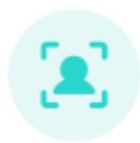


ILT 6 - Fundamental Flutter

9 Khrisna Indrawan E. P

Head of Reviewer
Dicoding Indonesia

Aturan **Instructor/Expert Led Session**



Fokus Penuh



Always-on Camera



Raise hand atau Chat jika ingin bertanya



Mute jika sedang tidak berbicara

Porsi Skor Penilaian (Tech)

Jenis Aktivitas	Porsi Skor
Rata-rata Kuis pada ILT	20%
Aktivitas selama ILT (bertanya, menjawab, membantu diskusi)	10%
Aktivitas Forum Diskusi*	10%
Rata-rata Exam/Submission Project	60%

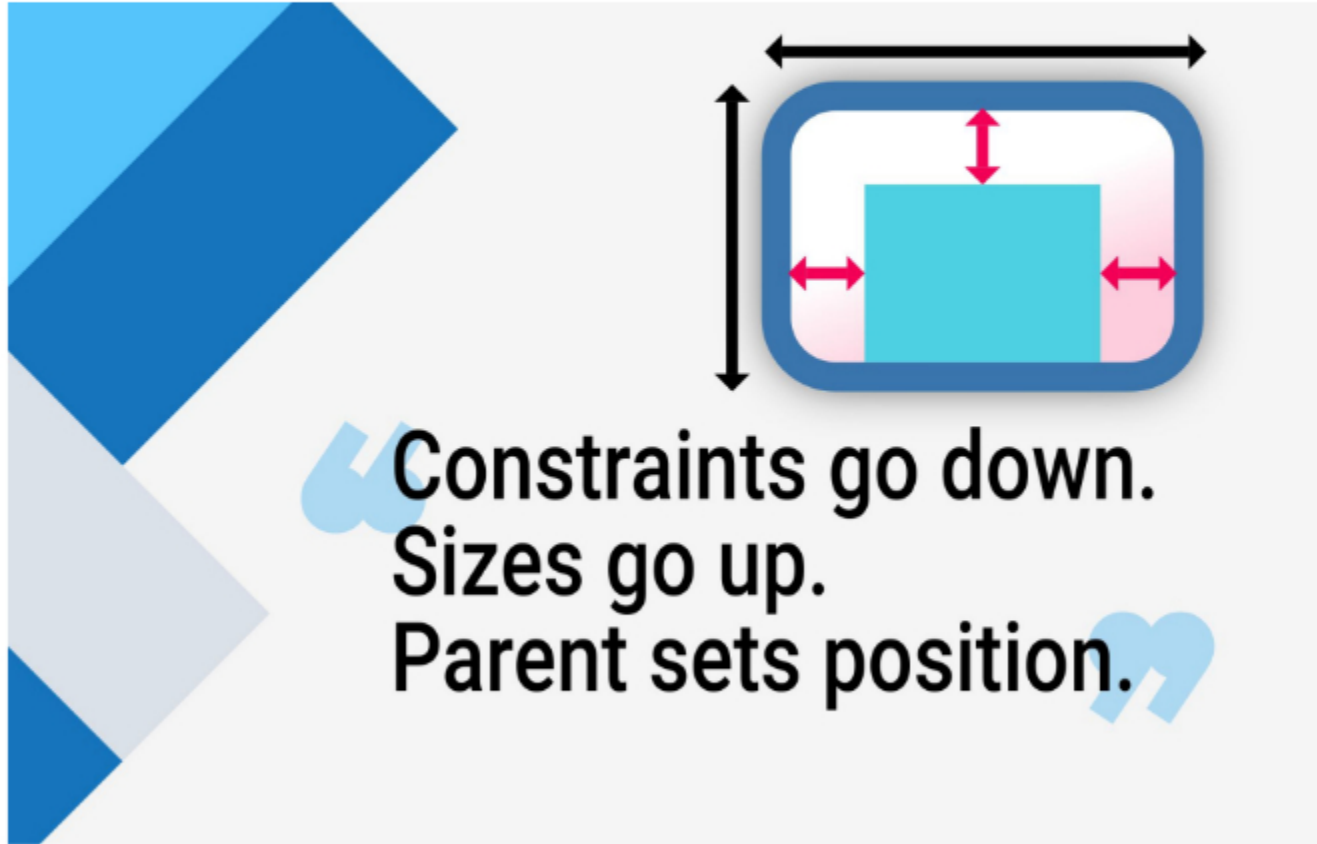
**Jika ditemukan peserta spamming pada forum diskusi, maka nilai terbaik yang bisa didapat adalah 60 (dari skala 100)*

