

Rumah Coding Course

"Android Intermediate"

Studi Kasus Aplikasi "NearDeal"



Day 2

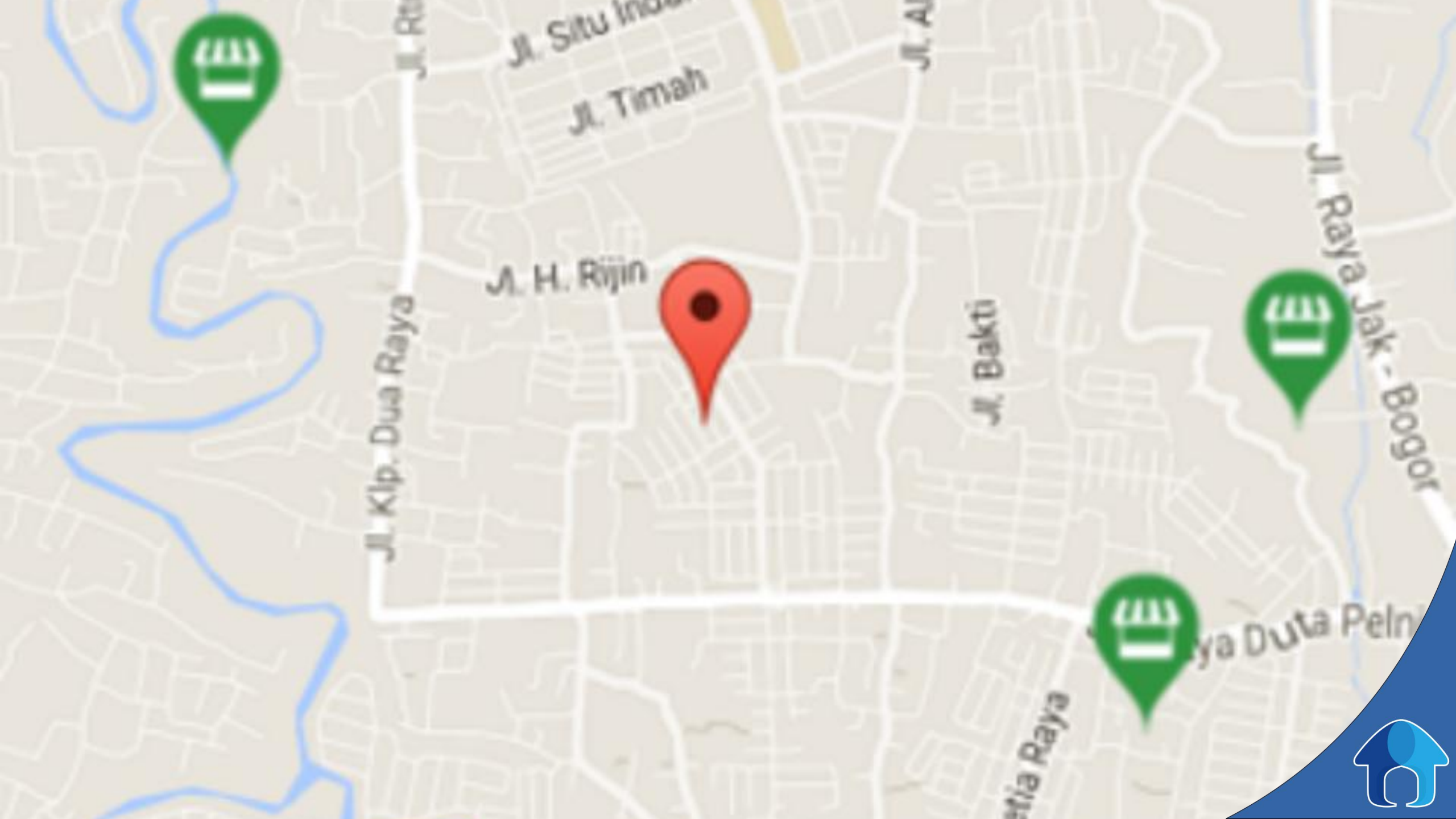


Course Syllabus

Day 2

- Google Maps
- Tab Layout
- Draw store locations on Maps





Add Google Play Service

<input checked="" type="checkbox"/>	Google Play Billing Library, rev 5	5.0.0	Installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input checked="" type="checkbox"/>	Google Play services	46	Installed
<input type="checkbox"/>	Google Web Driver	2	Not installed



