**LAPORAN**

***(Pemrograman Mobile)***

**(APLIKASI TAMPILAN *SHOPPING* *LIST*, DAN TAMPILAN OVERLAY MENAMBAHKAN BARANG BARU KE *SHOPPING LIST*)**

****

|  |  |  |
| --- | --- | --- |
| **Nama** | **:** | Alvina Leony |
| **NPM** | **:** | 221510003 |
| **Dosen** | **:** | Saut Pintubipar Saragih, S.Kom., M.MSI. |

# KATA PENGANTAR

Dengan puji syukur kepada Tuhan Yang Maha Esa karena hanya dengan kasih dan karunia-Nya penulis mampu menyelesaikan proposal ini yang berjudul “**APLIKASI MENAMPILKAN *SHOPPING LIST*, DAN TAMPILAN MENAMBAHKAN BARANG BARU KE *SHOPPING LIST***”. Proposal ini telah disusun untuk memenuhi salah satu syarat memperoleh kelulusan untuk mata kuliah Pemrograman Mobile pada jurusan Sistem Informasi. Selama menuliskan proposal penelitian ini, penulis telah mengalami kesulitan dan kendala, hingga penulis sadar bahwa proposal penelitian ini masih sangat jauh dari kesempurnaan. Oleh karena itu, penulis berharap akan kritik dan saran yang membangun demi kesempurnaan proposal penelitian ini.

Dengan kesempatan ini, penulis ingin menyatakan ucapan terima kasih sebesar-besarnya kepada Bapak Saut Pintubipar Saragih, S.Kom., M.MSI. selaku dosen pengajar mata kuliah ini yang telah memberikan arahan dan bimbingan kepada penulis selama proses penyelesaian proposal penelitian ini. Penulis juga ingin mengucapkan terima kasih kepada Bapak Tukino, S.Kom., M.SI. sebagai dosen pembimbing yang juga memberikan nasehat dan saran pada proses penulisan proposal penelitian ini.

Penulis sangat berharap bahwa proposal ini dapat bermanfaat dan dapat membantu para pembaca.

Batam, 9 November 2024

Alvina Leony

# **DAFTAR ISI**

[KATA PENGANTAR i](#_Toc182049815)

[DAFTAR ISI ii](#_Toc182049816)

[DAFTAR GAMBAR iii](#_Toc182049817)

[DAFTAR LAMPIRAN iv](#_Toc182049818)

[BAB I 1](#_Toc182049819)

[PENDAHULUAN 1](#_Toc182049820)

[1.1. Latar Belakang 1](#_Toc182049821)

[1.2. Rumusan Masalah 2](#_Toc182049822)

[1.3. Batasan Masalah 2](#_Toc182049823)

[1.4. Tujuan Penelitian 2](#_Toc182049824)

[1.5. Manfaat Penelitian 2](#_Toc182049825)

[BAB II 3](#_Toc182049826)

[PEMBAHASAN 3](#_Toc182049827)

[2.1. Teori Khusus 3](#_Toc182049828)

[2.1.1. Flutter 3](#_Toc182049829)

[2.1.2. Dart 3](#_Toc182049830)

[2.1.3. Android Studio 3](#_Toc182049831)

[2.1.4. Figma 4](#_Toc182049832)

[2.1.5. Layout 4](#_Toc182049833)

[2.2. Teori Umum 4](#_Toc182049834)

[2.2.1. Aplikasi Mobile 4](#_Toc182049835)

[2.2.2. Layout 4](#_Toc182049836)

[2.2.3. Checklist 5](#_Toc182049837)

[BAB III 6](#_Toc182049838)

[PEMBAHASAN 6](#_Toc182049839)

[3.1. Desain Sistem Projek Layout Aplikasi *Shopping Llist* 6](#_Toc182049840)

[3.2. User Interface (UI) Layout Aplikasi Shopping List 6](#_Toc182049841)

[3.3. Coding Layout Aplikasi Shopping List 12](#_Toc182049842)

[DAFTAR PUSTAKA 27](#_Toc182049843)

[LAMPIRAN 28](#_Toc182049844)

# DAFTAR GAMBAR

[Gambar 3. 1 *Use Case Diagram* Aplikasi *Layout Shopping List* 6](#_Toc182049794)

[Gambar 3. 2 *UI Layout Design* Aplikasi *Shopping List* pada Figma 7](#_Toc182049795)

[Gambar 3. 3 *Layout Header* Aplikasi *Shopping List* 7](#_Toc182049796)

[Gambar 3. 4 *UI Design Vertical* *Scroll Checklist* 8](#_Toc182049797)

[Gambar 3. 5 Teks pada *Checklist* 8](#_Toc182049798)

[Gambar 3. 6 *Checkbox* pada *Checklist* 8](#_Toc182049799)

[Gambar 3. 7 *Trash Icon* *(Remove Item) Button* 9](#_Toc182049800)

[Gambar 3. 8 *Add New Item UI Button* 9](#_Toc182049801)

[Gambar 3. 9 *Overlay Form Add New Item* 9](#_Toc182049802)

[Gambar 3. 10 *Header* pada *Overlay Form Add New Item* 9](#_Toc182049803)

[Gambar 3. 11 *Textbox* pada *Overlay Form Add New Item* 10](#_Toc182049804)

[Gambar 3. 12 *Save Button* pada *Overlay Form Add New Item* 10](#_Toc182049805)

[Gambar 3. 13 *UI* *Overlay* Menambah Item baru setelah dijalankan pada *Figma* 10](#_Toc182049806)

[Gambar 3. 14 Komponen *Checklist* dan *Delete Icon Button* 11](#_Toc182049807)

[Gambar 3. 15 Komponen *Textbox* 11](#_Toc182049808)

[Gambar 3. 16 Tampilan *UI* Aplikasi *Shopping List* setelah dijalankan pada *Figma* 12](#_Toc182049809)

[Gambar 3. 17 *Coding* Aplikasi *Layout Shopping List* pada *Android Studio* 24](#_Toc182049810)

[Gambar 3. 18 *Screenshot* Aplikasi *Shopping List* *Default* pada *Android Studio* 24](#_Toc182049811)

[Gambar 3. 19 Screenshot Aplikasi *Shopping List Item Checked* pada *Android Studio* 25](#_Toc182049812)

[Gambar 3. 20 Screenshot Aplikasi *Shopping List Item Deleted* pada *Android Studio* 25](#_Toc182049813)

[Gambar 3. 21 Screenshot *Overlay Form* Menambah Item Baru pada Aplikasi *Shopping List* 26](#_Toc182049814)

# DAFTAR LAMPIRAN

[Lampiran 1. Desain Aplikasi *Shopping List* menggunakan *Figma* 28](#_Toc182049845)

[Lampiran 2. Menjalankan Desain *Layout* Aplikasi *Shopping List* pada *Figma* 28](#_Toc182049846)

[Lampiran 3. Menjalankan Tampilan *Overlay Form Add New Item* pada *Figma* 29](#_Toc182049847)

[Lampiran 4. *Coding* *Aplikasi Shopping List* menggunakan *Android Studio* 29](#_Toc182049848)

[Lampiran 5. Aplikasi *Shopping List* menggunakan *Android Studio* 30](#_Toc182049849)

# BAB I

# PENDAHULUAN

## Latar Belakang

Dalam sebuah aplikasi mobile, tampilan *layout* aplikasi *mobile* tersebut memiliki dampak signifikan kepada pengalaman pengguna selama memakai aplikasi tersebut. Dari faktor warna yang dipilih untuk elemen-elemen tertentu, letak dan ukuran elemen satu dengan elemen lainnya, serta kemudahan pengguna berinteraksi dengan komponen pada aplikasi itu, *layout* aplikasi yang efisien memastikan bahwa pengguna dapat dengan mudah menggunakan aplikasi yang telah dirancang.

Melanjutkan dari laporan penelitan sebelumnya, dimana pengguna telah merancang sebuah *widget checklist* dengan bahasa pemrograman Dart dan framework Flutter pada Android Studio, penulis akan mengembangkan widget tersebut lebih lanjut dengan membangun tampilan *layout* aplikasi yang mudah dipahami dan digunakan.

Untuk membantu mempercepat waktu yang dibutuhkan untuk *programming,* dibutuhkan sebuah desain *layout interface* yang membantu memberi gambaran akan tampilan akhir aplikasi. Untuk memenuhi kebutuhan itu, penulis lanjut menggunakan *Figma,* aplikasi/perangkat lunak yang digunakan untuk desain produk digital seperti *mock-up, mindmap,* dan tampilan *interface* aplikasi yang ingin dikembangkan.

