

Rumah Coding Course

"Android Intermediate"

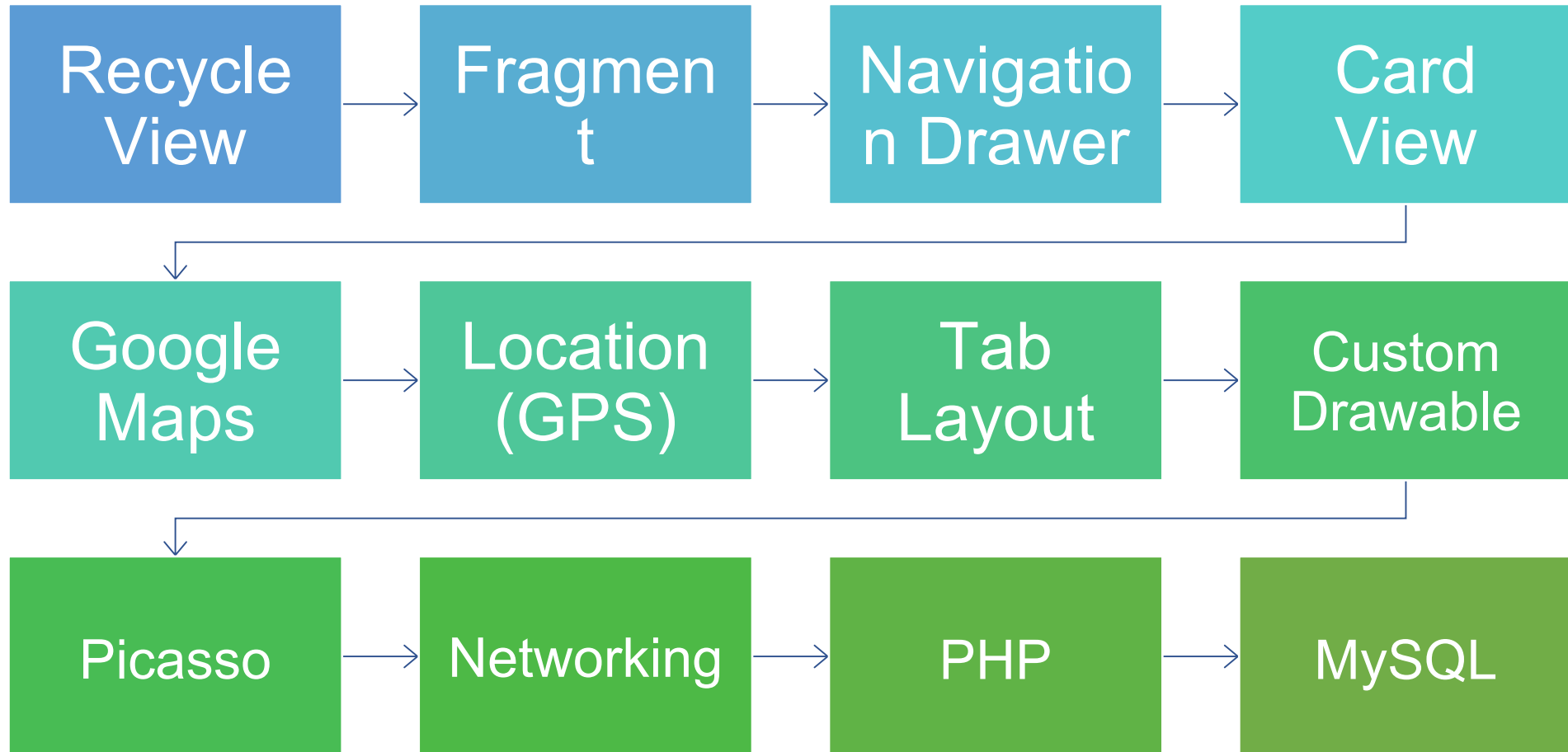
Studi Kasus Aplikasi "NearDeal"



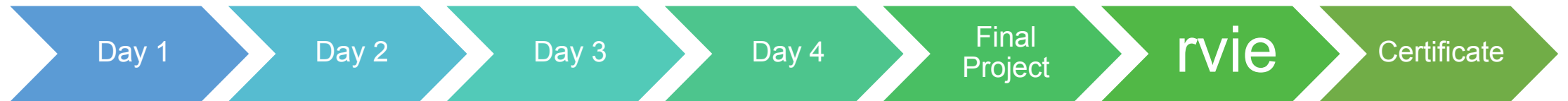
Day 1



Course Syllabus



Course Overview



Final Project Duration: **1 Week**



Course Breakdown

Day 1

- Fragment
- Recycle View
- Navigation Drawer
- Card View
- Create store list
- Create product list