Certificate Fingerprint

- Debug Keystore:
 - OSX & Linux: ~/.android/
 - Windows XP: C:\Documents and Settings\.android\
 - Windows 7: C:\Users\<user>\.android\
- Production Keystore: Create your production keystore
- `keytool -list -v -keystore keystore.name`



Getting API Key

- Goto <https://console.developers.google.com>
- Create or select a project.
- Click **Continue** to enable the Google Maps Android API.
- On the **Credentials** page, get an **API key**.
- From the dialog displaying the API key, select **Restrict key** to set an Android restriction on the API key.
- In the **Restrictions** section, select **Android apps**, then enter your app's SHA-1 fingerprint and package name



Add the API to App

- Add following line to AndroidManifest.xml under application tag

```
<meta-data  
    android:name="com.google.android.gms.version"  
    android:value="@integer/google_play_services_version" />
```

```
<meta-data  
    android:name="com.google.android.geo.API_KEY"  
    android:value="@string/google_maps_key" />
```

- compile 'com.google.android.gms:play-services-maps:11.4.2'



Google Map V2

- Add marker on Google Map

```
LatLng monasLatLng = new LatLng(-6.175642, 106.827521);  
mMap.addMarker(new MarkerOptions().position(monasLatLng).title("Monas"));
```

- Move and zoom Camera

```
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(myLatLng, 15f));
```

- Add custom marker on Google Map

```
mMap.addMarker(new MarkerOptions().position(myLatLng).title(name)  
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.store)));
```



Google Map V2

- Use this layout

```
<fragment  
    android:id="@+id/map"  
    android:name="com.google.android.gms.maps.SupportMapFragment"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```



Task #1

Create MapsActivity: Draw marker for **monas** on the map

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    LatLng mylocation = new LatLng(-6.1725799, 106.8210477);
    mMap.addMarker(new MarkerOptions().position(mylocation).title("Monas"));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(mylocation, 15f));
}
```



Task #2

Create menu action bar on **StoreListFragment** and send device location to **MapsActivity**.

```
Intent intent = new Intent(getActivity(), MapsActivity.class);  
intent.putExtra(KEY_LAT, mLat);  
intent.putExtra(KEY_LNG, mLng);  
startActivity(intent);
```



Task #3

Edit MapsActivity: Draw your location marker on the maps

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    LatLng mylocation = new LatLng(mLat,mLng);
    mMap.addMarker(new MarkerOptions().position(mylocation).title("MyLocation"));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(mylocation, 15f));
}
```



Task #4

Modify StoreListFragment: Send store data to MapsActivity

```
if (id == R.id.action_open_map){  
    Gson gson = new Gson();  
    Type listOfObject = new TypeToken<List<Store>>(){}.getType();  
    String list = gson.toJson(mStoreList, listOfObject);  
  
    Intent intent = new Intent(getActivity(), MapsActivity.class);  
    intent.putExtra(KEY_LAT, mLat);  
    intent.putExtra(KEY_LNG, mLng);  
    intent.putExtra(MapsActivity.STORES_KEY, list);  
    startActivity(intent);  
}
```



Download this icon
<http://s.id/dE0>



Task #5

Modify MapsActivity: Create new method
drawStores().

Draw marker for all stores on the map



Task #5 Answer
<https://pastebin.com/zneB68sr>



YAY

WE DID IT

memes.com



Task #6

Use PHPMYAdmin to create 4 products for every stores.



Task #7

Create new file: **get_product.php**: Return all products from specified store, use parameter `store_id`



Task #7 Answer
<https://s.id/nd2s1>



Task #8

Create new class: **ProductResponse**

(use www.jsonschema2pojo.org to generate class from json)



Task #8 Answer

s.id/bWU



Task #8.5

Modify ApiEndPoint.java: Add new **getProduct** end point

```
@GET("get_product.php")  
Call<ProductResponse> getProduct(@Query("store_id") String storeId);
```



Task 9

Create new layout: **item_product.xml**



Gramedia Depok
Rp 12345678



Task #9 Answer

s.id/ca8



Task #10

Create new Adapter: **ProductItemAdapter**



Task #10 Answer

s.id/cGf



Task #11

Create new fragment: ProductListFragment (Blank Fragment)



Task #12

Create new tabbed activity: **ProductActivity**

(Use navigation style Action Bar Tabs (with View Pager))

Just change your AndroidManifest to run test this activity



Task #13

Modify ProductActivity to contain 2 tabs: **Product** and **Deal**.

