

**LAPORAN PROYEK AKHIR**  
**MOBILE APPLICATION DEVELOPMENT**  
**APLIKASI JUAL BUKU BEKAS “STUDEPOSITORY”**



**Disusun Oleh :**

Kumala Dewi Chandra Asih (00000043238)

Aulia Kharisma Putri (00000051728)

Felicia Hanna Ratna (00000052746)

Yuki Sabrina Nakaya (00000053421)

Deandra Yasin Winarno (0000067755)

**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS TEKNOLOGI DAN INFORMATIKA**  
**UNIVERSITAS MULTIMEDIA NUSANTARA**  
**TANGERANG GANJIL 2021/2022**

## **KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa. Karena atas rahmat dan anugerahNya kami dapat menyelesaikan laporan proyek akhir dengan judul “Mobile Application Development Aplikasi Jual Buku Bekas Studepository” ini dengan lancar.

Adapun tujuan dari Penyusunan laporan proyek akhir ini untuk memenuhi tugas Mobile Application Development. Kami berterima kasih kepada yang terhormat Bapak Agus Sulaiman selaku pembimbing materi dalam pembuatan proposal ini. Harapan kami bahwa proposal ini dapat bermanfaat bagi para pembaca untuk menambah wawasan dan pengetahuan mengenai proses pembelian buku bekas di aplikasi kami Studepository.

Kami menyadari bahwa laporan proyek akhir ini masih jauh dari sempurna. Oleh karena itu kami menerima kritik dan saran agar pembuatan laporan akhir kami lebih baik kedepanya.

Tangerang, 1 Desember 2021

## DAFTAR ISI

<b>BAB 1</b>	<b>5</b>
PENDAHULUAN	5
1.1 LATAR BELAKANG	5
1.2 TUJUAN	7
1.3 MANFAAT	7
<b>BAB 2</b>	<b>7</b>
PEMBAHASAN	8
2.1 Menu dan Fitur pada Aplikasi Studepository	8
2.1.1 Splashscreen	8
2.1.2 Sign Up	8
2.1.3 Log In	8
2.1.4 Main Menu	8
2.1.5 Home Activity	9
2.1.6 Belanja Activity	9
2.1.7 Maps Activity	9
2.1.8 Checkout Details	9
2.1.9 Payment	9
2.1.10 Finish Payment	10
2.1.11 Profile	10
2.1.12 Categories One	10
2.1.13 Fragment 1 (View Pager)	10
2.1.14 Fragment 2 (Web View)	10
2.1.15 Item Menu	10
2.2 Konsep Bisnis Aplikasi Studepository	11
2.3 Proses Bisnis Aplikasi Studepository	12
2.4 ERD Diagram Studepository	14
2.5 Capture Layout Aplikasi “Studepository”	15
2.5.0 Cover (Splashscreen)	15
2.5.1 Main Activity	15
2.5.2 Sign Up	16
2.5.3 Log In	17
2.5.4 Home Activity	17
2.5.5 Add Book	18
2.5.6 Item List Book	19
2.5.7 Belanja Activity	19
2.5.9 Checkout Details	21

2.5.10 Payment Activity	21
2.5.11 Finish Payment	22
2.5.12 Profile Activity	23
2.5.13 Categories One	23
2.5.14 Fragment 1 (ViewPager)	24
2.5.16 Item Menu	25
2.6 Capture Coding yang Digunakan Dalam Aplikasi “Studepository”	26
2.6.0 SplashScreen	26
2.6.1 MainActivity	27
2.6.2 Sign Up	27
2.6.3 Login	28
2.6.4 MainMenu	30
2.6.5 HomeActivity	30
2.6.6 AddBook	31
2.6.7 Item List Book	33
2.6.8 Belanja Activity	34
2.6.9 Maps Activity	35
2.6.10 Checkout Details	36
2.6.11 Payment	36
2.6.12 Finish Payment	37
2.6.13 Categories One	37
2.6.14 Fragment 1 (View Pager)	38
2.6.15 Fragment 2 (WebView)	39
2.6.16 Item Menu	39
<b>BAB III</b>	<b>41</b>
<b>PENUTUP</b>	<b>41</b>
3.1 Kesimpulan	41
3.2 Penutup	42
3.3 Peran Anggota	42

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 LATAR BELAKANG**

Buku merupakan kumpulan lembaran kertas yang tersusun menjadi satu. Pada dasarnya, buku menyimpan peran yang begitu besar bagi masyarakat dalam menambah wawasan dan meningkatkan pengetahuan diri. Persaingan dunia usaha buku di Indonesia semakin ketat, setiap perusahaan atau toko berlomba-lomba untuk menarik pelanggan dan mempertahankan keberadaannya di pasar. Semakin bertambahnya toko buku modern dan non modern di Indonesia membuat masyarakat mempunyai banyak pilihan untuk berbelanja. Di Indonesia semakin banyak jumlah toko buku modern dan non modern yang membuat masyarakat memiliki banyak pilihan untuk berbelanja. Bagi konsumen hal ini memberikan banyak keuntungan karena dapat memilih sesuai dengan kebutuhan dan keinginannya.

Kemajuan teknologi berkembang sangat pesat untuk menunjang dunia perdagangan seperti buku bekas. Sistem penjualan buku bekas yang berjalan hingga saat ini masih dilakukan secara manual sehingga hal ini menimbulkan masalah karena penginputan data buku yang masih dilakukan secara manual. Dalam melihat stok buku memakan waktu yang banyak dan menimbulkan kesalahan data. Masalah lain diantaranya masih kesusahan dalam mencari buku saat transaksi tidak didukung dengan tempat yang tertata rapi sesuai kategori buku yang ada.

Jual beli buku bekas sudah banyak dilakukan. Beberapa penjual tradisional yang bergerak dalam bidang jual beli buku bekas belum bisa membuat sistem berbasis web karena adanya kendala pengetahuan yang minim. Ada beberapa toko buku bekas yang berada di Jakarta namun beberapa toko buku ini perlu memanfaatkan teknologi informasi untuk meningkatkan kinerja dan pendapatan agar dapat bersaing dan bertahan dalam pasar.

Beberapa toko buku bekas ini dalam menangani penjualan ataupun persediaan buku bekas masih menggunakan sistem manual, sehingga ketika pelanggan menanyakan buku bekas yang dibutuhkan sesuai kriteria tertentu, maka beberapa

toko ini membutuhkan waktu yang cukup lama hanya untuk mencari buku dengan kriteria yang diajukan oleh pelanggan.

Pengolahan dan penyimpanan data buku dibutuhkan suatu aplikasi komputer untuk memudahkan setiap kegiatan yang ada. Sehingga aplikasi ini dapat digunakan untuk mempermudah dalam pencatatan transaksi

Perihal penjualan produk secara online dapat seringkali bermunculan dalam masa kini, produk yang dapat dijual berkondisi baru ataupun bekas. Dalam konsep proyek kami, aplikasi tersebut berbasis e-commerce dimana produk dapat di jual dan beli secara online. Panduan aplikasi e-commerce dapat memberi pengguna untuk saling membeli produk atau menjual produk kepada satu sama lain. Produk-produk yang di jual dalam e-commerce dapat ditemukan oleh yang sedang membutuhkan produk tertentu.

Konsep jual beli antara pengguna e-commerce dapat memanfaatkan satu sama lain oleh cara memenuhi kebutuhan tersendiri. Jual beli dapat memberi keuntungan bagi para pengguna yang tidak ingin menggunakan barang-barang yang tidak lagi dipakai, walaupun produk rentan tidak dijual secara langsung. Aplikasi e-commerce memberi oportunitas bagi pengguna untuk dapat melakukan jual beli serta membangun usaha yang dimiliki oleh pengguna. Begitu dengan tujuan utama aplikasi e-commerce, para pengguna dapat melakukan mendapatkan keuntungan dari hasil jual barang dalam aplikasi tersebut.

