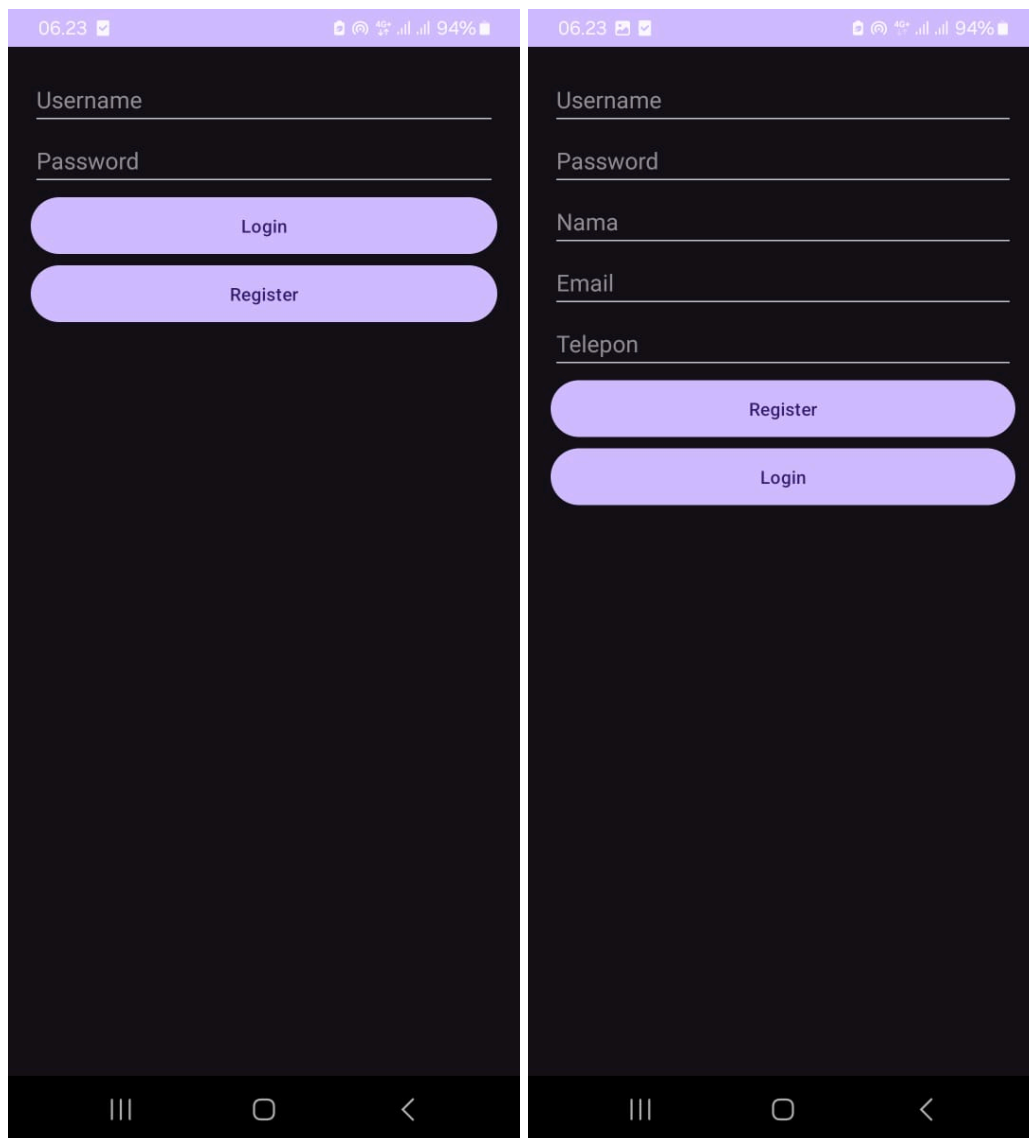


Vater E. T. Prawira
C11210039

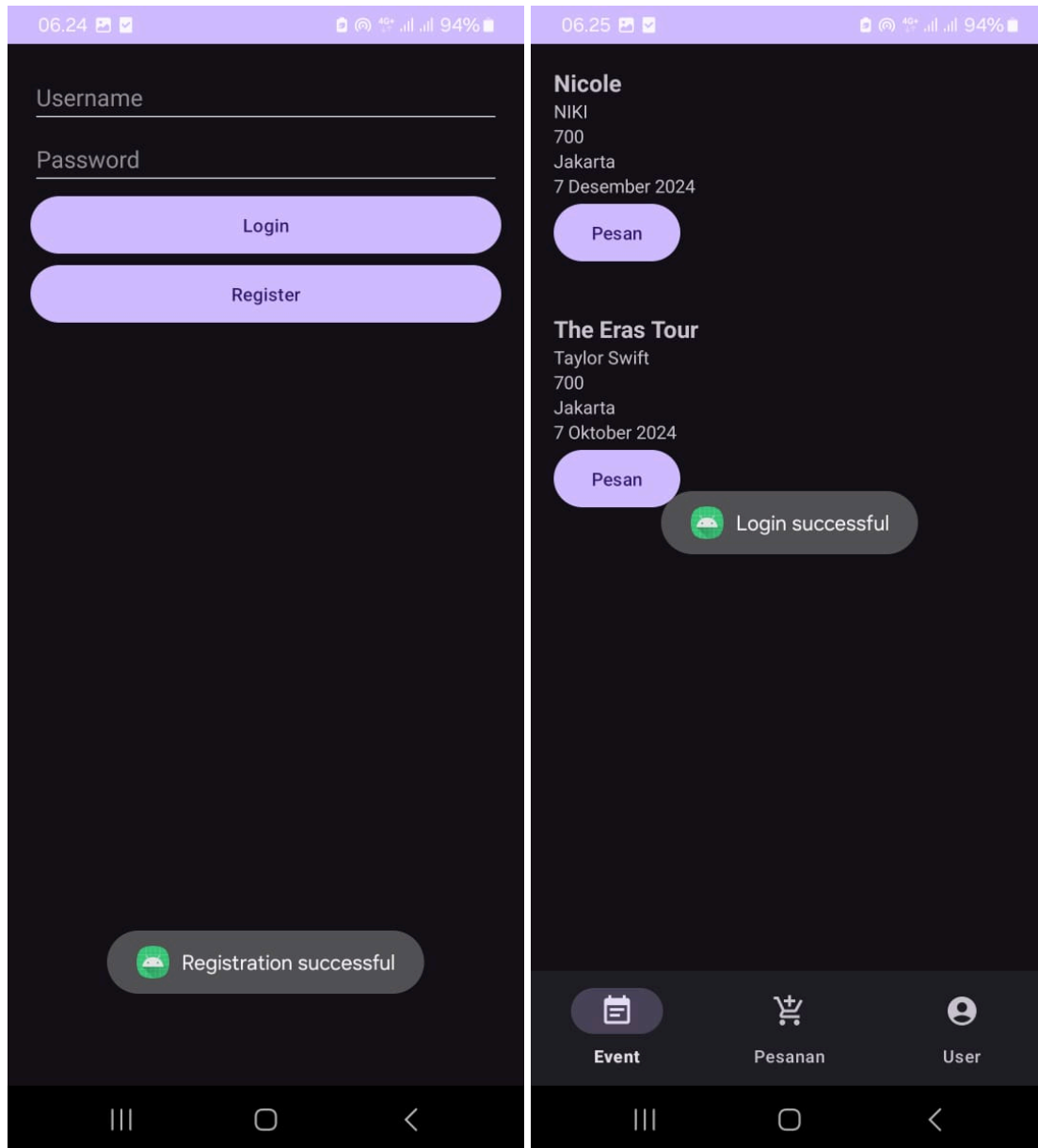
Laporan Implementasi Mobile Client Sistem Pemesanan Tiket Konser Online

Github: <https://github.com/vaterprawira/Project-PAT>

Ketika aplikasi pertama kali dibuka, user akan diarahkan ke halaman login. Jika user belum mempunyai akun, user dapat melakukan registrasi atau membuat akun.