Constraints

Masalah saat menyusun layout:

- Posisi atau ukuran widget tidak sesuai?
- Overflow?
- Unbounded height?

Layout Constraints

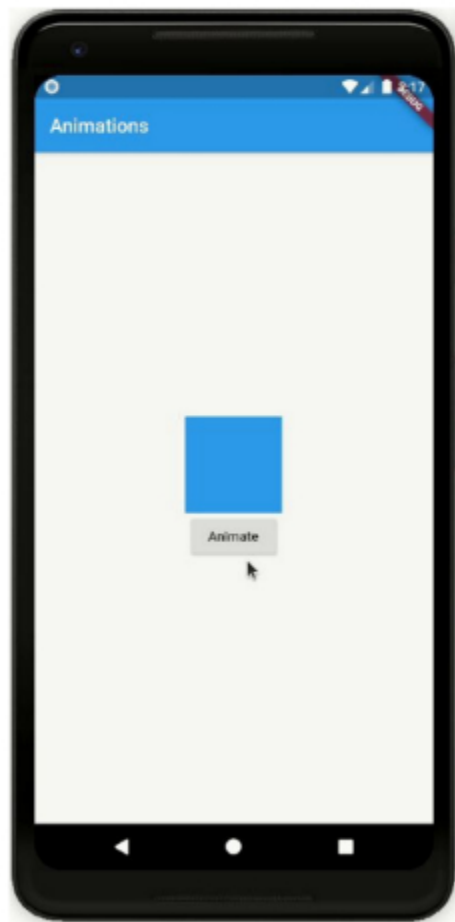


Layout Constraints



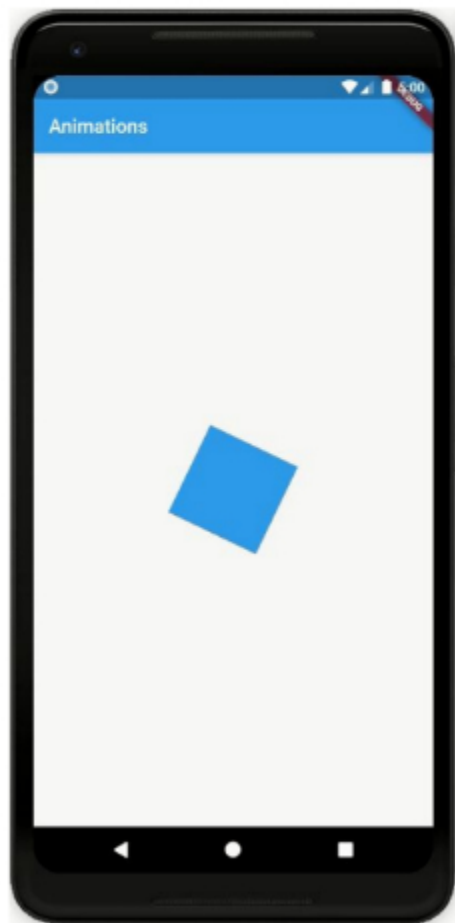
Animations

Built-in implicit animation



```
class _AnimationPageState extends State<AnimationPage> {  
  bool _isBig = false;  
  double _size = 100.0;  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Animations'),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            AnimatedContainer(  
              duration: Duration(seconds: 1),  
              color: Colors.blue,  
              height: _size,  
              width: _size,  
            ),  
            ElevatedButton(  
              child: Text('Animate'),  
              onPressed: () {  
                setState(() {  
                  _size = _isBig ? 200 : 100;  
                  _isBig = !_isBig;  
                });  
              },  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

Tween animation builder



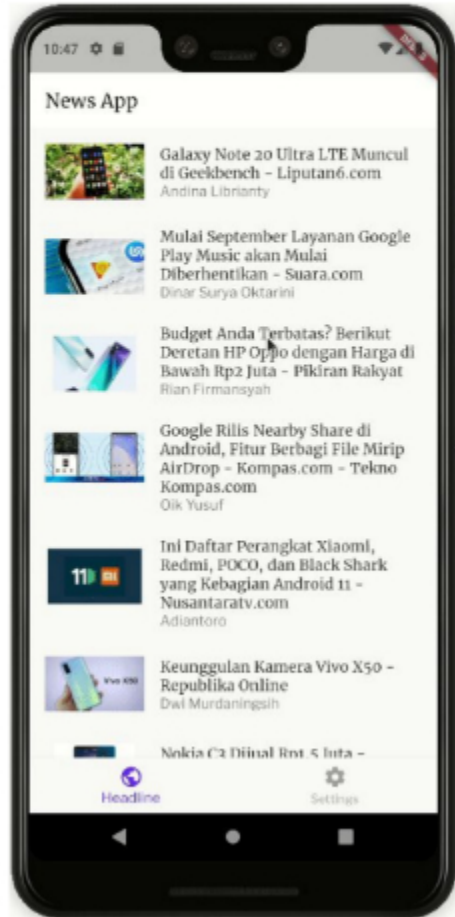
```
class _AnimationPageState extends State<AnimationPage> {  
  double _size = 100.0;  
  
  Tween<double> _animationTween = Tween(begin: 0, end: pi * 2);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Animations'),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            TweenAnimationBuilder<double>(  
