: Colors.white,  
                        padding: EdgeInsets.symmetric(horizontal: 32, vertical: 16),  
                        shape: RoundedRectangleBorder(  
                            borderRadius: BorderRadius.circular(12),  
                        ),  
                        elevation: 2,  
                    ),  
                ),  
            ],  
        ),  
    );  
}
```

```
Widget _buildCartContent() {
    return Column(
        children: [
            Expanded(
                child: ListView.builder(
                    padding: EdgeInsets.all(16),
                    itemCount: widget.cartProducts.length,
                    itemBuilder: (context, index) {
                        final product = widget.cartProducts[index];
                        return _buildCartItem(product, index);
                    },
                ),
            ),
            _buildCartSummary(),
        ],
    );
}

Widget _buildCartItem(Product product, int index) {
    return Container(
        margin: EdgeInsets.only(bottom: 16),
        decoration: BoxDecoration(
            color: widget.isDarkMode ? Colors.grey[800] : Colors.white,
            borderRadius: BorderRadius.circular(16),
            boxShadow: [
                BoxShadow(
                    color: Colors.black.withOpacity(0.05),
                    blurRadius: 8,
                    offset: Offset(0, 2),
                ),
            ],
        ),
        child: Padding(
            padding: EdgeInsets.all(16),
            child: Row(
                children: [
                    GestureDetector(
                        onTap: () {
                            Navigator.push(
                                context,
                                MaterialPageRoute(
                                    builder: (context) => ProductDetailPage(
                                        product: product,
                                        onFavoriteToggle: (p) {}, // Placeholder since cart doesn't handle favorites
                                        onAddToCart: widget.onAddToCart,
                                        isFavorite: false, // Placeholder
                                    ),
                            ),
                        ),
                );
            ],
        ),
    );
}
```





```
});

ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
        content: Text('${product.name} dihapus dari keranjang!'),
        backgroundColor: Colors.red,
        duration: Duration(seconds: 2),
    ),
);
},
icon: Icon(
    Icons.delete_outline,
    color: Colors.red[400],
),
style: IconButton.styleFrom(
    backgroundColor: Colors.red[50],
    shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(8),
    ),
),
),
),
],
),
),
),
);
}

Widget _buildCartSummary() {
    return Container(
        padding: EdgeInsets.all(24),
        decoration: BoxDecoration(
            color: widget.isDarkMode ? Colors.grey[800] : Colors.white,
            borderRadius: BorderRadius.only(
                topLeft: Radius.circular(24),
                topRight: Radius.circular(24),
            ),
            boxShadow: [
                BoxShadow(
                    color: Colors.black.withOpacity(0.1),
                    blurRadius: 12,
                    offset: Offset(0, -2),
                ),
            ],
        ),
        child: Column(
            children: [
                Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                        Text(
                            'Total ${widget.cartProducts.length} item${widget.cartProducts.length > 1 ? 's' : ""}',
                            style: TextStyle(

```

```
        fontSize: 16,
        fontWeight: FontWeight.w500,
        color: widget.isDarkMode ? Colors.white : Colors.black,
    ),
),
Text(
'Rp${NumberFormat.decimalPattern('id').format(totalPrice)}',
style: TextStyle(
fontSize: 20,
fontWeight: FontWeight.w700,
color: Colors.blue[600],
),
),
],
),
SizedBox(height: 24),
SizedBox(
width: double.infinity,
child: ElevatedButton(
onPressed: () {
ScaffoldMessenger.of(context).showSnackBar(
SnackBar(
content: Text('Checkout berhasil! Terima kasih telah berbelanja.'),
backgroundColor: Colors.green,
duration: Duration(seconds: 3),
),
);
// Clear cart after checkout
setState(() {
widget.cartProducts.clear();
});
},
style: ElevatedButton.styleFrom(
backgroundColor: Colors.blue[600],
foregroundColor: Colors.white,
padding: EdgeInsets.symmetric(vertical: 16),
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(12),
),
elevation: 2,
shadowColor: Colors.blue.withOpacity(0.3),
),
child: Text(
'Checkout Sekarang',
style: TextStyle(
fontSize: 16,
fontWeight: FontWeight.w600,
letterSpacing: 0.5,
),
),
),
),
```

```
        ),  
        ],  
        ),  
    );  
}  
}
```

### **favorites\_page.dart**

