Daftar Isi

[1. Library 2](#_Toc95685784)

[2. Handle Loading 2](#_Toc95685785)

[3. NavigationBottom 2](#_Toc95685786)

[4. Passing Data / Send data to Another Activity 4](#_Toc95685787)

[5. HashMap 5](#_Toc95685788)

[6. Ganti Background Tint Image 5](#_Toc95685789)

[7. Session 5](#_Toc95685790)

[8. Get Device Id 6](#_Toc95685791)

[9. Get Date Now 6](#_Toc95685792)

[10. String Random 6](#_Toc95685793)

[11. Chat Whatsapp 6](#_Toc95685794)

[12. Alert Dialog 6](#_Toc95685795)

[13. Image Slider 7](#_Toc95685796)

[14. Spinner 9](#_Toc95685797)

[15. Circle Image View 9](#_Toc95685798)

[16. Retrofit 10](#_Toc95685799)

[17. Retrofit POST 11](#_Toc95685800)

[18. Retrofit GET dengan Recycleview 12](#_Toc95685801)

[19. Retrofit Recycleview Filter 16](#_Toc95685802)

[20. Retrofit dengan Authorization 18](#_Toc95685803)

[21. Pilih gambar dari gallery 18](#_Toc95685804)

[22. Upload Foto FirebaseStorage 19](#_Toc95685805)

[23. Chat Message dengan Firebase 20](#_Toc95685806)

Android

# Library

1. Anko

ext.anko\_version='0.10.8'

implementation "org.jetbrains.anko:anko:$anko\_version"

1. Picasso

implementation 'com.squareup.picasso:picasso:2.71828'

# Handle Loading

lateinit var handler: Handler

handler = Handler()  
handler.postDelayed(**{** startActivity(intentFor<LoginActivity>().clearTask().newTask())  
 finish()  
**}**, 3000)

# NavigationBottom

1. Class

class HomeAdminActivity : AppCompatActivity() {  
  
 private val onNavigationItemSelectedListener =  
 BottomNavigationView.OnNavigationItemSelectedListener **{** item **->** when (item.*itemId*) {  
 R.id.*navigation\_home* -> {  
 *supportFragmentManager*.beginTransaction().replace(  
 R.id.*framehome*,  
 HomeAdminFragment()  
 ).commit()  
 return@OnNavigationItemSelectedListener true  
 }  
 R.id.*navigation\_report* -> {  
 *supportFragmentManager*.beginTransaction().replace(  
 R.id.*framehome*,  
 ReportFragment()  
 ).commit()  
 return@OnNavigationItemSelectedListener true  
  
 }  
 R.id.*navigation\_produk* -> {  
 *supportFragmentManager*.beginTransaction().replace(  
 R.id.*framehome*,  
 ProdukFragment()  
 ).commit()  
 return@OnNavigationItemSelectedListener true  
  
 }  
  
 R.id.*navigation\_profil* -> {  
 *supportFragmentManager*.beginTransaction().replace(  
 R.id.*framehome*,  
 ProfilFragment()  
 ).commit()  
 return@OnNavigationItemSelectedListener true  
  
 }  
  
 }  
  
 false  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_home\_admin*)  
 val navView: BottomNavigationView = findViewById(R.id.*nav\_viewhome*)  
  
 navView.setOnNavigationItemSelectedListener(onNavigationItemSelectedListener)  
  
 moveToFragment(HomeAdminFragment())  
  
 }  
  
 private fun moveToFragment(fragment: Fragment) {  
 val fragmentTrans = *supportFragmentManager*.beginTransaction()  
 fragmentTrans.replace(R.id.*framehome*, fragment)  
 fragmentTrans.commit()  
 }  
}

1. XML

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <FrameLayout android:layout\_width="match\_parent" android:layout\_height="match\_parent"  
 android:layout\_above="@+id/nav\_viewhome"  
 android:id="@+id/framehome"  
 >  
  
 </FrameLayout>  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/nav\_viewhome"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/white"  
 app:menu="@menu/bottom\_nav\_menu"  
 app:itemIconTint="@color/hijau"  
 android:layout\_alignParentBottom="true"  
 app:labelVisibilityMode="labeled"  
  
 />  
</RelativeLayout>

1. Menu

<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="@string/title\_home" />  
  
 <item  
 android:id="@+id/navigation\_produk"  
 android:icon="@drawable/ic\_dashboard\_black\_24dp"  
 android:title="@string/produk" />  
  
 <item  
 android:id="@+id/navigation\_report"  
 android:icon="@drawable/ic\_dashboard\_black\_24dp"  
 android:title="@string/report" />  
 <item  
 android:id="@+id/navigation\_profil"  
 android:icon="@drawable/ic\_dashboard\_black\_24dp"  
 android:title="@string/profil" />  
</menu>

# Passing Data / Send data to Another Activity

1. Class Awal

*startActivity*<DataPembeliActivity>("hargatotal" *to* hargatotal)

