

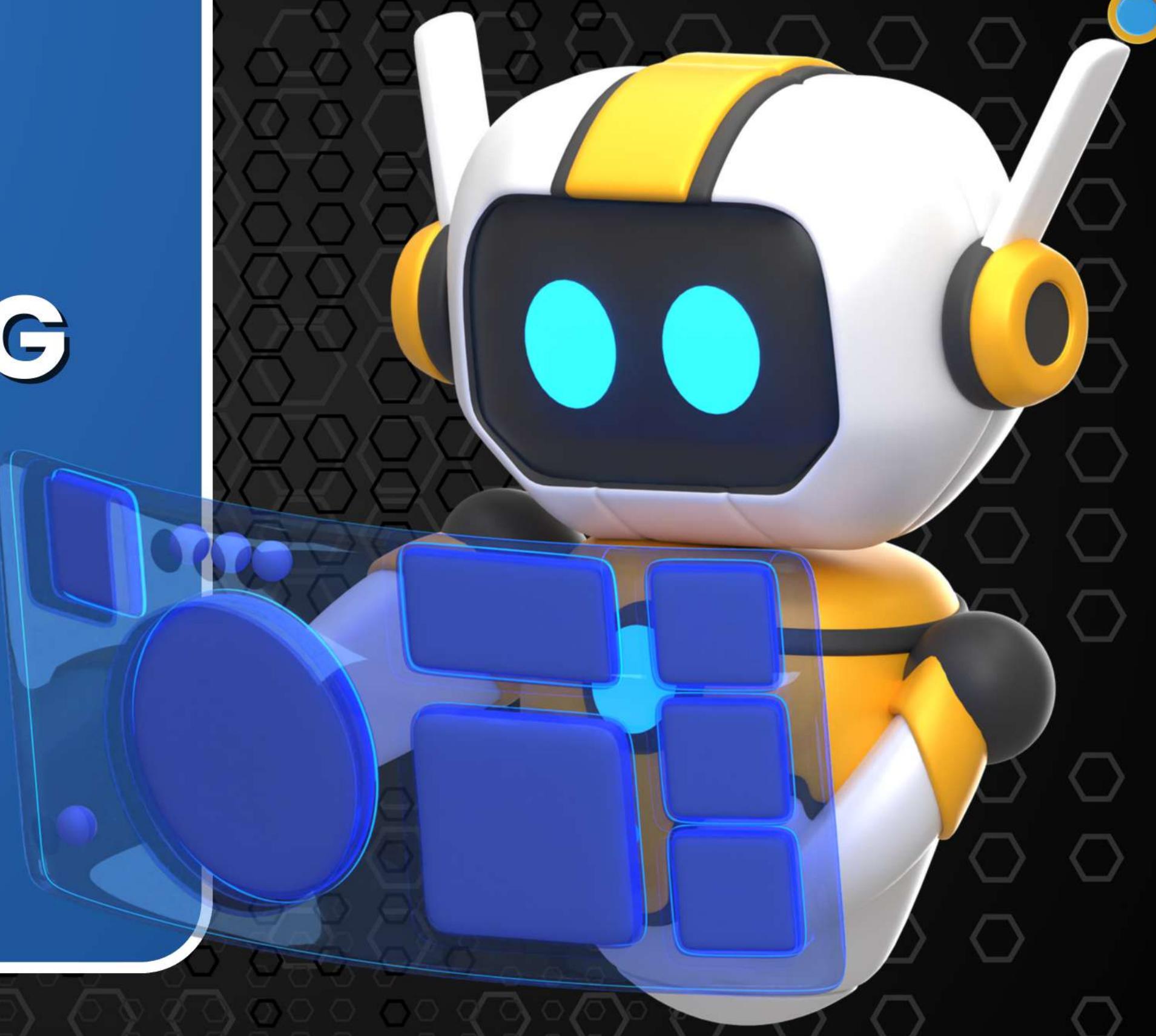


Universitas Pamulang
Sistem Informasi

MOBILE PROGRAMMING

PERTEMUAN 6

Yudisti Prayigo Permana, S.Kom., M.Kom.