Aktivitas dalam e-commerce pada umumnya dapat memiliki beragam aktivitas yang dapat dilakukan oleh aplikasi termasuk jual beli. Transaksi jual beli barang merupakan satu aktivitas usaha yang paling umum dilakukan oleh setiap pengguna. Kebutuhan hayati yang semakin semakin tinggi dan jumlah warga yang pula semakin banyak, menciptakan transaksi jual beli semakin tinggi dan semakin mendesak berdasarkan tahun ke tahun. Oleh lantaran itu dibutuhkan wadah yang bisa mempermudah penjual untuk mempublikasikan dan mempromosikan barang yang ia jual kepada para pembeli, sebagai akibatnya transaksi antara penjual dan pembeli bisa berjalan lancar. Walaupun sekarang ini semakin bermunculan website-website yang menyediakan tempat transaksi jual beli secara online.

Tetapi warga yang mempunyai gerak tinggi dan mempunyai kesibukannya masing-masing, tetap kesulitan melakukan transaksi jual beli. Hal ini dikarenakan sebagian besar website tadi diperuntukkan bagi pengguna internet melalui desktop,

sebagai akibatnya transaksi jual beli terhambat dan tidak bisa dilakukan di mana saja. Sedangkan dengan tingginya kegiatan dan gerak warga, warga membutuhkan tempat yang lebih mudah dan efisien, sebagai akibatnya warga bisa melakukan transaksi dimanapun dan kapanpun.

Karena permasalahan tersebut akhirnya diciptakan sebuah layanan pelaksanaan jual beli yang berbasis lokasi memakai sistem operasi Android yang diharapkan bisa membantu dan mempermudah para pelaku ekonomi yang merupakan penjual dan pembeli untuk menemukan tempat yang mudah, aman, cepat, dan efisien dalam bertransaksi. apabila dibandingkan M-Commerce semacam OLX atau Carousell, pelaksanaan yang kami kembangkan mempunyai fitur berbasis lokasi, dimana user bisa mengetahui posisi user lain. Dengan fitur ini, diharapkan akan muncul kemudahan antara pembeli dan penjual, dikarenakan mereka bisa mengetahui posisi orisinil masing-masing pihak. Selain itu menggunakan fitur berbasis lokasi, pembeli bisa mengetahui posisi penjual yang terdekat berdasarkan posisinya.

## **1.2 TUJUAN**

Tujuan :

- Mempermudah masyarakat untuk melakukan pembelian barang bekas (second)
- Mengurangi adanya pembuangan barang yang masih dalam kondisi baik ataupun layak pakai.
- Menyediakan platform bagi penjual untuk mencari pelanggan.
- Keberlanjutan bagi lingkungan dengan mengurangi produksi barang berlebihan.

## **1.3 MANFAAT**

Manfaat :

- Mengurangi limbah barang bekas.
- Memberi peningkatan penjualan barang bekas.
- Mempermudah masyarakat dalam mencari ataupun membeli barang bekas.
- Memberikan jaminan kualitas dari barang sesuai dengan kualitas yang dideskripsikan penjual.
- Menyediakan barang dengan harga murah dengan kualitas terjamin

## **BAB 2**

### **PEMBAHASAN**

#### **2.1 Menu dan Fitur pada Aplikasi Studepository**

##### **2.1.1 *Splashscreen***

Splashscreen merupakan tampilan awal yang muncul pada saat membuka aplikasi Studepository. Background warna Studepository berwarna hijau dan dengan logo tumpukan buku. Logo tumpukan buku ini berarti Studepository menjual berbagai macam buku-buku bekas yang para pembeli dapat memilih sesuai dengan keinginan.

##### **2.1.2 *Sign Up***

Fitur Sign Up berfungsi sebagai pengguna ketika ingin masuk ke aplikasi kami. Pada umumnya jika pengguna tidak memiliki sebuah akun dalam aplikasi maka pengguna diharuskan untuk melakukan pendaftaran melalui Sign Up dalam aplikasi yang membutuhkan data-data pribadi para pengguna seperti e-mail, nama pengguna, dan password.

##### **2.1.3 *Log In***

Fitur untuk pengguna yang ingin masuk ke aplikasi dengan menggunakan akun yang telah dibuat setelah *Sign Up*. Dengan ini pengguna dapat mengakses aplikasi dengan data yang sama di berbagai handphone.

##### **2.1.4 *Main Menu***

Laman pertama yang ditampilkan kepada pengguna aplikasi sesudah melakukan Sign Up dan Log In dalam aplikasi, dalam fitur Main Menu beberapa fitur lainnya dapat ditemukan dan dapat ditelusuri lebih lanjut oleh para pengguna tergantung kebutuhan yang ingin dituju.



### **2.1.5 Home Activity**

Setiap produk-produk yang tertampil kan berada di Home Activity serta setiap fitur yang berhubungan dengan Home Activity. Seperti daftar buku yang ditampilkan sebagai product.

### **2.1.6 Belanja Activity**

Shop Activity, fitur ini akan menampilkan detail dari buku yang ingin kita lihat secara detail, seperti melihat sinopsi dari buku, rating buku, hingga lokasi penjual buku tersebut.

### **2.1.7 Maps Activity**

Fitur Maps Activity ini menunjukkan dimana tempat barang yang dijual tersebut. Fitur ini memiliki tampilan seperti google maps dimana kita bisa melihat jarak dan lokasi antara lokasi kita dengan lokasi penjual buku tersebut.

### **2.1.8 Checkout Details**

Pada fitur Check out user dapat membeli buku setelah melihat detail dari buku tersebut, bahkan lokasi dimana penjual buku tersebut berada. Dalam fitur ini tersedia payment secara Cash On Delivery (COD) dimana penjual dan pembeli akan bertemu di lokasi yang telah ditentukan dan melakukan pembayaran secara langsung.

### **2.1.9 Payment**

Tahap yang akan ditelusuri oleh pengguna setelah melakukan *checkout*. Dalam aplikasi Studepository, pengguna hanya dapat membayar dengan cash dan bertemu di tempat secara tatap muka yaitu COD (Cash on Delivery).

#### **2.1.10 *Finish Payment***

Memberi status payment telah berhasil dilakukan dan telah terkonfirmasi bahwa produk yang terpilih akan dibeli oleh pengguna. Setelah layout payment.

#### **2.1.11 *Profile***

Menampilkan data dari pemilik akun, seperti nama, email, dan username pengguna yang *log-in* dan setelah melakukan registrasi oleh fitur sign up dalam aplikasi.

#### **2.1.12 *Categories One***

Dimana berbagai buku-buku dikelompokkan sesuai jenis buku, setiap buku akan memiliki tampilan dan deskripsi spesifikasinya yang dapat diperiksa oleh pengguna secara detail.

#### **2.1.13 *Fragment 1 (View Pager)***

Fitur view pager ini digunakan untuk mengganti dari satu halaman ke halaman lain yang berisi buku-buku. Sehingga memudahkan pengguna dalam melihat lebih jelas buku tersebut dan lebih efisien untuk digunakan.

#### **2.1.14 *Fragment 2 (Web View)***

Fitur web view digunakan untuk memberi link informasi terkait tentang aplikasi Studepository ini, pada laman tersebut kami memasukan link/url instagram kami untuk memudahkan pengguna dalam menemukan akun sosial media kami.