```
@Override
public int getCount() {
    // Show 1 total pages.
    return 2;
}

@Override
public CharSequence getPageTitle(int position) {
    switch (position) {
        case 0:
            return "Product";
        case 1:
            return "Deal";
    }
    return null;
}
```



Task #14

Attach ProductListFragment to ProductActivity on tab 1

(You will use SectionPagerAdapter to attach a fragment into tabbed layout instead of fragment manager like we did before)

```
@Override
public Fragment getItem(int position) {
    switch (position){
        case 0:
            return mProductListFragment;
        case 1:
            return null;
    }

    return null;
}
```



Task #15

Modify `fragment_product_list.xml`: Add RecyclerView

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="id.co.rumahcoding.neardeal.fragments.StoreListFragment">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</FrameLayout>
```



Task #16

Modify item_store.xml: Give your CardView an id

(We will access it from **StoreItemAdapter**)

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <android.support.v7.widget.CardView
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="0dp"
        app:cardElevation="10dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="1dp"
        android:layout_marginBottom="1dp">
```



Task #17

Modify **StoreItemAdapter**: Get CardView from layout

```
public class ViewHolder extends RecyclerView.ViewHolder {  
    public CardView container;  
    public ImageView imageView;  
    public TextView nameTextView;  
    public TextView dealTextView;  
  
    public ViewHolder(View itemView){  
        super(itemView);  
        container = itemView.findViewById(R.id.container);  
        imageView = itemView.findViewById(R.id.imageView);  
        nameTextView = itemView.findViewById(R.id.tv_name);  
        dealTextView = itemView.findViewById(R.id.tv_deal);  
    }  
}
```



Task #17.5

Modify StoreItemAdapter: Add this codes (We will make the store clickable)

```
public interface OnItemClickListener {  
    public void onItemClick(String storeId);  
}  
  
public void setListener(OnItemClickListener listener) {  
    mListener = listener;  
}  
  
private OnItemClickListener mListener;
```



Task #17.8

Modify StoreItemAdapter: Now we listen for click on the **CardView**

```
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    final Store store = mStoreList.get(position);

    holder.nameTextView.setText(store.getName());
    holder.dealTextView.setText("Belum Ada Deal!");

    Picasso.with(mContext).load(store.getPhoto()).into(holder.imageView);

    if (mListener != null){
        holder.container.setOnClickListener((view) -> {
            mListener.onItemClick(store.getId());
        });
    }
}
```



Take a break,
take a breath!



Task #18

Modify **StoreListFragment**: implements
StoreItemAdapter.OnItemClickListener

```
public class StoreListFragment extends Fragment implements  
    StoreItemAdapter.OnItemClickListener {
```



Task #19

Modify **StoreListFragment**: set adapter listener adapter to **this**.

```
mStoreList = new ArrayList<>();  
mAdapter = new StoreItemAdapter(getActivity(), mStoreList);  
mAdapter.setListener(this);  
recyclerView.setAdapter(mAdapter);
```



Task #19.5

Modify **StoreListFragment**: implements **onItemClick** method and send **storeId** to **ProductActivity**.

```
@Override
public void onItemClick(String storeId) {
    Intent intent = new Intent(getActivity(), ProductActivity.class);
    intent.putExtra(ProductActivity.KEY_STORE_ID, storeId);
    startActivity(intent);
}
```



Task #20

Modify **ProductActivity**: Send **storeId** to **ProductListFragment**.

```
public class SectionsPagerAdapter extends FragmentPagerAdapter {
    private ProductListFragment mProductListFragment;

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);

        String storeId = getIntent().getStringExtra(KEY_STORE_ID);
        Bundle argument = new Bundle();
        argument.putString(ProductListFragment.KEY_STORE_ID, storeId);

        mProductListFragment = new ProductListFragment();
        mProductListFragment.setArguments(argument);
    }
}
```



Task #21

Modify **ProductListFragment**: Use **GridLayoutManager** with **spanCount 2** for **recycler view**