Penelitian ini bertujuan untuk mengembangkan lebih lanjut *widget* yang sebelumnya telah dirancang dengan menambahkan tampilan *layout* aplikasi *shopping list* yang interaktif pada *Android* *Studio*. Penelitian ini berharap untuk menghasilkan *layout* aplikasi serta tampilan *overlay* menambah benda baru kepada *shopping list* yang fungsional, menarik, mudah dipahami dan digunakan serta membantu meningkatkan pengalaman pengguna selama memakai aplikasi tersebut.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka rumusan masalah pada penelitian ini berupa:

1. Bagaimana merancang tampilan *layout shopping list* pada Android Studio*?*
2. Bagaimana merancang *overlay* menambah benda baru kepada *shopping list* pada Android Studio?

## Batasan Masalah

Untuk menjelaskan lebih jelas pembahasan laporan agar hasil akhir penelitian ini spesifik dan detail, maka penelitian ini dibatasi pada:

1. Proses merancang *layout* aplikasi *shopping list* pada *Android Studio* menggunakan *framework Flutter* dan bahasa pemrograman *Dart.*
2. Proses merancang *overlay* menambah benda baru kepada *shopping list* pada *Android Studio* menggunakan *framework Flutter* dan bahasa pemrograman *Dart*

## Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan pada poin diatas, berikut tujuan makalah ini:

1. Merancang tampilan *layout* aplikasi *shopping list* yang efisien, fungsional dan *user-friendly.*

## Manfaat Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan pada poin diatas, berikut manfaat makalah ini:

1. Membangun tampilan layout aplikasi *shopping list* yang membantu menghemat waktu ketika mengembangkah aplikasi *shopping list* yang dinamis.

# BAB II

# PEMBAHASAN

## Teori Khusus

### Flutter

*Flutter* adalah *framework open-source* yang menawarkan performa setara dengan aplikasi *native* dan mempercepat pengembangan aplikasi lintas platform untuk *Android* dan *iOS*. Dibuat dengan bahasa pemrograman *Dart*, *Flutter* memungkinkan developer menggunakan satu basis kode tanpa perlu mempelajari dua bahasa pemrograman. (Raharjo, 2021).

*Flutter* memberi beberapa keunggulan kepada *developer* berupa: *fast* *development*, *expressive* *and* *flexible* *UI* serta *native performance*. *Fast* *development* memungkinkan proses pembuatan aplikasi yang lebih efisien bagi *developer*. *Expressive and flexible UI* memberikan tampilan yang menarik untuk *Android (Material Design)* dan *iOS (Cupertino),* dengan berbagai API, scrolling yang alami, serta kemampuan mengenali platform secara otomatis. Selain itu, *Flutter* juga memiliki fitur hot reload yang mempermudah pengembangan, memungkinkan desain antarmuka, penambahan fitur, dan *debugging* (Suryono & Hardiansah, 2020).

### Dart

*Dart* dirancang sebagai bahasa pemrograman yang efisien untuk membangun aplikasi cepat di berbagai platform. *Dart* fokus untuk meningkatkan produktivitas *developer* dalam pengembangan *multi-platform*, dengan runtime yang fleksibel untuk mendukung berbagai *framework* aplikasi. Sebagai fondasi dari Flutter, *Dart* tidak hanya menyediakan bahasa dan *runtime* untuk aplikasi *Flutter*, tetapi juga mendukung berbagai tugas penting pengembang seperti memformat kode, melakukan analisis, dan menjalankan pengujian (Dart, 2024).

### Android Studio

Android Studio adalah sebuah *Integrated Development Environment (IDE)* yang dirancang untuk pengembangan aplikasi *Android*, dibangun dari basis *IntelliJ* *IDEA*. *IDE* ini menawarkan berbagai fitur untuk meningkatkan efisiensi developer, seperti emulator yang cepat dan kaya fitur, lingkungan terintegrasi untuk pengembangan di semua perangkat *Android*, fitur *Instant Run* yang memungkinkan perubahan langsung tanpa perlu membangun *APK* baru, sistem versi fleksibel berbasis *Gradle*, dukungan terintegrasi untuk *Google Cloud Platform*, dan masih banyak fitur lainnya (Suryana, 2018).

### Figma

*Figma* adalah alat desain berbasis *web* yang memungkinkan pengguna membuat desain digital berkualitas tinggi, seperti website dan aplikasi. *Figma* mendukung kolaborasi tim secara *real*-*time*, memudahkan pemberian masukan dan pengelolaan revisi. Fitur-fiturnya meliputi elemen desain, *grid*, *library*, dan prototyping. Elemen desain membantu menambahkan bentuk, *grid* menjaga tata letak rapi, *library* menyimpan elemen untuk digunakan kembali, dan *prototyping* memungkinkan pembuatan interaksi serta animasi (Sakti Wibowo, 2023).

### Layout

Pada *Flutter, widget* merupakan komponen-komponen penting yang membangun *user* *interface* yang akan dilihat oleh para pengguna. Beberapa contoh *widget* berupa *button, textbox, icon, appbar, image, container* dan lebih banyak lagi.

## Teori Umum

### Aplikasi Mobile

Aplikasi mobile adalah sejenis perangkat lunak yang dirancang untuk berjalan di ponsel, tablet, atau smartphone. Aplikasi ini menawarkan layanan yang mirip dengan yang dapat diakses di PC (Putra et al., 2023).

### Layout

Layout merupakan pengaturan komposisi komponen antarmuka pengguna pada suatu aplikasi. Layout ini menentukan letak komponen, ukuran komponen-komponen tertentu, menentukan apa yang harus ditekankan dan apa yang tidak. Selain itu, layout juga menentukan *widget* apa yang dipakai pada aplikasi itu.

### Checklist

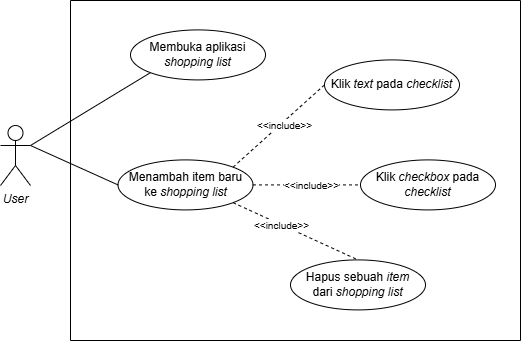
*Checklist* merupakan sebuah metode mencatat dan merekam kumpulan data. Dengan *checklist*, pengguna dapat lebih mudah mengatur tugas atau benda dengan lebih efisien, sehingga memastikan bahwa semua tugas atau item yang diperlukan dapat dipantau dan ditangani dengan baik. Pada aplikasi *shopping list* yang akan dibangun dalam proposal penelitian ini, checklist digunakan untuk menandai bahan-bahan atau barang yang dibutuhkan untuk sebuah *user*. Pengguna dapat mencentang setiap bahan atau barang yang ada, sehingga meningkatkan efisiensi mereka.

# BAB III

# PEMBAHASAN

## Desain Sistem Projek Layout Aplikasi *Shopping Llist*

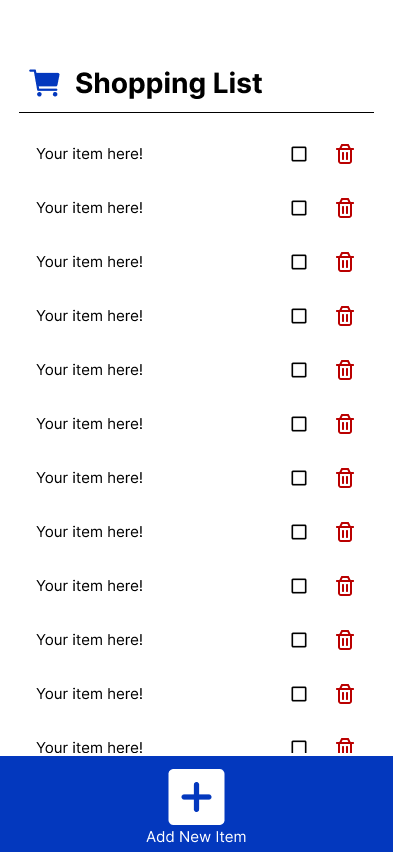
Berikut desain sistem menggunakan diagram *Use Case* yang telah dirancang untuk projek *layout* aplikasi *shopping list* ini:



Gambar 3. *Use Case Diagram* Aplikasi *Layout Shopping List*

## User Interface (UI) Layout Aplikasi Shopping List

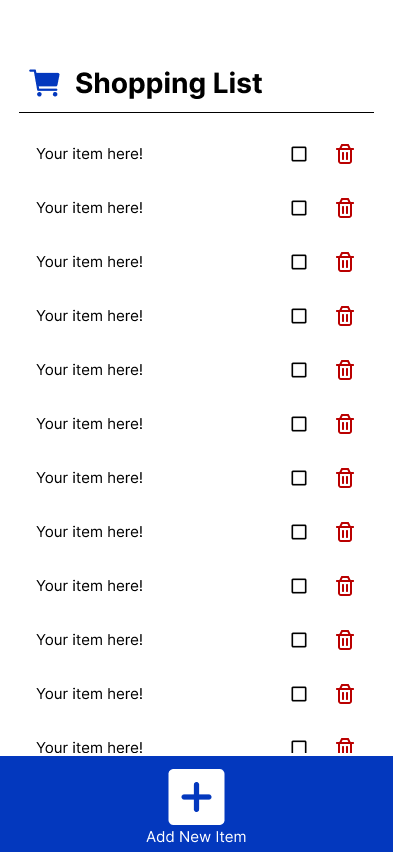
Berikut merupakan tampilan *user interface* (antarmuka pengguna) *layout* aplikasi *shopping list* yang telah dirancang menggunakan Figma:



Gambar 3. 2 *UI Layout Design* Aplikasi *Shopping List* pada Figma

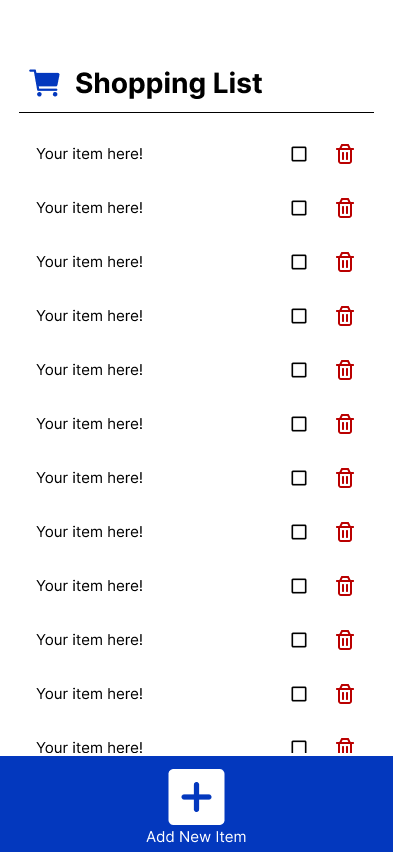
Ketika aplikasi *shopping list* ini dibuka*,* tampilan tersebut akan dibagi menjadi 3 bagian utama, yaitu;

* Header dengan *icon*, yang mengandung *icon* utama aplikasi serta heading teks “Shopping List” dalam warna biru gelap dengan efek *bold.* Header ini ditutup oleh divider dalam bentuk garis warna hitam horizontal dibawah teks dan ikon.



Gambar 3. 3 *Layout Header* Aplikasi *Shopping List*

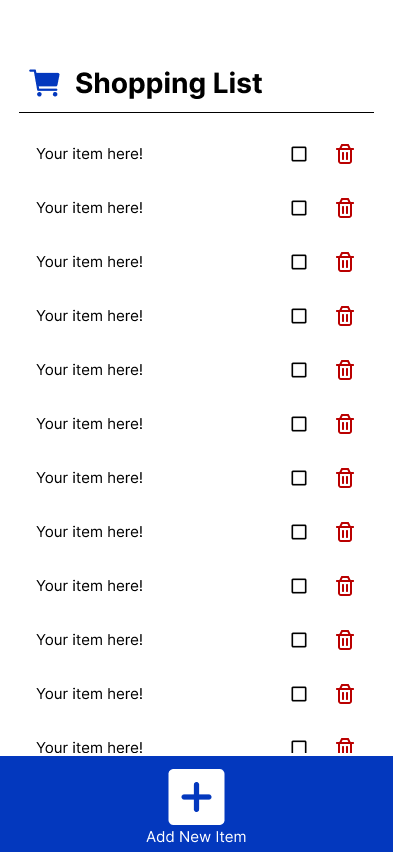
* Vertical scrolling checklist, yaitu tampilan checkbox yang pengguna dapat interaksi dengan klik dan scroll.



Gambar 3. 4 *UI Design Vertical* *Scroll Checklist*

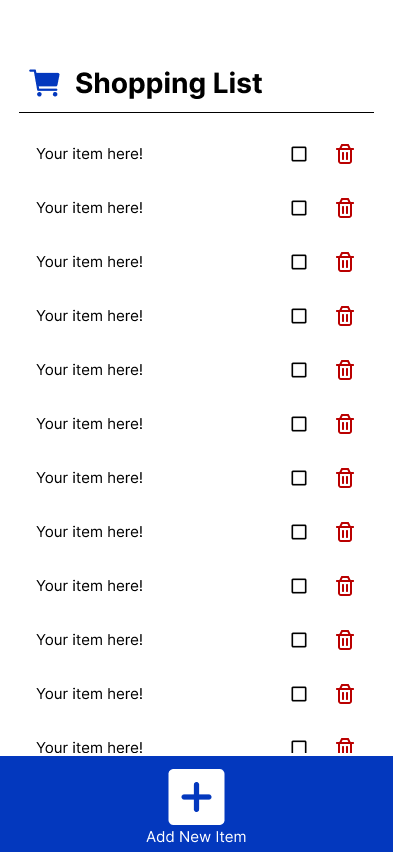
Dalam vertical scrolling checklist ini, checklist tersebut dibagi menjadi 3 bagian yaitu:

* Teks yang merepresentasikan benda yang telah ditambah kedalam *shopping list.* Pengguna juga dapat klik pada teks tersebut untuk menyatakan bahwa item itu sudah dipenuhi atau belum.



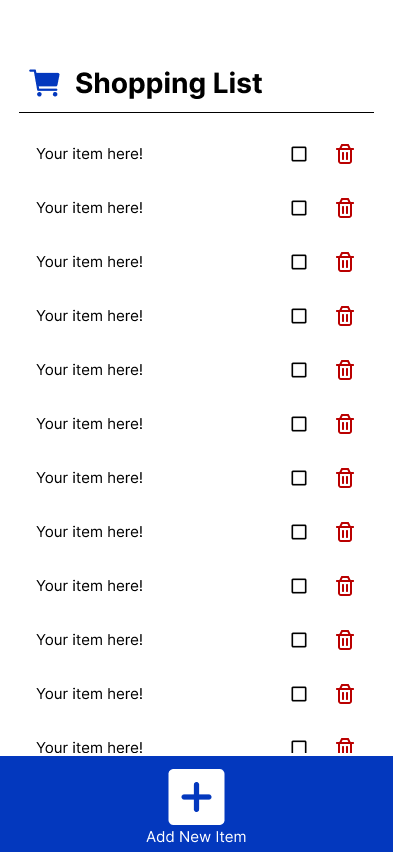
Gambar 3. 5 Teks pada *Checklist*

* Icon checkbox yang berfungsi sebagai sebuah tanda apakah benda sudah dipenuhi atau belum. Apabila sudah dipenuhi dan pengguna telah klik, *checkbox* ini akan berubah menjadi warna hijau.



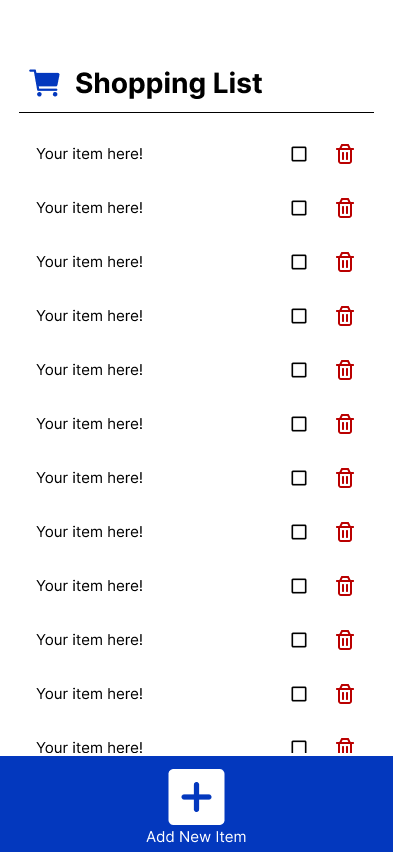
Gambar 3. 6 *Checkbox* pada *Checklist*

* Icon tempat sampah atau *trashcan,* yang berfungsi sebagai suatu tombol ikon yang dapat diklik oleh pengguna untuk menghapus item dari *shopping list.*



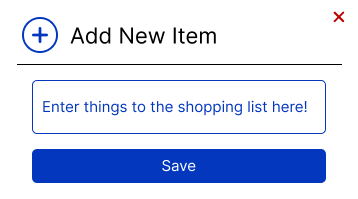
Gambar 3. 7 *Trash Icon* *(Remove Item) Button*

* *Add new item icon button,* yang berada dibawah layar antarmuka pengguna dengan latar belakang warna biru gelap dan icon menambah warna putih.



