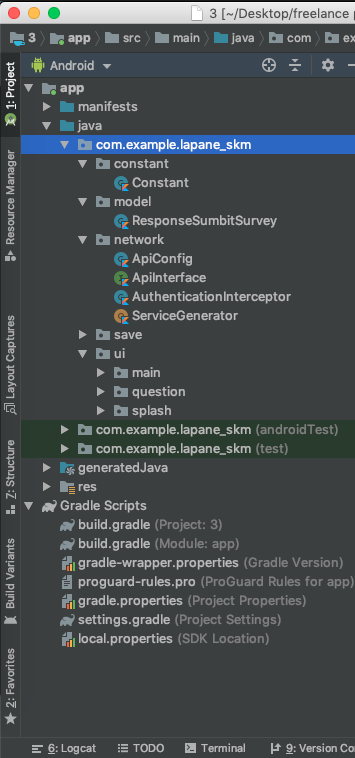
Laporan SKM



Pada projek SKM terdapat 5 package (folder) utama yaitu contant, model, network, save dan Ui, berikut penjelasan masing-masing package da nisi dalam class dalam package.

1. Constant

Pada package constant terdapat 1 file class Constant, berikut isi dari file constant

open class Constant {  
  
 val USERNAME = "eskm\_kshu"  
 val PASS ="Enigm4777$#"  
}

isi dari file constant ada 2 variabel untuk menampung username dan password yang dipakai pada basic auth ketika hit ke server.

1. Model

Pada package model terdapat 1 class yaitu ResponseSubmitSurvey

package com.example.lapane\_skm.model  
  
import com.google.gson.annotations.SerializedName  
  
data class ResponseSumbitSurvey(  
  
 @field:SerializedName("usia")  
 val usia: String? = null,  
  
 @field:SerializedName("persyaratan")  
 val persyaratan: String? = null,  
  
 @field:SerializedName("id\_isi")  
 val idIsi: String? = null,  
  
 @field:SerializedName("tgl\_survey")  
 val tglSurvey: String? = null,  
  
 @field:SerializedName("layanan")  
 val layanan: String? = null,  
  
 @field:SerializedName("id\_survey")  
 val idSurvey: String? = null,  
  
 @field:SerializedName("gender")  
 val gender: String? = null,  
  
 @field:SerializedName("pendidikan")  
 val pendidikan: String? = null,  
  
 @field:SerializedName("id\_sator")  
 val idSator: String? = null,  
  
 @field:SerializedName("perilaku")  
 val perilaku: String? = null,  
  
 @field:SerializedName("kompetensi")  
 val kompetensi: String? = null,  
  
 @field:SerializedName("pekerjaan")  
 val pekerjaan: String? = null,  
  
 @field:SerializedName("sarpras")  
 val sarpras: String? = null,  
  
 @field:SerializedName("penanganan")  
 val penanganan: String? = null,  
  
 @field:SerializedName("smp")  
 val smp: String? = null,  
  
 @field:SerializedName("waktu")  
 val waktu: String? = null,  
  
 @field:SerializedName("jenis")  
 val jenis: String? = null  
)

isi dari class ResponseSubmitSurvey yaitu sebagai pojo untuk menampung hasil response ketika submit jawaban pertanyaan.

1. Network

Isi dari package network sebagai berikut :

-ApiConfig = file ini sudah tidak dipakai

-ApiInterface

package com.example.lapane\_skm.network  
  
import com.example.lapane\_skm.model.ResponseSumbitSurvey  
import retrofit2.Call  
import retrofit2.http.Field  
import retrofit2.http.FormUrlEncoded  
import retrofit2.http.GET  
import retrofit2.http.POST  
  
interface ApiInterface {  
  
 @POST("survey")  
 @FormUrlEncoded  
 fun postData(  
 @Field("sator")sator : Int,  
 @Field("layanan") layanan : Int,  
 @Field("usia") usia : String,  
 @Field("gender") gender : String,  
 @Field("pendidikan") pendidikan : String,  
 @Field("pekerjaan") pekerjaan : String,  
 @Field("persyaratan") persyaratan : Int,  
 @Field("smp") smp : Int,  
 @Field("waktu") waktu : Int,  
 @Field("jenis") jenis : Int,  
 @Field("kompetensi") kompotensi : Int,  
 @Field("perilaku") perilaku : Int,  
 @Field("penanganan") penanganan : Int,  
 @Field("sarpras") sarpras : Int  
  
 ): Call<ResponseSumbitSurvey>  
}

api interface untuk menampung parameter ketika post data di server, apiinterface salah satu kelas yang dibutuhkan ketika menggunakan libaray retrofit2.