Day 2

- Google Maps
- Tab Layout
- Create store maps

Day 3

- FCM
- Shared Preferences
- Android Notification

Day 4

- Custom Drawable
- Google Admob

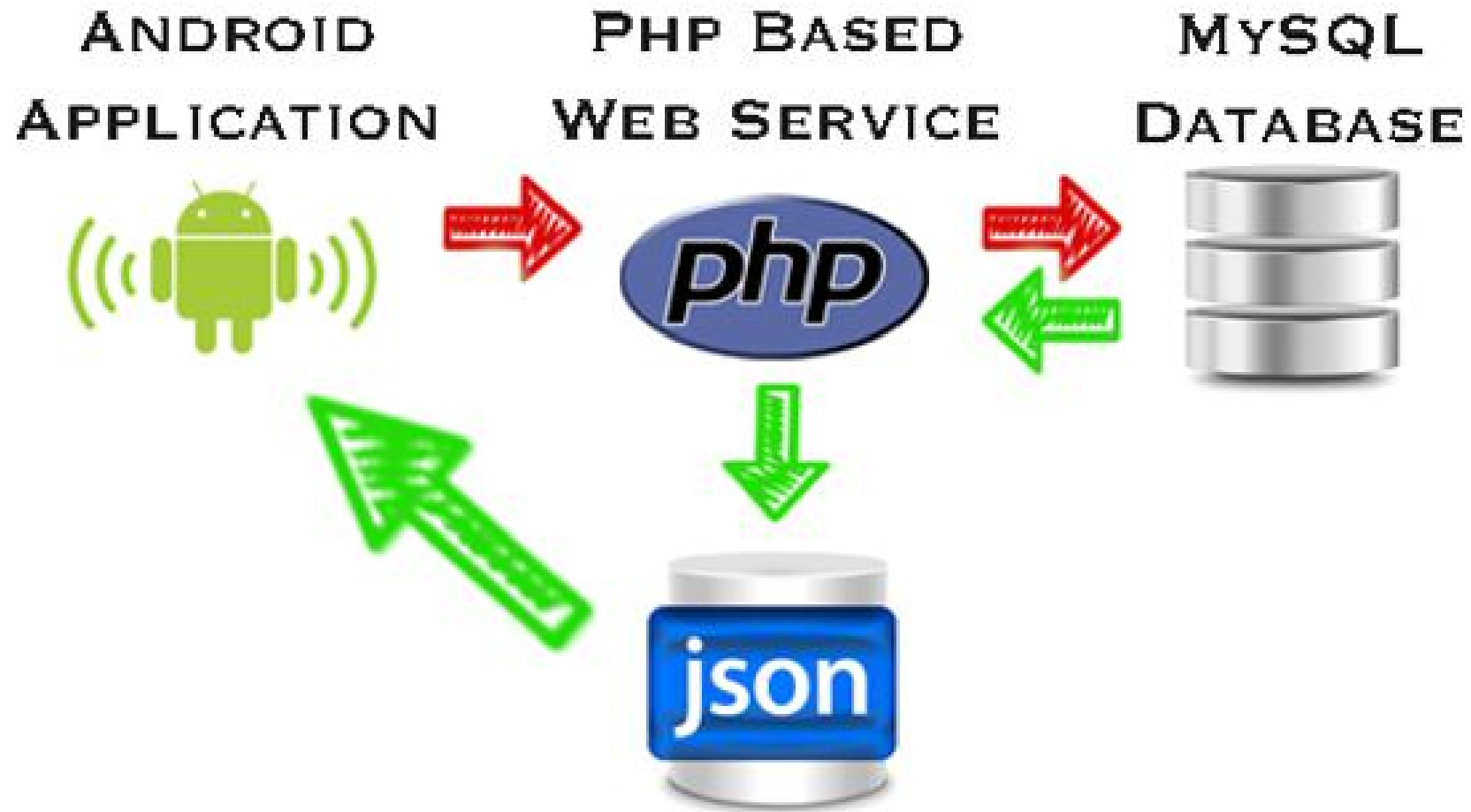


Development Tool





- JDK 7/8
- Android Studio
- Xampp/Wamp







Development Tool




List Store

 62% 21:08


 Near Deal  



D'Mall Depok
Belum Ada Deal!

