JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438

STRUKTUR FLUTTER BLOC DALAM APLIKASI PEMESANAN TIKET KAPAL BERBASIS MOBILE

Rikie Kartadie*[1)], Riki Hikmanto[2)], Yudhi Kusnanto[3)], Pius Dian Widi Anggoro[4)]
1. Unversitas Teknologi Digital Indonesia, Indonesia
2. Unversitas Teknologi Digital Indonesia, Indonesia
3. Unversitas Teknologi Digital Indonesia, Indonesia
4. Unversitas Teknologi Digital Indonesia, Indonesia

Article Info
Kata Kunci: *BLoC; Flutter; Mobile;* Pemesanan Tiket

Keywords: *BLoC; Flutter; Mobile; Ticket Booking*

Article history:
Received 30 May 2023
Revised 13 June 2023
Accepted 27 June 2023
Available online 1 December 2023

DOI :
https://doi.org/10.29100/jipi.v8i4.4712

* Corresponding author.
Rikie Kartadie
E-mail address:
rikie@utdi.ac.id

ABSTRAK
Framework Flutter yang menggunakan bahasa pemrograman Dart dalam pengkodeannya. SDK atau framework open source Flutter dibuat oleh Google untuk membantu pengembang membuat atau mengembangkan aplikasi yang berjalan pada sistem operasi Android dan iOS. Karena state-management system pada intinya adalah sebuah teori, ada banyak state-management system yang dikembangkan oleh para penghobi dan profesional sebelum pembuatan Flutter. Dengan memanfaatkan struktur BLoc Flutter penelitian ini dilaksanakan untuk membuat aplikasi yang akan digunakan pada smartphone dengan sistem operasi Android dan iOS sebagai media pemesanan bagi PT. DLN dan Firebase Database sebagai Databasenya. Hasil uji aplikasi berjalan baik, setiap BLoC telah berhasil. Dan aplikasi tidak mengalami kendala yang besar. Uji BLoC Cubit dengan transaksi sukses dan gagal dapat berjalan dengan baik.

ABSTRACT
The Flutter framework uses the Dart programming language in its coding. The Flutter open source SDK or framework was created by Google to help developers create or develop applications that run on the Android and iOS operating systems. Since state-management systems are essentially a theory, there were many state-management systems developed by hobbyists and professionals before the creation of Flutter. By utilizing the BLoc Flutter structure, this research was carried out to create an application that will be used on smartphones with the Android and iOS operating systems as an ordering medium for PT. DLN and Firebase Database as the database. The application test results went well, every BLoC was successful. And the application didn't experience any major problems. Cubit's BLoC test with successful and failed transactions can run well.

I. PENDAHULUAN

PERDAGANGAN sangat bergantung pada pelabuhan. Jika pelabuhan dikelola dengan baik dan efektif, perdagangan akan berkembang, dan industri lokal akan berkembang sendiri. Pelabuhan sangat penting karena, menurut sejarah kota-kota metropolitan di Negara kepulauan seperti Indonesia, pelabuhan memengaruhi pertumbuhan kota. Pelabuhan berfungsi sebagai penghubung antara pembangunan jalan raya, jaringan rel kereta api, dan tempat distribusi. Pelabuhan memiliki peran yang tidak kalah penting sebagai pusat perdagangan dan ekonomi serta tempat berbagai bisnis seperti pelayaran dan keagenan, pergudangan, *freight forwarding*, dan sebagainya [1].

Salah satu kapal bernama Kmp Dln Oasis berangkat dari Lombok ke Surabaya dan kembali ke Banyuwangi. Ini adalah kapal baru yang mengambil jurusan dari Lombok, yang sebelumnya dilakukan oleh Kmp Legundi merupakan kapal yang sanagt penting. Namun, pelayanannya masih bersifat manual, seperti pemesanan tiket dan informasi tentang jadwal keberangatan. Pelanggan harus datang ke kantor PT.DLN, pada loket pelabuhan yang sudah disediakan, atau melalui telepon untuk mendapatkan informasi tentang pemesanan dan harga tiket. Hal ini menyebabkan antrian pembelian tiket di loket dan kurang efisien dalam melayani pelanggan.

Pulau Lombok berada di Nusa Tenggara Barat. Selat Lombok memisahkan pulau dari Bali di sebelah barat dan selat Alas dari Sumbawa di sebelah timur. Karena banyaknya tempat wisata, Pulau Lombok dan Bali menarik banyak wisatawan dari berbagai negara. Salah satu cara untuk pergi ke pulai ini adalah melalui laut atau udara. Namun, karena banyaknya barang yang harus dibawa dan biayanya yang terjangkau, transportasi laut lebih populer untuk perjalanan antara pulau Bali dan Lombok. Namun, untuk menggunakan layanan transportasi ini harus mengantri untuk membeli tiket karena pembelian hanya dapat dilakukan di loket pelabuhan [2].

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438

Dengan perkembangan teknologi di bidang perangkat mobile yang diterapkan untuk sarana transportasi, penggunaan perangkat mobile untuk transportasi bus menjadi sangat jarang. Ini karena perangkat mobile sudah mendukung fitur internet. Hal ini dapat dimanfaatkan oleh perusahaan jasa transportasi darat untuk mengatasi masalah penyajian informasi, terutama terkait ketersediaan tiket dan proses pemesanan yang masih dilakukan secara manual. Aplikasi pemesanan tiket yang dapat diinstal pada perangkat mobile juga dapat membantu perusahaan transportasi mengatasi masalah yang terjadi.

