LAPORAN TUGAS AKHIR
TEKNOLOGI PEMGRORAMAN MOBILE

Disusun oleh :

NIM : 123210011

NAMA : Muhammad Harish Wijaya

KELAS : IF-B

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA

2024

DAFTAR ISI

GAMBARAN PROJEK

CelebMeet adalah sebuah aplikasi mobile yang dirancang untuk memudahkan pengguna dalam memesan tiket untuk acara meet & greet dengan selebriti favorit mereka. Dikembangkan menggunakan bahasa pemrograman Dart dan framework Flutter, CelebMeet memberikan pengalaman pengguna yang responsif dan mulus di berbagai perangkat. Pengguna dapat menjelajahi jadwal acara meet & greet yang diadakan oleh berbagai selebriti. Aplikasi ini memberikan pengguna akses langsung untuk memesan tiket, melihat detail acara, dan mengatur pertemuan dengan selebriti yang mereka kagumi. CelebMeet memberikan kemudahan dan kenyamanan bagi pengguna untuk merencanakan dan mengikuti acara meet & greet selebriti dengan mudah.

CelebMeet memiliki beberapa fitur utama yang membedakannya, antara lain: sistem login yang aman dengan enkripsi data menggunakan session dan penyimpanan dalam database untuk menjaga keamanan dan privasi pengguna; koneksi dengan Web API untuk mengambil data tentang selebriti; bottom navigation yang terdiri dari profil pengguna, saran & kesan, serta opsi logout; menu konversi waktu dan uang untuk memudahkan pengguna dalam menghitung perbedaan waktu antar wilayah serta konversi mata uang saat berinteraksi dengan pengguna dari berbagai negara; dan fasilitas pencarian, pemilihan, dan notifikasi untuk memungkinkan pengguna mencari, memilih, dan mendapatkan pemberitahuan tentang selebriti favorit mereka.

LOGIN ENKRIPSI DENGAN SESSION DAN DATABASE

1. Menu register dengan enkripsi data password
   Langkah Pengerjaan :

   - Proses enkripsi menggunakan algoritma Advanced Encryption Standard (AES) yang diimplementasikan melalui paket encrypt di Dart.
   - Tambahkan di yaml dependencies : encrypt: ^5.0.3 (untuk enkripsi).
   - Pembuatan sebuah kunci (key) dan vektor inisialisasi (IV) dibuat. Kunci ini digunakan untuk mengenkripsi dan mendekripsi data.
   - Penkripsi dilakukan menggunakan objek Encrypter dari paket encrypt.

   Penjelasan Kode :

SOURCE CODE	PENJELASAN
`if (user.isNotEmpty && pass.isNotEmpty) {` `  final key =` `encrypt.Key.fromUtf8('projektpm0000000000000000000000');` `  final iv = encrypt.IV.fromUtf8("1234567890123456");` `  final encrypter =` `encrypt.Encrypter(encrypt.AES(key));` `  // Buka loginBox` `  var box = Hive.box<LoginModel>(loginBox);` `  // await box.clear(); // hapus semua data di box` `  if (!box.containsKey(user)) {` `  // Tambahkan username and password to Hive` `  final encryptedpass = encrypter.encrypt(pass, iv: iv);` `  var newAkun = LoginModel(username: user,` `password: encryptedpass.base64);` `  await box.put(user, newAkun);`	Sebuah kunci (key) dan vektor inisialisasi (IV) dibuat. Kunci ini digunakan untuk mengenkripsi dan mendekripsi data. Kunci diinisialisasi dengan sebuah string 'projektpm0000000000000000000000' dan IV diinisialisasi dengan string '1234567890123456'. Setelah kunci dan IV dibuat, proses enkripsi dilakukan menggunakan objek Encrypter dari paket encrypt. Tekstual password yang dimasukkan oleh pengguna dienkripsi sebelum disimpan dalam database. Dalam kode tersebut, password dienkripsi menggunakan kunci dan IV yang telah disiapkan.

2. Menu login dengan akses database Hive, session SharedPreferences, dan dekripsi password
   Langkah Pengerjaan :

   - Tambahkan dependencies :
     shared_preferences: ^2.2.3 (untuk session menggunakan sharedpref),
     hive_flutter: ^1.1.0 (untuk database menggunakan hive),
     hive: ^2.0.4,
     path_provider: ^2.0.3

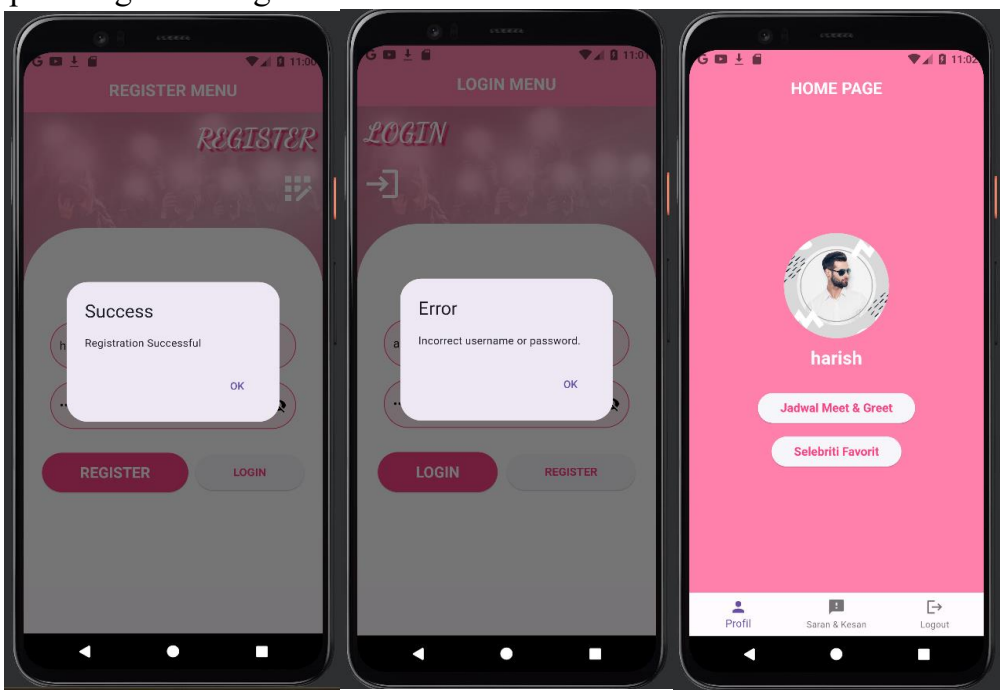
- Tambahkan dev_dependencies:
  hive_generator: ^1.1.0,
  build_runner: ^2.1.2
- Fungsi main dibuat async + buat variabel global String loginBox = 'loginBox', lalu tambahkan await Hive.initFlutter(); untuk inisialisasi.
- Buat file model yang berisi object dari loginBox dan tambahkan part 'namafile.g.dart'; sebelum generate adapter.
- Jalankan di terminal untuk generate adapter : dart run build_runner build jika berhasil muncul file baru namafile.g.dart.
- Tambahkan di main untuk memanggil adapter yang baru saja digenerate : Hive.registerAdapter<LoginModel>(namaAdapter());
  await Hive.openBox<LoginModel>(loginBox); //openbox bernama loginBox
- Menggunakan Box di login untuk mengecek data di database. Saat pengecekan password didekripsi menggunakan algoritma dan kunci saat enkripsi.
- Membuat SharedPref bernama loginData dan membuat fungsi untuk mengecek sesi, apakah user sudah pernah login/belum.
- Jika berhasil login , Sharedpref akan menyimpan data username dan boolean newSesi : false.