Gambar 3. 8 *Add New Item UI Button*

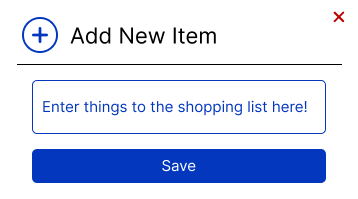
*Icon button* ini digunakan untuk menampilkan overlay form menambahkan item baru ini:



Gambar 3. 9 *Overlay Form Add New Item*

Tampilan overlay form menambah item baru kepada *shopping list,* terdiri atas 3 bagian, yaitu:

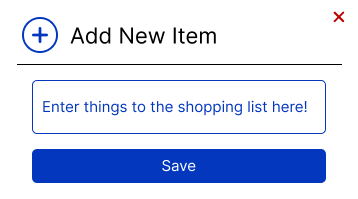
* Header dengan icon tutup overlay, yang berisi icon tambah statis, judul overlay “Add New Item” dan button menutup overlay tersebut dalam bentuk *icon* X warna merah.



Gambar 3. 10 *Header* pada *Overlay Form Add New Item*

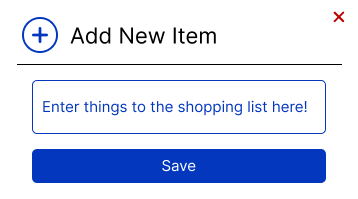
Header ini ditutup oleh divider dalam bentuk garis warna hitam horizontal dibawah teks dan ikon.

* Textbox simpel yang digunakan untuk merepresentasikan kotak yang akan digunakan oleh menginput item baru



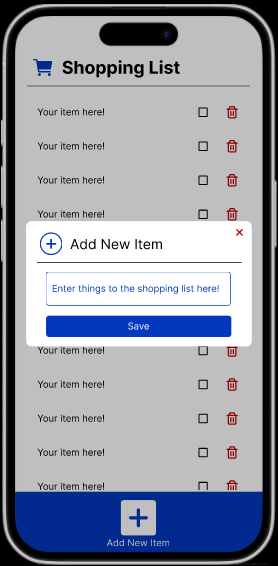
Gambar 3. 11 *Textbox* pada *Overlay Form Add New Item*

* Save button yang akan dikembangkan lebih lanjut untuk menyimpan item baru yang telah diinput oleh pengguna.



Gambar 3. 12 *Save Button* pada *Overlay Form Add New Item*

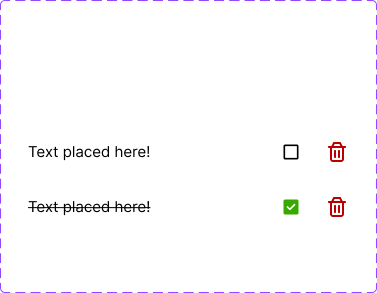
Berikut tampilan aplikasi setelah *icon button add new item* diklik oleh pengguna:



Gambar 3. 13 *UI* *Overlay* Menambah Item baru setelah dijalankan pada *Figma*

Berikut 2 komponen interaktif yang ada pada desain antarmuka pengguna ini:

* *Checkbox with delete button:*



Gambar 3. 14 Komponen *Checklist* dan *Delete Icon Button*

Komponen ini menggunakan interaksi *On Tap,* dimana apabila pengguna klik teks atau *checkbox,* tampilan teks dan *checkbox* akan berubah menjadi *state checked* atau *unchecked.* Sementara, apabila pengguna klik icon *trashcan*, item tersebut akan langsung hilang dari *shopping list.*.

Dalam *state unchecked* atau *default,* teks berwarna hitam dengan checkbox kosong yang memiliki *border* hitam. Sementara dalam *state checked,* teks berwarna hitam akan menambahkan efek *strikethrough* dan checkbox warna hijau dengan *icon* centang.

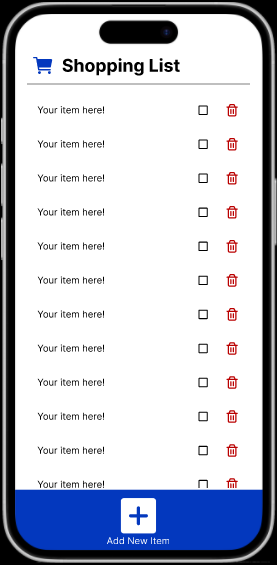
* *Simple textbox:*



Gambar 3. 15 Komponen *Textbox*

Komponen ini akan memberi efek textbox yang kedap kedip seakan menunggu pengguna untuk menginput nama item yang akan mereka tambahkan ke *shopping list.*

*Prototype* ini dirancang mengikuti spesifikasi ukuran iPhone 16. Berikut tampilan *prototype widget checklist* setelah dijalankan:



Gambar 3. 16 Tampilan *UI* Aplikasi *Shopping List* setelah dijalankan pada *Figma*

Berikut tautan untuk *file Figma* ini:

<https://www.figma.com/design/XGqArhQsZR8edX2HimRZst/Shopping-LIst?node-id=0-1&t=zxkXO6tU4Gq2tAQ9-1>

## Coding Layout Aplikasi Shopping List

Widget checklist ini dibangun menggunakan bahasa pemrograman Dart dan *framework* Flutter dengan emulator *device* Pixel 6a API 33Android 13.0 (“Tiramisu”) pada Android Studio.

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 theme: ThemeData(  
 primaryColor: Colors.*blueAccent*,  
 ), // ThemeData  
 home: const MyHomePage(title: 'Project #2'),  
 ); // MaterialApp  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key, required this.title});  
  
 final String title;  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 List<Map> categories = [  
 {"name": "Broom", "isChecked": false},  
 {"name": "Minced Beef", "isChecked": false},  
 {"name": "Sesame Oil", "isChecked": false},  
 {"name": "Mouse Trap", "isChecked": false},  
 {"name": "Spoons", "isChecked": false},  
 {"name": "Cabinet", "isChecked": false},  
 {"name": "Apple", "isChecked": false},  
 {"name": "Wipes", "isChecked": false},  
 {"name": "Puzzle", "isChecked": false},  
 {"name": "A4 Paper", "isChecked": false},  
 {"name": "Cinnamon", "isChecked": false},  
 {"name": "Chili Flakes", "isChecked": false},  
 {"name": "Toilet Paper", "isChecked": false},  
 ];  
  
 void \_deleteItem(int index) {  
 setState(() {  
 categories.removeAt(index);  
 });  
 }  
  