-AutenticationInterceptor

package com.example.lapane\_skm.network  
  
import okhttp3.Interceptor  
import okhttp3.Request  
import okhttp3.Response  
  
import java.io.IOException  
  
class AuthenticationInterceptor(private val authToken: String) : Interceptor {  
  
 @Throws(IOException::class)  
 override fun intercept(chain: Interceptor.Chain): Response {  
 val original = chain.request()  
  
 val builder = original.newBuilder()  
 .header("Authorization", authToken)  
  
 val request = builder.build()  
 return chain.proceed(request)  
 }  
}

AutenticationInterceptor yaitu sebagai header ketika menggunakan library retrofit2 dengan menggunakan fitur auth.

-ServiceGenerator

package com.example.lapane\_skm.network  
  
import android.text.TextUtils  
import okhttp3.Credentials  
import okhttp3.OkHttpClient  
import retrofit2.Retrofit  
import retrofit2.converter.gson.GsonConverterFactory  
  
  
object ServiceGenerator {  
  
  
 val API\_BASE\_URL = "http://103.51.131.60/service-eskm/api/"  
  
 private val httpClient = OkHttpClient.Builder()  
  
 private val builder = Retrofit.Builder()  
 .baseUrl(API\_BASE\_URL)  
 .addConverterFactory(GsonConverterFactory.create())  
  
 private var retrofit = builder.build()  
  
 fun <S> createService(serviceClass: Class<S>): S {  
 return createService(serviceClass, null, null)  
 }  
  
 fun <S> createService(  
 serviceClass: Class<S>, username: String?, password: String?  
 ): S {  
 if (!TextUtils.isEmpty(username) && !TextUtils.isEmpty(password)) {  
 val authToken = Credentials.basic(username, password)  
 return createService(serviceClass, authToken)  
 }  
  
 return createService(serviceClass, null)  
 }  
  
 fun <S> createService(  
 serviceClass: Class<S>, authToken: String?  
 ): S {  
 if (!TextUtils.isEmpty(authToken)) {  
 val interceptor = AuthenticationInterceptor(authToken!!)  
  
 if (!httpClient.interceptors().contains(interceptor)) {  
 httpClient.addInterceptor(interceptor)  
  
 builder.client(httpClient.build())  
 retrofit = builder.build()  
 }  
 }  
  
 return retrofit.create(serviceClass)  
 }  
}

ServiceGenerator adalah satu kelas yang menghendle fungsi ketika menggunakan library retrofit2 yaitu untk hendel hit ke API.

1. Save

Pada package save terdapat 14 class, semua class fungsinya sama yaitu untuk menyimpan jawaban dari pertanyaan yang dijawab, salah satu contoh seperti baris kode dibawah ini :

PrevSafeQuestionSatu

package com.example.lapane\_skm.save  
  
import android.content.Context  
import android.content.SharedPreferences  
  
class PrefSaveQuestionSatu (private val context: Context){  
  
 val sp: SharedPreferences = context.getSharedPreferences("image",0)  
 val editor: SharedPreferences.Editor = sp.edit()  
  
 fun setData(image:String?){  
 editor.putString("uid",image)  
 editor.commit()  
 }  
  
 fun getData() : String?{  
  
 return sp.getString("uid",null)  
  
 }  
}

1. Ui



pada package ui terdapat 3 package lagi yaitu sebagai berikut :

-package main

\*BaseActivity

package com.example.lapane\_skm.ui.main  
  
import android.content.Intent  
import android.os.Bundle  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
import com.afollestad.materialdialogs.MaterialDialog  
import com.example.lapane\_skm.R  
import com.valdesekamdem.library.mdtoast.MDToast  
import kotlin.system.exitProcess  
  
open class BaseActivity : AppCompatActivity() {  
  
  
 lateinit var loading: MaterialDialog  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 val materialDialog = MaterialDialog.Builder(this)  
 materialDialog.progress(true, 0)  
 materialDialog.cancelable(false)  
 loading = materialDialog.build()  
  