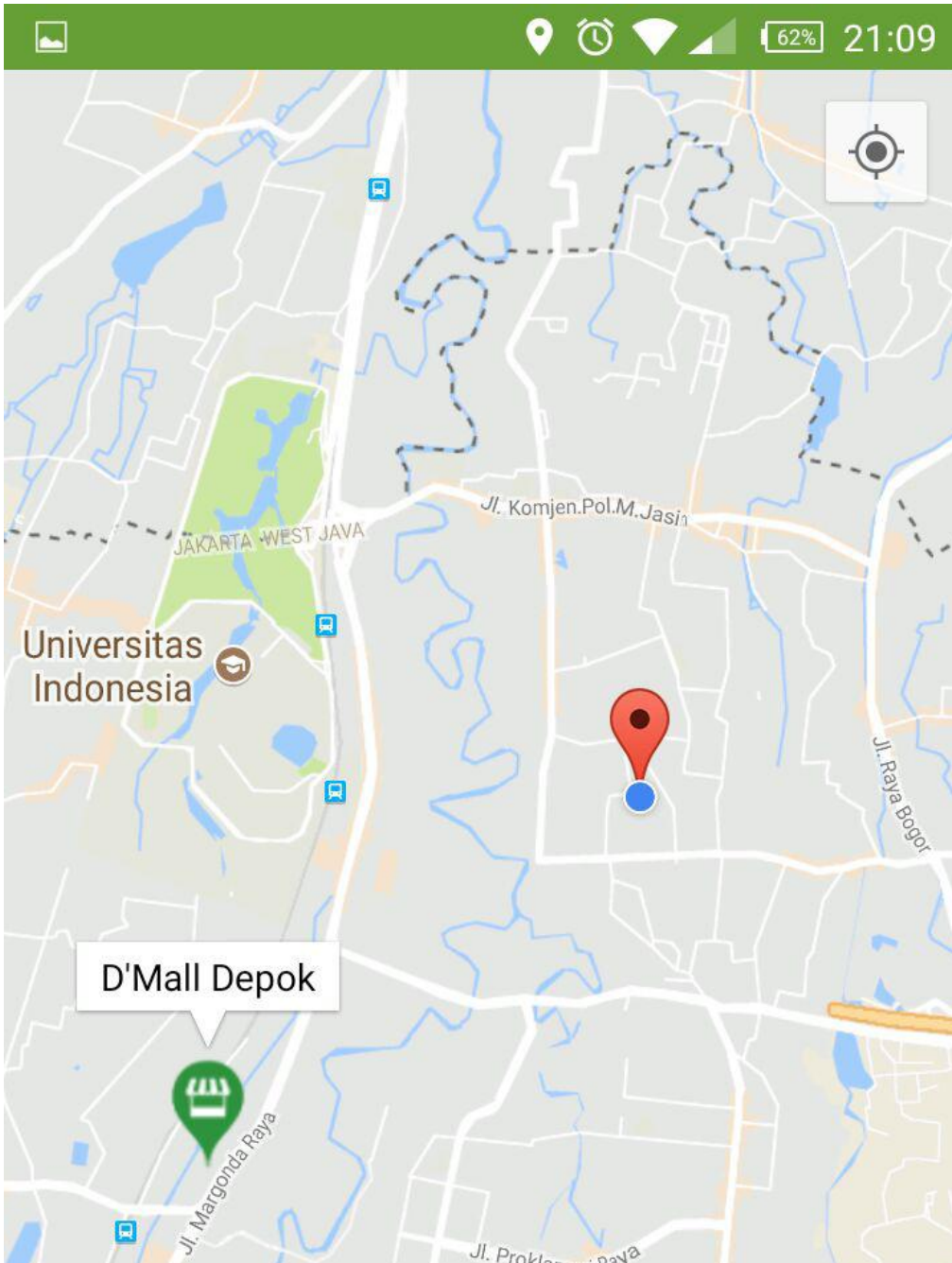


Ace Hardware Depok
Belum Ada Deal!