 void showOverlay(BuildContext context) {  
 late OverlayEntry overlayEntry;  
 overlayEntry = OverlayEntry(  
 builder: (context) => Stack(  
 children: [  
 Positioned.fill(  
 child: GestureDetector(  
 onTap: () {  
 overlayEntry.remove();  
 },  
 child: Container(  
 color: Colors.*black*.withOpacity(0.5),  
 ), // Container  
 ), // Gesture Detector  
 ), // Positioned.fill  
 // Pop-up Dialog  
 Positioned(  
 top: MediaQuery.*of*(context).size.height \* 0.4,  
 left: MediaQuery.*of*(context).size.width \* 0.1,  
 child: Material(  
 color: Colors.*transparent*,  
 child: Container(  
 width: MediaQuery.*of*(context).size.width \* 0.8,  
 height: MediaQuery.*of*(context).size.height \* 0.24,  
 padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 12),  
 decoration: BoxDecoration(  
 color: Colors.*white*,  
 borderRadius: BorderRadius.circular(12),  
 ), // BoxDecoration  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 const Row(  
 children: [  
 Icon(  
 Icons.*add\_circle\_outline*,  
 size: 32.0,  
 color: Colors.*blueAccent*,  
 ), // Icon  
 SizedBox(width: 10),  
 Text(  
 "Add New Item",  
 style: TextStyle(  
 fontSize: 24,  
 ), // TextStyle  
 ), // Text  
 ],  
 ), // Row  
 IconButton(  
 icon: const Icon(  
 Icons.*close*,  
 color: Colors.*red*,  
 size: 18.0,  
 ), // Icon  
 onPressed: () {  
 overlayEntry.remove();  
 },  
 ), // IconButton  
 ],  
 ), // Row  
 const Divider(  
 color: Colors.*black*,  
 ), // Divider  
 const SizedBox(  
 height: 7.0,  
 ), // SizedBox  
 const TextField(  
 decoration: InputDecoration(  
 enabledBorder: OutlineInputBorder(  
 borderSide: BorderSide(color: Colors.*blueAccent*),  
 ), // OutlineInputBorder  
 focusedBorder: OutlineInputBorder(  
 borderSide: BorderSide(color: Colors.*blue*),  
 ), // OutlineInputBorder  
 border: OutlineInputBorder(),  
 hintText: 'Enter things to the shopping list here!',  
 ), // InputDecoration  
 cursorColor: Colors.*blueAccent*,  
 style: TextStyle(  
 color: Colors.*blueAccent* ), // TextStyle  
 ), // Text  
 const SizedBox(  
 height: 6.0,  
 ), // SizedBox  
 ElevatedButton(  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*blueAccent*,  
 foregroundColor: Colors.*white*,  
 minimumSize: const Size.fromHeight(40),  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(5)  
 ) // RoundedRectangleBorder  
 ),  
 onPressed: () {},  
 child: const Text(  
 'Save',  
 ), // Text  
 ), // ElevatedButton  
 ],  
 ), // Column  
 ), // Container  
 ), // Material  
 ), // Positioned  
 ],  
 ), // Stack  
 ); // OverlayEntry  
 Overlay.*of*(context).insert(overlayEntry);  
 }  
  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text(widget.title),  
 ), // AppBar  
 body: SingleChildScrollView(  
 child: Padding(  
 padding: const EdgeInsets.all(20),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 const Row(  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: [  
 Icon(  
 Icons.*shopping\_cart*,  
 size: 40.0,  
 color: Colors.*blueAccent*,  
 ),  
 SizedBox(  
 width: 14.0,  
 ), // SixedBox  
 Text(  
 "Shopping List",  
 style: TextStyle(  
 fontSize: 32,  
 fontWeight: FontWeight.*bold*,  
 ), // TextStyle  
 ), // Text  
 ],  
 ), // Row  
 const Divider(  
 color: Colors.*black*,  
 ), // Divider  
 Column(  
 children: categories.asMap().entries.map((entry) {  
 int index = entry.key;  
 Map favorite = entry.value;  
 return ListTile(  
 contentPadding: EdgeInsets.*zero*,  
 title: Row(  
 children: [  
 Expanded(  
 child: CheckboxListTile(  
 activeColor: Colors.*green*,  
 onChanged: (val) {  
 setState(() {  
 favorite['isChecked'] = val ?? false;  
 });  
 },  
 value: favorite['isChecked'],  
 title: Text(  
 favorite['name'] ?? '',  
 style: TextStyle(  
 fontSize: 20,  
 decoration: favorite['isChecked']  
 ? TextDecoration.*lineThrough* : TextDecoration.*none*,  
 ), // TextStyle  
 ), // Text  
 ), // CheckboxListTile  
 ), // Expanded  
 IconButton(  
 icon: const Icon(Icons.*delete*, color: Colors.*red*),  
 onPressed: () => \_deleteItem(index),  
 ), // IconButton  
 ],  
 ), // Row  
 ); // ListTile  
 }).toList(),  
 ), // Column  
 ],  
 ), // Column  
 ), // Padding  
 ), // SingleChildScrollView  
 bottomNavigationBar: Container(  
 height: 100,  
 color: Colors.*blueAccent*,  
 child: InkWell(  
 onTap: () => showOverlay(context),  
 child:  
 Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Container(  
 margin: const EdgeInsets.only(top: 3.0),  
 child: const Icon(  
 Icons.*add\_box*,  
 color: Colors.*white*,  
 size: 70.0,  
 ), // Icon  
 ), // Container  
 const Text(  
 'Add New Item',  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 16,  
 fontWeight: FontWeight.*bold*,  
 ), // TextStyle  
 ), // Text  
 ], // <Widget>[]  
 ), // Column  
 ), // InkWell  
 ), // Container  
 ); // Scaffold  
 }  
}

Perintah import digunakan untuk mengimpor semua widget dan *property* dari Material Design hingga dapat digunakan dalam proyek ini.

import 'package:flutter/material.dart';

Fungsi main() adalah titik masuk utama aplikasi. runApp(const MyApp()) menjalankan aplikasi dan menampilkan widget MyApp() sebagai root (akar) dari aplikasi.

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

Clas MyApp() adalah widget kelas yang menentukan tema dan struktur dasar aplikasi. Ini menggunakan widget MaterialApp untuk mengatur tema (warna utama biru) dan mengatur MyHomePage sebagai halaman utama aplikasi.

class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 theme: ThemeData(  
 primaryColor: Colors.*blueAccent*,  
 ), // ThemeData  
 home: const MyHomePage(title: ''),  
 ); // MaterialApp  
 }  
}

MyHomePage adalah widget stateful yang berisi daftar belanja. title di sini digunakan sebagai judul aplikasi di AppBar.

class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key, required this.title});  
  
 final String title;  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}

Di dalam kelas \_MyHomePageState, terdapat beberapa elemen dan fungsi untuk mengelola daftar belanja:

* Variabel categories, yaitu daftar item belanja, masing-masing berupa peta (map) dengan nama item dan status checkbox (isChecked), apakah item tersebut sudah dipilih atau belum.

class \_MyHomePageState extends State<MyHomePage> {  
 List<Map> categories = [  
 {"name": "Broom", "isChecked": false},  
 {"name": "Minced Beef", "isChecked": false},  
 {"name": "Sesame Oil", "isChecked": false},  
 {"name": "Mouse Trap", "isChecked": false},  
 {"name": "Spoons", "isChecked": false},  
 {"name": "Cabinet", "isChecked": false},  
 {"name": "Apple", "isChecked": false},  
 {"name": "Wipes", "isChecked": false},  
 {"name": "Puzzle", "isChecked": false},  
 {"name": "A4 Paper", "isChecked": false},  
 {"name": "Cinnamon", "isChecked": false},  
 {"name": "Chili Flakes", "isChecked": false},  
 {"name": "Toilet Paper", "isChecked": false},  
 ];

* Fungsi \_deleteItem menghapus item dari daftar berdasarkan indeksnya. setState digunakan untuk memperbarui UI setelah item dihapus.

void \_deleteItem(int index) {  
 setState(() {  
 categories.removeAt(index);  
 });  
 }

* Fungsi showOverlay menampilkan dialog pop-up overlay untuk menambah item baru. OverlayEntry() digunakan untuk membuat layer transparan dengan latar belakang semi-transparan dan kotak dialog di tengahnya.

void showOverlay(BuildContext context) {  
 late OverlayEntry overlayEntry;  
 overlayEntry = OverlayEntry(  
 builder: (context) => Stack(  
 children: [  
 Positioned.fill(  
 child: GestureDetector(  
 onTap: () {  
 overlayEntry.remove();  
 },  
 child: Container(  
 color: Colors.*black*.withOpacity(0.5),  
 ), // Container  
 ), // Gesture Detector  
 ), // Positioned.fill

* Dialog Pop-up ini berisi kolom input (TextField) untuk memasukkan nama item baru, dan tombol "Save". Jika pengguna klik pada area di luar dialog atau tombol close, overlay akan dihapus (overlayEntry.remove()).

