

PERANCANGAN PROGRAM APLIKASI TOKO BUKU ONLINE “BUKUPEDIA”

UNIVERSITAS MULTIMEDIA NUSANTARA

Ditulis dalam rangka memenuhi penilaian mata kuliah Mobile Application Development





UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Nama Anggota Kelompok :

Agym Dimas Tian	(00000058149)
Arlynandhita Felisya Putri Wibowo	(00000061610)
Benz Martheen W Jonash	(00000059822)
Jericho Alexander Gunawan	(00000058345)
Nicholas Suwandi Wijaya	(00000057903)
Rheinald Adyatma Rizqi Wibowo	(00000058204)

HALAMAN PENGESAHAN DOKUMEN

Dokumen ini telah disetujui sebagai laporan akhir untuk pengerjaan proyek akhir program studi Sistem Informasi mata kuliah Mobile Application Development yang akan menjadi pengguna akhir dari dokumen ini.

NIM	Nama	Program	Tanda Tangan
00000058149	Agym Dimas Tian	Sistem Informasi	
00000061610	Arlynandhita Felisya Putri Wibowo	Sistem Informasi	
00000059822	Benz Martheen W Jonash	Sistem Informasi	
00000058345	Jericho Alexander Gunawan	Sistem Informasi	
00000057903	Nicholas Suwandi Wijaya	Sistem Informasi	
00000058204	Rheinald Adyatma Rizqi Wibowo	Sistem Informasi	

Topik	:	Aplikasi Toko Buku Online
Judul Project (Indonesia)	:	PERANCANGAN PROGRAM APLIKASI TOKO BUKU ONLINE “BUKUPEDIA”
Pembimbing	:	<< L00079 - Agus Sulaiman >>

BAB 1

PENDAHULUAN

1.1 Deskripsi Aplikasi

Aplikasi yang kami kembangkan, merupakan sebuah inovasi yang telah dikembangkan sebelumnya. Dengan mengembangkan inovasi yang sudah ada, kelompok kami ingin membuat aplikasi *E-commerce*. Pengembangan serta pembuatan aplikasi yang kita kembangkan, karena ingin membuat aplikasi seperti *E-commerce* yang pada umumnya menggunakan *mobile device*. Sehingga pada kesempatan yang baik ini, pada kelompok kami akan membuat sebuah aplikasi *E-commerce* berdasarkan *mobile* sesuai dengan mata kuliah yang kami tempuh.

Aplikasi yang kita kembangkan merupakan aplikasi pembelian buku yang menggunakan perangkat *mobile*. Sehingga para *user* dapat menggunakan aplikasi yang kita kembangkan untuk membeli buku secara *real-time*. Kemudian pada bahasa pemrograman yang kami gunakan dalam aplikasi ini adalah bahasa Java dan juga XML. Dengan banyaknya saingan dalam dunia perindustrian *E-commerce* ini, maka dengan itu aplikasi ini dapat bersaing dalam skala lingkup yang kecil terlebih dahulu. Selanjutnya target pasar dari aplikasi *E-commerce* ini merujuk kepada siapapun yang hendak membeli buku secara cepat, dan efisien.

1.2 Latar Belakang

Buku merupakan peranan penting dalam pembelajaran pengembangan ilmu pengetahuan. Sejak dahulu hingga sekarang ini buku masih digunakan meskipun sudah terdapat media - media baru lainnya. Di Indonesia, buku berperan penting dalam mendukung kemajuan di dalam bidang pendidikan. Untuk mendapatkan sebuah buku bukanlah hal yang sulit, karena buku dapat ditemukan di toko buku untuk mendapatkannya. Dengan semakin berkembangnya teknologi maka semakin banyak dan bertambahnya toko buku modern dan internasional yang dapat membuat masyarakat mempunyai banyak pilihan untuk membeli buku.

Dari zaman ke zaman manusia tidak luput dari berbelanja karena pada dasarnya hal tersebut sudah menjadi kewajiban untuk memenuhi kebutuhannya. Dengan memanfaatkan teknologi, hal tersebut dapat dipermudah karena dapat dilakukan dimana saja dan kapan saja melalui berbagai macam device seperti *smartphone* atau komputer. Namun, berbelanja merupakan aktivitas yang sulit bagi sebagian orang yang sibuk, tidak memiliki banyak waktu luang, dan selalu bepergian. Selain itu, orang-orang merasa aktivitas ini tidak nyaman karena mengharuskan mereka untuk mengunjungi situs web, memasukkan alamat situs web di browser, memilih menu, dan memutuskan apakah akan melakukan pembelian. Di zaman sekarang, pada abad ke-20 ini aplikasi *mobile* sudah sangat mempengaruhi segala aspek kehidupan pada masyarakat. Menurut Simarmata (2006 : iii) “ Mobile

internet akan membawa revolusi terhadap cara bergaul dengan keluarga, dan tetangga, cara berbisnis, cara memperoleh hiburan, cara mengelola keuangan dan lain-lain”.

Pada penelitian kali ini, kelompok kami akan membuat dan merancang suatu aplikasi mobile Bukupedia yang memungkinkan orang-orang untuk berbelanja dengan praktis. Dimana, bukupedia merupakan salah satu perusahaan dagang yang bergerak di dalam dunia edukasi dan hiburan yang menjual segala macam buku yang dapat memenuhi kebutuhan pembeli. Dengan begitu para user dapat mengakses dan melakukan transaksi dengan mudah menggunakan smartphone atau komputer mereka. Kemudian dengan adanya aplikasi ini, mereka dapat senantiasa bergerak (*mobile*) tanpa harus meluangkan waktu mereka untuk singgah ke toko buku untuk mencari buku yang ingin mereka beli. Berdasarkan uraian diatas, solusi yang akan ditulis dan diangkat dalam penulisan laporan ini berjudul “PERANCANGAN PROGRAM APLIKASI TOKO BUKU ONLINE “BUKUPEDIA” UNIVERSITAS MULTIMEDIA NUSANTARA”.

1.3 Rumusan Masalah

Dari latar belakang diatas ada berbagai macam permasalahan yang dapat diidentifikasi, maka dari itu permasalahan yang akan dikaji dalam project ini akan dirumuskan sebagai berikut:

1. User tidak dapat membeli buku secara cepat dan efisien.
2. Proses transaksi yang masih dilakukan secara manual sehingga menyulitkan pelanggan yang berlokasi jauh untuk melakukan proses transaksi
3. Manajemen buku yang tidak teratur dan dikelompokkan sesuai kategori buku yang menyulitkan staff kesusahan dalam melakukan pencarian buku
4. Proses pembuatan laporan keuangan yang masih dilakukan dengan manual yang memungkinkan data laporan hilang atau tidak terbaca lagi karena rentang waktu yang lama.

1.4 Tujuan Pembuatan Aplikasi

Berdasarkan rumusan masalah yang dibuat oleh kelompok kami, maka terdapat tujuan yang ingin dicapai dari pembuatan aplikasi yaitu, sebagai berikut:

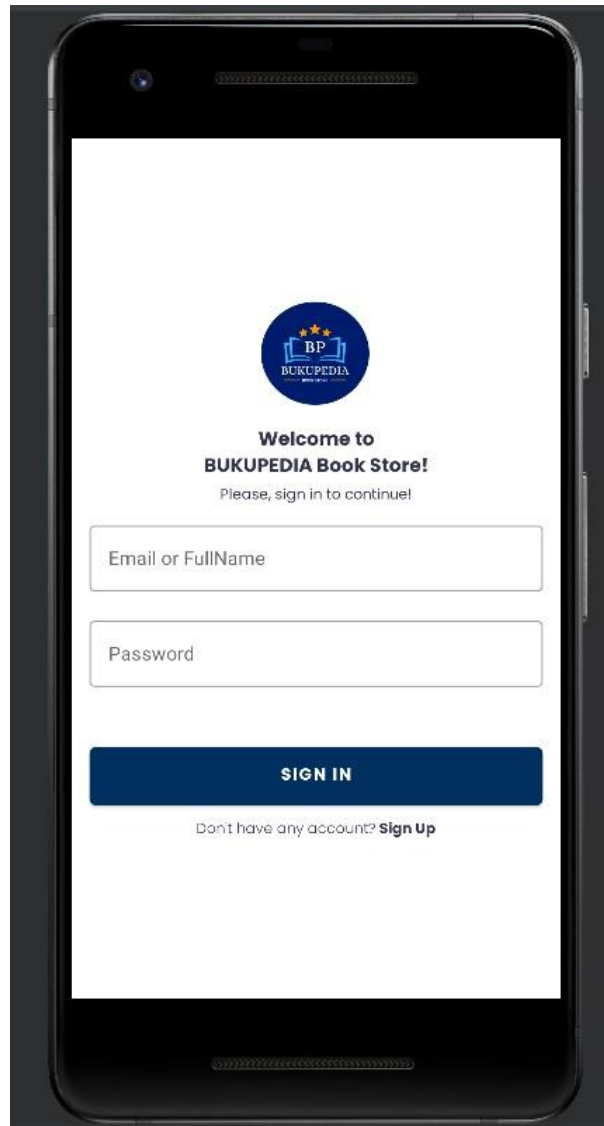
1. Membantu konsumen dalam melakukan proses transaksi dimana saja dan kapan saja
2. Untuk memberikan kemudahan bagi pembeli dalam mendapatkan buku tanpa harus datang secara langsung
3. Dapat meningkatkan penjualan tanpa harus membangun toko yang membutuhkan modal besar
4. Membuat dan merancang aplikasi toko buku online berbasis aplikasi

BAB 2

PEMBAHASAN

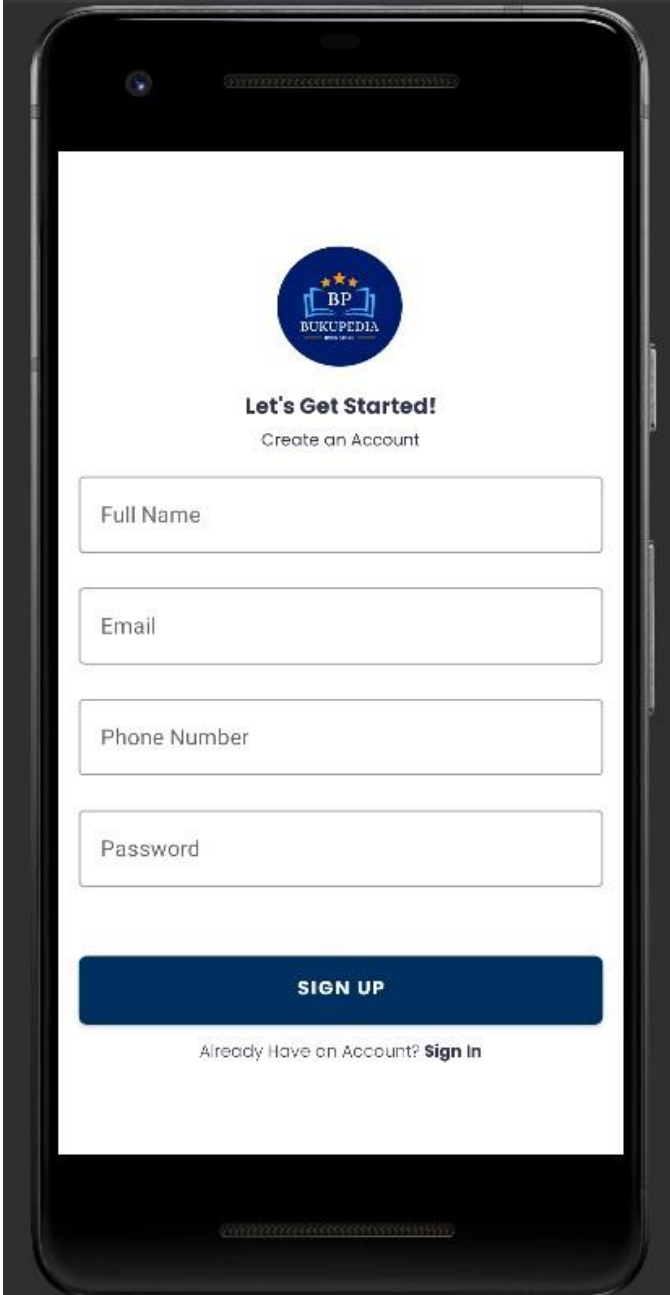
2.1 Menu dan Fitur pada Aplikasi BUKUPEDIA

2.1.1 Sign in



Fitur ini digunakan ketika user ingin masuk ke dalam aplikasi setelah mempunyai akun, yaitu dengan cara mendaftarkan terlebih dahulu ke fitur yang sudah tersedia pada aplikasi tersebut yaitu Sign Up. Dengan ini user dapat mengakses aplikasi dengan mengisi data yang sudah disediakan.

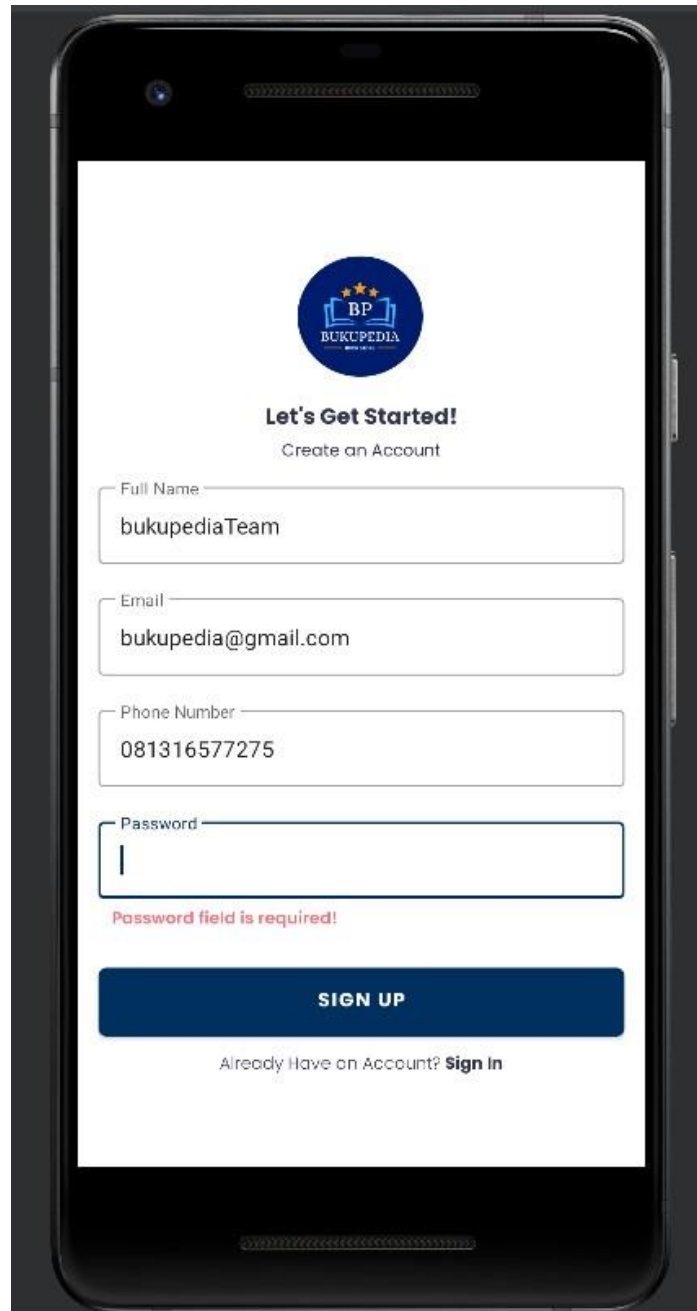
2.1.2 Sign Up



The image shows a smartphone screen displaying the 'Sign Up' interface for the 'BP BUKUPEDIA' application. At the top, there is a circular logo with 'BP' and 'BUKUPEDIA' text. Below the logo, the text 'Let's Get Started!' is followed by 'Create an Account'. The form consists of four input fields: 'Full Name', 'Email', 'Phone Number', and 'Password'. A blue 'SIGN UP' button is positioned below the fields. At the bottom, there is a link that says 'Already Have an Account? Sign In'.

Fitur Sign Up berfungsi untuk user baru yang ingin masuk ke dalam aplikasi. Jika pengguna tidak memiliki sebuah akun dalam aplikasi maka user harus mengisi data - data pribadinya terlebih dahulu dan dapat melakukan pendaftaran melalui Sign Up.

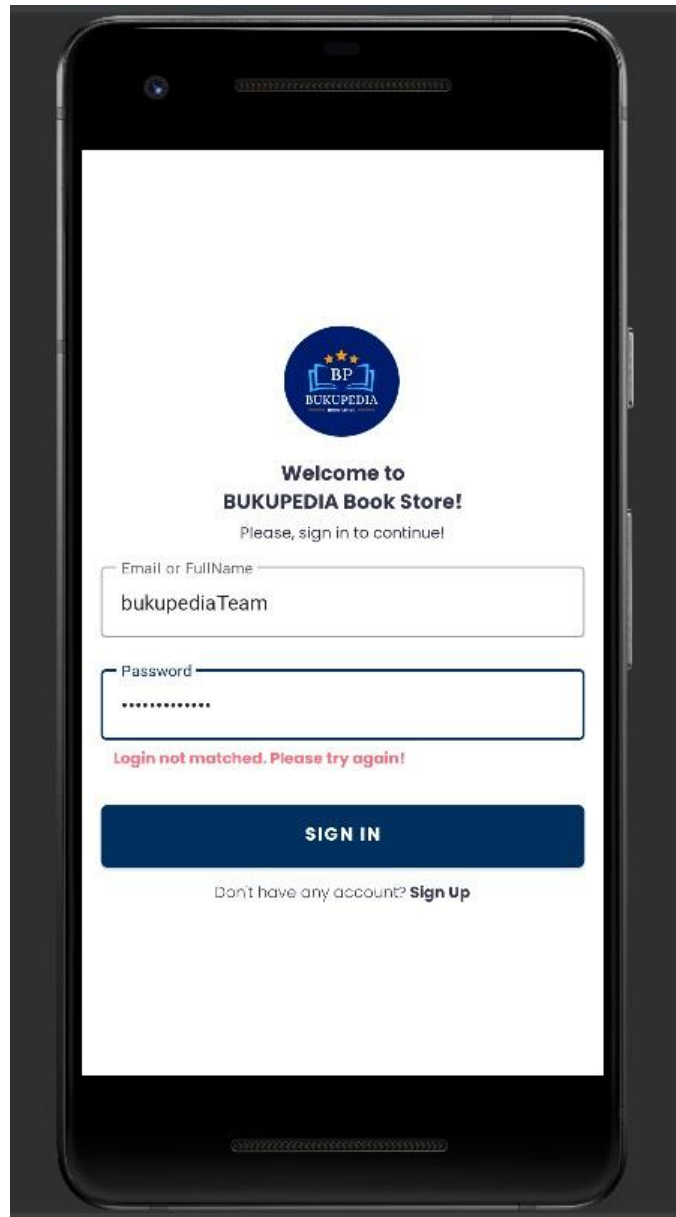
2.1.3 Error Sign Up



The image shows a mobile application interface for BUKUPEDIA. At the top, there is a blue circular logo with a book icon and the text "BP BUKUPEDIA". Below the logo, the text "Let's Get Started!" is displayed in bold, followed by "Create an Account" in a smaller font. The form contains four input fields: "Full Name" with the value "bukupediaTeam", "Email" with the value "bukupedia@gmail.com", "Phone Number" with the value "081316577275", and "Password" which is currently empty. Below the password field, a red error message "Password field is required!" is displayed. At the bottom of the form, there is a blue "SIGN UP" button and a link that says "Already Have an Account? Sign in".

Fitur Error Sign Up berfungsi untuk memberi tahu user bahwa ada text view yang harus dipenuhi (required) oleh user seperti minimal nama, harus menggunakan (“@”) pada password user dan lain lain. Hal ini diperlukan agar user tidak sembarangan mengisi data yang diperlukan oleh BUKUPEDIA dalam memproses data yang ada.

2.1.4 Error Sign In



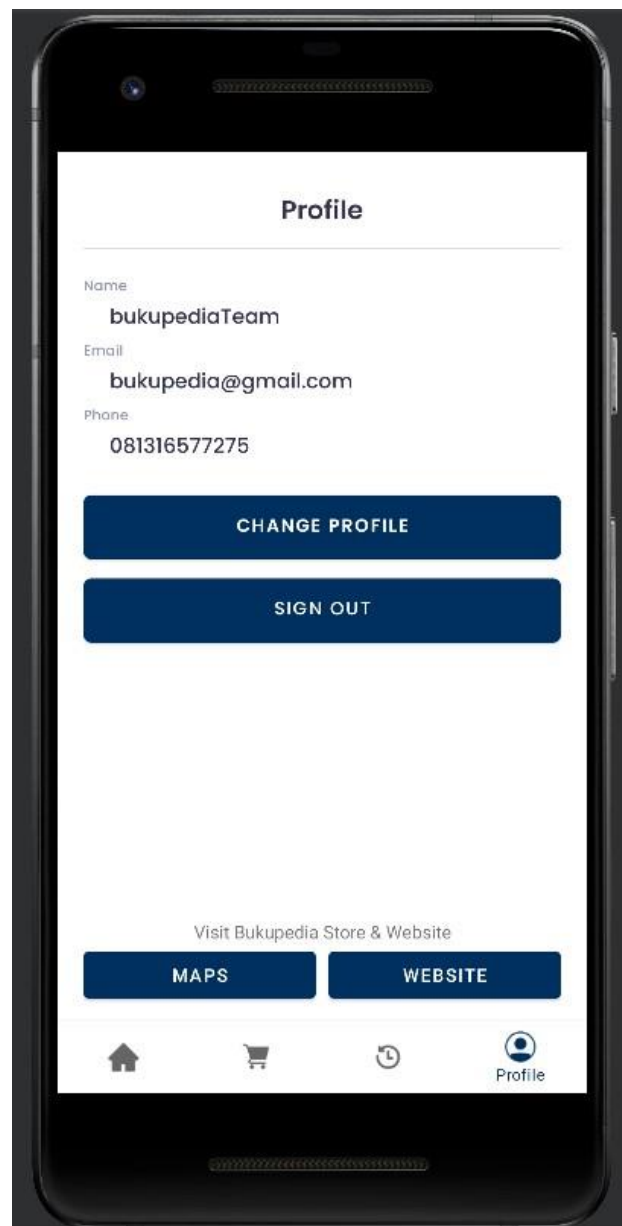
Fitur Error Sign In berfungsi untuk memberi tahu user bahwa ia salah memasukkan Email atau Password yang telah ia isi dalam kolom tersebut. Hal ini digunakan agar mencegah user untuk tidak sembarangan dalam mengisi email / password yang telah ia isi pada laman Sign Up.