1. Class Tujuan

val bundle: Bundle? = *intent*.*extras*hargatotal = bundle!!.getInt("hargatotal")

# HashMap

val usermap: MutableMap<String, Any?> = HashMap()  
usermap["uid"] = uid  
usermap["nama"] = nama

# Ganti Background Tint Image

val acceptint = DrawableCompat.wrap(holder.accept.*background*)  
val rejecttint = DrawableCompat.wrap(holder.reject.*background*)  
DrawableCompat.setTint(acceptint,context.*resources*.getColor(R.color.*abumatang*))  
DrawableCompat.setTint(rejecttint,context.*resources*.getColor(R.color.*abumatang*))  
holder.accept.*background* = acceptint  
holder.reject.*background* = rejecttint

# Session

class SessionManager(private val context: Context?) {  
 val privateMode = 0  
 val privateName ="login"  
 var Pref : SharedPreferences?=context?.getSharedPreferences(privateName,privateMode)  
 var editor : SharedPreferences.Editor?=Pref?.edit()  
  
 private val islogin = "login"  
 fun setLogin(check: Boolean){  
 editor?.putBoolean(islogin,check)  
 editor?.commit()  
 }  
  
 fun getLogin():Boolean?  
 {  
 return Pref?.getBoolean(islogin,false)  
 }  
   
 private val isloginadmin = "loginadmin"  
 fun setLoginadmin(check: Boolean){  
 editor?.putBoolean(isloginadmin,check)  
 editor?.commit()  
 }  
  
 fun getLoginadmin():Boolean?  
 {  
 return Pref?.getBoolean(isloginadmin,false)  
 }  
}

# Get Device Id

1. Manifest

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />  
<uses-permission android:name="android.permission.READ\_PRIVILEGED\_PHONE\_STATE" />

1. Function

@SuppressLint("HardwareIds")  
fun getDeviceUniqueID(activity: Activity): String? {  
 return Secure.getString(  
 activity.*contentResolver*,  
 Secure.*ANDROID\_ID* )  
}

1. Penggunaan

getDeviceUniqueID(this).*toString*()

# Get Date Now

val sdf = SimpleDateFormat("dd/M/yyyy hh:mm:ss")  
val currentDate = sdf.format(Date())

# String Random

fun kodeorder(): String {  
 val charPool: List<Char> = ('A'..'Z') + ('0'..'9')   
 val outputStrLength = (5)  
 return (1..outputStrLength)  
 .*map* **{** kotlin.random.Random.nextInt(0, charPool.size) **}** .*map*(charPool::get)  
 .*joinToString*("")  
}

# Chat Whatsapp

binding.btnchat.setOnClickListener **{** val number = "6282138161919"  
 val url = "https://api.whatsapp.com/send?phone=$number"  
 val i = Intent(Intent.*ACTION\_VIEW*)  
 i.setPackage("com.whatsapp")  
 i.*data* = Uri.parse(url)  
 startActivity(i)  
**}**

# Alert Dialog

val builder = AlertDialog.Builder(this@DetailActivity)  
builder.setTitle("Data gaji")  
builder.setMessage("Hapus gaji bulan ini ? ")  
builder.setPositiveButton(android.R.string.*yes*) **{** dialog, which **->** *toast*("iya")  
**}**builder.setNegativeButton(android.R.string.*no*) **{** dialog, which **->** *toast*("tidak")  
**}**builder.show()

# Image Slider

1. Implementation

//Slider  
implementation 'com.github.smarteist:autoimageslider:1.4.0'

1. XML

<com.smarteist.autoimageslider.SliderView  
 android:id="@+id/imageSlider"  
 android:layout\_width="match\_parent"  
 android:layout\_height="153dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginEnd="24dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout2"  
 app:sliderAnimationDuration="600"  
 app:sliderAutoCycleDirection="back\_and\_forth"  
 app:sliderAutoCycleEnabled="true"  
 app:sliderIndicatorAnimationDuration="600"  
 app:sliderIndicatorGravity="center\_horizontal|bottom"  
 app:sliderIndicatorOrientation="horizontal"  
 app:sliderIndicatorSelectedColor="#5A5A5A"  
 app:sliderIndicatorUnselectedColor="#FFF"  
 app:sliderScrollTimeInSec="1"  
 app:sliderStartAutoCycle="true" />

1. Layout

<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 app:cardCornerRadius="9dp"  
 android:layout\_height="match\_parent">  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scaleType="fitCenter"  
 android:id="@+id/imageSlider"/>  
</com.google.android.material.card.MaterialCardView>