Penjelasan Kode :

SOURCE CODE	PENJELASAN
	Proses dekripsi dilakukan menggunakan algoritma yang sama dengan proses enkripsi sebelumnya, yaitu Advanced Encryption Standard (AES) yang diimplementasikan melalui paket encrypt di Dart. Penggunaan Hive terlihat pada bagian pengecekan username dan password yang tersimpan di dalam box Hive. Untuk mengetahui apakah pengguna sudah login sebelumnya atau tidak, aplikasi menggunakan session yang disimpan menggunakan SharedPreference.

``dart import 'package:hive_flutter/hive_flutter.dart'; part 'login_model.g.dart'; @HiveType(typeId: 1) class LoginModel extends HiveObject { @HiveField(0) String username; @HiveField(1) String password; LoginModel({required this.username, required this.password}); } ```	Membuat model dart untuk LoginModel dan jalankan di terminal untuk generate adapter (dart run build_runner) build jika berhasil muncul file baru namafile.g.dart.

3. Tampilan login dan register




```
I/flutter (25174): Username: harish, Password: XFsk+T/35VdGgBaU0SY1og==
I/flutter (25174): Username: harish, Password Decrypted: admin
```

```
I/flutter (25174): Username: harish, Password Decrypted: admin
I/flutter (25174): LOGIN SUCCESSFUL!
```

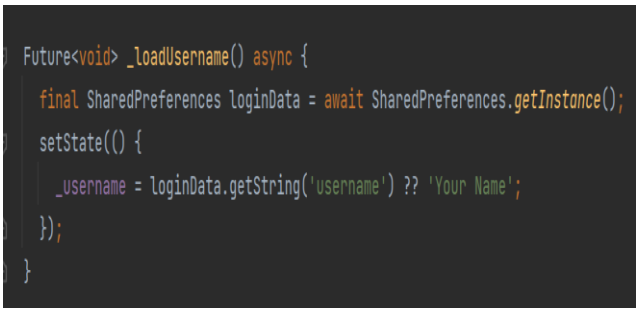
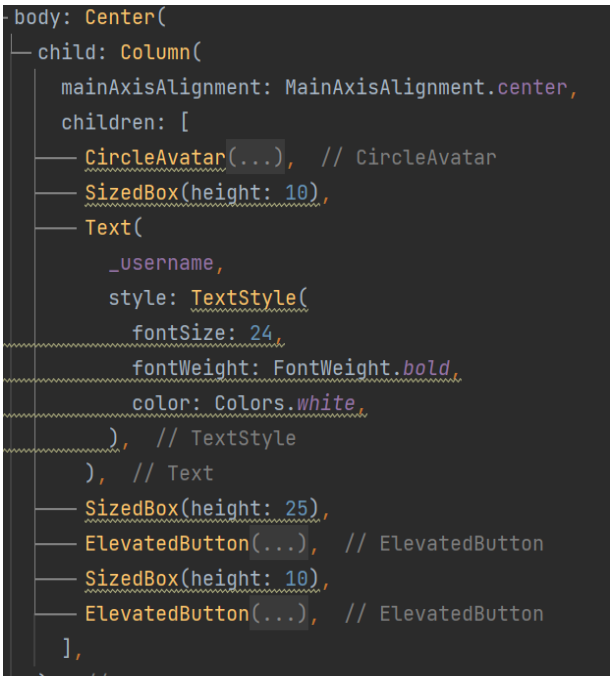
MENU BOTTOM NAVIGATION

1. Menu profil

   Langkah Pengerjaan :

   - Membuat method _loadUsername() untuk memuat nama pengguna dari SharedPreferences saat halaman profil dibuka.
   - Membuat tampilan profil berupa foto profil pengguna dan nama pengguna.
   - Meletakkan tombol untuk menu Jadwal Meet & Greet dan Selebriti Favorit.
   - Membuat navigasi ke halaman lain.

   Penjelasan Kode :

SOURCE CODE	PENJELASAN
	_loadUsername(): Metode ini menggunakan Shared Preferences untuk mendapatkan nama pengguna yang telah disimpan sebelumnya. Jika nama pengguna tidak tersedia, maka akan ditampilkan 'Your Name'.
	CircleAvatar: Widget ini menampilkan foto profil pengguna. Gambar diambil dari URL yang disediakan. Text: Widget ini menampilkan nama pengguna yang telah dimuat menggunakan _loadUsername(). ElevatedButton: Widget ini menampilkan tombol untuk menu Jadwal Meet & Greet dan Selebriti Favorit.

2. Menu saran & kesan

   Langkah Pengerjaan :

   - Gunakan widget Scaffold sebagai kerangka utama halaman.
   - Membuat input saran dan kesan menggunakan widget Card.
   - Mengatur Dekorasi.