Penelitian yang dilakukan oleh Himawan Udin Hatari dkk, Aplikasi pemesanan tiket kapal laut online berbasis android. Aplikasi ini mempermudah pelayanan masyarakat dalam pembelian tiket, mengetahui informasi jadwal keberangkatan, mengetahui harga tiket, sehingga dengan adanya aplikasi ini proses pemesanan dapat dilakukan dengan cepat, mudah dan efisien [3], penelitian ini dirancang untuk membantu calon penumpang dalam hal mempermudah pemesanan tiket secara praktis. Penegmbangan aplikasi ini menggunakan Android Studio sebagai pengembangan aplikasi dan *website* sebagai penyedia data untuk mempermudah pengelolaan tiket dan administrasi lainnya.

Penelitian yang dilakukan oleh Khairatin Pemilihan kapal laut menggunakan algoritma apriori dalam aplikasi reservasi tiket berbasis web dengan mengimplementasikan algoritma apriori untuk pemilihan kapal laut dan mengetahui aturan asosiasi yang terbentuk berdasarkan nilai *minimum support* dan nilai *minimum confidence* yang ditentukan[4].

Framework Flutter yang menggunakan bahasa pemrograman *Dart* dalam pengkodeannya. *SDK* atau *framework open source Flutter* dibuat oleh *Google* untuk membantu pengembang membuat atau mengembangkan aplikasi yang berjalan pada sistem operasi Android dan iOS. Dalam pembuatan aplikasi, *framework Flutter* berbeda dari yang lainnya karena semua kode dikompilasi dalam kode *native*-nya (*Android NDK, LLVM*, dan *AOT-compiled*) tanpa ada *interpreter* dalam prosesnya, sehingga proses *compiling* menjadi lebih cepat. [5].

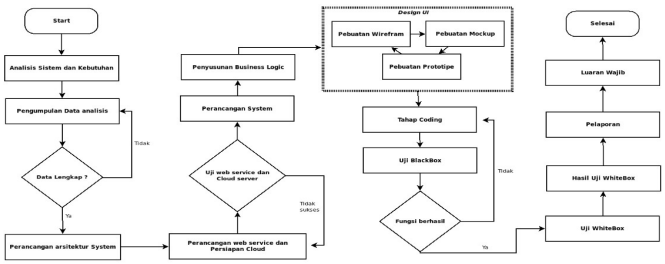
Dalam *Flutter*, setiap bagian dibuat dalam bentuk *widget*. Dua *widget*, *stateful* dan *stateless*, memiliki perlakukan *state* yang berbeda. *Stateless* merupakan *widget* yang dimuat secara statis dimana seluruh konfigurasi
yang dimuat di dalamnya telah diinisialisasikan sejak awal *widget* itu dimuat. Artinya, *state* jenis ini tidak dapat diubah dan tidak akan pernah berubah. *Stateless* biasa digunakan hanya untuk tampilan saja seperti *button, item, box container*, dan lain-lain. *Statefull* merupakan suatu *widget* yang sifatnya dinamis atau dapat berubah-ubah, kebalikan dari *stateless widget*. Pada *statefull*, dapat menggunakan *property Init State* yang berfungsi untuk menginisialisasi *state* yang akan pertama kali dijalankan. *Statefull widget* dapat mengubah atau mengupdate tampilan, menambah *widget* lainnya, mengubah nilai *variabel*, *icon*, warna, dan masih banyak lagi [6].

Karena *state-management system* pada intinya adalah sebuah teori, ada banyak *state-management system* yang dikembangkan oleh para penghobi dan profesional sebelum pembuatan Flutter. Namun, situs resmi Flutter dan beberapa saran pakar sepertinya menyarankan agar tim pengembangan Flutter bersama dengan pengembang komunitas telah mengidentifikasi tiga *state-management system* yang dapat dipilih oleh pengembang Flutter: BLoC, Scoped Model, dan Redux [7].

Tujuan yang ingin diperoleh dari penelitian ini adalah memanfaatkan struktur BLoc *Flutter* untuk membuat aplikasi yang akan digunakan pada *smartphone* dengan sistem operasi *Android* dan *iOS* sebagai media pemesanan bagi PT. DLN dan *Firebase Database* sebagai Databasenya.

II. METODE PENELITIAN

Struktur *Flutter BLoC* dalam aplikasi ini merupakan bagian yang tidak terpisahkan dari seluruh pengambangan aplikasi, yang nantinya akan memberikan beberapa Bloc untuk setiap presentasi yang ditampilkan dan menghubungkannya kepada service yang digunakan. Alur penelitian seperti terdeskripsi diatas dapat dilihat pada Gambar 1 berikut ini.



Gambar. 1. Arsitektur Sistem

Alur Penelitian dibagi menjadi 4 tahap besar, yang pertama adalah (1) Tahap pengumpulan data dan analisi sistem

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438

dan kebutuhan; (2) Tahap kedua adalah tahap persiapan *webservice* dan persiapan *server cloud*; (3) Tahap ketiga adalah tahap pembutan/*design UI* dan tahap *coding*; kemudian (4) Tahap empat adalah tahap pengujian dengan pengujian *blackbox* dan *whitebox*. Pada artikel ini, kami hanya akan berfokus pada pengujian *whitebox* yang berkaitan dengan *struktur Flutter BloC*.

A. Analisis Sistem dan kebutuhan

1) Kebutuhan input

Pada sistem informasi yang dibangun membutuhkan masukan atau input agar sistem berjalan sesuai dengan tujuan dibuatnya sistem. Berikut input yang dibutuhkan : a) Data mengenai pelabuhan; b) Data mengenai harga tiket; c) Data tiket kapal.

2) Kebutuhan proses

Kebutuhan proses sistem ini adalah : a) Proses data jadwal oleh admin; b) Manajemen data (memasukan, memperbarui, dan menghapus data jadwal); c) Proses pemilihan data tiket; d) Proses menampilkan data tiket.