#### **2.1.15 *Item Menu***

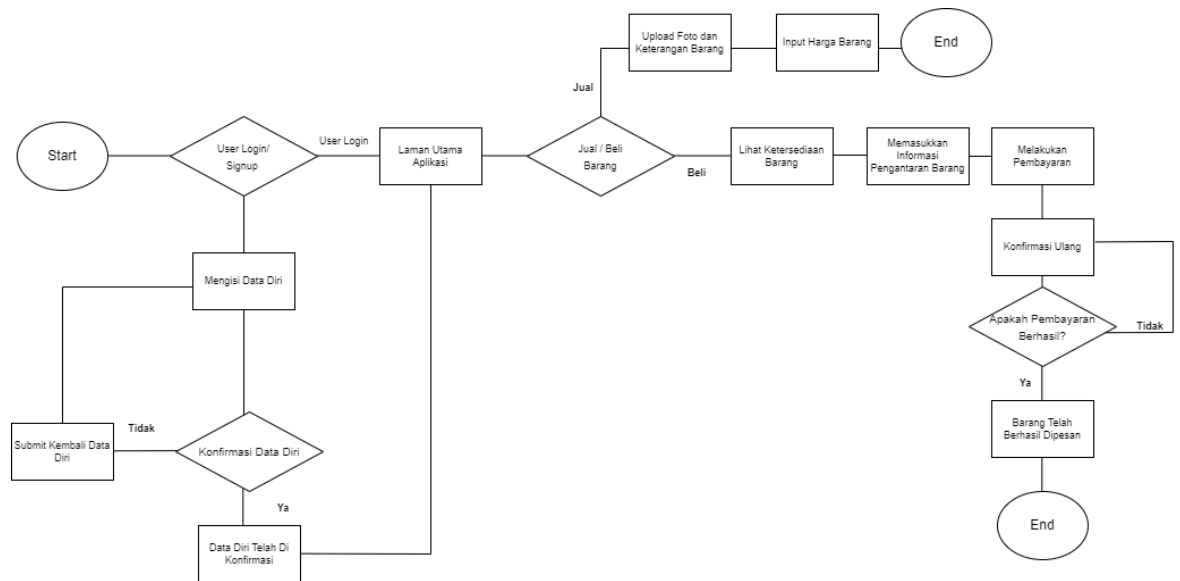
Fitur yang terdapat di dalam Item Menu ada empat yaitu Home Activity, Belanja Activity, Profile Activity dan Book Activity. Item menu ini sebagai penyambung ke main menu. Dengan adanya item menu tersebut maka akan muncul tampilan vector pada buttom navigation.

## **2.2 Konsep Bisnis Aplikasi Studepository**

Terutamanya aplikasi Studepository berbasis penjualan atau pembelian buku-buku yang dipasarkan dalam aplikasi. Tetapi alasan kami untuk memutuskan buku-buku sebagai produk yang diutamakan dalam aplikasi Studepository, dapat memberi untuk mendukung setiap kalangan mahasiswa membaca dan memanfaatkan penggunaan buku. Pada aplikasi e-commerce secara umumnya tidak mengutamakan penjualan buku dan sulit untuk mencari buku berkondisi bekas dengan harga yang terjangkau. Dengan adanya Perpustakaan Nasional di Indonesia, kami menerima sebuah gagasan dari tujuan Perpustakaan Nasional. Oleh pelayanan Perpustakaan Nasional walaupun seiringnya terjadi suatu pandemi yang dapat mengurangi kunjungan, Perpustakaan Nasional telah meluncurkan aplikasi yang dapat memberi aksesibilitas kepada pengguna jika ingin melakukan sebuah peminjaman buku. Seluruh interaksi dilaksanakan secara online sesaat dilayani oleh pustakawan. Dari perubahan dan inovasi yang cukup signifikan dilakukan oleh Perpustakaan Nasional, telah ditunjukkan bahwa pelayanan tetap berjalan walaupun waktu operasional dan aksesibilitas secara tatap muka dibatasi.

Dengan ini kami mengembangkan aplikasi berkonsep seperti e-commerce tetapi hanya berfokus bagi buku-buku yang berkondisi bekas.

## 2.3 Proses Bisnis Aplikasi Studepository



Berikutnya pada diagram diatas merupakan proses bisnis aplikasi “Studepository”.

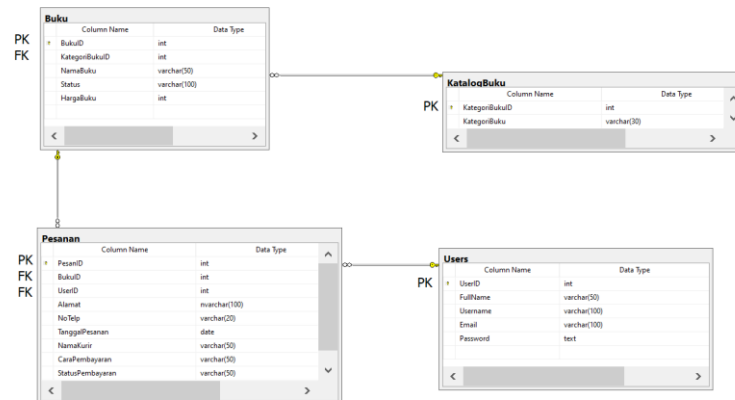
Proses pertama dalam proses bisnis dimulai dengan “start”. Sesaat pengguna sudah memiliki sebuah akun, pengguna dapat ditelusuri langsung dengan “User Login” ataupun “Admin Login” kepada proses selanjutnya. Jika pengguna tidak memiliki akun sebagai User maka pengguna dibutuhkan untuk mendaftar sebuah akun melalui proses “Signup”. Dalam pendaftaran akun, pengguna akan diperlukan untuk mengisi data - data diri seperti nama, e-mail, dan password sesuai kebutuhan sistem agar pengguna dapat masuk kepada proses selanjutnya, sebelum pendaftaran akun selesai pengguna dapat mengkonfirmasi data yang telah di isi pengguna akan ditelusuri kepada laman utama aplikasi, jika pengguna membatalkannya agar pengguna dapat mengisi data kembali.

Sesudah pengguna dapat berhasil masuk, pengguna dapat memutuskan menjadi pembeli atau penjual melalui fitur “Jual” atau “Beli”. Jika pengguna ingin menjual suatu barang maka fitur “Jual” dipilihkan, dan dilanjutkan kepada proses mengupload foto barang, keterangan barang, dan menginput harga barang.

Sedangkan bagi pengguna yang ingin membeli suatu barang maka pengguna akan memilih “Beli”, proses yang dihadapi berikutnya merupakan melihat ketersediaan barang, jika pengguna ingin membeli barang tersebut maka berikutnya pengguna diperlukan untuk “Masukkan Informasi Pengantaran Barang” seperti alamat tujuan dan cara pengiriman. Sesudah pengguna menyelesaikan proses “Memasukkan Informasi Pengantaran Barang”, pengguna diharuskan untuk melakukan transaksi untuk membayar barang yang dibeli melalui proses “Melakukan Pembayaran”.

Sesudah “Melakukan Pembayaran” pengguna akan memutuskan untuk konfirmasi dalam proses “Konfirmasi Ulang” jika pengguna ingin menyelesaikan pembelian maka pengguna memilih “Ya” dan status pembelian barang dibelikan telah berhasil, namun sebaliknya yaitu “Tidak” jika pengguna ingin membatalkan pembelian.

## 2.4 ERD Diagram Studepository



Berikutnya merupakan gambar diagram ERD aplikasi “Studepository”.

Terlihat empat tabel - tabel yang terdapat dari ERD aplikasi “Studepository”. Dalam gambar diagram atas terdapat empat tabel yang berhubungan yaitu tabel “Buku”, “Pesanan”, “KatalogBuku”, dan “Users”.

Data dari tabel masing - masing saling membutuhkan maka dari itu ke-empat tabel tersebut berhubungan satu sama lain serta fungsi tabel - tabel telah disinkronisasikan. Data dari tabel “KategoriBuku” dibutuhkan oleh tabel “Buku” yaitu KatalogBukuID. Data BukuID dan UserID dari tabel “Buku” dan “Users” dibutuhkan oleh tabel “Pesanan”.

Primary key terdapat dalam kolom AdminID dalam tabel “AdminStudepository”, kolom BukuID dalam tabel “Buku”, kolom KatalogBukuID dalam tabel “KategoriBuku”, kolom PesanID dalam tabel “Pesanan”, dan kolom UserID dalam tabel “Users”.

