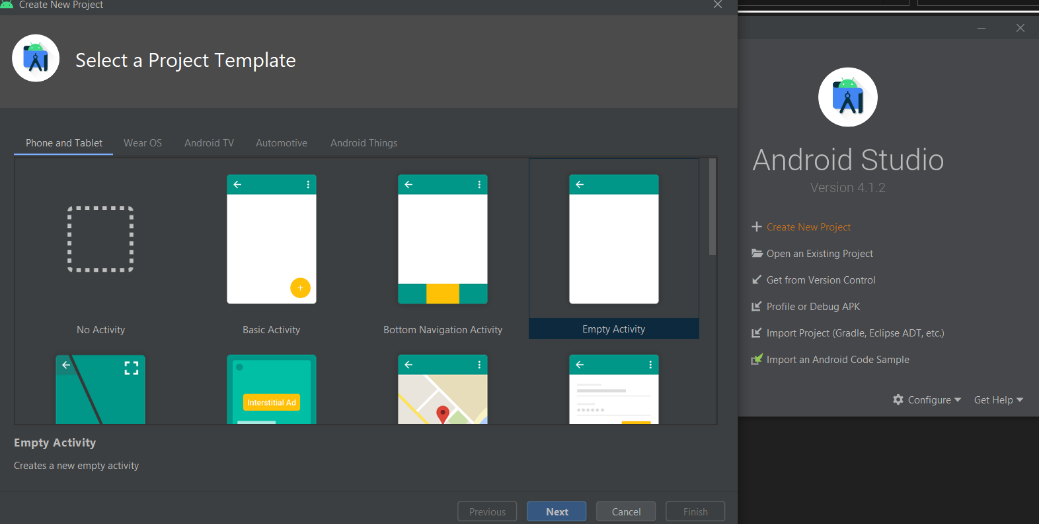
**Tugas Pertemuan Ke – 6**

**Bab 6 :WORKMANAGER**

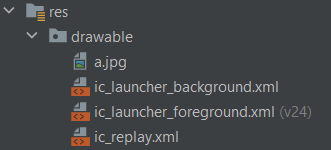
|  |  |
| --- | --- |
| **Nama** | Arga Enusjaya Putra |
| **Nim** | 1918111 |
| **Kelas** | B |
| **Pemberi Tugas** | Fernanda Kurnia Sella |

1. **Nama projek yang ditugaskan (**Membuat project sederhana mengimplementasikan workmanager didalamnya**)**
2. **Langkah – langkah pengerjaan**
3. Buat *project* dengan klik *Create New Project* >> *Empty Activity* >> berikan nama projek dengan bahasa *Java* >> Klik *Finish*



Gambar 6.1 : Halaman awal create projek android studio

1. Buat *Asset* dan masukan kedalam folder *res* >> *drawable* untuk nantinya digunakan mendesain UI tampilan aplikasi yang dapat berupa file *.xml*



Gambar 6.2 : Memasukan Vector Asset

1. Buka *Graddle* *Script* pada *build*.*graddle* lalu tambahkan *dependency* dibawah ini, lalu klik “*Sync* *Now*” pada *pop up* *graddle* yang akan muncul.

|  |
| --- |
| implementation 'com.loopj.android:android-async-http:1.4.11'  implementation "androidx.work:work-runtime:2.7.1" |

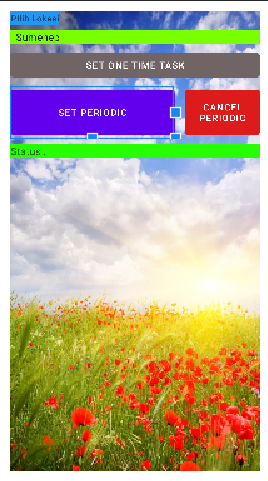
1. Lalu buka gradle.properties yang didalamnya nanti berisi api key yang telah digenerate melaui website