```
import 'package:flutter/material.dart';  
import 'package:intl/intl.dart';  
import 'product.dart';  
import 'product_detail_page.dart';  
  
class FavoritesPage extends StatefulWidget {  
    final List<Product> favoriteProducts;  
    final Function(Product) onFavoriteToggle;  
    final Function(Product) onAddToCart;  
    final bool isDarkMode;  
  
    const FavoritesPage({  
        super.key,  
        required this.favoriteProducts,  
        required this.onFavoriteToggle,  
        required this.onAddToCart,  
        required this.isDarkMode,  
    });  
  
    @override  
    _FavoritesPageState createState() => _FavoritesPageState();  
}  
  
class _FavoritesPageState extends State<FavoritesPage> with TickerProviderStateMixin {  
    late AnimationController _animationController;  
    late Animation<double> _fadeAnimation;  
  
    @override  
    void initState() {  
        super.initState();  
        _animationController = AnimationController(  
            duration: const Duration(milliseconds: 600),  
            vsync: this,  
        );  
        _fadeAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(  
            CurvedAnimation(parent: _animationController, curve: Curves.easeOut),  
        );  
        _animationController.forward();  
    }  
  
    @override
```

```
void dispose() {
    _animationController.dispose();
    super.dispose();
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: widget.isDarkMode ? Colors.grey[900] : Colors.grey[50],
        appBar: AppBar(
            title: Text(
                'Produk Favorit',
                style: TextStyle(
                    fontWeight: FontWeight.w600,
                    fontSize: 20,
                    letterSpacing: 0.5,
                ),
            ),
            backgroundColor: widget.isDarkMode ? Colors.grey[900] : Colors.white,
            foregroundColor: widget.isDarkMode ? Colors.white : Colors.black,
            elevation: 0,
            centerTitle: true,
        ),
        body: FadeTransition(
            opacity: _fadeAnimation,
            child: widget.favoriteProducts.isEmpty
                ? _buildEmptyFavorites()
                : _buildFavoritesContent(),
        ),
    );
}

Widget _buildEmptyFavorites() {
    return Center(
        child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                Icon(
                    Icons.favorite_border,
                    size: 100,
                    color: widget.isDarkMode ? Colors.grey[600] : Colors.grey[400],
                ),
                SizedBox(height: 24),
                Text(
                    'Belum Ada Favorit',
                    style: TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.w600,
                        color: widget.isDarkMode ? Colors.white : Colors.black,
                        letterSpacing: 0.5,
                    ),
                ),
            ],
        ),
    );
}
```

```
        ),
        SizedBox(height: 8),
        Text(
            'Produk yang Anda sukai akan muncul di sini',
            style: TextStyle(
                fontSize: 16,
                color: widget.isDarkMode ? Colors.grey[400] : Colors.grey[600],
            ),
            textAlign: TextAlign.center,
        ),
        SizedBox(height: 32),
        ElevatedButton.icon(
            onPressed: () => Navigator.pop(context),
            icon: Icon(Icons.shopping_bag),
            label: Text('Jelajahi Produk'),
            style: ElevatedButton.styleFrom(
                backgroundColor: Colors.blue,
                foregroundColor: Colors.white,
                padding: EdgeInsets.symmetric(horizontal: 32, vertical: 16),
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                ),
                elevation: 2,
            ),
        ),
    ],
),
),
);
},
);
}
}

Widget _buildFavoritesContent() {
return Padding(
padding: EdgeInsets.all(16),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
    '${widget.favoriteProducts.length} Produk Favorit',
    style: TextStyle(
        fontSize: 18,
        fontWeight: FontWeight.w500,
        color: widget.isDarkMode ? Colors.white : Colors.black,
    ),
),
SizedBox(height: 16),
Expanded(
child: GridView.builder(
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,
crossAxisSpacing: 16,
mainAxisSpacing: 16,
```

```
        childAspectRatio: 0.7,
    ),
    itemCount: widget.favoriteProducts.length,
    itemBuilder: (context, index) {
        final product = widget.favoriteProducts[index];
        return _buildFavoriteCard(product);
    },
),
),
],
),
);
}