## 2.5 Capture Layout Aplikasi “Studepository”

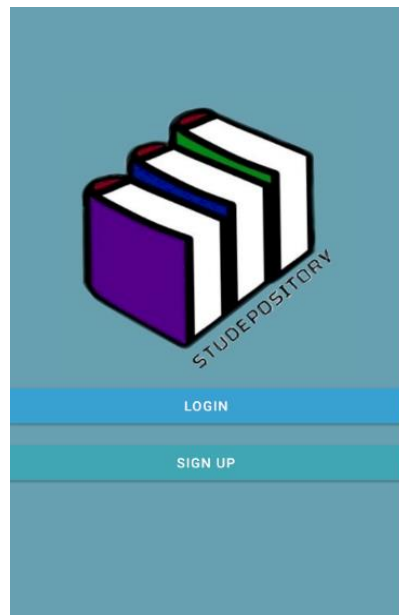
### 2.5.0 Cover (Splashscreen)

Pada tampilan awal terdapat Splashscreen. Splashscreen merupakan tampilan awal yang muncul pada saat membuka aplikasi Studepository. Pada tampilan Splashscreen dibawah terdapat sebuah logo Studepository.



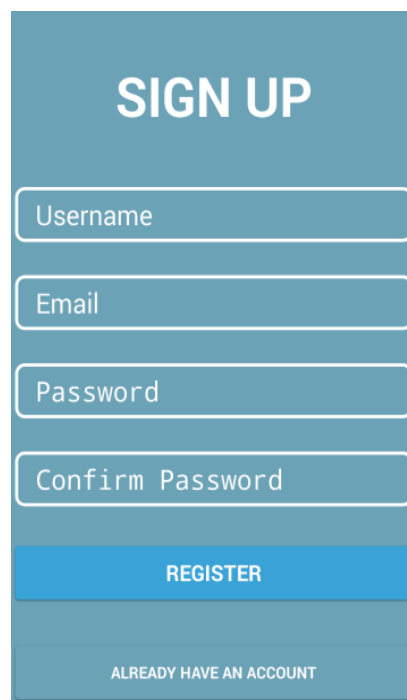
### 2.5.1 Main Activity

Terdapat dua fitur setelah splashscreen ditampilkan dalam main activity yaitu fitur log in jika pengguna telah memiliki sebuah akun atau mendaftar dan sign up bagi pengguna baru. Kedua fitur yang terdapat dalam layout main activity dapat memberi akses bagi pengguna kepada home activity. Secara teknis pengguna diharapkan untuk mengingat username dan password yang telah di daftar untuk mengakses home activity dan dapat melakukan aktivitas dalam aplikasi.



### 2.5.2 Sign Up

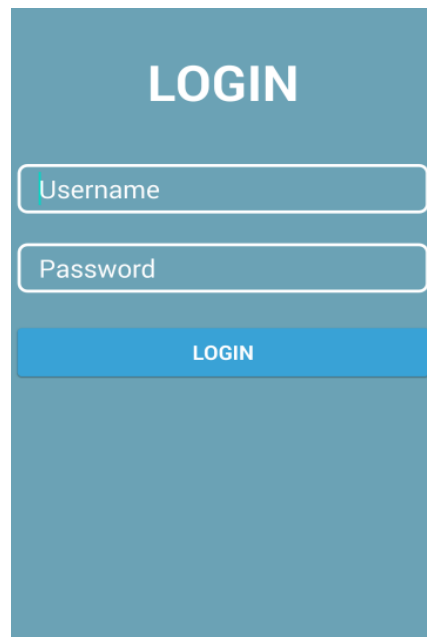
Setelah tampilan Login dan Sign Up. Jika user tersebut belum mempunyai akun maka user tersebut harus membuat akun terlebih dahulu untuk masuk ke tampilan menu home activity aplikasi Studepository. Pada tampilan Sign Up terdapat beberapa data diri yang harus diisi oleh para user. Data diri seperti Username, Email, Password, dan Confirm Password. Jika user tersebut sudah mengisi ketentuan data diri tersebut, user langsung dapat menekan tombol Register kemudian akan diarahkan otomatis ke Log In.

The image shows a mobile application sign up screen. At the top, the text 'SIGN UP' is displayed in large white letters on a blue background. Below this, there are four input fields with rounded corners and white borders, each containing a placeholder text: 'Username', 'Email', 'Password', and 'Confirm Password'. Below the input fields, there is a blue button with the text 'REGISTER' in white. At the bottom, there is a link that says 'ALREADY HAVE AN ACCOUNT' in small white text on a blue background.



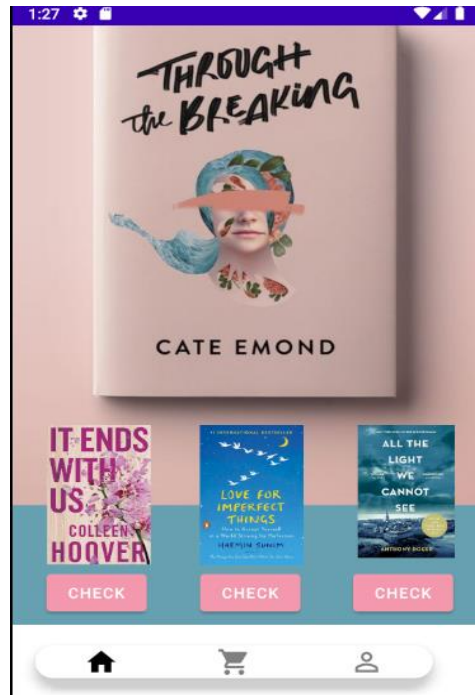
### 2.5.3 Log In

Jika user telah melakukan Sign Up maka user dapat melakukan Log In dalam aplikasi Studepository sebelum dapat menelusuri ke dalam laman menu Home Activity aplikasi Studepository. Terdapat Username dan Password yang dapat diisi oleh para user untuk melakukan Log In terlebih dahulu. Para user diharapkan untuk mengisi sesuai Username dan Password yang telah di simpan dari tahap Sign Up yang user telah lakukan.

A mobile application login screen with a teal background. At the top, the word "LOGIN" is displayed in white, bold, uppercase letters. Below it are two white input fields with rounded corners and thin white borders. The first field is labeled "Username" and the second is labeled "Password". Below these fields is a solid blue button with the word "LOGIN" in white, uppercase letters. The bottom half of the screen is a solid teal rectangle.

### 2.5.4 Home Activity

Tampilan di bawah merupakan tampilan menu pada Home Activity. Pada tampilan Home Activity terdapat tampilan awal buku-buku bekas yang dijual di aplikasi Studepository. Dalam Home Activity navigasi bar dapat ditemukan yaitu home activity tersendiri yang berlambangkan rumah, shop activity berlambangkan trolley, dan profile activity melambangkan orang.



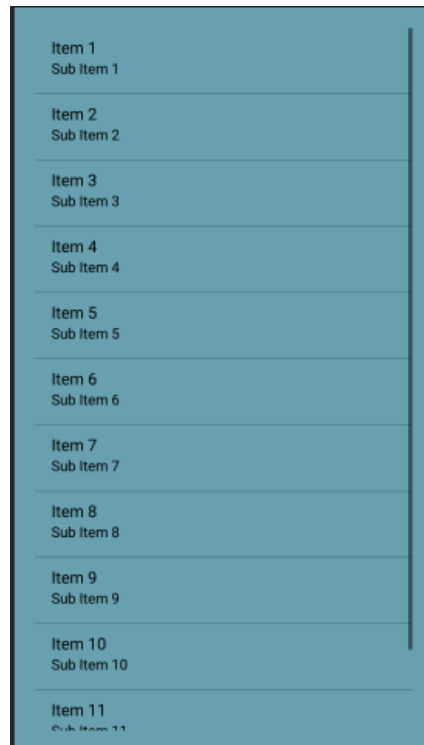
### 2.5.5 Add Book

Dalam layout AddBook terdapat fitur-fitur dimana para pengguna dapat mengisi judul buku, penulis buku, dan toko yang menjual buku yang terpilih sendiri. Untuk melakukan pencarian buku yang terpilih oleh para pengguna.

A screenshot of a mobile application interface titled 'Checkout Buku'. The background is a solid teal color. The title 'Checkout Buku' is displayed in white text at the top. Below the title, there are three input fields with labels: 'Nama Buku:', 'Nama Penulis:', and 'Toko:'. Each label is followed by a horizontal line representing the input field. At the bottom center of the form is a pink button with the word 'SEARCH' in white capital letters.