1. Adapter

package com.satelit.satelitpaint.adapter  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ImageView  
import com.satelit.satelitpaint.R  
import com.smarteist.autoimageslider.SliderViewAdapter  
import com.squareup.picasso.Picasso  
  
class ShowSliderAdapter() :  
 SliderViewAdapter<ShowSliderAdapter.VH>() {  
 private var mSliderItems = ArrayList<String>()  
 fun renewItems(sliderItems: ArrayList<String>) {  
 mSliderItems = sliderItems  
 notifyDataSetChanged()  
 }  
  
 fun addItem(sliderItem: String) {  
 mSliderItems.add(sliderItem)  
 notifyDataSetChanged()  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup): VH {  
 val inflate: View = LayoutInflater.from(parent.*context*).inflate(R.layout.*listshowslider*, null)  
 return VH(inflate)  
 }  
  
 override fun onBindViewHolder(viewHolder: VH, position: Int) {  
 //load image into view  
 Picasso.get().load(mSliderItems[position]).fit().centerCrop().into(viewHolder.imageView)  
 }  
  
 override fun getCount(): Int {  
 return mSliderItems.size  
 }  
  
 inner class VH(itemView: View) : ViewHolder(itemView) {  
 var imageView: ImageView = itemView.findViewById(R.id.*imageSlider*)  
  
 }  
}

1. Class

private fun setImageInSlider(images: ArrayList<String>, imageSlider: SliderView) {  
 val adapter = ShowSliderAdapter()  
 adapter.renewItems(images)  
 imageSlider.setSliderAdapter(adapter)  
 imageSlider.*isAutoCycle* = true  
 imageSlider.startAutoCycle()  
}

fun getslider(){  
 val imageList: ArrayList<String> = ArrayList()

imageList.add(hasil.foto0.*toString*())

imageList.add(hasil.foto1.*toString*())

imageList.add(hasil.foto2.*toString*())

setImageInSlider(imageList, binding.imageSlider)  
}

# Spinner

1. XML

<Spinner  
 android:id="@+id/*spnpenolongankelahiran* "  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginRight="20dp"  
 android:background="@drawable/kotakisi" />

1. Code

val datakelamin = *arrayOf*("Dokter","Bidan/Perawat","Dukun","Lainnya")  
val spinner = *find*<Spinner>(R.id.*spnpenolongankelahiran*)  
if (spinner!=null){  
 val adapter = ArrayAdapter(this,  
 android.R.layout.*simple\_spinner\_item*, datakelamin)  
 spinner.*adapter* = adapter  
  
 spinner.*onItemSelectedListener* = object :  
 AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(parent: AdapterView<\*>,  
 view: View, position: Int, id: Long) {  
 penolongankelahiran = datakelamin[position]  
 }  
  
 override fun onNothingSelected(parent: AdapterView<\*>) {  
 // write code to perform some action  
 }  
 }  
  
}

# Circle Image View

1. Implementasi terlebih dahulu

implementation 'de.hdodenhof:circleimageview:3.1.0'

1. Circle

<de.hdodenhof.circleimageview.CircleImageView

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/profile\_image"

android:layout\_width="96dp"

android:layout\_height="96dp"

android:src="@drawable/profile"

app:civ\_border\_width="2dp"

app:civ\_border\_color="#FF000000"/>

# Retrofit

1. Implementasi

//Retrofit  
implementation 'com.squareup.retrofit2:retrofit:2.6.2'  
implementation 'com.squareup.retrofit2:converter-gson:2.5.0'  
implementation 'com.squareup.retrofit2:converter-scalars:2.6.2'  
implementation 'com.squareup.retrofit2:adapter-rxjava2:2.4.0'  
implementation "com.squareup.okhttp3:okhttp:3.12.1"  
implementation 'com.squareup.okhttp3:logging-interceptor:3.11.0'  
implementation "com.squareup.okhttp3:okhttp-urlconnection:3.10.0"

1. Tambahkan permission di manifest

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

1. Tambahkan perintah berikut di bagian application manifest

android:usesCleartextTraffic="true"

1. Buat package webservice/ApiClient

class ApiClient {  
 companion object{  
 private var retrofit : Retrofit? = null  
 private var opt = OkHttpClient.Builder().*apply* **{** connectTimeout(30, TimeUnit.*SECONDS*)  
 readTimeout(30, TimeUnit.*SECONDS*)  
 writeTimeout(30,TimeUnit.*SECONDS*)  
 **}**.build()  
  
 private fun getClient() : Retrofit{  
 return if (retrofit ==null){  
 retrofit = Retrofit.Builder().*apply* **{** client(opt)  
 baseUrl("http://192.168.0.106:8000/api/")  
 addConverterFactory(GsonConverterFactory.create())  
 **}**.build()  
 retrofit!!  
 }else{  
 retrofit!!  
 }  
 }  
  
 fun instance() = getClient().create(ApiService::class.*java*)  
 }  
  
}

1. Buat class di webservice/ApiService

interface ApiService {  
   
}

# Retrofit POST

1. Buatlah method POST di webservice/ApiService

@FormUrlEncoded  
@POST("login")  
fun login(  
 @Field("email") email : String,  
 @Field("password") password: String  
): Call<PostResponse>