Widget _buildFavoriteCard(Product product) {
    return Card(
        elevation: 4,
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(16),
        ),
        child: InkWell(
            onTap: () {
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) => ProductDetailPage(
                            product: product,
                            onFavoriteToggle: widget.onFavoriteToggle,
                            onAddToCart: widget.onAddToCart,
                            isFavorite: true,
                        ),
                    ),
                );
            },
        );
    },
    borderRadius: BorderRadius.circular(16),
    child: Stack(
        children: [
            Padding(
                padding: const EdgeInsets.all(12.0),
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.start,
                    children: [
                        Expanded(
                            child: GestureDetector(
                                onTap: () {
                                    Navigator.push(
                                        context,
                                        MaterialPageRoute(
                                            builder: (context) => ProductDetailPage(
                                                product: product,
                                                onFavoriteToggle: widget.onFavoriteToggle,
                                            ),
                                        ),
                                    );
                                },
                            ),
                        ),
                    ],
                ),
            ),
        ],
    ),
}
```





```
)  
else  
Text(  
'Rp${NumberFormat.decimalPattern('id').format(product.price)}',  
style: TextStyle(  
fontSize: 18,  
fontWeight: FontWeight.w700,  
color: Colors.green[600],  
),  
,  
SizedBox(height: 4),  
Row(  
children: [  
Icon(  
Icons.star,  
color: Colors.amber,  
size: 16,  
),  
SizedBox(width: 4),  
Text(  
product.rating.toString(),  
style: TextStyle(  
fontSize: 14,  
color: widget.isDarkMode ? Colors.white70 : Colors.black,  
),  
,  
Spacer(),  
Text(  
'Stok: ${product.stock}',  
style: TextStyle(  
fontSize: 12,  
color: widget.isDarkMode ? Colors.white70 : Colors.grey[600],  
),  
,  

```



```
borderRadius: BorderRadius.circular(4),  
),  
child: Text(  
'NEW',  
style: TextStyle(  
color: Colors.white,  
fontSize: 10,  
fontWeight: FontWeight.bold,  
),  
),  
),  
),  
if (product.isHot)  
Container(  
margin: EdgeInsets.only(top: 4),  
padding: EdgeInsets.symmetric(horizontal: 6, vertical: 2),  
decoration: BoxDecoration(  
color: Colors.red,  
borderRadius: BorderRadius.circular(4),  
),  
child: Text(  
'HOT',  
style: TextStyle(  
color: Colors.white,  
fontSize: 10,  
fontWeight: FontWeight.bold,  
),  
),  
),  
),  
if (product.isOnSale)  
Container(  
margin: EdgeInsets.only(top: 4),  
padding: EdgeInsets.symmetric(horizontal: 6, vertical: 2),  
decoration: BoxDecoration(  
color: Colors.orange,  
borderRadius: BorderRadius.circular(4),  
),  
child: Text(  
'SALE',  
style: TextStyle(  
color: Colors.white,  
fontSize: 10,  
fontWeight: FontWeight.bold,  
),  
),  
),  
),  
],  
),  
),  
// Favorite button  
Positioned(  
top: 8,
```

```
        right: 8,
        child: IconButton(
          icon: Icon(
            Icons.favorite,
            color: Colors.red,
          ),
          onPressed: () => widget.onFavoriteToggle(product),
          iconSize: 24,
        ),
      ),
    ],
  ),
),
);
}
}
```

#### home\_page.dart

```
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'dart:async';
import 'product.dart';
import 'product_detail_page.dart';
import 'cart_page.dart';
import 'favorites_page.dart';
import 'account_page.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> with TickerProviderStateMixin {
  late PageController _pageController;
  Timer? _timer;

  final List<Product> products = [
    Product(
      id: 1,
      name: "iPhone 17 Pro",
      price: 15000000,
      description: "iPhone Terbaru",
      imageUrl: "assets/iPhone 17 Pro.png",
      rating: 4.5,
      stock: 15,
      category: "iPhone",
    )
  ];
}
```