|  |
| --- |
| Weather\_ApiKey = "ebe30b3ceea0587df02fb16aa1664c5b" |

1. Pada *activity*\_*main*.*xml* buat tampilan *layout* yang berisi design

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout 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"  android:orientation="vertical"  android:layout\_margin="16dp"  android:background="@drawable/a"  tools:context=".MainActivity">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:background="#2196F3"  android:text="Pllih Lokasi"  android:textStyle="bold" />  <Spinner  android:id="@+id/sp\_city"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  android:background="#78FF00"  android:entries="@array/city\_list"  android:spinnerMode="dialog" />  <Button  android:id="@+id/btn\_one\_time\_task"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  android:backgroundTint="#706464"  android:text="Set One Time Task" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_marginTop="8dp">  <Button  android:id="@+id/btn\_periodic\_task"  android:layout\_width="match\_parent"  android:layout\_height="80dp"  android:layout\_marginRight="8dp"  android:layout\_weight="1"  android:text="Set Periodic" />  <Button  android:id="@+id/btn\_cancel\_periodic\_task"  android:layout\_width="235dp"  android:layout\_height="match\_parent"  android:layout\_marginLeft="8dp"  android:layout\_weight="1"  android:backgroundTint="#DD1B1B"  android:enabled="false"  android:text="Cancel Periodic"  android:textColor="#FFFFFF" />  </LinearLayout>  <TextView  android:id="@+id/tv\_status"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  android:background="#26FF00"  android:text="Status : "  android:textSize="16sp" />  </LinearLayout> |

Hasil dari tampilan yang telah kita buat dengan memasukan.



Gambar 6.3 : Tampilan activity\_main.xml

1. Tambahkan *SourceCode* pada bagian *Values* >> *String* untuk memilih kota yang ingin ditampikan nanti cuacanya.

|  |
| --- |
| <resources>  <string name="app\_name">TugasP6\_WorkManager</string>  <string-array name="city\_list">  <item>Sumenep</item>  <item>Pamekasan</item>  <item>Sampang</item>  <item>Bangkalan</item>  </string-array>  </resources> |

1. Buatlah *java class* yang bernama WeatherCityWorkmanager

Graphical user interface, application

Description automatically generated

Gambar 6.4 : Membuat class baru java

1. Buatlah *Button* pada *layout* activity\_main.*xml* seperti pada dibawah

Tabel 6.1 : activity\_main.xml

|  |  |  |
| --- | --- | --- |
| No. | Nama Komponen | Keterangan |
| 1 | Button1 | @+id/btn\_one\_time\_task |
| 2 | Button2 | @+id/btn\_periodic\_task |
| 3 | Button3 | @+id/btn\_cancel\_periodic\_task |

*Source Code WeatherCityWorkmanager*

|  |
| --- |
| package com.example.tugasp6\_workmanager;  import android.app.Notification;  import android.app.NotificationChannel;  import android.app.NotificationManager;  import android.content.Context;  import android.media.RingtoneManager;  import android.net.Uri;  import android.os.Build;  import android.os.Looper;  import android.util.Log;  import androidx.annotation.NonNull;  import androidx.core.app.NotificationCompat;  import androidx.core.content.ContextCompat;  import androidx.work.Worker;  import androidx.work.WorkerParameters;  import com.loopj.android.http.AsyncHttpClient;  import com.loopj.android.http.AsyncHttpResponseHandler;  import com.loopj.android.http.SyncHttpClient;  import org.json.JSONException;  import org.json.JSONObject;  import java.text.DecimalFormat;  import cz.msebera.android.httpclient.Header;  public class WeatherCityWorkManager extends Worker {  private static final String CHANNEL\_ID = "Work\_Manager\_Channel01";  private static final CharSequence CHANNEL\_NAME = "WorkManagerChannel";  public static final String EXTRA\_CITY = "city" ;  private static final String TAG = WeatherCityWorkManager.class.getSimpleName();  private Result resultStatus;  public WeatherCityWorkManager(@NonNull Context context, @NonNull WorkerParameters workerParams) {  super(context, workerParams);  }  @NonNull  @Override  public Result doWork() {  String city = getInputData().getString(EXTRA\_CITY);  return getCurrentWeather(city);  }  private Result getCurrentWeather(String city){  Log.d(TAG,"getCurrentWeather : Started.....");  Looper.prepare();  SyncHttpClient client = new SyncHttpClient();  String url = "http://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid="+ BuildConfig.ApiKey;  client.get(url, new AsyncHttpResponseHandler() {  @Override  public void onSuccess(int statusCode, Header[] headers, byte[] responseBody) {  String result = new String(responseBody);  Log.d(TAG, result);  try {  JSONObject responseObject = new JSONObject(result);  String currentWeather = responseObject.getJSONArray("weather").getJSONObject(0).getString("main");  String description = responseObject.getJSONArray("weather").getJSONObject(0).getString("description");  double tempInKelvin = responseObject.getJSONObject("main").getDouble("temp");  double tempInCelcius = tempInKelvin - 273;  String temperature = new DecimalFormat("##.##").format(tempInCelcius);  String title = "Weather of " + city;  String message = currentWeather + ", " + description + " with " + temperature + " celcius";  int notifId = 201;  showNotification(getApplicationContext(), title, message, notifId);  Log.d(TAG, "onSuccess: finished");  resultStatus = Result.success();  } catch (JSONException e){  e.printStackTrace();  showNotification(getApplicationContext(), "Not Success ApiKey not connected", e.getMessage(), 201);  Log.d(TAG,"onSuccess: Failed");  resultStatus = Result.failure();  }  }  @Override  public void onFailure(int statusCode, Header[] headers, byte[] responseBody, Throwable error) {  showNotification(getApplicationContext(), "Not Success", error.getMessage(), 201);  Log.d(TAG,"onFailure: Failed");  resultStatus = Result.failure();  }  });  return resultStatus;  }  private void showNotification(Context context, String title, String message, int notifId){  NotificationManager notificationManager = (NotificationManager) context.getSystemService(context.NOTIFICATION\_SERVICE);  Uri alarmSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);  NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(context, CHANNEL\_ID)  .setContentTitle(title)  .setSmallIcon(R.drawable.ic\_replay)  .setContentText(message)  .setColor(ContextCompat.getColor(context, android.R.color.transparent))  .setVibrate(new long[]{1000, 1000, 1000, 1000})  .setSound(alarmSound)  .setPriority(NotificationCompat.PRIORITY\_HIGH)  .setDefaults(NotificationCompat.DEFAULT\_ALL);  if(Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O){  NotificationChannel notificationChannel = new NotificationChannel(CHANNEL\_ID, CHANNEL\_NAME, NotificationManager.IMPORTANCE\_HIGH);  notificationChannel.enableVibration(true);  notificationChannel.setVibrationPattern(new long[]{1000, 1000, 1000, 1000});  mBuilder.setChannelId(CHANNEL\_ID);  if(notificationManager != null){  notificationManager.createNotificationChannel(notificationChannel);  }  }  Notification notification = mBuilder.build();  if(notificationManager != null){  notificationManager.notify(notifId,notification);  }  }  } |