3) Kebutuhan output

Kebutuhan output pada sistem ini adalah : a) Informasi jadwal kapal sesuai data yang dimasukan pada sistem; b) Informasi jadwal kapal yang akan ditampilkan berupa nama kapal, waktu keberangkata, tempat pelabuhan, dan deskripsi kapal.

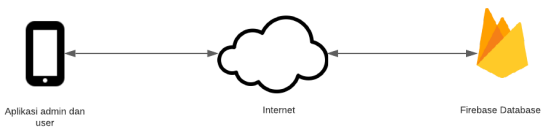
Dalam membangun sistem menggunakan perangkat lunak dibawah ini : a) MacOS digunakan sebagai Sistem Operasi karena sistem operasi iOS hanya di dukung oleh MacOS; b) Android Studio digunakan sebagai text editor untuk membangun aplikasi; c) Firebase digunakan sebagai database; d) Flutter sebagai bahasa pemrograman; e) SDK(Software Development Kit); f) Figma untuk membuat desain UI(User Interface) dari aplikasi.

Metode yang digunakan dalam proses pengumpulan data adalah wawancara yang digunakan untuk mencari bahan materi yang berhubungan dalam pembuatan aplikasi dengan narasumber yang ada di Pelabuhan Lembar.

B. Design Sistem

1) Arsitektur Sistem

Gambar 2 menjelaskan tentang arsitektur sistem yang dimana *user* dan *admin* menggunakan satu aplikasi yang sama, dengan aplikasi tersebut harus memiliki koneksi ke internet terlebih dahulu lalu akan di *database* kita menggunakan *Firebase Database Firestore* untuk mendapatkan data akan tetapi *user* atau pengguna diminta untuk *login* terlebih dahulu disini *Firebase* juga menyediakan layanan autentikasi yang bisa digunakan oleh *user* untuk *login*.

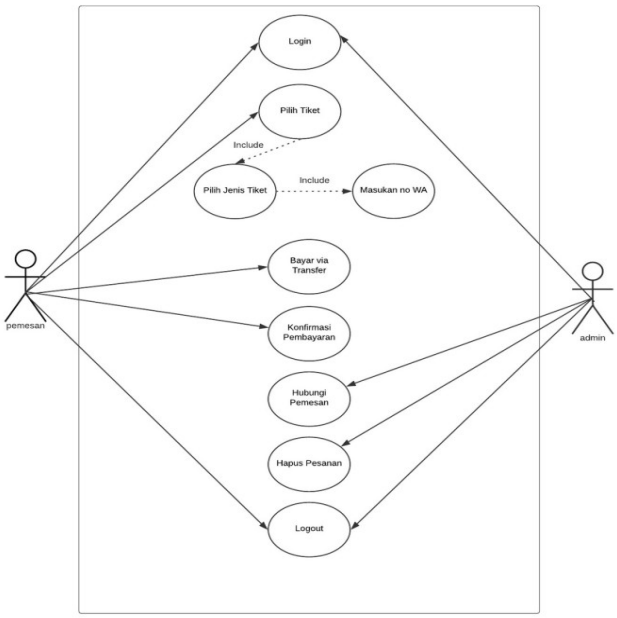


Gambar. 2. Arsitektur Sistem

2) Usecase Diagram

Use case diagram untuk pemesan dan admin. pemesan dan admin bisa masuk langsung ke aplikasi, dan setelah itu pemesan memilih tiket kapal kemudian melakukan pemesanan, setelah itu pemesan menerima detail pembayaran, kemudian melakukan pembayaran secara langsung via transfer setelah itu konfirmasi pensanan dan bukti pesanan kepada admin. Untuk admin berhak untuk menghubungipemesan dan menghapus pesanan yang diterima dapat dilihat pada Gambar 3.

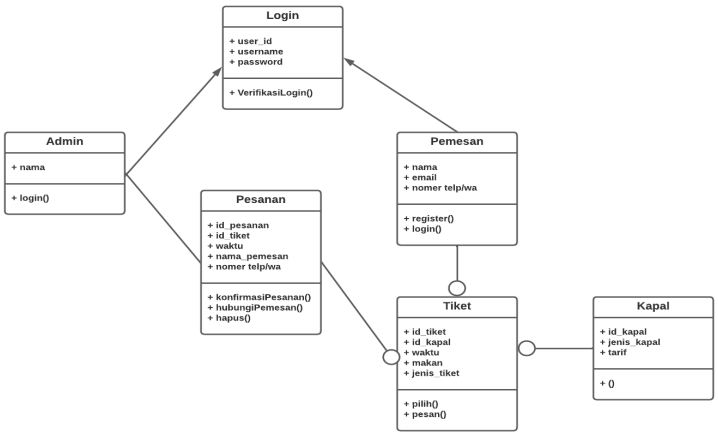
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438



Gambar. 3. *UseCase Diagram*

3) Class Diagram

Class diagram pada Gambar 4 menggunakan generalisasi serta asosiasi antara kelas serta atribut-atribut yang melekat pada kelas tersebut. Asosiasi antar kelas ini pun menjadi *database* yang diletakkan pada *firebase.*