### 2.5.6 Item List Book

Merupakan layout dimana seluruh data terisi yang merupakan nama buku, penulis, dan nama toko telah tercatat dan ditampilkan secara berurutan dalam layout tersebut.



Item 1 Sub Item 1
Item 2 Sub Item 2
Item 3 Sub Item 3
Item 4 Sub Item 4
Item 5 Sub Item 5
Item 6 Sub Item 6
Item 7 Sub Item 7
Item 8 Sub Item 8
Item 9 Sub Item 9
Item 10 Sub Item 10
Item 11 Sub Item 11

### 2.5.7 Belanja Activity

Fitur dimana pengguna dapat menelusuri deskripsi produk, dalam Aplikasi Studepository pengguna dapat melihat rangkuman buku dan nama penulis yang terpilih serta dapat melihat titik tempat lokasi penjual oleh button Cek Lokasi.



### 2.5.8 MapView

Menampilkan titik lokasi produk dengan fitur map yang dapat ditelusuri sesudah pengguna memencet fitur button cek lokasi. Map tersebut ditampilkan secara *default* yang hanya memberi map dalam rupa 2D.



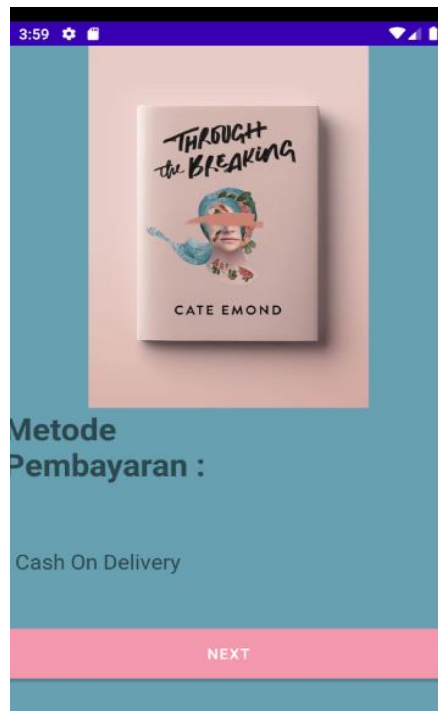
### 2.5.9 Checkout Details

Sebelum tahap checkout para pengguna dapat melihat informasi lebih lanjut mengenai buku yang ingin dibeli oleh pengguna tersendiri. Layout checkout details terdapat nama buku, deskripsi buku, dan harga yang dipasarkan. Pengguna dapat melanjutkan kepada bagian fitur checkout dengan memencet button checkout yang berada di layout.



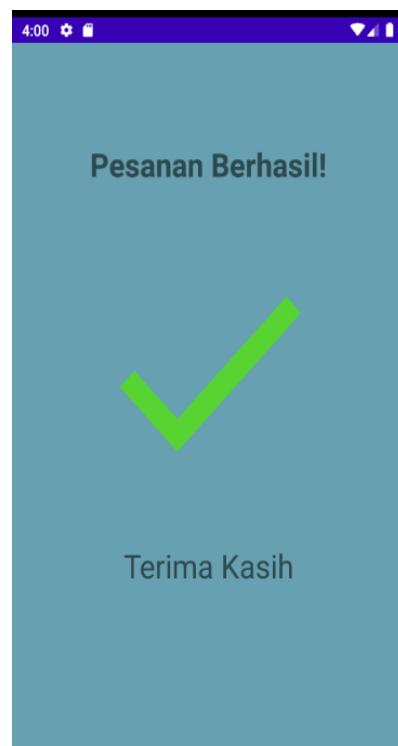
### 2.5.10 Payment Activity

Setelah pengguna dapat melakukan pemilihan buku melalui checkout, untuk bertransaksi pengguna dapat melakukan proses tersebut dalam laman payment activity. Dalam payment activity terdapat penjelasan metode pembayaran yang dapat dilakukan yaitu Cash on Delivery (COD). Pengguna lalu memencet button next untuk melanjutkan kepada laman konfirmasi pembayaran.



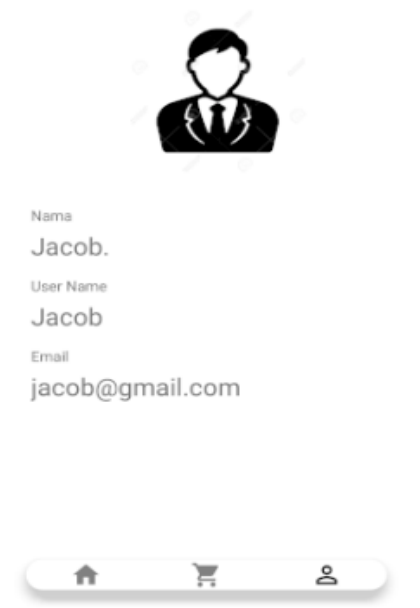
### 2.5.11 Finish Payment

Jika pengguna telah melakukan konfirmasi dalam tahap payment activity, maka layout finish payment telah mengindikasikan kepada pengguna jika pembayaran telah dikonfirmasi dan berhasil. Menampilkan logo centang hijau dalam layout finish payment.



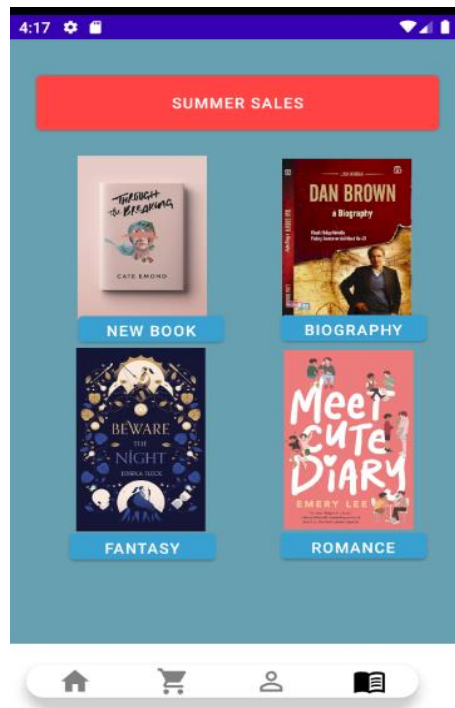
### 2.5.12 Profile Activity

Profile activity menampilkan data dari akun pengguna yang di input, didalam activity pengguna dapat melihat nama lengkap, username, dan juga email.



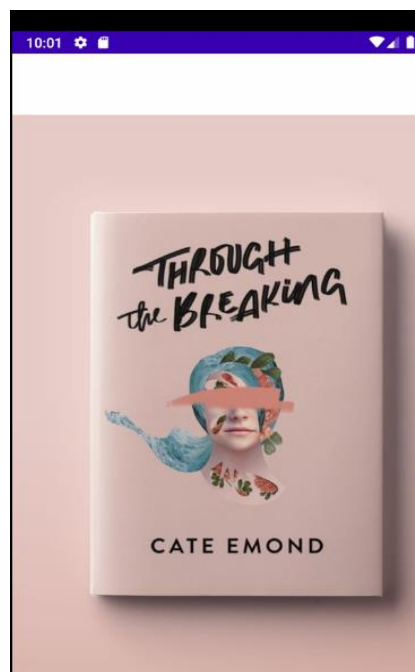
### 2.5.13 Categories One

Bagian layout catalog setiap buku-buku yang termasuk “Summer Sales” akan ditampilkan dalam laman tersebut. Terdapat empat buku yang ditampilkan dalam layout dengan tema buku masing-masing.



#### 2.5.14 Fragment 1 (ViewPager)

Tampilan buku masing-masing yang dapat menampilkan sampul buku yang terpilih oleh para pengguna aplikasi dan dapat melihat sampul buku secara detail.