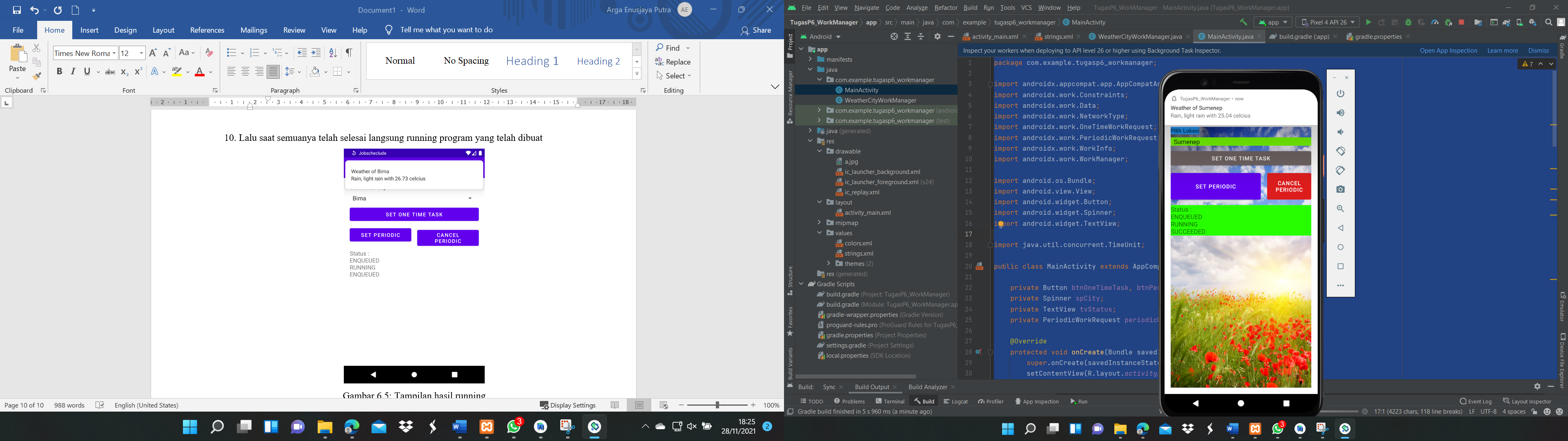
Pada Program diatass untuk melakukan pengecekan apakah apikey tersebut berhasil terpanggil atau tidak jika iya maka akan muncul kota yang kita pilih itu sedang terjadi hujan atau cuacanya cerah lalu ditambahkan Celcius dan Kelvin untuk mengukur suatu skala suhu dan akan muncul notifikasi melalui workmanager dan jika tidak berhasil akan menampilkan pesan eror.