Penjelasan Kode :

SOURCE CODE	PENJELASAN
	Container: Widget ini digunakan untuk menempatkan seluruh konten halaman. Dalam kode ini, digunakan untuk menerapkan dekorasi gradient background dan gambar latar belakang. SingleChildScrollView: Digunakan agar konten dapat di-scroll jika melebihi ukuran layar. Text: Digunakan untuk menampilkan teks, seperti judul "Saran & Kesan". Card: Widget ini digunakan untuk menampilkan input Kesan dan Saran. Dalam kode ini, digunakan untuk memberikan latar belakang pada bagian input Kesan dan Saran.

3. Menu Logout

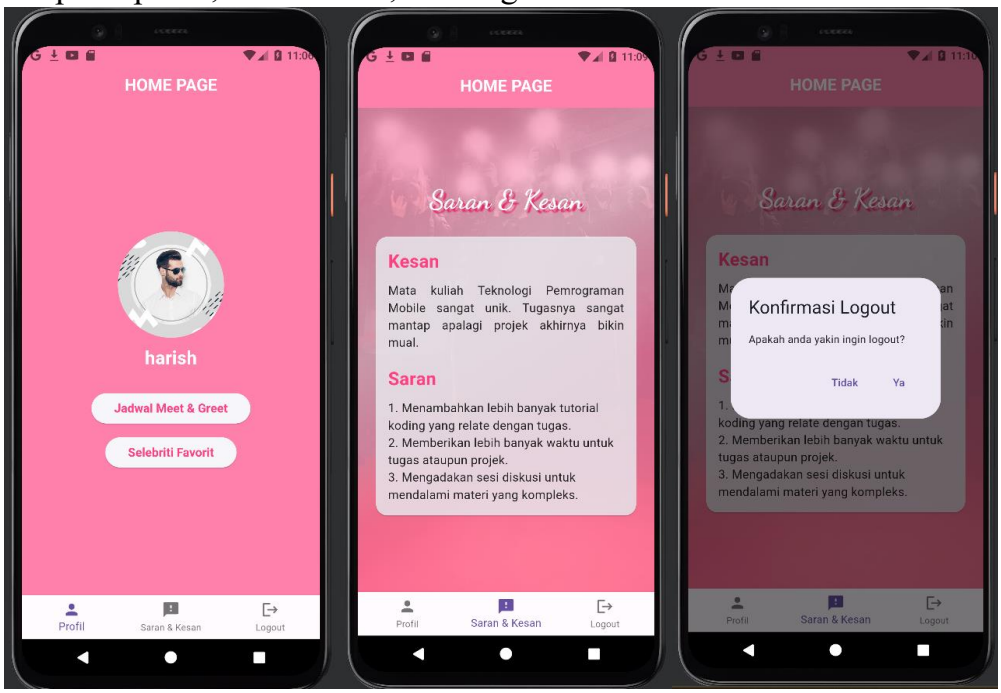
Langkah Pengerjaan :

- Membuat fungsi onTabTapped yang menerima parameter berupa index dari tab yang ditekan.
- Dalam fungsi ini, dilakukan pengecekan apakah index tab yang ditekan adalah 2 (tab logout) atau bukan jika index adalah 2, tampilkan dialog konfirmasi logout.
- Jika user memilih untuk konfirmasi logout maka data yang ada dalam SharedPreferences semuanya dihapus.

Penjelasan Kode :

SOURCE CODE	PENJELASAN
`return AlertDialog(` `  title: Text("Konfirmasi Logout"),` `  content: Text("Apakah anda yakin ingin logout?"),` `  actions: [` `  TextButton(...),  // TextButton` `  TextButton(` `  onPressed: () {` `  // Lakukan logout` `  SharedPreferences.getInstance().then((logindata) {` `  logindata.setBool('login', true);` `  logindata.remove('favoriteNames');` `  logindata.remove('favoritePhotos');` `  logindata.remove('favoriteInstagrams');` `  logindata.remove('username');` `  print('All data cleared.');` `  Navigator.pushReplacement(` `  context,` `  MaterialPageRoute(builder: (context) => MeetCelebApp()),` `  );` `  });` `  },`	Membuat dialog konfirmasi logout menggunakan AlertDialog. SharedPreferences.getInstance() : Mengambil instance dari SharedPreferences untuk menyimpan data login dan preferensi. logindata.setBool('login', true): Mengubah nilai boolean 'login' menjadi true, menandakan bahwa pengguna sudah logout. logindata.remove(...): Menghapus data-data yang terkait dengan login dan preferensi favorit pengguna.