1. Penggunaan

api.login(email, password).enqueue(object : Callback<PostResponse>{  
 override fun onResponse(  
 call: Call<PostResponse>,  
 response: Response<PostResponse>  
 ) {  
 try {  
 if (response.body()!!.data == 1) {  
 sessionManager.setToken(response.body()!!.token)  
 sessionManager.setLogin(true)  
 progressDialog.dismiss()  
 *toast*("login berhasil")  
 *startActivity*<HomeActivity>()  
 finish()  
 } else {  
 progressDialog.dismiss()  
 Snackbar.make(view, "Email atau password salah", Snackbar.*LENGTH\_SHORT*).show()  
  
 }  
  
  
 }catch (e : Exception){  
 progressDialog.dismiss()  
 *info* **{** "dinda ${e.message }${response.code()} " **}** }  
 }  
  
 override fun onFailure(call: Call<PostResponse>, t: Throwable) {  
 progressDialog.dismiss()  
 *info* **{** "dinda ${t.message}" **}** Snackbar.make(view, "Silahkan coba lagi", Snackbar.*LENGTH\_SHORT*).show()  
 }  
  
})

# Retrofit GET dengan Recycleview

1. Response Model

data class ArrayResponse(  
 @SerializedName("message")  
 @Expose  
 val message: String,  
 @SerializedName("data")  
 @Expose  
 val data: List<TransaksiModel>  
)

1. TransaksiModel

data class TransaksiModel(  
 @SerializedName("id")  
 @Expose  
 val id: Int,  
 @SerializedName("foto")  
 @Expose  
 val foto: String,  
 )

1. ApiService

//Get Bukti Transfer  
@GET("pembayaran")  
fun getbuktitransfer(  
 @Header("Authorization") token : String,  
 @Query("email") email : String  
): Call<ArrayResponse>

1. Layout Xml

<?xml version="1.0" encoding="utf-8"?>  
<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_margin="10dp"  
 app:cardCornerRadius="8dp"  
 android:layout\_height="wrap\_content">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:padding="15dp"  
 android:layout\_height="wrap\_content">  
 <TextView  
 android:id="@+id/txttipe"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/roboto\_medium"  
 android:text="Tarik DANA"  
 android:textColor="@color/hijaumateng"  
 android:textFontWeight="600"  
 android:textSize="15dp" />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="10dp"  
 android:fontFamily="@font/roboto\_medium"  
 android:textColor="@color/black"  
 android:text="2 Jan, 19 26 WIB"  
 android:layout\_below="@+id/txttipe"  
 android:layout\_marginTop="5dp"  
 android:id="@+id/txttanggal"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="10dp"  
 android:fontFamily="@font/roboto\_medium"  
 android:textColor="@color/black"  
 android:text="Tarik dana ke rekening Bank BRI"  
 android:layout\_below="@+id/txttanggal"  
 android:layout\_marginTop="5dp"  
 android:id="@+id/txtketerangan"  
 />  
 <TextView  
 android:id="@+id/txtjumlah"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:fontFamily="@font/roboto\_medium"  
 android:text="Rp. 3000.000"  
 android:textColor="@color/black"  
 android:textSize="13dp" />  
 <TextView  
 android:id="@+id/txtstatus"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignRight="@+id/txtjumlah"  
 android:layout\_alignBottom="@id/txtketerangan"  
 android:fontFamily="@font/roboto\_medium"  
 android:text="Tarik DANA"  
 android:textColor="@color/hijaumateng"  
 android:textFontWeight="600"  
 android:textSize="11dp" />  
 </RelativeLayout>  
</com.google.android.material.card.MaterialCardView>

1. Adapter

class PembayaranViewHolder(  
 private val notesList: MutableList<TransaksiModel>,  
 private val context: Context,  
  
 ) : RecyclerView.Adapter<PembayaranViewHolder.ViewHolder>() {  
  
 //database  
 lateinit var auth: FirebaseAuth  
 var userId: String? = null  
 private var dialog: Dialog? = null  
  
  
 interface Dialog {  
 fun onClick(position: Int)  
 }  
  
 fun setDialog(dialog: Dialog) {  
 this.dialog = dialog  
 }  
  
 override fun getItemCount(): Int {  
 return notesList.size  
 }  
  
 inner class ViewHolder internal constructor(view: View) : RecyclerView.ViewHolder(view) {  
 internal var tipe: TextView  
 internal var tanggal: TextView  
 internal var keterangan: TextView  
 internal var jumlah: TextView  
 internal var status: TextView  
  
  
 init {  
 tipe = view.findViewById(R.id.*txttipe*)  
 tanggal = view.findViewById(R.id.*txttanggal*)  
 keterangan = view.findViewById(R.id.*txtketerangan*)  
 jumlah = view.findViewById(R.id.*txtjumlah*)  
 status = view.findViewById(R.id.*txtstatus*)  
  