Berikut adalah tampilan aplikasi ketika user berhasil melakukan registrasi dan login. Ketika user berhasil login maka user langsung memasuki menu event yang dimana di halaman tersebut menampilkan event-event yang langsung dapat dipesan.



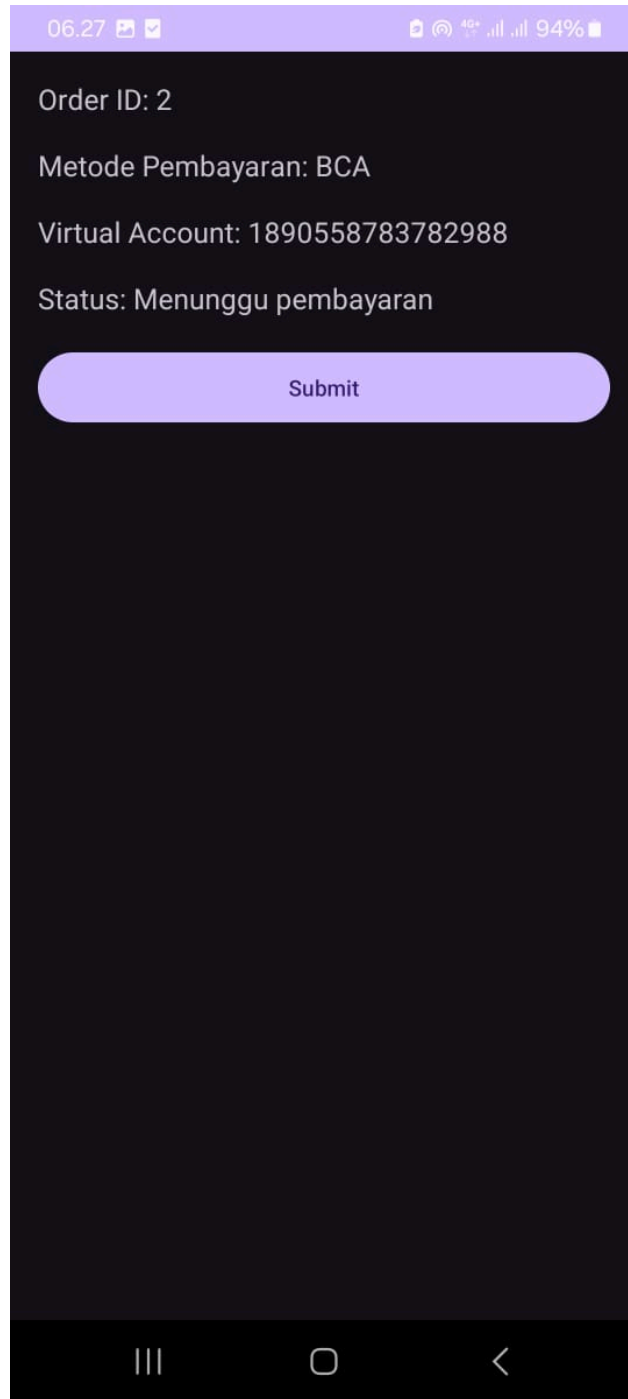
Berikut merupakan tampilan halaman jika user hendak ingin melakukan pemesanan tiket dimana di halaman tersebut terdapat kategori tiket dalam bentuk pilihan yang user harus memilih kategori tiket apa yang hendak ingin dibeli dan berapa jumlahnya. Selain itu, di halaman tersebut terdapat tombol Add yang dimana berfungsi untuk menambah kolom pengisian tiket. Jika sudah user dapat melakukan submit dan user akan berpindah halaman ke halaman yang akan menampilkan total harga dari pesanan user dan user diwajibkan untuk memilih metode pembayaran yang akan digunakan oleh user.

The image displays two screenshots of a mobile application interface for ticket booking.

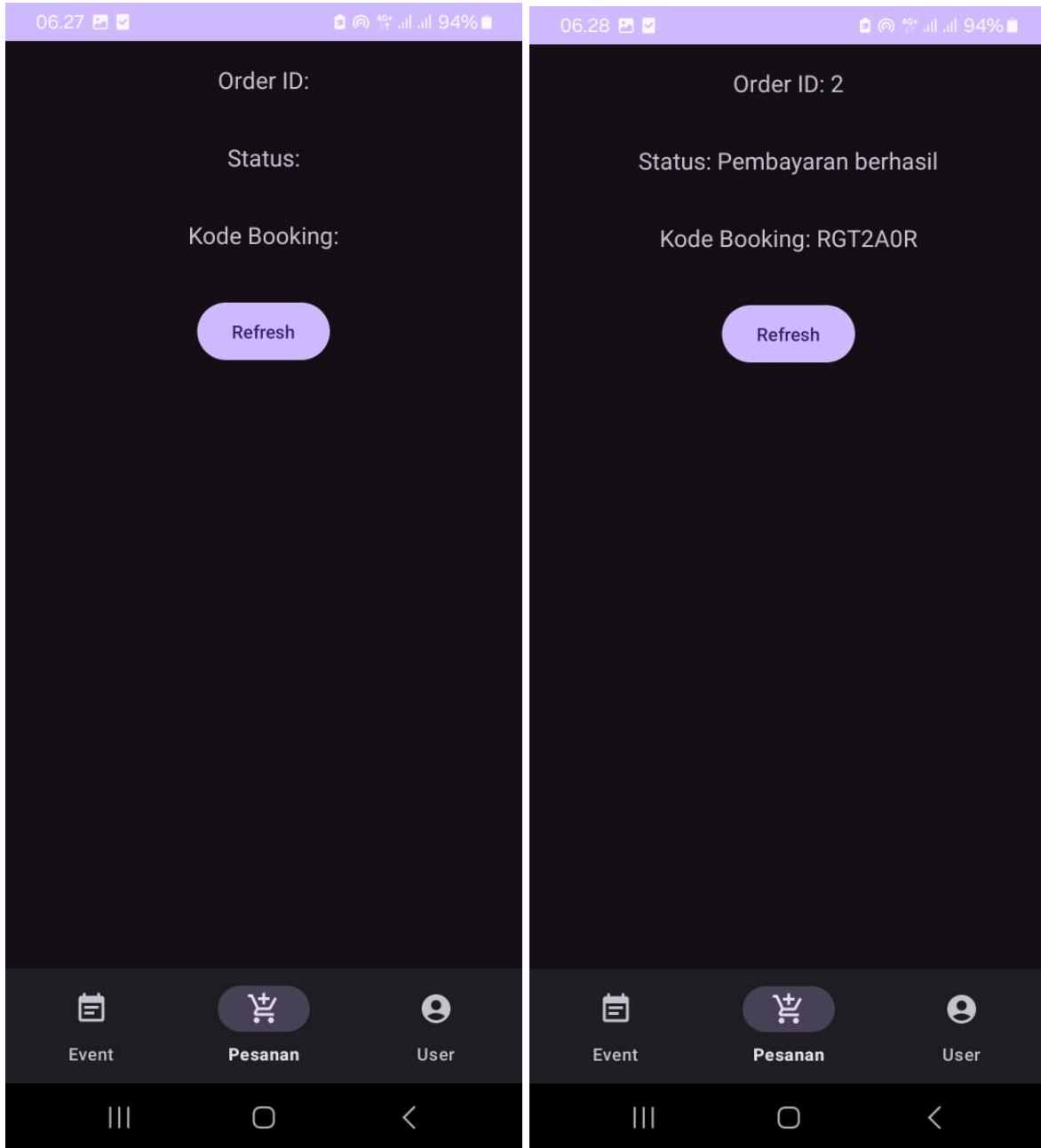
Left Screenshot: This screen shows a form for selecting tickets. It features two rows of selection options. Each row has radio buttons for "VIP" and "Regular", a text input field for the quantity, and an "Add" button. The first row has "3" entered in the quantity field, and the second row has "4" entered. Below the input fields are "Add" and "Submit" buttons.

