

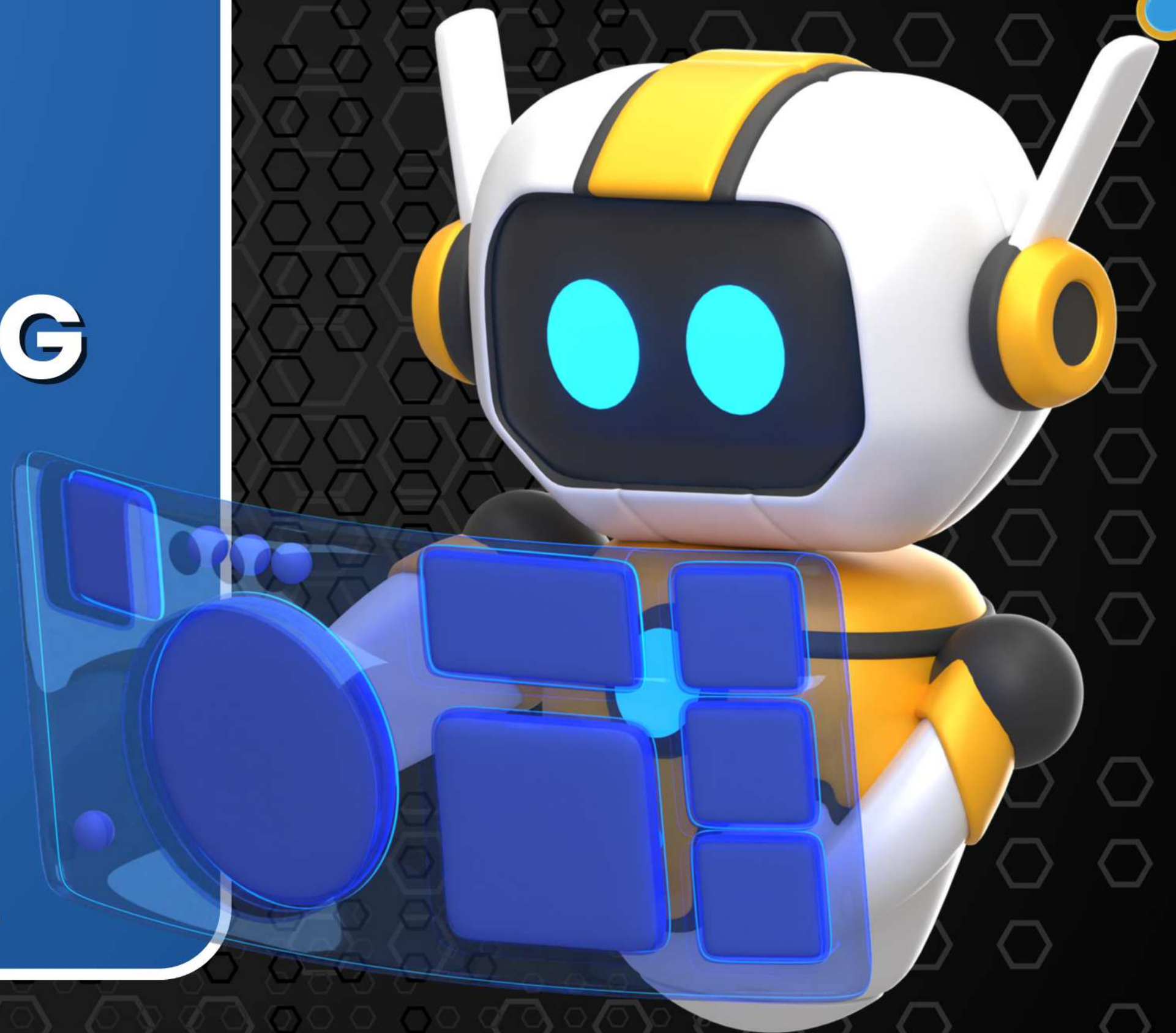


Universitas Pamulang
Sistem Informasi

MOBILE PROGRAMMING

PERTEMUAN 8

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





TOPIK PEMBAHASAN

- Pengenalan Spinner 01
- Karakteristik Spinner 02
- Cara Kerja RadioButton 03
- Flutter 04
- Kesimpulan 05





Apa itu Spinner?

Spinner, dalam konteks antarmuka pengguna (UI), terutama pada aplikasi seluler dan web, adalah indikator visual yang memberi tahu pengguna bahwa aplikasi sedang melakukan suatu proses di latar belakang dan mereka perlu menunggu proses tersebut selesai. Ini adalah cara untuk memberikan umpan balik kepada pengguna agar mereka tidak mengira aplikasi macet atau tidak responsif.