```
    isNew: true,  
    isHot: true,  
,  
Product(  
  id: 2,  
  name: "13-inch MacBook Air",  
  price: 20000000,  
  description: "Laptop performa tinggi untuk bekerja dan gaming",  
  imageUrl: "assets/Mac.png",  
  rating: 4.8,  
  stock: 8,  
  category: "MacBook",  
  isHot: true,  
,  
Product(  
  id: 3,  
  name: "iPad Air",  
  price: 8000000,  
  description: "Tablet portabel dengan layar HD dan stylus",  
  imageUrl: "assets/iPad Air M3.png",  
  rating: 4.0,  
  stock: 18,  
  category: "iPad",  
,  
Product(  
  id: 4,  
  name: "iPhone 15",  
  price: 12000000,  
  description: "iPhone dengan kamera canggih",  
  imageUrl: "assets/iPhone17.png",  
  rating: 4.6,  
  stock: 20,  
  category: "iPhone",  
  isOnSale: true,  
  originalPrice: 14000000,  
,  
Product(  
  id: 5,  
  name: "MacBook Air",  
  price: 18000000,  
  description: "Laptop ringan untuk produktivitas",  
  imageUrl: "assets/MacBook Pro M4 Pro.png",  
  rating: 4.7,  
  stock: 12,  
  category: "MacBook",  
  isNew: true,  
,  
Product(  
  id: 6,  
  name: "iPad Pro",  
  price: 15000000,
```

```
description: "Tablet profesional dengan Apple Pencil",
imageUrl: "assets/iPad ProM4.png",
rating: 4.8,
stock: 10,
category: "iPad",
isHot: true,
),
];
}

String searchQuery = "";
String selectedCategory = 'All';
List<Product> filteredProducts = [];
List<Product> favoriteProducts = [];
List<Product> cartProducts = [];
bool isDarkMode = false;

late AnimationController _animationController;
late Animation<double> _fadeAnimation;

final List<String> promoImages = [
'assets/iPhone 17 Pro.png',
'assets/Mac.png',
'assets/iPhone17.png',
'assets/MacBook Pro M4 Pro.png',
'assets/iPad ProM4.png',
'assets/iPad Air M3.png',
];
}

@Override
void initState() {
super.initState();
filteredProducts = products;
_pageController = PageController();
_startAutoSlide();
_animationController = AnimationController(
duration: const Duration(milliseconds: 500),
vsync: this,
);
_fadeAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(
CurvedAnimation(parent: _animationController, curve: Curves.easeIn),
);
_animationController.forward();
}

@Override
void dispose() {
_timer?.cancel();
_pageController.dispose();
_animationController.dispose();
super.dispose();
}
```

```
void _startAutoSlide() {
    _timer = Timer.periodic(Duration(seconds: 3), (timer) {
        if (_pageController.hasClients) {
            int nextPage = (_pageController.page?.toInt() ?? 0) + 1;
            if (nextPage >= 4) nextPage = 0;
            _pageController.animateToPage(
                nextPage,
                duration: Duration(milliseconds: 500),
                curve: Curves.easeInOut,
            );
        }
    });
}

void _filterProducts() {
    setState(() {
        filteredProducts = products.where((product) {
            final matchesSearch = product.name.toLowerCase().contains(searchQuery.toLowerCase()) ||
                product.description.toLowerCase().contains(searchQuery.toLowerCase());
            final matchesCategory = selectedCategory == 'All' || product.category == selectedCategory;
            return matchesSearch && matchesCategory;
        }).toList();
    });
}

void _toggleFavorite(Product product) {
    setState(() {
        if (favoriteProducts.contains(product)) {
            favoriteProducts.remove(product);
        } else {
            favoriteProducts.add(product);
        }
    });
}

void _addToCart(Product product) {
    setState(() {
        cartProducts.add(product);
    });
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text('${product.name} ditambahkan ke keranjang!'),
            backgroundColor: Colors.green,
            action: SnackBarAction(
                label: 'Lihat Keranjang',
                textColor: Colors.white,
                onPressed: () {
                    // Navigate to cart page (would be implemented)
                },
            ),
        ),
    );
}
```

```
        ),  
    );  
}  
  
List<String> get categories => ['All', ...products.map((p) => p.category).toSet()];  
  