Right Screenshot: This screen displays the total price and payment options. At the top, it shows "Total Harga: 5800". Below this, there are radio buttons for "BCA" and "BNI". A large "Submit" button is positioned at the bottom of the form.

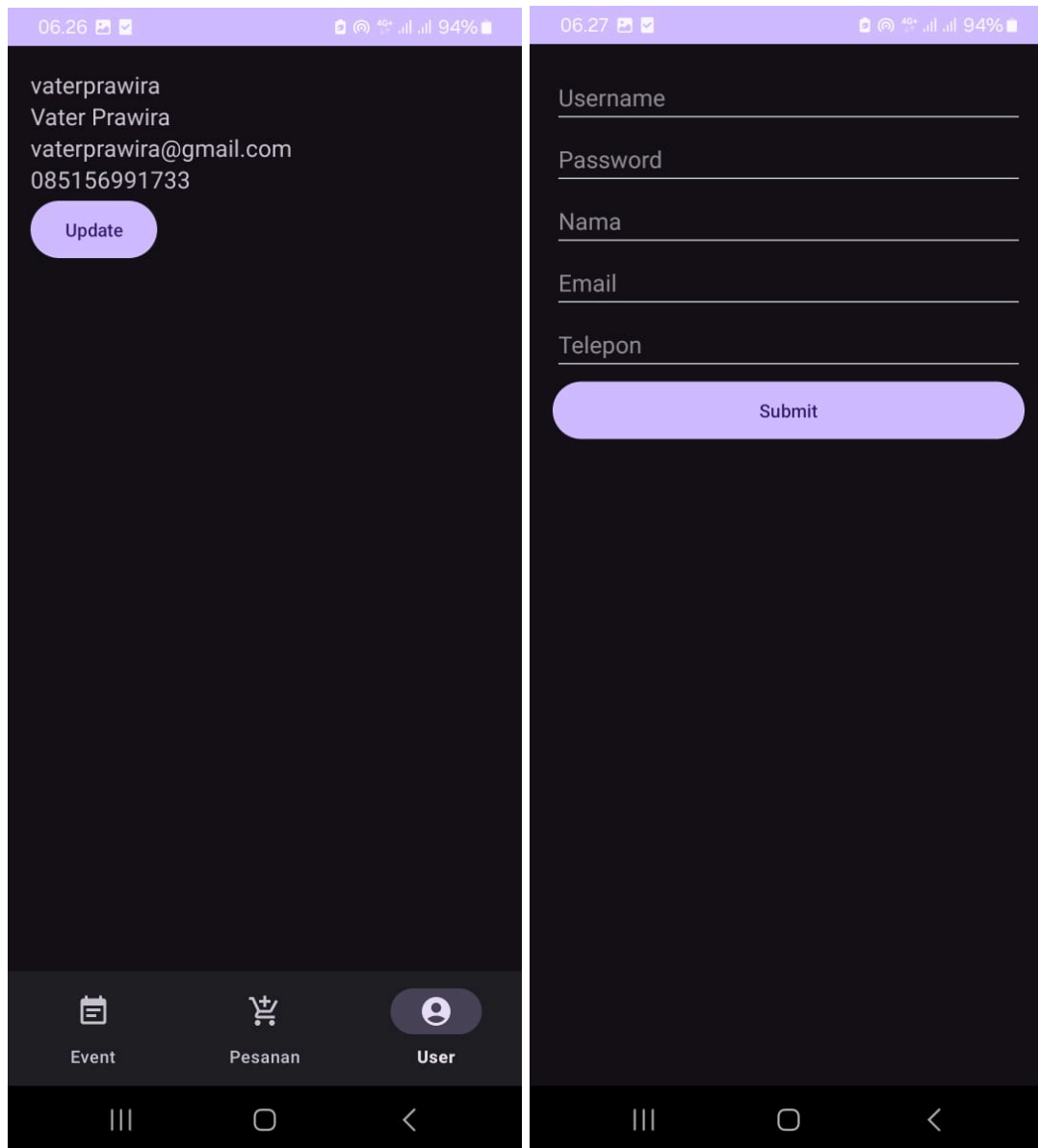
Setelah user melakukan submit terhadap metode pembayaran yang telah user pilih, maka user akan berpindah ke halaman yang akan menampilkan detail pembayaran dimana terdapat informasi mengenai metode pembayaran yang dipilih dan virtual account yang dapat digunakan user untuk melakukan pembayaran.



Setelah user melakukan pembayaran, user dapat ke menu pesanan untuk melihat apakah pembayarannya telah terkonfirmasi oleh sistem atau belum. Jika belum maka tampilannya akan seperti gambar yang kiri, jika sudah terkonfirmasi maka tampilannya akan seperti gambar kanan yang dimana akan menampilkan order id, status, dan kode booking yang akan digunakan.



Selain itu pada menu user, user dapat melihat data diri user mengenai username, nama, email, dan no telepon. Jika user ingin mengupdate data diri user maka user bisa mengklik tombol update untuk ke halaman update data diri dan user bisa mengisi data diri yang baru.



Penjelasan source code

LoginActivity.java

Kelas `LoginActivity` dalam aplikasi Android ini menangani login pengguna. Ketika aplikasi dimulai, `onCreate` diinisialisasi untuk menghubungkan komponen UI ke variabel yang sesuai menggunakan `findViewById`. `loginButton` memiliki `OnClickListener` yang memicu `LoginTask` untuk melakukan proses login secara asinkron. `registerButton` membuka aktivitas registrasi `RegisterActivity`.

Kelas `LoginTask`, yang merupakan kelas internal `AsyncTask`, mengelola koneksi HTTP POST ke server dengan URL "http://192.168.68.46:8000/api/pat/login". `doInBackground` mengirimkan data login (username dan password) yang di-encode menggunakan metode `getPostDataString`. Jika respons HTTP adalah 200 (HTTP_OK), respons JSON dibaca dan dikonversi ke string. `onPostExecute` memproses respons JSON. Jika respons berisi "message", pengguna dianggap berhasil login, sebuah `Toast` menampilkan pesan, dan aplikasi berpindah ke `MainActivity` sambil membawa `registerId` dari JSON. Jika ada "error", pesan error ditampilkan. Jika login gagal tanpa pesan error yang jelas, pesan umum ditampilkan.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;

public class LoginActivity extends AppCompatActivity {

    private EditText usernameEditText;
```

```

private EditText passwordEditText;
private Button loginButton;
private Button registerButton;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    usernameEditText = findViewById(R.id.username);
    passwordEditText = findViewById(R.id.password);
    loginButton = findViewById(R.id.loginButton);
    registerButton = findViewById(R.id.registerButton);

    loginButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            new LoginTask().execute(
                usernameEditText.getText().toString().trim(),
                passwordEditText.getText().toString().trim()
            );
        }
    });

    registerButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Buka RegisterActivity
            Intent intent = new Intent(LoginActivity.this,
RegisterActivity.class);
            startActivity(intent);
        }
    });
}

private class LoginTask extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {
        String username = params[0];
        String password = params[1];
        String response = "";

        try {
            URL url = new URL("http://192.168.68.46:8000/api/pat/login");
            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
            connection.setRequestMethod("POST");

```