2.1.5 Searching Book



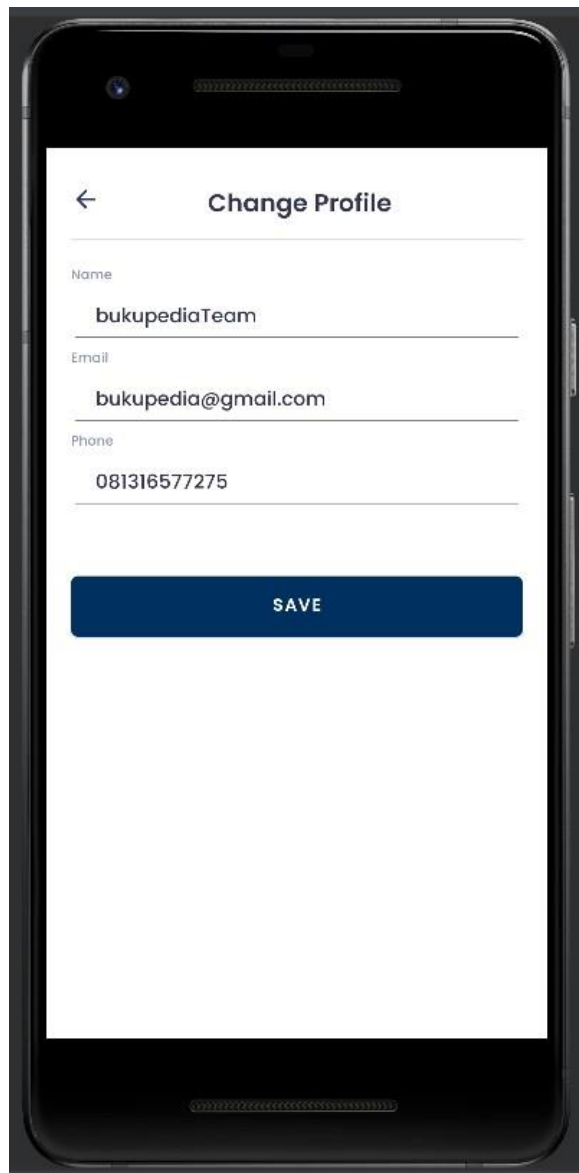
Fitur Search dapat digunakan oleh user untuk mencari buku yang ingin dibeli oleh user. Dalam fitur tersebut, nantinya tampilan yang muncul akan sesuai dengan abjad yang diinput user. Dalam menu search tersebut juga akan mengarahkan user untuk masuk kedalam book detail activity untuk dapat melihat buku yang ingin dilihat secara lebih lanjut.

2.1.6 Profile



Profile disini berguna untuk menampilkan data dari pemilik akun dan username pengguna yang sudah melakukan Log in dan telah melakukan registrasi pada awal yaitu pada fitur Sign Up dalam aplikasi.

2.1.7 Change Profile



← Change Profile

Name
bukupediaTeam

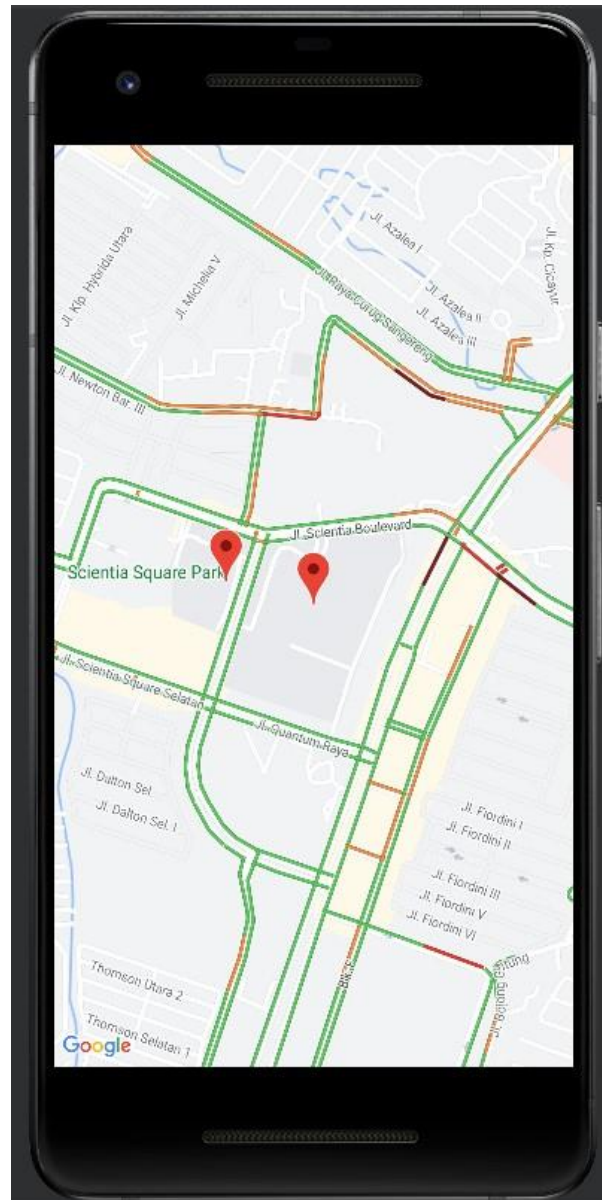
Email
bukupedia@gmail.com

Phone
081316577275

SAVE

Change Profile disini user diberikan kebebasan untuk mengedit profile mereka seperti nama,email, dan nomor telepon.

2.1.8 Maps



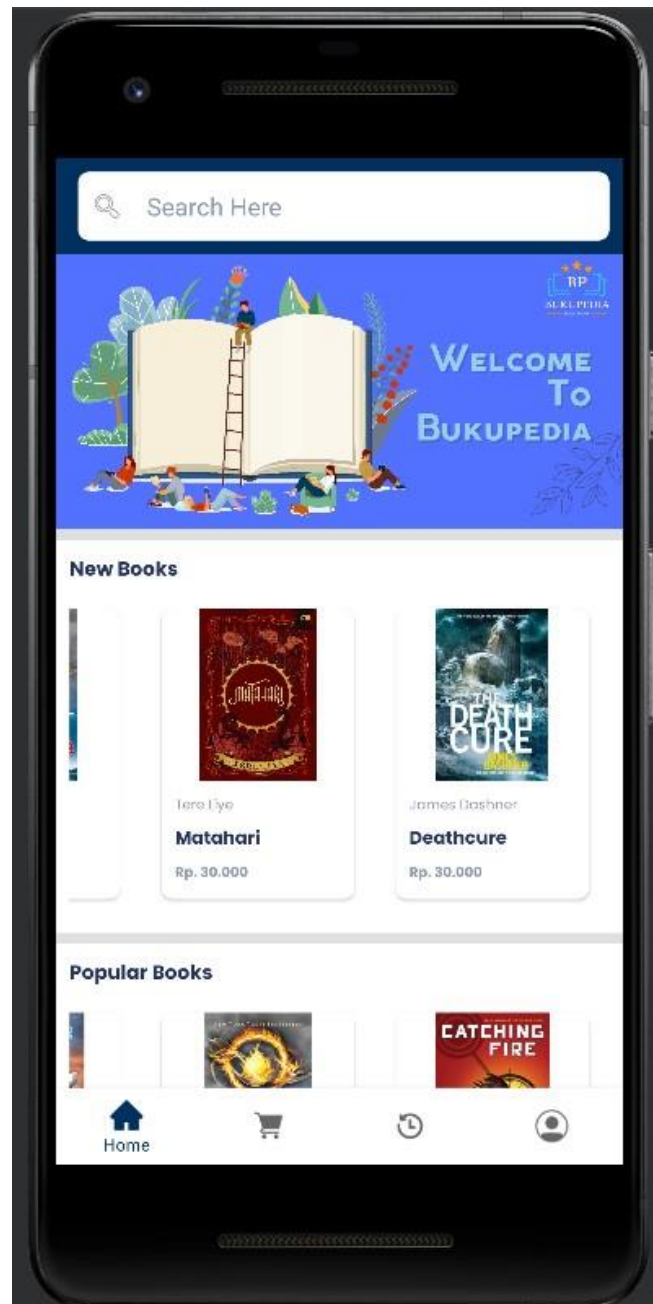
Fitur maps dipergunakan oleh user dalam melihat lokasi perusahaan BUKUPEDIA.

2.1.9 WebView



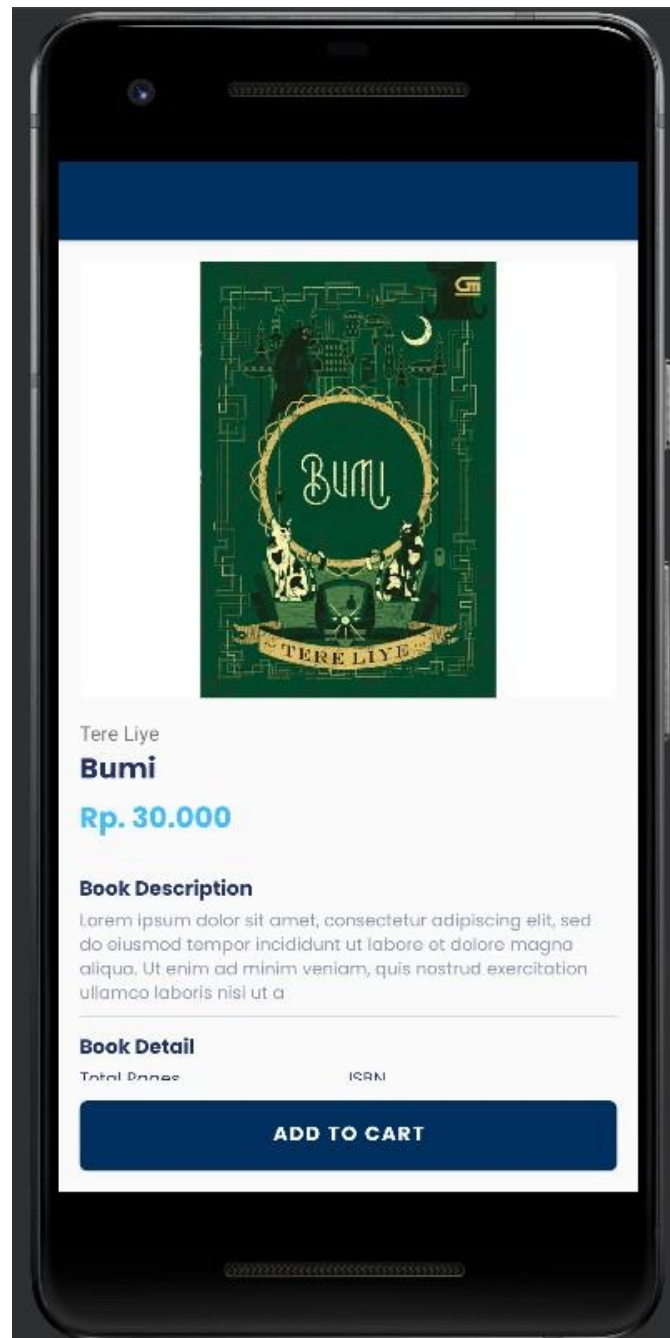
Fitur WebView dipergunakan oleh user untuk melihat website resmi dari BUKUPEDIA.

2.1.10 Home



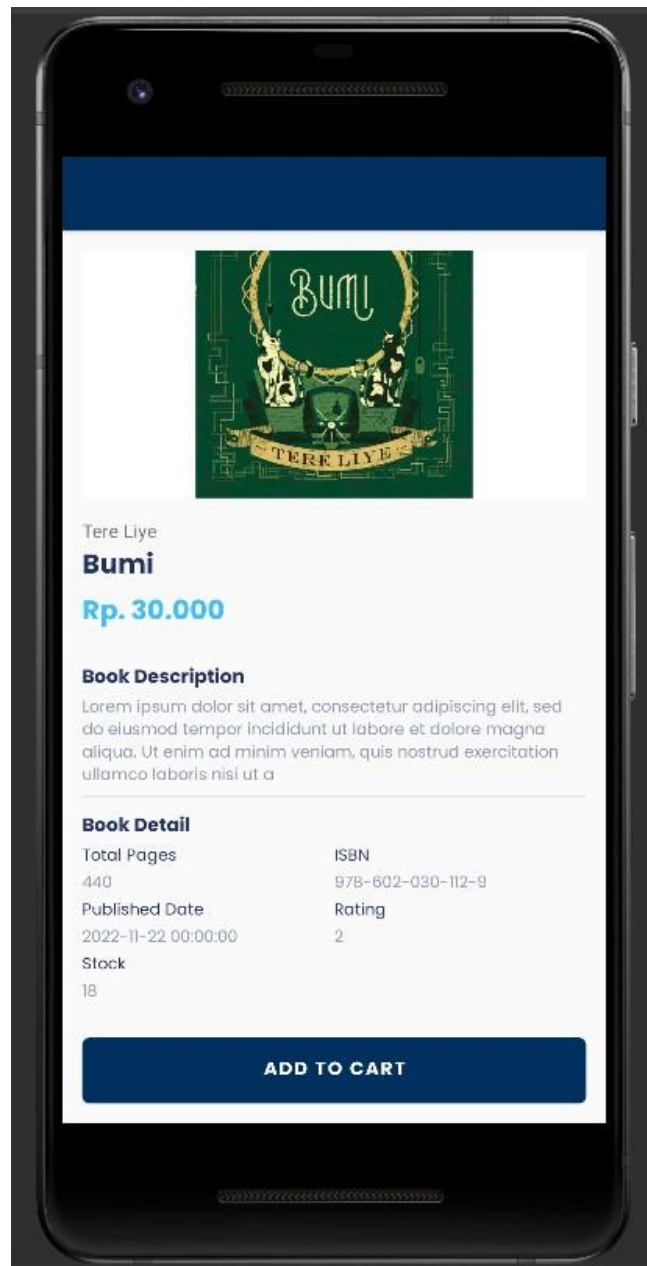
Tampilan Home merupakan halaman yang pertama kali dilihat user saat mengakses aplikasi. Hal ini ditujukan untuk menarik perhatian orang dengan berbagai macam pilihan buku yang ada. Tampilan Home ini berisi banner (yang dapat digerakkan secara otomatis), Tampilan New Books dan Tampilan Popular Books.

2.1.11 Detail Buku



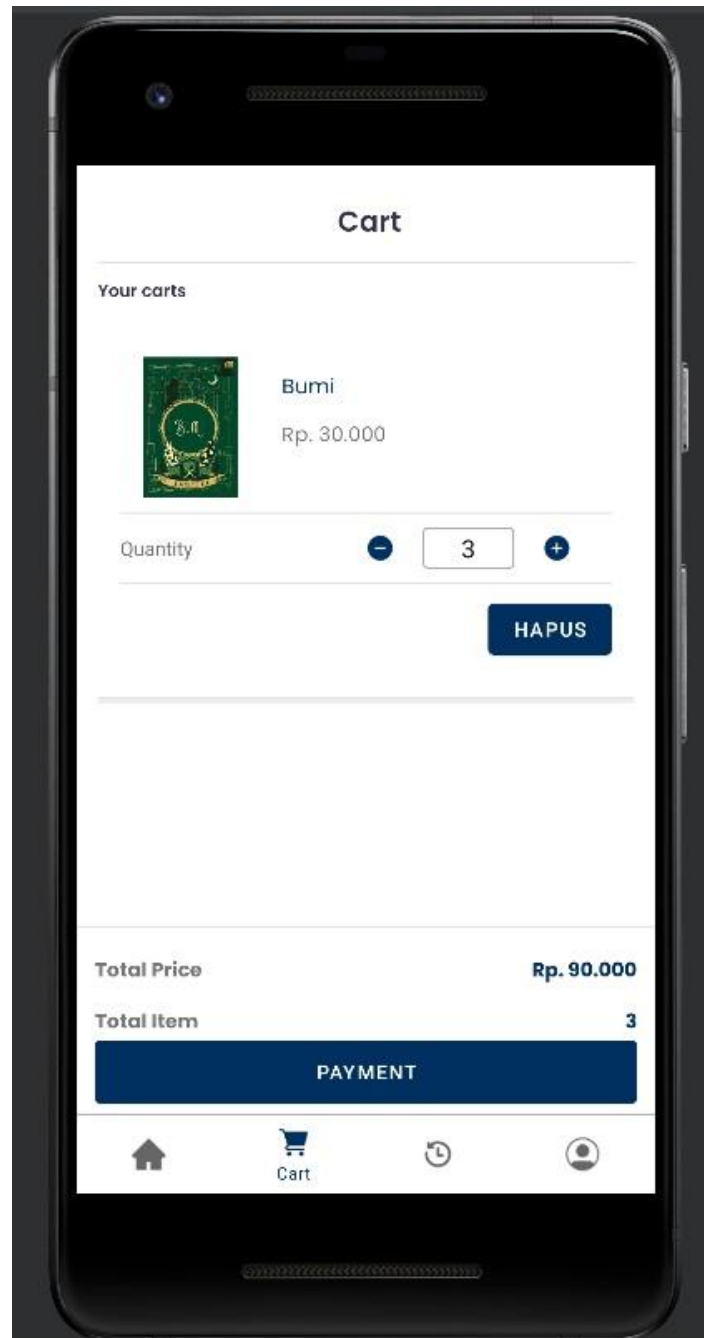
Detail Buku yang terdapat deskripsi dari buku tersebut lalu berikutnya terdapat harga dari buku tersebut, dan juga penulis buku tersebut.

2.1.12 Detail Buku (2)



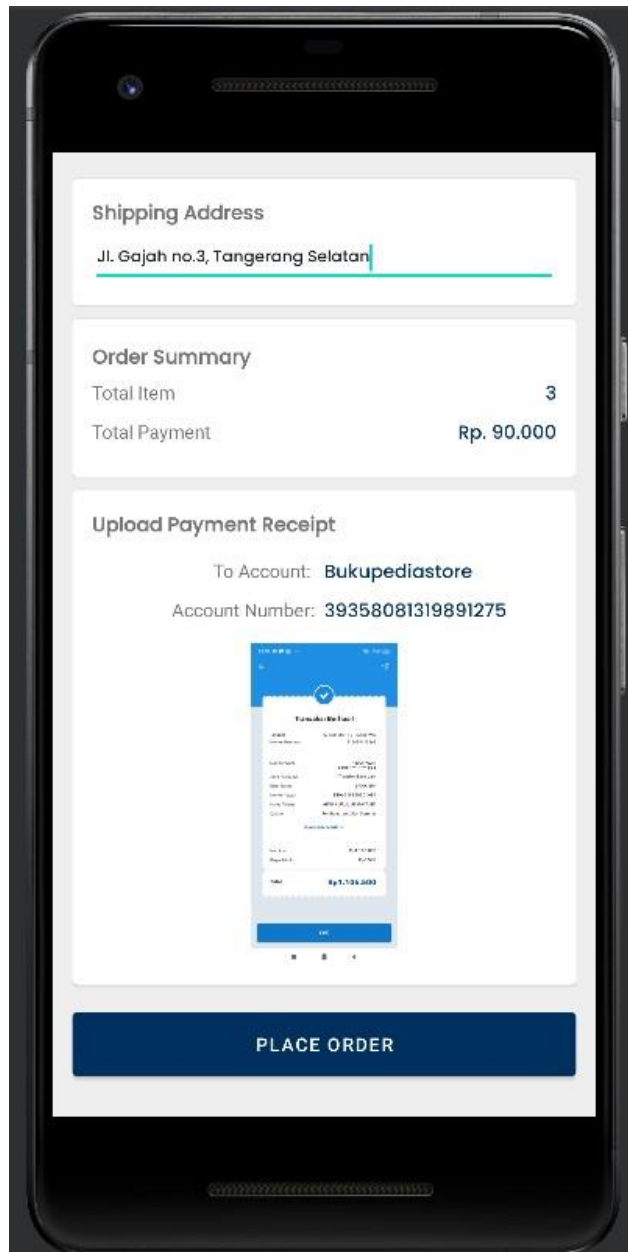
Detail Buku yang kedua yakni terdapat jumlah halaman pada buku tersebut, lalu kapan buku tersebut di launch/rilis. Lalu ada jumlah stok dari buku tersebut dan yang terakhir ada rating dari buku tersebut.

2.1.13 Cart



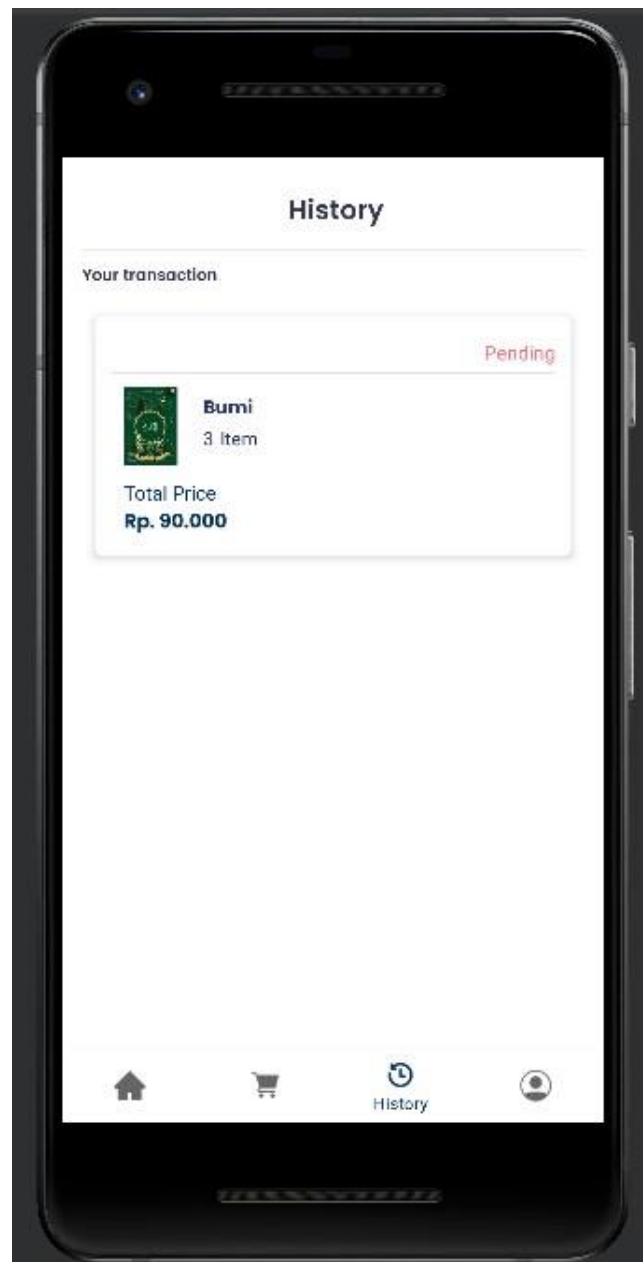
Tampilan cart ini dimana user dapat mengedit quantity jumlah buku yaitu dengan cara menambahkan dan mengurangi jumlah buku yang ingin dimasukan ke dalam cart terlebih dahulu. Ketika sudah menentukan jumlah quantity yang diinginkan maka user dapat memilih item dan melakukan pembelian.