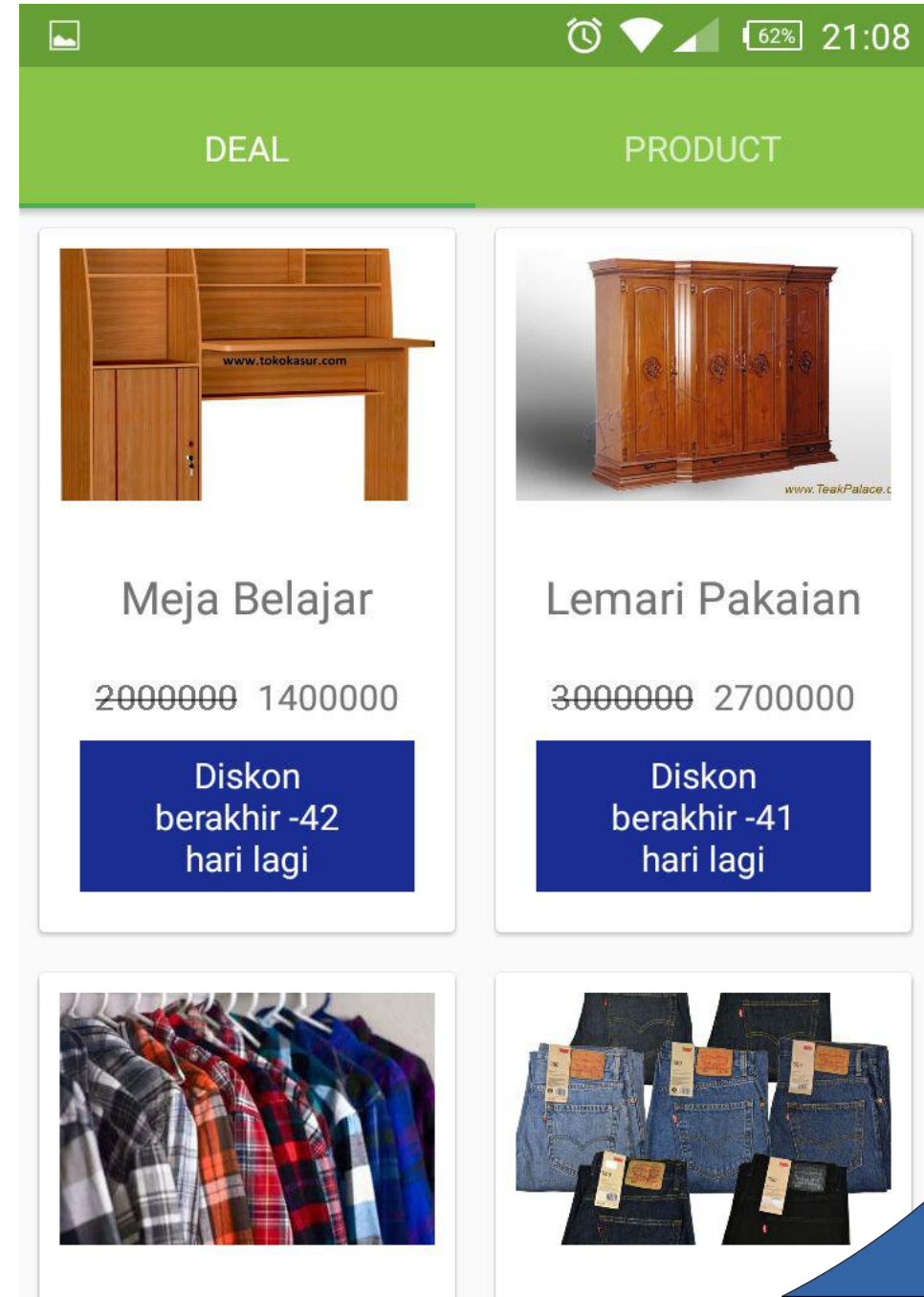
Gramedia Matraman
Belum Ada Deal!