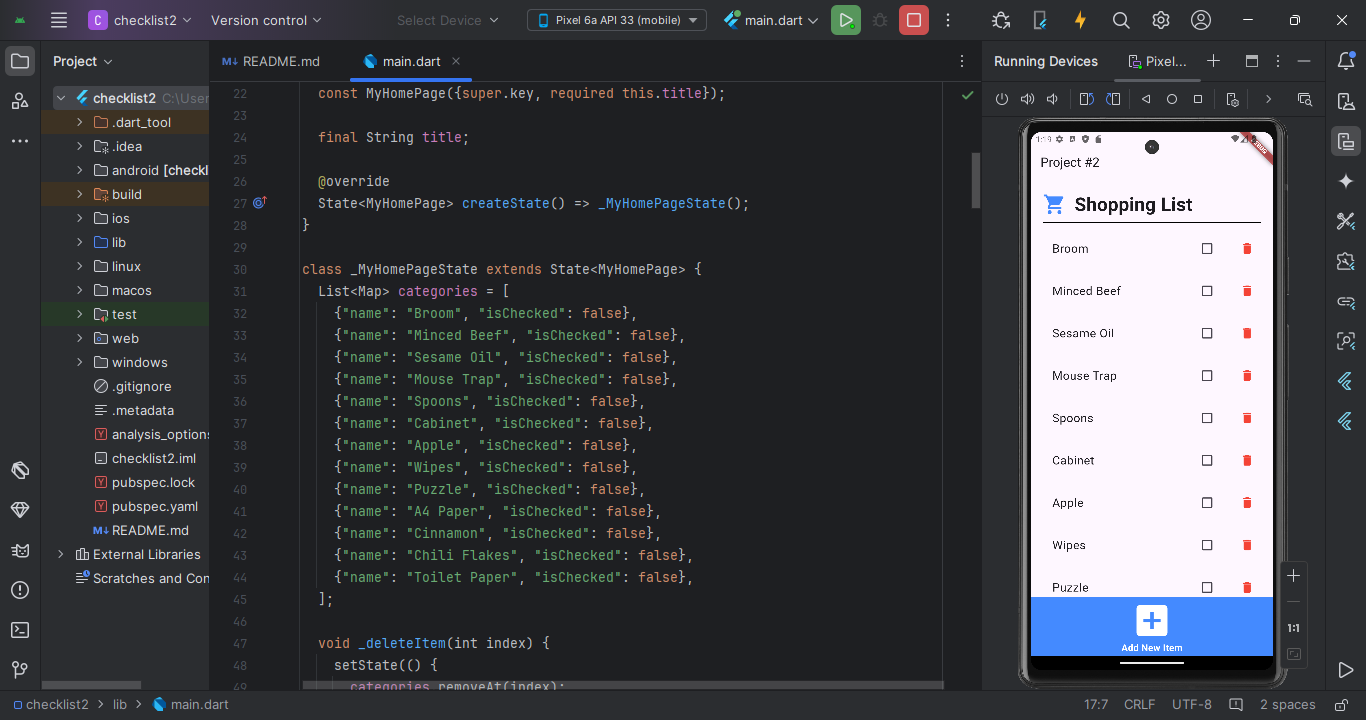
// Pop-up Dialog  
 Positioned(  
 top: MediaQuery.*of*(context).size.height \* 0.4,  
 left: MediaQuery.*of*(context).size.width \* 0.1,  
 child: Material(  
 color: Colors.*transparent*,  
 child: Container(  
 width: MediaQuery.*of*(context).size.width \* 0.8,  
 height: MediaQuery.*of*(context).size.height \* 0.24,  
 padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 12),  
 decoration: BoxDecoration(  
 color: Colors.*white*,  
 borderRadius: BorderRadius.circular(12),  
 ), // BoxDecoration  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 const Row(  
 children: [  
 Icon(  
 Icons.*add\_circle\_outline*,  
 size: 32.0,  
 color: Colors.*blueAccent*,  
 ), // Icon  
 SizedBox(width: 10),  
 Text(  
 "Add New Item",  
 style: TextStyle(  
 fontSize: 24,  
 ), // TextStyle  
 ), // Text  
 ],  
 ), // Row  
 IconButton(  
 icon: const Icon(  
 Icons.*close*,  
 color: Colors.*red*,  
 size: 18.0,  
 ), // Icon  
 onPressed: () {  
 overlayEntry.remove();  
 },  
 ), // IconButton  
 ],  
 ), // Row  
 const Divider(  
 color: Colors.*black*,  
 ), // Divider  
 const SizedBox(  
 height: 7.0,  
 ), // SizedBox  
 const TextField(  
 decoration: InputDecoration(  
 enabledBorder: OutlineInputBorder(  
 borderSide: BorderSide(color: Colors.*blueAccent*),  
 ), // OutlineInputBorder  
 focusedBorder: OutlineInputBorder(  
 borderSide: BorderSide(color: Colors.*blue*),  
 ), // OutlineInputBorder  
 border: OutlineInputBorder(),  
 hintText: 'Enter things to the shopping list here!',  
 ), // InputDecoration  
 cursorColor: Colors.*blueAccent*,  
 style: TextStyle(  
 color: Colors.*blueAccent* ), // TextStyle  
 ), // Text  
 const SizedBox(  
 height: 6.0,  
 ), // SizedBox  
 ElevatedButton(  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*blueAccent*,  
 foregroundColor: Colors.*white*,  
 minimumSize: const Size.fromHeight(40),  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(5)  
 ) // RoundedRectangleBorder  
 ),  
 onPressed: () {},  
 child: const Text(  
 'Save',  
 ), // Text  
 ), // ElevatedButton  
 ],  
 ), // Column  
 ), // Container  
 ), // Material  
 ), // Positioned  
 ],  
 ), // Stack  
 ); // OverlayEntry  
 Overlay.*of*(context).insert(overlayEntry);  
 }

Method build menampilkan elemen UI utama aplikasi dengan menggunakan Scaffold(). Berikut beberapa bagian penting dari UI ini:

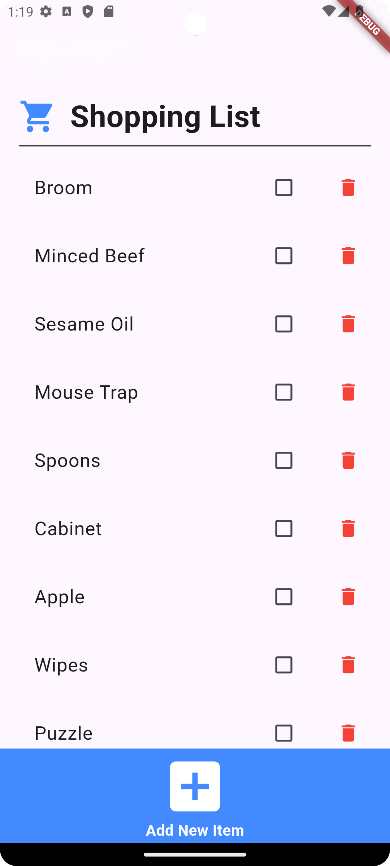
* AppBar: Menampilkan judul aplikasi sesuai properti title.
* Daftar item pada *shopping list*: Ditampilkan menggunakan ListTile, yang memungkinkan pengguna untuk mencentang setiap item dengan checkbox dan menghapus item melalui tombol delete.
* bottomNavigationBar: Ini adalah tombol di bagian bawah layar yang, ketika ditekan, memanggil fungsi showOverlay() untuk menampilkan dialog tambah item baru.

@override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text(widget.title),  
 ), // AppBar  
 body: SingleChildScrollView(  
 child: Padding(  
 padding: const EdgeInsets.all(20),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 const Row(  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: [  
 Icon(  
 Icons.*shopping\_cart*,  
 size: 40.0,  
 color: Colors.*blueAccent*,  
 ),  
 SizedBox(  
 width: 14.0,  
 ), // SixedBox  
 Text(  
 "Shopping List",  
 style: TextStyle(  
 fontSize: 32,  
 fontWeight: FontWeight.*bold*,  
 ), // TextStyle  
 ), // Text  
 ],  
 ), // Row  
 const Divider(  
 color: Colors.*black*,  
 ), // Divider  
 Column(  
 children: categories.asMap().entries.map((entry) {  
 int index = entry.key;  
 Map favorite = entry.value;  
 return ListTile(  
 contentPadding: EdgeInsets.*zero*,  
 title: Row(  
 children: [  
 Expanded(  
 child: CheckboxListTile(  
 activeColor: Colors.*green*,  
 onChanged: (val) {  
 setState(() {  
 favorite['isChecked'] = val ?? false;  
 });  
 },  
 value: favorite['isChecked'],  
 title: Text(  
 favorite['name'] ?? '',  
 style: TextStyle(  
 fontSize: 20,  
 decoration: favorite['isChecked']  
 ? TextDecoration.*lineThrough* : TextDecoration.*none*,  
 ), // TextStyle  
 ), // Text  
 ), // CheckboxListTile  
 ), // Expanded  
 IconButton(  
 icon: const Icon(Icons.*delete*, color: Colors.*red*),  
 onPressed: () => \_deleteItem(index),  
 ), // IconButton  
 ],  
 ), // Row  
 ); // ListTile  
 }).toList(),  
 ), // Column  
 ],  
 ), // Column  
 ), // Padding  
 ), // SingleChildScrollView  
 bottomNavigationBar: Container(  
 height: 100,  
 color: Colors.*blueAccent*,  
 child: InkWell(  
 onTap: () => showOverlay(context),  
 child:  
 Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Container(  
 margin: const EdgeInsets.only(top: 3.0),  
 child: const Icon(  
 Icons.*add\_box*,  
 color: Colors.*white*,  
 size: 70.0,  
 ), // Icon  
 ), // Container  
 const Text(  
 'Add New Item',  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 16,  
 fontWeight: FontWeight.*bold*,  
 ), // TextStyle  
 ), // Text  
 ], // <Widget>[]  
 ), // Column  
 ), // InkWell  
 ), // Container  
 ); // Scaffold  
 }  
}

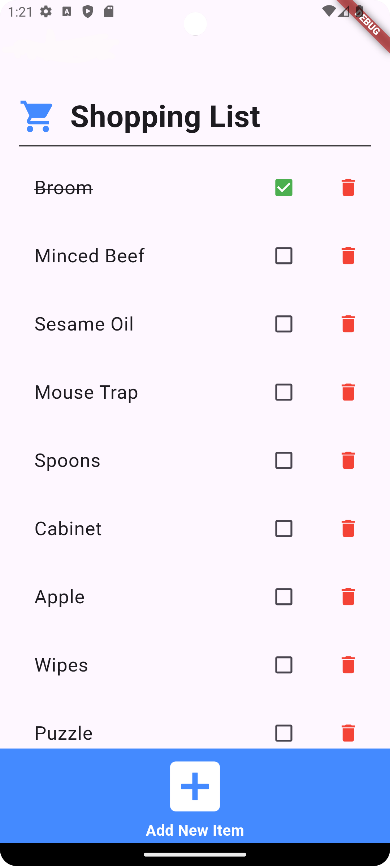
Berikut hasil akhir layout aplikasi *shopping checklist* yang telah dirancang pada Android Studio:



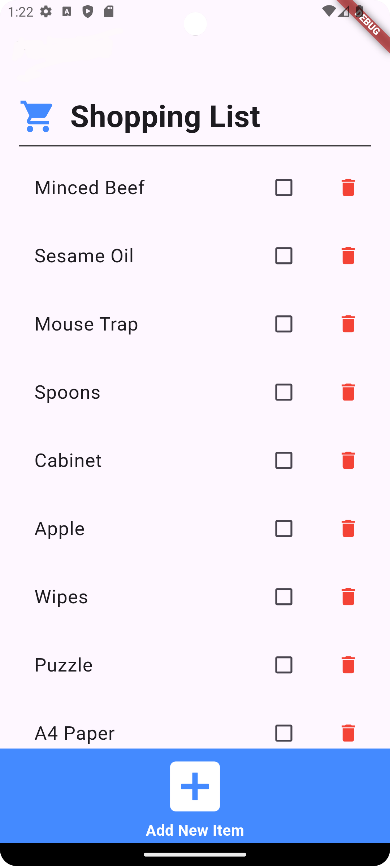
Gambar 3. 17 *Coding* Aplikasi *Layout Shopping List* pada *Android Studio*



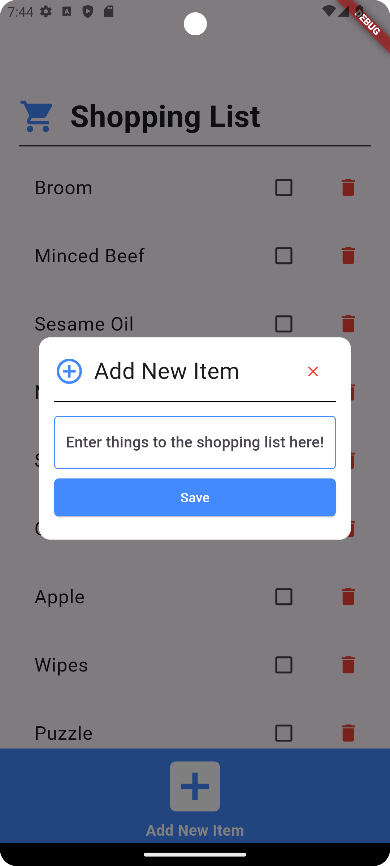
Gambar 3. 18 *Screenshot* Aplikasi *Shopping List* *Default* pada *Android Studio*



Gambar 3. 19 Screenshot Aplikasi *Shopping List Item Checked* pada *Android Studio*



Gambar 3. 20 Screenshot Aplikasi *Shopping List Item Deleted* pada *Android Studio*



Gambar 3. 21 Screenshot *Overlay Form* Menambah Item Baru pada Aplikasi *Shopping List*

# DAFTAR PUSTAKA

Dart. (2024, May 30). *Dart Overview*. Dart. https://dart.dev/overview

Putra, Y. W. S., Dawis, A. M., Novi, N., Natsir, F., Fitria, F., Widhiyanti, A. A. S., Hasan, F. N., Somantri, S., & Maniah, M. (2023). *PENGANTAR APLIKASI MOBILE* (W. Andriyani & Erlangga, Eds.; 1st ed.).

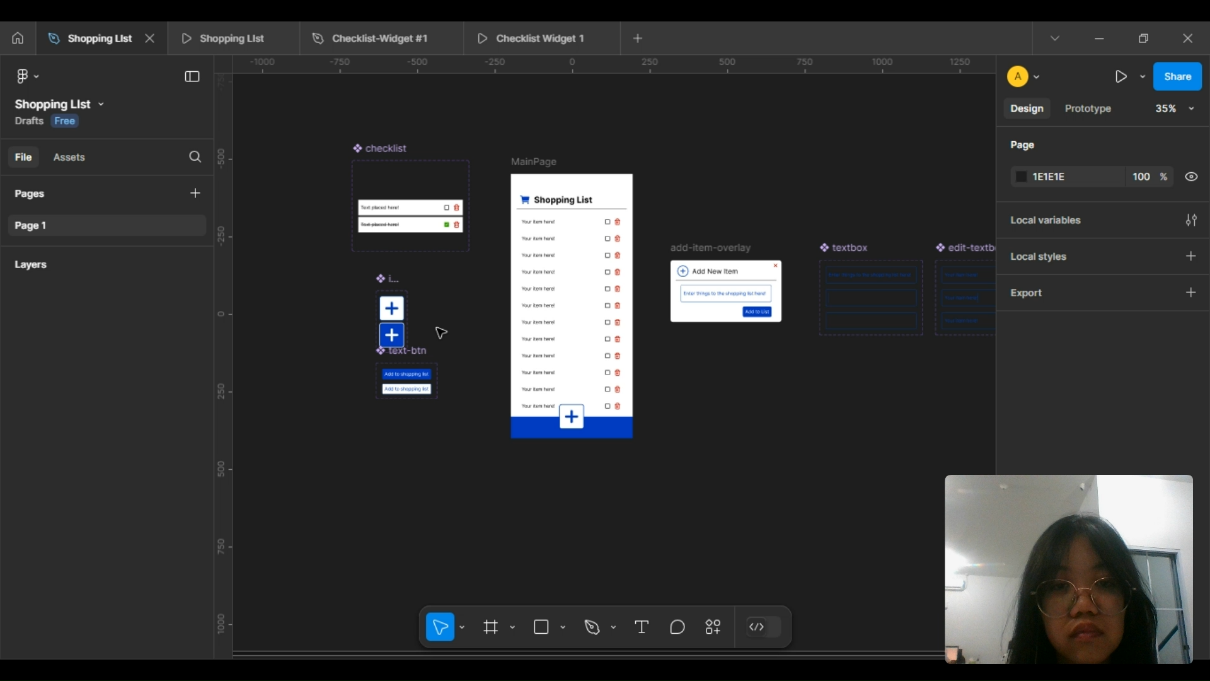
Raharjo, B. (2021). *Pemrograman Android Dengan Flutter*. Informatika Bandung.

Sakti Wibowo, H. (2023). *Desain Interaktif dengan Figma Panduan Praktis untuk Pemula dan Profesional* (W. Anita, Ed.; First Printing). Tiram Media.

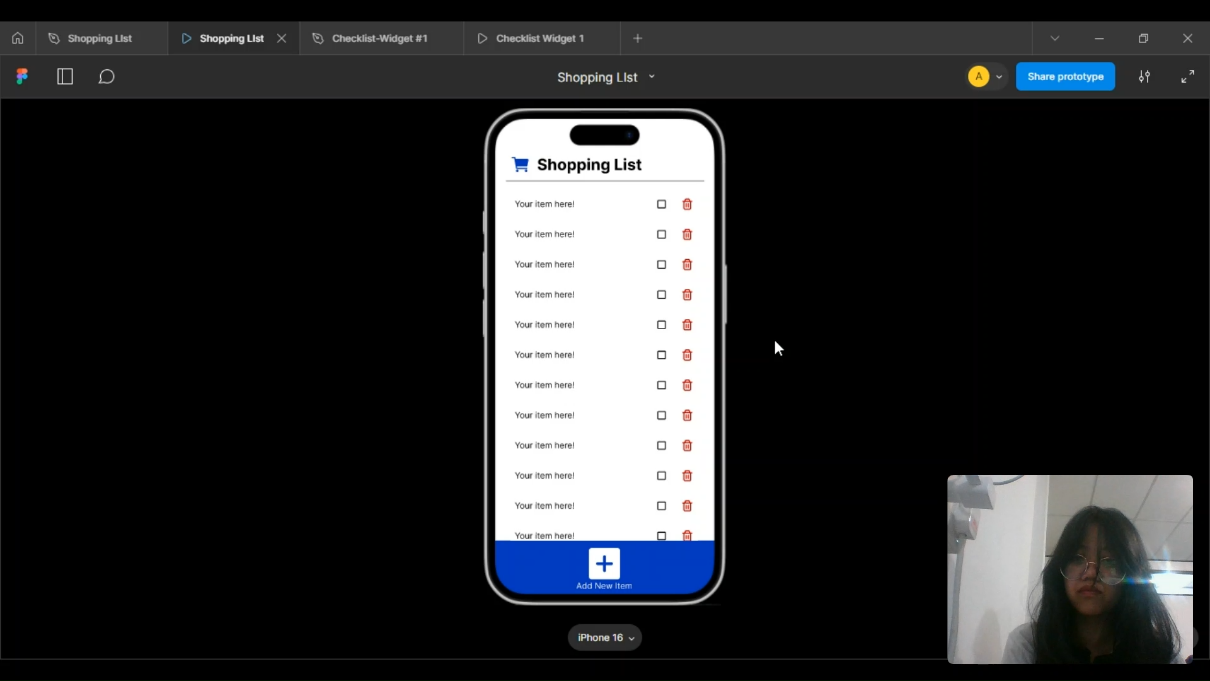
Suryana, D. (2018). *Android Studio Belajar Android Studio*. https://www.google.co.id/books/edition/Android\_Studio/wIhyDwAAQBAJ?hl=en&gbpv=0

Suryono, B. S., & Hardiansah. (2020). *Panduan Lengkap membuat aplikasi Android dan iOS dari Nol*.