```

        connection.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
        connection.setDoOutput(true);

        Map<String, String> postData = new HashMap<>();
        postData.put("username", username);
        postData.put("pass", password);

        OutputStream os = connection.getOutputStream();
        BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(os, "UTF-8"));
        writer.write(getPostDataString(postData));
        writer.flush();
        writer.close();
        os.close();

        int responseCode = connection.getResponseCode();
        if (responseCode == HttpURLConnection.HTTP_OK) {
            BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            String inputLine;
            StringBuilder sb = new StringBuilder();

            while ((inputLine = in.readLine()) != null) {
                sb.append(inputLine);
            }

            in.close();
            response = sb.toString();
            Log.d("LoginActivity", String.valueOf(responseCode));
            Log.d("LoginActivity", response);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return response;
}

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);

    try {
        JSONObject jsonObject = new JSONObject(result);
        if (jsonObject.has("message")) {
            Toast.makeText(LoginActivity.this,
jsonObject.getString("message"), Toast.LENGTH_SHORT).show();

```

```

        // Pass the registerId to MainActivity
        Intent intent = new Intent(LoginActivity.this,
MainActivity.class);
        intent.putExtra("registerId",
jsonObject.getInt("registerId"));
        startActivity(intent);
        finish();
    } else if (jsonObject.has("error")) {
        Toast.makeText(LoginActivity.this,
jsonObject.getString("error"), Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(LoginActivity.this, "Login failed. Please try
again.", Toast.LENGTH_SHORT).show();
    }
} catch (JSONException e) {
    e.printStackTrace();
    Toast.makeText(LoginActivity.this, "An error occurred. Please
try again.", Toast.LENGTH_SHORT).show();
}
}

private String getPostDataString(Map<String, String> params) throws
Exception {
    StringBuilder result = new StringBuilder();
    boolean first = true;

    for (Map.Entry<String, String> entry : params.entrySet()) {
        if (first)
            first = false;
        else
            result.append("&");

        result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
    }

    return result.toString();
}
}
}

```

RegisterActivity.java

Kelas `RegisterActivity` dalam aplikasi Android ini menangani pendaftaran pengguna. Ketika aplikasi dimulai, metode `onCreate` diinisialisasi untuk menghubungkan komponen UI ke variabel yang sesuai menggunakan `findViewById`. `registerButton` memiliki `OnClickListener` yang memicu `RegisterTask` untuk melakukan proses pendaftaran secara asinkron. `loginButton` membuka aktivitas login `LoginActivity`.

Kelas `RegisterTask`, yang merupakan kelas internal `AsyncTask`, mengelola koneksi HTTP POST ke server dengan URL "http://192.168.68.46:8000/api/pat/register". `doInBackground` mengirimkan data pendaftaran (username, password, nama, email, dan telepon) yang di-encode menggunakan metode `getPostDataString`. Jika respons HTTP adalah 200 (HTTP_OK), respons JSON dibaca dan dikonversi ke string. `onPostExecute` memproses respons JSON. Jika respons berisi "status" dengan nilai 200, pengguna dianggap berhasil mendaftar, sebuah `Toast` menampilkan pesan sukses, dan aplikasi berpindah ke `LoginActivity`. Jika ada "error", pesan error ditampilkan.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;

public class RegisterActivity extends AppCompatActivity {

    private EditText usernameEditText;
    private EditText passwordEditText;
    private EditText namaEditText;
    private EditText emailEditText;
    private EditText teleponEditText;
    private Button registerButton;
    private Button loginButton;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);

    usernameEditText = findViewById(R.id.username);
    passwordEditText = findViewById(R.id.password);
    namaEditText = findViewById(R.id.nama);
    emailEditText = findViewById(R.id.email);
    teleponEditText = findViewById(R.id.telepon);
    registerButton = findViewById(R.id.registerButton);
    loginButton = findViewById(R.id.loginButton);

    registerButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            new RegisterTask().execute(
                usernameEditText.getText().toString().trim(),
                passwordEditText.getText().toString().trim(),
                namaEditText.getText().toString().trim(),
                emailEditText.getText().toString().trim(),
                teleponEditText.getText().toString().trim()
            );
        }
    });

    loginButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Kembali ke LoginActivity
            Intent intent = new Intent(RegisterActivity.this,
LoginActivity.class);
            startActivity(intent);
        }
    });
}

private class RegisterTask extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {
        String username = params[0];
        String password = params[1];
        String nama = params[2];
        String email = params[3];
        String telepon = params[4];
        String response = "";
    }
}

```

```

        try {
            URL url = new URL("http://192.168.68.46:8000/api/pat/register");
            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
            connection.setRequestMethod("POST");
            connection.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
            connection.setDoOutput(true);

            Map<String, String> postData = new HashMap<>();
            postData.put("username", username);
            postData.put("pass", password);
            postData.put("nama", nama);
            postData.put("email", email);
            postData.put("telepon", telepon);

            OutputStream os = connection.getOutputStream();
            BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(os, "UTF-8"));
            writer.write(getPostDataString(postData));
            writer.flush();
            writer.close();
            os.close();

            int responseCode = connection.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) {
                BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
                String inputLine;
                StringBuilder sb = new StringBuilder();

                while ((inputLine = in.readLine()) != null) {
                    sb.append(inputLine);
                }

                in.close();
                response = sb.toString();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return response;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
    }

```

```

        try {
            JSONObject jsonObject = new JSONObject(result);
            if (jsonObject.has("status") && jsonObject.getInt("status") ==
200) {
                Toast.makeText(RegisterActivity.this, "Registration
successful", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(RegisterActivity.this,
LoginActivity.class);
                startActivity(intent);
                finish();
            } else if (jsonObject.has("error")) {
                Toast.makeText(RegisterActivity.this,
jsonObject.getString("error"), Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    private String getPostDataString(Map<String, String> params) throws
Exception {
        StringBuilder result = new StringBuilder();
        boolean first = true;

        for (Map.Entry<String, String> entry : params.entrySet()) {
            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
        }

        return result.toString();
    }
}

```

MainActivity.java

Kelas `MainActivity` dalam aplikasi Android ini bertindak sebagai aktivitas utama yang menampung beberapa fragmen dan mengelola navigasi menggunakan `BottomNavigationView`. Ketika aktivitas dibuat, layout `activity_main` dimuat, dan `registerId` diperoleh dari `Intent` yang dikirim dari aktivitas sebelumnya. `BottomNavigationView` digunakan untuk menampilkan menu navigasi di bagian bawah layar. Listener untuk item yang dipilih pada `BottomNavigationView` ditentukan, dan berdasarkan item menu yang dipilih, fragmen yang sesuai dimuat: `EventFragment`, `PesananFragment`, atau `UserFragment`. Jika `PesananFragment` dipilih dan `orderId` tersedia di `Intent`, data `orderId` diteruskan ke fragmen melalui bundle. Demikian pula, `registerId` diteruskan ke `UserFragment` jika tersedia. Secara keseluruhan, kelas ini mengelola navigasi antar fragmen dalam aplikasi dengan menggunakan `BottomNavigationView` dan memastikan bahwa data yang relevan diteruskan ke fragmen yang tepat sesuai kebutuhan.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {

    private int registerId = -1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Get the registerId from the intent
        Intent intent = getIntent();
        if (intent != null && intent.hasExtra("registerId")) {
            registerId = intent.getIntExtra("registerId", -1);
        }