              tween: _animationTween,  
              duration: Duration(seconds: 3),  
              builder: (context, double value, child) {  
                return Transform.rotate(  
                  angle: value,  
                  child: Container(  
                    color: Colors.blue,  
                    height: _size,  
                    width: _size,  
                  ),  
                );  
              },  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

Explicit Animation



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Animations'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          RotationTransition(
            turns: _animationController,
            child: Container(
              padding: const EdgeInsets.all(16.0),
              child: FlutterLogo(
                size: 100,
              ),
            ),
          ),
        ],
      ),
    ),
  );
}
```

Hero Animation



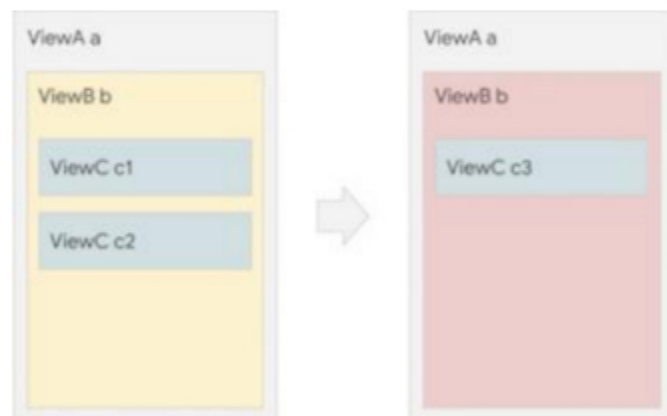
```
ListTile(  
  leading: Hero(  
    tag: article.urlToImage,  
    child: Image.network(  
      article.urlToImage,  
      width: 100,  
    ),  
  ),  
  ...  
);
```

State Management

Imperative vs Declarative UI

- Pada UI imperatif, kita bisa langsung mengubah tampilan UI berdasarkan id atau identifier lain.