Map Mode

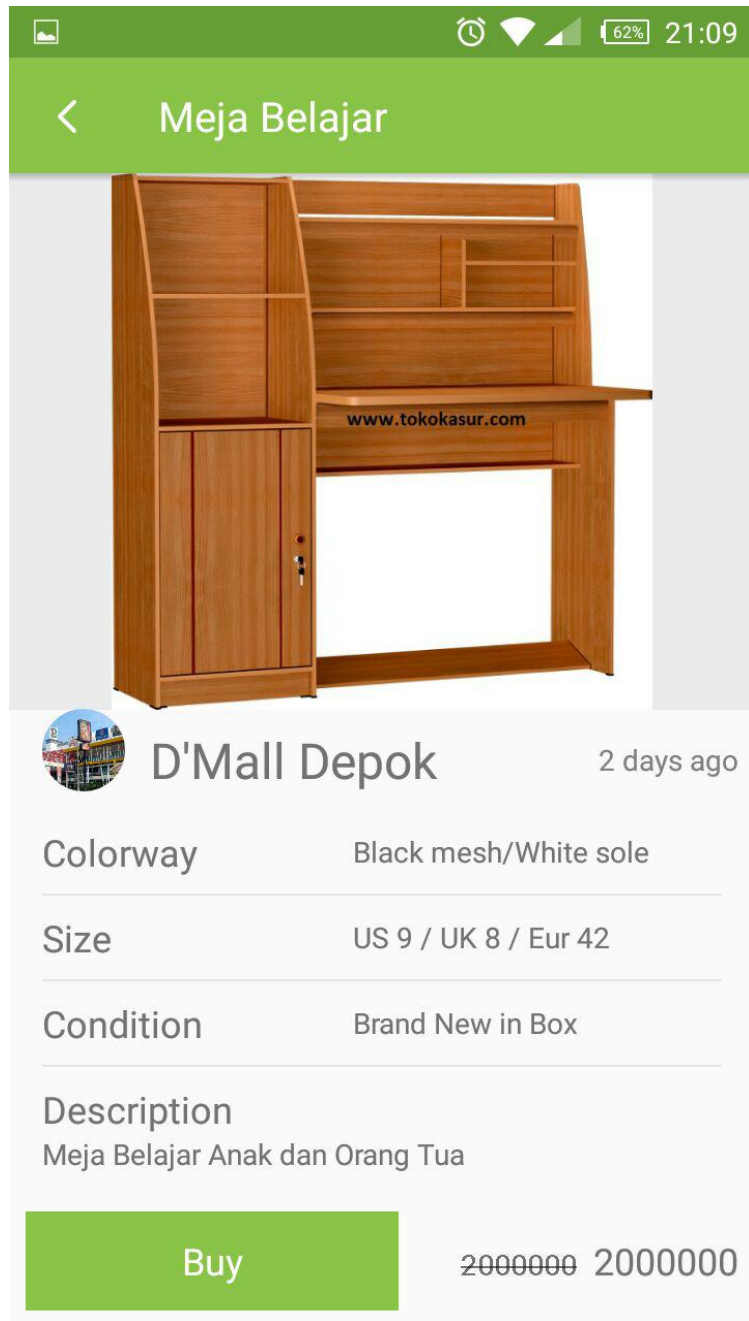
List Deal



The image shows a mobile application interface with a green header bar. The header contains a status bar at the top with icons for signal, Wi-Fi, battery (62%), and time (21:08). Below the status bar, the header has two tabs: 'DEAL' and 'PRODUCT'. The main content area displays four product cards in a 2x2 grid. Each card features an image of a product, its name, a price reduction, and a discount banner. The products are: a wooden study desk, a wooden wardrobe, a rack of plaid shirts, and a collection of jeans. The discount banners for the first two products indicate a 42-day and 41-day discount respectively.

Product	Original Price	Discounted Price	Discount Period
Meja Belajar	2000000	1400000	Diskon berakhir -42 hari lagi
Lemari Pakaian	3000000	2700000	Diskon berakhir -41 hari lagi
Plaid Shirts	-	-	-
Jeans	-	-	-





Detail Screen



Try it!
s.id/dCs



Preparing Database

Store

- Id unsigned int not null auto_increment
- Name varchar(50) not null
- Lat double not null
- Lng double not null
- Photo text not null
- Telp varchar(20) null
- Description text null
- Open_hour varchar(20) not null
- Address text null
- Created_at date not null

product

- Id unsigned int not null auto_increment
- Store_id unsigned int not null
- Name varchar(50) not null
- Price unsigned int not null
- Description text not null
- Photo text not null

deal

- Id unsigned int not null auto_increment
- Product_id unsigned int not null
- Start_date date not null
- End_date date not null
- Discount unsigned int not null



Task #1

Create 5 sample store. 3 store around you and 2 far from your location



JavaScript Object Notation (JSON)

- JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and client application, as an alternative to XML.
- The two primary parts that make up JSON are keys and values. Together they make a key/value pair.
- There are two types json component: object and arrays



JSON Object

```
{  
  "age" : "24",  
  "hometown" : "Missoula, MT",  
  "gender" : "male"  
}
```



JSON Arrays

```
[{  
  "name" : "Jason",  
  "age" : "24",  
  "gender" : "male"  
},  
{  
  "name" : "Kyle",  
  "age" : "21",  
  "gender" : "male"  
}]
```