        BottomNavigationView bottomNavigationView =
            findViewById(R.id.bottom_navigation);
        bottomNavigationView.setOnNavigationItemSelectedListener(new
            BottomNavigationView.OnNavigationItemSelectedListener() {
```

```

@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
    Fragment selectedFragment = null;
    int itemId = item.getItemId();

    if (itemId == R.id.nav_event) {
        selectedFragment = new EventFragment();
    } else if (itemId == R.id.nav_pesanan) {
        selectedFragment = new PesananFragment();
        if (intent != null && intent.hasExtra("orderId")) {
            Bundle bundle = new Bundle();
            bundle.putInt("orderId", intent.getIntExtra("orderId",
-1));

            selectedFragment.setArguments(bundle);
        }
    } else if (itemId == R.id.nav_user) {
        selectedFragment = new UserFragment();
        // Pass the registerId to the UserFragment
        if (registerId != -1) {
            Bundle bundle = new Bundle();
            bundle.putInt("registerId", registerId);
            selectedFragment.setArguments(bundle);
        }
    }

    if (selectedFragment != null) {
        loadFragment(selectedFragment);
    }

    return true;
}

});

// Load the default fragment
if (savedInstanceState == null) {
    bottomNavigationView.setSelectedItemId(R.id.nav_event); // Change to
the default selected menu
}

}

private void loadFragment(Fragment fragment) {
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
}
}

```


EventFragment.java

Kelas `EventFragment` dalam aplikasi Android ini berfungsi untuk menampilkan daftar acara dalam bentuk daftar menggunakan `RecyclerView`. Fragmen ini menggunakan `EventAdapter` untuk menghubungkan data acara dengan tampilan daftar dan memungkinkan pengguna untuk memesan tiket acara tertentu dengan mengklik tombol pesan. `RecyclerView` digunakan untuk menampilkan daftar acara dalam bentuk gulir vertikal, sedangkan `EventAdapter` berfungsi sebagai adapter yang menghubungkan data acara ke tampilan daftar dan menangani klik tombol pesan. Daftar acara disimpan dalam `List<Event> eventList`. Pada metode `onCreateView`, layout fragmen (`fragment_event.xml`) diinflasi dan tampilan yang dihasilkan dikembalikan. Setelah itu, metode `onViewCreated` dipanggil untuk menginisialisasi `RecyclerView`, membuat instance `EventAdapter`, dan mengatur layout manager serta adapter untuk `RecyclerView`, kemudian menjalankan `FetchEventTask` untuk mengambil data acara dari API. Kelas `FetchEventTask` memiliki metode `doInBackground` yang melakukan operasi jaringan untuk mengambil data acara dari URL API dalam bentuk JSON dan mengembalikannya sebagai string. Setelah data diambil, metode `onPostExecute` akan mem-parsing data JSON untuk mendapatkan detail acara dan menambahkannya ke `eventList`, kemudian `EventAdapter` diberitahu untuk memperbarui tampilan. Jika terjadi kesalahan dalam pengambilan atau parsing data, pesan kesalahan akan ditampilkan. Secara keseluruhan, ketika fragmen dibuat, layout diinflasi, `RecyclerView` diinisialisasi, dan `EventAdapter` diatur dengan daftar acara serta listener klik tombol pesan.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class EventFragment extends Fragment {
```

```

private RecyclerView recyclerView;
private EventAdapter eventAdapter;
private List<Event> eventList;

public EventFragment() {
    // Required empty public constructor
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_event, container, false);
}

@Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        recyclerView = view.findViewById(R.id.recycler_view);
        recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
        eventList = new ArrayList<>();
        eventAdapter = new EventAdapter(eventList, new
EventAdapter.OnOrderClickListener() {
            @Override
            public void onOrderClick(Event event) {
                Intent intent = new Intent(getActivity(), OrderActivity.class);
                intent.putExtra("eventId", event.getId());
                startActivity(intent);
            }
        });
        recyclerView.setAdapter(eventAdapter);
        new FetchEventTask().execute();
    }

private class FetchEventTask extends AsyncTask<Void, Void, String> {
    @Override
    protected String doInBackground(Void... voids) {
        try {
            URL url = new
URL("http://192.168.68.46:8000/api/pat/event/mobile");
            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
            conn.setRequestMethod("GET");
            BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            String inputLine;
            StringBuilder content = new StringBuilder();
            while ((inputLine = in.readLine()) != null) {

```

```

        content.append(inputLine);
    }
    in.close();
    conn.disconnect();
    return content.toString();
} catch (Exception e) {
    e.printStackTrace();
    return null;
}
}

@Override
protected void onPostExecute(String result) {
    if (result != null) {
        try {
            JSONObject jsonObject = new JSONObject(result);
            JSONArray events =
jsonObject.getJSONObject("response").getJSONArray("events");
            for (int i = 0; i < events.length(); i++) {
                JSONObject eventObj = events.getJSONObject(i);
                int id = eventObj.getInt("id");
                String namaKonser = eventObj.getString("nama_konser");
                String artist = eventObj.getString("artist");
                int harga = eventObj.getInt("harga");
                String lokasi = eventObj.getString("lokasi");
                String tanggal = eventObj.getString("tanggal");
                eventList.add(new Event(id, namaKonser, artist, harga,
lokasi, tanggal));
            }
            eventAdapter.notifyDataSetChanged();
        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(getActivity(), "Error parsing data",
Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(getActivity(), "Failed to fetch data",
Toast.LENGTH_SHORT).show();
    }
}
}
}

```

OrderActivity.java

Kelas `OrderActivity` dalam aplikasi Android ini berfungsi untuk menangani pemesanan tiket acara oleh pengguna. Aktivitas ini menyediakan antarmuka bagi pengguna untuk memilih kategori tiket, memasukkan jumlah tiket yang diinginkan, dan mengirimkan pesanan mereka. Dalam metode `onCreate`, beberapa elemen UI seperti `LinearLayout` untuk menampung input tiket, tombol untuk menambah tiket, dan tombol untuk mengirimkan pesanan diinisialisasi. Kategori tiket diambil dari API menggunakan kelas `FetchEventDataTask` yang merupakan subkelas `AsyncTask`. Data kategori tiket diambil dari API dalam bentuk JSON dan diolah untuk menambah tombol radio ke grup tombol radio secara dinamis berdasarkan kategori yang tersedia.

Metode `addTicketField` digunakan untuk menambahkan tampilan input tiket baru ke kontainer tiket setiap kali tombol "Add" ditekan. Setiap tampilan input tiket terdiri dari `EditText` untuk jumlah tiket dan `RadioGroup` untuk memilih kategori tiket. Metode `submitOrder` dikaitkan dengan tombol "Submit" dan bertugas mengumpulkan semua input pengguna, memverifikasi bahwa semua bidang telah diisi, dan kemudian mengirimkan pesanan menggunakan kelas `SubmitOrderTask`. Kelas `SubmitOrderTask` merupakan subkelas `AsyncTask` yang menangani pengiriman data pesanan ke server melalui permintaan HTTP POST. Data pesanan dikumpulkan dari input pengguna, diubah menjadi format JSON, dan dikirim ke server. Jika pesanan berhasil dikirim, pengguna diarahkan ke aktivitas pembayaran `PaymentActivity` dengan ID pesanan yang diterima dari respons server.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;
```

```

public class OrderActivity extends AppCompatActivity {

    private LinearLayout ticketContainer;
    private Button addButton, submitButton;
    private List<EditText> jumlahFields;
    private List<RadioGroup> kategoriFields;
    private List<String> kategoriList;
    private int eventId;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order);

        ticketContainer = findViewById(R.id.ticket_container);
        addButton = findViewById(R.id.add_button);
        submitButton = findViewById(R.id.submit_button);
        jumlahFields = new ArrayList<>();
        kategoriFields = new ArrayList<>();
        kategoriList = new ArrayList<>();
        eventId = getIntent().getIntExtra("eventId", -1);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addTicketField();
            }
        });

        submitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                submitOrder();
            }
        });