 }  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 val view = LayoutInflater.from(parent.*context*).inflate(R.layout.*list\_transaksi*, parent, false)  
  
 return ViewHolder(view)  
  
 }  
  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
  
 val note = notesList[position]  
 holder.tipe.*text* = "${note.tipe.*toUpperCase*()}"  
 holder.tanggal.*text* = "${note.tanggal} ${note.jam}"  
 holder.keterangan.*text* = "${note.keterangan}"  
 holder.jumlah.*text* = "${note.jumlah}"  
 holder.status.*text* = "${note.status .toString()}"  
 holder.itemView.setOnClickListener **{** if(dialog != null){  
 dialog!!.onClick(holder.*layoutPosition*)  
 }  
 **}** }  
  
}

1. Class

private lateinit var mAdapter: PembayaranViewHolder

binding.rvtransaksi.*layoutManager* = LinearLayoutManager(requireContext().*applicationContext*)  
binding.rvtransaksi.setHasFixedSize(true)  
(binding.rvtransaksi.*layoutManager* as LinearLayoutManager).*orientation* =  
 LinearLayoutManager.*VERTICAL*api.getbuktitransfer(  
 sessionManager.getToken().*toString*(),  
 sessionManager.getEmail().*toString*()  
)  
 .enqueue(object : Callback<ArrayResponse> {  
 override fun onResponse(  
 call: Call<ArrayResponse>,  
 response: Response<ArrayResponse>  
 ) {  
 try {  
 if (response.*isSuccessful*) {  
 val notesList = *mutableListOf*<TransaksiModel>()  
 val data = response.body()  
 for (hasil in data!!.data) {  
 notesList.add(hasil)  
 mAdapter = PembayaranViewHolder(notesList, requireContext().*applicationContext*)  
 binding.rvtransaksi.*adapter* = mAdapter  
 mAdapter.notifyDataSetChanged()  
 }  
 } else {  
 *toast*("gagal mendapatkan response")  
 }  
 } catch (e: Exception) {  
 *info* **{** "dinda ${e.message}" **}** }  
 }  
  
 override fun onFailure(call: Call<ArrayResponse>, t: Throwable) {  
 *info* **{** "dinda ${t.message}" **}** }  
  
 })

# Retrofit Recycleview Filter

1. XML

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".SearchProdukActivity">  
  
 <SearchView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:id="@+id/search"  
 android:background="@drawable/kotakinputproduk"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/rvsearch1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="24dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/search" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/rvsearch2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="24dp"  
 android:visibility="gone"  
 android:layout\_marginEnd="24dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/search" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. Oncreate

showsearch()  
rvsearch1.*layoutManager* = LinearLayoutManager(this)  
rvsearch2.*layoutManager* = LinearLayoutManager(this)

1. New Function

fun showsearch(){  
 api.getproduct().enqueue(object :Callback<ResponseProduk>{  
 override fun onResponse(  
 call: Call<ResponseProduk>,  
 response: Response<ResponseProduk>  
 ) {  
 val datalist = response.body()  
  
 mAdapter = datalist?.*let* **{** ProdukCustomerAdapter(**it**.data) **}** rvsearch1.*adapter* = mAdapter  
  
 search.*imeOptions* = EditorInfo.*IME\_ACTION\_SEARCH* search.setOnQueryTextListener(object : SearchView.OnQueryTextListener{  
 override fun onQueryTextSubmit(action: String?): Boolean =false  
  
 override fun onQueryTextChange(action: String?): Boolean {  
 if (action!=null){  
 if (action.*isEmpty*()){  
 rvsearch1.*visibility* = View.*VISIBLE* rvsearch2.*visibility* = View.*GONE* }else if (action.length >2){  
 val filter = datalist?.data?.*filter* **{ it**.nama!!.*contains*("$action",true) **}** mAdapter = ProdukCustomerAdapter(filter as List<ProdukModel>)  
  
 if (action.*isNotEmpty*()){  
 rvsearch2.*visibility* = View.*VISIBLE* rvsearch2.*adapter* = mAdapter  
 rvsearch1.*visibility* = View.*INVISIBLE* }else{  
 rvsearch1.*visibility* = View.*VISIBLE* rvsearch2.*visibility* = View.*GONE* }  
 }  
 }  
 return false  
 }  
  
 })  
 }  
  
 override fun onFailure(call: Call<ResponseProduk>, t: Throwable) {  
 *TODO*("Not yet implemented")  
 }  
  
 })  
}

# Retrofit dengan Authorization

1. Interface

@FormUrlEncoded  
@POST("insertproduct")  
fun insertproduct(  
 @Header("Authorization") token : String,  
): Call<ResponseModel>

1. Implementasi

api.insertproduct("Bearer ${sessionManager.getToken().*toString*()}")