TOPIK PEMBAHASAN

- Pengenalan CheckBox
- Fungsi CheckBox
- Cara Kerja CheckBox
- Flutter
- Kesimpulan

01
02
03
04
05





Apa itu CheckBox?

CheckBox merupakan komponen UI Android yang termasuk dalam keluarga widget compoundButton (seperti RadioButton, ToggleButton), CheckBox dapat dalam dua status: dicentang (Checked) atau tidak dicentang (Unchecked)





Apa itu CheckBox ?

CheckBox sangat berguna untuk :

1. Memungkinkan pengguna memilih satu atau lebih opsi dari sekumpulan pilihan yang tidak saling eksklusif
2. Mengaktifkan atau menonaktifkan suatu fitur atau pengaturan
3. Menyatakan persetujuan atau penerimaan terhadap suatu ketentuan





Atribut CheckBox

Saat menggunakan CheckBox terdapat berbagai atribut untuk mengontrol tampilan dan perilakunya. Berikut merupakan beberapa atribut yang paling umum dan penting :

- android:id – Untuk memberikan ID unik pada CheckBox. ID digunakan untuk mereferensikan CheckBox dari kode Java/Kotlin. Contoh :
`android:id="@+id/checkBoxSetuju"`
- android:layout_width – Menentukan lebar CheckBox
- android:layout_height – Menentukan tinggi CheckBox
- android:text – Teks yang ditampilkan disamping CheckBox. Contoh :
`android:text="Saya setuju dengan persyaratan"`
- android:checked – Atribut yang menentukan status awal CheckBox. true jika dicentang (checked) dan false jika tidak dicentang(unchecked)



Atribut CheckBox

- android:enabled – Atribut yang menentukan CheckBox dapat diinteraksikan oleh pengguna. True (default) berarti aktif, false berarti tidak aktif (biasanya abu-abu)
- android:textColor – Warna teks CheckBox. Bisa berupa nilai warna hex (#RRGGBB) atau referensi ke sumber daya warna
- android:textSize – Ukuran teks CheckBox. Biasanya dalam satuan sp (scale pixels)
- android:padding, android:paddingLeft, android:paddingRight, android:paddingTop, android:paddingBottom – Menentukan ruang disekitar teks dan kotak CheckBox
- android:drawableLeft, android:drawableRight, android:drawableTop, android:drawableBottom – Memungkinkan menambahkan gambar disisi teks CheckBox.
- android:buttonTint – Mengubah warna kotak centang
- android:onClick – Menentukan nama metode di Activity yang akan dipanggil Ketika CheckBox diklik.



Contoh Penggunaanya

Pada contoh disamping :

- Memiliki beberapa CheckBox dengan ID unik (checkboxJava, checkboxKotlin, checkboxPython, checkboxCplusplus)
- Masing-masing CheckBox memiliki teks yang ditampilkan disampingnya
- checkboxJava diatur untuk dicentang secara default (android:checked="true")
- checkboxCplusplus diatur agar tidak dapat diinteraksikan (android:enabled="false")
- Ada juga sebuah Button untuk memicu aksi setelah pengguna memilih opsi

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pilih Bahasa Pemrograman:"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginBottom="8dp"/>

    <CheckBox
        android:id="@+id/checkBoxJava"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Java"
        android:checked="true"/>

    <CheckBox
        android:id="@+id/checkBoxKotlin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kotlin"/>

    <CheckBox
        android:id="@+id/checkBoxPython"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Python"/>

    <CheckBox
        android:id="@+id/checkBoxCplusplus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="C++"
        android:enabled="false"/>

    <Button
        android:id="@+id/buttonSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit Pilihan"
        android:layout_marginTop="16dp"/>

</LinearLayout>
```



Contoh Penggunaanya

Penjelasan Kode :

- Di dalam metode `onCreate()`, kita mendapatkan referensi ke setiap CheckBox dan Button menggunakan `findViewById()`
- Untuk setiap CheckBox, kita memasang `OnCheckedChangeListener`. Listener ini akan dipanggil setiap kali status centang CheckBox berubah.
- Untuk Button, kita memasang `OnClickListener` yang akan dipanggil Ketika tombol diklik.