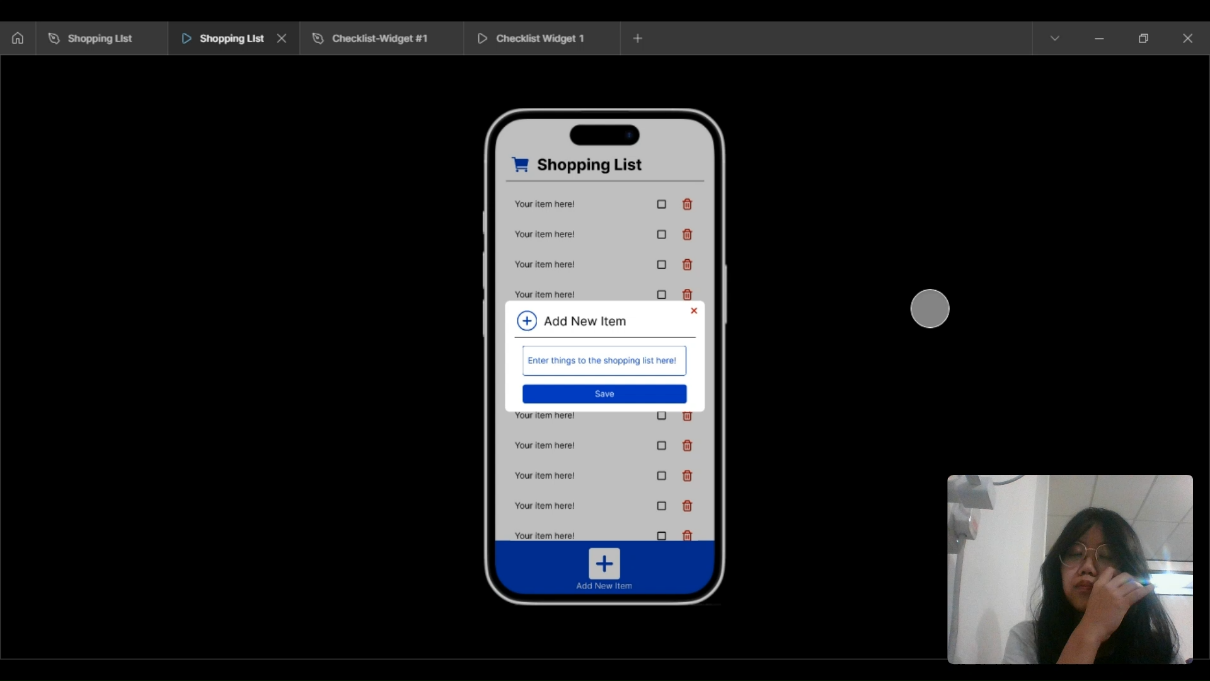
# LAMPIRAN



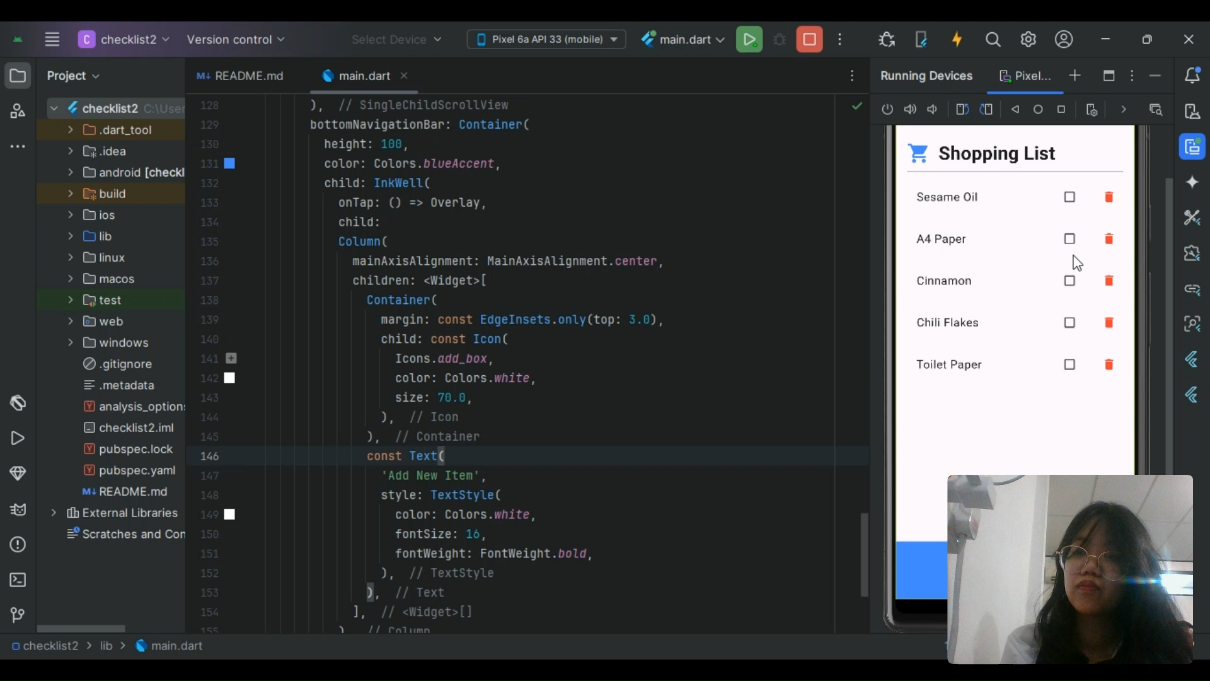
Lampiran 1. Desain Aplikasi *Shopping List* menggunakan *Figma*



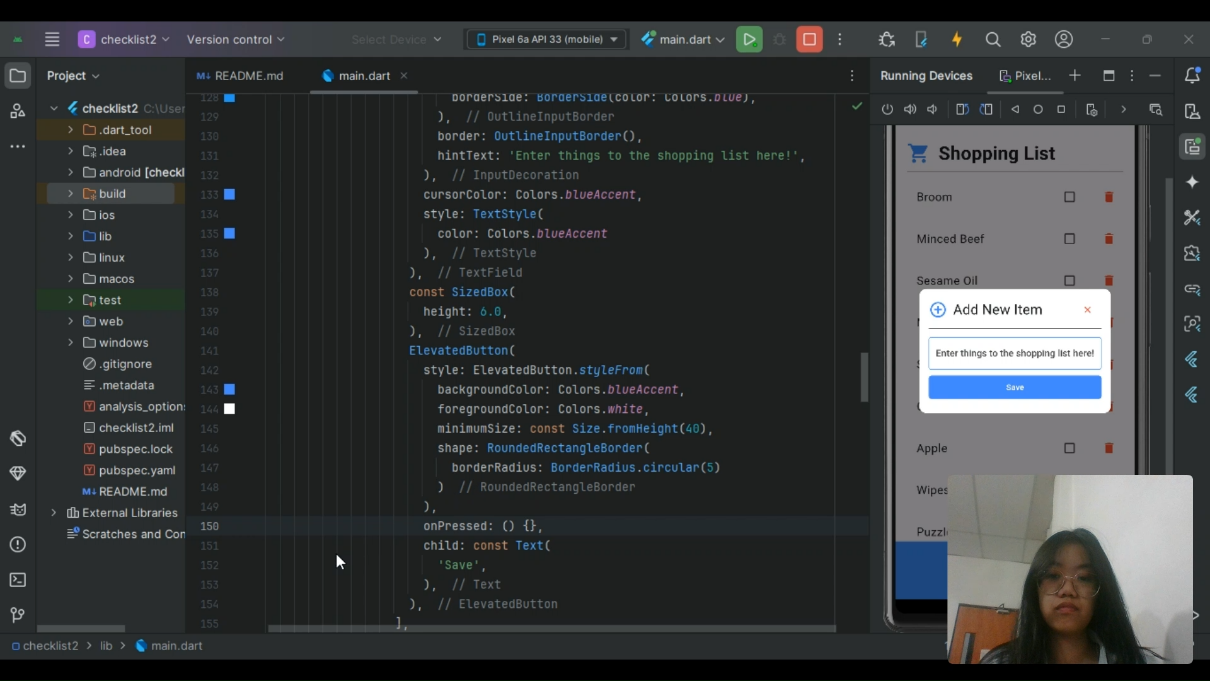
Lampiran 2. Menjalankan Desain *Layout* Aplikasi *Shopping List* pada *Figma*



Lampiran 3. Menjalankan Tampilan *Overlay Form Add New Item* pada *Figma*



Lampiran 4. *Coding* *Aplikasi Shopping List* menggunakan *Android Studio*



Lampiran 5. Aplikasi *Shopping List* menggunakan *Android Studio*