4. Tampilan profil, saran kesan, dan logout



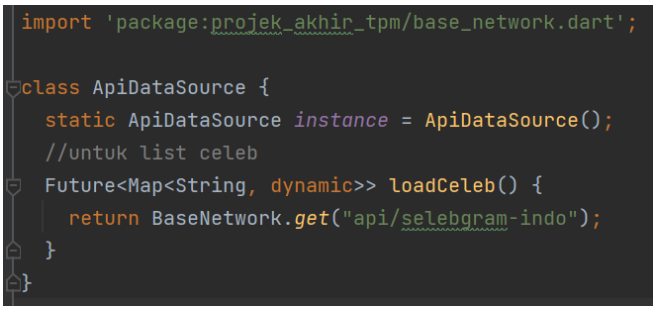
API

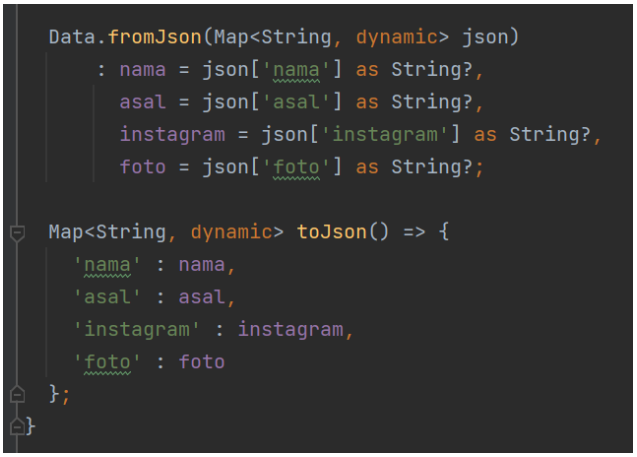
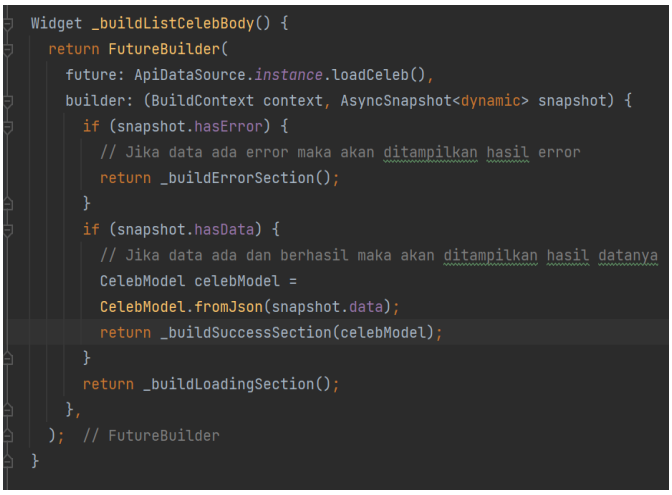
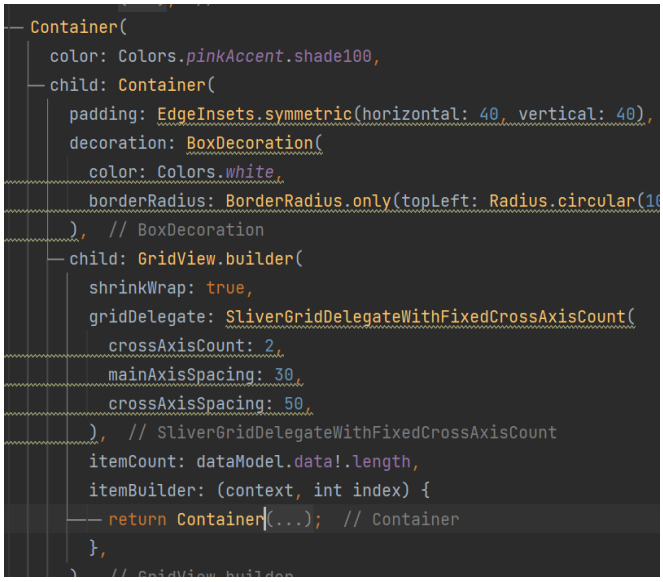
1. Menu celeb list

   Langkah pengerjaan :

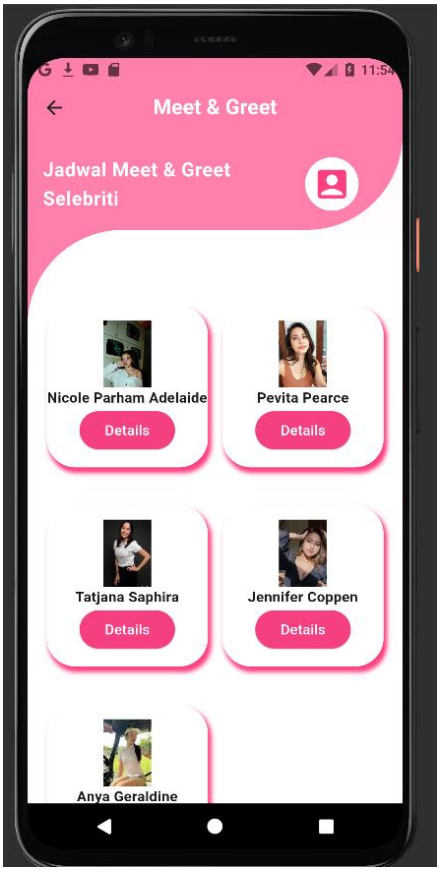
   - Tambahkan pada dependencies .yaml: http: ^0.13.4
   - Tambahkan permission pada AndoridManifest.xml (android/app/src/main) : <uses-permission android:name="android.permission.INTERNET" />
   - Membuat BaseNetwork untuk link request dari API.
   - Membuat api data source sebagai controller untung endpoint loadCeleb().
   - Membuat model CelebModel() dengan bantuan https://tiendung01023.github.io/json_to_dart/
   - Panggil model yang dibutuhkan.
   - Membuat tampilan celeb list yang menampilkan semua data dari endpoint API.

   Penjelasan Kode :

SOURCE CODE	PENJELASAN
	Membuat base url http dari url : https://dev-api-q2od2bwu5a-uc.a.run.app/
	Membuat api data source untuk mengambil endpoint yaitu : api/selebgram-indo

	Mengubah data model dengan mengkonversi model yang berupa JSON menjadi Model Dart.
	Mengecek snapshot data jika data ada maka akan mengembalikan _buildSuccessSection
	GridView.builder: menampikan semua data dari API yang sudah diconvert menjadi model dart

2. Tampilan celeb list



I/flutter (25174): [BASE_NETWORK] - BaseNetwork - response : {"data":[{"nama":"Nicole Parham Adelaide","asal":"Jakarta",
"instagram":"https://www.instagram.com/nicoleparham_","foto":"https://i.pinimg.com/564x/98/9b/e2/989be23653fd2e69ce15276159fa64a8.jpg"},
{"nama":"Pevita Pearce","asal":"Jakarta","instagram":"https://www.instagram.com/pevpearce/","foto":"https://i.pinimg
.com/564x/04/8a/38/048a381589d226a5a6d435ef00d203d4.jpg"},{"nama":"Tatjana Saphira","asal":"Jakarta","instagram":"https://www.instagram
.com/tatjanasaphira/","foto":"https://i.pinimg.com/564x/14/a4/96/14a4961a3bf6752fb14a5878d097ba91.jpg"},{"nama":"Jennifer Coppen",
"asal":"Jakarta","instagram":"https://www.instagram.com/jennifercoppenreal20/","foto":"https://i.pinimg
.com/564x/36/1d/cf/361dcf27ec2fb7c0981f5e0e04bb1ae8.jpg"},{"nama":"Anya Geraldine","asal":"Jakarta","instagram":"https://www.instagram