Karakteristik Utama Spinner

- Indikator Visual Aktivitas: Fungsi utamanya adalah untuk memberikan umpan balik visual kepada pengguna bahwa aplikasi sedang melakukan suatu proses di latar belakang. Animasi spinner menunjukkan bahwa aplikasi tidak diam atau macet.
- Animasi Berulang: Spinner umumnya berupa animasi yang bergerak secara terus-menerus selama proses berlangsung. Pergerakan ini (berputar, bergerak maju mundur, atau bertambah panjang) adalah ciri khasnya.
- Indikasi Waktu Tunggu: Kehadiran spinner secara implisit memberi tahu pengguna bahwa mereka perlu menunggu proses tersebut selesai. Ini membantu mengelola ekspektasi pengguna.



Karakteristik Utama Spinner

- **Non-Interaktif:** Selama spinner ditampilkan, pengguna biasanya tidak dapat berinteraksi dengan elemen UI lainnya di area yang tertutup oleh spinner (terutama jika menggunakan overlay). Ini mencegah pengguna melakukan tindakan yang tidak diinginkan saat proses sedang berjalan.
- **Dapat Dikustomisasi:** Tampilan spinner (ukuran, warna, ketebalan garis, gaya animasi) seringkali dapat disesuaikan oleh pengembang agar sesuai dengan desain keseluruhan aplikasi dan merek.



Karakteristik Spinner Berdasarkan Jenisnya

Circular Spinner (Indikator Progres Melingkar):

- Bentuk: Biasanya berupa lingkaran atau sebagian lingkaran yang berputar.
- Indikasi Progres: Umumnya indeterministik, artinya tidak memberikan indikasi seberapa jauh atau berapa lama lagi proses akan berlangsung.
- Penggunaan Umum: Cocok untuk operasi yang durasinya tidak dapat diprediksi atau ketika persentase kemajuan tidak relevan untuk ditampilkan.

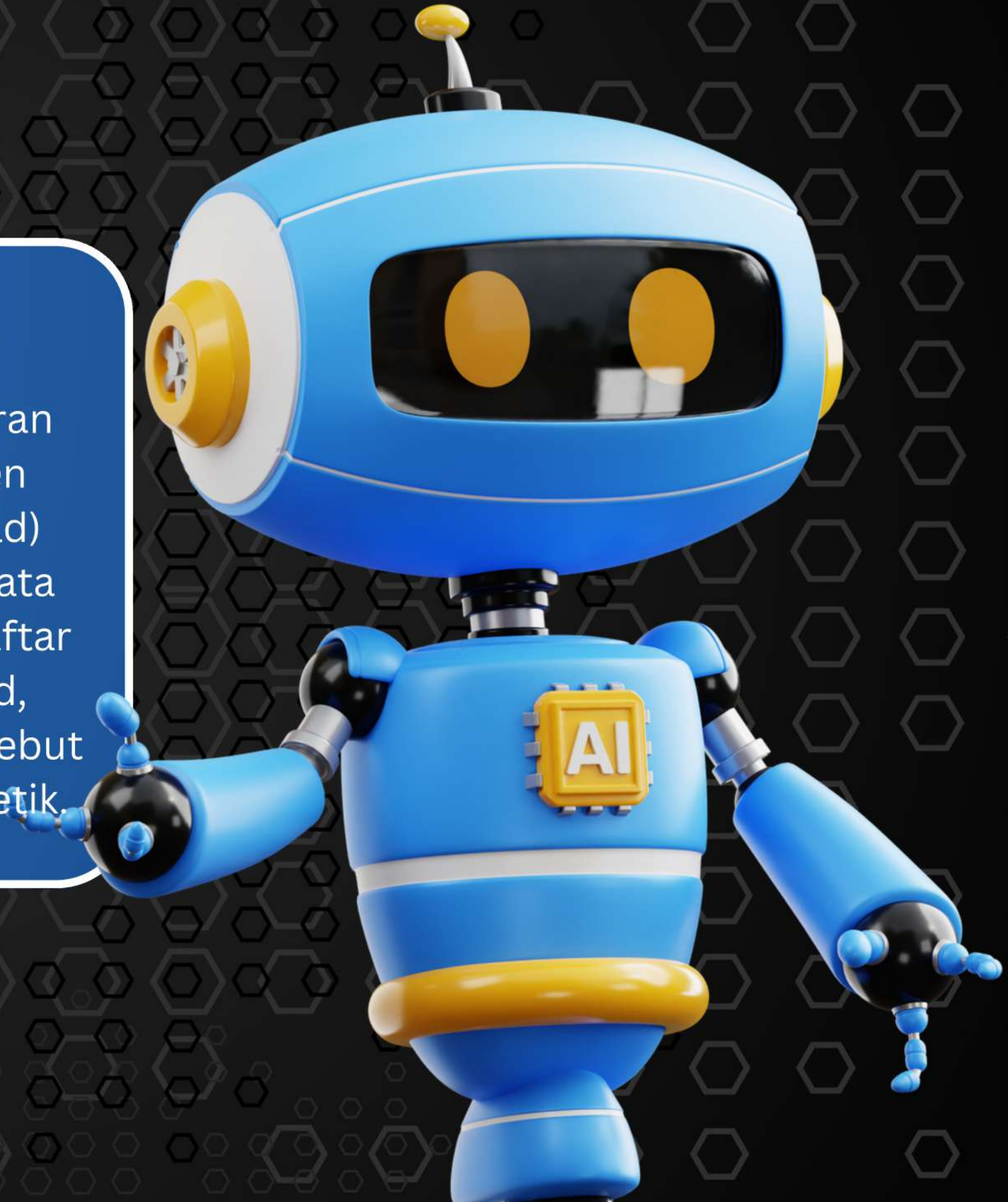
Linear Spinner (Indikator Progres Linear):