 }  
  
 fun intentTo(target: Class<\*>) {  
 startActivity(Intent(this, target))  
 }  
  
fun showErrorMessage(message: String?) {  
 MDToast.makeText(this, message, MDToast.*LENGTH\_SHORT*, MDToast.*TYPE\_ERROR*).show()  
 }  
  
 fun showSuccessMessage(message: String?) {  
 MDToast.makeText(this, message, MDToast.*LENGTH\_SHORT*, MDToast.*TYPE\_SUCCESS*).show()  
 }  
  
 fun showInfoMessage(message: String?) {  
 MDToast.makeText(this, message, MDToast.*LENGTH\_SHORT*, MDToast.*TYPE\_INFO*).show()  
 }  
  
 fun showLongErrorMessage(message: String?) {  
 MDToast.makeText(this, message, MDToast.*LENGTH\_LONG*, MDToast.*TYPE\_ERROR*).show()  
 }  
  
 fun showLongSuccessMessage(message: String?) {  
 MDToast.makeText(this, message, MDToast.*LENGTH\_LONG*, MDToast.*TYPE\_SUCCESS*).show()  
 }  
  
 fun showLongInfoMessage(message: String?) {  
 MDToast.makeText(this, message, MDToast.*LENGTH\_LONG*, MDToast.*TYPE\_INFO*).show()  
 }  
  
 fun showProgressDialog() {  
 loading.setContent(R.string.*progress\_dialog*)  
 loading.show()  
 }  
 fun showProgressDialog(message: String?) {  
 loading.setContent(message)  
 loading.show()  
 }  
  
 fun updateMessageDialog(message: String?) {  
 loading.setContent(message)  
 }  
  
 fun dismissProgressDialog() {  
 if (loading.*isShowing*)  
 loading.dismiss()  
 }  
  
 override fun onBackPressed() {  
 val builder = AlertDialog.Builder(this@BaseActivity)  
 builder.setTitle("Info ")  
 builder.setMessage("apakah anda ingin membatalkan pengisian pertanyaan ?")  
 //builder.setPositiveButton("OK", DialogInterface.OnClickListener(function = x))  
  
 builder.setPositiveButton(android.R.string.*yes*) **{** dialog, which **->** finish()  
 *exitProcess*(0)  
  
 **}** builder.setNegativeButton(android.R.string.*no*) **{** dialog, which **->  
  
 }** builder.show()  
 }  
  
  
  
}

class BaseActivity yaitu sebagai base yang terdapat beberapa fungsi di dalamnya seperti memnunculkan dialog, pesan dan alertdialog. Class ini di extend dibeberapa activity untuk di pakai fungsi di dalamnya.

\*MainActivity

package com.example.lapane\_skm.ui.main  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import com.example.lapane\_skm.R  
import com.example.lapane\_skm.save.\*  
import com.example.lapane\_skm.ui.question.questionsatu.QuestionSatuActivity  
import kotlinx.android.synthetic.main.activity\_main.\*  
import org.jetbrains.anko.startActivity  
import kotlin.system.exitProcess  
  
class MainActivity : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 tv\_mulai\_survey.setOnClickListener **{** this?.*let* **{** PrefSaveQuestionSatu(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestionDua(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestionTiga(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestionEmpat(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestionLima(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion6(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion7(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion8(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion9(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion10(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion11(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion12(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion13(**it**).setData("null") **}** this?.*let* **{** PrefSaveQuestion14(**it**).setData("null") **}** *startActivity*<QuestionSatuActivity>()  
 **}** }  
  
 override fun onBackPressed() {  
 super.onBackPressed()  
 finish()  
 }  
}

MainActivity adalah activity yang ke dua muncul setelah page splashscreen, pada activity ini ada tombol submit untuk memulai menjawab survey. Class MainActivity ketika button di klik semua penyimpanan yang disimpan di preference di set null.