2.1.14 Checkout



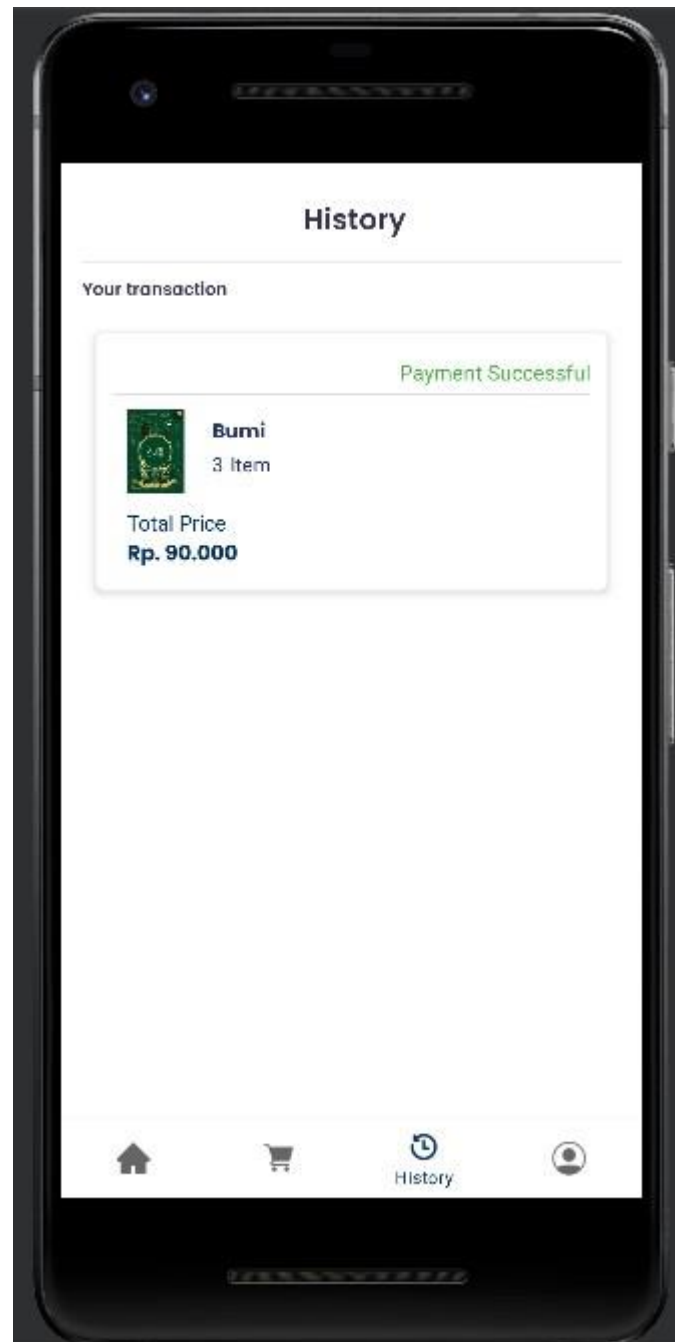
Checkout merupakan dimana user yang sudah yakin melakukan pembayaran dan user disini dapat melihat detail buku terlebih dahulu sebelum melakukan transaksi dan user juga dapat melihat judul buku, genre dan penulisnya terlebih dahulu. Dalam fitur ini user harus memasukan tujuan lokasi yang harus dikirimkan. Kemudian tahap yang terakhir user harus memberikan buktinya terlebih dahulu dalam bentuk screenshot kemudian user dapat melanjutkan ke step berikutnya.

2.1.15 History



History berisi dari pembelian buku yang telah dibeli dan akan muncul dari pesanan yang telah di place order sebelumnya. History berisi dari buku apa yang telah dibeli, status pembelian (apakah sudah dibayar atau belum) dll. Dalam tampilan diatas dapat dilihat bahwa status pembayaran sekarang masih dalam keadaan pending.

2.1.16 Payment Success



Di dalam history diatas dapat dilihat bahwa history diatas menampilkan bahwa pembayaran telah berhasil dilakukan oleh user.

2.2. Capture Coding Yang Digunakan Dalam Aplikasi “BUKUPEDIA”

2.2.1. ChangeProfileActivity

```
package com.example.buku.activity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.buku.Handler.Constant;
import com.example.buku.Handler.RequestHandler;
import com.example.buku.Handler.SQLite;
import com.example.buku.R;
import com.example.buku.adapter.Session;
import org.json.JSONException;
import org.json.JSONObject;
```

```

import java.util.HashMap;

import java.util.Map;

import java.util.regex.Pattern;

public class ChangeProfileActivity extends AppCompatActivity {

    // Inisial awal buat edit-text -/dkk

    EditText edtName, edtEmail, edtPhone, edtAddress;

    Button btnSave;

    ImageView imgBack;

    String pesan;

    TextView TULISAN_EROR;

    // memanggil sqlite

    private SQLite dbsqlite;

    // pembatasan buat Jumlah nama

    final Pattern NAMA_PATTERN =

        Pattern.compile("^" +

            "{5,}" //at least 5 characters

        );

    // pembatasan buatumlah no telepon

    final Pattern telepon_PATTERN =

        Pattern.compile("^" +

            "{9,}" //at least 9 characters

        );

    // buat dipakai dibawah

    String name, email, noHP_USER, hape, id_data_User;

    String namaa, email1, id_sql;

    @Override

```

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_change_profile);

    // dipanggil di on create

    inisialisasi();

    // ambil data dari sql lite untuk isi nama biar bisa nyocokin update
    dibawah

    dbsqlite = new SQLite(getApplicationContext());

    HashMap<String, String> user = dbsqlite.getUserDetails();

    // dan yang diambil adalah data nama dan email dari SQLite

    name = user.get("nama");

    email = user.get("email");

    id_sql = user.get("id");

    // ambil dari home activity buat ambil data bawaan berupa nomer hp
    yang berada pada session!

    HomeActivity activity = new HomeActivity();

    noHP_USER = activity.getUserDATA_NOHAPE();

    id_data_User = activity.getUserDATA_ID();

    // masukkan kedalam textnya biar ga koosong!

    edtName.setText(name);

    edtEmail.setText(email);

    edtPhone.setText(noHP_USER);

    // dia buat tombol kembali

    mainButton();

    getSupportActionBar().hide();

}
```

```

private void inisialisasi() {

    //inisialisasi();// init awal semua data yang dibutuhkan

    edtName = findViewById(R.id.edt_addressss);

    edtEmail = findViewById(R.id.edt_email);

    edtPhone = findViewById(R.id.edt_phone);


    TULISAN_EROR = findViewById(R.id.tv_errortext_UPDATE);

    btnSave = findViewById(R.id.btn_save);

    imgBack = findViewById(R.id.img_back);

}

private boolean pengubahan() {

    //get text

    namaa = edtName.getText().toString().trim();

    email1 = edtEmail.getText().toString().trim();

    hape = edtPhone.getText().toString().trim();


    return false;

}

public void SAVE_BTN(View view) {

    //PENGECEKAN ketika nama kosong DLL dia keluar notif !

    pengubahan();

    if (namaa.length() < 1) {

        TULISAN_EROR.setText("Name field is required!");

        edtName.requestFocus();

        return;

    }
}

```



```
// KETIKA USERNAME LEBIH KECIL DARI 5 MAKA MASUK KE CLASS DIBAWAH YANG  
MENGURUSNYA
```

```
if (! (NAMA_PATTERN).matcher(namaa).matches()){  
  
    TULISAN_EROR.setText("A Minimum of 5 letters is required for the  
FullName !");  
  
    edtName.requestFocus();  
  
    return;  
  
}
```

```
// JIKA EMAIL KOSONG MAKA DIA TURUN MENEGAMBIL CLASS YANG MENGURUSNYA  
DIBAWAH
```

```
if (email1.isEmpty()) {  
  
    if (email1.length()<5) {  
  
        TULISAN_EROR.setText("Email field is required ! ");  
  
        edtEmail.requestFocus();  
  
    }  
  
    return;  
  
}
```

```
// JIKA EMAIL TIDAK SESUAI DENGAN KETENTUAN EMAIL SECARA GLOBAL  
SEMISAL
```

```
// @BLABLA.COM DIA AKAN MENGELUARKAN NOTIF
```

```
if (!Patterns.EMAIL_ADDRESS.matcher(email1).matches()){  
  
    TULISAN_EROR.setText("The email address you entered is not valid !  
");  
  
    edtEmail.requestFocus();  
  
    return;  
  
}
```

```
// jika NO TELP ANKGANYA CUMAN 1 DIA MENAMPILKAN PESAN!
```

```
if (hape.length()<1) {  
  
    TULISAN_EROR.setText("Phone number field is required!");  
  
}
```

```

        edtPhone.requestFocus();

        return;
    }

    if (hape.length() > 12) {

        TULISAN_EROR.setText("Your Phone number is too long!");

        edtPhone.requestFocus();

        return;
    }

    // JIKA NO TELPON TIDAK SESUAI DENGAN KETENTUAN YAITU NGACO NOMERNYA (
    MINIMAL 8 DIGIT )

    // MAKA DIA EROR (MENAMPILKAN PESAN)

    if (! (telepon_PATTERN).matcher(hape).matches()) {

        TULISAN_EROR.setText("The phone number you entered is not
valid!");

        edtPhone.requestFocus();

        return;
    }

    else{

        // progress dialog dulu bos biar keren

        final ProgressDialog progressDialog = new ProgressDialog(this);

        progressDialog.setMessage("Updating.....");

        progressDialog.show();

        pengubahan();

        /** mulailah menggunakan library volley dibawah ini untuk
mengubungkan ke api lalu ke database,

        * dibawah ini menggunakan metode post, yang berarti metode HTTP
Request

        * yang digunakan untuk membuat data baru dengan menyisipkan data
dalam body saat request dilakukan

        */

```

```

        StringRequest request = new StringRequest(Request.Method.POST,
Constant.URL_UPDATE,

        new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                //adapter.notifyDataSetChanged();

                try {

                    JSONObject jsonObject = new
JSONObject(response);

                                // mengambil json object untuk mendapatkan
nilai dari api yang telah dibuat

                    int sukses = jsonObject.getInt("success");

                    pesan = jsonObject.getString("message");

                                if (sukses == -1){ // email udah ada yang
pakek jadi ga bisa update datanya

                                progressDialog.dismiss();

                                TULISAN_EROR.setText(pesan);

                                edtEmail.requestFocus();

                                return;

                                }

                                if (sukses == 0){ // data gagal diupdate
karena jaringan ataupun karena masalah lainnya

                                progressDialog.dismiss();

                                TULISAN_EROR.setText(pesan);

                                edtEmail.requestFocus();

                                edtName.requestFocus();

                                edtPhone.requestFocus();

                                return;

```

```

    }

    if (sukses == -2 ) { // datanya kosong atau
gagal untuk di update karena kosong

        progressDialog.dismiss();

        TULISAN_EROR.setText(pesan);

        edtEmail.requestFocus();

        edtName.requestFocus();

        edtPhone.requestFocus();

    }

    if (sukses == 1) { // data berhasil diupdate

        // inside on response method we are

        // setting our edit text to empty.

        edtName.setText("");

        edtEmail.setText("");

        edtPhone.setText("");

        // on below line we are displaying a toast
message as data updated.

        Toast.makeText(ChangeProfileActivity.this,
        "Data Updated..", Toast.LENGTH_SHORT).show();

        dbsqlite.updateUser(id_sql, namaa,
email1);

        // masukan session ye ges !

        Session.getInstance(getApplicationContext())

            .userUpdated(

                hape

            );

        //membuka activity baru buat refresh

        Intent i = new Intent(getApplicationContext(),
        HomeActivity.class);

```

```

        startActivity(i);

        // menyatakan selesai

        finish();

        progressDialog.dismiss();

    }

}

        catch (JSONException jsonObject) { // ketika eror
dia akan ngeprint dimana jsonobjectnya eror

            jsonObject.printStackTrace();

        }

    }

    }, new Response.ErrorListener() {

        @Override

        public void onErrorResponse(VolleyError error) { // jika eror
toast akan keluar tulisan fail to update data

            // displaying toast message on response failure.

            Toast.makeText(ChangeProfileActivity.this, "Fail to update
data..", Toast.LENGTH_SHORT).show();

            progressDialog.dismiss();

        }

    }) {

        /**

        * metode dibawah ini adalah yang digunakan untuk memasukkan
data imputan dari aplikasi kedalam api,

        * dengan mencocokkan pada apa yang telah dideklarasikan pada
methode post di api (php - nya), sehingga

        * bisa terhubung.... kedalam api dan bisa mennyampaikannya
kedalam database bahwa terdapat insert - baru

        */

        @Override

```

```

        protected Map<String, String> getParams() throws
AuthFailureError {

            Map<String, String> params = new HashMap<>();

            params.put("id", id_data_User);

            params.put("nama", namaa);

            params.put("email", email);

            params.put("nohp", hape);

            return params;

        }

    };

    // baru bisa dijalankan

    RequestHandler.getInstance(this).addToRequestQueue(request);

}

}

private void mainButton() {

    // tombol kembali

    imgBack.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            Intent i = new Intent(getApplicationContext(), HomeActivity.class);

            startActivity(i);

        }

    });

}

}

```

Code diatas digunakan untuk melakukan pergantian profile yang dilakukan oleh pengguna (user). Activity tersebut tersambung dengan layout change profile yang menunjukkan bahwa data tersebut dapat diupdate oleh user dengan ketentuan-ketentuan saat user mendaftarkan akunya saat pertama kali (minimal karakter nama,

minimal panjang nomor hp, dll), Disini juga menerapkan konsep sqlite yang digunakan dalam mencocokkan update dengan data yang sudah ada di dalam database.

2.2.2. CheckOut Activity

```
package com.example.buku.activity;

import static
com.example.buku.Handler.Constant.url_upload_bukti_pembayaran_bukan_gambar;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.Context;

import android.content.Intent;

import android.graphics.Bitmap;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.telephony.ims.RegistrationManager;

import android.util.Base64;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import androidx.activity.result.ActivityResult;

import androidx.activity.result.ActivityResultCallback;

import androidx.activity.result.ActivityResultLauncher;

import androidx.activity.result.contract.ActivityResultContracts;

import androidx.appcompat.app.AppCompatActivity;
```

```
import com.android.volley.AuthFailureError;

import com.android.volley.DefaultRetryPolicy;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import com.example.buku.Handler.Constant;

import com.example.buku.Handler.RequestHandler;

import com.example.buku.Handler.SQLite;

import com.example.buku.R;

import com.example.buku.dto.BookCartDTO;

import com.example.buku.dto.BookDTO;

import com.example.buku.fragment.CartFragment;

import com.example.buku.fragment.HistoryFragment;

import org.json.JSONException;

import org.json.JSONObject;


import java.io.ByteArrayOutputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;


public class CheckoutActivity extends AppCompatActivity {

    // Inisial awal buat edit-text -/dkk

    TextView Total_payment_id,tv_summary_item;
```



```

Button btnCheckout;

EditText edt_address;

String HARGA_KESELURUHAN;

String TOTAL_BUKU_YANG_DI_BELI;

ImageView img;

String address;

Bitmap bitmap;

String id_data_User;

BookCartDTO bookCartDTO;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    // jangan lupa ambil intent buat datanya

    bookCartDTO = (BookCartDTO)
getIntent().getSerializableExtra("data_123");

    setContentView(R.layout.activity_checkout);

    inisialisasi();

    mainButton();

    getSupportActionBar().hide();

    img = findViewById(R.id.image_piew);

    /**

     * dibawah ini digunakan untuk intent kedalam media store storage
masing masing - dari user

     * gunanya untuk mengambil gambar untuk proses upload di checkout buku

     */

    ActivityResultLauncher<Intent> activityResultLauncher =

```

```

        registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new
ActivityResultCallback<ActivityResult>() {

            @Override

            public void onActivityResult(ActivityResult result) {

                if(result.getResultCode() == Activity.RESULT_OK) {

                    Intent data = result.getData();

                    Uri uri = data.getData();

                    try {

                        bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), uri);

                        img.setImageBitmap(bitmap);

                    } catch (IOException e) {

                        e.printStackTrace();

                    }

                }

            }

        });

img.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Intent intent = new Intent(Intent.ACTION_PICK);

        intent.setData(MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

        activityResultLauncher.launch(intent);

    }

});

}

private boolean pengubahan(){ //penampungan sementara

    address = edt_address.getText().toString().trim();

```

```

        return false;
    }

    private void inisialisasi(){ // inisialisasi

        btnCheckout = findViewById(R.id.btn_checkout);

        Total_payment_id = findViewById(R.id.Total_payment_id);

        tv_summary_item = findViewById(R.id.Summary_item_id);

        Intent intent = getIntent();

        HARGA_KESELURUHAN = intent.getStringExtra("loadsPosition");

        TOTAL_BUKU_YANG_DI_BELI = intent.getStringExtra("loads_summary");

        Total_payment_id.setText("Rp." +HARGA_KESELURUHAN + "00");

        tv_summary_item.setText(TOTAL_BUKU_YANG_DI_BELI);

        edt_address = findViewById(R.id.edt_addressss);

        HomeActivity activity = new HomeActivity();

        id_data_User = activity.getUserDATA_ID(); // buat id_user
    }

    private void mainButton(){

        // button checkout dikasi listener biar bisa di tekan dan menjalankan aksinya

        btnCheckout.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                pengubahan();

                if (edt_address.length()<1) { // jika address blm di isi, maka notif dibawah ini akan keluar " please input your address corectly"

                    edt_address.requestFocus();

                    Toast.makeText(CheckoutActivity.this,"Please input your address correctly",Toast.LENGTH_SHORT).show();

                }

                else{

```

```

        // menjalankan sebagaimana proses - penguploadan gambar -
        beserta data pada public void dibawah

        kedalam_transaction_item();

        upload_gambar();

    }

}

});

}

// upload book_id, id_user, id_cart, qty (TOTAL), total_price, Address.

public void kedalam_transaction_item(){

    /** mulailah menggunakan library volley dibawah ini untuk menghubungkan
    ke api lalu ke database,

    * dibawah ini menggunakan metode post, yang berarti metode HTTP
    Request

    * yang digunakan untuk membuat data baru dengan menyisipkan data
    dalam body saat request dilakukan

    */

    StringRequest request = new StringRequest(Request.Method.POST,
    Constant.url_upload_bukti_pembayaran_bukan_gambar, new
    Response.Listener<String>() {

        @Override

        public void onResponse(String response) {

            try {

                JSONObject jsonObj = new JSONObject(response);

                // mengambil json object untuk mendapatkan nilai dari api
                yang telah dibuat

                int sukses = jsonObj.getInt("success");

                String pesan = jsonObj.getString("message");

                if (sukses == -1) {

                    Toast.makeText(CheckoutActivity.this, pesan,
                    Toast.LENGTH_SHORT).show();

                    return;

```

```

        }

        // Error in registration! // user failed to store

        // jika gagal didaftarkan

        if (sukses == -2){

            Toast.makeText(CheckoutActivity.this, pesan,
Toast.LENGTH_SHORT).show();

            return;

        }

        // jika ada kesalan dalam PENGISIAN KE DALAM CART

        if (sukses == -3) {

            Toast.makeText(CheckoutActivity.this, pesan,
Toast.LENGTH_SHORT).show();

            return;

        }

    } catch (Exception ex) { // ketika eror

        Toast.makeText(CheckoutActivity.this, "Something Wrong
with your connection", Toast.LENGTH_SHORT).show();

        Log.e("Error: ", ex.toString());

        ex.printStackTrace();

    }

}

}, new Response.ErrorListener() {

    @Override

    public void onErrorResponse(VolleyError error) { // ketika eror

        Toast.makeText(CheckoutActivity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();

        Log.e("Error: ", error.getMessage());

    }

}

) {

```

```

    /**
        * metode dibawah ini adalah yang digunakan untuk memasukkan data
        imputan dari aplikasi kedalam api,

        * dengan mencocokkan pada apa yang telah dideklarasikan pada
        methode post di api (php - nya), sehingga

        * bisa terhubung.... kedalam api dan bisa mennyampaikannya
        kedalam database bahwa terdapat insert - baru

    */

    @Override

    protected Map<String, String> getParams() throws AuthFailureError
    {

        Map<String, String> params = new HashMap<>();

        /**    yang dipake dibawah ini (nyocokin php(api))

        * upload book_id, id_user, id_cart, qty (TOTAL), total_price,
        Address.