# Pilih gambar dari gallery

private val REQUEST\_PICK\_IMAGE = 2  
var filePath: Uri? = null

var datafoto: ByteArray? = null

binding.btnGambar.setOnClickListener **{** pilihfile()  
**}**

private fun pilihfile() {  
 val currentAPIVersion = Build.VERSION.*SDK\_INT* if (currentAPIVersion >= Build.VERSION\_CODES.*M*) {  
 if (ContextCompat.checkSelfPermission(  
 this,  
 Manifest.permission.*READ\_EXTERNAL\_STORAGE* ) != PackageManager.*PERMISSION\_GRANTED* ) {  
 if (ActivityCompat.shouldShowRequestPermissionRationale(  
 this,  
 Manifest.permission.*READ\_EXTERNAL\_STORAGE* )  
 ) {  
 dialog\_permis(  
 "External storage", Manifest.permission.*READ\_EXTERNAL\_STORAGE* )  
 } else {  
 ActivityCompat  
 .requestPermissions(  
 this,  
 *arrayOf*(Manifest.permission.*READ\_EXTERNAL\_STORAGE*),  
 REQUEST\_PICK\_IMAGE  
 )  
 }  
  
 }  
 }  
 //Intent to pick image  
 val intent = Intent(Intent.*ACTION\_PICK*)  
 intent.*type* = "image/\*"  
 startActivityForResult(intent, REQUEST\_PICK\_IMAGE)  
  
}  
  
fun dialog\_permis(  
 msg: String,  
 permission: String  
) {  
 val alertBuilder: AlertDialog.Builder = AlertDialog.Builder(this)  
 alertBuilder.setCancelable(true)  
 alertBuilder.setTitle("Permission necessary")  
 alertBuilder.setMessage("$msg permission is necessary")  
 alertBuilder.setPositiveButton(android.R.string.*yes*,  
 DialogInterface.OnClickListener **{** dialog, which **->** ActivityCompat.requestPermissions(  
 (this as Activity?)!!, *arrayOf*(permission),  
 REQUEST\_PICK\_IMAGE  
 )  
 **}**)  
 val alert: AlertDialog = alertBuilder.create()  
 alert.show()  
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 super.onActivityResult(requestCode, resultCode, data)  
 if (resultCode == *RESULT\_OK*) {  
 if (requestCode == REQUEST\_PICK\_IMAGE) {  
 filePath = data?.*data* Picasso.get().load(filePath).fit().centerCrop().into(binding.imggambar)  
 convert()  
 }  
 }  
  
}

fun convert(){  
 val bmp = MediaStore.Images.Media.getBitmap(*contentResolver*, filePath)  
 val baos = ByteArrayOutputStream()  
 bmp.compress(Bitmap.CompressFormat.*JPEG*, 25, baos)  
 datafoto = baos.toByteArray()  
  
}

# Upload Foto FirebaseStorage

private var storageReference: StorageReference? = null

private fun uploadfile() {  
val fileref =  
 FirebaseStorage.getInstance().*reference*.child(  
 System.currentTimeMillis().toString() + ".jpg"  
 )  
var uploadTask: StorageTask<\*>  
uploadTask = fileref.putBytes(datafoto!!)  
uploadTask.continueWithTask(*Continuation*<UploadTask.TaskSnapshot, Task<Uri>> **{** task **->** if (!task.*isSuccessful*) {  
 }  
 return@Continuation fileref.*downloadUrl***}**).addOnCompleteListener **{** task **->** val downloadUrl = task.*result***}**

}

# Chat Message dengan Firebase

1. Hubungkan akun dengan firebase
2. Implementation Groupy

implementation "com.xwray:groupie:2.1.0"

1. Buat model ChatMessage

class ChatMessage(  
 val id: String,  
 val text: String,  
 val fromId: String,  
 val toId: String,  
 val timestamp: Long  
) {  
 constructor() : this("", "", "", "", -1)  
}

1. Buat DateUtils

object DateUtils {  
  
 val fullFormattedTime = SimpleDateFormat("d MMM, h:mm a", Locale.*US*) // the format of your date  
 val onlyTime = SimpleDateFormat("h:mm a", Locale.*US*) // the format of your date  
 val onlyDate = SimpleDateFormat("d MMM", Locale.*US*) // the format of your date  
  
 fun getFormattedTime(timeInMilis: Long): String {  
 val date = Date(timeInMilis \* 1000L) // \*1000 is to convert seconds to milliseconds  
  
 return when {  
 isToday(date) -> onlyTime.format(date)  
 isYesterday(date) -> "Yesterday"  
 else -> onlyDate.format(date)  
 }  
  
 }  
  
 fun getFormattedTimeChatLog(timeInMilis: Long): String {  
 val date = Date(timeInMilis \* 1000L) // \*1000 is to convert seconds to milliseconds  
 val fullFormattedTime = SimpleDateFormat("d MMM, h:mm a", Locale.*US*) // the format of your date  
 val onlyTime = SimpleDateFormat("h:mm a", Locale.*US*) // the format of your date  
  
 return when {  
 isToday(date) -> onlyTime.format(date)  
 else -> fullFormattedTime.format(date)  
 }  
  