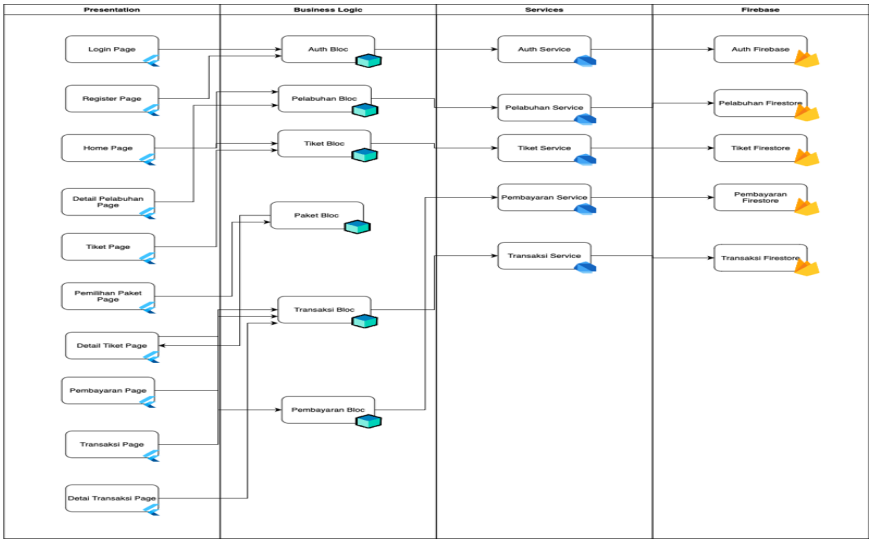


Gambar. 4. *Class Diagram*

4) Struktur Business Logic

Gambar 5 dibawah menjelaskan tentang struktur tampilan bisnis logika yang akan dibuat pada aplikasi dengan menerapkan Flutter BLoC sebagai perantara antara tampilan dengan service dan firebasenya.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438



Gambar. 5. *Struktur Business Logic*

C.Rancangan Antarmuka

Rancangan antarmuka merupakan gambaran umum desain tampilan yang akan digunakan atau dibuatpada sistem. Perancangan UI yang dilaksanakan adalah untuk memandu pengguna secara visual melalui antarmuka produk. Hal ini untuk menciptakan pengalaman intuitif yang tidak mengharuskan pengguna untuk berpikir terlalu banyak dalm menjalankan aplikasi. Salah satu contoh design antar muka adalah terlihat pada gambar 6.

Gambar. 6. *Halaman Paket Tiket*

D. Proses Pengujian

Proses Pengujian merupakan hal terpenting dalam sebuah perangkat lunak yang bertujuan untuk menemukan kesalahan ataupun kekurangan pada perangkat lunak yang akan diuji. Pada penelitian ini dilakukan pengujian perangkat lunak dengan menggunakan metode *white box testing*. Pengujian dengan metode *white box* dilakukan untuk meyakinkan semua perintah dan kondisi dieksekusi secara minimal. Untuk testing otomatis, digunakan unit testing yang tersedia pada *library Flutter*, yakni *unit test* dan *widget test*. Disini digunakan *unit test* untuk melakukan pengujian pada fungsi pada BLoC pemesanan.

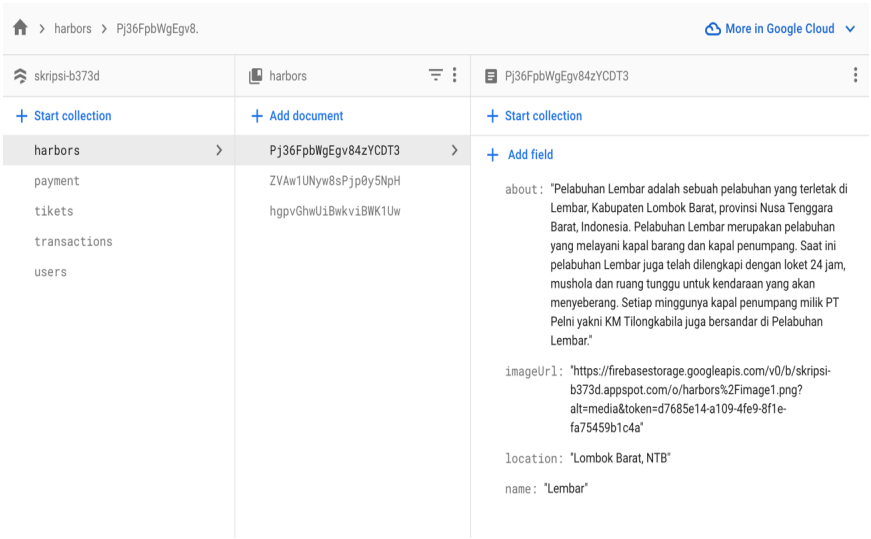
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438

III. HASIL DAN PEMBAHASAN

Berdasarkan analisis dan perancangan sistem yang telah dilakukan, maka penulisan kode program dapat dilakukan, sehingga dapat diharapkan program yang dihasilkan nantinya mampu memenuhi kebutuhan yang diharapkan secara optimal. Beberapa *Collection* penting dalam sistem kami sampaikan sebagai berikut:

A. Collection

2) Collection Kapal

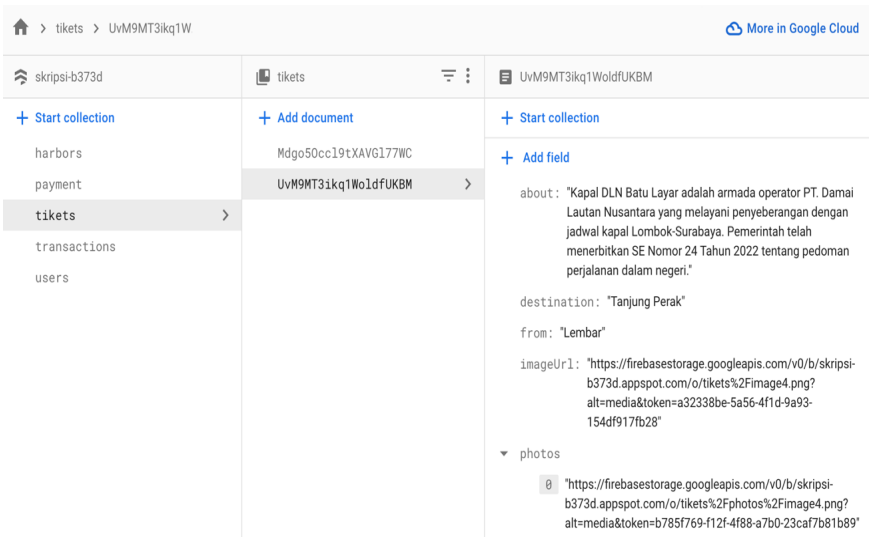


Gambar. 7. *Collection kapal*

Pada Gambar 7 diatas merupakan sebuah data pelabuhan yang didalamnya terdapat lokasi, deskripsi, nama daerah dan gambar pelabuhan. *Collection* kapal ini datanya berada pada firebase sehingga dapat dilihat bahwa dokumen kapal terinkripsi. *Collection Kapal* ini berada pada *Pelabuhan BLoC* yang terkoneksi pula dengan *pelabuhan service,* seperti tergambar pada Gambar 5.