1. Ubahlah *Code* pada *MainActivity*.*java* seperti dibawah

|  |
| --- |
| package com.example.tugasp6\_workmanager;  import androidx.appcompat.app.AppCompatActivity;  import androidx.work.Constraints;  import androidx.work.Data;  import androidx.work.NetworkType;  import androidx.work.OneTimeWorkRequest;  import androidx.work.PeriodicWorkRequest;  import androidx.work.WorkInfo;  import androidx.work.WorkManager;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.Spinner;  import android.widget.TextView;  import java.util.concurrent.TimeUnit;  public class MainActivity extends AppCompatActivity {  private Button btnOneTimeTask, btnPeriodicTask, btnCancelPeriodicTask;  private Spinner spCity;  private TextView tvStatus;  private PeriodicWorkRequest periodicWorkRequest;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  btnOneTimeTask = findViewById(R.id.btn\_one\_time\_task);  btnPeriodicTask = findViewById(R.id.btn\_periodic\_task);  btnCancelPeriodicTask = findViewById(R.id.btn\_cancel\_periodic\_task);  spCity = findViewById(R.id.sp\_city);  tvStatus = findViewById(R.id.tv\_status);  btnOneTimeTask.setOnClickListener(new View.OnClickListener(){  @Override  public void onClick(View v){  startOneTimeTask();  }  });  btnPeriodicTask.setOnClickListener(new View.OnClickListener(){  @Override  public void onClick(View v){  startPeriodicTask();  }  });  btnCancelPeriodicTask.setOnClickListener(new View.OnClickListener(){  @Override  public void onClick(View v){  cancelPeriodicTask();  }  });  }  private void startOneTimeTask(){  tvStatus.setText("Status : " );  Data data = new Data.Builder()  .putString(WeatherCityWorkManager.EXTRA\_CITY,spCity.getSelectedItem().toString())  .build();  Constraints constraints = new Constraints.Builder()  .setRequiredNetworkType(NetworkType.CONNECTED)  .build();  OneTimeWorkRequest oneTimeWorkRequest = new OneTimeWorkRequest.Builder(WeatherCityWorkManager.class)  .setInputData(data)  .setConstraints(constraints)  .build();  WorkManager.getInstance(MainActivity.this).enqueue(oneTimeWorkRequest);  WorkManager.getInstance(MainActivity.this)  .getWorkInfoByIdLiveData(oneTimeWorkRequest.getId())  .observe(MainActivity.this, workInfo -> {  String status = workInfo.getState().name();  tvStatus.append("\n" + status);  });  }  private void startPeriodicTask(){  tvStatus.setText("Status : " );  Data data = new Data.Builder()  .putString(WeatherCityWorkManager.EXTRA\_CITY,spCity.getSelectedItem().toString())  .build();  Constraints constraints = new Constraints.Builder()  .setRequiredNetworkType(NetworkType.CONNECTED)  .build();  periodicWorkRequest = new PeriodicWorkRequest.Builder(WeatherCityWorkManager.class, 15, TimeUnit.MINUTES)  .setInputData(data)  .setConstraints(constraints)  .build();  WorkManager.getInstance(MainActivity.this).enqueue(periodicWorkRequest);  WorkManager.getInstance(MainActivity.this)  .getWorkInfoByIdLiveData(periodicWorkRequest.getId())  .observe(MainActivity.this, workInfo -> {  String status = workInfo.getState().name();  tvStatus.append("\n" + status);  btnCancelPeriodicTask.setEnabled(false);  if (workInfo.getState() == WorkInfo.State.ENQUEUED){  btnCancelPeriodicTask.setEnabled(true);  }  });  }  private void cancelPeriodicTask(){  WorkManager.getInstance(MainActivity.this).cancelWorkById(periodicWorkRequest.getId());  }  } |

Pada *Source* *Code* digunakan untuk menunjukan ketika mengklik suatu button akan menampilkan status atau hasil apakah berhasil atau tidak dan ini tergantung pada benarnya APIKey yang telah kita masukan apabila APIKey tersebut salah maka dalam Status akan mengeluarkan pesan FAILED yang berarti data yang terdapat pada open weather gagal diambil.

1. Lalu saat semuanya telah selesai langsung running program yang telah dibuat



Gambar 6.5: Tampilan hasil running

1. **Link repository github**

https://github.com/andriantamvan24/tugas6-1918040-AndrianKurniawan