```
// Mendapatkan referensi ke element UI
checkBoxJava = findViewById(R.id.checkBoxJava);
checkBoxKotlin = findViewById(R.id.checkBoxKotlin);
checkBoxPython = findViewById(R.id.checkBoxPython);
buttonSubmit = findViewById(R.id.buttonSubmit);
textViewHasil = findViewById(R.id.textViewHasil); // Asumsikan ada TextView di layout

// Menangani perubahan status CheckBox menggunakan setOnCheckedChangeListener
checkBoxJava.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        String message = isChecked ? "Java dipilih" : "Java tidak dipilih";
        Toast.makeText(MainActivity.this, message, Toast.LENGTH_SHORT).show();
    }
});

checkBoxKotlin.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        String message = isChecked ? "Kotlin dipilih" : "Kotlin tidak dipilih";
        Toast.makeText(MainActivity.this, message, Toast.LENGTH_SHORT).show();
    }
});

checkBoxPython.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        String message = isChecked ? "Python dipilih" : "Python tidak dipilih";
        Toast.makeText(MainActivity.this, message, Toast.LENGTH_SHORT).show();
    }
});

// Menangani Klik pada tombol Submit
buttonSubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        List<String> selectedLanguages = new ArrayList<>();
        if (checkBoxJava.isChecked()) {
            selectedLanguages.add("Java");
        }
        if (checkBoxKotlin.isChecked()) {
            selectedLanguages.add("Kotlin");
        }
        if (checkBoxPython.isChecked()) {
            selectedLanguages.add("Python");
        }