3) Collection tiket

Pada Gambar 8 merupakan sebuah data tiket yang didalamnya terdapat tujuan, keberangkatan, foto, deskripsi dan harga tiket.

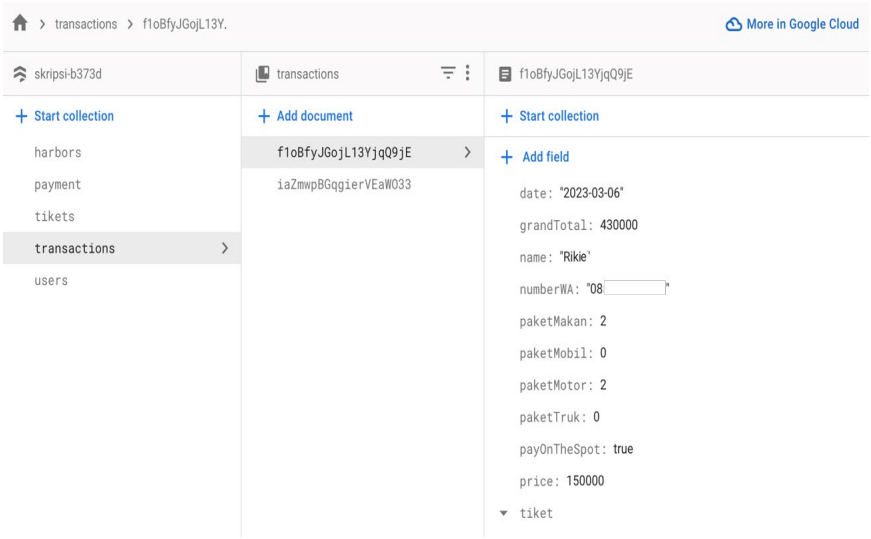


Gambar. 8. *Collection tiket*

4) Collection transaksi

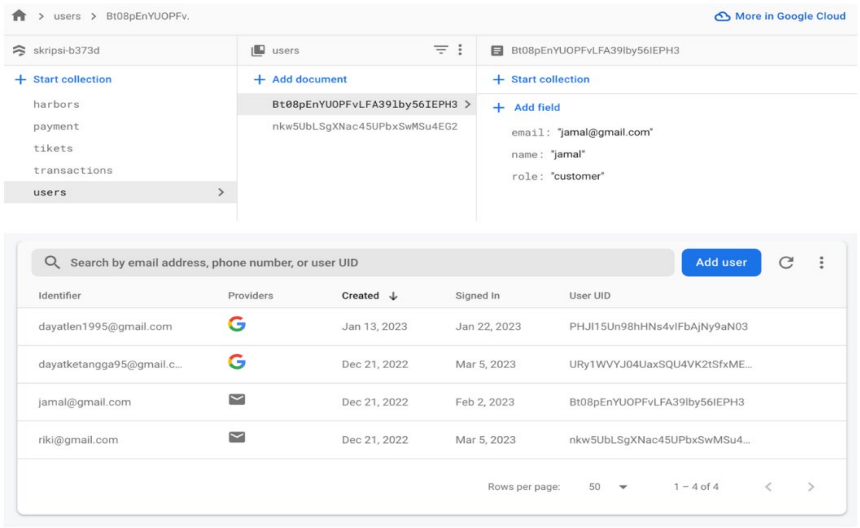
Pada Gambar 9 kita menyimpan sebuah data transaksi dari pelanggan yang didalamnya terdapat paket, data keberangkatan, total harga, nama pemesan, nomor *whatsapp*, harga dan tiket yang dipilih.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438



Gambar. 9. *Collection transaksi*

5) Collection user



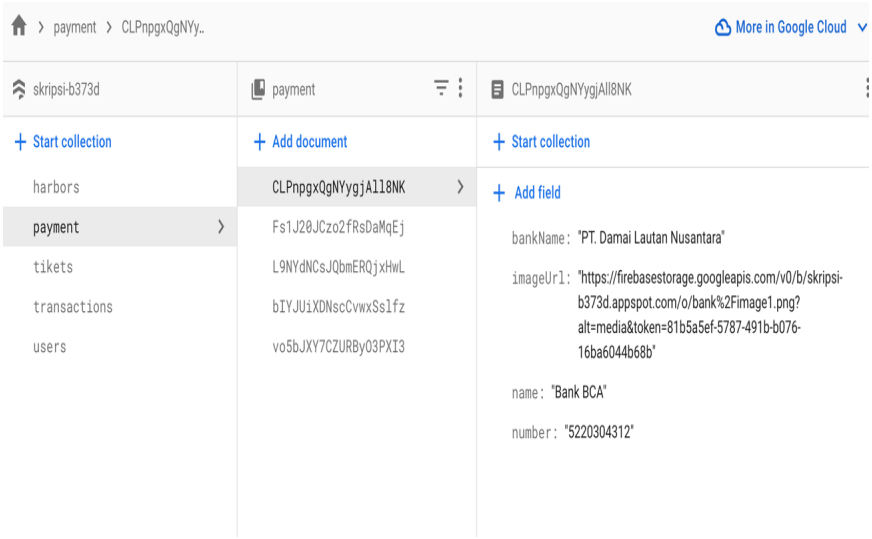
Gambar. 10. *Collection user*

Pada Gambar 10 diatas dibuat *authentication login* untuk pelanggan dengan dua metode, yakni *login* menggunakan akun *Google* secara langsung atau mendaftar terlebih dahulu. Jika *register* maka disimpan data seperti *email*, nama dan *password* pelanggan.