KONVERSI MATA UANG DAN WAKTU

1. Menu pembelian tiket

Langkah Pengerjaan :

- Tambahkan di yaml dependencies : intl: ^0.19.0 (untuk konversi).
- Mendefinisikan kelas CurrencyConverter untuk melakukan konversi mata uang. Ini termasuk mengonversi jumlah uang dari satu mata uang ke mata uang lainnya.
- Membuat kelas TicketPurchasePage sebagai StatefulWidget yang akan menampilkan halaman pembelian tiket.
- Menambahkan properti celeb dan i ke kelas TicketPurchasePage untuk menerima data selebriti dan indeksnya.
- Menginisialisasi state awal untuk zona waktu, mata uang, jumlah tiket, dan status favorit.
- Membuat fungsi-fungsi untuk mengonversi waktu, mendapatkan harga tiket, menghitung total harga, dan meluncurkan URL.
- Mengimplementasikan logika untuk menambahkan atau menghapus data favorit ke dan dari SharedPreferences.
- Membangun antarmuka pengguna dengan berbagai widget untuk menampilkan detail acara, termasuk gambar selebriti, detail acara, zona waktu, mata uang, jumlah tiket, harga, dan tombol untuk membeli tiket.
- Menampilkan dialog pembayaran ketika tombol "Beli Tiket" ditekan, dimana pengguna dapat memilih bank dan mengonfirmasi pembayaran.

Penjelasan Kode :

SOURCE CODE	PENJELASAN
	CurrencyConverter: Kelas ini menyediakan metode formatCurrency untuk mengonversi jumlah uang dari satu mata uang ke mata uang lainnya berdasarkan pilihan pengguna. Terdapat 4 Case yaitu USD, IDR, MYR, dan JPY.

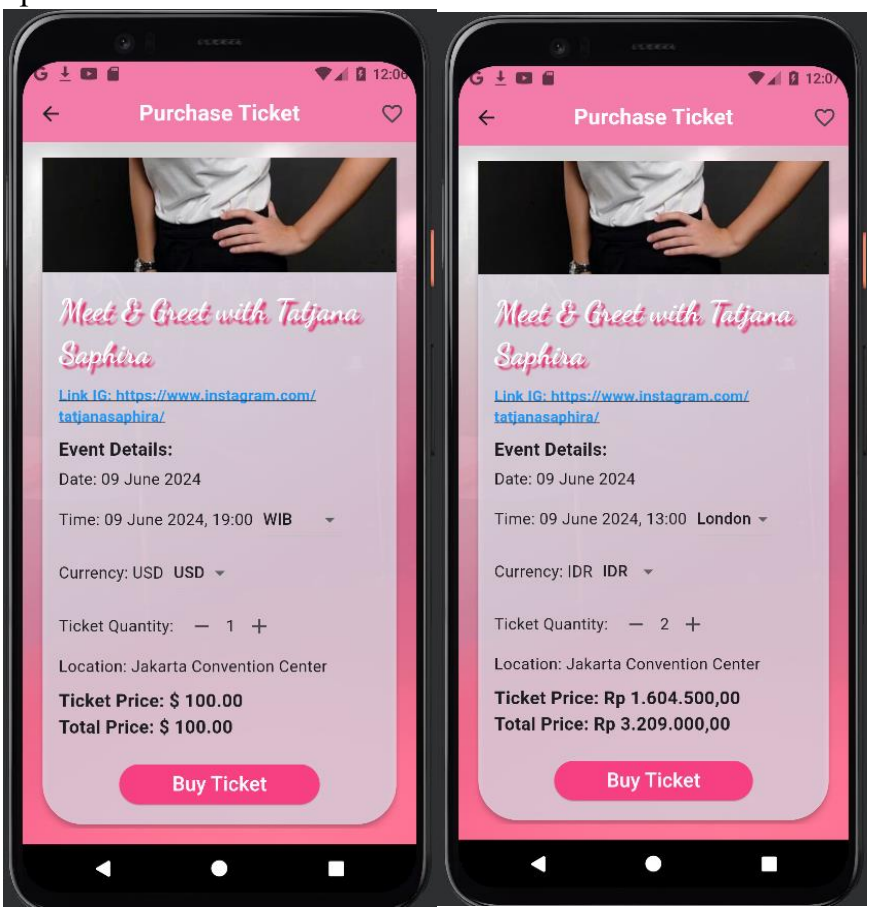
```dart
// Class untuk fungsi konversi mata uang
class CurrencyConverter {
  static final _formatter =
      NumberFormat.currency(locale: 'en_US', symbol: '\$ ');

  static String formatCurrency(double amount, {String currency = 'USD'}) {
    switch (currency) {
      case 'USD':
        return _formatter.format(amount);
      case 'IDR':
        return NumberFormat.currency(locale: 'id_ID', symbol: 'Rp ')
            .format(amount * 16045); // 1 USD = 16045 IDR (Mei 2024)
      case 'MYR':
        return NumberFormat.currency(locale: 'ms_MY', symbol: 'RM ')
            .format(amount * 4.712); // 1 USD = 4.712 MYR (Mei of 2024)
      case 'JPY':
        return NumberFormat.currency(locale: 'ja_JP', symbol: '¥ ')
            .format(amount * 156.97); // 1 USD = 156.97 JPY (Mei 2024)
      default:
        return _formatter.format(amount);
    }
  }
}
```

```dart
// Fungsi untuk konversi waktu
String convertTime(DateTime dateTime, String timeZone) {
  // Set time zone
  dateTime = dateTime.toUtc(); // dikonversi ke UTC
  print("UTC : $dateTime");
  switch (timeZone) {
    case 'WIB':
      dateTime = dateTime.add(Duration(hours: 7)); // UTC+7 for WIB
      print("new time : $dateTime");
      break;
    case 'WITA':
      dateTime = dateTime.add(Duration(hours: 8)); // UTC+8 for WITA
      print("new time : $dateTime");
      break;
    case 'WIT':
      dateTime = dateTime.add(Duration(hours: 9)); // UTC+9 for WIT
      print("new time : $dateTime");
      break;
    case 'London':
      dateTime = dateTime.add(Duration(hours: 1)); // UTC+1 for London
      print("new time : $dateTime");
      break;
  }
```

convertTime: Fungsi ini mengonversi waktu acara ke zona waktu yang dipilih pengguna dan mengembalikannya sebagai string dalam format tertentu.