Task #2

Create new file `get_store.php`
and return all stores in json format



Task #2 Answer
<https://pastebin.com/jkneXLmz>



Create New Project

💬 App Name: **NearDeal**

💬 Min SDK: **15**

💬 Package: **id.co.rumahcoding.neardeal**



Task #4
Create Navigation Drawer Activity. Name it
MainActivity



Retrofit

- **Retrofit** performs HTTP requests against an API running on a server somewhere in the Internet. Executing those requests from an Android application requires the Internet permission to open network sockets.
- compile 'com.squareup.retrofit2:retrofit:2.3.0'



Task #2.5

Add internet permission to
AndroidManifest.xml and add retrofit library to
build.gradle (module)

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```
compile 'com.squareup.retrofit2:retrofit:2.3.0'  
compile 'com.squareup.retrofit2:converter-gson:2.3.0'
```



Task #3

Create new class: ApiClient.java

```
public static Retrofit getClient() {  
    if (retrofit == null) {  
        retrofit = new Retrofit.Builder()  
            .baseUrl(BASE_URL)  
            .addConverterFactory(GsonConverterFactory.create())  
            .build();  
    }  
  
    return retrofit;  
}
```



Task #3 Answer
<https://pastebin.com/7r6ma4uM>



Task #3.5

Create new package: **responses**



Task #3.5

Create new class: **StoreResponse**.
Use jsonschema2pojo.org to generate it
from json



Task #3.6

Create new class: ApiEndPoint.java