        $bookid = $_POST['book_id'];

        $userid = $_POST['user_id'];

        $cartid = $_POST['cart_id'];

        $qty      = ($_POST['qty']);

        $total_price      = ($_POST['total_price']);

        $Address      = ($_POST['address']);

        */

        params.put("book_id", bookCartDTO.id);

        params.put("user_id", id_data_User);

        params.put("cart_id", bookCartDTO.id_cart);

        params.put("qty", TOTAL_BUKU_YANG_DI_BELI);

        params.put("total_price", HARGA_KESELURUHAN);

        params.put("address", address);

        return params;
    }

```

```

    }

    };

// baru bisa dijalankan

    RequestHandler.getInstance((Context) this).addToRequestQueue(request);

}

// upload gambar ke payment.

public void upload_gambar() {

    // deklarasi dulu

    ByteArrayOutputStream byteArrayOutputStream;

    byteArrayOutputStream = new ByteArrayOutputStream();

    // bitmap itu gambarnya yang diambil dari storage yee,...

    if(bitmap != null) {

        bitmap.compress(Bitmap.CompressFormat.JPEG, 100,
byteArrayOutputStream);

        byte [] bytes = byteArrayOutputStream.toByteArray();

        final String base64Image = Base64.encodeToString(bytes,
Base64.DEFAULT);

        //RequestQueue queue =
Volley.newRequestQueue(getApplicationContext());

        final ProgressDialog progressDialog = new ProgressDialog(this);

        progressDialog.setMessage("Please Wait..");

        // dibuat progress dialognya tidak bisa di cancel, agar keren :v,
maksudnya biar tau kalo proses penguploadsedang jalan

        progressDialog.setCancelable(false);

        progressDialog.show();

        /** mulailah menggunakan library volley dibawah ini untuk
mengubungkan ke api lalu ke database,

        * dibawah ini menggunakan metode post, yang berarti metode HTTP
Request

```

** yang digunakan untuk membuat data baru dengan menyisipkan data dalam body saat request dilakukan*

**/*

```
StringRequest stringRequest = new
StringRequest(Request.Method.POST, Constant.url_upload_bukti_pembayaran,

    new Response.Listener<String>() {

        @Override

        public void onResponse(String response) {

            try {

                // sama seperti yang diatas ya kurang lebihh

                JSONObject jsonObj = new JSONObject(response);

                int sukses = jsonObj.getInt("success");

                if (sukses == 1) {

                    Toast.makeText(getApplicationContext(),
"Upload Payment Success ^.^", Toast.LENGTH_SHORT).show();

                    // masukin ke home page

                    progressDialog.dismiss();

                    Intent i = new Intent(getApplicationContext(),
HomeActivity.class);

                    startActivity(i);

                } else

                    Toast.makeText(CheckoutActivity.this,
"Failed to upload image", Toast.LENGTH_SHORT).show();

            } catch (JSONException e) {

                e.printStackTrace();

            }

        }

    }, new Response.ErrorListener() {

        @Override
```



```

        public void onErrorResponse(VolleyError error){

            Toast.makeText(getApplicationContext(),
error.getLocalizedMessage(), Toast.LENGTH_SHORT).show();

        }

    }}{

        protected Map<String, String> getParams() {

            Map<String, String> paramV = new HashMap<>();

            paramV.put("img", base64Image);

            paramV.put("user_id", id_data_User);

            paramV.put("total_harga", HARGA_KESELURUHAN);

            return paramV;

        }

    };

    /**
     * dibawah ini digunakan untuk handle klo internet slow, jadi
     * dia timeoutnya di x 0
     *
     * retriesnya ga usah diulang, jdi ga ngebug, baru setelah itu
     * bisa request deh....
     */

    stringRequest.setRetryPolicy(new DefaultRetryPolicy(

        DefaultRetryPolicy.DEFAULT_TIMEOUT_MS * 0,

        DefaultRetryPolicy.DEFAULT_MAX_RETRIES -1,

        DefaultRetryPolicy.DEFAULT_BACKOFF_MULT ));

    Volley.newRequestQueue(this).add(stringRequest);

}

else {

    Toast.makeText(getApplicationContext(), "Select Image First! ",
Toast.LENGTH_SHORT).show();

    return;

}

```

```
}  
  
}
```

Code diatas digunakan user dalam melakukan checkout dalam pembelian buku yang mewajibkan user untuk mengisi alamat dan mengupload bukti pembayaran dan nantinya akan masuk kedalam database BOOKPEDIA.

2.2.3. Detail Book Activity

```
package com.example.buku.activity;  
  
import android.app.ProgressDialog;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.example.buku.Handler.Constant;  
import com.example.buku.Handler.RequestHandler;  
import com.example.buku.R;
```

```

import com.example.buku.dto.BookDTO;

import com.example.buku.fragment.HomeFragment;

import com.squareup.picasso.Picasso;

import org.json.JSONObject;

import java.util.HashMap;

import java.util.Map;


public class DetailBookActivity extends AppCompatActivity implements
View.OnClickListener{

    // Inisial awal buat txtview -/dkk

    ImageView imgBuku;

    TextView tvAuthor, tvJudulBuku, tvHargaBuku, tvDescBuku, tvJmlHal,
tvTglTerbit, tvIsbn, tvPenerbit, tvStock;

    Button btnAddCart;

    BookDTO bookDTO;

    int qty = 1;

    String id_buku, id_data_User, harga_buku;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_detail_book);

        // ambil intent data dari bookdto

        bookDTO = (BookDTO) getIntent().getSerializableExtra("data");

        inisialisasi();

        getSupportActionBar().hide();

        btnAddCart.setOnClickListener(this);

    }

```

```
private void inisialisasi(){ // inilisilisasi awal

    imgBuku = findViewById(R.id.img_buku);

    tvAuthor = findViewById(R.id.tv_author);

    tvJudulBuku = findViewById(R.id.tv_judul_buku);

    tvHargaBuku = findViewById(R.id.tv_harga_buku);

    tvDescBuku = findViewById(R.id.tv_desc_buku);

    tvJmlHal = findViewById(R.id.tv_jml_hal);

    tvTglTerbit = findViewById(R.id.tv_tgl_terbit);

    tvIsbn = findViewById(R.id.tv_isbn);

    tvPenerbit = findViewById(R.id.tv_penerbit);

    tvStock = findViewById(R.id.tv_stock);

    btnAddCart = findViewById(R.id.btn_add_cart);

    tvAuthor.setText(bookDTO.author);

    tvJudulBuku.setText(bookDTO.name);

    tvHargaBuku.setText(bookDTO.price);

    tvStock.setText(BookDTO.stock);

    // Picasso dipakai biar bisa load image, ini library yang keren

    Picasso.get().load(bookDTO.image).into(imgBuku);

    pengubahan();

    // set textnya ya bro dari book dto. apa apa , yang mau diambil
    datanya berdasarkan tempatnya

    tvDescBuku.setText(bookDTO.description);

    tvJmlHal.setText(bookDTO.total_pages);

    tvTglTerbit.setText(bookDTO.published_at);

    tvIsbn.setText(bookDTO.isbn);

    tvPenerbit.setText(bookDTO.rating);

}
```

```

public void pengubahan() {

    HomeActivity activity = new HomeActivity();

    // user_id, book_id, qty, price

    id_data_User = activity.getUserDATA_ID(); // buat id_user

    id_buku = bookDTO.id; // buat id_buku

    harga_buku = tvHargaBuku.getText().toString().trim(); // buat
harga_buku

}

public void memasukkan_kedalam_cart() {

    /**

        * progress dialog, display the progress of any running task or an
operation. An Alert Dialog is a type of alert

        * message displayed over the screen that lets users choose between
options to respond to the message of the alert.

    */

    final ProgressDialog progressDialog = new ProgressDialog(this);

    progressDialog.setMessage("Please Wait..");

    progressDialog.show();

    /** mulailah menggunakan library volley dibawah ini untuk menghubungkan
ke api lalu ke database,

        * dibawah ini menggunakan metode post, yang berarti metode HTTP
Request

        * yang digunakan untuk membuat data baru dengan menyisipkan data
dalam body saat request dilakukan

    */

    StringRequest request = new StringRequest(Request.Method.POST,
Constant.URL_ADD_CART_USER, new Response.Listener<String>() {

        @Override

        public void onResponse(String response) {

            try {

                JSONObject jObj = new JSONObject(response);

```

```

        // ambil dari database ( json)

        int sukses = jsonObj.getInt("success");

        String pesan = jsonObj.getString("message");

        if (sukses == -1) {

            progressDialog.dismiss();

            Toast.makeText(DetailBookActivity.this, pesan,
Toast.LENGTH_SHORT).show();

            return;

        }

        // Error in registration!  // user failed to store

        // jika gagal didaftarkan

        if (sukses == -2){

            progressDialog.dismiss();

            Toast.makeText(DetailBookActivity.this, pesan,
Toast.LENGTH_SHORT).show();

            return;

        }

        // jika ada kesalan dalam PENGISIAN KE DALAM CART

        if (sukses == -3) {

            progressDialog.dismiss();

            Toast.makeText(DetailBookActivity.this, pesan,
Toast.LENGTH_SHORT).show();

            return;

        }

        else{

            progressDialog.dismiss();

            Toast.makeText(DetailBookActivity.this, "Cool, you
have successfully added to your cart ! ", Toast.LENGTH_SHORT).show();

            // masukin ke sign in

```

```

        Intent i = new Intent(getApplicationContext(),
HomeActivity.class);

        startActivity(i);

        // menyatakan selesai

        finish();

    }

} catch (Exception ex) { // jika eror

    Log.e("Error: ", ex.toString());

    ex.printStackTrace();

}

}

}, new Response.ErrorListener() {

    @Override

    public void onErrorResponse(VolleyError error) { // jika eror

        Toast.makeText(DetailBookActivity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();

        progressDialog.dismiss();

        Log.e("Error: ", error.getMessage());

    }

}

) {

    /**

        * metode dibawah ini adalah yang digunakan untuk memasukkan data
imputan dari aplikasi kedalam api,

        * dengan mencocokkan pada apa yang telah dideklarasikan pada
methode post di api (php - nya), sehingga

        * bisa terhubung.... kedalam api dan bisa mennyampaikannya
kedalam database bahwa terdapat insert - baru

        */

    @Override

```

```

        protected Map<String, String> getParams() throws AuthFailureError
    {
        Map<String, String> params = new HashMap<>();

        params.put("user_id", id_data_User);

        params.put("book_id", id_buku);

        params.put("qty", String.valueOf(qty)); // default

        params.put("price", harga_buku);

        return params;

    }

};

// baru bisa dijalankan

RequestHandler.getInstance((Context) this).addToRequestQueue(request);

}

@Override

public void onClick(View v) { // karena dibawah kita implement view on
click listener

    if(v.getId()==R.id.btn_add_cart){

        try { // dihandle jika ade yang eror ya beb maka pake tangkep dan
coba

            memasukkan_kedalam_cart();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

}

```

Pada codingan ini untuk menampilkan detail pada buku. Bagian code yang digunakan ini kita menggunakan metode library picaso agar dapat melakukan load image. Kita juga menggunakan library volley untuk menghubungkan API lalu ke database, metode yang digunakan adalah POST , yang berarti menggunakan metode HTTP request. Metode selanjutnya dengan cara

mencocokkan pada apa yang telah dideklarasikan pada metode post pada API (php -nya), sehingga bisa terhubung ke dalam API dan bisa terhubung kedalam database bahwa terdapat insert baru.

2.2.4. HomeActivity

```
package com.example.buku.activity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.viewpager.widget.ViewPager;
import com.example.buku.R;
import com.example.buku.adapter.Session;
import com.example.buku.adapter.ViewPagerAdapter;
import com.example.buku.auth.SignInActivity;
import com.example.buku.databinding.ActivityHomeBinding;
import com.example.buku.fragment.CartFragment;
import com.example.buku.fragment.HistoryFragment;
import com.example.buku.fragment.HomeFragment;
import com.example.buku.fragment.ProfileFragment;
import com.google.android.material.bottomnavigation.BottomNavigationView;
```

```
import com.google.android.material.navigation.NavigationBarView;

public class HomeActivity extends AppCompatActivity implements
NavigationBarView.OnItemSelectedListener, ViewPager.OnPageChangeListener {

    // inilisaisiasi

    ViewPager pager;

    private ActivityHomeBinding binding;

    BottomNavigationView navView;

    // ambil dari session

    private String INTENT_NOMOR_HAPE =
Session.getInstance(this).getusernomer_Hape();

    private String INTENT_ID_USER = Session.getInstance(this).getKeyUserId();

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        binding = ActivityHomeBinding.inflate(getLayoutInflater());

        setContentView(binding.getRoot());

        if(!Session.getInstance(this).isLoggedIn()){

            finish();

            startActivity(new Intent(this, SignInActivity.class));

            return;

        }

        navView = findViewById(R.id.nav_view);

        navView.setOnItemSelectedListener(this);

        // Passing each menu ID as a set of Ids because each

        // menu should be considered as top level destinations.

        getSupportActionBar().hide();

        pager = findViewById(R.id.pager);

        pager.setOnPageChangeListener(this);
```

```

        ViewPagerAdapter adapter = new
ViewPagerAdapter(getSupportFragmentManager());

        adapter.addFragment(new HomeFragment());

        adapter.addFragment(new CartFragment());

        adapter.addFragment(new HistoryFragment());

        adapter.addFragment(new ProfileFragment());

        pager.setAdapter(adapter);

    }

    // BIAR INTENT BISA DIKIRIM

    public String getUserDATA_NOHAPE() {

        return INTENT_NOMOR_HAPE;

    }

    public String getUserDATA_ID() {

        return INTENT_ID_USER;

    }

    public void pencet(){ // ini method buat log out krn di profile manager
adalah fragment

        Session.getInstance((this)).logout();

        finish();

        startActivity(new Intent(this, SignInActivity.class));

    }

    public void tombol_map(){ // intent (PINDAH TEMPAT BEB), KE MAP

        startActivity(new Intent(this, MapsActivity.class));

    }

    public void tombol_webview(){ // intent (PINDAH TEMPAT BEB), KE WEBPIEW

        startActivity(new Intent(this, WebViewActivity.class));

```

```

}

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

    switch (menuItem.getItemId()) {

        case R.id.navigation_home:

            pager.setCurrentItem(0);

            break;

        case R.id.navigation_cart:

            pager.setCurrentItem(1);

            break;

        case R.id.navigation_history:

            pager.setCurrentItem(2);

            break;

        case R.id.navigation_profile:

            pager.setCurrentItem(3);

            break;

    }

    return true;

}

@Override

public void onPageScrolled(int position, float positionOffset, int
positionOffsetPixels) {

    switch (position) {

        case 0:

            navView.getMenu().findItem(R.id.navigation_home).setChecked(true);

            break;

        case 1:

```

```

navView.getMenu().findItem(R.id.navigation_cart).setChecked(true);

        break;

        case 2:

navView.getMenu().findItem(R.id.navigation_history).setChecked(true);

        break;

        case 3:

navView.getMenu().findItem(R.id.navigation_profile).setChecked(true);

        break;

    }

}

@Override

public void onPageSelected(int position) {

}

@Override

public void onPageScrollStateChanged(int state) {

}

}

```

Pada activity ini juga dilakukan inisialisasi dan beberapa method lainnya. selain itu, terdapat pengambilan session berupa nomor hp dan id pengguna. binding merupakan hal yang sama dengan findviewbyid, perbedaannya adalah binding tidak perlu menginisialisasikannya satu per satu. pada activity ini juga terdapat code untuk melakukan View Pager dan navigasi agar dapat melakukan pemindahan halaman dari 1 page ke page lainnya.

2.2.5. SearchActivity

```

package com.example.buku.activity;

import static com.example.buku.Handler.Constant.URL_NEW_BOOK;

```

```
import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.SearchView;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.Request;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import com.example.buku.R;

import com.example.buku.adapter.NewBookAdapter;

import com.example.buku.adapter.SearchListAdapter;

import com.example.buku.dto.BookDTO;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;

import java.util.List;

public class SearchActivity extends AppCompatActivity {

    // DEKLARASI AWAL

    RecyclerView recyclerView;

    SearchView searchView;

    List<BookDTO> data;
```

```

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_search);

    // SAMA DEKLARASI AWAL

    recyclerView = findViewById(R.id.search_book);

    recyclerView.setLayoutManager(new LinearLayoutManager(this));

    loadData();

    getSupportActionBar().hide();

    // set up the SearchView

    searchView = findViewById(R.id.searchView);

    searchView.clearFocus();

    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener()
{

    @Override

    public boolean onQueryTextSubmit(String query) {

        return false;

    }

    @Override

    public boolean onQueryTextChange(String newText) {

        filterList(newText);

        return true;

    }

    });

}

private void filterList(String searchQuery) { // ambil data dari arraylist
bookdto class

    ArrayList<BookDTO> filteredData = new ArrayList<>();

```

```

        for (BookDTO item : data) {

            if (item.name.toLowerCase().contains(searchQuery.toLowerCase())){

                filteredData.add(item);

            }

        }

        ArrayList<String> newData = new ArrayList<>();

        for (BookDTO item : filteredData) {

            newData.add(item.name);

        }

        SearchListAdapter adapter = new SearchListAdapter(this, newData);

        adapter.setOnClickListener(new SearchListAdapter.ItemClickListener() {

            @Override

            public void onItemClick(View view, int position) { // ketika di
tekan itu searchnya maka dia akan intent data sesuai posisi

                Intent i = new Intent(view.getContext(),
DetailBookActivity.class);

                i.putExtra("data", filteredData.get(position));

                startActivity(i);

            }

        });

        recyclerView.setAdapter(adapter);

    }

    void loadData() {

        // data to populate the RecyclerView with

        // data to populate the RecyclerView with

        // to do: ambil data dari API

        data = new ArrayList<BookDTO>();

        StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL_NEW_BOOK,

```



```

        new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                try {

                    //converting the string to json array object

                    JSONArray array = new JSONArray(response);

                    //traversing through all the object

                    for (int i = 0; i < array.length(); i++) {

                        //getting product object from json array

                        JSONObject product = array.getJSONObject(i);

                        //adding the product to product list

                        data.add(new BookDTO(product));

                    }

                } catch (JSONException e) {

                    e.printStackTrace();

                }

            }

        },

        new Response.ErrorListener() {

            @Override

            public void onErrorResponse(VolleyError error) {

                Log.e("Error: ", error.getMessage());

            }

        });

// sama buat volley request

Volley.newRequestQueue(this).add(stringRequest);

ArrayList<String> newData = new ArrayList<>();

for (BookDTO item : data) {

```

```

        newData.add(item.name);

    }

    SearchListAdapter adapter = new SearchListAdapter(this, newData);

    adapter.setOnClickListener(new SearchListAdapter.ItemClickListener() {

        @Override

        public void onItemClick(View view, int position) {

            Intent i = new Intent(view.getContext(),
DetailBookActivity.class);

            i.putExtra("data",data.get(position));

            startActivity(i);

        }

    });

    recyclerView.setAdapter(adapter);

}
}

```

Code diatas digunakan oleh pengguna untuk melakukan Search pada aplikasi. Mula mula dideklarasikan terlebih dahulu Searchview, recycleView dan juga data. Code recycleview digunakan untuk bisa terjadinya recycle pada UI ketika pengguna sedang melakukan scrolling pada aplikasi. Code searchView digunakan untuk melakukan search pada data BookDTO dengan cara memfilter data yang sesuai denga String yang dimasukan

2.2.6. WebView Activity

```

package com.example.buku.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import com.example.buku.Handler.Constant;

import com.example.buku.R;

```

```

public class WebViewActivity extends AppCompatActivity {

    private WebView webView;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_web_view);

        // ini class buat web view

        webView = (WebView) findViewById(R.id.webview);

        webView.setWebViewClient(new WebViewClient());

        // diambil dari Class Constant

        webView.loadUrl(Constant.url_web_view);

        getSupportActionBar().hide();

    }

}

```

Code diatas digunakan untuk melakukan WebView dimana code akan membuka website berdasarkan URL yang dicantumkan.

2.2.7. Maps Activity

```

package com.example.buku.activity;

import androidx.fragment.app.FragmentActivity;

import android.os.Bundle;

import com.example.buku.R;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

```

```

import com.google.android.gms.maps.model.MarkerOptions;

    public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {

        // YA INI ADALAH MAP LAH MAU APALAGI

        private GoogleMap mMap;

        @Override

        protected void onCreate(Bundle savedInstanceState) {

            super.onCreate(savedInstanceState);

            setContentView(R.layout.activity_maps);

            // Obtain the SupportMapFragment and get notified when the map is
            ready to be used.

            SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager()

                .findFragmentById(R.id.map);

            mapFragment.getMapAsync(this);
        }

        /**
         * Manipulates the map once available.
         * This callback is triggered when the map is ready to be used.
         * This is where we can add markers or lines, add listeners or move the
         camera. In this case,
         * we just add a marker near Sydney, Australia.
         * If Google Play services is not installed on the device, the user will
         be prompted to install
         * it inside the SupportMapFragment. This method will only be triggered
         once the user has
         * installed Google Play services and returned to the app.
         */
        @Override

        public void onMapReady(GoogleMap googleMap) {

            mMap = googleMap;

```

```

// Add a marker in Sydney and move the camera

LatLng appLoc = new LatLng(-6.257385, 106.618320);

LatLng apptong = new LatLng(-6.2569947235539285, 106.61683238284874);

mMap.addMarker(new MarkerOptions().position(appLoc).title("Our
Location"));

mMap.addMarker(new MarkerOptions().position(apptong).title("Our IT
Office"));

mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(appLoc,16));

mMap.setTrafficEnabled(true);

}
}

```

Codingan ini berfungsi untuk menampilkan Maps pada perusahaan BUKUPEDIA. Mendeklarasikan Longitude dan Latitude sebagai penanda lokasi maps yang diperlukan.

2.2.8. HistoryItemAdapter

```

package com.example.buku.adapter;

import android.content.Context;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;
import com.example.buku.R;

```

```
import com.example.buku.dto.BookHistoryDTO;

import com.squareup.picasso.Picasso;

import java.util.ArrayList;

import java.util.List;

public class HistoryItemAdapter extends
RecyclerView.Adapter<HistoryItemAdapter.ViewHolder> implements Filterable {

    private List<BookHistoryDTO> mData;

    private LayoutInflater mInflater;

    private HistoryItemAdapter.ItemClickListener mClickListener;

    // data is passed into the constructor

    public HistoryItemAdapter(Context context, ArrayList<BookHistoryDTO> data)
    {

        this.mInflater = LayoutInflater.from(context);

        this.mData = data;

    }

    // inflates the row layout from xml when needed

    @Override

    public HistoryItemAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,
int viewType) {

        View view = mInflater.inflate(R.layout.fragment_book_item_history,
parent, false);

        return new HistoryItemAdapter.ViewHolder(view);

    }

    // binds the data to the TextView in each row

    @Override

    public void onBindViewHolder(ViewHolder holder, int position) {

        BookHistoryDTO data = mData.get(position);

        Picasso.get().load(data.image).into(holder.imgBuku);

        holder.tvHargaBuku.setText("Rp. " +data.total_price+ ".000");

        holder.tvJudulBuku.setText(data.name);

    }

}
```

```

        holder.tvquantity.setText(data.qty);

        holder.tvstatus.setText(data.status);

        int conditionColor;

        if(data.status.equals("1")){

            holder.tvstatus.setText("Pending");

            conditionColor = Color.parseColor("#FB7181");

        } else{

            holder.tvstatus.setText("Payment Successful");

            conditionColor = Color.parseColor("#4BB543");

        }

        holder.tvstatus.setTextColor(conditionColor);

    }

    // total number of rows

    @Override

    public int getItemCount() {

        return mData.size();

    }

    @Override

    public Filter getFilter() {

        return null;

    }

    // stores and recycles views as they are scrolled off screen

    public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {

        ImageView imgBuku;

        TextView tvAuthor, tvHargaBuku, tvstatus, tvquantity, tvJudulBuku;

        ViewHolder(View itemView) {

            super(itemView);

```

```

        imgBuku = itemView.findViewById(R.id.img_buku);

        tvJudulBuku = itemView.findViewById(R.id.tv_judul_buku);

        tvHargaBuku = itemView.findViewById(R.id.tv_harga_buku);

        tvstatus = itemView.findViewById(R.id.tv_status);

        tvquantity = itemView.findViewById(R.id.tv_qty);

        itemView.setOnClickListener(this);

    }

    @Override

    public void onClick(View view) {

        if (mClickListener != null) mClickListener.onItemClick(view,
getAdapterPosition());

    }

}

// allows clicks events to be caught

public void setClickListener(HistoryItemAdapter.ItemClickListener
itemClickListener) {

    this.mClickListener = itemClickListener;

}

// parent activity will implement this method to respond to click events

public interface ItemClickListener {

    void onItemClick(View view, int position);

}

}

```

Code diatas merupakan code adapter untuk menampilkan history pembelian buku yang sudah dilakukan oleh user dengan menggunakan recycle view yang akan menampilkan kolom buku.

2.2.9. ItemCartAdapter

```
package com.example.buku.adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;
import com.example.buku.R;
import com.example.buku.dto.BookCartDTO;
import com.squareup.picasso.Picasso;
import java.util.List;

public class ItemCartAdapter extends
RecyclerView.Adapter<ItemCartAdapter.ViewHolder> implements Filterable {

    private List<BookCartDTO> mData;

    private LayoutInflater mInflater;

    private ListItemClickListener mClickListener;

    public interface ListItemClickListener {

        void on_plus_click(BookCartDTO item);

        void on_min_click(BookCartDTO item);

        void on_hapus_cart(BookCartDTO item);
    }
}
```

```

    }

    // data is passed into the constructor

    public ItemCartAdapter(Context context, List<BookCartDTO> data,
ListItemClickListener mClickListener) {

        this.mInflater = LayoutInflater.from(context);

        this.mData = data;

        this.mClickListener = mClickListener;

    }

    // inflates the row layout from xml when needed

    @Override

    public ItemCartAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,
int viewType) {

        View view = mInflater.inflate(R.layout.layout_cart_item, parent,
false);

        return new ItemCartAdapter.ViewHolder(view);

    }

    // binds the data to the TextView in each row

    @Override

    public void onBindViewHolder(ItemCartAdapter.ViewHolder holder, int
position) {

        BookCartDTO data = mData.get(position);

        Picasso.get().load(data.image).into(holder.imgBuku);

        holder.tvJudulBuku.setText(data.name);

        holder.tvHargaBuku.setText(data.price);

        holder.tvHargaBuku.setText("Rp. "+ data.price+"0");

        holder.edtQty.setText(data.qty);

        // holder.TV_ID_CART_USER.setText(data.id_cart);

        holder.btnHapus.setOnClickListener(new View.OnClickListener() {

            @Override

```

```

        public void onClick(View v) {

            mClickListener.on_hapus_cart(data);

        }

    });

    holder.btnPlus.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            mClickListener.on_plus_click(data);

        }

    });

    holder.btnMin.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            mClickListener.on_min_click(data);

        }

    });

}

// total number of rows

@Override

public int getItemCount() {

    return mData.size();

}

@Override

public Filter getFilter() {

    return null;

}

// stores and recycles views as they are scrolled off screen

public class ViewHolder extends RecyclerView.ViewHolder {

```

```

        ImageView imgBuku;

        TextView tvJudulBuku, tvHargaBuku, TV_ID_CART_USER;

        Button btnMin, btnPlus, btnHapus;

        EditText edtQty;

        ViewHolder(View itemView) {

            super(itemView);

            imgBuku = itemView.findViewById(R.id.img_buku);

            tvJudulBuku = itemView.findViewById(R.id.tv_judul_buku);

            tvHargaBuku = itemView.findViewById(R.id.tv_harga_buku);

            btnMin = itemView.findViewById(R.id.btn_min);

            btnPlus = itemView.findViewById(R.id.btn_plus);

            btnHapus = itemView.findViewById(R.id.btn_hapus);

            edtQty = itemView.findViewById(R.id.edt_qty);

            TV_ID_CART_USER = itemView.findViewById(R.id.ID_BUKU_CART);

        }

    }

}

```

Code diatas merupakan code adapter untuk menampilkan cart pembelian buku yang terdapat pada cart user dengan cara menampilkan recycle view. Recycle view ini nantinya akan menampilkan list pembelian buku oleh user. selain itu, pada list ini juga ditambahkan fitur button untuk melakukan penambahan, pengurangan dan penghapusan buku pada cart.

2.2.10. NewBookAdapter

```

package com.example.buku.adapter;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

```

```
import android.view.ViewGroup;

import android.widget.Filter;

import android.widget.Filterable;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.buku.R;

import com.example.buku.dto.BookDTO;

import com.squareup.picasso.Picasso;

import java.util.List;

public class NewBookAdapter extends
RecyclerView.Adapter<NewBookAdapter.ViewHolder> implements Filterable {

    private List<BookDTO> mData;

    private LayoutInflater mInflater;

    private NewBookAdapter.ItemClickListener mClickListener;

    // data is passed into the constructor

    public NewBookAdapter(Context context, List<BookDTO> data) {

        this.mInflater = LayoutInflater.from(context);

        this.mData = data;

    }

    // inflates the row layout from xml when needed

    @Override

    public NewBookAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {

        View view = mInflater.inflate(R.layout.layout_book_item, parent,
false);

        return new NewBookAdapter.ViewHolder(view);

    }

}
```

```

// binds the data to the TextView in each row

@Override

    public void onBindViewHolder(NewBookAdapter.ViewHolder holder, int
position) {

        BookDTO data = mData.get(position);

        Picasso.get().load(data.image).into(holder.img_buku);

        holder.tv_author.setText(data.author);

        holder.tv_judul_buku.setText(data.name);

        holder.tv_harga_buku.setText(data.price);

        holder.tv_harga_buku.setText("Rp. " + data.price+"0");

    }

// total number of rows

@Override

public int getItemCount() {

    return mData.size();

}

@Override

public Filter getFilter() {

    return null;

}

// stores and recycles views as they are scrolled off screen

public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {

    TextView tv_author, tv_judul_buku, tv_harga_buku;

    ImageView img_buku;

    ViewHolder(View itemView) {

        super(itemView);

        tv_author = itemView.findViewById(R.id.tv_author);

```

```

        tv_judul_buku = itemView.findViewById(R.id.tv_judul_buku);

        tv_harga_buku = itemView.findViewById(R.id.tv_harga_buku);

        img_buku = itemView.findViewById(R.id.img_buku);

        itemView.setOnClickListener(this);

    }

    @Override

    public void onClick(View view) {

        if (mClickListener != null) mClickListener.onItemClick(view,
getAdapterPosition());

    }

}

// allows clicks events to be caught

public void setClickListener(NewBookAdapter.ItemClickListener
itemClickListener) {

    this.mClickListener = itemClickListener;

}

// parent activity will implement this method to respond to click events

public interface ItemClickListener {

    void onItemClick(View view, int position);

}

}

```

Code diatas merupakan adapter untuk menampilkan buku buku yang nantinya muncul di Home screen BOOKPEDIA dengan menggunakan recycle view.

2.2.11. SearchListAdapter

```

package com.example.buku.adapter;

import android.content.Context;

```

```
import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Filter;

import android.widget.Filterable;

import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.buku.R;

import java.util.List;

public class SearchListAdapter extends
RecyclerView.Adapter<SearchListAdapter.ViewHolder> implements Filterable {

    private List<String> mData;

    private LayoutInflater mInflater;

    private ItemClickListener mClickListener;

    // data is passed into the constructor

    public SearchListAdapter(Context context, List<String> data) {

        this.mInflater = LayoutInflater.from(context);

        this.mData = data;

    }

    // inflates the row layout from xml when needed

    @Override

    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View view = mInflater.inflate(R.layout.search_list_tile, parent,
false);

        return new ViewHolder(view);

    }

    // binds the data to the TextView in each row

    @Override

    public void onBindViewHolder(ViewHolder holder, int position) {
```



```

        String data = mData.get(position);

        holder.myTextView.setText(data);

    }

    // total number of rows

    @Override

    public int getItemCount() {

        return mData.size();

    }

    @Override

    public Filter getFilter() {

        return null;

    }

    // stores and recycles views as they are scrolled off screen

    public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {

        TextView myTextView;

        ViewHolder(View itemView) {

            super(itemView);

            myTextView = itemView.findViewById(R.id.search_result_text);

            itemView.setOnClickListener(this);

        }

        @Override

        public void onClick(View view) {

            if (mClickListener != null) mClickListener.onItemClick(view,
getAdapterPosition());

        }

    }

    // allows clicks events to be caught

    public void setClickListener(ItemClickListener itemClickListener) {

```

```

        this.mClickListener = itemClickListener;
    }

    // parent activity will implement this method to respond to click events
    public interface ItemClickListener {

        void onItemClick(View view, int position);

    }
}

```

Code diatas merupakan adapter untuk melakukan pencarian buku pada bagian search dengan menggunakan recycle view yang akan menampilkan list buku berdasarkan ketikan user.

2.2.12. Session

```

package com.example.buku.adapter;

import android.content.Context;
import android.content.SharedPreferences;
import android.util.Log;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;
import com.example.buku.Handler.RequestHandler;

/**
 * Menyimpan sesi login agar kapanpun aplikasi dibuka tetap login sebagai user
 * masing-masing
 */
public class Session {

    private static Session instance;

    private static final String SHARED_PREF_NAME = "mysharedpref123";

    public static final String KEY_NAME = "USER_NAME";

    public static final String KEY_EMAIL = "USER_EMAIL";

```

```

public static final String KEY_USER_ID = "USER_ID";

public static final String KEY_USER_NOHP = "USER_NOHP";

private static Context ctx;

private Session(Context context) {

    ctx = context;

}

public static synchronized Session getInstance(Context context) {

    if (instance == null) {

        instance = new Session(context);

    }

    return instance;

}

public boolean userlogin (String id, String namee, String email, String
nohp ){

    /** jadi kalo login, data yang masuk di session ya

    * id, name, email, nohp

    */

    SharedPreferences sharedPreferences
=ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

    SharedPreferences.Editor editor = sharedPreferences.edit();

    editor.putString(KEY_USER_ID, id);

    editor.putString(KEY_NAME, namee);

    editor.putString(KEY_EMAIL, email);

    editor.putString(KEY_USER_NOHP, nohp);

    editor.apply();

    return true;

}

public boolean userUpdated (String nohp ){ // u[date nomer hape ae

```

```

        SharedPreferences sharedPreferences
=ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        SharedPreferences.Editor editor = sharedPreferences.edit();

        editor.putString(KEY_USER_NOHP, nohp);

        editor.apply();

        return true;

    }

    public boolean isLoggedIn (){ // pengecekan apabila user udah login apa lom

        SharedPreferences sharedPreferences =
ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        if(sharedPreferences.getString(KEY_NAME, null) != null){

            return true;

        }

        return false;

    }

    public boolean logout (){ // digunakan untuk log out dri session

        SharedPreferences sharedPreferences =
ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        SharedPreferences.Editor editor = sharedPreferences.edit();

        editor.clear();

        editor.apply();

        return true;

    }

    /**
     * dibawah ini digunakan untuk mengambil data jadi di get apa datanya,
     * lalu rertunr apa yang diminta dari session yang
     * udah di insert sebelumnya
     * @return
     */

    public String getUsername() {

```

```

        SharedPreferences sharedPreferences =
ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        return sharedPreferences.getString(KEY_NAME, null);

    }

    public String getusernomer_Hape() {

        SharedPreferences sharedPreferences =
ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        return sharedPreferences.getString(KEY_USER_NOHP, null);

    }

    public String getEmail_USER() {

        SharedPreferences sharedPreferences =
ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        return sharedPreferences.getString(KEY_EMAIL, null);

    }

    public String getKeyUserId(){

        SharedPreferences sharedPreferences =
ctx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);

        return sharedPreferences.getString(KEY_USER_ID, null);

    }

}

```

code diatas merupakan code untuk menahan sebuah session user jika user sedang Log In atau tidak. Fungsi dari code diatas adalah untuk menahan sesion aplikasi baik setelah user melakukan log in maupun belum melakukan log in.

2.2.13. User

```

package com.example.buku.adapter;

public class User {

    private String username, email, nohp;

    public User(String username, String email, String nohp) {

```

```

        this.username = username;

        this.email = email;

        this.nohp = nohp;
    }

    public String getUsername() {

        return username;
    }

    public void setUsername(String username) {

        this.username = username;
    }

    public String getEmail() {

        return email;
    }

    public void setEmail(String email) {

        this.email = email;
    }

    public String getNohp() {

        return nohp;
    }

    public void setNohp(String nohp) {

        this.nohp = nohp;
    }
}

```

Codingan ini berfungsi untuk menampung data pada user

2.2.14. ViewPagerAdapter

```

package com.example.buku.adapter;

```

```

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentManager;

import androidx.fragment.app.FragmentPagerAdapter;

import androidx.lifecycle.Lifecycle;

import java.util.ArrayList;

public class ViewPagerAdapter extends FragmentPagerAdapter {

    // ini adalah view pager adapter, agar bisa geser gserrrrrrrrr

    private ArrayList<Fragment> fragmentList = new ArrayList<>();

    public ViewPagerAdapter(@NonNull FragmentManager fm) {

        super(fm);

    }

    public void addFragment(Fragment fragment) {

        fragmentList.add(fragment);

    }

    @NonNull

    @Override

    public Fragment getItem(int position) {

        return fragmentList.get(position);

    }

    @Override

    public int getCount() {

        return fragmentList.size();

    }

}

```

Codingan ini berfungsi untuk melakukan penggeseran pada dari layout satu ke layout lain-nya

2.2.15. SignInActivity

```
package com.example.buku.auth;

import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.buku.Handler.Constant;
import com.example.buku.Handler.RequestHandler;
import com.example.buku.Handler.SQLite;
import com.example.buku.R;
import com.example.buku.activity.HomeActivity;
import com.example.buku.adapter.Session;
import com.google.android.material.textfield.TextInputEditText;
```



```

import org.json.JSONObject;

import java.util.HashMap;

import java.util.Map;

import java.util.regex.Pattern;

public class SignInActivity extends AppCompatActivity {

    // Inisial awal buat text-piew -/dkk

    Button btnSignIn;

    TextInputEditText edtEmail, edtPassword;

    TextView tvBtnSignup;

    TextView tvErrorTextLogin;

    String email, passwordw;

    String pesan;

    // pembatasan buat nama

    final Pattern NAMA_PATTERN =

        Pattern.compile("^" +

            ".{5,}" //at least 5 characters

        );

    // pembatasan buat password

    final Pattern PASSWORD_PATTERN =

        Pattern.compile("^" +

            "(?=.*[0-9])" + //at least 1 digit

            "(?=.*[a-z])" + //at least 1 lower case letter

            "(?=.*[A-Z])" + //at least 1 upper case letter

            "(?=.*[a-zA-Z])" + //any letter

            "(?=.*[!@#$%^&+=])" + //at least 1 special character

            "(?=\\S+$)" + //no white spaces

```

```

        ".{8,}" + //at least 8 characters
        "$");

private SQLite dbsqlite;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_sign_in);

    // SQLite database handler

    dbsqlite = new SQLite(getApplicationContext());

    /**
     * pengecekan dilakukan agar tau apakah user ini login apa belum,
     jadi klo udah login ia akan ke home activity
     *
     * sebaliknya maka dia akan steyp di activity ini untuk log - in
     terlebih dahulu
     */

    if(Session.getInstance(this).isLoggedIn()){

        finish();

        startActivity(new Intent(this, HomeActivity.class));

        // return;

    }

    inisialisasi();

    // ngehilangin support action barnya

    getSupportActionBar().hide();

    Sign_up_Button();

}

private void inisialisasi() {

    btnSignIn = findViewById(R.id.btn_sign_in); //ga gw pake karena gw
lanngsung pake di xml ( Sign_in)

```

```

        edtEmail = findViewById(R.id.edt_email);

        edtPassword = findViewById(R.id.edt_password);

        tvBtnSignup = findViewById(R.id.tv_btn_signup);

        tvErrorTextLogin = findViewById(R.id.tv_errortext_login);

    }

    public void Sign_In (View view) {

        // PENAMPUNGAN VARIABLE YANG TELAH DIRUBAH DIBAWAH

        pengubahan();

        // JIKA EMAIL KOSONG MAKA DIA TURUN MENEGAMBIL CLASS YANG MENGURUSNYA
        DIBAWAH

        if (email.isEmpty()) {

            LoginEmail(edtEmail); // ambil kebawah

            return;

        }

        // JIKA EMAIL/username TIDAK SESUAI DENGAN KETENTUAN bagaimana dia
        login menggunakan fullname/email

        // seperti @BLABLA.COM DIA AKAN MENGELUARKAN NOTIF

        if (! NAMA_PATTERN.matcher(email).matches()) {

            tvErrorTextLogin.setText("The email address or full name you
            entered is invalid !");

            edtEmail.requestFocus();

            return;

        }

        // JIKA PASSWORD TIDAK SESUAI DENGAN KETENTUAN YAITU DIKIT ISINYA
        (MINIMAL 8 KARAKTER)

        // MAKA DIA EROR (MENAMPILKAN PESAN)

        if (edtPassword.length() < 1) {

            tvErrorTextLogin.setText("Password field is required!");

            edtPassword.requestFocus();

```

```

        return;

    }

    // JIKA PASSWROD TIDAK SESUAI DENGAN KETENTUAN DIA NGACO ISINYA

    //MAKA PESAN NOTIF AKAN KELUAR

    if (! (PASSWORD_PATTERN).matcher(passwordw).matches()){

        tvErrorTextLogin.setText("At least with 8 Characters," +

            " 1 Lower case and 1 Upper case and have 1 Special

character in your Password !");

        edtPassword.requestFocus();

    }

    // jika sqlite belum memiliki data berupa email dan password ini, dia
akan menjalankan ke public void menjalankan aksi 1

    if(!dbsqlite.checkUser(email, passwordw)){

        menjalankan_aksi1();

    }

    else {

        // jika memiliki data berupa email dan password ini, dia akan
menjalankan ke public void menjalankan aksi 2

        menjalankan_aksi2();

    }

}

public void menjalankan_aksi1(){ // kalo sqlite blom ada data email dn
password

    /**

        * progress dialog, display the progress of any running task or an
operation. An Alert Dialog is a type of alert

        * message displayed over the screen that lets users choose between
options to respond to the message of the alert.

    */

    final ProgressDialog progressDialog = new ProgressDialog(this);

    progressDialog.setMessage("Please Wait..");

```

```

progressDialog.show();

pengubahan();

/** mulailah menggunakan library volley dibawah ini untuk menghubungkan
ke api lalu ke database,

* dibawah ini menggunakan metode post, yang berarti metode HTTP
Request

* yang digunakan untuk membuat data baru dengan menyisipkan data
dalam body saat request dilakukan

*/

StringRequest request = new StringRequest(Request.Method.POST,
Constant.URL_LOGIN, new Response.Listener<String>() {

@Override

public void onResponse(String response) {

try {

JSONObject jsonObj = new JSONObject(response);

// mengambil json object untuk mendapatkan nilai dari api
yang telah dibuat

String namauser = jsonObj.getString("namee"); // ambil dari
database ( json)

String emailuser = jsonObj.getString("email");

String nohpuser = jsonObj.getString("nohp");

String id_user = jsonObj.getString("id_user123"); // ambil
dari database ( json)

int sukses = jsonObj.getInt("success");

pesan = jsonObj.getString("message");

if (sukses == -1){ // jika respon - 1 dia akan keluarin
pesan dri json

progressDialog.dismiss();

tvErrorTextLogin.setText(pesan);

edtEmail.requestFocus();

edtPassword.requestFocus();

return;

```

```

    }

    if (sukses == -2) { // sama

        progressDialog.dismiss();

        tvErrorTextLogin.setText(pesan);

        edtEmail.requestFocus();

        edtPassword.requestFocus();

        return;

    }

    if (sukses == 1) { // ini jika berhasil maka responnya
adalah 1 dari json dan akan toast "apa"

        progressDialog.dismiss();

        edtEmail.setText("");

        edtPassword.setText("");

        tvErrorTextLogin.setText("");

        Toast.makeText(SignInActivity.this, pesan,
Toast.LENGTH_SHORT).show();

        /**
         *   masukin data kedalam sqlite
         */

        dbsqlite.addUser(namauser, emailuser, passwordw);

        // masukin ke home page

        Intent i = new Intent(getApplicationContext(),
HomeActivity.class);

        startActivity(i);

        // masukin session ye ges !