        // Fetch event data
        new FetchEventDataTask().execute();
    }

    private void addTicketField() {
        View ticketView = getLayoutInflater().inflate(R.layout.item_ticket,
ticketContainer, false);
        EditText jumlahField = ticketView.findViewById(R.id.jumlah_field);
        RadioGroup kategoriGroup = ticketView.findViewById(R.id.kategori_group);

        // Dynamically add radio buttons based on kategoriList
        for (String kategori : kategoriList) {
            RadioButton radioButton = new RadioButton(this);

```

```

        radioButton.setText(kategori);
        radioButton.setTag(kategori);
        kategoriGroup.addView(radioButton);
    }

    jumlahFields.add(jumlahField);
    kategoriFields.add(kategoriGroup);
    ticketContainer.addView(ticketView);
}

private void submitOrder() {
    List<TicketOrder> ticketOrders = new ArrayList<>();
    for (int i = 0; i < jumlahFields.size(); i++) {
        String jumlahStr = jumlahFields.get(i).getText().toString();
        int selectedKategoriId =
kategoriFields.get(i).getCheckedRadioButtonId();
        if (selectedKategoriId == -1 || jumlahStr.isEmpty()) {
            Toast.makeText(this, "Please fill out all fields",
Toast.LENGTH_SHORT).show();
            return;
        }

        int jumlah = Integer.parseInt(jumlahStr);
        String kategori =
findViewById(selectedKategoriId).getTag().toString();
        ticketOrders.add(new TicketOrder(kategori, jumlah));
    }

    new SubmitOrderTask(ticketOrders).execute();
}

private class FetchEventDataTask extends AsyncTask<Void, Void, String> {
    @Override
    protected String doInBackground(Void... voids) {
        try {
            URL url = new
URL("http://192.168.68.46:8000/api/pat/event/mobile/" + eventId);
            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
            conn.setRequestMethod("GET");
            BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            String inputLine;
            StringBuilder content = new StringBuilder();
            while ((inputLine = in.readLine()) != null) {
                content.append(inputLine);
            }
            in.close();
            conn.disconnect();

```

```

        return content.toString();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

@Override
protected void onPostExecute(String result) {
    if (result != null) {
        try {
            JSONObject jsonObject = new JSONObject(result);
            JSONArray kategoriArray =
jsonObject.getJSONObject("response").getJSONArray("events").getJSONObject(0).ge
tJSONArray("kategori");

            for (int i = 0; i < kategoriArray.length(); i++) {
                kategoriList.add(kategoriArray.getString(i));
            }

            addTicketField(); // Add initial ticket field after fetching
categories

        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(OrderActivity.this, "Error parsing event
data", Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(OrderActivity.this, "Failed to fetch event data",
Toast.LENGTH_SHORT).show();
    }
}

}

private class SubmitOrderTask extends AsyncTask<Void, Void, String> {
    private List<TicketOrder> ticketOrders;

    public SubmitOrderTask(List<TicketOrder> ticketOrders) {
        this.ticketOrders = ticketOrders;
    }

    @Override
    protected String doInBackground(Void... voids) {
        try {
            URL url = new URL("http://192.168.68.46:8000/api/pat/order");
            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
            conn.setRequestMethod("POST");

```

```

        conn.setRequestProperty("Content-Type", "application/json");
        conn.setDoOutput(true);

        JSONObject jsonBody = new JSONObject();
        jsonBody.put("eventId", eventId);
        JSONArray kategoriTiketArray = new JSONArray();
        for (TicketOrder order : ticketOrders) {
            JSONObject orderObject = new JSONObject();
            orderObject.put("kategori", order.getKategori());
            orderObject.put("jumlah", order.getJumlah());
            kategoriTiketArray.put(orderObject);
        }
        jsonBody.put("kategori_tiket", kategoriTiketArray);

        BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(conn.getOutputStream()));
        writer.write(jsonBody.toString());
        writer.flush();
        writer.close();

        int responseCode = conn.getResponseCode();
        Log.d("SubmitOrderTask", "Response code: " + responseCode);
        if (responseCode == HttpURLConnection.HTTP_OK) {
            BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            StringBuilder response = new StringBuilder();
            String line;
            while ((line = in.readLine()) != null) {
                response.append(line);
            }
            in.close();
            Log.d("SubmitOrderTask", "Response: " +
response.toString());
            return response.toString();
        } else {
            BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getErrorStream()));
            StringBuilder errorContent = new StringBuilder();
            String line;
            while ((line = in.readLine()) != null) {
                errorContent.append(line);
            }
            in.close();
            Log.e("SubmitOrderTask", "Error response: " +
errorContent.toString());
            return "Failed to place order: " + errorContent.toString();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```



```

        Log.e("SubmitOrderTask", "Exception: " + e.getMessage());
        return "Error placing order: " + e.getMessage();
    }
}

@Override
protected void onPostExecute(String result) {
    try {
        Log.d("SubmitOrderTask", "Result: " + result);
        JSONObject jsonObject = new JSONObject(result);

        if (jsonObject.getInt("status") == 200) {
            int orderId = jsonObject.getInt("order_id");
            Intent intent = new Intent(OrderActivity.this,
PaymentActivity.class);
            intent.putExtra("orderId", orderId);
            startActivity(intent);
            finish();
        } else {
            String error = jsonObject.optString("error", "Order
failed");
            Toast.makeText(OrderActivity.this, "Order failed: " + error,
Toast.LENGTH_SHORT).show();
        }
    } catch (Exception e) {
        e.printStackTrace();
        Toast.makeText(OrderActivity.this, "Error parsing order
response: " + e.getMessage(), Toast.LENGTH_SHORT).show();
    }
}

private class TicketOrder {
    private String kategori;
    private int jumlah;

    public TicketOrder(String kategori, int jumlah) {
        this.kategori = kategori;
        this.jumlah = jumlah;
    }

    public String getKategori() { return kategori; }
    public int getJumlah() { return jumlah; }
}
}

```

PaymentActivity.java

Kelas `PaymentActivity` dalam aplikasi Android ini berfungsi untuk menangani proses pembayaran pesanan tiket acara oleh pengguna. Dalam metode `onCreate`, elemen-elemen UI seperti `RadioGroup` untuk metode pembayaran, `TextView` untuk menampilkan total harga, dan tombol "Submit" diinisialisasi. ID pesanan diambil dari `Intent` yang mengirimkan pengguna ke aktivitas ini. Data detail pesanan dan metode pembayaran diambil dari API menggunakan kelas `FetchOrderDetailsTask`, yang merupakan subkelas `AsyncTask`.