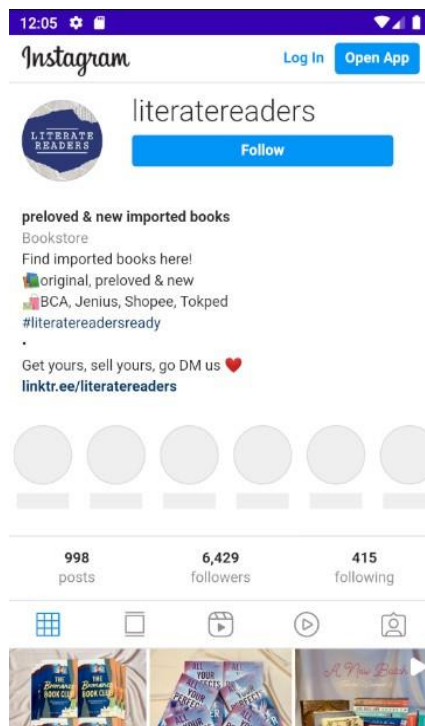


### 2.5.15 Fragment 2 (WebView)

Tampilan laman dari Instagram diatas merupakan pemilihan kami dalam WebView dengan link. WebView dalam aplikasi Studepository dapat memberi lihat tampilan laman link yang disimpan dalam webview kepada pengguna dari browser web.

Link WebView Instagram :

[https://instagram.com/literatereaders?utm\\_medium=copy\\_link](https://instagram.com/literatereaders?utm_medium=copy_link).



### 2.5.16 Item Menu

Tampilan pada Item Menu terdiri dari sebuah empat tabel yang nantinya akan tersambung ke masing-masing bottom navigation. Item Menu dibuat dengan vector. Terdapat vector home, shop. person, dan book.



## 2.6 Capture Coding yang Digunakan Dalam Aplikasi “Studepository”

### 2.6.0 SplashScreen

```
this.requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.activity_splash_screen);

final Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        startActivity(new Intent(getApplicationContext(), MainActivity.class));
        finish();
    }
}, delayMillis: 3000L); //3000 L = 3 detik
}
```

Code diatas digunakan untuk menambahkan splash screen pada aplikasi. Splash screen tersebut akan muncul saat aplikasi pertama kali dibuka. Desain yang kami gunakan mempunyai tujuan untuk meningkatkan user experience dalam menggunakan aplikasi Studepository.

### 2.6.1 MainActivity

```

Button btnsignup = (Button)findViewById(R.id.btn_signup);
btnsignup.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(getApplicationContext(), SignUp.class);
        startActivity(i);
    }
});

Button btnlogin = (Button)findViewById(R.id.btn_login);
btnlogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(getApplicationContext(), Login.class);
        startActivity(i);
    }
});

```

Code ini digunakan untuk melakukan pemanggilan komponen dan pemindahan (*intent*) fungsi *button activity* dari class MainActivity ke class login dan signup.

### 2.6.2 Sign Up

```

23 Username = findViewById(R.id.username);
24 Email = findViewById(R.id.email);
25 Password = findViewById(R.id.password);
26 repassword = findViewById(R.id.repassword);
27 Signup = findViewById(R.id.signup);
28 Login = findViewById(R.id.login);
29 DB = new DBHelper(context);
30
31 Signup.setOnClickListener(new View.OnClickListener() {
32     @Override
33     public void onClick(View v) {
34         String User = Username.getText().toString();
35         String email = Email.getText().toString();
36         String Pass = Password.getText().toString();
37         String repass = repassword.getText().toString();
38
39         if(TextUtils.isEmpty(User) || TextUtils.isEmpty(email) || TextUtils.isEmpty(Pass) || TextUtils.isEmpty(repass))
40             Toast.makeText(context, "All fields Required", Toast.LENGTH_SHORT).show();
41         else{
42             if(Pass.equals(repass)){
43                 Boolean checkuser = DB.checkusername(User);
44                 if(checkuser==false){
45                     Boolean insert = DB.insertData(User, Pass);
46                     if(insert==true){
47                         Toast.makeText(context, "Registered Successfully", Toast.LENGTH_SHORT).show();
48                         Intent intent = new Intent(getApplicationContext(), Login.class);
49                         startActivity(intent);
50                     }else{
51                         Toast.makeText(context, "Registration Failed", Toast.LENGTH_SHORT).show();
52                     }
53                 }else{
54                     Toast.makeText(context, "User already exists", Toast.LENGTH_SHORT).show();
55                 }
56             }
57         }
58     }
59 }

```

Code diatas digunakan untuk melakukan *signup*, dimana di dalam *code* kami menggunakan *sqlite database*. Pengguna harus meregistrasikan diri untuk masuk ke tampilan *login*. Pengguna juga tidak bisa mendaftarkan diri dengan akun yang sama, karena data yang sama akan di *decline* dari sistem.

C

```

    }else{
        Toast.makeText( context: SignUp.this, text: "Passwords are not matching", Toast.LENGTH_SHORT).show();
    }
}
});
Login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), Login.class);
        startActivity(intent);
    }
});

```

Code diatas digunakan ketika layout register ditekan, maka kami akan menghubungkan ke class login untuk memasukkan data diri yang telah diregistrasikan di class signup ke akun studepository.

### 2.6.3 Login

```

username = findViewById(R.id.username1);
password = findViewById(R.id.password1);
login = findViewById(R.id.login1);
DB= new DBHelper( context: this);

login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String user = username.getText().toString();
        String pass = password.getText().toString();

        if(TextUtils.isEmpty(user) || TextUtils.isEmpty(pass))
            Toast.makeText( context: Login.this, text: "All fields Required", Toast.LENGTH_SHORT).show();
        else{
            Boolean checkuserpass = DB.checkusernamepassword(user, pass);
            if(checkuserpass==true){
                Toast.makeText( context: Login.this, text: "Login Successful", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(getApplicationContext(), MainPageOne.class);
                startActivity(intent);
            }else{
                Toast.makeText( context: Login.this, text: "Login Failed", Toast.LENGTH_SHORT).show();
            }
        }
    }
});

```

Code diatas digunakan untuk melakukan *login* yang dilakukan oleh pengguna (*user*) *code* tersebut menggunakan *sqlite database*, dimana pengguna perlu melakukan registrasi *username* dan *password* sehingga pada saat melakukan *login*, data pengguna dapat dideteksi.

### 2.6.4 DBHelper

```
public class DBHelper extends SQLiteOpenHelper {

    public static final String DBNAME = "login.db";
    public DBHelper(Context context) { super(context, name: "login.db", factory: null, version: 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table users(username TEXT primary key, password TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("drop table if exists users");
    }

    public Boolean insertData(String username, String password){
        SQLiteDatabase db= this.getWritableDatabase();
        ContentValues values = new ContentValues();

        values.put("Username", username);
        values.put("Password", password);

        long result= db.insert( table: "users", nullColumnHack: null, values);
        if (result==1) return true;
        else
            return false;
    }
}
```

Code diatas digunakan untuk membaca dan memasukkan informasi dari signup dan login ke database. Hal ini mempermudah kami untuk mengakses data dari setiap pengguna.

```
public Boolean checkusername(String username) {
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery( sql: "select * from users where username=?", new String[]{username});
    if(cursor.getCount()>0)
        return true;
    else
        return false;
}

public Boolean checkusernamepassword(String username, String password) {
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery( sql: "select * from users where username=? and password=?", new String[]{username, password});
    if (cursor.getCount() > 0)
        return true;
    else
        return false;
}
```

Code diatas digunakan untuk mengakses data dari hasil *database signup* dan *login*. Hal ini membantu untuk mengakses dari satu baris data ke data yang lainnya.

### 2.6.4 MainMenu

```

navigationView = findViewById(R.id.bottom_nav);
getSupportFragmentManager().beginTransaction().replace(R.id.framefragment, new HomeActivity()).commit();
navigationView.setSelectedItemId(R.id.nav_home);

navigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
        Fragment fragment = null;
        switch (menuItem.getItemId()){
            case R.id.nav_home:
                fragment = new HomeActivity();
                break;

            case R.id.nav_shop:
                fragment = new ShopActivity();
                break;

            case R.id.nav_person:
                fragment = new ProfileActivity();
                break;
        }
        getSupportFragmentManager().beginTransaction().replace(R.id.framefragment, fragment).commit();
        return true;
    }
});

```

Code ini digunakan untuk mengatur bottom navigation yang terdapat di HomeActivity. Sehingga pengguna bisa mengklik button tersebut. Terdapat empat bottom navigation di main menu yaitu Home Activity, Belanja Activity, Profile Activity dan Book Activity.