        Session.getInstance(getApplicationContext())

            .userlogin(

                id_user, namauser, emailuser, nohpuser

```

```

        );

        // menyatakan selesai

        finish();

    }

} catch (Exception ex) { // jika eror

    progressDialog.dismiss();

    tvErrorTextLogin.setText("Login not matched. Please try
again!");

    Log.e("Error: ", ex.toString());

    ex.printStackTrace();

    Log.e("anyText", response);

}

}

}, new Response.ErrorListener() {

    @Override

    public void onErrorResponse(VolleyError error) { // jika eror

        Log.e("Error: ", error.getMessage());

        Toast.makeText(SignInActivity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();

        progressDialog.dismiss();

    }

}

) {

    /**

        * metode dibawah ini adalah yang digunakan untuk memasukkan data
imputan dari aplikasi kedalam api,

        * dengan mencocokkan pada apa yang telah dideklarasikan pada
methode post di api (php - nya), sehingga

```

```

        * bisa terhubung.... kedalam api dan bisa mennyampaikannya
        kedalam database bahwa terdapat insert - baru

        */

@Override

protected Map<String, String> getParams() throws AuthFailureError
{

    Map<String, String> params = new HashMap<>();

    // kondisi dibawah ini digunakan apabila login dengan
menggunakan email

    if (Patterns.EMAIL_ADDRESS.matcher(email).matches()) {

        params.put("email",email);

        params.put("password", passwordw);

    }

    else {

        // kondisi dibawah ini digunakan apabila login dengan
menggunakan nama panjang / username

        params.put("name", email);

        params.put("password", passwordw);

    }

    return params;

}

};

// baru di request deh

RequestHandler.getInstance(this).addToRequestQueue(request);

}

public void menjalankan_aksi2(){// kalo sqlite blom ada data email dn
password

/**

    * progress dialog, display the progress of any running task or an
operation. An Alert Dialog is a type of alert

```



```

        edtEmail.requestFocus();

        edtPassword.requestFocus();

        return;
    }

    if (sukses == -2){ // sama belum bisa

        progressDialog.dismiss();

        tvErrorTextLogin.setText(pesan);

        edtEmail.requestFocus();

        edtPassword.requestFocus();

        return;
    }

    if (sukses == 1) { // ini respon yang berhasil

        progressDialog.dismiss();

        edtEmail.setText("");

        edtPassword.setText("");

        tvErrorTextLogin.setText("");

        Toast.makeText(SignInActivity.this, pesan,
Toast.LENGTH_SHORT).show();

        // masukin ke home page

        Intent i = new Intent(getApplicationContext(),
HomeActivity.class);

        startActivity(i);

        // masukin session ye ges !

        Session.getInstance(getApplicationContext())

            .userlogin(

                id_user, namauser, emailuser, nohpuser

            );

        // menyatakan selesai

        finish();
    }

```

```

    }

    } catch (Exception ex) {

        progressDialog.dismiss();

        tvErrorTextLogin.setText("Login not matched. Please try
again!");

        Log.e("Error: ", ex.toString());

        ex.printStackTrace();

        Log.e("anyText", response);

    }

}

}, new Response.ErrorListener() {

    @Override

    public void onErrorResponse(VolleyError error) {

        Log.e("Error: ", error.getMessage());

        Toast.makeText(SignInActivity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();

        progressDialog.dismiss();

    }

}

) {

    /**

        * metode dibawah ini adalah yang digunakan untuk memasukkan data
imputan dari aplikasi kedalam api,

        * dengan mencocokkan pada apa yang telah dideklarasikan pada
methode post di api (php - nya), sehingga

        * bisa terhubung.... kedalam api dan bisa mennyampaikannya
kedalam database bahwa terdapat insert - baru

    */

    @Override

    protected Map<String, String> getParams() throws AuthFailureError
{

```

```

        Map<String, String> params = new HashMap<>();

        // kondisi dibawah ini digunakan apabila login dengan
        menggunakan email

        if (Patterns.EMAIL_ADDRESS.matcher(email).matches()) {

            params.put("email", email);

            params.put("password", passwordw);

        }

        else {

            // kondisi dibawah ini digunakan apabila login dengan
            menggunakan nama panjang / username

            params.put("name", email);

            params.put("password", passwordw);

        }

        return params;

    }

};

RequestHandler.getInstance(this).addToRequestQueue(request);

}

private boolean pengubahan() {

    email = edtEmail.getText().toString().trim();

    passwordw = edtPassword.getText().toString().trim();

    return false;

}

private boolean LoginEmail(EditText edtEmail) {

    pengubahan();

    if (email.length() < 5) {

        tvErrorTextLogin.setText("Email field is required ! ");

        edtEmail.requestFocus();

        return false;
    }
}

```

```

    }

    else {

        tvErrorTextLogin.setText("");

        edtEmail.requestFocus();

        return true;

    }

}

private void Sign_up_Button() {

    tvBtnSignup.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            Intent i = new Intent(getApplicationContext(), SignUpActivity.class);

            startActivity(i);

        }

    });

}

}

```

Code ini berfungsi untuk memasukan data ke user ke dalam SQLite, melakukan pengecekan pada inputan user kedalam database dan jika user berhasil maka akan lanjut ke proses selanjutnya yang menampung data user pada session yang menuju ke HomePage BUKUPEDIA.

2.2.16. SignUpActivity

```

package com.example.buku.auth;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;

```

```
import android.util.Log;

import android.util.Patterns;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.example.buku.Handler.Constant;

import com.example.buku.Handler.RequestHandler;

import com.example.buku.Handler.SQLite;

import com.example.buku.R;

import com.example.buku.adapter.User;

import org.json.JSONObject;

import java.util.HashMap;

import java.util.Map;

import java.util.regex.Pattern;

public class SignUpActivity extends AppCompatActivity {

    // iniliasi awal

    Button btnSignUp;

    TextView tvBtnSignIn;

    TextView tvErrorText;

    EditText edtName, edtEmail, edtPass, edtPhone;
```

```

String nama, email, passwordw, nohp;

// pembatasan buat nama

final Pattern NAMA_PATTERN =

    Pattern.compile("^" +

        "{5,}"          //at least 5 characters

    );

// pembatasan buat no telepon

final Pattern telepon_PATTERN =

    Pattern.compile("^" +

        "{9,}"          //at least 9 characters

    );

// pembatasan buat password

final Pattern PASSWORD_PATTERN =

    Pattern.compile("^" +

        "(?=.*[0-9])" +          //at least 1 digit

        "(?=.*[a-z])" +          //at least 1 lower case letter

        "(?=.*[A-Z])" +          //at least 1 upper case letter

        "(?=.*[a-zA-Z])" +       //any letter

        "(?=.*[@#$%^&+=])" +    //at least 1 special character

        "(?=\s+$)" +            //no white spaces

        "{8,}" +                 //at least 8 characters

        "$");

private SQLite dbsqlite;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_sign_up);

    // SQLite database handler

```

```

        dbsqlite = new SQLite(getApplicationContext());

        inisialisasi();

        getSupportActionBar().hide();

        Sign_in_Button();

        User user;
    }

    // perapihan

    private void inisialisasi() {

        edtName = findViewById(R.id.edt_addressss);

        edtEmail = findViewById(R.id.edt_email);

        edtPass = findViewById(R.id.edt_password);

        edtPhone = findViewById(R.id.edt_phone);

        btnSignUp = findViewById(R.id.btn_sign_up); // gw taroh di
        SignUp (LAYOUT)

        tvBtnSignIn = findViewById(R.id.tv_btn_signin);

        tvErrorText = findViewById(R.id.tv_errortext);

    }

    public void Sign_Up(View view) {

        /**

         * progress dialog, display the progress of any running task or an
         operation. An Alert Dialog is a type of alert

         * message displayed over the screen that lets users choose between
         options to respond to the message of the alert.

         */

        final ProgressDialog progressDialog = new ProgressDialog(this);

        progressDialog.setMessage("Please Wait..");

        // PENAMPUNGAN VARIABLE YANG TELAH DIRUBAH DIBAWAH

        pengubahan();

        //PENGECEKAN ketika nama kosong dia keluar notif !

```



```
        if (nama.length() < 1) {

            tvErrorText.setText("Name field is required!");

            edtName.requestFocus();

            return;

        }

        // KETIKA USERNAME LEBIH KECIL DARI 5 MAKA MASUK KE CLASS DIBAWAH YANG
MENGURUSNYA

        if (! (NAMA_PATTERN).matcher(nama).matches()) {

            registernama(edtName);

            return;

        }

        // JIKA EMAIL KOSONG MAKA DIA TURUN MENEGAMBIL CLASS YANG MENGURUSNYA
DIBAWAH

        if (email.isEmpty()) {

            registerEmail(edtEmail); // ambil kebawah

            return;

        }

        // JIKA EMAIL TIDAK SESUAI DENGAN KETENTUAN EMAIL SECARA GLOBAL
SEMISAL

        // @BLABLA.COM DIA AKAN MENGELUARKAN NOTIF

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {

            tvErrorText.setText("The email address you entered is not valid !
");

            edtEmail.requestFocus();

            return;

        }

        // jika NO TELP ANKGANYA CUMAN 1 DIA MENAMPILKAN PESAN!

        if (nohp.length() < 1) {

            tvErrorText.setText("Phone number field is required!");

            edtPhone.requestFocus();

        }

    }

}
```

```

        return;

    }

    if (nohp.length()>12){

        tvErrorText.setText("Your Phone number is too long!");

        edtPhone.requestFocus();

        return;

    }

    // JIKA NO TELPON TIDAK SESUAI DENGAN KETENTUAN YAITU NGACO NOMERNYA (
    MINIMAL 8 DIGIT )

    // MAKA DIA EROR (MENAMPILKAN PESAN)

    if (! (telepon_PATTERN).matcher(nohp).matches()) {

        tvErrorText.setText("The phone number you entered is not valid!");

        edtPhone.requestFocus();

        return;

    }

    // JIKA PASSWORD TIDAK SESUAI DENGAN KETENTUAN YAITU DIKIT ISINYA
    (MINIMAL 8 KARAKTER)

    // MAKA DIA EROR (MENAMPILKAN PESAN)

    if (edtPass.length()<1) {

        tvErrorText.setText("Password field is required!");

        edtPass.requestFocus();

        return;

    }

    // JIKA PASSWROD TIDAK SESUAI DENGAN KETENTUAN DIA NGACO ISINYA

    //MAKA PESAN NOTIF AKAN KELUAR

    if (! (PASSWORD_PATTERN).matcher(passwordw).matches()){

        tvErrorText.setText("At least with 8 Characters," +

            " 1 Lower case and 1 Upper case and have 1 Special
            character in your Password !");
    }

```



```

        // Error in registration! // user failed to store

        // jika gagal didaftarkan

        if (sukses == -2){

            progressDialog.dismiss();

            tvErrorText.setText(pesan);

            edtEmail.requestFocus();

            return;

        }

        // jika ada kesalan dalam pendaftaran

        if (sukses == -3) {

            progressDialog.dismiss();

            tvErrorText.setText(pesan);

            edtEmail.requestFocus();

            return;

        }

        else{

            progressDialog.dismiss();

            edtName.setText("");

            edtEmail.setText("");

            edtPass.setText("");

            edtPhone.setText("");

            Toast.makeText(SignUpActivity.this, pesan,
Toast.LENGTH_SHORT).show();

            //      String name, String email, String nohp, String
password ,String created_at (buat masukan sqlite)

            dbsqlite.addUser(nama, email, passwordw);

            // masukan ke sign in

            Intent i = new Intent(getApplication(),
SignInActivity.class);

```

```

        startActivity(i);

        // menyatakan selesai

        finish();

    }

} catch (Exception ex) {

    Log.e("Error: ", ex.toString());

    ex.printStackTrace();

}

}

}, new Response.ErrorListener() {

    @Override

    public void onErrorResponse(VolleyError error) { // jika eror

        Log.e("Error: ", error.getMessage());

        Toast.makeText(SignUpActivity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();

        progressDialog.dismiss();

    }

}

) {

    /**

        * metode dibawah ini adalah yang digunakan untuk memasukkan
data imputan dari aplikasi kedalam api,

        * dengan mencocokkan pada apa yang telah dideklarasikan pada
methode post di api (php - nya), sehingga

        * bisa terhubung.... kedalam api dan bisa mennyampaikannya
kedalam database bahwa terdapat insert - baru

        */

    @Override

```

```

        protected Map<String, String> getParams() throws
AuthFailureError {

            Map<String, String> params = new HashMap<>();

            params.put("name", nama);

            params.put("email", email);

            params.put("password", passwordw);

            params.put("nohp", nohp);

            return params;

        }

    };

    // baru di request

    RequestHandler.getInstance(this).addToRequestQueue(request);

}

}

```

```

private boolean pengubahan() {

    // penampungan sementara

    nama = edtName.getText().toString().trim();

    email = edtEmail.getText().toString().trim();

    passwordw = edtPass.getText().toString().trim();

    nohp = edtPhone.getText().toString().trim();

    return false;

}

private void registernama(EditText edtName) {

    pengubahan();

    tvErrorText.setText("A Minimum of 5 letters is required for the
FullName !");

    edtName.requestFocus();

    return;
}

```

```

    }

    private boolean registerEmail(EditText edtEmail) {

        pengubahan();

        if (email.length() < 5) {

            tvErrorText.setText("Email field is required ! ");

            edtEmail.requestFocus();

            return false;

        }

        else {

            tvErrorText.setText("");

            edtEmail.requestFocus();

            return true;

        }

    }

    private void Sign_in_Button() {

        // bwat pindah intent ke sign in activity, apabila udah ada akun

        tvBtnSignIn.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                Intent i = new Intent(getApplicationContext(), SignInActivity.class);

                startActivity(i);

            }

        });

    }

}

```

Code ini berfungsi untuk pengguna yang ingin menjadi user dengan cara mendaftarkan data dirinya dengan syarat harus mengisi nama, email , nomer hp , dan password. Sehingga data

tersebut akan pada SQLite,database MySQL yang terhubung menggunakan API JSON pada PHP.

2.2.17. BookCartDTO

```
package com.example.buku.dto;

import org.json.JSONException;
import org.json.JSONObject;
import java.io.Serializable;

public class BookCartDTO implements Serializable {

    /**Tempat Penampungan data pada keranjang (CART)
     */

    public String id;
    public String name;
    public String author;
    public String price;
    public String description;
    public String image;
    public String total_pages;
    public String published_at;
    public String isbn;
    public String rating;
    public String qty;
    public String id_cart;

    public BookCartDTO(JSONObject product) {

        try {

            id = product.getString("id");

            name = product.getString("name");

            author = product.getString("author");
```



```

        price = product.getString("price");

        description = product.getString("description");

        image = product.getString("image");

        total_pages = product.getString("total_pages");

        published_at = product.getString("published_at");

        isbn = product.getString("isbn");

        rating = product.getString("rating");

        qty = product.getString("qty");

        id_cart = product.getString("id_cart");

    } catch (JSONException e) {

    }

}
}

```

Code ini digunakan untuk menampung data keranjang buku setiap user yang diambil dari database menggunakan JSON API sebagai penghubung.

2.2.18. BookDTO

```

package com.example.buku.dto;

import org.json.JSONException;
import org.json.JSONObject;
import java.io.Serializable;

public class BookDTO implements Serializable {

    /**
     * Tempat Penampungan data pada BUKU
     */

    public String image;

    public String author;

```

```
public String name;

public String price;

public String description;

public String total_pages;

public String published_at;

public String isbn;

public String rating;

public static String stock;

public String id;


public BookDTO(JSONObject product) {

    try {

        image = product.getString("image");

        author = product.getString("author");

        name = product.getString("name");

        price = product.getString("price");

        id = product.getString("id");

        description = product.getString("description");

        total_pages = product.getString("total_pages");

        published_at = product.getString("published_at");

        isbn = product.getString("isbn");

        rating = product.getString("rating");

        stock = product.getString("stock");

    } catch (JSONException e) {

    }

}

}
```

Code ini digunakan untuk menampung data setiap buku yang diambil dari database menggunakan JSON API sebagai penghubung. Sehingga data buku tersebut dapat di tampilkan pada RecyclerView di HomeActivity.

2.2.19. Book History DTO

```
package com.example.buku.dto;

import org.json.JSONException;
import org.json.JSONObject;
import java.io.Serializable;

public class BookHistoryDTO {

    /**
     * Tempat Penampungan data PADA BUKU HISTORI PEMBELIAN USER
     */

    public String id;

    public String book_id;

    public String id_user;

    public String id_cart;

    public String name;

    public String image;

    public String qty;

    public String total_price;

    public String Address;

    public String status;

    public BookHistoryDTO(JSONObject product) {

        try {

            id = product.getString("id");

            book_id = product.getString("book_id");

            id_user = product.getString("id_user");
```

```

        id_cart = product.getString("id_cart");

        name = product.getString("name");

        image = product.getString("image");

        qty = product.getString("qty");

        total_price = product.getString("total_price");

        Address = product.getString("Address");

        status = product.getString("status");

    } catch (JSONException e) {

    }

}
}

```

Code ini digunakan untuk menampung data history setiap buku user yang dibeli dan diambil dari database menggunakan JSON API sebagai penghubung. Sehingga data buku tersebut dapat di tampilkan pada RecyclerView HistoryFragment.

2.2.20. BookItemFragment

```

package com.example.buku.fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.cardview.widget.CardView;
import androidx.fragment.app.Fragment;
import com.example.buku.R;

```

```
import com.example.buku.activity.DetailBookActivity;

public class BookItemFragment extends Fragment {

    CardView cvLayoutBook;

    ImageView imgBook;

    TextView tvAuthor, tvJudulBuku, tvHargaBuku;

    @Override

    public View onCreateView(LayoutInflater inflater, ViewGroup container,

                             Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.layout_book_item, container,
false);

        inisialisasi(v);

        mainButton();

        return v;

    }

    private void inisialisasi(View v){

        cvLayoutBook = v.findViewById(R.id.cv_layout_book);

        imgBook = v.findViewById(R.id.img_buku);

        tvAuthor = v.findViewById(R.id.tv_author);

        tvJudulBuku = v.findViewById(R.id.tv_judul_buku);

        tvHargaBuku = v.findViewById(R.id.tv_harga_buku);

    }

    private void mainButton(){

        cvLayoutBook.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                Intent i = new Intent(getActivity(),
DetailBookActivity.class);

                startActivity(i);

            }

        })

    }

}
```

```
});  
  
}  
  
}
```

Codingan ini berguna hanya untuk melakukan inisiasi awal pada BookItemFragment

2.2.21. BookItemHistoryFragment

```
package com.example.buku.fragment;  
  
import android.os.Bundle;  
  
import android.view.LayoutInflater;  
  
import android.view.View;  
  
import android.view.ViewGroup;  
  
import android.widget.ImageView;  
  
import android.widget.LinearLayout;  
  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
  
import androidx.fragment.app.Fragment;  
  
import com.example.buku.R;  
  
public class BookItemHistoryFragment extends Fragment {  
  
    LinearLayout llProduct;  
  
    ImageView imgBuku;  
  
    TextView tvAuthor, tvJudulBuku, tvHargaBuku;  
  
    public View onCreateView(@NonNull LayoutInflater inflater,  
  
                             ViewGroup container, Bundle savedInstanceState) {  
  
        View v = inflater.inflate(R.layout.layout_book_item_history,  
container, false);  
  
        inisialisasi(v);  
  
        return v;  
  
    }  
  
}
```

```

private void inisialisasi(View v){

    llProduct = v.findViewById(R.id.ll_product);

    imgBuku = v.findViewById(R.id.img_buku);

    tvAuthor = v.findViewById(R.id.tv_author);

    tvJudulBuku = v.findViewById(R.id.tv_judul_buku);

    tvHargaBuku = v.findViewById(R.id.tv_harga_buku);

}
}

```

Codingan ini berguna hanya untuk melakukan inisiasi awal pada BookHistoryFragment

2.2.22. CartFragment

```

package com.example.buku.fragment;

import static
com.example.buku.Handler.Constant.URL_DELETE_CART_FRAGMENT_USER;

import static com.example.buku.Handler.Constant.URL_MENAMPILKAN_DATA_CART;

import android.content.Context;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.TextView;

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 com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import com.example.buku.R;

import com.example.buku.activity.CheckoutActivity;

import com.example.buku.activity.HomeActivity;

import com.example.buku.adapter.ItemCartAdapter;

import com.example.buku.dto.BookCartDTO;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

public class CartFragment extends Fragment implements
ItemCartAdapter.ListItemClickListener{

    // inilisiasi awal

    RecyclerView rvcart;

    ArrayList<BookCartDTO> data; // pemanggilan awal arraylist bookCart DTO

    ItemCartAdapter adapter;

    Button btnPayment;

    TextView totalHarga;

    TextView TOTAL_barang_BUKU;
```



```

Intent i;

double total_harga;

int total_barang;

int ii= 0;

String hargaTotal;

@Override

public void onCreate(@Nullable Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

}

public View onCreateView(@NonNull LayoutInflater inflater,

                        ViewGroup container, Bundle savedInstanceState) {

    View v = inflater.inflate(R.layout.fragment_cart, container, false);

    inisialisasi(v, container.getContext());

    loadData();

    return v;

}

private void inisialisasi(View v, Context context) { // inisiasi awal

    TOTAL_barang_BUKU = v.findViewById(R.id.tv_total_barang);

    totalHarga = v.findViewById(R.id.tv_total_harga);

    btnPayment = v.findViewById(R.id.btn_payment);

    rvcart = v.findViewById(R.id.rv_cart);

    rvcart.setLayoutManager(new LinearLayoutManager(context,
LinearLayoutManager.VERTICAL, true));

    data = new ArrayList<>();

    adapter = new ItemCartAdapter(context, data, this);

    rvcart.setAdapter(adapter);

    adapter.notifyDataSetChanged();

```

```

        final int position = 0;

        btnPayment.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                try {

                    // int a = getArguments().getInt("position");

                    if(ii == 0){

                        Toast.makeText(getActivity(), "You dont have any book
in your cart !", Toast.LENGTH_SHORT).show();

                    }

                    else{

                        // dia intent ke checkout activity buat ngirim data
berdasarkan posisi

                        i = new Intent(getActivity(), CheckoutActivity.class);

                        i.putExtra("data_123", data.get(position));

                        i.putExtra("loadsPosition", hargaTotal);

                        i.putExtra("loads_summary",
TOTAL_barang_BUKU.getText().toString());

                        startActivity(i);

                    }

                }catch (Exception e) {

                    Toast.makeText(getActivity(), "You dont have any book in
your cart !", Toast.LENGTH_SHORT).show();

                    e.printStackTrace();

                }

            }

        });

    }

    private void loadData() {

        // digunakan buat ambil id_user dari session !!

```

```

    HomeActivity activity = (HomeActivity) getActivity();

    String id_data_User = activity.getUserDATA_ID();

    /** mulailah menggunakan library volley dibawah ini untuk menghubungkan
    ke api lalu ke database,

        * dibawah ini menggunakan metode post, yang berarti metode HTTP
    Request

        * yang digunakan untuk membuat data baru dengan menyisipkan data
    dalam body saat request dilakukan

    */

    StringRequest stringRequest = new StringRequest(Request.Method.POST,
    URL_MENAMPILKAN_DATA_CART,

        new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                try {

                    //converting the string to json array object

                    JSONArray array = new JSONArray(response);

                    data.clear();

                    //traversing through all the object

                    for (ii = 0; ii < array.length(); ii++) {

                        //getting product object from json array

                        JSONObject product = array.getJSONObject(ii);

                        //adding the product to product list

                        data.add(new BookCartDTO(product));

                    }

                    adapter.notifyDataSetChanged();

                    hitung_total_harga();

                } catch (JSONException e) { // jika eror

                    e.printStackTrace();

                }

            }

        }

    );

```

```

    }

    },

    new Response.ErrorListener() { // jika eror

        @Override

        public void onErrorResponse(VolleyError error) {

            Log.e("Error: ", error.getMessage());

        }

    }) {

        /**

         * metode dibawah ini adalah yang digunakan untuk memasukkan data
         inputan dari aplikasi ke dalam api,

         * dengan mencocokkan pada apa yang telah dideklarasikan pada
         method post di api (php - nya), sehingga

         * bisa terhubung.... kedalam api dan bisa menyampaikannya ke
         dalam database bahwa terdapat insert - baru

         */

        @Override

        protected Map<String, String> getParams() throws AuthFailureError
    {

        Map<String, String> params = new HashMap<>();

        // digunakan untuk mengambil data user id dari sessionn yaa!!!

        params.put("id_user", id_data_User);

        return params;

    }

    };

    // baru di request dah

    Volley.newRequestQueue(getActivity()).add(stringRequest);

}

```

```

private void load_Delete(BookCartDTO item) {

    /** mulailah menggunakan library volley dibawah ini untuk
    menghubungkan ke api lalu ke database,

        * dibawah ini menggunakan metode post, yang berarti metode HTTP
        Request

        * yang digunakan untuk membuat data baru dengan menyisipkan data
        dalam body saat request dilakukan

        */

    StringRequest request = new StringRequest(Request.Method.POST,
    URL_DELETE_CART_FRAGMENT_USER,

        new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                //adapter.notifyDataSetChanged();

                try {

                    JSONObject jsonObject = new JSONObject(response);

                    int sukses = jsonObject.getInt("success");

                    String pesan = jsonObject.getString("message");

                    if (sukses == 1) { // data berhasil diupdate

                        // inside on response method we are

                        // setting our edit text to empty.