Setelah data pesanan berhasil diambil, `TextView` total harga diperbarui dan `RadioGroup` diisi dengan opsi metode pembayaran yang tersedia. Metode `submitPayment` dikaitkan dengan tombol "Submit" dan bertugas untuk mengirimkan metode pembayaran yang dipilih oleh pengguna ke server. Kelas `SubmitPaymentTask` merupakan subkelas `AsyncTask` yang menangani pengiriman data pembayaran ke server melalui permintaan HTTP POST. Data pembayaran dikumpulkan dari input pengguna, diubah menjadi format JSON, dan dikirim ke server. Jika pembayaran berhasil diproses, pengguna diarahkan ke aktivitas `PaymentInfoActivity` dengan ID pembayaran yang diterima dari respons server.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

public class PaymentActivity extends AppCompatActivity {

    private RadioGroup paymentMethodGroup;
    private TextView totalPriceTextView;
    private Button submitButton;
    private int orderId;
    private String[] paymentMethods;

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_payment);

    paymentMethodGroup = findViewById(R.id.payment_method_group);
    totalPriceTextView = findViewById(R.id.total_price);
    submitButton = findViewById(R.id.submit_button);

    orderId = getIntent().getIntExtra("orderId", -1);

    new FetchOrderDetailsTask().execute(orderId);

    submitButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            submitPayment();
        }
    });
}

private void submitPayment() {
    int selectedMethodId = paymentMethodGroup.getCheckedRadioButtonId();
    if (selectedMethodId == -1) {
        Toast.makeText(this, "Please select a payment method",
Toast.LENGTH_SHORT).show();
        return;
    }

    String selectedMethod =
findViewById(selectedMethodId).getTag().toString();
    new SubmitPaymentTask().execute(orderId, selectedMethod);
}

private class FetchOrderDetailsTask extends AsyncTask<Integer, Void, String>
{
    @Override
    protected String doInBackground(Integer... params) {
        int orderId = params[0];
        try {
            URL url = new URL("http://192.168.68.46:8000/api/pat/order/" +
orderId);
            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
            conn.setRequestMethod("GET");
            BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            String inputLine;
            StringBuilder content = new StringBuilder();
            while ((inputLine = in.readLine()) != null) {

```

```

        content.append(inputLine);
    }
    in.close();
    conn.disconnect();
    return content.toString();
} catch (Exception e) {
    e.printStackTrace();
    return null;
}
}

@Override
protected void onPostExecute(String result) {
    if (result != null) {
        try {
            JSONObject jsonObject = new JSONObject(result);
            JSONObject response = jsonObject.getJSONObject("response");
            int totalHarga = response.getInt("total_harga");
            JSONArray banksArray = response.getJSONArray("bank");

            totalPriceTextView.setText("Total Harga: " + totalHarga);

            paymentMethods = new String[banksArray.length()];
            for (int i = 0; i < banksArray.length(); i++) {
                String bank = banksArray.getString(i);
                paymentMethods[i] = bank;

                RadioButton radioButton = new
RadioButton(PaymentActivity.this);
                radioButton.setText(bank);
                radioButton.setTag(bank);
                paymentMethodGroup.addView(radioButton);
            }
        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(PaymentActivity.this, "Error parsing data",
Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(PaymentActivity.this, "Failed to fetch data",
Toast.LENGTH_SHORT).show();
    }
}

}

private class SubmitPaymentTask extends AsyncTask<Object, Void, String> {
    @Override
    protected String doInBackground(Object... params) {
        int orderId = (int) params[0];

```

```

        String metodePembayaran = (String) params[1];
        try {
            URL url = new URL("http://192.168.68.46:8000/api/pat/payment");
            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type", "application/json");
            conn.setDoOutput(true);

            JSONObject jsonBody = new JSONObject();
            jsonBody.put("orderId", orderId);
            jsonBody.put("metode_pembayaran", metodePembayaran);

            BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(conn.getOutputStream()));
            writer.write(jsonBody.toString());
            writer.flush();
            writer.close();

            int responseCode = conn.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) {
                BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                String inputLine;
                StringBuilder content = new StringBuilder();
                while ((inputLine = in.readLine()) != null) {
                    content.append(inputLine);
                }
                in.close();
                return content.toString();
            } else {
                BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getErrorStream()));
                String inputLine;
                StringBuilder content = new StringBuilder();
                while ((inputLine = in.readLine()) != null) {
                    content.append(inputLine);
                }
                in.close();
                return content.toString();
            }
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    @Override
    protected void onPostExecute(String result) {

```

```

        if (result != null) {
            try {
                Log.d("PaymentActivity", "Response: " + result);
                JSONObject jsonObject = new JSONObject(result);
                if (jsonObject.has("payment_id")) {
                    int paymentId = jsonObject.getInt("payment_id");
                    Intent intent = new Intent(PaymentActivity.this,
PaymentInfoActivity.class);
                    intent.putExtra("paymentId", paymentId);
                    startActivity(intent);
                } else {
                    throw new Exception("Unexpected response: " + result);
                }
            } catch (Exception e) {
                e.printStackTrace();
                Toast.makeText(PaymentActivity.this, "Error parsing
response", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(PaymentActivity.this, "Failed to submit payment",
Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

PesananActivity.java

Kelas `PesananActivity` dalam aplikasi Android ini berfungsi untuk menampilkan detail pesanan tiket acara kepada pengguna. Pada metode `onCreate`, elemen-elemen UI seperti `TextView` untuk menampilkan ID pesanan, status, kode booking, dan tombol "View Details" diinisialisasi. ID pesanan diambil dari `Intent` yang mengirimkan pengguna ke aktivitas ini. Jika ID pesanan valid, detail pesanan diambil dari API menggunakan kelas `FetchOrderDetailsTask`, yang merupakan subkelas `AsyncTask`. Setelah data pesanan berhasil diambil, `TextView` diperbarui dengan informasi yang diterima, seperti ID pesanan, status, dan kode booking. Tombol "View Details" memungkinkan pengguna untuk melihat detail lebih lanjut dari pesanan tersebut di dalam `MainActivity`, khususnya di dalam `PesananFragment`.

Kelas `FetchOrderDetailsTask` bertugas mengambil data pesanan dari API melalui permintaan HTTP GET. Data yang diterima dari API kemudian diuraikan menjadi objek JSON, dan nilai-nilai yang relevan diekstrak untuk diperbarui di UI. Secara keseluruhan, kelas `PesananActivity` menyediakan antarmuka untuk menampilkan detail pesanan pengguna dan memungkinkan pengguna untuk menavigasi ke halaman yang lebih rinci mengenai pesanan mereka.

```
package com.example.concertapp;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class PesananActivity extends AppCompatActivity {

    private static final String TAG = "PesananActivity";
    private TextView orderIdTextView;
    private TextView statusTextView;
    private TextView kodeBookingTextView;
    private Button viewDetailsButton;
    private int orderId;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pesanan);

        orderIdTextView = findViewById(R.id.orderId);
        statusTextView = findViewById(R.id.status);
        kodeBookingTextView = findViewById(R.id.kodeBooking);
        viewDetailsButton = findViewById(R.id.viewDetails);

        orderId = getIntent().getIntExtra("orderId", -1);

        if (orderId != -1) {
            new FetchOrderDetailsTask().execute(orderId);
        } else {
            Log.e(TAG, "orderId is invalid");
            Toast.makeText(this, "Invalid orderId", Toast.LENGTH_SHORT).show();
        }
    }
}
```

```

        viewDetailsButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PesananActivity.this,
MainActivity.class);
                intent.putExtra("fragment", "PesananFragment");
                intent.putExtra("orderId", orderId);
                startActivity(intent);
            }
        });
    }

    private class FetchOrderDetailsTask extends AsyncTask<Integer, Void, String>
    {
        @Override
        protected String doInBackground(Integer... params) {
            int orderId = params[0];
            try {
                URL url = new URL("http://192.168.68.46:8000/api/pat/order/" +
orderId);
                HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
                conn.setRequestMethod("GET");
                BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                String inputLine;
                StringBuilder content = new StringBuilder();
                while ((inputLine = in.readLine()) != null) {
                    content.append(inputLine);
                }
                in.close();
                conn.disconnect();
                return content.toString();
            } catch (Exception e) {
                Log.e(TAG, "Error fetching order details: " + e.getMessage(),
e);
                return null;
            }
        }

        @Override
        protected void onPostExecute(String result) {
            if (result != null) {
                try {
                    JSONObject jsonObject = new JSONObject(result);
                    JSONObject response = jsonObject.getJSONObject("response");
                    String orderId = response.getString("orderId");
                    String status = response.getString("status");
                    String kodeBooking = response.getString("kode_booking");