6) Collection payment

Pada Gambar 11 dibuat data *payment* untuk membuat metode pembayaran pada pelanggan untuk menyelesaikan transaksi tiket yang sudah di pesan. Di dalam data *payment* terdapat nama bank, nama pemilik rekening, nomer pemilik rekening dan gambar bank.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438



Gambar. 11. *Collection payment*

B. Logic Code

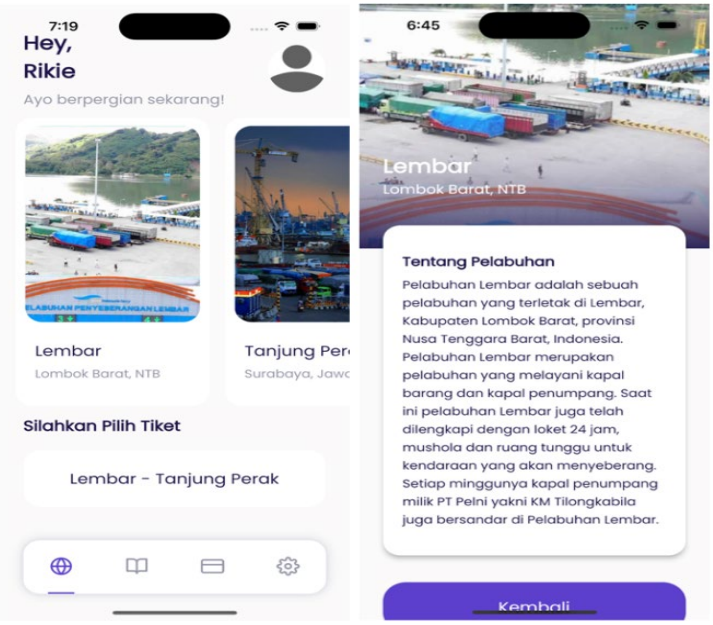
Berikut adalah contoh *logic code* yang program inti dari banyaknya *source code* pada aplikasi untuk menampilkan *list* detail dari pelabuhan. Untuk program kita mengambil data dari *collection kapal* kemudian membuat *try catch* untuk menghindari *error exception* yang tidak diinginkan *user*. Pada try kita membuat sebuah *list* dengan model data pelabuhan. Kemudian mengembalikan data *mapping* yang sudah diubah menganjadi *list*.

```
// insiasi collection harbor
final _harborFirestore =
FirebaseFirestore.instance.collection('harbors');

Future<List<HarborModel>> fetchHarbor() async {
  try {
    // mengambil seluruh data harbor
    QuerySnapshot result = await _harborFirestore.get();

    // menampilkan data list harbor dalam bentuk model json
    List<HarborModel> harbors = result.docs
        .map((e) =>
            HarborModel.fromJson(e.id, e.data() as Map<String,
dynamic>))
        .toList();
    return harbors;
  } catch (e) {
    // jika terjadi error maka akan menampilkan error
    rethrow;
  }
}
```

Logic Code 1. List pelabuhan

Pada Gambar 12 menampilkan halaman detail pelabuhan dan halaman awal yang menampilkan *list* dari pelabuhan dan *list* dari tiket yang bisa dipilih oleh pengguna. Kita juga menampilkan nama dari pelanggan dan ketika pelanggan sudah *login* pelanggan tidak perlu *login* berkali kali cukup sekali maka aplikasi langsung otomatis menampilkan halaman utama. Halaman utama juga menampilan *bottom* navigasi untuk pengguna membuka halaman *list* pesanan pengguna atau pengaturan, *bottom* navigasi ini akan memudahkan pengguna untuk berpindah halaman. Sedangkan untuk halaman detail pelabuhan kita menampilkan data dari pelabuhan seperti nama, deskripsi lokasi dan gambar pelabuhan dan memberikan tombol untuk kembali ke halaman utama.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438



Gambar. 12. Tampilan halaman utama

C.Hasil Uji Coba

Bedasarkan implementasi yang telah dilakukan maka pengujian dapat dilakukan. Sehingga dapat diketahui hasil yang diperoleh mampu memenuhi kebutuhan yang diharapkan secara optimal. Selanjutnya dibuatkan kode program untuk tiap pengujian.

2) PaketBloc testing

Pada pengijian ini, *base code testing* menggunakan *package* tambahan *bloc_testing* pada unit testing sehingga kita dapat memanggil fungsi *blocTest* untuk mendeklarasikan *build* untuk membuat *instace* dari *BLoC*, *expact* untuk memberikan hasil yang diharapkan dan *act* untuk melalukan aksi atau perubahan yang akan di lakukan pada BLoC. Contoh *base code* nya adalah sebagai merikut:

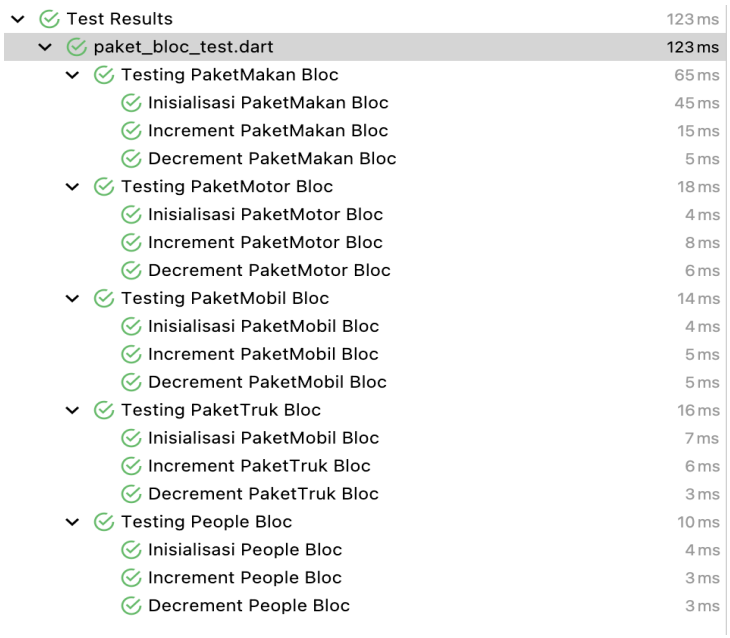
```
void main() {
      group("Testing PaketMakan Bloc", () {
         blocTest("Inisialisasi PaketMakan Bloc",
            build: () => PaketMakanCubit(), expect: () => []);

         blocTest("Increment PaketMakan Bloc",
            build: () => PaketMakanCubit(),
            expect: () => [1],
            act: (bloc) {
               bloc.increment();
            });

         blocTest("Decrement PaketMakan Bloc",
            build: () => PaketMakanCubit(),
            expect: () => [0],
            act: (bloc) {
               bloc.decrement();
            });
});

...
```

Hasil Uji WhiteBox

Pada Gambar 13 menampilkan semua testing yang dilakukan pada *BLoC*. Kelempok pertama kita mendeklaraasikan *paketMakanBloc* kemudian memanggil fungsi *increment* dan *decrement* apakah *state* terbaca dengan baik pada setiap fungsi yang ada di *paketMakanBloc*. Kelempok kedua kita mendeklaraasikan *paketMotorBloc* kemudian memanggil fungsi *increment* dan *decrement* apakah *state* terbaca dengan baik pada setiap fungsi yang ada di *paketMotorBloc*. Kelempok ketiga kita mendeklarasikan *pakatMobilBloc* kemudian

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438

memanggil fungsi *increment* dan *decrement* apakah *state* terbaca dengan baik pada setiap fungsi yang ada di *pakatMobilBloc*. Kelempok keempat mendeklaraasikan *paketTrukBloc* kemudian memanggil fungsi *increment* dan *decrement* apakah *state* terbaca dengan baik pada setiap fungsi yang ada di *paketTrukBloc*. Kelempok kelima kita mendeklaraasikan *paketPeopleBloc* kemudian memanggil fungsi *increment* dan *decrement* apakah state terbaca dengan baik pada setiap fungsi yang ada di *paketPeopleBloc*. Semua kelompok yang ditesting berhasil dijalakan karena fungsi yang di harapkan dan dijalankan sesuai dengan yang ada di masing masing *BLoC*.



Gambar. 13. Hasil *Testing PacketBLoC*

3) TransactionBloc testing

Pada pengijian ini, *base code testing* menggunakan *package* tambahan *bloc_testing* pada unit testing dan *moctail* untuk depedensi objek yang telah di *mock* agar dapat menggunakan API yang intuinsif yang kita gunakan seperti *when* pada program kita sehingga ketika memanggil fungsi *blocTest* untuk mendeklarasikan *build* untuk membuat *instace* dari *BLoC*, *exact* untuk memberikan hasil yang diharapkan dan *act* untuk melalukan aksi atau perubahan yang akan dilakukan pada *mocktail when* untuk mengembalikan jawaban dari API intuinsif agar dapat dipantau pada *event BLoC StateTransactions*.

```
class MockTransactionService extends Mock implements
TransactionService {}

void main() {
  late TransactionCubit transactionCubit;
  late TransactionService transactionService;

final transactionModel = {
  'totalPerson': 1,
  'userId': 'URy1WVYJ04UaxSQU4VK2tSfxMEw2',
  'name': 'Rike',
  'grandTotal': 180000,
  'price': 150000,
  'paketMakan': 2,
  'paketMotor': 0,
  'paketMobil': 0,
  'paketTruk': 0,
  'numberWA': "08          ",
  'date': "2023-03-10",
  'payOnTheSpot': true,
  'tiket': {
  "id": "UvM9MT3ikq1WoldfUKBM",
  'from': 'Lembar',
```