                        loadData();

                        // on below line we are displaying a toast
                        message as data updated.

                        Toast.makeText(getActivity(), "Successfully
                        Delet the Book from your Cart!", Toast.LENGTH_SHORT).show();

                    }

                }

            }

        },

        catch (JSONException jsonObject) { // jika error

            jsonObject.printStackTrace();

```

```

    }

    }, new Response.ErrorListener() {

        @Override

        public void onErrorResponse(VolleyError error) { // jika eror

            // displaying toast message on response failure.

            Toast.makeText(getActivity(), "Fail to DELETE data..",
Toast.LENGTH_SHORT).show();

        }

    }) {

        @Override

        protected Map<String, String> getParams() throws AuthFailureError
{

            Map<String, String> params = new HashMap<>();

            params.put("id_cart", item.id_cart);

            params.put("id_book", item.id);

            return params;

        }

    };

    Volley.newRequestQueue(getActivity()).add(request);

}

@Override

public void on_plus_click(BookCartDTO item) {

    /**

        * function yang digunakan untuk pengubahan apabila jumlah cart di
tambah dia akan menambah angkanya

    */

    // membuat memori yg berbeda

    ArrayList <BookCartDTO> data = (ArrayList<BookCartDTO>)
this.data.clone();

```

```

        int index = data.indexOf(item);

        item.qty = Integer.toString(Integer.parseInt(item.qty) + 1);

        data.set(index, item);

        // data memori lama ( lain )

        this.data.clear();

        // data di add yang baru dari yg memori baru

        this.data.addAll(data);

        //notify di senggol adapternya biar sadar klo ada perubahan

        adapter.notifyDataSetChanged();

        //hitung total

        hitung_total_harga();

    }

    @Override

    public void on_min_click(BookCartDTO item) {

        /**

         * function yang digunakan untuk pengubahan apabila jumlah cart di
         tambah dia akan mengurangi angkanya

         */

        if (Integer.parseInt(item.qty) == 1 ){ // ini set qty == 1 jdi ga bisa
kurang dri ini

            return;

        }

        // membuat memori yg berbeda

        ArrayList <BookCartDTO> data = (ArrayList<BookCartDTO>)
this.data.clone();

        int index = data.indexOf(item);

        item.qty = Integer.toString(Integer.parseInt(item.qty) - 1);

        data.set(index, item);

        // data memori lama ( lain )

```

```

        this.data.clear();

        // data di add yang baru dari yg memori baru

        this.data.addAll(data);

        // di senggol adapternya biar sadar klo ada perubahan

        adapter.notifyDataSetChanged();

        //hitung total

        hitung_total_harga();

    }

    @Override

    public void on_hapus_cart(BookCartDTO item) {

        /**
         * tombol listener yang digunakan untuk menghapus data didalam cart
user
         */

        load_Delete(item);

    }

    private void hitung_total_harga(){

        /**
         * function yang digunakan untuk perhitungan
         */

        total_harga = 0;

        total_barang = 0;

        for (BookCartDTO item : data) {

            total_harga += Double.parseDouble(item.qty) *
Double.parseDouble(item.price); // perhitungan dri bookcartdto qty * price

            total_barang += Integer.parseInt(item.qty); // total barang
didapetin dri bookcartdto.qty

        }

        hargaTotal = Double.toString(total_harga);
    }

```



```

        totalHarga.setText("Rp. " + hargaTotal + "00"); // set total harga
        menjadi rp. sekian

        TOTAL_barang_BUKU.setText(Integer.toString(total_barang)); // ini
        total harga

    }

}

```

Code ini digunakan untuk menampilkan data buku yang akan dibeli di keranjang. Nantinya data tersebut akan di ambil dari database berdasarkan id setiap user yang berbeda menggunakan JSON API sebagai penghubung dari aplikasi ke dalam databae. Terdapat fungsi lain pada code ini yaitu perhitungan total buku yang dibeli, penghapusan buku pada keranjang, penambahan dan pengurangan buku yang dibeli.

2.2.23. CartItemFragment

```

package com.example.buku.fragment;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.fragment.app.Fragment;
import com.example.buku.R;
import com.example.buku.dto.BookDTO;
import java.util.ArrayList;

public class CartItemFragment extends Fragment {

    ImageView imgBuku;

    TextView tvJudulBuku, tvHargaBuku, tvid_id_cart;
}

```

```

Button btnMin, btnPlus, btnHapus;

EditText edtQty;

BookDTO bookDTO;

CartFragment cartFragment;

ArrayList<BookDTO> data = new ArrayList<>();

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

                           Bundle savedInstanceState) {

    // Inflate the layout for this fragment

    View v = inflater.inflate(R.layout.layout_cart_item, container,
false);

    //bookDTO = (BookDTO) getIntent().getSerializableExtra("data");

    inisialisasi(v);

    return v;

}

private void inisialisasi(View v) {

    imgBuku = v.findViewById(R.id.img_buku);

    tvJudulBuku = v.findViewById(R.id.tv_judul_buku);

    tvHargaBuku = v.findViewById(R.id.tv_harga_buku);

    btnMin = v.findViewById(R.id.btn_min);

    btnPlus = v.findViewById(R.id.btn_plus);

    btnHapus = v.findViewById(R.id.btn_hapus);

    edtQty = v.findViewById(R.id.edt_qty);

    tvid_id_cart = v.findViewById(R.id.ID_BUKU_CART);

}

}

```

Code ini digunakan hanya untuk inisiasi awal pada CartItemFragment

2.2.24. HistoryFragment

```
package com.example.buku.fragment;

import android.content.Context;

import android.os.Bundle;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import com.example.buku.Handler.Constant;

import com.example.buku.R;

import com.example.buku.activity.HomeActivity;

import com.example.buku.adapter.HistoryItemAdapter;

import com.example.buku.dto.BookHistoryDTO;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;
```

```

import java.util.HashMap;

import java.util.Map;

public class HistoryFragment extends Fragment {

    // deklarasi awal.

    RecyclerView rvHistoryTransaction;

    public View onCreateView(@NonNull LayoutInflater inflater,

                             ViewGroup container, Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.fragment_history, container,
false);

        inisialisasi(v, getContext());

        loadData(container.getContext());

        return v;

    }

    private void inisialisasi(View v, Context context){

        //buat layout menjadi vertical

        rvHistoryTransaction = v.findViewById(R.id.rv_history_transaction);

        rvHistoryTransaction.setLayoutManager(new LinearLayoutManager(context,
LinearLayoutManager.VERTICAL,true));

    }

    private void loadData(Context context) {

        // digunakan buat ambil id_user dari session !!

        HomeActivity activity = (HomeActivity) getActivity();

        String id_data_User = activity.getUserDATA_ID();

        ArrayList<BookHistoryDTO> data = new ArrayList<>(); // deklarasi
arraylist pad class book history DTO

        /** mulailah menggunakan library volley dibawah ini untuk
menghubungkan ke api lalu ke database,

        * dibawah ini menggunakan metode post, yang berarti metode HTTP
Request

```

```

        * yang digunakan untuk membuat data baru dengan menyisipkan data
        dalam body saat request dilakukan

        */

        StringRequest stringRequest = new StringRequest(Request.Method.POST,
Constant.URL_HISTORI_CART_PEMBELIANN,

        new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                try {

                    //converting the string to json array object

                    JSONArray array = new JSONArray(response);

                    //traversing through all the object

                    for (int i = 0; i < array.length(); i++) {

                        //getting product object from json array

                        JSONObject product = array.getJSONObject(i);

                        //adding the product to product list

                        data.add(new BookHistoryDTO(product));

                        HistoryItemAdapter adapter = new
HistoryItemAdapter(context, data);

                        rvHistoryTransaction.setAdapter(adapter);

                        adapter.notifyDataSetChanged();

                    }

                } catch (JSONException e) { // jika error

                    e.printStackTrace();

                }

            }

        },

        new Response.ErrorListener() { // jika error

            @Override

```

```

        public void onErrorResponse(VolleyError error) {

            Log.e("Error: ", error.getMessage());

        }

    }}{

        /**

         * metode dibawah ini adalah yang digunakan untuk memasukkan data
         imputan dari aplikasi kedalam api,

         * dengan mencocokkan pada apa yang telah dideklarasikan pada
         methode post di api (php - nya), sehingga

         * bisa terhubung.... kedalam api dan bisa mennyampaikannya
         kedalam database bahwa terdapat insert - baru

         */

        @Override

        protected Map<String, String> getParams() throws AuthFailureError
        {

            Map<String, String> params = new HashMap<>();

            // digunakan untuk mengambil data user id dari sessionn yaa!!!

            params.put("id_user", id_data_User);

            return params;

        }

    };

    // di request deh

    Volley.newRequestQueue(getActivity()).add(stringRequest);

}

}

```

Code ini digunakan untuk menampilkan data hasil pembelian setiap buku yang dibeli oleh masing - masing user. Data history pembelian buku tersebut di ambil dari database menggunakan JSON API agar terhubung ke dalam database. Kemudian tampilanya akan ditampilkan pada RecyclerView dari adapter yang sudah ada pada class HistoryItemAdapter

2.2.25. HomeFragment

```
package com.example.buku.fragment;

import static com.example.buku.Handler.Constant.URL_NEW_BOOK;
import static com.example.buku.Handler.Constant.URL_POPULAR;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ViewFlipper;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.buku.R;
import com.example.buku.activity.DetailBookActivity;
import com.example.buku.activity.SearchActivity;
```

```

import com.example.buku.adapter.NewBookAdapter;

import com.example.buku.dto.BookDTO;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;

public class HomeFragment extends Fragment {

    // deklarasi awal

    LinearLayout ll_searchView;

    ViewFlipper v_flipper;

    RecyclerView rvNewBook;

    RecyclerView rvPopularBook;

    public View onCreateView(@NonNull LayoutInflater inflater,

                             ViewGroup container, Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.fragment_home, container, false);

        inisialisasi(v, container.getContext());

        mainButton();

        // buat load di awal awal, berdasarkan buku baru dan juga buku yang
        populer pada recyleview tampilan awal

        loadData_bukuNEW_BOOK(container.getContext());

        LoadData_buku_POPULAR_BOOK(container.getContext());

        return v;

    }

    private void inisialisasi(View v, Context context){ // inisiasi awal

        ll_searchView = v.findViewById(R.id.ll_searchView);

        // ini buat bannerrrr

        int[] images = new int[] {R.drawable.carousel1, R.drawable.carousel2,
        R.drawable.carousel3, R.drawable.carousel4};

        v_flipper = v.findViewById(R.id.v_flipper);

```



```

        for (int image:images){

            flipperImages (image);

        }

        rvPopularBook = v.findViewById(R.id.rv_popular_book);

        rvPopularBook.setLayoutManager(new LinearLayoutManager(context,
LinearLayoutManager.HORIZONTAL,true));

        rvNewBook = v.findViewById(R.id.rv_new_book);

        rvNewBook.setLayoutManager(new LinearLayoutManager(context,
LinearLayoutManager.HORIZONTAL,true));

    }

    private void mainButton(){ // ke search activity ketika ditekan

        ll_searchView.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                Intent i = new Intent(getActivity(), SearchActivity.class);

                startActivity(i);

            }

        });

    }

    public void flipperImages(int image){ // buat function agar banner bisa
geser geser

        ImageView imageView = new ImageView(getContext());

        imageView.setBackgroundResource(image);

        v_flipper.addView(imageView);

        v_flipper.setFlipInterval(4000);

        v_flipper.setAutoStart(true);

        v_flipper.setInAnimation(getContext(), android.R.anim.slide_in_left);

```

```

        v_flipper.setOutAnimation(getContext(),
android.R.anim.slide_out_right);

    }

    private void loadData_bukuNEW_BOOK(Context context){ // load data buku
baru dri api yg diambil dri database

        ArrayList<BookDTO> data = new ArrayList<>(); // deklarasi arraylist
Bookdto

        /** mulailah menggunakan library volley dibawah ini untuk menghubungkan
ke api lalu ke database,

        * dibawah ini menggunakan metode GET, GET digunakan untuk request
data,

        * State server tidak berubah walaupun di-GET terus-terusan kecuali
data di server dirubah oleh client yang lain.

        */

        StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL_NEW_BOOK,

            new Response.Listener<String>() {

                @Override

                public void onResponse(String response) {

                    try {

                        //converting the string to json array object

                        JSONArray array = new JSONArray(response);

                        //traversing through all the object

                        for (int i = 0; i < array.length(); i++) {

                            //getting product object from json array

                            JSONObject product = array.getJSONObject(i);

                            // masukin ke arraylistnya berdasarkan dri
object api product

                            data.add(new BookDTO(product));

                        }

```

```

// ISI_Datanya();

NewBookAdapter adapter = new
NewBookAdapter(context, data);

adapter.setOnClickListener(new
NewBookAdapter.ItemClickListener() {

    /**
     * ketika bukunya ditekan maka recyleviewnya
    akan intent data, berdasarkan bukunya
     * sehingga di detail activity bisa ngebaca
    bahwa ini buku apa
     * @param view
     * @param position
     */

    @Override

    public void onItemClick(View view, int
position) {

        Intent i = new Intent(getActivity(),
DetailBookActivity.class);

        i.putExtra("data", data.get(position));

        startActivity(i);

    }

});

// disenggol adapternya biar sadar, (masukin
datanya kedalam recyleview dri adapter tdi)

rvNewBook.setAdapter(adapter);

adapter.notifyDataSetChanged();

} catch (JSONException e) { // jika error

    e.printStackTrace();

}

},

```

```

        new Response.ErrorListener() { // jika eror

            @Override

            public void onErrorResponse(VolleyError error) {

                Log.e("Error: ", error.getMessage());

            }

        });

// adding our stringrequest to queue

Volley.newRequestQueue(getActivity()).add(stringRequest);

}

public void LoadData_buku_POPULAR_BOOK(Context context){ // LOAD API DRI
database buku popular book

    ArrayList<BookDTO> data = new ArrayList<>(); // deklarasi awal
arraylist bookdto

    /** mulailah menggunakan library volley dibawah ini untuk menghubungkan
ke api lalu ke database,

    * dibawah ini menggunakan metode GET, GET digunakan untuk request
data,

    * State server tidak berubah walaupun di-GET terus-terusan kecuali
data di server dirubah oleh client yang lain.

    */

    StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL_POPULAR,

        new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                try {

                    //converting the string to json array object

                    JSONArray array = new JSONArray(response);

                    //traversing through all the object

                    for (int i = 0; i < array.length(); i++) {

                        //getting product object from json array

```

```

        JSONObject product = array.getJSONObject(i);

        //adding the product to product list

        data.add(new BookDTO(product));

    }

    // ISI_Datanya();

    NewBookAdapter adapter = new
NewBookAdapter(context, data);

    adapter.setOnClickListener(new
NewBookAdapter.ItemClickListener() {

        /**
         * ketika bukunya ditekan maka recyleviewnya
akan intent data, berdasarkan bukunya
         * sehingga di detail activity bisa ngebaca
bahwa ini buku apa
         * @param view
         * @param position
         */

        @Override

        public void onItemClick(View view, int
position) {

            Intent i = new Intent(getActivity(),
DetailBookActivity.class);

            i.putExtra("data", data.get(position));

            startActivity(i);

        }

    });

    // disenggol adapternya biar sadar, (masukin
datanya kedalam recyleview dri adapter tdi)

    rvPopularBook.setAdapter(adapter);

    adapter.notifyDataSetChanged();

} catch (JSONException e) {

```

```

        e.printStackTrace();
    }
}

},

new Response.ErrorListener() { // jika eror

    @Override

    public void onErrorResponse(VolleyError error) {

        Log.e("Error: ", error.getMessage());

    }

});

// direquest dah

Volley.newRequestQueue(getActivity()).add(stringRequest);

}

}

```

Code ini digunakan untuk menampilkan tampilan awal pada aplikasi BUKUPEDIA. Pada home terdapat tampilan RecyclerView yang menampilkan buku baru dan populer buku yang di ambil dari database menggunakan JSON API. Pada tampilan home terdapat banner yang telah dibuat dan dapat bergeser sesuai interval yang dapat diatur. Kemudian pada setiap masing - masing RecyclerView yang menampilkan data buku dari setiap adapter dapat di tekan, sehingga akan diarahkan kepada DetailActivity buku tersebut. Selanjutnya terdapat fitur yaitu untuk mencari buku menggunakan adapter class SearchListAdapter.

2.2.26. ProfileFragment

```

package com.example.buku.fragment;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

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 com.example.buku.Handler.SQLite;

import com.example.buku.R;

import com.example.buku.activity.ChangeProfileActivity;

import com.example.buku.activity.HomeActivity;

import com.example.buku.activity.WebViewActivity;

import com.example.buku.adapter.Session;

import com.example.buku.auth.SignInActivity;

import android.webkit.WebSettings;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import java.util.HashMap;

public class ProfileFragment extends Fragment {

    // inisiasi awal awal

    TextView tvName, tvEmail, tvPhone;

    Button btnChangeProfile, btnSignOut, mapp, button123;

    private Object ProfileFragment;

    public View onCreateView(@NonNull LayoutInflater inflater,

                             ViewGroup container, Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.fragment_profile, container,
false);
```

```

        inisialisasi(v);

        mainButton();

        return v;
    }

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

    }

    private void inisialisasi(View v){

        tvName = v.findViewById(R.id.tv_name);

        tvEmail = v.findViewById(R.id.tv_email);

        tvPhone = v.findViewById(R.id.tv_phone);

        btnChangeProfile = v.findViewById(R.id.btn_change_profile);

        btnSignOut = v.findViewById(R.id.btn_sign_out);

        mapp = v.findViewById(R.id.buat_map);

        button123 = v.findViewById(R.id.button123);

        HomeActivity activity = (HomeActivity) getActivity();

        String noHP_USER = activity.getUserDATA_NOHAPE();

        String id_data_User = activity.getUserDATA_ID(); // ga dipake

        // Fetching user details from SQLite

        // Menyiapkan data user dari SQLite

        SQLite dbEntry = new SQLite(getActivity());

        HashMap<String, String> user = dbEntry.getUserDetails();

        // dan yang diambil adalah data nama dan email dari SQLite

        String name = user.get("nama");

        String email = user.get("email");

        // di set text

        tvName.setText(name);
    }

```



```

        tvEmail.setText(email);

        tvPhone.setText(noHP_USER);

    }

    private void mainButton() {

        btnChangeProfile.setOnClickListener(new View.OnClickListener() { //
btn buat ke change profil activity

            @Override

            public void onClick(View v) {

                Intent i = new Intent(getActivity(),
ChangeProfileActivity.class);

                startActivity(i);

            }

        });

        btnSignOut.setOnClickListener(new View.OnClickListener() { // btn
signinout

            @Override

            public void onClick(View v) {

                SQLite dbEntry = new SQLite(getActivity());

                HomeActivity activity = (HomeActivity) getActivity();

                // menghapus data di SQLite jika di logout. sehingga tidak ada
penyimpanan di local database

                dbEntry.deleteUsers();

                activity.pencet();//panggil method buat log out, krn di
fragment sini ga bisa gitu

                // dia tempatnya ada di home activity

            }

        });

        mapp.setOnClickListener(new View.OnClickListener() { // buat nampilin
map

            @Override

            public void onClick(View v) {

```

```

        HomeActivity activity = (HomeActivity) getActivity();

        activity.tombol_map();

    }

});

button123.setOnClickListener(new View.OnClickListener() { // buat
website

    @Override

    public void onClick(View v) {

        HomeActivity activity = (HomeActivity) getActivity();

        activity.tombol_webview();

    }

});

}

}

```

Code ini digunakan untuk menampilkan nama, nomer hp, dan email yang diambil dari database MySQL menggunakan JSON API dan SQLite. Kemudian terdapat button untuk update data user tersebut, button maps, button sign out untuk mengeluarkan sesion pada user, dan button website BUKUPEDIA.

2.2.27. Constant

```

package com.example.buku.Handler;

public class Constant {

    /**

        * CLASS INI DIGUNAKAN UNTUK MENAMPUNG URL URL PADA API SEHINGGA LEBIH
        GAMPANG MEMANGGILNYA PADA CLASS LAIN

        * kami telah mencoba beberapa cara, untuk menggunakan database pribadi
        (server localhost, menggunakan 000webhostapp,

        * dan akhirnya kami menggunakan hostingan bernama jagoan hosting...

