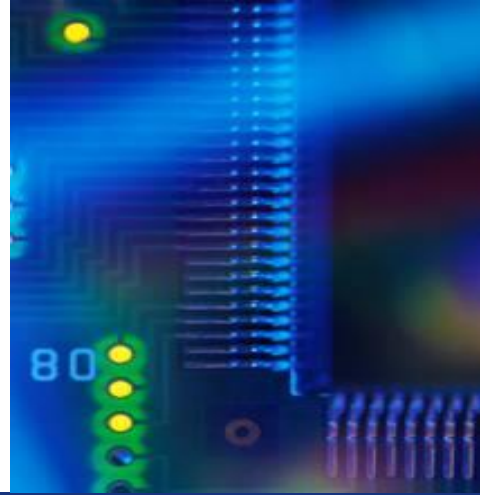
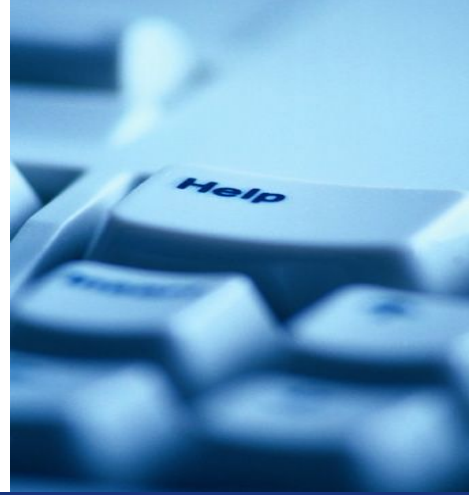




**UNIVERSITAS
BUDI LUHUR**



FAKULTAS TEKNOLOGI INFORMASI

Mobile Programming

[PG119/ 3 SKS]

Riskiana Wulan, MKom



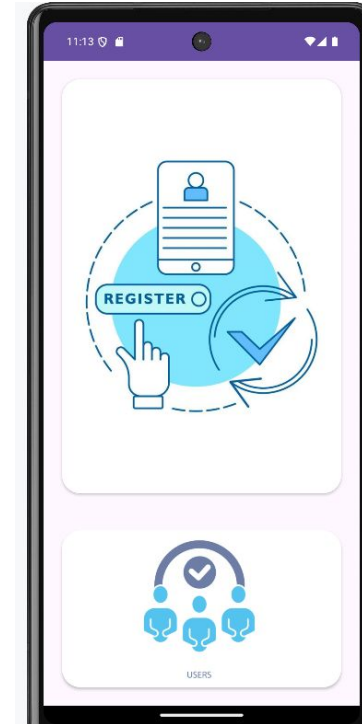
CARDS

CARDVIEW

- A **Card** is a container that can hold UI elements like text, images, and buttons in a structured format.
- Typically used to display items in a list or grid format.
- Customizable with Material Design principles.
- Supports shadows, elevation, and rounded corners.
- Gradle Dependency

```
implementation "androidx.cardview:cardview:1.0.0"
```

<https://developer.android.com/develop/ui/views/layout/cardview>



Card Layout with Text

```
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    app:cardCornerRadius="4dp"
    app:cardElevation="2dp">

    <TextView
        android:id="@+id/cardText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Card Content" />
</androidx.cardview.widget.CardView>
```

Card Layout with images

```
<androidx.cardview.widget.CardView
    android:id="@+id/cardview_users"
    android:layout_width="wrap_content"
    android:layout_margin="24dp"
    android:layout_height="wrap_content"
    app:cardCornerRadius="24dp"
    android:layout_weight="1">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/users"/>
</androidx.cardview.widget.CardView>
```

Activity

```
CardView cardViewUsers = findViewById(R.id.cardview_users);  
cardViewUsers.setOnClickListener(v -> {  
    Intent intent = new Intent( packageContext: HomeActivity.this, ViewUsersActivity.class);  
    startActivity(intent);  
});
```



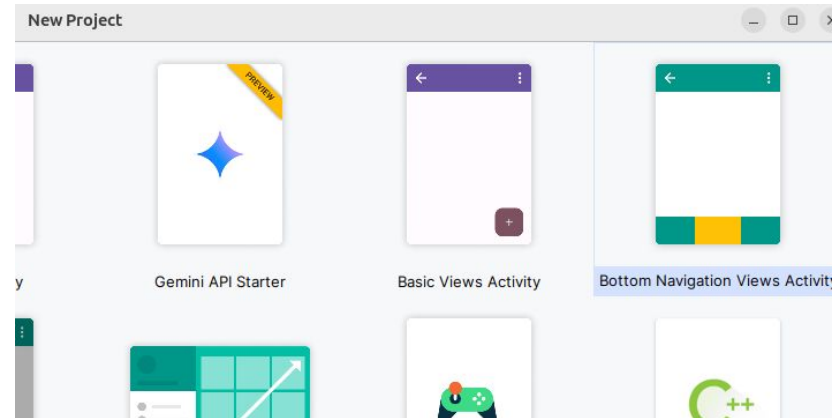
FRAGMENT

- A [Fragment](#) represents a reusable portion of your app's UI.
- A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events.
- Fragments can't live on their own. They must be *hosted* by an activity or another fragment. The fragment's view hierarchy becomes part of, or *attaches to*, the host's view hierarchy.