```
RecyclerView recyclerView = view.findViewById(R.id.recycler_view);  
GridLayoutManager layoutManager = new GridLayoutManager(getActivity(), spanCount: 2);  
recyclerView.setLayoutManager(layoutManager);
```



Task #22

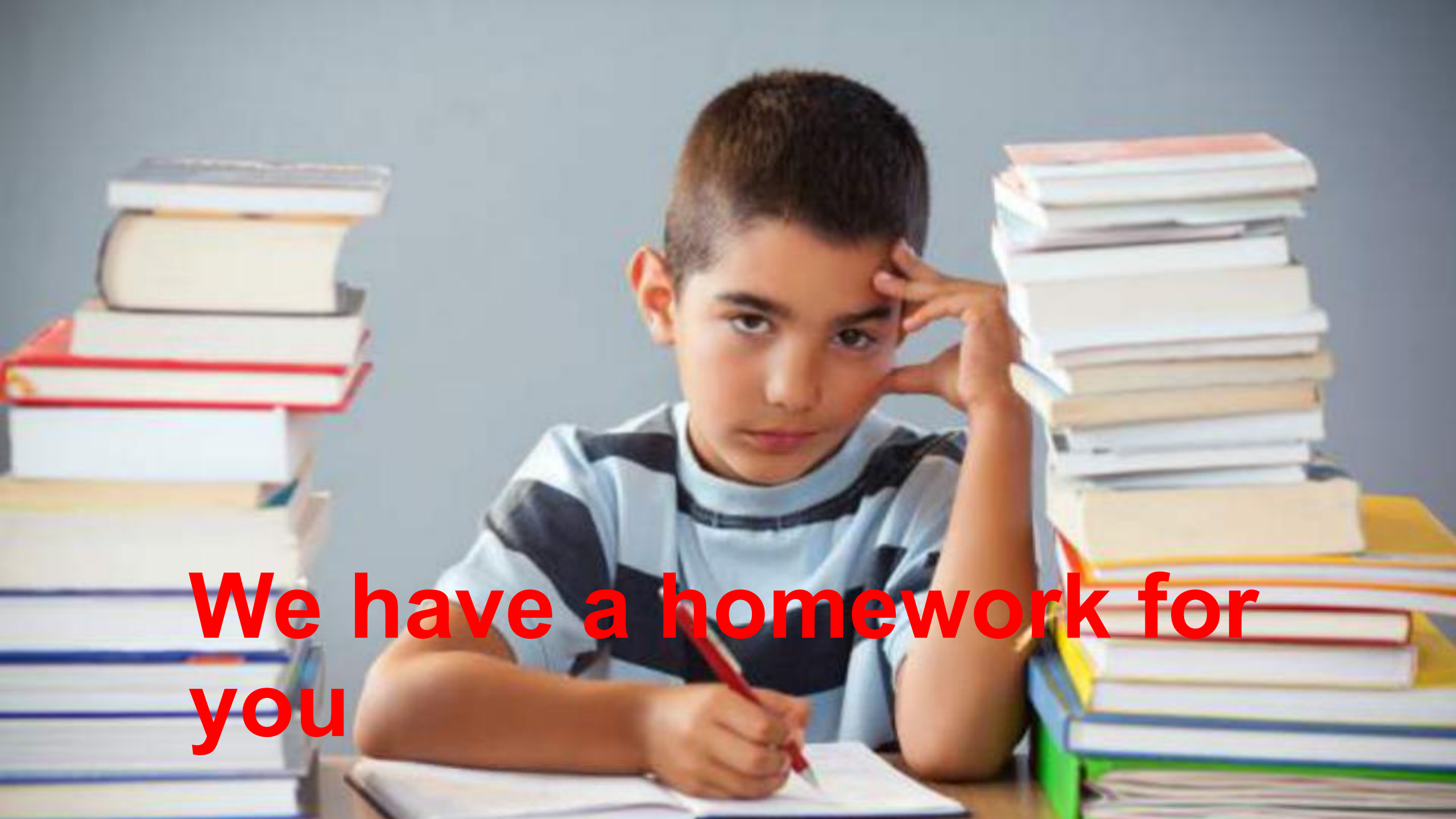
Modify **ProductListFragment**: Create new method `loadProducts()`



Task #22 Answer

s.id/dqd





**We have a homework for
you**

Homework

- Showing deal list on tab 2
- Discount price counted from field discount on table **deal**
- If today less than **start_date** ==> **Diskon dimulai xx hari lagi**
- If today between **start_date** and **end_date** ==> **Diskon berakhir xx hari lagi**
- If today greater than **end_date** ==> **Diskon sudah selesai**
- Follow step on the next slide



Meja Belajar

~~2000000~~ 1400000

Diskon
berakhir -42
hari lagi



Lemari Pakaian

~~3000000~~ 2700000

Diskon
berakhir -41
hari lagi



Baju Kemeja

~~250000~~ 137500

Diskon
berakhir -40



Celana Jeans

~~300000~~ 150000

Diskon
berakhir -35



Homework

- Generate 4 deal for each store on table **deal**
- Create PHP script: **get_deal.php**
- Create xml: **item_deal.xml**
- Create adapter: **DealItemAdapter.java**
- Create fragment: **DealListFragment.java**
- Attach fragment **DealListFragment** to **ProductActivity** on tab 2