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: Text(  
                'Toko Kanesuuu_ ',  
                style: TextStyle(  
                    fontSize: 24,  
                    fontWeight: FontWeight.w600,  
                    letterSpacing: 1.2,  
                ),  
            ),  
            backgroundColor: Colors.transparent,  
            elevation: 0,  
            flexibleSpace: Container(  
                decoration: BoxDecoration(  
                    gradient: LinearGradient(  
                        colors: [  
                            Color(0xFF1a1a2e),  
                            Color(0xFF16213e),  
                            Color(0xFF0f3460),  
                        ],  
                        begin: Alignment.topLeft,  
                        end: Alignment.bottomRight,  
                    ),  
                ),  
            ),  
            foregroundColor: Colors.white,  
        ),  
        body: Container(  
            decoration: BoxDecoration(  
                gradient: LinearGradient(  
                    colors: [  
                        Color(0xFFf8f9fa),  
                        Color(0xFFe9ecef),  
                    ],  
                    begin: Alignment.topCenter,  
                    end: Alignment.bottomCenter,  
                ),  
            ),  
            child: Padding(  
                padding: const EdgeInsets.fromLTRB(16, 8, 16, 16),  
                child: Column(  
                    crossAxisAlignment: CrossAxisAlignment.start,  
                    children: [  

```



```

decoration: BoxDecoration(
  color: Colors.white,
  borderRadius: BorderRadius.circular(16),
  boxShadow: [
    BoxShadow(
      color: Colors.black.withOpacity(0.1),
      blurRadius: 8,
      offset: Offset(0, 4),
    ),
  ],
),
child: TextField(
  onChanged: (value) {
    searchQuery = value;
    _filterProducts();
  },
  decoration: InputDecoration(
    hintText: 'Cari produk...',
    hintStyle: TextStyle(color: Colors.grey[500]),
    prefixIcon: Icon(Icons.search, color: Color(0xFF0f3460)),
    border: InputBorder.none,
    contentPadding: EdgeInsets.symmetric(horizontal: 16, vertical: 14),
  ),
  style: TextStyle(color: Colors.black, fontSize: 16),
),
),
SizedBox(height: 12),

// Category Filter
SizedBox(
  height: 45,
  child: ListView.builder(
    scrollDirection: Axis.horizontal,
    itemCount: categories.length,
    itemBuilder: (context, index) {
      final category = categories[index];
      final isSelected = category == selectedCategory;
      return Container(
        margin: EdgeInsets.only(right: 10),
        child: FilterChip(
          label: Text(
            category,
            style: TextStyle(
              fontSize: 13,
              fontWeight: FontWeight.w500,
              color: isSelected ? Colors.white : Color(0xFF0f3460),
            ),
          ),
          selected: isSelected,
          onSelected: (selected) {
            setState(() {

```

```

        selectedCategory = category;
        _filterProducts();
    });
},
backgroundColor: Colors.white,
selectedColor: Color(0xFF0f3460),
checkmarkColor: Colors.white,
elevation: 2,
shadowColor: Colors.black.withOpacity(0.1),
padding: EdgeInsets.symmetric(horizontal: 12, vertical: 6),
),
);
},
),
),
),
SizedBox(height: 12),

Text(
'Produk ${selectedCategory == 'All' ? 'Terbaru' : selectedCategory}',
style: TextStyle(
fontSize: 24,
fontWeight: FontWeight.w700,
color: Color(0xFF0f3460),
letterSpacing: 0.5,
),
),
),
SizedBox(height: 12),

Expanded(
child: RefreshIndicator(
onRefresh: () async {
// Simulate refresh
await Future.delayed(Duration(seconds: 1));
setState(() {
filteredProducts = products;
searchQuery = '';
selectedCategory = 'All';
});
},
child: FadeTransition(
opacity: _fadeAnimation,
child: GridView.builder(
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,
crossAxisSpacing: 12,
mainAxisSpacing: 12,
childAspectRatio: 0.75,
),
itemCount: filteredProducts.length,
itemBuilder: (context, index) {
return ProductCard(

```



```
        ),
    ],
    onTap: (index) {
        switch (index) {
            case 0:
                // Already on home page
                break;
            case 1:
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) => FavoritesPage(
                            favoriteProducts: favoriteProducts,
                            onFavoriteToggle: _toggleFavorite,
                            onAddToCart: _addToCart,
                            isDarkMode: isDarkMode,
                        ),
                    ),
                );
                break;
            case 2:
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) => CartPage(
                            cartProducts: cartProducts,
                            onRemoveFromCart: (product) {
                                setState(() {
                                    cartProducts.remove(product);
                                });
                            },
                            onAddToCart: _addToCart,
                            isDarkMode: isDarkMode,
                        ),
                    ),
                );
                break;
            case 3:
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) => AccountPage(),
                    ),
                );
                break;
        }
    },
),
),
);
}
}
```