- Bentuk: Berupa garis horizontal yang bergerak dari satu sisi ke sisi lain atau bertambah panjang.
- Indikasi Progres: Bisa deterministik (menampilkan persentase kemajuan jika properti value diisi) atau indeterministik (bergerak tanpa henti dari satu ujung ke ujung lain).
- Penggunaan Umum: Cocok untuk operasi di mana kemajuan dapat diukur atau divisualisasikan secara linear, seperti mengunduh file atau mengunggah data.



Apa itu AutoComplete Textfield

Autocomplete TextField, atau sering disebut juga dengan saran otomatis atau pelengkapan otomatis, adalah sebuah elemen antarmuka pengguna (UI) dalam bentuk kolom teks (TextField) yang memiliki kemampuan untuk memberikan daftar saran kata atau frasa secara otomatis saat pengguna mulai mengetik. Daftar saran ini biasanya muncul di bawah atau di sekitar TextField, memungkinkan pengguna untuk memilih salah satu saran tersebut untuk melengkapi atau mengganti apa yang sedang mereka ketik.





Karakteristik AutoComplete TextField

- Mendengarkan Input Pengguna: TextField ini secara aktif memantau karakter yang diketik oleh pengguna.
- Pencarian Berdasarkan Input: Berdasarkan input yang telah diketik, sistem akan melakukan pencarian dalam sekumpulan data (database, daftar kata, API, dll.) untuk menemukan kata atau frasa yang sesuai atau relevan.
- Menampilkan Daftar Saran: Hasil pencarian yang relevan akan ditampilkan dalam bentuk daftar di bawah atau di sekitar TextField.
- Interaksi Pengguna: Pengguna dapat melihat daftar saran ini dan memilih salah satunya dengan mengetuk atau mengkliknya.
- Pengisian Otomatis: Ketika pengguna memilih sebuah saran, teks di dalam TextField akan otomatis terisi dengan saran yang dipilih.
- Penyaringan Dinamis: Daftar saran akan terus diperbarui secara dinamis seiring dengan perubahan input pengguna. Semakin banyak karakter yang diketik, daftar saran biasanya akan menjadi lebih spesifik dan lebih sedikit.



Manfaat AutoComplete TextField

- Meningkatkan Kecepatan Input: Pengguna dapat menemukan dan memilih kata atau frasa yang diinginkan dengan lebih cepat daripada mengetik seluruhnya.
- Mengurangi Kesalahan Ketik: Dengan memilih dari saran yang diberikan, risiko kesalahan ejaan atau penulisan dapat diminimalkan.
- Meningkatkan Pengalaman Pengguna: Fitur ini membuat interaksi dengan aplikasi atau situs web menjadi lebih efisien dan intuitif.
- Membantu Pengguna Menemukan Informasi: Dalam konteks pencarian, autocomplete dapat membantu pengguna menemukan istilah atau produk yang mungkin tidak mereka ketahui ejaannya secara pasti atau mungkin tidak terpikirkan sebelumnya.
- Panduan Input: Autocomplete dapat memandu pengguna dalam memasukkan data dengan format yang benar atau memilih dari opsi yang valid.