 }  
  
 fun isYesterday(d: Date): Boolean {  
 return DateUtils.isToday(d.*time* + DateUtils.*DAY\_IN\_MILLIS*)  
 }  
  
 fun isToday(d: Date): Boolean {  
 return DateUtils.isToday(d.*time*)  
 }  
}

1. XML

<androidx.swiperefreshlayout.widget.SwipeRefreshLayout xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/swiperefresh"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/white"  
 tools:context=".messages.ChatLogActivity">  
 <com.google.android.material.appbar.AppBarLayout  
 android:id="@+id/appBarLayout3"  
 android:layout\_width="match\_parent"  
 android:background="@color/white"  
 android:layout\_height="60dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" >  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
 <RelativeLayout  
 android:layout\_width="40dp"  
 android:id="@+id/btn\_back"  
 android:layout\_height="match\_parent"  
 >  
 <ImageButton  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/ic\_baseline\_arrow\_back\_24"  
 android:layout\_centerVertical="true"  
 android:layout\_marginLeft="20dp"  
 />  
  
  
 </RelativeLayout>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Nama Driver"  
 android:fontFamily="@font/roboto\_mono"  
 android:textSize="14dp"  
 android:id="@+id/txt\_uidlawan"  
 android:textColor="#204975"  
 android:layout\_marginTop="10dp"  
 android:layout\_toRightOf="@id/btn\_back"  
 android:layout\_marginLeft="10dp"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Plat Nomor"  
 android:fontFamily="@font/roboto\_mono"  
 android:textSize="14dp"  
 android:layout\_below="@+id/txt\_uidlawan"  
 android:id="@+id/txt\_platnomor"  
 android:textColor="#A4A4A4"  
 android:layout\_marginTop="5dp"  
 android:layout\_toRightOf="@id/btn\_back"  
 android:layout\_marginLeft="20dp"  
 />  
  
  
  
  
 </RelativeLayout>  
 </com.google.android.material.appbar.AppBarLayout>  
  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerview\_chat\_log"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="#e4e4e4"  
 app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"  
 app:layout\_constraintBottom\_toTopOf="@+id/buttons\_container"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/appBarLayout3" />  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/buttons\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="#e4e4e4"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent">  
  
 <EditText  
 android:id="@+id/edittext\_chat\_log"  
 android:layout\_width="0dp"  
 android:layout\_height="45dp"  
 android:layout\_marginBottom="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:ems="10"  
 android:hint="Ketik Pesan...."  
 android:inputType="text"  
 android:paddingLeft="16dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/btnsend"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/btnsend"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginTop="8dp"  
 android:src="@drawable/ic\_baseline\_send\_24"  
 app:fabCustomSize="45dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:layout\_marginRight="16dp" >  
 </com.google.android.material.floatingactionbutton.FloatingActionButton>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

1. XML chat\_from\_row

<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="12dp"  
 android:layout\_marginTop="10dp"  
 >  
  
 <TextView  
 android:id="@+id/txt\_chatdriver"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:background="@drawable/kotakchatkiri"  
 android:maxWidth="240dp"  
 android:paddingStart="16dp"  
 android:paddingLeft="16dp"  
 android:paddingTop="8dp"  
 android:paddingEnd="20dp"  
 android:paddingRight="20dp"  
 android:paddingBottom="8dp"  
 android:text="This is my message that will wrap into multiple lines and keep going"  
 android:textColor="@android:color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.11"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_timedriver"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginRight="8dp"  
 android:text="16 Sep, 2:05 PM"  
 android:textSize="11sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="@+id/txt\_chatdriver"  
 app:layout\_constraintHorizontal\_bias="0.11"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_chatdriver" />  
  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. XML chat\_to\_row

<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="12dp"  
 android:layout\_marginTop="15dp"  
 >  
  
 <TextView  
 android:id="@+id/txt\_chatuser"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:background="@drawable/kotakchatkanan"  
 android:maxWidth="240dp"  
 android:paddingStart="16dp"  
 android:paddingLeft="16dp"  
 android:paddingTop="8dp"  
 android:paddingEnd="20dp"  
 android:paddingRight="20dp"  
 android:paddingBottom="8dp"  
 android:text="This is my message that will wrap into multiple lines and keep going"  
 android:textColor="@android:color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.93"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_timeuser"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginTop="4dp"  
 android:text="16 Sep, 2:05 PM"  
 android:textSize="11sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="@+id/txt\_chatuser"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_chatuser"  
 android:layout\_marginRight="8dp" />  
  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. Class

private fun sendmessage() {  
 val text = edittext\_chat\_log.*text*.toString()  
 if (text.*isEmpty*()) {  
 *toast*("Jangan kosongi pesan")  
 return  
 }  
  