```
// Imperative style  
b.setColor (red)  
b.clearChildren ()  
ViewC c3 = new ViewC (...)  
b.add (c3)
```



Imperative vs Declarative UI

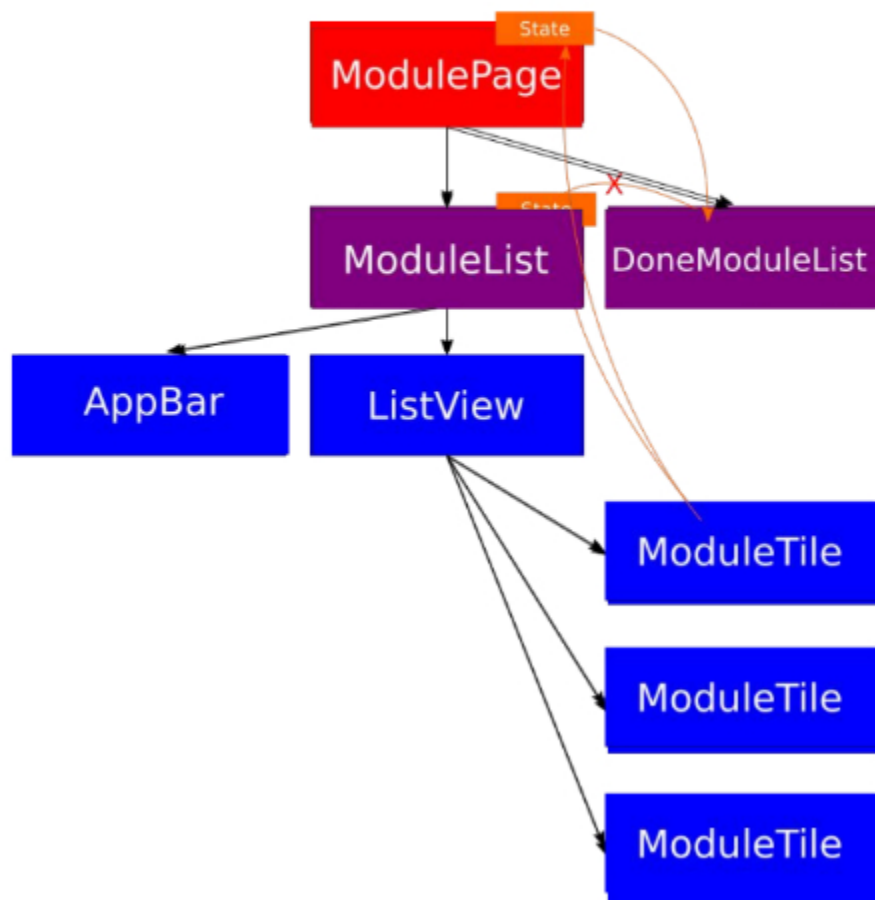
- Sementara pada UI deklaratif, UI akan dibangun ulang berdasarkan state saat itu.