```



```

        orderIdTextView.setText("Order ID: " + orderId);
        statusTextView.setText("Status: " + status);
        kodeBookingTextView.setText("Kode Booking: " + kodeBooking);
    } catch (Exception e) {
        Log.e(TAG, "Error parsing data: " + e.getMessage(), e);
        Toast.makeText(PesananActivity.this, "Error parsing data",
Toast.LENGTH_SHORT).show();
    }
    } else {
        Log.e(TAG, "Failed to fetch data");
        Toast.makeText(PesananActivity.this, "Failed to fetch data",
Toast.LENGTH_SHORT).show();
    }
    }
}
}
}

```

UserFragment.java

`UserFragment` adalah sebuah fragmen dalam aplikasi Android yang digunakan untuk menampilkan detail pengguna. Fragmen ini menampilkan informasi seperti nama pengguna, nama lengkap, email, dan nomor telepon. Pada metode `onCreateView`, elemen-elemen UI diinisialisasi, termasuk `TextView` untuk menampilkan detail pengguna dan sebuah tombol "Update" yang memungkinkan pengguna untuk memperbarui informasi mereka. ID pengguna diperoleh dari `Bundle` yang diterima saat fragmen dibuat. Jika ID pengguna valid, data pengguna diambil dari API menggunakan `AsyncTask` bernama `FetchUserDetailsTask`. `AsyncTask` ini melakukan permintaan HTTP GET ke API untuk mengambil data pengguna berdasarkan ID. Setelah data berhasil diambil, JSON response diuraikan untuk mendapatkan detail pengguna yang kemudian diperbarui di UI fragmen. Tombol "Update" mengarahkan pengguna ke `UpdateUserActivity` untuk memperbarui informasi pengguna. Secara keseluruhan, `UserFragment` menyediakan antarmuka yang sederhana dan intuitif untuk menampilkan dan memperbarui informasi pengguna dalam aplikasi Android.

```

package com.example.concertapp;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import androidx.annotation.NonNull;

```

```

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class UserFragment extends Fragment {

    private TextView usernameTextView, nameTextView, emailTextView,
    phoneTextView;
    private Button updateButton;

    @SuppressWarnings("MissingInflatedId")
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
    ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_user, container, false);

        usernameTextView = view.findViewById(R.id.username);
        nameTextView = view.findViewById(R.id.name_text);
        emailTextView = view.findViewById(R.id.email);
        phoneTextView = view.findViewById(R.id.phone_text);
        updateButton = view.findViewById(R.id.update_button);

        updateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getActivity(),
                UpdateUserActivity.class);
                intent.putExtra("registerId",
                getArguments().getInt("registerId"));
                startActivity(intent);
            }
        });

        int userId = getArguments().getInt("registerId", -1);
        if (userId != -1) {
            new FetchUserDetailsTask().execute(userId);
        }

        return view;
    }

    private class FetchUserDetailsTask extends AsyncTask<Integer, Void,
    JSONObject> {
        @Override

```

```

        protected JSONObject doInBackground(Integer... params) {
            int userId = params[0];
            try {
                URL url = new URL("http://192.168.68.46:8000/api/pat/register/"
+ userId);
                HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
                connection.setRequestMethod("GET");

                BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
                StringBuilder result = new StringBuilder();
                String line;
                while ((line = reader.readLine()) != null) {
                    result.append(line);
                }
                reader.close();

                return new JSONObject(result.toString());
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }

        @Override
        protected void onPostExecute(JSONObject response) {
            if (response != null) {
                try {
                    JSONObject user =
response.getJSONObject("response").getJSONArray("events").getJSONObject(0);
                    usernameTextView.setText(user.getString("username"));
                    nameTextView.setText(user.getString("nama"));
                    emailTextView.setText(user.getString("email"));
                    phoneTextView.setText(user.getString("telepon"));
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

UpdateUserActivity.java

`UpdateUserActivity` adalah sebuah aktivitas dalam aplikasi Android yang memungkinkan pengguna untuk memperbarui detail pengguna mereka. Saat aktivitas ini dibuka, elemen-elemen antarmuka pengguna seperti `EditText` untuk nama pengguna, kata sandi, nama lengkap, email, dan nomor telepon diinisialisasi dari layout XML yang sesuai. Data registerId diperoleh dari intent yang memulai aktivitas ini. Pada saat tombol "Submit" ditekan, data yang dimasukkan oleh pengguna diambil dari masing-masing `EditText`, dan `AsyncTask` bernama `UpdateUserDetailsTask` dijalankan. AsyncTask ini melakukan permintaan HTTP PUT ke API yang sesuai untuk memperbarui detail pengguna berdasarkan registerId yang diberikan.

Di dalam `doInBackground` AsyncTask, objek JSON dibuat untuk mengemas data pengguna yang akan diperbarui, seperti username, password, nama, email, dan nomor telepon. Data ini kemudian dikirim ke server menggunakan output stream dari koneksi HTTP. Metode `onPostExecute` akan menangani respons dari server: jika permintaan berhasil (HTTP response code 200), aktivitas akan ditutup kembali ke layar sebelumnya. Jika gagal, pesan kesalahan dapat ditampilkan menggunakan Toast atau Snackbar.

```
package com.example.concertapp;

import android.annotation.SuppressLint;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONObject;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;

public class UpdateUserActivity extends AppCompatActivity {

    private EditText usernameEditText, passwordEditText, nameEditText,
emailEditText, phoneEditText;
    private Button submitButton;
    private int registerId;

    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_user);

        usernameEditText = findViewById(R.id.username);
        passwordEditText = findViewById(R.id.password);
        nameEditText = findViewById(R.id.name_text);
        emailEditText = findViewById(R.id.email);
```

```

phoneEditText = findViewById(R.id.phone_text);
submitButton = findViewById(R.id.submit_button);

// Get the registerId from the intent
registerId = getIntent().getIntExtra("registerId", -1);

submitButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username = usernameEditText.getText().toString();
        String password = passwordEditText.getText().toString();
        String name = nameEditText.getText().toString();
        String email = emailEditText.getText().toString();
        String phone = phoneEditText.getText().toString();

        new UpdateUserDetailsTask().execute(registerId, username,
password, name, email, phone);
    }
});
}

private class UpdateUserDetailsTask extends AsyncTask<Object, Void, Boolean>
{
    @Override
    protected Boolean doInBackground(Object... params) {
        int userId = (int) params[0];
        String username = (String) params[1];
        String password = (String) params[2];
        String name = (String) params[3];
        String email = (String) params[4];
        String phone = (String) params[5];

        try {
            URL url = new URL("http://192.168.68.46:8000/api/pat/register/"
+ userId);
            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
            connection.setRequestMethod("PUT");
            connection.setRequestProperty("Content-Type",
"application/json");
            connection.setDoOutput(true);

            JSONObject userData = new JSONObject();
            userData.put("username", username);
            userData.put("pass", password);
            userData.put("nama", name);
            userData.put("email", email);
            userData.put("telepon", phone);

```

```
        OutputStream os = connection.getOutputStream();
        os.write(userData.toString().getBytes());
        os.flush();
        os.close();

        int responseCode = connection.getResponseCode();
        return responseCode == HttpURLConnection.HTTP_OK;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

@Override
protected void onPostExecute(Boolean success) {
    if (success) {
        finish();
    } else {
        // Show an error message (Toast or Snackbar)
    }
}
}
```