```
class ProductCard extends StatelessWidget {
    final Product product;
    final VoidCallback onTap;
    final bool isFavorite;
    final VoidCallback onFavoriteToggle;
    final VoidCallback onAddToCart;
    final bool isDarkMode;

    const ProductCard({
        super.key,
        required this.product,
        required this.onTap,
        required this.isFavorite,
        required this.onFavoriteToggle,
        required this.onAddToCart,
        required this.isDarkMode,
    });

    @override
    Widget build(BuildContext context) {
        return Card(
            elevation: 4,
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(12),
            ),
            child: InkWell(
                onTap: onTap,
                borderRadius: BorderRadius.circular(12),
                child: Stack(
                    children: [
                        Padding(
                            padding: const EdgeInsets.all(12.0),
                            child: Column(
                                crossAxisAlignment: CrossAxisAlignment.start,
                                children: [
                                    Expanded(
                                        child: Container(
                                            width: double.infinity,
                                            decoration: BoxDecoration(
                                                borderRadius: BorderRadius.circular(8),
                                                image: DecorationImage(
                                                    image: AssetImage(product.imageUrl),
                                                    fit: BoxFit.cover,
                                                ),
                                            ),
                                        ),
                                    ),
                                    SizedBox(height: 8),
                                    Text(
                                        product.name,
                                    ),
                                ],
                            ),
                        ),
                    ],
                ),
            ),
        );
    }
}
```

```

style: TextStyle(
  fontSize: 16,
  fontWeight: FontWeight.bold,
  color: Colors.black,
),
maxLines: 1,
overflow: TextOverflow.ellipsis,
),
SizedBox(height: 4),
if (product.isOnSale && product.originalPrice != null)
Row(
  children: [
    Text(
      'Rp${NumberFormat.decimalPattern('id').format(product.originalPrice)}',
      style: TextStyle(
        fontSize: 14,
        decoration: TextDecoration.lineThrough,
        color: Colors.grey,
      ),
    ),
    SizedBox(width: 8),
    Text(
      'Rp${NumberFormat.decimalPattern('id').format(product.price)}',
      style: TextStyle(
        fontSize: 18,
        fontWeight: FontWeight.bold,
        color: Colors.red[700],
      ),
    ),
  ],
)
else
Text(
  'Rp${NumberFormat.decimalPattern('id').format(product.price)}',
  style: TextStyle(
    fontSize: 18,
    fontWeight: FontWeight.bold,
    color: Colors.green[700],
  ),
),
SizedBox(height: 4),
Row(
  children: [
    Icon(
      Icons.star,
      color: Colors.amber,
      size: 16,
    ),
    SizedBox(width: 4),
    Text(
      product.rating.toString(),
    )
  ],
)

```





**main.dart**

```
import 'package:flutter/material.dart';
import 'home_page.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Toko Kanesuuu_',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        fontFamily: 'Roboto',
        textTheme: TextTheme(
          headlineLarge: TextStyle(
            fontSize: 32,
            fontWeight: FontWeight.w700,
            letterSpacing: -0.5,
          ),
          headlineMedium: TextStyle(
            fontSize: 28,
            fontWeight: FontWeight.w600,
            letterSpacing: -0.25,
          ),
          headlineSmall: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.w600,
            letterSpacing: 0,
          ),
          titleLarge: TextStyle(
            fontSize: 22,
            fontWeight: FontWeight.w500,
            letterSpacing: 0,
          ),
          titleMedium: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w500,
            letterSpacing: 0.15,
          ),
          titleSmall: TextStyle(
            fontSize: 14,
            fontWeight: FontWeight.w500,
            letterSpacing: 0.1,
          ),
          bodyLarge: TextStyle(
            fontSize: 16,
```

```
fontWeight: FontWeight.w400,
letterSpacing: 0.5,
),
bodyMedium: TextStyle(
fontSize: 14,
fontWeight: FontWeight.w400,
letterSpacing: 0.25,
),
bodySmall: TextStyle(
fontSize: 12,
fontWeight: FontWeight.w400,
letterSpacing: 0.4,
),
labelLarge: TextStyle(
fontSize: 14,
fontWeight: FontWeight.w500,
letterSpacing: 0.1,
),
labelSmall: TextStyle(
fontSize: 11,
fontWeight: FontWeight.w500,
letterSpacing: 0.5,
),
),
cardTheme: CardThemeData(
elevation: 4,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(16),
),
),
elevatedButtonTheme: ElevatedButtonThemeData(
style: ElevatedButton.styleFrom(
elevation: 2,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(12),
),
padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),
),
),
inputDecorationTheme: InputDecoration(
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(12),
),
filled: true,
fillColor: Colors.grey[50],
),
chipTheme: ChipThemeData(
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(8),
),
),
),
```