```
public interface ApiEndPoint {  
    @GET("get_store.php")  
    Call<StoreResponse> getStore();  
}
```

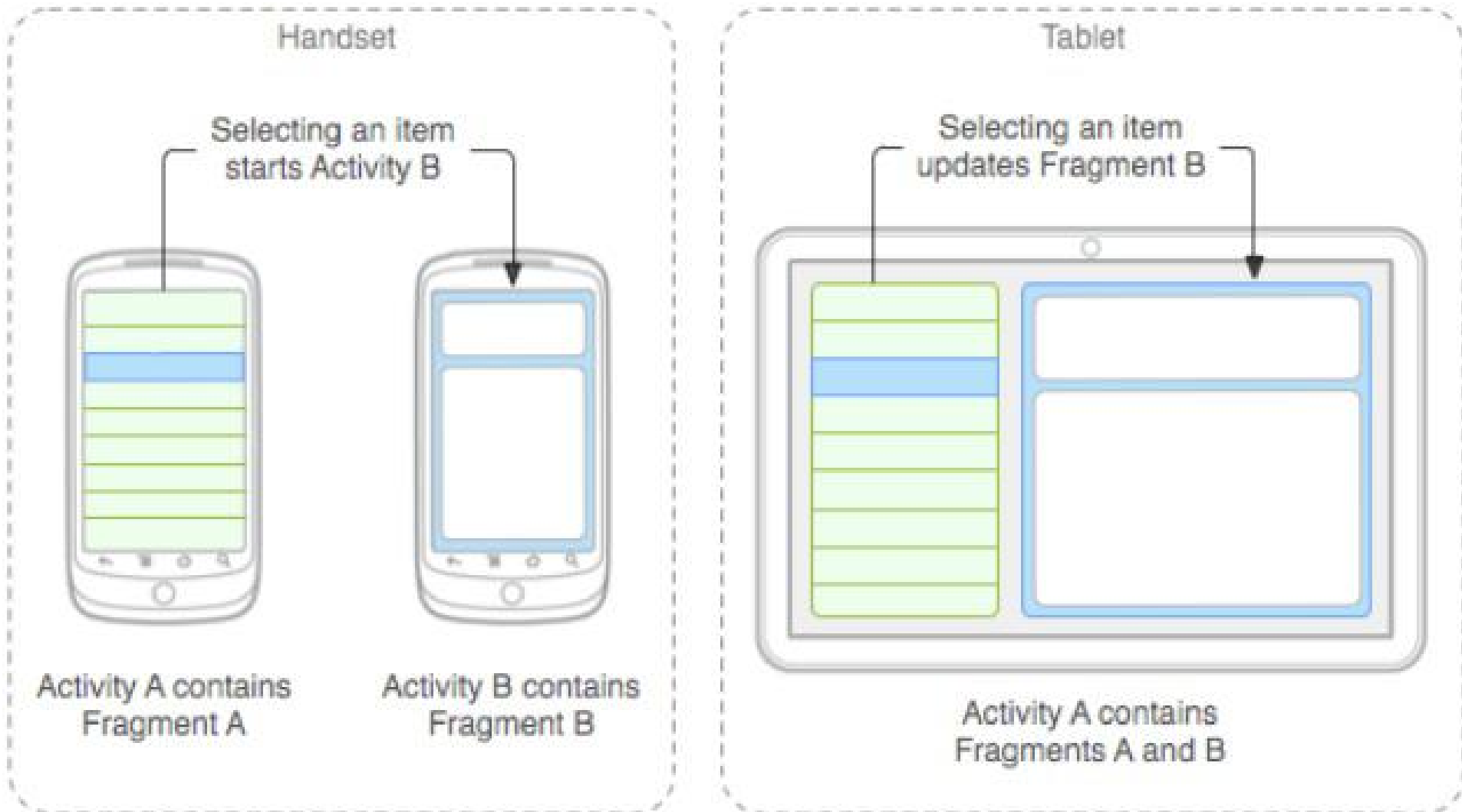


Fragment

- A **fragment** is usually used as part of an activity's user interface and contributes its own layout to the activity.
- To provide a layout for a **fragment**, you must implement the `onCreateView()` callback method, which the **Android** system calls when it's time for the **fragment** to draw its layout.
- For simplicity we can call fragment as sub activity



Fragment



Task #5

Create new package **fragments**



Task #6

Create new fragment: StoreListFragment



Task #6 Answer
<https://pastebin.com/HST6VVAX>



Task #7

Attach StoreListFragment to Main Activity



Task #7 Answer

Open **MainActivity** and append this code on **onCreate** method:

<https://pastebin.com/q8Ufi4YB>



Task #9 Answer
<https://pastebin.com/T0vWxCvp>



Task #10
Save your time, grab some utilities
class: **<https://s.id/dBm>**



Task #11

Create new method: **loadStores()** on
StoreListFragment

This method should get all stores from server



Task #11 Answer
<https://pastebin.com/xEFfGLmi>



WE DID IT!



Recycle View

- **RecyclerView** is flexible and efficient version of ListView. It is an container for rendering larger data set of views that can be recycled and scrolled very efficiently.
- **RecyclerView** is like traditional ListView widget, but with more flexibility to customizes and optimized to work with larger datasets.



Task #12

Create new layout file: **item_store.xml**



Ace Hardware Depok
Belum Ada Deal!



Card View

- **CardView** extends the `FrameLayout` class and lets you show information inside cards that have a consistent look across the platform. CardView widgets can have shadows and rounded corners.
- **compile 'com.android.support:cardview-v7:21.0.+'**



Task #13

Create new package: **adapters**



Task #14

Create new adapter class: **StoreItemAdapter**



Task #14 Answer
<https://pastebin.com/dU2zfKy5>