Terdapat 4 Case yaitu WIB, WITA, WIT, dan London.

2. Tampilan pembelian tiket



PENCARIAN, PEMILIHAN, DAN NOTIFIKASI

1. Menu Favorit

Langkah Pengerjaan :

- Membuat kelas FavoritePage sebagai StatefulWidget yang akan menampilkan daftar selebriti favorit.
- Menggunakan SharedPreferences untuk menyimpan dan mengambil daftar nama selebriti favorit, foto, dan Instagram dari penyimpanan lokal.
- Implementasi fungsi loadFavoriteData untuk memuat data favorit dari penyimpanan lokal saat widget diinisialisasi.
- Membuat fungsi removeFavorite untuk menghapus selebriti dari daftar favorit dan memperbarui penyimpanan lokal.
- Menambahkan fungsi pencarian untuk menampilkan dialog pencarian saat tombol pencarian ditekan.
- Menampilkan daftar selebriti favorit dalam ListView, dimana setiap item terdiri dari foto, nama, tautan Instagram, dan ikon hati sehingga terdapat pemilihan untuk menghapus selebriti dari daftar favorit.
- Menambahkan notifikasi menggunakan snackbar jika terdapat perubahan baik dalam penambahan atau penghapus dari data favorit.

Penjelasan Kode :

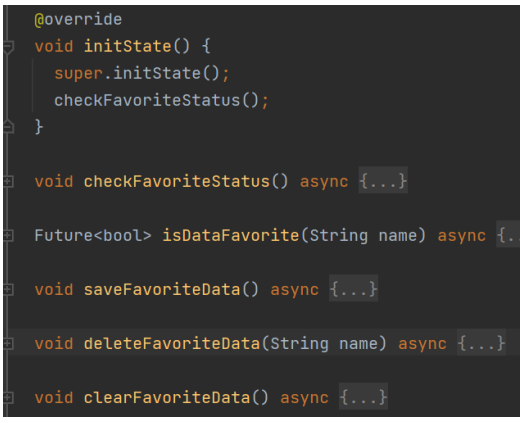
SOURCE CODE	PENJELASAN
	buildSuggestions: Metode ini digunakan untuk menampilkan saran pencarian saat pengguna belum memasukkan kueri atau kuerinya kosong. Daftar saran ini juga akan disaring berdasarkan kesamaan teks dengan daftar nama selebriti favorit. where((name)=> name.toLowerCase().contains(query.to LowerCase())): Ini adalah ekspresi fungsi yang digunakan untuk melakukan filter pada favoriteNames. Ekspresi ini berfungsi untuk memeriksa apakah name (nama selebriti) mengandung query yang dimasukkan oleh pengguna

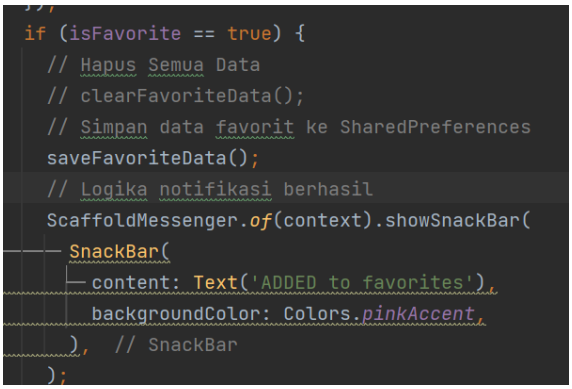
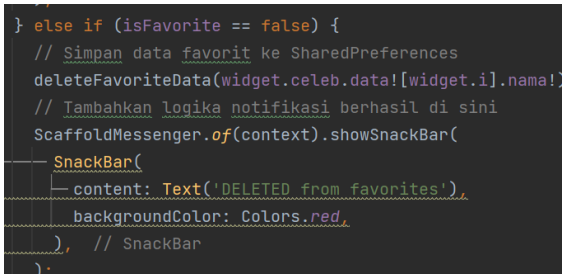
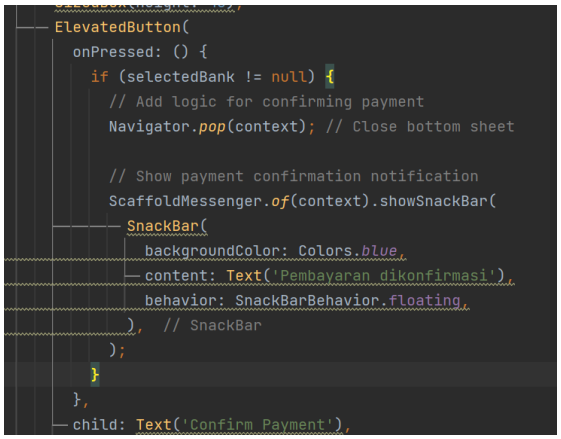
	Implementasi fungsi loadFavoriteData untuk memuat data favorit dari penyimpanan lokal saat widget diinisialisasi.
	Implementasi fungsi removeFavorite untuk menghapus selebriti dari daftar favorit dan memperbarui penyimpanan lokal. Disini user bisa memilih mana daftar yang mau diremove dari list favorit.