```
        ),  
        home: HomePage(),  
        debugShowCheckedModeBanner: false,  
    );  
}  
}
```

### main.dart

```
import 'package:flutter/material.dart';  
import 'home_page.dart';  
  
void main() {  
    runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
    const MyApp({super.key});  
  
    @override  
    Widget build(BuildContext context) {  
        return MaterialApp(  
            title: 'Toko Kanesuuu_ ',  
            theme: ThemeData(  
                primarySwatch: Colors.blue,  
                fontFamily: 'Roboto',  
                textTheme: TextTheme(  
                    headlineLarge: TextStyle(  
                        fontSize: 32,  
                        fontWeight: FontWeight.w700,  
                        letterSpacing: -0.5,  
                    ),  
                    headlineMedium: TextStyle(  
                        fontSize: 28,  
                        fontWeight: FontWeight.w600,  
                        letterSpacing: -0.25,  
                    ),  
                    headlineSmall: TextStyle(  
                        fontSize: 24,  
                        fontWeight: FontWeight.w600,  
                        letterSpacing: 0,  
                    ),  
                    titleLarge: TextStyle(  
                        fontSize: 22,  
                        fontWeight: FontWeight.w500,  
                        letterSpacing: 0,  
                    ),  
                    titleMedium: TextStyle(  
                        fontSize: 16,  
                        fontWeight: FontWeight.w500,  
                    ),  
                ),  
            ),  
        );  
    }  
}
```

```
letterSpacing: 0.15,  
,  
titleSmall: TextStyle(  
  fontSize: 14,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.1,  
,  
bodyLarge: TextStyle(  
  fontSize: 16,  
  fontWeight: FontWeight.w400,  
  letterSpacing: 0.5,  
,  
bodyMedium: TextStyle(  
  fontSize: 14,  
  fontWeight: FontWeight.w400,  
  letterSpacing: 0.25,  
,  
bodySmall: TextStyle(  
  fontSize: 12,  
  fontWeight: FontWeight.w400,  
  letterSpacing: 0.4,  
,  
labelLarge: TextStyle(  
  fontSize: 14,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.1,  
,  
labelSmall: TextStyle(  
  fontSize: 11,  
  fontWeight: FontWeight.w500,  
  letterSpacing: 0.5,  
,  
,  
cardTheme: CardThemeData(  
  elevation: 4,  
  shape: RoundedRectangleBorder(  
    borderRadius: BorderRadius.circular(16),  
,  
,  
elevatedButtonTheme: ElevatedButtonThemeData(  
  style: ElevatedButton.styleFrom(  
    elevation: 2,  
    shape: RoundedRectangleBorder(  
      borderRadius: BorderRadius.circular(12),  
,  
    padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),  
,  
,  
inputDecorationTheme: InputDecorationTheme(  
  border: OutlineInputBorder(  
    borderRadius: BorderRadius.circular(12),
```

```
        ),
        filled: true,
        fillColor: Colors.grey[50],
    ),
    chipTheme: ChipThemeData(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(8),
        ),
    ),
),
),
),
),
),
home: HomePage(),
debugShowCheckedModeBanner: false,
);
}
}
```

#### product\_detail\_page.dart

```
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'product.dart';

class ProductDetailPage extends StatelessWidget {
    final Product product;
    final Function(Product) onFavoriteToggle;
    final Function(Product) onAddToCart;
    final bool isFavorite;

    const ProductDetailPage({
        super.key,
        required this.product,
        required this.onFavoriteToggle,
        required this.onAddToCart,
        required this.isFavorite,
    });

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text('Detail Produk'),
                backgroundColor: Colors.blue,
                foregroundColor: Colors.white,
            ),
            body: SingleChildScrollView(
                child: Column(
                    children: [
                        Container(
                            height: 300,
                            width: double.infinity,
```