- Package Question

pada package ini menyimpan ui (activity) pertanyaan, semua activity atau fragmentnya sama yaitu untuk menampung pertanyaan, baris kodenya salah satunya seperti berikut :

\*QuestionSatuActivity

package com.example.lapane\_skm.ui.question.questionsatu  
  
import android.os.Bundle  
import android.view.LayoutInflater  
import android.view.View  
import androidx.appcompat.app.AlertDialog  
import androidx.fragment.app.Fragment  
import androidx.recyclerview.widget.LinearLayoutManager  
import com.example.lapane\_skm.R  
import com.example.lapane\_skm.save.\*  
import com.example.lapane\_skm.ui.main.BaseActivity  
import com.example.lapane\_skm.ui.main.MainActivity  
import com.example.lapane\_skm.ui.question.fragment\_dua.QuestionDuaFragment  
import kotlinx.android.synthetic.main.activity\_question\_satu.\*  
import kotlinx.android.synthetic.main.layout\_recyclerview\_dialog.\*  
  
class QuestionSatuActivity : BaseActivity() {  
  
 var isDataQuestion : String? = null  
 private var savedInstanceState: Bundle? = null  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_question\_satu*)  
  
 tv\_silahkan\_pilih\_question.*text* = "Silahkan Pilih satuan Kerja"  
  
  
 tv\_silahkan\_pilih\_question.setOnClickListener **{** listDataQuestion()  
  
 **}** isDataQuestion = PrefSaveQuestionSatu(this).getData().*toString*()  
  
 if (isDataQuestion == null){  
  
 }  
 tv\_next.*visibility* = View.*GONE* tv\_prev.*visibility* = View.*GONE* tv\_next.setOnClickListener **{** openFragment(QuestionDuaFragment())  
  
  
 **}** }  
  
 private fun listDataQuestion() {  
  
 var data: List<ModelSatuanKerja> =  
 *listOf*(  
 ModelSatuanKerja(  
 "1",  
 "Biro Kerjasama, Humas, dan Umum, Jakarta"  
 ),  
 ModelSatuanKerja(  
 "2",  
 "Pusat Pemanfaatan Penginderaan Jauh, Jakarta"  
 ),  
 ModelSatuanKerja(  
 "3",  
 "Pusat Teknologi dan Data Penginderaan Jauh, Jakarta"  
 ),  
 ModelSatuanKerja(  
 "4",  
 "Pusat Sains dan Teknologi Atmosfer, Bandung "  
 ),  
 ModelSatuanKerja("5", "Pusat Sains Antariksa, Bandung"),  
 ModelSatuanKerja(  
 "6",  
 "Pusat Teknologi Penerbangan, Bogor "  
 ),  
 ModelSatuanKerja("7", "Pusat Teknologi Roket, Bogor "),  
 ModelSatuanKerja(  
 "8",  
 "Pusat Teknologi Satelit, Bogor "  
 ),  
 ModelSatuanKerja(  
 "9",  
 "Pusat Teknologi Informasi dan Komunikasi Penerbangan dan Antariksa, Jakarta "  
 ),  
 ModelSatuanKerja(  
 "10",  
 "Balai Pengamatan Antariksa dan Atmosfer, Pasuruan "  
 ),  
 ModelSatuanKerja(  
 "11",  
 "Balai Pengamatan Antariksa dan Atmosfer, Sumedang "  
 ),  
 ModelSatuanKerja(  
 "12",  
 "Stasiun Bumi Penginderaan Jauh, Parepare "  
 ),  
 ModelSatuanKerja(  
 "13",  
 "Balai Kendali Satelit, Pengamatan Antariksa dan Atmosfer dan Penginderaan Jauh, Biak"  
 ),  
 ModelSatuanKerja(  
 "14",  
 "Balai Uji Teknologi dan Pengamatan Antariksa dan Atmosfer, Garut "  
 ),  
 ModelSatuanKerja(  
 "15",  
 "Balai Pengamatan Antariksa dan Atmosfer, Agam"  
 ),  
 ModelSatuanKerja(  
 "16",  
 "Balai Pengamatan Antariksa dan Atmosfer, Pontianak "  
 )  
 )  
  
  
 val mDialogView = LayoutInflater.from(this).inflate(R.layout.*layout\_recyclerview\_dialog*, null)  
 val mBuilder = AlertDialog.Builder(this)  
 .setView(mDialogView)  
 .setTitle("Pilih Satuan Kerja")  
 //show dialog  
 val mAlertDialog = mBuilder.show()  
  