### 2.6.5 HomeActivity

```

@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.activity_home, container, attachToRoot: false);
    return v;
}

```

Code ini digunakan sebagai bottom navigation pada Home Activity. Code container tersebut menampilkan bottom navigation home activity saat melakukan run. Saat di klik pada emulator home activity maka akan tampilan halaman home activity. Dimana pada tampilan Home Activity tersebut berisi tiga jenis buku bekas yang dijual pada aplikasi Studepository.

## 2.6.6 AddBook

```
package com.prak8_00000052746.project;

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

import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.AuthFailureError;
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.prak8_00000052746.project.ItemListBook;
import com.prak8_00000052746.project.R;

import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class AddBook extends AppCompatActivity {


    EditText edtNama, edtPenulis, edtNamaToko;
    Button btnSearch;
    String nama, penulis, toko;
    String url_tambah_buku = "https://feliciahanna.000webhostapp.com/addBuku.php";
```

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

    btnSearch = (Button)findViewById(R.id.btnSearch);
    edtNama = (EditText)findViewById(R.id.edtNama);
    edtPenulis = (EditText)findViewById(R.id.edtPenulis);
    edtNamaToko = (EditText)findViewById(R.id.edtNamaToko);

    btnSearch.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            nama = edtNama.getText().toString();
            penulis = edtPenulis.getText().toString();
            toko = edtNamaToko.getText().toString();

            RequestQueue queue = Volley.newRequestQueue( context: AddBook.this);

            StringRequest stringRequest = new StringRequest(Request.Method.POST, url_tambah_buku, new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject jsonObj = new JSONObject(response);
                        int sukses = jsonObj.getInt( "name: "success");
                         (sukses == 1) {

                        }else{

                        }
                    } catch (Exception ex) {
                        Log.e( "tag: "Error", ex.toString());
                        ex.printStackTrace();
                    }
                }
            });
```

```

    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("tag", "Error", error.getMessage());
            Toast.makeText(context, AddBook.this, "Check Your Internet Connection", Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<>();
            params.put("nama", nama);
            params.put("penulis", penulis);
            params.put("toko", toko);
            return params;
        }

        @Override
        public Map<String, String> getHeaders() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("Content-Type", "application/x-www-form-urlencoded");
            return params;
        }
    };
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(stringRequest);
    Toast.makeText(getApplicationContext(), "Data Berhasil Dimasukan", Toast.LENGTH_SHORT).show();
    Intent a = new Intent(getApplicationContext(), ItemListBook.class);
    startActivity(a);
}

});
}
}

```

Untuk melakukan penyambungan dalam layout AddBook berikut diatas merupakan algoritma yang terdapat dalam layout AddBook agar berfungsi. Fitur intent, bundle, log, view, button, edittext, dan toast dapat di import dalam java AddBook agar setiap widget dapat bekerja. Kami menggunakan volley sebagai pendukung pemindahan data yang di input dalam layout AddBook kepada database di phpmyadmin oleh fungsi web-hosting. JSON digunakan dalam proses AddBook untuk menyimpan data dalam layout ItemListBook. Bagi map dan hashmap digunakan untuk program dapat mencatat data dari objek yang ingin digunakan. ID fitur yang digunakan dalam layout adalah btnSearch, edtNama, edtPenulis, dan edtNamaToko.



## 2.6.7 Item List Book

```

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.widget.ListAdapter;
import android.widget.ListView;

import android.widget.SimpleAdapter;
import android.widget.Toast;

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 org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;

public class ItemListBook extends AppCompatActivity {
    ListView listView;
    ArrayList<HashMap<String,String>> list_buku;
    String url_get_buku = "https://feliciahanna.000webhostapp.com/getBuku.php";

    private static final String TAG_Buku="data";
    private static final String TAG_Nama="nama";
    private static final String TAG_Penulis="penulis";
    private static final String TAG_Toko="toko";

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

        list_buku = new ArrayList<>();
        listView = findViewById(android.R.id.list);

        RequestQueue queue = Volley.newRequestQueue( context: ItemListBook.this);
        StringRequest stringRequest = new StringRequest(Request.Method.POST, url_get_buku, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObj = new JSONObject(response);
                    JSONArray book = jsonObj.getJSONArray(TAG_Buku);

                    for (int i = 0; i < book.length(); i++) {
                        JSONObject a = book.getJSONObject(i);
                        String nama = a.getString(TAG_Nama);
                        String penulis = a.getString(TAG_Penulis);
                        String toko = a.getString(TAG_Toko);

                        HashMap<String, String> map = new HashMap<>();
                        map.put("nama", nama);
                        map.put("penulis", penulis);
                        map.put("toko", toko);

                        list_buku.add(map);
                        String[] from = {"nama", "penulis", "toko"};
                        int[] to = {R.id.edtNama, R.id.edtPenulis, R.id.edtNamaToko};

                        ListAdapter adapter = new SimpleAdapter(
                            context: ItemListBook.this, list_buku, R.layout.itemlist, from, to);
                        listView.setAdapter(adapter);
                    }
                } catch (Exception ex) {
                    Log.e( tag: "Error", ex.toString());
                }
            }
        }
    ), new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e( tag: "Error", error.getMessage());
            Toast.makeText( context: ItemListBook.this, text: "Check Your Internet Connection", Toast.LENGTH_SHORT).show();
            finish();
        }
    }
    );
    queue.add(stringRequest);
}

```

Sesudah AddBook, pengguna akan berlanjut kepada ItemListBook. Setelah pengguna telah mengisi data-data dalam layout AddBook. Data tersebut akan tercatat dalam listview sebagai daftar-daftar data yang telah dimasuki.

### 2.6.8 Belanja Activity

```
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

public class BelanjaActivity extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.activity_belanja, container, attachToRoot false);

        return v;
    }

    @Override
    public void onStart() {
        super.onStart();

        Button btnMaps = (Button) getActivity().findViewById(R.id.btnDetails);
        btnMaps.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getContext(), MapsActivity.class));
            }
        });
    }
}
```

Code yang digunakan diatas sama seperti Home Activity. Code tersebut berfungsi agar saat bottom navigation tersebut di klik maka akan memunculkan activity tampilan tersebut. Pada belanja activity tampilan awal yang digunakan yaitu tampilan buku beserta deskripsi dan harga buku tersebut. Button yang terdapat pada Java diatas berfungsi sebagai memindahkan dari satu activity ke activity lainnya. Dari button yang terdapat pada tampilan belanja activity yang memiliki id btnMaps dan pada start activity menggunakan intent maka akan pindah ke activity selanjutnya yaitu MapsActivity.

## 2.6.9 Maps Activity

```

1  package com.prak8_00000052746.project;
2
3  import ...
19
20  public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
21
22      private GoogleMap mMap;
23      private ActivityMapsBinding binding;
24
25      @Override
26      protected void onCreate(Bundle savedInstanceState) {
27          super.onCreate(savedInstanceState);
28
29          binding = ActivityMapsBinding.inflate(getLayoutInflater());
30          setContentView(binding.getRoot());
31
32          // Obtain the SupportMapFragment and get notified when the map is ready to be used.
33          SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
34              .findFragmentById(R.id.map);
35          mapFragment.getMapAsync(this);
36      }
37
38      @Override
39      public void onMapReady(GoogleMap googleMap) {
40          mMap = googleMap;
41
42          // Add a marker in Sydney and move the camera
43          LatLng sydney = new LatLng(-34, 151);
44          mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
45          mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
46
47          Button btnCheckDetails = (Button) findViewById(R.id.btnCheckDetails);
48          btnCheckDetails.setOnClickListener(new View.OnClickListener() {
49              @Override
50              public void onClick(View view) {
51                  startActivity(new Intent(getApplicationContext(), CheckoutDetails.class));
52              }
53          });
54      }

```

Code di atas digunakan untuk menampilkan google maps atau titik lokasi dari tempat buku tersebut dijual. Pada bagian LatLng terdapat latitude dan longitude yaitu sebagai jarak lokasi yang digunakan untuk ditampilkan pada maps. Tittle marker lokasi aplikasi Studepository yaitu "Title in Sydney". Terdapat button cek lokasi yang dapat sambungkan dengan code maps tersebut. Untuk memindahkan dari satu activity google maps ke checkout details kami menggunakan start activity intent.

### 2.6.10 Checkout Details