```
// Declarative style
return ViewB(
  color: red,
  child: ViewC(...),
)
```

$$\text{UI} = f(\text{state})$$

The layout on the screen Your build methods The application state

State Management



State Management Packages

provider

3118 110 100%
LIKES PUB POINTS POPULARITY

A wrapper around InheritedWidget to make them easier to use and more reusable.

v 4.3.2+3 / 5.0.0-nullsafety.2 • Updated: Dec 7, 2020 [dash-overflow.net](#) [Flutter Favorite](#)

flutter_bloc

1613 110 99%
LIKES PUB POINTS POPULARITY

Flutter Widgets that make it easy to implement the BLoC (Business Logic Component) design pattern. Built to be used with the bloc state management package.

v 6.1.1 / 7.0.0-nullsafety.0 • Updated: Nov 11, 2020 [bloclibrary.dev](#) [Flutter Favorite](#)

flutter_redux

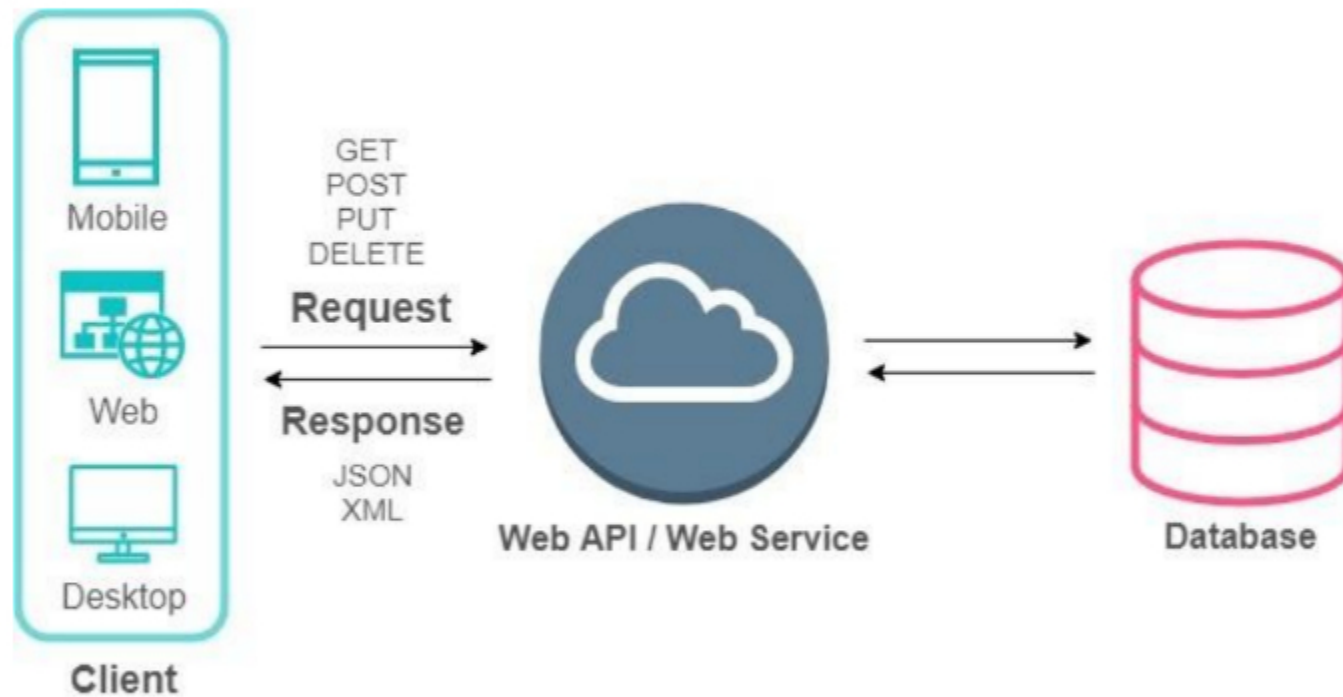
197 110 97%
LIKES PUB POINTS POPULARITY

A set of utility Widgets that Provide and Connect to a Redux Store

v 0.7.0 • Updated: Oct 18, 2020 [brianegan.com](#) [Flutter Favorite](#)

HTTP/Network Request

Web API



Network Request Packages

http

1795

LIKES

110

PUB POINTS

100%

POPULARITY

A composable, multi-platform, Future-based API for HTTP requests.

v 0.12.2 • Updated: Jul 16, 2020 • [dart.dev](https://pub.dev/packages/http)

dio

1433

LIKES

100

PUB POINTS

99%

POPULARITY

A powerful Http client for Dart, which supports Interceptors, FormData, Request Cancellation, File Downloading, Timeout etc.

v 3.0.10 • Updated: Aug 7, 2020 • [flutterchina.club](https://pub.dev/packages/dio)

Local Storage

Macam Storage

- Shared preferences
 - Key-value
 - Menyimpan tipe data dasar (num, string, boolean)
- Database lokal
 - SQLite
 - Data disimpan dalam tabel
- File