 val rvSatuanKerja = mAlertDialog.rv\_lokasi\_satuan\_kerja  
 rvSatuanKerja.*layoutManager* = LinearLayoutManager(this)  
 val adapterSatuanKerja = SatuanKerjaAdapter(this, data)  
 rvSatuanKerja.*adapter* = adapterSatuanKerja  
 adapterSatuanKerja.notifyDataSetChanged()  
  
 adapterSatuanKerja.setOnItemClickListener(object :  
 SatuanKerjaAdapter.OnItemClickListener {  
 override fun onItemClick(position: Int) {  
  
 mAlertDialog.dismiss()  
 tv\_silahkan\_pilih\_question.*text* = data.get(position).title.*toString*()  
 PrefSaveQuestionSatu(this@QuestionSatuActivity).setData(data.get(position).id)  
 tv\_next.*visibility* = View.*VISIBLE* }  
  
  
 })  
  
   
  
  
  
 }  
  
  
  
 public fun openFragment(fragment: Fragment) {  
 if (savedInstanceState == null) {  
 *supportFragmentManager* .beginTransaction()  
 .replace(R.id.*root\_question\_satu*, fragment, fragment.*javaClass*.*simpleName*)  
 .commit()  
 }  
 }  
  
  
  
}

pada setiap activity/fragment question fungsinya untuk menampilkan pertanyaan yang ada dalam method listDataQuestion(), data itu di tampilkan di adapater. Setiap activity/fragment mempunyai adapter untuk menampung / manajamen list pertanyaan, salah satu contoh adapter sebagai berikut :

\*SatuanKerjaAdapter

package com.example.lapane\_skm.ui.question.questionsatu  
  
import android.content.Context  
import android.net.Uri  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import androidx.recyclerview.widget.RecyclerView  
import com.example.lapane\_skm.R  
import kotlinx.android.synthetic.main.item\_layout.view.\*  
  
class SatuanKerjaAdapter (var context: Context?, var list: List<ModelSatuanKerja>) :  
 RecyclerView.Adapter<SatuanKerjaAdapter.ViewHolder>() {  
  
 private var mOnItemClickListener: OnItemClickListener? = null  
 private lateinit var uri: Uri  
  
 interface OnItemClickListener {  
 fun onItemClick(position: Int)  
 }  
  
 // fun mActivity(activity: Activity) {  
// this.activity = activity  
// }  
 fun setOnItemClickListener(mItemClickListener: OnItemClickListener) {  
 this.mOnItemClickListener = mItemClickListener  
 }  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 holder.tvTitle.*text* = list.get(position).title  
 holder.itemView.setOnClickListener **{** mOnItemClickListener?.onItemClick(position)  
 **}** }  
  
 override fun onCreateViewHolder(p0: ViewGroup, viewType: Int): ViewHolder {  
 var v = LayoutInflater.from(context).inflate(R.layout.*item\_layout*, p0, false)  
 return ViewHolder(v)  
 }  
  
 override fun getItemCount(): Int {  
 return list.size  
 }  
  
  
  
 class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val tvTitle = itemView.tv\_title  
  
 }  
}

fungsi adapter sendiri agar bisa manajemen list yang banyak dan lebih dinamis, adapater di panggil di activity / fragment sedangkan datanya di inisialisasi di activity atau di fragment.

- Package Splash

\*SplashActivity

splashActivtiy sebagai page pertama kali yang muncul ketika aplikasi di buka, page ini hanya menampilkan informasi awal ketika aplikasi dibuka berikut baris kode dari splashactivity :

package com.example.lapane\_skm.ui.splash  
  
import android.os.Build  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.os.Handler  
import com.example.lapane\_skm.R  
import com.example.lapane\_skm.ui.main.BaseActivity  
import com.example.lapane\_skm.ui.main.MainActivity  
  
class SplashActivity : BaseActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_splash*)  
  
 Handler().postDelayed(**{** if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*JELLY\_BEAN*) {  
 intentTo(MainActivity::class.*java*)  
 }  
 finish()  
 **}**, 5000)  
 }  
}

splashactivity terdapat fungsi handler yang bertugas memberi jedah sekian detik untuk menampilkan page splash ketika aplikasi dibuka pertama kali.