```
package com.prak8_00000052746.project;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class CheckoutDetails extends AppCompatActivity {

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

        Button btnCek= (Button)findViewById(R.id.btnDetails);
        btnCek.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(), Payment.class));
            }
        });
    }
}
```

Code di atas digunakan untuk melihat informasi lebih lanjut mengenai buku yang ingin dibeli oleh pengguna tersendiri. Dalam code tersebut terdapat Button dan intent untuk memunculkan dari activity CheckOut Details ke activity Payment.

### 2.6.11 Payment

```
package com.prak8_00000052746.project;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Payment extends AppCompatActivity {

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

        Button btnNext= (Button)findViewById(R.id.btnNext);
        btnNext.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(), FinishPayment.class));
            }
        });
    }
}
```

Code di atas digunakan untuk melakukan Payment yang dilakukan secara cash on delivery oleh penjual dan pembeli. Fitur yang di import merupakan intent, bundle,

view, dan button. Button pada Payment Activity ini yang memiliki id “btnNext” nantinya akan masuk ke dalam activity Finish Payment Class dengan menggunakan bantuan Intent.

### 2.6.12 Finish Payment

```
package com.prak8_00000052746.project;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class FinishPayment extends AppCompatActivity {

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

Code di atas digunakan untuk melakukan Payment yang nantinya akan menghasilkan output notifikasi pembayaran yang berhasil dilakukan.

### 2.6.13 Categories One

```
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

public class CategoriesOne extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.activity_categories_one, container, attachToRoot: false);

        return v;
    }

    @Override
    public void onStart() {
        super.onStart();

        Button btnpager = (Button) getActivity().findViewById(R.id.button2);
        btnpager.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getContext(), ViewPager.class));
            }
        });
    }
}
```

Code di atas digunakan untuk melihat kategori dari setiap buku. Dimana terdapat 5 button yaitu button Summer Sales untuk melihat detail dari buku-buku yang ada, kemudian terdapat 4 button untuk setiap kategori buku yang tersedia yaitu New Book, Biography, Fantasy dan Romance dimana ketika button-button tersebut di klik salah satunya akan menampilkan banyak buku yang terdapat dalam kategori tersebut.

#### 2.6.14 Fragment 1 (View Pager)

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;
import android.os.Bundle;

public class ViewPager extends AppCompatActivity {
    androidx.viewpager.widget.ViewPager mViewPager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_pager);

        mViewPager = (androidx.viewpager.widget.ViewPager) findViewById(R.id.pager);
        mViewPager.setAdapter(new ViewPagerAdapten(getSupportFragmentManager()));
    }

    public class ViewPagerAdapten extends FragmentPagerAdapter {
        public ViewPagerAdapten(FragmentManager fm) {
            super(fm, BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
        }

        public Fragment getItem(int position) {
            if (position == 0) {
                return new Fragment1();
            } else {
                return new Fragment2();
            }
        }

        /**
         * Return the number of views available.
         */
        @Override
        public int getCount() { return 2; }
    }
}
```

Code diatas yaitu digunakan untuk code viewpager dimana code tersebut berfungsi untuk membuat banyak halaman yang dapat berganti dengan cara di swipe ke slide selanjutnya yaitu ke kanan.

### 2.6.15 Fragment 2 (WebView)

```
package com.prak8_00000052746.project;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ProgressBar;

public class Fragment2 extends Fragment {
    private WebView webView;
    private ProgressBar bar;

    @Nullable
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.activity_fragment2, container, attachToRoot false);

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

        webView.setWebViewClient(new WebViewClient());
        webView.getSettings().setJavaScriptEnabled(true);
        webView.loadUrl("https://www.instagram.com/literatereaders/");

        return v;
    }
}
```

Code di atas yaitu webview digunakan untuk memberi tampilan laman link dari akun sosmed yang disimpan dalam web view kepada pengguna dari browser web, hal ini ditujukan untuk memudahkan pengguna dalam menemukan akun sosial media kami tanpa harus membuka tab yang berbeda

### 2.6.16 Item Menu

```
navigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
        Fragment fragment = null;
        switch (menuItem.getItemId()){
            case R.id.nav_home:
                fragment = new HomeActivity();
                break;

            case R.id.nav_shop:
                fragment = new BelanjaActivity();
                break;

            case R.id.nav_person:
                fragment = new ProfileActivity();
                break;

            case R.id.nav_book:
                fragment = new CategoriesOne();
                break;
        }
        getSupportFragmentManager().beginTransaction().replace(R.id.framefragment, fragment).commit();
        return true;
    }
});
```

Code di atas terdiri dari sebuah empat tabel yang nantinya akan tersambung ke masing-masing bottom navigation. Item Menu dibuat dengan vector. Terdapat vector home, shop, person, dan book. Pada item menu tersebut untuk membuat menu Item yang nantinya akan tersambung ke main menu pembuatan java aplikasi studepository menggunakan switch, fragment, case dan break.



## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Aplikasi Studepository merupakan aplikasi jual buku bekas. Pada era saat ini masyarakat cenderung membeli buku-buku baru, Padahal masih banyak buku bekas yang masih bagus dan layak untuk dipakai. Tujuan dibuatnya aplikasi Studepository ini yaitu untuk memudahkan para book lovers untuk membeli buku bekas yang masih layak dipakai. Aplikasi kami menyediakan berbagai macam jenis buku bekas mulai dari genre romance, fiksi, fantasy dan masih banyak lainnya. Sehingga para pembeli nantinya dapat mencari buku sesuai dengan keinginannya. Aplikasi Studepository memiliki beberapa fitur-fitur seperti Home Activity, Belanja Activity, Profile Activity dan Book Activity. Dimana didalam setiap bottom Navigation tersebut terdapat proses dari sebuah pencarian buku bekas hingga akhir pembayaran buku. Berdasarkan penjelasan kesimpulan aplikasi Studepository tersebut dapat ditarik kesimpulan sebagai berikut:

1. Pembangunan aplikasi berbasis android ini memudahkan bagi user pembeli untuk memperoleh informasi buku yang ditawarkan oleh Studepository, serta memfasilitasi kemudahan yang diberikan kepada konsumen dalam melakukan pemesanan secara online.
2. Aplikasi Studepository yang sudah dibangun ini berhasil menampilkan beberapa fitur seperti profile, home, catalog,
3. dengan adanya aplikasi Studepository masyarakat dapat dengan mudah mendapatkan informasi seputar buku second hand yang ditawarkan.

### 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

**Aulia Kharisma Putri** : Pembuatan Laporan, Pembuatan Categories, Merancang UI Aplikasi, Pembuatan Categories One, Pembuatan Fragment 1(ViewPager) dan Fragment 2 (WebView).

**Deandra Yasin Winarno** : Pembuatan Layout profile, Pembuatan layout Belanja Activity, Pembuatan Laporan, Editor Video

**Felicia Hanna Ratna** : Pembuatan halaman Login-Register, merancang desain user interface aplikasi, SQLite, Pembuatan Laporan

**Kumala Dewi Chandra Asih**: Pembuatan Bottom Navigation (HomeActivity, Belanja Activity, Profile Activity, Book Activity), Pembuatan Item menu, Pembuatan Google Maps, Pembuatan Checkout Details, Payment,dan Finish Payment, Pembuatan laporan.

**Yuki Sabrina Nakaya** : Pembuatan file-file PHP, Pembuatan halaman AddBook dan ItemList, Merancang logo aplikasi, Pembuatan diagram proses bisnis studepository, Pembuatan diagram ERD studepository.