        String hasil = selectedLanguages.isEmpty() ?
            "Tidak ada bahasa yang dipilih" :
            "Bahasa yang dipilih: " + String.join(", ", selectedLanguages);
        textViewHasil.setText(hasil);
    }
});
}
```



Flutter

Dalam Flutter, CheckBox merupakan sebuah widget yang memungkinkan pengguna untuk memilih atau tidak memilih suatu opsi. Secara visual, CheckBox ditampilkan sebagai kotak ketik yang dapat dicentang (checked) atau tidak dicentang (unchecked).





Properti Pada CheckBox

Widget CheckBox memiliki beberapa property penting yang memungkinkan untuk mengontrol tampilan dan perilakunya :

- Value – Sebuah nilai yang menentukan apakah CheckBox saat ini dicentang true atau tidak false. Properti ini harus dikelola oleh State dari StatefulWidget
- onChanged – Sebuah fungsi callback yang dipanggil Ketika nilai CheckBox berubah (Ketika pengguna mengetuknya).
- tristate – Sebuah nilai yang menentukan CheckBox dapat memiliki tiga status : true (dicentang), false (tidak centang) dan null (indeterminate).
- activeColor – Warna yang digunakan untuk mengisi kotak centang dan tanda centang saat CheckBox dicentang
- checkColor – Warna dari ikon centang itu sendiri saat CheckBox dicentang



Properti Pada CheckBox

- inactiveColor – Warna dari border kotak centang saat CheckBox tidak dicentang
- side – Properti yang memungkinkan menyesuaikan border CheckBox saat tidak dicentang menggunakan objek BorderSide
- shape – Bentuk dari kotak centang. Dapat menggunakan RoundedRetangleBorder untuk memberikan sudut yang melengkung
- visualDensity – Mengontrol kepadatan visual widget. Ini dapat mempengaruhi ukuran dan spasi CheckBox
- materialTapTargetSize – Mengontrol ukuran area sentuh minimum untuk widget
- focusNode – Mengelola focus keyboard untuk CheckBox
- autofocus – Jika true, widget ini akan mendapatkan focus awal saat widget dibangun
- mouseCursor – Kursor mouse yang ditampilkan saat pointer mouse berada di atas CheckBox



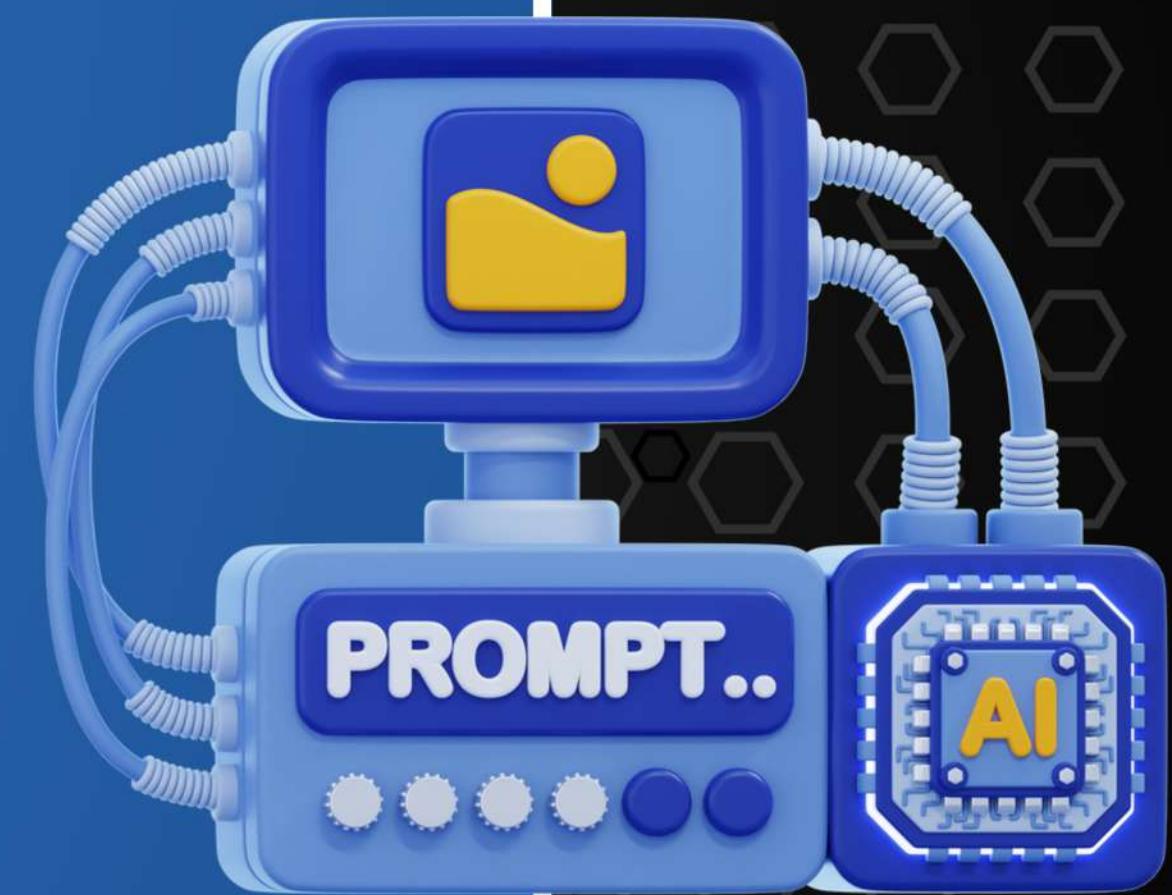
Properti Pada CheckBox

- overlayColor – Warna overlay saat CheckBox dalam keadaan focus atau ditekan
- splashRadius – Radius dari efek ripple saat CheckBox ditekan
- fillColor – Mengontrol warna latar belakang kotak centang berdasarkan statusnya (dicentang, tidak dicentang, indeterminate). Ini menggunakan MaterialStateProperty<Color?>



Cara Menggunakan CheckBox

1. Buat StatefulWidget – Widget induk yang akan menampung CheckBox dan mengelola statusnya