<https://developer.android.com/guide/fragments>

Sample

Pilih new Project -> Bottom Navigation Views Activity

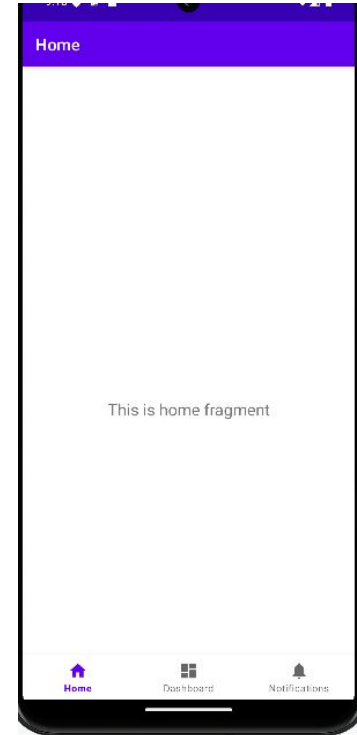


Main Component

- **MainActivity:** Hosts the Bottom Navigation View.
- **Fragments:**
 - HomeFragment
 - DashboardFragment
 - NotificationsFragment
- **Menu Resource File:**
 - res/menu/bottom_nav_menu.xml contains navigation items.

Layout Files:

- activity_main.xml: Contains the Bottom Navigation View.
- Fragment layouts for each tab.



Main Activity

- **Key Functionality:**
 - Initialize Bottom Navigation View.
 - Set up navigation

```

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding; 3 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of IDs because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
            R.id.navigation_home, R.id.navigation_dashboard, R.id.navigation_notifications)
            .build();
        NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment_activity_mair
        NavigationUI.setupActionBarWithNavController( activity: this, navController, appBarConfiguration);
        NavigationUI.setupWithNavController(binding.navView, navController);
    }
}
  
```

bottom_nav_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="Home" />

    <item
        android:id="@+id/navigation_dashboard"
        android:icon="@drawable/ic_dashboard_black_24dp"
        android:title="Dashboard" />

    <item
        android:id="@+id/navigation_notifications"
        android:icon="@drawable/ic_notifications_black_24dp"
        android:title="Notifications" />

</menu>
```

View Model

- **Definition:** A ViewModel is part of Android Architecture Components and is designed to store and manage UI-related data in a lifecycle-conscious way.
- **Benefits:**
 - Retains data during configuration changes (e.g., screen rotation).
 - Decouples UI logic from business logic.
 - Shares data between Fragments.

```
public class HomeViewModel extends ViewModel { 2 usages

    private final MutableLiveData<String> mText; 3 usages

    public HomeViewModel() { no usages
        mText = new MutableLiveData<>();
        mText.setValue("This is home fragment");
    }

    > public LiveData<String> getText() { return mText; }
}
```

Fragment

```
public class HomeFragment extends Fragment {

    private FragmentHomeBinding binding; 4 usages

    public View onCreateView(@NonNull LayoutInflater inflater,
                           ViewGroup container, Bundle savedInstanceState) {
        HomeViewModel homeViewModel =
            new ViewModelProvider( owner: this).get(HomeViewModel.class);

        binding = FragmentHomeBinding.inflate(inflater, container, attachToParent: false);
        View root = binding.getRoot();

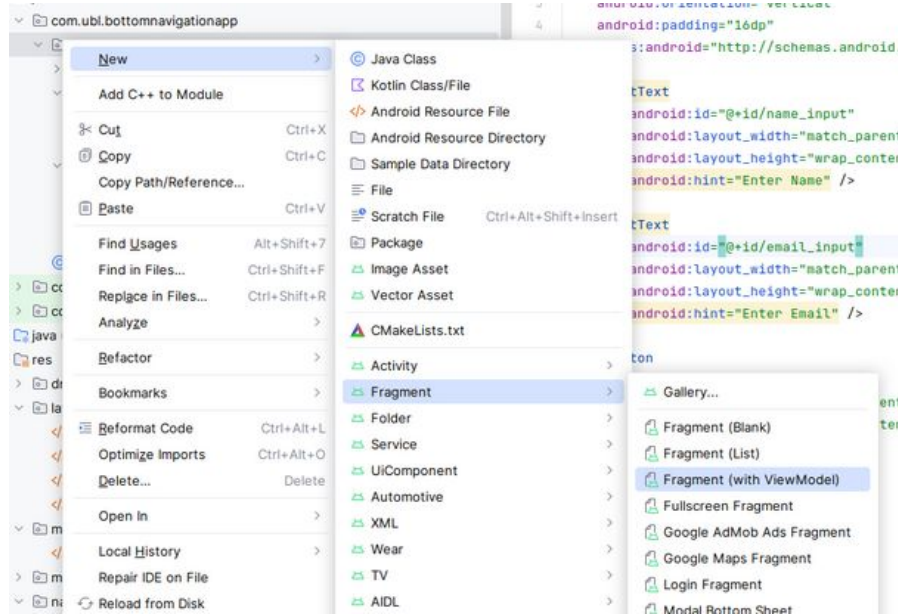
        final TextView textView = binding.textHome;
        homeViewModel.getText().observe(getViewLifecycleOwner(), textView::setText);
        return root;
    }
}
```



Sample Create User Fragment

Create new fragment

New->Fragment->Fragment with View Model



Akan
menggenerate
fragment class,
view model
class dan layout
file

ViewModel