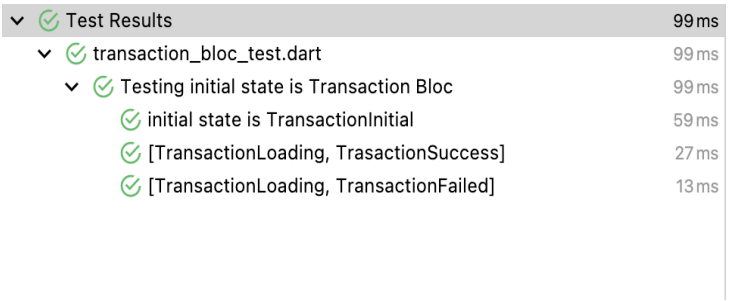
```
  'destination': 'Tanjung Perak',
  'imageUrl':
     'https://firebasestorage.googleapis.com/v0/b/skripsi-
b373d.appspot.com/o/tikets%2Fimage4.png?alt=media&to-
ken=a32338be-
5a56-4f1d-9a93-154df917fb28',
  'about':
      'Kapal DLN Batu Layar adalah armada operator PT. Damai
Lautan Nusantara yang melayani penyeberangan dengan jadwal ka-
pal
Lombok-Surabaya. Pemerintah telah menerbitkan SE Nomor 24 Ta-
hun 2022
tentang pedoman perjalanan dalam negeri.',
   'price': 150000,
   'photos': [
     'https://firebasestorage.googleapis.com/v0/b/skripsi-
b373d.appspot.com/o/tikets%2Fphotos%2Fimage4.png?alt=media&to-
ken=b78
5f769-f12f-4f88-a7b0-23caf7b81b89',
     'https://firebasestorage.googleapis.com/v0/b/skripsi-
b373d.appspot.com/o/tikets%2Fphotos%2Fimage5.png?alt=media&to-
ken=c43
4a5c3-4ab1-4313-a300-924d0ae08fba',
     'https://firebasestorage.googleapis.com/v0/b/skripsi-
b373d.appspot.com/o/tikets%2Fphotos%2Fimage6.png?alt=media&to-
ken=96f
4ba80-7bf2-4792-b353-1c68060ad1b6'
     ],
  }
};

setUp(() {
  transactionService = MockTransactionService();
  transactionCubit = TransactionCubit(transactionService);
});

group("Testing initial state is Transaction Bloc", () {
  blocTest("initial state is TransactionInitial",
      build: () => transactionCubit, expect: () => []);

  blocTest("[TransactionLoading, TrasactionSuccess]",
      build: () => transactionCubit,
      expect: () => [
            isA<TransactionLoading>(),
            isA<TransactionSuccess>(),
         ],
      act: (TransactionCubit cubit) {
        when(() =>
transactionService.fetchTransaction()).thenAnswer(
            (_) async => [
                  TransactionModel.fromJson(
                      'uwPs9YZrissKdUBCbdDq', transac-
tionModel)
                ]);
          cubit.fetchTransaction();
        });
    blocTest("[TransactionLoading, TransactionFailed]",
        build: () => transactionCubit,
        expect: () => [
          isA<TransactionLoading>(),
          isA<TransactionFailed>(),
        ],
        act: (TransactionCubit cubit) {
          when(() =>
transactionService.fetchTransaction()).thenAnswer(
                  (_) async => [
                TransactionModel.fromJson(
                    '', transactionModel)
              ]);
          cubit.fetchTransaction();
```

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1427-1438

```
                });
        });  }
```

Hasil Uji WhiteBox

Pada Gambar 14 menampilkan semua testing yang dilakukan pada *BLoC Cubit State* dimana menghasilkan fungsi transaksi gagal dan transaksi sukses berjalan dengan baik, untuk ***transaksi gagal*** pada Pogram diatas terlihat pada *when* kita memberikan data dengan *id* yang kosong sehingga data transaksi yang dikeluarkan gagal karena id yang seharusnya ada tetapi dimasukan masukan kosong sedangkan untuk transaksi sukses diberikan id untuk membaca apakah data dari id transaksi ada atau tidak. pada program diatas data transaksi disini ada maka transaksi yang dikembalikan pada Program *Cubit State* berjalan baik.



Gambar. 14. Hasil *Testing TransactionBLoC*

IV. KESIMPULAN

Dari hasil pengujian yang dilakukan, aplikasi telah berjalan dengan baik, setiap BLoC telah berhasil. Dan aplikasi tidak mengalami kendala yang besar. Uji BLoC Cubit dengan transaksi sukses dan gagal dapat berjalan dengan baik.

Untuk kedepannya, beberapa fitur yang belum diaplikasikan seperti fitur edit untuk penjadwalan ulang pemesanan, fitur pembayaran sehingga tidak perlu lagi konfirmasi pembayaran kepada admin, dan perbaikan User Interface sesuai dengan perkembangan.

UCAPAN TERIMA KASIH

Kami ucapkan terimakasih kepada seluruh civitas akademika Universitas Teknologi Digital Indonesia dan Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi, dengan dana hibah PDP dengan kontrak induk No. 181/E5/PG.02.00.PL/2023 dan kontrak turunan No.0423.21/LL5-INT/AL.04/2023.

DAFTAR PUSTAKA

[1] A. Hasoloan, "Sistem dan Prosedur Operasional Pelayanan Kapal dan Barang Berbasis Online Pada PT. Pelabuhan Indonesia I (Persero) Cabang Pelabuhan Belawan", Publik Reform. 2017;3(2).
[2] *M. Z. Kotbi, 2018, "Desain Piranti Lunak Penerbitan Tiket Angkutan Penyeberangan: Studi Kasus Pelabuhan Lembar", https://repository.its.ac.id/50610/%0Ahttps://repository.its.ac.id/50610/1/04411340000024_undergraduate_thesis.pdf*
[3] H. U. Hatari. et.al., 2019, "Aplikasi Pemesanan Tiket Kapal Laut Online Berbasis Android," Jurnal Teknologi Informasi, 2617(2), 12–17.
[4] Khairatin, 2019, "Pemilihan kapal laut menggunakan algoritma apriori dalam aplikasi reservasi tiket berbasis web (studi kasus: Pulau Tidung-Kali Adem)" , In Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta, http://repository.uinjkt.ac.id/dspace/handle/123456789/47485%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/47485/1/KHAIRATIN-FST.pdf
[5] *D, Muhar. ,2018, "Tutorial Flutter #1: Pengenalan dan Persiapan Pemrograman Mobiledengan Flutter," [Online] Tersedia: https://www.petanikode.com/flutter-linux/.*
[6] M. R. Alwi, 2022 "Pengenalan State Management Flutter dan Jenis-jenisnya." Caraguna.Com. https://caraguna.com/pengenalan-state-management-flutter/
[7] L. Hoang, 2019, "State Management Analyses of the Flutter Application", Doctoral dissertation, BSc. Thesis. Metropolia University of Applied Sciences