2. Deklarasikan Variabel Status – Didalam kelas _State dari StatefulWidget, deklarasikan sebuah variable boolean (atau nullable Boolean jika tristate merupakan true untuk menyimpan status CheckBox)
3. Gunakan Widget CheckBox - Didalam metode build() dari _State, gunakan widget CheckBox
4. Hubungan Properti value – Set property value dari CheckBox ke variable status yang telah di deklarasikan





Cara Menggunakan CheckBox

1. Buat StatefulWidget – Widget induk yang akan menampung CheckBox dan mengelola statusnya
2. Deklarasikan Variabel Status – Didalam kelas _State dari StatefulWidget, deklarasikan sebuah variable boolean (atau nullable Boolean jika tristate merupakan true untuk menyimpan status CheckBox)
3. Gunakan Widget CheckBox - Didalam metode build() dari _State, gunakan widget CheckBox
4. Hubungan Properti value – Set property value dari CheckBox ke variable status yang telah di deklarasikan
5. Implementasikan onChanged – Berikan sebuah fungsi callback ke property onChanged. Didalam callback ini :
 - Terima nilai Boolean baru dari CheckBox
 - Gunakan setState() untuk memperbarui variable status dengan nilai baru. Ini akan memicu Pembangunan ulang UI dan memperbarui tampilan CheckBox
6. Opsi Gunakan Row untuk teks disampingnya – Biasanya, akan menempatkan CheckBox di dalam Row Bersama dengan widget Text yang menjelaskan opsi yang dapat dipilih



ANDROID VS FLUTTER

| Fitur / Aspek | Android (Java/Kotlin) | Flutter (Dart) |
|--------------------|---|--|
| Bahasa Pemrograman | Java / Kotlin | Dart |
| Struktur UI | XML + Activity/Fragment | Langsung dalam kode Dart (Widget Tree) |
| Widget/Komponen | CheckBox (XML) + Java/Kotlin logic | Checkbox atau CheckboxListTile |
| Binding data | Manual (via <code>setOnCheckedChangeListener</code>) | Reactive (via <code>setstate</code>) |
| Built-in label | Tidak (label pakai <code>TextView</code> terpisah) | Ya, dengan <code>CheckboxListTile</code> |



ANDROID VS FLUTTER

| Aspek | Android | Flutter |
|-------------------|--|--|
| Struktur UI | Terpisah antara XML dan logic | Menjadi satu dalam widget |
| State Management | Manual, event listener diatur terpisah | Reactive dengan <code>setState</code> atau provider |
| UI Builder | Deklaratif XML, imperative logic | Full deklaratif |
| Label di Checkbox | Harus pakai <code>TextView</code> terpisah | Built-in via <code>CheckboxListTile</code> |
| Layouting | <code>LinearLayout</code> , <code>ConstraintLayout</code> , dll. | <code>Row</code> , <code>Column</code> , <code>Stack</code> , dll. |
| Cross-platform? | No (Android only) | Yes (Android, iOS, web, desktop) |



ANDROID VS FLUTTER