```
decoration: BoxDecoration(
    image: DecorationImage(
        image: NetworkImage(product.imageUrl),
        fit: BoxFit.cover,
    ),
),
),
),
Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
            Text(
                product.name,
                style: TextStyle(
                    fontSize: 24,
                    fontWeight: FontWeight.bold,
                ),
            ),
            SizedBox(height: 8),
            Row(
                children: [
                    Icon(
                        Icons.star,
                        color: Colors.amber,
                    ),
                    SizedBox(width: 4),
                    Text(
                        product.rating.toString(),
                        style: TextStyle(fontSize: 18),
                    ),
                    Spacer(),
                    Container(
                        padding: EdgeInsets.symmetric(horizontal: 8, vertical: 4),
                        decoration: BoxDecoration(
                            color: product.stock > 0 ? Colors.green : Colors.red,
                            borderRadius: BorderRadius.circular(4),
                        ),
                        child: Text(
                            product.stock > 0 ? 'Tersedia' : 'Habis',
                            style: TextStyle(
                                color: Colors.white,
                                fontWeight: FontWeight.bold,
                            ),
                        ),
                    ),
                ],
            ),
            SizedBox(height: 16),
            Text(
                'Rp${NumberFormat.decimalPattern('id').format(product.price)}',
            ),
        ],
    ),
);
```

```
style: TextStyle(
  fontSize: 28,
  fontWeight: FontWeight.bold,
  color: Colors.green[700],
),
),
SizedBox(height: 16),
Text(
  'Deskripsi Produk',
  style: TextStyle(
    fontSize: 18,
    fontWeight: FontWeight.bold,
  ),
),
SizedBox(height: 8),
Text(
  product.description,
  style: TextStyle(
    fontSize: 16,
    height: 1.5,
  ),
),
SizedBox(height: 24),
if (product.stock > 0)
  SizedBox(
    width: double.infinity,
    child: ElevatedButton(
      onPressed: () {
        onAddToCart(product);
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.blue,
        foregroundColor: Colors.white,
        padding: EdgeInsets.symmetric(vertical: 16),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(8),
        ),
      ),
      child: Text(
        'Tambah ke Keranjang',
        style: TextStyle(fontSize: 18),
      ),
    ),
  ),
  if (product.stock <= 0)
    SizedBox(
      width: double.infinity,
      child: ElevatedButton(
        onPressed: null,
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.grey,
```

```
padding: EdgeInsets.symmetric(vertical: 16),
shape: RoundedRectangleBorder(
  borderRadius: BorderRadius.circular(8),
),
),
),
child: Text(
  'Stok Habis',
  style: TextStyle(
    fontSize: 18,
    color: Colors.white,
  ),
),
),
),
),
),
),
],
),
),
],
),
),
),
);
}
}
```

#### product.dart

```
class Product {
  final int id;
  final String name;
  final double price;
  final String description;
  final String imageUrl;
  final double rating;
  final int stock;
  final String category;
  final bool isNew;
  final bool isHot;
  final bool isOnSale;
  final double? originalPrice;

  const Product({
    required this.id,
    required this.name,
    required this.price,
    required this.description,
    required this.imageUrl,
    required this.rating,
    required this.stock,
    required this.category,
    this.isNew = false,
```

```
this.isHot = false,  
this.isOnSale = false,  
this.originalPrice,  
});  
  
factory Product.fromJson(Map<String, dynamic> json) {  
    return Product(  
        id: json['id'],  
        name: json['name'],  
        price: json['price'].toDouble(),  
        description: json['description'],  
        imageUrl: json['imageUrl'],  
        rating: json['rating'].toDouble(),  
        stock: json['stock'],  
        category: json['category'] ?? 'General',  
        isNew: json['isNew'] ?? false,  
        isHot: json['isHot'] ?? false,  
        isOnSale: json['isOnSale'] ?? false,  
        originalPrice: json['originalPrice']?.toDouble(),  
    );  
}  
  
bool get isFavorite => false; // This will be managed by state management  
}
```

### Kesimpulan

Proyek ini berhasil memenuhi instruksi tugas dengan menerapkan class model Product untuk struktur data dan navigasi halaman menggunakan Flutter. Aplikasi ini dapat diperluas dengan menambahkan state management (seperti Provider atau Bloc) untuk persistensi data favorit dan keranjang, serta integrasi API untuk data dinamis.