2. Menu tiket dan pembayaran
   Langkah Pengerjaan :

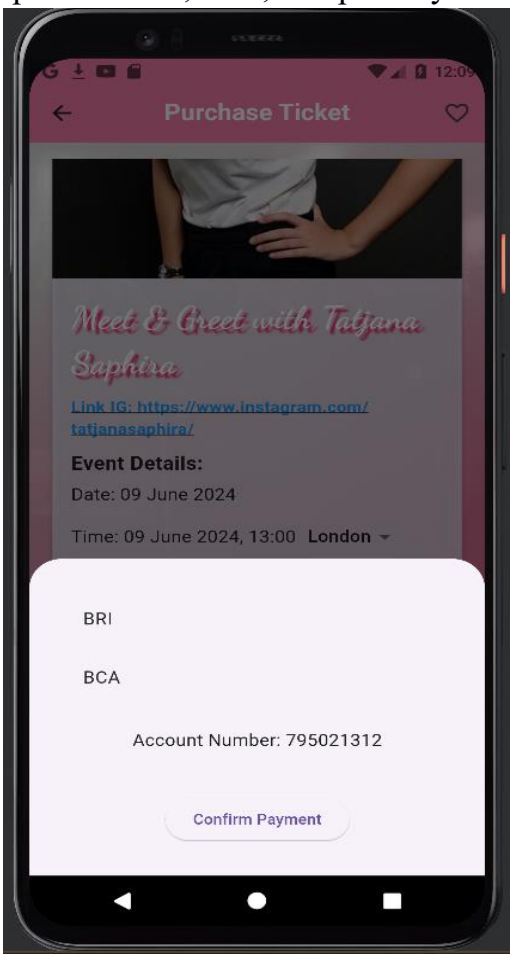
   - Membuat fungsi yang berhubungan dengan logic favorit (check,save,delete,clear).
   - Membuat dialog pembayaran agar pengguna dapat memilih saat ingin membeli tiket.
   - Membuat notifikasi saat ada perubahaan pada list data favorit dan notifikasi ketika pembeli sudah konfirmasi pembayaran.

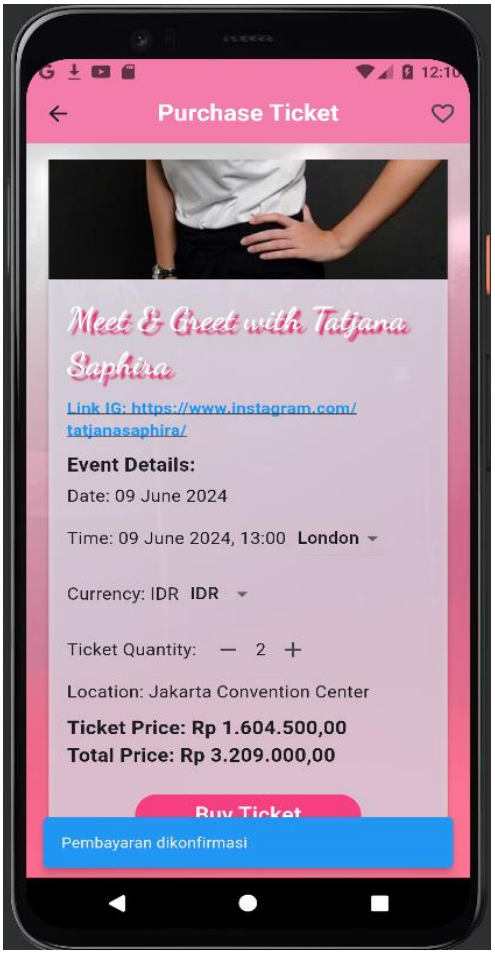
   Penjelasan Kode :

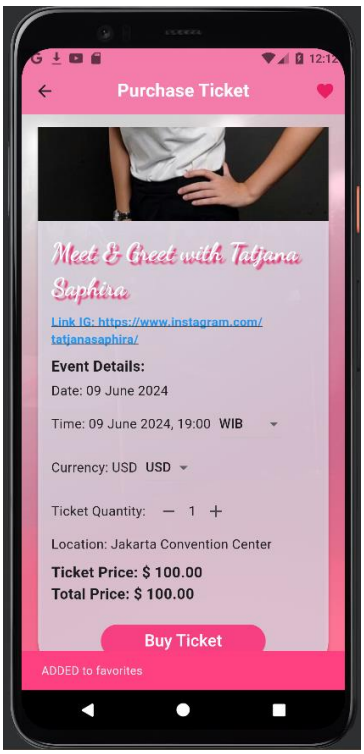
SOURCE CODE	PENJELASAN
	Membuat fungsi untuk cek, menambah, menghapus data favorit.

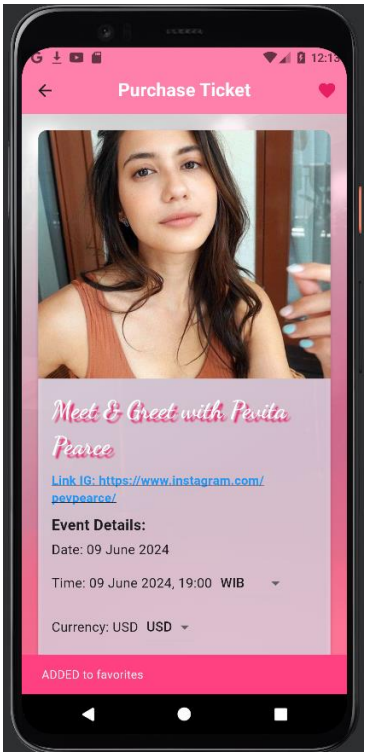
	Jika terdapat penambahan data favorit maka akan memunculkan notifikasi bahwa data berhasil ditambahkan.
	Jika terdapat penghapusan data favorit maka akan memunculkan notifikasi bahwa data berhasil dihapus.
	ListTile digunakan untuk menampilkan pilihan bank pembayaran. Ketika pengguna memilih salah satu diantaranya, variabel selectedBank diatur pilihan.
	Ketika pengguna menekan tombol ini, logika yang diperlukan untuk mengonfirmasi pembayaran dilakukan. Setelah pembayaran dikonfirmasi, notifikasi muncul untuk memberi tahu pengguna bahwa pembayaran telah dikonfirmasi. Notifikasi ini ditampilkan sebagai snack bar di bagian bawah layar dan memberi umpan balik visual kepada pengguna.