```
1 <LinearLayout
2     android:layout_width="match_parent"
3     android:layout_height="wrap_content"
4     android:orientation="horizontal">
5
6     <CheckBox
7         android:id="@+id/checkbox_news"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content" />
10
11    <TextView
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="Berlangganan newsletter" />
15 </LinearLayout>
```

```
1 class _CheckboxExampleState extends State<CheckboxExample> {
2     bool isChecked = false;
3
4     @override
5     Widget build(BuildContext context) {
6         return Scaffold(
7             appBar: AppBar(title: Text("Checkbox Flutter")),
8             body: CheckboxListTile(
9                 title: Text("Berlangganan newsletter"),
10                value: isChecked,
11                onChanged: (bool? value) {
12                    setState(() {
13                        isChecked = value!;
14                    });
15                    final message = value! ? "Subscribed" : "Unsubscribed";
16                    ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(message)));
17                },
18            ),
19        );
20    }
21 }
```



INSTALLATION

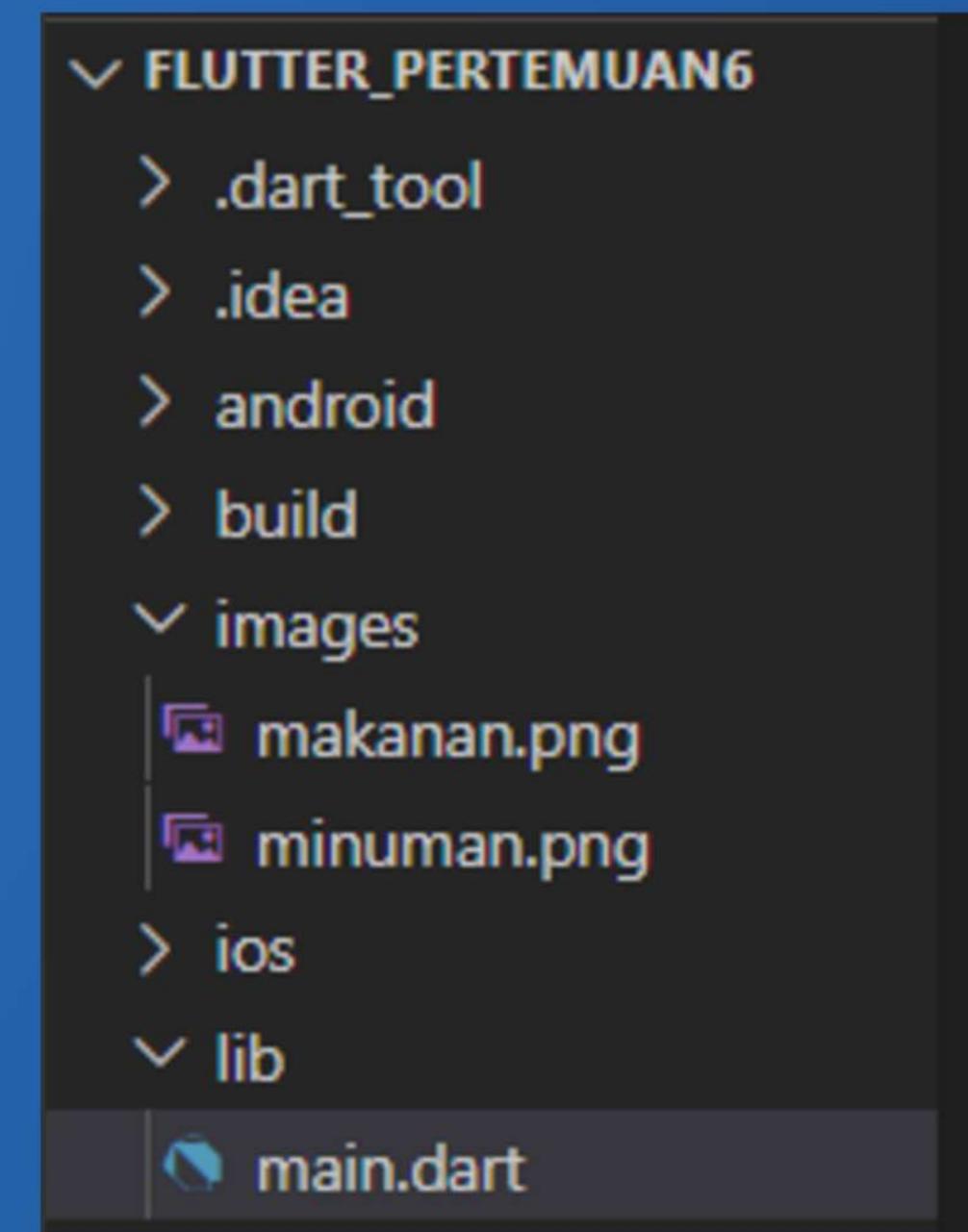
Pada pubspec.yaml tambahkan :