Task #15

Modify **StoreListFragment** and display all stores in **RecyclerView**



Task #15 Answer
<https://pastebin.com/XHaxM7WQ>

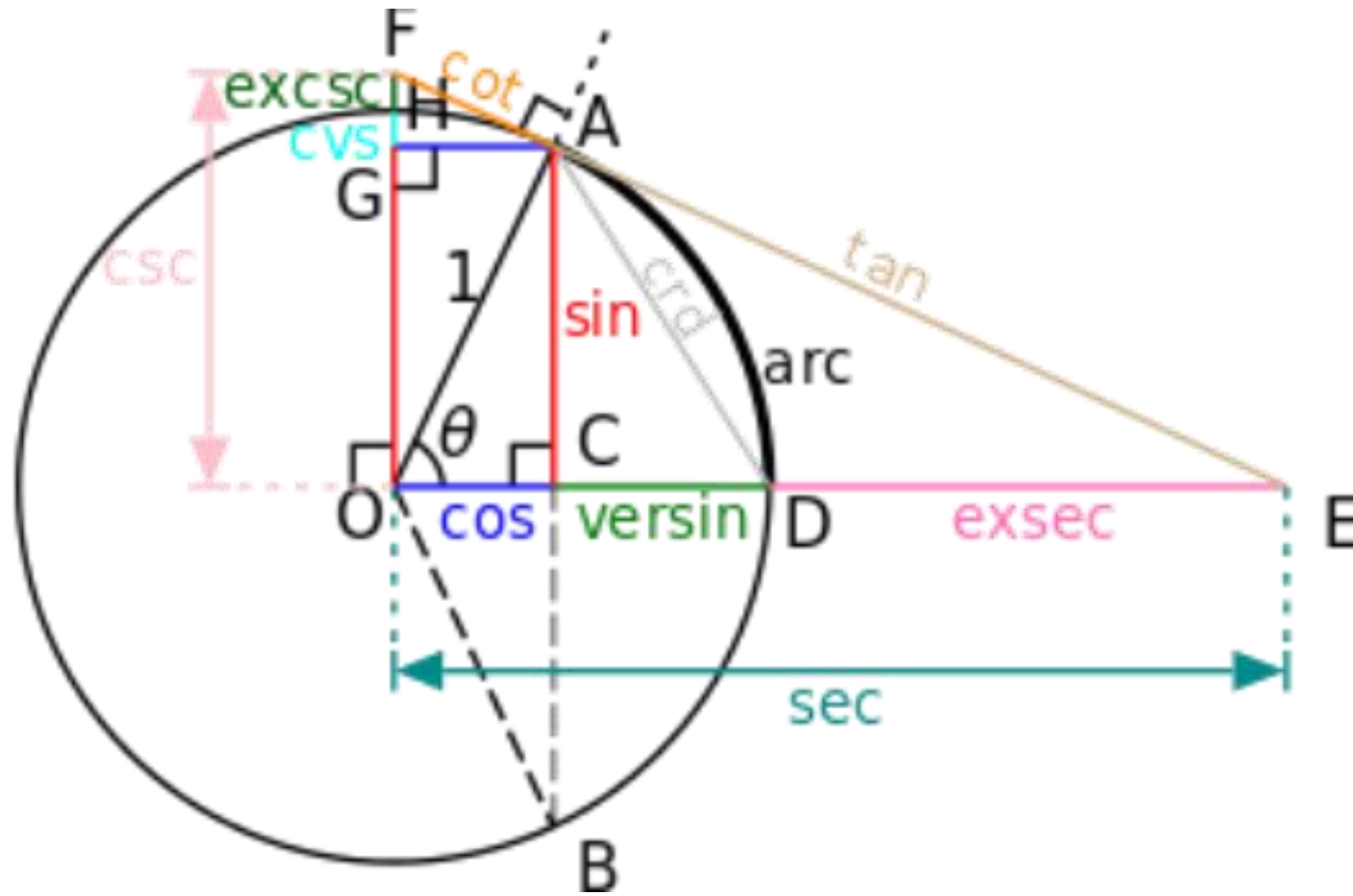


Thank You!

WE DID IT AGAIN



Haversine Formula



MySQL Great Circle Distance

- Find distance between two geo point (lat1,lng1) and (lat2,lng2)
- Here is the formula:

$$(6371 * \text{acos}(\cos(\text{radians}(\text{lat1})) * \cos(\text{radians}(\text{lat2})) \\ * \cos(\text{radians}(\text{lng2}) - \text{radians}(\text{lng1})) + \sin(\text{radians}(\text{lat1})) * \\ \sin(\text{radians}(\text{lat2}))))$$


Task #16

Modify get_store.php: Displaying only store near you



Task #16 Answer
<https://pastebin.com/R8Z7BAgt>



Finding Your Location

- Add permission **ACCESS_FINE_LOCATION**
- Check permission: `ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)`
- Request permission:
`ActivityCompat.requestPermissions(this,
 new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
 1);`
- `LocationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, this);`



Task #16

Modify MainActivity: request device's location



Task #16 Answer
<https://pastebin.com/ytpRGUXi>



Task #17

Sending argument (lat,lng) from **MainActivity** to
StoreListFragment



Task #17 Answer
<https://pastebin.com/zYdie3BX>



Task #18

Edit **StoreListFragment**: Catch location data sent from **MainActivity**



Task #18 Answer
<https://pastebin.com/xEynzK9P>



Task #19

Edit **ApiEndPoint**, add param **lat** and **lng**

```
public interface ApiEndPoint {  
    @GET("get_store.php")  
    Call<StoreResponse> getStore(@Query("lat") String lat,  
                                @Query("lng") String lng);  
}
```



Task #20

Edit **StoreListFragment**: Display only stores near you



Picasso

- A powerful image downloading and caching library for Android
- compile 'com.squareup.picasso:picasso:2.5.2'
- Picasso.with(context).load("http://i.imgur.com/DvpvklR.png").into(imageView);



Task #21

Edit StoreItemAdapter: Display store image