```
public class NotificationsViewModel extends ViewModel { 2 usages

    private final MutableLiveData<User> userLiveData = new MutableLiveData<>(); 2 usages

    public void setUser(User user) { 1 usage
        userLiveData.setValue(user);
    }

    public LiveData<User> getUser() { no usages
        return userLiveData;
    }
}
```

Fragment Class

```
public class NotificationsFragment extends Fragment {
    private NotificationsViewModel viewModel; 2 usages

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

        viewModel = new ViewModelProvider( owner: this).get(NotificationsViewModel.class);

        EditText nameInput = view.findViewById(R.id.name_input);
        EditText emailInput = view.findViewById(R.id.email_input);
        Button saveButton = view.findViewById(R.id.save_button);

        saveButton.setOnClickListener(v -> {
            String username = nameInput.getText().toString();
            String email = emailInput.getText().toString();

            User user = new User(username, email);
            viewModel.setUser(user);
        });

        return view;
    }
}
```

Fragment layout

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <EditText
        android:id="@+id/name_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Name" />

    <EditText
        android:id="@+id/email_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Email" />

    <Button
        android:id="@+id/save_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save" />

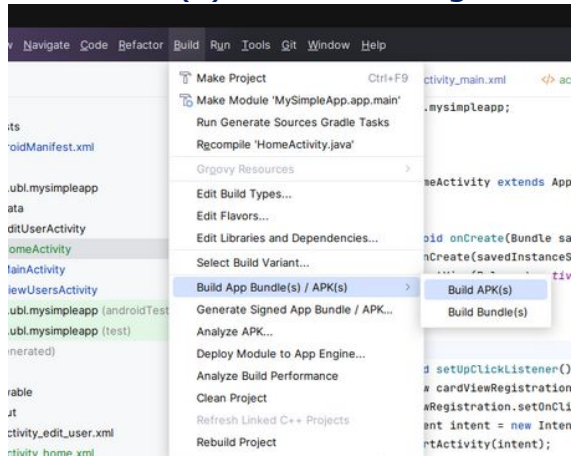
</LinearLayout>
```



Generate APK

Pada Android Studio

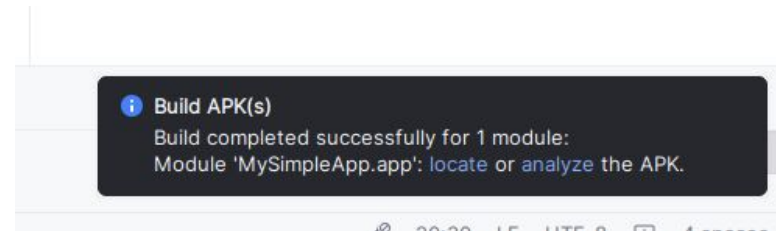
Pilih Build -> Build App Bundle(s)/APK(s) ->
Build APK(s) untuk testing



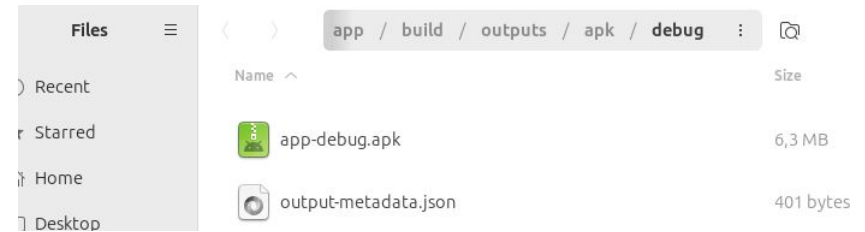
Pilih Build Bundles(s) jika ingin mengupload
android app ke Google Play

<https://developer.android.com/studio/publish/app-signing#sign-apk>

Lalu pilih locate pada popup build yang muncul
pada kanan bawah



Akan muncul lokasi apk yg terbentuk



File .apk dapat langsung diinstall pada mobile
android