```
! pubspec.yaml
54 flutter:
61   # To add assets to your application, add an assets section like this:
62   assets:
63     - img/profil.png
64     # - images/a_dot_ham.jpeg
```

Buat susunan folder dan dua file seperti gambar disamping.

Dengan cara :

- Buat folder baru diluar folder lib beri nama foldernya dengan ‘images’
- Tambahkan dua gambar dengan nama ‘makanan.png’ dan ‘minuman.png’





SOURCE CODE

main.dart

```
 1 import 'package:flutter/material.dart';
 2
 3 void main() {
 4   runApp(const MyApp());
 5 }
 6
 7 class MyApp extends StatelessWidget {
 8   const MyApp({super.key});
 9
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       title: 'Restoku Menu',
14       debugShowCheckedModeBanner:
15       false, theme: ThemeData(
16         primarySwatch: Colors.blue,
17       ),
18       home: const CheckboxPage(),
19     );
20   }
21 }
```

```
 1 class CheckboxPage extends StatefulWidget {
 2   const CheckboxPage({super.key});
 3
 4   @override
 5   State<CheckboxPage> createState() => _CheckboxPageState();
 6 }
 7
 8 class _CheckboxPageState extends State<CheckboxPage> {
 9   bool minuman = false;
10   bool makanan = false;
11   List<String> menu = [];
12   final TextEditingController _namaController = TextEditingController();
13   final TextEditingController _mejaController = TextEditingController();
14
15   void updateMenu() {
16     menu.clear();
17     if (minuman) {
18       menu.addAll(['Teh Manis', 'Kopi', 'Air Mineral']);
19     }
20     if (makanan) {
21       menu.addAll(['Nasi Goreng', 'Kwetiau Goreng', 'Mie Tek-Tek']);
22     }
23   }
24
25   void _showConfirmationDialog() {
26     if (_namaController.text.isEmpty || _mejaController.text.isEmpty) {
27       ScaffoldMessenger.of(context).showSnackBar(
28         const SnackBar(content: Text('Harap isi nama dan nomor meja terlebih
29 dahulu')),
30         return;
31     }
32     if (menu.isEmpty) {
34       ScaffoldMessenger.of(context).showSnackBar(
35         const SnackBar(content: Text('Harap pilih minimal satu menu')),
36       );
37     }
38   }
39 }
```





SOURCE CODE

```
1 showDialog(  
2   context: context,  
3   builder: (BuildContext context) {  
4     return AlertDialog(  
5       title: const Text('Konfirmasi Pesanan'),  
6       content: Text('Apakah Anda yakin ingin memproses pesanan untuk ${_namaController.text} (Meja  
7 ${_mejaController.text})?'),  
8       TextButton(  
9         onPressed: () => Navigator.of(context).pop(),  
10        child: const Text('Tidak'),  
11      ),  
12      TextButton(  
13        onPressed: () {  
14          Navigator.of(context).pop();  
15          _processOrder();  
16        },  
17        child: const Text('Ya'),  
18      ),  
19    ],  
20  );  
21};  
22});  
23}  
24}
```



SOURCE CODE

```
 1 void _processOrder() {
 2     // Show toast message
 3     ScaffoldMessenger.of(context).showSnackBar(
 4         SnackBar(
 5             content: Text('Pesanan ${_namaController.text} (Meja ${_mejaController.text}) sedang
 6 diproses'), duration: const Duration(seconds: 2),
 7         ),
 8     );
 9
10    // Clear all form fields
11    setState(() {
12        _namaController.clear();
13        _mejaController.clear();
14        minuman = false;
15        makanan = false;
16        menu.clear();
17    });
18 }
19
20 @override
21 void dispose() {
22     _namaController.dispose();
23     _mejaController.dispose();
24     super.dispose();
25 }
26
```





SOURCE CODE