    */
}

```

```

    //private static final String ROOT_URL =
"http://192.168.1.20/MOBILE%20APPLICATION/BUAT_UAS/"; // asli ini

    //private static final String ROOT_URL = "https://si-
book.000webhostapp.com/BUAT%20UAS/"; // webb 000hosting free

    //private static final String ROOT_URL = "http://si-
book.42web.io/BUAT_UAS/"; // infinity web

    private static final String ROOT_URL =
"http://www.bookoemen.my.id/BUAT_UAS/";

    //BUAT AMBIL FOTO

    private static final String ROOT_URL2 =
"https://www.bookoemen.my.id/BUAT_UAS/buat_foto/";

    public static final String URL_REGISTER =
ROOT_URL+"Register_Percobaan.php";

    public static final String URL_LOGIN = ROOT_URL+"login.php";

    public static final String URL_UPDATE = ROOT_URL+"UPDATE.php";

    public static final String URL_NEW_BOOK = ROOT_URL2+"jsontdate.php";

    public static final String URL_POPULAR = ROOT_URL2+"json.php";

    public static final String URL_ADD_CART_USER =
ROOT_URL+"cartpercobaan.php";

    public static final String URL_MENAMPILKAN_DATA_CART = ROOT_URL2 +
"json_Menampilkan_Cart.php";

    public static final String URL_UPDATE_CART = ROOT_URL + "UPDATE_CART.php";
// ga dipake akrena ya ga jadi digunakan

    public static final String URL_DELETE_CART_FRAGMENT_USER = ROOT_URL +
"DELETE_CarT.php";

    public static final String url_web_view ="https://www.bookoemen.my.id";

    // upload pembayaran

    public static final String url_upload_bukti_pembayaran = ROOT_URL +
"upload_bukti_pembayaran.php";

    public static final String url_upload_bukti_pembayaran_bukan_gambar =
ROOT_URL + "TRANSACTION_ITEM.php";

    // historii

```

```
public static final String URL_HISTORI_CART_PEMBELIANN = ROOT_URL +  
"history.php";  
}
```

Class ini menampung data URL yang digunakan untuk meminta API JSON kedalam server hosting.

2.2.28. RequestHandler

```
package com.example.buku.Handler;  
  
import android.content.Context;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.toolbox.Volley;  
public class RequestHandler {  
    /**  
     * CLASS INI DIGUNAKAN untuk menjadi request handler dari volley  
     */  
    private static RequestHandler instance;  
    private RequestQueue requestQueue;  
    private static Context ctx;  
    private RequestHandler(Context context) {  
        ctx = context;  
        requestQueue = getRequestQueue();  
    }  
    public static synchronized RequestHandler getInstance(Context context) {  
        if (instance == null) {  
            instance = new RequestHandler(context);  
        }  
    }  
}
```

```

    }

    return instance;
}

public RequestQueue getRequestQueue() {

    if (requestQueue == null) {

        // getApplicationContext() is key, it keeps you from leaking the
        // Activity or BroadcastReceiver if someone passes one in.

        requestQueue =
Volley.newRequestQueue(ctx(getApplicationContext());

    }

    return requestQueue;
}

public <T> void addToRequestQueue(Request<T> req) {

    getRequestQueue().add(req);

}

}

```

Class ini digunakan untuk me-request Library Volley.

2.2.29. SQLite

```

package com.example.buku.Handler;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import java.util.HashMap;

```

```

/**
 * Menyimpan data di handphone masing-masing berbasis SQLite Database
 */

public class SQLite extends SQLiteOpenHelper{

    private static final String TAG = SQLite.class.getSimpleName();

    // All Static variables

    // Database Version

    private static final int DATABASE_VERSION = 9;

    // Database Name

    private static final String DATABASE_NAME = "lofRYYfffDyyc0Raldb77";

    // Login table name

    private static final String TABLE_USER = "user3ATDewGORYvvf57553";

    // Login Table Columns names

    private static final String KEY_ID = "id";

    private static final String KEY_NAME = "name";

    private static final String KEY_EMAIL = "email";

    //private static final String KEY_NoHP = "nohp";

    private static final String KEY_PASSWORD = "passwordw";

    //private static final String KEY_CREATED_AT = "created_at";

    public SQLite(Context context)

    {

        super(context, DATABASE_NAME, null, DATABASE_VERSION);

    }

    // Creating Tables

    // Membuat tabel SQLite

    @Override

    public void onCreate(SQLiteDatabase db) {

```

```

        String CREATE_LOGIN_TABLE = "CREATE TABLE " + TABLE_USER + "("
            + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_NAME +
            " TEXT,"
            + KEY_EMAIL + " TEXT," + KEY_PASSWORD + " TEXT" + ")";

        db.execSQL(CREATE_LOGIN_TABLE);

        Log.d(TAG, "Database tables created");
    }

    // Upgrading database

    @Override

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        // Drop older table if existed

        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER);

        // Create tables again

        onCreate(db);

    }

    /**
     * Storing user details in database
     * Menyimpan data user kedalam database SQLite
     * */

    public void addUser(String nama, String email, String password) {

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(KEY_NAME, nama); // Name

        values.put(KEY_EMAIL, email); // Email

        values.put(KEY_PASSWORD, password); // password

        // Inserting Row

        long id = db.insert(TABLE_USER, null, values);

        db.close(); // Closing database connection
    }

```

```

        Log.d(TAG, "New user inserted into sqlite: " + id);
    }

    // below is the method for updating our courses

    public void updateUser(String id, String nama, String email) {

        // calling a method to get writable database.

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        // on below line we are passing all values

        // along with its key and value pair.

        values.put(KEY_ID, id); // Name

        values.put(KEY_NAME, nama); // Name

        values.put(KEY_EMAIL, email); // Email

        // on below line we are calling a update method to update our database
        and passing our values.

        // and we are comparing it with name of our course which is stored in
        original name variable.

        db.update(TABLE_USER, values, "id=?", new String[]{id});

        db.close();

//        db.insert(TABLE_USER, null, values);
//        db.close(); // Closing database connection
    }

    /**
     * Getting user data from database
     * Mengambil data user dari Database SQLite
     */

    public HashMap<String, String> getUserDetails() {

        HashMap<String, String> user = new HashMap<String, String>();

        String selectQuery = "SELECT * FROM " + TABLE_USER;

        SQLiteDatabase db = this.getReadableDatabase();

```



```

        Cursor cursor = db.rawQuery(selectQuery, null);

        // Move to first row
        cursor.moveToFirst();

        if (cursor.getCount() > -1) {

            user.put("nama", cursor.getString(1));

            user.put("email", cursor.getString(2));

            user.put("password", cursor.getString(3));

            user.put("id", cursor.getString(0));

        }

        cursor.close();

        db.close();

        // return user

        Log.d(TAG, "Fetching user from Sqlite: " + user.toString());

        return user;
    }

    /**
     * Recreate database Delete all tables and create them again
     * Menghapus isi database SQLite untuk diperbarui.
     * */
    public void deleteUsers() {

        SQLiteDatabase db = this.getWritableDatabase();

        // Delete All Rows

        db.delete(TABLE_USER, null, null);

        db.close();

        Log.d(TAG, "Deleted all user info from sqlite");

    }

```

```

/**
 * This method to check user exist or not
 *
 * @param email
 * @param password
 * @return true/false
 */

public boolean checkUser(String email, String password) {

    // array of columns to fetch

    String[] columns = {

        KEY_ID

    };

    SQLiteDatabase db = this.getReadableDatabase();

    // selection criteria

    String selection = KEY_EMAIL + " = ?" + " AND " + KEY_PASSWORD + " =
?";

    // selection arguments

    String[] selectionArgs = {email, password};

    // query user table with conditions

    /**
     * Here query function is used to fetch records from user table this
     function works like we use sql query.

     * SQL query equivalent to this query function is

     * SELECT user_id FROM user WHERE user_email =
'jack@androidtutorialshub.com' AND user_password = 'qwerty';

     */

    Cursor cursor = db.query(TABLE_USER, //Table to query

        columns, //columns to return

        selection, //columns for the WHERE clause

```

```

        selectionArgs,                //The values for the WHERE clause

        null,                        //group the rows

        null,                        //filter by row groups

        null);                      //The sort order

    int cursorCount = cursor.getCount();

    cursor.close();

    db.close();

    if (cursorCount > 0) {

        return true;

    }

    return false;

}

}

```

Class ini digunakan untuk membuat database SQLite baru pada setiap masing - masing handphone user. Didalamnya terdapat fungsi membuat tabel baru untuk masing - masing akun user, melakukan pengecekan terhadap user yang sudah melakukan login atau tidak, melakukan logout, dan mengupdate data user.

2.3.1 Proses Bisnis Aplikasi BUKUPEDIA

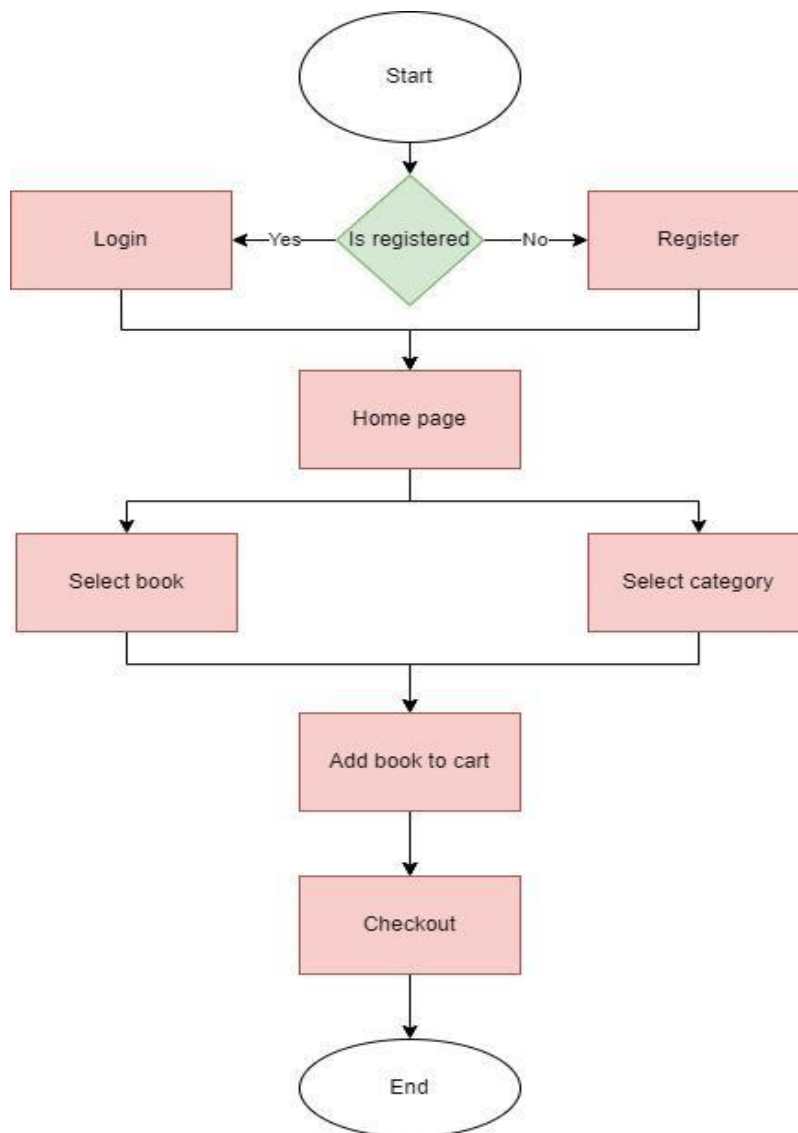


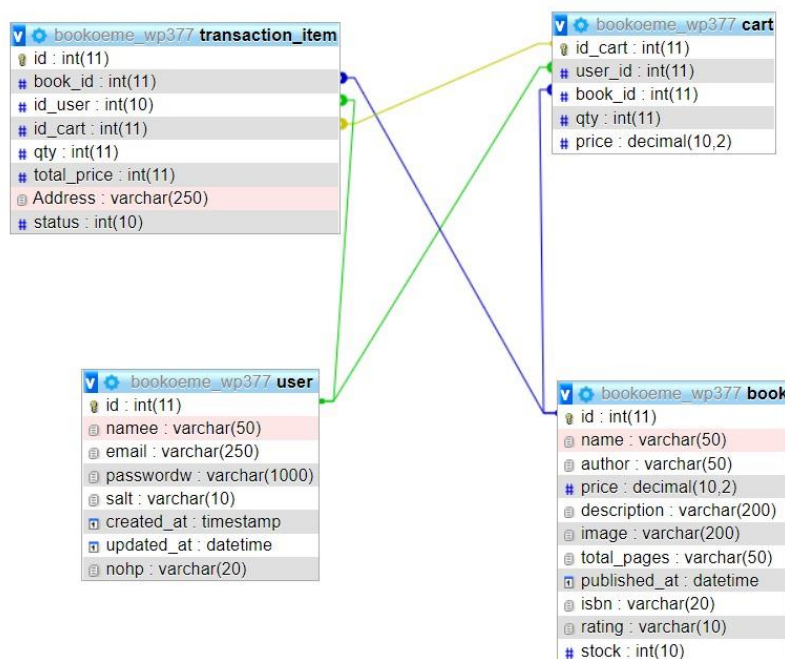
Diagram diatas berikut merupakan proses bisnis dari aplikasi “BUKUPEDIA”. Proses pertama dalam proses bisnis disini akan dimulai dengan “start”. Ketika pengguna sudah memiliki akun, pengguna dapat langsung dengan proses “Login” kepada proses selanjutnya. Jika pengguna tidak memiliki akun sebagai User maka pengguna harus daftar terlebih dahulu untuk mengisi sebuah akun melalui proses “Signup”. Dalam pendaftaran akun, pengguna akan diperlukan untuk mengisi data - data dirinya seperti nama, email, nomor telepon, dan password sesuai kebutuhan sistem agar pengguna dapat masuk kepada proses yang selanjutnya. Ketika pengguna dapat berhasil masuk, disini pengguna juga dapat login hanya menggunakan namanya saja (Fullname) untuk melakukan login pada aplikasi.

Sesudah pengguna sudah masuk ke aplikasinya , pengguna disini akan langsung melihat tampilan “Home Page” disini pengguna juga dapat melihat berbagai macam fitur di dalam “Home Page”. Pengguna disini langsung dapat melihat lihat buku dan pengguna dapat melakukan “Select book” dan melakukan “Select category”. Ketika pengguna sudah menentukan buku yang ingin

dibeli maka pengguna tersebut dapat melakukan “Add book to cart”. Pada proses ini pengguna akan memasukan buku kedalam cart kemudian pengguna juga dapat menentukan jumlah buku yang ingin dibeli melalui “Add book to cart”.

Ketika pengguna sudah menyelesaikan proses tersebut maka masuk ke proses selanjutnya yaitu “Checkout” disini pengguna harus menuliskan alamat tujuan untuk dikirimkan. Kemudian pengguna juga harus mengirimkan bukti berupa screenshoot untuk admin cek apakah pengguna sudah berhasil melakukan pembayaran. Sesudah melakukan pembayaran pengguna akan konfirmasi jika ingin menyelesaikan pembelian maka pengguna akan langsung ke Payment dan status pembelian barang akan di cek oleh admin. Selanjutnya admin akan melihat apakah bukti screenshoot yang dikirimkan pengguna sudah masuk jika “Ya” maka status pembayaran akan akan berhasil, namun sebaliknya yaitu jika “Tidak” maka Admin akan melakukan reject dalam pembelian.

2.3.2 ERD Diagram BUKUPEDIA



Gambar diatas merupakan gambar diagram ERD dari aplikasi Bookpedia. Terlihat bahwa terdapat 4 tabel utama yang digunakan dalam database Bookpedia yaitu “user”, “book”, “cart”, dan “transaction_item”. Data dari tabel masing - masing saling membutuhkan maka dari itu keempat tabel tersebut berhubungan satu sama lain serta fungsi tabel - tabel telah disinkronisasikan. Data dari user diperlukan untuk masuk ke dalam table transaction item dan table cart yang berguna untuk mencocokkan id user dengan keranjang per masing masing user agar tidak tercampur dengan cart milik user lain, dan begitu pula dengan transaction item yang menampilkan history sesuai dengan per masing masing user.

Terdapat 4 primary key utama yang terdapat pada masing masing table diantaranya id pada table book, id_cart pada table cart, id pada table transaction_item, dan yang terakhir adalah id pada table user.

2.3.3 Sql Trigger Query

Selain itu kelompok kami juga membuat sql trigger query yang berfungsi untuk mengurangi stok pada table buku apabila terjadi transaksi yang dilakukan oleh user agar dapat mencocokkan stok dengan gudang.

(Trigger KurangStok)

```
BEGIN
    UPDATE book set stock = stock - NEW.qty
    WHERE id = NEW.book_id;
END
```

BAB III

PENUTUP

3.1 Kesimpulan

Aplikasi Bukupedia merupakan aplikasi jual buku online yang dapat membantu user untuk membeli buku tanpa harus datang ke toko buku secara luring. Pada saat ini, masyarakat terkadang tidak mempunyai waktu untuk pergi membeli buku ke toko buku kesukaan mereka, yang mana hal tersebut akan membuat mereka tidak dapat membaca buku yang mereka inginkan. Bukupedia menyediakan berbagai solusi seperti user tidak perlu repot untuk pergi untuk mencari buku mereka karena akan diantarkan ke alamat yang mereka daftarkan, user dapat mencari buku kesukaan mereka dengan aplikasi Bukupedia dengan smartphone mereka masing-masing, dan masih banyak lagi.

Aplikasi kamu juga menyediakan berbagai jenis buku yang dapat memikat hati masyarakat dengan berbagai jenis genre dan judul yang dapat diakses oleh user. Aplikasi Bukupedia juga memiliki beberapa fitur-fitur seperti Sign Up Activity, Sign In Activity, Change Profile Activity, dan masih banyak lagi yang dapat memenuhi kebutuhan user dalam menggunakan aplikasi Bookpedia. Berdasarkan penjelasan diatas, aplikasi Bookpedia tersebut dapat ditarik kesimpulan sebagai berikut :

1. Pembangunan aplikasi berbasis android ini memudahkan bagi user pembeli untuk memperoleh informasi buku yang ditawarkan oleh aplikasi Bookpedia, serta memfasilitasi kemudahan yang diberikan kepada konsumen dalam melakukan pemesanan secara online.
2. Aplikasi Bookpedia yang sudah dibangun ini berhasil menampilkan beberapa fitur seperti Home, Maps, Webview, Change Profile, View Pager, Transaction, History, SqlLite, Search, dan Cart.
3. Dengan adanya aplikasi Bookpedia, masyarakat dapat dengan mudah mendapatkan informasi seputar buku yang ingin mereka beli.

3.2 Penutup

Berikut dari kami ditampilkan atas secara keseluruhan adalah mengenai aplikasi yang telah disusun oleh anggota-anggota kelompok kami. Fitur yang telah diimplementasikan dengan teliti agar dapat berfungsi dengan lancar tanpa kesalahan yang dapat menghambat aktivitas dalam aplikasi. Tujuan dari aplikasi yang kami telah buat dikerjakan secara perlahan-lahan sehingga setiap fitur dapat digunakan oleh setiap pengguna.

3.3 Peran Anggota

Nama	NIM	Peran
Agym Dimas Tian	00000058149	Membuat desain logo dan banner, Membantu mengerjakan PHP pada server hosting dan Proposal
Arlynandhita Felisya Putri Wibowo	00000061610	Mengerjakan fungsional aplikasi pada java, desain XML aplikasi, View Pager, RecyclerView, Search View, ERD, dan Proposal
Benz Martheen W Jonash	00000059822	Membantu membuat PHP untuk Transaction dan juga Upload Image, dan Proposal
Jericho Alexander Gunawan	00000058345	Membantu membuat design UI pada aplikasi, Membuat dan mengedit isi Proposal, Mencari data yang dibutuhkan aplikasi.
Nicholas Suwandi Wijaya	00000057903	Mengerjakan Maps Activity, Web Activity, PHP, MySQL, ERD, dan Proposal
Rheinald Adyatma Rizqi Wibowo	00000058204	Mengerjakan fungsional aplikasi pada java dan PHP, SQLite, MySQL, ERD, RecyclerView, dan Proposal

REFERENSI

Dalisay, M. D. I. M., Dalisay, M., Dalisay, I. M., View all of Mike Dalisay's posts., says:, M. D., says:, L., says:, B. N., says:, L., says:, F. J., Says:, A. M., Says:, D., says:, E. A., says:, D. E., Says: أحمد عيسى, says:, L. S., says:, I., Says:, S., says:, B. K., says:, E., ... says:, M. A. H. (2022, March 22). PHP shopping cart tutorial using mysql database – step by step guide! CodeOfaNinja. Retrieved December 15, 2022, from <https://www.codeofaninja.com/php-shopping-cart-tutorial-using-mysql-database/>

Jurnal Mobile Commerce jaidan. 123dok.com. (n.d.). Retrieved September 15, 2022, from <https://123dok.com/document/y9dwd8vq-jurnal-mobile-commerce-jaidan.html>

Welcome to Lumbung Pustaka UNY. Welcome to Lumbung Pustaka UNY - Lumbung Pustaka UNY. (n.d.). Retrieved December 15, 2022, from <https://eprints.uny.ac.id/>

LAMPIRAN

Link Video:

https://drive.google.com/drive/u/2/folders/1QXLEKbLPUfYZRDVUAi4_mEj8mhF50p2f

