

Berikut ini adalah salah satu penerapan unit testing yang dilakukan oleh LearnFazz untuk Bloc fitur See All Comments (Comment List).

Pertama-tama, perlu dilakukan import package flutter_test, mockito, Comment List Bloc, dan package-package lain yang bersangkutan.

```
import 'package:bloc/bloc.dart';
import 'package:flutter_test/flutter_test.dart';
import 'package:learn_fazz/blocs/comment_list/comment_list.dart';
import 'package:learn_fazz/models/comment_list_result.dart';
import 'package:learn_fazz/repositories/comment_repository.dart';
import 'package:mockito/mockito.dart';

import '../test_helpers/mock_classes.dart';
import '../test_helpers/simple_bloc_delegate.dart';
import '../test_helpers/test_factories/comment_factory.dart';
```

Pada setUp, kami membuat mock instansi dari CommentRepository menggunakan mockito. Kemudian kami juga menginisiasi CommentListBloc dengan mockCommentRepository sebagai dependensinya.

```
void main() {
  CommentListBloc bloc;
  CommentRepository mockCommentRepository;

  final CommentListState initialState =
    GetCommentListSuccess(CommentListResult.empty());

  setUp(() {
    mockCommentRepository = MockCommentRepository();
    bloc = CommentListBloc(commentRepository: mockCommentRepository);
  });
```

Test pertama yang kami lakukan adalah sanity test untuk memastikan bahwa ketika dispose dilakukan maka state dari Bloc tidak diupdate.

```
test('dispose should not emit new states', () {
  expectLater(bloc.state, emitsInOrder([]));
  bloc.dispose();
});
```

Untuk melakukan test pada CommentList success, perlu dibuat skenario yaitu untuk pemanggilan fungsi getCommentList akan dihasilkan objects list dari CommentList. Kemudian kami membuat sebuah setup mengenai ekspektasi event apa saja yang dihasilkan oleh Bloc untuk mengetahui apakah comment list sukses dijalankan. Ekspektasi tersebut dideklarasikan pada expectLater dimana salah satu parameternya emitsInOrder berisi pergantian state dari initialState, GetCommentListLoading(), hingga CommentListSuccess(). Pada line terakhir dilakukan bloc dispatch untuk test fungsi yang sudah dibuat.

```
test('CommentList success', () {
  when(mockCommentRepository.getCommentList(any, any)).thenAnswer(
    (_) => Future<CommentListResult>.value(
      CommentFactory.constructDefaultCommentListResult(),
    ),
  );
  final _initialState = initialState;
```

```

    expectLater(
      bloc.state,
      emitsInOrder([
        _initialState,
        GetCommentListLoading(),
        GetCommentListSuccess(CommentFactory.constructDefaultCommentListResult())
      ])).then((_) => bloc.dispose());
    bloc.dispatch(GetCommentList(courseId: 1, offset: 0, postId: 1));
  });

```

Test untuk CommentList fail memiliki kode yang hampir sama dengan CommentList success. Perbedaannya terletak pada skenario awal yang melakukan throw 404 error untuk membuat getCommentList fail. Perbedaan lainnya adalah state emitsInOrder terakhir merupakan GetCommentListFailure().

```

test('CommentList should fail', () {
  when(mockCommentRepository.getCommentList(any, any)).thenThrow('404 Error');

```

```

    expectLater(
      bloc.state,
      emitsInOrder([
        initialState,
        GetCommentListLoading(),
        GetCommentListFailure('404 Error')
      ])).then((_) => bloc.dispose());

    bloc.dispatch(GetCommentList(courseId: 1, offset: 0, postId: 1));
  });
}

```

muslim abdul jabbar