```
1  @override
2  Widget build(BuildContext context) {
3    return Scaffold(
4      appBar: AppBar(
5        title: const Align(
6          alignment: Alignment.centerLeft,
7          child: Text(
8            'Restoku Menu',
9            style: TextStyle(
10               fontSize: 20,
11               fontWeight: FontWeight.bold,
12             ),
13           ),
14         ),
15         backgroundColor: Colors.blueAccent[100],
16       ),
17       body: SingleChildScrollView(
18         padding: const EdgeInsets.all(16),
19         child: Column(
20           children: [
21             Container(
22               padding: const EdgeInsets.symmetric(horizontal: 16, vertical:
23 8),
24               decoration: BoxDecoration(
25                 color: Colors.blue,
26                 borderRadius: BorderRadius.circular(8),
27               ),
28               child: const Text(
29                 'Menu Restoranku',
30                 style: TextStyle(
31                   fontSize: 18,
32                   fontWeight: FontWeight.w600,
33                   color: Colors.white,
34                 ),
35               ),
36             ),
37           ],
38         ),
39       ),
40     );
41   }
42 }
```

```
1  const SizedBox(height: 20),
2  TextField(
3    controller: _namaController,
4    decoration: const InputDecoration(
5      labelText: 'Nama Pemesan',
6      border: OutlineInputBorder(),
7      prefixIcon: Icon(Icons.person),
8    ),
9  ),
10 const SizedBox(height: 15),
11 TextField(
12   controller: _mejaController,
13   decoration: const InputDecoration(
14     labelText: 'Nomor Meja',
15     border: OutlineInputBorder(),
16     prefixIcon: Icon(Icons.table_restaurant),
17   ),
18 ),
19 const SizedBox(height: 20),
20 Card(
21   child: SizedBox(
22     height: 70,
23     width: 250,
24     child: Row(
25       children: [
26         Checkbox(
27           value: minuman,
28           onChanged: (value) {
29             setState(() {
30               minuman = value!;
31               updateMenu();
32             });
33           },
34         ),
35         activeColor: Colors.lightBlueAccent,
36         checkColor: Colors.black,
37       ],
38     ),
39   ),
40   width: 100,
41   Image.asset('../images/minuman.png', width: 100, height:
42   const SizedBox(width: 15),
43   const Text('Minuman', style: TextStyle(fontSize: 18)),
44   ],
45   ),
46 );
```



SOURCE CODE

```
1      const SizedBox(height: 20),
2      Card(
3        child: SizedBox(
4          height: 70,
5          width: 250,
6          child: Row(
7            children: [
8              Checkbox(
9                value: makanan,
10               onChanged: (value) {
11                 setState(() {
12                   makanan = value!;
13                   updateMenu();
14                 });
15               },
16               activeColor: Colors.lightBlueAccent,
17               checkColor: Colors.black,
18             ),
19             const SizedBox(width: 10),
20             Image.asset('../images/makanan.png', width: 100, height:
21             100),
22             const SizedBox(width: 15),
23             const Text('Makanan', style: TextStyle(fontSize: 18)),
24           ],
25         ),
26       ),
27     ),
28   const SizedBox(height: 30),
29   Text(
30     menu.isEmpty
31       ? 'Tidak ada menu yang dipilih'
32       : 'Menu yang dipilih: ${menu.join(', ')}`,
33     style: const TextStyle(fontSize: 16),
34   ),
35   const SizedBox(height: 30),
36   ElevatedButton(
37     onPressed: _showConfirmationDialog,
38     style: ElevatedButton.styleFrom(
39       backgroundColor: Colors.blue,
40       padding: const EdgeInsets.symmetric(horizontal: 30, vertical:
41       15),
42     ),
43     child: const Text(
44       'Proses Pesanan',
45       style: TextStyle(fontSize: 16, color: Colors.white),
46     ),
47   ),
48   ],
49 ),
50 ),
51 );
52 }
53 }
```





HASIL

Restoku Menu

localhost:65453

Restoku Menu

Menu Restoranku

Nama Pemesan

Nomor Meja

Minuman

Makanan

Tidak ada menu yang dipilih

Proses Pesanan

Pesanan ANDI (Meja 125) sedang diproses





Thank You!