Installation

Ketikan pada Terminal :

```
$ flutter pub add async_button_builder
```

```
flutter pub outdated
```

Pubspec.yaml

```
dependencies:  
  async_button_builder: ^3.0.0+1
```

Pada main.dart tambahkan :

```
import 'package:async_button_builder/async_button_builder.dart';
```




SOURCE CODE

main.dart

```
import 'package:flutter/material.dart';  
import 'package:async_button_builder/async_button_builder.dart';  
  
void main() => runApp(const RegistrationApp());  
  
class RegistrationApp extends StatelessWidget {  
  const RegistrationApp({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      home: Scaffold(  
        appBar: AppBar(  
          backgroundColor: Colors.blue, // Warna biru untuk AppBar  
          title: const Text('Form Registrasi'),  
        ),  
        body: const RegistrationForm(),  
      ),  
    );  
  }  
}
```





SOURCE CODE

```
class RegistrationForm extends StatefulWidget {  
  const RegistrationForm({super.key});  
  
  @override  
  State<RegistrationForm> createState() => _RegistrationFormState();  
}  
  
class _RegistrationFormState extends State<RegistrationForm> {  
  final TextEditingController _usernameController =  
    TextEditingController();  
  final TextEditingController _passwordController =  
    TextEditingController();  
  static const List<String> _hobbyOptions = <String>[  
    'Futsal',  
    'Badminton',  
    'Voli',  
    'Berenang',  
    'Mancing'  
  ];  
}
```




SOURCE CODE

```
String _selectedHobby = '';

@override
Widget build(BuildContext context) {
  return SingleChildScrollView( // Agar bisa discroll jika layar kecil
    child: Padding(
      padding: const EdgeInsets.all(20),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start, // Posisi form di atas
        children: [
          const Text(
            'Silakan isi data berikut:',
            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
          ),
          const SizedBox(height: 10),
          TextField(
            controller: _usernameController,
            decoration: const InputDecoration(labelText: 'Username'),
          ),
          const SizedBox(height: 10),
          TextField(
            controller: _passwordController,
            decoration: const InputDecoration(labelText: 'Password'),
            obscureText: true,
          ),
          const SizedBox(height: 10),
          Autocomplete<String>(
            optionsBuilder: (TextEditingValue textEditingValue) {
              if (textEditingValue.text.isEmpty) {
                return const Iterable<String>.empty();
              }
              return _hobbyOptions.where((String option) {
                return option
                  .toLowerCase()
                  .contains(textEditingValue.text.toLowerCase());
              });
            },
          ),
        ],
      ),
    ),
  );
}
```

```
onSelected: (String selection) {
    setState(() {
        _selectedHobby = selection;
    });
},
fieldViewBuilder: (context, controller, focusNode, onFieldSubmitted) {
    return TextField(
        controller: controller,
        focusNode: focusNode,
        decoration: const InputDecoration(labelText: 'Hobi'),
    );
},
),
const SizedBox(height: 30),
AsyncButtonBuilder(
    child: const Text('Register'),
    loadingWidget: const CircularProgressIndicator(),
    onPressed: () async {
        await Future.delayed(const Duration(seconds: 2));
        // Proses registrasi (dummy)
    },
    builder: (context, child, callback, buttonState) {
        final buttonColor = buttonState.when(
            idle: () => Colors.blue,
            loading: () => Colors.grey,
            success: () => Colors.green,
            error: (err, stack) => Colors.red,
        );
        return SizedBox(
            width: double.infinity,
            child: ElevatedButton(
                onPressed: callback,
                style: ElevatedButton.styleFrom(
                    backgroundColor: buttonColor,
                    padding: const EdgeInsets.symmetric(vertical: 14),
                ),
                child: child,
            ),
        );
    },
),
),
),
),
),
);
}
}
```




HASIL

localhost:59123

localhost:59123

Form Registrasi

Silakan isi data berikut:

Username

Password

Hobi

Register





TUGAS

Buatlah Form Registrasi dengan kreasi masing-masing dan mengkombinasikan penggunaan Toast, alertDialog, Spinner, AutoComplete TextField dan dengan menggunakan Salomon

Tidak boleh ada yang sama

Batas pengumpulan 18 Mei 2025





Universitas Pamulang
Sistem Informasi

Thank You!