 - Membuat berkas yang disimpan pada internal atau external storage

Local Storage Packages

sqflite

1216 100 100%
LIKES PUB POINTS POPULARITY

Flutter plugin for SQLite, a self-contained, high-reliability, embedded, SQL database engine.

v 1.3.2+1 / 2.0.0-nullsafety.2 • Updated: Oct 28, 2020 • [tekartik.com](https://pub.dev/packages/sqflite) [Flutter Favorite](#)

shared_preferences

2314 100 100%
LIKES PUB POINTS POPULARITY

Flutter plugin for reading and writing simple key-value pairs. Wraps NSUserDefaults on iOS and SharedPreferences on Android.

v 0.5.12+4 • Updated: Nov 3, 2020 • [flutter.dev](https://pub.dev/packages/shared_preferences) [Flutter Favorite](#)

moor_flutter

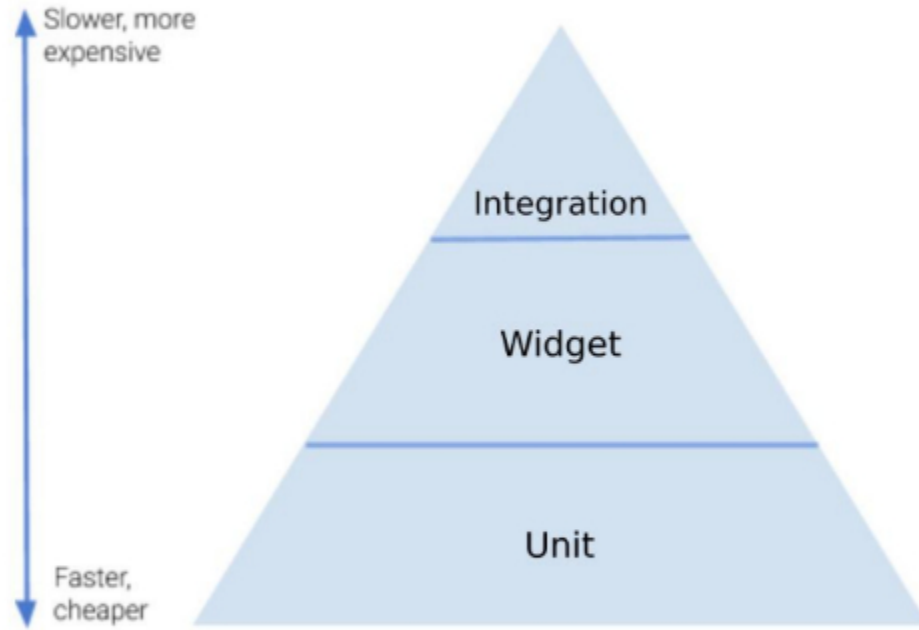
184 100 95%
LIKES PUB POINTS POPULARITY

Flutter implementation of moor, a safe and reactive persistence library for Dart applications

v 3.1.0 • Updated: Jul 6, 2020 • [simonbinder.eu](https://pub.dev/packages/moor_flutter)

Testing

Automated Testing



Automated Testing

	Unit	Widget	Integration
Keyakinan (Confidence)	Rendah	Lebih tinggi	Paling tinggi
Biaya pemeliharaan (Maintenance cost)	Rendah	Lebih tinggi	Paling tinggi
Ketergantungan (Dependencies)	Sedikit	Banyak	Paling banyak
Kecepatan eksekusi (Execution speed)	Cepat	Cepat	Lambat

dicoding

Kampus
Merdeka
INDONESIA JAYA



[LINK]

Pertanyaan sesi Talks

- Sudah berapa lama menggunakan Flutter?
- Apa impresi pertama saat menggunakan Flutter?
- Apakah sudah pernah mengembangkan aplikasi yang production ready menggunakan Flutter?
- Sebelumnya sudah pernah ngoding native? Jika iya apa perbedaan signifikan dengan mengembangkan aplikasi menggunakan Flutter?
- Kira-kira ada gambaran gak prospek Flutter kedepannya bakal seperti apa?

**akan diskip saat presentasi*