 val fromId = sessionManager.getReferal().*toString*()  
 val toId = "admin"  
  
 val notificationMessage: MutableMap<String, Any> =  
 HashMap()  
 notificationMessage["message"] = text  
 notificationMessage["from"] = fromId  
  
 val reference = FirebaseDatabase.getInstance().getReference("/chat/user-messages/$fromId/admin").push()  
 val toReference = FirebaseDatabase.getInstance().getReference("/chat/user-messages/admin/$fromId").push()  
  
 val chatMessage = ChatMessage(reference.*key*!!, text, fromId, toId, System.currentTimeMillis() / 1000)  
 reference.setValue(chatMessage)  
 .addOnSuccessListener **{** edittext\_chat\_log.*text*.clear()  
 recyclerview\_chat\_log.smoothScrollToPosition(adapter.*itemCount* - 1)  
 **}** toReference.setValue(chatMessage)  
  
  
 val latestMessageRef = FirebaseDatabase.getInstance().getReference("/chat/statusnotif/admin")  
 latestMessageRef.child("status").setValue(true)  
  
 val latestMessageToRef = FirebaseDatabase.getInstance().getReference("/chat/latest-messages/admin/$fromId")  
 latestMessageToRef.setValue(chatMessage)  
}  
  
private fun ambildata() {  
 binding.swiperefresh.*isEnabled* = true  
 binding.swiperefresh.*isRefreshing* = true  
  
 val fromId = sessionManager.getReferal().*toString*()  
 val toId = "admin"  
 val ref = FirebaseDatabase.getInstance().getReference("/chat/user-messages/$fromId/$toId")  
  
 ref.addListenerForSingleValueEvent(object : ValueEventListener {  
 override fun onCancelled(databaseError: DatabaseError) {  
 }  
  
 override fun onDataChange(dataSnapshot: DataSnapshot) {  
  
 if (!dataSnapshot.hasChildren()) {  
 binding.swiperefresh.*isRefreshing* = false  
 binding.swiperefresh.*isEnabled* = false  
 }  
 }  
 })  
  
 ref.addChildEventListener(object : ChildEventListener {  
 override fun onCancelled(databaseError: DatabaseError) {  
 }  
  
 override fun onChildMoved(dataSnapshot: DataSnapshot, previousChildName: String?) {  
 }  
  
 override fun onChildChanged(dataSnapshot: DataSnapshot, previousChildName: String?) {  
 }  
  
 override fun onChildAdded(dataSnapshot: DataSnapshot, previousChildName: String?) {  
 dataSnapshot.getValue(ChatMessage::class.*java*)?.*let* **{** if (**it**.fromId == sessionManager.getReferal().*toString*()) {  
 adapter.add(ChatToItem(**it**.text, **it**.timestamp))  
  
 } else {  
 adapter.add(ChatFromItem(**it**.text, **it**.timestamp))  
  
 }  
 **}** binding.recyclerviewChatLog.scrollToPosition(adapter.*itemCount* - 1)  
 binding.swiperefresh.*isRefreshing* = false  
 binding.swiperefresh.*isEnabled* = false  
 }  
  
 override fun onChildRemoved(dataSnapshot: DataSnapshot) {  
 }  
  
 })  
}  
  
class ChatFromItem(val text: String, val timestamp: Long) : Item<ViewHolder>() {  
  
 override fun bind(viewHolder: ViewHolder, position: Int) {  
  
  
 viewHolder.itemView.txt\_chatdriver.*text* = text  
 viewHolder.itemView.txt\_timedriver.*text* = getFormattedTimeChatLog(timestamp)  
  
  
 /\* if (!user.profileImageUrl!!.isEmpty()) {  
 val requestOptions = RequestOptions().placeholder(R.drawable.no\_image2)  
 Glide.with(targetImageView.context)  
 .load(user.profileImageUrl)  
 .thumbnail(0.1f)  
 .apply(requestOptions)  
 .into(targetImageView)  
 }\*/  
 }  
  
 override fun getLayout(): Int {  
 return R.layout.*chat\_from\_row* }  
  
}  
  
class ChatToItem(val text: String, val timestamp: Long) : Item<ViewHolder>() {  
  
 override fun bind(viewHolder: ViewHolder, position: Int) {  
 viewHolder.itemView.txt\_chatuser.*text* = text  
 viewHolder.itemView.txt\_timeuser.*text* = getFormattedTimeChatLog(timestamp)  
  
 }  
  
 override fun getLayout(): Int {  
 return R.layout.*chat\_to\_row* }  
  
}

1. Cara penggunaan

binding.swiperefresh.setColorSchemeColors(ContextCompat.getColor(requireContext().*applicationContext*,R.color.*abumatang*))  
binding.recyclerviewChatLog.*adapter* = adapter  
ambildata()  
binding.btnsend.setOnClickListener **{** sendmessage()  
**}**