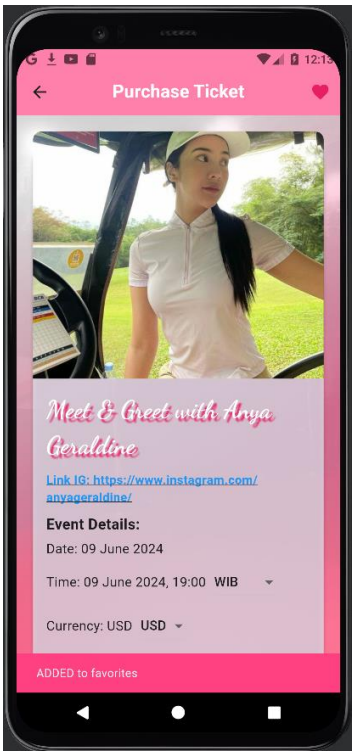
3. Tampilan favorit, tiket, dan pembayaran

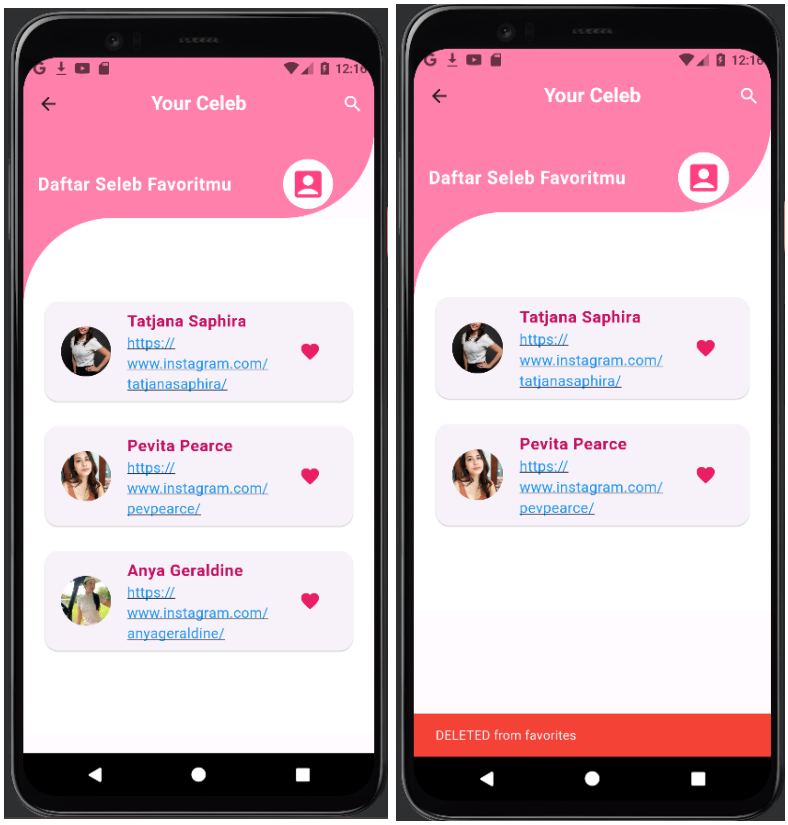


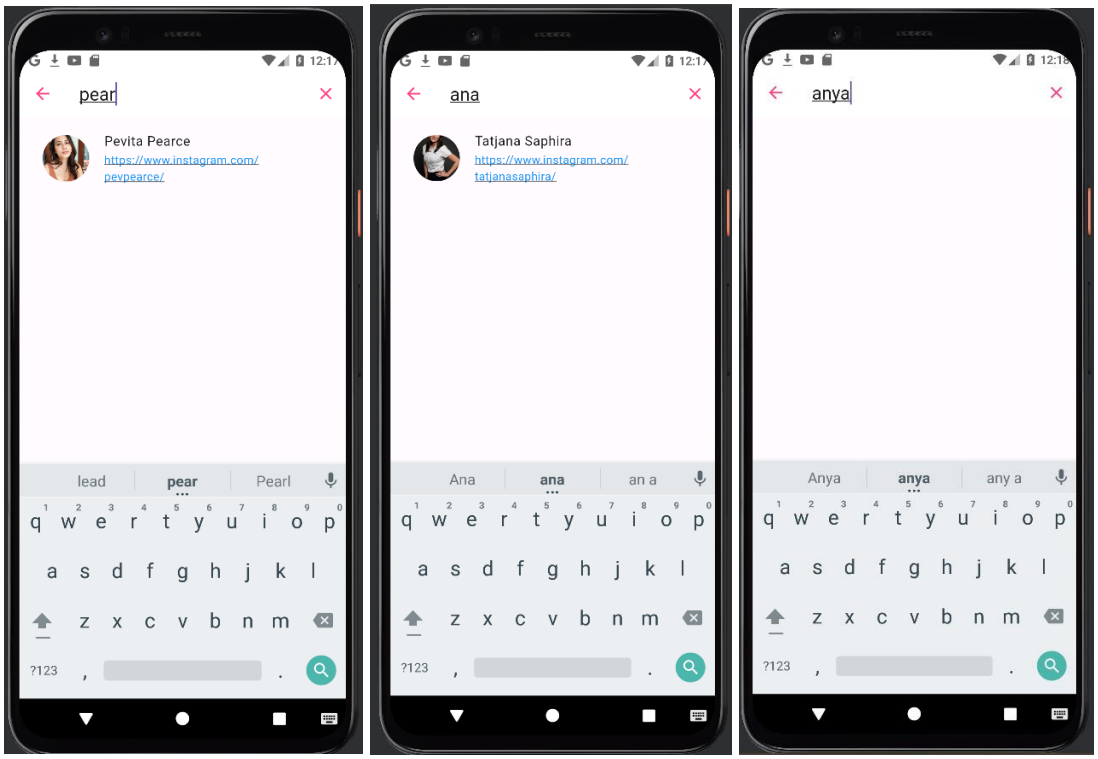









Your Celeb
Daftar Seleb Favoritmu
Tatjana Saphira
https://www.instagram.com/tatjanasaphira/
Pevita Pearce
https://www.instagram.com/pevpearce/
Anya Geraldine
https://www.instagram.com/anyageraldine/
Your Celeb
Daftar Seleb Favoritmu
Tatjana Saphira
https://www.instagram.com/tatjanasaphira/
Pevita Pearce
https://www.instagram.com/pevpearce/
DELETED from favorites


pear
Pevita Pearce
https://www.instagram.com/pevpearce/
ana
Tatjana Saphira
https://www.instagram.com/tatjanasaphira/
anya

LAMPIRAN

1. Link GitHub :
   https://github.com/MasPey/projek_akhir_tpm/
2. Link API :
   https://dev-api-q2od2bwu5a-uc.a.run.app/api/selebgram-indo